



# Desarrollo e integración de un sistema de toma de decisiones basado en simulación e inteligencia artificial para sistemas de manufactura flexible en la industria 4.0

**Ing. Carolina Saavedra Sueldo**

Tesis presentada para obtener el grado de  
Doctora en Ingeniería  
Mención Tecnología Electromecánica

Directores:

Dr. Mariano De Paula  
Dr. Sebastián Aldo Villar

Facultad de Ingeniería  
Universidad Nacional del Centro de la Provincia de Buenos Aires

Olavarría, Buenos Aires, Argentina  
Julio 2025





# Índice general

Índice de figuras	VIII
Índice de tablas	IX
Índice de algoritmos	XI
Acrónimos	XIII
Dedicatoria	XV
Agradecimientos	XVII
Resumen	XIX
Abstract	XXI
Prefacio	XXIII
<b>I Contexto y Estado del arte</b>	<b>1</b>
<b>1 Introducción</b>	<b>3</b>
1.1 Motivación . . . . .	5
1.2 Objetivos general y específicos . . . . .	6
1.3 Contribuciones de la tesis . . . . .	7
1.4 Estructura de la tesis . . . . .	9
<b>2 Estado del Arte</b>	<b>11</b>
2.1 Cambio de paradigma: Hacia la Industria 4.0 . . . . .	11
2.2 Lean 4.0 y el manejo eficiente de materiales en plantas manufactureras	13
2.3 Simbiosis entre la Simulación e IA para la TDD . . . . .	17
<b>3 Antecedentes metodológicos</b>	<b>21</b>
3.1 Sistemas flexibles de producción . . . . .	21
3.2 Modelado y simulación de sistemas productivos . . . . .	23
3.3 Gemelos digitales . . . . .	26
3.4 Optimización metaheurística . . . . .	29
3.5 Inteligencia artificial para la toma de decisiones . . . . .	31
3.5.1 Aprendizaje por refuerzo . . . . .	32
3.5.2 Aprendizaje por refuerzo profundo . . . . .	35

## II Sistema autónomo de toma de decisiones para la Industria 4.0 39

<b>4</b>	<b>Componentes de los gemelos digitales para entornos robotizados</b>	<b>41</b>
4.1	Estructura de un gemelo digital para sistemas de manufactura . . . . .	41
4.2	Integración de ROS en entornos industriales . . . . .	44
4.3	Entornos de simulación de planta . . . . .	46
4.4	Conclusiones del capítulo . . . . .	49
<b>5</b>	<b>Desarrollo e integración del SATDD</b>	<b>51</b>
5.1	Arquitectura software: Integración ROS - Entorno de simulación . . . . .	51
5.2	Modelo conceptual de la Arquitectura . . . . .	54
5.3	Sistema autónomo de toma de decisiones basado en un gemelo digital	56
5.4	Prueba de concepto de la arquitectura en la Celda de manufactura . .	58
5.5	Aproximación a la simulación semi-automática en el desarrollo de GDs	62
5.5.1	Relevamiento del sistema físico . . . . .	63
5.5.2	Modelado computacional . . . . .	64
5.6	Conclusiones del capítulo . . . . .	67
<b>6</b>	<b>Aplicación del SATDD para el manejo de materiales en piso de planta</b>	<b>69</b>
6.1	Caso de Estudio . . . . .	70
6.1.1	Definición del problema . . . . .	70
6.1.2	Modelo conceptual . . . . .	70
6.1.3	Modelo digital . . . . .	72
6.2	Integración digital y SATDD para el manejo de materiales . . . . .	73
6.3	Resultados . . . . .	75
6.3.1	Q-learning . . . . .	75
6.3.2	SARSA . . . . .	77
6.3.3	DQN . . . . .	78
6.4	Conclusiones del capítulo . . . . .	81
<b>7</b>	<b>Algoritmo de optimización para la gestión eficiente de materiales con minimización de desperdicios</b>	<b>83</b>
7.1	Definición del problema . . . . .	84
7.2	Configuración del entorno computacional . . . . .	87
7.3	Algoritmo seno-coseno discreto reforzado . . . . .	88
7.3.1	Configuración del $R^*DSCA$ . . . . .	91
7.4	Resultados . . . . .	93
7.4.1	Evaluación del $R^*DSCA$ . . . . .	93
7.4.2	Comparación de rendimiento . . . . .	98
7.4.3	Evaluación del $R^*DSCA$ para la reducción de desperdicios . .	100
7.5	Conclusiones del capítulo . . . . .	103

<b>III Conclusiones y futuros desarrollos</b>	<b>105</b>
<b>8 Las PyMEs y la transición digital 4.0</b>	<b>107</b>
8.1 Aspectos fundamentales del relevamiento . . . . .	108
8.2 Resultados del relevamiento . . . . .	111
8.3 Conclusiones del capítulo . . . . .	114
<b>9 Conclusiones de la tesis</b>	<b>117</b>
9.1 Contribuciones y resultados clave . . . . .	117
9.2 Propuestas para trabajo futuro . . . . .	118
<b>A Arquitectura software: Detalle de códigos</b>	<b>121</b>
<b>Bibliografía</b>	<b>130</b>



# Índice de figuras

3.1	Relaciones en un sistema de manufactura flexible. . . . .	23
3.2	Simbología de la técnica IDEF-SIM. . . . .	25
3.3	Jerarquía funcional de entidades en un sistema de fabricación. . . . .	27
3.4	Esquema del aprendizaje por refuerzo. . . . .	33
3.5	Esquema del algoritmo DQN. . . . .	36
4.1	Relación funcional de un GD para un sistema de fabricación. . . . .	42
4.2	Sistema de fabricación robotizado hipotético y simplificado. . . . .	46
4.3	Gemelo digital de una hipotética planta de fabricación sencilla. . . . .	48
4.4	Interfaz de FlexSim. . . . .	49
4.5	Interfaz de Tecnomatix Plant Simulation. . . . .	49
5.1	Arquitectura software para integrar ROS y un simulador de procesos de fabricación. . . . .	52
5.2	Diagrama UML de SimulAI. . . . .	55
5.3	Sistema de TDD basado en un agente de AR utilizando el GD de una hipotética planta de fabricación simple. . . . .	57
5.4	Celda de manufactura. Vista principal de cinta transportadora. . . . .	59
5.5	Celda de manufactura. Vista principal de cinta transportadora y mesa giratoria. . . . .	59
5.6	Arquitectura particular de conexiones en la celda de manufactura. . . . .	60
5.7	Nodos y tópicos utilizados en el experimento. . . . .	61
5.8	Comparación de latencias en ambos sentidos de comunicación. Experimento 1. . . . .	62
5.9	Comparación de latencias en ambos sentidos de comunicación. Experimento 2. . . . .	62
5.10	Esquema conceptual para un relevamiento de un sistema físico. . . . .	64
5.11	Modelo genérico para la simulación semi-automática. . . . .	66
5.12	Modelo generado a partir de la Tabla 5.3. . . . .	66
6.1	IDEF-SIM del primer caso de estudio. . . . .	71
6.2	Modelo de simulación del primer caso de estudio. . . . .	72
6.3	Recompensa acumulada por episodio implementando el algoritmo Q-learning. . . . .	77
6.4	Recompensa acumulada por episodio implementando el algoritmo SARSA. . . . .	78
6.5	Recompensa acumulada por episodio implementando el algoritmo DQN. . . . .	80
7.1	IDEF-SIM del segundo caso de estudio. . . . .	86
7.2	Modelo de simulación del segundo caso de estudio. . . . .	87

7.3	Algoritmo $R^*DSCA$ . . . . .	91
7.4	Funciones para el parámetro $r_1$ . . . . .	92
7.5	Modelo de simulación del sub-problema A. . . . .	94
7.6	Modelo de simulación del sub-problema B. . . . .	95
7.7	Modelo de simulación del sub-problema C. . . . .	96
7.8	Modelo de simulación del sub-problema D. . . . .	97
7.9	Curvas de convergencia $\bar{D}_t$ para el sub-problema D. . . . .	100
8.1	Madurez tecnológica del sector metalmeccánico de Olavarría. . . . .	111
8.2	Madurez tecnológica de las empresas encuestadas, según su tamaño. . . . .	111
8.3	Tipos de tecnologías utilizadas por área funcional. . . . .	113
A.1	Paquetes en python de la integración de la prueba. . . . .	126
A.2	Diagrama UML de la prueba de concepto. . . . .	126

# Índice de tablas

5.1	Estadísticas de latencias medidas para el Experimento 1. . . . .	61
5.2	Estadísticas de latencias medidas para el Experimento 2. . . . .	62
5.3	Tipo de información a completar en la hoja de cálculo. . . . .	65
6.1	Parámetros probados en los algoritmos Q-learning, SARSA y DQN. . .	76
6.2	Parámetros adicionales probados para el algoritmo DQN. . . . .	79
6.3	Mejor rendimiento para el algoritmo DQN. . . . .	80
7.1	Datos básicos del segundo caso de estudio. . . . .	85
7.2	Estados para la configuración del $R^*DSCA$ . . . . .	92
7.3	Experimentos del $R^*DSCA$ para el sub-problema A. . . . .	94
7.4	Experimentos del $R^*DSCA$ para el sub-problema B. . . . .	96
7.5	Experimentos del $R^*DSCA$ para el sub-problema C. . . . .	97
7.6	Experimentos del $R^*DSCA$ para el sub-problema D. . . . .	98
7.7	Experimentos del DSCA para los cuatro sub-problemas. . . . .	99
7.8	Comparación entre DSCA y $R^*DSCA$ . . . . .	99
7.9	Resultados para un período de producción de 2 horas con $T=100$ . . .	102
7.10	Resultados para un período de producción de 8 horas con $T=100$ . . .	102
7.11	Reducción de desperdicios para un período de producción de 2 horas con $T=100$ . . . . .	102
7.12	Reducción de desperdicios para un período de producción de 8 horas con $T=100$ . . . . .	102
7.13	Resultados para un período de producción de 8 horas con $T=200$ . . .	102
7.14	Reducción de desperdicios para un período de producción de 8 horas con $T=200$ . . . . .	103
8.1	Ejes de análisis y dimensiones. . . . .	110



# Índice de algoritmos

1	Algoritmo de Seno Coseno . . . . .	30
2	Algoritmo Discreto de Seno Coseno . . . . .	31
3	Q - learning . . . . .	76
4	SARSA . . . . .	78
5	Deep Q - Network . . . . .	79
6	Q-learning embedded Sine Cosine Algorithm . . . . .	89
7	<i>R*DSCA</i> . . . . .	89



# Acrónimos

**AR** Aprendizaje por Refuerzo.

**CVRP** Capacitated Vehicle Routing Problem.

**DQN** Deep Q-learning Network.

**DSCA** Discrete Sine Cosine Algorithm.

**ERP** Enterprise Resource Planning.

**ESPM** Entorno de Simulación de Planta de Manufactura.

**GD** Gemelo Digital.

**I4.0** Industria 4.0.

**IA** Inteligencia Artificial.

**IoT** Internet de las Cosas.

**MES** Manufacturing Execution System.

**PyMEs** Pequeñas y Medianas Empresas.

**QLESCA** Q-learning embedded Sine Cosine Algorithm.

**RNA** Redes Neuronales Artificiales.

**ROS** Robot Operating System.

**SATDD** Sistema Autónomo de Toma de Decisiones.

**SCA** Sine Cosine Algorithm.

**SMA** Sistema Multiagente.

**TDD** Toma de Decisiones.

**TIC's** Tecnología de la Información y las Comunicaciones.

**TPS** Tecnomatix Plant Simulation.

**TSP** Travelling Salesman Problem.



*A mis padres, Estela y Walter, y a mi hermano Martín.*



# Agradecimientos

Esta tesis no hubiera sido posible sin el apoyo y la ayuda de muchas personas que han colaborado conmigo en estos últimos años de estudio y trabajo de investigación.

Quiero agradecer en primer lugar a mi Director, Dr. Mariano De Paula, por impulsarme siempre y contagiarme las ganas de investigar, por su ayuda en cada paso de esta carrera y por todas sus correcciones y charlas a lo largo de este camino. A mi Co-Director, Dr. Sebastián Villar, por su fundamental apoyo en el comienzo de la beca y por su buena predisposición y dedicación siempre.

A todo el grupo de trabajo que hemos formado y con el cual pudimos avanzar en muchas publicaciones y proyectos este tiempo: además de mis Directores ya mencionados, agradezco al Dr. Gerardo Acosta, al Ing. Ivo Perez Colo y a la Mg. Ing. Geraldina Roark por su colaboración, entusiasmo y por ser grandes compañeros.

A mis padres, que son fundamentales en mi vida, que siempre me han apoyado, guiado y acompañado.

A mi hermano Martín, por ser mi persona incondicional.

A la Facultad de Ingeniería de la Universidad Nacional del Centro de la Provincia de Buenos Aires, donde cursé mis estudios de grado, donde trabajo actualmente y donde me formo día a día. A la Comisión de Investigaciones Científicas de la Pcia. de Buenos Aires y al CONICET, por permitirme estudiar un Doctorado y por impulsar la investigación y la ciencia en Argentina.

A la Universidad Pública.

A todos ellos, muchas gracias.



# Resumen

El alto grado de incertidumbre y heterogeneidad del contexto contemporáneo presenta múltiples desafíos para las industrias manufactureras. En este contexto, la Cuarta Revolución Industrial ha proporcionado herramientas y tecnologías para ayudar a las empresas a enfrentar estos desafíos. La interconexión de sistemas productivos, la disponibilidad y análisis de datos en tiempo real y la automatización de procesos son algunas de las tecnologías que han surgido en este nuevo paradigma.

Además, la filosofía Lean, centrada en maximizar el valor para el cliente, ha demostrado ser una herramienta eficaz para abordar diversas problemáticas organizacionales. Esta metodología se basa en tres pilares fundamentales: la eliminación de desperdicios, la mejora continua y la optimización de procesos. La integración de estos principios con la evolución de la tecnología ha dado paso al concepto de Lean 4.0, un enfoque organizacional que busca mejorar la eficiencia y la eficacia de los procesos. Con Lean 4.0, las empresas pueden adaptarse a un entorno empresarial en constante cambio, mejorando la calidad, reduciendo costos y aumentando la satisfacción del cliente.

En este sentido, en la era actual, los sistemas de fabricación inteligente buscan alcanzar un alto nivel de flexibilidad y eficiencia. La manera de lograr esta meta es mediante la incorporación de automatización, junto con sistemas robotizados autónomos y sistemas de toma de decisiones capaces de reconfigurar las tareas en tiempo real según las demandas y exigencias externas.

Alcanzar esa flexibilidad mencionada reduciendo desperdicios no es una cuestión trivial, lo que implica, muchas veces, tener que tomar decisiones con objetivos contrapuestos. En otras palabras, se debe ser capaz de formular y resolver un problema de optimización multi-objetivo. En este sentido, la simulación de sistemas ha sido una herramienta valiosa para abordar estas problemáticas. Sin embargo, en la nueva era 4.0 el paradigma de gemelo digital ha ocupado su lugar. Dichos gemelos emplean la simulación en conjunto con una variedad de datos provenientes de diferentes equipos y sistemas físicos de planta, para mantener actualizados continuamente sus modelos digitales del mundo en un esquema de retroalimentación en un entorno virtual que facilita la toma de decisiones.

La heterogeneidad de hardware y software existente requiere del desarrollo de arquitecturas de software para que los diversos componentes se integren e interactúen mediante el intercambio de información. Para el rápido desarrollo de los gemelos digitales de plantas de fabricación que incluyen algunos dispositivos robóticos, es ventajoso utilizar simuladores de eventos discretos para el modelado de procesos de producción y los entornos virtuales existentes de los robots y dispositivos automáticos.

Uno de los aportes del trabajo de tesis es el diseño y desarrollo de una arquitectura software que integra un simulador de procesos de manufactura con

---

el sistema operativo de robots Robot Operating System. La propuesta permite intercambiar fácilmente información entre los componentes de la fábrica para el desarrollo rápido de gemelos digitales de sistemas de fabricación robotizados. Además, permite una integración sencilla entre el gemelo digital con un sistema autónomo de toma de decisiones. Dicha arquitectura se prueba con el simulador Tecnomatix Plant Simulation de Siemens<sup>®</sup> y la distribución libre de Robot Operating System, Melodic. Se presenta una instancia de esta arquitectura software para dos estudios de caso complejos relacionados al manejo de materiales típicos de plantas de fabricación, demostrando su fácil integración con un sistema autónomo de toma de decisiones.

Otro de los aportes más destacados del trabajo de tesis tiene que ver con el desarrollo de estrategias de toma de decisiones para abordar los problemas típicos de los sistemas de manufactura flexible. Las propuestas desarrolladas se basan principalmente en el paradigma de aprendizaje por refuerzos y el emergente (al menos, al momento de inicio de la estancia doctoral) aprendizaje por refuerzos profundo. Por otro lado, combinando ideas de estas metodologías, se propuso una metodología metaheurística basada en el reciente y poco explorado algoritmo de población Seno Coseno, para la resolución de problemas de toma de decisiones óptimas en dominios continuos. La versión desarrollada se diferencia principalmente en que puede ser aplicada en la resolución de problemas de optimización con variables discretas, lo que permite usarla en problemas de toma de decisiones típicos de sistemas de producción manufacturera.

Uno de los problemas típicos, y que cada vez recibe más atención, es el manejo de materiales y/o partes en proceso en los sistemas productivos. La motivación principal para abordar esta problemática eficientemente, en sistemas cada vez más complejos, radica en el hecho de que esta tarea no agrega valor. A su vez, es uno de los pilares relacionados con los ocho desperdicios de la metodología Lean. En este contexto, la propuesta desarrollada en esta tesis, justamente, puede aprovecharse para minimizar estos tipos de desperdicios en sistemas productivos que deben cambiar y adaptar la configuración de sus instalaciones para hacer frente a los cambios de requerimientos en materia de producción.

Los desarrollos de esta tesis fueron concebidos con el objetivo de ser tecnológicamente transferibles y extrapolables a plantas industriales de la región, con un enfoque particular en las Pequeñas y Medianas Empresas. Habiendo desarrollado en el grupo de investigación un relevamiento de las pequeñas y medianas empresas industriales metalmecánicas de la ciudad de Olavarría respecto a sus capacidades tecnológicas, se ha podido comprobar que hay un proceso incipiente de transformación digital en estas empresas y es necesario contribuir al avance de esta transición hacia la Industria 4.0.

# Abstract

The high uncertainty and heterogeneity of the contemporary context present multiple challenges for manufacturing industries. In this context, the Fourth Industrial Revolution has provided tools and technologies to help companies face these challenges. The interconnection of production systems, the availability and analysis of real-time data, and process automation are some technologies that have emerged in this new paradigm.

Furthermore, the Lean philosophy, focused on maximizing customer value, has proven to be an effective tool for addressing various organizational issues. This methodology is based on three fundamental pillars: waste elimination, continuous improvement, and process optimization. Integrating these principles with the evolution of technology has given rise to the concept of Lean 4.0, an organizational approach that seeks to improve the efficiency and effectiveness of processes. With Lean 4.0, companies can adapt to a constantly changing business environment, improving quality, reducing costs, and increasing customer satisfaction.

In this sense, in the current era, smart manufacturing systems seek to achieve a high level of flexibility and efficiency. The way to achieve this goal is through the incorporation of automation, along with autonomous robotic systems and decision-making systems capable of reconfiguring tasks in real time based on external demands and requirements.

Achieving this flexibility by reducing waste is not a trivial matter, which often involves making decisions with conflicting objectives. In other words, one must be able to formulate and solve a multi-objective optimization problem. In this sense, system simulation has been a valuable tool for addressing these problems. However, in the new 4.0 era, the digital twin paradigm has taken its place. These twins use simulation in conjunction with a variety of data from different equipment and physical plant systems to keep their digital models of the world continuously updated in a feedback loop in a virtual environment that facilitates decision-making.

The heterogeneity of existing hardware and software requires the development of software architectures for the integration and interaction of various components through information exchange. For the rapid development of digital twins of manufacturing plants that include some robotic devices, it is advantageous to use discrete event simulators for production process modeling, as well as the existing virtual environments of robots and automated devices.

One of the contributions of the thesis work is the design and development of a software architecture that integrates a manufacturing process simulator with the Robot Operating System. The proposal allows for easy information exchange between factory components for the rapid development of digital twins of robotic manufacturing systems. Furthermore, it enables simple integration of the digital twin with an autonomous decision-making system. This architecture is tested using

---

the Siemens<sup>®</sup>Tecnomatix Plant Simulation simulator and the free distribution of the Robot Operating System, Melodic. An instance of this software architecture is presented for two complex case studies related to typical materials handling in manufacturing plants, demonstrating its easy integration with an autonomous decision-making system.

Another of the most notable contributions of this thesis relates to developing decision-making strategies to address typical problems in flexible manufacturing systems. The proposals developed are primarily based on the reinforcement learning paradigm and emerging (at least at the start of the doctoral stay) deep reinforcement learning. Furthermore, combining ideas from these methodologies, a metaheuristic methodology based on the recent and little-explored Sine Cosine population algorithm was proposed for solving optimal decision-making problems in continuous domains. The developed version differs mainly because it can be applied to solving optimization problems with discrete variables, allowing it to be used in decision-making problems typical of manufacturing production systems.

One of the typical problems, and one that is receiving increasing attention, is the handling of materials and parts in production systems. The main motivation for addressing this issue efficiently, in increasingly complex systems, is that this task does not add value. In turn, it is one of the pillars related to the eight wastes of the Lean methodology. In this context, the proposal developed in this thesis can be used to minimize these types of waste in production systems that must change and adapt the configuration of their facilities to meet changing production requirements.

The developments in this thesis were conceived to be technologically transferable and extrapolated to industrial plants in the region, with a particular focus on small and medium-sized enterprises. Having conducted a survey of small and medium-sized metalworking industrial companies in the city of Olavarría regarding their technological capabilities, the research group has found that there is an incipient process of digital transformation in these companies, and it is necessary to contribute to advancing this transition toward Industry 4.0.

# Prefacio

El presente documento resume la investigación llevada a cabo durante los últimos cinco años para acceder al grado de Doctora en Ingeniería, Mención Tecnología Electromecánica, de la Facultad de Ingeniería de la Universidad Nacional del Centro de la Provincia de Buenos Aires (UNICEN). Esta investigación contó con la dirección del Dr. Ing. Mariano de Paula y la co-dirección del Dr. Ing. Sebastián A. Villar.

Este trabajo pudo realizarse gracias al apoyo y financiamiento de dos organismos públicos argentinos, mediante Becas Doctorales, los cuales son la Comisión de Investigaciones Científicas de la Provincia de Buenos Aires (CIC) y el Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET).

Las tareas desarrolladas se enmarcan en las actividades de investigación llevadas a cabo en el Núcleo INTELYMEC, perteneciente al Centro de Investigaciones en Física e Ingeniería del Centro de la Provincia de Buenos Aires (CIFICEN), y en el Departamento de Ingeniería Industrial, ambos pertenecientes al ámbito de la Facultad de Ingeniería de la UNICEN; donde se ha realizado un gran número de trabajos y actividades relacionadas con la industria de manufactura flexible y la robótica.

Entre los proyectos destacados se encuentran: “Sistemas de control inteligente”, “Modelización y simulación como apoyo a la TDD en sistemas industriales complejos” y “Métodos de simulación en la Industria 4.0 como apoyo a la toma de decisiones” del Programa de Incentivos S.P.U. Además, los proyectos de vinculación y transferencia tecnológica financiados por el CONICET “Relevamiento de capacidades tecnológicas para la transición digital de PyMEs industriales del sector Metalmecánico” y “Gemelos digitales para asistencia en la toma de decisiones en sistemas de manufactura del sector PyME basados en técnicas mixtas de simulación e inteligencia artificial”, como así también “Metodologías para la Toma de Decisiones autónomas de Sistemas Ciberfísicos de Producción Flexible” financiado por el FONCYT, “Robots autónomos para automatización de batimetrías e inspecciones” financiado por FONARSEC–EMPRETECNO 2016 y “Vehículo Autónomo Submarino” entre la empresa Y-TEC y el grupo INTELYMEC, entre otros.

Gracias al trabajo conjunto con el equipo del Laboratorio de Automatización y Robótica Inteligente del Núcleo INTELYMEC y la colaboración con equipos de otras Universidades, se han podido desarrollar numerosas publicaciones en Congresos y Revistas nacionales e internacionales, que contribuyeron a la divulgación y validación de lo realizado.

Cabe mencionar que desde el inicio del Doctorado, a fines del año 2019, hasta el día de hoy, los progresos tecnológicos mundiales han sido notables. Estos grandes cambios se han visto acelerados aún más por la pandemia global vivida. La inteligencia artificial es hoy una herramienta al alcance de todos. En este contexto, las organizaciones en general, y las empresas en particular, han avanzado

---

en forma considerable en su digitalización y en la mejora de sus procesos. Esto confirma la hipótesis inicial planteada en el plan de trabajo de esta tesis, cuyo objetivo general fue desarrollar un sistema autónomo de toma de decisiones para industrias de manufactura flexibles. Creemos que los aportes realizados en esta investigación pueden ser un soporte y un puntapié para muchas industrias de la región, especialmente pequeñas y medianas empresas, que pueden verse beneficiadas en esta transición hacia el Paradigma 4.0.

# Parte I

## Contexto y Estado del arte



# Capítulo 1

## Introducción

La industrialización ha cambiado el mundo desde finales del siglo XVIII. Desde ese momento hasta hoy, se han vivido cuatro revoluciones industriales, que han traído diversos desafíos a todo nivel, a escala mundial.

En Gran Bretaña, la primera de estas revoluciones permitió la producción en masa utilizando energía hidráulica y a vapor en lugar de energía puramente humana y animal. Los productos finales ya no eran exclusivamente artesanales, sino que se producían con maquinaria [IBM, 2018]. Este hito se considera un punto de quiebre en la historia de la humanidad, ya que ha impactado en todos los aspectos de la vida diaria, como la economía, la política, la sociedad, la salud y el transporte. Además, ha permitido el desarrollo del comercio y el auge de los negocios [Groumpos, 2021].

Un siglo después, la segunda revolución industrial introdujo las líneas de montaje y el uso de gas, petróleo y energía eléctrica. Los avances en las comunicaciones por teléfono y telégrafo, junto con estas nuevas fuentes de energía, aceleraron la producción en masa y trajeron un cierto grado de automatización a los procesos de manufactura. Un caso paradigmático de la época fue el de los automóviles, con Henry Ford como ejemplo de estos cambios en el proceso productivo, donde se podía fabricar más rápido y a menor costo.

Las computadoras, las telecomunicaciones avanzadas y el análisis de datos se incorporaron a los procesos manufactureros durante la tercera revolución industrial, que comenzó a mediados del siglo XX luego de la Segunda Guerra Mundial. La digitalización de las fábricas comenzó con la incorporación de controladores lógicos programables (PLC) para ayudar a automatizar algunos procesos y recopilar y compartir datos. Muchos elementos analógicos se digitalizaron, aparecieron los semiconductores e Internet cambió drásticamente el comercio mundial [Groumpos, 2021].

Finalmente, en este nuevo siglo la cuarta revolución industrial, también conocida como Industria 4.0 (I4.0), se caracteriza por una mayor automatización y el uso de máquinas y fábricas inteligentes. Los datos recopilados ayudan a producir bienes en toda la cadena de valor de manera más eficiente y productiva. Para mejorar la flexibilidad, la personalización masiva permite a los fabricantes satisfacer mejor las demandas de los clientes [IBM, 2018]. El término de I4.0 se acuñó en Alemania hace aproximadamente diez años en relación a una estrategia gubernamental que buscó impulsar a la industria y a los sistemas ciberfísicos, para promover la digitalización en la producción. [Barbieri et al., 2022].

Por lo tanto, podemos resumir a las revoluciones industriales explicadas

anteriormente en los siguientes términos: la primera como una revolución mecánica, la segunda como una eléctrica, la tercera, una revolución de la automatización y la cuarta, de la digitalización.

En definitiva y según lo mencionado, los modos de producción actuales se están moviendo hacia una filosofía de producción personalizada en la que se acentúa cada vez más la utilización de recursos y servicios rentados dentro de los sistemas productivos para tratar de conseguir una alta flexibilidad y capacidad de reconfiguración [Kusiak, 2018]. Esto hace que los sistemas sean cada vez más complejos y con una mayor tendencia a incorporar otros sub-sistemas automatizados y/o robóticos para la realización de una multiplicidad de tareas a lo largo de todo el proceso productivo [Zhong et al., 2017]. Sin embargo, en esta nueva configuración de sistemas de producción, existen inevitablemente interferencias dinámicas asociadas con diversas fuentes (por ejemplo, la aleatoriedad de los pedidos, la baja repetitividad del proceso de producción, el uso de servicios y recursos externos, etc.) que hacen que la producción sincronizada pueda verse perturbada en detrimento de la eficiencia y efectividad de respuesta del sistema productivo [Zolotová et al., 2020].

Un común denominador que se plantea en esta nueva era de digitalización de los sistemas flexibles de producción es la disposición de una gran cantidad de datos, provenientes de fuentes heterogéneas, para la Toma de Decisiones (TDD). Sin embargo, para el desarrollo de sistemas artificiales competentes para la TDD en tiempo real en piso de planta, es imperioso el uso de técnicas de procesamiento y extracción de la información implícita en los grandes volúmenes de datos generados. Estos sistemas de manufactura representan un caso especial de sistemas dinámicos complejos, muchas veces compuestos por una multiplicidad de elementos, como por ejemplo Robots autónomos, Simulación computacional, Sistemas ciberfísicos, Internet de las cosas Internet de las Cosas (IoT), Ciberseguridad, Tecnología de la Información y las Comunicaciones (TIC's), Manufactura aditiva, Realidad aumentada, Ciencia de grandes datos e Inteligencia Artificial (IA), siempre considerando una perspectiva sustentable [Hartini and Ciptomulyono, 2015, Stock and Seliger, 2016]. Para una gestión eficiente y eficaz, es necesario coordinar dinámicamente los diversos elementos de estos sistemas. Es aquí donde el paradigma de manufactura esbelta o *Lean manufacturing* puede conjugarse con el de la I4.0 de manera conceptual de modo tal que todos los elementos correctamente coordinados pueden lograr cumplir con las metas de la filosofía de manufactura esbelta, como lo son la optimización de la producción y la reducción de desperdicios con desempeño sustentable mediante el planteo de objetivos tanto económicos, sociales como de impacto ambiental [Mayr et al., 2018]. Estos dos enfoques podrían conjugarse en una simbiosis positiva para dar respuesta a los requerimientos en términos de flexibilidad de los sistemas de producción con cadenas de suministros cada vez más complejas [Mrugalska and Wyrwicka, 2017].

En este sentido, en las configuraciones actuales de los sistemas productivos modernos intervienen una variedad de equipos y recursos de software y hardware, además de recursos humanos, que deben actuar coordinadamente y de manera autónoma para hacer frente a la variación y alta frecuencia de las tareas de producción causadas por los requisitos personalizados de los clientes [Baskaran et al., 2019]. En este contexto, el paradigma de simulación clásico ha dejado de ser el componente central para la formulación de sistemas autónomos de TDD debido,

principalmente, a su característica estática. Por ello, y dada la fuerte irrupción del IoT industrial [González García et al., 2018, Dejene et al., 2020] junto a los sistemas ciberfísicos [Ding et al., 2020], su lugar ha sido ocupado por el concepto de Gemelo Digital (GD). Un GD es una representación digital integral de un producto o proceso individual, que incluye las propiedades, condición y comportamiento del objeto de la vida real a través de modelos y datos. Es un conjunto de modelos realistas que pueden simular su comportamiento real en el entorno implementado. El GD se desarrolla junto con su gemelo físico y sigue siendo su contraparte virtual durante todo el ciclo de vida del producto [Haag and Anderl, 2018].

Si bien, desde hace tiempo, existe una tecnología estable en lo que se refiere a cada uno de los componentes individuales que conforman un sistema de manufactura flexible, la integración entre las diferentes tecnologías para el desarrollo de los GDs es una cuestión que no ha sido abordada profusamente [Wang et al., 2020] y que es de vital relevancia para el desarrollo de propuestas de sistemas autónomos de manufactura flexible.

## 1.1 Motivación

La presente tesis aborda como eje general la **problemática** inherente a la necesidad de generar nuevas propuestas para el desarrollo de sistemas de TDD en el segmento de la manufactura flexible dentro del contexto de la I4.0.

Particularmente en esta investigación, se pretendió realizar aportes concretos para el desarrollo e integración de un Sistema Autónomo de Toma de Decisiones (SATDD) basado en métodos y técnicas de simulación de eventos discretos e IA. En el desarrollo de tales sistemas, se buscó que sean tecnológicamente transferibles y extrapolables a plantas industriales de la región, poniendo especial atención en las pequeñas y medianas empresas.

Las Pequeñas y Medianas Empresas (PyMEs) productivas representan un caso emblemático de sistemas de producción flexibles que requieren estrategias dinámicas de TDD dado que, frecuentemente, se enfocan en la fabricación de pequeños lotes o productos totalmente personalizados y necesitan reconfigurarse periódicamente para satisfacer sus necesidades. Adicionalmente, la exigencia de trabajar en un entorno incierto y fluctuante les demanda cada vez más y mejores habilidades de integración, flexibilidad y adaptación para alcanzar el nivel de resistencia necesario que les permita manejar situaciones adversas imprevistas. La escala y variedad de los desafíos que enfrentan las organizaciones actuales va en aumento y la innovación es una de las maneras de avanzar en el mercado.

Sin embargo, en el sector productivo PyME de nuestro país, si bien cuentan con personas idóneas y equipos automatizados responsables de la realización de determinadas tareas, como pueden ser soldadura, cortes de material, manipulación, etc., las mismas son ejecutadas de manera aislada sin demasiada o nula interacción con la información del contexto [Mon and Del Giorgio, 2022]. Además, en el empresariado PyME subyace el dilema de que los robots y equipos automáticos deben ser usados para realizar trabajos de producción en serie, o para grandes lotes, ya que cada vez que deben reprogramarse por parte de un experto para realizar otras tareas, debe invertirse una cantidad de dinero considerable, ocasionando, además, excesivos tiempos de inactividad [Mon and Del Giorgio, 2023]. Entonces, repensar los sistemas productivos de las PyMEs para aprovechar los avances tecnológicos actuales

propios de esta nueva era de “*fábricas inteligentes*” [Kusiak, 2018] e incorporar sistemas autónomos de TDD puede resolver ese dilema a la vez que se contribuye notablemente a mejorar su competitividad en términos de eficiencia y resiliencia. Sin embargo, para lograr esta transformación se requiere imperiosamente de soluciones innovadoras que, justamente, provengan del ámbito de las ingenierías.

Vale la pena mencionar que en la región de influencia del lugar de trabajo existe un número importante de empresas de producción industrial que potencialmente obtendrán los beneficios de la incorporación de las novedosas propuestas desarrolladas. La actividad industrial de la región está fuertemente conformada por la industria cementera, la industria cerámica y también por la industria metalmeccánica y de producción de máquinas agrícolas, las cuales no son ajenas al fenómeno global de transformación digital. También es para destacar la existencia de un número importante de pequeñas y medianas empresas automatizadas que pertenecen principalmente al rubro metalúrgico (fábricas de calefactores, de aberturas de aluminio, de carrocerías de camiones, entre otras), las cuales son potenciales receptoras de los futuros desarrollos, como lo demuestran estudios previos [Wagner et al., 2017, Enke et al., 2018, Alkhoraif et al., 2019].

## 1.2 Objetivos general y específicos

La **hipótesis** de este trabajo de investigación presume que en un contexto de incertidumbre y variabilidad las tecnologías de información, junto con los nuevos métodos de aprendizaje bio-inspirados, posibilitan el desarrollo de novedosas metodologías para el procesamiento, fusión e integración de datos heterogéneos provenientes de diferentes fuentes de información, para dar lugar a propuestas de sistemas de decisión autónomos capaces de dar soporte a sistemas de manufactura flexible en tiempo real y permitirles adaptarse a las exigencias de entornos reales, complejos y cambiantes.

La presente tesis tiene como **objetivo general** estudiar las potencialidades de metodologías del campo de la IA y la simulación de sistemas de producción para el desarrollo de una propuesta de un Sistema Autónomo de Toma de Decisiones (SATDD) para industrias de manufactura flexible, teniendo en cuenta las premisas del paradigma Lean 4.0, compatible especialmente con las posibilidades tecnológicas de las pequeñas y medianas empresas de la región.

Los **objetivos particulares** son los siguientes:

- Proponer un sistema de TDD autónomo capaz de responder en tiempo real en un proceso de manufactura flexible, incorporando métodos de IA basados en simulación.
- Desarrollar una propuesta metodológica de asistencia inteligente para la reducción de desperdicios en sistemas de manufactura, alineada con la metodología Lean y especialmente compatible con las posibilidades de las PyMEs.
- Realizar pruebas comparativas de distintas aproximaciones generadas.
- Implantar los mejores resultados en la celda experimental disponible en el lugar de trabajo y en diversos casos de estudio, para validar la investigación.

- Transferir el conocimiento al medio mediante su publicación en revistas y congresos nacionales e internacionales relacionados a la temática.

### 1.3 Contribuciones de la tesis

Como consecuencia del trabajo realizado, se han conseguido validar y divulgar los resultados obtenidos en la comunidad científica nacional e internacional a través de un número de publicaciones que se listan a continuación en orden cronológico. Estas publicaciones están directamente relacionadas con el objetivo de la tesis y/o forman parte de la presente memoria.

#### Publicaciones en revistas internacionales:

1. Alicia Mon; Horacio Del Giorgio; Geraldina Roark; **Carolina Saavedra Sueldo**; Ivo Pérez Colo; Mariano De Paula. "SME users in Industry 4.0. Study of the Metalworking Industry in Argentina". CLEI Electronic Journal: Latin-american Center for Informatics Studies. Vol.28 n°1. 2025. e-issn 0717-5000. <https://doi.org/10.19153/cleiej.28.1.0>
2. **Carolina Saavedra Sueldo**; Ivo Perez Colo; Mariano De Paula; Sebastián Villar; Gerardo G. Acosta. "Simulation-Based Metaheuristic Optimization Algorithm for Material Handling". Journal of Intelligent Manufacturing. 2024. <https://doi.org/10.1007/s10845-024-02327-0>
3. Ivo Perez Colo; **Carolina Saavedra Sueldo**; Mariano De Paula; Gerardo G. Acosta. "Intelligent approach for the industrialization of deep learning solutions applied to fault detection". Expert Systems with Applications. Vol. 233. 2023. ISSN 0957-4174. <https://doi.org/10.1016/j.eswa.2023.120959>
4. **Carolina Saavedra Sueldo**; Ivo Perez Colo; Mariano De Paula; Sebastián Villar; Gerardo G. Acosta. "ROS-based architecture for fast digital twin development of smart manufacturing robotized systems". Annals of Operation Research. 322, 75–99. Springer. 2022. ISSN 0254-5330. <https://link.springer.com/article/10.1007/s10479-022-04759-4>
5. **Carolina Saavedra Sueldo**; Mariano De Paula; Sebastián Villar; Gerardo G. Acosta. "Integration of ROS and Tecnomatix for the development of digital twins based decision-making systems for smart factories". IEEE Latin America Transactions. Vol.19 n°9. pp. 1546 - 1555. 2021. ISSN 1548-0992. <https://latamt.ieeer9.org/index.php/transactions/article/view/4925>

#### Publicaciones en congresos nacionales e internacionales:

1. Geraldina Roark; **Carolina Saavedra Sueldo**; Ivo Pérez Colo; Mariano De Paula; Gerardo G. Acosta. "Impulsando la transformación digital. Un caso de vinculación entre la universidad y PyMEs metalmecánicas". Jornadas de Vinculación Universidad-Industria (JUI), 53 Jornadas Argentinas de Informática (JAIIO). Bahía Blanca. 2024. <https://revistas.unlp.edu.ar/JAIIO/article/view/17939>

2. Alicia Mon; Horacio Del Giorgio; Geraldina Roark; **Carolina Saavedra Sueldo**; Ivo Pérez Colo; Mariano De Paula. “Usuarios PyMEs en la Industria 4.0. Estudio de la Industria Metalmeccánica en Argentina”. X Jornadas Iberoamericanas de Interacción Humano-Computadora JIHCI2024. Universidad Tecnológica de Pereira, Colombia. 2024.
3. Geraldina Roark; Manuel Ezequías Vázquez; Ivo Perez Colo; **Carolina Saavedra Sueldo**; Daniela Marisol Berdun. “Madurez Tecnológica en PyMEs Industriales de Olavarría: perspectivas de su preparación para la Transformación Digital”. XVI Congreso Internacional de Ingeniería Industrial - AACINI - UTN FRSN. San Nicolás. 2023. <https://riii.fi.mdp.edu.ar/index.php/AACINI-RIII/article/view/107>
4. **Carolina Saavedra Sueldo**; Ivo Perez Colo; Mariano De Paula; Sebastián Villar; Gerardo G. Acosta. “Optimización y control del flujo de materiales en procesos de producción flexibles utilizando aprendizaje profundo”. XIX Reunión de Trabajo en Procesamiento de la Información y el Control. Instituto de Automática, Facultad de Ingeniería, Universidad Nacional de San Juan, San Juan, Argentina. 2021. ISBN: 978-987-88-2891-6.
5. **Carolina Saavedra Sueldo**; Sebastián Villar; Mariano De Paula; Silvia Urrutia; Gerardo G. Acosta. “Integración de ROS y Tecnomatix para el desarrollo de gemelos digitales en sistemas de manufactura flexible”. International Conference of Production Research - Americas. Universidad Nacional del Sur, Bahía Blanca, Argentina. 2021. ISSN 2619-1865.

#### **Libros:**

1. Transformación Digital en PyMEs Metalmeccánicas de Olavarría : desafíos y oportunidades. Geraldina Roark, Carolina Saavedra Sueldo, Ivo Perez Colo, Mariano De Paula y Gerardo G. Acosta. 1a ed - Tandil : Universidad Nacional del Centro de la Provincia de Buenos Aires, 2024. Libro digital, PDF. ISBN 978-950-658-620-1. <https://www.ridaa.unicen.edu.ar/handle/123456789/3960>

#### **Desarrollos:**

Otro aspecto a destacar es el desarrollo, en base al presente estudio, de una librería Python llamada *Simulai*, la cual es una biblioteca libre que facilita el uso de técnicas de IA para la optimización de variables derivadas de la simulación de sistemas de fabricación flexibles. El link del paquete es el siguiente: <https://pypi.org/project/simulai/>, mientras que la documentación está disponible en <https://simulai.readthedocs.io/en/latest/?badge=latest>

#### **Otras contribuciones:**

Además de las publicaciones antes mencionadas, los métodos y técnicas empleados en esta investigación han sido útiles para aportar en otros estudios y generar publicaciones que, aunque no están directamente vinculadas al propósito de esta tesis, es relevante resaltar.

1. **Carolina Saavedra Sueldo**; Bruno Boato; Mariano de Paula; Luis Avila. “Optimal energy allocation using Sine Cosine Algorithm for electrical microgrids”. Expert Systems with Applications (Under review).
2. Bruno Boato; **Carolina Saavedra Sueldo**; Luis Avila; Mariano de Paula. “An improved Soft Actor-Critic strategy for optimal energy management”. IEEE Latin America Transactions. Vol 21 n°9. pp. 958–965. 2023. ISSN 1548-0992. <https://latamt.ieeer9.org/index.php/transactions/article/view/7791>
3. Ivo Perez Colo; **Carolina Saavedra Sueldo**; Mariano De Paula; Geraldina Roark; Sebastián Villar; Gerardo G. Acosta. “Sistema inteligente para la detección de fallas basado en redes profundas auto-ajustables”. 2022 IEEE Biennial Congress of Argentina - ARGENCON. 2022. doi: 10.1109/ARGENCON55245.2022.9939789.

## 1.4 Estructura de la tesis

El presente documento se encuentra organizado en tres partes. La primera parte es el marco introductorio del estudio y contextualiza la problemática. La segunda parte contiene el desarrollo del Sistema Autónomo de Toma de Decisiones (SATDD) y su validación. Asimismo, se presentan los casos de estudio utilizados para validar la investigación, junto con un algoritmo desarrollado en este trabajo para optimizar el manejo de materiales. Finalmente, la tercera parte expone las investigaciones en curso, las conclusiones de la tesis y las perspectivas para futuros trabajos.

A continuación, se detalla la organización de los capítulos restantes.

- El Capítulo 2 presenta una revisión exhaustiva de los trabajos relacionados y el estado del arte en la temática abordada. Se analiza la evolución de los sistemas de producción y su gestión en los últimos años, identificando las transformaciones más relevantes impulsadas por la digitalización y la automatización industrial. Además, se examina cómo las tecnologías emergentes han impactado en los modelos de producción y en la TDD. Se realiza también una integración entre técnicas y metodologías consolidadas con las nuevas tendencias tecnológicas, demostrando su compatibilidad y las sinergias que pueden generarse para alcanzar los objetivos estratégicos en el ámbito empresarial.
- El Capítulo 3 aporta los antecedentes metodológicos del estudio. Se abordan temas clave como la manufactura flexible, el modelado de sistemas y la simulación de eventos discretos, destacando su papel en la optimización de procesos industriales. Se incluyen también los GDs, su definición, evolución y aplicaciones en distintos sectores productivos. Adicionalmente, se presentan las técnicas de optimización metaheurísticas y los enfoques de IA, analizando su potencial para mejorar la TDD empresariales. Se establece una conexión entre estas herramientas y su impacto en la eficiencia operativa, la reducción de costos y la mejora en la capacidad de respuesta ante cambios en el entorno.
- El Capítulo 4 detalla los componentes necesarios para el desarrollo de un GD y el posterior sistema de TDD vinculado. Se explican los entornos de

simulación empleados y su relación con la plataforma Robot Operating System (ROS), destacando sus características principales y los beneficios que aportan a la automatización y control de procesos. Se describen también modelos hipotéticos de planta y la consecuente comunicación entre los distintos agentes del sistema industrial.

- El Capítulo 5 desarrolla la propuesta de un SATDD basado en un GD, explicando detalladamente la arquitectura diseñada y la metodología seguida para su implementación. Se incluyen las fases de diseño y desarrollo del sistema, así como la validación de la solución en un entorno experimental. Para ello, se presenta su aplicación en la celda de manufactura de la Facultad de Ingeniería de la UNICEN, proporcionando un análisis detallado de la programación y configuración del sistema.
- El Capítulo 6 describe la aplicación del SATDD en el primero de los casos de estudio analizados, donde se aborda el problema del manejo de materiales en el piso de planta. Contiene su respectiva introducción, desarrollo y resultados obtenidos utilizando algoritmos de Aprendizaje por Refuerzo (AR). Se demuestra en este capítulo la viabilidad y fácil integración del sistema desarrollado en un entorno productivo.
- El Capítulo 7 expone el segundo caso de estudio, en el cual se desarrolla un algoritmo propio que se añade al SATDD para optimizar el manejo de materiales e influir en los desperdicios del sistema productivo, combinando las tecnologías de la I4.0 con la filosofía Lean. Se detallan los resultados obtenidos en pruebas experimentales, evidenciando mejoras significativas en la productividad y sustentabilidad del proceso industrial.
- El Capítulo 8, por su parte, presenta investigaciones realizadas por el equipo de trabajo en este último período relacionadas a la transición digital de las PyMEs industriales de la ciudad. Se analiza el impacto de la digitalización en este sector y se destacan los desafíos y oportunidades que enfrentan en su camino hacia la transformación digital. Se demuestra la importancia de la propuesta de tesis en este contexto, mostrando cómo su implementación puede facilitar la adopción de nuevas tecnologías y mejorar la competitividad de las empresas mencionadas.
- El Capítulo 9 finalmente sintetiza las conclusiones del estudio, destacando los aportes de la investigación y su relevancia en el ámbito de la manufactura inteligente. Se discuten las contribuciones del trabajo, así como sus aplicaciones en escenarios industriales reales. Además, se presentan posibles desarrollos futuros relacionados a la temática, que incluyen sistemas multi-agente y simulación automática para la industria.

# Capítulo 2

## Estado del Arte

### 2.1 Cambio de paradigma: Hacia la Industria 4.0

Las empresas manufactureras actuales se enfrentan a una amplia gama de obstáculos y desafíos. La producción tiende a ser personalizada y las demandas cambian y se diversifican rápidamente. Además, la velocidad del cambio en el medio ambiente externo está aumentando vertiginosamente. Factores como el cambio climático, el triple impacto, las pandemias globales, la revolución digital, los cambios culturales, entre otros, enfrentan a las organizaciones a una difícil realidad. En este contexto, el paradigma de la I4.0 se refiere a un conjunto de transformaciones productivas e institucionales [Kusiak, 2018]. En este ámbito, surge el concepto de “*fábricas inteligentes*”, el cual implica que objetos físicos, máquinas, productos e incluso piezas se integran en las redes de información de la empresa y se comunican en tiempo real de forma horizontal entre sí, es decir, en diferentes áreas de la misma empresa, y verticalmente con clientes, usuarios y proveedores, por fuera de la organización.

Como se mencionó anteriormente, la I4.0 también incluye tecnologías tales como los sistemas ciberfísicos, IoT, análisis de *big data*, computación en la nube, fabricación en la nube, realidad virtual y aumentada, robótica y simulación [Ullah et al., 2024]. Estas tecnologías conforman un ecosistema digital que busca optimizar los procesos industriales mediante la automatización avanzada, la TDD basada en datos y la interconexión de dispositivos inteligentes. Por lo tanto, considera un amplio conjunto de tecnologías, tanto software como hardware, que operan en sistemas integrados [Benitez et al., 2023], con el objetivo de mejorar la eficiencia operativa, reducir costos y aumentar la flexibilidad de la producción para responder de manera ágil a las necesidades del mercado.

Uno de los principales objetivos de este paradigma es lograr una mayor productividad, colaboración, agilidad, adaptabilidad, trazabilidad, calidad y seguridad en los procesos de manufactura. La convergencia de estas tecnologías permite no solo optimizar la producción, sino también desarrollar modelos de negocio más sostenibles y centrados en el cliente. En este sentido, la interconectividad y la digitalización de la manufactura facilitan la implementación de estrategias de producción *Just in Time*, mantenimiento predictivo y optimización de la cadena de suministro. Asimismo, la integración de la IA y el aprendizaje automático en los sistemas de manufactura permite mejorar la TDD y predecir fallos o ineficiencias en la producción con una precisión sin precedentes [Perez Colo et al., 2023, Galindo-

Salcedo et al., 2022].

En este contexto, la evolución continúa, y en la actualidad ya se habla de la Industria 5.0 como un nuevo paradigma, que incluye conceptos como una visión humanizada de la industria, la fabricación circular y sostenible, y las necesidades futuras de la sociedad [Lu et al., 2023, Pizoñ and Gola, 2023]. Se espera que tanto la I4.0, impulsada por la tecnología, como la Industria 5.0, impulsada por la sociotécnica, continúen su progreso lado a lado, revolucionando la fabricación [Barata and Kayser, 2023]. Este nuevo enfoque pone un énfasis particular en la colaboración entre humanos y máquinas, asegurando que la automatización y la IA sean utilizadas simbióticamente para mejorar las condiciones laborales y fomentar una producción más ética y responsable. Hay autores que plantean que los paradigmas de I4.0 e I5.0 pueden coexistir en un híbrido sustentable que asuma que la inteligencia colectiva es el motor de la toma de decisiones eficiente, centrada en el ser humano, responsable y resiliente [Golovianko et al., 2023]. Este modelo de manufactura híbrido busca aprovechar las capacidades de las máquinas inteligentes sin descuidar el valor de la creatividad, la intuición y la ética humana en la TDD estratégicas dentro de las organizaciones.

