



UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE HIDALGO
ESCUELA SUPERIOR DE CD. SAHAGÚN



MODELO DE COMUNICACIÓN Y CONTROL ENTRE UN SISTEMA
FÍSICO Y UN EMULADOR VIRTUAL

T E S I S
QUE PARA OBTENER EL TÍTULO DE
LICENCIADO EN INGENIERÍA INDUSTRIAL

PRESENTA
DANIEL ROBLES CARRASCO

ASESOR
DR. ISAÍAS SIMÓN MARMOLEJO

Septiembre 2019

Agradecimientos

A la UAEH donde me he formado, me han apoyado y he conocido personas que han impactado en mi vida (profesional y académica).

Al Dr. Isaías Simón Marmolejo por su paciencia, motivación, apoyo, tiempo y conocimientos para el desarrollo de esta tesis.

A mis padres por todo el apoyo y amor durante estos años, sin ustedes no hubiera sido posible. Por estar en las buenas y en las malas, siempre con un consejo y palabras de aliento.

Resumen

Los sistemas de manufactura modernos requieren de la adaptación constante de las nuevas tendencias, buscan una mayor flexibilidad y agilidad ante perturbaciones, necesidades o crecimiento de la demanda, por lo que desarrollar y proponer arquitecturas que integren al total de sus elementos y variables es el siguiente paso para el cambio de la industria.

El problema después de crear un Sistema Holónico de Manufactura (HMS, por sus siglas en inglés) según se observa en la literatura, es crear estrategias de control en tiempo real basadas en eventos y procesamiento de información distribuida que actúen sobre los sistemas de producción flexibles. A ello, y en beneficio de aportar conocimiento valido en la novedad del tema Industria 4.0. El presente trabajo aborda el problema de desarrollar Sistemas de Manufactura Holónicos, considerando el tema de comunicación e interacción de los componentes de arquitecturas propuestas en la literatura.

Visto desde la perspectiva anterior, se utilizan un sistema físico para controlar una celda de manufactura flexible (virtual). Para ello, se propone un modelo que de manera general incluye los siguientes componentes: Un sistema físico, un controlador lógico programable, un módulo para la comunicación y el software para la simulación de los sistemas de manufactura.

La propuesta ejemplifica la solución a la comunicación entre dos sistemas con objetivos diferentes. Los resultados de las diferentes pruebas muestran que es posible avanzar a un mayor grado de automatización y control sobre los sistemas de manufactura que implique inteligencia artificial, con respecto a la forma en que las industrias trabajan actualmente.

Abstract

Modern manufacturing systems require the constant adaptation of new trends, seek greater flexibility and agility in the face of disturbances, needs or demand growth, so developing and proposing architectures that integrate the total of its elements and variables is as follows step to change the industry.

The problem after creating a Holonic Manufacturing System (HMS), as seen in the literature, is to create real-time control strategies based on events and distributed information processing that act on flexible production systems. . To this, and for the benefit of providing valid knowledge in the novelty of the Industry 4.0 theme, this work is a basis for the development of the Holonic Manufacturing Systems, addressing the issue of communication and interaction of the architectural components proposed in the literature .

Viewed from the previous perspective, a physical system is used to control a flexible (virtual) manufacturing cell. To this end, a model is proposed that generally includes the following components: A physical system, a programmable logic controller, a module for communication and software for the simulation of manufacturing systems.

The proposal exemplifies the solution to communication between two systems with different objectives. The results of the different tests show that it is possible to advance to a greater degree of automation and control over manufacturing systems that involve artificial intelligence, with respect to the way in which industries currently work.

INDICE DE CONTENIDO GENERAL

Resumen	I
Abstract	II
CAPÍTULO 1. CONTEXTO DE LA INVESTIGACIÓN	1
Introducción	2
Organización del Documento.....	3
1.1 Planteamiento del Problema	4
1.2 Solución Propuesta	5
1.3 Justificación	5
1.4 Objetivos.....	6
1.4.1 Objetivo General	6
1.4.2 Objetivos Específicos	6
1.5 Aportaciones	6
1.6 Metodología de Investigación	7
CAPÍTULO 2. MARCO TEÓRICO Y CONCEPTOS GENERALES	9
Introducción	10
2.1 Sistemas de Producción	10
2.1.1 Sistema de Manufactura Tradicional.....	10
2.2 Sistemas Holónicos de Manufactura	13
2.3 Inteligencia Artificial.....	15
2.3.1 Inteligencia Artificial Distribuida	15
2.3.2 Sistemas de Manufactura Basados en Agentes Inteligentes	16
2.4 Agentes Inteligentes	17

2.4.1 Herramientas para su Desarrollo	20
2.4.2 JADE	21
2.4.3 FIPA	22
2.5 Sistemas Cíber-Físicos	22
2.5.1 Gemelos Digitales.....	24
2.6 Comunicación	26
2.6.1 Los Protocolos	26
2.6.2 El Protocolo TCP	26
2.6.3 Redes de Comunicación en la Industria	27
2.7 Sistemas de Eventos Discretos.....	33
2.7.1 Simulación de Eventos Discretos.....	33
2.7.2 Software FlexSim	34
2.8 Controlador Lógico Programable	35
CAPÍTULO 3. METODOLOGÍA Y CONFIGURACIÓN	37
3.1 Metodología: Arquitectura de Dos Capas.....	38
3.2 Propuesta de Investigación	41
3.3 Componentes del Proyecto	41
3.3.1 Controlador en Tiempo Real/PLC	41
3.3.1.1 Variables de Red	44
3.3.1.2 Protocolo de Comunicación TCP/IP	44
3.3.2 Configuración del PLC.....	49
3.3.3 Componentes Físicos	53
3.3.3.1 Configuración del Software FlexSim	53
3.3.3.2 Interfaz de Control Físico	56

CAPÍTULO 4. PRUEBAS Y ANÁLISIS	59
Pruebas	60
4.1 Caso de estudio	61
4.2 Simulación con FlexSim	71
4.3 Conexión con Interfaz Hombre Máquina	75
CAPÍTULO 5. RESULTADOS FINALES	79
5.1 Resultados Generales	80
5.2 Conclusiones	81
5.3 Recomendaciones	81
5.4 Trabajos Futuros	82
Referencias	83

Índice de Figuras

Figura 1.1. Metodología para desarrollo del proyecto.....	8
Figura 2.1. Evolución de los procesos de producción. Fuente: (Oztemel, 2010)	11
Figura 2.2. Evolución de los métodos de manufactura. Fuente: (Oztemel, 2010)	12
Figura 2.3. Modelo conceptual de un holón ADACOR. Fuente: Leitão y Restivo.....	14
Figura 2.4. Arquitectura de un agente inteligente. Fuente: (D'Inverno y Luck, 2001)	19
Figura 2.5. Convergencia en paradigmas de sistemas de manufactura.....	23
Figura 2.6. Logotipo de la empresa FlexSim.	35
Figura 2.7. PLC Allen-Bradley Micro820. Fuente: (Automation, 2018)	36
Figura 2.8. PLC Siemens S7-1200. Fuente: (Siemens, 2019)	36
Figura 3.1. Arquitectura de agentes de dos capas.....	38
Figura 3.2. Encapsulamiento de mensaje en Modbus TCP/IP.....	45
Figura 3.3. PLC Micro820 de Allen-Bradley.....	50
Figura 3.4. Pantalla para seleccionar el modelo de PLC en Interfaz Hombre Máquina....	50
Figura 3.5. Lenguajes soportados en el PLC.....	51
Figura 3.6. Variables globales agregadas.....	51
Figura 3.7. Diagrama de escalera (LD) utilizado.....	52
Figura 3.8. Configuración de Modbus en PLC.....	52
Figura 3.9. Protocolos disponibles en FlexSim.....	53
Figura 3.10. Conexión en Emulation.....	54
Figura 3.11. Agregando sensores para Modbus TCP/IP.....	55
Figura 3.12. Agregando Eventos en sensores.....	55
Figura 3.13. Controles en Emulation.....	56
Figura 3.14. Control físico del PLC.....	57
Figura 3.15. Entradas en PLC.....	57
Figura 3.16. Salidas en PLC	58
Figura 4.1. Plano esquemático de la celda de manufactura de la UAEH.....	61
Figura 4.2. Diagrama BOM de cada producto.....	63
Figura 4.3. Gráfica de procesos para la tarea de Fresado del producto MILL_TEST ...	67
Figura 4.4. Gráfica de procesos para la tarea de Torneado del producto TURN_TEST..	68

Figura 4.5. Diagrama de proceso-ensamble del producto GAME-PROD.....	69
Figura 4.6. Objetos de FlexSim.....	72
Figura 4.7. Estación 1 (almacén).....	73
Figura 4.8. Estación 2 (maquinados).....	74
Figura 4.9. Estación 3 (ensamble).....	74
Figura 4.10. Modelo propuesto.....	75
Figura 4.11. Diagrama de escalera (LD) utilizado.....	77
Figura 4.12. Connections.....	77
Figura 4.13. Variables.....	78

Índice de Tablas

Tabla 2.1. Estándares y protocolos más comunes en la industria. Fuente: (Hurtado, 2019).....	29
Tabla 3.1. Especificaciones de Micro 820 Allen Bradley. Fuente: (Automation, 2018).....	41
Tabla 3.2. Elementos del encabezado MBAP. Fuente: (Modbus Organization, 2005-2019).....	46
Tabla 3.3. Bloques de memoria. Fuente: (Modbus Organization, 2005-2019).....	47
Tabla 3.4. Códigos de funciones. Fuente: (Modbus Organization, 2005-2019).....	48
Tabla 3.5. Composición de una trama Modbus TCP/IP	49
Tabla 4.1. Datos característicos de MP, componentes y producto.....	64
Tabla 4.2. Secuencia de operaciones para cada tipo de trabajo.....	66
Tabla 4.3. Máquinas, dispositivos y procesos.....	71

CAPÍTULO 1. CONTEXTO DE LA INVESTIGACIÓN

Introducción

Organización del Documento

1.1 Planteamiento del Problema

1.2 Solución Propuesta

1.3 Justificación

1.4 Objetivos

 1.4.1 Objetivo General

 1.4.2 Objetivos Específicos

1.5 Aportaciones

1.6 Metodología

Introducción

El éxito de una empresa ya no se mide por su capacidad para producir varios productos a un coste eficaz y rentable, ahora se mide por su capacidad de flexibilidad, agilidad y versatilidad (Botti y Giret, 2008), recientemente las compañías se dirigen más hacia la virtualización y la colaboración a corto plazo, esto cambia su antiguo objetivo que es la optimización del proceso, por las nuevas estrategias globales que van desde las funciones de diseño hasta el envío del producto. Lo anterior permite una mejor integración de la empresa, organización distribuida, cooperación, agilidad, escalabilidad, flexibilidad, calidad, tolerancia a fallos, adaptabilidad a los cambios del entorno y robustez contra la ocurrencia de fallos o cambios, mismas que se han convertido en la clave del éxito en muchas organizaciones (Rosenzweig y Easton, 2013).

Lo anterior, puede lograrse si las herramientas, máquinas y entornos de fabricación industrial integrados con control masivo de software se convierten en objetivos predeterminados de una organización. Con esta integración, los métodos de fabricación se dirigen hacia sistemas de manufactura totalmente automatizados y no tripulados, que pueden ser muy flexibles, reconfigurables, reutilizables, e interoperables, así como autónomos e inteligentes, considerados como una de las formas útiles de tratar con las expectativas del cliente (Oztemel, 2010).

Crear estrategias de control en tiempo real basadas en eventos y procesamiento de información distribuida que actúen sobre los sistemas de producción flexibles, no es una tarea fácil. El presente proyecto de investigación muestra la implementación de un puente de comunicación básico entre un Sistema Físico simple y un Modelo de Simulación de Eventos Discretos mismo que emula a recursos físicos de una celda de manufactura.

Para tal efecto, se espera que la comunicación entre ambos sistemas se realice en un modo síncrono en intención de demostrar el flujo de información. A ello, el presente documento plantea un caso de estudio fundamentado en una metodología específica, además, el documento, expone antecedentes históricos al tema, la metodológica empleada y los resultados de un caso de estudio particular.

Organización del Documento

Este documento de tesis se ha estructurado mediante 5 capítulos, de la siguiente forma:

El capítulo 1, hace una introducción de la propuesta, el planteamiento del problema, justificación, los objetivos y la metodología con la que se desarrolla el presente trabajo, así como de otros proyectos relacionados.

Dentro del capítulo 2, se muestra el marco teórico y algunos conceptos generales de los sistemas de holónicos de manufactura, se presentan los conceptos principales y secciones del modelo. Se hace énfasis en los requisitos de las características de los HMS, así como los conceptos sobre el protocolo Modbus.

En el capítulo 3, se plantea la construcción del modelo y su desarrollo que fundamento la presente tesis. Esto se hace en términos de modelos y herramientas típicas con los que se aborda una referencia de Sistema de Múltiples Agentes (MAS).

El capítulo 4, describe un caso de estudio particular donde se comprueba la comunicación entre un sistema físico y un sistema virtual.

Finalmente, en el capítulo 5 se exponen los resultados, conclusiones, recomendaciones y trabajos futuros que se pueden desarrollar con las bases presentadas. Estos puntos se analizan y se da una conclusión para que se tengan en cuenta en trabajos basados en el modelo propuesto o que contengan características similares a esta.

1.1 Planteamiento del Problema

En la actualidad, en el uso de los Sistemas Holónicos de Manufactura, es de suma importancia implementar estrategias que permitan controlar en tiempo real sistemas de producción y procesamiento de información distribuida, al no contar con los recursos suficientes o prever si es una propuesta óptima para montar un MAS, se utilizan los emuladores de eventos discretos los cuales nos acercan más al funcionamiento de un sistema real, dándonos la oportunidad de generar estrategias de control del MAS.

Ante este primer planteamiento, se hace evidente la necesidad de la comunicación entre las diferentes partes, los cuales son el MAS y el emulador de eventos discretos, con el objetivo de permitir el funcionamiento como uno y también cumpliendo con la adquisición de datos para su posterior análisis.

Con la información antes comentada, el identificar un medio de comunicación que logre la tarea de enviar y recibir datos es un punto de interés a tratar en esta investigación, al igual que elegir un protocolo disponible en el medio de comunicación para el tratamiento de los datos. Esto puede ser un tema complejo, debido a que es necesario obtener datos del emulador para su manejo y distribución a un nivel superior y viceversa, lo antes dicho, es evidente dado que en la literatura revisada los lenguajes de programación utilizados para el desarrollar estos sistemas son diferentes.

En resumen, este trabajo busca comunicar dos diferentes sistemas haciendo uso de un protocolo de comunicación adecuado que permita validar el flujo de información entre ambos sistemas.

De acuerdo con la descripción del problema se formula el siguiente enunciado holopráxico de la investigación (Barrera, 2012):

- ¿Cómo se puede realizar la comunicación entre dos programas de propósito diferente, que pueda sentar las bases para el control mutuo?

1.2 Solución Propuesta

La comunicación es una parte esencial cuando se habla de internet o dentro de la vida cotidiana, debido a que nos permite relacionarnos con otras personas, lo que a su vez crea la interacción de una sociedad. La presente aportación trata sobre la comunicación entre un MAS y un emulador de eventos discretos. Los capítulos de este documento describen mediante una metodología específica el desarrollo de los pilares que serán base para el escalamiento futuro del proyecto de investigación, se desarrollara un Cliente haciendo uso del protocolo Modbus TCP/IP (por sus siglas en inglés, Transmission Control Protocol/Internet Protocol), permitiendo comunicación bidireccional.

1.3 Justificación

El lograr visualizar la reorganización y/o estructuración de los sistemas de producción flexibles y/o de ambiente virtual controlados por un HMS depende en gran medida de los mecanismos de control y de los protocolos de comunicación entre los sistemas, lo que genera una necesidad de comprender el flujo de los datos entre los elementos de la red, esto, nos permitirá agilizar la intercomunicación y la interacción de los elementos como un sistema auto operativo.

Considerando lo mencionado en el párrafo anterior, el presente trabajo propone realizar la comunicación por medio del protocolo Modbus TCP/IP, mismos que permiten comunicar componentes que trabajan con diferentes lenguajes de programación, ejemplo de ellos son los agentes inteligentes que se desarrollan en JAVA y el simulador de eventos discretos que está desarrollado en C++.

La utilización del modelo propuesto proyecta una idea clara del uso y distribución de información entre un MAS y un emulador virtual, aportando conocimiento científico a los temas de investigación respectivos a sistemas de manufactura inteligentes.

1.4 Objetivos

1.4.1 Objetivo General

Crear un módulo de comunicación entre un sistema físico y un sistema de manufactura virtual, haciendo uso del protocolo TCP/IP, que permita el transporte de información en tiempo real, con la finalidad de lograr interacción de las partes y generando un entorno virtual que actúe según los datos recibidos.

1.4.2 Objetivos Específicos

- Dar fundamentos teóricos a la investigación mediante el análisis de bibliografía referida a sistemas de producción, manufactura inteligentes y simulación de eventos discretos.
- Emular una línea de producción a través de la simulación de eventos discretos, con la finalidad de visualizar en tercera dimensión el comportamiento del conjunto de elementos del mismo;
- En JADE, crear un módulo de comunicación utilizando Sockets y el protocolo TCP/IP para permitir la comunicación;
- Validar el modelo de comunicación y flujo de la información mediante la experimentación de ejercicios variados y,
- Hacer un análisis de datos, información y resultados obtenidos para evidenciar los alcances de la investigación.

1.5 Aportaciones

Los aportes de este trabajo son definir los temas importantes para identificar los elementos de la arquitectura de dos capas de Thomas y Alois (2015), comprobando la teoría expuesta y adaptando el modelo propuesto con herramientas desarrolladas por terceros y de fácil acceso. Se trata de ofrecer un modelo de desarrollo de fácil implementación, para su posterior desarrollo y adaptando las necesidades requeridas para llegar a un Sistema de Holónico de manufactura eficiente.

CONTEXTO DE LA INVESTIGACIÓN

La investigación desarrolla el uso del protocolo Modbus TCP/IP, mismo que es utilizado en el sector industrial por su fácil implementación y adaptación a los requerimientos solicitados por los fabricantes. Dicho protocolo se usa en la interacción de los elementos planteados en la arquitectura holónica antes mencionado.

1.6 Metodología de Investigación

Metodología empleada en el desarrollo del proyecto a través de 4 fases, como se muestra en la Figura 1.1:

Fase 1: Estado del arte y especificación de requerimientos

En la primera fase fue revisado el estado del arte; principalmente se revisaron los paradigmas utilizados en la representación del control de sistemas de manufactura inteligente, identificando los elementos constitutivos, requerimientos, atributos y tipologías existentes.

De igual forma, se hace hincapié en el análisis e identificación de los requerimientos y exigencias de los sistemas de control implementables en sistemas de eventos discretos.

Fase 2: Análisis

La fase número 2 analiza y diferencia las bondades y características encontradas entre diferentes paradigmas respectivos al control que ejercen los HMS sobre los sistemas de manufactura flexibles o de eventos discretos.

Puntualmente, se hace hincapié en el análisis de los medios de comunicación empleados entre los sistemas antes mencionados.

Fase 3: Propuesta

Se propone el modelo de un sistema de comunicación básico entre un MAS y el software de ambiente virtual de nombre FlexSim. Se hace uso del protocolo TCP/IP.

El alcance de la aplicación propuesta permite el flujo de información simple entre los elementos integrantes de un MAS y el emulador virtual desarrollado por la empresa FlexSim Software Inc.

Fase 4: Implementación y validación del prototipo

En esta última fase se implementa la propuesta y se realiza validación cuantitativa del tipo de información que fluye sobre el MAS y el emulador virtual, con la finalidad de comprobar el grado de efectividad en la obtención de datos.

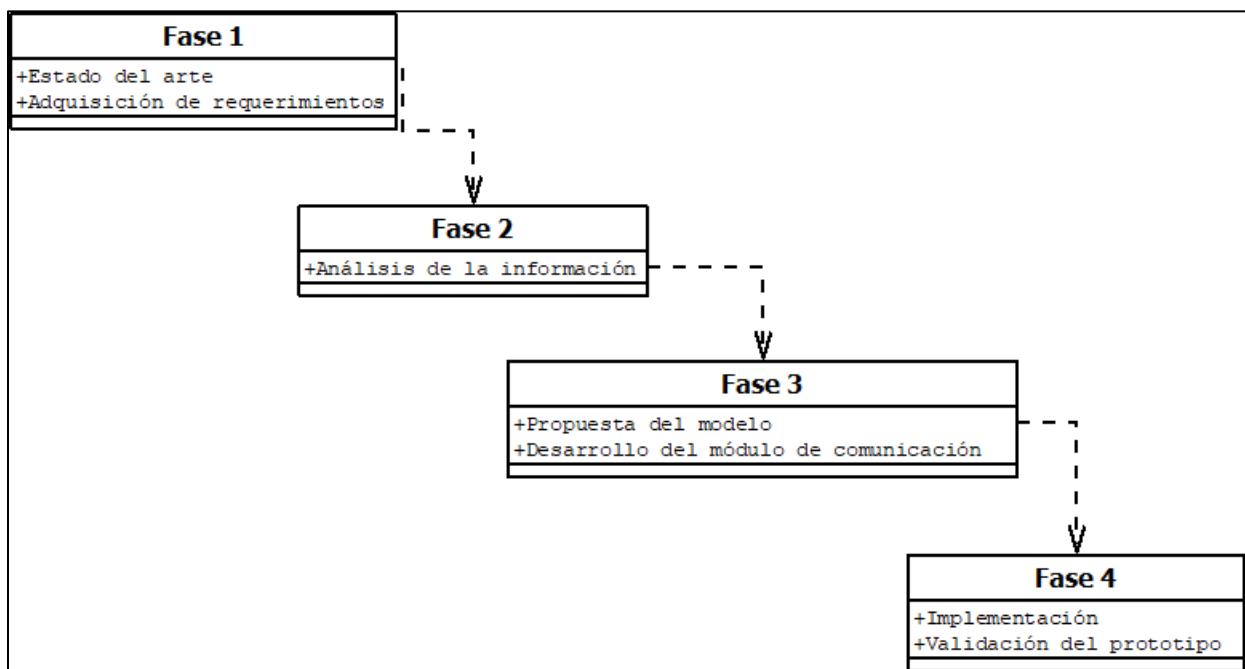


Figura 1.1. Metodología para desarrollo del proyecto.

