

# Trabajo Fin de Máster

## Máster en Sistema de Energía Eléctrica

### HMI Para el control y monitorización de bancada de motor con autómatas programables

Autor: Rubén Alejandro Martínez Bravo

Tutores: Pedro Cruz-Romero, Luis Fernando Castaño Castaño.

Dpto. de Ingeniería Eléctrica, Escuela Técnica  
Superior de Ingeniería  
Universidad de Sevilla

Sevilla, 2021







Trabajo Fin de Máster

Máster en Sistemas de Energía Eléctrica

# **HMI para el control y monitorización de bancada de motor con autómatas programables**

Autor:

Ing. Rubén Alejandro Martínez Bravo

Tutores:

Dr. Pedro Luis Cruz Romero

Dr. Luis Fernando Castaño Castaño

Dpto. Ingeniería Eléctrica

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2021



Proyecto Fin de Máster: HMI para el control y monitorización de bancada de motor con autómeta programable

Autor: Ing. Rubén Alejandro Martínez Bravo

Tutores: Dr Pedro Luis Cruz Romero

Dr. Luis Fernando Cataño Castaño

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2021

El Secretario del Tribunal



A toda mi familia y amigos





# AGRADECIMIENTOS

---

Inicialmente agradecer a la persona que me ha forjado y me ha entregado la fuerza para completar toda y cada una de mis metas, mi madre. Agradecer a mis abuelos por la fuerza y el amor entregado, en especial a mi abuela, estés donde estén espero que sigas estando orgullosa de mí. A mis tíos, a mi prima-hermana-ninja-amorosa y mi esposa por ser esas dos gotas de desorden que me entregan todo el amor del mundo. A mi papá por llamarme en cuando más lo necesitaba y apoyarme desde que llegué a este gran país. A todos y cada uno de mis amigos, los de allá, los de acá, los que compartimos piso, los del norte, a los compis del Máster porque si con el covid ya era bastante el apoyo fue decisivo. A los profes del Máster por su nivel, a mis excelentes tutores, a la ETSI y a esa gran ciudad que es Sevilla, nunca vi que brillara tanto el sol. Gracias a todos y todas.



# RESUMEN

---

**E**n el presente Trabajo Fin de Máster se exponen todo los pasos y peculiaridades para realizar el control de una bancada de motor con autómatas programables a través de un variador de velocidad. El desarrollo del proyecto culmina con la elaboración de un HMI para el control de la bancada que permitirá al Departamento de Ingeniería Eléctrica, facilitar a los estudiantes la comprensión del funcionamiento del control de las máquinas eléctricas en el campo de la automatización y las comunicaciones.

Inicialmente en el documento se describen los componentes del sistema para entender sus funciones dentro del mismo, así como los protocolos de comunicación y tipos de enlaces por los que están conectados.

Luego se procede a describir los pasos para establecer la comunicación entre los diferentes componentes del sistema y la configuración establecida en el variador de frecuencia, sirviendo como apoyo para la expansión futura del proyecto y la asimilación de conocimientos en el control de máquinas eléctricas y comunicaciones industriales.

Posteriormente, con el fin de explicar el desarrollo del programa para el PLC utilizado, se detalla los procedimientos ejecutados dentro del software Unity Pro XL, así como la elaboración de las pantallas de operador del HMI a partir de herramientas del propio software.

Por último, se crea un ejemplo de funcionamiento a través de una guía detallada de uso con el objetivo de mejorar la toma de contacto y facilitar el desarrollo de guías de laboratorio.

El desarrollo del proyecto y la elaboración del documento correspondiente se ejecuta y redacta de forma detallada para facilitar la comprensión del mismo, servir de bibliografía y de apoyo para cumplir los objetivos de departamento.

**Keywords:** HMI; Control de bancada de motores; Unity Pro XL.



# ABSTRACT

---

In this Master's Final Project, all the steps and peculiarities are exposed to control a engine mount with a programmable automaton through a speed variator.

The development of the project culminates with the elaboration of an HMI for the control of the bench that will allow the Department of Electrical Engineering to facilitate to the students the understanding of the operation of the control of electrical machines in the field of automation and communications.

Initially, the document describes the components of the system to understand their functions within it, as well as the communication protocols and types of links through which they are connected.

Then we proceed to describe the steps to establish communication between the different components of the system and the configuration established to the Variable frequency drive, serving as support for the future expansion of the project and the assimilation of knowledge in the control of electrical machines and industrial communications.

Subsequently, in order to explain the development of the program for the PLC used, the procedures executed within the Unity Pro XL Software are detailed, as well as the elaboration of the HMI operator screens using tools from the Software itself.

Finally, an operating example is created through a detailed user guide with the aim of improving contact making and facilitating the development of laboratory guides.

The development of the project and the preparation of the corresponding document is executed and written in detail to facilitate its understanding, serve as a bibliography and support to meet the objectives of the department.

**Keywords:** HMI; Engine bank control; Unity Pro XL.



# ÍNDICE

---

AGRADECIMIENTOS .....	ix
RESUMEN .....	xi
ABSTRACT.....	xiii
1. INTRODUCCIÓN .....	1
1.1 Motivaciones del Proyecto.....	1
1.2 Objetivos .....	1
1.3 Estructura del Documento.....	1
2 COMPONENTES Y SU CONEXIONADO.....	3
2.1 Componentes.....	3
2.1.1 Variador de Frecuencia .....	3
2.1.2 Plataforma de automatización Modicon M340 con procesador BMXP342020.....	3
2.1.3 Motor de inducción .....	4
2.1.4 Freno de histéresis.....	4
2.1.5 Codificador .....	5
2.2 Software de desarrollo Unity Pro XL .....	6
2.3 Esquema de conexionado.....	6
2.3.1 Freno de histéresis, motor de inducción y <i>encoder</i> con Altivar 71.....	7
2.3.2 Altivar 71, Modicon 340, Unity Pro XL.....	8
3 CONFIGURACIONES INICIALES .....	9
3.1 Crear un proyecto en Unity Pro .....	9
3.2 Tareas y secciones.....	11
3.3 Configuración de comunicación entre Modicon M340 y Altivar ATV71 .....	13
3.4 Configuración de comunicación entre Unity Pro XL y Modicon M340.....	14
3.5 Ajustes de Programación .....	17
3.6 Configuración de Altivar 71 .....	17
3.7 Pantallas de Operador .....	20
4 PROGRAMACIÓN Y DESARROLLO DEL HMI .....	25
4.1 Estándar de control de movimiento CiA 402.....	25
4.2 Pantallas de Operador .....	26
4.2.1 Pantalla 0: Configuraciones Iniciales.....	27
4.2.2 Pantalla 1: Marcha_Paro .....	31
4.2.3 Pantalla 2: Supervisión.....	36
.....	39
4.2.4 Pantalla 3: Guardar Configuración.....	40
5 EJEMPLO DE FUNCIONAMIENTO.....	45
5.1 Guía.....	45
6 CONCLUSIONES .....	55
Futuras líneas de investigación .....	55
REFERENCIAS.....	57
GLOSARIO DE TÉRMINOS Y SIGLAS.....	58





# ÍNDICE DE TABLAS

---

Tabla 1. Características del Altivar.....	3
Tabla 2. Módulos presentes en el autómata utilizada. ....	3
Tabla 3. Datos de chapa en función de la frecuencia del motor de inducción empleado.....	4
Tabla 4. Tabla de características del Codificador ( <i>encoder</i> ) empleado. ....	6
Tabla 5. Elementos que componen la Pantalla de Operador #0: Configuraciones Iniciales. ....	29
Tabla 6. Elementos que componen la Pantalla de operador #1: Marcha-Paro.....	34
Tabla 7. Elementos que componen la Pantalla de Operador #2: Supervisión.....	38
Tabla 8. Elementos que componen la Pantalla de Operador #3: Guardar_Configuración.....	42



# ÍNDICE DE FIGURAS

Figura 1. Datos de Chapa del Motor de inducción .....	4
Figura 2. Freno de Histéresis .....	5
Figura 3. Freno de Histéresis acoplado en la bancada .....	5
Figura 4. Codificador (encoder) incremental .....	6
Figura 5. Logotipo del programa Unity Pro XL .....	6
Figura 6. Símbolos de señales utilizados .....	7
Figura 7. Conexionado entre bancada con frenos de histéresis y codificador, con Altivar 71 .....	7
Figura 8. Conexionado entre Altivar 71 con PLC Modicon M340 y el software Unity Pro XL ejecutado en una PC .....	8
Figura 9. Opción para crear nuevo proyecto .....	9
Figura 10. Selección del procesador y bastidor .....	9
Figura 11. Elección de firmware compatible con PLC .....	10
Figura 12. Bastidor con fuente, procesador y cinco SLOTS disponibles .....	10
Figura 13. Menú de selección de módulos digitales .....	11
Figura 14.. Menú de selección de módulos analógicos .....	11
Figura 15. Tipos de tareas del sistema .....	12
Figura 16. Ruta para añadir nueva sección .....	12
Figura 17. Ventana para nombrar programa y seleccionar lenguaje de programación .....	12
Figura 18. Ventana de programación .....	13
Figura 19. Ruta de configuración de SerialPort y Ethernet .....	13
Figura 20. Ventana de configuración serie (Modbus RTU) .....	14
Figura 21. Ruta para añadir una nueva red .....	14
Figura 22. Nombrar y seleccionar tipo de red .....	14
Figura 23. Red EtherNet creada .....	15
Figura 24. Configuración de la red EtherNet .....	15
Figura 25. Ubicación del botón Validar .....	15
Figura 26. Ruta de Puerto EtherNet del procesador BMX P34 2020 .....	16
Figura 27. Insertar configuración Ethernet creada .....	16
Figura 28. Selección de configuración creada .....	16
Figura 29. Antes y después de cargar la configuración .....	16
Figura 30. Pestañas Seleccionadas en Ajustes del Proyecto .....	17
Figura 31. Insertar los parámetros del motor en el ATV71 .....	18
Figura 32. Ejecución de AUTOAJUSTE .....	18
Figura 33. Valor por defecto de I térmica = 3.5 A. ....	18
Figura 34. Valor establecido de límite de intensidad .....	18
Figura 35. Configuración de codificador en Altivar 71 .....	19
Figura 36. El control por defecto es SVC por U .....	19
Figura 37. Inicialmente se selecciona el Perfil E/S luego se selecciona “Separado” .....	19
Figura 38. Configuración Modbus del ATV71 .....	20
Figura 39. Configurar REARRANQUE AUTOMÁTICO .....	20
Figura 40. Añadir una nueva Pantalla de Operador .....	20
Figura 41. Asignar Valor a Pantalla de Operador .....	20
Figura 42. Nombres de las Pantallas del Sistema .....	21
Figura 43. IOSEditos, Opciones de la barra de herramienta para el diseño .....	21
Figura 44. Ubicación de librería de diseños .....	21
Figura 45. Ejemplo de figura prediseñadas .....	21
Figura 46. Ruta de tabla de variables elementales .....	22
Figura 47. Ejemplo de una variable tipo BOOL creada .....	22
Figura 48. Botón de comando dentro de IOSEditos .....	22
Figura 49. Botón de comando después de creado .....	22
Figura 50. Propiedades de Objeto de Entrada .....	23
Figura 51. Instancia del Botón de Comando .....	23
Figura 52. Mapa de estados de estándar CiA402 .....	25
Figura 53. Lógica de programación de la Pantalla de Operador: Configuraciones Iniciales .....	28

# ÍNDICE DE FIGURAS

Figura 54 Diseño de la Pantalla de Operador: Configuraciones iniciales.....	30
Figura 55. Selección de parada configurada en el ATV71. ....	31
Figura 56. Advertencia con parada configurada.....	31
Figura 57. Lógica de programa de la Pantalla de operador: Marcha-Paro.....	32
Figura 58. Diagrama de Flujo de programa de la función Control Manual. ....	33
Figura 59. Diseño de la pantalla de operador. Marcha-Paro.....	35
Figura 60. Bloques de Funciones Derivadas creadas para mostrar en formato texto los mensajes de error del manual. ....	36
Figura 61. Lógica de programa de la Pantalla de operador: Supervisión. ....	37
Figura 62. Diseño de la Pantalla de Operador: Supervisión.....	39
Figura 63. Opción del el Altivar 71 para introducir el código de acceso que permite el trabajo con memoria ....	40
Figura 64. Lógica de programación de la Pantalla de Operador: Configuraciones Iniciales.....	41
Figura 65. Diseño de la Pantalla de Operador: Guardar_Configuración.....	43
Figura 66. Pantalla de operador que componen el HMI.....	45
Figura 67. Botón “Conectar”.....	45
Figura 68. Botón “RUN”.....	45
Figura 69. Botón “Enviar Proyecto a PLC”.....	45
Figura 70. Botón “Habilitar Variables de escritura”.....	45
Figura 71. Botón de HMI en Pantalla Completa.....	45
Figura 72. Parte superior de la pantalla de operador “Configuraciones Iniciales”.....	46
Figura 73. Mapa de Estado CiA402 dispuesto en la parte baja de Configuraciones Iniciales.....	46
Figura 74. Pantalla Configuraciones Iniciales al presionar "Configurar Conexión".....	47
Figura 75. Mapa de estado en la parte inferior de la pantalla mostrando el Estado 3 como estado activo. ....	47
Figura 76. Mapa de estado, con Estado 4 activo, luego de presionar “2. Cerrar Interruptor”. ....	48
Figura 77. Tipo de Parada de Emergencia seleccionada por defecto.....	48
Figura 78. Valor de Referencia Inicial.....	48
Figura 79. Indicador señalando que el Motor está detenido. ....	48
Figura 80. Motor en Marcha.....	48
Figura 81. Estado 5 Activo, motor en funcionamiento. ....	49
Figura 82. Animación de motor para mostrar el sentido de giro. ....	49
Figura 83. Advertencia indicando que el sistema permanece en Estado.....	49
Figura 84. Al ejecutar una Parada de Emergencia a través de Paro Configurado, se permanece en Estado 5.....	49
Figura 85. Orden de parada siguiendo el Botón de Emergencia.....	50
Figura 86. Motor detenido y desconectado, Estado 2 activo. ....	50
Figura 87. Selección de tipo de parada: Parada Rápida y esperar a operador.....	50
Figura 88. Se ejecuta nuevamente una parada de Emergencia.....	50
Figura 89. Estado 6 Activo, pues se modificó QSTD.....	50
Figura 90. La ventana de operador “Supervisión” cobra importación mientras el estado permanezca en Estado 5. ....	51
Figura 91. Parte superior de la Pantalla de Operador “Supervisión” donde se monitorizan los valores.....	51
Figura 92. El sistema antes pequeñas variaciones en la referencia de frecuencia.....	52
Figura 93. Parte superior de la pantalla "Guardar_Configuración".....	52
Figura 94. Pantalla "Guarda_Configuración" con las nuevas opciones después de introducir el PIN.....	53
Figura 95. Montaje en laboratorio de todos lo equipos del sistema.....	53

# 1. INTRODUCCIÓN

---

**E**n este capítulo introductorio se presentan las motivaciones que dieron lugar a este proyecto, los objetivos que persigue y como estarán compuestos cada uno de sus seis capítulos.

## 1.1 Motivaciones del Proyecto

Dentro del ámbito universitario en todas las ramas de la ingeniería es necesario contar con herramientas que aporten conocimientos prácticos en el control de máquinas y comunicaciones industriales.

Al momento de comenzar con la investigación, el Departamento de Ingeniería Eléctrica contaba con los equipos y licencias que permitían desarrollar un HMI para controlar un banco de pruebas en desuso, lo cual fue resignificado para crear una herramienta útil en estas áreas.

Para ello, era necesario documentar el control de un sistema compuesto por un PLC, un variador de frecuencia, una bancada con codificador y freno electromagnético, a través de una licencia de Unity Pro XL. De esta forma, como resultado, se obtiene una herramienta para las prácticas de laboratorio y un documento donde se detalla la creación del HMI, la configuración de los equipos y los protocolos que los une.

La explotación de este sistema no solo complementará las prácticas de asignaturas relacionadas con las máquinas eléctricas y las comunicaciones industriales, sino que los estudiantes dispondrán de una guía documentada sobre la creación de estos sistemas dentro del Departamento de Ingeniería Eléctrica de la Escuela Técnica Superior de Ingeniería de la Universidad de Sevilla.

## 1.2 Objetivos

El objetivo principal del proyecto es *el desarrollo de un HMI para el control y monitorización de bancada de motor con autómatas programables*. Con el desarrollo del mismo el Departamento de Ingeniería Eléctrica contará con una herramienta para mejorar las prácticas de asignaturas relacionadas con las máquinas eléctricas y comunicaciones; de esta forma se profundizará en los conocimientos de control y comunicación industrial. Para ello, se ha dividido el documento en varios capítulos orientados a explicar de forma intuitiva cada una de las partes, protocolos de comunicación empleados y lógica de programación seguida. El HMI se ejecutará desde un ordenador que enviará las consignas a un PLC, encargado de controlar un variador de frecuencia conectado a una bancada compuesta por un motor, con freno de histéresis y codificador. Este objetivo principal se desglosa en los siguientes pasos detallados:

- 1-Descripción de cada uno de los elementos que componen el sistema, su conexión y la relación entre ellos.
- 2-Exponer la configuración realizada en el variador de velocidad y en el software para conseguir un estado inicial a partir del cual se ejecuten correctamente los programas.
- 3-Desarrollo de los programas de control y las pantallas de operador que componen el HMI.
- 4-Presentar mediante un ejemplo de uso del Proyecto Final, que exponga las fortalezas y futuras líneas de desarrollo.

## 1.3 Estructura del Documento

El documento estará compuesto por seis capítulos donde se explica el desarrollo del proyecto y su composición, para el cumplimiento de los objetivos.

### Capítulo 1. Introducción

En este capítulo se exponen las motivaciones que dieron lugar al Trabajo Fin de Máster, las problemáticas que se resolverán, los objetivos para cumplir cada una de las metas y la propia estructura del documento.

### Capítulo 2. Componentes y su conexionado

Con este segundo capítulo se detallan las características principales de cada uno de los componentes, el tipo de acoplamiento y los principales ajustes de los protocolos de comunicación presentes.

### Capítulo 3. Configuraciones iniciales

Dada la importancia del documento de explicar de forma detalla la creación del proyecto y el cumplimiento de los objetivos didácticos, en este capítulo se detallan tanto los pasos iniciales en el Software Unity Pro XL donde se creará el HMI, como la configuración establecida en el Variador de Frecuencia.

### Capítulo 4. Programación y desarrollo del HMI

Este capítulo será el encargado de explicar la lógica de programación seguida, las instrucciones ejecutadas y el desarrollo de las pantallas de operador dentro del Unity pro XL.

### Capítulo 5. Ejemplos de funcionamiento

Para hacer más amigable la toma de contacto con el HMI, se creará una guía que evidencie el funcionamiento del sistema siguiendo una línea de instrucciones.

### Capítulo 6. Conclusiones

Es el capítulo donde se muestran las conclusiones sobre el cumplimiento de los objetivos del proyecto, se valoran el impacto de las decisiones que se tomaron y se identifican futuras líneas de investigación.

## 2 COMPONENTES Y SU CONEXIONADO

**D**urante el capítulo se exponen los componentes del proyecto, como están conectados unos con otros y el software utilizados para el desarrollo del HMI.

### 2.1 Componentes

#### 2.1.1 Variador de Frecuencia

Utilizar un variador de frecuencia, además de mejorar la eficiencia durante el uso de un motor de inducción, nos entrega una amplia versatilidad en el control del mismo. El modelo en particular el ATV71H075M3 de Telemecanique, dispone tanto de comunicación CANopen como Modbus, pero para el desarrollo del proyecto utilizaremos este último protocolo mediante el conector RJ45 en cara frontal.

Con la ayuda de este variador de velocidad y con el codificador 2RMHF 1024-D de tipo incremental, se podrá cerrar el lazo de control con los parámetros que se fijarán desde el autómatas. Las principales especificaciones de mayor relevancia para este proyecto se detallan en la Tabla 1.

Tabla 1. Características del Altivar

Características	Valor
Potencia de motor	0.37 kW (monofásica) 0.75 kW (trifásica)
Potencia en HP	1 Hp (trifásica) 0.5 Hp (monofásica)
Tensión de Alimentación	200 – 240 V
Filtro CEM <sup>1</sup>	Integrado
Potencia Aparente	1.4 kVA en 240 V monofásica 2.2 kVA en 240 V trifásica
Corriente de CC de la Red	5 kA
Protocolo del puerto de comunicación	Modbus CANopen
Interfaz física para Modbus	2-wire RS 485
Velocidad de Transmisión Modbus (Conector RJ45 en cara frontal)	9600 bps, 19200 bps

#### 2.1.2 Plataforma de automatización Modicon M340 con procesador BMXP342020

El PLC Modicon M340 es un autómatas modular de alto rendimiento y ampliable pensado para automatización distribuida dentro de un SCADA, concebido por Telemecanique, marca de Schneider Electric, como pieza de su arquitectura Transparent Ready<sup>2</sup>. En el desarrollo del proyecto se utilizará como sistema aislado, pero al estar dotado de Conexión Ethernet TCP/IP ofrece soluciones de automatización y control remoto [1] para futuros proyectos de esta línea.

Para el desarrollo de este trabajo de investigación se utilizó una maqueta pedagógica compuesta por los siguientes módulos.

Tabla 2. Módulos presentes en el autómatas utilizada.

Módulo	Características
Rack	BMX XBP 0600
Fuente	BMX CPS 2000
Procesador	BMX P34 2020
E/S Digitales	BMX DDM 16022
E/S Analógicas	BMX AMM 0600

<sup>1</sup> Filtro de compatibilidad electromagnética elimina los efectos no deseados de las interferencias de este tipo

<sup>2</sup> Arquitectura para operar como una sola red aglutinando EtherNet/IP y Modbus/TCP

El procesador BMX P342020 cuenta con 4096 kB de memoria tipo RAM, así como 256 kB de datos y 3584 kB para contantes y símbolos del sistema del tipo EEPROM. Dispone de conteo rápido, registro con 64 tareas de eventos, comunicación Modbus RTU/ASCII sobre RS485 y RS232C, Ehternet TCP/IP y USB. Con estas características es más que suficiente para el desarrollo del proyecto, donde el PLC para controlar el Altivar 71, enviará las referencias a seguir y recogerá los datos a mostrar en la interfaz de operario en la PC.

**2.1.3 Motor de inducción**

El motor presente en la bancada es un motor de inducción asíncrono, trifásico y con rotor de jaula de ardilla. Con este se pretende acercar a los estudiantes al control de un motor ampliamente usado en la industria [2] presente en diferentes equipos mecánicos, bombas, cintas transportadoras, entre otras aplicaciones.

Analizando los datos del catálogo del fabricante [3] se obtiene que es un motor de 4 polos, con cuerpo de aluminio, ventilación exterior IC 411, de servicio continuo S1, aislamiento de clase F y protección IP 65. La información eléctrica de la placa de características se muestra y detalla a continuación.

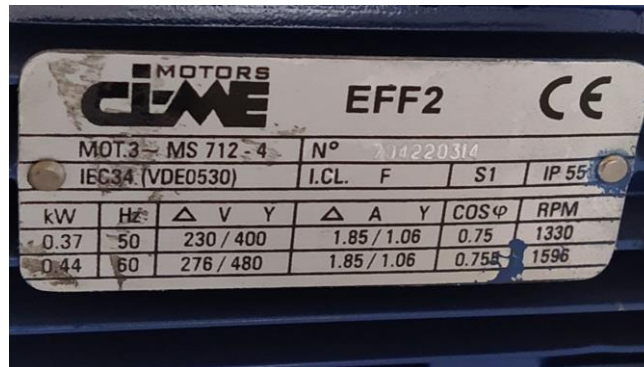


Figura 1. Datos de Chapa del Motor de inducción

Tabla 3. Datos de chapa en función de la frecuencia del motor de inducción empleado.

Característica	kW	Tensión (V) con Conexión Triángulo/Estrella	Intensidad (A) con Conexión Triángulo/Estrella	Factor de potencia (cos φ)	Revoluciones por minuto (RPM)
Valores a 50 HZ	0.37	230 / 400	1.85 / 1.06	0.75	1330
Valores a 60 Hz	0.44	276 / 480	1.85 / 1.06	0.755	1596

**2.1.4 Freno de histéresis**

Este tipo de freno también nombrado como freno de corrientes parásitas, freno magnético, de inducción o en los países anglosajones como “Eddy current brake”, se utiliza para detener o ralentizar objetos disipando energía en forma de calor. Tiene aplicación en muchas industrias, sobre todo en la automoción y equipos de gran tonelaje como trenes y camiones de carga [4].

Para entender su funcionamiento y ayudar a la comprensión del mismo, analizamos como la ley de Faraday y Lenz de inducción electromagnética intervienen en este equipo.

En presencia de un flujo magnético y si este flujo cambia con el tiempo, bien porque cambia el campo, cambia la superficie o la posición relativa entre estos, aparece una FEM inducida proporcional a los cambios anteriormente descritos, y esta a su vez, provoca que aparezcan corrientes en el inducido. Teniendo en cuenta la ley de Lenz que nos plantea que estas FEM inducida se opone a la variación del



flujo magnético que las produjo, ante un conductor con volumen no despreciable, las corrientes que aparecen en el inducido reciben el nombre de corrientes parásitas o corrientes de Foucault<sup>3</sup>. Estas a su vez crean en el conductor un efecto de electroimanes que se oponen al campo magnético aplicado, con una fuerza proporcional al movimiento lo suficientemente fuerte como para frenar el eje del motor utilizado en este proyecto [5] [6].

De esta manera, al carecer de contacto físicos le otorga un bajo desgaste y alto índice de repetibilidad a este equipo que, sumado a la posibilidad de variar la intensidad del mismo, se muestra como un complemento perfecto para las simulaciones con fines didácticos que se describirán en este proyecto [7].



Figura 2. Freno de Histéresis



Figura 3. Freno de Histéresis acoplado en la bancada.

