



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

CAMPUS D'ALCOI

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Politécnica Superior de Alcoy

Automatización y monitorización de un proceso industrial
virtualizado con Factory I/O

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Muelas Aspano, Marcos

Tutor/a: Hilario Caballero, Adolfo

CURSO ACADÉMICO: 2023/2024

Resumen

En este trabajo de fin de grado se ha diseñado y desarrollado la automatización de un proceso industrial virtualizado. También se ha diseñado y programado la visualización de los datos de la planta industrial. De tal forma este proceso será virtualizado en Factory I/O el cual nos permitirá simular nuestra planta industrial. El controlador empleado será PLC Siemens S7-1200 y la programación se realizará en el entorno de TIA Portal. Donde se desarrollarán los distintos modos de funcionamiento. Además de automatizar el proceso, en este proyecto se desarrollará una interfaz de usuario en una pantalla táctil industrial (HMI). De tal forma en la que se pueda controlar el proceso de forma sencilla y eficaz. Se gestionarán los permisos de los usuarios los cuales tienen acceso a la pantalla y las alarmas resultantes durante el proceso.

Finalmente se hará una recogida de los datos en planta los cuales se almacenarán en una base de datos. Además de ser mostrados mediante distintos tipos de gráficas en Grafana.

Palabras clave: Automatización; PLC, TIA Portal; Grafana; Factory I/O.

Resum

En este treball de fi de grau s'ha dissenyat i desenvolupat l'automatització d'un procés industrial virtualitzat. També s'ha dissenyat i programat la visualització de les dades de la planta industrial. De tal forma este procés serà virtualitzat en Factory I/O el qual ens permetrà simular la nostra planta industrial. El controlador emprat serà PLC Siemens S7-1200 i la programació es realitzarà a l'entorn de TIA Portal. On es desenvoluparan els diferents modes de funcionament. A més d'automatitzar el procés, en este projecte es desenvoluparà una interfície d'usuari en una pantalla tàctil industrial (HMI). De tal forma en la qual es puga controlar el procés de manera senzilla i eficaç. Es gestionaran els permisos dels usuaris els quals tenen accés a la pantalla i les alarmes resultants durant el procés.

Finalment es farà una recollida de les dades en planta els quals s'emmagatzemaran en una base de dades. A més de ser mostrats mitjançant diferents tipus de gràfiques en Grafana.

Paraules clau: *Automatització; PLC, TIA Portal; Grafana; Factory I/O;*

Abstract

In this final degree project, the automation of a virtualized industrial process has been designed and developed. It has also been designed and programmed the data visualization of the industrial plant. In this way this process will be virtualized in Factory I/O which will allow us to simulate our industrial plant. The controller used will be PLC Siemens S7-1200 and the programming will be done in the TIA Portal environment. Where the different operating modes will be developed. In addition to automate the process, this project will develop a user interface on an industrial touch screen (HMI). In such a way that the process can be controlled in a simple and efficient way. The permissions of the users who have access to the screen and the resulting alarms during the process will be managed.

Finally, data will be collected in the plant and stored in a database. In addition to being displayed through different types of graphs in Grafana.

Keywords: *Automation; PLC; TIA Portal; Grafana; Factory I/O;*

Índice general

Resumen	I
Índice general	III
Índice de figuras	V
Índice de tablas	VIII
1. Introducción	1
1.1. Motivación	2
1.2. Ventajas e inconvenientes de la automatización	2
2. Objetivos y alcance	4
3. Descripción del proceso que se va a automatizar	6
3.1. Primer prototipo	6
3.2. Prototipo final	8
4. Especificaciones de funcionamiento	11
4.1. Estructura de la línea de producción	11
4.2. Descripción de los modos de funcionamiento de la línea de producción	15
4.3. Diseño HMI	19
4.4. Datos a monitorizar con Grafana	20
5. Descripción de la solución	22
5.1. Sensores y actuadores incluidos en los procesos	22
5.2. Alarmas de emergencia	38
5.3. Dispositivos de control y comunicaciones utilizados	39
5.4. Arquitectura de comunicaciones	42
5.5. Otras aplicaciones utilizadas	44
6. Descripción del desarrollo	49
6.1. Desarrollo del programa que automatiza el proceso	49
6.2. Desarrollo de la aplicación para el HMI	73
6.3. Configuración de KEPServerEX	78
6.4. Configuración de Factory IO	79

6.5. Configuración de Node-RED	80
6.6. Configuración de InfluxDB	85
6.7. Configuración de Grafana	87
7. Resultados obtenidos	93
8. Conclusiones y mejoras	94
Bibliografía	95

Índice de figuras

3.1. Mock up del primer prototipo de la línea de producción	7
3.2. Primer prototipo dentro de una escena de Factory I/O	8
3.3. "Mock up" del diseño final de la línea de producción	10
3.4. Diseño final dentro de una escena de Factory I/O	10
4.1. Figura del panel de mandos	14
4.2. GRAFCET de los modos de funcionamiento. Fuente: elaboración propia	16
5.1. Sensor de difusión	23
5.2. Matriz de luces	23
5.3. Transportador de rodillos	24
5.4. Cinta transportadora	24
5.5. Plataforma giratoria o turn table	25
5.6. Pick & place	25
5.7. Centro de mecanizado	26
5.8. Emisor de objetos	26
5.9. Eliminador de objetos	27
5.10. Producto de materia prima	27
5.11. Base de producto	28
5.12. Caja pequeña	28
5.13. Palet cuadrado	29
5.14. Caja apilable	29
5.15. Partes de la línea de producción	30
5.16. Cinta de alimentación, parte inicial de la línea de producción	31
5.17. Turntable o plataforma giratoria	32
5.18. Máquina de inicio 1	33
5.19. Recogida de palets 1	33
5.20. Máquina de procesado	34
5.21. Máquina de entrega de bases	35
5.22. Máquina de fin de proceso 1	36
5.23. Máquina de inicio 2	36
5.24. Recogida de palets 2	37
5.25. Máquina de fin de proceso 2	38
5.26. PLC Siemens S7-1200, CPU 1214C AC/DC/Rly	40

5.27. SIMATIC HMI, KTP700 Basic PN	42
5.28. Flujo de la arquitectura OPC UA. Fuente: elaboración propia	43
5.29. Flujo de la arquitectura para el desarrollo del proyecto. Fuente: elaboración propia, con logos de cada fabricante	44
5.30. Logo de TIA Portal V17	45
5.31. Logo de Factory I/O	46
5.32. Logo de KEPServerEX	46
5.33. Logo de Node-RED	47
5.34. Logo de InfluxDB	47
5.35. Logo de Grafana	48
6.1. Ejemplo de la organización de las variables booleanas de salida. - PLC Tags TIA Portal	50
6.2. Bloques de función de selección del modo de funcionamiento del panel y HMI	52
6.3. Tipo de datos creado para la selección de los modos	53
6.4. Red para la resolución de conflictos entre panel y HMI	54
6.5. Red para la resolución de conflictos entre panel y HMI	55
6.6. GRAFCET de los modos de funcionamiento/trabajo	56
6.7. Red modos automático paro y manual, parte del modo automático paro	58
6.8. Red modos automático paro y manual, parte restante del modo automático paro	58
6.9. Red modos automático paro y manual, parte del modo manual	59
6.10. Red modos automático paro y manual, parte restante del modo manual	59
6.11. GRAFCET de la Cinta de alimentación	62
6.12. GRAFCET de Máquina de inicio 1, Máquina de inicio 2 y Máquina de entrega de bases	62
6.13. GRAFCET del Turntable o plataforma giratoria	63
6.14. GRAFCET de Recogida de palets 1 y Recogida de palets 2	64
6.15. GRAFCET de Máquina de procesado	64
6.16. GRAFCET de Máquina de fin de proceso 1 y Máquina de fin de proceso 2	64
6.17. FBs Cinta de alimentación y Turntable o plataforma giratoria modo automático y conversor de datos de la matriz de luces	67
6.18. FB Máquina de inicio 1 modo automático	67
6.19. FB Máquina de inicio 2 modo automático	68
6.20. FB Máquina de procesado modo automático	68
6.21. FB Máquina de entrega de bases modo automático	69
6.22. FBs Recogida de palets 1 y Recogida de palets 2 modo automático	69
6.23. FBs Máquina de fin de proceso 1 y Máquina de fin de proceso 2 modo automático	70
6.24. FCs que gestionan las alarmas, parte 1	71
6.25. FCs que gestionan las alarmas, parte 2	71
6.26. FCs que gestionan las alarmas, parte 3	71
6.27. Tags del HMI asociados a los del PLC	73
6.28. Creación de usuarios y roles	73
6.29. HMI - Vista Home	74
6.30. HMI - User Management	75
6.31. Creación de las alarmas en el HMI	75
6.32. HMI - Alarmas	76
6.33. HMI - Modos	77
6.34. HMI - Modos - Turntable	77
6.35. Tags creados en KEPServerEX	78
6.36. OPC Quick Client de Kepserver	79

6.37. Conexión con el driver de KEPServerEX	79
6.38. Conexión con los Tags de KEPServerEX	80
6.39. Ejecución de Node-RED	81
6.40. Inject node - Configuración	82
6.41. Inject node - Configuración	83
6.42. InfluxDB out node - Configuración (Parte 1)	83
6.43. InfluxDB out node - Configuración (Parte)	83
6.44. Debug node - Configuración	84
6.45. Información que muestra el debug node	84
6.46. Flujos realizados - Parte 1	85
6.47. Flujos realizados - Parte 2	85
6.48. Creación de la BBDD con Influx por consola	86
6.49. Ver las medidas creadas en la BBDD desde la consola de Influx	86
6.50. Servidor de InfluxDB en ejecución	87
6.51. Conexión grafana con Influx	88
6.52. Dashboards creados	89
6.53. Panel de los contadores de los modos	90
6.54. Panel de los materiales	91
6.55. Panel de los temporizadores de las cintas de las máquinas	92
7.1. Vídeo Media UPV: "Control de una línea de producción virtualizada con monitorización de datos"	93
7.2. Vídeo Media UPV: "Funcionamiento del modo automático marcha en una línea de producción en Factory I/O"	93
7.3. Vídeo Media UPV: "Monitorización del tiempo de las cintas con Grafana"	93

Índice de tablas

6.1. Ecuaciones de estado activo de la Cinta de alimentación	65
6.2. Ecuaciones de set de la Cinta de alimentación	65
6.3. Ecuaciones de salida de la Cinta de alimentación	65

1 Introducción

La automatización en la industria ha aportado grandes beneficios a la hora de gestionar distintos tipos de procesos. Desde beneficios económicos, trabajar procesos con gran precisión, reducción de tiempos de producción, hasta de seguridad, eliminando y evitando riesgos laborales.

A pesar de tener tan buenos beneficios también existen sus inconvenientes ya que los costes iniciales al realizar la inversión para poder automatizar los procesos suelen ser altos, además de la propia obsolescencia de las máquinas. O la incorporación de nueva maquinaria o nueva tecnología. Ya que por lo general suele ser económicamente poco asequible y requiere de un proceso de adaptación a lo que ya estaba establecido anteriormente. Los beneficios e inconvenientes serán desarrollados en la [sección 1.2](#).

Para este proyecto se ha realizado una la virtualización mediante el simulador 3D Factory I/O de un proceso industrial desarrollado desde su inicio. El control del proceso ha sido llevado a cabo mediante un PLC haciendo uso de diversos modos de funcionamiento. También se ha desarrollado una interfaz de usuario sencilla en una pantalla táctil industrial (HMI). Donde se tendrá acceso a las pantallas según el permiso que tengan los usuarios. Desde las propias pantallas se podrán gestionar las alarmas de emergencia de la línea de producción se podrán dar de alta los usuarios, pero no de manera persistente. Y además se podrán controlar los modos de funcionamiento y actuadores desde el HMI.

Serán tomados datos de la línea de producción y almacenados en la base de datos de InfluxDB. Y una vez tomados los datos serán mostrados a tiempo real en la aplicación para la monitorización Grafana. Para llevar a cabo todo esto se ha hecho uso del servidor OPC UA KEPServerEX. Con el cual se comunicará el PLC, el HMI y Node-RED, que es el que se encargará de guardar los datos en InfluxDB.

Este capítulo sirve como capítulo introductorio exponiendo la motivación en la [sección 1.1](#) y las ventajas e inconvenientes en la [sección 1.2](#). Posterior a este capítulo se dará paso a la descripción del proceso que será virtualizado y automatizado en [Capítulo 3](#) donde se verá un primer prototipo desarrollado y el prototipo final que incluye el proceso a automatizar. Las especificaciones del funcionamiento que ha de llevar el proyecto será expuesto en diversas secciones en el [Capítulo 4](#). En el [Capítulo 5](#) se expondrá la solución

que se ha adoptado para llevar a cabo la automatización, gestión de alarmas, control de la línea de producción y monitorización. En el [Capítulo 6](#) se explica como se ha sido y como se ha desarrollado la solución que se ha llevado a cabo en el proyecto. Para el [Capítulo 7](#) han sido grabados distintos vídeos donde se muestra el funcionamiento al completo del proyecto. Y finalmente en el [Capítulo 8](#) se exponen a las conclusiones que se ha llegado con el proyecto y posibles mejoras que se podrían implementar en un futuro.

1.1 Motivación

Como bien sabemos la historia de la industria ha pasado por distintas etapas y revoluciones. Actualmente nos encontramos en la cuarta revolución industrial, a esta revolución también se le conoce como industria 4.0. Donde se gestionan digitalmente los elementos que intervienen en la cadena de producción de manera simultánea para obtener distintas cantidades masivas de datos en tiempo real y con esto optimizar los procesos y fases involucradas.

Durante estos años han surgido y evolucionado distintos tipos de tecnologías que se han hecho populares mundialmente, como por ejemplo:

- **Internet de las Cosas (IoT):** da la capacidad de poder conectar elementos electrónicos con internet. Como por ejemplo: asistentes personales, relojes digitales, PLCs, etc... En este tipo de tecnología se ha hecho muy popular el uso de placas de pequeño tamaño que son programables, las cuales tienen pines de entrada y salida. Las placas más relevantes en esta tecnologías son las de Arduino (microcontrolador), Raspberry PI (miniordenador)
- **Big Data:** realiza una interpretación y tratamiento de grandes cantidades de datos y de datos a tiempo real para mejorar los procesos de producción
- **Inteligencia artificial:** se encarga de dar solución a problemas de manera automática teniendo en cuenta datos de entrada

Estas tecnologías mencionadas se pueden trabajar en conjunto para así sacar un mayor provecho de ellas. La motivación del alumno para desarrollar este proyecto ha venido dada al cursar distintas asignaturas donde se han visto este tipo de tecnologías o similares.

1.2 Ventajas e inconvenientes de la automatización

A la hora de realizar cualquier tipo de implementación o cuando iniciamos un proyecto siempre es necesario hacer un balance para ver si nos supone rentable. Ya que si no lo realizamos estamos corriendo el riesgo de que cometer gastos innecesarios, perder recursos importantes y el tiempo invertido no habrá merecido nada la pena.

Por lo que tenemos que analizar las ventajas e inconvenientes que nos puede proporcionar nuestra implementación del modo automático. Las ventajas e inconvenientes han sido elaborados y desarrollados desde mi punto de vista.

1.2.1 *Ventajas automatización*

- **Reducción de los errores cometidos de los humanos:** Al no intervenir humanos de manera activa se disminuye en gran cantidad los errores que estos puedan cometer.
- **Eficiencia y velocidad en los procesos:** Las máquinas trabajan de manera constante sin apenas pausas o interrupciones, salvo en casos excepcionales como en los que ocurra algún fallo o sea activada una alarma. Y también pueden trabajar a velocidades elevadas y constantes.
- **Funcionamiento continuo:** Al estar automatizada la línea de producción esta podría funcionar de manera continuada e ininterrumpida durante un largo periodo de tiempo.
- **Abaratamiento de los costes:** A pesar que de manera inicial la automatización de una línea de producción pueda ser costosa, a la larga se ahorraría en muchos gastos de la mano de obra de manera significativa.
- **Garantizar mayor seguridad:** Al disminuir la exposición que sufren los trabajadores entornos donde tenían que usar manualmente las máquinas se asegura una mayor seguridad para estos.
- **Extracción de datos:** Cuando se automatiza un proceso es mucho más sencillo poder registrar datos detallados sobre la propia producción o el funcionamiento de la maquinaria.

1.2.2 *Inconvenientes automatización*

- **Coste inicial:** Las inversiones que se realizan en automatizar una línea de producción de manera inicial suelen ser caros, y esto es un gran inconveniente que puede tirar esta implementación para atrás.
- **Fallos de las propias máquinas:** Las máquinas en un momento dado podrían empezar a fallar, dado a su deterioro por el paso del tiempo o por defectos. Por lo que habría que estar atento a ello.
- **Pérdida de empleos:** Al implementar este modo muchos de los puestos de trabajo serían prescindibles. Por lo que muchos trabajadores se quedarían sin su puesto. Para solventar en gran medida esto habría que establecer cierta balanza en la que los trabajadores tengan cierta implicación/interacción con el automatismo.
- **Especialización:** Haría falta trabajadores que estén especializados que sean capaces de operar y realizar los mantenimientos adecuados de los sistemas automatizados.
- **Seguridad:** Es cierto que al haber menos interacción humana se reduce gran parte de los riesgos. Pero aún así está claro que todos no pueden ser evitados. Como por ejemplo: incendios, colisiones de las máquinas, ataques cibernéticos, accidentes con productos químicos, etc. . .

2 Objetivos y alcance

La finalidad de este proyecto es la automatización y monitorización de un proceso industrial que va a ser virtualizado.

Para llevar a cabo la finalidad del proyecto se han marcado los siguientes objetivos:

1. Idear el proceso que se va a tratar en la planta industrial.
2. Desarrollo de tres modos de funcionamiento para la planta industrial. Los cuales serán: modo manual, modo automático paro y modo automático marcha.
3. Virtualización de la planta donde se realiza el proceso industrial en Factory I/O.
4. Desarrollar una gestión de las alarmas generadas.
5. Desarrollar una interfaz de usuario simple y fácil de usar en el HMI para la gestión de la planta que además tenga una gestión de usuarios.
6. Almacenamiento de los datos recogidos en base de datos.
7. Monitorización de los datos almacenados de manera gráfica.

La programación ha sido desarrollada en TIA Portal para el PLC Siemens S7-1200 mayoritariamente en el lenguaje SCL y una muy breve parte de la programación ha sido desarrollada en lenguaje Ladder. La interfaz de usuario ha sido desarrollada en el HMI KTP700 Basic PN.

El servidor de comunicaciones empleado ha sido KEPServerEX que trabajará en conjunto con el PLC y Factory I/O para llevar a cabo la virtualización. Y también se trabajara con Node-RED para obtener los datos del servidor de comunicaciones y almacenarlos en la base de datos de InfluxDB.

La monitorización recogerá los datos almacenados en nuestra base de datos InfluxDB para que sean mostrados gráficamente.

Los únicos productos que se aceptarán en la virtualización para ser clasificados en la línea de producción solamente serán el [Producto de materia prima](#) y la [Caja pequeña](#). Y tendrán que ir obligatoriamente encima de un [Palet cuadrado](#).

Por parte del HMI los nuevos usuarios no se guardarán de manera persistente. Es decir, cuando este sea apagado el registro de los nuevos usuarios añadidos durante la sesión serán borrados. Esto se debe a las limitaciones que tiene el modelo del HMI empleado. Si se quiera dar de alta un nuevo usuario se tendrá que hacer desde el propio programa de TIA Portal.

Finalmente no se le aplicará ninguna capa de seguridad al proyecto, ya que su objetivo final es la monitorización de los datos extraídos. Y la implementación del proyecto no ha sido contemplada para que se realice en un entorno que sea distinto al local.

3 Descripción del proceso que se va a automatizar

El proceso a automatizar será virtualizado en el entorno de simulación Factory I/O. Para poder tener claro un proceso que se va a automatizar se han realizado dos prototipos distintos. El primer prototipo realizado ha servido como base de que elementos de Factory I/O iban a ser empleados y ver la complejidad que podía tener. Y el prototipo final el cual ya adopta un mayor conocimiento de los elementos que tiene Factory I/O y una vision más amplia de como diseñar un problema, en este caso el proceso industrial, y como poder abordarlo. En la [sección 3.2](#) encontraremos el proceso a automatizar que va a ser desarrollado.

3.1 Primer prototipo

Desde el primer momento estaba claro que tecnologías se iban a usar en el proyecto ([Capítulo 5](#)), pero no todas. Y también estaba claro su objetivo principal ([Capítulo 2](#)). Por lo que se desarrolló un primer un “mock up” (diseño inicial) de la vista en planta de la línea de producción para tener una primera idea de cuáles serían los posibles sensores y actuadores que se usarían para desarrollar la solución. Y además saber cómo serían distribuidos.

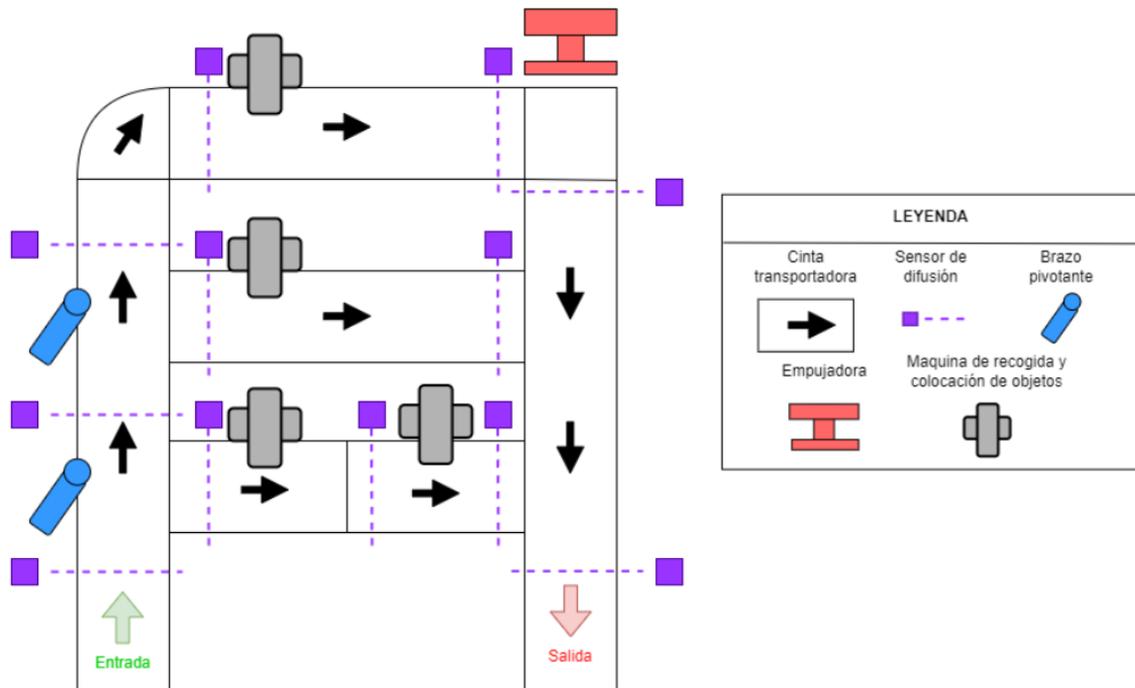


Figura 3.1: Mock up del primer prototipo de la línea de producción

La idea de este primer prototipo de manera muy resumida y rápida, ya que no es más que un prototipo. Trata de que cuando entra una pieza esta es detectada por los sensores de la cinta inicial. Y según la altura de la pieza, los brazos pivotantes o la cinta circular desplazarán la pieza a una de las cintas intermedias donde se procesará la pieza. Una vez procesada la pieza esta será desplazada a una cinta de salida donde será recogida.

Una vez hecho este primer “mock up” se pasó a Factory I/O con sus respectivos sensores y actuadores. Cabe destacar que la colocación de algunos sensores y actuadores no es la misma que la del “mock up”.

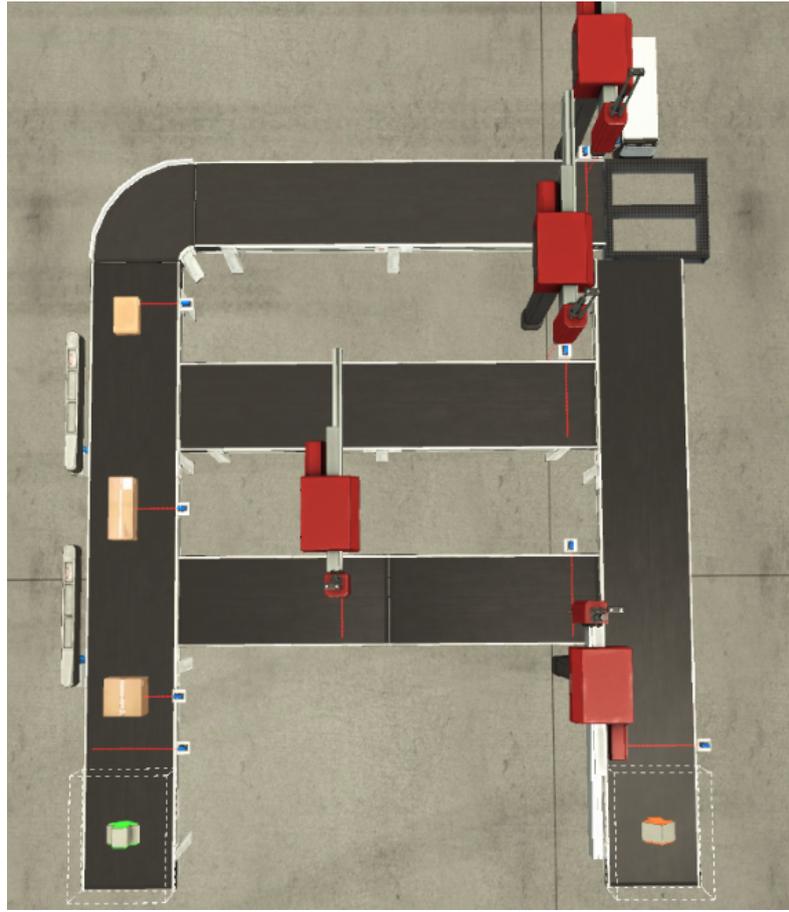


Figura 3.2: Primer prototipo dentro de una escena de Factory I/O

Finalmente realizando pruebas básicas solamente en Factory I/O usando los actuadores para probar el funcionamiento de como podría ser el automatismo.

Al terminar de crear el prototipo en Factory I/O se empezaron a desarrollar los diagramas GRAFCET y las ecuaciones de estado. Pero cuando estos se desarrollaron me di cuenta de que los procesos que se realizaban eran repetitivos. Ya que la variedad de los actuadores empleados era muy escasa. Tampoco es necesario hacer un automatismo realmente complejo. Pero tenía la idea de hacerlo un poco más variado para explorar todas las opciones que ofrecen los actuadores de Factory I/O.

3.2 Prototipo final

Una vez ya habiendo explorado mejor los actuadores que tiene Factory I/O para que fuese un poco más variado que en el primer prototipo. Se empezó a desarrollar el “mockup” del diseño final del proceso que tendría el automatismo para el proyecto. Para este prototipo final como es evidente también se ha realizado su respectivo "mock up", como se observa en la [figura 3.3](#). Y su respectiva implementación en una escena de Factory I/O para virtualizar la línea de producción como se muestra en la [figura 3.4](#)

Este diseño final consta de tres partes. La primera parte consta de la entrada de los productos a la línea de producción. Estos productos serán productos pequeños y productos grandes. Representados por [Producto de materia prima](#) y [Caja pequeña](#), respectivamente. En esta primera parte será detectada la altura de los productos gracias a la matriz de luces. Para posteriormente, en la segunda parte, ser desplazado por la plataforma rotatoria. Esta plataforma rotatoria desplazará los productos hacia la línea de la derecha si se tratan de los productos grandes. Y en caso contrario los productos serán desplazados hacia la línea de la izquierda.

El proceso de la línea de la derecha (también llamada línea de productos grandes) es simple. El producto llegará hasta la máquina de recogida y colocación de objetos. Esta máquina transportará el objeto hacia una línea final donde. Donde se colocará la caja pequeña encima de un palet para ser finalmente entregada. Y mientras se transporta el producto a la línea final la cinta se moverá para apartar el palet de la línea y no entorpecer la entrada de nuevos productos a esta.

