

Tema A1b. Automatización y Control Mecánico: Aplicación de estándares en la automatización.

“Comparación de estándares IEC 61131-3 e IEC 61499 para implementar sistemas de control distribuido”

Ernesto Monroy Cruz^{a,e,*}, Luis Rodolfo García Carrillo^b, Luis Alberto Cruz Salazar^{c,d}, Brayan Daniel Hernández Rojo^e

^aPosgrado CIATEQ A.C. Centro de Tecnología Avanzada Unidad Hidalgo, Av. Diesel Nacional #1, Parque Industrial Ciudad Sahagún, Hgo., México.

^bKlipsch School of Electrical and Computer Engineering, New Mexico State University, 1125 Frenger St, Las Cruces, NM 88001, Estados Unidos.

^cUniversidad Antonio Nariño - Sede Cartagena, Av. Crisanto Luque Diagonal 22 # 48A - 64, Cartagena De Indias, Colombia.

^dInstitute of Automation and Information Systems, Technical University of Munich, Boltzmannstr.15, 85748 Garching cerca a Munich, Alemania.

^eTecnológico Nacional de México Campus Atitalaquía, Av. Tecnológico No. 9, Col. Tezoquiipa, C.P. 42970, Atitalaquía, Hgo., México.

*Autor contacto: ernesto.mz@atitalaquia.tecnm.mx

RESUMEN

Este artículo presenta la comparación de estándares IEC 61131-3 e IEC 61499 en el diseño e implementación de la automatización de un proceso de almacenaje compuesto principalmente por dispositivos electromecánicos, empleando la arquitectura de control centralizado y distribuido a partir de plataformas de programación disponibles para cada estándar. Como parte de la metodología aplicada, se describe desde el planteamiento del caso de estudio, selección de entornos de desarrollo y la generación de los controles hasta la emulación de los mismos, a fin de determinar las diferencias entre los estándares. Como resultado principal, se hace notar que la norma IEC 61131-3 es suficiente y común cuando se requiere implementar un control centralizado en el proceso. Por otro lado, el estándar IEC 61499 permite desarrollar aplicaciones de forma más directa en áreas de automatización industrial donde la arquitectura de control distribuido es un requisito indispensable.

Palabras Clave: Automatización industrial, IEC 61499, IEC 61131-3, Control distribuido, Bloque de función.

ABSTRACT

This paper presents the comparison of the IEC 61131-3 and IEC 61499 standards in the design and implementation of the automation of a storage process composed mainly of electromechanical devices using the centralized and distributed control architecture from the programming platforms available for each standard. As part of the applied methodology, it is described from the approach of the case study, selection of development environments and the generation of controls to their emulation in order to determine the differences between these standards. As a main result, it is noted that the IEC 61131-3 standard is enough and common when it is required to implement a centralized control in the process. On the other hand, the IEC 61499 standard allows to develop applications more directly in areas of industrial automation where distributed control architecture is an essential requirement.

Keywords: Industrial Automation, IEC 61499, IEC 61131-3, Distributed Control, Function Block.

1. Introducción

El diseño de sistemas de automatización y control muestra una tendencia consolidada hacia lo que se ha denominado inteligencia distribuida [1,2]. Esta concepción ha demostrado ser apropiada para responder a las exigencias actuales que se imponen a los sistemas de producción, orientadas a mejorar la agilidad para responder a las perturbaciones del entorno productivo y

para mejorar la reconfigurabilidad de los sistemas de control y de los procesos.

1.1. Trabajos relacionados

De acuerdo a [3] las nuevas tecnologías han permitido el desarrollo de la industria, transformándola así desde la industria tradicional 1.0 hasta la actual, denominada comúnmente Industria 4.0 o Cuarta Revolución Industrial [4,5], que interpreta un rápido crecimiento por

la penetración de los sistemas avanzados y sus diversas tecnologías que prometen brindar sistemas de manufactura cada vez más inteligentes. Actualmente, a pesar de que los controladores basados en PC y los sistemas de control distribuido se pueden utilizar en esta área, el controlador lógico programable (PLC) "todavía se utiliza para el control en casi todas las plantas de fabricación en el mundo" y se considera como uno de los "inventos de automatización más importantes de todos los tiempos" [6], considerándose fundamental en el futuro de la automatización industrial [7]. Ante este escenario, los programas o códigos en el control de procesos o máquinas de la industria se ejecutan comúnmente en PLC's debido a que resultan ser computadoras digitales en las que su hardware está diseñado para tener un alto grado de fiabilidad y disponibilidad en entornos industriales. Por tanto, el software de control es un componente central en los sistemas de automatización actuales y su complejidad aumenta continuamente [8]. La industria ha desarrollado dos estándares diferentes para hacer frente a la generación de software en automatización industrial, el IEC 61131-3 habla sobre los lenguajes de programación [9] para PLC centralizado y cíclico; y el IEC 61499 introduce los bloques de funciones [10] para sistemas de automatización distribuidos controlados por eventos.

Debido a que en la actualidad se pueden utilizar ambos estándares para implementar el código de control de automatización, las ventajas y desventajas de IEC 61499 e IEC 61131-3 con respecto a la reutilización, modelos de ejecución y diseño basado en componentes se han evaluado y discutido [7,11,12]. Comparaciones en particular dan muestra de las propiedades de ambos estándares con relación a la programación orientada a objetos [4], métricas en los esfuerzos de programación [13], ventajas en relación con portabilidad, interoperabilidad, mayor reutilización y distribución, entre otras comparaciones [11].

1.2. Principal contribución

El esfuerzo en la programación y la implementación del control en los procesos de fabricación para obtener industrias cada vez más inteligentes sigue siendo un tema de interés para la comunidad científica. Lo anterior, ha generado en los ingenieros de automatización la necesidad de conocer y evaluar los estándares de control en los ámbitos de manufactura. Por lo tanto, este artículo presenta la comparación de metodologías de implementación del control de un sistema distribuido basado en el estándar IEC-61499 y un sistema de lógica centralizada basado en el estándar IEC-61131-3 programado a partir de la lógica de escalera, aplicados a un proceso de almacenaje automático compuesto por dos estaciones. La finalidad de esta comparación es hacer notar las ventajas y desventajas que tiene aplicar cualquiera de los dos estándares industriales. Así mismo, determinar cuál de ellos es más eficiente al emplearlos en

la automatización de sistemas de manufactura que involucren componentes electromecánicos como principales dispositivos de actuación.