### 2.1.5 Codificador

Para cerrar el lazo de control y leer la velocidad en tiempo real en el eje del motor, se utiliza un codificador, “enconder” en inglés, incremental en el proyecto. Específicamente el modelo 2RMHF 1024-D del fabricante SCANCON. Su bajo precio en el mercado lo hacen idóneo para este tipo de aplicaciones y su uso con fines de referencia y seguridad, permiten desarrollar el control más avanzado frente al control en lazo abierto. Por lo que, a pesar del avance de los nuevos variadores de frecuencia, sigue siendo un elemento imprescindible para el control de motores de inducción.

Este en particular, es un enconder incremental electromecánico, con salida de controlador de línea. Es capaz de entregar un tren de pulso equivalente a la velocidad de giro que desempeña el eje del motor y utiliza una salida diferencial que entrega prestaciones como gran inmunidad al ruido e interferencias, de cara a su instalación. La familia de estos codificadores mayoritariamente entrega dos pulsos, A y B, con los que podemos conocer el sentido de giro, pero en ambientes con grandes cantidades de ruido eléctrico se recomienda utilizar aquellos que dispongan, además, de sus “complementarias”  $\bar{A}$  y  $\bar{B}$ . De esta forma se puede aumentar la distancia de cableado y ser más eficiente de cara al ruido dado que en este caso no se comparte una referencia, sino que el resultado del tren de pulsos es la diferencia entre la señal con la información y la complementaria de la señal original [8].

---

<sup>3</sup> Descubierta por el físico francés Léon Foucault en 1851, también llamadas "corrientes torbellino o turbillonarias" y como eddy current en inglés.



Figura 4. Codificador (encoder) incremental.

Tabla 4. Tabla de características del Codificador (encoder) empleado.

Característica	Valor
Tipo	Incremental
Alimentación	4.5 Vdc – 30 Vdc / 35 mA Máx
Resolución	1024 ppr (pulsos por revolución)
Salida	Dos canales en cuadratura A, B y sus complementarios $\bar{A}$ y $\bar{B}$
Tipo de Salida	Controlador de Línea Diferencial

## 2.2 Software de desarrollo Unity Pro XL

Este software tiene como finalidad programar diferentes plataformas de Autómatas Programables dentro de las marcas de Telemecanique y Schneider. La plataforma utilizada en el proyecto es la Modicon M340, totalmente compatible con el mismo. Con el uso de este software y ventanas muy organizadas, gestores de variables y pantallas de operador, se logra crear tanto el fichero que ejecutará el PLC, como el HMI para su control y monitorización.



Figura 5. Logotipo del programa Unity Pro XL

Permite crear y ejecutar diferentes tipos de tareas, estas pueden ser de tipo Eventos, Rápidas (FAST) y MAST (Principal), con este mismo orden de prioridad. El proyecto se fragmentó en cuatros sistemas, atendiendo a cada uno de los sinópticos que componen el HMI que se crearon en la herramienta “Pantallas de Operador” de este software. Cada una de las pantallas de operador creadas en el capítulo cuatro, cuenta con una sección de programa equivalente cuya lógica se muestra a través de los diagramas de flujo dentro del mismo capítulo. De esta forma se logra una mayor organización del proyecto y capacidad de interpretación en futuras líneas de investigación.

El software es capaz de manipular los diferentes lenguajes de programación industrial del mercado y entrega la capacidad de confeccionar un HMI a través de las citadas pantallas de operador. De esta forma, a través de Unity Pro XL, se crearon las secciones del programa principal que ejecuta el PLC Modicon M340 y se diseñó la interfaz software HMI compuesta por cuatro pantallas de operador, para el control y monitorización de todo el sistema.

## 2.3 Esquema de conexionado

El sistema está compuesto por dos grupos que intercambian los valores medidos e instrucciones a través del variador de frecuencia ATV71. El primer grupo, descrito en el apartado 2.3.1, está compuesto por un motor de inducción, un freno de histéresis para simular cargas variables y un codificador para medir

la velocidad en el eje del motor. Por otra parte, el grupo del apartado 2.3.2 está compuesto por la inteligencia a nivel de campo (Modicon M340), el ATV 71 como maestro del Bus de Campo y por el HMI ejecutado en Unity Pro a nivel de supervisión.

Para describir el esquema de conexionado se utilizaron, imágenes que facilitan identificar los componentes (Figura 7 y Figura 8) sin embargo para la conexión entre los mismo se utilizaron los símbolos de las señales expuestas en las normas ANSI/ISA – S5.1 cuya leyenda de los enlaces presentes se muestran a continuación (Figura 6).

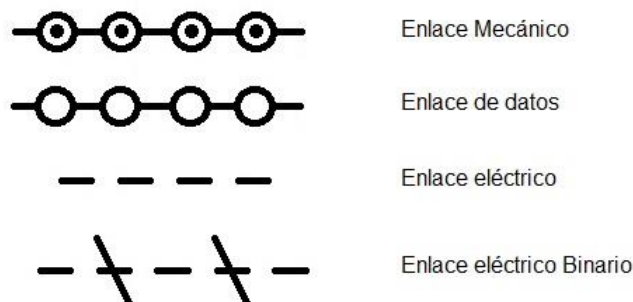


Figura 6. Símbolos de señales utilizados

### 2.3.1 Freno de histéresis, motor de inducción y *encoder* con Altivar 71

El freno de histéresis tiene en su interior un disco de aluminio acoplado al eje del rotor del motor de inducción. Este último, a su vez, tiene acoplado por el otro extremo un codificador que envía un tren de pulsos equivalente a la velocidad en el eje del motor de inducción al variador de velocidad. Con este dispositivo conseguimos cerrar el lazo de control del motor y al variar la intensidad del freno electromagnético, podemos simular cambios de carga, crear diferentes escenarios y observar el comportamiento del sistema en general en cada uno.

El freno posee un selector de intensidad (Figura 3), pudiendo variar el valor de la potencia de frenado, observar los cambios en la pantalla de usuario y de esta forma la simulación de cada situación puedes ser estudiada y conocer en tiempo real el comportamiento de las diferentes variables.

El codificador envía un tren de pulsos (1024 por revolución) a través de par trenzado, equivalentes a la velocidad que registra en el eje del motor. La presencia de este sensor, además de entregar valores reales para seguimiento de la referencia, permite ser utilizado con fines de seguridad, pues en un control en lazo abierto es menos probable detectar fallos mecánicos e incluso un sentido de giro erróneo de cara al variador.



Figura 7. Conexionado entre bancada con frenos de histéresis y codificador, con Altivar 71.

### 2.3.2 Altivar 71, Modicon 340, Unity Pro XL

El apartado de control y monitorización del sistema lo componen tres dispositivos: El variador de frecuencia Altivar ATV71, el autómatas Modicon M340 y un ordenador ejecutando Unity Pro XL.

El enlace entre el Altivar ATV71 y el PLC M340 se realiza con par trenzado y apantallado con conector RJ45. La configuración Modbus en este enlace es la siguiente: velocidad de 19 200 baudios, 8 bit de datos y 1 de parada, con 10 segundo de espera, el variador es configurado como esclavo por defecto, mientras que el autómatas se ha configurado como maestro controlando el módulo de procesamiento BMX P342020 del M340.

La conexión entre el autómatas y el ordenador que ejecuta el software Unity Pro XL, se realiza mediante un enlace Ethernet TCP/IP sobre par trenzado y apantallado con conector RJ45. El uso de este estándar, con alto nivel de penetración en la industria, puede exportar los conocimientos que se adquieran con este proyecto a sistemas de mayor escala de automatización.

El autómatas (Modicon M340) actúa como intermediario entre la supervisión de las pantallas de operador y el Altivar, pieza clave de inteligencia a nivel de Bus del grupo descrito en el apartado anterior. Las órdenes que se realizan en las Pantallas de operador, que funciona como interfaz hombre - maquina ("HMI"), son interpretadas y manipuladas por el PLC para comunicarse con el Variador y ejecutarlas. El flujo de información también es recíproco, el PLC normaliza la información que le envía y mide el variador de frecuencia a mostrar en el HMI, así como almacenar el estado de las variables y del sistema en general.



Figura 8. Conexión entre Altivar 71 con PLC Modicon M340 y el software Unity Pro XL ejecutado en una PC

# 3 CONFIGURACIONES INICIALES

En este capítulo se pretenden describir los pasos iniciales que se desarrollaron tanto en el software Unity Pro XL como en el variador de frecuencia ATV71. Concretamente cómo se crea la base sobre la que se desarrolla el proyecto actual y sobre la que se puede hacer cualquier sistema con el hardware descrito en el Capítulo 2.

Se pretende abarcar desde la familiarización con el software creando el proyecto inicial, las secciones dentro del programa principal y el establecimiento de la comunicación, hasta los parámetros eléctricos en el variador de frecuencia y la configuración para la comunicación de este último con el PLC.

## 3.1 Crear un proyecto en Unity Pro

Una vez en Unity Pro XL, para crear un nuevo proyecto es necesario ir al menú superior *Fichero* -> *Nuevo* para pasar a seleccionar los componentes del PLC.

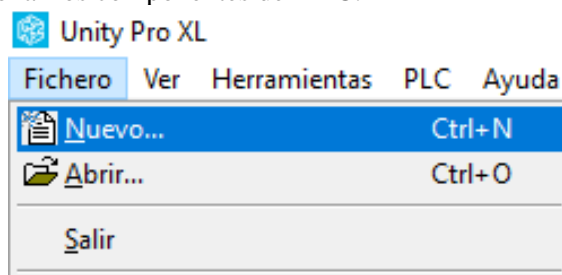


Figura 9. Opción para crear nuevo proyecto

En el PLC del proyecto disponemos de una CPU de tipo BMX P342020 con comunicación Modbus y Ethernet en un bastidor BMX XBP 0600, se seleccionan ambas como se muestra en la Figura 10 y presionamos en Aceptar.

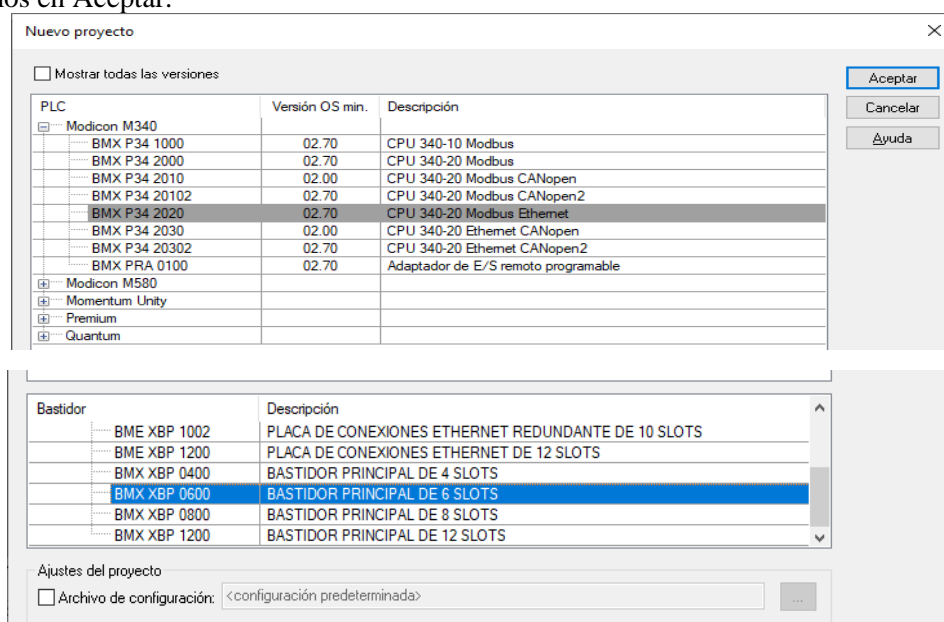


Figura 10. Selección del procesador y bastidor.

En el proyecto no se utilizan las entradas analógicas o digitales, pero para futuras líneas pueden ser de mucha ayuda y dado que la maqueta didáctica en cuestión dispone de estos módulos, se añadirán para evitar posibles mensajes de error.

Como se puede observar en la figura 10, mostrada anteriormente, la versión mínima del software del PLC es la 02.70; pero el UnityPro cuenta con un modo de compatibilidad donde le podremos asignar la versión 1.00 que tiene el firmware del procesador utilizado (Figura 11).

Para ellos seguimos la ruta dentro de la vista estructural a la izquierda de la ventana. Configuración -> Bus PLC -> BMX XBP 0600 y doble clic en este último. Esta acción muestra la ventana que aparece a la derecha de la siguiente imagen y en el desplegable superior se puede seleccionar la versión del firmware de la licencia del procesador en cuestión. En este caso se selecciona la versión 1.00.

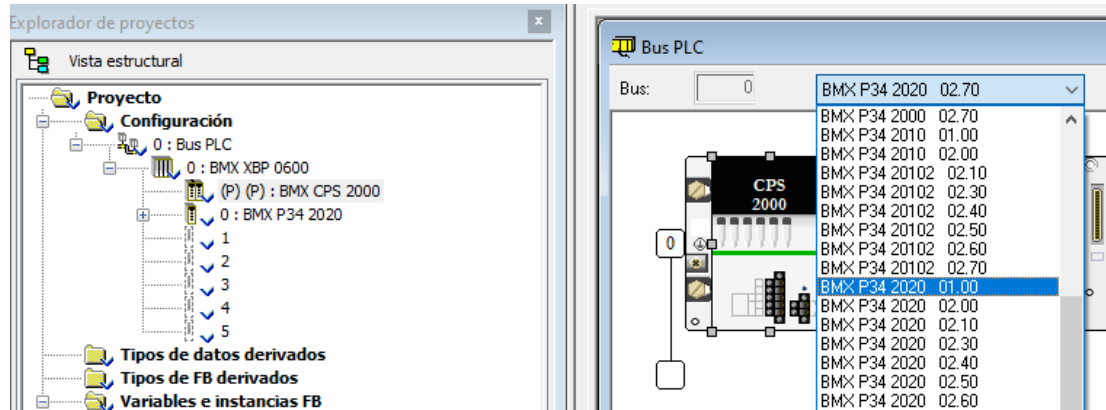


Figura 11. Elección de firmware compatible con PLC.

Con la configuración establecida hasta el momento es suficiente para realizar un control Modbus de cualquier esclavo conectado al PLC, pero el PLC utilizado dispone de otros dos módulos disponibles, uno de E/S digitales y otro de E/S analógicas. En futuros proyectos ambos pueden resultar necesarios para ampliaciones de este proyecto por lo que se describirá cómo añadir cada uno.

En esta figura 12 se muestra una imagen del bastidor su fuente de alimentación predeterminada y cinco slots, donde el primero (El número "0") está ocupado por el procesador. Al dar doble clic sobre el slot 1 (como muestra la foto) nos aparece la ventana de la figura 13 a partir de la cual iremos añadiendo los módulos de nuestro sistema descritos en el Capítulo 2.

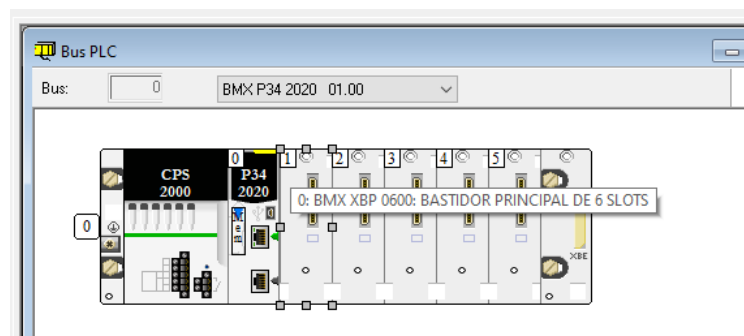


Figura 12. Bastidor con fuente, procesador y cinco SLOTS disponibles

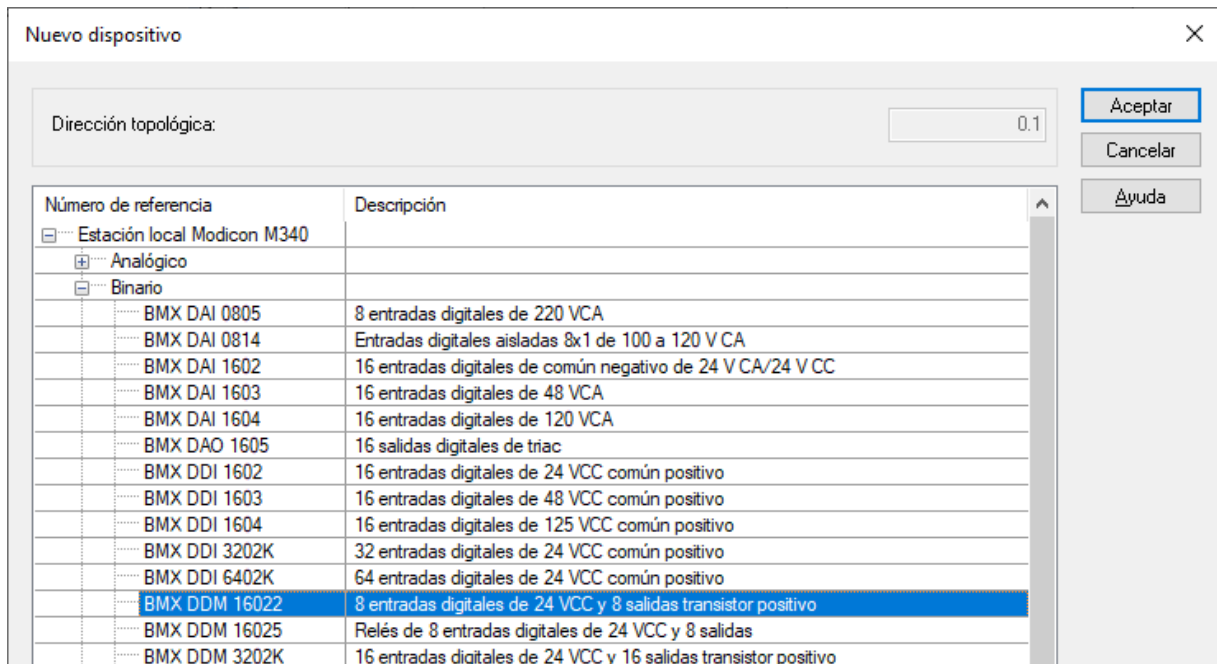


Figura 13. Menú de selección de módulos digitales

Inicialmente se añade el módulo de entradas y salidas digitales, el BMX DDM16022 de 8 entradas digitales y 8 salidas de lógica positiva. Presionando en aceptar se añade al Bastidor. Con el módulo de las entradas y salidas analógicas se repite el proceso, pero clicando en el *slot 2* y a continuación seleccionando dentro de los módulos analógicos el AMM 0600 con 4 entradas y dos salidas analógicas.

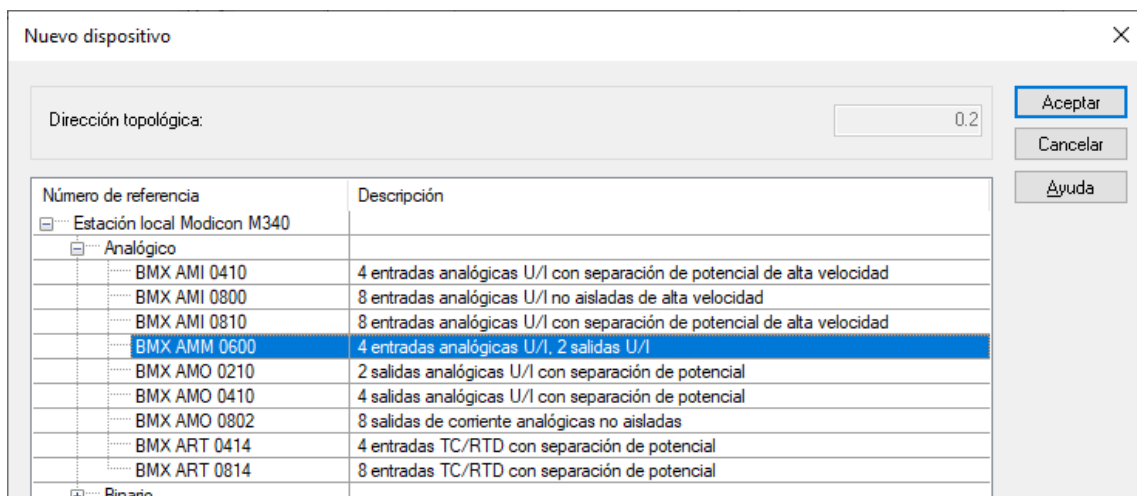


Figura 14.. Menú de selección de módulos analógicos

### 3.2 Tareas y secciones

Una vez seleccionados todos los componentes pasamos a crear la ventana de programación.

En Unity Pro XL cada programa está dividido en tareas que pueden ser programadas en distintos lenguajes de programación en secciones y subrutinas. En dependencia del procesador, puede operar en modo simple o modo multitarea. En modo simple solo se ejecuta la tarea principal (MAST) que a su vez se descompone en secciones de códigos y subrutinas sin importar el modelo del procesador. El procesador en cuestión (BMX P342020) opera en multitarea, y aunque en el proyecto solo se utiliza la tarea MAST, a continuación, se describirán las tareas FAST y Eventos que pueden ser muy útiles en

futuros proyectos donde se requieran mayores ciclos de lecturas y distintos niveles de prioridades.

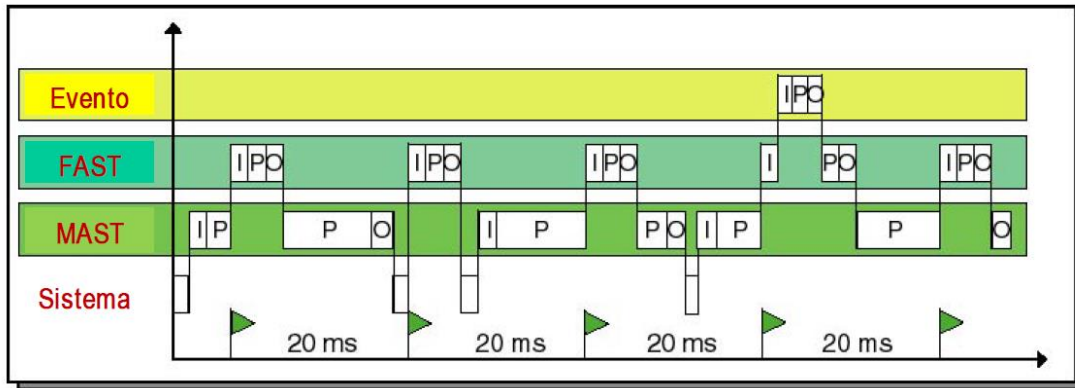


Figura 15. Tipos de tareas del sistema.

En esta figura 15 se observan los tres diferentes tipos de tareas, los intervalos que componen la figura determinan un ciclo de lectura o entrada (I), intervalo de procesamiento (P) y de salidas o resultados (O). De esta manera se puede apreciar como la más prioritaria es la tarea de Evento, luego las tareas FAST y después la tarea MAST (Principal) que se ejecuta casi a la velocidad de ejecución de un ciclo completo.

Como anteriormente se menciona, el proyecto hará uso de una sola tarea MAST para simplificar su estudio y entendimiento de la lógica de programación. Para ello, a continuación, se detallan los pasos de su creación.

En el explorador de proyectos se dirige a *Programa -> Tareas -> MAST -> Secciones*, se realiza clic derecho en esta opción y luego en *Nueva Sección* como se muestra en la imagen (Figura 16). Esta acción muestra una ventana donde se puede nombrar la sección y escoger el lenguaje de programación (Figura 17).

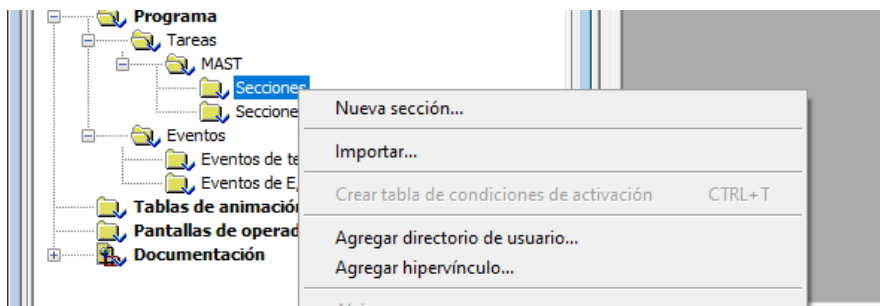


Figura 16. Ruta para añadir nueva sección.

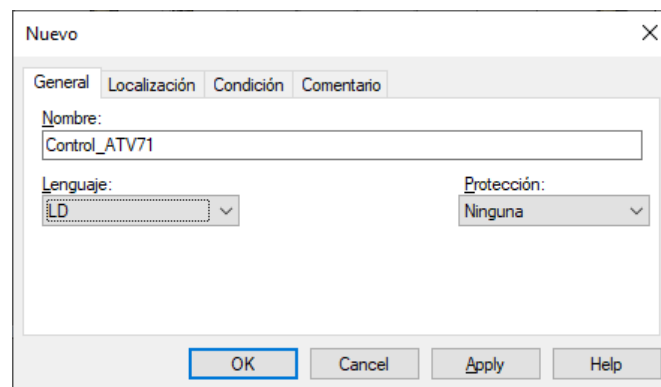


Figura 17. Ventana para nombrar programa y seleccionar lenguaje de programación



En este caso, a modo de ejemplo, la sección se nombró: “Control\_ATV71” y se seleccionó el lenguaje LD (Ladder o de Contactos<sup>4</sup>).

Al presionar “OK” se crea una sección y se abre una ventana donde podemos comenzar a añadir lógica de programación y realizar el programa como se observa en la figura 18.