CAPÍTULO 2. MARCO TEÓRICO Y CONCEPTOS GENERALES

2.1 Sistemas de Producción

2.1.1 Sistema de Manufactura Tradicional

2.2 Sistemas Holónicos de Manufactura

2.3 Inteligencia Artificial

2.3.1 Inteligencia Artificial Distribuida

2.3.2 Sistemas de Manufactura Basados en Agentes Inteligentes

2.4 Agentes Inteligentes

2.4.1 Herramientas Para su Desarrollo

2.4.2 JADE

2.4.3 FIPA

2.5 Sistemas Cíber-Físicos

2.5.1 Gemelos Virtuales

2.6 Comunicación

2.6.1 Los Protocolos

2.6.2 El Protocolo TCP

2.6.2.1 El Modelo TCP/IP

2.6.3 Redes de Comunicación en la Industria

2.6.3.1 Protocolos y Estándares Existentes en la Industria

2.6.3.2 Protocolo Modbus

2.7 Sistemas de Eventos Discretos

2.7.1 Simulación por Eventos Discretos

2.7.2 Software FlexSim

2.8 Controlador Lógico Programable

Introducción

En ingeniería un modelo es la representación gráfica o esquemática de una realidad, que nos sirve para organizar y comunicar de forma clara los elementos que se involucran. A esto, la ciencia moderna ofrece una colección creciente de métodos, técnicas y teorías acerca de diversos tipos de modelos. Los apartados de este capítulo buscan introducir al lector en temas teóricos a fin con los objetivos de esta investigación, con la única intención de avanzar en el conocimiento necesarios para la propuesta de un modelo de comunicación exitoso entre un sistema de MAS y uno de eventos discretos.

2.1 Sistemas de Producción

Los sistemas de producción son una serie de elementos organizados que se relacionan y que interactúan entre sí, estos elementos pueden ser maquinas, personas, materiales o procedimientos. Para esclarecer la idea del comportamiento de los mismos el tema 2.2.1 detalla características de interés.

2.1.1 Sistema de Manufactura Tradicional

La manufactura tradicional involucra una gran cantidad de actividades dependientes entre sí, formadas por distintas entidades (como materiales, herramientas, máquinas, energía y seres humanos) y que estos a su vez forman un sistema. También, se debe resaltar que el término “proceso” implica el uso de recursos de una organización, con el fin de obtener un valor. Cuando se crea un producto o se presta un servicio, se hace necesario contar con un proceso establecido. Los procesos globales son llevados por actividades en toda organización, en las que se representan todas las funciones distribuidas en la organización. De esta forma, los sistemas de producción se caracterizan por ser complejos debido a la cantidad de elementos, físicos y humanos, que generalmente se hacen difíciles de predecir y controlar. Lo anterior aunado a las necesidades siempre cambiantes de los clientes y la competitividad global están

forzando a un cambio primordial en los estilos de fabricación y configuración de organizaciones industriales.

Los Sistemas de Manufactura Tradicionales (SMT, por sus siglas en inglés), sufrieron cambios con la llegada de la globalización, los SMT estaban conformados por un conjunto de etapas las cuales eran: diseño, planeación de procesos, planeación y control de la producción, manufactura, gestión de calidad y almacenamiento y envío (Oztemel, 2010), con los años algunos aspectos influyeron su desempeño, estos son:

- Cambios tecnológicos en los sistemas mecánicos de producción.
- Métodos de manufactura.
- Cambios en la planeación de la producción y administrativos.
- Cambios ambientales.
- Expectativa y necesidades de los clientes.

Actualmente las operaciones de mecanizado se han transformado en sistemas autónomos capaces de manejar actividades de manufactura relacionadas con todos los aspectos de producción y cuestiones medioambientales, los avances tecnológicos también fueron implementados en la manufactura y así cambiando su infraestructura (véase la Figura 2.1).

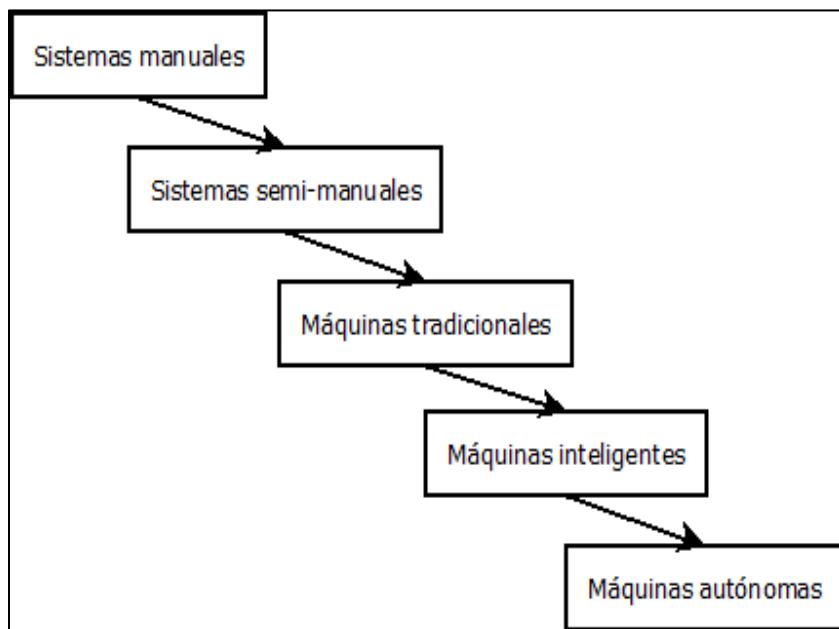


Figura 2.1. Evolución de los procesos de producción. Fuente: (Oztemel, 2010).

En la Figura 2.1, muestra la evolución que sufrieron los procesos de producción los cuales se adaptaron con la llegada de nuevas tecnologías que eran implementadas en la industria, permitiendo también la evolución de los métodos de manufactura, como se muestra en la Figura 2.2.

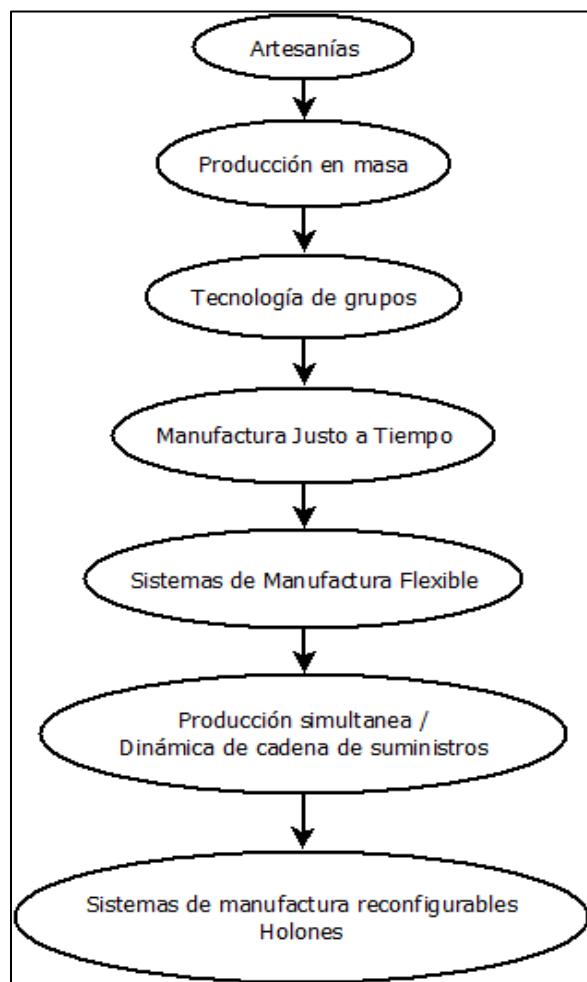


Figura 2.2. Evolución de los métodos de manufactura. Fuente: (Oztemel, 2010).

Los cambios también afectaron a los métodos de manufactura como se muestra en la Figura 2.2, adoptando un nuevo marco para los Sistemas de Manufactura Inteligentes (SMI, por sus siglas en inglés) (Botti y Giret 2008), cuyos principios son fundamentados en el uso de la Inteligencia Artificial (IA).

2.2 Sistemas Holónicos de Manufactura

Simón-Marmolejo, López-Ortega, y Ramos-Velasco (2016) mencionan que nuevos métodos de fabricación están surgiendo hacia sistemas de manufactura totalmente automatizados y no tripulados, que podría ser muy flexible, reconfigurable, reutilizable, e interoperables, así como autónomos e inteligentes considerados como una de las formas útiles de tratar con las expectativas del cliente. Uno de estos enfoques que permitan una mejor gestión de la producción en los aspectos tocantes es el concepto de sistemas holónicos. El desarrollo del concepto Sistemas Holónicos de Manufactura es guiado por métodos y principios de ingeniería de software, con lo que se hace latente la necesidad en la gestión de datos, información y conocimiento: los datos en su mayoría son estructurados de acuerdo con modelos de datos, en los sistemas holónicos los datos son contenidos en ontologías propias, lo que conlleva a la consistencia de problemas en el intercambio.

El concepto holónico surge en la obra del autor y filósofo húngaro Arthur Koestler en 1967, quien intentó capturar el comportamiento de los sistemas complejos, considerándolos ser un todo y a la vez una parte (Koestler, 1967). Para describir una unidad básica de la organización en los sistemas biológicos y sociales, Koestler inventó la palabra "holón", que viene de la combinación de la palabra griega "holo" que significa todo o entero, y del sufijo "on" que expresa una parte o una partícula, los holones se comportan "en parte como totalidades y completamente como partes", Koestler también propuso el concepto de Open-Ended Hierarchy (OEH, por sus siglas en inglés) o Jerarquía Indefinida como una arquitectura formada de holones, llamada holarquía (Radu F. Babiceanu, 2006). Una Holarquía, es un conjunto de holones que pueden cooperar para alcanzar una meta o un objetivo, la misma, define reglas básicas para la cooperación de estos, limitando su autonomía. Una de sus características más llamativas es que su estructura se puede adaptar y optimizar constantemente, ofreciendo una gran versatilidad a los cambios que se pueden desarrollar en el entorno. Algunas de las cualidades holónicas son: autonomía, reactividad, proactividad, cooperación, reorganización, racionalidad, aprendizaje, recursión y procesamiento físico.

Los elementos que componen a un holón son, “cuerpo”, en el que se desarrollan los procesos de transformación, transporte, almacenamiento (realizados por reactores, compresores, almacenes, etc.), “cabeza”, aquí se desarrollan los procesos de toma de decisiones respecto de la producción, basados en el conocimiento que se tiene del proceso de producción y de los recursos (hombres y máquinas), y “cuello”, éste es la interfaz entre el “cuerpo” y la “cabeza”, es toda la infraestructura teleinformática que almacena, soporta y transporta la información. En relación a este último, frecuentemente hace uso del Protocolo de Control de Transmisión (TCP, por sus siglas en inglés) y del Protocolo de Internet (IP, por sus siglas en inglés) para la comunicación de los datos digitales. El TCP/IP es clasificado funcionalmente a nivel de red según el modelo del Sistema de Interconexión Abierto (OSI, por sus siglas en inglés). En la Figura 2.4 se muestran los componentes de un holón.

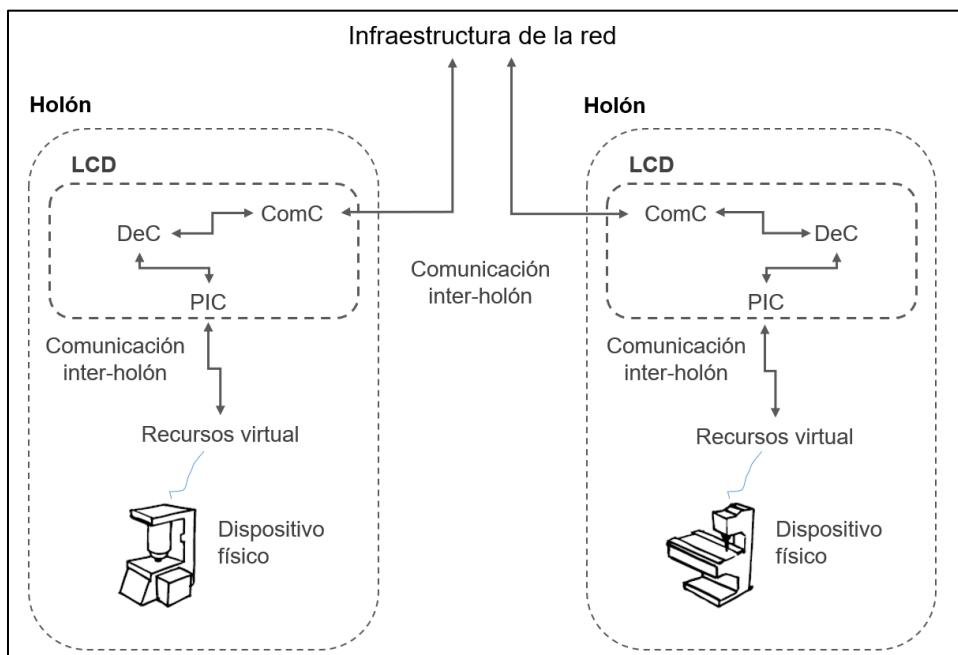


Figura 2.3. Modelo conceptual de un holón ADACOR. Fuente: Leitão y Restivo, 2006.

Finalmente, es importante aclarar que el análisis y los resultados de la investigación propuesta por (Leitão y Restivo, 2006), suscitan el tema de manufactura holónico desde el punto de vista conceptual, y muestran que, *para entender e integrar holones*, la tecnología MAS es la una plataforma identificada que representa dispositivos físicos

(máquinas, robots, cintas transportadoras, almacenes, productos, etc.) y entidades no físicas (órdenes de clientes, órdenes de manufactura, planes de producción, programas globales de producción, entre otros) de un sistema de producción.

2.3 Inteligencia Artificial

La IA es definida como la automatización de las actividades que se relacionan con procesos del pensamiento y razonamiento humano, actividades como la toma de decisiones y resolución de problemas (Russell y Norvig, 2004). También se define como el campo de estudio que busca explicar y emular el comportamiento inteligente en términos de procesos computacionales (Schalkoff, 1990).

Dentro de la IA, existen diversas áreas de estudio como: Agentes Inteligentes, Sistemas Basados en Conocimiento, Procesamiento del Lenguaje Natural, Reconocimiento de Patrones, Machine Learning, Redes Neuronales, entre otras más. Los MAS son un campo importante en la inteligencia artificial distribuida que se dedica al estudio de técnicas y el conocimiento necesario para la coordinación y distribución del mismo, además de estudiar las acciones en un entorno con múltiples agentes que por lo general se usan en la industria. Más información se detalla en el siguiente apartado.

2.3.1 Inteligencia Artificial Distribuida

La Inteligencia Artificial Distribuida (DAI, por sus siglas en inglés) es un campo de la IA, mismo que se caracterizan por una arquitectura formada por componentes inteligentes (Agentes) y módulos que interactúan de forma coordinada, teniendo ventajas en su utilización de enfoque sobre los paradigmas convencionales (Serrano y Ossowski, 1998), como son:

- Aprovechan la distribución natural del dominio, mejorando el rendimiento, la robustez, reutilización y mantenimiento.
- Generan un valor añadido con una mejor aceptabilidad en la sociedad, favoreciendo la aceptación de estructuras ya existentes en las organizaciones humanas y facilitan la interacción hombre-máquina.
- Favorecen el desarrollo de modelos cognitivos de cooperación y coordinación como fenómenos complejos y generan evidencia para teorías lingüísticas, psicológicas, sociológicas y filosóficas respecto a estos temas.

Las virtudes antes mencionadas han hecho posible que la tecnología DAI sea empleada para solucionar problemas de programación, coordinación y control de operaciones dentro de los sistemas de manufactura inteligentes.

2.3.2 Sistemas de Manufactura Basados en Agentes Inteligentes

Los nuevos requerimientos para los sistemas de control actual son las unidades de sistemas distribuidas y descentralizadas, conocimiento y habilidades, mecanismos de comunicación para la integración de sistemas distribuidos, rápida adaptación en el control y la reacción a disturbios (falla y cambios de organización) e interacción entre dispositivos físicos. Desde el punto de vista de MAS los “Sistemas de manufactura inteligente” consisten en una red integrada de recursos distribuidos, conocimiento de control y procesamiento de materiales.

MAS han sido reconocidos como un promisorio paradigma para la próxima generación de sistemas de manufactura (Shen, Q., y H. and Norrie, 2006), como un enfoque importante para desarrollar sistemas de manufactura inteligente distribuida. En resumen, la comunidad científica ha logrado aplicar tecnología de agentes para la integración de sistemas de manufactura, planeación y programación de producción, y manufactura holónica como una implementación metodológica (Deen, 2003), entre otras aplicaciones específicas. En este sentido es posible identificar desarrollos en sistemas de control basados en agentes que son autónomos, organizados y reactivos (Deen, 2003).

2.4 Agentes Inteligentes

A principios de los noventa los agentes inteligentes y los sistemas basados en agentes comenzaron a convertirse en un importante tema de investigación. Estos sistemas basados en agentes se han relacionado con aplicaciones para la vida en una enorme cantidad de disciplinas, especialmente en los sectores industriales y de servicios donde los avances maduran constantemente.

Posteriormente, se introduce el paradigma de agentes de software inteligentes en el campo de la DAI y su utilización en el área de fabricación, con el objetivo de diseñar sistemas distribuidos con un alto grado de complejidad como una técnica para entender y simular sociedades o bien sistemas de producción (D'Inverno y Luck, 2001).

Un agente es una entidad capaz de percibir su entorno, procesar tales percepciones y responder o actuar en su entorno de manera racional, percibiendo con sensores y afectando al entorno con actuadores (Wooldridge, 2009), estos continuamente realizan procesos que les indican qué hacer y cómo, estas entidades se comunican con otros agentes para resolver de forma adecuada su trabajo. Son capaces de incorporar más de una tecnología de AI; éstos pueden aprender y trabajar al mismo tiempo mediante la percepción de su medio ambiente con la ayuda de sensores y actuadores en un entorno abierto y dinámico en el sentido de estar siempre observando y actuando sobre él con el fin de alcanzar los objetivos definidos (Wooldridge y Jennings, 1995).

En el entendido, para referirse a esta entidad o poder hacer uso del término agente o agentes avanzados se dan algunas propiedades que se tienen que cumplir.

- **Autonomía:** Un agente inteligente, tiene control sobre su comportamiento (opera sin la intervención directa de seres humanos u otras entidades del mundo exterior). Tiene el control exclusivo sobre su estado interno y sus objetivos y es la única instancia que puede cambiar.
- **Respuesta/Situación:** Un agente está equipado con sensores y actuadores que desde su interfaz directa con su entorno percibe su medio ambiente al recibir

inundaciones sensoriales de él. Responde de manera oportuna al cambio relevante en el mismo a través de sus actuadores.

- **Proactividad:** Un agente más sofisticado actúa no sólo de manera respondiente, sino que puede ser oportunista y actuar por iniciativa (puede anticipar proactivamente posibles cambios en su entorno y reaccionar ante ellos).
- **Orientado a objetivos:** Un agente inteligente está dirigido por objetivos. Esto implica que toma iniciativa donde quiera que haya una oportunidad de trabajar hacia sus metas.
- **Comportamiento inteligente:** Un agente tiene amplia experiencia y conocimiento en un área específica y bien definida. Son capaces de tratar y resolver problemas en este dominio.

La Figura 2.3, muestra la arquitectura general de un agente inteligente, el cual consta de tres componentes esenciales: percepción, cognición y acción. La percepción recibe entradas provenientes del medio ambiente a través de sensores y las transmite al módulo de la cognición para ser procesadas. Este proceso puede incluir filtros y priorizar el orden de importancia. La cognición implica procesar la información percibida y, en consecuencia, tomar una decisión, este proceso requiere de distintos métodos de sistemas inteligentes, tales como aprendizaje e implantación. Y el mecanismo de la acción de un agente, puede hacer frente a situaciones inesperadas y adaptarse a nuevas situaciones lo más rápido posible.

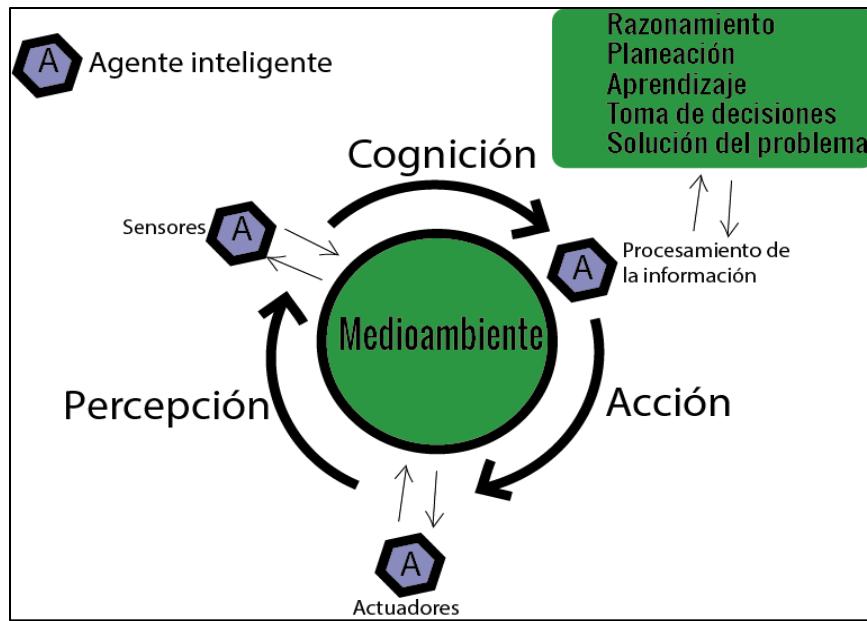


Figura 2.4. Arquitectura de un agente inteligente. Fuente: (D'Inverno y Luck, 2001).

Por otro lado, los agentes inteligentes se pueden clasificar como:

- Sistemas reactivos
- Sistemas cognitivos
- Sistemas deliberativos

El sistema reactivo busca que un agente se asemeje a un actor, reaccionando sólo a estímulos externos, con poca o ninguna comunicación. El enfoque cognitivo de agentes es considerado a menudo como agentes bastante complejos, que tiene internamente modelos simbólicos del mundo y de sus habilidades, con capacidad de razonamiento y comunicación sofisticada. Un agente deliberativo, respectivamente cognitivo, es sinónimo de agente proactivo o una especialización del mismo. Su comportamiento y arquitectura son razonablemente sofisticados.

