

**UNIVERSIDAD MIGUEL HERNÁNDEZ DE
ELCHE**

ESCUELA POLITÉCNICA SUPERIOR DE ELCHE

**GRADO EN INGENIERÍA ELECTRÓNICA Y
AUTOMÁTICA INDUSTRIAL**



UNIVERSITAS
Miguel Hernández

**DISEÑO Y OPTIMIZACIÓN EN TIEMPO REAL DE
UN SISTEMA DE CONTROL DE PRODUCCIÓN Y
MANTENIMIENTO MEDIANTE LA INTEGRACIÓN
DE UNA INTERFAZ HMI PARA UN ROBOT SCARA
EN LA INDUSTRIA DEL CALZADO**

TRABAJO FIN DE GRADO

Marzo- 2026

**AUTOR: Érika Riquelme
Espinosa**

**DIRECTORES: Jonatan Lloret Reina
Carlos Pérez Vidal**

INDICE

1. INTRODUCCIÓN Y CONTEXTO	3
1.1 MOTIVACIÓN Y OBJETO DEL PROYECTO.....	11
1.2 METODOLOGÍA EMPLEADA.....	12
1.3 ALCANCE DEL TRABAJO.....	13
2. DESCRIPCIÓN GENERAL DEL SISTEMA	14
3. COMPONENTES DEL SISTEMA DE PRODUCCIÓN	15
3.1 AUTÓMATA.....	15
3.1.1 CARACTERÍSTICAS PRINCIPALES.....	16
3.1.2 COMUNICACIONES CON OTROS DISPOSITIVOS.....	16
3.1.3 HERRAMIENTAS UTILIZADAS PARA SU PROGRAMACIÓN	17
3.1.4 PRUEBAS REALIZADAS	18
3.1.5 CONTROL DE SEGURIDAD Y VALIDACIÓN	19
3.2 HMI	20
3.2.1 CARACTERÍSTICAS Y HERRAMIENTAS UTILIZADAS EN SU DESARROLLO.....	20
3.2.2 MODOS DE OPERACIÓN Y PANTALLAS DISEÑADAS.....	21
3.2.3 NIVELES DE USUARIO Y ACCESO	23
4.1 EXPLICACIÓN BREVE DE LA ESTRUCTURA DEL PROGRAMA	25
4.2 DIAGRAMAS DE FLUJO	26
5. NUEVAS IMPLEMENTACIONES DEL SISTEMA	31
5.1 PANTALLA DE CONTROL DE PRODUCCIÓN.....	32
5.1.1 ARQUITECTURA DE PROGRAMACIÓN.....	32
5.1.2 DESARROLLO DE LA PROGRAMACIÓN	33
5.1.3 DISEÑO DE LA INTERFAZ	40
5.2 PANTALLA DE MANTENIMIENTO.....	46
5.2.1 ARQUITECTURA DE PROGRAMACIÓN.....	47
5.2.2 DESARROLLO DE LA PROGRAMACIÓN	47
5.2.3 DISEÑO DE LA INTERFAZ	49
5.3 INTEGRACIÓN FUNCIONAL DE LA PROGRAMACIÓN	52
5.4 INCORPORACIÓN DEL HMI.....	54
5.3.1 CARACTERÍSTICAS TÉCNICAS	57
5.3.2 MONTAJE E INSTALACIÓN DEL TERMINAL	58
6. BIBLIOGRAFÍA	59

1. INTRODUCCIÓN Y CONTEXTO

La robótica industrial se ha convertido en algo necesario en las empresas modernas, ya que refuerza la mejora de la productividad, precisión y seguridad en los procesos de fabricación. Para entender cómo hemos llegado a este nivel de desarrollo, es necesario entender cómo la revolución industrial marcó el inicio de una nueva forma de producir.

Durante el siglo XVIII llegó la Revolución industrial con la máquina de vapor y comenzó la sustitución de la fabricación manual por la mecanizada. Tal como se analiza en el recorrido evolutivo propuesto por Barrientos en [1] después de la llegada de la electricidad y en el siglo XX la informática y la electrónica continuaron con la segunda revolución industrial permitiendo la automatización de procesos más complejos.

La justificación principal de la automatización es la eterna búsqueda del aumento de la productividad, dado que además de acelerar la producción en muchos casos permite un procesamiento de datos a tiempo real, facilitando la detección adelantada de fallos. Asimismo la automatización garantiza muchas otras mejoras, como el aumento de la calidad. Como se desprende de los principios de manufactura en [2], la reducción de errores humanos, el aumento de la precisión y sobre todo la reducción del riesgo en tareas peligrosas y físicamente exigentes donde evitar la intervención directa humana puede aumentar su seguridad laboral.

El siguiente paso a la automatización es la robótica industrial, puesto que los robots no están destinados únicamente a realizar tareas repetitivas, sino que han pasado a formar parte de un sistema inteligente que es capaz de adaptarse a una producción variable, definiéndose formalmente como un

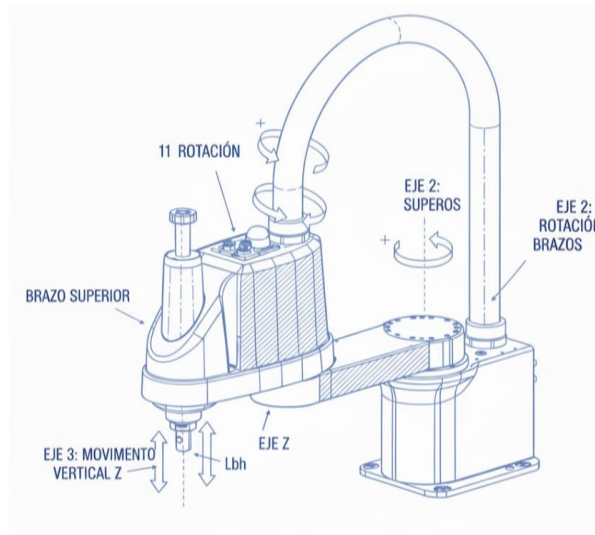
Manipulador multifuncional controlado automáticamente y reprogramable en la ISO [3]. Los cuales están dotados con múltiples sensores y varios equipos conectados y programados para analizar y responder a tiempo real ante cambios de la producción o del entorno, permitiendo a la industria moderna alcanzar aún mayor velocidad, precisión y seguridad de las que nunca antes había logrado.

Como consecuencia de la complejidad de estos sistemas, cada proyecto industrial está condicionado por diversas necesidades, determinando las piezas a seleccionar, siendo la elección del robot el primer paso fundamental en este proceso.

Para ello, es preciso considerar que los robots se clasifican en diferentes tipos según su morfología y configuración articular, dotando a cada clase de una variada combinación entre su espacio de trabajo, velocidad, precisión y capacidad de carga, analizando las

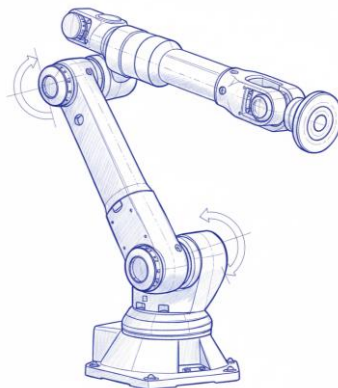
El robot SCARA (Selective Compliance Assembly Robot Arm) tiene una configuración ideal para ensamblaje rápido. La estructura, ilustrada en la Figura 2, integra tres articulaciones de revolución y una articulación prismática para el eje vertical explicadas en la clasificación morfológica detallada en [2]. Esta combinación resulta en una excepcional resistencia a la carga vertical, una precisión en el eje Z y manteniendo una gran velocidad.

Figura 2. Esquema estructural y configuración de articulaciones de un robot tipo SCARA. Imagen generada mediante IA.



El Robot Articulado es uno de los más comunes y flexibles de la industria, ya que tiene una forma que simula la de un brazo humano. Se compone por varias articulaciones de revolución, normalmente entre cuatro y seis, concediendo destreza y gran capacidad de adaptación. Una de las composiciones de articulaciones más versátiles y comunes en la actualidad se representa en la Figura 3.

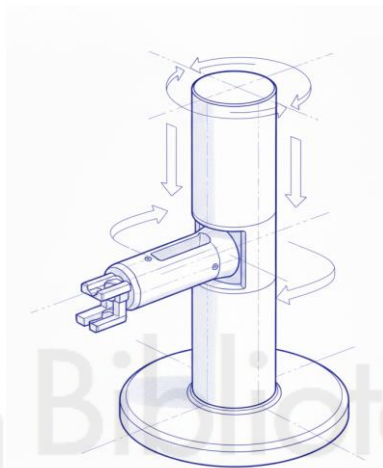
Figura 3. Esquema estructural de un brazo robótico antropomórfico basado en articulaciones de revolución. Imagen generada mediante IA.



El Robot Cilíndrico recibe su nombre debido a su área de trabajo, la cual tiene forma de cilindro. Es el resultado de la combinación de una articulación de

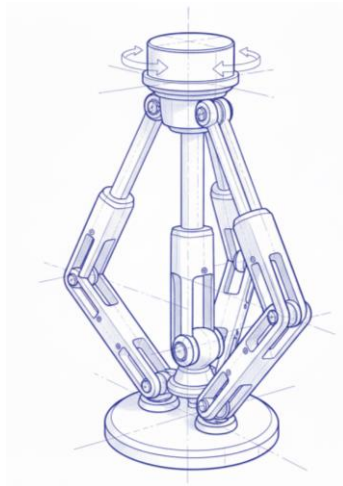
revolución con dos articulaciones prismáticas, como ejemplo en la Figura 4 se muestra una estructura básica que integra las articulaciones necesarias. Destaca por su gran robustez y notable capacidad para manipular máquinas y herramientas.

Figura 4. Esquema de un robot cilíndrico basado en la combinación de articulaciones RPP (Rotacional-Prismática-Prismática). Imagen generada mediante IA.



Por último, se presenta en la Figura 5 el Robot Paralelo también conocido como Delta, se caracteriza por su estructura compuesta por múltiples brazos articulados en paralelo que conectan la base con el efector final. Como se puede apreciar en dicho esquema gracias a la reducción de inercias que se consigue con este diseño este tipo de robot puede conseguir mucha velocidad y dinamismo para movimientos rápidos y efectivos.

Figura 5. Esquema estructural y configuración de articulaciones de un robot Paralelo (Delta). Imagen generada mediante IA.



Una vez destacadas las ventajas de cada modelo de robot, se analizan las necesidades del proyecto teniendo en cuenta que se va a automatizar la colocación de unas piezas recortadas en una plancha de material. A esta clase de operación se le conoce como proceso de pick and place (recoger y colocar) bajo la definición técnica de manipulación industrial recogida en [4], tradicionalmente este trabajo habría sido realizado por operarios, pero al ser un trabajo iterativo puede automatizarse para reducir el error y mejorar el ritmo de trabajo aumentando su productividad y eficiencia.

Las necesidades técnicas y específicas de esta tarea de manipulación y apilamiento de piezas recortadas son:

- Alta velocidad, esencial para maximizar la tasa de producción y eficiencia del proceso.
- Capacidad de manejo de piezas diversas y variables, el principal desafío técnico de este proyecto es la variación de las piezas y materiales lo cual requiere de una gran capacidad de adaptación.

- Alta precisión del eje vertical es fundamental la capacidad de ejercer una fuerza muy controlada para poder apilar las piezas con la exactitud requerida, asegurando que cada una quede perfectamente alineada y estable en la pila sin dañar el material.

Considerando los tipos de robot más comunes, se ha realizado una evaluación para determinar cuál era el más adecuado.

El robot cartesiano ha sido descartado, dado que, aun contando con la precisión necesaria, carece de la velocidad suficiente para seguir el ritmo de producción exigido. El robot articulado también ha sido excluido porque su gran flexibilidad y destreza son redundantes para esta tarea, además tampoco alcanza la velocidad requerida para el proyecto. Por otro lado, el robot paralelo (Delta) cumple los requisitos de velocidad, pero tiene un espacio reducido de trabajo, limitando el tamaño de las planchas que se pueden utilizar.

El robot SCARA tiene una configuración perfecta para esta tarea dado que cumple con la velocidad de operación exigida y ofrece tanto la resistencia como la precisión vertical necesarias garantizando la exactitud al mover las piezas sin riesgo de deformación. Por ello, el robot SCARA se sitúa como la solución óptima para el sistema de pick and place en la industria del calzado que se va a optimizar; equilibra la velocidad, la precisión y el coste de la manera más eficiente para el proyecto tal y como se demuestra en el análisis comparativo de rendimiento de Nof en [5].

Con la finalidad de cumplir con los estándares de productividad exigidos, se ha optado por una configuración de cinemática paralela. En la Figura 6, se puede apreciar el modelo seleccionado: el robot SCARA DELTA DRSA0LMOSANN002.

Figura 6: Robot SCARA Delta modelo DRSA0LMOSANN002 y esquema de su arquitectura mecánica.



Caracterizado por mantener los motores anclados a la base fija, reduciendo significativamente la masa móvil y, por consiguiente, la inercia del brazo. Esta ventaja mecánica lo convierte en la solución ideal para operaciones en líneas de alta velocidad que requieren ciclos de tiempo mínimos y una precisión constante.

Una vez se ha elegido el modelo de robot que cumplirá la función principal del proyecto, el siguiente paso es la elección de un PLC (Controlador Lógico Programable) y una HMI (Interfaz Hombre-Máquina) adecuados para centralizar el control y la supervisión del sistema.

Al igual que en la selección del robot, se debe analizar la marca y el modelo del controlador, considerando una serie de características técnicas indispensables. Para comenzar, dada su integración en un sistema de automatización complejo, el dispositivo debe ser completamente adaptable y, sobre todo,

disponer de capacidad de expansión necesaria para futuras adiciones de entradas y salidas.

Cabe destacar que la parte más innovadora en la realización del proyecto ha sido prescindir de la controladora tradicional del robot, aumentando la necesidad de una velocidad de procesamiento de PLC superior a la estándar, asegurando que estará capacitado para realizar todas las tareas sin margen de error como se define en los estándares de programación lógica de la norma IEC 61131-3 [6]. Por lo tanto, la capacidad se define por la cantidad y tipo de entradas y salidas solicitadas, la potencia de la CPU para ciclos a tiempo real, los protocolos de comunicación de campo y por el software asociado.

Al tratarse de un entorno industrial, es esencial que el PLC esté diseñado para soportar las condiciones ambientales de operación, previniendo fallos o roturas provocadas por factores como la temperatura y la humedad alcanzando un grado de protección adecuado conforme a las especificaciones del IEC 60529 [7]. Finalmente, de entre todas las opciones que satisfagan estos requisitos técnicos y de adaptación, se evaluará el costo total y la calidad del soporte técnico ofrecido por el fabricante.

El PLC seleccionado para satisfacer todo el control del sistema es PLC Delta AX-8xxEP0 Windows Series (PC-Based Motion Controller) debido a que está diseñado para ser capaz de realizar la tarea del controlador y todas las demás que se analizarán en posterioridad. Asegura la conectividad con el resto de los componentes y al ser del mismo fabricante que el robot añade facilidades en la programación y conexión. Para utilizar la función destacada se requieren librerías de cinemática específicas instaladas en el entorno de desarrollo DIADesigner-AX proporcionadas por el fabricante [8], permitiendo mantener el control tanto de la lógica como del movimiento en una única CPU, eliminando las latencias de comunicación, pero exigiendo un nivel más avanzado de programación que resuelva la importación de las librerías de la marca y el uso de las funcionalidades de movimiento coordinado (SoftMotion) siguiendo los estándares de bloques de función internacionales definido por PLCopen Technical Committee en [9].

Tras seleccionar los componentes principales que van a formar parte del sistema y definir la centralización de la arquitectura, se diseña el programa para el autómatas, la pantalla y se sincronizan mediante el protocolo elegido.

La programación de control se convierte en el núcleo funcional del sistema, traduciendo la lógica secuencial, los algoritmos de la cinemática y la gestión de fallos del proceso a un código robusto y modular que ejecuta el PLC. Este programa se ocupa de que la celda robotizada funcione de forma segura y coordinada y de establecer la comunicación necesaria para el intercambio de datos con el resto de los componentes. Durante su desarrollo, se prioriza la estandarización, lo cual es clave para que el sistema pueda seguir actualizándose, sea fácil de mantener y su documentación resulte clara.

El diseño del HMI de este proyecto supera la funcionalidad básica de 'arranque y paro' que comúnmente se observa en algunos proyectos similares, contemplando una monitorización constante de variables y sensores, permitiendo realizar ajustes en tiempo real sobre la tarea (velocidad o altura), implementando al mismo tiempo un detallado diagnóstico de errores. Todo ello fundamentado por los principios de interfaces de alto rendimiento establecidos por ISA-101.01 [10]. La versatilidad resultante es esencial, para permitir la evolución de la industria actual hacia la industria 4.0, la cual requiere de máquinas que, además de la ejecución de tareas repetitivas, dispongan de comunicación inteligente y flexible entre ellas.

