

**APLICACIÓN DE LA ARQUITECTURA PROPUESTA EN EL ESTÁNDAR IEC
61499 A UN PROCESO DE MANUFACTURA RECONFIGURABLE**



**ERIKA NATALIA GARZÓN YAMÁ
JUAN GABRIEL UMAÑA GONZÁLEZ**

Tesis de pregrado en Ingeniería en Automática Industrial

Director:
Ermilso Díaz Benachí
Ingeniero en Automática Industrial

**Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Electrónica Instrumentación y Control
Ingeniería en Automática industrial
Popayán, Septiembre de 2016**

**APLICACIÓN DE LA ARQUITECTURA PROPUESTA EN EL ESTÁNDAR IEC
61499 A UN PROCESO DE MANUFACTURA RECONFIGURABLE**

ERIKA NATALIA GARZÓN YAMÁ
JUAN GABRIEL UMAÑA GONZÁLEZ

Trabajo de grado presentado a la Facultad de Ingeniería Electrónica y
Telecomunicaciones de la Universidad del Cauca para la obtención del Título de:
Ingeniera en Automática Industrial

Director:
Ermilso Díaz Benachí
Ingeniero en Automática Industrial

**Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Electrónica Instrumentación y Control
Ingeniería en Automática industrial
Popayán, Septiembre de 2016**

Agradecimientos

Gracias a Dios por reconfortarnos en los momentos más difíciles y guiar con su luz las horas de incertidumbre.

Gracias a nuestras familias por su esfuerzo y compromiso incansable con nuestra formación tanto humana como académica.

Gracias a Alois Zoitl por su guía y consejos. A Cristian Sánchez, Jorge Betancourt y Luis Cruz por su constante disposición y ayuda.

Resumen Estructurado

Este trabajo pretende alcanzar los siguientes objetivos:

Objetivo General

Modelar un sistema de manufactura reconfigurable aplicando la arquitectura de modelos propuestos en el estándar IEC 61499-1

Objetivos Específicos

- Establecer los criterios necesarios para la definición de un proceso de manufactura reconfigurable.
- Caracterizar el estándar IEC 61499 para la aplicación en procesos de manufactura reconfigurable.
- Simular en una herramienta software el modelo del proceso de manufactura reconfigurable bajo el estándar IEC 61499-1.

Con el fin de cumplir estos objetivos, la monografía se ha dividido en cinco capítulos así:

En el capítulo 1, se da una introducción a lo que es un sistema de manufactura reconfigurable o RMS, sus objetivos y características que deberán ser más tarde alcanzadas por el modelo en este trabajo propuesto.

En el capítulo 2, se introducen los principios más relevantes del estándar IEC 61499, los diferentes modelos que componen su propuesta de arquitectura y las necesidades que este debe cumplir para dar soporte a un RMS.

En el capítulo 3, se propone el caso de estudio seleccionado, se describen las operaciones de manufactura que debe ser capaz de realizar el sistema y los productos que debe fabricar a partir de ellas.

En el capítulo 4, se desarrolla el modelado del sistema, los diferentes modelos de la arquitectura del estándar son aplicados al caso de estudio y se desarrolla la simulación, que será utilizada para verificar la reconfigurabilidad de los modelos hechos en el capítulo anterior, obteniendo así los resultados del trabajo.

En el capítulo 5, se presentan las conclusiones del trabajo y las posibilidades de futuros trabajos de investigación con base en este.

Palabras clave: sistemas de manufactura Reconfigurable - RMS, IEC 614499, Sistemas de Control Distribuido - DCS, Bloque de Función – FB.

Structured abstract

This paper aims to achieve the next goals:

General objective

Model a reconfigurable manufacturing system using the architecture proposed by IEC 61499.

Specific objective

- Establish the criteria to define a reconfigurable manufacturing system.
- Characterize the standard IEC 61499 to use it on reconfigurable manufacturing systems.
- Simulate in a software tool the model of the reconfigurable manufacturing system under the IEC 61499 standard.

In order to accomplish these objectives this paper has been divided into six chapters:

In chapter 1 a brief introduction to what is a reconfigurable manufacturing system is given, its goals and characteristics which will be accomplished by the model proposed on this paper later.

In chapter 2 the most relevant principles of IEC 61499 are introduced, the model that compose the architecture it proposes and the needs the standard must fulfill to support a RMS.

In chapter 3 the case study is chosen, the manufacturing operations the system must do are describes and the products manufactured as well.

In chapter 4 the modeling of the system is developed, the different models of the standard are applied to the case study and the simulation is made to verify the reconfigurability, from that we obtain the results of this work.

In chapter 5 the conclusions area presented and the future investigation work is proposed.

Key words: Reconfigurable Manufacturing System- RMS, IEC 614499, Distributed Control Systems, Function Block.

Contenido

	Pág.
Agradecimientos	3
Resumen Estructurado	4
Structured abstract.....	5
Contenido	6
Lista de figuras.....	8
Lista de tablas.....	10
Abreviaturas.....	11
Capítulo 1: ¿Qué son los Sistemas de Manufactura Reconfigurable – RMS?	12
1.1 INTRODUCCIÓN.....	12
1.2 SISTEMAS DE MANUFACTURA	13
1.2.1 Sistemas de Manufactura en Línea o Línea de Manufactura Dedicada – DML	14
1.2.2 Sistemas de Manufactura Flexible – FMS	14
1.2.3 Sistemas de Manufactura Reconfigurables – RMS	15
1.3 Sistemas De Manufactura Reconfigurables – RMS	16
1.3.1 Características	17
1.3.2 Principios	19
1.3.3 Herramientas de modelado para RMS:	20
1.3.4 Máquinas reconfigurables	21
1.3.5 RAMP – UP	26
1.3.6 Criterios de reconfigurabilidad para una RMT	27
Capítulo 2: Caracterización del estándar.....	30
2.1 Controladores de arquitectura abierta.....	31
2.1.1 Máquinas reconfigurables con Arquitectura Abierta	31
2.2 IEC 61499	32
2.2.1 Modelos y conceptos de IEC 61499	33
2.2.2 Modelo de control estandarizado	41
2.2.3 Adaptación del estándar.....	42
Capítulo 3: Definición del caso de estudio.....	44

3.1 Definición del proceso de mecanizado	44
3.1.1 Torneado.....	47
3.1.2 Taladrado.....	48
3.1.3 Fresado.....	49
3.2 Definición de la familia de productos	51
3.2.1 Taburete de cuatro patas	52
3.2.2 Silla de madera para adultos:.....	54
Capítulo 4: Desarrollo de la arquitectura de control para el caso de estudio	57
4.1 Desarrollo de la arquitectura mediante IEC 61499.....	57
4.1.1 Modelos del sistema	58
4.1.2 modelo de dispositivo.....	58
4.1.3 configuración de la comunicación	59
4.1.4 Modelo de aplicación	61
4.1.5 Modelo de recurso	66
4.1.6 Modelo de bloque de función	72
4.2 Simulación del sistema	86
4.2.1 Personalización.....	87
4.2.2 Convertibilidad	89
4.2.3 Modularidad	91
4.2.4 Integridad	93
4.2.5 Diagnosticabilidad	95
4.2.6 Escalabilidad	97
4.3 Resultados	98
4.3.1 Adaptabilidad de la capacidad productiva de la máquina	98
4.3.2 Desarrollo de un sistema para una familia de productos	98
4.3.3 Sensibilidad a cambios en su estructura.....	98
4.3.4 Detección de fallas en el proceso de reconfiguración.....	99
Capítulo 5: Conclusiones y trabajos futuros	100
5.1 CONCLUSIONES	100
5.2 TRABAJOS FUTUROS	101
BIBLIOGRAFÍA	104

Lista de figuras

	Pág.
Figura 1.1: Definición de manufactura de forma técnica y económica.....	13
Figura 1.2: ciencia base de los RMS.....	20
Figura 1.3: Estructura de un prototipo de controlador RMT.....	23
Figura 1.4: Máquina arquetipo RMT - diseño conceptual de la Universidad de Michigan.....	25
Figura 1.5: Proceso de diseño de una máquina reconfigurable.....	25
Figura 2.1: Modelo de sistema propuesto por el estándar IEC 61499.....	34
Figura 2.2: Modelo de dispositivos propuesto por el estándar IEC 61499.....	35
Figura 2.3: Modelo de recurso propuesto por el estándar IEC 61499.....	36
Figura 2.4: Modelo de aplicación propuesto por el estándar IEC 61499.....	37
Figura 2.5: Características de un bloque de función.....	39
Figura 2.6: Gráfica de control de ejecución ECC.....	39
Figura 2.7: Aplicación de reconfiguración dinámica para la adaptación de aplicaciones de control.....	41
Figura 2.8: Estructura jerárquica en capas.....	42
Figura 3.1: Esquema básico de RMT propuesta en paralelo con una máquina CNC tradicional.....	45
Figura 3.2: Procesos a realizar en el sistema de manufactura caso de estudio.....	46
Figura 3.3: Red de Petri de la forma de operación de la máquina reconfigurable caso de estudio.....	46
Figura 3.4: Proceso de torneado.....	47
Figura 3.5: Proceso de taladrado.....	48
Figura 3.6: Proceso de fresado frontal.....	50
Figura 3.7: Taburete unipersonal para adulto.....	53
Figura 3.8: Red de Petri del proceso de mecanizado de las piezas que componen el producto taburete de cuatro patas.....	54
Figura 3.9: Silla clásica de madera para adulto.....	55
Figura 3.10: Red de Petri del proceso de mecanizado de las piezas que componen el producto silla para adulto.....	56
Figura 4.1: modelo de sistema para el caso de estudio.....	58
Figura 4.2: modelo de dispositivo para el HMI.....	59
Figura 4.3: modelo de dispositivo para la RMT.....	59
Figura 4.4: Bloques de interfaz de servicio SUBSCRIBE y PUBLISH.....	60
Figura 4.5: Aplicación app_datosPc.....	62
Figura 4.6: Aplicación app_datosMk.....	63
Figura 4.7: Aplicación app_control_punta.....	64
Figura 4.8: Aplicación app_Reconfiguracion.....	65
Figura 4.9: subaplicación_control.....	66
Figura 4.10: recurso PC.HMI.....	67

Figura 4.11: recurso PC.COM_MK.....	68
Figura 4.12: Recurso MAQUINA.COM_PC.....	69
Figura 4.13: Recurso MAQUINA.TALADRADO.....	70
Figura 4.14: Recurso MAQUINA.FRESADO.....	71
Figura 4.15: Recurso MAQUINA.TORNEADO.....	72
Figura 4.16: FB VELM1.....	73
Figura 4.17: FB VELM2.....	74
Figura 4.18: FB vel_taladro.....	75
Figura 4.19: FB decision.....	76
Figura 4.20: FB Tipo_Operacion.....	77
Figura 4.21: FB tipo_silla.....	78
Figura 4.22: FB silla1.....	79
Figura 4.23: FB out_silla.....	83
Figura 4.24: FB control_pid.....	84
Figura 4.25: FB derivar.....	84
Figura 4.26: FB integrar.....	85
Figura 4.27: FB calculo_pid.....	86
Figura 4.28: Recurso PC.HMI y su representación en redes de Petri.....	88
Figura 4.29: Aplicación de reconfiguración y su representación en redes de Petri....	89
Figura 4.30: Red de FBs del bloque compuesto Tipo_Operacion y su representación en redes de Petri.....	90
Figura 4.31: Red de FBs de la subaplicación <i>Subapp_control</i> y su representación en redes de Petri.....	92
Figura 4.32: Red de FBs del bloque compuesto <i>VelM1</i> y su representación en redes de Petri.....	93
Figura 4.33: Recurso <i>MAQUINA.TORNEADO</i> y su representación en redes de Petri.....	94
Figura 4.34: Red de FBs del bloque compuesto <i>Control_pid_vel1</i> y su representación en redes de Petri.....	96

Lista de tablas

	Pág.
<i>Tabla 1.1: comparación entre los DML, FMS y RMS.....</i>	16
Tabla 1.2: Tipo de máquinas y sistemas de manufactura.....	17
Tabla 2.1: Capacidades de un sistema de control para máquinas reconfigurables ...	32
Tabla 2.2: Comparación entre el estándar IEC 61499 y IEC 61131	33
Tabla 3.1: operaciones relacionadas con torneado	48
Tabla 3.2: operaciones relacionadas con el taladrado.....	49
Tabla 3.3: operaciones de fresado frontal	50
Tabla 4.1: Distribución de direcciones IP.	60

Abreviaturas

DML	Sistema de manufactura en Línea
FMS	Sistema de Manufactura Flexible
CNC	Control Numérico por Computadora
RMS	Sistema de Manufactura Reconfigurable
RMT	Máquina Herramienta Reconfigurable
FB	Bloque de Función
HMI	Interfaz Hombre Máquina
ECC	Gráfico de Control de Ejecución

Capítulo 1: ¿Qué son los Sistemas de Manufactura Reconfigurable – RMS?

1.1 INTRODUCCIÓN

Los últimos tiempos traen consigo múltiples desafíos que deben enfrentar las empresas de manufactura en el contexto del mercado globalizado, desafíos tales como: precios competitivos, reducción de tiempos de producción, cantidades bajas y fluctuantes de producto, personalización de estos, calidad y durabilidad, entre otros factores ya conocidos en el mercado [1], por lo que se hace necesario aplicar sistemas de producción emergentes, pero con buen potencial para mejorar prácticas y aumentar la competitividad de la industria; sin embargo, lo más importante es la generación de un marco o teoría común para esos avances, debido a que a lo largo del siglo XX los fabricantes han estado expuestos a tendencias que son generadas en gran parte como desarrollos propietarios por parte de las compañías líderes.

La velocidad con que se mueve el mercado, requiere que las empresas manufactureras reaccionen cada vez más rápido y de forma rentable para posicionarse y ganar clientes; la capacidad de respuesta requiere que el sistema pueda cambiar su volumen de producción a medida que el mercado crezca y se adapte a las nuevas solicitudes de producto, estos ajustes exigen rapidez pero deben ser configurados de tal forma que puedan minimizar los problemas de calidad, ya que esto repercutirá en clientes insatisfechos, es por esto que la industria requiere un nuevo enfoque de fabricación [2].

Una respuesta efectiva a los cambios del mercado con un costo satisfactorio es actualmente el mayor desafío que enfrenta la industria, debido al avance de la

competencia que viene de todas partes del mundo, la volatilidad de los mercados y los factores que a estos afectan. Para encarar los problemas que se han evidenciado, existen tres tipos de sistemas de manufactura: la manufactura en línea y la manufactura flexible, los cuales son los más utilizados actualmente [3] y la manufactura reconfigurable, la cual es un paradigma industrial reciente, y a su vez también uno de los más prometedores. Cada uno de ellos presenta varias características útiles al momento de mejorar la productividad de la empresa, con el fin de alcanzar esos propósitos se ha desarrollado el enfoque modular de manufactura, el cual es más eficiente y reactivo frente a esos desafíos, debido a que dota al sistema productivo con la capacidad de reorganizarse de diferentes formas según convenga la situación, lo que le otorga la cualidad de flexibilidad al sistema e integra los últimos avances en el campo de la instrumentación industrial.

1.2 SISTEMAS DE MANUFACTURA

La manufactura es definida en [1] como un proceso físico y/o químico que se aplica a un material para agregarle valor al mismo, y generar un producto que pueda ser deseable al mercado; estos cambios pueden ir desde una reacción para crear un material totalmente nuevo, un simple cambio de apariencia o ensamblaje de estos; para realizar estas operaciones se hace necesario el uso de los recursos con que dispone la compañía como el personal, la maquinaria, los materiales y la energía, la figura 1.1 ilustra la definición de manufactura desde dos enfoques diferentes, la técnica, que muestra los recursos físicos que se necesitan para obtener los productos, y la económica, donde analiza el dinero que se debe invertir y las ganancias obtenidas tras el proceso de manufactura.



Figura 1.1: Definición de manufactura de forma técnica y económica, tomada y adaptada de [1].

Las empresas buscan posicionarse en el mercado, por eso han venido desarrollando metodologías que le permitan diferenciarse de la competencia y atraer clientes; con la revolución industrial, la aparición de nuevas máquinas y la automatización de procesos se han desarrollado varios sistemas, estos son:

1.2.1 Sistemas de Manufactura en Línea o Línea de Manufactura Dedicada – DML

En 1913 Henry Ford marcó el inicio de la producción en masa con la invención de la línea de producción para automóviles, por lo cual ese tipo de sistemas se desarrollaron a lo largo de todo el resto del siglo en los países industrializados, con el fin de generar bienes de consumo en grandes cantidades, lo que estableció grandes márgenes de rentabilidad para las industrias hasta mediados de 1990 [2]. Por medio de este sistema se fabrica un solo producto en grandes volúmenes, lo que crea un costo por unidad producida bajo, debido a las reducciones en los costos de aseguramiento de la calidad por cuenta de la consistencia del proceso, y la casi inexistente necesidad de adquirir equipos nuevos por largos periodos de tiempo [4]. La manufactura dedicada, compuesta por soluciones de automatización fija y económica hacen que se goce de una capacidad reducida para desarrollar nuevos productos, personalizarlos o manejar volúmenes variables de producción, lo que genera en algunas ocasiones respuestas deficientes frente a los cambios en el mercado y los patrones de consumo de los compradores [5].

1.2.2 Sistemas de Manufactura Flexible – FMS

Los Sistemas de Manufactura Flexible, nacen de la necesidad de diversificar la producción; como una mejora de las Líneas Manufactura Dedicadas se desarrolló esta nueva propuesta de automatización, la cual se ha caracterizado por producir una variedad de piezas que pueden ser modificadas pero no cuentan con la capacidad de los DML¹ de producir grandes volúmenes [6]. Esta forma de producción requiere de altos costos de inversión para su realización, ya que está compuesta por arreglos de máquinas interconectadas por un medio de un transporte automatizado de material, las máquinas preferidas para estos sistemas son las máquinas de control numérico por computadora CNC y de automatización programable

¹ A partir de ahora cuando se hable de DML se refiere a Sistemas de Manufactura en Línea

organizadas en estaciones de trabajo [9], y un control centralizado que hace las veces de supervisor y controla el medio de transporte automático, con base en la lectura de las salidas de las estaciones de trabajo [14]; sin embargo, este tipo de manufactura ha sido poco adoptado por la industria, ya que es costoso y tiene bajos niveles de productividad como se aprecia en la tabla 1.1 [3].

1.2.3 Sistemas de Manufactura Reconfigurables – RMS

Los Sistemas de Manufactura Reconfigurable son el nuevo reto de las empresas como respuesta al rápido avance de la competencia y las exigencias del mercado, los RMS² son un tipo de sistema que se ajusta en cuanto a su diseño y operación [3] para desarrollar nuevas oportunidades de producción en la empresa; estos sistemas cuentan con la capacidad y funcionalidad de los DML y los FMS³, respectivamente, solo que aquí estas características son dinámicas y se ajustan a lo que se necesita en el momento que se requiere [15]. Además, la reconfigurabilidad presente en estos ofrece beneficios económicos en cuanto a la optimización y reducción de tiempos en los procesos de producción requeridos al fabricar una familia de productos. La tabla 1.1 muestra una comparación entre los tres tipos de sistemas y sus características.

Características	DML	FMS	RMS
Productividad	Alta	Baja	Alta
Costo de tiempo de vida	Bajo, para una sola pieza cuando se usa plenamente	Razonable, para la producción de varias piezas simultáneamente	Media, para una producción alta y mediana, con demanda variable y nuevas piezas durante todo su ciclo de vida
Tipo de mercado	Estable	Predecible	Incierto
El nivel de diferencia entre el sistema de manufactura y la variación de la demanda	Muy alta	Media	Baja

Tabla 1.1: comparación entre los DML, FMS y RMS [2] [10] [11].

² A partir de ahora cuando se hable de RMS se refiere a Sistemas de Manufactura Reconfigurables

³ A partir de ahora cuando se hable de FMS se refiere a Sistemas de Manufactura en Línea

<i>Continuación de la tabla 1.1</i>			
Características	DML	FMS	RMS
Estructura del sistema	Fija	Cambiable	Cambiable
Estructura de la máquina	Fija	Fija	Cambiable
Proceso tecnológico	Fijo	Necesita ser adaptado	Sensible a cambios
Flexibilidad	No	General	Personalizada (alrededor de una familia de productos)
Escalabilidad	No	Si	Si
Tiempo de reconfiguración para una nueva familia de productos	18-30 meses	1-3 días	2-4 semanas
Operación de herramientas simultáneamente	Si	No	Posible
Tiempo de ajuste para una nueva familia de productos	2-4 meses	1-2 días	1-2 semanas

Tabla 1.1: comparación entre los DML, FMS y RMS [2] [10] [11].

La implementación de los RMS, genera un volumen de producción alto y variedad de productos. Además, busca asegurar la calidad de los procesos y productos de tal forma que satisfagan tanto al cliente como a la economía de la empresa; la familia de productos ayuda a que la empresa pueda posicionar su marca mejor en el mercado, ya que se pueden producir más piezas que se adapten y acoplen a los rangos de trabajo de las máquinas con las que cuenta el proceso de manufactura, la reconfiguración hace que el sistema sea más rápido y eficiente para responder exitosamente a la competencia, lo que será indispensable en los años venideros [16].

1.3 Sistemas De Manufactura Reconfigurables – RMS

La competitividad de una empresa no solo depende de la alta calidad a bajo costo, sino que requiere de una respuesta rápida a los cambios en el mercado y las nuevas tendencias de los clientes. Los Sistemas de Manufactura Reconfigurable pretenden

enfrentar esta nueva dinámica, optimizando la capacidad y la funcionalidad, debido a que sus componentes se ajustan rápidamente a las circunstancias (nueva demanda de productos, integración de nuevas tecnologías, cambios en los productos, etcétera) [6]. Lo que se busca es obtener una serie de alternativas para modificar el sistema, de tal forma que se obtenga una respuesta oportuna y rentable, ya sea eliminando, agregando o reordenando elementos del sistema de manufactura para producir una familia de productos [7]; todo esto de forma rápida y sin demandar grandes inversiones de dinero.

Tanto el software como el hardware son claves en la reconfiguración⁴ del sistema, pero también deben ser considerados las futuras actualizaciones y la vida útil de sus componentes, de esta forma se asegura que se puedan hacer cambios en los productos o introducción de nuevos en el momento que se requiera, además, la estructura y la organización de toda la línea de producción debe diseñarse de forma integrada. Estos atributos se resumen en la tabla 1.2 para dar un diseño funcional al sistema:

	Máquinas fijas	Máquinas reconfigurables
Sin software	Máquinas manuales, líneas de manufactura dedicadas DML.	No es posible.
Software de control fijo	Sistemas de Manufactura Flexible FMS, máquinas CNC, robots, pórticos.	Pórticos reconfigurables, máquinas reconfigurables.
Software reconfigurable	Guía autónoma de vehículos AGV, arquitectura abierta de control.	RMS

Tabla 1.2: Tipo de máquinas y sistemas de manufactura, tomada y adaptada de [13].

1.3.1 Características

Los RMS cuentan con algunas características funcionales que le confieren sus cualidades especiales de flexibilidad y productividad, como:

⁴ Según el Oxford English Dictionary, reconfiguración es el arreglo de partes o elementos organizados de diferentes formas, tipos o combinaciones.

- *Personalización:* esta es una de las características más importantes, ya que diferencia al sistema de los demás y lo hace más productivo debido a que le permite ofrecer productos a la medida de cada cliente. Los RMS pueden producir una familia de productos, es decir varias piezas que cuentan con características y geometría parecidas, trabajan en los mismos rangos y tienen procesos similares [13].
- *Convertibilidad:* hace referencia a la capacidad del sistema para adaptarse rápidamente a nuevos y diferentes productos, esto puede implicar cambios de accesorios o integración de nuevas máquinas en el sistema [13].
- *Diagnosticabilidad:* la capacidad de identificar rápidamente los problemas o fallas que ocurran en el sistema o en la calidad de la familia de productos, esto requiere que el sistema se sintonice rápidamente cuando es reconfigurado para que siempre se cumpla con los estándares de calidad de la empresa [13].
- *Escalabilidad:* es la respuesta rápida y eficiente del sistema para cambiar el volumen de producción y adaptarse a la demanda del mercado [6].
- *Modularidad:* la capacidad de los componentes del sistema (software y hardware) para funcionar como unidades independientes o de trabajar en equipos de diferentes formas [6]. Esta propiedad aumenta las alternativas de trabajo del sistema de manufactura de forma óptima e integral. Además, permite la reutilización de componentes e introducción de nuevas tecnologías que hacen que el sistema tenga más variedad y personalización de los productos [2].
- *Integrabilidad:* capacidad del sistema para recibir nuevos paquetes tecnológicos, reordenar las máquinas y sus accesorios para formar un sistema reconfigurable [6].

Un sistema que cumpla con estas seis características, asegura un mejor rendimiento ya que los costos se reducen, el tiempo se minimiza y la calidad se asegura, por otra parte esto mejora la competitividad para responder eficientemente a la demanda, de esta forma el sistema adquiere capacidades como el cambio en los productos (materiales, diseño) y cambio de alternativas de operación, y no solo se habla de software y hardware reconfigurable, la creatividad y la innovación del talento humano presente en el proceso de manufactura, también es importante para llevar a cabo un proceso de reconfiguración exitoso [10].

1.3.2 Principios

Un RMS cumple con los siguientes principios definidos en [2]:

A. Un RMS proporciona recursos de producción ajustables para responder a los cambios del mercado impredecibles y eventos del sistema intrínsecos:

- La capacidad de un RMS puede ser rápidamente escalable en pequeños incrementos.
- La funcionalidad de un RMS puede adaptarse rápidamente a los nuevos productos.
- Las capacidades de ajuste incorporados de un RMS facilitan una respuesta rápida a fallas inesperadas en los equipos.

B. Un sistema RMS está diseñado en torno a una familia de productos, es decir, un conjunto de productos con características similares como tamaño, forma, utilidad, etcétera. Los cuales están denominados bajo una misma marca, que para este caso serían fabricados por el mismo sistema (para entender cómo se diseña una familia de productos se puede dirigir al anexo A). El diseño del RMS debe contar con la suficiente flexibilidad para producir todos los miembros de esa familia.

C. Las características básicas de un RMS deben ser incorporadas en el sistema como un todo, así como sus componentes (mecánicos, comunicaciones y control).

Son varias las investigaciones que se han empezado a desarrollar para obtener sistemas reconfigurables, pero aún no hay una metodología formalizada que permita desarrollar este tipo de manufactura, se debe aprovechar la teoría hasta ahora planteada y buscar herramientas formales que permitan la creación de nuevas soluciones, contribuyendo al avance de la reconfiguración para su aplicación. En la figura 1.2 se puede observar los diferentes campos de aplicación de estos sistemas, que se consolidan en lo que puede ser considerado como la ciencia de los RMS y las características esenciales para su desarrollo.

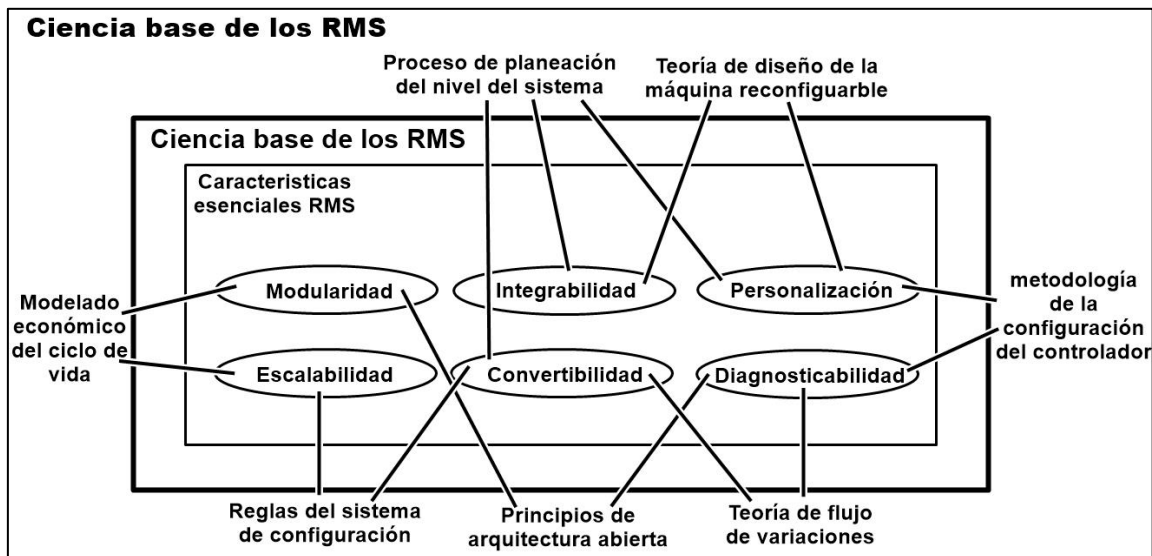


Figura 1.2: ciencia base de los RMS, tomada y adaptada de [12].

1.3.3 Herramientas de modelado para RMS:

- **Redes de Petri**

Las redes de Petri son una técnica utilizada para modelar el comportamiento de un sistema dinámico, aunque sin brindar mucha información sobre la estructura de este; estas ofrecen varias ventajas frente a otras técnicas debido a la amplia utilización en el modelado de sistemas, una sólida base formal matemática y su amplia acogida y utilización dentro de la industria. Además, se han desarrollado varios avances alrededor de estas redes, como técnicas de diseño para mejorar los sistemas, detección de fallas, etcétera. Adicional a esto, estas redes son la técnica predilecta para representar el funcionamiento de los sistemas distribuidos [21].

- **Arquitectura IEC 61499**

El estándar IEC 61499 trajo grandes avances al campo del control distribuido y ha dado luz a la estandarización de la semántica y funcionalidad de estos sistemas. El estándar ha sido aceptado de buena forma por la industria, sin embargo aún presenta algunas ambigüedades como se muestra en [22], lo que ha retardado la completa integración de sus principios y una mayor apropiación de los conceptos de reconfigurabilidad.

La arquitectura propuesta en IEC 61499-1 propone una estructura de control descentralizada con el fin de obtener un sistema más flexible. Además, el uso de

capas que define el estándar aplicando sus diferentes modelos, subaplicaciones y bloques compuestos, promueve la modularidad del sistema, esto refleja una reducción del nivel de complejidad del sistema, el cual se descompone mediante el uso de esta herramienta; todo esto bajo el concepto de Bloque Funcional FB, que es un pequeño sistema de control distribuido con sus entradas y salidas independientes que se interconectan a un flujo de eventos [24].

Con base en las ventajas de descentralización y modularidad que ofrece el estándar, y debido al respaldo que brinda por tratarse de un documento formal y debidamente documentado por una entidad de prestigio internacional, se opta por escogerlo como modelo de arquitectura para desarrollar este proyecto y darle forma al componente lógico reconfigurable de la máquina, que será responsable de ejecutar el caso de estudio. En el siguiente capítulo se estudiará más a fondo las ventajas del uso de esta herramienta.