A esto, es importante hacer notar que los deseos representan los objetivos del agente, es decir, describen lo que el agente desea lograr. Se puede distinguir entre objetivos a corto plazo y objetivos a largo plazo. Los objetivos a largo plazo son los que realmente conducen el comportamiento de un agente y, por lo tanto, son comparativamente

estables y abstractos. Los objetivos a corto plazo sólo reflejan las metas que el agente desea lograr en una situación específica. Éstos pueden expresar como máximo lo que el agente puede hacer en esta situación específica y, por lo general, sólo tienen un carácter temporal. La planificación inteligente es un proceso complejo especialmente si el plan resultante es comparativamente sofisticado y abarca un gran espacio de solución que crece exponencialmente. Durante este tiempo de planificación, el entorno puede cambiar de una manera que hace que la ejecución del plan real sea obsoleta o sub-óptimo. Por lo tanto, una replanificación inmediata puede ser necesaria. Al respecto, según (Vlahavas y Vrakas, 2005), tienen la idea de que los agentes deliberativos son especialmente útiles cuando se requiere pasar de una situación razonable a una sofisticada, sin embargo, no sucede en tiempo real debido a su capacidad para producir soluciones de alta calidad, independientes del dominio.

2.4.1 Herramientas para su Desarrollo

Atendiendo a lo antes mencionado, MAS es la única plataforma identificada como la herramienta de modelado para el desarrollo de sistemas holónicos. Las características de autonomía, cooperación, reactividad y proactividad de los agentes hacen de una organización de agentes o arquitectura MAS una herramienta adecuada para aplicar los conceptos de HMS (Babiceanu, 2006). Por lo tanto, la investigación respectiva a HMS está fuertemente relacionada con la investigación del MAS en la comunidad de DAI. Ulieru, Stefanou, y Norrie (2000) declaró, "el paradigma de sistemas de múltiples agentes parece ser muy adecuado para la implementación de una abstracción holónica en un problema que se distribuye fundamentalmente en la naturaleza".

De esta manera y dado que Java Agent DEvelopment Framework (JADE) es la tecnología software de sistemas de múltiples agentes, ampliamente aceptada por la comunidad de sistema holónicos (Leitão y Pavel, 2011), los principales proyectos de investigación en el tema han recurrido a esta plataforma en intención de proporcionar un entorno de ejecución para agentes software. Con ello, es posible la implementación de distintos tipos de holones como agentes JADE, usando para ello la clase Agent

proporcionada para tal efecto y acorde con las especificaciones Foundation for Intelligent Physical Agents (FIPA), y una serie de librerías Java que facilitan implementar los principales componentes de los agentes: los comportamientos y las interacciones (Araúzoa, Martínez, Laviós, y Martína, 2015). Así, esta clase Agent, que tiene ya implementadas las funcionalidades básicas de los agentes (búsqueda de otros agentes, comunicación, gestión de comportamientos, gestión de la línea de espera de menajes, etc.), se extiende para crear las clases que representan cada tipo holón de un sistema de manufactura (holones productos, holones tareas, holones operativo y holones supervisor), añadiendo en cada caso los comportamientos e interacciones descritos en el apartado anterior.

La siguiente sección describe características generales de las herramientas empleadas en el modelado de HMS. Se inicia la descripción de JADE y sus componentes más importantes, se dan conceptos respectivos al módulo para la comunicación y los protocolos necesarios, y por último se describe un ejemplo del software más empleado para simular un entorno real en el que podrán actuar los agentes inteligentes.

2.4.2 JADE

Java Agent DEvelopment Framework (JADE) es un Framework de software implementado en el lenguaje Java y como requisito mínimo para su uso es tener la versión 5 del Kit de desarrollo de Java (JDK, por sus siglas en inglés). Simplifica la creación e implementación de sistemas multiagentes a través de un software intermedio, esta librería cumple con las especificaciones Foundation for Intelligent Physical Agents (FIPA), y la configuración puede ser controlada a través de una Interfaz Gráfica de Usuario (GUI, por sus siglas en inglés) remota. Un sistema basado en JADE se puede ejecutar y comunicar entre máquinas que no comparten el mismo sistema operativo.

2.4.3 FIPA

La Fundación para Agentes Físicos Inteligentes (FIPA, por sus siglas en inglés) es un organismo para el desarrollo y establecimiento de estándares de software para sistemas basados en agentes inteligentes. De los estándares FIPA los más ampliamente adoptados son las especificaciones de Lenguaje de Comunicación de Agentes (FIPA-ACL, por sus siglas en inglés) (FIPA, 1996-2018).

2.5 Sistemas Cíber-Físicos

Los sistemas cíber-físicos (CPS) son sistemas de entidades computacionales colaborativas que están en conexión intensiva con el mundo físico circundante y sus procesos en curso, proporcionando y utilizando, al mismo tiempo, los servicios de acceso y procesamiento de datos disponibles en el Internet. En otras palabras, el CPS se puede caracterizar generalmente como "sistemas físicos y de ingeniería cuyas operaciones son monitoreadas, controladas, coordinadas e integradas por un núcleo de computación y comunicación" (Rajkumar et al. 2010).

Un sistema físico está compuesto por equipos, maquinaria, objetos y cosas reales, el hardware, mientras que un sistema cibernetico es aquel que tiene la capacidad de procesos de control y comunicación tanto en las máquinas como en los seres vivos.

La interacción entre los elementos físicos y ciberneticos es de importancia clave: "CPS se trata de la intersección, no de la unión, de lo físico y lo cibernetico. No es suficiente entender por separado los componentes físicos y los componentes computacionales. Debemos entender su interacción (Lee y Seshia 2014).

Las características esenciales de los sistemas ciber-físicos son la capacidad de relacionarse con los objetos físicos para monitorizar y/o controlar, y la utilización de la información disponible en el mundo virtual, pudiendo tener en algunos casos capacidad de aprender y evolucionar.

MARCO TEÓRICO Y CONCEPTOS GENERALES

La última tendencia en la personalización masiva es la digitalización en la fabricación, también conocida bajo el término Industria 4.0 o CPS. La noción de la fusión ciber-física es adoptada cada vez más por los fabricantes. Sin embargo, cómo converger los mundos físicos y ciberneticos de la fabricación sigue siendo uno de los desafíos específicos.

El término Industria 4.0 va asociado a la cuarta revolución industrial que se puede considerar como un nuevo nivel de digitalización, organización y control de toda la cadena de valor del ciclo de vida de los productos, su trazabilidad, la conectividad máquina a máquina y la personalización del cliente (M. Rüßmann, et al 2015)

La Industria 4.0 tiene como base múltiples tecnologías, (Big Data, Cloud Computing, Fabricación Aditiva, Impresión 3D, Robótica Colaborativa etc.), algunas ya consolidadas y otras en proceso de desarrollo, uno de los elementos que posibilitaran el cambio de paradigma son los denominados sistemas ciber-físicos (CPS).

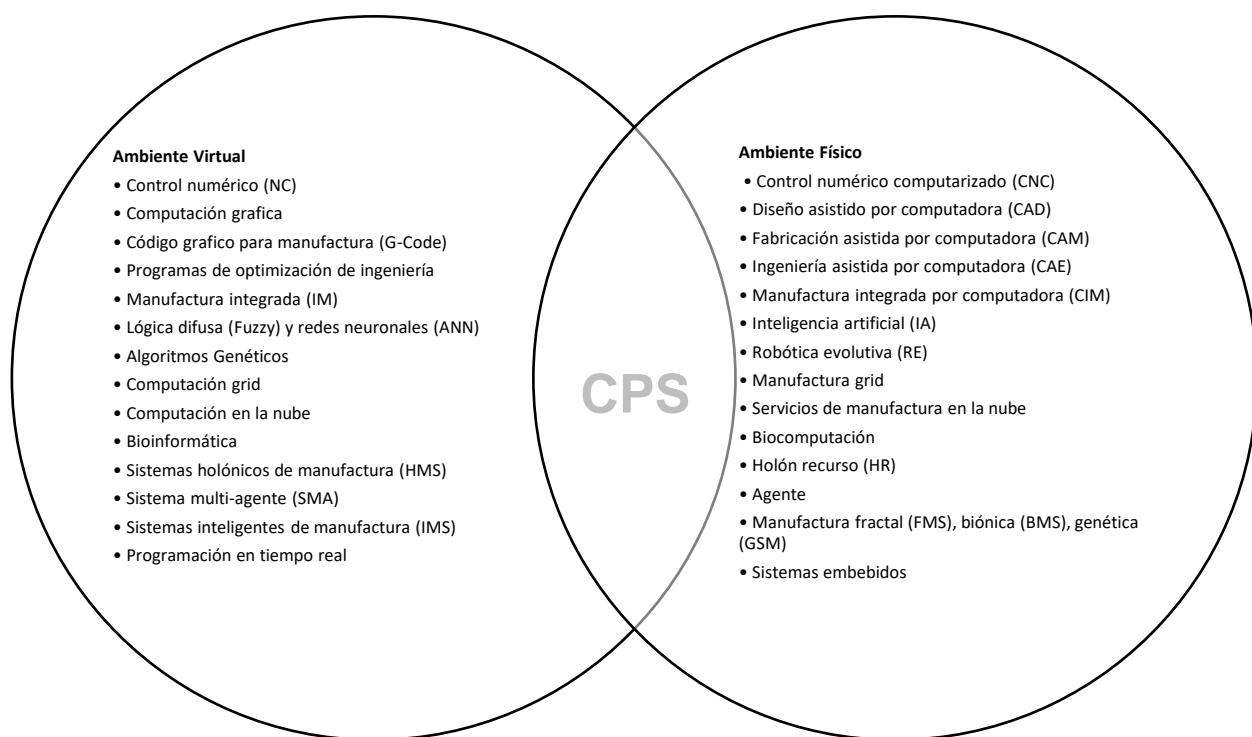


Figura 2.5. Convergencia en paradigmas de sistemas de manufactura. Fuente: (Saumeth C. et al. 2015; ElMaraghy & Monostori 2014; Monostori et al. 2016)

2.5.1 Gemelos Digitales

Un gemelo digital es una réplica digital de una entidad física viva o no viviente. Al unir el mundo físico y el virtual, los datos se transmiten a la perfección, lo que permite que la entidad virtual exista simultáneamente con la entidad física. (Saddik, A. 2018)

El gemelo digital consiste en crear modelos virtuales de alta fidelidad para objetos físicos en forma digital para simular sus comportamientos, generará una planificación de fabricación más razonable y un control de producción preciso para ayudar a lograr fabricación inteligente, a través de la conectividad bidireccional entre los mundos virtual y físico de la fabricación. (Hochhalter et al. 2008)

Rob Spiegel (2019), director de desarrollo de cartera para fabricación digital en Siemens PLM Menciona que las funciones básicas de un gemelo digital son:

- Entender el diseño y la producción de productos: Los datos capturados y administrados en el gemelo digital ofrecen una gran cantidad de información sobre el producto, sus simulaciones, su composición, sus procesos de producción y su comportamiento esperado en el campo.
- Predicción: El gemelo digital recopila y administra los datos necesarios para predecir cómo se comportará el producto desde la producción hasta su vida útil en el campo. Predecir el comportamiento de un producto en función de las condiciones y restricciones con precisión en cómo se comportará el producto en diferentes condiciones y entornos, además, predecir las operaciones de la planta durante su producción.
- Optimizar: Utilizar el gemelo digital para optimizar el producto, la planta y el rendimiento del producto en el campo a través de la automatización y simulación de la toma de decisiones.

En el entendido, un gemelo digital proporciona una forma efectiva para la integración ciber-física de la fabricación. Mientras tanto, los servicios de fabricación inteligentes podrían optimizar todos los procesos de negocio y el procedimiento de operación de fabricación, para lograr un nuevo nivel de productividad más alto.

El gemelo digital se puede dividir en tres niveles, es decir, nivel de unidad, nivel de sistema y nivel SoS. El gemelo digital de unidad es el equipo (el equipo es la unidad más pequeña dentro de las actividades de fabricación). Con respecto al gemelo digital a nivel

MARCO TEÓRICO Y CONCEPTOS GENERALES

de sistema, una línea de producción inteligente compuesta por maquinas herramienta, robots, etc. es a nivel de sistema. Para los gemelos digitales a nivel unidad y nivel de sistema, los modelos virtuales son el mapeo de ultra alta fidelidad del equipo físico. Los atributos básicos, el estado en tiempo real y otros datos se transmiten a los modelos virtuales para impulsar la simulación y la predicción. Luego, los parámetros de los modelos virtuales se retroalimentan para optimizar las entidades físicas. El gemelo digital de nivel de sistema puede considerarse como la integración de un gemelo digital de nivel de unidad múltiple, que cooperan entre sí. Múltiples gemelos digitales de nivel de unidad o múltiples gemelos digitales de nivel de sistema constituyen el gemelo digital de nivel SoS, es decir, un sistema complejo. Con respecto al gemelo digital a nivel SoS (por ejemplo, el taller) la gestión precisa del taller y las operaciones confiables, que son inseparables de los servicios, son muy importantes para la fabricación inteligente. (Tao et al., 2017; Wang y Xu, 2014)

Una representación completa de Digital Twin consta de cuatro elementos que son (1) espacio real, (2) espacio virtual, (3) el enlace de datos del espacio real al espacio virtual, (4) y el enlace de información del espacio virtual al espacio real y virtual subespacios (se refiere a una o más representaciones digitales del producto). Un producto físico opera en el espacio real, mientras que una máquina virtual (es decir, un modelo de simulación) se crea en el espacio virtual. Una máquina virtual puede construirse usando una simulación multicuerpo. Con este enfoque, es posible construir un modelo de simulación basado en la física detallada para analizar la dinámica de productos complejos como una máquina móvil.

La comunicación entre el sistema físico y el gemelo digital permite obtener análisis inteligentes de datos y monitorización de sistemas para evitar problemas antes de que se produzcan, prevenir tiempos de inactividad, desarrollar nuevas oportunidades de negocios, planificar el futuro mediante simulaciones y personalizar la producción a los requerimientos del cliente.

2.6 Comunicación

Teóricamente, la comunicación se define como: “El objetivo de un sistema de comunicación consiste en asegurar entre una fuente de información y un destinatario una relación mediante la cual la primera afecte la conducta del segundo.” (Jaime, 1975)

Teniendo en cuenta lo anterior, la comunicación es el proceso de transmisión y recepción de datos, es un acto complejo que implica la conexión entre dos o más dispositivos, la información se puede transmitir de forma analógica, digital o mixta y a través de diversos medios (aire, vacío, cable de cobre, fibra óptica, entre otras) por medio de ondas electromagnéticas.

2.6.1 Los Protocolos

En este sentido, los protocolos son un sistema de reglas que permiten que dos o más entidades se comuniquen entre ellas para transmitir información por medio de cualquier medio físico. Se tratan de las reglas o el estándar que definen la sintaxis, semántica y sincronización de la comunicación, así como también los posibles métodos de recuperación por errores, estos se encuentran en cada capa, cabe mencionar que cada capa puede tener más de un protocolo implementado. (Tanenbaum y Wetherall, 2012)

2.6.2 El Protocolo TCP

El Protocolo de Control de Transmisión (TCP, por sus siglas en inglés), se diseñó específicamente para proporcionar un flujo de bytes confiable de extremo a extremo a través de una interred no confiable, esta diseñó para adaptarse de manera dinámica a las propiedades de la interred y sobreponerse a muchos tipos de fallas.

Cada máquina que soporta TCP tiene una entidad de transporte TCP, ya sea un procedimiento de biblioteca, un proceso de usuario o que sea parte del kernel. En todos los casos, maneja flujos TCP e interactúa con la capa Protocolo de Internet (IP, por sus

siglas en inglés). Una entidad TCP acepta flujos de datos de usuario de procesos locales, los divide en fragmentos que no excedan los 64 KB (por lo general son 1460 bytes de datos para ajustarlos en una sola trama Ethernet con los encabezados IP y TCP), y envía cada pieza como un datagrama IP independiente. Cuando los datagramas que contienen datos TCP llegan a una máquina, se pasan a la entidad TCP, la cual reconstruye los flujos de bytes originales. (Tanenbaum y Wetherall, 2012).

2.6.2.1 El Modelo TCP/IP

El servicio TCP se obtiene al hacer que tanto el servidor como el cliente creen puntos terminales, llamados sockets. Cada socket tiene un número (dirección) que consiste en la dirección IP del host y un número de 16 bits que es local para ese host, llamado puerto. Un puerto es el nombre TCP para un Punto de Acceso al Servicio de Transporte (TSAP, por sus siglas en inglés). Para obtener el servicio TCP, hay que establecer de manera explícita una conexión entre un socket en una máquina y un socket en otra máquina.

Todas las conexiones TCP son full dúplex y de punto a punto. Full dúplex significa que el tráfico puede ir en ambas direcciones al mismo tiempo. Y de punto a punto esto significa que cada conexión tiene exactamente dos puntos terminales. Una conexión TCP es un flujo de bytes, no un flujo de mensajes (Tanenbaum y Wetherall, 2012).

2.6.3 Redes de Comunicación en la Industria

La intercomunicación de sistemas y procesos industriales es un tema elemental para cualquier MS (Sistemas de Manufactura por sus siglas en inglés), estos enlaces se han empleado en equipos de instrumentación y sistemas de automatización, donde es necesaria una baja tasa de transferencia de datos entre equipos.

Estas comunicaciones tienden a ser un factor importante para la mejora de los procesos de fabricación, la reducción de costes, el incremento en la calidad, la mejora de la eficiencia y la mayor flexibilidad en los procesos de producción.

Las comunicaciones deben poseer unas características específicas para responder a las necesidades de la comunicación en tiempo real, deben resistir a un ambiente hostil donde hay gran cantidad de ruido electromagnético y condiciones ambientales duras. En el uso de comunicaciones industriales se puede hablar de dos áreas: una comunicación a nivel de campo, y una comunicación hacia el SCADA. En estos casos la transmisión de datos se realiza en tiempo real, o por lo menos, con una mínima demora.

Es sabido que existen diferentes tipos de redes en una empresa u organización como lo son: Red de factoría, de planta, de célula y de campo, sin embargo, en este trabajo nos enfocaremos a las Redes de campo. Este tipo de red se define según (Hurtado, 2019), como “un sistema de dispositivos de campo (sensores y actuadores) y dispositivos de control, que comparten un bus digital serie bidireccional para transmitir informaciones entre ellos, sustituyendo a la convencional transmisión analógica punto a punto”. Además, permiten sustituir el cableado entre sensores, actuadores y los elementos de control, estos son de bajo costo, bajos tiempos de respuesta, permiten la transmisión en serie sobre un bus digital de datos con capacidad de interconectar controladores con todo tipo de dispositivos de entrada / salida, sencillos, y permitir controladores clientes/esclavos inteligentes.

Por otra parte, las redes de campo presentan las siguientes ventajas:

- **Flexibilidad:** Permite realizar una simple conexión eléctrica al bus y una posterior configuración/programación, normalmente remota (desde la sala de control).
- **Seguridad:** Transmisión simultánea de señales de diagnóstico de sensores y actuadores, permitiendo así instalaciones más seguras.
- **Precisión:** Transmisión totalmente digital para variables analógicas.
- **Facilidad de mantenimiento:** Facilidad en el diagnóstico del funcionamiento de un instrumento y realizar calibraciones de forma remota desde la sala de control en caso de tener falla o mal funcionamiento. Esto permite localizar rápidamente conexiones erróneas en la instalación, con lo que los errores de conexión son menores y más rápidamente solucionados (reducción de los tiempos de parada y pérdidas de producción).

MARCO TEÓRICO Y CONCEPTOS GENERALES

- **Reducción de la complejidad del sistema de control en términos de hardware:** Reducción drástica del cableado, se elimina la necesidad de grandes armarios de conexiones para el control del equipamiento asociado, reducción del número de PLCs y reducción de tiempo de instalación y personal necesario para ello.

No obstante, una de sus desventajas es la posible rotura del cable, porque, en caso de estar conectados en serie provocaría la desconexión de los dispositivos que le siguen.

2.6.3.1 Protocolos y Estándares Existentes en la Industria

En la industria existen diferentes protocolos de comunicación, en la Tabla 2.1 se muestra de manera resumida una comparación de los estándares y protocolos que utiliza.

Tabla 2.1. Estándares y protocolos más comunes en la industria. Fuente: (Hurtado, 2019).

PROTOCOLO / ESTANDAR	TOPOLOGÍA	MEDIO FÍSICO	VELOCIDAD	DISTANCIA SEGMENTO	NODOS POR	ACCESO AL MEDIO
P-NET	Anillo	Par trenzado apantallado	76.8 kbps	1.2 km	125	Paso de testigo Maestro/esclavo
PROFIBUS	Bus lineal Anillo Estrella Árbol	Par trenzado apantallado Fibra óptica	Hasta 12Mbps	Hasta 9.6 Km y 90 Km	125	Paso de testigo Maestro/esclavo
WORLDFIP	Bus lineal	Par trenzado apantallado Fibra óptica	Hasta 1 Mbps y 5Mbps	Hasta 5 Km y 20 Km	64	Arbitro de bus
HART	Bus lineal	Cable 2 hilos	1.2 kbps	3.000 m	30	Maestro/esclavo
MODBUS RTU	Bus lineal	Par trenzado	Hasta 19.2Kbps	1 km	248	Maestro/esclavo
MODBUS TCP/IP	Estrella	Par trenzado	10/100 Base-T (10/100 Mb/s)	1.2 km	255	Cliente/Servidor
INTERBUS-S	Anillo	Par trenzado	500 kbps	400 m	256	Paso de testigo
BITBUS	Bus lineal	Par trenzado Fibra óptica	Hasta 1.5Mbps	Hasta 1.200m	29	Maestro/esclavo

CAPÍTULO 2

CAN	Bus lineal	Par trenzado	Hasta 1 Mbps	Hasta 1.000m	127-64	CSMA/CD con arbitraje de bit
SDS	Bus lineal	Cable de 4 hilos	Hasta 1 Mbps	500 m	64	CSMA
DEVICENET	Bus lineal	Par trenzado	Hasta 500 Kbps	Hasta 500 m	64	CSMA/CDBA
CONTROLNET	Bus lineal Árbol Estrella	Coaxial Fibra óptica	5 Mbps	Hasta 3.000m	48	CTDMA
SERIPLEX	Bus lineal	Cable 4 hilos apantallado	98 kbps	1.500m	300	Maestro/esclavo
AS-i	Bus lineal Árbol - Estrella	Cable 2 hilos	167 kbps	Hasta 200 m	32-62	Maestro/esclavo
LON WORKS	Bus Anillo Libre	Par trenzado Fibra óptica Red eléctrica Coaxial Radio Infrarrojos	Hasta 1.25 Mbps	Hasta 2.700 m	64	CSMA/CA
ARCNET	Bus Estrella	Par trenzado Fibra óptica Coaxial	2.5 Mbps	122 m	255	Paso de testigo
M-BUS	Bus lineal	Cable 2 hilos	Hasta 9.6 Kbps	1.000 m	250	Arbitro de bus
UNI-TELWAY	Bus lineal	Par trenzado apantallado	Hasta 19.2Kbps	20 m	Hasta 28	Maestro/esclavo
COMPOBUS/S	Bus lineal	Cable de 2 o 4 hilos	Hasta 750 Kbps	Hasta 500 m	32	Maestro/esclavo

Los protocolos de comunicación que se muestran en la Tabla 2.1 son algunos de los estándares más comunes utilizados en la industria, con los que se define la forma de conexión y comunicación entre los diferentes dispositivos de la red de automatización, con esto se permite agregar o quitar fácilmente elementos de la red, o reproducir el sistema, además de establecer la forma en la que se envía la información.

