



Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE GRADO

*Desarrollo de proyectos virtuales de automatización mediante la
aplicación de Redes de Petri y Lenguaje Ladder*

Grado en Ingeniería Mecánica

ALUMNO: Elena Albaladejo Pozo

DIRECTORES: José González Prieto

Milagros Fernández Gavilanes

CURSO ACADÉMICO: 2024-2025

Universida_{de}Vigo



Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE GRADO

*Desarrollo de proyectos virtuales de automatización mediante la
aplicación de Redes de Petri y lenguaje Ladder*

Grado en Ingeniería Mecánica
Intensificación en Tecnología Naval
Cuerpo General

UniversidadeVigo

RESUMEN

Este Trabajo de Fin de Grado desarrolla proyectos virtuales de automatización naval mediante Redes de Petri y lenguaje Ladder, herramientas clave para modelar y programar sistemas industriales. Su objetivo es ofrecer una metodología práctica para enseñar automatización a estudiantes del Grado en Ingeniería Mecánica de la Escuela Naval Militar. Primero, analiza métodos actuales, resaltando las Redes de Petri por su precisión matemática y Ladder por su sencillez industrial.

Se presentan tres proyectos prácticos: el primero automatiza un sistema de propulsión híbrido naval con módulos de control operacional, mantenimiento preventivo y gestión de alarmas. El segundo optimiza la operación logística de un muelle naval mediante carga automática, almacenamiento robótico y gestión energética autónoma. El tercer proyecto automatiza la seguridad operativa del buque con sistemas de detección y extinción de incendios, control de accesos y mantenimiento general.

La metodología propuesta permite aprendizaje efectivo sin necesidad de hardware físico, reduce errores operativos, optimiza tiempos y fortalece competencias técnicas esenciales, mejorando la preparación profesional de futuros oficiales navales.

PALABRAS CLAVE

Automatización, Redes de Petri, Lenguaje *Ladder*, Simulación, Tecnología naval

AGRADECIMIENTOS

En primer lugar, quiero expresar mi más profundo agradecimiento a mis tutores, José González Prieto y Milagros Fernández Gavilanes, por su valioso apoyo, paciencia y dedicación a lo largo de este trabajo. Sin su guía y consejos, este proyecto no habría sido posible.

A mi familia, especialmente a mis padres, Ana y Ramón, y a mi hermana Isabel, quienes son los pilares fundamentales de mi vida. Gracias por vuestro apoyo incondicional, cariño constante y por impulsarme siempre hacia adelante; sin vosotros no estaría donde estoy hoy.

A mis amigos de la Escuela Naval Militar, quienes han estado a mi lado en cada circunstancia, tanto en los buenos momentos como en los más difíciles. Gracias por vuestra camaradería, amistad sincera y por acompañarme siempre en este camino.

Asimismo, agradezco profundamente a mis amigas que desde pequeña me han acompañado, quienes con su ánimo constante me han impulsado a alcanzar mis metas y sueños, brindándome motivación y fuerza en todo momento.

Una mención especial a Cayetano, un gran compañero cuyo apoyo en los momentos difíciles ha sido esencial. Gracias por estar ahí, poniendo buena cara incluso cuando el bigote te lo ponía difícil, eres una criaturita única.

Finalmente, y de manera especial, agradezco a Dios por brindarme la fortaleza, la sabiduría y las oportunidades que han hecho posible llegar hasta aquí.

CONTENIDO

Contenido	5
Índice de Figuras	8
Índice de Tablas.....	11
1 Introducción y objetivos	12
1.1 Contexto y motivación	12
1.2 Objetivos	13
1.3 Estructura de la memoria	13
2 Estado del arte	16
2.1 Métodos de representación del flujo en sistemas	16
2.1.1 Diagramas de bloques funcionales	16
2.1.2 Autómatas finitos	17
2.1.3 Redes de Petri	19
2.1.4 Diagramas de estado	20
2.1.5 Bond graphs	21
2.1.6 Comparación entre métodos y justificación del uso de redes de Petri	23
2.2 Lenguajes para la implementación de redes de Petri	23
2.2.1 Lenguaje Ladder	24
2.2.2 Redes de Petri de alto nivel	24
2.2.3 Lenguaje Sequential Function Chart	25
2.2.4 Comparación de los lenguajes	27
2.3 Proyectos de procesos automatizados	27
2.3.1 Proyecto de embotellado automatizado	29
2.3.2 Proyecto de extracción y análisis automatizado	30
3 Desarrollo del trabajo	32
3.1 Proyecto 1: Automatización operativa y preventiva de un sistema de propulsión híbrido naval.	33
3.1.1 Descripción del sistema de propulsión híbrida	34
3.1.2 Módulo 1: Gestión y control de un sistema de propulsión híbrida.....	35
3.1.2.1 Enunciado	35
3.1.2.2 Tabla de variables.....	36
3.1.2.3 Red de Petri del análisis de gestión y control.....	37
3.1.2.4 Programación y visualización	39
3.1.3 Módulo 2: Gestión del mantenimiento preventivo del sistema de propulsión híbrido.....	48
3.1.3.1 Enunciado.....	48