1.3. Estructura del manuscrito

El manuscrito se organiza de la siguiente manera, en la Sección 2 se presentan los estándares IEC 61131-3 e IEC 61499 y plataformas de aplicación selectas. La Sección 3 brinda el desarrollo de aplicaciones para el caso de estudio de automatización basado en ambos estándares. Finalmente, la Sección 4 indica la conclusión y el trabajo a futuro.

2. Estándares y plataformas de implementación

Para estructurar aplicaciones de control, IEC desarrolló el concepto de Bloque de Función (FB, por sus siglas en inglés). Similar a los circuitos integrados en el diseño de circuitos eléctricos, un FB encapsula una cierta funcionalidad y se puede conectar a otros FBs a través de sus entradas y salidas de datos.

2.1. Bloque de funciones en IEC 61131-3

IEC 61131-3 fue el primer paso en la estandarización de los autómatas programables y sus periféricos, incluyendo los lenguajes de programación que se deben utilizar. Al más alto nivel de la estructura de este estándar, ver Figura 1, el software requerido para solucionar un problema de control particular puede ser formulado como una configuración. Una configuración es específica para un tipo de sistema de control, incluyendo las características del hardware: procesadores, direccionamiento de la memoria para los canales de E/S y otras capacidades del sistema. Dentro de una configuración, se pueden definir uno o más recursos. Se puede entender el recurso como un procesador capaz de ejecutar programas IEC.

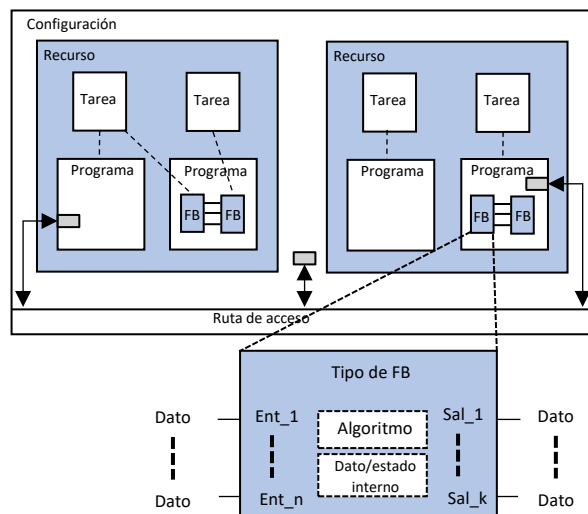


Figura 1 - Representación gráfica del concepto de bloque de función en IEC 61131-3 (adaptada de [14]).

Típicamente, un programa es una interacción de Funciones y Bloques Funcionales, con capacidad para intercambiar datos. En IEC 61131-3 un FB puede contener un solo algoritmo, que puede ser escrito en cualquiera de los lenguajes de programación IEC 61131-3: IL, ST, LD, FBD y SFC, también en otros lenguajes como: C, C++, y Java. Debido a esta característica -de contener solo un algoritmo-, el concepto FB de IEC 61131-3 es a menudo comparado con un módulo en lenguajes de programación secuencial. Sin embargo, no debemos descuidar que un FB encapsula no solo un algoritmo sino también datos, y que puede mantener su estado entre invocaciones. Estas son las características de los lenguajes de programación orientada a objetos, que son más sofisticados en facilitar la reutilización, como se presenta en [4]. Sin embargo, IEC 61499 es mejor respecto a IEC 61131-3 en características de composición simple y múltiple de objetos, y los pointers de destructor/constructor [4].

2.2. Bloque de funciones en IEC 61499

La estructura IEC 61499 está organizada jerárquicamente mediante los modelos de sistema, dispositivo y recurso [10], apoyándose entre ellos y sobre los cuales se establecen los modelos de aplicación y FB. Estos modelos son ilustrados en la Fig. 2.

En la arquitectura del IEC 61499 el principal elemento es también el FB, ver Fig. 2(d), pero dicho concepto ha sido extendido en varios caminos para incorporar nuevos desarrollos desde el dominio de la ingeniería de software. El modelo del sistema mapea/distribuye una o más aplicaciones definiendo qué partes de las aplicaciones se ejecutan en qué dispositivos/recursos particulares. Un modelo de dispositivo/recurso gestiona la conexión a la interfaz de proceso (sensores/actuadores a través del bus de E/S del dispositivo) y la interfaz de comunicación a otros dispositivos utilizados por la aplicación distribuida.

La extensión más llamativa es la Gráfica de Control de Ejecución de eventos (Execution Control Chart, ECC). Un FB en IEC 61499 permanece pasivo hasta que lo active un evento de entrada. El comportamiento dinámico del FB está definido por la ECC que procesa las entradas de eventos y genera salidas de eventos. Así, el FB se ejecuta y produce eventos de salida y datos. La ECC ha sido criticada por hacer más complicadas las aplicaciones en IEC 61499 en comparación con las aplicaciones de IEC 61131-3 debido a las conexiones adicionales entre eventos de FBs. Sin embargo, en contraste con IEC 61131-3, donde el orden de ejecución de FB es determinado implícitamente por la herramienta de desarrollo, la interfaz de eventos en IEC 61499 permite especificación explícita de la secuencia de ejecución de FBs. Esto le da al desarrollador un nuevo nivel de flexibilidad y portabilidad que normalmente no es posible en IEC 61131-3 [5,15].

