

**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**Facultad de Ingeniería en Electricidad y Computación**

Ingeniería de detalle de un sistema integral de monitoreo y control para la automatización de andenes en centros logísticos a través de un sistema SCADA con Inteligencia Artificial.

**PROYECTO INTEGRADOR**

Previo a la obtención

del Título de:

**Ingeniero en Electrónica y Automatización Industrial**

**Presentado por:**

Luis Alejandro Bonilla Lucas

Christian Jair Castillo Esparza

GUAYAQUIL - ECUADOR

Año: 2022

## **DEDICATORIA**

Este proyecto va dedicado a Dios y a mi familia, a mis abuelos Miriam Esperanza Calahorrano y Feliciano Arnulfo Lucas que están disfrutando en el cielo y siempre me cuidan, a mi pilar y apoyo incondicional que es mi madre, Mariela Adriana Lucas, a mi hermana María José Bonilla y mi padre José Luis Bonilla. Fueron indispensables para poder culminar mi carrera universitaria, me apoyaron siempre y me dieron fuerzas para nunca rendirme, el esfuerzo tiene sus recompensas.

Este es un éxito de los muchos que se vienen. También se lo dedico a todos mis amigos y profesores de buen corazón que conocí en el transcurso de esta etapa.

**Alejandro Bonilla**

El presente proyecto va dedicado a mi querida Familia, y en especial a mis padres que, durante todo este camino, han sido el apoyo incondicional, estando presentes en los momentos más difíciles y ayudándome a superarlos.

Y a mis colegas, compañeros y amigos que he conocido en el transcurso de esta etapa. Gracias por los buenos momentos, enseñanzas y por compartir sus conocimientos que permitieron alcanzar este hito en mi vida. Este logro es tanto mío como de ustedes.

**Christian Castillo**

## **AGRADECIMIENTOS**

Les agradezco con todo mi corazón a Dios y a mi familia, gracias a ellos soy la persona que soy hoy en día, me supieron educar con valores y metas inquebrantables. A mis profesores por todas sus enseñanzas y por guiarme en toda esta etapa universitaria y lograr convertirme en un gran profesional. Especialmente en esta última etapa al PhD. César Martín y al MSc. Dennys Cortez, quienes nos ayudaron con sus conocimientos a mi colega Christian Castillo y a mí, para poder lograr este proyecto de titulación.

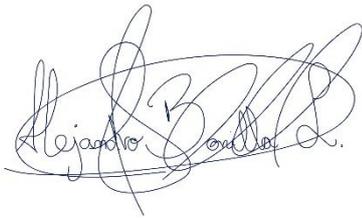
**Alejandro Bonilla**

El presente proyecto se lo agradezco a mis padres Walter Castillo e Inés Esparza, su apoyo incondicional y orientación han sido esenciales en mi camino hacia la presente meta. Gracias por ser mi pilar de motivación y por creer en mí sin importar lo que pase.

**Christian Castillo**

## DECLARACIÓN EXPRESA

"Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; **Luis Alejandro Bonilla Lucas** y **Christian Jair Castillo Esparza**, y damos nuestro consentimiento para que la ESPOI realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"

A handwritten signature in black ink, appearing to read "Alejandro Bonilla Lucas". The signature is stylized and somewhat cursive.

Luis Alejandro Bonilla  
Lucas

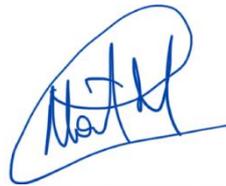
A handwritten signature in black ink, appearing to read "Christian Jair Castillo Esparza". The signature is stylized and somewhat cursive.

Christian Jair Castillo  
Esparza

## EVALUADORES

.....  
**MSc. Dennys Cortez Alvarez**

PROFESOR DE LA MATERIA



.....  
**PhD. César Martín Moreno**

PROFESOR TUTOR

## RESUMEN

La automatización de andenes en centros logísticos se ha convertido en una tendencia debido a los numerosos beneficios que proporciona en términos de eficiencia operativa, mejoras en la gestión de la carga y reducción en los tiempos de espera. Además, ayuda a reducir los costos laborales y mejora la seguridad industrial. Este proyecto propone la ingeniería a detalle, diseño y simulación de un sistema integral de monitoreo y control para la automatización de 8 andenes en el centro logístico de una multinacional en Ecuador, a través de un sistema SCADA junto con Inteligencia Artificial.

Como parte de la solución se diseñó y simuló un sistema SCADA en donde se visualiza de forma atractiva toda la situación en el área. Controlando y recibiendo señales de todos los sensores, finales de carrera y cámaras de seguridad a través de un PLC. En el sistema SCADA se integró un sistema de Inteligencia Artificial a través de una pasarela de comunicación OPC UA, que permite la visualización de la disponibilidad de los andenes por medio de las cámaras de seguridad.

Al realizar lo detallado se obtuvo como respuesta un sistema integral, donde por medio de las diferentes pantallas del SCADA se logró un monitoreo y control absoluto de la situación en tiempo real de los 8 andenes, reduciendo así tiempos muertos y agilitando la operación en la zona, además de aportar significativamente en temas de seguridad industrial.

Con este proyecto se demostró la viabilidad de la integración de ambos sistemas, y se observó que combinar tecnologías convencionales junto con tecnologías actuales e inteligencia artificial son puntos muy diferenciales en la industria y de gran escalabilidad. Además, contribuye significativamente al ODS “Industria, Innovación e Infraestructura”, y se correlaciona con el propósito de vida de la empresa, “Hacer de la sustentabilidad algo cotidiano”.

**Palabras Clave:** Sistema SCADA, Inteligencia Artificial, Monitoreo, Control

## **ABSTRACT**

*The automation of platforms in logistics centers has become a trend due to the numerous benefits it provides in terms of operational efficiency, improvements in load management, and reduction in waiting times. In addition, it helps reduce labor costs and improves industrial safety. This project proposes the detailed engineering, design, and simulation of an integral monitoring and control system for the automation of 8 platforms a multinational's logistics center in Ecuador, through a SCADA system together with Artificial Intelligence.*

*As part of the solution, a SCADA system was designed and simulated in which the entire situation in the area is attractively displayed. Controlling and receiving signals from all sensors, limit switches, and safety cameras through a PLC. An Artificial Intelligence system was integrated into the SCADA system through an OPC UA communication gateway, which allows the visualization of the availability of the platforms through the security cameras.*

*As a result, an integral system was obtained, whereby means of the different SCADA screens, absolute monitoring and control of the situation in real-time of the 8 platforms was achieved. This reduces downtime and speeds up operations in the area, in addition to making a significant contribution to industrial safety.*

*This project demonstrated the feasibility of integrating both systems and showed that combining conventional technologies with current technologies and artificial intelligence are very differential points in the industry and of great scalability. In addition, it contributes significantly to the SDG "Industry, Innovation and Infrastructure", and correlates with the life purpose of the company, "Make Sustainable Living Commonplace".*

**Keywords:** *SCADA System, Artificial Intelligence, Monitoring, Control*

# ÍNDICE GENERAL

EVALUADORES.....	5
RESUMEN.....	I
<i>ABSTRACT</i> .....	II
ÍNDICE GENERAL.....	III
ABREVIATURAS .....	VI
ÍNDICE DE FIGURAS.....	VII
ÍNDICE DE TABLAS .....	X
CAPÍTULO 1 .....	11
1.    Introducción .....	11
1.1    Descripción del problema .....	13
1.2    Justificación del problema.....	15
1.3    Objetivos.....	16
1.3.1    Objetivo General .....	16
1.3.2    Objetivos Específicos .....	16
1.4    Marco teórico .....	17
1.4.1    Automatización de procesos Industriales .....	17
1.4.2    PLC .....	18
1.4.3    Sistema SCADA.....	18
1.4.4    Sensores .....	19
1.4.5    Actuadores .....	20
1.4.6    Gemelo Digital ( <i>Digital Twin</i> ).....	21
1.4.7    Inteligencia Artificial.....	22
CAPÍTULO 2.....	23

2.	Metodología .....	23
2.1	Definición del proceso de integración del proyecto.....	23
2.2	Identificación y levantamiento de la información.....	25
2.3	Diseño Del Sistema de Control y Adquisición de Datos (SCADA).....	25
2.3.1	Configuraciones generales.....	25
2.3.2	Descripción de bloques FC .....	26
2.3.3	Descripción del bloque de rutina principal (main).....	30
2.3.4	Diseño de pantallas del SCADA.....	31
2.3.5	Topología de red de comunicación del sistema .....	34
2.4	Diseño y entrenamiento del modelo de Inteligencia Artificial (Deep Learning) 37	
2.4.1	Adquisición del Dataset para el entrenamiento del modelo de deep learning 38	
2.4.2	Preprocesamiento de información de entrada.....	40
2.4.3	Algoritmo de predicción y modelo de IA usando Deep Learning.....	43
2.4.4	Algoritmo optimizado usando Deep Learning.....	46
CAPÍTULO 3.....		47
3.	Resultados y Análisis.....	47
3.1	Programación PLC y Comunicación Industrial – Simulación .....	47
3.2	Ejecución del Runtime SCADA.....	49
3.3	Sistema de Visión Artificial o por Computadora.....	54
3.3.1	Ejecución de la red neuronal convolucional profunda inicial .....	54
3.3.2	Pruebas del modelo de red neuronal convolucional profunda inicial.....	56
3.3.3	Ejecución de la red neuronal convolución profunda optimizada y principal 59	
3.3.4	Pruebas del modelo de red neuronal convolución profunda optimizada y principal 61	

3.3.5 Pruebas finales del modelo de Deep Learning optimizado principal .....	63
CAPÍTULO 4 .....	66
4. Conclusiones Y Recomendaciones.....	66
Conclusiones .....	66
Recomendaciones .....	67
BIBLIOGRAFÍA .....	69
5. Bibliography .....	69
APÉNDICE.....	71

## ABREVIATURAS

ESPOL	Escuela Superior Politécnica del Litoral
ODS	Objetivos de desarrollo sostenible de la ONU
CND	Centro Nacional de Distribución
HPC	Home and Personal Care
LATAM	Latinoamérica
EC	Ecuador
SHE	Security, Health, and Environment
PLC	Controlador Lógico Programable
I/O	Entradas y Salidas
SCADA	Supervisión, Control y Adquisición de datos
HMI	Interfaz máquina-hombre
DL	Deep Learning
CPS	Sistemas cibernéticos
IA	Inteligencia Artificial
FRAMES	Fotogramas o cuadros por segundo
RGB	Red, Green, Blue
COLAB	Colaboratory
OPC UA	Arquitectura unificada de comunicaciones de plataforma abierta
SAT	Pruebas de aceptación en sitio
FAT	Pruebas de aceptación de fábrica
FC	Bloque de función
RT	Relé térmico
TF	TensorFlow
ML	Aprendizaje automático
DL	Aprendizaje profundo
PN	Profinet
PC	Computador
Reset	Reinicio
DB	Bloque de Datos

## ÍNDICE DE FIGURAS

Figura 1: Racks para las 10700 posiciones del centro logístico.....	11
Figura 2: Área de los 40 andenes de carga/descarga .....	12
Figura 3: Sistema Industrial .....	17
Figura 4: Sensores.....	19
Figura 5: Esquema gráfico, metodología desarrollo del proyecto IA-SCADA .....	24
Figura 6 Proceso de despacho con medidas de control de seguridad.....	26
Figura 7 Bloque de función FormatoAnden .....	27
Figura 8 Rutina habilitador Permisivo Puerta.....	27
Figura 9 Rutinas condicionales de operación del andén.....	28
Figura 10 Rutinas condicionales estados de operación del andén .....	28
Figura 11 Mapeo señal marca-contacto físico .....	30
Figura 12 Mapeo señal marca-contacto físico .....	30
Figura 13 Uso de FC1 con las marcas generadas para cada andén .....	31
Figura 14 Pantalla inicio de sesión del sistema .....	32
Figura 15 Pantalla de control de alarmas .....	32
Figura 16 Pantalla monitoreo de andenes de carga para despacho.....	33
Figura 17 Pantalla de estación de mantenimiento del PLC y módulos de entrada y salida .....	33
Figura 18 Topología de red.....	34
Figura 19 Vista del Proyecto de dispositivos y redes en TIA Portal .....	35
Figura 20 Asignación de IP y máscara de subred para la red PN/IE .....	35
Figura 21 Instalación del paquete de OPC UA para python .....	36
Figura 22 Rutina de comunicación para envío de datos desde la predicción de la IA hasta la base de datos del SCADA.....	36
Figura 23: Denominación de un andén libre .....	37
Figura 24: Denominación de un andén ocupado .....	37
Figura 25: Primera parte del código para desfragmentar un video en frames .....	38
Figura 26: Segunda parte del código para desfragmentar un video en frames .....	39
Figura 27: Segmentación de los datos.....	39
Figura 28: Conexión del drive e importación de librerías a usarse en el proyecto .....	40
Figura 29: Creación de las variables train_ds y val_ds.....	41

Figura 30: Creación de la variable <code>class_names</code> y código para muestra de 16 datos aleatorios del Dataset .....	42
Figura 31: Muestra de 16 datos aleatorios del Dataset con sus clases respectivas ..	42
Figura 32: muestra del batch, pixeles de los datos y canal.....	43
Figura 33: Creación de la red neuronal convolucional con sus diferentes capas .....	44
Figura 34: Compilación del modelo y el summary del mismo .....	45
Figura 35: Resultado del <code>summary()</code> , es el detalle de la red neuronal convolucional, con sus capas respectivas.....	45
Figura 36: Mismo modelo de red neuronal previamente detallado, pero integrada la capa de Dropout antes de la última. ....	46
Figura 37: Simulación de la red profinet IE 1 .....	47
Figura 38: Compilación y ejecución de bloque main.....	48
Figura 39: Tabla de variables enlazadas al SCADA .....	49
Ilustración 40: Compilación del sistema SCADA.....	49
Figura 41: Comprobación del envío de datos desde el código en Python al SCADA	50
Figura 42: Ejecución del runtime – pantalla principal.....	51
Figura 43: Pantalla de visualización del estado de andenes (8 andenes) .....	51
Figura 44: Ejecución del runtime – pantalla de señales del PLC .....	52
Figura 45: Entrenamiento de la red neuronal convolucional inicial .....	54
Figura 46: Fragmento de código hecho con numpy y matplotlib.....	55
Figura 47: Gráficas Entrenamiento vs Validación (Accuracy) y Entrenamiento vs Validación (Loss) .....	55
Figura 48: Código con OpenCV para redimensionar un frame o imagen .....	57
Figura 49: Predicción de un andén ocupado o full con exactitud de 74.65%.....	58
Figura 50: Predicción de un andén libre o empty con exactitud del 100%.....	59
Figura 51: Entrenamiento de la red neuronal convolucional optimizada principal.....	60
Figura 52: Gráficas Entrenamiento vs Validación (Accuracy) y Entrenamiento vs Validación (Loss)   Para el modelo optimizado .....	60
Figura 53: Predicción por parte de la red optimizada de un andén ocupado o full con exactitud de 75.95% .....	62
Figura 54: Predicción por parte de la red optimizada de un andén libre o empty con exactitud de 100% .....	63

Figura 55: Fragmento de código para la visualización de una matriz de 25 datos con sus respectivas predicciones ..... 64

Figura 56: Visualización de la matriz de 25 imágenes con sus respectivas etiquetas reales y predicciones por la IA optimizada..... 65

## ÍNDICE DE TABLAS

Tabla 1 Checklist para pruebas de aceptación del cliente .....	52
Tabla 2: Tabla de variables de entrada (Marcas).....	71
Tabla 3: Tabla de variables de salida (Marcas) .....	72
Tabla 4: Tabla de variables de salida (físicas) .....	73
Tabla 5: : Tabla de variables de salida (físicas) .....	74

# CAPÍTULO 1

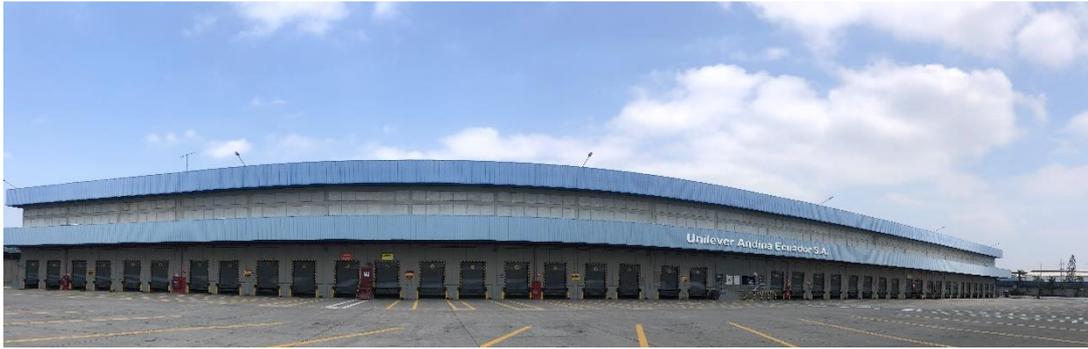
## 1. INTRODUCCIÓN

En una empresa multinacional global, justamente en LATAM Ecuador, en el sector llamado como CND (centro nacional de distribución), tiene como función principal ejercer en la etapa final para la gestión en Supply Chain. Se encarga de recibir toda la producción de la planta y otros países de Latinoamérica, para almacenarlos a la espera de ser expedidas y distribuidas a todo el Ecuador y también para exportación, ya sean a empresas grandes, pequeñas, fábricas u otros almacenes o bodegas.

En la empresa el sector es denominado como Supply Chain Logistics LATAM Ecuador, dentro del site denominado Planta Guayas, y está ubicado en el km 25 vía Daule en la ciudad de Guayaquil, provincia del Guayas, dentro de este site se encuentra CND HPC, que es el CND para la distribución de productos Home and Personal Care, en el cual el almacén logístico cuenta con estructuras metálicas llamadas racks logísticos para poder guardar, almacenar y conservar todos los productos, todos los racks del almacén tienen la capacidad para almacenar 10700 posiciones, en otras palabras 10700 pallets completos de productos (*ver figura 1*), para los cuales el área también cuenta con 40 andenes de carga/descarga de productos (*ver figura 2*).

**Figura 1: Racks para las 10700 posiciones del centro logístico**





**Figura 2: Área de los 40 andenes de carga/descarga**

El espacio total que ocupa CND HPC son 11800 m<sup>2</sup>, comparándolos con los 120800 m<sup>2</sup> de todo el site nos damos cuenta de que CND ocupa una parte considerable de todo el site Planta Guayas.

Dada a que el área logística es una de las más importantes en cuestiones de negocios, es un sector clave que debe ser automatizado con las tecnologías actuales para así poder resolver los problemas de una forma más sencilla y eficaz posible, además de crear soluciones escalables e innovadoras.

Actualmente nos encontramos en la era de la digitalización, esto conlleva a que la sociedad como la conocemos hoy en día ya está dando o ya ha dado el gran salto hacia lo digital, transformando tanto hábitos de vida personales como hábitos profesionales, convirtiendo todo aquello que antes se lo realizaba por medio de papel, en un proceso absolutamente digital, a través de apps móviles o procesos automatizados que les permitan gestionar cualquier circunstancia que estén pasando e incluso poder almacenar datos y poder manejarlos de cualquier forma para poder ahorrar tiempo y dinero, e inclusive otras varias variables de interés, según sea el caso.

El propósito de vida de la compañía es, **“Hacer de la sustentabilidad algo cotidiano”**, están decididos a demostrar que su modelo de negocio con propósito y orientado al futuro ofrece un rendimiento excepcional, es por esto que uno de los factores más importantes de la digitalización es reducir en lo posible el uso de papel, esto conlleva al resultado que la compañía en Ecuador sea más eficiente y

sostenible, con menor uso del área para su almacenaje, además de la capacidad de reducir fuertemente el coste de su manipulación e incidencias que pueden ocurrir en los procesos logísticos de gran tamaño.

La automatización de procesos en el sector logístico en una de las tendencias en franca expansión en los últimos años, en donde cada vez más empresas a nivel mundial apuestan por la automatización en su logística, porque como ya lo dijimos anteriormente, esta es una de las áreas claves para el negocio. Además automatizar un sector tan grande como lo es el de la logística conlleva varios beneficios, y entre los cuales está la de mayor seguridad industrial, o como se lo denominaría en la compañía, contribuirá mucho a la parte de SHE ( *Security, Health and Environment*), ya que los procesos automatizados eliminan considerablemente los daños en los productos derivados de la intervención de los colaboradores o trabajadores, y también previniendo los accidentes que involucren maquinarias, equipos de carga o descarga de producto u operarios.