3.1.3.2 Tabla de variables.....	49
3.1.3.3 Red de Petri del mantenimiento predictivo	50
3.1.3.4 Programación y visualización	52
3.1.4 Módulo 3: Gestión de alarmas y fallos del sistema de propulsión híbrida.....	54
3.1.4.1 Enunciado.....	54
3.1.4.2 Tabla de variables.....	55
3.1.4.3 Red de Petri del sistema de gestión de alarmas y fallos.	56
3.1.4.4 Programación y visualización	60
3.2 Proyecto 2: Automatización logística y energética de un muelle de carga.....	64
3.2.1 Descripción del sistema de carga automatizada	64
3.2.2 Módulo 1: Lancha con plataforma de carga automática.....	64
3.2.2.1 Enunciado.....	64
3.2.2.2 Tabla de variables.....	65
3.2.2.3 Red de Petri de lancha con carga automática	66
3.2.2.4 Programación y visualización	67
3.2.3 Módulo 2: Gestión de la carga y almacenamiento en el muelle automatizado.....	70
3.2.3.1 Enunciado.....	70
3.2.3.2 Tabla de variables.....	72
3.2.3.3 Red de Petri de la gestión de carga y almacenamiento en el muelle automatizado .72	
3.2.3.4 Programación y visualización	74
3.2.4 Módulo 3: Gestión de energía y recarga de baterías.....	77
3.2.4.1 Enunciado.....	78
3.2.4.2 Tabla de variables.....	78
3.2.4.3 Red de Petri de la gestión de energía y recarga de baterías.....	79
3.2.4.4 Programación y visualización	81
3.3 Proyecto 3: Automatización de la seguridad operativa y mantenimiento general en buques navales.	83
3.3.1 Descripción del sistema	83
3.3.2 Módulo 1: Sistema de detección y extinción de incendios.....	83
3.3.2.1 Enunciado.....	83
3.3.2.2 Tabla de variables.....	84
3.3.2.3 Red de Petri	85
3.3.2.4 Programación y visualización	88
3.3.3 Módulo 2: Sistema de conteo de personas y control de puertas.	91
3.3.3.1 Enunciado.....	92
3.3.3.2 Tabla de variables.....	92
3.3.3.3 Red de Petri	93
3.3.3.4 Programación y visualización	96

3.3.4 Módulo 3: Sistema de mantenimiento automatizado.....	99
3.3.4.1 Enunciado.....	99
3.3.4.2 Tabla de variables.....	100
3.3.4.3 Red de Petri	100
3.3.4.4 Programación y visualización	102
4 Conclusiones y líneas futuras	105
5 Bibliografía.....	107
Anexo I: Diccionario de Siglas, acrónimos y abreviaturas	109
Anexo II: Implicaciones Sociales, Económicas, y Ambientales	110
Anexo III: Reflexiones Éticas y Sociales	111
Anexo IV: Código del proyecto 1, 2 y 3	112