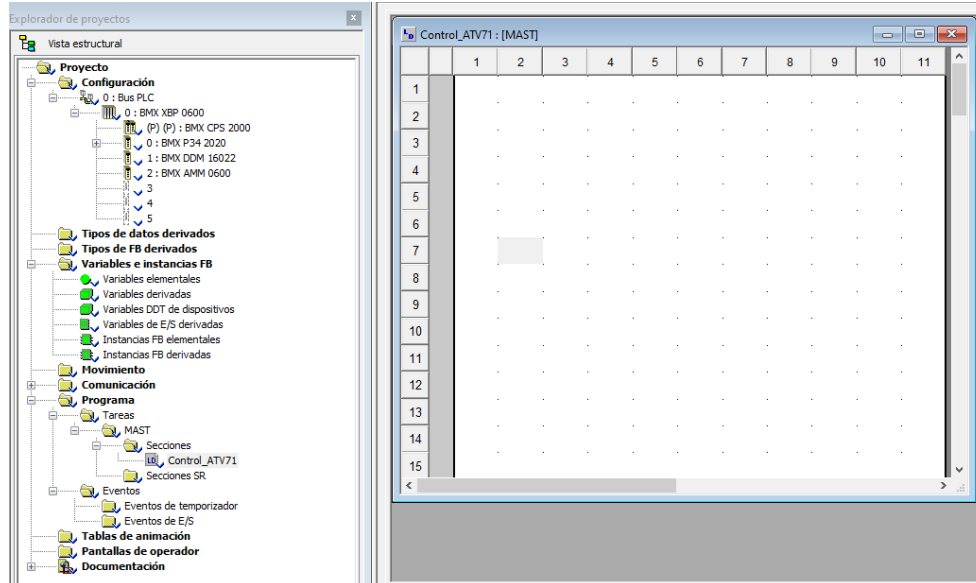


Figura 18. Ventana de programación

### 3.3 Configuración de comunicación entre Modicon M340 y Altivar ATV71

El Modicon M340 utiliza dos tipos de comunicaciones para realizar sus funciones, una para comunicarse con el PC (Unity Pro XL, pantallas de operador) y otra para leer y enviar los parámetros al variador de velocidad. En el capítulo anterior se describieron los valores con lo que fueron configurados y ahora se procederá a mostrar la configuración que es necesario establecer desde el Unity Pro XL.

En dependencia del procesador que se disponga será la comunicación disponible. En nuestro caso, al desplegar la ventana del procesador en la *Vista estructural*, como se muestra en la figura 19, aparecen tanto el SerialPort como Ethernet.

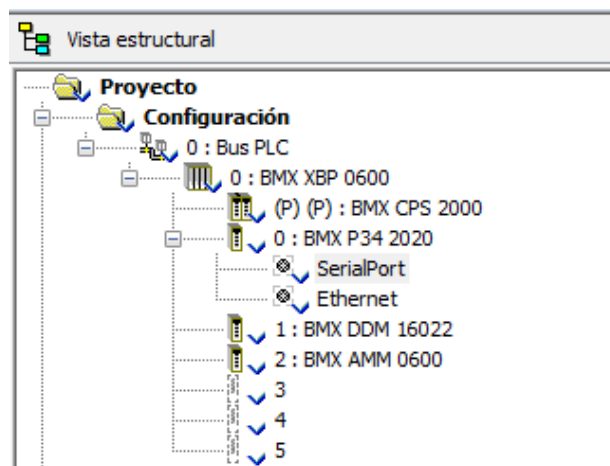


Figura 19. Ruta de configuración de SerialPort y Ethernet

<sup>4</sup> Lenguaje de programación orientado a gráficos que se asemeja a la estructura de un circuito eléctrico

Realizando doble clic sobre el *SerialPort* (Figura 19) se abre una ventana donde se introduce la configuración de comunicación Modbus con el Altivar 71 (Ver figura 20).

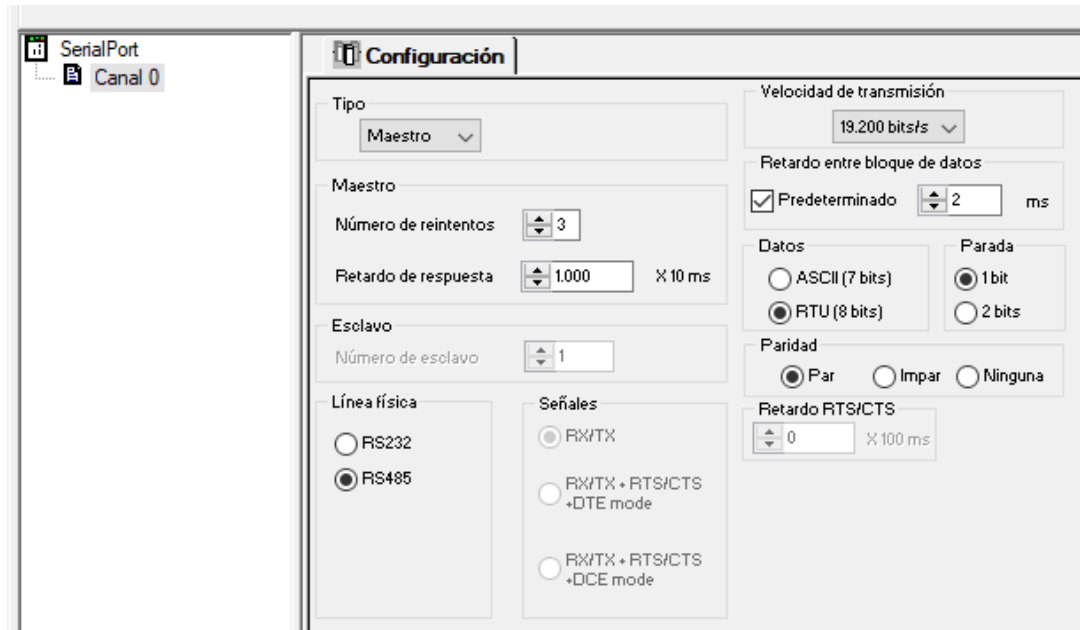


Figura 20. Ventana de configuración serie (Modbus RTU)

El puerto de comunicación serie es configurado como maestro de comunicación Modbus RTU, 8 bits de datos, uno de parada y paridad Par a 19200 bits/s con un retardo máximo de comunicación de 10 segundos.

### 3.4 Configuración de comunicación entre Unity Pro XL y Modicon M340

Para la comunicación del Modicon M340 con la pantalla de operador dentro del UnityPro XL es necesario configurar el puerto Ethernet. La configuración de este protocolo es diferente a la anterior y es necesario crear la configuración en un fichero aparte, pues se entiende que desde una misma licencia de Unity Pro XL es posible configurar varios PLCs con distintas especificaciones de configuración cada uno.

Para realizarlo, se parte de la *Vista estructural*, siguiendo la ruta *Proyecto -> Comunicación -> Redes* al realizar clic derecho, se llega a “*Nueva Red...*” como se muestra en la figura 21.

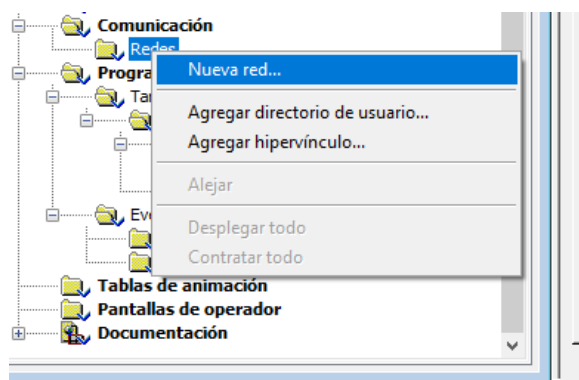


Figura 21. Ruta para añadir una nueva red

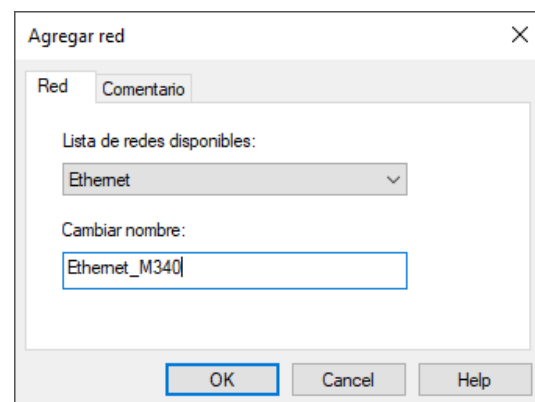


Figura 22. Nombrar y seleccionar tipo de red

En la ventana que aparece a continuación en el desplegable “*Lista de redes disponibles*” se selecciona “*Ethernet*” y la nombramos, en nuestro caso “*Ethernet\_M340*” (Figura 22).

Al presionar “OK” ya aparece la red creada (Figura 23), realizando doble clic se abre la ventana de configuración (Figura 24).

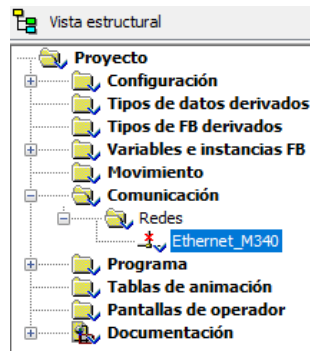


Figura 23. Red EtherNet creada

La configuración Ethernet debe obedecer a las normas y propiedades de la red donde opera. Como se puede observar en la figura 24, la dirección IP asignada no sigue ningún orden, pues forma parte de un entorno aislado PLC-PC

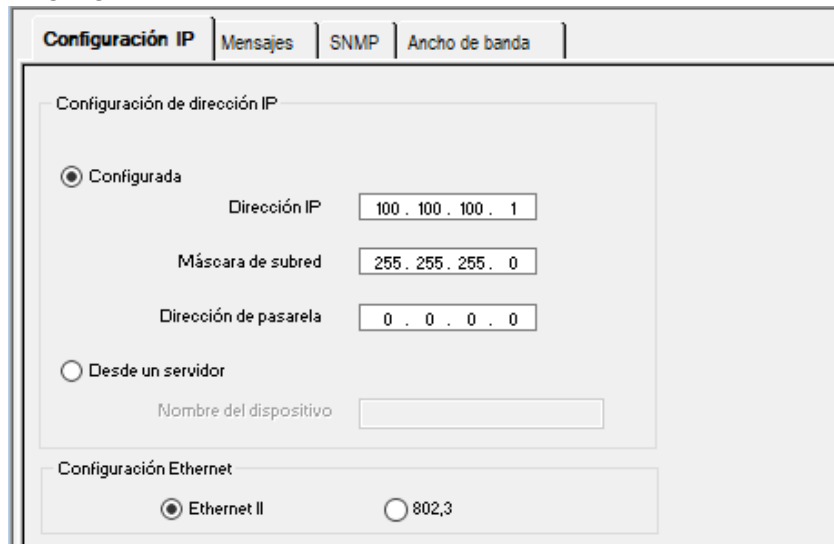


Figura 24. Configuración de la red EtherNet

En la ventana de configuración solo se seleccionó la familia del modelo del procesador, se indicó la IP que tendrá el PLC y la máscara de subred. Para validar los cambios hay que presionar el botón “Validar” en el menú superior (Figura 25).

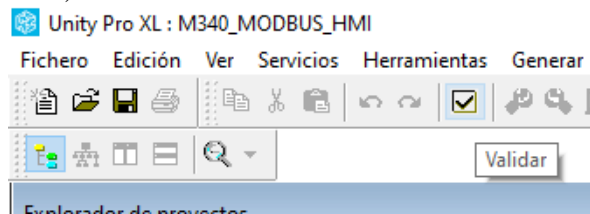


Figura 25. Ubicación del botón Validar

Una vez establecida la configuración se procede a añadirla al módulo de nuestra CPU, en nuestro caso BMX P34 2020. En la vista estructural del explorador de proyectos, nos dirigimos a *Proyecto* -> *Configuración* -> *Bus PLC* -> *BMX XBP 0600* -> *0: BMW P34 2020* -> *Ethernet* (ver figura 26) y se realiza doble clic.

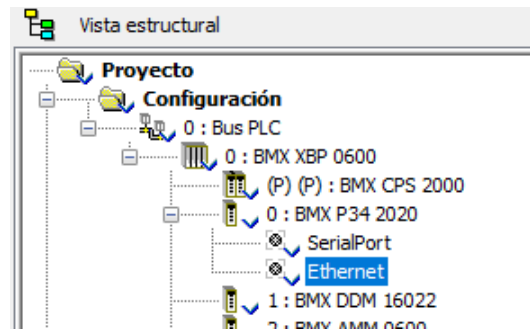


Figura 26. Ruta de Puerto EtherNet del procesador BMX P34 2020

Esta acción abre una ventana donde permite seleccionar el tipo de función del puerto, nuestro procesador solo dispone de TCP-IP y para ello como muestra la figura 27 dentro del campo “Función” se selecciona la opción “ETH TCP IP”.

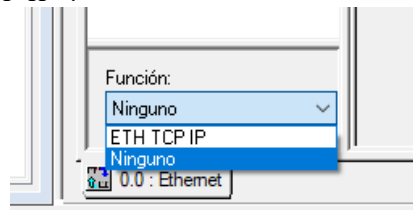


Figura 27. Insertar configuración Ethernet creada

Automáticamente cambian los campos y aparece el desplegable “Conexión de red” (Figura 28), donde seleccionamos la configuración establecida en el apartado anterior: “Ethernet\_M340”. (figura 22).

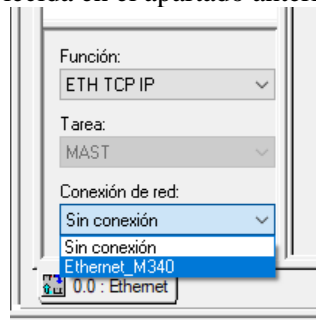


Figura 28. Selección de configuración creada

La configuración se valida clicando en el botón de la figura 25. Fijarse que el símbolo de la configuración de red cambia, donde antes había una cruz roja, aparece un PLC indicando que es la configuración cargada a la CPU (ver Figura 29). Esta notación gráfica tiene relevancia en aquellos casos donde se dispone de varios tipos de configuraciones para acoplar varios M340 dentro de una red (empresarial, privada, entre otros).

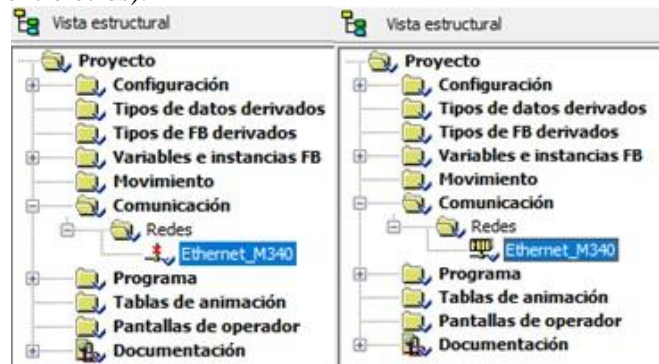


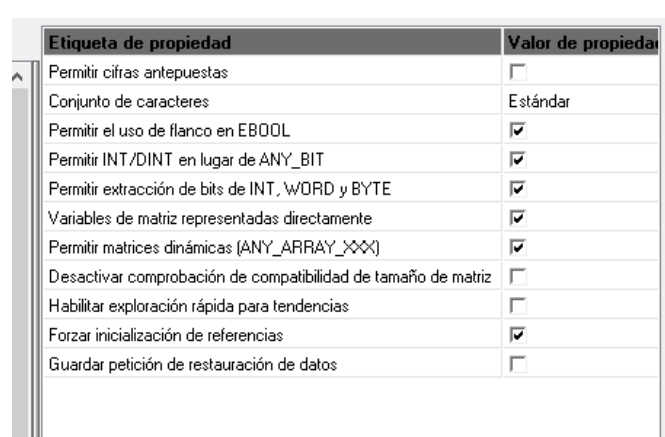
Figura 29. Antes y después de cargar la configuración

Con un programa generado y estas configuraciones validadas, basta con conectarse al PLC y enviar el proyecto para que quede almacenado en su memoria interna.

### 3.5 Ajustes de Programación

Durante la realización del programa se utilizaron técnicas de programación para la cuales hay que configurar previamente el proyecto, de lo contrario entregaría varios errores e impedirían generar una versión lista para descargar.

El software manipula varios tipos de archivos, los más relevantes son: \*.STU, \*.STA y \*.XEF. Los ficheros STU son ficheros que se generan automáticamente al guardar el proyecto con nombre propio y no es necesario tener una versión generada (lista para descargar en el PLC) para que se creen. Sin embargo, para obtener un fichero \*.STA, se necesita tener una versión generada y por tanto sin errores, su mayor ventaja es su pequeño tamaño de almacenamiento lo que lo hace ideal para compartirlo por correo electrónico, entre otras. Por otra parte, tenemos los archivos de intercambio de aplicación Unity Pro \*.XEF muy útiles para exportar proyecto a versiones más antigua del de este software o viceversa [9], con el inconveniente que algunos metadatos no se exportan y hay que volver a configurarlos en el destino. Ejemplo de esto, particularidades del lenguaje de programación y variables que se solucionan en Herramientas -> Ajustes del Proyecto (Figura 30) como se explicará más adelante.



Etiqueta de propiedad	Valor de propiedad
Permitir cifras antepuestas	<input type="checkbox"/>
Conjunto de caracteres	Estándar
Permitir el uso de flanco en EBOOL	<input checked="" type="checkbox"/>
Permitir INT/DINT en lugar de ANY_BIT	<input checked="" type="checkbox"/>
Permitir extracción de bits de INT, WORD y BYTE	<input checked="" type="checkbox"/>
Variables de matriz representadas directamente	<input checked="" type="checkbox"/>
Permitir matrices dinámicas (ANY_ARRAY_XXX)	<input checked="" type="checkbox"/>
Desactivar comprobación de compatibilidad de tamaño de matriz	<input type="checkbox"/>
Habilitar exploración rápida para tendencias	<input type="checkbox"/>
Forzar inicialización de referencias	<input checked="" type="checkbox"/>
Guardar petición de restauración de datos	<input type="checkbox"/>

Figura 30. Pestañas Seleccionadas en Ajustes del Proyecto

En el menú superior *Herramientas -> Ajustes del Proyecto* no llegamos al apartado de *Variables*. Este menú puede variar según las versiones, pero sigue la misma lógica. Habilitar el uso de las variables EBOOL para recoger el estado de transición, habilitar el uso de las de las variables INT/DINT en la manipulación de arreglos, la extracción de bits de palabras (independiente al tipo de datos) y permitir las matrices dinámicas. Estas opciones se observan en la figura 30.

Los ajustes que se acaban de detallar, junto con la asignación de la dirección IP del Modicon M340, son dos ajustes que hay que realizar en el destino cada vez que se exporte un archivo \*.XEF a otra versión de Unity Pro XL.

### 3.6 Configuración de Altivar 71

Para el correcto funcionamiento del sistema, se necesita configurar el variador de frecuencia para que se comunice con el PLC, para reconocer el codificador (“Encoder”) que tiene acoplado y, sobre todo, insertar los valores eléctricos del motor de inducción con el que se cuenta.

En el presente apartado se describen las configuraciones que se realizaron, adaptadas a un sistema de control remoto de un motor de inducción en lazo cerrado. No se variaron parámetros de control manual u otras características del variador de frecuencia que no se describan.

El primer punto que se configurará es el menú *1.4 CONTROL MOTOR* (Figura 31) puesto que es aquí donde se introducen los valores de la placa de características del motor que se mencionaron en el Capítulo anterior (apartado 2.1.3). Es un paso crítico donde una mala configuración puede ocasionar graves errores en el funcionamiento del motor durante el control externo.



Figura 31. Insertar los parámetros del motor en el ATV71



Figura 32. Ejecución de AUTOAJUSTE

Una vez configurados todos los parámetros del motor -nunca antes- se realiza un autoajuste (Figura 32). Este proceso que tarda entre uno a dos segundos mide valores internos del motor, autoselecciona los valores y límites de operación para crear un estado de partida. Una vez concluido, a partir del mismo se realiza el resto de configuraciones.

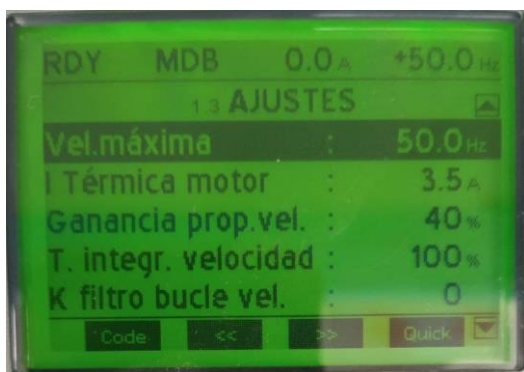


Figura 33. Valor por defecto de I térmica = 3.5 A.



Figura 34. Valor establecido de límite de

Dos de los valores que se configuraron (en el menú 1.3 AJUSTES) fueron Intensidad térmica y el límite de intensidad, valor por el que limitamos la intensidad máxima que se le podría inyectar al motor. A este último parámetro el autoajuste le había asignado 3.5 A (Figura 33), valor que se consideró alto y que pasamos a cambiar por 1,5 In, en este caso 2,8 A. También, como medida de seguridad, conociendo los objetivos didácticos del proyecto, se limitó el variador para que la máxima corriente que inyecte al motor sea de 2 A (Figura 34).

Después de este paso, volvemos al menú 1.4 CONTROL MOTOR para seleccionar el tipo de control y configurar el codificador. El Altivar dispone de varios tipos de control, por defecto establece el SVC por U, un control vectorial por flujo en tensión, en lazo abierto (Figura 36). Dado que en nuestro proyecto disponemos de un codificador, que nos permite hacer un control más preciso con mejores garantías de seguridad, la opción CVF, Control vectorial de flujo en corriente, en lazo cerrado, es la más indicada. Este requiere tener configurado un codificador, por lo que sin tocar nada aún, se realiza la configuración del codificador (encoder).

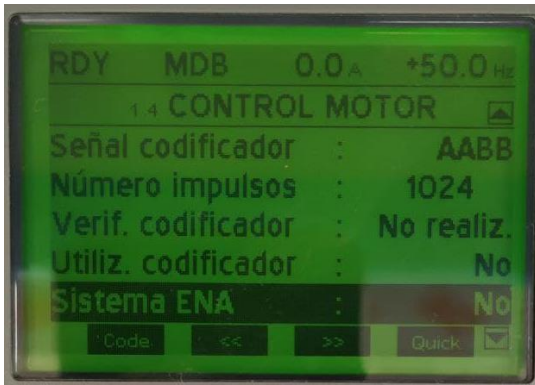


Figura 35. Configuración de codificador en Altivar 71



Figura 36. El control por defecto es SVC por U

Esta configuración consiste básicamente en seleccionar el tipo de señal, dada la capacidad de nuestro codificador de detectar la dirección de giro y la salida diferencial, se selecciona AABB (Figura 35), se introduce el número de pulsos por revolución (1024 ppr) y se pincha la opción *Verif. Codificador*. Luego se puede ir al *Tipo de control motor*, seleccionar *CVF* y arrancar el motor, en caso de errores, basta con revisar el orden de rotación de fases (ABC), que el codificador esté correctamente conectado, así como el orden de las fases entre el motor y el variador sigan la rotación ABC. El orden de estas configuraciones es crítico para mantener la integridad del sistema y ahorrar tiempo de puesta en marcha.



Figura 37. Inicialmente se selecciona el Perfil E/S luego se selecciona “Separado”

El variador de velocidad es configurado por defecto en *Perfil E/S* (figura 37), de ayuda para las configuraciones anteriores, pero no el que más flexibilidad entrega en el control externo. Para ello se selecciona en su lugar el perfil *Separado*. Los dos canales de control se establecen vía Modbus y las conmutaciones de canal también se establecen cuando el canal 1 esté activo (Figura 37). Esta configuración permite que el Altivar siga las referencias e instrucciones que le llegarán vía Modbus.

El siguiente paso es introducir los parámetros de configuración Modbus, los mismos que se establecieron en el autómata. Dentro del menú *1.9 COMUNICACIÓN -> MODBUS RED* (Figura 38) se establece una velocidad de 19 200 baudios, 8 bits de datos, 1 de parada y paridad Par con 10 s de tiempo de espera (Timeout).

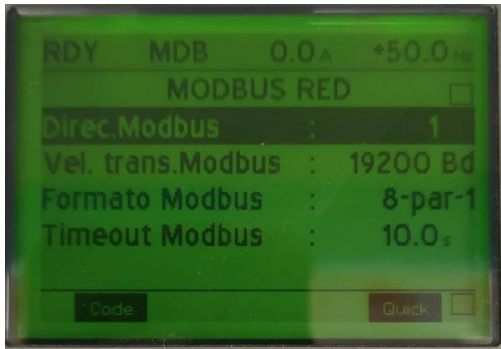


Figura 38. Configuración Modbus del ATV71.



Figura 39. Configurar REARRANQUE AUTOMÁTICO

Es importante activar *REARRANQUE AUTO* en el menú 1.8 *GESTIÓN DE FALLOS* (Figura 39), de esta forma siempre que ocurra un fallo, y este sea reconocido, el Altivar pasa a estado inicial sin tener que rearmar manualmente todo el sistema. Si la falta es grave, o por cuestiones del programa, por ejemplo, una parada de emergencia con bloqueo, es necesaria la intervención física de un operario para el rearme.

El Perfil *Separado* utiliza gran parte de los estados del estándar CiA402 y DRIVECOM<sup>5</sup>. Actualmente este perfil constituye un estándar ampliamente utilizado y disponible en diferentes protocolos de comunicación: Modbus, CANopen, Fipio, ModbusPlus, Uni-Telway, Profibus DP, DeviceNet y INTERBUS [10]. De esta forma, la lógica de programación y de los estados no cambia independientemente del protocolo de comunicación que se emplee. En el capítulo siguiente se explican las peculiaridades del estándar, sus estados y cómo se basa la lógica de control sobre los mismos.

### 3.7 Pantallas de Operador

En las pantallas de Operador se crean tanto los botones para el control del proceso como las interfaces donde se muestran las principales variables de control. Cada botón, símbolo e imagen está enlazada a una variable en específico, a no ser que el elemento tenga un papel meramente decorativo.

