

Propuesta de Aplicación del estándar IEC 61499-1 en un caso de estudio



**Jorge Alberto Betancourt Muñoz
Cristian David Sánchez Cobo**

Universidad del Cauca

**Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Electrónica Instrumentación y Control
Línea de Investigación Automatización
Popayán, Julio del 2016**

Propuesta de Aplicación del estándar IEC 61499-1 en un caso de estudio



Documento Final de Trabajo de Grado para optar al
Título de:
Ingeniero en Automática Industrial

Jorge Alberto Betancourt Muñoz
Cristian David Sánchez Cobo

Director: MG. Ermilso Díaz Benachí
Codirector: Esp Vladimir Trujillo Arias

Universidad del Cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Electrónica Instrumentación y Control
Línea de Investigación Automatización
Popayán, Julio del 2016

Nota de aceptación

Director

Jurado

Jurado

Popayán, Julio de 2016.

Resumen

En la actualidad la Automatización de plantas y máquinas es cada vez más compleja debido a la utilización de dispositivos de campo inteligentes o controladores de diferente proveedor en donde el estándar actualmente empleado ¹IEC-61131 no supe las necesidades de los nuevos sistemas de control. El diseño y control de sistemas de automatización muestra una tendencia consolidada hacia lo que se ha llamado control distribuido, este sistema ha demostrado ser adecuado para satisfacer las actuales necesidades en las técnicas de producción, técnica que está destinada a optimizar el entorno productivo.

Los inconvenientes a los cuales se enfrenta la automatización industrial, son el resultado de los requisitos cada vez más estrictos del sector productivo, dentro de los que se encuentran; el cumplimiento de las normas de calidad, la necesidad de generar productos a bajo costo a partir de una reingeniería o reconfiguración de los recursos presentes, la eficiencia en el manejo de los tiempos y de los recursos utilizados, la productividad y efectividad con que se deben crear los sistemas de control, la necesidad de descentralizar los técnicas de control que se han adquirido.

Se busca adquirir nuevas tecnologías para el desarrollo de los procesos de manufactura dentro de las industrias para obtener las mejores características en los modelos de programación y uso de estándares industriales que surgen con el tiempo y que son aplicables y de gran utilidad a los procesos, por lo cual se crea el estándar IEC-61499, publica su primera edición en el año 2005 cumpliendo un papel importante dentro del control distribuido, mostrando una serie de nuevas mejoras específicas hacia los sistemas de producción y la forma de operarlos, de esta manera se puntualizan cinco progresos respecto a estándares previos; *portabilidad, interoperabilidad, configurabilidad, reconfigurabilidad y distribución.*

Debido a que el estándar aporta modelos y arquitectura, y no una metodología inicial de cómo aplicarlo, su aceptación a nivel industrial es baja. El objetivo del presente trabajo es aportar una propuesta que le permita al estándar aumentar las tasas de adopción. Se presentara un estudio detallado del estándar IEC-61499 en aplicaciones distribuidas y se propone un procedimiento para la identificación de elementos que compone la arquitectura, además se generara una propuesta de aplicación del procedimiento, para el desarrollo de un prototipo en uno de los laboratorios de la Universidad.

¹ IEC: International Electrotechnical Commission

TABLA DE CONTENIDO

INTRODUCCIÓN	12
1. CONCEPTOS FUNDAMENTALES.....	13
1.1. Control de procesos.....	13
1.1.1. Estructura general de los sistemas de control industrial.	14
1.2. Sistema SCADA (sistema de control, supervisión y de adquisición de datos).....	15
1.2.1. Funciones principales de un sistema SCADA.	15
1.2.2. Elementos de un sistema SCADA.....	16
1.2.3. Programación de los sistemas de control en sistemas SCADA	17
1.2.4. Ventajas y desventajas de un sistema SCADA.....	18
1.3. Sistemas de control Distribuido DCS.....	20
1.3.1. Generalidades de un Sistema de Control Distribuido.....	21
1.3.2. Componentes funcionales de un DCS.	22
1.3.3. Ventajas de un DCS.....	24
1.3.4. Desventajas de un DCS.....	24
1.4. Aspectos Comparativos entre un SCADA y un DCS	24
1.5. Necesidades de los sistemas de control distribuido.	25
1.6. Estándar IEC 61499	28
2. CONCEPTOS Y MODELOS DEL IEC 61499.....	30
2.1. Arquitectura propuesta en IEC 61499.....	30
2.1.1. Modelo de bloque de función básico	31
2.1.2. Bloques de función compuestos	36
2.1.3. Bloque de función de interfaz de servicio	36
2.1.4. Modelo de aplicación.....	41
2.1.5. Modelo de recurso.....	42
2.1.6. Modelo de subaplicación.....	43
2.1.7. Modelo de dispositivo.....	43
2.1.8. Modelo del sistema	44
2.1.9. Bloques de función de evento	45
2.1.10. Ejemplo modelado de sistema de control de nivel.	48

3. PROCEDIMIENTO PARA LA IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL DISTRIBUIDO BAJO EL ESTÁNDAR IEC 61499-1.....	51
3.1. Procedimiento para la identificación de los elementos necesarios en una aplicación de control distribuido con IEC 61499.....	51
3.1.1. Identificación de un proceso distribuido.....	52
3.1.2. Elaboración del diagrama P&ID.....	54
3.1.3. Establecer la arquitectura de control distribuido para el proceso de producción.....	56
3.1.4. Modelado con IEC 61499 según el estándar.....	57
4. APLICACIÓN CONCEPTUAL DEL PROCEDIMIENTO DESARROLLADO PARA UN PROTOTIPO EN EL LABORATORIO DEL PIAI.....	65
4.1. Estado de las plantas del laboratorio del PIAI.....	65
4.1.1. Planta Multivariable.....	65
4.1.2. Planta Sistema a Eventos Discretos.....	67
4.1.3. Planta de Temperatura.....	69
4.2. IDENTIFICACIÓN DE UN PROCESO DISTRIBUIDO.....	71
4.3. Diagrama P&ID y Tabla de instrumentos del proceso.....	73
4.3.1. Lista de variables.....	75
4.3.2. Lista de equipos.....	76
4.4. Arquitectura del sistema propuesto.....	76
4.5. Modelado con IEC-61499.....	79
4.5.1. Identificación de dispositivos.....	79
4.5.2. Secuencia de operaciones.....	79
4.5.3. Funcionalidad de dispositivos.....	83
4.5.4. Entornos de Desarrollo y de Ejecución de la Norma IEC-61499.....	84
4.5.5. HERRAMIENTAS HARDWARE COMPATIBLES CON IEC 61499.....	86
4.5.6. Identificación de los Bloques de Función.....	87
4.5.7. Aplicación de control.....	100
4.5.8. MODELO DE RECURSO.....	103
4.5.9. CONFIGURACION DE COMUNICACION.....	104
4.5.10. Modelo de dispositivo/Sistema.....	111
4.5.11. Lista de Chequeo.....	114
5. CONCLUSIONES.....	116
6. BIBLIOGRÁFICA.....	117

INDICE DE FIGURAS

FIGURA 1. PIRÁMIDE CIM.....	14
FIGURA 2. ESQUEMA DE LOS ELEMENTOS DE UN SISTEMA SCADA	17
FIGURA 3. ARQUITECTURA DE UN SISTEMA SCADA	19
FIGURA 4. SIMATIC PCS 7 DE SIEMENS.....	21
FIGURA 5. CONEXIONES EN UN SISTEMA DE CONTROL DISTRIBUIDO.....	22
FIGURA 6. COMPONENTES FUNCIONALES DE UN DCS.....	24
FIGURA 7. RED DE BLOQUES DE FUNCIÓN IEC 61131-3	26
FIGURA 8. RED DE BLOQUES DE FUNCIÓN REALIMENTADOS CON IEC 61131-3	27
FIGURA 9. RED DE BLOQUES DE FUNCIÓN REALIMENTADOS CON IEC 61131-3	28
FIGURA 10. MODELO DE BLOQUE DE FUNCIÓN DEFINIDO POR IEC 61499.....	32
FIGURA 11. GRÁFICO DE CONTROL DE EJECUCIÓN, DEFINIDO POR IEC 61499.....	34
FIGURA 12. MODELO DE BLOQUE DE FUNCIÓN COMPUESTO DEFINIDO POR IEC 61131...35	35
FIGURA 13. MODELO DE BLOQUE DE FUNCIÓN COMPUESTO DEFINIDO POR IEC 61499...35	35
FIGURA 14. MODELO DE BLOQUE DE FUNCIÓN COMPUESTO DEFINIDO POR IEC 61499...36	36
FIGURA 15. BLOQUE IO_WRITER.....	37
FIGURA 16. BLOQUE IO_READER.....	38
FIGURA 17. BLOQUE PUBLISH.....	39
FIGURA 18. BLOQUE SUSBCRIBE.....	40
FIGURA 19. COMUNICACIÓN DE LOS BLOQUES PUBLISH-SUBSCRIBE.....	41
FIGURA 20. MODELO DE APLICACIÓN DEFINIDO POR IEC 61499.....	42
FIGURA 21. MODELO DE RECURSO DEFINIDO POR IEC 61499.....	43
FIGURA 22. MODELO DE SUBAPLICACIÓN DEFINIDO POR IEC 61499	43
FIGURA 23. MODELO DE DISPOSITIVO DEFINIDO POR IEC 61499	44
FIGURA 24. MODELO DE SISTEMA DEFINIDO POR IEC 61499.....	45
FIGURA 25. EVENTO SPLIT.....	45
FIGURA 26. EVENTO E_MERGE.....	46
FIGURA 27 EVENTO E_PERMIT	46
FIGURA 28. EVENTO E_RESTART.....	47
FIGURA 29. EVENTO E_CYCLE	47
FIGURA 30. DIAGRAMA DE CONTROL DE EJECUCIÓN BLOQUE PID	49
FIGURA 31. EJEMPLO DE CONTROL DE NIVEL CON EL MODELADO DE IEC 61499	50
FIGURA 32. PROCEDIMIENTO PARA LA IDENTIFICACIÓN DE LOS ELEMENTOS.....	51
FIGURA 33. MODELO FÍSICO DE ISA-S88.01	53
FIGURA 34. MODELO FÍSICO ISA-S88.01. CÉLULA.....	54
FIGURA 35. DIAGRAMA P&ID PAPEL TISSUE	55
FIGURA 36. ELEMENTOS PRINCIPALES DE UNA ARQUITECTURA DISTRIBUIDA	56
FIGURA 37. MODELAMIENTO CON IEC 61499	57
FIGURA 38. SISTEMA DE MANUFACTURA FLEXIBLE	58
FIGURA 39. RED DE PETRI CORRESPONDIENTE A LOS ESTADOS DEL SISTEMA DE MANUFACTURA FLEXIBLE	58
FIGURA 40. IDENTIFICACIÓN DEL BLOQUE IO_READER.....	60

FIGURA 41. IDENTIFICACIÓN DEL BLOQUE IO_ WRITER.....	60
FIGURA 42. IDENTIFICACIÓN DEL BLOQUE SUSCRIBE	61
FIGURA 43. IDENTIFICACIÓN DEL BLOQUE PUBLISH	62
FIGURA 44. ELEMENTOS QUE COMPONEN EL MODELO DE APLICACIÓN DEFINIDO POR IEC 61499.....	63
FIGURA 45. MODELO DE SISTEMA, DISPOSITIVO, RECURSO DEFINIDO POR IEC 61499....	64
FIGURA 46. DIAGRAMA PI&D PLANTA MULTIVARIABLE	66
FIGURA 47. PLANTA MULTIVARIABLE ELEMENTOS DE CAMPO	67
FIGURA 48. PLANTA MULTIVARIABLE ELEMENTOS DE CONTROL	67
FIGURA 49. DIAGRAMA PI&D PLANTA SED.	68
FIGURA 50. PLANTA SED ELEMENTOS DE CAMPO	68
FIGURA 51. PLANTA SED ELEMENTOS DE CONTROL	69
FIGURA 52. DIAGRAMA PI&D PLANTA DE TEMPERATURA.	70
FIGURA 53. PLANTA DE TEMPERATURA ESTRUCTURA FÍSICA.	70
FIGURA 54. PLANTA DE TEMPERATURA ELEMENTOS DE CONTROL	70
FIGURA 55. MODELO FÍSICO DE ISA-88.01 (ZUMO DE NARANJA).....	72
FIGURA 56. DIAGRAMA P&ID CASO DE ESTUDIO	74
FIGURA 57. ARQUITECTURA CASO DE ESTUDIO.	78
FIGURA 58 RED DE PETRI VÁLVULA FY 101-2	80
FIGURA 59 RED DE PETRI VÁLVULA FY 101-1	80
FIGURA 60 RED DE PETRI VÁLVULA FY 102-2	81
FIGURA 61. RED DE PETRI FY 102-1	81
FIGURA 62. RED DE PETRI VÁLVULA FY 103	82
FIGURA 63. RED DE PETRI VÁLVULA FY 104-1	83
FIGURA 64. BLOQUE DE FUNCIÓN INICIO_CICLO.....	88
FIGURA 65. BLOQUE DE FUNCIÓN PID.....	89
FIGURA 66. BLOQUE DE FUNCIÓN DERIVADA	90
FIGURA 67. BLOQUE DE FUNCIÓN INTEGRAL.....	91
FIGURA 68. BLOQUE DE FUNCIÓN CÁLCULO_PID	92
FIGURA 69. BLOQUE DE FUNCIÓN CÁLCULO PID.	93
FIGURA 70. CTROL_TK_NIVEL1.	95
FIGURA 71. CTROL_TK_NIVEL2.	96
FIGURA 72. CTROL_TEMPERATURA.....	98
FIGURA 73. CTROL_MEZCLADOR.....	99
FIGURA 74. APLICACIÓN DE CONTROL.	102
FIGURA 75. FLUJO DE DATOS ENTRE RECURSOS.	104
FIGURA 76. CONFIGURACIÓN DE DIRECCIONES IP.	105
FIGURA 77. RECURSO_DISPLAY	106
FIGURA 78. RECURSO R_C-100.	108
FIGURA 79. RECURSO R_LC-101-1	109
FIGURA 80. RECURSO R_LC-102-1	110
FIGURA 81. RECURSO R_TC-103	112

FIGURA 82. RECURSO R_SC-104.....113
FIGURA 83. MODELO DE SISTEMA.114

ÍNDICE DE TABLAS

TABLA 1. DIFERENCIAS ENTRE UN SISTEMA SCADA Y UN DCS	25
TABLA 2. ESTADOS-TRANSICIONES	33
TABLA 3. EVENTOS Y DATOS DEL BLOQUE IO_WRITER	37
TABLA 4. EVENTOS Y DATOS DEL BLOQUE IO_READER	38
TABLA 5. EVENTOS Y DATOS DEL BLOQUE PUBLISH.....	39
TABLA 6. EVENTOS Y DATOS DEL BLOQUE SUSBCRIBE	40
TABLA 7. LISTA DE INSTRUMENTOS P&ID	75
TABLA 8. LISTA DE VARIABLES CASO DE ESTUDIO	75
TABLA 9. LISTA DE EQUIPOS CASO DE ESTUDIO.....	76
TABLA 10. FUNCIONALIDAD DE LOS DISPOSITIVOS DE CONTROL	83
TABLA 11. DESCRIPCIÓN DE EVENTOS Y DATOS DEL BLOQUE DERIVADA.....	90
TABLA 12. DESCRIPCIÓN DE EVENTOS Y DATOS DEL BLOQUE INTEGRAL.....	91
TABLA 13. DESCRIPCIÓN DE EVENTOS Y DATOS DEL BLOQUE CÁLCULO: PID	92
TABLA 14. DESCRIPCIÓN DE EVENTOS Y DATOS DEL BLOQUE CTROL_SISTEMA.....	94
TABLA 15. DESCRIPCIÓN DE EVENTOS Y DATOS DEL BLOQUE CTROL_TK_NIVEL1	95
TABLA 16. DESCRIPCIÓN DE EVENTOS Y DATOS DEL BLOQUE CTROL_TK_NIVEL2.....	97
TABLA 17. DESCRIPCIÓN DE EVENTOS Y DATOS DEL BLOQUE CTROL_TEMPERATURA.	98
TABLA 18. DESCRIPCIÓN DE EVENTOS Y DATOS DEL BLOQUE CTROL_MEZCLADOR	100
TABLA 19. DISPOSITIVOS EMPLEADOS EN LA APLICACIÓN DE CONTROL.....	103
TABLA 20. DISPOSITIVOS CON SU RESPECTIVO RECURSO.....	103
TABLA 21. LISTA DE CHEQUEO.....	114

LISTA DE ANEXOS

ANEXO A. Estándar IEC 61131.

ANEXO B. Características de los componentes, Lista de materiales de la arquitectura propuesta

INTRODUCCIÓN

Los sistemas de automatización están presentes cada vez más en las empresas de producción, debido a las exigencias actuales del mercado estos sistemas deben responder de forma eficiente a cambios en los requisitos de producción, cumpliendo con características de flexibilidad, calidad en los procesos, adaptabilidad e interoperabilidad [1].

La forma de lograr estos objetivos es la utilización de arquitecturas distribuidas o DCS (*Distributed Control Systems*) los cuales satisfacen las necesidades actuales de las industrias manufactureras. Estas arquitecturas pretenden descentralizar los sistemas de producción a diferencia de los controles centralizados, fundamentándose en que el mercado es variable y se necesitan sistemas que estén en la capacidad de responder rápidamente a cambios en la demanda manteniendo un funcionamiento estable mediante un uso eficiente de los recursos, generando productos de excelente calidad que permitan una alta competitividad en el mercado global.

En el año 2005 se publicó el estándar ²IEC-61499 como una metodología para el modelado de sistemas de control distribuido. El elemento central de la norma son los Bloques de Función (FB), que permite la encapsulación de funcionalidades tales como: software de control, operaciones matemáticas, comunicación, entre otros, todo esto por medio de algoritmos.

En su primera parte “IEC 61499-1”, presenta los modelos y la arquitectura para representar un DCS [2], la carencia de procedimientos en la implementación del estándar y la deficiencia de herramientas aplicables tanto hardware como software, ocasionan que no se tome en cuenta de manera generalizada en aplicaciones industriales de control distribuido [3]

Partiendo de lo expuesto, se concluye que se debe concebir propuestas alrededor de procedimientos de implementación del estándar, los cuales generen un espacio de confianza en su aplicación y que permitan crear casos de éxito en la industria. Como un primer acercamiento en la búsqueda de este objetivo, se presenta un estudio detallado del IEC 61499-1, para posteriormente establecer una propuesta procedimental de aplicación alrededor del IEC 61499-1, propuesta que será soportada conceptualmente en el laboratorio de control de procesos del PIAI³.

² IEC, International Electrotechnical Commission

³ PIAI, Programa Ingeniería en Automática Industria

CAPÍTULO I

1. CONCEPTOS FUNDAMENTALES.

En este capítulo se exponen inicialmente los elementos que integran un sistema de control, consecutivamente se presentan las dos tecnologías que actualmente se emplean para el control de procesos industriales: sistemas SCADA, sistemas de control distribuido (DCS), mostrando definiciones, comparaciones, características y arquitecturas.

Posteriormente se enunciarán las necesidades actuales de los sistemas de control distribuidos, finalmente se presentará una introducción al estándar IEC 61499.

1.1. Control de procesos.

El objetivo fundamental de la automatización industrial es aliviar al operador de tareas repetitivas y de reducida exigencia para que se concentre en “obtener la mayor producción con la mínima inversión” exigiendo que el proceso de fabricación sea *evolutivo* y *flexible* a través de la interacción de actividades tecnológicas, informáticas, económicas y administrativas [4]. De esta manera se logrará eficiencia, optimización y gran eficacia en los procesos de manufactura donde se implemente el concepto.

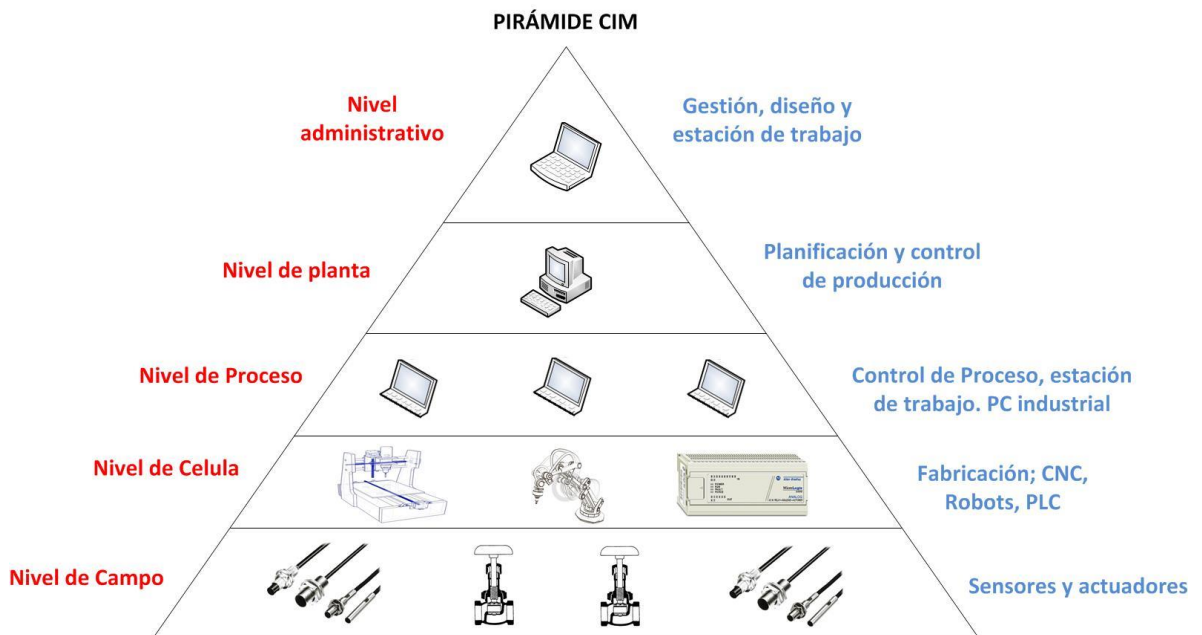
Al principio, los procesos industriales fueron controlados manualmente por un operador. El operador observaba lo que sucedía, hacía ajustes basados en las instrucciones de operación y en el propio conocimiento que tenía del proceso. La adecuada reacción de un operador experimentado mediaba entre una evolución normal del proceso y otra errática. Debido a los problemas que se presentaron, como manejo de una sola base de datos, agilidad en la toma de decisiones, ineficacia a la hora de actuar cuando se presentaban problemas, entre otros. Surgen los *Controladores Locales* para comenzar a obtener mejores productos y con mayor calidad. Un controlador local permite a un operario llevar el *control de varios “Lazos” del proceso* [5], pero como todo avance que surge y que está en proceso de investigación, tuvo ciertos errores, debían ser repartidos a través de toda la planta. Esta distribución ocasionaba pérdidas de tiempo en ajustes que se hacían de forma aleatoria y con más frecuencia.

En adelante se crearon los sistemas de supervisión con ordenadores centralizados con un nivel de seguridad adecuado y evitando que una “falla” de este paralice todo el sistema. Se empezaron a utilizar *Controladores Analógicos* vinculados directamente al proceso, optimizados para la variable que debían controlar. Estos controladores son ahora los que realmente controlan el proceso, dejando al ordenador central la función de los cambios de puntos de consigna, es decir, el valor de referencia con el que se ha de comparar la variable controlada para mantenerla siempre optimizada. Esta combinación de actuaciones recibe el nombre de control supervisorio o control de puntos de consigna (SPC, Set Point Control) [5].

1.1.1. Estructura general de los sistemas de control industrial.

En la *Figura 1*. se muestra los diferentes niveles dentro de la Pirámide CIM⁴, señalando; nivel de Proceso, nivel de Célula y nivel de Campo, siendo estas las que se implementan normalmente en la industria [6]

Figura 1. Pirámide CIM



Fuente: Modificado de [5]

⁴ CIM: Manufactura integrada por computadora

En el **nivel de procesos** se encuentran los dispositivos de hardware y software que brindan la información necesaria para llevar a cabo las operaciones de planta, con una interfaz amigable y entendible para el operador.

En el **nivel de célula y campo**, están los dispositivos (PLCs y/o PC Industriales) que permiten llevar a cabo las acciones de control en conjunto con los actuadores. Los elementos que conforman el proceso de manufactura dentro del piso/planta conforman una **comunicación vertical** (entre los distintos procesos y su control) y **comunicaciones horizontales** (entre distintos elementos de control)

Se procede a describir las dos arquitecturas de control industrial más utilizadas para los niveles descritos anteriormente: sistemas SCADA y DCS.

1.2. Sistema SCADA (sistema de control, supervisión y de adquisición de datos).

Históricamente, un sistema SCADA⁵ [7] fue propuesto para controlar variables discretas y controlar proceso BATCH⁶ y se define como un conjunto de componentes, tecnologías hardware y software para implementar las funciones del sistema de supervisión tales como adquirir los datos, realizar el monitoreo remoto [8], razonar acerca del comportamiento del proceso, b proponer y ejecutar acciones manteniendo las condiciones de operación normales en caso de fallas. Es un sistema basado en computadores que permiten supervisar y controlar a distancia una instalación de cualquier tipo[9]

1.2.1. Funciones principales de un sistema SCADA.

Las principales funciones [10] que tiene un sistema SCADA están integradas por la *Supervisión remota de instalaciones y equipos*; permitiendo al operador conocer el estado de desempeño de las instalaciones y los equipos alojados en la planta, lo que permite dirigir las tareas de mantenimiento y estadística de fallas. El *Control remoto de instalaciones y equipos*, activando o desactivando los equipos remotamente, de manera automática y manual. Además es posible ajustar los parámetros, valores de referencia, algoritmos de control, entre otros. El

⁵ SCADA; "Supervisory Control and Data Acquisition". Adquisición de datos y control de supervisión.

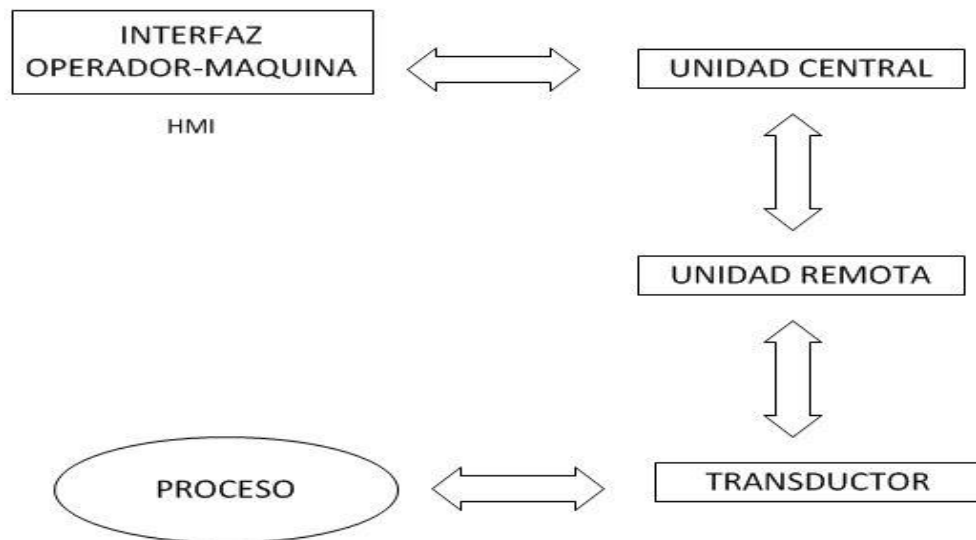
⁶ BATCH: Procesamiento por lotes

Procesamiento de datos; es el conjunto de datos adquiridos que conforman la información que alimenta el sistema, es procesada, analizada, comparada con datos anteriores y con otros puntos de referencia, dando como resultado el análisis de toda la información por medio de diagramas. *Visualización grafica de forma dinámica*; el sistema es capaz de brindar imágenes en movimiento que representen el comportamiento del proceso, dándole al operador la impresión de estar presente dentro de la planta. Estos gráficos también pueden corresponder a curvas de las señales controladas. *Generación de reportes*; el sistema debe permitir generar informes con datos estadísticos del proceso en un tiempo determinado por el operador. La *representación de alarmas*; a través de las señales de alarma se logra alertar al operador frente a una falla. La *Programación de eventos*; se refiere a la posibilidad de programar subprogramas que brinden automáticamente reportes, estadísticas, graficas de curvas, activación de tareas automáticas, etc. [11]

1.2.2. Elementos de un sistema SCADA

Los elementos de un sistema SCADA [12] son: *Una interfaz HMI* que permite la interacción del ser humano con los medios tecnológicos implementados. Una *unidad Terminal maestra (MTU)* que muestra a los servidores y al software HMI responsable para comunicarse con el equipo en campo (RTU, PLC, etc.). La *unidad terminal Remota* es instalada en una posición remota que obtiene datos, los descifra en un formato y los transmite de nuevo a una MTU. La RTU también recoge la información del dispositivo principal y pone los procesos en ejecución que son dirigidos por la MTU. Los *sistemas de comunicación*; se encargan de la transferencia de la información del punto donde se realizan las operaciones, hasta el punto donde se supervisa y controla el proceso [13]. Lo conforman los transmisores, receptores y medios de comunicación. Por último se encuentran los *Transductores* que son capaces de transformar o convertir un determinado tipo de energía de entrada, en otra de diferente salida.

Figura 2. Esquema de los elementos de un sistema SCADA



Fuente: Tomada y Modificada de [12]

Para realizar el intercambio de datos entre los dispositivos de campo y la estación central de control y gestión, se requiere un medio de comunicación, existen diversos medios y pueden ser cableados (cable coaxial, fibra óptica, cable telefónico) o no cableados (microondas, ondas de radio, comunicación satelital), cada fabricante emplea diferentes protocolos de comunicación, ya que no existe un estándar definido para la estructura de los mensajes [10], [14]

1.2.3. Programación de los sistemas de control en sistemas SCADA

En el año de 1992 fue creado el estándar IEC-61131 [8], con el fin de estandarizar los lenguajes de programación, debido a que por esa época cada proveedor generaba su propio lenguaje de programación.

IEC 61131-3 define 5 lenguajes de programación (ver Anexo A) de los cuales dos son textuales y tres son gráficos siendo los siguientes [15].

Lenguajes Textuales:

- Lista de Instrucciones (IL, Instruction List)
- Texto Estructurado (ST, Structured Text)

Lenguajes Gráficos:

- Diagrama de contactos (LD, Ladder Diagram)
- Diagrama de Bloques de funcionales (FBD, Function Block Diagram)
- Gráfica de función secuencial (SFC, Sequential Function Chart)

Los PLC que componen los sistemas SCADA, se programan normalmente en diagrama de escalera o "Ladder", siendo este el tipo de lenguaje empleado, como se mencionó anteriormente. IEC 61131-3 logro estandarizar los lenguajes de programación, con la aparición de nuevos requisitos, y cambios constantes en la demanda de la industria, el estándar comenzó a presentar dificultades en: los lenguajes que se presentan de forma cíclica, siendo ejecutados línea por línea, la gran cantidad de código ha hecho que el mantenimiento de estos programas sea complicado, además de que los cambios en los requisitos hagan que prácticamente se desechen programas completos por su falta de *flexibilidad*.

1.2.4. Ventajas y desventajas de un sistema SCADA

A continuación se muestran algunas de las ventajas y desventajas que representa implementar o adquirir un sistema SCADA para un proceso industrial.

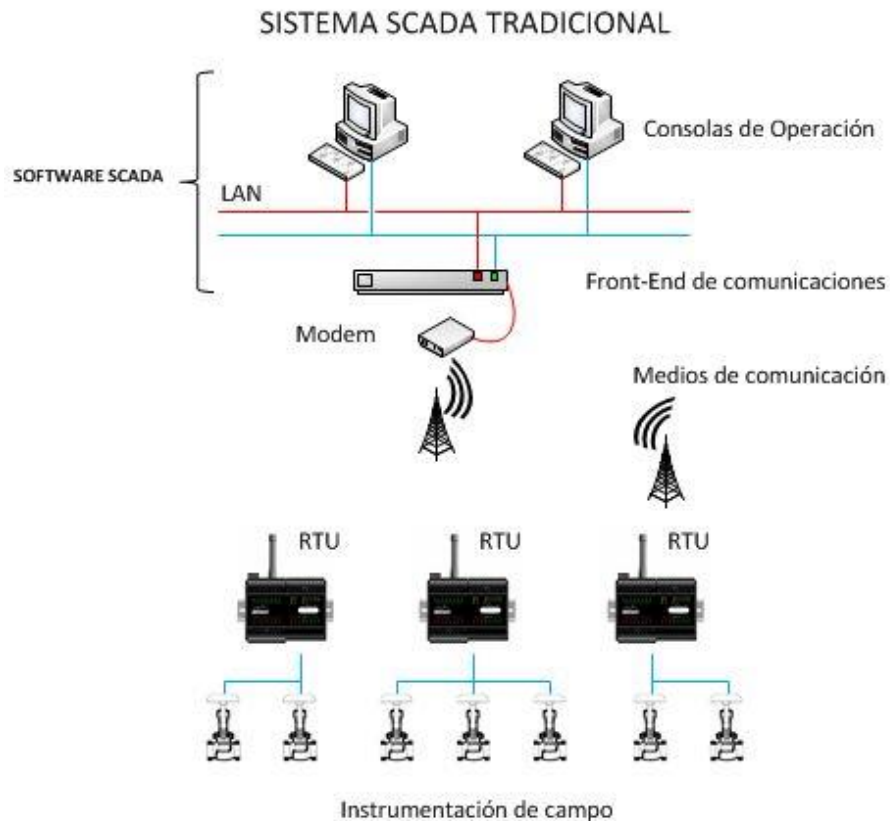
1.2.4.1. Ventajas

- La computadora puede registrar y almacenar una gran cantidad de datos.
- Los datos pueden mostrarse de la manera requerida por el usuario.
- Se pueden conectar al sistema muchos sensores distribuidos sobre una gran área.
- Se puede recolectar muchos y diversos tipos de datos desde los dispositivos distribuidos en red.
- Se implementan en sitios geográficamente dispersos [11] que pueden llegar a tener miles de kilómetros [13].
- Mejoran la eficacia del proceso de monitoreo y control proporcionando información para toma de decisiones [10].
- Se puede actuar y variar las variables de control en tiempo real [11].
- Sus componentes son diseñados por distintos proveedores sin coordinación entre sí [10].

1.2.4.2. Desventajas

- Programación compleja.
- Inexistencia de reloj global (en ocasiones)
- Fallos independientes; (aunque el sistema sea más robusto)
- Inseguridad al momento de operar.
- Las decisiones o acciones últimas de control recaen sobre el operario [10]

Figura 3. Arquitectura de un sistema SCADA



Fuente: Tomado y Modificada de [14]

Los SCADA han servido como sistema centralizado donde se monitorea el estado del proceso que puede estar a varios kilómetros de distancia. Los procesos que se encuentren ubicados dentro la planta industrial, requerirá de los sistemas de control distribuido, siendo este, el tema de estudio del presente documento. A continuación se definen los sistemas de control distribuido.

1.3. Sistemas de control Distribuido DCS.

Los sistemas de control distribuido (DCS) nacieron en la década de los 70, permitiendo descentralizar los lazos de control [6], realizan control automático en lazo cerrado, antes de empezar con la definición se debe hacer la siguiente aclaración: el termino DCS se utiliza para referirse al sistema, no al elemento que realiza el control sobre el sistema.

Un DCS en general hace referencia a un sistema en el cual los elementos de control no son centrales en una ubicación, sino distribuidos a través del sistema con cada subsistema controlado por uno o más controladores[9]. El sistema completo de controladores son interconectados por una red de comunicaciones para su monitoreo.

