



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL**

**FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO  
CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y  
AUTOMATISMO**

**TEMA:**

Desarrollo de un sistema SCADA para la supervisión de un proceso industrial  
utilizando inteligencia artificial

**AUTOR:**

Sabando García, Klever Alejandro

Trabajo de titulación previo a la obtención del título de  
**INGENIERO ELECTRÓNICO EN CONTROL Y AUTOMATISMO**

**TUTOR:**

Ing. Mendoza Merchán, Eduardo Vicente M.gs.

**Guayaquil, Ecuador**

**28 de febrero del 2020**



UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO  
CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y  
AUTOMATISMO

### **CERTIFICACIÓN**

Certificamos que el presente trabajo de titulación fue realizado en su totalidad por **Sabando García, Klever Alejandro** como requerimiento para la obtención del título de **INGENIERO ELECTRÓNICO EN CONTROL Y AUTOMATISMO**.

TUTOR

---

Ing. Mendoza Merchán, Eduardo Vicente M.gs.

DIRECTOR DE LA CARRERA

---

Ing. Heras Sánchez, Miguel Armando M.gs.

Guayaquil, 28 de febrero del 2020



UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO  
CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y  
AUTOMATISMO

### **DECLARACIÓN DE RESPONSABILIDAD**

**Yo, Sabando García, Klever Alejandro**

#### **DECLARO QUE:**

El Trabajo de Titulación: **“DESARROLLO DE UN SISTEMA SCADA PARA LA SUPERVISIÓN DE UN PROCESO INDUSTRIAL UTILIZANDO INTELIGENCIA ARTIFICIAL”**, previo a la obtención del título de **Ingeniero Electrónico en Control y Automatismo**, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

Guayaquil, 28 de febrero del 2020

#### **EL AUTOR**

---

Sabando García, Klever Alejandro



UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO  
CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y  
AUTOMATISMO

### **AUTORIZACIÓN**

**Yo, Sabando García, Klever Alejandro**

Autorizo a la Universidad Católica de Santiago de Guayaquil a la **publicación** en la biblioteca de la institución del Trabajo de Titulación: **“DESARROLLO DE UN SISTEMA SCADA PARA LA SUPERVISIÓN DE UN PROCESO INDUSTRIAL UTILIZANDO INTELIGENCIA ARTIFICIAL”**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

**Guayaquil, 28 de febrero del 2020**

**EL AUTOR:**

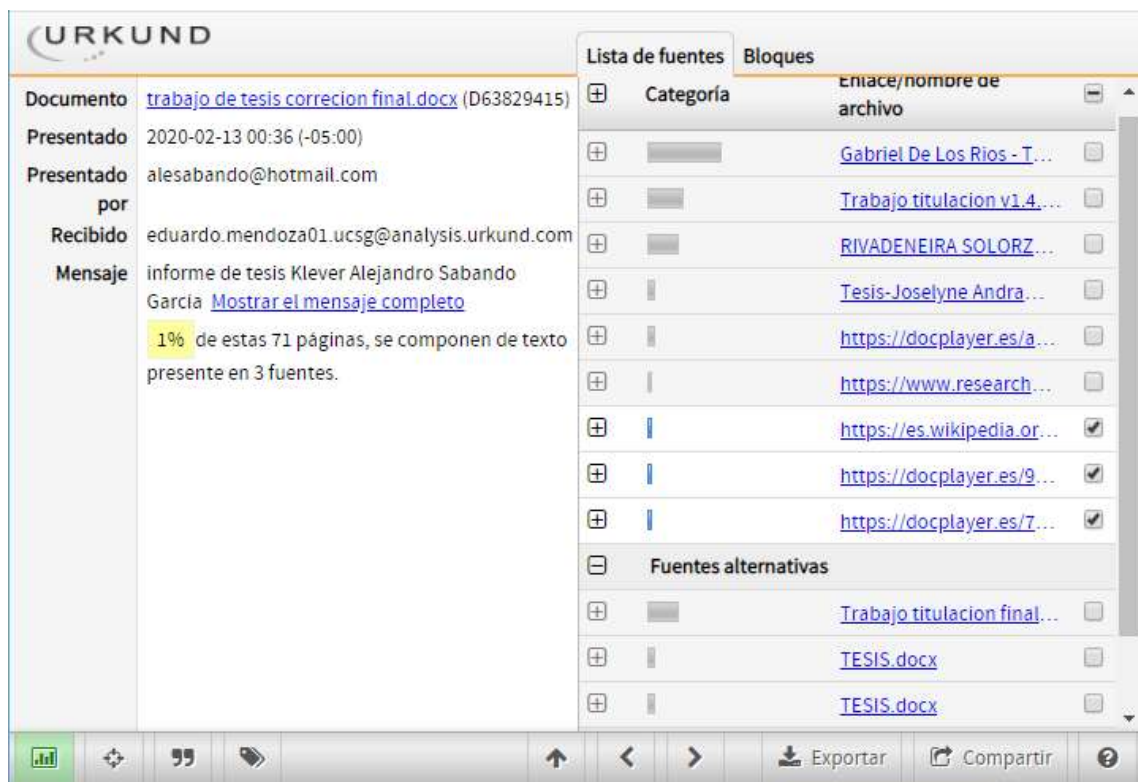
---

Sabando García, Klever Alejandro

## REVISIÓN DE TRABAJO DE TITULACIÓN CON PROGRAMA URKUND CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y AUTOMATISMO

**TÍTULO:** “Desarrollo de un sistema SCADA para la supervisión de un proceso industrial utilizando inteligencia artificial”.

**AUTOR:** SABANDO GARCÍA KLEVER ALEJANDRO



The screenshot displays the URKUND interface. On the left, a message pane shows details for a document titled 'trabajo de tesis correccion final.docx (D63829415)', presented on 2020-02-13 00:36 (-05:00) by alesabando@hotmail.com to eduardo.mendoza01.ucsg@analysis.orkund.com. The message content includes 'informe de tesis Klever Alejandro Sabando Garcia' and a link to 'Mostrar el mensaje completo'. A yellow highlight indicates that 1% of the 71 pages consist of text from 3 sources.

On the right, the 'Lista de fuentes' (List of sources) pane is active, showing a table with columns for 'Categoría' and 'Enlace/nombre de archivo'. The sources listed include:

Categoría	Enlace/nombre de archivo
	Gabriel De Los Rios - T...
	Trabajo titulacion v1.4...
	RIVADENEIRA SOLORZ...
	Tesis-Joselyne Andra...
	<a href="https://docplayer.es/a...">https://docplayer.es/a...</a>
	<a href="https://www.research...">https://www.research...</a>
	<a href="https://es.wikipedia.or...">https://es.wikipedia.or...</a>
	<a href="https://docplayer.es/9...">https://docplayer.es/9...</a>
	<a href="https://docplayer.es/7...">https://docplayer.es/7...</a>
Fuentes alternativas	
	Trabajo titulacion final...
	TESIS.docx
	TESIS.docx

The bottom toolbar contains icons for document analysis, navigation, and actions like 'Exportar' and 'Compartir'.

Después de analizar el resultado enviado por el programa Urkund, se determinó que el trabajo de titulación del estudiante: **SABANDO GARCÍA KLEVER ALEJANDRO**, observa un porcentaje inferior al 4% de coincidencias con otros documentos encontrados en el internet.

.....  
**Ing. Eduardo Mendoza Merchán, Mgs**  
**DOCENTE TUTOR**

## AGRADECIMIENTO

Agradezco de forma especial a mis padres, que con el fruto diario del esfuerzo y amor a su vocación me han enseñado el verdadero significado del trabajo. A mi padre Klever Bienvenido Sabando García, de quien he aprendido la importancia de la disciplina en el trabajo y el cumplimiento satisfactorio del mismo. A mi madre Liria Marixy García Moreira, quién me ha inculcado a esforzarme cada día más en mis labores propias con el objetivo de alcanzar las metas más grandes que han sido propuestas; al amor que se demuestra en el constante compartir, no solamente de premios y regalos, sino a dar y vivir experiencias junto a los seres más queridos, la familia.

A los docentes de la Facultad Técnica Para el Desarrollo, quienes nos han guiado por el camino de la sabiduría, y que, nos han demostrado una y otra vez, que, en la práctica, el conocimiento adquirido en las aulas de clases no suele ser suficiente para poder solucionar los problemas que surgen, y que, al final, dicho conocimiento emerge de la experiencia laboral. Al Ing. Celso Bayardo Bohórquez Escobar, quien me ha brindado su mano de apoyo en los momentos más críticos de la carrera profesional, enseñándome la importancia de mirar hacia adelante en los momentos más difíciles, asimismo, aprendiendo de los errores cometidos en el transcurso de la vida estudiantil y profesional.

Al Ing. Manuel Romero Paz decano de la Facultad de Educación Técnica para el Desarrollo y al Ing. Armando Heras Sánchez, director de las carreras de Control y Automatismo, Telecomunicaciones, y Electricidad de esta prestigiosa facultad, quienes, con empeño y dedicación a través de sus años de trabajo, han dirigido con excelencia a los futuros profesionales que realizarán su labor y aportes a la sociedad.



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL**  
FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO  
CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y  
AUTOMATISMO

**TRIBUNAL DE SUSTENTACIÓN**

f. \_\_\_\_\_

Ing. Heras Sánchez, Miguel Armando M.gs.

**DIRECTOR DE CARRERA**

f. \_\_\_\_\_

Ing. Córdova Rivadeneira, Luis Silvio M.gs.

**COORDINADOR DEL ÁREA DE ELECTRÓNICA**

f. \_\_\_\_\_

Ing. Luis Orlando Philco Asqui M.gs.

**OPONENTE**

## ÍNDICE

<b>INTRODUCCIÓN</b> .....	2
<b>CAPÍTULO 1</b> .....	6
1.1. Planteamiento del problema.....	6
1.2. Justificación del problema.....	7
1.3. Objetivos.....	8
1.3.1. Objetivo general.....	8
1.3.2. Objetivos específicos.....	8
1.4. Metodología de la investigación.....	9
1.5. Métodos de investigación.....	9
1.6. Tareas de investigación.....	10
<b>CAPÍTULO 2: FUNDAMENTOS TEÓRICOS</b> .....	11
2.1. Estado del arte.....	11
2.2. Inteligencia artificial.....	17
2.2.1. Aplicaciones de los sistemas de inteligencia artificial en la industria.....	18
2.2.2. Desafíos para la I.A. en la industria.....	19
2.2.3. Lógica difusa.....	21
2.3. Sistemas SCADA.....	27
2.3.1. Arquitectura de un sistema SCADA.....	28
2.3.2. CODESYS.....	30
2.4. Línea de ensamblaje industrial.....	35
<b>CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA SCADA INTELIGENTE</b> .....	37
3.1. Consideraciones iniciales.....	37
3.1.1. Descripción del sistema SCADA inteligente.....	37
3.1.2. Caracterización del sistema SCADA inteligente.....	39
3.1.3. Funciones del sistema SCADA inteligente.....	40
3.2. Análisis del sistema SCADA inteligente.....	41
3.2.1. Arquitectura del sistema SCADA.....	42
3.2.2. Módulos del sistema.....	48
3.2.3. Visualización general de la planta y mediciones del proceso.....	60
3.2.4. Análisis de los procedimientos desarrollados previos a la simulación de la línea de ensamblaje industrial.....	63
3.3. Software.....	67
3.3.1. Diseño del algoritmo inteligente a través de la técnica de lógica difusa utilizando MATLAB.....	70
<b>CAPÍTULO 4: PRESENTACIÓN, ANÁLISIS DE RESULTADOS Y DISCUSIÓN</b> .....	83
3.4. Pruebas de prototipo.....	83
3.4.1. Pruebas de funcionamiento del sistema de I.A. aplicada a la simulación del sistema SCADA para la red industrial a través del algoritmo de lógica difusa.....	85
4.1. Presentación de resultados.....	90
4.1.1. Resultados del software.....	90



<b>4.1.2. Resultados de pruebas de prototipo .....</b>	<b>92</b>
<b>4.2. Discusión de resultados .....</b>	<b>93</b>
<b>4.2.1. Cumplimiento de objetivos .....</b>	<b>93</b>
<b>4.2.2. Análisis de la metodología .....</b>	<b>94</b>
<b>4.2.3. Análisis de la solución al problema de investigación .....</b>	<b>94</b>
<b>4.2.4. Discusión final.....</b>	<b>95</b>
<b>CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES .....</b>	<b>96</b>
<b>5.1. Conclusiones .....</b>	<b>96</b>
<b>5.2. Recomendaciones .....</b>	<b>96</b>
<b>5.3. Trabajo futuro .....</b>	<b>98</b>
<b>Referencias Bibliográficas .....</b>	<b>99</b>

## ÍNDICE DE FIGURAS

### CAPÍTULO 2

Fig. 2.1. Predix. – Robots colaborativos para guías manuales en la automatización.	18
Fig. 2.2. Temperatura analizada en lógica difusa.	21
Fig. 2.3. Diseño general de un sistema SCADA	27
Fig. 2.4. Niveles funcionales de una operación de control para la fabricación.	28
Fig. 2.5. Computación en la nube.	29
Fig. 2.6. Administrador de componentes y funciones disponibles para el software de CODESYS	31

### CAPÍTULO 3

Fig. 3.1. Diagrama de flujo del sistema SCADA empleado para la simulación	37
Fig. 3.2. Arquitectura del sistema SCADA	42
Fig. 3.3. Arquitectura de red del sistema SCADA	45
Fig. 3.4. Arquitectura del módulo de configuración	49
Fig. 3.5. Diagrama de flujo que indica el control del proceso	50
Fig. 3.6. Almacenamiento ordenado de las variables disponibles en CODESYS	52
Fig. 3.7. Diagrama de flujo de alarma de piezas	53
Fig. 3.8. Cuadro de alarmas configuradas para la simulación del proceso	54
Fig. 3.9. Vinculación de las variables de alarma de la simulación a los elementos de supervisión.	55
Fig. 3.10. Visualización de la activación de las alarmas en la simulación con sus palancas de respuesta.	55
Fig. 3.11. Interfaz gráfica de la visualización en CODESYS	57
Fig. 3.12. Lista de dispositivos a los que se le pueden transmitir la información del proceso.	58
Fig. 3.13. Diagrama de flujo que representa la lista de operaciones de decisión ejecutadas en la planta	61
Fig. 3.14. Ventana de visualización en el programa de CODESYS para el entorno de ensamblaje vehicular	62
Fig. 3.15. Coche deportivo Charger R/T de 1970	63
Fig. 3.16. Carrocería y chasis del Charger R/T de 1970	64
Fig. 3.17. Separación gráfica de las piezas de la carrocería y traslado al entorno de CODESYS	65
Fig. 3.18. Diagrama de casos de uso para el software del sistema	67
Fig. 3.19. Diagrama de secuencias para el software del sistema	68
Fig. 3.20. Diagrama de estados para el software del sistema	69
Fig. 3.21. Diagrama de flujo del algoritmo de detección de colisiones “HITBOX”	73
Fig. 3.22. Diagrama de flujo del algoritmo de visión artificial.	74
Fig. 3.23. Diagrama de flujo de la implementación del algoritmo de I.A. en CODESYS	76
Fig. 3.24. Diagrama de flujo de la interfaz de I.A. por lógica difusa.	77
Fig. 3.25. Vista general del agente de lógica difusa para el posicionamiento	78

Fig. 3.26. Función de pertenencia para los rangos de posición de la pieza en el eje x .....	79
Fig. 3.27. Función de pertenencia para los rangos de posición de la pieza en el eje y .....	79
Fig. 3.28. Función de pertenencia de los rangos de posición del brazo robótico en el eje x .....	80
Fig. 3.29. Función de pertenencia de los rangos de posición del brazo robótico en el eje y .....	81
Fig. 3.30. Función de pertenencia que indica la probabilidad de ensamblaje para la salida del proceso de lógica difusa .....	82
Fig. 3.31. Programa de visualización para el entorno de ensamblaje vehicular en CODESYS .....	85
Fig. 3.32. Simulación en MATLAB de probabilidad baja de ensamblaje .....	87
Fig. 3.33. Diagrama de barras que indica la probabilidad baja de ensamblaje .....	88
Fig. 3.34. Simulación en MATLAB de probabilidad media de ensamblaje .....	89
Fig. 3.35. Diagrama de barras que indica la probabilidad media de ensamblaje .....	89
Fig. 3.36. Simulación en MATLAB de probabilidad alta de ensamblaje .....	90
Fig. 3.37. Diagrama de barras que indica la probabilidad alta de ensamblaje .....	90
Fig. 3.38. Simulación en MATLAB de la condición de ensamblaje .....	91
Fig. 3.39. Diagrama de barras que indica la condición de ensamblaje .....	91

#### **CAPÍTULO 4**

Fig. 4.1. Simulación en MATLAB en la que se obtiene la superficie generada por las reglas de inferencia del algoritmo de lógica difusa .....	94
Fig. 4.2. Gráfica que muestra la salida del proceso de cada prueba de prototipo en relación con su función de pertenencia .....	96

## ÍNDICE DE TABLAS

### CAPÍTULO 2

Tabla 2.1. Operadores de lógica difusa .....	21
Tabla 2.2. Estructura de acceso a la plataforma de CODESYS con estructuras de Python.....	33

### CAPÍTULO 3

Tabla 3.1. Variables asignadas a las piezas del proceso .....	70
Tabla 3.2. Variables de posición y tamaño de las piezas (visualización) .....	70
Tabla 3.3. Variables del sistema y tamaño de cada una .....	70
Tabla 3.4. Reglas de inferencia de las posibles posiciones para el ensamblaje .....	82
Tabla 3.5. La posición de las piezas .....	86

## **RESUMEN**

En el presente trabajo de titulación se ha realizado el diseño de un sistema SCADA para la supervisión de un proceso industrial aplicando inteligencia artificial a través de algoritmos de lógica difusa, para ello se debe diseñar y simular el entorno industrial a través del programa CODESYS, el cual permite programar dicho proceso, y a la vez utilizando el programa de I.A. para realizar la supervisión inteligente de la simulación. El estudio realizado es de tipo correlacional, lo cual implica la relación entre las variables de posicionamiento en un proceso de ensamblaje vehicular. Este tipo de procesos necesitan manejar estrictos estándares de montaje para poder cumplir con las expectativas de la actual industria automovilística, la cual requiere de una amplia capacidad de procesamiento para realizar la administración y toma de decisiones con respecto a esta clase de aplicaciones. Los resultados de las pruebas de simulación arrojaron los siguientes resultados: Las mediciones de la posición de las piezas a trasladar se realizaron correctamente, la supervisión y control inteligente de los elementos del proceso, cintas transportadoras, brazos robóticos, piezas, etc. El algoritmo basado en lógica difusa, para la supervisión de la simulación del proceso industrial de tipo línea de ensamblaje vehicular, logró optimizar el control, la supervisión y adquisición de datos del proceso analizado.

**PALABRAS CLAVE:** SCADA INTELIGENTE, INTERFAZ DE COMUNICACIÓN, LÓGICA DIFUSA, CONTROL INDUSTRIAL, FUNCIÓN DE PERTENENCIA, DEFUZIFICACIÓN, CODESYS, MAMDANI.

## **ABSTRACT**

In the present degree work, the design of a SCADA system for the supervision of an industrial process has been carried out applying artificial intelligence through algorithms of fuzzy logic, for this the industrial environment must be designed and simulated through the CODESYS program, the which allows to program this process, and at the same time using the AI program to perform intelligent simulation monitoring. The study carried out is of a correlational type, which implies the relationship between the positioning variables in a vehicular assembly process. These types of processes need to handle strict assembly standards in order to meet the expectations of the current automobile industry, which requires ample processing capacity to perform the administration and decision making regarding this kind of applications. The results of the simulation tests yielded the following results: The measurements of the position of the pieces to be transferred were carried out correctly, the intelligent supervision and control of the process elements, conveyor belts, robotic arms, parts, etc. The algorithm based on fuzzy logic, for the supervision of the simulation of the industrial process of the vehicle assembly line type, managed to optimize the control, supervision and data acquisition of the analyzed process.

**KEY WORDS:** INTELLIGENT SCADA, COMMUNICATION INTERFACE, DIFFUSE LOGIC, INDUSTRIAL CONTROL, MEMBERSHIP FUNCTION, DEFUZZIFICATION, CODESYS, MAMDAMI

## INTRODUCCIÓN

Los procesos de automatización industrial cada vez se han vuelto cada vez más complejos y permiten realizar controles de procesos industriales cada vez más precisos. Ghosh (2018) plantea que la definición de automatización es cada vez más generalizada y al mismo tiempo económica, debido a los desarrollos en términos de accesibilidad para la pequeña y mediana empresa; con la entrada de la inteligencia artificial, se introduce el término de fábricas autónomas, el cual hace referencia a la poca o nula intervención humana dentro de los procesos realizados en planta, con un método establecido para el desarrollo de estas tecnologías inteligentes la cual es la lógica difusa, esta ha permitido desarrollar técnicas de control a través de reglas de inferencia que el ser humano utiliza constantemente en la vida cotidiana, gracias a esto, se pueden hacer sistema relativamente sencillos de realizar y a bajo costo. Sin embargo estos algoritmos son ineficaces a la hora de analizar información incompleta e imprecisa.

*Supervisory Control and Data Acquisition*, SCADA o sistemas de supervisión, control y adquisición de datos, esta tecnología sumada a la implementación de técnicas de inteligencia artificial ha planteado un nuevo comienzo para el manejo de toda la información de una planta industrial, a través de interfaces gráficas, sistemas de comunicación que se necesiten utilizar en cada caso, para el control y supervisión de esta. Asimismo, sirve de interfaz entre los niveles de control como los PLC o estaciones de trabajo, y los de gestión o toma de decisiones como las redes corporativas.

En la actualidad, la 4ta revolución industrial trajo consigo nuevos sistemas SCADA basados en IoT o Internet de las Cosas, los cuales permiten la transmisión y recepción

de datos a través de la nube, estas nuevas aplicaciones dieron paso a tecnologías de la computación como lo es la inteligencia artificial, la cual ha brindado muchas mejoras en términos de productividad y eficiencia del control de un proceso.

En el marco del estado del arte, se han referenciado investigaciones, en las cuales que los autores definen como un progreso efímero el desarrollo de la tecnología de la automatización, el control o mando de sistema complejo es uno de los mayores problemas en el campo de la inteligencia artificial, casos como el convertidor DC-DC de entradas múltiples (MIDC) que integra diferentes fuentes renovables como una tecnología en evolución gracias a las técnicas difusas de control, modelos matemáticos y enfoques numéricos en el dominio del tiempo para simular la dinámica de un barco o embarcación hundida que se eleva del fondo marino mediante sistemas de flotabilidad, etc.

Con la implementación de esta aplicación dual del sistema SCADA inteligente se desea dejar un software de uso didáctico para los estudiantes de la Facultad de Educación Técnica para el Desarrollo de la Universidad Católica de Santiago de Guayaquil, en el cual se conozcan los procedimientos a realizar para el desarrollo y posterior implementación de un sistema de supervisión, control y adquisición de datos a través de técnicas de lógica difusa.

Asimismo, también se espera disminuir los costos de diseño, instalación y funcionalidad de los sistemas utilizados actualmente para la fabricación en un entorno industrial, y a su vez brindará una automatización dinámica y versátil en términos de seguridad, mano de obra y toma de decisiones frente a los problemas que surgen constantemente en los momentos de operación normales dentro de la industria.



La realización de este proyecto de tesis se debe a la necesidad de desarrollar e interactuar por el momento, con una aplicación de inteligencia artificial en una simulación de un proceso industrial; en un principio se deseaba realizar la interacción con una aplicación de redes neuronales, pero debido a la complejidad del tema, las limitaciones del hardware en el que se implementaría la simulación de la aplicación y el tiempo de desarrollo requerido, se optó por una aplicación de lógica difusa, siendo esta la más adecuada para cumplir con los requerimientos anteriores durante el transcurso del proyecto de titulación.

Como objetivos se tiene planteados como objetivo principal desarrollar un algoritmo basado en lógica difusa aplicado en un sistema SCADA, para supervisar el posicionamiento y ensamblaje en una línea de producción vehicular y como objetivos específicos se encuentran: Definir los modelos matemáticos y parámetros de funcionamiento del entorno industrial escogido, determinadas a través de sus ecuaciones características, implementar un sistema de supervisión y control utilizando de la técnica de lógica difusa, a través de la programación de un script en lenguaje Python, ejecutar pruebas de supervisión del proceso industrial a través del sistema SCADA, y operatividad del script del sistema inteligente, a través de la comparación de las variables resultantes de la simulación, y la validación visual del correcto funcionamiento de este.

Como conclusiones se destaca el logro de la simulación del proceso industrial de tipo línea de ensamblaje vehicular a través del sistema SCADA y el algoritmo de I.A. empleados con lo cual se consiguió optimizar el control, supervisión y adquisición de datos de la producción, además, con la optimización y actualización continua del

programa se puede lograr resultados mucho más precisos en términos de posicionamiento y ensamblaje.

Los resultados de la aplicación a través de la técnica de lógica difusa generaron resultados aceptables con respecto a la eficiencia, cabe recalcar que la selección de este método se basa en las ventajas de costo-tiempo, además de la facilidad de programar y modificar en los momentos que se presentaron fallas técnicas en las simulaciones.

Para trabajos futuros, se plantea comenzar con la implementación de un servidor OPC en el cual el sistema SCADA desarrollado en CODESYS y el script programado en lenguaje Python para el algoritmo de lógica difusa, puedan transmitir y recibir las iteraciones producidas en las variables de la interfaz de usuario debido a los cambios en la simulación del proceso y del control inteligente del script de Python, de este modo, se puede realizar la lectura y escritura de las variables en tiempo real con las ventajas de computación en paralelo que ofrecen los servidores y así mejorar el rendimiento del algoritmo de I.A. disminuyendo los retrasos por tiempo de ejecución provenientes del hardware empleado, a su vez, se puede integrar técnicas de control y supervisión inteligente por redes neuronales siempre y cuando se disponga de estos servidores por la amplia generación de datos en el tiempo de ejecución de la simulación.

## CAPÍTULO 1

### 1.1. Planteamiento del problema

En la actualidad, los nuevos requerimientos de producción la industrial masiva, que utilizan altos estándares de calidad, empiezan a necesitar de cambios a nivel de producción con mejoras en la velocidad de los procesos, y una mayor eficiencia de los mismos, por lo que dentro del análisis de la pirámide de automatización industrial, se encuentran cambios graduales con respecto al desplazamiento de la mano de obra humana con respecto a la maquinaria, es decir, el ser humano ha llegado en cierto punto a ser reemplazado totalmente por sistemas automáticos de producción industrial a nivel de planta. Estos avances tecnológicos no solo permiten mantener un flujo de producción industrial constante, sino también durante las 24 horas del día.

Para empezar, Wang (2015) define a los sistemas SCADA como cualquier software que permite la adquisición remota de los datos de un proceso y a su vez permita, a través de herramientas de comunicación que se necesiten utilizar en cada caso, el control de este. Las nuevas aplicaciones basadas en el internet de las cosas dieron paso a tecnologías de la computación como lo es la inteligencia artificial, la cual brinda muchas mejoras en términos de productividad y eficiencia del control de un proceso.

En la actualidad, los sistemas de supervisión incorporan interfaces de comunicación con programas de I.A., uno de los más comunes es el controlador difuso. Según Wang (2015), un controlador difuso directo adaptativo que no requiere un modelo matemático preciso del sistema bajo control es capaz de incorporar reglas de control difuso if-then directamente en los controladores, garantizando la estabilidad global del sistema de circuito cerrado resultante en el sentido de que todas las señales involucradas estén uniformemente delimitadas.

Sin embargo, esto causa problemas a la hora de analizar procesos industriales que se rigen por modelos matemáticos discretos, ya que la función matemática es cambiante en el tiempo, la lógica difusa obtiene resultados de control menos precisos y con márgenes de error amplios, según Silva (2018) los algoritmos de control son "sólidos" o "inflexibles" y generalmente no pueden manejar un control inteligente "flexible" que puede implicar razonamiento e inferencia utilizando información incompleta, vaga, no decisiva y cualitativa, aprendizaje y autoorganización a través de experiencias y conocimientos pasados. Con este punto de vista, ¿Qué efecto tiene la aplicación de inteligencia artificial por lógica difusa optimizar la supervisión del proceso de línea de ensamblaje vehicular?

## **1.2. Justificación del problema**

Ghosh (2018), afirma que la aplicación dual de los sistemas SCADA en conjunto con software de inteligencia artificial, disminuyen los términos de complejidad del diseño de un sistema de supervisión, por lo que una implementación de este tipo de aplicaciones disminuirá drásticamente los costos de diseño, instalación y funcionalidad de los sistemas utilizados actualmente para la fabricación en un entorno industrial, y a su vez brindará una automatización dinámica y versátil en términos de seguridad, mano de obra y toma de decisiones frente a los problemas que surgen constantemente en condiciones de operación normales dentro de la industria.

En la ciudad de Guayaquil, estos sistemas aún no se encuentran al alcance masivo para las empresas, esto se debe a la fabricación de productos, en su gran parte de tipo artesanal, además de la producción de materia prima para exportación internacional. Según Zumba (2018), la industria Ecuatoriana en general aún mantiene condiciones artesanales o rústicas, en el mejor de los casos obsoletas, con máquinas como tableros

eléctricos, botoneras para operarios, líneas de producción de lazo abierto con una cantidad mínima de PLC'S para operaciones de conmutación y controles de fuerza que aún siguen siendo supervisadas por mano de obra humana dentro los procesos industriales, que con el paso del tiempo aún siguen manteniendo estas técnicas de trabajo que datan desde las primeras intervenciones de instrumentación para la automatización de las fábricas.

