



**UNIVERSIDAD POLITECNICA SALESIANA
SEDE QUITO**

CARRERA DE INGENIERIA INDUSTRIAL

**EVALUACIÓN DEL ALGORITMO DE CUELLO DE BOTELLA MOVIL
UTILIZANDO INTELIGENCIA ARTIFICIAL EN PROCESOS DE
MANUFACTURA**

Trabajo de titulación previo a la obtención del

Título de Ingenieros Industriales

AUTORES: ANDRE DAMIAN CANGAS TAPIA

JUAN MANUEL CANDO CRUZ

TUTOR: LUIS FERNANDO TIPAN VERGARA

Quito-Ecuador

2025

**CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE
TITULACIÓN**

Nosotros, André Damián Cangas Tapia con documento de identificación N.º 1724994486 y Juan Manuel Cando Cruz con documento de identificación N.º 1722632195; manifestamos que: Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Quito, 30 de enero del año 2025

Atentamente



André Damián Cangas Tapia

1724994486



Juan Manuel Cando Cruz

1722632195

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACION A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Nosotros, André Damián Cangas Tapia con documento de identificación N.º 1724994486 y Juan Manuel Cando Cruz con documento de identificación N.º 1722632195, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Artículo Académico, “EVALUACIÓN DEL ALGORITMO DE CUELLO DE BOTELLA MOVIL UTILIZANDO INTELIGENCIA ARTIFICIAL EN PROCESOS DE MANUFACTURA”, el cual ha sido desarrollo para optar por el título de: Ingenieros Industriales, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente. En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Quito, 30 de enero del año 2025

Atentamente



André Damián Cangas Tapia

1724994486



Juan Manuel Cando Cruz

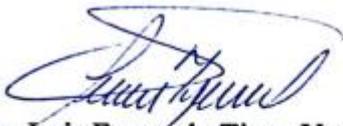
1722632195

CERTICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Luis Fernando Tipan Vergara, con documento de identificación N° 1717329005 Docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: EVALUACIÓN DEL ALGORITMO DE CUELLO DE BOTELLA MOVIL UTILIZANDO INTELIGENCIA ARTIFICIAL EN PROCESOS DE MANUFACTURA, realizado por André Damián Cangas Tapia con documento de identificación N.º 1724994486 y Juan Manuel Cando Cruz con documento de identificación N.º 1722632195, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto Técnico, que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito, 30 de enero del año 2025

Atentamente



Ing. Luis Fernando Tipan Vergara M.Sc.

C.I 1717329005

Docente Tutor

Lista de contenidos

Introducción	10
Objetivos	13
CAPITULO I.....	15
1 MARCO TEORICO.....	15
1.1 Cuello de botella.....	15
1.2 Heurística Cuello de Botella Móvil (CBM)	15
1.3 Métodos para abordar los cuellos de botella	16
1.4 Inteligencia Artificial en Manufactura	16
1.5 Definición de Metaheurística.....	19
1.5.1 Ventajas de la Metaheurística.....	20
1.6 Simulación de procesos industriales.....	20
1.7 Simulink cómo herramienta de simulación	20
1.8 Machine Learning.....	21
1.9 Manufactura en la Industria de Embotellado de Aceites	22
1.10 Herramientas de Análisis y Evaluación de Procesos	22
1.11 Industria 4.0.....	23
1.12 Áreas de aplicación de la Inteligencia Artificial	24
1.13 Simulación y Optimización	25
1.13.1 Simulación de Eventos Discretos	25
1.13.2 Integración y Simulación.....	26
1.13.3 Ejemplos de Aplicaciones	26
Capitulo II	28
2 MÉTODOS Y MATERIALES	28
2.1 Descripción de la planta embotelladora.....	28
2.2 Infraestructura de la Planta.	28

2.3	Loyaut de la línea de producción.....	30
2.4	Inteligencia Artificial.....	31
2.4.1	Generalidades	31
2.4.2	Implementación de Machine Learning al Algoritmo Cuello de Botella Móvil	31
2.4.3	Funciones utilizadas	31
2.4.4	Interpretación grafica del Algoritmo Cuello de Botella Móvil potenciado por Machine Learning	33
2.5	Herramientas de Análisis.....	35
2.5.1	Simulink	35
2.5.2	Flexsim	35
2.6	Estructura del modelado de sistema.	35
2.7	Bloques Utilizados.....	36
2.7.1	Explicación de los bloques.	36
CAPITULO III		38
3	RESULTADOS Y DISCUSIÓN	38
3.1	Formulación del problema.....	38
3.2	Grafo del proceso de manufactura.....	38
3.3	Simulación del proceso de manufactura.....	40
3.3.1	Situación inicial de la planta.....	41
3.4	Evaluación utilizando algoritmo Cuello de Botella Móvil.....	43
3.5	Evaluación utilizando algoritmo Cuello de Botella Móvil utilizando Inteligencia Artificial	45
3.6	Árbol de decisiones	48
3.6.1	Funcionamiento	49
3.7	Conclusión	50
3.8	Explicación	50
3.8.1	Cargar la Base de datos.	50
3.8.2	Preparar los Datos para el modelo.	51
3.8.3	Dividir los datos en conjunto de entrenamiento (80%) y prueba (20%).....	51

3.8.4	Creación de la partición de datos.....	51
3.8.5	Seleccionar el conjunto de entrenamiento.	52
3.8.6	Seleccionar el conjunto de prueba.	52
3.8.7	Entrenar el Modelo.....	53
3.8.8	Carga del modelo y predicción de cuellos de botella.	54
3.8.9	Predecir el cuello de botella.	54
3.8.10	Generar la Secuencia de Trabajo.....	54
3.8.11	Escalabilidad.	54
3.8.12	Máquinas predichas cómo Cuellos de Botella por hora	56
3.8.13	Eficiencia del Proceso por Hora	57
3.8.14	Matriz de Confusión.....	58
3.8.15	Árbol de decisiones	59
3.8.16	Distribución del Tiempo de Retraso.....	60
3.8.17	Comparación de Resultados Reales vs Predichos	61
3.9	Complejidad Computacional.	61
3.9.1	Tiempo de ejecución.....	65
3.10	Eficiencia.....	68
3.11	Eficacia.....	69
3.12	Ventajas de utilizar Machine Learning en el Algoritmo Cuello de Botella Móvil	69
3.12.1	Adaptación al entorno de la empresa.....	69
3.12.2	Simplificación en la toma de información.....	69
3.12.3	Facilidad de interpretación	70
3.12.4	Desventajas de utilizar Machine Learning en el Algoritmo Cuello de Botella Móvil ..	70
3.13	Conclusiones	71
3.14	Recomendaciones.....	72

Lista de Figuras

Figura 1. Secuencia Implementación Mercedes-Benz [2]	19
Figura 2. Componentes de Industria y Logística 4.0[6].....	23
Figura 3. Algoritmo Machine Learning (Fuente: Elaboración propia).....	31
Figura 5. Simulación embotellado de aceites (Fuente: Elaboración Propia)	40
Figura 7. Máquinas Predichas cómo Cuellos de Botella por Hora	56
Figura 8. Eficiencia del Proceso por Hora	57
Figura 9. Matriz Confusión 100 máquinas.....	58
Figura 10. Árbol de Decisiones con 20 máquinas.....	59
Figura 11. Distribucion del Tiempo de Retraso (fuente: Elaboración Propia)	60
Figura 12. Comparación de Resultados Reales vs Predichos	61
Figura 13. Uso de recursos previo a evaluación de Algoritmo	63
Figura 14. Uso de recursos durante evaluación de Algoritmo	64
Figura 15. Número de máquinas vs tiempo de ejecución	66
Figura 16. Número de tareas vs tiempo de ejecución	67
Figura 17. Pico máximo de consumo utilizando Machine Learning	68

Lista de tablas

Tabla 1 Resultados Coca Cola [1]	17
Tabla 2 Resultados Mercedes-Benz [2].....	18
Tabla 3. Aplicaciones Inteligencia Artificial [6].....	24
Tabla 4. Aplicación Machine Learning[10]	26
Tabla 5. Características Planta Embotelladora (Fuente: Elaboración propia).....	29
Tabla 6 Estructura Árbol de Decisiones (Fuente: Elaboración propia).....	34
Tabla 7 Bloques utilizados Simulink (Fuente: Elaboración propia)	36
Tabla 8 Capacidad de Llenado (Fuente: Elaboración propia)	39
Tabla 9 Actividad de Máquinas (Fuente: Elaboración propia)	39
Tabla 10 Ruta de trabajos (Fuente: Elaboración Propia)	39
Tabla 11 AvgStayTime luego de CBM (Fuente: Elaboración Propia)	44
Tabla 12. Resultados Simulación 20 máquinas utilizando CBM implementado Machine Learning	55
Tabla 13. Características Equipo para evaluación.....	62
Tabla 14 Comparación tiempos de ejecución (Fuente: Elaboración Propia)	65
Tabla 15 Calculo memoria utilizada (Fuente: Elaboración Propia)	68

Resumen

El presente estudio evaluó la efectividad del Algoritmo de Cuello de Botella Móvil (CBM) potenciado con Inteligencia Artificial (IA) en una planta embotelladora de aceites. Se realizó un análisis comparativo entre el método tradicional basado en la experiencia de los trabajadores y la implementación del CBM con IA. La metodología incluyó la recolección de datos en la planta, la simulación del proceso con Simulink y la implementación de un algoritmo híbrido.

Los resultados mostraron que, en plantas con baja complejidad, la experticia de los trabajadores puede ser tan eficiente como los algoritmos automatizados. Sin embargo, en entornos con un mayor número de máquinas, el CBM con IA mejoró la precisión en la identificación de cuellos de botella, optimizando la producción en un 22.73%. La integración de Machine Learning permitió una adaptación dinámica a las condiciones del sistema, mejorando la planificación y distribución de tareas.

Se concluyó que la implementación del CBM con IA es una herramienta eficiente para mejorar la productividad en manufactura, siempre que se disponga de una base de datos adecuada. La escalabilidad del algoritmo lo hace viable en empresas de distintos tamaños, facilitando la toma de decisiones en entornos de producción dinámicos.

Palabras clave: Cuello de botella móvil, inteligencia artificial, manufactura, machine learning, optimización.

Abstract

This study evaluated the effectiveness of the Mobile Bottleneck Algorithm (CBM) enhanced with Artificial Intelligence (AI) in an oil bottling plant. A comparative analysis was conducted between the traditional method, relying on workers' expertise, and the implementation of CBM with AI. The methodology included data collection from the plant, process simulation using Simulink, and the implementation of a hybrid algorithm.

The results showed that in low-complexity plants, worker expertise can be as efficient as automated algorithms. However, in environments with a larger number of machines, CBM with AI improved the accuracy in identifying bottlenecks, optimizing production by 22.73%. The integration of Machine Learning enabled dynamic adaptation to system conditions, enhancing task planning and distribution.

It was concluded that implementing CBM with AI is an efficient tool for improving productivity in manufacturing, provided that an adequate database is available. The algorithm's scalability makes it viable for companies of different sizes, facilitating decision-making in dynamic production environments.

Keywords: Mobile bottleneck, artificial intelligence, manufacturing, machine learning, optimization.

Introducción

Antecedentes

Las empresas en los procesos de manufactura tienden a tener cuellos de botella se originan por limitaciones en recursos, capacidad o procesos, interrumpiendo el flujo de trabajo, reduciendo la eficiencia, generando tiempos de espera y pérdidas económicas.

Problema

El uso de algoritmos de Cuello de Botella Móvil (CBM) ayudan a optimizar procesos y disminuir los cuellos de botella. La problemática se centra en que este algoritmo necesita muchos recursos para poder funcionar además que en entornos dinámicos dificultaría su adaptabilidad, adicionando que la calidad y falta de datos puede afectar en la búsqueda de cuellos de botella. Por otro lado, en empresas pequeñas como medianas al momento de querer escalar en su nivel de producción por aumento de pedidos se encuentran con problemas como un sistema que soporte este tipo de algoritmos que es costoso dado de su complejidad y no tienen los recursos suficientes para poder solventarlo.

Justificación

Conociendo el problema que puede tener este tipo de algoritmo en cuanto al uso de recursos computacionales, el uso de la IA (Inteligencia artificial) es de gran ayuda pues esta puede seguir analizando el sistema en tiempo real y utilizando una menor cantidad de recursos computacionales.

La evaluación se centra en una embotelladora de aceites, donde se utiliza en mayor parte la experticia de los colaboradores para resolver cuellos de botella, pues no se cuenta con un sistema de cómputo robusto que permita controlar algoritmos altamente complejos que vuelvan eficiente su proceso. Por lo que vuelve un sistema idóneo para poder aplicar y evaluar el uso de la inteligencia artificial.

Objetivos

Objetivo General

- Evaluar la efectividad del Algoritmo Cuello de Botella Móvil utilizando la IA (Inteligencia Artificial) en la optimización de embotellamiento de aceite mediante el análisis comparativo entre el proceso que lleva actualmente la embotelladora y el Algoritmo de Cuello de Botella Móvil utilizando la (IA) Inteligencia Artificial para comprobar su rendimiento dentro de la embotelladora.

Objetivos Específicos

- Analizar la eficiencia del proceso llevado por la expertiz de los colaboradores y la eficiencia del “Algoritmo de cuello de botella móvil utilizando la Inteligencia Artificial”, mediante la comparativa de tiempo de procesamiento, adaptabilidad de cambios, calidad de datos y escalabilidad.
- Comprobar la escalabilidad del Algoritmo cuello de botella móvil utilizando Inteligencia Artificial, mediante la simulación del proceso de embotellamiento de aceite para así corroborar su efectividad.
- Analizar el uso de recursos computacionales al momento de utilizar Inteligencia Artificial, mediante la comprobación de graficas de usos de recursos que nos brinda Simulink, para comprobar si el uso de esta tecnología es óptimo para este tipo de empresas.

Metodología

En el trabajo se utilizará la siguiente metodología.

- **Fuente de Datos:** Se realizará una recolección de datos en la planta de manufactura, además de entrevistas con los operadores de la planta
- **Implementación del algoritmo híbrido:** Se llevará a cabo la implementación del algoritmo de cuello de botella móvil en la simulación para su análisis.
- **Simulación con SIMULINK:** Utilizando esta herramienta de simulación observaremos de manera visual y objetiva los datos producidos para su correcto análisis.

CAPITULO I

1 MARCO TEORICO

1.1 Cuello de botella

Un cuello de botella es un punto de restricción en un proceso que limita su capacidad o rendimiento. Se lo puede explicar cómo un embudo: si el cuello es demasiado estrecho, el flujo se verá obstaculizado. A nivel de producción, un cuello de botella es una máquina lenta, un proceso manual o una escasez de materiales que provoca tiempos muertos y cómo consecuencia retrasos.

Por lo tanto, el cuello de botella es un problema para todo tipo de industria, pues conlleva varios problemas que traen consigo pérdidas económicas, por eso existen técnicas que ayudan a disminuir los cuellos de botella, cómo son los métodos Heurísticos y Metaheurísticos.[1]

Cómo ejemplos más habituales de cuello de botella tenemos:

- Procesos mal definidos
- Flujos de trabajo mal diseñados
- Falta de recursos humanos
- Falta de capacidad de la maquinaria
- Trabajos no automatizados
- Mala previsión

1.2 Heurística Cuello de Botella Móvil (CBM)

El cuello de botella móvil (CBM), creado por Adams, Balas y Zawack en 1989, es una heurística para programación de talleres, Su enfoque principal consiste en secuenciar una de las máquinas del taller basándose en su nivel de criticidad, el cual se determina según la secuencia de máquinas que impacta la función objetivo. [2]

Cómo funciona este algoritmo es de la siguiente manera

El algoritmo comienza determinando el cuello de botella inicial estableciendo una secuencia de trabajos óptima para la máquina, luego se establecen restricciones que afectan la secuenciación de las siguientes máquinas. Determinamos y secuenciamos la nueva máquina cuello de botella entre aquellas no secuenciadas y repetimos hasta secuenciar todas las máquinas.[2]. La Heurística Cuello de botella móvil es una heurística eficaz para resolver problemas, pero cuando se tiene un número de máquinas alto se complica de forma exponencial a medida que aumentan las máquinas por lo que se empieza a volver más lento en dar una solución.

1.3 Métodos para abordar los cuellos de botella

La forma habitual de abordar cuellos de botella en pequeñas empresas es mediante observación empírica dado el número de máquinas con las que se cuenta, pero tenemos métodos para poder resolverlos como optimizadores mediante métodos de Lagrange, Recocion Simulada, Algoritmos Genéticos, Colonias de hormigas, Optimización con enjambre de partículas y Búsqueda Tabú [3], Los Algoritmos basados en genética y Colonia de hormigas son métodos de resolución, que se asimilan al Machine Learning pues en base buscan mantener los mejores resultando y seguir probando sin caer en soluciones estancadas.