1.1 MOTIVACIÓN Y OBJETO DEL PROYECTO

El objetivo principal de este proyecto es mejorar la productividad mediante la automatización de alto rendimiento en una tarea de "pick and place". Para ello se ha estudiado la mejora de cada componente del sistema, principalmente centrándose en conseguir una programación que permita aumentar la eficiencia de los tiempos de trabajo llegando a una velocidad de operación óptima.

Además, para conseguir disminuir los tiempos de reacción, se ha procurado diseñar una aplicación lo más accesible y funcional para el control del sistema por parte del operario. Esta interfaz no solo facilitará la rápida intervención y

resolución de cualquier problema, sino que gestionará la monitorización a tiempo real de las variables críticas recibidas desde el PLC, agilizando el diagnóstico de fallos y acortando los tiempos de parada al máximo.

Con la finalidad de reducir los fallos de funcionamiento en el sistema de producción se ha incorporado al programa un apartado de control destinado a la gestión de alarmas de mantenimiento, orientado a transformar el mantenimiento reactivo en preventivo. Esto permite minimizar los tiempos de inactividad no planificados y asegurar la continuidad de la producción incrementando directamente los índices de producción.

1.2 METODOLOGÍA EMPLEADA

El proyecto se ha desarrollado tomando como ejemplo la metodología en V (V-Model), la cual constituye un estándar en ingeniería de control y en la automatización industrial. Tomando como referencia el flujo de diseño mecatrónico de la VDI 2206I [11].

Esta metodología exige que, tras dividir el proyecto en partes, cada una de ellas complete el siguiente ciclo: Definición del Objetivo, Diseño e Implementación.

De esta manera cada una de las etapas de desarrollo queda vinculada con una etapa de pruebas correspondiente, culminando en una Integración y Validación completa final. Esta metodología respalda la calidad del proyecto de principio a fin asegurándose que todos los componentes individuales encajen y funcionen de la manera más óptima.

Las etapas principales que se han seguido para el desarrollo del proyecto son las siguientes:

- Análisis de requisitos del HMI y las tareas del PLC, definiendo la arquitectura de control y los elementos de hardware y software necesarios.
- Diseño de la arquitectura de comunicación entre componentes, configurando las comunicaciones pertinentes entre el PLC con el HMI.

- Programación del PLC en el entorno DIADesigner-AX (basado en CoDeSys), diseñando e implementando la lógica de funcionamiento con los componentes necesarios de seguridad.
- Diseño e implementación de la aplicación en DiaScreen, desarrollando la interfaz gráfica para el control y supervisión.
- Fase de Verificación y Validación HMI, comprobando el correcto funcionamiento, la comunicación entre dispositivos y la facilidad de uso por parte del operario.

1.3 ALCANCE DEL TRABAJO

El alcance del Trabajo de Fin de Grado se centra en la implementación y automatización de un proceso de “pick and place” en una línea de producción de calzado, buscando la óptima productividad y eficiencia.

Componentes y Software

El desarrollo del trabajo describe y analiza la implementación del Robot y PLC ambos de Delta Electronics, detalla las peculiaridades de la programación realizadas para el control y la seguridad del robot mediante la herramienta de Codesys.

Desarrolla un apartado de análisis de la producción orientado al balance y la optimización de esta, compuesto tanto por la programación como por el diseño de la interfaz. Además de una pantalla orientada exclusivamente a la gestión de alarmas de mantenimiento completamente parametrizables.

Las limitaciones que tiene el TFG para la investigación e implementación son las siguientes, en este trabajo no se incluye un análisis económico del sistema y los resultados obtenidos, son específicos para los modelos de PLC y Robot elegidos en el propio proyecto, dado que la programación y tiempos mínimos sólo se garantizan para estos modelos.

Hardware y Configuración

El proyecto no incluye el dimensionamiento eléctrico, ni ningún tipo de plano. No se documenta ni explica la calibración o montaje físico de ninguna parte del sistema ya incluidas en el proyecto inicial, tampoco se documentará la elección y programación de sensores y cinta.

Se contempla la sustitución del PC por un panel HMI homologado, respondiendo a unas necesidades operativas del sistema y describiendo brevemente su instalación física.

Funcionalidad

El sistema de control de producción y mantenimiento se basa en un análisis tanto de valores registrados por sensores externos como las propias variables del PLC, mostrando la información correspondiente en la interfaz pero las acciones de mejora de producción requieren de un operario que realice la evaluación con las respuestas obtenidas. De la misma manera los avisos configurados en el control de mantenimiento se realizan en función de los valores determinados por el usuario destinado a ello y se notificarán mediante la interfaz.

2. DESCRIPCIÓN GENERAL DEL SISTEMA

El sistema automatizado en el que se basa este proyecto tiene como objetivo gestionar la manipulación de piezas previamente cortadas. Las tareas principales son recoger las planchas de material, separar las piezas útiles y desechar el material restante.

La instalación está formada por el sistema de transporte, el robot industrial, la unidad de control y la interfaz de supervisión. El sistema de transporte está compuesto por dos cintas independientes, la primera se ocupa de la alimentación del sistema, introduciendo las planchas cortadas sobre las cuales se va a trabajar, mientras que la segunda se destina a evacuar las piezas una vez separadas y procesadas. La labor de pick and place del proceso recae

sobre el robot SCARA, el cual mueve las piezas entre ambas cintas, separándolas de los restos del material.

La lógica que gestiona todos los actuadores del sistema teniendo en cuenta sensores y señales es la Unidad de control, formada por un autómata programable. Este es el maestro del sistema, se ocupa del movimiento de las cintas, del robot y de gestionar paradas de emergencia y evitar errores.

Por último, la interfaz de supervisión es la que relaciona un sistema automatizado con la interacción humana. Actualmente la interfaz se lleva a cabo mediante un ordenador conectado al PLC que ejerce las funciones de HMI. Permite al personal una supervisión del funcionamiento a tiempo real, el control en modo manual o automático y ofrece una visión de los errores detectados en cada momento lo que facilita el diagnóstico de problemas y la rápida puesta en marcha.

3. COMPONENTES DEL SISTEMA DE PRODUCCIÓN

El sistema de automatización se articula mediante la integración de una serie de elementos, todos seleccionados rigurosamente en función de sus características técnicas. Cada uno de ellos programados y conectados para trabajar coordinadamente. La arquitectura del sistema se compone de: autómata, HMI, robot, cintas transportadoras, sensores y cámara de visión.

3.1 AUTÓMATA

Como unidad de control se ha seleccionado el PLC Delta AX-8xxEP0 Windows Series. Según las especificaciones técnicas del fabricante, este dispositivo consta de una arquitectura PC-Based que permite el control de movimiento a alta velocidad utilizando el protocolo EtherCAT. Gracias a ello, cumple con las necesidades de capacidad de expansión, velocidad de procesamiento y robustez para entornos industriales. Se adjunta en el Anexo I el datasheet completo y el Anexo II con el manual de uso, donde se detallan los consumos eléctricos y los rangos de temperatura de operación que garantizan su fiabilidad.

3.1.1 CARACTERÍSTICAS PRINCIPALES

- El controlador AX-8xxEP0 Windows series es un PLC de alto rendimiento de la marca Delta Electronics, puesto que tiene la capacidad de ocuparse de las secuencias lógicas que debe seguir el sistema, de notificar los errores detectados correctamente y gestionar las reacciones del sistema ante ellos, de los cálculos de trayectoria del robot en base a las coordenadas recibidas y del control de producción y mantenimiento a implementar.
- Entradas/Salidas (E/S): aunque por sí solo no consta de muchas, principalmente se utiliza con expansiones por bus de campo dado que son necesarias para llevar el control de cada componente del sistema.
- Procesador: Intel Multi-Core.
- Memoria: 4 GB de RAM /128 GB SSD
- Función habitual: Control de movimiento avanzado de robótica.

3.1.2 COMUNICACIONES CON OTROS DISPOSITIVOS.

Para optimizar el flujo de información según su prioridad técnica, se ha implementado una segregación de protocolos en el proyecto: los sensores de posición de las cintas y la recepción de coordenadas se integra mediante protocolo Modbus TCP en configuración Slave, por el contrario los actuadores tanto del robot como de las cintas se centralizan en un bus EtherCAT. Esta arquitectura garantiza una adquisición de datos estable y la conexión de ejes en tiempo real que se requiere en el control de un robot de alta precisión como este.

En lo que respecta a la interfaz simulada en el PC para el control y monitorización del sistema, se ha establecido una conexión Ethernet utilizando el protocolo nativo de CODESYS. Gracias a la configuración de símbolos, las variables del programa se gestionan mediante un direccionamiento simbólico llamado etiquetas (tags), esto agiliza los cambios en el software y garantiza la integridad de los datos durante el mapeo en el diseño de la interfaz.

3.1.3 HERRAMIENTAS UTILIZADAS PARA SU PROGRAMACIÓN

- El Entorno de Desarrollo (IDE): DIADesigner-AX

El programa utilizado es DIADesigner-AX que tiene como núcleo CODESYS, aunque a diferencia del tradicional consta de algunos drivers necesarios preinstalados necesarios para la complejidad del control del robot.

Este Software se ha utilizado en este proyecto para la totalidad de su programación; tanto las secuencias básicas de funcionamiento del sistema como en el control de movimiento de cada componente, se ocupa de todos los cálculos aritméticos y actúa de base de datos para el historial de producción. En otros casos se suele utilizar para la confección y configuración de la interfaz pero dada la complejidad requerida de este proyecto, la interfaz que este programa puede ofrecer es insuficiente.

El entorno de CODESYS cumple la normativa IEC 61131-3 y permite mezclar diferentes lenguajes, consta de : ST (Texto estructurado) se usa para lógica matemática y bucles y CFC (Continuous Function Chart) que es un entorno más visual.

- Herramientas de Configuración de Movimiento (SoftMotion)

Dentro del software DIADesigner-AX se incluyen configuradores gráficos que suplen a la programación de ecuaciones matriciales de forma manual: El configurador de grupo de ejes y el ECAM: herramientas que definidas por PLCopen Technical Committee en los estándares de abstracción de control de movimiento de [9], permiten delegar el cálculo de las ecuaciones matriciales al sistema de control.

El Configurador de Grupo de Ejes se ocupa de resolver la cinemática directa e inversa del robot a tiempo real. Transforma las órdenes de movimiento del usuario en posiciones exactas para cada servomotor, siempre teniendo en cuenta las restricciones y dimensiones físicas del robot.

El editor de levas (ECAM) permite el diseño y la gestión de levas electrónicas, sirve para definir gráficamente la curva de posición y calcula la interpolación entre puntos medios mediante polinomios de quinto orden. Esto es necesario para mantener una continuidad en la velocidad y aceleración, eliminando las sacudidas.

- Herramientas de diagnóstico y “tuning”

El Trace y el Watch List son herramientas esenciales para verificar el funcionamiento del robot, permitiendo el monitoreo y graficado de variables en tiempo real. A través de ellas es posible visualizar la diferencia entre los valores objetivos y los reales, así como editar parámetros manualmente.

- Simulación y Gemelo Digital (Digital Twin)

Gracias a la potencia de la que consta el AX-8 es posible realizar simulaciones avanzadas sin tener el robot real, la herramienta que se ha creado para esto es CODESYS Depictor (o Visualización 3D), permite modelar el robot en 3D y vincular cada una de las variables del programa a él .

3.1.4 PRUEBAS REALIZADAS

Previamente a la integración del PLC en el sistema, se han ejecutado una serie de pruebas de diagnóstico y validación. Este proceso de verificación preliminar asegura que tanto el hardware como la lógica programada responden correctamente. Los tipos de pruebas que se realizan en este caso son las siguientes:

- Pruebas Unitarias (Offline): Utilizando el simulador nativo de CODESYS, se verifica el comportamiento del software forzando entradas y variables. Permiten evaluar la respuesta de la lógica de control de forma segura y exhaustiva antes de la conexión con el PLC físico.
- Pruebas de Entrada/Salida Una de las primeras pruebas que se centran en el hardware, confirmar el estado de los botones físicos y su entrada en el PLC además de confirmar la fiabilidad mecánica de las salidas que accionará este

en su tarea. Aunque en este proyecto la mayoría de los botones no son físicos las entradas del PLC serán utilizadas para sensores y por lo tanto deben pasar pruebas similares.

- Protocolo de errores: Este tipo de pruebas simulando fallos mediante el forzado de variables, el envío de coordenadas incorrectas o la alteración del estado del sistema. Una vez detectado el error, se comprueba que la actuación del PLC sea la idónea en cada caso. Estas comprobaciones son cruciales para prevenir posibles accidentes derivados de un mal funcionamiento.

3.1.5 CONTROL DE SEGURIDAD Y VALIDACIÓN

El control de seguridad que implementa un PLC se compone de una serie de dispositivos y complementos del código que se unen para crear una red de seguridad en la máquina que proteja la integridad de esta y sobre todo proporciona seguridad a las personas que trabajan con ella.

Una de las medidas utilizadas y más conocidas es la seguridad por hardware, la tradicional seta que interfiere en el funcionamiento de cualquier parte móvil, esta se incluye en cada apartado de la programación que realice una acción móvil parándola de manera inmediata o en algunas partes del sistema interfiere directamente con la potencia que alimenta a algunos componentes.

Durante todo el programa, existen enclavamientos que interfieren en ciertas acciones, de manera similar, el programa incluye otras protecciones lógicas que interfieren en todo tipo de acciones y evitan el acceso a determinados modos ciertos momentos o sin las comprobaciones previas necesarias, por ejemplo; cuando recibe unas coordenadas solo puede moverse hasta ellas en caso de recibir una respuesta afirmativa del comprobador de límites, de esta manera el robot no tratará de moverse fuera de su área de trabajo, aunque se lo pidan.

Más allá de las restricciones lógicas, resulta imperativo verificar la integridad del sistema de gestión de alarmas. Este módulo debe garantizar la transición hacia un estado seguro ante cualquier anomalía detectada. Además, se ha

implementado un protocolo de jerarquización que permite al sistema priorizar las alertas críticas sobre los avisos informativos, optimizando así la capacidad de respuesta del operario tal y como se detalla en los principios de gestión de alarma de la norma ISA 101 en [10].

3.2 HMI

Para el diseño inicial y durante la fase de desarrollo, se optó por la emulación de la interfaz mediante un PC industrial, descartando temporalmente la integración de un panel físico. Esta decisión, aunque estratégica en la etapa de prototipado, se verá revertida en el apartado de nuevas implementaciones, donde se detalla su sustitución definitiva.

En esta primera fase, el diseño de la pantalla es plenamente funcional a través de la simulación en el PC industrial, facilitando la iteración rápida de la interfaz de usuario, ya que permite realizar una actualización dinámica de esta, pudiendo validar el entorno de control sin necesidad de compilar y transferir los datos a un hardware externo. Al centralizar todo el proceso de control y visualización en un único nodo (el PC), se optimizan los recursos favoreciendo al equipo económicamente al prescindir del coste de adquisición de hardware específico y reducir los tiempos de despliegue asociados al análisis de requisitos de montaje e instalación física de la pantalla.

3.2.1 CARACTERÍSTICAS Y HERRAMIENTAS UTILIZADAS EN SU DESARROLLO.

La aplicación empleada durante el desarrollo y despliegue del emulador ha sido DiaDesigner - HMI. Este software está creado para el desarrollo de interfaces mediante un entorno gráfico avanzado que incluye herramientas necesarias para el diseño, simulación y despliegue de sistemas de supervisión y control.

Dicha solución cuenta con una extensa librería de objetos gráficos parametrizables, estos posibilitan una representación visual del estado del robot, transformando la gestión de variables complejas en información gráfica comprensible para el operario. La herramienta de configuración de

comunicación ha viabilizado la implementación de protocolos industriales estándar, estableciendo un canal robusto fácilmente. Esta arquitectura garantiza la comunicación bidireccional a tiempo real posibilitando el envío de consignas al robot con la latencia mínima característica del proyecto. Para salvaguardar la operativa se ha utilizado un módulo indicado para la gestión de usuarios, optimizando la creación y edición de usuarios lo cual en otras marcas dependería completamente del PLC, en ella se puede restringir cada objeto para restricción niveles de usuarios. Consta de un editor de macros que utiliza el scripting para automatizar cálculos y lógicas mediante variables en el HMI, disminuyendo carga al PLC. Y por último y más importante la posibilidad de simulación tanto online como offline, cuya utilización ha sido clave para realizar una depuración exhaustiva previa a la implementación final.

3.2.2 MODOS DE OPERACIÓN Y PANTALLAS DISEÑADAS.

La interfaz HMI se ha diseñado para proporcionar un control intuitivo y seguro de la celda. El desarrollo del código y pantalla está segmentado en distintos modos de operación, permitiendo que cada pantalla presente únicamente las funciones necesarias para la tarea en curso. Esta separación lógica no sólo dota de mayor claridad visual al operador, sino que refuerza la seguridad operativa al restringir comandos críticos según el modo seleccionado.