Por otra parte el proceso en la línea izquierda (línea de productos pequeños) es similar inicialmente al de la línea de productos grandes. Los productos pequeños pasarán a una cinta la cual los desplazará hasta una estación de fabricación donde se procesarán. Una vez procesados saldrán hacia una cinta donde se recogerán los productos mediante una máquina de recogida y de colocación de objetos para posicionarlos en la cinta final y ser entregados de la misma manera que en la línea de objetos grandes. Cabe destacar que esta línea final esperará a que se hayan situado dos productos sobre la caja apilable del palet.

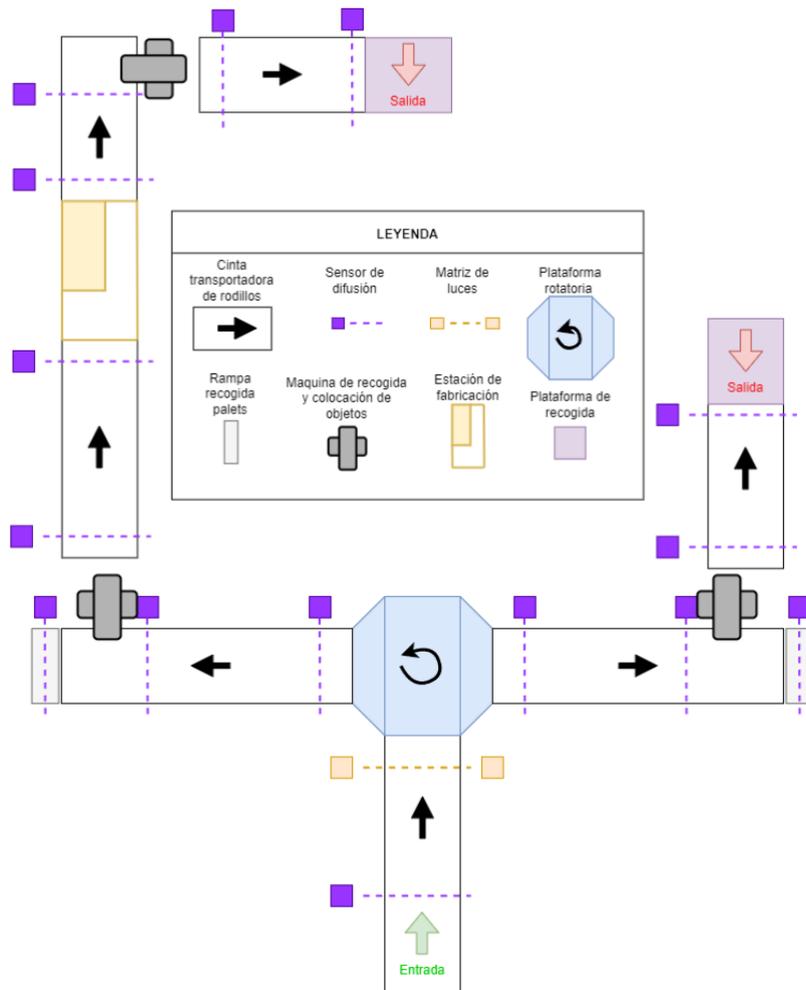


Figura 3.3: "Mock up" del diseño final de la línea de producción



Figura 3.4: Diseño final dentro de una escena de Factory I/O

4 Especificaciones de funcionamiento

En las especificaciones del funcionamiento se comentará sobre la propia estructura que tiene la línea de producción para poder llevar a cabo su proceso, los distintos modos de funcionamiento que se van a desarrollar.

4.1 Estructura de la línea de producción

El apartado de estructura tratará de exponer la estructura que tiene la línea de producción, los [Sensores y actuadores empleados en la línea de producción](#) y las [Piezas/productos empleados para cada parte de manera breve](#). Con la ayuda del "mock up"["Mock up" del diseño final de la línea de producción](#) podremos terminar de diseñar nuestra [Diseño final dentro de una escena de Factory I/O](#).

Hay un total de diez partes en las que la estructura ha sido dividida, como se puede ver en [Partes en las que se ha dividido la línea de producción](#). Algunas de estas partes están repetidas ya que las tareas que se realizan son similares y esto en parte simplifica el trabajo de programación de la línea de producción.

A continuación se mencionan los sensores, actuadores y objetos (que se colocan/eliminan en esa parte en concreto) implicados en las partes de la línea de producción de manera simplificada y a modo de recordatorio:

■ Cinta de alimentación

- Transportador de rodillos
- Emisor de objetos: colocará un palet cuadrado y sobre este un producto de materia prima o una caja pequeña
- Sensor de difusión al inicio
- Matriz de luces

- **Plataforma giratoria o Turntable**
 - Plataforma giratoria Turntable
- **2 Máquinas de inicio (una para materia prima y otra para cajas pequeñas)**
 - Transportador de rodillos
 - Sensor de difusión al inicio del transportador de rodillos
 - Pick & Place de dos ejes
 - Sensor de difusión junto al Pick & Place de dos ejes
- **2 Retiradas de palets**
 - Rampa
 - Sensor de difusión al inicio de la rampa
 - Eliminador de objetos
- **Máquina de procesado**
 - Cinta transportadora
 - Sensor de difusión al inicio de la cinta transportador
 - Sensor de difusión al final de la cinta transportadora
 - Centro de mecanizado: procesa la materia prima entrante y la convierte en el objeto base de producto
- **Máquina de entrega de bases (entrega productos pequeños)**
 - Cinta transportadora
 - Sensor de difusión al inicio de la cinta transportadora
 - Pick & Place de dos ejes
 - Sensor de difusión junto al Pick & Place de dos ejes
- **2 Máquinas de fin (una para bases de producto y otra para cajas pequeñas)**
 - Transportador de rodillos

- Emisor de objetos: colocará un palet cuadrado y una caja apilable encima del palet
- Sensor de difusión al inicio del transportador de rodillos junto al emisor de objetos
- Sensor de difusión al final del transportador de rodillos
- Eliminator de objetos al final del transportador de rodillos

Además de todas estas partes mencionadas anteriormente se ha colocado un panel para poder cambiar el modo de funcionamiento de la propia línea de producción.

En este panel encontraremos un botón que se tendrá que pulsar en caso de que haya sucedido un problema extremadamente grave con la línea de producción. Al ser pulsado se activará la emergencia general de la línea y se encenderá la propia luz de emergencia del panel.



Figura 4.1: Figura del panel de mandos

Tendremos un interruptor de encendido, el cual nos permitirá hacer uso de los interruptores del panel. Habrá otro interruptor que es el que hace que el HMI pueda cambiar también los modos de funcionamiento en conjunto con el panel. Y habrán otros dos últimos interruptores que nos permitirán cambiar los modos de funcionamiento. Uno nos permitirá cambiar entre el modo manual al automático y el otro, estando dentro del modo automático podremos cambiar del modo paro al modo marcha. Finalmente el modo de funcionamiento que esté activado será reflejado en uno de los tres leds correspondientes que están situados en el panel.

4.2 Descripción de los modos de funcionamiento de la línea de producción

En todas las líneas de producción actuales existen numerosos modos de funcionamiento. Estos modos de funcionamiento hacen que el control de las máquinas de una planta industrial sea mucho más accesible y robusto frente a fallos en el proceso. Así que para este proyecto se han decidido implementar los siguientes tres modos de funcionamiento:

- **Modo manual:** Modo en el que los operarios tienen el control total de la línea de producción pudiendo así manejar los actuadores a su antojo.
- **Modo automático paro:** Modo en el que toda la línea de producción está parada donde no se tiene el control. Se puede haber entrado en este modo por decisión de los operarios o porque haya saltado alguna alarma.
- **Modo automático marcha:** Modo en el que la línea de producción funciona con total autonomía siguiendo las instrucciones que se le ha programado.

4.2.1 Selección del modo de funcionamiento

Los modos de funcionamiento irán cambiando en base a la decisión que tome el operario. En el panel de mandos situado en la escena de Factory I/O hay un botón que sirve para dotar al HMI del control de los modos de funcionamiento. Al decidir utilizar esta función ni el HMI ni el panel de mandos tendrán un control completo de los modos de funcionamiento. Ambos tendrán que seguir las siguientes restricciones:

- El panel de mandos tiene que estar encendido para poder realizar el cambio de modo
- Una vez encendido el panel se activará el modo manual
- Desde el modo manual solo se podrá cambiar al modo automático paro si el interruptor correspondiente ha sido activado y no está la alarma de emergencia general activada
- Desde el modo automático paro se cambiará al modo automático marcha si se pulsa el interruptor correspondiente y no está la alarma de emergencia general activada
- Desde el modo automático paro se cambiará al modo manual si está la alarma de emergencia general activada
- Desde el modo automático marcha se cambiará al modo manual si se pulsa el interruptor correspondiente y está la alarma de emergencia general activada
- Desde el modo automático marcha se cambiará al modo automático paro si se pulsa el interruptor correspondiente y no está la alarma de emergencia general activada

Y además, tanto el panel como el HMI, tendrán que estar en el mismo estado (modo de funcionamiento seleccionado). Es decir, si por ejemplo queremos cambiar al modo

automático marcha tanto el HMI como el panel de mandos tendrán que estar previamente en el modo automático paro y darle a su interruptor correspondiente para cambiar al modo automático marcha.

A continuación se mostrará un diagrama en lenguaje GRAFCET (se explica que es un GRAFCET en la [subsección 4.2.5](#)):

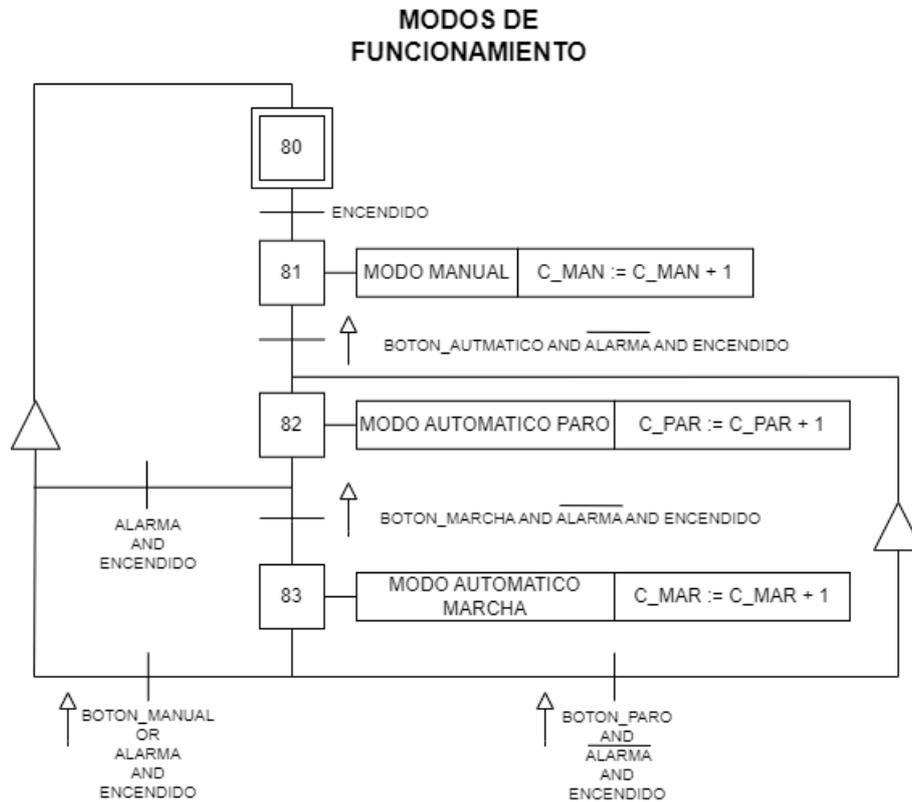


Figura 4.2: GRAFCET de los modos de funcionamiento. Fuente: elaboración propia

Como se ve en la [figura 6.6](#) en el propio GRAFCET se observan contadores. Estos contadores nos serán útiles para poder realizar la monitorización de la línea de producción. En este caso la monitorización será para saber cuantas veces se ha activado cada modo de funcionamiento.

4.2.2 Descripción del modo manual

El modo manual es activado por los operarios mediante el panel de mandos, y en conjunto por el HMI. Este modo trata de que el operario sea el que tenga el control total de las máquinas y no el automatismo.

Para este proyecto el control de las máquinas se ha desarrollado en el HMI mediante el uso de interruptores que solamente se muestran y se pueden usar cuando se entra en el modo manual. Estos interruptores tienen la función de activar o desactivar los actuadores de las máquinas. Además para el proyecto se ha decidido que también se mostrarán el

estado de los sensores en modo automático. Esto lo he encontrado bastante útil ya que los operarios desde el propio HMI pueden ver si existe algún problema con el sensor (en Factory I/O no podemos simular esto, pero sería muy útil en el mundo real), si hay algún objeto se ha quedado bloqueando el haz de luz del sensor o para que el operario sepa si el producto se encuentra junto al sensor, para posteriormente poder manejar el producto con las máquinas.

4.2.3 Descripción del modo automático paro

El modo automático paro, como tal se trata del más simple de los tres modos que han sido implementados en el proyecto. Este modo puede ser cambiado por parte del operario mediante el panel de mandos, y también en conjunto con el HMI. Pero por otra parte este modo puede ser activado si se ha activado alguna alarma en el sistema.

Lo que hace este modo es desactivar por completo todos los actuadores que están implicados. Es decir, si salta la alarma general se desactivarán todos los actuadores. Y si se activa la alarma de una parte en concreto de una máquina solamente se desactivarán los actuadores de esa parte, para que el resto de la línea de producción pueda continuar con normalidad. El funcionamiento de las alarmas ha sido detallado previamente en la [sección 5.2](#).

La única complejidad de este modo solamente reside en la propia gestión de las alarmas que realicen los operarios. Para esto en el HMI se han introducido dos botones en cada parte de las máquinas. Estos botones son llamados 'RESET' y 'CONTINUE'. La función que tiene el botón 'RESET' como su propio nombre indica es devolver a la parte de la línea de alimentación que ha entrado en una alarma a su estado inicial al desactivar la alarma. Esta acción se realizaría cuando alguna parte del proceso ha podido fallar y si no se empieza desde el estado inicial se va a volver a cometer algún fallo. Y por lo tanto volvería a saltar la alarma. Y el botón 'CONTINUE' hace que al desactivar la alarma la parte de la línea que ha entrado en estado de alarma continúe por el estado en el que se ha quedado. Justo al contrario que el botón de 'RESET' esta acción se realizaría cuando se ha podido solucionar el problema y sabemos que el proceso puede continuar por donde ha sido parado sin volver a producir de nuevo una alarma.

4.2.4 Descripción del modo automático marcha

Esta descripción nos dará toda la información necesaria sobre el funcionamiento del modo automático marcha el cual ha sido implementado en la línea de producción. Cabe destacar que este ha sido el modo más complejo de implementar. Ya que se ha tenido que estudiar en detalle el funcionamiento de los [Sensores y actuadores empleados en la línea de producción](#). Y aplicar su uso correspondiente en las partes de la línea de producción para que todo el flujo de la línea de producción funcione correctamente.

El modo automático marcha trata de que la propia línea de producción funcione sin ningún tipo de intervención humana y que la línea opere de manera independiente. Para esto se tiene que seguir un conjunto de órdenes/instrucciones que tienen que ser bien detalladas para su correcto funcionamiento. Por otra parte cabe destacar que la intervención humana

es completamente necesaria ya que se pueden producir fallos, es necesario que alguien responsable sea capaz de tomar decisiones teniendo los conocimientos de como se ha implementado el modo automático marcha, siempre será necesario revisar la calidad de la producción y del correcto funcionamiento de la línea, implementación de posibles mejoras, etc. . .

4.2.5 Diseño del programa en TIA Portal

Para poder desarrollar el programa se ha tenido que hacer un diseño previo del modo automático marcha. Este diseño consta de lenguaje GRAFCET y ecuaciones lógicas (las ecuaciones lógicas solo se han usado para el desarrollo del modo automático).

Lenguaje GRAFCETs

GRAFCET (Michel, 1990) se trata de un diagrama que describe la representación de un proceso que se quiere automatizar definido por estados, condiciones y variables. El automatismo de la línea de producción para este proyecto ha sido diseñado de manera inicial en lenguaje GRAFCET. Los grafkets cumplen con la función que han de desarrollar cada una de las partes de la línea de producción.

De forma inicial, para realizar este diseño se pensó en hacer un total de diez GRAFCETs, uno por cada parte implicada en la línea de producción. Pero conforme se fueron desarrollando se observaba que algunos se podrían reutilizar para distintas máquinas ya que los procesos que hacían eran muy similares o iguales. Así que esto nos ahorraría desarrollar un total de diez GRAFCETs y simplificar todo este diseño en un total de seis diagramas GRAFCET. Y también tenemos que contar con el GRAFCET del panel de mandos por lo que tendríamos unos siete GRAFCETs. Todos los GRAFCET han sido creados mediante la web [Draw.io](https://draw.io) y son de elaboración propia.

Ecuaciones lógicas

Las ecuaciones lógicas se tratan de representaciones matemáticas de las relaciones lógicas y las variables binarias (variables booleanas).

Obtendremos estas ecuaciones de los GRAFCETs que hemos diseñado previamente. Y nos apoyaremos en ellas como si fuesen una especie de pseudocódigo para obtener el código en SCL e implementar el programa en TIA Portal. Se han creado tres tipos de ecuaciones lógicas, los cuales son los siguientes:

- **Ecuaciones de estado activo:** Describen el flujo que ha de seguir del sistema
- **Ecuaciones de set:** Relacionan las variables de estado del sistema con las variables de entrada
- **Ecuaciones de salida:** Relacionan las variables de salida con las variables de estado activo

4.3 Diseño HMI

Para el HMI se deberá desarrollar una interfaz que sea simple visualmente y que sea fácil de usar. Implementará la gestión de usuarios los cuales tendrán sus propios permisos. Se podrá visualizar las alarmas que han sido activadas y poder gestionarlas adecuadamente. Y se gestionará la línea de producción.

4.3.1 Gestión de usuarios y permisos

Al HMI solo tendrán acceso los usuarios creados que tengan los permisos correspondientes. Y además existirán los siguientes roles de usuario:

- **Administrator group:** Acceso total al sistema. Creación de usuarios, visualización de alarmas y control de la línea de producción.
- **Mantenimiento:** Se les permite tomar control de la línea de producción, ver las alarmas, pero no gestionar los usuarios
- **Basic:** Solo pueden iniciar sesión

Los usuarios creados son los siguientes:

- **Nombre: admin - contraseña: admin - permiso: Administrator group**
- **Nombre: mant - contraseña: mant - permiso: Mantenimiento group**
- **Nombre: basic - contraseña: basic - permiso: Basic**

4.3.2 Diseño de las vistas HMI

Como se ha comentado antes el diseño de la interfaz ha de ser simple ya que los entornos industriales suelen emplear este tipo de diseños. Esto se debe a que reduce considerablemente el riesgo de cometer errores, la adaptación de los operarios a la interfaz es rápida. Y además aumenta la eficiencia a la hora de llevar a cabo las operaciones que se desean de manera rápida y precisa.

Tendremos cuatro vistas principales:

- **Home:** Vista inicial posterior a hacer login donde podremos acceder al resto de vistas principales
- **Gestión de usuarios:** En esta vista se podrán crear usuarios y modificar sus permisos
- **Alarmas:** Se mostrarán las alarmas del sistema

- **Modos:** Es la vista más completa de todas. En esta vista se tendrá acceso a la gestión de los modos de la línea de producción desde el HMI.

Podremos observar datos de la planta de producción, como el número de palets recogidos y entregas realizadas.

Habrà una imagen interactiva de la línea de producción donde se mostrarà mediante un icono donde se han producido concretamente las alarmas y otro para saber que parte de la línea está activa.

Y además en esta imagen interactiva si presionamos sobre una parte de la línea de producción en concreto accederemos a la manipulación en concreto de esa parte. Que al estar en el modo manual podremos manipular sus actuadores y podremos ver que sensores tiene activos. También al entrar en el modo de alarma se nos mostrarán dos botones. Un botón llamado 'RESET' (Devolverá la maquina al estado inicial) y otro llamado 'CONTINUE' (La máquina continuará por el estado en el que se ha quedado antes de ser parada). Que al ser pulsados realizarán sus funciones específicas y resolverán las alarmas.

4.4 Datos a monitorizar con Grafana

Los datos que se monitorizarán mediante Grafana se mostrarán en paneles de distintos tipos. Y además los datos serán poco variados ya que Factory I/O no tiene una amplia variedad de sensores y actuadores como para tomar una amplia cantidad de distintos tipos de medidas.

La monitorización realizada en Grafana tendrá las siguientes ventanas con sus respectivos paneles:

- **Contadores modos del panel de mandos** (Se hará el conteo de cuantas veces se ha cambiado a cada tipo de modo en el panel de mandos):
 - Panel de número de veces en modo manual, representado numéricamente y en una gráfica de barras.
 - Panel de número de veces en modo automático paro, representado numéricamente y en una gráfica de barras.
 - Panel de número de veces en modo automático marcha, representado numéricamente y en una gráfica de barras.
- **Panel de los materiales** (Conteo de palets recogidos, productos a la espera de ser entregados y entregas realizadas):
 - Panel de número de piezas pequeñas procesadas a la espera, representado numéricamente.

- Panel de número de piezas grandes procesadas a la espera, representado numéricamente.
 - Panel de número de piezas pequeñas entregadas, representado numéricamente.
 - Panel de número de piezas grandes entregadas, representado numéricamente.
 - Panel de número de palets de piezas pequeñas recogidos, representado numéricamente.
 - Panel de número de palets de piezas grandes recogidos, representado numéricamente.
 - Panel de entregas generales realizadas, representado por un gráfico de pastel para hacer una comparativa.
- **Panel temporizadores cintas** (Se contabilizará el tiempo que pasan las cintas activas):
- Panel del tiempo de la máquina de inicio de piezas pequeñas (maquina de inicio 1), representado temporalmente en una gráfica lineal
 - Panel del tiempo de la máquina de inicio de piezas grandes (maquina de inicio 2), representado temporalmente en una gráfica lineal
 - Panel del tiempo de la máquina de entrega de bases, representado temporalmente en una gráfica lineal

5 Descripción de la solución

En este capítulo se verá solución tomada para el proyecto para llevar a cabo la virtualización de la línea de producción, el control sobre esta y la monitorización de los datos.

5.1 Sensores y actuadores incluidos en los procesos

Para realizar este proyecto se ha decidido usar Factory I/O ya que no se dispone de un entorno industrial real para poder realizar las pruebas. Por lo que los sensores y actuadores que estén incluidos/involucrados en los procesos de la línea de producción se escogerán los que Factory I/O incluye.

Gracias a virtualizar la implementación de la línea de producción en las escenas de Factory I/O se ha podido poner a prueba todo el funcionamiento sin causar ningún tipo de accidente/inconveniente en un entorno físico.

5.1.1 *Sensores y actuadores empleados en la línea de producción*

Como se ha visto anteriormente en la [figura 3.4](#) la línea consta de diversos sensores y actuadores. Y esa será la distribución de las estaciones que tendrá el proyecto. Los sensores y actuadores utilizados en el proyecto se detallarán en las siguientes subsecciones.

Sensores

- **Sensor de difusión:** sensor fotoeléctrico que es capaz de detectar objetos sólidos. Se usará este sensor para saber si los productos entran o salen de las distintas partes de la línea de producción.
- **Matriz de luces:** este es un sensor el cual porta de un conjunto de haces de luz de manera paralela para crear un campo de detección. Existen diversas configuraciones dentro de Factory I/O para este sensor. Pero nosotros utilizaremos la configuración numérica para obtener un valor entero de la altura de los productos. Este sensor estará

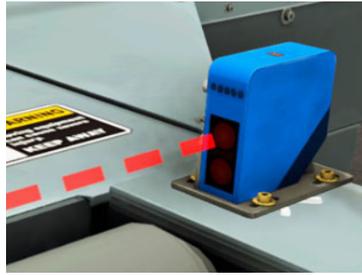


Figura 5.1: Sensor de difusión

únicamente situado al final de la cinta inicial para poder etiquetar los productos como grandes o pequeños.

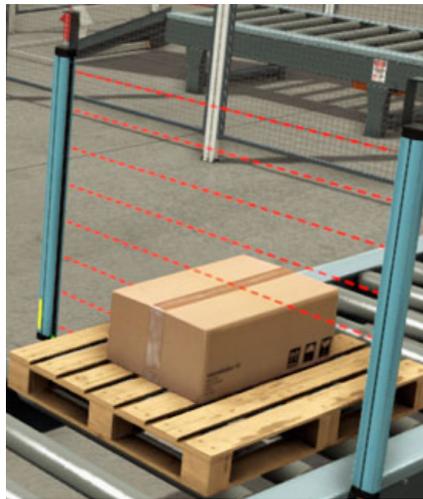


Figura 5.2: Matriz de luces

Aparición de los productos está en un intervalo entre 12 y 15 segundos. Los productos que aparecen son Box (S) y Blue Raw Material. Y lo hacen encima de un Square pallet.

Al final de su producción los Box (S) son recogidos en un Square pallet que aparece entre 5 y 6 segundos. Y para los Blue Material son recogidos en un Stackable Box que está situado encima de un Square pallet (tiempo de aparición similar al anterior).

Actuadores

- **Transportador de rodillos:** se trata del típico transportador de objetos. Es capaz de controlarse mediante configuraciones analógicas y digitales. Para el proyecto se ha decidido emplear la configuración digital. Ya que en esta configuración solo se nos permite girar los rodillos en un sentido y no es necesario que rote en el sentido opuesto, además de tener una velocidad constante.



Figura 5.3: Transportador de rodillos

- **Cinta transportadora:** se trata de la típica cinta transportadora, la cual es capaz de transportar objetos que no son pesados. Y su funcionamiento es similar al del transportador de rodillos.



Figura 5.4: Cinta transportadora

- **Plataforma giratoria o Turntable:** se trata de una plataforma de la cual su uso general está orientado a la ordenación de objetos. Esta porta rodillos capaces de transportar objetos en ambas sentidos, además de sensores inicial y final que detectan si el objeto está entrando o saliendo de la plataforma. Y también es capaz de girar con tal de posicionar el objeto que porta, sabremos si ha girado mediante un sensor que tiene incorporado. Para el proyecto utilizaremos la configuración monoestable. Esta configuración hace que el sensor de giro mande una señal de activación cuando la plataforma ha girado completamente y el giro solo se puede realizar en un solo

sentido. El hecho de que solo se nos permite girar la plataforma en un solo sentido nos ahorra un poco la tarea de programar este actuador.

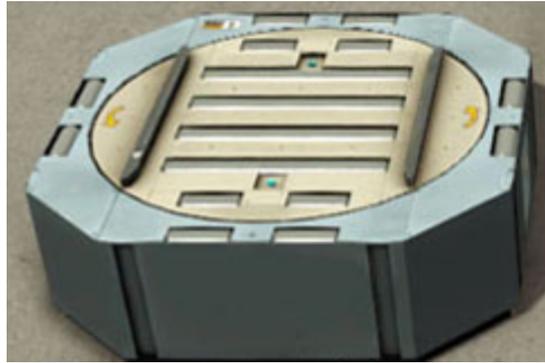


Figura 5.5: Plataforma giratoria o turn table

- **“Pick & Place” de dos ejes:** esta máquina es usada para recoger y colocar los objetos que le llegan. Es conveniente que los objetos estén lo más alineados posibles a la máquina para que esta los recoja y los coloque de una manera adecuada. La configuración usada ha sido la digital. De manera muy resumida esta máquina es capaz de rotar de manera horizontal. Además de desplazar su pinza de manera vertical y hacer que coja los objetos y los suelte a conveniencia y de tener sus propios sensores.



Figura 5.6: Pick & place

- **Centro de mecanizado:** este tipo de estación/máquina se utiliza para procesar los objetos y fabricar tapas y bases a partir de productos de materia prima. Consta de un robot articulado. El cual está a la espera de que un producto de materia prima sea colocado en la entrada del centro mecanizado. El robot recogerá el producto y lo colocará en la máquina CNC donde se procesará. Una vez procesado el robot volverá a recoger y colocar el objeto, pero esta vez en la salida, completando así el ciclo de la estación. El estado de la estación es mostrado mediante leds. El progreso de procesado de la máquina será indicado en una pantalla. Y la estación porta un botones manuales para empezar, detener y reiniciar el proceso (De esto no se hará

mucho uso en el proyecto, ya que se gestionará directamente con la programación en TIA Portal) .