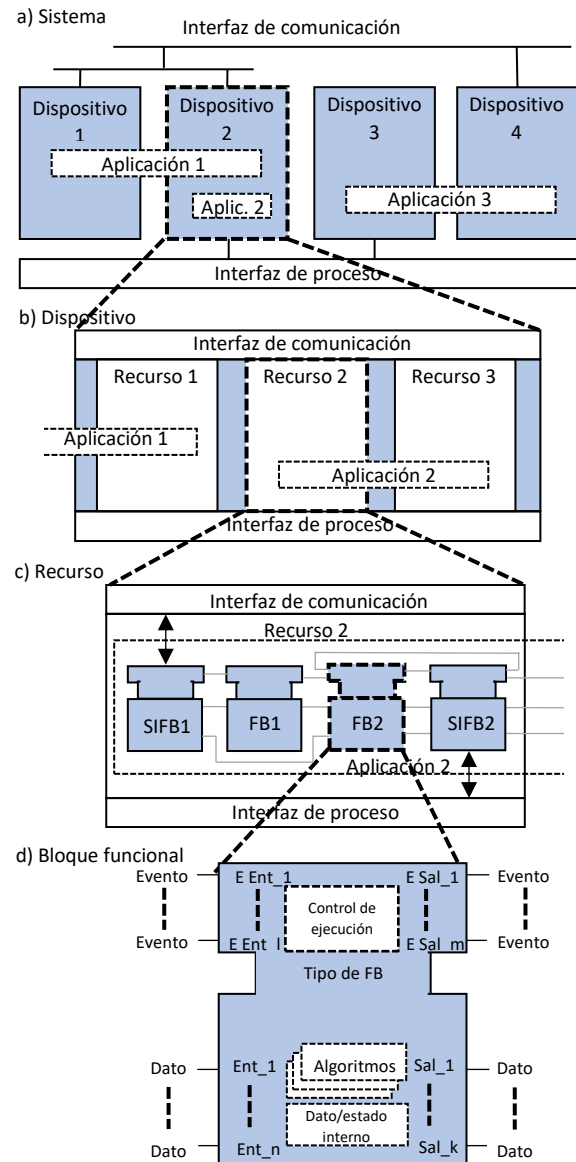


Figura 2 - Representación gráfica de los modelos en IEC 61499 (adaptada de [16]).

2.3. Plataformas de IEC 61131-3 e IEC 61499

Los modelos de FBs antes descritos se pueden llevar a cabo en plataformas disponibles para ambos estándares, comparados en trabajos preliminares en [4]. La Tabla 1 resume las aplicaciones disponibles para el desarrollo de proyectos basados en IEC 61131-3 e IEC-61499, en relación a sus principales características diferenciadoras (ej. tipo de software, protocolos).

La combinación de modelos de aplicaciones, sistemas y dispositivos/recursos permite el diseño de aplicaciones independientemente del hardware de automatización subyacente, la distribución entre dispositivos heterogéneos sin esfuerzo de programación y que esos dispositivos operen entre sí siguiendo modelos de datos

Tabla 1 – Plataformas* disponibles de IEC 61131-3 / IEC 61499.

Plataforma	Norma aplicable	Tipo de Software	Programación basada en	Protocolos
FBDK 2	IEC 61499, IEC 61131-3	Académico	LAD, ST, FBs, ECC, CFB, SIFB	No menciona
4DIAC-IDE	IEC 61499, IEC 61131-3	Académico	ST, FBs, ECC, CFB, SIFB	OPC UA, Modbus, MQTT, openPOWERLINK, Ethernet
nxtSTUDIO	IEC 61499, IEC 61131-3	Industrial	LAD, FBD, ST, SFC, IL, CAT, FBs	EtherCAT, Ethernet / IP, Modbus
ISaGRAF / OpenPCS	IEC 61499, IEC 61131-3	Industrial	LAD, FBD, ST, SFC, IL, CAT, FBs	Ethernet Industrial, Profibus, DeviceNet, Modbus TCP, CANopen, EtherCAT
RSLogix Micro English	IEC 61131-3	Industrial	LAD	Ethernet Industrial, Profibus, Modbus TCP
Simatic Imap	IEC 61499 (parcial)	Industrial	FBs	Profinet, Profibus
TIA Portal (Step 7)	IEC 61131-3	Industrial	LAD, FBD, AWL, SFC, SCL	Ethernet Industrial, Profibus, DeviceNet, Modbus TCP, CANopen, EtherCAT
CoDeSys / TwinCAT	IEC 61131-3	Industrial	LAD, FBD, ST, SFC, IL	Ethernet Industrial, Profibus, DeviceNet, Modbus TCP, CANopen, EtherCAT

*Fuente: Adaptadas de [4]. *Académico* se refiere al tipo de licencia de acceso abierto, opuesta a *Industrial* que por lo general es necesario pagar para su uso.

y comunicaciones estandarizadas en todas las redes (de nuevo sin programación adicional), y menos en la complejidad del programa [13].

La estructura de bloques de funciones basada en eventos de la norma IEC 61499, con tratamiento de entradas, salidas y variables internas locales, coincide estrechamente con la noción de las IT de objetos, métodos y parámetros. Esta fuerte encapsulación es una característica clave de los componentes de software [6]. Así, el estándar IEC 61499 cubre la necesidad de tener un software modular que se pueda utilizar para distribuir el control de procesos industriales y ofrece posibilidades avanzadas y formas de diseñar aplicaciones más descentralizadas de control [2]. Por otro lado, el IEC 61131-3 conserva las propiedades tradicionales de la programación centralizada en PLCs; esto es, los datos de un dispositivo controlador solo pueden ser manipulados por el FB utilizando la interfaz de la unidad de organización del programa, con parámetros y variables externas. No es usual que exista herencia como suele ocurrir en otros lenguajes de programación de alto nivel (por ejemplo, C++ o Java). No obstante, la última edición de IEC 61131-3 agrega funciones para que la programación orientada a objetos sea posible, pero estas características no deben considerarse tan bien como los lenguajes modernos de hoy (diseñados con énfasis de software para IT) con su jerarquía de clases típica [4].

3. Desarrollo de aplicaciones

En este apartado se describen los pasos para las implementaciones en ambos estándares, ver Fig. 3, aplicadas a un caso de estudio, a fin de identificar las principales diferencias entre ellas.

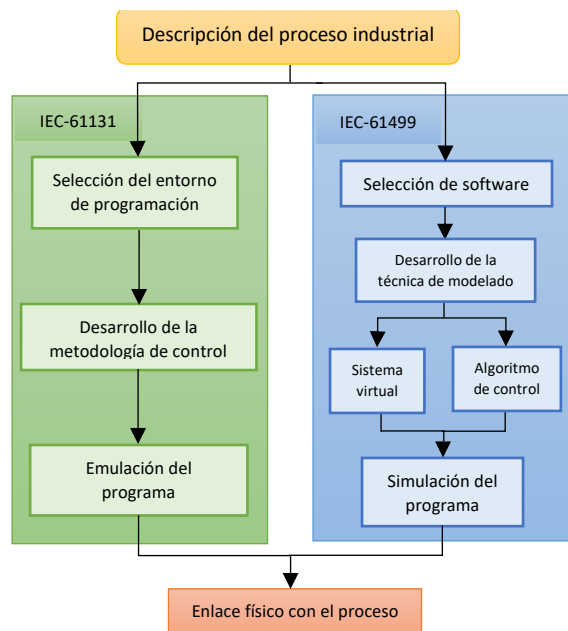


Figura 3 - Etapas para la implementación en cada uno de los estándares.