Del total de los protocolos identificados y revisados en la literatura, el proyecto aquí descrito hace uso del protocolo Modbus TCP/IPv4, más información al respecto se muestra en el siguiente tema.

2.6.3.2 Protocolo Modbus

Modbus es un protocolo para la comunicación, publicado por Modicon en 1979, diseñado para funcionar con equipos industriales tales como Controladores Lógicos Programables (PLCs, por sus siglas en inglés), computadoras, motores, sensores y otros tipos de dispositivos físicos de entrada/salida. Es un protocolo estándar de comunicación, siendo uno de los más comunes en la industria para la comunicación de dispositivos electrónicos. (Sánchez, 2012)

A esto, existen diferentes variantes del protocolo, mismas que se mencionan a continuación:

- **Modbus RTU:** Es la implementación más común para Modbus, utiliza la comunicación en serie y hace uso de una representación binaria compacta de los datos para el protocolo de comunicación. El formato de Supervisión con una Unidad Remota (RTU, por sus siglas en inglés), sigue a los comandos/datos con una suma de Comprobación de Redundancia cíclica (CRC, por sus siglas en inglés) como un mecanismo de comprobación de errores para garantizar la fiabilidad de los datos. Un mensaje Modbus RTU debe transmitirse continuamente sin vacilaciones entre caracteres.
- **Modbus ASCII:** Se utiliza en la comunicación en serie y hace uso de caracteres ASCII para el protocolo de comunicación. El formato de Código Estándar Estadounidense para el Intercambio de Información (ASCII, por sus siglas en inglés), utiliza un checksum de Control de Redundancia Longitudinal (LRC, por sus siglas en inglés).
- **Modbus TCP/IP o Modbus TCP:** Se trata de una variante Modbus utilizada para comunicaciones a través de redes TCP/IP, conectándose a través del puerto 502. No requiere un cálculo de suma de verificación (checksum), ya que las capas inferiores ya proporcionan protección de checksum. En esta hay dos versiones la TCP/IPv4 en la que se utiliza el direccionamiento IP más común (192.168.0.1, por decir un ejemplo) y TCP/IPv6 ocupando el nuevo direccionamiento IP en el que

está pensado para admitir más dispositivos en la red (2001:0db8:85a3:08d3:1319:8a2e:0370:7334, por decir un ejemplo).

- **Modbus sobre TCP/IP o Modbus sobre TCP o Modbus RTU/IP:** Esta es una variante de Modbus que difiere del Modbus TCP en que se incluye una suma de comprobación en la carga útil como en Modbus RTU.
- **Modbus sobre UDP:** Algunos han experimentado con el uso de Modbus sobre el Protocolo de Datagramas de Usuario (UDP, por sus siglas en inglés) en redes IP, lo que elimina los gastos generales necesarios para TCP.
- **Modbus Plus (Modbus+, MB+ o MBP):** Es una versión extendida del protocolo y privativa de Schneider Electric y a diferencia de las otras variantes, soporta comunicaciones peer to peer entre múltiples masters. Requiere un coprocesador dedicado para manejar Control de Enlace de Datos de Alto Nivel (HDLC, por sus siglas en inglés). Se requiere hardware especial para conectar Modbus Plus a un ordenador, normalmente una tarjeta diseñada para bus de Estándar de Arquitectura Industrial (ISA, por sus siglas en inglés), Interconexión de Componentes Periféricos (PCI, por sus siglas en inglés) o Asociación Internacional de Tarjetas de Memoria para Ordenadores Personales (PCMCIA, por sus siglas en inglés).
- **Pemex Modbus:** Esta es una extensión de Modbus estándar con soporte para datos históricos y de flujo. Fue diseñado para la compañía petrolera Petróleos Mexicanos (Pemex) para su uso en el control de procesos y nunca alcanzó un uso generalizado.
- **Enron Modbus:** Esta es otra extensión del estándar Modbus desarrollada por Enron Corporation con soporte para variables enteras de 32 bits y de punto flotante y datos históricos y de flujo. Los datos históricos cumplen con un estándar de la industria del Instituto Americano de Petróleo (API, por sus siglas en inglés) según la forma en que deben almacenarse los datos.

Por lo general se conecta a través del puerto 502, no requiere un cálculo de suma de verificación (checksum en inglés), ya que las capas inferiores ya proporcionan protección de checksum. Una suma de verificación es una función hash que tiene el propósito de detectar cambios accidentales en una secuencia de datos para proteger la integridad.

La especificación Modbus/TCP define un estándar interoperable en el campo de la automatización industrial, el cual es simple de implementar para cualquier dispositivo que soporta sockets TCP/IP, basado en la arquitectura Cliente/Servidor. La información antes descrita fue tomada de la referencia (Modbus Organization, 2005-2019).

2.7 Sistemas de Eventos Discretos

Un sistema es un objeto complejo cuyas partes o componentes se relacionan con otros componentes (Bunge, 1999) y un evento discreto es una acción que ocurre sólo una vez en el tiempo por lo cual se vuelve única en el sistema y puede ocasionar que el estado del sistema cambie. Estos sistemas se pueden encontrar en la vida cotidiana de una persona o en instarías, un ejemplo de esto último es cuando ocurre un accidente de derramar café sobre la computadora, estos sucesos no son controlados y pueden ocasionar cambios en la rutina de la persona.

2.7.1 Simulación de Eventos Discretos

La simulación es la representación de algún proceso o sistema real a través del tiempo, ya sea hecha manualmente o en una computadora, la simulación involucra la generación de un historial artificial de un sistema y su observación para obtener inferencias relacionadas con las características operativas del sistema real (Banks, Carson, Nelson, y Nicol, 2005). La simulación inicia con un modelo, estos, son una descripción física o matemática de un sistema, objeto o evento y usualmente representa un punto particular de acción en el tiempo (Beaverstock, Greenwood, Lavery, y Nordgren, 2012). Asimismo, los modelos son empleados para estudiar y determinar la representación de un sistema real de manera abstracta con la intención de predecir el comportamiento del mismo. Por lo que la simulación de eventos discretos es una técnica informática de modelado dinámico de sistemas. Se caracteriza por un control en la variable del tiempo que permite controlar el rango de evolución de eventos con referente a modelos propuestos para ver su desarrollo y efectividad del modelo a futuro. Un requisito para aplicar esta técnica es

que las variables que definen el sistema no cambien su comportamiento durante el intervalo simulado, pero que aceptan cambios como fallas.

2.7.2 Software FlexSim

Para visualizar el desarrollo de las operaciones de un modelo de simulación de eventos discretos, frecuentemente se hace uso de un software de simulación 3D. Algunos ejemplos de software hallados en la literatura son: ProModel, Arena, Simio, Witness, FlexSim, entre otros.

El software FlexSim cuyo logo se muestra en la Figura 2.3, fue desarrollado por Bill Nordgren, Cliff King, Roger Hullinger, Eamonn Lavery y Anthony Johnson. FlexSim es un software de simulación de eventos discretos, permite modelar y entender con precisión los problemas básicos de un sistema sin la necesidad de programaciones complicadas, esto debido a que ofrece una forma sencilla al desarrollar el modelo de simulación. Frecuentemente es utilizado en temas como:

- Fabricación: producción, línea de montaje, tienda de trabajo, etc.
- Manejo de materiales: sistemas de transportador, vehículo de guiado automático (AGV, por sus siglas en inglés), embalaje y almacenamiento.
- Logística y distribución: la operación terminal de contenedores, diseño de la cadena de suministro, diseño de flujo, servicio y almacenamiento de trabajo, centros distribución, etc.
- Transporte: tráfico (sistema de carretera), tránsito de flujo peatonal de la estación, nave marítima coordinación, congestión de tráfico personalizado, etc.



Figura 2.6. Logotipo de la empresa FlexSim. Fuente: (FlexSim Software Products, 2019).

2.8 Controlador Lógico Programable

Un Controlador Lógico Programable es conocido como un autómata programable, es una computadora utilizada para automatizar procesos electromecánicos, tales como el control de la maquinaria de la fábrica en líneas de montaje o atracciones mecánicas. (Laughton y Warne, 2003).

Los PLCs están diseñados para múltiples señales de entrada y de salida, rangos de temperatura ampliados, inmunidad al ruido eléctrico y resistencia a la vibración y al impacto, por su uso en la industria con maquinaria pesada. Los programas para el control de funcionamiento de la máquina se suelen almacenar en baterías, copia de seguridad o en memorias no volátiles. Un PLC es un ejemplo de un sistema de tiempo real, donde los resultados de salida deben ser producidos en respuesta a las condiciones de entrada dentro de un tiempo limitado, de lo contrario no producirá el resultado deseado.

Hay diversas compañías dedicadas a la automatización que ofrecen diferentes modelos de PLCs, ejemplos de las mismas son: Allen Bradley, Talvent, Emerson, Invensys, Siemens, entre otras. Las Figuras 2.6 y 2.7 muestran los modelos Micro820 y S7-1200 de las empresas Allen-Bradley y Siemens, respectivamente.



Figura 2.7. PLC Allen-Bradley Micro820. Fuente: (Automation, 2018).



Figura 2.8. PLC Siemens S7-1200. Fuente: (Siemens, 2019).

Cabe destacar que la presente propuesta de investigación hace uso del modelo Allen Bradley Micro 820 para demostrar la comunicación entre un sistema virtual y uno físico. Más detalles de este PLC se exponen en el siguiente capítulo.

CAPÍTULO 3. METODOLOGÍA Y CONFIGURACIÓN

3.1 Metodología: Arquitectura de Dos Capas

3.2 Propuesta de Investigación

3.3 Componentes del Proyecto

3.3.1 Controlador en Tiempo Real/PLC

3.3.1.1 Variables de Red

3.3.1.2 Protocolo TCP/IP

3.3.2 Configuración de PLC

3.3.3 Componentes Físicos

3.3.3.1 Configuración del Software FlexSim

3.3.3.2 Interfaz de Control Físico

3.1 Metodología: Arquitectura de Dos Capas

En la referencia (Thomas y Alois, 2015), se dan los pilares preliminares bajo los cuales se sustenta la presente investigación. Como puede observarse, la Figura 3.1 se categoriza en dos áreas, la primera incluye a los sistemas para la Supervisión, Control y Adquisición de Datos con Interfaz Hombre Maquina para los Sistemas de Ejecución de Fabricación (SCADA/HMI/MES, por siglas en inglés), mientras que en la segunda se describe la estructura de un Agente Industrial. Un Agente Industrial se constituye bajo la integración de tres niveles los cuales son sub áreas del mismo; Sistemas de Múltiples Agentes (MAS, por sus siglas en inglés), Controlador en Tiempo Real/PLC y Componentes físicos, las dos primeras sub áreas también se pueden referenciar como la Capa de Control de Agentes y la Capa de Control Reactiva (RCL, por sus siglas en inglés).

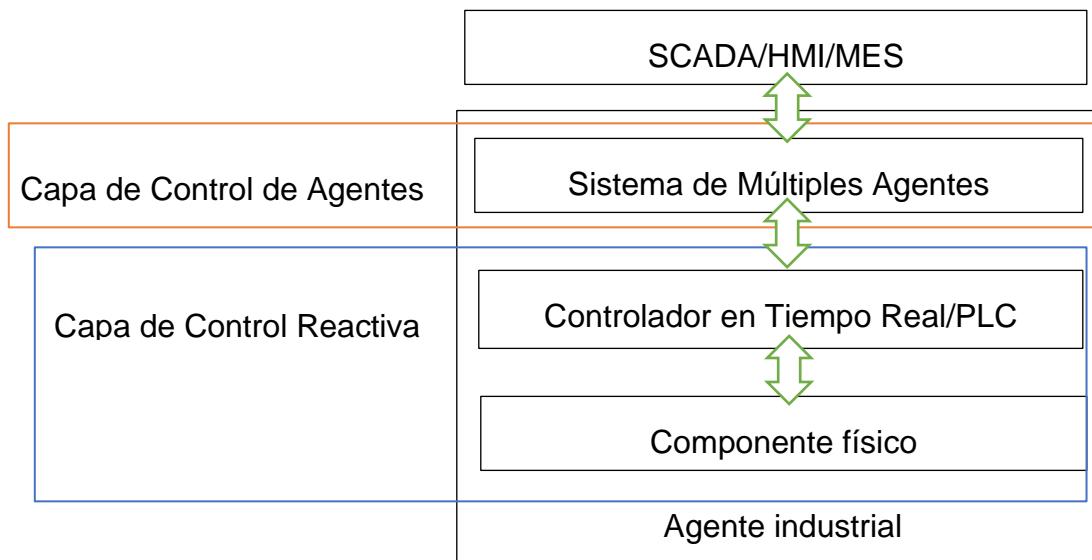


Figura 3.1. Arquitectura de agentes de dos capas. Fuentes: Christensen, 2003 y Leitão, 2009.

El Sistema de Control de Agentes, expande los dominios de aplicación potenciales de un MAS y al mismo tiempo, agrega la flexibilidad requerida a los Sistemas de Control y Medición de Procesos Industriales (IPMCS, por sus siglas en inglés). MAS es responsable de las tareas que implican control estratégico de nivel superior, mismas que

consisten en la planificación de la ejecución, la cooperación y la negociación con otros subsistemas, y el control de supervisión particular de la sección del IPMCS asociado.

Por otro lado, la RCL, como su nombre lo indica, se encarga de proporcionar la operación de control en tiempo real para la parte del IPMCS. Normalmente, los programas de control de RCL se ejecutan en pequeños dispositivos de control integrados en el IPMCS. Estos dispositivos de control están conectados a sensores para conocer el estado de IPMCS y de actuadores que lo influyen. Las tareas principales del RCL son: 1) la lectura de los valores de los sensores a partir de las señales de entrada, 2) el preprocesamiento de estos valores (según los estados actuales y las leyes de control de cálculo del modo de ejecución) y 3) la generación de valores apropiados para los actuadores conectados a las salidas de los dispositivos de control.

La clave en dicha arquitectura es la interfaz de interacción entre el MAS y el RCL. Desde aquí, el MAS debe poder solicitar servicios del RCL (por ejemplo, realizar ciertas operaciones de maquinado, etc.) y obtener retroalimentación del RCL sobre el estado de ejecución de los servicios (por ejemplo, tiempo restante, calidad acabada, entre otras). Además, se debe permitir que el MAS cambie los parámetros del RCL (por ejemplo, ajuste el RCL a los modos de ejecución o los productos que se producen). Finalmente, el MAS requiere retroalimentación de estado del RCL. Esta retroalimentación de estado debe ser iniciada por el RCL y los puertos en el estado general de la parte IPMCS controlada (por ejemplo, el agotamiento de suministros), especialmente en condiciones críticas (por ejemplo, fallos inesperados en los equipos de producción).

Por lo tanto, la RCL tiene que proporcionar servicios que le permitan al MAS reconfigurar y cambiar el programa de control de la RCL de acuerdo con las necesidades actuales de los planes del MAS.

A esto, es necesario puntualizar que, al diseñar un sistema de dos niveles, se deben tener en cuenta los requisitos de seguridad y las condiciones de protección para la interfaz de interacción. Así como también es necesario seleccionar tecnología apropiada para la RCL, así como protocolos de comunicación adecuados (en el caso particular,

Modbus TCP/IPv4). Normalmente, solo un agente interactuará directamente con la RCL, ya que este agente representa la parte de control de IPMCS en el MAS. Por otro lado, es importante hacer notar que la interacción entre MAS y RCL no perturba al comportamiento del RCL, de tal forma que el mismo no pueda cumplir con sus restricciones en tiempo real y, por lo tanto, no permita ejercer control de manera segura sobre el IPMCS asociado. Cabe destacar que, generalmente el MAS y RCL, deben estar acoplados de manera flexible, de modo que el RCL también pueda funcionar sin un sistema de agentes.

Resumiendo, en sistemas y entornos industriales, la RCL proporciona al sistema de agentes las restricciones de ejecuciones necesarias en tiempo real para la aplicación de control. La RCL acepta comandos y parámetros desde el MAS y le envía devuelta información de su estado. Una funcionalidad clave para sistemas totalmente flexibles y adaptativos es que el RCL proporciona servicios y funciones que permiten al MAS reconfigurar las aplicaciones de control por tiempo limitado.

En otro orden de ideas, para lograr la interoperabilidad sería necesaria una forma estandarizada de intercambio de información entre ambas capas. La estandarización dada por la Comisión Internacional de Electrotécnica (IEC, por sus siglas en inglés), normaliza el diseño de hardware, instalación, pruebas, documentación, programación y comunicación de los PLCs. A ello, existen dos normas utilizadas para el control en tiempo real, estas son: la IEC 61131 para Controladores Lógicos Programables (IEC, 2012a) y la IEC 61499 para la Automatización Distribuida (IEC, 2012b).

Después de puntualizar los niveles e interacción de la arquitectura citada, el siguiente apartado describe los componentes, protocolos y herramientas empleadas durante el desarrollo del proyecto.

3.2 Propuesta de Investigación

La investigación propuesta hace uso de la arquitectura de agentes de dos capas, sin embargo, es centrada en la Capa de Control Reactiva (Controlador en tiempo real/PLC) y en el control ejercido sobre el componente físico. Para representar los componentes físicos se consideran: 1) un emulador 3D (software FlexSim) que represente al sistema real y 2) una interfaz de control físico o botonera. Cabe destacar que el presente documento se enfoca hacia la capa de control reactiva.

3.3 Componentes del Proyecto

En esta sección se detallará de cada componente del proyecto, así como de su configuración básica para la interacción de los componentes entre sí.

3.3.1 Controlador en Tiempo Real/PLC

El Controlador Lógico Programable utilizado en el presente trabajo es un Micro PLC de 20 puntos Micro820 (2080-LC20-20QWB) de la marca Allen-Bradley, mismo que fue diseñado para pequeñas máquinas independientes y proyectos remotos de automatización con puertos incorporados de Ethernet y serie. Puede funcionar como una RTU (Unidad Terminal Remota) para máquinas remotas con compatibilidad con RTU y TCP Modbus. Tiene soporte integrado para 4 entradas de temperatura de termistor para su uso como un DDC (controlador digital directo) para sistemas de gestión de edificios. La Tabla 3.1 muestra las especificaciones detalladas del modelo utilizado.

Tabla 3.1. Especificaciones de Micro 820 Allen Bradley. Fuente: (Automation, 2018).

ATRIBUTO	VALOR
Para uso con	Bulletin 2080
Número de E/S	20
Número de entradas	12
Tipo de entrada	Analógica, Dc
Categoría de tensión	24 V dc

Tipo de salida	Relé
Interfaz	Ethernet
Número de salidas	8
Tipo de red	Ethernet
Tipo de puerto de comunicación	RJ45, RS232, RS485
Número de puertos de comunicación	3
Dimensiones	90 x 104 x 75 mm
Temperatura de funcionamiento máxima	+65°C
Tipo de montaje	Carril DIN
Entradas/Salidas máximas	12/8
Cumplimiento normativo	Certificación para ubicaciones peligrosas de Norteamérica, IEC 60529, NEMA 250
Longitud	90mm
Anchura	104mm
Profundidad	75mm
Temperatura de funcionamiento mínima	-20°C

Para darle fundamento al sistema de control de programación se hace uso de la norma IEC 61131 antes mencionada, específicamente para el tema de programación se utilizó la sección 3 de la misma (IEC 61131-3), la cual se refiere al control distribuido de procesos industriales empleando el lenguaje gráfico Diagramas Ladder (LD, por sus siglas en inglés), que a su vez permite una configuración cíclica. Aquí, la investigación se enfoca, principalmente en el uso de un sistema PLC que consta de un solo controlador.

METODOLOGÍA Y CONFIGURACIÓN

Es importante hacer notar que la IEC61131-3 define cuatro lenguajes de programación, dos gráficos y dos textuales, que abarcan diferentes aspectos de la configuración del PLC. Estos lenguajes se clasifican en dos grupos, los cuales son:

- Alto nivel: son lenguajes gráficos, como Diagramas Ladder o de escalera (LD, por sus siglas en inglés) y Diagramas de Funciones en Bloques (FBD, por sus siglas en inglés).
- Bajo nivel: son lenguajes textuales, como Lista de Instrucciones (IL, por sus siglas en inglés) y Texto Estructurado (ST, por sus siglas en inglés).

El diagrama de escalera (LD) representa los diagramas de cableado utilizados para describir la lógica de relé (dispositivo electromagnético). Es ideal para tareas de control pequeñas y para especificar interbloqueos. Por otro lado, los diagramas de FBD permiten a los usuarios colocar y conectar gráficamente los Bloques de Función (FB, por sus siglas en inglés), estos últimos se parecen a los diagramas lógicos utilizados en el diseño lógico digital.

Los lenguajes textuales son la lista de instrucciones y el texto estructurado. Una lista de instrucciones es un lenguaje tipo ensamblador que permite a los usuarios especificar tareas de control a un nivel muy bajo. El texto estructurado es un lenguaje imperativo de alto nivel con una sintaxis similar a Pascal o Ada, es adecuado para la programación de algoritmos complejos, como los algoritmos de control de bucle cerrado.