1.4 Inteligencia Artificial en Manufactura

La Inteligencia Artificial (IA) en la manufactura ya ha desarrollado diversos proyectos enfocados en mejorar la rentabilidad de las empresas. Uno de los ejemplos destacados es Coca-Cola, que ha implementado un sensor de presión virtual basado en aprendizaje automático. Este sistema permite realizar diagnósticos más precisos en los dispensadores de bebidas.

Tabla 1 Resultados Coca Cola [1]

Indicador	Resultados
Predicciones de presión	Se logró alcanzar una precisión del 91% en las predicciones de presión.
Costos	Se eliminó la necesidad de adquirir sensores costosos, lo que generó importantes ahorros para la empresa
Adaptación tecnológica	Se transformó un módulo estándar de control de caudal en un componente inteligente, con capacidad de diagnóstico autónomo

Mercedes-Benz es otra empresa que ha incorporado redes neuronales profundas en la simulación de sensores de hardware. Según Katja Deuschl, desarrolladora de IA en Mercedes-Benz, “Esta fue la primera vez que simulamos sensores con redes neuronales en una de nuestras ECU de tren motriz. Sin MATLAB y Simulink, tendríamos que utilizar un tedioso proceso de codificación manual que era muy lento y propenso a errores.”

Con esta solución, Mercedes-Benz logró convertir los modelos de Deep Learning desarrollados en QKeras en código que puede ser desplegado en una ECU (Unidad de Control Electrónica) de automoción. Los resultados obtenidos incluyen

Tabla 2 Resultados Mercedes-Benz [2]

Indicador	Resultados
Cumplimiento de requisitos técnicos	El modelo cumple con los estrictos requerimientos de CPU, memoria y rendimiento necesarios para su implementación.
Proceso flexible	Se establece un flujo de trabajo adaptable que permite realizar cambios y ajustes de manera eficiente.
Incremento en la velocidad de desarrollo	El tiempo de desarrollo aumenta en un 600%, gracias a la automatización y simplificación del proceso

El proceso que lleva a cabo Mercedes Benz se lo puede ver en la Ilustración.

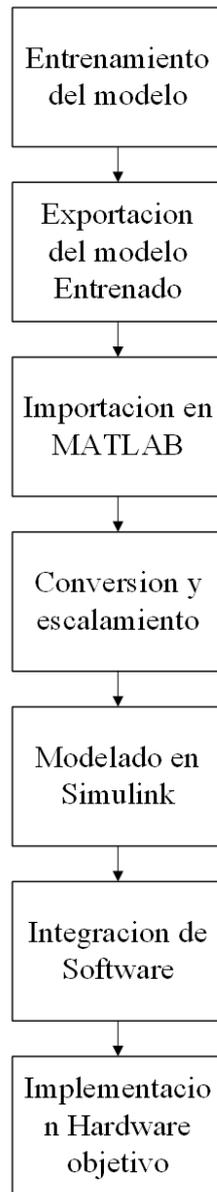


Figura 1. Secuencia Implementación Mercedes-Benz [2]

1.5 Definición de Metaheurística

Metaheurística cómo una serie de procesos iterativos que tienen como objetivo la optimización mediante la utilización de algoritmos aproximados y búsqueda de propósito general, que busca explorar adecuadamente el espacio de búsqueda.[3]

1.5.1 Ventajas de la Metaheurística

Algoritmos basados en Metaheurística tienen ventajas cómo:

1. Fácil Comprensión
2. Implementación sencilla
3. Aplicables en varios campos dado que son generativos
4. Cuentan con la capacidad de combinarse con otras áreas
5. Capaces de resolver problemas complejos de forma ágil.

[4]

1.6 Simulación de procesos industriales

Los simuladores son herramientas de gran importancia dentro de las compañías pues ayudan a buscar la forma de predecir cómo se comporta un sistema. La forma en que funciona es por medio de modelos que permite obtener una visión y una intuición más clara sobre el comportamiento del sistema, estos simuladores a menudo utilizan gráficos potentes con una amplia parametrización para que el efecto de los parámetros pueda explotarse muy rápidamente.[5]

1.7 Simulink cómo herramienta de simulación

Simulink es utilizado para simular, analizar y modelar sistemas dinámicos, que se encuentra en Matlab. Su interfaz gráfica permite ver los modelos que se van creando.[6]. Esta herramienta al estar compuesta de una serie de Toolboxes, permite el control, procesamiento de señales, identificación, redes neuronales, lógica borrosa, control robusto, control adaptativo, manipulación simbólica de ecuaciones adicional los resultados que esta herramienta brinde puede presentarlos en gráficos de 2 y 3 dimensiones.[7]

Simulink es un software que funciona por ventanas las cuales se van construyendo en base a la unión de diferentes bloques, el cual nos permite no solamente realizar un sistema sino unirlo a otros sistemas y convertirlo en un modelo total.[7]

1.8 Machine Learning

El aprendizaje automático, también conocido como Machine Learning, es una disciplina dentro de la inteligencia artificial (IA) que se enfoca en crear algoritmos y modelos capaces de aprender y mejorar su desempeño de manera independiente a partir de datos. A diferencia de los sistemas tradicionales que dependen de reglas programadas manualmente para realizar tareas específicas, el aprendizaje automático utiliza los datos para descubrir patrones, identificar relaciones y generar predicciones. Este enfoque ha adquirido gran relevancia debido al rápido aumento en la disponibilidad de datos y al desarrollo de tecnologías con mayor capacidad de procesamiento computacional.

Existen tres categorías principales de aprendizaje automático: supervisado, no supervisado y por refuerzo. En el aprendizaje supervisado, los algoritmos son entrenados con datos etiquetados que contienen entradas y salidas conocidas, lo que permite al modelo predecir resultados futuros a partir de nuevos datos. Por su parte, el aprendizaje no supervisado opera con datos no etiquetados, buscando estructuras ocultas o patrones dentro de los conjuntos de datos. Finalmente, el aprendizaje por refuerzo se basa en la interacción de un agente con su entorno, donde aprende a tomar decisiones óptimas mediante un proceso de prueba y error, utilizando recompensas y penalizaciones como guía.

Las aplicaciones del aprendizaje automático son diversas e incluyen áreas como el reconocimiento de voz, la visión artificial, los sistemas de recomendación, la detección de fraudes, el diagnóstico médico y los vehículos autónomos. Estas aplicaciones suelen apoyarse en técnicas como redes neuronales profundas, máquinas de soporte vectorial, árboles de decisión y modelos probabilísticos, como los clasificadores Bayesianos.

Sin embargo, el aprendizaje automático enfrenta varios retos, como el tratamiento de datos incompletos, la comprensión de modelos complejos y la existencia de sesgos en los datos, que pueden llevar a resultados inexactos o discriminatorios.

1.9 Manufactura en la Industria de Embotellado de Aceites

En la Manufactura enfocada en la Industria de Aceites viene implicado una serie de procesos que van desde la extracción pasando por el embotellamiento hasta la distribución al cliente final. Enfocándonos en el embotellamiento de Aceites este es un proceso crucial dado que en esta fase vienen implicadas varias etapas, de forma simplificada estas etapas son Almacenamiento, embotellamiento y despacho. Este proceso puede variar dependiendo de la complejidad de la planta embotelladora.

Actualmente se tiene una gran variedad de aceites comestibles por lo que se necesita varios mecanismos de protección contra factores como la luz, humedad y transporte lo que puede volver complejo el sistema de producción de este producto.[8]

1.10 Herramientas de Análisis y Evaluación de Procesos

En este análisis se considerarán flujos continuos, tomando en cuenta las entradas, salidas y el stock dentro del área de evaluación. Este enfoque permite observar el comportamiento del sistema de manera detallada y en secuencia, lo que facilita tanto su seguimiento como la identificación de patrones y áreas de mejora.

Al analizar estos flujos de forma integral, se obtiene una visión clara de cómo interactúan las distintas variables en el sistema, lo que permite realizar ajustes precisos para optimizar su desempeño. [9].

En este caso, el análisis se enfoca en identificar y priorizar los productos con mayor impacto en el nivel de stock, con el propósito de optimizar la gestión de inventarios y garantizar la disponibilidad de los productos más críticos. Este enfoque permite mantener un equilibrio entre el nivel de stock y la calidad del servicio, asegurando una utilización eficiente de los recursos mientras se satisfacen las necesidades del cliente de manera efectiva. [10].

1.11 Industria 4.0

La Industria 4.0 se refiere al proceso de digitalización de la Industria, dado que los métodos y sistemas actuales están alcanzando su capacidad máxima. Se están implementando innovaciones como Aplicaciones, redes Low Power que se tendrán que gestionar a través de la Big Data por lo que la interconexión entre las máquinas y los softwares son de gran ayuda.

El internet de las cosas es un pilar importante, pues es donde todo va a estar conectado lo que a su vez genera grandes volúmenes de datos el cual va a ser manejado gracias a Big Data Analytics, Business-Intelligence y analíticas Predictivas [6]. Podemos ver los componentes de la Industria y Logística 4.0 en la Ilustración 2.

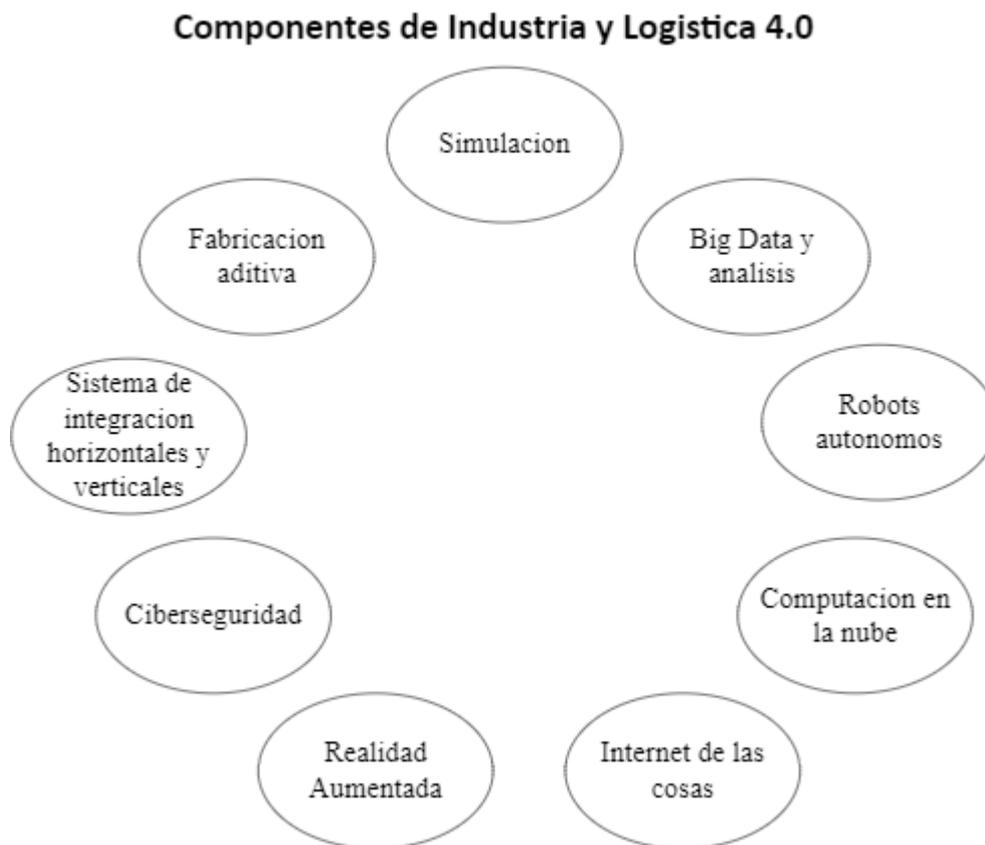


Figura 2. Componentes de Industria y Logística 4.0[6]

1.12 Áreas de aplicación de la Inteligencia Artificial

La Inteligencia Artificial (IA) se ha vuelto altamente relevante en la actualidad, gracias a los numerosos beneficios que aporta para resolver problemas en múltiples áreas. Su capacidad multifacética la posiciona como una herramienta esencial para mejorar procesos, tomar decisiones informadas mediante datos y aumentar la eficiencia en una amplia gama de sectores.

En la tabla siguiente, se incluyen ejemplos destacados de las principales aplicaciones de la Inteligencia Artificial, resaltando su influencia en distintos ámbitos.

Tabla 3. Aplicaciones Inteligencia Artificial [6]

Área	Aplicaciones
Ingeniería	Organización de la producción Optimización de procesos Cálculo de estructuras Planificación y logística Diagnóstico de fallos Toma de decisiones
Robótica y automática	Sistemas adaptativos de rehabilitación Interfases cerebro-computadora Sistemas de visión artificial Sistemas de navegación automática
Física y matemáticas	Demostración automática de teoremas Análisis cualitativo de sistemas no-lineales Caracterización de sistemas complejos
Informática	Procesado de lenguaje natural Criptografía Teoría de juegos Linguística computacional
Biología	Análisis de estructuras biológicas Genética médica y molecular
Economía	Análisis financiero y bursátil Análisis de riesgos Estimación de precios en productos derivados Minería de datos Marketing y fidelización de clientes
Medicina	Ayuda al diagnóstico Análisis de imágenes biomédicas Procesado de señales fisiológicas

Cómo se muestra en la Tabla, el uso de la Inteligencia Artificial (IA) es altamente útil en múltiples áreas. En el ámbito de la Ingeniería, uno de los temas más relevantes es la optimización de procesos, ya que la IA permite identificar mejoras significativas en las operaciones. Esto puede traducirse en un ahorro considerable de recursos, cómo dinero y tiempo, dentro de los procesos de manufactura, lo que la convierte en una herramienta indispensable para mejorar la eficiencia y la rentabilidad en el sector.

1.13 Simulación y Optimización

La simulación y la optimización son herramientas fundamentales en la ingeniería y las ciencias aplicadas para modelar, analizar y mejorar el desempeño de sistemas complejos. En este apartado teórico, se analizarán los conceptos, fundamentos relacionados. y aplicaciones de tres técnicas clave: la simulación de eventos discretos utilizando software cómo Flexsim y la optimización de sistemas a través de algoritmos potenciados por Machine Learning.

1.13.1 Simulación de Eventos Discretos

La simulación de eventos discretos (Sistemas donde los cambios ocurren en puntos específicos del tiempo), representando procesos secuenciales y estados definidos es una técnica de modelado que permite representar el comportamiento dinámico de sistemas en los que los cambios de estado ocurren en instantes de tiempo específicos. Software cómo Flexsim facilitan la creación de modelos detallados de sistemas reales, permitiendo evaluar diferentes escenarios y tomar decisiones informadas.

- Flexsim: Se destaca por su interfaz visual intuitiva y su capacidad para modelar sistemas en 3D. Es ampliamente utilizado en la industria manufacturera y logística para optimizar procesos, diseñar plantas y evaluar diferentes configuraciones.

1.13.2 Integración y Simulación

La integración de la simulación de eventos discretos con técnicas de optimización respaldadas por Machine Learning representa una metodología efectiva para abordar desafíos complejos en el ámbito de la ingeniería. Mediante el uso de modelos impulsados por Machine Learning, se puede determinar la configuración óptima de un sistema que maximice o minimice una métrica específica de rendimiento.

Proceso:

1. Construcción del modelo de simulación: Se desarrolla un modelo detallado del sistema.
2. Definición de la función objetivo: Se establece la métrica que se desea optimizar cómo tiempo de ciclo, utilización de recursos o el costo total.
3. Configuración del algoritmo potenciado por Machine Learning: Se ingresa la base de datos, entrenamiento del algoritmo con la base de datos.
4. Ejecución de la optimización: El modelo de simulación busca la mejor solución hasta encontrar la configuración óptima.

1.13.3 Ejemplos de Aplicaciones

Tabla 4. Aplicación Machine Learning[10]

Aplicación	Explicación
Diseño de líneas de producción	Optimizar la disposición de las máquinas, el tamaño de los lotes y las políticas de programación para minimizar el tiempo de ciclo y maximizar la producción
Planificación de la logística	Optimizar las rutas de transporte, la asignación de recursos y la gestión de inventarios para reducir costos y mejorar la eficiencia
Diseño de sistemas de salud	Optimizar la distribución de recursos, la programación de citas y la gestión de camas de hospital para reducir los tiempos de espera y mejorar la calidad de la atención.

La simulación de eventos discretos y el algoritmo potenciado por Machine Learning son herramientas complementarias que permiten abordar problemas complejos de ingeniería de manera efectiva. Al integrar estas técnicas, es posible obtener soluciones óptimas para una amplia variedad de sistemas.

Capítulo II

2 MÉTODOS Y MATERIALES

2.1 Descripción de la planta embotelladora

La planta embotelladora en la cual se evalúa el algoritmo, se encuentra ubicada en Quito Ecuador cantón Rumiñahui, avenida Ilalo y Rio Curaray. La empresa se encarga del embotellamiento de aceites comestibles previamente fabricados, para luego la distribución a los diferentes canales de venta y posterior consumo.