1.3.4 Máquinas reconfigurables

Gran parte de los trabajos realizados anteriormente, se han enfocado en la reconfiguración de máquinas, realizando procesos de mecanizado que se requieren en gran parte de los procesos de manufactura, con el fin de obtener una respuesta rápida para crear nuevos productos y para cualquier proceso en general; por eso, este trabajo está enfocado en el desarrollo de una arquitectura de control para este tipo de procesos adquiriendo las características de un RMS, para esto se ha desarrollado en [5] el concepto de Máquina Herramienta Reconfigurable RMT⁵, donde la máquina maneja un propósito general y tienen un rango de trabajo para una familia de productos que cumplen con sus especificaciones.

La familia de productos para una RMT debe cumplir con ciertas características similares como su geometría, los materiales utilizados, la actividad que requieren para ser transformados, entre otras. Un punto clave para obtener una máquina reconfigurable es la modularidad que es una de las características de los RMS, esta puede ser adquirida mediante el uso del estándar IEC 61499 con el uso de los Bloques de Función. Además, el desarrollo de ese tipo de herramientas requiere un amplio conocimiento de su funcionamiento mediante el modelado del sistema, pero

⁵ A partir de ahora cuando se habla de RMT se refiere a Máquina Herramienta Reconfigurable.

aún no existe una técnica específica para obtener el modelo [10] y por tal razón se busca el uso de herramientas formales como el estándar IEC 61499.

Los objetivos básicos de las RMT propuestos en [5] son:

- *“Aumentar la tasa de producción de la máquina mediante la adición de dispositivos de mecanizado (exactamente la capacidad necesaria).”*
- *“Adaptar la funcionalidad de la máquina al cambiar su geometría para adaptarse a la producción de un nuevo miembro de una familia de piezas (exactamente la funcionalidad necesaria).”*

Se tiene la capacidad y la funcionalidad, pero la fabricación de una máquina reconfigurable requiere de la adición de componentes y cambios que permitan una buena integración y la personalización de los mismos, lo cual requiere inversiones que pueden llegar a ser costosas. Además, un punto importante del diseño es el componente lógico del funcionamiento del sistema al que se va a enfocar este trabajo, ya que se requiere un comportamiento eficiente para evitar daños en la producción cuando la máquina es reconfigurada y una respuesta oportuna, de esto depende la calidad del producto y la fidelidad de los clientes. El centro de investigación de ingeniería de la Universidad de Michigan, ha desarrollado dos máquinas RMT, una máquina de inspección reconfigurable y otra máquina de montaje reconfigurable para el sector automotriz.

El diseño de este tipo de máquinas pretende obtener beneficios económicos al momento de ser adquirida, en lugar de comprar una máquina con muchas funciones, como los centros de mecanizado que actualmente se encuentran en el mercado, se compra una máquina más simple con la funcionalidad que realmente se ajuste al proceso de manufactura requerido [5]. Las RMT son un híbrido entre las máquinas robustas y con altos volúmenes de producción de los DML y el uso de la tecnología CNC de los FMS [25]. La RMT brinda una solución a los cambios de los productos como son la geometría de la pieza que se está procesando, la capacidad de producción, los procesos requeridos y las propiedades del material [26]. La tabla 1.3 muestra estas diferencias.

	DML	RMT	Máquinas CNC
Estructura de la máquina	Fija	Ajustable	Fija
Enfoque de diseño	Pieza	Familia de Piezas	Máquina
Escalabilidad	No	Si	Si
Flexibilidad	No	Personalizada	General
Herramientas operando simultáneamente	Si	Si	No

Tabla 1.3: RMT, combinación de las características de máquinas de un DML y las CNC, tomada y adaptada de [26].

El control de la máquina debe contar con una arquitectura abierta que permita integrar nuevas tecnologías y ser modular para que sea fácil de personalizar, características que pueden ser adquiridas mediante el uso de una arquitectura de control distribuida. Un software de control se ha desarrollado para las RMT's, llamado configurador de control, este cuenta con un HMI y su herramienta de simulación, su estructura se muestra en la figura 1.3 (la selección del software para el desarrollo de este trabajo se encuentra el en anexo D).

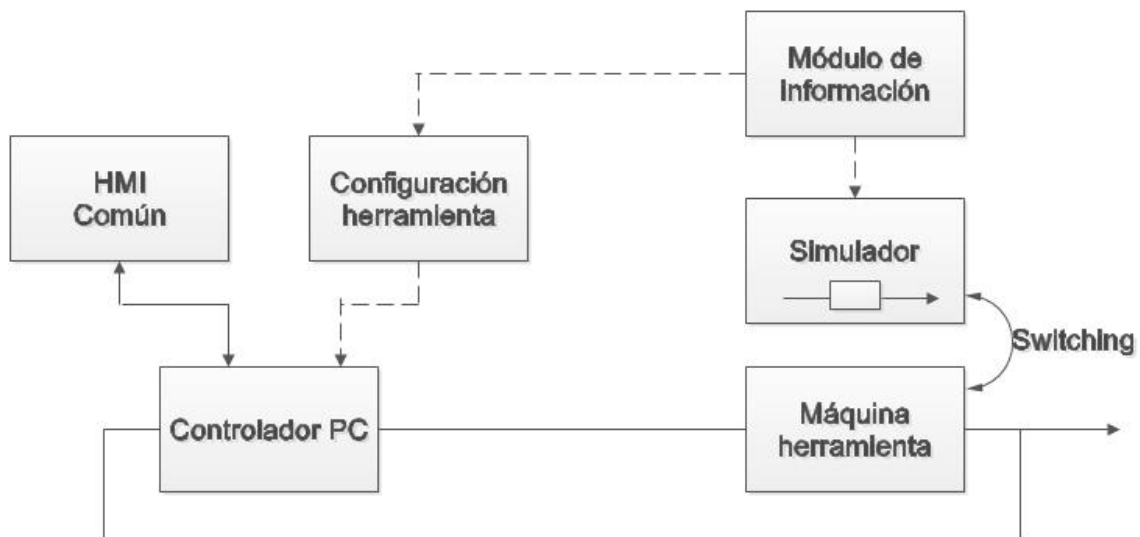


Figura 1.3: Estructura de un prototipo de controlador RMT.

En cuanto al diseño mecánico de la máquina, esta debe alcanzar todos los movimientos (grados de libertad) necesarios para realizar un conjunto de características del producto y ser hábil para cambiar los requisitos de operación, lo cual implica que posea una mecánica modular. PREMADE es una herramienta de diseño CAD/CAM que permite el desarrollo de algoritmos para estas máquinas; otro punto que requiere atención es la rigidez de la máquina en todas sus

configuraciones, que la reorganización de sus módulos sea rentable y la precisión geométrica que debe tener para asegurar la calidad de las piezas procesadas [25].

- **Tipos de RMT's**

En [26] se han definido tres tipos de máquinas herramientas reconfigurables RMT:

- **Máquinas herramienta modulares**

Una herramienta reconfigurable no necesariamente es modular, esta propiedad exige que las partes que componen la máquina tengan un alto nivel de detalle para que haya un buen acoplamiento de las mismas, como ejemplo las conexiones y la fuente de alimentación que pueden convertirse en una gran limitante en el funcionamiento de la máquina y su geometría, además, la calibración no se logra de forma rápida. Por todo esto, se deben establecer interfaces de conexión y asegurar que los módulos sean autónomos e independientemente funcionales para garantizar la eficiencia de la máquina, características que hasta el momento no se han logrado implementar en este tipo de máquinas.

- **Multi-herramienta**

Son máquinas que pueden realizar varias tareas y de forma simultánea, como agujeros sobre una pieza y cambiar los accesorios para obtener otro resultado, o bien estos pueden ser reconfigurados; quitar, agregar o reubicar los accesorios son tareas que se pueden realizar en una máquina multi-herramienta para obtener un uso eficiente de la máquina. Este tipo de máquinas tienen una respuesta más rápida que las de CNC propias de un sistema flexible, pero son más lentas en comparación con una máquina de un sistema en línea, ya que estas se dedican a una tarea repetitiva, mientras que las máquinas reconfigurables son flexibles para una familia de piezas.

- **Arquetipo RMT**

Este tipo de máquinas poseen un sistema no ortogonal (excepto cuando está en posición horizontal) que realiza tareas de taladrado y fresado sobre superficies inclinadas, con el objetivo de disminuir los DOF en comparación con una CNC que quisiera realizar los mismos movimientos. Un ejemplo de esto es el prototipo conceptual desarrollado en el Centro de Investigación de Ingeniería de la Universidad

de Michigan, que se muestra en la figura 1.4 donde la máquina puede moverse con un ángulo de inclinación de 15° a 60°.

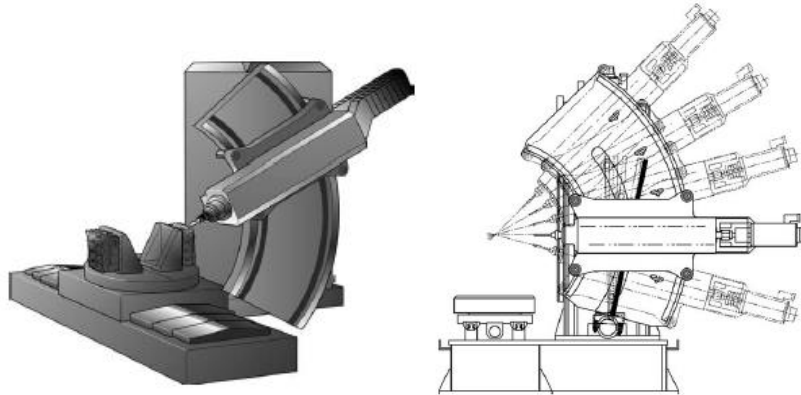


Figura 1.4: Máquina arquetipo RMT - diseño conceptual de la Universidad de Michigan [26].

Los pasos básicos de ingeniería para el diseño de una máquina reconfigurable se resumen en la figura 1.5.

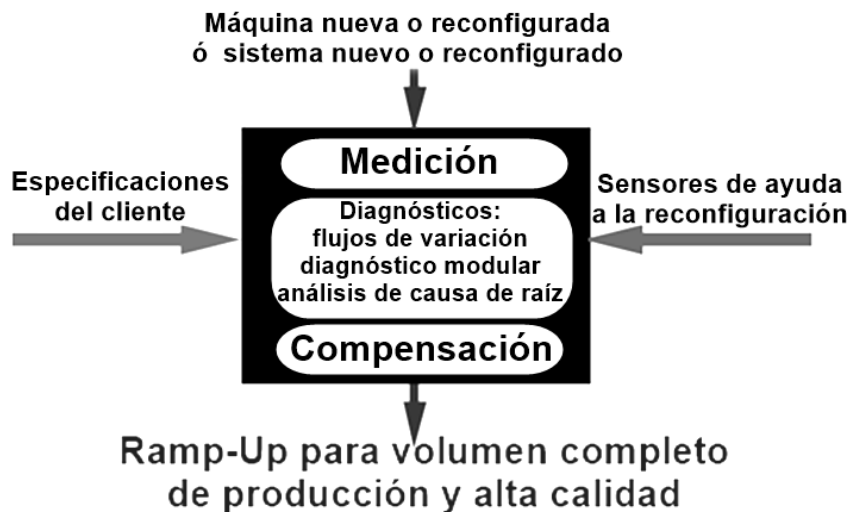


Figura 1.5: Proceso de diseño de una máquina reconfigurable [27].

En la medición se tiene que definir el tipo de sensor o sensores para determinar las medidas de la pieza, supervisar el movimiento de la máquina, velocidad del motor y demás datos relevantes para el control de la máquina; con todos estos datos se realiza el diagnóstico, para evitar fallas en el sistema y establecer rutas de comunicación que permita al operario actuar de forma oportuna, este diagnóstico debe tener un buen nivel de detalle para poder determinar si son fallas de la

máquina, por la calidad de la pieza o mala instalación de los módulos. Con el diagnóstico se realiza la compensación que son las acciones correctivas, ya sea de forma automática o manual.

1.3.5 RAMP – UP

Reducir el tiempo de aceleración es un factor crítico que debe tenerse en cuenta al momento de hacer la reconfiguración del sistema. Si el sistema no responde rápidamente a los cambios no se obtendrá la eficiencia y los resultados de aplicar un RMS no serán exitosos. Este concepto es definido en [13] como:

“Periodo Ramp-up: el periodo de tiempo que le toma a un sistema recientemente introducido o reconfigurado para alcanzar sus niveles de producción diseñados, sostenibles y de largo plazo en términos de rendimiento y calidad de la pieza”.

En [3] se define como el tiempo que tarda un sistema en alcanzar una producción sostenible con niveles de calidad estable, es decir, es el tiempo que toma la rampa de subida en la gráfica de producción y calidad para estabilizarse cuando se tiene un producto nuevo o reconfigurado. La reducción del tiempo de ramp-up se vuelve crítico para mantener la competitividad de los RMS debido a las múltiples configuraciones y calibraciones a las que se debe someter el sistema, lo que genera múltiples periodos de ramp-up durante el funcionamiento del mismo. Para conseguir tiempos de ramp-up cortos es necesaria la medición de errores durante el funcionamiento de la planta para corregir las fallas que están afectando la capacidad productiva o la calidad de los productos fabricados, para esto es indispensable el uso de sensores y de herramientas matemáticas que permitan identificar rápidamente la fuente del error y solucionarla lo antes posible [13]. Los principios de trabajo de los sistemas de calidad tipo Seis Sigma⁶ o Calidad Total, son muy utilizados para la identificación de errores [3]. Por otro lado, los RMS cuentan con la característica de diagnosticabilidad y su dinámica hace que el sistema responda más ágilmente para prevenir y/o corregir fallas.

⁶seis sigma es una teoría sobre calidad para detección de errores.

1.3.6 Criterios de reconfigurabilidad para una RMT

Ya se han definido todas las características y atributos que se deben alcanzar en el desarrollo de un proceso de manufactura reconfigurable. Con base en la teoría descrita anteriormente, se han establecido cuatro criterios para la evaluación de los resultados que se obtengan en el proyecto. Estos criterios se establecieron con base en los objetivos que se deben alcanzar para el desarrollo de una máquina reconfigurable que será el caso de estudio de este trabajo.

- **Adaptabilidad de la capacidad productiva de la máquina**

Objetivo relacionado: Aumentar la tasa de producción de la máquina mediante la adición de dispositivos de mecanizado (exactamente la capacidad necesaria).

Características relacionadas: convertibilidad, integrabilidad, escalabilidad y personalización.

Este criterio se ha definido con base en el primer objetivo de una RMT, con el fin de adaptar la producción no solo en cantidad sino en variedad que es el propósito de los RMS al definirse como la unión de la capacidad de los DML y la funcionalidad de los FMS; además, este criterio está relacionado con las características: convertibilidad permitiendo que el sistema añada nuevos productos, personalización realizando cambios sobre los productos de forma práctica, escalabilidad para manejar diferentes volúmenes de producción e integrabilidad para ajustar los componentes existentes en el sistema dotándolo con la capacidad de manejar los cambios que se requieran sobre la máquina que se desea reconfigurar.

- **Desarrollo de un sistema para una familia de productos**

Objetivo relacionado: Adaptar la funcionalidad de la máquina al cambiar su geometría para adaptarse a la producción de un nuevo miembro de una familia de piezas (exactamente la funcionalidad necesaria).

Características relacionadas: personalización y modularidad.

Este criterio se ha definido para cumplir de buena manera el segundo objetivo de una RMT, ya que uno de sus rasgos principales de los RMS es ofrecer una gama amplia pero limitada de productos para mejorar el rendimiento de la producción y diferenciarse de un FMS; esto le permite una personalización más ágil ya que se tienen establecidas unas características que se deben cumplir para ser parte de la familia de productos. Además, la modularidad facilita creación de la familia de

productos, ya que se puede descomponer la fabricación de estos en diferentes funciones estandarizadas para su reutilización en nuevos productos.

- **Sensibilidad a cambios en su estructura**

Objetivo relacionado: Adaptar la funcionalidad de la máquina al cambiar su geometría para adaptarse a la producción de un nuevo miembro de una familia de piezas (exactamente la funcionalidad necesaria).

Características relacionadas: Escalabilidad, integrabilidad y personalización.

El tercer criterio dispone un buen cumplimiento del segundo objetivo de una RMT, en otras palabras, dota al sistema con la capacidad de manejar los cambios en su estructura, para aumentar la funcionalidad de la misma. Este criterio aportará a la máquina reconfigurable la posibilidad de variar sus volúmenes de producción relacionado con la escalabilidad del sistema y reordenar su lógica de funcionamiento aportando al proceso de reconfiguración definida en la integrabilidad de los RMS. Al mismo tiempo, los cambios son relacionados con la característica de personalización para lograr que la máquina cambie su lógica de funcionamiento para generar nuevos productos.

- **Detección de fallas en el proceso de reconfiguración**

Objetivo relacionado: Aumentar la tasa de producción de la máquina mediante la adición de dispositivos de mecanizado (exactamente la capacidad necesaria).

Características relacionadas: diagnosticabilidad y modularidad.

Como ya se ha definido anteriormente, los RMS son sistemas que quieren fidelizar a sus clientes ofreciendo buenos productos, la manera adecuada para llegar a esto es desarrollando procesos confiables; se debe buscar que el proceso de manufactura genere la cantidad y variedad evitando daños y pérdidas de material, esto se apoya en la característica de diagnosticabilidad para que el sistema reconozca fallas en los procesos que se llevan a cabo en la máquina y estas sean controladas. Por otro lado, la modularidad se relaciona en este criterio ya que facilita el diagnostico de fallas ya que es más fácil de entender el sistema al ser descompuesto en varias partes que reducen su complejidad. De esta forma se apoya el cumplimiento del primer objetivo de las RMT mejorando la productividad de la máquina.

Capítulo 2: Caracterización del estándar

Existen varias herramientas software para controlar un sistema dinámico, estas requieren de cierta autonomía para coordinar y controlar sus componentes y ser actualizable para suplir los cambios en la producción. El diseño de este tipo de sistemas requiere una arquitectura que mezcle la jerarquía del control tradicional y la autonomía del nuevo para aportar la flexibilidad propia de los RMS [36], en este sentido se debe tener especial cuidado debido a que ya existe un proceso que no es reconfigurable y se van a dar los criterios para que lo sea, la existencia de un software y hardware establecidos que deben entrar a cumplir con las propiedades de los RMS, implica la necesidad de un buen modelo de los componentes que se tienen lo que puede llegar a ser una tarea compleja y aquí entra a jugar un papel importante la modularidad para facilitar la interpretación e integrar mejor la nueva propuesta de forma escalable.

Como forma de solucionar los problemas que traen los RMS a nivel de control, en [26] se hace un resumen sobre varios tipo de arquitecturas de control abiertas que se desarrollan en el mundo y que permitirían facilitar el manejo de estos sistemas, debido a su facilidad para de integrar nuevos módulos de control a través de librerías o técnicas de programación orientada a objetos, lo que permite la expansión y reutilización del software, aumentando así la reconfigurabilidad del mismo. Algunos sistemas desarrollados con estas características son el alemán HÜMNOS, el americano OMAC-TEAM y el japonés OSEC.

2.1 Controladores de arquitectura abierta

La IEEE ha definido un sistema abierto como:

“Un sistema abierto proporciona capacidades que permitan la correcta utilización de aplicaciones que se ejecutan en una amplia variedad de plataformas de múltiples proveedores, inter-operar con otras aplicaciones del sistema, y la presentación de un estilo coherente de interacción con el usuario” [26].

Las características propias de los RMS son requeridas para este tipo de control, que además lo hacen compatible con aplicaciones distribuidas que hacen de un sistema descentralizado una propuesta de control más eficiente, ya que su estructura es más fácil de comprender y configurar, su comunicación horizontal genera cierta autonomía que repercute en respuestas más rápidas y reducción de fallas en un proceso. Al mismo tiempo, el desarrollo de tecnologías de este tipo permite ampliar, cambiar y reutilizar componentes del sistema de control [26], por esto es importante considerar las características de los dispositivos que implementan esta arquitectura, en la siguiente sección se da un breve panorama sobre estos.

2.1.1 Máquinas reconfigurables con Arquitectura Abierta

El diseño modular de la máquina repercute directamente en el desarrollo de los controladores abiertos y para esto se debe tener en cuenta la flexibilidad del equipo físico que siempre va a ser limitada, por lo tanto puede requerir modificaciones manuales al momento de realizar una reconfiguración. Por otro lado, el diseño del controlador va a ser más amplio en este sentido ya que permite manejar una mejor dinámica, apoyado en herramientas estándar que permitan desarrollar un sistema modular y fácilmente adaptable al hardware a controlar [3], las capacidades del sistema de control para este tipo de máquinas deben tener características especiales que le permitan alcanzar estas funcionalidades, en la tabla 2.1 se resumen estas características.

Capacidad	Significado	Opciones de hardware	Opciones de software
Modular	Soporta estructuras descentralizadas	Sistemas estandarizados y distribuidos comercialmente	Módulos de aplicación distribuidos
Interoperable	Los componentes pueden cooperar		Intercambio de datos estandarizado
Portable	Integración de componentes en diferentes entornos		APIs ⁷ estandarizadas
Escalable	Desempeño incrementado, la topología puede ser modificada	Mejora del procesador	Tareas de software paralelas
Extensible	La funcionalidad puede ser mejorada	Cartas plug-in adicionales	Módulos de aplicación adicionales

Tabla 2.1: Capacidades de un sistema de control para máquinas reconfigurables, tomada y adaptada de [3]

2.2 IEC 61499

Actualmente el estándar IEC 61499 desarrollado por la Comisión Electrónica Internacional goza de buena aceptación dentro de la industria, este presenta una arquitectura descentralizada para el desarrollo de sistemas de control distribuidos, que como ya se ha mencionado anteriormente permiten el desarrollo de un proceso de control reconfigurable y se ha detectado en trabajos pasados como en [30] que cumple las características de los RMS. El estándar propone la forma de usar el modelo de bloques de función (definidos por IEC 61131-3 entre sus lenguajes de programación de PLCs) dentro de un sistema de control distribuido [32]. Sin embargo, han habido avances en el desarrollo de otros tipos de estructuras de control diferentes a los sistemas distribuidos, con base en el estándar; como se muestra en [33], estructuras de tipo jerárquico presentan ciertas ventajas en el campo de la reconfigurabilidad del sistema, para alcanzar esas características, es necesario hacer ciertos cambios a la disposición clásica propuesta, maximizando el uso de ciertas herramientas propuestas en el mismo estándar como será explicado más adelante.

⁷ Application Programming Interface permite el intercambio de datos entre aplicaciones [31]

Gracias al estándar IEC 61499 se aumenta la flexibilidad y eficiencia de los procesos ya que se cuenta con una comunicación tanto horizontal como vertical; además, la modularidad facilita el diagnóstico del proceso. La comparación entre este y su antecesor se pueden observar en la tabla 2.2, donde se pueden evidenciar las ventajas que se adquieren.

Características	Estándar IEC	
	IEC 61499	IEC 61131
Ejecución del programa	Dirigida por eventos	Cíclico
Arquitectura de programación	Sistemas completos	Controladores individuales
Tiempo real	Habilitado	Habilitado
Compatibilidad entre programas	Puede ser procesada sin importar/exportar	Está importado/exportado
Estructura del control lógico	Puede ser dividido entre múltiples controladores	Es centralizado
Inteligibilidad entre diferentes marcas	Diferentes computadores soportan librerías (CATs)	Usualmente las librerías no son soportadas

Tabla 2.2: Comparación entre el estándar IEC 61499 y IEC 61131, tomada y adaptada de [32].

Los bloques de función son herramientas útiles ya que permiten la modularidad y por lo tanto reutilización del software, dando solución a pequeños problemas de control. Estos encapsulan algoritmos de control que dependen de una lista de variables de entrada y escriben en sus salidas los resultados de esos algoritmos, facilitando el entendimiento del sistema por parte de personas que no sean especialistas en el tema, soluciones completas son desarrolladas a través de la conexión de varios bloques de función. Además, el método propuesto por el estándar es independiente de la implementación por lo que puede ser aplicado a varios sectores.

2.2.1 Modelos y conceptos de IEC 61499

Los modelos propuestos por el estándar se caracterizan por su alto grado de modularidad y estructura en capas, con esto se logra facilitar el entendimiento de sistemas complejos, además de brindar características de usabilidad y facilitar el trabajo con los sistemas de control distribuido como:

- **Modelo de sistema**

Este modelo se muestra en la figura 2.1 y evidencia la relación entre los dispositivos que componen el sistema y las formas de comunicación que existen entre ellos como las redes industriales, una aplicación puede estar contenida dentro de un solo dispositivo o repartida entre varios, este tipo de aplicaciones distribuidas tienen la función de redes de comunicación entre diferentes bloques de función, que son cargadas como fragmentos de aplicación en los diferentes dispositivos, siendo ese tipo de aplicaciones lo que garantiza el flujo de eventos y datos dentro del sistema. Este modelo resulta muy práctico a la hora de modelar un sistema reconfigurable, ya que permite verlo como la unión de varios dispositivos conectados a través de redes de comunicación, lo que aumenta la modularidad del mismo y facilita su entendimiento y así se acerca a las características definidas en la sección 1.3.1.

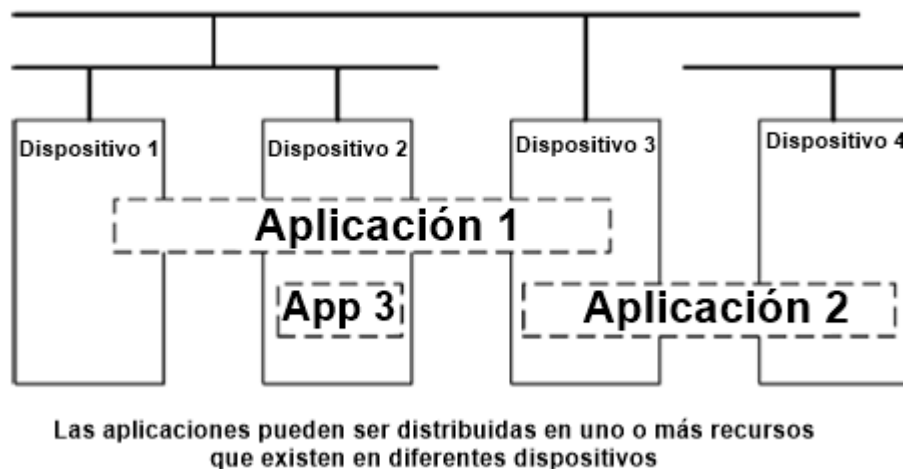


Figura 2.1: Modelo de sistema propuesto por el estándar IEC 61499, tomada y adaptada de [8].

- **Modelo de dispositivo**

Este modelo trae de nuevo el concepto de Recurso, el cual es similar al recurso propuesto por el estándar IEC 61131-3, lo que le provee independencia de ejecución y control de redes de los bloques de función.

El modelo de dispositivo cuenta con una interfaz de proceso que posibilita el intercambio de información con las entradas y salidas del dispositivo físico y de la misma forma hay interfaces de comunicación, los cuales le dan soporte a través de redes externas con dispositivos remotos como se ve en la Figura 2.2. En este modelo

se muestra la flexibilidad que propone el estándar, y que es vital para los sistemas de manufactura reconfigurable en la medida en que cada dispositivo contiene varias aplicaciones, que se ejecutan cuando son necesarias a diferencia de IEC 61131 donde la ejecución es cíclica.

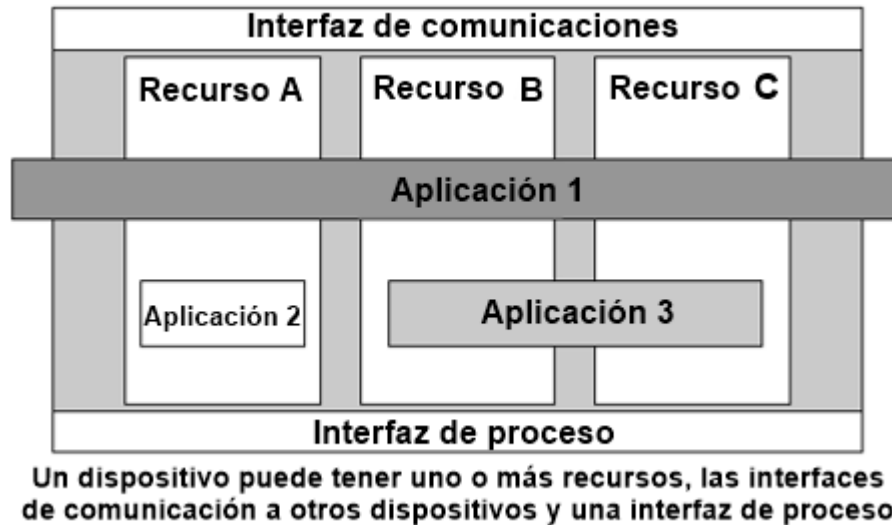


Figura 2.2: Modelo de dispositivos propuesto por el estándar IEC 61499, tomada y adaptada de [8].

- **Modelo de recurso**

Los recursos dan soporte a la ejecución de uno o más fragmentos de aplicación de bloques de función, los bloques de función están asignados para los recursos dentro de los dispositivos interconectados. El recurso provee interfaces con el sistema de comunicaciones y hacia el proceso específico del dispositivo, es decir a servicios y subsistemas externos que están estrechamente relacionados con el dispositivo. En la figura 2.3 se ilustra cómo están conectados los diferentes bloques de función a través de datos y eventos. La función de Scheduling⁸ que se ve en la parte inferior, es la encargada de garantizar la correcta ejecución de los algoritmos de los bloques, el bloque de función de Interfaz de Servicio hace las veces de comunicador entre los bloques y las interfaces de recurso. Una de las características más importantes de los recursos, es su independencia, es decir que un recurso puede ser manipulado sin afectar a los demás recursos de la red o el dispositivo; sin embargo, hace falta cierto nivel de coordinación entre los diferentes recursos o fragmentos mientras se manipula alguno de ellos.

⁸ Scheduling: planeación, programación, ordenamiento de tareas y prioridades.

El modelo de recurso da soporte a características como la integrabilidad e interoperabilidad del sistema, debido a que permite la ejecución de aplicaciones distribuidas a través de fragmentos de estas, coordinados gracias a bloques de servicio que le comunican con otros dispositivos remotos, lo que permite la adición de nuevas funcionalidades en forma de aplicaciones distribuidas.

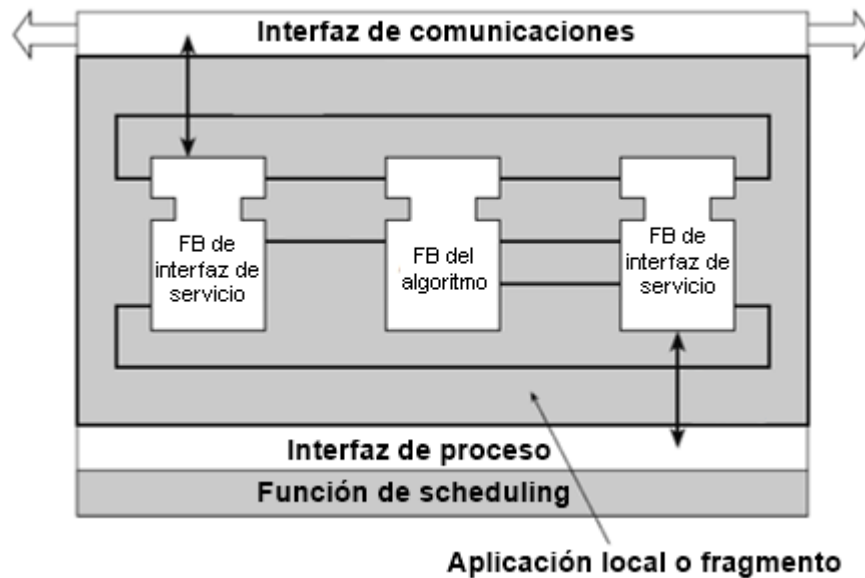


Figura 2.3: Modelo de recurso propuesto por el estándar IEC 61499, tomada y adaptada de [8].