Antes de crearla, se debe conocer la lógica que va a seguir la interfaz a diseñar, las variables que necesitan una representación gráfica y el papel que desempeña el conjunto dentro del sistema general. El capítulo siguiente seguirá esta lógica de diseño para mostrar cada una de las pantallas de operador que se diseñaron, explicando la lógica que se requería, las variables que intervienen y, por último, su representación gráfica.

Para crear una Pantalla de Operador, se parte del *Explorador de proyectos* -> *Proyectos* > *Pantallas de Operador* y se hace clic derecho para insertar una nueva Pantalla como se observa en la Figura 40.

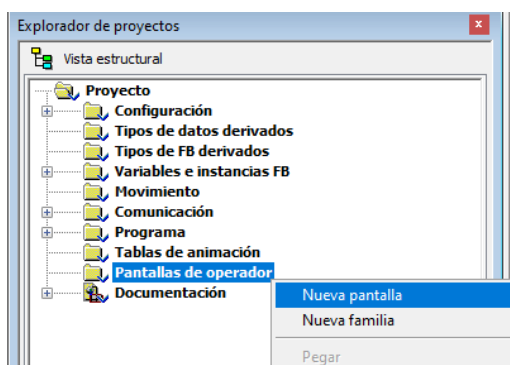


Figura 40. Añadir una nueva Pantalla de Operador

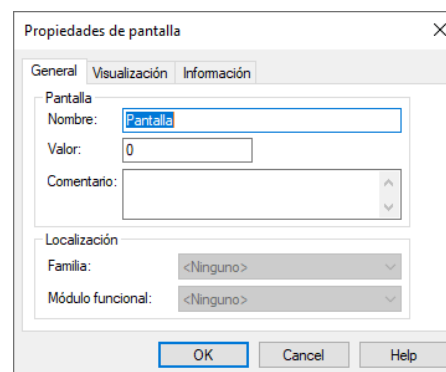


Figura 41. Asignar Valor a Pantalla de Operador

<sup>5</sup> Perfil de control de movimiento concebido para redes InterBus pero actualmente exportado a múltiples protocolos de comunicación.



En la ventana *Propiedades de pantalla* (Figura 41) se introduce el nombre que se le dará a esta Pantalla de Operador y Valor para poder activarla, por ejemplo, el HMI de este Proyecto lo componen cuatro pantallas de operador: *Configuraciones Iniciales* (Valor: 0), *Supervisión* (Valor: 2), *Marcha\_Paro* (Valor 1) y *Guardar\_Configuración* (Valor: 3) (Figura 42).

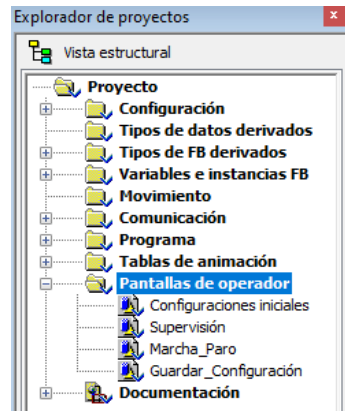


Figura 42. Nombres de las Pantallas del Sistema

Inicialmente se tendrá una gran hoja en blanco que poco a poco se deberá ir llenando con los elementos que ejecutarán las funcionalidades de cada Pantalla.

Dentro del UnityPro XL disponemos de dos opciones para hacer nuestros diseños: los elementos de la barra de herramientas y la *Librería de Pantallas de Operador*. Ambos puedes coexistir en el mismo diseño sin ningún tipo de problemas de compatibilidad, pues básicamente los segundos fueron creados agrupando varios diseños de la barra de herramientas uniendo rectángulos, elipses y curvas.

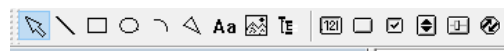


Figura 43. IOSEditos, Opciones de la barra de herramienta para el diseño

La parte de la barra de herramienta (Figura 43) que se comenta, recibe el nombre de *IOSEditor* y con este limitado catálogo de opciones y tiempo de diseño, se creó el cien por ciento de las pantallas de operador, logrando así una total compatibilidad con todas las versiones de Unity Pro XL.

Si en una aplicación puntual se necesita hacer uso de la *Librería de Pantallas de Operador*, basta con ir en la barra superior a *Herramientas -> Librería de Pantallas de Operador* (Figura 44) o simplemente presionar en nuestro teclado ALT + 4. De esta forma tendremos acceso a Imágenes pre-diseñadas, ejemplo de la categoría *Botones* se observan en la Figura 45.

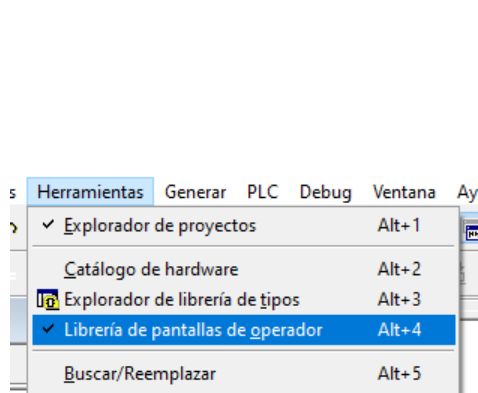


Figura 44. Ubicación de librería de diseños.

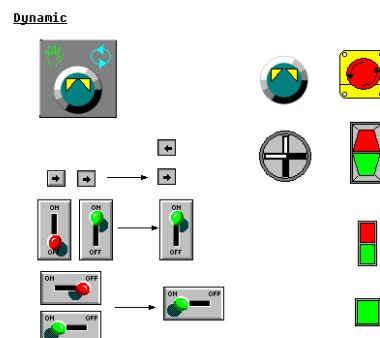


Figura 45. Ejemplo de figura prediseñadas

Los diseños de estas librerías se recomiendan en proyectos donde su uso sea justificado, pues en ambientes donde se trabaje con varias versiones de Unity Pro puede ser un inconveniente. Exportar el

proyecto a otra versión a través de *Archivos de intercambio de aplicación Unity Pro* con la extensión \*.XEF, no incluye las imágenes de esta librería y puede resultar un percance en el funcionamiento del programa

Ahora bien, para demostrar cómo se crearon cada uno de los objetos y como están interconectados con las variables del sistema, se explicará el procedimiento mediante un ejemplo.

Dentro de UnityPro existen varios tipos de variables y funciones que se pueden utilizar dentro de los programas. Bien pueden ser variables de tipos de datos elementales (bool, int, string...), derivados (estructuras, arrays...) o de entrada / salida, vinculadas con los módulos hardware instalados.

Para crear una variable elemental, por ejemplo: una variable de tipo BOOL, con el objetivo de energizar y desconectar el motor, se sigue la ruta dentro del explorador de proyecto, *Proyecto > Variables e instancias FB > Variables elementales* (Figura 46). Inicialmente aparecerá la lista vacía con la primera línea lista para añadir una variable, realizando doble clic se pueden añadir un nombre y escoger el Tipo de dato. En la figura 47 se puede observar la variable después de creada.

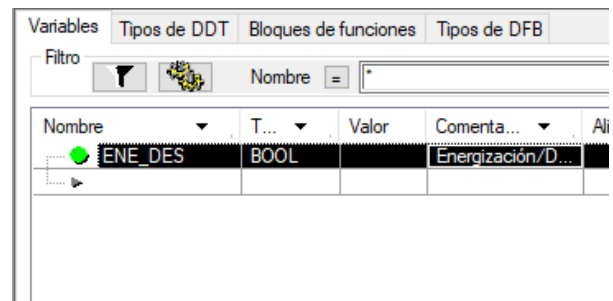
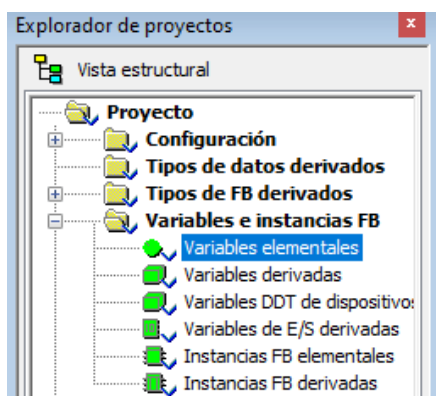


Figura 46. Ruta de tabla de variables elementales

Figura 47. Ejemplo de una variable tipo BOOL creada

Ahora es necesario crear el objeto que controlará esta variable. Para ello, en la pantalla de Operador creada, se inserta un objeto compatible con el tipo de variable seleccionada anteriormente, de tipo BOOL, *Botón de Comando* (Figura 48) es un muy buen ejemplo de ello.

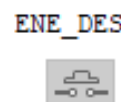
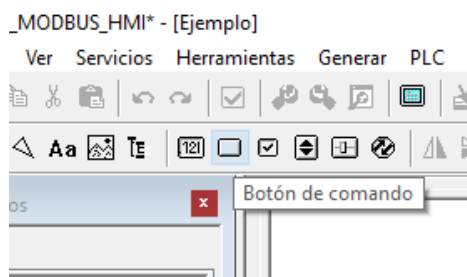


Figura 48. Botón de comando dentro de IOSEditos.

Figura 49. Botón de comando después de creado

Al realizar doble clic sobre el Botón de comandos (Figura 49) se abre la ventana con las propiedades del objeto, en este caso como se va a controlar una variable booleana, se navega hasta la ventana *Control* (Figura 50) y clicamos en los tres puntos a la derecha del campo *Variable*. Esta acción abre la ventana de selección de instancias donde estará la variable que creamos anteriormente y queda como se observa en la Figura 51.

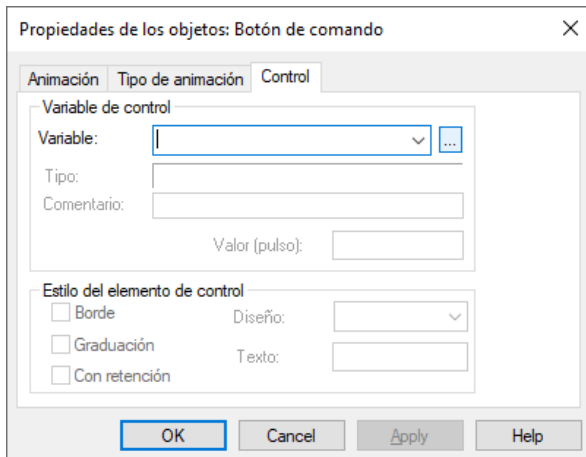


Figura 50. Propiedades de Objeto de Entrada

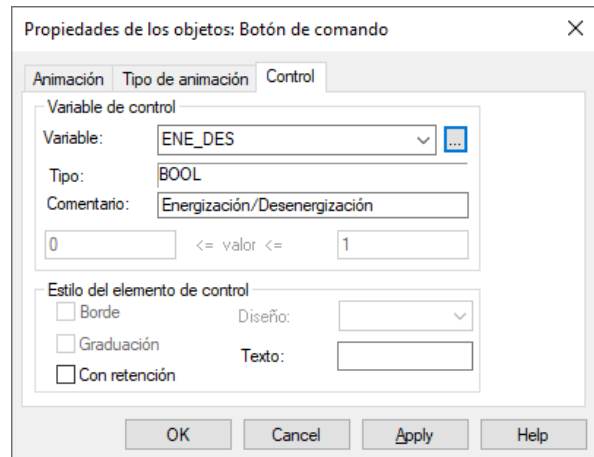


Figura 51. Instancia del Botón de Comando

Cabe destacar que en caso que sea un objeto utilizado para mostrar variables de salida (Ejemplo: Tipo Texto para mostrar advertencias), es necesario ir a la ventana *Animación* que se observa en las figuras anteriores, puesto que funcionará como campo de salida y no de entrada. El procedimiento posterior para instanciar la variable se repite.

Con este ejemplo, se muestra cómo se realiza la instanciación de variables en pantalla de operador de forma general. El uso de cada objeto y variable en el programa dependerá de la lógica a seguir.



# 4 PROGRAMACIÓN Y DESARROLLO DEL HMI

En el capítulo se explicará cómo se desarrolló el HMI en las pantallas de operado de Unity Pro XL. Para ello se utilizaron todas las configuraciones y dispositivos descritos en capítulos anteriores. Inicialmente se describirá el perfil de control del motor de inducción con el estándar CiA402, un perfil ampliamente utilizado para el control de movimiento en la industria.

Una vez expuestos los estados del proceso se explicarán los diagramas de flujos del programa, las variables que intervienen y como serán representadas. Se concluye el capítulo con el diseño de las ventanas de operador.

## 4.1 Estándar de control de movimiento CiA 402

Este estándar de control de motor permite que los variadores de frecuencia hagan uso de un perfil de funcionamiento basados en una máquina de estados [11]. Siendo más específico, dispone de 8 estados de los cuales el 90 % del tiempo se alterna entre 5 de ellos (estados: 2, 3, 4, 5 y 6) y en caso de falta se transita por los estados 7 y 8 (Figura 52).

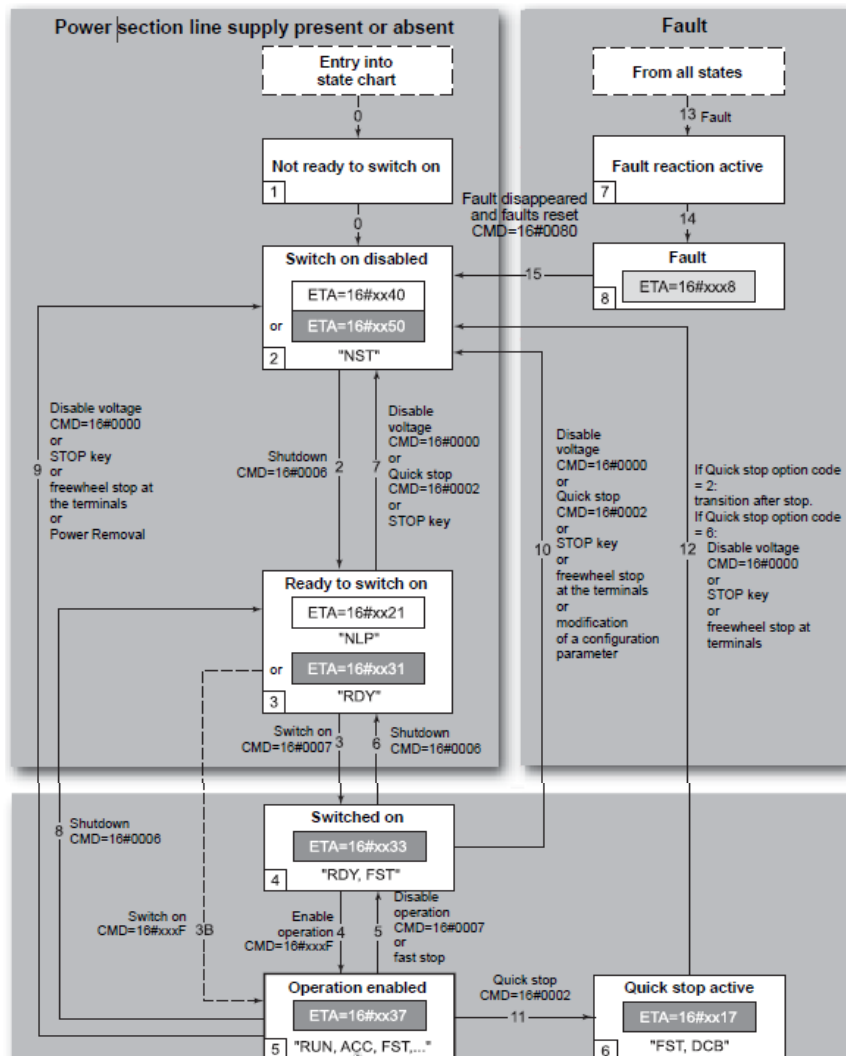


Figura 52. Mapa de estados de estándar CiA402

El Estado 1 es un estado de tránsito, por el cual se transita al conectar el Variador, después de Cargar un nuevo proyecto o por acceso a un fichero en memoria para restablecer una versión y pasa automáticamente al Estado 2. En este estado el motor está desconectado y no tiene alimentación, el

variador se encuentra en “Parada Libre”. Para poder salir de este estado es necesario enviar a través de la palabra de control *CMD* (Dirección Lógica: 8501) el valor 16#0006 que indica la autorización de conexión a la línea, pero sin habilitar las operaciones y por tanto el cambio al *Estado 3*. En el *Estado 3* el sistema está listo para arrancar y existen tanto dos posibilidades para cambiar a un estado superior, como para regresar al *Estado 2*. Las dos opciones con las que se retorna al *Estado 2* son Parada Rápida (*CMD* = 16#0002) y Desconexión de Tensión (*CMD* = 16#0000).

Para poner el motor en Marcha y saltar directamente al *Estado 5*, se necesita enviar a la palabra de control *CMD* el valor 16#000F. En este estado el motor arrancará solo si tiene referencia de frecuencia, o sea, si a la dirección lógica de referencia de frecuencia 8502 se le ha enviado un valor diferente de cero (Ejemplo: 50 Hz). Por otra parte, existe una vía más protocolar que a partir del *Estado 3* no pone en marcha el sistema directamente, sino que alimenta el motor y lo mantiene paralizado hasta el arranque. Al enviarle a la palabra de control *CMD* = 16#0007, estando en el *Estado 3*, se pasa al *Estado 4*. Este paso puede ser fundamental en escenarios donde el motor necesite estar detenido ya que el eje puede estar accionando mecanismos que transporten cargas delicadas (Ejemplo: Pequeños elevadores industriales o cintas transportadoras en pendiente, que necesiten un estado con el motor bloqueado y no con el eje libre). Para poner el motor en marcha desde el *Estado 4*, si ya se envió la referencia de frecuencia a la dirección lógica 8502, basta con enviarle a la palabra de control *CMD* = 16#000F.

En cualquiera de estos estados, la vía rápida para retornar al *Estado 2* y desconectar el motor en caso de emergencia o por necesidades del programa, es enviar a la palabra de control el valor 16#0000 (*CMD* = 16#0000). En el *Estado 5*, es imposible hacer cambios de variables eléctricas dentro del Altivar, así como realizar una orden de acceso a memoria desde el HMI puesto que el motor está en marcha. Por otra parte, si se ejecuta una parada rápida desde este estado (enviar a la palabra de control *CMD* = 16#0002) el Variador pasa al *Estado 6* donde en dependencia del parámetro QSTD (Código de opción de parada rápida, dirección 8651) permanece en él o transita al *Estado 2*. El parámetro QSTD se preconfigura al inicio del programa o cuando estime conveniente el operador, si se le asigna el valor 02h (enviando a QSTD = 16#0002) solo se utiliza el *Estado 6* como estado de tránsito y llega al *Estado 2* casi al instante. De enviarle 16#0006 a la palabra de control, el variador permanece en este estado hasta que el operador presione el botón STOP en la pantalla del variador o se le envíe desde el HMI el parámetro de desconexión (*CMD* = 16#0000). En cualquiera de estos dos escenarios el paso final es el *Estado 2*.

Por otra parte, el *Estado 7* es un estado de transición por el que recorre el variador en caso de error o falta, independiente de su tipo, para terminar en el *Estado 8*. En dicho estado, el sistema está en falta, el motor detenido y bloqueado. Para salir del *Estado 8*, una vez que se haya solucionado la configuración errónea, revisando la conexiones y los equipos (siguiendo los códigos de error en pantalla de Altivar y de HMI) se envía a la palabra de control *CMD* = 16#0080, para formalizar el reconocimiento del error. De no ser una falta grave (desconexión Modbus, por ejemplo) y estar activada la opción de autoarranque en el Altivar, explicado en el capítulo anterior, se llega al estado inicial (*Estado 2*) para recibir nuevas instrucciones.

#### 4.2 Pantallas de Operador

La interfaz software HMI diseñada la componen cuatro pantallas de operador. La puesta en punta del flujo de programa está concebida para que comience por la primera pantalla, *Configuraciones iniciales*; se realice el arranque y parada a través del *Marcha-Paro*; la supervisión de los principales parámetros y de las variables de estado se monitoriza dentro de *Supervisión* y para guardar la configuración actual, acudir a *Guardar\_Configuración*.

Cada una de las pantallas elaboradas abarca un grupo de funcionalidades estándar para el control de motores de inducción a través de variadores de velocidad. Gran parte de los controles tienen como objetivo exprimir al máximo el perfil CiA402 explicado anteriormente y las capacidades del ATV71. Por tal motivo se decidió incorporar al final de cada pantalla de operador el estado actual del perfil CiA402. De esta forma los estudiantes pueden ver en tiempo real cómo ciertas instrucciones desencadenan cambios de estados.

### 4.2.1 Pantalla 0: Configuraciones Iniciales

Esta pantalla de configuraciones iniciales, entre otras funciones, se encarga de mostrar en cada ciclo de programa los valores de las palabras de estado ETA y ETI. Gran parte del programa decodifica los bits de ambas palabras para conocer el comportamiento puntual del programa, por ejemplo: el sentido de giro del motor o la presencia de fallos en el variador de velocidad. Otra de las funcionalidades que posee esta pantalla es la capacidad de configurar el arranque del motor siguiendo las dos hojas de ruta que nos permite el Estándar CiA402: configuración hasta el *Estado 3* o el camino más protocolar hasta el *Estado 4*. Esta última alternativa es fundamental cuando se manipulen cargas frágiles o se necesite un estado inicial con el motor bloqueado, ambas configuraciones son explicados a detalle en el Capítulo 5.

Como se mencionaba en el apartado 4.1 el perfil CiA402 tiene la posibilidad de configurar varias acciones una vez que se ejecute la parada rápida. Por una parte, podemos seleccionar que con la ejecución de la parada rápida sea necesario el rearme manual del sistema con posible intervención de operador, o bien configurarlo para que transite fluidamente hasta el *Estado 2* y seguir con el control remoto.

Dentro de esta pantalla de puesta en punta, se integró también la posibilidad de insertar una referencia de frecuencia inicial, de esta manera al pasar a la Pantalla *Marcha-Paro* es más fluido el arranque del motor. También se trató por seguridad que no se pudieran insertar valores desequilibrados de frecuencia, aceptando enviar solo aquellos mayores que 44 Hz y menores o iguales que 55 Hz.

La Figura 53 muestra la lógica de programación con las funcionalidades aquí descrita y propias del perfil utilizado explicado en el apartado 4.1. Esta lógica es almacenada una sección de programa con el mismo nombre que esta pantalla de operador.

Cada una de las variables que se presentan en el diagrama flujo llevan el mismo nombre que se le otorgó en el diseño final (Figura 54), para facilitar su comprensión y función dentro de la Tabla 5. Esta tabla, que sigue la misma lógica que las otras tres (Tabla 5, Tabla 6, Tabla 7), pretende mostrar la relación entre cada objeto con el que diseñan las pantallas de operador con su nombre, el tipo de dato de la variable que manipula, con la dirección y nombre del registro Modbus con el que se relaciona.

El diseño final de la Pantalla 0: *Configuraciones iniciales* agrupa cada uno de los elementos de la tabla número 5 para complementar su papel dentro de la lógica de programa que le corresponde a esta pantalla y que se muestra en el diagrama de flujo de la Figura 53. Dentro del diseño, además de cada uno de los elementos aquí descritos, se añade un menú superior para la navegación entre las diferentes pantallas de operador y en la parte inferior un mapa de los estados del estándar CiA402 (Figura 54). Estos dos elementos son comunes en cada una de las cuatro pantallas de operador creadas.