Por otra parte, los avances acelerados en robótica [Chung et al., 2025, Kashyap and Parhi, 2025], que ya permiten la interacción fluida entre robots y su comunicación tanto entre sí como con las personas [Kim et al., 2024], abren el camino hacia un futuro próximo en el que los sistemas de producción incorporarán una amplia gama de robots colaborativos y especializados. Estos sistemas estarán capacitados para ejecutar diversas tareas, desde operaciones de procesamiento hasta actividades relacionadas con el manejo, transporte y gestión de materiales y productos en curso, incrementando así la autonomía y versatilidad de las plantas industriales. La automatización robótica ha evolucionado significativamente en las últimas décadas, permitiendo no solo mejorar la precisión y velocidad de las operaciones industriales, sino también aumentar la seguridad en entornos de manufactura peligrosos. También es un hecho que la robótica colaborativa ha venido para quedarse y es uno de los pilares de la industria 5.0 [Demir et al., 2019]. Los robots colaborativos o cobots están diseñados para interactuar con los trabajadores de manera segura, facilitando la automatización flexible y personalizada sin la necesidad de estructuras de seguridad rígidas.

Hoy en día, es frecuente que los fabricantes de robots y equipos automatizados proporcionen simuladores informáticos de sus productos, que son la base para crear los GDs de cada equipo (entidad) en cuestión. Sin embargo, muchas de estas soluciones disponibles de forma aislada no ofrecen una solución integral desde la perspectiva del sistema de fabricación industrial. La diversidad y complejidad de cada una de las entidades que intervienen en un proceso productivo, hacen que los simuladores diseñados para modelar completamente un sistema productivo puedan no reflejar el sistema en su conjunto con un grado suficiente de fidelidad, ya que tienden a simplificar la dinámica de las entidades que lo componen. Por lo tanto, el GD de todo el sistema de fabricación industrial que se pretende crear podría verse perjudicado. Por esta razón, para la creación de un GD de un proceso productivo, sería conveniente aprovechar las cualidades de los simuladores de plantas de manufactura (en nuestro caso, simuladores de eventos discretos) con cada uno de los simuladores específicos de cada entidad o agente que interviene en el proceso productivo. Pensar de esta manera es una opción ventajosa ya

que podría aprovecharse para crear el GD del proceso de fabricación en su conjunto. Particularmente, la simulación de eventos discretos permite modelar con mayor precisión la variabilidad y la aleatoriedad de los procesos industriales, proporcionando un entorno virtual más realista para la experimentación y la optimización de estrategias de producción. Entonces, pensar en el desarrollo de una arquitectura de software basada en ROS que permita una fácil integración de las entidades existentes entre sí y un simulador de procesos de fabricación parece ser una opción prometedora, y que no ha sido propuesta en la literatura, para el desarrollo de GD de sistemas de fabricación inteligente. De esta manera, permite por una parte controlar el flujo de interacción entre los diferentes agentes que intervienen en el proceso productivo y emplear fácilmente técnicas de IA para la TDD sobre las acciones que deba realizar cada agente. De esta manera, se pueden conseguir sistemas de producción con mejores capacidades de adaptación a los eventos y cambios que surjan del mercado (por ejemplo, por la demanda de productos personalizados) y/o del propio entorno operativo (por ejemplo, ocurrencia de fallas, paradas no planificadas, etc.).

## 2.2 Lean 4.0 y el manejo eficiente de materiales en plantas manufactureras

Las tendencias actuales en la manufactura y la gestión de operaciones plantean desafíos significativos en la administración eficiente de tareas dentro de los sistemas productivos. En un contexto dinámico y altamente competitivo, típico de nuestros tiempos, los sistemas productivos actuales y venideros deberán poseer un alto grado de flexibilidad, adaptabilidad y capacidad de respuesta para afrontar las demandas del presente y del futuro. La capacidad de una organización para optimizar sus procesos de producción no solo influye en su eficiencia operativa, sino que también es determinante para su sostenibilidad y competitividad en mercados cada vez más exigentes. Por ello, la reducción sistemática de desperdicios en cada ciclo productivo es una condición imprescindible para garantizar la mejora continua y la optimización de los recursos disponibles.

El concepto de *Lean Manufacturing* o Manufactura Esbelta, por su traducción al español, se ha consolidado como una filosofía integral de gestión empresarial y, a su vez, como una estrategia de largo plazo [Womack et al., 2007]. Este enfoque, desarrollado en la década de 1950 por la empresa Toyota en Japón, se basa en el principio de “*hacer más con menos*”, lo que implica una utilización más eficiente de los recursos sin comprometer la calidad del producto o servicio final. Lean Manufacturing se centra en la identificación y eliminación de todas aquellas actividades que no aportan valor, con el fin de reducir costos, mejorar la calidad y aumentar la eficiencia global del sistema productivo [Alkhoraiif et al., 2019].

La metodología Lean se focaliza en las actividades que generan valor real para el cliente. Para lograrlo, Lean Manufacturing incorpora diversas herramientas y métodos de ingeniería y gestión que han demostrado su eficacia en la optimización de procesos. Entre las más destacadas se encuentran: 5S, Just in Time (JIT), Mantenimiento Productivo Total (TPM), Mejora Continua (Kaizen), Gestión de la Calidad Total (TQM), Cero Defectos, Kanban, Estandarización de tareas y Value Stream Mapping (VSM), entre otros [Shah and Patel, 2018, Akkari and Valamede,

2020]. Estas técnicas no solo mejoran la eficiencia operativa, sino que también contribuyen a generar una cultura organizacional orientada hacia la excelencia y la mejora continua.

Uno de los pilares fundamentales de Lean Manufacturing es la identificación y minimización de los ocho tipos de desperdicios que pueden afectar la eficiencia de un sistema productivo. Originalmente, Taiichi Ohno estableció siete desperdicios clave en el siglo XX [Ohno, 1988], a los cuales se ha sumado en años recientes el desperdicio asociado al talento humano no aprovechado. Estos desperdicios incluyen la sobreproducción, las esperas, los movimientos innecesarios, el transporte, el exceso de inventario, los defectos en los productos, el sobreprocesamiento y la subutilización de las capacidades y habilidades del personal.

La reducción de estos desperdicios tiene un impacto directo en la eficiencia operativa, la reducción de costos y la capacidad de respuesta de la organización frente a los cambios del entorno. La identificación temprana de estas ineficiencias constituye el primer paso para implementar estrategias de optimización que permitan alcanzar los objetivos de productividad y sostenibilidad. En este sentido, la aplicación de Lean Manufacturing no solo mejora los indicadores de desempeño interno, sino que también fortalece la competitividad de la empresa en un contexto industrial cada vez más orientado hacia la digitalización.

Actualmente, las ideas de Lean Manufacturing y de la I4.0 se han integrado en un nuevo paradigma organizacional llamado “*Lean 4.0*” [Langlotz et al., 2021]. Los métodos de un paradigma pueden complementarse con tecnología del otro, y viceversa. Si bien estos paradigmas, Lean Manufacturing e I4.0, pueden parecer bastante diferentes, Lean 4.0 ofrece oportunidades prometedoras para el futuro de las organizaciones si éstas pueden combinarlos sinérgicamente para mejorar su eficiencia [Marinelli et al., 2021].

El concepto de Lean 4.0 ha surgido con fuerza en varias investigaciones últimamente, combinando la filosofía Lean con el nuevo paradigma de I4.0 y tratándolos como elementos complementarios. Entre ellos, se encuentran trabajos donde se revisan o plantean conceptos respecto a esta combinación de factores, o se especifican posibles metodologías para su aplicación [Dillinger et al., 2021, Dillinger et al., 2022, Kolla et al., 2019]. Algunos investigadores sugieren que una combinación de Lean, simulación y optimización puede ser crucial en el futuro para mejorar la eficiencia de las organizaciones. Esta combinación puede perfeccionar el proceso tradicional de TDD, acelerar las mejoras y reconfiguraciones del sistema y apoyar el aprendizaje organizacional [Uriarte et al., 2018]. Sin embargo, existen pocos artículos de aplicación concretos en varios estudios de caso.

En el trabajo de [Cifone et al., 2021] se proponen dos formas actuales para esta integración entre Lean y el paradigma 4.0. La primera perspectiva sugiere Lean como base para la implementación de la I4.0, argumentando que los procesos controlados y optimizados pueden ser un requisito previo para cualquier proceso de digitalización. En contraste, la segunda perspectiva muestra a la I4.0 como un complemento necesario a Lean, basándose en la gran personalización de la demanda que implica la situación actual, donde la I4.0 representa un medio que Lean puede explotar para adaptarse a las nuevas tendencias del mundo. Sin embargo, un común denominador que se encuentra en ambas perspectivas es el caso de la manipulación o manejo de materiales. Este concepto, se relaciona con el movimiento y almacenamiento dentro y alrededor de un establecimiento

industrial. El manejo adecuado de materiales ofrece oportunidades para mejorar la productividad, reducir desechos, minimizar los accidentes industriales y reducir la mano de obra, entre otros objetivos [Purushothama, 2013]. En consecuencia, desde el punto de vista de la manufactura esbelta, el transporte puede definirse como una operación necesaria pero que no agrega valor a un proceso de fabricación. Sin embargo, el manejo de materiales es vital en una instalación de fabricación, pero con frecuencia añade tiempo y gastos innecesarios [Adeodu et al., 2023, Yamazaki et al., 2017]. El transporte ineficiente de materias primas o productos en proceso influye directamente en la productividad y rentabilidad empresarial. También afecta otros desperdicios Lean, como las esperas en la fabricación y los inventarios terminados y en proceso.

En el caso de las PyMEs, la gestión de materiales es esencial, dado que suelen tener una producción intermitente, por lotes y/o personalizada. En otras palabras, se producen una variedad de productos diferentes, que generalmente requieren de diversas tareas y procesamientos por lo cual deben seguir secuencias de proceso diferentes complejizando la planificación de la producción. Por otra parte, para cumplir con los requerimientos productivos, muchas veces las configuraciones de los pisos de planta deben modificarse, por lo que las estrategias para el movimiento de materiales tienen que ser replanificadas.

Lo dicho anteriormente, está relacionado con dos problemas fundamentales que han sido tratados durante años en la optimización de operaciones: el problema del viajante de comercio (*Travelling Salesman Problem (TSP)*) y el problema de enrutamiento de vehículos capacitados (*Capacitated Vehicle Routing Problem (CVRP)*). Estos problemas se consideran problemas combinatorios *NP-hard*. En el TSP, dada una lista de nodos y la distancia entre ellos, se debe encontrar el camino más corto posible para visitar cada nodo y regresar al nodo original [Wang and Tang, 2021]. Formalmente, una instancia de un TSP se puede definir en un gráfico como un conjunto de nodos y bordes. Cada nodo tiene sus características y una posible solución al problema pasa por definir un camino, es decir, una secuencia específica de nodos que satisfaga las restricciones del problema. Por otro lado, en el caso específico de un CVRP, existe otro nodo llamado almacén que se adiciona al problema. Los nodos originales representan clientes, cada uno con una demanda particular a satisfacer. Cada ruta posible inicia desde el depósito y el transporte debe visitar secuencialmente un subconjunto de clientes [Wu et al., 2019]. El objetivo es minimizar la distancia del recorrido. Esta situación se corresponde con el caso del problema de manejo de materiales en piso de planta, donde dado un punto inicial se visitan diferentes puestos de trabajo para abastecerlos.

En sistemas de producción flexibles, el piso de producción podría modificarse. En consecuencia, utilizar métodos de optimización basados en grafos para abordar problemas de minimización de recorridos podría resultar engorroso, ya que deben refactorizarse cada vez que hay un cambio en la configuración del sistema productivo. En cambio, el uso de métodos basados en simulación de eventos discretos o, aún más, estrategias basadas en GDs proporciona una descripción más realista del sistema [Cheng et al., 2020].

La intersección de tecnologías avanzadas y aplicaciones industriales implica el desarrollo de tecnología de GD y estrategias de colaboración humano-máquina para crear entornos operativos más eficientes. En este sentido, para la tarea de manejo de materiales, la integración de robots móviles autónomos y robots manipuladores

[Ghodsian et al., 2022] en el piso de planta podría traer nuevas formas de gestionar el material a lo largo de los procesos de transformación y su manipulación en los almacenes. Por ejemplo, en el trabajo reciente de [Stączek et al., 2021], se utilizó un GD para evaluar la exactitud de los supuestos de diseño adoptados para la fase inicial de implementación de un vehículo móvil autónomo en la sala de producción de una empresa. Sin embargo, el trabajo se centra principalmente en desarrollar eficientemente el GD. Un paso adelante se analiza en [Kiyokawa et al., 2023], donde una revisión bibliográfica muestra los enfoques predominantes centrándose en comprender las definiciones actuales de dificultad y complejidad y los problemas pendientes del sistema de ensamblaje de colaboración entre humanos y robots. De manera similar, esta nueva tendencia de integración humano-máquina se aborda en [Lu et al., 2023], donde se presenta un marco basado en GDs para coordinar equipos de humanos y robots, enfocándose en la tecnología de GD como un nodo de comunicación central para permitir la colaboración entre estas entidades. Todos estos avances deberían permitirnos tener información más precisa y en tiempo real sobre el entorno operativo, lo que hace que el manejo de materiales, ya sea realizado únicamente por robots o en colaboración con humanos, requiera algoritmos de optimización eficientes para resolver este tipo de problemas.

Muchas propuestas se han realizado para optimizar el manejo de materiales modelándolo como un TSP o CVRP. En la literatura se pueden encontrar una vasta cantidad de iniciativas para resolver estos problemas, basadas en algoritmos evolutivos, diversos métodos de optimización matemáticos [Dündar et al., 2022, Singh et al., 2022] hasta métodos de IA. En [Akhand et al., 2015] utilizan métodos de inteligencia de enjambre para la optimización de rutas en problemas CVRP, mientras que en [Skinderowicz, 2022] usan la optimización de colonias de hormigas para resolver grandes instancias del TSP. En [Nazari et al., 2018] y en [Otoni et al., 2022] aplican el aprendizaje por refuerzo como técnica de optimización y en el caso de [Hu et al., 2020] usan redes neuronales artificiales para la resolución.

Dentro del campo de la optimización, hace relativamente poco tiempo se propuso un algoritmo metaheurístico simple llamado Algoritmo de Seno Coseno (Sine Cosine Algorithm (SCA)) [Mirjalili, 2016], que ha mostrado excelentes resultados en varios tipos de problemas. Actualmente, se utiliza en una gran cantidad de aplicaciones y se puede encontrar en múltiples investigaciones en varios campos de aplicación. Por ejemplo, en [Karmouni et al., 2022], se utiliza para controlar y mejorar el rendimiento del monitoreo en un sistema fotovoltaico; en [Issa, 2021], contribuye a medir pares de secuencias biológicas en el contexto del COVID-19, en [Kuo and Wang, 2022] se usa en un problema de clasificación con datos mixtos, en [Lyu et al., 2021] se usa para la predicción de la resistencia a la compresión axial de columnas circulares y en [Daoui et al., 2021] se utiliza para la protección de derechos de autor de imágenes.

Sin embargo, a pesar del potencial del SCA, al menos durante la revisión de la literatura, no se han encontrado aplicaciones relevantes en los sistemas de TDD sobre el manejo de materiales en los procesos de fabricación.

El artículo original muestra que el SCA puede ser muy eficaz para resolver problemas reales con espacios de búsqueda restringidos y desconocidos y que converge, evitando un óptimo local. Además, tiene un tiempo de ejecución razonable y una implementación sencilla [Benmessaoud et al., 2021]. Si bien esto es cierto, varios autores desarrollaron mejoras o variantes de este algoritmo que aumentaron su robustez o velocidad de convergencia agregando ciertas

modificaciones metaheurísticas o hibridaciones, como en [Abdel-Baset et al., 2019], [Li et al., 2018a] y [Al-qaness et al., 2018].

Sin embargo, se han encontrado pocos artículos donde se traten problemas con soluciones de naturaleza discreta [Yang et al., 2020], y no se han desarrollado trabajos donde se pueda aplicar SCA a sistemas de fabricación o problemas de manejo de materiales. Sólo se ha reportado en la literatura el artículo de [Gupta et al., 2022], donde proponen e implementan el Algoritmo Discreto Seno Coseno (Discrete Sine Cosine Algorithm (DSCA)) y lo aplican a la programación de semáforos urbanos.

Por otro lado, para resolver problemas de enrutamiento muchos trabajos utilizan la teoría de grafos para su resolución [Lei et al., 2022, Duhamel et al., 2011]. Sin embargo, como ya se ha mencionado, en las representaciones con modelos de grafos deben realizarse muchas simplificaciones para modelar al sistema o, en caso contrario, utilizar una gran cantidad de variables para la formulación del problema [Gouveia et al., 2019]. En este contexto, por lo tanto, al utilizar simulación de eventos discretos junto con los modelos digitales de los agentes que intervienen en el sistema de fabricación, no sólo podemos representar el sistema, sino que también es posible aprovechar el potencial de estos simuladores muy avanzados para analizar y comprender la dinámica general del sistema en su conjunto. Además, esto nos permite incluir eventos de fallas, estadísticas de producción, niveles de stock, entre otros, los que nos da en definitiva, una representación más realista del proceso de producción y consecuentemente, la solución al problema de manejo de materiales podrá ir adaptándose en tiempo real a la situación o estado del sistema físico.

## 2.3 Simbiosis entre la Simulación e IA para la TDD

Desde hace años, la simulación de los procesos productivos es una herramienta que ha ganado un lugar preponderante para la toma de distintos tipos de decisiones, que van desde el propio diseño del sistema, rediseño de alguna parte, evaluación de las estrategias de secuenciación de tareas, entre muchísimas otras [Mourtzis et al., 2014]. En muchos casos, se ha utilizado en la formulación de problemas de TDD en sistemas industriales, como problemas de optimización [De Paula and Martínez, 2012], incluidos problemas multi-objetivo [Sofiane and Djamila, 2019]. Por ejemplo, en [Amiri et al., 2019], se abordó un problema de asignación de recursos en un sistema productivo para gestionar tanto la incertidumbre como la flexibilidad en la secuencia de trabajo. Para resolverlo, se formuló un problema de optimización multiobjetivo y se desarrolló un modelo de simulación de eventos discretos que representaba el proceso de producción, permitiendo así encontrar una solución eficiente. Por otro lado, la simulación por computadora también se ha utilizado para identificar parámetros óptimos como el tamaño del lote, el stock, la secuencia de procesamiento de las órdenes de trabajo o la frecuencia de entrega [Block et al., 2018]. Se propusieron enfoques de simulación que incorporan cambios externos mediante la actualización de los parámetros del sistema para proporcionar características adaptativas para la TDD [Barrera-Diaz et al., 2018, Ben Moussa et al., 2019].

Con respecto a los métodos y técnicas de simulación, actualmente se utilizan para el diseño de sistemas de fabricación modernos, como sistemas de fabricación flexibles [Jeong and Jung, 2012] y sistemas de fabricación reconfigurables [Gola, 2018]. Por ejemplo, en el artículo de [Molenda et al., 2017] se presenta una propuesta basada en simulación para el desarrollo de una metodología de evaluación con el fin

de realizar una estimación económica de sistemas de fabricación flexibles; en [Khedri Liraviasl et al., 2015] el objetivo es investigar y respaldar la reconfigurabilidad de un sistema de fabricación mediante la aplicación de una técnica de modelado híbrido de simulación de eventos discretos y basada en agentes; en [Li et al., 2018b] se propone un sistema de fabricación reconfigurable autónomo conectado con una gestión descentralizada de líneas autónomas individuales que pueden reconfigurarse para diversas tareas de fabricación a través de módulos de fabricación y en [Qudeiri, 2017] utilizan la simulación para un sistema de producción para encontrar el enrutamiento casi óptimo para sistemas de fabricación flexibles. Sin embargo, si bien la simulación ha sido muy empleada y aún se sigue empleando, tiene algunas limitaciones para la formulación de sistemas autónomos de TDD en línea, debido principalmente a su característica estática. En otras palabras, los métodos tradicionales de simulación tienen escasa posibilidad de reconfiguración del modelo frente a cambios constantes en la configuración del sistema real, dado que es necesario la intervención de un experto para volver a formularlo. Por ello, y considerando la fuerte irrupción del internet industrial de las cosas [González García et al., 2018] y los sistemas ciberfísicos [Ding et al., 2020], su lugar ha sido ocupado por el enfoque del GD [Haag and Anderl, 2018].

El desarrollo confiable y seguro del internet industrial de las cosas ha potenciado la interoperabilidad de los sistemas de fabricación [Dejene et al., 2020, Mourad et al., 2020] al mismo tiempo que ha facilitado la interacción entre los diferentes componentes y subsistemas que integran cada sistema productivo [Laurindo et al., 2019]. En este sentido, las posibilidades tecnológicas actuales han fortalecido el concepto de GD y se han comenzado a utilizar en investigaciones en modelos de producción, mantenimiento predictivo y en los servicios post-venta, aunque aún se plantean muchos desafíos en la construcción de un GD fiable en la práctica [Lu et al., 2020, Melesse et al., 2020]. Este concepto puede verse como un elemento central de la fabricación futura, las aplicaciones impulsadas por GDs desafiarán y cambiarán los fundamentos de los sistemas y operaciones de fabricación.

Igualmente, en el desarrollo de los GDs, los métodos y técnicas de simulación de sistemas juegan un papel preponderante en la abstracción y representación de los sistemas de producción [Cheng et al., 2020]. En este sentido, los GDs utilizan la simulación en un esquema de intercambio de información con una variedad de datos de diferentes equipos y sistemas físicos para actualizar continuamente los modelos digitales de las entidades que conforman el sistema productivo, con el fin de reflejar cualquier cambio que ocurra en los equivalentes físicos de dichas entidades (o agentes) a lo largo del tiempo [Meski et al., 2019]. De esta manera se puede lograr un esquema de retroalimentación en un entorno virtual adaptativo que facilite la TDD en base a estos GDs. Entonces, bajo circunstancias y condiciones de incertidumbre, los sistemas de producción podrían beneficiarse significativamente del enfoque de control, seguimiento y supervisión basado en GDs para obtener información completa sobre los elementos y componentes necesarios para la TDD [Zhang et al., 2020].

No obstante, para todas las propuestas que impliquen el uso de un GD, es de suma importancia el desarrollo de arquitecturas de integración que permitan la interacción entre los subsistemas físicos involucrados y sus homólogos digitales para dar lugar a la creación de dichos gemelos. En este sentido, se han realizado pocas propuestas para aquellos sistemas de producción en los que interviene algún equipo robótico.

Por ejemplo, en el trabajo reciente de [Illmer and Vielhaber, 2019], se presenta un método de sincronización basado en simulación para la validación virtual de sistemas de producción en la fase de planificación. Se utiliza una aplicación informática en la que el GD del proceso de fabricación virtual se divide en un GD del proceso relacionado con el producto y otro relacionado con los recursos. La metodología se verifica mediante un simulador de una estación automatizada de montaje de chasis de automóviles para el relevo físico de los trabajadores en la zona de montaje final. En [Qian et al., 2020] se propone un mecanismo de colaboración inteligente para determinar la configuración óptima de los recursos en un sistema ensamblador asumiendo que la intercomunicación entre los diferentes recursos se establece a través del Internet Industrial de los datos recolectados por lectores RFID transmitidos por chips NFC (comunicación de campo cercano). De manera similar, en [Leng et al., 2020] los autores proponen un método para la reconfiguración automática del sistema de fabricación basado en el uso de un GD, integrando simulación y optimización iterativa. Se introduce una nueva clase de máquinas herramienta de arquitectura abierta (OAMT) con módulos intercambiables, que permiten adaptar el sistema de producción a cambios en la planificación, demostrando mejoras en el rendimiento con mínima sobrecarga operativa. En el trabajo reciente de [Xia et al., 2021] se propone un enfoque basado en GDs y aprendizaje por refuerzo profundo para automatizar y optimizar el control inteligente en sistemas de manufactura, para abordar el problema de programación de la secuencia de operación. Los autores de [Rolle et al., 2020] presentaron recientemente una arquitectura modular para implementar un GD para un caso de línea de producción controlada por un PLC para la industria de procesos, mientras que en [Wang et al., 2020] se propone una arquitectura colaborativa para una plataforma de Internet industrial, llamada sistema operativo industrial (SO-industrial), que aloja el controlador industrial, el hilo digital y los microservicios para lograr un entorno de sistema de información empresarial cooperativo para los sistemas de fabricación. No obstante, estas propuestas tienen limitaciones cuando en el proceso de producción intervienen sistemas robóticos heterogéneos.

Aunque existe una tecnología de simulación estable y suficientemente desarrollada, que todavía se está perfeccionando, la integración digital de las partes físicas y de software para el desarrollo de un GD de sistemas de producción es un tema que recientemente está recibiendo especial atención [Wang et al., 2020, Xia et al., 2021]. Además, es de vital relevancia el análisis de la información del sistema en tiempo real sobre los modelos de los sistemas de fabricación autónomos y flexibles cuando se recopila información heterogénea de múltiples fuentes [Fan et al., 2021]. Algunas empresas como ABB han comenzado a implementar simuladores propietarios para sus robots, permitiendo la creación de GDs [Gonzalez, 2020]. Asimismo, desarrollos propietarios como Actin o TransBot, permiten el control de un sistema robótico basado en GD [Dockter, 2021, Flaherty, 2021]. Sin embargo, para los casos de sistemas de producción en los que intervienen varios sistemas robóticos de diferente naturaleza, las propuestas de arquitecturas de integración de acceso abierto con sistemas de decisión autónomos son prácticamente nulas, lo que justamente es una de las motivaciones de este trabajo de tesis.



# Capítulo 3

## Antecedentes metodológicos

El presente capítulo tiene como propósito presentar brevemente las principales metodologías y técnicas sobre las que se sustentan los desarrollos de esta tesis. A lo largo de este apartado, se examinan las principales teorías, enfoques y métodos que han servido de base para la investigación, proporcionando un marco de referencia integral. Se abordan tanto los fundamentos conceptuales como las herramientas metodológicas esenciales para garantizar un análisis riguroso y coherente. Además, se establece una base que facilita el análisis de los capítulos siguientes, permitiendo comprender la pertinencia de cada enfoque y su contribución al del estudio. De este modo, se busca no solo justificar la selección de las metodologías empleadas, sino también destacar su impacto en la validez y fiabilidad de los resultados obtenidos.

### 3.1 Sistemas flexibles de producción

La principal característica de los sistemas de manufactura flexible, como su nombre lo indica, es contar con capacidad suficiente para poder reaccionar y adaptarse a diversas situaciones. Como se ha señalado en el Capítulo 2, las organizaciones requieren cada vez mayor adaptabilidad para hacer frente al contexto actual. Estos sistemas comprenden no solo los procesos de manufactura y ensamblaje, sino también el flujo de materiales, el transporte y la comunicación necesarios para reconfigurar la producción acorde a los cambios en los patrones de demanda que, en definitiva, influyen directamente sobre el sistema productivo.

Generalmente, los objetivos empresariales comprenden aumentar la productividad, disminuir costos y satisfacer a los clientes, aunque actualmente empiezan a tenerse en cuenta otros objetivos, como la sustentabilidad, el impacto social y el entorno laboral, entre otros. En relación a esto, los métodos de producción flexible están surgiendo como una alternativa que tiene la capacidad y autonomía necesarias para modificar tanto las instalaciones físicas como su planificación de manera dinámica. Esta característica les confiere capacidad para adaptarse a los cambios impuestos por las condiciones de demanda, buscando mantener o incrementar la productividad mediante la reducción de todo tipo de desperdicios, a la vez que permite asignar tareas de mayor responsabilidad a los recursos humanos de una organización. Asimismo, un buen grado de flexibilidad en los procesos de manufactura permite una mayor personalización de los productos, lo que representa una ventaja competitiva en mercados donde la demanda varía constantemente.

Este tipo de sistema de producción se caracteriza por incorporar un considerable grado de automatización y tecnología de la información, con el objetivo de conseguir sistemas fabriles ágiles y eficientes [Jacobs and Chase, 2018]. De esta manera, los modelos de sistemas de producción flexibles son apropiados para facilitar la fabricación de cantidades medianas de una diversidad de productos [Weckenborg et al., 2024]. La versatilidad es una característica especialmente atractiva para las PyMEs manufactureras, que, generalmente, poseen un tipo de flujo de proceso intermitente, trabajando a demanda de sus clientes y, en muchos casos, de forma personalizada, con una alta variedad y bajo volumen de producción de un determinado producto. Entonces, el hecho de poder contar con sistemas productivos con capacidad de adaptación les permite a las empresas enfrentar variaciones en la demanda (ya sea por fluctuaciones o cambios de tipo de productos) manteniendo su competitividad sin incurrir en costos excesivos o generar desperdicios innecesarios.

Entre algunas de las características que se pueden mencionar de este tipo de sistemas flexibles se encuentran la producción descentralizada, trabajadores con alto grado de capacitación y posibilidad de realizar varias funciones dentro del proceso productivo, equipos robotizados, trabajo en equipo, respuesta inmediata frente a averías, mejor utilización del capital, mejor calidad del producto, menores inventarios, entre otras [Heizer and Render, 2004]. Todas estas cualidades no solo contribuyen a optimizar la eficiencia del sistema, sino que también fomentan un entorno de trabajo más dinámico y colaborativo, donde los empleados pueden aportar soluciones innovadoras para la mejora continua de los procesos.

En este nuevo enfoque, los sistemas de fabricación combinan seres humanos, maquinarias y estaciones de trabajo conectadas entre sí por un flujo común de material e información, lo que los vuelve sistemas más complejos que los tradicionales. En consecuencia, tanto desde la etapa de diseño del sistema productivo como en la etapa de operación y reconfiguración, es necesario tomar una elevada cantidad de decisiones, que integran aspectos como determinar la capacidad de la planta, los recursos necesarios, el diseño en sí del proceso productivo, el sistema de manejo de materiales, la planificación de la producción y el control del sistema en su totalidad [Weckenborg et al., 2024]. La incorporación de las tecnologías ciberfísicas junto a la IA en el ámbito industrial permite pensar en el desarrollo de sistemas de toma de decisiones cada vez más sofisticados para mejorar la eficiencia y reducir tiempos de respuesta ante cambios frecuentes y/o imprevistos.

En la Figura 3.1 se destacan los principales elementos de un sistema de producción flexible desde una perspectiva sinérgica. No solo las personas y los equipos deben ser adaptables a diversos escenarios, sino que el proceso de TDD debe ser rápido y eficaz. En este sentido, las tecnologías de la I4.0 son el puente para conseguir sistemas de producción realmente adaptables, ya que es posible disponer y procesar una gran cantidad de datos en tiempo real en línea mediante diferentes herramientas y técnicas de análisis de datos e inteligencia artificial para la toma de decisiones. En consecuencia, tener capacidad de integrar datos en tiempo real con modelos predictivos y algoritmos de optimización puede contribuir a mejorar la eficiencia operativa y proporcionar a las organizaciones una ventaja estratégica en un mercado en constante evolución.

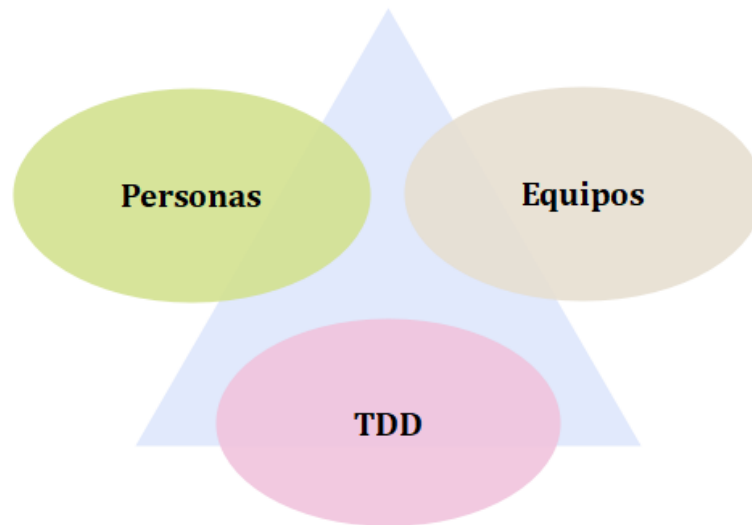


Figura 3.1: Relaciones en un sistema de manufactura flexible.

## 3.2 Modelado y simulación de sistemas productivos

La bibliografía disponible sobre Modelado y Simulación de Sistemas es vasta y abarca diversos aspectos para diferentes tipos de sistemas [Zeigler et al., 2018, Zhou and Huang, 2024]. Se abordan desde el modelado y simulación de sistemas físicos y biológicos hasta sistemas socioeconómicos e industriales. Particularmente, en el caso de este trabajo de tesis, nos centraremos en presentar las herramientas metodológicas adecuadas para el modelado y simulación de sistemas productivos.

En términos generales, durante el desarrollo de un proyecto de simulación podemos distinguir tres grandes fases. La primera fase es la concepción, en la cual se desarrolla y valida un modelo conceptual y un modelo de datos del sistema en estudio. A continuación, sigue la fase de implementación que implica la construcción, implementación y validación del modelo computacional propiamente dicho. Finalmente, la tercera fase es una etapa de análisis en la que se utiliza el modelo de simulación computacional para el diseño y desarrollo de experimentos. Seguidamente, los resultados obtenidos de la experimentación deben ser analizados para obtener las conclusiones y recomendaciones del estudio [Montevecchi et al., 2010].

El modelado de un sistema productivo es fundamental para poder comprender la estructura y relaciones entre las entidades que lo componen. Un modelo es una abstracción del sistema en estudio que, comúnmente, asume una serie de supuestos e hipótesis simplificadoras respecto al sistema que se está modelando. Un modelo puede ser cualquier representación física, matemática o lógica de un sistema, entidad, fenómeno o proceso. La finalidad del modelado es capturar las características esenciales del sistema y proporcionar un marco analítico para su estudio. La idea más usual de un modelo de simulación radica en su definición como un conjunto de instrucciones, normas, ecuaciones o restricciones destinadas a reproducir el comportamiento del sistema en términos de entrada y salida [Zeigler et al., 2018]. De esta manera, mediante la simulación computacional del modelo de un determinado

sistema, es posible evaluar el impacto de diferentes variables y configuraciones en su desempeño. De esta forma es posible realizar una experimentación exhaustiva sin necesidad de intervenir directamente en la operación del sistema real. En consecuencia, la simulación es una herramienta que resulta especialmente valiosa para la gestión de los sistemas productivos, ya que permite ensayar las consecuencias de diferentes acciones sobre el sistema.

En este sentido, la simulación se ha transformado en una herramienta de análisis indispensable para la toma de decisiones ya que permite evaluar el impacto de los cambios en el sistema sin que sea necesario modificarlo en la realidad. En otras palabras, trabajar con un modelo simulado permite liberarse de limitaciones y restricciones, evitando o disminuyendo los costos de las pruebas. Si algo no es efectivo en el ámbito simulado, las empresas tienen la posibilidad de reiniciar la simulación y probar con diferentes parámetros, lo cual permite perfeccionar los productos y procesos.

El modelado y la simulación computacional han evolucionado significativamente con el desarrollo de nuevas tecnologías y herramientas analíticas. Actualmente, existen distintas metodologías para la representación de los diferentes tipos de sistemas. Cada metodología ofrece ventajas específicas dependiendo del tipo de sistema a simular y los objetivos del estudio. La elección del método de modelado adecuado es fundamental para garantizar la validez y aplicabilidad de los resultados obtenidos, proporcionando información clave para la toma de decisiones en entornos productivos dinámicos y complejos.

Particularmente, los sistemas de producción manufacturera son un caso especial de sistemas dinámicos híbridos, conocidos como sistemas de eventos discretos [Rolle et al., 2020]. En este sentido, pueden ser modelados y simulados bajo el paradigma de simulación de eventos discretos [Furian et al., 2015]. La simulación de eventos discretos es una metodología ampliamente difundida en el ambiente empresarial y ha comprobado su utilidad como soporte para apoyar la toma de decisiones. Principalmente, en el pasado se ha usado profusamente para la asistencia con la toma de decisiones relacionadas con la planificación de la producción e inventarios, así como con el diseño de los sistemas de producción y sus cadenas de suministro [Mourtzis et al., 2014]. Sin embargo, en el contexto actual, los tomadores de decisiones de las empresas productivas se enfrentan cada vez más a situaciones complejas donde deben maximizar objetivos productivos, económicos y sustentables, que generalmente entran en conflicto entre sí.

Estos modelos de simulación de eventos discretos son adecuados para representar procesos y situaciones complejas de la realidad en función de los eventos que puedan ocurrir, en diferentes momentos de tiempo, por lo que pueden representar tanto situaciones típicas de un proceso productivo (rotura de inventarios, fallas de equipos, aparición de imponderables), como así también estimar un determinado nivel de utilización de recursos relacionados con dichas situaciones. Asimismo, se incorporan las probabilidades asociadas a la ocurrencia de cada evento, en función de las características específicas de los elementos que intervienen en el sistema. En conjunto, estos elementos y sus interacciones permiten simular la evolución dinámica del proceso bajo análisis. De esta manera, los modelos posibilitan la conceptualización y gestión de un proceso productivo en términos de los eventos discretos que pueden presentarse y cuyo impacto repercute tanto en los clientes como en los demás componentes del sistema [Zeigler et al., 2018]. En definitiva,

la simulación de eventos discretos es una metodología que se puede utilizar para estudiar los sistemas y procesos cuyos estados van cambiando con el tiempo de forma discreta.

Existen distintas técnicas para modelar los sistemas de eventos discretos. En este trabajo de tesis, para el desarrollo de los modelos de los casos de estudio abordados en los próximos capítulos, se utiliza la técnica IDEF-SIM (*Integrated Definition Methods – Simulation*) desarrollada por [Montevechi et al., 2010]. Esta metodología IDEF modificada emplea componentes lógicos que se encuentran en técnicas como IDEF0 e IDEF3 [Waissi et al., 2015], pero de forma parecida a la lógica de interpretación de procesos que se emplea comúnmente en proyectos de simulación de eventos discretos. Así, mediante la técnica IDEF-SIM se puede apreciar un incremento en la funcionalidad del modelo conceptual, lo que podría simplificar la programación, comprobación y validación del modelo de simulación y la generación de escenarios. En consecuencia, los componentes lógicos empleados en la metodología IDEF-SIM motivan al encargado del modelo teórico a enfocarse en la información que, en un futuro, resultará crucial para el modelo computacional. Los elementos y símbolos utilizados en esta técnica comprenden la representación de entidades, funciones, recursos, controles, reglas, conexiones, entre otros, como puede verse en la Figura 3.2 [Montevechi et al., 2010], lo que brinda información valiosa para el análisis posterior.






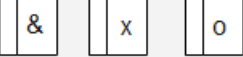

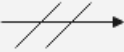


Elementos	Simbología
Entidades	
Funciones	
Flujo de entidades	
Recursos	
Controles	
Reglas	
Movimiento	
Flujo de entrada	
Fin del sistema	
Conexión con otra figura	

Figura 3.2: Simbología de la técnica IDEF-SIM.

En definitiva, un buen modelado conceptual es fundamental para comprender el sistema correctamente y llegar a representarlo de forma precisa en un simulador computacional, con el fin de obtener información certera para la posterior TDD.

### 3.3 Gemelos digitales

La irrupción de los sistemas ciberfísicos ha cambiado paradigmáticamente la forma en que se conciben los sistemas industriales, permitiendo la integración efectiva de todas las entidades que intervienen en un sistema manufacturero [Laurindo et al., 2019]. Estos sistemas combinan el mundo físico y el digital mediante el uso de sensores, dispositivos interconectados y plataformas de análisis avanzadas, facilitando una interconexión e intercambio de información en tiempo real entre máquinas, productos y operadores. De esta manera, es posible lograr una interacción continua entre los diferentes componentes de un sistema productivo, lo que permite gestionar las entidades físicas de manera confiable, segura, colaborativa, robusta y eficiente [Cassandras and Lafortune, 2008]. Este nivel de integración se lo conoce como Gemelo Digital y no solo contribuye a optimizar los procesos de producción, sino que también permite la implementación de estrategias de mantenimiento predictivo, monitoreo remoto y control de calidad automatizado.

Desde esta perspectiva holística, los sistemas de fabricación integrados pueden conceptualizarse como estructuras compuestas por dos partes principales: una parte física y una parte digital. Como se esquematiza en la Figura 3.3, la parte física comprende todas las entidades y recursos empleados en los procesos de fabricación, incluyendo maquinaria, robots, operadores y sistemas logísticos. Por otro lado, la parte digital o virtual incorpora aplicaciones y servicios ubicuos que permiten la gestión avanzada de datos y la optimización de procesos. Esta capa digital incluye subsistemas con capacidades para el procesamiento y análisis de datos en tiempo real, modelado y simulación de procesos productivos, así como sistemas inteligentes de toma de decisiones basados en inteligencia artificial y aprendizaje automático.

La convergencia entre estos dos componentes ha dado lugar a un nuevo paradigma en la manufactura, donde los sistemas ciberfísicos posibilitan la creación de entornos de producción más ágiles, autónomos y adaptativos. Esto no solo mejora la eficiencia operativa, sino que también permite a las organizaciones responder con mayor rapidez a los cambios en la demanda, personalización de la producción y reducción de costos sin comprometer la calidad del producto. En este contexto, el desarrollo de arquitecturas flexibles y escalables para la integración de los sistemas ciberfísicos resulta fundamental para el desarrollo de la I4.0 y la consolidación de fábricas inteligentes.

Regresando al detalle de la Figura 3.3, en el sistema físico, representado esquemáticamente en la parte inferior, se encuentran interconectadas las diferentes entidades involucradas en el proceso de producción. En una clasificación general, pueden ser agrupadas como Equipos, Materiales, Operarios y Herramientas. Todas estas entidades pueden compartir información y mantener actualizado su estado en tiempo real gracias a las diferentes tecnologías actuales como, por ejemplo, los protocolos Ethernet, ROS, sistemas de geo-referenciación, etiquetas/lectores RFID, etc. De esta forma, en la parte física no sólo se recopilan e intercambian datos, sino que también se genera comunicación con la parte digital donde se producen decisiones o acciones que posteriormente se ejecutan en la parte física. Es decir,

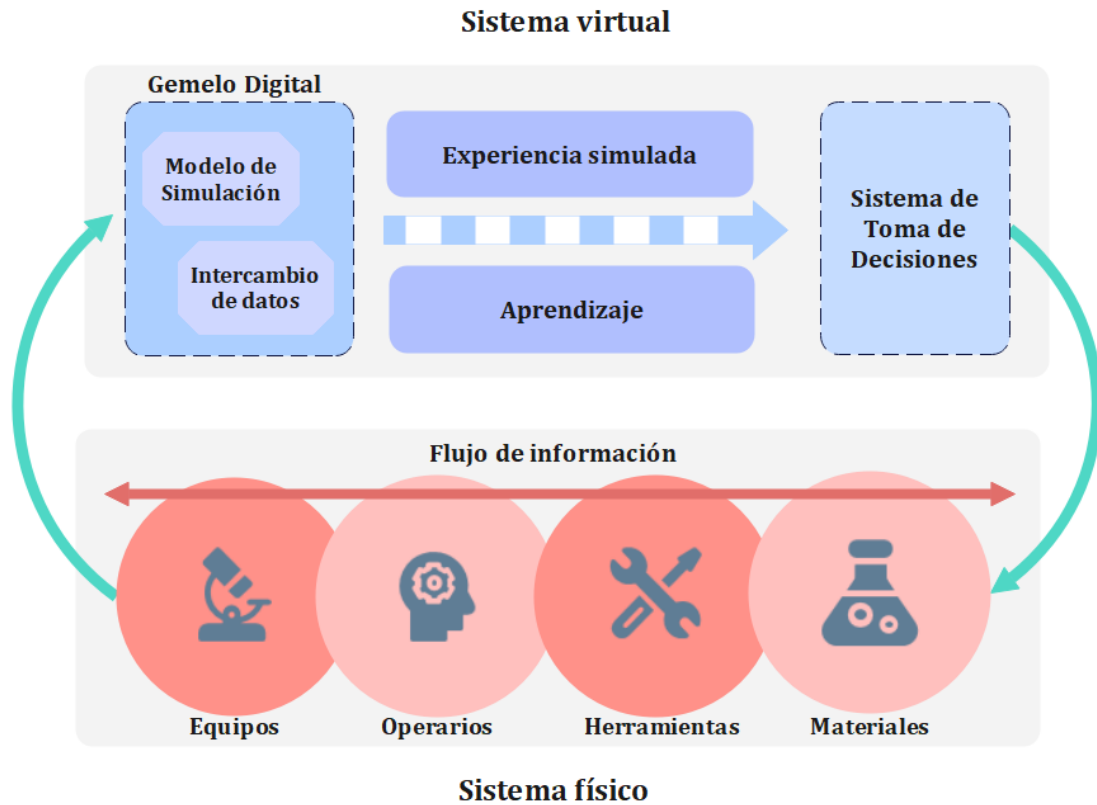


Figura 3.3: Jerarquía funcional de entidades en un sistema de fabricación.

mediante esta comunicación, la parte digital puede afectar los procesos físicos y viceversa; lo que es una característica fundamental de los gemelos digitales.

La parte digital, representada en la parte superior de la Figura 3.3, contiene distintos subsistemas, servicios y aplicaciones que desarrollan distintas funciones que permiten a las entidades intervinientes en la fabricación operar coordinadamente para optimizar el desempeño del sistema productivo en su conjunto. Actualmente, muchos campos de investigación enfocan sus esfuerzos hacia el desarrollo de propuestas para este sistema virtual. Particularmente, todos ellos buscan aprovechar las bondades y beneficios de diferentes metodologías para generar propuestas que sustenten el desarrollo de sistemas autónomos de TDD, los cuales deben establecer las acciones que se deben ejecutar para gestionar el sistema físico de producción [Kunath and Winkler, 2018]. En este sentido, el concepto de GD ocupa un lugar clave y está centrando la atención de muchas investigaciones para el desarrollo de metodologías y técnicas que permitan la creación de manera fiable de tales gemelos [Sun et al., 2020].

El concepto de GD se originó en el año 2003 en la NASA, durante una conferencia sobre gestión del ciclo de vida del producto. Su descripción inicial lo define como una representación virtual de un producto físico que integra información detallada sobre éste, en el contexto de la gestión del ciclo de vida [Jones et al., 2020]. La función principal de los GDs es proporcionar una descripción física, funcional e integral del sistema en su conjunto. Los GDs son modelos virtuales para reproducir de manera realista las propiedades, comportamientos y reglas de las entidades de la parte física [He et al., 2019]. En otras palabras, los modelos virtuales y las entidades

físicas tienen una apariencia y comportamientos similares, es decir, son gemelos.

Para el desarrollo de los GDs, las técnicas de simulación juegan un papel central ya que deben operar en un esquema de intercambio de información con una variedad de datos, provenientes de los diferentes equipos y sistemas físicos de planta, para mantener actualizados continuamente los modelos digitales de cada una de las entidades con el fin de reflejar cualquier cambio que ocurra en sus equivalentes físicos en el tiempo. La interacción entre el mundo físico y los GDs tiene una importancia clave debido a que los modelos virtuales y el mundo físico deben evolucionar conjuntamente, ya que este último posee un comportamiento dinámico y, por ejemplo, una misma entidad puede mostrar diversas propiedades en diferentes momentos. Por esto, los datos recogidos en tiempo real del mundo físico se deben analizar y compartir con el mundo virtual en un esquema sincronizado para mantener actualizados los modelos que permiten simular el proceso físico y su evolución.

Además, el concepto de GD permite integrarse en todo el ciclo de vida del producto a desarrollar y del proceso productivo, desde las fases de diseño y desarrollo hasta la operación y mantenimiento de los mismos. De esta manera, se pueden tomar decisiones críticas con mayor certidumbre, optimizando el rendimiento general de la organización. El gemelo digital para productos facilita el diseño, simulación y comprobación virtual de productos, que abarcan desde la mecánica hasta la física, la electrónica y el software. El gemelo digital para producción asiste en la planificación, simulación, proyección y optimización de todos los procesos productivos, mejorando maquinaria, líneas e incluso fábricas y plantas completas durante todo su ciclo de vida [Siemens, 2024].

Un aspecto fundamental de los sistemas de fabricación integrados, como ya se mencionó, es la comunicación entre el sistema físico y el sistema virtual. En este sentido, existen diversos protocolos de comunicación usados en el ámbito industrial y en el Internet de las Cosas. Un protocolo de comunicación se refiere a un conjunto de reglas que habilitan a dos entidades ubicadas en un mismo sistema a establecer una comunicación, con la finalidad de transmitir información a través de distintas variables. Elegir el protocolo de IoT más adecuado implica evaluar con exactitud diversos criterios como límite de consumo energético, ancho de banda de datos y latencia, calidad del servicio, ciberseguridad, etc. [Gerodimos et al., 2023]. Podemos nombrar como los protocolos más utilizados para la capa de aplicación de estas tecnologías a MQTT, CoAP, REST, XMPP y AMQP. Adicionalmente, existen otros protocolos utilizados en las demás capas que pueden encontrarse en una arquitectura IoT, como las capas de transporte, de red y física. Respecto a los protocolos mencionados, MQTT se ha convertido en un estándar en la industria debido a su simplicidad, eficiencia y seguridad, según diversas fuentes [Luna et al., 2020, Amazon Web Services, 2024].

Cuando estamos frente a entornos robotizados, que pueden ser utilizados para realizar tareas de fabricación de partes, transporte de materiales, control de calidad, entre otras, se necesitan también características extra en los sistemas de comunicación para poder interactuar con estos dispositivos ya sea para controlarlos, coordinarlos o simplemente monitorearlos. En este sentido, hace varios años se desarrolló ROS, el cual es un sistema operativo de código abierto destinado a sistemas robóticos. Algunas de las ventajas que ofrece ROS son el rápido desarrollo del comportamiento de robots, la reutilización de algoritmos, el hecho de poder trabajar con diferentes lenguajes de programación, además de que proporciona una

solución viable para sistemas de control distribuido permitiendo que cada robot trabaje de forma modular. En la Sección 4.2 se presenta con mayor detalle este sistema, con sus características, ventajas y posibilidades.

### 3.4 Optimización metaheurística

En términos generales, la optimización implica encontrar los valores óptimos de los parámetros o variables involucradas en un problema para maximizar o minimizar el resultado final. En términos matemáticos, la formulación de un problema de optimización busca maximizar o minimizar una función objetivo  $f(x)$  sujeta a ciertas restricciones. Esta función objetivo se debe formular de tal forma que contemple, cuantitativamente, todos los aspectos que se deseen optimizar (costos, desperdicios, cantidad de materias primas, etc.). En consecuencia, dicha función objetivo puede ser tan compleja como se requiera. Resolver un problema de optimización consiste en encontrar las magnitudes de las variables intervinientes en la función objetivo tales que maximizan (o minimizan) su resultado. Estas variables, que se conocen como variables de decisión, pueden manipularse e influyen en el comportamiento del sistema [Winston, 2004].

Existen varios enfoques para resolver diferentes problemas de optimización [Thirunavukkarasu et al., 2023, Gambella et al., 2021]. Entre los muchos métodos para abordar este tipo de problemas, se encuentran los métodos metaheurísticos. Una característica distintiva de estos métodos es que no requieren necesariamente de un modelo matemático del sistema a optimizar y, simplemente, trabajan modificando las entradas del sistema y monitorean sus salidas para maximizarlas o minimizarlas. Por esta razón, a estos métodos de optimización también se los suele llamar “métodos caja negra”. Esto constituye una ventaja dado que es posible utilizarlos en interacción con sistemas simulados digitalmente sin necesidad de un modelo matemático explícito. Además, los métodos de optimización de caja negra están diseñados para tratar sistemas complejos con comportamiento también complejo. Sin embargo, estos métodos pueden converger lentamente o quedarse estancados en mínimos locales. Por otro lado, los métodos metaheurísticos, que son algoritmos aproximados de optimización y búsqueda de propósito general, son fundamentales para resolver problemas grandes porque pueden producir resultados aceptables en un período de tiempo razonable, lo que los convierte en sustitutos adecuados de algoritmos precisos [Karimi-Mamaghan et al., 2022].