- **Modelo de aplicación**

Según IEC 61499 una aplicación es una red de bloques de función interconectados a través de flujos de datos y eventos. Una aplicación puede estar fragmentada y distribuida entre múltiples recursos como se muestra en la figura 2.4, las aplicaciones también pueden estar descompuestas en subaplicaciones, las cuales tienen la apariencia externa de un bloque de función, sin embargo por dentro está compuesta de una red de bloques de función que pueden a su vez estar distribuidos entre varios recursos. El estándar propone una estructura auto similar para las funciones, lo que permite incluso que una subaplicación pueda ser dividida en más subaplicaciones y así en adelante como sea necesario. A nivel de aplicación se definen los flujos de datos y eventos necesarios para el funcionamiento del sistema, además se garantiza que el flujo de eventos sea el correcto para la ejecución de los diferentes algoritmos dentro de los bloques de función.

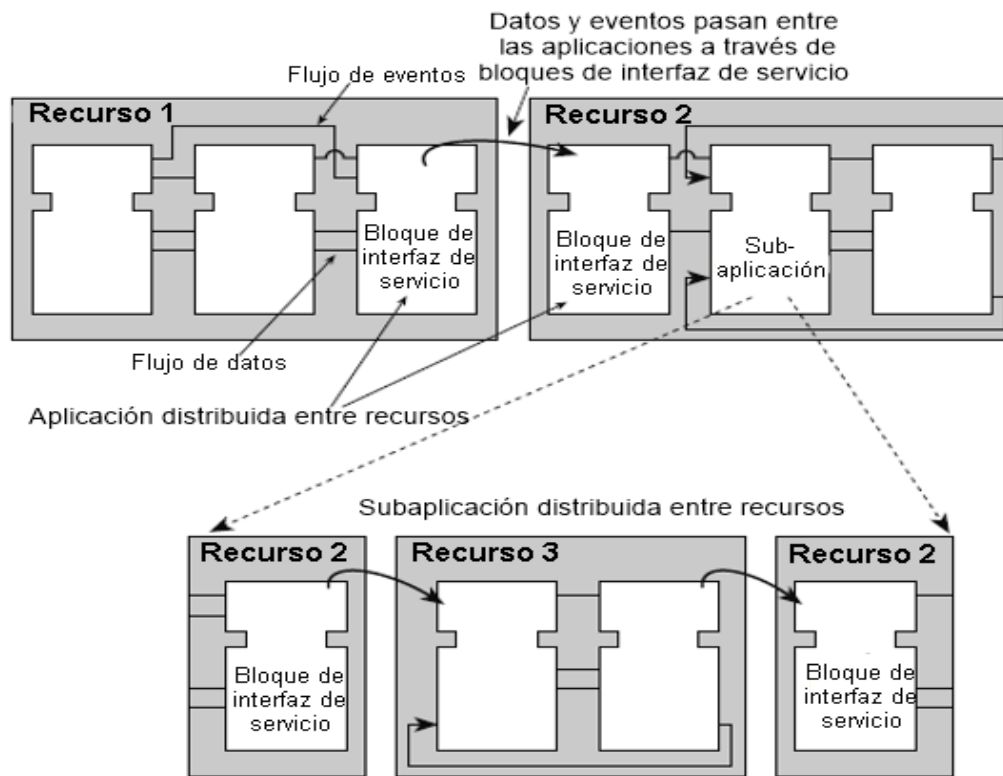


Figura 2.4: Modelo de aplicación propuesto por el estándar IEC 61499, tomada y adaptada de [8].

Los bloques de función utilizados para construir las redes que componen a las aplicaciones, son en algunos casos instancias de bloques de función genéricos, aunque de igual forma el estándar permite que algunos bloques tengan valores iniciales preconfigurados.

- **Modelo de bloque de función FB⁹**

El modelo de FB es el corazón del estándar y la unidad básica que sostiene toda la arquitectura que este propone. Un FB es una unidad funcional de software cuya estructura de datos puede ser manipulada por uno o varios algoritmos. Este concepto no es novedoso en la industria, sin embargo el estándar IEC 61499 incluye algunas funcionalidades nuevas tomando como base el FB definido en IEC 61131, entre ellas están:

- Cada bloque tiene un conjunto de entradas de eventos que reciben eventos desde otros bloques.

⁹ A partir de ahora cuando se hable de FB se refiere a Bloque de Función.

- Existe una o más salidas de eventos, las cuales son usadas para propagarlos hacia otros bloques.
- Existe un conjunto de entradas de datos que son recibidos desde otros bloques.
- Existe un conjunto de salidas de datos, las cuales son usadas para transmitir los resultados de la ejecución de los algoritmos del bloque hacia otros bloques.
- Cada bloque tiene un conjunto de variables internas, las cuales son utilizadas durante la ejecución de los algoritmos del mismo.
- El comportamiento del bloque está definido en función de algoritmos e información de sus estados. Usando la información de esos estados se pueden modelar varias estrategias para definir el orden de ejecución de los algoritmos del bloque frente a diferentes eventos de entrada.

También IEC 61499 describe varios tipos de FBs, los cuales varían en la estructura de datos que manejan y en los algoritmos que pueden ser configurados dentro de cada uno, lo que hace que cada tipo de bloque deba tener un nombre de tipo y un nombre de instancia, los cuales deben ser siempre visibles en cualquier representación gráfica del mismo.

En la figura 2.5 se resaltan las principales características de los FBs, en la parte superior del bloque está ubicado el Gráfico de Control de Ejecución, el cual funciona como una máquina de estados y su tarea es mapear los eventos recibidos por el bloque para determinar el orden de ejecución de los algoritmos. La parte baja del bloque contiene los algoritmos y los datos internos, los cuales no son visibles al usuario.

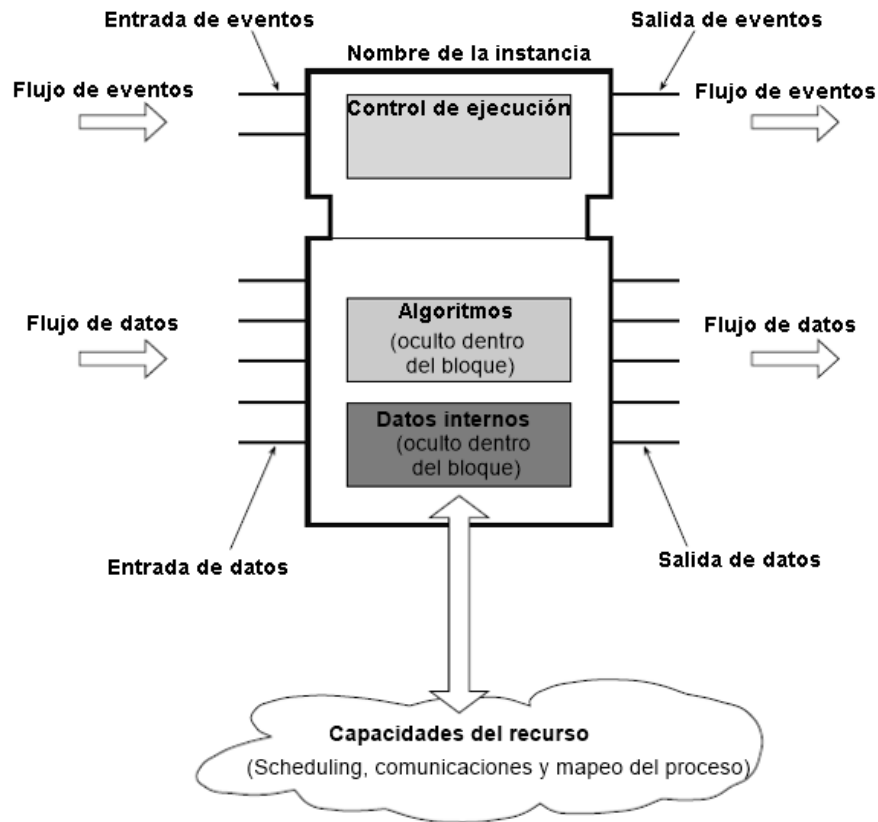


Figura 2.5: Características de un bloque de función, tomada y adaptada de [8].

Una de las características más importantes de un FB es su ECC¹⁰, este gráfico sirve para entender el funcionamiento del FB desde un punto de vista de comportamiento en estados, cada estado está asociado a una acción, ya sea la ejecución un algoritmo o el detonamiento de un evento; para cambiar de estado es necesaria la ocurrencia de un evento; una de las maneras más prácticas para entenderlo (figura 2.6) es relacionándola con el funcionamiento de una red de Petri tradicional.

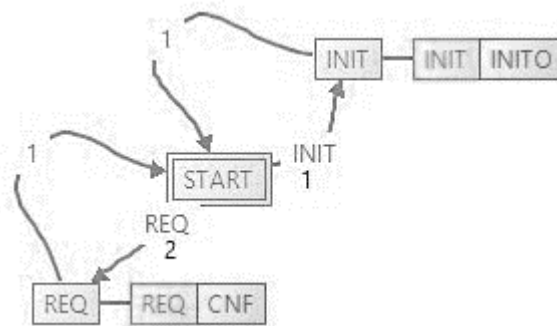


Figura 2.6: Gráfica de control de ejecución ECC.

¹⁰ Gráfico de Control de Ejecución.

- **Tipos de bloques de función:**

Una de las características más importantes del estándar es la capacidad de definir diferentes tipos de FBs, lo cual permite caracterizar el comportamiento del bloque, las interfaces de comunicación que utiliza y el tipo de algoritmos que ejecuta, los principales tipos de bloques definidos por el estándar son:

- **Bloque de función básico:** este es el FB general que se utiliza para la construcción de sistemas, su comportamiento está definido por los algoritmos que son en él programados y que se ejecutan de acuerdo con los eventos de entrada que recibe lo que genera un cambio de estados dentro del bloque, los cuales son posteriormente mapeados en los algoritmos usando un mecanismo conocido como Execution Control Chart o ECC [8].

- **Bloque de función compuesto:** en este tipo de bloque su comportamiento interno está definido por una red de FBs instanciados, la cual está interconectada por datos y eventos de igual forma que las redes de niveles superiores.

- **Bloque de función de interfaz de servicio:** este bloque está dedicado a los intercambios de información por lo que brinda soporte de comunicaciones entre una red de FBs con servicios externos, como serían otros FBs ubicados en dispositivos remotos o comunicación con algún componente hardware del sistema.

- **Subaplicaciones:**

Las subaplicaciones pueden ser vistas como FBs compuestos que pueden ser distribuidas en más de un recurso, que a su vez es lo que las diferencia de los bloques compuestos, generalmente sus componentes y conexiones pueden reutilizarse en distintas conexiones de red. Esta definición que provee el estándar es de gran utilidad para obtener un sistema de control distribuido; para lograr la conexión entre los recursos de la subaplicación se pueden utilizar FBs de interfaz de servicio Publish y Subscribe, cuyo funcionamiento se describirá más adelante, conectados mediante un red de comunicaciones (Ethernet o bus de campo) que permite el intercambio de datos; pero también existen diferentes FBs como Solicitud/Petición y de Respuesta, entre otros definidos por el estándar dependiendo de la comunicación que se requiera. Además, los bloques encargados de la

comunicación van a requerir parámetros adicionales¹¹ al del control para el envío de datos, por ejemplo: direcciones IP [8].

2.2.2 Modelo de control estandarizado

La manufactura moderna requiere nuevos cambios y adaptación de sus funciones para dar una respuesta oportuna en el mercado. Este trabajo propone una estrategia ágil y rápida, la reconfiguración a nivel de máquina mediante la arquitectura de control propuesta por el estándar IEC 61499 que le permita al sistema cumplir con la funcionalidad y la capacidad requerida para una familia de productos, los cuales son definidos en el siguiente capítulo. Son muchos los sectores económicos que requieren mejoras tecnológicas e innovación para suplir la demanda y por esto la reconfiguración dinámica es una buena metodología para dar respuesta a estas solicitudes, evitando paradas innecesarias, fallas o pérdidas del capital de una empresa. Además, la automatización basada en eventos de los FB's mejora la sincronización para la reconfiguración en aplicaciones distribuidas [34]. De forma general, se presenta la figura 2.7 donde se muestra el uso de una aplicación de reconfiguración acoplada con una aplicación de control distribuida entre dispositivos. El procedimiento de reconfiguración se muestra como una solicitud de reconfiguración o RCA, para cada proceso que se encarga de supervisar y modificar la ejecución de la aplicación de control [35].

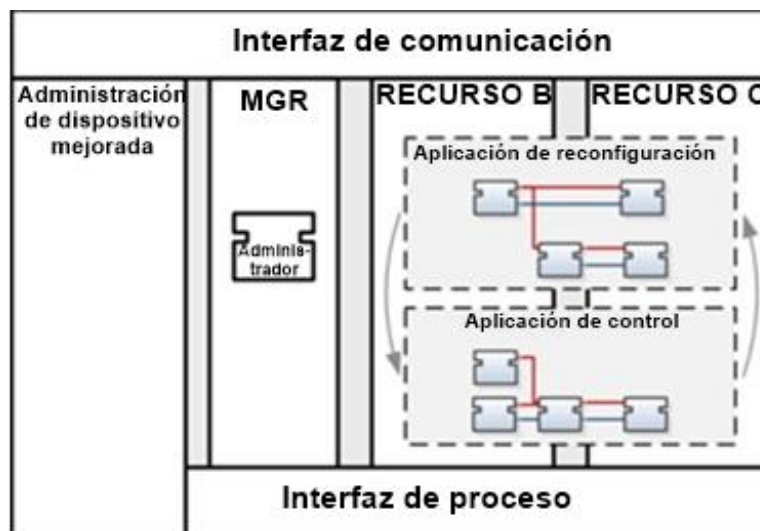


Figura 2.7: Aplicación de reconfiguración dinámica para la adaptación de aplicaciones de control, tomada y adaptada de [35].

¹¹ Los protocolos no son definidos por la norma, pero sí el servicio de comunicación mediante FBs.

2.2.3 Adaptación del estándar

Con el fin de facilitar los procesos de reconfiguración, algunos investigadores europeos han propuesto una arquitectura jerárquica en capas [33], en la cual los bloques de las capas inferiores tienen la función de estar en contacto directo con el *hardware* del sistema efectuando acciones de control más sencillas e inmediatas, mientras que las capas superiores tienen funciones más complejas de supervisión, coordinación y control (figura 2.8).

Con el fin de facilitar el funcionamiento de ese tipo de estructuras basadas en un estándar evidentemente enfocado al control descentralizado, trabajos como el de Alois Zoitl¹² maximiza el uso de subaplicaciones y en menor medida bloques compuestos, con el fin de simplificar la visualización de sistemas tan complejos, además explota el uso de adaptadores con el propósito de mejorar las comunicaciones necesarias.

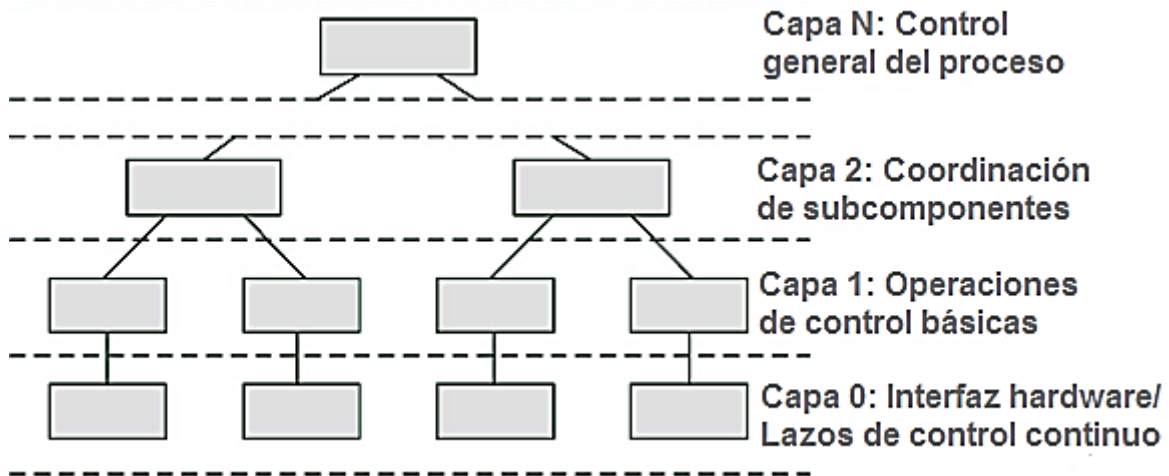


Figura 2.8: Estructura jerárquica en capas.

Una de las tareas más complejas que propone estas adaptaciones, es la comunicación vertical, en especial en las capas intermedias debido a la separación que se debe hacer entre la comunicación hacia arriba y hacia abajo, lo que divide la función de comunicación en dos subaplicaciones que de igual forma deben estar coordinadas, sin contar las tareas propias de control donde haya lugar; es por esto que se privilegia el uso de subaplicaciones en la implementación de este modelo

¹² Dr. Del Instituto Austriaco de Tecnología, y uno de los grandes impulsores de la reconfiguración a través de IEC 61499, referenciado en [33][34][35].

aumentando de esa forma los retrasos en la ejecución del modelo, lo que hace más necesario el correcto funcionamiento del ECC. Como se muestra en [33], una vez superados los desafíos anteriormente mencionados, es posible construir un sistema jerárquico funcional tomando los principios y herramientas del estándar IEC 61499.

Capítulo 3: Definición del caso de estudio

El desarrollo de esta propuesta está orientado hacia el control de un proceso de mecanizado de piezas que se lleva a cabo en varias empresas de manufactura, el cual se desarrollará sobre una RMT, que es uno de los mecanismos utilizados para la conversión de un sistema de producción tradicional en un RMS debido a que las capacidades de estos dispositivos los constituyen en un sistema por sí mismos. Por esto se van a plantear algunas sugerencias para modificar una máquina CNC propia de un sistema de manufactura flexible por la variabilidad de movimientos que esta puede llegar a realizar, lo cual puede aumentar la funcionalidad de la máquina que se desea modelar. El proceso de control se va a desarrollar alrededor de tres actividades primarias, las cuales van a tener ciertas variaciones para lograr procesar una familia de productos, todo esto con el fin de adquirir un sistema que cumpla con los principios de los RMS.

Este tipo de proceso se ha tomado con el fin de aumentar la productividad propia de un FMS y obtener una estructura convertible; además, se le agregan las características de un RMS, para lograr esto es importante el diseño del controlador definido en el capítulo 2 mediante el estándar IEC 61499 y así tener un Ramp Up corto (sección 1.3.7). La definición de los diferentes procedimientos que se pretenden realizar con la nueva máquina que se propone, se exponen de forma general en este capítulo resaltando las variables importantes que se deben tener en cuenta para el desarrollo del modelo de control.

3.1 Definición del proceso de mecanizado

El proceso de mecanizado consiste en tres actividades primarias que son taladrado, fresado y torneado, para esto se toma una máquina CNC que pueda realizar

movimientos en los ejes de coordenadas X, Y y Z. La máquina escogida para el montaje en un trabajo posterior va a requerir cambios en la punta¹³ la cual actuará directamente sobre la pieza para que pueda realizar las tareas planteadas, ya que cada una requiere de una punta especial y para esto, se sugiere un magazín¹⁴ que es utilizado en los centros de mecanizado adecuado para este tipo de cambios de forma automática, además se necesita que la máquina cuente con un motor extra en la base para que de esa forma se puedan girar las piezas y efectuar la tarea de torneado, en la figura 3.1 se muestra un esquema ilustrativo de la máquina sugerida.

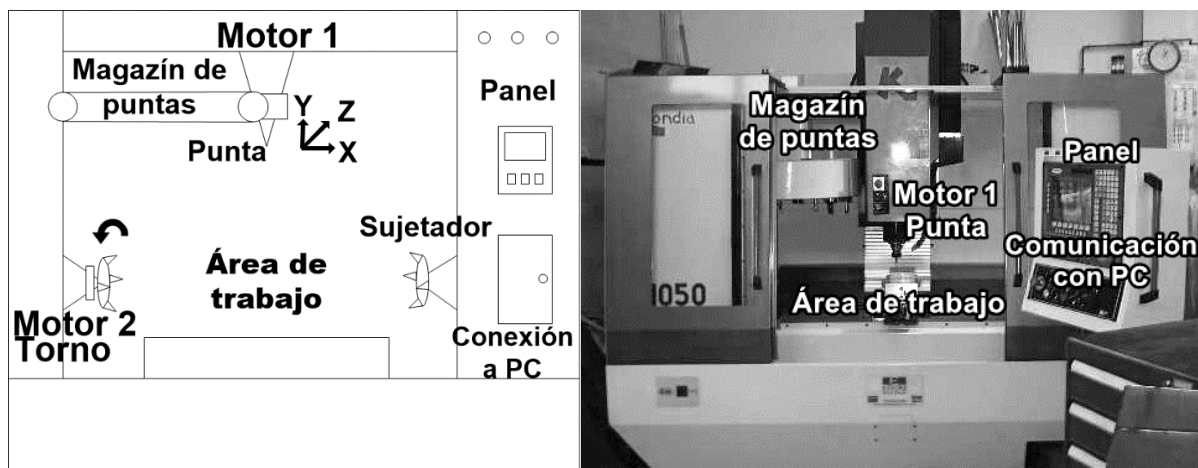


Figura 3.1: Esquema básico de RMT propuesta en paralelo con una máquina CNC tradicional, fotografía tomada y adaptada de [45]

El esquema ilustrado en la figura 3.1, a la izquierda se tiene la RMT que muestra el motor 1 que manipula la punta de trabajo que actúa sobre la pieza de trabajo, en la parte inferior está el motor 2 que gira para la operación de torneado, la cual se explicará a continuación, mientras sostiene el material a procesar con ayuda de los sujetadores y es esta función la mayor diferencia con el centro de mecanizado CNC, que se ve a la derecha de la imagen; el área de trabajo se define de acuerdo al tamaño de las piezas que se van a fabricar, que están restringidos por la familia de productos, este concepto se aclarará más adelante; el panel y la conexión a pc sirven para que el operario interactúe con la máquina, por ejemplo, para la entrega del modelo de la silla CAD.

¹³ Herramienta de trabajo que entra en contacto directo con la pieza a mecanizar, puede ser una broca.

¹⁴ Dispositivo que adosado a los centros de mecanizado ofrece una variedad de puntas o brocas de trabajo que son intercambiadas de manera automática, este dispositivo va girando y se detiene cuando ubica la punta requerida.

Con esto se busca una reconfiguración de cambios en el proceso de mecanizado mediante modificaciones en la máquina que le van a permitir aumentar su funcionalidad [26], así se cumple el primer objetivo de una RMT y se establece el primer criterio definido en el capítulo 1 para aumentar la producción y la variedad de productos, en el anexo B se puede encontrar un diagrama P&ID donde se ilustra la estructura básica de la máquina. Los procesos que se deben realizar para la ejecución de estas actividades son mostrados en la figura 3.2.

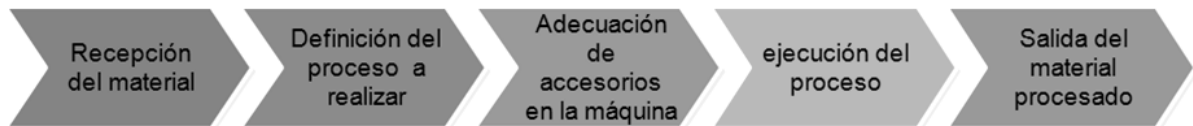


Figura 3.2: Procesos a realizar en el sistema de manufactura caso de estudio.

La operación de la máquina de forma general, sobre la cual se va a realizar el proceso se puede observar en la figura 3.3, de esta manera el proceso del caso de estudio es convertible adaptándose a nuevos productos rápidamente, característica propia de los RMS. A continuación, se explican las tareas de mecanizado que debe realizar el sistema.

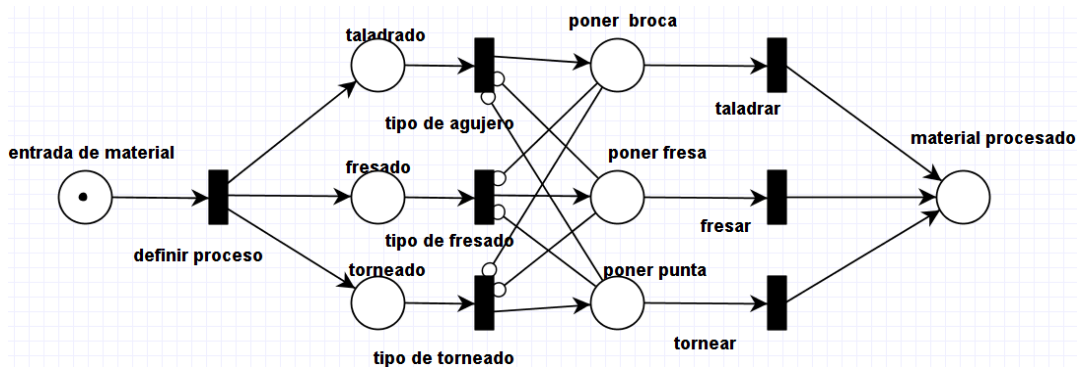


Figura 3.3: Red de Petri de la forma de operación de la máquina reconfigurable caso de estudio.

Como se puede observar en la figura anterior, el proceso está diseñado en torno a tres tareas, sin embargo, es fácilmente escalable para acoger nuevas tareas de mecanizado de ser necesario. Debido a que se plantea realizar una reconfiguración a nivel de proceso, la lógica de funcionamiento se diseñará en torno a una red de bloques capaz de tomar decisiones, como se mostrará en el capítulo siguiente.

3.1.1 Torneado

Este proceso consiste en moldear la pieza de trabajo mediante el movimiento la herramienta de corte mientras la pieza cilíndrica gira [37]. En este proceso se deben tener en cuenta las siguientes variables: distancia entre la pieza y el desplazamiento de la máquina, dimensiones de la pieza, diámetro del agujero del husillo que actúa sobre la pieza, velocidad, potencia de trabajo del motor. La ecuación 3.1 se utiliza para el cálculo de la velocidad de rotación y la figura 3.4 ilustra este proceso.

$$N = \frac{v}{\pi D_o} \quad (3.1)$$

Donde N: velocidad de rotación [rev/min]

v: velocidad de corte [m/min]

D_o : diámetro de la pieza [m]

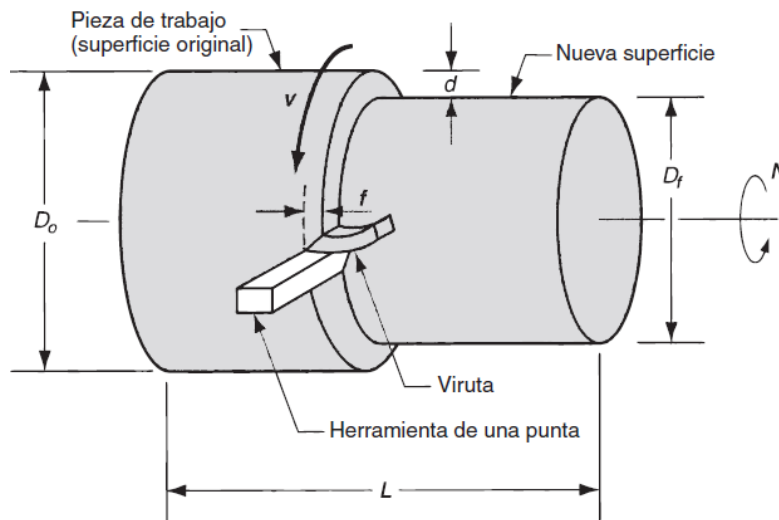


Figura 3.4: Proceso de torneado, tomada de [37].

Dado que el proyecto debe ser definido en torno a una familia de productos, se presentan las diferentes operaciones relacionadas con el torneado necesarias para el caso de estudio que se ilustran en la tabla 3.1. Otras operaciones y demás aspectos técnicos no son de interés en el estudio, ya que este proyecto no pretende implementar la máquina sino diseñar el sistema de control, sin embargo algunos de esos aspectos son documentados en el anexo B. Se pretende desarrollar dos actividades derivadas del torneado donde se obtienen diversas formas, todo esto

mediante el cambio de la punta que actúa sobre la pieza y la configuración del sistema de control para obtener el resultado deseado.

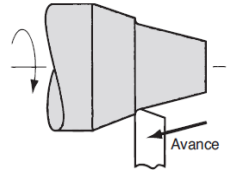
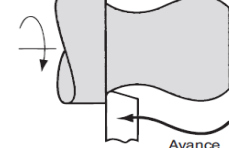
Operación	Descripción	vista
Torneado ahusado o cónico	La herramienta avanza en cierto ángulo creando una forma cónica.	
Torneado de contornos	La herramienta avanza siguiendo un contorno diferente a la línea recta, creando así una forma contorneada.	

Tabla 3.1: operaciones relacionadas con torneado [37].

3.1.2 Taladrado

Consiste en realizar agujeros redondos sobre una pieza mediante una broca que se introduce mientras gira y el diámetro queda definido por el tamaño de la herramienta [37]. Este proceso se puede observar de forma general en la figura 3.5.

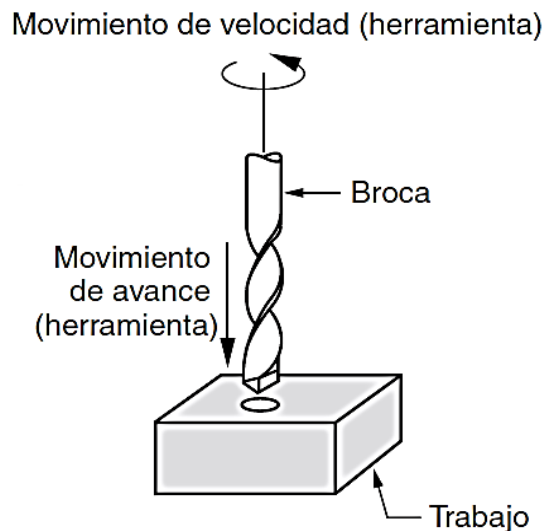


Figura 3.5: Proceso de taladrado, tomada de [38].