Asimismo, la industria automovilística presenta nuevas técnicas para automatizar sus procesos de diseño, ensamblaje y acabados del producto final, los cuales disminuyen drásticamente los procesos de construcción de un vehículo, de hecho, a través de investigaciones de la empresa TCS se asevera que la inversión en I.A. les ayudó a reducir costos de producción en un 10%., los fabricantes de autos también reportaron un incremento en ingresos de un 12% en promedio en las áreas relacionadas a iniciativas de IA. Sin embargo, a nivel regional, la implementación de estas tecnologías es más escasa a comparación de la inversión global.

### **1.3. Objetivos**

#### **1.3.1. Objetivo general**

Desarrollar un algoritmo basado en lógica difusa aplicado en un sistema SCADA, para supervisar el posicionamiento y ensamblaje en una línea de producción vehicular.

#### **1.3.2. Objetivos específicos**

- Definir los modelos matemáticos y parámetros de funcionamiento del entorno industrial escogido, determinadas a través de sus ecuaciones características.
- Implementar un sistema de supervisión y control utilizando de la técnica de lógica difusa, a través de la programación de un script en lenguaje Python.

- Ejecutar pruebas de supervisión del proceso industrial a través del sistema SCADA, y operatividad del script del sistema inteligente, a través de la comparación de las variables resultantes de la simulación, y la validación visual del correcto funcionamiento de este.

#### **1.4. Metodología de la investigación**

La investigación es de tipo correlacional, por lo que durante la realización del trabajo se tesis intervendrán diversas variables que surgen del modelado del proceso industrial de ensamblaje vehicular y en la posterior simulación, estas variables tendrán una relación que debe ser guardada en una base de datos, para ser después analizadas, y posteriormente descritas en una tabla de resultados, la cual definirá el porcentaje de eficiencia de la simulación.

El enfoque de la investigación es de tipo cuantitativo, por lo que se realizarán cálculos en relación con las variables relacionadas al posicionamiento y ensamblaje de las partes de un vehículo de uso común, además de las tabulaciones de los resultados en tablas para su fácil identificación y posterior uso en otras actividades.

#### **1.5. Métodos de investigación**

Los métodos que se han seleccionado para el desarrollo del presente trabajo de investigación son los siguientes:

Análisis y síntesis, a través de este método se puede valorar las fuentes de información bibliográfica, artículos científicos y criterios de expertos, relacionado con el objeto de estudio. Dentro de esta etapa se realizará la adquisición de información con respecto al tema propuesto.

Analítica: se define como la etapa en la cual se analiza de forma disgregada o como un todo nuestro sistema de simulación, en la cual se incluyen los modelos matemáticos del sistema, los parámetros de funcionamiento, entre otras variables resultantes de la operación de entorno industrial.

### **1.6. Tareas de investigación**

- Búsqueda del estado del arte actualizado con referencia al trabajo que se desarrollará.
- Conceptualización y anexo de la información recopilada en el marco teórico del trabajo de titulación.
- Desarrollo y aplicación de los conceptos investigados en las tareas que requieran análisis teórico y numérico.
- Determinación de las posibles variables que se utilizarán en el programa del SCADA.
- Recolectar información del entorno de visualización del SCADA y los elementos que intervienen en la simulación.
- Consultar documentación de la norma IEC 61131-3 para el texto estructurado del sistema SCADA.
- Desarrollo de alarmas para las variables críticas que intervienen en la visualización.
- Consultar la documentación del script de Python para la ejecución en el software de CODESYS.
- Investigar información para la programación de un algoritmo basado en la técnica de inteligencia artificial por lógica difusa.
- Implementación de gráficas para el seguimiento del proceso y almacenamiento de datos para la determinación de la eficiencia del sistema.

## CAPÍTULO 2: FUNDAMENTOS TEÓRICOS

### 2.1. Estado del arte

Para el desarrollo de los conceptos relacionados con sistemas SCADA e inteligencia artificial, además del marco teórico correspondiente, se expondrá a continuación el estado del arte con referencia a estos temas, los cuales fueron desarrollados a partir de fuentes actualizadas que permitirán una mejor comprensión del tema.

En las microrredes de CA isleñas que consisten en fuentes de energía renovables (RES), sistema de almacenamiento de energía a base de batería (BESS) y cargas, el BESS equilibra la diferencia entre la potencia y las cargas RES entregando / absorbiendo esa diferencia. Sin embargo, el estado de carga y la potencia de carga / descarga de la batería deben mantenerse dentro de sus límites de diseño, independientemente de las variaciones en la demanda de carga o la potencia intermitente de la RES. En este documento, se propone un controlador de supervisión basado en lógica difusa para asegurar que la energía y la energía de la batería no excedan sus límites de diseño y mantengan un flujo de energía estable. La microrred considerada en este documento consiste en un PV, batería, carga y unidad auxiliar auxiliar. El controlador de lógica difusa altera la frecuencia del bus de CA, que utilizan los controladores locales de las unidades paralelas para reducir la potencia generada por el PV o para complementar la potencia de la unidad auxiliar. El rendimiento del FLC propuesto se verifica por simulación y resultados experimentales. (Badwawi, Walid, Mallick & Abusara, 2018).

El convertidor DC-DC de entradas múltiples (MIDC) que integra diferentes fuentes renovables es una tecnología en evolución. Las investigaciones en este campo dieron varias topologías y configuraciones de convertidores de CC a CC de múltiples



entradas, para integrar múltiples fuentes. En este trabajo, primero se identifica una topología y se explora un controlador con administración de energía. Se selecciona una serie MIDC no aislada con tres fuentes de entrada para inyectar energía a la red de CC. Esta topología unidireccional con menos interruptores se selecciona para integrar fuentes renovables como celdas fotovoltaicas, eólicas, celdas de combustible, etc. a la red. La topología actúa como un convertidor elevador para integrar fuentes de bajo voltaje a la red de alto voltaje. La administración y control de energía de MIDC, conectado a la red, es una tarea difícil, ya que aún no se exploran técnicas viables. El controlador de lógica difusa se propone como control de supervisión para la administración de energía. El control de supervisión de lógica difusa (FLSC) decide la cantidad de energía que cada fuente debe suministrar, según la disponibilidad de energía. El controlador utilizado para el control de voltaje y corriente es un controlador PI simple. Los estudios de simulación de circuito cerrado se realizan con MATLAB / SIMULINK y los resultados experimentales obtenidos con OPAL-RT se utilizan para validar la configuración identificada para la administración de energía. Las principales ventajas de este convertidor son menos conteo de componentes, estructura compacta y utilización eficiente de la energía. (Rani, Navasree, George & Ashok, 2019).

Los autores definen como un progreso efímero en el desarrollo de la tecnología de la automatización, el control o gobierno del sistema es uno de los mayores problemas en el campo de la inteligencia artificial. En este documento se trata de llegar a un objetivo predeterminado con la información del sensor, es decir, un sistema capaz de realizar suposiciones adecuadas para el desempeño del monitoreo continuo de un servidor basado en un sistema SCADA. La metodología es de tipo analítica y a su vez experimental, ya que se realizan cálculos y estimaciones del comportamiento del sistema SCADA para la supervisión en conjunto con la aplicación de inteligencia

artificial, con posteriores implementaciones físicas con equipos de bajo costo. Los resultados de la investigación concluyen que el proyecto se realizó con una placa de Arduino en la cual la interfaz de usuario del proceso fue desarrollada en un sistema embebido y la información fue subida a un servidor utilizando conectividad con una base de datos en línea. (Sanap, Nawale, Kapse, Kale, & Korade, 2015).

En este documento se propone un nuevo sistema de innovación de monitoreo y control para conseguir un monitoreo en tiempo real de un sistema híbrido de energía renovable eólica-fotovoltaica. El objetivo de esta aplicación es tener un desempeño de medición de variables eléctricas en tiempo real, los cuales pueden ser transferidos de forma efectiva a un centro de monitoreo remoto utilizando comunicación por una red de intranet. La metodología usada fue de tipo experimental como resultado de utilizar implementaciones con varias etapas de PLC y medidores de poder digitales, además de consistir en dos generadores de inducción de viento, generadores, inversores, y un módulo fotovoltaico. Se ha concluido que los resultados de tipo simulación y experimental de este proyecto en el ámbito de las técnicas de monitoreo y control propuestas pueden conseguir control de supervisión en tiempo real además de la adquisición de datos de forma remota para varias formas de sistemas de energía renovable. (Liu, 2015).

Este artículo presenta un modelo matemático y un enfoque numérico en el dominio del tiempo para simular la dinámica de un barco / embarcación hundida que se eleva del fondo marino mediante sistemas de flotabilidad (inflado de gas) en una forma adecuada para integrar técnicas de control para garantizar la estabilidad hidrodinámica para una operación de salvamento segura y viable. De acuerdo con las ecuaciones de dos grados de libertad del movimiento de la embarcación de cuerpo rígido en el plano

de buceo, un controlador de modo deslizante convencional está diseñado como el controlador principal para regular la velocidad de flujo del gas de llenado dentro de las bolsas de elevación y un controlador PID está diseñado como El controlador secundario para regular la purga de gas a través de las válvulas montadas en bolsas elevadoras. Luego, un controlador de lógica difusa de supervisión está diseñado para monitorear o cambiar entre los controladores primario y secundario en función del requisito de flotabilidad. A partir de los estudios de simulación, se encuentra que el controlador supervisor de lógica difusa es capaz de mantener la estabilidad hidrodinámica definiendo adecuadamente las reglas lingüísticas difusas, que se crea en base a la experiencia del autor en la realización de simulación numérica utilizando controladores primarios y secundarios. (Velayudhan, 2019).

Uno de los principales factores que pueden aumentar la eficiencia de las turbinas eólicas (WT) es el control simultáneo de las diferentes partes en varias áreas de operación. El principal problema asociado con el diseño de control en el generador eólico es la presencia de un sistema asimétrico en el modelo dinámico del sistema, lo que hace complicado un esquema de control de supervisión genérico para la administración de energía de WT. En consecuencia, el controlador de supervisión se puede utilizar como el bloque de construcción principal de un controlador de parque eólico (offshore), que cumple con los requisitos del código de red y puede aumentar la eficiencia y la protección de los WT en (región II y III) al mismo tiempo. Este documento propone un nuevo controlador de supervisión adaptativo eficaz para la gestión y protección óptimas simultáneamente de un WT híbrido, en ambas regiones (II y III). Para este fin, el modo de deslizamiento de segundo orden con la ley de control de super-torsión de ganancia adaptativa y el control de lógica difusa se utilizan en el lado de la máquina, el lado de las baterías y los convertidores del lado de la red, para

lograr cuatro objetivos de control: (1) control de la velocidad del rotor para rastrear el valor óptimo; (2) control adaptativo (modo conmutativo) para el seguimiento del punto de máxima potencia (MPPT) o límite de potencia en varias regiones; (3) regula el voltaje promedio del enlace de CC cerca de su valor nominal; (4) asegura: una regulación uniforme con alta calidad de suministro de energía inyectada en la red, una corrección satisfactoria del factor de potencia y un alto rendimiento armónico en relación con la fuente de CA y eliminando el efecto parloteo. Los resultados de amplios estudios de simulación demuestran que el sistema de control de supervisión propuesto garantiza el seguimiento de las señales de referencia con un alto rendimiento armónico a pesar de las incertidumbres de perturbaciones externas. (Meghni, Dib, Azar & Saadoun, 2018).

El objetivo de este estudio fue desarrollar un sistema de control avanzado para optimizar el rendimiento de los reactores anaerobios de lecho fijo. El controlador tenía como objetivo maximizar la producción de biometano mientras controlaba el contenido de ácidos grasos volátiles en el efluente. Para este propósito, se desarrolló, ajustó y validó un controlador de lógica difusa en un reactor anaeróbico de lecho fijo a escala piloto (350L) que trata las aguas residuales de la bodega cruda. Los resultados mostraron que el controlador fue capaz de optimizar adecuadamente el rendimiento del proceso, maximizando la producción de metano en términos de caudal de metano, lo que resultó en un rendimiento promedio de metano de aproximadamente  $0.29 \text{LCH}_4\text{g}^{-1} \text{COD}$ . Por otro lado, el controlador mantuvo el contenido de ácidos grasos volátiles en el efluente cerca del límite máximo establecido ( $750 \text{ mg CODL}^{-1}$ ). Se espera que los resultados de este estudio faciliten a los ingenieros de planta establecer una estrategia de control óptima que permita un rendimiento adecuado del proceso con la máxima productividad de biometano. (Robles et. al.,2018).

Los autores de este documento establecen que las cajas de cambio han probado ser los mejores contribuidores hacia los tiempos de baja en las turbinas eólicas. El objetivo de este artículo busca reducir las fallas en las cajas de cambio provenientes del sistema de engranajes, indicaciones tempranas de posibles piezas de desgaste de la caja de engranajes pueden ser usadas para un efectivo mantenimiento predictivo, además de esto, los autores también hacen una propuesta de una red neuronal artificial usando la información de un sistema de supervisión, control y adquisición de datos. La metodología propuesta es principalmente de carácter experimental, por lo que el desarrollo se basa en un control implementado al sistema de engranajes para corregir los fallos de la caja de cambios, proceso por el cual ha pasado por varias etapas de pruebas y supervisión. Los resultados han demostrado que el enfoque de la red neuronal artificial propuesta en condiciones base es capaz de identificar severos daños en los componentes que han sido monitoreados con anterioridad. (Bangalore & Tjernberg, 2015).

La estimación del estado es una de las herramientas primarias para los sistemas de administración de energía (EMS). Los datos de estimaciones tradicionales son medidos con sistemas SCADA los cuales son más bien de baja precisión y no incluyen mediciones de las fases de los voltajes. El objetivo de la investigación es realizar una implementación de un algoritmo de estimación de estado el cual es capaz de incorporar las mediciones del SCADA y de la unidad de administración de energía (PMU). La metodología es de tipo experimental por lo que incorpora equipos para la medida de energía, análisis de variables en estados de energización y pruebas del funcionamiento sincronizado de ambas aplicaciones. Los análisis de los resultados obtenidos de los diferentes casos de estudio desarrollados pueden resaltar algunas ventajas al incorporar

las mediciones de PMU's en las estimaciones del estado de los sistemas de energización. (Salgado, Esquivel, & Guizar, 2015).

Las configuraciones híbridas que involucran energías renovables y dispositivos de almacenamiento plantean ciertos desafíos con respecto a sus estrategias de administración de energía, como la generación de energía intermitente y fluctuante a partir de fuentes renovables, la energía disponible que varía en el tiempo en los sistemas de almacenamiento o sus limitaciones máximas de carga y descarga. Es obligatorio observar estos aspectos para desarrollar una estrategia inteligente de gestión de energía dentro del sistema híbrido. Este capítulo presenta una estrategia de control para la operación coordinada de un generador de energía eólica y dos dispositivos de almacenamiento de energía diferentes. El esquema de control propuesto se basa en una lógica difusa para monitorear el estado de carga de los sistemas de almacenamiento, mientras define sus referencias de energía para cumplir con una demanda de red impuesta. La estrategia de control ha sido evaluada a través de la simulación en diferentes condiciones de operación, demostrando una regulación satisfactoria de los parámetros monitoreados y un suministro adecuado de los requisitos de la red. (Sarrias, Fernández, García & Jurado, 2019).

## **2.2. Inteligencia artificial**

En el campo de ciencias computacionales, la inteligencia artificial o A.I. en inglés (artificial intelligence), a veces llamada inteligencia de máquina es la inteligencia mostrada por los sistemas computacionales, robóticos o de carácter industrial, en contraste con la inteligencia natural demostrada por los humanos y otros animales. Estas ciencias definen las investigaciones sobre I.A. como el estudio de agentes inteligentes; cualquier dispositivo que percibe su entorno y realiza acciones para

maximizar las posibilidades de alcanzar sus metas de forma exitosa. Siendo más específicos, Kaplan y Haenlein (2018) definen la I.A. como “La habilidad de un sistema para interpretar de forma correcta datos externos, para aprender de dicha información, y para utilizar estos aprendizajes con el objetivo de conseguir metas específicas a través de la adaptación flexible”.

### **2.2.1. Aplicaciones de los sistemas de inteligencia artificial en la industria**

Según Chen et. al. (2017), la inteligencia artificial y los sistemas basados en el conocimiento han sido una rama de investigación activa de la inteligencia artificial durante todo el ciclo de vida del producto para el diseño del producto, la planificación de la producción, la distribución y los servicios de campo. Los sistemas de fabricación y las fábricas electrónicas no utilizaron el término "IA", pero amplían el modelado de sistemas de ingeniería para permitir la integración completa de los elementos en el ecosistema de fabricación para la gestión inteligente de operaciones.

#### **i) Robots colaborativos para la mejora de la productividad**

La inteligencia artificial es considerada como uno de los aspectos más importantes para las aplicaciones de automatización industrial. Con la ayuda de la I.A., el alcance y el ritmo de la automatización han cambiado fundamentalmente. Las tecnologías de inteligencia artificial aumentan el rendimiento y amplían la capacidad de las aplicaciones de inteligencia artificial convencionales. Un ejemplo son los robots colaborativos.



Fig. 2.1. Predix. – Robots colaborativos para guías manuales en la automatización.

Fuente: (Bélanger, 2015)

Los robots colaborativos se utilizan para trabajos de cooperación que requieren de la intervención de más de un robot. Sin embargo, se debe notar que este tipo de colaboración utiliza robots industriales normales, pero con un dispositivo adicional que "siente" las fuerzas que el trabajador está aplicando en la herramienta del robot. Bélanger (2015) indica que el dispositivo más popular para lograr dicha colaboración es un Sensor de par de fuerza, como el Robotiq FT 150. Se observa que este tipo de dispositivo simplemente leerá las fuerzas aplicadas en la herramienta del robot. Este tipo de colaboración solo se aplica al robot mientras realiza esta función particular, lo que significa que mientras el robot esté funcionando en sus otros modos, el robot debe tener un reemplazo en su lugar, y si ese caso no es posible, se espera a que finalice su trabajo.

### **2.2.2. Desafíos para la I.A. en la industria**

Los desafíos de la IA industrial para desbloquear el valor del sistema radican en la transformación de datos sin procesar en predicciones inteligentes para una toma de decisiones rápida. En general, existen cuatro desafíos principales en la realización de la IA industrial.



## **i) Datos**

Los sistemas de ingeniería ahora generan una gran cantidad de datos y la industria moderna es, de hecho, un entorno del “big data”. Sin embargo, los datos industriales generalmente están estructurados, pero pueden ser de baja calidad. Las cuestiones de las "3B"(en inglés: Bad, broken, background) del big data industrial son:

- *Errores*: La calidad de los datos puede ser deficiente y, a diferencia de otras aplicaciones orientadas al consumidor, los datos de los sistemas industriales suelen tener significados físicos claros, lo que dificulta la compensación de la calidad con el volumen.
- *Fallo*: Los datos recopilados para los modelos de aprendizaje automático de capacitación generalmente carecen de un conjunto completo de condiciones de trabajo y estados de salud / modos de falla, lo que puede causar falsos positivos y falsos negativos en la implementación en línea de los sistemas de I.A.
- *Fondo*: Los patrones de datos industriales pueden ser altamente transitorios e interpretarlos requiere experiencia en el dominio, que difícilmente puede ser aprovechada simplemente extrayendo datos numéricos. Lee (2015).

## **ii) Velocidad**

El proceso de producción se realiza rápidamente y el equipo y la pieza de trabajo pueden ser costosos, las aplicaciones de I.A. deben aplicarse en tiempo real para poder detectar anomalías de inmediato y evitar desperdicios y otras consecuencias. Las soluciones basadas en la nube pueden ser poderosas y rápidas, pero aun así no cumplirían con ciertos requisitos de eficiencia de computación. La computación perimetral puede ser una mejor opción en tal escenario.

### **iii) Requisitos de alta fidelidad**

A diferencia de los sistemas de recomendaciones de la I.A. orientados al consumidor que tienen una alta tolerancia a los falsos positivos y negativos, incluso una tasa muy baja de falsos positivos o negativos puede costar la credibilidad total de los sistemas de I.A. Según Yao (2017), las aplicaciones industriales de inteligencia artificial suelen tratar problemas críticos relacionados con la seguridad, la fiabilidad y las operaciones. Cualquier falla en las predicciones podría incurrir en un impacto económico y / o de seguridad negativo en los usuarios y desalentarlos a confiar en los sistemas de inteligencia artificial.

### **iv) Interpretabilidad**

Además de la precisión de la predicción y la fidelidad del rendimiento, los sistemas industriales de inteligencia artificial también deben ir más allá de los resultados de la predicción y proporcionar un análisis de la causa raíz de las anomalías. Esto requiere que, durante el desarrollo, los científicos de datos necesiten trabajar con expertos en dominios e incluir conocimientos de dominio en el proceso de modelado, y que el modelo aprenda y acumule conocimientos tales como el conocimiento.

### **2.2.3. Lógica difusa**

La lógica difusa es una forma de lógica de muchos valores en la que los valores de verdad de las variables pueden ser cualquier número real entre 0 y 1 inclusive. Se emplea para manejar el concepto de verdad parcial, donde el valor de verdad puede oscilar entre completamente verdadero y completamente falso. Por el contrario, en la lógica booleana, los valores de verdad de las variables solo pueden ser los valores enteros 0 o 1.

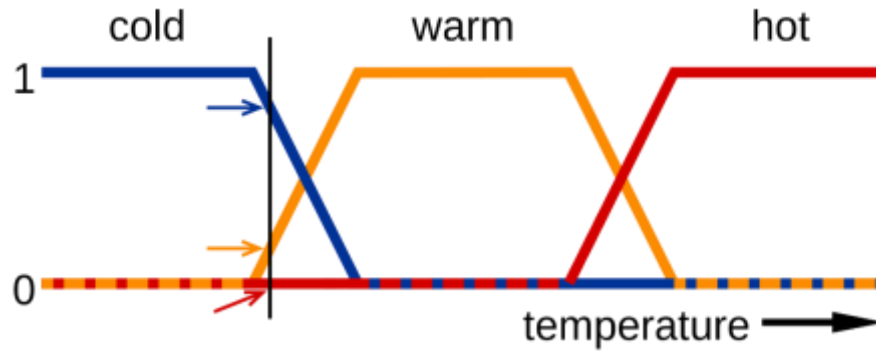


Fig. 2.2. Temperatura analizada en lógica difusa.

Fuente: (Wierman,2012)

En esta imagen, los significados de las expresiones frío, cálido y caliente están representados por funciones que representan una escala de temperatura. Un punto en esa escala tiene tres "valores de verdad", uno para cada una de las tres funciones. La línea vertical en la imagen representa una temperatura particular que miden las tres flechas (valores de verdad). Como la flecha roja apunta a cero, esta temperatura puede interpretarse como "no caliente". La flecha naranja (apuntando a 0.2) puede describirlo como "ligeramente caliente" y la flecha azul (apuntando a 0.8) "bastante fría".

La lógica difusa funciona con los valores de pertenencia de una manera que imita la lógica booleana. Para este fin, los reemplazos para operadores básicos AND, OR, NOT, deben estar disponibles. Hay varias formas de hacerlo. Un reemplazo común se llama los operadores de Zadeh:

Tabla 2.1. Operadores de lógica difusa

Boolean	Fuzzy
AND(x,y)	MIN(x,y)
OR(x,y)	MAX(x,y)
NOT(x)	1 - x

Fuente: (Di Stefano y Bruno N., 2013)

Para verdadero con valor 1 y falso con valor 0, las expresiones difusas producen el mismo resultado que las expresiones booleanas. También hay otros operadores, de naturaleza más lingüística, llamados coberturas que pueden aplicarse. Estos son generalmente adverbios tales como muy, o algo, que modifican el significado de un conjunto utilizando una fórmula matemática. En lógica matemática, hay varios sistemas formales de "lógica difusa", la mayoría de los cuales se encuentran en la familia de lógicas difusas de norma  $t$ . Las proposiciones lógicas difusas más importantes son:

- Lógica difusa proposicional basada en la norma  $t$  de tipo MTL, es una axiomatización de la lógica donde la conjunción se define mediante una norma  $t$  continua izquierda y la implicación se define como el residuo de la misma norma. Sus modelos corresponden a construcciones algebraicas de tipo MTL que son retículas residenciales integrales limitadas conmutativas pre lineales.
- La lógica difusa proposicional básica BL es una extensión de la lógica MTL donde la conjunción se define mediante una norma  $t$  continua, y la implicación también se define como el residuo de la norma ya mencionada. Sus modelos corresponden a construcciones algebraicas de tipo BL.
- La lógica difusa de Łukasiewicz es la extensión de la lógica difusa básica BL, donde la conjunción estándar es la norma  $t$  de Łukasiewicz. Tiene los axiomas de la lógica difusa básica más un axioma de negación doble, y sus modelos corresponden a construcciones algebraicas de tipo MV.
- La lógica difusa de Gödel es la extensión de la lógica difusa básica BL donde la conjunción es la  $t$ -norma de Gödel. Tiene los axiomas de BL más un

axioma de idempotencia de conjunción, y sus modelos se llaman modelos algebraicos de tipo G.

- La lógica difusa del producto es la extensión de la lógica difusa básica BL donde la conjunción es la norma t del producto. Tiene los axiomas de BL más otro axioma para la cancelatividad de conjunción, y sus modelos se denominan álgebras de productos.
- La lógica difusa con sintaxis evaluada (a veces también llamada lógica de Pavelka), denotada por EVL, es una generalización adicional de la lógica matemática difusa. Mientras que los tipos anteriores de lógica difusa tienen una sintaxis tradicional y una semántica de muchos valores, en EVL también se evalúa la sintaxis. Esto significa que cada fórmula tiene una evaluación. La axiomatización de EVL proviene de la lógica difusa de Łukasiewicz. Una generalización del teorema de integridad de Gödel clásico es demostrable en EVL.

En esta sección, se describe brevemente cómo la lógica difusa extiende la lógica booleana clásica (o, de manera equivalente, cómo la teoría de conjuntos difusa generaliza la teoría de conjuntos cantoriana). Dado un universo (nítido) del discurso X, un subconjunto borroso A de X se caracteriza por su función de pertenencia.

$$\mu_A : X \rightarrow [0,1],$$

Donde para  $x \in X$ , el número  $\mu_A(x)$  se interpreta como el grado de pertenencia de x en el conjunto difuso A o, de manera equivalente, como el valor de verdad de la declaración 'x es un elemento de A'. La función de pertenencia de un conjunto difuso es una generalización natural de la función característica de un subconjunto (clásico) A de X,

$$1_A : X \rightarrow \{0,1\},$$

Asignando a cada elemento  $x$  en  $X$  el valor 1 cuando  $x$  pertenece a  $A$ , y el valor 0 en caso contrario. Para generalizar las operaciones de conjuntos teóricos como intersección y unión (o las correspondientes operaciones lógicas booleanas conjunción y disyunción, respectivamente), se necesitan normas triangulares y subelementos: una norma triangular (norma  $t$ ) es una operación binaria en  $[0; 1]$ , es decir, una función  $T: [0,1]^2 \rightarrow [0,1]$ , que es conmutativo, asociativo, monótono en ambos componentes y satisface la condición de contorno

$$T(x, 1) = x. \quad (1)$$

Si  $T$  es una norma  $t$ , entonces la norma dual triangular (conorma  $t$ )  $S: [0,1]^2 \rightarrow [0,1]$ , está definido por

$$S(x, y) = 1 - T(1 - x, 1 - y). \quad (2)$$

Hay muchas, de hecho, infinitas, normas  $t$  y conormas  $t$ , solo algunas de las cuales se utilizan en aplicaciones. Las normas  $t$  más importantes, junto con sus conormas tipo  $t$  duales, son las siguientes:

*Mínimo  $T_M$ , Máximo  $S_M$*

$$T_M(x, y) = \min(x, y), \quad S_M(x, y) = \max(x, y) \quad (3)$$

*Producto  $T_p$ , Suma Probabilística  $S_p$*

$$T_p(x, y) = x \cdot y, \quad S_p(x, y) = x + y - x \cdot y \quad (4)$$

*Norma  $t$  de Lukasiewicz  $T_L$ , Suma delimitada  $S_L$*

$$T_L(x, y) = \max(x + y - 1, 0), \quad S_L(x, y) = \min(x + y, 1) \quad (5)$$

Norma t más debil  $T_W$ , conorma T más fuerte  $S_W$

$$T_W(x, y) = \begin{cases} \min(x, y) & \text{si } \max(x, y) = 1 \\ 0 & \text{de otro modo,} \end{cases} \quad (6)$$

$$S_W(x, y) = \begin{cases} \min(x, y) & \text{si } \max(x, y) = 0 \\ 1 & \text{de otro modo,} \end{cases} \quad (7)$$

Dada una t-norma T, su doble conorma t S, y los subconjuntos difusos A, B de un universo X, las funciones de pertenencia de la intersección  $A \cap B$ , la unión  $A \cup B$  y el complemento  $A^c$  se dan por:

$$\mu_{A \cap B}(x) = T(\mu_A(x), \mu_B(x)), \quad (8)$$

$$\mu_{A \cup B}(x) = S(\mu_A(x), \mu_B(x)), \quad (9)$$

$$\mu_{A^c}(x) = 1 - \mu_A(x), \quad (10)$$

Los valores  $A \setminus B(x)$ ,  $A [B(x)$  y  $A^c(x)$  describen los valores de verdad de las afirmaciones 'x es elemento de A Y x es elemento de B', 'x es elemento de A O x es el elemento de B', y 'x NO es el elemento de A', respectivamente.