3.1. Descripción del caso de estudio

Un sistema industrial de almacenamiento momentáneo se ha seleccionado como proceso para la aplicación en la comparación de estándares. El proceso de automatización en almacenes a nivel industrial, es de alto impacto para el diseño de sistemas logísticos. Este efecto se logra solo cuando todos los componentes del sistema son controlados mediante PLC's en arquitectura distribuida. Los equipos de almacenamiento modernos suelen estar unidos por un medio de un transporte versátil intermedio (en este caso, una banda transportadora), ya que de esa manera aumenta la eficiencia de todo el sistema. Así, en caso de producirse un fallo en alguna estación o alguna de ellas ya esté completa al 100%, las demás pueden seguirse llenando, al menos durante el tiempo de operación del abastecimiento.

El sistema de almacenamiento momentáneo, ver Figura 4, se conforma de dos estaciones para almacenar de forma vertical un número máximo de piezas. Cada estación tiene un juego de 3 cilindros neumáticos (dos horizontales de simple efecto y uno vertical de doble efecto) que permiten la colocación y sujeción de las piezas en los contenedores. Cada uno de los pistones tienen 2 sensores -de nomenclatura consecutiva- para indicar su posición, por ejemplo: SC1 para indicar que está retraído y SC2 para detectar cuando está extendido, ambos del cilindro 1 y así sucesivamente. Además, el sistema cuenta con una banda transportadora de las piezas, que tiene montado sensores (SP1 y SP2) para detectar cuando una pieza se encuentra disponible para ser colocada en la estación correspondiente. Los botones

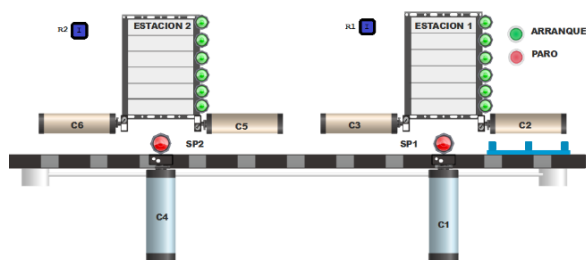


Figura 4 - Estación de almacenamiento momentáneo.

R1 y R2 detectan cuando se extrae alguna pieza almacenada. Finalmente, se tienen dos botones pulsadores, uno de arranque y uno de paro, ambos del sistema en general.

El proceso comienza cuando se da un pulso al botón de arranque (BA), la banda transportadora (M0) desplaza de derecha a izquierda las piezas a colocar en alguna de las estaciones con espacio disponible. Si el contador de almacenaje de alguna de las estaciones refleja que hay disponibilidad para almacenar -teniendo en todo momento prioridad la estación 1 por la cercanía con la llegada de las piezas-, se ejecuta la secuencia de movimiento de los pistones respectivos. Por ejemplo, si fuera la estación 1; cuando la pieza toca el sensor SP1 se desactivan las válvulas de los cilindros C2 y C3 (C2H y C3H) para el retroceso simultáneo de dichos cilindros, que inicialmente están extendidos para sujetar a las piezas que se tengan almacenadas. Una vez retraídos ambos cilindros, se permite la extensión del cilindro C1 (C1H) para elevar la pieza y colocarla en la estación. Cuando la pieza colocada en almacén es detectada por SA1, el cilindro C1 comienza a retroceder (C1L) al mismo tiempo que el indicador visual de nivel de almacenaje se actualiza. Al tener a C1 en su posición inicial (retraído) se vuelven a activar las válvulas de los cilindros C2 y C3 (C2H y C3H) para retener las piezas, quedando disponible nuevamente la estación para colocar otra pieza más cuando sea detectada en la posición respectiva de la banda. Si en determinado momento, ambas estaciones se llenan, el accionamiento de la banda transportadora se bloquea, hasta que se extraiga alguna pieza para liberar espacio para en el almacén.

Tabla 2 – Entradas y salidas del proceso.

Ent. Sist.	Sal. Sist.	Ent. Est. 1	Sal. Est. 1	Ent. Est. 2	Sal. Est. 2
BA	M0	SP1	C1H	SP2	C4H
BP		SA1	C1L	SA2	C4L
		SC1	C2H	SC7	C5H
		SC2	C3H	SC8	C6H
		SC3		SC9	
		SC4		SC10	
		SC5		SC11	
		SC6		SC12	
		R1		R2	

En cualquier momento que se presione el botón de paro (BP), el sistema de detiene por completo y continúa hasta que se vuelva a dar un pulso de marcha. En la Tabla 2 se muestran las entradas y salidas que se establecieron del sistema completo.

3.2. Implementación en IEC 61131-3

Para iniciar la implementación con el estándar IEC 61131-3 es necesario definir la plataforma de programación. En este punto, se decidió utilizar el software RS-Logix Micro English de Rockwell Automation. Este entorno de programación, además de contar con un sistema de comunicación serial con el procesador central de un emulador, cuenta con la comunicación del PLC físico, en este trabajo se consideró utilizar un Micrologix 1000 Analog de Allen Bradley. También, el archivo de datos del software RS-Logix Micro English, establece por default archivos de datos de la memoria del controlador, que sirven para ser direccionados a los elementos funcionales en los ramales de la estructura del programa.