En el campo de los algoritmos metaheurísticos, existe un tipo específico de familia de algoritmos que son los basados en poblaciones. Se pueden encontrar cinco clases principales: basados en la evolución, que se inspiran en los fenómenos evolutivos de la naturaleza (como los algoritmos genéticos, por ejemplo); basados en inteligencia de enjambre, en los que la fuente de información son los comportamientos colectivos en la naturaleza (por ej. colonias de hormigas); basados en eventos provocados por el hombre (como el algoritmo Harmony search); basados en la física, que imitan reglas y principios físicos (como el Algoritmo de Búsqueda Gravitacional) y basados en las matemáticas, como el SCA [Benmessaoud et al., 2021].

Un algoritmo metaheurístico basado en población recientemente propuesto es el desarrollado por [Mirjalili, 2016] llamado Algoritmo de Seno Coseno (SCA). Este algoritmo fue propuesto para optimizar problemas reales con espacios de búsqueda desconocidos y ha sido poco explorado en relación a otros comparables de su misma

familia. Resumidamente, el funcionamiento de este algoritmo de optimización inicia ensayando varias soluciones candidatas iniciales aleatorias que se actualizan en comparación con la mejor solución disponible. Estas soluciones se encuentran cerca o lejos de la mejor solución, basadas en un modelo matemático de funciones seno y coseno. Además, existen cuatro parámetros integrados en el algoritmo  $r_1, r_2, r_3$  y  $r_4$  que permiten gestionar la exploración y explotación del espacio de búsqueda o zona factible. Cuando decimos exploración, nos referimos a que el algoritmo pueda encontrar nuevas soluciones, alejadas de las ya vistas; mientras que la explotación se refiere a la capacidad de refinar las soluciones ya encontradas y mejorarlas. La ventaja de este algoritmo, respecto a los demás basados en la población, es que muestra una alta eficiencia con una fácil implementación. La ecuación fundamental es la siguiente:

$$X_i^{t+1} = \begin{cases} X_i^t + r_1 * \sin(r_2) * |r_3 P_i^t - X_i^t|, & r_4 < 0,5 \\ X_i^t + r_1 * \cos(r_2) * |r_3 P_i^t - X_i^t|, & r_4 \geq 0,5 \end{cases} \quad (3.1)$$

donde  $X_i^t$  es la posición de la solución actual en la iteración  $t$  y la dimensión  $i$  y  $P_i^t$  es la posición de la mejor solución encontrada hasta el momento. La fórmula de  $r_1$  es  $r_1 = a - t \frac{a}{T}$ , donde  $a$  es una constante,  $t$  la iteración actual y  $T$  el número máximo de iteraciones del algoritmo.  $r_2$  es un número aleatorio que varía entre 0 y  $2\pi$ ,  $r_3$  es un número aleatorio que varía entre 0 y 2, y  $r_4$  es un número aleatorio que oscila entre 0 y 1. El pseudocódigo del SCA se muestra a continuación en el Algoritmo 1.

---

**Algorithm 1** Algoritmo de Seno Coseno

---

- 1: **Definir**  $T =$  máximo número de iteraciones
  - 2: **Inicializar** un conjunto de agentes de búsqueda o soluciones ( $X$ )
  - 3: **Evaluar** cada agente en la función objetivo
  - 4: **Actualizar** la mejor solución encontrada hasta el momento ( $P = X^*$ )
  - 5: **Mientras**  $t < T$  :
  - 6:     **Actualizar**  $r_1, r_2, r_3$  y  $r_4$
  - 7:     **Actualizar** las posiciones de los agentes de búsqueda usando la Ec. 3.1
  - 8:     **Evaluar** cada agente en la función objetivo
  - 9:     **Devolver** la mejor solución encontrada como la solución global
- 

En base a este algoritmo, podemos encontrar una variante del mismo, en este caso para la resolución de problemas de naturaleza discreta. Esta variante se conoce como DSCA o Algoritmo Discreto de Seno Coseno, y fue desarrollada por [Gupta et al., 2022]. La ecuación fundamental del Algoritmo 1 se discretiza de la siguiente manera para el DSCA:

$$X_{i,nueva}^{t+1} = X_i^t \oplus (C_i^t \otimes (X_d^t \ominus X_i^t)) \quad (3.2)$$

$$C_i^t = \begin{cases} |r_1 * \sin(r_2)|, & r_3 > 0,5 \\ |r_1 * \cos(r_2)|, & r_3 \leq 0,5 \end{cases} \quad (3.3)$$

donde el parámetro  $r_1 = \frac{t}{T}$  asciende linealmente entre 0 y 1,  $r_2$  es un número aleatorio que oscila entre 0 y  $2\pi$ , y  $r_3$  es un número aleatorio que oscila entre 0 y 1.

Como en el SCA original,  $X_i^t$  es la posición de la solución actual, donde  $X_d^t$  es la mejor solución obtenida hasta ahora. El parámetro  $C_i^t$  se utiliza para decidir la cantidad de cambio requerida en el vector de posición del candidato actual para obtener un nuevo estado. Es decir, en  $X_d^t \ominus X_i^t$  se calcula la diferencia de posiciones de los vectores: cuántos componentes en el vector solución  $X_i$  son diferentes de la mejor solución encontrada  $X_d$ . Después de la multiplicación en la Ecuación 3.2, se redondea el valor final para actualizar la solución, cambiando el número de características o componentes obtenidos.

Como se puede ver comparando las Ecuaciones 3.1 y 3.2, en el algoritmo discreto hay un parámetro menos a considerar ( $r_1, r_2, r_3$ ) con respecto al SCA. El pseudocódigo del DSCA se muestra en el Algoritmo 2.

---

**Algorithm 2** Algoritmo Discreto de Seno Coseno

---

- 1: **Definir**  $T$  = máximo número de iteraciones
  - 2: **Inicializar** un conjunto de agentes de búsqueda o soluciones ( $X$ )
  - 3: **Evaluar** cada agente en la función objetivo
  - 4: **Actualizar** la mejor solución encontrada hasta el momento ( $X_d$ )
  - 5: **Mientras**  $t < T$  :
  - 6:     **Calcular** para cada agente la diferencia en características de los vectores:  
 $X_d^t \ominus X_i^t$
  - 7:     **Actualizar**  $r_1, r_2$  y  $r_3$ ; y **calcular**  $C_i^t$
  - 8:     **Actualizar** las posiciones de los agentes de búsqueda usando la Ec. 3.2
  - 9:     **Evaluar** cada agente en la función objetivo
  - 10:    **Devolver** la mejor solución encontrada como la solución global
- 

Como el rendimiento del DSCA depende en gran medida de los parámetros  $r_i$ , en esta tesis se propone una versión mejorada con un algoritmo de AR incorporado para el aprendizaje de políticas de autoajuste de parámetros, lo cual se verá en el Capítulo 7.

### 3.5 Inteligencia artificial para la toma de decisiones

Desde la Segunda Guerra Mundial aproximadamente, los humanos han comenzado a desarrollar máquinas y sistemas capaces de responder y aprender de manera similar a los seres vivos. El término Inteligencia Artificial (IA) se acuñó hacia 1960 para definir los métodos algorítmicos bioinspirados capaces de dotar a los ordenadores de capacidades de razonamiento y toma de decisiones [del Brío and Molina, 2006]. Desde entonces, con el avance exponencial de la tecnología, el aumento de la capacidad de cómputo y la mejora en la velocidad de procesamiento de datos, han surgido diversas técnicas de IA capaces de abordar una amplia variedad de problemas en distintos ámbitos del conocimiento y la industria.

En la actualidad, la IA ofrece un enorme potencial para la industria, que puede ser aprovechado para que los procesos productivos sean más eficientes, flexibles y confiables. La creciente digitalización industrial ha facilitado la generación, procesamiento y análisis de grandes volúmenes de datos, los cuales constituyen la base para la creación de representaciones digitales de plantas y sistemas de manufactura completos. Además, estos datos pueden utilizarse para integrar métodos y técnicas de IA en tiempo real, para la toma de decisiones de distinta

naturaleza, como por ejemplo, decisiones operativas vinculadas a la planificación de la producción, la gestión de recursos, control de equipos, entre otras. La combinación de IA con tecnologías como el Internet de las Cosas (IoT) y los sistemas ciberfísicos está transformando los sistemas de manufactura, volviéndolos cada vez más inteligentes y con capacidades de autoadaptación frente a condiciones operativas dinámicas.

Las técnicas de IA pueden clasificarse en tres grandes categorías: aprendizaje supervisado, aprendizaje no supervisado y aprendizaje por refuerzo. Estas técnicas difieren principalmente en la cantidad y tipo de información de entrada que reciben, así como en la cantidad de intervención humana requerida durante el proceso de aprendizaje. En el aprendizaje supervisado, el algoritmo se entrena con un conjunto de datos etiquetados, lo que significa que cada entrada de datos se asocia con una salida conocida. Este enfoque es ampliamente utilizado para resolver problemas de clasificación y regresión. Esto puede aprovecharse en un sistema industrial de muchas maneras, como por ejemplo, para el reconocimiento de piezas defectuosas a partir de imágenes, la predicción de demanda de un producto, la predicción de fallas de un equipo a partir de los valores sensados de alguna magnitud (ej. vibraciones) entre otros [Rehman et al., 2025, Perez Colo et al., 2023]. Por el contrario, en el aprendizaje no supervisado, el algoritmo no recibe etiquetas predefinidas, sino que debe inferir patrones y estructuras subyacentes en los datos. Estos métodos se pueden aplicar, por ejemplo, para detectar patrones respecto a la segmentación de clientes, la detección de anomalías, entre otros [Kiefer et al., 2024, Akman et al., 2023]. A diferencia de los dos anteriores, el aprendizaje por refuerzo se basa en la interacción de un agente con un entorno, donde el algoritmo aprende a tomar decisiones óptimas que maximizan las recompensas obtenidas por los efectos de las acciones ejecutadas. El AR se ha convertido en una herramienta clave en la optimización de procesos industriales, la robótica y los sistemas autónomos. A continuación, se presentan resumidamente los principales elementos del aprendizaje por refuerzos dado que, junto a los algoritmos de optimización metaheurísticos presentados anteriormente, constituyen las bases metodológicas de los desarrollos de este trabajo de tesis.

### 3.5.1 Aprendizaje por refuerzo

En el AR o *Reinforcement Learning*, un agente trata de aprender qué hacer en una determinada situación basándose en estímulos obtenidos por el efecto de sus decisiones, trabajando a prueba y error. En otras palabras, el agente artificial aprende cómo asignar acciones a situaciones (estados) para maximizar una señal de recompensa numérica, con el fin de resolver una tarea específica de toma de decisiones secuenciales. En el AR, no se le “dice” de antemano al agente qué acciones tomar en determinadas situaciones, sino que debe descubrir aquellas que le permiten obtener la mayor cantidad de recompensa esperada. En los casos más interesantes y desafiantes, el AR es una alternativa poderosa para trabajar con sistemas en los que las acciones pueden tener un efecto retardado [Sutton and Barto, 1998, Sutton and Barto, 2018].

El AR es un enfoque computacional para aprender a partir de la interacción. En relación a la terminología utilizada, el agente es el tomador de decisiones y el entorno es todo lo que existe por fuera del agente. La política es el comportamiento del agente de aprendizaje en un instante específico. En términos generales, una

política representa una relación entre los estados que se perciben del ambiente y las acciones que se deben tomar cuando se está en dichos estados. En ciertas situaciones, la política puede ser una función sencilla, mientras que en otras podría tener una representación mucho más compleja.

En síntesis, en un problema de aprendizaje por refuerzos, el agente interactúa con el entorno durante una secuencia de pasos de tiempo discretos, tomando decisiones y obteniendo recompensas. Formalmente, este proceso se conoce como proceso de decisión de Markov, en el cual en cada paso de tiempo  $t$ , el agente recibe alguna representación del estado del ambiente  $s_t$  y en función de dicho estado selecciona una acción  $a_t$  siguiendo una política de actuación  $\pi(s_t)$ . Un paso de tiempo después, y como consecuencia de la acción tomada, recibe una recompensa numérica  $r_{t+1}$  y evoluciona a un nuevo estado  $s_{t+1}$ . Lo que se busca en el entrenamiento del sistema es maximizar la recompensa total a largo plazo [Sutton and Barto, 1998]. Este bucle se esquematiza en la Figura 3.4.

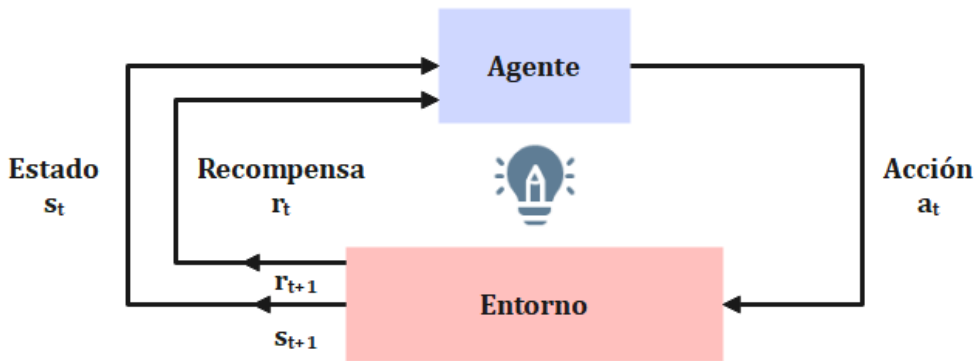


Figura 3.4: Esquema del aprendizaje por refuerzo.

En consecuencia, en un problema de toma de decisiones de Markov, en el momento  $t$ , una acción es un vector de decisiones  $\mathbf{a}_t$  seleccionadas para la acción que se aplica al sistema. Durante el proceso de aprendizaje, el agente interactúa con el sistema ejerciendo una acción  $\mathbf{a}_t \in A \subseteq \mathbb{R}^{n_a}$  y, luego de eso, el sistema evoluciona desde el estado  $\mathbf{s}_t \in S \subseteq \mathbb{R}^{n_s}$  a  $\mathbf{s}_{t+1}$  y el agente recibe una señal numérica  $r_t$  llamada recompensa (o castigo) que proporciona una medida de qué tan buena fue la acción tomada en  $\mathbf{s}_t$  en términos de la transición de estado observada. Las recompensas se dan como sugerencias sobre el logro de objetivos o el comportamiento óptimo.

La señal de recompensa define así cuáles son los eventos buenos y malos para el agente. Haciendo un paralelismo con los sistemas biológicos, podríamos pensar en las recompensas como análogas a las experiencias de placer o dolor en los animales. Por lo tanto, la señal de recompensa constituye el fundamento para modificar la política; si una acción elegida por la política es complementada con una recompensa baja, entonces la política podría ser modificada para escoger otra acción diferente en esa circunstancia en el futuro [Yau and McNally, 2023].

De esta manera, el objetivo de los métodos de AR es obtener la política óptima  $\pi^*$  que satisfaga la Ecuación 3.4, donde  $J_\pi$  es la recompensa total esperada bajo la política de control  $\pi$ . La política óptima  $\pi^*$  define las acciones óptimas ( $\mathbf{a}_t$ ) para diferentes estados del sistema ( $\mathbf{s}_t$ ), teniendo en cuenta las recompensas tanto a corto como a largo plazo.

$$J^* = \max_{\pi} J_{\pi} = \max_{\pi} E_{\pi} \{R_t | \mathbf{s}_t = \mathbf{s}\} \quad (3.4)$$

Supongamos que bajo una política  $\pi$  dada, la recompensa acumulada esperada  $V^{\pi}(\mathbf{s})$ , o función de valor durante un cierto intervalo de tiempo, es una función de  $\mathbf{s}^{\pi}$ , donde  $\mathbf{s}^{\pi} = \{\mathbf{s}_t\}_{t=1}^{t=n}$  son los valores de estado correspondientes y  $\mathbf{a}^{\pi} = \{\mathbf{a}_t\}_{t=1}^{t=n}$  define la política o secuencia específica de las acciones del agente. La secuencia  $\mathbf{s}^{\pi}$  de transiciones de estado da lugar a recompensas  $\{r_t\}_{t=1}^{t=n}$ . En tareas continuas, sin un único estado final, la suma descontada de recompensas futuras

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \quad (3.5)$$

se utiliza para definir la función de valor de estado esperado para una política del estado  $\mathbf{s}$ , como:

$$V^{\pi}(\mathbf{s}) = E_{\pi} \{G_t | \mathbf{S}_t = \mathbf{s}\} = E_{\pi} \left\{ \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | \mathbf{S}_t = \mathbf{s} \right\} \quad (3.6)$$

donde  $\gamma \in (0, 1]$  es el factor de descuento que pondera las recompensas futuras. Esta última ecuación se conoce como ecuación de Bellman y su importancia radica en la relación explícita que establece entre el valor actual y los valores de los estados sucesores. De manera similar, la función de valor de estado-acción se define como:

$$Q^{\pi}(\mathbf{s}, \mathbf{a}) = E_{\pi} \{G_t | \mathbf{S}_t = \mathbf{s}, \mathbf{A}_t = \mathbf{a}\} = E_{\pi} \left\{ \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | \mathbf{S}_t = \mathbf{s}, \mathbf{A}_t = \mathbf{a} \right\} \quad (3.7)$$

Cuando el agente se encuentra en un estado  $\mathbf{s}$  y ejecuta la política óptima  $\pi^*$ ,  $V^*(\mathbf{s})$  se utiliza para indicar la recompensa máxima obtenida con descuento. Por lo tanto, la función de valor de estado óptima asociada que satisface la ecuación de Bellman para todos los estados  $\mathbf{s}$  es:

$$V^*(\mathbf{s}_t) = \max_a \{r_t + \gamma E_{\mathbf{s}_{t+1}} [(V^*(\mathbf{s}_{t+1}) | \mathbf{s}_t, \mathbf{a}_t)]\} \quad (3.8)$$

donde  $\mathbf{a}_t = \pi^*(\mathbf{s}_t)$ . De manera similar, la función óptima de valor de estado-acción  $Q^*$  se define por:

$$Q^*(\mathbf{s}_t, \mathbf{a}_t) = r_t + \gamma E_{\mathbf{s}_{t+1}} [(V^*(\mathbf{s}_{t+1}) | \mathbf{s}_t, \mathbf{a}_t)] \quad (3.9)$$

tal que  $V^*(\mathbf{s}) = \max_a Q^*(\mathbf{s}, \mathbf{a})$  para todos los  $\mathbf{s}$ . Una vez que se conoce  $Q^*$  a través de interacciones, la política óptima se puede obtener directamente a través de:

$$\pi^*(\mathbf{s}) = \operatorname{argmax}_a Q^*(\mathbf{s}, \mathbf{a}) \quad (3.10)$$

En definitiva, la señal de recompensa indica lo que es bueno en un sentido inmediato, mientras que una función de valor específica lo que es bueno en el largo plazo. Por ejemplo, un estado puede generar constantemente una recompensa instantánea baja, pero mantener un valor elevado debido a que es seguido de cerca por otros estados que generan recompensas elevadas.

En el AR para resolver diversos problemas de toma de decisiones, existen diferentes algoritmos. Algunos de los más conocidos, de carácter tabular, son *Q-Learning* o *Sarsa*. Ambos algoritmos se explicarán y utilizarán en el Capítulo 6 para la resolución de uno de los casos de estudio presentados. Los algoritmos tabulares son simples, ya que representan las funciones de valor como vectores, matrices o tablas, debido a que los espacios de estado y acción de los problemas son lo suficientemente manejables para hacerlo. En este caso, los métodos a menudo pueden encontrar soluciones exactas, es decir, pueden encontrar exactamente la función de valor óptima y la política óptima.

Normalmente, los algoritmos de AR cuando trabajan en grandes espacios de estados y acciones, emplean aproximadores funcionales en lugar de tablas, como por ejemplo redes neuronales profundas, para aproximar su política de actuación  $\pi(\cdot)$  y las funciones de valor empleadas en dichos algoritmos [Goodfellow et al., 2016]. En la próxima subsección se introducen estas ideas en lo que se conoce como Aprendizaje por refuerzo profundo.

### 3.5.2 Aprendizaje por refuerzo profundo

En determinadas ocasiones, estamos en presencia de un entorno con muchos estados y acciones posibles que dificultan el proceso de aprendizaje y la ejecución de los algoritmos tabulares. En muchas tareas, el espacio de estados puede ser combinatorio y demasiado grande. En tales casos es muy costoso, o prácticamente imposible, encontrar una política óptima o la función de valor óptima incluso en el límite de tiempo y datos infinitos. Por lo tanto, se busca encontrar una buena solución aproximada utilizando recursos computacionales limitados [Sutton and Barto, 1998]. Una opción para lograrlo son las Redes Neuronales Artificiales (RNA).

Las RNA son sistemas de procesamiento, hardware o software, que copian esquemáticamente la estructura neuronal del cerebro para tratar de reproducir sus capacidades. Por lo tanto, son capaces de aprender de la experiencia a partir de señales o datos provenientes del exterior [Haykin, 2009]. Los tres conceptos clave de los sistemas nerviosos que se pretenden emular en las redes artificiales son: paralelismo de cálculo, memoria distribuida y adaptabilidad al entorno [del Brío and Molina, 2006].

La estructura de un sistema neuronal artificial es similar, jerárquicamente, a uno biológico. El elemento esencial es la neurona artificial. Cada neurona posee entradas con pesos sinápticos específicos que determinan el valor a ser procesado en su interior mediante una función de activación que devuelve la salida de la neurona. La agrupación de neuronas conectadas entre sí determina una RNA. La misma se organiza en niveles denominados capas. Esta red, junto con las interfaces de entrada y salida, formará el sistema global de aprendizaje.

Las RNA responden a un conjunto de estímulos de entradas (datos), recepcionados por la primera capa de neuronas, una regla de propagación de las señales hacia adelante y un valor final procesado por la última capa. Generalmente, en cuanto a su topología o estructura, las RNA distinguen tres tipos de capas de neuronas: de entrada, de salida y ocultas (que no tienen interacción con el exterior). Las RNA aprenden examinando los registros individuales, generando una predicción para cada registro y realizando ajustes a las ponderaciones cuando realizan una predicción, ya sea correcta o incorrecta. Este proceso se ejecuta

reiteradas veces, mejorando sus predicciones hasta haber alcanzado uno o varios criterios de detención.

Deep Q-learning Network (DQN) (Figura 3.5), otro de los algoritmos que se utilizaron en este trabajo, es el primer método de aprendizaje profundo desarrollado que ha demostrado capacidades de convergencia en el aprendizaje [Mnih et al., 2015]. Fue propuesto por DeepMind en el año 2015. El algoritmo combina AR con RNA; es decir, aproxima una función de valor de estado en el marco del *Q-Learning* con una red neuronal profunda. Esto permite utilizar el aprendizaje por refuerzo en casos más complejos y reales, donde el agente interactúa con un entorno que posee varios parámetros a tener en cuenta y la información es multidimensional, aprovechando las ventajas que proporcionan las RNA. En otras palabras, el algoritmo DQN permite generalizar el aprendizaje para trabajar con grandes espacios de estados; a la vez que permite trabajar con acciones discretas, situación que es muy frecuente en los problemas de toma de decisiones en sistemas de manufactura, como se verá en capítulos posteriores.

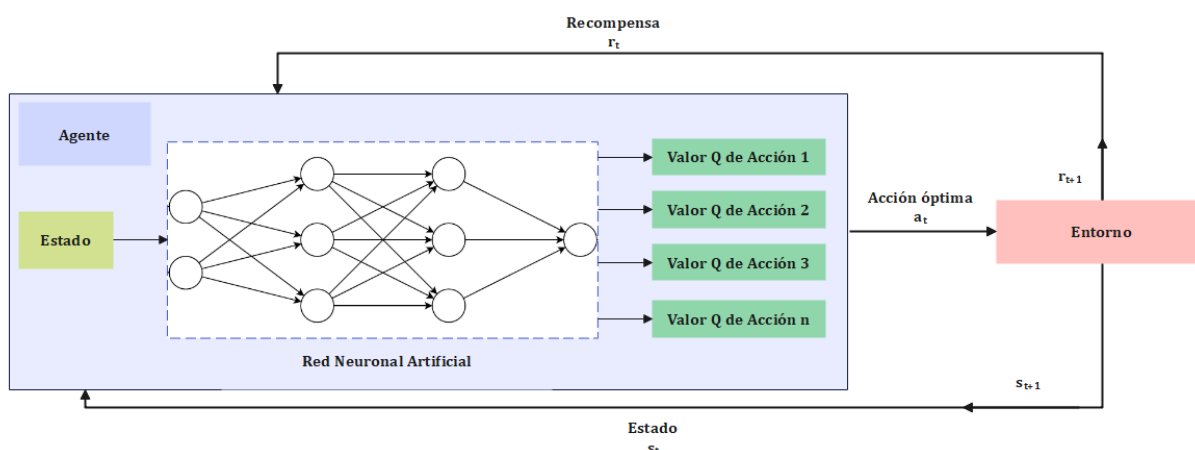


Figura 3.5: Esquema del algoritmo DQN.

Por ejemplo, en el artículo original [Mnih et al., 2015], el algoritmo DQN se probó en diversos juegos Atari tomando varios fotogramas como vectores de estado de entrada para cada acción. Se demostró que una sola arquitectura de red puede aprender con éxito las políticas de control en diversos entornos con mínimo conocimiento previo, utilizando el mismo algoritmo e hiperparámetros en cada juego.

Uno de los elementos distintivos del algoritmo DQN es la utilización del mecanismo de repetición de experiencia o “*Experience replay*”. Para cada iteración del agente con el entorno realizada durante la fase de entrenamiento, la información obtenida de la transición de estado ( $s_t, a_t, r_t, s_{t+1}$ ) se almacena en un conjunto de datos, llamado “*Replay buffer*”. Esto permite que durante el aprendizaje se extraigan subconjuntos de datos (transiciones almacenadas) de la memoria al azar para alimentar un mecanismo de aprendizaje off-line que evita los problemas ocasionados por la correlación en la secuencia de datos. Con esto se logra obtener mayor estabilidad en el proceso de aprendizaje global. Por otra parte, el algoritmo utiliza dos RNA durante el entrenamiento y optimización. Por un lado, la Q-Network (*policy net*) para la estimación del valor de función actual, la que, a su vez, se optimiza teniendo en cuenta una red objetivo congelada (*target net*) la cual posee

una actualización periódica con los últimos pesos obtenidos en cada paso. Esta última permite obtener los valores de función esperados y, por lo tanto, el cálculo del error (*loss*) respecto a la política. La actualización constante de los pesos de la red permite que el entrenamiento sea más estable al evitar oscilaciones a corto plazo.

Todas las características mencionadas permiten obtener un algoritmo de AR con mayor robustez, reduciendo la inestabilidad y divergencias que pudieran existir al utilizar una RNA para el procesamiento de datos y aprendizaje inteligente en entornos complejos.



## Parte II

# Sistema autónomo de toma de decisiones para la Industria 4.0



# Capítulo 4

## Componentes de los gemelos digitales para entornos robotizados

En este capítulo se conceptualiza sobre la composición de la estructura de un GD para los sistemas de manufactura que involucran robots en sus procesos. Se presenta un enfoque jerárquico basado en tres niveles interconectados. Además, se analiza la infraestructura de software necesaria para la implementación de un GD, destacando el uso de ROS para efectuar la comunicación entre los modelos digitales y los dispositivos físicos. Finalmente, se exploran los entornos de simulación que facilitan la representación del sistema productivo y su interacción con la realidad operativa.

### 4.1 Estructura de un gemelo digital para sistemas de manufactura

Desde una perspectiva jerárquica, los GDs pueden conceptualizarse como estructuras compuestas por tres niveles diferenciados pero interrelacionados: *Entidades, Información y Sistema*. La Figura 4.1 ilustra esta organización mediante una composición solapada de los tres niveles, evidenciando la interacción continua entre ellos y su rol en la construcción de un GD robusto y funcional.

En el primer nivel se ubican los modelos digitales de cada una de las entidades que intervienen en el sistema físico. Cada una de estas entidades debe ser representada con la mayor fidelidad posible para garantizar que el GD refleje con precisión el comportamiento del sistema real. Por entidad, nos referimos específicamente a maquinarias y equipos automáticos y robotizados. Actualmente, existen múltiples soluciones y plataformas en el ámbito industrial que ofrecen herramientas especializadas para modelar digitalmente diferentes tipos de entidades, como máquinas o robots. Para cada una de estas se han desarrollado simuladores computacionales de alto rendimiento, cuya eficacia ha sido ampliamente validada en diversas aplicaciones industriales. Por ejemplo, en un sistema productivo pueden intervenir desde equipos robotizados para manipulación de partes hasta equipos que realizan alguna operación unitaria de transformación físico-química. En consecuencia, cada equipo (entidad) puede ser representado digitalmente usando el entorno de simulación específico (muchas veces provisto por el fabricante). En este sentido, cada entidad presente en el proceso productivo puede concebirse

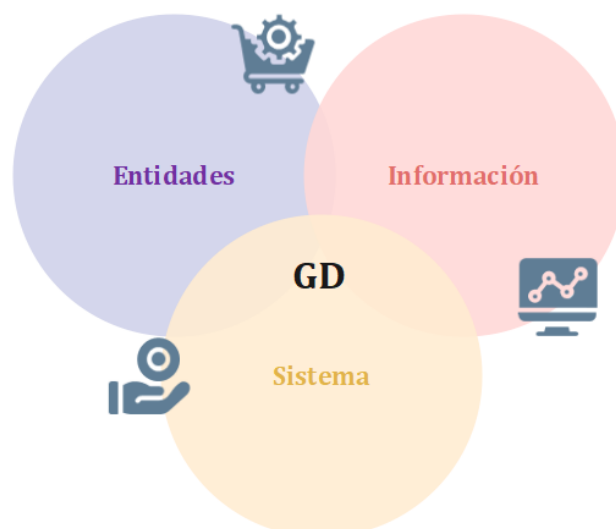


Figura 4.1: Relación funcional de un GD para un sistema de fabricación.

como un subsistema independiente, el cual, a su vez, formará parte de un sistema más amplio e interconectado. La disponibilidad de herramientas computacionales avanzadas permite la creación de modelos digitales de alta fidelidad, asegurando que cada subsistema sea representado con el nivel de detalle necesario para su integración dentro del GD global. En muchos casos, los simuladores de cada entidad son desarrollados y provistos por fabricantes o desarrollados en una comunidad específica. Justamente, una de las propuestas de esta tesis consiste en aprovechar esta situación. Es decir, se busca generar una propuesta que permita integrar fácilmente los simuladores especializados, de cada una de las entidades que componen el sistema productivo, con un simulador de planta para conseguir un GD del proceso productivo en su conjunto. De esta manera, es posible lograr representar digitalmente todos los agentes interactuantes en el proceso productivo de manera rápida y efectiva.

Además de la representación de máquinas y equipos, en el primer nivel también es fundamental modelar la interacción entre los operarios y el sistema de manufactura, para poder evaluar su impacto en los procesos y la eficiencia operativa. La presencia de operadores y personal técnico desempeña un papel fundamental en múltiples procesos de producción, por lo que su comportamiento puede ser modelado digitalmente mediante herramientas de simulación especializadas [Baskaran et al., 2019]. Este enfoque permite analizar la influencia de factores humanos en la productividad y seguridad del sistema, optimizando la asignación de tareas y mejorando las condiciones laborales mediante simulaciones avanzadas.

En un sistema productivo, las entidades no operan de manera aislada, sino que están interrelacionadas y comparten datos e información, lo que implica que su modelado debe considerar estas interacciones de manera sinérgica y sistémica. Es decir, cada una de las entidades debe estar adecuadamente modelada y vinculada correctamente con las demás (relaciones causales [Sterman, 2000]) cuando se genera el modelo digital del sistema productivo en su conjunto. En otras palabras, por ejemplo, de nada sirve modelar de manera precisa y aislada un centro de trabajo, un robot manipulador o un operario si no se definen correctamente todas las relaciones y propiedades necesarias dentro del entorno fabril en el que se están empleando.

La precisión y utilidad del GD dependen en gran medida de la coherencia en la integración de estos modelos individuales dentro de un marco sistémico cohesionado.

Una de las mayores dificultades que nos encontramos al intentar generar un modelo digital del sistema productivo en su conjunto, tiene que ver con la heterogeneidad y la complejidad de cada una de las entidades que pueden intervenir en un sistema de manufactura flexible. Por esto, para la generación de GDs de alta fidelidad es conveniente utilizar cada uno de los simuladores específicos y adecuados para cada entidad. En este sentido, el desarrollo de un GD para sistemas completos de producción manufacturera presenta un desafío adicional: la vinculación efectiva entre diferentes entornos de simulación y, además, con las entidades físicas en tiempo real. Hasta el momento, este aspecto ha sido escasamente abordado en la literatura científica, lo que subraya la necesidad de estrategias innovadoras para lograr una integración efectiva.

El segundo nivel de la jerarquía de un GD corresponde a la capa de *Información*. En este nivel, cada uno de los subsistemas modelados no opera de manera aislada, sino que mantiene un comportamiento dinámico basado en el intercambio continuo de datos. Dicho intercambio puede involucrar datos sobre tipo y cantidad de materiales, energía, insumos, datos operacionales, señales de control, entre otros elementos críticos dentro del proceso productivo. La recopilación, almacenamiento y procesamiento de estos datos permiten generar información estructurada que facilita la toma de decisiones en tiempo real. En este punto, la implementación de sistemas de monitoreo y adquisición de datos es un aspecto crítico para capturar la información relevante de cada entidad y transformarla en conocimiento accionable.

Finalmente, en el nivel de *Sistema* se consolida la integración de todas las entidades y flujos de información dentro de una estructura holística y funcional. En este nivel, el GD se convierte finalmente en una réplica virtual del sistema de manufactura, permitiendo la supervisión, el análisis y la optimización de los procesos en un entorno digital. La interoperabilidad entre los diferentes modelos digitales y la sincronización con los datos del sistema físico posibilitan la simulación de escenarios, la predicción de fallas y la mejora de la eficiencia operativa mediante estrategias de optimización basadas en IA y aprendizaje automático.

En conclusión, la construcción de un GD requiere una estructuración jerárquica que permita una representación precisa de cada entidad individual, la integración de los flujos de información y la consolidación de un modelo de sistema que refleje fielmente la realidad operativa. Esta metodología garantiza que el GD no solo funcione como una réplica digital, sino como un entorno dinámico capaz de evolucionar en respuesta a las condiciones cambiantes del sistema productivo.

A continuación, se presentará el desarrollo de una arquitectura de software que facilita la interconexión rápida entre distintos entornos de simulación y dispositivos físicos. Esto tiene la ventaja de poder utilizar una representación digital adecuada de cada una de las entidades que comúnmente se encuentran en los sistemas de manufactura flexible. Esta arquitectura, basada en ROS, actúa como un puente entre los modelos digitales y el sistema físico, garantizando un flujo de datos bidireccional que habilita el monitoreo, la simulación y la optimización en tiempo real. La implementación de esta solución no solo mejora la capacidad de representación de los GDs, sino que también permite integrar aplicaciones para la toma de decisiones autónoma y la optimización de procesos industriales.

## 4.2 Integración de ROS en entornos industriales

En el pasado, la interacción y comunicación con los robots, así como entre ellos, era compleja y altamente dependiente de la plataforma robótica utilizada. La falta de estandarización limitaba la interoperabilidad y dificultaba la integración de nuevas tecnologías en los sistemas robotizados. Sin embargo, en la última década, esta situación ha cambiado radicalmente gracias al desarrollo y la expansión del Sistema Operativo Robot, conocido como ROS. Este entorno de código abierto se ha convertido en una herramienta clave para el desarrollo de aplicaciones robóticas, facilitando la modularidad, la escalabilidad y la integración de hardware y software heterogéneos.

ROS es un metasisistema operativo diseñado para su implementación en robots, proporcionando los servicios esenciales que se esperan de un sistema operativo convencional, incluyendo la abstracción de hardware, el control de dispositivos de bajo nivel, la implementación de funciones de uso común, el intercambio de mensajes entre procesos y la gestión de paquetes. Además, ofrece un conjunto de herramientas y bibliotecas que permiten el desarrollo, la ejecución y la prueba de código en múltiples plataformas computacionales [O’Kane, 2014]. Esta flexibilidad lo ha posicionado como un estándar en la comunidad de diseñadores y fabricantes de robots, promoviendo el desarrollo colaborativo y la reutilización de software.

A nivel técnico, ROS está compuesto por un conjunto de procesos heterogéneos e independientes denominados nodos, los cuales se ejecutan simultáneamente y tienen la capacidad de intercambiar información mediante un sistema de comunicación basado en tópicos. Un nodo puede tanto publicar como suscribirse a un determinado tópico, lo que facilita la transmisión eficiente de datos entre distintos módulos del sistema [Koubaa, 2016]. Esta estructura heterogénea permite que los desarrolladores encapsulen nodos programados en distintos lenguajes, como Python y C++, promoviendo la interoperabilidad y facilitando el crecimiento incremental de los sistemas robóticos.

Una de las principales ventajas de ROS radica en su capacidad para integrar nuevos dispositivos, tanto sensores como actuadores, de manera homogénea, sin importar si utilizan controladores propietarios o estándares diferentes. Esto posibilita la expansión y actualización de un sistema sin necesidad de rediseñar su infraestructura de software, lo que resulta particularmente beneficioso para proyectos de investigación tecnológica, desarrollo de prototipos y aplicaciones industriales en constante evolución. La versatilidad y modularidad de ROS han permitido su adopción en un amplio rango de aplicaciones, desde la robótica colaborativa hasta la automatización de procesos de manufactura avanzada.

En resumen, las principales características y ventajas de ROS incluyen: (1) la abstracción de hardware y el control de dispositivos de bajo nivel; (2) un eficiente sistema de intercambio de mensajes entre procesos; (3) la gestión de paquetes para facilitar la organización y reutilización del código; (4) herramientas especializadas para el desarrollo, prueba y ejecución de software en diversos lenguajes de programación; (5) capacidades de computación distribuida, permitiendo la ejecución de procesos en múltiples sistemas interconectados; (6) la reutilización de software mediante la integración de paquetes previamente desarrollados; y (7) la posibilidad de realizar pruebas rápidas y simulaciones avanzadas en entornos controlados. Gracias a estos atributos, todos los robots equipados con ROS comparten un mismo

criterio de comunicación, lo que simplifica la integración de nuevos módulos y mejora la eficiencia del desarrollo.

En la creación de GDs, la estandarización de los protocolos de comunicación es fundamental. Los GDs, al estar formados por modelos simulados enriquecidos con datos del sistema físico, requieren una interacción eficiente con las entidades reales. Para satisfacer esta necesidad, una arquitectura de software basada en ROS facilita la integración. Gracias a esta integración, todas las entidades que generan y comparten información, como dispositivos robóticos, operadores, máquinas-herramienta, sistemas de toma de decisiones y otros componentes del ecosistema de manufactura, se interconectan a través de nodos ROS específicos. Esto permite el intercambio de datos en tiempo real, lo que posibilita la actualización dinámica de los modelos digitales y mejora la precisión de las simulaciones y predicciones.

Asimismo, para el desarrollo de un GD en un sistema de manufactura completo, la utilización de entornos de simulación especializados en el modelado de plantas de fabricación se presenta como una estrategia adecuada para la rápida construcción del GD. Estos entornos permiten la representación detallada de todas las entidades involucradas, asegurando que los vínculos funcionales entre ellas sean correctamente modelados. La combinación de ROS con plataformas de simulación industrial no solo mejora la capacidad de monitoreo y control del sistema de manufactura, sino que también facilita la implementación de estrategias de optimización basadas en IA y aprendizaje automático.

En síntesis, la utilización de ROS en la construcción de un GD representa un avance significativo ya que posibilita la integración rápida de equipos robóticos en sistemas de manufactura inteligentes. Su capacidad para estandarizar la comunicación entre entidades heterogéneas, facilitar la interoperabilidad de hardware y software, y permitir el desarrollo de modelos digitales altamente precisos refuerza su rol como una tecnología fundamental en la evolución de la I4.0. Con la creciente adopción de estrategias de manufactura inteligente, la combinación de ROS con GDs abre nuevas posibilidades para la integración de sistemas de toma de decisiones autónoma, por ejemplo, para la planificación y coordinación dinámica de tareas en busca de lograr una mejora continua en el rendimiento del proceso productivo en su conjunto.

Teniendo en cuenta la idea anterior, supongamos el caso representado en la Figura 4.2. La misma muestra una representación esquemática de un sistema de fabricación hipotético en el que su proceso de producción involucra inicialmente diferentes dispositivos robóticos para realizar diferentes tareas (entidades verdes en la figura). Cada una de estas entidades puede ser digitalmente simulada mediante su correspondiente simulador (entidades rosas). Entonces, para el desarrollo del GD del sistema productivo, podemos aprovechar la arquitectura de nodos ROS (círculos lilas) de los dispositivos robóticos y sus simuladores específicos e interconectarlos mediante la arquitectura propuesta basada en ROS.

Bajo este esquema, supongamos ahora que deseamos evaluar una modificación en el sistema de producción y queremos evaluar el desempeño del mismo al incorporar nuevas entidades en proceso de fabricación. Es decir, por ejemplo, deseamos analizar el impacto de incluir  $m$  robots adicionales en el sistema para realizar alguna tarea específica. Precisamente, el impacto de esta modificación se debe evaluar modificando el GD del sistema e incorporando dichas  $m$  entidades. Entonces, aprovechando la posibilidad de incluir dinámicamente entidades mediante la

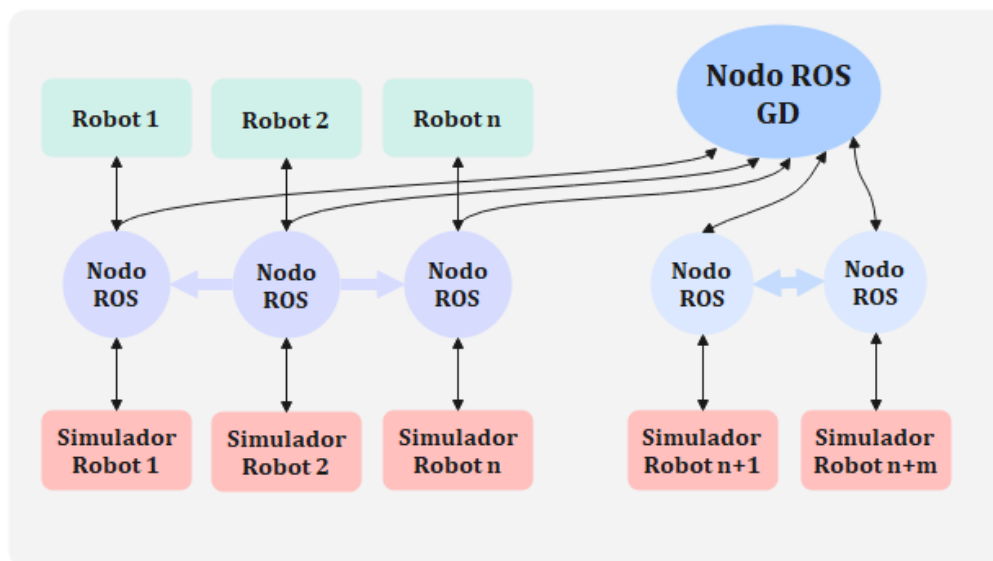


Figura 4.2: Sistema de fabricación robotizado hipotético y simplificado.

incorporación de nodos y simuladores, podemos añadir fácilmente estos dispositivos al GD de la planta mediante el correspondiente Nodo ROS (generalmente lo proporciona el fabricante o se encuentra disponible en una comunidad específica) y su simulador. Entonces, para analizar el impacto de esta configuración alternativa en el desempeño del proceso de fabricación en su conjunto, podemos incluir un modelo de eventos discretos de dicho sistema productivo que interactúa con los GDs de las entidades involucradas. Esta cuestión se abordará en la siguiente sección.

### 4.3 Entornos de simulación de planta

Los avances en las metodologías de simulación de eventos discretos [Zeigler et al., 2018] han impulsado en la última década el desarrollo en masa de múltiples entornos de simulación de eventos discretos adecuados para modelar, simular y optimizar procesos industriales. Sin embargo, uno de los principales desafíos que enfrentan los analistas al desarrollar un GD radica en la dificultad de vincular estos entornos con los equipos físicos de planta y en la adquisición de datos en tiempo real. La capacidad de establecer un esquema eficiente de intercambio de datos es fundamental para garantizar la sincronización entre el GD y su contraparte física, permitiendo un monitoreo preciso y la optimización del sistema productivo en tiempo real.

Una característica que vuelve atractivos a ciertos entornos de simulación de plantas industriales, como Tecnomatix Plant Simulation (TPS)<sup>1</sup> o FlexSim<sup>2</sup>, es la posibilidad de acceder a licencias de prueba libres y gratuitas. Esto resulta particularmente útil para la modelización de sistemas productivos en pequeñas y medianas empresas, donde la optimización de costos y la accesibilidad a herramientas de simulación avanzadas son factores determinantes. Estas plataformas no solo permiten la evaluación de escenarios sin necesidad de realizar inversiones iniciales

<sup>1</sup><https://plm.sw.siemens.com/en-US/tecomatix/plant-simulation-software/>

<sup>2</sup><https://www.flexsim.com/>

significativas, sino que también brindan un entorno robusto para analizar y mejorar la eficiencia operativa de los procesos productivos.

Además de su accesibilidad, estos entornos de simulación ofrecen la ventaja de una integración relativamente sencilla con software y hardware. Esto permite un intercambio de información, por ejemplo, entre un agente de software para análisis de datos y controladores lógicos programables (PLC) presentes en el piso de planta. Esta capacidad es esencial en la manufactura moderna, dado que en muchas plantas industriales prevalece una gran variedad de equipos de control y automatización que requieren interoperabilidad con los sistemas de simulación. La conexión con hardware físico permite validar estrategias de toma de decisiones antes de su implementación en la planta real, reduciendo riesgos y optimizando tiempos de puesta en marcha.

Otra ventaja significativa de emplear entornos de simulación de plantas es que poseen interfaces gráficas intuitivas, que permiten a los usuarios modelar sistemas de fabricación de manera dinámica y rápida. Estas herramientas facilitan la construcción de modelos al permitir la selección de componentes específicos, su posicionamiento dentro del entorno virtual y la definición de relaciones entre los distintos elementos del sistema. Además, la mayoría de estos entornos incorporan lenguajes de programación específicos que posibilitan la configuración de lógicas de acción y automatización de procesos, lo que amplía su aplicabilidad en entornos industriales con dinámicas operativas complejas. Por ejemplo, TPS utiliza el lenguaje SimTalk y FlexSim usa FlexScript. Cabe destacar que muchos de estos lenguajes son ad-hoc, es decir, han sido desarrollados específicamente para un simulador en particular. Esto implica que su aplicación es limitada fuera de ese entorno y que cuentan con una comunidad de usuarios reducida, lo que dificulta su adopción, aprendizaje y soporte técnico.

Pero, en este sentido, un aspecto clave que hace aún más atractivos a estos entornos de simulación es su capacidad para integrarse con el lenguaje de programación Python. En la propuesta que se presentará en este trabajo, esta integración desempeña un papel fundamental, ya que permite la comunicación fluida entre distintas instancias de simulación y su contraparte física. La conectividad con Python posibilita la vinculación de los modelos de simulación con simuladores dinámicos específicos que representan entidades clave dentro del sistema de manufactura.

Asimismo, una de las mayores ventajas de la integración con Python es la facilidad con la que se pueden incorporar bibliotecas avanzadas de procesamiento de datos e IA, herramientas que están en auge en la era de la I4.0. Algunas de las bibliotecas más relevantes incluyen Pandas y Numpy, utilizadas para el análisis y manipulación de datos; Scikit-Learn, Scipy y TensorFlow, fundamentales para la aplicación de técnicas de aprendizaje automático; y PyTorch, ampliamente empleado en el desarrollo de redes neuronales profundas. La posibilidad de combinar herramientas de simulación con estas bibliotecas de IA no solo permite optimizar los procesos productivos, sino que también facilita la implementación de modelos predictivos y estrategias de mantenimiento inteligente basadas en datos en tiempo real.

En conclusión, la evolución de los entornos de simulación ha permitido que su aplicación en la manufactura flexible sea cada vez más viable y eficiente. La capacidad de integración con hardware en lazo cerrado, la facilidad de modelado mediante interfaces gráficas intuitivas y la compatibilidad con Python y bibliotecas

avanzadas de IA convierten a estas herramientas en pilares fundamentales para el desarrollo de GDs. Estas características refuerzan la importancia de su adopción en la I4.0, donde la interconectividad, el análisis de datos y la automatización son esenciales para la optimización de la producción y la mejora continua de los sistemas industriales.

Entonces, retomando el ejemplo dado anteriormente y mostrado en la Figura 4.2, la Figura 4.3 muestra el caso en el que se incorpora un Entorno de Simulación de Planta de Manufactura (ESPM) para el modelado del sistema de fabricación.

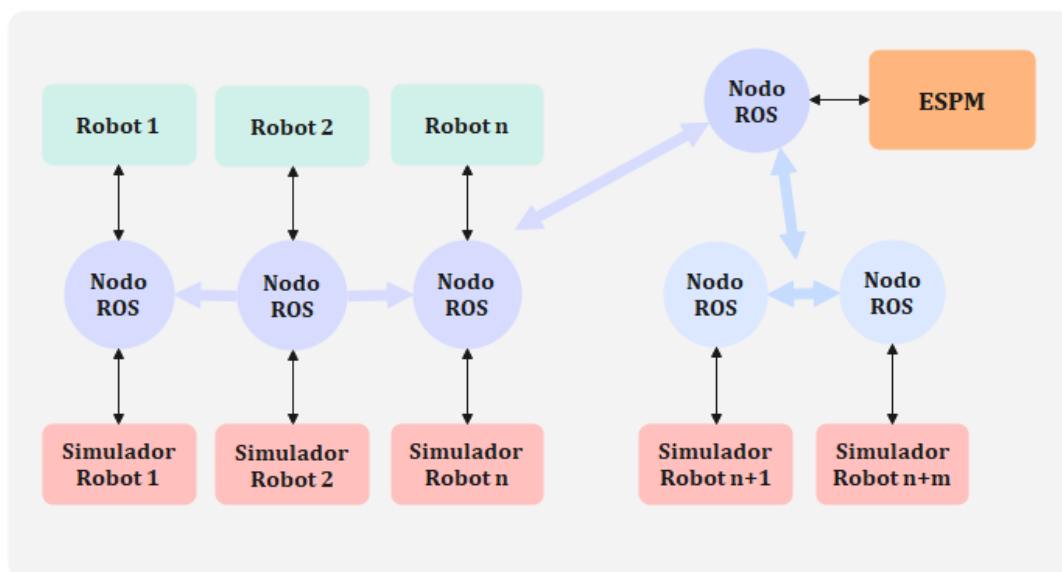


Figura 4.3: Gemelo digital de una hipotética planta de fabricación sencilla.

De esta manera, el ESPM permite tratar con todas las entidades vinculadas a través de su funcionalidad y al mismo tiempo se produce el intercambio de datos, que pueden ser utilizados para la actualización de cada entidad digital involucrada. Por lo tanto, el comportamiento dinámico de las entidades digitales se puede actualizar continuamente y, en consecuencia, sus comportamientos se pueden reflejar inmediatamente en el GD de todo el sistema de fabricación.

Además de las entidades robóticas, el ESPM permitirá simular y programar otras máquinas, estaciones, personas involucradas, recursos y aspectos logísticos, así como configurar diversas características del sistema de producción en su conjunto. Estos simuladores también proporcionan diferentes tipos de métricas sobre fallas, mantenimiento, calidad, consumo de energía, costos, manejo de personal, entre muchos otros. La utilización del ESPM contribuye a mejorar el flujo de materiales, el uso de recursos y la logística en todos los niveles de la planificación de planta.