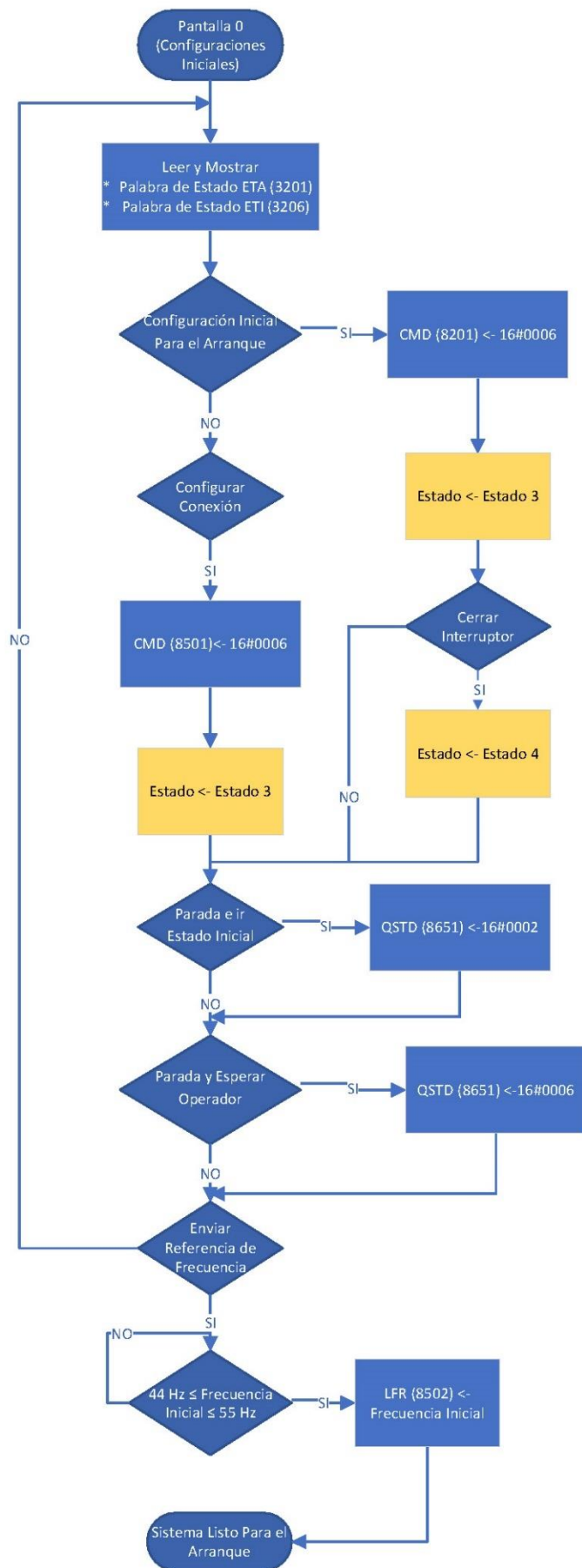
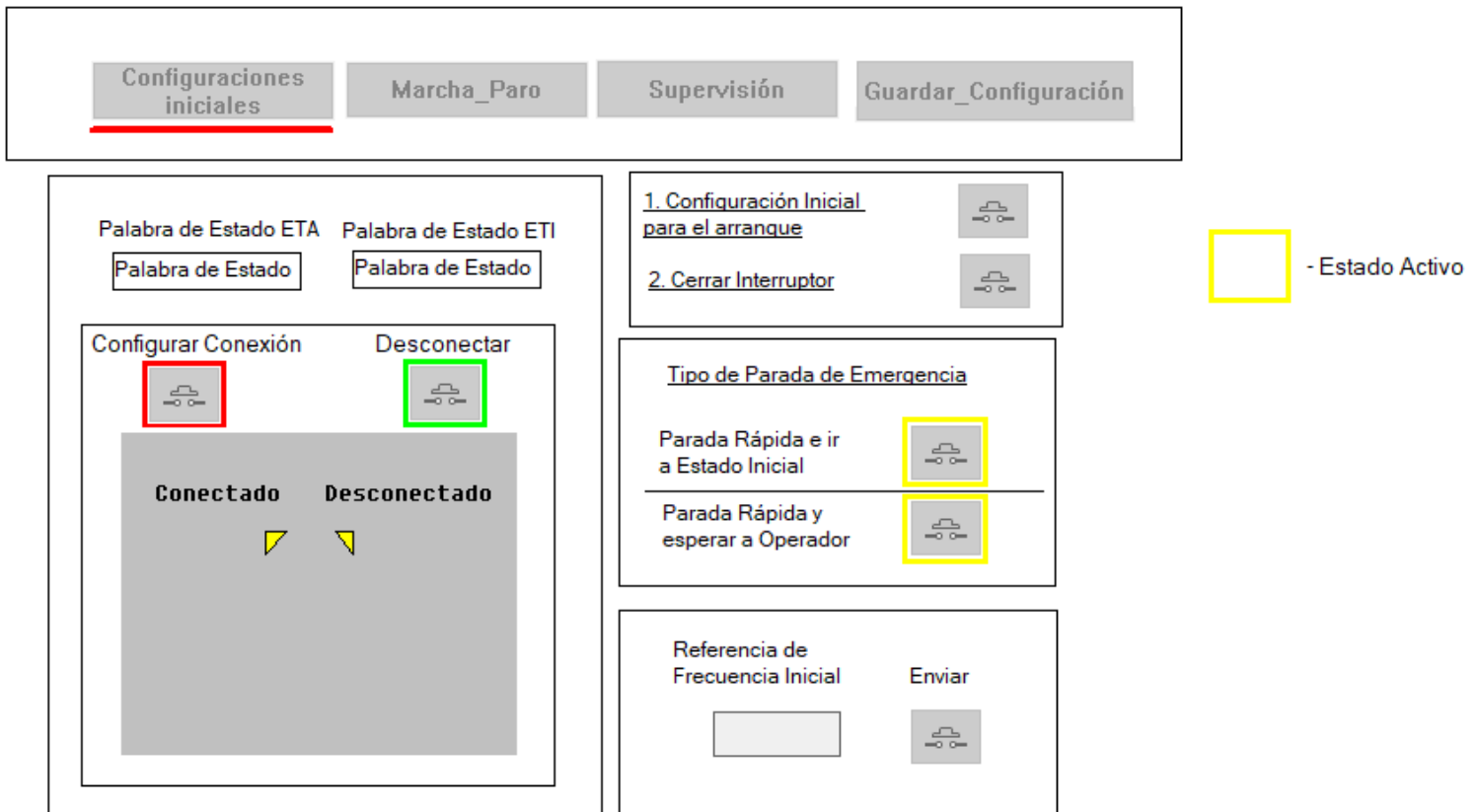


Figura 53. Lógica de programación de la Pantalla de Operador: Configuraciones Iniciales.



Tabla 5. Elementos que componen la Pantalla de Operador #0: Configuraciones Iniciales.

Tipo de Objeto en Unity Pro	Nombre de Objeto	Tipo de Dato	Acción MODBUS o Programa	Dirección Lógica, Código y tipo (R-Lectura, W-Escritura)	Comentario
Texto	Palabra de Estado ETA	Entero (INT)	Lectura en registro de 16 bits	3201 (ETA) (R/W)	Lectura de la palabra de estado ETA, palabra de estado principal que permite conocer si el Altivar está en funcionamiento o esperando referencia, en estado de alarma, si existiese falta (tanto del motor como del VF) y dirección de rotación.
Texto	Palabra de Estado ETI	Entero (INT)	Lectura en registro de 16 bits	3206 (ETI) (R)	Lectura de la palabra de estado ETI, palabra de estado secundaria, nos muestra posibles problemas con las EEPROM, el protocolo por el que se está realizando la comunicación (MODBUS o CANopen) y las transiciones de aceleración, desaceleración, frenado.
Imagen	Desconectar	Booleano (BOOL)	Escritura de 16 bit	8501 (CMD) (R/W)	Botón para entrar en el <i>Estado 2</i> del mapa de estado del estándar CiA402. Se parte de este estado cuando se energiza el variador y son reconocidas las faltas (Si existiesen). Ejecutarlo durante el funcionamiento en cualquier otro estado equivalente a parada libre en los terminales.
Botón de Comando	Configurar Conexión	Booleano (BOOL)	Escribir en un registro de 16 bits el comando 06h	8501 (CMD) (R/W)	Se envía un 06h a la palabra de control Cmd, Partiendo del <i>Estado 2</i> significa la conexión del Motor, pero aún detenido.
Botón de Comando	Configuración Inicial Para el Arranque	Booleano (BOOL)	Escribir en un registro de 16 bits el comando 06h	8501 (CMD) (R/W)	Ejerce la misma acción que el botón anterior, pero aquí se utilizar como paso intermedio para llegar al <i>Estado 4</i> . Constituye el paso inicial de la secuencia que se sigue cuando se manipulan cargas delicadas.
Botón de Comando	Cerrar Interrupto	Booleano (BOOL)	Escritura en un registro de 16 bits el valor 07h	8501 (CMD) (R/W)	Es el segundo paso habilitado para seguir la secuencia de arranque con cargas delicadas. Nos permite ir del <i>Estado 3</i> al <i>Estado 4</i> y habilitar la operación de arranque.
Campo de Entrada	Referencia de Frecuencia Inicial	Entero (INT)	Guardar valor en Memoria	-	El valor de referencia inicial a enviar se recoge por el campo de entrada y se exige que se encuentre entre 44 Hz y 50 Hz por seguridad.
Botón de Comando	Envío de referencia	Booleano (BOOL)	Escritura en registro de 16 bits el Valor de Referencia Inicial	8502 (LFR) (R/W)	Se envía la referencia de frecuencia con la que inicialmente arrancará el VF.
Botón de Comando	Parada Rápida e ir al estado Inicial	Booleano (BOOL)	Escritura en un registro de 16 bits el valor 02h	8651 (QSTD) (R/W)	Al enviar al registro QSTD el valor 02h, se indica que cuando se ejecute la parada rápida el Variador transitará por el <i>Estado 6</i> y pasará automáticamente al <i>Estado 2</i>
Botón de Comando	Parada Rápida y esperar a operador	Booleano (BOOL)	Escritura en un registro de 16 bits el valor 06h	8651 (QSTD) (R/W)	Se envía a QSTD el valor 06h, cuando se ejecute la parada rápida el Variador permanecerá en el <i>Estado 6</i> hasta que intervenga el operador o se desconecte el motor (CMD = 16#0000).
Examinar Pantalla	Menú de selección de Pantallas	-	-	-	Menú para alternar entre las diferentes pantallas del operador.



Mapa de Estado según Estándar CiA402 (DriveCOM)

- Estado Activo

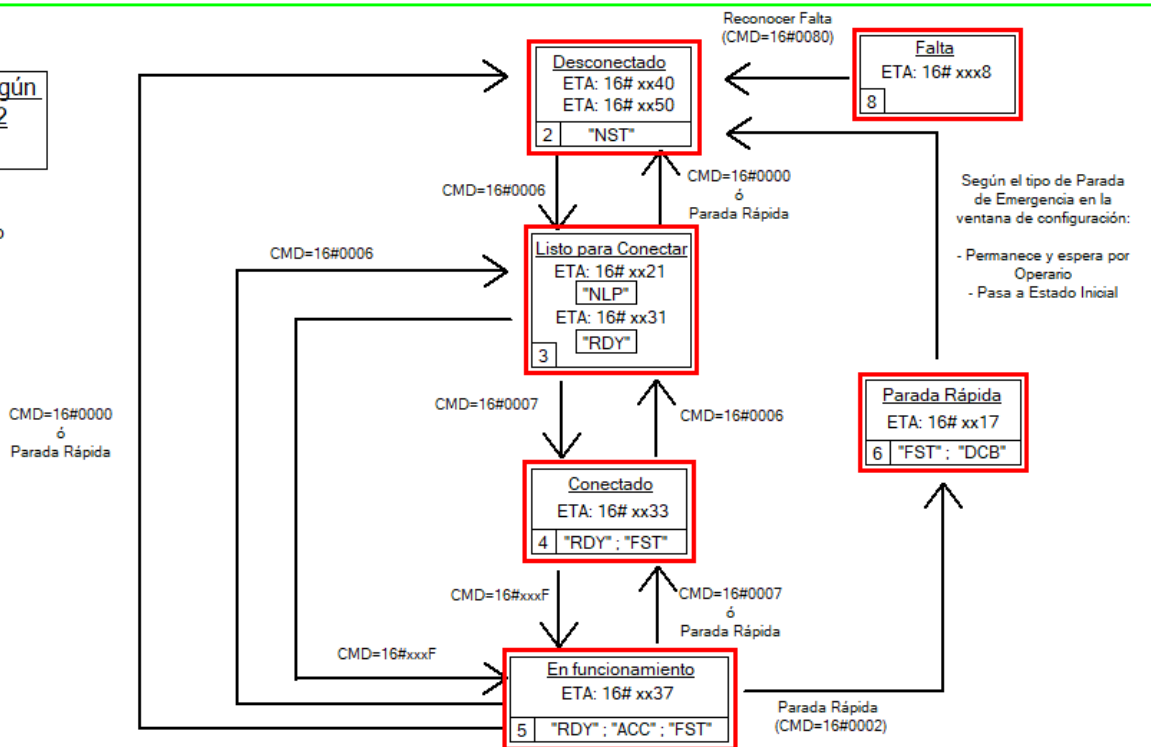


Figura 54 Diseño de la Pantalla de Operador: Configuraciones iniciales.

#### 4.2.2 Pantalla 1: Marcha\_Paro

Durante el desarrollo del proyecto se decidió agrupar en una misma pantalla de operador las funcionalidades para encender, detener y cambiar el sentido de giro del motor (Figura 59). Así como añadir la capacidad de ejecutar órdenes manuales, enviando a una dirección Modbus concreta un valor determinado. Si se realizaron correctamente los procedimientos en la pantalla *Configuraciones iniciales*, el variador debería encontrarse en cualquiera de los dos estados que admite una instrucción de arranque (CMD=16#0006), el *Estado 3* y *Estado 4*. Esta lógica se muestra en la Figura 57 donde el motor después de una orden de marcha comenzará a girar según como se haya ordenado a través de la palabra de control secundaria CMI. El sentido de giro se puede establecer con el motor detenido o en movimiento.

Para los procedimientos de parada rápida hay dos opciones muy diferentes: *Emergencia* y *Paro Configurado en ATV71*. El primero, tal y como se destaca en el diagrama de flujo, hace uso de la configuración de parada que se hizo referencia durante el desarrollo de la pantalla *Configuraciones iniciales*. En dependencia del valor asignado a QSTD (código de opción de parada rápida) el sistema transitará hasta el estado inicial (QSTD=16#0002) o permanecerá en *Estado 6* (QSTD=16#0006) hasta que intervenga un operador o bien se desconecte el motor en la *Pantalla 0*. Por otra parte, *Paro configurado en ATV71* hace uso de una opción propia del variador de frecuencia ATV71. En el MENÚ VARIADOR > 1.7 FUNCIONES APLICACIÓN > CONFIGURACIÓN PARADA (Figura 55), se seleccionó: RAMPA. De esta forma cuando se ejecute el paro configurado el motor experimentará una parada en rampa.

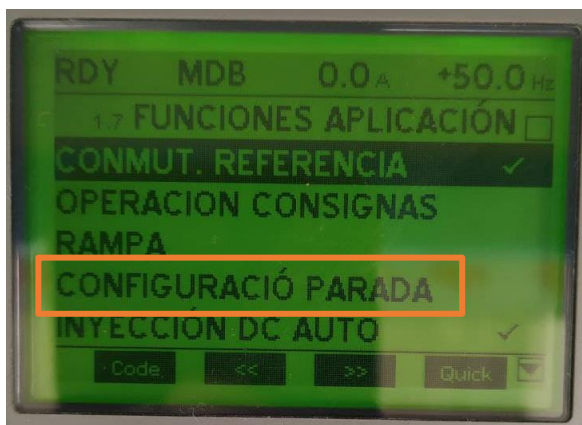


Figura 55. Selección de parada configurada en el ATV71.

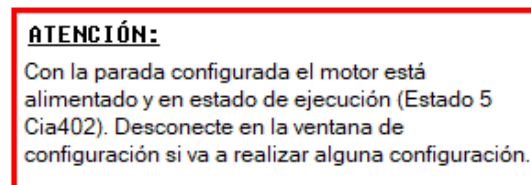


Figura 56. Advertencia con parada configurada

Una peculiaridad de este tipo de parada y para la cual se ha preparado un cartel de advertencia (Figura 56) que aparece cuando se selecciona esta configuración, es que el variador permanece en *Estado 5*, o sea, el motor está alimentado, pero bloqueado. La advertencia pretende informar que durante este estado muchas de las configuraciones habituales no son permitidas y algunas, como acceso a memoria, pueden desencadenar fallos. Esta opción se incorporó con fines didácticos y para futuros proyectos, puesto que está concebida para motores con tipos de servicio S2 y S3, dado que en nuestro caso el motor está concebido para servicios S1.

El cuadro de “Control Manual” es independiente, se concibió para enviar comandos Modbus de interés y que no se hayan recogido en la lógica de programación distribuida en las cuatro pantallas de operador. Esta opción es fundamental en caso que se incurra en faltas graves y se necesite realizar un rescate del motor a distancia en aplicaciones reales e implementadas en la industria. Únicamente envía a la dirección introducida en el campo “Dirección”, el valor insertado en formato hexadecimal dentro del campo “Valor”.

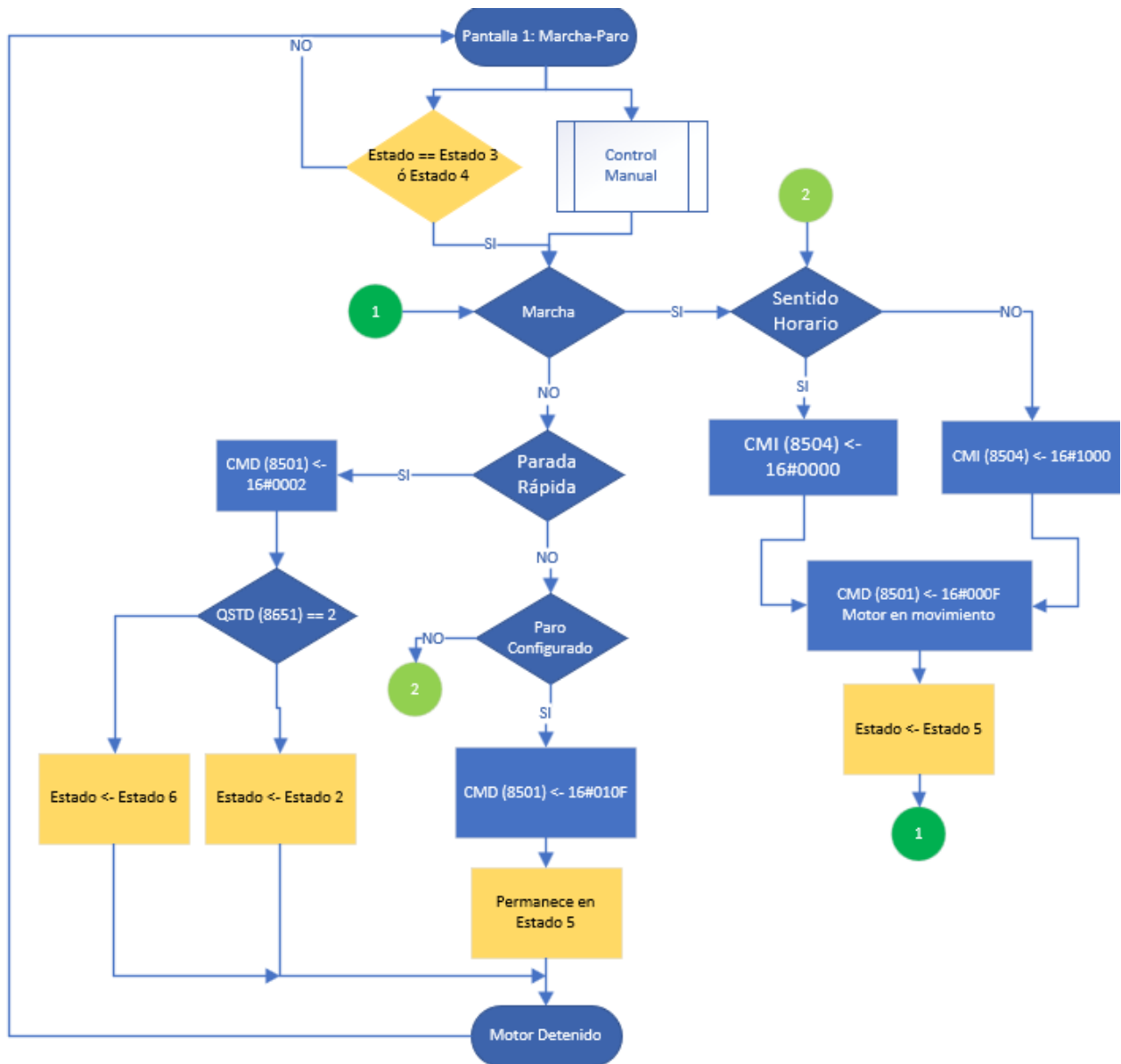


Figura 57. Lógica de programa de la Pantalla de operador: Marcha-Paro

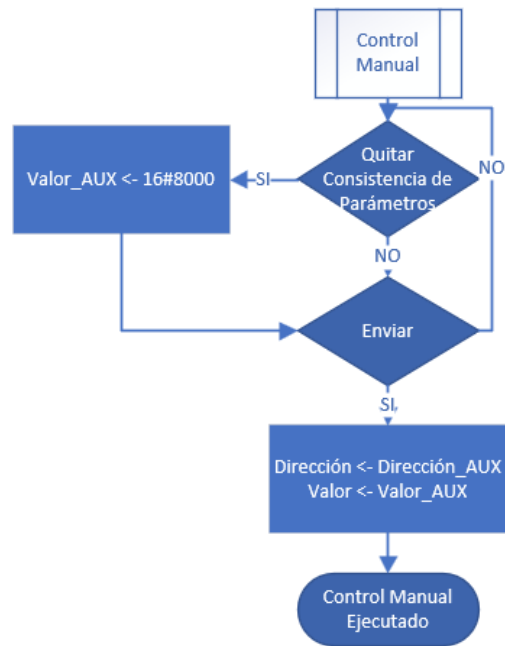


Figura 58. Diagrama de Flujo de programa de la función Control Manual.

Tabla 6. Elementos que componen la Pantalla de operador #1: Marcha-Paro

Tipo de Objeto en Unity Pro	Nombre de Objeto	Tipo de Dato	Acción MODBUS o Programa	Dirección Lógica, Código y tipo (R-Lectura, W-Escritura)	Comentario
Botón de Comando	Marcha	Booleano (BOOL)	Escritura de 0Fh en un registro de 16bits	8501 (CMD) (R/W)	Con la ejecución de este botón se le ordena al ATV71 a través de la palabra de control CMD que el motor pase a estado Marcha (RUN).
Botón de Comando	Paro Emergencia	Booleano (BOOL)	Escritura de 02h en un registro de 16bits	8501 (CMD) (R/W)	El Motor experimenta una parada Rápida y en dependencia del Registro QSTD de las Configuraciones Iniciales Permanece en <i>Estado 6</i> o transita por este para llegar al <i>Estado 2</i> .
Botón de Comando	Paro Configurado en ATV71	Booleano (BOOL)	Escritura de 10Fh en un registro de 16bits	8501 (CMD) (R/W)	El Motor se detiene según el tipo de parada seleccionada ATV71 y también, en dependencia del Registro QSTD, permanece en <i>Estado 6</i> o transita por este para llegar al <i>Estado 2</i> .
Botón de Comando	Sentido Horario	Booleano (BOOL)	Escribe 0000h en el registro de 16 bits	8504 (CMI) (R/W)	Por seguridad se utiliza la palabra de control secundaria CMI, escribir 16#0000 garantiza el bit 12 en '0' e indicar el sentido horario.
Botón de Comando	Sentido Anti_Horario	Booleano (BOOL)	Escribe 1000h en registro de 16 bits	8504 (CMI) (R/W)	Al escribir 16#1000 en la palabra de control CMI, garantiza poner el bit 12 en uno ('1') e indicar el sentido anti_horario.
Imagen	Indicador de Sentido Giro	Entero (INT)	Bucle personalizado de animación	-	A través del bit 15 de la Palabra de Estado Principal ETA y con un contador se realizaron animaciones para Mostrar el sentido de Giro del Motor en una Imagen.
Rectángulo	Indicador de Marcha	Booleano (BOOL)	Máscara al bits 2 de la Palabra de Estado ETA	-	El uso del bit 2 de la palabra de estado ETA permite conocer de manera íntegra si está en funcionamiento el motor. Si el valor es 0 se mostrará el rectángulo que abarca el botón Paro, si es si valor es 1, se muestra el rectángulo del motor Marcha.
Botón de Comando	Enviar Control Manual	Booleano (BOOL)	Escritura en Registro de 16 bits	AUX (AUX) (R/W)	A través de este control el usuario tiene la posibilidad de enviar a una dirección MODBUS concreta el valor que estime necesario. De esta forma se puede realizar configuraciones a distancia que no hayan quedado plasmadas en el HMI.
Botón de Comando	Quitar Consistencia Parámetros	Booleano (BOOL)	Escritura en Registro de 16 bits el valor 16#8000	8504 (CMI) (R/W)	Con esta escritura en la palabra de control secundaria CMI se elimina el chequeo de parámetros, esta acción puede ser de utilidad cuando se necesite hacer escrituras en las memorias del Variador.