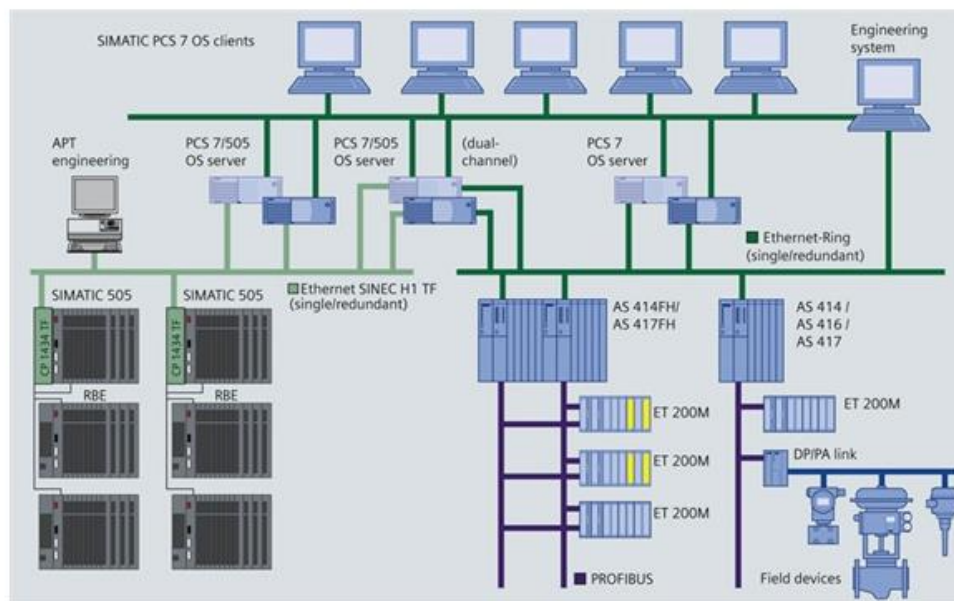
En sus inicios los DCS fueron creados para realizar el control a variables analógicas, y controlar procesos continuos, procesos que no podían detenerse; en la actualidad presentan una solución integral entre software y hardware, es decir, un proveedor se encarga de proponer la arquitectura del sistema [16], donde se desarrolla centralizadamente una sola aplicación, que contiene: las configuraciones y los programas de control de todos los controladores, además de la configuración de la interfaz gráfica [3], alarmas, tendencias, etc. Las redes también están incluidas, permitiendo comunicación entre los dispositivos [17], mediante protocolos de comunicaciones propietarios, por lo cual se logra ahorrar y reducir drásticamente el tiempo de implementación [18]

Algunos sistemas de control distribuido propietarios ofrecidos por los diferentes proveedores son:

- ABB : 800xA [19]
- Emerson: DeltaV y Ovation [20]
- Siemens: Simatic pcs 7 [21]
- Honeywell: Experion PKS [22]
- Yokogawa: CENTUM VP y CENTUM CS [23]

La *Figura 4* muestra la arquitectura presentada por el sistema de control distribuido SIMATIC PCS 7 de Siemens.

Figura 4. SIMATIC PCS 7 de Siemens.



Fuente: Tomada de [24]

1.3.1. Generalidades de un Sistema de Control Distribuido.

Un DCS se desarrolló con el concepto de distribuir la carga de trabajo entre distintos controladores, con el fin de poder gestionar muchos parámetros a la vez. Estos equipos se interconectan entre sí, cada uno está especializado en una función o en un área de la planta [25]; intentando que cada controlador sea lo más independiente posible del resto.

Se utilizan en general en los sistemas de control en los cuales existen un conjunto de subprocesos o áreas funcionales [26], que deben trabajar de forma ordenada y coordinada, estando distribuido físicamente en toda la instalación de la planta o industria [22]. Cada unidad posee un elemento de control independiente o dimensionado de acuerdo a los requerimientos de procesos de la unidad.

El DCS presenta distribución de las tareas del proceso por equipos de control [27], se establece comunicación horizontal, los controladores empleados presentan “redundancia”, este concepto contempla que si un dispositivo de control entra en falla y le impide cumplir las funciones asignadas, cualquiera de los controladores

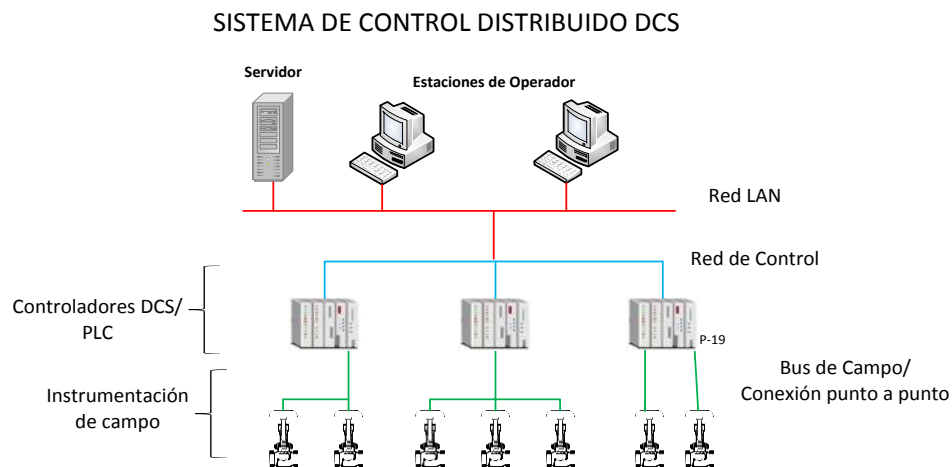
adjuntos al sistema debe estar en la capacidad de retomar las tareas del dispositivo que sale de operación.

La configuración DCS utiliza objetos de control definidos automáticamente al software de programación, lo que simplifica la configuración [28]. Al configurar una etiqueta, queda para todo el sistema, ahorrando tiempo y mejorando la calidad. Por ejemplo, si se tiene una válvula de salida, hay una librería para no crear la lógica desde cero.

El Sistema de Control Distribuido propietario se encuentra compuesto por una sola base de datos, instrumentos de campo, de acondicionamiento y procesamiento de señal, dispositivos de control (Controladores DCS, PLCs), Servidores, Estaciones de Operador y redes de comunicación entre campo, control y operador, que hacen del DCS un sistema que visualiza, documenta y controla el funcionamiento del proceso en tiempo real [29].

Los DCS constan de varios niveles de control, la *Figura 5*. Muestra los niveles y elementos que conforman un DCS

Figura 5. Conexiones en un sistema de control distribuido



Fuente: Tomada y Modificada de [3]

1.3.2. Componentes funcionales de un DCS.

Dentro de los componentes funcionales de un DCS se Incluyen, Estaciones de Trabajo (Operaciones de ingeniería), controladores, tarjetas de I/O, Bus I/O, una

red de control de alta velocidad y tecnología de control. Se muestra un diagrama de los principales componentes de un sistema de control distribuido en la *Figura 6*.

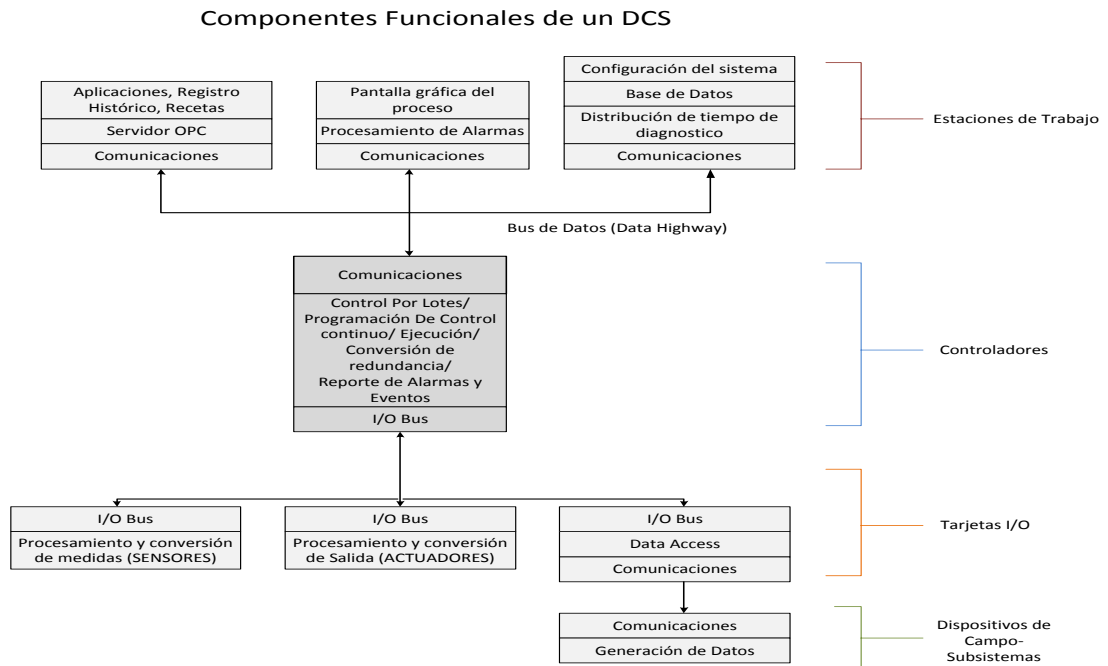
La instrumentación de campo se encarga de recolectar el valor de la variable de la planta y de enviar las señales de control para tomar acción en el proceso físico. Además, estos dispositivos pueden recolectar variables adicionales como Tag's, fallas, diagnósticos, estado del equipo, variables secundarias, etc. Aparte de poder configurarlos remotamente desde una estación de mantenimiento, esto será posible si tienen la capacidad de comunicarse mediante algún protocolo de campo como: Hart, Profibus, DeviceNet, Etc[30].

Los *Controladores* DCS, PLCS recogen las señales de dispositivos análogos, digitales, mediante buses de campo o conexiones punto a punto. Posteriormente de la recepción de estos datos, se ejecuta la lógica de control, recepcionando y enviando señales de/hacia campo, ejecutando control tipo continuo o BATCH (Lote). La red de control contiene los datos de los instrumentos, del controlador y hace que estén disponibles para las estaciones de operación, estaciones de ingeniería, base de datos de control, sistemas de alarmas, generadores de reportes, etc. Una funcionalidad clave del sistema es tener redundancia para la red de datos principal o datahighways, controladores, I/O, buses de campo y en algunos casos Estaciones tolerante a fallas (Workstation) [31].

Los componentes (Característicos) principales de una arquitectura DCS son:

- Configuración.
- Comunicaciones.
- Control.
- Alarmas y eventos.
- Diagnóstico.
- Redundancia.
- Datos históricos.
- Seguridad.
- Integración Software/Hardware.
- Base de datos.
- Único proveedor.

Figura 6. Componentes funcionales de un DCS



Fuente: Tomada y modificado de [31]

1.3.3. Ventajas de un DCS.

- Alto grado de automatización.
- Las estaciones son independientes unas de otras.
- Implementación rápida.

1.3.4. Desventajas de un DCS.

- Arquitectura definida por un proveedor.
- Compatibilidad de las comunicaciones.
- Interoperabilidad.

1.4. Aspectos Comparativos entre un SCADA y un DCS

Las diferencias que enmarcan a estos dos tipos de sistemas de control son amplias, A continuación se muestran algunas de las diferencias, mediante una tabla comparativa.

Tabla 1. Diferencias entre un sistema SCADA y un DCS

ASPECTO	SCADA	DCS
Tipo de arquitectura	CENTRALIZADA	DISTRIBUÍDA
Tipo de control predominante	SUPERVISION	REGULATORIO
Funcionamiento del sistema	Adquisición de datos y control de supervisión	Visualiza, documenta y controla.
Área de acción	Áreas geográficamente distribuidas	Área de la planta
Unidades de adquisición de datos	Remotas, PLCs	Controladores DCS, PLCs
Medios de comunicación	Radio, satélite, líneas telefónicas, conexión directa, LAN, WAN	Redes de área local, conexión directa
Funcionalidad	No aplica	Redundancia.

Fuente: Tomada y Modificada de [13]

La industria requiere de sistemas de control que con el pasar del tiempo ofrezca una mayor flexibilidad a los cambios que se den en la demanda sobre nuevas especificaciones del producto, siendo indispensable pensar en adquirir tecnologías que sean capaces de innovar y de optimizar los recursos, solventando las futuras necesidades a bajos costos. Por lo anterior, se detallaran las nuevas necesidades para los sistemas de control distribuido.

1.5. Necesidades de los sistemas de control distribuido.

Como se mencionó anteriormente los sistemas de control distribuido propietarios, tienen una arquitectura especificada por el fabricante, tendiendo a ser sistemas de solo configurar, es decir al estar ya definidos, las empresas de manufactura deben adaptarse a la propuesta dada. Estos sistemas suelen ser muy costosos, imposibilitando a las pequeñas empresas a realizar una implementación de ellos, además al tener protocolos de comunicación propietarios, se dificulta el añadirle elementos hardware y software al sistema, de acuerdo algún requerimiento específico.

La dinámica de la industria predice que no hay procesos fijos, por lo cual cada empresa tiene sus propias características, y exigencias de automatización, lo que hace necesario arquitecturas, que permitan disponer de plantas personalizadas e

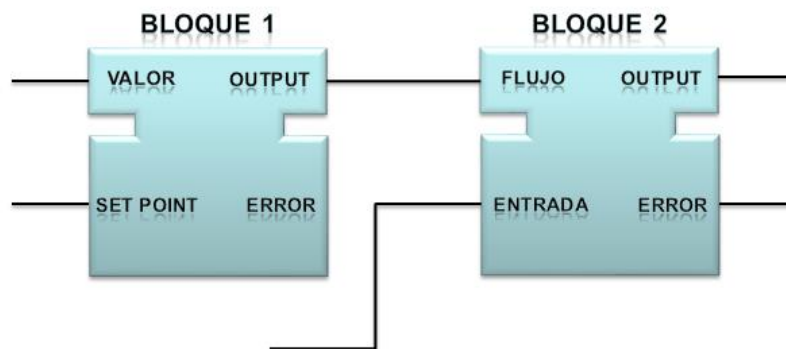
individualizadas, con el fin de que las líneas de producción tengan la posibilidad de ser adaptadas a nuevos procesos lo más rápidamente [15].

El desarrollo de los microprocesadores le ha permitido a los PLC tener más protagonismo en los DCS, debido a su alta velocidad, funcionalidad y costo relativamente bajo. Ahora es posible con los PLC modernos en hardware construir una "copia" de un sistema de control distribuido. Sin embargo, lo que actualmente le falta a los PLC es el mismo nivel de integración de hardware y software, necesario para construir sistemas de control distribuido funcional, es decir como realmente viene de fábrica. [12]

La programación actualmente de los DCS se realiza por medio de los bloques de función definidos en el estándar IEC 61131-3, presentando una serie de limitaciones definidas a continuación [32]

- Los bloques de función se puede enlazar con sólo unir las conexiones de flujo de datos entre la entrada del bloque y variables de salida, como se observa en la *Figura 7*
- El orden normal de ejecución es determinado por la dependencia del bloque de función con los otros bloques; normalmente se ejecuta de izquierda a derecha porque los bloques de la derecha dependen de los valores de salida de bloques de la izquierda.
- Cada bloque de función proporciona un único algoritmo interno que es ejecutado cuando el bloque de funciones es invocado.

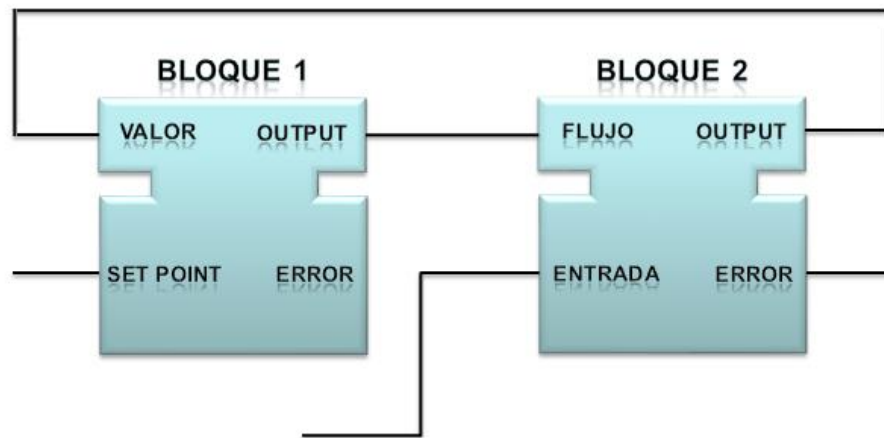
Figura 7. Red de bloques de función IEC 61131-3



Fuente: Tomada y Modificada de [32]

- Sin embargo, cuando se introduce una red de realimentación *Figura 8*, el orden de ejecución no puede ser determinado a partir del diagrama, ya que la ejecución de ambos bloques depende de un valor de salida del otro bloque. En una red compleja, es muy difícil para el sistema de programación determinar un orden de ejecución válido.

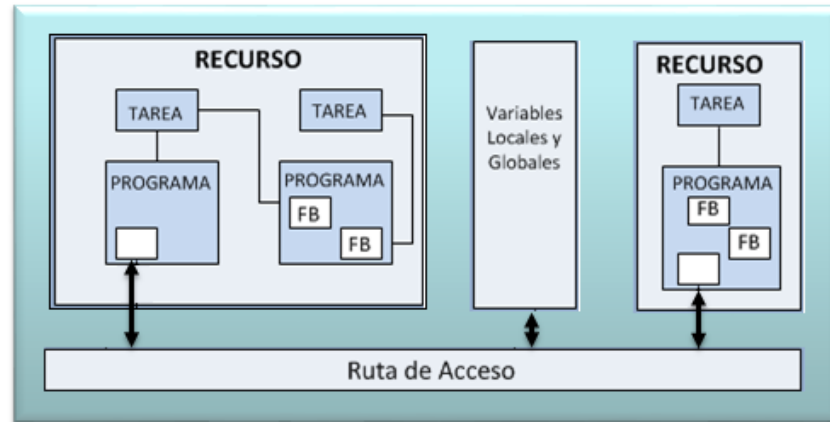
Figura 8. Red de bloques de función realimentados con IEC 61131-3



Fuente: Tomada y Modificada [32]

- Para superar el problema de red de bloques realimentados (ítem anterior), los diferentes proveedores proporcionan mecanismos adicionales que definen el orden de ejecución de los bloques. Por ejemplo, el usuario puede ver una lista de los bloques de función y asignar manualmente un orden de ejecución. Estos mecanismos, están fuera del estándar IEC 61131-3.
- El estándar IEC 61131-3, no presenta una arquitectura distribuida, debido a que el modelo software está enfocado para un PLC multirecurso (ver *Figura 9*), lo cual genera problemas de comunicación entre diferentes dispositivos.

Figura 9. Red de bloques de función realimentados con IEC 61131-3



Fuente: Tomada y Modificada [15]

1.6. Estándar IEC 61499

Debido a las limitaciones antes mencionadas del estándar IEC 61131-3. En el año 2005 la Comisión Electrotécnica Internacional (IEC) lanzó la norma internacional IEC 61449 que actualmente aún se encuentra en proceso de consolidación.

El objetivo principal del estándar IEC-61449 es conseguir un alto grado de control distribuido, pudiéndose utilizar sistemas embebidos existentes para descentralizar la lógica de control haciendo más flexible los sistemas [22]. Es desarrollado como una metodología para modelar sistemas de control distribuidos abiertos [33]. Define un modelo general y una metodología para describir bloques de función en un formato que es independiente de la implementación.

La metodología puede ser usada por diseñadores de sistemas para construir sistemas de control distribuido. Permitiendo modelar un sistema, el cual está compuesto por la conexión lógica de bloques de función [34], en el capítulo 2 se definirá la arquitectura y los modelos que aporta el estándar.

IEC 61499 intenta generar sistemas de control distribuido abiertos, buscando que los elementos que lo integran se conecten por medio de redes estandarizadas, intentado ser la primera metodología para el diseño de software distribuido que pueda ser embebido dentro de cualquier sistema DCS. Lo que se intenta es tratar

de igualar, lo que paso con el PC y el internet, un paquete de software puede ser instalado en cualquier computador y comunicado por una red abierta.

La carencia de procedimientos en la implementación del estándar IEC 61499 y la deficiencia de herramientas aplicables tanto hardware como software, ocasionan que no se tome en cuenta de manera generalizada en aplicaciones industriales de control distribuido [2].

Aunque ya empiezan a surgir proveedores de sistemas de control distribuido que ofrecen a sus clientes soluciones basadas en el estándar IEC 61499 [35], intentando posicionarlo a nivel industrial.

De acuerdo al análisis y los conceptos establecidos en este capítulo, los sistemas de control están formados por las necesidades de cada uno de los procesos, es decir, es preciso establecer características que puedan diferenciar un control del otro. Un sistema SCADA, al igual que un DCS llega a presentar distribución de la información por medio de las redes de comunicación. Lo más importante para poder diferenciarlos, es la distribución del control. Un DCS además de tener distribución de la información, tiene el control distribuido de cada uno de sus lazos, por medio de sensores y actuadores inteligentes, es ahí donde radica la diferencia de estos dos sistemas de control, cabe resaltar que con los avances de la tecnología, es cada vez más delgada la línea que separa un SCADA de un DCS. El resultado de implementar un sistema de control distribuido, es un control *robusto* y a la *vanguardia* de muchas de las necesidades que surgen en la actualidad, debido a que se introducen dispositivos inteligentes, se crea redundancia, confiabilidad y gran calidad en el proceso de producción.

Partiendo de lo expuesto anteriormente, se concluye que se debe concebir propuestas alrededor de procedimientos de implementación del estándar, los cuales generen un espacio de confianza en su aplicación y que permitan crear casos de éxito en la industria.

El presente documento propone un procedimiento para la implementación de un sistema de control distribuido de acuerdo a la definición de IEC 61499-1.

CAPÍTULO II

2. CONCEPTOS Y MODELOS DEL IEC 61499

Este capítulo presenta la arquitectura, conceptos y modelos que aporta el estándar IEC 61499, para el modelado de sistemas de control distribuido.

2.1. Arquitectura propuesta en IEC 61499.

Ha sido desarrollado específicamente como una arquitectura para el modelado de sistemas de control distribuidos, IEC 61499 no es una metodología de programación, sino que proporciona terminología, modelos y conceptos para permitir la implementación de un sistema de control distribuido por medio de bloques de función (FB) que pueden ser interconectados para definir el comportamiento de un sistema [36]. Actualmente se encuentra en un proceso de consolidación, proporcionando más funcionalidades, que solventan algunas de las limitaciones de su predecesora, el estándar “IEC 61131” [15]. Siendo este el primer paso hacia metodologías de programación estandarizadas para sistemas distribuidos [32]. En la actualidad el modelado para sistemas de control distribuido no es tenido en cuenta, en la ingeniería de los sistemas de automatización industrial, el desarrollo y aplicación de modelos rara vez se llevan a cabo de forma sistemática [37].

IEC 61499 fue diseñado por el comité técnico (TC 65⁷) de medida, control y automatización de procesos industriales, siendo aprobada la primera versión en agosto de 2005 [38] compuesta de cuatro partes [39]:

1. **Arquitectura.** IEC 61499-1, contiene requisitos generales, definiciones y modelos de referencia. Reglas para la declaración de tipos de FB y reglas para su comportamiento.
2. **Requisitos de herramienta software.** IEC 61499-2, define requisitos de herramientas software, que soportan la ejecución de las tareas de ingeniería de sistemas y especificación de tipos de FB.
3. **Manual Informativo.** IEC 61499-3, contiene la información para el entendimiento, la aceptación y la aplicabilidad, tanto de la arquitectura

⁷ TC: Technical Committee

IPMCS⁸, como de herramientas software que cumplan con las especificaciones del estándar.

4. **Reglas y Perfiles de Conformidad.** IEC 61499-4, contiene la definición de las reglas para el desarrollo de perfiles de conformidad, las cuales especifican las características para implementar los apartados 1 y 2.

Ahora en adelante cuando se mencione el estándar IEC 61499 se hace referencia a su primera parte (IEC 61499-1), aclarando lo anterior se procede a revisar los distintos modelos introducidos por el estándar IEC 61499.

La arquitectura genérica y jerárquica de modelos, propuesta en el estándar permite entender la organización del sistema y sus componentes. Desarrollando una nueva estructura para aplicaciones de control distribuido [15], a continuación se procede a describir cada uno de los modelos definidos por la norma:

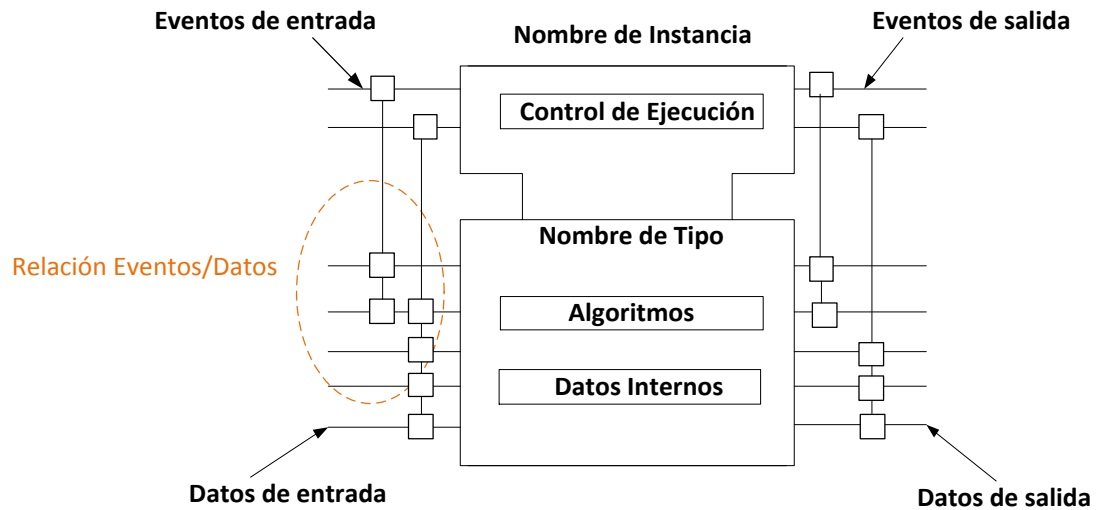
2.1.1. Modelo de bloque de función básico

El modelo de bloque de función básico, es la base de toda la arquitectura del estándar IEC 61499, permite encapsular cierta funcionalidad como lo es: control, operaciones matemáticas, comunicación, etc. [15].

En la *Figura 10*. Se representan las principales características de un bloque de función definido por IEC 61499. La parte superior del bloque de función, se denomina "Control de ejecución", la parte inferior del bloque de función, contiene los Algoritmos y Datos Internos, los cuales se ocultan dentro del bloque de función [40].

⁸ IPMCS, Sistemas de Control y Medida de Procesos Industriales

Figura 10. Modelo de bloque de función definido por IEC 61499.



Fuente: Tomada y Modificada de [32]

Las principales características de un bloque de función se resumen de la siguiente manera [41]:

- Cada bloque tiene un conjunto de eventos de entrada, que puede recibir eventos de otros bloques de función a través de conexiones de eventos.
- Hay uno o más eventos de salida, que pueden ser utilizados para propagar eventos hacia otros bloques.
- Hay un conjunto de datos de entrada, que permiten recibir los datos que van a ser procesados desde el bloque de función, estos datos tendrán diferentes tipos (BOOL, REAL, ETC).
- Existe un conjunto de datos de salida, empleados para transmitir los valores de los datos producidos dentro del bloque de función a otros bloques, estos datos tendrán diferentes tipos (BOOL, REAL, ETC).
- Cada bloque podrá tener un conjunto de datos internos que se utiliza para realizar cálculos internos.
- Para el desarrollo de los algoritmos se pueden utilizar lenguajes de programación de alto nivel tales como C, C++, o los lenguajes estandarizados bajo la norma IEC 61131.

- Cada tipo de bloque de función tiene un nombre de tipo y un nombre de instancia. Estos siempre se debe mostrar cuando el bloque se represente gráficamente.
- Cada bloque indicara la relación entre Eventos/Datos, es decir que eventos requieren determinados datos.

2.1.1.1. Control de Ejecución

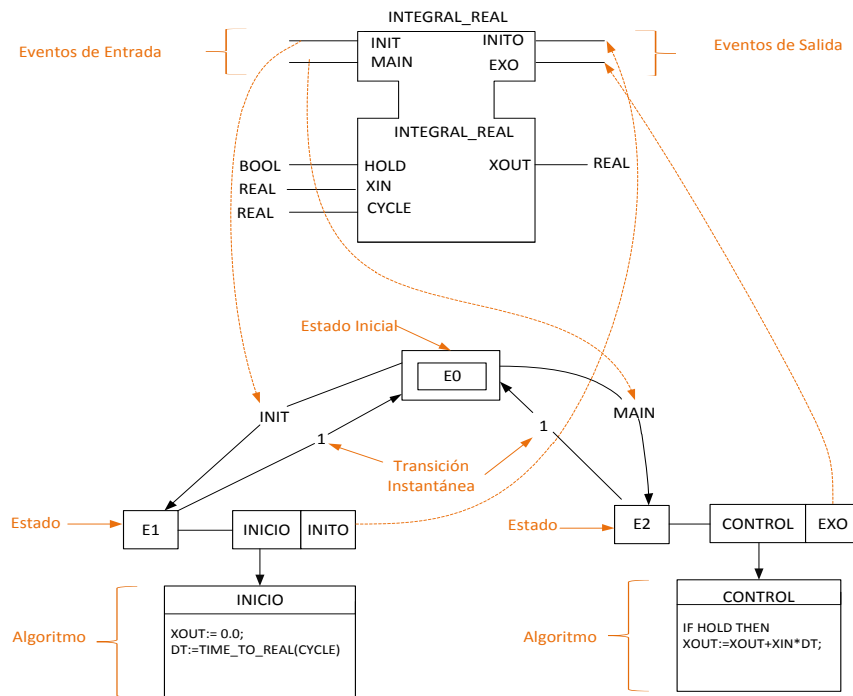
El Control de ejecución (CE), está dado en términos de una máquina de estados, donde los eventos de entrada serán transiciones, su función es asignar los eventos a los algoritmos; es decir, se define qué algoritmos especificados en la parte inferior del bloque se activan con la llegada de varios eventos y que eventos de salida se activan, la *Figura 11*. Representa el bloque INTEGRAL_REAL y su diagrama de CE, este bloque tiene como principal función calcular el valor de la integral para un control PID. El estado inicial del CE es E0, en respuesta a la entrada del evento INIT, el estado de CE cambia al estado E1, estado donde se ejecutará el algoritmo INICIO, y sobre la terminación de la ejecución de este algoritmo, se generara el evento de salida INITO y la Transición Instantánea 1, posteriormente se retornara al estado E0 [42]. La *Tabla 2* enumera las condiciones de transición para el cambio de estados de este ejemplo.

Tabla 2. Estados-Transiciones

Hacia	Transición	Desde
E1	INIT(Evento)	E0
E0	1(Instantánea)	E1
E2	MAIN(Evento)	E0
E0	1(Instantánea)	E2

Fuente: Tomada y Modificada de [32]

Figura 11. Gráfico de control de ejecución, definido por IEC 61499.



Fuente: Tomada y Modificada de [32]

2.1.1.2. Diferencias entre los bloques de función de IEC 61499 e IEC 61131

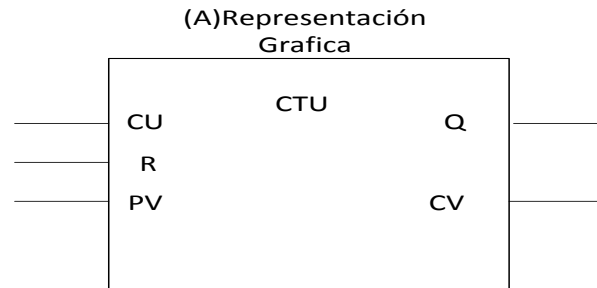
Los bloques de función definidos por IEC 61499 en comparación de los especificados por IEC 61131, diferencian eventos de datos, permite encapsular más de un algoritmo, a partir de la clasificación por eventos, a continuación con un ejemplo se expresarán las diferencias [43].

La *Figura 12*, representa un bloque de función definido por IEC-61131, la parte A muestra su representación gráfica mientras la parte B los algoritmos con los que cuenta el bloque, en este caso posee 2 algoritmos que dependiendo de los valores de las entradas "CU" y "R" serán activados [43].

IEC 61499 diferencia entre las conexiones de eventos y datos. La *Figura 13* representa un bloque de función definido por IEC 61499, presenta las mismas funciones que el bloque definido en la *Figura 12*, CU y R, en este caso se manejan como eventos, este bloque proporciona dos eventos de salida, CUO y RO. Cuenta

con 2 algoritmos “R” y “CU”, los cuales son controlados por el diagrama de control de ejecución [43].

Figura 12. Modelo de bloque de función compuesto definido por IEC 61131.



(B) Algoritmo

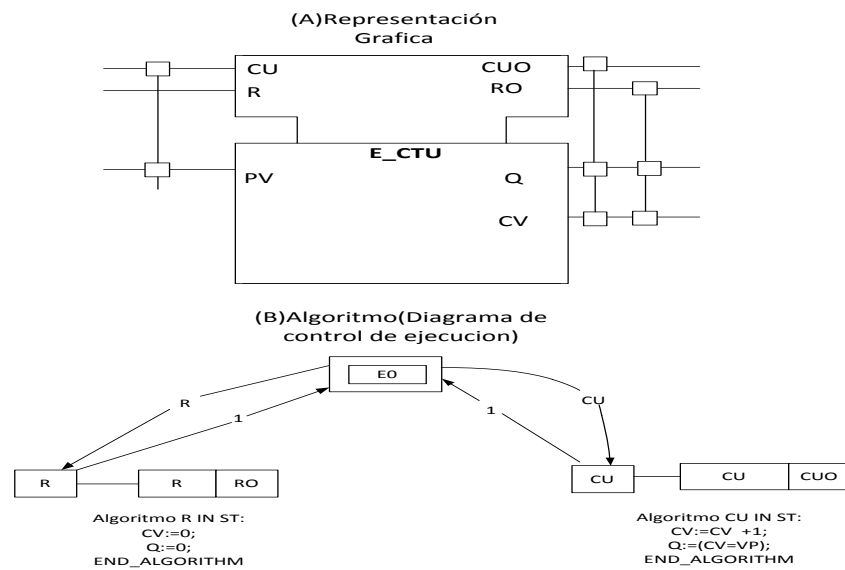
```

IF R
  THEN CV:=0
       Q:=0;
END IF
IF CU
  THEN CV:=CV +1;
       Q:=(CV=VP);
END IF
    
```

Fuente: Tomada y Modificada de [43]

Bloque de función de IEC 61499 con su respectivo diagrama de ejecución.

Figura 13. Modelo de bloque de función compuesto definido por IEC 61499.

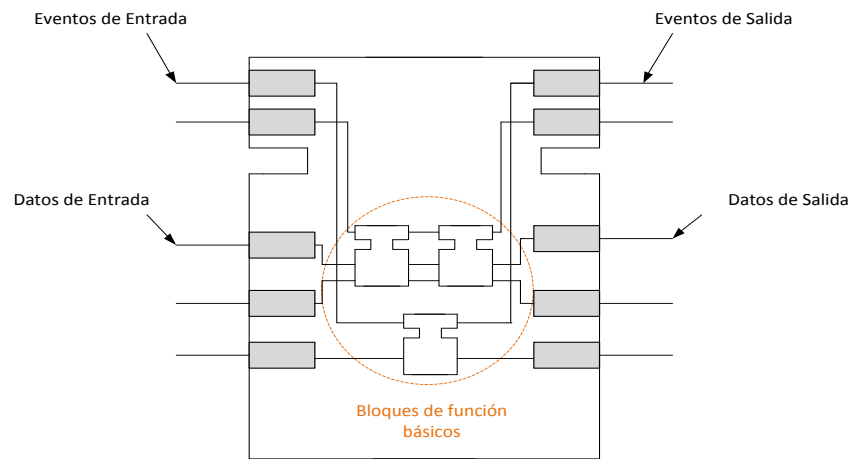


Fuente: Tomada y Modificada de [43]

2.1.2. Bloques de función compuestos

Los bloques de función compuestos, son bloques de función formados de bloques de función básicos, como lo indica la *Figura 14*. Así, dentro de cada bloque compuesto se puede encapsular una cantidad ilimitada de bloques de función básicos [44].

Figura 14. Modelo de bloque de función compuesto definido por IEC 61499.