En este proyecto se contemplan dos modos de funcionamiento para la ejecución de la tarea Pick & Place, el automático y el manual. El modo manual ha sido diseñado para operar el proceso de forma controlada y pausada, facilitando los ajustes del sistema, las comprobaciones de puntos y el rearmado tras una parada inesperada. Por el contrario, el modo automático está planteado para trabajar de forma cíclica y autónoma; al eliminar las esperas por confirmación del operador se optimizan las trayectorias y se maximiza el rendimiento de la producción.

El modo manual permite realizar diagnósticos y puestas en marcha controladas del sistema puesto que su lógica de control está diseñada para que la transición entre estados requiera la validación del operador desde la interfaz

HMI. Al condicionar cada movimiento a la confirmación del paso anterior, se garantiza un entorno seguro para el ajuste de posiciones y la calibración de puntos, pasos indispensables antes de iniciar la producción en serie.

El modo automático es el estado operativo principal del sistema y está diseñado para la ejecución de ciclos ininterrumpidos. Su característica diferencial es la implementación de blending, una técnica de control de movimiento que, a diferencia del posicionamiento punto a punto convencional, encadena las trayectorias sin detener los accionamientos entre cotas como se establece en las directrices de PLCopen en [9]. Esta gestión de la cinemática permite mantener una inercia constante, reduciendo drásticamente tanto los tiempos de ciclo como el estrés mecánico de los componentes. Al eliminar la necesidad de confirmaciones externas, el sistema garantiza una producción continua por lotes, supervisada únicamente por las rutinas de detección de errores.

PANTALLAS DISEÑADAS Y SU EXPLICACIÓN

Las interfaces se han dividido en pantallas de control y de supervisión. Las de supervisión se destinan exclusivamente a visualizar la producción y el registro de errores. Por su parte, las de control permiten editar parámetros y ejecutar las acciones diarias del sistema, tales como calibraciones, activaciones o paradas de ciclo.

Pantallas de supervisión

- Main: Es el panel principal de monitorización. Permite verificar el estado operativo del robot y da acceso a subpantallas para visualizar la posición de la cinta, las coordenadas de recogida, el estado de los ejes y el vacío de las ventosas.
- Alarms: Interfaz de diagnóstico que muestra el registro histórico de eventos y fallos del sistema.

Nuevas Implementaciones :

- Maintenance: Panel dedicado al seguimiento de los mantenimientos programados de los componentes mecánicos.
- Saved: Permite la consulta de datos históricos de producción almacenados mensualmente.
- Product: Centrada en el control de producción a tiempo real. Aunque permite ajustes menores, como añadir piezas rechazadas, su función principal es el seguimiento del rendimiento actual.

Pantallas de control

- Drives: Permite la edición y ajuste de los parámetros dinámicos de los accionamientos del robot.
- Frames & Tools: Gestión de los sistemas de referencia y calibración de las dimensiones de la herramienta (TCP).
- User: Interfaz para la gestión de perfiles de usuario y niveles de acceso a la máquina.
- Settings: Panel de configuración para los parámetros del movimiento.
- Service Mode: Diseñada para el mantenimiento técnico, permite el movimiento individual (JOG) de cada articulación para pruebas de funcionamiento.
- Points Target & Joints Target: Pantallas específicas para la programación y definición de los puntos de destino en coordenadas cartesianas y articulares.

3.2.3 NIVELES DE USUARIO Y ACCESO

La división de accesos y privilegios es un requisito fundamental para garantizar la integración y seguridad operativa de la producción. Restringir el acceso, evita el riesgo de errores humanos, que los operarios no cualificados puedan modificar variables críticas arriesgando el correcto funcionamiento del sistema y la seguridad de las personas y la planta.

Esta segmentación de accesos se ha creado mediante el uso de usuarios dentro de la interfaz. Tal como se establece en los estándares de seguridad para sistemas de control industrial de la IEC 62443 [13], el uso de una jerarquía de privilegios que asigna a cada operario un usuario individual permite restringir la modificación de parámetros de control además de información sensible dependiendo del nivel de autorización.

La estructura se ha fragmentado en tres estadios, el Nivel Operario, las facultades se limitan al control básico del proceso y a la visualización sobre la producción actual. Por encima de este, el Nivel Mantenimiento desbloquea la interfaz de mantenimiento dando acceso a ajustes de los parámetros de cada alarma. Finalmente, el nivel Seguridad, normalmente conocido como Administrador, este otorga acceso global y sin restricciones a la totalidad de configuraciones de la aplicación, garantizando así la integridad y el control del sistema.

4. PROGRAMACIÓN DE LA LÓGICA DE CONTROL

Después de haber descrito la arquitectura del hardware del sistema este apartado trata de explicar el desarrollo y la implementación del software de control. La programación en un sistema de control es una parte muy importante del proyecto, se encarga de la sincronización de todo el entorno, controla la seguridad operativa y ejecuta cada algoritmo del proceso de manera automatizada.

Como se puede analizar en la Figura 6 se ha implementado una arquitectura modular para optimizar los tiempos de ciclo y garantizar la estabilidad del sistema, siguiendo las directrices de estructuración de programas industriales normalizados en IEC 61131-3 en [6]. En las siguientes secciones se detalla la lógica de cada bloque, cómo interactúan entre sí y los criterios de operación que permiten alcanzar los objetivos de producción.

4.1 EXPLICACIÓN BREVE DE LA ESTRUCTURA DEL PROGRAMA

El sistema está estructurado en cuatro módulos de control: Movimiento del robot (Pick and Place), Movimiento de la cinta principal, Movimiento de la cinta auxiliar, y Cámara/Visión Artificial. La interacción coordinada entre estos bloques y el flujo de datos del sistema se detallan en la Figura 7, donde se presenta el diagrama de flujo de la arquitectura lógica del programa.

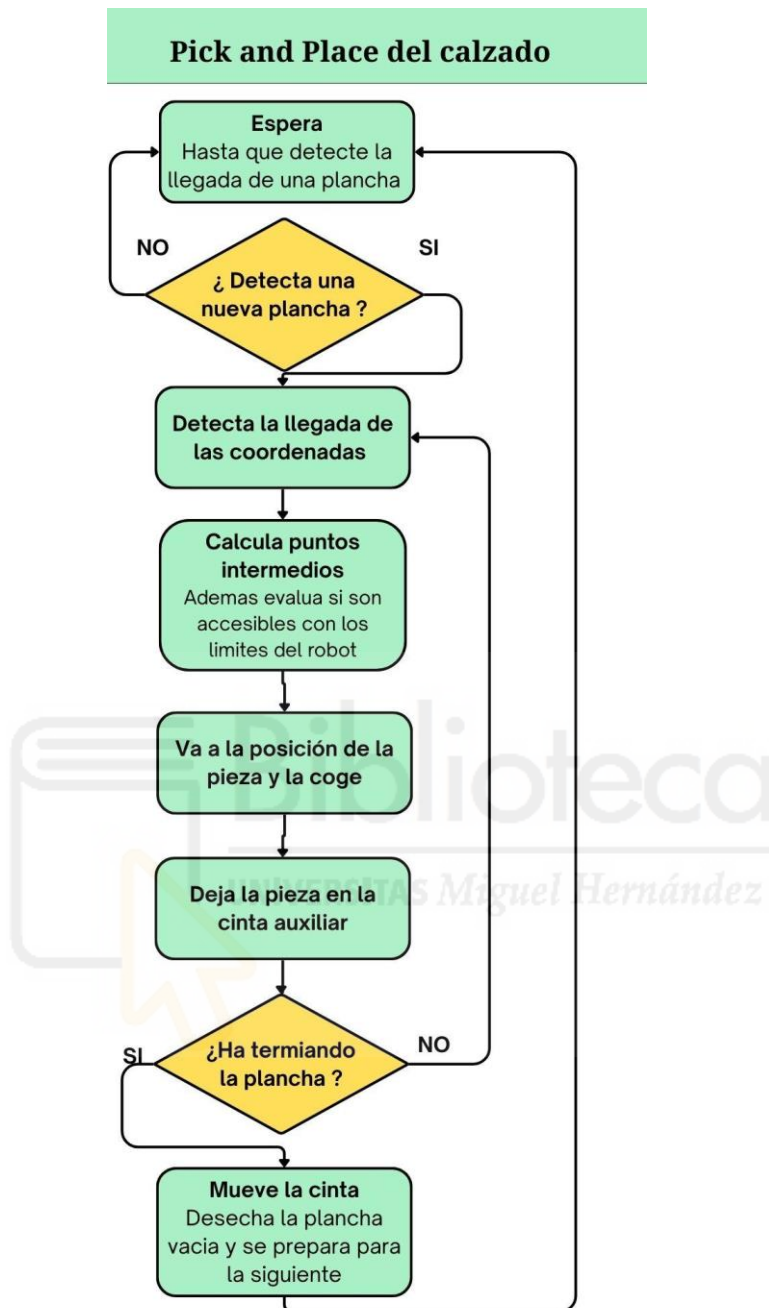
- Pick and Place: Controla el ciclo principal de trabajo, al arrancar la producción, mueve el robot a la posición de inicio para evitar que el ciclo empiece en un punto no deseado. Después, el código lleva al robot a la posición de recogida, donde activa la ventosa y traslada la pieza al punto de descarga en la cinta auxiliar. Además, lleva un recuento de las piezas colocadas para saber cuándo debe avanzar la cinta auxiliar. Debido a su nivel de importancia, se puede ver en forma de diagrama de flujo en la Figura 8.
- Movimiento del Robot : Cuando el robot recibe las coordenadas, comprueba su validez y las va ejecutando, recogiendo las piezas para dejarlas en la cinta auxiliar. Gracias a los límites de posición y velocidad, mantiene un control para evitar salirse de sus límites físicos o interaccionar accidentalmente con su entorno.
- Cinta Principal: Cuando se han recogido todas las piezas de la plancha de material actual, se activa el movimiento de la cinta principal para desechar la plancha que ya no cuenta con ninguna de las piezas recortadas previamente y esperar la detección de la llegada de la siguiente plancha. La cinta principal tiene cambios de comportamientos dependiendo del modo del programa en cada momento, estos se han descrito con más detalle en la Figura 9.
- Cinta Auxiliar: En caso de alcanzar el máximo de piezas en la cinta auxiliar, se activa su movimiento para despejar el área de colocación, cuando ha terminado, el robot continúa con su labor dejando las piezas en ella. Puesto que tiene funcionamientos ligeramente distintos en los dos modos del sistema, se ha implementado la explicación con un diagrama de flujo en la Figura 10.

- Cámara: El proceso se inicia cuando la cámara detecta la llegada de la plancha y comienza a enviar las coordenadas de las piezas detectadas en ella al PLC mediante comunicación Modbus. Envía las coordenadas una a una hasta que se han recogido todas las piezas recortadas de la plancha, y después espera la llegada de la siguiente.
- Control de Errores e Incidencias: El control de errores se realiza de manera centralizada mediante una variable clave llamada índice. Esta variable tomará un valor numérico específico que corresponde al error ocurrido y activará una bandera de estado en cada caso. Esto garantiza la fiabilidad del programa dado que no permite que haya dos errores a la vez, facilitando además la priorización de alertas. Una de las incidencias críticas es la recepción de coordenadas de pick que están fuera del alcance operativo del robot. En este caso, el robot se queda en estado de espera con el error correspondiente encendido en la variable índice, y no reanuda la operación hasta que recibe coordenadas de pick distintas y válidas.

4.2 DIAGRAMAS DE FLUJO

La Figura 6 presenta el diagrama de flujo correspondiente al funcionamiento general de todo el proyecto. Esta representación no detalla la programación interna, sino que ilustra la lógica operativa del conjunto en marcha, facilitando así la comprensión del rol que desempeña el equipo dentro de la planta.

Figura 6. Diagrama de flujo general del programa.

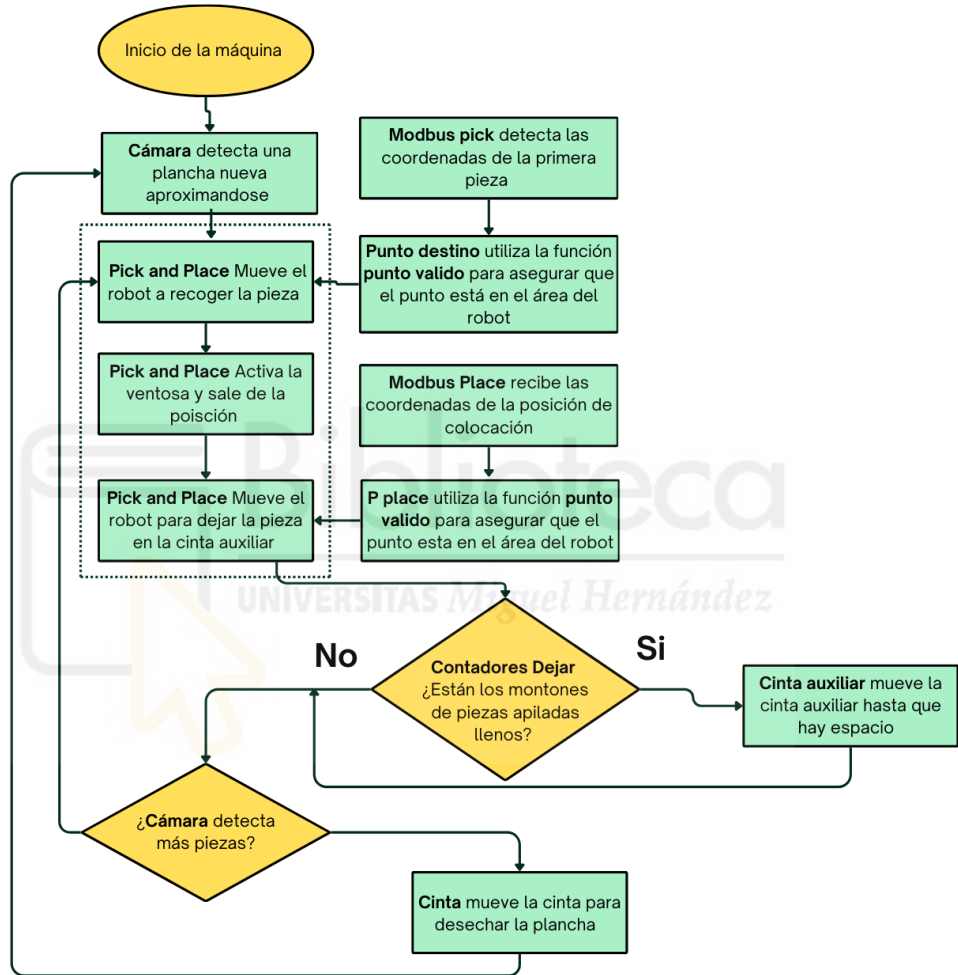


Tras el análisis individual de cada módulo realizado en el apartado anterior, la Figura 7 representa la interconexión entre los bloques lógicos. Esta ilustración describe la secuencia lineal que sigue el código para alcanzar el comportamiento deseado, sin profundizar en algoritmos específicos; su objetivo

es exponer la jerarquía entre las distintas tareas y el propósito que cumple cada una en las diferentes etapas del proceso.

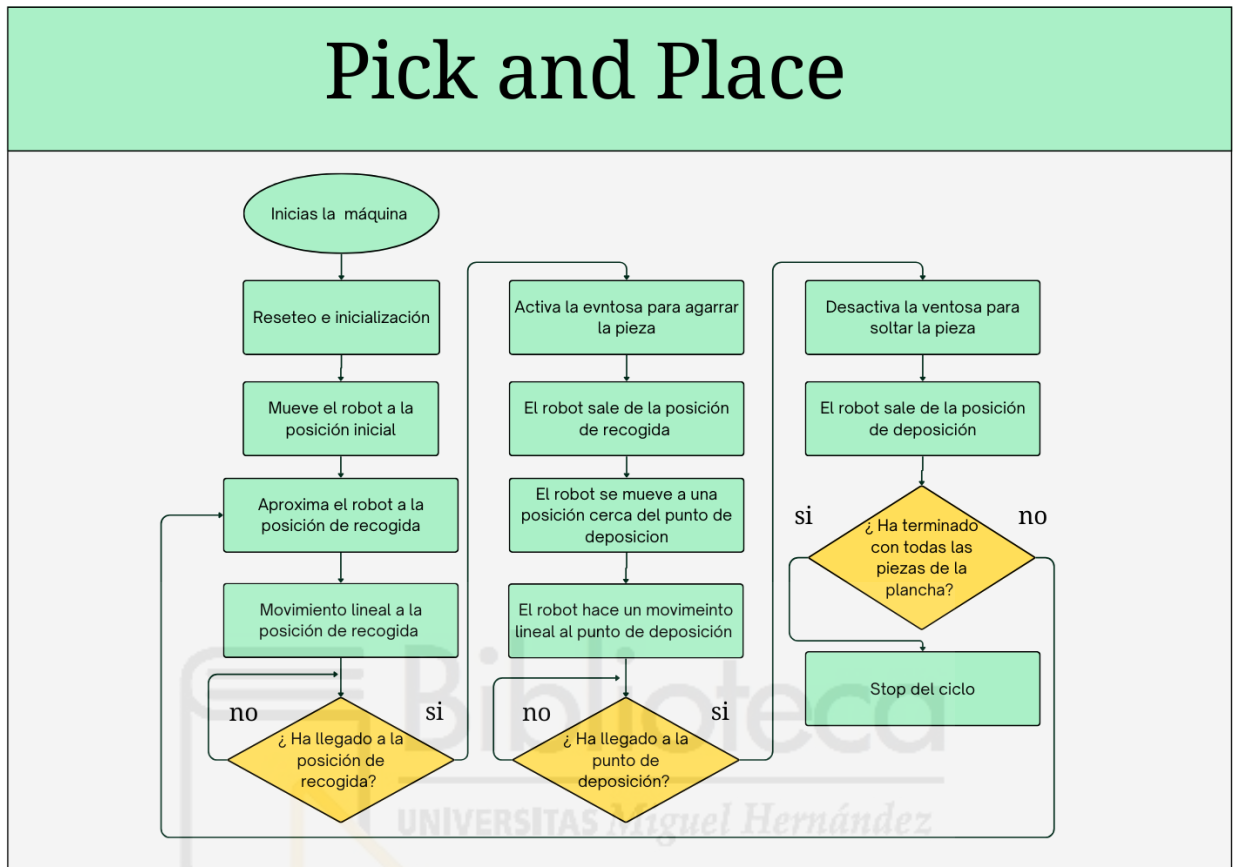
Figura 7. Diagrama de flujo que relaciona las funciones del programa.