Dada una t-norma T, su doble conorma t S y proposiciones P, Q con valores de verdad  $\|P\|$  y  $\|Q\|$ , respectivamente, hay dos extensiones principales de la implicación booleana  $P \Rightarrow Q$ :

Implicación S,

$$\|P \Rightarrow_s Q\| = S(1 - \|P\|, \|Q\|), \quad (11)$$

Implicación R,

$$||P \Rightarrow_R Q|| = \sup\{\alpha \in [0,1] \mid T(\alpha, ||P|| \leq ||Q||)\}. \quad (12)$$

En la lógica booleana, es decir, con valores de verdad 0 y 1 solamente, las implicaciones S y R siempre son equivalentes, lo que ya no es cierto en la lógica difusa: para la norma  $t$  de  $T_M$ , la implicación S correspondiente se convierte en la implicación de Kleene-Dienes y la implicación de R es la implicación de Gödel que es bastante diferente. Tenga en cuenta, sin embargo, que en el caso de Lukasiewicz, la norma  $t$  de  $T_L$ , las implicaciones S y R coinciden: ambos producen la implicación de Lukasiewicz.

### 2.3. Sistemas SCADA

Antunes et al. (2018) definen a los SCADA como el control de supervisión y adquisición de datos, es decir una arquitectura de sistema de control que utiliza computadoras, comunicaciones de datos en red e interfaces gráficas de usuario para la gestión de supervisión de procesos de alto nivel, pero utiliza otros dispositivos periféricos como el controlador lógico programable (PLC) y controladores PID discretos para Interfaz con la planta de proceso o maquinaria. El uso de SCADA también se ha considerado para la gestión y las operaciones del proceso impulsado por el proyecto en la construcción.

Cada sitio de campo realiza el control local de los actuadores y supervisa sus sensores asociados. Una Unidad de Telemetría Remota (RTU) es una unidad de control y adquisición de datos para propósitos especiales, diseñada para soportar las operaciones de las estaciones remotas SCADA.



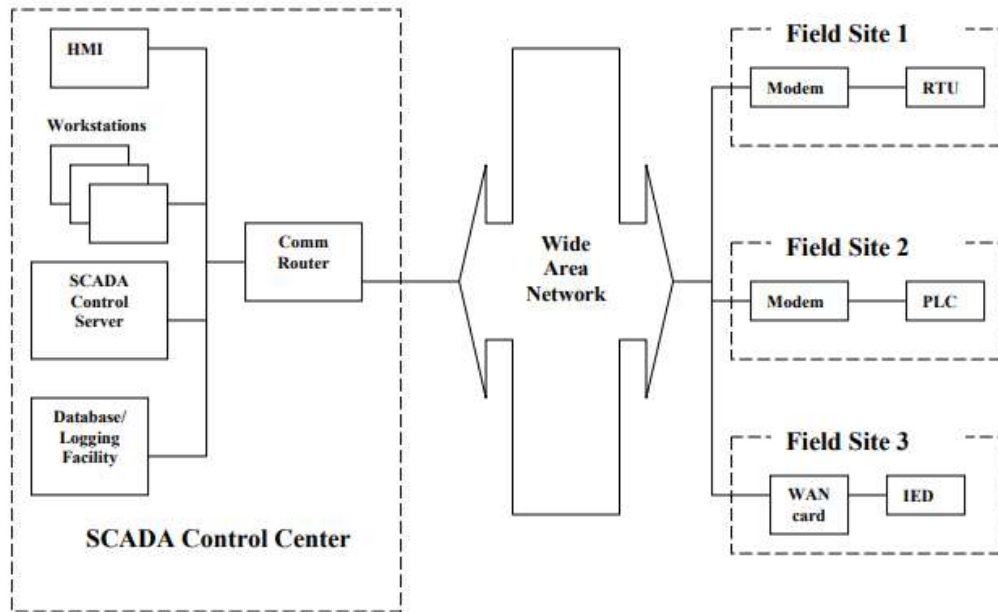


Figura 2.3. Diseño general de un sistema SCADA

Fuente: (Tang, 2012)

RTUs son dispositivos de campo a menudo equipados con interfaces de radio inalámbricas para soportar situaciones en las que las comunicaciones por cable no son posibles. Un controlador lógico programable (PLC) es una pequeña computadora industrial diseñada originalmente para realizar funciones lógicas ejecutadas por hardware eléctrico como relés, interruptores y temporizadores / contadores mecánicos. Los PLC de hoy se han convertido en controladores sofisticados con la capacidad de controlar procesos complejos.

### 2.3.1. Arquitectura de un sistema SCADA

El atributo clave de un sistema SCADA es su capacidad para realizar una operación de supervisión en una variedad de otros dispositivos propietarios. El diagrama adjunto es un modelo general que muestra los niveles de fabricación funcional mediante el control computarizado.

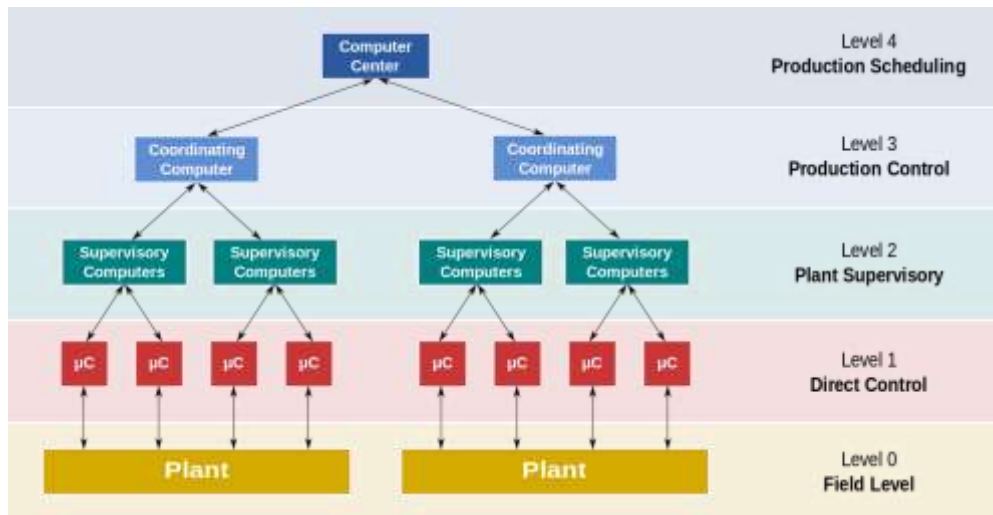


Fig. 2.4. Niveles funcionales de una operación de control para la fabricación.

Fuente: (Tang, 2012)

Refiriéndose al diagrama:

- El nivel 0 contiene los dispositivos de campo, como los sensores de flujo y temperatura, y los elementos de control finales, como las válvulas de control.
- El nivel 1 contiene los módulos de entrada / salida (E / S) industrializados y sus procesadores electrónicos distribuidos asociados.
- El nivel 2 contiene las computadoras de supervisión, que recopilan información de los nodos del procesador en el sistema y proporcionan las pantallas de control del operador.
- El nivel 3 es el nivel de control de producción, que no controla directamente el proceso, sino que se ocupa de monitorear la producción y los objetivos.
- El nivel 4 es el nivel de programación de la producción.

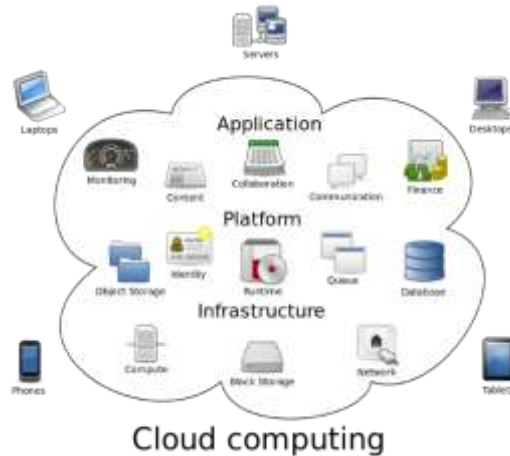


Fig. 2.5. Computación en la nube.

Fuente: (Baburajan, 2014)

Metáfora de computación en la nube: el grupo de elementos en red que proporcionan servicios no necesita ser dirigido o administrado individualmente por los usuarios; en cambio, toda la suite del hardware y software administrada por el proveedor puede considerarse como una nube amorfa.

### 2.3.2. CODESYS

Codesys (normalmente estilizado como CODESYS, un mantel de puertos para el sistema de desarrollo de controladores, anteriormente se lo diseñó como CoDeSys) es un entorno de desarrollo para la programación de aplicaciones de controladores según el estándar industrial internacional IEC 61131-3.

CODESYS es desarrollado y comercializado por la compañía de software alemana 3S-Smart Software Solutions ubicada en la ciudad bávara de Kempten. La versión 1.0 se lanzó en 1994. Las licencias de CODESYS son gratuitas y pueden instalarse legalmente sin protección contra copia en otras estaciones de trabajo. La herramienta

de software cubre diferentes aspectos de la tecnología de automatización industrial con una sola superficie.

### **i) Ingeniería de programación**

Los cinco lenguajes de programación para la programación de aplicaciones definidos en la IEC 61131-3 están disponibles en el entorno de desarrollo CODESYS.

- IL (lista de instrucciones) es un lenguaje de programación tipo ensamblador (ahora está en desuso, pero está disponible para compatibilidad con versiones anteriores).
- ST (texto estructurado) es similar a la programación en Pascal o C.
- LD (diagrama de escalera) permite al programador combinar virtualmente contactos y bobinas de relé.
- FBD (diagrama de bloques de funciones) permite al usuario programar rápidamente expresiones tanto booleanas como analógicas
- SFC (tabla de funciones secuenciales) es conveniente para la programación de procesos y flujos secuenciales

También se dispone de un editor gráfico adicional disponible en CODESYS no definido en el estándar IEC:

- CFC (Continuous Function Chart, en inglés) es un tipo de editor FBD a mano alzada. Aparte del editor de FBD orientado a la red donde las conexiones entre las entradas, los operadores y las salidas se configuran automáticamente, el programador debe dibujarlas. Todas las casillas se pueden colocar libremente, lo que hace posible programar bucles de retroalimentación sin variables provisionales.

## ii) Características

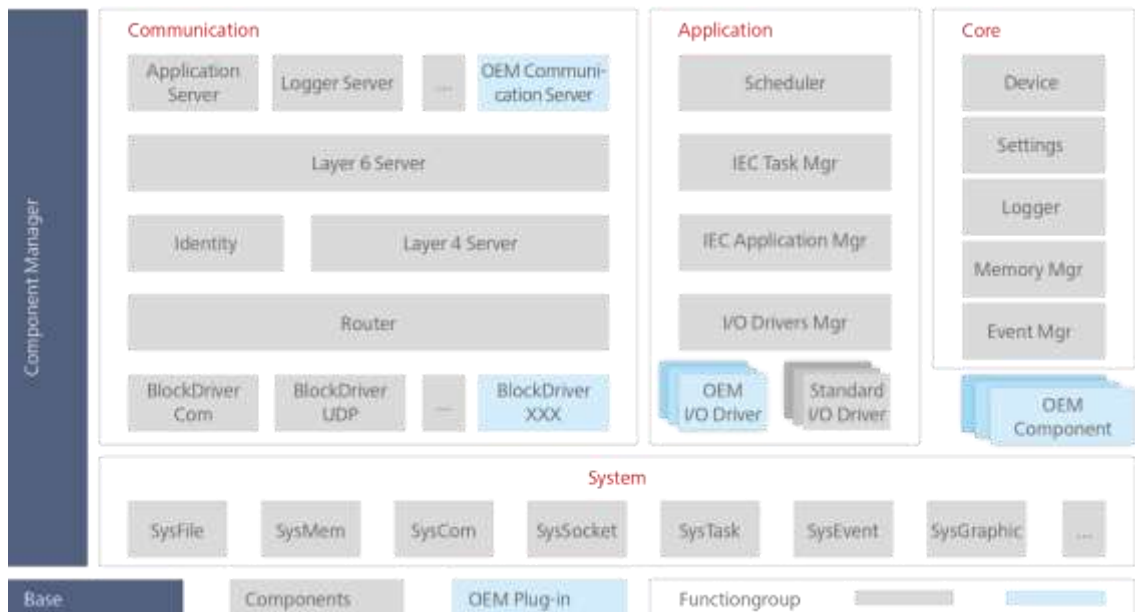


Fig. 2.6. Administrador de componentes y funciones disponibles para el software de CODESYS

Fuente: (CODESYS, 2018)

El tamaño y la funcionalidad del sistema de tiempo de ejecución modular se pueden escalar, fácil adaptación a los diferentes requisitos y entornos del sistema, además se habla de la realización de un dispositivo de automatización inteligente con gastos calculables y alta aceptación en el mercado: los fabricantes de componentes inteligentes se benefician de las amplias características de CODESYS en el desarrollo de sus dispositivos:

- Producto disponible como un sistema modular de tiempo de ejecución de una sola fuente para diferentes plataformas de dispositivos.
- Soporte para todos los procesadores comunes y sistemas operativos para sistemas industriales.

- Gastos de desarrollo proyectables: desarrollo de controladores industriales con características técnicas pronunciadas por medio de interfaces definidas y soporte de adaptación calificado.
- Numerosos sistemas y plataformas de referencia disponibles (por ejemplo, para sistemas ARM / Cortex populares o sistemas integrados).
- Implementación de propiedades específicas del dispositivo por medio de módulos de biblioteca (por ejemplo, para funciones de E / S propias).
- Adaptación y capacidad de ampliación de la interfaz de usuario mediante módulos de “plug-in” específicos del fabricante (por ejemplo, plantillas, editores, configuradores y asistentes).
- Integración en arquitecturas de sistemas existentes mediante interfaces o estándares de propiedad exclusiva, como OPC / OPC UA.

### **iii) Estructuras básicas de Python para la interfaz con el sistema de supervisión inteligente**

Todos los objetos y comandos que CODESYS proporciona para las secuencias de comandos también se encuentran en el módulo “scriptengine” de Python. Cada vez que se inicia un script, resulta un `<code>` implícito de tipo `scriptengine import *` `</code>`. Esto permite un fácil acceso a CODESYS. Sin embargo, si el script importa módulos que requieren acceso a las API de CODESYS, estos módulos tienen que importar el módulo “scriptengine” por sí mismos. A continuación, se encontrarán los principales objetos (categorías) que se pueden usar en los scripts de Python como puntos de entrada. Para obtener documentación completa sobre los puntos de entrada, consulte la documentación de referencia de API para el motor de secuencias de comandos CODESYS.

Tabla 2.2. Estructura de acceso a la plataforma de CODESYS con las sentencias de Python

<b>Objetos</b>	<b>Descripción</b>
System	<p>Acceso a funcionalidades generales de CODESYS, tales como:</p> <ul style="list-style-type: none"> <li>• Salir de CODESYS</li> <li>• Manejo de la interfaz de usuario general</li> <li>• Acceso a la memoria de mensajes (incluidos los mensajes del compilador).</li> <li>• Control de retardo y barras de progreso.</li> </ul>
Projects	<p>Accede al proyecto CODESYS como un árbol de objetos que combina las tres vistas del navegador (dispositivos, POU, módulos) en un árbol del proyecto. También permite cargar, crear, guardar y cerrar proyectos.</p> <p>Para la mayoría de los objetos en un proyecto, existen métodos especiales con funcionalidad detallada, por ejemplo, compilación, acceso a ST POU, exportación, importación, configuración de dispositivos, etc.</p>
online	<p>Acceso a funcionalidades online, tales como:</p> <ul style="list-style-type: none"> <li>• Iniciar sesión en dispositivos y aplicaciones</li> <li>• Gestión de los datos de acceso (nombre de usuario, contraseña)</li> <li>• Realización de escaneos de red.</li> <li>• Gestión de pasarela</li> </ul>
librarymanager	Permite la administración de repositorios de bibliotecas y la visualización, instalación y eliminación de bibliotecas.
Device_repository	Manejo de repositorios de dispositivos; Importación y exportación de descripciones de dispositivos.
Modulerepository	Gestión de los módulos de CODESYS Application Composer y los repositorios de CODESYS Application Composer.

Fuente: Elaborado por el autor

En el recuadro anterior se describen las funciones más comunes utilizadas para interactuar con el software CODESYS a través de un script generado en lenguaje Python en el cual se pueden utilizar dichas sentencias para realizar diversas acciones en el programa, desde interactuar con los mensajes del compilador, imprimir

comandos, lectura y escritura de datos, hasta la descarga de librerías en línea y el envío de las variables de cualquier simulación en formato “.project” por correo electrónico.

#### **2.4. Línea de ensamblaje industrial**

Las líneas de ensamblaje están diseñadas para la organización secuencial de trabajadores, herramientas o máquinas y piezas. La moción de los trabajadores se minimiza en la medida de lo posible. Todas las piezas o ensamblajes se manejan ya sea mediante transportadores o vehículos motorizados, como elevadores de horquilla o gravedad, sin camiones manuales. El levantamiento pesado lo realizan máquinas como grúas aéreas o montacargas. Cada trabajador generalmente realiza una operación simple.

De acuerdo con Henry Ford (1922), los principios para las líneas de ensamblaje son los siguientes:

- (1) Coloque las herramientas y los hombres en la secuencia de la operación de modo que cada parte del componente recorra la menor distancia posible durante el proceso de acabado.
- (2) Utilice toboganes de trabajo u otra forma de transportador para que cuando un trabajador complete su operación, se deje caer la pieza siempre en el mismo lugar, que siempre debe ser el lugar más conveniente para su mano, y si es posible haga que la gravedad lleve la pieza a el próximo trabajador por su cuenta.
- (3) Utilice líneas de montaje deslizantes (en este caso motorizadas), por las cuales las piezas a ensamblar se entregan a distancias convenientes.



### **i) Ensamblaje por modelos mixtos**

La línea de ensamblaje de modelos mixtos consiste en ensamblar diferentes tipos de productos en la misma línea de producción de acuerdo con diferentes secuencias tecnológicas. La configuración de tareas, el tiempo de operación, las restricciones lógicas entre tareas y otros requisitos de producción varían según los productos. Por lo tanto, las tareas en cada estación variarán cuando sea necesario.

Las características del modelado permiten ignorar el costo de la conversión de un tipo a otro. Proporciona la capacidad del sistema de producción para responder rápidamente a los cambios en la demanda y no necesita mantener un gran stock de repuestos. La planificación de la línea de ensamblaje de modelos mixtos de automóviles en ADC primero debe resolver el problema del equilibrio. En la actualidad, el método de planificación es asignar cada operación de la unidad a diferentes procesos de trabajo, y cada proceso es responsable de un trabajador o más trabajadores y uno o más equipos principales. La secuenciación de la línea de ensamblaje de modelos mixtos de automóviles puede minimizar el tiempo de inactividad total y el tiempo de sobrecarga, y mejorar la eficiencia de toda la línea de ensamblaje.

La línea de ensamblaje de modelos mixtos es una forma de sistema de ensamblaje flexible. Está compuesto por un equipo de ensamblaje flexible, un dispositivo de almacenamiento y transporte de logística, un sistema de programación de control computarizado y el personal de administración de producción necesario para cumplir con la producción multivariada. Incluye múltiples unidades flexibles de trabajo de montaje.

## **CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA SCADA INTELIGENTE**

### **3.1. Consideraciones iniciales del sistema.**

Para el desarrollo del sistema de control inteligente se necesita tener en cuenta las necesidades del proceso industrial y el alcance de este a nivel de producción, para ello se deben analizar en detalle los requerimientos de producción en un determinado intervalo de tiempo, los modelos que rigen a los procesos de la red de ensamblaje vehicular, los cuales permiten tener un conocimiento ideal del funcionamiento del sistema, de esta forma se obtiene la pauta para empezar a moldear la aplicación de I.A.

#### **3.1.1. Descripción del sistema SCADA inteligente**

La aplicación que se ha desarrollado cuenta con varias especificaciones que permiten tener una idea clara de las prestaciones que ofrece el sistema de supervisión inteligente, las cuales son críticas para el correcto funcionamiento y ejecución constante de la simulación del sistema de línea de ensamblaje.

En la operación del sistema SCADA inteligente, se encuentran puntos como la simulación de las variables que representan a el proceso industrial en condiciones de operación normales, es decir los valores asignados a cada elemento que compone la visualización y el comportamiento de cada uno, posteriormente se realizará el almacenamiento de datos del proceso de manera continua en una base de datos local proporcionada por la API de la aplicación de CODESYS.

A continuación, se describirá el sistema en la siguiente gráfica de forma breve:

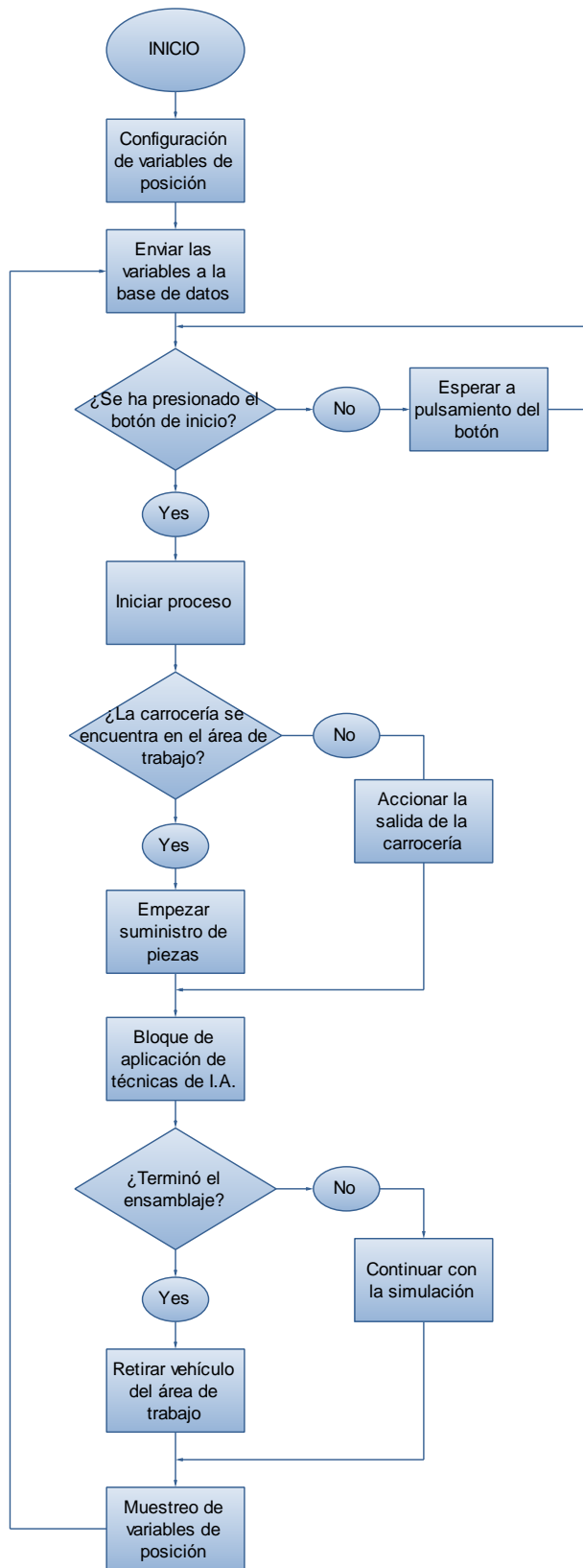


Fig. 3.1. Diagrama de flujo del sistema SCADA empleado para la simulación

Fuente: El autor

Continuando con el control y supervisión inteligentes de la simulación de la línea de producción, realizado a través de un script de inteligencia artificial el cual ejecuta el control inteligente a través de la técnica de lógica difusa utilizando el método Mamdani que se explicará más adelante; el programa además incluye un algoritmo de detección de colisiones o “hitbox”, este realizará las funciones de los sistemas de posicionamiento físicos integrados por sensores ópticos, las comparaciones de las variables de posición realizadas a través de este método permitirán obtener un control más preciso en la simulación, cuya función representa el pilar fundamental del control de todo el sistema de ensamblaje industrial.

### **3.1.2. Caracterización del sistema SCADA inteligente**

Dentro de los parámetros que se van a aplicar, el sistema debe funcionar siguiendo una estricta norma de ajuste continuo, pero básicamente se basa en realizar los siguientes pasos que han sido explicados anteriormente. El diseño final para simulación de nuestro entorno industrial consta de las siguientes partes:

- Interfaz de programación de sistemas SCADA llamado CODESYS para la adquisición de datos continuos y supervisión del funcionamiento del sistema en condiciones de operación a través de las variables y los comportamientos que rigen el proceso de línea de ensamblaje vehicular.
- Elementos para controlar y supervisar en el proceso de ensamblaje: Carrocería y piezas del vehículo modelo Charger R/T de 1970.
- Sensores de posición ópticos (simulados por algoritmos de detección de colisiones) para transmitir las variables requeridas por el sistema de supervisión inteligente para el ensamblaje de las partes del vehículo.

- Línea de producción compuesta por cintas transportadoras y brazos robóticos para el suministro y manipulación de materiales hacia el área de trabajo.
- Herramientas externas API (Python) para diseñar las rutinas de programa compuestas por el sistema inteligente encargado de la supervisión y control inteligentes de la simulación los cuales debe responder continuamente a los cambios producidos en la simulación.
- Sistema de almacenamiento para los datos del proceso, proveída por la API del software de CODESYS, que permita la interacción constante entre el software de CODESYS y el programa de I.A.
- Hoja de cálculo virtual que se utilizará para el almacenaje de las variables del proceso por parte del usuario y para realizar las gráficas de los resultados correspondientes al proceso.

### **3.1.3. Funciones del sistema SCADA inteligente**

El proyecto de la “simulación de un sistema SCADA a nivel industrial a través de la aplicación de un sistema de inteligencia artificial” se define como un conjunto de funciones que deben sincronizarse en las condiciones de operación de la simulación, es por eso por lo que hay que tener en claro todas las funciones que se realizarán con la aplicación de la I.A.:

La función principal del sistema se define como:

- La supervisión continúa de la base de datos y de la simulación del proceso en base a las variables generadas por los cambios de este en la línea de ensamblaje industrial.

Para las funciones secundarias se encuentran las siguientes:

- La adquisición de los datos de la simulación de forma continua.
- Simulación del entorno del proceso industrial de línea de ensamblaje vehicular a través de la visualización de este en el software de supervisión.
- Mediciones de posicionamiento en el transcurso de la simulación.
- Transmisión de las variables al PLC del nivel de campo.
- Adquisición de las variables por el RTU y la base de datos para acumular la información del proceso.
- Envío de la información del proceso al sistema de supervisión inteligente.
- Interfaz de comunicación con el programa de I.A. aplicada por la técnica de lógica difusa.
- Ejecución de la máquina de inferencias en la que se ha definido los casos de posición para realizar el ensamblaje.

Así, se establecen las funciones y características del sistema, además, los fundamentos consisten en un método de ejercicio de dicha arquitectura. El control exacto de los sistemas de supervisión requiere los estándares industriales SCADA y también las últimas funciones generadas en el estado del arte para la I.A. que se va a evaluar.

### **3.2. Análisis del sistema SCADA inteligente**

A continuación, se presentará el análisis correspondiente a todos los elementos que se utilizarán durante la simulación. En primera instancia, a todo el diseño que involucra el sistema SCADA, para eso se hará hincapié a la relación que tiene con un sistema de adquisición de datos típico. Se empezará haciendo el análisis de las arquitecturas

utilizadas, los módulos del sistema, la visualización general de la planta y los procedimientos desarrollados previo a la simulación del sistema.

### **3.2.1. Arquitectura del sistema SCADA**

Para la arquitectura del sistema SCADA se plantearán los elementos que intervienen en la simulación de la aplicación, los cuales se pueden dividir en tres bloques principales:

- Software de adquisición de datos y control (SCADA).
- Sistemas de adquisición y mando (sensores y actuadores).
- Sistemas de interconexión (comunicaciones).

De esta forma, se puede interpretar con mayor facilidad la distribución de la arquitectura utilizada, la cual contiene los siguientes niveles jerárquicos:

- Nivel de proceso
- Nivel de campo
- Nivel de comunicaciones
- Nivel de supervisión

Adicional a esto, el diseño de la arquitectura se basa en las nuevas tecnologías aplicadas a la actual generación de sistemas SCADA basado en los avances propuestos en la 4ta revolución industrial: Iot o Internet de las Cosas (Internet of things en inglés), gracias a esto se pueden implementar esta clase de aplicación en la computación en la nube.