La planta está diseñada para la producción y distribución de aceite comestible en diferentes presentaciones, adaptándose a las necesidades del mercado y consumidores. Opera con un enfoque en la eficiencia, flexibilidad y optimización de recursos. Pues al disponer de un espacio limitado y poco personal este se vuelve un desafío operativo que influyen en el apareamiento de cuellos de botella.

Actualmente cuenta con 10 colaboradores entre personal administrativo y operativo, en el campo operativo 3 personas se encargan de la producción de aceites en las 3 diferentes líneas, 1 persona para el armado de cartones, 2 personas para el etiquetado de las botellas vacías para su posterior llenado.

Las personas encargadas del llenado también llevan los productos a su zona de almacenamiento y entregan el producto terminado al cliente.

2.2 Infraestructura de la Planta.

La planta cuenta con una estructura en 3 líneas de producción unidas a un solo módulo de almacenamiento con capacidad de 55 toneladas su distribución. En la tabla 2 se describe las características de las líneas de producción:

Tabla 5. Características Planta Embotelladora (Fuente: Elaboración propia)

Tarea	Característica	Capacidad máxima (unidades)	Tasa de producción (unidades/segundo)
1	Botella de 1 litro	55,000	0.40
2	Fundas de 1 litro	55,000	0.25
3	Bidones de 20 litros	55,000	0.50

Los trabajadores realizan el empaquetado para así garantizar que el producto que sale sea empaquetado 12 botellas de 1 litro en una caja y 12 fundas de 1 litro por caja en el caso de los Bidones van directo a los pallets.

Posterior al empaquetado las cajas se colocan en pallets hasta llegar a la capacidad máxima de 60 cajas por pallet y en el caso de los bidones 48 bidones por pallet.

Cuando los pallets se encuentran llenos estos son trasladados a la zona de almacenamiento para su posterior despacho al cliente destinado.

El proceso de producción es dinámico y en casos este tiende a detenerse hasta el que operador de la línea de producción lleve el producto a su zona de almacenamiento por lo tanto el uso de una persona es necesario para que el proceso no se detenga.

Seria notorio para el colaborador encargado de la bodega notar cuando un pallet se encuentre lleno y retirar, pero si existiera una producción variable o más líneas de producción esta tarea empieza a tener su dificultad por lo que un posterior análisis es un método adecuado para evitar que se generen colas y entrenar al operador cómo sería la correcta forma de retirar los pallets y proceder a ubicarlos.

2.3 Loyaut de la línea de producción.

En la Ilustración se observa el proceso de embotellamiento de aceites, el cual cuenta con un total de 4 personas ubicadas en puestos claves para el proceso. 3 personas encargadas de mantener el flujo de producción en cada línea correspondientemente y una cuarta encargada de manejar el montacargas de mano quien va a ser el encargado de recibir la orden en el cual se van a recoger el producto terminado para su posterior despacho.

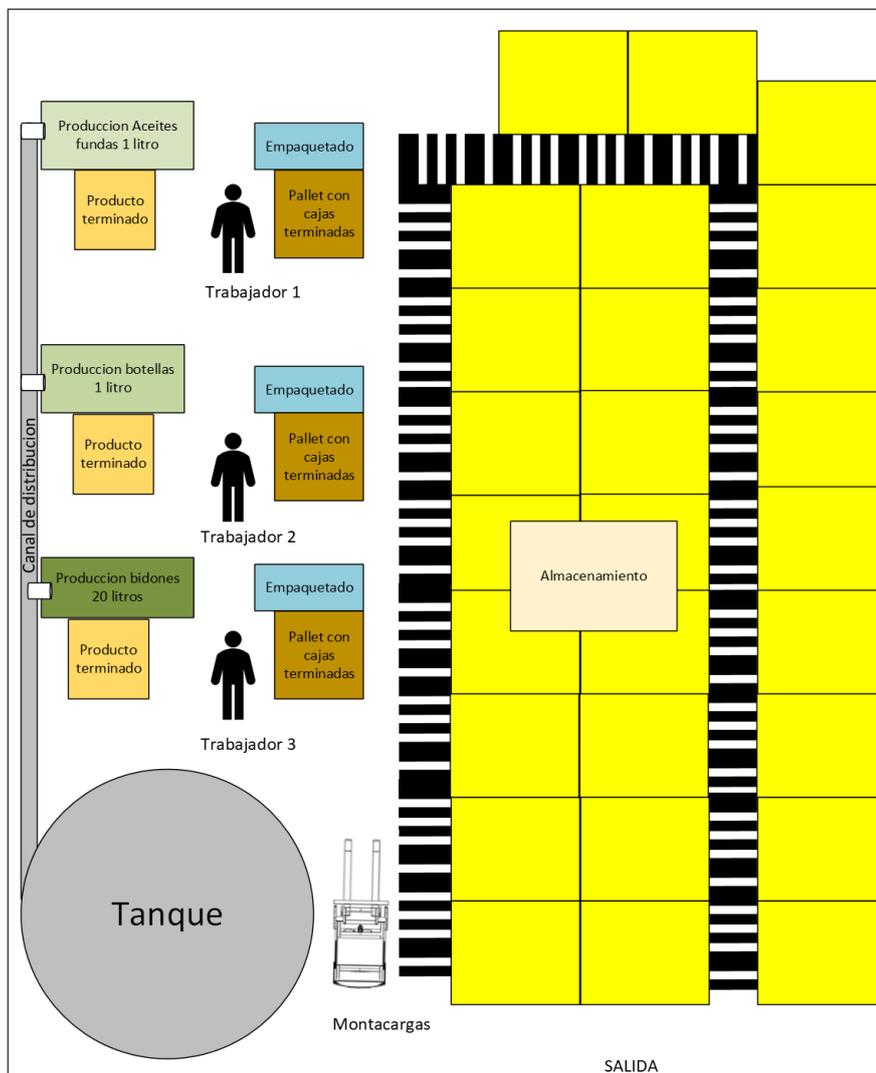


Figura 3. Loyaut Planta Embotelladora (Fuente: Elaboración propia)

2.4 Inteligencia Artificial

2.4.1 Generalidades

Dentro de la metaheurística se tiene varios tipos de algoritmos como los algoritmos genéticos, Optimización por enjambre de partículas (PSO), Optimización por colonias de hormigas (ACO) entre otros.

En este caso en particular para su optimización se va a validar el algoritmo cuello de botella móvil (CBM) potenciándolo con Machine Learning pues es un modelo que busca soluciones optimas tomando en cuenta muchas variables.

2.4.2 Implementación de Machine Learning al Algoritmo Cuello de Botella Móvil

Dentro de la implementación de Machine Learning al algoritmo Cuello de Botella Móvil se realiza iteraciones utilizando Prompts en Chat Gpt que a medida que generaba un algoritmo nos daba errores que a su vez eran solucionados.

A su vez Matlab que cuenta con “Statistics and Machine Learning Toolbox” brinda la posibilidad de utilizar Machine Learning de forma sencilla la cual se adapta a nuestras necesidades.

2.4.3 Funciones utilizadas

Evaluación.

```
modelo = fitctree(XTrain, yTrain); % Entrenar un árbol de decisión
yPred = predict(modelo, XTest); % Validar el modelo con datos de prueba
precision = sum(yPred == yTest) / numel(yTest) * 100; % Calcular precisión
fprintf('Precisión del modelo: %.2f%%\n', precision);
```

Figura 3. Algoritmo Machine Learning (Fuente: Elaboración propia)

Se realiza tres operaciones principales para entrenar, validar y evaluar un modelo de clasificación utilizando un árbol de decisión.

Entrenamiento del modelo.

```
modelo = fitctree (XTrain, yTrain);
```

- fitctree es una función de MATLAB que crea un modelo de árbol de decisión para clasificación.
- XTrain contiene las características de los datos de entrenamiento.
- yTrain contiene las etiquetas correspondientes a esos datos de entrenamiento.
- El modelo resultante se guarda en la variable modelo.

Validación del modelo.

```
yPred = predict (modelo, XTest);
```

- predict utiliza el modelo entrenado (modelo) para hacer predicciones sobre nuevos datos (XTest), que son los datos de prueba.
- El resultado son las etiquetas predichas (yPred) para los datos de prueba.

Cálculo de la precisión.

```
precision = sum (yPred == yTest) / numel(yTest) * 100.
```

- yPred == yTest compara las etiquetas predichas (yPred) con las etiquetas reales de los datos de prueba (yTest), devolviendo un vector lógico donde los valores true corresponden a predicciones correctas.
- sum (yPred == yTest) cuenta cuántas predicciones fueron correctas.
- numel(yTest) obtiene el número total de datos en el conjunto de prueba.

- La precisión se calcula cómo el porcentaje de predicciones correctas.

Por lo tanto, esta sección del algoritmo brinda la posibilidad de entrenarlo y validar su precisión para su posterior uso, utilizando un árbol de decisión con los datos de entrenamiento y poder realizar predicciones sobre datos de prueba.

2.4.4 Interpretación grafica del Algoritmo Cuello de Botella Móvil potenciado por Machine Learning

El algoritmo, al ser ejecutado, ofrece la posibilidad de observar su funcionamiento mediante un árbol de decisiones, lo cual permite visualizar de manera clara cómo se toman las decisiones en cada etapa del proceso. La implementación de esta funcionalidad es sencilla y se realiza utilizando una función específica.

Árbol de decisiones

```
view (modelo, 'Mode', 'graph'); % Modo gráfico
```

Esta función, al ser colocada al final del código, genera una representación gráfica del árbol de decisiones, facilitando el análisis y comprensión del comportamiento del algoritmo. A continuación, se presenta una muestra del árbol generado.

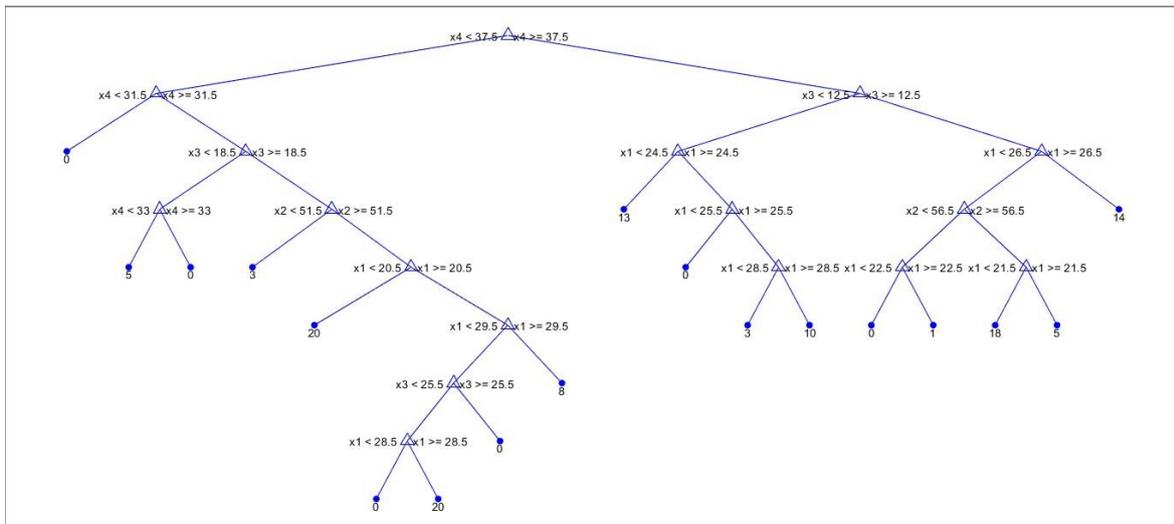


Figura 5. Árbol de Decisiones(Fuente: Elaboración propia)

Estructura del Árbol

Tabla 6 Estructura Árbol de Decisiones (Fuente: Elaboración propia)

Nodos Internos	Hojas	Ramas
Los triángulos representan los nodos de decisión del árbol.	Los círculos en la parte inferior del árbol representan las hojas.	Las líneas conectan los nodos internos y las hojas, representando el flujo de las decisiones tomadas en cada paso
Cada nodo interno contiene una condición basada en una de las características de entrada (en este caso, x_1 , x_2 , x_3 , o x_4).	Cada hoja corresponde a una clase o valor de salida determinado por las condiciones acumulativas desde la raíz hasta esa hoja.	Cada rama indica el resultado de una comparación: si se cumple la condición (rama izquierda) o si no se cumple (rama derecha).
Las condiciones se presentan en forma de comparaciones, como $x_4 < 31.5$. Estas dividen los datos en subconjuntos en función del resultado de la condición (verdadero o falso).		

2.5 Herramientas de Análisis

Para la evaluación del algoritmo se utiliza las siguientes herramientas.

2.5.1 Simulink

Se utiliza MATLAB y Simulink para modelar el sistema y validar el algoritmo utilizando Machine Learning, pues al ser un entorno que posibilita la simulación de procesos complejos utilizando bloques donde estos mismo pueden ser programados e incluir código que permite su configuración. Asimismo, facilita la evaluación del cuello de botella utilizando el mismo modelo.

2.5.2 Flexsim

Se utiliza FlexSim para la simulación de los eventos en la planta embotelladora de aceites, ya que esta herramienta ofrece un entorno intuitivo y fácil de usar para la creación y simulación de modelos. FlexSim permite visualizar y comprender cómo funciona el proceso de manera clara y detallada.

Una de las mayores fortalezas de esta herramienta radica en su capacidad para visualizar de manera clara dónde y cómo se generan las filas en el sistema, permitiendo identificar puntos críticos y oportunidades de optimización en el flujo operativo. Esta funcionalidad convierte a FlexSim en una solución fundamental para mejorar procesos en instalaciones industriales.

2.6 Estructura del modelado de sistema.

Para el modelado del sistema dentro de Simulink se utilizan bloques que son los encargados de dar forma modelo. En nuestro modelo utilizamos las fases de un proceso donde tenemos las entradas, el proceso y nuestras salidas de la siguiente forma.

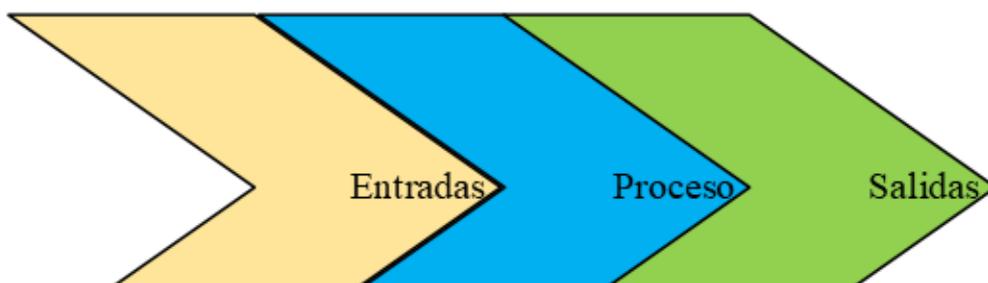


Figura 6. Estructura modelado de sistema(Fuente: Elaboración propia)

2.7 Bloques Utilizados.

En la siguiente tabla se indican los Bloques utilizados por fases.

Tabla 7 Bloques utilizados Simulink (Fuente: Elaboración propia)

Posibles Entradas	Proceso	Salidas
Constant	MATLAB Function	Display
 Constant Simulink — also in HDL Coder	 MATLAB Function Simulink/User-Defined Functions	 Display Simulink — also in HDL Coder

2.7.1 Explicación de los bloques.

Constant.

El bloque Constant tiene como objetivo proporcionar un valor constante como entrada al sistema, durante la simulación este bloque representa la tasa de producción por línea.

Este bloque es una posible entrada pues también se pueden agregar otros bloques que ayuden a brindar datos variables como distribuciones normales, distribuciones de Poisson, distribuciones uniformes y así el algoritmo tenga la posibilidad de solucionarlo.

MATLAB Funtion.

El bloque MATLAB Function tiene como objetivo ejecutar funciones o algoritmos personalizados escritos en MATLAB. Permite integrar cálculos avanzados, lógica condicional

y optimizaciones. Durante la simulación en este bloque es el lugar donde se incluye el algoritmo cuello de botella móvil para su respectiva evaluación.

Display.

El bloque Display tiene como objetivo mostrar los valores de las señales durante la simulación. Durante la simulación el bloque nos mostraba el orden en el cual se opera las líneas de producción y también indicar el tiempo estimado para completar la producción.

CAPITULO III

3 RESULTADOS Y DISCUSIÓN

3.1 Formulación del problema

Para la evaluación del algoritmo, se desarrolla un ejemplo práctico basado en una planta embotelladora de aceites, tal cómo se menciona en la unidad anterior. El objetivo principal es formular el problema de manera precisa y emplear herramientas adecuadas para su representación.

Mediante el uso del grafo, se analiza de manera estructurada la distribución del proceso de embotellado de aceites, identificando las distintas etapas y sus interacciones. Este enfoque facilita la comprensión del funcionamiento del sistema, permitiendo la identificación de posibles cuellos de botella y la optimización del rendimiento del proceso.