El cálculo de la velocidad de rotación para esta operación se realiza mediante la ecuación 3.2. Cumpliendo con los requerimientos de los RMS para una familia de productos, se plantean dos actividades afines al proceso de taladrado que serán

ejecutadas para la elaboración de los productos definidos más adelante, estas se muestran en la tabla 3.2.

$$N = \frac{v}{\pi D} \quad (3.2)$$

Donde N: velocidad de rotación [rev/min]

v: velocidad de corte [mm/min]

D_o: diámetro de la broca [mm]

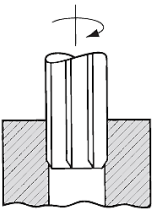
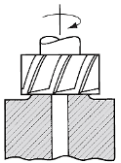
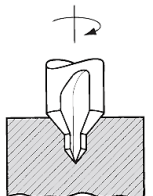
Operación	Descripción	Vista
Escariado	Un escariador aumenta el tamaño del agujero ligeramente, mejora la tolerancia en su diámetro y el acabado superficial.	
Refrentado	Genera una superficie maquinada ¹⁵ plana en la pieza de trabajo en el área donde se ubique.	
Centrado	Realiza el agujero inicial que mejora la precisión de la siguiente operación.	

Tabla 3.2: operaciones relacionadas con el taladrado [37].

3.1.3 Fresado

Es una operación de corte interrumpido debido a que la herramienta que actúa sobre la pieza de trabajo posee unos dientes que entran y salen en cada rotación. Existen dos tipos de fresado básico, que son el frontal y el periférico, debido a la estructura de la máquina planteada para el caso de estudio se va a presentar solo el primero que se muestra en la figura 3.6. La velocidad de rotación se calcula mediante la ecuación 3.3.

$$N = \frac{v}{\pi D} \quad (3.3)$$

¹⁵ Maquinado: remoción de exceso de material [38].

Donde N : velocidad de rotación [rev/min]
 v : velocidad de corte [mm/min]
 D_o : diámetro exterior de la fresa [mm]

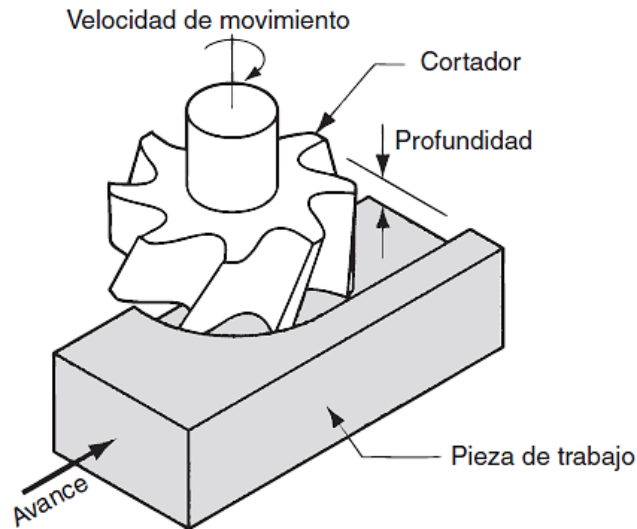


Figura 3.6: Proceso de fresado frontal, tomada de [37].

El fresado frontal se ejecuta perpendicular a la pieza para generar diferentes perfiles y piezas con curvaturas [39]. Para finalizar con la definición de las actividades del proceso caso de estudio que se han planteado, se tienen dos actividades de fresado que generan diversas formas y se presentan en la tabla 3.3 que serán ejecutadas para la fabricación de los productos.

Operación	Descripción	Vista
De perfiles	Corta una pieza desde la periferia.	
De contorno superficial	Genera una superficie tridimensional mientras avanza su punta redonda.	

Tabla 3.3: operaciones de fresado frontal [37].

El segundo criterio definido en el capítulo 1: aseguramiento de la calidad del proceso se da mediante las consideraciones para cada uno de los procesos definidos. La

diagnosticabilidad es una de las características más importantes para el seguimiento de este criterio, la cual se adquiere con el desarrollo del controlador aplicando la arquitectura del estándar IEC 61499. Además, la modularidad mediante los FB's permite una respuesta más rápida y eficiente reflejada en la productividad de la máquina. Con el planteamiento de las diferentes operaciones en cada uno de los tres procesos propuestos para el sistema de manufactura se da cumplimiento al segundo objetivo para una RMT: "Adaptar la funcionalidad de la máquina al cambiar su geometría para adaptarse a la producción de un nuevo miembro de una familia de piezas". De esta forma se da cumplimiento a los dos últimos criterios definidos en el primer capítulo para adquirir de las características de los RMS: escalabilidad, integrabilidad, personalización y modularidad. El desarrollo de todas estas operaciones en la máquina, genera un sistema capaz de producir una familia de productos gracias a la sensibilidad adquirida en su estructura que es fácil de modificar.

3.2 Definición de la familia de productos

Agrupando todas las tareas que se han definido en la anterior sección se define la familia de productos, la cual consiste en la fabricación de sillas que van a requerir de varios de esos procesos para su fabricación, teniendo en cuenta el diseño de los productos presentes en un RMS (ver anexo A), se toma como base para el diseño del controlador dos modelos de sillas que se desean elaborar en la máquina, mediante el proceso de reconfiguración. Con los dos modelos escogidos se podrá observar que la familia puede adquirir nuevos productos que cumplan con características parecidas y adaptar la funcionalidad de la máquina.

Las sillas serán elaboradas de madera por lo que se restringe la capacidad de la máquina para este material debido a las calibraciones de potencia de trabajo y dureza de las herramientas necesarias para trabajarlo, estas limitaciones diferencian a este sistema de un FMS el cual podría trabajar con un rango de productos más amplio, sin embargo aún presenta una flexibilidad suficiente para poder producir nuevos miembros de la familia de productos que cuenten con las características similares de material, procesos de transformación y dimensiones a las de los productos inicialmente propuestos. Al ser capaz de producir las piezas de madera necesarias para la elaboración de los productos en una sola máquina se diferencia

de un tradicional sistema DML, el cual requeriría una línea de producción exclusiva para cada tipo de estas piezas. Además, el ramp up diferenciará al sistema de los otros, este tiempo juega un papel importante en el control del proceso para adquirir una respuesta oportuna.

La familia de productos llamada “sillas de madera” cuenta con 2 miembros iniciales los cuales son:

3.2.1 Taburete de cuatro patas

El taburete para adultos, debido a su forma y características es el producto con la mayor uniformidad entre las piezas que lo componen. El producto terminado se muestra en la figura 3.7.

Las piezas que componen el taburete se han organizado en tres categorías, con el fin de simplificar el modelado de los procesos de mecanizado necesarios:

- Patas: 4 piezas de mayor longitud las cuales requieren procesos de torneado por cuestiones de diseño y taladrado para hacer posible su ensamblaje.
- Travesaños: 3 piezas de menor longitud que necesitan ser taladradas y torneadas para ser usadas como piezas de soporte para el taburete lo que le da estabilidad al mismo.
- Asiento: es la pieza superior y la que requiere mayor trabajo de mecanizado con herramienta tipo fresa, debido a sus dimensiones y forma.



Figura 3.7: Taburete unipersonal para adulto, tomada de [40].

En total se tienen 8 piezas que componen el producto *Taburete de cuatro patas*. La figura 3.8 muestra la representación de la fabricación de este producto mediante el modelado por medio de una red de Petri con los procesos de mecanizado que se requieren para obtener cada una de las piezas y el conteo de estas. Las transiciones muestran la operación que se debe realizar y los lugares representan la pieza procesada.

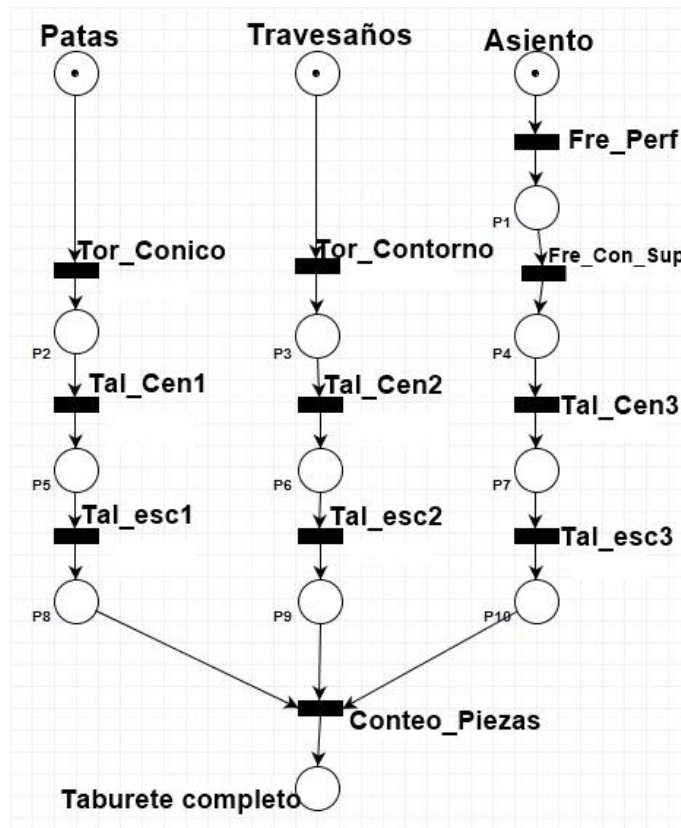


Figura 3.8¹⁶: Red de Petri del proceso de mecanizado de las piezas que componen el producto taburete de cuatro patas.

3.2.2 Silla de madera para adultos:

Esta silla clásica mostrada en la figura 3.9 está compuesta por tablonces de madera con grosores variables, estas piezas requieren varios trabajos de mecanizado, especialmente taladrado, para hacer posible su ensamblaje como un producto terminado.

De igual forma que en el producto anterior, las piezas que componen esta silla se han categorizado en 5 tipos de piezas:

- Asiento y espaldar: piezas de madera que por sus dimensiones y características de mecanizado no se puede emparentar con otras piezas del producto.

¹⁶ Las transiciones nombran las operaciones definidas anteriormente de forma abreviada; tor hace referencia a torneado, tal es taladrado, esc corresponde a escariado, cen es centrado, fre quiere decir fresado, con_sup corresponde a contorno superficial y per de perfiles.

- Travesaños: 4 piezas gruesas de madera que comparten características como dimensiones y trabajos similares por lo que se han agrupado, las piezas tienen la función de dar soporte a la silla.
- Soportes laterales: 2 piezas finas y largas que dan estabilidad a la silla.
- Patas traseras: 2 piezas grandes que tienen por función ayudar a dar soporte al marco de la silla y sostiene el espaldar.
- Patas delanteras: estas 2 piezas tienen procesos de mecanizado idénticos por lo que conforman una misma categoría.



Figura 3.9: Silla clásica de madera para adulto, tomada de [40].

Los procesos de mecanizado que requieren las 12 piezas del producto silla para adulto, son representados en la siguiente red de Petri (figura 3.10). Además, al final de la fabricación de cada pieza se realiza el conteo para finalizar con el proceso de manufactura.

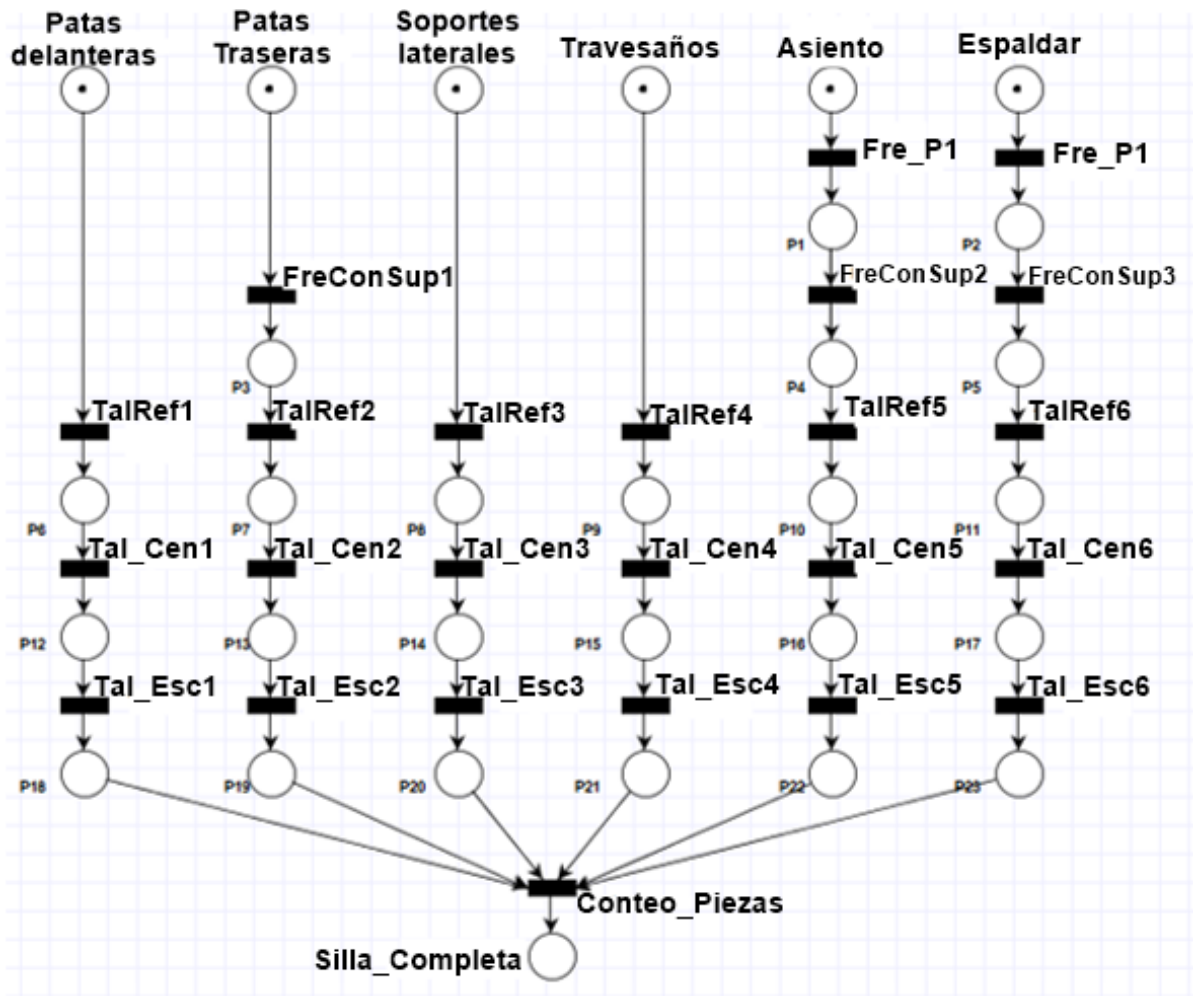


Figura 3.10¹⁷: Red de Petri del proceso de mecanizado de las piezas que componen el producto silla para adulto.

¹⁷ Las transiciones nombran las operaciones definidas anteriormente de forma abreviada; Tal es taladrado, Esc corresponde a escariado, Cen es centrado, Ref es Refrentado, Esc se refiere a escariado, Fre quiere decir fresado y ConSup corresponde a contorno superficial.

Capítulo 4: Desarrollo de la arquitectura de control para el caso de estudio

Tomando el estándar IEC 61499 y las características de los RMS, en este capítulo se desarrolla el modelo de control reconfigurable con ayuda del software 4DIAC-IDE, la justificación de la elección de esta herramienta se presenta en el anexo C. Este trabajo solo se enfoca en el nivel lógico y no en el desarrollo físico de la máquina. De esta forma se podrán obtener componentes reutilizables con interfaces estandarizadas. Por medio del estándar IEC 61499 se expone una arquitectura de control modular con componentes distribuidos, que proporciona los mecanismos para una adecuada reconfiguración [41].

Desde el entorno de ingeniería se plantea el sistema de la figura 4.1 para el desarrollo de las aplicaciones de control y reconfiguración, que serán programadas mediante el lenguaje estructurado propuesto por el estándar IEC 61131 con su representación gráfica mediante los FB's.

4.1 Desarrollo de la arquitectura mediante IEC 61499

La caracterización de los diferentes modelos propuestos por el estándar, se desarrolla desde su vista más general con el modelo de sistema inspirado en el proceso definido en el capítulo 3 hasta llegar a los bloques básicos con el nivel de detalle que se requiera. De esta forma se garantiza la modularidad del proceso y su integrabilidad. Los modelos desarrollados para el caso de estudio son los siguientes:

4.1.1 Modelos del sistema

Esta es la vista general de los elementos con los que cuenta el sistema. Se tienen dos dispositivos que son la máquina RMT y el computador con la interfaz HMI con la que el usuario puede interactuar con el sistema y cargar el diseño de los productos. El intercambio de información entre los dos dispositivos se realiza por medio de la red Ethernet, también se pueden apreciar los distintos recursos con que cuenta cada dispositivo y el color de cada uno, rojo para PC y verde para MAQUINA; de aquí en adelante los bloques que se mapeen en alguno de los recursos, se coloreará del color del dispositivo correspondiente. El modelo se presenta en la figura 4.1.

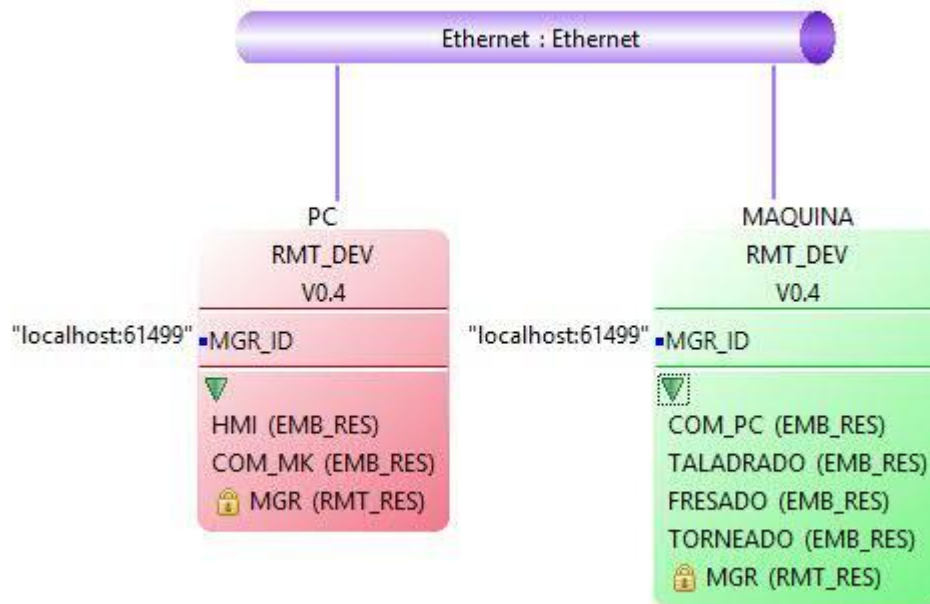


Figura 4.1: modelo de sistema para el caso de estudio.

4.1.2 modelo de dispositivo

Las figuras 4.2 y 4.3 muestran los dos dispositivos con los que cuenta el sistema que son el computador que posee la interfaz HMI y la máquina RMT, respectivamente. Aquí se muestran los diferentes recursos y las aplicaciones requeridas para su funcionamiento.

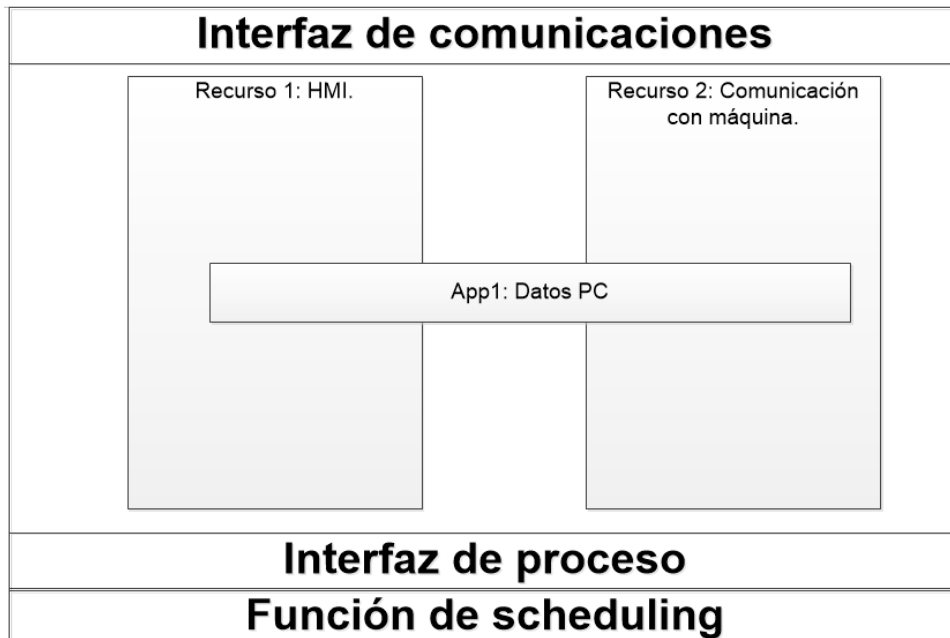


Figura 4.2: modelo de dispositivo para el HMI.

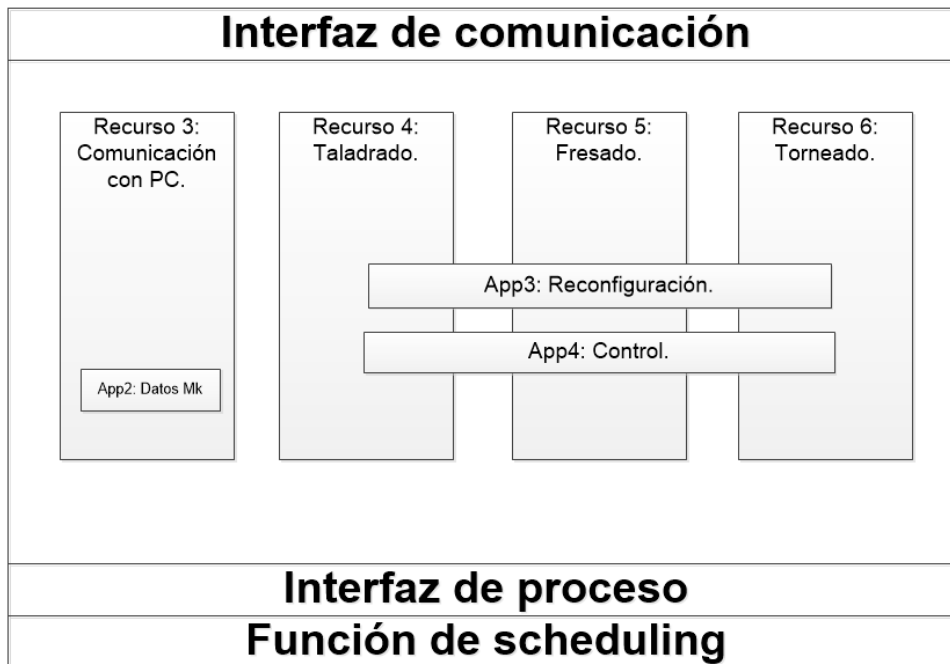


Figura 4.3: modelo de dispositivo para la RMT.

4.1.3 configuración de la comunicación

Con el fin de facilitar la comunicación entre los diferentes componentes del sistema, se han utilizado bloques de Interfaz de Servicio (SIFB), los cuales son:

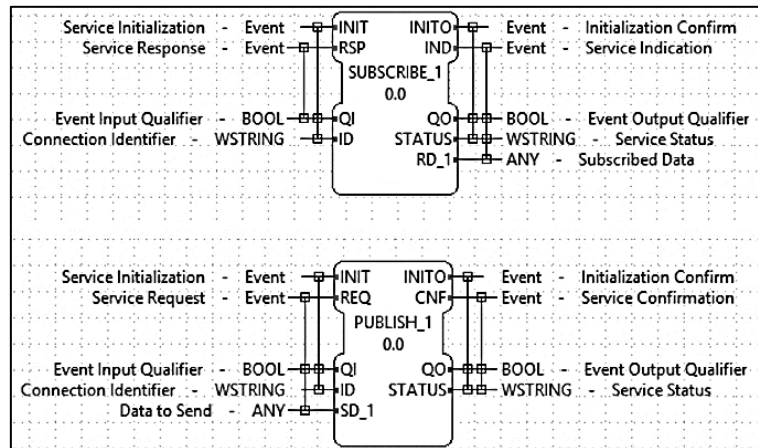


Figura 4.4: Bloques de interfaz de servicio SUBSCRIBE y PUBLISH.

Los bloques SUBSCRIBE y PUBLISH están diseñados con el fin de transmitir información entre diferentes componentes de un sistema, para ello cuentan con los eventos de entrada y salida comunes a otros bloques, sin embargo en los campos de salida de datos para el SUBSCRIBE y datos de entrada para el PUBLISH tienen dos campos especiales: ID en el cual se escribe la dirección IP del bloque con el que se quiere realizar la comunicación, para que esta se realice ambos bloques deben tener configurada la misma dirección; y RD1 que se refiere a la información recibida y SD1 que es la información que se desea transmitir, estas entradas o salidas son de tipo ANY, lo que significa que puede manejar cualquier tipo de dato, permitiendo que estos bloques sean muy flexibles y útiles.

Con estos bloques se realizan las comunicaciones a través de direcciones IP gracias a la comunicación tipo Ethernet, para repartir estas direcciones de manera ordenada se realizó la distribución que se muestra a continuación en la tabla 4.1:

Dispositivo 1	Dispositivo 2
IP: 192.168.0.0	IP: 192.168.3.0
HMI	COM_PC
192.168.1.0	192.168.4.0
COM_MK	TALADRADO
192.168.2.0	192.168.5.0
-----	FRESADO
-----	192.168.6.0
-----	TORNEADO
-----	192.168.7.0

Tabla 4.1: Distribución de direcciones IP.

Con las direcciones claramente distribuidas, es fácil saber cómo direccionar los flujos de información entre las subdivisiones del sistema, las direcciones de los FBs se definen usando los dígitos más significativos de la dirección IP del recurso al que corresponden y variando los dígitos menos significativos con el fin de evitar duplicados.

4.1.4 Modelo de aplicación

Se desarrollaron cuatro aplicaciones para cumplir con las diferentes tareas que desarrolla la RMT distribuidas a través de los diferentes recursos con los que dispone el sistema, siempre pensando en darle modularidad al mismo. Estas se muestran a continuación:

- **Aplicación app_datosPc**

En esta aplicación se encuentran los bloques encargados de la lectura de datos que se envía desde el software CAD en el que se modela la pieza a mecanizar, de igual forma que el tipo de producto que se desea realizar, el cual se obtiene a través de una HMI. Esta aplicación usa dos tipos de FB, IN_ANY el cual se encarga de permitir la entrada de información desde algún componente externo al sistema y, READER que tiene por función leer esa información, almacenarla dentro de sí para luego transmitirla y que sea útil al sistema. Esta aplicación se muestra en la figura 4.5.

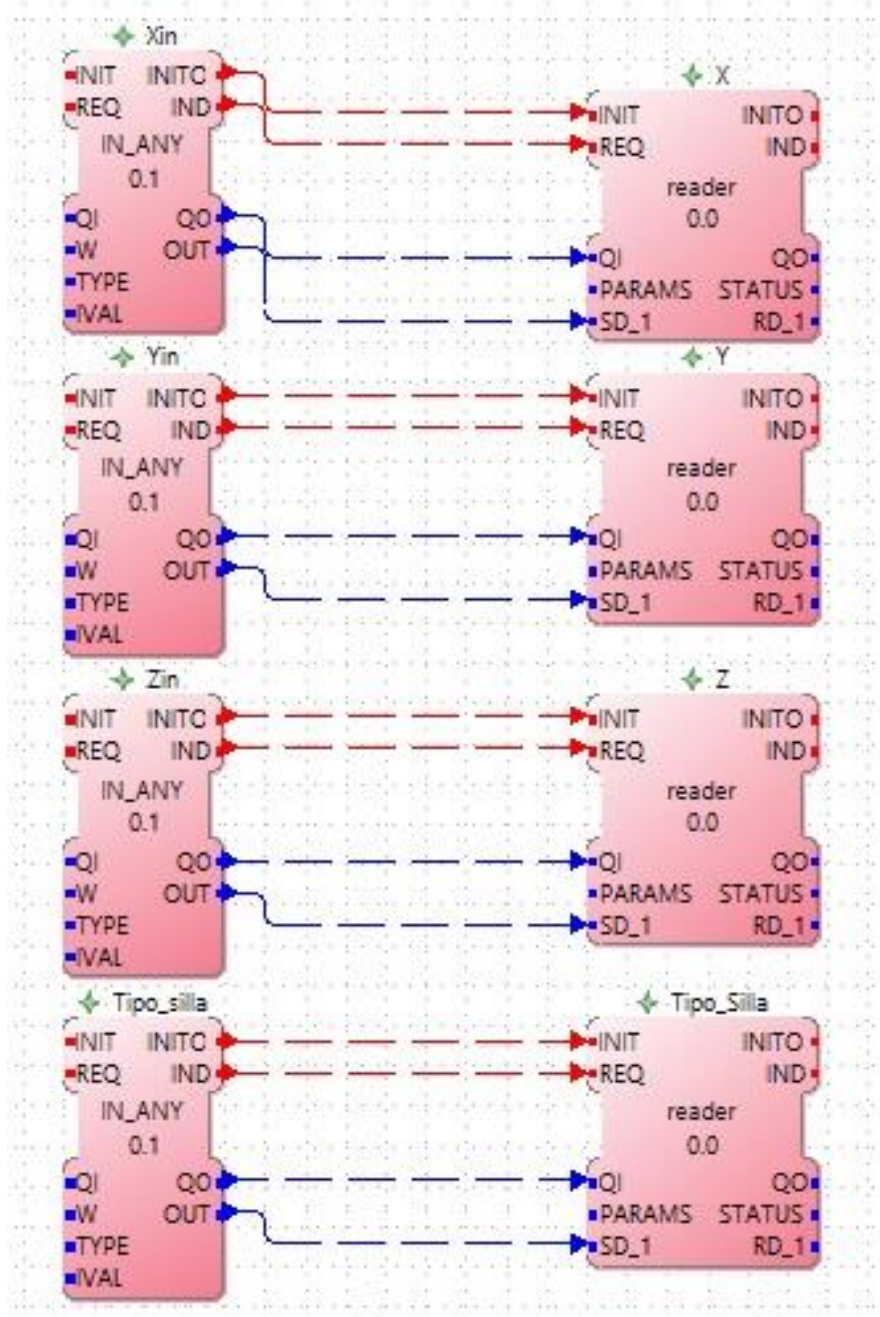


Figura 4.5: Aplicación *app_datosPc*.

- **Aplicación *app_datosMk***

La aplicación de datos de la máquina tiene la única función de albergar los bloques **READER**, que se encargan de obtener la información de los sensores del sistema. Esta se muestra en la figura 4.6.