Relación entre los programas



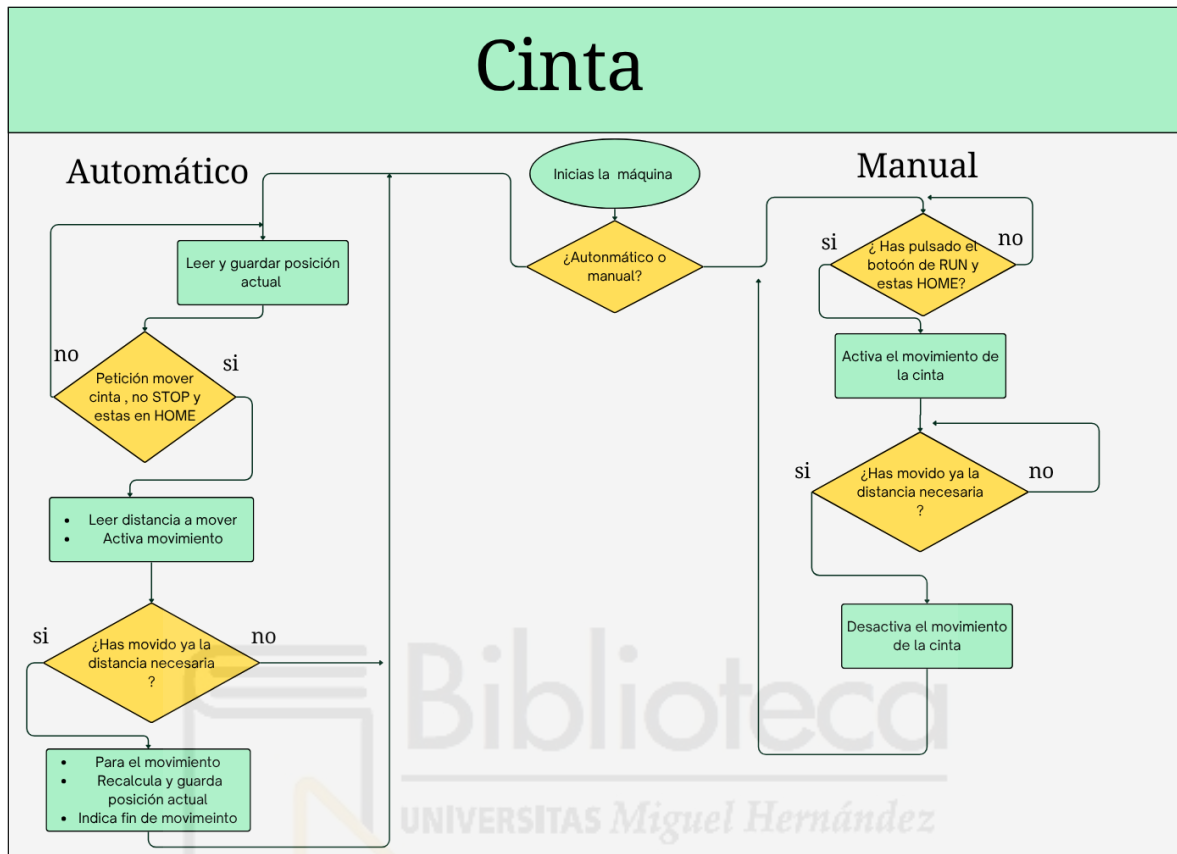
En la Figura 8 se detalla la lógica de control de uno de los módulos principales: el pick and place. En este diagrama se expone el funcionamiento de la rutina de forma secuencial y exhaustiva, especificando tanto las acciones ejecutadas como algunas de las comprobaciones de estado necesarias para completar un ciclo de trabajo íntegro.

Figura 8. Diagrama de flujo que representa el programa que se encarga del movimiento del Robot.



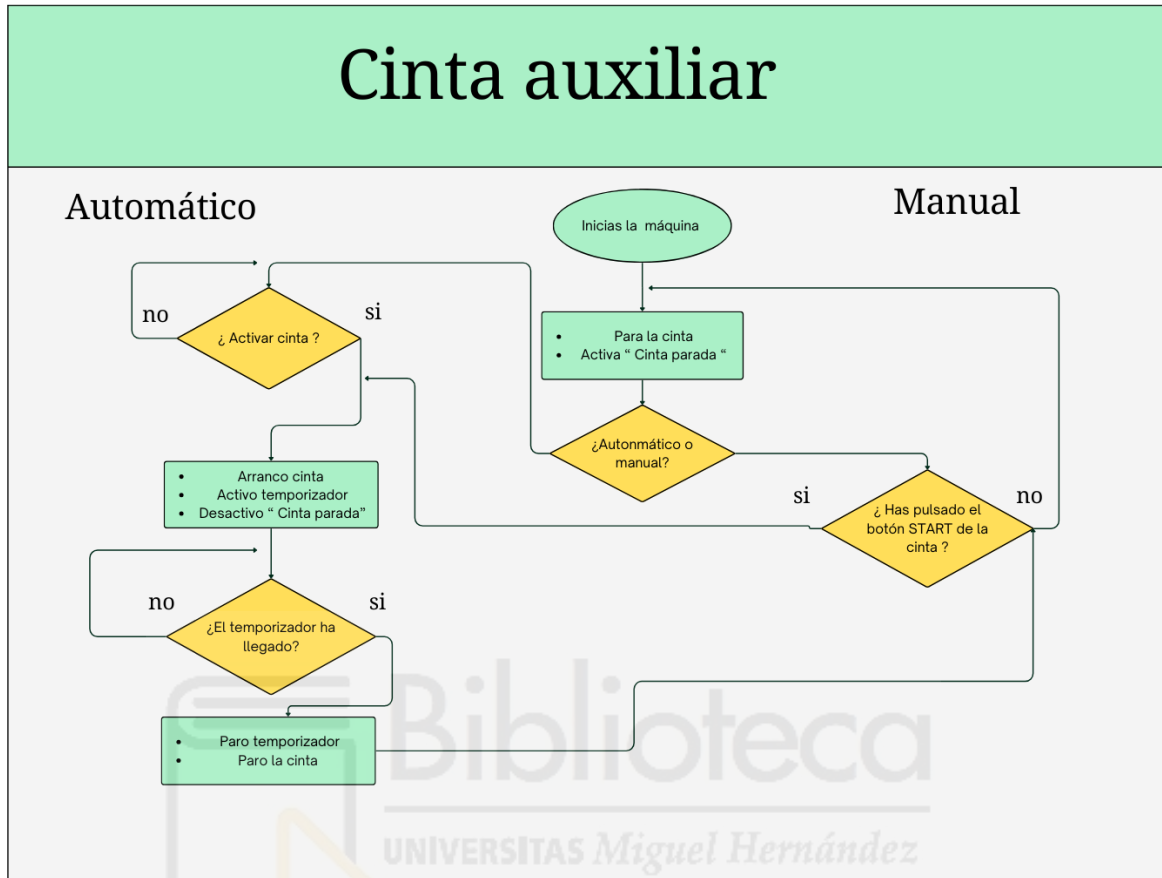
El diagrama de la Figura 9 detalla la secuencia operativa del programa encargado de la cinta principal. En él se distinguen dos ramificaciones lógicas que representan los distintos modos de funcionamiento, permitiendo apreciar las comprobaciones y validaciones específicas que el sistema ejecuta en cada uno para su continuidad.

Figura 9. Diagrama de flujo del programa que se encarga del movimiento de la cinta principal.



La Figura 10 detalla los pasos del programa y la distinción operativa entre los modos manual y automático. En ella se observa que ambas modalidades comparten la misma lógica secuencial, con la única diferencia de que el modo manual requiere la intervención directa del operario para autorizar cada movimiento, condicionando así el avance del ciclo a las órdenes recibidas.

Figura 10. Diagrama de flujo del programa que se encarga del movimiento de la cinta auxiliar.



5. NUEVAS IMPLEMENTACIONES DEL SISTEMA

Para este sistema ya completamente funcional se sugieren varias actualizaciones que permiten mejorar su funcionamiento. Se ha creado e implementado la programación necesaria para llevar un control de la producción obtenido a tiempo real en la interfaz, creando y gestionando un historial. Se propone también la adición de un administrador de mantenimiento adaptable en función de las necesidades del sistema. Por último, se ha seleccionado y estudiado la conexión de una pantalla HMI que ocupe el rol que actualmente desempeña un PC, haciendo mediante un análisis de necesidades.

5.1 PANTALLA DE CONTROL DE PRODUCCIÓN

La finalidad de esta función añadida es agilizar el análisis tanto diario como posterior del flujo de producción y errores para conseguir una optimización continua de esta. Para ello se ha diseñado una visualización a tiempo real de cada variable determinante de la producción, y una base de datos gestionada mediante diferentes turnos de trabajo, facilitando su análisis.

5.1.1 ARQUITECTURA DE PROGRAMACIÓN

Tal como se ha visto en las secciones anteriores, la arquitectura del software del programa principal se basa en un enfoque modular y jerárquico: un código principal coordina al resto de módulos, los cuales se ocupan de tareas específicas. En esta implementación se ha mantenido dicha lógica, con el objetivo de dar concordancia al sistema y facilitar sus ajustes futuros.

Los módulos que integran el programa de monitorización son los siguientes:

- Producción: Es el corazón del programa, contiene temporizadores y algoritmos que se ejecutan en vivo durante la jornada y permite visualizar el historial de fabricación de días anteriores.
- Gestor de Turnos: Se encarga de controlar los tiempos de cada jornada y de registrar los datos recopilados al completar cada turno, organizando la información para su adecuado almacenamiento.
- Rendimiento: Este bloque realiza los cálculos más complejos al cierre del día (disponibilidad, calidad y rendimiento). Estos valores se guardan para que el equipo pueda consultarlos durante la jornada siguiente.

La distribución de las tareas permite realizar un seguimiento en vivo de la fabricación, almacenar y consultar producciones pasadas para realizar análisis frecuentes, evaluar la eficiencia del día anterior para comprobar la mejora continua y administrar los turnos de trabajo de forma organizada.

5.1.2 DESARROLLO DE LA PROGRAMACIÓN

A continuación, se desglosan las funciones de la pantalla de producción, se ha descrito cada bloque con el apoyo de su correspondiente segmento de código.

El programa que se ocupa del control de la producción se organiza en torno a la estructura de un Turno, este actúa como la unidad de información principal donde se recopilan y almacenan todos los parámetros operativos de la fabricación en cada turno.

El comienzo del conteo a tiempo real se ejecuta de forma automática al comenzar la producción, se contabilizan tanto las piezas correctamente procesadas como las rechazadas, además de registrarse la duración de la jornada y los intervalos de inactividad derivados de errores en el sistema.

Este módulo se encarga de calcular, en tiempo real, el porcentaje de cumplimiento del objetivo de producción y de actualizar todos los indicadores visualizados en la Interfaz que se muestra en la Figura 18. Entre estos datos se incluyen la velocidad de procesamiento actual y la duración exacta del último ciclo, definido este último como el intervalo necesario para la recogida de una plancha completa. Asimismo, a través de la pantalla de control, el sistema permite gestionar el reinicio de los registros y la consulta de datos históricos de manera directa mediante la pantalla expuesta en la Figura 20 y su reinicio.

La programación central de esta mejora está compuesta por varias partes, comienza en la Figura 11 con algunas de las comprobaciones lógicas de las que se ocupa.

Figura 11. Primera parte de la programación realizada para la función denominada Producción.

```

// INICIO DE TURNO, cuando no se ha iniciado turno y detecta que se produce una pieza
IF Control_Produccion.Turno_actual.Activo=FALSE AND (Contadores_Dejar.Zona_Derecha_A
Control_Produccion.Turno_actual.Activo := TRUE;
Piezas_minuto := 1; //Resetea la velocidad del último minuto
END_IF;
//PIEZA PRODUCIDA, siempre y cuando el turno ya se haya declarado como activo
IF Control_Produccion.Turno_actual.Activo=TRUE AND (Contadores_Dejar.Zona_Derecha_A OR (
Control_Produccion.Piezas_turno:= Control_Produccion.Piezas_turno+1;
Piezas_minuto:= Piezas_minuto+1;
END_IF;
//PIEZA RECHAZADA, siempre que se active la señal de rechazo se sumará
IF R_TRIG_B_RECHAZO.Q OR R_TRIG_ERROR.Q THEN
Control_Produccion.Piezas_rechazadas := Control_Produccion.Piezas_rechazadas + 1 ;
END_IF;
// DETECTA ERROR, activa temporizador de tiempo en parada
IF indice= 1 OR indice= 2 OR indice= 3 THEN
ERROR_ALCANCE:=TRUE;
ELSE
ERROR_ALCANCE:=FALSE;
END_IF;
// Reseteo producción último mes, resetea el contador de días guardados
IF R_TRIG_RESET.Q THEN
Control_Produccion.Guardado_lleno:= FALSE;
Control_Produccion.idx_guardado := 1;
END_IF;

```

En la Figura 12 añade la lógica que permite al operario ajustar el objetivo de la jornada a petición, en caso de no requerir se atribuye un objetivo por defecto. También incluye la copia del día seleccionado en el historial para mostrarlo.

Figura 12. Segunda parte de la programación realizada para la función denominada Producción.

```

// Copiamos el día que vayamos a mostrar de los guardados, si estamos intentando mostrar uno no guardado se muestra vacío
IF (Control_Produccion.idx_guardado <> 0) AND (Control_Produccion.NMostrar <> 0 ) AND (Control_Produccion.NMostrar <=Cont
Control_Produccion.Dia_mostrar := Control_Produccion.Prod_guardada[Control_Produccion.NMostrar];
ELSE
Control_Produccion.Dia_mostrar:= Dia_Vacio ;
END_IF
//----- Cálculos
// El botón objetivo se pulsará cuando se haya introducido un número( objetivo de turno )
// Es un botón que permanece pulsado
IF B_Objetivo THEN
Control_Produccion.Ritmo_esperado := INT_TO_REAL(Control_Produccion.Objetivo_prod) / 480.0;
Control_Produccion.Porcentaje := 100.0 * (INT_TO_REAL(Control_Produccion.Piezas_turno) / INT_TO_REAL(Control_Produccion.C
ELSE // Si el trabajador no pone su propio objetivo, se pone el estándar 8000
Control_Produccion.Ritmo_esperado := Objetivo_defecto / 480.0;
Control_Produccion.Porcentaje := 100.0 * (INT_TO_REAL(Control_Produccion.Piezas_turno) / objetivo_defecto);
END_IF;

```

Por último, en la Figura 13 se muestra toda la parte de la programación que permite llevar un control a tiempo real, la iniciación de los temporizadores y los pasos que se llevan a cabo para su funcionamiento.