## **1.1 Descripción del problema**

Actualmente en CND - HPC de la multinacional en Ecuador, todo el control que se tiene de los 40 andenes de carga/descargas disponibles del área se lo realiza de forma manual por parte de los colaboradores. No existe ningún sistema de control logístico automatizado de los andenes, tampoco sistemas que permitan controlar los tiempos para mejorar la eficacia en las expedidas o recepciones a nivel LATAM de productos, y críticamente tampoco existen protocolos automatizados a nivel de seguridad industrial, para asistir ante una situación crítica que este sucediendo en un uno o varios andenes, agilitando la comunicación hacia la oficina corporativa que se encuentra en la mitad de toda el área del CND.

En meses pasados ante la situación actual, han ocurrido accidentes en donde el conductor del camión parqueado en el área externa del andén arranca sin

que el producto haya sido completamente embarcado o desembarcado, además como el área de los 40 andenes es extensa no se sabe el estado actual de los andenes en tiempo real, cuál está disponible, cual no y cuales están en mantenimiento. No existen indicadores que permitan una comunicación eficiente, ya que la calidad y el tiempo de información que se transmite entre colaboradores en ocasiones se distorsiona, por ende, la situación de emergencia o crítica que este sucediendo en ese momento, puede tardar en resolverse o inclusive ocasionar una reacción en cadena del problema.

De acuerdo a lo anterior, la problemática radica en cómo se realizará un monitoreo y control automatizado sobre la situación en tiempo real de los andenes y en tener un control más generalizado sobre los tiempos involucrados en la operación de despacho y recepción de productos en el centro logístico HPC, y a futuro poder mejorar tanto la seguridad en esta zona de la cadena de suministros como implementar inteligencia comercial, obteniendo los indicadores claves de rendimiento tales como reducir tiempos muertos, además de reducir riesgos con los datos obtenidos.

El plan que tiene considerado la compañía es un plan piloto, en donde se contempla aplicar la automatización para resolver estos problemas ya planteados primeramente en 8 andenes, con un presupuesto de 30 mil USD para su implementación y puesta en marcha total. Es por esto por lo que los 8 andenes serán muy bien estudiados para poder seleccionarlos entre los 40 en total. Este proyecto de automatización e innovación deberá marcar un antes y un después en la operación actual, para que en consecuencia la compañía tenga resultados realmente positivos de forma significativa.

## 1.2 Justificación del problema

La multinacional llegó a Ecuador en 1995, y hasta la actualidad el CND no tiene un control en los andenes y sus alrededores de forma automatizada, toda operación logística más allá del control de inventario es manual. Nuestra solución es la más apropiada, ya que enfocamos el proyecto hacia una era y perspectiva digitalizada y automatizada orientada a la industria 4.0 reforzando el control y monitoreo en tiempo real del área de los andenes tanto externa como interna, de tal manera que se mejore su eficiencia y así mismo también contribuimos en el tema de SHE para el centro logístico.

Además, y muy importante, como se lo dijo anteriormente, el propósito de la Multinacional Global es hacer de la sustentabilidad algo cotidiano. Por ende, se tiene como prioridad que cualquier proyecto de innovación y automatización a ejecutarse en la compañía contribuya a los ODS de la ONU, y es por esto por lo que este proyecto se basa en los siguientes ODS:

**ODS n. 9, Industria, Innovación e Infraestructura**, porque la industrialización inclusiva y sostenible junto con la innovación e infraestructura potencializan de una forma significativa la generación de empleos decentes y aumento de ingresos, siendo este uno de los objetivos más importantes de la ONU y su meta es cumplirlo hasta el 2030, en donde países en vías de desarrollo tal como lo es Ecuador, tiene que darse prisa y este proyecto es un paso más hacia esta meta.

También se contribuye al **ODS n. 8, trabajo decente y crecimiento económico** y al **ODS n. 12, producción y consumo responsable**, que significa hacer más y mejor con menos, aumentando de forma significativa la eficiencia de los recursos, en este caso la eficiencia energética.

## **1.3 Objetivos**

### **1.3.1 Objetivo General**

Realizar de manera exitosa la ingeniería a detalle de un sistema integral de monitoreo y control para la automatización de 8 andenes de un centro nacional logístico a través de un sistema SCADA.

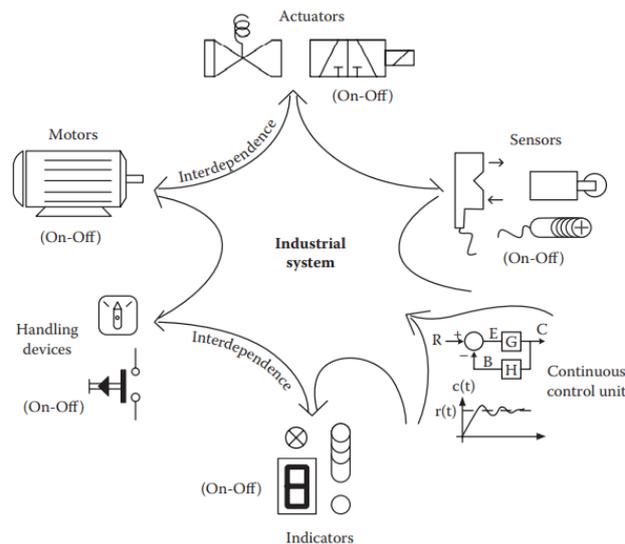
### **1.3.2 Objetivos Específicos**

- Diseño del sistema de control de forma escalable en función de las variables de interés para gestionar de manera exitosa la reducción y agilización de tiempos muertos.
- Diseñar un sistema de monitoreo que se integre al sistema de control para poder comunicar de manera exitosa los componentes físicos con el software.
- Construcción y evaluación de un modelo de simulación para verificar la fiabilidad del sistema en general.
- Refuerzo de la seguridad industrial a través del sistema automatizado y la digitalización de las variables de seguridad.
- Reducción y agilización de tiempos con el control de información en los despachos a través de un sistema de gestión de datos e indicadores visuales.
- Desarrollo de un sistema de inteligencia artificial con cámaras para la detección de producto en los diferentes andenes.

## 1.4 Marco teórico

### 1.4.1 Automatización de procesos Industriales

Previo a desglosar los conceptos de monitoreo y control, es necesario recoger conceptos sobre automatización de procesos, concepto utilizado desde los inicios del siglo XVII cuyo proceso comienza mediante la implementación de elementos electromecánicos aplicando mecanismos complejos interviniendo elementos como los relés, motores, temporizadores, etc. Por lo que desde aquel momento comenzó una revolución en la tecnología orientada a la industria. Con el pasar de los años y con el impacto de las primeras máquinas autónomas potenciadas por microcontroladores nacen los controladores lógico-programables los cuales dan un paso hacia la automatización moderna.



**Figura 3: Sistema Industrial**

Extraído de "Introduction to Industrial Automation" de Stamatios Manesis pg. 3

Por lo que necesariamente es imprescindible conocer los conceptos esenciales de cada uno de los elementos que gobiernan en un sistema de automatización industrial actuales y los que en esta nueva revolución industrial intervendrán.

### 1.4.2 PLC

En la automatización de procesos, se implementan dispositivos que puedan soportar condiciones ambientales extremas tales como altas temperaturas, condiciones abrasivas en el ambiente, alta contaminación mediante ruido eléctrico, vibraciones producidas por las máquinas del proceso. Estos dispositivos conocidos como PLC deben ser robustos y soportar estas condiciones extremas y son utilizados en todo tipo de industria conocida [1].

### 1.4.3 Sistema SCADA

Según sus siglas, Supervisory Control and Data Acquisition, un sistema SCADA es un sistema o herramienta que facilita el control y la supervisión de uno o varios procesos que nos permite observar variables de interés del proceso a través de la detección de desvíos, anomalías mediante avisos y alarmas.

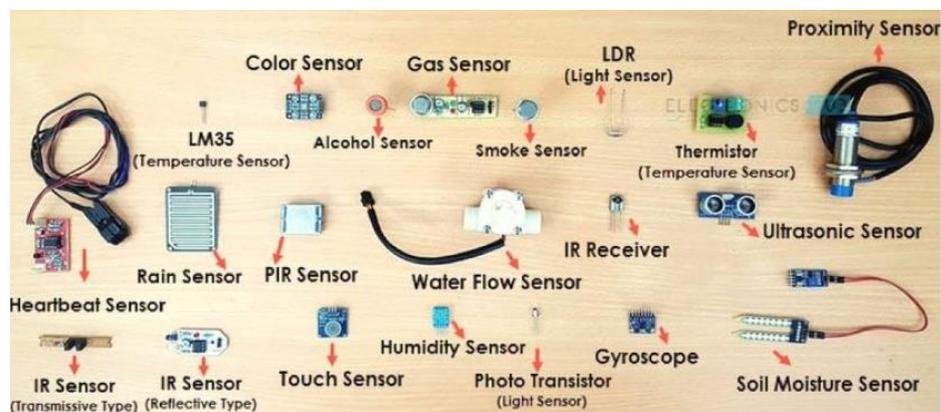
Según la página AutexOpen se tiene como funciones adicionales al sistema SCADA [2]:

- Monitorización en tiempo real: el sistema SCADA puede recopilar y transmitir datos en tiempo real, lo que permite una monitorización constante, elemento clave de la supervisión de los operadores de los equipos y/o plantas de producción.
- Análisis de datos: el sistema SCADA puede analizar lo obtenido mediante la adquisición de datos para generar alarmas o registrar eventos.
- Generación de alarmas: si se detecta un problema, el sistema SCADA puede generar alarmas para notificar al operario.

#### 1.4.4 Sensores

Un sensor es un artefacto/dispositivo que nos posibilita analizar, medir y procesar cambios o estímulos en el ambiente. Los sensores transforman los fenómenos físicos en una señal ya sea analógica o digital medible y que a su vez nos posibilita cuantificar las señales y observarlas en un HMI (Interfaz hombre-máquina) [3].

Los sensores son utilizados en gran medida en la vida cotidiana de las personas e industria.



**Figura 4: Sensores**

Fuente: Dewesoft [¿Qué es un Sensor y Qué Hace? | Dewesoft](#)

En la industria, es imprescindible tener en cuenta el estado del proceso, variables físicas y químicas que conllevan al óptimo control de esta por lo que los sensores sobre un proceso son de suma importancia nos permite llevar y cuantificar diferentes magnitudes físicas o químicas del proceso y transformarlas a una señal eléctrica analógica o digital de tal modo que podamos automatizar los procesos.

Aplicaciones concurrentes de los sensores en la industria según la página SDI [4]:

- Variables de proceso, como la temperatura, la presión, el nivel, el flujo, etc.

- Variables eléctricas, como el voltaje, la corriente y la frecuencia.
- Variables mecánicas, como la rotación, el número de ciclos, la posición, la dirección de desplazamiento, las presiones estáticas y dinámicas, tal como es la de proximidad.
- Variables ambientales, como humedad, vibración, velocidad o dirección del viento.

#### 1.4.5 Actuadores

Los actuadores son dispositivos que permiten la transformación de un tipo de energía a movimiento mecánico para una actividad del proceso en específico. Los actuadores pueden ser lineales o giratorio. Los de tipo lineal permiten el desplazamiento mecánico en una dirección específica que permite empujar o tirar, mientras que los de tipo giratorio permite convertir energía en movimientos circulares. Generalmente se utilizan energías de tipo eléctrica, neumática o hidráulica (Las dos últimas aplicadas como gradientes de presión) [5].

Existen diferentes tipos de actuadores los cuales se clasifican en [5]:

- Actuator mecánico. Funcionan transformando un tipo de movimiento (rotacional o lineal). Funcionan combinando diferentes componentes como engranajes, poleas, cadenas, resortes y guías.
- Actuadores neumáticos. Su fuente de energía es el aire comprimido, lo que les permite responder rápidamente para iniciar y detener las operaciones. Son seguros, potentes, fiables y económicos.
- Actuator hidráulico. Usan fluidos como el aceite para crear movimientos que requieren mayor fuerza para manejar cargas pesadas.
- Actuator eléctrico. Estos son limpios, fáciles de usar y están listos para usar. Requieren electricidad de una fuente de alimentación externa para accionar el motor y convertir la energía eléctrica en fuerza mecánica.

- Actuador térmico. Como sugiere su nombre, utilizan energía térmica o magnética para crear movimiento. Se caracterizan por su bajo peso, muy económicos y ofrecen una alta densidad de potencia.

#### **1.4.6 Gemelo Digital (*Digital Twin*)**

Los gemelos digitales se visualizan como una tecnología disruptiva en la simulación y análisis de plantas industriales para el aumento del performance o rendimiento de los beneficios de la transformación digital de las fábricas, una transformación en la que ya han participado muchas empresas y que se intensificará en los próximos años.

Los gemelos digitales son representaciones digitales "vivas" de los equipos y procesos que componen una fábrica de manera computarizada, conectados a través de "sistemas cibernéticos" (CPS) a los sistemas reales que representan. Con información en tiempo real de los procesos, historial de operatividad de máquinas y de mantenimiento, y la aplicación de técnicas de aprendizaje automático, se pueden obtener modelos de alta precisión que se comportan de manera muy similar a los sistemas reales. [6]

Algunas de las ventajas que nos proporciona los gemelos digitales para procesos productivo y su índice de calidad según la página "*ennmotive*" [7]:

- Mantenimiento predictivo y correctivo
- Reducción de paradas de producción.
- Optimización de costes de línea de producción
- Eficiencia energética
- Calidad del producto
- Tiempo de ingeniería para el diseño de modelos 3D paramétricos
- Controles de software de simulación según requisitos técnicos
- Sensores para recogida de datos
- Ingeniería de datos para análisis (aprendizaje automático)

- Plataforma de monitoreo de comunicaciones

#### **1.4.7 Inteligencia Artificial**

La inteligencia artificial explica el contexto de cómo las máquinas copian/reproducen características de la inteligencia del ser humano para el desarrollo de actividades y que mientras a medida que avanzan en su proceso iterativo, mejoran en cada paso a partir de la información procesada. La IA (Inteligencia Artificial) se enfoca en el desarrollo de actividades robustas y complejas que antes necesitaban el capital humano para el cumplimiento de estos procesos o algoritmos tales como por ejemplo comunicación chatbot con clientes o juegos de ajedrez [9].

# CAPÍTULO 2

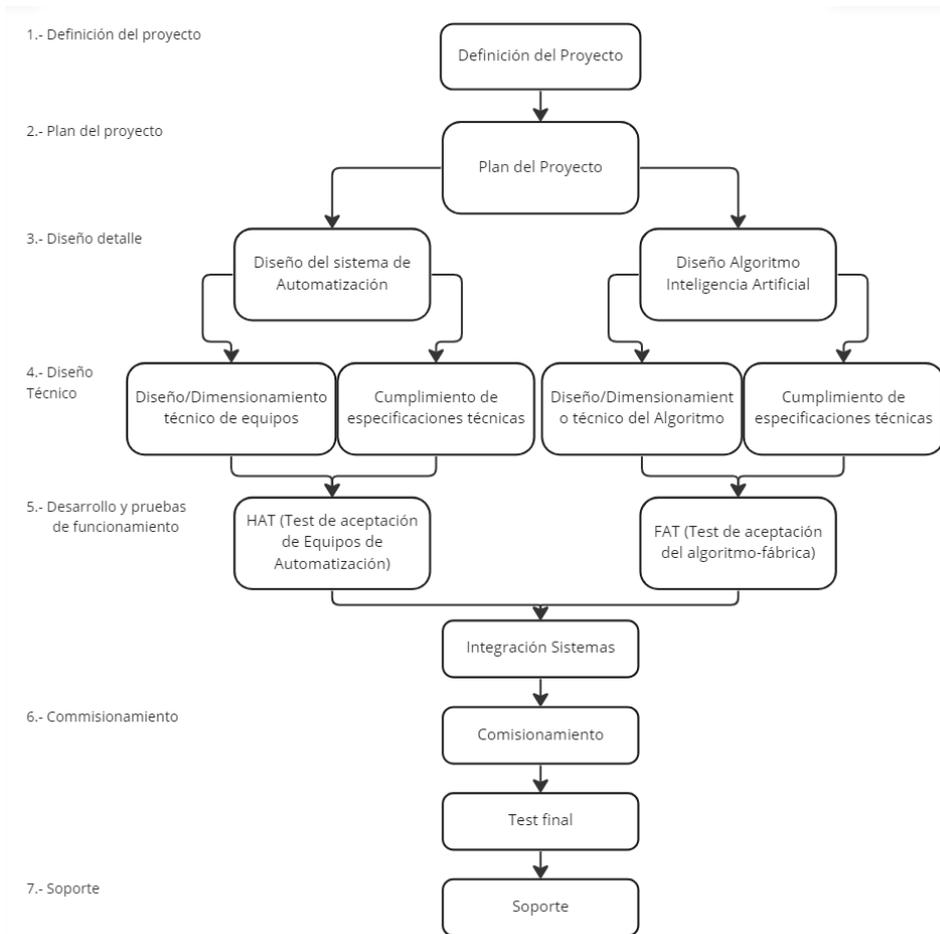
## 2. METODOLOGÍA

En el presente capítulo se describen los procedimientos utilizados para el desarrollo de la ingeniería de detalle del proyecto, que incluye el estudio y manejo del sistema logístico de andenes, la creación y desarrollo de sistemas SCADA, creación de un algoritmo de control para recibir y enviar señales físicas cómo también virtuales para el caso de las señales recibidas del modelo de inteligencia artificial, así como la programación, detalle de librerías y lenguaje de programación usado para el desarrollo de la red neuronal y entrenamiento de ésta, además de la pasarela de comunicación entre el sistema SCADA y la IA para la visualización efectiva de los datos.

### 2.1 Definición del proceso de integración del proyecto

Con la finalidad de desarrollar un plan definido tanto para el desarrollo de una inteligencia artificial como el del sistema SCADA, se deben tomar a consideración todos los elementos o prerrequisitos que engloba el desarrollo del proyecto. Para el entrenamiento de la Inteligencia Artificial se utiliza el lenguaje de programación Python más la implementación de la librería de Tensorflow con el framework Keras, que es una librería de código abierto para el desarrollo y entrenamiento de redes neuronales, además del uso de la librería OpenCV para casos puntuales que se detallarán más adelante.

Para un seguimiento del desarrollo del proyecto es imprescindible tener a consideración un flujo del trabajo en donde se involucran las partes que se preparan, procesan y prueban a través de la prueba de aceptación para una óptima conformidad en la entrega del proyecto. En la figura 5 se desenvuelve el proceso que se presenta a consideración para el desarrollo de proyectos.



**Figura 5: Esquema gráfico, metodología desarrollo del proyecto IA-SCADA**

**Definición del proyecto:** En la fase de definición, se lleva a cabo una evaluación de las necesidades del cliente. Se examina el proceso de diseño y se elabora un plan para la automatización, ilustrando las opciones técnicas al cliente.

**Plan del proyecto:** Es esencial para registrar la estructura del proyecto, así como las responsabilidades, aprobaciones, acuerdos y medidas de seguridad. Un componente clave del plan de proyecto es el proceso para la aprobación y validación de cambios, y cómo se manejarán las implicaciones financieras y de planificación.

**Diseño de la solución y diseño técnico:** Corresponde a la etapa compuesta por componentes de hardware y software. Durante la fase de diseño, se ajustarán con precisión las interacciones entre el hardware y el software. La fase de Diseño Funcional incluye la elaboración de los detalles técnicos del hardware y la especificación y registro de los requerimientos funcionales del software

**Desarrollo y pruebas de funcionamiento:** En la Fase de Integración, los paneles de control son ensamblados y el software es programado. Los ingenieros instalan y conectan los cables en el lugar, supervisados por un gerente de montaje. La configuración de simulación también puede utilizarse para capacitar a los operadores o familiarizar a los técnicos de mantenimiento con el sistema.