Fuente: Tomada y Modificada de [32]

2.1.3. Bloque de función de interfaz de servicio

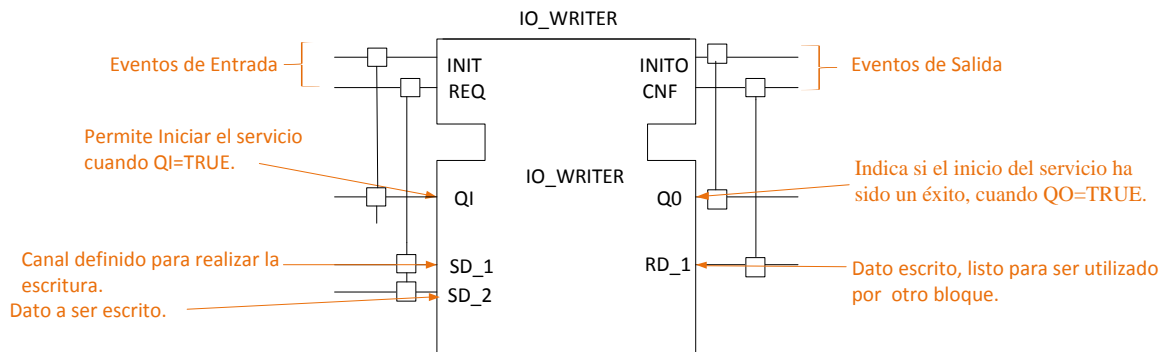
Los bloques de interfaz de servicio (SI), se encargan de proveer los servicios que los bloques necesitan, entre los servicios más comunes están la escritura y lectura de entradas/salidas (I/O) y el envío/recepción de datos entre dispositivos diferentes [44], a diferencia de los bloques de función básicos, estos bloques no cuentan con control de ejecución, ni algoritmos, a continuación se detalla su comportamiento de acuerdo a los eventos y datos.

2.1.3.1. Bloques IO_WRITER y IO_READER

Los bloques de función de interfaz de servicio que permiten la escritura y lectura de entradas/salidas son: IO_WRITER y el IO_READER [45]

El bloque IO_WRITER representado en la *Figura 15*. se utiliza para escribir valores en las salidas físicas del dispositivo [15].

Figura 15. Bloque IO_WRITER.



Fuente: Tomada y Modificada de [32]

En la *Tabla 3* se describen los eventos y datos del bloque de función IO_WRITER

Tabla 3. Eventos y datos del bloque IO_WRITER

Eventos Entrada	Datos Entrada
INIT: Al activarse este evento, inicia el servicio de escritura.	QI: Esta entrada de datos requiere tomar el valor verdadero (True), para completar el inicio del servicio de escritura.
REQ: Al activarse este evento, permite realizar la escritura.	SD_1: Define el canal por el cual se realizará la escritura del dato correspondiente.
	SD_2: Esta entrada recibe el dato a ser escrito.
Eventos Salida	Datos Salida
INITO: Este evento se activa cuando el bloque ha completado su inicialización.	QO: Esta salida al tomar el valor verdadero (True), indicara que la inicialización del bloque se realizó con éxito.
CNF: Este evento al activarse indica que el dato ha sido escrito.	RD_1: El valor del dato escrito estará disponible en esta salida.

Fuente: Tomada y Modificada de [32]

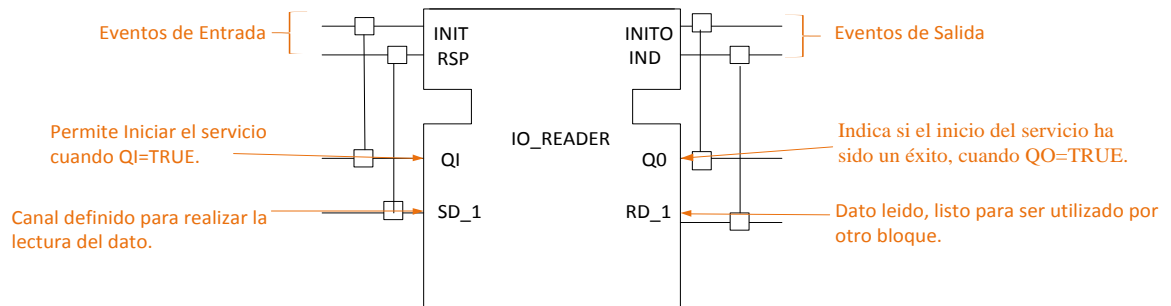
Por cada actuador (válvulas, bombas, variador, etc.) que tenga asociado el dispositivo tendrá un bloque de función IO_WRITER donde se tendrán los siguientes datos principales de entrada/salida.

- a. Entrada: El canal de salida empleado y definir la señal de salida empleada

- b. Salida: La señal de salida disponible para utilizar.

El bloque IO_READER representado en la *Figura 16*. Permite leer los valores de las entradas físicas del dispositivo, la *Tabla 5* describe los eventos y datos involucrados en este bloque.

Figura 16. Bloque IO_READER.



Fuente: Tomada y Modificada de [32]

En la *Tabla 4* se describen los eventos y datos del bloque de función IO_READER

Tabla 4. Eventos y datos del bloque IO_READER

Eventos Entrada	Datos Entrada
INIT: Al activarse este evento, inicia el servicio de lectura.	QI: Esta entrada de datos requiere tomar el valor verdadero (True), para completar el inicio del servicio de lectura.
RSP: Al activarse este evento, permite realizar la lectura.	SD_1: Define el canal por el cual se realizará la lectura del dato correspondiente.
Eventos Salida	Datos Salida
INITO: Este evento se activa cuando el bloque ha completado su inicialización.	QO: Esta salida al tomar el valor verdadero (True), indicara que la inicialización del bloque se realizó con éxito.
IND: Este evento al activarse indica que el dato ha sido leído.	RD_1: El valor del dato leído estará disponible en esta salida.

Fuente: Tomada y Modificada de [32]

Por cada sensor que tenga asociado el dispositivo, tendrá un bloque de función IO_READER, donde se tendrán los siguientes datos principales de entrada/salida

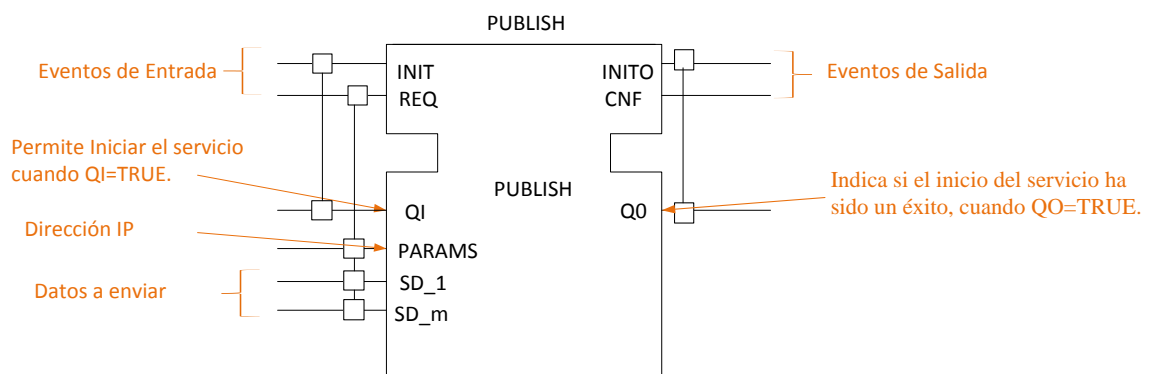
- a. Entrada: El canal que realizara la lectura.
- b. Salida: El valor del sensor leída.

2.1.3.2. Bloques PUBLISH y SUBSCRIBE

Los bloques de función de interfaz de servicio que permiten la comunicación entre dispositivos diferentes por medio del envío/recepción de datos son: PUBLISH y SUBSCRIBE.

El bloque PUBLISH representado en la *Figura 17* se utiliza para el envío de datos [42].

Figura 17. Bloque PUBLISH.



Fuente: Tomada y Modificada de [32]

En la *Tabla 5* se describen los eventos y datos del bloque de función PUBLISH

Tabla 5. Eventos y datos del bloque PUBLISH

Eventos Entrada	Datos Entrada
INIT: Al activarse este evento, inicia el servicio de envío de datos.	QI: Esta entrada de datos requiere tomar el valor verdadero (True), para completar el inicio del servicio de envío de datos.
RSP: Al activarse este evento, permite realizar el envío de datos.	PARAMS: Esta entrada contiene los datos de direccionamiento de red (por ejemplo, dirección IP).
	SD_1, SD_m: Esta entrada recibe los datos a ser enviados.
Eventos Salida	Datos Salida
INITO: Este evento se activa cuando el bloque ha completado su inicialización.	QO: Esta salida al tomar el valor verdadero (True), indicara que la inicialización del bloque se realizó con éxito.
CNF: Este evento al activarse indica que los datos han sido enviados.	

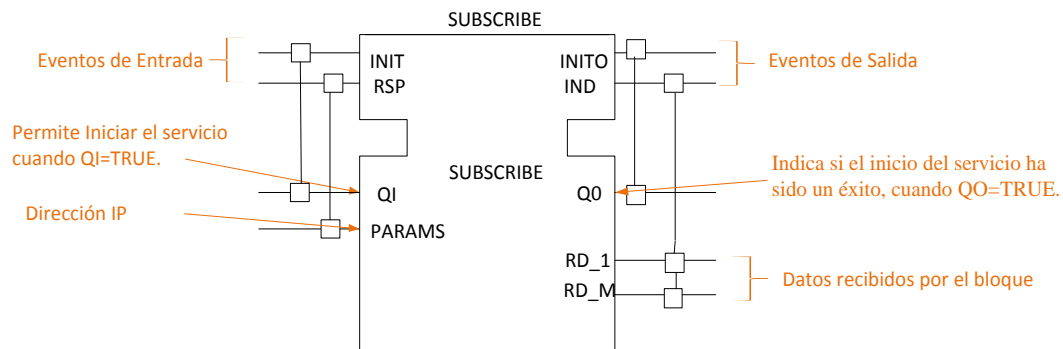
Fuente: Tomada y Modificada de [32]

Se empleara el bloque PUBLISH en caso de enviar datos a otros dispositivos, se definirá de la siguiente manera para los datos de entrada/salida:

- a. Entrada: La dirección IP del dispositivo, que será el encargado de enviar los datos
- b. Salida: Los datos a ser enviados

El bloque SUBSCRIBE representado en la *Figura 18* se utiliza para la recepción de datos [42].

Figura 18. Bloque SUBSCRIBE.



Fuente: Tomada y Modificada de [32]

En la *Tabla 6* se describen los eventos y datos del bloque de función SUBSCRIBE

Tabla 6. Eventos y datos del bloque SUBSCRIBE

Eventos Entrada	Datos Entrada
INIT: Al activarse este evento, inicia el servicio de recepción de datos.	QI: Esta entrada de datos requiere tomar el valor verdadero (True), para completar el inicio del servicio de recepción de datos.
RSP: Al activarse este evento, permite realizar la recepción de datos.	PARAMS: Esta entrada contiene los datos de direccionamiento de red.
Eventos Salida	Datos Salida
INITO: Este evento se activa cuando el bloque ha completado su inicialización.	QO: Esta salida al tomar el valor verdadero (True), indicara que la inicialización del bloque se realizó con éxito.
IND: Este evento al activarse indica que los datos han sido recibidos.	RD_1, RD_M: En esta salida se encuentran los datos recibidos.

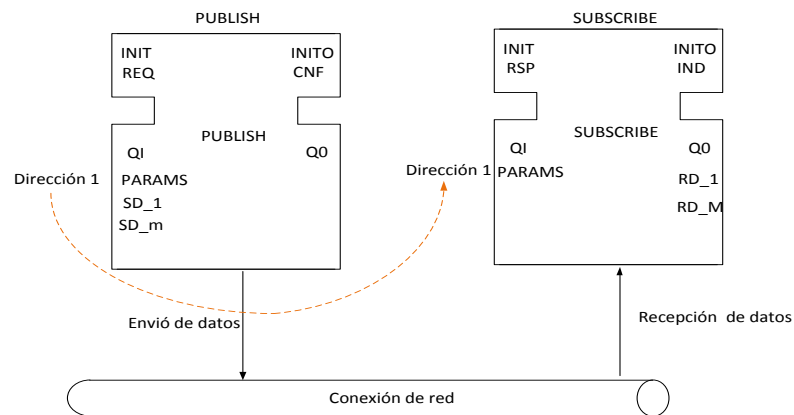
Fuente: Tomada y Modificada de [32]

Se empleara el bloque SUSCRIBE en caso de recibir datos de otros dispositivos, se definirá los siguientes datos de entrada/salida.

- a. Entrada: La dirección IP del dispositivo que envía datos
- b. Salida: Los datos recibidos, listos para utilizar.

Los Bloque de función de PUBLISH y SUBSCRIBE se implementan en diferentes dispositivos y se comunican a través de la conexión de red, como lo ilustra la *Figura 19*, para asegurar una adecuada comunicación entre estos dos bloques de función, el valor de la entrada PARAMS deben coincidir [42].

Figura 19. Comunicación de los bloques PUBLISH-SUBSCRIBE.



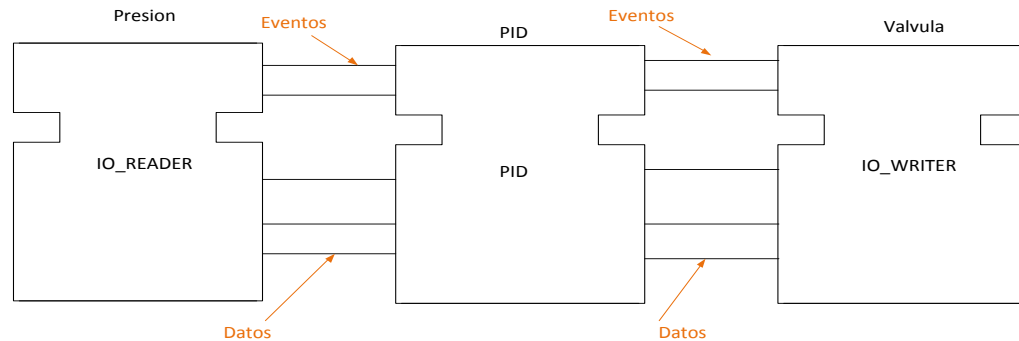
Fuente: Tomada y Modificada de [32]

A partir de los bloques de función especificados, se generan los modelos que da el estándar. A continuación se muestran y especifican cada uno de ellos.

2.1.4. Modelo de aplicación

Se define como una red de bloques de función interconectados, unidos por flujos de eventos y datos, con el fin de resolver un problema determinado en el control de la automatización, por ejemplo el conjunto de bloques utilizados para controlar la presión de un proceso. La *Figura 20* [46] representa el modelo de aplicación, el cual puede estar conformado por bloques básicos, interfaz de servicio o compuestos.

Figura 20. Modelo de aplicación definido por IEC 61499.



Fuente: Tomada y Modificada de [32]

2.1.5. Modelo de recurso

La *Figura 21*. Muestra las principales características de un recurso, definido por IEC 61499. El recurso está conformado por las Interfaces de comunicación y proceso, además de determinada aplicación. Cada recurso tiene la característica de ser independiente, esto implica que puede ejecutar bloques de función de forma separada, a continuación se describen las interfaces que integran el modelo de recurso [47].

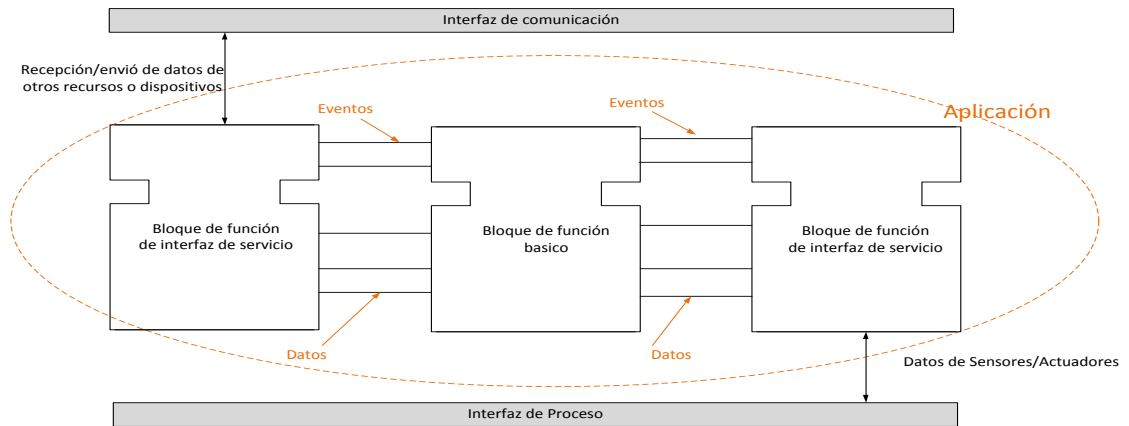
2.1.5.1. Interfaz de proceso

La interfaz de proceso le permitirá a los bloques recibir datos de sensores, y enviar datos a los actuadores [15].

2.1.5.2. Interfaz de comunicación

La Interfaz de comunicación se utiliza para enviar y recibir datos entre diferentes recursos [15].

Figura 21. Modelo de recurso definido por IEC 61499.

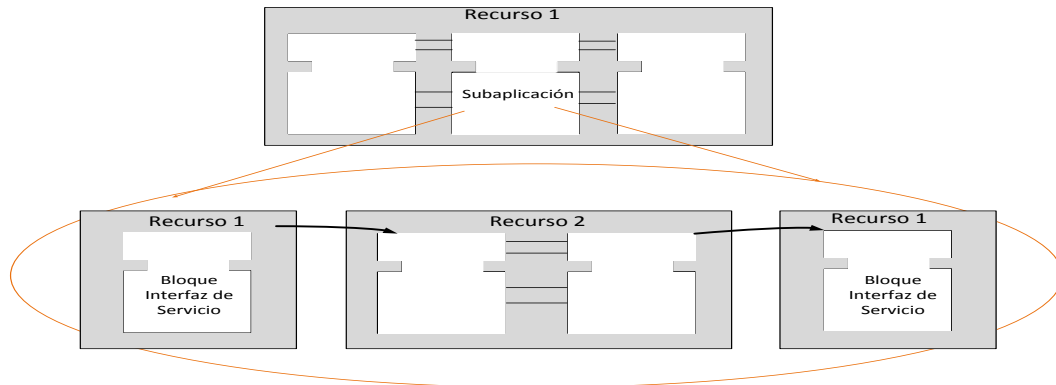


Fuente: Tomada y Modificada de [32]

2.1.6. Modelo de subaplicación

Una subaplicación puede considerarse como una forma especial de un bloque de función compuesto, que está diseñado para ser “distribuido”, es decir, puede encapsular bloques de más de un recurso [32], la Figura 22 representa el modelo de subaplicación.

Figura 22. Modelo de subaplicación definido por IEC 61499



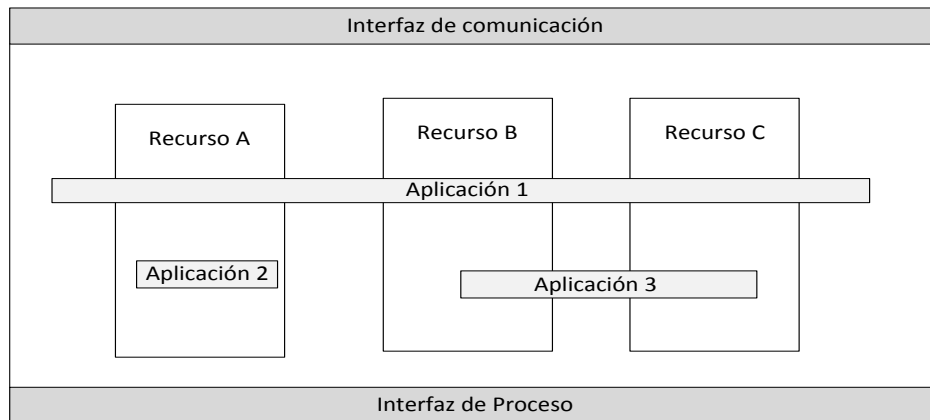
Fuente: Tomada y Modificada de [32]

2.1.7. Modelo de dispositivo

El modelo de dispositivo que se aprecia en la Figura 23. Representa la entidad física que va a ejecutar la red de bloques de función. Se podría asimilar a un

sistema embebido, PC, sensor inteligente, actuador inteligente. El dispositivo es capaz de comunicarse con otros dispositivos a través de interfaces de comunicación. Al mismo tiempo es igualmente capaz de comunicarse con el proceso físico a través de una interfaz de proceso [44].

Figura 23. Modelo de dispositivo definido por IEC 61499

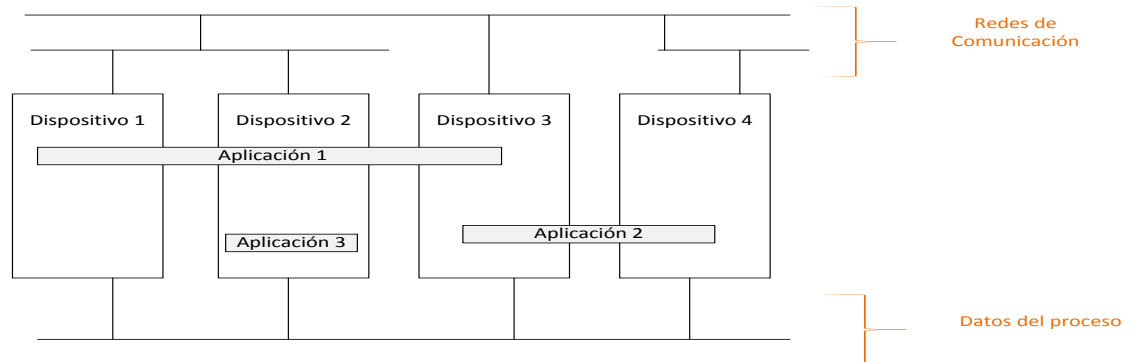


Fuente: Tomada y Modificada de [32]

2.1.8. Modelo del sistema

Para IEC 61499 un sistema de control distribuido se define como el conjunto de dispositivos conectados por medio de redes de comunicación. El modelo de sistema es el elemento de mayor nivel dentro de la arquitectura, engloba todos los dispositivos capaces de comunicarse y el conjunto de aplicaciones dentro del sistema DCS [44]. De acuerdo a la *Figura 24* una aplicación puede existir en un solo dispositivo o tener una funcionalidad distribuida en un número de dispositivos. Por ejemplo la aplicación 1, se utiliza en los dispositivos 1, 2 y 3. Mientras que la aplicación 3, solo será utilizada en el dispositivo 2.

Figura 24. Modelo de sistema definido por IEC 61499.



Fuente: Tomada y Modificada de [32]

2.1.9. Bloques de función de evento

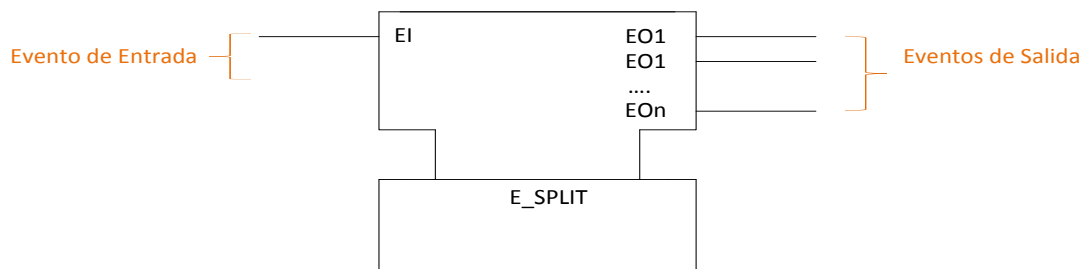
Los bloques de función de evento permiten iniciar el primer bloque de función en cada red de bloques de función, además de dividir los eventos para activar simultáneamente múltiples redes de bloques, entre otras necesidades [32].

Ahora se procede a revisar los tipos de bloques de función de eventos definidos en el estándar IEC 61499, que permiten una amplia gama de escenarios de eventos a modelar.

2.1.9.1. Evento E_SPLIT

El Evento E_SPLIT produce dos o más eventos de salida simultáneos sobre la recepción de un solo evento de entrada, como se muestra en la Figura 25.

Figura 25. Evento SPLIT.

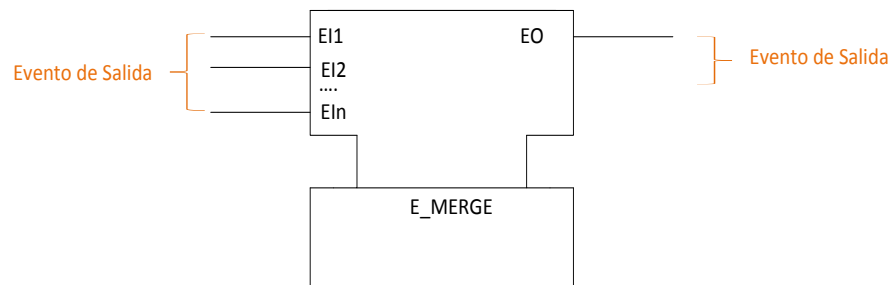


Fuente: Tomada y Modificada de [32]

2.1.9.2. Evento E_MERGE

El Evento E_MERGE produce un único flujo de eventos de salida mediante la fusión de los eventos entrantes en las entradas de evento EI1 a EIn. Esto puede ser utilizado para combinar dos o más eventos como se ve en la *Figura 26*.

Figura 26. Evento E_MERGE.

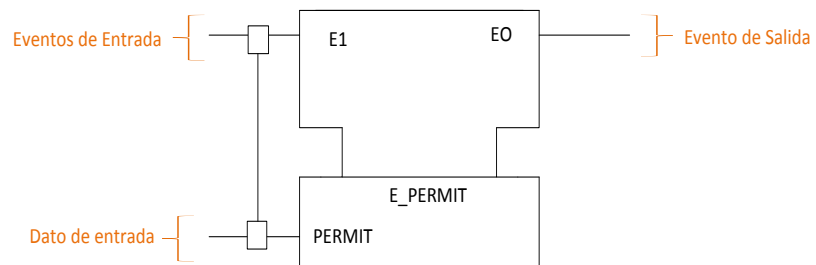


Fuente: Tomada y Modificada de [32]

2.1.9.3. Evento E_PERMIT

.La *Figura 27* representa el bloque de evento E_PERMIT, recibe el evento de entrada E1, cuando el valor del dato "PERMIT" es activado, permite activar el evento de salida E0.

Figura 27 Evento E_PERMIT

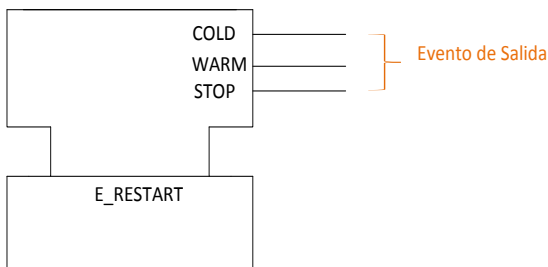


Fuente: Tomada y Modificada de [32]

2.1.9.4. Evento Restart

El Evento E_Restart proporciona eventos simples cuando el recurso arrancó en frío (COLD), caliente (WARM) y también cuando el recurso se detiene por algún agente externo (STOP), como se aprecia en la *Figura 28*. Es probable que la mayoría de las aplicaciones requieran al menos una instancia de este bloque para iniciar el evento disparador que inicia la ejecución de una cadena de bloques de función en una red.

Figura 28. Evento E_Restart.



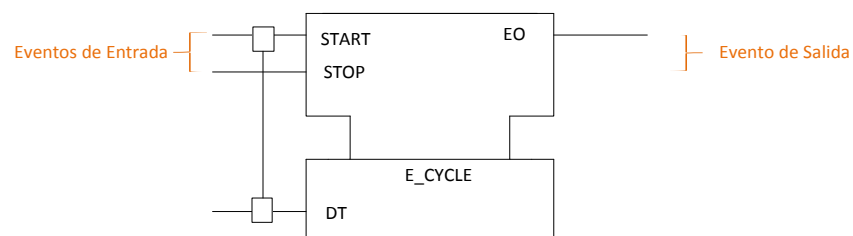
Fuente: Tomada y Modificada de [32]

2.1.9.5. Evento CYCLE

El Evento CYCLE es usado para producir un flujo de eventos a intervalos regulares.

La secuencia de eventos se inicia cuando un evento llega a la entrada START y se detiene cuando un evento llega a la entrada STOP. El período entre los eventos de salida está dado por el valor de DT *Figura 29*.

Figura 29. Evento E_CYCLE



Fuente: Tomada y Modificada de [32].

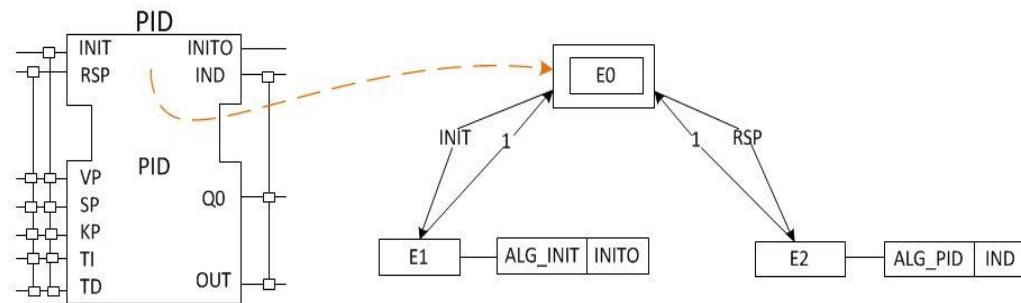
2.1.10. Ejemplo modelado de sistema de control de nivel.

Para el ejemplo de la *Figura 31*. Se considera un sistema de control de nivel, el modelo del sistema propuesto por el estándar según lo especificado en este ejemplo contiene; dos modelos de dispositivos (Dispositivos A, Dispositivo B), cada uno de los dispositivos contiene un modelos de recurso, es decir, encontramos un recurso por cada dispositivo, en este caso tenemos el recurso A1 para el modelo de dispositivo A y el recurso B1, para el modelo de dispositivo B.

Se emplea un modelo de aplicación para la implementación de un tipo de control PID, la cual está compuesta por los siguientes bloques.

- El bloque interfaz de servicio SENSOR (color azul claro) que es una instancia de IO_READER, es el encargado de leer la señal del sensor de nivel que viene desde la planta a controlar.
- El bloque de función PID (color azul claro), proporciona el algoritmo de control que ajusta el nivel del recipiente o TK1. Cuenta con dos eventos de entrada "INIT" y "RSP". El evento "INIT" se utiliza para inicializar las variables internas utilizadas por el algoritmo PID, al finalizar este proceso se activara el evento "INIT0". El evento "RSP" provoca la ejecución del algoritmo PID, esto hace que el algoritmo pueda tomar los últimos valores para la variable del proceso (VP), el punto de ajuste (SP) y calcular el nuevo valor de salida para accionar la válvula, cuando el algoritmo PID ha terminado, el evento de salida "IND" se produce junto con la salida "OUT", la cual entrega el valor calculado por el PID siendo Q0 un dato de validación, es decir si Q0 es igual a TRUE el algoritmo ha termino. Lo anterior se ve reflejado en el diagrama EC de la *Figura 30*. En este bloque se asume que los valores ingresados para los datos SP, KP, TI, TD son constantes.
- El bloque interfaz de servicio VALVULA (color verde) que es una instancia de IO_WRITER, es el encargado de recibir la señal entregada por el bloque PID.

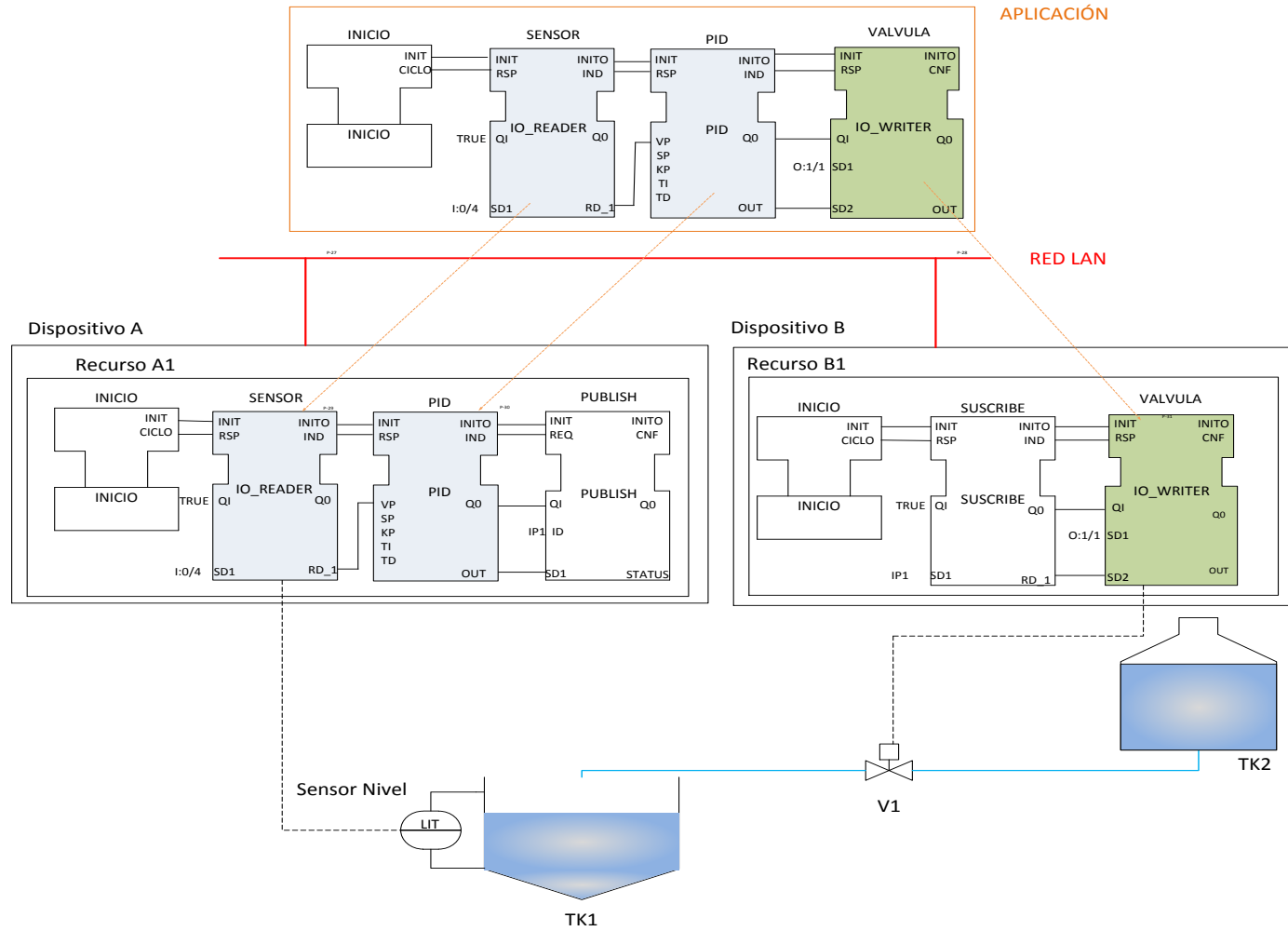
Figura 30. Diagrama de Control de Ejecución bloque PID



Fuente: Tomada y Modificada de [32]

Finalmente se distribuye la aplicación entre los dispositivos (ver Figura 31), por medio de sus recursos, el dispositivo A recibe de la aplicación los bloques SENSOR y PID, el bloque PUBLISH, envía el valor calculado por el bloque PID al dispositivo B, cual recibe el valor mediante el bloque SUBSCRIBE, y le asigna el valor recibido al bloque VALVULA. El bloque inicio es el encargado de inicializar la secuencia de eventos requerida por cada dispositivo.

Figura 31. Ejemplo de control de nivel con el modelado de IEC 61499



Fuente: Autores.