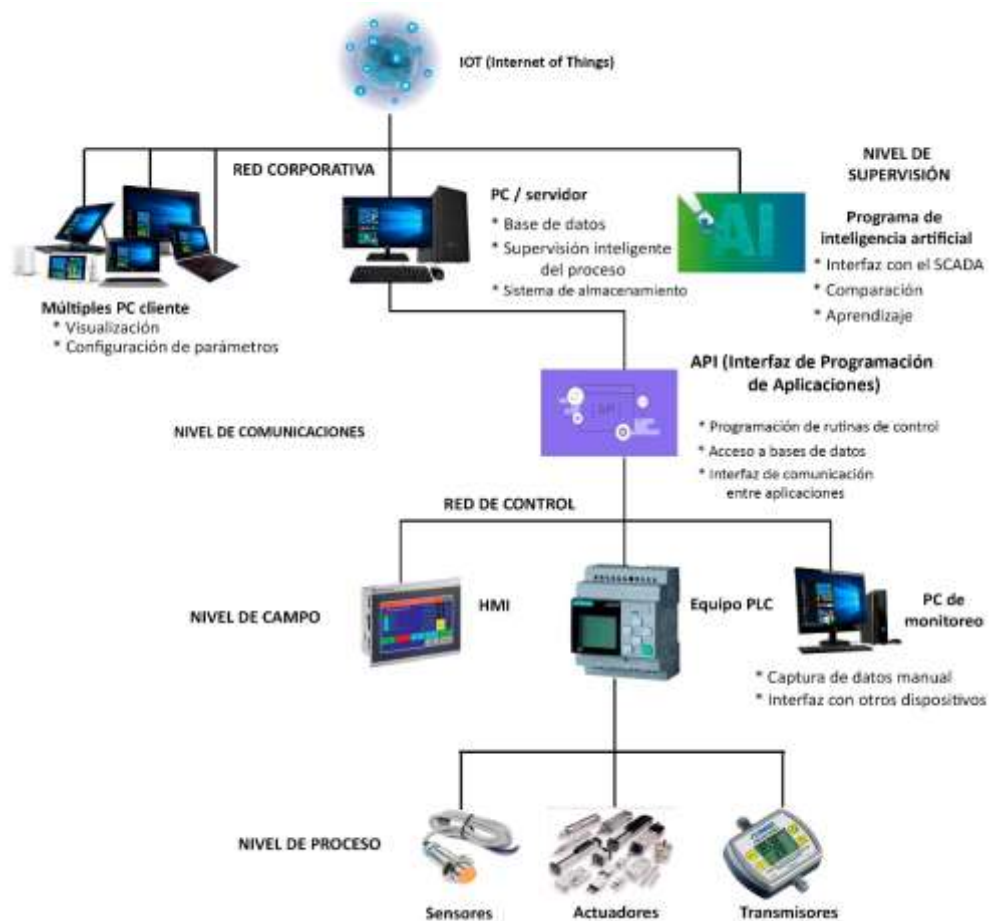


Fig. 3.2. Arquitectura del sistema SCADA

Fuente: El autor

La figura anterior muestra la arquitectura SCADA integrada que admite TCP / IP, UDP y otros protocolos de comunicación basados en IP, así como protocolos industriales como Modbus TCP, Modbus sobre TCP o Modbus sobre UDP. Todo esto puede funcionar a través de redes celulares, privadas de radio o satélite.

Dentro del nivel de supervisión se encuentran los elementos que intervienen directamente con el proceso, los sensores adquirirán información de este continuamente y estas mediciones serán enviadas a través del transmisor correspondiente al PLC que rige dichos elementos a través del protocolo Ethernet industrial, el cual utiliza buses de campo, cualquier cambio en la respuesta que genere



el SCADA se verá reflejada en las salidas del PLC y posteriormente en los actuadores que controlan el proceso.

Para el nivel de campo las mediciones adquiridas por el PLC serán transmitidas al HMI que contiene los elementos correspondientes en la visualización la cual puede ser operada en tiempo real por el usuario, al mismo tiempo estas variables también serán enviadas a un PC de monitoreo que supervise continuamente el estado de los elementos del nivel de campo y proceso, y, a su vez, generar alarmas cuando existan problemas en estos niveles.

En el nivel de comunicaciones se necesita realizar una interfaz que permita la interacción entre el sistema SCADA y el algoritmo de inteligencia artificial, para eso se utiliza la API (Interfaz de Programación de Aplicaciones) de CODESYS, la cual contiene los módulos de interacción con la API externa que rige el script de I.A. y un intérprete de dicho lenguaje, facilitando las labores de comunicación entre ambas plataformas y evitando la implementación de un servidor (comúnmente se utilizan OPC) para la transmisión y recepción de datos entre ambas API.

La elección de la comunicación existente depende de la caracterización de una serie de factores. Los factores son la lejanía, las comunicaciones disponibles en los sitios remotos (si hubiesen), la infraestructura de comunicaciones existente, la frecuencia de sondeo y las tasas de datos. Estos factores impactan la decisión final para la arquitectura SCADA. Por lo tanto, una revisión de la evolución de los sistemas SCADA permite una mejor comprensión de muchos problemas de seguridad. De este modo se escogió la API de CODESYS como medio de comunicación para ambas aplicaciones.

El nivel de supervisión corresponde a las labores hechas por el sistema SCADA inteligente, inteligente por la interacción con un algoritmo de I.A., en este nivel se recibe la información actual del proceso y gracias a la labor de la I.A. se ejecutarán modificaciones según los requerimientos necesarios en la simulación, dichas modificaciones en las variables serán enviadas de vuelta a los niveles de campo y proceso para así recibir el cambio esperado.

#### **i) Arquitectura de red**

Para obtener una vista genérica de la arquitectura de red, la comunicación entre el SCADA y el script inteligente funciona como la única interfaz entre el centro de control y el nivel de campo y proceso. De esta forma se emplea la API de CODESYS la cual se implementa generalmente con software específico del proveedor y sus servicios se basan en el estándar IEC 61131-3 que contiene el módulo *scriptengine* el cual puede ser llamado desde el comando de importación del lenguaje de programación Python.

La siguiente figura proporciona una vista genérica de una arquitectura de red SCADA, que se utiliza a lo largo de este documento como marco para discutir los elementos funcionales, de diseño y de interconectividad del sistema SCADA empleado.

En la siguiente figura, se observa que el sistema SCADA tiene dos componentes principales: la red industrial (sitio inferior de la figura anterior) y la conexión a la nube y redes WAN (sitio superior de la figura anterior). Los dos componentes están conectados entre sí a través de un cortafuego.

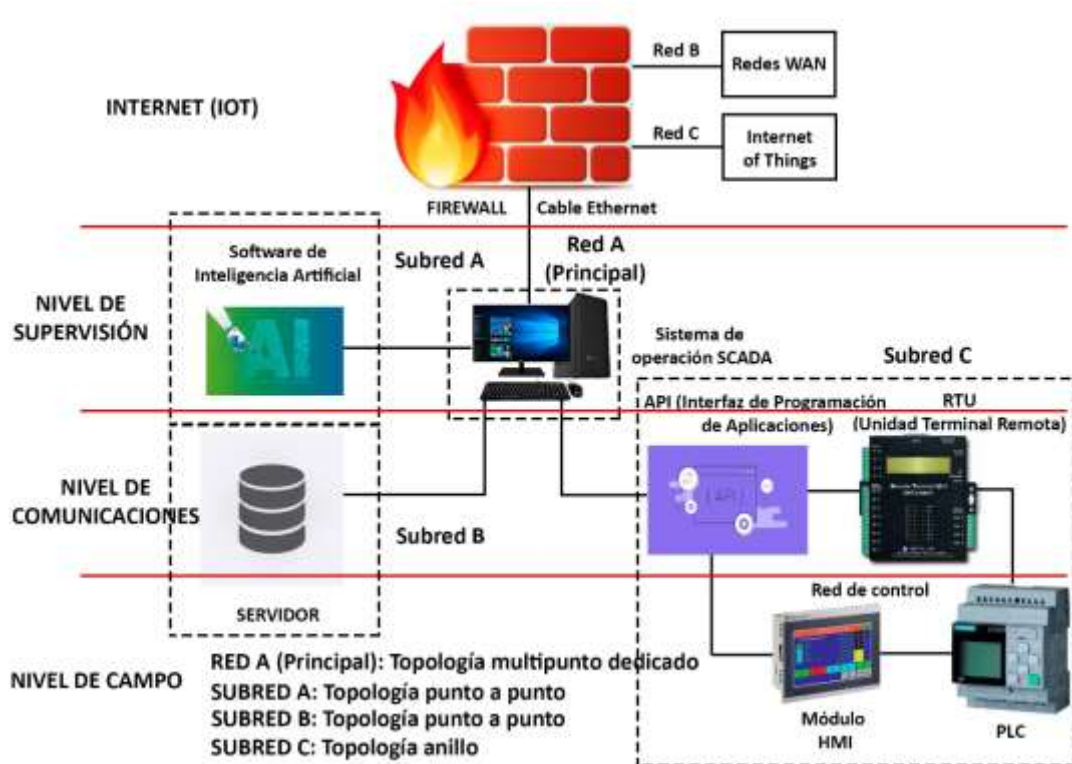


Fig. 3.3. Arquitectura de red del sistema SCADA

Fuente: El autor

La red industrial es el centro de las operaciones que envían información del proceso al sistema SCADA. Además, cuando el PLC transmite la información del proceso al sistema de supervisión a través de los RTU (Terminales de acceso remoto), los componentes adicionales incluyen interfaces hombre-máquina (HMI), estaciones de trabajo de ingeniería, historiadores de datos de plantas, bases de datos y varios recursos compartidos, los cuales son utilizados para la visualización de la simulación por parte del usuario.

La red principal representa el sistema SCADA en el cual todos los buses de campo de las subredes correspondientes convergen:

- La subred A representa el bloque de inteligencia artificial, el cual se comunica con el SCADA a través de la API de CODESYS.
- La subred B el cual representa el sistema de almacenamiento que emplea el SCADA para guardar información del proceso.
- La subred C corresponde al nivel de campo y de proceso del entorno industrial, aquí se encuentran los elementos de comunicación, visualización y control que ejecutarán las acciones de mando recibidas por el sistema SCADA inteligente de la red principal.

A su vez, en el nivel de Internet se tienen el resto de las redes, la red B corresponde a las redes WAN o redes de área amplia, para dar cobertura de red a sitios remotos de la planta y la red C corresponde al Iot o internet de las cosas el cual establece capacidades al SCADA para la computación en la nube.

El sistema SCADA está conectado a internet y además puede comunicar a otros SCADA mediante sistemas de red de área amplia (WAN) a través de líneas de datos o teléfono. Estos sistemas utilizan Ethernet o conexiones de fibra óptica para transmitir datos entre los nodos con frecuencia. Este método de comunicación representa a la familia de SCADA de tercera generación que utilizan controladores lógicos programables (PLC) para monitorear y ajustar los operadores de marcado de rutina solo en caso de que se requieran decisiones importantes. Además, puede informar el estado del proceso en tiempo real mediante el uso de la escala horizontal, criterio básico de la escalabilidad de la computación en la nube, considerado como criterio de la arquitectura de cuarta generación; por lo tanto, se pueden implementar algoritmos de control más complejos que son prácticamente difíciles de implementar en PLC tradicionales.

### 3.2.2. Módulos del sistema

Los módulos del sistema representan las tareas que realiza el sistema SCADA con la aplicación desarrollada, las cuales se proceden a definir en etapas que contienen diversas tareas que se deben realizar previa, durante y después de la simulación del proceso en cuestión.

Es así como se tienen los siguientes módulos:

- Configuración
- Control del proceso
- Registro y archivado
- Alarmas y eventos
- Interfaz gráfica
- Comunicaciones

En la configuración se establece y detalla el entorno de trabajo según las variables y los cambios que requiera la simulación en su momento.

El control del proceso se refiere a procesos o elementos que se utilizan para la programación de tareas que requieran una variable del proceso cuando ocurra un evento determinado.

Registro y archivado muestra el almacenamiento temporal de valores en tamaños limitados y patrones cíclicos, cuando el registro se llena, se guardan copias en un archivo en la memoria, que puede ser accedido en cualquier momento por el usuario.

Las alarmas y eventos representan los momentos en los que la simulación experimenta cambios notables en sus variables, para el primer caso si dichos cambios

alteran el funcionamiento correcto del sistema notificarán al usuario para que pueda realizar las respectivas correcciones, y en el segundo se muestra en pantalla los cambios generados o actualizaciones de los estados del proceso.

La interfaz gráfica contiene la visualización de los elementos que intervienen en el proceso, aquí se realizó la pantalla de usuario con los elementos de control, texto y actuadores.

Las comunicaciones establecen el intercambio de información entre los elementos de la planta, la arquitectura implementada y los elementos de gestión involucrados. Aquí se establecen los controles que permiten el intercambio de información entre los elementos de campo y los computadores que reciben y almacenan dicha información.

#### **i) Configuración**

La aplicación dada para el sistema SCADA en la figura 3.5 se divide en áreas de proceso. Cada una de esas áreas tiene operaciones claramente definidas que deben realizarse. Por ejemplo, los motores de las cintas transportadoras pueden usar dos o tres motores, que pueden funcionar en modo de avance / espera. El programa de automatización en el controlador está configurado para operar los motores en base a los puntos de ajuste ingresados por el operador artificial y las tareas asignadas.

En la interfaz de la visualización intervienen algunos elementos, primero la visualización del HMI que permite la observación continua de cualquier cambio que surja durante el proceso, asimismo, el sistema SCADA se comunica constantemente con su PLC para consultar el estado de la simulación.

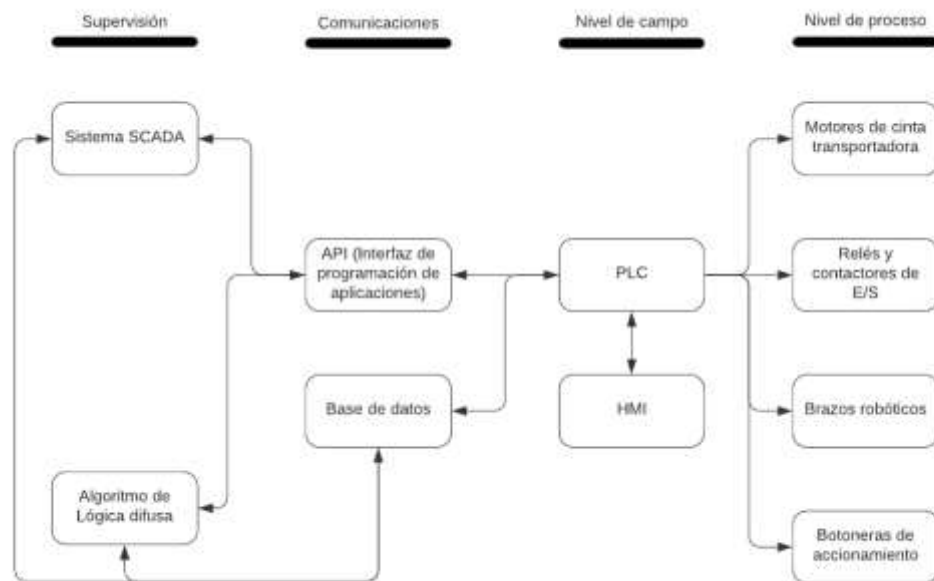


Fig. 3.4. Arquitectura del módulo de configuración

Fuente: El autor

En este caso el PLC envía datos sobre el funcionamiento de motor de la cinta transportadora y de las cadenas del riel, los brazos robóticos y las botoneras que se utilizan para las funciones de marcha y paro del proceso, además, el SCADA estará en constante escritura y lectura de su base de datos, consultando cualquier cambio de las variables de posicionamiento que haya realizado el programa de I.A. para efectuar el control de la simulación.

## ii) Control del proceso

En este punto se utilizan las variables preprogramadas para realizar acciones de mando o condicionamientos al sistema dependiendo de las líneas de código que se encargan de la lectura y escritura de los valores que se van generando, en este caso, la programación realizada en CODESYS es de tipo SL (lenguaje estructurado).

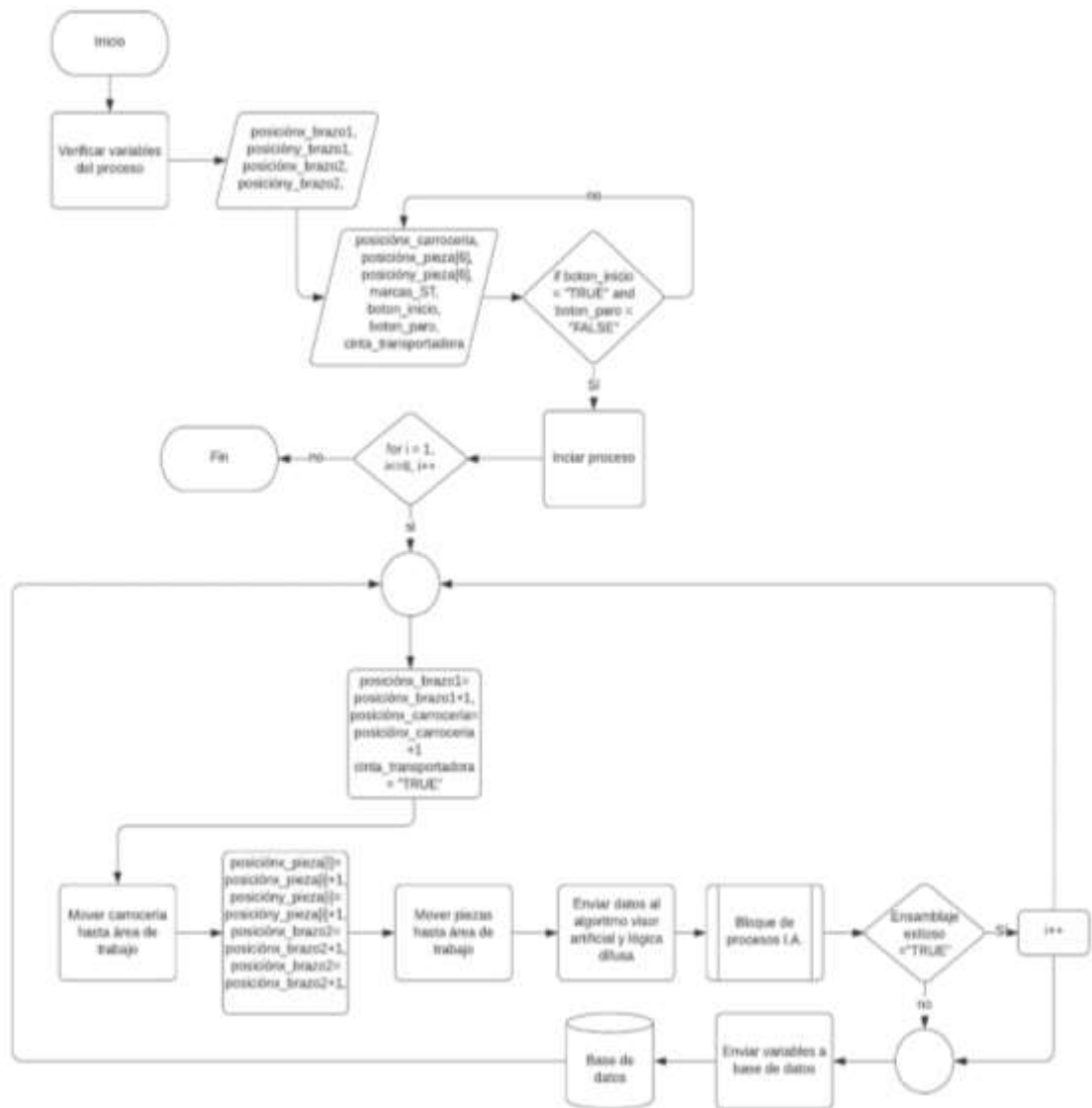


Fig. 3.5. Diagrama de flujo que indica el control del proceso

Fuente: El Autor

Además, se encuentran variables que intervienen en el funcionamiento del programa, las variables de `marcas_ST`, `botón_inicio`, `botón_paro`; son de tipo booleanas, estas variables permiten manejar puntos de consigna para el cambio de secuencias en el programa, además, las variables de tipo posición son de tipo entero, estas indican la posición de las piezas en el área de trabajo en el que se va a desarrollar la simulación. Además, el SCADA modificará los valores de posición hasta que alcancen el área de trabajo, después de esto, el sistema comunicará a los algoritmos



inteligentes, que los elementos están preparados para empezar con la aplicación de la inteligencia artificial, esta realizará el posicionamiento adecuado ejecutando sus estructuras de decisión y reglas de inferencia difusas, basadas en el método Mamdani, posteriormente el algoritmo decidirá si se debe realizar el ensamblaje o seguir posicionando la pieza actual, luego estos datos se enviarán de vuelta al SCADA para que pueda efectuar estos cambios, finalizando con el almacenamiento de estos valores en una base de datos externa.

### **iii) Registro y archivado**

En esta sección el programa almacena las variables en una ventana la cual puede ser accedida en tiempo real por cualquier dispositivo de o software de control, el proceso de almacenamiento se realiza de forma ordenada y según las líneas de programación, en este caso de texto estructurado, esto cual facilita las tareas de búsqueda y selección de variables tanto como para el usuario como para el dispositivo o aplicación externa la cual puede disponer en tiempo real dicha información.

Para el almacenamiento de las variables el software vincula estos datos, y los almacena en una variable global perteneciente al archivo en el cual fue construido, además, guarda el tipo de dato para mayor facilidad de detección; estos elementos también pueden ser modificados desde esta ventana en el caso que se requiera reasignar, eliminar, cambiar el tipo de datos o moverlos a otra variable global.

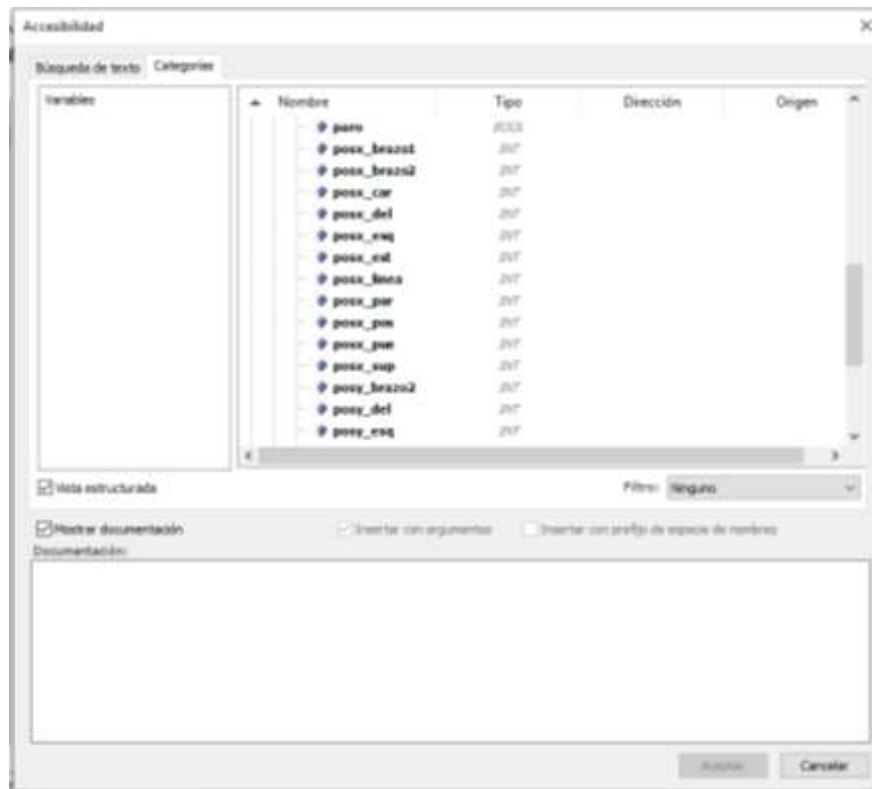


Fig. 3.6. Almacenamiento de las variables disponibles en la base de datos de CODESYS

Fuente: El autor

#### iv) Alarmas y eventos

En condiciones normales de operación de la simulación del proceso industrial, pueden ocurrir eventos que no estén considerados en los parámetros de la aplicación, como disfunciones en los ejes de los motores de la línea transportadora o de los brazos robóticos, obstrucciones que impidan la continuidad del proceso, etc. En estas circunstancias el sistema debe responder a estas situaciones y detener el proceso, y en la medida de lo posible, detectar el origen del evento previo a la desconexión del sistema.

Asimismo, pueden ocurrir eventos que indiquen el progreso de la simulación, dichos eventos pueden ser registrados directamente en la ventana de comandos a través del algoritmo de lógica difusa, o en una ventana de alarmas, la cual indique la secuenciación de todos los estados relacionados al proceso

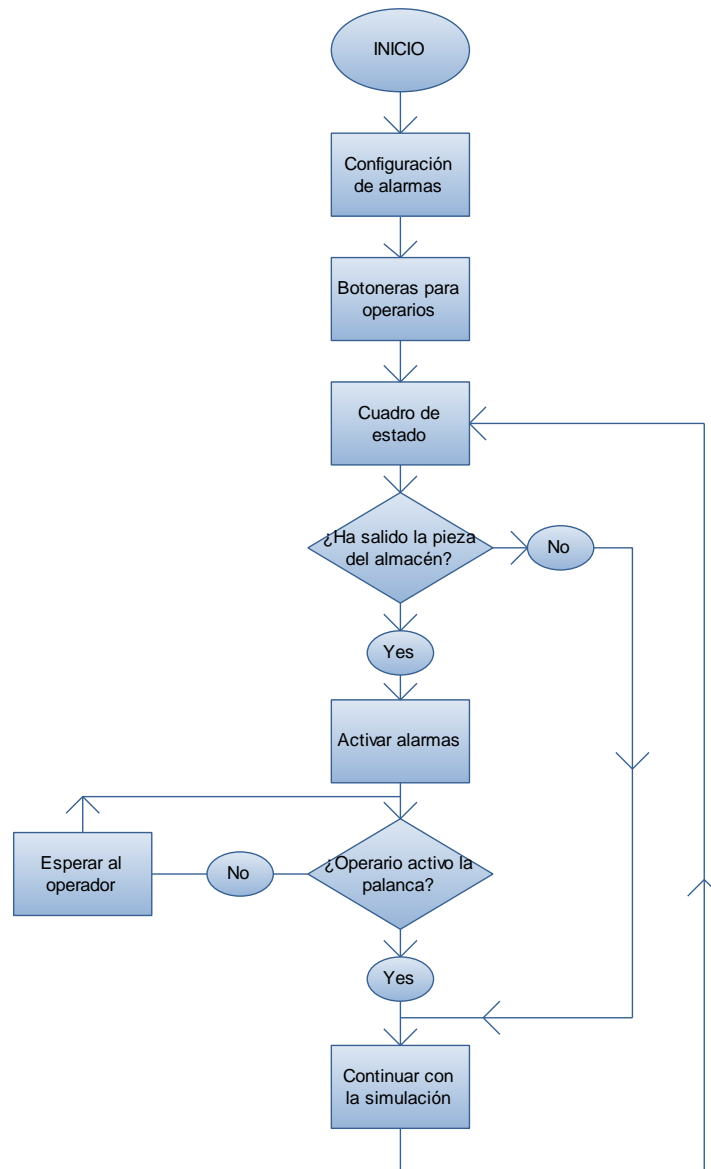


Fig. 3.7. Diagrama de flujo de alarma de piezas

Fuente: El autor

En la simulación del programa, pueden ocurrir eventos que no estén dentro de los parámetros de la programación tales como un movimiento mal realizado, inserción incorrecta de una pieza al chasis, y en este caso, el corte de suministro de piezas para el ensamblaje, este evento daría lugar a la generación de una alarma que debe ser notificada al operador del proceso en la visualización en línea del sistema SCADA, estas notificaciones pueden venir con texto incluido para facilitar el reconocimiento

del error y la hora de la alarma; cuando una notificación suceda, el operador debe verificar el problema que esté sucediendo, dar paso a la corrección y posteriormente, presionar las palancas para continuar con la operación.

En este proyecto se muestra cómo acceder a objetos de alarma mediante un programa. Con el objeto AlarmManager, es posible acceder a las alarmas activas y almacenadas. También es posible registrar un cliente para recibir eventos de alarma. En las siguientes figuras, se muestran las alarmas ingresadas en la aplicación, además para realizar una gestión de alarmas simple, se crean dos alarmas simples de diferentes clases y configurando los elementos requeridos. La configuración de alarma debe agregarse a la aplicación en el proyecto, y luego debe agregarse el Grupo de alarma para introducir las condiciones de alarma.



Fig. 3.8. Cuadro de alarmas configuradas para la simulación del proceso

Fuente: El autor

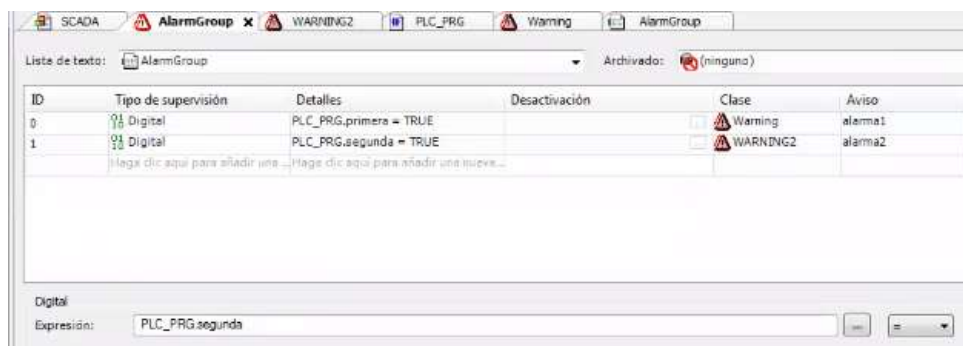


Fig. 3.9. Vinculación de las variables de alarma de la simulación a los elementos de supervisión

Fuente: El autor

El proyecto de muestra contiene dos tablas de alarmas. El primero (tipo Alarmtable) muestra los eventos activos y almacenados. Se pueden crear nuevas alarmas mediante los botones debajo de la tabla. La segunda tabla simula una tabla de alarmas y muestra las alarmas almacenadas.