**Puesta en marcha del sistema:** Está diseñada para causar un mínimo impacto. El rendimiento del sistema será evaluado en colaboración con el cliente y se concluirá con una Prueba de Aceptación en el Sitio (SAT) y el sistema estará listo para ser utilizado. Una vez que se hayan resuelto los problemas pendientes y se haya proporcionado la documentación se finalizará el proyecto.

**Soporte:** Entregado el sistema, en caso de que el cliente/usuario presente inconvenientes, o temas de mantenimiento de los activos, se procede a realizar un soporte dependiendo de la necesidad presentada.

## **2.2 Identificación y levantamiento de la información**

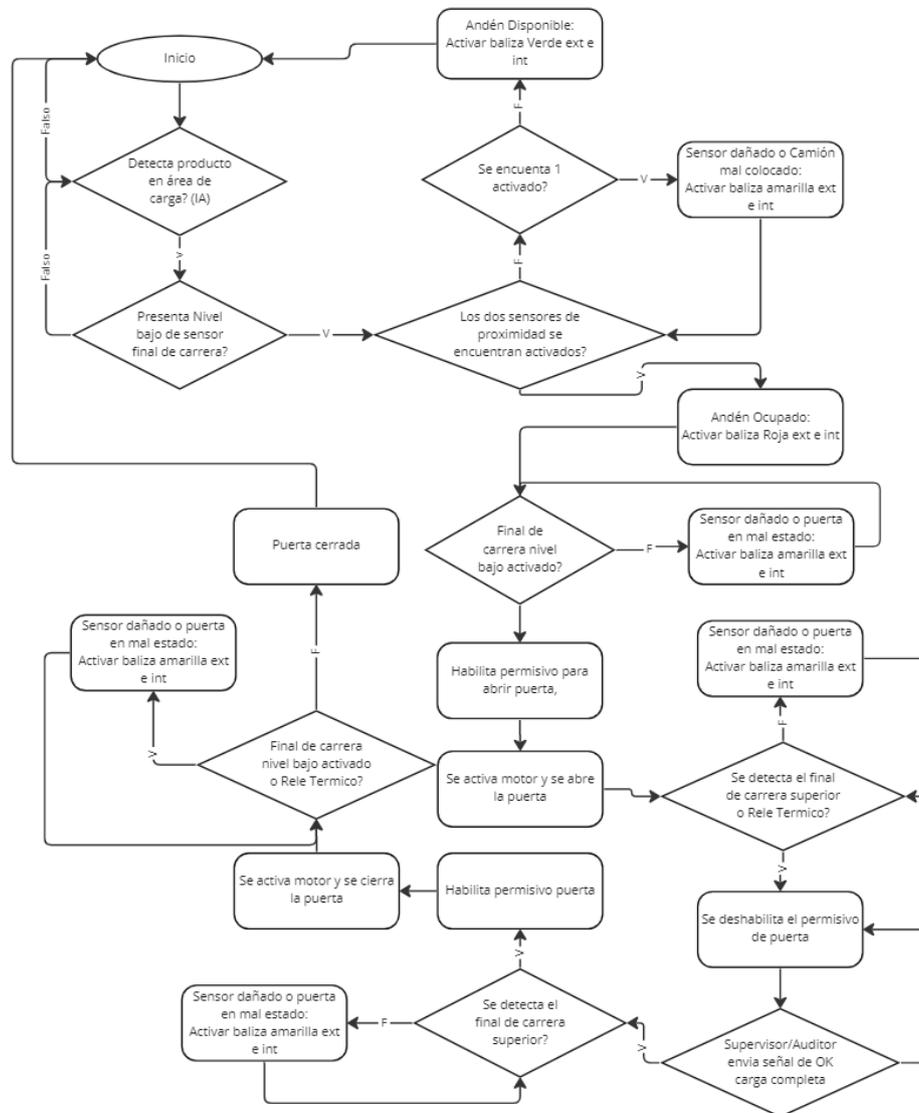
Para comenzar con el diseño del sistema, se define en primer lugar el caso de estudio el cual presenta un sistema logístico de una empresa sin control en la gestión de despacho, únicamente operando a nivel ERP mediante confirmaciones de solicitudes de forma manual. Es por esto que se encuentra aquí una oportunidad única en donde se puede atacar y solucionar varios problemas por medio de automatización industrial e inteligencia artificial, combinando así tanto la automatización industrial estándar en el mercado con tecnologías actuales en auge hoy en día, como son las redes neuronales convolucionales, dando soluciones no tan complejas para el entendimiento del cliente pero sí óptimas y eficaces a la hora de monitorizar y controlar el área logística de la multinacional.

## **2.3 Diseño Del Sistema de Control y Adquisición de Datos (SCADA)**

### **2.3.1 Configuraciones generales**

Se presentan diferentes secciones en donde se describe tanto la descripción del módulo/bloques FC, controles de Marca sobre salidas/Entradas físicas, así como

centralización de datos a través del bloque de DB1, conocido como bloque de datos. En la siguiente figura, se presenta un diagrama de flujo que se debe seguir para el correcto funcionamiento del proceso de despacho en el centro logístico.



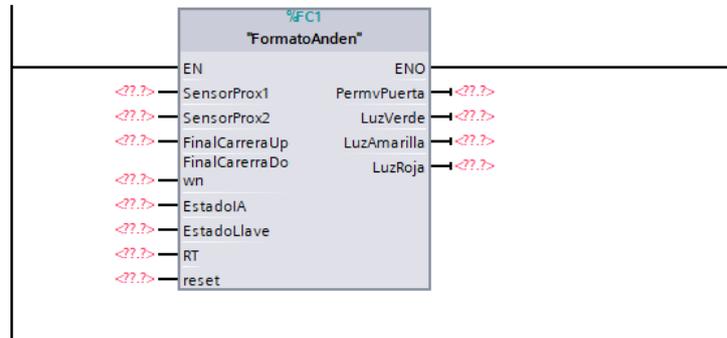
**Figura 6 Proceso de despacho con medidas de control de seguridad**

### 2.3.2 Descripción de bloques FC

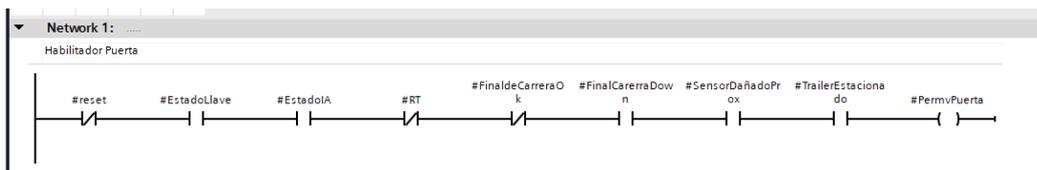
#### Bloque FC1 FormatoAnden

Se presenta el caso que el estado, control y monitoreo de los andenes es de forma paralela (cada andén interactúa de manera independiente). La implementación de la rutina se facilita debido a que permite a través de un bloque de función, en este caso llamado FormatoAndén, involucrar todas las variables que se presentan en el andén dentro de un bloque formato y de ahí duplicarlo para los demás. Cabe

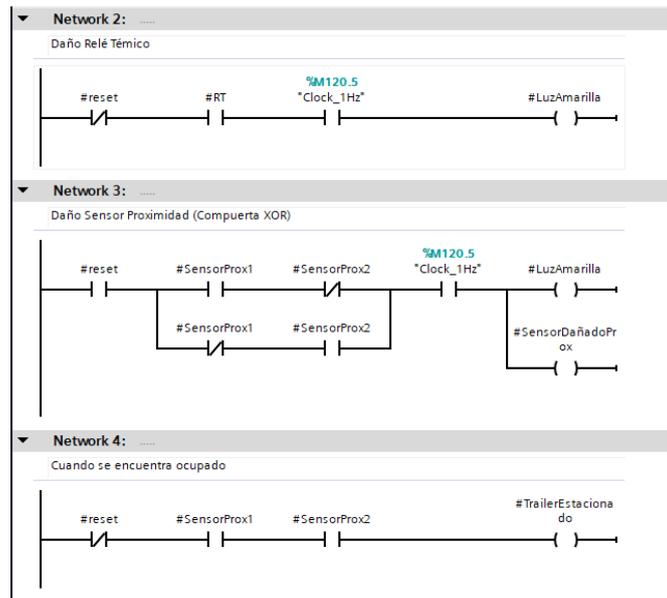
destacar que en esta rutina se deben considerar diferentes casos tales como permisivos, condicionales de apertura, en caso de que exista falla en los sensores. Entonces, considerando lo comentado anteriormente se presenta el bloque de función generado tal como se muestra en la figura 7.



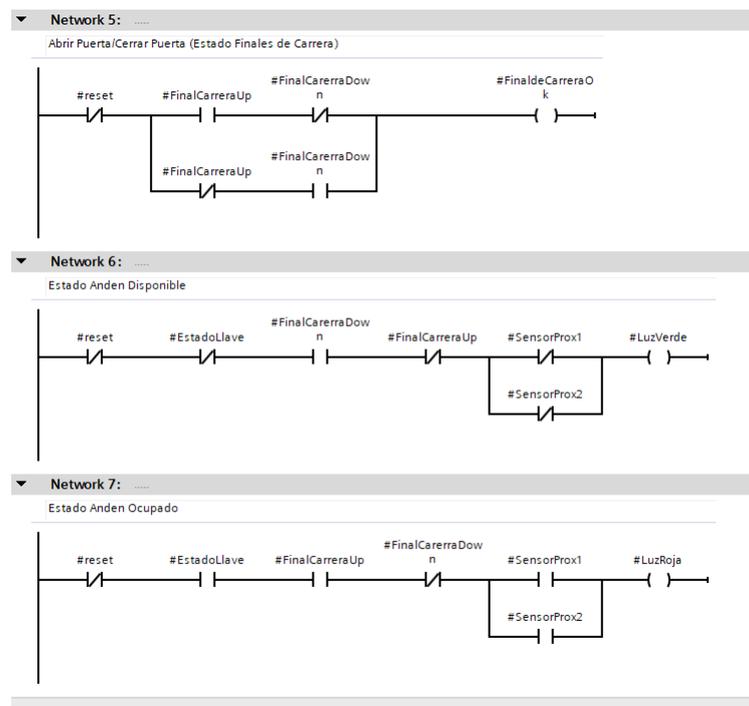
**Figura 7 Bloque de función FormatoAnden**



**Figura 8 Rutina habilitador Permisivo Puerta**



**Figura 9 Rutinas condicionales de operación del andén**



**Figura 10 Rutinas condicionales estados de operación del andén**

El bloque posee 8 entradas las cuales son:

SensorProx1 y SensorProx2, siendo los sensores que serán instalados en las puertas de cada andén para la detección de los vehículos y su correspondiente habilitador.

FinalCarreraUp y FinalCarreraDown, siendo los finales de carrera instalados en la puerta de tal modo que se pueda enviar una señal al operador/supervisor el estado constante de la puerta ante cualquier situación.

EstadoIA, es un dato de predicción de la inteligencia artificial por visión por computador que varía entre los estados 0 o 1, siendo 0 estimando que el área de despacho se encuentra vacío, mientras que 1 estimando que el área de despacho se encuentra completo.

EstadoLlave, siendo el indicador del sensor que va a recoger las llaves del transportista una vez haya llegado al lugar/andén indicado, siendo permisivo para que la rutina de despacho comience.

RT, siendo el relé térmico del motor de la puerta que presenta, en caso de algún fallo con el motor de la puerta, es imprescindible tener a consideración cualquier caso de atasco o fallos en la puerta para poder solventar eficientemente el problema.

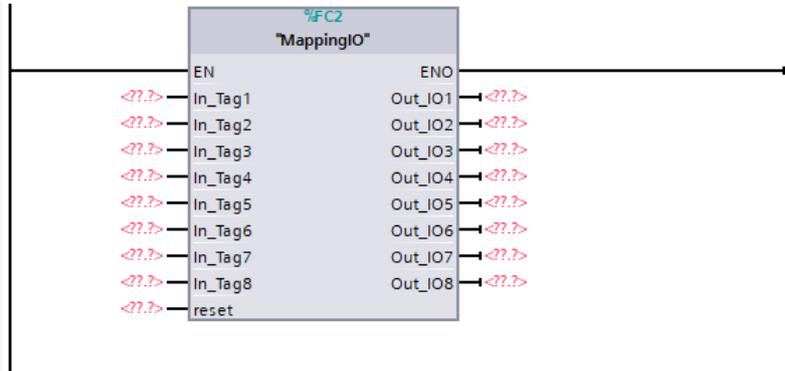
Por último, se presenta el botón reset que para cualquier bloque forma parte como contacto cerrado y posibilita un reinicio de las señales de salida en el bloque.

### **Bloque FC2 MappingIO**

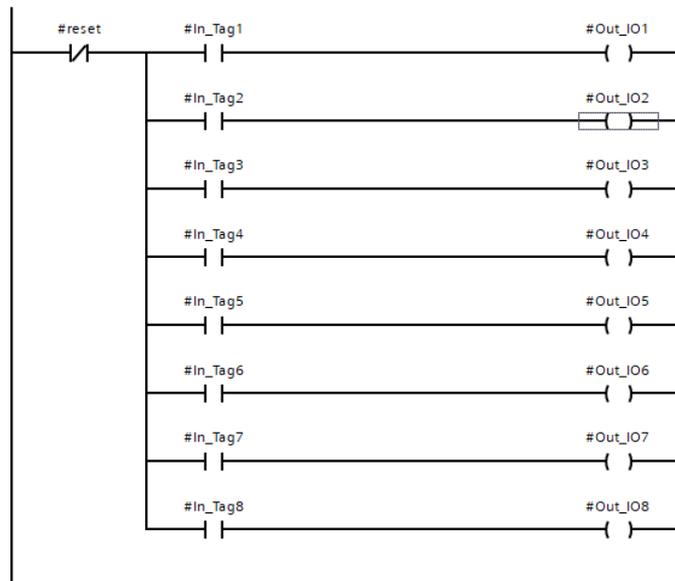
Este bloque permite hacer un enlace entre la marca resultante de una rutina emparejada con la salida física deseada, es decir, presenta una entrada y salida correspondiente, siendo la entrada la marca generada y la salida un contacto físico, de tal modo que enviar una señal a una marca antes de enviarla a una salida física en un PLC es importante por varias razones:

- Protección contra fallos de hardware: al enviar la señal a una marca primero, se puede detectar y corregir cualquier problema en la señal antes de que afecte al hardware.
- Mayor flexibilidad en el diseño del sistema: al enviar la señal a una marca, se pueden aplicar lógica adicional y realizar operaciones complejas antes de enviar la señal a la salida física.
- Control de tiempo: Enviar la señal a una marca permite controlar el tiempo de la señal, permitiendo tiempos de retardo o tiempos de activación específicos antes de enviar la señal a la salida física.

- Mayor seguridad: Al utilizar marcas intermedias, se puede implementar un mayor nivel de seguridad en el sistema, permitiendo controlar el acceso a las salidas físicas y monitorear su estado.



**Figura 11 Mapeo señal marca-contacto físico**

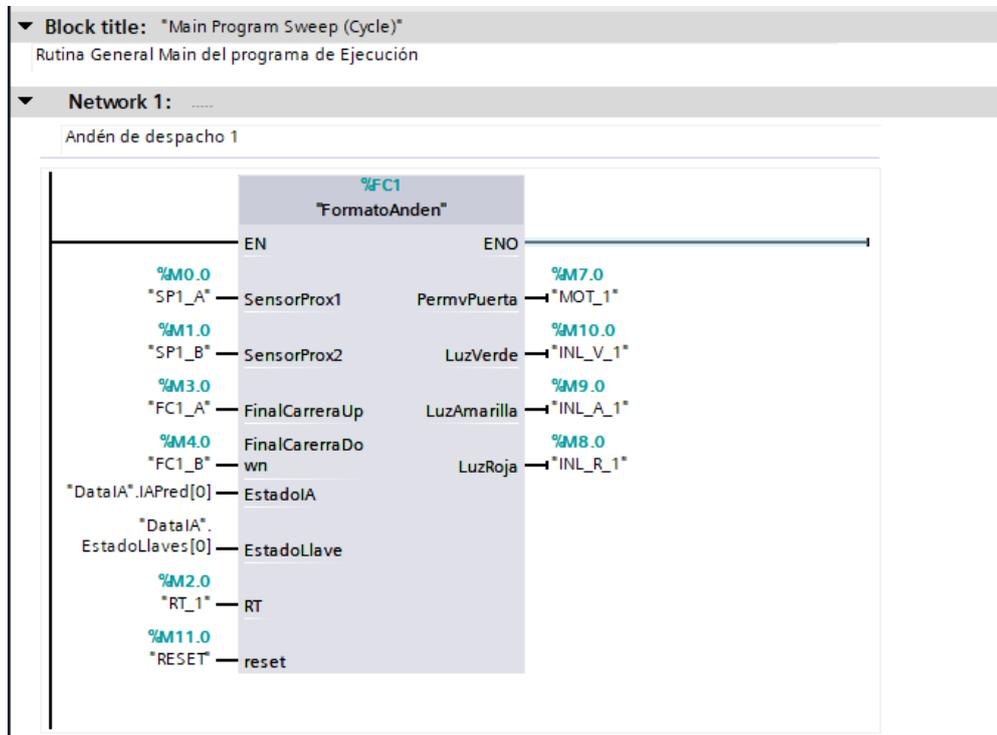


**Figura 12 Mapeo señal marca-contacto físico**

### 2.3.3 Descripción del bloque de rutina principal (main)

El bloque *MAIN* es el bloque principal de programación en un PLC, donde se escribe la lógica principal del sistema de control automático, se establecen las condiciones iniciales, se definen las variables y se escribe el código que se ejecutará continuamente mientras el PLC esté en funcionamiento. Es donde se lleva a cabo la lógica principal del sistema de control automático y toma decisiones y controla acciones del sistema.

Dentro de este bloque tenemos ya el uso de los bloques mostrados con anterioridad con sus respectivas marcas de entrada y salida tal como se puede observar en la siguiente figura.



**Figura 13 Uso de FC1 con las marcas generadas para cada andén**

En anexos, encontrará la tabla general de variables de entrada y de salida, en términos de marcas para un entendimiento general de las marcas utilizadas para este proyecto.

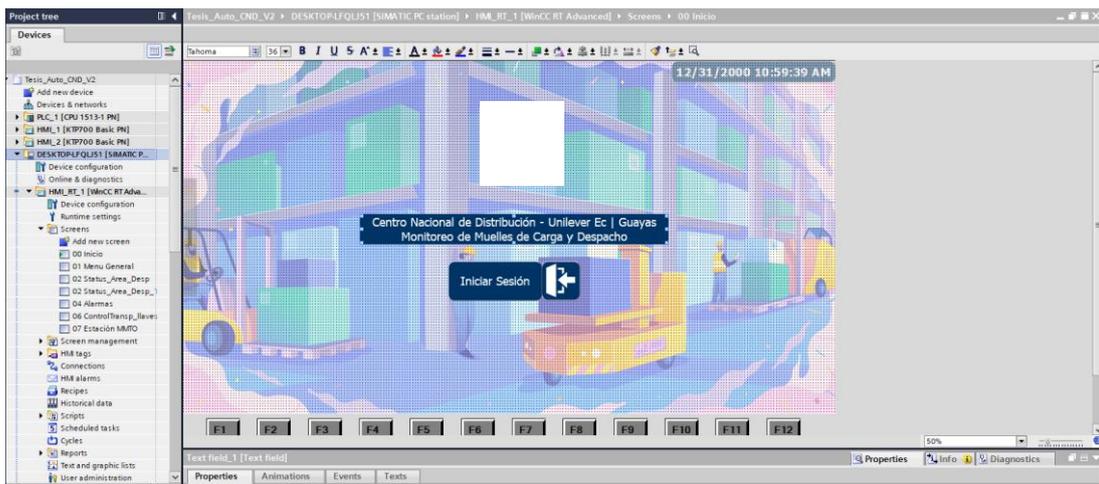
### 2.3.4 Diseño de pantallas del SCADA

Para esta parte del diseño, se toma en cuenta tener un cliente en donde se ejecuta el *runtime* o conocido como ejecutable del programa creado. Para el caso del software utilizado, WinCC Advance de la empresa Siemens, se procede con el diseño del sistema SCADA a través del diseño y la visualización y control de señales del centro logístico.