Figura 13. Tercera parte de la programación de producción, temporizadores.

```

TEMP_MINUTO (IN := Control_Produccion.Turno_actual.Activo, PT := T#1M);
TEMP_TURNO ( IN:= Control_Produccion.Turno_actual.Activo, PT:= T#8H);
TEMP_CICLO (IN:= NOT CintaEnMovimiento , PT:=T#24H);

//GUARDADO DE TIEMPO DE CADA TURNO, Logica de recuperación de tiempo de turno en caso de reinicio
IF Control_Produccion.Turno_actual.Activo THEN
    Control_Produccion.Turno_actual.Minutos := (TIME_TO_REAL(TEMP_TURNO.ET + Control_Produccion.Tiempo_turno) / 60000.0);
    Control_Produccion.Tiempo_Turno := REAL_TO_TIME(Control_Produccion.Turno_actual.Minutos * 60000.0);
END_IF
// PIEZAS PRODUCIDAS EN EL ÚLTIMO MINUTO
IF TEMP_MINUTO.Q THEN
    Control_Produccion.Velocidad_Minuto := Piezas_minuto;
    TEMP_MINUTO(IN := FALSE);
    Piezas_minuto:=0;
END_IF;

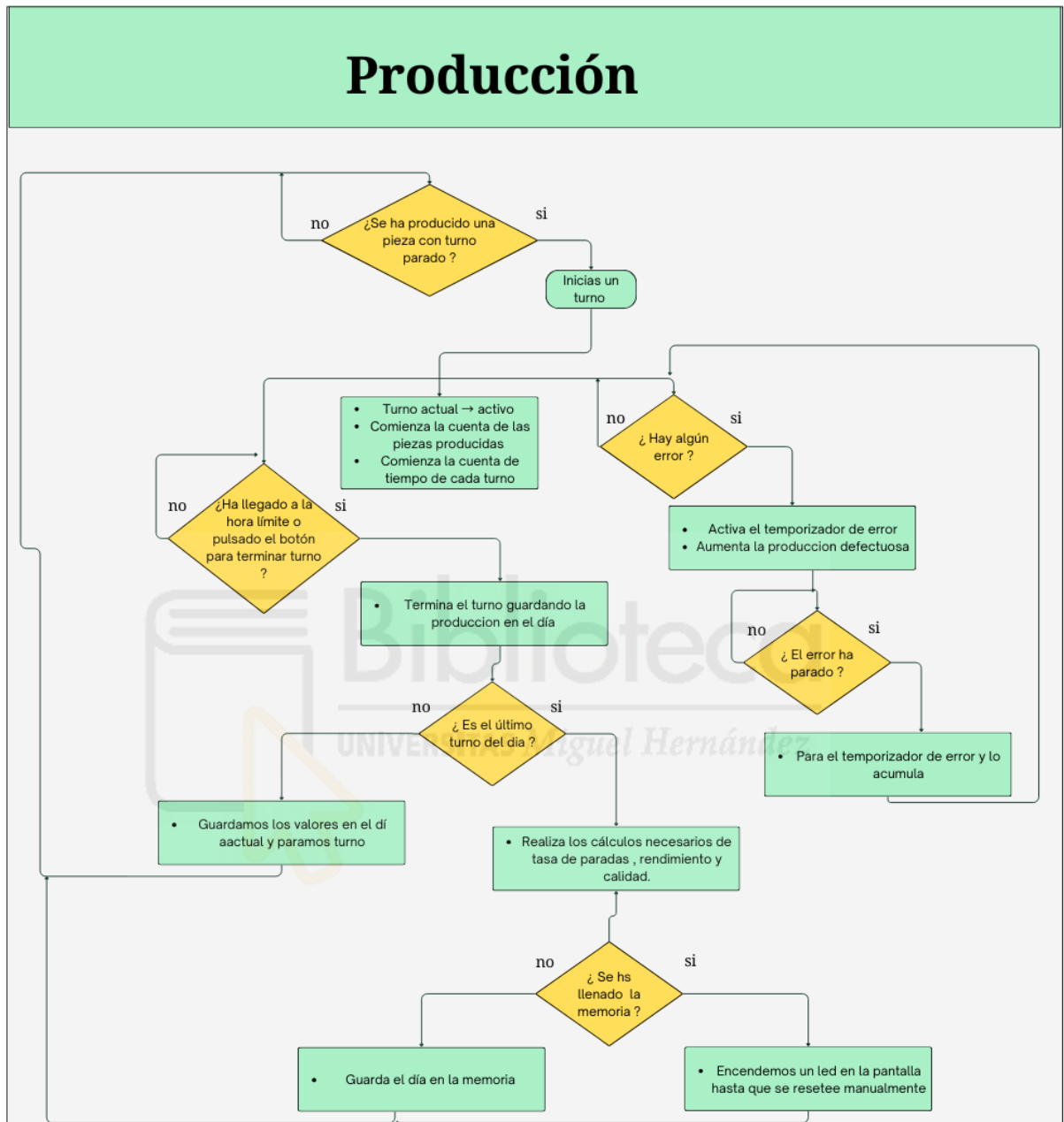
// TIEMPO DE CICLO (CICLO = CAMBIO DE PLANCHA )
IF R_TRIG_CINTA.Q THEN
    Control_Produccion.Tiempo_ciclo:= TEMP_CICLO.ET;
END_IF;

// TIEMPO DE JORNADA QUE PASA PARADO EL SISTEMA POR ERROR
IF R_TRIG_ERROR.Q THEN
    ERROR_ACTIVO:= TRUE;
    Control_Produccion.Tiempo_inicio_error:=TIME();
END_IF;
IF ERROR_ACTIVO AND NOT (ERROR_ALCANCE) THEN
    Control_Produccion.Tiempo_error_turno:= Control_Produccion.Tiempo_error_turno+(TIME()-Control_Produccion.Tiempo_inicio_error);
    ERROR_ACTIVO:= FALSE;
END_IF;

```

Como visibilizar la explicación de la programación es fundamental, se ha representado diagrama de flujo de la Figura 14, este esquema permite repasar paso a paso el desarrollo de la lógica seguida.

Figura 14. Diagrama de flujo de la lógica de control para la interfaz de monitorización de producción.



GESTOR DE TURNOS

Esta función coordina la finalización de la jornada, activándose de forma automática tras 4 horas de actividad o de manera manual a través de la interfaz como se puede observar en la Figura 19. Al concluir cada turno, el sistema almacena los datos en el registro del día y reinicia los parámetros necesarios para la siguiente etapa. En caso de tratarse del segundo y último turno de cada día de trabajo, el programa procede además a realizar los cálculos de rendimiento, disponibilidad, calidad y OEE, actualizando estos indicadores en la pantalla del operario y efectuando el guardado definitivo de toda la información en la memoria del PLC.

En la Figura 15 se hacen los guardados necesarios al terminar cada turno, se actualiza el tiempo en error en caso de que el cambio de turno se haga durante este.

Figura 15. Primera parte de la programación del Gestor de turnos.

```
IF Enable THEN
//FIN DE CADA TURNO
IF idx_turno < 3 THEN
// SI HAY ERROR ACTIVO, LO PARA PARA LA CORRECTA ACTUALIZACIÓN
IF Error_activo THEN
Error_activo:= FALSE;
Control_Produccion.Tiempo_error_turno:= Control_Produccion.Tiempo_error_turno+(TIME()-Control_Produccion.Tiempo_inicio_error);
END IF;
//GUARDA DATOS DE PRODUCCIÓN
Control_Produccion.Turno_actual.diPiezasFabricadas:= Control_Produccion.Piezas_turno;
Control_Produccion.Turno_actual.diPiezasRechazadas := Control_Produccion.Piezas_rechazadas;
Control_Produccion.Turno_actual.Minutos_parado := TIME_TO_REAL(Control_Produccion.Tiempo_error_turno)/60000.0;
Control_Produccion.Turno_actual.Minutos := TIME_TO_REAL (TempTurnoET)/60000.0 ;
Control_Produccion.Turno_actual.Activo := FALSE;
// CALCULO DE VELOCIDAD POR HORAS
HorasReales:= Control_Produccion.Turno_actual.Minutos/60.0;
IF HorasReales <> 0 THEN
Control_Produccion.Turno_actual.Piezas_hora := INT_TO_REAL( Control_Produccion.Piezas_turno)/ HorasReales;
ELSE
Control_Produccion.Turno_actual.Piezas_hora := 0.0;
END IF;
// GUARDA EL TURNO EN EL DIA, PASA AL SIGUIENTE TURNO
Control_Produccion.Dia_actual[idx_turno] := Control_Produccion.Turno_actual;
Control_Produccion.Turno_actual := Turno_Vacio;
idx_turno := idx_turno + 1;
Control_Produccion.Piezas_turno:=1;
Control_Produccion.Piezas_rechazadas:=0;
Control_Produccion.Tiempo_error_turno:= T#0S;
Control_Produccion.Tiempo_turno := T#0S;
```

En la Figura 16 se observa la segunda parte del código del gestor de turnos, únicamente se contempla la posibilidad del final del día y llamado de la función Rendimiento analizada posteriormente. Se comprueba que no se ha llegado al

máximo de días almacenado en el historial y en ese caso se notifica mediante la pantalla HMI.

Figura 16. Segunda parte de la programación del Gestor de turnos.

```
// EN CASO DE FINALIZAR JORNADA
IF idx_turno = 3 THEN
  //RENDIMIENTO, CALIDAD Y TASAS DE PARADA
  calculado:= Rendimiento(Control_produccion.Dia_actual);
  // GUARDA EL DIA
  IF NOT Control_Produccion.Guardado_lleno THEN
    Control_Produccion.Prod_guardada[Control_Produccion.idx_guardado] := Control_Produccion.Dia_actual;
    Control_Produccion.idx_guardado := Control_Produccion.idx_guardado + 1;
  // AUMENTA EL CONTADOR DÍAS GUARDADOS
  IF Control_Produccion.idx_guardado > 29 THEN
    // CONTADOR LLENO
    Control_Produccion.Guardado_lleno:= TRUE;
  ELSE
    Control_Produccion.Guardado_lleno:= FALSE;
  END_IF;
END_IF;
// RESET
Control_Produccion.Dia_actual := Dia_Vacio;
idx_turno := 1;
END_IF;
END_IF;
```

RENDIMIENTO

Al final del día se ejecuta la función Rendimiento mostrada en las Figuras 17 y 18, formada por una rutina de cálculos destinada a ofrecer una visualización del rendimiento al operario correspondiente, permitiendo a este observar los cambios obteniendo una comparativa clara de cada día. Se ha decidido no ocupar memoria del PLC con estos resultados, ya que los datos utilizados para el cálculo sí se guardan para su exportación. Así, cualquier cálculo puede volver a generarse fuera del sistema cuando sea necesario sin saturar el controlador.

Los cálculos realizados comprenden la disponibilidad, la calidad, el rendimiento y el OEE (Overall Equipment Effectiveness); estos cuatro indicadores son esenciales para permitir una evaluación precisa de la producción por diversas razones. En primer lugar, la disponibilidad mide el tiempo productivo real frente al tiempo planificado previamente, de modo que, ante un valor bajo, el análisis permite concluir un exceso de tiempos muertos, ya sea por averías o paradas no programadas. Por otro lado, el rendimiento facilita la comparación entre la velocidad real de manufactura y la capacidad teórica máxima, detectando el

impacto de las microparadas o las disminuciones de ritmo debidas al desgaste mecánico o a fluctuaciones de potencia en el sistema. Finalmente, la calidad representa la relación entre las unidades fabricadas conformes y la totalidad del output, siendo un factor crítico en sistemas de alta velocidad donde cualquier desviación podría generar un alto índice de errores en poco tiempo.

En conclusión, cada indicador analiza una parcela específica del funcionamiento del sistema, pero es al observar el OEE cuando se resume el estado global de la producción, ya que este índice es el resultado de consolidar los tres factores anteriores. Esta métrica ofrece una visión equilibrada y real de la salud general de la planta; no obstante, resulta fundamental monitorizar cada factor de forma independiente para identificar de inmediato el origen exacto de cualquier anomalía detectada, garantizando así una respuesta técnica rápida y eficiente. Estos algoritmos únicamente se ejecutan al final del día y no tras cada turno para ofrecer una visión completa de la jornada y evitar los desajustes de cada relevo ofreciendo una planificación más robusta.

En la Figura 17 se detalla el cálculo del rendimiento, la disponibilidad, la calidad y la OEE del proceso durante la jornada.

Figura 17. Programación de la función rendimiento.

```
//Suma de turnos
TotalMinutos := Dia[1].Minutos + Dia[2].Minutos;
TotalParado := Dia[1].Minutos_parado + Dia[2].Minutos_parado;
TotalFab := Dia[1].diPiezasFabricadas + Dia[2].diPiezasFabricadas;
TotalRech := Dia[1].diPiezasRechazadas + Dia[2].diPiezasRechazadas;

// El rendimiento se calculara siempre del dia anterior
Control_Produccion.Rendimiento:= 100.0*(TotalFab) /24000.0;
// Calidad piezas total - rechazadas / total *100
IF TotalFab + TotalRech <> 0 THEN
    Control_Produccion.Calidad := 100.0*(TotalFab- Totalrech)/(TotalFab);
ELSE
    Control_Produccion.Calidad:= 0.0;
END_IF;

// Disponibilidad (Tiempo total / Tiempo que no esta parada) y TASA DE PARADAS
IF Control_Produccion.Turno_actual.Minutos <> 0 THEN
    Control_Produccion.Disponibilidad := 100.00*((TotalMinutos - TotalParado)/TotalMinutos) ;
ELSE
    Control_Produccion.Disponibilidad :=0.0;
END_IF;

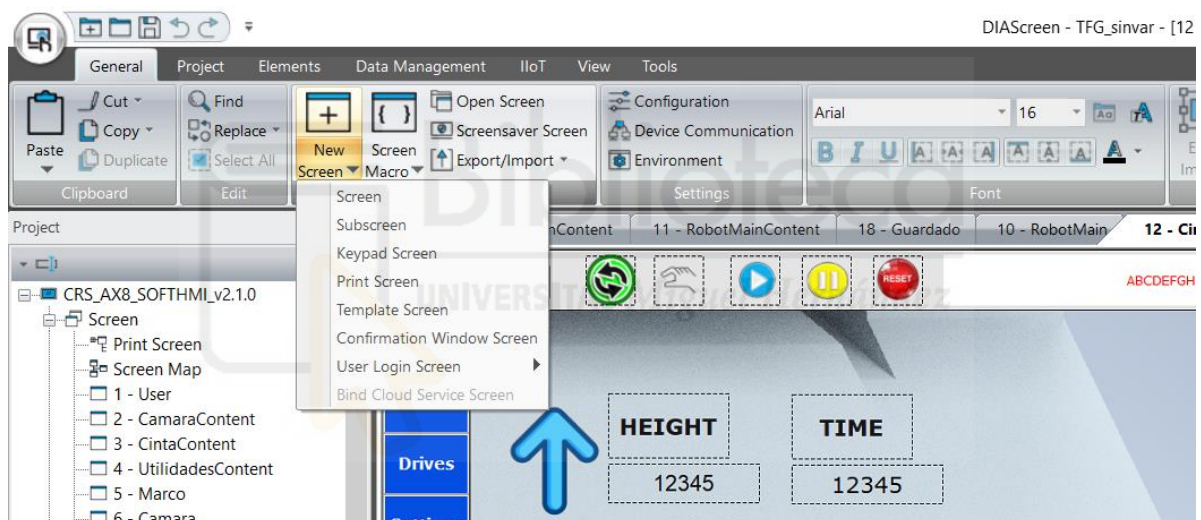
// oee= disponibilidad * rendimiento * calidad
Control_Produccion.OEE:=Control_Produccion.Disponibilidad*Control_Produccion.Rendimiento*Control_Produccion.Calidad/10000.0;
```

5.1.3 DISEÑO DE LA INTERFAZ

Para el diseño de la interfaz actual del programa se ha utilizado la aplicación DiaScreen, a continuación, se muestran las principales herramientas que se han usado para la ampliación descrita anteriormente.

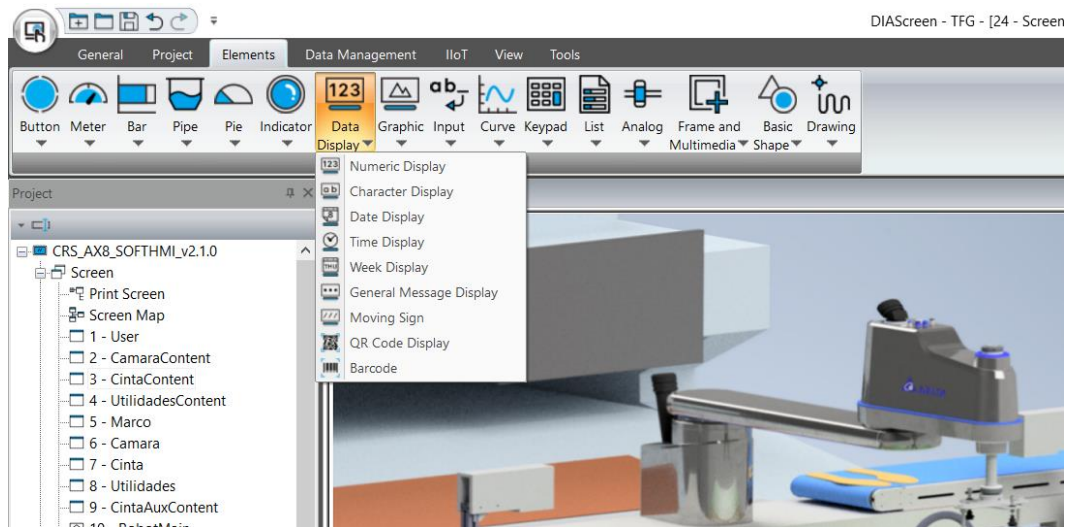
Para generar una interfaz editable, accedemos al apartado general y seleccionamos la opción 'New Screen'. Como se muestra en la Figura 18, el software ofrece diversos tipos de plantillas, desde un lienzo convencional en blanco u otros formatos como ventanas emergentes hasta configuraciones predefinidas diseñadas específicamente para la gestión de acceso de usuarios.