Figura 5.7: Centro de mecanizado

- **Emisor de objetos:** este elemento es el que realiza la aparición de los objetos que serán usados en las escenas de Factory I/O. Podemos escoger los objetos que queremos que aparezcan dentro de su superficie. Dentro de los objetos queremos que sean creados por el emisor existen dos tipos. Las bases (palets y cajas) y las piezas (materiales, cajas, productos procesados. . .). También permite programar los intervalos de tiempo en los que aparecen los objetos, el límite de objetos que es capaz de emitir, orientación de estos, si está activado. . . En resumen es uno de los elementos más clave en Factory I/O, ya que nos permite en parte controlar una parte del flujo de la línea de producción.

El emisor de objetos puede ser programado mediante TIA Portal. Pero para este proyecto no será necesario. Ya que realizando pequeños ajustes en las configuraciones sobre este elemento en Factorio I/O no es más que suficiente. Porque lo que nos interesa es que hagan que aparezcan los objetos en un intervalo de tiempo y unos objetos en concreto.

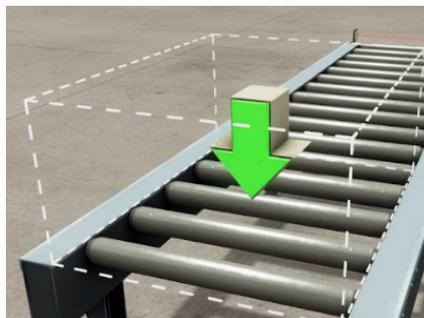


Figura 5.8: Emisor de objetos

- **Eliminador de objetos:** este elemento puede realizar acciones similares a las del emisor de objetos, la única diferencia es hace desaparecer por completos los objetos de la escena. También puede ser programado mediante TIA Portal. Pero con la configuración que viene por defecto en Factory I/O es más que suficiente. No nos interesa hacer nada complejo.

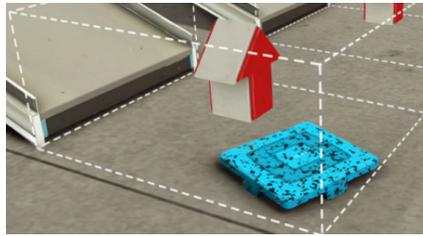


Figura 5.9: Eliminador de objetos

5.1.2 Objetos usados en la línea de producción

Para el proyecto se han usado diferentes objetos, los cuales son creados en la escena de Factory I/O mediante el emisor de objetos. Estos objetos se dividen en dos tipos: piezas y bases. En las siguientes subsecciones serán detallados.

Piezas/productos

Las piezas se tratan de objetos que pueden ser procesados o no y que no sirven para transportar o apilar objetos. Las piezas empleadas en el proyecto son las siguientes.

- **Materia prima:** se trata de un objeto simple el cual puede ser procesado en el *Centro de mecanizado* para convertirse en un objeto llamado [Base de producto](#).



Figura 5.10: Producto de materia prima

- **Base de producto:** Objeto que ha sido previamente procesado. Este puede ser ensamblado con una tapa. Para este proyecto el ensamblado con la tapa no ha sido contemplado. Simplemente se realizará la entrega de la base.



Figura 5.11: Base de producto

- **Caja pequeña:** Caja de pequeñas dimensiones la cual no puede ser procesada y se realizará la entrega de manera íntegra de esta.



Figura 5.12: Caja pequeña

Bases

Las bases se tratan de objetos que son capaces de transportar o apilar otros objetos con el fin de agruparlos. Las bases que se han usado son las siguientes:

- **Palet cuadrado:** se ha usado esta base porque al intentar transportar objetos en el *Transportador de rodillos* los objetos se solían quedar atascados entre los rodillos y no se desplazaban adecuadamente. Lo que hacía imposible en muchas ocasiones llevar a cabo toda la producción de la línea.

Así que desde el inicio de la línea de producción los objetos/productos aparecen encima del palet cuadrado.



Figura 5.13: Palet cuadrado

- **Caja apilable:** Esta base se ha usado en las zonas finales de la línea de producción. En conjunto con el palet cuadrado. Es decir que el palet cuadrado porta encima una caja apilable. Y la caja apilable contendrá los objetos en su interior. He decidido hacerlo de esta manera para que sea un poco más elegante la entrega de objetos y además para que la máquina *‘Pick & Place’ de dos ejes* sea capaz de colocar los objetos dentro de la caja de manera correcta.



Figura 5.14: Caja apilable

5.1.3 Partes en las que se ha dividido la línea de producción

La línea de producción se ha decidido dividir en un total de 10 partes/secciones para que el proceso de programación de la línea de producción en sus distintos modos de funcionamiento no sea tan complejo a la hora de desarrollarlo y entenderlo. En esta sección se mencionará el funcionamiento que tiene cada una de las partes implicadas en la línea de producción, a donde irán desplazando los productos y los sensores y actuadores que tiene cada una de ellas.

En la [figura 5.15](#) se pueden ver las partes en las que ha sido dividida la línea de producción.

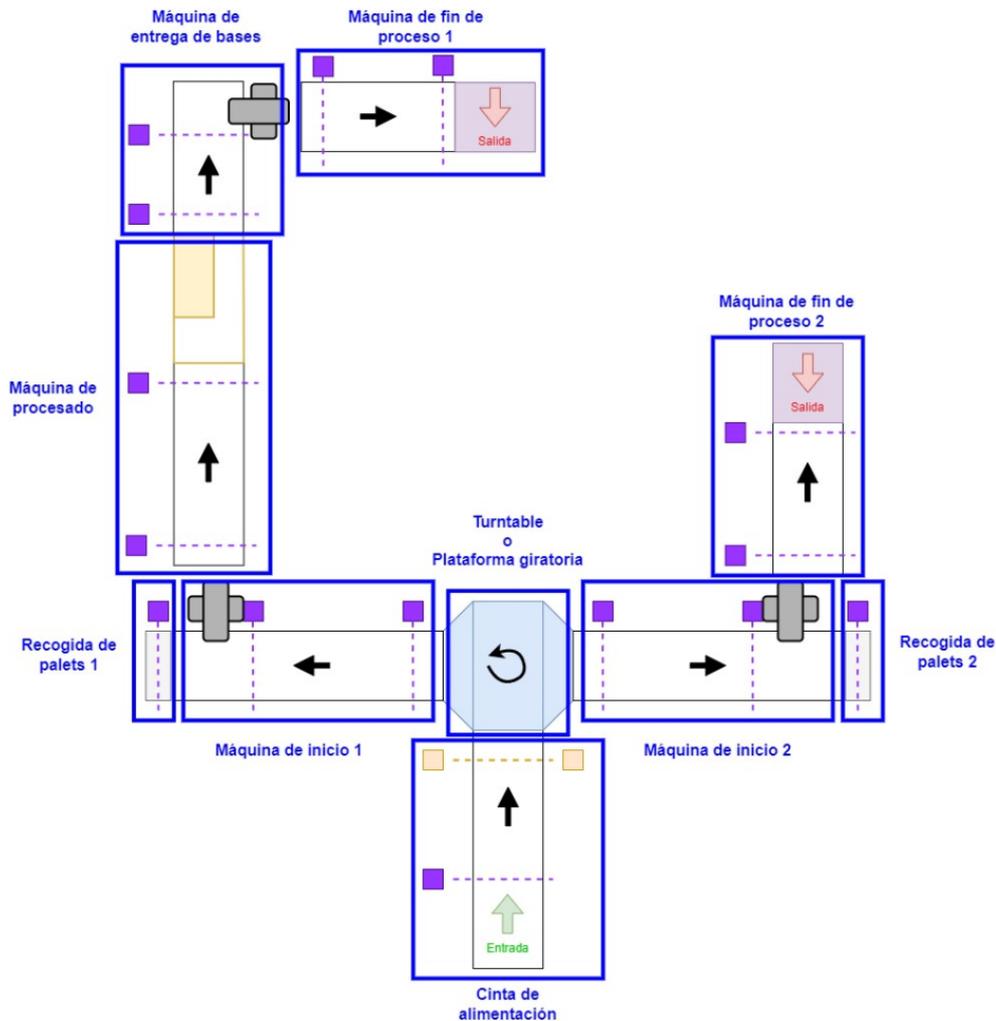


Figura 5.15: Partes de la línea de producción

A continuación se describirá las funciones que realizan las distintas partes/secciones de la línea de producción. Se mencionarán los **Sensores y actuadores** empleados en la línea de producción y **Objetos** usados en la línea de producción

Cinta de alimentación

Esta es la parte inicial de la línea de producción. Está formada por un **Transportador de rodillos**, **Sensor de difusión**, **Emisor de objetos** y una **Matriz de luces**.

Se ha usado un emisor de objetos para simular como si un operario estuviera colocando los objetos en el transportador de rodillos. Se irán colocando los objetos mediante el emisor en un intervalo de tiempo. Se colocará un **Palet cuadrado** y justo encima una **Caja pequeña** o un **Producto de materia prima** de manera aleatoria, las probabilidades de que se coloque uno de estos objetos son las mismas.

El transportador de rodillos se deberá de mover cuando el sensor de difusión colocado al inicio detecte un objeto y el transportador de rodillos desplazará los objetos hasta el final de su recorrido. Y justo en el final mediante la matriz de luces se medirá la altura de los objetos para empezar la clasificación de objetos. La clasificación se realizará en el Turntable (subsección 5.1.3).

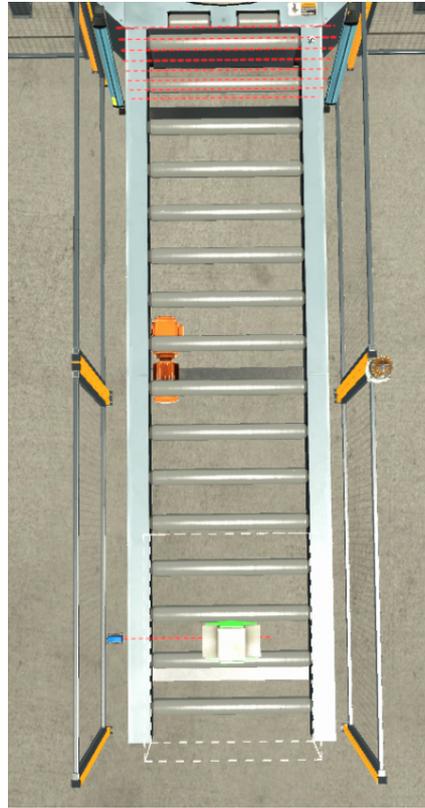


Figura 5.16: Cinta de alimentación, parte inicial de la línea de producción

Turntable o plataforma giratoria

Esta parte solo consta de una sola máquina, la cual incorpora diversos actuadores y sensores. Se trata de **Plataforma giratoria o turn table**. En esta parte se recibirán los objetos a los que se les ha medido la altura para ser clasificados. La **Cinta de alimentación, parte inicial de la línea de producción** estará desplazando el objeto y este lo detectará mediante su sensor trasero, una vez detectado empezará a girar los rodillos desplazando el objeto hasta que el límite delantero haya detectado el objeto.

Cuando el objeto esté activando el límite delantero el turntable rotará 90° en sentido antihorario. Y el objeto será desplazado hacia adelante si su altura es de 192 mm y atrás si es de 224 mm. Hay que tener en cuenta que el turntable ha rotado por lo que la dirección ya no será la misma que la de antes. Para el cálculo de la altura se ha tenido también en cuenta la altura del palet.

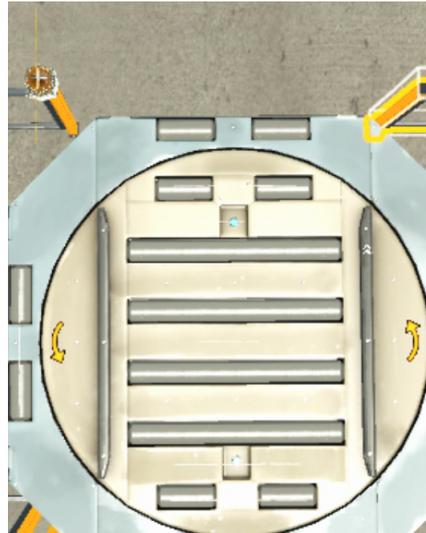


Figura 5.17: Turntable o plataforma giratoria

Los objetos desplazados hacia adelante se enviarán a [Máquina de inicio 1](#) y los desplazados hacia atrás a [Máquina de inicio 2](#).

Máquina de inicio 1

La máquina de inicio 1 está formada por un [Transportador de rodillos](#), dos [Sensor de difusión](#) y una máquina [Pick & place](#). Los objetos enviados hacia adelante por la [Turntable o plataforma giratoria](#) son detectados por uno de los sensores de difusión que están colocados al inicio del transportador de rodillos. Una vez detectado el objeto este será desplazado por el transportador de rodillos hasta llegar a otro sensor de difusión que estará colocado al lado de la máquina pick & place para que pueda coger solamente la materia prima.

Una vez el pick & place coge la [Producto de materia prima](#) la cinta se desplazará durante un breve periodo de tiempo apartado el palet que se ha quedado sin utilidad alguna enviándolo a la parte de [Recogida de palets 1](#). Y durante este proceso la pick & place enviará la materia prima realizando rotaciones y desplazamientos de su brazo para que el objeto llegue a la parte de la [Máquina de procesado](#).



Figura 5.18: Máquina de inicio 1

Recogida de palets 1

Esta parte es la más simple de todas. Una vez la [Máquina de inicio 1](#) está desplazando el [Palet cuadrado](#) en su proceso final este caerá por la rampa de la recogida de palets y será detectado por el [Sensor de difusión](#) al final de la rampa. Y una vez haya caído por la rampa este será eliminado de la escena mediante un [Eliminador de objetos](#) para simular que el palet ha sido recogido y no nos estorbe.

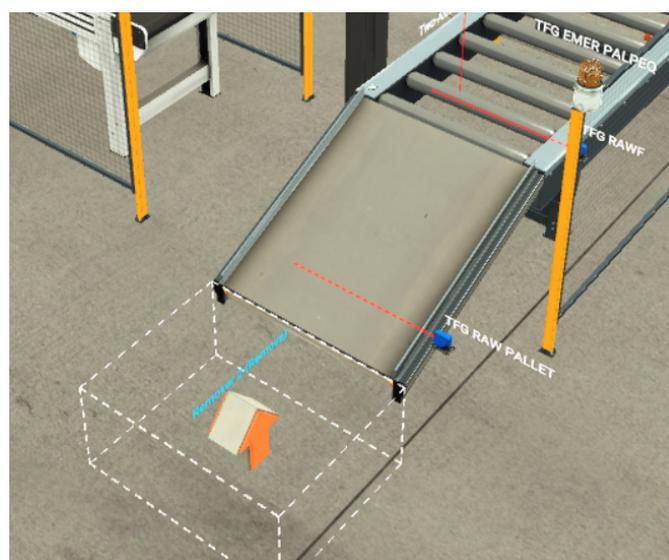


Figura 5.19: Recogida de palets 1

Máquina de procesado

Desde la [Máquina de inicio 1](#) se irán recibiendo las materias primas mediante el pick & place. Estas serán dejadas al inicio de [Cinta transportadora](#) donde serán detectadas por un [Sensor de difusión](#). Una vez detectadas por el sensor de difusión la cinta comenzará a desplazar las materias primas sobre ella. A mitad de la cinta se encuentran una especie de rodillos que sirven para ajustar correctamente las materias primas en la cinta, ya que el pick & place las podría colocar en una posición incorrecta sobre la cinta. Esto nos solventará este problema de que los productos se puedan quedar bloqueando el acceso al [Centro de mecanizado](#).

Y una vez la materia prima llega al final de la cinta esta es detectada por otro sensor de difusión el cual hace que el centro de mecanizado se active. Y una vez activado el brazo robótico del centro de mecanizado se encargaría de recoger la materia prima y colocarla dentro del centro de mecanizado para transformar la materia prima en una [Base de producto](#). Finalmente, una vez ya se ha obtenido la base de producto el brazo robótico retiraría la base y la sacaría hacia afuera del centro de mecanizado, justo donde se encuentra la [Máquina de entrega de bases](#).

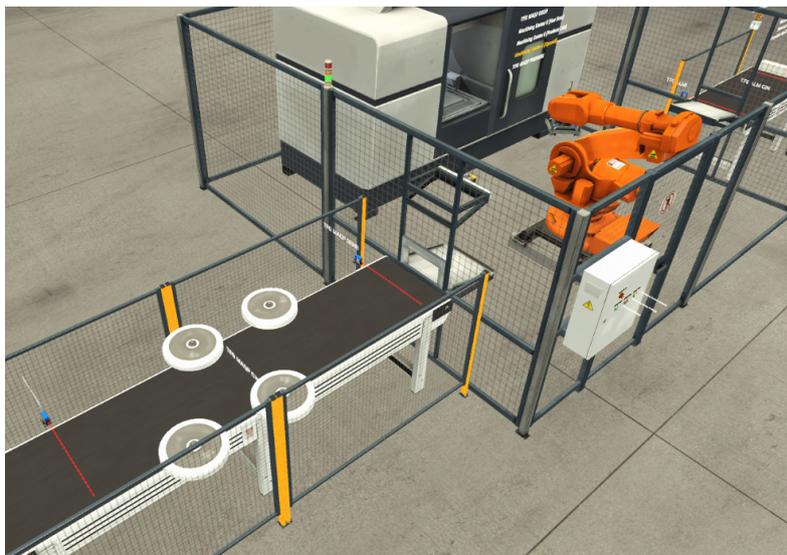


Figura 5.20: Máquina de procesado

Máquina de entrega de bases

El funcionamiento de la maquina de entrega de bases es similar al de la [Máquina de inicio 1](#). Los objetos que llegarían a esta parte serían bases de materia prima desde la [Máquina de procesado](#). El actuador que transportaría las bases sería una [Cinta transportadora](#) y el desplazamiento por la cinta vendría dado gracias a dos [Sensor de difusión](#). Y las [Base de producto](#) serían recogidas mediante un [Pick & place](#) el cual sería activado por el sensor de difusión del final de la cinta. Y las bases de producto serían enviadas a una de las partes finales de la línea llamada [Máquina de fin de proceso 1](#).

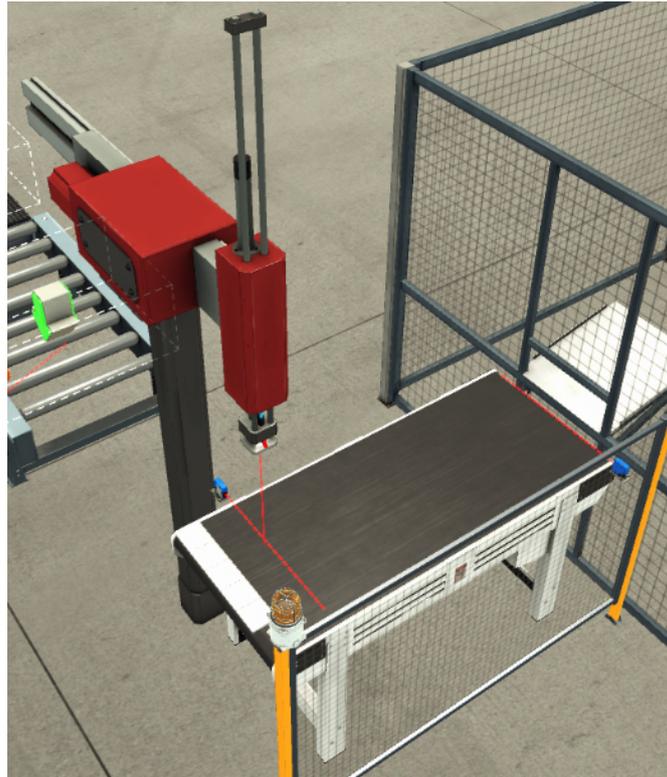


Figura 5.21: Máquina de entrega de bases

Máquina de fin de proceso 1

Esta se trata de una de las dos partes finales involucradas en la línea de producción. Tenemos un [Transportador de rodillos](#), un [Sensor de difusión](#) al inicio, y otro al final. Además al inicio hay un [Emisor de objetos](#) el cual creará [Palet cuadrados](#) y sobre estos pondrá una [Caja apilable](#), donde se introducirán los objetos que llegan de [Máquina de entrega de bases](#). Para este proyecto en esta parte en concreto se ha decidido permitir que solo se introduzcan dos [Base de producto](#) por caja apilable con tal de darle un poco de dinamismo a esta parte. Una vez introducidas las dos bases de producto en las cajas apilables el transportador de rodillos comenzará a funcionar, en caso contrario esperará hasta que se hayan introducido dos bases.

Y Finalmente una vez está el transportador de rodillos en movimiento y se ha detectado la caja al final del transportador, el palet, la caja y las dos bases serán eliminadas de la escena para simular su entrega/recogida por parte del personal de la planta.

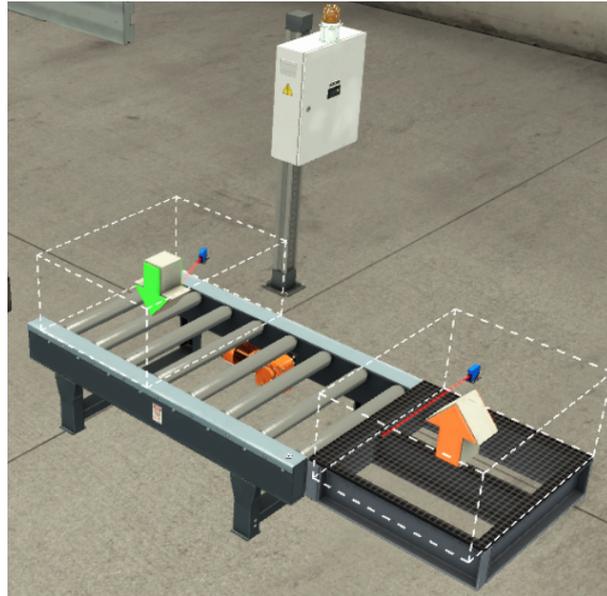


Figura 5.22: Máquina de fin de proceso 1

Máquina de inicio 2

Esta parte tiene los mismos actuadores y sensores que [Máquina de inicio 1](#), además de tener un funcionamiento similar. Lo único que la diferencia es que los objetos los recibe cuando el [Turntable o plataforma giratoria](#) desplaza los objetos hacia atrás. Y estos objetos se tratan de cajas pequeñas. Las cuales serán recogidas por el pick & place y enviadas a [Máquina de fin de proceso 2](#).

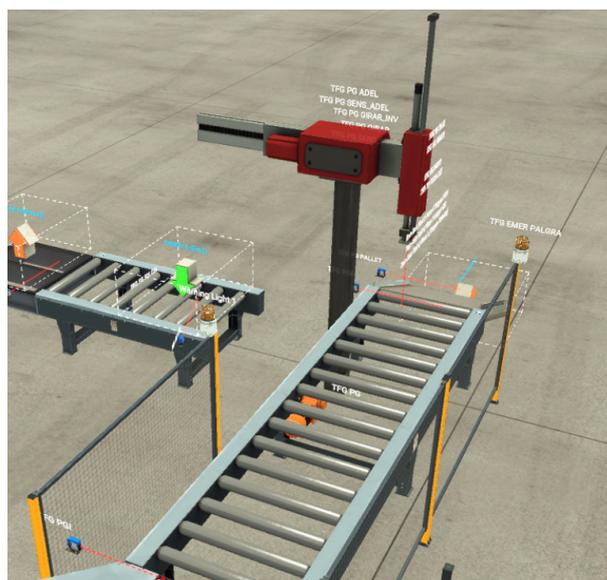


Figura 5.23: Máquina de inicio 2

Recogida de palets 2

La recogida de palets en esta parte funciona igual que en la parte de [Recogida de palets 1](#) y tiene los mismos sensores y actuadores.

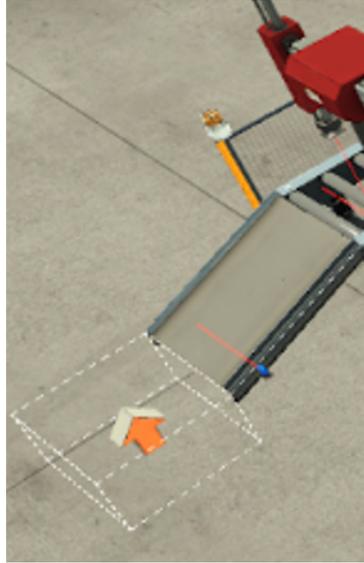


Figura 5.24: Recogida de palets 2

Máquina de fin de proceso 2

Este se trata de la otra parte final involucrada en la línea de producción. Los sensores y actuadores son iguales que en [Máquina de fin de proceso 1](#). Y el proceso a seguir también es similar. Lo único que cambia es que no se emplean cajas apilables y los productos son [Caja pequeñas](#). Pero el límite en vez de ser dos como con las bases de producto esta vez es uno. Por lo que una vez colocada una caja pequeña encima de un palet se realizaría la entrega y la caja sería recogida.

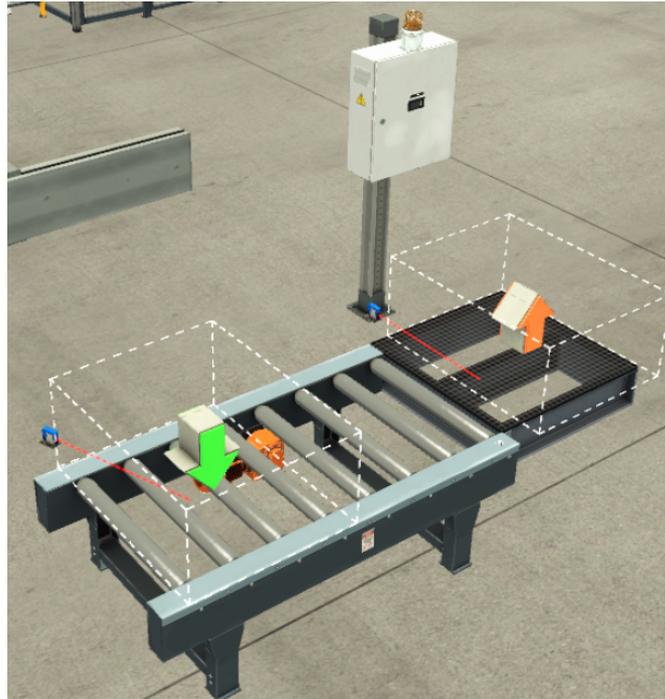


Figura 5.25: Máquina de fin de proceso 2

5.2 Alarmas de emergencia

En toda línea de producción es muy necesario tener el conocimiento de cuando se ha producido un error. Es por ello que se ha de ser previsor para poder evitarlos lo antes posible y de no ser así, al menos que ocasionen los mínimos daños posibles. Una gran ayuda para evitar los errores previstos durante el desarrollo son las alarmas de emergencia (Berger, 2014). Las cuales se activarán en cuanto uno de estos errores previstos haya sido detectado. Y pararán las partes en concreto donde se hayan producido. Las partes donde no se haya producido las alarmas continuarán con su funcionamiento con total autonomía.

Para este proyecto se ha puesto una alarma de emergencia por cada parte de la línea de producción, con su respectiva luz de emergencia. La intención de esto es para que no hayan demasiadas, pero que al menos haya un poco de variedad. Las alarmas que se han puesto son las siguientes:

- **Seta de emergencia general:** Esta es la emergencia más grave que puede ocurrir en la línea de producción. Esta alarma saltaría cuando el operario presiona la seta de emergencia situada en el panel. La seta de emergencia se pulsaría en caso de que haya ocurrido un problema extremadamente grave o no previsto en el resto de alarmas. Y lo que ocasionaría es que todo el automatismo se pararía por completo. Entrando en el modo manual para que los operarios sean capaces de resolver el problema. Y reiniciando los estados de todas las máquinas.

- **Error en la cinta de alimentación:** Saltaría cuando el operario pulsa la seta de emergencia que se encuentra al lado de la cinta de alimentación. Parando solamente la [Cinta de alimentación](#).
- **Producto no llega a la máquina:** Se producirá esta alarma cuando los transportadores de rodillos o la cinta transportadora de [Máquina de inicio 1](#), [Máquina de inicio 2](#) y [Máquina de entrega de bases](#) haya estado activa más de doce segundos. Parando así la máquina en la que se haya producido la alarma.

Esta alarma se ha desarrollado para evitar saber si el objeto se ha caído de la cinta o ha sido retirado accidentalmente para no bloquear lo máximo posible el resto de máquinas.

- **Objetos detectados en el sensor frontal y trasero:** Saltaría cuando se hayan activado el sensor frontal y trasero del [Turntable o plataforma giratoria](#).