ÍNDICE DE FIGURAS

Figura 2-1 Representación básica de un Diagrama de Bloques Funcionales [2]	17
Figura 2-2 Ejemplo de la representación de un autómata finito determinista [8]	18
Figura 2-3 Ejemplo de la representación de una red de Petri [11]	19
Figura 2-4 Ejemplo de la representación de un diagrama de estado [16]	21
Figura 2-5 Ejemplo de una representación de <i>bond graphs</i> [22]	22
Figura 3-1 Red de Petri del análisis de gestión y control de un sistema de propulsión híbrido. Fuente propia.	38
Figura 3-2 Programación <i>Codesys</i> del sistema de gestión de propulsión híbrida, bloque HMI.	40
Figura 3-3 Programación <i>Codesys</i> del sistema de gestión de propulsión híbrida, bloque MAIN....	41
Figura 3-4 Programación <i>Codesys</i> del sistema de gestión de propulsión híbrida, bloque OUTPUTS.	42
Figura 3-5 Programación <i>Codesys</i> del sistema de gestión de propulsión híbrida, bloque PLACES.	43
Figura 3-6 Programación <i>Codesys</i> del sistema de gestión de propulsión híbrida, bloque TEMP_COUNT.	43
Figura 3-7 Programación <i>Codesys</i> del sistema de gestión de propulsión híbrida, bloque TRANSITIONS.	44
Figura 3-8 Programación <i>Codesys</i> del sistema de gestión de propulsión híbrida, bloque VISUALIZACIÓN.....	45
Figura 3-9 Representación visual del modelo del sistema de gestión de propulsión híbrida apagado.	46
Figura 3-10 Representación visual del modelo del sistema de gestión de propulsión híbrida encendido, en modo manual, con la turbina de gas y motor eléctrico 1 activado.....	47
Figura 3-11 Representación visual del modelo del sistema de gestión de propulsión híbrida encendido, en modo automático, con una prestación del 75 por ciento de potencia.	47
Figura 3-12 Red de Petri del sistema de gestión del mantenimiento preventivo de la propulsión híbrida.	51
Figura 3-13 Programación <i>Codesys</i> del sistema de gestión de mantenimiento, bloque TRANSITIONS.	52
Figura 3-14 Representación visual del modelo del sistema de gestión de mantenimiento recargando la turbina de gas y el motor eléctrico 2.	53
Figura 3-15 Red de Petri del sistema de gestión de alarmas y fallos (1).	58
Figura 3-16 Red de Petri del sistema de gestión de alarmas y fallos (2).	59
Figura 3-17 Programación <i>Codesys</i> del sistema de gestión de alarmas y fallos, bloque TRANSITIONS_M1.	60
Figura 3-18 Programación <i>Codesys</i> del sistema de gestión de alarmas y fallos, bloque TRANSITIONS_M3.	61

Figura 3-19 Representación visual del sistema de gestión de alarmas y fallos, en espera de la finalización de la recarga de gas en la turbina para comenzar con la reparación de esta.	62
Figura 3-20 Representación visual del sistema de gestión de alarmas y fallos, reparando la turbina.	62
Figura 3-21 Red de Petri de lancha con plataforma de carga automática.	67
Figura 3-22 Programación <i>Codesys</i> del sistema del muelle de carga automatizada, bloque TRANSITIONS_M2 (1).	68
Figura 3-23 Programación <i>Codesys</i> del sistema del muelle de carga, bloque TRANSITIONS_M2 (2).	68
Figura 3-24 Representación visual del sistema del muelle de carga automatizada desactivado.....	69
Figura 3-25 Representación visual del sistema del muelle de carga automatizada, realizando descarga en P3.....	70
Figura 3-26 Red de Petri de la gestión de carga y almacenamiento en el muelle (1).	73
Figura 3-27 Red de Petri de la gestión de carga y almacenamiento en el muelle (2).	74
Figura 3-28 Programación <i>Codesys</i> del sistema de gestión de carga y almacenamiento en el muelle, bloque PLACES_M2.	75
Figura 3-29 Programación <i>Codesys</i> del sistema de gestión de carga y almacenamiento en el muelle, bloque TRANSITIONS_M2.	75
Figura 3-30 Representación visual del sistema de gestión de carga y almacenamiento en el muelle desactivado.....	76
Figura 3-31 Representación visual del sistema de gestión de carga y almacenamiento en el muelle, descargando en A.....	76
Figura 3-32 Representación visual del sistema de gestión de carga y almacenamiento en el muelle, descargando en B.....	77
Figura 3-33 Red de Petri de la gestión de energía y recarga de baterías.....	80
Figura 3-34 Programación <i>Codesys</i> del sistema de gestión de energía y recarga de baterías, bloque TRANSITIONS_M3.....	81
Figura 3-35 Representación visual de la gestión de energía y recarga de baterías, desactivado	81
Figura 3-36 Representación visual de la gestión de energía y recarga de baterías, retirando la plataforma	82
Figura 3-37 Representación visual de la gestión de energía y recarga de baterías, cargando la batería de la grúa.....	82
Figura 3-38 Red de Petri del sistema de detección y extinción de incendios	87
Figura 3-39 Programación <i>Codesys</i> del sistema de detección y extinción de incendios, bloque TRANSITIONS_M1.....	88
Figura 3-40 Programación <i>Codesys</i> del sistema de detección y extinción de incendios, bloque TRANSITIONS_M1.....	89
Figura 3-41 Programación <i>Codesys</i> del sistema de detección y extinción de incendios, bloque PLACES_M1	89
Figura 3-42 Representación visual del sistema de detección y extinción de incendios, desactivado	90