A continuación, se muestran imágenes de las interfaces gráficas de FlexSim y TPS (Figuras 4.4 y 4.5), con su visualización 3D. Podemos apreciar en la primera imagen tablas y gráficas con métricas que permiten monitorear el proceso. Como se detalló anteriormente, las interfaces son intuitivas, con barras de herramientas que permiten fácilmente la creación de los objetos.

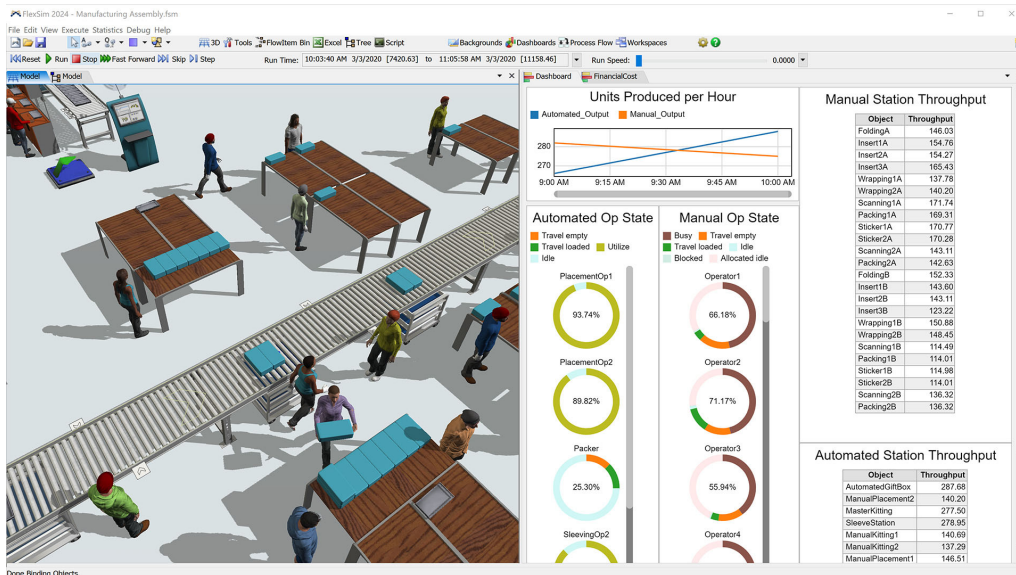


Figura 4.4: Interfaz de FlexSim.

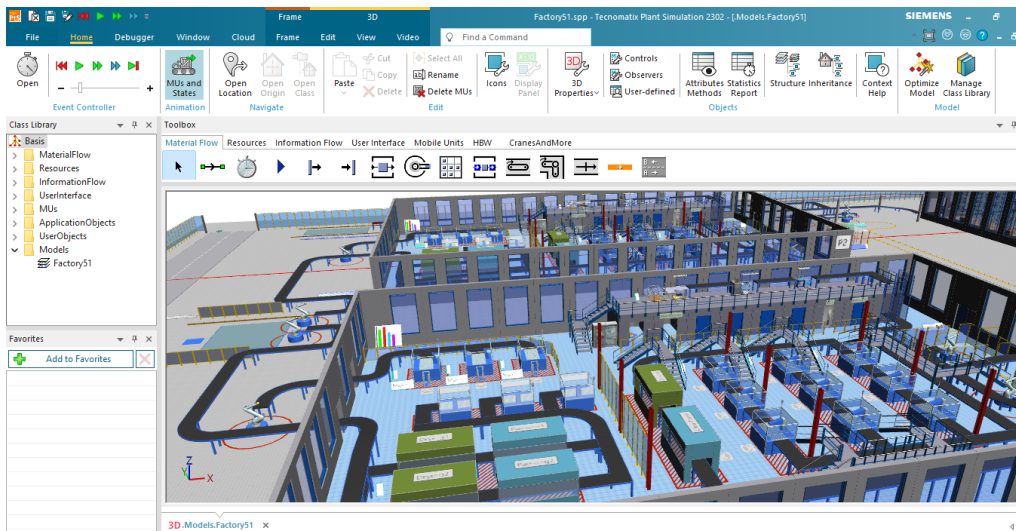


Figura 4.5: Interfaz de Tecnomatix Plant Simulation.

## 4.4 Conclusiones del capítulo

En el presente capítulo, se han explorado los fundamentos y las estrategias clave para la integración de los GDs en entornos de manufactura flexible, resaltando la importancia de la interoperabilidad entre hardware, software y plataformas de simulación. Se ha evidenciado que el desarrollo de una arquitectura robusta para la implementación de GDs requiere la combinación de diversas herramientas tecnológicas, como el sistema operativo de robots ROS, entornos de simulación avanzados y bibliotecas especializadas de procesamiento de datos e IA. Esta sinergia permite crear modelos digitales de alta fidelidad, capaces de reflejar con precisión el comportamiento del sistema físico y optimizar los procesos productivos mediante la toma de decisiones basadas en datos.

Uno de los aspectos centrales analizados ha sido el papel fundamental de ROS en la estandarización de la comunicación entre robots, sensores y sistemas de

control. Su capacidad para integrar dispositivos heterogéneos y su compatibilidad con múltiples lenguajes de programación han facilitado el desarrollo incremental de sistemas inteligentes, consolidándolo como una herramienta esencial en la I4.0. Además, la arquitectura basada en nodos y tópicos permite la interconexión eficiente de los componentes del sistema, asegurando una transmisión fluida de información en tiempo real y favoreciendo la actualización dinámica de los modelos digitales.

En este contexto, los entornos de simulación desempeñan un rol crucial en la validación de estrategias operativas y en la optimización de procesos industriales. Herramientas como TPS y FlexSim han demostrado ser soluciones efectivas para modelar sistemas de manufactura, especialmente por su capacidad de integración con hardware y su interfaz gráfica intuitiva. Su compatibilidad con el lenguaje de programación Python amplía significativamente sus aplicaciones, permitiendo integrar, por ejemplo, algoritmos avanzados de inteligencia para mejorar los procesos de toma de decisiones.

Además, se ha destacado la importancia de la correcta vinculación entre los modelos de simulación y los equipos físicos en planta. La creación de GDs debe considerar no solo la modelización precisa de cada entidad, sino también la adecuada interconexión entre ellas dentro del sistema productivo. En este sentido, se ha subrayado la necesidad de utilizar entornos de simulación específicos para cada tipo de entidad, asegurando que las relaciones funcionales y operativas sean representadas con la mayor fidelidad posible.

Por último, la convergencia entre GD, IA y procesamiento de datos en la manufactura flexible abre nuevas posibilidades para la optimización continua de los sistemas industriales. La integración de bibliotecas especializadas como Pandas, Numpy, Scikit-Learn y TensorFlow potencia el análisis predictivo y la automatización de la toma de decisiones, permitiendo una mayor adaptabilidad a los cambios en la demanda y mejorando la resiliencia de los procesos productivos.

En conclusión, el desarrollo de una arquitectura flexible y escalable para la implementación de GDs representa un paso clave hacia la digitalización de la industria manufacturera. La estandarización de protocolos de comunicación, la interoperabilidad entre sistemas y la incorporación de herramientas de IA permiten avanzar hacia fábricas inteligentes con capacidades de autogestión y optimización en tiempo real. La aplicación de estos conceptos en entornos productivos no solo mejora la eficiencia y la reducción de costos, sino que también sienta las bases para el desarrollo de nuevas estrategias de manufactura sostenible, sustentable y altamente adaptativa.

# Capítulo 5

## Desarrollo e integración del SATDD

En el presente capítulo se presenta el desarrollo de una arquitectura de software que permite integrar todos los elementos presentados en los capítulos anteriores. Dicha arquitectura permite articular el sistema ROS con simuladores de procesos de manufactura, para conseguir una representación precisa y dinámica del entorno productivo. Además, la arquitectura está diseñada para contemplar la posibilidad de incorporar un sistema autónomo de toma de decisiones, que tiene la posibilidad de supervisar, controlar y optimizar los procesos de fabricación mediante la implementación de cualquier tipo de algoritmo. En los casos de estudio que se mostrarán en los siguientes capítulos, se utilizarán algoritmos avanzados de IA.

### 5.1 Arquitectura software: Integración ROS - Entorno de simulación

En esta sección, introducimos detalladamente y por partes la arquitectura de software desarrollada que permite la integración entre ROS y entornos de simulación de planta. Utilizaremos la Figura 5.1, que esquematiza la arquitectura desarrollada, para explicar el desarrollo.

En la parte gris de la Figura 5.1, tenemos encerrado en un rectángulo a ROS. Este componente se encuentra integrado por dos nodos, o procesos heterogéneos e independientes, denominados *roscore* y *Simulación de Planta*. El nodo *roscore* representa una colección de procesos internos de ROS que facilitan la comunicación entre nodos dentro de una red heterogénea. El nodo *Simulación de Planta* representa el proceso que tiene la responsabilidad de dirigir una planta de manufactura simulada. A su vez, este proceso puede comunicarse con otros nodos por medio de tópicos del tipo suscriptores o publicadores. Por ejemplo, esto permite que una posible modificación en el comportamiento de alguna entidad que compone el sistema productivo pueda ser probada en el mundo virtual y, en caso de resultar conveniente, mediante el nodo *Simulación de Planta* dicho cambio puede ser publicado directamente a través de un tópico para modificar el comportamiento de la entidad real de la planta. A su vez, el nodo *Simulación de Planta* puede suscribirse a un tópico de datos proporcionado por una entidad real, lo que reflejaría el comportamiento que esté teniendo la entidad física. En consecuencia, este esquema de intercambio de información bidireccional permite generar el GD para un sistema industrial, como se explicó en capítulos anteriores.

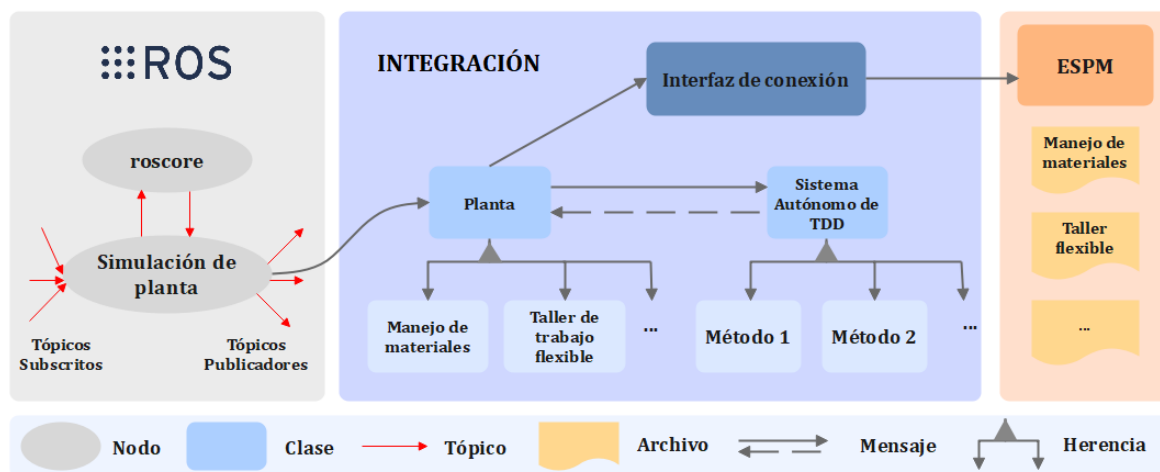


Figura 5.1: Arquitectura software para integrar ROS y un simulador de procesos de fabricación.

Por otro lado, en la parte color naranja de la Figura 5.1, se encuentra encerrado en un rectángulo el simulador de procesos de manufactura o ESPM. Para el caso particular de esta investigación, el simulador que se usará es el ya nombrado TPS. Con este software es posible diseñar plantas de manufactura adaptadas a cada problemática en particular con el objetivo de simular sus procesos de producción. Entonces, se pueden crear tantos modelos de una misma planta como sean necesarios. Cada uno de los modelos desarrollados se almacena en archivos externos con extensión *.spp*. Por ejemplo, en la Figura 5.1 se muestra un caso en el que para un mismo sistema productivo se desarrollaron dos modelos diferentes en el simulador TPS. Uno de los modelos permite generar el GD del sistema para analizar el problema puntual de manejo de materiales y, separadamente, otro modelo es para crear el GD del flujo de producción de un piso de planta flexible. De esta forma es posible generar tantas versiones digitales como sean necesarias para realizar análisis desde distintas perspectivas con diferentes propósitos. Como se explicó en la Sección 4.3, este entorno de simulación es de muy fácil utilización y brinda gran versatilidad. En el caso puntual de TPS, es un software estándar para modelado integrado, gráfico, jerárquico y orientado a objetos para simular y visualizar sistemas y procesos discretos y continuos, con animación en 2D y 3D. Su lenguaje de programación para los métodos internos y las entidades a simular se denomina SimTalk. Que sea un software orientado a objetos va en concordancia justamente con la integración que se desarrolló. Cada entidad simulada en TPS es un objeto y posee identidad, atributos y métodos.

En la parte central de la Figura 5.1, delimitada por el rectángulo violeta, se encuentra la capa de integración entre ROS y el ESPM. Como se puede ver, se propone un patrón de diseño de clases, siguiendo el paradigma de programación orientada a objetos, para el desarrollo de la arquitectura de software propuesta en este trabajo. El diseño propuesto se implementó y testeó íntegramente utilizando el lenguaje de programación Python. La comunicación entre ROS y el ESPM se realiza mediante la clase denominada **Interfaz de Conexión**, que tiene la responsabilidad de proporcionar puntos de acceso externos que realizan alteraciones dentro del software de simulación. Estos puntos de acceso comprenden la carga de un modelo de una planta en particular para ser procesado por el simulador, establecer y obtener

datos de una entidad cualquiera simulada, comenzar, parar y pausar simulaciones, guardar y cerrar modelos, entre otras acciones permitidas por el software TPS. Para implementar esta comunicación se utilizó la librería win32com, la cual permite intervenir de forma externa en cualquier sistema que sea ejecutado dentro del sistema operativo de Windows<sup>®</sup>. La clase **Interfaz de Conexión** representa el patrón de diseño *Mediator* utilizado comúnmente en el desarrollo de software de la programación orientada a objetos, el cual tiene la responsabilidad de encapsular la comunicación entre los objetos [Shaw and Garlan, 1996].

Siguiendo con el desarrollo de la arquitectura de software propuesta, como se puede observar el nodo *Simulación de Planta* de ROS ejecuta procesos mediante el pasaje de mensajes a una planta en particular utilizando la clase **Planta**. La clase **Planta** representa una clase abstracta y sigue el patrón de diseño denominado herencia simple, la cual abstrae todo el comportamiento general de cualquier planta de manufactura simulada [Bass et al., 2013]. Precisamente, la clase **Planta** ejecuta procesos mediante el pasaje de mensajes por medio de la clase **Interfaz de Conexión**, la cual representa un comportamiento general que requiere cualquier planta de manufactura simulada coordinada externamente. En este ejemplo de una jerarquía simple se tienen dos clases de plantas de manufactura concretas, denominadas Taller de Trabajo Flexible y Manejo de Materiales. Sin embargo, se pueden definir tantas clases planta como sea necesario implementar. Cada clase planta concreta implementa el comportamiento de coordinación de una planta simulada por el software de simulación. Por ejemplo, la clase concreta Manejo de Materiales tiene la responsabilidad de cargar dentro del software de simulación el archivo Manejo de Materiales con extensión *.spp*, cambiar el comportamiento de la planta simulada ante el establecimiento de los parámetros de las entidades simuladas, comenzar, parar y extraer resultados de una simulación, entre otras funciones. Como se mencionó, la comunicación siempre se realiza a través de los puntos de acceso brindados por la clase **Interfaz de Conexión**. Se debe tener en cuenta que cada comportamiento específico, de cada clase concreta de **Planta**, depende del caso de estudio que se esté analizando. Es decir, cada objeto derivado de la clase abstracta **Planta** permite redefinir sus métodos así como incorporar nuevos comportamientos (funciones) de acuerdo a las necesidades particulares del caso de estudio en cuestión. Por lo tanto, la extensión de una nueva planta concreta que dirija de forma externa una planta simulada por el software TPS es directa, simplemente se debe crear una nueva clase que herede de la clase **Planta** y codificar el comportamiento específico requerido, promoviendo la escalabilidad de la arquitectura de software propuesta.

Paralelamente a la jerarquía de planta, se tiene una herencia simple con la clase abstracta **Sistema Autónomo de TDD**. Esta clase abstrae todo el comportamiento general de cualquier método de toma de decisiones autónoma provisto por la IA u otra técnica utilizada. Una clase concreta representa la aplicación de un método particular de AR, por ejemplo. Se puede notar que el agregado de un nuevo método de decisión autónoma es directo, simplemente se debe crear una clase que herede de la clase **Sistema Autónomo de TDD** y codificar el método en particular, nuevamente fomentando la escalabilidad de la arquitectura de software propuesta.

Estas jerarquías paralelas interrelacionadas entre las clases abstractas **Planta** y **Sistema Autónomo de TDD** representan una combinación de patrones de diseño conocidos como *Factory Method* y *Observer* o *Publish-Subscribe* utilizados comúnmente en el desarrollo de software de la programación orientada a objetos

[Shaw and Garlan, 1996]. El patrón *Factory Method* tiene la responsabilidad de construir o fabricar objetos de un subtipo determinado, es decir, plantas simuladas, en la que se le puede o no aplicar dentro de cada subtipo un método en particular, en este caso, un proceso del sistema de decisión autónoma. Por otro lado, el patrón *Observer* o *Publish-Subscribe* define una dependencia entre el observador (la clase *Planta*) y el observado (la clase *Sistemas Autónomo de TDD*). De esta manera, una planta cualquiera, es decir, el observador, ejecuta un proceso en particular del sistema de decisión autónoma, es decir, el observado (ver Figura 5.1 el paso de mensaje de una flecha color gris). Luego, cuando el observado cambia de estado, por ejemplo, cuando el método de sistema de decisión autónoma obtiene un resultado en la optimización de un proceso en particular, el observador es notificado automáticamente (ver Figura 5.1 el paso de mensaje de una flecha color gris a trazos), teniendo nuevamente el control de la planta, y a su vez, el resultado del proceso de optimización para que, seguidamente, altere con el resultado obtenido la planta simulada en cuestión.

Observando con una mirada abstracta la figura podemos notar que desde cualquier nodo de ROS podemos crear numerosas plantas de manufactura a la vez, de igual o de distinto tipo, que interactúen con el simulador de procesos de manufactura TPS, y a su vez, a cada tipo de planta se le pueden aplicar o no métodos de un sistema de decisión autónoma. Para aclarar esta idea, supongamos un ejemplo sencillo en el que en una planta de fabricación queremos gestionar y optimizar el flujo de materiales. De esta manera, en primer lugar, podemos modelar e implementar la dinámica del sistema en el módulo abstracto de *Planta*. Luego, para integrar el proceso de fabricación con un sistema de apoyo a la decisión, podemos utilizar otro módulo abstracto llamado *Sistema Autónomo de TDD*. Este módulo puede integrar diferentes sistemas de toma de decisiones alojados en la clase heredada *Métodos*. En los capítulos subsiguientes, esto se comprenderá mejor con los casos de estudio presentados.

## 5.2 Modelo conceptual de la Arquitectura

A continuación, para facilitar una comprensión integral de la arquitectura presentada, se proporcionan representaciones gráficas en forma de diagramas UML. El Lenguaje Unificado de Modelado (UML, por sus siglas en inglés) constituye un estándar ampliamente adoptado en el ámbito del desarrollo de software para la representación gráfica de sistemas y sus relaciones. Estos modelos desempeñan un papel fundamental en el análisis y diseño de arquitecturas, pues permiten estructurar y documentar los componentes de un sistema de manera clara y concisa. A través de la abstracción y la estandarización de sus elementos, UML posibilita la identificación de relaciones entre módulos, la detección de posibles dependencias y la validación de diseños antes de su implementación efectiva.

Asimismo, la combinación de diagramas UML con fragmentos de código Python permite no solamente ilustrar la estructura conceptual del sistema, sino también establecer una correspondencia directa entre los modelos teóricos y su materialización en un entorno de desarrollo real. De este modo, se favorece una transición fluida entre el diseño arquitectónico y la implementación práctica, promoviendo la modularidad, la reutilización del código y el mantenimiento eficiente del software.

Por lo tanto, en la Figura 5.2 se expone la parte central de la arquitectura,

es decir, la integración visualizada en el rectángulo violeta de la Figura 5.1, como diagrama UML. Toda esta parte de la arquitectura se desarrolló en una librería llamada *Simulai*.

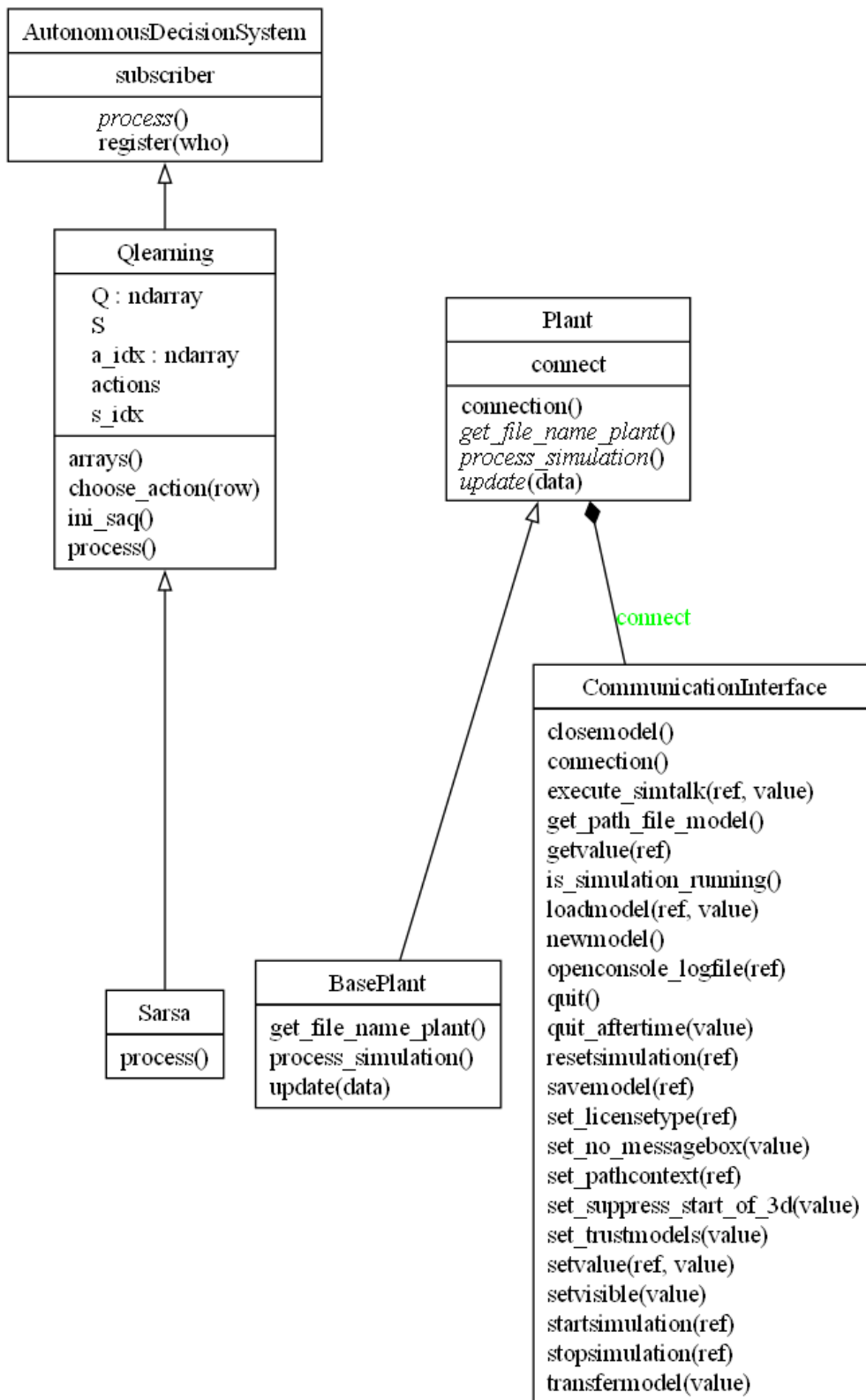


Figura 5.2: Diagrama UML de Simulai.

Respecto al código Python desarrollado, se agregan en el Apéndice al final del

documento, los correspondientes a las clases Interfaz de conexión, Planta y Sistema Autónomo de TDD. La Clase Interfaz de conexión posee funciones para interactuar con el simulador, como por ejemplo setear y obtener valores de los objetos simulados, abrir y cerrar modelos, correr simulaciones, ejecutar código SimTalk, entre muchas otras. Por otra parte, de las clases abstractas Planta y Sistema Autónomo de TDD se derivarán las clases particulares a emplear en un caso de estudio o ejemplo concreto.

Cabe destacar que para facilitar la reproducibilidad de la propuesta, la misma se encuentra a disposición en un repositorio web <sup>1</sup>, además del desarrollo de la librería *Simulai* mencionada anteriormente.

## 5.3 Sistema autónomo de toma de decisiones basado en un gemelo digital

Como ya se ha señalado, el paradigma de fabricación centrado en el cliente exige que las empresas busquen alternativas innovadoras para mejorar su competitividad y capacidad de respuesta ante entornos dinámicos. En este contexto, el desarrollo de los SATDD se presenta como una propuesta para optimizar la productividad global del sistema de fabricación y, en consecuencia, el uso eficiente de los recursos. Estos sistemas tienen la capacidad de asistir en la toma de decisiones en tiempo real, facilitando la adaptación de la producción a las fluctuaciones en la demanda de los clientes y a las perturbaciones inherentes a un sistema productivo [Ding et al., 2020]. La capacidad de estos sistemas para responder de manera proactiva a cambios en los procesos productivos representa un avance significativo en la gestión de la manufactura moderna.

Para que los SATDD sean eficaces, es fundamental que operen con información precisa y en tiempo real, lo que les permite formular políticas de actuación robustas y adaptativas. Esto requiere la integración de tecnologías avanzadas, como el IoT, la IA y los sistemas ciberfísicos, que facilitan la captura, procesamiento y análisis de grandes volúmenes de datos en entornos industriales [Ding et al., 2020, Kusiak, 2018]. Mediante el uso de modelos predictivos y aprendizaje autónomo, estos sistemas pueden, por ejemplo, prever fallos, mejorar la planificación de rutas internas y administrar de forma eficiente los inventarios y procesos operativos [Jay Lee, 2015].

En el pasado, los modelos digitales se utilizaban casi exclusivamente para analizar y validar el comportamiento de sistemas e instalaciones industriales de producción altamente automatizados. Sin embargo, la evolución tecnológica ha impulsado un cambio de paradigma, promoviendo la digitalización de la manufactura y la aplicación de IA en los procesos de toma de decisiones. En la actualidad, las tecnologías emergentes, como la simulación avanzada, la analítica de datos en tiempo real y el modelado de GDs, han ampliado significativamente el alcance de las herramientas de planificación digital, permitiendo su aplicación en sistemas de manufactura flexibles y reconfigurables [Thomas H. Uhlemann, 2017].

El desarrollo y la implementación de SATDD no solo permiten mejorar la eficiencia operativa, sino que también fomentan la sostenibilidad de los procesos productivos al reducir desperdicios, minimizar tiempos de inactividad y optimizar la utilización de los recursos. Este enfoque, alineado con los principios de la I4.0, facilita la transición hacia fábricas inteligentes, en las que la interconectividad y la

---

<sup>1</sup>[https://github.com/INTELYMEC/ROS\\_Tecnomatix](https://github.com/INTELYMEC/ROS_Tecnomatix)

automatización avanzada son elementos fundamentales para la competitividad en un mercado globalizado. En este sentido, la combinación de planificación digital, IA y GD representa un hito en la evolución de los sistemas productivos, estableciendo nuevas oportunidades para la optimización y personalización de la producción.

La propuesta de desarrollo de GDs para sistemas de manufactura es, justamente, uno de los pilares para el desarrollo de los SATDD basados en el paradigma del AR. Como mencionamos anteriormente en la integración presentada en la Sección 5.1, la arquitectura propuesta puede admitir diferentes métodos de toma de decisiones y no se limita a uno exclusivo. En particular, en este trabajo se ha desarrollado un SATDD basado en el paradigma de AR con el fin de mostrar la versatilidad de la propuesta de incorporar un método de toma de decisiones basado en algoritmos provenientes de este tema candente en la comunidad de IA.

En la Figura 5.3 se muestra un caso que usa un sistema de toma de decisiones basado en un agente de AR, el cual interactúa con el GD de la hipotética planta de fabricación simple, definida anteriormente en el Capítulo 4. En este esquema de AR, un agente inteligente interactúa con su entorno, eligiendo acciones cuyo objetivo es maximizar la recompensa esperada. Puede distinguirse en la figura que el agente recibe información del nodo ROS principal, ejecuta el algoritmo y luego devuelve las soluciones a los diversos nodos que la requieran. Estos nodos, por su parte, pueden actualizar parámetros tanto de los robots físicos como de sus simuladores; como así también del ESPM. Esta retroalimentación se utilizará para los diversos problemas que surjan en el piso de planta o para cualquier oportunidad de mejora u optimización del proceso productivo.

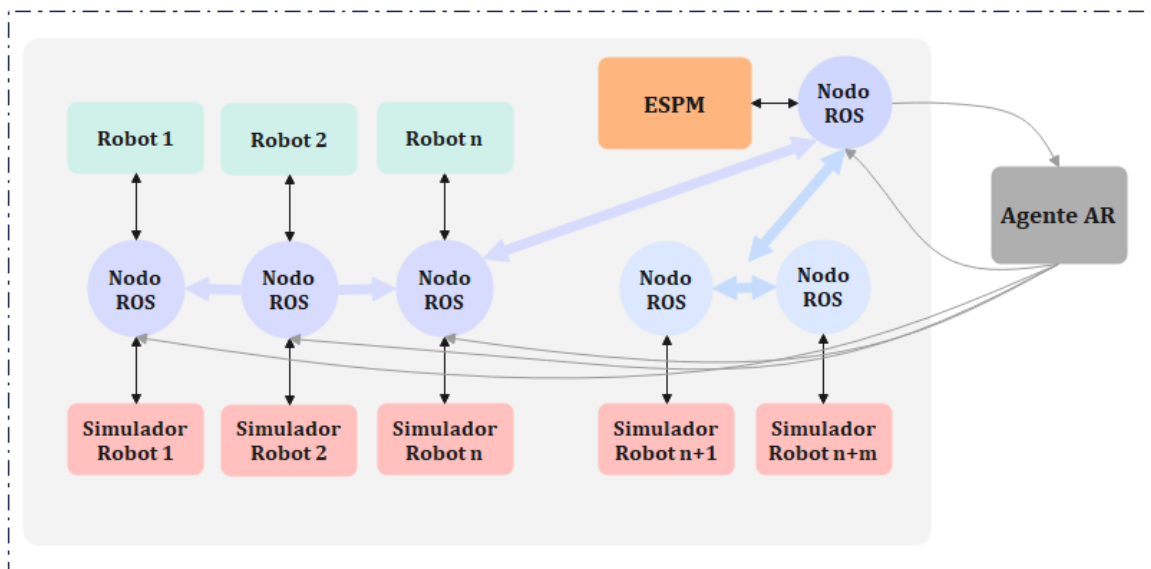


Figura 5.3: Sistema de TDD basado en un agente de AR utilizando el GD de una hipotética planta de fabricación simple.

Una consideración teórica es el hecho de que, para aplicaciones prácticas, los procesos de decisión de Semi-Markov son modelos para describir los problemas de toma de decisiones secuenciales en entornos estocásticos y tienen distribuciones de tiempo de estadía más generales y, por lo tanto, tienen aplicaciones más amplias en muchos sistemas prácticos. [Das et al., 1998, Paternina-Arboleda and Das,

2001, Paternina-Arboleda and Das, 2005]. Sin embargo, para el caso de nuestra aplicación práctica y sin perder generalidad, podemos asumir que este problema de toma de decisiones puede describirse perfectamente como un problema de decisión de Markov [Monahan, 1982] el cual está definido por cuatro elementos, a saber, el espacio de estados  $S$ , el espacio de acción  $A$ , la probabilidad de transición de estado  $P$  y la función de recompensa  $r_w(\cdot)$  (véase Sección 3.5.1).

Resumiendo, dentro de los muchos objetivos a lograr en una organización industrial, un SATDD puede utilizarse para maximizar la eficiencia productiva de las instalaciones ya existentes, minimizar el stock y los tiempos de producción, optimizar la utilización de los recursos, minimizar la inversión en futuros proyectos, mejorar distribuciones de planta y ubicación de instalaciones, entre muchos otros.

## 5.4 Prueba de concepto de la arquitectura en la Celda de manufactura

La Facultad de Ingeniería de la UNICEN dispone de una celda de manufactura equipada con diversos elementos industriales a escala de laboratorio, la cual se utilizó para llevar a cabo una prueba de concepto de la arquitectura desarrollada. Este entorno experimental permitió evaluar la viabilidad del sistema en un contexto realista, validando la integración entre los dispositivos físicos y el entorno de simulación.

Para este experimento, se seleccionaron dos elementos físicos de la celda y se estableció su vinculación con el simulador TPS a través de ROS, asegurando una interoperabilidad eficiente. La prueba tuvo como objetivo principal utilizar la arquitectura en un ambiente controlado y verificar la capacidad de la misma para lograr interconectar los distintos componentes del sistema. Los elementos considerados para este propósito son una cinta transportadora equipada con dos alimentadores, una mesa giratoria, un PLC Siemens y el simulador de planta. Como se ha explicado, todos estos elementos son integrados mediante nodos ROS. Las figuras 5.4 y 5.5 muestran distintas perspectivas de una zona de la celda de manufactura donde se pueden ver la cinta transportadora y la mesa giratoria, respectivamente. La Figura 5.6 muestra la instancia particular de la arquitectura de conexión entre la Cinta Transportadora, la Mesa Giratoria y el simulador de planta TPS.

En base a la arquitectura desarrollada, se implementó una clase derivada de Planta, denominada **Celda de Manufactura**, y se desarrolló un archivo de simulación en TPS que representa digitalmente la celda virtual. Además, se creó una clase derivada de Sistema Autónomo de TDD denominada **Método1**, la cual define un comportamiento básico que permite activar y desactivar los objetos simulados en el entorno virtual. Esta activación se ejecuta en respuesta a la detección de cambios en los objetos físicos a través de ROS, estableciendo así una comunicación bidireccional entre los dispositivos reales y sus representaciones digitales.

Este enfoque, dado por la arquitectura desarrollada, no solo permite replicar fielmente los estados y comportamientos de los elementos físicos en el entorno virtual, sino que también sienta las bases para el desarrollo de sistemas de manufactura más avanzados, donde la sincronización entre los mundos real y digital sea clave para la optimización de los procesos.



Figura 5.4: Celda de manufactura. Vista principal de cinta transportadora.



Figura 5.5: Celda de manufactura. Vista principal de cinta transportadora y mesa giratoria.

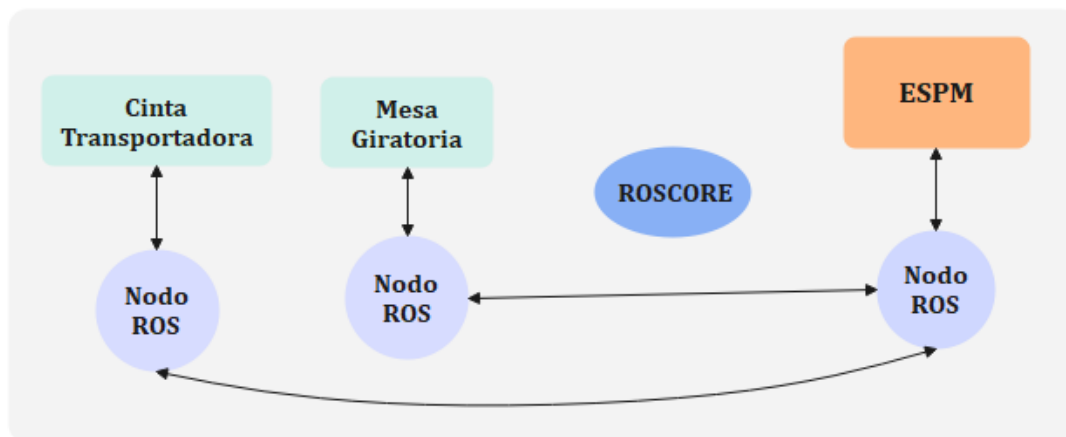


Figura 5.6: Arquitectura particular de conexiones en la celda de manufactura.

Tal lo explicado en la Sección 4.2, ROS consiste en un conjunto de procesos independientes, denominados nodos, los cuales se ejecutan de manera concurrente y establecen comunicación mediante el intercambio de mensajes a través de tópicos. Esta arquitectura modular permite la distribución de tareas entre múltiples componentes, facilitando la escalabilidad y el desarrollo de sistemas complejos en robótica.

En primera instancia, se llevó a cabo la inicialización del nodo *roscore*, el cual constituye el núcleo central de la infraestructura de ROS. Este nodo cumple un rol fundamental al encargarse de la gestión del sistema de mensajería, la asignación de nombres a los nodos y la coordinación del flujo de datos dentro del entorno de ejecución. Al actuar como un servidor maestro, *roscore* posibilita la detección e interacción entre los distintos nodos que conforman el sistema, permitiendo que estos publiquen y suscriban información en los tópicos correspondientes. A partir de su activación, los componentes del sistema pueden integrarse de manera estructurada, garantizando la interoperabilidad y el flujo eficiente de datos en tiempo real.

En la Figura 5.7, se pueden observar los tópicos utilizados en esta conexión específica. En este esquema, los elementos reales publican información en el tópico “*chatter real*”, mientras que el ESPM realiza sus publicaciones en el tópico “*chatter virtual*”. Como ejemplo de un caso de modelado e integración dentro del entorno de simulación, en el Apéndice del documento se presenta el código desarrollado en lenguaje Python utilizado para la creación del objeto virtual Cinta Transportadora y el ESPM, donde se ilustra el proceso de publicación y recepción de mensajes a través de estos tópicos, permitiendo la comunicación y sincronización entre el entorno físico y el virtual, además de diagramas UML de la integración.

Para establecer la conexión con el PLC Siemens <sup>2</sup> desde Python, se empleó la librería *snap7*, la cual permite la comunicación mediante el protocolo S7. En esta implementación, *snap7* genera un objeto virtual de tipo Cliente, que se encarga de conectarse al PLC real, el cual actúa como Servidor. Esta conexión facilita el intercambio de datos en tiempo real, permitiendo la sincronización del estado del objeto virtual con su contraparte física en la celda de manufactura.

Una vez desarrollada la conexión entre los elementos e implementada la

<sup>2</sup>Modelo SIMATIC S7 1200

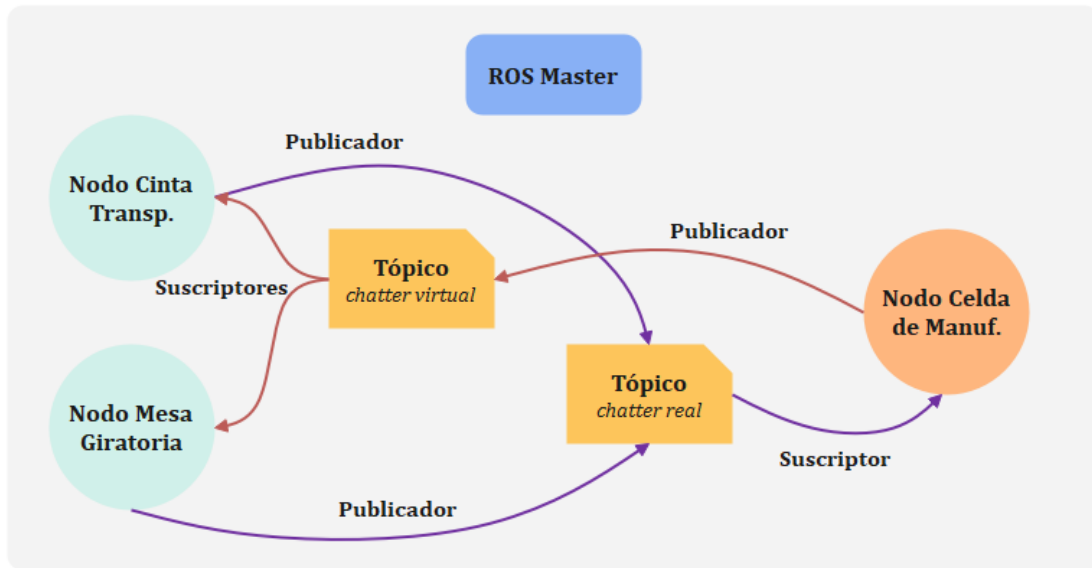


Figura 5.7: Nodos y tópicos utilizados en el experimento.

arquitectura, se validó lo expuesto con una métrica de desempeño. Para esto, se calculó la latencia en la sincronización entre los entornos físico y virtual. La latencia es el tiempo transcurrido entre un cambio de estado en el dispositivo físico (por ejemplo, la activación de la cinta transportadora) y su reflejo en el modelo simulado.

El objetivo de este experimento es medir y comparar las latencias de comunicación en ambas direcciones. Por lo tanto, se midieron las latencias en ambos flujos de comunicación: real a virtual, y virtual a real. Para ello, se incluyeron marcas de tiempo en los mensajes ROS intercambiados entre los nodos, calculando la diferencia temporal al momento de recepción. Luego, se almacenaron los datos y se graficaron para su posterior análisis. Para constatar los cambios de estado de los elementos en ambos mundos, aleatoriamente se seleccionaba cada cierto tiempo si los elementos estaban encendidos o apagados. Para considerar que el sistema es saludable, ambas latencias (real a virtual y virtual a real) deben ser bajas ( $<0.5$  segundos) y estables, lo que indica una comunicación óptima.

A continuación, en las figuras 5.8 y 5.9 se pueden apreciar las gráficas comparativas de resultados de las latencias para el sistema en dos experimentos realizados. En las Tablas 5.1 y 5.2 se resumen las medidas estadísticas de las mediciones. Se puede observar que las latencias se mantienen dentro de rangos aceptables para la sincronización de la celda real y su contraparte virtual (GD). En ambos casos, no se superan los 0.01 segundos de latencia. En conclusión, estos resultados validan la factibilidad del esquema de integración desarrollado para aplicaciones de manufactura flexible y permiten cuantificar el desempeño de la arquitectura implementada.

Flujo	Latencia promedio	Latencia mínima	Latencia máxima
Real $\rightarrow$ Virtual	0.003390 s	0.000997 s	0.007529 s
Virtual $\rightarrow$ Real	0.003302 s	0.001956 s	0.007002 s

Tabla 5.1: Estadísticas de latencias medidas para el Experimento 1.

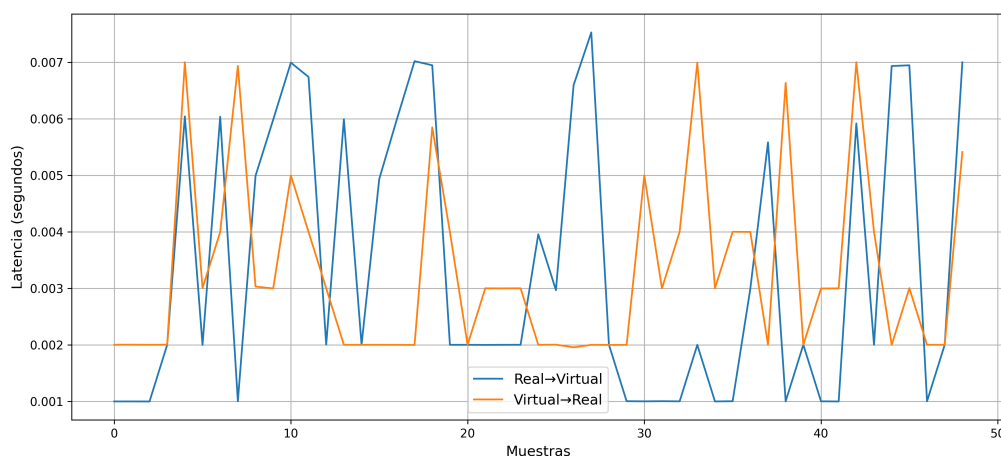


Figura 5.8: Comparación de latencias en ambos sentidos de comunicación. Experimento 1.

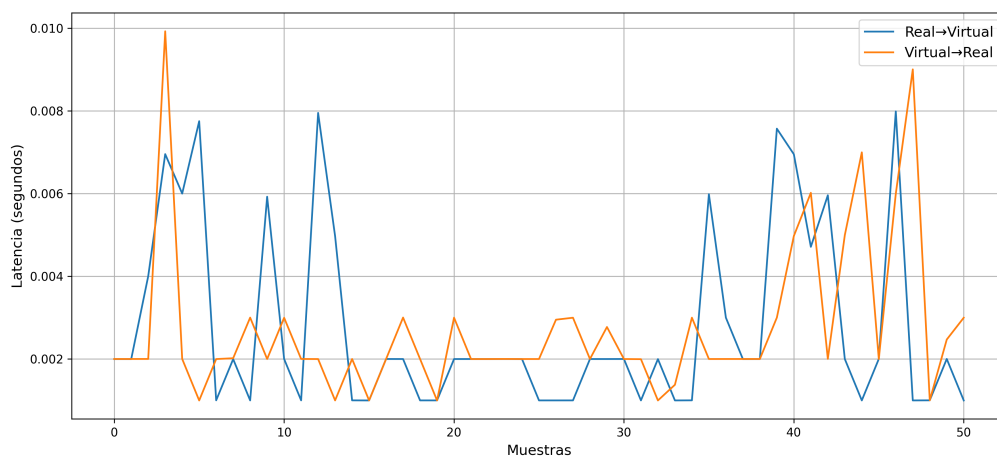


Figura 5.9: Comparación de latencias en ambos sentidos de comunicación. Experimento 2.

Flujo	Latencia promedio	Latencia mínima	Latencia máxima
Real → Virtual	0.002798 s	0.000999 s	0.007986 s
Virtual → Real	0.002735 s	0.000999 s	0.009922 s

Tabla 5.2: Estadísticas de latencias medidas para el Experimento 2.

## 5.5 Aproximación a la simulación semi-automática en el desarrollo de GDs

En este apartado se presenta un primer acercamiento metodológico a la simulación semi-automática, enmarcado dentro de los avances desarrollados durante la investigación doctoral. Este trabajo sienta las bases para un enfoque eficiente en la construcción rápida de modelos de simulación destinados a la implementación de GDs, con aplicaciones en entornos productivos, logísticos y de servicios. Una discusión más extensa sobre esta línea de investigación se abordará en el capítulo

final, en relación con futuras líneas de trabajo.

El término simulación semi-automática hace referencia al proceso de modelado de sistemas físicos mediante la utilización y combinación de diferentes técnicas. Por ejemplo, se podrían utilizar técnicas de reconocimiento de objetos por visión artificial, técnicas para identificación automática de componentes del sistema productivo, en combinación con técnicas de localización de objetos para reconstruir el layout de planta. Sin embargo, la aprobación final debería estar sujeta a la supervisión humana. A diferencia de los enfoques tradicionales, que requieren un modelado manual exhaustivo en entornos de simulación como TPS (o cualquier otro software equivalente), esta metodología busca reducir significativamente los tiempos de implementación y los costos asociados, manteniendo altos niveles de precisión y adaptabilidad [Tao et al., 2019]. Esto contribuye al objetivo de agilizar el desarrollo de los GDs.

En contextos industriales, donde los sistemas productivos están sujetos a modificaciones frecuentes, contar con capacidad de generar modelos de simulación de forma ágil y efectiva constituye una ventaja distintiva ya que permite aprovechar las bondades de la simulación para la formulación de sistemas de TDD autónomos. La propuesta aquí presentada se centra en realizar una formulación para automatizar dos etapas fundamentales del proceso de creación de un modelo de simulación: el relevamiento del sistema y el modelado computacional (véase Sección 5.5.2). Cada una de estas etapas incorpora estrategias para minimizar la intervención manual sin comprometer la fiabilidad del modelo resultante.

### 5.5.1 Relevamiento del sistema físico

El relevamiento de datos constituye la fase inicial y crítica en el desarrollo de cualquier modelo de simulación, ya que determina la calidad, precisión y exhaustividad de los datos empleados en su construcción. Tradicionalmente, este proceso se ha realizado de manera manual, especialmente en empresas con bajo nivel de digitalización, a través de entrevistas, observación directa y recopilación documental. No obstante, en organizaciones con mayor madurez tecnológica, es posible adoptar estrategias semiautomáticas que optimicen esta etapa, permitiendo reducir errores y mejorar la eficiencia del proceso.

Independientemente del nivel de automatización, el relevamiento de datos tiene como objetivo recolectar información clave sobre el sistema en estudio para modelarlo con la mayor fidelidad posible. En un entorno productivo, esto implica la recopilación de parámetros como tiempos de proceso, tasas de fallas y modos de fallo, niveles de inventario, porcentaje de productos defectuosos, desempeño de los operarios, entre otros.

Entre los enfoques avanzados utilizados en esta etapa podemos mencionar los sistemas de IoT, que mediante sensores embebidos permiten obtener datos en tiempo real sobre tiempos de ciclo, tasas de fallas, niveles de inventario y condiciones operativas de las maquinarias. Además, los sistemas de gestión automatizados (tales como sistemas Enterprise Resource Planning (ERP) y Manufacturing Execution System (MES)) proporcionan registros estructurados sobre la operación del sistema, facilitando la extracción de datos clave sin necesidad de intervención manual. Por otro lado, y aumentando la complejidad, las tecnologías de visión artificial y procesamiento de imágenes actuales posibilitan el monitoreo de líneas de ensamblaje

y la detección de anomalías sin necesidad de una medición manual, mejorando la precisión en el control de calidad de productos y procesos. Finalmente, la conexión con bases de datos históricas permite acelerar la obtención de parámetros relevantes, como tasas de defectos o disponibilidad de equipos, proporcionando una base sólida para el modelado del sistema.

El resultado de esta fase es un conjunto de datos estructurados y validados, generalmente organizados en tablas o bases de datos, que servirán como insumo para la siguiente etapa. La Figura 5.10 esquematiza conceptualmente la fase de relevamiento de datos e información.

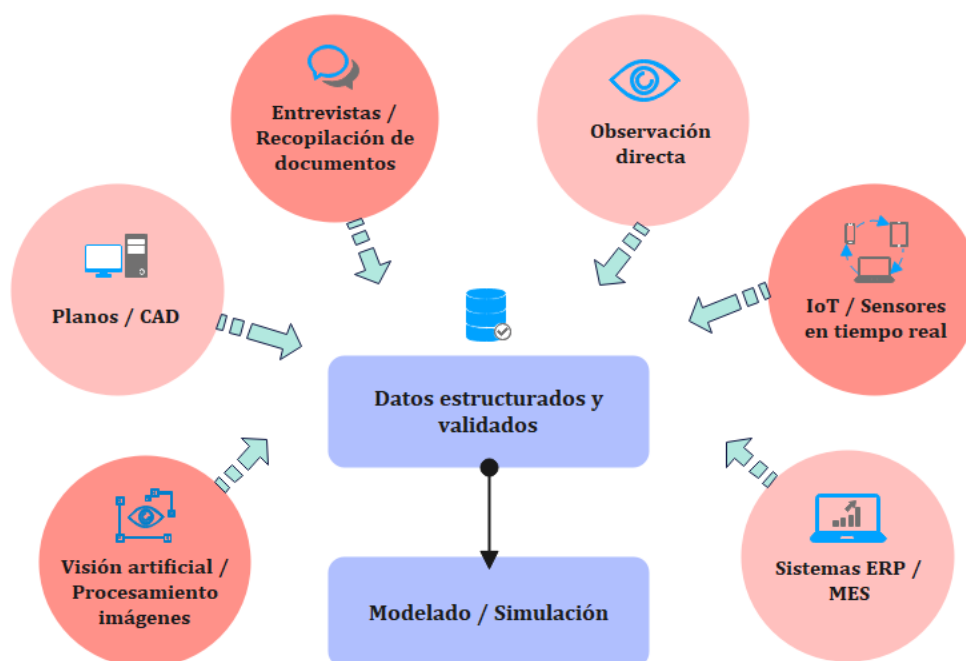


Figura 5.10: Esquema conceptual para un relevamiento de un sistema físico.