Se ha realizado la implementación de esta alarma para evitar que haya algún objeto que pudiese bloquear el transportador de rodillos.

- **Palet atascado:** Saltaría la alarma cuando el palet no haya sido entregado en unos cuatro segundos desde que se tenía previsto ser entregado parando así [Recogida de palets 1](#) y [Recogida de palets 2](#).
- **Problema detectado con el producto:** Saltaría la alarma cuando el operario pulse la seta de emergencia que se ha colocado en [Máquina de procesado](#). La causa de esta alarma es porque la pieza se puede quedar bloqueada en el hueco que tiene esta parte antes de ser procesada o que pueda haber algún error procesando las piezas.
- **Varios objetos en la cinta:** En [Máquina de fin de proceso 1](#) y [Máquina de fin de proceso 2](#) se encuentran objetos que activen sus sensores de inicio y de fin a la vez. Esta alarma ha sido creada para evitar que hayan objetos que bloqueen la entrega final de los productos y lleguen a su destino de manera adecuada.

En todas las alarmas menos en la de la 'Seta de emergencia general' se tendrán las opciones en el HMI para que los operarios deseen continuar por el estado en el que se ha quedado las máquinas o reiniciándolos, volviendo así al estado inicial de las máquinas.

5.3 Dispositivos de control y comunicaciones utilizados

Los dispositivos de control y de comunicaciones empleados en la línea de producción virtualizada son aquellos que se encontraban en el laboratorio. En las subsecciones de esta misma sección se mencionarán los dispositivos utilizados con su respectivas descripciones y características.

5.3.1 PLC Siemens S7-1200

Un PLC (Programmable Logic Controller) se trata de uno de los dispositivos electrónicos más populares en la industria. Como su nombre indica es aquel que dirige la lógica de los procesos industriales. Ya sea automatizando los procesos o bien controlándolos de forma manual.

Los PLC tienen tanto entradas y salidas analógicas como digitales para poder conectarse poder recibir las señales de los sensores o poder tomar el control de los actuadores. Para este proyecto no se hará uso ni de las entradas, ni de las salidas. Ya que se trabaja en un entorno virtualizado y no físico. Por lo que también para lo que conlleva la programación mediante el PLC solo se harán uso de las variables que se creen en memoria. Para poder programar un PLC será necesario hacer uso de un entorno de desarrollo de software. Además, para el proyecto se ha hecho uso del modelo de PLC que hay en el laboratorio. Se trata del PLC Siemens S7-1200, en concreto el modelo CPU 1214C AC/DC/Rly con la versión V1.4.

Algunas de las características del PLC Siemens S7-1200, CPU 1214C AC/DC/Rly son:

- 100KB Work Memory
- 120/240VAC Power supply
- HMI and PLC, to PLC Communication
- PROFINET interface for programming



Figura 5.26: PLC Siemens S7-1200, CPU 1214C AC/DC/Rly

5.3.2 HMI KTP700 Basic PN

Un HMI (Human Machine Interface) (NOLASCO SANDOVAL, 2021) es un panel donde el usuario es capaz de comunicarse con máquinas o un software de una manera interactiva. La definición puede ser un poco ambigua. Porque un dispositivo móvil según esta descripción, también podría serlo. Pero con HMI nos referimos a los paneles interactivos situados en entornos industriales. Los HMI son usados por los operarios de las plantas industriales para poder realizar monitorizaciones del estado del sistema, ver datos en tiempo real, visualizar y gestionar alarmas y realizar el control de procesos.

El HMI empleado en el proyecto ha sido el SIMATIC HMI, el modelo KTP700 Basic PN y con la versión 14.0.1.0. Cabe destacar que al pertenecer al modelo BASIC las características que este porta son muy simples. Ya que por ejemplo los usuarios que se crean mediante el HMI y los permisos que se les da no son persistentes. Ya que al apagar el HMI estos datos son eliminados. Por lo que lo más conveniente sería dar a los usuarios de alta con los respectivos permisos dentro en el entorno de desarrollo. Además el este modelo de HMI no permite crear scripts dentro del entorno para poder gestionar las vistas creadas o las variables, en este aspecto es un poco primitivo. Pero para el desarrollo de este proyecto no ha supuesto ningún tipo de problema.

Algunas de las características del SIMATIC HMI, KTP700 Basic PN:

- **7" TFT Display**
- **800 X 480 pixel**
- **64K colors**
- **Key and Touch operation**
- **8 Function keys**
- **PROFINET interface**



Figura 5.27: SIMATIC HMI, KTP700 Basic PN

5.4 Arquitectura de comunicaciones

La arquitectura de comunicaciones es una de las estructuras más importantes a desarrollar y a seguir cuando tratamos de conectar distintos dispositivos y tecnologías. Nos aporta una visión mucho más amplia de una parte importante del proyecto y nos sirve como una especie de hoja de ruta.

5.4.1 *Diseño de la arquitectura para la comunicación*

Lo primero de todo y lo más importante es saber que arquitectura/modelo se va a utilizar para poder realizar las comunicaciones con los distintos elementos/aplicaciones del sistema. La arquitectura en la que se ha basado este proyecto ha sido en la arquitectura de los sistemas OPC UA (Open Platform Communications Unified Architecture)(González et al., 2019).

Esta arquitectura se trata de un estándar muy popular entre las comunicaciones industriales, es utilizado en sistemas automatizados y aquellos que requieren ser controlados. Uno de los puntos fuertes más importantes es que permite la compatibilidad entre los distintos dispositivos y sistemas industriales. Además de proporcionar una manera segura en la que poder realizar intercambio de datos. El flujo de la arquitectura OPC UA es mostrado en la [figura 5.28](#).

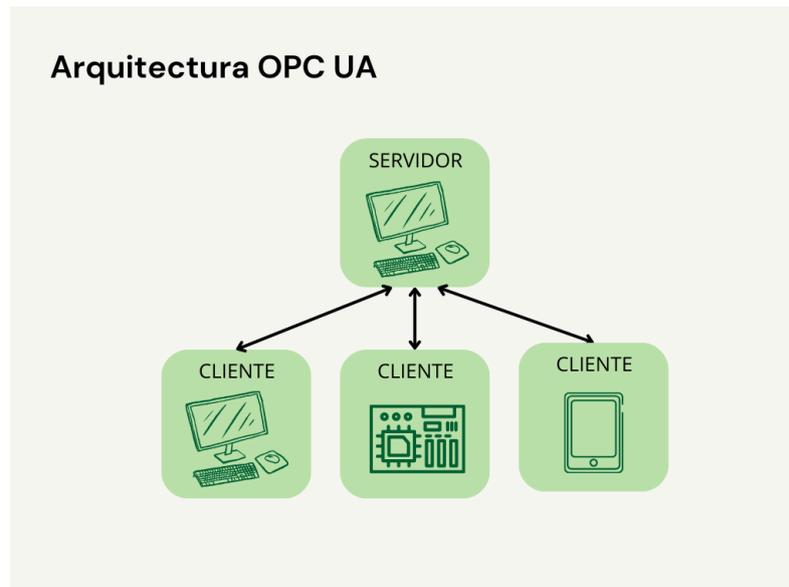


Figura 5.28: Flujo de la arquitectura OPC UA. Fuente: elaboración propia

5.4.2 *Diseño de la arquitectura del sistema*

Una vez sabemos cual es el protocolo de comunicación que vamos a usar para poder desarrollar el proyecto el siguiente paso es establecer las aplicaciones a usar. Además para este proyecto todas las implementaciones se han realizado en el entorno local. La propia programación realizada, las configuraciones y las conexiones realizadas con las tecnologías usadas serán comentadas en apartados posteriores.

En la figura [figura 5.29](#) se observa el diseño de la arquitectura que se ha elaborado para la solución del proyecto. Esta incluye los dispositivos de control y comunicaciones mencionados en [Dispositivos de control y comunicaciones utilizados](#). También incluye parte del diseño de la arquitectura para la comunicación, mencionado en [subsección 5.4.1](#). Y de como ha sido integrado el diseño para la comunicación con las aplicaciones usadas que se encuentran en la [sección 5.5](#).

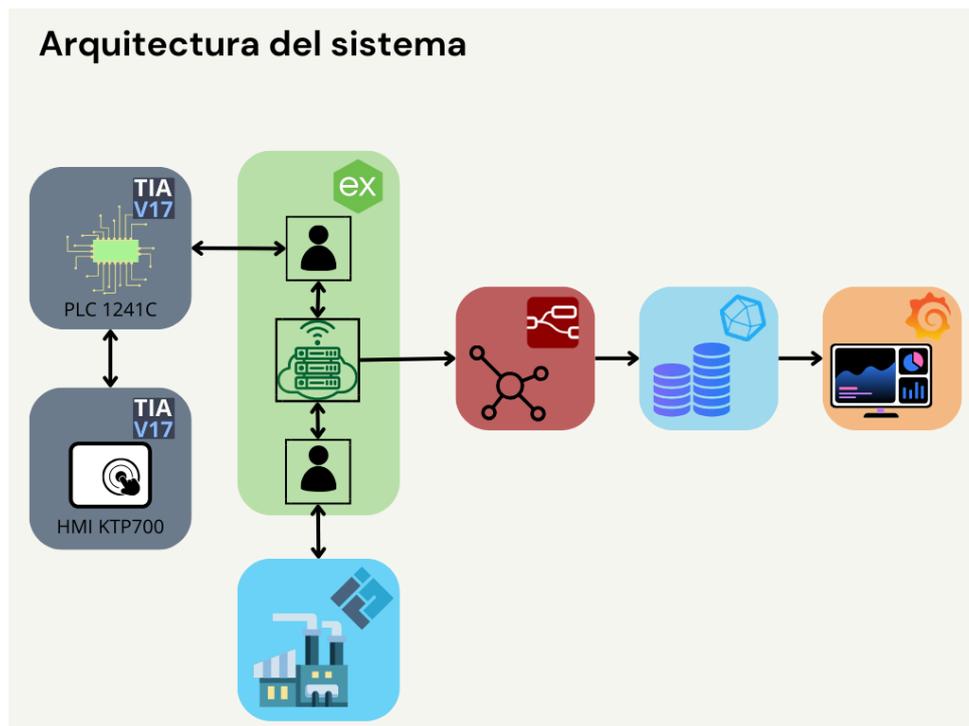


Figura 5.29: Flujo de la arquitectura para el desarrollo del proyecto. Fuente: elaboración propia, con logos de cada fabricante

Las aplicaciones empleadas serán descritas en la [sección 5.5](#).

5.5 Otras aplicaciones utilizadas

En esta sección se mencionarán las aplicaciones que han sido usadas para llevar a cabo el desarrollo del proyecto. Desde la programación del PLC hasta la monitorización final de los datos.

5.5.1 TIA Portal

Para poder realizar la programación del PLC y realizar toda la implementación necesaria del HMI para dotarlo del control, visualización de datos, gestión de alarmas... Es necesario usar un entorno de desarrollo de software que sea compatible con estos. Como ya hemos visto antes usaremos [KEPServerEX](#) y [HMI KTP700 Basic PN](#), ya que son compatibles con el entorno de desarrollo de software TIA Portal V17, del cual vamos a hacer uso.

Este entorno permite programar y configurar varios dispositivos de Siemens, como es el caso. Para llevar esto a cabo emplea distintos lenguajes de programación como Ladder, FBD y SCL. Se hará uso solamente de Ladder y SCL en el proyecto. Mayoritariamente SCL (Structured Control Lenguaje), el cual es un lenguaje de programación de texto estructurado, porque es cómodo la programación que pueda ser compleja. Y haremos uso

de Ladder para aquella programación más simple, ya que en este lenguaje se programa de manera visual y viene perfecto para esto.

La forma en la que se programa en TIA Portal es mediante bloques, en los cuales se les puede especificar los tres lenguajes de programación anteriormente comentados. Estos bloques son:

- **Bloque de organización (OB):** Este tipo de bloque se procesa de manera cíclica. Y pertenecen a un grado superior dentro del programa. Ya que se puede llamar en ellos a otros bloques. Existen varios tipos como el principal (donde se ejecuta la parte principal del programa), de arranque (donde solo se ejecutaría en el arranque del programa, de tiempo... Y dentro de los bloques de organización se pueden establecer prioridades.
- **Bloque de función (FB):** Se tratan de bloques que se programan de manera lógica en las que se almacenan los valores de las variables de manera permanente cuando son instanciados. Además de que una vez ha sido procesado el propio bloque las variables de estos siguen siendo accesibles.
- **Función (FC):** Son bloques de código los cuales no tienen una memoria dedicada. Por lo que no podemos declarar variables nuevas dentro de estos a diferencia de los FB.
- **Bloque de datos (DB):** Estos pueden ser creados de manera directa seleccionándolos. O una vez instanciamos un FB tendremos la opción de crearlos para poder almacenar las variables estáticas de los FB.

Además usaremos TIA Portal para realizar la conexión entre el PLC y el HMI. Donde ambos dispositivos se conectarán mediante sus interfaces PN/IE (PN: PROFINET, IE: Industrial Ethernet). Las cuales hacen uso del estándar de comunicación industrial PROFINET (Dias et al., 2018). PROFINET permite la transmisión de datos en tiempo real y la comunicación de los dispositivos de sistemas industriales. Y por otra parte Industrial Ethernet se trata de la propia adaptación de la tecnología Ethernet convencional a entornos industriales.



Figura 5.30: Logo de TIA Portal V17. Haz clic sobre el logo para acceder a la web del fabricante

5.5.2 *Factory I/O*

Factory I/O (Salcedo Romero de Ávila, 2022) es una plataforma de virtualización industrial completamente interactiva. Se trata de un entorno en 3D donde se permite colocar distintos tipos de maquinaria, actuadores, sensores y objetos. Se hace uso de las escenas, las cuales son el conjunto de configuraciones y elementos que representan el entorno virtual. Dentro de las escenas se permite establecer distintos tipos de configuración a cada uno de los elementos. Como por ejemplo controlar la propia maquinaria de manera analógica o digital.



Figura 5.31: Logo de Factory I/O. Haz clic sobre el logo para acceder a la web del fabricante

5.5.3 *KEPServerEX*

KEPServerEX se trata de una plataforma donde se pueden crear servidores de comunicación industrial. Soporta una gran cantidad de protocolos. Como por ejemplo OPC UA (OLE for Process Control) (“Configuración del servidor OPC UA en KEPserverEX: habilitar acceso de cliente remoto”, 2023), del cual haremos uso. Ya que estamos simulando un entorno industrial. Y como hemos visto en [Diseño de la arquitectura para la comunicación](#) es muy útil cuando tratamos de conectar distintas tecnologías las cuales tienen que interactuar entre ellas en tiempo real.

Ofrece distintos mecanismos de seguridad, aunque para este proyecto no se ha empleado ninguno en KEPServerEX. Y otra de sus características es que nos permite ver el registro de los datos a tiempo real en el propio servidor para saber en todo momento lo que está ocurriendo.

KEPServerEX lo usaremos como servidor. Para poder conectar el [KEPServerEX](#) con [Factory I/O](#). De esta manera con nuestro PLC seremos capaz de tomar el control de toda la maquinaria que se encuentra en la escena de Factory I/O.



Figura 5.32: Logo de KEPServerEX. Haz clic sobre el logo para acceder a la web del fabricante

5.5.4 *Node-RED*

Node-RED (Lekić & Gardašević, 2018) se trata de una plataforma open source en la que se programa visualmente mediante distintos tipos de nodos estableciendo flujos sobre estos, aunque también podemos escribir nuestros propios scripts si lo deseamos. Trata de conectar aplicaciones, servicios y dispositivos de forma simple y cómoda. Es popular en sistemas de IoT ya que ofrece una gran cantidad de protocolos y servicios, hay una gran cantidad de bibliotecas aportadas por la comunidad, visualización de datos...

Para poder usar Node-Red lo primero que debemos hacer es instalar Node.js. Node.js, de manera breve se trata de un entorno de tiempo de ejecución en cual nos servirá para ejecutar Node-RED.

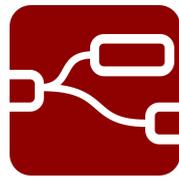


Figura 5.33: Logo de Node-RED. Haz clic sobre el logo para acceder a la web del fabricante

5.5.5 *InfluxDB*

InfluxDB (Naqvi et al., 2017) se trata de una base de datos open source la cual trabaja con series temporales. Su principal fuerte es que se pueden tanto almacenar, como consultar datos que cambian con el flujo del tiempo. Por lo que para nuestra monitorización sobre los contadores que tendremos y los temporizadores nos viene perfecto. Además de lo anteriormente comentado su velocidad de escritura/lectura está optimizada para ser elevada, lo que convierte a InfluxDB en una base de datos muy útil para obtener datos sobre sensores y actuadores. También es capaz de manejar una gran cantidad de volúmenes de datos. Y es soportada por numerosas plataformas que hacen uso de las fuentes de datos.



Figura 5.34: Logo de InfluxDB. Haz clic sobre el logo para acceder a la web del fabricante

5.5.6 Grafana

Grafana (Chakraborty & Kundan, 2021) se trata de una de las plataformas de monitorización open source más populares hoy en día. Con Grafana se pueden observar datos en tiempo real y visualizarlos mediante distintos tipos de paneles preestablecidos o personalizados. Además, Grafana para mostrar los datos necesita conectarse a una fuente de datos, como pueden ser una base de datos, web APIs, distintos sistemas de monitoreo, etc. . .



Figura 5.35: Logo de Grafana. Haz clic sobre el logo para acceder a la web del fabricante

6 Descripción del desarrollo

Para poder elaborar el desarrollo se ha tenido que hacer primero una descripción de la solución que vamos a adoptar, visto en la [Capítulo 5](#). Donde se menciona los distintos dispositivos y tecnologías que vamos a emplear para llevar a cabo el desarrollo del proyecto. Y donde además la arquitectura de comunicaciones ([sección 5.4](#)) nos indica la comunicación de los distintos dispositivos y aplicaciones.

6.1 Desarrollo del programa que automatiza el proceso

Anteriormente se ha comentado que es [TIA Portal](#) y la forma en la que ese programa. En base a esto haremos nuestro programa en el cual necesitaremos cierta organización.

6.1.1 *Distribución de las variables en memoria*

Lo primero de todo es conocer bien la distribución de las variables en la memoria del PLC ya que podría haber solapes entre estas. Ya que si dos variables solapasen las mismas posiciones de memoria se produciría un comportamiento indeseado e inconsistente del programa que no podríamos solucionar durante la ejecución de este. También hay que tener en cuenta que cada tipo de datos ocupa una cantidad de posiciones en memoria distinta así que esto también es necesario tenerlo en cuenta.

Estas variables hay que darlas de alta en los PLC Tags de TIA Portal. Además es necesario especificar las variables en memoria porque estamos realizando una virtualización y tendremos que enviárselas a nuestro servidor OPC [Logo de KEPServerEX](#), para que este le envíe la información a Factory I/O y los actuadores y sensores funcionen. Lo mismo ocurre con los datos que se van a monitorizar.

La distribución escogida para las variables ha sido la siguiente:

- **Entradas booleanas:** Entre las direcciones 100 - 200
- **Salidas booleanas:** Entre las direcciones 200 - 300

- **Entradas numéricas:** Entre las direcciones 300 - 400
- **Salidas numéricas:** Entre las direcciones 400 - 500
- **Variables del HMI en el PLC:** Entre las direcciones 500 - 600
- **Alarmas:** Desde la dirección 600 en adelante

PLC tags					
		Name	Tag table	Data type	Address ▲
93		TURN_ADEL	Default tag table	Bool	%M213.0
94		TURN	Default tag table	Bool	%M213.5
95		TURN_ATRAS	Default tag table	Bool	%M213.7

Figura 6.1: Ejemplo de la organización de las variables booleanas de salida. - PLC Tags TIA Portal

6.1.2 Bloques de organización principales del programa

Para la ejecución de nuestro programa será necesaria la implementación de dos bloques de organización (OBs). Estos son los siguientes:

- **Startup:** Bloque de organización que se ejecuta antes que cualquier otro. Muy útil si queremos asegurarnos de que no hay ningún actuador activo desde el inicio del programa. Solo se ejecuta al comienzo del programa una sola vez
- **Main:** Es un bloque de organización que se ejecuta de manera cíclica mientras que el programa esté activo

Bloque de organización 'Startup'

Dentro del OB de 'Startup' realizado en SCL, tendremos los distintos FCs que hacen que las distintas partes solo tengan activo su estado inicial. Y también todos los contadores para que estos sean inicializados con un valor de '0'.

```

// RESETEA TODAS LAS MÁQUINAS
%FCs para solo tener el estado inicial activo
"Reset_cina"();
"Reset_turntable"();
"Reset_maq_peq"();
"Reset_maq_gra"();
"Reset_maq_entr"();
"Reset_sens_palet_peq"();
"Reset_sens_palet_gra"();
"Reset_proces"();
"Reset_fin_peq"();
"Reset_fin_gra"();

// RESETEA LOS CONTADORES PARA QUE INICIALMENTE ESTÉN A 0
"PANEL_N_MAN" := 0;

```

```
"PANEL_N_MAR" := 0;
"PANEL_N_PAR" := 0;

"FIN_NPIEZAS" := 0;
"FIN_NPIEZAS_PG" := 0;

"RAW_N_PALLET" := 0;
"PG_N_PALLET" := 0;

"FIN_NENTRE_PG" := 0;
"FIN_NENTRE" := 0;
```

Listing 6.1: Código del OB de Startap

A continuación se muestra un pequeño ejemplo de lo que hace uno de estos FCs internamente.

```
% ESTADO INICIAL ACTIVO
"TURNTABLE_DB".X20 := TRUE;

% RESTO DE ESTADOS Y ALARMAS DESACTIVADOS
"TURNTABLE_DB".X21 := FALSE;
"TURNTABLE_DB".X22 := FALSE;
"TURNTABLE_DB".X23 := FALSE;
"TURNTABLE_DB".X24 := FALSE;
"TURNTABLE_DB".X25 := FALSE;
"TURNTABLE_DB".X26 := FALSE;
"TURNTABLE_DB".X27 := FALSE;

"AlarmasHMI".turn := FALSE;
"TURNTABLE_DB".alarma := FALSE;
```

Listing 6.2: Código del FC que resetea los estados del Turntable - Reset_turntable

Bloque de organización 'Main'

Este bloque de organización ya viene dado por TIA Portal, no es necesario añadirlo. Su programación inicial es en Ladder. Pero podemos añadirle los distintos FBs Y FCs que pueden estar programados en SCL y se ejecutarán sin problema.

El desarrollo de este bloque de función es demasiado extenso como para explicarlo en este subapartado. La explicación estará en la [subsección 6.1.3](#).

6.1.3 Desarrollo 'Main'

En este apartado se explicará de manera extensa el propio desarrollo del bloque de función 'Main'. El bloque de organización 'Main' ha sido dividido en distintas redes. Las cuales son las siguientes:

Selección de modos en el panel físico y el HMI

En esta red se ha desarrollado la selección de los modos de funcionamiento por parte del panel y el HMI. Se ha usado el mismo bloque de función (FB) ya que la programación de la selección del modo de estos dos es exactamente igual. Este FB se llama 'BOTONES_PANEL' Pero cada uno tiene su bloque de datos (DB) ya son dispositivos que han de trabajar por separado y no se tienen que solapar. En la [figura 6.2](#) se muestra la red con las entradas y salidas de sus respectivos FBs.

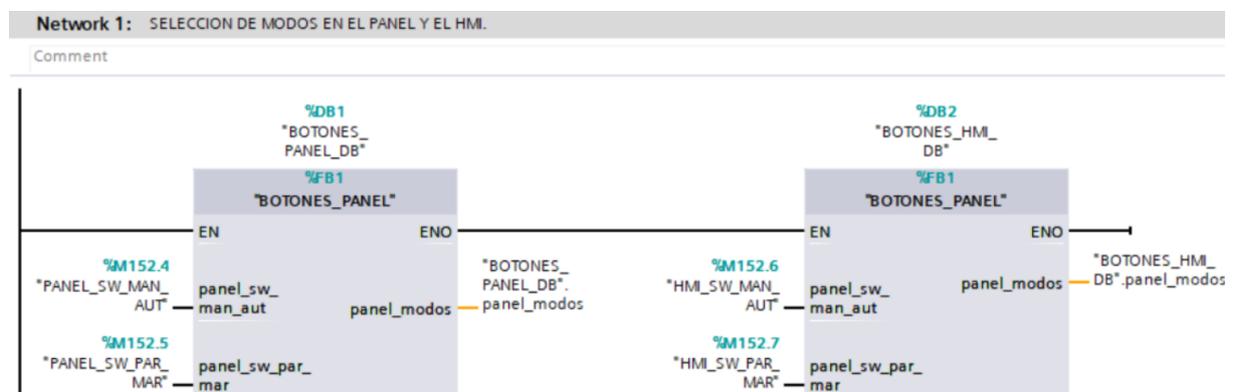


Figura 6.2: Bloques de función de selección del modo de funcionamiento del panel y HMI

Las variables de entrada del panel son los propios interruptores que se encuentran en el panel para el cambio de modo de funcionamiento. Y las variables de entrada del HMI son los interruptores que tiene este para cambiar los modos.

```
IF #panel_sw_man_aut = FALSE AND #panel_sw_par_mar = FALSE THEN
    #panel_modos.manual := TRUE;
    #panel_modos.automatico_paro := FALSE;
    #panel_modos.automatico_marcha := FALSE;
ELSIF #panel_sw_man_aut AND #panel_sw_par_mar = FALSE THEN
    #panel_modos.manual := FALSE;
    #panel_modos.automatico_paro := TRUE;
    #panel_modos.automatico_marcha := FALSE;
ELSIF #panel_sw_man_aut AND #panel_sw_par_mar THEN
    #panel_modos.manual := FALSE;
    #panel_modos.automatico_paro := FALSE;
    #panel_modos.automatico_marcha := TRUE;
END_IF;
```

Listing 6.3: Código FB 'BOTONES_PANEL'

Las variables de salida son las internas de los bloques de datos (DB) que tiene como tipo de datos específico creado para que guarde el modo de funcionamiento. Esta variable se llama 'panel_modos'. Y contiene los valores para saber que modo está activo en cada dispositivo.

Name	Data type
▼ Output	
▼ panel_modos	"User_data_type_1"
stop	Bool
manual	Bool
automatico_paro	Bool
automatico_marcha	Bool

Figura 6.3: Tipo de datos creado para la selección de los modos

Resolución de los modos entre el panel y el HMI

Como ya se ha comentado el panel es el dispositivo principal que debe ser capaz de controlar los modos de funcionamiento de la línea de producción. Y el HMI para este caso quedaría en un segundo plano. Es por eso que en el panel tenemos que indicar si vamos a usar el HMI para que realice en conjunto el cambio de modo de funcionamiento. Ya que el HMI no debe de realizarlo en solitario. Así que el HMI se debe de sincronizar con el HMI estando los dos en el mismo modo de funcionamiento. La red resultante es mostrada en la [figura 6.4](#).