Los algoritmos de control de los sistemas flexibles de manufactura se basan en el empleo de PLCs. Estos dispositivos realizan un control centralizado y cíclico consistente en: adquisición del estado de los sensores, ejecución de los algoritmos de control y la actualización del estado de los actuadores. Para la aplicación del control con el estándar IEC 61131-3 en el caso de estudio se ha elegido la programación en diagrama de escalera. Esta programación es similar al diagrama eléctrico y tiene como base principal el principio de funcionamiento de relés electromecánicos funcionando al mismo tiempo. Debido a que el sistema descrito de almacenaje automático implica como principales actuadores a cilindros neumáticos, la metodología de cascada resulta útil y simple para generar gran parte del diagrama de escalera. Este método permite agrupar la secuencia definida de cada estación bajo las condiciones de operación antes mencionadas. Por ejemplo, cuando se activa la secuencia en la primera estación sucede el siguiente orden de activaciones (+) y desactivaciones (-) de los cilindros neumáticos:

$$(C2-, C3 -), C1+, C1-, (C2+, C3 +)$$

Siguiendo los pasos de la metodología de cascada se obtienen y transforman las ecuaciones de los grupos y las electroválvulas, a conexiones de contactos y bobinas tanto auxiliares como de salida en el diagrama de escalera. Dentro del programa se realiza el direccionamiento de los archivos de datos del procesador, estableciendo las entradas de los sensores del proceso, así como las salidas contenidas para los actuadores dentro del sistema. De igual manera, se configuran las funciones complementarias, como los contadores que indican el número de piezas contenidas en cada estación. Los sensores SA1 y SA2 están conectados como entradas a

las funciones de conteo de las estaciones 1 y 2 respectivamente, con la finalidad de detectar cuando una nueva pieza se ha almacenado en la estación. En la Fig. 5 se observa una parte del diagrama de escalera, donde están asignadas algunas etiquetas de electroválvulas para el accionamiento de los pistones involucrados en el proceso. Así, todas las entradas, salidas, bobinas auxiliares y contadores son direccionados con los archivos de memoria del PLC.

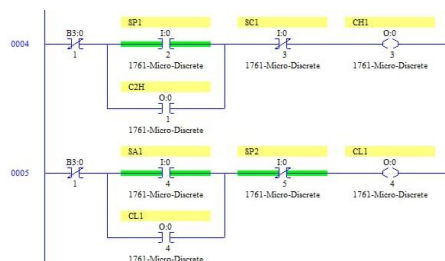


Figura 5 – Segmento de la estructura secuencial del programa Ladder.

Posteriormente, se procede a realizar la configuración de la conexión del programa con el emulador. Para ello, se establece la comunicación de RS-Micro Logix English con el driver del Emulador (RS Logix Emulate). La configuración de este driver se lleva a cabo mediante el programa RS-Linx Classic. La relación que existe entre estos tres programas para poder verificar el funcionamiento del diagrama de escalera se visualiza en la Fig. 6. En consecuencia, se logra el enlace de comunicación serial para emular la conexión física del PLC Micrologix 1000 Analog.

Una vez configurada la conexión, es posible descargar y validar el funcionamiento del código. La verificación se lleva a cabo mediante el forzamiento de las señales de entrada (como los sensores y botones) directamente en la estructura del programa, para así poder visualizar la activación de las salidas (motor de banda y solenoides de

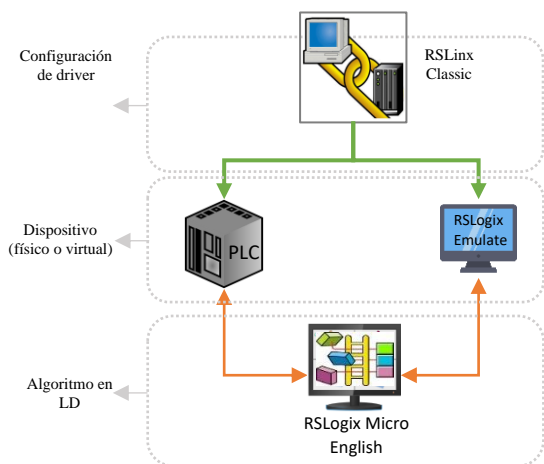


Figura 6 – Relación de software para la configuración de drivers y generación de diagrama de escalera.

las electroválvulas). Por ejemplo, en la Fig. 7 se observa un instante de una parte del programa ejecutándose. Se aprecia el contacto NA asociado al sensor SA1 envía señal al conteo (ascendente) de piezas en la estación 1. Para efectos de simulación de la extracción de una pieza de cierto almacén con el estándar IEC 61131-3, se agregan también a R1 o R2 como señal de entrada en el conteo (descendente) de cada estación respectiva, de esta forma es posible actualizar el valor de los contadores que reflejan la disponibilidad de los almacenes.

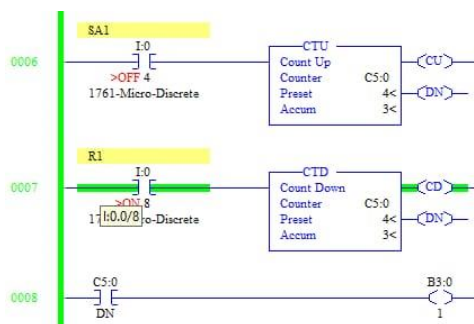


Figura 7 – Ejecución del programa descargado al emulador del PLC.

3.3. Implementación en IEC 61499

Para el desarrollo de la solución en IEC 61499 se decidió utilizar el software nextSTUDIO [17] de Schneider Electric. Esta es una plataforma de programación caracterizada por contar con el lenguaje de programación de bloques funcionales FB, siendo un lenguaje aplicable por el estándar IEC 61499 para el desarrollo del control de procesos. El software nextSTUDIO integra la ingeniería de lógica de control, visualización, conexión de E/S, pruebas, simulación y documentación en una sola herramienta de desarrollo. Por tanto, es posible realizar el diseño y la implementación de un sistema de control multinivel distribuido que permite la ejecución de nuevas estrategias en el control de procesos, siendo esto útil en nuestro caso de estudio. La arquitectura distribuida contempla el uso de dos controladores DCSmini de Schneider Electric, uno para cada estación.