### 5.5.2 Modelado computacional

Una vez completado el relevamiento, comienza la etapa de modelado computacional que se enfoca en traducir los datos recopilados en un entorno de simulación ejecutable, en nuestro caso será TPS. Aquí, la intervención manual tradicional, que implica la creación de objetos, la configuración de parámetros y la programación de lógicas de comportamiento, puede optimizarse mediante técnicas de automatización parcial.

Una primera aproximación en este sentido es reducir tiempos en la creación de simulaciones mediante el uso de código reutilizable, o mediante la estandarización del proceso. También, la incorporación de automatización parcial, para reducir la intervención humana en la simulación. Esto podría ser de mucha ayuda en casos donde las líneas de producción sufren constantes modificaciones. Si bien la reutilización de código para múltiples instancias y el desarrollo de una metodología

adaptable requieren mayor trabajo en la etapa de diseño, luego implicaría una reducción del tiempo en próximos casos de estudio.

En este sentido, en esta tesis se desarrolló un código Python que permite, haciendo uso de la arquitectura desarrollada en la Sección 5.1, la creación automática del modelo en TPS (archivo *.spp*) mediante el uso de funciones nativas del simulador y la integración con datos estructurados en Excel. Se inicializa a partir de un modelo genérico, y en base a datos del layout de la planta y características de los objetos a simular que se encuentran en la hoja de cálculo, se automatiza una primera versión del sistema simulado. De esta manera, el usuario solo debe completar la planilla necesaria en vez de perder tiempo colocando objeto por objeto en la simulación y seteando sus parámetros en cada ventana. Una vez creado el modelo y guardado el archivo, el mismo se puede seguir personalizando para mayor detalle.

Este enfoque reduce errores humanos en la transcripción de parámetros, acelera la creación de modelos base (dejando la personalización avanzada para fases posteriores) y facilita la escalabilidad, al permitir la replicación del método en múltiples casos de estudio.

Un formato posible de plantilla es el mostrado en la Tabla 5.3. Se observan solo algunas características de los objetos. Obviamente, no todos los objetos poseen los mismos atributos y no es necesario completar todas las columnas (eso dependerá de cada objeto en particular). Sí será necesario marcar las conexiones entre objetos para que el modelo se genere ya conectado.

ID del equipo	Tipo de equipo	X Pos	Y Pos	Tiempo de procesamiento (s)	Tiempo de set-up (s)	Capacidad (u)
SO	Source	150	100			
S1	Source	150	600			
A1	Station	250	150	20	400	
A2	Station	250	250	25	400	
A3	Station	250	350	40	400	
A4	Station	250	450	40	400	
A5	Station	250	600	20	200	
B1	Buffer	350	150	5		10
B2	Buffer	350	250	5		20
B3	Sorter	350	350			30
B4	Sorter	350	600			40
D1	Drain	500	500	10		

Tabla 5.3: Tipo de información a completar en la hoja de cálculo.

A continuación, podemos ver las interfaces de TPS en dos momentos distintos: el modelo genérico (Figura 5.11) y el modelo generado a partir del ejemplo de la tabla (Figura 5.12). Los códigos del ejemplo se encuentran disponibles en el repositorio GitHub ya mencionado.

Es importante resaltar que, aunque el ejemplo presentado es de escala reducida y la integración realizada es relativamente sencilla, constituye un punto de partida significativo en la exploración del uso de documentos disponibles en la empresa

como fuente de datos para la generación de modelos de simulación. La capacidad de aprovechar información preexistente en la organización no solo optimiza el proceso de modelado inicial, sino que también mejora la precisión y representatividad del sistema simulado, facilitando su alineación con la realidad operativa.

No obstante, para potenciar el alcance y la aplicabilidad de esta metodología, resulta fundamental continuar investigando e integrando una mayor variedad de formatos y fuentes de información. La incorporación de archivos multimedia como

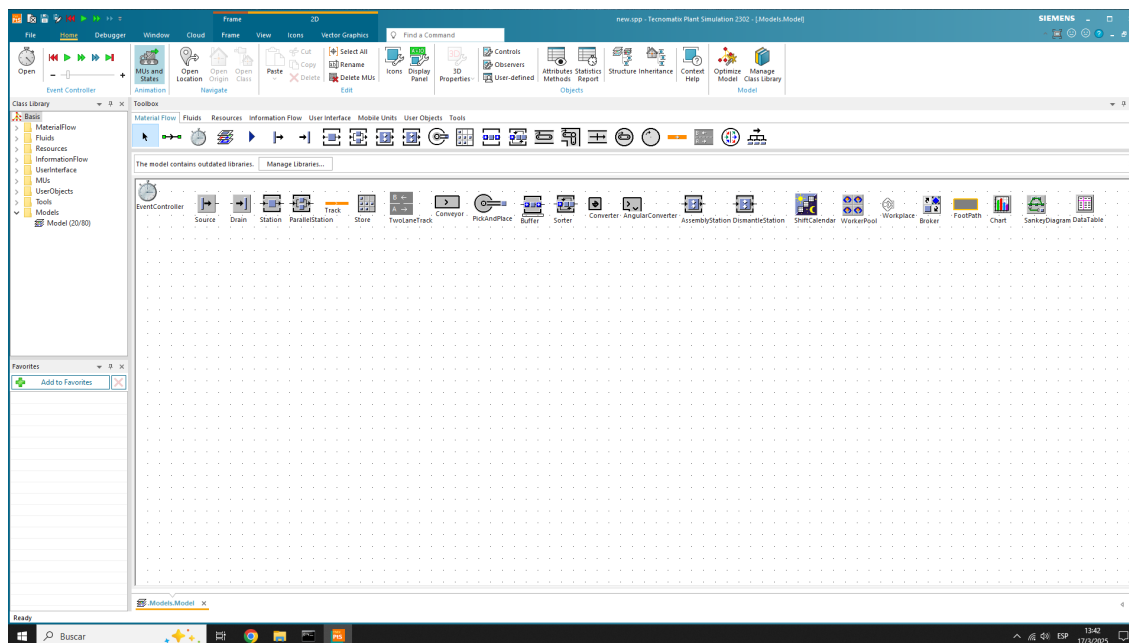


Figura 5.11: Modelo genérico para la simulación semi-automática.

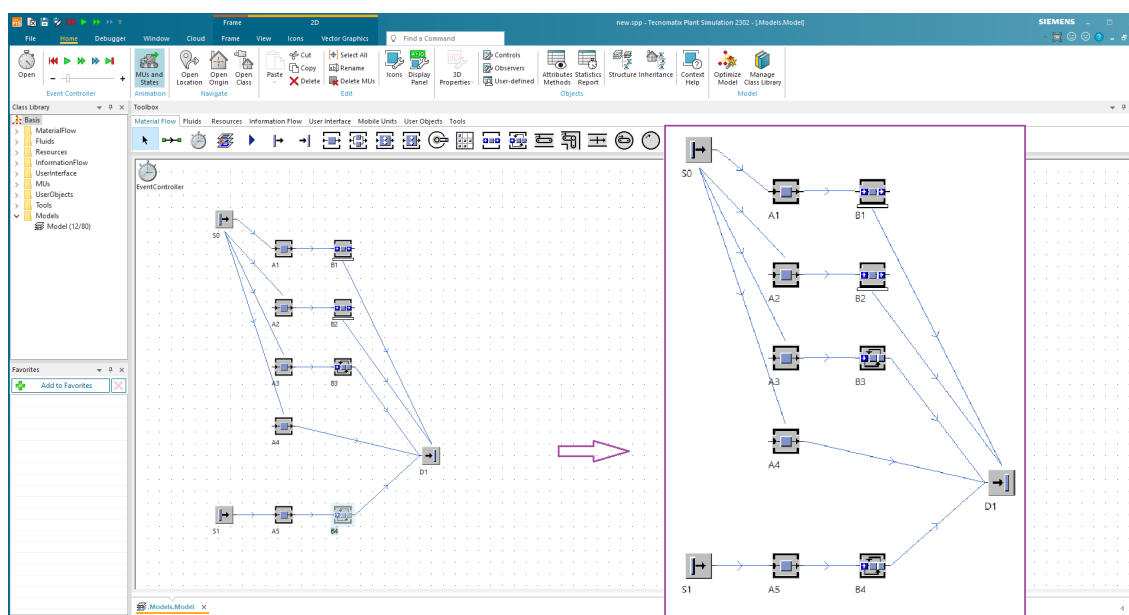


Figura 5.12: Modelo generado a partir de la Tabla 5.3.

videos e imágenes, así como documentos técnicos en formatos como planos, archivos .pdf y modelos CAD, permitiría enriquecer la construcción del GD y mejorar la

parametrización del entorno simulado. Esta evolución contribuiría a una mayor automatización del proceso de modelado, reduciendo el tiempo y los recursos necesarios para su implementación y aumentando la accesibilidad de las herramientas de simulación para distintos sectores industriales.

En este sentido, el desarrollo de estrategias avanzadas de procesamiento de datos, reconocimiento de patrones y extracción de información de documentos heterogéneos representa un desafío clave para futuras investigaciones. La combinación de estas técnicas con IA y aprendizaje automático podría favorecer la creación de modelos cada vez más dinámicos y adaptativos, consolidando el papel de la simulación como una herramienta esencial para la toma de decisiones en entornos industriales complejos y altamente cambiantes.

## 5.6 Conclusiones del capítulo

En el presente capítulo se ha expuesto el desarrollo de una arquitectura software que combina ROS y un entorno de simulación junto con un SATDD para sistemas de manufactura flexible. Esta integración permite una comunicación eficiente entre modelos digitales y dispositivos físicos, facilitando la implementación de GDs en entornos industriales con altos requerimientos de automatización y adaptabilidad.

Se explicó en detalle esta arquitectura y se realizó una prueba de concepto de la misma en la celda de manufactura disponible en el lugar de trabajo. Esta prueba permitió validar el enfoque propuesto, así como evaluar su capacidad para representar desde agentes físicos (como la mesa y la cinta) hasta agentes abstractos (como el sistema de TDD) e integrarlos en un modelo completo del sistema en un entorno de simulación. Además, se destacó la importancia del uso de ROS como infraestructura de comunicación, proporcionando una base sólida para la integración de sensores, actuadores y modelos digitales en un ecosistema de manufactura inteligente.

En los dos capítulos siguientes se aplicará este desarrollo a dos casos de estudio relacionados con el manejo de materiales en piso de planta. Estos casos permitirán evaluar la adaptabilidad de la arquitectura propuesta en escenarios industriales específicos, demostrando su potencial para optimizar procesos y mejorar la eficiencia operativa mediante la incorporación de GDs.

Por último, se realizó una primera aproximación al tema de la simulación semi-automática para el desarrollo de los gemelos, buscando disminuir el tiempo que conlleva el modelado de los sistemas industriales. Este enfoque representa un paso fundamental hacia la reducción de costos y tiempos de implementación, promoviendo el uso de técnicas avanzadas de simulación y automatización en el diseño y mantenimiento de sistemas de manufactura flexible.



# Capítulo 6

## Aplicación del SATDD para el manejo de materiales en piso de planta

En los sistemas de manufactura flexible, el manejo de materiales representa un desafío complejo que impacta directamente en la eficiencia y productividad del sistema de fabricación, como se ha mencionado en el Capítulo 2. La correcta gestión del flujo de materiales es esencial para garantizar que las materias primas, los componentes y los productos semielaborados se transporten de manera óptima a lo largo de la línea de producción, minimizando tiempos de espera y reduciendo desperdicios. Este problema se vuelve aún más crítico en entornos de manufactura donde la producción es altamente variable y debe adaptarse rápidamente tanto a cambios en la demanda como a configuraciones de productos, así como a la configuración del sistema productivo.

Uno de los principales retos en la gestión de materiales dentro de estos sistemas es la sincronización entre los procesos productivos y los sistemas de transporte interno, como transportadores automatizados, vehículos de guiado autónomo (AGV) y robots móviles. La ineficiencia en este aspecto puede generar cuellos de botella, acumulación innecesaria de inventario en proceso y retrasos en la entrega de productos terminados. Por lo tanto, mejorar la eficiencia de los procesos complejos de fabricación, en particular la logística y el manejo de materiales, es un problema difícil y que ha recibido especial atención, principalmente para la planificación de tareas en esquemas de producción tradicionales. Sin embargo, no hay demasiada evidencia para esquemas de producción flexibles, cuando cambian, por ejemplo, algunas de las condiciones de los pisos de producción.

La combinación de tecnologías 4.0 con metodologías de producción Lean, que buscan eliminar desperdicios y mejorar la eficiencia operativa, ha demostrado ser una opción efectiva para optimizar la logística interna en sistemas de manufactura flexible. En este contexto, la automatización inteligente del manejo de materiales es un factor clave para mejorar la competitividad de las empresas manufactureras, ya que minimiza las operaciones sin valor agregado y aumenta la capacidad de respuesta ante las exigencias del mercado.

En este capítulo, se presenta una propuesta de sistema de TDD para la gestión del manejo de materiales en un sistema de manufactura. La creación del modelo digital en el software TPS y su integración con ROS para el desarrollo de los

correspondientes GDs se realizó siguiendo los pasos descritos en las secciones anteriores. Finalmente, se presenta una propuesta de integración con un sistema de toma de decisiones que emplea un agente artificial de AR. En el problema planteado se buscará lograr varios objetivos, como minimizar las distancias recorridas, el uso de los medios de transporte y las demoras en las entregas que interrumpan al sistema productivo.

## 6.1 Caso de Estudio

Particularmente, para el análisis se utiliza y adapta el caso de estudio presentado en el trabajo de [Govindaiah and Pey, 2019], el cual muestra un problema complejo y realista de manejo de materiales en un proceso manufacturero. El objetivo es generar planes de manejo de materiales más eficientes que respondan a una demanda variable en una instalación de producción representativa.

### 6.1.1 Definición del problema

El manejo de materiales en la planta implica la asignación y transporte de partes entre estaciones de trabajo utilizando diversos tipos de equipos (carretillas elevadoras, transportadores autónomos, etc.). Debido a la complejidad del sistema y el número de variables involucradas, encontrar estrategias óptimas de transporte resulta difícil mediante métodos tradicionales. Para abordar este problema, se desarrolló un modelo basado en AR, el cual aprende a mejorar la planificación del transporte de materiales a lo largo del tiempo, respecto a determinadas variables.

En este caso, el sistema bajo análisis está compuesto por diez estaciones de trabajo ( $S_i \forall i = 1, 2, \dots, 10$ ) que consumen simultáneamente dos tipos de partes distintas ( $P_1, P_2$ ). En cada estación de trabajo existen *buffers* de almacenamiento para cada parte ( $B_k \forall k = 1, 2, \dots, 20$ ), cuyas capacidades son iguales al doble de la cantidad de partes de cada tipo consumidas en una hora. Hay nueve vehículos de tres tipos diferentes ( $V_{je} \forall j \in \{1, 2, 3\} \vee e \in \{1, 2, 3\}$ ) y se desarrollan distintos planes de manejo de materiales preestablecidos. Cada plan de manejo de materiales tiene asociada una ruta a seguir ( $W_c$ ), un transporte específico ( $T_d$ ), además de partes y cantidades de partes a llevar a las estaciones. Los equipos de transporte no son todos iguales, es decir, cada tipo de vehículo,  $V_{je}$ , tiene su propia capacidad de carga y puede llevar cualquier tipo de parte.

### 6.1.2 Modelo conceptual

Para una comprensión más detallada del sistema correspondiente al caso de estudio, se desarrolla el diagrama IDEF-SIM del mismo (Figura 6.1). Como se explicó en un capítulo anterior, este diagrama constituye una herramienta fundamental para el modelado conceptual, previo a la simulación computacional, permitiendo una representación estructurada de las principales entidades, funciones, recursos y relaciones que integran el sistema productivo.

En la Figura 6.1, se pueden identificar claramente las estaciones de trabajo (rectángulos celestes S1 a S10) que conforman el sistema, cada una de ellas acompañada de sus respectivos *buffers* de materiales (rectángulos verdes B1 a B20),

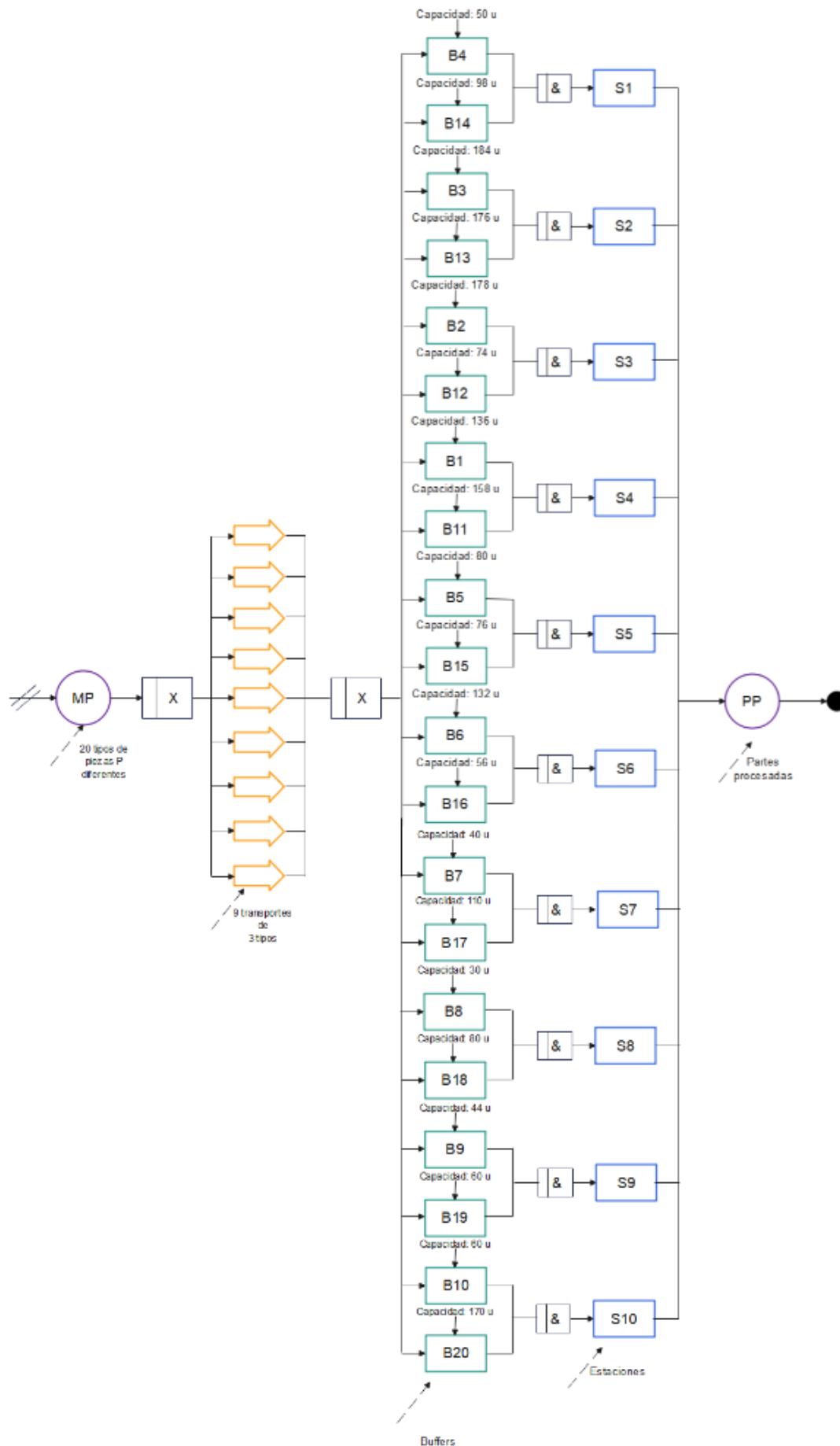


Figura 6.1: IDEF-SIM del primer caso de estudio.

los cuales desempeñan un papel clave en la gestión de los flujos productivos. Además, se representan los medios de transporte utilizados para la movilidad de insumos y productos dentro del sistema, así como la información operativa esencial que debe considerarse en la construcción del modelo digital, como las capacidades de los *buffers*.

El diagrama IDEF-SIM permite establecer una base sólida para la simulación, proporcionando una visión estructurada de los procesos y sus interdependencias. Al definir de manera explícita los recursos disponibles y su interacción con las entidades del sistema, este enfoque facilita la identificación de cuellos de botella, restricciones operativas y posibles estrategias de optimización. Asimismo, el uso de esta metodología mejora la trazabilidad del modelo, garantizando que la transición desde el modelado conceptual hasta la implementación digital en el software de simulación se realice de manera consistente y precisa. Esto permite reducir el margen de error en la fase de experimentación, optimizando la eficiencia y confiabilidad de los resultados obtenidos en la simulación computacional.

### 6.1.3 Modelo digital

Una vez desarrollado el modelo conceptual y con la información disponible del caso de estudio, se procede a la digitalización del sistema en el software de simulación seleccionado. Este proceso implica la parametrización y configuración detallada de cada componente del sistema productivo, asegurando una representación fiel de la realidad operativa. Para ello, se emplean las diversas librerías del simulador y se personalizan los objetos de acuerdo con las características específicas del caso de estudio. La Figura 6.2 muestra una captura del modelo desarrollado en el software TPS, donde se evidencia la disposición de los distintos elementos y su interacción dentro del sistema.

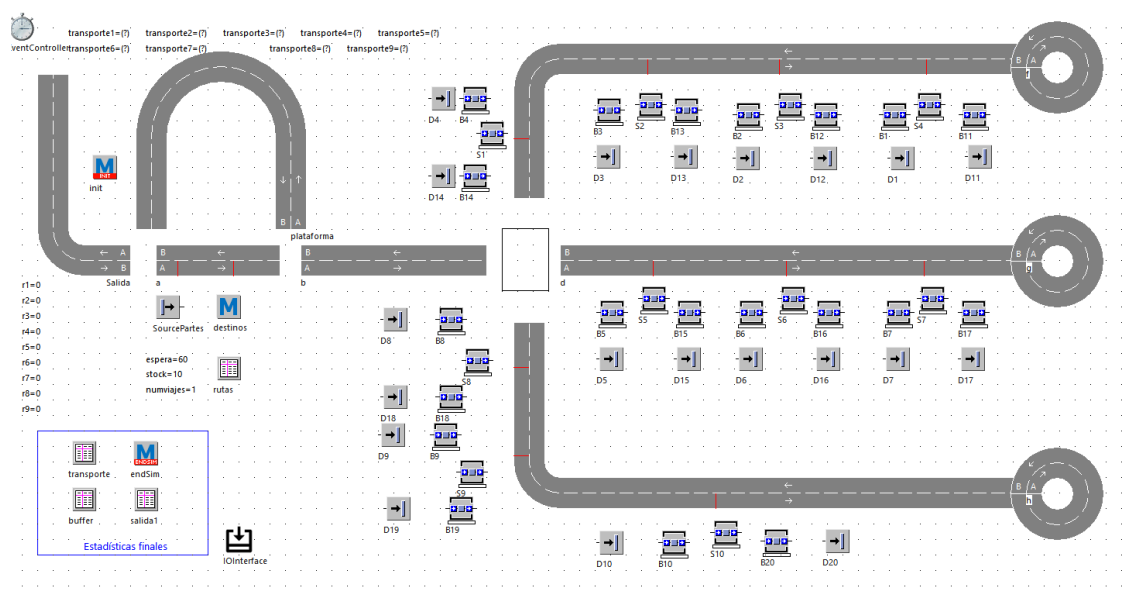


Figura 6.2: Modelo de simulación del primer caso de estudio.

En este modelo digitalizado, se encuentran representadas todas las entidades que conforman el sistema en estudio, incluyendo estaciones de trabajo, equipos de

transporte, zonas de almacenamiento y los flujos de materiales. Además, se han configurado las vías de circulación con sus respectivos sentidos de tránsito, los puntos de carga y descarga de materiales, así como los tiempos y reglas de operación de cada elemento. La modelación de estos aspectos es crucial para garantizar que la simulación refleje de manera precisa las dinámicas reales del sistema productivo.

Para optimizar la ejecución del modelo y permitir un análisis detallado de su desempeño, se incorporan tablas de datos y métodos propios del simulador TPS, los cuales facilitan la recopilación y procesamiento de información en tiempo real. Estos elementos permiten la programación del funcionamiento del sistema, el monitoreo de variables clave y la recopilación de estadísticas relevantes al finalizar cada corrida de simulación. Dichas estadísticas incluyen tiempos de ciclo, utilización de recursos, eficiencia en el manejo de materiales y niveles de ocupación de las estaciones de trabajo.

El modelo digital desarrollado no solo permite la validación del diseño conceptual, sino que también proporciona una herramienta analítica poderosa para evaluar distintos escenarios y estrategias de optimización. Mediante la experimentación con diferentes configuraciones y parámetros, es posible identificar oportunidades de mejora, minimizar cuellos de botella y aumentar la eficiencia operativa. Este enfoque basado en simulación resulta fundamental para la toma de decisiones informadas en la planificación y gestión de sistemas productivos <sup>1</sup>.

## 6.2 Integración digital y SATDD para el manejo de materiales

El modelo desarrollado en el simulador TPS queda definido mediante un conjunto de parámetros. Haciendo uso de la arquitectura propuesta (Sección 5.1), podemos acceder externamente, mediante el nodo ROS creado, a los datos necesarios para el análisis posterior. De esta manera, los resultados finales de una corrida de simulación son la entrada para alimentar a un sistema de TDD.

Para poder evaluar y eventualmente cambiar los planes de acción, se recolecta determinada información del sistema en estudio. Por ejemplo, las distancias recorridas por cada medio de transporte y el nivel de ocupación de los mismos. También, el nivel máximo de llenado de los buffers respecto a la capacidad definida, y la porción del tiempo en que se llenan completamente estos buffers. Por último, los porcentajes de trabajo y de espera en cada una de las salidas del sistema.

A continuación se presentan los elementos para la formulación del problema de AR, tal lo detallado en la Sección 3.5.1. A menos que se diga lo contrario, se omite el subíndice  $t$  de representación del tiempo a efectos de simplificar la notación.

### Estado del sistema:

Para la representación del estado del sistema, se consideran las magnitudes de tres de las variables que impactan en la planificación del manejo de materiales, a saber: el tiempo de espera al cargar los transportes ( $T_i$ ), el stock inicial de las estaciones ( $s_0$ ) y la cantidad de viajes por vehículos que pueden realizarse ( $M$ ), mientras que los demás parámetros del sistema se mantienen constantes. Estas

<sup>1</sup>[https://github.com/INTELYMEC/ROS\\_Tecnomatix](https://github.com/INTELYMEC/ROS_Tecnomatix)

variables han sido seleccionadas debido a su impacto significativo en la eficiencia operativa y la dinámica del flujo de materiales dentro del sistema productivo.

De esta manera, el vector de estado del sistema se define como  $s_t = (T_l, s_0, M)$ , encapsulando la información esencial para la toma de decisiones en cada instante de tiempo. Para la discretización del estado del sistema, se establece que el tiempo  $T_l$  se segmenta en intervalos de 10 segundos dentro del rango [60, 300] segundos. El stock inicial  $s_0$  puede adoptar valores entre [10, 50], con incrementos de 10 unidades, mientras que la cantidad de viajes por vehículo se restringe a valores enteros dentro del intervalo [1, 5]. Esta discretización permite reducir la complejidad del problema sin comprometer la representatividad de los estados posibles del sistema.

El espacio de estados  $S$  se define como el producto cartesiano de los subconjuntos que representan los valores posibles de cada componente del vector de estados  $\mathbf{s}$ , tal que  $\mathbf{s} \in \mathbf{S}$ . En consecuencia, la cardinalidad del espacio de estados es  $|S| = 625$ , lo que implica que el agente debe evaluar y seleccionar entre una amplia gama de combinaciones posibles.

Dado el conjunto de estados posibles, el objetivo del agente es encontrar la combinación óptima de  $T_l$ ,  $s_0$  y  $M$  para cualquier estado del sistema  $\mathbf{s} \in \mathbf{S}$ . En otras palabras, debe determinar la transición hacia el estado óptimo del sistema que maximice la recompensa esperada, de acuerdo con la función de valor óptimo definida en la Ecuación 3.8. Esta formulación permite modelar el problema como un proceso de decisión en etapas, en el cual la selección de estrategias adecuadas resulta fundamental para la optimización del rendimiento del sistema de manejo de materiales.

### **Acciones:**

Considerando que el agente debe identificar la combinación óptima de las variables de decisión y teniendo en cuenta que el estado del sistema puede transitar libremente entre cualquiera de los 625 estados posibles, se establece que el espacio de acciones  $A$  es equivalente al espacio de estados. Esto implica que el conjunto de acciones disponibles para el agente está compuesto por 625 opciones distintas, lo que le permite seleccionar como siguiente estado tanto el mismo en el que se encuentra actualmente, como cualquiera de los 624 estados restantes. Es decir, las acciones consisten en establecer las transiciones de estados.

Esta formulación proporciona al agente un alto grado de flexibilidad en la toma de decisiones, dado que no se imponen restricciones sobre las transiciones entre estados. Así, el agente puede explorar de manera amplia el espacio de búsqueda, evaluando múltiples combinaciones de variables con el objetivo de optimizar la función de recompensa. Además, la posibilidad de elegir permanecer en el estado actual permite que el algoritmo de AR pueda estabilizar soluciones óptimas sin necesidad de generar cambios innecesarios en la configuración del sistema.

La definición del espacio de acciones de esta manera resulta fundamental para la convergencia eficiente del proceso de aprendizaje, ya que facilita la identificación de políticas óptimas sin imponer restricciones artificiales en la evolución del sistema. En este sentido, la estructura del espacio de decisiones del agente maximiza su capacidad de adaptación a distintas condiciones operativas, permitiendo mejorar progresivamente el desempeño del sistema en función de la recompensa acumulada.

**Función de recompensa:**

En este problema, el objetivo es optimizar el rendimiento del sistema considerando el grado de movimientos, el uso de *buffers* y los tiempos de espera de las estaciones. Para ello, se define una función de pérdida que permite cuantificar el impacto de estos factores en el desempeño global del sistema. La función de pérdida se expresa como:

$$\mathcal{L}(\mathbf{a}|\mathbf{s}) = C_1w_1 + C_2w_2 + C_3w_3 \quad (6.1)$$

donde  $C_1$  representa el porcentaje de ocupación de los transportes,  $C_2$  denota la fracción de tiempo en la que los *buffers* alcanzan su capacidad máxima, y  $C_3$  corresponde al porcentaje del tiempo total de espera para consumir las piezas. Para ponderar adecuadamente estos factores, se asignan pesos específicos a cada uno de ellos:  $w_1 = 0,3$ ,  $w_2 = 0,3$  y  $w_3 = 0,4$ . Esta distribución otorga una mayor relevancia al retraso en la recepción de las piezas, dado que este fenómeno puede interrumpir el flujo de trabajo del sistema y generar ineficiencias operativas significativas.

A partir de esta formulación, se define la función de recompensa instantánea como:

$$r(\mathbf{s}') = \frac{1}{\mathcal{L}(\mathbf{a}|\mathbf{s})}. \quad (6.2)$$

Dado que la recompensa es inversamente proporcional a la función de pérdida, los algoritmos de AR deben enfocarse en maximizar la recompensa esperada, lo que equivale a minimizar los costos operativos asociados con los transportes, los *buffers* y los tiempos de espera. Esta estrategia de optimización garantiza una gestión eficiente de los recursos del sistema, promoviendo un equilibrio entre los distintos factores que inciden en su rendimiento.

La formulación de esta función de recompensa permite que el proceso de converja hacia políticas óptimas de toma de decisiones, asegurando que las estrategias de manejo de materiales evolucionen de manera progresiva hacia configuraciones más eficientes y adaptativas. En este sentido, la estructura de la función de pérdida y su correspondiente función de recompensa constituyen elementos fundamentales para guiar el proceso de optimización dentro del marco del aprendizaje automático aplicado a la manufactura inteligente.

## 6.3 Resultados

Para demostrar la capacidad de integración y versatilidad de la arquitectura propuesta, se implementaron tres algoritmos diferentes basados en AR para la toma de decisiones en el módulo del SATDD, los cuales se detallan en las siguientes subsecciones. Además, se realizó una comparación de desempeño entre ellos. Las formulaciones de los algoritmos se implementaron en lenguaje Python, incluidas bibliotecas específicas (numpy, Pytorch, matplotlib y otras). Los códigos fuente están disponibles en el repositorio de GitHub presentado anteriormente.

### 6.3.1 Q-learning

Inicialmente, en el módulo del SATDD de la arquitectura propuesta se integró un agente artificial basado en el conocido algoritmo *Q-learning* [Watkins and Dayan,

1992]. Este algoritmo es un algoritmo de control sin modelo basado en métodos de diferencias temporales con respecto a la función de valor de estado-acción ( $Q$ ). El aprendizaje de diferencia temporal es una combinación de ideas de programación dinámica y método Monte Carlo. Este aprendizaje puede funcionar directamente de la experiencia, sin un modelo de la dinámica del entorno. Además actualiza las estimaciones basándose en parte en otras estimaciones aprendidas, sin esperar un resultado final (bootstrapping) [Sutton and Barto, 1998].

$Q$ -learning es considerado uno de los avances más importantes y tempranos en el campo del AR. Utiliza la función de valor de estado-acción para encontrar directamente la política óptima  $\pi^*$  (Ecuación 3.10). Se ha demostrado que  $Q$ -learning converge con una probabilidad de 1 a la política óptima. Su pseudocódigo se describe a continuación en el Algoritmo 3. La ecuación fundamental puede verse en la línea 7 del mismo. En dicha ecuación encontramos dos hiperparámetros a tener en cuenta: el factor de descuento  $\gamma$  ya mencionado anteriormente en el Capítulo 3 y la tasa de aprendizaje  $\alpha$ , que influye en la velocidad de convergencia del algoritmo.

---

**Algorithm 3**  $Q$  - learning
 

---

**Require:** tamaño de paso  $\alpha \in (0, 1]$ ,  $\epsilon > 0$

```

1: Inicializar  $Q(s,a)$  para todo  $s \in S$ ,  $a \in A$ 
2: for episodio=0 to  $i$  do
3:   Inicializar  $S$ 
4:   for paso=0 to  $j$  do
5:     Elegir  $A$  de  $S$  utilizando políticas derivadas de  $Q$  ( $\epsilon$  - greedy)
6:     Tomar acción  $A$ , observar  $R$ ,  $S'$ 
7:      $Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma \max_a Q(S', a) - Q(S, A)]$ 
8:      $S \leftarrow S'$ 
9:   end for
10: end for

```

---

Para ajustar el algoritmo, se buscaron y seleccionaron los hiperparámetros correspondientes. Para ello, se llevaron a cabo exploraciones repetidas utilizando la conocida metodología de búsqueda en cuadrícula [Yang and Shami, 2020]. Esta metodología implica la combinación de varios hiperparámetros y la evaluación del desempeño de cada combinación. Así, el espacio de búsqueda estará formado por todas las combinaciones posibles y se seleccionará aquella que genere mejores resultados. Los valores de los hiperparámetros utilizados en el procedimiento de optimización de todos los algoritmos desarrollados se muestran en la Tabla 6.1.

Factor de descuento $\gamma$	0.1	0.5	0.99	-	-	-
Tasa de aprendizaje $\alpha$	0.0001	0.001	0.01	0.1	0.5	0.9

Tabla 6.1: Parámetros probados en los algoritmos  $Q$ -learning, SARSA y DQN.

El algoritmo se ejecutó durante 500 episodios de 100 pasos cada uno. El factor de exploración,  $\epsilon$ , disminuye exponencialmente desde un valor inicial  $\epsilon = 1$  hasta un valor final  $\epsilon = 0,1$  para el número total de episodios analizados. Así, inicialmente, el algoritmo tiene un comportamiento puramente exploratorio y, a medida que transcurren los episodios de entrenamiento, comienza a prevalecer la explotación del conocimiento aprendido.

El valor de la tasa de aprendizaje que proporcionó los mejores resultados para la estrategia de búsqueda en cuadrícula del algoritmo *Q-learning* es  $\alpha=0,9$ ; mientras que el factor de descuento es  $\gamma=0,5$ .

Los resultados obtenidos del algoritmo *Q-learning* se observan en la Figura 6.3 que muestra la recompensa obtenida por el agente durante 500 episodios de entrenamiento. Como se puede observar, el agente aprende a mejorar su política de actuación, lo que se refleja en un aumento al alza del valor de la recompensa, que se estabiliza a partir, aproximadamente, del episodio 300 en adelante.

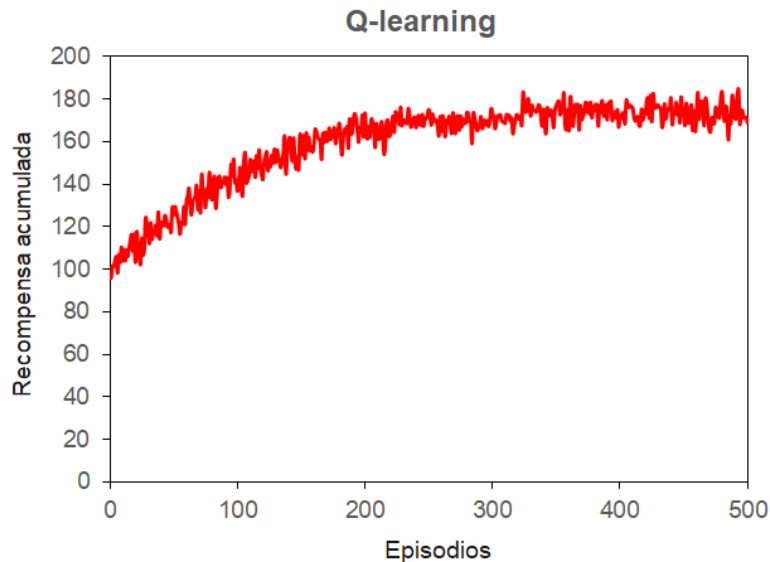


Figura 6.3: Recompensa acumulada por episodio implementando el algoritmo *Q-learning*.

### 6.3.2 SARSA

Para demostrar la versatilidad de la arquitectura propuesta, en el módulo SATDD se implementó otra metodología de toma de decisiones comparable basada en el algoritmo SARSA [Sutton and Barto, 1998]. El pseudocódigo del algoritmo SARSA implementado se proporciona en el Algoritmo 4.

La ecuación fundamental se presenta en la línea 8 y es similar al algoritmo anterior, con la diferencia que se requiere también la próxima acción a tomar  $A'$ . De aquí el nombre SARSA que simboliza la transición:  $s_t, a_t, r_{t+1}, s_{t+1}$  y  $a_{t+1}$ .

La diferencia fundamental entre *Q-learning* y SARSA reside en que el primer algoritmo es *off-policy*, mientras que el segundo es *on-policy*. Esto significa que abordan de forma diferente la paradoja de exploración-explotación, lo cual afecta su convergencia y su desempeño práctico en diferentes tipos de entornos. Los métodos dentro de la política como SARSA mejoran directamente la política que sigue el agente, mientras que los métodos fuera de la política como *Q-learning* pueden usar datos de una política diferente para su estimación.

De manera análoga al caso anterior, el ajuste de los hiperparámetros se realizó mediante la metodología de búsqueda en cuadrícula con los valores de la Tabla 6.1. La mejor recompensa obtenida después de un entrenamiento de 500 episodios es

**Algorithm 4** SARSA

---

**Require:** tamaño de paso  $\alpha \in (0, 1]$ ,  $\epsilon > 0$

- 1: Inicializar  $Q(s,a)$  para todo  $s \in S$ ,  $a \in A$
- 2: **for** episodio=0 **to**  $i$  **do**
- 3:   Inicializar  $S$
- 4:   Elegir  $A$  de  $S$  utilizando políticas derivadas de  $Q$  ( $\epsilon - greedy$ )
- 5:   **for** paso=0 **to**  $j$  **do**
- 6:     Tomar acción  $A$ , observar  $R$ ,  $S'$
- 7:     Elegir  $A'$  de  $S'$  utilizando políticas derivadas de  $Q$  ( $\epsilon - greedy$ )
- 8:      $Q(S,A) \leftarrow Q(S,A) + \alpha[R + \gamma Q(S', A') - Q(S,A)]$
- 9:      $S \leftarrow S'$ ;  $A \leftarrow A'$
- 10:   **end for**
- 11: **end for**

---

usando  $\alpha=0.01$  y un factor de descuento  $\gamma=0.5$ . El entrenamiento se realizó siguiendo la estrategia  $\epsilon$ -greedy presentada en el algoritmo  $Q$ -learning. La Figura 6.4 muestra la recompensa obtenida durante el entrenamiento SARSA.

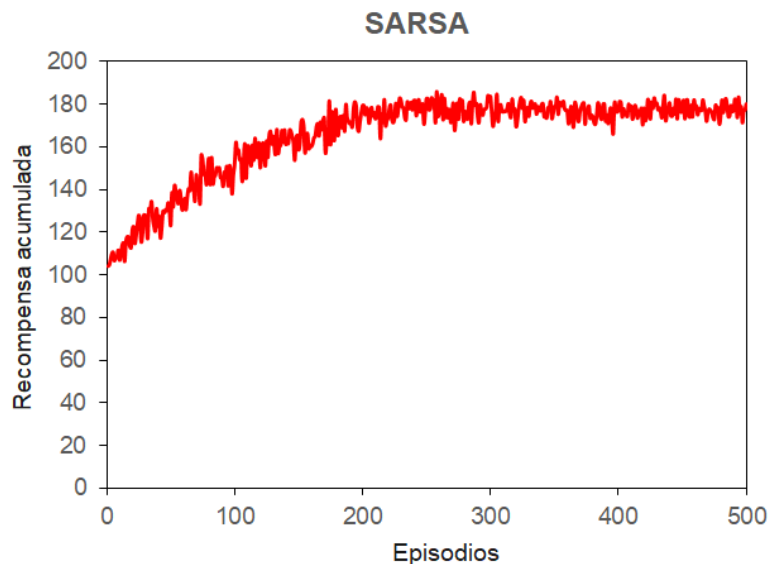


Figura 6.4: Recompensa acumulada por episodio implementando el algoritmo SARSA.

Con estos casos sencillos, se demuestra que se pueden probar fácilmente distintas aproximaciones para la TDD. Esto es posible gracias a la modularidad que admite la arquitectura propuesta.

### 6.3.3 DQN

Finalmente, se propone una formulación basada en un algoritmo más sofisticado, el algoritmo DQN [Mnih et al., 2015] (visto en la Sección 3.5.2), para abordar este problema de toma de decisiones para el manejo de materiales. El pseudocódigo del algoritmo DQN implementado se proporciona en el Algoritmo 5. En esta formulación

se utiliza una estrategia de exploración  $\epsilon$ -greedy, experience replay y una red neuronal profunda para la aproximación de políticas y funciones de valor.

---

**Algorithm 5** Deep Q - Network
 

---

```

1: Inicializar memory replay
2: Inicializar el valor de acción de la función principal  $Q$  con pesos aleatorios  $w$ 
3: Inicializar el valor de acción de la función objetivo  $Q^*$  con pesos  $w^*=w$ 
4: Cargar datos iniciales: estados  $S_t$  y acciones  $A_t$ 
5: for episodio=0 to  $i$  do
6:   Restablecer el estado inicial
7:   for paso=0 to  $j$  do
8:     Con probabilidad  $\epsilon$  seleccionar aleatoriamente una acción  $A_t$ 
9:     Ejecutar la acción y observar la recompensa.
10:    Identificar el próximo estado  $S_{t+1}$ 
11:    Guardar transición a la memoria
12:    Tomar una muestra aleatoria de la memoria.
13:    Calcular  $Q$  para  $S_t$  y  $S_{t+1}$ 
14:    Calcular  $Q$  esperada
15:    Calcular la perdida de la función  $Q$  actual y  $Q$  esperada
16:   end for
17:   Actualizar  $Q^*$ 
18:   Actualizar  $\epsilon$ 
19: end for

```

---

Al igual que en los casos anteriores, el ajuste de los hiperparámetros se realizó mediante la metodología de búsqueda en cuadrícula con los valores de la Tabla 6.1, agregando otros parámetros importantes para este algoritmo. Los mismos se presentan en la Tabla 6.2.

Batch Size	25	50	100	500
Target Update	50	100	500	1000

Tabla 6.2: Parámetros adicionales probados para el algoritmo DQN.

La política y las funciones de valor se aproximan mediante redes neuronales artificiales que toman el vector de estado del sistema como entrada, mientras que la salida se define con una función ReLu [Agarap, 2018]. A su vez, se definen dos capas ocultas en la red de 64 y 124 neuronas, respectivamente, utilizando las funciones de activación del tipo ReLU. El agente fue entrenado para 500 episodios y un decaimiento  $\epsilon$  igual al de los casos anteriores.

Finalmente, el mejor rendimiento del algoritmo DQN se obtuvo con los valores presentados en la Tabla 6.3. Los resultados se presentan en la Figura 6.5, que muestra la evolución de la recompensa acumulada. Como se puede observar, la recompensa final acumulada obtenida es mayor que las dos metodologías anteriores. Esto muestra la capacidad del algoritmo DQN para converger más rápidamente hacia una política óptima.

A través de este estudio de caso de aplicación simplificado, fue posible demostrar cómo, utilizando la arquitectura propuesta, se pueden integrar fácilmente diferentes

Parámetro	Valor
Tasa de aprendizaje $\alpha$	0.001
Factor de descuento $\gamma$	0.1
Target Update	1000
Batch Size	50
Optimizador	Adam
Función de pérdida	MSELoss

Tabla 6.3: Mejor rendimiento para el algoritmo DQN.

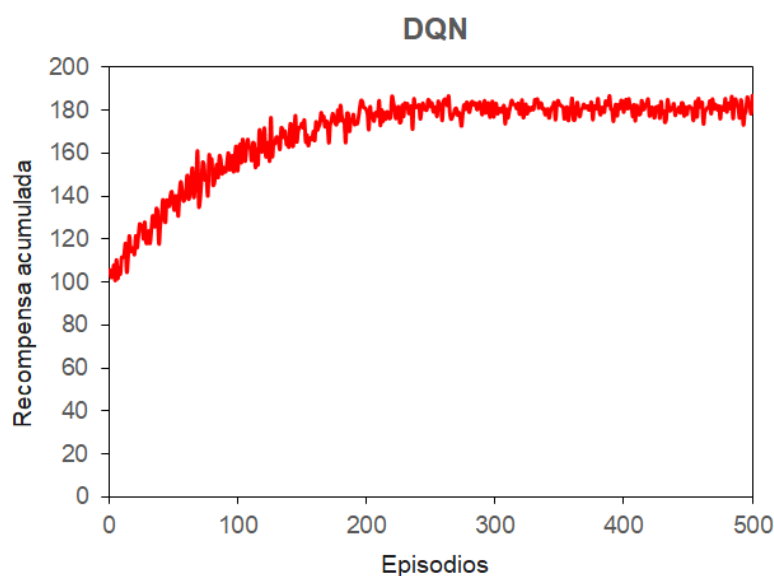


Figura 6.5: Recompensa acumulada por episodio implementando el algoritmo DQN.

metodologías de toma de decisiones basadas en inteligencia artificial en el módulo SATDD e interactuar con un modelo digital de un sistema de fabricación en particular. Estos resultados preliminares nos permiten ver el potencial de la propuesta para desarrollar gemelos digitales, especialmente para plantas industriales donde se utilice cualquier sistema autónomo, y su fácil integración con un sistema de toma de decisiones.

Hay que mencionar que el planteamiento del problema de aprendizaje por refuerzo depende en gran medida del caso de estudio abordado. Por lo tanto, debe realizarse una formulación específica para cada caso de estudio para plantear el problema como un proceso de decisión de Markov. Sin embargo, cabe señalar que esta cuestión es una característica de cualquier sistema de toma de decisiones basado en un algoritmo de AR o cualquier otro.

Si bien aquí se implementan sistemas de toma de decisiones basados en la metodología de AR, se podría implementar cualquier otro tipo de sistema de decisión, como los basados en sistemas expertos, sistemas de lógica difusa, redes bayesianas, entre otros.

## 6.4 Conclusiones del capítulo

En el presente capítulo, se desarrolló un caso de estudio en el que la arquitectura presentada y el SATDD se utilizan para la optimización del manejo de materiales mediante algoritmos de AR. La implementación demuestra la facilidad de integración de estas técnicas de inteligencia artificial para la toma de decisiones dentro de un entorno de manufactura flexible.

Entre las principales fortalezas de este estudio se destaca el uso de la simulación y los GDs como herramientas fundamentales para la experimentación y optimización de procesos sin interferir en la producción real de una organización. Estas tecnologías permiten evaluar múltiples estrategias operativas, identificar oportunidades de mejora y predecir el impacto de distintas configuraciones antes de su implementación en entornos físicos, lo que reduce costos y minimiza los riesgos asociados a cambios en la línea de producción.

Otra ventaja clave es la adaptabilidad del enfoque, ya que puede aplicarse a diversos escenarios dentro de la manufactura flexible, ajustándose a distintos modelos de producción y niveles de automatización. Esto proporciona una solución versátil y escalable, capaz de integrarse en sistemas industriales con necesidades cambiantes y dinámicas. Además, el estudio contribuye significativamente a la mejora en la eficiencia operativa, ya que la implementación de estrategias basadas en simulación permite, como en el caso de estudio visto, reducir tiempos de espera, optimizar movimientos de materiales y minimizar la dependencia de buffers intermedios, promoviendo un flujo de producción más ágil y continuo.

Por último, un aspecto innovador del enfoque es la incorporación de IA y AR, que permiten mejorar continuamente las decisiones en el sistema a medida que se recopilan nuevos datos. Estas tecnologías facilitan la adaptación automática del modelo a variaciones en la demanda, el entorno de producción y otros factores críticos, asegurando una optimización constante del desempeño del sistema.

En conjunto, estos elementos posicionan el estudio como una contribución relevante para la evolución de los sistemas de manufactura inteligente, alineándose con los principios de la I4.0 y la digitalización de procesos productivos. Si bien, la implementación en fábricas existentes puede ser difícil debido a la resistencia al cambio y la falta de personal capacitado; la transición a estas tecnologías puede ser gradual, comenzando con proyectos piloto. Además, los algoritmos de AR y las simulaciones pueden actualizarse continuamente con datos en tiempo real, lo que permite adaptarse a cambios en la producción y minimizar errores derivados de datos inexactos.



# Capítulo 7

## Algoritmo de optimización para la gestión eficiente de materiales con minimización de desperdicios

Como se estableció al comienzo de este trabajo de tesis, una de las principales motivaciones consiste en generar nuevas propuestas para el desarrollo de sistemas de TDD para casos de sistemas de manufactura flexible aprovechando las posibilidades que brinda la I4.0. En este sentido, usar los GDs en las formulaciones de sistemas de TDD autónomos es una tendencia en crecimiento en esta era de transición, para pasar del anterior concepto “*simulation-based*” hacia el nuevo “*twin-based*”. Como ya se ha presentado, para que esto sea posible es necesario poder lograr una integración entre los agentes físicos del sistema de manufactura y los agentes digitales. Con todos los desarrollos realizados y descritos hasta el capítulo anterior se ha demostrado que es posible vincular cada una de las entidades intervinientes, desde dispositivos físicos hasta agentes abstractos como lo es un sistema de toma de decisiones autónomo. Como ya se ha visto, en las formulaciones previas del módulo de TDD se han realizado y probado formulaciones basadas en algoritmos del campo de la IA, específicamente del AR, para resolver un problema típico como es el manejo de materiales en pisos de planta. Sin embargo, en este capítulo se pretende dar un paso más en la formulación de un SATDD, que pueda abordar un caso más complejo mediante el uso de técnicas avanzadas de optimización e IA, con capacidad para dar respuesta al problema de optimización de manejo de materiales en pisos de planta flexibles, minimizando el consumo de recursos y reduciendo los desperdicios asociados a la gestión del flujo de materiales.