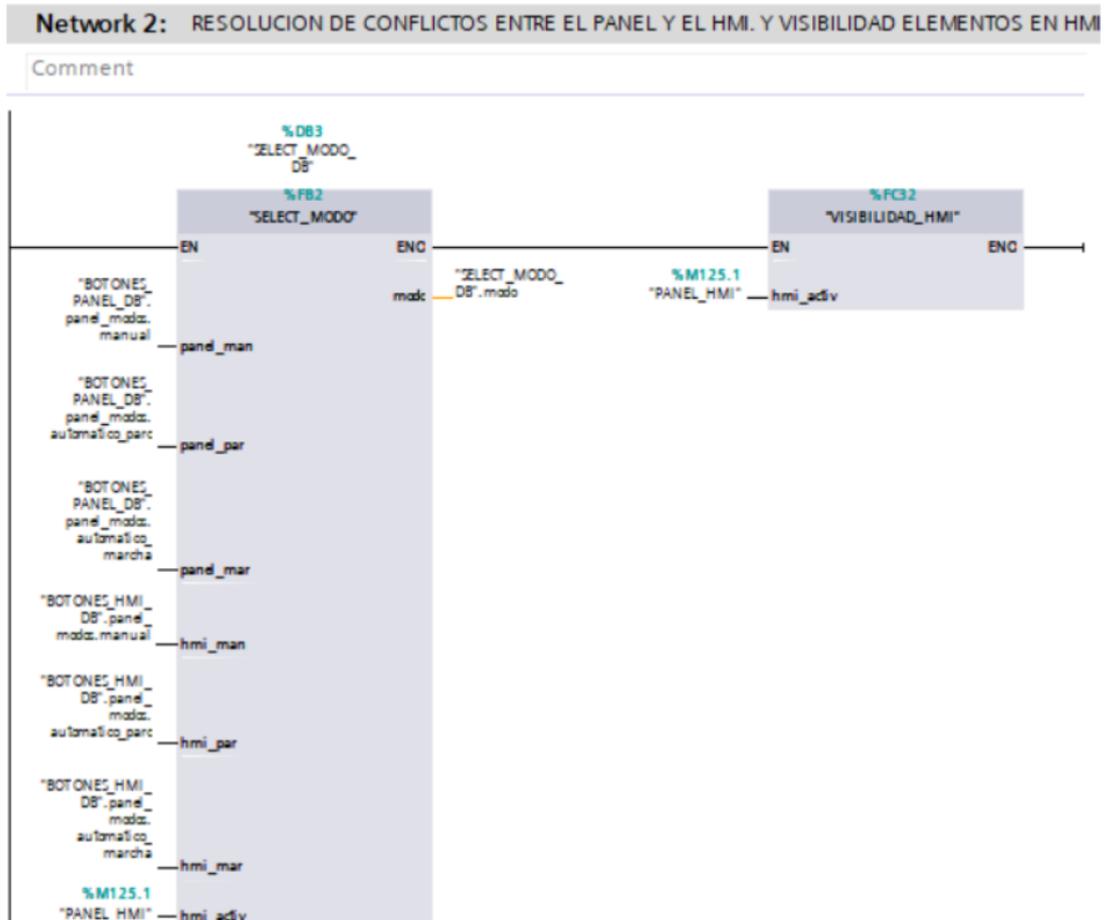


Figura 6.4: Red para la resolución de conflictos entre panel y HMI

En el FB 'SELECT_MODO' es donde se resolverá los conflictos entre los dos dispositivos. Y saldrá finalmente el modo en el que va a entrar la línea de producción.

```

IF (#panel_man AND #hmi_man AND #hmi_activ)
  OR (#panel_man AND NOT #hmi_activ) THEN
  #modo.manual := TRUE;
  #modo.automatico_paro := FALSE;
  #modo.automatico_marcha := FALSE;
ELSIF (#panel_par AND #hmi_par AND #hmi_activ)
  OR (#panel_par AND NOT #hmi_activ) THEN
  #modo.manual := FALSE;
  #modo.automatico_paro := TRUE;
  #modo.automatico_marcha := FALSE;
ELSIF (#panel_mar AND #hmi_mar AND #hmi_activ)
  OR (#panel_mar AND NOT #hmi_activ) THEN
  #modo.manual := FALSE;
  #modo.automatico_paro := FALSE;
  #modo.automatico_marcha := TRUE;
END_IF;

```

Listing 6.4: Código FB 'SELECT_MODO'

Y en el FB 'VISIBILIDAD_HMI' la función que hace es más secundaria. Ya que hace que se muestren iconos en el HMI de las partes de la línea de producción para saber que esa parte en concreto está activa. Así que no se entrará en detalle.

Generación de alarmas y cambio de modo

En esta red formada por dos FB. La FB 'ALARMAS' será la que active la alarma de emergencia general si la seta de emergencia ha sido pulsada. La seta de emergencia funciona con lógica negativa. Es por eso que se le ha pasado un contacto cerrado en la entrada, para que cuando la seta de emergencia sea pulsada se active la emergencia general y la luz correspondiente. Además de esto lo que también hace es enviar la alarma que haya saltado al HMI para que luego este muestre su propia alerta para saber cual ha sido el problema que la ha ocasionado.

Y el FB 'MODO_DE_TRABAJO' realiza la función del cambio de modo gracias a la resolución de conflictos entre en panel de mandos y el HMI que se ha realizado antes. Además de cambiar de modo se activarían los leds correspondientes en el panel que muestren que modo está activo. Y también contienen contadores para saber cuantas veces se ha cambiado a cada modo y mostrarlos en la futura monitorización con GRAFANA. Para desarrollar el FB 'MODO_DE_TRABAJO' se ha realizado un GRAFCET (figura 6.6) previo para que la programación en SCL del FB no sea tan costosa.

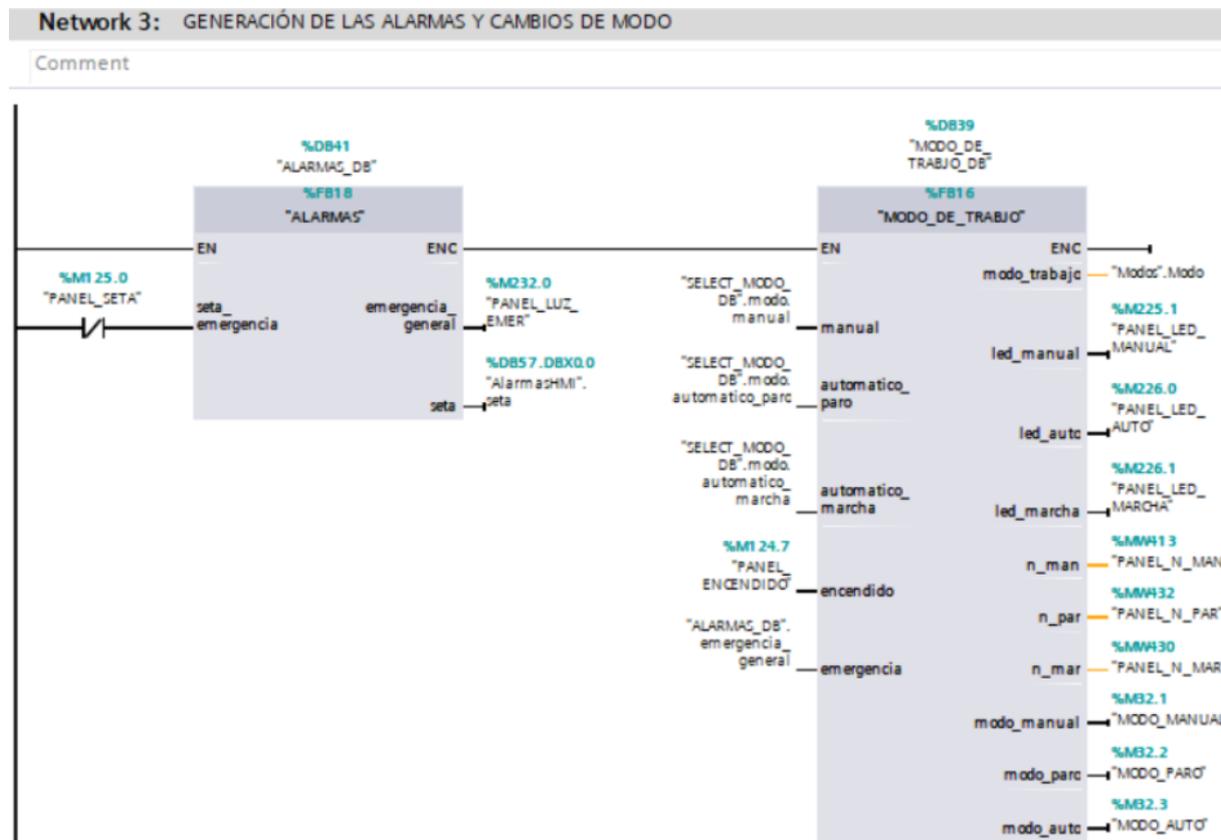


Figura 6.5: Red para la resolución de conflictos entre panel y HMI

```

#emergencia_general := #seta_emergencia;

#seta := #seta_emergencia;

"EMER_CINA" := "AlarmasHMI".cin_alim_seta;
"EMER_TURN" := "AlarmasHMI".turn;
"EMER_MAQPEQ" := "AlarmasHMI".maqi_tarda;
"EMER_MAQGRA" := "AlarmasHMI".maqi2_tarda;
"EMER_MAQENT" := "AlarmasHMI".maqe_tarda;
"EMER_PALPEQ" := "AlarmasHMI".maqi_pallet_atasc;
"EMER_PALGRA" := "AlarmasHMI".maqi2_pallet_atasc;
"EMER_PROCES" := "AlarmasHMI".maq_proces_seta;
"EMER_FINPEQ" := "AlarmasHMI".fin_peq;
"EMER_FINGRA" := "AlarmasHMI".fin_pg;

```

Listing 6.5: Código FB 'ALARMAS'

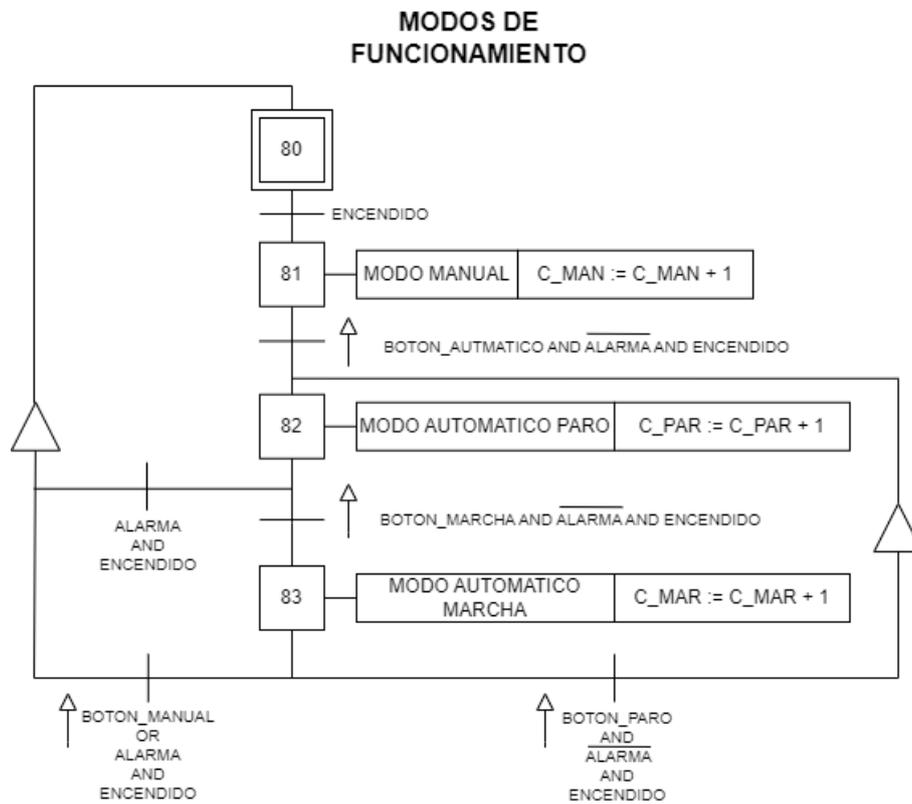


Figura 6.6: GRAFCET de los modos de funcionamiento/trabajo

```

%// contadores de los cambios de estados que se han realizado
#c_man(CU := #X81,
       PV := 0,
       CV => #n_man);

#c_par(CU := #X82,
       PV := 0,
       CV => #n_par);

```

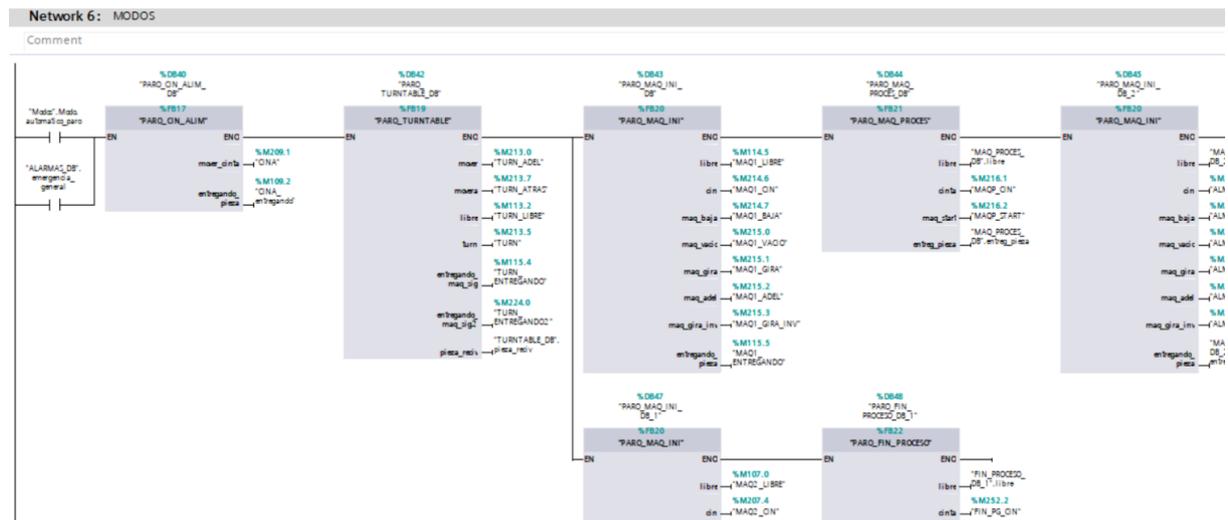



Figura 6.7: Red modos automático paro y manual, parte del modo automático paro



Figura 6.8: Red modos automático paro y manual, parte restante del modo automático paro

Solo se mostrará el código en SCL del FB 'PARO_TURNTABLE'. El resto de códigos de FBs no se mostrarán porque la lógica del código es exactamente igual, pero las variables cambien en función de la parte de la línea de producción. Las salidas de las FBs serán los propios actuadores de las máquinas.

```
#mover := FALSE;
#movera := false;
%//#libre := FALSE;
#turn := false;
%//#entregando_maq_sig := false;
```


En el siguiente código se mostrará como ejemplo el FB 'MANUAL_TURN_TABLE':

```
///  
// El boton del HMI activa el actuador que mueve los rodillos  
#mover := #b_mover;  
  
#movera := #b_movera;  
///  
//#libre := #b_libre;  
#turn := #b_turn;  
///  
//#entregando_maq_sig := #b_entregando_maq_sig;  
///  
//#entregando_maq_sig2 := #b_entregando_maq_sig2;  
///  
//#pieza_reciv := #b_pieza_reciv;
```

Listing 6.8: Código FB 'MANUAL_TURN_TABLE'

Modo automático marcha

Para poder desarrollar esta red en el bloque de organización Main se han tenido que seguir los siguientes pasos:

- Creación de los diagramas GRAFCETs para cada parte de la línea
- Creación de las ecuaciones de estado para cada parte de la línea
- Desarrollar los correspondientes bloques de función en lenguaje SCL en base a las ecuaciones de estado
- Vincular las variables en memoria para las entradas y salidas de los bloques de función
- Condiciones de entrada para activar todos los bloques de función. Estas condiciones se tratan de que el modo automático haya sido activado y que no haya saltado la alarma de emergencia general.
- Condiciones individuales previas a activar los FBs en base a las alarmas. Estas condiciones se tratan de las propias alarmas individuales de cada parte de la línea. Es decir si salta una alarma en una parte esa parte será parada. Pero el resto de la línea continuará en modo automático marcha sin ningún tipo de impedimento.

El desarrollo de los GRAFCETs ha seguido las descripciones dadas en [Partes en las que se ha dividido la línea de producción](#) para cada parte.

De forma inicial, para realizar este diseño se pensó en hacer un total de diez GRAFCETs, uno por cada parte implicada en la línea de producción. Pero conforme se fueron desarrollando se observaba que algunos se podrían reutilizar para distintas máquinas ya que los procesos que hacían eran muy similares o iguales. Así que esto nos ahorraría desarrollar un total de diez GRAFCETs y simplificar todo este diseño en un total de seis diagramas GRAFCET. Por otra parte esto también ahorrará bastante tiempo en el desarrollo de la obtención de las ecuaciones lógicas y en la programación en TIA Portal del automatismo.

Estos GRAFCETs lo que pretenden es que cada parte de la línea de producción sea capaz de seguir una secuencia de instrucciones de manera ordenada. Y cada parte sea capaz de comunicarse con el resto mediante las variables de condición de los GRAFCET como la de 'ACTIVACION' que se encuentra en la mayoría de estos y las acciones llamadas generalmente 'ENTREGA'. Los GRAFCETs desarrollados serán mostrados en esta misma subsección.

CINTA DE ALIMENTACIÓN

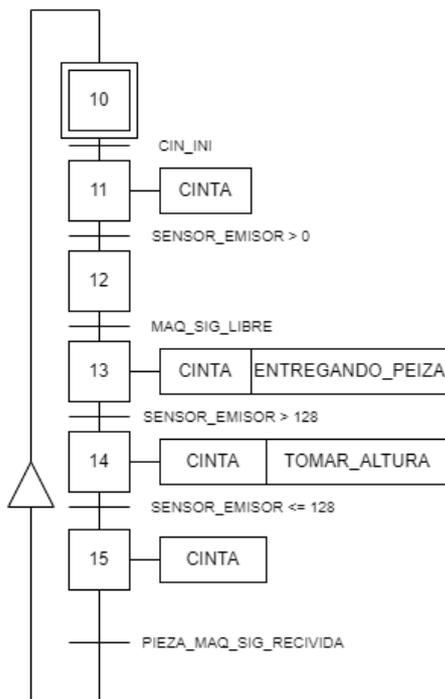


Figura 6.11: GRAFCET de la Cinta de alimentación

**MÁQUINA DE INICIO 1 Y 2
Y MÁQUINA DE
ENTREGA DE BASES**

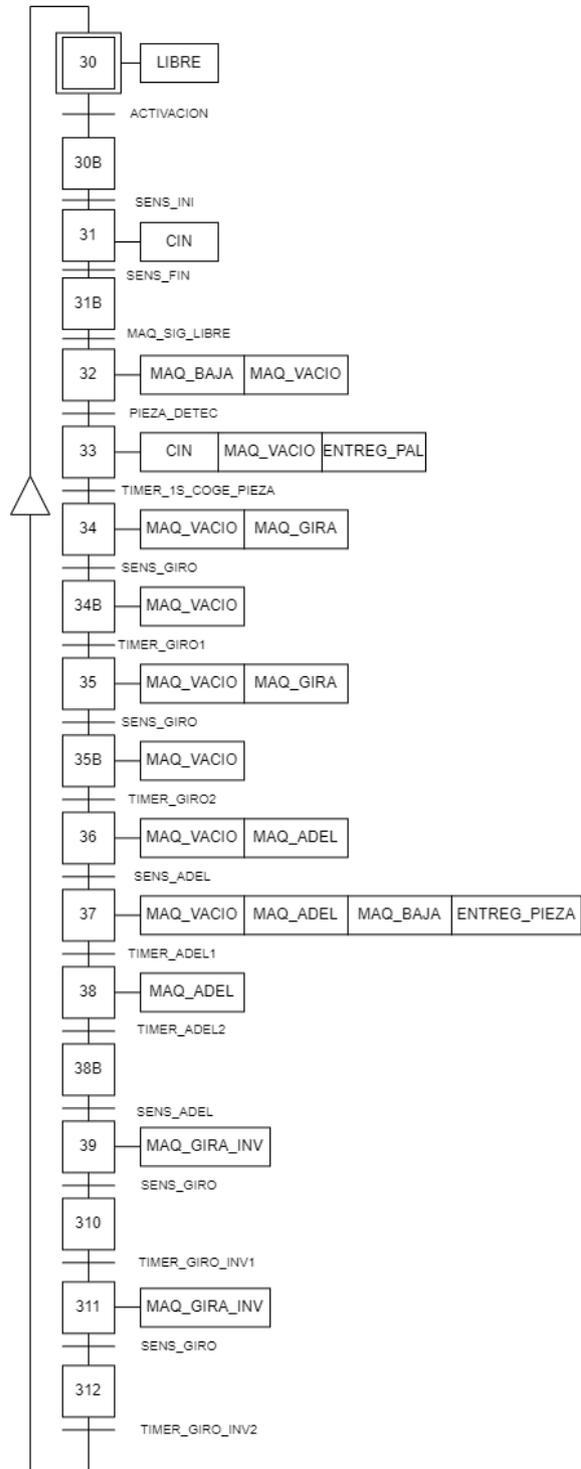


Figura 6.12: GRAFCET de Máquina de inicio 1, Máquina de inicio 2 y Máquina de entrega de bases

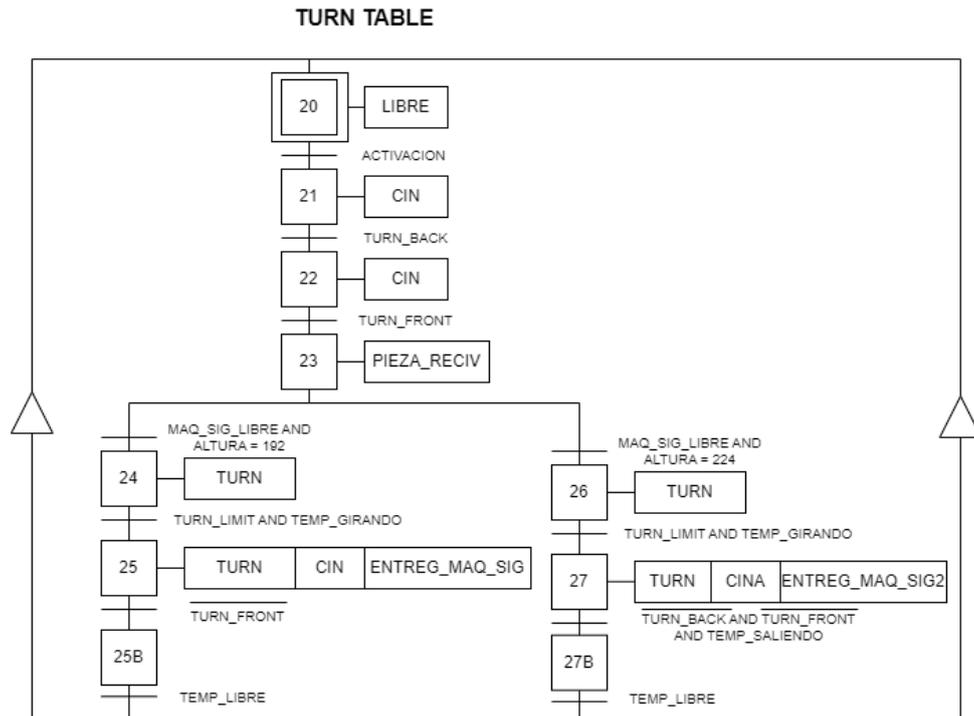


Figura 6.13: GRAFCET del Turntable o plataforma giratoria

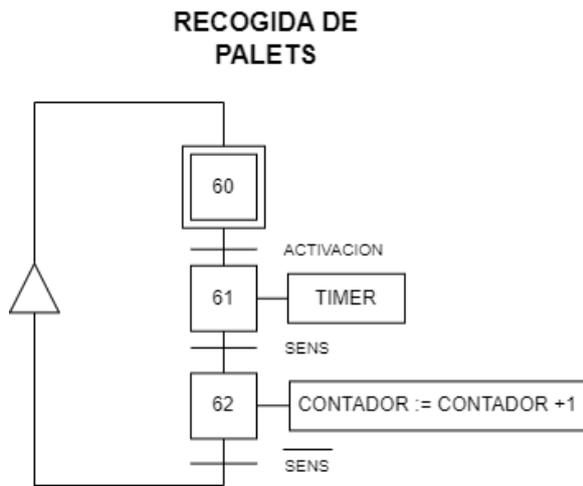


Figura 6.14: GRAFCET de Recogida de palets 1 y Recogida de palets 2

MÁQUINA DE PROCESADO

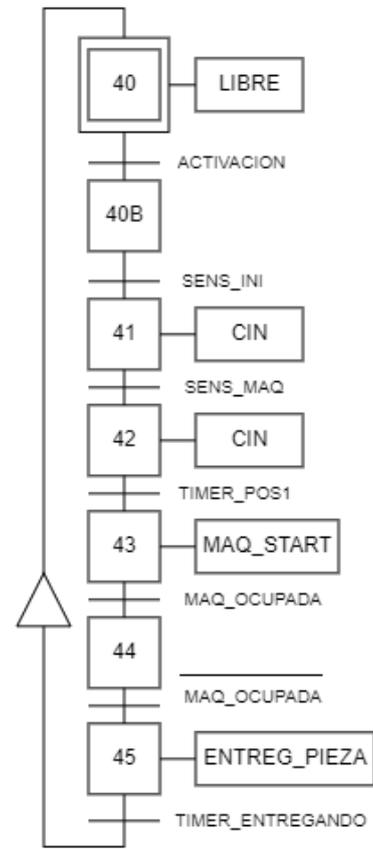


Figura 6.15: GRAFCET de Máquina de procesado

MÁQUINA DE FIN DE PROCESO 1 Y 2

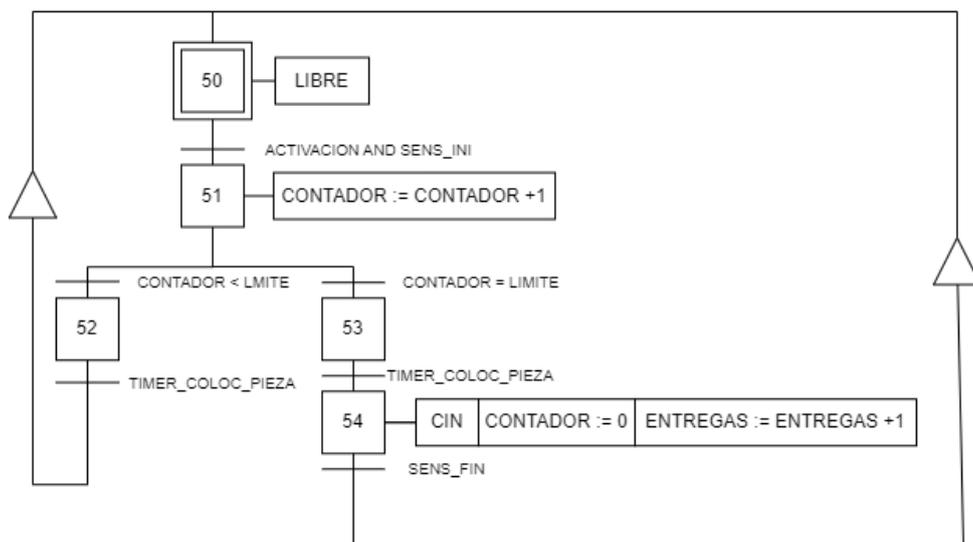


Figura 6.16: GRAFCET de Máquina de fin de proceso 1 y Máquina de fin de proceso 2

Para el desarrollo del modo automático marcha se han desarrollado las ecuaciones lógicas correspondientes a los propios GRAFCETs. Pero solo se mostrará un ejemplo de éstas porque hay algunas que son muy extensas. En este ejemplo se mostrarán las respectivas ecuaciones lógicas para la implementación de la parte de la [Cinta de alimentación](#) en su modo automático marcha.