Figura 18. Situación de los tipos de pantalla.



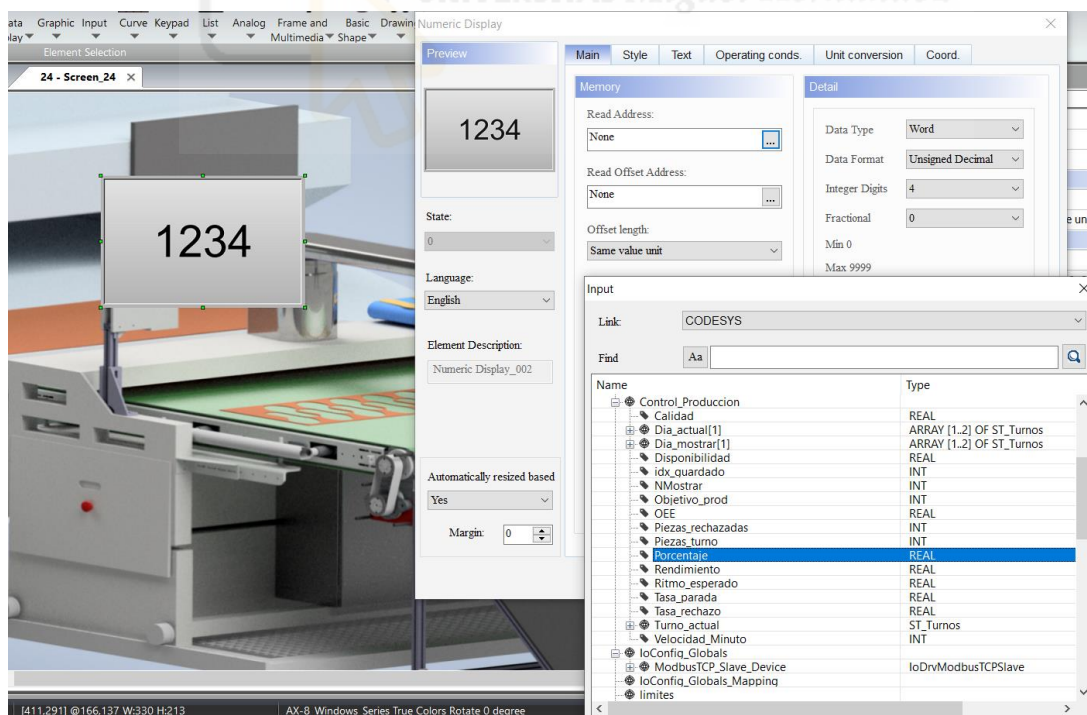
El Numeric Display es el objeto más recurrente en la pantalla de producción debido a la naturaleza informativa de la pantalla. En la Figura 19 se ilustra su localización en el programa.

Figura 19. Situación de los elementos Data Display.



Este elemento tiene muchas opciones a la hora de configurarlo, se van a describir las más relevantes, el primer paso siempre es seleccionar la variable que vaya a representar, ya sea importada del PLC directamente o pertenezca a la propia memoria del HMI, en la Figura 20 vemos un ejemplo de la selección realizada según la lista de datos compartidos de CODESYS.

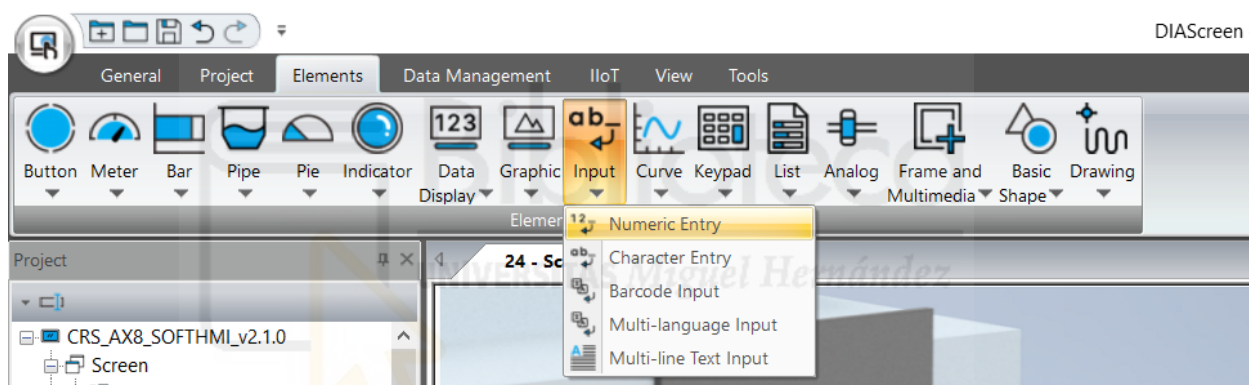
Figura 20. Ajustes del objeto Numeric Display.



Tras seleccionar la variable a representar, el programa ofrece inicialmente una sugerencia automática sobre el tipo de dato que podría corresponderle. No obstante, es fundamental revisar y asegurar esta configuración según la naturaleza del valor garantizando una lectura correcta. Por ejemplo, al mostrar un porcentaje (tipo real), se debe seleccionar el tipo Double Word en formato Floating, asegurando que los decimales se visualicen sin errores de interpretación.

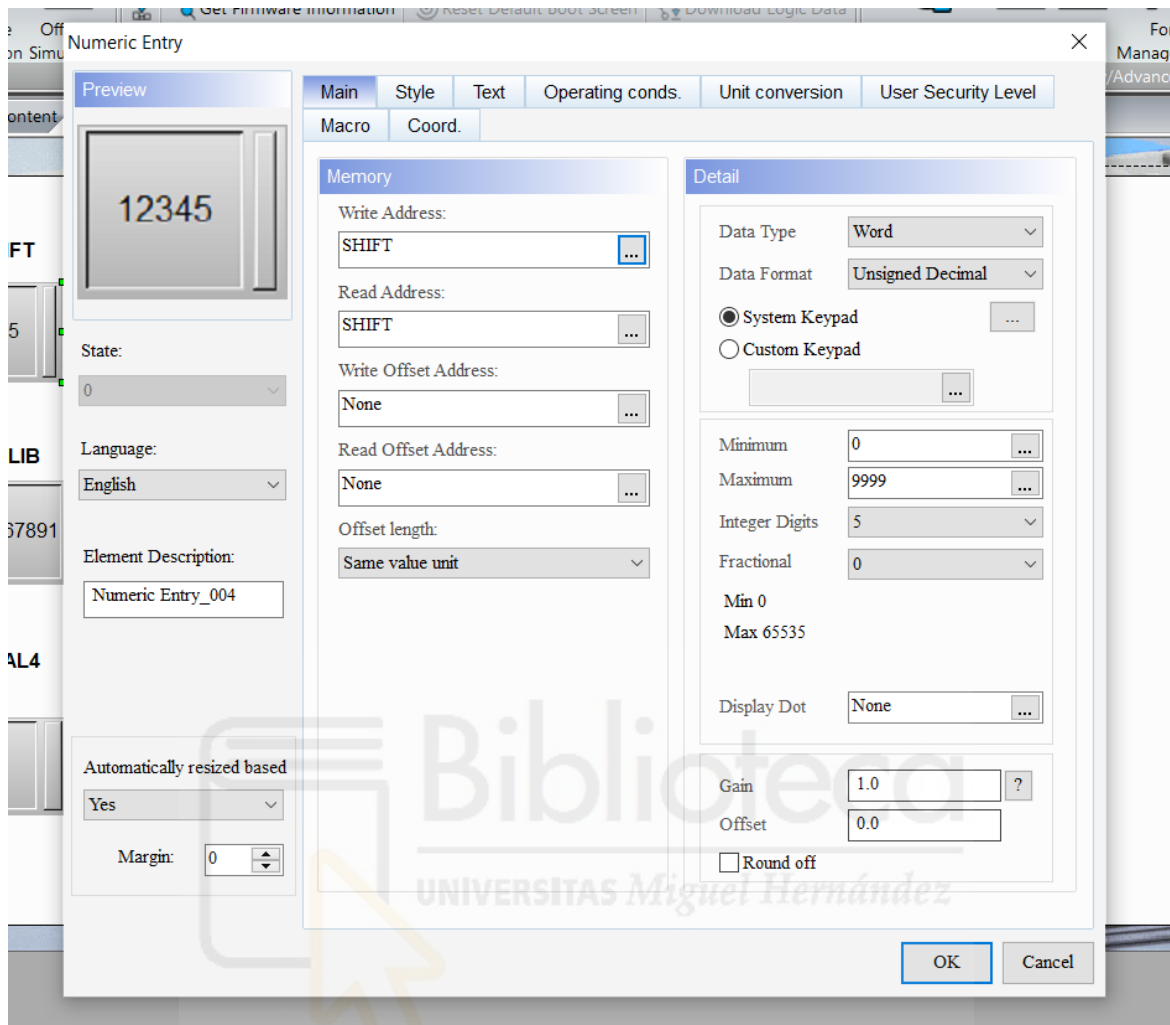
Otro componente esencial es el Numeric Entry, ubicado en el apartado de Input mostrado en la Figura 21.

Figura 21. Situación de los elementos de Input.



Este objeto permite la introducción de valores numéricos destinados tanto a registros del PLC como a cálculos internos de la propia pantalla. Como se observa en la configuración de la Figura 22, la definición del tipo de objeto es crítica, permitiendo establecer límites máximos y mínimos, habilitar el uso de decimales y aplicar ganancias o retardos. Además, como se aprecia en la barra superior de configuración, existen múltiples opciones que trascienden los ajustes estéticos; entre ellas, la realización de conversiones de unidades y la restricción de acceso mediante niveles de usuario, una función clave empleada habitualmente en la gestión de alarmas de mantenimiento.

Figura 22. Ajustes del objeto Numeric Entry.



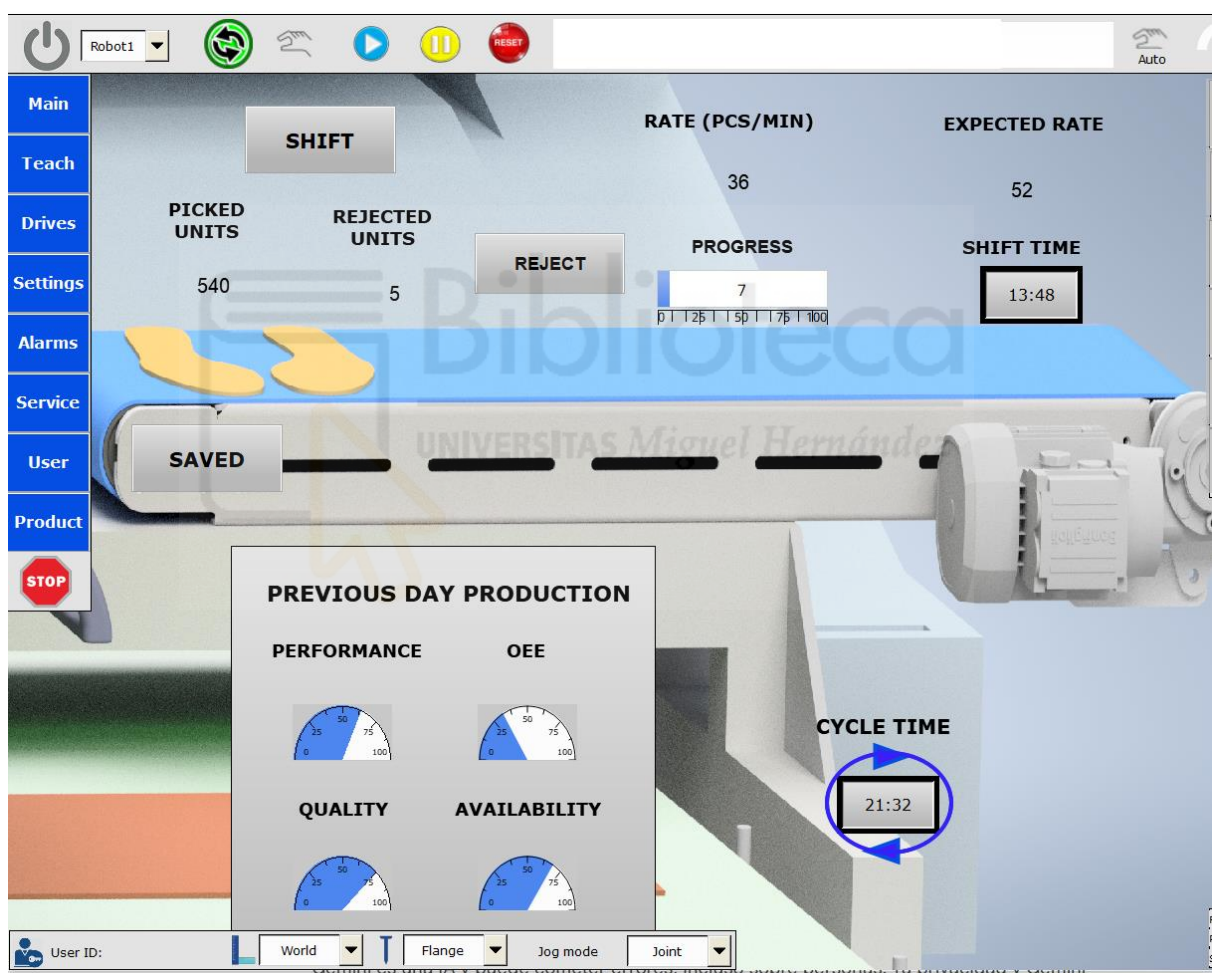
Las interfaces de uso instaladas que se muestran en las Figuras 23, 24 y 25 dinamizan el funcionamiento de la máquina facilitando el control de esta en un entorno visual intuitivo, esto reduce los tiempos de respuesta del operador. Gracias a la Figura 26 se puede visualizar los periodos de mayor productividad mediante una tabla dinámica, permitiendo identificar patrones de eficiencia y optimizar el trabajo.

Adicionalmente, mediante la arquitectura de la aplicación, se ha creado una jerarquía de usuarios robusta, la cual garantiza la diferenciación de perfiles limitando los controles del sistema a cada uno de los roles de cada usuario. Esto además de reforzar la seguridad del uso, facilita la gestión administrativa

clara, y permite que la aplicación se adapte a las necesidades específicas de cada nivel de mando del sistema.

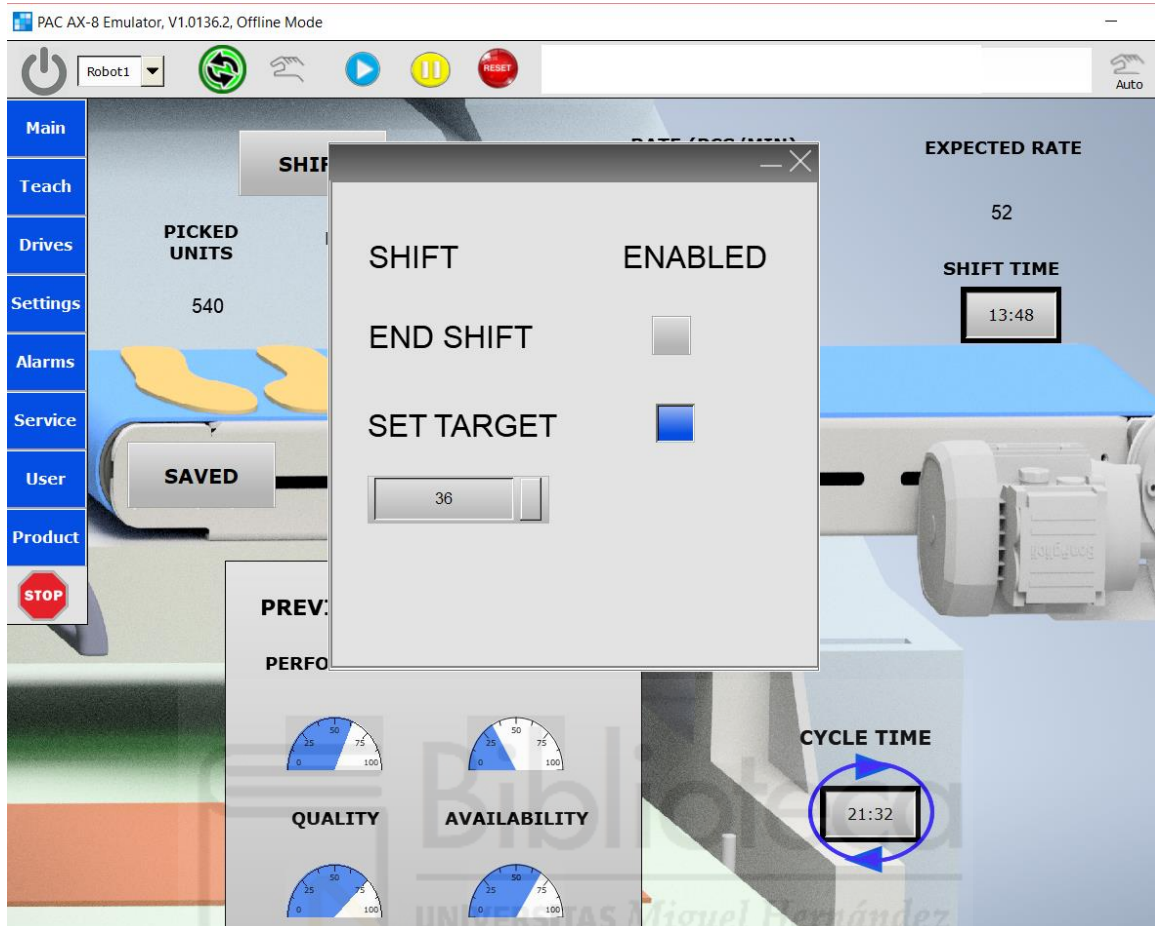
La figura 23 muestra a primera vista la información necesaria de la producción actual al alcance del usuario. Permite visualizar la OEE de la jornada anterior para una fácil observación continua, el tiempo del último ciclo y acceso a los ajustes del turno y objetivo.