Como se ha detallado en el Capítulo 3, el algoritmo que se implementará en este estudio corresponde a un enfoque metaheurístico desarrollado en el marco de esta investigación. Dicho algoritmo se fundamenta en variantes existentes, específicamente el SCA y su versión mejorada DSCA, e incorpora estrategias de AR para mejorar la parametrización y adaptabilidad. La inclusión del AR permite al algoritmo ajustar dinámicamente sus parámetros a medida que aprende de las iteraciones previas, favoreciendo la convergencia hacia soluciones óptimas en menor tiempo y con mayor estabilidad.

El principal objetivo de este caso de estudio es lograr una reducción significativa en las distancias recorridas por los equipos de transporte, optimizando así el uso de los recursos logísticos dentro de la planta. Para ello, se evaluará el impacto de las

soluciones generadas sobre la reducción de desperdicios en el sistema al finalizar la simulación.

En este análisis, se considerarán múltiples escenarios de simulación con diferentes configuraciones de demanda y disponibilidad de recursos, permitiendo evaluar la robustez y eficacia del algoritmo desarrollado. Asimismo, se validará su desempeño comparándolo con otras técnicas, con el fin de demostrar su superioridad en términos de adaptabilidad y precisión en la optimización del manejo de materiales en entornos industriales complejos.

## 7.1 Definición del problema

En la presente investigación, se adapta el problema presentado en [Gola and Kłosowski, 2019] como estudio de caso, con el objetivo de analizar y optimizar el manejo de materiales en un entorno de manufactura flexible. En particular, este caso de referencia presenta un alto nivel de complejidad debido a las restricciones operativas y logísticas que caracterizan el sistema. El problema planteado involucra la optimización de las rutas de los vehículos guiados automáticamente (AGV) en una planta de fabricación con un flujo de materiales dinámico y demanda variable.

El sistema de manufactura considerado en este caso de estudio está compuesto por 20 estaciones de trabajo y un único vehículo autónomo encargado del transporte de piezas entre las estaciones. El sistema de transporte cumple dos funciones principales en momentos específicos del tiempo: la entrega de piezas a los puestos de trabajo y la recolección de piezas una vez finalizado su procesamiento. Para ello, se establece un único punto de salida y llegada del vehículo, el cual corresponde a la estación de expedición. Una vez que el vehículo ha partido de este punto, la ruta de transporte permanece fija hasta su regreso, sin posibilidad de modificaciones dinámicas durante su recorrido.

En este apartado, se preservan varios aspectos fundamentales del problema original, tales como la disposición de la planta, los tiempos de operación de las estaciones de trabajo y las capacidades de carga asignadas a cada una de ellas. Asimismo, se mantiene la estructura de existencias iniciales (stock), la cual equivale a la carga establecida en cada estación. La Tabla 7.1 contiene toda la información acerca de los parámetros de configuración del sistema. Como puede verse, para cada una de las estaciones de trabajo, se tiene el stock inicial y el tiempo de procesamiento.

Como se realizó para el caso de estudio presentado en el capítulo anterior, en la Figura 7.1 se presenta el modelo conceptual del problema. De manera análoga, se usó la metodología IDEF-SIM como primera instancia de representación y análisis del sistema. En la próxima subsección se desarrollará el modelo digital correspondiente.

El diagrama IDEF-SIM además de ser la primera instancia de representación del sistema, permite identificar rápidamente y a “simple vista” ciertas operaciones que no agregan valor, como lo son, en este caso, las operaciones de transporte de materiales. Sin embargo, a pesar de esta posibilidad que brinda esta herramienta, minimizar los movimientos de partes en sistemas productivos no es una cuestión trivial y requiere de la formulación de estrategias sofisticadas para la optimización de estos movimientos, ya que generalmente estamos frente a un problema de tipo *NP-hard*, como ya hemos discutido en el Capítulo 2. Una planificación eficiente de los desplazamientos impacta directamente en la reducción de tiempos de espera, el uso eficiente de recursos y la mejora de la productividad global del sistema. En este

Estación	Stock inicial (u)	Tiempo de procesamiento de una carga (s)
1	50	4000
2	60	4200
3	20	4000
4	20	4800
5	10	4000
6	40	4000
7	40	4000
8	30	3900
9	20	4000
10	10	4000
11	20	4000
12	50	5000
13	30	3750
14	10	4000
15	20	4000
16	30	6600
17	10	4000
18	50	4000
19	25	3500
20	25	3500

Tabla 7.1: Datos básicos del segundo caso de estudio.

sentido, la presente investigación busca evaluar la aplicabilidad de técnicas avanzadas de optimización y AR en la resolución de este problema, proporcionando soluciones más flexibles y adaptativas que las estrategias tradicionales.

El objetivo final de este estudio es minimizar la distancia recorrida por el vehículo en cada ruta programada dentro del sistema de manufactura. Para lograrlo, es fundamental optimizar la planificación de los desplazamientos del vehículo, asegurando un uso eficiente de los recursos logísticos y reduciendo los tiempos de espera y los costos operativos asociados.

Como paso previo a la optimización de rutas, se establece un mecanismo de decisión basado en reglas fijas predeterminadas que determina a qué estaciones debe desplazarse el vehículo en cada ciclo operativo. Este proceso de selección de destinos se lleva a cabo considerando el estado actual de los *buffers* en cada estación, lo que permite priorizar aquellas ubicaciones con necesidades urgentes de abastecimiento o recolección de materiales.

Además, se establece una lógica de decisión para determinar si el transporte debe llevar materiales a una estación o retirar los productos terminados. Esta decisión depende de múltiples factores, entre ellos, el nivel de stock disponible en cada estación, el tiempo restante para que se agoten las piezas en proceso y la cantidad de material acumulado que debe ser retirado. De este modo, el sistema garantiza una gestión eficiente del flujo de materiales, evitando interrupciones en la producción y

asegurando un equilibrio óptimo en la utilización de los recursos de almacenamiento y transporte.

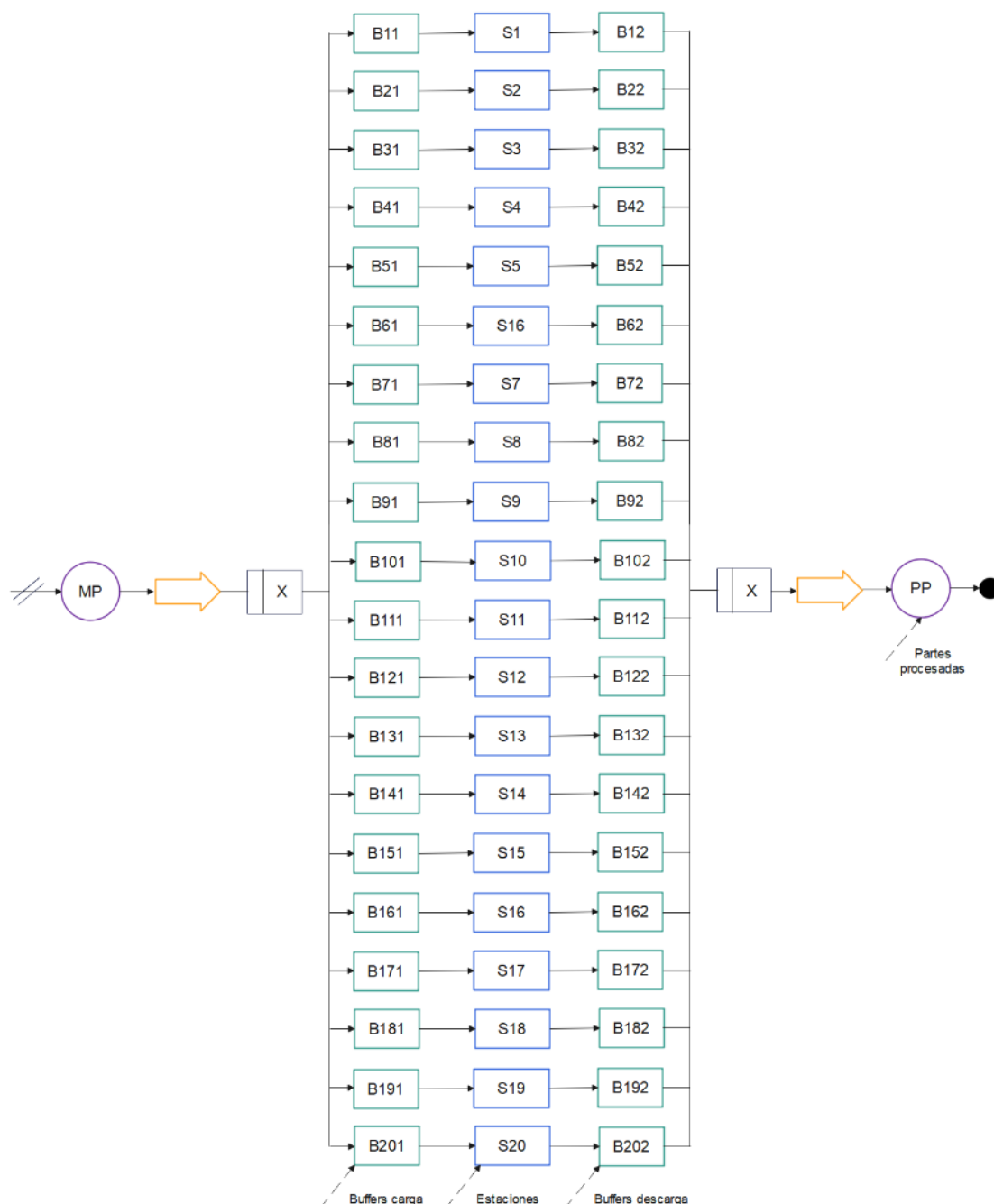


Figura 7.1: IDEF-SIM del segundo caso de estudio.

La implementación de esta estrategia de optimización no solo contribuye a la reducción de la distancia recorrida por el vehículo, sino que también permite mejorar la sincronización de los procesos productivos, minimizar el riesgo de cuellos de botella y aumentar la eficiencia general del sistema. En este sentido, la integración de modelos avanzados de planificación de rutas, combinados con técnicas de optimización basadas en aprendizaje automático y heurísticas, puede proporcionar mejoras significativas en la gestión del transporte interno dentro de entornos de manufactura flexible.



20algorithm.

## 7.3 Algoritmo seno-coseno discreto reforzado

La optimización del manejo de materiales en los sistemas de producción se refiere al movimiento eficiente de materias primas, partes en proceso y productos terminados dentro de una instalación de producción durante la fabricación. Resolver este problema comúnmente requiere decisiones que implican elegir una acción específica de un conjunto finito de opciones o elecciones, donde las variables involucradas solo pueden tomar valores discretos. Además, mayoritariamente, en cualquier sistema productivo existen una cantidad de restricciones que deben cumplirse para que la solución encontrada sea factible. Por ello, utilizar un modelo lo más cercano a la realidad (“gemelo”), como un modelo de simulación (visto en la sección anterior), permite encontrar soluciones más adaptadas al caso real.

Como ya se indicó en el Capítulo 3, el algoritmo DSCA es un buen candidato para resolver este tipo de problema. Aunque el SCA se ha utilizado en varios trabajos anteriores, el DSCA es un método novedoso que ha recibido poca atención y aún no se ha utilizado para abordar problemas de manejo de materiales. Sin embargo, como veremos en la Sección 7.4 de resultados, el DSCA estándar tiene un rendimiento inferior al algoritmo propuesto en esta tesis para resolver el problema. Esto se debe principalmente a que su convergencia se ve afectada por los valores establecidos de los parámetros  $r_i$  en la Ecuación 3.3.

Por lo tanto, para mejorar el rendimiento de este algoritmo, se ha utilizado como inspiración otro algoritmo basado en SCA, llamado *Q-learning embedded Sine Cosine Algorithm (QLESCA)*, propuesto recientemente por [Hamad et al., 2022]. QLESCA incluye una formulación del *Q-learning* para la selección de parámetros en el SCA. El algoritmo *Q-learning* ayuda a cada agente QLESCA a moverse adaptativamente en el espacio de búsqueda de acuerdo con su desempeño, donde la recompensa o penalización se otorga en función del logro del agente. Cada agente tiene una Tabla  $Q$ , donde se obtienen los parámetros  $r_1$  y  $r_3$  en función del estado y acción elegidos. El Algoritmo 6 describe el pseudocódigo original de QLESCA.

Aunque QLESCA mejora el rendimiento de SCA, fue desarrollado para abordar problemas de optimización en dominios continuos, y el problema de optimización que se trata en este trabajo requiere soluciones discretas. Entonces, tomando en cuenta los Algoritmos 2 y 6, se desarrolló el Algoritmo seno-coseno discreto reforzado, denominado *R\*DSCA* descrito en el pseudocódigo que se muestra a continuación en el Algoritmo 7.

Una novedad importante de la propuesta del *R\*DSCA* es que se ha desarrollado un mecanismo para encontrar una forma óptima de aprender a aumentar el parámetro  $r_1$ . El coeficiente  $r_1$  se conoce como parámetro de control de transición y su función principal es lograr un equilibrio entre diversidad y convergencia durante el procedimiento de búsqueda [Gupta et al., 2022]. Preferiblemente, el valor de este parámetro debería aumentarse entre iteraciones para proporcionar una transición de la exploración a la explotación del espacio de búsqueda. Como veremos a continuación, incluir un mecanismo de aprendizaje para la selección de  $r_1$  mejora sustancialmente el rendimiento del algoritmo. De esta forma, el parámetro de control de transición  $r_1$  podría actualizarse siguiendo alguna ley de comportamiento. Sin embargo, la posibilidad de aprender una ley de comportamiento introduce un grado

---

**Algorithm 6** *Q-learning embedded Sine Cosine Algorithm*

---

- 1: **Definir**  $T =$  máximo número de iteraciones
  - 2: **Inicializar** un conjunto de agentes de búsqueda ( $X$ ) y las correspondientes Tablas  $Q$
  - 3: **Evaluar** cada agente en la función objetivo y guardar los resultados
  - 4: **Actualizar** la mejor solución encontrada hasta el momento ( $P = X^*$ )
  - 5: **Mientras**  $t < T$  :
    - 6:   **Para** cada agente:
      - 7:       **Calcular** el estado actual  $S_t$  para cada agente
      - 8:       **Elegir** una acción  $A_t$  y actuar
      - 9:       **Actualizar**  $r_1, r_2, r_3$  y  $r_4$
    - 10:    **Actualizar** las posiciones de los agentes de búsqueda usando la Ec. 3.1
    - 11:    **Actualizar** la mejor solución de cada agente:
      - 12:       **Si** la solución es mejor: Recompensa = 1
      - 13:       **Sino** Recompensa = -1
    - 14:    **Actualizar** las Tablas  $Q$  usando la Ec. de *Q-learning*
  - 15: **Devolver** la mejor solución encontrada como la solución global
- 

---

**Algorithm 7** *R\*DSCA*

---

- 1: **Definir**  $T =$  máximo número de iteraciones
  - 2: **Inicializar** un conjunto  $\mathcal{X}$  de agentes de búsqueda y las correspondientes tablas  $Q_i$  para cada agente  $X_i \in \mathcal{X}$
  - 3: **Evaluar**  $f(X_i) \forall X_i \in \mathcal{X}$  y guardar los resultados,  $f_i$ , tal que  $f_i = f(X_i) \in \mathcal{F}, i = 1, 2, \dots, |\mathcal{X}|$
  - 4: **Actualizar** la mejor solución encontrada hasta el momento, tal que  $X_d = \max_{X \in \mathcal{X}} \{f_i \in \mathcal{F}\}$
  - 5: **Mientras**  $t < T$  :
    - 6:   **Para** cada  $X_i^t \in \mathcal{X}$ :
      - 7:       **Si**  $p < \varepsilon$  :
        - 8:           **Definir** aleatoriamente las posiciones de los agentes  $X_i^t$  y evaluar  $f(X_i^t)$
      - 9:       **Sino**:
        - 10:           **Obtener** el estado actual  $s_t$  como el  $s^* \in S$  más cercano a  $\delta_i^t$  (Ec. 7.1)
        - 11:           **Con**  $\pi_i^*$  obtener  $r_1 = \pi_i^*(s_i) = \max_r Q(S, r)$  (Ec. 3.10)
        - 12:           **Actualizar**  $r_2$  y  $r_3$  aleatoriamente
        - 13:           **Calcular**  $C_i^t$  como en la Ec. 3.3
        - 14:           **Actualizar** la posición del agente de búsqueda  $X_i^t$  usando la Ec. 3.2
        - 15:           **Evaluar**  $f(X_i^t)$
        - 16:           **Calcular** la recompensa comparando  $f(X_i^t)$  con la mejor solución encontrada del agente
    - 17:    **Actualizar**  $Q_i$  usando la Ec. de *Q-learning*
  - 18: **Devolver** la mejor solución encontrada como la solución global  $X_d$
-

adicional de libertad al algoritmo que puede aprovecharse para mejorar su rendimiento. Es decir, en lugar de fijar una función que describa ese comportamiento de antemano, podemos aprender a construir una función monótona creciente a partir de un conjunto finito  $\mathcal{R}$  de funciones más elementales  $r_{1,j} \in \mathcal{R}, j = 1, \dots, N$  usando el algoritmo *Q-learning*. Entonces, para implementar una versión tabular del algoritmo *Q-learning* de una manera sencilla, debemos tener un conjunto de estados finitos. Aún así, para convertir el problema de selección  $r_1$  en un problema de decisión de Markov finito puramente discreto, se debe definir una estrategia para tratar con estados discretos.

Por lo tanto, definimos el espacio de estados  $S$  en el dominio  $[0, 1]$ , tal que cada  $s_\tau$  es un intervalo contenido en  $S$ , tal que  $s_\tau \subset S$  con  $s_\tau = 1, \dots, m$  y los estados  $s_1, \dots, s_m$ , son intervalos congruentes en el dominio de  $S$ . Además, definimos la distancia relativa actual  $\delta_i^t$  entre las soluciones encontradas y la solución óptima como:

$$\delta_i^t = \frac{X_d^t \ominus X_i^t}{l} \quad (7.1)$$

donde el numerador indica el número de características diferentes entre la solución encontrada y la solución óptima, y el denominador indica la longitud de estos vectores de solución. Entonces, el estado actual  $s_t$  será el  $s^* \in S$  más cercano al valor actual  $\delta_i^t$ . Es decir, para cada agente en cada instante de tiempo se calculará  $\delta_i^t$  y se encontrará el estado correspondiente.

De esta manera, usando esta estrategia, conseguimos obtener una máquina de estados finita para describir el problema de decisión de Markov, y en consecuencia, el algoritmo *Q-learning* podría usarse para obtener la función óptima para  $r_1$  según el estado actual  $s_t$ , siguiendo la función de valor de estado-acción dada en la ecuación de *Q-learning*.

Volviendo a la explicación del Algoritmo 7, puede verse que se parte de una solución aleatoria. Luego se evalúa cada agente mediante la función objetivo y se guarda la solución propuesta. Entonces se obtiene la mejor solución. Continuando con la parte central que es la evolución poblacional, como podemos ver, en ocasiones con una probabilidad  $\varepsilon$  se toma una solución aleatoria (líneas 7-8) y se evalúa mediante simulación, continuando por otro lado la evolución del algoritmo.

Entre las líneas 10-17 puede verse la sección del *Q-learning*, donde para cada agente se obtiene el estado actual y la ejecución de la acción, como se explicó anteriormente. Una vez que se calculan los valores de  $r_1$ , se actualizan aleatoriamente  $r_2$  y  $r_3$ . Usando las Ecuaciones 3.2 y 3.3, se actualizan las posiciones del agente y se guardan los resultados de la evaluación de su función. Los elementos que son iguales se mantienen para actualizar la posición del agente, y el orden de las características diferentes se actualiza aleatoriamente para generar una nueva solución. Para la recompensa del *Q-learning*, comparamos las mejores soluciones obtenidas para cada agente con la solución actual del agente, y luego se actualizan las Tablas  $Q$ . Finalmente, se devuelve la mejor solución para la población.

Para una mejor comprensión del algoritmo desarrollado, se muestra un diagrama de flujo del mismo en la Figura 7.3.

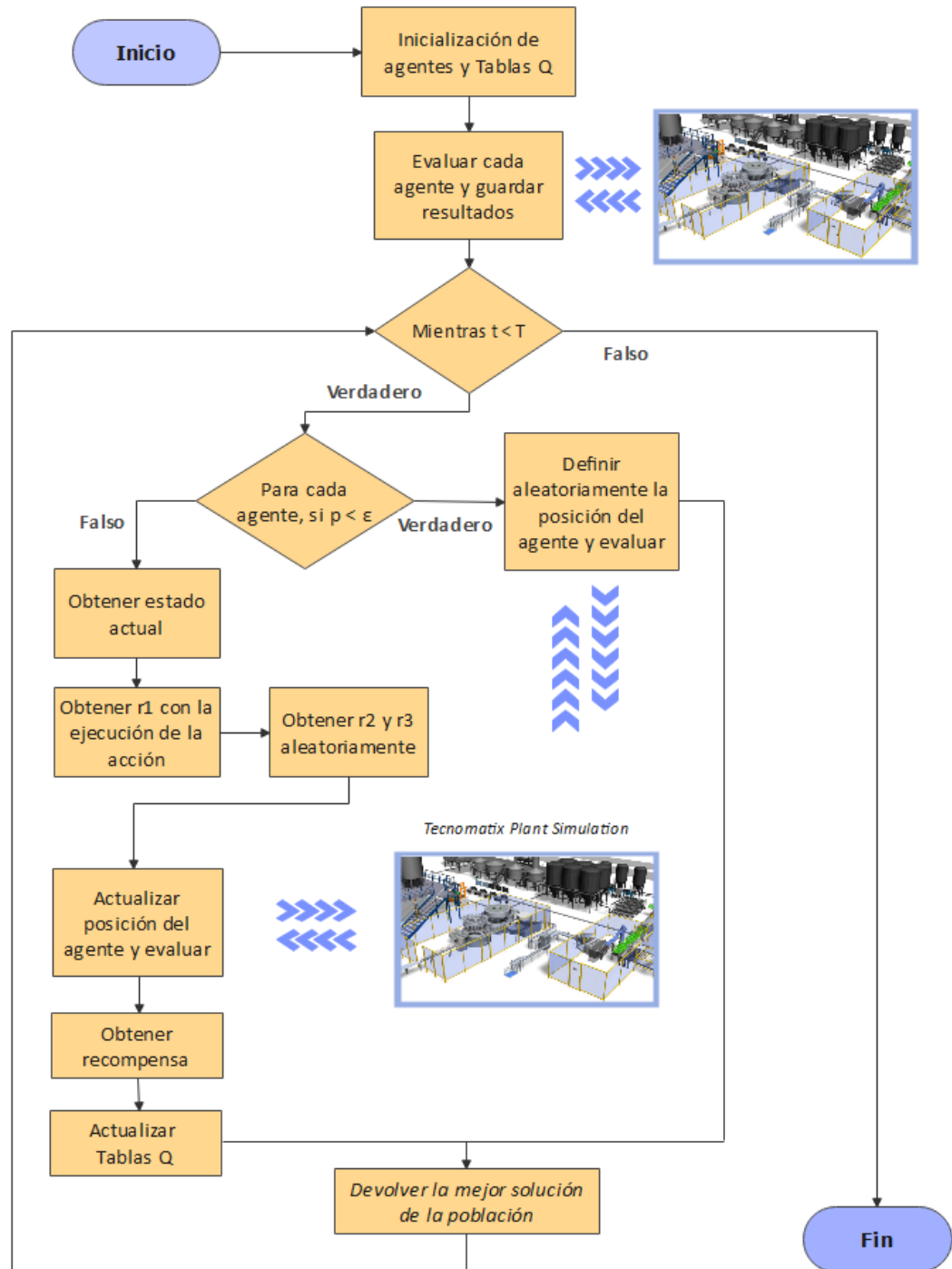


Figura 7.3: Algoritmo  $R^*DSCA$

### 7.3.1 Configuración del $R^*DSCA$

Para este problema particular presentado en el presente capítulo se definirán aquí las principales características de la propuesta. Como se mencionó,  $Q-learning$  encuentra los mejores valores de  $r_{1,j}$  para cada iteración. Se eligieron cuatro funciones crecientes para el parámetro  $r_{1,j}$ . Tres de ellas se trabajaron en el intervalo (0, 1) y el resto en el intervalo (0, 0.5). Las Ecuaciones 7.2, 7.3, 7.4 y 7.5 muestran las leyes definidas para  $r_{1_1}$ ,  $r_{1_2}$ ,  $r_{1_3}$  y  $r_{1_4}$  respectivamente. La Figura 7.4 muestra las gráficas

correspondientes a un intervalo de 30 iteraciones. Por lo tanto, las acciones  $A_t$  del algoritmo serán las diferentes funciones  $r_{1,j}$  con  $j = 1, \dots, 4$ .

$$r_{1_1} = \frac{t}{T} \quad (7.2)$$

$$r_{1_2} = \frac{t}{2T} \quad (7.3)$$

$$r_{1_3} = 1 - \epsilon^{-(t^{0,5})} \quad (7.4)$$

$$r_{1_4} = \frac{t^2}{T^2} \quad (7.5)$$

Respecto al espacio de estados necesario  $S$  en el dominio  $[0, 1]$  para definir el modelo de AR, se definieron cinco estados en base a la distancia relativa actual  $\delta_i^t$  (Ecuación 7.1), tal que  $s_\tau \subset S$  con  $s_\tau = 1, \dots, 5$ . Los estados pueden verse en la Tabla 7.2. Por lo tanto, en nuestra propuesta, cada tabla  $Q$  tiene cinco estados y cuatro acciones.

En cuanto a la recompensa, se mantuvo lo detallado en el Algoritmo 6; es decir, si la solución encontrada por el agente es mejor que la obtenida hasta el momento, la recompensa es 1; de lo contrario, es -1.

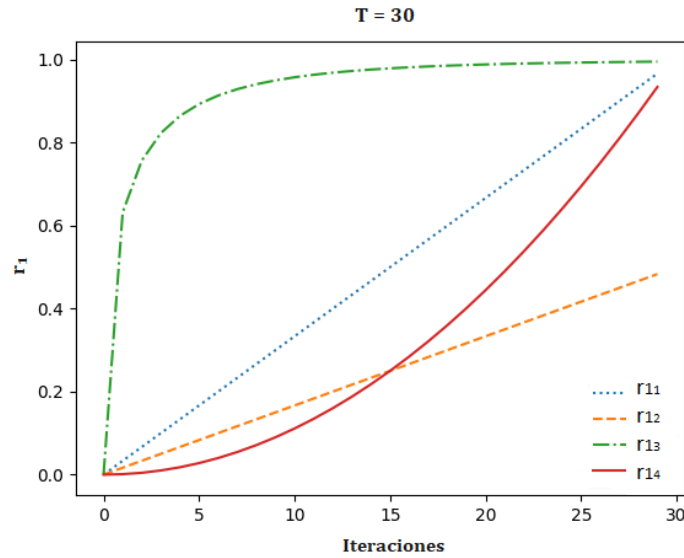


Figura 7.4: Funciones para el parámetro  $r_1$ .

$S_t$	1	2	3	4	5
Valores $S_t$	[0; 0.2]	(0.2; 0.4]	(0.4; 0.6]	(0.6; 0.8]	(0.8; 1]

Tabla 7.2: Estados para la configuración del  $R^*DSCA$ .

Finalmente,  $\epsilon$  se fijó en 0,20. Al aplicar  $Q$ -learning, los parámetros internos  $\alpha$  y  $\gamma$  se establecieron en valores de 0,1 y 0,9 respectivamente. Estos valores son los más utilizados empíricamente en la literatura. Además, el agente fue entrenado siguiendo una estrategia de exploración  $\epsilon$ -greedy, con un decaimiento de  $\epsilon$  con el transcurso de los episodios desde  $\epsilon = 1$  hasta un valor mínimo  $\epsilon = 0.01$ , para mejorar la estrategia de exploración-explotación.

## 7.4 Resultados

La presente sección se encuentra dividida en tres partes. Inicialmente, para mostrar el rendimiento del algoritmo de optimización  $R^*DSCA$  propuesto, se consideran cuatro subproblemas (o escenarios) con diferentes niveles de complejidad basados en el problema presentado en la Sección 7.1. Se comienza abordando subproblemas menos complejos, luego otro de complejidad intermedia y finalmente se aborda un subproblema más complejo que los anteriores. Cuando nos enfrentamos a problemas manejables con unas pocas estaciones de trabajo, podemos encontrar las soluciones óptimas globales simulando todas las soluciones posibles. De esta forma, aprovechamos el hecho de contar con el simulador de planta, que nos permite simular todas las soluciones posibles e identificar con precisión la óptima. Por lo tanto, la solución que proporcione el mejor valor de la función objetivo será la óptima. Luego, esta solución óptima se utilizará para comparar la solución encontrada usando la propuesta del  $R^*DSCA$ .

Luego, se muestra una comparación de rendimiento utilizando el algoritmo  $DSCA$ , y finalmente, una vez que se demostró el desempeño de la propuesta, se aborda el problema completo presentado en la Sección 7.1, para demostrar el desempeño del algoritmo  $R^*DSCA$  propuesto para un problema complejo de manejo de materiales que considera la reducción de desperdicios y para el cual es prácticamente imposible hallar la solución óptima de manera analítica. Alternativamente, encontrar la solución global óptima utilizando simulaciones computacionales intensivas es impracticable cuando se enfrentan escenarios complejos con varias estaciones de trabajo, ya que se requerirán tiempos de procesamiento computacional extremadamente grandes (o prohibitivos). Como se ha señalado anteriormente, el problema principal contiene las 20 estaciones de trabajo. Por lo tanto, esto da lugar a  $2.432902e+18$  (20!) rutas posibles o soluciones factibles.

### 7.4.1 Evaluación del $R^*DSCA$

#### Sub-problema A

Para empezar, el subproblema inicial es sencillo, compuesto por solo cinco estaciones de las 20 que componen el sistema original completo. En este caso, solo se consideran las estaciones de trabajo #1, #4, #6, #15 y #16, y la Figura 7.5 muestra la configuración del sistema resultante. Para analizar el rendimiento del algoritmo propuesto, se diseñaron cuatro experimentos de prueba para este subproblema. Los resultados de cada experimento se resumen en la Tabla 7.3.

Para determinar cómo afecta la inicialización a la solución encontrada por el algoritmo, se han realizado experimentos con una población inicial fija y con una aleatoria (como se indica en la 2da columna de la Tabla 7.3). En ambos casos, la población inicial incluye seis soluciones factibles iniciales. Además, para determinar cómo el número de iteraciones afecta la solución encontrada por el algoritmo, se han realizado experimentos con diferentes números de iteraciones  $T$ , presentados en la tercera columna de la tabla. Como se explicó anteriormente, para este problema la solución teórica óptima, denotada como  $O_{ts}$ , se encuentra mediante la simulación de todas las soluciones factibles y se proporciona en la cuarta columna de la tabla.

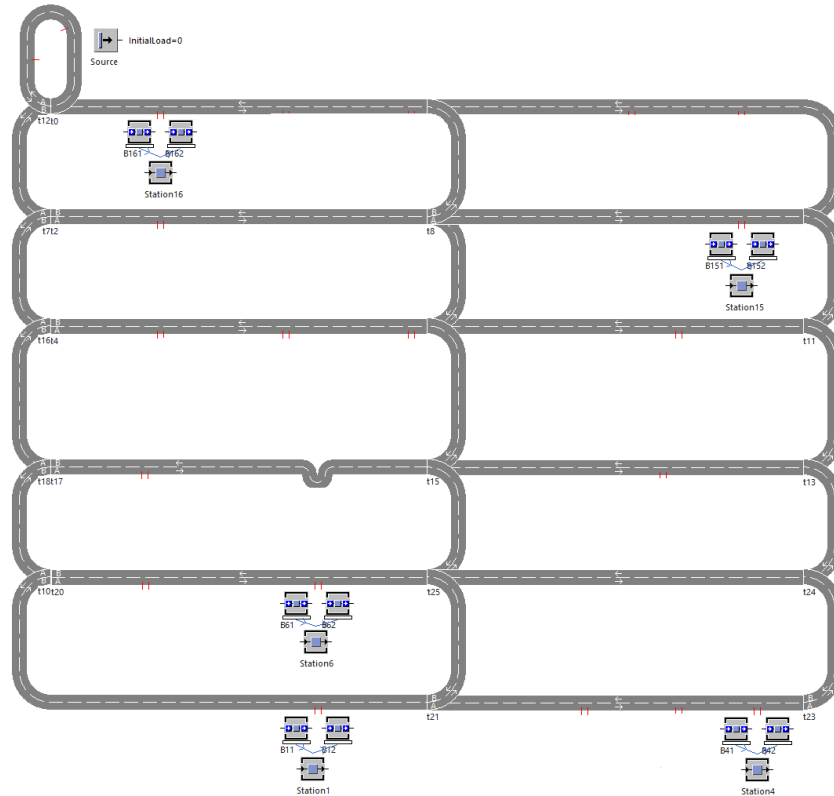


Figura 7.5: Modelo de simulación del sub-problema A.

Experimento	Inicio	T	$O_{ts}$	$\bar{D}$	$D^*$
1	fijo	30	374.3 m	386.6 m	374.3 m
2	fijo	50	374.3 m	380.6 m	374.3 m
3	random	30	374.3 m	385.0 m	374.3 m
4	random	50	374.3 m	375.9 m	374.3 m

Tabla 7.3: Experimentos del  $R^*DSCA$  para el sub-problema A.

Para cada experimento, debido a la naturaleza estocástica del  $R^*DSCA$ , el valor  $\bar{D}$ , que se muestra en la quinta columna de la tabla, corresponde al promedio de los resultados ( $D_{t_i}$ ) obtenidos para  $m = 10$  repeticiones de dicho experimento. Se calcula como en la Ecuación 7.6. La solución óptima  $D^*$ , encontrada por el algoritmo, se da en la última columna de la Tabla 7.3, y se obtiene como en la Ecuación 7.7.

$$\bar{D}_t = \frac{\sum_{i=1}^m D_{t_i}}{m}; \quad t = 1..T \quad (7.6)$$

$$D_t^* = \max_j \{D_{t_j}\} \quad ; \quad j = 1..m \quad (7.7)$$

Se puede comprobar que, para este subproblema simplificado, el algoritmo siempre encuentra la solución óptima, es decir  $D^* = O_{ts}$ . Por otra parte, como puede verse en la penúltima columna de la Tabla 7.3, las soluciones promedio halladas están muy cerca de las soluciones óptimas  $O_{ts}$ , lo que puede interpretarse como

que el algoritmo tiene una tendencia a converger rápidamente hacia la solución óptima. Además, si bien explora el espacio de soluciones factibles, lo hace de manera balanceada manteniéndose dentro de una región de búsqueda acotada.

***Sub-problema B***

En este caso se aborda una variante con la misma complejidad que en el caso anterior, es decir, se tienen en cuenta también solo cinco estaciones de trabajo, pero ahora las estaciones de trabajo son las #2, #7, #9, #14 y #19. Por lo tanto, la configuración del sistema resultante es la de la Figura 7.6 y, en consecuencia, las restricciones del problema son distintas; por ende, el problema es diferente y su solución también lo será. De la misma manera que en el caso anterior, repetimos los mismos experimentos, y los resultados obtenidos se resumen en la Tabla 7.4. Una vez más, el desempeño exitoso del algoritmo puede verse en todas las pruebas, ya que logra consistentemente alcanzar la solución óptima, es decir,  $D^* = O_{ts}$ . Además, otra cuestión que se puede observar es que la inicialización del algoritmo no afecta su rendimiento. Por lo tanto, consideraremos solo inicios aleatorios para los siguientes experimentos para evitar sesgos.

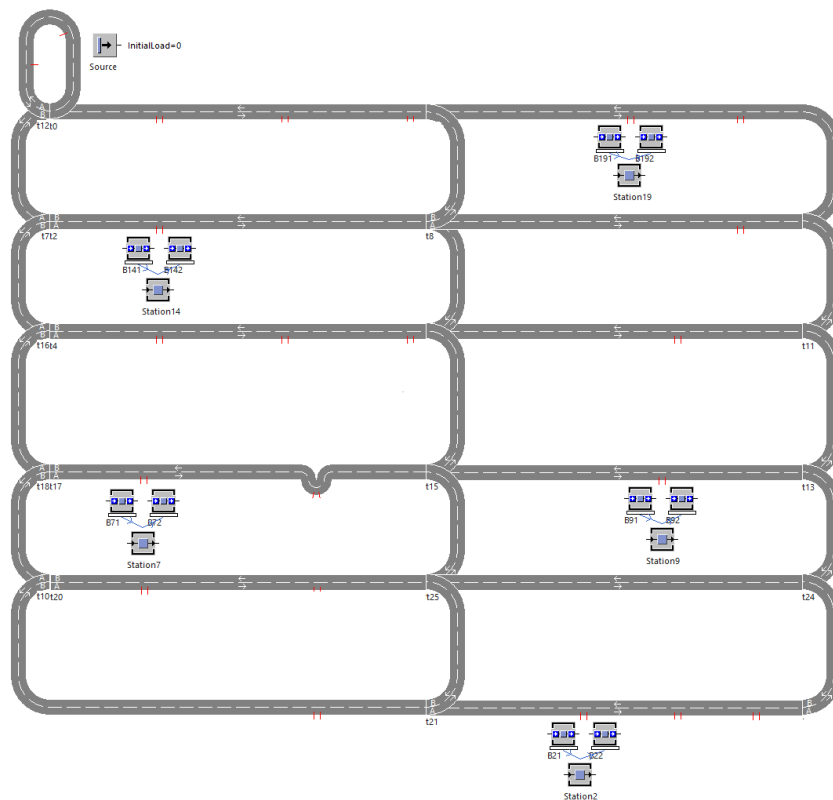


Figura 7.6: Modelo de simulación del sub-problema B.

Tanto para el sub-problema A como para el sub-problema B, el algoritmo de optimización propuesto alcanzó con éxito la solución óptima del problema. Esto demuestra que, además de llegar a la solución óptima, la propuesta es apropiada para su uso en este tipo de sistemas de producción reconfigurables.

Experimento	Inicio	T	$O_{ts}$	$\bar{D}$	$D^*$
1	fijo	30	382.1 m	386.7 m	382.1 m
2	fijo	50	382.1 m	382.2 m	382.1 m
3	random	30	382.1 m	384.0 m	382.1 m
4	random	50	382.1 m	382.2 m	382.1 m

Tabla 7.4: Experimentos del  $R^*DSCA$  para el sub-problema B.

A continuación, para profundizar en el análisis, se presentarán otros casos más complejos.

### Sub-problema C

Este subproblema plantea un escenario más complejo que los anteriores y está compuesto por siete estaciones de trabajo, las cuales son #2, #5, #7, #9, #11., #14 y #19. La configuración del piso de planta resultante se muestra en la Figura 7.7.

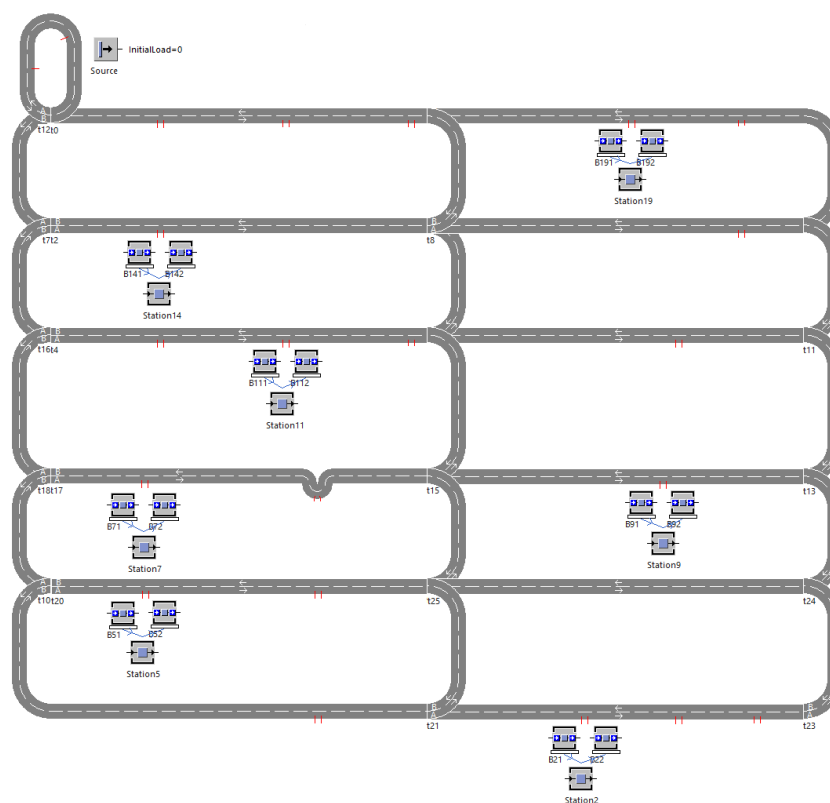


Figura 7.7: Modelo de simulación del sub-problema C.

Este subproblema es más complejo que los anteriores porque aumenta el espacio de soluciones factibles. En consecuencia, como es sabido, en estos casos es necesario aumentar el número de iteraciones para que el algoritmo pueda explorar alternativas factibles hacia la búsqueda de la solución óptima. Por lo tanto, se incluirán tres experimentos adicionales, aumentando el número máximo de iteraciones  $T$ .

La Tabla 7.5 resume los resultados obtenidos. Como puede verse, el algoritmo

siempre encuentra la solución óptima ( $D^* = O_{ts}$ ), incluso cuando se ejecuta con pocas iteraciones. Como en los casos anteriores, en la 4ta columna de la tabla, se puede observar que a medida que  $T$  aumenta, el rendimiento del algoritmo mejora ya que la media de las soluciones óptimas encontradas disminuye. Cuando se llega a las 500 iteraciones, el promedio de las soluciones se acerca bastante a la solución óptima.

Experimento	T	$O_{ts}$	$\bar{D}$	$D^*$
1	30	504.1 m	534.2 m	504.1 m
2	50	504.1 m	525.5 m	504.1 m
3	100	504.1 m	517.2 m	504.1 m
4	200	504.1 m	515.0 m	504.1 m
5	500	504.1 m	509.4 m	504.1 m

Tabla 7.5: Experimentos del  $R^*DSCA$  para el sub-problema C.

### Sub-problema D

Para finalizar con el análisis de rendimiento del algoritmo, vamos a considerar un problema complejo en el que tenemos diez estaciones de trabajo, las cuales son: #1, #2, #4, #8, #9, #11, #12, #14, #15 y #19 y la configuración del sistema de producción es como en la Figura 7.8. A diferencia de los casos anteriores, no tenemos la solución óptima teórica  $O_{ts}$  (10! soluciones factibles). Por lo tanto, analizamos el rendimiento del algoritmo a medida que aumenta el número de iteraciones  $T$ .

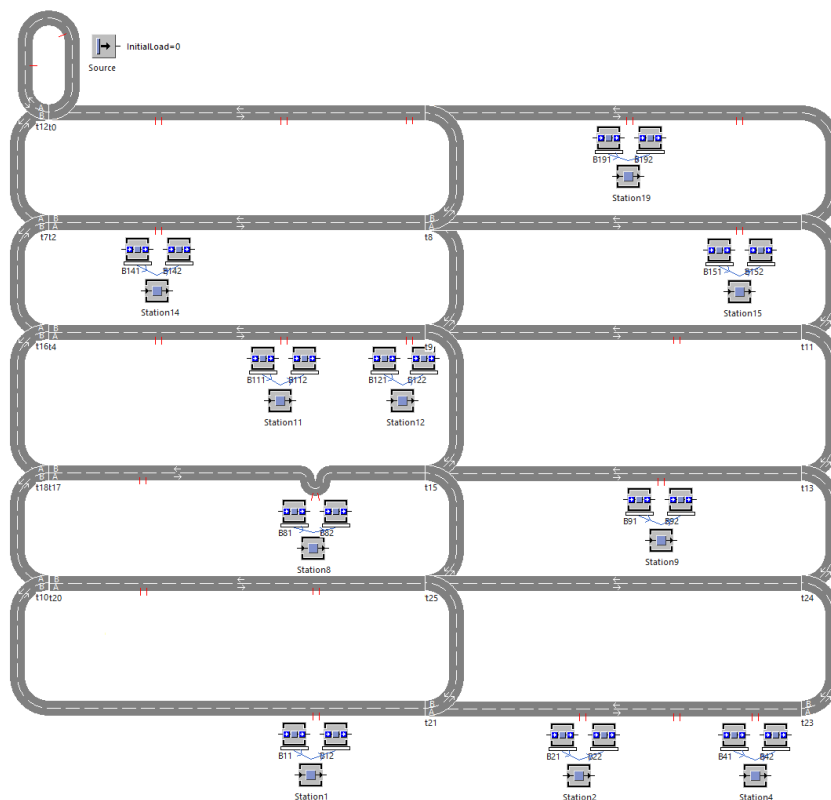


Figura 7.8: Modelo de simulación del sub-problema D.

La Tabla 7.6 muestra los resultados experimentales obtenidos. Se observa que cuanto mayor es el número de iteraciones  $T$ , menores son las distancias promedio obtenidas  $\bar{D}$ . Las soluciones óptimas encontradas  $D^*$  también son mejores.

Experimento	T	$\bar{D}$	$D^*$
1	50	724.1 m	677.2 m
2	100	684.0 m	653.1 m
3	150	671.2 m	611.3 m
4	200	658.0 m	598.9 m
5	500	613.5 m	566.4 m

Tabla 7.6: Experimentos del  $R^*DSCA$  para el sub-problema D.

Como se ha señalado, el algoritmo metaheurístico basado en simulación propuesto, que llamamos  $R^*DSCA$ , contribuye a determinar las políticas de manejo de materiales para sistemas de producción flexibles, donde la reconfiguración de las instalaciones se puede llevar a cabo con frecuencia para cumplir con diferentes requisitos de producción. Dependiendo del producto a elaborar en cada momento y las actividades que implique el proceso productivo del mismo, pueden ir variando las rutas necesarias para cumplir en tiempo y forma con el plan de manejo de materiales.

## 7.4.2 Comparación de rendimiento

A continuación, se realiza una comparación del desempeño de la propuesta con otra metodología comparable para abordar los subproblemas vistos (A a D). Específicamente, utilizamos el DSCA original (Algoritmo 4). Para la aplicación del DSCA [Gupta et al., 2022] en este caso particular, la diferencia en características de los vectores (visto en la Sección 3.4) implica la discrepancia en la secuencia de las rutas probadas. Además, con una cierta probabilidad ( $p = 0,20$ ), se agrega la elección de soluciones aleatorias para aumentar la exploración de soluciones factibles.

De la misma forma, como en el caso anterior, todas las pruebas se repitieron diez veces con seis soluciones iniciales para cada experimento. Resumiendo, a continuación en la Tabla 7.7 se encuentran los resultados de los experimentos para estos cuatro subproblemas diferentes. Se verifica que en casi todos los casos, el DSCA encuentra la solución óptima, pero el  $R^*DSCA$  logra mejores resultados promedio, demostrando un mejor rendimiento general.

Finalmente, para realizar esta comparación de rendimiento, se han considerado aquellos experimentos para los que se tiene la solución óptima  $O_{ts}$ . Entonces, para cada experimento realizado para cada subproblema, en la Tabla 7.8, se calcula la desviación entre el  $\bar{D}$  obtenido por DSCA y por  $R^*DSCA$ , respectivamente, y la solución óptima  $O_{ts}$ .

Analizando los resultados presentados en la Tabla 7.7 y la Tabla 7.8, se puede inferir que el rendimiento de la propuesta del  $R^*DSCA$  es mayor que el DSCA, alcanzando menos desviaciones en 11 de los 13 primeros casos con soluciones óptimas conocidas.

En el caso del subproblema D, para la comparación, se construyeron algunas gráficas que muestran las curvas de convergencia de ambos algoritmos (Figura 7.9) utilizadas para encontrar una solución óptima para este problema planteado. Al

Experimento	Inicio	Estaciones	T	$O_{ts}$	$\bar{D}$	$D^*$	
A	1	fijo	5	30	374.3 m	382.4 m	374.3 m
A	2	fijo	5	50	374.3 m	384.6 m	374.3 m
A	3	random	5	30	374.3 m	388.3 m	374.3 m
A	4	random	5	50	374.3 m	386.6 m	374.3 m
B	1	fijo	5	30	382.1 m	390.4 m	382.1 m
B	2	fijo	5	50	382.1 m	386.6 m	382.1 m
B	3	random	5	30	382.1 m	395.6 m	382.1 m
B	4	random	5	50	382.1 m	388.5 m	382.1 m
C	1	random	7	30	504.1 m	545.4 m	517.3 m
C	2	random	7	50	504.1 m	533.1 m	504.1 m
C	3	random	7	100	504.1 m	539.5 m	504.1 m
C	4	random	7	200	504.1 m	518.6 m	504.1 m
C	5	random	7	500	504.1 m	508.1 m	504.1 m
D	1	random	10	50	-	744.7 m	678.4 m
D	2	random	10	100	-	708.6 m	659.2 m
D	3	random	10	150	-	691.3 m	659.7 m
D	4	random	10	200	-	669.3 m	631.4 m
D	5	random	10	500	-	623.3 m	571.5 m

Tabla 7.7: Experimentos del DSCA para los cuatro sub-problemas.

tener un inicio aleatorio del algoritmo, existen diferentes puntos de partida, pero se puede observar una convergencia a distancias más pequeñas en  $R^*DSCA$  en todas las figuras.

Experimento	Error del DSCA	Error del $R^*DSCA$
A1	2.175 %	3.295 %
A2	2.766 %	1.695 %
A3	3.730 %	2.860 %
A4	3.295 %	0.435 %
B1	2.156 %	1.182 %
B2	1.176 %	0.001 %
B3	3.525 %	0.491 %
B4	1.664 %	0.004 %
C1	8.178 %	5.962 %
C2	5.737 %	4.243 %
C3	7.018 %	2.600 %
C4	2.862 %	2.161 %
C5	0.785 %	1.047 %

Tabla 7.8: Comparación entre DSCA y  $R^*DSCA$ .