Después de las consideraciones anteriores, es importante mencionar que el soporte de comunicación de los dispositivos de control basados en IEC 61131-3 se agrupan en tres principales categorías, estas son:

- Variables de red
- FB de comunicación
- Interacción con sistemas de alto nivel

Dada la necesidad del modelo propuesto, en lo subsiguiente se abordarán conceptos exclusivos a variables de red, protocolo de comunicación, la configuración del PLC y obtención de datos con un software desarrollado en Java.

3.3.1.1 Variables de Red

Las variables de red, que no están explícitamente definidas en IEC 61131-3, son una tensión lógica y específica del proveedor del modelo de configuración estrechamente acoplado a través de la red, formando configuraciones acopladas de forma suelta. Eso significa que un conjunto de variables globales, son llamadas variables de red, se replica a través de la red en todos los recursos involucrados.

Cuando un programa en un dispositivo escribe en una de las variables, la infraestructura de comunicación actualiza esta variable en todos los demás dispositivos. Dado que varios programas pueden escribir al mismo tiempo, se introducen diferentes protecciones de acceso (es decir, escritura, lectura/escritura, solo lectura, escritura solamente), lo que ayuda a garantizar la coherencia. Actualmente, no existe un estándar o enfoque común para las variables de red. El propósito principal de las variables de red es la comunicación horizontal entre los dispositivos en línea. Por lo tanto, este medio de comunicación no es adecuado para la interacción RCL y MAS. Para este propósito, en el documento se muestra el uso del protocolo Modbus TCP/IP y LD.

3.3.1.2 Protocolo de Comunicación TCP/IP

Modbus es un protocolo de comunicaciones publicado por Modicon (véase capítulo dos), el cual no tiene estado, es decir, cada solicitud del servidor es tratada independientemente por el cliente y es considerada una nueva solicitud no relacionada a las anteriores, esto hace que las transacciones de datos sean resistentes a fallos debido a ruido.

Al utilizar el protocolo TCP orientado a conexión, se permite el control de una transacción individual encapsulándola en una conexión la cual pueda ser identificada, supervisada, y cancelada sin requerir acción específica de parte de las aplicaciones cliente y servidor. Esto da una amplia tolerancia a cambios del desempeño de la red, y permite que herramientas de seguridad tal como firewalls y proxys puedan ser fácilmente añadidas para aumentar la seguridad en la red.

En adición, TCP permite establecer un gran número de conexiones concurrentes, de este modo el cliente puede utilizar varias veces una conexión previamente establecida o crear una nueva, en el momento de realizar una transacción de datos.

De igual forma, el protocolo Modbus define una Unidad de Datos de Protocolo (PDU, por sus siglas en inglés) simple, independiente de las capas de comunicación subyacentes. La trama del protocolo MODBUS puede introducir algunos campos adicionales en la Unidad de Datos de la Aplicación (ADU, por sus siglas en inglés). Véase la Figura 3.2.

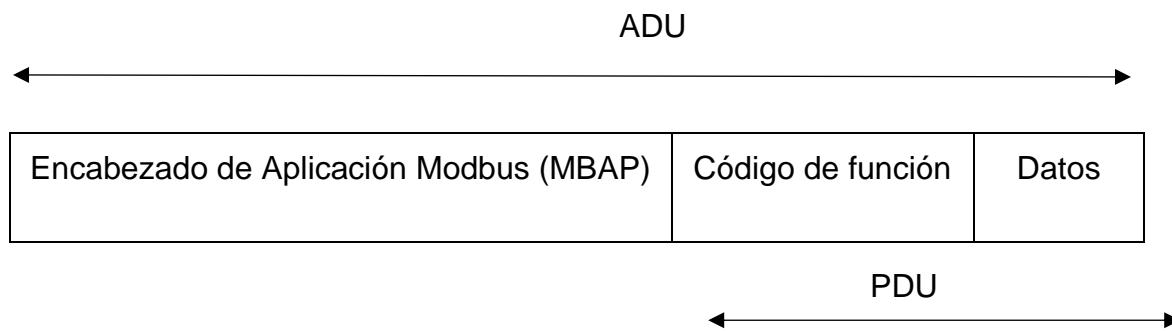


Figura 3.2. Encapsulamiento de mensaje en Modbus TCP/IP. Fuente: (Modbus Organization, 2005-2019).

El encabezado (MBAP), es dedicado en TCP/IP para identificar la Unidad de datos de la aplicación Modbus. Así, el encabezado tiene cuatro campos que servirán para la comunicación (Tabla 3.2).

Tabla 3.2. Elementos del encabezado MBAP. Fuente: (Modbus Organization, 2005-2019).

CAMPOS	LONGITUD	DESCRIPCIÓN
Identificador de transacción	2 bytes	Identificación de una Solicitud Modbus / Transacción de respuesta. Se utiliza para el emparejamiento de transacciones, el servidor Modbus se copia en la respuesta el identificador de transacción de la solicitud.
Identificador de protocolo	2 bytes	Su utiliza 0 como identificador de que se está utilizando el protocolo Modbus TCP/IP.
Longitud	2 bytes	Es el número del conteo de los siguientes bytes que continúan en la trama enviada.
Identificador de la unidad	1 byte	Identificador de un esclavo remoto conectado en serie.

Todas las solicitudes y respuestas Modbus están diseñadas de tal forma que el receptor pueda verificar que un mensaje está completo, esto es verificado en la sección de longitud MBAP con la que es posible contar la porción de datos que debe de continuar, de esta forma es posible saber si la trama está completa o existe la ocurrencia de un error.

Cuando Modbus utiliza el TCP, la información de longitud es agregada en el prefijo (o encabezado) para permitir al receptor reconocer los límites del mensaje, incluso si el mensaje ha sido dividido en múltiples paquetes para la transmisión. La existencia de reglas de longitud implícitas o explícitas, y el uso de un código de chequeo de error (en Ethernet) resulta en una probabilidad muy pequeña de corrupción no detectada sobre un mensaje de solicitud o respuesta.

Continuando con la descripción de la Figura 3.2, dentro del módulo “Código de función”, son asignadas una serie de identificadores que tienen un comportamiento específico que los clientes pueden implementar de manera flexible en base al comportamiento de la aplicación deseada. Así, la especificación de la PDU define conceptos básicos para el

METODOLOGÍA Y CONFIGURACIÓN

acceso y manipulación de datos; sin embargo, un cliente puede manejar datos de una manera que no esté definida explícitamente en la especificación.

Por otro lado, los datos disponibles por medio de Modbus son almacenados, en general, en uno de los cuatro bancos de datos o rangos de dirección o tablas: bobinas (Coils), entradas discretas (Discrete Input), registros de retención (Holding Register) y registros de entrada (Input Registers). Estos bancos de datos definen el tipo y los derechos de acceso de los datos contenidos. Los dispositivos clientes tienen acceso directo a estos datos, los cuales son alojados localmente en los dispositivos. Los datos disponibles por medio de Modbus generalmente son un subconjunto de la memoria principal del dispositivo. En contraste, los maestros Modbus deben solicitar el acceso a estos datos a través de diversos códigos de función. El comportamiento de los bloques se describe en la Tabla 3.3.

Tabla 3.3. Bloques de memoria. Fuente: (Modbus Organization, 2005-2019).

BLOQUE DE MEMORIA	TIPO DE DATOS	ACCESO DE SERVIDOR	ACCESO DE CLIENTE	ACCIONES
Bobinas	Booleano	Lectura/Escritura	Lectura/Escritura	Leer y escribir
Entradas Discretas	Booleano	Solo Lectura	Lectura/Escritura	Solo lectura
Registros de Entrada	Palabra sin signo	Solo Lectura	Lectura/Escritura	Solo lectura
Registros de Retención	Palabra sin signo	Lectura/Escritura	Lectura/Escritura	Leer y escribir

Los códigos de funciones públicas según (Modbus Organization, 2005-2019) realizan diferentes acciones con los bloques de memoria, estas se muestran en la Tabla 3.4.

Tabla 3.4. Códigos de funciones. Fuente: (Modbus Organization, 2005-2019).

FUNCIONES	CÓDIGO DECIMAL	CÓDIGO HEXADECIMAL
Leer bobinas	01	01
Leer entradas discretas	02	02
Leer registros de retención	03	03
Leer registros de entrada	04	04
Escribir bobina simple	05	05
Escribir un registro simple	06	06
Leer estatus de excepción	07	07
Diagnóstico	08	08
Obtener contador de eventos Com	11	0B
Obtener registro de eventos Com	12	0C
Escribir múltiples bobinas	15	0F
Escribir múltiples registros	16	10
ID del servidor de reportes	17	11
Leer el archivo de registro	20	14
Escribir el archivo de registro	21	15
Registro de escritura de máscara	22	16
Leer/Escribir múltiples registros	23	17
Leer la cola FIFO	24	18
Leer identificación del dispositivo	43	2B
Transporte de interfaz encapsulado	43	2B
Puede abrir referencia general	43	2B

De las funciones mostradas en la tabla anterior, dentro de las más comunes podemos listar a las primeras cuatro:

- **Código 01 (Leer bobinas):** Se utiliza para leer los estados de las salidas de los registros del 1 al 1999.

- **Código 02 (Leer entradas discretas):** Se utiliza para leer los estados de las entradas discretas de los registros del 1 al 1999.
- **Código 03 (Leer registros de retención):** Se utiliza para leer el contenido de una posición de memoria, este tipo de memorias son de lectura y escritura. Es quizás la función más importante del Protocolo Modbus pues por medio de estos registros se puede obtener información de todo lo que ocurre en el equipo, es solamente mover la información que se quiere compartir al registro de retención.
- **Código 04 (Leer registros de entrada):** Se utiliza para leer el contenido de una posición de memoria, este tipo de memorias son de solo lectura. Muchos equipos los utilizan para compartir información de sus entradas analógicas en caso de que las posean.

Dado lo anterior, la Tabla 3.5, muestra la posición de los elementos que conforman una trama del protocolo Modbus TCP/IP, así como también de la longitud en bytes de cada campo, cabe mencionar que está representado en hexadecimal.

Tabla 3.5. Composición de una trama Modbus TCP/IP.

IDENTIFICADOR DE TRANSACCIÓN		IDENTIFICADOR DE PROTOCOLO		LONGITUD		IDENTIFICADOR DE UNIDAD	CÓDIGO DE FUNCIÓN	DATOS		
2 bytes		2 bytes		2 bytes		1 byte	1 byte	Varios (4, 6, ...)		
1	0	0	0	0	5	22	4	2	0	35

3.3.2 Configuración del PLC

Para desarrollar la configuración del PLC Micro820 de Allen-Bradley, se hizo uso del software Interfaz Hombre Máquina. El mismo, permite programar al controlador y desarrollar su integración con el editor HMI (Interfaz Hombre Máquina, por sus siglas en inglés), fue desarrollado con base en la tecnología comprobada de Rockwell Automation y Microsoft Visual Studio. Una imagen del mismo puede verse en la Figura 3.3.



Figura 3.3. PLC Micro820 de Allen-Bradley.

Para desarrollar la programación del PLC dentro del software antes mencionado fue necesario establecer una serie de pasos ordenados:

1) Creación de un nuevo proyecto. Para dar de alta un nuevo proyecto es necesario indicar el modelo de controlador a configurar, para nuestro caso se hará uso del modelo 2080-LC20-20QWB tal como se muestra en la Figura 3.4. Cabe destacar que el PLC con la serie Micro 820 de Allen Bradley cuenta con tres modelos diferentes.

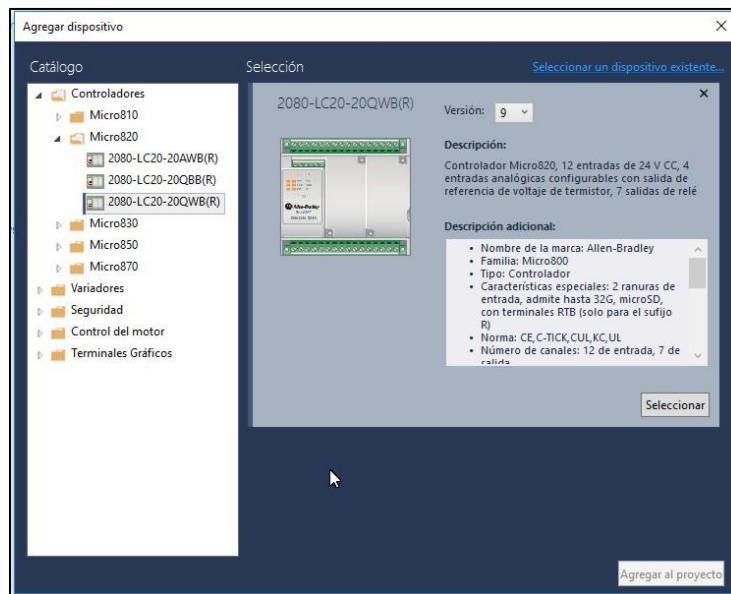


Figura 3.4. Pantalla para seleccionar el modelo de PLC en Interfaz Hombre Máquina.

2) **Selección de lenguaje.** Posterior a la creación del proyecto, en la sección de programas se elige según las necesidades de entre tres lenguajes que soporta el PLC (ST, LD y FBD), en el caso particular se hizo uso del lenguaje LD, mismo que se rige por la norma IEC 61131-3 (Figura 3.5).

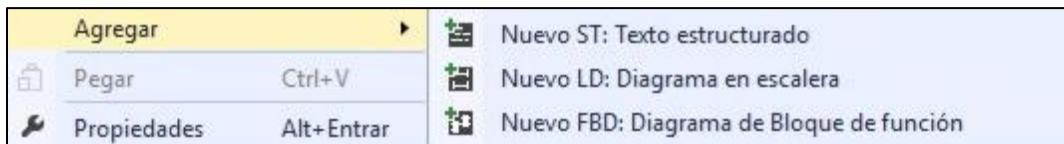
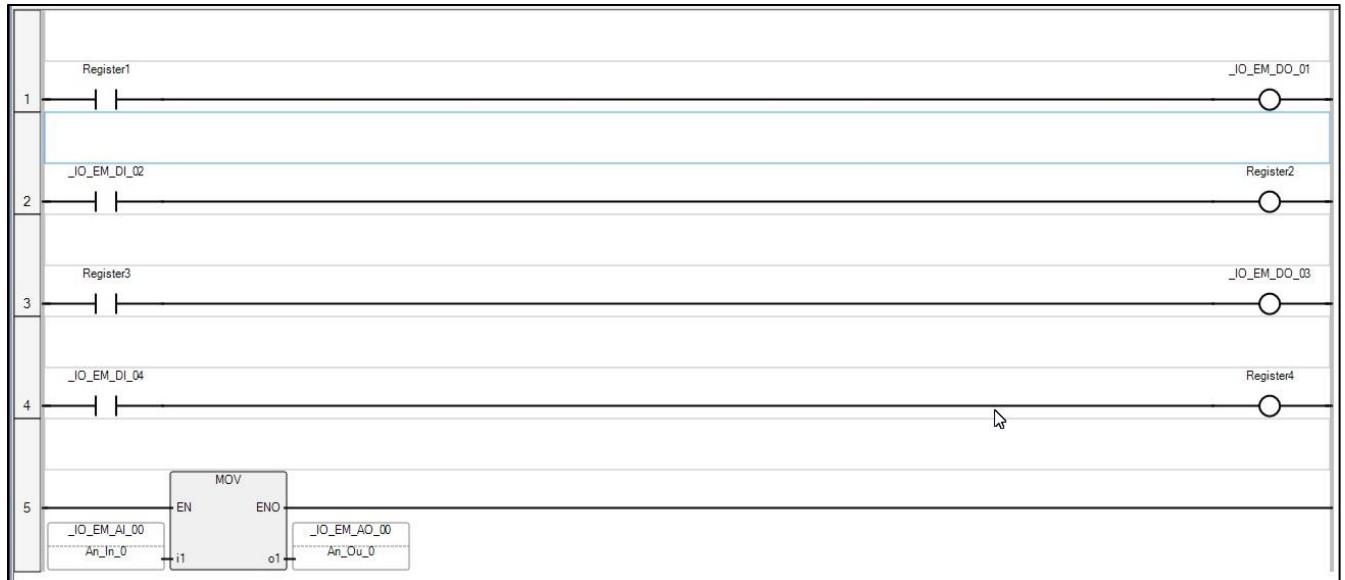


Figura 3.5. Lenguajes soportados en el PLC.

3) **Programación del controlador.** Continuando con la programación de los diagramas, se utilizó los elementos básicos para poder trabajar en conjunto con los elementos físicos. El diseño de los diagramas utilizados permite el control de estados booleanos sin la necesidad de toma de decisiones como compuertas AND, OR o NOT desde la configuración del PLC. En nuestro caso, se declararon cuatro variables globales de tipo booleano (Figura 3.6), mismas que permitirán ser utilizadas en la programación, habiendo dicho lo anterior, se ocuparon también contactos abiertos y bobinas para permitir una comparación de las condiciones, cuyo resultado implica la devolución de una respuesta (Figura 3.7). Además, se configuró un contador el cual mide el voltaje que maneja el equipo, mismo que en un futuro permitirá evitar sobrecargar el controlador y evitar en lo posible quemarlo o causar un corto circuito.

Register1		BOOL	▼	TRUE	TRUE
Register2		BOOL	▼	FALSE	TRUE
Register3		BOOL	▼	FALSE	FALSE
Register4		BOOL	▼	TRUE	TRUE

Figura 3.6. Variables globales agregadas.

**Figura 3.7** Diagrama de escalera (LD) utilizado.

4) Configuración de Modbus. Para concluir la configuración y la comunicación del módulo Modbus TCP/IP, es necesario realizar actualizaciones en la pantalla principal “Asignación de Modbus”, es decir, fue necesario agregar la conexión de las cuatro variables globales antes declaradas con las direcciones de bloques que maneja Modbus (Figura 3.8).

**Figura 3.8.** Configuración de Modbus en PLC.

Una vez configurado el PLC será necesario conectar el mismo con los componentes del sistema físico, a ello el siguiente apartado explica su desarrollo.

3.3.3 Componentes Físicos

Para representar la capa de control reactiva y los componentes físicos definidos en el modelo correspondiente a la Figura 3.1 se utilizó el software FlexSim y el PLC Micro820.

3.3.3.1 Configuración del Software FlexSim

En intención de emular el comportamiento de elementos físicos de un sistema de manufactura real se hace uso de la herramienta FlexSim, misma que mediante la configuración de diversos escenarios permitió comprobar el flujo de información entre este software y el componente físico, donde para ello la configuración lógica del PLC hizo uso del módulo Emulation que ofrece el mismo software. Emulation soporta los siguientes protocolos para la comunicación (Figura 3.9):

- OPC DA (Servidor de Acceso a Datos)
- Modbus TCP/IPv4
- Modbus TCP/IPv6
- Modbus RTU

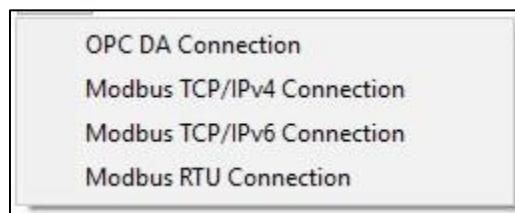


Figura 3.9. Protocolos disponibles en FlexSim.

Del total de los protocolos anteriores, la investigación fue centrada en el uso del protocolo Modbus TCP/IPv4. Se eligió el protocolo por su facilidad de implementación e interpretación de la información, además de que la herramienta permite configurar su lógica del controlador como cliente o servidor dependiendo de las necesidades de implementación.

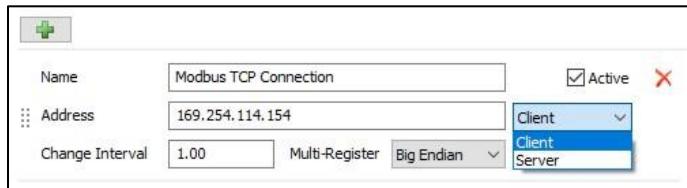


Figura 3.10. Conexión en Emulation.

La Figura 3.10 muestra el módulo “Conections” donde fue necesaria la configuración de la conexión, misma que implica proporcionar la información necesaria para comunicar al PLC con el software de simulación, donde además es posible decidir si desea que la conexión esté activa o inactiva. Si la conexión está activa, FlexSim leerá y escribirá datos directamente desde un servidor o PLC real. Si la conexión está inactiva, en su lugar extraerá datos del modelo de simulación teórica que se creó en FlexSim. También permite nombrar la conexión en caso de que tenga más de una red en el mismo simulador.

Además de lo anterior es necesario configurar la dirección IPv4 (Address) para poder iniciar la conexión con el cliente o servidor, también es necesario definir Intervalo de Frecuencia o Cambio (Change Interval) mismo que implica la frecuencia (tiempo en segundos) con la que se actualizan o refrescan datos, y el Registro Múltiple (Multi-Register) deberá permitir la recepción de más de un byte e interpretarlo correctamente.

Continuando con la declaración de variables, es necesario agregar un nuevo elemento utilizando la Conexión Modbus, se inició con la agregación sensores (Figura 3.11). Por otro lado, la herramienta ofrece diferentes configuraciones como la asignación de nombres para diferenciar un sensor de otros, es posible observar el tipo de conexión que emplea un registro (bloque al que se tendrá acceso), el número de registro es la dirección

del registro de bloque y finalmente, la asociación de objeto asocia a un elemento o recurso del modelo de simulación.

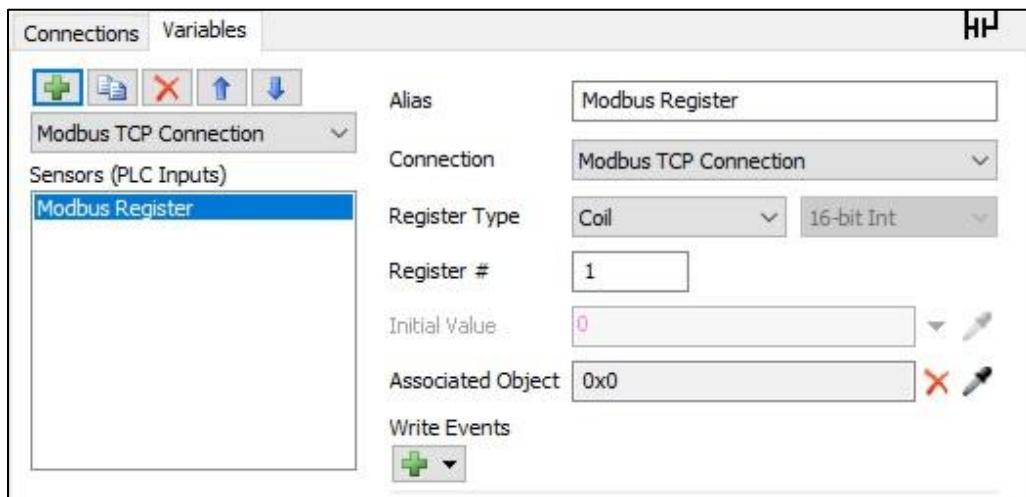


Figura 3.11. Agregando sensores para Modbus TCP/IP.