Tabla 6.1: Ecuaciones de estado activo de la [Cinta de alimentación](#)

ECUACIONES DE ESTADO ACTIVO

$$X_{10} = S_{10} + X_{10} \cdot \overline{S_{11}}$$

$$X_{11} = S_{11} + X_{11} \cdot \overline{S_{12}}$$

$$X_{12} = S_{12} + X_{12} \cdot \overline{S_{13}}$$

$$X_{13} = S_{13} + X_{13} \cdot \overline{S_{14}}$$

$$X_{14} = S_{14} + X_{14} \cdot \overline{S_{15}}$$

$$X_{15} = S_{15} + X_{15} \cdot \overline{S_{10}}$$

Tabla 6.2: Ecuaciones de set de la [Cinta de alimentación](#)

ECUACIONES DE SET

$$S_{10} = FirstScan + X_{15} \cdot PIEZA_MAQ_SIG_RECIB$$

$$S_{11} = X_{10} \cdot CIN_INI$$

$$S_{12} = X_{11} \cdot SENSOR_EMISOR > 0$$

$$S_{13} = X_{12} \cdot MAQUINA_SIG_LIBRE$$

$$S_{14} = X_{13} \cdot SENSOR_EMISOR > 128$$

$$S_{15} = X_{14} \cdot SENSOR_EMISOR \leq 128$$

Tabla 6.3: Ecuaciones de salida de la [Cinta de alimentación](#)

ECUACIONES DE SALIDA

$$cinta = X_{11} + X_{13} + X_{14} + X_{14}$$

$$ENTREGANDO_PIEZA = X_{13}$$

$$TOMAR_ALTURA = X_{14}$$

El código en SCL correspondiente a las ecuaciones lógicas mostradas anteriormente de la cinta de alimentación se ha desarrollado en un bloque de función. Al igual que en el

resto de modos se ha empleado FBs aquí exactamente igual para cada parte de la línea indexada.

```
%%ECUACIONES DE SET
#S10 := "FirstScan" OR (#X15 AND #pieza_maq_sig_reciv);
#S11 := #X10 AND #cina_ini;
#S12 := #X11 AND #sensor_emisor > 0;
#S13 := #X12 AND #maq_sig_libre;
#S14 := #X13 AND #sensor_emisor > 128;
#S15 := #X14 AND #sensor_emisor <= 128;

%%ECUACIONES DE ESTADO ACTIVO
#X10 := #S10 OR #X10 AND NOT #S11;
#X11 := #S11 OR #X11 AND NOT #S12;
#X12 := #S12 OR #X12 AND NOT #S13;
#X13 := #S13 OR #X13 AND NOT #S14;
#X14 := #S14 OR #X14 AND NOT #S15;
#X15 := #S15 OR #X15 AND NOT #S10;

%%ECUACIONES DE SALIDA
#mover_cinta := #X11 OR #X13 OR #X14 OR #X15;
#tomar_altura := #X14;
#entregando_pieza := #X13;

IF #tomar_altura = TRUE THEN
    #altura := #sensor_emisor;
END_IF;

IF NOT #seta THEN
    #mover_cinta := FALSE;
    #alarma := TRUE;
ELSE
    #alarma := FALSE;
END_IF;
```

Listing 6.9: Código FB 'CIN_ALIM' para el modo marcha automático

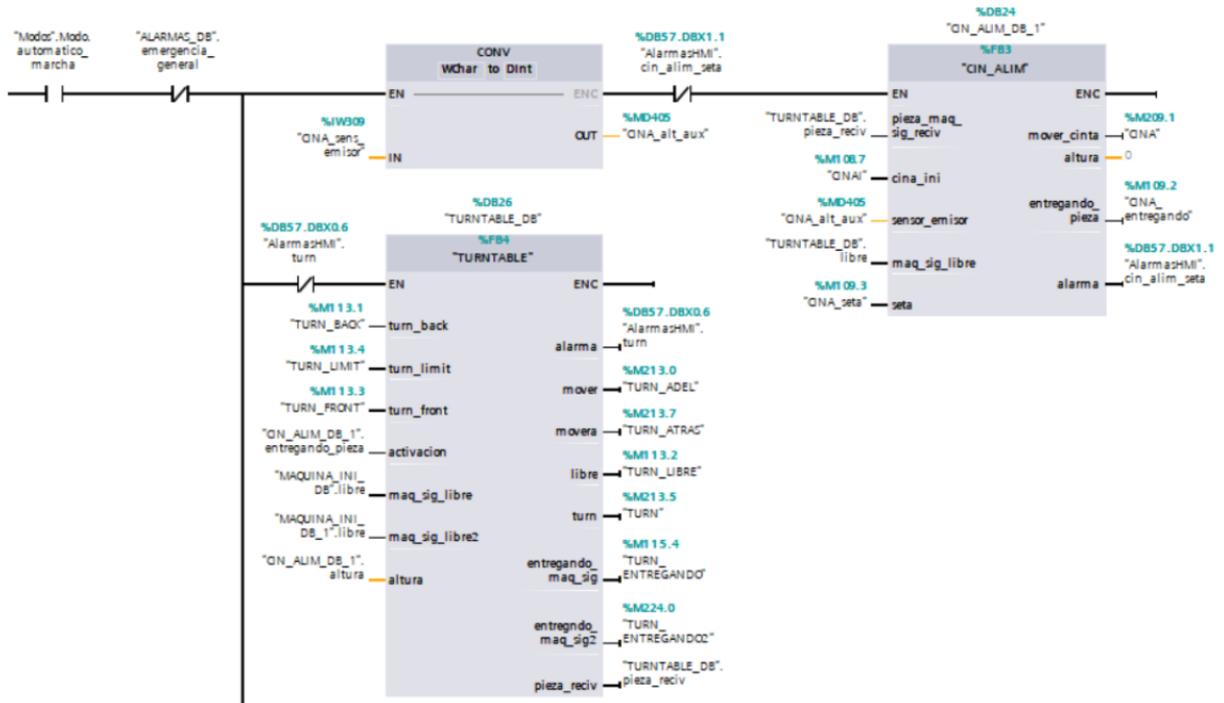


Figura 6.17: FBs Cinta de alimentación y Turntable o plataforma giratoria modo automático y convertor de datos de la matriz de luces

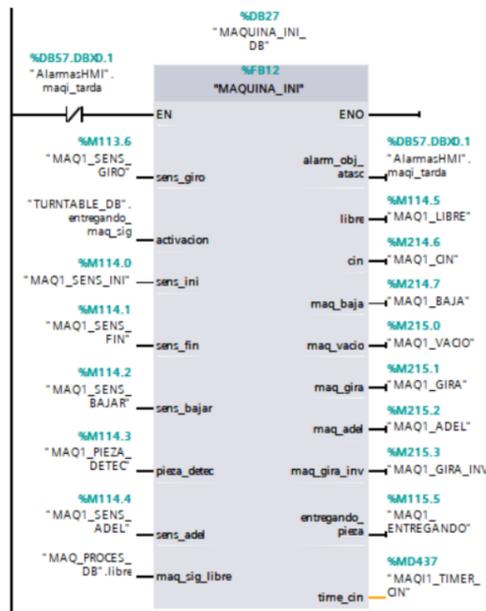


Figura 6.18: FB Máquina de inicio 1 modo automático

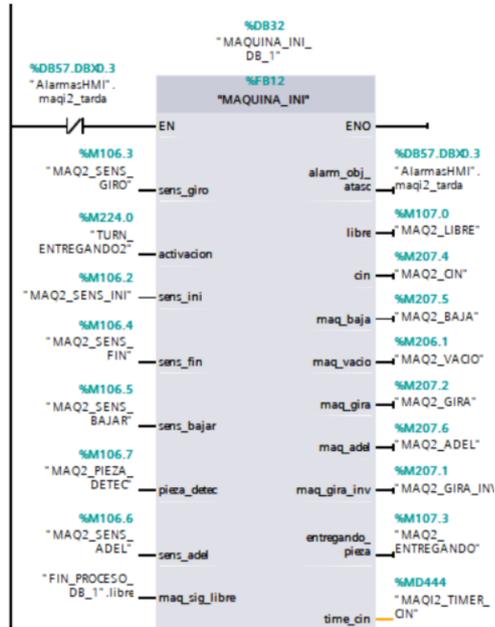


Figura 6.19: FB Máquina de inicio 2 modo automático

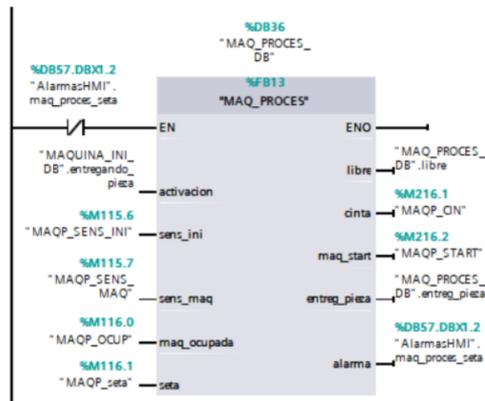


Figura 6.20: FB Máquina de procesado modo automático

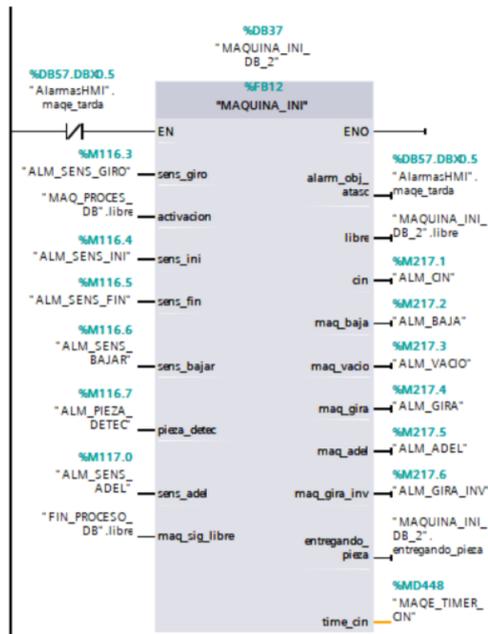


Figura 6.21: FB Máquina de entrega de bases modo automático

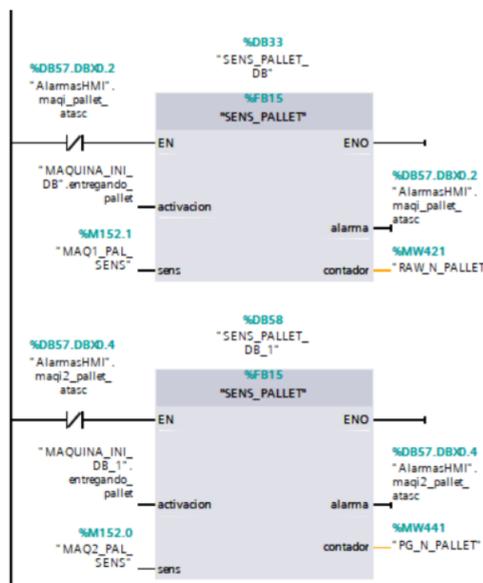


Figura 6.22: FBs Recogida de palets 1 y Recogida de palets 2 modo automático

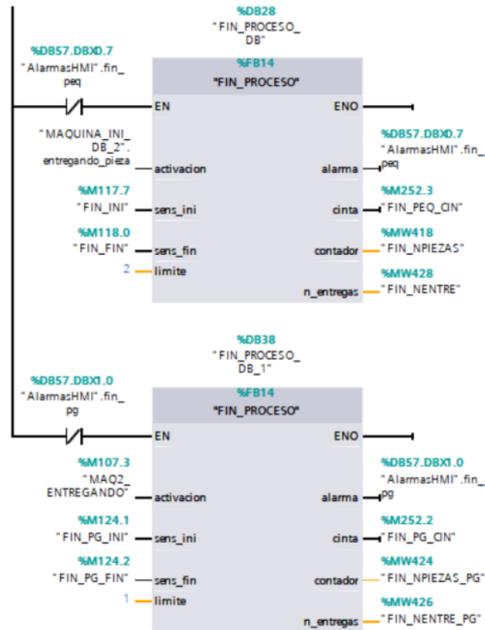


Figura 6.23: FBs Máquina de fin de proceso 1 y Máquina de fin de proceso 2 modo automático

Control de alarmas

La red de control de alarmas como su propio nombre indica trata sobre la gestión que se hace de todas las alarmas. Las alarmas han sido definidas en la [sección 5.2](#).

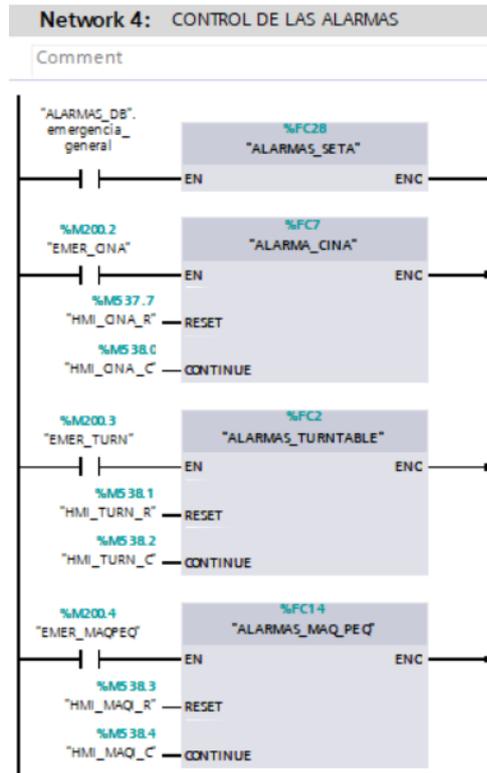


Figura 6.24: FCs que gestionan las alarmas, parte 1

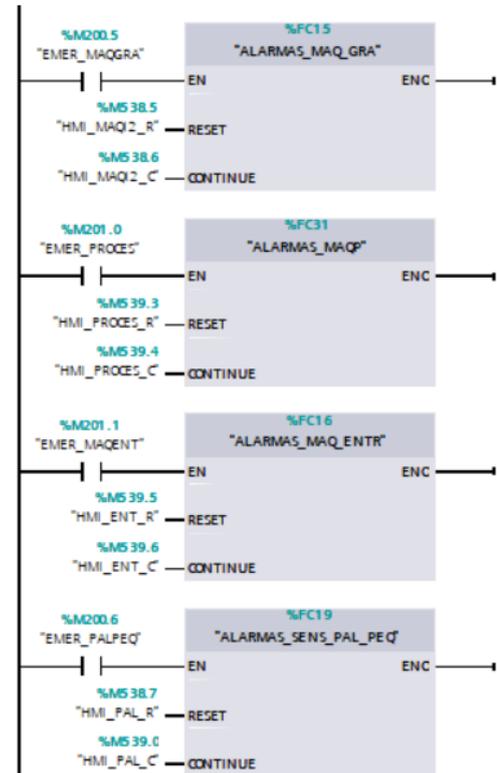


Figura 6.25: FCs que gestionan las alarmas, parte 2

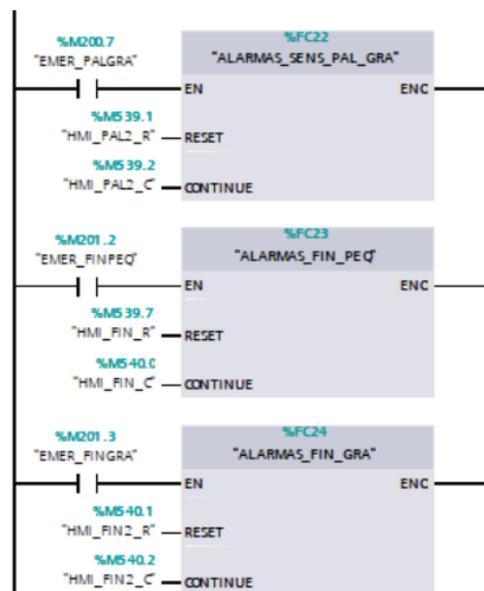


Figura 6.26: FCs que gestionan las alarmas, parte 3

En las figuras figura 6.24, figura 6.25 y figura 6.26 anteriores vemos los distintos FCs que se han creado para tratar las alarmas. En este caso se han decidido usar funciones en vez de bloque de función porque no haremos uso de las variables estáticas. La tarea que se va a hacer dentro del FC 'ALARMAS_SETA' son reiniciar todos los estados de

la línea de producción. Por otra parte el resto de FCs. Además de ser activadas por su correspondiente alarma tienen como entradas los botones de 'RESET' y 'CONTINUE' del HMI correspondientes con su parte.

Lo que se hace en código del FC 'ALARMA_CINA', es que según el botón que sea pulsado se llamará a su función correspondiente.

```
// FLANCOS ASCENDENTES DE LOS BOTONES
"CINA_CONTROL_RESET"(CLK:=#RESET);
"CINA_CONTROL_CONTI"(CLK:=#CONTINUE);

IF "CINA_CONTROL_RESET".Q THEN
  "Reset_cina"();
END_IF;

IF "CINA_CONTROL_CONTI".Q THEN
  "Continue_cina"();
END_IF;
```

Listing 6.10: Código FC 'ALARMA_CINA' para el control de la resolución de la alarma de la [Cinta de alimentación](#)

Y finalmente en este último código en SCL del FC 'Reset_cina' estamos modificando los valores asociados al bloque de datos del modo automático de la línea de alimentación. Se están negando todos los estados y las alarmas. Salvo el primer estado que quedaría en activo ya que estamos reiniciando la máquina.

```
//ACTIVACION DE LOS ESTADOS INICIALES
//DEL MODO AUTOMÁTICO MARCHA
"CIN_ALIM_DB_1".X10 := TRUE;

// RESTO DE ESTADOS EN FALSE
//
"CIN_ALIM_DB_1".X11 := FALSE;
"CIN_ALIM_DB_1".X12 := FALSE;
"CIN_ALIM_DB_1".X13 := FALSE;
"CIN_ALIM_DB_1".X14 := FALSE;
"CIN_ALIM_DB_1".X15 := FALSE;

// ALARMAS en FALSE
//
"AlarmasHMI".cin_alim_sete := FALSE;

"CIN_ALIM_DB_1".alarma := FALSE;
```

Listing 6.11: Código FC 'Reset_cina' para el control de la resolución de la alarma de la [Cinta de alimentación](#)

El funcionamiento del resto de los FCs para el control/gestión de las alarmas es exactamente igual pero adaptado para las cada parte de la línea de producción.

6.2 Desarrollo de la aplicación para el HMI

Ya hemos visto una breve explicación de que tipo de dispositivo es un HMI en la [subsección 5.3.2](#). Para poder hacer uso del HMI lo primero que tendremos que hacer es establecer una conexión entre el HMI y el PLC. Para este proyecto la conexión se ha realizado en el propio entorno local, así que los dos dispositivos pertenecen a la misma subred.

Una vez realizada la conexión lo que tendremos que hacer es asociar las variables del HMI a las del PLC para que estos dos puedan comunicarse y poder realizar las acciones que queremos con el HMI. Como los cambios de modos, usar los actuadores y ver el estado de los sensores desde el HMI. Para ello tendremos que dar de alta las variables del HMI en el apartado del 'HMI Tags' en la configuración de Tia Portal del HMI. Y decirle con que Tag del PLC queremos vincularlo. En caso de que no exista el Tag del PLC con el que lo queremos vincular lo podemos crear en ese instante. Ya que TIA Portal nos ofrece su ayuda.

HMI tags						
Name	Tag table	Data type	Connection	PLC name	PLC tag	Address
HMI_FIN2_R	Default tag table	Bool	HMI_Connection_1	PLC_1	HMI_FIN2_R	
HMI_MAQ_ENT_ACTIV	Default tag table	Bool	HMI_Connection_1	PLC_1	HMI_MAQ_ENT_ACTIV	
HMI_MAQ_INI_GRA_ACTIV	Default tag table	Bool	HMI_Connection_1	PLC_1	HMI_MAQ_INI_GRA_ACTIV	
HMI_MAQ_INI_PEQ_ACTIV	Default tag table	Bool	HMI_Connection_1	PLC_1	HMI_MAQ_INI_PEQ_ACTIV	
HMI_MAQ_ADEL	Default tag table	Bool	HMI_Connection_1	PLC_1	HMI_MAQ_ADEL	
HMI_MAQ_BAJA	Default tag table	Bool	HMI_Connection_1	PLC_1	HMI_MAQ_BAJA	

Figura 6.27: Tags del HMI asociados a los del PLC

6.2.1 Vistas del HMI

En el apartado [sección 4.3](#) ya hemos visto el diseño que ha de seguir el HMI y su la interfaz. Lo primero de todo ha sido crear los usuarios con sus correspondiente roles.

Users						
Name	Password	Automatic logoff	Logoff time	Number	Comment	
admin	*****	<input type="checkbox"/>	5	1	The user 'Administrator' is as...	
mant	*****	<input type="checkbox"/>	5	2		
basic	*****	<input checked="" type="checkbox"/>	5	3		
<Add new>						

Groups						
Member of	Name	Number	Display name	Password aging	Comment	
<input checked="" type="radio"/>	Administrator group	1	Administrator group	<input type="checkbox"/>	The 'Administrator' group is initially granted all rights.	
<input type="radio"/>	Users	2	Users	<input type="checkbox"/>	The 'Users' group is initially granted 'Operating' rights.	
<input type="radio"/>	Mantenimiento	3	Mantenimiento	<input type="checkbox"/>	Usuario con acceso a las maquinas y alarmas	
<input type="radio"/>	Basic	4	Basic	<input type="checkbox"/>	Usuario con permisos basicos	

Figura 6.28: Creación de usuarios y roles

Después de esto se han desarrollado las vistas a las que se les ha puesto una restricción de permisos para que solo los usuarios privilegiados puedan acceder. Esto ha sido comentado en la [subsección 5.3.2](#).

Vista Home

Esta es la página principal una vez se ha conseguido iniciar sesión. Desde aquí podremos acceder a la vista de 'User Management' (gestión de usuarios), 'Modos' (control de los modos y máquinas) y 'Alarmas' (visualización de las alarmas que han sido activadas). Cabe recalcar que en todo momento cuando salte una alarma saltará una alerta para avisarlos de esta. Así que no tendremos que navegar siempre hasta la vista de 'Alarmas'.

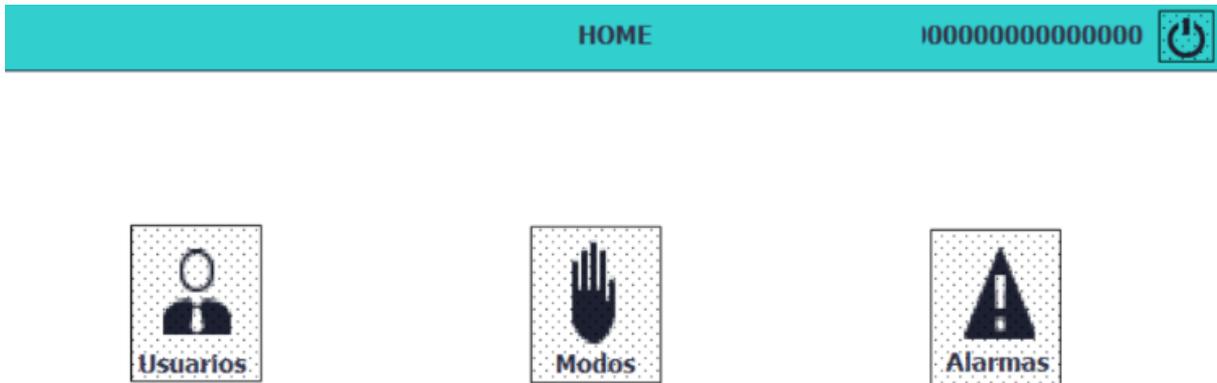


Figura 6.29: HMI - Vista Home

Vista User Management

En esta vista los usuarios que tengan permiso para acceder podrán crear usuarios, asignarles permisos y establecer si se desea un tiempo en el que sería desconectado automáticamente si se encuentra ausente. Desde esta vista y el resto que se comentarán podemos ir a la vista anterior mediante el botón indicado por la flecha arriba a la izquierda. Arriba a la derecha tenemos un botón de desconexión y aparece el nombre de usuario que está actualmente conectado en el HMI.



Figura 6.30: HMI - User Management

Vista Alarmas

Se permite visualizar a los usuarios privilegiados las alarmas que han ido activándose. Como ya se ha mencionado antes para tener una mejor gestión de las alarmas también nos aparecería una notificación independientemente de la vista en la que nos encontremos donde se nos dirá que alarma ha saltado.

Las alarmas han sido creadas en el HMI de la siguiente manera como se puede observar en la figura [figura 6.31](#).

Discrete alarms							
ID	Name	Alarm text	Alarm class	Trigger tag	Trigge..	Trigger address	HMI acknowl...
1	Seta_emer	Seta de emergencia activada	Errors	alarma_set...	8	%DB57.DBX0.0	<No tag>
2	MAQ_TARDA	Maquina inicio 1: producto no llega a la maquina	Errors	alarma_maqi...	9	%DB57.DBX0.1	<No tag>
4	MAQ_PALLET_TAR...	Maquina inicio 1: palet atascado	Errors	alarma_maqi...	10	%DB57.DBX0.2	<No tag>
5	MAQI2_TARDA	Maquina inicio 2: producto no llega a la maquina	Errors	alarma_maqi...	11	%DB57.DBX0.3	<No tag>
7	MAQI2_PALLET_TA...	Maquina inicio 2: palet atascado	Errors	alarma_maqi...	12	%DB57.DBX0.4	<No tag>
8	MAQE_TARDA	Maquina entrega: producto no llega a la maquina	Errors	alarma_maqe..	13	%DB57.DBX0.5	<No tag>
10	TURN	Turn table: productos detectados en sensor frontal y trasero	Errors	alarma_turn	14	%DB57.DBX0.6	<No tag>
11	FIN PEQ	Fin peq: varios objetos en la cinta	Errors	alarma_fin_peq	15	%DB57.DBX0.7	<No tag>
12	FIN PG	Fin pgran: varios objetos en la cinta	Errors	alarma_fin_pg	0	%DB57.DBX1.0	<No tag>
3	CINA_SETA	Cin alim: se ha detectado un error en la cinta de alimentación	Errors	alarma_cin_al.	1	%DB57.DBX1.1	<No tag>
6	MAQP_SETA	Maquina proces: se ha detectado un error con el producto	Errors	alarma_maqp	2	%DB57.DBX1.2	<No tag>

Figura 6.31: Creación de las alarmas en el HMI

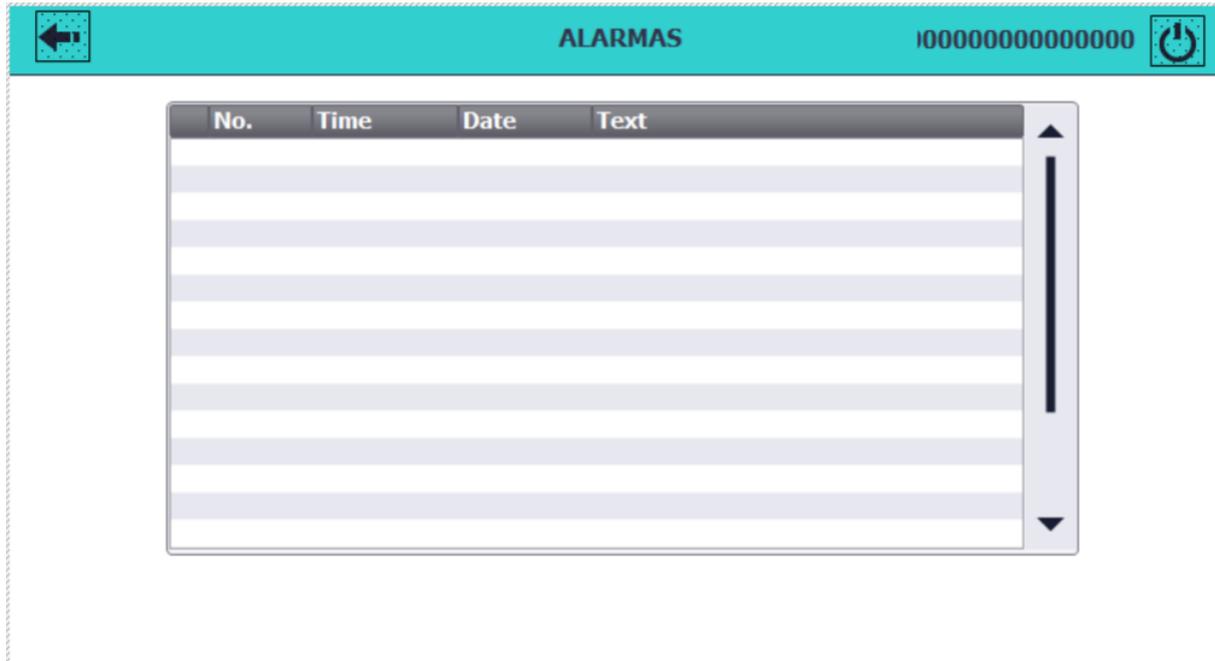


Figura 6.32: HMI - Alarmas

Vista Modos

Esta es sin duda la vista más completa de todas las que han sido desarrolladas. Se nos muestran datos a tiempo real de la planta como la recogida de palets y las entregas realizadas. Podemos ver en un led llamado 'MANIPULACIÓN HMI' si el panel de mandos nos ha dado acceso para tomar el control con el HMI. Podemos controlar los modos, y tenemos un botón para ocultar el panel de los modos y no activarlos de manera accidental. Por otra parte tenemos una imagen representativa de la línea de producción donde mediante el icono de una caja se nos muestra que parte está activa. Y mediante el icono de una alerta en que parte en específico ha saltado una alarma.

Finalmente al pulsar sobre una parte se nos abriría otra nueva vista donde podremos controlar las máquinas cuando estemos en modo manual. Y cuando haya saltado una alarma de una parte concreta la podremos gestionar mediante los botones que aparecerán solo para ese instante llamados 'CONTINUE' y 'RESET' de los que ya hemos hablado previamente.