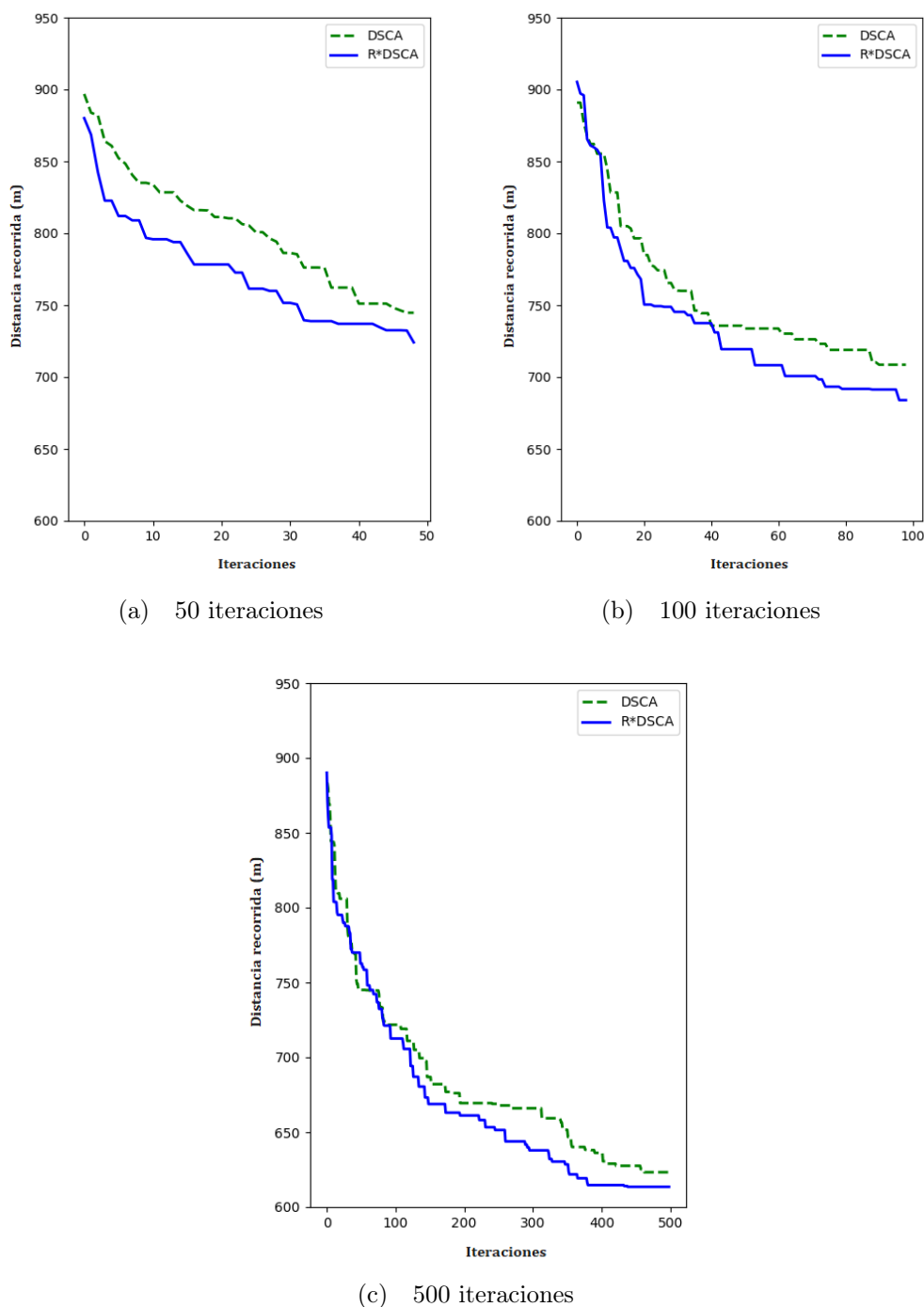


Figura 7.9: Curvas de convergencia  $\bar{D}_t$  para el sub-problema D.

### 7.4.3 Evaluación del $R^*DSCA$ para la reducción de desperdicios

Resumiendo hasta este punto, se han probado los algoritmos en diferentes subproblemas con diferentes configuraciones y verificado su efectividad. Se ha demostrado que el algoritmo desarrollado  $R^*DSCA$  tiene el mejor rendimiento. Ahora comienza otra etapa de pruebas, donde se realizan experimentos para el sistema completo de 20 estaciones (Figura 7.2). Por lo tanto, este problema es aún

más complejo que los anteriores y tiene demasiadas soluciones factibles. Además, se comprobará que la propuesta desarrollada no sólo optimiza las distancias, sino que también contribuye a reducir otros residuos Lean, lo cual colabora a reducir costos, mejorar la calidad y aumentar la eficiencia global del sistema productivo.

Para esta etapa, se prueban varias reglas basadas en diferentes secuencias de rutas de transporte que las organizaciones pueden usar en sus operaciones diarias contra el desempeño del *R\*DSCA*. El objetivo será verificar si la optimización lograda por el *R\*DSCA* del manejo de los materiales impacta en los desperdicios Lean del sistema, específicamente en transporte, esperas e inventarios, lo cual se comprobará con los resultados finales que arroja el simulador.

Los primeros tres casos implican que la empresa no tiene un sistema de optimización incorporado, por lo que el transporte se dirige en un orden determinado para cumplir la ruta requerida siguiendo una regla. En el **caso 1**, el orden de la secuencia o regla de programación es aleatorio. Este sería el caso de una organización que no prevé ninguna planificación para el transporte de materiales por falta de conocimientos o recursos. En el **caso 2**, el vehículo se dirige en orden numérico ascendente a las estaciones, siguiendo el trazado del layout. En el **caso 3**, el transporte mantiene las prioridades dadas por la parte inicial del sistema (esta parte se menciona y explica en la Sección 7.1). Este sería el escenario con más lógica para el manejo de materiales porque considera el estado de las estaciones y sus buffers al momento de armar la ruta. Finalmente, el **caso 4** es la aplicación de la propuesta del *R\*DSCA* con  $T=100$ .

El límite del número de puestos de trabajo que debe cubrir el transporte en un mismo viaje es de diez; y en todos los casos se parte de una misma situación inicial de la simulación general para poder hacer una comparación apropiada.

Se realizan diez repeticiones de los casos 1 y 4, que contienen aleatoriedad. Los casos 2 y 3 son fijos para una situación inicial dada. Se verifica en el software TPS que las distancias recorridas por el vehículo, el tiempo que deben esperar las estaciones por las piezas y el stock remanente al finalizar las simulaciones son menores con el algoritmo *R\*DSCA* desarrollado. Por lo tanto, es posible reducir tres desperdicios importantes en cualquier sistema de producción, impactando significativamente en la eficiencia, en los costos y la productividad empresarial.

Las Tablas 7.9 y 7.10 muestran los mejores resultados obtenidos para cada caso. El período de producción simulado en la primera tabla es de 2 horas para cada prueba, mientras que en la segunda tabla es de 8 horas. Como aclaración, la última columna de estas tablas muestra el tiempo de espera de la estación del sistema que más tuvo que esperar de las veinte estaciones que lo componen.

Las Tablas 7.11 y 7.12 muestran los porcentajes de reducción de desperdicios Lean al comparar los primeros tres casos con nuestra propuesta *R\*DSCA*. En unas pocas pruebas y con un pequeño número de iteraciones del algoritmo ( $T=100$ ), la minimización de dicho desperdicio es significativa en ambos casos.

Finalmente, para concluir este apartado y los experimentos realizados, se probó la simulación del período general de producción de 8 horas con las cuatro reglas mencionadas, pero ya sin restricciones en el número de estaciones que puede cubrir el transporte en un solo recorrido. Se supone hipotéticamente que el transporte posee la capacidad de carga para cubrir las 20 estaciones si es necesario, es decir, no existen restricciones de capacidad de carga. En este escenario, la aplicación del *R\*DSCA* se realiza con  $T=200$ . Los resultados se muestran en las Tablas 7.13 y 7.14. Una

	Distancia total recorrida	Stock final total	Máxima espera en estación
Regla 1	3709.01 m	681 u	11.85 %
Regla 2	3453.09 m	703 u	11.93 %
Regla 3	3670.53 m	681 u	11.36 %
$R^*DSCA$	2923.67 m	660 u	9.46 %

Tabla 7.9: Resultados para un período de producción de 2 horas con  $T=100$ .

	Distancia total recorrida	Stock final total	Máxima espera en estación
Regla 1	18422.83 m	1064 u	16.39 %
Regla 2	15907.94 m	831 u	15.45 %
Regla 3	18160.87 m	1072 u	10.45 %
$R^*DSCA$	15902.96 m	725 u	8.50 %

Tabla 7.10: Resultados para un período de producción de 8 horas con  $T=100$ .

	Transporte	Stock	Espera
Regla 1 vs. $R^*DSCA$	↓ 21.17 %	↓ 3.08 %	↓ 20.17 %
Regla 2 vs. $R^*DSCA$	↓ 15.33 %	↓ 6.12 %	↓ 20.67 %
Regla 3 vs. $R^*DSCA$	↓ 20.35 %	↓ 3.08 %	↓ 16.70 %

Tabla 7.11: Reducción de desperdicios para un período de producción de 2 horas con  $T=100$ .

	Transporte	Stock	Espera
Regla 1 vs. $R^*DSCA$	↓ 13.68 %	↓ 31.86 %	↓ 48.15 %
Regla 2 vs. $R^*DSCA$	v 0.03 %	v 12.76 %	↓ 45.00 %
Regla 3 vs. $R^*DSCA$	↓ 12.43 %	↓ 32.37 %	↓ 18.63 %

Tabla 7.12: Reducción de desperdicios para un período de producción de 8 horas con  $T=100$ .

vez más se confirma un alto rendimiento del algoritmo y una gran reducción de desperdicios, principalmente en el tiempo de espera de las estaciones, lo cual es muy beneficioso para el flujo de trabajo del proceso y la productividad final.

	Distancia total recorrida	Stock final total	Máxima espera en estación
Regla 1	18728.23 m	808 u	29.33 %
Regla 2	16248.55 m	726 u	17.88 %
Regla 3	19397.50 m	1014 u	29.05 %
$R^*DSCA$	15782.52 m	705 u	11.74 %

Tabla 7.13: Resultados para un período de producción de 8 horas con  $T=200$ .

	Transporte	Stock	Espera
Regla 1 vs. $R^*DSCA$	↓ 15.73 %	↓ 12.75 %	↓ 59.97 %
Regla 2 vs. $R^*DSCA$	↓ 2.87 %	↓ 2.89 %	↓ 34.34 %
Regla 3 vs. $R^*DSCA$	↓ 18.64 %	↓ 30.47 %	↓ 59.59 %

Tabla 7.14: Reducción de desperdicios para un período de producción de 8 horas con  $T=200$ .

## 7.5 Conclusiones del capítulo

La principal contribución mostrada en este capítulo es el desarrollo de un algoritmo de optimización para abordar problemas que involucran variables de decisión discretas. En términos generales, la propuesta consiste en una formulación discreta del Algoritmo Seno Coseno con un algoritmo de AR embebido que permite aprender la ley de comportamiento para los parámetros de exploración/explotación.

Se abordó un problema complejo para evaluar en profundidad el desempeño del algoritmo. Además, a partir de dicho problema, se plantearon subproblemas más manejables para probar y comparar rigurosamente la propuesta con otras metodologías, obteniendo resultados exitosos.

Las nuevas tendencias en los sectores productivos reclaman desarrollos que apoyen a los sistemas de producción flexibles impulsados por la demanda de productos personalizados. De hecho, el paradigma Lean 4.0 puede contribuir significativamente a este requerimiento al promover la combinación del concepto de aprovechamiento de recursos y reducción de desperdicios con tecnologías 4.0. En este sentido, la propuesta fue concebida principalmente para contribuir a esta problemática, ya que fue planteada para abordar el problema del manejo de materiales para sistemas de manufactura flexible. Como se enfatizó previamente a lo largo de la tesis, el manejo de materiales es una de las tareas que causan la ineficiencia más significativa en un proceso de producción. El tiempo invertido en mover materiales dentro de un taller es una tarea sin valor agregado en el producto final.

Asimismo, el uso de un simulador como TPS nos permite estudiar el sistema de producción completo y monitorear los desperdicios Lean asociados al manejo de materiales. Se determinó que la propuesta contribuye a reducir tres desperdicios: transporte, inventario y esperas.

En definitiva, la solución propuesta es escalable y adaptable a diferentes configuraciones de manufactura, facilitando la toma de decisiones en entornos dinámicos con alta variabilidad en la demanda. Si bien los métodos tradicionales pueden funcionar para problemas más simples, la combinación de metaheurísticas con inteligencia artificial permite abordar problemas más complejos y dinámicos, que las técnicas convencionales no pueden resolver de manera eficiente. Además, aunque cada industria tiene desafíos específicos, la metodología es modular y adaptable. Se pueden ajustar los parámetros y modelos para diferentes tipos de manufactura, desde producción en masa hasta manufactura flexible y personalizada.

Como trabajo posterior, sería interesante evaluar el desempeño de la propuesta para otro tipo de problemas de optimización relacionados con la filosofía Lean 4.0,

como la minimización de los impactos ambientales de las actividades de manufactura en plantas flexibles.

## Parte III

# Conclusiones y futuros desarrollos



# Capítulo 8

## Las PyMEs y la transición digital 4.0

En una etapa avanzada en el desarrollo del trabajo de tesis, y en el marco del proyecto “Relevamiento de capacidades tecnológicas para la transición digital de PyMEs industriales del sector Metalmeccánico” financiado por CONICET, un grupo de investigadores de la Facultad de Ingeniería de la UNICEN realizamos un estudio minucioso sobre la realidad tecnológica de las PyMEs metalmeccánicas de Olavarría. En este relevamiento se pudo comprobar que es factible aprovechar los desarrollos de esta tesis en la transición digital de pequeñas y medianas organizaciones de la región, tal el objetivo inicial planteado al comienzo del Doctorado.

El estudio se propuso evaluar el nivel de preparación de las empresas del sector metalmeccánico de Olavarría para la adaptación al emergente paradigma de las “*fábricas inteligentes*”. A partir de los resultados obtenidos, se evidencia que este sector se encuentra en una fase incipiente de transformación digital, caracterizada por una serie de desafíos estructurales y tecnológicos. Entre los principales obstáculos identificados se destacan la brecha tecnológica existente, el retraso en la interconexión y comunicación entre máquinas, la limitada implementación de tecnologías de ciberseguridad y automatización, así como la escasez de talento especializado en estas áreas. Estos factores representan barreras significativas para la digitalización efectiva del sector. Sin embargo, también se ha identificado un creciente interés por parte de los empresarios en la adopción de herramientas digitales como la automatización de procesos, el uso de IA y el análisis de datos, reconociendo el impacto positivo que estas tecnologías pueden tener en la eficiencia y productividad de las empresas [Roark et al., 2024c].

Según el relevamiento realizado, si bien las empresas enfrentan dificultades como la falta de conocimiento, inversión y tiempo, se observa una clara disposición hacia la transformación digital de procesos. Esta apertura a la adopción de tecnologías emergentes indica una voluntad de superar los desafíos existentes y avanzar hacia una manufactura más inteligente. En particular, la percepción positiva sobre el impacto de la robótica en la eficiencia y calidad de los productos refleja un interés creciente en la modernización de los procesos productivos. La experiencia con beneficios tangibles derivados de estas tecnologías puede actuar como un catalizador para una mayor adopción en el futuro [Roark et al., 2024a].

No obstante, el estudio también revela la ambigüedad y novedad del concepto de I4.0 en el sector, lo que se traduce en una escasez de casos exitosos como referencia. Factores como las prácticas culturales arraigadas, los hábitos organizacionales y la resistencia al cambio representan obstáculos significativos para la incorporación de

nuevas tecnologías. Estos elementos requieren estrategias específicas para fomentar una mentalidad organizacional más abierta a la innovación y a la reconfiguración de procesos. A pesar de que se reconoce el valor de tecnologías avanzadas como la impresión 3D, la IA, la realidad virtual y aumentada, el big data y la analítica de datos, su adopción aún es limitada y enfrenta barreras tanto técnicas como culturales.

La elección de enfocar este estudio en las PyMEs del sector metalmeccánico se fundamenta en su rol estratégico dentro de la economía local, regional y nacional. Este sector no solo representa una contribución significativa, aportando el 29.2% de la masa salarial nacional, sino que también es un motor clave de generación de empleo, con más de 350.000 puestos de trabajo directos y aproximadamente 980.000 empleos indirectos. Con más del 88% de estas empresas clasificadas como PyMEs y superando los 24.000 establecimientos, el sector se consolida como un pilar fundamental de la economía argentina, actuando como proveedor clave en diversas cadenas de valor [ADIMRA, 2019].

El análisis detallado de estas compañías no solo proporciona un diagnóstico preciso sobre su estado actual, sino que también permite identificar tendencias y desafíos extrapolables a otros sectores industriales. De esta manera, los hallazgos obtenidos aportan a la comprensión global de la transformación digital en el tejido empresarial argentino, proporcionando información valiosa para el diseño de estrategias de modernización y adopción tecnológica. A continuación, se presentan de manera resumida los aspectos fundamentales del estudio y los principales resultados obtenidos en relación con la implementación y el impacto de las tecnologías digitales en el sector.

## 8.1 Aspectos fundamentales del relevamiento

Para el trabajo desarrollado, se implementó una metodología diseñada específicamente por el equipo de investigación con el propósito de evaluar las capacidades tecnológicas de las PyMEs industriales metalmeccánicas. Con este fin, se desarrolló un cuestionario estructurado en ocho secciones, compuesto por un total de 51 preguntas cerradas. Este instrumento incluyó el modelo de evaluación InTICs<sup>®</sup> [Mon and Del Giorgio, 2021], una herramienta analítica creada por investigadores de la Universidad Nacional de La Matanza, con quienes se trabajó en conjunto, permitiendo un análisis detallado sobre el uso y adopción de TIC's en las empresas del sector.

El proceso de recolección de datos se llevó a cabo mediante dos modalidades complementarias: formularios en línea a través de Google Forms y entrevistas presenciales realizadas por encuestadores capacitados. Este enfoque mixto permitió garantizar tanto la accesibilidad como la confiabilidad de la información recopilada, asegurando una representación precisa de la situación de las empresas metalmeccánicas en relación con la transformación digital. La capacitación de los encuestadores se diseñó para reducir sesgos en la recopilación de datos y optimizar la calidad de las respuestas obtenidas.

La población objetivo del estudio incluyó empresas de capital privado pertenecientes al sector metalmeccánico, clasificadas como micro, pequeñas o medianas empresas, con un mínimo de cuatro meses de operación y personal registrado en nómina salarial. A partir de criterios predefinidos, se identificaron

43 empresas que cumplieran con estas condiciones. La selección de estas empresas se realizó a través de fuentes oficiales, tales como la Secretaría de Desarrollo Económico de la Municipalidad de Olavarría, la Unión Industrial de Olavarría y la Asociación de Parques Industriales de Olavarría, garantizando la validez y relevancia de la muestra.

Para la determinación del tamaño muestral, se aplicó la técnica de muestreo aleatorio simple, lo que permitió asegurar una representación equitativa de las unidades de análisis dentro del sector. Como resultado de este proceso, se seleccionaron de manera probabilística 33 empresas, proporcionando una base estadísticamente significativa para la evaluación de las capacidades tecnológicas y digitales de la industria metalmeccánica en la región. La aplicación de esta metodología permitió obtener una visión detallada del estado actual de la transformación digital en estas organizaciones, identificando tanto las barreras existentes como las oportunidades para la implementación de tecnologías avanzadas en el sector.

El índice InTICs<sup>®</sup> ha sido ampliamente aplicado en estudios previos y validado en diversas publicaciones de alcance nacional e internacional [Mon and Del Giorgio, 2022, Mon and Del Giorgio, 2023], consolidándose como una herramienta metodológica robusta para la evaluación del grado de digitalización y adopción tecnológica en entornos industriales. Su estructura se basa en la categorización de productos tecnológicos en tres componentes fundamentales: *Software, Hardware e Infraestructura*. Estos componentes han sido diseñados para reflejar las particularidades de cada tipo de tecnología y su impacto en las distintas áreas funcionales de una organización, permitiendo un análisis detallado de la integración tecnológica dentro del sector productivo.

El modelo InTICs<sup>®</sup> clasifica las áreas funcionales de una empresa tipo en seis categorías principales, en función de las actividades que éstas desempeñan dentro de la estructura organizacional. Estas áreas funcionales son: *Logística, Producción, Ventas, Dirección, Contabilidad y Finanzas, e Ingeniería*. Cada una de ellas representa un ámbito clave en el que las TIC's pueden generar mejoras significativas en términos de eficiencia, productividad y competitividad.

La inclusión de estas áreas funcionales en la metodología permite evaluar con precisión el nivel de madurez digital de una empresa, identificando tanto sus fortalezas como las brechas tecnológicas existentes. Además, el enfoque estructurado del índice facilita la comparación entre distintas organizaciones, proporcionando un marco de referencia para el diseño de estrategias de modernización y optimización de procesos a través de la adopción de tecnologías avanzadas. De esta manera, el índice InTICs<sup>®</sup> se presenta como una herramienta clave para la planificación y gestión de la transformación digital en el sector industrial.

Una vez identificados los productos tecnológicos en su intersección con cada área funcional, el índice genera una valoración para cada producto, que resulta de una ponderación entre el tiempo que llevan en el mercado, el aporte de información sensible para la toma de decisiones, la complejidad del problema que resuelve, el impacto en la mejora de procesos y la contribución a una mayor eficiencia en el uso de recursos. El valor total obtenido de esta ponderación se categoriza en tres niveles: Nivel Básico para aquellas empresas que cuentan con tecnología antigua; Nivel Medio para aquellas empresas que tienen tecnología de actualidad media; y Nivel Avanzado para aquellas empresas que tienen tecnología avanzada. Los productos TIC's identificados por el índice son 74 en las diferentes áreas funcionales. No todos

los productos incluidos en el último nivel Avanzado son de última generación, pero sí resultan imprescindibles para la integración y el funcionamiento de los demás. Por lo tanto, el conjunto de productos del nivel Avanzado que define el índice incluye los productos tecnológicos que componen la I4.0.

A su vez, las preguntas del cuestionario fueron distribuidas en diferentes secciones: identificación de la empresa, datos de la persona encuestada, caracterización general de la empresa, capacidades tecnológicas, tecnologías emergentes, recursos humanos: habilidades y formación, y vinculación de la empresa con otras instituciones. En la Tabla 8.1 se presentan las dimensiones de análisis relacionadas con cada uno de estos ejes.

Eje		Dimensiones
1	Estructura organizacional	-Diversidad e inclusión -Arraigo local y experiencia -Tamaño de las empresas -Actividad principal -Figura legal
2	Estrategia y toma de decisiones	-Posicionamiento estratégico -Prioridades competitivas del sector -Comprensión de tecnologías 4.0 -Gestión estratégica de datos para la TDD -Barreras internas para la transformación digital -Desafíos y obstáculos del sector
3	Tecnologías de la información y digitalización	-Gestión de tecnología de la información -Nivel de digitalización de procesos y documentos -Uso de equipos de hardware (InTICs <sup>®</sup> ) -Uso de equipamiento de infraestructura (InTICs <sup>®</sup> ) -Uso de software (InTICs <sup>®</sup> ) -Implementación de tecnologías de la I4.0 -Adopción de soluciones de ciberseguridad -Madurez tecnológica según InTICs <sup>®</sup>
4	Gestión de datos y trazabilidad	-Trazabilidad de productos y control de materias primas -Gestión de datos para mantenimiento de planta y equipos -Gestión de indicadores en procesos productivos
5	Talento humano	-Habilidades digitales críticas -Nivel gral. de formación y habilidades tecnológicas por área -Interés por oferta de capacitación en tecnologías 4.0 -Perfiles con habilidades específicas
6	Vínculos estratégicos	-Soluciones 4.0 de interés para implementar en el sector -Tareas a automatizar en el futuro -Soluciones de IA de interés -Vínculos empresariales: conexiones estratégicas

Tabla 8.1: Ejes de análisis y dimensiones.

Cabe destacar que la evaluación del nivel de madurez tecnológica, que se realiza a través del modelo InTICs<sup>®</sup>, está ubicado en el Eje de análisis 3 de la Tabla 8.1.

## 8.2 Resultados del relevamiento

Los resultados obtenidos luego del proceso de relevamiento muestran que la mayoría de las empresas encuestadas de la región tienen un nivel medio de madurez tecnológica (75.8%), siendo una minoría las pertenecientes al nivel básico, como puede observarse en la Figura 8.1. Igualmente, se encontraron variaciones en el diagnóstico dependiendo de su tamaño y actividad económica. Las microempresas tienden a tener un nivel básico de madurez, mientras que las medianas y grandes empresas muestran una mayor adopción de tecnologías un poco más avanzadas. Esto se detalla en la Figura 8.2. Cabe mencionar que, de la muestra total de empresas encuestadas, la mayoría, un 52 %, fueron microempresas con menos de 15 empleados. Además, la actividad principal relevada fue la fabricación de productos metálicos, con un 60 % de empresas en esta actividad vinculadas principalmente a la elaboración de recipientes, tanques, depósitos de metal, estructuras metálicas, calefactores a leña y camas articuladas.

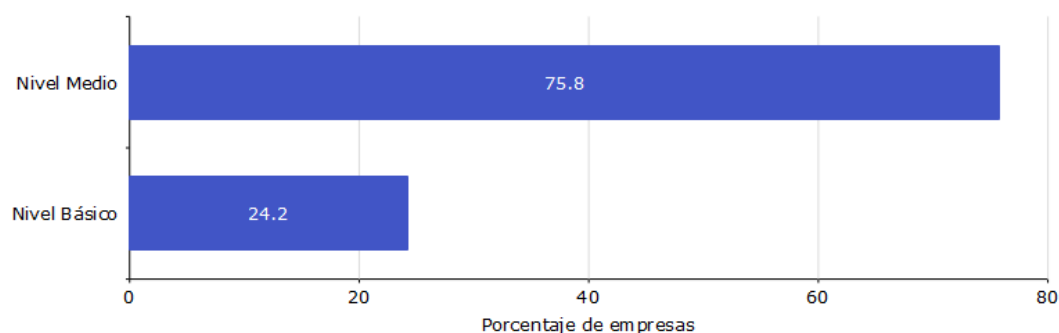


Figura 8.1: Madurez tecnológica del sector metalmeccánico de Olavarría.

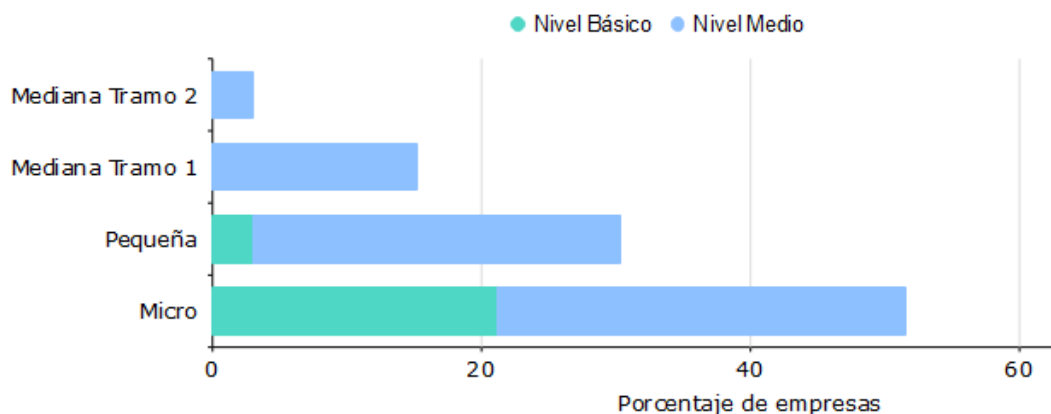


Figura 8.2: Madurez tecnológica de las empresas encuestadas, según su tamaño.

En cuanto a la implementación de productos tecnológicos en cada área funcional, se identificó la utilización de herramientas como impresoras 3D, sistemas avanzados de control en producción y sistemas de geolocalización para estrategias de ventas mediante publicidad digital y realidad virtual. El estudio muestra que entre el 50 % y el 58 % de las compañías analizadas emplean tecnologías básicas en la mayoría de sus áreas funcionales. Sin embargo, las áreas de producción e ingeniería son las que exhiben un mayor grado de sofisticación tecnológica, como se evidencia en la Figura 8.3.

Para profundizar en la aplicación de tecnologías de la I4.0 en el sector metalmecánico de Olavarría, cabe destacar que el 60 % de las empresas utilizan servicios en la nube, además de contar con sistemas de seguridad para datos críticos en más del 50 % de los casos. Respecto al empleo de sistemas de seguridad de infraestructura crítica, se observa una implementación restringida, perceptible solo en el 20 % de las compañías analizadas. Además, se demuestra que la aplicación de sensores ya comprende al 30 % de las compañías estudiadas, mientras que la utilización de robots abarca al 15 % de los participantes en la encuesta.

No obstante, existe una limitada aplicación de tecnologías en auge, como la realidad virtual (6 %), las impresoras 3D (9 %), el big data y la IA (6 %). Esta situación representa tanto un desafío como una oportunidad para la innovación dentro del sector, evidenciando la necesidad de estrategias que fomenten la integración de estas herramientas en los procesos productivos.

En relación con el grado de capacitación y las competencias digitales en las distintas áreas funcionales, el estudio revela una falta de conocimiento por parte de los empresarios sobre las capacidades reales de su personal. Este hallazgo resalta la importancia de desarrollar metodologías específicas para evaluar con precisión las necesidades de formación y actualización tecnológica del talento humano dentro de las empresas. La carencia de programas estructurados de capacitación podría constituir un obstáculo para la transformación digital, lo que refuerza la necesidad de generar iniciativas orientadas a la mejora de las competencias digitales en el sector.

En síntesis, estos hallazgos proporcionan una visión integral sobre el estado actual de la incorporación de tecnologías de la I4.0 en las PyMEs metalmecánicas de Olavarría y su región de influencia. La información obtenida no solo permite identificar los niveles de adopción tecnológica y sus desafíos asociados, sino también comprender la manera en que las empresas y sus colaboradores interactúan con las herramientas digitales.

Finalmente, los resultados de este estudio facilitan la identificación de oportunidades de mejora y la formulación de estrategias que impulsen el desarrollo tecnológico del sector. Además, los datos recopilados poseen un valor estratégico significativo, ya que pueden ser utilizados por autoridades públicas y organismos de apoyo industrial para el diseño de programas de actualización tecnológica y la definición de políticas educativas alineadas con las demandas del nuevo paradigma productivo.

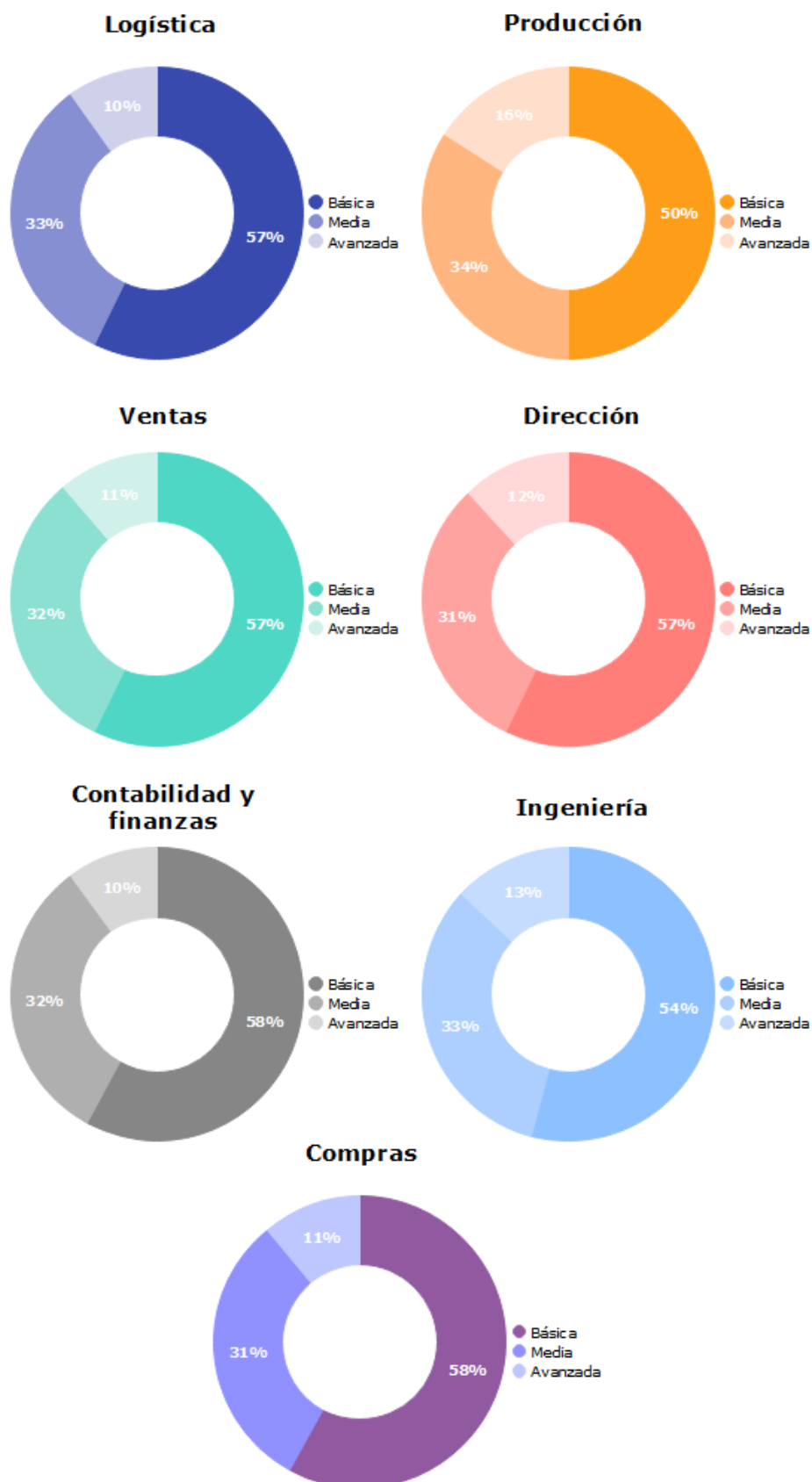


Figura 8.3: Tipos de tecnologías utilizadas por área funcional.

Los resultados detallados eje por eje, junto a un análisis FODA del sector en general y planes de acción recomendados, se pueden encontrar en el Libro “Transformación Digital en PyMEs Metalmecánicas de Olavarría: desafíos y oportunidades”. Geraldina Roark, Carolina Saavedra Sueldo, Ivo Perez Colo, Mariano De Paula y Gerardo G. Acosta. 1a ed - Tandil : Universidad Nacional del Centro de la Provincia de Buenos Aires, 2024. Libro digital <sup>1</sup>, PDF. ISBN 978-950-658-620-1.

## 8.3 Conclusiones del capítulo

La I4.0, que ha sido adoptada por grandes empresas y países desarrollados desde hace varios años, ha comenzado a implementarse de manera gradual en nuestro país. Pero si hablamos de PyMEs, este camino es aún incipiente. En este sentido, para liderar a las empresas, especialmente a las PyMEs, en su transformación hacia el nuevo paradigma digital de la I4.0, es fundamental identificar qué tecnologías son aplicables y cómo se utilizan en los distintos ámbitos para evaluar su situación y desarrollar estrategias que fortalezcan su desarrollo tecnológico.

El relevamiento realizado por el equipo de investigación, presentado en este capítulo, proporciona una metodología estandarizada para evaluar de manera objetiva el nivel de desarrollo tecnológico de las empresas. Este enfoque contribuye en dos aspectos principales. En primer lugar, facilita la identificación de los productos tecnológicos adoptados por las empresas en cada área funcional, donde los usuarios interactúan con estas tecnologías para desempeñar sus funciones. En segundo lugar, permite validar la metodología y mejorar la herramienta de aplicación, incrementando su reproducibilidad.

Como trabajo futuro, sería conveniente replicar el estudio incluyendo un mayor número de empresas industriales de diversas regiones productivas del país, a través de un muestreo por conglomerados. Además, considerar la inclusión de otros sectores productivos relevantes como el minero, el alimenticio y el textil, entre otros. Esta ampliación permitiría obtener mayor información sobre la adopción de tecnologías de la I4.0 en Argentina.

En definitiva, el trabajo realizado durante los años 2023 y 2024 en relación a las PyMEs industriales de la región, que comprende no solo el relevamiento mostrado en este capítulo, sino también varios encuentros presenciales con empresarios, talleres, capacitaciones y reuniones, además de desarrollos tecnológicos puntuales en empresas locales, demuestra que los hallazgos de esta tesis son transferibles a estas organizaciones. Además, se evidencia la necesidad de una mayor colaboración entre la Universidad y la Industria para alcanzar resultados más significativos.

En este sentido, los resultados del estudio y las propuestas desarrolladas en este trabajo tienen un punto de encuentro común. Esto es, la incorporación de tecnologías de código abierto y simuladores con licencias libres (adecuados para aplicaciones a pequeña escala). Estas herramientas ofrecen un marco accesible y adaptable que permite transferir los desarrollos realizados, justamente, hacia la planificación y optimización del movimiento de partes en la industria metalmecánica local. Esta necesidad ha sido destacada de manera reiterada por los empresarios del sector, quienes enfrentan desafíos recurrentes asociados a la falta de organización logística

---

<sup>1</sup><https://www.ridaa.unicen.edu.ar/handle/123456789/3960>

y a la manipulación de cargas pesadas (actividades que consumen tiempo y esfuerzo sin agregar valor al producto final). La articulación de estas tecnologías responde, por tanto, a una necesidad concreta manifestada por las PyMEs locales, al tiempo que sienta las bases para una colaboración sostenida entre la Universidad y la Industria en pos de soluciones innovadoras.

El impacto potencial de Lean 4.0 y el desarrollo de GDs representa una oportunidad estratégica para transformar el sector manufacturero regional, especialmente en el contexto de las PyMEs metalmecánicas. En este sentido, los GDs actúan como catalizadores de la transformación digital al permitir una representación virtual dinámica de los procesos físicos, facilitando una visión integral del sistema productivo en tiempo real. Esta capacidad resulta fundamental para una toma de decisiones autónoma e informada, contribuyendo a reducir tiempos de inactividad, mejorar la asignación de recursos y adaptar los procesos a las demandas fluctuantes del entorno.

En el caso de las PyMEs locales, donde los recursos suelen ser limitados, el impacto de estas tecnologías puede ser determinante. La adopción de Lean 4.0, potenciada por la incorporación de GDs, permite alcanzar un nuevo umbral de eficiencia operativa sin requerir grandes inversiones en infraestructura. Además, estos enfoques promueven una mayor resiliencia organizacional al dotar a las empresas de mecanismos predictivos y adaptativos ante escenarios adversos. La capacidad de simular, predecir y optimizar el comportamiento del sistema productivo antes de intervenir físicamente sobre él, representa una ventaja competitiva sustancial que puede reducir errores y mejorar la calidad de los productos. En consecuencia, la implementación de esta sinergia tecnológica se proyecta como un factor clave para incrementar la competitividad del tejido industrial regional y favorecer su sostenibilidad a largo plazo.



# Capítulo 9

## Conclusiones de la tesis

El presente capítulo constituye el cierre de la investigación y ofrece una síntesis de los principales aportes y contribuciones desarrolladas a lo largo del estudio, en función de los objetivos planteados inicialmente. Además, brinda posibles líneas de trabajo futuras en la temática puntual y en otras áreas de aplicación que se quieren abordar.

### 9.1 Contribuciones y resultados clave

Uno de los principales aportes de este estudio es el desarrollo de una arquitectura de software innovadora para la creación de gemelos digitales en entornos de manufactura flexible. Esta propuesta se basa en la integración del sistema operativo de robots ROS y el simulador de procesos de manufactura TPS, permitiendo una interconexión fluida y una interacción eficiente entre los diferentes componentes del sistema. A través del intercambio de datos e información, la arquitectura facilita la comunicación entre hardware y software, asegurando la interoperabilidad de los elementos que conforman un sistema de fabricación robotizado flexible.

Otro aporte significativo de esta investigación es el desarrollo de una metodología sistemática para la generación de un gemelo digital de un sistema de manufactura y su integración con un sistema autónomo de toma de decisiones (SATDD). Este enfoque proporciona un entorno digital conectado de manera natural con la planta de manufactura real mediante ROS, lo que permite monitorear, controlar y optimizar los procesos productivos en tiempo real. La capacidad de esta metodología para replicar con precisión el comportamiento del sistema físico representa un avance clave en la implementación de tecnologías de manufactura inteligente.

Adicionalmente, se implementó un sistema autónomo de toma de decisiones basado en el paradigma del aprendizaje por refuerzo (AR) para abordar el problema del manejo de materiales en el piso de planta. Se utilizaron tres formulaciones diferentes de algoritmos de AR (*Q-learning*, SARSA y DQN) con el objetivo de comparar su desempeño en términos de eficiencia y adaptabilidad. Los resultados obtenidos en los experimentos demostraron la capacidad de integración de la propuesta en un caso de estudio, al mismo tiempo que se estableció un proceso sistemático para la definición del problema en el contexto del AR, especificando el espacio de estados y acciones. Como se discutió en capítulos previos, la optimización del manejo de materiales en una planta productiva es esencial, dado que esta tarea no agrega valor directo, pero puede representar una mejora significativa en la eficiencia

global del sistema cuando se optimiza adecuadamente.

Además, en un segundo caso de estudio, se exploró el uso de algoritmos metaheurísticos de optimización para la reducción de desperdicios en el problema de manejo de materiales, enmarcado en la filosofía Lean 4.0. Como resultado de esta investigación, se desarrolló una versión mejorada del algoritmo basado en Seno Coseno, el  $R^*DSCA$ , demostrando su efectividad en la optimización de rutas de transporte y reducción de tiempos improductivos. Para evaluar rigurosamente la propuesta, se desglosó el problema en subproblemas más manejables, lo que permitió comparar los resultados obtenidos con otras metodologías existentes. Las pruebas realizadas confirmaron que el algoritmo propuesto ofrece mejoras significativas en la eficiencia del flujo de materiales, especialmente frente a problemas de grandes dimensiones.

Los hallazgos de ambos casos de estudio resaltan el potencial de la metodología propuesta para el desarrollo de gemelos digitales, especialmente en sistemas de producción que incorporan algún sistema robótico, así como su integración con un sistema de TDD. Es importante destacar que, aunque en este trabajo se implementaron algoritmos de AR y metaheurísticos, la arquitectura desarrollada es lo suficientemente flexible para incorporar otros enfoques de toma de decisiones, como sistemas expertos, lógica difusa, redes bayesianas o modelos híbridos. Asimismo, la inclusión de algoritmos para la optimización de hiperparámetros representa una posible línea de investigación futura con aplicaciones de alto impacto en la mejora de estos sistemas. En este sentido, la arquitectura desarrollada permite integrar con facilidad un módulo para optimizar o ajustar los hiperparámetros.

Finalmente, se presentaron trabajos del equipo de investigación relacionados con la transición digital de las PyMEs industriales de la región [Roark et al., 2024a, Roark et al., 2024b, Roark et al., 2024c, Mon et al., 2024], demostrando que las herramientas y metodologías desarrolladas en esta investigación son completamente transferibles a este tipo de organizaciones. Esto refuerza la aplicabilidad de la propuesta en un contexto real y resalta su potencial para contribuir al avance tecnológico de las pequeñas y medianas empresas en los próximos años, fortaleciendo su competitividad y sostenibilidad en el mercado global.

## 9.2 Propuestas para trabajo futuro

Dentro de los paradigmas de las Industrias 4.0 y 5.0 se ha avanzado muchísimo en los últimos años en diversos desarrollos teóricos, metodológicos y prácticos, y la investigación científica ha sido numerosa. Lo mismo sucede con la IA. El campo sufrió una explosión respecto a la cantidad y calidad de trabajos y la incidencia en la vida diaria de la IA en personas y empresas ha sido exponencial. Por lo tanto, hay diversas áreas y aspectos para seguir explorando respecto a la optimización de procesos industriales utilizando técnicas y herramientas actuales.

Una de estas áreas se relaciona con el desarrollo de sistemas multiagente inteligentes para la industria. Un Sistema Multiagente (SMA) consta de múltiples agentes que interactúan en un entorno compartido y toman sus propias decisiones para lograr objetivos comunes o conflictivos. En este sentido, el enfoque de SMA podría transformar la industria manufacturera dado que permitiría otorgar flexibilidad, mejorar la eficiencia y la capacidad de respuesta en entornos de producción complejos, sujetos a condiciones de operación cambiantes y con múltiples

entidades realizando diferentes tareas en el sistema productivo. Un SMA inteligente está compuesto por múltiples agentes o entidades autónomas e inteligentes que cooperan, compiten o se coordinan entre sí para realizar una tarea respondiendo a uno o más objetivos específicos, como por ejemplo optimizar el tiempo de proceso [Leusin et al., 2018], gestionar el mantenimiento predictivo y/o supervisar la cadena de suministro en tiempo real en un sistema productivo. Por otra parte, los SMA favorecen la descentralización y toma de decisiones locales basadas en información dinámica, lo que es un aspecto atractivo, por su capacidad de adaptación a cambios y perturbaciones del entorno.

Las nuevas tendencias se centran en la integración de los SMA con tecnologías emergentes como la IA [Mariani and Omicini, 2023, Bahrpeyma and Reichelt, 2022], el IoT y la computación en la nube. En un futuro muy próximo, probablemente el enfoque de SMA será el rector de los desarrollos de la IA distribuida, lo que favorecerá probablemente el desarrollo de sistemas manufactureros autónomos que funcionen de manera completamente automatizada, desde el diseño de productos hasta la logística final. Otro paso clave para el desarrollo de estos tipos de sistemas será la estandarización de protocolos de comunicación entre agentes para mejorar la interoperabilidad y la toma de decisiones en entornos cambiantes.

Otra área de aplicación en tendencia de estos sistemas es la gestión de la energía. Actualmente, la creciente dependencia de las fábricas inteligentes en componentes avanzados que requieren energía eléctrica, sumada a la necesidad de fuentes de energía eficientes, confiables y rentables, ha impulsado la adopción de redes inteligentes capaces de adaptarse y distribuir energía de manera óptima según la demanda. Actualmente existe una tendencia creciente en la transformación en las políticas energéticas que fomentan la producción de energía eléctrica a partir de distintas fuentes. De esta forma, deben existir propuestas que les permitan, por ejemplo, a las PyMEs, integrarse y lograr una gestión eficiente en un mercado de consumidores-productores. En este contexto, el aprendizaje por refuerzos multiagente ha sido aplicado para mejorar la gestión, eficiencia e integración de agentes en estas redes. Tanto las redes inteligentes como los sistemas de manufactura inteligente comparten características y objetivos fundamentales, como la comunicación en tiempo real, la integración de tecnologías avanzadas y la automatización de procesos, lo que permite desarrollar soluciones conjuntas que optimicen su rendimiento y sostenibilidad [Bahrpeyma and Reichelt, 2022].

Sin embargo, a pesar de las prometedoras aplicaciones de los SMA, aún existen desafíos en su adopción generalizada. En [Leitao and Strasser, 2016] ya enfatizaban la necesidad crítica de estandarización en tecnologías SMA para alinearse con las prácticas industriales existentes y facilitar una implementación más amplia. Además, en [Gorodetsky et al., 2020] señalan que, si bien los SMA son reconocidos por su potencial en el diseño de sistemas adaptativos complejos, su aplicación práctica en la industria aún es limitada, lo que requiere una mayor exploración y desarrollo. La brecha existente en la aplicación práctica de estos conocimientos se evidencia aún más en las PyMEs, actores clave del sistema productivo nacional y regional. Por lo tanto, repensar estas organizaciones desde la perspectiva de los SMA podría proporcionar ventajas significativas, optimizando su capacidad de respuesta y sostenibilidad en el tiempo.

Por otro lado, una línea de investigación prometedora para futuros trabajos es la automatización parcial o total en la generación de los GDs. La simulación semi-

automática de GDs es un campo en constante evolución que busca integrar diversas tecnologías para mejorar la eficiencia, precisión y escalabilidad de estos sistemas. Este enfoque combina procesos automatizados con una mínima intervención humana, lo que facilita la creación de modelos digitales que representan fielmente los sistemas físicos y permiten una adaptación ágil a entornos dinámicos.

El desarrollo de GDs totalmente automatizados en entornos industriales representa un avance significativo dentro del paradigma de la I4.0, ya que posibilita la creación, actualización y operación de modelos virtuales sin intervención humana directa. La automatización en este ámbito abarca desde la adquisición y procesamiento de datos hasta la actualización en tiempo real del modelo digital, proporcionando una representación más precisa y dinámica del sistema físico [Tao et al., 2019].

Herramientas avanzadas como el modelado basado en datos y los algoritmos de optimización desempeñan un papel clave en la reducción de errores de interpretación y en la aceleración del despliegue de los GDs en aplicaciones industriales [Negri et al., 2017]. Un área de desarrollo relevante en esta dirección es la incorporación de redes neuronales profundas para el análisis predictivo y la calibración automática de parámetros, lo que permitiría mejorar la capacidad de adaptación de los modelos digitales [Rasheed et al., 2020]. En entornos de manufactura inteligente, la simulación automática de GDs se ha aplicado con éxito en la predicción de fallos en maquinaria, utilizando flujos de datos continuos provenientes de sensores [Liu et al., 2021].

A pesar de sus múltiples ventajas, la automatización completa en la simulación de GDs enfrenta importantes desafíos técnicos. Entre ellos, se encuentra la necesidad de grandes volúmenes de datos de entrenamiento, la gestión de incertidumbre en entornos complejos y la dependencia de infraestructuras robustas para garantizar la confiabilidad de los modelos [Fuller et al., 2020]. Además, la ciberseguridad es un aspecto crucial que debe abordarse para proteger la integridad de los datos y evitar vulnerabilidades en los sistemas industriales interconectados.

Las futuras investigaciones podrían enfocarse en mejorar la robustez de los algoritmos, así como en el desarrollo de marcos de validación estandarizados que permitan garantizar la confiabilidad de los modelos [Madni et al., 2019]. Asimismo, se requieren avances en protocolos de certificación autónoma y en la implementación de esquemas de auto-aprendizaje que faciliten la adopción de GDs en sectores estratégicos y a gran escala.

En conclusión, la automatización en la generación de GDs está transformando la forma en que estos modelos son diseñados y utilizados, generando oportunidades significativas en diversas industrias, como la manufactura, la salud y la gestión urbana. No obstante, su éxito dependerá de la superación de desafíos técnicos y de la integración de enfoques multidisciplinarios que permitan desarrollar sistemas confiables, seguros y adaptativos en entornos industriales altamente dinámicos.

Es importante resaltar que, en ambas áreas mencionadas, tanto los SMA para la industria como la automatización parcial o total de los GDs, el enfoque propuesto sigue centrándose en las PyMEs locales y regionales, ya que son un factor clave en el tejido industrial de la zona. En este sentido, la implementación de estas tecnologías en sus procesos es una oportunidad estratégica para transformar su competitividad y brindarles nuevas habilidades para afrontar los cambios en el entorno.