CAPÍTULO III

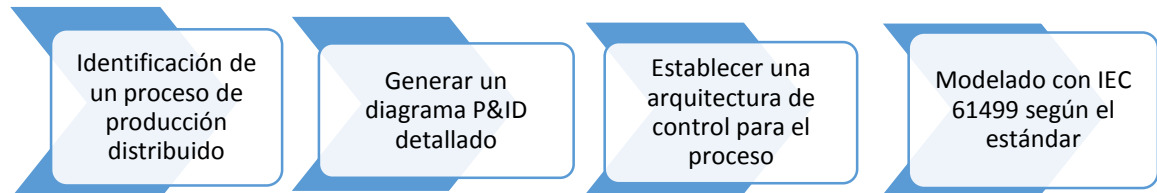
3. PROCEDIMIENTO PARA LA IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL DISTRIBUIDO BAJO EL ESTÁNDAR IEC 61499-1.

En el desarrollo temático de este capítulo se presenta la propuesta de un procedimiento para un sistema distribuido bajo el estándar IEC 61499-1, se resume en lograr identificar los bloques necesarios para el proceso que se desea realizar, disminuyendo el tiempo de planeación e implementación de un sistema de control distribuido. La identificación del proceso es importante para el procedimiento.

3.1. Procedimiento para la identificación de los elementos necesarios en una aplicación de control distribuido con IEC 61499.

Se realiza el procedimiento general de la *Figura 32* para la identificación de la arquitectura de control, modelos del proceso y bloques de función necesarios en la implementación de un sistema de control distribuido bajo el estándar IEC 61499-1.

Figura 32. Procedimiento para la identificación de los elementos.



Fuente: Los Autores

Para la aplicación del procedimiento se deben tener en cuenta dos aspectos importantes:

1. Si es un proceso que está establecido, debe tener documentación y diagramas de ingeniería que ayuden al análisis e identificación de sus funciones para mejorar el proceso y distribuir dichas funciones. De no ser así no se podrá aplicar el procedimiento.
2. Si se parte con un proyecto nuevo y se está en el punto de querer desarrollar e innovar con nuevas tecnologías y sistemas distribuidos, deberá cumplir con las características que se muestran en la *Identificación de un proceso distribuido*

3.1.1. Identificación de un proceso distribuido.

Para poder identificar si el proceso que está establecido es candidato a implementar el procedimiento debe cumplir y tener las siguientes características:

1. Descripción del proceso, observar la secuencia o flujo del proceso y reconocer las posibles etapas que contenga.
2. Identificar qué tipo de proceso es, continuo, BATCH, discreto.
3. NO estar geográficamente distribuido, es decir, que todas las etapas del proceso de producción se den dentro de la planta de producción
4. Identificar si el proceso se puede dividir en subprocesos o unidades y observar las operaciones que hacen parte de él.

Si el proceso es nuevo y se piensa implementar de acuerdo a los lineamientos de este procedimiento, deberá cumplir o desarrollar las siguientes actividades:

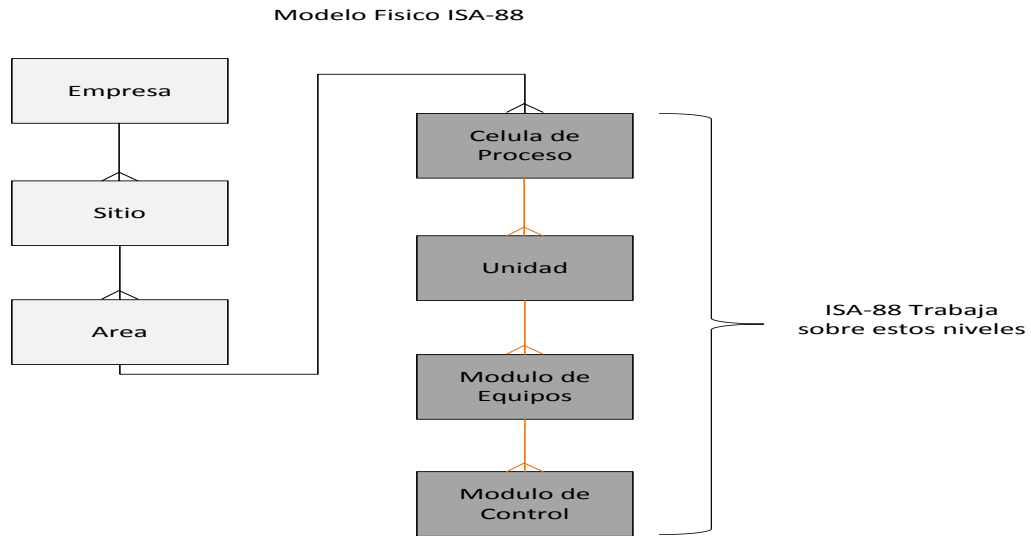
1. Describir el proceso, tratando de organizar la secuencia y reconociendo las posibles etapas que contenga.
2. El proceso debe cumplir con que sea Continuo, BATCH o Discreto.
3. Ubicar el proceso dentro de la empresa, es decir, que la elaboración y puesta en marcha no necesite o depende de ubicar dispositivos en lugares remotos o por fuera del área física que integra la fábrica.
4. Dividir el proceso en subprocesos o unidades. Se propone el uso del estándar ISA S88.01 [48] para llevar a cabo este ítem. El modelo a utilizar para obtener las unidades o célula de proceso como la describe el estándar "ISA S88.01" es el siguiente:

Modelo físico de ISA-S88.01

El modelo físico se puede utilizar para describir los activos físicos de una empresa en términos de empresa, sitio, área, célula de proceso, unidades, módulos de equipos y módulos de control. Se organiza generalmente de forma jerárquica tal como se describe en la *Figura 33*. El modelo cuenta con siete niveles, comenzando en la parte superior con una empresa, un sitio y un área. Estos tres niveles se definen con frecuencia por consideraciones comerciales y no se modelan. Los tres niveles superiores son parte del modelo para identificar correctamente la relación de los equipos de nivel

inferior a la empresa de fabricación. A partir de la célula de proceso se empieza a modelar y es allí donde ISA S88.01 hace su enfoque.

Figura 33. Modelo Físico de ISA-S88.01



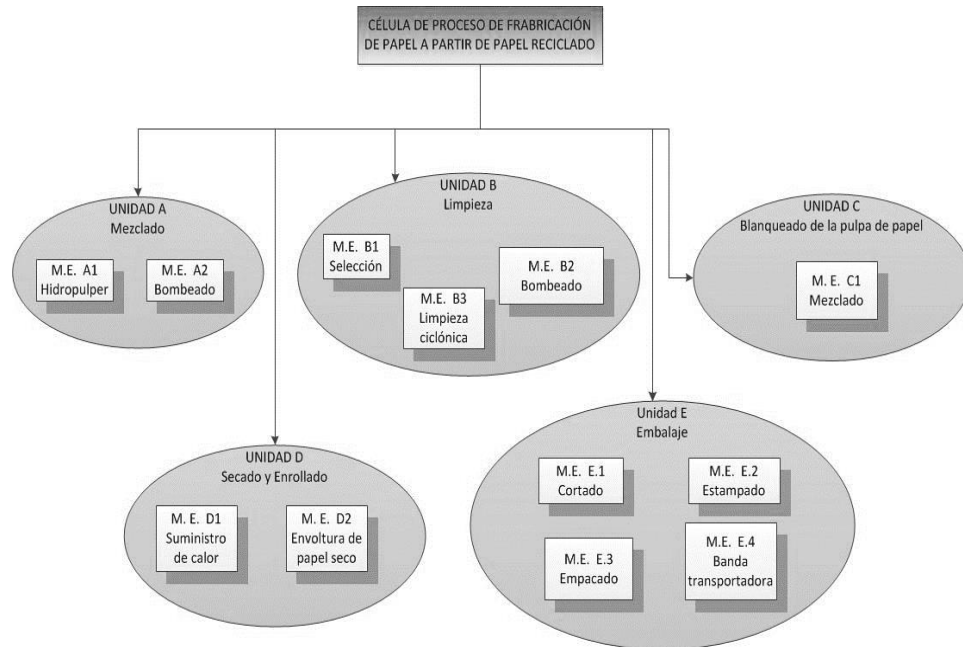
Fuente: Tomada y modificada de [49]

Después de conocer el esquema del modelo físico, será de gran utilidad llegar hasta el punto de especificar las unidades que contiene la célula de proceso. Estas unidades son la correlación con los subprocesos. Obtenida la división, se puede argumentar que el sistema es distribuido y que presenta una distribución de sus variables de control.

Célula de proceso

Una célula de proceso contiene todas las unidades o subproceso, módulos de equipos y módulos de control necesarios. Ese es el punto que se busca para saber que el proceso es distribuible. Para mayor claridad la *Figura 34* muestra un ejemplo de la célula de proceso en los que se observa claramente cada una de sus unidades (subproceso). Esta división se realizó para una fábrica que produce papel Tissue a base de papel reciclado. Contiene la Célula con cinco unidades (Mezclado, limpieza, blanqueado, secado, enrollado y embalaje)

Figura 34. Modelo físico ISA-S88.01. Célula.



Fuente: Tomada y modificada de [49]

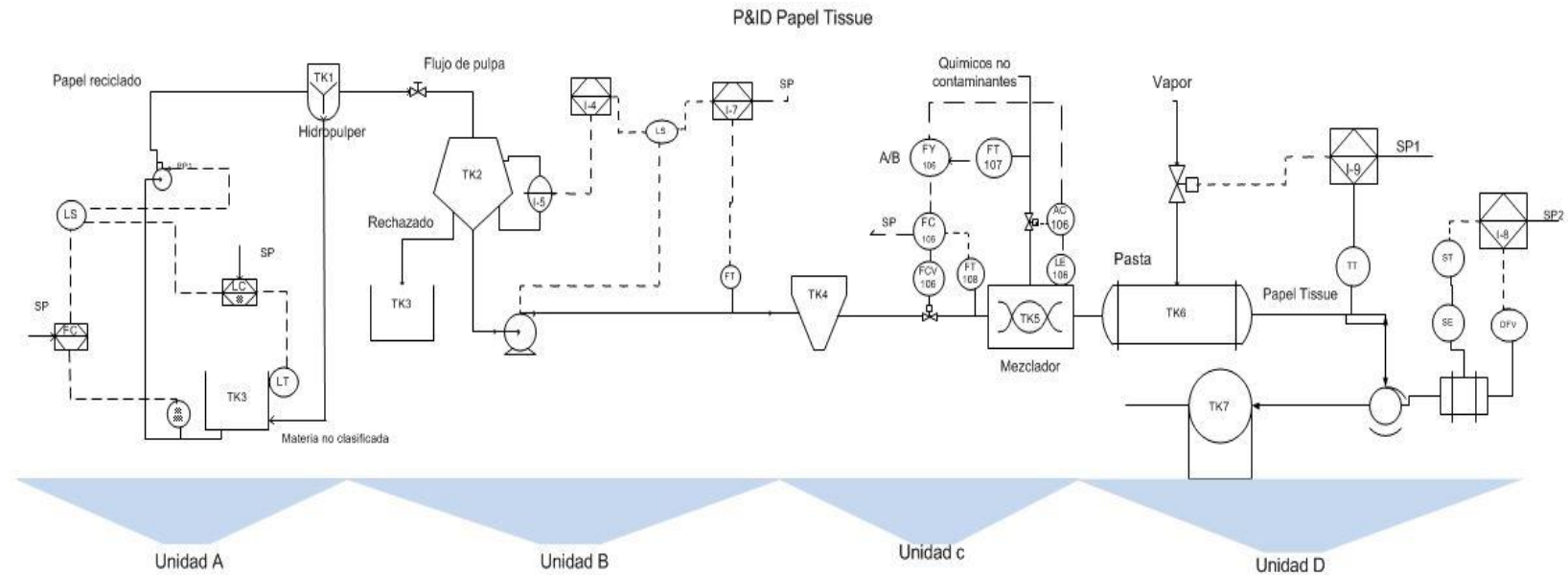
El siguiente paso, es la elaboración del diagrama PI&D, diagrama que ofrece ventajas para seguir desarrollando el procedimiento que está enfocado en la implementación del estándar en un proceso industrial.

3.1.2. Elaboración del diagrama P&ID.

Un P&ID es lo que se define como un diagrama de tuberías e instrumentación, muestra el flujo del proceso en las tuberías, así como los equipos instalados y la instrumentación. Estos diagramas están compuestos por símbolos estandarizados que permiten identificar todos los componentes que conforman un proceso, como tuberías, número de líneas de tuberías, dimensiones, válvulas, controladores, alarmas, equipos, niveles, drenajes entre otros, se elabora apoyado por la familia de estándares ISA S5.

Por ejemplo se tiene el diagrama P&ID, *Figura 35*. La unidad de mezclado del proceso de fabricación de papel Tissue.

Figura 35. Diagrama P&ID papel Tissue



Fuente: Tomada y modificada de [49]

Es de vital importancia realizar un buen y detallado diagrama P&ID. Con este diagrama se aproxima para comenzar a relacionar los modelos del estándar IEC 61499. La manera en que se organice, permitirá ver los subprocesos o unidades, los elementos de control, sensores y actuadores que contiene el proceso. Establecido este esquema se podrá analizar la arquitectura y en base en ello, los modelos. Por ejemplo; para el caso de la unidad de mezclado y según el diagrama P&ID *Figura 35*. De ahí, el subproceso de mezclado contiene tres controladores, por cada uno hay a lo menos un sensor, basados en esto y en el análisis hecho sobre el estándar (IEC 61499) se establecerá más adelante que depende de las variables de control y de los controladores vistos en este ítem el modelo de dispositivo de IE61499

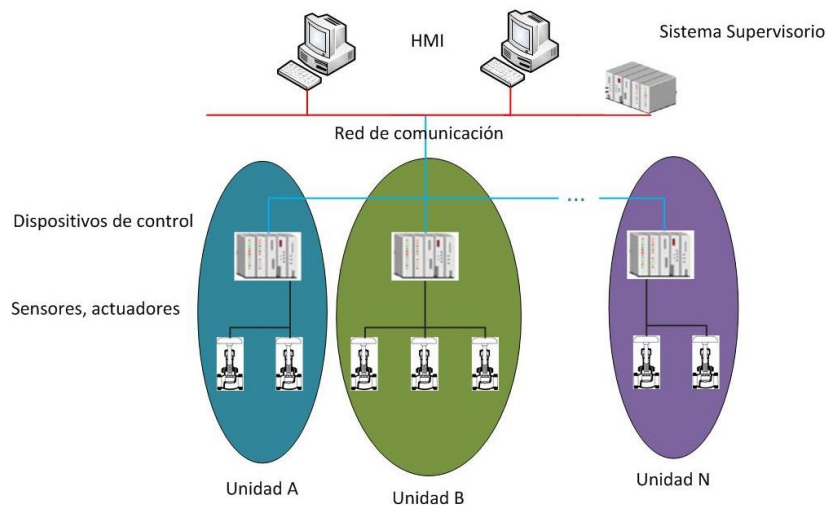
3.1.3. Establecer la arquitectura de control distribuido para el proceso de producción.

De acuerdo al criterio que genera el estándar IEC 61499 para un arquitectura de control distribuido. Estableciendo que los elementos o equipos deberán estar interconectados por medio de una red de comunicación, basándose en las características que ofrece el diagrama P&ID detallado. A continuación se muestran los componentes básicos con los que debe contar una arquitectura de control distribuido, elementos que en la mayoría de los casos son identificables en el diagrama P&ID. Los demás componentes hacen parte de los equipos de comunicación.

- Elementos de control. (Dispositivos de control)
- Red de comunicación. (Bus de campo, Red de campo)
- Sensores y actuadores. (Instrumentos de campo)

La *Figura 36* muestra un ejemplo de los elementos que contiene una arquitectura y deja ver la relación con el diagrama P&ID, buscando especificar las unidades y observando la distribución de las variables de control en cada uno de los procesadores.

Figura 36. Elementos principales de una arquitectura distribuida

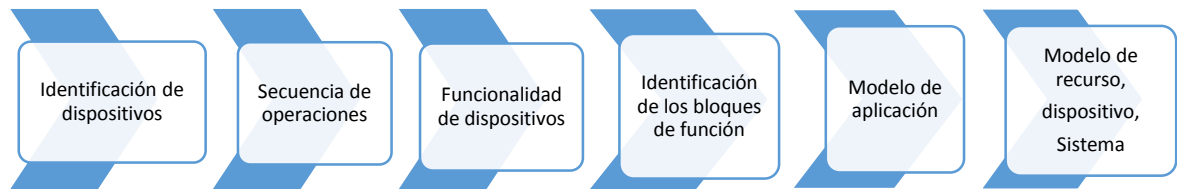


Fuente: Autores

3.1.4. Modelado con IEC 61499 según el estándar.

A partir de los casos de estudios empleados por [50],[51],[15], se identificaron criterios para generar los diferentes modelos aportados por IEC 61499. La *Figura 37* describe una serie de pasos, que permitirán al sistema ser modelado por el estándar y generar los bloques de función que son necesarios en la aplicación del proceso.

Figura 37. Modelamiento con IEC 61499



Fuente: Autores

3.1.4.1. Identificación de dispositivos.

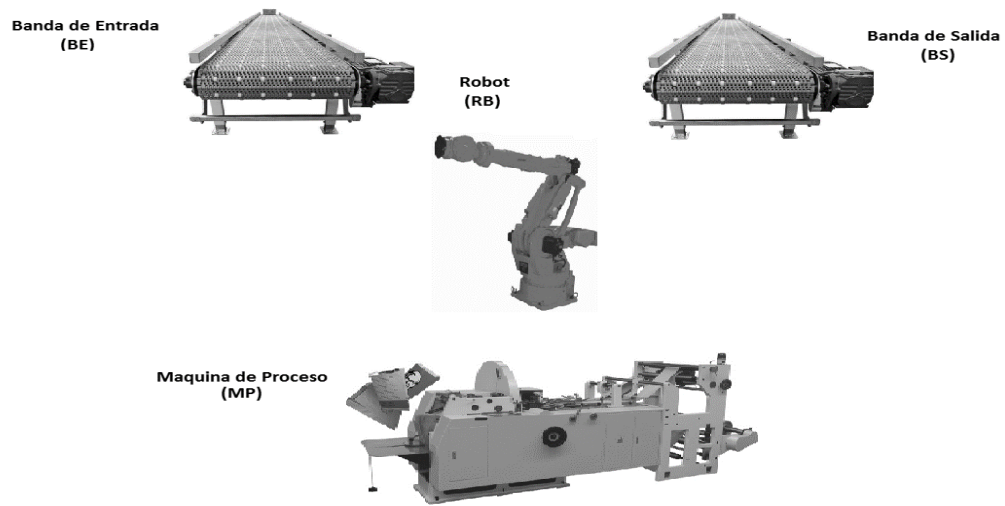
A partir de la arquitectura desarrollada, se identifican los dispositivos que componen el proceso: PLCs, Robots, PCs, CNC, HMI Programables, Sensores Y Actuadores Inteligentes, es decir cualquier elemento de la arquitectura que posea un procesador.

3.1.4.2. Secuencia de operaciones.

A partir de los dispositivos identificados se define una secuencia deseada de operaciones, así como las posibles secuencias anormales que pueden ocurrir. Esto se puede hacer de manera informal, o mediante el uso de notaciones más formales, como las redes de Petri (RdP), Gráficas de funciones secuenciales (SFC). A continuación un ejemplo que permitirá por medio de RdP obtener la secuencia de operaciones.

- **Modelo de brazo de robot.** Este ejemplo considera un sistema de manufactura (*Figura 38*) donde se tiene una banda transportadora por donde arriban piezas para ser procesadas, estas son tomadas por un brazo de robot que las deposita en una máquina para su procesamiento. Al finalizar el proceso, el brazo de robot vuelve en una banda de salida [52].

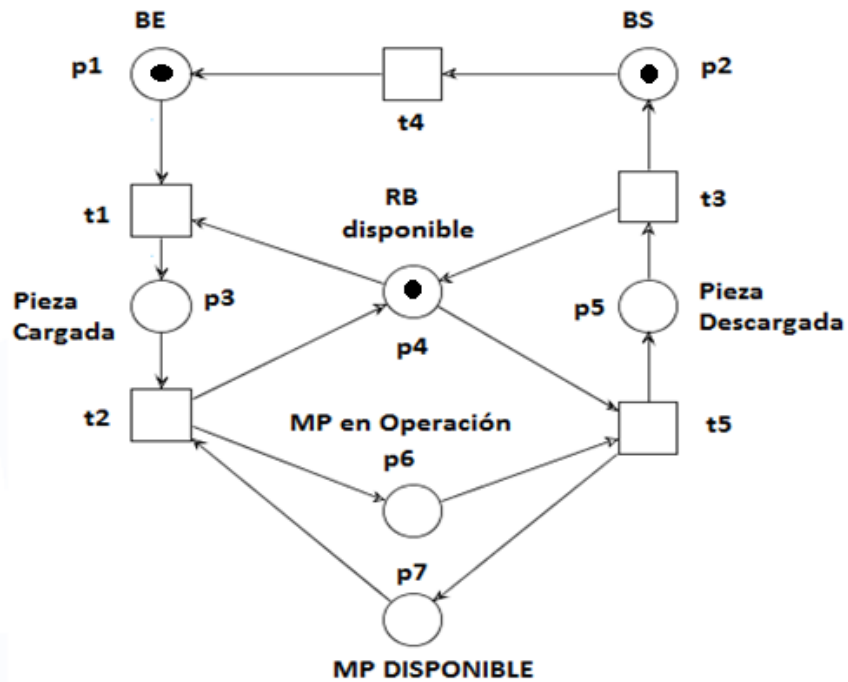
Figura 38. Sistema de manufactura flexible



Fuente: Tomada y Modificada de [52]

La Red de Petri correspondiente al sistema se presenta en la Figura 39. Los estados del sistema se denotan como lugares de la Red de Petri.

Figura 39. Red de Petri correspondiente a los estados del sistema de manufactura flexible



Fuente: Autores

3.1.4.3. Funcionalidad de dispositivos.

La secuencia de operaciones del sistema permite identificar las funciones que cumple cada dispositivo. A continuación se describen algunas de estas funciones:

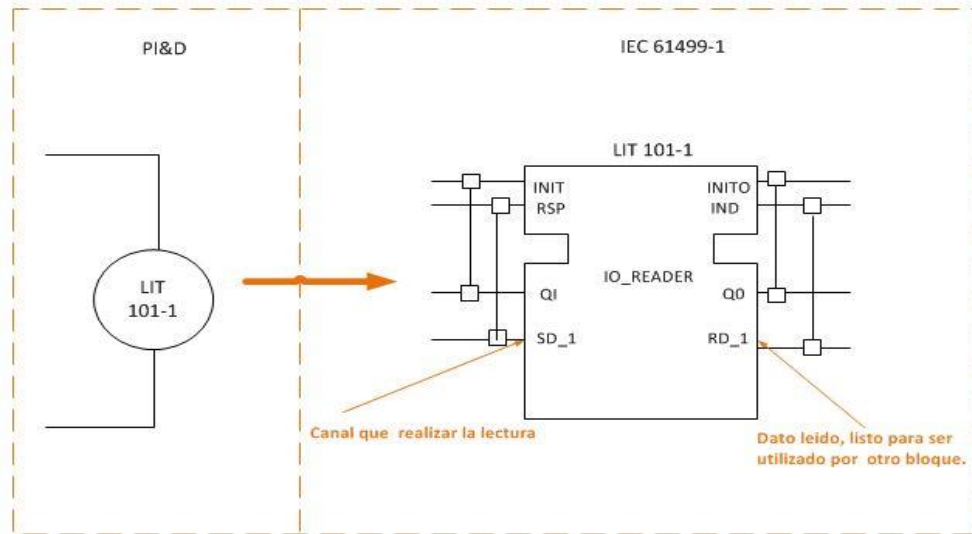
- *Control PID*: Empleado en caso de controlar una variable (Presión, temperatura, nivel) de forma continua.
- *Control ON/OFF*: el controlador ON-OFF o de dos posiciones es empleado para mantener el valor de la variable manipulada en dos estados.
- *Supervisión y Control*: Son funciones realizadas comúnmente por las HMI, se resaltan algunas de ellas; visualizar variables, supervisar niveles de alerta, monitoreo de variables, entre otras.
- *Funciones específicas*: Son las operaciones específicas que realiza un dispositivo, por ejemplo, el fresado para una maquina CNC.

3.1.4.4. Identificación de los bloques de función.

A continuación se especificaran los bloques de función empleados en cada dispositivo de acuerdo a la funcionalidad establecida y basándose en los elementos que el dispositivo necesita para cumplirla, de acuerdo al diagrama PI&D.

- *Bloques de eventos*: Cada dispositivo definido dentro de la arquitectura tendrá como mínimo un bloque de evento de inicio, es decir, el bloque E_RESTART (*ver Figura 28*. Evento E_Restart.) y otro para realizar la ejecución cíclica de los bloques, es decir el bloque E_CYCLE (*ver Figura 29*. Evento E_CYCLE)
- *Bloques de Entrada*: Por cada sensor que tenga asociado el dispositivo, tendrá un bloque de función IO_READER, la *Figura 40* muestra la identificación del bloque de función, donde se tendrán los siguientes datos principales de entrada/salida
 - c. Entrada: El canal que realizara la lectura.
 - d. Salida: El valor del sensor leída.

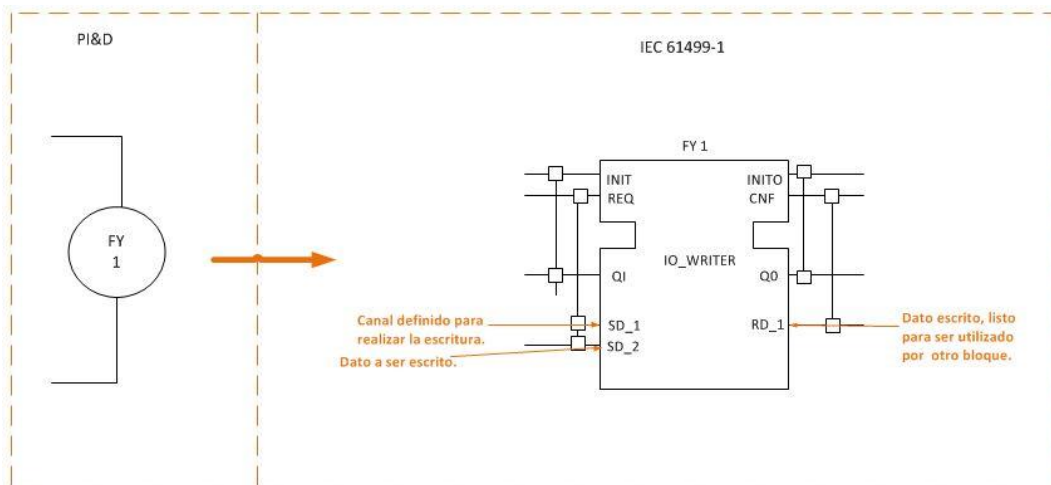
Figura 40. Identificación del bloque IO_READER



Fuente: Autores

- **Bloques de salida:** Por cada actuador (válvulas, bombas, variador, etc.) que tenga asociado el dispositivo tendrá un bloque de función IO_WRITER la Figura 41 muestra la identificación del bloque de función, donde se tendrán los siguientes datos principales de entrada/salida.
 - Entrada: El canal de salida empleado y definir la señal de salida empleada
 - Salida: La señal de salida disponible para utilizar.

Figura 41. Identificación del bloque IO_WRITER.



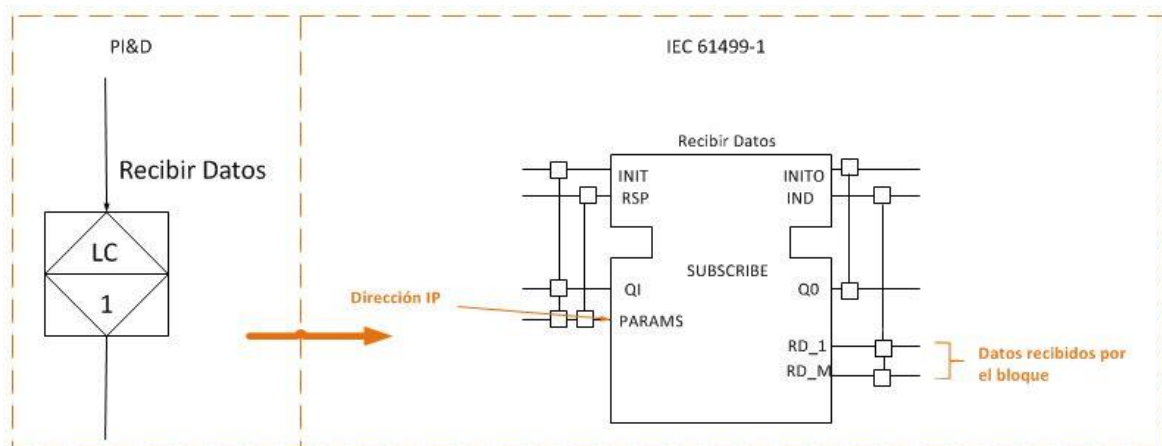
Fuente: Autores

- **Bloques de función de acuerdo a la funcionalidad:** Para realizar el control y funciones específicas que requieran cada dispositivo se emplearan bloques de función básicos. En estos bloques se determinan lo siguiente:
 - a. La encapsulación de la funcionalidad del dispositivo se debe generar mediante bloques de función básicos o bloques de función compuestos.
 - b. Definir los diagramas de control de ejecución, algoritmos, los lenguajes de programación (*ver Figura 11. Gráfico de control de ejecución, definido por IEC 61499.*) que se emplearan en los bloques, de acuerdo a la secuencia de operaciones.

Ver la Figura 30. Diagrama de Control de Ejecución bloque PIDSirve como ejemplo para la funcionalidad del dispositivo, está basada en realizar el control PID al proceso.

- **Bloques de Recepción de datos:** El dispositivo tendrá bloques de función SUSCRIBE, la *Figura 42. Identificación del bloque SUSCRIBE en caso de recibir datos de otros dispositivos, se definirá los siguientes datos de entrada/salida.*
 - c. Entrada: La dirección IP del dispositivo que envía datos
 - d. Salida: Los datos recibidos, listos para utilizar.

Figura 42. Identificación del bloque SUSCRIBE



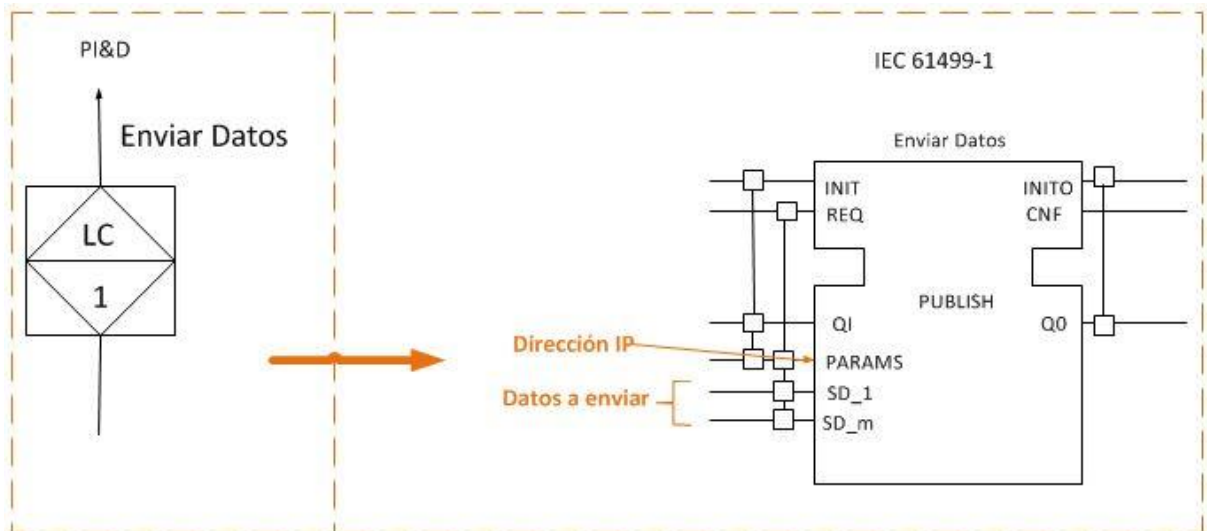
Fuente: Autores

- **Bloques de Envío de datos:** El dispositivo tendrá bloques de función como los de la *Figura 43. La Identificación del bloque PUBLISH en caso de enviar*

datos a otros dispositivos se definirá de la siguiente manera para los datos de entrada/salida:

- c. Entrada: La dirección IP del dispositivo, que será el encargado de enviar los datos
- d. Salida: Los datos a ser enviados

Figura 43. Identificación del bloque PUBLISH



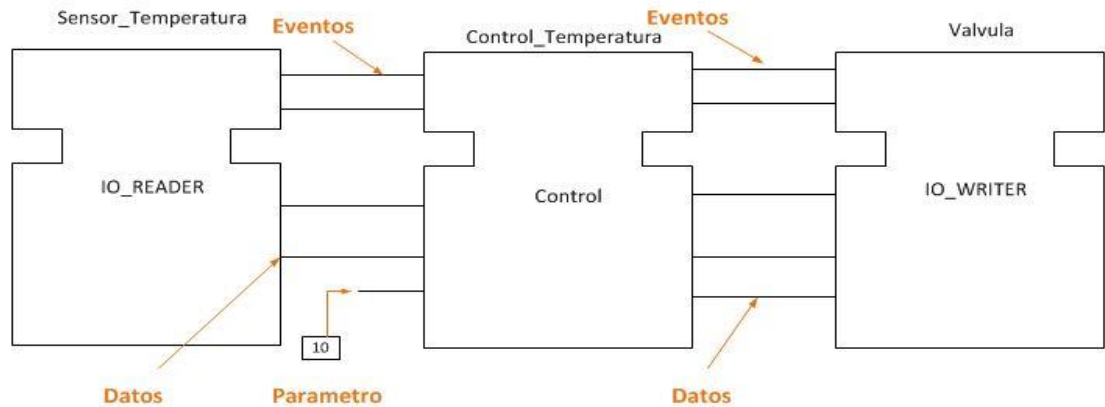
Fuente: Autores

3.1.4.5. Modelo de aplicación

El modelo de aplicación estará orientado a solucionar el control para determinado proceso, constara de la unión de los bloques identificados en el ítem anterior (*Identificación de los bloques de función*), por medio de datos, eventos y la definición de parámetros⁹ requeridos, basados en la secuencia de operaciones, a nivel de ejemplo la *Figura 44*. Describe el modelo de aplicación empleado para realizar el control de temperatura.