Al terminar de configurar la parte de conexiones se necesita indicar la acción o evento que tiene que realizar, algunos ejemplos son: indicar el cambio de estado de una bobina en un cierto tiempo, indicar el paso de un objeto sobre un Photo eye, entre otros (Figura 3.12).

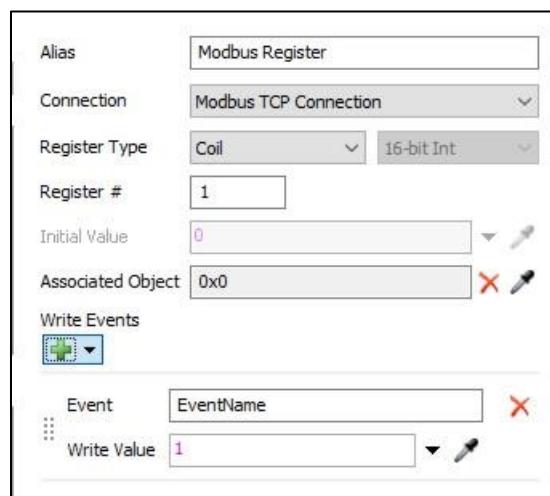


Figura 3.12. Agregando Eventos en sensores.

Continuando con los controles, la Figura 3.13 muestra la misma configuración para la comunicación que los sensores, con la diferencia de permitir cambios externos y elegir entre 24 acciones distintas.

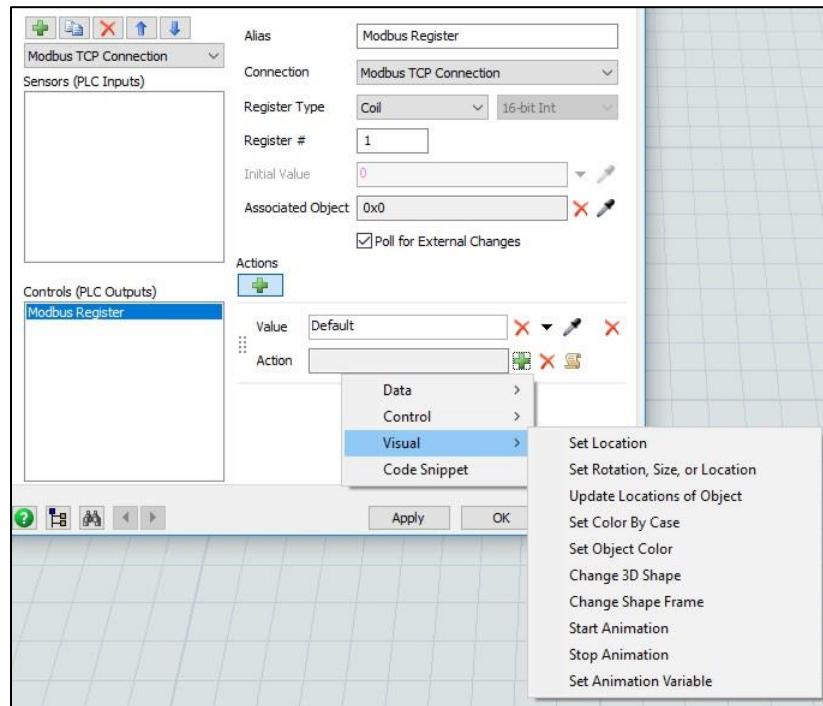


Figura 3.13. Controles en Emulation.

Para concluir con el apartado, es necesario describir la interfaz de control físico. Véase el siguiente apartado.

3.3.3.2 Interfaz de Control Físico

Para configurar al PLC con un componente físico, se hizo uso de un cajón tipo switch que permite encender y/o apagar el controlador, el sistema posee cuatro salidas conectadas con botones que se iluminarán en caso de un cambio o acción programada, además de contar con cuatro mandos (entradas) que incluyen dos switch y dos de botón, respectivamente, el mismo, también incluye un medidor para saber el voltaje con el que se está trabajando.



Figura 3.14. Control físico del PLC.

Para poner en operación el sistema antes descrito, es indispensable realizar una conexión cableada entre las entradas y salidas del PLC (Figura 3.15 y 3.16), adicionalmente, es necesario el uso del software Interfaz Hombre Máquina donde se indican al PLC las entradas y salidas con las que se va a trabajar. Más detalles al tema, se dan en los modelos experimentales del capítulo siguiente.

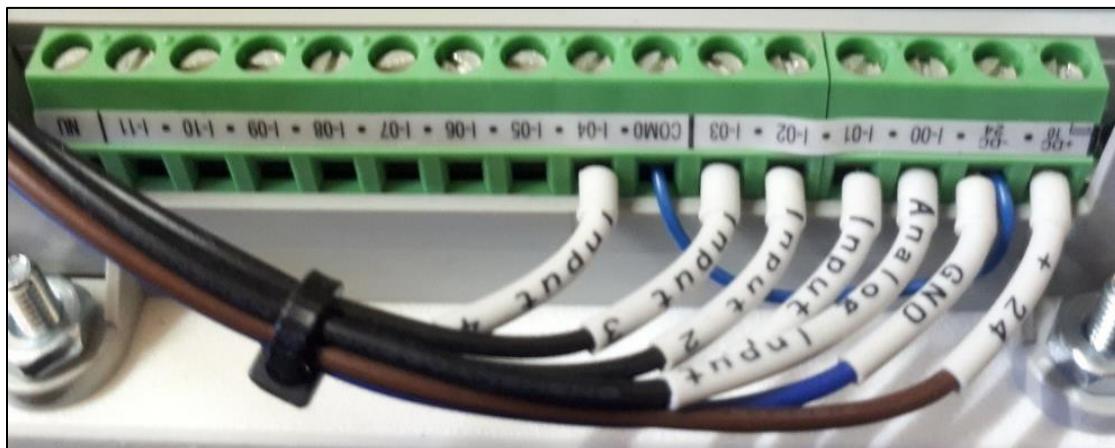


Figura 3.15. Entradas en PLC.

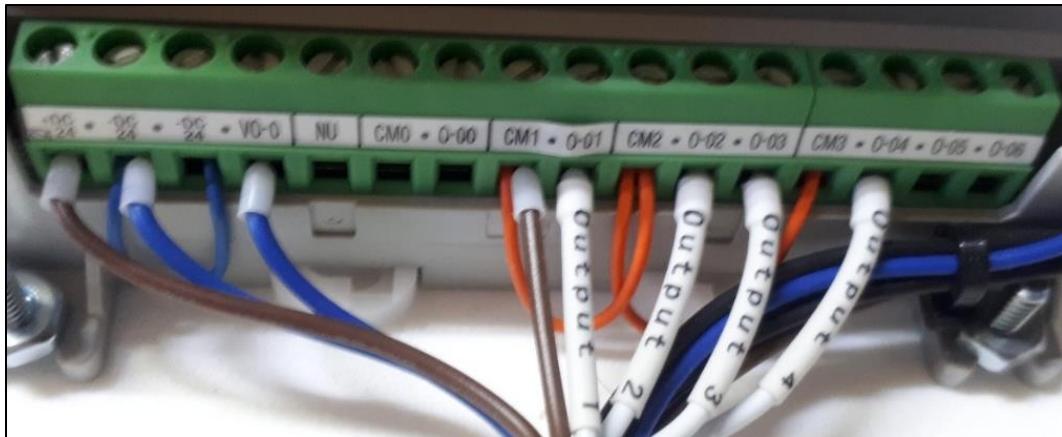


Figura 3.16. Salidas en PLC.

El siguiente capítulo describe el uso de la configuración antes mencionada, con intención de probar bajo diferentes escenarios los usos y ventajas del modelo propuesto.

CAPÍTULO 4. PRUEBAS Y ANÁLISIS

Pruebas

- 4.1 Caso de estudio**
- 4.2 Simulación con FlexSim**
- 4.3 Conexión con Interfaz Hombre Máquina**

Pruebas

La industria 4.0 o la cuarta revolución industrial plantea introducir nuevas tecnologías que afecten a todas las esferas de la sociedad como es el caso de la economía, medicina, transporte, arte, la manufactura, entre otras. Dichas tecnologías (redes 5G, realidad virtual, realidad aumentada, sistemas gemelos, entre otras) redefinen la forma de interactuar y procesar la información abriendo nuevos horizontes que afectan a la industria, permitiendo generar nuevas oportunidades de innovación. A ello, las pruebas mostradas a continuación observan resultados básicos, pero de interés para futuras investigaciones en áreas como: Sistemas de Manufactura Holónicos que cubran diferentes necesidades de la industria o bien, Sistemas Gemelos o Sistemas Ciber-Físicos, los cuales son novedad en la industria 4.0.

En el entendido, para comprobar el flujo de información entre los sistemas analizados y mostrar el uso de la metodología de dos capas, se planteó un caso de estudio que tiene como objetivo demostrar paulatinamente el desarrollo e implementación de cada uno de los elementos de la capa llamada “Control Reactiva”.

4.1 Caso de estudio

Derivado lo anterior, se tiene el objetivo de identificar y definir los elementos de un control físico y un simulador dentro de una celda de manufactura virtual. La finalidad es comunicar adecuadamente a un sistema físico con un modelo virtual a través de un PLC, en intención de alterar el comportamiento de ambos sistemas.

Celda de manufactura

La celda de manufactura de la UAEH es vista como un sistema tipo Job Shop flexible, equipado con tres estaciones de trabajo e integradas con una banda transportadora, y un software de operación y control Open CIM (véase la Figura 4.1):

- Estación 1 (almacén): Se constituye por un almacén automático de 36 posiciones (ASRS) INTELITEK ASRS36u y es operado por un robot (36ASRS1).
- Estación 2 (maquinados): Es un FMS equipado con un torno (CNC) DOOSAN LYNX220, una fresadora (CNC) SUNMILL JHV550 y un robot de 6 grados de libertad FANUC LR Mate 200id.
- Estación 3 (ensamble e inspección): Cuanta con un robot de 6 grados de libertad MOTOMAN MH6-10, una estación de control de calidad con sistema de visión VIEW FLEX y una mesa para ensamble con dispositivos de suministro y sujeción.

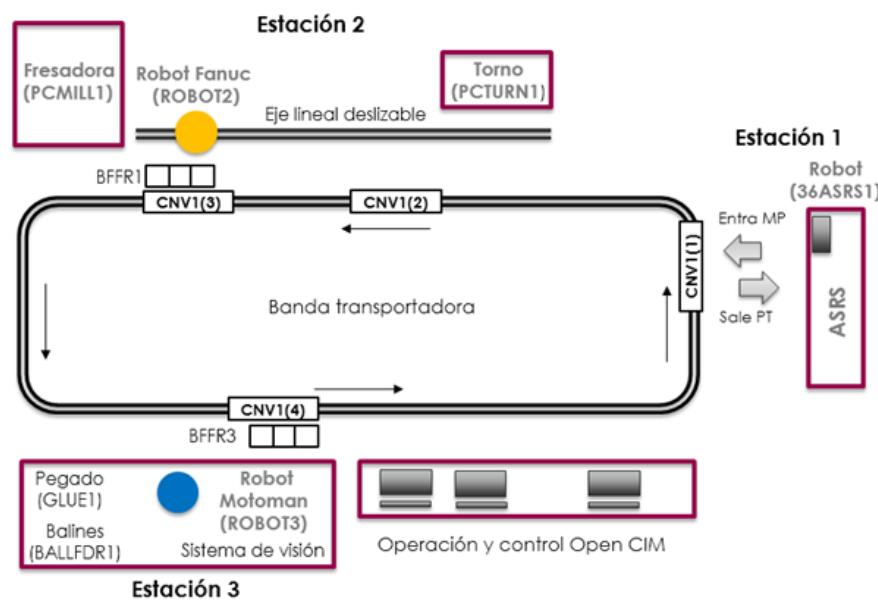


Figura 4.1. Plano esquemático de la celda de manufactura de la UAEH.

En un sistema Job Shop existen n trabajos y cada trabajo visita m número de máquinas siguiendo una ruta predeterminada, dependiendo del modelo de cada trabajo, éste podría visitar una misma máquina en una o más ocasiones, por lo que el sistema Job Shop es sujetos a la recirculación de trabajos [Michael, P. 2009]. El sistema Job Shop flexible que sirve como modelo real en la presente investigación, consta de un conjunto de tres estaciones de trabajo y cada estación de trabajo se compone de una serie de m máquinas o robots. Cada trabajo n tiene una ruta predeterminada de visitas a un número de estaciones de trabajo donde puede ser procesada en alguna de las máquinas que componen dicha estación visitada. De esta manera, cuando es solicitada una orden de producción, esta es programada (software de operación y control Open CIM) de manera que cada una de las estaciones de la celda estén enteradas de las tareas y roles que se deberán desempeñar para efectuar la orden de producción. Así cada máquina es responsable de la realización de las operaciones de manufactura para cada trabajo solicitado.

Datos relevantes a los productos

Cliente y orden del cliente

En un inicio, el administrador de la celda de manufactura interactúa con el sistema de control Open CIM para definir a un cliente (Customer) agregándole alguno de los productos ofertados por el sistema (Product), una cantidad de unidades requeridas (Units Per Product), un valor de prioridad (Priority) y la fecha de entrega (Due Date). Esta asociación de datos a una serie de productos es llamada orden del cliente (Customer Order) la cual incluye la referencia hacia un producto, su cantidad, prioridad y fecha de entrega.

Productos

Los productos son descritos por las especificaciones del modelo (Model Specification) y por las especificaciones del proceso (Process Specification).

- Las especificaciones del modelo, contienen todas las propiedades de cada modelo específico (Model Properties), como; datos técnicos y características estructurales. Los datos técnicos incluyen la forma geométrica (Computer-aided Design, CAD) e imágenes (Pictures) del producto, y las características estructurales de cada producto incluyen materia prima (Raw Material) y subproductos (Sub-assemblies) con sus partes de ensamble o componentes (Components) que lo constituyen. Dichas especificaciones marcan la diferencia entre los distintos tipos de productos, por lo tanto estas especificaciones incluidas en cada modelo delimitan los requerimientos de calidad.
- Las especificaciones del proceso definen el proceso de manufactura o ensamble al cual deben ser sometidas las materias primas o los componentes de cada producto, a esto se le llama plan de procesos.

Ambas especificaciones antes descritas son necesarias para que un producto pueda ser manufacturado (Manufactured) y entregado como producto terminado (Finished Goods).

En este documento, se describe a manera de ejemplo una orden de un cliente constituida por un paquete de tres tipos de productos los cuales son llamados: MILL_TEST, TURN_TEST y GAME-PROD. Los dos primeros productos requieren de MP, mientras el último de ellos necesita de un sub-ensamble por lo que hace necesaria la demanda de componentes. Véase en Figura 4.2 la lista de materiales (Bill of Material, BOM) de cada uno de los tres productos.

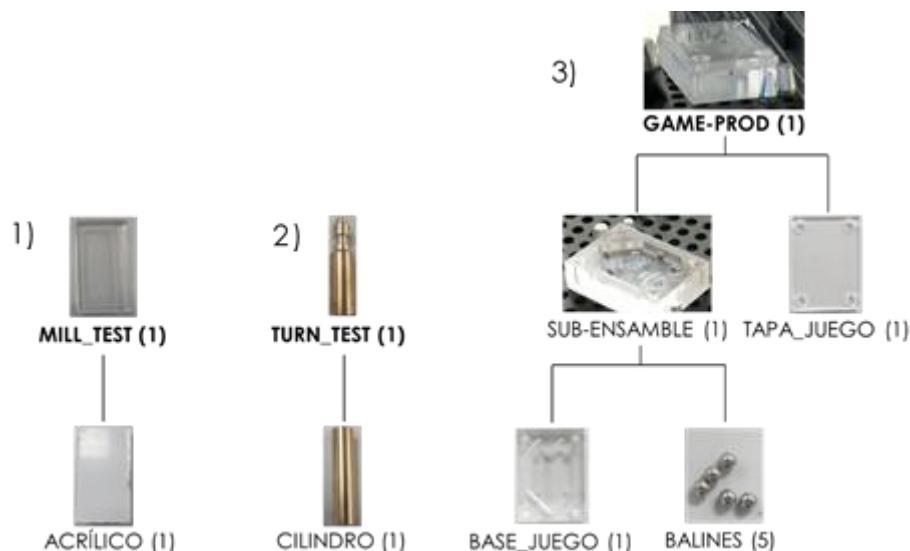


Figura 4.2. Diagrama BOM de cada producto.

La Figura 4.2 muestra que es necesario ACRÍLICO como MP para el producto MILL_TEST y MP CILINDRO para el producto TURN_TEST. El ensamble del producto GAME-PROD requiere de tres componentes: BASE_JUEGO, BALINES y TAPA_JUEGO, además de un sub-ensamble que es también parte componente del mismo.

Suministros

El abastecimiento de MP como de los componentes BASE_JUEGO y TAPA_JUEGO es responsabilidad del almacén automático (ASRS1) de la celda de manufactura, el cual cuenta con suficientes elementos disponibles para la orden. Para la tarea de sub-ensamble se requiere del suministro de “BALINES”, y en el ensamble se hace necesario la agregación de “PEGAMENTO”. El abastecimiento tanto en el sistema alimentador de balines como en la unidad automática de pegado es asumido previamente y de manera suficiente por el administrador del sistema. La Tabla 4.1 muestra las diferentes partes y los datos característicos de las mismas para su identificación.

Tabla 4.1 Datos característicos de MP, componentes y productos.

Nombre de la parte	ID de la parte	Número de Catalogo	Suministro	No. de Template	Parámetros de manufactura	Descripción
ACRÍLICO	11	ACR001	ASRS1	020001	\$TEMPLATEID	Block de acrílico
CILINDRO	1	CIL001	ASRS1	010001	\$TEMPLATEID	Cilindro de bronce
BASE_JUEGO	21	BAS001	ASRS1	030001	ASRS1	Base de acrílico para el juego
TAPA_JUEGO	31	TAP001	ASRS1	040001	ASRS1	Tapa de acrílico para el juego
BALINES		BAL001	BALLFDR1	-	-	Balines para el sub-ensamble del juego
PEGAMENTO	44	PEG001	GLUE1	-	-	Pegamento para ensamble final
MILL_TEST	19	-	-	020001	ASRS1	Producto final de fresado
TURN_TEST	9	-	-	010001	ASRS1	Producto final de torneado
GAME-PROD	22	-	-	030001	RACK1	Producto final de ensamble

Operaciones

Como se dijo anteriormente, las especificaciones del proceso describen el plan de procesos (Process Plan), que es una lista de operaciones necesarias para producir un producto (por ejemplo: maquinados, ensambles o sub-ensambles, manipulación,

transporte, mantenimiento o inspección) de entre una serie de tiempos de operación posibles. Así bien, una operación (Operation) es un trabajo que debe ser ejecutado en un orden específico con el fin de producir el producto y es caracterizada por un conjunto de información como: tiempo estimado de procesamiento (Estimated Processing Time), descripción (Description), precedencia (Precedence) y requisitos técnicos (Technical Requirements).

Para ilustrar como es que cada producto sigue una ruta de producción previamente definida, la Tabla 4.2 describe la secuencia de operaciones necesarias para cada tipo de trabajo. Así mismo las Figuras 4.3, 4.4 y 4.5 muestran las gráficas para los productos MILL_TEST, TURN_TEST y el diagrama de proceso-ensamble del producto GAME-PROD, respectivamente.

Tabla 4.2 Secuencia de operaciones para cada tipo de trabajo.