3.2 Grafo del proceso de manufactura

Se utiliza el grafo disyuntivo, el cual es utilizado en la programación de tareas en sistemas de producción tipo Job Shop (Sistema de producción flexible que fabrica productos personalizados en lotes pequeños con flujos de trabajo variables) o en problemas de optimización.

Este tipo de representación gráfica incluye nodos (vértices), que representan las operaciones a realizar, y arcos (aristas), que indican la precedencia entre operaciones, señalando que una operación debe completarse antes de iniciar la siguiente.

Para la evaluación del algoritmo, se considera la siguiente secuenciación de trabajo. Esta secuencia describe las tareas a realizar, las rutas que sigue el aceite desde su almacenamiento en tanques hasta su almacenamiento final en bodegas cómo producto terminado, y los tiempos de procesamiento asociados. El proceso incluye los trabajos:

Tabla 8 Capacidad de Llenado (Fuente: Elaboración propia)

Trabajo 1	Botellas de 1 litro
Trabajo 2	Fundas de 1 litro
Trabajo 3	Bidones 1 litro

El proceso de manufactura utiliza 10 máquinas.

Tabla 9 Actividad de Máquinas (Fuente: Elaboración propia)

Máquina 1	Cola botellas de 1 litro
Máquina 2	Cola fundas de 1 litro
Máquina 3	Cola bidones de 20 litros
Maquina 4	Cola de cajas
Maquina 5	Cola de Cajas con Botellas
Maquina 6	Cola de Cajas con bolsas
Maquina 7	Cola de Pallets
Maquina 8	Llenado de botellas de 1 litro
Maquina 9	Llenado de fundas de 1 litro
Maquina 10	Llenado de bidones de 20 litros

Las tareas no necesariamente incluyen todas las máquinas, a continuación, se muestra la ruta que siguen los trabajos en la planta de embotellamiento.

Tabla 10 Ruta de trabajos (Fuente: Elaboración Propia)

Trabajos	Ruta
Botellas	M8-M1-M13-M11-M4-M16-M5-M12-M7-M19
Fundas	M9-M2-M14-M17-M4-M11-M6-M20-M7-M12
Bidones	M10-M3-M15-M18-M7-M12

3.3 Simulación del proceso de manufactura

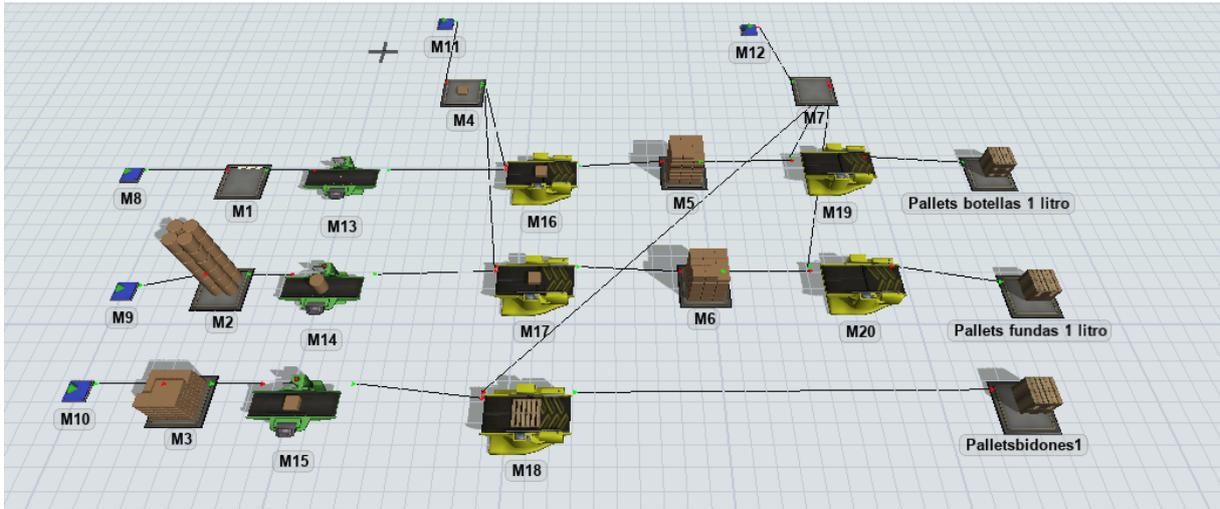


Figura 4. Simulación embotellado de aceites (Fuente: Elaboración Propia)

Para evaluar el sistema de manufactura, se desarrolla una simulación utilizando la aplicación FlexSim. En esta simulación, se organiza diversos bloques de manera estratégica para replicar el proceso de embotellado de aceites.

El modelo incluye bloques que representan las distintas etapas y componentes del sistema, proporcionando una representación detallada del flujo de trabajo. Esta configuración permite analizar el rendimiento del proceso y evaluar su eficiencia.

En la tabla correspondiente se detalla el nombre del bloque, el asignado a cada componente y su descripción, para aclarar su función dentro del modelo.

El flujo del proceso se produce de la siguiente manera:

1. Fluido inicial es suministrado desde el Tráiler Cisterna hacia el Tanque.
2. El fluido se transforma en botellas, fundas, y bidones en sus respectivas líneas de llenado
3. Las unidades llenas se almacenan en colas antes de ser empaquetadas (Queue).
4. Los bloques Empaquetado combinan los productos con cajas hasta llenar un pallet para formar los lotes finales.
5. Los operadores transportan los lotes empaquetados o bidones al área de salida.

3.3.1 Situación inicial de la planta

En el proceso de toma de datos dentro de la planta, se obtiene el promedio de producción por la línea de trabajo. Los resultados incluyen los siguientes valores clave, que permiten analizar el rendimiento de cada línea y detectar áreas de mejora. Estos datos son fundamentales para realizar un diagnóstico preciso y plantear estrategias de optimización.

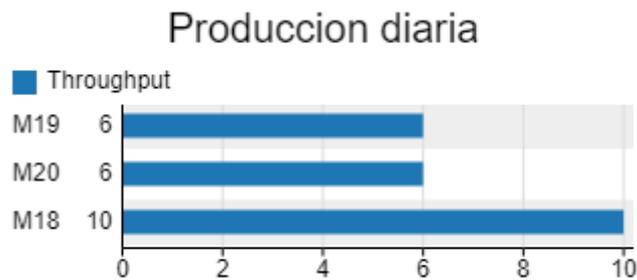


Figura 10. Producción Inicial Diaria (Fuente: Elaboración Propia)

Para la producción diaria se toma 8 horas de trabajo y límites en cuanto a producción, el almacenamiento máximo es de 28 pallets.

Luego de cada máquina de producción se tiene espacios de almacenamiento lo que da lugar a colas, en la simulación se da cola en específico en 3 espacios estos son las máquinas M1-M2-M3 las cuales son colas post producción sin empaquetado.

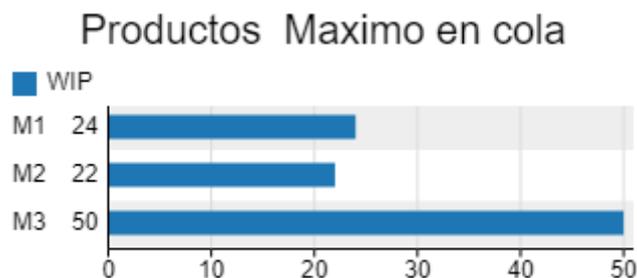


Figura 10. Productos en cola sin aplicar optimización (Fuente: Elaboración Propia)

Se observa que las principales máquinas que tienden a tener cuello de botella son las máquinas M1, M2, M3 son máquinas que cargan con los cuellos de botella por lo que optimizando máquinas subsecuentes ayudaríamos con estos cuellos de botella.

Para un análisis completo también tomaremos un indicador adicional el cual es el de tiempo promedio de un producto en cola (AvgStaytime)

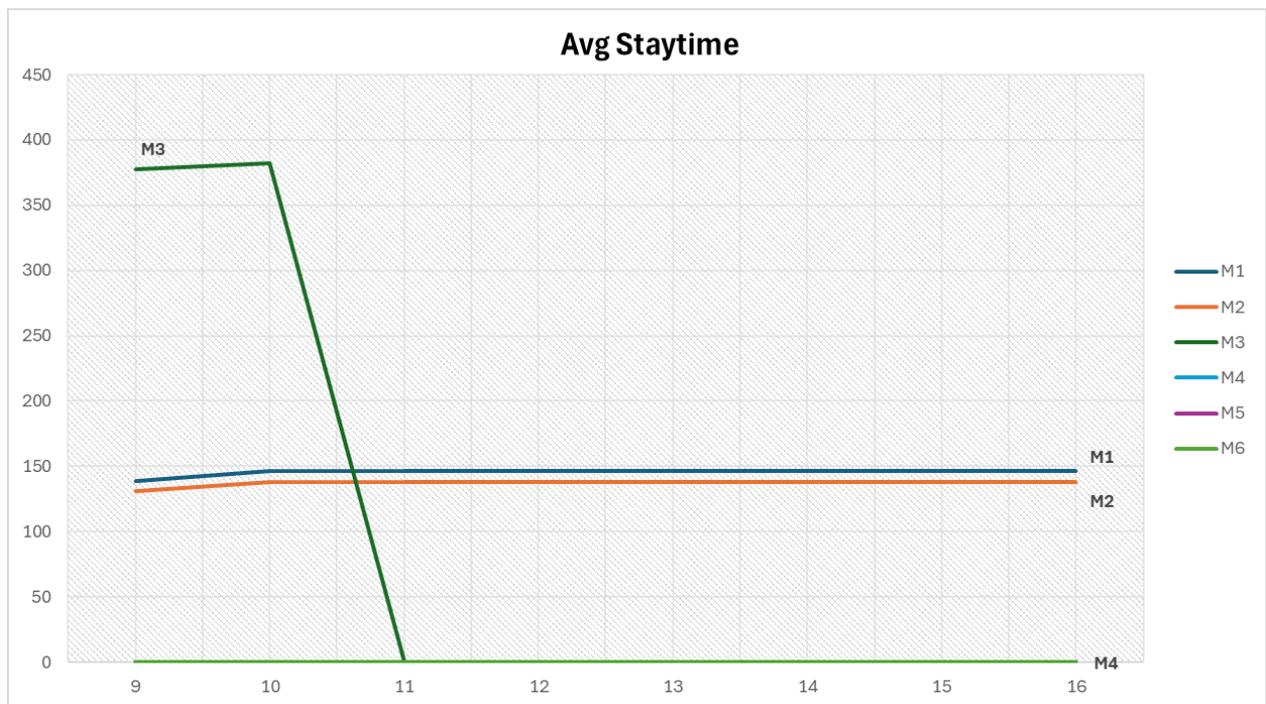


Figura 11. AvgStaytime sin aplicar optimización

En la figura 11 se observa que la estación M3 tiene un tiempo de espera superior a 350min y luego baja a 0 cerca de las 12pm pues esto indica que a esa hora su producción terminó, en cambio las máquinas M1, M2 son máquinas que continúan trabajando y su promedio de espera es cercana a los 150 minutos.

Para su optimización se realiza la evaluación con el algoritmo cuello de botella móvil luego se realiza la evaluación utilizando el algoritmo cuello de botella móvil pero esta vez potenciado por Inteligencia Artificial.

3.4 Evaluación utilizando algoritmo Cuello de Botella Móvil

Al evaluar el algoritmo CBM (Cuello de Botella Móvil) se tiene las siguientes sugerencias:

Priorizar máquinas en el siguiente orden: [10 11 12 9 8 13 14 15]

Los resultados nos indica a que máquinas se tiene que dar prioridad siendo la primera la que mayor prioridad hay que dar y la ultima la que menor prioridad se tiene dar.

Aumentamos en un 100% la producción de cada una.

La razón por la cual se aumenta un 100% es porque se incrementa otra persona en el proceso, luego de incrementar la producción en las máquinas 11 y 12 se incrementa la producción dando los siguientes resultados;

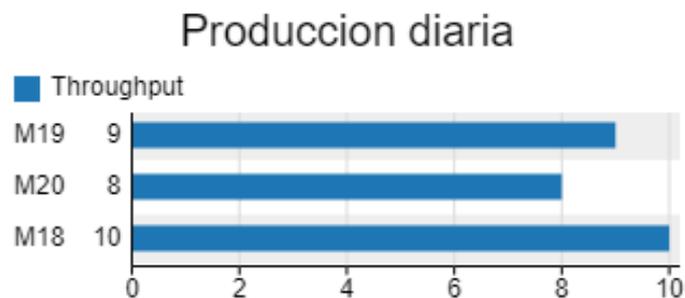


Figura 11: Producción Diaria con CBM (Elaboración Propia)

Esto resulta en un incremento de la producción del 22.73%, lo que demuestra que el algoritmo cumple su función de manera efectiva al optimizar la planta de procesamiento de aceites.

Dentro de la producción se identifica que, al optimizar, aumenta el número de máquinas con colas dentro de sus procesos. Este fenómeno sugiere que la optimización redistribuye las limitaciones operativas, lo que genera acumulaciones en otras partes del flujo de trabajo.

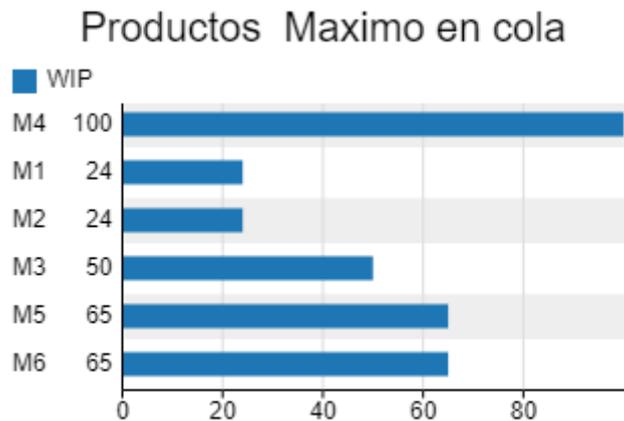


Figura 12. Productos máximos en cola con CBM

El aumento en la producción de cajas en la estación M11 provoca un incremento en las colas de las estaciones M4, M5 y M6. Esto genera que las máquinas M16, M17 y M18 trabajen de manera continua, sin interrupciones por falta de material.

Para analizar el impacto de estos cambios, se observa el tiempo de espera promedio (AvgStaytime) de los productos en la Figura 13, donde se detallan los resultados.

Tabla 11 AvgStayTime luego de CBM (Fuente: Elaboración Propia)

Object	Avg Staytime	Max Staytime
M1	65.90	96
M2	22.00	44
M3	379.45	730
M4	1034.19	4362
M5	0.30	20
M6	0.17	12

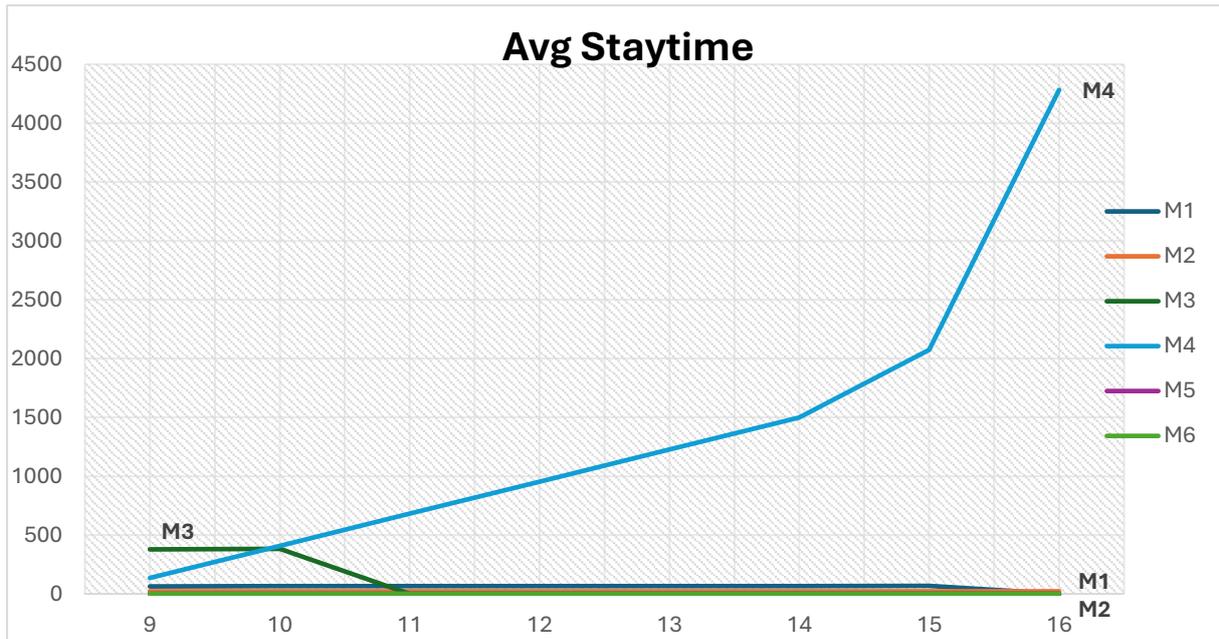


Figura 13 AvgStaytime con CBM

Se observa que la estación M3 toma protagonismo al aumentar su producción, lo que indica que incrementar la producción en la estación M11 provoca que otras máquinas también intensifiquen su trabajo. Esto demuestra que el Algoritmo CBM (Cuello de Botella Móvil) es una herramienta eficaz para optimizar sistemas de manufactura, al identificar y ajustar dinámicamente los puntos críticos en el flujo de trabajo.