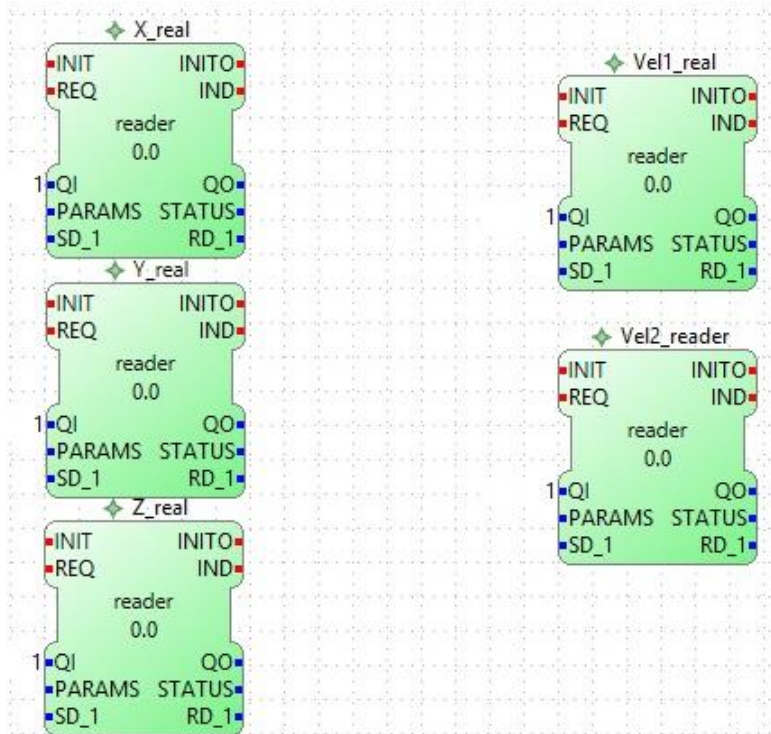


Figura 4.6: Aplicación app_datosMk.

- **Aplicación app_control_punta**

Esta aplicación es la más grande de todo el sistema, su función principal es la de hacer control sobre las velocidades y posiciones de la punta de trabajo que está en contacto directo con la pieza. A pesar de contener muchos bloques, su entendimiento es sencillo debido a que todos siguen un mismo patrón: bloques suscritores que obtienen tanto las variables de set point como los datos medidos directamente del proceso a través de sensores, esos datos entran al bloque PID donde se ejecutan los cálculos del control y finalmente se escribe el resultado o esfuerzo de control a través de bloques WRITER, los cuales tienen acceso a las salidas físicas del dispositivo. Debido al amplio rango de acción de esta aplicación, se ha distribuido entre los tres recursos de mecanizado. La aplicación se muestra en la figura 4.7.

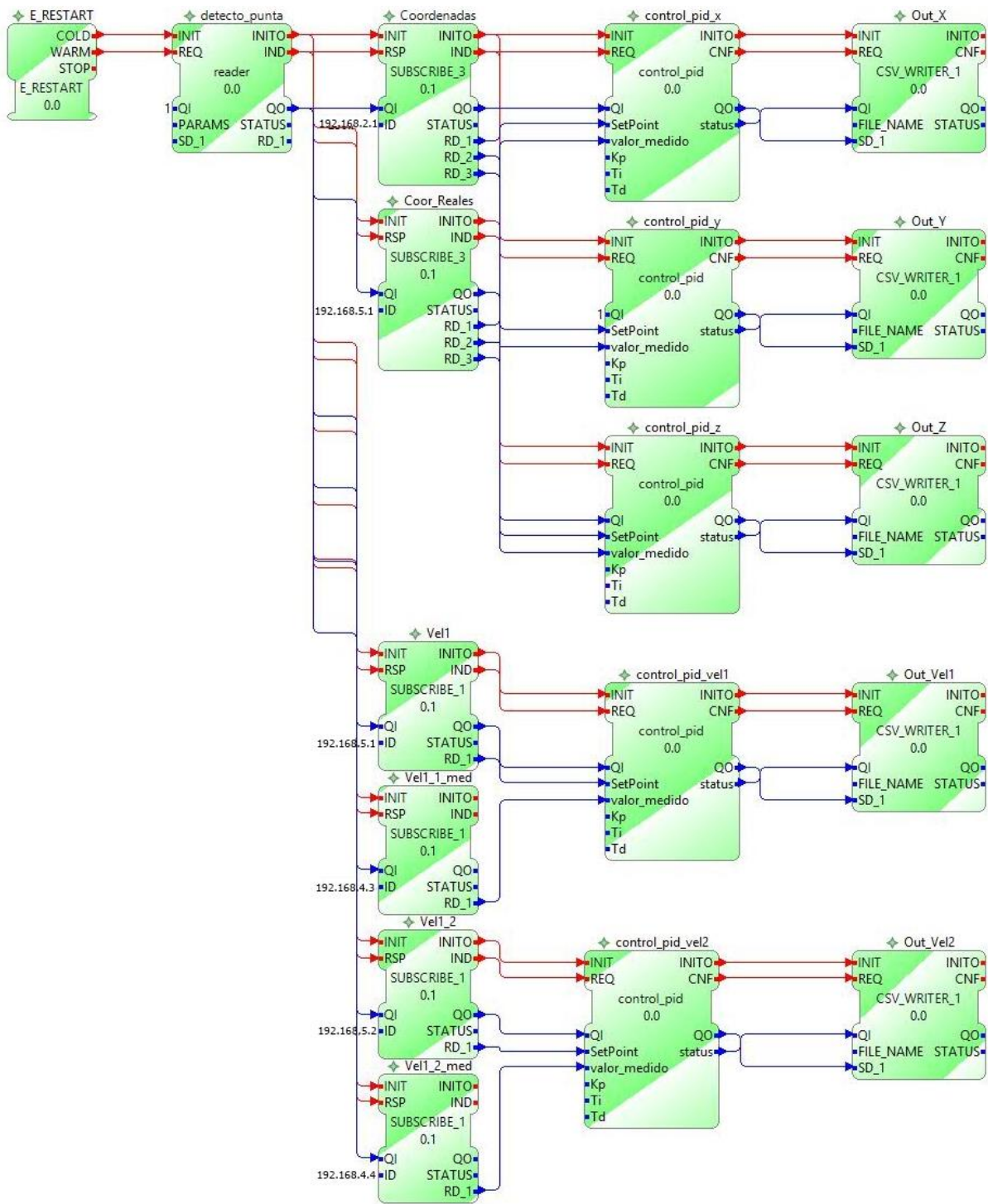


Figura 4.7: Aplicación `app_control_punta`.

- **Aplicación app_Reconfiguracion**

La aplicación de reconfiguración es el corazón del sistema y de este proyecto, es una aplicación que se encarga de reconocer el producto que se desea realizar y con base en eso, decidir qué operación de manufactura se debe efectuar; para realizar esa tarea se compone de un bloque subscritor de donde obtiene la información del producto, el bloque compuesto Tipo_Operacion que se encarga de seleccionar la tarea de mecanizado y la Subaplicación de control en la cual se calculan las velocidades de los motores con base en el tipo de operación. La figura 4.8 muestra la representación de esta aplicación.

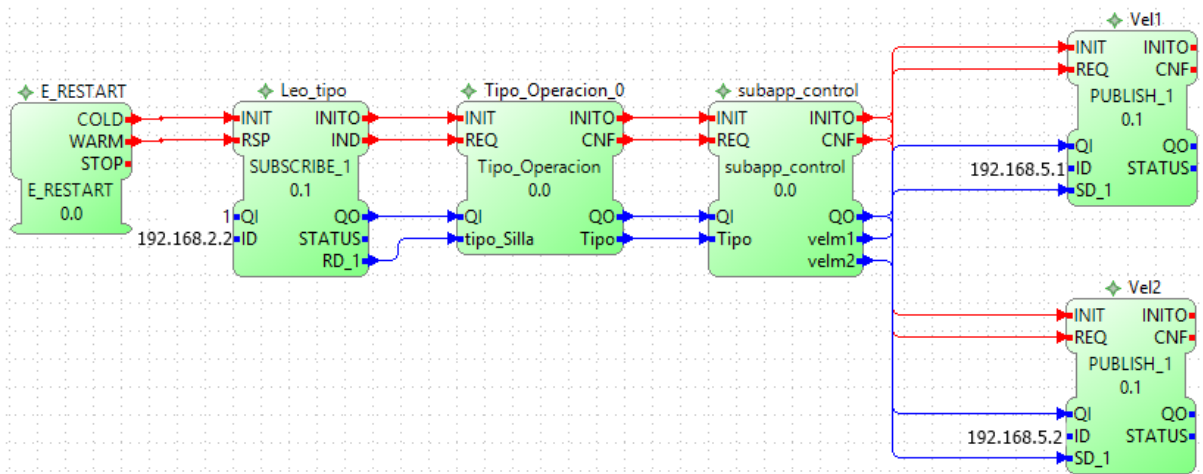


Figura 4.8: Aplicación app_Reconfiguracion.

La subaplicación que se puede observar en la figura 4.8, representada mediante el FB subapp_control se presenta en la figura 4.9 con su red de trabajo interna. Esta lee las coordenadas de la forma que debe adoptar la pieza, el bloque decision (Figura 4.19), inicializa cada uno de los calculadores de velocidad (Figuras 4.16 y 4.17) dependiendo del tipo de operación y estos después de hacer sus cálculos internos, generan la salida de esta red embebida, estos bloques se explican más adelante.

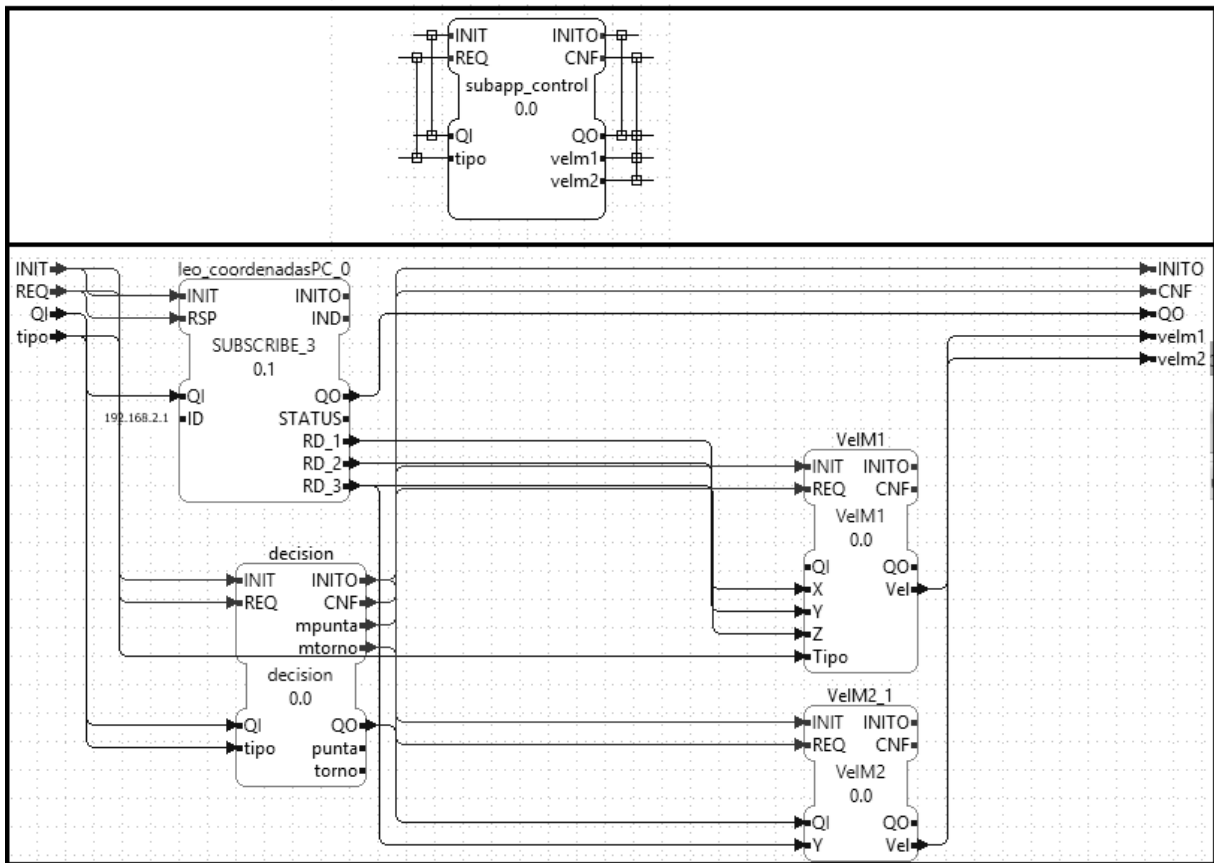


Figura 4.9: subaplicación_control.

4.1.5 Modelo de recurso

Se han definido las tres actividades de taladrado, fresado y torneado como recursos independientes que permiten la ejecución de todas las operaciones definidas para la RMT del caso de estudio. Además, se cuenta con el HMI como un recurso y la comunicación para la interacción entre los dos dispositivos, y también las aplicaciones distribuidas encargadas de coordinar el funcionamiento de los diferentes módulos que componen el sistema. A continuación, se presentan los diferentes modelos para cada uno de ellos.

- **Recurso: HMI**

Sobre el recurso HMI se han mapeado los bloques IN_ANY de la aplicación que administra los datos del PC (figura 4.5) con el fin de que puedan ingresar estos al sistema, en la parte inferior el conjunto de subscritor junto con los bloques OUT_ANY imprime en la pantalla del operador las mediciones de la dimensión real del producto

obtenidas por los sensores de la máquina. El modelado de este recurso se muestra en la figura 4.10.

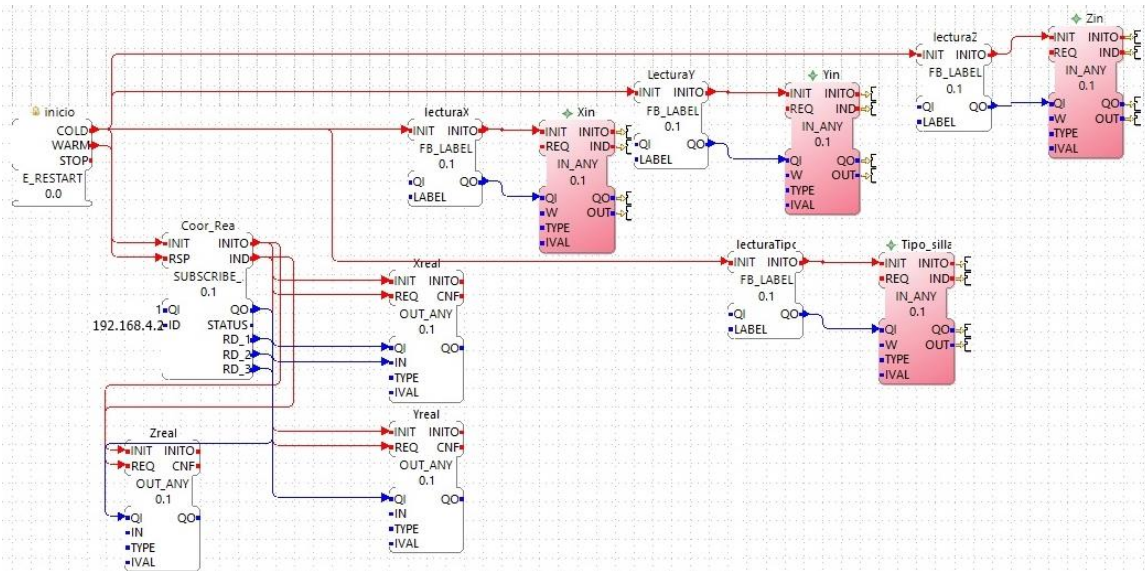


Figura 4.10: recurso PC.HMI.

- **Recurso: comunicación del PC hacia la máquina COM_MK**

En este recurso se terminan de mapear el resto de los bloques que componen la aplicación que administra los datos del PC, y la función de este recurso es transmitir esa información hacia la máquina para que pueda ejecutar el mecanizado con base en esos parámetros. El modelado de este recurso se muestra en la figura 4.11.

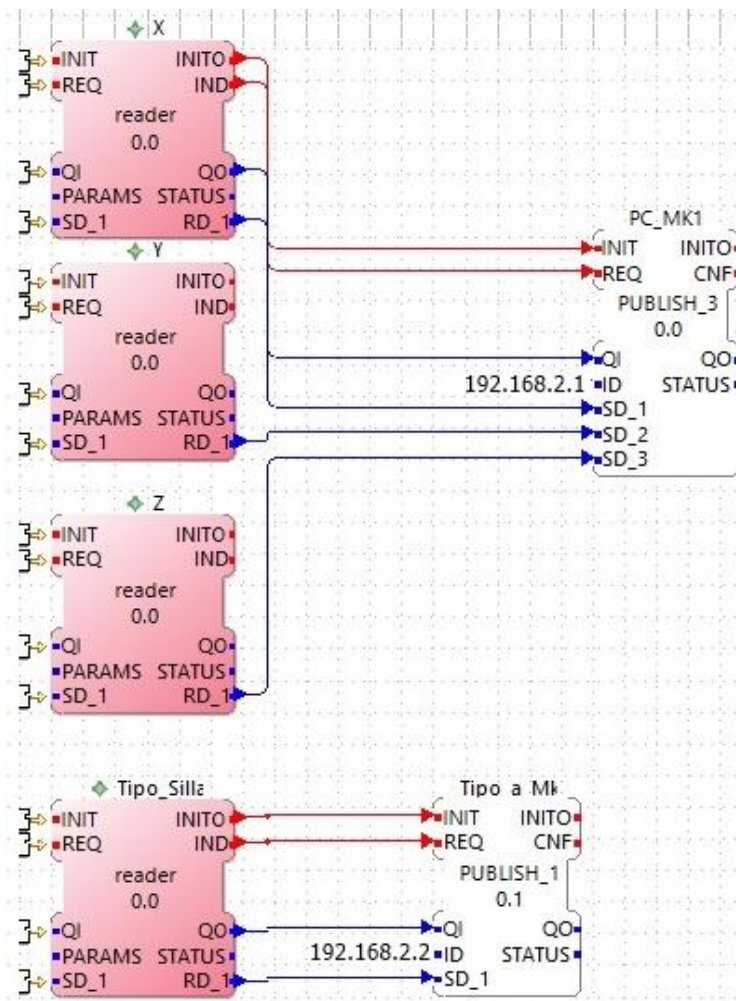


Figura 4.11: recurso PC.COM_MK.

- **Recurso: comunicación de la máquina hacia el PC COM_PC**

Este recurso del dispositivo Máquina, es el encargado de recibir y enviar información hacia el otro dispositivo y hacia otras aplicaciones y recursos del dispositivo, para eso se vale de los bloques de la aplicación que administra los datos de la máquina (figura 4.6) y de bloques subscriptores. El modelado de este recurso se muestra en la figura 4.12.

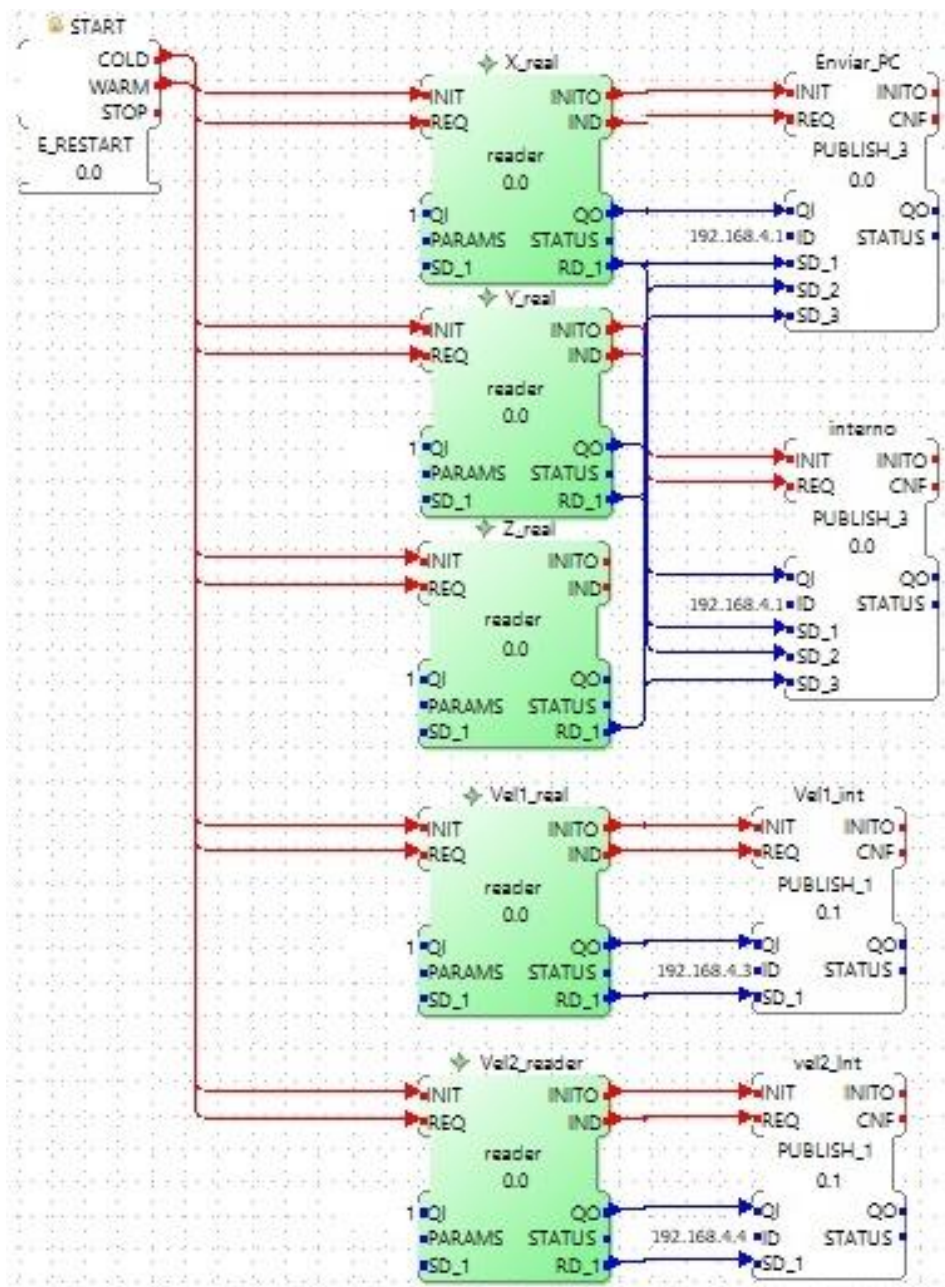


Figura 4.12: Recurso MAQUINA.COM_PC.

- **Recurso: TALADRADO**

Este recurso sirve de contenedor para las acciones de control de las variables espaciales X y Y, esta estructura de control está mapeada desde la aplicación de control de la punta (Figura 4.7). El modelado de este recurso se muestra en la figura 4.13.

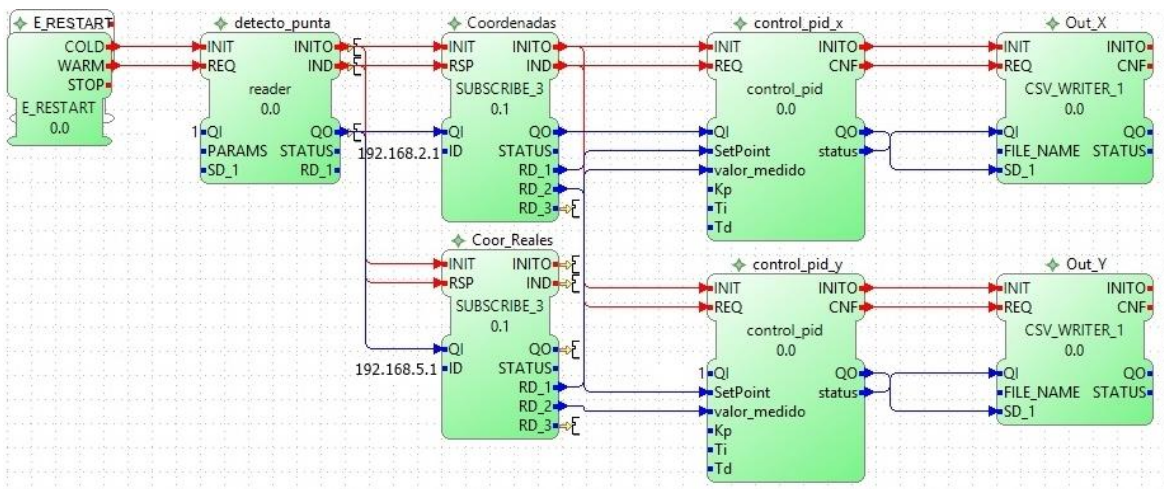


Figura 4.13: Recurso MAQUINA.TALADRADO.

- **Recurso: FRESADO**

Este recurso sirve de contenedor para las acciones de control de la variable Z y la velocidad del motor 1, es decir el motor de la punta, esta estructura de control está mapeada desde la aplicación de control de la punta (figura 4.7). El modelado de este recurso se muestra en la figura 4.14.

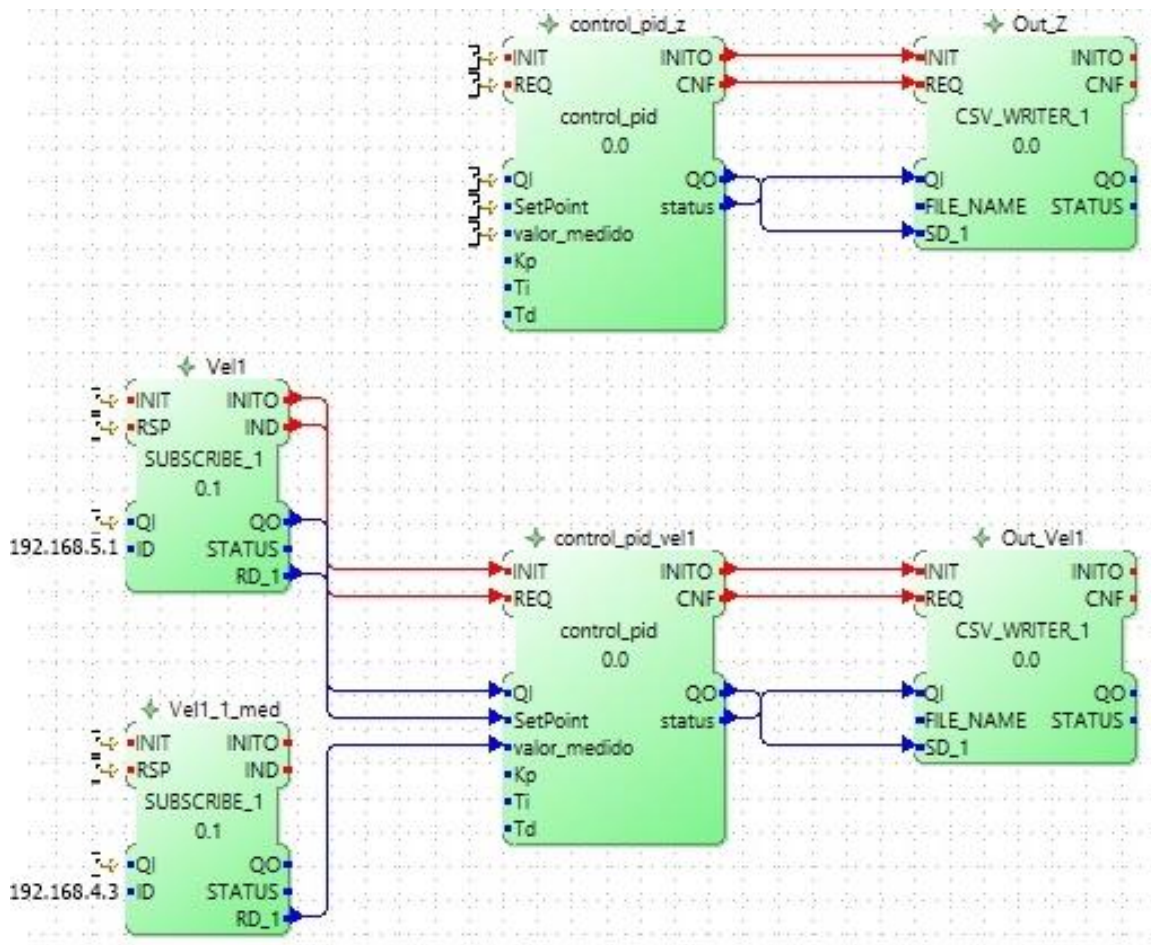


Figura 4.14: Recurso MAQUINA.FRESADO.

- **Recurso: TORNEADO**

Este recurso sirve de contenedor para las acciones de control de la velocidad del motor 2, es decir el motor que hace girar la pieza para ser torneada, esta estructura de control está mapeada desde la aplicación de control de la punta (figura 4.7). Además contiene a la función de reconfiguración (figura 4.8). El modelado de este recurso se muestra en la figura 4.15.

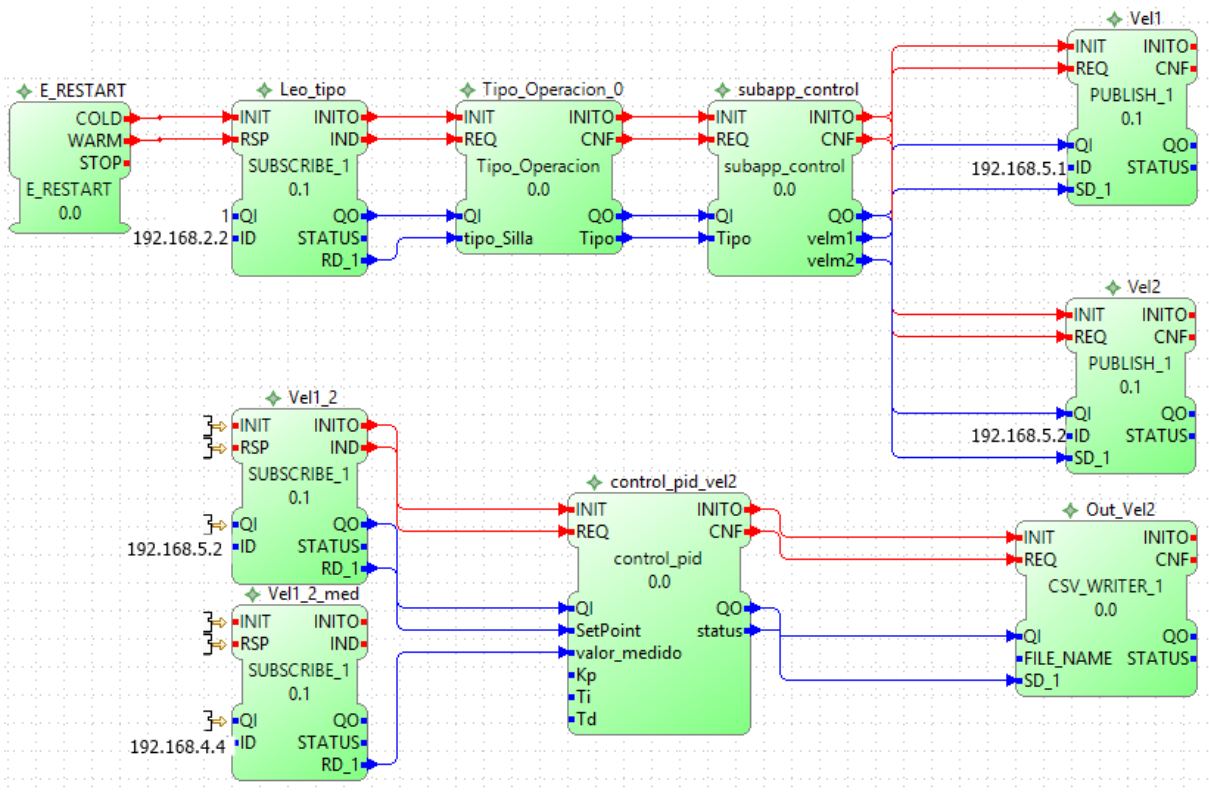


Figura 4.15: Recurso MAQUINA.TORNEADO.