⁹ Un parámetro es un tipo de valor específico y constante

Figura 44. Elementos que componen el modelo de aplicación definido por IEC 61499



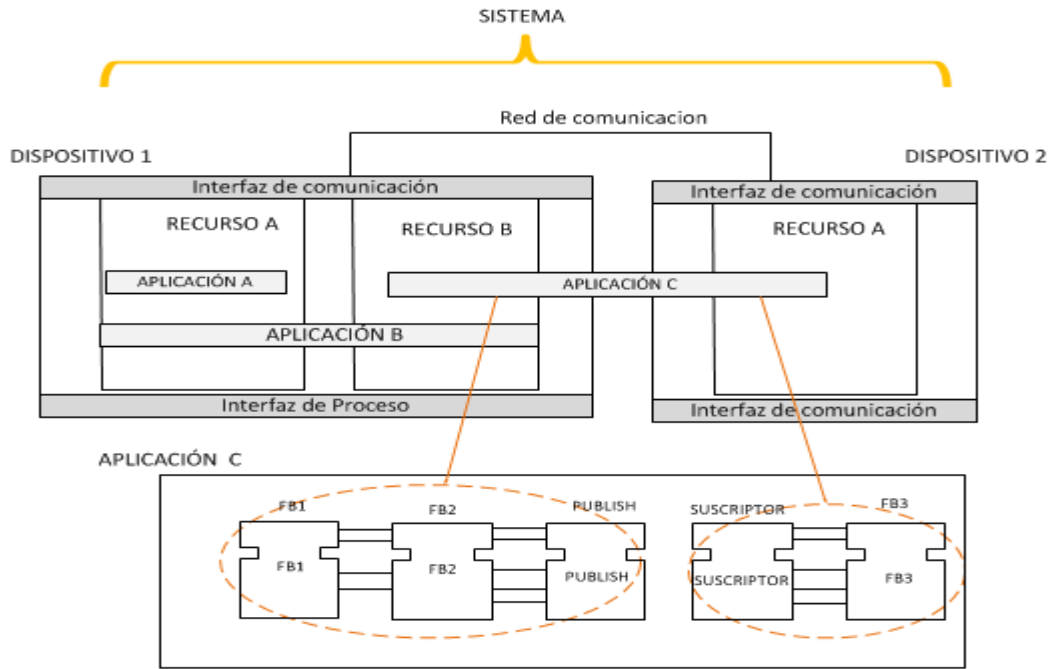
Fuente: Autores

3.1.4.6. Modelo de Recurso, Dispositivo, Sistema.

A continuación se indican los pasos a seguir para definir el modelo de recurso, dispositivo, sistema, teniendo como referencia la figura *Figura 45*

- A. Definir que parte de las aplicaciones desarrolladas tendrá cada dispositivo, de acuerdo a la funcionalidad identificada, es decir los diferentes controles realizados al proceso por el dispositivo.
- B. Las aplicaciones tendrán las siguientes características:
 - Podrán estar definidas dentro de uno o más recursos dentro de un mismo dispositivo.
 - Podrán estar definidas dentro de dos o más recursos, en diferentes dispositivos, para este caso se emplearán bloques publicador/suscriptor.
- C. De acuerdo al grado de independencia de las aplicaciones en un dispositivo, este podrá presentar varios recursos.
- D. Seleccionar el protocolo de comunicación para el envío de información entre los dispositivos, por medio de los bloques PUBLISH/SUSCRIPTOR, además de identificar las respectivas direcciones IP empleadas. El estándar propone el protocolo UDP/IP[51]
- E. El sistema estará representado por la interconexión de los diferentes dispositivos por medio de una red de comunicación.

Figura 45. Modelo de sistema, dispositivo, recurso definido por IEC 61499



Fuente: Autores

CAPITULO IV

4. APLICACIÓN CONCEPTUAL DEL PROCEDIMIENTO DESARROLLADO PARA UN PROTOTIPO EN EL LABORATORIO DEL PIAI

De acuerdo al procedimiento que se generó, se propone aplicarlo en un caso de estudio enfocado en el laboratorio de Control de Procesos de la Universidad del Cauca. Para llevar a cabo lo anterior se seleccionara un proceso industrial distribuido, el cual a partir de sus característica será representado por las plantas con que cuente el laboratorio, esto con el fin de obtener un sistema de control distribuido, sistema que permitirá realizar la implementación del procedimiento descrito en el CAPITULO III, el cual permite identificar cada uno de los pasos y las características del modelado con IEC 61499.

4.1. Estado de las plantas del laboratorio del PIAI

El proceso seleccionado es la producción de Zumo de naranja, proceso que presenta 3 subprocesos (térmico, nivel y mezcla de líquidos), a partir de estos se realiza la identificación de las posibles plantas ubicadas en el laboratorio que cumplen con los subprocesos mencionados.

De acuerdo a un análisis detallado de las plantas que posee el laboratorio; el cual permitió establecer el estado actual de cada una de las plantas y la parte funcional que presentan, se seleccionaron las siguientes plantas con su respectivo subproceso: Multivariable subproceso Nivel, sistemas a eventos discretos subproceso mezcla de líquidos, temperatura. Subproceso térmico.

Se procede a realizar la identificación de cada una de las plantas seleccionadas, posteriormente se realizara la descripción del proceso elegido.

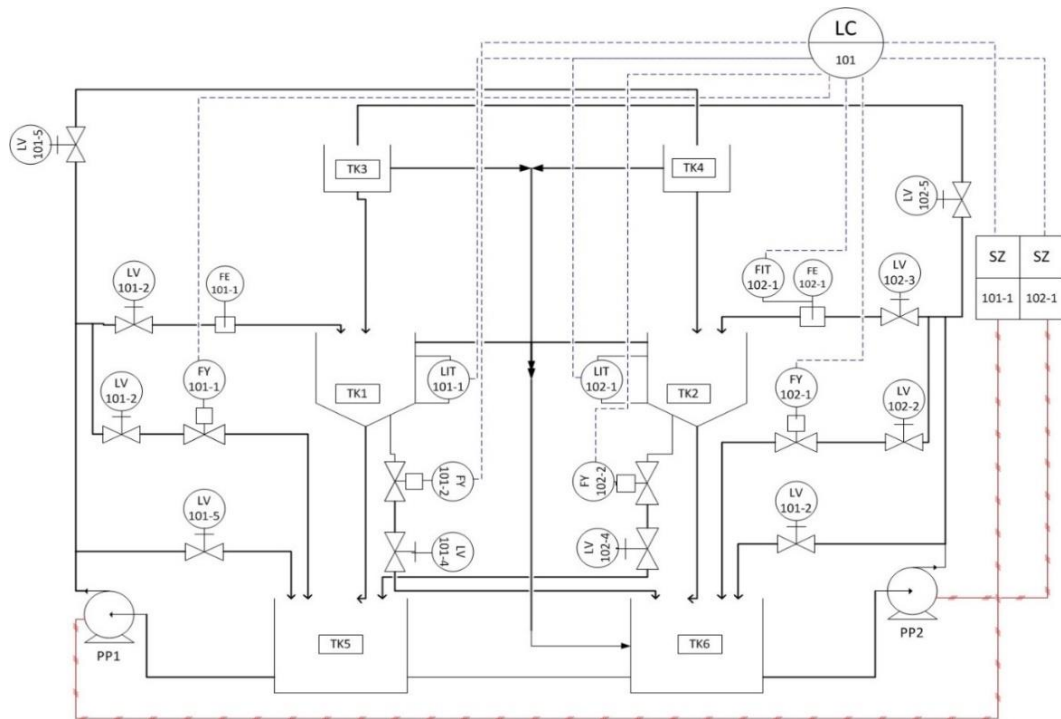
4.1.1. Planta Multivariable.

La planta Multivariable se encarga de controlar el nivel en dos tanques, por medio de las variables manipuladas caudal, está formada por 4 tanques interactuantes, dos de área constante y dos de área variable, sensores de flujo, sensores de nivel, electroválvulas, válvulas manuales, variadores para modificar la frecuencia generada por las dos motobombas[53]

La *Figura 46* representa el diagrama PI&D de la planta, la *Figura 47* indica la estructura física de la planta identificando componentes del diagrama PI&D. El nivel de los tanques es controlada por las motobombas, de acuerdo al nivel

requerido, el flujo de cada motobomba aumentara o disminuirá. El sistema se encuentra conectado a un panel de control (*Figura 48*), compuesto por: Instrumentación que se encarga de recibir las señales de los transmisores, así como de activar las motobombas que impulsan el caudal hacia los tanques. La cantidad de caudal entregado al sistema es controlado por los variadores de velocidad monofásicos, las señales de control son enviadas a través del Pc por medio de una tarjeta de adquisición de datos y una tarjeta de interfaz de comunicación con el proceso, se usan relés de estado sólido para convertir las señales de la tarjeta de adquisición, estas solo manejan 5V, mientras que las electroválvulas operan a 110V [53]

Figura 46. Diagrama PI&D planta Multivariable



Fuente: Autores

En la *Figura 47* se observa la planta multivariable que hay en el laboratorio de la Universidad, se hace una aproximación de los elementos de control, sensores y actuadores con los que cuenta

Figura 47. Planta Multivariable elementos de campo



Fuente: Tomada y Modificada de [53].

Figura 48. Planta Multivariable elementos de control



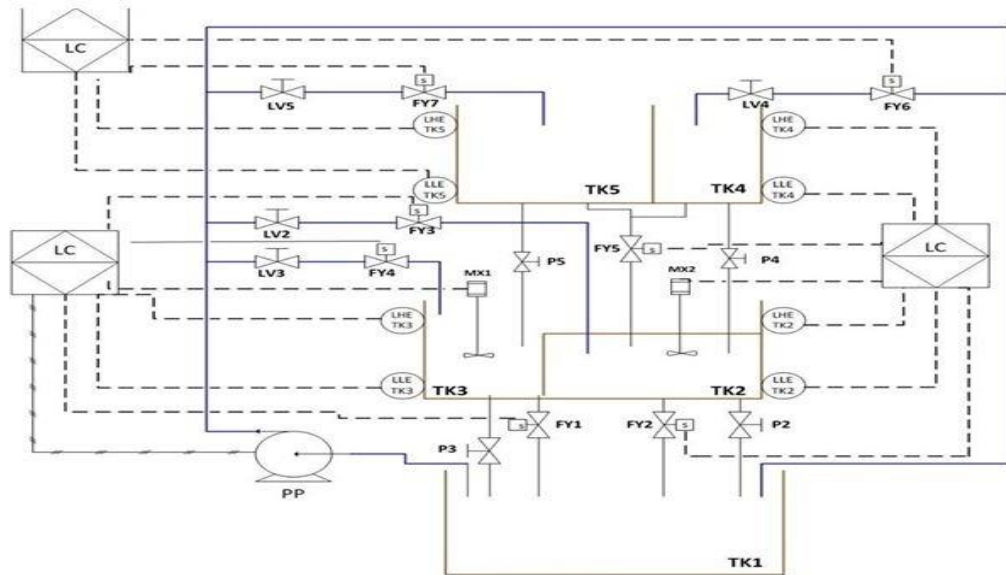
Fuente: Tomada y Modificada de [53]

4.1.2. Planta Sistema a Eventos Discretos.

La planta de Sistemas a Eventos Discretos (SED) es un sistema compuesto por una motobomba, un circuito hidráulico, electroválvulas, agitadores, detectores de nivel y varios tanques de almacenamiento (Figura 50), conectados a un panel de control (Figura 51). La planta SED está diseñada para emular un proceso de mezcla de sustancias. El propósito de la planta es permitir la implementación de

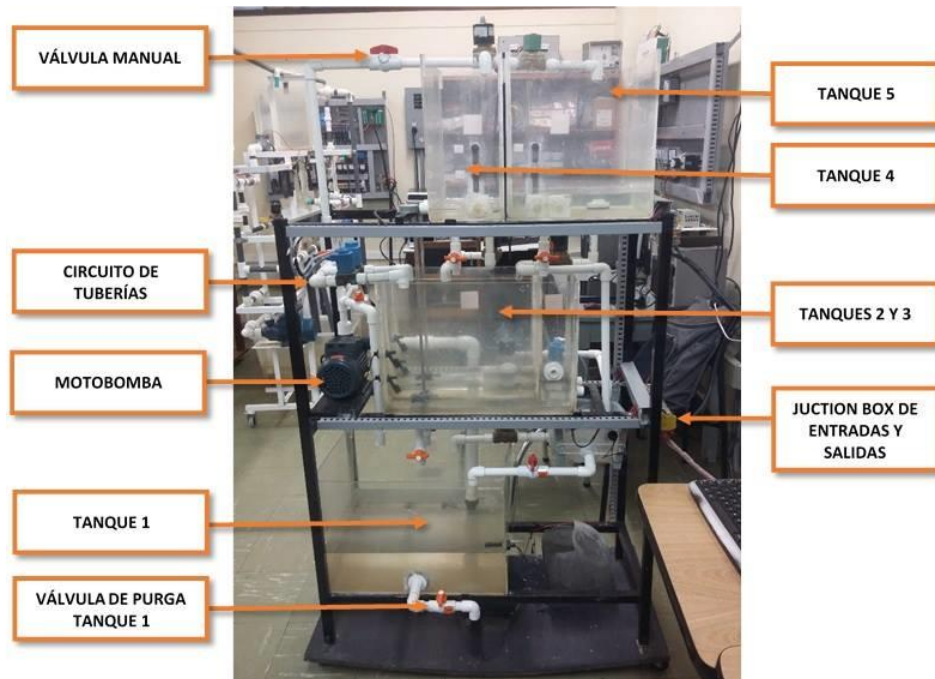
sistemas a eventos discretos. En la *Figura 49* se describe el diagrama P&ID de la planta SED, observando un sistema de control centralizado[54].

Figura 49. Diagrama PI&D planta SED.



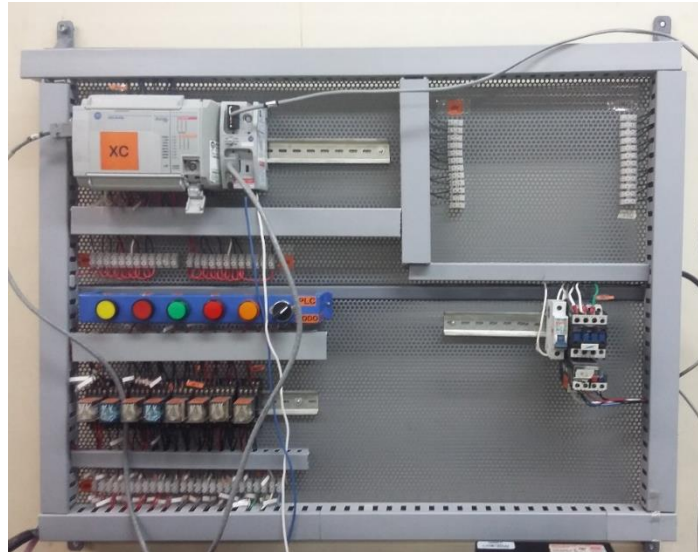
Fuente: Autores

Figura 50. Planta SED elementos de campo



Fuente: Tomada y Modificada de [54]

Figura 51. Planta SED elementos de control



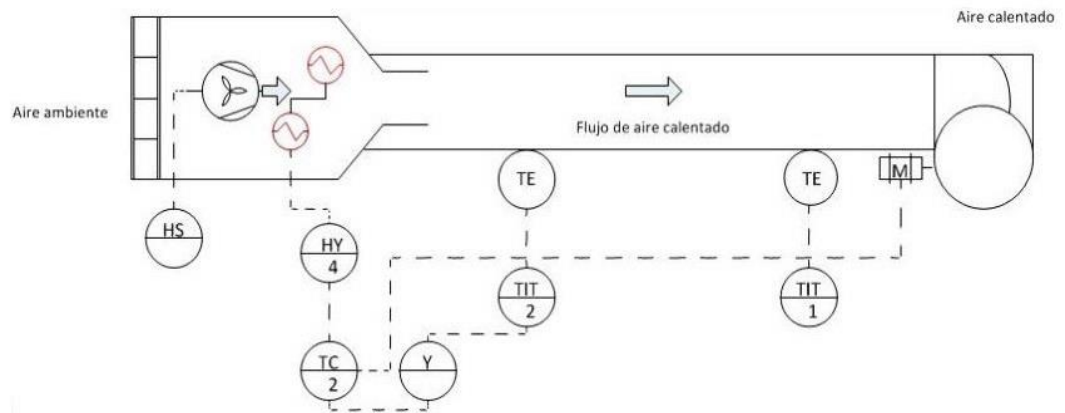
Fuente: Tomada y Modificada de [54]

4.1.3. Planta de Temperatura.

La planta de temperatura es un conducto cilíndrico por donde viaja aire impulsado por un ventilador y calentado por un sistema de resistencias. El caudal de aire se puede modificar según la posición de una válvula de persiana. Se cuenta con dos sensores de temperatura ubicados a distancias diferentes en el conducto cilíndrico. Permite la manipulación de la potencia aplicada al sistema de resistencias y la generación de disturbios consistentes en cambios en el caudal de aire tomado por el ventilador y en la salida del mismo. La *Figura 52* muestra el diagrama P&ID de la planta de Temperatura [55].

La *Figura 53* describe la estructura física de la planta, está compuesta por un tubo cilíndrico unido a una cámara donde ingresa aire a temperatura ambiente. Al interior de la cámara se encuentra un ventilador, el cual conduce el flujo de aire aspirado a una zona de calentamiento con 2 resistencias de calefacción de 1100w conectadas en serie y alimentadas por una fuente de 220v, la temperatura al interior del cilindro depende de la potencia o el suministro de voltaje hacia las resistencia calefactora, el voltaje se manipula desde el controlador(*Figura 54*) [55].

Figura 52. Diagrama PI&D planta de Temperatura.



Fuente: Autores

Estructura física de la planta de temperatura.

Figura 53. Planta de Temperatura estructura física.



Fuente: Tomada y Modificada de [55]

Figura 54. Planta de temperatura elementos de control



Fuente: Tomada y Modificada de [55]

4.2. IDENTIFICACIÓN DE UN PROCESO DISTRIBUIDO

De acuerdo a las plantas seleccionadas del laboratorio de control de procesos, se procede a generar el sistema de control distribuido, susceptible de ser modelado en estándar IEC 61499.

1. Descripción del proceso, organizar la secuencia y reconocer las posibles etapas que contenga

La producción de zumo de naranja, es el proceso elegido como se mencionó anteriormente; el zumo de naranja es un zumo de frutas en forma de líquido obtenido de exprimir el interior de las naranjas, es un producto alimenticio compuesto de diversos ingredientes, contiene una gran cantidad de vitamina C. Sobre las propiedades microbiológicas se pueden controlar con procesos térmicos que disminuyen las poblaciones tales como la Pasteurización mediante HTST¹⁰, con este tratamiento se previene sobre todo de las bacterias lácticas a las que afortunadamente se puede decir que presentan una resistencia baja a los tratamientos térmicos.

El proceso contara con las siguientes etapas: control de nivel de pulpa de naranja en un tanque y nivel de agua azucarada en otro tanque, posteriormente se realiza la mezcla homogénea de las dos sustancias, la mezcla será pasteurizada para evitar bacterias y cumplir con características necesarias para su consumo del producto.

Conocido el proceso, se pretende aplicar el procedimiento generado en el capítulo 3 de esta monografía, describiendo cada uno de sus ítems.

2. Identificar si el proceso es continuo, BATCH o discreto.

Se ha identificado el proceso de Zumo de naranja como BATCH, debido a que su producción se da en cantidades finitas de material, se someten estas cantidades de materiales de entrada a un conjunto ordenado de actividades de tratamiento durante un periodo finito.

3. No está geográficamente distribuido. el proceso de *Zumo de naranja* se da en un espacio delimitado, por ende, no es necesaria una distribución geográfica dentro de la empresa para poder llevar a cabo la producción

4. Dividir el proceso en subproceso o unidades.

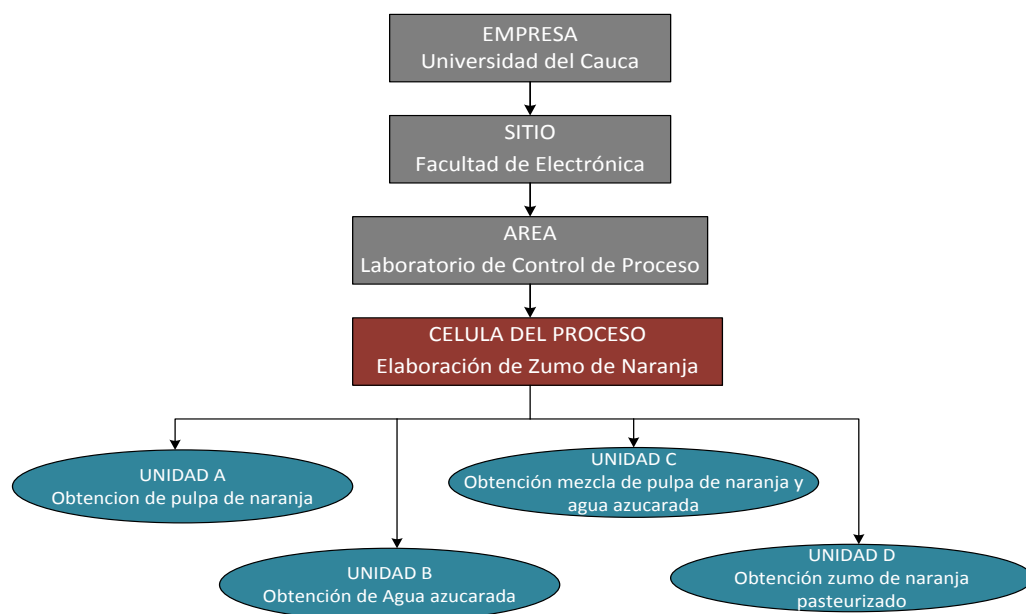
¹⁰ HTST(High Temperature/Short Time); Pasteurización relámpago o Pasteurización Flash

De acuerdo a la descripción que se hizo del procedimiento y a los diagramas que se presentaron, se obtiene una gran ventaja a la hora de organizar o dividir el proceso en subprocesos o unidades. En el procedimiento se propone el uso del estándar ISA S88.01, en específico del modelo físico. Se desarrolla ese modelo para poder observar las unidades planteadas dentro del proceso fundamental.

Modelo físico de ISA S88.01

El modelo físico general del proceso al cual se aplicara el procedimiento. La *Figura 55* muestra su jerarquización así: a nivel de empresa está la Universidad del Cauca, ubicada en la ciudad de Popayán, Colombia; a nivel de sitio se encuentra la Facultad de Ingeniería Electrónica de la Universidad del Cauca, situada al norte de Popayán, y a nivel de área está el Laboratorio de control de procesos del programa de Ingeniería en Automática Industrial. En cuanto a la planta de elaboración de Zumo de naranja, su célula de proceso se conforma jerárquicamente de 4 unidades: obtención de pulpa de naranja, obtención de agua azucarada, obtención de la mezcla de pulpa de naranja y agua azucarada, por ultimo esta la obtención de zumo de naranja pasteurizado.

Figura 55. Modelo Físico de ISA-88.01 (Zumo de naranja)



Fuente: Autores

Después de conocer el esquema del modelo físico, es de gran utilidad llegar hasta el punto de especificar las unidades que contiene la célula de proceso. Estas unidades son la correlación con los subprocesos una vez obtenida la división, se puede argumentar que el sistema es distribuido y presenta una distribución de sus variables de control, de acuerdo a las definiciones dadas en el Capítulo I.

Al enfocarnos en cada unidad, se obtiene la información detallada de los activos físicos que componen todo el proceso. En resumen la descripción física de la célula de proceso de Zumo de naranja por esta razón se especifica cada una de las unidades de la siguiente manera:

Para la *unidad A*; obtención de pulpa de naranja, se requiere controlar el nivel de pulpa de fruta, por lo cual se destina un tanque para almacenar la cantidad deseada.

La *Unidad B*; realiza el mismo procedimiento de la Unidad A, solo que el fluido es diferente, para este caso, se habla de agua azucarada necesaria para crear zumo de naranja con características y niveles de mezclado importantes.

Para el caso de la *Unidad C*; empleara una marmita, encargada de mezclar y dejar de forma homogénea la primera etapa del Zumo de naranja, esta mezcla se hace en conjunto con la pasteurización, es decir, mientras la sustancia se vuelve homogénea (pulpa de naranja y agua azucarada) se le inyecta calor sobre la mezcla.

Por ultimo esta la *Unidad D*; empleara un calefactor encargado de generar la temperatura ideal para pasteurizar la sustancia, por medio de tubería se ingresa el calor ideal a la marmita, para pasteurizar el *Zumo de naranja* y generar las condiciones ideales para finalizar con el producto

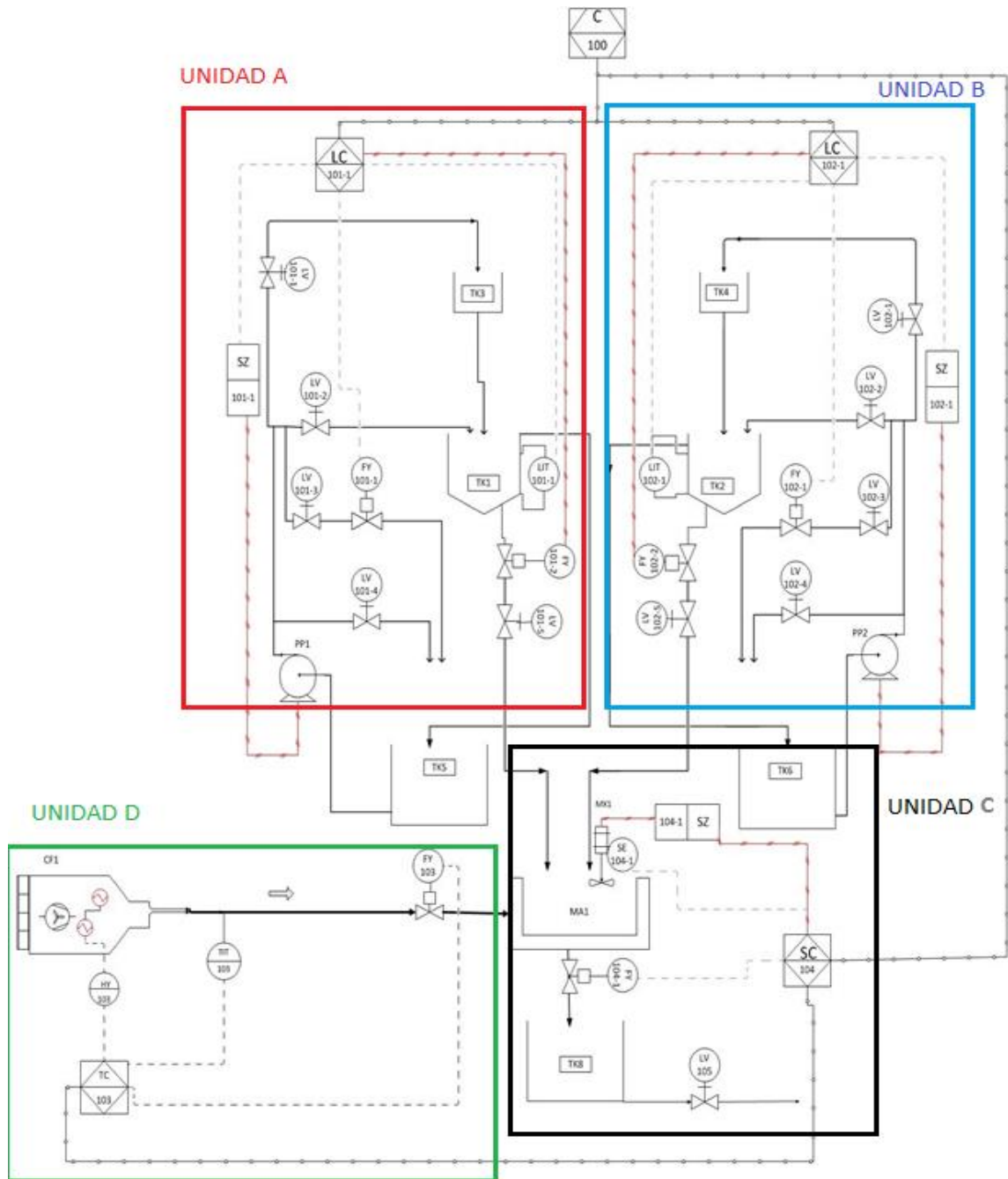
El siguiente ítem, muestra el diagrama P&ID general de planta, el diseño y los elementos que contiene, están hechos de acuerdo a los componentes con los que cuentan las plantas del laboratorio y con los que se puede contar para tratar de implementar el sistema

4.3. Diagrama P&ID y Tabla de instrumentos del proceso.

Basados en el estudio del proceso, como la definición, la descripción e investigación de las plantas que hay en el laboratorio, se llega a plantear el

siguiente diagrama PI&D, *Figura 56*. Definiendo un sistema distribuido, capaz de soportar múltiples lazos de control, integración de tres de las plantas que mejor se ajustan para implementar el DCS. Además, se muestra en la *Tabla 7* la lista de instrumentos que integra el sistema.

Figura 56. Diagrama P&ID caso de estudio



Fuente: Autores

Basados en el diagrama P&ID, se listan los instrumentos del sistema, implementado para el laboratorio del PIAI.

Tabla 7. Lista de Instrumentos P&ID

Lista de instrumentos		
Tag	Descripción	Servicio
PP1	Motobomba	Impulso de fluido
PP2	Motobomba	Impulso de fluido
C	Controlador Maestro	Control del proceso
TC	Controlador de Temperatura	Control de Temperatura
SC	Controlador de Velocidad	Control de Velocidad
LC 101-1	Controlador de Nivel	Control de Nivel
LC 102-1	Controlador de Nivel	Control de Nivel
LV	Valvula manual	Seguridad
FY	Electro Valvula	Paso de Fluido
FIT	Trasmisor Indicador de Flujo	Trasmitir e Indicar
LIT	Trasmisor Indicador de Nivel	Trasmitir e Indicar
TIT	Trasmisor indicador de Temperatura	Trasmitir e Indicar
SZ	Variador de Velocidad	Cambios de rpm
FE	Sensor de Flujo	Medir
TE	Sensor de Temperatura	Medir
SE	Sensor de Velocidad	Medir
TK	Tanque	Almacenar
MX	Mezclador	Mezclar
HS	Interruptor Manual	Interrumpir el paso de corriente
HY	Interruptor Eelectrico	Interrumpir el paso de corriente
Y	Transductor de Voltaje	Convertir señales
MA	Marmita	Mezclado y almacenado
CF	Calefactor	Proporciona aire caliente

Fuente: Autores

4.3.1. Lista de variables

Se listan los tipos de variables que contiene el proceso planteado.

Tabla 8. Lista de Variables caso de estudio

Parte del proceso	Variable	Tipo	Esquema de control	Rango	Unidad de medida
Control del Nivel	Nivel Agua Azucara	V.C	Feedback	0 – 10	Lt
	Caudal de entrada de Agua Azucarada.	V.M		0 – 100	GPM
	Nivel Pulpa de Naranja	V.C		0 – 10	Lt
	Caudal de entrada de pulpa de naranja	V.M.		0 – 100	GPM

Mezclado	Composición de la mezcla	V.C.	Feedback	70 – 30	%
	Velocidad del Motor	V.M.		900-3600	RPM
	Nivel Zumo de naranja	V.M		0 - 10	Lt
Pasteurización	Temperatura	V.C.	Feedback	110	°C
	Potencia eléctrica	V.M.		0-120	W

Fuente: Autores

4.3.2. Lista de equipos

Se listan los equipos de acuerdo a lo planteado y observado en los diagramas P&ID y en la descripción del proceso.

Tabla 9. Lista de Equipos caso de estudio

TAG	Equipo	Descripción
TK1	Tanque control nivel Pulpa naranja	Se realiza el control de nivel de la pulpa de naranja
TK2	Tanque control nivel agua azucarada	Se realiza el control de nivel del agua azucarada
TK3	Tanque Bypass	Almacena materia prima de Pulpa de naranja cuandoTK1 llega al límite.
TK4	Tanque Bypass	Almacena materia prima de agua azucarada cuandoTK2 llega al límite.
TK5	Tanque materia prima, pulpa de naranja	Contiene la materia prima (Pulpa de Naranja) de todo el proceso
TK6	Tanque materia prima, agua azucarada	Contiene la materia prima (Agua azucarada) de todo el proceso
MA1	Marmita	Mezcla y pasteuriza la cantidad de materia prima controlada en TK1 y TK2
CF1	Calefactor	Es el encargado de suministrar temperatura que se requiera a la marmita
TK8	Silo	Almacena todo la producción de zumo de naranja.

Fuente: Autores

4.4. Arquitectura del sistema propuesto.

Basados en las investigaciones hechas, en los elementos (Controladores, actuadores, sensores), en los procesos y en el diagrama P&ID, se plantea la arquitectura para el sistema. Esta arquitectura fue diseñada con ayuda del software Integrated Architecture Builder (IAB), permitiéndonos definir rápidamente

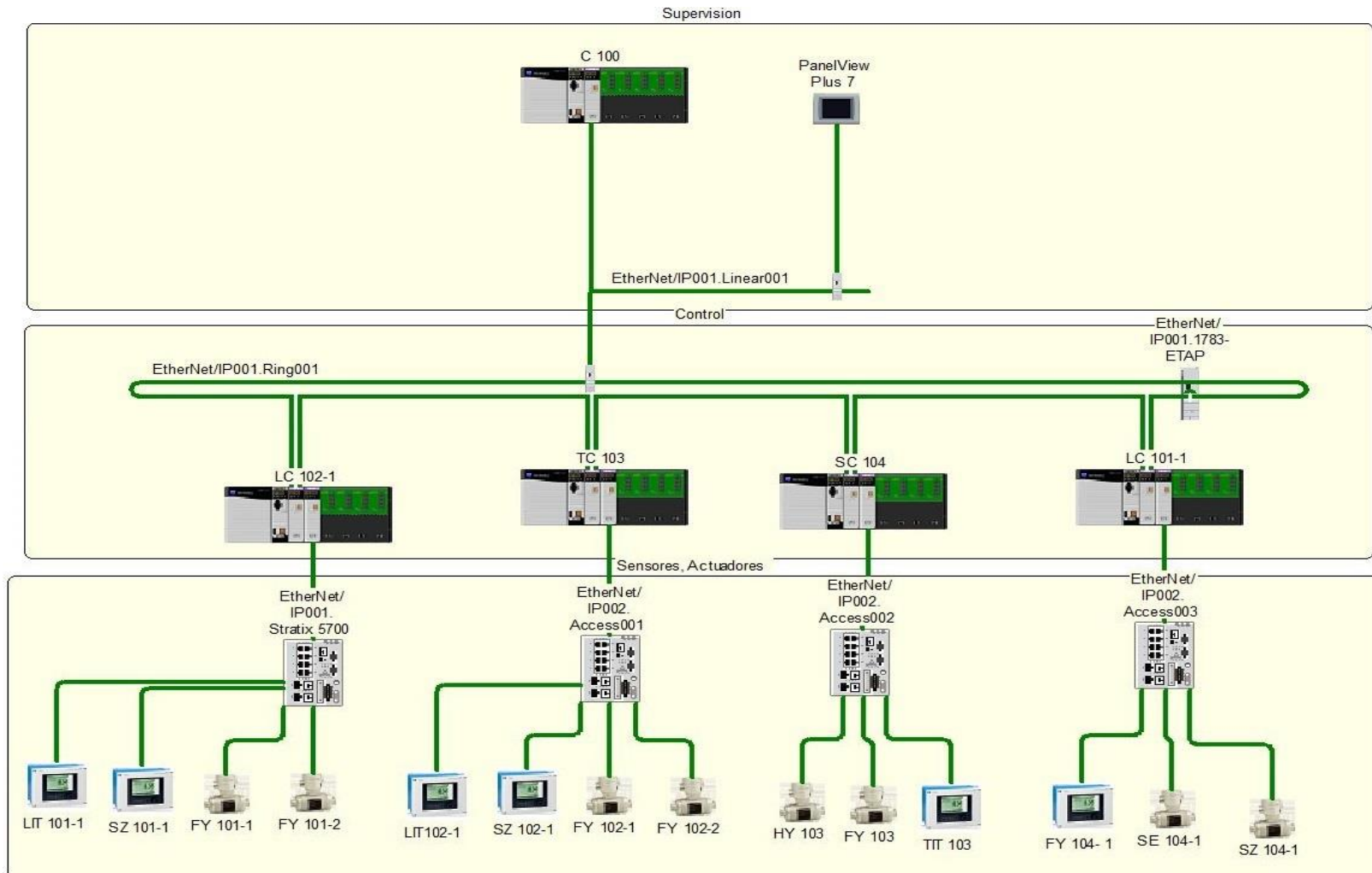
componentes de un sistemas de automatización, genera fácilmente listas de materiales y cotizaciones, además se pueden configurar sistemas basados en logix, y generar lista de materiales para redes basados en Netlinx, es decir, DeviceNet, ControlNet, y Ethernet/IP.

Los componentes que se muestran en la siguiente arquitectura son un modelo para acercar al lector a implementar el diseño que se obtuvo, con IAB. Se trabajó con esta herramienta para conocer de cerca lo que se usa en la industria. La *Figura 57* muestra la arquitectura y representa algunos de los equipos necesarios para el sistema de control distribuido. Además se plantea el software ISaGRAF para la programación de estos controladores en sistemas como el estudiado, ISaGRAF es desarrollado por Rockwell Automation, lanzada en el año 2014, y se están realizando aplicaciones que han dado buenos resultados a nivel industrial por la flexibilidad que posee; esta herramienta ha logrado mejorar sus primeras implementaciones del nuevo estándar, además de las tradicionales funciones IEC 61131-3 [56]. Según los fabricantes, las mejoras incluyen a los bloques de funciones básicas y los modelos de bloques de funciones compuestas, utilizados en los programas, mediante el uso de librerías de diferentes marcas de componentes de control distribuido.