IEC 61499 es visto como la siguiente generación de estándares en sistemas de automatización. Por el momento, en la práctica industrial son pocos los sistemas basados en IEC 61499, una de las razones es la ambigüedad semántica que lleva a diferentes interpretaciones a la hora de su implementación [7], sin embargo, actualmente una gran cantidad de trabajos de investigación utilizan los conceptos básicos del estándar [18]. Para obtener el control mediante la norma IEC 61499 se han definido dos fases en el proceso de diseño: la primera consta en generar el modelo virtual del proceso a controlar, en este caso, el sistema de almacenaje, y la segunda fase trata sobre el desarrollo del control descentralizado de las estaciones.

a) *Sistema Virtual*: El diseño del sistema virtual se crea utilizando un proceso de agregación de funciones, se va construyendo cada capa, hasta llegar al sistema completo. Es decir, cada dispositivo físico (cilindro, sensor, botón) se puede modelar con uno o varios FBs y los flujos de información entre ellos vienen definidos por las conexiones de datos. Con el estándar IEC 61499, la simulación de extracción de una pieza de cierto almacén es contemplada internamente en el modelo virtual del sistema, por lo que no refleja una conexión física hacia los controles de las estaciones. Solo se actualiza el valor acumulado en el contador de piezas y así se determina la disponibilidad de los almacenes. En la Fig. 8 se visualiza el FB que encapsula la secuencia de otros bloques funcionales, es decir, el modelo de aplicación o CAT del sistema completo de almacenamiento obtenido en nxtSTUDIO. En el FB se aprecian las señales de entrada y salida del caso de estudio. Este bloque funcional está conformado por modelos básicos, compuestos y de aplicación. Primordialmente, se encuentran aquellos FBs relacionados con los componentes del proceso de almacenaje (cilindros neumáticos, sensores, interruptores, indicadores y objetos a almacenar). Por otra parte, en la Fig. 9 se aprecia el modelo virtual resultante, el cual representa la descripción del sistema de almacenamiento momentáneo de nuestra aplicación. Se pueden visualizar las dos estaciones en conjunto con sus tres pistones respectivos, así como la banda transportadora y los sensores del proceso.

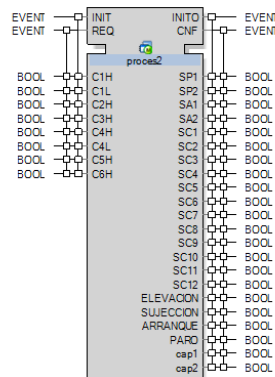


Figura 8 - Bloque funcional de encapsulamiento de la aplicación de la estación.

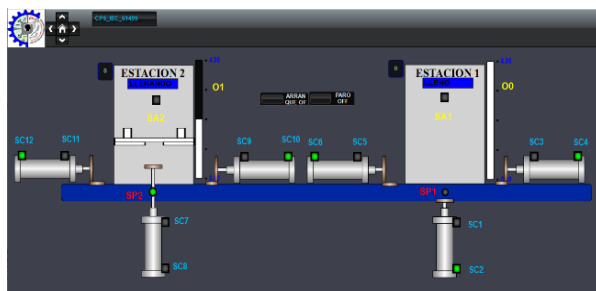


Figura 9 – Modelo virtual en nxtSTUDIO del sistema de almacenaje.

b) *Control descentralizado*: En esta etapa, se han generado dos bloques para el control descentralizado, uno por cada estación. De acuerdo a la norma IEC 61499, este bloque funcional es asignado como un modelo básico. En él, se definen los eventos y variables que serán ejecutadas conforme a la estructuración del ECC. En la Fig. 10 se visualiza el FB para el control de la estación 1. El de la estación 2 es muy similar, solo difiere en la definición de las entradas y salidas.

El ECC contenido en los FBs de cada cilindro representa el algoritmo para el control descentralizado, ver Figura 11. En este caso, el ECC está conformado por 6 estados y 5 algoritmos. La secuencia de operación de las electroválvulas en cada estación, se ejecuta dependiendo de las transiciones asignadas. Cada que se cumple alguna condición, se ejecuta el siguiente estado. Así, se realiza el algoritmo interno que contiene cada etapa. Cuando ya se ha ejecutado el algoritmo, el evento de salida de la interfaz se activa para validar que el bloque secuencial se ha ejecutado correctamente. De esta forma, se logra el control de movimiento de los cilindros de acuerdo a los requisitos de operación de cada estación.

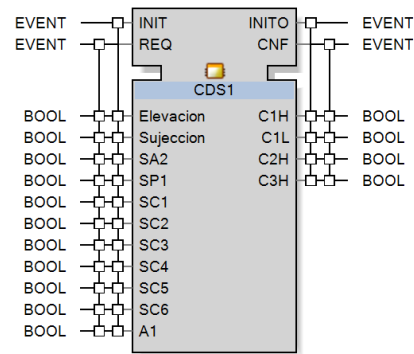


Figura 10 - Bloque funcional básico del control descentralizado de la primera estación.

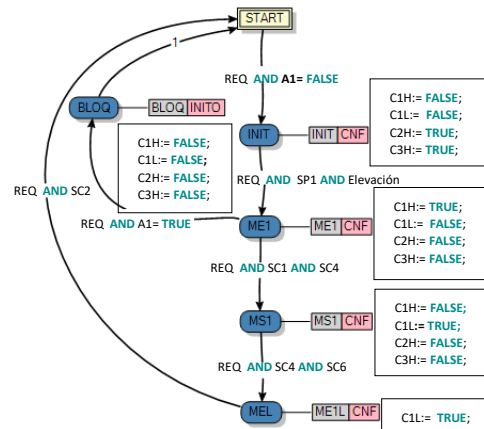


Figura 11 – ECC de la estación 1.

Una vez desarrollado el FB del modelo virtual y los FBs de los controles para cada estación, se procede a enlazarlos mediante las conexiones de entradas y salidas. Posteriormente, se les asigna un mapeo de dispositivo de control a cada uno de los bloques funcionales dentro de la aplicación general. Así, es creado el acceso a una red de prueba local enlazando cada uno de los dispositivos mediante una IP individual y ejecutados mediante tres RunTimes de nxtSTUDIO en una PC; uno para el modelo virtual del sistema presentado en la Sección 3.1 -que representa el proceso real y es visualizado en la HMI- y dos para los controladores DCSmini de las estaciones de almacenaje, ver Fig. 12. Por lo tanto, cada estación del sistema está diseñado para ser controlado por un PLC individual e identificable por su puerto ID único. El segmento Ethernet modela el canal de comunicación para todas las instancias de dispositivos en nxtSTUDIO. En la Fig. 13 se muestra la relación de los FBs y el estado de sus variables en un instante de ejecución del programa al validar el funcionamiento del código desarrollado.