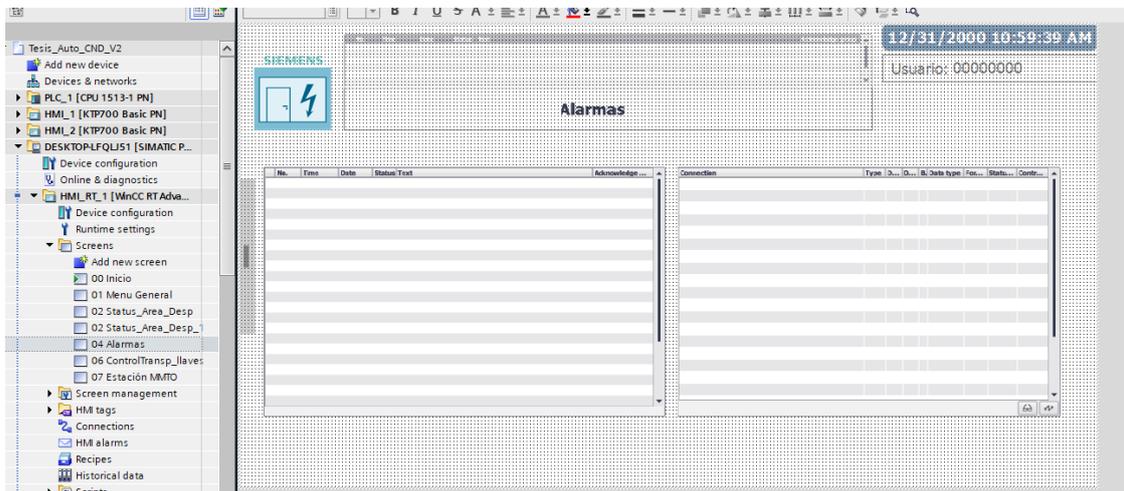
Cumpliendo con las necesidades del cliente, el sistema SCADA contiene los siguientes puntos:

- Sistema de gestión de alarmas: presenta la capacidad de generar alarmas en caso de fallos o condiciones anómalas en los procesos.

- Sistema de control de usuarios y seguridad: presenta la capacidad de generar alarmas en caso de fallos o condiciones anómalas en los procesos.
- Escalabilidad: El sistema es escalable para adaptarse a las necesidades cambiantes del negocio y permitir el crecimiento del sistema.
- Monitoreo Remoto: El sistema permite el monitoreo remoto de los procesos a través de una interfaz de usuario amigable. Adicional a este punto, tener la capacidad de generar una ventaja en tiempos de mantenimiento.



**Figura 14 Pantalla inicio de sesión del sistema**



**Figura 15 Pantalla de control de alarmas**

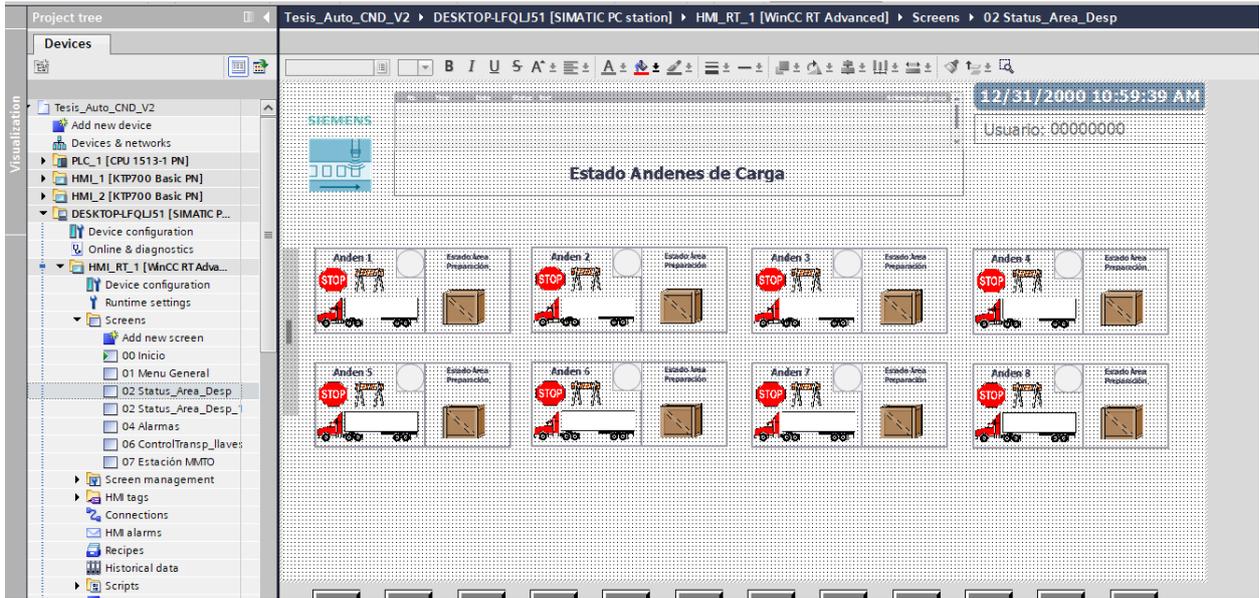


Figura 16 Pantalla monitoreo de andenes de carga para despacho

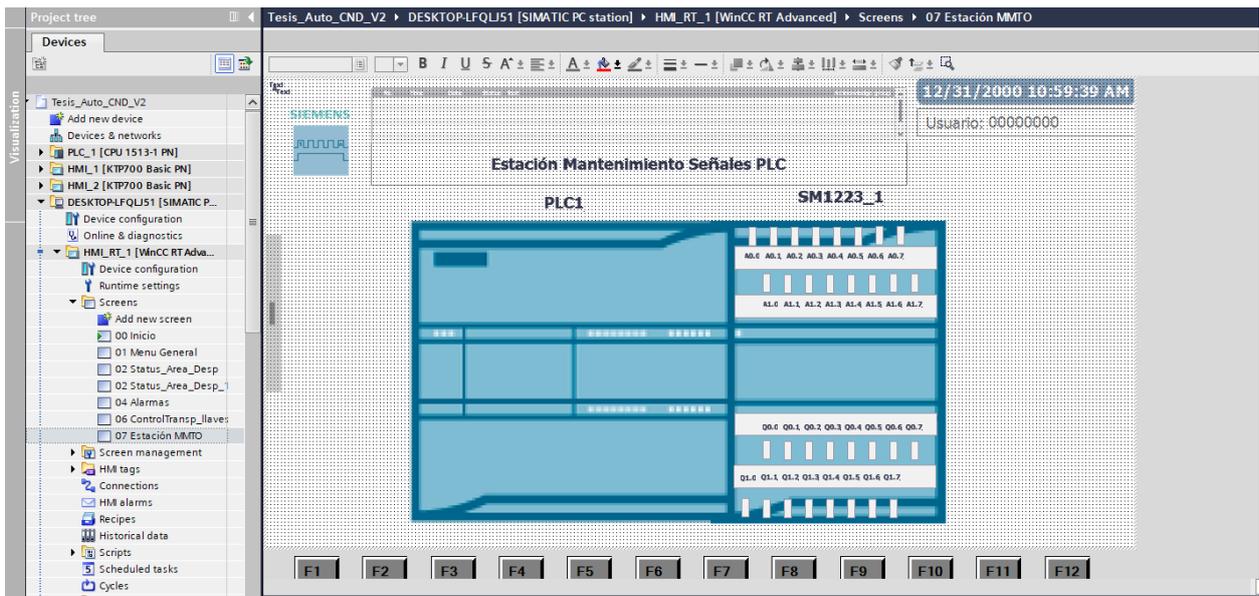
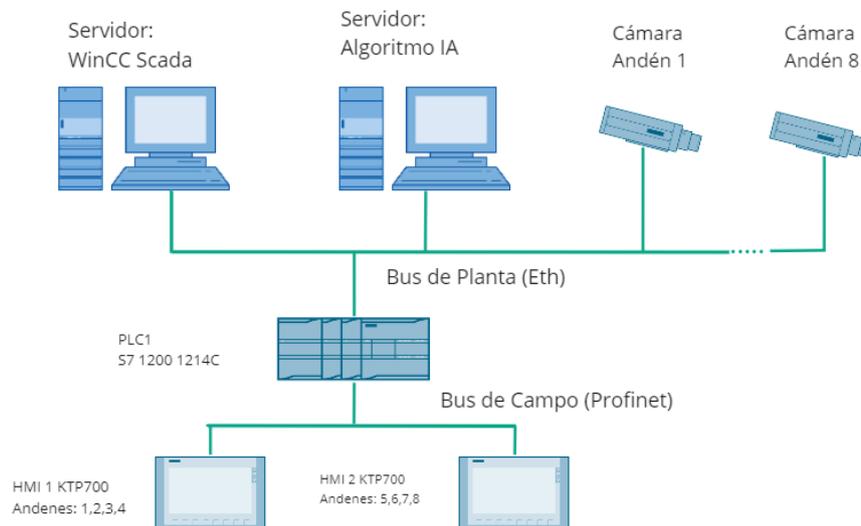


Figura 17 Pantalla de estación de mantenimiento del PLC y módulos de entrada y salida

### 2.3.5 Topología de red de comunicación del sistema

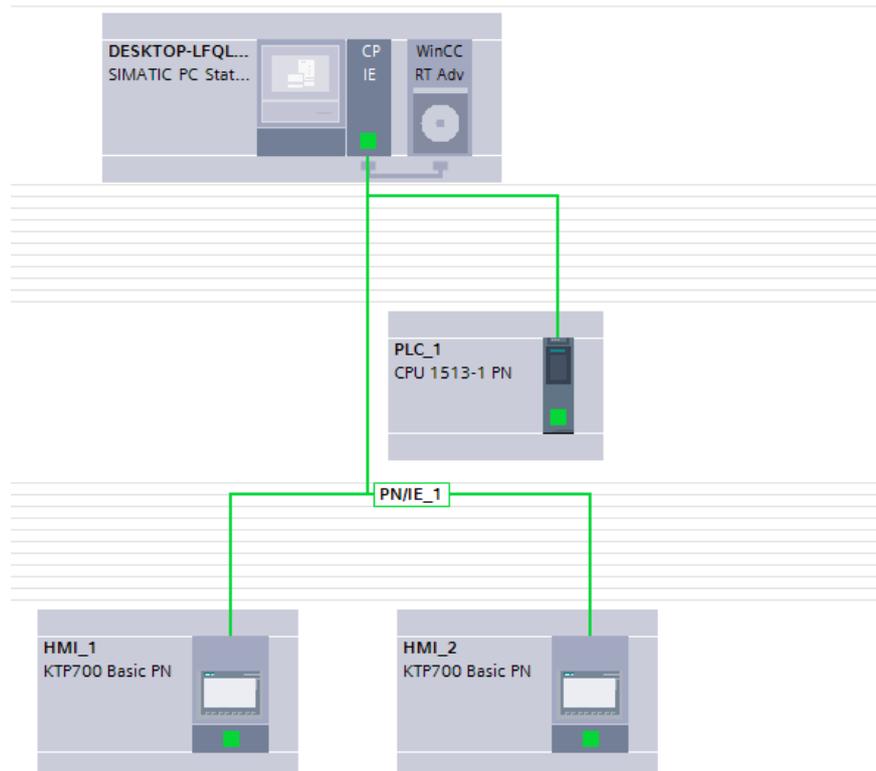
El sistema se compone, según lo detallado anteriormente, en la figura 18 se pueden observar los siguientes puntos:

- Un controlador en donde presenta la rutina de gestión de andenes mediante las señales digitales que se presenten en tiempo real.
- Pantallas HMI en donde se visualiza el estado de estas variables en campo.
- Una máquina virtual o servidor en donde se encuentra el cliente WinCC Scada en donde se presentarán las variables a monitorearse.
- Sistema de monitoreo de cámaras a través de ethernet.

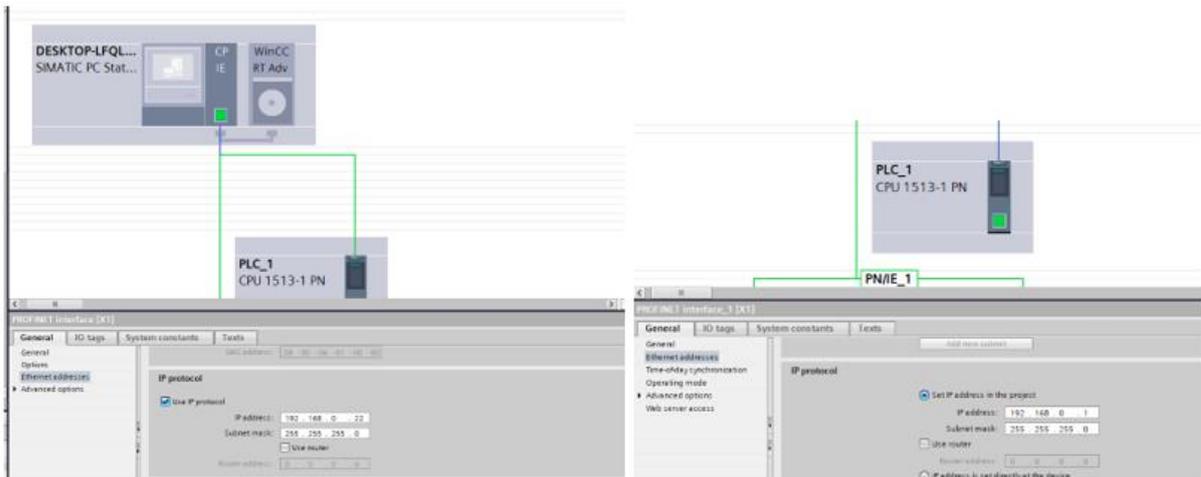


**Figura 18 Topología de red**

Para temas de servicio de automatización, dentro del programa TIA Portal se pueden visualizar los componentes que se encuentran de este sistema.



**Figura 19 Vista del Proyecto de dispositivos y redes en TIA Portal**



**Figura 20 Asignación de IP y máscara de subred para la red PN/IE**

Entonces, para enviar los datos generados por la IA hacia el PLC y monitorear estos parámetros a través del SCADA, se realiza un código en Python que permite realizar la comunicación OPC UA a través de la librería opcua de código abierto y así extraer información del cliente OPC:

## Instalando librería OPCUA

```
pip install opcua
```

```
Collecting opcuaNote: you may need to restart the kernel to use updated packages.
```

Figura 21 Instalación del paquete de OPC UA para python

```
import sys
sys.path.insert(0, "..")

from opcua import Client
from opcua import ua

client=Client('opc.tcp://192.168.0.1:4840')
# client = Client("opc.tcp://admin@localhost:4840/freepcua/server/") #connect using a user

try:
    client.connect()
    # Client has a few methods to get proxy to UA nodes that should always be in address space such as Root or Objects
    root = client.get_root_node()
    print("Objects node is: ", root)

    #Recogemos el nodo del OPC Cliente
    nodo_arrayIA=client.get_node('ns=3;s="DataPLC"."IA_Pred"') #En el parametro s se coloca el nombre del DB.variable
    print("nodo IA: ",nodo_arrayIA)
    ArrayIA=nodo_arrayIA.get_value() #Recogemos el valor del array
    print("El valor del array de predicción en el PLC es: ", ArrayIA)

    #Para enviar un valor al cliente OPCUA con PLC físico
    #nodo_arrayIA.set_value([0,0,0,0,0,0]) #Se coloca el array de predicción para enviar al cliente

    #Para enviar un valor al cliente OPCUA con PLCsim
    dv=ua.DataValue(ua.Variant([0,1,1,0,1,1,1],ua.VariantType.Boolean))
    nodo_arrayIA.set_value(dv)
    ArrayIA=nodo_arrayIA.get_value() #Recogemos el valor del array
    print("El valor del array de predicción en el PLC es: ", ArrayIA)

finally:
    client.disconnect()
    print("Ha finalizado")
```

Figura 22 Rutina de comunicación para envío de datos desde la predicción de la IA hasta la base de datos del SCADA

## 2.4 Diseño y entrenamiento del modelo de Inteligencia Artificial (Deep Learning)

El problema que se requiere solucionar es la detección del estado de los andenes de carga/descarga, observar en tiempo real si el andén esta libre o está ocupado con carga, en las imágenes a continuación se puede observar a qué se denomina un andén libre y ocupado. Para poder solucionar esta problemática se usa un modelo de inteligencia artificial basado en el lenguaje de programación **Python**, ya que este lenguaje actualmente es el más adoptado en temas de machine learning y **deep learning**.

Por medio de cámaras de seguridad industriales se obtiene video, y por consecuencia gracias a la librería **OpenCV**, se desfragmenta ese video en frames, para obtener imágenes en tiempo real, y estas imágenes pasarlas a procesar y analizar a la red neuronal convolucional profunda, desarrollada y entrenada con la librería **TensorFlow** y el framework **Keras**, para finalmente predecir el estado de la imagen (el frame o imagen del andén capturado en tiempo real).



**Figura 23: Denominación de un andén libre**



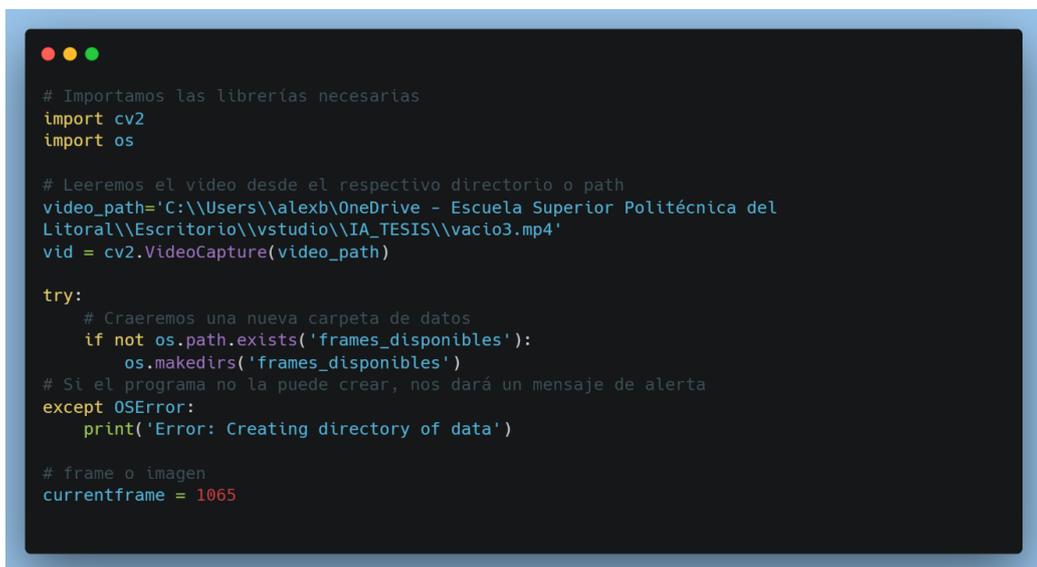
**Figura 24: Denominación de un andén ocupado**

### 2.4.1 Adquisición del Dataset para el entrenamiento del modelo de deep learning

Para el entrenamiento de una red neuronal óptima se necesita un Dataset que sea equilibrado tanto en calidad como cantidad, estos datos son las imágenes de andenes libres y ocupados, en diferentes escenarios y condiciones, para que así el Dataset tenga diversidad a la hora de entrenar el modelo de inteligencia artificial. Se realiza la respectiva adquisición de fotografías en el centro logístico en varios escenarios correspondiente al estudio de este proyecto, pero esto no satisface la demanda de imágenes o datos que se necesita para poder entrenar al modelo de una forma eficaz.

Por este motivo se utiliza la grabación de videos en vez de la adquisición de fotografías una por una, ya que con la librería OpenCV, se puede tomar un video y desfragmentarlo en frames, y así se obtienen varias imágenes en distintos escenarios de una forma rápida y eficaz. Se procesan varios videos grabados como parte de este estudio, y al procesarlos y juntarlos con las imágenes adquiridas, se obtiene un total de 997 datos o imágenes de andenes tanto libres como ocupados.