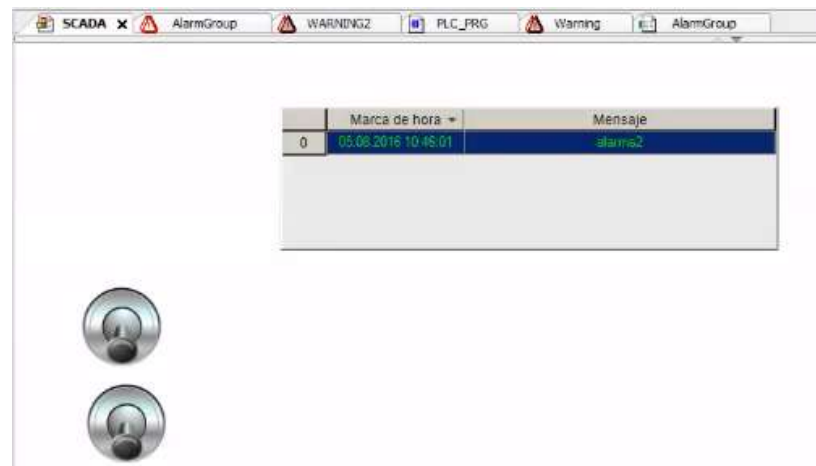


Fig. 3.10. Visualización de la activación de las alarmas en la simulación con sus palancas de respuesta

Fuente: El autor

#### v) Interfaz gráfica

La interfaz del operador, también conocida como interfaz hombre-máquina (HMI) permite la interacción entre el usuario y el sistema (el ordenador). La capacidad de ingresar señales y leer la salida le permite al operador supervisar el adecuado funcionamiento del sistema. Este control puede ser representado a través de las herramientas gráficas que se encuentran en pantalla.

En la interfaz aparecen los elementos que intervienen en el proceso, en la siguiente figura se tienen los siguientes elementos:

- Botoneras que contienen los pulsadores de marcha y paro
- Lámparas que indican los estados anteriores
- Un temporizador que muestra el tiempo que ha transcurrido en la simulación

- Carrocería del vehículo Charger R/T
- Piezas del vehículo: Parabrisas, estribos, puertas, parachoques delantero, parachoques trasero y el travesaño superior
- Brazos robóticos para el agarre y posterior traslado de la carrocería y las piezas que serán ensambladas
- Cintas transportadoras
- Carriles de movimiento para los brazos robóticos

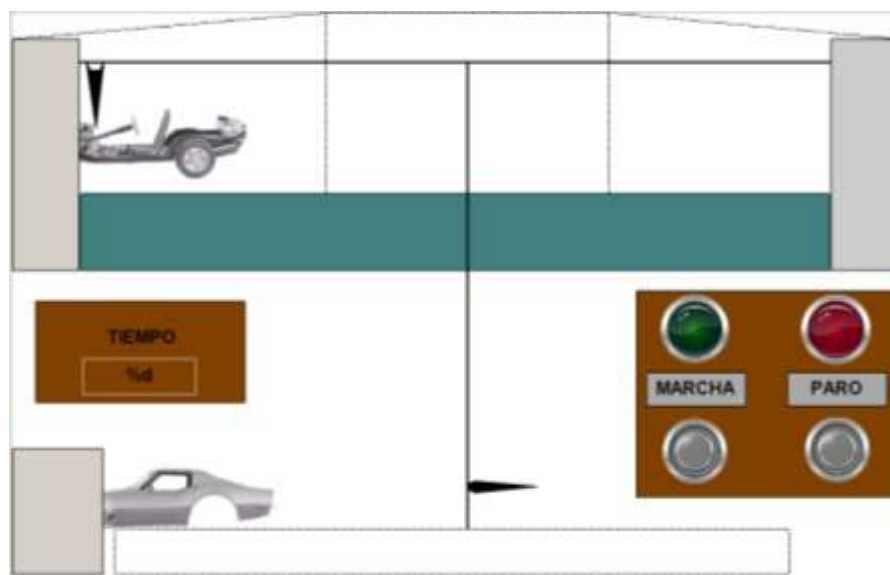


Fig. 3.11. Interfaz gráfica del sistema de supervisión

Fuente: El autor

En la figura 3.11. se encuentran los elementos con los que el usuario puede interactuar, primero el operador inicia la simulación presionando el botón de marcha, el cual a su vez da paso al ingreso de las partes que se ensamblarán en la carrocería del vehículo en cuestión; además del botón de paro, este elemento detendrá el movimiento de la línea de ensamblaje en el caso de requerirse, ya sea por alarmas u otro evento que efectúe cambios inesperados en el proceso, así como realizar modificaciones en el sistema o las partes a ensamblar en la visualización del SCADA.

## vi) Comunicaciones

Las comunicaciones que realiza el sistema SCADA, pueden darse de diferentes formas, entre el SCADA y los elementos del proceso, y con la interfaz implementada con el algoritmo de lógica difusa, esto permite transferir o recibir información multiplataforma, entre la planta y la arquitectura de hardware que soporta el SCADA, y el algoritmo de lógica difusa, la comunicación se realiza a través de la API de CODESYS a través del módulo “scriptengine” el cual puede reconocer estructuras de programación del intérprete de Python, así se define la comunicación entre diferentes plataformas, siendo este caso entre la plataforma de CODESYS y el software de Python.

En la figura 3.12., se representan los dispositivos con los cuales el software de CODESYS se puede comunicar por defecto, en esta muestra se pueden observar algunos protocolos comunes para la comunicación entre plataformas de desarrollo, la comunicación comienza con el acceso a las variables del proceso que serán constantemente almacenadas en la API de CODESYS, la cual enviará estos valores al intérprete de Python que transformará el código para ser legible por el algoritmo de lógica difusa, después de este evento, el algoritmo generará un resultado que será enviado de vuelta a dicha interfaz, así se continúa con el envío de datos al PLC de campo simulado para que efectúe los cambios en el proceso.

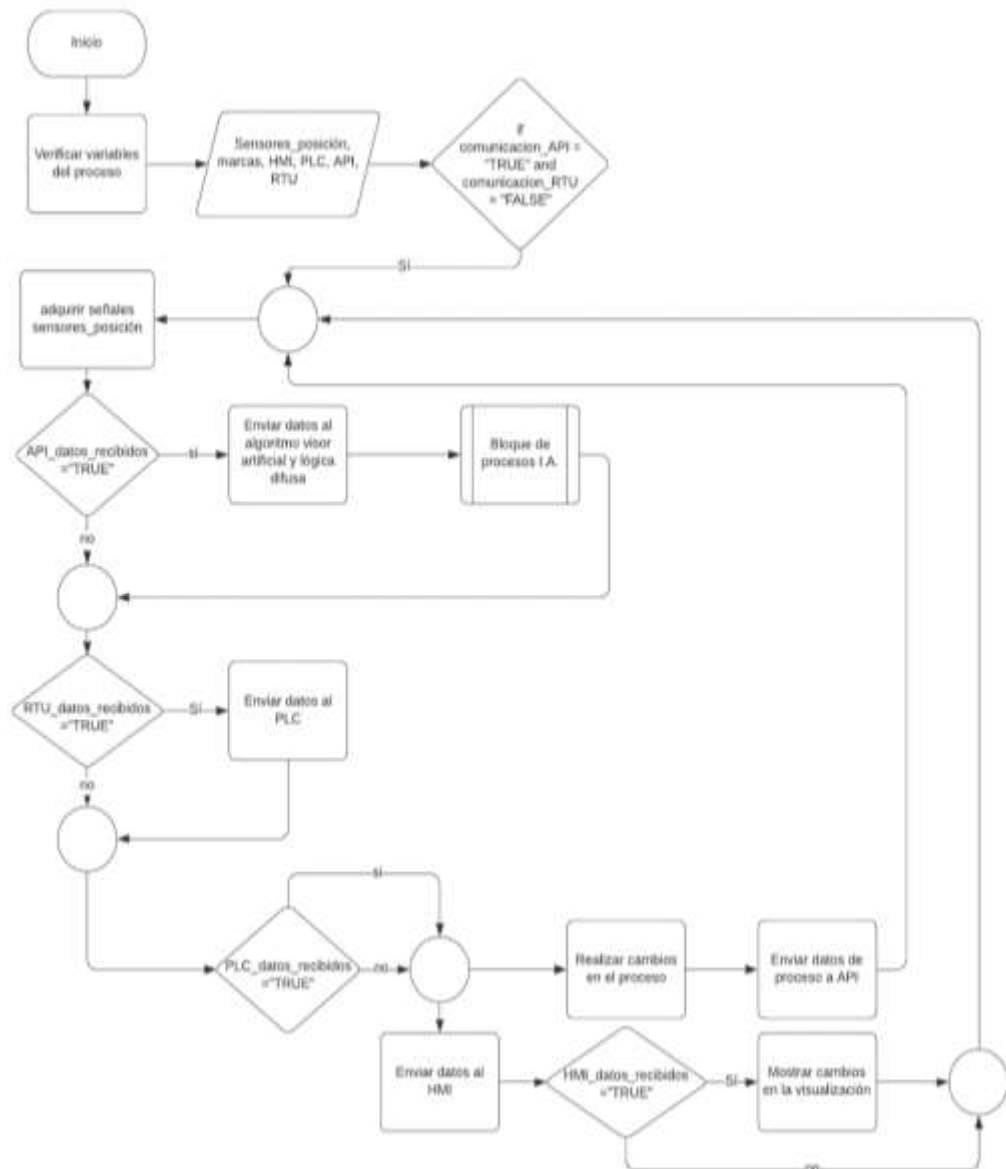


Fig. 3.12. Diagrama de flujo de las comunicaciones que existentes en el sistema

Fuente: El autor

La información básica del programa y los dispositivos vinculados se encuentran en esta ventana, la cual, dado el alcance de este trabajo, se tiene el PLC CODESYS Control Win V3 X64 en simulación, en este diagrama también están configuradas las tareas que controla el programa, la visualización de la interfaz gráfica, variables, así como la posibilidad de agregar nuevas tareas y visualizaciones a la aplicación, además, se pueden modificar los intervalos cíclicos de repetición del programa, así como la velocidad de transmisión de datos a otras plataformas de control a través de la API.



### 3.2.3. Visualización general de la planta y mediciones del proceso

En el entorno de simulación, se ha dejado un espacio en el HMI para el muestreo de las variables de posicionamiento durante la ejecución de la simulación por lo que a continuación se determinará el procedimiento de medición de la posición de las piezas del sistema SCADA a través de la programación de dichas funciones en texto estructurado.

Para la programación del algoritmo,

**Paso 1:** Después de leer la pieza que procederá a salir, se permite que las señales de entrada se dirijan al despacho en el "cerebro" o computadora / CPU / MPU de tal manera que se puedan preguntar dudas libremente.

**Paso 2:** El sistema de I.A. debe programarse para mantener al robot en la ruta correcta. Esta se puede dividir en tres partes: valores, operaciones, estructuras.

**Paso 3:** En función de las entradas recibidas, se determina si se necesita alguna modificación en la velocidad y la dirección de los robots. Entre todos los pronósticos, el robot decidirá la acción más adecuada a realizar.

**Paso 4:** Se Envían las señales de control calibradas tardías a dispositivos de velocidad.

**Paso 5:** Se dispone de un área para acumular el programa de computadora. Se lo realiza con el fin de ir almacenando las lecturas del sensor.

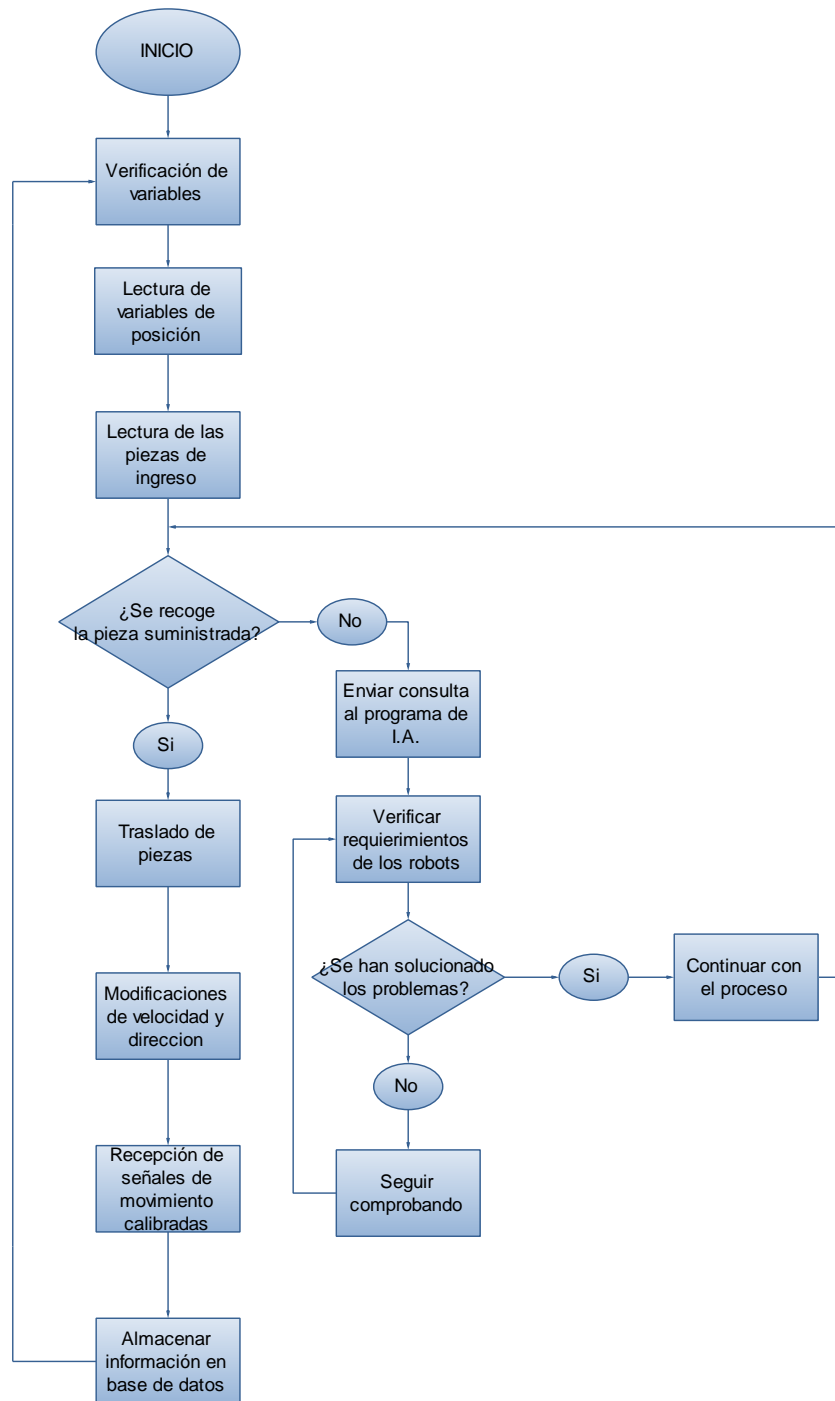


Fig. 3.13. Diagrama de flujo que representa la lista de operaciones de decisión ejecutadas en la planta

Fuente: El autor

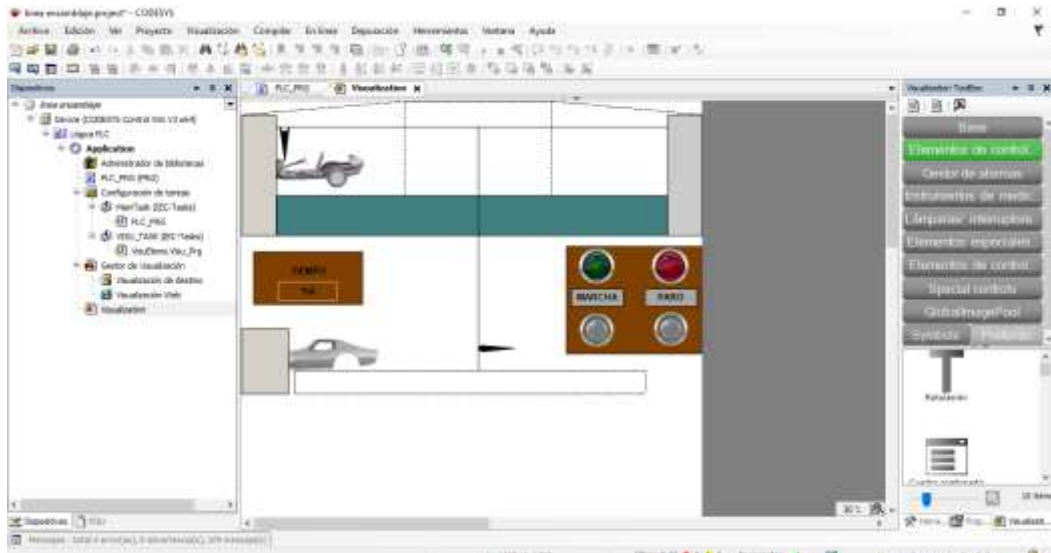


Fig. 3.14. Ventana de visualización en el programa de CODESYS para el entorno de ensamblaje vehicular

Fuente: El autor

Una vez realizados los procedimientos anteriores se procede al diseño del entorno industrial en el software de CODESYS el cual permite ingresar objetos para poder supervisar el correcto funcionamiento de la visualización del SCADA, en la imagen se puede observar la carrocería y el chasis listos para empezar su ciclo de producción, un par de brazos robóticos para poder movilizar las piezas que estén dentro de su rango alcance, cintas transportadoras las cuales realizan la entrega de las piezas a sus respectivos brazos, un botón de marcha para ejecutar el inicio de la simulación y un botón de paro para detener el proceso cuando sea requerido, además de un recuadro en el cual se ha asignado un temporizador en donde se podrá visualizar el tiempo transcurrido desde el inicio de la simulación en marcha.

### 3.2.4. Análisis de los procedimientos desarrollados previos a la simulación de la línea de ensamblaje industrial

Antes de empezar con las respectivas pruebas que se realizarán con la simulación de la línea de ensamblaje industrial, se deben realizar algunos procedimientos o pasos a seguir para la preparación de las técnicas que se utilizarán en conjunto con el sistema inteligente, es así como se inicia con la búsqueda del tipo de vehículo que se desea ensamblar, en este caso, se desea obtener un modelo de carro sencillo y con una demanda mínima de piezas de carrocería.



Fig. 3.15. Coche deportivo Charger R/T de 1970

Fuente: (Sosa, 2014)

Este tipo de vehículo deportivo, modelo modificado del Charger de 1970, contiene pocas piezas en la carrocería y su chasis está diseñado desde una perspectiva superior, es decir, que su travesaño horizontal superior era móvil y hacía las veces de vehículo convertible, además el motor sobresale del compartimento del capó dejándolo visible y expuesto a la intemperie.

Tal como se explicaba anteriormente, el chasis de ese modelo cubría solamente la parte inferior del vehículo, teniendo la apariencia de un clásico deportivo, lo cual para este proyecto cumple las condiciones requeridas para el diseño del proceso de

ensamblaje, esto es debido a la facilidad de reconocimiento visual y posterior posicionamiento de las piezas de la carrocería, y al mismo tiempo para evitar tiempos de operación prolongados en la simulación en cuanto al tiempo de montaje de cada pieza, con lo cual se reduce los períodos de trabajo para cada vehículo, a la vez que se pueden obtener una cantidad constante de datos por cada ciclo de trabajo.



Fig. 3.16. Carrocería y chasis del Charger R/T de 1970

Fuente: (Sosa, 2014)

La imagen anterior ha sido seleccionada para la simulación del proceso de ensamblaje vehicular, la carrocería de este vehículo ha sido modificada según las necesidades de la simulación, así que permite una mayor facilidad para la división y manipulación de sus piezas en el entorno industrial a supervisar, lo que permite una mayor comodidad al sistema inteligente para poder seleccionar los puntos de ensamblaje adecuados para la fabricación final del vehículo.

De esta forma se obtienen los objetos de la imagen, realizando recortes siguiendo los bordes de cada uno para adquirir piezas individuales con un recuadro oculto de ‘hitbox’ o cuadro de golpe para la carrocería, el chasis se procede a recortarlo como

una sola pieza individual debido a que las piezas serán “ensambladas” en posiciones específicas de este, por lo que necesita ser manipulado de forma independiente y en una línea de producción distinta, la cual tendrá sus propios parámetros de funcionamiento, y al mismo tiempo debe sincronizarse con los movimientos de la línea de ensamblaje para que no se produzcan errores en la localización y ubicación de las respectivas piezas.

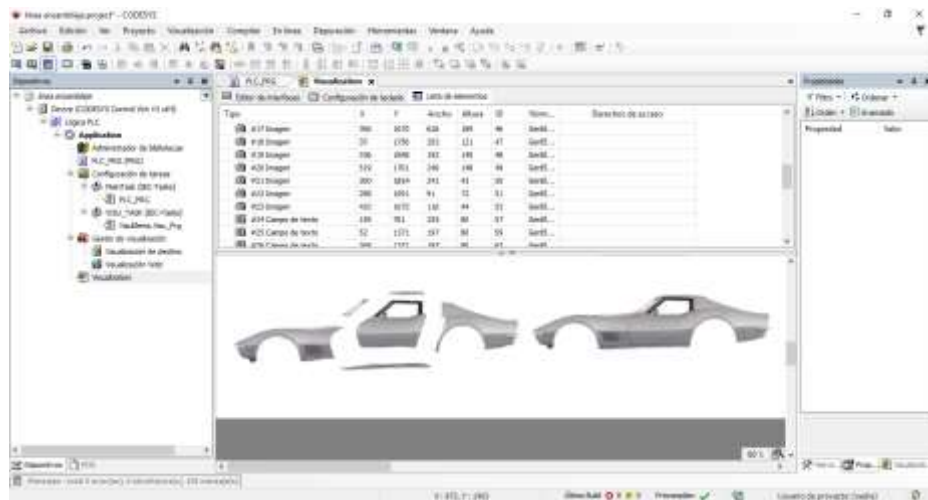


Fig. 3.17. Separación gráfica de las piezas de la carrocería y traslado al entorno de CODESYS

Fuente: El autor

Se ha realizado una comparación de la imagen de la carrocería con sus piezas ya separadas para determinar las dimensiones individuales de cada una y programar su perímetro de colisión o “hitbox”, concepto que utilizará el algoritmo de control inteligente junto con otros subprocesos para la selección, movimiento y posicionamiento de cada pieza en el lugar correspondiente del chasis del vehículo, el cual también fue insertado en el entorno de CODESYS, después de la realización de los pasos ya mencionados, se procede con la programación de los entornos de simulación correspondientes.

### 3.3. Software

A continuación, se explicarán los métodos utilizados para el diseño del sistema de supervisión inteligente, en el cual se detallan las funciones específicas que realiza el sistema SCADA y el algoritmo inteligente para el monitoreo del proceso.

En el sistema intervienen varios actores que deben realizar un determinado número de tareas para mantener el proceso en funcionamiento, estos interactúan en uno o varios niveles, a continuación, se detallan las tareas que realiza cada agente que interviene en el sistema:

**Sistema SCADA:** El sistema SCADA es el encargado del control y supervisión inteligentes de la simulación, por lo que debe enviar las variables a los siguientes subsistemas a través de la API.

**API:** Encargado de ejecutar todas las rutinas de cambio en el proceso, y de realizar la interfase con los algoritmos de “Hitbox” y lógica difusa para que estos puedan manipular el proceso según las necesidades que surjan en un evento determinado.

**Algoritmo “Hitbox”:** Este algoritmo hace las funciones de los sensores de posicionamiento óptico que se implementan físicamente, recibe las variables de posición de la API del sistema, compara dichas variables con respecto al resto de elementos que intervienen en el proceso, y detecta alguna colisión que exista entre dos o varios objetos, cuando la detección se realice, se enviará una señal al algoritmo de lógica difusa y al visor artificial, el cual aplicará el método Mandami que se describirá más adelante.

**Visor artificial:** Cuando recibe la señal de detección del algoritmo de “Hitbox” ejecuta la función de retroalimentación hacia la API del sistema, la cual inicia la alarma

de colisión del SCADA y muestra con indicadores visuales los objetos que se encuentren colisionando o superponiendo.

**Algoritmo de lógica difusa:** Encargado del control inteligente del proceso, recibe la señal de colisión del algoritmo de “Hitbox”, para luego determinar el posicionamiento adecuado en el cual se realizará el ensamblaje de la pieza que se esté trasladando en dicho momento, ejecuta el método Mandami que contiene las reglas de inferencia usadas para encontrar el punto óptimo de ensamblaje.

**Base de datos:** Se comunica con los agentes del sistema SCADA, algoritmo de lógica difusa, y algoritmo de “Hitbox” esto con el fin de almacenar las variables que se generan en dichos subprocesos, los cuales contienen las variables que intervienen en el proceso.

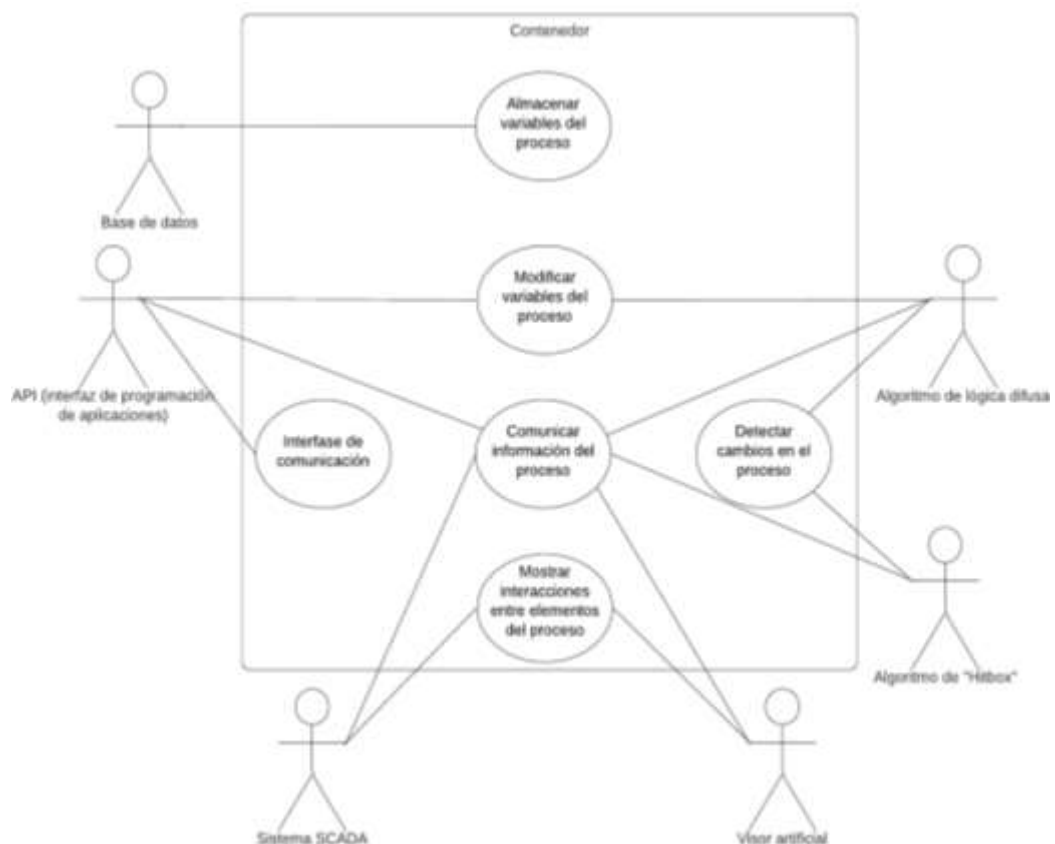


Fig. 3.18. Diagrama de casos de uso para el software del sistema  
Fuente: El autor



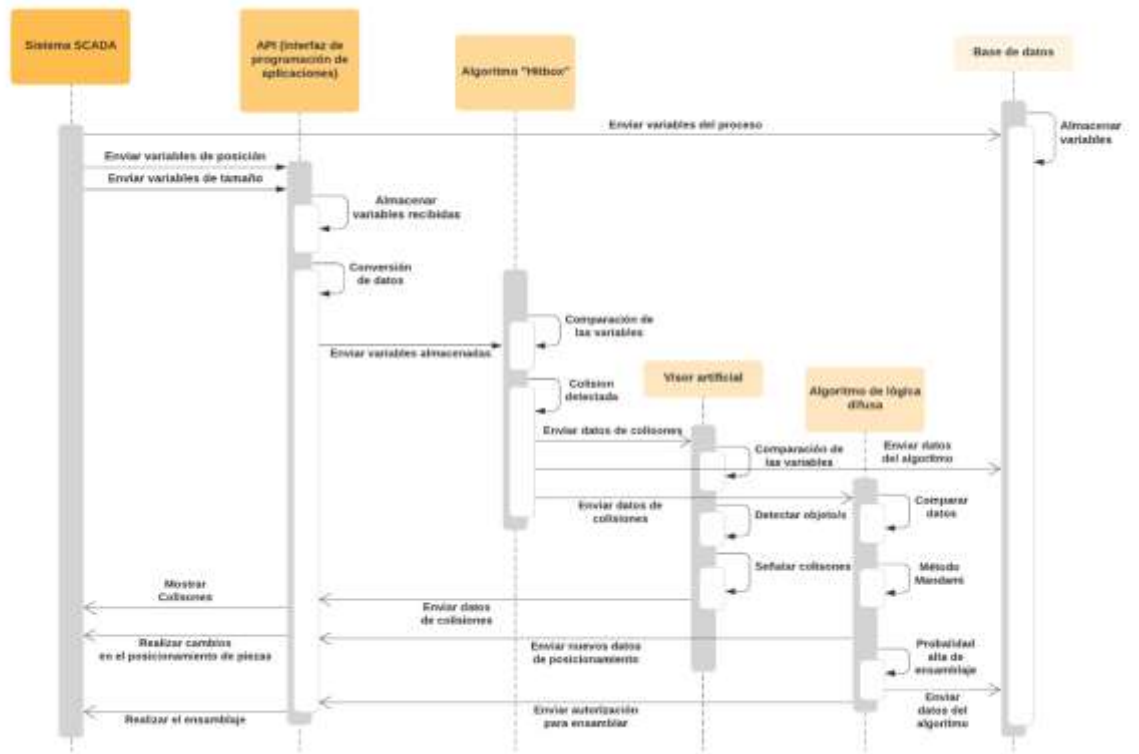


Fig. 3.19. Diagrama de secuencias para el software del sistema

Fuente: El autor

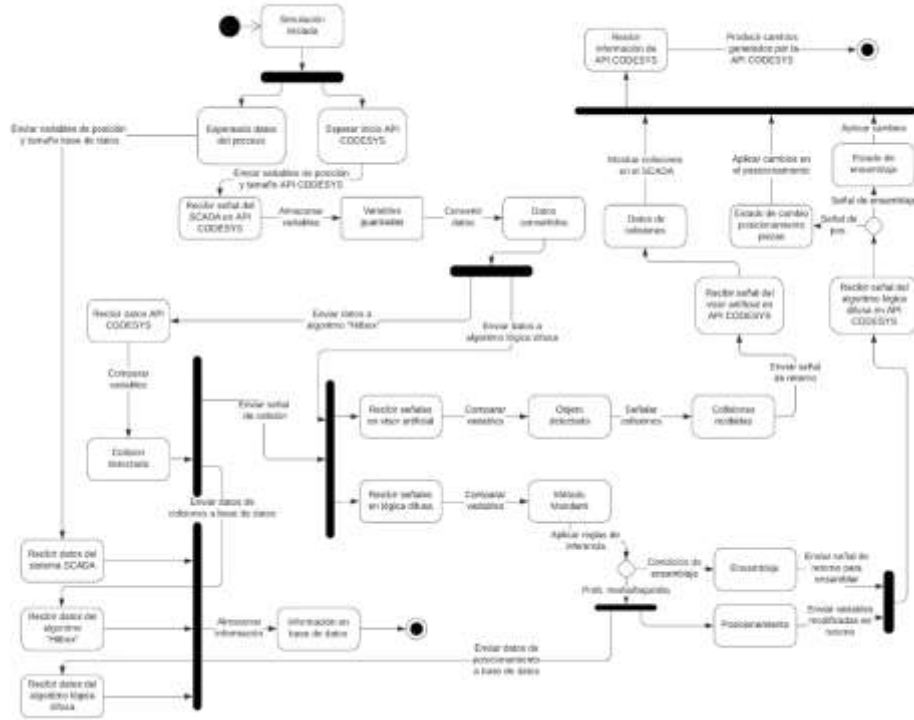


Fig. 3.20. Diagrama de estados para el software del sistema









Fuente: El autor

### 3.3.1. Diseño del algoritmo inteligente a través de la técnica de lógica difusa utilizando MATLAB

A continuación, se expresan los métodos utilizados para controlar el posicionamiento de las piezas vehiculares para el ensamblaje, los siguientes procedimientos se basan en las reglas de inferencia que se utilizan para diseñar un programa de I.A. basado en lógica difusa.