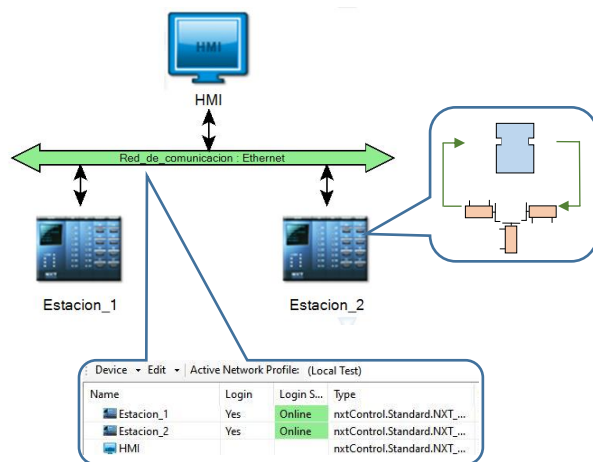


Figura 12 – Enlace de la red de comunicación Ethernet entre los dispositivos de control y el modelo de sistema en prueba local.

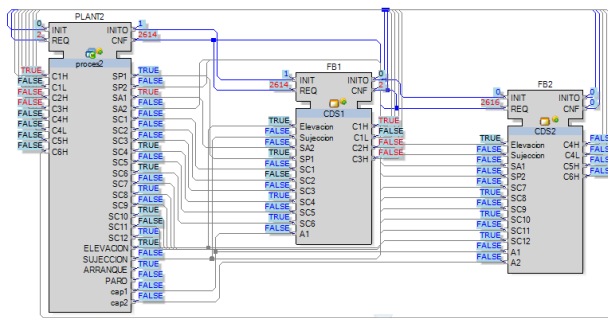


Figura 13 – Conexión y ejecución de los FBs de la aplicación principal en control distribuido.

3.4. Enlace físico con el proceso

Una vez que los algoritmos de control han sido validados mediante simulación, la conexión hacia los puertos físicos tanto en IEC 61131-3 para el Micrologix 1000 Analog como en IEC 61499 para los controladores DCSmini de Schneider Electric es relativamente simple. En el primer caso, se requiere reconfigurar el driver de Emulador a RS232 y establecer los parámetros de comunicación del puerto COM del PLC físico con la computadora, todo esto en el entorno de RS Linx Classic. En el otro caso, el software nxtSTUDIO permite desarrollar el enlace físico para crear una topología y comunicación de dispositivos de red con la aplicación. La conexión física entre los dispositivos de control se permite gracias a las bibliotecas del software que permiten el direccionamiento de las señales de entrada y de salida.

3.5. Análisis de implementaciones

Al realizar la programación del proceso de almacenaje aplicando distintos estándares y validar el funcionamiento a través de entornos de simulación, se hacen notar los siguientes puntos:

- El enfoque centralizado y la rigidez del estándar IEC 61131-3 hace que las estructuras de control impliquen una respuesta limitada a la distribución del control que hoy exige la Industria 4.0 y sus características, tales como: el Plug & Produce, flexibilidad, agilidad, programación orientada a objetos entre otras introducidas en [4,5]. Respecto al segundo enfoque, el estándar IEC 61499 ofrece la forma distribuida de diseñar las aplicaciones de control en la automatización industrial debido a que se adaptan mejor a las soluciones de modularidad. Ver Sec. 3.2 y Sec. 3.3.
- El fallo del único hardware de control en IEC 61131-3 representa el paro completo de todo el sistema. Como ventaja de la implementación con IEC 61499, se observa que, debido a la distribución de la aplicación, en caso de que algún hardware de control falle, no implica el paro total del sistema sino solo de la parte asignada al dispositivo dañado, incrementando la disponibilidad de los recursos. Ver Sec. 3.1.
- En términos de adaptabilidad, en IEC 61131-3 tan pronto como se necesite agregar otra estación de almacenaje en el proceso, se debe adaptar/modificar el diagrama de escalera de acuerdo a las condiciones de operación del nuevo elemento. Entonces, a medida que el tamaño del sistema a controlar crece, la complejidad y el tiempo de desarrollo también aumentan. Por otro lado, con IEC 61499 basado en el principio de encapsulación, cada módulo protege sus algoritmos internos y variables de los otros componentes del sistema de control, haciendo que tanto el desarrollo como la modificación de software

sean más fáciles. Por lo tanto, al tener la necesidad de agregar una nueva estación, existe la posibilidad de generar los nuevos componentes a partir de los ya creados, al reutilizar sus funcionalidades del software. Ver Sec. 3.3.

- Respecto a los softwares utilizados en esta aplicación, aunque el desarrollo en IEC 61499 pueda parecer más complicado en comparación con IEC 61131-3 debido a las conexiones adicionales que resultan de los eventos en los bloques funcionales, estas conexiones permiten especificar de forma más clara y directa el orden de ejecución de los FBs, a diferencia de la secuencia de ejecución en IEC 61131-3 limitada por la herramienta de software. Lo que resulta en contar con mayor flexibilidad en el desarrollo de control con IEC 61499 en comparación con IEC 61131-3.

4. Conclusiones y trabajo a futuro

Aplicar las plataformas para el desarrollo de control distribuido de procesos bajo los estándares industriales IEC 61131-3 e IEC 61499 permite incorporar estrategias nuevas en la ingeniería de la automatización que garanticen la robustez, viabilidad y eficiencia para el ciclo de mejora continua dentro de los sistemas de manufactura actuales, ej. programación orientada a objetos de tipo industrial.

Cabe mencionar que el estándar IEC 61131-3 ha sido ampliamente aceptado en el dominio de la automatización industrial. Sin embargo, se afirma que esta norma no aborda hoy en día los nuevos requisitos de los sistemas industriales complejos entorno a la Industria 4.0, que incluyen, portabilidad, interoperabilidad, mayor reutilización y distribución. Para hacer frente a estas restricciones, la norma IEC 61499 ha mostrado ser una tecnología madura para permitir la automatización inteligente.