3.5 Evaluación utilizando algoritmo Cuello de Botella Móvil utilizando Inteligencia Artificial

En el proceso de evaluación, partiremos con 8 máquinas con el objetivo de verificar si el modelo está en la capacidad de aprender

El algoritmo analiza los datos históricos de producción, identifica cuellos de botella usando Machine Learning y genera un plan de trabajo optimizado para mejorar cómo la eficiencia del proceso.

El algoritmo muestra las órdenes de trabajo generadas para cada hora, ordenando las máquinas según la prioridad basada en el tiempo promedio de permanencia (AvgStaytime).

Precisión del modelo: 66.67%
Modelo entrenado y guardado.
Órdenes de trabajo por hora:
Hora 1: Priorizar máquinas en el siguiente orden: [10 11 12 9 8 13 14 15]

El porcentaje de precisión que el modelo alcanzó en la predicción durante su validación. Una precisión del 66.67% sugiere que aproximadamente dos tercios de las predicciones del modelo son correctas.

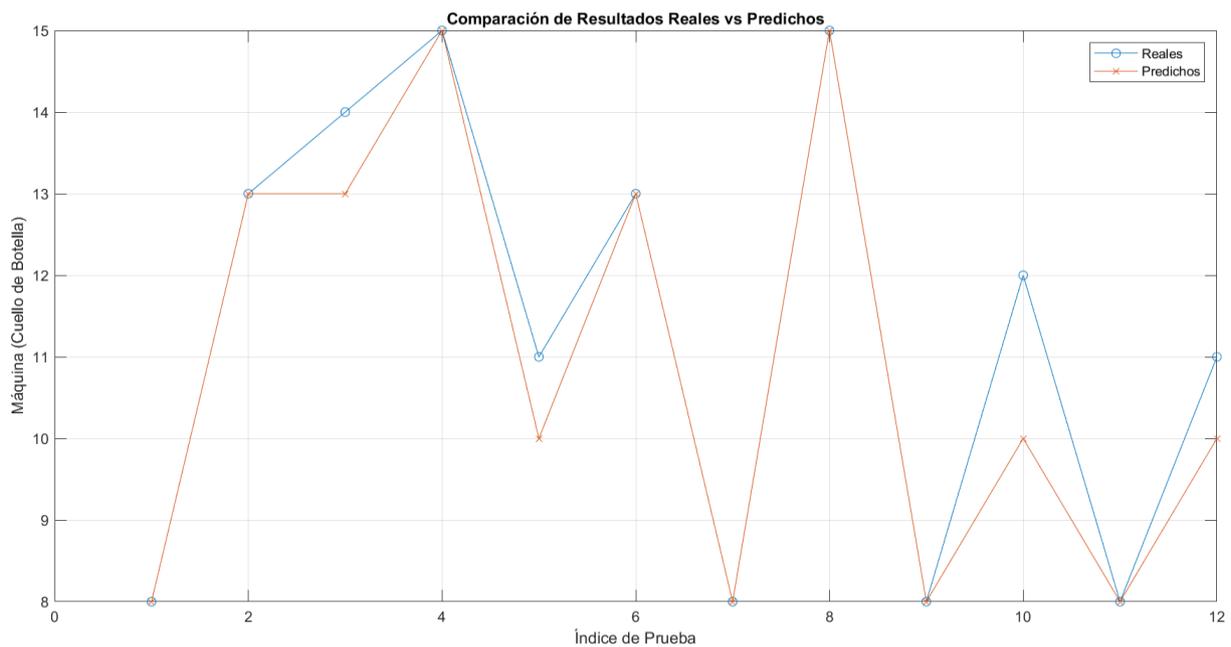


Figura12. Comparación resultados reales vs predichos 8 máquinas (Fuente: Elaboración Propia)

El gráfico muestra cómo el algoritmo puede dar falsos positivos pues el modelo predice un cuello de botella que no estaba presente en los datos reales y falsos negativos pues modelo no identificó un cuello de botella que estaba presente en los datos reales.

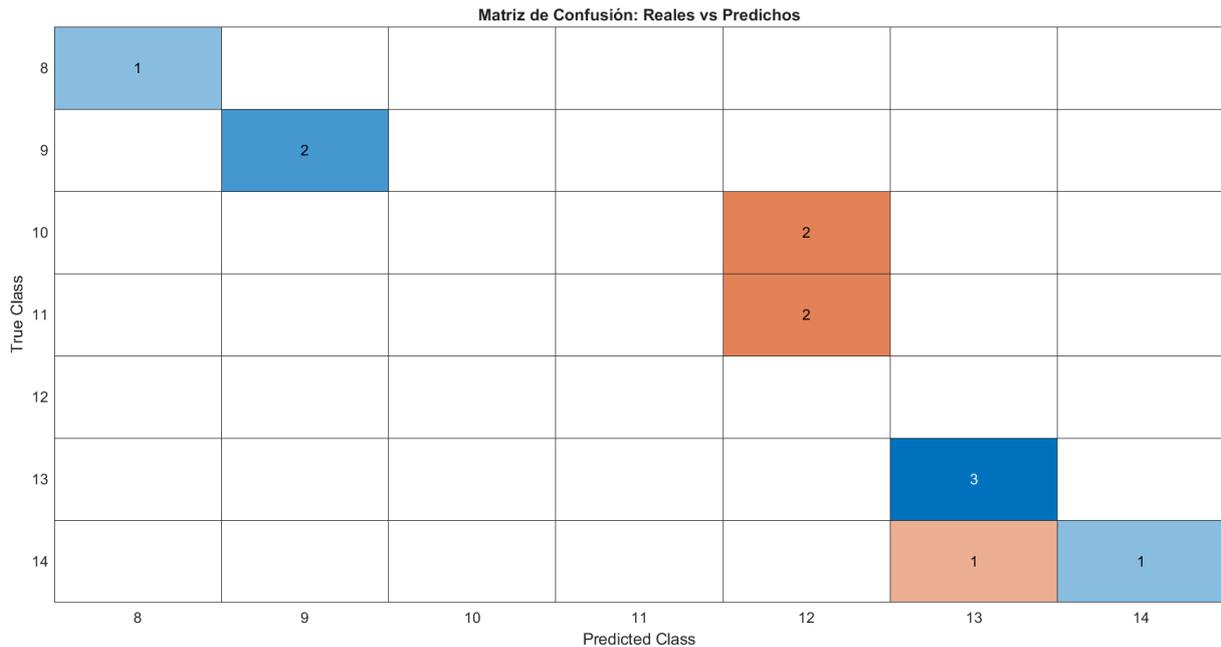


Figura 13. Matriz confusión 8 máquinas (Fuente: Elaboración Propia)

La matriz de confusión proporciona una representación visual del desempeño del modelo de Machine Learning en la clasificación de las máquinas que generan cuellos de botella.

El eje Y ("True Class") representa las clases reales, es decir, las máquinas identificadas como cuellos de botella en los datos reales.

El eje X ("Predicted Class") corresponde a las clases predichas por el modelo.

Cada celda muestra la cantidad de veces que una clase real fue clasificada como otra clase. Por ejemplo:

En la intersección de las filas y columnas etiquetadas como "9", el modelo predice correctamente dos veces que la máquina "9" era un cuello de botella.

Sin embargo, también se observa un error, ya que la máquina "14" es incorrectamente clasificada como "13" en una ocasión.

Esta matriz es fundamental para evaluar el desempeño del modelo, permitiendo identificar:

Aciertos: Representados en la diagonal principal, donde las predicciones coinciden con las clases reales.

Errores de clasificación: Representados fuera de la diagonal, donde las predicciones no coinciden con las clases reales.

```

Precisión del modelo: 98.57%
Modelo entrenado y guardado.
El cuello de botella predicho es la Máquina: 1
El cuello de botella predicho es la Máquina: 0
El cuello de botella predicho es la Máquina: 1
Secuencia sugerida para evitar cuellos de botella:
Priorizar Máquina 1
Priorizar Máquina 0
Priorizar Máquina 1
Número de errores: 3
Eficiencia del modelo: 98.57%
Ingrese el número de pedidos para cada máquina [M1, M2, M3]: [100, 100, 100]
Tiempo estimado para completar los pedidos en Máquina 1: 76.92 minutos
Tiempo estimado para completar los pedidos en Máquina 2: 88.24 minutos
Tiempo estimado para completar los pedidos en Máquina 3: 93.75 minutos
Precisión del modelo: 98.57%
Tasa de error del modelo: 1.43%
    
```

Figura 14. Resultados machine Learning con 3 máquinas (Fuente: Elaboración Propia)

Al evaluar el modelo su tiempo de ejecución es sumamente corto de tan solo 0.001235 segundos lo que lo hace un algoritmo ágil.

A partir del árbol de decisiones que toma en cuenta la Inteligencia Artificial tenemos lo siguiente.

3.6 Árbol de decisiones

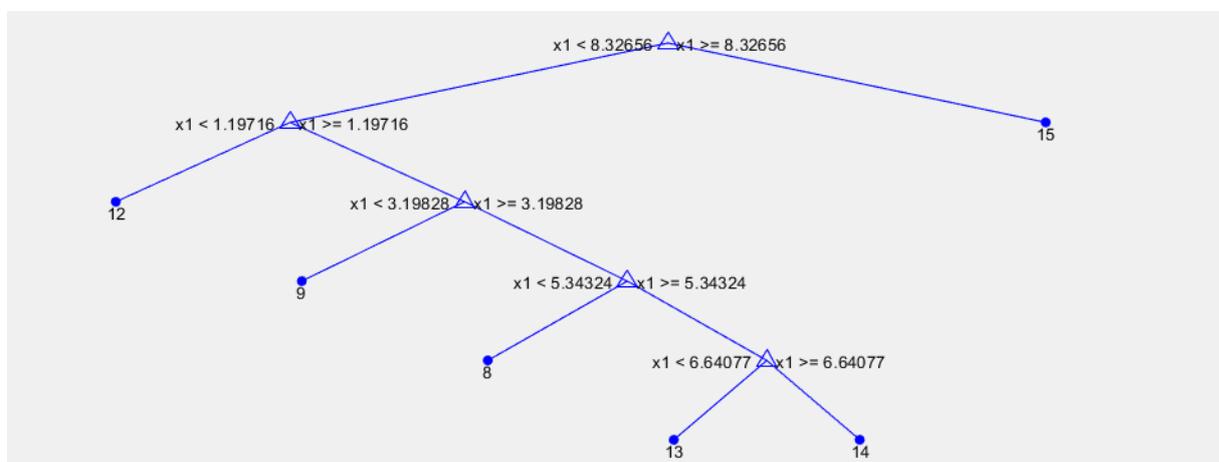


Figura 15. Árbol Decisiones con 8 Máquinas

Cada nodo en el árbol representa una condición basada en un umbral de una variable (en este caso, marcada x_1), que divide los datos en ramas según el valor de dicha característica. Los nodos triangulares indican decisiones, mientras que los nodos finales (hojas) representan las predicciones finales del modelo, etiquetadas con las máquinas identificadas ("12", "9", "15").

3.6.1 Funcionamiento

Nodo Raíz.

El árbol comienza con una decisión en el nodo raíz ($x_1 < 8.32656$), donde los datos se dividen en dos ramas:

Si la condición es verdadera ($x_1 < 8.32656$), cómo el modelo sigue la rama izquierda.

Si es falsa ($x_1 \geq 8.32656$), sigue la rama derecha.

Nodos Intermedios.

Las decisiones intermedias dividen los datos en subconjuntos más pequeños basados en umbrales adicionales ($x_1 < 1.19716$, $x_1 \geq 3.19828$).

Nodos Hoja.

Los nodos hoja (círculos azules) contienen las predicciones del modelo, indicando la máquina clasificada cómo cuello de botella. Por ejemplo:

Si $x_1 < 1.19716$, la máquina clasificada es "12".

Si $x_1 \geq 6.64077$, la máquina clasificada es "14".

Interpretación.

Este árbol es un modelo interpretable, lo que significa que permite entender cómo se toman las decisiones en el proceso de clasificación. Los umbrales reflejan las divisiones en los datos que el modelo considera más relevantes para identificar cuellos de botella. Por ejemplo, los nodos

más cercanos al nodo raíz (cómo $x1 < 8.32656$) son los criterios más importantes, mientras que los nodos más profundos ($x1 \geq 6.64077$) reflejan condiciones más específicas.

3.7 Conclusión

Dentro del sistema de la planta de procesamiento de aceites donde los datos son limitados la precisión de la inteligencia artificial es sumamente alta dado el número de máquinas que son 8 por lo que es factible su uso dentro de ella siempre y cuando se dispongan de los datos para su continuo entrenamiento. Para verificar a profundidad el algoritmo se procede a mejorar la base de datos disponible añadiendo más tipos de datos y así poder comprobar su escalabilidad.

Los datos de optimización son los mismos a los otorgados dentro del algoritmo sin Machine Learning por lo que de igual manera es útil para este tipo de optimizaciones.

Código Fuente utilizado para la evaluación de Cuello de botella utilizando Inteligencia Artificial en Planta de procesamiento de aceite con 8 líneas de producción.

3.8 Explicación

3.8.1 Cargar la Base de datos.

Para la carga de los datos dentro del algoritmo utilizamos el siguiente código

```
baseDatos = readtable('Basedatos.csv'); % Cargar base de datos
disp ('Base de datos cargada:');
disp (head(baseDatos, 10)); % Mostrar las primeras 10 filas
```

El Código carga una base de datos creada en formato .csv, siguiendo la siguiente distribución:

```
Hora, Maquina, FlujoProduccion, CapacidadAlmacenamiento, CargaActual, TiempoRetraso, CuelloBotella
1,1,80.0,60,60.0,5,1
```

La base de datos tiene los datos necesarios para su análisis y esta debe alimentarse para que el algoritmo siga aprendiendo.

Cómo adicional se muestra los 10 primeros datos para tener una referencia de lo que se analiza, este proceso no es necesariamente necesario, pero sirve cómo ayuda.

3.8.2 Preparar los Datos para el modelo.

En esta sección se prepara la base de datos para que sea utilizado por el modelo de Machine Learning, el código a utilizar es el siguiente:

```
X = baseDatos{:, {'FlujoProduccion', 'CapacidadAlmacenamiento', 'CargaActual',  
'TiempoRetraso'}};  
y = baseDatos.CuelloBotella; % Etiqueta (máquina que es cuello de botella)
```

Donde:

- **baseDatos:** Es una tabla que contiene los datos cargados previamente desde un archivo
- **{'FlujoProduccion', 'CapacidadAlmacenamiento', 'CargaActual', 'TiempoRetraso'}:** Estas son las columnas de la tabla baseDatos que se utilizan como características o variables predictoras para el modelo.
- **baseDatos.CuelloBotella:** Selecciona la columna CuelloBotella de la tabla baseDatos

En este apartado tenemos que X es la entrada del modelo, es decir, las características que el algoritmo usa para aprender patrones. Adicional “Y” la cual indica la salida o etiqueta del modelo, que el algoritmo trata de predecir (la máquina que es el cuello de botella).

3.8.3 Dividir los datos en conjunto de entrenamiento (80%) y prueba (20%).

Para realizar la evaluación primero se divide los datos para entrenar, donde el 80% de los valores son para el entrenamiento y el 20% para la prueba. Este paso es fundamental pues es donde el algoritmo se entrena para su posterior uso, para este paso se utiliza el siguiente código:

```
cv = cvpartition(height(baseDatos), 'HoldOut', 0.2);  
XTrain = X(training(cv), :);  
yTrain = y(training(cv), :);  
XTest = X(test(cv), :);  
yTest = y(test(cv), :);
```

3.8.4 Creación de la partición de datos.

cvpartition: Es una función de MATLAB que crea una partición de los datos en dos subconjuntos:

Conjunto de entrenamiento: Se utiliza para entrenar el modelo.

Conjunto de prueba: Se utiliza para evaluar el modelo.

height(baseDatos): Devuelve el número de filas (observaciones) en la tabla baseDatos. Esto indica cuántos ejemplos hay en total.

'HoldOut': Especifica el método de partición. En este caso, se retiene un porcentaje de los datos para el conjunto de prueba.