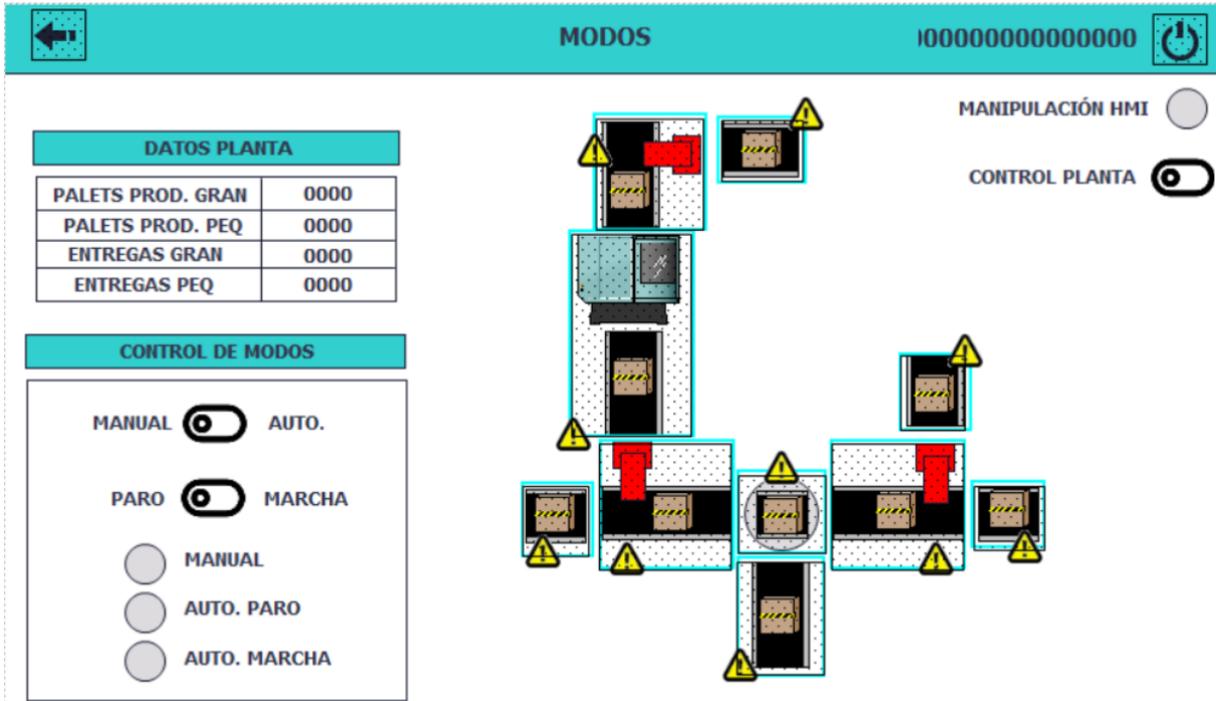


Figura 6.33: HMI - Modos

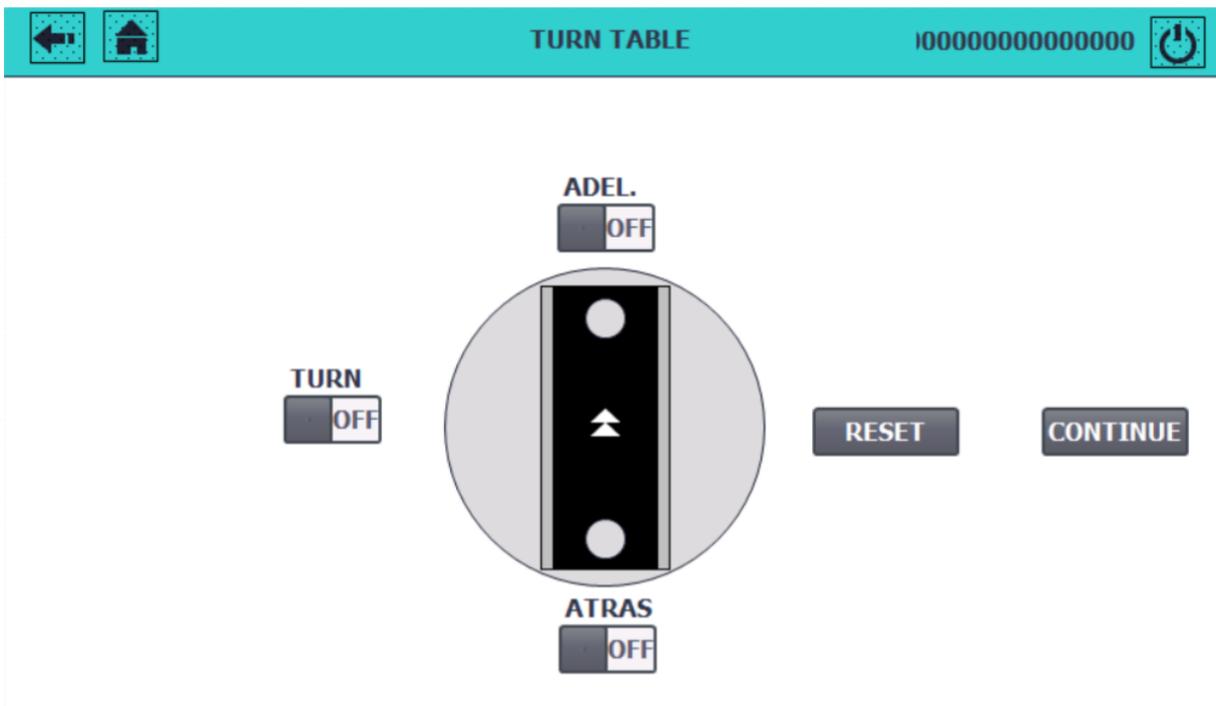


Figura 6.34: HMI - Modos - Turntable

6.3 Configuración de KEPServerEX

Una vez ya hemos realizado el desarrollo en TIA Portal tanto por parte del PLC como por parte del HMI tendremos que conectarnos a nuestro servidor de comunicaciones KEP-ServerEX. Este servidor nos ayudará a virtualizar la línea de producción y a realizar la monitorización.

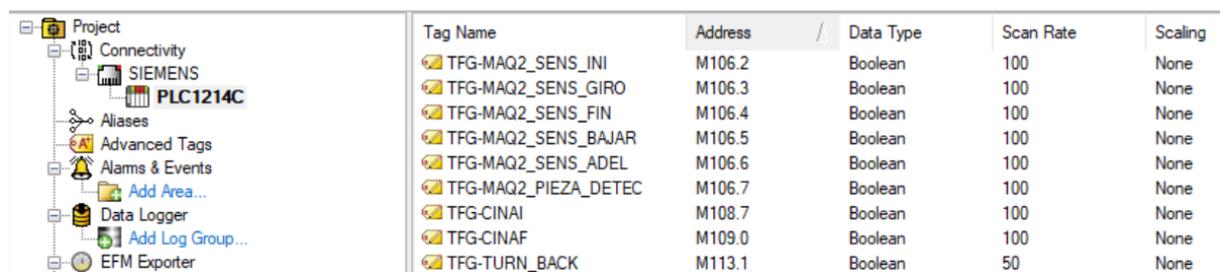
6.3.1 Crear conexión

Lo primero para poder crear una conexión es necesario añadir en un nuevo canal que use el driver de Siemens TCP/IP Ethernet. Ya que es el que usa el PLC SIMATIC S7-1200 con el que estamos trabajando. Y después le asignaremos la dirección IP del PLC para que el servidor pueda acceder a este y el adaptador de red al que se ha de conectar.

6.3.2 Crear Tags

Para crear los Tags en KEPServerEX tendremos que tener en cuenta las variables en memoria que hemos creado del PLC (los PLC Tags) y pasarselos a KEPServerEX. Una vez realizada la conexión añadiremos los tags. Estos tags serán los mismos que hemos añadido en nuestro proyecto de Tia portal. Serán creados dentro del dispositivo que hemos añadido anteriormente.

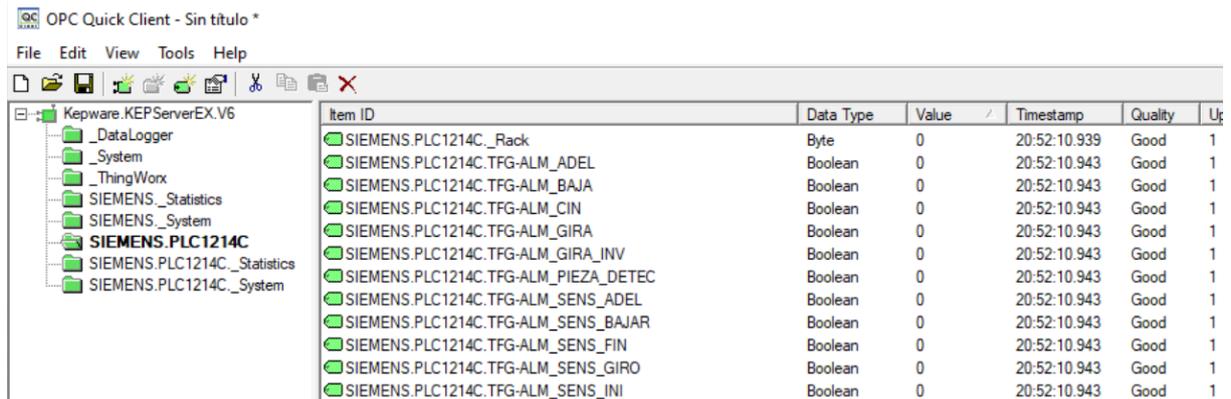
Les especificaremos un nombre (no tiene por qué ser el mismo que en Tia portal) y su dirección (esta sí que tiene que ser la misma ya que tenemos que obtener los valores del PLC).



Tag Name	Address	Data Type	Scan Rate	Scaling
TFG-MAQ2_SENS_INI	M106.2	Boolean	100	None
TFG-MAQ2_SENS_GIRO	M106.3	Boolean	100	None
TFG-MAQ2_SENS_FIN	M106.4	Boolean	100	None
TFG-MAQ2_SENS_BAJAR	M106.5	Boolean	100	None
TFG-MAQ2_SENS_ADEL	M106.6	Boolean	100	None
TFG-MAQ2_PIEZA_DETEC	M106.7	Boolean	100	None
TFG-CINAI	M108.7	Boolean	100	None
TFG-CINAF	M109.0	Boolean	100	None
TFG-TURN_BACK	M113.1	Boolean	50	None

Figura 6.35: Tags creados en KEPServerEX

Todos los tags que añadiremos en Kepserverser serán los que queramos usar tanto en Factory I/O (para los sensores y actuadores) y en NODE-RED para que estos sean solamente leídos. Los Tags que se usen de manera interna como por ejemplo los del HMI no serán necesario crearlos en Kepserverser. Además ejecutaremos el OPC Quick Client de Kepserverser. Aquí podremos ver desde el propio servidor el valor que toman los Tags en cada momento y la calidad de estos. Ya que si su calidad es buena (indicada como “Good”) todo esta funcionando correctamente dentro del servidor. En caso contrario el servidor se podría haber desconectado o hay algún problema con los Tags.



Item ID	Data Type	Value	Timestamp	Quality	Up
SIEMENS.PLC1214C._Rack	Byte	0	20:52:10.939	Good	1
SIEMENS.PLC1214C.TFG-ALM_ADEL	Boolean	0	20:52:10.943	Good	1
SIEMENS.PLC1214C.TFG-ALM_BAJA	Boolean	0	20:52:10.943	Good	1
SIEMENS.PLC1214C.TFG-ALM_CIN	Boolean	0	20:52:10.943	Good	1
SIEMENS.PLC1214C.TFG-ALM_GIRA	Boolean	0	20:52:10.943	Good	1
SIEMENS.PLC1214C.TFG-ALM_GIRA_INV	Boolean	0	20:52:10.943	Good	1
SIEMENS.PLC1214C.TFG-ALM_PIEZA_DETEC	Boolean	0	20:52:10.943	Good	1
SIEMENS.PLC1214C.TFG-ALM_SENS_ADEL	Boolean	0	20:52:10.943	Good	1
SIEMENS.PLC1214C.TFG-ALM_SENS_BAJAR	Boolean	0	20:52:10.943	Good	1
SIEMENS.PLC1214C.TFG-ALM_SENS_FIN	Boolean	0	20:52:10.943	Good	1
SIEMENS.PLC1214C.TFG-ALM_SENS_GIRO	Boolean	0	20:52:10.943	Good	1
SIEMENS.PLC1214C.TFG-ALM_SENS_INI	Boolean	0	20:52:10.943	Good	1

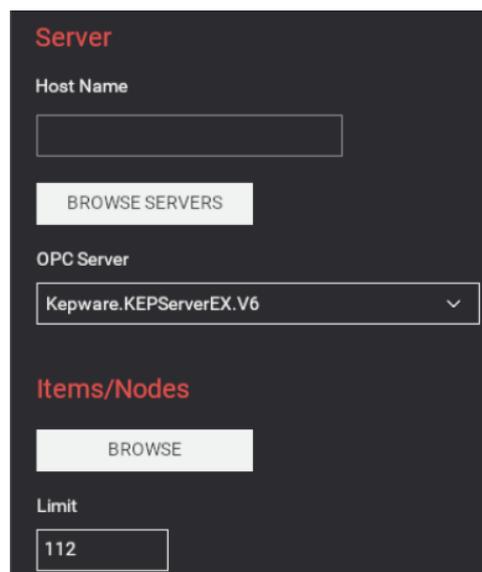
Figura 6.36: OPC Quick Client de Kepserver

6.4 Configuración de Factory IO

Una vez con la escena creada con las [Partes en las que se ha dividido la línea de producción](#) nos pondremos a realizar la conexión con KEPServerEX para dotar así al PLC del control de la escena de FactoryIO.

6.4.1 Conexión con KEPServerEX

La conexión en Factory IO se realiza mediante su driver OPC Client DA/UA. Dentro de la configuración de este driver le diremos cual es el servidor OPC al que nos queremos conectar y cuantos nodos (tags) queremos añadirle. Y después de este paso podremos realizar la vinculación de los sensores y actuadores con los Tags de KEPServerEX.



The screenshot shows a configuration window with the following fields and buttons:

- Server** section:
 - Host Name: [Empty text box]
 - BROWSE SERVERS: [Button]
 - OPC Server: [Dropdown menu showing 'Kepware.KEPSEServerEX.V6']
- Items/Nodes** section:
 - BROWSE: [Button]
- Limit** section:
 - Limit: [Text box containing '112']

Figura 6.37: Conexión con el driver de KEPServerEX

6.4.2 Vinculación con los Tags de KEPServerEX

La vinculación de los sensores y actuadores con los Tags que tenemos en Kepserver es muy sencilla. Ya que simplemente tendremos que arrastrarlos hacia el Tag con el que queremos que se conecten para que así estos se puedan comunicar con el servidor y que el servidor se pueda comunicar con los sensores y actuadores modificando su estado.

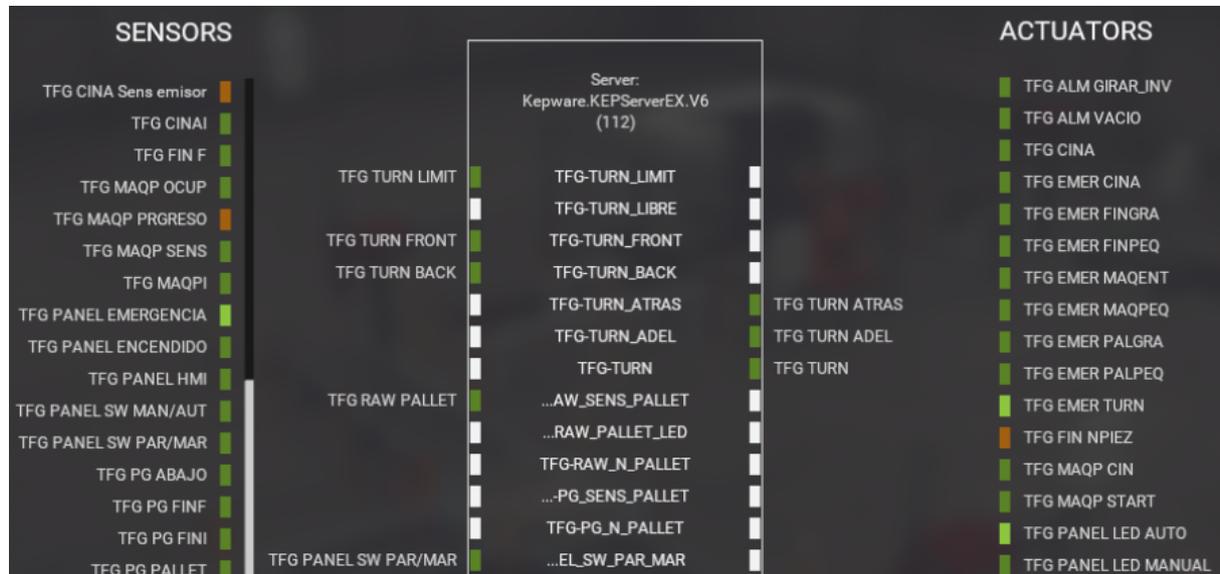


Figura 6.38: Conexión con los Tags de KEPServerEX

6.5 Configuración de Node-RED

Para poder instalar NODE-RED lo primero que tenemos que hacer es instalar Node.js en nuestro equipo según viene por defecto. En este caso hemos instalado la versión v18.16.0. Una vez instalado ejecutaremos la consola “Node.js commad prompt” que se nos ha descargado. Y en esta consola ejecutaremos el comando correspondiente para la instalación de Node-RED.

```
npm install -g --unsafe-perm node-red
```

Listing 6.12: Instalación NODE-RED por consola

Y ahora ya tendremos instalada la versión 3.0.2 de Node-RED. Una vez instalado en la misma consola ejecutaremos el comando “node-red” para así ejecutar el programa de Node-RED. Cuando ejecutemos este comando se nos mostrará la dirección donde se está ejecutando Node-RED. En este caso será ‘http://127.0.0.1:1880/'. Donde la dirección IP será la local de nuestro equipo y el puerto será el puerto que usa Node-RED. Al poner la dirección anterior en el navegador podemos ver como se despliega la aplicación de Node-RED.

```
node-red
Welcome to Node-RED
=====
28 Jun 20:40:45 - [info] Node-RED version: v3.0.2
28 Jun 20:40:45 - [info] Node.js version: v18.16.0
28 Jun 20:40:45 - [info] Windows_NT 10.0.19045 x64 LE
28 Jun 20:40:48 - [info] Loading palette nodes
28 Jun 20:40:50 - [info] Dashboard version 3.4.0 started at /ui
28 Jun 20:40:55 - [info] Settings file : C:\Users\mmueasp\.node-red\settings.js
28 Jun 20:40:55 - [info] Context store : 'default' [module=memory]
28 Jun 20:40:55 - [info] User directory : C:\Users\mmueasp\.node-red
28 Jun 20:40:55 - [warn] Projects disabled : editorTheme.projects.enabled=false
28 Jun 20:40:55 - [info] Flows file : C:\Users\mmueasp\.node-red\flows.json
28 Jun 20:40:55 - [info] Server now running at http://127.0.0.1:1880/
28 Jun 20:40:55 - [warn]

-----
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----
28 Jun 20:40:55 - [info] Starting flows
```

Figura 6.39: Ejecución de Node-RED

6.5.1 Tipos de nodos usados en Node-RED y en el proyecto

Los nodos en Node-RED se tratan de elementos que aportan distintas funciones según el tipo de estos. Son usados para construir flujos de trabajo mediante una programación visual. La manera en la que se programa visualmente es mediante nodos que vienen incluidos por defecto o instalando librerías dentro de Node-RED las cuales nos proporcionarán nuevos nodos.

Inject nodes

Un inject node se trata de una función básica de Node-RED la cual inicia un flujo de datos. En este tipo de nodo le asignaremos un nombre para saber que dato está extrayendo. Por otra parte, lo que queremos que devuelva el nodo es un mensaje. Este mensaje está especificado en las propiedades del inject node como “msg”. “msg” es un objeto usado en Node-RED que se usa para pasar información de un nodo a otro. Dos de las propiedades del mensaje son “payload” y “topic”.

“Payload” (conocido también como carga útil) es la información que se transporta dentro del mensaje. Como por ejemplo podría ser el valor de humedad que ha detectado un sensor, valores de acelerometría o como en nuestro caso los contadores o temporizadores que tenemos en nuestro proyecto.

“Topic” (conocido como tema) es la etiqueta que se le asigna al mensaje, define el tipo de información que contiene el mensaje. Como, por ejemplo, categorizar que el mensaje contiene la información de la humedad, de un contador o de un temporizador.

En las propiedades del inject node le especificaremos que el “payload” va a ser un “timestamp”. Y en el “topic” le especificaremos el nodeId de el tag que tengamos creado en Kepserver que le queramos pasar a Node-RED. Este nodeId lo obtenemos con UaExpert. También podemos especificarle cada cuanto tiempo el inject node envía esa información al siguiente nodo.

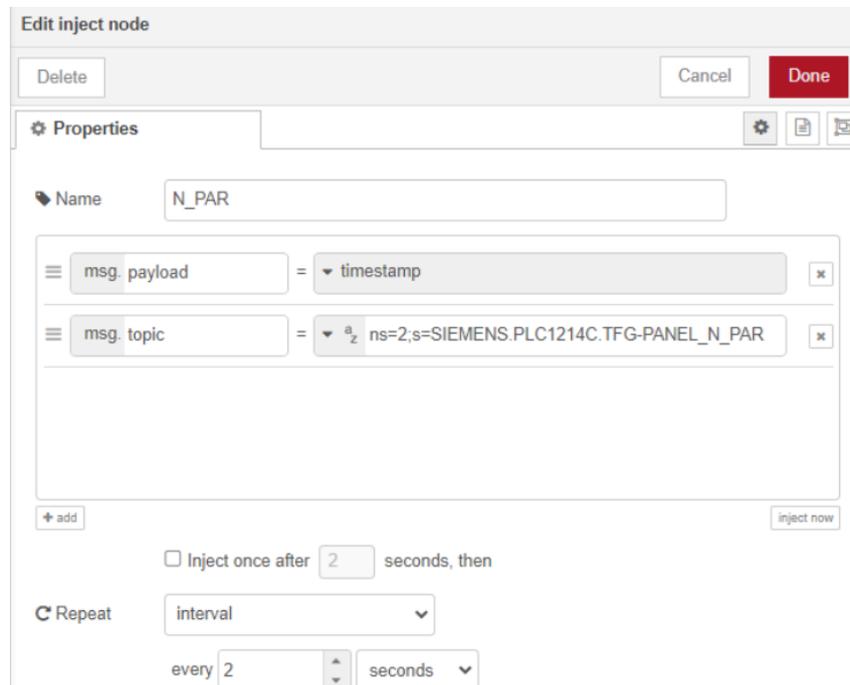


Figura 6.40: Inject node - Configuración

Como se observa en la figura [figura 6.40](#) esta un ejemplo de Inject node que ha sido añadido al proyecto. Además en el Topic se ha especificado el "ns"(NamespaceIndex, espacio de nombres) que se usa para identificar los nodos dentro del servidor OPC y el "s" se trata del identificador del Tag. Para obtener estos datos mencionados se ha tenido que hacer uso de la herramienta UaExpert y realizar una conexión con nuestro servidor de KEPServerEX.

OpcUa

Una vez tengamos la paleta instalada en NODE-RED escogeremos el nodo “Opc Ua Client”. En el parámetro “endpoint” especificaremos el destino de la red al cual nos queremos conectar, en este caso pondremos la dirección de nuestro servidor de Kepserver. E indicaremos en “action” ya que queremos leer los datos. Cabe destacar que los también podemos escribir sobre los datos, navegar sobre estos, suscribirnos/desuscribirnos o realizar mas acciones. Pero en este caso solo nos interesa leer. Además, estos datos de salida que envía este nodo pueden ser tratados mediante distintos nodos de función en NODE-RED. En caso de extrajésemos datos complejos podría ser interesante pasarlos por los nodos de funciones que se ofrecen.

Figura 6.41: Inject node - Configuración

InfluxDB out node

La información del nodo Opc ua client saldrá hacia un nodo de influxdb out node. Este se trata de un nodo de salida que se usa para escribir en las bases de datos de influxdb.

En este tipo de nodo especificaremos un nombre para saber que dato se enviara a la base de datos. La propia configuración del servidor y la base de datos a la que queremos enviar los datos mostrada en la [figura 6.42](#) y [figura 6.43](#). Y el tipo de medición. Aquí es donde crearemos las mediciones en lugar de usar el cliente de influxdb (fichero ejecutable “influx”)

Figura 6.42: InfluxDB out node - Configuración (Parte 1)

Figura 6.43: InfluxDB out node - Configuración (Parte 2)

Debug node

Se trata de un nodo que nos ayuda a realizar la depuración y la visualización de los datos durante el desarrollo del flujo. Con este nodo imprimiremos la información por consola. Una buena practica es usar un debug node cada vez que vayamos a añadir una medición nueva y queramos enviar los datos a influx para saber que todo funciona correctamente. Simplemente tendremos que añadir el output, en este caso el “payload” para que nos saque los datos por la ventana de “debug messages”.

Este nodo lo podemos activar o desactivar a nuestro antojo sin necesidad de borrarlo. Por lo que lo podemos tener en nuestro flujo de comunicación siempre y desactivarlo o activarlo según nuestra conveniencia.

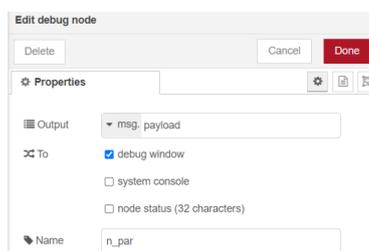


Figura 6.44: Debug node - Configuración

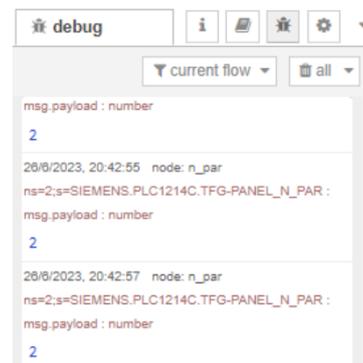


Figura 6.45: Información que muestra el debug node

6.5.2 Resultado del flujo realizado

Así se nos mostraría el flujo completo de todos los Tags que queramos leer de Kepserver y enviárselo a Influx desde NODE-RED.