Con esta cantidad se satisface la demanda de datos que se requiere para poder entrenar la red neuronal profunda, a continuación, se muestra el código hecho en Python con la librería OpenCV.



```
# Importamos las librerías necesarias
import cv2
import os

# Leeremos el video desde el respectivo directorio o path
video_path='C:\\Users\\alex\\OneDrive - Escuela Superior Politécnica del Litoral\\Escritorio\\vstudio\\IA_TESIS\\vacio3.mp4'
vid = cv2.VideoCapture(video_path)

try:
    # Craeremos una nueva carpeta de datos
    if not os.path.exists('frames_disponibles'):
        os.makedirs('frames_disponibles')
# Si el programa no la puede crear, nos dará un mensaje de alerta
except OSError:
    print('Error: Creating directory of data')

# frame o imagen
currentframe = 1065
```

Figura 25: Primera parte del código para desfragmentar un video en frames

```
while (True):

    # Lectura del frame
    success, frame = vid.read()
    if success:
        # Seguiremos creando imágenes o frames hasta que analice todo el
        video    name = './frames_disponibles/frame_000' + str(currentframe) + '.jpg'
        print('Creating..' + name)

        # Escribiremos las imágenes extraídas
        cv2.imwrite(name, frame)
        # incrementando el contador para que
        # muestre cuántos frames se han creado
        currentframe += 1
    else:
        break

# Liberamos todo el espacio y las ventanas una vez hecho
vid.release()
cv2.destroyAllWindows()
```

**Figura 26: Segunda parte del código para desfragmentar un video en frames**

Con los datos recopilados en esta sección ya detallada, se da forma al Dataset, el cual se descompone en datos de entrenamiento haciendo referencia a la carpeta **training**, y datos de validación haciendo referencia a la carpeta **validation**, ambas carpetas en su interior tienen dos carpetas, **full** y **empty**, donde están las imágenes recopiladas de andenes libres y ocupados, y con esto se da como completado el Dataset para el entrenamiento del modelo de IA, a continuación se puede ver de manera más visual la segmentación del Dataset.

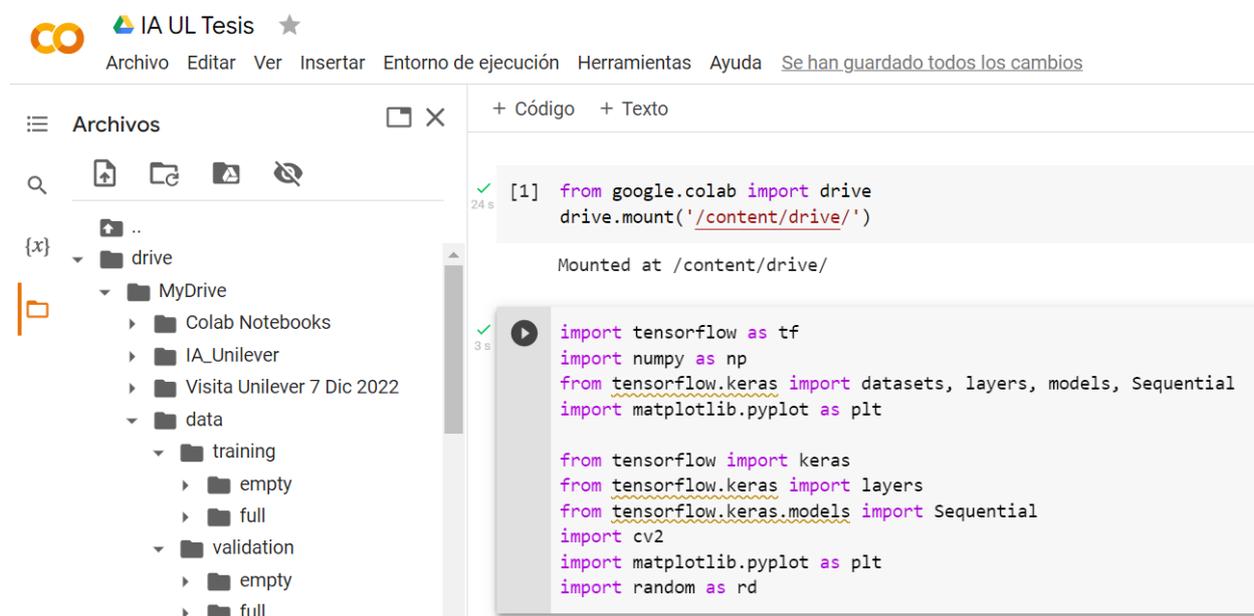
```
-Data
  -Training
    -Full
    -Empty
  -Validation
    -Full
    -Validation
```

**Figura 27: Segmentación de los datos**

## 2.4.2 Preprocesamiento de información de entrada

Para el resto del procesamiento se usa **Google Colaboratory** o “Colab”, el cual es un producto de **Google Research**, que permite **ejecutar código en Python en línea**. Se utiliza este producto ya que, al ser una herramienta en línea hecha específicamente para escribir código en el lenguaje de programación escogido, no tiene problemas de incompatibilidad de librerías o frameworks por las distintas versiones que se utilizan, problemas que, si se pueden tener a la hora de ejecutar y usar librerías en local, se refiere a local al alojar todo el código en memoria física de una computadora y no en línea.

En instancia se sube todo el Dataset ya construido anteriormente a Google Drive, y se conecta este drive a Colab, permitiendo leer toda la información para su posterior uso, además de importar todas las librerías a usar inicialmente. La única librería que se usa adicional a las ya mencionadas es matplotlib, la cual permite visualizar los datos de una forma más agradable a la vista y graficar tendencias o datos críticos que se decidan o escojan mostrar.



The screenshot shows the Google Colaboratory interface. On the left, a file explorer shows a directory structure: 'Archivos' with subfolders 'drive', 'MyDrive', 'Colab Notebooks', 'IA\_Unilever', 'Visita Unilever 7 Dic 2022', 'data', 'training', 'validation', and 'full'. The main area displays two code cells. The first cell, executed in 24 seconds, contains the code: `from google.colab import drive; drive.mount('/content/drive/')`, with the output 'Mounted at /content/drive/'. The second cell, executed in 3 seconds, contains the code: `import tensorflow as tf; import numpy as np; from tensorflow.keras import datasets, layers, models, Sequential; import matplotlib.pyplot as plt; from tensorflow import keras; from tensorflow.keras import layers; from tensorflow.keras.models import Sequential; import cv2; import matplotlib.pyplot as plt; import random as rd`.

**Figura 28: Conexión del drive e importación de librerías a usarse en el proyecto**

Posterior a esto, se alojan en variables y se muestra la cantidad de elementos que tiene el Dataset, tanto datos de entrenamiento como datos de validación para la

red neuronal. Anteriormente se indicó que en modelos de redes neuronales siempre se usa el 80% del Dataset para entrenar al modelo y el 20% para validarlo, es por esto que, al mostrar las cantidades de datos, se muestra que en total se tiene 996 datos o imágenes, y de esos 797 son para entrenamiento y 199 para validación del modelo.

```
▶ train_ds = tf.keras.utils.image_dataset_from_directory(  
    '/content/drive/MyDrive/data/training',  
    validation_split=0.2,  
    subset="training",  
    seed=123,  
    image_size=(400,200))
```

```
↳ Found 996 files belonging to 2 classes.  
Using 797 files for training.
```

```
[4] val_ds = tf.keras.utils.image_dataset_from_directory(  
    '/content/drive/MyDrive/data/training',  
    validation_split=0.2,  
    subset="validation",  
    seed=123,  
    image_size=(400,200))
```

```
Found 996 files belonging to 2 classes.  
Using 199 files for validation.
```

**Figura 29: Creación de las variables train\_ds y val\_ds**

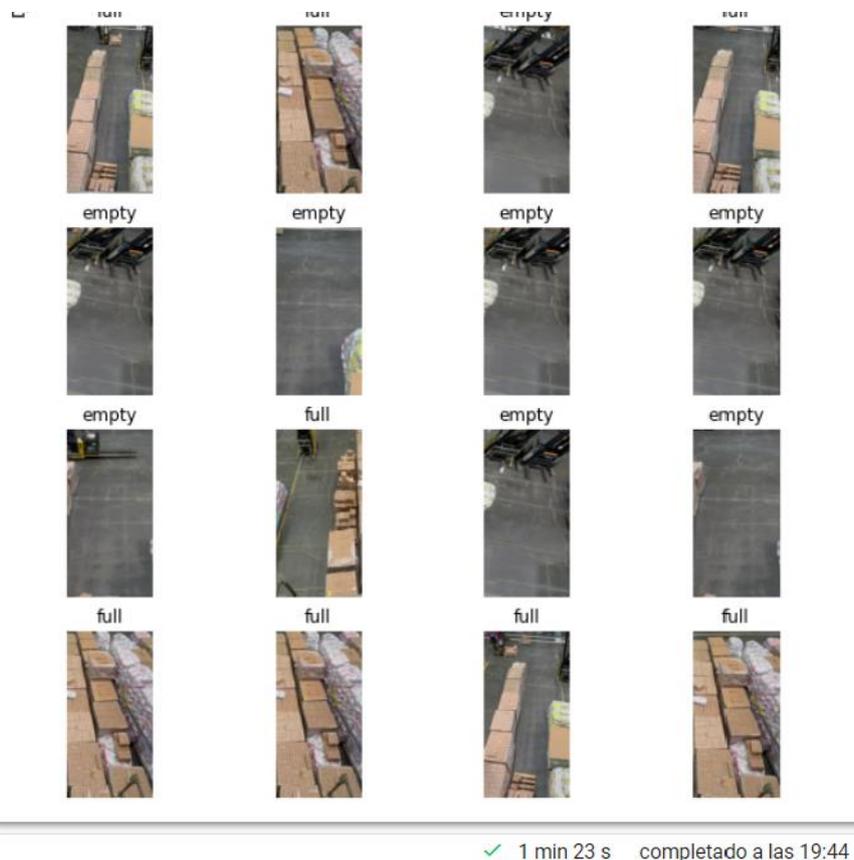
Adicionalmente, se carga cuantas clases tiene el Dataset. Para esta configuración se tienen las dos clases anteriormente mencionadas, **empty** y **full**, solo dos tipos de datos, **si la imagen es un andén libre es empty** y **si la imagen es un andén ocupado es full**. Adicional se realiza un fragmento de código usando la librería de numpy y matplotlib, para poder coger una muestra de 16 datos aleatorios y mostrarlos de manera visual con sus clases, verificando que el Dataset está correctamente clasificado inicialmente.

```
[5] class_names = train_ds.class_names
    print(class_names)

['empty', 'full']

[6] plt.figure(figsize=(10, 10))
    for images, labels in train_ds.take(1):
        for i in range(16):
            ax = plt.subplot(4, 4, i + 1)
            plt.imshow(images[i].numpy().astype("uint8"))
            plt.title(class_names[labels[i]])
            plt.axis("off")
```

**Figura 30: Creación de la variable class\_names y código para muestra de 16 datos aleatorios del Dataset**



**Figura 31: Muestra de 16 datos aleatorios del Dataset con sus clases respectivas**

Se puede visualizar que al ejecutar el fragmento de código mencionado previamente se muestran 16 datos aleatorios, que son imágenes de andenes libres y ocupados, con sus respectivas etiquetas o clases correspondientes.

En la detección de un andén libre es **requerimiento del cliente** que no se tomen en cuenta si justamente en ese frame obtenido por la cámara se encuentre en el andén un montacarga o pallets vacíos, esto no significa que el andén está ocupado, por otro lado, se lo debe clasificar como un andén libre o empty, ya que ese andén no tiene ningún pallet con producto.

### 2.4.3 Algoritmo de predicción y modelo de IA usando Deep Learning

Para poder entrenar el modelo de red neuronal profunda utilizando técnicas de deep learning se debe tener claro como estará estructurada la red, cuantas capas y neuronas tendrá. Para aquello se hace un análisis profundo de las redes neuronales más usadas en diferentes inteligencias artificiales hoy en día, además la capa de entrada siempre es una matriz de pixeles, en donde se determina que la red neuronal aceptará imágenes de 400 x 200 pixeles, es decir la capa de entrada a la red neuronal tiene 80 mil neuronas.

A continuación, se muestra el tamaño de batch o lote que toma el modelo en cada iteración, en este caso es 32, es decir el modelo toma un lote de 32 imágenes al azar en cada iteración para entrenarse y ajustar los valores en la red neuronal, además el canal que se utiliza es 3 es decir RGB, ya que los colores son claves, hay gran diferencia en colores con respecto a un andén disponible a uno que este ocupado, es por esto que el parámetro de colores es muy importante para el entrenamiento.

```
[ ] for image_batch, labels_batch in train_ds:
    print(image_batch.shape)
    print(labels_batch.shape)
    break

(32, 400, 200, 3)
(32,)
```

**Figura 32: muestra del batch, pixeles de los datos y canal**

En base a lo anterior, se crea la red neuronal convolucional profunda, usando deep learning. Se crea un modelo sequential o secuencial que lo que realiza en

términos cortos es poner cada capa creada para la red una detrás de otra, hasta la última capa.

La capa de entrada como ya se había mencionado acepta imágenes de 400 x 200 píxeles y por 3 canales (RGB), detrás de esta se tienen las capas convolucionales, estas capas siempre realizan una convolución de los datos y un pooling (en términos de programación), pero en breves palabras básicamente estas capas encuentran rasgos característicos específicos y puntuales de las imágenes, que son muy valiosos para el entrenamiento, y el pooling transforma esas características en números. Se tienen 3 capas convolucionales de 16, 32 y 64 neuronas respectivamente.

Luego de estas se tiene una capa de flatten, básicamente esta ayuda a aplanar los datos que estaban en 2D o 2 dimensiones a una sola dimensión, para que luego se integre una capa adicional de clasificación de datos la cual es dense de 128 neuronas.

Por último, se tiene la capa de salida, la cual es una capa de clasificación y se le pasa la cantidad de clases que tiene el Dataset, en este caso son 2 clases, por ende, la última capa tendrá 2 neuronas, ya que las predicciones que haga el modelo de IA posterior a su entrenamiento solo tendrán dos posibilidades, empty o full, respectivamente para un andén libre u ocupado.

```
[ ] num_classes = len(class_names)

model_unilever = Sequential([
    layers.Rescaling(1./255, input_shape=(400, 200, 3)),
    layers.Conv2D(16, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(32, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(64, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dense(num_classes)
])
```

**Figura 33: Creación de la red neuronal convolucional con sus diferentes capas**

Posterior a crear el modelo, se lo compila con su optimizador correspondiente a redes neuronales convolucionales, junto a una función de pérdida, la cual indica y ayuda a ajustar los datos de la red neuronal, y por último en las métricas se tiene el accuracy, este parámetro ayuda a saber que óptima es la red neuronal. Luego se muestra un summary o resumen de toda la red para entenderla de mejor manera.

```
[ ] model_unilever.compile(optimizer='adam',
                           loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
                           metrics=['accuracy'])

[ ] model_unilever.summary()
```

**Figura 34: Compilación del modelo y el summary del mismo**

Model: "sequential"

Layer (type)	Output Shape	Param #
rescaling (Rescaling)	(None, 400, 200, 3)	0
conv2d (Conv2D)	(None, 400, 200, 16)	448
max_pooling2d (MaxPooling2D)	(None, 200, 100, 16)	0
conv2d_1 (Conv2D)	(None, 200, 100, 32)	4640
max_pooling2d_1 (MaxPooling2D)	(None, 100, 50, 32)	0
conv2d_2 (Conv2D)	(None, 100, 50, 64)	18496
max_pooling2d_2 (MaxPooling2D)	(None, 50, 25, 64)	0
flatten (Flatten)	(None, 80000)	0
dense (Dense)	(None, 128)	10240128
dense_1 (Dense)	(None, 2)	258

=====  
 Total params: 10,263,970  
 Trainable params: 10,263,970  
 Non-trainable params: 0

**Figura 35: Resultado del summary(), es el detalle de la red neuronal convolucional, con sus capas respectivas**

En la figura 35, se puede leer y visualizar de mejor manera toda la red neuronal profunda anteriormente ya detallada, y al final refleja que el número de parámetros a ajustarse y entrenarse son más de 10 millones, exactamente son 10,263,970. Con estos datos se puede estar seguro que al entrenar al modelo se van a obtener buenos resultados, los cuales se detallan en el **capítulo 3**.

#### 2.4.4 Algoritmo optimizado usando Deep Learning

Para poder optimizar la red neuronal, se puede usar una de las técnicas de deep learning más usadas actualmente, y se llama **Dropout**, la cual consiste en desconectar neuronas al azar, para que toda la red neuronal se entrene de forma uniforme. No van a existir neuronas más entrenadas que otras, en cambio todas las neuronas estarán entrenadas de una forma uniforme y optima.

Esta capa de Dropout siempre y comúnmente se la integra una capa antes de la final, y se le da un parámetro de 0.2, el cual es un estándar en redes neuronales, esto quiere decir que cada neurona en esa capa tiene un 20% de posibilidades que no sea escogida para el entrenamiento, y vaya a otra en cada iteración.

```
[ ] model_unilever_dropout = Sequential([
    layers.Rescaling(1./255, input_shape=(400, 200, 3)),
    layers.Conv2D(16, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(32, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(64, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dropout(0.2),
    layers.Dense(num_classes)
])

[ ] model_unilever_dropout.compile(optimizer='adam',
    loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
    metrics=['accuracy'])
```

**Figura 36: Mismo modelo de red neuronal previamente detallado, pero integrada la capa de Dropout antes de la última.**

Los detalles y resultados obtenidos y mejorados con esta optimización de la red neuronal se detallan en el **capítulo 3**.

# CAPÍTULO 3

## 3. RESULTADOS Y ANÁLISIS

De acuerdo con el alcance y objetivos planteados en el proyecto integrador, en el presente capítulo se evidencian los resultados y análisis obtenidos a partir del desarrollo y simulación del sistema SCADA con conexión mediante OPC UA para el envío de datos, el cual es el sistema integral de monitoreo y control para la automatización de 8 andenes del centro nacional de la multinacional, a través del diseño del sistema SCADA conectado al sistema de inteligencia artificial simulado, el cual es la red neuronal convolucional profunda. También se analiza el rendimiento del sistema en términos de escalabilidad y capacidad de adaptarse a cambios en el entorno con un adjunto de un conjunto de pasos a confirmarse para su aprobación de correcto funcionamiento en sitio.

### 3.1 Programación PLC y Comunicación Industrial – Simulación

Se consiguió establecer la conexión entre el sistema que comanda las rutinas, en este caso se utiliza el PLC 1513-1 PN, con el PC, y las pantallas HMI. La comunicación establecida fue a través de Profinet, donde el estado de compilación y descarga del programa se realizó satisfactoriamente.