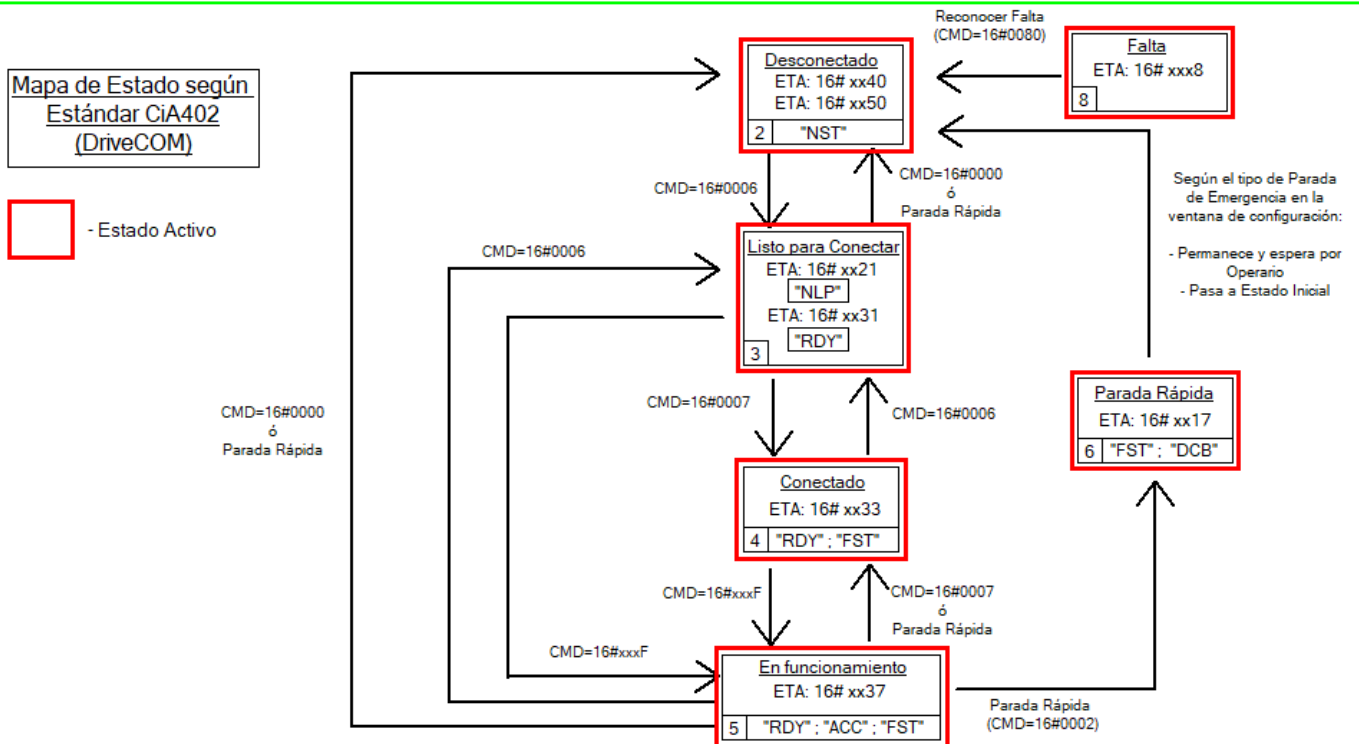
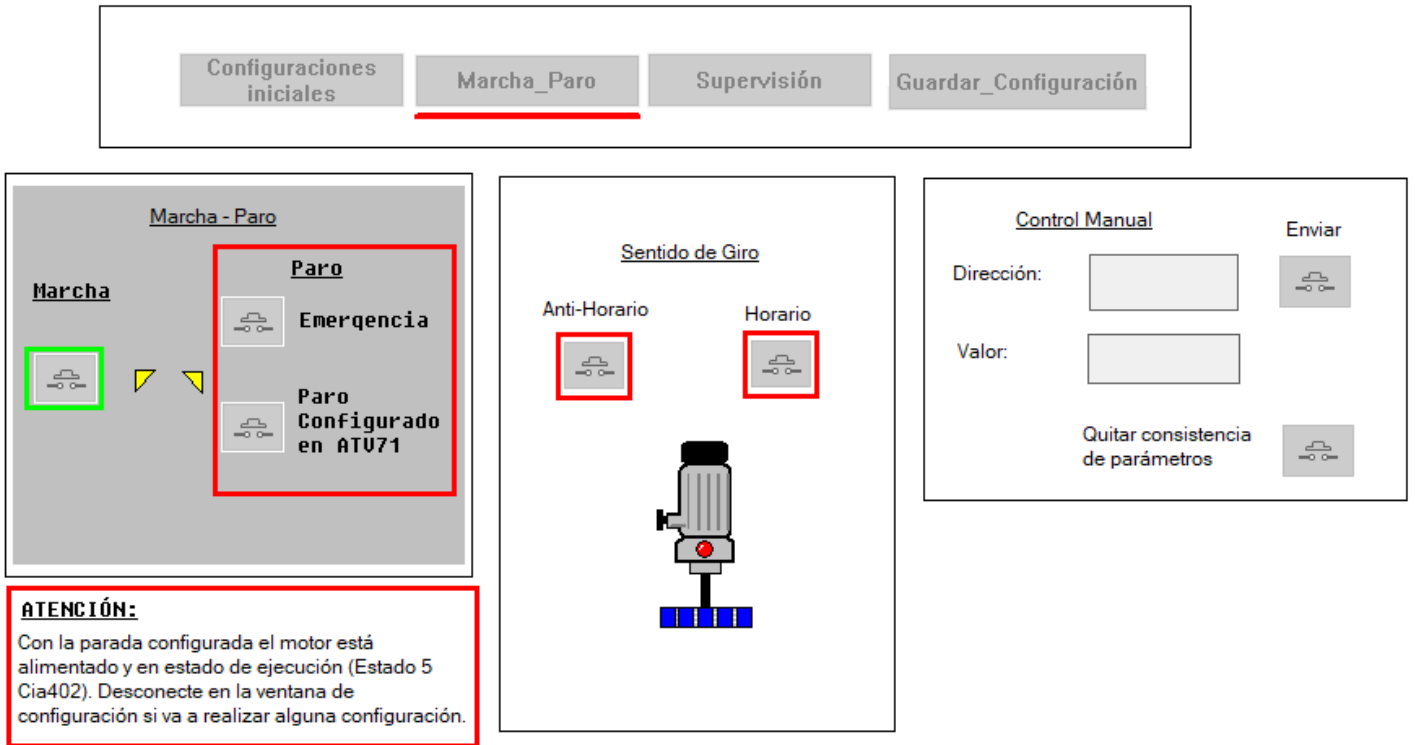


Figura 59. Diseño de la pantalla de operador. Marcha-Paro

### 4.2.3 Pantalla 2: Supervisión

Dentro de la pantalla de operador *Supervisión*, es donde se monitorizan en tiempo real cada uno de los principales parámetros del motor: Tensión, Intensidad, Par, Frecuencia y Velocidad. Paralelo a este proceso el programa chequea los códigos de fallo, tanto los propios del CiA402 (dirección Modbus 8606) como los del Altivar (dirección 7121) y en caso de error los muestra, incluso durante una falta grave, pues permanece anclado a memoria.

En caso de falta, una vez que se reconozca el agravante que conllevó a la ejecución de la misma, es necesario que un operario “in situ” o por el control manual, ejecute el comando de reconocimiento de falta. Con esta acción el sistema retorna al estado inicial, el Estado 2 y consiste en enviar un ‘1’ al bit 7 de la palabra de control CMD, en hexadecimal corresponde al valor 16#0080 que se observa el diagrama de flujo de esta sección del programa (Figura 61).

Para obtener el valor de los principales parámetros, se leen los siguientes registros: LCr (3204) Intensidad actual, UOP (3208) Tensión del Motor, Otrn (3216) Par de salida y SPdl (8641) Velocidad actual. Posteriormente se muestra su valor en un campo de texto y en un gráfico de tendencias simultáneamente.

En esta pantalla de operador es donde se concibe el cambio de referencia, después de que en *Configuraciones iniciales* se solicitara un valor inicial para el arranque. El valor de frecuencia se muestra en un campo de texto y se habilitan dos herramientas para introducir un nuevo valor de referencia, una de navegación rápida: “Indicador de escala” y otra para pequeñas variaciones: “Cuadro de giro”. En ambos casos, una vez que se llegue al valor requerido es necesario pulsar el botón “Enviar” para que se efectúe el cambio de referencia.

La lógica de programación expuesta se muestra en un diagrama de flujo presentado en la Figura 61. Cada uno de los objetos utilizados y su funcionamiento se describen en la Tabla 7, donde se pueden localizar cada una de las características de los objetos representados en la pantalla de operador (Figura 62). En esta pantalla de operador también se muestra el menú de navegación superior y el mapa de estado CiA402, comunes para las 4 pantallas que componen el HMI.

Al observar el diseño final en la Figura 62, cargado de objetos, contrasta con una lógica de programación sencilla y fluida presentada en la Figura 61. Gran parte del tiempo de diseño implicó crear los gráficos de tendencias y medir e insertar la escala de los mismos. En el apartado de programación destaca la creación de dos funciones personalizadas para decodificar los códigos de error y mostrar la descripción brindada en el manual.

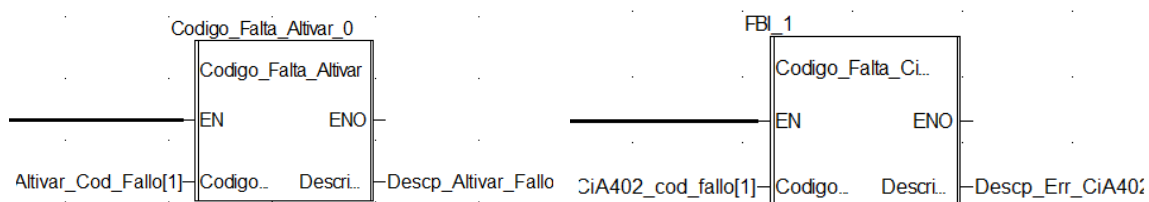


Figura 60. Bloques de Funciones Derivadas creadas para mostrar en formato texto los mensajes de error del manual.

Su funcionamiento es sencillo, reciben el código de error en formato entero (INT) y luego son decodificados a su descripción equivalente en formato STRING (cadena de caracteres) para disminuir considerablemente el tiempo de detección de error por parte del operador.



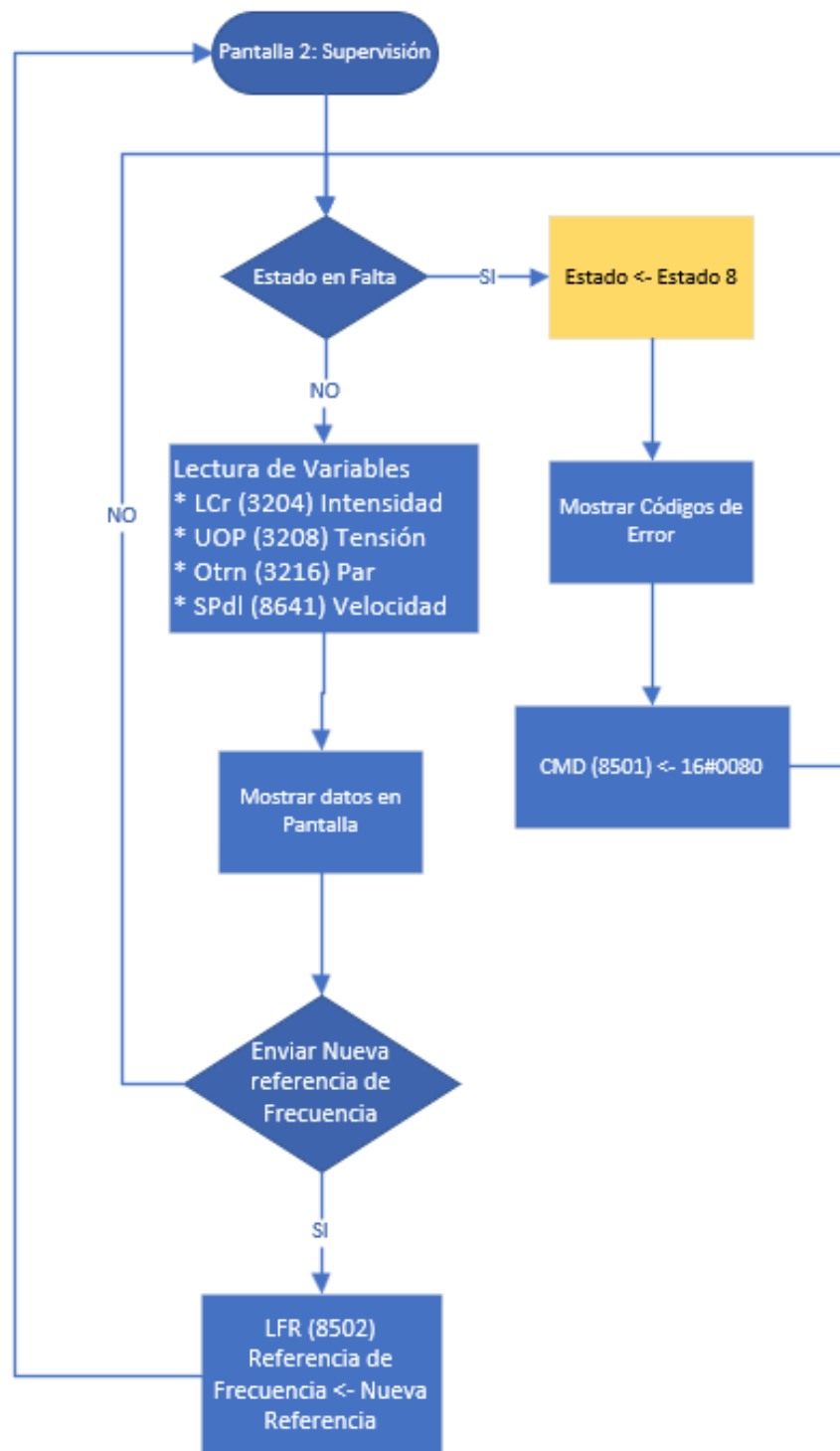


Figura 61. Lógica de programa de la Pantalla de operador: Supervisión.

Tabla 7. Elementos que componen la Pantalla de Operador #2: Supervisión

Tipo de Objeto en Unity Pro	Nombre de Objeto	Tipo de Dato	Acción MODBUS o Programa	Dirección Lógica, Código y tipo (R-Lectura, W-Escritura)	Comentario
Texto	Código de Error	Entero (INT)	Lectura de 16 bits	8606 (Errd) (R)	El código de error de estándar CiA402, permite conocer el último error del sistema, incluso ante un fallo de CPU. Este registro es muy útil para los arranques pos-fallo donde no se conocen las causas del mismo.
Texto	Estado Altivar 71	Entero (INT)	Lectura de 16 bits	7121 (LFt) (R)	Código de falta del Altivar, permite localizar de manera eficaz posibles errores de referencia, aperturas no controladas de lazo de control, sobreintensidades y fallos electrónicos del variador de frecuencia.
Botón de Comando	Reconocimiento de la Falta	Booleano (BOOL)	Escritura registro de 16bits	8601 (CMd) (R/W)	Se genera un pulso de subida en el bit 7 (Escribir 80h) de la palabra de control indicando el reconocimiento de las mismas y pasar al estado 2 de desconexión de red y esperar las consignas de inicialización.
Indicador de Escala	Cambio frecuencia de referencia	Entero (INT)	Escritura de 16bits	8502 (LFR) (R/W)	El valor indicado en el indicador se envía como nueva referencia de frecuencia en cada ciclo de programa.
Texto	Valor referencia de frecuencia	Entero (INT)	Lectura de 16 bits	3203 (FrH) (R)	El valor de referencia de frecuencia se lee a través de un registro habilitado para ello.
Texto	Intensidad del Motor	Entero (INT)	Lectura de 16 bits	3204 (LCr) (R)	Se muestra en un campo de salida el valor de la intensidad del motor con escala de 0,1 A.
Campo de entrada	Gráfico de Intensidad	Entero (INT)	Lectura de 16 bits	3205 (LCr) (R)	El objeto campo de entrada se convierte en un gráfico de tendencia para mostrar el estado de la intensidad en el tiempo.
Texto	Tensión del motor	Entero (INT)	Lectura de 16 bits	3208 (UOP) (R)	Se muestra en campo de salida el valor de la tensión del motor con escala de 1V.
Campo de entrada	Gráfico de Tensión	Entero (INT)	Lectura de 16 bits	3208 (UOP) (R)	El objeto campo de entrada se convierte en un gráfico de tendencia para mostrar el estado de la intensidad en el tiempo.
Texto	Par Motor	Entero (INT)	Lectura de 16 bits	3216 (Otrn) (R)	Se muestra en un campo de salida el valor del Par motor con escala de 0,01 Nm.
Campo de entrada	Gráfico Par	Entero (INT)	Lectura de 16 bits	3216 (Otrn) (R)	El objeto campo de entrada se convierte en un gráfico de tendencia para mostrar el comportamiento del Par Motor en el tiempo.
Texto	Velocidad	Entero (INT)	Lectura de 16 bits	8641 (FrOd) (R)	Se muestra en un campo de salida la velocidad del motor en tiempo real.
Campo de entrada	Gráfico Velocidad	Entero (INT)	Lectura de 16 bits	8641 (FrOd) (R)	El objeto campo de entrada se convierte en un gráfico de tendencia para mostrar el comportamiento en tiempo real de la velocidad en el eje del motor.

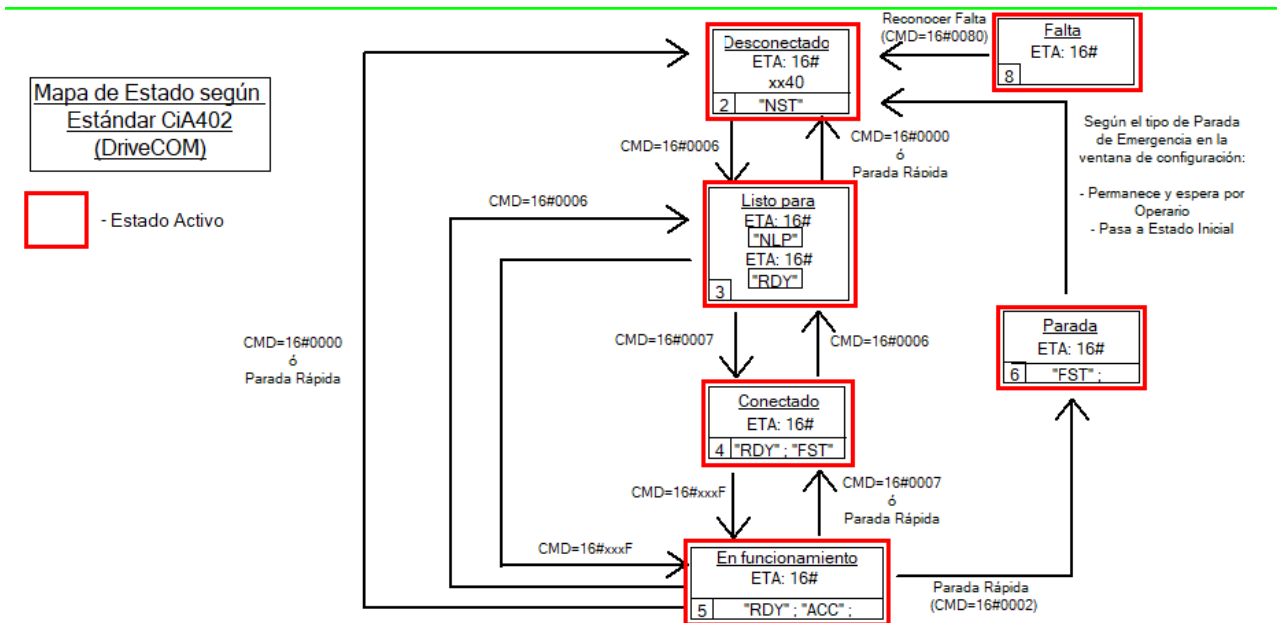
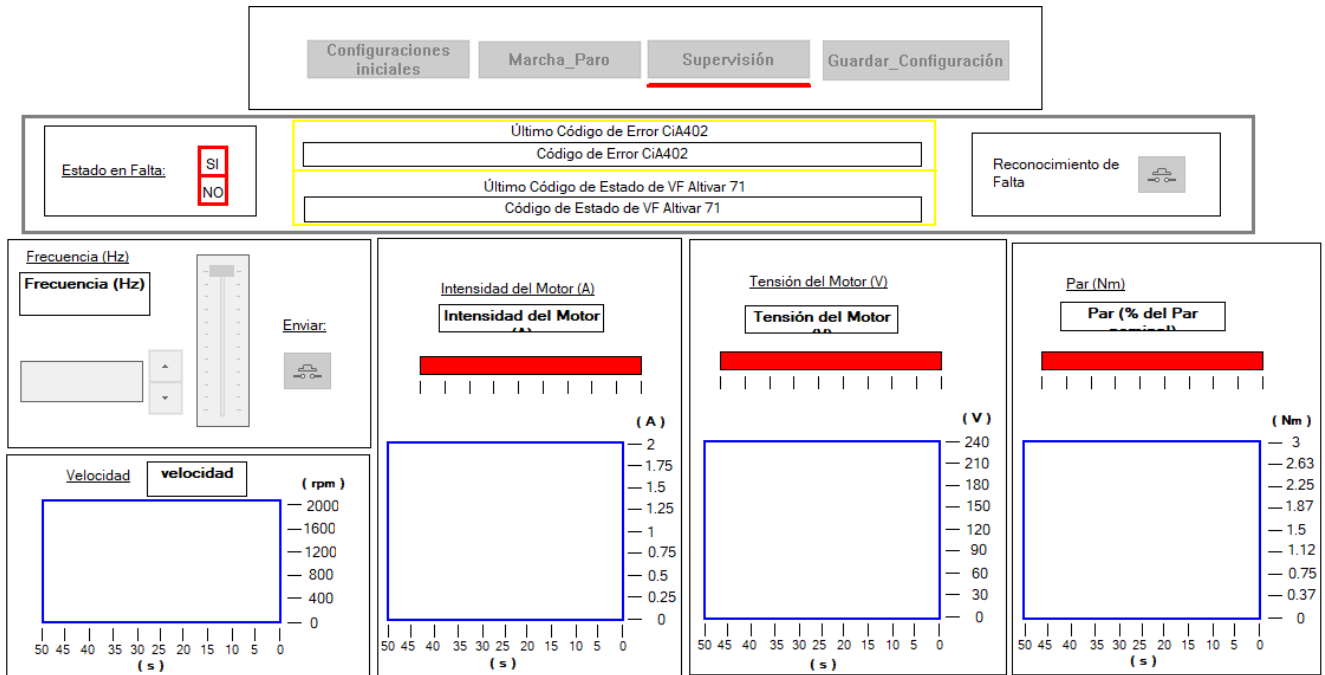


Figura 62. Diseño de la Pantalla de Operador: Supervisión

#### 4.2.4 Pantalla 3: Guardar Configuración

El desarrollo de esta pantalla de operador pretende englobar en un mismo sitio las funcionalidades de acceso a memoria. Tanto una orden de escritura o lectura en una EEPROM determinada deben ejecutarse cuando el motor esté detenido y no en *Estado 5*, de no ser así puede saltar un fallo por seguridad.

Para que las acciones que se ejecuten en esta pantalla de operador surjan efecto, es necesario que en el variador de velocidad se haya introducido el código de acceso de operario en el *MENÚ GENERAL* (Figura 63). Por defecto el código de acceso es 6969 y se ha preferido dejarlo así pues no interviene con los objetivos académicos del proyecto. El sentido de esta norma de seguridad por defecto radica en que a través de las opciones de esta pantalla de operador se pueden modificar y borrar las configuraciones establecidas e incluso, reiniciarlas de fábrica.

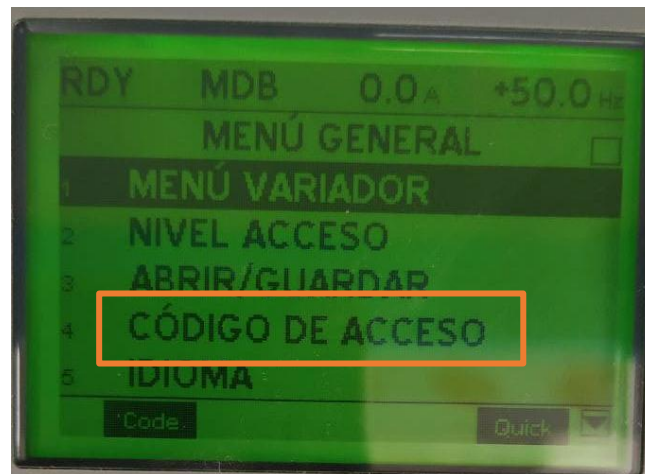


Figura 63. Opción del el Altivar 71 para introducir el código de acceso que permite el trabajo con memoria

La ventana que muestra el numero de la EEPROM con la configuración en uso hace uso de un registro específico y no es necesaria una lectura adicional: Configuración Activa (CNFS) con dirección Modbus 8020. En la Figura 65 se puede observar cómo se dotó la capacidad de cargar y guardar una configuración en una EEPROM determinada.

Desde esta pantalla de operador también es posible reiniciar los valores a configuración de fábrica. Es posible reiniciar todo el valor, o una parte de los mismos. En la Pantalla Guardar configuración se muestran las opciones posibles. Para ejecutar esta acción es necesario enviar inicialmente el grupo de parámetros a reiniciar (Lista de grupos de parámetros -FrY- 3022) y luego enviar el valor 64 al mismo registro que se utiliza para cargar una configuración, “Cargar configuración” (FCS) dirección Modbus: 8002. En este caso, en lugar de acceder a una EEPROM determinada, hace uso de una sección de memoria con los datos de fábrica.

Dentro de la lógica de programa se añadió la capacidad de impedir que se ejecute una configuración de fábrica si no se introduce un PIN (Figura 65). El pin a validar es 6969, al igual que el código de acceso por defecto para evitar posibles confusiones. De esta manera se evita que por error se eliminen datos sensibles.

En la Tabla 8 se describen de manera detalla las características de los objetos utilizados que componen la pantalla de operador *Guardar\_configuracion* (Figura 65), cuya lógica de funcionamiento se describe en el diagrama de la Figura 64.

Es necesario destacar que el acceso con frecuencia a la memoria puede disminuir su vida útil, de hecho, están concebidas para tener una vida de 100 000 ejecuciones de lecturas o escrituras.