La BOM¹¹ de la arquitectura (Ver Anexo B) se observara las características de cada uno de los componentes, tales como procesadores, slot's, tipo de entradas y salidas, módulos de comunicación entre otras

¹¹ BOM: Bill Of Materials

Figura 57. Arquitectura Caso de Estudio.



Fuente: Autores

4.5. Modelado con IEC-61499

A continuación se procede aplicar los pasos que tiene como objetivo modelar el sistema con IEC 61499.

4.5.1. Identificación de dispositivos

Los dispositivos identificados a partir de la arquitectura desarrollada son los siguientes:

- DISPLAY (Panel View)
- C-100
- LC-101-1
- LC-102-1
- TC-103
- SC-104

4.5.2. Secuencia de operaciones.

Para entender el comportamiento del sistema se describe la secuencia de operaciones que realiza cada dispositivo, empleando redes de Petri híbridas[57] para detallar el control que realizan los dispositivos en las diferentes unidades, para los elementos que realizan control ON/OFF.

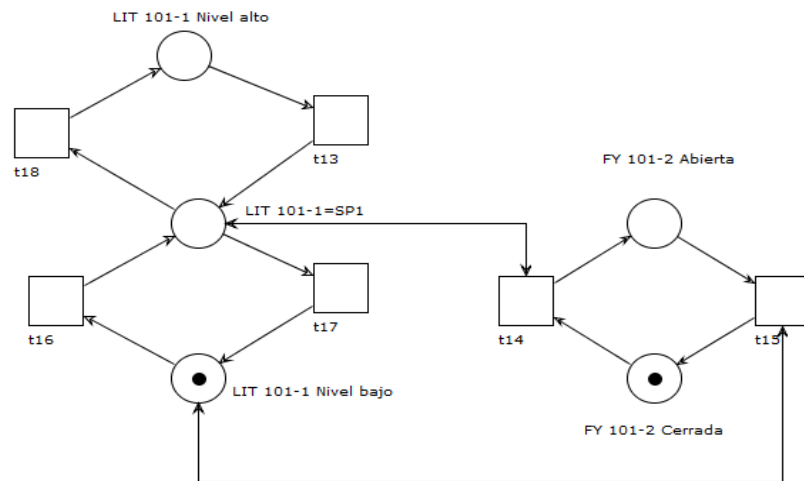
4.5.2.1. C-100

Controlador central, encargado de que la secuencia de operaciones que realiza cada dispositivo se lleve a cabo en el orden en que se han definido.

4.5.2.2. LC-101-1

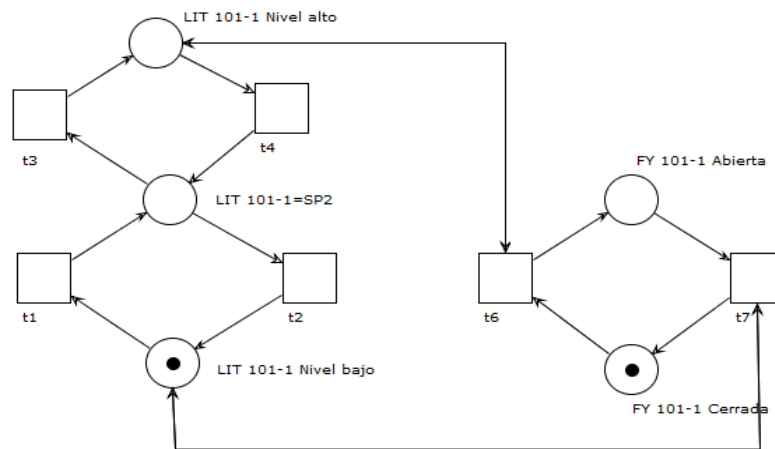
- Se encarga de controlar el nivel en TK1, por medio del variador por SZ 101-1, de acuerdo al valor de SP1 (set point TK1), el elemento que indica el nivel en el tanque es LIT 101-1, este control se realizará mediante un PID.
- La válvula FY 101-2 se abrirá cuando LIT 101-1 sea igual al SP1, se cerrará cuando LIT 101-1 sea menor del SP1 (Nivel bajo), *la Figura 58* representa lo anterior.
- La válvula FY 101-1 se abrirá en caso de emergencia, cuando LC 101-1 este en Nivel alto (LIT 101-1 sea mayor del SP1), y se desactivará cuando LC 101-1 este en el Nivel bajo (*Figura 59*).

Figura 58 Red de Petri Válvula FY 101-2



Fuente: Autores

Figura 59 Red de Petri Válvula FY 101-1



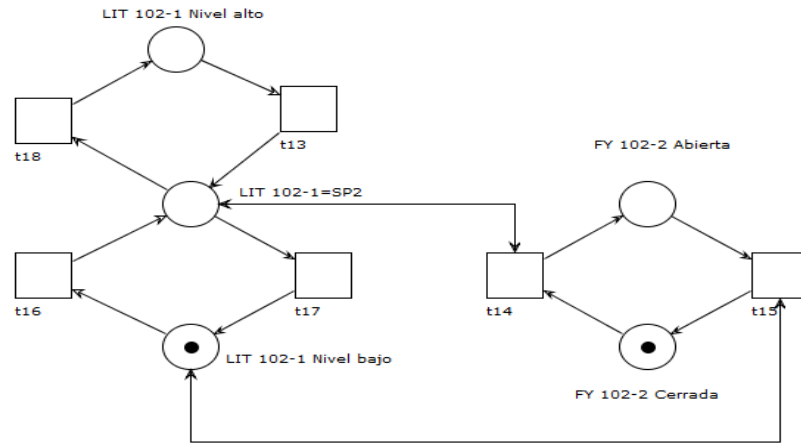
Fuente: Autores

4.5.2.3. LC-102-1

- Se encarga de controlar el nivel en TK2, por medio del variador por SZ 102-1, de acuerdo al valor de SP2 (set point TK2), el elemento que indica el nivel en el tanque es LIT 102-1, este control se realizará mediante un PID.

- La válvula FY 102-2 se abrirá cuando LIT 102-1 sea igual al SP2, se cerrará cuando LIT 102-1 sea menor del SP2 (Nivel bajo), la Figura 60 representa lo anterior.

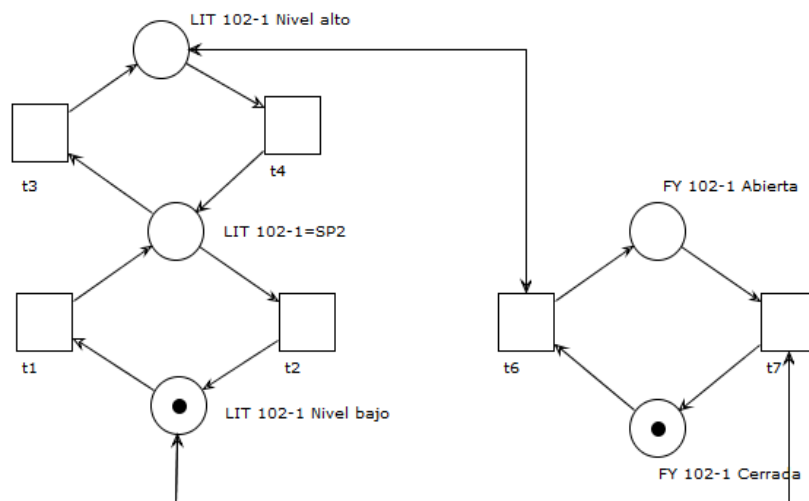
Figura 60 Red de Petri Válvula FY 102-2



Fuente: Autores

- La válvula FY 102-1 se abrirá en caso de emergencia, cuando LC 102-1 este en Nivel alto (LIT 102-1 sea mayor del SP2), y se desactivará cuando LC 102-1 este en el Nivel bajo (Figura 61).

Figura 61. Red de Petri FY 102-1

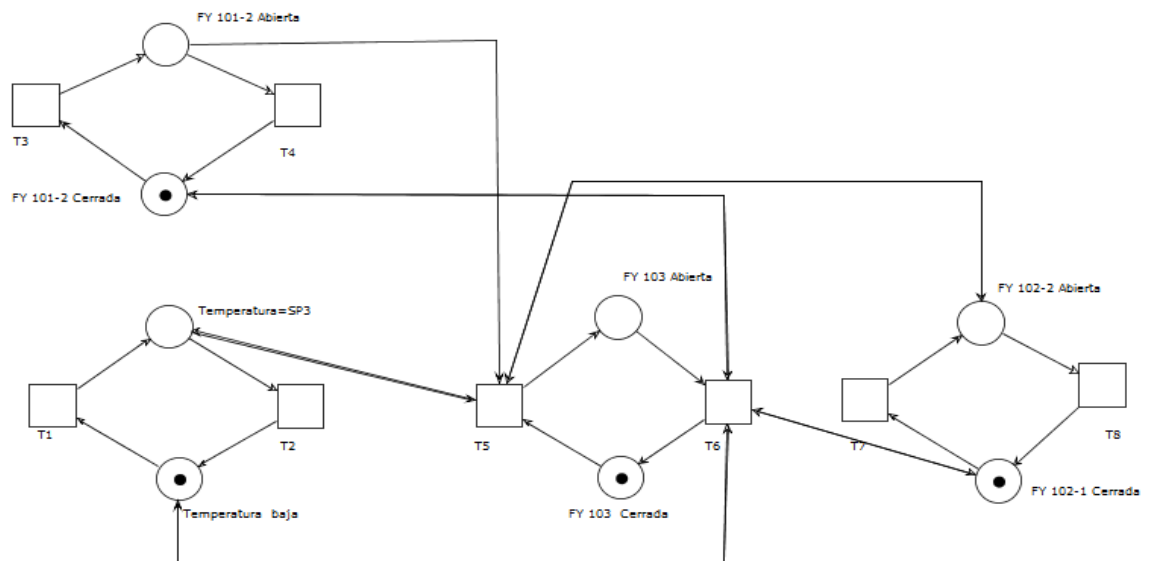


Fuente: Autores

4.5.2.4. TC-103

- Controla la temperatura en CF1 por medio de la potencia entregada a HY 103, este control se realiza por medio de un controlador PID.
- La válvula FY 103, se abre cuando FY 101-2 y FY 102-2 estén abiertas y la temperatura en CF1 alcanza el SP3 (set point de temperatura), se cierra cuando FY 101-2 y FY 102-2 estén cerradas (*Figura 62*).

Figura 62. Red de Petri Válvula FY 103

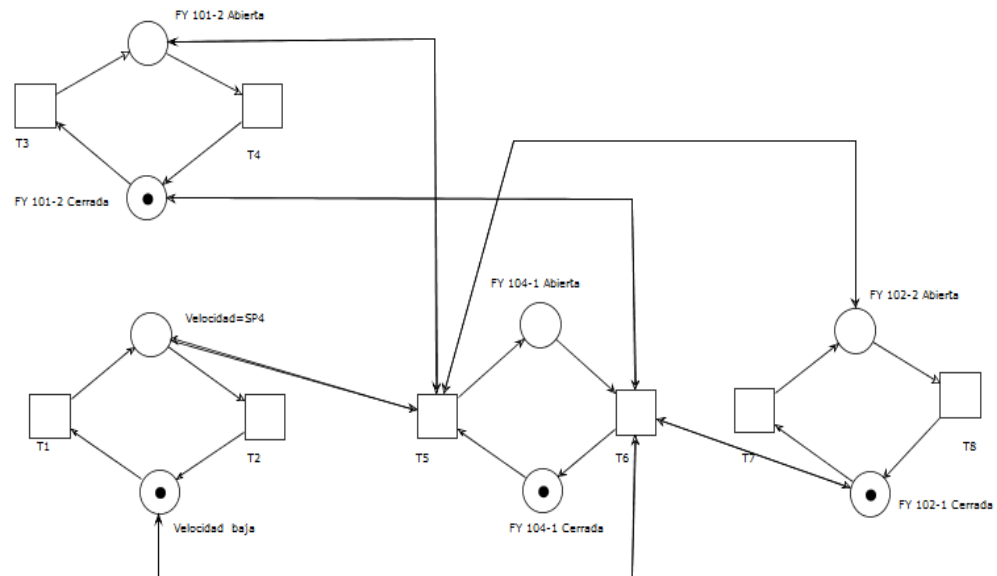


Fuente: Autores

4.5.2.5. SC-104

- El mezclador MX1 se pone en marcha cuando las válvulas FY 101-2 y FY 102-2 están abiertas. Regula la velocidad (por medio de SZ 104-1) mediante el sensor SE 104-1, que esta acoplado al mezclador MX1.
- Cuando MX1 alcance el SP4 (set point de la velocidad de MX1), deben pasar 10 minutos para activar la válvula FY 104-1 (*Figura 63*).

Figura 63. Red de Petri Válvula FY 104-1



Fuente: Autores

4.5.3. Funcionalidad de dispositivos.

Con la ayuda de la siguiente tabla, se procede a describir las diferentes funciones que cumple cada dispositivo para modelar el sistema con IEC 61499.

Tabla 10. Funcionalidad de los dispositivos de control

	Descripción			
HMI PRINCIPAL	Envía los parámetros de control PID y recibe el estado de las variables controladas.			
C-100	verificar que la secuencia de operaciones se cumpla, activa o desactiva las funciones que realiza cada controlador			
	Control PID	Descripción	Control ON/OFF	Descripción
LC-101-1	Si	Control de nivel en TK1 por medio de SZ 101-1	Si	Control de las válvulas FY 101-2, FY 101-1 depende de TK1
LC-102-1	Si	Control de nivel en TK2 por medio de SZ 102-1	Si	Control de las válvulas FY 102-1, FY 102-2 depende de TK2
TC-103	Si	Control de temperatura en MA1 por medio de HY-103	Si	Control de la válvula FY 103
SC-104	Si	Control de la variación de la frecuencia en MX1 por medio de SZ 104-1	Si	Control de la válvula FY 104

Fuente: Autores

4.5.4. Entornos de Desarrollo y de Ejecución de la Norma IEC-61499

Para el desarrollo de modelos del estándar IEC 61499, el primer paso es la selección de la herramienta de desarrollo, como se mencionó en el capítulo II, IEC 61499-2 define los requerimientos software, a continuación se presentan 4 herramientas relevantes para la construcción de los modelos, con sus respectivos entornos de desarrollo y ejecución, más adelante se indican las herramientas hardware compatibles con el estándar.

4.5.4.1. Entornos de Desarrollo

El entorno de desarrollo integrado, denominado IDE (siglas en inglés: Integrated Development Enviroment), está compuesto por un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). A continuación se procede a describir los 4 entornos de desarrollo para IEC -61499[15].

- FBDK:

Functional Block Development Kit fue la primera herramienta software desarrollada. Es una aplicación en Java, desarrollada por uno de los creadores del estándar para representar gráficamente las aplicaciones de la norma, pero con el tiempo se ha desarrollado permitiendo actualmente testear los bloques e intercambiarlos en forma de XML[15], la herramienta tiene soporte y se distribuye de forma gratuita por la empresa Holoblocs Inc[44][58].

- 4DIAC-IDE

Es una herramienta opensource basada en el entorno de desarrollo Eclipse. Además de ser robusta y sencilla de usar permite la creación de bloques de función básicos, sistemas, aplicaciones, etc, fue creada en el año 2000 por PROFACTOR GmbH & Viena University of Technology [59][60].

- nxtOne:

Esta es una aplicación comercial desarrollada por la compañía nxtControl, que ofrece soporte para la norma IEC 61499[61], incluye funciones de debugging y de test, a diferencia de 4DIAC. Por otra parte también incluye una funcionalidad CAT (Compound Automation Types) la cual aporta un conjunto de elementos de control prediseñados[44].

- ISaGRAF:

Al igual que nxtOne es una herramienta comercial desarrollada por la compañía ICE Triplex que pertenece. Pertenece a Rockwell automation [62]. Esta herramienta soporta la IEC 61499 conjuntamente con la IEC 61131. Al igual que 4DIAC y nxtOne, tiene funcionalidades de debug, test y soporte oficial. Se ha utilizado en el desarrollo de muchas aplicaciones en componentes de diferentes vendedores[44].

4.5.4.2. Entornos de ejecución

El entorno de ejecución (siglas en ingles: RTE, Runtime Enviroment), se encargara de gestionar el Hardware del dispositivo sobre el que se ejecuta el modelo, entre otras cosas deberá gestionar la memoria, las comunicaciones y las entradas/salidas del dispositivo. Se puede asemejar el entorno de ejecución a un sistema operativo que se encarga de ejecutar el programa generado por el entorno de desarrollo. A continuación se procede a describir los entornos de ejecución existentes[44].

- FBRT:

Es el primer entorno de ejecución que se desarrolló en Java embebido para probar los modelos desarrollados con FBDK[58].

- FORTE

Se desarrolló por el mismo equipo que 4DIAC-IDE, es una plataforma de ejecución opensource, permite la ejecución de sistemas en tiempo real en una amplia variedad de dispositivos[60], ya que se ha diseñado para ser independiente de la plataforma en la que se ejecute [15]

- nxtIECRT

Este entorno de ejecución pertenece a la compañía ntxControl, fue lanzado al mismo tiempo que la herramienta de desarrollo nxtSTUDIO[63]. Está basado en FORTE pero incluye servicios y funciones adicionales [44].

- ISaGRAF Runtime:

Desarrollado por la compañía ICE Triplex este entorno de ejecución se emplea para los modelos desarrollados por ISAGRAF[44][62].

4.5.5. HERRAMIENTAS HARDWARE COMPATIBLES CON IEC 61499

Las herramientas hardware encontradas de acuerdo con los entornos de ejecución, se detallan a continuación

4.5.5.1. WAGO PFC200

El controlador WAGO PFC200, es una solución hardware, funciona con el entorno de ejecución FORTE, presentando las siguientes características encontradas en la página del fabricante[64]:

- **Número de módulos de E / S:**64
- **CPU:** 32 bits RISC ARM7TDMI
- **Memoria RAM:** 16 Mbytes SDRAM, 32 Kbytes NOVRAM
- **Flash:** 4 Mbytes
- **EEPROM:** 4 Kbytes
- **sistema operativo:** Linux (Kernel versión 2.6)
- **Fuente de alimentación:** 24 V DC (-15% + 20%)
- **Máxima corriente de entrada:** (24 V) 500 mA
- **Eficiencia de la fuente de alimentación:** 87%
- **Consumo de corriente interna:** (5 V) 300 mA

4.5.5.2. NxtDCSmini

NxtDCSmini es una solución hardware propuesta por la empresa ntxControl, funciona con el entorno de ejecución nxtIECRT, presentando las siguientes características encontradas en la página del fabricante[65]:

- **Fieldbus:** Ethercat
- **Salida digital:** 8x 24VDC
- **Entrada digital:** 8x 24VDC
- **Entrada analógica:** 2x 0-10VDC
- **Puerto RS-232:** 1
- **Puertos LAN:**2

4.5.5.3. QORIQ_ETHERCAT

QORIQ_ETHERCAT es la solución hardware que permite implementar el entorno de ejecución ISaGRAF Runtime, las características que esta herramienta presenta se describen a continuación[66]:

- **Procesador :** QorIQ® P1025 dual-core
- **FieldBus:** EtherCAT
- **Entrada digital:** 4 entradas digitales

- **Salida digital:** 4 salidas digitales

El entorno de ejecución FBRT dentro de la investigación realizada, presenta como propuesta la herramienta hardware NETMASTER[67], la cual no fue tomada en cuenta por estar descontinuada por su fabricante el sist.

Los autores de este documento no han encontrado casos de uso con estas herramientas (Hardware, software) en aplicaciones reales de control industrial con IEC 61499, se identifica su uso mayoritario en el ámbito académico en laboratorios de investigación, la documentación también es poca cuando se trata de simular o implementar una solución tecnológica en base al estándar. Por lo anterior se optó por representar los modelos mediante una herramienta de desarrollo.

4.5.6. Identificación de los Bloques de Función

El primer paso para modelar el sistema con IEC 61499 es la selección de la herramienta de desarrollo, en este caso se escogió 4DIAC [45], por poseer una interfaz intuitiva de fácil manejo, además de traer bloques de función ya predefinidos de acuerdo a IEC 61499. Las imágenes que se presentaron para representar los modelos han sido copiadas desde la herramienta.

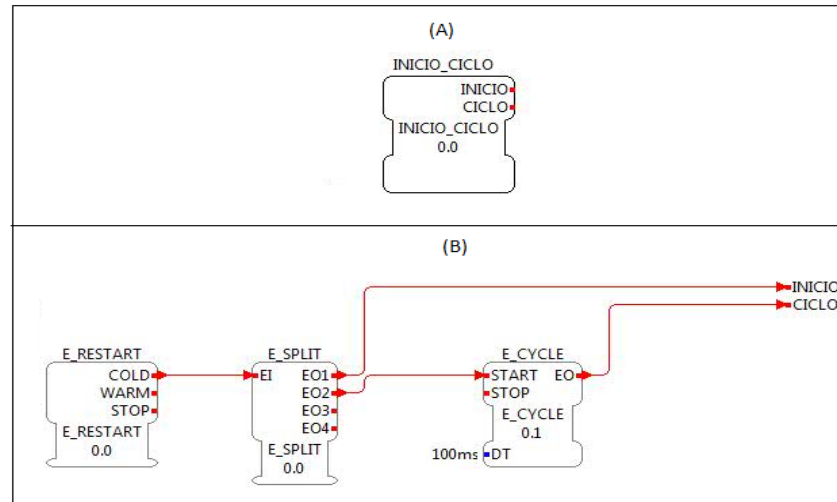
A continuación se procede a presentar los diferentes bloques de función empleados por los dispositivos, en primera instancia se presentaron los bloques generales, que se pueden aplicar en cualquier dispositivo como lo son los bloques de eventos y el del controlador PID.

Los bloques de interfaz de servicio empleados por los dispositivos, no se especificaron, debido a que su funcionamiento se detalla en el capítulo 2.

4.5.6.1. Bloques de eventos

La Figura 64 representa el bloque de función Inicio_Ciclo, la parte (A) muestra la interfaz externa utilizada, la parte (B) muestra su estructura interna, es el encargado de contener los bloques E_RESTART y E_CYCLE, los cuales generaran los eventos 'INIT' y 'CICLO'. El evento INIT genera la inicialización de los bloques de función que integran el dispositivo. El evento Ciclo permite la ejecución de los bloques cada 100ms, con el fin de refrescar el valor entregado por los diferentes sensores.

Figura 64. Bloque de función Inicio_Ciclo.

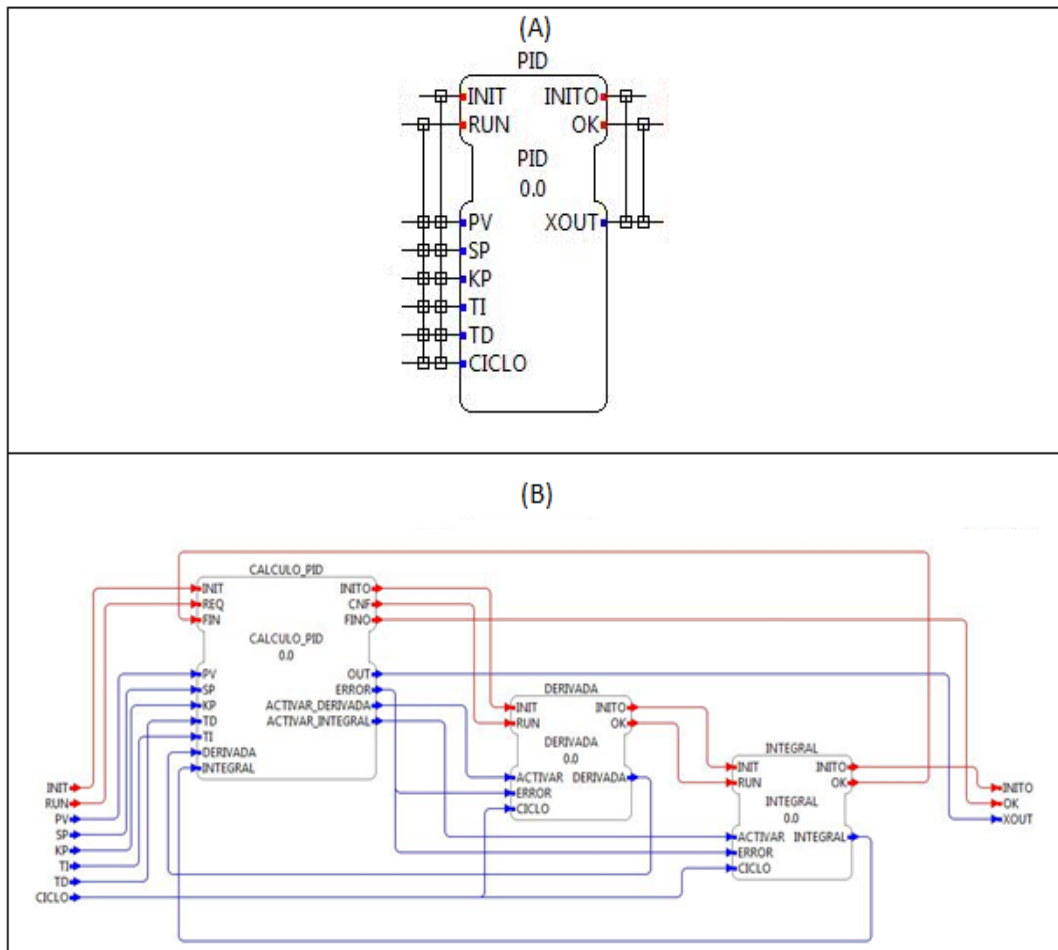


Fuente: Autores.

4.5.6.2. PID

El bloque de función PID es compuesto y está representado por la *Figura 65*, la parte (A) muestra la interfaz externa utilizada, la parte (B) muestra su estructura interna, compuesta por los bloques: CALCULO_PID, DERIVADA, INTEGRAL, posterior a esto se hace la descripción de dichos bloques.

Figura 65. Bloque de función PID

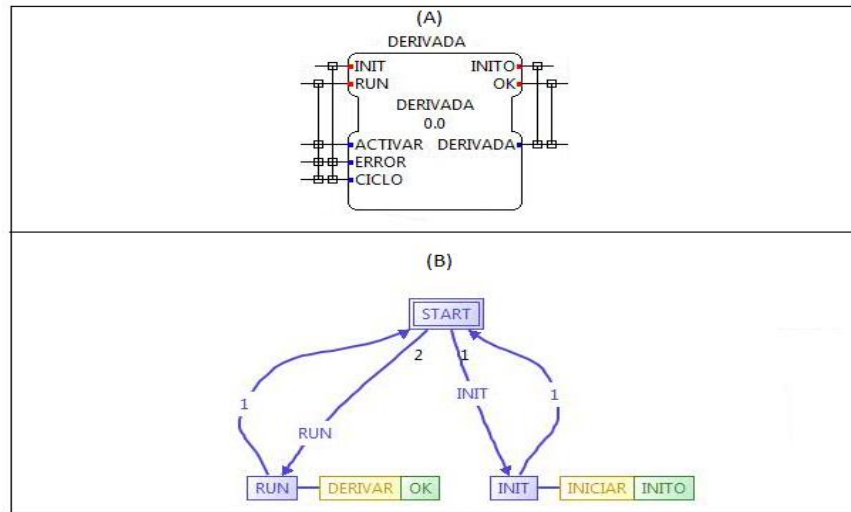


Fuente: Autores

4.5.6.2.1. DERIVADA

La Figura 66 representa el bloque DERIVADA, la parte (A) muestra la interfaz externa utilizada, La parte (B) muestra su respectivo grafico de control de ejecución, por ser un bloque de función básico.

Figura 66. Bloque de función DERIVADA



Fuente: Autores

La Tabla 11 describe la relación entre eventos y datos del bloque DERIVADA, se definen los valores de los datos empleados y algoritmos en lenguaje estructurado.

Tabla 11. Descripción de eventos y datos del bloque DERIVADA.

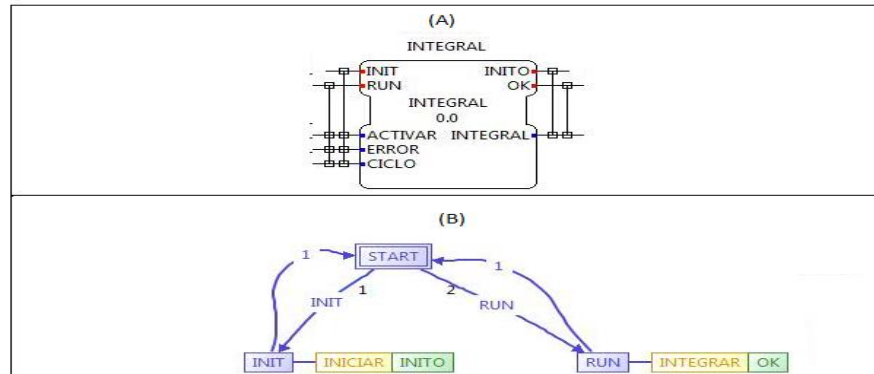
Eventos Entrada	Datos Entrada
INIT: Este evento permite inicializar el bloque DERIVADA.	ACTIVAR: Esta entrada de datos al toma el valor verdadero (True), permite realizar el cálculo de la derivada.
RUN: Este evento cada vez que es activado, permite al bloque DERIVADA, realizar el cálculo de la derivada.	ERROR: Es el valor del error al cual se le va calcular la derivada.
	CICLO: Es el valor del tiempo de muestreo para el cálculo de la derivada.
Eventos Salida	Datos Salida
INIT0: Este evento indica que el DERIVADA ha complementado su inicialización.	DERIVADA: Tiene listo el valor calculado de la derivada.
OK: Este evento es activado cuando el bloque DERIVADA, requiere hacer un nuevo cálculo.	
Variable Interna	
X1: REAL	
Algoritmo: INICIAR	Algoritmo: DERIVAR
DERIVADA := 0.0; X1 := ERROR;	IF ACTIVAR THEN DERIVADA := (ERROR - X1)/CICLO X1 := ERROR END_IF;

Fuente: Autores

4.5.6.2.2. INTEGRAL

La *Figura 67* representa el bloque INTEGRAL, la parte (A) muestra la interfaz externa utilizada, la parte (B) muestra su respectivo grafico de control de ejecución, por ser un bloque de función básico.

Figura 67. Bloque de función INTEGRAL.



Fuente: Autores

La *Tabla 12* describe la relación entre eventos y datos del bloque integral, se definen los valores de los datos empleados y algoritmos.

Tabla 12. Descripción de eventos y datos del bloque INTEGRAL.

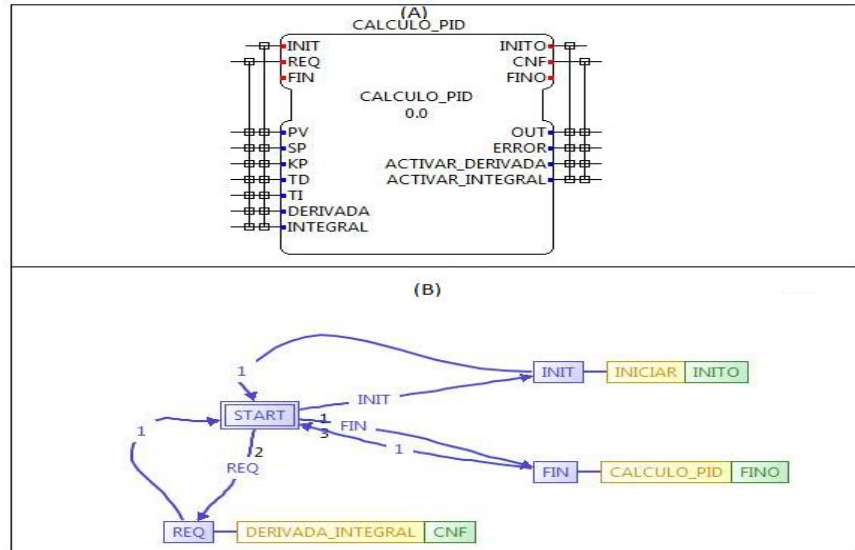
Eventos Entrada	Datos Entrada
INIT: Este evento permite inicializar el bloque INTEGRAL.	ACTIVAR: Esta entrada de datos al toma el valor verdadero (True), permite realizar el cálculo de la integral.
RUN: Este evento cada vez que es activado, permite al bloque INTEGRAL, realizar el cálculo de la INTEGRAL.	ERROR: Es el valor del error al cual se le va calcular la integral.
	CICLO: Es el valor del tiempo de muestreo para el cálculo de la integral.
Eventos Salida	Datos Salida
INITO: Este evento indica que el bloque INTEGRAL ha complementado su inicialización.	INTEGRAL: Tiene listo el valor calculado de la integral.
OK: Este evento es activado cuando el bloque INTEGRAL, requiere hacer un nuevo cálculo.	
Algoritmo: INICIAR	Algoritmo: INTEGRAR
INTEGRAL := 0.0;	IF ACTIVAR THEN INTEGRAL := INTEGRAL + ERROR * CICLO; END_IF;

Fuente: Autores

4.5.6.2.3. CALCULO_PID

La *Figura 68* representa el bloque CALCULO_PID, la parte (A) muestra la interfaz externa utilizada, La parte (B) muestra su respectivo grafico de control de ejecución, por ser un bloque de función básico.

Figura 68. Bloque de función CÁLCULO_PID



Fuente: Autores

La *Tabla 13* describe la relación entre eventos y datos del bloque integral, se definen los valores de los datos empleados y algoritmos.

Tabla 13. Descripción de eventos y datos del bloque CÁLCULO: PID

Eventos Entrada	Datos Entrada
INIT: Este evento permite inicializar el bloque CALCULO_PID.	PV: Esta entrada de datos al toma el valor verdadero (True), permite realizar el cálculo de la integral.
	SP: Es el valor de referencia para la variable de proceso.
	KP: Es el valor de la constante proporcional.
RUN: Este evento permite activar el bloque CALCULO_PID,	TI: Es el valor del tiempo integral.
	TD: Es el valor del tiempo derivativo.
	DERIVAR: Es el valor del tiempo de muestreo para el cálculo de la integral.

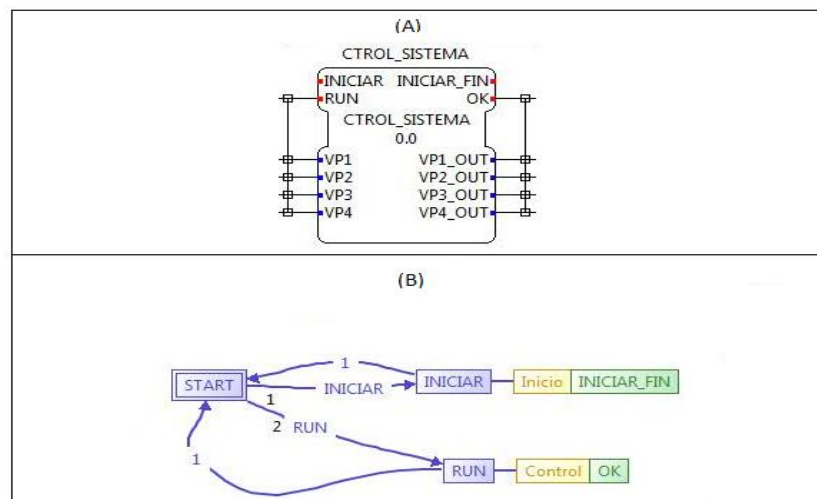
	INTEGRAL: Es el valor del tiempo de muestreo para el cálculo de la integral.
Eventos Salida	Datos Salida
INIT0: Este evento indica que el bloque INTEGRAL ha complementado su inicialización.	OUT: Tiene listo el valor calculado de la integral.
	ERROR: Obtiene el error calculado a partir de la variable de proceso y el set point.
OK: Este evento es activado cuando el bloque INTEGRAL, requiere hacer un nuevo cálculo.	ACTIVAR_DERIVADA: Al colocarse esta entrada en TRUE, permite activar el bloque DERIVADA.
	ACTIVAR_INTEGRAL
Algoritmo: INICIAR	Algoritmo: DERIVADA INTEGRAL
XOUT := 0.0;	ERROR := SP - PV; DERIVADA:= TRUE; INTEGRAL := TRUE;
Algoritmo: CALCULAR PID	
$XOUT := (ERROR + (1/TI) * INTEGRAL + TD * DERIVADA) * KP$	

Fuente: Autores

4.5.6.3. Bloques de función del dispositivo C 100

La Figura 69 representa el bloque CTROL_SISTEMA, la parte (A) muestra la interfaz externa utilizada, la parte (B) muestra su respectivo grafico de control de ejecución.