Figura 23. Interfaz de usuario del software de control de producción durante la fase de simulación.



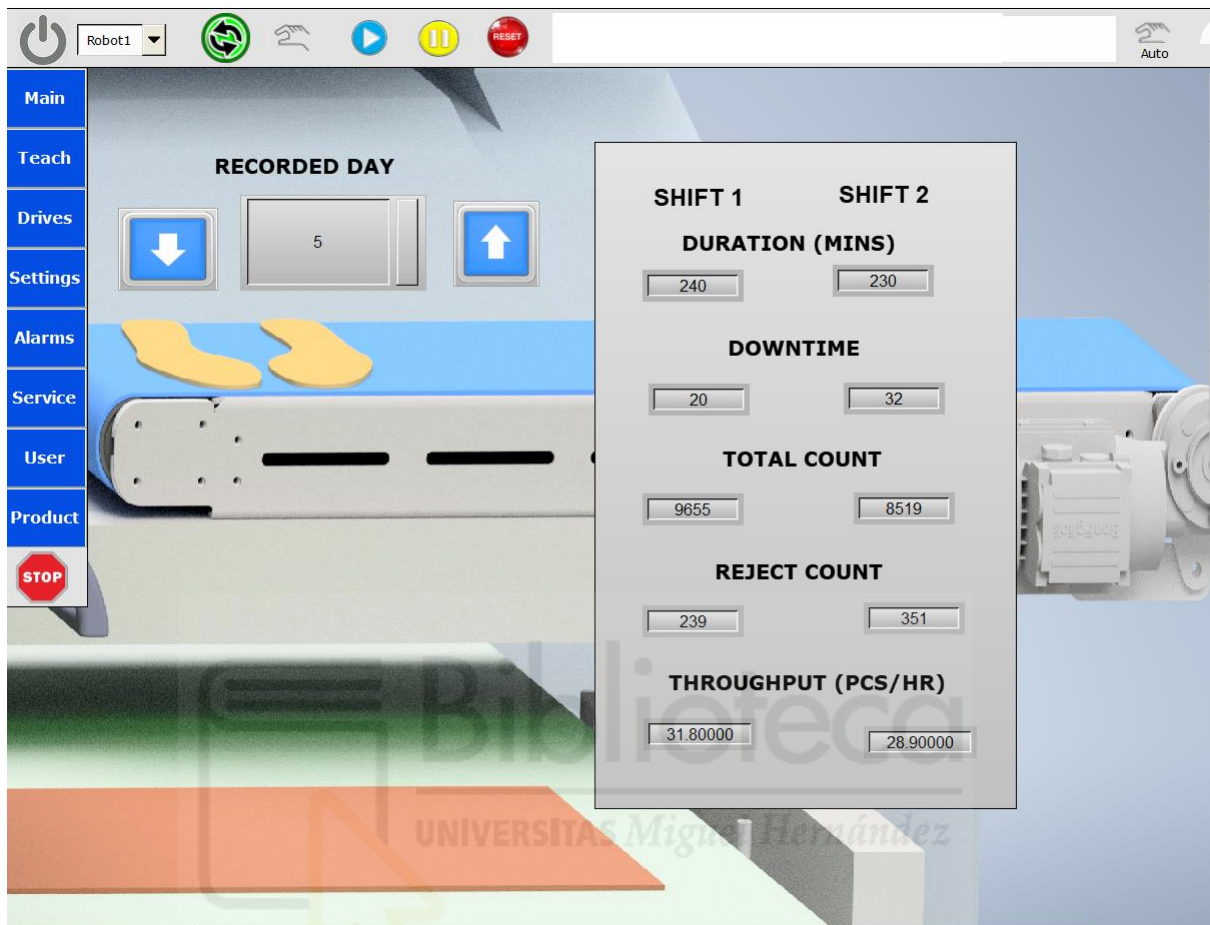
Al pulsar el turno se puede visualizar el estado actual de este (activo o parado) en la Figura 24, permite fijar un objetivo personal en este turno y darlo por finalizado en caso de que se termine antes de la hora estimada.

Figura 24. Ventana de diálogo para la personalización del objetivo de turno.



Se puede acceder a la función de historial al pulsar el botón denominado saved que se observa en la Figura 23 se accede a la ventana de la Figura 25 donde se puede navegar por el historial de producciones del mes, permitiendo visualizar todos los datos necesarios para detectar anomalías.

Figura 25. Interfaz de usuario para la consulta del historial de registros de producción.



5.2 PANTALLA DE MANTENIMIENTO

Durante la confección de la pantalla que se va a ocupar de las notificaciones del mantenimiento mostrada en la Figura 33 me he enfocado en la intuitividad y la usabilidad de las alarmas. La finalidad es tener el pleno control sobre las notificaciones de mantenimiento del sistema, pudiendo ayudar a reducir considerablemente los posibles problemas futuros.

Para ello se ha dispuesto de diferentes alarmas con límites y avisos fácilmente configurables, de esta manera se permite una clara y rápida visualización del tiempo restante para el próximo mantenimiento de cada parte del sistema

5.2.1 ARQUITECTURA DE PROGRAMACIÓN

La arquitectura dispuesta para las tareas de mantenimiento como se aprecia en la Figura 28 se basa en una estructura de temporizadores, los cuales he configurado en un registro de almacenamiento persistente para asegurar el correcto funcionamiento frente a interrupciones de suministro eléctrico.

5.2.2 DESARROLLO DE LA PROGRAMACIÓN

MANTENIMIENTO

La programación que se ha realizado como se puede observar en las Figuras 26 y 27 se basa en una serie de temporizadores configurables por el usuario que supervisan distintas acciones del funcionamiento, se han dispuesto cinco distintos como ejemplo. La gestión de los temporizadores consta de un control de reinicio general que se evaluará como haber realizado todos los mantenimientos necesarios y de 5 botones individuales para reiniciar cada una de las tareas por separado.

Este sistema se coordina con el gestor de estados de mantenimiento, el cual evalúa constantemente si una de las acciones de mantenimiento está cerca de su vencimiento o si ya ha caducado su tiempo de validez.

Respecto a la gestión de alertas, se ha aplicado un criterio de jerarquía para mantener la seguridad, por el cual, si el sistema de control detecta cualquier alarma técnica o de seguridad, se priorizará en la pantalla sobre avisos de mantenimiento, evitando que estos, los cuales se encienden más a menudo, opaquen el sistema de seguridad.

En la Figura 26 se aprecia que, al conseguir un guardado remanente, este se actualizará cada segundo, evitando que un corte de corriente lo reinicie.

Figura 26. Programación de temporizadores del mantenimiento.

```
// Bucle que cuenta cada segundo
Tempseg(IN:= (Control_Produccion.Turno_actual.Activo AND NOT (Tempseg.Q)) ,PT:= T#1S);

// cada segundo acumula el tiempo de funcionamiento de la máquina
IF Tempseg.Q THEN
    Mantenimiento.ValBat:=Mantenimiento.ValBat+T#1S;
END_IF
// Resetea el tiempo en caso de realizar el mantenimiento
IF Mantenimiento.BotBat OR Mantenimiento.BotMant THEN
    Mantenimiento.ValBat := T#0S;
END_IF;
```

GESTOR DE ESTADOS DE MANTENIMIENTO

La programación del gestor de estados de mantenimiento se encuentra en la Figura 27, este envía el estado de cada temporizador por separado y como resultado le asigna un valor del 0 al 2. Posteriormente en caso de que alguno de los temporizadores pase del punto crítico, además de en la pantalla destinada al mantenimiento, muestra por pantalla una alarma informando al usuario, siempre y cuando no haya otra más importante.

Figura 27. Programación del gestor de estado del mantenimiento.

```
//Transforma las entradas de la interfaz a tiempo
LimBat      := DINT_TO_TIME(Mantenimiento.IntLimBat*36000000);

AvisoBat    := DINT_TO_TIME(Mantenimiento.IntAvisoBat *36000000);

//Evalua el estado de las alarmas de mantenimiento
// Estado 0 todo okay
//Estado 1 falta poco para el mantenimiento
// Estado 2, momento del mantenimiento

IF Mantenimiento.ValBat > LimBat THEN
    EstadoBat := 2;
ELSIF Mantenimiento.ValBat > (LimBat - AvisoBat) THEN
    EstadoBat := 1;
ELSE
    EstadoBat:=0;
END_IF;

// siempre y cuando no haya un alarma más importante encendida, avisa cuando algún componente necesita mantenimiento
IF ((EstadoBat + EstadoCinta + EstadoRev + EstadoAce + EstadoTrans > 0) AND indice =0) THEN

    indice := 5;
END_IF;
```

Para complementar la explicación del sistema se ha añadido la Figura 28 con el diagrama de flujo de uno de sus temporizadores

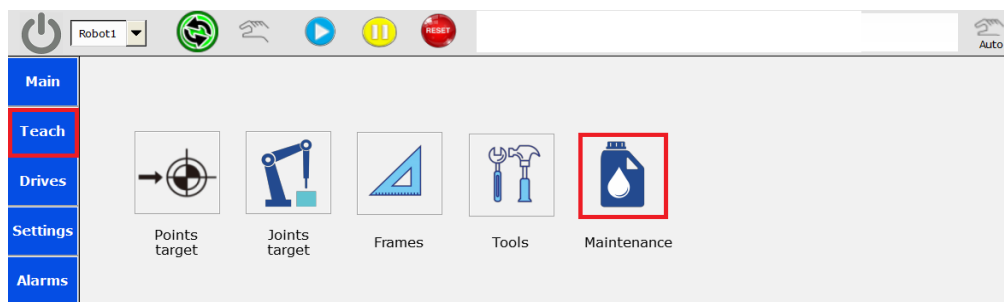
Figura 28. Diagrama de flujo de la lógica de control para el módulo de mantenimiento preventivo.



5.2.3 DISEÑO DE LA INTERFAZ

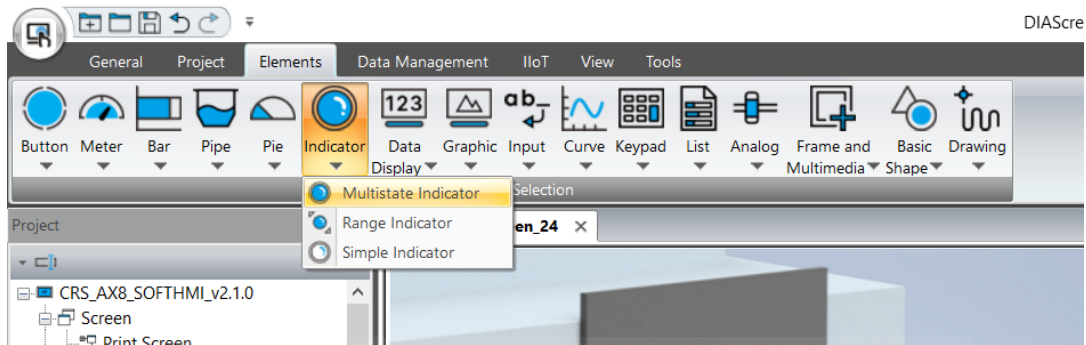
El mantenimiento se ha añadido dentro de la ventana denominada 'Teach', debido a que esta sección centraliza diversos accesos directos y funciones auxiliares del sistema. Como se puede apreciar en la Figura 29, esta ubicación permite una navegación rápida hacia otras herramientas de diagnóstico y configuración, optimizando el flujo de trabajo del personal técnico.

Figura 29. Ubicación del módulo de mantenimiento.



En el desarrollo de la pantalla de alarmas de mantenimiento, destaca un componente no detallado anteriormente, el Multistate Indicator. Este objeto se localiza en el apartado de indicadores, tal como se ilustra en la Figura 30. Se trata de un elemento dinámico diseñado para representar visualmente el estado de una variable específica mediante una lógica de estados múltiples.

Figura 30. Situación de los elementos Indicadores.



El Multistate Indicator permite configurar distintos estados en función del valor del tipo de dato de la variable vinculada. A cada estado se le pueden asignar diferentes atributos estéticos, iconos y textos diferenciados para notificar con precisión los cambios o incidencias en el proceso. En las Figuras 31 y 32 se observan las modificaciones realizadas para señalar los valores de alarma.

Figura 31. Ajustes del objeto Multistate Indicator, variable en estado 1.

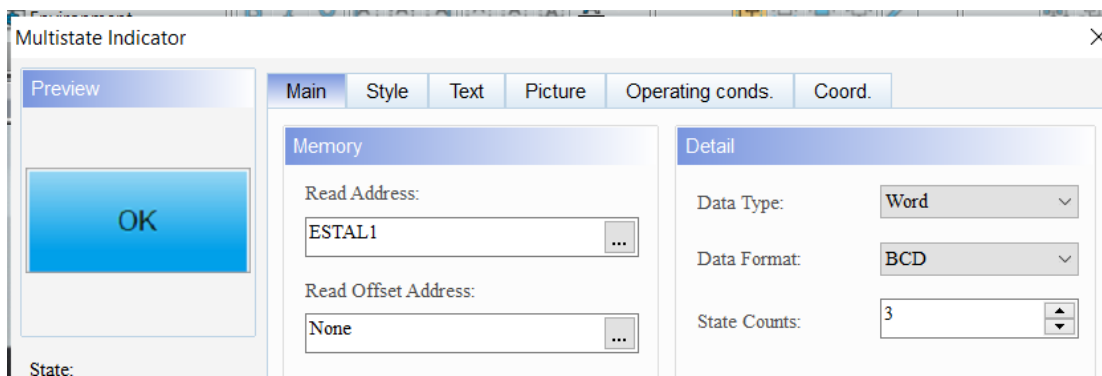
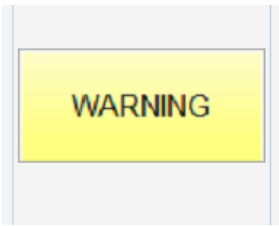
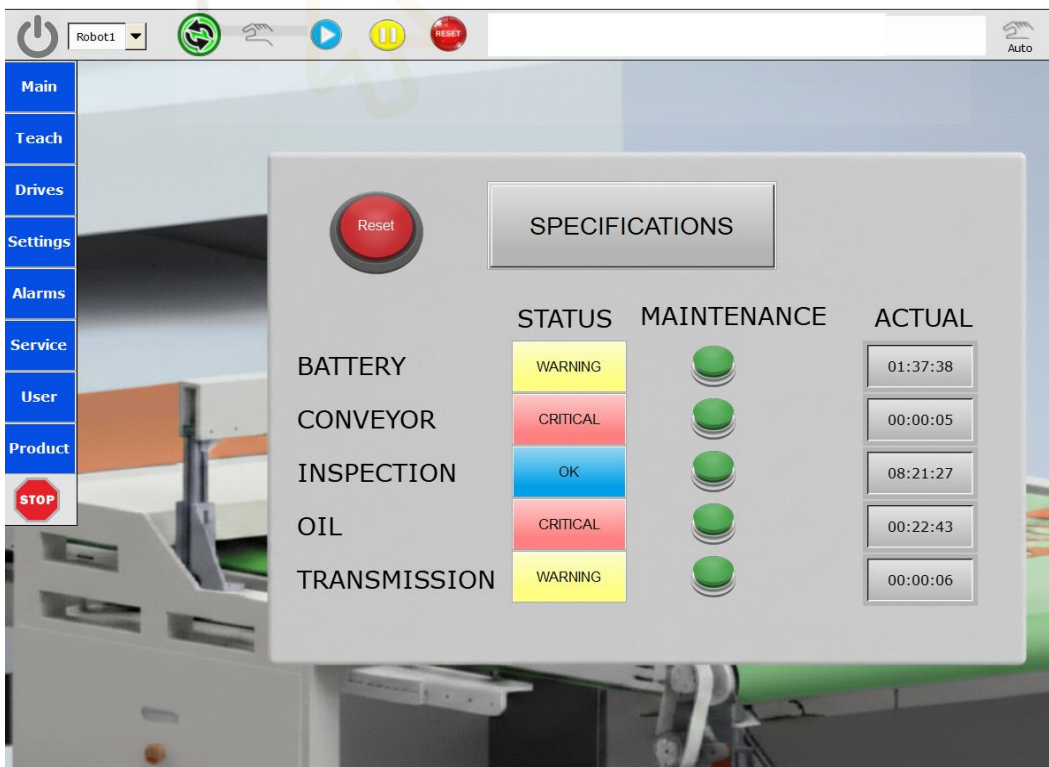


Figura 32. Ajustes del objeto Multistate Indicator, variable en estado 2.