Figura 3-43 Representación visual del sistema de detección y extinción de incendios, modo manual	90
Figura 3-44 Representación visual del sistema de detección y extinción de incendios, automático	91
Figura 3-45 Red de Petri del sistema de control de personas y control de puertas (1)	94
Figura 3-46 Red de Petri del sistema de control de personas y control de puertas (2)	95
Figura 3-47 Programación <i>Codesys</i> del sistema de control de personas y control de puertas, bloque PLACES_M2 (1).....	96
Figura 3-48 Programación <i>Codesys</i> del sistema de control de personas y control de puertas, bloque PLACES_M2 (2).....	96
Figura 3-49 Programación <i>Codesys</i> del sistema de control de personas y control de puertas, bloque PLACES_M2 (3).....	97
Figura 3-50 Programación <i>Codesys</i> del sistema de control de personas y control de puertas, bloque TRANSITIONS_M2 (1)	97
Figura 3-51 Programación <i>Codesys</i> del sistema de control de personas y control de puertas, bloque TRANSITIONS_M2 (2)	98
Figura 3-52 Representación visual del sistema de control de personas y control de puertas, desactivado.....	98
Figura 3-53 Representación visual del sistema de control de personas y control de puertas, activado	99
Figura 3-54 Red de Petri del sistema de mantenimiento automatizado	101
Figura 3-55 Programación <i>Codesys</i> del sistema de mantenimiento automatizado, PLACES_M3 (1)	102
Figura 3-56 Programación <i>Codesys</i> del sistema de mantenimiento automatizado, PLACES_M3 (2)	103
Figura 3-57 Programación <i>Codesys</i> del sistema de mantenimiento automatizado, PLACES_M3 (3)	103
Figura 3-58 Representación visual del sistema de mantenimiento automatizado, desactivado	104
Figura 3-59 Representación visual del sistema de mantenimiento automatizado, desactivado	104

ÍNDICE DE TABLAS

Tabla 2-1 Comparativa entre métodos de representación de flujos en sistemas.	23
Tabla 2-2 Comparativa entre lenguajes para implementación de redes de Petri.....	27
Tabla 3-1 Variables del análisis de gestión y control de un sistema de propulsión híbrido.....	37
Tabla 3-2 Variables del sistema de gestión de mantenimiento preventivo.	50
Tabla 3-3 Variables del sistema de gestión de alarmas y fallos.	56
Tabla 3-4 Variables de lancha con plataforma de carga automática.	66
Tabla 3-5 Variables de la gestión de carga y almacenamiento en el muelle.....	72
Tabla 3-6 Variables del sistema de gestión de energía y recarga de baterías.....	79
Tabla 3-7 Variables del sistema de detección y extinción de incendios	85
Tabla 3-8 Variables del sistema de control de personas y control de puertas.....	93
Tabla 3-9 Variables del sistema de mantenimiento automatizado	100
Tabla 0-1 Siglas, acrónimos y abreviaturas.....	109
Tabla 0-1 Códigos de los distintos proyectos.....	112

1 INTRODUCCIÓN Y OBJETIVOS

1.1 Contexto y motivación

La automatización y el control de sistemas juegan un papel fundamental en la optimización de procesos industriales, mejorando la eficiencia, la seguridad y la fiabilidad de las operaciones. En el ámbito naval, estos aspectos adquieren una relevancia aún mayor, dado que, por ejemplo, la correcta gestión de los sistemas de propulsión, energía y mantenimiento es crucial para garantizar el óptimo desempeño de las embarcaciones.

En el marco del plan de estudios del Grado en Ingeniería Mecánica, título oficial de la Universidad de Vigo, que se imparte en el Centro Universitario de la Defensa (CUD-ENM) en la Escuela Naval Militar, la asignatura de Automática constituye un pilar esencial para la formación de futuros oficiales de la Armada. Con el objetivo de fortalecer sus competencias en automatización, control de procesos y programación de sistemas, se ha diseñado un conjunto de prácticas orientadas a proporcionar un aprendizaje aplicado y alineado con distintos ámbitos dentro de la Armada.