Primero se procede a justificar el uso de inteligencia artificial aplicado en el sistema SCADA; para empezar se tiene una gran cantidad de variables que se deben analizar y modificar en la ejecución del proceso, se tiene las variables de asignadas a las imágenes de cada pieza del vehículo y las variables de posición y tamaño de cada pieza que se va a controlar en la simulación, además se tienen las botoneras con sus respectivas lámparas que indican el estado de la simulación.

Tabla 3.1. Variables asignadas a las piezas del proceso

ID	Nombre de archivo	Imagen	Tipo de vínculo
carroceria	carroceria corregido.png		Vínculo con archivo
delantera	delantera corregido.png		Vínculo con archivo
esqueleto	esqueleto.png		Vínculo con archivo
estribos	estribos corregido.png		Vínculo con archivo
parabrisas	parabrisas corregido.png		Vínculo con archivo
posterior	posterior corregido.png		Vínculo con archivo
puerta	puerta corregido.png		Vínculo con archivo
superior	superior corregido.png		Vínculo con archivo

Fuente: El autor

Tabla 3.2. Variables de posición y tamaño de las piezas (visualización)

Tipo	X	Y	Ancho	Altura	ID	Nombre	Derechos de acceso
 #6 Imagen	0	200	556	199	44	GenElemInst_45	
 #17 Imagen	0	1003	626	189	46	GenElemInst_47	
 #18 Imagen	0	1058	293	121	47	GenElemInst_48	
 #19 Imagen	0	1035	192	140	48	GenElemInst_49	
 #20 Imagen	0	1029	240	148	49	GenElemInst_50	
 #21 Imagen	0	1136	241	43	50	GenElemInst_51	
 #22 Imagen	0	1103	91	72	51	GenElemInst_52	
 #23 Imagen	0	1135	116	44	53	GenElemInst_54	
 #13 Interruptor de presión	1469	917	167	147	37	GenElemInst_38	
 #14 Interruptor de presión	1791	913	167	147	39	GenElemInst_40	

● #40 Lámpara	1465	637	201	155	88	GenElemInst_88	
● #41 Lámpara	1782	636	231	159	90	GenElemInst_90	

Fuente: El autor

Adicionalmente, las líneas de código programadas en lenguaje estructurado aparecen aquí, el entorno de CODESYS permite asignar las variables que serán utilizadas en la programación en una ventana de variables, lo que permite el rápido acceso a la variable y el tipo de dato asignado.

Tabla 3.3. Variables del sistema y tamaño de cada una

<b>Variable</b>	<b>Tipo de dato</b>	<b>Tamaño</b>
contador	entero doble	32 bits
tiempo	entero doble	32 bits
cont	entero	16 bits
posx_car	entero	16 bits
posx_esq	entero	16 bits
posy_esq	entero	16 bits
posx_del	entero	16 bits
posy_del	entero	16 bits
posx_pos	entero	16 bits
posy_pos	entero	16 bits
posx_par	entero	16 bits
posy_par	entero	16 bits
posx_pue	entero	16 bits
posy_pue	entero	16 bits
posx_est	entero	16 bits
posy_est	entero	16 bits
posx_sup	entero	16 bits
posy_sup	entero	16 bits
posx_brazo1	entero	16 bits
posx_linea	entero	16 bits
posx_brazo2	entero	16 bits
posy_brazo2	entero	16 bits
led_paro	booleano	8 bits
inicio	booleano	8 bits
proceso1	booleano	8 bits
bandera	booleano	8 bits
bandera1	booleano	8 bits
bandera2	booleano	8 bits

bandera3	booleano	8 bits
bandera4	booleano	8 bits
bandera5	booleano	8 bits
bandera6	booleano	8 bits
bandera7	booleano	8 bits
inv_car	booleano	8 bits
inv_esq	booleano	8 bits
inv_del	booleano	8 bits
inv_pos	booleano	8 bits
inv_sup	booleano	8 bits
inv_est	booleano	8 bits
inv_par	booleano	8 bits
inv_pue	booleano	8 bits
marcha	booleano	8 bits
paro	booleano	8 bits

Fuente: El autor

Es así como por la gran cantidad de variables que debe manejar el sistema se utiliza el algoritmo de lógica difusa el cual puede relacionar todas las variables de posicionamiento descritas en la tabla para generar una lista de resultados dependiendo de la ubicación de la pieza y el brazo robótico que la está moviendo.

Ahora se procede a explicar el proceso que realizan los algoritmos para relacionar y analizar todas las variables de posición, tamaño, y del proceso que han sido introducidas anteriormente, para eso la API de CODESYS transmite las variables del proceso al lenguaje intérprete del algoritmo, que en este caso es Python, este recibe los datos en cuestión y procede a analizar las variables, toma las variables de posición y tamaño de las piezas y del brazo robótico para comprobar si se encuentran en contacto, esto se realiza comparando dichas variables, la colisión se detecta cuando los valores de posición y de tamaño de un objeto coinciden con el otro, es decir, se analiza la conjunción del conjunto de rangos de cada variable y su posición en el plano bidimensional.

Después envía una señal de colisión al algoritmo de visión artificial y al de lógica difusa, los cuales reciben las variables ya descritas, la función del visor es iterar en la tabla de variables de cada objeto para proceder a reconocer cuales están interactuando y realizar ciertos ajustes dependiendo el caso, y el algoritmo de lógica difusa reconocerá los elementos que pertenecen al desarrollo de la simulación, es decir las piezas que deben ser trasladadas hasta el punto donde deben ser ensambladas en la carrocería y el brazo robótico que posiciona y ensambla las piezas.

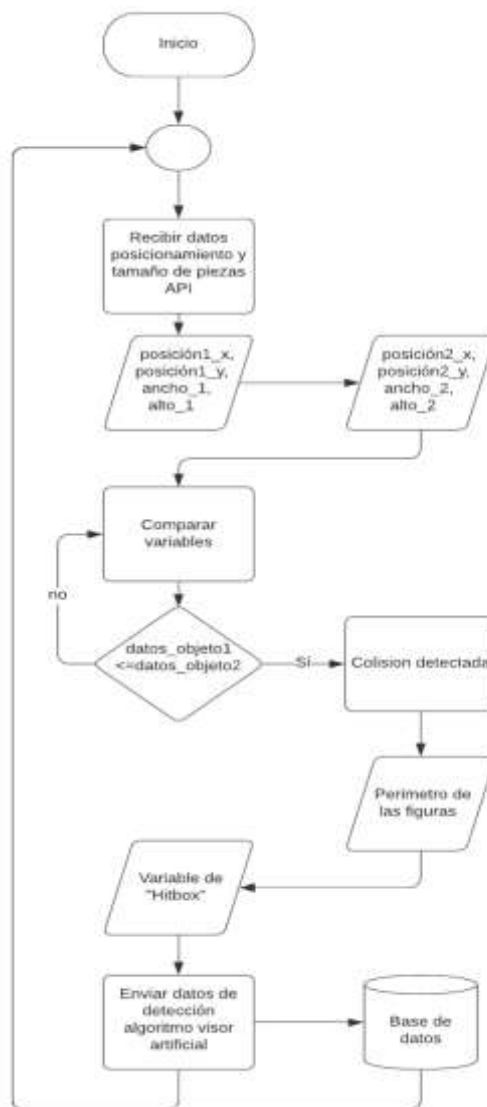


Fig. 3.21. Diagrama de flujo del algoritmo de detección de colisiones “HITBOX”

Fuente: El autor

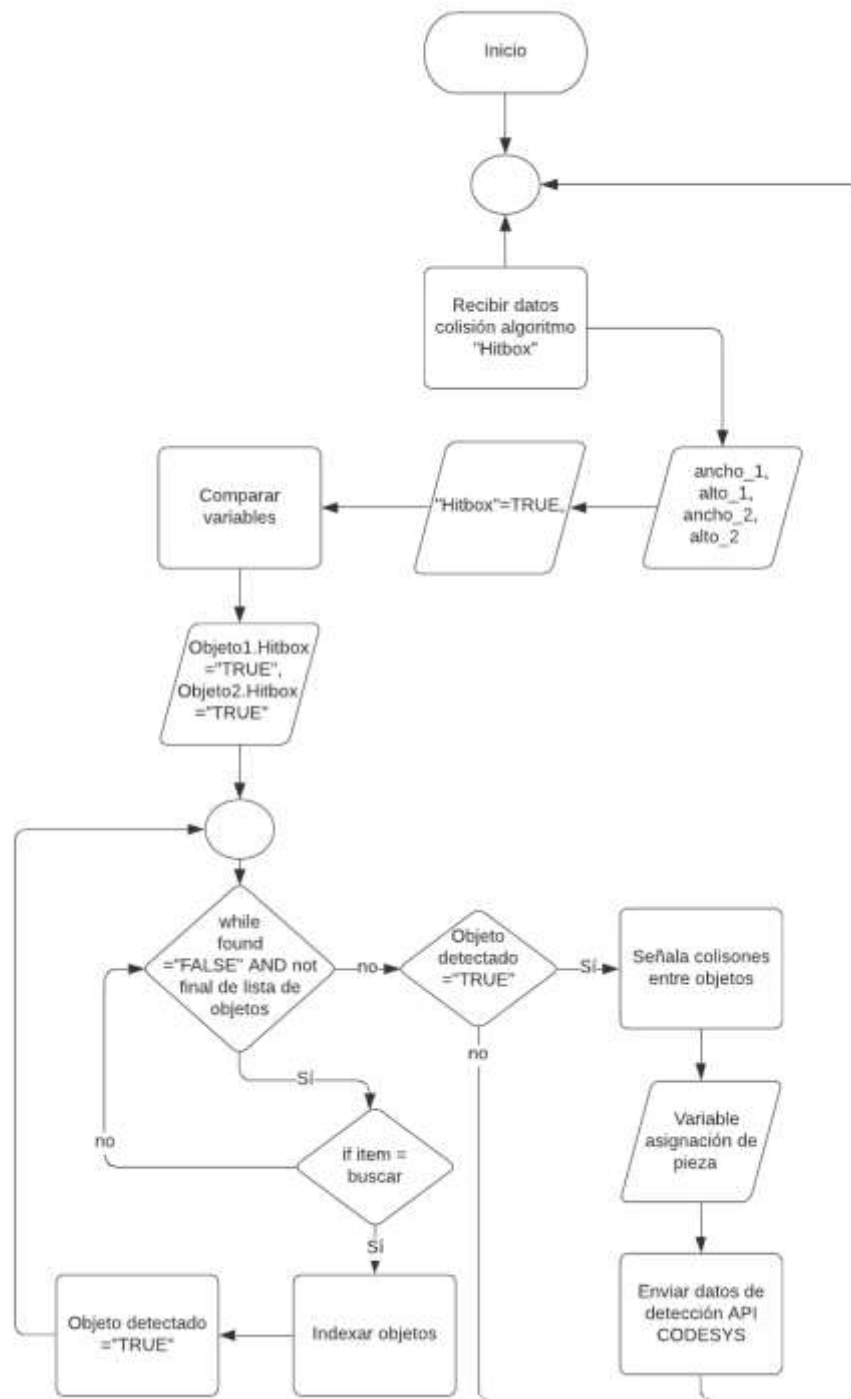


Fig. 3.22. Diagrama de flujo del algoritmo de visión artificial

Fuente: El autor

Para terminar, el procedimiento por el cual el sistema SCADA puede recibir e interpretar las sentencias de control inteligente generados por el algoritmo de lógica difusa se describen a continuación:

- Se programa la interfaz en lenguaje Python importando el módulo scriptengine de la API de CODESYS, este contiene las sintaxis, sentencias, comandos, funciones que el módulo necesita para poder interactuar con la interfase gráfica, las variables y el texto estructurado del SCADA.
- Luego el algoritmo procede a leer dichas variables y las almacena en una cadena string, para su posterior análisis, importando el módulo de lógica difusa.
- El algoritmo procede a realizar su función y generar un resultado de salida, en string, el cual nuevamente debe ser interpretado por el módulo scriptengine, generando un resultado de salida para las variables que han sido leídas al inicio de este.
- Por último la API reescribe el valor de la variable a través de comandos con texto estructurado, el cual es lenguaje estándar IEC 61131-3 para los controladores lógicos programables y sistemas SCADA.

La siguiente figura muestra una breve descripción gráfica de los pasos que el algoritmo debe seguir para realizar la interfaz entre ambas plataformas de desarrollo de aplicaciones.



Fig. 3.23. Diagrama de flujo de la implementación del algoritmo de I.A. en CODESYS

Fuente: El autor

También se ha generado un diagrama de flujo para explicar las funciones de la técnica de inteligencia artificial por lógica difusa que ha sido aplicada, en el cual se muestra un esquema breve que explica la interfaz realizada entre el sistema SCADA y el algoritmo de control inteligente para con los datos de la simulación.

El diagrama anterior muestra la estructura de decisión realizada por la inteligencia artificial, nótese que estas estructuras adquieren en primera instancia, las variables de posición de la pieza y el robot y la señal de colisión del algoritmo de “Hitbox”, para posteriormente aplicar el algoritmo de lógica difusa por el método Mandami, este representa uno de los puntos base de las estructuras inteligentes basadas en esta clase de algoritmos.



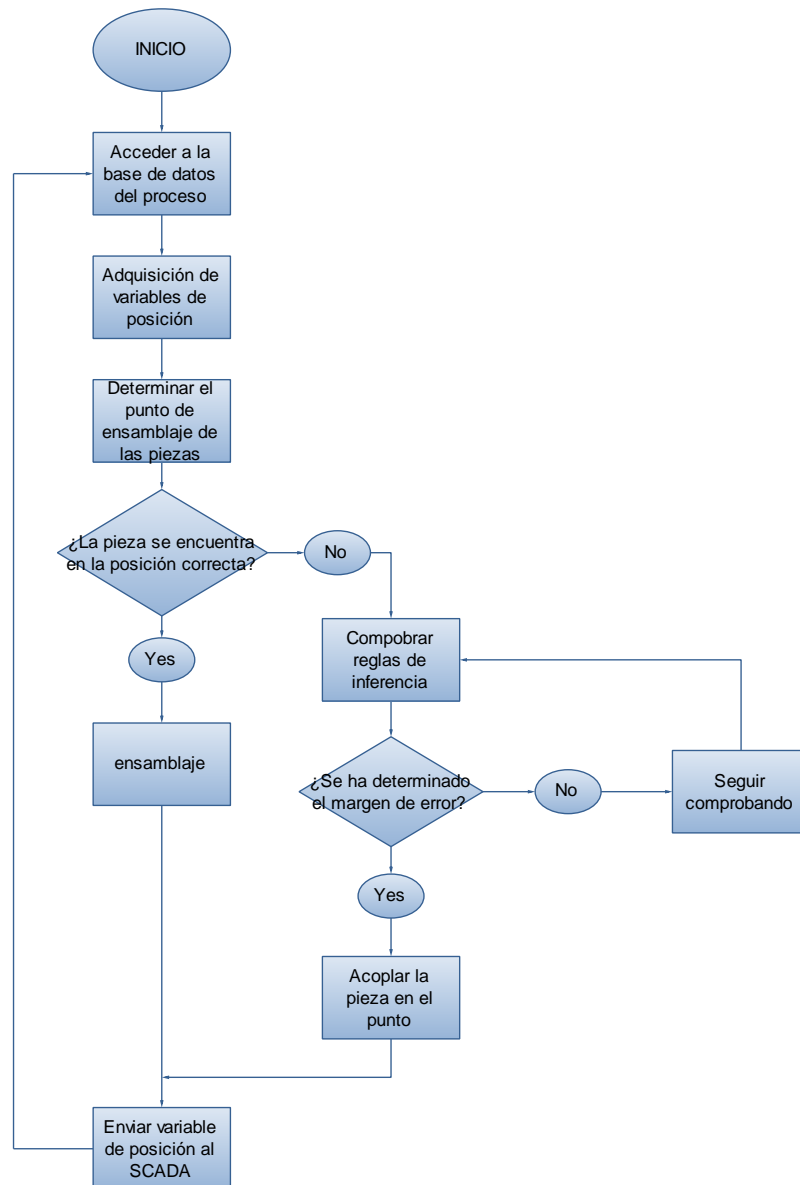


Fig. 3.24. Diagrama de flujo de la interfaz de I.A. por lógica difusa

Fuente: El autor

También se debe añadir que la estructura de decisión y el proceso de posicionamiento comienza desde el punto en el que la pieza se encuentra en el área designada para ser recogida por el primer brazo robótico que se encuentra en la cinta transportadora y termina en el ensamblaje de la pieza en la carrocería del vehículo.

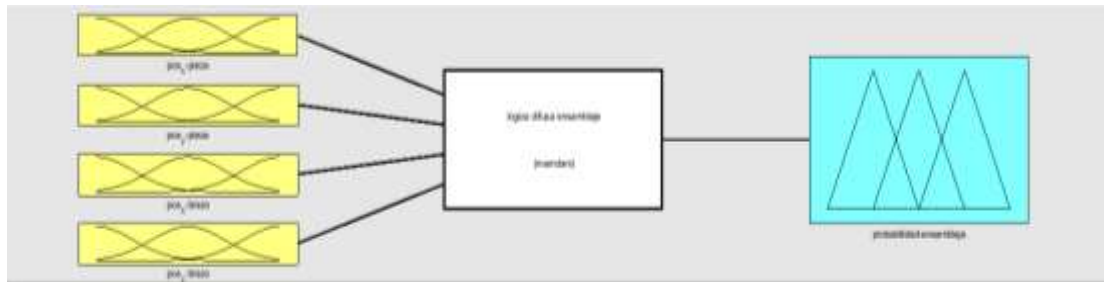


Fig. 3.25. Vista general del agente de lógica difusa para el posicionamiento

Fuente: El autor

En la figura anterior se observa el algoritmo de lógica difusa simulado en MATLAB, este define los rangos de posicionamiento que utilizará el programa para calcular el punto óptimo de ensamblaje, para eso se analizarán los métodos difusos que determinan si se ha llegado al rango aproximado de ensamblaje; dichos métodos serán explicados a continuación.

**i) Posición de la pieza del vehículo en el eje x**

Se procede a determinar los posibles estados que puede llegar a tener el objeto en el entorno de trabajo:

- Sector izquierdo: 0 píxeles – 4 píxeles
- Sector central: 3.5 píxeles – 6.5 píxeles
- Sector derecho: 6 píxeles – 10 píxeles

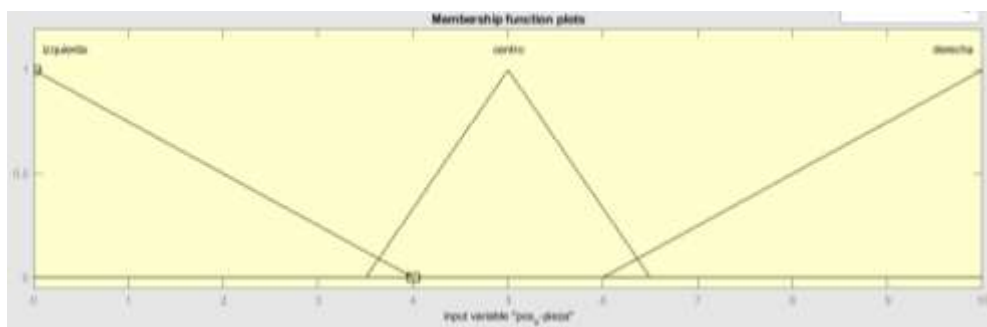


Fig. 3.26. Función de pertenencia para los rangos de posición de la pieza en el eje x

Fuente: El autor

## ii) Posición de la pieza del vehículo en el eje y

Se realiza el mismo procedimiento utilizado para determinar los rangos del eje x:

- Sector inferior: 0 píxeles – 4 píxeles
- Sector central: 3.5 píxeles – 6.5 píxeles
- Sector superior: 6 píxeles – 10 píxeles

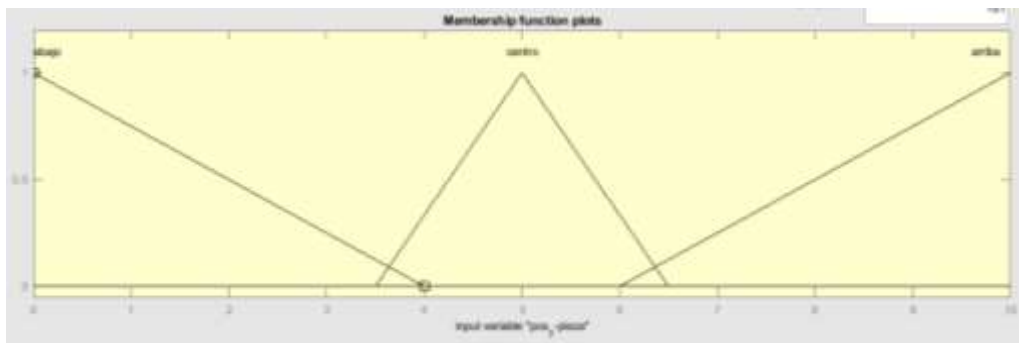


Fig. 3.27. Función de pertenencia para los rangos de posición de la pieza en el eje y

Fuente: El autor

## iii) Posición del brazo robótico en el eje x

En este caso el algoritmo debe adquirir las dimensiones de la pieza que está trasladando el brazo robótico para determinar la proporción entre el tamaño total de la pieza y el par ordenado en el que se encuentra el brazo robótico sosteniéndola, para según eso determinar el punto real en el que se encuentra.

- Sector izquierdo: 0 % – 40 % del ancho de la pieza
- Sector central: 35 % – 65 % del ancho de la pieza
- Sector derecho: 60 % – 100 % del ancho de la pieza

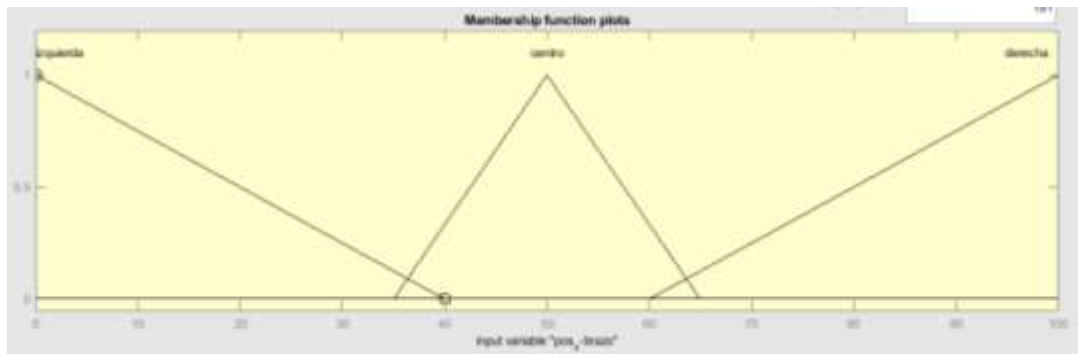


Fig. 3.28. Función de pertenencia de los rangos de posición del brazo robótico en el eje x

Fuente: El autor

#### iv) Posición del brazo robótico en el eje y

Se realiza el mismo método aplicado para el caso anterior:

- Sector inferior: 0 % – 40 % del ancho de la pieza
- Sector central: 35 % – 65 % del ancho de la pieza
- Sector superior: 60 % – 100 % del ancho de la pieza

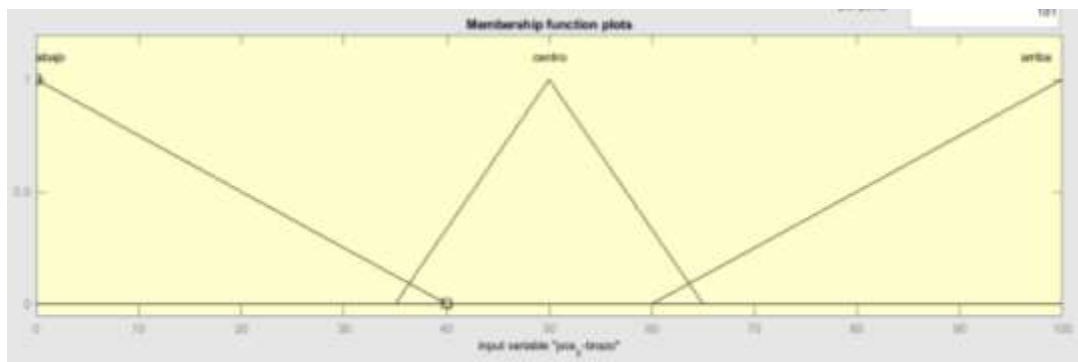


Fig. 3.29. Función de pertenencia de los rangos de posición del brazo robótico en el eje y

Fuente: El autor

#### v) Probabilidad de ensamblaje

Con la información de los rangos adquirida, ahora se procede a determinar los intervalos porcentuales para calcular la probabilidad de ensamble de las piezas. Primero el brazo robótico tiende a buscar el centro de la pieza para luego ser tomado

como nuevo punto de referencia para hallar el área correspondiente al sector de ensamblaje.

Luego se procede a analizar los resultados obtenidos en las entradas correspondientes, para eso se utiliza el método de implicación por el valor mínimo de las reglas de inferencia en el cual se aplicó el método AND (que trunca el valor de las funciones al mínimo de la regla), luego el método de agregación que consiste en sumar las funciones de pertenencia obtenidas a través de cada una de las implicaciones generadas a través del máximo de las funciones, y por último de defusifica la función obtenida para general una magnitud escalar que represente dicha función, esto se realiza con el método del centroide que obtiene la razón entre el producto de la iteración del sumatorio con los vértices de la función agregada y el sumatorio de dichos vértices.

- Probabilidad baja: 0 % – 30 %
- Probabilidad media: 30% – 60 %
- Probabilidad alta: 60 % – 90 %
- Ensamblaje: 90 % – 100 %

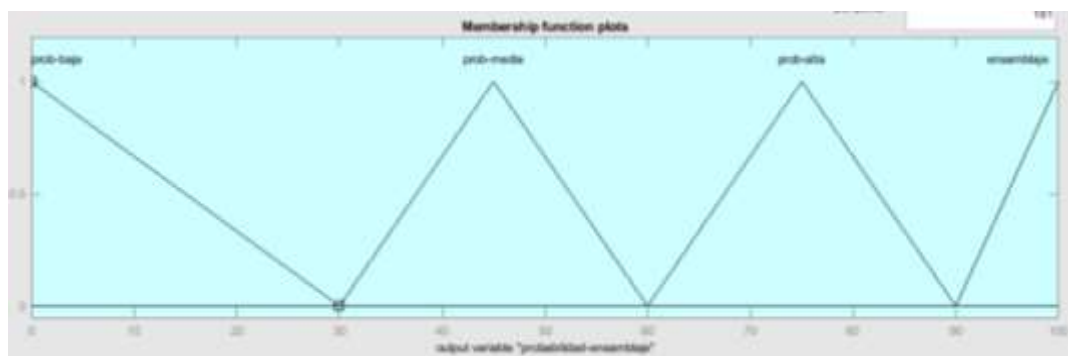


Fig. 3.30. Función de pertenencia que indica la probabilidad de ensamblaje para la salida del proceso de lógica difusa

Fuente: El autor

Por último, se establecen las reglas de inferencia que determinan la estructura de decisión para el sistema de inteligencia artificial, aquí se relacionarán las variables de posición de la pieza que se está transportando y del brazo robótico que ha recogido previamente las piezas en la cinta transportadora del entorno de visualización, y la probabilidad de ensamblaje en dependencia a la posición actual de esta con respecto al punto de acople.