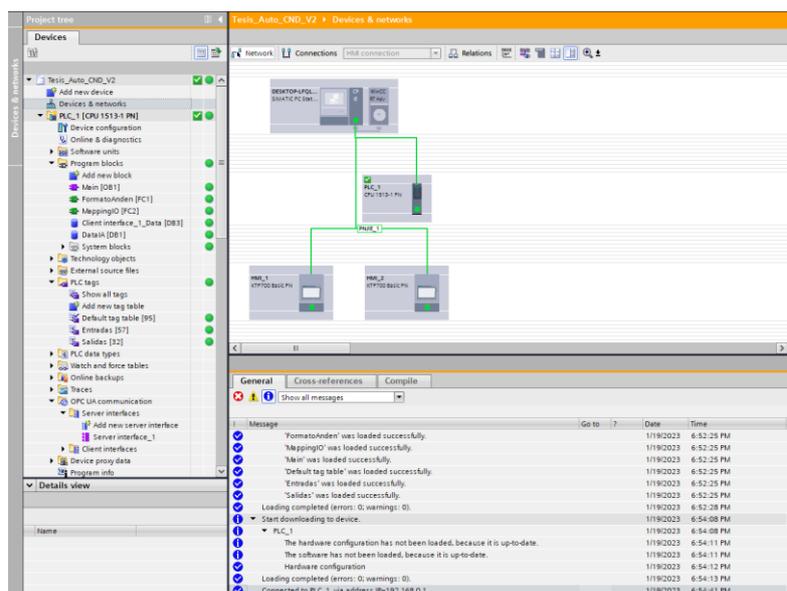
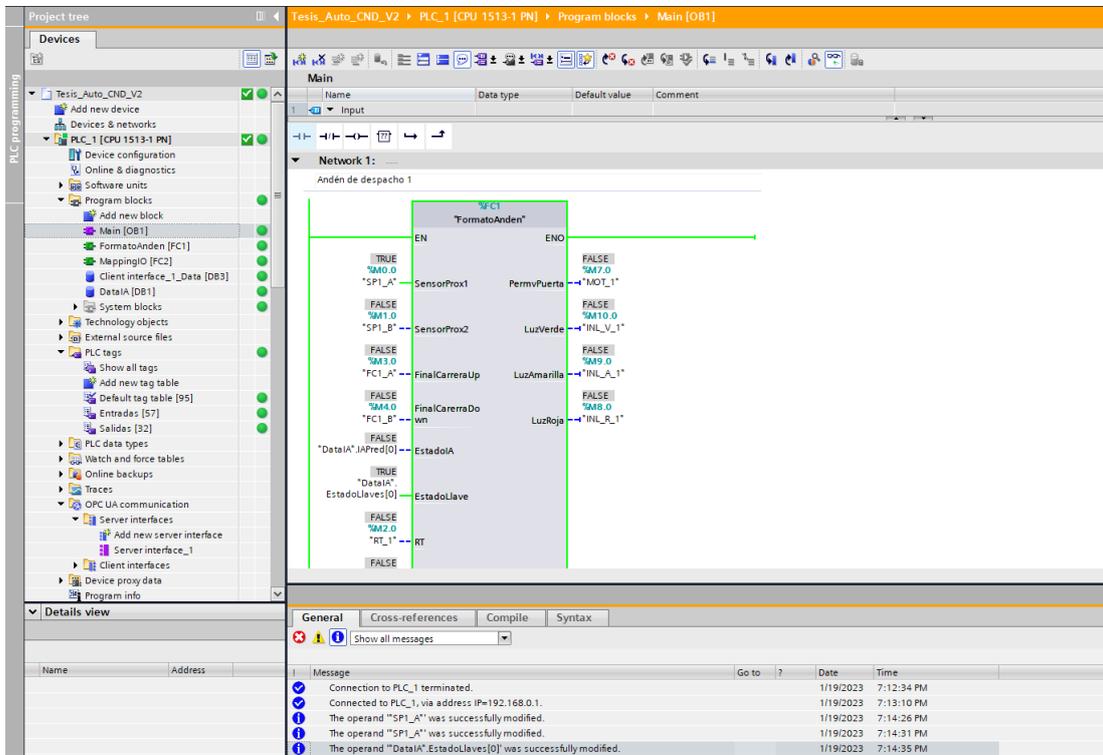


Figura 37: Simulación de la red profinet IE 1

Adicional a esto, se puede ver en la siguiente figura 38, la ejecución del programa exitosamente y los permisos empleados para la habilitación de las señales de salida.



**Figura 38: Compilación y ejecución de bloque main**

En la tabla de variables se evidencia la respuesta que se obtiene al momento de presentar los casos en las variables de entrada con la respuesta obtenida, modificando los valores de salida. En la figura 39 se presentan una de las dos tablas de variables creadas en el proyecto.

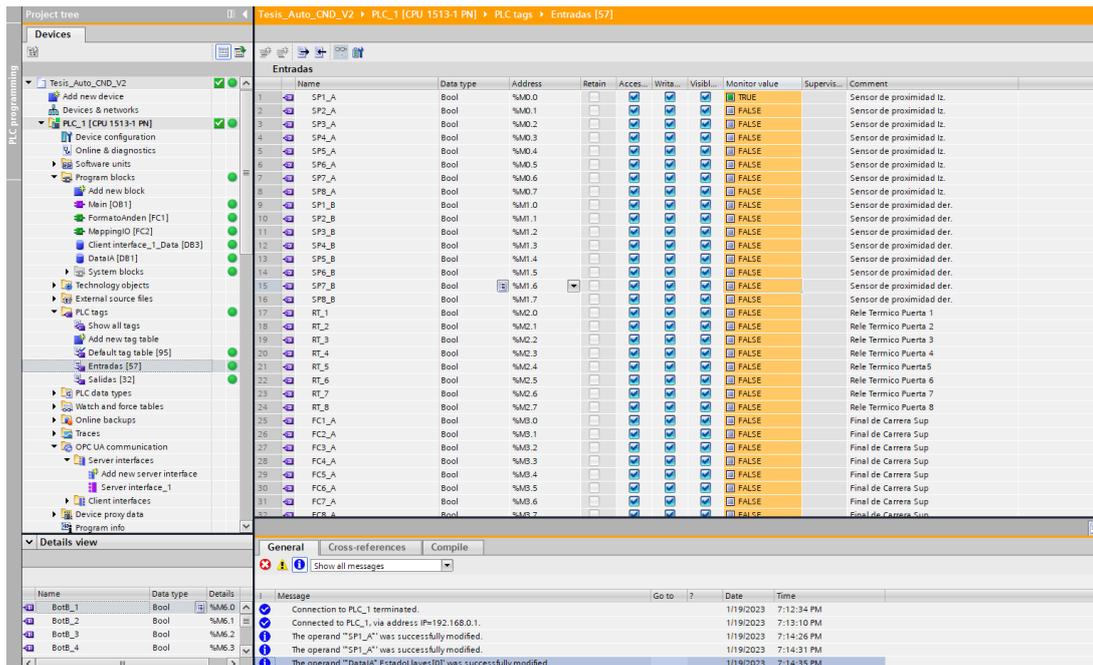


Figura 39: Tabla de variables enlazadas al SCADA

### 3.2 Ejecución del Runtime SCADA

Para el diseño y ejecución del runtime del sistema SCADA implementado en WINCC se vincula las variables del proceso que maneja el PLC y se las observan en las pantallas implementadas. En las siguientes imágenes se puede ver la exitosa compilación del sistema SCADA y la generación del runtime que permite visualizar las pantallas creadas para englobar el sistema.

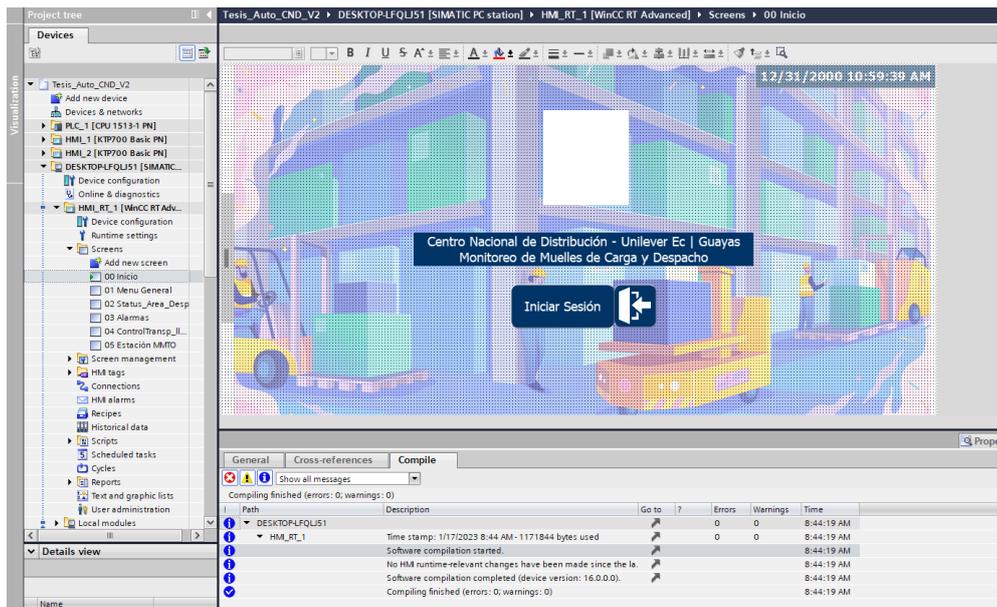
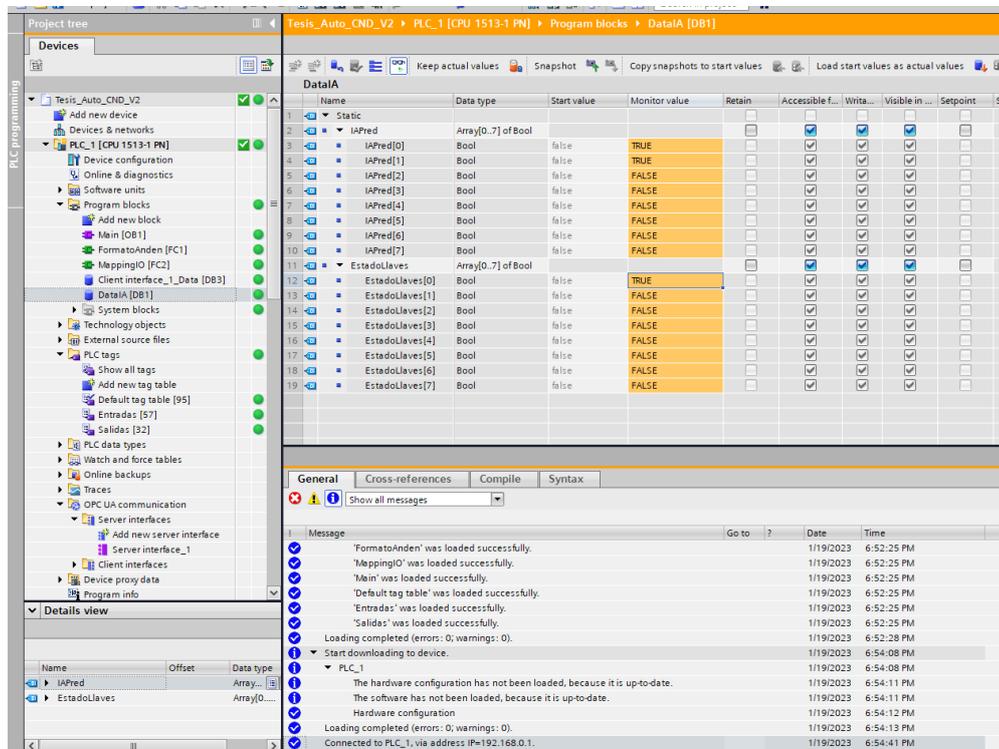


Ilustración 40: Compilación del sistema SCADA

En la siguiente figura 41, se puede comprobar el estado de los bloques de datos creados para el manejo de las señales entrantes de la predicción de la IA así como los datos habilitadores para el control de las llaves.



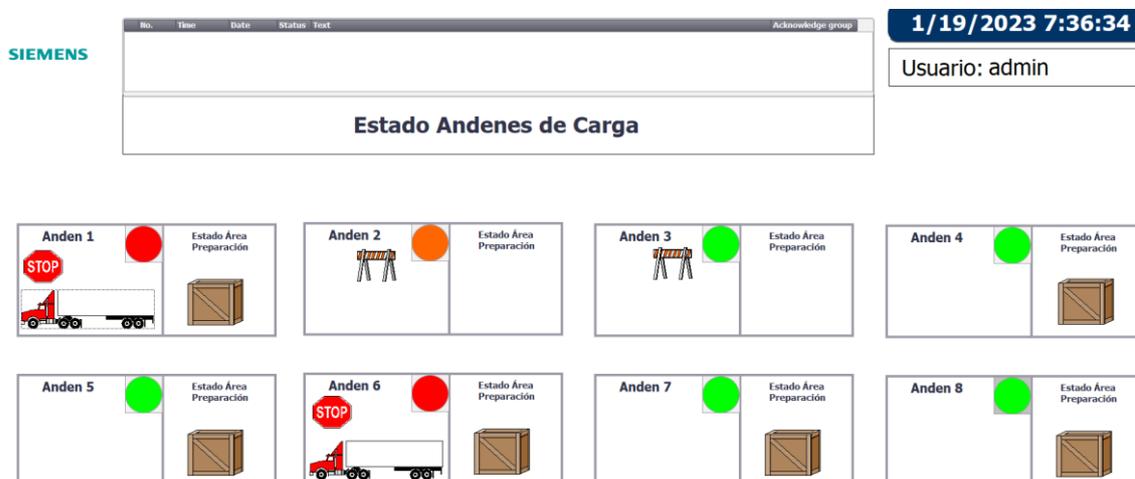
**Figura 41: Comprobación del envío de datos desde el código en Python al SCADA**

En la siguiente figura 42, se puede apreciar el control de accesos a través de usuarios y dependiendo de los usuarios asignados es posible acceder a varias configuraciones del sistema, es decir, varias pantallas del sistema.



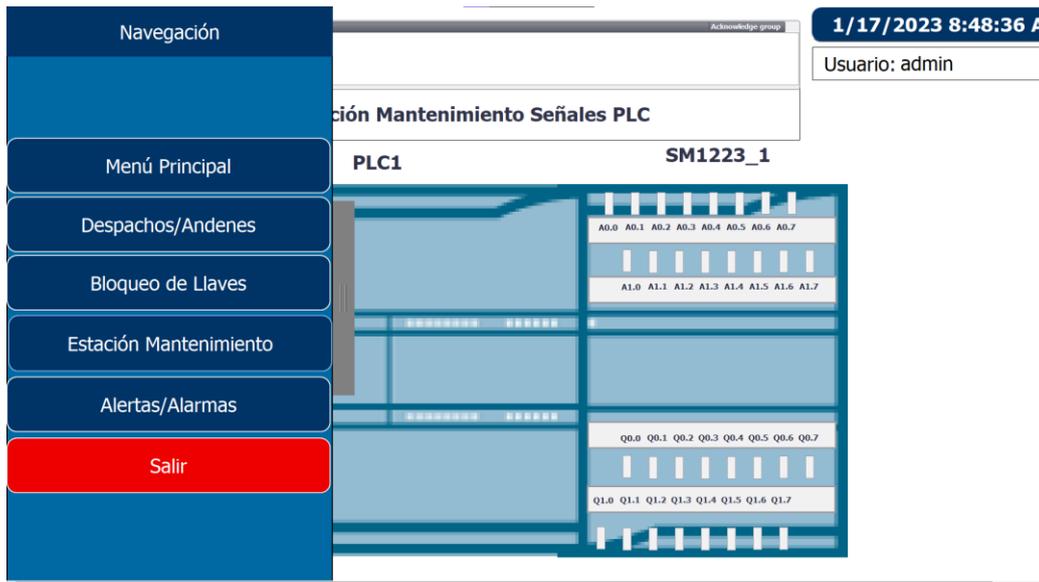
**Figura 42: Ejecución del runtime – pantalla principal**

En la siguiente figura 43, se logra visualizar el detalle de la rutina de señales que posee cada andén, el cual presenta varias señales de avisos en donde indicarán estar en proceso de carga del andén



**Figura 43: Pantalla de visualización del estado de andenes (8 andenes)**

En la siguiente pantalla, figura 44 presenta la visualización de las físicas que salen y llegan a cada uno de los puertos y módulos del PLC, el cual nos permite tener una estación para la resolución de problemas en términos eléctricos.



**Figura 44: Ejecución del runtime – pantalla de señales del PLC**

A continuación, se presenta una lista de verificación para pruebas de aceptación del cliente de sistemas SCADA con servidor de código en Python con comunicación OPC UA y sistema de monitoreo por cámaras a contemplarse en sitio:

**Tabla 1 Checklist para pruebas de aceptación del cliente**

	Verificar que el sistema SCADA esté configurado y funcionando correctamente.
	Verificar que los puntos de datos sean correctamente recolectados y registrados.
	Verificar que las alarmas y los eventos sean correctamente generados y manejados.
	Verificar la integridad de los datos y la capacidad del sistema para recuperarse de fallos.
	Verificar la capacidad del sistema para cumplir con los requisitos de seguridad.
	Verificar que el sistema cumpla con las regulaciones y normas aplicables.
	Verificar que la interfaz del usuario sea fácil de usar y cumpla con los requisitos del cliente.

	Realizar pruebas de rendimiento y verificar que el sistema cumpla con los requisitos de capacidad y escalabilidad.
	Verificar la compatibilidad del sistema con los sistemas existentes en el cliente.
	Verificar que se hayan proporcionado toda la documentación y el soporte necesarios para el uso y mantenimiento del sistema.
	Verificar que el servidor de código en Python esté configurado y funcionando correctamente.
	Verificar que el servidor esté correctamente conectado y comunicándose con el sistema SCADA mediante OPC UA.
	Verificar que el código en Python esté funcionando correctamente y cumpliendo con los requerimientos del proyecto.
	Verificar que el sistema de monitoreo por cámaras esté configurado y funcionando correctamente.
	Verificar que las cámaras estén correctamente conectadas y grabando imágenes.
	Verificar que el sistema de monitoreo por cámaras sea capaz de grabar y almacenar imágenes adecuadamente y cumplir con los requerimientos de resolución y calidad.
	Verificar que el sistema de monitoreo por cámaras sea capaz de generar alarmas en caso de detección de eventos específicos.

### 3.3 Sistema de Visión Artificial o por Computadora

Para predecir de manera óptima y con tecnologías actuales el estado de los andenes en el centro logístico se desarrolló y entrenó dos modelos de redes neuronales convolucionales profundas usando técnicas de Deep Learning, en las cuales se evidencia diferencias entre los resultados de ambas redes, dando así un modelo mejor que otro, específicamente el modelo usando la técnica de Dropout ya detallada anteriormente.

#### 3.3.1 Ejecución de la red neuronal convolucional profunda inicial

En este apartado se da la ejecución del algoritmo detallado en el capítulo 2, apartado 2.4.3. A continuación se puede visualizar el entrenamiento que tiene la red neuronal, el cual recorre todo el Dataset durante 4 épocas distintas, se define este número de épocas, ya que se llega a la conclusión que con tal número no se tendrá poco entrenamiento, y tampoco se tendrá sobre entrenamiento, la red neuronal esta equilibrada con respecto a esto.

```
[ ] epochs=4
    history = model_unilever.fit(
        train_ds,
        validation_data=val_ds,
        epochs=epochs
    )

Epoch 1/4
25/25 [=====] - 197s 8s/step - loss: 0.7488 - accuracy: 0.8419 - val_loss: 0.0151 - val_accuracy: 0.9950
Epoch 2/4
25/25 [=====] - 4s 142ms/step - loss: 0.0229 - accuracy: 0.9937 - val_loss: 0.0061 - val_accuracy: 1.0000
Epoch 3/4
25/25 [=====] - 4s 141ms/step - loss: 0.0051 - accuracy: 0.9987 - val_loss: 0.0013 - val_accuracy: 1.0000
Epoch 4/4
25/25 [=====] - 4s 140ms/step - loss: 0.0028 - accuracy: 0.9987 - val_loss: 4.5020e-04 - val_accuracy: 1.0000
```

**Figura 45: Entrenamiento de la red neuronal convolucional inicial**

En la figura 45 se visualiza que a medida que la red neuronal se está entrenando finalmente llega a una precisión en el ajuste de 0.9987, en la cual su escala va de 0 a 1, y a una pérdida de datos de 0.0028, que básicamente es 0, en donde se puede analizar que la red neuronal convolucional en este caso es óptima para su uso.