4.1.6 Modelo de bloque de función

A continuación se exponen los diferentes bloques de función que se crearon con el fin de personalizar el sistema y darle las características de reconfiguración requeridas.

- **Bloques para la configuración de la velocidad**

Se cuenta con dos bloques compuestos VELM1 para el motor de la punta que actúa sobre la pieza de trabajo, el cual se muestra en la figura 4.16 y el bloque VELM2 para el motor que hace que gire la pieza constantemente mientras se realiza la operación de torneado, este se muestra en la figura 4.17. En la parte inferior de las figuras se puede observar la red de trabajo que compone a los bloques.

El FB de la figura 4.16 contiene la información de la velocidad que requiere el motor principal que es el que manipula la punta que actúa sobre el producto. Se tienen dos eventos de entrada, INIT es el encargado de inicializar el FB y permite que todas las

variables de entrada asociadas a este puedan muestrear los valores que le entran, al igual que el evento REQ con sus variables asociadas. Los datos de entrada son QI que al leer un valor verdadero realiza el cálculo correspondiente para este FB, las coordenadas x, y y z que son tomadas para los diferentes tipos de operación dependiendo de lo que se vaya a realizar y la variable tipo, para saber si se va a taladrar o fresar definidos como 1 y 2, respectivamente. A la salida se tiene el estado del bloque y el valor de la velocidad con la que va a operar el motor, estos valores salen cuando el bloque se ha ejecutado con éxito y activa los eventos de salida indicando su finalización y su disponibilidad con INITO y CNF, respectivamente.

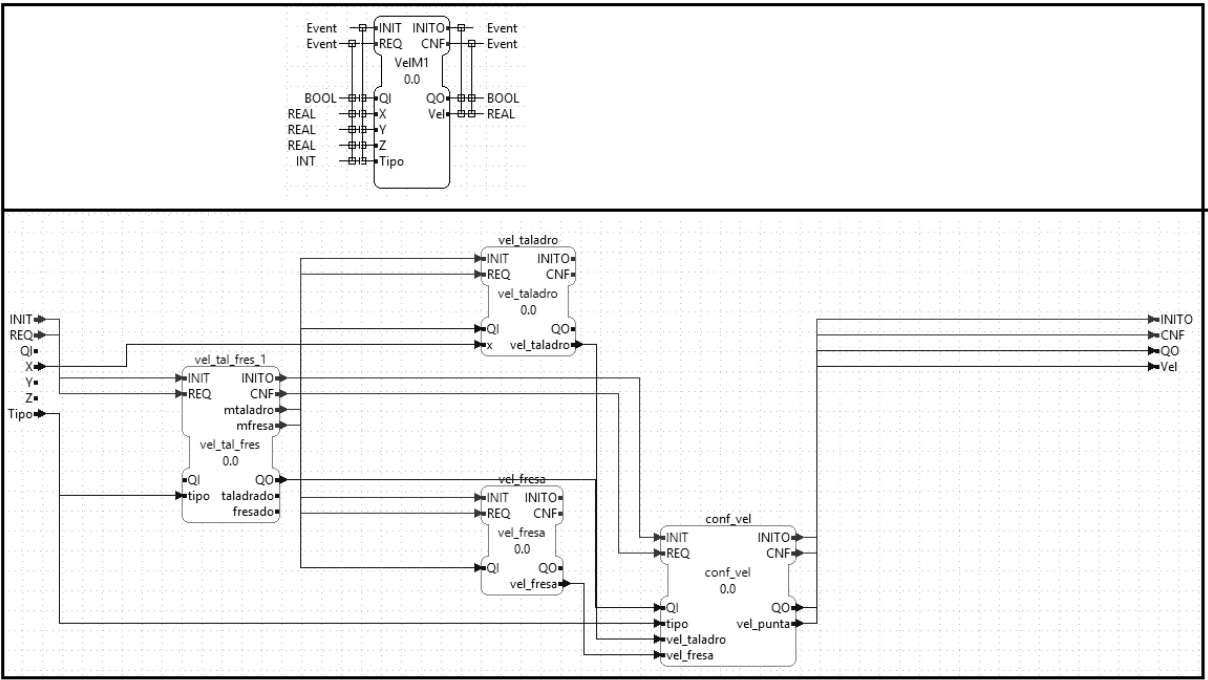


Figura 4.16: FB VELM1.

El bloque VELM2 de la figura 4.17 define la velocidad para el motor secundario cuando es requerido en el proceso de torneado, utilizado para rotar constantemente la pieza de trabajo. Se tienen dos eventos de entrada, INIT es el encargado de inicializar el FB y permite que todas las variables de entrada asociadas a este puedan muestrear los valores que le entran y el evento REQ con sus variables asociadas para realizar el cálculo de la velocidad. Los datos de entrada son QI para el estado del bloque que al ser verdadero permite calcular la velocidad y la coordenada Y, que es el eje de desplazamiento sobre el cual se mueve la punta que transforma la pieza de trabajo. A la salida se tiene el estado del proceso realizado por

el bloque y *vel* que es el valor de la velocidad con la que va a operar el motor para el torneado. Con la ejecución del bloque se activan los eventos de salida asociados con sus respectivas variables que son INITO, indicando la finalización del cálculo realizado en la red de trabajo interna y CNF, indicando que el bloque ya está disponible para un nuevo cálculo.

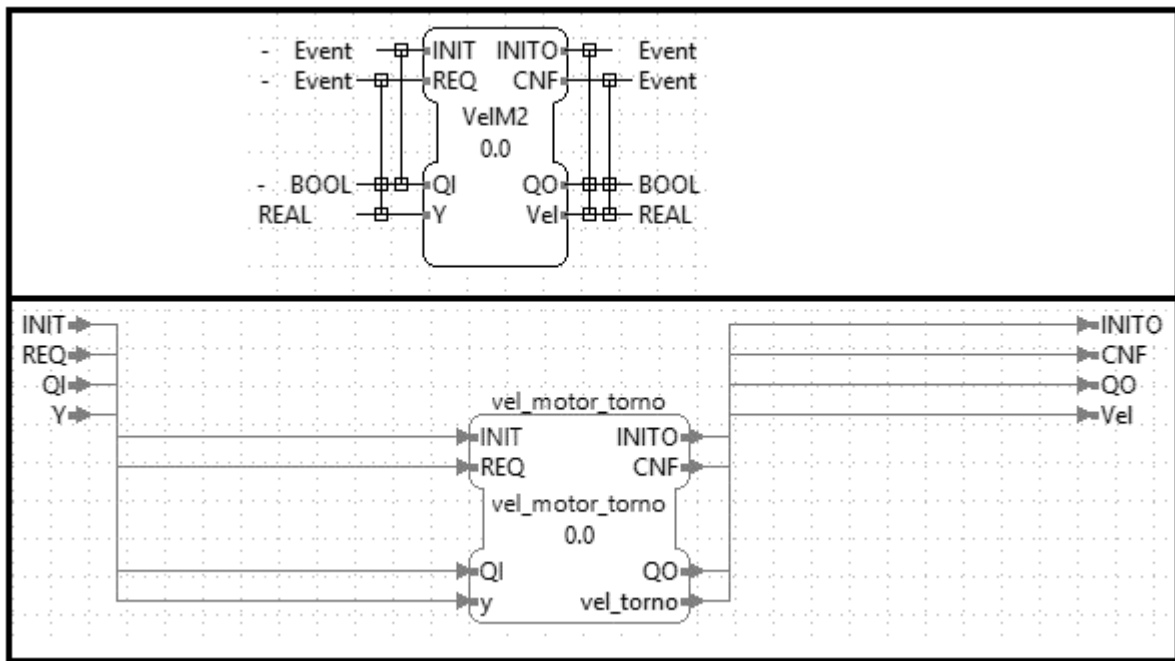


Figura 4.17: FB VELM2.

Se tienen los bloques: *vel_fresa*, *vel_taladro* y *vel_motor_torno* encargados de hacer el cálculo de la velocidad de rotación que se presenta en el capítulo 3. A manera de ejemplo se muestra el bloque *vel_taladro* en la figura 4.18 (para más información de los demás bloques ver el anexo D).

La figura 4.18 muestra a la izquierda la representación del bloque con sus respectivos eventos y datos de entrada y salida; a la derecha el gráfico de control de ejecución con los eventos y algoritmos asociados, estos en lenguaje estructurado ST como lo estipula el estándar IEC 61499. Este bloque se encarga de calcular la velocidad de rotación que se muestra en la ecuación 3.2 del capítulo 3. Se tiene el evento de entrada INIT para inicializar el bloque y el evento REQ que permite realizar el cálculo de la velocidad para la operación de taladrado. Entre los datos de salida se tiene QI que toma un valor TRUE para realizar el cálculo de la velocidad y se tiene *x* para definir la dimensión del diámetro de la broca con la que se corte la pieza de

trabajo. Como datos de salida se tiene QO, indicando el estado actual de la variable calculada en este bloque que es la velocidad de rotación y *vel_taladro* que es el valor calculado. Por último, los eventos de salida son INITO indicando el fin de su ejecución y CNF para indicar que el bloque ya está disponible para un nuevo cálculo.

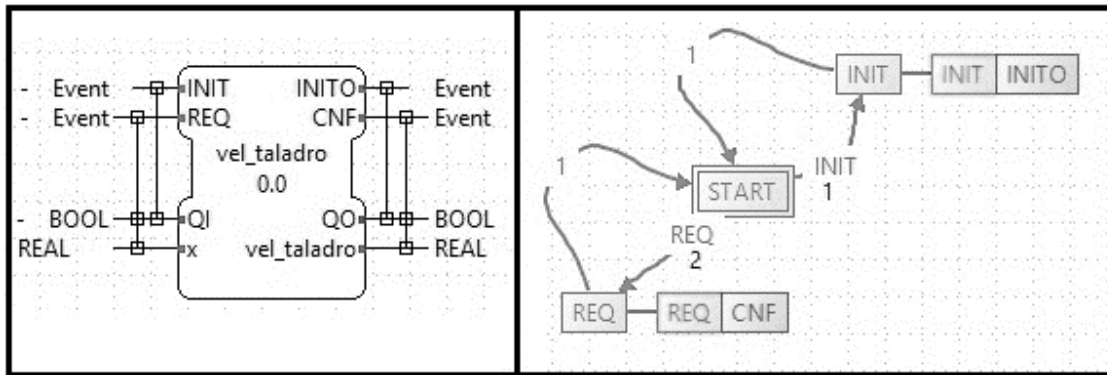


Figura 4.18: FB *vel_taladro*.

Los algoritmos para el bloque *vel_taladro* se presentan a continuación:

Algoritmo INIT

```
vel_corte := 7500; /*mm/min*/
```

Algoritmo REQ

```
vel_taladro := vel_corte/(3.1416*x);  
QO := TRUE;
```

El FB *decision* mostrado en la figura 4.19 es el encargado de leer el tipo de operación que se desea realizar (torneado, taladrado o fresado) para saber si solo debe activar el motor encargado de manejar la punta o el que mueve la pieza para torneear. Para esto se definen las tres operaciones con números enteros, donde 1 es taladrado, 2 es fresado y 3 torneado, estos valores entran mediante la variable *tipo*. Se tiene el evento de entrada INIT para inicializar el bloque y el evento REQ para iniciar la toma de la decisión. A la entrada se tiene el dato de QI que lee un valor TRUE para ejecutar el bloque y el tipo de operación a realizar, como datos de salida se tiene QO indicando el estado del FB, *punta* para activar el motor de la punta que actúa sobre la pieza de trabajo y *torno* para activar el motor secundario utilizado en la operación de torneado. Por último, los eventos de salida que son INITO indicando el fin de su ejecución, CNF para indicar que el bloque ya está disponible y los eventos: *mpunta* y *mtorno* que activan los bloques que realizan los cálculos de la velocidad de los motores de la punta y el torno, respectivamente.

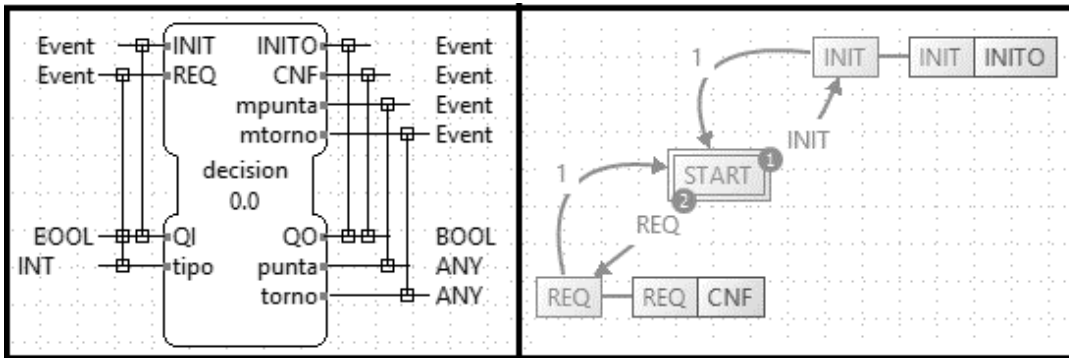


Figura 4.19: FB decision.

El algoritmo realizado para el bloque de la figura 4.7 es el siguiente:

Algoritmo INIT:

QI := **TRUE**;

Algoritmo REQ

IF tipo =1 **THEN**

 QO := **TRUE**;
 punta := **TRUE**;
 torno := **FALSE**;

ELSE

IF tipo = 2 **THEN**

 QO := **TRUE**;
 punta := **TRUE**;
 torno := **FALSE**;

ELSE

IF tipo = 3 **THEN**

 QO := **TRUE**;
 punta := **FALSE**;
 torno := **TRUE**;
 END_IF;

END_IF;

END_IF;

El FB *vel_tal_fres* cumple con la misma función que el bloque decisión solo que este decide si debe activar la velocidad para que la máquina taladre o frese. De la misma forma el FB *conf_vel* recibe las velocidades del taladro o la fresa, de acuerdo a la operación que se va a realizar y envía la velocidad de la punta calculada (la explicación de todos los bloques se puede ver en el anexo D).

- **Bloques para la configuración del producto a realizar**

Como ya se ha dicho en el capítulo 3 se definieron dos modelos de sillas como productos base para la generación de la familia de productos, estas se fabrican mediante las operaciones estándar ya definidas. La fabricación del producto se desarrolló mediante un bloque compuesto llamado *Tipo_Operacion*, este se representa en la figura 4.20, en la parte superior se tiene su representación física con sus eventos y variables de entrada y salida, y en la parte inferior la red de trabajo que lo compone.

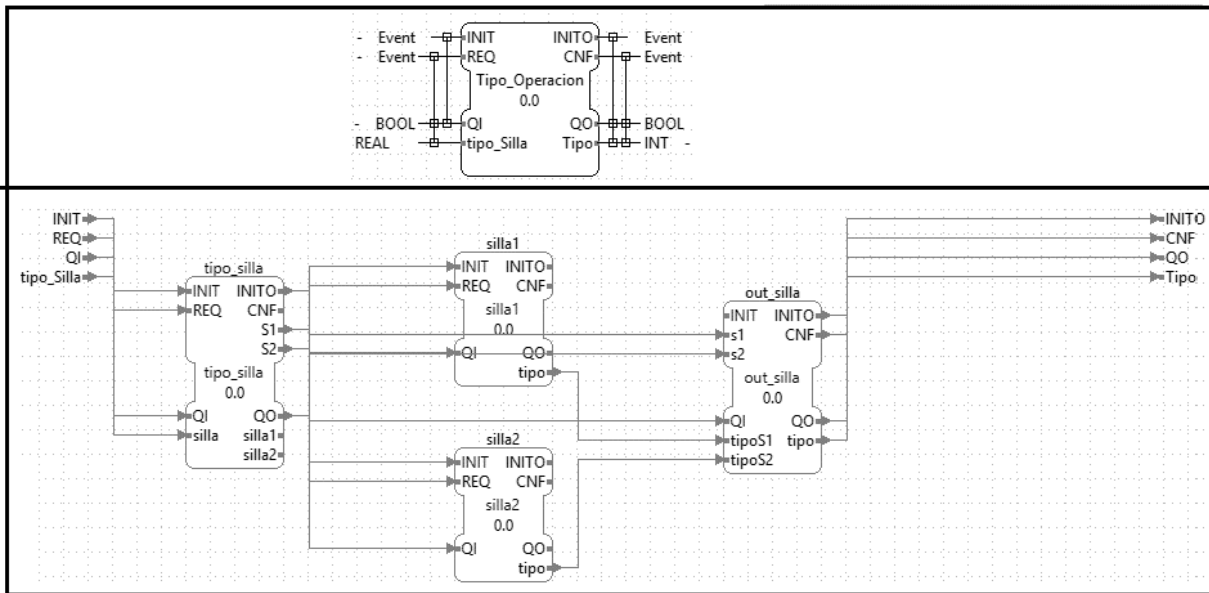


Figura 4.20: FB *Tipo_Operacion*.

Tipo_Operacion se ha creado con el fin de identificar el producto que se va a realizar y de acuerdo a esto activar las diferentes operaciones que se requieren y las piezas que se deben producir. El evento *INIT* inicializa el bloque de función y permite la lectura de la entrada *QI* que envía un estado verdadero para que se realice el proceso definido para este bloque. Se definieron dos modelos de sillas que entran por la variable de entrada *tipo_silla* que puede ser 1 o 2, la lectura de esta variable se realiza mediante la activación del evento *REQ* que ejecuta la red de trabajo y sale la variable *tipo* que define la secuencia de operaciones según el producto.

Para conocer el tipo de silla y activar el proceso para su fabricación correspondiente se tiene el bloque llamado *tipo_silla* que se muestra en la figura 4.21 con su respectivo ECC. Este FB cuenta con el evento de entrada *INIT* para inicializar el

bloque y el evento REQ que permite realizar su función. A la entrada se tiene el dato de QI que toma un valor TRUE para ejecutar el bloque y silla que indica el producto que se va a fabricar. Como datos de salida se tiene QO indicando el estado del FB y silla1 y silla2 que activan los eventos de salida asociados. Por último, los eventos de salida que son INITO indicando el fin de su ejecución, CNF para indicar que el bloque ya está disponible y los eventos que activan el procedimiento para el producto solicitado S1 y S2.

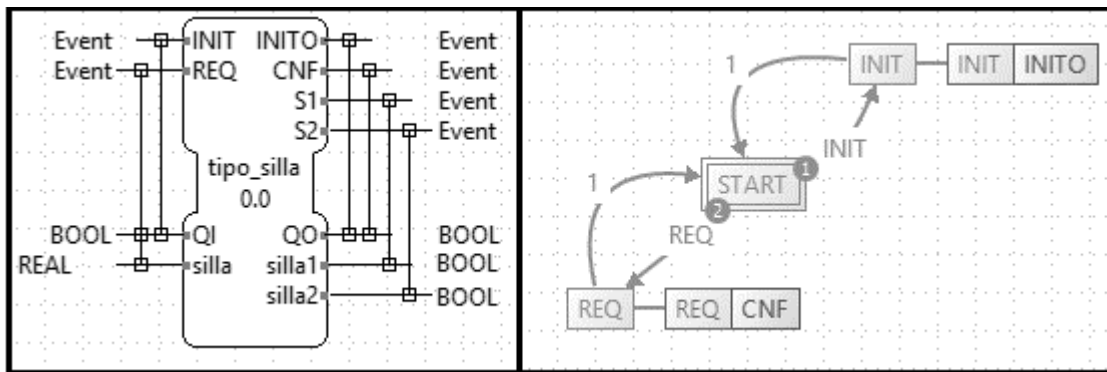


Figura 4.21: FB tipo_silla.

Los algoritmos presentes en el bloque de la figura 4.21 son:

Algoritmo INIT

QI := TRUE;

Algoritmo REQ

IF silla = 1 **THEN**

silla1 := TRUE;

silla2 := FALSE;

QO := TRUE;

ELSE

IF silla = 2 **THEN**

silla2 := TRUE;

silla1 := FALSE;

QO := TRUE;

END_IF;

END_IF;

Los bloques llamados silla1 y silla2 contienen las operaciones que se deben realizar y el conteo de las piezas obtenidas para la fabricación de estas. A manera de ejemplo se muestra el FB silla1 con su respectivo ECC en la figura 4.22 y las variables internas definidas ahí (la explicación del FB silla2 se puede ver en el anexo D). El evento de entrada INIT inicializa el bloque y el evento REQ permite iniciar las operaciones para este producto, a la entrada se tiene el dato QI que lee el valor

TRUE para ejecutar el bloque; como datos de salida se tiene QO indicando el estado del FB y tipo con el tipo de producto fabricado, los eventos de salida son: INITO indicando el fin de la ejecución y CNF para indicar que el bloque ya está disponible.

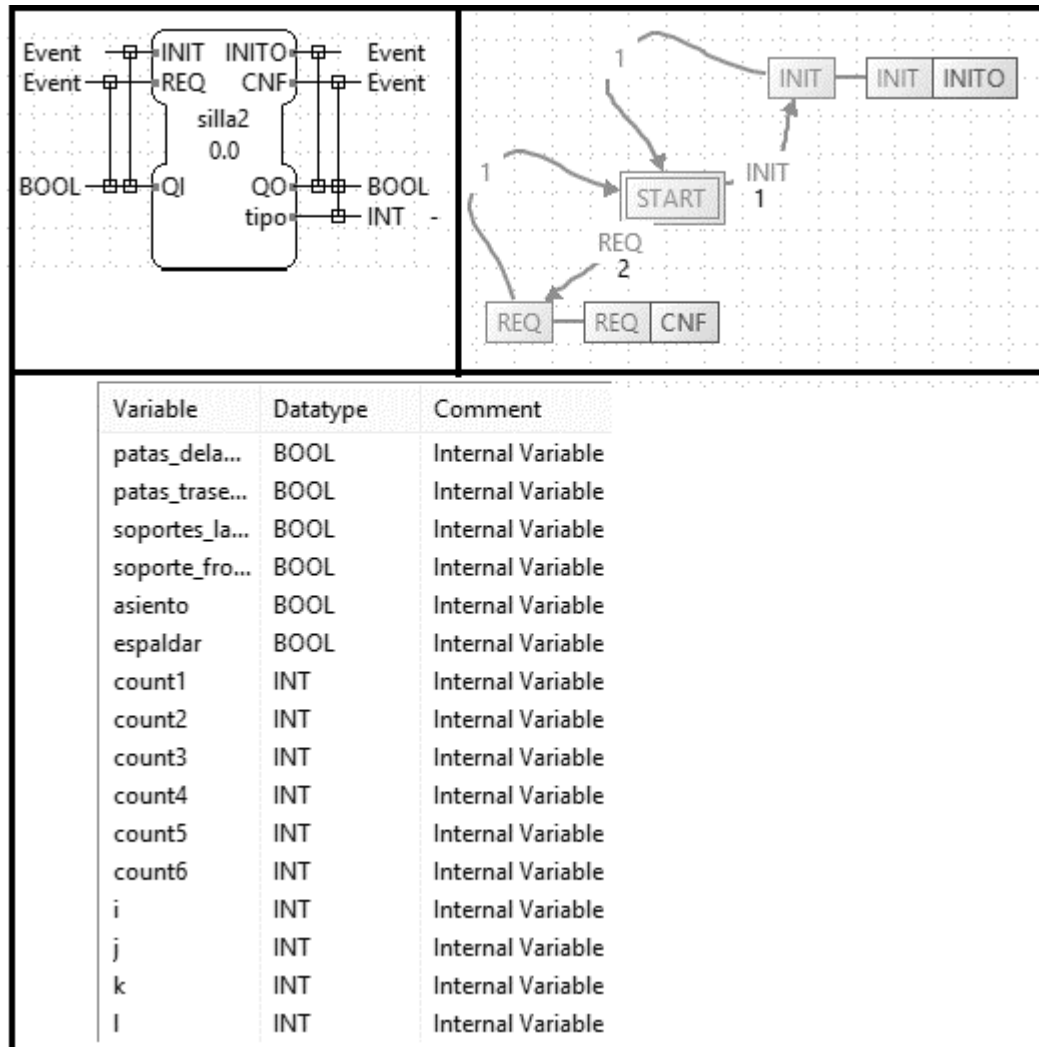


Figura 4.22: FB silla1.

El algoritmo ejecutado en el bloque de la figura 4.22 se muestra a continuación:

Algoritmo INIT

QI := **TRUE**;

Algoritmo REQ:

i := 0;

j := 0;

k := 0;

l := 0;

count1 := 0;

```

count2 := 0;
count3 := 0;
count4 := 0;
count5 := 0;
count6 := 0;
patas_delanteras := FALSE;
patas_traseras := FALSE;
soportes_laterales := FALSE;
soporte_frontal := FALSE;
asiento := FALSE;
espaldar := FALSE;
/*patas_delanteras 2*/
WHILE i < 2 DO
    WHILE count1 < 3 DO
        IF count1 = 0 THEN
            tipo := 2;
        ELSE
            IF count1 = 1 THEN
                tipo := 1;
            ELSE
                IF count1 = 2 THEN
                    tipo := 1;
                END_IF;
            END_IF;
        END_IF;
        count1 := count1 + 1;
    END_WHILE;
    i := i + 1;
    patas_delanteras := TRUE;
END_WHILE;
/*patas_traseras 2*/
WHILE j < 2 DO
    WHILE count2 < 4 DO
        IF count2 = 0 THEN
            tipo := 2;
        ELSE
            IF count2 = 1 THEN
                tipo := 2;
            ELSE
                IF count2 = 2 THEN
                    tipo := 1;
                ELSE
                    IF count2 = 2 THEN
                        tipo := 1;
                    END_IF;
                END_IF;
            END_IF;
        END_IF;
        count2 := count2 + 1;
    END_WHILE;
    j := j + 1;
    patas_traseras := TRUE;
END_WHILE;

```



```

/*soportes_laterales 2*/
WHILE k < 2 DO
    WHILE count3 < 3 DO
        IF count3 = 0 THEN
            tipo := 2;
        ELSE
            IF count3 = 1 THEN
                tipo := 1;
            ELSE
                IF count3 = 2 THEN
                    tipo := 1;
                END_IF;
            END_IF;
        END_IF;
        count3 := count3 + 1;
    END_WHILE;
    k := k + 1;
    soportes_laterales := TRUE;
END_WHILE;
/*soporte_frontal 4*/
WHILE l < 4 DO
    WHILE count4 < 4 DO
        IF count4 = 0 THEN
            tipo := 2;
        ELSE
            IF count4 = 1 THEN
                tipo := 2;
            ELSE
                IF count4 = 2 THEN
                    tipo := 1;
                ELSE
                    IF count4 = 2 THEN
                        tipo := 1;
                    END_IF;
                END_IF;
            END_IF;
        END_IF;
        count4 := count4 + 1;
    END_WHILE;
    l := l + 1;
    soporte_frontal := TRUE;
END_WHILE;
/*asiento */
WHILE count5 < 4 DO
    IF count5 = 0 THEN
        tipo := 2;
    ELSE
        IF count5 = 1 THEN
            tipo := 2;
        ELSE
            IF count5 = 2 THEN
                tipo := 1;
            ELSE

```

```

                                IF count5 =3 THEN
                                tipo :=1;
                                END_IF;
                                END_IF;
                                END_IF;
                                END_IF;
                                count5 := count5 +1;
                                asiento := TRUE;
                                END_WHILE;
/*espaldar */
WHILE count6 <4 DO
                                IF count6 = 0 THEN
                                tipo := 3;
                                ELSE
                                IF count6 = 1 THEN
                                tipo := 1;
                                ELSE
                                IF count6 =2 THEN
                                tipo :=1;
                                ELSE
                                IF count6 =3 THEN
                                tipo :=1;
                                END_IF;
                                END_IF;
                                END_IF;
                                END_IF;
                                count6 := count6 +1;
                                espaldar := TRUE;
                                END_WHILE;

```

Por último está el FB *out_silla* que lee el tipo de silla solicitada y envía la secuencia de fabricación del producto, este bloque se muestra en la figura 4.23. Se cuenta con el evento de entrada INIT para inicializar el bloque y los eventos *s1* y *s2* para saber el tipo de silla a realizar; a la entrada están los datos QI que toma un valor TRUE para ejecutar el bloque y la secuencia de fabricación *tipoS1* y *tipoS2*. Como datos de salida se tiene QO indicando el estado del FB y *tipo* indicando la secuencia requerida; al final, los eventos de salida que son INITO indicando el fin de su ejecución y CNF para indicar que el bloque ya está disponible para ser utilizado nuevamente.

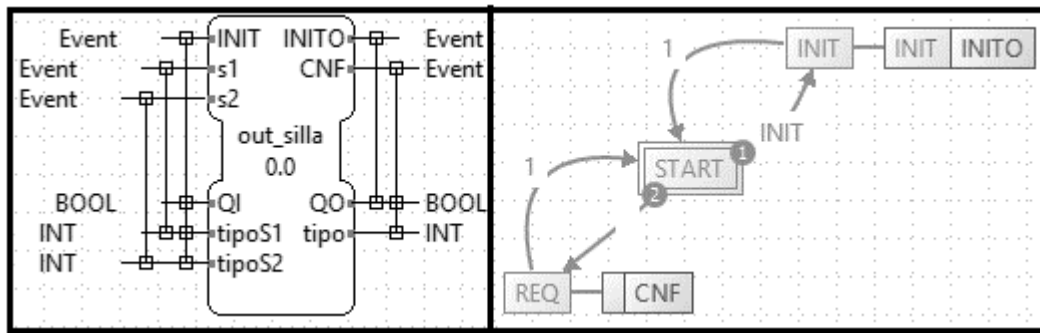


Figura 4.23: FB out_silla.

Los algoritmos ejecutados en el bloque out_silla son los siguientes:

Algoritmo INIT:

QI := **TRUE**;

Algoritmo s1:

tipo := tipoS1;

QO := **TRUE**;

Algoritmo s2:

tipo := tipoS2;

QO := **TRUE**;

- **Bloques de control**

El desarrollo del bloque de control PID se realiza mediante la creación de un bloque compuesto, la representación de este con su red de trabajo se muestra en la figura 4.24. Este bloque se inicializa mediante el evento INIT y el evento REQ para la lectura de los datos de entrada, a la entrada se tiene el estado QI que al ser verdadero ejecuta el control PID, el valor de referencia *SetPoint* y la variable medida desde el proceso en *valor_medido*, las constantes proporcional K_p , derivativa T_d e integral T_i ; tras su ejecución se genera el esfuerzo de control requerido y el estado que indica la finalización de la ejecución del bloque, con esto se activan los eventos de salida INITO y CNF que indican que ha finalizado la ejecución y que se puede realizar nuevamente el proceso, respectivamente.