Figura 69. Bloque de función CÁLCULO PID.



Fuente: Autores

La *Tabla 14* describe la relación entre eventos y datos del CTROL_SISTEMA, se definen los valores de los datos empleados y algoritmos.

Tabla 14. Descripción de eventos y datos del bloque CTROL_SISTEMA.

Eventos Entrada	Datos Entrada
INICIAR: Este evento permite inicializar las variables del bloque CTROL_SISTEMA.	VP1: Esta entrada recibe el estado del nivel de TK1.
RUN: Este evento permite realizar el algoritmo, para activar o desactivar los controladores requeridos.	VP2: Esta entrada recibe el estado del nivel de TK2.
	VP3: Esta entrada recibe el estado actual de la temperatura en CF1.
	VP4: Esta entrada recibe el estado actual de la velocidad en MX1.
Eventos Salida	Datos Salida
INICIAR_FIN: Este evento indica que el bloque CTROL_SISTEMA. Ha complementado su inicialización.	VP1_OUT: Esta salida envía el estado el estado del nivel de TK1.
OK: Este evento es activado cuando el bloque CTROL_SISTEMA, requiere revisar el estado actual de los controladores.	VP2_OUT: Esta salida envía el estado del nivel de TK2.
	VP3_OUT: Esta salida envía el estado actual de la temperatura en CF1.
	VP4_OUT: Esta salida envía el estado la velocidad en MX1.
Algoritmo: INICIO	Algoritmo: CONTROL
VP1_OUT=TRUE; VP2_OUT=TRUE; VP3_OUT=FALSE; VP4_OUT=FALSE;	IF(VP1="TRUE" AND VP2="TRUE")THEN VP1_OUT="TRUE"; VP2_OUT="TRUE"; VP3_OUT="FALSE"; VP4_OUT="FALSE"; END_IF IF(VP3="TRUE" AND VP4="TRUE") THEN VP1_OUT="TRUE"; VP2_OUT="TRUE"; VP3_OUT="FALSE"; VP4_OUT="FALSE"; END_IF

Fuente: Autores

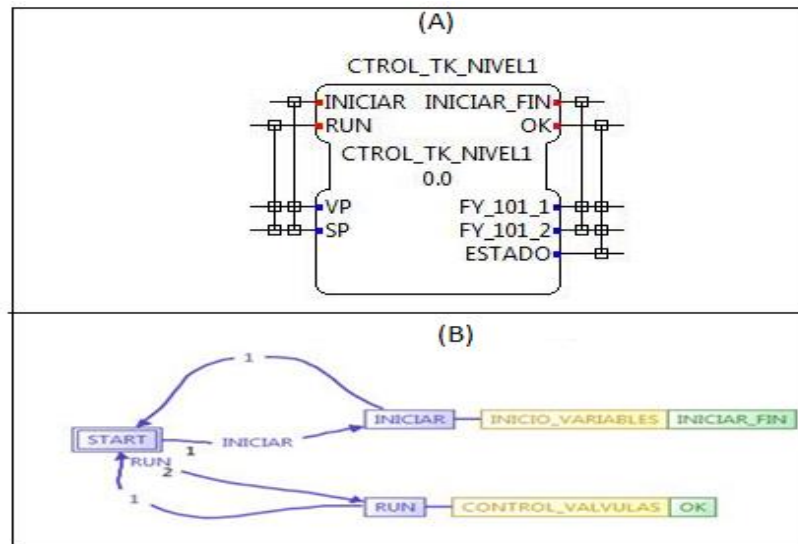
4.5.6.4. Bloques de función del dispositivo LC101-1

Se presentan los bloques de función empleados por el dispositivo LC 101-1 y que hacen parte del modelado del sistema

4.5.6.4.1. CTROL_TK_NIVEL1

Es el bloque encargado de realizar el control de las válvulas FY 101-1 y FY 102-1, la *Figura 70* representa el bloque, la parte (A) muestra la interfaz externa utilizada, La parte (B) muestra su respectivo grafico de control de ejecución.

Figura 70. CTROL_TK_NIVEL1.



Fuente: Autores

La *Tabla 15* describe la relación entre eventos y datos del bloque CTROL_TKNIVEL1 se definen los valores de los datos empleados y algoritmos.

Tabla 15. Descripción de eventos y datos del bloque CTROL_TK_NIVEL1

Eventos Entrada	Datos Entrada
INICIAR: Este evento permite inicializar las variables del bloque CTROL_TKNIVEL1.	VP: Esta entrada recibe el valor de la variable de proceso, en este caso el nivel en TK1.
RUN: Este evento permite iniciar el algoritmo que controlara las válvulas FY 101-1 y FY 102-1.	SP: Esta entrada recibe el valor del set point.
Eventos Salida	Datos Salida
INICIAR_FIN: Este evento indica que el bloque CTROL_TKNIVEL1 ha complementado su inicialización.	FY 101-1: Entrega el valor calculado, a la válvula FY 101-1.
OK: Este evento es activado cuando el bloque CTROL_TKNIVEL1, requiere hacer un nuevo cálculo.	FY 101-2: Entrega el valor calculado, a la válvula FY 101-2.
	ESTADO: Entrega el estado actual de la variable de proceso, este valor es enviado al controlador central

Algoritmo: INICIO_VARIABLES	Algoritmo: CONTROL_VALVULAS
VP=0, SP=0, FY_101_1=FALSE, FY_101_2=FALSE; ESTADO= FALSE;	<pre> IF (VP=SP) THEN FY_101_2=TRUE (*Se active la válvula FY_101_2*) ESTADO=TRUE ELSE FY_101_2=FALSE (*Se desactiva la válvula FY_101_2*) ESTADO=FALSE END_IF IF (VP>SP) THEN FY_101_1=TRUE (*Se active la válvula FY_101_1*) ELSE FY_101_1=FALSE (*Se desactiva la válvula FY_101_1*) END_IF </pre>

Fuente: Autores

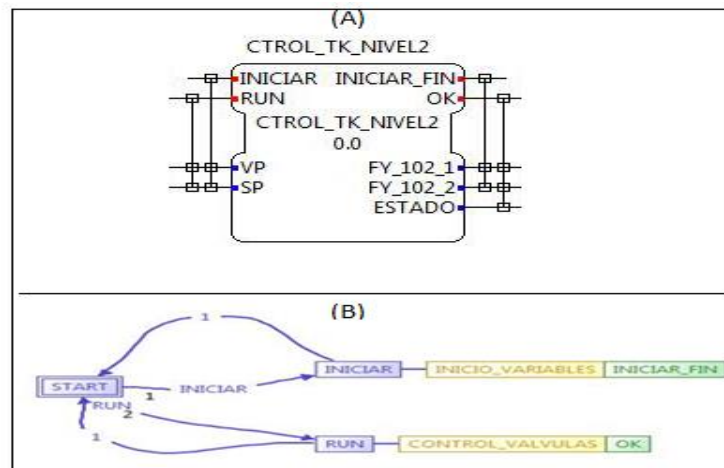
4.5.6.5. Bloques de función del dispositivo LC101-2

A continuación se presentan los bloques empleados por el dispositivo LC 102-1.

4.5.6.5.1. CTROL_TK_NIVEL2

Es el bloque encargado de realizar el control de las válvulas FY 102-2 y FY 102-1, la *Figura 71* representa el bloque, la parte (A) muestra la interfaz externa utilizada, la parte (B) muestra su respectivo grafico de control de ejecución, posteriormente se describe los algoritmos empleados en el bloque.

Figura 71. CTROL_TK_NIVEL2.



Fuente: Autores

La *Tabla 16* describe la relación entre eventos y datos del bloque CTROL_TKNIVEL2 se definen los valores de los datos empleados y algoritmos.

Tabla 16. Descripción de eventos y datos del bloque CTROL_TK_NIVEL2.

Eventos Entrada	Datos Entrada
INICIAR: Este evento permite inicializar las variables del bloque CTROL_TKNIVEL2.	VP: Esta entrada recibe el valor de la variable de proceso, en este caso el nivel en TK2.
RUN: Este evento permite iniciar el algoritmo que controlara las válvulas FY-102-1 y FY-102-1.	SP: Esta entrada recibe el valor del set point.
Eventos Salida	Datos Salida
INICIAR_FIN: Este evento indica que el bloque CTROL_TKNIVEL2 ha complementado su inicialización.	FY_102-1: Entrega el valor calculado, a la válvula FY 102-1.
OK: Este evento es activado cuando el bloque CTROL_TKNIVEL2, requiere hacer un nuevo cálculo.	FY_102-2: Entrega el valor calculado, a la válvula FY 102-2.
	ESTADO: Entrega el estado actual de la variable de proceso, este valor es enviado al controlador central.
Algoritmo: INICIO_VARIABLES	Algoritmo: CONTROL_VALVULAS
VP=0, SP=0, FY_102_1=FALSE, FY_102_2=FALSE; ESTADO= FALSE;	IF (VP=SP) THEN FY_102_2=TRUE (*Se active la válvula FY_102_2*) ESTADO=TRUE ELSE FY_102_2=FALSE (*Se desactiva la válvula FY_102_2*) ESTADO=FALSE END_IF END_IF IF (VP>SP) THEN FY_102_1=TRUE (*Se active la válvula FY_102_1*) ELSE FY_102_1=FALSE (*Se desactiva la válvula FY_102_1*) END_IF

Fuente: Autores

4.5.6.6. Bloques de función del dispositivo TC 103

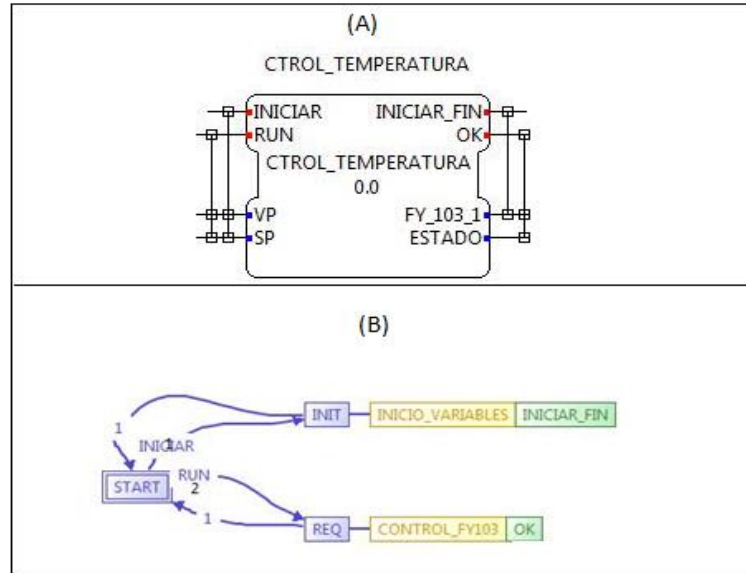
A continuación se presentan los bloques empleados por el dispositivo TC 103.

4.5.6.6.1. CTROL_TEMPERATURA

Es el bloque encargado de realizar el control de las válvulas FY_103, la *Figura 72* representa el bloque, la parte (A) muestra la interfaz externa utilizada, la parte (B)

muestra su respectivo grafico de control de ejecución, posteriormente se describe los algoritmos empleados en el bloque.

Figura 72. CTROL_TEMPERATURA.



Fuente: Autores

La Tabla 17 describe la relación entre eventos y datos del bloque CTROL_TEMPERATURA se definen los valores de los datos empleados y algoritmos.

Tabla 17. Descripción de eventos y datos del bloque CTROL_TEMPERATURA.

Eventos Entrada	Datos Entrada
INICIAR: Este evento permite inicializar las variables del bloque CTROL_TEMPERATURA.	VP: Esta entrada recibe el valor de la variable de proceso, en este la temperatura.
RUN: Este evento permite iniciar el algoritmo que controlara las válvulas FY-103	SP: Esta entrada recibe el valor del set point.
Eventos Salida	Datos Salida
INICIAR_FIN: Este evento indica que el bloque CTROL_TEMPERATURA ha complementado su inicialización.	FY_103: Entrega el valor calculado, a la válvula FY 103.
OK: Este evento es activado cuando el bloque CTROL_TEMPERATURA, requiere hacer un nuevo cálculo.	Estado: Entrega el estado actual de la variable de proceso, este valor es enviado al controlador central.
Algoritmo: INICIO_VARIABLES	Algoritmo: CONTROL_FY103
VP=0, SP=0, FY_103=FALSE;	IF (VP=SP) THEN FY_103 TRUE (*Se active la válvula FY_103*)

ESTADO=FALSE	ESTADO=TRUE ELSE FY_103=FALSE (*Se desactiva la válvula FY_103*) ESTADO=FALSE END_IF
--------------	--

Fuente: Autores

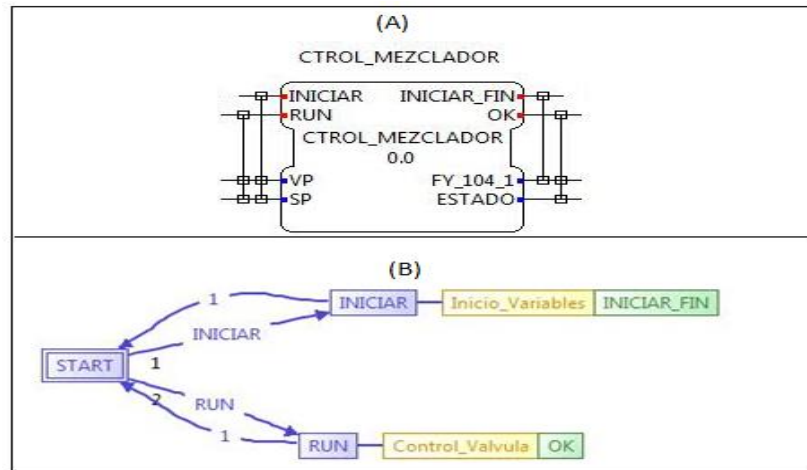
4.5.6.7. Bloques de función del dispositivo SC 104

A continuación se presentan los bloques empleados por el dispositivo SC 104.

4.5.6.7.1. CTROL_MEZCLADOR

Bloque encargado de realizar el control de la válvula FY 104-1, la *Figura 73* representa el bloque CTROL_MEZCLADOR, la parte (A) muestra la interfaz externa utilizada, la parte (B) muestra su respectivo grafico de control de ejecución, posteriormente se describe los algoritmos empleados en el bloque.

Figura 73. CTROL_MEZCLADOR



Fuente: Autores

La *Tabla 18* describe la relación entre eventos y datos del bloque CTROL_VELOCIDAD se definen los valores de los datos empleados y algoritmos.

Tabla 18. Descripción de eventos y datos del bloque CTROL_MEZCLADOR

Eventos Entrada	Datos Entrada
INICIAR: Este evento permite inicializar las variables del bloque CTROL_VELOCIDAD.	VP: Esta entrada recibe el valor de la variable de proceso, para este caso la velocidad del mezclador.
RUN: Este evento permite iniciar el algoritmo que controlara las válvulas FY-104	SP: Esta entrada recibe el valor del set point.
Eventos Salida	Datos Salida
INICIAR_FIN: Este evento indica que el bloque CTROL_VELOCIDAD ha complementado su inicialización.	FY_104: Entrega el valor calculado, a la válvula FY 104-1.
OK: Este evento es activado cuando el bloque CTROL_VELOCIDAD, requiere hacer un nuevo cálculo.	ESTADO: Entrega el estado actual de la variable de proceso, este valor es enviado al controlador central.
Algoritmo: INICIO_VARIABLES	Algoritmo: CONTROL_VALVULA
VP=0, SP=0, FY_104=FALSE ESTADO=FALSE;	IF (VP=SP) THEN delay(10000) (*Retardo de 10 min*) FY_104-1 TRUE (*Se active la válvula FY_104-1*) ESTADO=TRUE ELSE FY_104=FALSE (*Se desactiva la válvula FY_104*) ESTADO=FALSE END_IF

Fuente: Autores

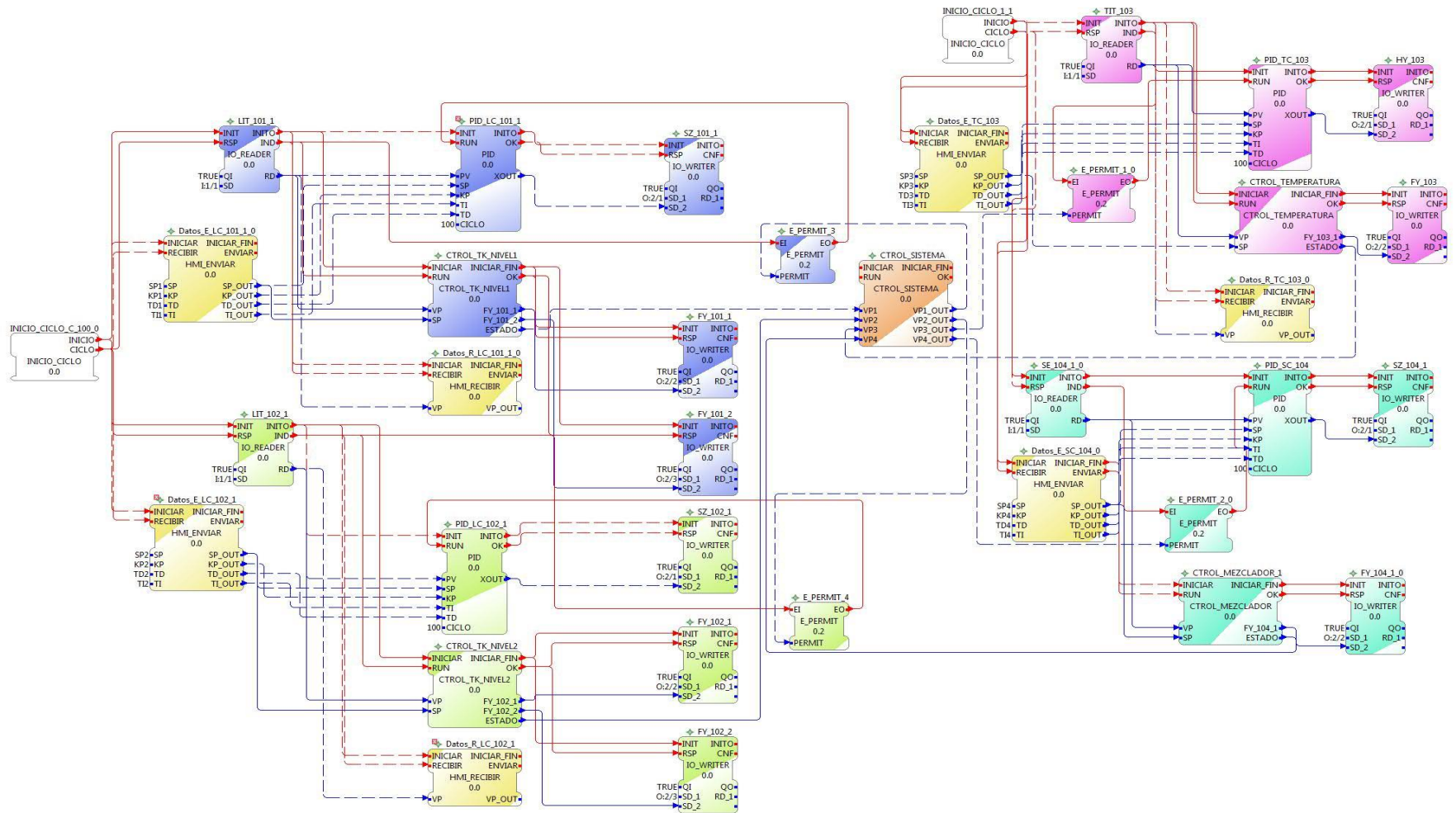
4.5.7. Aplicación de control

La figura 74 representa el modelo de aplicación para el sistema, el cual estará basado en la interconexión de los bloques de función mencionados en el ítem anterior, la conexión estará basada en la secuencia de operaciones, las líneas rojas representan los eventos con los que se activan los diferentes bloques, las líneas azules representan el flujo de datos entre bloques. El bloque Inicio_Ciclo, da la señal inicial para activar los diferentes bloques de función, los bloques de tipo IO_READER, permitirán realizar la lectura de las diferentes variables que controlara el proceso (ver tabla 8), los bloques de tipo IO_WRITER permitirán enviar los valores deseados a los actuadores. Los bloques de tipo PID serán los encargados de realizar control sobre las variables de proceso, recibirán los parámetros (SP,KP,TD,TI), por medio del bloque de tipo HMI_ENVIAR, el bloque de tipo CTROL_NIVEL-TK1, será el encargado de controlar las válvulas FY 101-1 y FY101-2, CTROL_NIVEL-TK2 será el encargado de controlar las válvulas FY

102-1 y FY102-1, el bloque CTROL_TEMPERATURA será el encargado de realizar el control de la válvula FY 103, el bloque CTROL_MEZCLADOR se emplea en la válvula FY 104-1. Finalmente los bloques de tipo HMI_RECIBIR, obtendrá el valor de las variables controladas en el proceso. El bloque de evento E_PERMIT será el encargado de activar los diferentes bloques de control PID, de acuerdo a la señal enviada por el bloque CTROL_SISTEMA.

Cabe destacar que de acuerdo a los recursos físicos con que cuenta el laboratorio de control de procesos, la aplicación estará en los diferentes dispositivos de forma centralizada, debido a que el laboratorio no cuenta con sensores ni actuadores inteligentes.

Figura 74. Aplicación de control.



Fuente: Autores

El color de los bloques de función presente en la aplicación de control indica que estarán presentes en diferentes dispositivos, *la tabla 19* muestra por color a que dispositivo pertenece cada bloque.

Tabla 19. Dispositivos empleados en la aplicación de control.

Dispositivo	Color
DISPLAY	Yellow
LC 101-1	Blue
LC 101-2	Light Green
TC 103	Pink
SC 104	Cyan
C 100	Orange

Fuente: Autores

La siguiente etapa del proceso es distribuir la aplicación entre los diferentes recursos con que cuenta cada dispositivo, de acuerdo a la funcionalidad definida en este caso cada dispositivo tendrá un recurso.

4.5.8. MODELO DE RECURSO.

Los recurso definidos por IEC 61499, intentan distribuir la aplicación antes diseñada, permitiendo que cada parte de esa aplicación se ejecute de manera independiente. Para lo cual cada recurso tendrá que emplear en cada dispositivo los bloques SUBSCRIBE y PUBLISH, con IP pertenecientes a la red de área local configurada.

La *Tabla 20*, describe los diferentes recursos que tendrán los dispositivos y la unidad que tienen a cargo.

Tabla 20. Dispositivos con su respectivo recurso.

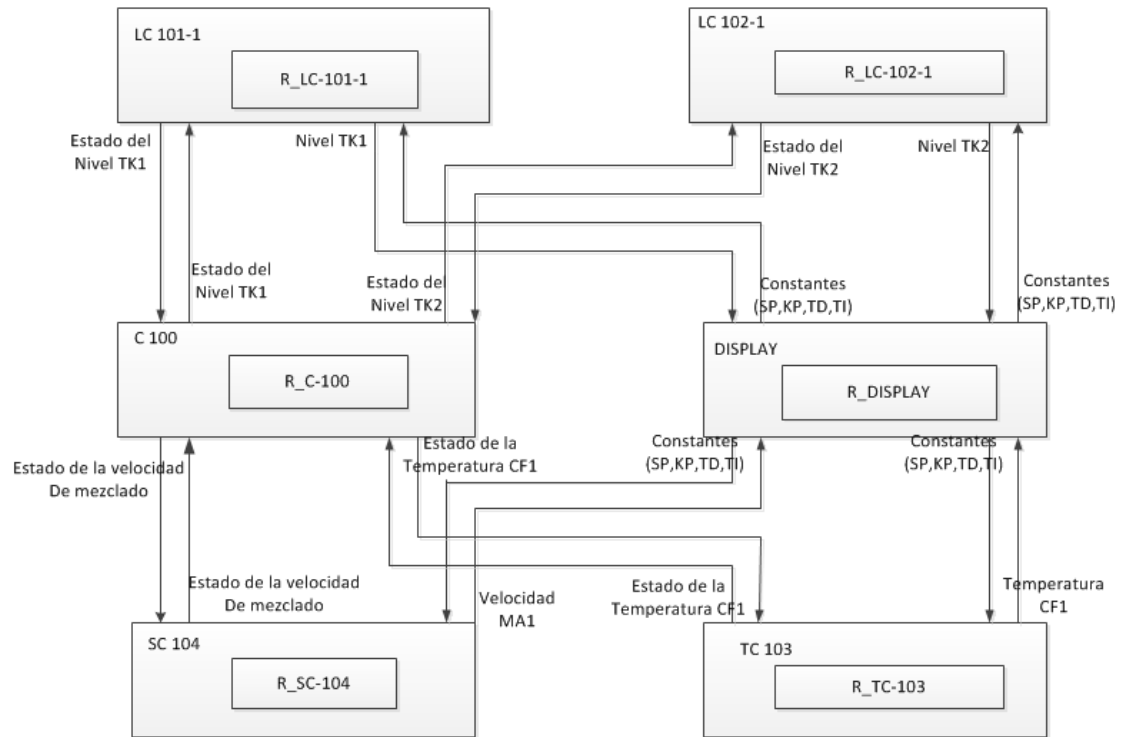
Dispositivo	Recurso	Unidad
C 100	R_C-100	
DISPLAY	R_DISPLAY	
LC 101-1	R_LC_101-1	Unidad A
LC 102-1	R_LC_102-1	Unidad B
TC 103	R_TC-103	Unidad C
SC 104	R_SC-104	Unidad D

Fuente: Autores

La *Figura 75* describe los dispositivos empleados con su respectivo recurso, se aprecian los diferentes datos que son intercambiados entre cada dispositivo, estos

datos serán finalmente los que se envíen por los bloques PUBLISH Y SUSCRIBE en cada recurso.

Figura 75. Flujo de datos entre recursos.



Fuente: Autores

4.5.9. CONFIGURACION DE COMUNICACION

Para la comunicación entre dispositivos se emplean bloques PUBLISH/SUSCRIPTOR, los cuales utilizarán el protocolo UDP/IP[51] mediante el empleo de direcciones IP multicast y un puerto. El bloque de función PUBLISH enviará datos a través de la red con dirección IP: puerto y el bloque de función SUSCRIPTOR recibirá la información con la misma dirección IP. Las direcciones IP multicast tienen un rango de 224.0.0.0 - 239.255.255.255, la selección del puerto es libre. La *Figura 76* muestra las diferentes direcciones IP escogidas, indicando que dirección utilizará cada dispositivo para enviar información a los diferentes dispositivos con que interactúa de acuerdo a la

Figura 76. Configuración de direcciones IP.

	LC 101-1	LC 101-2	TC 103	SC 104
DISPLAY	225.0.0.0:01	225.0.0.0:02	225.0.0.0:03	225.0.0.0:04
	LC 101-1	LC 101-2	TC 103	SC 104
C 100	225.0.0.1:01	225.0.0.1:02	225.0.0.1:03	225.0.0.1:04
	DISPLAY	C 100		
LC 101-1	225.0.0.2:01	225.0.0.2:02		
	DISPLAY	C 100		
LC 101-2	225.0.0.3:01	225.0.0.3:02		
	DISPLAY	C 100		
TC 103	225.0.0.4:01	225.0.0.4:02		
	DISPLAY	C 100		
SC 104	225.0.0.5:01	225.0.0.5:02		

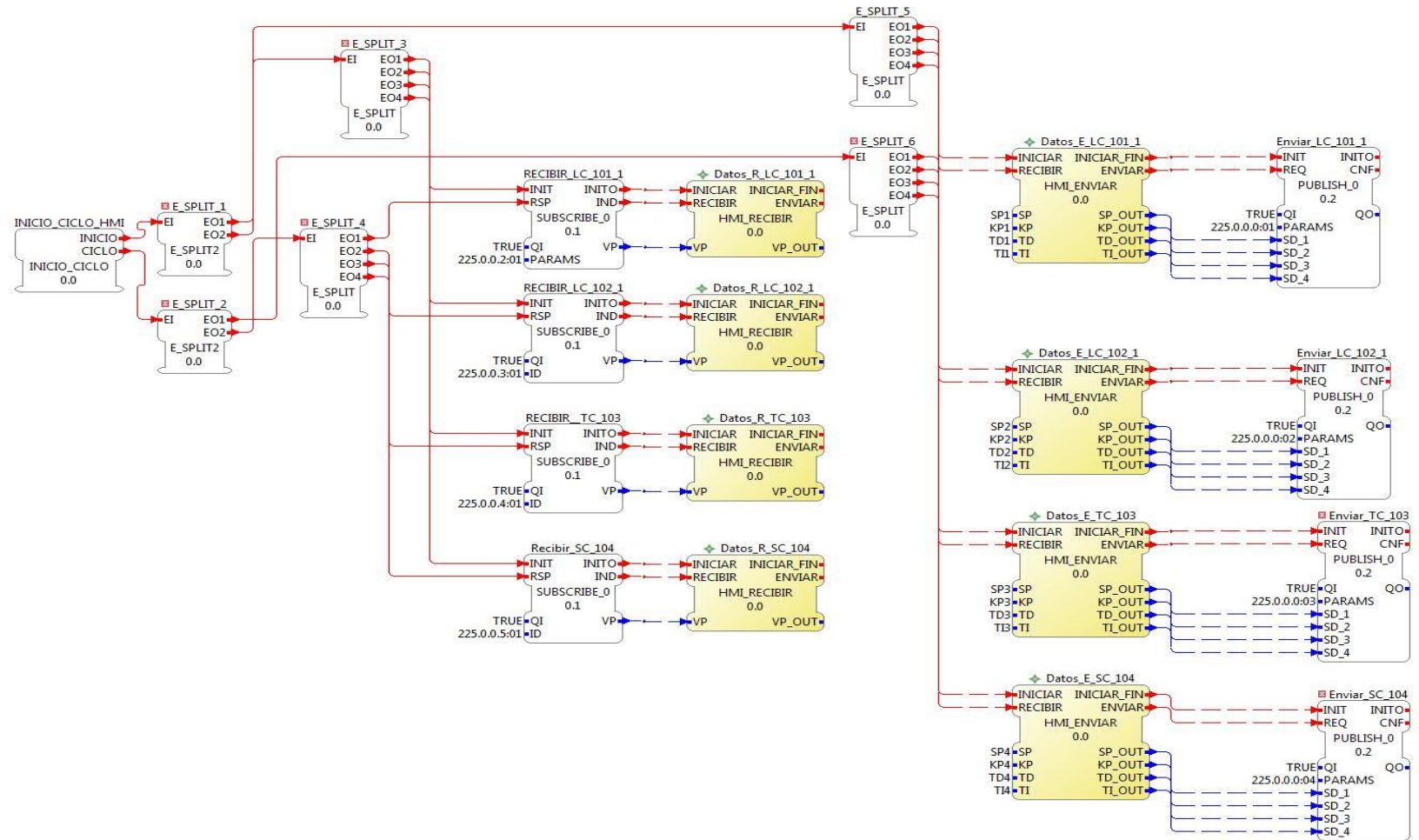
Fuente: Autores

Se procede a describir los elementos que integran cada recurso.

A) RECURSO R_DISPLAY

La *Figura 77* permite representar el recurso “**R_DISPLAY**” empleado por el dispositivo DISPLAY, utiliza 4 bloques de tipo HMI_RECIBIR, con el fin de obtener el valor actual del estado de las variables de proceso, los cuales llegan al recurso por medio de bloques de tipo SUSCRIBE. También emplea bloques de tipo HMI_ENVIAR, estos permiten enviar las constantes del control PID, a los controladores PLC por medio del bloque PUBLISH, empleando las IPs adecuadas.

Figura 77. Recurso_DISPLAY



Fuente: Autores

B) Recurso R_C-100

La *Figura 78* permite representar el recurso “**R_C-100**”, implementado por el dispositivo C 100. Utiliza el bloque de tipo CTROL_SISTEMA, con el fin de obtener el estado del proceso entregado por cada controlador, estos datos son recibidos por medio de bloques SUSCRIBE, y a partir de estos estados se actualiza el valor de estado de los controladores, de acuerdo a la secuencia de operaciones por medio del bloque PUBLISH.

C) Recurso R_LC-101-1

La *Figura 79* permite representar el recurso “**R_LC-101-1**” utilizado por el dispositivo LC 101-1, relacionándolo con los elementos empleados en la unidad A. El bloque LIT 101-1 de tipo IO_READER permite obtener el valor actual del nivel en el TK1, ese dato será enviado al bloque de tipo PID, encargado de calcular el valor que requiere el variador de velocidad SZ 101-1.

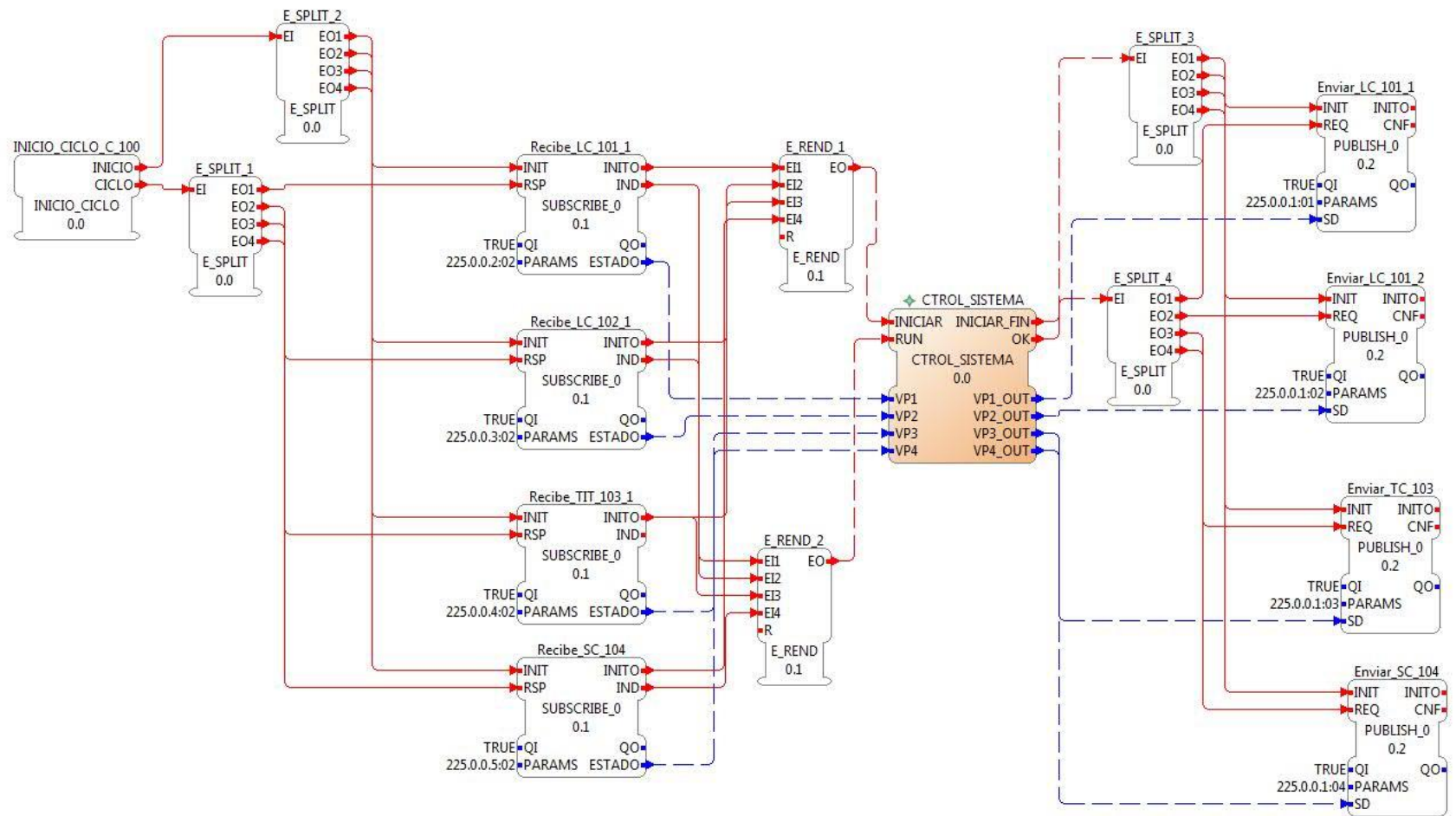
Por medio del bloque RECIBE_HMI de tipo SUSCRIBE, el recurso recibe los valores de las constantes del PID. El bloque CTROL_TK_NIVEL1 se encarga de realizar el control ON/OFF, permitiendo activar o desactivar las válvulas FY 101-1 y FY 101-2, enviando el estado al controlador C-100 por medio del bloque ENVIAR_C_100 de tipo PUBLISH.