A continuación, también se tiene un fragmento de código realizado con matplotlib y numpy para visualizar y analizar de mejor manera las etapas de entrenamiento y validación que tuvo la red neuronal con respecto al tiempo y épocas transcurridas para la preparación del modelo.

```
[ ] acc = history.history['accuracy']
    val_acc = history.history['val_accuracy']

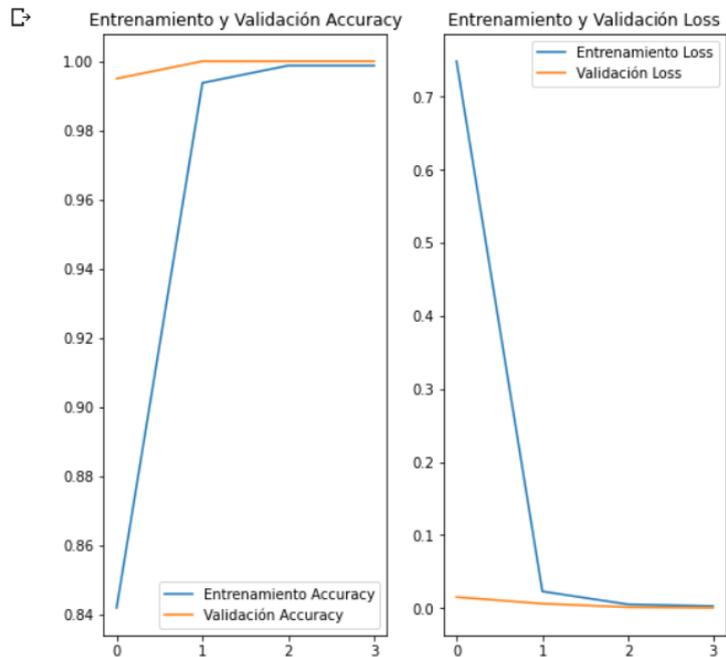
    loss = history.history['loss']
    val_loss = history.history['val_loss']

    epochs_range = range(epochs)

    plt.figure(figsize=(8, 8))
    plt.subplot(1, 2, 1)
    plt.plot(epochs_range, acc, label='Entrenamiento Accuracy')
    plt.plot(epochs_range, val_acc, label='Validación Accuracy')
    plt.legend(loc='lower right')
    plt.title('Entrenamiento y Validación Accuracy')

    plt.subplot(1, 2, 2)
    plt.plot(epochs_range, loss, label='Entrenamiento Loss')
    plt.plot(epochs_range, val_loss, label='Validación Loss')
    plt.legend(loc='upper right')
    plt.title('Entrenamiento y Validación Loss')
    plt.show()
```

**Figura 46: Fragmento de código hecho con numpy y matplotlib**



**Figura 47: Gráficas Entrenamiento vs Validación (Accuracy) y Entrenamiento vs Validación (Loss)**

En la figura 47 se puede visualizar a la izquierda la gráfica de Entrenamiento vs Validación para el accuracy o precisión de la red neuronal. El modelo parte de un valor bajo en la etapa de entrenamiento (línea azul), pero a medida que el tiempo y las épocas pasan, la precisión llega al valor mencionado de 0.9987, un valor muy bueno en el campo de redes neuronales. Así mismo en la etapa de validación (línea naranja), es cuando ya el modelo prueba los resultados de su entrenamiento, vemos que parte de un valor más alto que cuando se estaba entrenando, porque la red neuronal en este punto ya adquirió conocimiento, y finalmente llega a una precisión aproximada a la del entrenamiento, en la cual ambas convergen en un punto próximo. Dando, así como resultado que la precisión del modelo es altamente confiable para su puesta en marcha.

De igual forma también se puede visualizar en la figura 47 la gráfica de Entrenamiento vs Validación para Loss o pérdida de datos, lo preferible siempre en redes neuronales es que este valor llegue a 0, ya que esto representa que no haya pérdida de datos en el modelo. Como podemos visualizar ambas gráficas tanto la de entrenamiento (azul) como la de validación (naranja) llegan a converger en algún punto cercano a cero. Resultando así tanto que la etapa de entrenamiento como la de validación fueron óptimas en la red neuronal convolucional.

### **3.3.2 Pruebas del modelo de red neuronal convolucional profunda inicial**

El modelo de red neuronal profunda ya está listo, por ende, hay que probarlo en situaciones reales o imágenes o frames distintos. Para esto se sabe que la red neuronal solo acepta imágenes de entrada de 400 x 200 píxeles, por ende, toda imagen captada por la cámara de seguridad hay que cambiarle el tamaño o redimensionarla, y es aquí donde se usa la librería OpenCV nuevamente. A continuación, se muestra el código para la tarea mencionada.

```

def resize(image, width=None, height=None, inter=cv2.INTER_AREA):
    # Extraemos las dimensiones originales.
    (original_height, original_width) = image.shape[:2]

    if width is None and height is None:
        return image
    if width is None:
        ratio = height / float(original_height)
        width = int(original_width * ratio)
    else:
        ratio = width / float(original_width)
        height = int(original_height * ratio)
    new_size = (width, height)

    return cv2.resize(image, new_size, interpolation=inter)

image_path = '/content/drive/MyDrive/data/validation/full/20221110_133746977_i05 1.jpg'

image2=cv2.imread(image_path)
resized = cv2.resize(image2, (200,400), interpolation=cv2.INTER_AREA)

```

**Figura 48: Código con OpenCV para redimensionar un frame o imagen**

Una vez se tiene la imagen del andén o frame redimensionado de entrada a la red neuronal, se puede intentar predecir qué tipo de imagen es, si es un andén libre correspondiente a la clase empty o si es un andén ocupado, correspondiente a la clase full.

En la figura 49, se muestra el ingreso a la red neuronal de una imagen correspondiente a un andén ocupado. Con el propósito de verificar la predicción que da el modelo, que resulta ser **full**. Este resultado es correcto dado que el andén está ocupado, con una exactitud de un **74.65%**. Se da como resultado que el modelo entrenado previamente está prediciendo de una forma correcta y con una exactitud altamente confiable.

```
[ ] input_arr = tf.keras.preprocessing.image.img_to_array(resized)
input_arr = np.array([input_arr]) # Convert single image to a batch.
predictions = model_unilever.predict(input_arr)
plt.imshow(image2)
```

```
1/1 [=====] - 0s 66ms/step
<matplotlib.image.AxesImage at 0x7f53f53225e0>
```



```
[ ] score = tf.nn.softmax(predictions[0])
print(
    "Este frame o imagen es: '{} ' con un {:.2f} % de exactitud."
    .format(class_names[np.argmax(score)], 100 * np.max(score))
)
```

```
Este frame o imagen es: 'full' con un 74.65 % de exactitud.
```

**Figura 49: Predicción de un andén ocupado o full con exactitud de 74.65%**

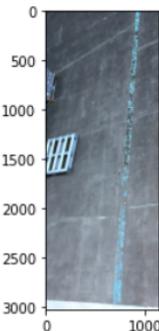
En la figura 50, por otro lado, se ingresa una imagen de un andén libre a la red neuronal, para verificar su predicción. Se puede visualizar en la figura que en el andén hay un pallet vacío, esto no quiere decir que sea un andén ocupado, ya que estas fueron las consideraciones del cliente. Como resultado del entrenamiento óptimo del modelo, se obtiene una predicción de un andén de clase empty, lo que quiere decir que es un andén libre, y con una exactitud del 100%, que es correcta.

```
[ ] image_path = '/content/drive/MyDrive/data/validation/empty/20221110_140835205_iOS 2.jpg'

image2=cv2.imread(image_path)
resized = cv2.resize(image2, (200,400), interpolation=cv2.INTER_AREA)

input_arr = tf.keras.preprocessing.image.img_to_array(resized)
input_arr = np.array([input_arr]) # Convert single image to a batch.
predictions = model_unilever.predict(input_arr)
score = tf.nn.softmax(predictions[0])
print(
    "Este frame o imagen es: '{} con un {:.2f} % de exactitud."
    .format(class_names[np.argmax(score)], 100 * np.max(score))
)
plt.imshow(image2)

1/1 [=====] - 0s 16ms/step
Este frame o imagen es: 'empty' con un 100.00 % de exactitud.
<matplotlib.image.AxesImage at 0x7f79b6717e20>
```



**Figura 50: Predicción de un andén libre o empty con exactitud del 100%**

### 3.3.3 Ejecución de la red neuronal convolución profunda optimizada y principal

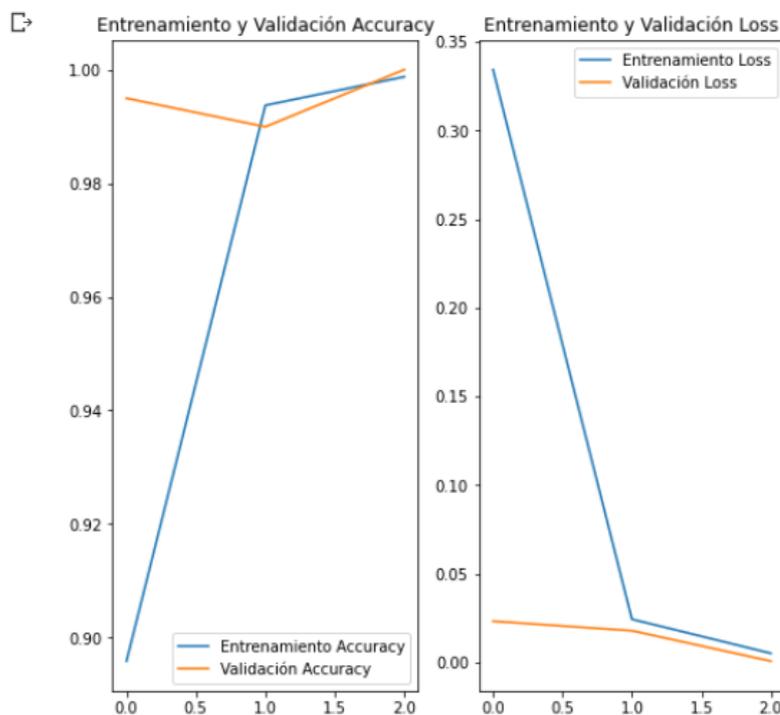
En este apartado se da la ejecución del algoritmo detallado en el capítulo 2, apartado 2.4.4. A continuación se puede visualizar el entrenamiento que tiene la red neuronal optimizada por la técnica Dropout de deep learning. De igual manera el modelo en esta ocasión recorre durante 3 épocas distintas todo el Dataset para su entrenamiento y validación.

```
[ ] epochs=3
history = model_unilever_dropout.fit(
    train_ds,
    validation_data=val_ds,
    epochs=epochs
)

Epoch 1/3
25/25 [=====] - 5s 148ms/step - loss: 0.3343 - accuracy: 0.8959 - val_loss: 0.0232 - val_accuracy: 0.9950
Epoch 2/3
25/25 [=====] - 4s 138ms/step - loss: 0.0243 - accuracy: 0.9937 - val_loss: 0.0179 - val_accuracy: 0.9899
Epoch 3/3
25/25 [=====] - 4s 142ms/step - loss: 0.0051 - accuracy: 0.9987 - val_loss: 7.1689e-04 - val_accuracy: 1.0000
```

**Figura 51: Entrenamiento de la red neuronal convolucional optimizada principal**

En la figura 51 se puede visualizar que a medida que la red neuronal se está entrenando finalmente llega a una precisión en el ajuste de 0.9987 y a una pérdida de datos alrededor de cero. Además, en la etapa de validación llega a una precisión de 0.9899 y casi 1, en donde podemos observar que el modelo mejora, y también la pérdida de datos decrece.



**Figura 52: Gráficas Entrenamiento vs Validación (Accuracy) y Entrenamiento vs Validación (Loss) | Para el modelo optimizado**

En la figura 52, se pueden observar las mismas gráficas que se mostraron con el modelo de red neuronal inicial, pero ahora son del modelo optimizado, en donde se puede evidenciar que tanto en la etapa de entrenamiento como en la etapa de validación para el accuracy o ajustamiento de datos mejora significativamente. Dando como resultado que la validación es ligeramente mayor que los resultados obtenidos en el entrenamiento. Esto quiere decir que el modelo obtuvo mejor respuesta cuando entro a la etapa de validación, lo cual es muy importante en el proceso.

Asimismo, en la gráfica entrenamiento vs validación (loss), se evidencia que la respuesta que obtiene la etapa de validación es mejor que la del entrenamiento, llegando a un valor más próximo a cero con respecto a la pérdida de datos en la validación. Se obtiene como resultado que el modelo optimizado con la técnica de deep learning Dropout mejora al modelo de red neuronal entrenado inicialmente.

#### **3.3.4 Pruebas del modelo de red neuronal convolución profunda optimizada y principal**

El modelo de red neuronal profunda optimizada y principal para el proyecto ya está listo, por ende, hay que probarlo en situaciones reales o imágenes o frames distintos como se lo hizo para el modelo inicial. Para esto se sabe de igual forma que la red neuronal solo acepta imágenes de entrada de 400 x 200 pixeles, por ende, antes de mandar a predecir el estado de un andén, hay que redimensionar la imagen obtenida a los pixeles ya mencionados como se lo hizo en el modelo inicial.

En la figura 53, se muestra el ingreso a la red neuronal optimizada la misma imagen correspondiente a un andén ocupado que se predijo con una exactitud del **74,65%** por parte de la red neuronal inicial, en cambio esta red optimizada de igual forma la predice correctamente ya que es **full**, pero con una exactitud del **75.95%**, la exactitud de predicción **augmentó en un 1,3%** con respecto a la red inicial. Con

esto se da como resultado que la red optimizada obtiene mejores predicciones que la red inicial.

```
[ ] image_path = '/content/drive/MyDrive/data/validation/full/20221110_133746977_i05_1.jpg'

image2=cv2.imread(image_path)
resized = cv2.resize(image2, (200,400), interpolation=cv2.INTER_AREA)

input_arr = tf.keras.preprocessing.image.img_to_array(resized)
input_arr = np.array([input_arr]) # Convert single image to a batch.
predictions = model_unilever_dropout.predict(input_arr)
score = tf.nn.softmax(predictions[0])
print(
    "Este frame o imagen es: '{} ' con un {:.2f} % de exactitud."
    .format(class_names[np.argmax(score)], 100 * np.max(score))
)

plt.imshow(image2)
```

WARNING:tensorflow:5 out of the last 9 calls to <function Model.make\_predict\_function.<locals>.<lambda> at 0x7f799fca1d00> are taking more than 0.5 seconds. The cause of the slowdown has not been identified. Speeding it up may be possible by reporting the issue at https://www.tensorflow.org/tutorials/troubleshooting/#tf.function-call-takes-longer-than-0.5s

1/1 [=====] - 0s 67ms/step  
Este frame o imagen es: 'full' con un 75.95 % de exactitud.  
<matplotlib.image.AxesImage at 0x7f799fca1d00>



**Figura 53: Predicción por parte de la red optimizada de un andén ocupado o full con exactitud de 75.95%**

En la figura 54, ahora se ingresa una imagen de un andén libre a la red neuronal, optimizada para verificar su predicción. Se puede visualizar que en el andén existe un montacarga, lo cual fue un requerimiento del cliente que si existe algún montacarga estacionado por error y no hayan pallets con producto alrededor, el andén se denomine como libre.

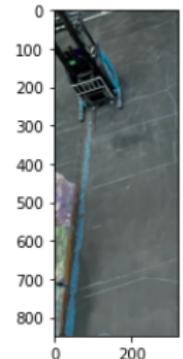
En el entrenamiento al modelo se le paso muchos ejemplos en situaciones distintas, por lo que se espera que lo prediga correctamente, y es lo que se evidencia. El modelo predice la imagen como empty, lo cual quiere decir que es un andén libre, con una exactitud del 100%.

```
[ ] #image_path = '/content/drive/MyDrive/data/validation/empty/20221110_140835205_iOS 2.jpg'
image_path = '/content/drive/MyDrive/data/validation/empty/frame_0001136.jpg'

image2=cv2.imread(image_path)
resized = cv2.resize(image2, (200,400), interpolation=cv2.INTER_AREA)

input_arr = tf.keras.preprocessing.image.img_to_array(resized)
input_arr = np.array([input_arr]) # Convert single image to a batch.
predictions = model_unilever_dropoutPrueba.predict(input_arr)
score = tf.nn.softmax(predictions[0])
print(
    "Este frame o imagen es: '{} ' con un {:.2f} % de exactitud."
    .format(class_names[np.argmax(score)], 100 * np.max(score))
)
plt.imshow(image2)

1/1 [=====] - 0s 136ms/step
Este frame o imagen es: 'empty' con un 100.00 % de exactitud.
<matplotlib.image.AxesImage at 0x7f99cd605670>
```



**Figura 54: Predicción por parte de la red optimizada de un andén libre o empty con exactitud de 100%**

### 3.3.5 Pruebas finales del modelo de Deep Learning optimizado principal

Se escoge como modelo principal para el proyecto la red neuronal optimizada, porque como se pudo evidenciar en los apartados anteriores esta red tiene mejor respuesta en precisión que la red neuronal inicial. A continuación, se muestra el fragmento de código realizado con la librería numpy y matplotlib para la visualización de una matriz de 25 datos o imágenes aleatorias de diferentes andenes con las respectivas predicciones que da el modelo de red neuronal optimizado.

```

[11] import numpy as np
for imagenes_prueba, etiquetas_prueba in val_ds.take(1):
    imagenes_prueba = imagenes_prueba.numpy()
    etiquetas_prueba = etiquetas_prueba.numpy()
    predicciones = model_unilever_dropoutPrueba.predict(imagenes_prueba)

def graficar_imagen(i, arr_predicciones, etiquetas_reales, imagenes):
    arr_predicciones, etiqueta_real, img = arr_predicciones[i], etiquetas_reales[i], imagenes[i]
    plt.grid(False)
    plt.xticks([])
    plt.yticks([])
    plt.imshow(img[...,:0])

    etiqueta_prediccion = np.argmax(arr_predicciones)

    if etiqueta_prediccion == etiqueta_real:
        color = 'blue'
    else:
        color = 'red'

    plt.xlabel("{} \n predicción:({})".format(class_names[etiqueta_prediccion],
                                             class_names[etiqueta_real]),
              color=color)

filas = 5
columnas = 5
num_imagenes = filas*columnas
plt.figure(figsize=(2*columnas, 12))
for i in range(num_imagenes):
    plt.subplot(filas, 2*columnas, 2*i+1)
    graficar_imagen(i, predicciones, etiquetas_prueba, imagenes_prueba)

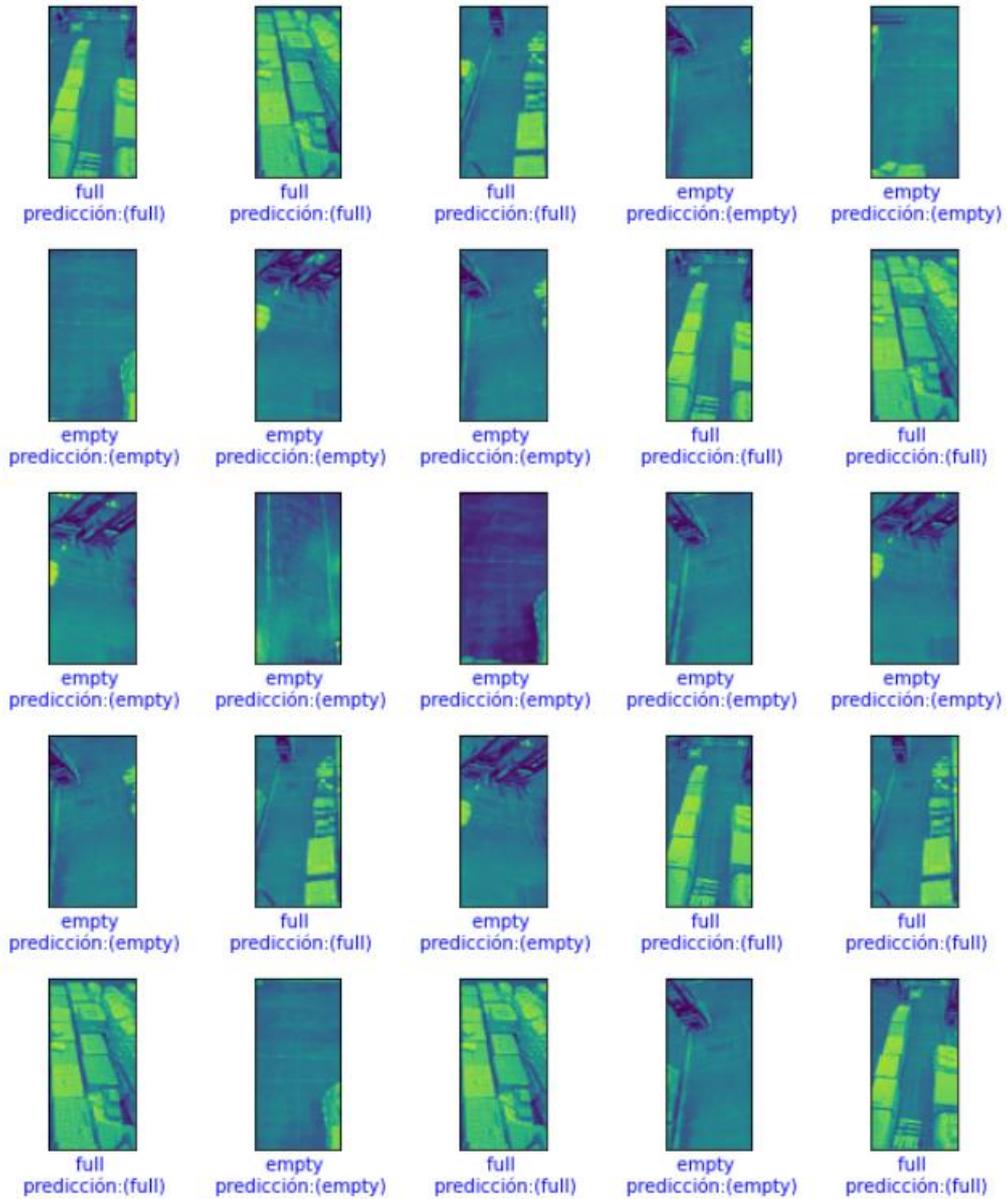
```

**Figura 55: Fragmento de código para la visualización de una matriz de 25 datos con sus respectivas predicciones**

En la figura 56, se puede visualizar el resultado del código presentado en la figura 55. Se observa una matriz 5x5, 25 datos o imágenes de andenes en diferentes situaciones, a debajo de cada imagen esta la etiqueta real o estado correcto del andén, y debajo de la etiqueta real se encuentra la predicción que dio el modelo de inteligencia artificial.