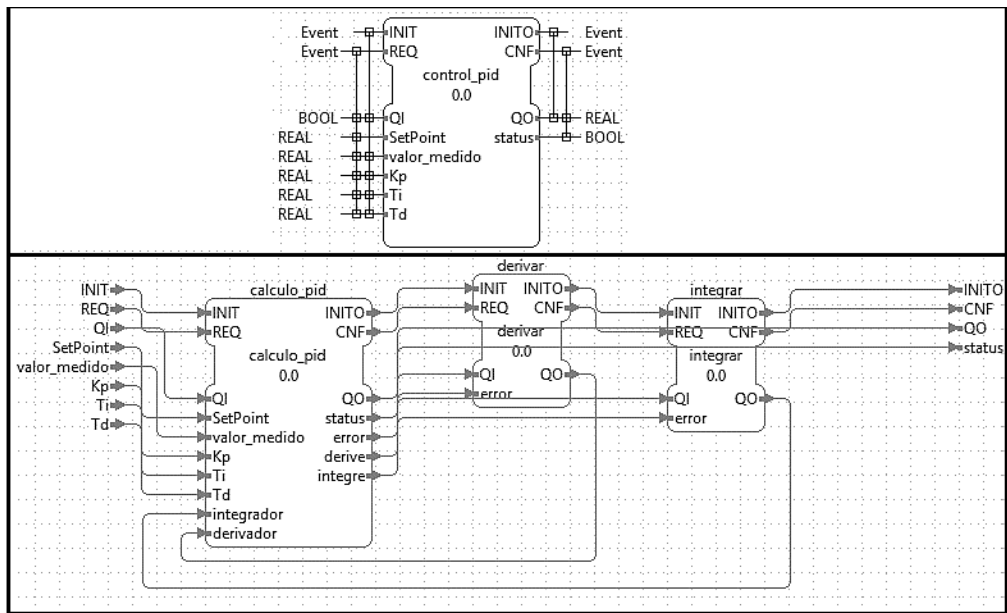


Figura 4.24: FB control_pid.

El bloque *derivar* se muestra en la figura 4.25, en el cual se realiza el cálculo de la derivada por medio del error; se cuenta con el evento de entrada INIT para inicializar el bloque y el evento REQ para realizar el cálculo de la derivada, a la entrada están el dato de QI que lee un valor TRUE para ejecutar el bloque y *error* para la lectura del error. Como dato de salida se tiene QO con el cálculo de la derivada, finalmente, los eventos de salida son INITO que indica el fin de la ejecución y CNF para indicar que el bloque ya está disponible para un nuevo cálculo.

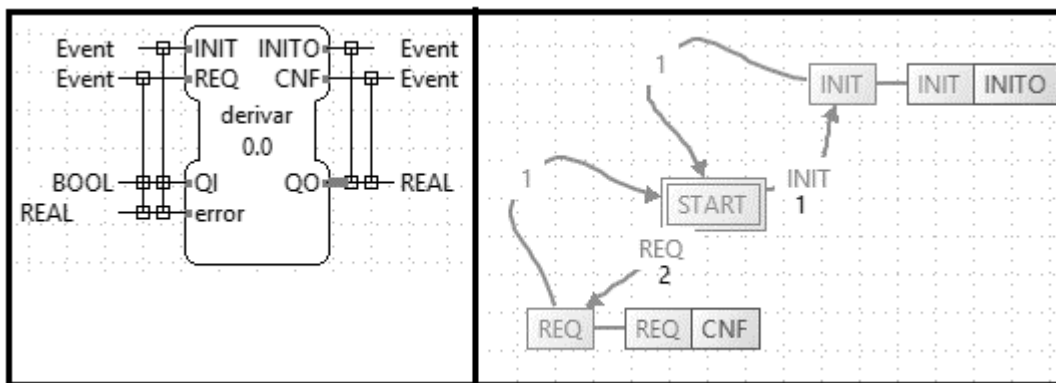


Figura 4.25: FB derivar.

En el bloque *derivar* se define una variable interna *e* a la cual se le asigna el valor del error, los algoritmos de este bloque se presentan a continuación:

Algoritmo INIT:

$e := \text{error};$

Algoritmo REQ:

$QO := (\text{error}-e)/100 \text{ /*100 es el valor del tiempo de muestreo */}$

$e := \text{error};$

El bloque *integrar* se muestra en la figura 4.26 en la que se observa la interfaz con los eventos de entrada y salida en la parte superior, así como sus variables de entrada y salida en la parte inferior y el ECC a la derecha de la figura. Se tiene el evento de entrada INIT para inicializar el bloque y el evento REQ para realizar el cálculo de la integral, a la entrada está el dato de QI que lee un valor TRUE para realizar el cálculo y *error* para la lectura del error. Como dato de salida está QO con el cálculo de la integral obtenida a partir del error; por último, los eventos de salida que son INITO indicando el fin de su ejecución y CNF que el bloque ya está disponible para un nuevo cálculo.

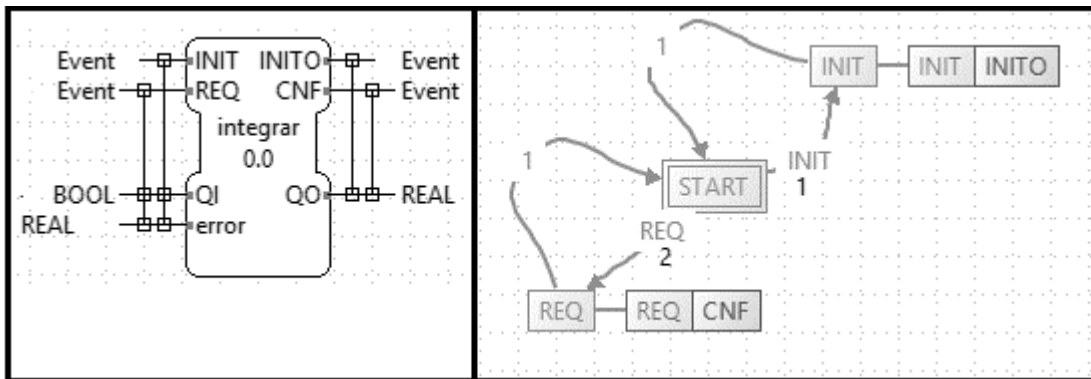


Figura 4.26: FB integrar.

Los algoritmos del bloque de la figura 4.26 son los siguientes:

Algoritmo INIT:

$QI := \text{TRUE};$

Algoritmo REQ

$QO := QO + \text{error}*100;$

Con los valores de la derivada y la integral se procede a calcular el esfuerzo de control mediante el bloque *calculo_pid* que se muestra en la figura 4.27. Se cuenta con el evento de entrada INIT para inicializar el bloque y el evento REQ para realizar el cálculo, a la entrada está el dato de QI que lee un valor TRUE para realizar el

cálculo, *SetPoint* que es el valor de referencia, *valor medido* obtenido desde los sensores del proceso, la constante proporcional K_p , la constante integral T_i , la constante derivativa T_d , el cálculo de la derivada como dato para *derivador* y el cálculo de la integral leído en la variable *integrador*. Entre los datos de salida está STATUS indicando el estado del bloque, QO con el esfuerzo de control calculado, *derive* que activa el FB *derivar* para un nuevo cálculo de la derivada e *integre* que activa el FB *integrar* para realizar la integral; finalmente, los eventos de salida que son INITO indicando el fin de su ejecución y CNF para indicar que el bloque ya está disponible para un nuevo cálculo.

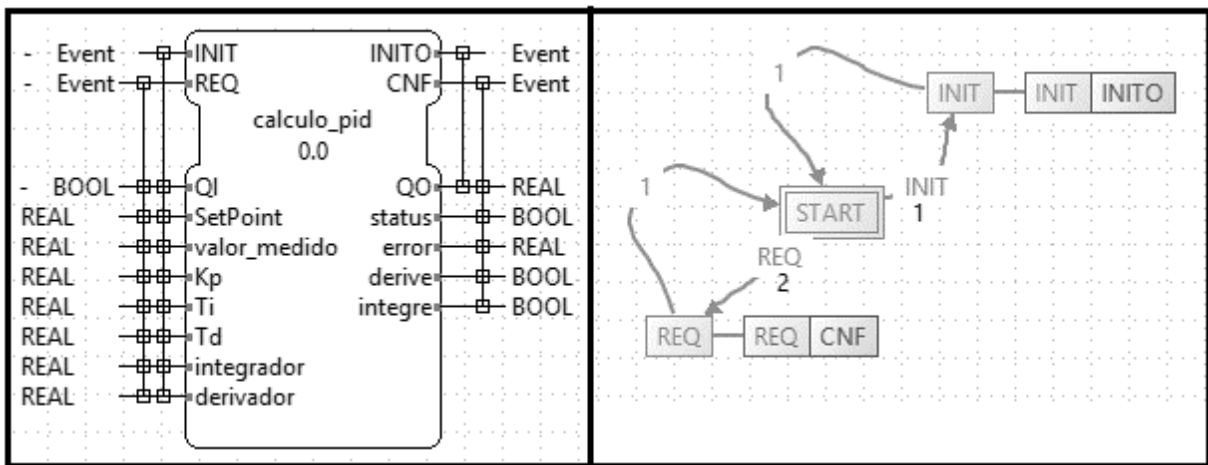


Figura 4.27: FB calculo_pid.

Los algoritmos ejecutados en el anterior FB son los siguientes:

Algoritmo INIT:

QI := **TRUE**;

Algoritmo REQ:

error := SetPoint-valor_medido;

derive := **TRUE**;

integre := **TRUE**;

QO := (error + (1/Ti)*integrador + Td*derivador) * Kp; /*tiempo de muestreo=100 */

status := **TRUE**;

4.2 Simulación del sistema

A continuación se presenta la simulación del proceso por medio del software *snoopy* para mostrar la dinámica de funcionamiento del modelo realizado en 4DIAC-IDE

traducido a redes de Petri jerárquicas debido a las similitudes que presentan ambos sistemas de modelado como es la ejecución por eventos y la capacidad de descomponer los sistemas en varias capas de complejidad, de esta forma se identifican las características de los RMS y el cumplimiento de los criterios establecidos en el capítulo 1 para la máquina caso de estudio.

Para verificar que el sistema es reconfigurable se confrontan cada una de las características de los RMS gracias a la simulación realizada en redes de Petri y apoyada por las características de IEC 61499, se escogieron ciertas partes de la simulación para ilustrar cada uno de los criterios; sin embargo, las redes que no fueron escogidas para hacer parte de este documento se encuentran en el anexo F.

4.2.1 Personalización

Esta característica se verifica con el inicio de la simulación en donde el usuario interactúa con la máquina haciendo las modificaciones a los productos que fabrica el sistema. Se va a mostrar la secuencia de ejecución del modelo desarrollado para la fabricación de una pieza para la silla1: taburete de cuatro patas, de esta forma se irá identificando la capacidad de reconfiguración del proceso. El modelo CAD del producto entra al recurso HMI que se muestra en la figura 4.28.

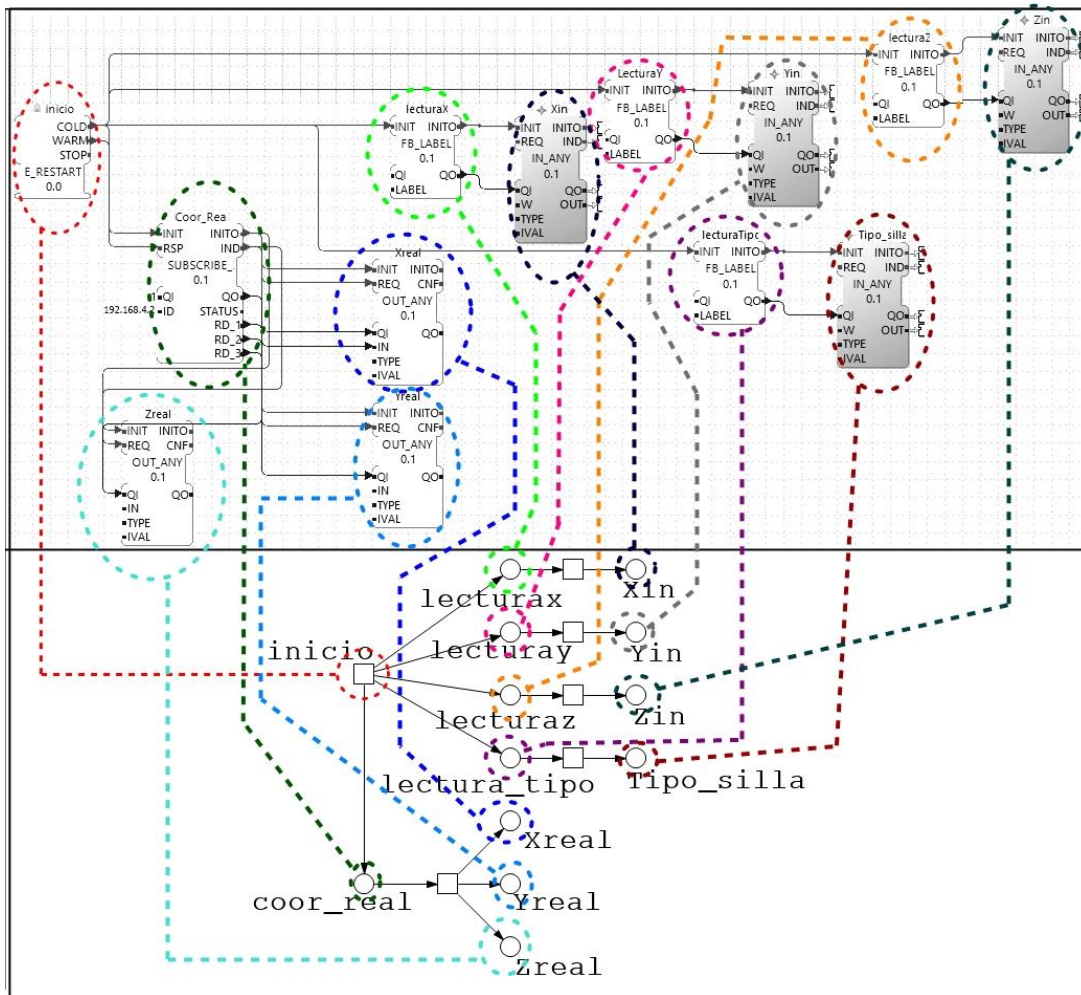


Figura 4.28: Recurso PC.HMI y su representación en redes de Petri.

Los lugares *lecturax*, *lecturay* y *lecturaz* representan los bloques que permiten la adquisición de información desde afuera del sistema, mientras que el lugar *lectura_tipo* representa el bloque que permite recibir información sobre el tipo de producto que se desea realizar, seguidamente están los bloques tipo IN_ANY, los cuales se encargan de leer las coordenadas del modelo CAD de la pieza y *Tipo_silla*, el cual lee que silla se va a fabricar, para este caso silla 1 para el taburete; estos bloques están mapeados desde la aplicación *com_mk*, por lo cual la secuencia de ejecución continua en el recurso *datos_mk*, en el cual están mapeados los otros bloques de esa aplicación. En este modelo es donde se pueden realizar las modificaciones en cuanto al acabado de los productos aportando a la personalización, por medio del modelo CAD, ya que la herramienta sigue la trayectoria que es enviada aquí para moldear la pieza de trabajo y no se requiere

conocimiento en RMS o el estándar, solo saber modificar el modelo CAD del producto; de acuerdo a esto se activan los diferentes procesos de mecanizado requeridos que se han definido en este modelo.

De esta forma se puede demostrar una buena personalización en este sistema caso de estudio.

4.2.2 Convertibilidad

Esta característica se ha definido como la capacidad por la cual el sistema es capaz de tolerar la adición de nuevos miembros a una familia de productos. Esta sucede en la aplicación de reconfiguración que agrupa la familia de productos y los procesos de mecanizado que realiza la máquina, esta aplicación ha sido mapeada en el recurso torneado se muestra en la figura 4.29.

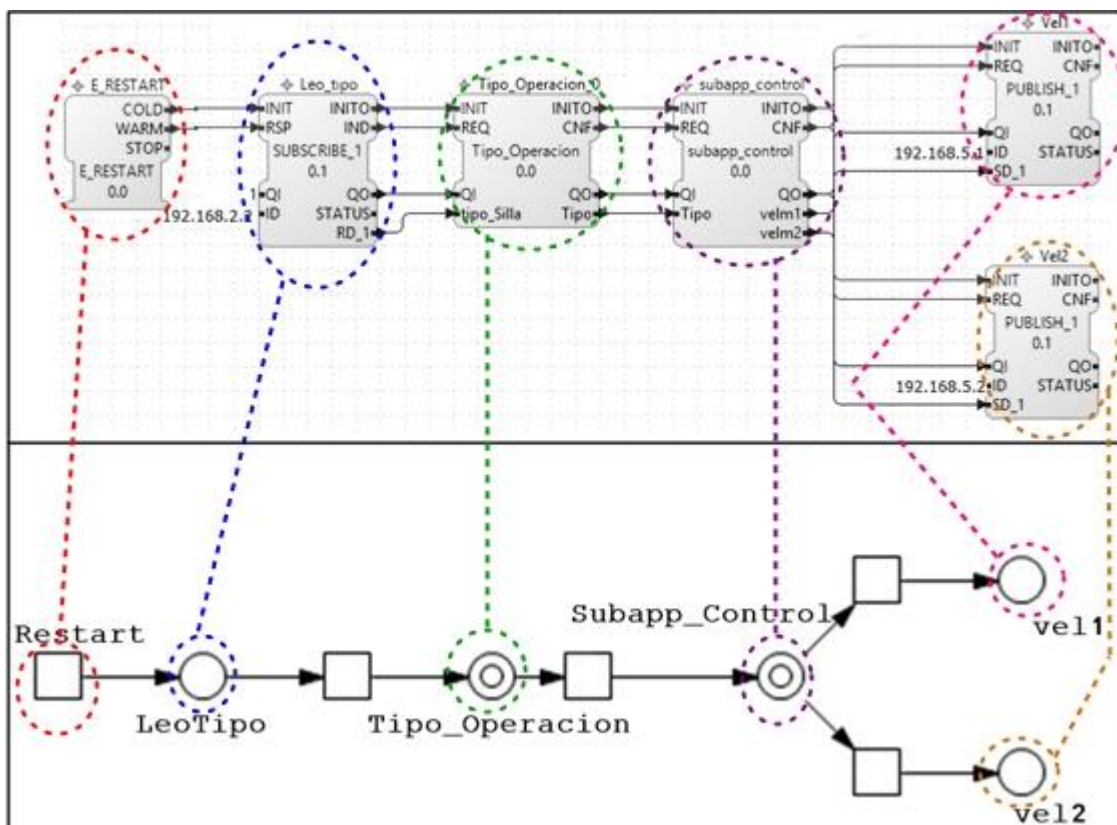


Figura 4.29: Aplicación de reconfiguración y su representación en redes de Petri.

En la figura 4.29, se muestra el lugar *LeoTipo* que representa el bloque de interfaz de servicio suscriptor que por su dirección IP (192.168.2.2) está relacionado

directamente con el bloque *Tipo_a_MK* desde el dispositivo PC, es decir que recibe la información sobre el tipo de producto que se desea realizar y es aquí donde se ingresan nuevos productos; esta información es transmitida al bloque compuesto *Tipo_Operacion* que se encarga de almacenar los nuevos productos identificando las piezas requeridas en un FB que debe ser añadido cuando se quiere agregar un nuevo producto permitiendo la convertibilidad del sistema, este bloque es representado en la red de Petri por un lugar compuesto, donde se identifican las piezas de cada producto almacenadas en los bloques *silla1* y *silla2* que se muestra en la figura 4.30.

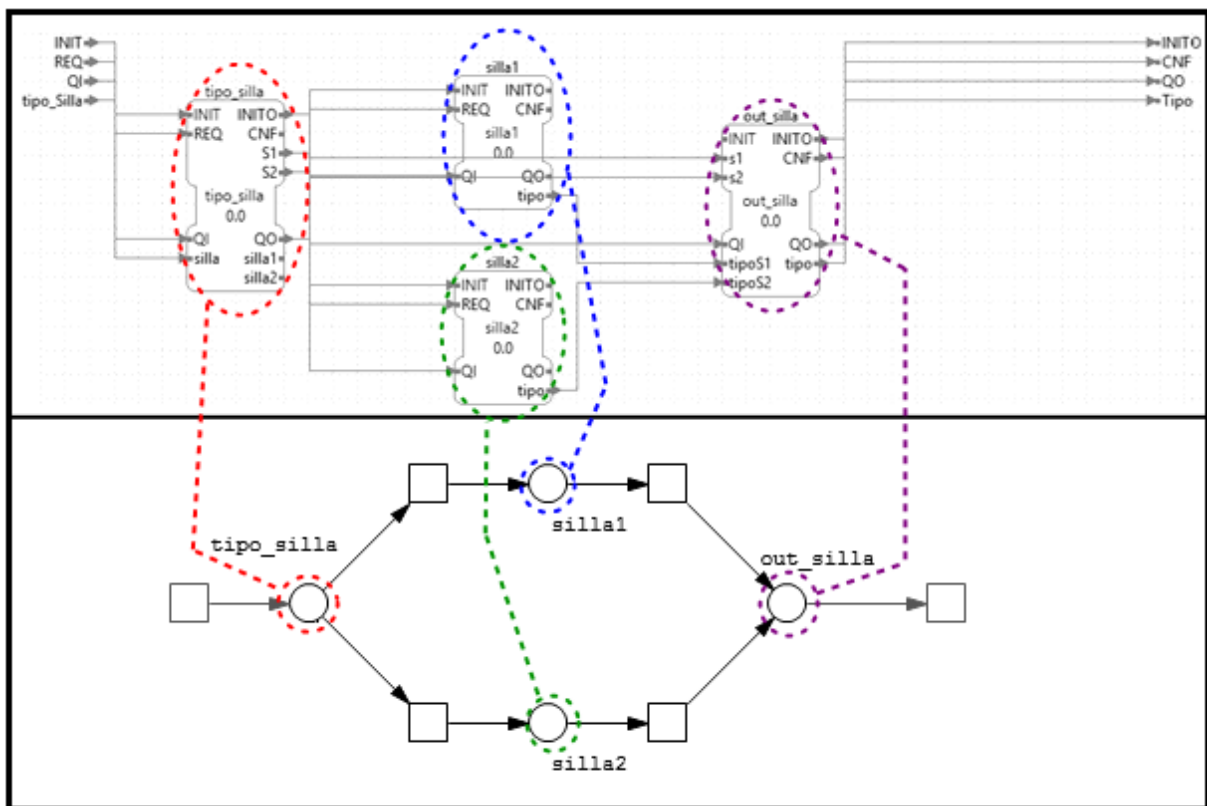


Figura 4.30: Red de FBs del bloque compuesto *Tipo_Operacion* y su representación en redes de Petri.

La subred de Petri que representa a este bloque compuesto contiene un lugar llamado *tipo_silla* el cual representa a un FB básico desarrollado para este proyecto con el fin de generar un evento diferente para cada modelo de producto, en la red de Petri se representa activando la transición de acuerdo al producto requerido, en este caso se activa el evento para la silla 1 o taburete, en la red se representa activando

la transición de la parte superior lo que activa al lugar llamado *silla1* que está representando al bloque de función que contiene la identificación de los procesos de mecanizado necesarios para la fabricación de las piezas del taburete, estos son activados mediante números enteros: 1 para taladrado, 2 para fresado y 3 para torneado; de esta forma se activan los diferentes FB con los procesos de mecanizado requeridos, esto se envía como una secuencia que pasa por el bloque *out_silla*.

Cuando se desea añadir los nuevos productos se requiere adicionar un nuevo FB con la identificación de las piezas conectado de la misma forma que los bloques *silla1* y *silla2* que en la red de Petri de la figura 4.30 sería un nuevo lugar, *tipo_silla* requiere una nueva salida *silla3* y *out_silla* una nueva entrada para enviar la activación de los distintos procesos de mecanizado que se necesitan para obtener el nuevo producto. El estándar no pone limitaciones en cuanto al tamaño de la red y la adición de un nuevo producto no aumenta considerablemente el tamaño ya que solo es un nuevo bloque y el resto del sistema es reutilizado para la ampliación de la familia de productos gracias a la programación de los diferentes FBs de forma genérica para facilitar su reutilización.

Con esto se puede verificar que la convertibilidad está trabajando de manera satisfactoria en el sistema desarrollado para este caso de estudio.

4.2.3 Modularidad

Con el desarrollo de la arquitectura del estándar para este sistema caso de estudio se programaron varias aplicaciones, subaplicaciones y bloques compuestos, mapeados en los diferentes recursos que se definieron para reducir la complejidad del sistema y se puede ver cada una de estas partes no acaparan un gran número de FB para ver más fácilmente la lógica de ejecución que puede ser reordenada. Además, los bloques programados se definieron de forma genérica para facilitar su reutilización, como ya se ha mostrado en el transcurso de la sección de simulación. La subaplicación de control que se muestra en la figura 4.31 es un ejemplo para verificar esta característica, esta hace parte de la aplicación de reconfiguración que se mostró en la figura 4.29.

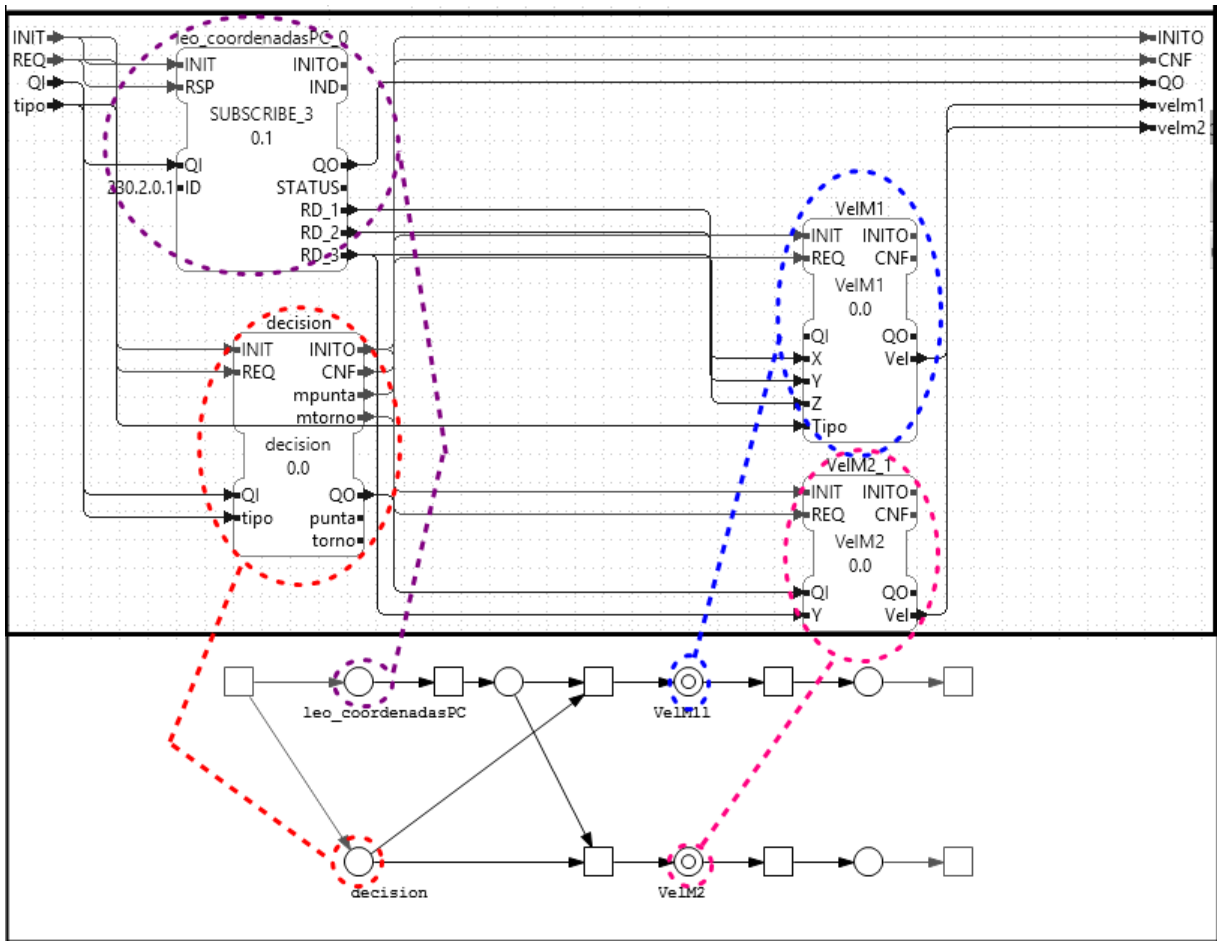


Figura 4.31: Red de FBs de la subaplicación Subapp_control y su representación en redes de Petri.

Continuando con el ejemplo de simulación, en la figura 4.31 están los procesos de mecanizado que trabajan de forma independiente con la familia de productos como lo define la modularidad de los RMS, al inicio de la subred de Petri para la subaplicación están los lugares *leo_coordenadasPC* el cual representa al bloque que recibe las coordenadas espaciales de las piezas del producto y al mismo tiempo entra en funcionamiento el lugar *decision* que permite activar los lugares compuestos *VelM1* o *VelM2* de acuerdo al motor que sea necesario para la tarea de mecanizado, para este caso el motor 1 de acuerdo a la figura 3.8, que permite observar que la pieza del asiento del taburete requiere trabajos de fresado y taladrado; la subred del bloque *VelM1* se muestra en la figura 4.32.

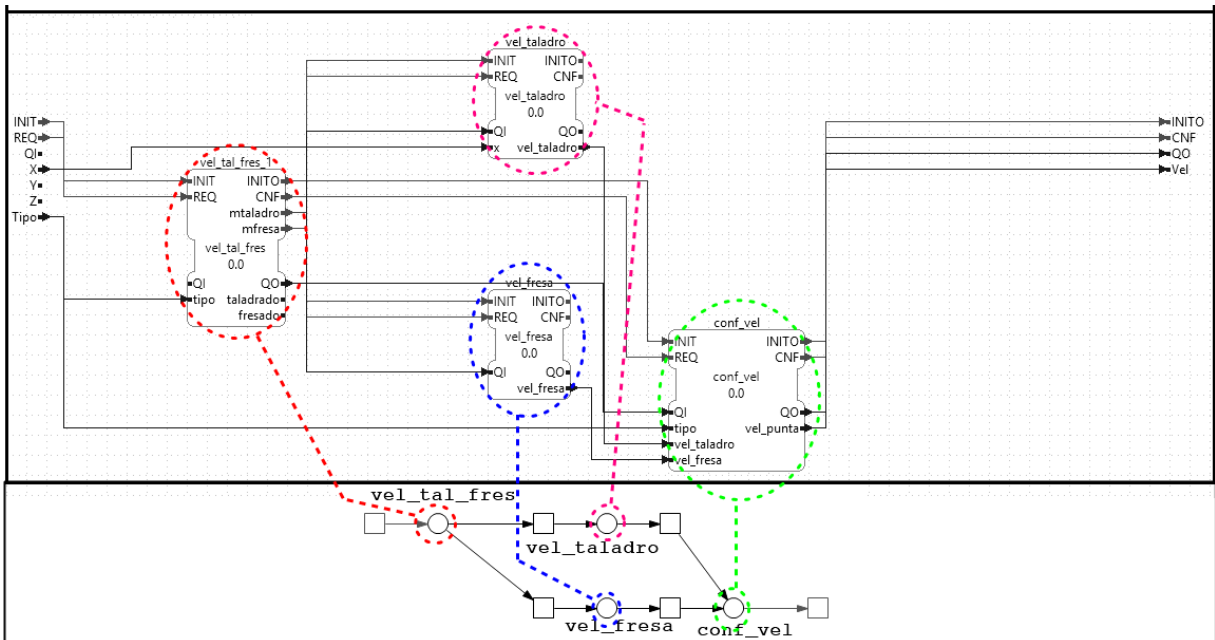


Figura 4.32: Red de FBs del bloque compuesto VelM1 y su representación en redes de Petri.