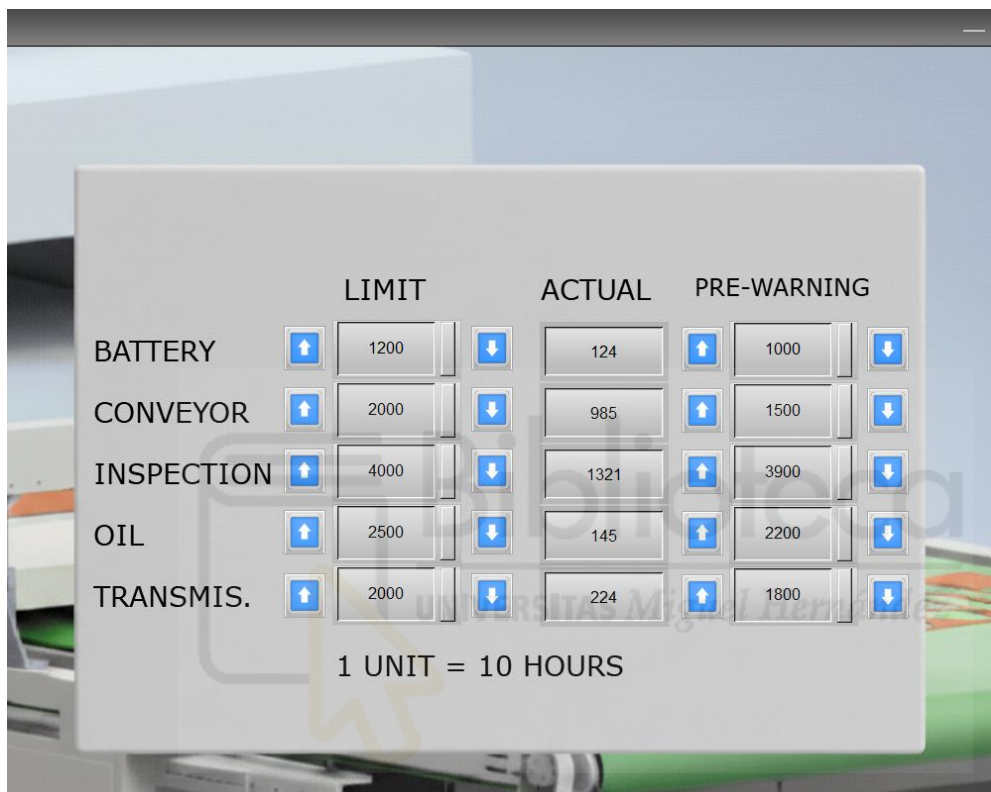
Al diseñar esta interfaz de la Figura 33, el objetivo principal es que cualquiera pueda ver el estado de las alarmas de un solo vistazo, sin distracciones. No es solo una cuestión de estética; dado que en la Figura 34 se puede observar que se han integrado paneles para ajustar umbrales y avisos preventivos que facilitan la gestión del mantenimiento diario. Respecto a la seguridad, el acceso a estos ajustes y al reseteo de los contadores está limitado a ciertos usuarios utilizando la gestión de usuarios que pone a nuestro alcance la aplicación de diseño. Solo los perfiles con alta prioridad pueden acceder, lo que evita que un usuario sin la formación adecuada marque como realizada una tarea de mantenimiento que, por normativa o seguridad, debe ejecutar un profesional.

Figura 33. Panel principal de navegación para la gestión del mantenimiento preventivo.



En la ventana emergente de la Figura 34 se puede editar los límites y avisos de cada acción de mantenimiento.

Figura 34. Módulo de gestión y edición de intervalos de mantenimiento preventivo



5.3 INTEGRACIÓN FUNCIONAL DE LA PROGRAMACIÓN

Partiendo de una conexión y configuración de variables ya establecida, el volcado de la programación nueva se basa en la creación de las nuevas funciones descritas anteriormente y en su correcta inicialización durante el funcionamiento del sistema. Una vez creadas e inicializadas se añaden las variables que requieran conectarse con la interfaz a la configuración de símbolos y se actualiza el archivo desde la aplicación DiaScreen. Gracias a que el protocolo utilizado para la conexión ha sido CODESYS, el direccionamiento de cada variable ha sido simbólico utilizando etiquetas (tags) del propio programa directamente en la interfaz, agilizando el mapeo y reduciendo los

errores al realizar cambios en el programa. Durante esta etapa se ha realizado el direccionamiento individual de cada actuador, registro de entrada e indicador de estado de la interfaz para que corresponda a su variable. Además de en caso específico de los ajustes de mantenimiento, se han limitado cada elemento de mando y campo de introducción de entrada a determinados usuarios destinados a ello.

Un aspecto crítico que se ha utilizado para esta implementación es la estructura construida para la organización, el envío y el almacenamiento de datos. Se ha diseñado una estructura llamada "Turno", la cual reúne la información correspondiente a la producción de cada turno; minutos de duración, tiempos de inactividad, unidades producidas y rechazadas y un registro booleano para identificar cuando está en transcurso el turno. Estos datos se organizan en un array de dos distancias, conformando la unidad de registro diario, utilizada para los cálculos de rendimiento y sobre todo en historial guardado.

La característica principal que la estructura anterior ha otorgado al programa es la parametrización del sistema, en cambios avanzados una estructura es fácilmente modificable, por ejemplo, para añadir un turno o cambiar las horas de todos bastaría con pequeños ajustes claros en el código. En complementación, a nivel usuario la interfaz mantiene su adaptabilidad permitiendo cambiar objetivos diarios y los parámetros del mantenimiento ajustándose en función de las necesidades de cada momento sin depender de editar la programación.

En la interfaz DIaScreen se ha priorizado la respuesta dinámica, vinculando los cambios cromáticos de los objetos con variables a tiempo real, reduciendo incertidumbre operativa. Con la finalidad de impedir errores producidos por las pantallas se han añadido bloques funcionales R_TRIG para la detección de flancos de subida en los actuadores que puedan causar error por efectos del asincronismo entre el ciclo de funcionamiento del PLC y la tasa de refresco de su interfaz instalada. Esto impide acciones redundantes como la finalización de

varios turnos sin querer o que se marque más de una pieza rechazada sin ser lo deseado.

5.4 INCORPORACIÓN DEL HMI

Con la finalidad de optimizar el sistema, el siguiente paso por el que se ha optado es la instalación de una pantalla HMI que sustituya al PC que está actualmente instalado como sistema de control. Las mejoras que ofrece este cambio son numerosas, tanto económicas como respecto a la seguridad y velocidad de producción. Desde el punto de vista financiero, aunque la pantalla suponga un coste inicial, a largo plazo reducirá los gastos por diferentes razones como las actualizaciones de hardware, además de no necesitar licencias de ningún tipo y por supuesto reduciendo día a día el consumo eléctrico.

Se puede asegurar que aumentará considerablemente la seguridad frente a amenazas externas dado que un ordenador es más vulnerable a este tipo de ataques y sobre todo evitará que el personal no autorizado realice ninguna manipulación no permitida, cumpliendo así la normativa IEC 62443 en [12].

Y por último una pantalla industrial favorece la producción dado que asegura una disponibilidad instantánea y no requiere de tanto tiempo de inactividad para actualizaciones de software tan a menudo como un ordenador.

Para la elección del modelo específico que se va a integrar en el proyecto se consideran diversos factores, algunos más determinantes que otros; lo más importante es asegurarse de que la conectividad del dispositivo sea compatible al PLC, puesto que este ya está seleccionado y se encuentra previamente operativo. El siguiente criterio para evaluar es la resistencia ambiental, como se ha comentado varias veces en el entorno laboral de una industria se requiere de una tolerancia más alta que en otros posibles entornos. Una vez esos dos criterios se cumplan se seleccionará la pantalla considerando el factor económico, las especificaciones visuales y la portabilidad del software que disponga cada opción.

Debido a que en este proyecto ya se está usando un PLC de la marca Delta y el diseño de la interfaz se ha desarrollado tomando como base una pantalla de esta marca, lo más eficiente es elegir una pantalla de esta misma firma, facilitando la sincronización entre ambos dispositivos y la migración de toda la interfaz. Analizando la arquitectura del PLC Delta AX-8xxEP0 Windows Series, el cual trabaja mayoritariamente por Ethernet para mantener el máximo rendimiento en tiempo real, se ha optado por dicha conectividad para la pantalla

Tras este análisis general de características, se ha seleccionado la serie DOP-100 del fabricante Delta, la cual consta de tres categorías según los requisitos del proyecto: Basic, Standard o Advanced. La Gama Basic, tratándose de la opción más económica consta de un procesador apto para el uso habitual de una pantalla en este ámbito, sin embargo su tamaño es reducido, llegando hasta 7 pulgadas y su carcasa de plástico solo alcanza un grado de protección IP55, pero únicamente uno de sus modelos consta de puerto Ethernet necesario para la conexión con el PLC planeada. Respecto a la Gama Standard, la necesidad de un puerto Ethernet queda cubierta con todos sus modelos, sus tamaños se extienden desde 7 pulgadas hasta 15 y tiene un mejor grado de protección, un IP65 incorporando la opción VNC o control remoto con uso limitado. Y por último la gama Advanced, esta opción tiene unos tamaños similares a los de la anterior pero los modelos respectivos a 12 y 15 pulgadas traen incorporada la versión multimedia haciéndolas aptas para la reproducción de videos, además todas las pantallas pertenecientes traen incorporado un segundo puerto Ethernet y la función Web server o HTTP.

Entre estas opciones se ha descartado la Gama Basic debido a su carencia de puerto Ethernet además de su baja protección y el reducido tamaño que ofrece dado que las ampliaciones que se han realizado requieren de una buena resolución. La siguiente Gama cumple las necesidades mínimas requeridas para el uso de la pantalla actual, incluso con las ampliaciones mencionadas, sin embargo, las pantallas Standard no tienen ninguna de las necesidades claves para la mejora propuesta en la cual se implementa una cámara a tiempo real.

Esta funcionalidad agradece un puerto Ethernet extra para una instalación más práctica y eficiente, además de la función multimedia que permite la fluidez de vídeo necesaria para esto. Por último y no menos importante, la gama superior incluye una mejora en la herramienta VNC o Control Remoto, la cual permite el control de la pantalla desde dispositivos móviles, contando con el doble de memoria Ram que la anterior permite que su uso sea mucho más competente que con la pantalla Standard, añadiendo la opción de utilizar el sistema Web server para incluso centrar la interfaz en una página web y que el resto de los dispositivos se conecten a ella.

Como conclusión; asumiendo que en el uso diario se va a incluir la comprobación del historial de datos recientemente implementado y una cámara que permita al usuario comprobar el estado del proceso desde la pantalla, la opción que se recomienda elegir es la gama Advanced para garantizar la compatibilidad de dichas funcionalidades, en caso contrario descartando la adición de la cámara y el uso de las funciones HTTP y VNC como algo habitual se recomienda elegir una pantalla de la gama Standar.

La elección de la DOP-112 WX mostrada en la Figura 35 cumple íntegramente con las especificaciones requeridas ya descritas anteriormente y con los cambios previstos, para especificaciones más concretas consultar en el Anexo III, el Datasheet. Este modelo proporciona el tamaño perfecto para la manipulación de un operador, evitando el encarecimiento innecesario del hardware o la incomodidad de su uso diario.

Figura 35. Terminal de interfaz operador (HMI) modelo DOP-112WX.



5.3.1 CARACTERÍSTICAS TÉCNICAS

INTERFAZ GRÁFICA

Pantalla de 12 pulgadas con una resolución de 1024 x 768 píxeles y una tecnología True color, garantizando una superficie de trabajo adecuada y una visualización fluida de vídeo.

CAPACIDAD DE PROCESAMIENTO

Procesador Dual Core de 1 GHz con alto rendimiento y Memoria RAM de 1 GB.

CONECTIVIDAD E INTERFAZ DE DATOS

Consta de dos puertos Ethernet integrados; por lo tanto, la implementación de la cámara se podrá hacer mediante uno de ellos y el otro usarse para la conexión con el PLC.

RESISTENCIA AMBIENTAL

Esta pantalla cuenta con la certificación IP65 / NEMA 4, asegurando en el panel frontal estanqueidad frente a polvo y proyecciones de agua. Soporta un rango térmico de hasta 50°C.

SOFTWARE Y SEGURIDAD

Presenta plena compatibilidad con DiaScreen y adicionalmente soporta el protocolo VNC, el cual habilita la opción futura de monitorización y control desde dispositivos móviles o tablets. Satisface las necesidades de seguridad multinivel diseñadas anteriormente en la planificación para el control del sistema.

5.3.2 MONTAJE E INSTALACIÓN DEL TERMINAL

Para optimizar la operatividad, se ha elegido una pantalla industrial de alta resolución y se ha planeado una instalación a medida. Este diseño físico permite al operador gestionar el sistema con más agilidad y sustituyendo un ordenador.

Con el objetivo de asegurar una respuesta inmediata entre el control y la interfaz, se van a definir las necesidades que el sistema debe cumplir. Se mencionará el procedimiento de conexión para la pantalla, en el Anexo IV se encuentra el manual de instrucciones que se recomienda consultar para información más concreta.

Para seguir los estándares de seguridad Industrial indicados en el manual el equipo está diseñado para su utilización entre 0° C y 50°C, siendo imperativo un espacio mínimo de un centímetro en la carcasa trasera para su correcta ventilación, evitando instalar cerca ninguna fuente de calor intenso o compartimentos estancos. Para la alimentación del dispositivo se requiere de una fuente de 24V DC aislada, verificada previamente para asegurar su estabilidad. Como acción de protección se debe conectar un terminal a la tierra en el HMI que proteja el hardware y mitigue el ruido electromagnético.

Para el conexionado se utiliza un cable Ethernet CAT6 apantallado, conectando el puerto RJ45 del HMI con el switch correspondiente al sistema, el uso del CAT6 asegura que el flujo de datos sea inmune a interferencias producidas por los motores o variadores cercanos. Para descargar en la pantalla el diseño ya creado y configurado se puede usar tanto la propia conexión Ethernet como por Vía USB.

Cuando se haya finalizado la descarga, la interfaz se reiniciará automáticamente iniciando los drivers de comunicación y estableciendo los enlaces correspondientes con el PLC y posteriormente podrá usarse.

6. BIBLIOGRAFÍA

[1] Barrientos, A., Barrientos Cruz, A., Peñín, L. F., & Balaguer, C. (2007). Fundamentos de robótica. McGraw-Hill Interamericana de España S.L.

[2] Groover, M. P. (2014). Automation, Production Systems, and Computer-integrated Manufacturing. PHI Learning.

[3] International Organization for Standardization. (2021). ISO 8373:2021 Robotics.

[4] Hodges, B., & Hallam, P. (1990). Industrial Robotics. Elsevier Science & Technology Books.

[5] Nof, S. Y. (Ed.). (1999). Handbook of Industrial Robotics. Wiley.

[6] International Electrotechnical Commission. (2013). IEC 61131-3:2013 Programmable controllers - Part 3.

[7] International Electrotechnical Commission. (1989). IEC 60529:1989 Degrees of protection provided by enclosures.

<https://es.scribd.com/document/463937393/PROGRAMACION-DEL-AUTOMATA-TWIDO-UTILIZANDO-LENGUAJE-DE-ALTO-NIVEL-GRAFICET>

[8] Delta Electronics, Inc. (2021). AX-8 Series Motion Controller User Manual: PLC-based Motion Control Architecture. DELTA.

https://downloadcenter.deltaww.com/en-US/DownloadCenter?v=1&CID=06&itemID=060210&downloadID=AX-8%20Series&sort_expr=cdate&sort_dir=DESC

[9] PLCopen Technical Committee. (2002). Function Blocks for Motion Control. https://www.plcopen.org/application/files/2117/4368/8699/mitsubishi_electric_ig-f-series_fx5-ssc-g_part4_0.pdf

[10] International Society of Automation (ISA). (2015). ANSI/ISA-101.01-2015, Interfaces hombre-máquina para sistemas de automatización de procesos. https://www.isa.org/products/ansi-isa-101-01-2015-human-machine-interfaces-for?_gl=1*vzl7lc*_up*MQ..*_ga*NjAyMzkxNzI2LjE3NzAyMDUwNTM.*_ga_8Z8VGE0R98*_czE3NzAyMDUwNDckbzEkZzEkdDE3NzAyMDUwNjEkajQ2JGwwJGg2NzA0MDE4NzM.

[11] Verein Deutscher Ingenieure. (2004). Design methodology for mechatronic systems. <https://es.scribd.com/document/689770774/Design-Methodology-Mechatronic-Systems-unlocked-33-56-Es>

[12] International Electrotechnical Commission. (2018). IEC 62443-4-2:2018 Security for industrial automation and control systems - Technical security requirements for IACS components.