Se evidencia que en cada una de las 25 imágenes de la matriz el modelo predice de forma óptima el estado correcto del andén, tomando todas las consideraciones del cliente previamente detalladas. Gracias a todas estas pruebas realizadas, finalmente se da como resultado que el modelo de la red neuronal optimizada esta

apta para su uso en tiempo real a través de las cámaras de seguridad del centro logístico de la multinacional.



**Figura 56: Visualización de la matriz de 25 imágenes con sus respectivas etiquetas reales y predicciones por la IA optimizada**

# CAPÍTULO 4

## 4. CONCLUSIONES Y RECOMENDACIONES

### Conclusiones

- La simulación y el monitoreo detallado del sistema logístico mediante a las herramientas de automatización permitió obtener información valiosa para la mejora continua, aumentando la eficiencia en los despachos y optimizando la gestión de la logística al utilizar un proceso de despacho.
- La implementación de herramientas de automatización con WinCC SCADA runtime permitió un monitoreo y control efectivo del sistema logístico, al mejorar parámetros de seguridad industrial de los andenes logísticos.
- Se desarrolló de forma óptima la rutina de despacho de ordenes utilizando el lenguaje de programación Ladder y bloques de funciones que permitieron el monitoreo de las señales de entrada y el control sobre las señales de salida reflejándose la programación sobre las pantallas del SCADA.
- Se demostró la viabilidad de la integración de ambos sistemas, el sistema SCADA con el sistema de inteligencia artificial, y se observó en lo evidenciado que combinar tecnologías convencionales junto con tecnologías actuales e inteligencia artificial son puntos muy diferenciales en la industria, además que son sistemas de gran escalabilidad y fácil mantenimiento. La automatización industrial junto con el área computacional de las IA da camino a soluciones no antes vistas en el mercado, que aportan significativamente al control y monitoreo generalizado de diversas áreas.
- Tras el análisis en la IA se puede concluir y deducir que la red neuronal convolucional desarrollada con técnicas de Deep Learning, es óptima y apta para su posterior puesta en marcha. La cual aporta significativamente al proyecto ya que la IA ataca la problemática de la disponibilidad de los

andenes, por ende, los colaboradores ya no tendrán que recorrer personalmente el centro logístico para comprobar esa disponibilidad, todo estará controlado y monitorizado desde la oficina central.

- Gracias a todo lo anterior expuesto, se concluye que el proyecto contribuye significativamente al ODS planteado anteriormente, *“Industria, Innovación e Infraestructura”*, además de correlacionarse con el propósito de vida de la compañía, el cual es *“Hacer de la sustentabilidad algo cotidiano”*. Con esto demostramos que tener un propósito tecnológico y orientado al futuro ofrece rendimientos excepcionales.

### **Recomendaciones**

- Para futuras implementaciones o mejoras en el sistema se recomienda agregar funciones que permitan tomar métricas tanto de tiempos de despacho como de paradas de emergencia permitiendo así el uso de estos datos para aumentar la eficiencia del centro logístico.
- En caso de requerir mejorar los niveles de seguridad de la información se recomienda implementar switches administrados. Además del uso de cortafuegos.
- Se debe desarrollar el sistema de inteligencia artificial siempre en un entorno de programación donde se conozca que no habrá problemas de incompatibilidad de librerías o de diferentes frameworks, porque esto conllevará a que no presente problemas de software en el transcurso del entrenamiento del modelo de red neuronal. Se recomienda usar Google Colab.
- Es necesario cuidar siempre el comportamiento con respecto al tiempo del modelo de IA, tanto en la etapa de entrenamiento como en la de validación, se recomienda graficar este proceso para una mejor visualización de lo que está sucediendo y así poder verificar que la red neuronal no tenga insuficiente entrenamiento o por lo contrario tenga un sobre entrenamiento.

- Siempre al desarrollar proyectos industriales de este tipo hay que empatizar con el cliente y entender sus necesidades, para así poder ofrecer soluciones innovadoras que combinen varias tecnologías utilizadas actualmente y sobre todo sean soluciones sencillas pero eficaces, que el cliente las entienda de una forma óptima y que impacten de manera positiva en la operación de la empresa.

# BIBLIOGRAFÍA

## 5. BIBLIOGRAPHY

- [1] SICMA21, «¿Qué es un PLC y cuándo se usa?,» SICMA21, 14 Octubre 2021. [En línea]. Available: [https://www.sicma21.com/que-es-un-plc/#Quien\\_invento\\_los\\_PLC](https://www.sicma21.com/que-es-un-plc/#Quien_invento_los_PLC). [Último acceso: 8 Noviembre 2022].
- [2] AUTEXOPEN, «Qué es un sistema SCADA? Información completa,» AUTEXOPEN, 6 Julio 2022. [En línea]. Available: <https://www.autex-open.com/automatizacion-industrial/que-es-un-sistema-scada-informacion-completa/>. [Último acceso: 8 Noviembre 2022].
- [3] G. M. Smith, «Dewesoft d.o.o.,» Dewesoft d.o.o., 09 Marzo 2020. [En línea]. Available: <https://dewesoft.com/es/daq/que-es-un-sensor#what-do-sensors-do>. [Último acceso: 11 Noviembre 2022].
- [4] SDI, «Tipos de Sensores Industriales,» SDI, 2020. [En línea]. Available: <https://sdindustrial.com.mx/blog/tipos-de-sensores-industriales/>. [Último acceso: 10 Noviembre 2022].
- [5] G. C. GROUP, «Qué es un actuador y cómo impacta en la generación de energía,» AUTYCOM, 15 Octubre 2022. [En línea]. Available: <https://www.autycom.com/que-es-un-actuador/>. [Último acceso: 10 Noviembre 2022].
- [6] ITI, «DIGITAL TWINS: Gemelos Digitales en la transición a la Industria 4.0,» ITI, 30 Junio 2020. [En línea]. Available: <https://www.iti.es/proyectosidi/proyecto-gemelos-digitales-industria-4-0/>. [Último acceso: 08 Noviembre 2022].
- [7] M. Honrubia, «Gemelo Digital y su uso como herramienta predictiva del medio industrial,» enmotive, 2020. [En línea]. Available: <https://www.enmotive.com/es/gemelo-digital-industrial/>. [Último acceso: 8 Noviembre 2022].
- [8] F. d. d. Barcelona, «Realidad virtual,» FIB, 2020. [En línea]. Available: <https://www.fib.upc.edu/retro-informatica/avui/realitatvirtual.html>. [Último acceso: 10 Noviembre 2022].

[9] Oracle, «¿Qué es la inteligencia artificial? Obtenga más información sobre la inteligencia artificial,» oracle, 2020. [En línea]. Available: <https://www.oracle.com/mx/artificial-intelligence/what-is-ai/>. [Último acceso: 10 Noviembre 2022].

# APÉNDICE

## APÉNDICE A

**Tabla 2: Tabla de variables de entrada (Marcas)**

Tabla de variables de entrada			
SP1_A	Bool	%M0.0	Sensor de proximidad Iz. 1
SP2_A	Bool	%M0.1	Sensor de proximidad Iz. 2
SP3_A	Bool	%M0.2	Sensor de proximidad Iz. 3
SP4_A	Bool	%M0.3	Sensor de proximidad Iz. 4
SP5_A	Bool	%M0.4	Sensor de proximidad Iz. 5
SP6_A	Bool	%M0.5	Sensor de proximidad Iz. 6
SP7_A	Bool	%M0.6	Sensor de proximidad Iz. 7
SP8_A	Bool	%M0.7	Sensor de proximidad Iz. 8
SP1_B	Bool	%M1.0	Sensor de proximidad der. 1
SP2_B	Bool	%M1.1	Sensor de proximidad der. 2
SP3_B	Bool	%M1.2	Sensor de proximidad der. 3
SP4_B	Bool	%M1.3	Sensor de proximidad der. 4
SP5_B	Bool	%M1.4	Sensor de proximidad der. 5
SP6_B	Bool	%M1.5	Sensor de proximidad der. 6
SP7_B	Bool	%M1.6	Sensor de proximidad der. 7
SP8_B	Bool	%M1.7	Sensor de proximidad der. 8
RT_1	Bool	%M2.0	Rele Termico Puerta 1
RT_2	Bool	%M2.1	Rele Termico Puerta 2
RT_3	Bool	%M2.2	Rele Termico Puerta 3
RT_4	Bool	%M2.3	Rele Termico Puerta 4
RT_5	Bool	%M2.4	Rele Termico Puerta5
RT_6	Bool	%M2.5	Rele Termico Puerta 6
RT_7	Bool	%M2.6	Rele Termico Puerta 7
RT_8	Bool	%M2.7	Rele Termico Puerta 8
FC1_A	Bool	%M3.0	Final de Carrera Sup 1
FC2_A	Bool	%M3.1	Final de Carrera Sup 2
FC3_A	Bool	%M3.2	Final de Carrera Sup 3
FC4_A	Bool	%M3.3	Final de Carrera Sup 4
FC5_A	Bool	%M3.4	Final de Carrera Sup 5
FC6_A	Bool	%M3.5	Final de Carrera Sup 6
FC7_A	Bool	%M3.6	Final de Carrera Sup 7
FC8_A	Bool	%M3.7	Final de Carrera Sup 8
FC1_B	Bool	%M4.0	Final de Carrera Inf 1
FC2_B	Bool	%M4.1	Final de Carrera Inf 2
FC3_B	Bool	%M4.2	Final de Carrera Inf 3
FC4_B	Bool	%M4.3	Final de Carrera Inf 4
FC5_B	Bool	%M4.4	Final de Carrera Inf 5
FC6_B	Bool	%M4.5	Final de Carrera Inf 6

FC7_B	Bool	%M4.6	Final de Carrera Inf 7
FC8_B	Bool	%M4.7	Final de Carrera Inf 8
BotS_1	Bool	%M5.0	Botonera subir 1
BotS_2	Bool	%M5.1	Botonera subir 2
BotS_3	Bool	%M5.2	Botonera subir 3
BotS_4	Bool	%M5.3	Botonera subir 4
BotS_5	Bool	%M5.4	Botonera subir 5
BotS_6	Bool	%M5.5	Botonera subir 6
BotS_7	Bool	%M5.6	Botonera subir 7
BotS_8	Bool	%M5.7	Botonera subir 8
BotB_1	Bool	%M6.0	Botonera bajar 1
BotB_2	Bool	%M6.1	Botonera bajar 2
BotB_3	Bool	%M6.2	Botonera bajar 3
BotB_4	Bool	%M6.3	Botonera bajar 4
BotB_5	Bool	%M6.4	Botonera bajar 5
BotB_6	Bool	%M6.5	Botonera bajar 6
BotB_7	Bool	%M6.6	Botonera bajar 7
BotB_8	Bool	%M6.7	Botonera bajar 8
RESET	Bool	%M11.0	Boton para reset/restaurar

**Tabla 3: Tabla de variables de salida (Marcas)**

Tabla de variables de salida (marcas)			
MOT_1	Bool	%M7.0	Permisivo Habilitar Movimiento Puerta
MOT_2	Bool	%M7.1	Permisivo Habilitar Movimiento Puerta
MOT_3	Bool	%M7.2	Permisivo Habilitar Movimiento Puerta
MOT_4	Bool	%M7.3	Permisivo Habilitar Movimiento Puerta
MOT_5	Bool	%M7.4	Permisivo Habilitar Movimiento Puerta
MOT_6	Bool	%M7.5	Permisivo Habilitar Movimiento Puerta
MOT_7	Bool	%M7.6	Permisivo Habilitar Movimiento Puerta
MOT_8	Bool	%M7.7	Permisivo Habilitar Movimiento Puerta
INL_R_1	Bool	%M8.0	Indicador Rojo Torre
INL_R_2	Bool	%M8.1	Indicador Rojo Torre
INL_R_3	Bool	%M8.2	Indicador Rojo Torre
INL_R_4	Bool	%M8.3	Indicador Rojo Torre
INL_R_5	Bool	%M8.4	Indicador Rojo Torre
INL_R_6	Bool	%M8.5	Indicador Rojo Torre
INL_R_7	Bool	%M8.6	Indicador Rojo Torre
INL_R_8	Bool	%M8.7	Indicador Rojo Torre
INL_A_1	Bool	%M9.0	Indicador amarillo Torre
INL_A_2	Bool	%M9.1	Indicador amarillo Torre
INL_A_3	Bool	%M9.2	Indicador amarillo Torre
INL_A_4	Bool	%M9.3	Indicador amarillo Torre

INL_A_5	Bool	%M9.4	Indicador amarillo Torre
INL_A_6	Bool	%M9.5	Indicador amarillo Torre
INL_A_7	Bool	%M9.6	Indicador amarillo Torre
INL_A_8	Bool	%M9.7	Indicador amarillo Torre
INL_V_1	Bool	%M10.0	Indicador verde Torre
INL_V_2	Bool	%M10.1	Indicador verde Torre
INL_V_3	Bool	%M10.2	Indicador verde Torre
INL_V_4	Bool	%M10.3	Indicador verde Torre
INL_V_5	Bool	%M10.4	Indicador verde Torre
INL_V_6	Bool	%M10.5	Indicador verde Torre
INL_V_7	Bool	%M10.6	Indicador verde Torre
INL_V_8	Bool	%M10.7	Indicador verde Torre

**Tabla 4: Tabla de variables de salida (físicas)**

Tabla de variables de salidas (físicas)			
MOT1_out	Bool	%Q0.0	Habilitador permisivo Motor Puerta 1
MOT2_out	Bool	%Q0.1	Habilitador permisivo Motor Puerta 2
MOT3_out	Bool	%Q0.2	Habilitador permisivo Motor Puerta 3
MOT4_out	Bool	%Q0.3	Habilitador permisivo Motor Puerta 4
MOT5_out	Bool	%Q0.4	Habilitador permisivo Motor Puerta 5
MOT6_out	Bool	%Q0.5	Habilitador permisivo Motor Puerta 6
MOT7_out	Bool	%Q0.6	Habilitador permisivo Motor Puerta 7
MOT8_out	Bool	%Q0.7	Habilitador permisivo Motor Puerta 8
LuzR_1	Bool	%Q1.0	Indicador Rojo Torre (Salida Física) 1
LuzR_2	Bool	%Q1.1	Indicador Rojo Torre (Salida Física) 2
LuzR_3	Bool	%Q1.2	Indicador Rojo Torre (Salida Física) 3
LuzR_4	Bool	%Q1.3	Indicador Rojo Torre (Salida Física) 4
LuzR_5	Bool	%Q1.4	Indicador Rojo Torre (Salida Física) 5
LuzR_6	Bool	%Q1.5	Indicador Rojo Torre (Salida Física) 6
LuzR_7	Bool	%Q1.6	Indicador Rojo Torre (Salida Física) 7
LuzR_8	Bool	%Q1.7	Indicador Amarillo Torre (Salida Física) 1
LuzA_1	Bool	%Q2.0	Indicador Amarillo Torre (Salida Física) 2
LuzA_2	Bool	%Q2.1	Indicador Amarillo Torre (Salida Física) 3
LuzA_3	Bool	%Q2.2	Indicador Amarillo Torre (Salida Física) 4
LuzA_4	Bool	%Q2.3	Indicador Amarillo Torre (Salida Física) 5
LuzA_5	Bool	%Q2.4	Indicador Amarillo Torre (Salida Física) 6
LuzA_6	Bool	%Q2.5	Indicador Amarillo Torre (Salida Física) 7
LuzA_7	Bool	%Q2.6	Indicador Amarillo Torre (Salida Física) 8
LuzA_8	Bool	%Q2.7	Indicador Amarillo Torre (Salida Física) 9
LuzV_1	Bool	%Q3.0	Indicador Verde Torre (Salida Física) 1
LuzV_2	Bool	%Q3.1	Indicador Verde Torre (Salida Física) 2
LuzV_3	Bool	%Q3.2	Indicador Verde Torre (Salida Física) 3

LuzV_4	Bool	%Q3.3	Indicador Verde Torre (Salida Física) 4
LuzV_5	Bool	%Q3.4	Indicador Verde Torre (Salida Física) 5
LuzV_6	Bool	%Q3.5	Indicador Verde Torre (Salida Física) 6
LuzV_7	Bool	%Q3.6	Indicador Verde Torre (Salida Física) 7
LuzV_8	Bool	%Q3.7	Indicador Verde Torre (Salida Física) 8

**Tabla 5: Tabla de variables de entradas (físicas)**

Tabla de variables de Entradas (físicas)			
SP1_Ai1	Bool	%I0.0	Sensor de proximidad lz. 1
SP1_Ai2	Bool	%I0.1	Sensor de proximidad lz. 2
SP1_Ai3	Bool	%I0.2	Sensor de proximidad lz. 3
SP1_Ai4	Bool	%I0.3	Sensor de proximidad lz. 4
SP1_Ai5	Bool	%I0.4	Sensor de proximidad lz. 5
SP1_Ai6	Bool	%I0.5	Sensor de proximidad lz. 6
SP1_Ai7	Bool	%I0.6	Sensor de proximidad lz. 7
SP1_Ai8	Bool	%I0.7	Sensor de proximidad lz. 8
SP1_Bi1	Bool	%I1.0	Sensor de proximidad der. 1
SP1_Bi2	Bool	%I1.1	Sensor de proximidad der. 2
SP1_Bi3	Bool	%I1.2	Sensor de proximidad der. 3
SP1_Bi4	Bool	%I1.3	Sensor de proximidad der. 4
SP1_Bi5	Bool	%I1.4	Sensor de proximidad der. 5
SP1_Bi6	Bool	%I1.5	Sensor de proximidad der. 6
SP1_Bi7	Bool	%I1.6	Sensor de proximidad der. 7
SP1_Bi8	Bool	%I1.7	Sensor de proximidad der. 8
RT_i1	Bool	%I2.0	Rele térmico 1
RT_i2	Bool	%I2.1	Rele térmico 2
RT_i3	Bool	%I2.2	Rele térmico 3
RT_i4	Bool	%I2.3	Rele térmico 4
RT_i5	Bool	%I2.4	Rele térmico 5
RT_i6	Bool	%I2.5	Rele térmico 6
RT_i7	Bool	%I2.6	Rele térmico 7
RT_i8	Bool	%I2.7	Rele térmico 8
FC1_Ai	Bool	%I3.0	Final de carrera A (Entrada física)
FC2_Ai	Bool	%I3.1	Final de carrera A (Entrada física)
FC3_Ai	Bool	%I3.2	Final de carrera A (Entrada física)
FC4_Ai	Bool	%I3.3	Final de carrera A (Entrada física)
FC5_Ai	Bool	%I3.4	Final de carrera A (Entrada física)
FC6_Ai	Bool	%I3.5	Final de carrera A (Entrada física)
FC7_Ai	Bool	%I3.6	Final de carrera A (Entrada física)
FC8_Ai	Bool	%I3.7	Final de carrera A (Entrada física)
FC1_Bi	Bool	%I4.0	Final de carrera B (Entrada Física)
FC2_Bi	Bool	%I4.1	Final de carrera B (Entrada Física)

FC3_Bi	Bool	%I4.2	Final de carrera B (Entrada Física)
FC4_Bi	Bool	%I4.3	Final de carrera B (Entrada Física)
FC5_Bi	Bool	%I4.4	Final de carrera B (Entrada Física)
FC6_Bi	Bool	%I4.5	Final de carrera B (Entrada Física)
FC7_Bi	Bool	%I4.6	Final de carrera B (Entrada Física)
FC8_Bi	Bool	%I4.7	Final de carrera B (Entrada Física)