Tabla 3.4. Reglas de inferencia de las posibles posiciones para el ensamblaje

PIEZA			BRAZO			Prob. Ensamblaje		
Posición x		Posición y	Posición x		Posición y			
S1. derecha	^	S2. abajo	^	S3. derecha	^	S4. abajo	→	prob baja
S1. derecha	^	S2. abajo	^	S3. derecha	^	S4. centro	→	prob baja
S1. derecha	^	S2. abajo	^	S3. derecha	^	S4. arriba	→	prob baja
S1. derecha	^	S2. abajo	^	S3. centro	^	S4. abajo	→	prob baja
S1. derecha	^	S2. abajo	^	S3. centro	^	S4. centro	→	prob media
S1. derecha	^	S2. abajo	^	S3. centro	^	S4. arriba	→	prob baja
S1. derecha	^	S2. abajo	^	S3. izquierda	^	S4. abajo	→	prob baja
S1. derecha	^	S2. abajo	^	S3. izquierda	^	S4. centro	→	prob baja
S1. derecha	^	S2. abajo	^	S3. izquierda	^	S4. arriba	→	prob baja
S1. derecha	^	S2. centro	^	S3. derecha	^	S4. abajo	→	prob baja
S1. derecha	^	S2. centro	^	S3. derecha	^	S4. centro	→	prob media
S1. derecha	^	S2. centro	^	S3. derecha	^	S4. arriba	→	prob baja
S1. derecha	^	S2. centro	^	S3. centro	^	S4. abajo	→	prob media
S1. derecha	^	S2. centro	^	S3. centro	^	S4. centro	→	prob alta
S1. derecha	^	S2. centro	^	S3. centro	^	S4. arriba	→	prob media
S1. derecha	^	S2. centro	^	S3. izquierda	^	S4. abajo	→	prob baja
S1. derecha	^	S2. centro	^	S3. izquierda	^	S4. centro	→	prob media
S1. derecha	^	S2. centro	^	S3. izquierda	^	S4. arriba	→	prob baja
S1. derecha	^	S2. arriba	^	S3. derecha	^	S4. abajo	→	prob baja
S1. derecha	^	S2. arriba	^	S3. derecha	^	S4. centro	→	prob baja
S1. derecha	^	S2. arriba	^	S3. derecha	^	S4. arriba	→	prob baja
S1. derecha	^	S2. arriba	^	S3. centro	^	S4. abajo	→	prob baja
S1. derecha	^	S2. arriba	^	S3. centro	^	S4. centro	→	prob media
S1. derecha	^	S2. arriba	^	S3. centro	^	S4. arriba	→	prob baja
S1. derecha	^	S2. arriba	^	S3. izquierda	^	S4. abajo	→	prob baja
S1. derecha	^	S2. arriba	^	S3. izquierda	^	S4. centro	→	prob baja
S1. derecha	^	S2. arriba	^	S3. izquierda	^	S4. arriba	→	prob baja
S1. centro	^	S2. abajo	^	S3. derecha	^	S4. abajo	→	prob baja
S1. centro	^	S2. abajo	^	S3. derecha	^	S4. centro	→	prob media
S1. centro	^	S2. abajo	^	S3. derecha	^	S4. arriba	→	prob baja
S1. centro	^	S2. abajo	^	S3. centro	^	S4. abajo	→	prob media

S1. centro	^	S2. abajo	^	S3. centro	^	S4. centro	→	prob alta
S1. centro	^	S2. abajo	^	S3. centro	^	S4. arriba	→	prob media
S1. centro	^	S2. abajo	^	S3. izquierda	^	S4. abajo	→	prob baja
S1. centro	^	S2. abajo	^	S3. izquierda	^	S4. centro	→	prob media
S1. centro	^	S2. abajo	^	S3. izquierda	^	S4. arriba	→	prob baja
S1. centro	^	S2. centro	^	S3. derecha	^	S4. abajo	→	prob media
S1. centro	^	S2. centro	^	S3. derecha	^	S4. centro	→	prob alta
S1. centro	^	S2. centro	^	S3. derecha	^	S4. arriba	→	prob media
S1. centro	^	S2. centro	^	S3. centro	^	S4. abajo	→	prob alta
S1. centro	^	S2. centro	^	S3. centro	^	S4. centro	→	ensamblaje
S1. centro	^	S2. centro	^	S3. centro	^	S4. arriba	→	prob alta
S1. centro	^	S2. centro	^	S3. izquierda	^	S4. abajo	→	prob media
S1. centro	^	S2. centro	^	S3. izquierda	^	S4. centro	→	prob alta
S1. centro	^	S2. centro	^	S3. izquierda	^	S4. arriba	→	prob media
S1. centro	^	S2. arriba	^	S3. derecha	^	S4. abajo	→	prob baja
S1. centro	^	S2. arriba	^	S3. derecha	^	S4. centro	→	prob media
S1. centro	^	S2. arriba	^	S3. derecha	^	S4. arriba	→	prob baja
S1. centro	^	S2. arriba	^	S3. centro	^	S4. abajo	→	prob media
S1. centro	^	S2. arriba	^	S3. centro	^	S4. centro	→	prob alta
S1. centro	^	S2. arriba	^	S3. centro	^	S4. arriba	→	prob media
S1. centro	^	S2. arriba	^	S3. izquierda	^	S4. abajo	→	prob baja
S1. centro	^	S2. arriba	^	S3. izquierda	^	S4. centro	→	prob media
S1. centro	^	S2. arriba	^	S3. izquierda	^	S4. arriba	→	prob baja
S1. izquierda	^	S2. abajo	^	S3. derecha	^	S4. abajo	→	prob baja
S1. izquierda	^	S2. abajo	^	S3. derecha	^	S4. centro	→	prob baja
S1. izquierda	^	S2. abajo	^	S3. derecha	^	S4. arriba	→	prob baja
S1. izquierda	^	S2. abajo	^	S3. centro	^	S4. abajo	→	prob baja
S1. izquierda	^	S2. abajo	^	S3. centro	^	S4. centro	→	prob media
S1. izquierda	^	S2. abajo	^	S3. centro	^	S4. arriba	→	prob baja
S1. izquierda	^	S2. abajo	^	S3. izquierda	^	S4. abajo	→	prob baja
S1. izquierda	^	S2. abajo	^	S3. izquierda	^	S4. centro	→	prob baja
S1. izquierda	^	S2. abajo	^	S3. izquierda	^	S4. arriba	→	prob baja
S1. izquierda	^	S2. centro	^	S3. derecha	^	S4. abajo	→	prob baja
S1. izquierda	^	S2. centro	^	S3. derecha	^	S4. centro	→	prob media
S1. izquierda	^	S2. centro	^	S3. derecha	^	S4. arriba	→	prob baja
S1. izquierda	^	S2. centro	^	S3. centro	^	S4. abajo	→	prob media
S1. izquierda	^	S2. centro	^	S3. centro	^	S4. centro	→	prob alta
S1. izquierda	^	S2. centro	^	S3. centro	^	S4. arriba	→	prob media
S1. izquierda	^	S2. centro	^	S3. izquierda	^	S4. abajo	→	prob baja
S1. izquierda	^	S2. centro	^	S3. izquierda	^	S4. centro	→	prob media
S1. izquierda	^	S2. centro	^	S3. izquierda	^	S4. arriba	→	prob baja
S1. izquierda	^	S2. arriba	^	S3. derecha	^	S4. abajo	→	prob baja
S1. izquierda	^	S2. arriba	^	S3. derecha	^	S4. centro	→	prob baja
S1. izquierda	^	S2. arriba	^	S3. derecha	^	S4. arriba	→	prob baja
S1. izquierda	^	S2. arriba	^	S3. centro	^	S4. abajo	→	prob baja

S1. izquierda	^	S2. arriba	^	S3. centro	^	S4. centro	→	prob media
S1. izquierda	^	S2. arriba	^	S3. centro	^	S4. arriba	→	prob baja
S1. izquierda	^	S2. arriba	^	S3. izquierda	^	S4. abajo	→	prob baja
S1. izquierda	^	S2. arriba	^	S3. izquierda	^	S4. centro	→	prob baja
S1. izquierda	^	S2. arriba	^	S3. izquierda	^	S4. arriba	→	prob baja

Fuente: El autor

### 3.4. Pruebas de prototipo

A continuación, se explicarán las pruebas realizadas con la simulación del proceso industrial, así como los elementos que intervienen en el sistema de supervisión inteligente, las variables generadas y los eventos de posicionamiento y ensamblaje que suceden en el desarrollo de este.

#### 3.4.1. Pruebas de funcionamiento del sistema de I.A. aplicada a la simulación del sistema SCADA para la red industrial a través del algoritmo de lógica difusa

La programación de las funciones primarias de la simulación del sistema SCADA fue realizada en texto estructurado (SL), uno de los lenguajes principales de programación de la norma IEC 61131-3 publicada por la comisión electrónica internacional. Sin embargo este lenguaje de programación carece de sentencias que permitan formar una estructura difusa para la toma de decisiones, visión artificial, detección de colisiones, etc. por lo que su estructura está basada en estructuras de control sencillas para aplicaciones de poca complejidad, por lo que al momento de implementar el sistema de inteligencia artificial que ha sido analizado, la API del programa permite la ejecución de scripts en segundo plano, para complementar las funciones del proceso en un sistema de control inteligente.



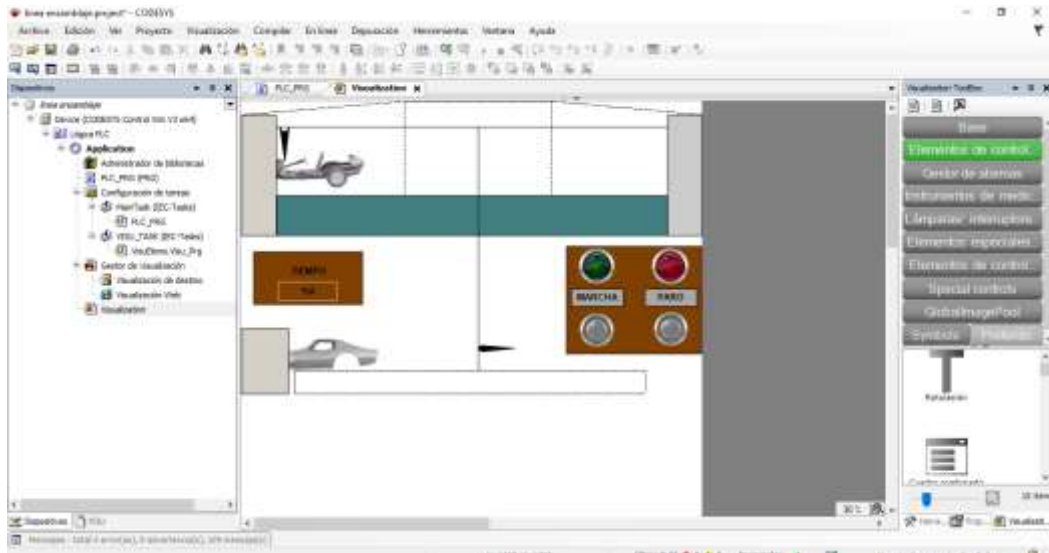


Fig. 3.31. Programa de visualización para el entorno de ensamblaje vehicular en CODESYS

Fuente: El autor

Una vez realizados los procedimientos anteriores se procede al diseño del entorno industrial en el software de CODESYS el cual permite ingresar objetos con el objetivo de visualizar la aplicación, en la imagen se puede observar la carrocería y el chasis listos para empezar su ciclo de producción, un par de brazos robóticos para poder movilizar las piezas que estén dentro del alcance de su posición, las líneas de producción las cuales realizan la entrega de las piezas a sus respectivos brazos, un botón de marcha para conmutar el inicio de la simulación y un botón de paro para detener el proceso cuando sea requerido, además de un cuadro en donde se podrá visualizar el tiempo transcurrido desde el inicio de la simulación.

El sistema se encarga de la toma de decisiones con respecto al posicionamiento de los brazos robóticos mientras trasladan y encuentran el punto de ensamblaje en el chasis del vehículo, además, se adjunta las variables relevantes del proceso de ensamblaje vehicular durante el ciclo de producción. La posición de las piezas, brazos robóticos, entre otras variables se muestran en la siguiente tabla:

Tabla 3.5. La posición de las piezas

Expresión	Tipo de datos	Valor
led_paro	BOOL	FALSE
inicio	BOOL	TRUE
proceso1	BOOL	TRUE
bandera	BOOL	FALSE
bandera1	BOOL	FALSE
bandera2	BOOL	TRUE
bandera3	BOOL	FALSE
bandera4	BOOL	FALSE
bandera5	BOOL	FALSE
bandera6	BOOL	TRUE
bandera7	BOOL	FALSE
cont	INT	1
contador	DINT	40
tiempo	DINT	201
marcha	BOOL	TRUE
paro	BOOL	FALSE
posx_car	INT	1154
posx_esq	INT	0
posy_esq	INT	0
posx_del	INT	1114
posy_del	INT	-820
posx_pos	INT	1481
posy_pos	INT	-830
posx_par	INT	1364
posy_par	INT	-905
posx_pue	INT	1347
posy_pue	INT	-831

Fuente: El autor

Esta tabla describe la posición de piezas para el ensamblaje del modelo del vehículo Charger R/T. También se utilizaron variables de tipo booleanas con el propósito de establecer estados de control, los cuales son manipulados por el programa de control inteligente por lógica difusa. Cuando el proveedor transporta las piezas a la cinta transportadora, se clasifican los componentes principales necesarios en cada proceso. Por ejemplo, el robot recoge las piezas de la cola de piezas y se dirige a la carrocería para realizar el ensamblaje. A continuación, se realizan las pruebas para comprobar la funcionalidad de los algoritmos empleados con cada estado que se genera en la salida del proceso, la probabilidad de ensamblaje, la cual se desea analizar para determinar el correcto funcionamiento del sistema. Las pruebas se realizaron conforme a los estados de salida que se generaron con las reglas de inferencia del método Mamdani, las barras muestran el posicionamiento de la pieza trasladada y el brazo robótico que la sostiene.



Fig. 3.32. Simulación en MATLAB de probabilidad baja de ensamblaje  
Fuente: El autor

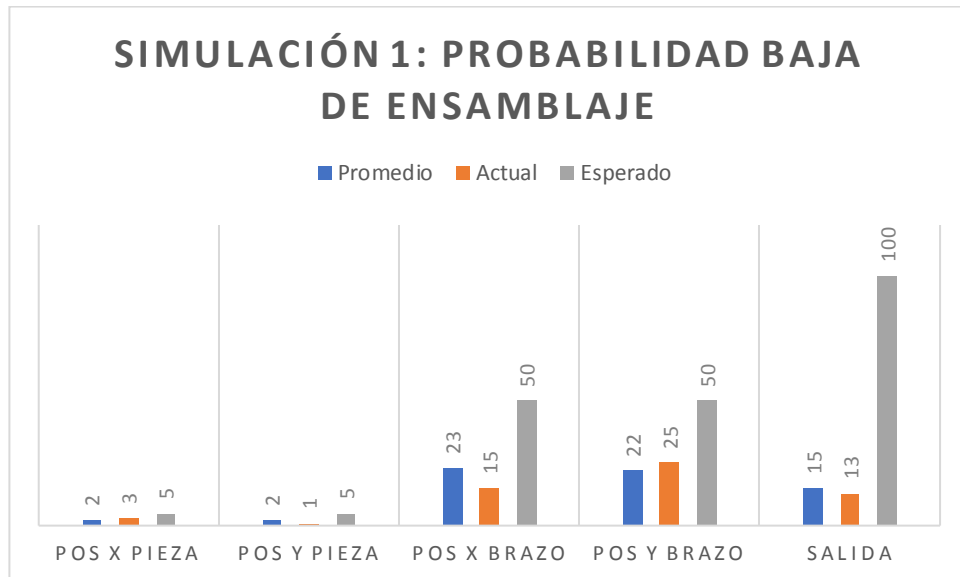


Fig. 3.33. Diagrama de barras que indica la probabilidad baja de ensamblaje  
Fuente: El autor

La barra azul representa la media de valores obtenidos de la simulación del proceso, la roja muestra los valores actuales obtenidos por el algoritmo de lógica difusa de la interfaz con el SCADA, y en gris los valores esperados para cada estado de prueba que se va a analizar; para la primera simulación, los valores de posicionamiento se encuentran alejados de los valores esperados para cada variable de entrada, por lo que la salida producida resulta en “Probabilidad baja de ensamblaje”, aquí el algoritmo de lógica difusa, empezará a mover la pieza según el valor de posición deseado que se muestra en la barra gris de la figura anterior.



Fig. 3.34. Simulación en MATLAB de probabilidad media de ensamblaje  
Fuente: El autor

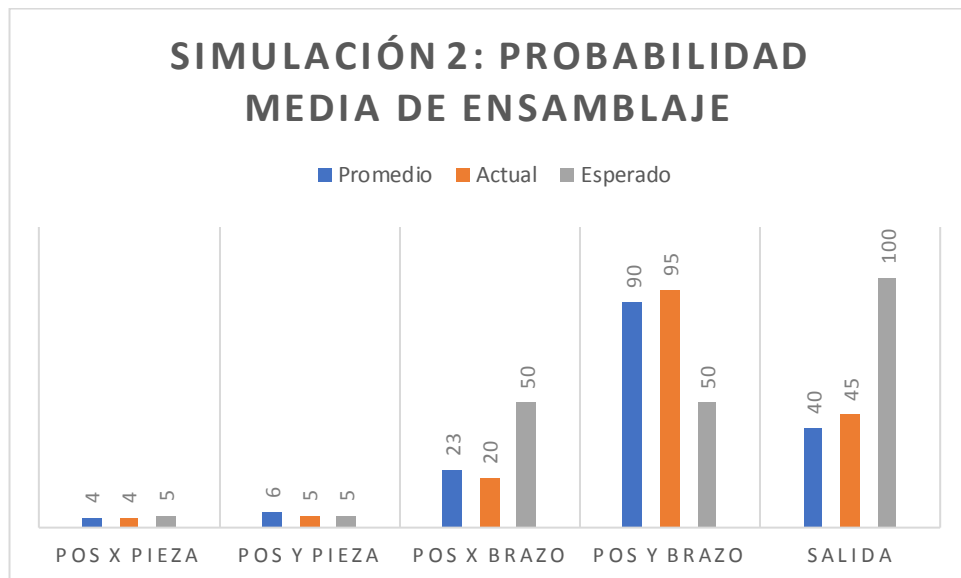


Fig. 3.35. Diagrama de barras que indica la probabilidad media de ensamblaje  
Fuente: El autor

Para la segunda simulación, las variables de posicionamiento de la pieza se encuentran en el rango esperado de posición, mientras que los valores del brazo robótico están alejados de este, por lo que la salida producida resulta en “Probabilidad media de ensamblaje”, en este caso el algoritmo debe continuar enviando valores de posicionamiento para mover la pieza hacia los valores deseados.

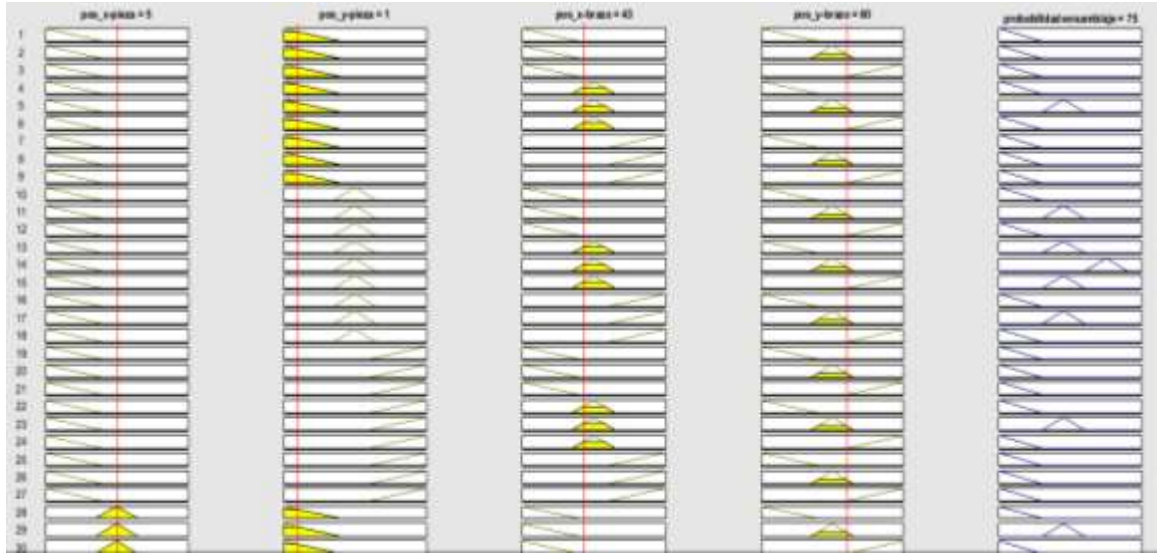


Fig. 3.36. Simulación en MATLAB de probabilidad alta de ensamblaje  
Fuente: El autor

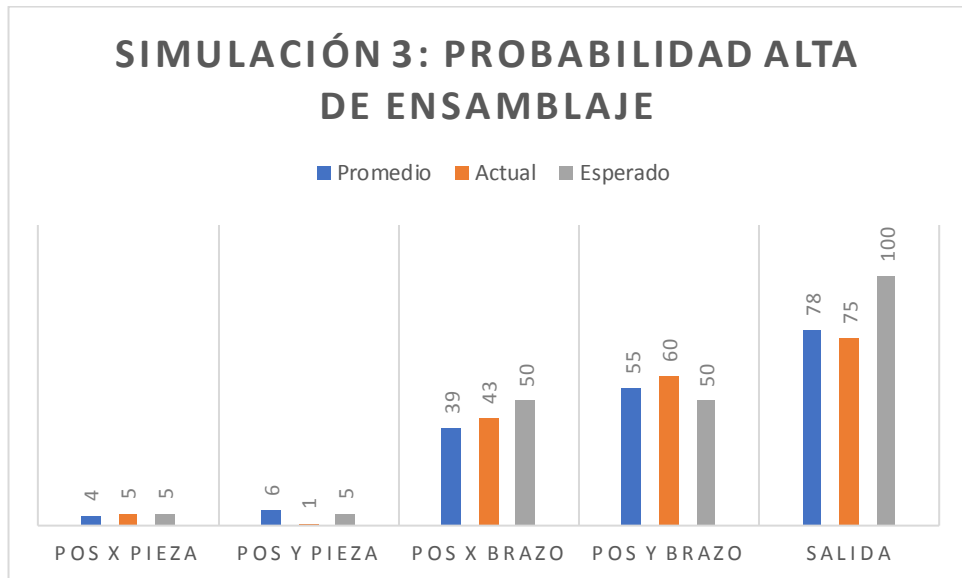


Fig. 3.37. Diagrama de barras que indica la probabilidad alta de ensamblaje  
Fuente: El autor

Para la tercera simulación, los variables de posicionamiento de la pieza y el brazo robótico se encuentran cerca del valor deseado, sin embargo, la posición y de la pieza se encuentra alejada del valor esperado, por lo que la salida producida resulta en “Probabilidad alta de ensamblaje” y el algoritmo de lógica difusa deberá acomodar el valor faltante para producir el estado que se desea.

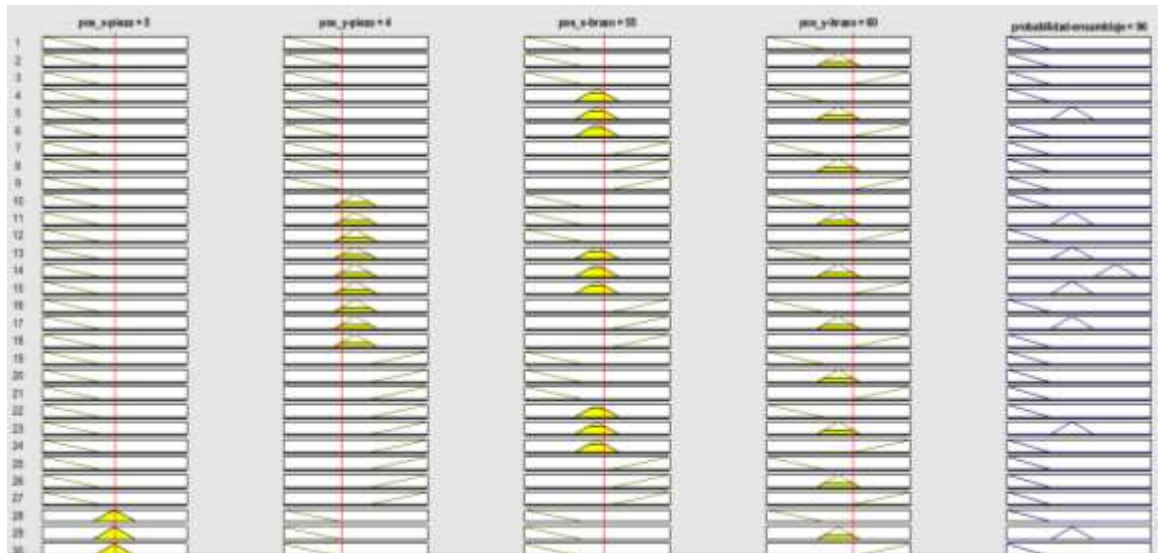


Fig. 3.38. Simulación en MATLAB de la condición de ensamblaje  
Fuente: El autor

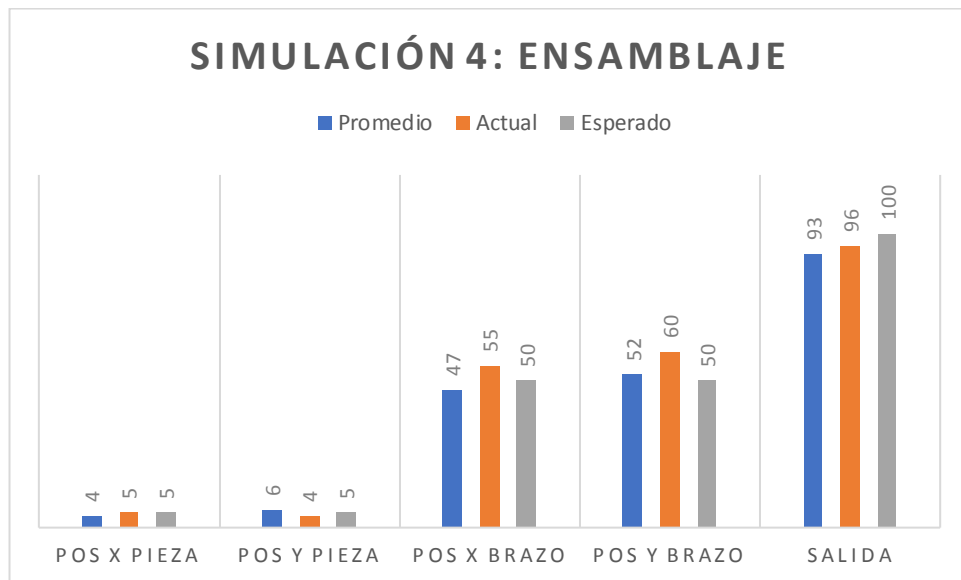


Fig. 3.39. Diagrama de barras que indica la condición de ensamblaje  
Fuente: El autor

Para la última simulación, los variables de posicionamiento de la pieza y el brazo robótico se encuentran en el rango deseado de valores, por lo que la salida generada por el algoritmo de lógica difusa resulta en “Ensamblaje”.

## **CAPÍTULO 4: PRESENTACIÓN, ANÁLISIS DE RESULTADOS Y DISCUSIÓN**

### **4.1. Presentación de resultados**

A continuación, se redactarán los resultados obtenidos de los entornos de aplicación, valores de eficiencia, tablas de los datos obtenidos de cada simulación, así como el análisis y descripción de las maniobras realizadas durante el desarrollo del trabajo de titulación.

El entorno industrial simulado se ha basado en una línea de ensamblaje vehicular, la línea de ensamblaje automovilística es una de tipo modelos mixtos (posicionamiento, visión artificial, etc.). Para satisfacer las necesidades diversificadas de los usuarios, fue necesario producir diferentes condiciones de operación de la misma línea de producción en un determinado período de tiempo. El ensamblaje es el proceso más complejo e importante en la producción de automóviles, y es el departamento de producción más exigente para los tiempos del proceso. Por lo tanto, generar modelos de posicionamiento óptico de piezas, como el algoritmo de detección de colisiones es de gran importancia para el diseño y el modelado del sistema logístico de la línea de ensamblaje de modelos mixtos.

#### **4.1.1. Resultados del software**

El porcentaje de eficiencia del sistema se determina por la razón entre la respuesta del sistema en la salida y la proporción del valor de entrada (función de pertenencia), es decir, se desea determinar cuál es el margen de error de la aplicación al realizar los controles inteligentes en condiciones establecidas por el usuario o programador,

entonces si la razón entre valor de la entrada y el valor de salida es unitaria, se tiene una eficiencia del 97%.

En la evaluación final del sistema se puede denotar en la gráfica entrada-salida, la relación entre los valores de posicionamiento en la entrada del proceso industrial que ha sido simulado como variable independiente, y la salida o respuesta como la variable dependiente, cabe destacar que la superficie de la función  $f(x)$  representa el valor real del proceso que fue simulado, y los puntos que están repartidos alrededor de la superficie representan la respuesta de la aplicación de la I.A. escogida para el control y supervisión del proceso, que en este caso fue utilizada la técnica de lógica difusa, como “operador artificial” en la toma de decisiones para el desarrollo adecuado de la simulación del proceso de tipo línea de ensamblaje.