Este Trabajo de Fin de Grado se embarca dentro de esta línea formativa, abordando el desarrollo de proyectos virtuales de automatización mediante el uso de redes de Petri y lenguaje *Ladder*, dos herramientas fundamentales en la modelización y programación de sistemas industriales. Las redes de Petri permiten representar, analizar y validar el comportamiento de los sistemas dinámicos, mientras que el lenguaje *Ladder*, ampliamente empleado en la programación de autómatas programables, posibilita la implementación práctica de dichos modelos en industriales.

Dado el creciente interés en la automatización dentro de la industria naval y la necesidad de contar con sistemas cada vez más eficientes y seguros, este trabajo busca proporcionar un marco metodológico para el diseño, simulación y validación de sistemas de control aplicados a entornos académicos y profesionales. La modelización previa mediante herramientas virtuales facilita la reducción de errores, optimiza tiempos de desarrollo y mejora la seguridad operativa, lo que representa un avance significativo tanto en la industria como en el ámbito educativo.

El uso de entornos virtuales en la enseñanza de automatización ofrece una alternativa eficaz y accesible a la experimentación con hardware físico, permitiendo a los alumnos interactuar con sistemas de control en escenarios simulados. De esta manera, este proyecto no solo contribuye al desarrollo de competencias técnicas, sino que también fomenta la integración de conocimientos de ingeniería mecánica con los procesos y terminología específicos del ámbito naval.

1.2 Objetivos

El presente Trabajo de Fin de Grado tiene como finalidad el desarrollo de un conjunto de proyectos de automatización bien estructurados y documentados, orientados al aprendizaje práctico de los alumnos del CUD-ENM. Estos proyectos están diseñados para fortalecer la formación de la asignatura de Automática, impartida en el cuarto curso del grado de Ingeniería Mecánica, con especial enfoque en su aplicación en el ámbito naval.

El objetivo general de este trabajo es la creación y documentación de proyectos prácticos de automatización, utilizando redes de Petri y lenguaje *Ladder*, herramientas ampliamente utilizadas en la modelización y programación de sistemas automatizados. A través de esta integración, se busca proporcionar una base metodológica sólida que permita a los estudiantes desarrollar competencias técnicas en automatización, control de procesos y programación de sistemas industriales, con una aplicación directa en los sistemas de la Armada.

Para lograr este propósito, se establecen unos objetivos específicos, el primero de ellos desarrollar un conjunto de procesos virtuales de automatización, aplicados a procesos industriales navales, que sirvan como base para la enseñanza de la asignatura de Automática y permitan la simulación de estrategias de control sin necesidad de hardware físico. Además, también el diseño y la implementación de modelos de control utilizando redes de Petri y lenguaje *Ladder*, facilitando la representación, análisis y simulación de sistemas automatizados. Proporcionar una metodología estructurada para el diseño y validación para sistemas de control, reduciendo errores, optimizando tiempos de desarrollo y mejorando la seguridad operativa. Integrar conceptos de ingeniería mecánica con procesos y terminología específica del ámbito naval, asegurando que los alumnos adquieran conocimientos adaptados a su futura labor en la Armada. Recrear situaciones aproximadas a la realidad en entornos virtuales, permitiendo dentro de lo que cabe a los estudiantes, experimentar con sistemas de control similares a los que encontrarán en su ejercicio profesional, fomentando así una comprensión más profunda y un aprendizaje aplicado. Y, por último, fortalecer las competencias en programación, modelización y control de sistemas automatizados, proporcionando herramientas que contribuyan a la preparación de los alumnos para los desafíos tecnológicos presentes en el sector naval y la industria en general.

A través de estos objetivos, este trabajo no solo busca mejorar el proceso de enseñanza-aprendizaje en el ámbito de la automatización, sino también contribuir a la formación de oficiales con una preparación técnica avanzada en tecnologías clave para la Armada.

1.3 Estructura de la memoria

Tras hacer referencia al marco en el que se encuadra el trabajo, contextualizándolo y tras haber definido los objetivos que se esperan alcanzar, se pasa a explicar la estructura que seguirá la presente memoria con el fin de organizar la presentación de los trabajos realizados, así como los resultados obtenidos, de la forma más clara posible. El documento está dividido en cinco capítulos a los que a la finalización de estos incluye la bibliografía que recoge todas las fuentes empleadas y los anexos que servirán de apoyo a lo largo del desarrollo del trabajo. Este trabajo se organiza como sigue:

Capítulo 1: Introducción y objetivos. En este primer apartado se expone el contexto y la motivación que han llevado al desarrollo de este trabajo, resaltando la importancia de la automatización en el ámbito naval y su aplicación en la formación de los alumnos de la Escuela Naval Militar. Asimismo, se definen los objetivos generales y específicos del trabajo, estableciendo las bases teóricas y prácticas sobre las cuales se desarrollará la investigación.