0.2: Indica que el 20% de las observaciones se reserva para el conjunto de prueba, mientras que el 80% restante se utiliza para el conjunto de entrenamiento.

3.8.5 Seleccionar el conjunto de entrenamiento.

training(cv): Devuelve un vector lógico que indica qué observaciones pertenecen al conjunto de entrenamiento (80% de los datos, en este caso).

X(training(cv), :): Selecciona las filas correspondientes al conjunto de entrenamiento en la matriz X (características).

y(training(cv), :): Selecciona las filas correspondientes al conjunto de entrenamiento en el vector y (etiquetas),

Por lo tanto, XTrain: Contiene las características para el entrenamiento, por otro lado, yTrain: contiene las etiquetas para el entrenamiento.

3.8.6 Seleccionar el conjunto de prueba.

test(cv): Devuelve un vector lógico que indica qué observaciones pertenecen al conjunto de prueba (20% de los datos, en este caso).

X(test(cv), :): Selecciona las filas correspondientes al conjunto de prueba en la matriz X (características).

y(test(cv), :): Selecciona las filas correspondientes al conjunto de prueba en el vector y (etiquetas).

Por lo tanto, XTest: Contiene las características para el conjunto de prueba adicional yTest: Contiene las etiquetas para el conjunto de prueba.

3.8.7 Entrenar el Modelo.

En esta sección se entrena el árbol de decisión para clasificar el cual luego lo valida con datos no vistos anteriormente y posterior calcula la precisión:

```
modelo = fitctree (XTrain, yTrain); % Entrenar un árbol de decisión
```

fitctree: Es una función de MATLAB que entrena un modelo de árbol de decisión para clasificación.

```
yPred = predict (modelo, XTest); % Validar el modelo con datos de prueba
```

predict: Utiliza el modelo entrenado (modelo) para hacer predicciones sobre un nuevo conjunto de datos (XTest).

```
precision = sum (yPred == yTest) / numel(yTest) * 100; % Calcular precisión
```

yPred == yTest: Compara las etiquetas predichas (yPred) con las etiquetas reales del conjunto de prueba (yTest). Devuelve un vector lógico donde true indica que la predicción fue correcta y false indica un error.

sum (yPred == yTest): Cuenta el número de predicciones correctas.

numel(yTest): Calcula el número total de observaciones en el conjunto de prueba.

3.8.8 Carga del modelo y predicción de cuellos de botella.

```
load('modeloEntrenado.mat'); % Cargar el modelo entrenado
```

Con el código se carga el modelo ya entrenado para su posterior uso.

3.8.9 Predecir el cuello de botella.

```
cuelloBotellaPredicho = predict (modelo, estadoActual);  
fprintf ('El cuello de botella predicho es la Máquina: %d\n',  
cuelloBotellaPredicho);
```

En esta sección se utiliza el modelo previamente entrenado para predecir cuales máquinas son cuellos de botella

3.8.10 Generar la Secuencia de Trabajo.

```
fprintf ('Secuencia sugerida para evitar cuellos de botella:\n');  
for i = 1: numel(cuelloBotellaPredicho)  
    fprintf ('Priorizar Máquina %d\n', cuelloBotellaPredicho(i));  
End
```

for i = 1: numel(cuelloBotellaPredicho): Itera por cada predicción en cuelloBotellaPredicho.

numel(cuelloBotellaPredicho) devuelve el número de predicciones (igual al número de máquinas).

3.8.11 Escalabilidad.

Para comprobar la escalabilidad del sistema, se realiza una prueba con 10 máquinas que conlleva a ser la totalidad de las máquinas que cuenta la planta procesadora de aceites. Para ello se realiza la simulación de 1 semana para todas las máquinas y tomando cómo cuello de botella a las máquinas que cuentan con Staytime (Tiempo de permanencia) más alto.

Precisión del modelo: 80.21%

Órdenes de trabajo por hora:

Tabla 12. Resultados Simulación 20 máquinas utilizando CBM implementado Machine Learning

Hora	Maquinas a Priorizar
1	16
2	[19 3 12]
3	[19 1 15]
4	[14 13 18]
5	[16 18 9]
6	[5 13]
7	[6 4 18 16]
8	[7 5 14]
9	[3 7]
10	[4 13 7]
11	[14 8 6 12]
12	[15 7 1 13]
13	11
14	[19 20]
15	[10 14 6]
16	[14 5]
17	[10 14]
18	4
19	[11 6 3 18]
20	2
21	5
22	[17 5]
23	[3 5]
24	[3 13 6]

Esto indicando el orden de prioridad para las 24 horas, al aplicar las prioridades mencionadas esto nos da los siguientes datos:

3.8.12 Máquinas predichas cómo Cuellos de Botella por hora

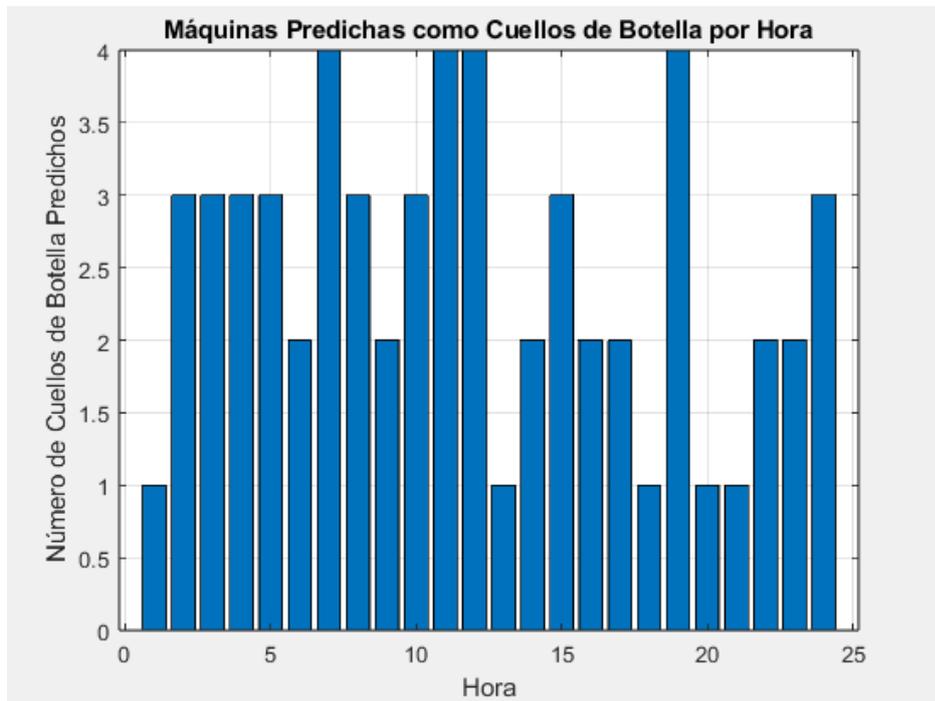


Figura 5. Máquinas Predichas cómo Cuellos de Botella por Hora

Se observa que el algoritmo proporciona información detallada sobre la cantidad de máquinas que presentan cuellos de botella por hora. En el análisis realizado, se identifica que, durante un período de 3 horas, hasta 4 máquinas presentaron cuellos de botella simultáneamente. Sin embargo, en un lapso de 5 horas, solo 1 máquina registró este problema.

Estos resultados sugieren que la aparición de cuellos de botella tiene un comportamiento cíclico, con fluctuaciones en la cantidad de máquinas afectadas a lo largo del tiempo. Este patrón puede ser clave para planificar ajustes y optimizaciones en el sistema de producción, permitiendo mitigar los impactos de estas restricciones de manera más eficiente.

3.8.13 Eficiencia del Proceso por Hora

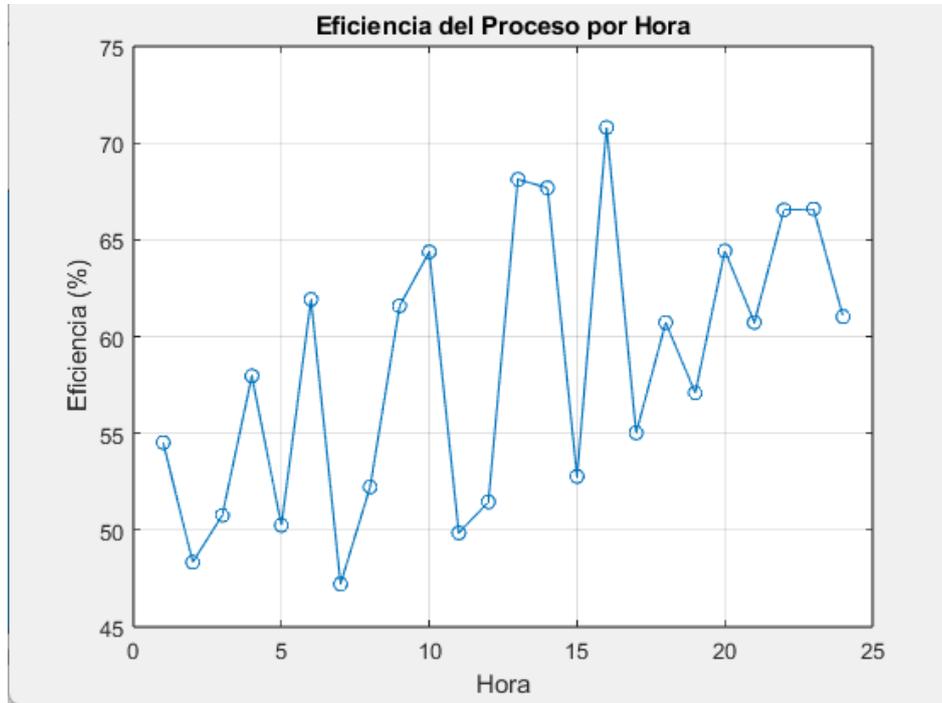


Figura 6. Eficiencia del Proceso por Hora

Tras analizar la comparativa de eficiencia, se observa una relación inversamente proporcional entre la cantidad de máquinas con cuellos de botella y la eficiencia del sistema de producción. Es decir:

Mayor cantidad de máquinas con cuellos de botella: La eficiencia disminuye significativamente.

Menor cantidad de máquinas con cuellos de botella: La eficiencia aumenta, mejorando el rendimiento general del sistema.

Estos resultados confirman que los cuellos de botella tienen un impacto considerable en la eficiencia de producción. Por lo tanto, es crucial identificar y abordar estos puntos críticos para optimizar el flujo de trabajo y mejorar el desempeño global del sistema.

3.8.14 Matriz de Confusión.

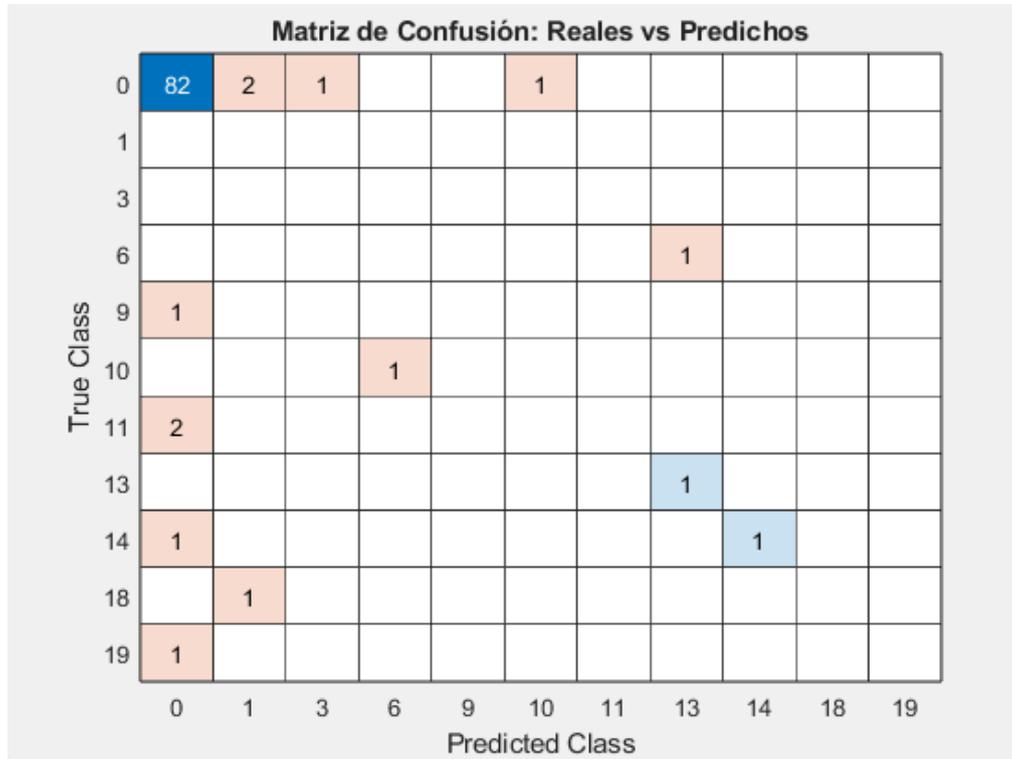


Figura 7. Matriz Confusión 100 máquinas

Al tratarse de un algoritmo que incorpora Machine Learning, existe la posibilidad de que cometa errores durante su ejecución. Esto se evidencia al analizar la matriz de confusión, donde se observan discrepancias entre los datos reales y los datos predichos por el modelo.

En este caso, el algoritmo alcanza una precisión del 80%, lo que indica que, aunque es capaz de realizar predicciones acertadas en la mayoría de los casos, aún presenta errores que deben ser considerados. Este margen de error destaca la importancia de seguir supervisando y ajustando el algoritmo, asegurando su mejora continua y optimización en futuros entrenamientos.

3.8.15 Árbol de decisiones

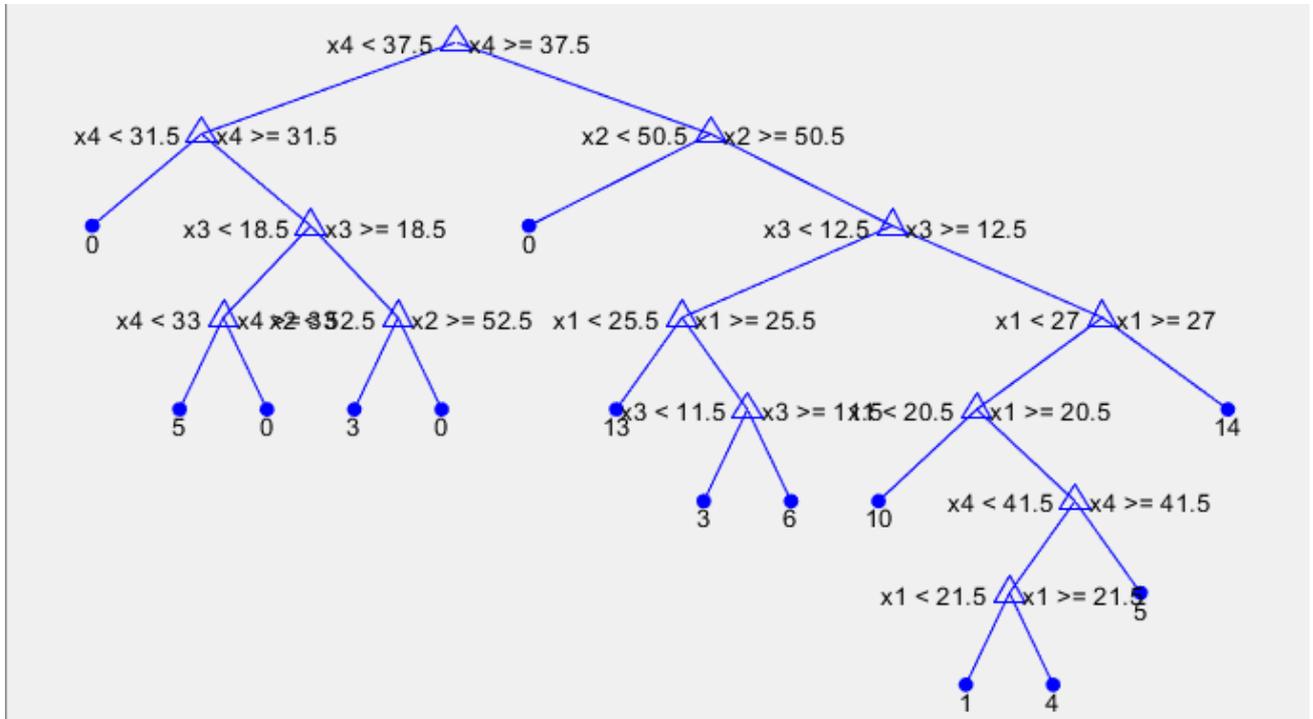


Figura 8. Árbol de Decisiones con 20 máquinas

La representación más comúnmente asociada con la Inteligencia Artificial es el árbol de decisiones, debido a su capacidad para estructurar y visualizar el proceso de toma de decisiones del algoritmo. Al analizar el árbol generado, se observa un aumento en la complejidad en comparación con la evaluación realizada con solo 3 máquinas.