No.	Fresado para MILL_TEST	Torneado para TURN_TEST	Ensamble para GAME-PROD
1	ASRS1 GET ACRILICO ASRS1	ASRS1 GET CILINDRO ASRS1	ASRS1 GET BASE_JUEGO ASRS1
2	36ASRS1 PLACE TEMPLATE#020001 on RFIDR1	36ASRS1 PLACE TEMPLATE#010001 on RFIDR1	ASRS1 GET TAPA_JUEGO ASRS1
3	RFIDR1 READID ACRILICO \$TEMPLATEID	RFIDR1 READID CILINDRO \$TEMPLATEID	36ASRS1 PLACE TEMPLATE#030001 on CNV1[1]
4	36ASRS1 PLACE TEMPLATE#020001 on CNV1[1]	36ASRS1 PLACE TEMPLATE#010001 on CNV1[1]	CNV1 DELIVER TEMPLATE to CNV1[4]
5	CNV1 DELIVER TEMPLATE to CNV1[3]	CNV1 DELIVER TEMPLATE to CNV1[3]	36ASRS1 PLACE TEMPLATE#040001 on CNV1[1]
6	ROBOT2 PLACE TEMPLATE on BFFR1	ROBOT2 PLACE TEMPLATE on BFFR1	CNV1 DELIVER TEMPLATE to CNV1[4]
7	ROBOT2 PLACE ACRILICO on PCMILL1	ROBOT2 PLACE CILINDRO on PCTURN1	ROBOT3 PLACE TEMPLATE on BFFR3
8	PCMILL1 MILL01 ACRILICO PCMILL1 599	PCTURN1 TURN01 CILINDRO PCTURN1 SET FLIPNEEDED=1 598	ROBOT3 PLACE BASE_JUEGO on BALLFDR1
9	PCMILL1 MILL02 ACRILICO PCMILL1 16	PCTURN1 TURN02 CILINDRO PCTURN1 SET FLIPNEEDED=0 16	BALLFDR1 FEED5 BASE_JUEGO BALLFDR1 FEED5
10	ROBOT2 PLACE MILL_TEST on TEMPLATE#020001[1]	PCTURN1 TORNEADO1 CILINDRO PCTURN1 SET FLIPNEEDED=0 402	ROBOT3 PLACE BASE_JUEGO on RACK1
11	ROBOT2 PLACE TEMPLATE#020001 on CNV1[3]	ROBOT2 PLACE TURN_TEST on TEMPLATE#010001[1]	CNV1 DELIVER TEMPLATE to CNV1[4]
12	CNV1 DELIVER TEMPLATE to CNV1[1]	ROBOT2 PLACE TEMPLATE#010001 on CNV1[3]	ROBOT3 PLACE TEMPLATE on BFFR3
13	36ASRS1 PLACE TEPLATE#020001 on ASRS1	CNV1 DELIVER TEMPLATE to CNV1[1]	ROBOT3 PLACE TAPA_JUEGO on RACK2
		36ASRS1 PLACE TEPLATE#010001 on ASRS1	ROBOT3 BASE BASE_JUEGO on JIG1
			BFFR3 GET FIX TEMPLATE BFFR3[2]
			ROBOT3 ASSEMBLE TAPA_JUEGO/1.1 on BASE_JUEGO[1]
			ROBOT3 PLACE GAME-PROD on TEMPLATE#030001[1]
			ROBOT3 PLACE TEMPLATE#040001 on CNV1[4]
			CNV1 DELIVER TEMPLATE to CNV1[1]
			ROBOT3 PLACE TEMPLATE#030001 on CNV1[4]
			36ASRS1 PLACE TEPLATE#040001 on ASRS1
			CNV1 DELIVER TEMPLATE to CNV1[1]
			36ASRS1 PLACE TEPLATE#030001 on ASRS1

Gráfica de procesos				Resumen		
Método actual	<input checked="" type="checkbox"/>	Método propuesto	<input type="checkbox"/>		Operación	2
Tarea: <u>Fresado</u>					Transporte	9
Producto: <u>MILL_TEST</u>					Inspección	-
					Demora	2
					Almacén	3
No.	Distancia recorrida (metros)	Tiempo de la operación (segundos)	Símbolos	Descripción		
1		5	○ ➡ □ D ▲	Almacén ASRS de MP		
2	1.40	16	○ ➡ □ D △	Robot (36ASRS1): TEMPLATE#020001, del ASRS a referencia RFIDR1		
3	0.10	3	○ ➡ □ D △	Robot (36ASRS1): TEMPLATE#020001, de RFIDR1 a estación CNV1(1), pallet del Conveyor		
4	4	15	○ ➡ □ D △	Conveyor: TEMPLATE#020001, de CNV1(1) a estación CNV1(3)		
5		21	○ ➡ □ D △	Espera en CNV1(3) por posicionamiento de ROBOT2		
6	0.15	14	○ ➡ □ D △	ROBOT2: TEMPLATE#020001, de CNV1(3) a almacén de producto en proceso (BFFR1)		
7		6	○ ➡ □ D ▲	Almacén BFFR1		
8	2	30	○ ➡ □ D △	ROBOT2: ACRILICO, de BFFR1 a PCMILL1		
9		70	● ➡ □ D △	PCMILL1: ACRILICO, maquinado MILL01 599		
11	2	34	○ ➡ □ D △	ROBOT2: MILL_TEST, de PCMILL1 a TEMPLATE#020001 en BFFR1		
13	0.15	14	○ ➡ □ D △	ROBOT2: TEMPLATE#020001, de BFFR1 a estación CNV1(3), pallet del conveyor		
14	7	29	○ ➡ □ D △	Conveyor: TEMPLATE#020001, de CNV1(3) a CNV1(1)		
15		5	○ ➡ □ D △	Espera en CNV1(1) por Robot (36ASRS1)		
16	1.5	15	○ ➡ □ D △	Robot (36ASRS1): TEMPLATE#020001, de CNV1(1) a posición (ASRS-36) del ASRS		
17			○ ➡ □ D ▲	Almacén ASRS de PT		

Figura 4.3. Gráfica de procesos para la tarea de “Fresado” del producto MILL_TEST.

Gráfica de procesos				Resumen		
Método actual	<input checked="" type="checkbox"/>	Método propuesto	<input type="checkbox"/>		Operación	3
Tarea: <u>Torneado</u>					Transporte	9
Producto: <u>TURN TEST</u>					Inspección	-
					Demora	2
					Almacén	3
No.	Distancia recorrida (metros)	Tiempo de la operación (segundos)	Símbolos	Descripción		
1		5	○ → □ D ▲	Almacén ASRS de MP		
2	1.40	16	○ ← □ D △	Robot (36ASRS1): TEMPLATE#010001, del ASRS a referencia RFIDR1		
	0.10	3	○ ← □ D △	Robot (36ASRS1): TEMPLATE#010001, de RFIDR1 a estación CNV1(1), pallet del Conveyor		
3	4	15	○ ← □ D △	Conveyor: TEMPLATE#010001, de CNV1(1) a estación CNV1(3)		
4		21	○ → □ D △	Espera en CNV1(3) por posicionamiento de ROBOT2		
5	0.15	14	○ ← □ D △	ROBOT2: TEMPLATE#010001, de CNV1(3) a almacén de producto en proceso (BFFR1)		
		6	○ → □ D ▲	Almacén BFFR1		
6	3.5	48	○ ← □ D △	ROBOT2: CILINDRO, de BFFR1 a PCTURN1		
7		53	● → □ D △	PCTURN1: CILINDRO, maquinado TURN01 598		
9	2	48	○ ← □ D △	ROBOT2: TURN_TEST, de PCTURN1 a TEMPLATE#010001 en BFFR1		
10	0.15	14	○ ← □ D △	ROBOT2: TEMPLATE#010001, de BFFR1 a estación CNV1(3), pallet del conveyor		
11	7	29	○ ← □ D △	Conveyor: TEMPLATE#010001, de CNV1(3) a CNV1(1)		
12		5	○ → □ D △	Espera en CNV1(1) por Robot (36ASRS1)		
13	1.5	15	○ ← □ D △	Robot (36ASRS1): TEMPLATE#010001, de CNV1(1) a posición (ASRS-36) del ASRS		
14			○ → □ D ▲	Almacén ASRS de PT		

Figura 4.4. Gráfica de procesos para la tarea de “Torneado” del producto TURN_TEST.

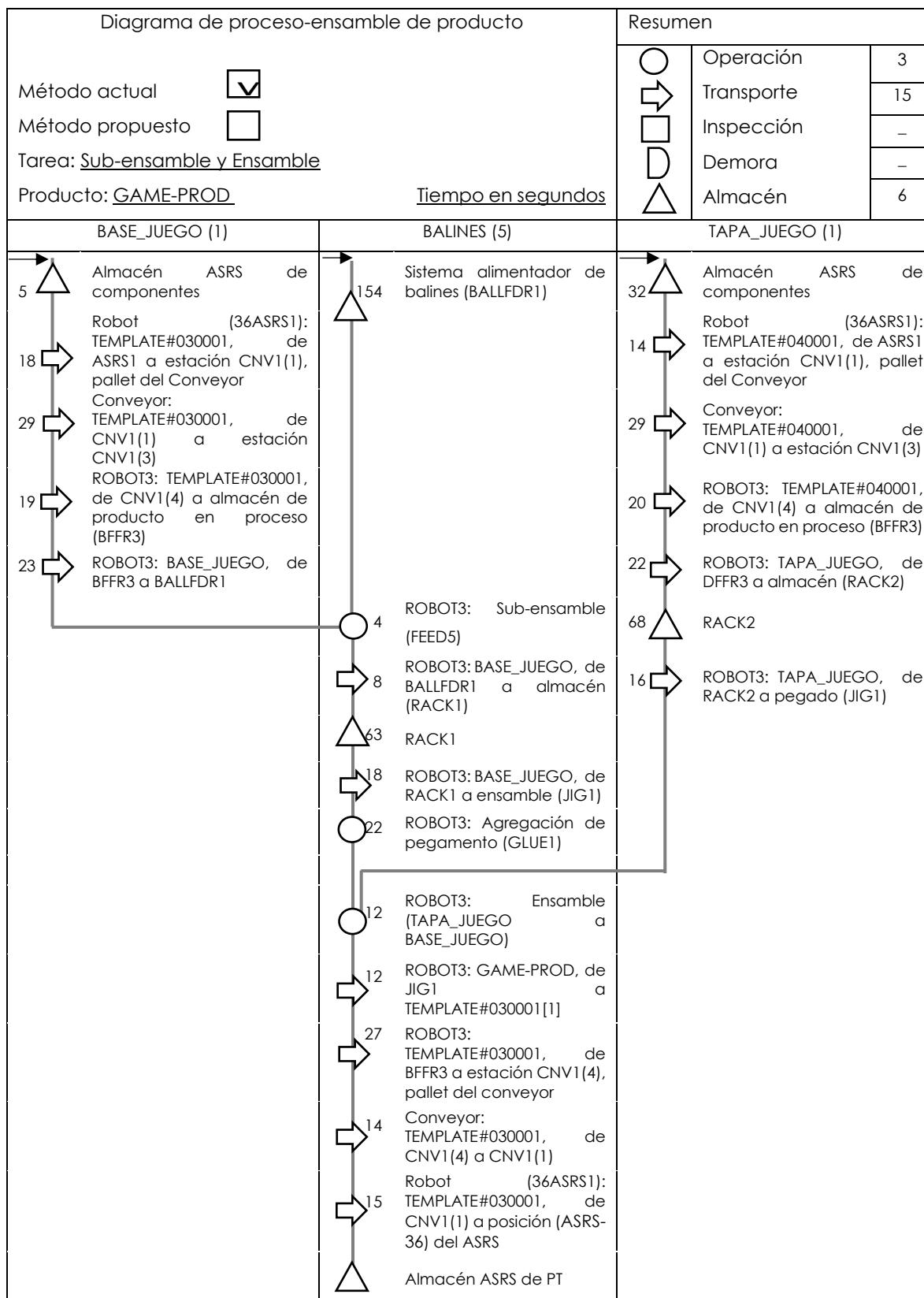


Figura 4.5. Diagrama de proceso-ensamble del producto GAME-PROD.

Como puedo observarse, para realizar un trabajo es necesaria una secuencia de operaciones de manufactura con restricciones temporales en cuanto a la prioridad de ejecución. Así también se observa que una operación es una acción ejecutada por maquinas o recursos responsables de cada acción como el fresado, torneado o la manipulación para el transporte de componentes por los robots, entre otros. Por ejemplo, en la segunda columna de la Tabla 4.2 la operación “PLACE TEMPLATE#020001 on RFIDR” indica al robot 36ASRS1 transportar el TEMPLATE#020001 de una referencia específica del almacén hasta una posición dada por el recurso RFIDR1. Seguido a esto la operación “PLACE TEMPLATE#020001 on CNV1[1]” implica colocar el template que contiene un acrílico dentro de un pallet ubicado en la posición CNV1[1] perteneciente al Conveyor. Después del transporte del template hasta un nodo específico del Conveyor (CNV1[3]), el robot (FANUC) toma el template “ROBOT2 PLACE TEMPLATE on BFFR1” y lo coloca en un almacén de producto en proceso (BFFR1). Del almacén de producto en proceso el robot (FANUC) toma un template disponible “PLACE ACRILICO on PCMILL1” y carga con el acrílico a la fresadora (PCNILL1) para ejecutar un maquinado. La fresadora ejecuta el maquinado (599) “MILL01 ACRILICO PCMILL1 599” y entrega un producto terminado. Esta serie de operaciones son parte del plan de procesos para el fresado del producto MILL_TEST.

De igual modo, las gráficas y diagrama antes mostrados indican los diferentes tiempos (segundos) y distancias (metros) en las distintas operaciones tanto de transporte, manipulación, maquinado, demora o almacenamiento. Los datos son necesario a la hora de estimar de los tiempos de producción.

El administrador del sistema control Open CIM, crea un plan producción (Production Plan) al tomar una o varias órdenes de clientes y convertirlas en órdenes de manufactura (Manufacturing Order). Una orden de manufactura es indexada a un objeto producto y comprende una lista de órdenes de trabajo. Una orden de trabajo (Work Order) es la descripción de una operación (o trabajo) y por lo tanto es parte de un plan de proceso. Las órdenes de trabajo están destinadas a ser ejecutadas por recursos (Resource) tales como fresadoras, tornos, robots, transportadores, entre otros. Cada recurso es una

entidad que puede ejecutar una determinada gama de trabajos (Process Capability), siempre y cuando esté disponible (Available), cuente con lo necesarios (por ejemplo: Supplies, Tools o Devices) y no se exceda su capacidad (Capacity).

La Tabla 4.3 da ejemplos de máquinas o dispositivos necesarios en el procesamiento de las operaciones solicitadas por los tres diferentes productos.

Tabla 4.3. Máquinas, dispositivos y procesos.

No.	Nombre de la máquina	Nombre del proceso	Tipo del proceso	Nombre del archivo o programa	Descripción
1	RFIDR1	READID	QC	RFID1	Lector de referencia ID de la parte
2	PCMILL1	MILL01	CNC	599	Maquinado 599 en fresadora
3	PCMILL1	MILL02	CNC	16	Maquinado 16 en fresadora
4	PCTURN1	TURN01	CNC	598 SET FLIPNEEDED=1	Maquinado 598 en torno
5	PCTURN1	TURN02 SIN FLIP	CNC	16 SET FLIPNEEDED=0	Maquinado 16 en torno
6	PCTURN1	TORNEADO1	CNC	402 SET FLIPNEEDED=0	Maquinado 402 en torno
7	BALLFDR1	FEED5	PROCESS	FEED5	Sub-ensamble o agregación de balines
8	JIG1	ASSEMBLE	ASSEMPLER	ASSEMBLE	Área de ensamble de BASE_JUEGO con TAPA_JUEGO
9	GLUE1	GLUE	PROCESS	FIX	Agregación de pegamento

4.2 Simulación con FlexSim

Dado el caso anterior se realizó la simulación de la celda de manufactura (figura 4.1) con ayuda del software FlexSim, se tomaron los tiempos mencionados anteriormente (figuras 4.3, 4.4 y 4.5) en unidades constantes para cada uno de los procesos, así como las distancias establecidas en la celda de manufactura real.

Se simularon las tres estaciones de la celda de manufactura estación 1 (almacén), estación 2 (maquinados), estación 3 (ensamble e inspección).

Este modelo fue construido con los objetos: Source, Robot, ASRSvehicle, Queue, Rack, Processor, Conveyor, Separator, Combiner y sink (Véase en Figura 4.6).

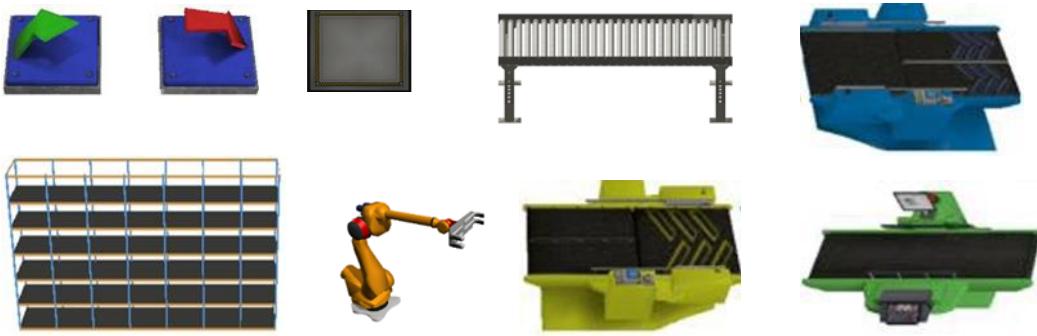


Figura 4.6. Objetos de FlexSim.

FlexSim permite la personalización de cada uno de los objetos que integran un modelo de simulación por lo que se crearon algunos objetos (productos, torno, fresadora, computadoras, mesas, etc.) para llegar a un nivel de detalle más alto.

Descripción

Estación 1 (almacén): Esta estación cuenta con dos racks, materia prima y producto terminado, los racks son abastecidos por 4 Combiner, cada uno es un producto diferente, en los Combiner se une la materia prima a una pequeña charola con la forma necesaria para entrar al TEMPLATE que permanece dentro de la banda transportadora.

La estación contiene un ASRSvehicle el cual busca la posición de la materia prima deseada dentro de los racks, este objeto nos ayuda a transportar el producto de los racks a dos pequeños almacenes para después ser llevados a la banda transportadora.

Por último, en la parte de la banda transportadora se cuenta con dos Combiner más, los cuales combinan la charola pequeña con el TEMPLATE para después ser llevados a las siguientes estaciones de trabajo, además, en esta estación se cuenta con un Separator el cual separa las charolas mandando la charola pequeña con el producto terminado al Rack de producto terminado y mandando la charola grande por la banda transportadora para poder ser utilizada por otro producto a fabricar (véase la figura 4.7).

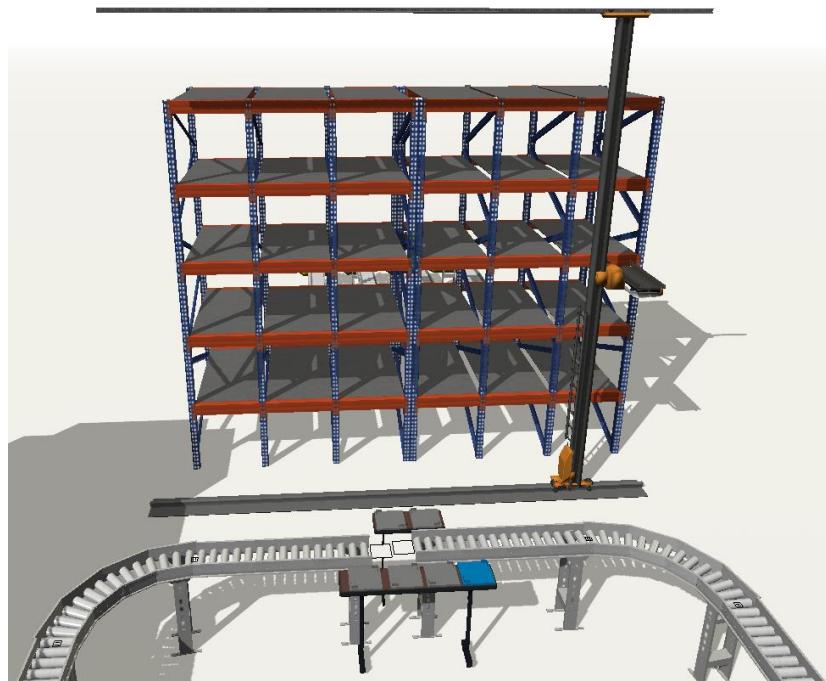


Figura 4.7. Estación 1 (almacén)

Estación 2 (maquinados): La estación contiene dos Processor los cuales representan la operación de un torno y una fresadora respectivamente, estos Processor están programados para trabajar al mismo tiempo que las maquinas reales en la celda de manufactura. La estación cuenta con un Robot que nos ayuda a transportar el producto de la banda trasportadora a una pequeña zona de almacenamiento de materia prima y producto terminado (MILL_TEST, TURN_TEST), esta zona de almacenamiento cuenta con un Separator el cual nos ayuda a separar la materia prima de las charolas pequeñas, cuenta con Queues para almacenar materia prima o charolas pequeñas y con Combiners que nos ayudan a combinar el producto terminado con una charola pequeña para después ser llevado a la banda transportadora y finalmente al Rack de producto terminado. La figura 4.8 ilustra al ROBOT2 quien realiza la función de llevar la materia prima al torno y la fresadora para ser procesada según sea necesario.

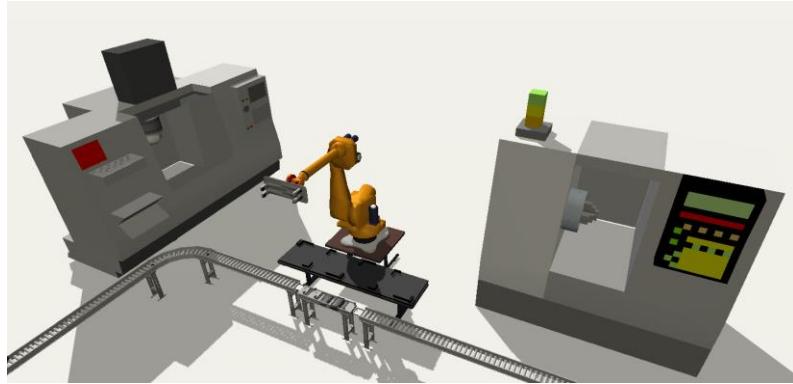


Figura 4.8. Estación 2 (maquinados)

Estación 3 (ensamble): Esta estación está conformada por una zona de ensamblaje y una zona de almacenamiento similar a la mencionada anteriormente en la estación 2 con la diferencia que en esta estación el ROBOT3 además de ayudarnos a transportar los objetos nos ayuda a hacer el ensamblaje (GAME-PROD).

Para simular los ensambles (producto GAME-PROD) se ocuparon cuatro Combiners programados para que el ROBOT3 realice el proceso, el orden de los Combiner es el siguiente: unión de BASE_JUEGO con BALINES (SUB-ENSAMBLE), unión de SUB-ENSAMBLE con TAPA_JUEGO. Finalmente, el producto terminado (GAME-PROD) se coloca sobre una charola pequeña y se une con el TEMPLATE (banda transportadora). La figura 4.9 muestra el ensamblaje del producto aquí descrito.

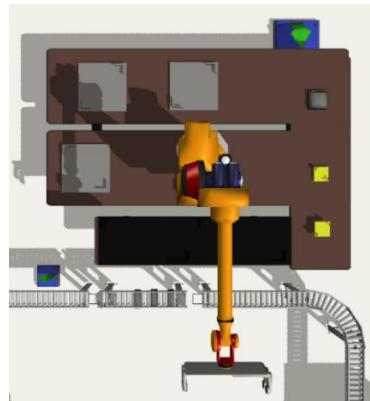


Figura 4.9. Estación 3 (ensamble).

4.3 Conexión con Interfaz Hombre Máquina

El presente apartado describe de manera puntual la integración del sistema de manufactura virtual simulado en el software FlexSim con el sistema físico, ambos comunicados a través del PLC de marca Allen-Bradley.