La subred de la figura 4.32 inicia con el lugar *vel_tal_fres* que representa al bloque de función básico desarrollado con el fin de activar el cálculo de cada una de las velocidades de funcionamiento, para este caso se genera el evento que activa el lugar *vel_taladrado*, este lugar representa al bloque que contiene la ecuación para el cálculo de la velocidad de taladrado que una vez calculada pasa al lugar *conf_vel* que representa al bloque que muestra a la salida del bloque este cálculo. De esta forma se ve la independencia de las diferentes operaciones de mecanizado que son activadas para trabajar sobre diferentes piezas para los productos corroborando la modularidad del sistema.

Todo esto evidencia un buen nivel de modularidad en el sistema.

4.2.4 Integridad

Esta característica será analizada desde el reordenamiento lógico del sistema desarrollado y la adición de nuevos componentes sobre este modelo. Para esto se continua con el ejemplo de simulación mostrando la figura 4.33 con el recurso *TORNEADO*, el cual contiene la fracción que hace el control del motor 1 desde la aplicación de control de la punta.

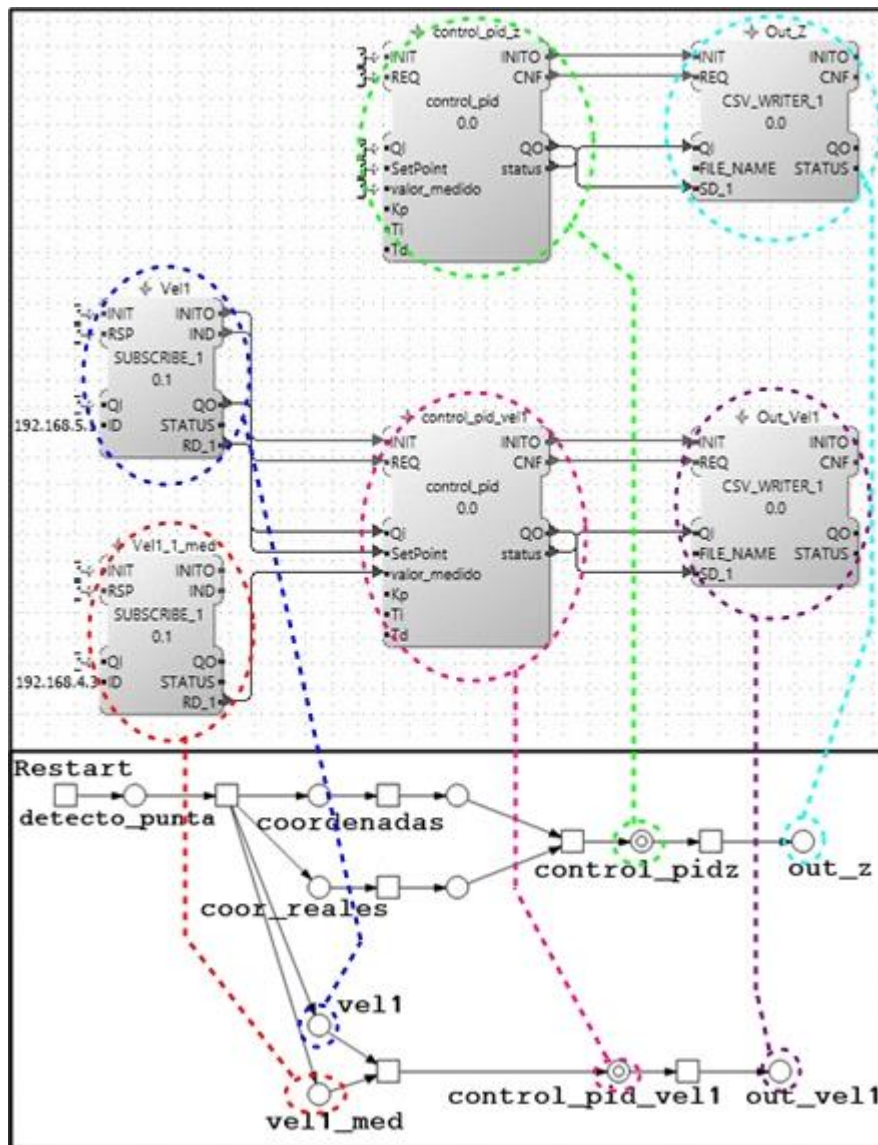


Figura 4.33: Recurso MAQUINA.TORNEADO y su representación en redes de Petri.

La red de Petri de la figura 4.33 inicia siempre con el lugar *detecto_punta*, este lugar no se ve reflejado en el modelo en 4DIAC debido a que está distribuido en otro recurso; sin embargo, se decidió representarlo siempre en el modelo de redes de Petri para mantenerse lo más fielmente posible a la ejecución real del modelo de IEC 61499, este lugar representa un sensor que lee las diferentes herramientas o puntas de trabajo limitadas por la capacidad del magazín para almacenarlas en la máquina y serán añadidas de acuerdo a los tipos de procesos de mecanizado que realmente se necesiten limitando la capacidad de la máquina para la familia de productos; la adición del magazín permite la sensibilidad a cambios en su estructura para añadir nuevas herramientas que pueden ser usadas gracias al desarrollo de bloques

genéricos que permiten la reutilización en varios procesos de mecanizado, estos pueden ser copiados y reordenados aportando a la integrabilidad del sistema. Por lo tanto se concluye que la integrabilidad del sistema desarrollado es buena.

4.2.5 Diagnosticabilidad

Al final de la figura 4.28, mostrada al inicio de la evaluación de la característica de personalización, está representado el bloque de interfaz de servicio *coor_real*, el cual recibe las mediciones desde la máquina para que luego sean impresos en la pantalla de este dispositivo a través de los bloques tipo OUT_ANY, de esta forma el usuario logra visualizar las dimensiones reales de la pieza durante el proceso de mecanizado y supervisar la fabricación del producto para que sea acorde al modelo CAD, esto contribuye a la diagnosticabilidad del proceso.

El lugar *vel1* representa al bloque de interfaz de servicio que recibe la velocidad calculada desde la aplicación de reconfiguración (figura 4.20), y el bloque *vel1_med* que recibe la velocidad medida desde los sensores de la máquina los cuales pasan al lugar compuesto *control_pid_vel1* que representa al bloque compuesto que contiene el controlador PID para esta variable; estos valores son recibidos por sensores que son leídos mediante bloques READER distribuidos en todo el sistema aportando la detección de fallas; continuado con la simulación se muestra control PID para el motor 1 en la figura 4.34.

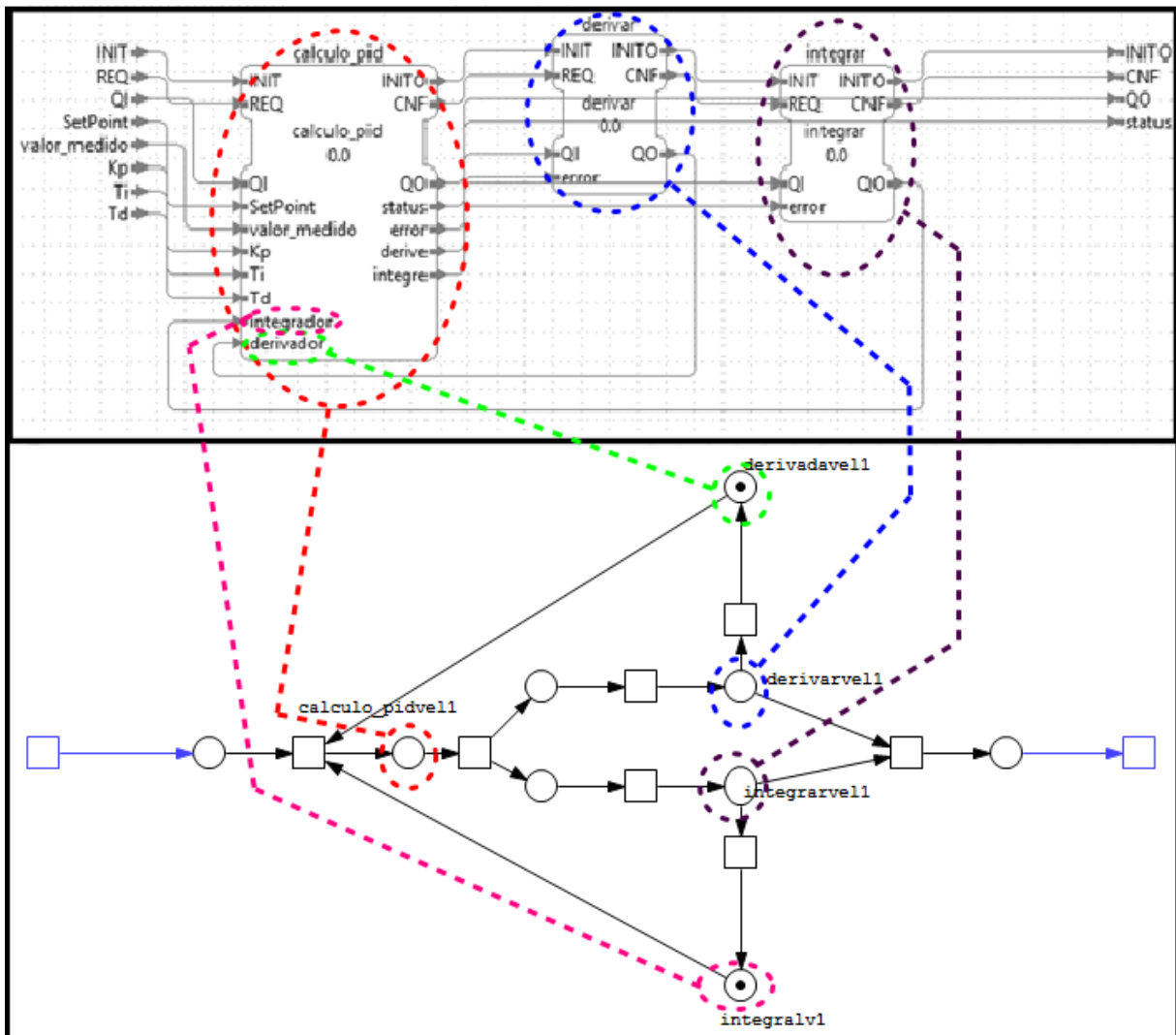


Figura 4.34: Red de FBs del bloque compuesto Control_pid_vel1 y su representación en redes de Petri.

La figura 4.34 verifica la detección de fallas y muestra la toma acciones correctivas para evitar daños en la pieza de trabajo asegurando que el motor trabaje con la velocidad adecuada en forma paralela con la aplicación de reconfiguración de tal forma que se controle la velocidad requerida de acuerdo al proceso de mecanizado necesario y aportando a la integración de los distintos módulos programados en este sistema (FB), para esto se cuenta con el lugar llamado *calculo_pidvel1* el cual es una instancia del bloque básico de control PID de 4DIAC, de este lugar salen dos eventos para activar los lugares *integrarvell1* y *derivarvell1*, los cuales representan a los bloques de función desarrollados con el fin de derivar e integrar el error que es calculado en el bloque PID con la información recibida desde el modelo de la figura

4.33, los lugares *integralv1* y *derivadavel1* representan las integrales y derivadas necesarias para que el bloque PID calcule el esfuerzo de control que pasa al lugar *out_vel1* de la figura 4.33, que representa a un bloque CS_WRITER que se encarga de imprimir ese esfuerzo de control en las salidas físicas de la máquina haciendo así un control efectivo a la velocidad de la punta que hace la tarea de mecanizado.

Sin embargo, en este trabajo no se ha desarrollado un generador de reportes de errores y dado que el estándar pone restricciones en cuanto al acceso a las variables internas y los algoritmos dentro del bloque no es posible identificar un error dentro de estos, no se puede conocer si los eventos están siendo activados correctamente.

Con esto se puede concluir que la diagnosticabilidad no se realiza eficientemente en este sistema.

4.2.6 Escalabilidad

Desde el punto de vista de la arquitectura desarrollada no hay restricciones a los volúmenes de producción y la ejecución por eventos da la flexibilidad de que cada proceso de mecanizado se ejecute de acuerdo a las condiciones externas. Sin embargo, la ejecución por eventos no se realiza por tiempos programados sino por la ocurrencia de estos, lo único programado es la velocidad de operación de los motores; en la figura 4.32 que muestra la activación de los procesos de mecanizado para el motor 1 y la velocidad de este, se puede observar que las fórmulas dependen de algunas variables como las dimensiones de las piezas, por lo que no es posible que el sistema regule conscientemente sus velocidades de producción, impidiendo la optimización de las secuencias de ejecución para mejorar los tiempos de operación. Esta característica queda dependiente de las capacidades físicas de la máquina que no fueron estudiadas en este trabajo.

Se interpreta que la escalabilidad del sistema es baja debido a la incapacidad del modelo desarrollado para modificar los tiempos de operación de la máquina.

4.3 Resultados

En esta sección se presentarán los resultados finales del proyecto que van a complementar la simulación anterior, redondeando los conceptos y mostrando las ideas finales de este trabajo.

4.3.1 Adaptabilidad de la capacidad productiva de la máquina

Como ya se observó en la sección de simulación, el sistema puede ser convertible ya que se pueden añadir nuevos procesos de mecanizado y productos debido a la modularidad para trabajar como unidades independientes, gracias a los bloques de interfaz de servicio se integra todo el sistema y llegan los distintos eventos activar los FB requeridos, de esta forma también pueden personalizarse los productos ya que la independencia de autonomía de los FB permite la reutilización para obtener diferentes modificaciones sobre las piezas de los productos que se elaboran.

4.3.2 Desarrollo de un sistema para una familia de productos

La familia de productos es almacenada en el bloque *Tipo_Operacion*, aquí se almacenan las diferentes piezas que se requieren en cada silla. El diseño de la aplicación de reconfiguración se ha realizado de forma modular utilizando subaplicaciones, bloques compuestos y dividiendo los diferentes productos y procesos de mecanizado para que trabajen de forma independiente y así facilitar la ejecución de los FB para trabajar de diferentes formas, de esta forma es más práctica la personalización del sistema para reordenarlo y modificar los productos, activando eventos para realizar diferentes procesos de mecanizado.

4.3.3 Sensibilidad a cambios en su estructura

Como se puede observar en los diferentes modelos desarrollados aplicando la arquitectura del estándar, los bloques programados son genéricos para facilitar el reordenamiento de estos y ser reutilizados de tal forma que se facilite la personalización y la integrabilidad del sistema. Por otro lado, la variación de los volúmenes de producción no es posible con este modelo ya que la ejecución por eventos hace que el sistema funcione por la ocurrencia de estos más no de tiempos programados y dificulta la optimización de las secuencias de ejecución, esto depende de factores externos para que la máquina cuente con las herramientas y las materias

primas a tiempo para trabajar adecuadamente, esto supone falencias en la escalabilidad del sistema.

4.3.4 Detección de fallas en el proceso de reconfiguración

El proceso de control que se ha modelado en este trabajo cuenta con sensores que envían la información al proceso de control mediante los bloques tipo READER para detectar fallas. Además, los FB solo se ejecutan cuando los algoritmos corren de forma correcta, de lo contrario se envía una salida que indica que ocurrió una falla dentro del FB aportando a la diagnosticabilidad de los RMS.

Este trabajo no consideró el reporte de errores y por las restricciones del estándar no es posible acceder a la información interna de un FB por lo que solo se puede conocer si el sistema está trabajando correctamente pero no se sabe que ocurre internamente en caso de una falla en el sistema, en este aspecto se debe asegurar una correcta programación de las variables internas para evitar paradas indeseadas. Para esto la modularidad aporta a una mejor comprensión del desarrollo del modelo evitando la complejidad de este y dividiendo el sistema en diferentes componentes que facilitan su entendimiento y programación.

De esta forma se puede concluir que la máquina caso de estudio fue diseñada para trabajar con la capacidad y funcionalidad exactamente necesaria, se pudo observar que se pueden añadir nuevos productos gracias a la independencia de los componentes por medio de la modularidad al igual que nuevos procesos de mecanizado, cumpliendo en buena manera con los objetivos de una RMT y adquiriendo un sistema con un buen nivel de reconfigurabilidad, reflejando las características de los RMS y el seguimiento de los criterios que fueron establecidos en este trabajo.

Capítulo 5: Conclusiones y trabajos futuros

5.1 CONCLUSIONES

Después del desarrollo de este proyecto se pudo concluir que:

- El establecimiento de los criterios necesarios para la tipificación de un RMS permitieron relacionar las características propias de este tipo de sistemas con los objetivos para una RMT.
- Se evaluaron varias opciones como arquitecturas y softwares de modelado llegando a la conclusión de que el estándar IEC 61499 cuenta con el respaldo teórico y las características suficientes para darle la formalidad requerida al modelo desarrollado, y con este, 4DIAC como herramienta computacional debido a sus facilidades en temas de reconfigurabilidad y manejo limpio de los lineamientos propuestos por el estándar.
- Durante el modelado del sistema de manufactura caso de estudio, se confirmó la necesidad de la maximización del uso de las herramientas del estándar como las subaplicaciones y bloques compuestos, con el propósito de simplificar las redes de FBs que de otro modo serían muy complejas, debido a las tareas de reconfiguración.
- Debido a la inmadurez de estas tecnologías, se encontró que no es posible realizar una simulación desde una herramienta especializada en IEC 61499 con los recursos que dispone este proyecto.
- Gracias a la flexibilidad del estándar, fue posible el desarrollo de bloques básicos y compuestos propios que hicieran funciones que no están contempladas dentro de este; que junto con las herramientas descritas en la conclusión anterior,

componen una caracterización del estándar para el caso concreto de su aplicación en los RMS.

- Se presentaron algunos problemas en cuanto a la estabilidad del software de modelado, ya que, según la explicación de uno de sus desarrolladores, este tipo de herramientas aún están en edad temprana y necesitan bastante trabajo para alcanzar la madurez necesaria para implementaciones mayores.
- La simulación a través de redes de Petri permitió ilustrar a través de una herramienta conocida el funcionamiento del modelo en caso de una implementación real, esto facilitó el entendimiento de la dinámica del estándar y la identificación del cumplimiento de las características de un RMS en el caso de estudio.
- El estándar IEC 61499, debido a su enfoque distribuido aporta varias ventajas que coinciden con los requerimientos de los RMS como la convertibilidad, personalización, modularidad e integrabilidad.
- El estándar también impuso ciertas restricciones a requerimientos como la escalabilidad y diagnosticabilidad debido a características propias del mismo entre las que se incluyen, la ejecución basada en eventos y la imposibilidad de tener acceso a la información interna de los FB.
- Con base en las características evaluadas, su alto rendimiento en 4 de ellos y bajo en los otros dos, se puede concluir que el modelo desarrollado cumple de buena manera con los principios de reconfigurabilidad para una RMT propuestos por Yoram Koren, es decir, que el sistema es en buena medida reconfigurable.

5.2 TRABAJOS FUTUROS

Teniendo en cuenta la anterior investigación, se puede dar continuidad abordando los siguientes temas:

- Desarrollar el modelo mecánico de la máquina RMT propuesta.
- Realizar el modelado de la interacción directa entre el modelo aquí desarrollado con los componentes físicos de la máquina RMT.

- Proponer el modelado de la interacción directa entre el modelo aquí desarrollado con los componentes software del PC, es decir, la HMI y la comunicación con la herramienta CAD.
- Desarrollar criterios de selección estandarizados para la arquitectura en la que se puede apoyar un RMS.
- Realizar una implementación del sistema planteado en un dispositivo de control embebido.
- Modelar otros tipos de procesos de manufactura (Batch, continuos), utilizando la arquitectura propuesta por el estándar IEC 61499 y aplicando los criterios de los RMS.

BIBLIOGRAFÍA

- [1] Z. M. Bi, S. Y. T. Lang, W. Sheng y L. Wang, “*RECONFIGURABLE MANUFACTURING SYSTEMS: STATE OF THE ART*”, Integrated Manufacturing Technologies Institute, National Research Council, London, ON, Canada, International Journal of Production Research, Vol. 46, No. 4, 967-992, 15 de febrero de 2008.
- [2] Y. Koren, M. Shpitalni, “Design of reconfigurable manufacturing systems”, The University of Michigan, Ann Arbor, USA, Israel Institute of Technology, Haifa, Israel, enero de 2011.
- [3] Y. Koren, U. Heisel, F. Jovane, T. Moriwaki, G. Pritschow, G. Ulsoy, H. Van Brussel, “*RECONFIGURABLE MANUFACTURING SYSTEMS*”, Universidad de Michigan, EE.UU; Universidad de Stuttgart, Alemania, Politécnico de Milán, Italia, Universidad de Kobe, Japón, Universidad Católica de Lovaina, Bélgica, 1999.
- [4] D. Anderson, “Implementing BTO&MC”, en *Built-to-Order & Mass Customization*, Ed. CIM Press, Cambria, California, USA, 2004, pp. 61-69.
- [5] Y. Koren, “*General RMS Characteristics. Comparison with Dedicated and Flexible Systems*”, in “*Reconfigurable Manufacturing Systems and Transformable Factories*”, A. Dashchenko, Springer-Verlag Berlin Heidelberg: Estados Unidos, 2006, pp. 27-43.
- [6] J. Valtierra-Olivares y J. Sausedo-Solorio, “*Reconfiguración autónoma de sistemas de manufactura mediante la optimización de funciones de desempeño del proceso*”, Twelfth LACCEI Latin American and Caribbean Conference for Engineering and Technology (LACCEI'2014), “Excellence In Engineering To Enhance a Country's Productivity” Guayaquil, Ecuador July, 22-24, 2014.

- [7] A. Farid, "*Reconfigurability Measurement in Automated Manufacturing Systems*", University of Cambridge, Cambridge, England, noviembre de 2007.
- [8] R. Lewis, "Modelling Control Systems Using IEC 61499", en *IET Control engineering series 59*, The Institution of Engineering and Technology, pp: 1-41, Londres, Reino Unido, 2008.
- [9] H. Shivanand, M. Benal, V. Koti, "FMS Introduction and Description", en *Flexible Manufacturing System*, Ed. New Age International Pvt Ltd, India, Diciembre de 2006, pp. 2-18.
- [10] V. Malhotra, T. Raj, A. Arora, "*Reconfigurable manufacturing system: an overview*", Department of Mechanical Engineering, YMCA Institute of Engineering, Faridabad, India, 2009.
- [11] J. Sánchez, M. Esteve, E. Rueda, M. Vicente, "*INVESTIGACIÓN SOBRE EL DISEÑO DE SISTEMAS DE FABRICACIÓN RECONFIGURABLES. Aplicación de los Sistemas Reconfigurables en el Sector del Mueble*", AIDIMA Instituto Tecnológico del Mueble, Madera, Embalaje y Afines, Valencia, España, diciembre de 2011.
- [12] Y. Koren, G. Ulsoy, "*Vision, Principles and Impact of Reconfigurable Manufacturing Systems*", "*Powertrain International*", Vol. 5, Number 3, pp. 14-21, 2002.
- [13] Y. Koren, "*Reconfigurable Manufacturing Systems*", in *The Global Manufacturing Revolution*, A. Sague. Ed. Hoboken: New Jersey, John Wiley & Sons, Inc., 2010, pp. 227-252.
- [14] R. Johnstone, J. Kurtzhaltz, "Flexible Manufacturing System", U.S. Patent 4472783 A, Septiembre, 18, 1984.
- [15] H. ElMaraghy, "*Flexible and reconfigurable manufacturing systems paradigms*", *International Journal of Flexible Manufacturing Systems*, 2006.

- [16] J. Bollinger, “*Visionary Manufacturing Challenges for 2020*”, National Research Council Report, National Academy Press, Washington, D.C., 1998.
- [17] R. Priego, U. Gangoiti, D. Orive y M. Marcos, “*Reconfiguración de sistemas de control basado en multiagentes*”, in *CEA-IFAC: Actas de las XXXV Jornadas de Automática*, vol. 35, no. 66, pp. 1-6, septiembre, 2014.
- [18] Jin-Lung Chirn and D. C. McFarlane, "A holonic component-based approach to reconfigurable manufacturing control architecture," in *Proceedings 11th International Workshop on Database and Expert Systems Applications, IEEE*, pp. 219-223, septiembre, 2000.
- [19] Comisión Electrónica Internacional, IEC 61499-1, 2012.
- [20] W. Covanich, D McFarlane, A.M. Farid, “*Guidelines for evaluating the ease of reconfiguration of manufacturing systems in Industrial Informatics*”, in *Industrial Informatics*, vol. 13, no. 16, pp.1214-1219, Julio, 2008.
- [21] A. Farid, “*Reconfigurability Measurement in Automated Manufacturing Systems*”, tesis doctoral, departamento de ingeniería, Universidad de Cambridge, Cambridge, R.U., 2007.
- [22] V. Vyatkin, “*The IEC61499 standard and its semantics*”, in *Industrial Electronics Magazine, IEEE*, vol.3, no.4, pp.40-48, Diciembre, 2009.
- [23] E. Querol, J. Romero, A. Estruch, F. Romero, “*NORMA IEC-61499 PARA EL CONTROL DISTRIBUIDO APLICACIÓN AL CNC*”, in *CEA-IFAC: Actas de las XXXV Jornadas de Automática*, vol. 35, no. 108, pp. 1-8, septiembre, 2014.
- [24] F. Gonzáles, M. García, “*Implementación de Sistemas Empotrados de Control Distribuidos bajo el Estándar IEC-61499*”, tesis de máster, Escuela Técnica Superior de Ingeniería, Universidad del País Vasco, España, 2013.

[25] R. Landers¹, B. Min², Y. Koren², “*Reconfigurable Machine Tools*”, University of Missouri¹, University of Michigan², Department of Mechanical Engineering, Rolla, Missouri¹, Ann Arbor, Michigan², 2001.

[26] Y. Koren, “*Reconfigurable Machines*”, in *The Global Manufacturing Revolution*, A. Sague. Ed. Hoboken: New Jersey, John Wiley & Sons, Inc., 2010, pp. 205-226.

[27] Y. Koren, A. Ulsoy, “*RECONFIGURABLE MANUFACTURING SYSTEMS*”, Engineering Research Center for Reconfigurable Machining Systems (ERC/RMS) University of Michigan, Ann Arbor, Michigan, septiembre de 1997.

[28] C. Yang, “Design and Validation Environment for IEC 61499 Function Blocks”, Tesis doctoral, Dep. de Ciencia de la Ingeniería, La Universidad de Auckland, Auckland, Nueva Zelanda, 2010.

[29] *Guía para Diseño, Construcción e Interpretación de Indicadores*, Edición 2, DANE, Bogotá D.C., 2013.

[30] N. Choski, D. MacFarlane, “*Reconfigurable Process Control Research*”, en *A distributed coordination approach to reconfigurable process control*, Springer-Verlag, Londres, 2008, pp. 15 - 40.

[31] I. Peña, C. Córcoles, C. Casado, “*El profesor 2.0: docencia e investigación desde la Red*”, “*UOC Papers: revista sobre la sociedad del conocimiento*”, ISSN-e 1885-1541, N^o. 3, 2006.

[32] L. Cruz, O. Rojas, "The future of industrial automation and IEC 61499-3 standard," in *Engineering Mechatronics and Automation (CIIMA)*, 2014 III International Congress of Engineering Mechatronics and Automation, vol. 3, no.10, pp.42-47, 22-24 Oct. 2014.

[33] A. Zoitl, H. Prähofer, “Guidelines and Patterns for Building Hierarchical Automation Solutions in the IEC 61499 Modeling Language”, *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2387-2396, noviembre, 2013.

[34] T. Strasser, M. Rooker, G. Ebenhofer y A. Zoitl, "Standardized Dynamic Reconfiguration of Control Applications in Industrial Systems" in International Journal of Applied Industrial Engineering, vol. 2, nº 1, pp. 57-73, 2014.

[35] A. zoitl, "*Reconfiguration Approach*". In *Real-Time Execution for IEC 61499*, ISA: United States of America, 2009, pp. 51-89.

[36] Z. M. Bi, S. Y. T. Lang, W. Sheng y L. Wang, "*RECONFIGURABLE MANUFACTURING SYSTEMS: STATE OF THE ART*", Integrated Manufacturing Technologies Institute, National Research Council, London, ON, Canada, International Journal of Production Research, Vol. 46, No. 4, 967-992, 15 de febrero de 2008.

[37] M. Groover, "*OPERACIONES DE MAQUINADO Y MÁQUINAS HERRAMIENTA*", en *FUNDAMENTOS DE MANUFACTURA MODERNA, Materiales, Procesos y Sistemas*, McGRAW-HILL/INTERAMERICANA EDITORES, S.A. DE C.V., 3ª ed.: Naucalpan de Juárez, México, 1997, pp. 505-536.

[38] M. Groover, "*TEORÍA DE MAQUINADO DE METALES*", en *FUNDAMENTOS DE MANUFACTURA MODERNA, Materiales, Procesos y Sistemas*, McGRAW-HILL/INTERAMERICANA EDITORES, S.A. DE C.V., 3ª ed.: Naucalpan de Juárez, México, 1997, pp. 481-504.

[39] S. Kalpakjian, "*Procesos de maquinado utilizados para producir diferentes formas: fresado, brochado, aserrado y limado; manufactura de engranes*", en *MANUFACTURA, INGENIERÍA Y TECNOLOGÍA*, Pearson Educación: México, 2008, pp. 723–759.

[40] A. White, Let's build something [Online]. Alaska, Estados Unidos: 2016. Disponible en: <http://www.ana-white.com/>

[41] T. Strasser, I. Mtiller, M. Schtipany, G. Ebenhofer, R. Mungenast, C. Stinder, A. Zoitl, O. Hummer, S. Thomas, H. Steininger, "*An advanced engineering environment for distributed & reconfigurable industrial automation & control systems based on IEC*

61499”, *Intelligent Production Machines and Systems*, vol. 1, no. 79 pp. 493-498, julio, 2006.

[42] H. ElMaraghy “*Changing and evolving products and Systems-Models and enablers*”, in *Changeable and reconfigurable manufacturing systems*, H. ElMaraghy: Springer Verlag London Limited, London, 2009, pp. 25-45.

[43] T. Besák, “*METHODOLOGY VERIFICATION*”, in *Usage of IEC 61131 and IEC 61499 Standards for Creating Distributed Control Systems*, Ilmenau: Biblioteca de la Universidad de Ilmenau, Alemania, 2012, pp. 63-100.

[44] Y. Koren, “*Customized, Personalized and Reconfigurable Products*”, in *The Global Manufacturing Revolution*, John Wiley & Sons, Inc.: Hoboken, New Jersey, 2010, pp. 72-102.

[45] Disermeg E.I.R.L. [Online]. Lima, Perú: 2016. Disponible en: <http://disermimeg-peru.com/index.html>