El árbol de decisiones identifica las elecciones realizadas por el algoritmo para determinar el orden más eficiente, basándose en una serie de decisiones secuenciales y específicas. Este enfoque proporciona una representación clara de cómo el algoritmo procesa los datos y optimiza las operaciones, incluso en sistemas con mayor complejidad.

3.8.16 Distribución del Tiempo de Retraso

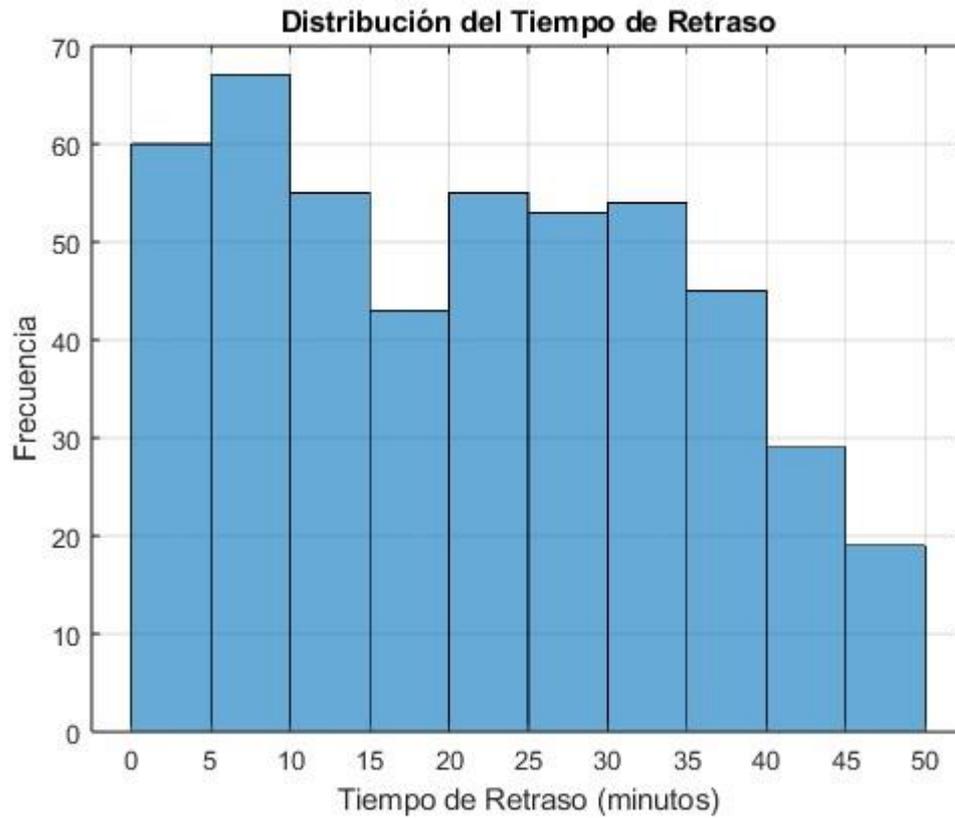


Figura 9. Distribucion del Tiempo de Retraso (fuente: Elaboración Propia)

La presencia de cuellos de botella genera retrasos en el sistema de producción. Según el análisis realizado, se observa que las máquinas experimentan retrasos de entre 5 y 10 minutos, repitiendo esta situación en menos de 70 ocasiones. Aunque la cantidad de datos registrada es alta, el impacto de estos retrasos puede considerarse relativamente bajo debido a su frecuencia limitada en el contexto del proceso global.

3.8.17 Comparación de Resultados Reales vs Predichos

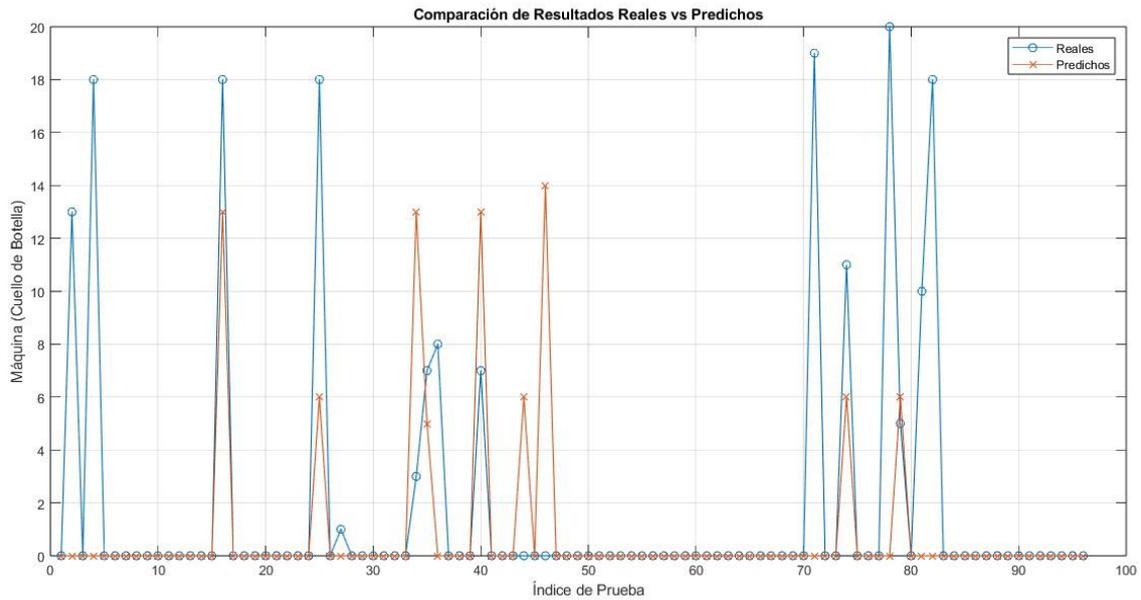


Figura 10. Comparación de Resultados Reales vs Predichos

Del mismo modo, al tratarse de un modelo basado en Machine Learning, se observa que la discrepancia entre los datos predichos y los datos reales no representa un problema significativo. Con una precisión superior al 80%, el algoritmo demuestra ser lo suficientemente confiable para su utilización en procesos de manufactura.

Este nivel de eficiencia permite que el algoritmo se integre como una herramienta útil para optimizar operaciones, aunque se recomienda su supervisión continua para minimizar posibles errores y mejorar su rendimiento en futuros ajustes.

3.9 Complejidad Computacional.

Para la evaluación de los algoritmos se lo realiza en una computadora portátil con las siguientes características:

Tabla 13. Características Equipo para evaluación

Procesador	AMD Ryzen 3 3250U with Radeon Graphics 2.60 GHz
RAM instalada	16,0 GB (13,9 GB utilizable)
Tipo de sistema	Sistema operativo de 64 bits, procesador x64
Memoria	250gb
Edición	Windows 11 Home
Versión	23H2
Compilación del SO	22631.4602

Esto convierte a la computadora en una opción accesible para empresas que buscan realizar procesos de optimización en sus líneas de producción, con un precio aproximado de \$500 en Ecuador. Esta característica la posiciona como una elección económica y viable para distintas organizaciones.

En la siguiente ilustración, se muestra el uso de recursos de la computadora en estado de reposo, sin la ejecución de la aplicación MATLAB.

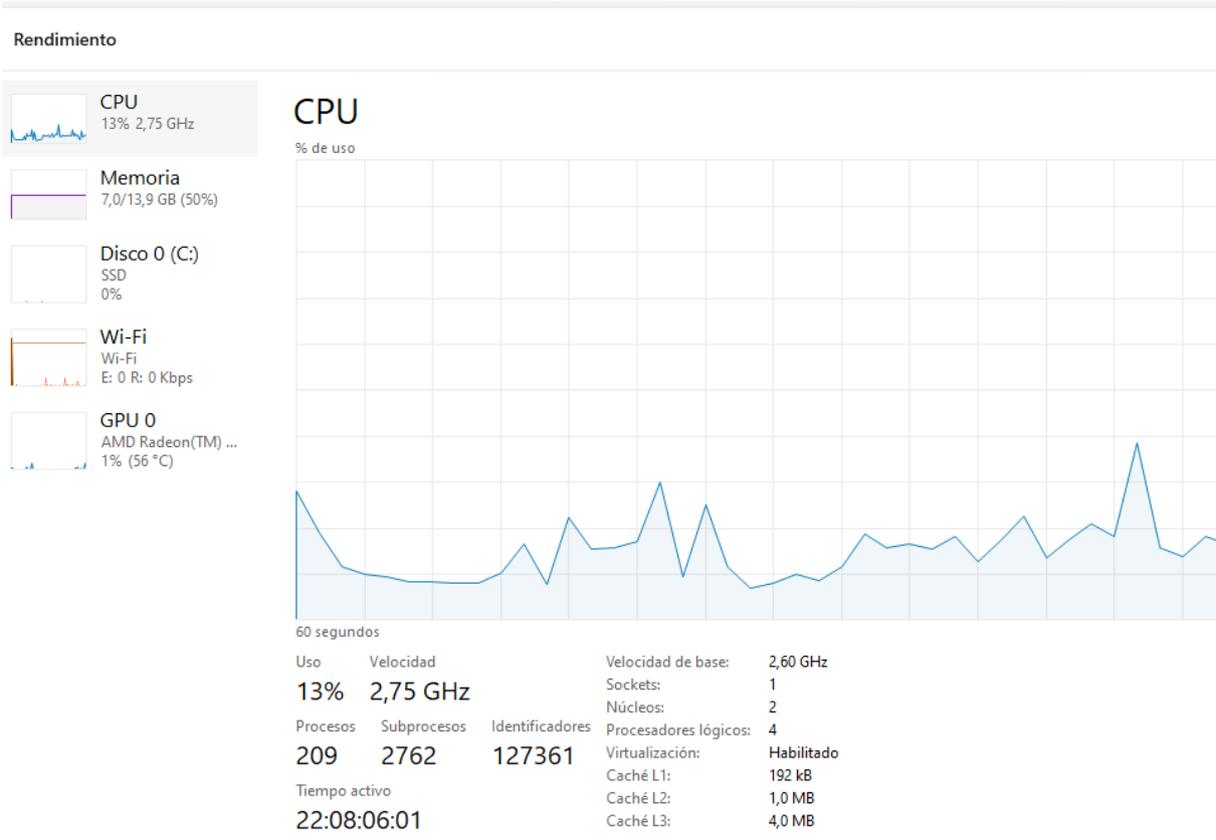


Figura 11. Uso de recursos previo a evaluación de Algoritmo

Se puede observar que la computadora no se encuentra sobrecargada con solo un uso del 13% de su CPU a continuación se mostrara el uso de recursos al realizar la evaluación del algoritmo cuello de botella móvil.

Rendimiento

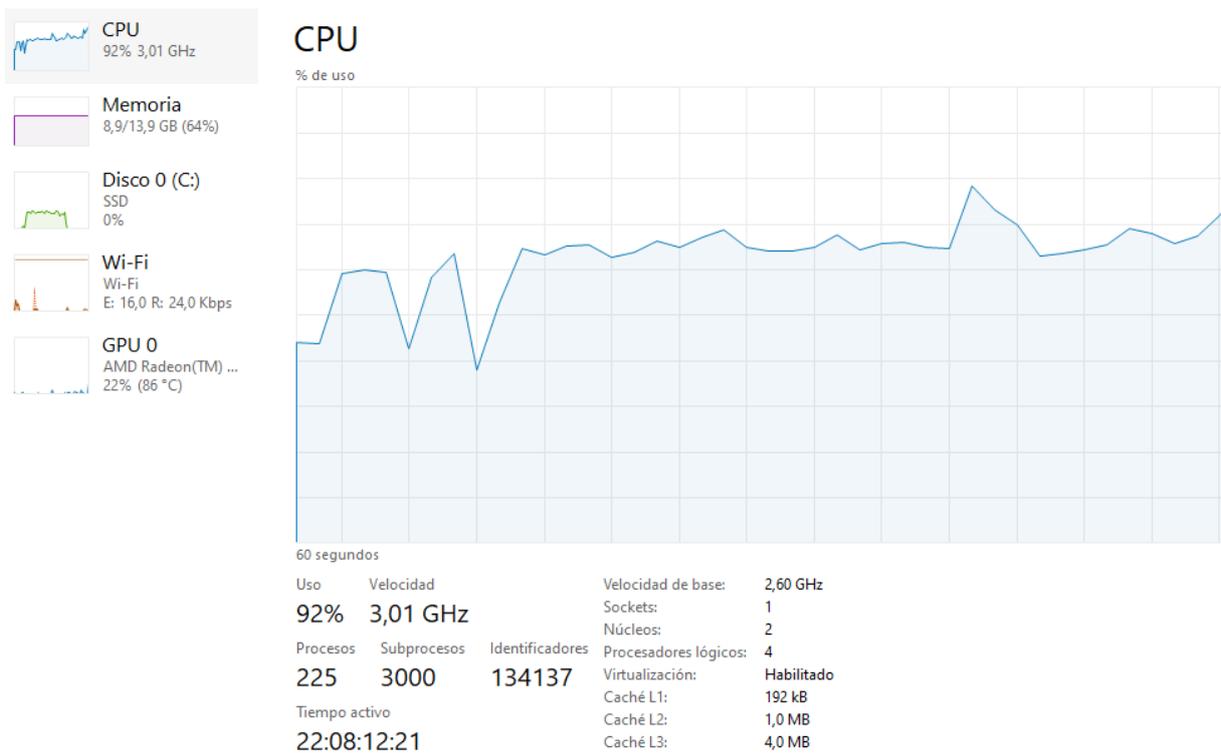


Figura 12. Uso de recursos durante evaluación de Algoritmo

En las simulaciones realizadas, se observa un máximo del 92% de ocupación del CPU durante la inicialización del programa. Posteriormente, el uso del CPU se estabiliza en un rango de 15% a 20%, indicando que el programa solo requiere una cantidad significativa de recursos al iniciar su funcionamiento.

Por lo tanto, no es necesario contar con una computadora de alto rendimiento para ejecutar este algoritmo de manera eficiente.

En cuanto al uso de la GPU, se determinó que es prácticamente inexistente. A pesar de contar con una GPU de entrada, esta no supera el 22% de uso, lo que confirma que el uso de una GPU o tarjeta de video dedicada es innecesario para este algoritmo.

3.9.1 Tiempo de ejecución

Para comparar ambos algoritmos se realiza una comparativa del tiempo de ejecución utilizando el código:

```
tic; % Inicia el temporizador
```

Esto al inicio del código para que el temporizador empiece a correr y

```
toc; % Muestra el tiempo transcurrido
```

Para finalizar el temporizador y comprobar el tiempo transcurrido.

Por lo tanto, los tiempos son:

Tabla 14 Comparación tiempos de ejecución (Fuente: Elaboración Propia)

Tiempos de ejecución	
Algoritmo Cuello de Botella Móvil	Algoritmo Cuello de Botella Móvil utilizando Inteligencia Artificial
0.889752 segundos	1.668541 segundos

Al comparar ambos tiempos de ejecución, se observa que el tiempo del Algoritmo de Cuello de Botella Móvil utilizando Inteligencia Artificial (IA) es significativamente mayor en comparación con el algoritmo de Cuello de Botella Móvil sin IA.

El principal factor que contribuye a esta diferencia es que el algoritmo con Machine Learning debe realizar evaluaciones y predicciones basadas en las bases de datos proporcionadas. Este proceso incluye el análisis de datos históricos y actuales, lo que incrementa el tiempo de cálculo para generar una respuesta. A pesar de esto, el uso de IA aporta una mayor precisión y flexibilidad en la optimización de sistemas complejos.

Al correr el código con una cantidad elevada de tareas, se recopilaron los siguientes datos, los cuales están representados en el gráfico a continuación.