D) Recurso R_LC-102-1

La *Figura 80* permite representar el recurso “**R_LC-102-1**” empleado por el dispositivo LC 102-1, relacionándolo con los elementos que integran la unidad B. El bloque LIT 102-1 de tipo IO_READER permite obtener el valor actual del nivel en el TK2, dato que es enviado al bloque de tipo PID, encargado de calcular el valor que requiere el variador de velocidad SZ 102-1. Por medio del bloque RECIBE_HMI de tipo SUSCRIBE se obtienen las constantes del controlador.

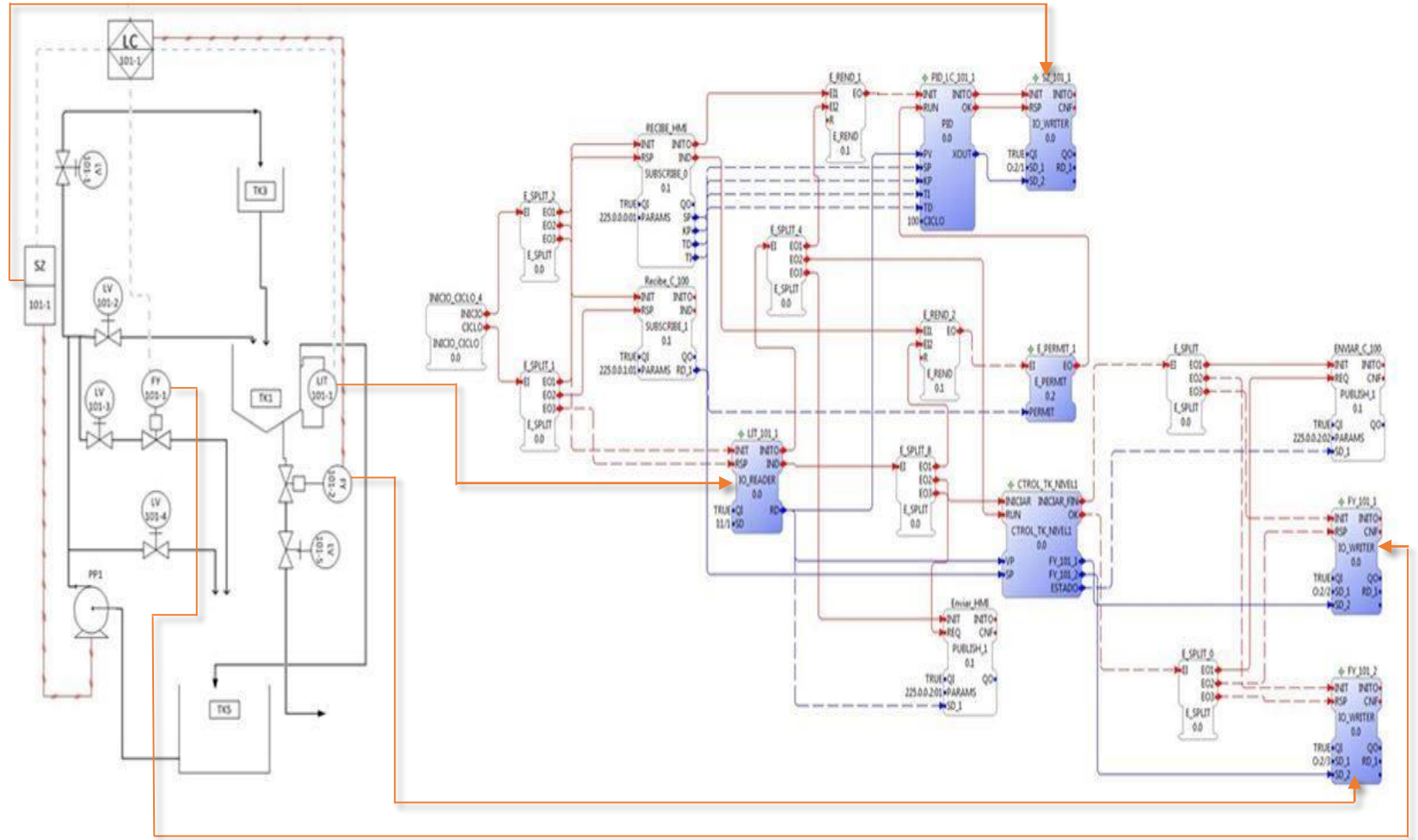
El bloque CTROL_TK_2 realiza el control a válvulas FY 102-1 y FY 102-2, permitiendo activar o desactivar, enviando el estado al controlador C 100 por medio del bloque ENVIAR_C_100 de tipo PUBLISH.

Figura 78. Recurso R_C-100.



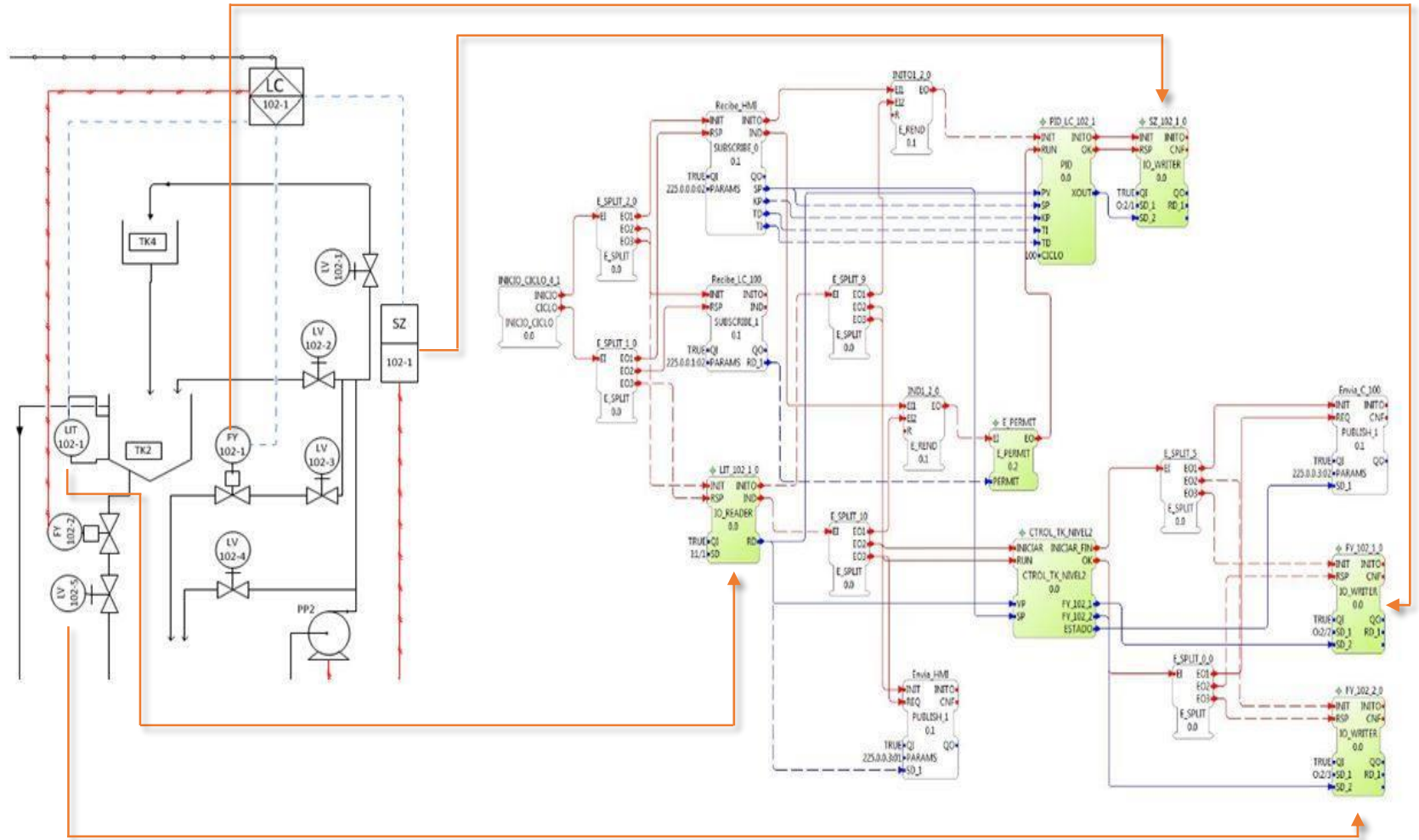
Fuente: Autores

Figura 79. Recurso R_LC-101-1



Fuente: Autores

Figura 80. Recurso R_LC-102-1



Fuente: Autores

E) Recurso R_TC-103

La *Figura 81* permite representar el recurso “**R_TC-103**” utilizado por el dispositivo TC 103, el cual está relacionado con los elementos de la unidad D.

El bloque TIT 103 de tipo IO_READER permite obtener el valor actual de la temperatura en el CF1, el cual será enviado al bloque PID_TC_103 de tipo PID, este se encarga de calcular el valor que requiere la resistencia, ese valor es enviado al bloque HY 103. Por medio del bloque RECIBE_HMI de tipo SUSCRIBE, el controlador recibe los valores de las constantes del PID. El bloque CTROL_TEMPERATURA se encarga de activar o desactivar la válvula FY 103.

F) Recurso R_SC-104

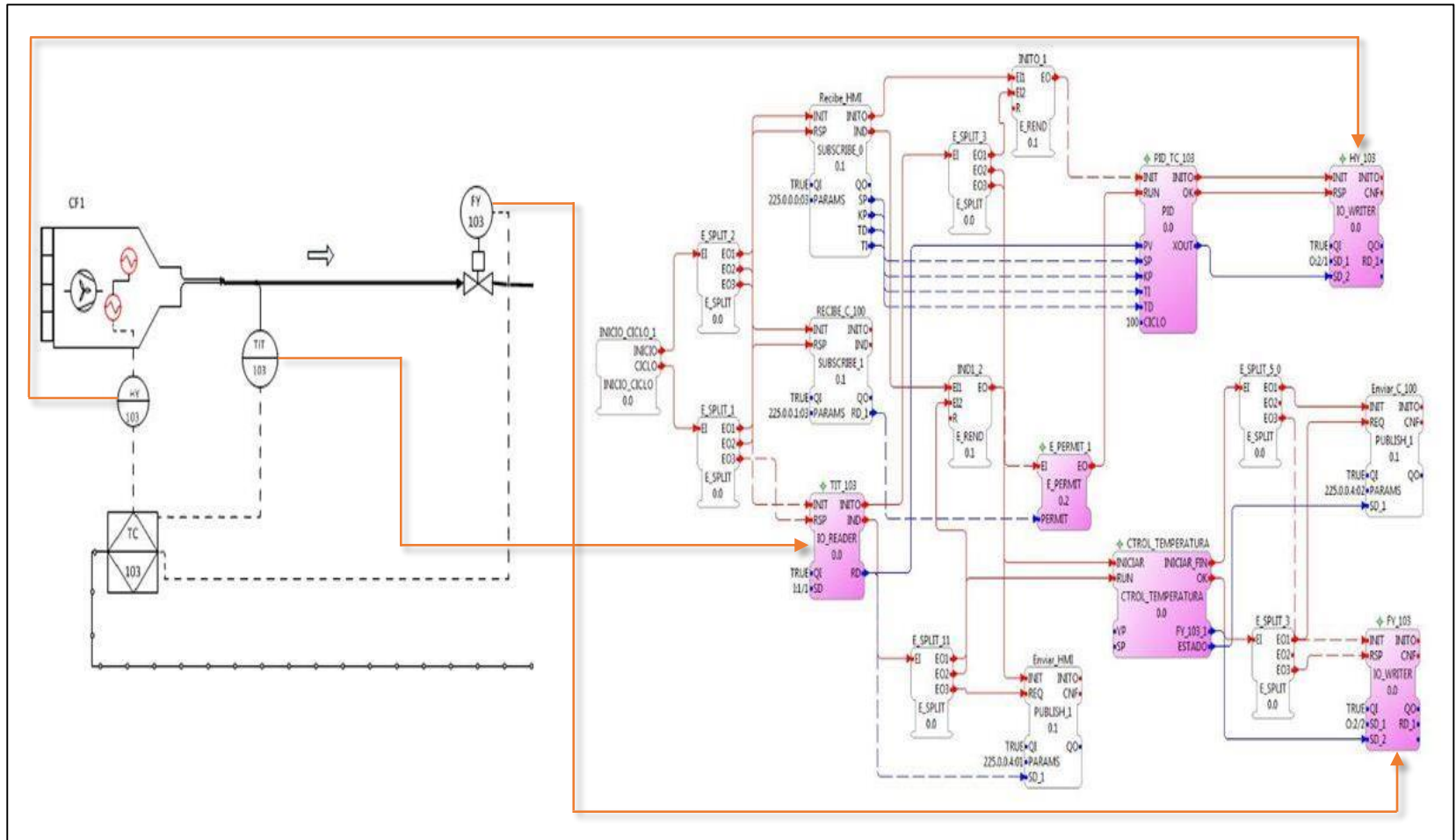
La *Figura 82* permite representar el recurso “**R_SC-104**” utilizado por el dispositivo SC-104, de acuerdo a los elementos que integran la unidad C. El bloque SE_104_1 de tipo IO_READER permite obtener el valor actual de la velocidad en el MX1, que será enviado al bloque PID_104_1 de tipo PID, este se encarga de calcular el valor que requiere el variador de velocidad SZ 104_1. Por medio del bloque Recibir_HMI de tipo SUSCRIBE, el recurso recibe los valores de las constantes del PID, del dispositivo DISPLAY.

El bloque CTROL_MX1 permite activar o desactivar la válvula FY 104-1, enviando el estado de la válvula al controlador C 100 por medio del bloque Enviar_C_100 de tipo PUBLISH.

4.5.10. Modelo de dispositivo/Sistema.

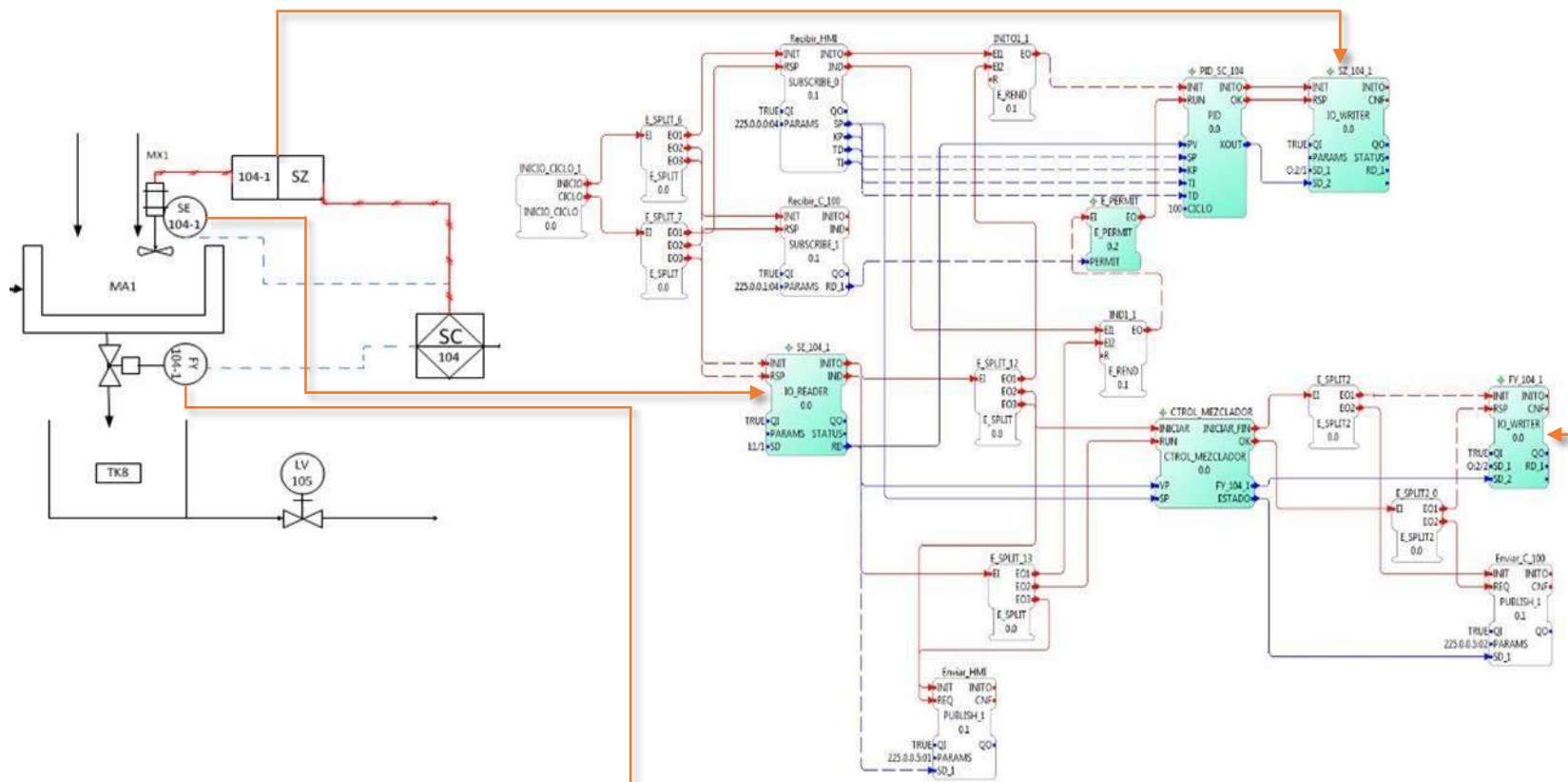
Finalmente la *Figura 83* representa el modelo del sistema con los dispositivos involucrados, se aprecia los recursos de cada dispositivo, con comunicación por medio de una red Ethernet.

Figura 81. Recurso R_TC-103



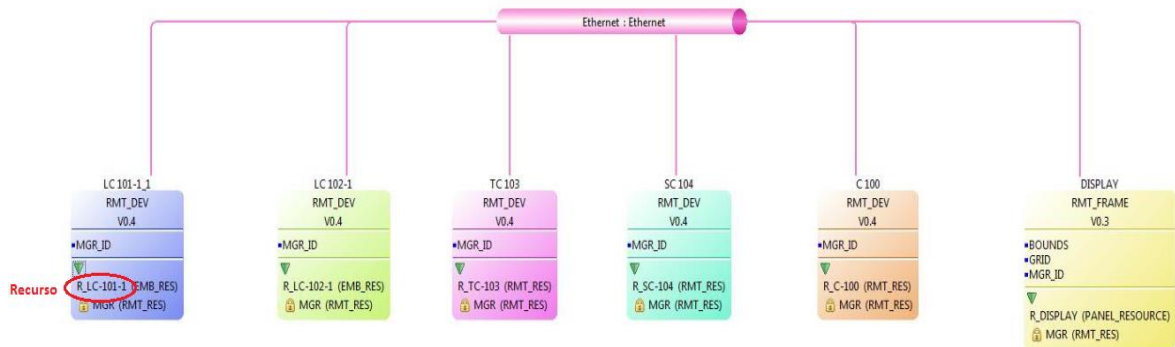
Fuente: Autores

Figura 82. Recurso R_SC-104



Fuente: Autores

Figura 83. Modelo de sistema.



Fuente: Autores

4.5.11. Lista de Chequeo

Finalmente en la *Tabla 21* se presenta una lista de chequeo con el fin de verificar las características del modelo desarrollado con IEC 61499, de acuerdo a las necesidades del proceso identificado.

Tabla 21. Lista de chequeo.

N°	Ítem	Cumple	No cumple	Observación
1	El modelo desarrollado en base al estándar IEC 61499 permite activar y desactivar la válvula FY 101-2.	SI		El bloque que realiza el control es el CTROL_TK_NIVEL1
2	El modelo permite activar y desactivar la válvula FY 102-2.	SI		El bloque que realiza el control es el CTROL_TK_NIVEL2
3	El modelo permite activar y desactivar la válvula FY 103.	SI		El bloque que realiza el control es CTROL_TEMPERATURA
3	El modelo permite activar y desactivar la válvula FY 104-1.	SI		El bloque que realiza el control es CTROL_MEZCLADOR
4	El modelo realiza control sobre el nivel en el TK1.	SI		El bloque que realiza el control es PID_LC_101_1
5	El modelo realiza control sobre el nivel en el TK2.	SI		El bloque que realiza el control es PID_LC_102_1
6	El modelo realiza control sobre la velocidad de MX1.	SI		El bloque que realiza el control es PID_SC_104

7	Se realiza control sobre la temperatura de CF1.	SI		El bloque que realiza el control es PID_TC_103
9	Se distribuye el modelo de aplicación entre los diferentes recursos.	SI		

Fuente: Autores

CAPÍTULO V

5. CONCLUSIONES

El concepto de SCADA y de DCS tiende a parecerse cada vez más debido al continuo avance de las tecnologías de comunicación, control y supervisión. Queriendo siempre llegar, a la mejor forma de poder controlar los procesos de manufactura

En la actualidad solo hay una línea muy delgada que separa estos dos sistemas de control, y es la distribución de los lazos de control en dispositivos inteligentes, tales como sensores y actuadores

El modelo software que genera IEC 61131 no es el adecuado para sistemas DCS, por ello es importante implementar el nuevo y enfocado estándar para sistemas de control distribuido IEC 61499 con los diferentes modelos que expone (modelo del sistema, modelo de dispositivo, entre otros)

La programación que comúnmente se hace de forma cíclica, es decir tipo escalera (Ladder) para los controladores en un SCADA presenta dificultades en la programación, siendo ejecutados línea por línea. Además la gran cantidad de código ha hecho que el mantenimiento de estos programas sea complicado y poco flexible.

Es posible ejecutar subprocesos de forma paralela debido a los bloques de función que genera IEC 61499, disminuyendo tiempos de fabricación, aumentando la flexibilidad de cada una de sus líneas de producción.

El procedimiento generado busca la implementación del estándar IEC 61499 bajo el modelado y seguimiento hecho del proceso, especificando los tipos de bloques de función que se deben usar de acuerdo al diagrama P&ID, arquitectura y documentación del proceso.

La aplicación del estándar a nivel industrial es nula, las herramientas con que se cuenta para poder soportarlo, se han aplicado solo en casos de estudio a nivel académico, por parte de laboratorios, empleando plataformas sencillas y pequeñas, esto implica la no aceptación del estándar.

Las futuras investigaciones deberán estar enfocadas en la implementación del estándar IEC 61499 mediante herramientas que cumplan con los requisitos definidos en su parte 2 y 3.

6. BIBLIOGRÁFICA

- [1] Institute Iacocca, "21. Century manufacturing enterprise strategy: an industry- led view."
- [2] A. Zoitl and V. Vyatkin, "IEC 61499 architecture for distributed automation: The 'glass half full' view," *IEEE Ind. Electron. Mag.*, vol. 3, no. 4, pp. 7–22, 2009.
- [3] B. Forero, "Caracterizacion del Sistema de Control Distribuido DCS HONEYWELL EXPERION de la unidad central del norte de la gerencia refinera de Barrancabermeja de Ecopetrol S.A," PONTIFICIA BOLIVARIANA, 2011.
- [4] O. Rojas Alvarado, "Cad / cam / cim integración empresarial," Popayán, 2014.
- [5] I. Al and C. D. E. Procesos, "Tema 4: control de procesos industriales. control distribuido.," 2014. [Online]. Available: <https://www.depeca.uah.es/depeca/repositorio/asignaturas/30387/Tema4.pdf>
- [6] J. P. Ferrari, "Sistemas de Control Distribuidos," Nacional de Rosario, 2005.
- [7] D. Bailey and E. Wright, "Practical SCADA for Industry," *Elsevier Sci.*, p. 304, 2003.
- [8] K. Stouffer, J. Falco, and K. Scarfone, "Guide to Industrial Control Systems (ICS) Security Recommendations of the National Institute of Standards and Technology," *NIST Spec. Publ.*, pp. 1–157, 2007.
- [9] A. O Pinzón, "ESTADO ACTUAL Y FUTURO DE LA INGENIERÍA DE CONTROL," *Univ. Pontif. Boliv. Secc. Bucaramanga*, 2010.
- [10] A. Rodriguez, *SISTEMAS SCADA*, 3ra ed. 2012.
- [11] J. Romanosa, D. Gallego, and R. Pacheco, "MINIPROYECTO AUTOMATIZACION INDUSTRIAL (AUTI)." p. 66, 2004.
- [12] A. Garcia and P. Urzola, "Sistema SCADA. Supervisión, Control y Adquisición de datos," *slide share*. [Online]. Available: <http://es.slideshare.net/nestorcusco/sistema-scada-24902242>.
- [13] E. Bernal, "SCADA & IED's," *IEEE Industrial Electronics Magazine*, p. 56, 2009.
- [14] E. Diaz and O. Rojas Alvarado, "Proyecto de Automatización I," 2013.
- [15] M. Vladimir, "Implementación de Sistemas Empotrados de Control Distribuidos bajo el Estándar IEC-61499," Universidad del Pais Vasco, 2013.
- [16] R. D'Andrea and G. E. Dullerud, "Distributed control design for spatially interconnected systems," *IEEE Trans. Automat. Contr.*, vol. 48, no. 9, pp. 1478–1495, 2003.
- [17] V. L. Trevathan, "A Guide to the Automation Body of Knowledge (2nd

- Edition),” *Isa*, 2006. [Online]. Available: <https://goo.gl/8RppLs>.
- [18] J. Villajulca, “Introducción a los DCS,” *Intrumentacion y control.net*, 2011. [Online]. Available: <http://www.instrumentacionycontrol.net/cursos-libres/automatizacion/curso-sistemas-de-control-distribuido-dcs/item/413-introduccion-a-los-dcs-sistemas-de-control-distribuido.html>.
- [19] ABB, “System 800xa,” *ABB PAGE*. [Online]. Available: <http://new.abb.com/control-systems/system-800xa>.
- [20] Emerson, “Emerson Process Management,” *Emerson page*. [Online]. Available: <http://www2.emersonprocess.com/en-us/brands/edservices/automationsystems/pages/automationsystems.aspx>.
- [21] Siemens, “SIMATIC PCS 7,” *The distributed control system for integrated automation*. [Online]. Available: <http://w3.siemens.com/mcms/process-control-systems/en/distributed-control-system-simatic-pcs-7/pages/distributed-control-system-simatic-pcs-7.aspx>.
- [22] Honeywell, “Experion PKS Orion – Process Control Beyond Distributed Control Systems,” *honeywell*. [Online]. Available: <https://www.honeywellprocess.com/en-US/explore/products/control-monitoring-and-safety-systems/integrated-control-and-safety-systems/experion-pks/Pages/experionpks.aspx>.
- [23] Yokogawa, “Yokogawa electric corporation,” *Yokogawa 100th Aniversary*. [Online]. Available: <http://www.yokogawa.com/dcs/centumvp/dcs-vp-index-en.htm>.
- [24] Siemens, “Siemens,” *TI 505*. .
- [25] M. Van Steen and A. S. Tanenbaum, *Sistemas Distribuídos: principios e paradigmas*, vol. paperback. 2007.
- [26] J. M. Hurtado, “Introducción a las Redes de Comunicación Industrial,” *Electr. I.E.S Him. - Linares*.
- [27] A. Rosado Muñoz, “SISTEMAS INDUSTRIALES DISTRIBUIDOS: Una filosofía de automatización,” *Apunt. Teor.*, pp. 3–17.
- [28] C. H. Yang and V. Vyatkin, “Design and validation of distributed control with decentralized intelligence in process industries: A survey,” *IEEE Int. Conf. Ind. Informatics*, pp. 1395–1400, 2008.
- [29] V. Vyatkin, “IEC 61499 as Enabler of Distributed and Intelligent Automation: State of the Art Review,” *IEEE Trans. Ind. Informatics*, vol. 7, pp. 768–781, 2011.
- [30] F. García and L. Casas, “Diseño E Implementación De Un Sensor Inteligente De Nivel,” *U Dist. Fr. Jose Caldas*, pp. 1–11, 2011.
- [31] J. Villajulca, “Componentes funcionales de un DCS: inspeccionando el esqueleto del sistema,” *Intrumentacion y control.net*, 2011. [Online]. Available: <http://www.instrumentacionycontrol.net/cursos-libres/automatizacion/curso-sistemas-de-control-distribuido-dcs/item/495->

componentes-funcionales-de-un-dcs-inspeccionando-el-esqueleto-del-sistema.html.

- [32] R. Lewis, "Modelling Control Systems Using IEC 61499, Applying function blocks to distributed systems," *IEE Publishing*, no. ISBN: 0 85296 796 9, 2001.
- [33] (IEC) International Electrotechnical Commission, "IEC 61499: Function blocks, Part 1-4.," *IEC 61499*, 2005. [Online]. Available: www.iec.ch.
- [34] N. Ferney, S. Velandia, J. David, and R. Fonseca, "Distribuido Sobre La Entrenadora Has 200," *Universidad Distrital*, 2012. .
- [35] INAMEX AUTOMATION S.A. de C.V., "INAMEX AUTOMATION," *Sistemas de Control Distribuido*, 2014. [Online]. Available: <http://www.inamex-fp.com/producto/81/sistemas-de-control-distribuido/>.
- [36] R. N. Nagel and R. Dove, "21st Century Manufacturing Enterprise Strategy: An Industry-Led View," *Iacocca Inst.*, pp. 1–58, 1991.
- [37] V. Vyatkin, H. Hanisch, and S. Karras, "IEC61499 as an Architectural Framework for Integration of Formal Models and Methods in Practical Control Engineering," *Structure*, 2002.
- [38] T. Strasser, C. Sünder, A. Zoitl, M. N. Rooker, and J. E. J. Brunnenkreef, "Enhanced IEC 61499 device management execution and usage for downtimeless reconfiguration," *IEEE Int. Conf. Ind. Informatics*, vol. 2, pp. 1163–1168, 2007.
- [39] A. Zoitl, C. Sunder, and I. Terzic, "Dynamic Reconfiguration of Distributed Control Applications with Reconfiguration Services based on IEC 61499," *Distrib. Intell. Syst. Collect. Intell. Its Appl.*, no. 2006. DIS 2006, IEEE Workshop on, June 2006, pp. 109–114, 2006.
- [40] K. Thramboulidis, "The function block model in embedded control and automation from IEC61131 to IEC61499," *WSEAS Trans. Comput.*, vol. 8, no. 9, pp. 1597–1609, 2009.
- [41] G. Doukas and K. Thramboulidis, "Implementation model alternatives for IEC 61499 function block networks," *IEEE Int. Conf. Ind. Informatics*, pp. 295–300, 2008.
- [42] W. Yang, "Implementation of IEC61499 Distributed Function Block Architecture for Industrial Measurement and Control Systems (IPMCS)," NATIONAL UNIVERSITY OF SINGAPORE, 2002.
- [43] K. Thramboulidis, "IEC 61499 vs. 61131: A Comparison Based on Misperceptions," *arXiv*, vol. 2013, no. August, pp. 3–5, 2013.
- [44] E. Querol, J. Ariel, A. Estruch, and F. Romero, "Norma IEC-61499 para el control distribuido. aplicación al CNC.," *Actas XXXV Jornadas Automática, 3-5 septiembre 2014, Val.*, no. ISBN-13: 978 84 697 05 89 6, pp. 1–8, 2014.
- [45] M. D. Khediya, "Reconfigurable and Distributed Process Control System Compliant IEC 61499 Function Blocks Model Structure," vol. 3, no. 2, pp.

274–278, 2014.

- [46] R. Diaz, M. Holm, and G. Adamson, “Development of a Manufacturing Cell in Compliance With IEC 61499 :,” UNIVERSITY OF SKOVDE, 2012.
- [47] L. Yoong, P. Roop, Z. Bhatti, and M. Y. Kuo, *Model-Driven Design Using IEC 61499*, ISBN 978-3., no. 17. Switzerland, 2015.
- [48] ISA Committee SP88, “Batch Control Part1: Models and Terminology, ISA - The Instrumentation Systems and Automation Society.” 1995.
- [49] E. Martinez, C. Sanchez, and J. Betancourt, “Proyecto de Automatizacion, Modelo físico basado en ISA 88 para la produccion de papel Tissue.” Popayán, p. 17, 2013.
- [50] G. J. CARLOS MERCÉ, “DESARROLLO DE UN SISTEMA DE CONTROL EN RED MEDIANTE EL ESTANDAR IEC-61499,” *Esc. Super. Tecnol. Y CIENCIAS Exp.*, p. 87, 2015.
- [51] A. Zoitl and T. Strasser, *Distributed Control Applications*, Taylor & F. 2016.
- [52] M. Zhou and K. Venkatesh, “Modeling, simulation, and control of flexible manufacturing systems,” *World Scientific Publishing.*, vol. 6. Series in Intelligent Control and Intelligent Automation 6, 1999.
- [53] A. Aguilar and C. Castro, “FAMILIA Y RECONOCIMIENTO PLANTA MULTIVARIABLE.” pp. 1–13, 2013.
- [54] F. Franco, “FAMILIRIZACION Y CONOCIMIENTO DE LA PLANTA DE EVENTOS DISCRETOS.” pp. 1–9, 2013.
- [55] F. Franco, “FAMILIRIZACION Y RECONOCIMIENTO PLANTA DE TEMPERATURA.” pp. 1–10, 2013.
- [56] L. International Electrotechnical Commission IEC, “Programmable Controllers Part 1. General - IEC61131-3 (third Edition),” *IEC Std*, 2013.
- [57] M. Indriago, “Desarrollo de un Procedimiento para Diseño de Sistemas de Control en Procesos con un Modelo de Integración Basado en Holones.,” UNIVERSIDAD DE LOS ANDES, 2011.
- [58] HOLOBLOC Inc., “Function Block Development Kit (FBDK) 2.4,” *Function Block Development Kit*, 2016. .
- [59] M. G. Sánchez, “Implementación de sistemas empotrados de control distribuidos bajo el estandar IEC-61499.” pp. 1–57, 2013.
- [60] 4diac, “4diac,” 2016. [Online]. Available: <https://eclipse.org/4diac/>.
- [61] NXTCONTROL, “nxtSTUDIO,” 2016. [Online]. Available: <http://www.nxtcontrol.com/engineering/>.
- [62] “ISAGRAF,” 2016. [Online]. Available: <http://www.isagraf.com/index.htm>.
- [63] “nxtIECRT,” 2016. [Online]. Available: <http://www.nxtcontrol.com/en/control/>.
- [64] “WAGO,” 2015. [Online]. Available: <http://global.wago.com/en/products/new-items/overview/pfc200-1.jsp>.

- [65] "nxtDCSmini," 2016. [Online]. Available:
<http://www.nxtcontrol.com/en/control-solution/>.
- [66] "QORIQ_ETHERCAT," 2012. [Online]. Available:
http://www.qnx.com/partners/QorIQ_P1025_PLC_Reference_Platform_FS_LR.pdf.
- [67] "NetmasterII," 2016. [Online]. Available:
<http://www.elsist.it/WebSite/Html/English/Products/Hardware/Obsoletes/Netsyst/EnNetmasterII.php>.