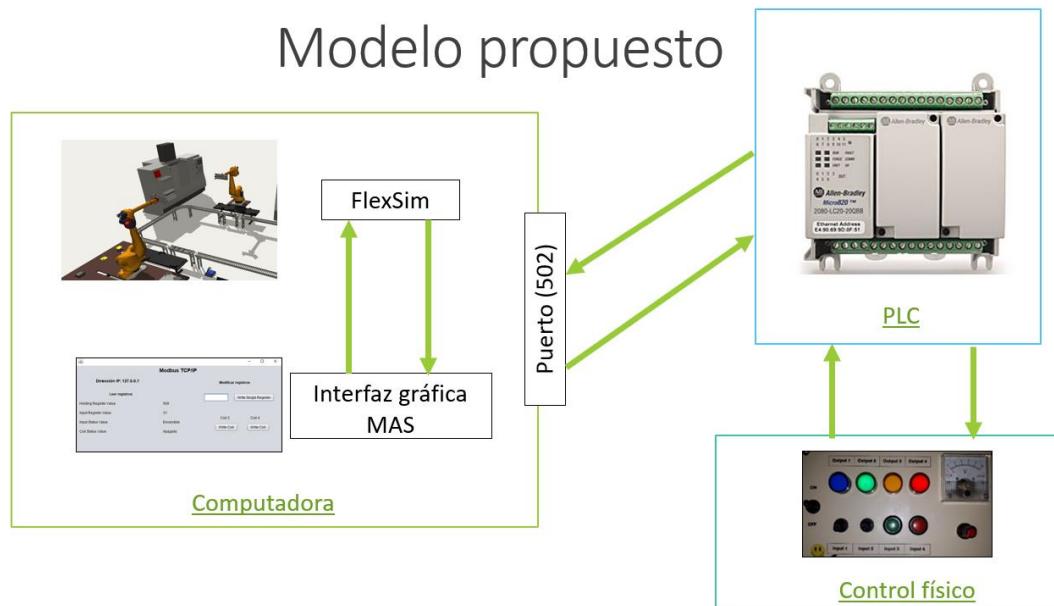


Figura 4.10 Diagrama de escalera (LD) utilizado.

La configuración del PLC Micro820 Allen-Bradley se hizo con el Software Interfaz Hombre Máquina, este software nos permite programar al controlador y desarrollar su integración con el editor.

Esta simulación está configurada de modo que la botonera reciba mensajes y de aviso por medio de luces, recibe señales cuando el simulador hace uso de sus recursos fijos y móviles como el torno (TURN_TEST), fresadora (MILL_TEST) y robots (ROBOT2 y ROBOT3). Cuando el torno y la fresadora están procesando la materia prima se envía una señal a la botonera para que encienda una luz respectivamente, de igual modo sucede con los robots pero en este caso envían una señal cuando toma algún objeto ya sea para llevar materia prima al torno o fresadora y cuando se está ensamblando algún producto. En el caso contrario, cuando la botonera envía señales al simulador este

reacciona de diferente manera, está configurado para que la botonera pueda controlar la velocidad de las bandas transportadoras que surten al almacén, se cuenta con dos botones que controlan dos bandas cada uno, un botón controla la velocidad de las bandas transportadora de ACRILICO y CILINDRO, el otro botón controla las bandas transportadoras de BASE_JUEGO Y TAPA_JUEGO.

Dentro de la configuración se utilizó el lenguaje en escalera y se declararon 6 variables globales de tipo booleano, dos variables de entrada para controlar el modelo virtual y 4 variables de salida para mandar señales al control físico:

Input 1: Controlar velocidad de banda transportadora para materia prima (TAPA_JUEGO, BASE_JUEGO)

Input 2: Controlar velocidad de banda transportadora para materia prima (ACRILICO, CILINDRO)

- Output 1: Enviar señal de PCTURN1 a control físico.
- Output 2: Enviar señal de PCMILL1 a control físico.
- Output 3: Enviar señal de ROBOT3 a control físico.
- Output 4: Enviar señal de ROBOT2 a control físico.

En la figura 4.11 muestre el diagrama en escalera del software Interfaz Hombre Máquina donde se liga a las variables antes descritas.



Figura 4.11 Diagrama de escalera (LD) utilizado.

Se configuro la dirección IPv4 (Address) para establecer la conexión con el cliente o servidor, de igual forma se definió el Intervalo de Frecuencia o Cambio (Change Interval) mismo que implica la frecuencia (tiempo en segundos) con la que se actualizan o refrescan datos, y el Registro Múltiple (Multi-Register) deberá permitir la recepción de más de un byte e interpretarlo correctamente (véase figura 4.12).

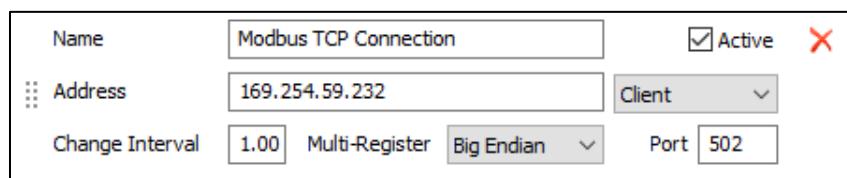
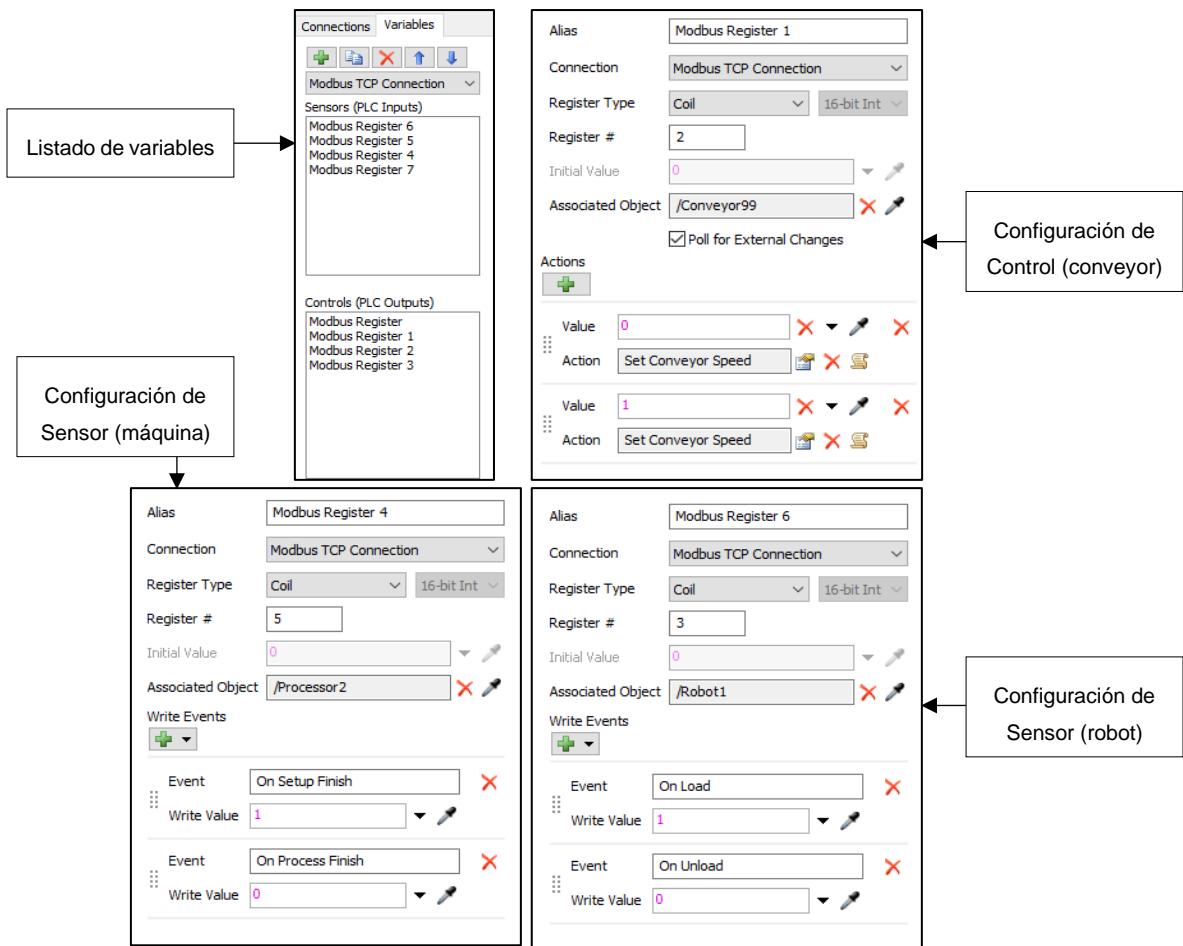


Figura 4.12 Connections

Hecho lo anterior se configuran las variables correspondientes al diagrama de escalera (véase figura 4.13) comunicando las 4 bandas transportadoras en Controls (PLC Outputs) y las maquinas (fresa, torno) y robots en Sensors (PLC Inputs).

**Figura 4.13 Variables**

Una vez descrito el caso de estudio el siguiente apartado describe los resultados alcanzados.

CAPÍTULO 5. RESULTADOS FINALES

- 5.1 Resultados Generales**
- 5.2 Conclusiones**
- 5.3 Recomendaciones**
- 5.4 Trabajos futuros**

5.1 Resultados Generales

- En el capítulo 2 se describe la fundamentación teórica de la investigación mediante el análisis de bibliografía referida a sistemas de producción, manufactura inteligentes y simulación de eventos discretos;
- Con la intención de comprender como opera un HMS se hizo una revisión completa del modelo holónico propuesto por la Escuela Superior de Cd. Sahagún de Universidad Autónoma del Estado de Hidalgo;
- Haciendo uso del software FlexSim se diseñó un modelo que emula el comportamiento de la celda de manufactura de la Universidad Autónoma del Estado de Hidalgo, con la finalidad de visualizar en tercera dimensión el comportamiento del conjunto de elementos del mismo;
- Se hizo una revisión del uso del software Interfaz Hombre Maquina, en intención de comprender la programación del PLC Micro820 de Allen-Bradley.
- Una vez comprendido el sistema físico y el modelo de simulación se logró a través del protocolo TCP/IP la comunicación efectiva entre ambos sistemas, y
- En intención de validar el modelo de comunicación se hicieron diferentes corridas de simulación, los resultados se pueden observar en el capítulo 4.

Fueron alcanzados los objetivos planteados al inicio del presente documento, se comprobó el uso de la metodología referida en el Capítulo 3 y se establecieron las bases hacia investigaciones futuras respectivas a MAS y HMS capaces de ejercer control en tiempo real sobre los sistemas de manufactura inteligentes, así como su aplicación en Sistemas Ciber-Físicos.

5.2 Conclusiones

Como conclusión, los resultados de la investigación amplían el entendimiento y uso de un protocolo de comunicación específico entre un sistema físico (PLC, botonera) y un sistema virtual. El protocolo de comunicación desarrollado en las diferentes pruebas garantiza la interacción de los recursos por medio de una red industrial típica, basada en TCP/IP. Así mismo, es prudente comentar que el modelo desarrollado en este documento tuvo como base arquitecturas y normas internacionales mismas que fueron propuestas con la finalidad de permitir que diferentes empresas las adopten.

De la misma manera este trabajo permitió determinar la estrategia de control para sistemas de manufactura simples, por lo que será posible una eventual mejora en la implementación de sistemas de control reactivos, la propuesta puede adaptarse y/o mejorarse para su aplicación a nuevos casos de estudio.

Adicional a lo antes comentado, adoptar tecnologías innovadoras a la industria se crean grandes ventajas tales como; control automático de la producción, planeación de la producción, coordinación de operaciones, escalamiento, tolerancia a errores, seguridad, y proyecciones futuras ante condiciones inciertas.

Como una opinión particular el desarrollo del presente proyecto aporta conocimiento valido al área de investigación en temas relacionados con ingeniería industrial, toda vez que se abordan tópicos de manufactura, sistemas de producción, control de la producción, automatización, planeación de operaciones, toma de decisiones, administración de recursos y simulación de sistemas de producción. Lo anterior, fortalece el perfil de egreso de futuros profesionistas quienes en su momento deseen iniciarse en el control de sistemas de manufactura.

5.3 Recomendaciones

Es recomendable seleccionar una arquitectura para la organización de agentes que encarguen del monitoreo del sistema emulado, buscando dar una jerarquía ya que no todos deberían de tener las mismas conexiones con los elementos lo que provocaría el caos.

5.4 Trabajos Futuros

Con la llegada de nuevos conceptos en la industria capaces de exprimir al máximo los beneficios de la transformación digital que conlleva abordar la Industria 4.0, llegando los Gemelos digitales o los Sistemas de Manufactura Holónicos, el proyecto presentado en este documento se puede ir adaptando a las necesidades de la Industria 4.0.

A ello, los modelos futuros deberán adoptar las siguientes características:

- Reconfigurabilidad.
- Robustez del sistema.
- Velocidad de respuesta.
- Adaptabilidad y portabilidad de los aplicativos.

Referencias

- Araúzoa, J., Martínez, R., Laviós, J., & Martína, J. (2015). Programación y control de sistemas de fabricación flexibles: un enfoque holónico. *Revista Iberoamericana de Automática e Informática Industrial*, 58-68.
- Automation, R. (18 de Noviembre de 2018). *Rockwell Automation*. Obtenido de Allen-Bradley: <https://ab.rockwellautomation.com/es/Programmable-Controllers/Micro820>
- Banks, J., Carson, J. S., Nelson, B. L., & Nicol, D. (2005). *Discrete-Event System Simulation*. USA: Prentice Hall.
- Barrera, J. H. (2012). *El proyecto de investigación*. Caracas: Quiron.
- Beaverstock, M., Greenwood, A., Lavery, E., & Nordgren, W. (2012). *Applied Simulation Modeling and Analysis using FlexSim*. Orem : FlexSim Software Products, Inc.
- Botti, V., & Giret , A. (2008). *ANEMONA, A Multi-agent Methodology for Holonic Manufacturing Systems*. Valencia: Springer.
- Bunge, M. (1999). *Diccionario de filosofía*. México: SIGLO XXI (MEXICO).
- Christensen, J. (2003). HMS/FB architecture and its implementation. En S. Deen, *Agent-Based Manufacturing: Advances in the Holonic Approach* (págs. 53-88). Berlin: Springer.
- D'Inverno, M., & Luck, M. (2001). *Understanding Agent Systems*. Berlin: Springer-Verlag.
- Deen, S. (2003). *Agent-Based Manufacturing – Advances in the Holonic Approach*. Heidelberg, Germany: Springer.
- Design, W. S. (11 de Mayo de 2017). *WinTECH Software Design*. Obtenido de WinTECH Software Design: <https://www.win-tech.com/>

REFERENCIAS

- FIPA. (1996-2018). *FIPA*. Obtenido de The Foundation for Intelligent Physical Agents:
<http://www.fipa.org/>
- FlexSim Software Products, I. (5 de Enero de 2019). *FlexSim*. Obtenido de FlexSim Problem Solved: <https://www.flexsim.com/es/>
- García, A., Simón-Marmolejo, I., & Cruz, R. (2018). Modelo de comunicación entre un sistema holónico y un sistema de eventos discretos. *Multi-disciplinary International Congress*. Apan.
- Gruber, T. R. (1993). *Knowledge Acquisition*. ELSEVIER.
- Hochhalter J, Leser WP, Newman JA, Gupta VK, Yamakov V, Cornell SR, Willard SA, Heber G. (2014). *Coupling Damage-Sensing Particles to the Digital Twin Concept*. NASA Center for AeroSpace Information.
- Hurtado, J. M. (13 de Febrero de 2019). *Info-PLC*. Obtenido de <http://www.infoplcc.net/documentacion/7-comunicaciones-industriales/2332-redes-comunicacion-industrial>
- IEC, I. E. (2012a). IEC 61131: Programmable Controllers-Part 3 Programming languages, Standard, third ed.
- IEC, I. E. (2012b). IEC 61499: Funcion Blocks-Part 1-4, Standard, second ed. .
- Jaime, G. (1975). *Una teoría cibernetica de la ciudad y su sistema*. Argentina: Ediciones Nueva Visión S.A.I.C.
- Koestler, A. (1967). *The ghost in the machine*. Arkana.
- Krizhevsky, A., Sutskever, I., & Geoffrey E., H. (2012). *Advances in neural information*.
- Laughton, M. A., & Warne, D. J. (2003). *Electrical Engineer's Reference book*. Newnes.

REFERENCIAS

- Lee, E.A.; Seshia, S.A. (2014). *Introduction to Embedded Systems - A Cyber-Physical Systems Approach*, Edition 1.5, LeeSeshia.org.
- Leitão, P. (2009). Agent-based distributed manufacturing control: A state-of-the-art survey. *Engineering Applications of Artificial Intelligence* (págs. 979-991). Elsevier.
- Leitão, P., & Pavel., V. (2011). Recent developments and future trends of industrial Agents. En *Lecture Notes in Computer Science*. (págs. 15-28). Verlag Berlin Heidelberg: Springer.
- Leitão, P., & Restivo, F. (2006). ADACOR: a holonic architecture for agile and adaptive manufacturing control. *Computers in Industry*, 121-130.
- L. Monostori, B. Kádár, T. Bauernhansl, S. Kondoh, S. Kumara, G. Reinhart, O. Sauer, G. Schuh, W. Sihn, K. Ueda. (2016). *Cyber-physical systems in manufacturing, CIRP Annals – Manuf. Technol.*
- Michael, P. (2009) *Planning and Scheduling in Manufacturing and Services*
- Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill Science.
- Modbus Organization, I. (2005-2019). *Modbus*. Obtenido de <http://www.modbus.org/>
- Moreno, W. M., & Cortés, A. C. (2018). *Tecnológico de Costa Rica*. Obtenido de Escuela de Ingeniería Electrónica: <http://www.ie.tec.ac.cr/pmwiki/pmwiki.php/>
- Nirenburg, S. (2004). *Ontological Semantics*. Baltimore: University of Maryland.
- Oztemel, E. (2010). *Artificial Intelligence Techniques for Networked Manufacturing Enterprises Management. Springer Series in Advanced Manufacturing*. United Kingdom: Cardiff.
- Radu F. Babiceanu, F. F. (2006). *Development and applications of holonic manufacturing systems: a survey*. Journal of Intelligent Manufacturing.

REFERENCIAS

- RADU, F. B., & F. FRANK, C. (2006). Development and applications of holonic manufacturing systems: a survey. En *Journal of Intelligent Manufacturing* (págs. 111-131). Springer.
- Rajkumar, M.; Ae, N.; Prasad, M.N.V.; Freitas, H. (2010) *Potential of Siderophore-Producing Bacteria for Improving Heavy Metal Phytoextraction*. Trends Biotechnol.
- Spiegel, R. (2019) The Three Basic Functions of the Digital Twinhttps://www.designnews.com/design-hardware-software/three-basic-functions-digital-twin/151859635660187
- Rosenzweig, E., & Easton, G. (2013). *Tradeoffs in manufacturing? A meta analysis and critique of the literature*.
- Rossmann-engineering. (2017). *EasyModbusTCP*. Obtenido de rossmann-engineering:
<http://easymodbustcp.net/en/>
- Rüßmann, M., et al. (2015) *Industry 4.0: The future of productivity and growth in manufacturing industries*. Boston Consulting Group (BCG).
- Russell, S., & Norvig, P. (2004). *Inteligencia Artificial, Un enfoque moderno*. Salamanca: Pearson, Prentice Hall.
- Saddik, A. El (2018). "Digital Twins: The Convergence of Multimedia Technologies". IEEE MultiMedia.
- Salazar, L. A. (2017). *Estructura de control para procesos de producción desde el paradigma de los Sistemas Holónicos de Manufactura*. Cartagena: Cartagena de Indias D.T. y C.
- Sánchez, R. (2012). *Implementación de protocolo de comunicaciones MODBUS/TCP para linux en lenguaje en c++. aplicación sobre analizadores de red SIEMENS SENTRON PAC4200*. Cartagena: Universidad Politécnica de Cartagena.

REFERENCIAS

- Schalkoff, J. (1990). *Artificial Intelligence*. EU: Springer.
- Serrano, A. G., & Ossowski, S. (1998). Inteligencia Artificial Distribuida y Sistemas Multiagente.
- Shen, W. H., Q., J. Y., & H. and Norrie, D. (2006). *Applications of agent-based systems in intelligent manufacturing: An updated review*. (Vol. Advanced Engineering Informatics Volume 20). Elsevier Ltd.
- Siemens. (5 de Febrero de 2019). *SIEMENS*. Obtenido de Siemens México: <https://new.siemens.com/mx/es.html>
- Simón-Marmolejo, I., López-Ortega, O., & Ramos-Velasco, L. E. (2016). Desarrollo de una Ontología de Acuerdo con el Paradigma de Sistemas Holónicos de Manufactura. *14th LACCEI International Multi-Conference for Engineering, Education, and Technology: "Engineering Innovations for Global Sustainability"*. San José, Costa Rica.
- SpA, T. I. (s.f.). *JADE*. Obtenido de <http://jade.tilab.com/>
- Tanenbaum, A. S., & Wetherall, D. J. (2012). *Redes de computadoras*. México: Pearson Educación.
- Tao F, Cheng Y, Cheng J, et al. (2017). *Theories and technologies for cyber-physical fusion in digital twin shop-floor*. Computer Integrated Manufacturing Systems.
- Thomas, S., & Alois, Z. (2015). Distributed real-time automation and control - Reactive control layer for industrial agents. En L. Paulo, & K. Stamatis, *Industrial Agents - Emerging Applications of Software Agents in Industry* (págs. 89-107). Germany: Elsevier.
- Ulieru, M., Stefanou, D., & Norrie, D. (2000). Holonic self-organization of multi-agent systems by fuzzy modeling with application to intelligent manufacturing.

REFERENCIAS

- Proceedings of the 2000 IEEE International Conference on Systems, Man, and Cybernetics* (págs. 1661-1666). Nashville, TN, USA: IEEE.
- Uschold, M. (1996). Ontologies: principles, methods and applications. *Knowledge Engineering Review*, 93-155.
- Vlahavas, I., & Vrakas, D. (2005). *Intelligent Techniques for Planning*. IGI Global.
- Wang X, Xu X. (2014). *Virtualise manufacturing capabilities in the cloud: requirements, architecture and implementation*. International Journal of Manufacturing Research.
- Wooldridge, M. (2009). *An Introduction to MultiAgent Systems*. United Kingdom: John Wiley & Sons Ltd. Second Edt.
- Wooldridge, M., & Jennings, N. R. (1995). *Agent Theories, Architectures, and Languages: A survey* (Vol. 890). Berlin: Springer-Verlag.
- zaphodikus. (2017). *Modbus PLC Simulator*. Obtenido de Modbus PLC Simulator: <http://www.plcsimulator.org/>