# Apéndice A

## Arquitectura software: Detalle de códigos

- Clase Interfaz de conexión:

```
import os
import pathlib
from functools import wraps
import attr
import win32com.client as win32

PATH = pathlib.Path(os.path.abspath(os.path.dirname(__file__)))

class ConnectionError(Exception):
    """Connection failed exception."""
    pass

class ModelNotFoundError(FileNotFoundError):
    """Custom error for Not found Model."""
    pass

def check_connection(method):
    """Check the connection status, returning an error."""

    @wraps(method)
    def wrapper(self, *args, **kwargs):
        if not self.is_connected:
            raise ConnectionError("Not connected", self.model_name)
        output = method(self, *args, **kwargs)
        return output

    return wrapper

@attr.s
class CommunicationInterface(object):
    """Definition of the function of communication."""
```

---

```

model_name = attr.ib()
is_connected = attr.ib(default=False)
plant_simulation = attr.ib(default="")

def get_path_file_model(self):
    """Return the complete file path."""
    path = str(PATH / self.model_name)
    if not os.path.exists(path):
        raise ModelNotFoundError(
            f"Model {self.model_name} does not exists"
        )
    return path

def connection(self):
    """Return the connection object."""
    path_file = self.get_path_file_model()
    self.plant_simulation = win32.Dispatch(
        "Tecnomatix.PlantSimulation.RemoteControl"
    )
    self.plant_simulation.loadModel(path_file)
    print("The connection was successful")
    self.is_connected = True
    return True

@check_connection
def setvisible(self, value):
    """Execute the application Tecnomatix."""
    self.plant_simulation.setVisible(value)

@check_connection
def setvalue(self, ref, value):
    """Set the values in the simulator."""
    self.plant_simulation.setValue(ref, value)

@check_connection
def getvalue(self, ref):
    """Get the values of the simulator."""
    return self.plant_simulation.getValue(ref)

@check_connection
def startsimulation(self, ref):
    """Make the simulation start."""
    self.plant_simulation.startSimulation(ref)

@check_connection
def resetsimulation(self, ref):
    """Make the simulation reset."""
    self.plant_simulation.resetSimulation(ref)

```

---

```

@check_connection
def stopsimulation(self, ref):
    """Make the simulation stop."""
    self.plant_simulation.stopSimulation(ref)

@check_connection
def closemodel(self):
    """Close the simulation model."""
    self.plant_simulation.CloseModel()

@check_connection
def execute_simtalk(self, ref, value):
    """Execute the simulation programming language."""
    self.plant_simulation.ExecuteSimTalk(ref, value)

@check_connection
def is_simulation_running(self):
    """Check if the simulation is running."""
    return self.plant_simulation.IsSimulationRunning()

@check_connection
def loadmodel(self, ref, value):
    """Perform the load of the model."""
    self.plant_simulation.LoadModel(ref, value)

@check_connection
def newmodel(self):
    """Create a new model."""
    self.plant_simulation.NewModel()

@check_connection
def openconsole_logfile(self, ref):
    """Open the simulation result in the console."""
    self.plant_simulation.OpenConsoleLogFile(ref)

@check_connection
def quit(self):
    """Clear all result."""
    self.plant_simulation.Quit()

@check_connection
def quit_vertime(self, value):
    """Clear all result after a time."""
    self.plant_simulation.QuitAfterTime(value)

@check_connection
def savemodel(self, ref):

```

---

```

        """Save the model result."""
        self.plant_simulation.SaveModel(ref)

    @check_connection
    def set_licensetype(self, ref):
        """Set the type of the license."""
        self.plant_simulation.SetLicenseType(ref)

    @check_connection
    def set_no_messagebox(self, value):
        """Delete the messages on the screen."""
        self.plant_simulation.SetNoMessageBox(value)

    @check_connection
    def set_pathcontext(self, ref):
        """Set the context."""
        self.plant_simulation.SetPathContext(ref)

    @check_connection
    def set_suppress_start_of_3d(self, value):
        """Eliminate the start of 3D model."""
        self.plant_simulation.SetSuppressStartOf3D(value)

    @check_connection
    def set_trustmodels(self, value):
        """Set the real model."""
        self.plant_simulation.SetTrustModels(value)

    @check_connection
    def transfermodel(self, value):
        """Transfer the model."""
        self.plant_simulation.TransferModel(value)

```

#### - Class Abstracta Planta:

```

from abc import ABCMeta, abstractmethod
import attr
import numpy as np
from interface import CommunicationInterface

@attr.s
class Plant(metaclass=ABCMeta):
    """Metaclass to generate various simulated manufacturing plants."""

    method = attr.ib()

    def __attrs_post_init__(self):
        """Attrs initialization."""

```

---

```

        self.method.register(self)

def connection(self):
    """Connect function."""
    file_name = self.get_file_name_plant()
    self.connect = CommunicationInterface(file_name)
    return self.connect.connection()

@abstractmethod
def get_file_name_plant(self):
    """Name of the given plant file."""
    pass

@abstractmethod
def process_simulation(self):
    """Simulate in Tecnomatix."""
    pass

@abstractmethod
def update(self, data):
    """Update. Simulation data."""
    pass

```

- Clase Abstracta Sistema Autónomo de TDD:

```

@attr.s
class AutonomousDecisionSystem(metaclass=ABCMeta):
    """Autonomous decision system class."""

    method = attr.ib(init=False)

    def __attrs_post_init__(self):
        """Attrs initialization."""
        self.method = ""

    def register(self, who):
        """Subscribe registration."""
        self.subscriber = who

@abstractmethod
def process(self):
    """Process."""
    pass

```

- Prueba de concepto de la arquitectura en la Celda de manufactura:

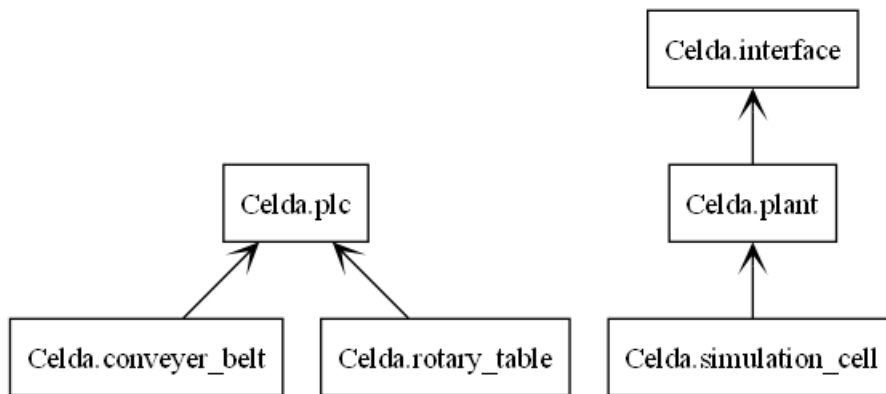


Figura A.1: Paquetes en python de la integración de la prueba.

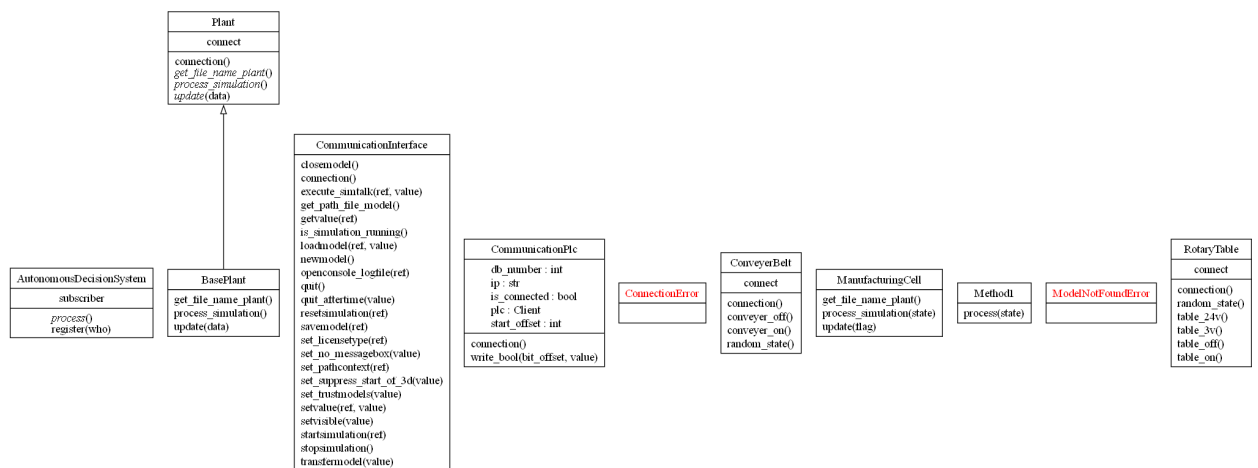


Figura A.2: Diagrama UML de la prueba de concepto.

```

# =====
# Cinta Transportadora:
# =====
@attr.s
class ConveyerBelt(object):
    """Class to turn off and turn on the conveyer belt."""

    def connection(self):
        """Connect function."""
        self.connect = CommunicationPlc()
        return self.connect.connection()

    def conveyer_on(self):
        bit_offset = 0
        value = 1
        self.connect.write_bool(bit_offset, value)
        state = 'ConveyerBelt ON'
  
```

---

```

        print('The conveyer belt is on')
        return state

def conveyer_off(self):
    bit_offset = 0
    value = 0
    self.connect.write_bool(bit_offset, value)
    state = 'ConveyerBelt OFF'
    print('The conveyer belt is off')
    return state

sim = None
conveyer = ConveyerBelt()
conveyer.connection()

def callbacksim(data):
    global sim
    rospy.loginfo(rospy.get_caller_id() + 'I heard %s', data.data)
    try:
        state_str, sent_time = data.data.split('|')
        sent_time = float(sent_time)
        received_time = time.time()
        latency = received_time - sent_time
        latencies_virtual_real.append(latency)
        sim = state_str
    except Exception as e:
        print(f"Error al parsear mensaje (virtual → real): {e}")

def talker_conveyer():
    global sim
    rospy.init_node('talkerconveyer', anonymous=True)
    pub = rospy.Publisher('chatter/real', String, queue_size=10)
    rospy.Subscriber('chatter/virtual', String, callbacksim)
    rate = rospy.Rate(0.1)

    while not rospy.is_shutdown():
        state = conveyer.random_state()
        # Display the message on the terminal
        rospy.loginfo(state)
        # Publish the message to the topic
        timestamp = time.time()
        msg = f"{state}|{timestamp}"
        pub.publish(msg)
        rate.sleep()

# =====
# Celda de Manufactura:
# =====

```

---

```

@attr.s
class ManufacturingCell(Plant):
    """Simulation of the cell."""

    filename = attr.ib()

    def get_file_name_plant(self):
        return self.filename

    def update(self, flag):
        if flag == 1:
            self.connect.setvalue(
                '.Models.Cell.ConveyerBelt1.Failed', False)
            print('Cinta Encendida')
            self.connect.startsimulation('.Models.Cell')
        elif flag == 2:
            self.connect.setvalue(
                '.Models.Cell.ConveyerBelt1.Failed', True)
            print('Cinta Apagada')
        elif flag == 3:
            self.connect.setvalue(
                '.Models.Cell.RotaryTable.Failed', False)
            print('Mesa Encendida')
            self.connect.startsimulation('.Models.Cell')
        elif flag == 4:
            self.connect.setvalue(
                '.Models.Cell.RotaryTable.Failed', True)
            print('Mesa Apagada')
        elif flag == 5:
            print('Mesa Encendida 3V')
        elif flag == 6:
            print('Mesa Encendida 24V')
        else:
            pass

    def process_simulation(self, state):
        if self.connection():
            self.connect.setvisible(True)
            self.connect.startsimulation('.Models.Cell')
            self.method.process(state)

@attr.s
class Method1(AutonomousDecisionSystem):
    """For the Cell class."""

    def process(self, state):
        flag = 0
        if state == 'ConveyerBelt ON':

```

---

```

        flag = 1
    elif state == 'ConveyerBelt OFF':
        flag = 2
    elif state == 'RotaryTable ON':
        flag = 3
    elif state == 'RotaryTable OFF':
        flag = 4
    elif state == 'RotaryTable 3V':
        flag = 5
    elif state == 'RotaryTable 24V':
        flag = 6
    else:
        pass
    self.subscriber.update(flag)

my_filename = 'CeldaFlexible.spp'
method = Method1()
plant = ManufacturingCell(filename=my_filename, method=method)
state = None

# Callback function to print the subscribed data on the terminal and save:
def callback(data):
    global state
    rospy.loginfo(rospy.get_caller_id() + 'I heard %s', data.data)
    try:
        state_str, sent_time = data.data.split('|')
        sent_time = float(sent_time)
        received_time = time.time()
        latency = received_time - sent_time
        print(f"Latencia recibida: {latency:.6f} segundos")

        latencies.append(latency)
        state = state_str
    except Exception as e:
        print(f"Error al parsear mensaje: {e}")

def cell():
    global state
    rospy.init_node('cell', anonymous=True)
    pub = rospy.Publisher('chatter/virtual', String, queue_size=10)
    rospy.Subscriber('chatter/real', String, callback)
    rate = rospy.Rate(0.1)

    while not rospy.is_shutdown():
        plant.method.process(state)
        if plant.connect.is_connected:
            s = 'OK'
        else:

```

---

```
        s = 'Simulation OFF'
        rospy.loginfo(s)
        timestamp = time.time()
        msg = f"{s}|{timestamp}"
        pub.publish(msg)
        rate.sleep()

if __name__ == '__main__':
    try:
        plant.process_simulation(state)
        cell()
        atexit.register(save_latencies)
    except rospy.ROSInterruptException:
        pass
```

# Bibliografía

- [Abdel-Baset et al., 2019] Abdel-Baset, M., Zhou, Y., and Hezam, I. (2019). Use of a sine cosine algorithm combined with simpson method for numerical integration. *International Journal of Mathematics in Operational Research*, 14(3):307–318.
- [Adeodu et al., 2023] Adeodu, A., Maladzhi, R., Katumba, M. G. K.-K., and Daniyan, I. (2023). Development of an improvement framework for warehouse processes using lean six sigma (dmaic) approach. a case of third party logistics (3pl) services. *Heliyon*, 9(4).
- [ADIMRA, 2019] ADIMRA (2019). Guía para una producción sustentable : sector metalmecánico. - 1a ed . Secretaría de Ambiente y Desarrollo Sustentable de la Nación.
- [Agarap, 2018] Agarap, A. F. (2018). Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*.
- [Akhand et al., 2015] Akhand, M. A. H., Peya, Z. J., Sultana, T., and Al-Mahmud (2015). Solving capacitated vehicle routing problem with route optimization using swarm intelligence. In *2015 2nd International Conference on Electrical Information and Communication Technologies (EICT)*, pages 112–117.
- [Akkari and Valamede, 2020] Akkari, A. and Valamede, L. (2020). Lean 4.0: A new holistic approach for the integration of lean manufacturing tools and digital technologies. *International Journal of Mathematical, Engineering and Management Sciences*, 5:851–868.
- [Akman et al., 2023] Akman, G., Yorur, B., Boyaci, A. I., and Chiu, M.-C. (2023). Assessing innovation capabilities of manufacturing companies by combination of unsupervised and supervised machine learning approaches. *Applied Soft Computing*, 147:110735.
- [Al-qaness et al., 2018] Al-qaness, M. A. A., Elaziz, M. A., and Ewees, A. A. (2018). Oil consumption forecasting using optimized adaptive neuro-fuzzy inference system based on sine cosine algorithm. *IEEE Access*, 6:68394–68402.
- [Alkhoraif et al., 2019] Alkhoraif, A., Rashid, H., and McLaughlin, P. (2019). Lean implementation in small and medium enterprises: Literature review. *Operations Research Perspectives*, 6:100089.
- [Amazon Web Services, 2024] Amazon Web Services (2024). ¿qué es mqtt?

- 
- [Amiri et al., 2019] Amiri, F., Shirazi, B., and Tajdin, A. (2019). Multi-objective simulation optimization for uncertain resource assignment and job sequence in automated flexible job shop. *Applied Soft Computing Journal*, 75:190–202.
- [Bahrpeyma and Reichelt, 2022] Bahrpeyma, F. and Reichelt, D. (2022). A review of the applications of multi-agent reinforcement learning in smart factories. *Frontiers in Robotics and AI*, 9.
- [Barata and Kayser, 2023] Barata, J. and Kayser, I. (2023). Industry 5.0 – past, present, and near future. *Procedia Computer Science*, 219:778–788. CENTERIS – International Conference on ENTERprise Information Systems / ProjMAN – International Conference on Project MANagement / HCist – International Conference on Health and Social Care Information Systems and Technologies 2022.
- [Barbieri et al., 2022] Barbieri, G., España, A., and Sanchez-Londoño, D. (2022). A taxonomy for levels of automation based on the industrial revolutions. *IFAC-PapersOnLine*, 55(17):368–373. 13th IFAC Symposium on Advances in Control Education ACE 2022.
- [Barrera-Diaz et al., 2018] Barrera-Diaz, C., Oscarsson, J., Lidberg, S., and Sellgren, T. (2018). Discrete Event Simulation Output Data-Handling System in an Automotive Manufacturing Plant. *Procedia Manufacturing*, 25:23–30.
- [Baskaran et al., 2019] Baskaran, S., Niaki, F. A., Tomaszewski, M., Gill, J. S., Chen, Y., Jia, Y., Mears, L., and Krovi, V. (2019). Digital human and robot simulation in automotive assembly using siemens process simulate: A feasibility study. *Procedia Manufacturing*, 34:986–994.
- [Bass et al., 2013] Bass, L., Clements, P., and Rick, K. (2013). *Software architecture in practice*. Addison-Wesley Professional; 3rd Edición.
- [Ben Moussa et al., 2019] Ben Moussa, F. Z., De Guio, R., Dubois, S., Rasovska, I., and Benmoussa, R. (2019). Study of an innovative method based on complementarity between ariz, lean management and discrete event simulation for solving warehousing problems. *Computers & Industrial Engineering*, 132:124–140.
- [Benitez et al., 2023] Benitez, G. B., Ghezzi, A., and Frank, A. G. (2023). When technologies become industry 4.0 platforms: Defining the role of digital technologies through a boundary-spanning perspective. *International Journal of Production Economics*, 260:108858.
- [Benmessaoud et al., 2021] Benmessaoud, G. A., Yassine, M., Seyedali, M., and Amar, R.-C. (2021). A comprehensive survey of sine cosine algorithm: variants and applications. *Artificial Intelligence Review*, 54:5469–5540.
- [Block et al., 2018] Block, C., Lins, D., and Kuhlenkötter, B. (2018). Approach for a simulation-based and event-driven production planning and control in decentralized manufacturing execution systems. *Procedia CIRP*, 72:1351–1356.

- 
- [Cassandras and Lafortune, 2008] Cassandras, C. G. and Lafortune, S. (2008). *Introduction to discrete event systems*. Springer Science+Business Media.
- [Cheng et al., 2020] Cheng, J., Zhang, H., Tao, F., and Juang, C. F. (2020). DT-II: Digital twin enhanced Industrial Internet reference framework towards smart manufacturing. *Robotics and Computer-Integrated Manufacturing*, 62(January 2019):101881.
- [Chung et al., 2025] Chung, D., Kim, J., Paik, S., Im, S., and Kim, H. (2025). Automated system of scaffold point cloud data acquisition using a robot dog. *Automation in Construction*, 170:105944.
- [Cifone et al., 2021] Cifone, F. D., Hoberg, K., Holweg, M., and Staudacher, A. P. (2021). ‘lean 4.0’: How can digital technologies support lean practices? *International Journal of Production Economics*, 241:108258.
- [Daoui et al., 2021] Daoui, A., Karmouni, H., Sayyouri, M., Qjidaa, H., Maaroufi, M., and Alami, B. (2021). New robust method for image copyright protection using histogram features and sine cosine algorithm. *Expert Systems with Applications*, 177:114978.
- [Das et al., 1998] Das, T., Gosavi, A., Mahadevan, S., and Marchallick, N. (1998). Solving semi-markov decision problems using average reward reinforcement learning. *Management Science*, 45.
- [De Paula and Martínez, 2012] De Paula, M. and Martínez, E. C. (2012). Optimal operation of discretely controlled continuous systems under uncertainty. *Industrial and Engineering Chemistry Research*, 51(42):13743–13764.
- [Dejene et al., 2020] Dejene, D., Tiwari, B., and Tiwari, V. (2020). Td2seciot: Temporal, data-driven and dynamic network layer based security architecture for industrial iot. *International Journal of Interactive Multimedia and Artificial Intelligence*, 6(4):146–156.
- [del Brío and Molina, 2006] del Brío, B. M. and Molina, A. S. (2006). *Redes neuronales y sistemas borrosos : introducción teórica y práctica*. Alfaomega Grupo Editor S.A., 3era edition.
- [Demir et al., 2019] Demir, K. A., Döven, G., and Sezen, B. (2019). Industry 5.0 and Human-Robot Co-working. *Procedia Computer Science*, 158(November):688–695.
- [Dillinger et al., 2022] Dillinger, F., Bernhard, O., and Reinhart, G. (2022). Competence requirements in manufacturing companies in the context of lean 4.0. *Procedia CIRP*, 106:58–63. 9th CIRP Conference on Assembly Technology and Systems.
- [Dillinger et al., 2021] Dillinger, F., Kagerer, M., and Reinhart, G. (2021). Concept for the development of a lean 4.0 reference implementation strategy for manufacturing companies. *Procedia CIRP*, 104:330–335. 54th CIRP CMS 2021 - Towards Digitalized Manufacturing 4.0.

- 
- [Ding et al., 2020] Ding, K., Lei, J., Chan, F. T., Hui, J., Zhang, F., and Wang, Y. (2020). Hidden Markov model-based autonomous manufacturing task orchestration in smart shop floors. *Robotics and Computer-Integrated Manufacturing*, 61(July 2019):1–9.
- [Dockter, 2021] Dockter, D. (2021). The digital twin and real-time adaptive robot control. <https://www.energid.com/blog/the-digital-twin-and-real-time-adaptive-robot-control>.
- [Duhamel et al., 2011] Duhamel, C., Lacomme, P., Quilliot, A., and Toussaint, H. (2011). A multi-start evolutionary local search for the two-dimensional loading capacitated vehicle routing problem. *Computers & Operations Research*, 38(3):617–640.
- [Dündar et al., 2022] Dündar, H., Soysal, M., Ömürgönülşen, M., and Kanellopoulos, A. (2022). A green dynamic tsp with detailed road gradient dependent fuel consumption estimation. *Computers & Industrial Engineering*, 168:108024.
- [Enke et al., 2018] Enke, J., Glass, R., Kreß, A., Hambach, J., Tisch, M., and Metternich, J. (2018). Industrie 4.0 – competencies for a modern production system: A curriculum for learning factories. *Procedia Manufacturing*, 23:267–272. “Advanced Engineering Education & Training for Manufacturing Innovation”8th CIRP Sponsored Conference on Learning Factories (CLF 2018).
- [Fan et al., 2021] Fan, Y., Yang, J., Chen, J., Hu, P., Wang, X., Xu, J., and Zhou, B. (2021). A digital-twin visualized architecture for flexible manufacturing system. *Journal of Manufacturing Systems*, 60:176–201.
- [Flaherty, 2021] Flaherty, N. (2021). Digital twin combines robot control and vr. <https://www.eenewseurope.com/en/digital-twin-combines-robot-control-and-vr/>.
- [Fuller et al., 2020] Fuller, A., Fan, Z., Day, C., and Barlow, C. (2020). Digital twin: Enabling technologies, challenges and open research. *IEEE Access*, 8:108952–108971.
- [Furian et al., 2015] Furian, N., O’Sullivan, M., Walker, C., Vössner, S., and Neubacher, D. (2015). A conceptual modeling framework for discrete event simulation using hierarchical control structures. *Simulation Modelling Practice and Theory*, 56:82–96.
- [Galindo-Salcedo et al., 2022] Galindo-Salcedo, M., Pertúz-Moreno, A., Guzmán-Castillo, S., Gómez-Charris, Y., and Romero-Conrado, A. R. (2022). Smart manufacturing applications for inspection and quality assurance processes. *Procedia Computer Science*, 198:536–541. 12th International Conference on Emerging Ubiquitous Systems and Pervasive Networks / 11th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare.
- [Gambella et al., 2021] Gambella, C., Ghaddar, B., and Naoum-Sawaya, J. (2021). Optimization problems for machine learning: A survey. *European Journal of Operational Research*, 290(3):807–828.

- 
- [Gerodimos et al., 2023] Gerodimos, A., Maglaras, L., Ferrag, M. A., Ayres, N., and Kantzavelou, I. (2023). Iot: Communication protocols and security threats. *Internet of Things and Cyber-Physical Systems*, 3:1–13.
- [Ghodsian et al., 2022] Ghodsian, N., Benfriha, K., Olabi, A., Gopinath, V., Arnou, A., zant, C. E., Charrier, Q., and helou, M. E. (2022). Toward designing an integration architecture for a mobile manipulator in production systems: Industry 4.0. *Procedia CIRP*, 109:443–448. 32nd CIRP Design Conference (CIRP Design 2022) - Design in a changing world.
- [Gola, 2018] Gola, A. (2018). Reliability analysis of reconfigurable manufacturing system structures using computer simulation methods. *Eksploatacja i Niezawodnosc - Maintenance and Reliability*, 21:90–102.
- [Gola and Kłosowski, 2019] Gola, A. and Kłosowski, G. (2019). Development of computer-controlled material handling model by means of fuzzy logic and genetic algorithms. *Neurocomputing*, 338:381–392.
- [Golovianko et al., 2023] Golovianko, M., Terziyan, V., Branytskyi, V., and Malyk, D. (2023). Industry 4.0 vs. industry 5.0: Co-existence, transition, or a hybrid. *Procedia Computer Science*, 217:102–113. 4th International Conference on Industry 4.0 and Smart Manufacturing.
- [Gonzalez, 2020] Gonzalez, C. M. (2020). Robotics blog: Digital twins for robot installations. <https://www.asme.org/topics-resources/content/robotics-blog-digital-twins-for-robot-installations>.
- [González García et al., 2018] González García, C., Núñez-Valdez, E. R., García-Díaz, V., Pelayo G-Bustelo, C., and Cueva Lovelle, J. M. (2018). A Review of Artificial Intelligence in the Internet of Things. *International Journal of Interactive Multimedia and Artificial Intelligence*, 5 (Special Issue on Artificial Intelligence Applications)(4):9–20.
- [Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. Adaptive computation and machine learning. MIT Press.
- [Gorodetsky et al., 2020] Gorodetsky, V., Skobelev, P., and Marik, V. (2020). System engineering view on multi-agent technology for industrial applications: barriers and prospects. *Cybernetics and Physics*, 9(1).
- [Gouveia et al., 2019] Gouveia, L., Leitner, M., and Ruthmair, M. (2019). Layered graph approaches for combinatorial optimization problems. *Computers & Operations Research*, 102:22–38.
- [Govindaiah and Pey, 2019] Govindaiah, S. and Pey, M. D. (2019). Applying reinforcement learning to plan manufacturing material handling Part 1: Background and formal problem specification. *ACMSE 2019 - Proceedings of the 2019 ACM Southeast Conference*, pages 168–171.
- [Groumpos, 2021] Groumpos, P. P. (2021). A critical historical and scientific overview of all industrial revolutions. *IFAC-PapersOnLine*, 54(13):464–471. 20th IFAC Conference on Technology, Culture, and International Stability TECIS 2021.

- 
- [Gupta et al., 2022] Gupta, S., Zhang, Y., and Su, R. (2022). Urban traffic light scheduling for pedestrian–vehicle mixed-flow networks using discrete sine–cosine algorithm and its variants. *Applied Soft Computing*, 120:108656.
- [Haag and Anderl, 2018] Haag, S. and Anderl, R. (2018). Digital twin – Proof of concept. *Manufacturing Letters*, 15:64–66.
- [Hamad et al., 2022] Hamad, Q. S., Samma, H., Suandi, S. A., and Mohamad-Saleh, J. (2022). Q-learning embedded sine cosine algorithm (qlesca). *Expert Systems with Applications*, 193:116417.
- [Hartini and Ciptomulyono, 2015] Hartini, S. and Ciptomulyono, U. (2015). The relationship between lean and sustainable manufacturing on performance: Literature review. *Procedia Manufacturing*, 4:38–45. Industrial Engineering and Service Science 2015, IESS 2015.
- [Haykin, 2009] Haykin, S. S. (2009). *Neural networks and learning machines*. Pearson Education, Upper Saddle River, NJ, third edition.
- [He et al., 2019] He, R., Chen, G., Dong, C., Sun, S., and Shen, X. (2019). Data-driven digital twin technology for optimized control in process systems. *ISA Transactions*, 95:221–234.
- [Heizer and Render, 2004] Heizer, J. and Render, B. (2004). *Principios de Administración de Operaciones*. Prentice Hall, quinta edición edition.
- [Hu et al., 2020] Hu, H., Jia, X., He, Q., Fu, S., and Liu, K. (2020). Deep reinforcement learning based agvs real-time scheduling with mixed rule for flexible shop floor in industry 4.0. *Computers & Industrial Engineering*, 149:106749.
- [IBM, 2018] IBM (2018). ¿qué es la industria 4.0? Accessed: febrero-2024.
- [Illmer and Vielhaber, 2019] Illmer, B. and Vielhaber, M. (2019). Synchronizing digital process twins between virtual products and resources – A virtual design method. *Procedia CIRP*, 84:532–537.
- [Issa, 2021] Issa, M. (2021). Expeditious covid-19 similarity measure tool based on consolidated sca algorithm with mutation and opposition operators. *Applied Soft Computing*, 104:107197.
- [Jacobs and Chase, 2018] Jacobs, F. R. and Chase, R. B. (2018). *Administración de Operaciones. Producción y Cadena de Suministro*. Mcgraw-Hill, decimoquinta edición edition.
- [Jay Lee, 2015] Jay Lee, Hung-An Kao, S. Y. (2015). Service innovation and smart analytics for industry 4.0 and big data environment. *Procedia CIRP*, 16:3–8.
- [Jeong and Jung, 2012] Jeong, S.-J. and Jung, H. (2012). Optimal buffer allocation in flexible manufacturing systems using genetic algorithm and simulation. *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, 6:1071–1080.
- [Jones et al., 2020] Jones, D., Snider, C., Nassehi, A., Yon, J., and Hicks, B. (2020). Characterising the digital twin: A systematic literature review. *CIRP Journal of Manufacturing Science and Technology*, 29:36–52.

- 
- [Karimi-Mamaghan et al., 2022] Karimi-Mamaghan, M., Mohammadi, M., Meyer, P., Karimi-Mamaghan, A. M., and Talbi, E.-G. (2022). Machine learning at the service of meta-heuristics for solving combinatorial optimization problems: A state-of-the-art. *European Journal of Operational Research*, 296(2):393–422.
- [Karmouni et al., 2022] Karmouni, H., Chouiekh, M., Motahhir, S., Qjidaa, H., Ouazzani Jamil, M., and Sayyouri, M. (2022). A fast and accurate sine-cosine mppt algorithm under partial shading with implementation using arduino board. *Cleaner Engineering and Technology*, 9:100535.
- [Kashyap and Parhi, 2025] Kashyap, A. K. and Parhi, D. (2025). Human-inspired dynamic obstacle and inter-collision avoidance algorithm for humanoid biped robots. *Robotics and Autonomous Systems*, page 105023.
- [Khedri Liraviasl et al., 2015] Khedri Liraviasl, K., ElMaraghy, H., Hanafy, M., and Samy, S. (2015). A framework for modelling reconfigurable manufacturing systems using hybridized discrete-event and agent-based simulation. *IFAC-PapersOnLine*, 48(3):1490–1495. 15th IFAC Symposium on Information Control Problems in Manufacturing.
- [Kiefer et al., 2024] Kiefer, D., Wezel, S., Böttcher, A., Grimm, F., Straub, T., Bitsch, G., and Van Dinther, C. (2024). Anomaly detection in hobbing tool images: Using an unsupervised deep learning approach in manufacturing industry. *Procedia Computer Science*, 232:2396–2405. 5th International Conference on Industry 4.0 and Smart Manufacturing (ISM 2023).
- [Kim et al., 2024] Kim, M. J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E., Lam, G., Sanketi, P., Vuong, Q., Kollar, T., Burchfiel, B., Tedrake, R., Sadigh, D., Levine, S., Liang, P., and Finn, C. (2024). Openvla: An open-source vision-language-action model.
- [Kiyokawa et al., 2023] Kiyokawa, T., Shirakura, N., Wang, Z., Yamanobe, N., Ramirez-Alpizar, I. G., Wan, W., and Harada, K. (2023). Difficulty and complexity definitions for assembly task allocation and assignment in human–robot collaborations: A review. *Robotics and Computer-Integrated Manufacturing*, 84:102598.
- [Kolla et al., 2019] Kolla, S., Minufekr, M., and Plapper, P. (2019). Deriving essential components of lean and industry 4.0 assessment model for manufacturing smes. *Procedia CIRP*, 81:753–758. 52nd CIRP Conference on Manufacturing Systems (CMS), Ljubljana, Slovenia, June 12-14, 2019.
- [Koubaa, 2016] Koubaa, A. (2016). *Robot Operating System (ROS) The Complete Reference*. Springer, 1st ed.
- [Kunath and Winkler, 2018] Kunath, M. and Winkler, H. (2018). Integrating the Digital Twin of the manufacturing system into a decision support system for improving the order management process. *Procedia CIRP*, 72:225–231.
- [Kuo and Wang, 2022] Kuo, T. and Wang, K.-J. (2022). A hybrid k-prototypes clustering approach with improved sine-cosine algorithm for mixed-data classification. *Computers & Industrial Engineering*, 169:108164.

- 
- [Kusiak, 2018] Kusiak, A. (2018). Smart manufacturing. *International Journal of Production Research*, 56(1-2):508–517.
- [Langlotz et al., 2021] Langlotz, P., Siedler, C., and Aurich, J. C. (2021). Unification of lean production and industry 4.0. *Procedia CIRP*, 99:15–20. 14th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 15-17 July 2020.
- [Laurindo et al., 2019] Laurindo, Q. M. G., Peixoto, T. A., and de Assis Rangel, J. J. (2019). Communication mechanism of the discrete event simulation and the mechanical project softwares for manufacturing systems. *Journal of Computational Design and Engineering*, 6(1):70–80.
- [Lei et al., 2022] Lei, K., Guo, P., Wang, Y., Wu, X., and Zhao, W. (2022). Solve routing problems with a residual edge-graph attention neural network. *Neurocomputing*, 508:79–98.
- [Leitao and Strasser, 2016] Leitao, P. and Strasser, T. (2016). Analyzing standardization needs for applying agent technology in industrial environments. In *Proceedings of the 2016 IEEE 25th International Symposium on Industrial Electronics (ISIE)*. 2016 IEEE 25th International Symposium on Industrial Electronics (ISIE) ; Conference date: 08-06-2016 Through 10-06-2016.
- [Leng et al., 2020] Leng, J., Liu, Q., Ye, S., Jing, J., Wang, Y., Zhang, C., Zhang, D., and Chen, X. (2020). Digital twin-driven rapid reconfiguration of the automated manufacturing system via an open architecture model. *Robotics and Computer-Integrated Manufacturing*, 63(November 2019).
- [Leusin et al., 2018] Leusin, M. E., Kück, M., Frazzon, E. M., Maldonado, M. U., and Freitag, M. (2018). Potential of a multi-agent system approach for production control in smart factories. *IFAC-PapersOnLine*, 51(11):1459–1464. 16th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2018.
- [Li et al., 2018a] Li, S., Fang, H., and Liu, X. (2018a). Parameter optimization of support vector regression based on sine cosine algorithm. *Expert Systems with Applications*, 91:63–77.
- [Li et al., 2018b] Li, X., Bayrak, E., Epureanu, B., and Koren, Y. (2018b). Real-time teaming of multiple reconfigurable manufacturing systems. *CIRP Annals*, 67.
- [Liu et al., 2021] Liu, M., Fang, S., Dong, H., and Xu, C. (2021). Review of digital twin about concepts, technologies, and industrial applications. *Journal of Manufacturing Systems*, 58:346–361. Digital Twin towards Smart Manufacturing and Industry 4.0.
- [Lu et al., 2023] Lu, W., Chen, J., Fu, Y., Pan, Y., and Ghansah, F. A. (2023). Digital twin-enabled human-robot collaborative teaming towards sustainable and healthy built environments. *Journal of Cleaner Production*, 412:137412.
- [Lu et al., 2020] Lu, Y., Liu, C., Wang, K. I., Huang, H., and Xu, X. (2020). Digital Twin-driven smart manufacturing: Connotation, reference model, applications and research issues. *Robotics and Computer-Integrated Manufacturing*, 61:101837.

- 
- [Luna et al., 2020] Luna, L. S., Caurin, G. A. P., and Tronco, M. L. (2020). Usage of mqtt, ros and aws in the manufacturing process of aircrafts. *32 ICAS Congress Proceedings- The International Council of the Aeronautical Sciences*.
- [Lyu et al., 2021] Lyu, F., Fan, X., Ding, F., and Chen, Z. (2021). Prediction of the axial compressive strength of circular concrete-filled steel tube columns using sine cosine algorithm-support vector regression. *Composite Structures*, 273:114282.
- [Madni et al., 2019] Madni, A. M., Madni, C. C., and Lucero, S. D. (2019). Leveraging digital twin technology in model-based systems engineering. *Systems*, 7(1).
- [Mariani and Omicini, 2023] Mariani, S. and Omicini, A. (2023). Special issue “advances in multi-agent systems”: Editorial. *Applied Sciences*, 13(5).
- [Marinelli et al., 2021] Marinelli, M., Deshmukh, A. A., Janardhanan, M., and Nielsen, I. (2021). Lean manufacturing and industry 4.0 combinative application: Practices and perceived benefits. *IFAC-PapersOnLine*, 54(1):288–293. 17th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2021.
- [Mayr et al., 2018] Mayr, A., Weigelt, M., Kühn, A., Grimm, S., Erll, A., Potzel, M., and Franke, J. (2018). Lean 4.0 - a conceptual conjunction of lean management and industry 4.0. *Procedia CIRP*, 72:622–628. 51st CIRP Conference on Manufacturing Systems.
- [Melesse et al., 2020] Melesse, T. Y., Pasquale, V. D., and Riemma, S. (2020). Digital twin models in industrial operations: A systematic literature review. *Procedia Manufacturing*, 42:267–272. International Conference on Industry 4.0 and Smart Manufacturing (ISM 2019).
- [Meski et al., 2019] Meski, O., Belkadi, F., Laroche, F., and Furet, B. (2019). Towards a knowledge-based framework for digital chain monitoring within the industry 4.0 paradigm. *Procedia CIRP*, 84:118–123.
- [Mirjalili, 2016] Mirjalili, S. (2016). Sca: A sine cosine algorithm for solving optimization problems. *Knowledge-Based Systems*, 96:120–133.
- [Mnih et al., 2015] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.
- [Molenda et al., 2017] Molenda, P., Drews, T., Oechsle, O., Butzer, S., and Steinhilper, R. (2017). A simulation-based framework for the economic evaluation of flexible manufacturing systems. *Procedia CIRP*, 63:201–206. Manufacturing Systems 4.0 – Proceedings of the 50th CIRP Conference on Manufacturing Systems.
- [Mon and Del Giorgio, 2021] Mon, A. and Del Giorgio, H. R. (2021). Evaluation of information and communication technologies towards industry 4.0. *Procedia Computer Science*, 180:639–648. Proceedings of the 2nd International Conference on Industry 4.0 and Smart Manufacturing (ISM 2020).

- 
- [Mon and Del Giorgio, 2022] Mon, A. and Del Giorgio, H. R. (2022). Analysis of industry 4.0 products in small and medium enterprises. *Procedia Computer Science*, 200:914–923. 3rd International Conference on Industry 4.0 and Smart Manufacturing.
- [Mon and Del Giorgio, 2023] Mon, A. and Del Giorgio, H. R. (2023). Technological transformation model for smes. *Procedia Computer Science*, 217:1076–1085. 4th International Conference on Industry 4.0 and Smart Manufacturing.
- [Mon et al., 2024] Mon, A., Giorgio, H. D., Roark, G., Saavedra Sueldo, C., Perez Colo, I., and Paula, M. D. (2024). Sme users in industry 4.0 study of the metalworking industry in argentina. *CLEI Electronic Journal*, 28:1–18.
- [Monahan, 1982] Monahan, G. E. (1982). State of the art - A survey of partially observable Markov Decision Processes: theory, models, and algorithms. *Management Science*, 28(1):1–16.
- [Montevechi et al., 2010] Montevechi, J. A. B., Fabiano, L., Pinho, A., Costa, R., Oliveira, M. L., and Silva, A. (2010). Conceptual modeling in simulation projects by mean adapted idef: An application in a brazilian tech company. In *Proceedings Winter Simulation Conference*, pages 1624–1635.
- [Mourad et al., 2020] Mourad, M. H., Nassehi, A., Schaefer, D., and Newman, S. T. (2020). Assessment of interoperability in cloud manufacturing. *Robotics and Computer-Integrated Manufacturing*, 61(June 2018):101832.
- [Mourtzis et al., 2014] Mourtzis, D., Doukas, M., and Bernidaki, D. (2014). Simulation in manufacturing: Review and challenges. *Procedia CIRP*, 25(C):213–229.
- [Mrugalska and Wyrwicka, 2017] Mrugalska, B. and Wyrwicka, M. K. (2017). Towards lean production in industry 4.0. *Procedia Engineering*, 182:466–473. 7th International Conference on Engineering, Project, and Production Management.
- [Nazari et al., 2018] Nazari, M., Oroojlooy, A., Snyder, L. V., and Takáč, M. (2018). Reinforcement learning for solving the vehicle routing problem. arXiv.
- [Negri et al., 2017] Negri, E., Fumagalli, L., and Macchi, M. (2017). A review of the roles of digital twin in cps-based production systems. *Procedia Manufacturing*, 11:939–948. 27th International Conference on Flexible Automation and Intelligent Manufacturing, FAIM2017, 27-30 June 2017, Modena, Italy.
- [Ohno, 1988] Ohno, T. (1988). *Toyota Production System: Beyond Large-Scale Production*. Taylor & Francis.
- [Ottoni et al., 2022] Ottoni, A. L. C., Nepomuceno, E. G., Oliveira, M. S. d., and Oliveira, D. C. R. d. (2022). Reinforcement learning for the traveling salesman problem with refueling. *Complex & Intelligent Systems*, 8:2001–2015.
- [O’Kane, 2014] O’Kane, J. M. (2014). *A Gentle Introduction to ROS*. JasonMatthew O’Kane.

- 
- [Paternina-Arboleda and Das, 2001] Paternina-Arboleda, C. D. and Das, T. K. (2001). Intelligent dynamic control policies for serial production lines. *IIE Transactions*, 33:65–77.
- [Paternina-Arboleda and Das, 2005] Paternina-Arboleda, C. D. and Das, T. K. (2005). A multi-agent reinforcement learning approach to obtaining dynamic control policies for stochastic lot scheduling problem. *Simulation Modelling Practice and Theory*, 13(5):389–406.
- [Perez Colo et al., 2023] Perez Colo, I., Saavedra Sueldo, C., De Paula, M., and Acosta, G. G. (2023). Intelligent approach for the industrialization of deep learning solutions applied to fault detection. *Expert Systems with Applications*, 233:120959.
- [Pizoń and Gola, 2023] Pizoń, J. and Gola, A. (2023). Human–machine relationship—perspective and future roadmap for industry 5.0 solutions. *Machines*, 11(2).
- [Purushothama, 2013] Purushothama, B. (2013). 1 - what is work quality? In Purushothama, B., editor, *Work Quality Management in the Textile Industry*, pages 1–86. Woodhead Publishing India.
- [Qian et al., 2020] Qian, C., Zhang, Y., Jiang, C., Pan, S., and Rong, Y. (2020). A real-time data-driven collaborative mechanism in fixed-position assembly systems for smart manufacturing. *Robotics and Computer-Integrated Manufacturing*, 61(July 2019).
- [Qudeiri, 2017] Qudeiri, J. E. A. (2017). Production simulator system for flexible routing optimization in flexible manufacturing systems. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 231(7):1237–1247.
- [Rasheed et al., 2020] Rasheed, A., San, O., and Kvamsdal, T. (2020). Digital twin: Values, challenges and enablers from a modeling perspective. *IEEE Access*, 8:21980–22012.
- [Rehman et al., 2025] Rehman, A. U., Jiao, W., Jiang, Y., Wei, J., Sohaib, M., Sun, J., E, S., Rehman, K. U., and Chi, Y. (2025). Deep learning in industrial machinery: A critical review of bearing fault classification methods. *Applied Soft Computing*, 171:112785.
- [Roark et al., 2024a] Roark, G., De Paula, M., Saavedra Sueldo, C., Perez Colo, I., and Acosta, G. G. (2024a). *Transformación digital en PyMEs Metalmecánicas de Olavarría: desafíos y oportunidades*. Universidad Nacional del Centro de la Provincia de Buenos Aires.
- [Roark et al., 2024b] Roark, G., Saavedra Sueldo, C., Perez Colo, I., Paula, M. D., and Acosta, G. G. (2024b). Impulsando la transformación digital: Un caso de vinculación entre la universidad y pymes metalmecánicas. *JAIIO, Jornadas Argentinas de Informática. Revista de la Sociedad Argentina de Informática e Investigación Operativa (SADIO)*, 10:59–72.

- 
- [Roark et al., 2024c] Roark, G., Vázquez, M., Perez Colo, I., Saavedra Sueldo, C., and Berdun, D. (2024c). Madurez tecnológica en pymes industriales de olavarria: perspectivas de su preparación para la transformación digital. *AACINI - Revista Internacional de Ingeniería Industrial*, 4(1):1–15.
- [Rolle et al., 2020] Rolle, R. P., Martucci, V. D. O., and Godoy, E. P. (2020). Architecture for Digital Twin implementation focusing on Industry 4.0. *IEEE Latin America Transactions*, 18(5):889–898.
- [Shah and Patel, 2018] Shah, D. and Patel, P. (2018). Productivity improvement by implementing lean manufacturing tools in manufacturing industry. *International Research Journal of Engineering and Technology (IRJET)*, 5:3794–3798.
- [Shaw and Garlan, 1996] Shaw, M. and Garlan, D. (1996). *Software architecture: perspectives on an emerging discipline*. Upper Saddle River, NJ.
- [Siemens, 2024] Siemens (2024). Supercharging the industry transformation with the comprehensive digital twin - white paper.
- [Singh et al., 2022] Singh, S., Singh, A., Kapil, S., and Das, M. (2022). Utilization of a tsp solver for generating non-retractable, direction favouring toolpath for additive manufacturing. *Additive Manufacturing*, page 103126.
- [Skinderowicz, 2022] Skinderowicz, R. (2022). Improving ant colony optimization efficiency for solving large tsp instances. *Applied Soft Computing*, 120:108653.
- [Sofiane and Djamila, 2019] Sofiane, K. and Djamila, H. (2019). A Temporal Distributed Group Decision Support System Based on Multi-Criteria Analysis. *International Journal of Interactive Multimedia and Artificial Intelligence*, 5(7):7–21.
- [Sterman, 2000] Sterman, J. D. (2000). *Business dynamics: Systems thinking and modeling for a complex world*. McGraw-Hill Education.
- [Stączek et al., 2021] Stączek, P., Pizoń, J., Danilczuk, W., and Gola, A. (2021). A digital twin approach for the improvement of an autonomous mobile robots (amr’s) operating environment—a case study. *Sensors*, 21(23).
- [Stock and Seliger, 2016] Stock, T. and Seliger, G. (2016). Opportunities of sustainable manufacturing in industry 4.0. *Procedia CIRP*, 40:536–541. 13th Global Conference on Sustainable Manufacturing – Decoupling Growth from Resource Use.
- [Sun et al., 2020] Sun, X., Bao, J., Li, J., Zhang, Y., Liu, S., and Zhou, B. (2020). A digital twin-driven approach for the assembly-commissioning of high precision products. *Robotics and Computer-Integrated Manufacturing*, 61.
- [Sutton and Barto, 1998] Sutton, R. S. and Barto, A. G. (1998). *Reinforcement learning: An introduction*. MIT Press.
- [Sutton and Barto, 2018] Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction, second edition*. Bradford Books.

- 
- [Tao et al., 2019] Tao, F., Zhang, H., Liu, A., and Nee, A. Y. C. (2019). Digital twin in industry: State-of-the-art. *IEEE Transactions on Industrial Informatics*, 15(4):2405–2415.
- [Thirunavukkarasu et al., 2023] Thirunavukkarasu, M., Sawle, Y., and Lala, H. (2023). A comprehensive review on optimization of hybrid renewable energy systems using various optimization techniques. *Renewable and Sustainable Energy Reviews*, 176:113192.
- [Thomas H. Uhlemann, 2017] Thomas H. Uhlemann, Christian Lehmann, R. S. (2017). The digital twin: Realizing the cyber-physical production system for industry 4.0. *Procedia CIRP*, 61:335–340.
- [Ullah et al., 2024] Ullah, H., Uzair, M., Jan, Z., and Ullah, M. (2024). Integrating industry 4.0 technologies in defense manufacturing: Challenges, solutions, and potential opportunities. *Array*, 23:100358.
- [Uriarte et al., 2018] Uriarte, A. G., Ng, A. H., and Moris, M. U. (2018). Supporting the lean journey with simulation and optimization in the context of industry 4.0. *Procedia Manufacturing*, 25:586–593. Proceedings of the 8th Swedish Production Symposium (SPS 2018).
- [Wagner et al., 2017] Wagner, T., Herrmann, C., and Thiede, S. (2017). Industry 4.0 impacts on lean production systems. *Procedia CIRP*, 63:125–131. Manufacturing Systems 4.0 – Proceedings of the 50th CIRP Conference on Manufacturing Systems.
- [Waissi et al., 2015] Waissi, G. R., Demir, M., Humble, J. E., and Lev, B. (2015). Automation of strategy using idef0 — a proof of concept. *Operations Research Perspectives*, 2:106–113.
- [Wang et al., 2020] Wang, J., Xu, C., Zhang, J., Bao, J., and Zhong, R. (2020). A collaborative architecture of the industrial internet platform for manufacturing systems. *Robotics and Computer-Integrated Manufacturing*, 61(August 2019).
- [Wang and Tang, 2021] Wang, Q. and Tang, C. (2021). Deep reinforcement learning for transportation network combinatorial optimization: A survey. *Knowledge-Based Systems*, 233:107526.
- [Watkins and Dayan, 1992] Watkins, C. J. C. H. and Dayan, P. (1992). Q-learning. *Machine Learning*, 8:279–292.
- [Weckenborg et al., 2024] Weckenborg, C., Schumacher, P., Thies, C., and Spengler, T. S. (2024). Flexibility in manufacturing system design: A review of recent approaches from operations research. *European Journal of Operational Research*, 315(2):413–441.
- [Winston, 2004] Winston, W. L. (2004). *Operations Research, Applications and Algorithms, Fourth edition*. Brooks/Cole—Thomson Learning.
- [Womack et al., 2007] Womack, J. P., Jones, D. T., and Roos, D. (2007). *The Machine That Changed the World: The Story of Lean Production—Toyota’s Secret*

---

*Weapon in the Global Car Wars That Is Now Revolutionizing World Industry*. Free Press.

- [Wu et al., 2019] Wu, Y., Song, W., Cao, Z., Zhang, J., and Lim, A. (2019). Learning improvement heuristics for solving routing problems. *arXiv*.
- [Xia et al., 2021] Xia, K., Sacco, C., Kirkpatrick, M., Saidu, C., Nguyen, L., Kircaliali, A., and Harik, R. (2021). A digital twin to train deep reinforcement learning agent for smart manufacturing plants: Environment, interfaces and intelligence. *Journal of Manufacturing Systems*, 58:210–230. Digital Twin towards Smart Manufacturing and Industry 4.0.
- [Yamazaki et al., 2017] Yamazaki, Y., Shigematsu, K., Kato, S., Kojima, F., Onari, H., and Takata, S. (2017). Design method of material handling systems for lean automation—integrating equipment for reducing wasted waiting time. *CIRP Annals*, 66(1):449–452.
- [Yang and Shami, 2020] Yang, L. and Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, 415:295–316.
- [Yang et al., 2020] Yang, Q.-Y., Chu, S. C., Pan, J.-S., and Chen, C.-M. (2020). Sine cosine algorithm with multigroup and multistrategy for solving cvrp. *Mathematical Problems in Engineering*, 2020:8184254.
- [Yau and McNally, 2023] Yau, J. O.-Y. and McNally, G. P. (2023). The rescorla-wagner model, prediction error, and fear learning. *Neurobiology of Learning and Memory*, 203:107799.
- [Zeigler et al., 2018] Zeigler, B. P., Muzy, A., and Kofman, E. (2018). *Theory of Modeling and Simulation, Discrete Event and Iterative System Computational Foundations*. Elsevier, third edition.
- [Zhang et al., 2020] Zhang, K., Qu, T., Zhou, D., Jiang, H., Lin, Y., Li, P., Guo, H., Liu, Y., Li, C., and Huang, G. Q. (2020). Digital twin-based opti-state control method for a synchronized production operation system. *Robotics and Computer-Integrated Manufacturing*, 63(November 2019):101892.
- [Zhong et al., 2017] Zhong, R. Y., Xu, X., Klotz, E., and Newman, S. T. (2017). Intelligent Manufacturing in the Context of Industry 4.0: A Review. *Engineering*, 3(5):616–630.
- [Zhou and Huang, 2024] Zhou, D. and Huang, D. (2024). A review on the progress, challenges and prospects in the modeling, simulation, control and diagnosis of thermodynamic systems. *Advanced Engineering Informatics*, 60:102435.
- [Zolotová et al., 2020] Zolotová, I., Papcun, P., Kajáti, E., Miškuf, M., and Mocnej, J. (2020). Smart and cognitive solutions for operator 4.0: Laboratory h-cpps case studies. *Computers & Industrial Engineering*, 139:105471.