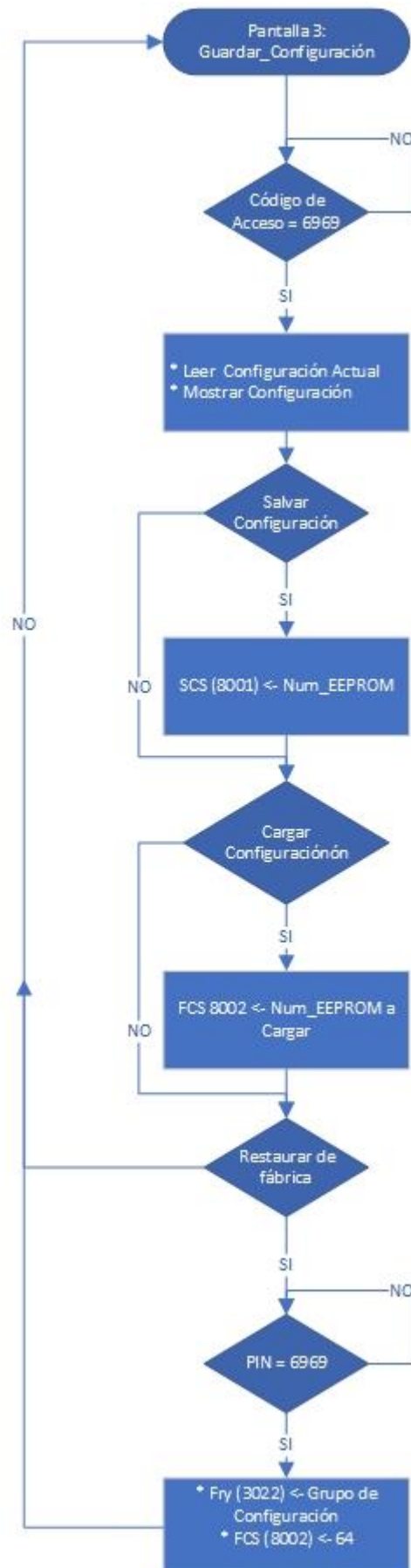
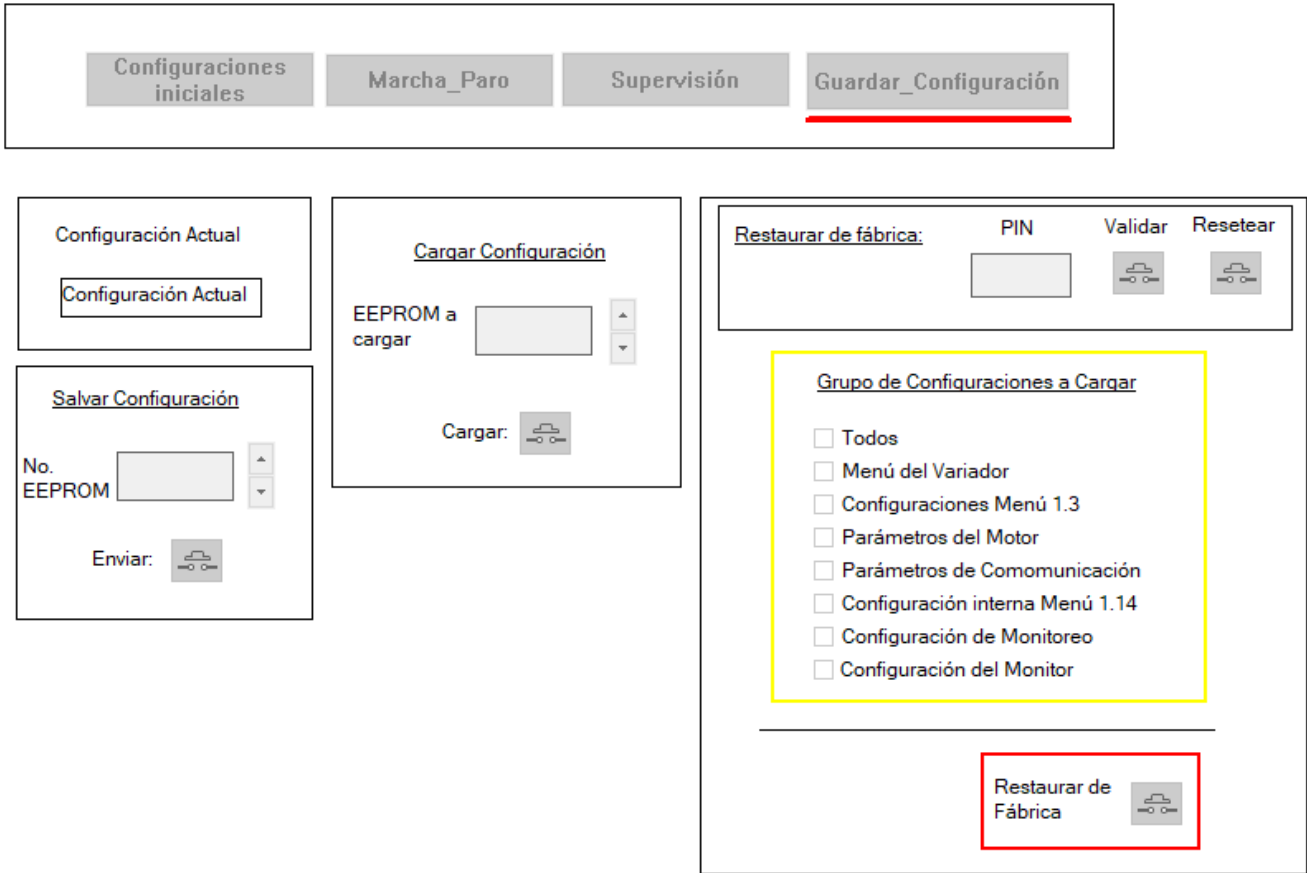


Figura 64. Lógica de programación de la Pantalla de Operador: Configuraciones Iniciales

Tabla 8. Elementos que componen la Pantalla de Operador #3: Guardar\_Configuración.

Tipo de Objeto en Unity Pro	Nombre de Objeto	Tipo de Dato	Acción MODBUS o Programa	Dirección Lógica, Código y tipo (R-Lectura, W-Escritura)	Comentario
Texto	Configuración actual	Entero (INT)	Lectura de 16 bits en cada ciclo de programa	8020 (CNFS) (R)	Se muestra la configuración presente en el Altivar 71.
Texto	Salvar configuración	Entero (INT)	Escritura en un registro de 16 bits	8001 (SCS) (R/W)	Este registro permite la posibilidad de seleccionar en dependencia del número que aquí se inserte, la memoria en la que se almacenará la configuración actual de tres posibles (0, 1 o 2).
Botón de Comando	Enviar configuración	Booleano (BOOL)	Escritura en un registro de 16 bits	8001 (SCS) (R/W)	Se guarda la configuración en el número de la EEPROM indicada en el bloque Salvar Configuración.
Texto	Cargar Configuración	Entero (INT)	Escritura en un registro de 16 bits	8002 (FCS) (R/W)	El número insertado en este bloque será la configuración que se cargará desde las 3 posibles EEPROM del ATV71 más otros dos estados, siendo las posibilidades 0=no restaurar, 1=cargar configuración de EEPROM 0, 2=Cargar configuración de la memoria 1, 3= cargar configuración de memoria 2 y 64=Restaurar a configuración de fábrica.
Botón de Comando	Cargar	Booleano (BOOL)	Escritura en un registro de 16 bits	8002 (FCS) (R/W)	Botón para ejecutar la acción de cargar la configuración.
Campo de Entrada	PIN	Entero (INT)	Validar que el valor sea 6969	-	Mediante un bucle sencillo se comprueba que el pin insertado sea igual a 6969, de ser así se muestran el resto de objetos.
Botón de Comando	Grupo a Cargar	Entero (INT)	Escritura en un registro de 16 bits	3022 (FrY) (R/W)	El número insertado en este bloque corresponderá con el grupo de configuraciones que se reiniciarán de fábrica. Pudiendo seleccionar, por ejemplo, que se carguen sólo los parámetros configurados en el monitor, los del motor, o todos.



Mapa de Estado según Estándar CiA402 (DriveCOM)

- Estado Activo

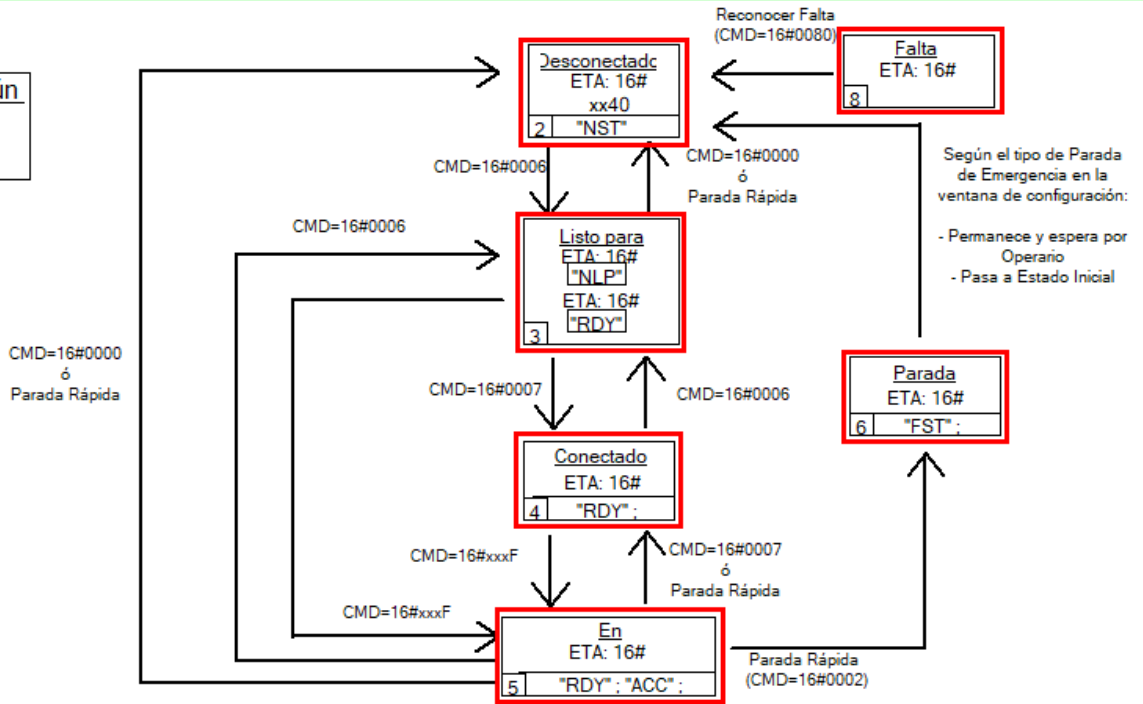


Figura 65. Diseño de la Pantalla de Operador: Guardar Configuración





# 5 EJEMPLO DE FUNCIONAMIENTO

El desarrollo de este capítulo abarca la puesta en marcha del HMI y el desarrollo de un ejemplo de uso muy enfocado para las primeras tomas de contacto en prácticas de laboratorio.

Inicialmente es necesario energizar el Variador de Frecuencia y el PLC, para dar paso al control a través de las pantallas de operador diseñadas en el apartado anterior. El fichero de Unity Pro fue nombrado como: “m340\_modbus\_hmi.stu”, una vez abierto, basta con realizar doble clic sobre las pantallas de operador diseñadas (Figura 66) para navegar entre ellas y comenzar a ejecutar comandos y tareas de supervisión.

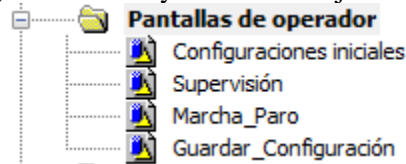


Figura 66. Pantalla de operador que componen el HMI

Para conectar el HMI con el PLC, dado que ambas partes están ejecutando el mismo programa, es necesario presionar el botón “Conectar” que se observa en la figura 67 y luego habilitar el funcionamiento clicando en “RUN” (Figura 68). En caso de que el PLC, por razones ajenas, o por tiempo en desuso, no cuente con la misma versión cargada; es necesario transferir la versión actual para poder intercambiar instrucción con el HMI. Basta con clicar sobre el icono de la Figura 69 para incorporar en el PLC la misma versión que en Unity Pro.



Figura 67. Botón “Conectar”.

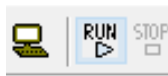


Figura 68. Botón “RUN”.



Figura 69. Botón “Enviar Proyecto a”

Antes de pasar al ejemplo propuesto, para poder interactuar con las pantallas de operador, es fundamental habilitar las variables de escritura (Figura 70) en cada una de ellas, de lo contrario será imposible introducir datos o presionar botones de comandos. Por otra parte, el uso de la pantalla completa puede mejorar exponencialmente la experiencia de usuario (Figura 71).

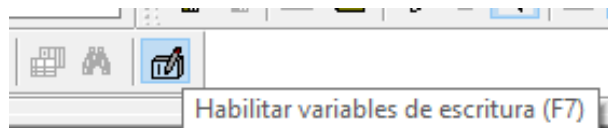


Figura 70. Botón “Habilitar Variables de escritura”.

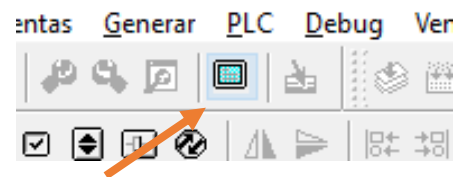


Figura 71. Botón de HMI en Pantalla Completa

## 5.1 Guía

Con este ejemplo se pretende hacer uso de las opciones más críticas y relevantes de cada pantalla de operador e incorporar aquellas que pueden ser relevantes para un estudiante que está realizando una práctica de laboratorio. De esta manera se da por descartado exponer el uso de todas y cada una de las opciones presentes en el proyecto.

El intercambio de instrucciones entre el HMI y el PLC para actuar sobre el variador de frecuencia debe comenzar en la pantalla de operador *Configuraciones Iniciales* (Figura 72). Inicialmente el Motor estará desconectado (Figura 73) y por tanto, en el *Estado 2* del Perfil CiA402.

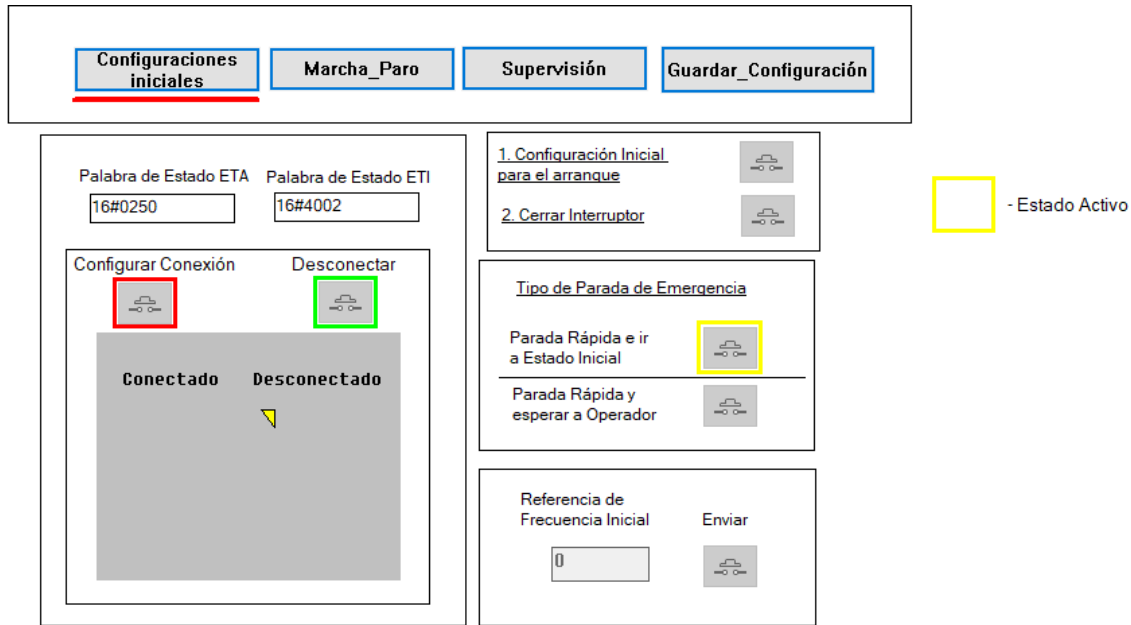


Figura 72. Parte superior de la pantalla de operador “Configuraciones Iniciales”

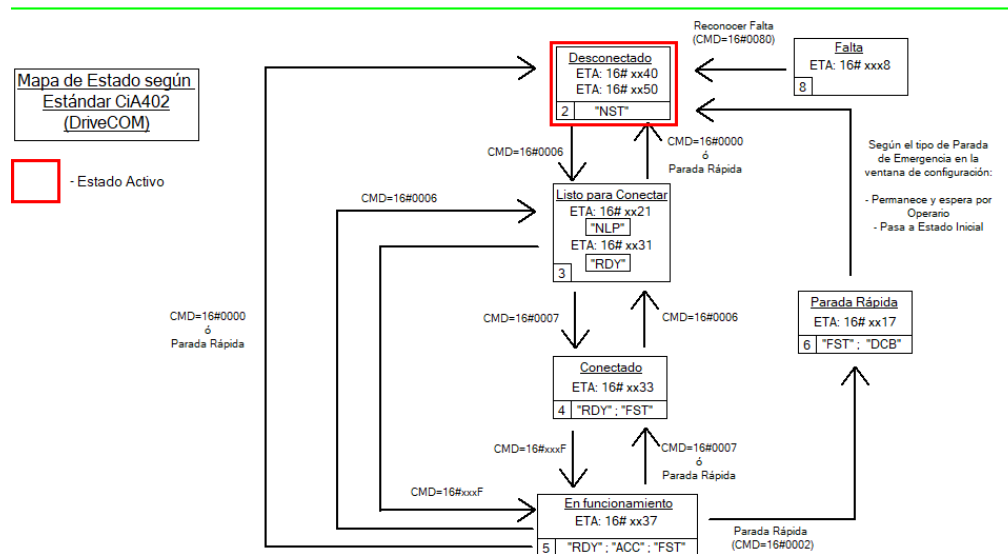


Figura 73. Mapa de Estado CiA402 dispuesto en la parte baja de Configuraciones Iniciales

El paso lógico siguiente sería presionar el botón de *Configurar Conexión* (Figura 74). Esta acción no significa necesariamente que se ha suministrado tensión en bornas del motor; pero sí que los mecanismos de seguridad internos del Variador de tensión están activos para el siguiente paso. De esta manera el motor pasa al *Estado 3* indicando que está “Listo para conectar” (Figura 75).

Figura 74. Pantalla Configuraciones Iniciales al presionar "Configurar Conexión"

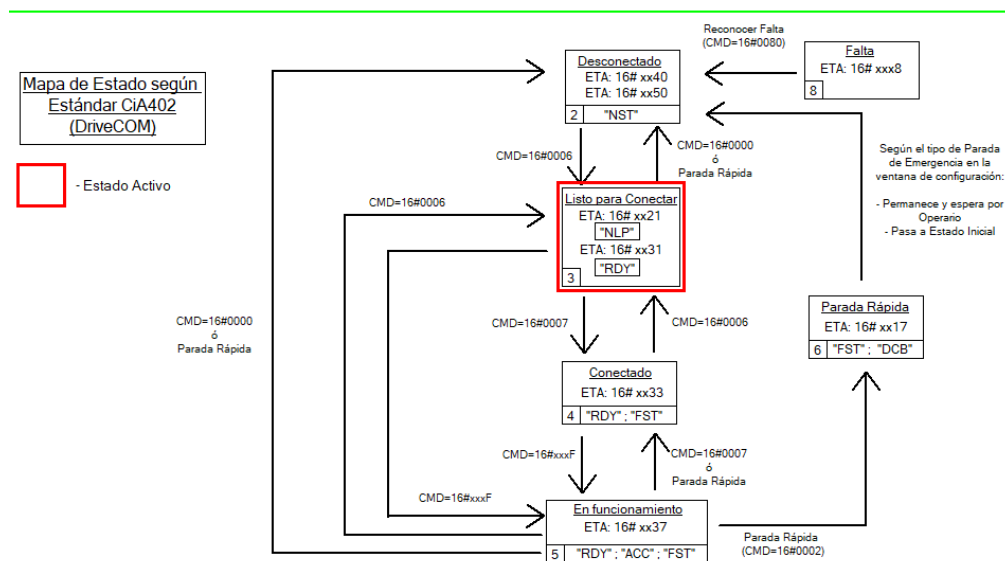


Figura 75. Mapa de estado en la parte inferior de la pantalla mostrando el Estado 3 como estado activo.

En este estado, si se ha enviado una referencia inicial, el Altivar puede ejecutar la orden de arrancada saltándose el *Estado 4*. Pero en este ejemplo se pretende describir un escenario donde el eje del motor manipula cargas delicadas o esteras en pendiente y es conveniente que el eje esté bloqueado antes del arranque. Por tal motivo se presiona el botón "2. Cerrar Interruptor" y el sistema conmuta al *Estado 4* (Figura 76) donde el motor está energizado y bloqueado. Presionar "Configuración Inicial para el arranque" ejecuta la misma acción (CMD=16#0006) que llevó al sistema del *Estado 2* al *Estado 3*; y ejecutarla desde el *Estado 4* actual, conmutaría al anterior.

En cuanto al tipo de parada de emergencia se deja inicialmente su valor por defecto: "Parada Rápida e ir al Estado Inicial" (Figura 77). Posteriormente se envía un valor de referencia inicial, en este caso 50 Hz (Figura 78). El siguiente paso dentro del flujo de programa con el objetivo del poner el motor en marcha, sería en la pantalla de operador *Marcha\_Paro*.

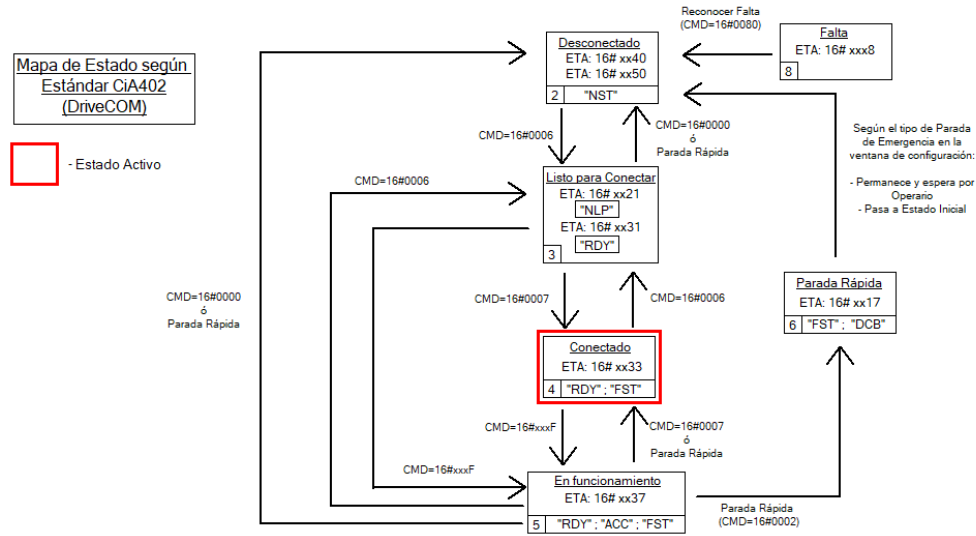


Figura 76. Mapa de estado, con Estado 4 activo, luego de presionar “2. Cerrar Interruptor”.

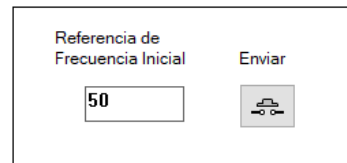
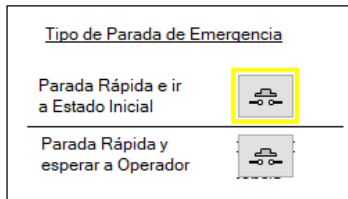


Figura 77. Tipo de Parada de Emergencia seleccionada por defecto.

Figura 78. Valor de Referencia Inicial

Inicialmente en la pantalla de operador “Marcha\_Paro” el motor está detenido (Figura 79) y presionando el botón de “Marcha” (Figura 80) se observa que la flecha de indicación conmuta su estado y posteriormente el motor pasa al Estado 5 (Figura 81). El motor está en marcha siguiendo la referencia establecida en “Configuraciones iniciales”.

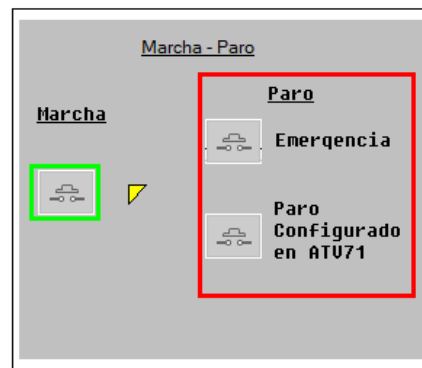
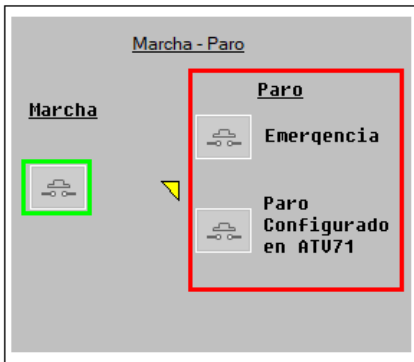


Figura 79. Indicador señalando que el Motor está detenido.

Figura 80. Motor en Marcha

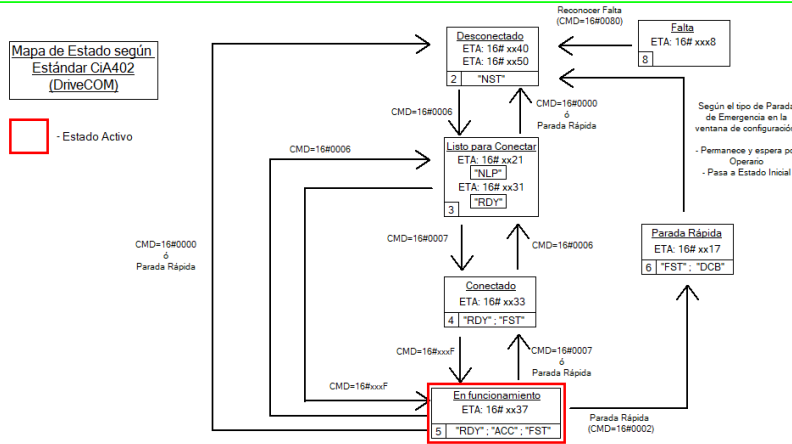


Figura 81. Estado 5 Activo, motor en funcionamiento.

Por defecto el sentido de giro es horario, visto desde el eje del motor y se animó una figura que ayuda a identificar rápidamente el sentido del movimiento (Figura 82) pues ilustra el motor con el freno de histéresis conectado justo como lo verían en el laboratorio.



Figura 82. Animación de motor para mostrar el sentido de giro.

Si se ejecuta una parada, presionando el botón “Paro Configurado en ATV71”, se lanza un cartel de advertencia (Figura 83) indicando que el motor, aunque esté detenido permanece en Estado 5 (Figura 84) y por tanto no es posible realizar configuraciones permanentes ni accesos a memoria del variador de frecuencia.

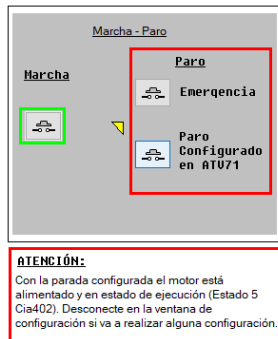


Figura 83. Advertencia indicando que el sistema permanece en Estado

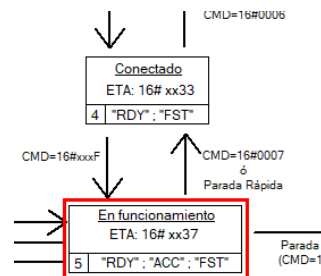


Figura 84. Al ejecutar una Parada de Emergencia a través de Paro Configurado, se permanece en Estado 5.