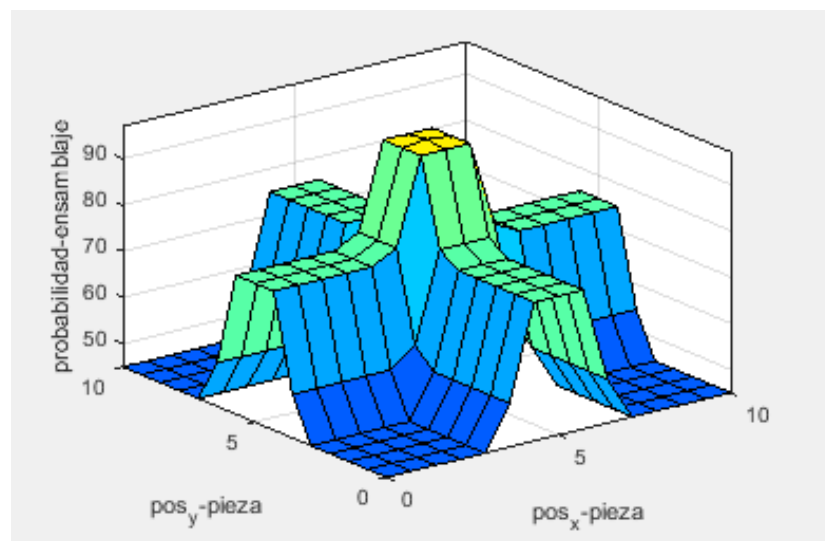


Fig. 4.1. Simulación en MATLAB en la que se obtiene la superficie generada por las reglas de inferencia del algoritmo de lógica difusa

Fuente: El autor

La figura 4.1 describe la superficie generada por las reglas de inferencia que rigen el algoritmo de lógica difusa, relacionando las variables de posición de la pieza del vehículo y del brazo robótico que debe ensamblar dicha pieza junto con la función de pertenencia asignada a la probabilidad de ensamblaje en la carrocería, los sectores



azulas representan la probabilidad baja de ensamble, debido a la lejanía del punto óptimo para este evento, los verdes la probabilidad media y alta ya que se acercan más al centro, y el sector amarillo, que representa la posible área donde se procederá a ensamblar la pieza correspondiente.

Así se realizó la estimación de los resultados con porcentajes de eficiencia, los cuales determinaron el proceso de aprendizaje de la I.A. diseñada, para la posterior comparación con los resultados del estado del arte que arrojaron efectividades de hasta un 97% por cada evento acontecido en las simulaciones de los que han sido realizadas.

#### **4.1.2. Resultados de pruebas de prototipo**

Los resultados de las pruebas de simulación arrojaron los siguientes resultados:

- Se realizaron 4 pruebas en total tomando en cuenta el número de estados que había sido establecido en el algoritmo de lógica difusa.
- El algoritmo de lógica difusa generó 81 reglas de inferencia tomando en cuenta el posicionamiento de la pieza y el brazo robótico que la traslada.
- El sistema tiende a buscar el punto más cercano al área bajo la curva del ensamblaje.
- La simulación permitió disponer de los valores de posición y realizar un promedio para cada caso de salida que surge de aplicar las reglas de inferencia al sistema de posicionamiento.
- Para este punto el sistema disponiendo sus parámetros de operación está por alcanzar a los valores promedios de la gráfica para la función de pertenencia de salida expresada en puntos porcentuales.

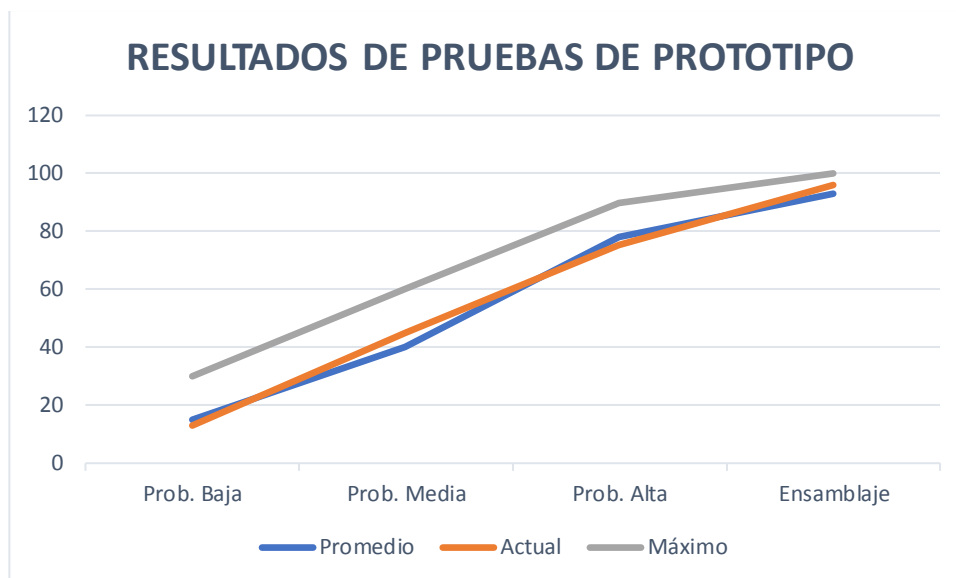


Fig. 4.2. Gráfica que muestra la salida del proceso de cada prueba de prototipo en relación con la función de pertenencia

Fuente: El autor

## 4.2. Discusión de resultados

En esta sección se analizarán los resultados obtenidos, y se evaluarán con referencia a los criterios de desarrollo de este trabajo de titulación, con la intención de dejar un punto de partida para posteriores proyectos de supervisión inteligente con la aplicación de las técnicas aquí utilizadas que han cumplido con los objetivos propuestos, la metodología, que asevere los métodos utilizados en un aspecto técnico y científico.

### 4.2.1. Cumplimiento de objetivos

En el siguiente apartado, se analizarán los resultados obtenidos en relación con el cumplimiento de los objetivos propuestos al inicio de este trabajo, los cuales determinaron las actividades que se realizaron a lo largo de este proyecto

#### i) Objetivo general

Se consiguió el desarrollo de un algoritmo basado en lógica difusa aplicado en un sistema SCADA y a su vez la supervisión y ensamblaje en una línea de producción

vehicular, la aplicación de la tecnología de inteligencia artificial trajo consigo una mejora a largo plazo en el control y supervisión inteligentes de la producción dentro del proceso descrito.

## **ii) Objetivos específicos**

Se establecieron los modelos matemáticos para el ingreso de las variables del entorno industrial de tipo línea de ensamble, los cuales permitieron simular de forma satisfactoria, los parámetros utilizados para la fabricación de productos en una línea de producción continua; sin embargo, los modelos fueron adquiridos de procesos de ensamble que cumplen con normas internacionales, por lo que, comparándolo con el desempeño de la industria a nivel nacional, se dejan muchos puntos abiertos o parámetros sin utilizar, esto es insatisfactorio si se desea llegar a un punto de equilibrio, sin condiciones de sobra, que solamente sobrecarguen información al sistema.

### **4.2.2. Análisis de la metodología**

La investigación es de tipo correlacional, por lo que durante la realización del trabajo se tesis se encontraron diversas variables que rigen el modelado de la operación del proceso industrial de ensamble vehicular y en la posterior simulación del sistema de supervisión inteligente, estas variables tendrán una relación que debe ser guardada en una base de datos, para ser después analizadas, y posteriormente descritas en una tabla de resultados, la cual definirá el porcentaje de eficiencia de la simulación.

### **4.2.3. Análisis de la solución al problema de investigación**

¿Puede la aplicación de inteligencia artificial por lógica difusa optimizar la supervisión del proceso de línea de ensamble vehicular? Los resultados del sistema de supervisión demostraron que se ha realizado el monitoreo inteligente de la

simulación de la línea de ensamblaje, en este monitoreo se observó que el posicionamiento de las piezas se volvía cada vez más preciso debido a los algoritmos de lógica difusa, “Hitbox” o detección de colisiones y el visor artificial utilizados por el programa inteligente de lógica difusa.

#### **4.2.4. Discusión final**

Las operaciones de ensamblaje de automóviles y los fabricantes de partes automotrices son algunos de los mayores usuarios de robots en la industria de fabricación de automóviles. Los robots son más fáciles de programar y desplegar que nunca, pero cada proyecto de integración viene con desafíos únicos. Es por eso por lo que los fabricantes interesados en adoptar robótica automotriz deben trabajar con algoritmos de integración adecuados para la adquisición de datos, supervisión y control inteligentes de la línea de producción en sus plantas industriales.

Las partes principales del proceso se dividen en el grupo de componentes después de la primera clasificación y se ponen en el almacenamiento de proceso. Las piezas de clase general con gran circulación se ponen en almacenamiento centralizado. Una estrategia de almacenamiento grupal de este tipo admite el uso de bastidores de almacenamiento de componentes especiales, Por un lado, puede mejorar el estado del suministro de piezas de repuesto y, por otro lado, puede posicionar las piezas en el área de trabajo de forma ordenada y precisa.

## CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES

### 5.1. Conclusiones

Se ha desarrollado un algoritmo basado en lógica difusa aplicado en un sistema SCADA, para supervisar el posicionamiento y ensamblaje en una línea de producción vehicular. A través de la simulación del proceso industrial y la posterior aplicación de la técnica de inteligencia artificial por lógica difusa se determinó un 97% de eficiencia según los valores mostrados en los resultados, gracias a esto se asegura la producción y calidad del producto final, además, si se siguen aplicando paquetes de mejoras a los algoritmos se puede lograr resultados mucho más precisos en términos de posicionamiento y ensamblaje.

Los algoritmos de detección de colisiones, visión artificial y lógica difusa fueron seleccionados y posteriormente implementados para desarrollar un sistema completo de control y supervisión inteligentes, todos estos algoritmos fueron programados en lenguaje Python

La ejecución de las pruebas correspondientes con el software de CODESYS para el sistema SCADA y posteriormente Python para el control inteligente del sistema se realizaron a través de la comparación de las variables que se generaron en la simulación, por ende, la aplicación ha demostrado las capacidades necesarias para realizar un control y supervisión inteligentes del entorno, y a su vez ajustarse a la gráfica de superficie que representa la función de pertenencia de la salida del proceso.

Los resultados de las simulaciones permitieron observar el funcionamiento de la técnica de control inteligente, de esta forma, se logró determinar la relación entre el posicionamiento de las piezas junto al brazo robótico y la probabilidad de ensamblaje

según las condiciones de entrada para el algoritmo desarrollado de lógica difusa como un sistema de adecuado para el control y supervisión del entorno industrial en interfaz con el sistema SCADA, esta técnica, demostró tener ventajas a nivel de costo y facilidad de programación.

El lenguaje estructurado (ST) ofreció muchas prestaciones a la hora de realizar la programación de la visualización del SCADA, asimismo, la generación de marcas gracias a este lenguaje de programación de PLC's y SCADA's permitió facilitar la comunicación entre ambas plataformas de desarrollo, permitiendo la lectura y escritura rápida del cambio generado en las variables que se requieren en la simulación del proceso.

## **5.2. Recomendaciones**

Se debe tomar en cuenta el tiempo para el desarrollo y simulación de este tipo de sistemas multiplataforma, debido a que las aplicaciones con inteligencia artificial, utilizando cualquier técnica, o método de operación, valga la redundancia, suelen ser complejos y necesitan una comprensión teórica y técnica avanzadas de tema, así como un buen nivel de programación, además de una extensa cantidad de tiempo para la preparación de este tipo de sistemas.

Los algoritmos de inteligencia artificial por lógica difusa, detección de colisiones "Hitbox" y el visor artificial se basan en modelos matemáticos avanzados como lo es el método Mandami, por lo que se recomienda, tener conocimientos intermedios-avanzados, acerca de conceptos como retro propagación, implicaciones S y R de Lukasiewicz, métodos de agregación, defusificación entre otras que sirven de base para empezar a desarrollar algoritmos de este tipo.

Se debe tomar en cuenta la necesidad de tener una base de datos para que las plataformas de desarrollo de CODESYS y Python puedan respaldar su información, así como solicitarla en todo momento, siendo este requisito importante para el correcto funcionamiento de la I.A.

Una vez realizados los pasos anteriores, se pueden utilizar los algoritmos de control inteligentes previamente implementados en el sistema SCADA, para el control, supervisión, toma de decisiones con respecto a la evolución de la simulación del proceso, es así como, el algoritmo de I.A. por lógica difusa permite obtener los valores correspondientes relacionados con la función de pertenencia que representa la salida del proceso a evaluar.

### **5.3. Trabajo futuro**

Para trabajos futuros, se plantea comenzar con la implementación de un servidor OPC en el cual el sistema SCADA desarrollado en CODESYS y el script programado en lenguaje Python para el algoritmo de lógica difusa, puedan transmitir y recibir las iteraciones producidas en las variables de la interfaz de usuario debido a los cambios en la simulación del proceso y del control inteligente del script de Python, de este modo, se puede realizar la lectura y escritura de las variables en tiempo real con las ventajas de computación en paralelo que ofrecen los servidores y así mejorar el rendimiento del algoritmo de I.A. disminuyendo los retrasos por tiempo de ejecución provenientes del hardware empleado, a su vez, se puede integrar técnicas de control y supervisión inteligente por redes neuronales siempre y cuando se disponga de estos servidores por la amplia generación de datos en el tiempo de ejecución de la simulación.

## Referencias Bibliográficas

- R. Al Badwawi, W. R. Issa, T. K. Mallick and M. Abusara, "Supervisory Control for Power Management of an Islanded AC Microgrid Using a Frequency Signalling-Based Fuzzy Logic Controller," in *IEEE Transactions on Sustainable Energy*, vol. 10, no. 1, pp. 94-104, Jan. 2019. doi: 10.1109/TSTE.2018.2825655
- Rani, P. H., Navasree, S., George, S., & Ashok, S. (2019). Fuzzy logic supervisory controller for multi-input non-isolated DC to DC converter connected to DC grid. *International Journal of Electrical Power & Energy Systems*, 112, 49-60.
- Velayudhan, A. K. D. (2019). Design of a supervisory fuzzy logic controller for monitoring the inflow and purging of gas through lift bags for a safe and viable salvaging operation. *Ocean Engineering*, 171, 193-201.
- Meghni, B., Dib, D., Azar, A. T., & Saadoun, A. (2018). Effective supervisory controller to extend optimal energy management in hybrid wind turbine under energy and reliability constraints. *International Journal of Dynamics and Control*, 6(1), 369-383.
- Robles, A., Capson-Tojo, G., Ruano, M. V., Latrille, E., & Steyer, J. P. (2018). Development and pilot-scale validation of a fuzzy-logic control system for optimization of methane production in fixed-bed reactors. *Journal of Process Control*, 68, 96-104.
- Sarrias-Mena, R., Fernández-Ramírez, L. M., García-Vázquez, C. A., & Jurado, F. (2019). Design of a Supervisory Control System Based on Fuzzy Logic for a Hybrid System Comprising Wind Power, Battery and Ultracapacitor Energy Storage System. In *Advanced Control and Optimization Paradigms for Wind Energy Systems* (pp. 189-212). Springer, Singapore.



- Inteligencia Artificial, clave en la industria automotriz. Ghosh, Subhanjan (17 de abril de 2018). Recuperado de <https://www.prensariotila.com/22692--Inteligencia-Artificial-clave-en-la-industria-automotriz.note.aspx>
- Bangalore, P., & Tjernberg, L. B. (2015). An Artificial Neural Network Approach for Early Fault Detection of Gearbox Bearings. *IEEE Transactions on Smart Grid*, 6(2), 980-987. <https://doi.org/10.1109/TSG.2014.2386305>
- Abbas, H.A. and Mohamed, A.M. (2011) 'Review in the design of web based SCADA systems based on OPC DA protocol', *International Journal of Computer Networks*, February, Vol. 2, No. 6, pp.266–277, Malaysia.
- Mamdani, E. H., & Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, 7(1), 1-13. [https://doi.org/10.1016/S0020-7373\(75\)80002-2](https://doi.org/10.1016/S0020-7373(75)80002-2)
- Sanap, S., Nawale, R., Kapse, S., Kale, A., & Korade, M. (2015). Exact virtualization of industrial environment on web using SCADA with artificial intelligence. En *2015 International Conference on Green Computing and Internet of Things (ICGCIoT)* (pp. 99-103). <https://doi.org/10.1109/ICGCIoT.2015.7380437>
- Kumar, P., Chandna, V. K., & Thomas, M. S. (2004). Fuzzy-genetic algorithm for pre-processing data at the RTU. *IEEE Transactions on Power Systems*, 19(2), 718-723. <https://doi.org/10.1109/TPWRS.2004.825924>
- Lee, R. C. T. (1993). Fuzzy Logic and the Resolution Principle. En D. Dubois, H. Prade, & R. R. Yager (Eds.), *Readings in Fuzzy Sets for Intelligent Systems* (pp. 442-452). Morgan Kaufmann. Recuperado de <http://www.sciencedirect.com/science/article/pii/B9781483214504500493>

- Horng, J.-H. (2008). Hybrid MATLAB and LabVIEW with neural network to implement a SCADA system of AC servo motor. *Advances in Engineering Software*, 39(3), 149-155. <https://doi.org/10.1016/j.advensoft.2007.02.006>
- Liu, and K. (2007). Implementation of a Web-Based Real-Time Monitoring and Control System for a Hybrid Wind-PV-Battery Renewable Energy System. En *2007 International Conference on Intelligent Systems Applications to Power Systems* (pp. 1-6). <https://doi.org/10.1109/ISAP.2007.4441615>
- Zaher, A., McArthur, S. D. J., Infield, D. G., & Patel, Y. (2009). Online wind turbine fault detection through automated SCADA data analysis. *Wind Energy*, 12(6), 574-593. <https://doi.org/10.1002/we.319>
- Salgado, A. R., Esquivel, C. R. F., & Guizar, J. G. C. (2015). SCADA and PMU Measurements for Improving Power System State Estimation. *IEEE Latin America Transactions*, 13(7), 2245-2251. <https://doi.org/10.1109/TLA.2015.7273784>
- Horng, J.-H. (2002). SCADA system of DC motor with implementation of fuzzy logic controller on neural network. *Advances in Engineering Software*, 33(6), 361-364. [https://doi.org/10.1016/S0965-9978\(02\)00020-0](https://doi.org/10.1016/S0965-9978(02)00020-0)
- Bravo, M., Machado, H., García, Z., Álvarez, A., Bonet, I., & Cabrera, N. E. (2006). Sistema de diagnóstico para la estimación de secciones en fallo en sistemas eléctricos de potencia. *Ingeniería Energética*, XXVII (2-3). Recuperado de <http://www.redalyc.org/resumen.oa?id=329127752008>
- Novosel, D., & King, R. L. (1994). Using artificial neural networks for load shedding to alleviate overloaded lines. *IEEE Transactions on Power Delivery*, 9(1), 425-433. <https://doi.org/10.1109/61.277714>
- Chen, B., Qiu, Y. N., Feng, Y., Tavner, P. J., & Song, W. W. (2011). Wind turbine SCADA alarm pattern recognition, 163-163. <https://doi.org/10.1049/cp.2011.0164>

- Pérez, F. (2017, diciembre 15). Sistemas SCADA. Recuperado de <http://www.auracen.com/sistemas-scada/>
- Kaplan Andreas; Michael Haenlein (2018) Siri, Siri in my Hand, who's the Fairest in the Land? On the Interpretations, Illustrations and Implications of Artificial Intelligence, Business Horizons
- Maloof, Mark. (2017) "Artificial Intelligence: An Introduction, p. 37". Georgetown University
- Li, D., & Du, Y. (2017). Artificial intelligence with uncertainty. CRC press.
- Bulfinch, Thomas - Stories of gods and heroes (1920) by Thomas Bulfinch with illustrations drawn in color by Sybil Tawse
- Christian, Brian; Griffiths, Tom (Abril 2017), "Chapter 7: Overfitting", Algorithms to Live By: The computer science of human decisions, William Collins, pp. 149–168, ISBN 978-0-00-754799-9
- Hawkins, Douglas M. (2004), "The problem of overfitting", Journal of Chemical Information and Modeling, 44.1: 1–12.
- Domingos, Pedro (2015). The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World. Basic Books. ISBN 978-0-465-06192-1.
- Russell & Norvig (2003), Intractability and efficiency and the combinatorial explosion, pp. 9, 21–22
- Sikos, Leslie F. (June 2017). Description Logics in Multimedia Reasoning. Cham: Springer. doi: 10.1007/978-3-319-54066-5. ISBN 978-3-319-54066-5

- Luger, George; Stubblefield, William (2004). *Artificial Intelligence: Structures and Strategies for Complex Problem Solving* (5th ed.). Benjamin/Cummings. ISBN 978-0-8053-4780-7
- Bishop, C. M. (2006), *Pattern Recognition and Machine Learning*, Springer, ISBN 978-0-387-31073-2
- Peter Menzel and Faith D'Aluisio. *Robosapiens*. Cambridge: The MIT Press, 2000. Pg. 66
- Waddell, Kaveh (2018). "Chatbots Have Entered the Uncanny Valley". *The Atlantic*. Retrieved 24 April 2018.
- Daniel Merkle; Martin Middendorf (2013). "Swarm Intelligence". In Burke, Edmund K.; Kendall, Graham. *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*. Springer Science & Business Media. ISBN 978-1-4614-6940-7
- Russell, Stuart J.; Norvig, Peter (2009). *Artificial Intelligence: A Modern Approach* (3rd ed.). Upper Saddle River, New Jersey: Prentice Hall. ISBN 978-0-13-604259-4.
- Dina Bass (20 September 2016). "Microsoft Develops AI to Help Cancer Doctors Find the Right Treatments". *Bloomberg*.
- Senthilingam, Meera (12 May 2016). "Are Autonomous Robots Your Next Surgeons?". *CNN*. Cable News Network. Archived from the original on 3 December 2016.
- Keremany, D; Goldbaum, M; Zhang, Kang. "Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning" (PDF). [www.cell.com](http://www.cell.com).

- Marwala, Tshilidzi; Hurwitz, Evan (2017). *Artificial Intelligence and Economic Theory: Skynet in the Market*. London: Springer. ISBN 978-3-319-66104-9.
- Dominik Scherer, Andreas C. Müller, and Sven Behnke: "Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition," In 20th International Conference Artificial Neural Networks (ICANN), pp. 92–101, 2010. doi:10.1007/978-3-642-15825-4\_10.
- Zell, Andreas (1994). "chapter 5.2". *Simulation Neuronaler Netze [Simulation of Neural Networks]* (in German) (1st ed.). Addison-Wesley. ISBN 978-3-89319-554-1.
- McGann, J. P. (2017). Poor human olfaction is a 19th-century myth. *Science*, 356(6338), eaam7263. <https://doi.org/10.1126/science.aam7263>
- Neal, Radford (25 July 2013). "Deferred evaluation in Renjin, Riposte, and pqR". Radford Neal's blog.
- Wierman, Mark J. "An Introduction to the Mathematics of Uncertainty: including Set Theory, Logic, Probability, Fuzzy Sets, Rough Sets, and Evidence Theory" (PDF). Creighton University. Archived (PDF) from the original on 30 July 2012.
- Antunes, Ricardo; Poshdar, Mani (2018). "Envision of an integrated information system for project-driven production in construction". *Proc. 26th Annual Conference of the International Group for Lean Construction (IGLC)*: 134–143. doi:10.24928/2018/0511. Retrieved 27 December 2018.
- Baburajan, Rajani (2011-08-24). "The Rising Cloud Storage Market Opportunity Strengthens Vendors".

R, D. (2009, enero 14). 8–PROGRAMACION DE PLC «Automatización industrial mediante PLCs. Recuperado de <https://davidrojasticsplc.wordpress.com/category/8-programacion-de-plc/>

"Cyberthreats, Vulnerabilities and Attacks on SCADA Networks" (PDF). Rosa Tang, berkeley.edu. Recuperado del original (PDF) publicado el 13 de Agosto del 2012.  
Ret

Bennett, S. (1993). A History of Control Engineering 1930-1955. London: Peter Peregrinus Ltd. On behalf of the Institution of Electrical Engineers. ISBN 978-0-86341-280-6.

"Motor Starters Start Stops Hand Off Auto". Exman.com. Archived from the original on 13 April 2014.

CODESYS. (2018). Why CODESYS? Recuperado de <https://www.codesys.com//the-system/why-codesys.html>

Peters, Tim (19 August 2004). "PEP 20 – The Zen of Python". Python Enhancement Proposals. Python Software Foundation.

Edwin Wise Robotics Demystified. McGraw-Hill.

Antonio M.Griolo, Member, IEEE, Jaime Chen, Manuel Daz, Daniel Garrido and Augusto Casaca, An Integrated WSA and SCADA System for Monitoring a Critical Infrastructure, IEEE transaction on industrial informatics, vol. 10, no.3, August 2014.

Predix Platform | GE Digital. (2017). Recuperado de <https://www.ge.com/digital/iiot-platform>

Bélanger-Barrette, M. (2015). What Does Collaborative Robot Mean? Recuperado de <https://blog.robotiq.com/what-does-collaborative-robot-mean>

Yao, M. (2017, abril 14). 4 Unique Challenges Of Industrial Artificial Intelligence. Recuperado de <https://www.forbes.com/sites/mariyayao/2017/04/14/unique-challenges-of-industrial-artificial-intelligence-general-electric/#68eacfc41305>

Boyer, Stuart A. (2010). SCADA Supervisory Control and Data Acquisition. USA: ISA - International Society of Automation. p. 179. ISBN 978-1-936007-09-7.

Groover, Mikell (2014). Fundamentals of Modern Manufacturing: Materials, Processes, and Systems.



## DECLARACIÓN Y AUTORIZACIÓN

Yo, **Sabando García, Klever Alejandro**, con C.C: # **1312830431** autor/a del trabajo de titulación: “**Desarrollo de un sistema SCADA para la supervisión de un proceso industrial utilizando inteligencia artificial**”I previo a la obtención del título de **Ingeniero Electrónico en Control y Automatismo** en la Universidad Católica de Santiago de Guayaquil.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, 28 de febrero del 2020

---

Nombre: **Sabando García, Klever Alejandro**

C.C: **1312830431**





## **REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA**

### **FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN**

<b>TEMA Y SUBTEMA:</b>	Desarrollo de un sistema SCADA para la supervisión de un proceso industrial utilizando inteligencia artificial.		
<b>AUTOR(ES)</b>	Sabando García, Klever Alejandro		
<b>REVISOR(ES)/TUTOR(ES)</b>	Mendoza Merchán, Eduardo Vicente		
<b>INSTITUCIÓN:</b>	Universidad Católica de Santiago de Guayaquil		
<b>FACULTAD:</b>	Facultad de Educación Técnica Para el Desarrollo		
<b>CARRERA:</b>	Ingeniería Electrónica en Control y Automatismo		
<b>TÍTULO OBTENIDO:</b>	Ingeniero Electrónico en Control y Automatismo		
<b>FECHA DE PUBLICACIÓN:</b>	28 de febrero del 2020	<b>No. DE PÁGINAS:</b>	106
<b>ÁREAS TEMÁTICAS:</b>	Controles Industriales, Sistemas SCADA, Inteligencia Artificial, Programación		
<b>PALABRAS CLAVES/ KEYWORDS:</b>	SCADA Inteligente, Interfaz De Comunicación, Lógica Difusa, Control Industrial, Función De Pertenencia, Defuzificación, CODESYS, Mamdani.		
<b>RESUMEN/ABSTRACT:</b>	<p>En el presente trabajo de titulación se ha realizado el diseño de un sistema SCADA para la supervisión de un proceso industrial aplicando inteligencia artificial a través de algoritmos de lógica difusa, para ello se debe diseñar y simular el entorno industrial a través del programa CODESYS, el cual permite programar dicho proceso, y a la vez utilizando el programa de I.A. para realizar la supervisión inteligente de la simulación. El estudio realizado es de tipo correlacional, lo cual implica la relación entre las variables de posicionamiento en un proceso de ensamblaje vehicular. Este tipo de procesos necesitan manejar estrictos estándares de montaje para poder cumplir con las expectativas de la actual industria automovilística, la cual requiere de una amplia capacidad de procesamiento para realizar la administración y toma de decisiones con respecto a esta clase de aplicaciones. Los resultados de las pruebas de simulación arrojaron los siguientes resultados: Las mediciones de la posición de las piezas a trasladar se realizaron correctamente, la supervisión y control inteligente de los elementos del proceso, cintas transportadoras, brazos robóticos, piezas, etc. El algoritmo basado en lógica difusa, para la supervisión de la simulación del proceso industrial de tipo línea de ensamblaje vehicular, logró optimizar el control, la supervisión y adquisición de datos del proceso analizado.</p>		
<b>ADJUNTO PDF:</b>	<input checked="" type="checkbox"/> SI	<input type="checkbox"/> NO	
<b>CONTACTO CON AUTOR/ES:</b>	<b>Teléfono:</b> +593-980410056	E-mail: alesabando@hotmail.com	
<b>CONTACTO CON LA INSTITUCIÓN (COORDINADOR DEL PROCESO UTE):</b>	<b>Nombre:</b> Mendoza Merchán, Eduardo Vicente		
	<b>Teléfono:</b> +593-9-9-85086815		
	<b>E-mail:</b> eduardo.mendoza01@cu.ucsg.edu.ec		
<b>SECCIÓN PARA USO DE BIBLIOTECA</b>			
<b>Nº. DE REGISTRO (en base a datos):</b>			
<b>Nº. DE CLASIFICACIÓN:</b>			
<b>DIRECCIÓN URL (tesis en la web):</b>			