Según se observa, es necesario aumentar el diseño de sistemas de control distribuido basados en el estándar IEC 61499, de manera que se demuestre las ventajas con mejor precisión y aumente su aceptación a nivel industrial, respecto a los desarrollados en IEC 61131-3. Lo anterior, dado que este último estándar suele estar frecuentemente integrado en la mayoría de programación de PLCs comerciales y otros dispositivos industriales, mientras que IEC 61499 tiene pocos protocolos y mejores opciones de plataformas de acceso abierto.

Finalmente, los procesos automatizados actuales requieren de cierto nivel de inteligencia artificial para no depender de un control central que generalice por completo su funcionamiento, en ese sentido, la metodología propuesta en este trabajo respecto al control distribuido puede aplicarse al desarrollo de sistemas de control de tipo industrial descentralizado y basado en arquitecturas inteligentes, ej. agentes industriales, considerado esto como uno de los principales trabajos a futuro en la presente investigación.

Agradecimientos

El autor Ernesto Monroy agradece al Consejo Nacional de Ciencia y Tecnología (CONACYT) por la beca otorgada para estudios de posgrado en el Centro de Tecnología Avanzada (CIATEQ) y el apoyo del Tecnológico Nacional de México (TecNM) Campus Atitalaquia. El autor Luis Cruz agradece las becas otorgadas en Colombia, por Minciencias convocatoria “756 Doctorados en el exterior” y a la Universidad Antonio Nariño bajo el “Programa de Formación de Alto Nivel PFAN”.

REFERENCIAS

- [1] G. Zapata M., E. Chacón R., and H. Flórez C., “Programación de PLC’s mediante componentes inteligentes distribuidos, para apoyar el cambio de paradigma en el diseño de estructuras de automatización,” in IIII (Tercera) Jornada de automatización de la industria petrolera JAIP, 2012, vol. 1a.edición, pp. 479–490.
- [2] V. Vyatkin, IEC 61499 Function Blocks for Embedded and Distributed Control Systems Design, Third Edition, Third Edit. International Society of Automation, 2016.
- [3] C. Valdiviezo-Abad and T. Bonini, “Intelligent automation in communication management,” *Doxa Comunicación. Revista interdisciplinaria de estudios de comunicación y ciencias sociales*, no. 29, pp. 169–196, Dec. 2019, doi: 10.31921/doxacom.n29a9.
- [4] L. A. Cruz S. and B. Vogel-Heuser, “Applying Core Features of the Object-Oriented Programming Paradigm by Function Blocks Based on the IEC 61131 and IEC 61499 Industrial Automation Norms,” in *Service Oriented, Holonic and Multi-agent Manufacturing Systems for Industry of the Future*, 1st ed., T. Borangiu, D. Trentesaux, P. Leitao, A. Giret Boggino, and V. Botti Navarro, Eds. Cham: Springer International Publishing, 2020, pp. 273–289. doi: 10.1007/978-3-030-27477-1_21.
- [5] A. Homay, A. Zoitl, M. de Sousa, and M. Wollschlaeger, “A Survey: Microservices Architecture in Advanced Manufacturing Systems,” in *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, Jul. 2019, pp. 1165–1168. doi: 10.1109/INDIN41052.2019.8972079.
- [6] K. Thramboulidis, “IEC 61499 as an Enabler of Distributed and Intelligent Automation: A State-of-the-Art Review—A Different View,” *Journal of Engineering*, vol. 2013, pp. 1–9, 2013, doi: 10.1155/2013/638521.
- [7] L. A. Cruz S. and O. A. Rojas A., “The future of industrial automation and IEC 61499 standard,” in *III International Congress of Engineering Mechatronics and Automation, CIIMA*, Oct. 2014, pp. 1–5. doi: 10.1109/ciima.2014.6983434.

- [8] W. ElMaraghy, H. ElMaraghy, T. Tomiyama, and L. Monostori, "Complexity in engineering design and manufacturing," *CIRP Annals*, vol. 61, no. 2, pp. 793–814, 2012, doi: 10.1016/j.cirp.2012.05.001.
- [9] IEC, "IEC 61131 – Programmable controllers, Part 3: Programming languages. International Electrotechnical Commission," 2013.
- [10] IEC, "IEC 61499: Function blocks for industrial-process measurement and control systems." 2005.
- [11] K. Thramboulidis, "IEC 61499 vs. 61131: A Comparison Based on Misperceptions," *Journal of Software Engineering and Applications*, vol. 06, no. 08, pp. 405–415, 2013, doi: 10.4236/jsea.2013.68050.
- [12] E. Monroy Cruz, L. R. García Carrillo, S. Patil, and N. C. Hellman, "Adaptability in IEC 61499-Based distributed control for automation applications," in 2022 IEEE 9th International Conference on Industrial Engineering and Applications (ICIEA), Apr. 2022.
- [13] P. Gsellmann, M. Melik-Merkumians, and G. Schitter, "Comparison of Code Measures of IEC 61131–3 and 61499 Standards for Typical Automation Applications," in 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), Sep. 2018, pp. 1047–1050. doi: 10.1109/ETFA.2018.8502464.
- [14] S. Cavalieri and M. G. Salafia, "Asset Administration Shell for PLC Representation Based on IEC 61131-3," *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.3013890.
- [15] A. Hopsu, U. D. Atmojo, and V. Vyatkin, "On Portability of IEC 61499 Compliant Structures and Systems," in 2019 IEEE 28th International Symposium on Industrial Electronics (ISIE), Jun. 2019, pp. 1306–1311. doi: 10.1109/ISIE.2019.8781290.
- [16] M. v. García, E. Irisarri, F. Pérez, E. Estévez, and M. Marcos, "Arquitectura de Automatización basada en Sistemas Ciberfísicos para la Fabricación Flexible en la Industria de Petróleo y Gas," *Revista Iberoamericana de Automática e Informática industrial*, vol. 15, no. 2, p. 156, Mar. 2018, doi: 10.4995/riai.2017.8823.
- [17] nxtControl, "Schneider Electric - nxtSTUDIO," 2007. <http://www.nxtcontrol.com/en/engineering/> (accessed May 14, 2022).
- [18] G. Lyu and R. W. Brennan, "Towards IEC 61499-Based Distributed Intelligent Automation: A Literature Review," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 4, pp. 2295–2306, Apr. 2021, doi: 10.1109/TII.2020.3016990.