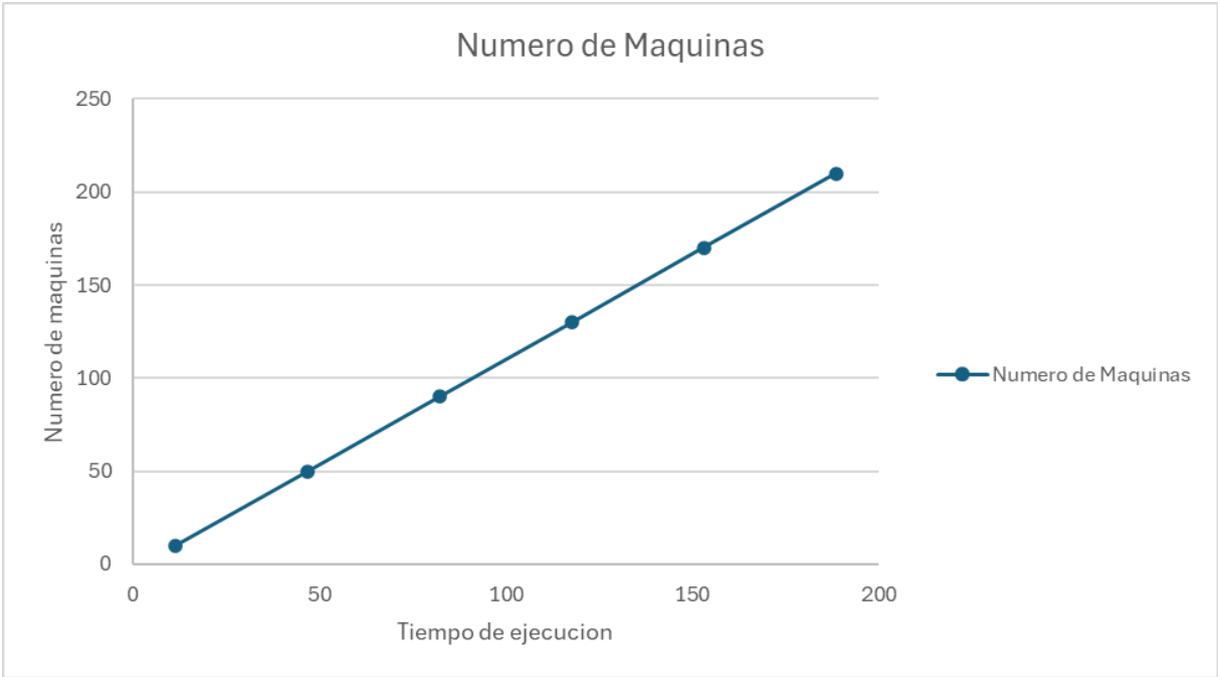


Figura 13. Número de máquinas vs tiempo de ejecución

Se puede ver que el tiempo de ejecución aumenta a medida que aumenta el número de máquinas, pero en este caso al ser una cantidad extremadamente alta hace que para muchas empresas sea complicado llegar a ese número por lo que dentro del modelo de estudio el algoritmo sigue siendo fiable.

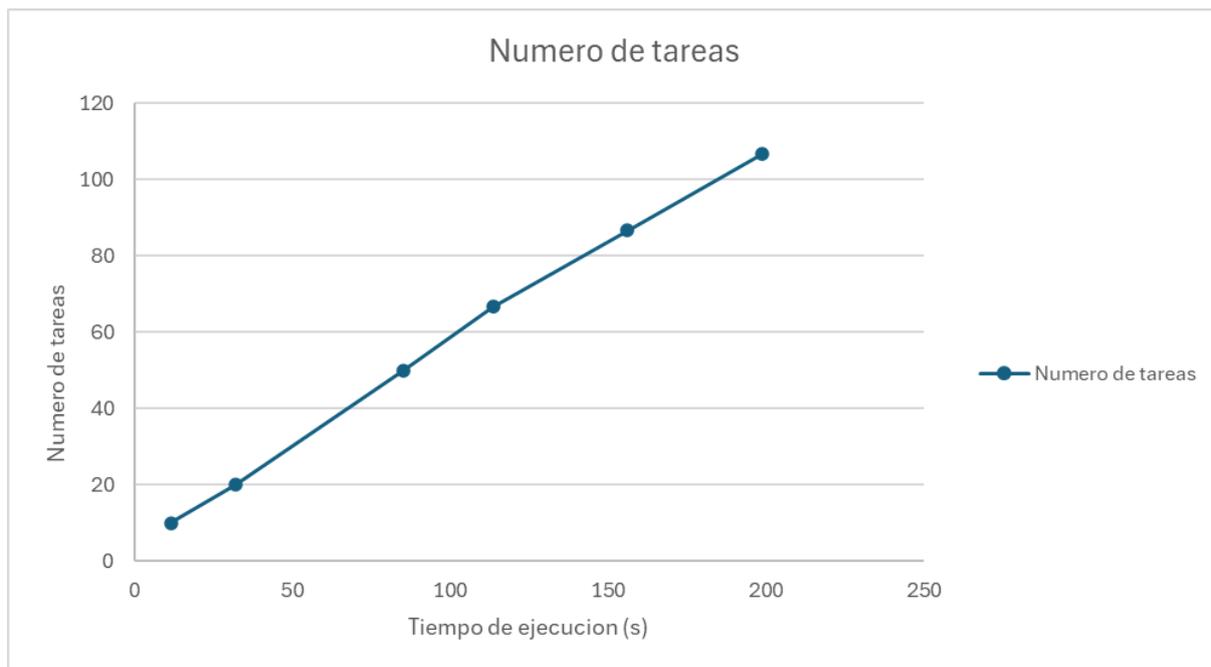


Figura 14. Número de tareas vs tiempo de ejecución

El análisis muestra que el tiempo de ejecución del algoritmo presenta un crecimiento lineal, lo que evidencia que, al incrementar el número de máquinas y tareas, el tiempo de ejecución no se eleva exponencialmente como se anticipa. Asimismo, el incremento en el número de máquinas impacta en el tiempo de ejecución de forma más moderada en comparación con el aumento en el número de tareas. Esto ocurre porque el algoritmo debe gestionar un volumen mayor de información al asignar las tareas entre las máquinas, lo que aumenta su nivel de complejidad.

Durante las simulaciones, el uso del CPU alcanza un máximo del 77%, lo que demuestra que el algoritmo es relativamente simple y puede ejecutarse de manera eficiente en computadoras estándar. Esta característica lo hace adaptable y viable para su implementación en diferentes entornos computacionales.

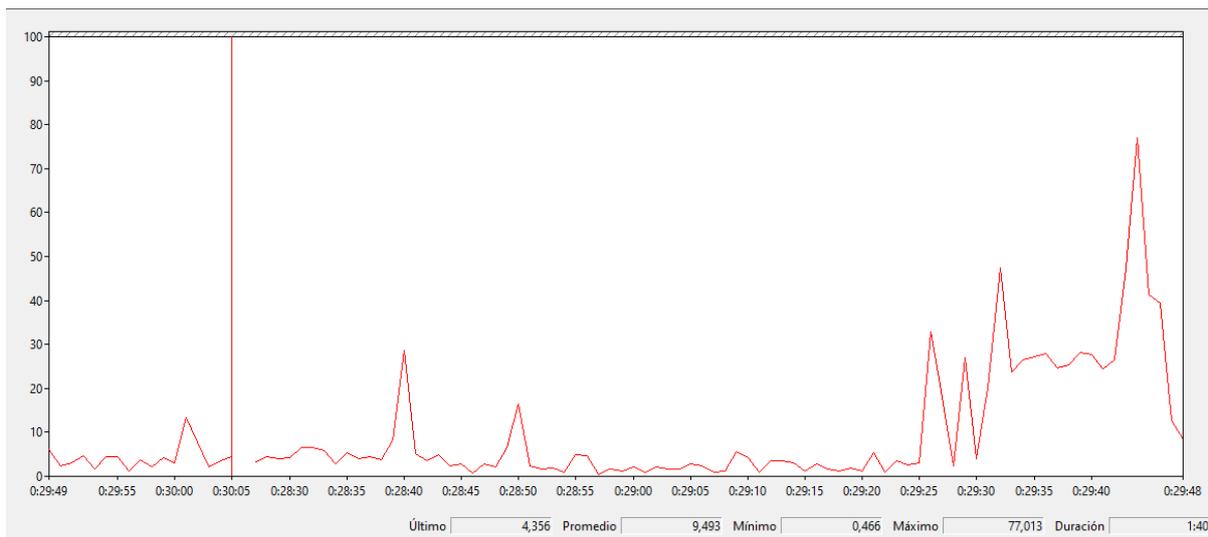


Figura 15. Pico máximo de consumo utilizando Machine Learning

Se observa que el máximo de consumo mientras corre el algoritmo utilizando Machine Learning es de 77.03%, dando como conclusión que la computadora no se ve sobre esforzada por el código.

3.10 Eficiencia

Luego de haber realizado el tiempo de ejecución se realiza un cálculo de la memoria utilizada por ambos algoritmos los resultados se observan en la tabla 15.

Tabla 15 Calculo memoria utilizada (Fuente: Elaboración Propia)

Memoria Utilizada	
Algoritmo Cuello de Botella Móvil	Algoritmo Cuello de Botella Móvil utilizando Inteligencia Artificial
4220 MB	2120 MB

De igual manera el algoritmo cuello de botella Móvil utilizando inteligencia artificial ocupa menor cantidad de memoria para su ejecución así se aumenta un proceso adicional como el entrenamiento del algoritmo.

3.11 Eficacia

Al momento de comprobar ambos algoritmos, estos dieron los mismos resultados por lo que su eficacia es la misma.

En el estado normal de la planta la producción producía 22 pallets de producto y con la optimización ahora produce 27 pallets por lo que ambos algoritmos son eficaces al momento de localizar cuellos de botella.

Para calcular que tan eficaz es la optimización se utiliza la siguiente formula:

$$\text{Eficacia} = \frac{\text{Producción después de optimización} - \text{Producción inicial}}{\text{Producción inicial}} \times 100\%$$

Dando como resultado que ambas maquinas son eficaces en un 22.73%, lo que indica que la producción es superior en ese porcentaje.

3.12 Ventajas de utilizar Machine Learning en el Algoritmo Cuello de Botella Móvil

3.12.1 Adaptación al entorno de la empresa

El algoritmo se adapta al entorno específico de la empresa al utilizar datos históricos como referencia para su posterior secuenciación. Esto le permite ajustarse a las particularidades de cada sistema y mejorar la precisión en la planificación.

3.12.2 Simplificación en la toma de información

Los procesos de producción suelen involucrar actividades secundarias relacionadas con las máquinas, las cuales a menudo pasan desapercibidas o se subestiman, generando un uso

ineficiente de recursos. Este algoritmo ayuda a considerar de manera intrínseca todas estas variables, lo que garantiza una visión más completa del sistema.

3.12.3 Facilidad de interpretación

A pesar de su mayor capacidad para manejar datos complejos, el algoritmo conserva la simplicidad conceptual del algoritmo de cuello de botella móvil. La principal diferencia radica en su capacidad para procesar y analizar la información de manera más detallada, manteniendo su fácil interpretación y aplicación.

3.12.4 Desventajas de utilizar Machine Learning en el Algoritmo Cuello de Botella Móvil

Dependencia de datos históricos

El algoritmo depende en gran medida de los datos históricos para su funcionamiento. Esto puede complicar su implementación, especialmente en la fase de evaluación, ya que requiere una base de datos bien estructurada y confiable para generar resultados precisos.

Susceptibilidad a la manipulación de datos

El algoritmo es vulnerable a la manipulación o errores en los datos de entrada. Si los datos en la base son modificados, ya sea de manera intencional o accidental, el algoritmo es propenso a generar resultados erróneos, comprometiendo la toma de decisiones basadas en sus predicciones.

Por lo tanto, la implementación de Machine Learning dentro del algoritmo de Cuello de Botella Móvil ofrece más ventajas que desventajas. Su capacidad para adaptarse y optimizar procesos lo convierte en una herramienta poderosa para mejorar la eficiencia en sistemas de producción.

Para prevenir las desventajas asociadas, es crucial implementar una recolección de datos consciente, asegurando su precisión y calidad. Además, es importante minimizar los sesgos en los datos, ya que estos pueden comprometer la eficacia del algoritmo. Con estas medidas, se garantiza un rendimiento óptimo y confiable del algoritmo en diferentes entornos.

3.13 Conclusiones

Luego de realizar las respectivas evaluaciones, concluimos lo siguiente:

1. Se concluye que, en el caso de una planta procesadora pequeña, con pocas máquinas, no se observa una variación significativa en el rendimiento. Los resultados obtenidos con los algoritmos son similares a los logrados mediante la experticia de los trabajadores. Sin embargo, esto no implica que la implementación de Machine Learning en el algoritmo sea innecesaria. Al aumentar el número de máquinas, el algoritmo enriquecido con Machine Learning ofrece valores más precisos y cercanos a la realidad. Además, su enfoque no se limita únicamente a los números de optimización, sino que incorpora más variables que aportan un análisis más completo del sistema. Por lo tanto, su uso, junto con una correcta recolección de datos, convierte al algoritmo en una herramienta óptima para su implementación en procesos de manufactura, especialmente en entornos más complejos y dinámicos.
2. En entornos con baja complejidad, la experiencia de los trabajadores demuestra ser altamente eficiente, alcanzando resultados cercanos a los obtenidos mediante las simulaciones de ambos algoritmos. Esto sugiere que, en sistemas simples, la intervención humana puede ser comparable en eficiencia a los métodos automatizados.
3. El uso de machine Learning puede ser una herramienta valiosa, especialmente cuando se dispone de una cantidad suficiente de datos de calidad que permite aproximarse al nivel de experticia de los trabajadores. Además, en sistemas con un mayor número de máquinas, el machine Learning puede facilitar la secuenciación y asignación óptima de tareas, mejorando la eficiencia global del sistema.
4. El algoritmo de Cuello de Botella Móvil es escalable, ya que puede adaptarse a un número arbitrario de máquinas siempre que la calidad de los datos de entrada sea óptima.

Por su parte, la integración de Machine Learning amplía las posibilidades del algoritmo al mejorar su capacidad de adaptación en sistemas más complejos.

5. Las simulaciones realizadas en MATLAB y Simulink demostraron ser fluidas incluso cuando se incrementa significativamente la cantidad de datos y tareas. Esto hace que estas herramientas sean adecuadas para su implementación en pequeñas industrias, donde los recursos computacionales pueden ser limitados.

3.14 Recomendaciones

La precisión y consistencia de los datos representan un elemento clave para garantizar el éxito del algoritmo. Es esencial que la recopilación de datos se realice con el mayor grado de exactitud posible, ya que estos constituyen la base para entrenar el modelo y sirven como guía en el proceso de toma de decisiones. Una recopilación inadecuada puede impactar de manera negativa en el rendimiento del algoritmo.

Se recomienda adaptar el algoritmo a sistemas de detección de datos en tiempo real para mejorar su eficacia. Estos sistemas pueden incluir sensores o cámaras que permitan recopilar información de manera continua y precisa, facilitando su implementación en procesos dinámicos.

Aunque el algoritmo puede aprender y mejorar, su uso debe ser supervisado constantemente. Si se detectan errores en su funcionamiento, es importante corregirlos e indicarlos al algoritmo para que ajuste su modelo y continúe aprendiendo de manera adecuada. Este enfoque asegura una mejora continua en su rendimiento.

- [1] MathWorks, “Coca-Cola desarrolla un sensor de presión virtual con aprendizaje automático para mejorar el diagnóstico de los dispensadores de bebidas,” MathWorks. [Online]. Available: https://la.mathworks.com/company/user_stories/coca-cola-develops-virtual-pressure-sensor-with-machine-learning-to-improve-beverage-dispenser-diagnostics.html
- [2] “Mercedes-Benz simula sensores de hardware con redes neuronales profundas,” MathWorks. [Online]. Available: https://la.mathworks.com/company/user_stories/mercedes-benz-simulates-hardware-sensors-with-deep-neural-networks.html
- [3] R. Benitez Iglesias, *Inteligencia artificial avanzada*. Editorial UOC, 2014. [Online]. Available: <https://elibro.net/es/lc/bibliotecaups/titulos/57582>
- [4] D. R. Vora and G. S. Bhatia, *Python Machine Learning Projects*. 2023. [Online]. Available: <https://bibliotecas.ups.edu.ec:2708/lib/upsal/reader.action?docID=30495007&ppg=1>
- [5] M. Martinez and D. Maestri, *Aceites vegetales no tradicionales: guia para la produccion y evaluacion de calidad*. Editorial Brujas, 2016. [Online]. Available: <https://elibro.net/es/lc/bibliotecaups/titulos/77036>
- [6] L. A. Mora Garcia, *Industria y logistica 4.0*. RA-MA Editorial, 2023. [Online]. Available: <https://elibro.net/es/lc/bibliotecaups/titulos/248867>
- [7] M. A. Gray, *Introduction to the Simulation of Dynamics Using Simulink*. Taylor & Francis Group, 2010. [Online]. Available: <https://elibro.net/es/lc/bibliotecaups/titulos/145271>
- [8] R. A. Britto Agudelo, “Programacion de la produccion en sistemas de manufactura tipo taller con el algoritmo combinado cuello de botella movil y busqueda tabu,” 2009, *Red Ingenieria y Universidad*. [Online]. Available: <https://elibro.net/es/lc/bibliotecaups/titulos/98805>
- [9] A. Jimenez Avello, M. Castro Gil, and J. M. Costa Garcia, *Simulacion de procesos y*

aplicaciones. Dextra Editorial, 2015. [Online]. Available:
<https://elibro.net/es/lc/bibliotecaups/titulos/133345>

- [10] C. A. Romano, *Optimizacion metaheuristica para ingenieros: aplicaciones en organizacion industrial*. Delta Publicaciones, 2018. [Online]. Available:
<https://elibro.net/es/lc/bibliotecaups/titulos/227317>