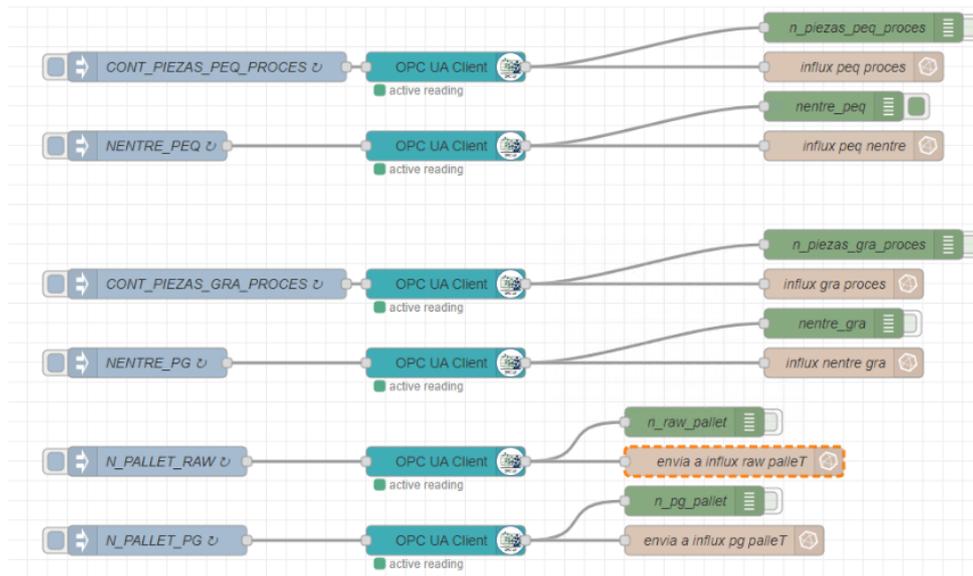


Figura 6.46: Flujos realizados - Parte 1

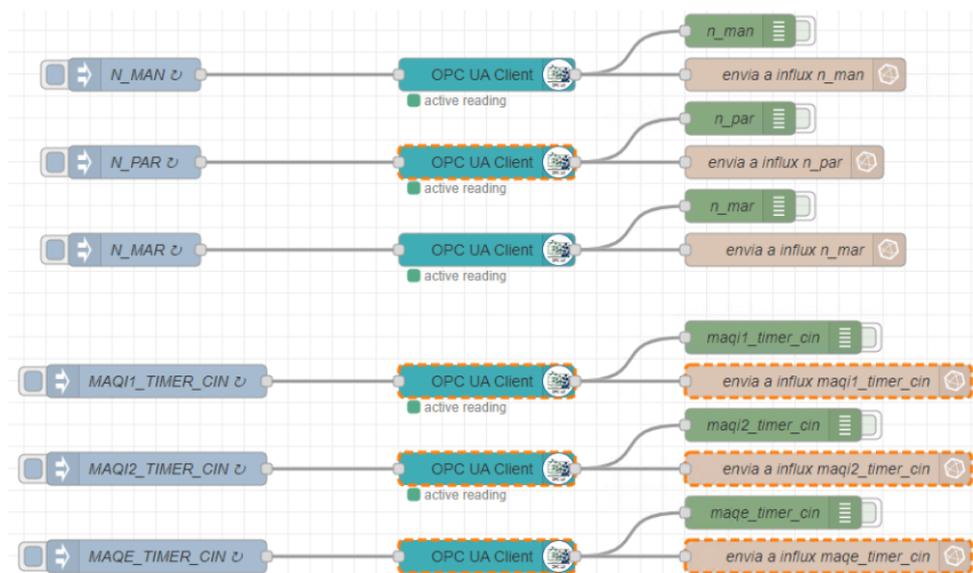


Figura 6.47: Flujos realizados - Parte 2

6.6 Configuración de InfluxDB

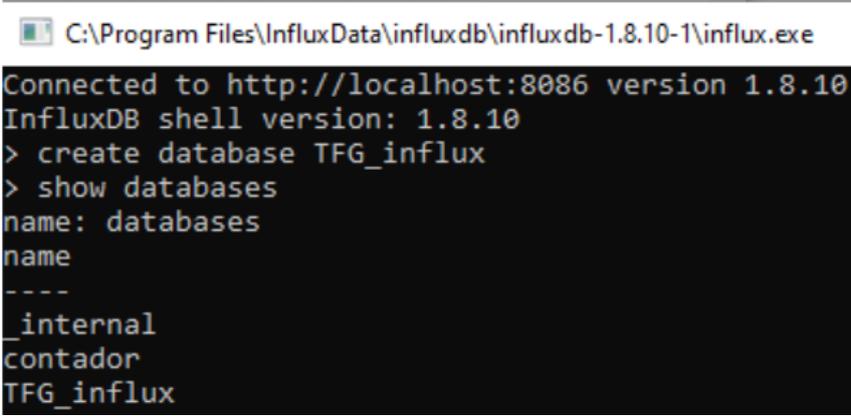
Se usará la versión 1.8.10 de influxdb para el proyecto. Cuando instalemos influxdb nos vendrá con diversos ejecutables. Solamente nos centraremos en usar los ejecutables “influx.exe” y “influxd.exe”. “influx.exe” se trata del cliente y “influxd.exe” se trata del servidor de influx.

También se encuentra un fichero de configuración llamado “influxdb.conf”. En este fichero podremos modificar la configuración donde se quieren almacenar los datos, el puerto que

usaremos y distintas configuraciones. En ese proyecto no ha sido modificado este fichero y se ha usado como viene tras la instalación de influx.

6.6.1 Creación de la base de datos

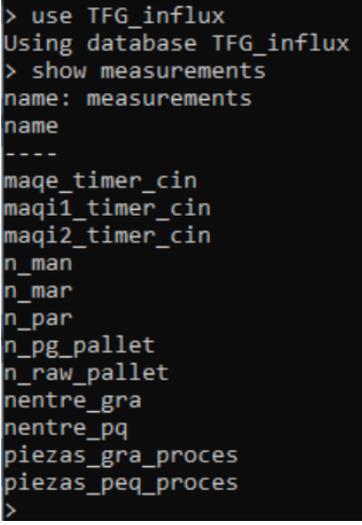
Para crear la base de datos usaremos el cliente “influx.exe”. Solo lo usaremos para la creación de la base de datos y nada mas. También lo podremos utilizar si queremos eliminar la base de datos y las medidas que tiene esta. No crearemos ninguna medida nada mas crear la base de datos. Esto lo haremos directamente en NODE-RED. Además podremos ver cuales son las medidas que tienen nuestras bases de datos. En nuestro caso solo usaremos una única base de datos la cual hemos llamado “TFG_influx”.



```
C:\Program Files\InfluxData\influxdb\influxdb-1.8.10-1\influx.exe
Connected to http://localhost:8086 version 1.8.10
InfluxDB shell version: 1.8.10
> create database TFG_influx
> show databases
name: databases
name
----
internal
contador
TFG_influx
```

Figura 6.48: Creación de la BBDD con Influx por consola

En la figura [figura 6.48](#) vemos como se crea la base de datos. Además podremos ver cuales son las medidas que tienen nuestra bases de datos mediante los siguientes comandos:

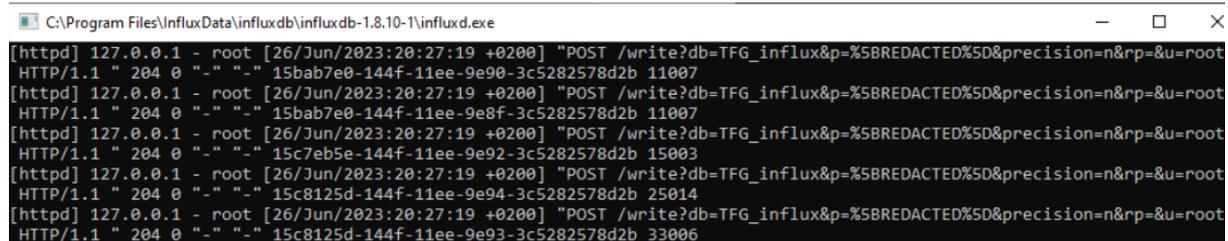


```
> use TFG_influx
Using database TFG_influx
> show measurements
name: measurements
name
----
maqe_timer_cin
maqi1_timer_cin
maqi2_timer_cin
n_man
n_mar
n_par
n_pg_pallet
n_raw_pallet
nentre_gra
nentre_pq
piezas_gra_proces
piezas_peq_proces
>
```

Figura 6.49: Ver las medidas creadas en la BBDD desde la consola de Influx

6.6.2 Ejecución del servidor

Como se ha comentado antes el servidor se arranca ejecutando el fichero “influxd.exe”. Al arrancarlo tendrá acceso a las bases de datos que hallamos creado en el cliente.



```
C:\Program Files\InfluxData\influxdb\influxdb-1.8.10-1\influxd.exe
[httpd] 127.0.0.1 - root [26/Jun/2023:20:27:19 +0200] "POST /write?db=TFG_influx&p=%5BREDACTED%5D&precision=n&rp=&u=root
HTTP/1.1 " 204 0 "-" "-" 15bab7e0-144f-11ee-9e90-3c5282578d2b 11007
[httpd] 127.0.0.1 - root [26/Jun/2023:20:27:19 +0200] "POST /write?db=TFG_influx&p=%5BREDACTED%5D&precision=n&rp=&u=root
HTTP/1.1 " 204 0 "-" "-" 15bab7e0-144f-11ee-9e8f-3c5282578d2b 11007
[httpd] 127.0.0.1 - root [26/Jun/2023:20:27:19 +0200] "POST /write?db=TFG_influx&p=%5BREDACTED%5D&precision=n&rp=&u=root
HTTP/1.1 " 204 0 "-" "-" 15c7eb5e-144f-11ee-9e92-3c5282578d2b 15003
[httpd] 127.0.0.1 - root [26/Jun/2023:20:27:19 +0200] "POST /write?db=TFG_influx&p=%5BREDACTED%5D&precision=n&rp=&u=root
HTTP/1.1 " 204 0 "-" "-" 15c8125d-144f-11ee-9e94-3c5282578d2b 25014
[httpd] 127.0.0.1 - root [26/Jun/2023:20:27:19 +0200] "POST /write?db=TFG_influx&p=%5BREDACTED%5D&precision=n&rp=&u=root
HTTP/1.1 " 204 0 "-" "-" 15c8125d-144f-11ee-9e93-3c5282578d2b 33006
```

Figura 6.50: Servidor de InfluxDB en ejecución

Una vez ejecutado el servidor podremos ver como se le van introduciendo los datos. Además podemos ver el código del estado de las peticiones HTTP. Con esto podremos saber si nuestro servidor está funcionando correctamente.

6.7 Configuración de Grafana

El primer paso para llevar a cabo la monitorización es conectarnos a una fuente de datos, en este caso a la base de datos InfluxDB. Simplemente tenemos que desde Grafana ir al menú y acceder al apartado de conexiones. Buscar influxDB y añadirla a nuestra lista de conexiones.

Después en el menú “Data sources” seleccionaremos la conexión que hemos agregado y le especificamos la url de InfluxDB donde le especificaremos la dirección IP y el puerto. La dirección IP en este caso es 127.0.0.1 (localhost) ya que estamos hosteando la base de datos de manera local. Y el puerto por defecto que ejecuta el servicio HTTP de InfluxDB para poder conectarnos finalmente con Grafana es el 8086. Hay que destacar que este puerto se puede modificar en el fichero de configuración de InfluxDB. Finalmente añadiremos el nombre de nuestra base de datos que estamos utilizando en InfluxDB. En la [figura 6.51](#) se observa la conexión creada.

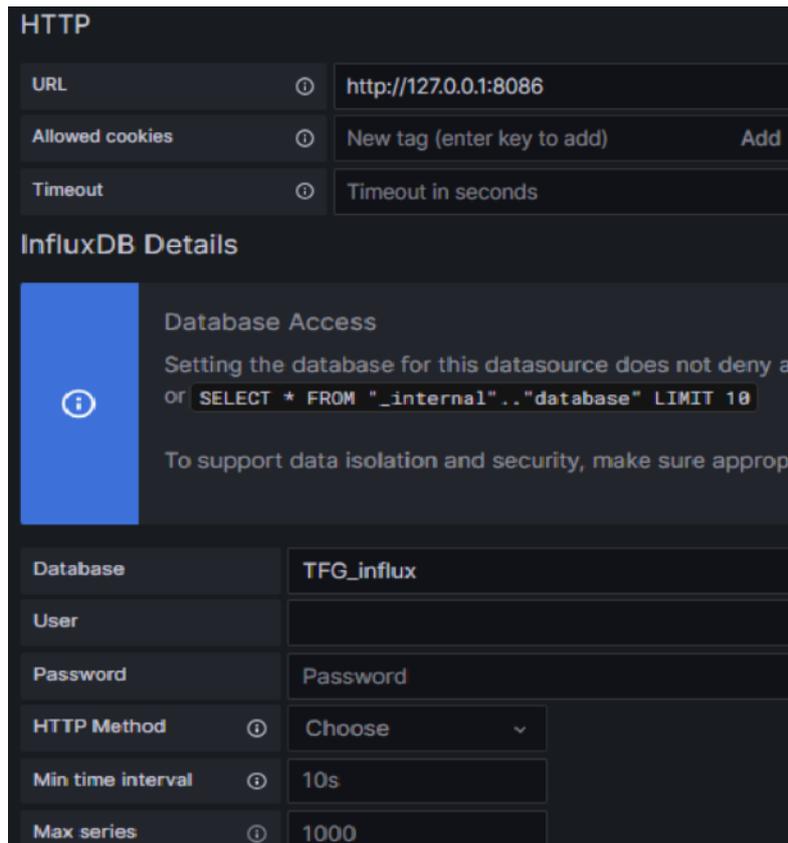


Figura 6.51: Conexión grafana con Influx

6.7.1 Dashboards y paneles

La forma en la que se visualizan los datos en Grafana es mediante pantallas de visualización llamadas “dashboards”. Los “dashboards” se tratan de ventanas donde podemos visualizar un conjunto de paneles. Y estos pueden ser agrupados en distintas carpetas para tenerlos más organizados. En la sección de “Dashboards” añadiremos un “Dashboard” nuevo. Para el Proyecto se han creado tres dashboards distintos como se ve en la [figura 6.52](#).

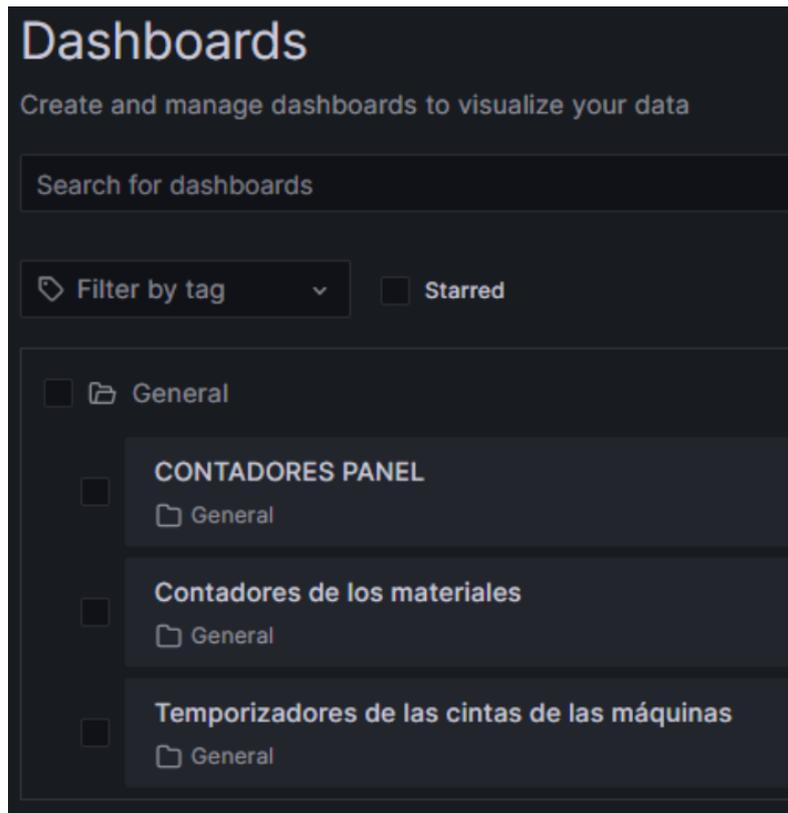


Figura 6.52: Dashboards creados

Panel contadores de los modos

En el “dashboard” contadores panel visualizamos el número de veces que se ha cambiado el modo del automatismo. Ya sea porque el operario lo ha cambiado el mismo o porque se haya activado la alarma de emergencia general y se haya cambiado de modo. Puede ser bastante útil para saber si la planta está funcionando de manera autónoma si no se han realizado muchos cambios de modo.

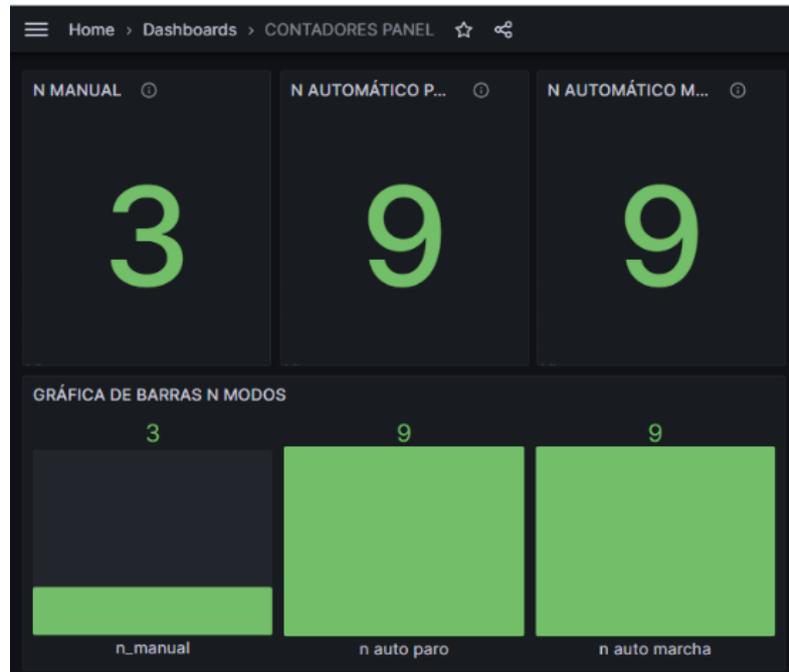


Figura 6.53: Panel de los contadores de los modos

Panel contadores de los materiales

El “dashboard” contadores de los materiales muestra el número de palets que han sido recogidos tanto de los productos pequeños como de los productos grandes.

Los productos que han sido procesados y están esperando para ser entregados. Y las entregas de cada tipo de producto. Además, en una gráfica de pastel se hace una comparativa de estas entregas. Esto es realmente útil para saber cual de los dos tipos de producto es el que más ha sido entregado al final de la línea de producción. Hay que tener en cuenta que el procesado de dos piezas pequeñas equivale a una entrega de piezas pequeñas, por lo que probablemente el número de entregas será inferior al de productos grandes.



Figura 6.54: Panel de los materiales

Panel temporizadores de las cintas

Y por último el dashboard temporizadores de las cintas de las máquinas nos indica el tiempo desde que el producto ha entrado en la cinta y se está desplazando hasta el final de esta para ser recogido por la máquina. De aquí podemos extraer cual es la cinta que se utiliza durante más tiempo y cuál es la que más veces se usa. En la figura [figura 6.55](#) vemos como la 'cinta de la maq entrega' es la que menos tiempo está activada (ya que es más corta) y es la que menos frecuencia de uso tiene respecto a las otras. Con lo que podemos llegar a la conclusión de que podríamos aumentar la velocidad de las cintas del inicio para aumentar la productividad de nuestra línea.

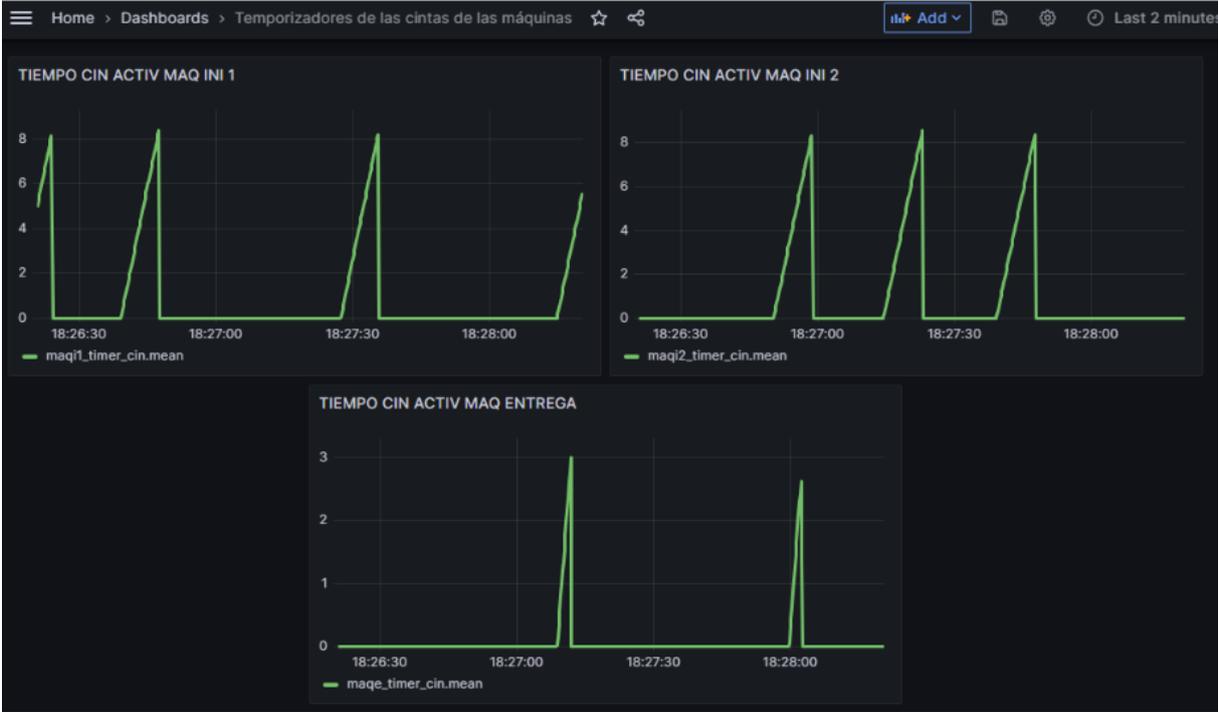


Figura 6.55: Panel de los temporizadores de las cintas de las máquinas

7 Resultados obtenidos

En este capítulo se adjuntarán distintos vídeos donde se mostrarán las pruebas que se han realizado. Entre estas pruebas podremos ver el funcionamiento a la perfección del modo automático, los cambios de modos y control por HMI y la monitorización de los datos con Grafana.

En las pruebas se verá la línea de producción virtualizada funcionando en conjunto con el control realizado del HMI.

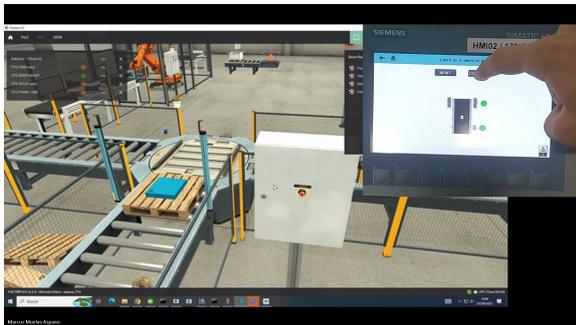


Figura 7.1: Vídeo Media UPV: "Control de una línea de producción virtualizada con monitorización de datos". Haz clic sobre la imagen para acceder al vídeo

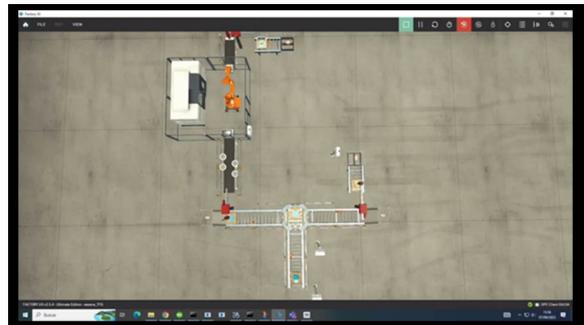


Figura 7.2: Vídeo Media UPV: "Funcionamiento del modo automático marcha en una línea de producción en Factory I/O". Haz clic sobre la imagen para acceder al vídeo

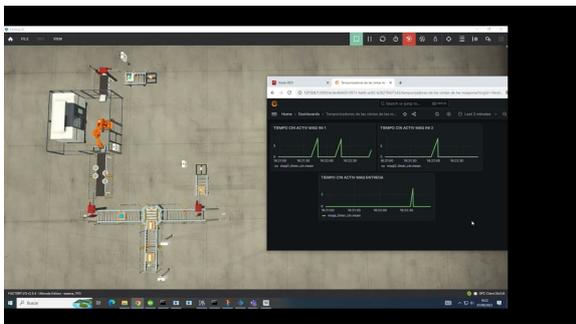


Figura 7.3: Vídeo Media UPV: "Monitorización del tiempo de las cintas con Grafana". Haz clic sobre la imagen para acceder al vídeo

8 Conclusiones y mejoras

Gracias a este proyecto final de carrera he podido profundizar más en el mundo de la automatización e industria digital. Y también he podido crear un proceso industrial, a pesar de las limitaciones de Factory I/O. Además, He aprendido a usar tecnologías que no había visto antes y realizando una investigación sobre estas tecnologías empleadas para saber como se tenía que desarrollar el programa y como realizar las conexiones ya sea por la parte de virtualizar la línea o realizando la monitorización de los datos de esta.

Una vez terminado el proyecto me he dado cuenta de la gran cantidad de tiempo que le he dedicado yendo al laboratorio a probar que todo funcionase correctamente. Ya que no dispongo de un PLC, ni de un HMI. La línea de producción ha quedado bastante completa a pesar de ser una virtualización pequeña. Pero como ya se ha comentado en el proyecto, se ha hecho uso de FBs en TIA Portal en caso de que se desee ampliar con las mismas partes. Así que por esta parte es bastante escalable. La parte de la monitorización realmente ha sido más o menos básica por así decirlo. Más que nada porque no podemos sacar otro tipo de datos para monitorizar. Si en Factory I/O hubiese más tipos de actuadores y sensores quizás se hubieran podido extraer datos sobre la humedad, temperatura, presión, etc...

Como mejora del proyecto se podría realizar una gestión de usuarios que fuese persistente usando otro modelo de HMI, pero por otra parte la limitación sería el presupuesto de este. También se podría añadir alguna parte más a la línea de producción ya que es escalable. Y por parte de la base de datos se podría realizar algún tipo de filtrado dentro de Node-RED para que no se almacenen datos anómalos o incluso que estos sean almacenados en otros tipos de medidas para poder evitar errores mostrandolos en Grafana. También habría sido interesante dockerizar Grafana e InfluxDB para que en poder distribuir y ejecutar parte de nuestra aplicación de una manera más sencilla. Ya que por parte de TIA Portal y Factory I/O no se podría dockerizar esa parte del desarrollo.

Por otra parte también cabe destacar que ha sido bastante satisfactorio el desarrollo de este proyecto aunque en algunos momentos avanzase muy poco. Pero al ver todo funcionar correctamente y tomar el control de la línea de producción, aunque sea solo una simulación, ha merecido la pena el tiempo invertido.

Bibliografía

- Berger, H. (2014). *Automating with SIMATIC S7-400 inside TIA portal: configuring, programming and testing with STEP 7 Professional*. Consultado el 6 de noviembre de 2023, desde <https://books.google.es/books?hl=es&lr=&id=2U31AwAAQBAJ&oi=fnd&pg=PP1&dq=tia+portal+plc+alarms&ots=uofiV-T6Xh&sig=Dpo3ctzcDs8PLUgFd-11MK5vSsI#v=onepage&q=tia%20portal%20plc%20alarms&f=false> (vid. pág. 38).
- Chakraborty, M., & Kundan, A. P. (2021). *Grafana*. Consultado el 5 de noviembre de 2023, desde <https://link.springer.com/book/10.1007/978-1-4842-6888-9> (vid. pág. 48).
- Configuración del servidor OPC UA en KEPserverEX: habilitar acceso de cliente remoto*. (2023). Consultado el 5 de noviembre de 2023, desde <https://polimedia.upv.es/visor/?id=0d3e20d0-0b2e-11ed-bc71-774ee224801f> (vid. pág. 46).
- Dias, A. L., Sestito, G. S., Turcato, A. C., & Brandão, D. (2018). *Panorama, challenges and opportunities in PROFINET protocol research*. IEEE. Consultado el 5 de noviembre de 2023, desde <https://ieeexplore.ieee.org/abstract/document/8627173> (vid. pág. 45).
- González, I., Calderón, A. J., Figueiredo, J., & Sousa, J. M. (2019). *A literature survey on open platform communications (OPC) applied to advanced industrial environments*. Consultado el 5 de noviembre de 2023, desde <https://www.mdpi.com/2079-9292/8/5/510> (vid. pág. 42).
- Lekić, M., & Gardašević, G. (2018). *IoT sensor integration to Node-RED platform*. IEEE. Consultado el 5 de noviembre de 2023, desde <https://ieeexplore.ieee.org/abstract/document/8345544> (vid. pág. 47).
- Michel, G. (1990). *Programmable logic controllers: architecture and applications*. Consultado el 5 de noviembre de 2023, desde <https://ieeexplore.ieee.org/abstract/document/8627173> (vid. pág. 18).

- Naqvi, S. N. Z., Yfantidou, S., & Zimányi, E. (2017). *Time series databases and influxdb*. Consultado el 5 de noviembre de 2023, desde https://www.devopsschool.com/blog/wp-content/uploads/2022/09/influxdb_2017.pdf (vid. pág. 47).
- NOLASCO SANDOVAL, L. A. (2021). *DISEÑO DE UN MÓDULO DIDÁCTICO DE SIMULACIÓN DE PROCESOS INDUSTRIALES USANDO PLC S7-1200 Y HMI KTP 700 BASIC. PARA EL LABORATORIO DE AUTOMATIZACIÓN Y CONTROL, DE LA UNIVERSIDAD NACIONAL TECNOLÓGICA DE LIMA SUR-UNTELS*. Consultado el 6 de noviembre de 2023, desde https://repositorio.untels.edu.pe/jspui/bitstream/123456789/526/1/T088A_44807430_T.pdf (vid. pág. 41).
- Salcedo Romero de Ávila, J. V. (2022). *Introducción a Factory I/O*. Consultado el 5 de noviembre de 2023, desde <https://media.upv.es/#/portal/video/f3236b10-9ddd-11eb-a479-ff81c039aecc> (vid. pág. 46).