Este tipo de parada es muy útil en escenarios muy puntuales y su principal ventaja radica en que se puede ejecutar a continuación una orden de marcha sin necesidad de otra configuración o cambio de estado. Si se vuelve a poner el motor en marcha y a continuación presiona el botón de “Emergencia” (Figura 85), el

siguiente estado del sistema dependerá de la configuración de parada establecida en la pantalla *Configuraciones iniciales* (Figura 72). Esta configuración actúa sobre el parámetro QSTD (Código de Parada de Emergencia) y como inicialmente se había dejado la configuración por defecto, “Parada Rápida e Ir al estado Inicial”, el flujo de programa se detiene en el *Estado 2*. Se vuelve al principio, motor detenido y desconectado (Figura 86).

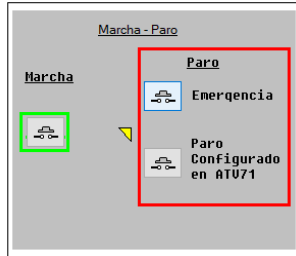


Figura 85. Orden de parada siguiendo el Botón de Emergencia

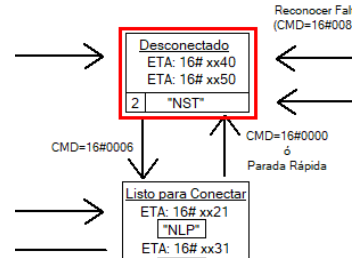


Figura 86. Motor detenido y desconectado, Estado 2 activo.

Si se repiten todos los pasos para llegar al *Estado 5* (Motor en marcha) y dentro de la pantalla “*Configuraciones iniciales*” se selecciona la opción “Parada Rápida y esperar a Operador” (Figura 87), al presionar nuevamente el botón “Emergencia” en la pantalla “Marcha\_Paro” (Figura 88) el flujo de programa se detiene en el *Estado 6* (Figura 89), motor detenido.

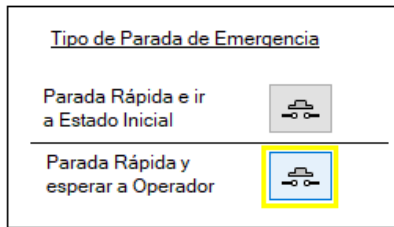


Figura 87. Selección de tipo de parada: Parada Rápida y esperar a operador

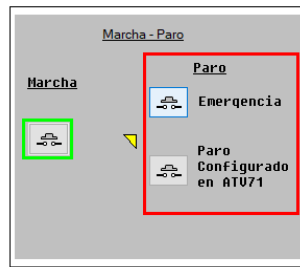


Figura 88. Se ejecuta nuevamente una parada de Emergencia

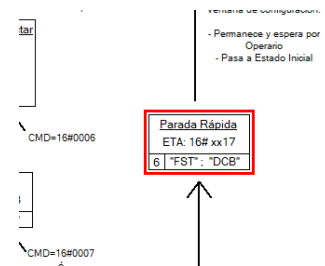


Figura 89. Estado 6 Activo, pues se modificó QSTD.

En este escenario, el sistema se puede desbloquear cuando un operador presiona el botón “STOP” en la pantalla del variador o simplemente desconectando el motor en la pantalla “*Configuraciones iniciales*” para nuevamente llegar al estado inicial, el *Estado 2*. Su principal uso viene dado cuando es necesario realizar una inspección del equipo o reconfigurar el sistema después de la parada.

Si bien con las operaciones dentro de estas dos pantallas de operador existe la posibilidad de conmutar entre diferentes estados del sistema, en “*Supervisión*” los objetos cobran importancia cuando el motor está en marcha (*Estado 5*) (Figura 90). Esto se debe a que la mayoría de los objetos que la componen son de tipo salida, encargados de mostrar valores o gráficos de tendencia a partir de variables del motor. Los campos que muestran los mensajes de error no necesitan que el sistema se encuentre en *Estado 5*, los registros a los que acceden tienen información en todo momento. Estos campos estarán siempre activos y mostrarán los últimos códigos de error de los estándares CiA402 y los propios del ATV71. Se mostrarán, aunque exista un fallo interno o se haya reiniciado el variador, siempre y cuando exista comunicación.

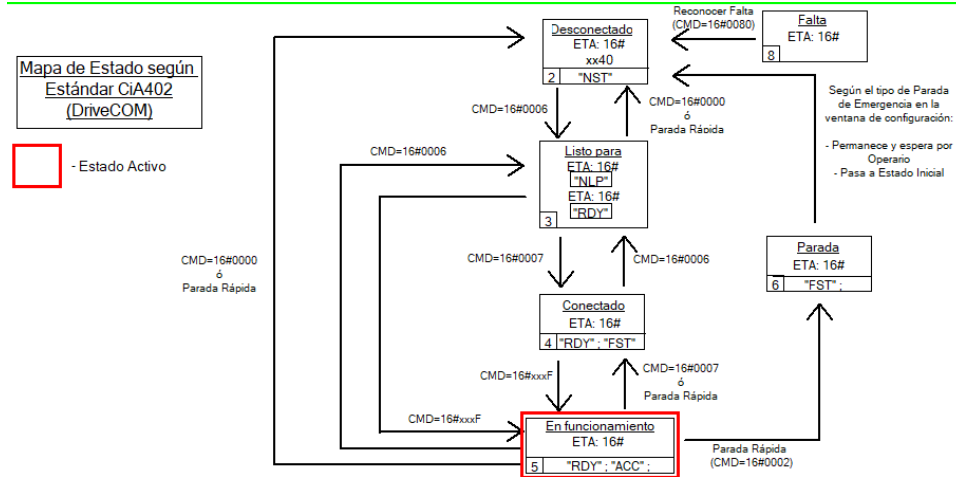


Figura 90. La ventana de operador "Supervisión" cobra importación mientras el estado permanezca en Estado 5.

Dentro de la pantalla de operador "Supervisión" las variables del Motor que se monitorizan son: Intensidad (A), Tensión (V), Par (Nm), Velocidad (rpm) y Frecuencia (Hz) (Figura 91). Para mostrar los valores medidos se utilizaron gráficos de tendencias, gráficos de barra horizontales y campos de salida numéricos con sus respectivos valores. De esta manera se pretende monitorizar en todo momento el comportamiento de estas variables. El sistema constantemente sigue un valor de frecuencia y es en esta ventana donde se puede introducir un valor de referencia diferente a la inicial, introducido anteriormente en la pantalla *Configuraciones iniciales*.

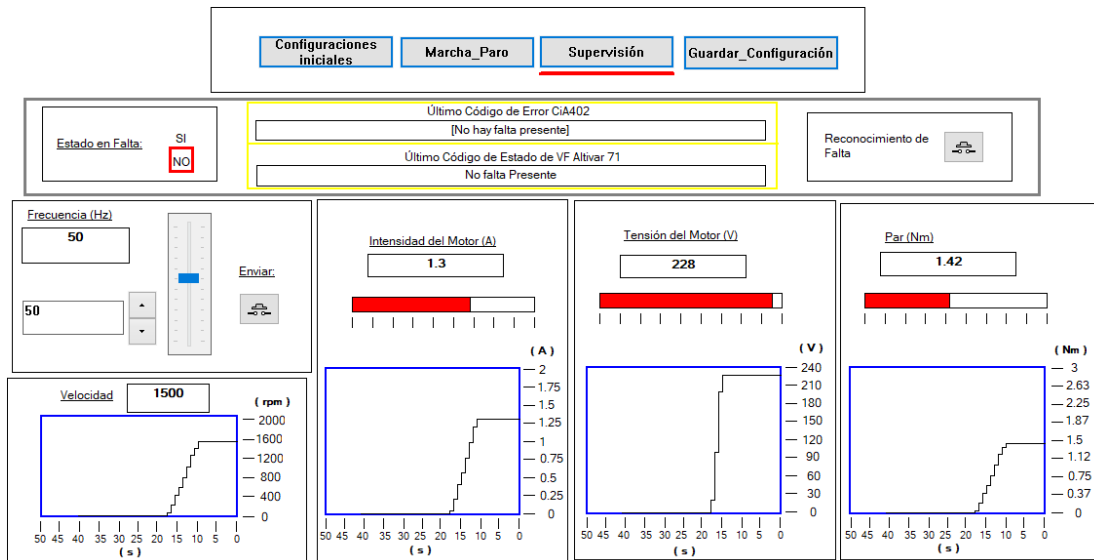


Figura 91. Parte superior de la Pantalla de Operador "Supervisión" donde se monitorizan los valores.

Un cambio en la referencia de frecuencia entrega bastante juego para entender el comportamiento de los parámetros del motor y su relación. Por otra parte, el uso del freno magnético puede arrojar escenarios con mucho valor académico para explicar el comportamiento de las máquinas eléctricas ante cargas variables. La Figura 92 muestra el sistema en un caso donde se varió la frecuencia de 50 Hz a 44 Hz.

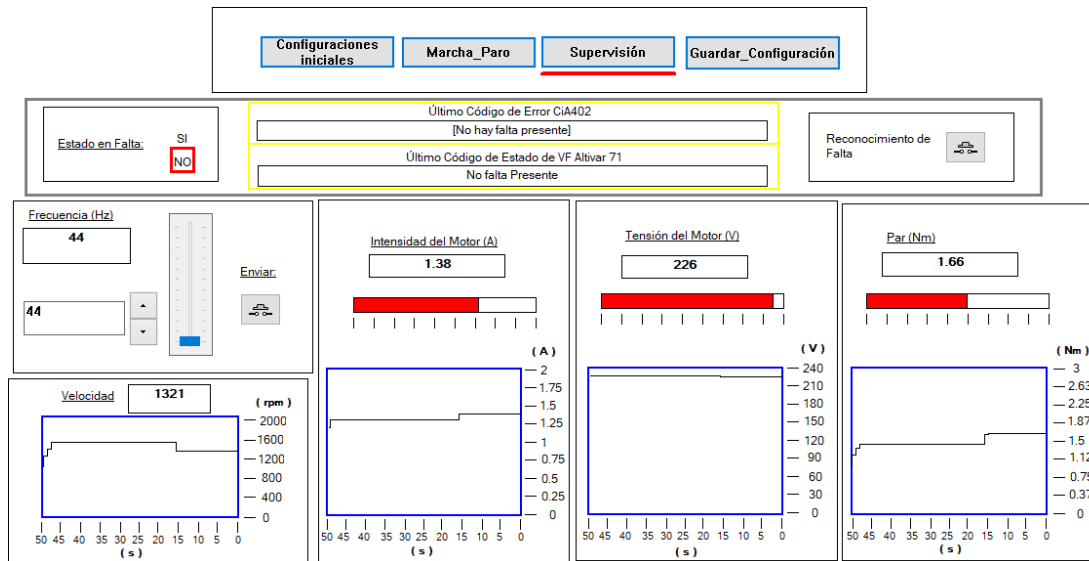


Figura 92. El sistema antes pequeñas variaciones en la referencia de frecuencia.

Para guardar la configuración establecida en el variador de frecuencia es necesario ir a la pantalla de operador "Guardar\_Configuración" (Figura 65). El variador de frecuencia utilizado dispone de tres memorias para almacenamiento de múltiples configuraciones de motor. Las mismas están enumeradas del 0 al 2 y como se puede observar en la Figura 93, la configuración actual esta almacenada en EEPROM número 0 (flecha azul). En el supuesto donde se quiere cambiar ciertos parámetros para ejercer determinadas pruebas y guardar la configuración actual, basta con seleccionar la memoria donde se desea salvar la configuración. En el ejemplo se escogió la EEPROM número 2 (flecha naranja) y se pulsó en "Enviar".

Mientras el sistema está en funcionamiento, las opciones para restablecimiento de fábrica que se presentaron en el apartado 4.2.4 no están disponibles, necesitando introducir un PIN para acceder (flecha verde). Esto impide que de manera accidental se elimine la configuración establecida.

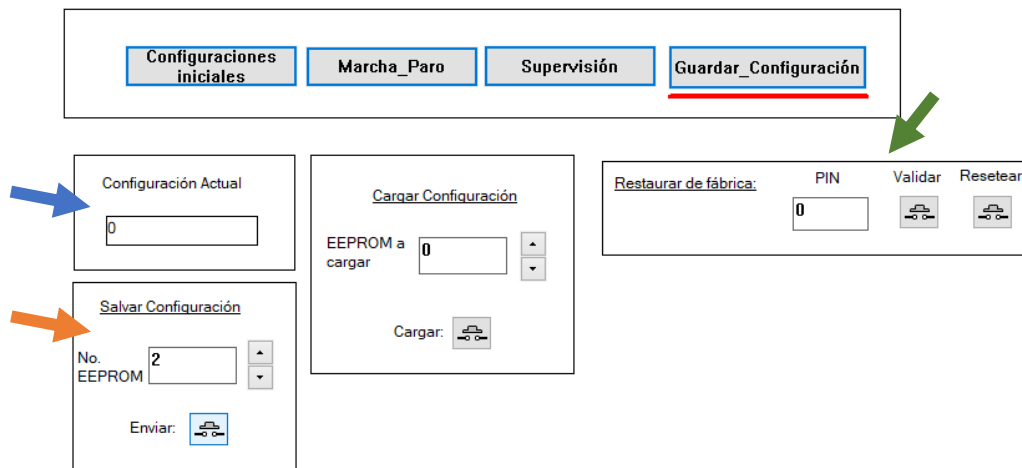


Figura 93. Parte superior de la pantalla "Guardar\_Configuración"

Si se introduce el valor 6969 en el campo "PIN" y posteriormente se pulsa en Validar aparecen las opciones de la Figura 94.



Figura 94. Pantalla "Guarda\_Configuración" con las nuevas opciones después de introducir el PIN.

Hasta aquí se ha presentado un ejemplo de flujo de trabajo haciendo uso del proyecto. Durante el mismo se navegó por las opciones más relevantes pudiendo variar en dependencia de las necesidades. El desarrollo de este ejemplo pretende servir como toma de contacto inicial de los estudiantes en prácticas de laboratorio, ayudando a desarrollar los conocimientos de las comunicaciones industriales y el control de máquinas eléctricas.

Los equipos a disposición de los estudiantes se pueden observar en la Figura 95, de izquierda a derecha: bancada: freno magnético + motor de inducción + *encoder*; variador de frecuencia, PLC y ordenador.



Figura 95. Montaje en laboratorio de todos lo equipos del sistema.

Como se puede observar, quedan muchos equipos y herramientas que pueden ser explotadas para profundización de los conocimientos en estas áreas. En el siguiente capítulo se proponen futuras líneas de investigación haciendo de funcionalidades no empleadas en este proyecto.



# 6 CONCLUSIONES

---

En función de los objetivos planteados en el proyecto, considerando el desarrollo de cada uno de los capítulos y para dar respuesta al objetivo principal, se ha desarrollado un HMI para el control y monitorización de bancada de motor. Para llegar a estos resultados, fue necesario la cumplimentación de cada uno de los objetivos individuales.

Se detallaron las principales características de cada uno de los componentes del sistema, los tipos de comunicación que utilizaban y la relación entre los mismos. Además de mostrar sus principios de funcionamiento y las configuraciones realizados tanto en el variador de frecuencia como del PLC utilizado para asegurar su compatibilidad.

Para la puesta en marcha del variador de frecuencia se mostraron las configuraciones ejecutadas en el ATV71, así como el orden de las mismas para evitar faltas graves. Para lograr una familiarización con el software de programación Unity Pro XL, se explicaron las configuraciones de puesta en marcha, el tipo de lenguaje de programación seleccionado y el uso de los menús de configuración más utilizados.

Para la elaboración de cada una de las pantallas de operador que forman el HMI se detalló su lógica de programación y cada uno de los elementos que las componen, ayudándonos de tablas y de diagramas de flujo de fácil comprensión. Una vez planteado el funcionamiento del sistema y la importancia del estándar CiA402 en el mismo, se expuso cómo se utilizó este para el cumplimiento de los objetivos y los principales estados por lo que transita.

Con el fin de cumplir los objetivos docentes del proyecto y servir como punto de partida para el desarrollo de futuras líneas de investigación, se desarrolló un capítulo donde se expone una guía de uso de este proyecto. Esta guía se centra en resaltar las principales características con las que se dotó el control y sus principales alternativas en el flujo de programa

Este proyecto pretende fomentar el uso de materiales propios de la universidad para tareas de investigación y enseñanza práctica. Se posiciona como una guía para expandir los conocimientos en el área de comunicaciones industriales y el control de maquinas eléctricas.

## **Futuras líneas de investigación**

Si bien el desarrollo del proyecto culmina con el cumplimiento de los objetivos propuestos, se consideran que aún existen líneas de investigación no exploradas en nuestros objetivos y que constituyen invitaciones para profundizar en áreas relevantes de este campo.

1. A través de las entradas analógicas y digitales dispuestas en el Modicon M340, se puede desarrollar escenario de control más complejos, donde el algoritmo de control manipule múltiples variables. Esto junto a al uso de las tareas tipo eventos y de tipo FAST posibilita hacer un correcto uso de interrupciones y de un algoritmo de control muy eficaz.
2. Teniendo en cuenta que el variador de frecuencia en cuestión, es utilizado como mecanismo de acción final y de seguridad, sus múltiples opciones de configuración le permiten tener un alto índice de escalabilidad. De esta forma el sistema puede ser adaptado para su integración en otros tipos de motores o en sistemas donde es necesario emplear un alto nivel de configuración a nivel de variador de velocidad.
3. La configuración por defecto para la manipulación de archivos del ATV71 es multiconfiguración. Como se explicó, solo es necesario introducir el PIN 6969 en el nivel de acceso del variador y disfrutar de la capacidad de almacenar distintas configuraciones del motor en diferentes EEPROM. Una alternativa muy valiosa sería utilizar el sistema en multimotor y configurando correctamente los circuitos de conmutación, se logra un sistema capaz de controlar procesos industriales en cascada con un solo variador de frecuencia, incurriendo en ahorros significativos.



# REFERENCIAS

---

- [1] S. Electric, Modicon M340 automation platform, Francia: Schneider Electric Industries SAS, 2014.
- [2] Julián Rodríguez Fernández, Luis Miguel Cerdá Filiu, Roberto Bezos Sánchez-Horneros, Automatismos industriales, Madrid España: Ediciones Paraninfo, SA, 2014.
- [3] CIME Motors, 3ra Ed, Motores asíncronos Trifásico y Monofásico (Catálogo) recuperado de <http://garciacostas.com/catalogos/cosgra/cime/cime.pdf>
- [4] Valderas, D., Sancho, J. I., Benli, K., Lehmann, H., Lancaster, G., Gurruchaga, K., Del Portillo, J. (2021). «Modeling the impact on wheel sensor readouts by eddy current brakes installed in high-speed trains». IEEE Transactions on Vehicular Technology, Vol. 70, n° 6, pp. 5592 – 5604, 2021
- [5] Gembali, S.G., Rao, A.N.B., Naresh, H. «Design and experimental analysis of electro magnetic braking system», Materials Today: Proceedings, n° 45, pp. 2833-2839, 2021.
- [6] Dutton, Guillermo., (Ed. 2016) Electromagnetic Components, Estados Unidos, Orange Apple
- [7] Gallo, Omar D.; Gonella, Rubén M. J.; Ferreyra, Diego M.; Díaz, Alberto J., «Freno de corrientes parásitas para ensayo de motores eléctricos», Ciencia. Docencia, Tecnología, Vol. 25, n° 54, pp. 283-296, 2017.
- [8] West Instruments S.A. Manual de Aplicación de encoders, México.
- [9] Schneider Electric, 2010, Unity Pro Lenguajes y estructura del programa. Manual de Referencia.
- [10] Felser, Max., Comparison of different interfaces and profiles for power drive systems, Berne University of Applied Sciences
- [11] Schneider Electric, 2013, Altivar 71 Communication parameters. Manual de Usuario.

# GLOSARIO DE TÉRMINOS Y SIGLAS

---

**Altivar:** Marca de variadores de frecuencia antiguamente de Telemecanique® actualmente fabricada y comercializada bajo Schneider Electric®. Son equipos muy utilizados para el control de movimiento en la industria, operando directamente sobre los motores.

**Bus de Campo:** Sistema jerárquico de control en la industria que comunica a los sensores y actuadores con su inteligencia de nivel de campo con niveles superiores como Control o Supervisión.

**CANopen:** Es un protocolo de comunicación muy utilizados en automatización, creados sin ánimo de lucro por fabricante y usuarios del bus CAN asociados dentro de una organización llamada CiA (*CAN in Automation*).

**CMD:** *Command Word*. Palabra de control del Altivar controlada mediante Modbus con dirección lógica 8501. Principal agente para el cambio de estados del Altivar durante el control de Motores.

**Codificador Incremental:** Del inglés *encoder* se entienden como sensores que convierten el movimiento en señales eléctricas que previamente fueron codificadas para conocer la velocidad y dirección de un movimiento específico. Generalmente acoplados en ejes de motores, ruedas y engranajes.

**EEPROM:** *Electrically Erasable Programmable Read-Only Memory*. (ROM programable y borrrable eléctricamente). Se utilizan comúnmente como sistema de almacenamiento de datos sin necesidad de estar energizado para la permanencia de los mismo.

**ETA:** *Status Word*. Palabra de estado del Altivar controlada por Modbus, en la dirección 3201. Indica los estados de alarmas, tipos de paradas y el chequeo de las referencias, entre otros.

**Ethernet:** Es un estándar de redes de computadoras para el desarrollo y uso de redes LAN (Red de Área Local). Se enfoca en definir tanto el cableado y señalización en la capa física, como los formatos de las tramas de dato en la capa de enlace del modelo OSI.

**ETI:** *Extended status Word*. Palabra extendida de estado del Altivar con dirección Modbus: 3206. Muestra las etapas de aceleración y frenado por lo que transita el motor, la consistencia de los parámetros y el puerto por el que se realiza el control.

**HMI:** *Human-machine interface*. Comúnmente están compuestos por interfases software o hardware donde un usuario pueda interactuar con un sistema. En la actualidad se define como pantallas donde se manipulan variables de una máquina o subproceso; es un término muy utilizado en entornos industriales.

**IP XX:** *Ingress Protection*. Índice de Protección o Grado de Protección, indica el valor de resistencia de un equipo al polvo y líquidos. El primer valor después de las letras “IP” indica la resistencia al polvo y partículas con un máximo de escala de 6; el segundo la resistencia frente a líquidos con un máximo de 8.

**PLC:** *Programmable Logic Controller*. En castellano es conocido como Controlador Lógico Programable o Autómata Programable. Es un dispositivo con amplia penetración en la industria para el control de equipos individuales o líneas de producción enteras, con la capacidad de controlar, alimentar y procesar datos de sensores y actuadores de un proceso. Destaca además por disponer de rangos de temperaturas ampliados, amplio abanico de comunicaciones, inmunidad al ruido y resistencias a vibraciones e impactos.

**QSTD:** *Quick stop option code*. Constituye el código de parada rápida del Altivar con la dirección Modbus: 8651. Es configurable a través de dos valores: el valor por defecto es 6, donde si se ejecuta una orden de

parada el sistema espera por la intervención del usuario, o 2 donde el sistema ejecuta la parada rápida y recorre hasta el estado inicial.

**RAM:** *Random Access Memory*. Traducido al castellano sería: Memoria de Acceso Aleatorio y son utilizados por las computadoras o dispositivos con capacidad de cómputo para almacenar las instrucciones que van ejecutando. No constituyen almacenamiento permanente, sino memoria para almacenar datos de procesos durante determinado tiempo.

**RJ45:** Constituye una interfaz física ligada al cableado de redes de computadoras. Dado que cuenta con ocho *pin*s (clavijas), es común verlo como extremo de cables UTP de par trenzado. También es común en la industria verlo albergar protocolo como RS 485 que solo utilizan tres de sus *pin*s.

**RS 232:** También conocida como RS 232C, es una interfaz física concebida para el intercambio de datos entre un terminal y un equipo de comunicación como un módem. En sus inicios fue muy utilizado para la interconexión de periféricos y aunque actualmente el uso del puerto USB es más extendido, en entornos industriales sigue siendo muy utilizada.

**RS 485:** Es una norma de comunicación serie ampliamente extendida en la industria, fruto de la colaboración de la Alianza de Industrias electrónicas (*Electronic Industries Alliance - EIA*) y la Asociación de la Industria de Telecomunicaciones (*Telecommunications Industry Association - TIA*) por lo que también se conoce como EIA/TIA-485. Describe las características eléctricas, físicas y tipológicas para conseguir una comunicación series de larga distancia, con alta inmunidad al ruido en redes multipunto a un precio muy económicos.

**SCADA:** *Supervisory Control And Data Acquisition*. en castellano se traduce como Supervisión, Control y Adquisición de Datos. Básicamente es software que se ejecuta en ordenadores en altos niveles de Sistemas de Control Distribuido. Generalmente manipulan un alto volumen de datos y variables a través de complejos sistemas de comunicación que, al ser tratados, gestionan las consignas de un sistema y permiten conocer casi en tiempo real el estado del mismo.

**Variador de Velocidad:** En la bibliografía anglosajona es comúnmente encontrarlo como VSD (*Variable Speed Drive*). A su vez, es un término utilizado para describir equipos que controlan la velocidad de giro de una máquina, independientemente de su principio. Por esta razón, en dependencia del área de estudio al que se haga referencia, se puede estar hablando de equipos diferentes. En este proyecto es utilizado como equivalente de variadores de frecuencia.

**Variador de Frecuencia:** También conocidos como VFD (*Variable Frequency Drive*) o AFD (*Adjustable Frequency Drive*) son equipos que han aprovechado los avances en las industrias de semiconductores y electrónica de potencia, para optimizar las formas de onda de la energía eléctrica que se le suministra a un motor corriente alterna. Entre sus ventajas destaca el ahorro de energía, aumento de la vida útil de los motores y la reducción de los niveles de alteraciones en las líneas eléctricas.