Capítulo 2: Estado del arte. Este capítulo presenta una revisión bibliográfica y un análisis del estado actual de la automatización industrial, con especial énfasis en el uso de redes de Petri y lenguaje *Ladder* en la modelización y control de sistemas. Además, se incluyen

referencias a aplicaciones similares en el sector naval e industrial, elevando sus ventajas y limitaciones. Así como investigación sobre los distintos métodos de representación de flujos de sistemas y los distintos lenguajes existen entes para representar las redes de Petri.

- **Capítulo 3: Desarrollo del trabajo.** En este apartado se describen en detalle los proyectos desarrollados en el trabajo, abordando tanto los aspectos técnicos como la metodología utilizada. Se presenta el diseño y la implementación de sistemas automatizados mediante redes de Petri y lenguaje *Ladder*, así como las estrategias utilizadas para su simulación y validación. Los proyectos específicos incluidos en este capítulo son los siguientes:
 - **Proyecto 1: Sistemas de propulsión híbrido en barcos.** Los diferentes módulos en los que se compone son:
 - Módulo 1: Analiza la gestión y control de un sistema de propulsión híbrido, considerando la transición entre diferentes modos de operación (eléctrico, gas e híbrido) y la optimización del consumo energético.
 - Módulo 2: Gestión del mantenimiento preventivo en sistemas navales. Se desarrolla un modelo de control orientado a la supervisión y mantenimiento de sistemas de propulsión, con especial atención en la prevención de fallos y la optimización de los recursos disponibles.
 - Módulo 3: Gestión de alarmas y fallos en sistemas de propulsión híbrida. Se diseña un sistema de supervisión y diagnóstico de fallos basado en redes de Petri interconectadas, con el fin de optimizar la seguridad y fiabilidad del sistema. Se integran técnicas de detección de fallos en motores eléctricos y turbinas de gas, además de estrategias para la corrección automatizada de anomalías en la operación del sistema.
 - **Proyecto 2: Muelle de carga automatizada.** Los diferentes módulos en los que se compone son:
 - Módulo 1: Implementación de una lancha con plataforma de carga automática. Se diseña un sistema de carga interno que optimiza el proceso de embarque de mercancías sin intervención manual.
 - Módulo 2: Gestión de la carga y almacenamiento en el muelle automatizado. Se modela el flujo de carga desde un almacén central hasta las embarcaciones, asegurando una distribución eficiente de los materiales mediante una plataforma robótica autónoma.
 - Módulo 3: Gestión de energía y recarga de baterías. Se desarrolla un sistema para garantizar la autonomía operativa de las plataformas robóticas y de la lancha automatizada, incorporando estaciones de carga y protocolos de optimización de energía.
 - **Proyecto 3: Plan de seguridad de un barco.** Los diferentes módulos en los que se compone son:
 - Módulo 1: Sistema de detección y extinción de incendios. Se implementa un sistema automatizado para detectar tempranamente situaciones de incendio, activando mecanismos automáticos de extinción. Se utiliza una red de Petri para modelar situaciones críticas y transiciones entre diferentes estados de alerta, permitiendo tanto un funcionamiento manual como automático del sistema.
 - Módulo 2: Sistema de conteo de personas y control de puertas. Se desarrolla un sistema automatizado encargado de monitorizar continuamente el número de personas a bordo mediante sensores de entrada y salida, así como la gestión automatizada de puertas de acceso, garantizando una adecuada gestión de la seguridad y prevención de riesgos. Las redes de Petri se emplean para gestionar y controlar estados y transiciones relacionadas con accesos, restricciones y emergencias.

- **Módulo 3: Sistema de mantenimiento automatizado.** Se implementa un sistema que realiza diagnósticos y planificación automática del mantenimiento general del buque, utilizando redes de Petri para simular y gestionar las operaciones de mantenimiento periódicas y predictivas. El objetivo es optimizar los recursos, prevenir averías críticas y mejorar la seguridad operativa del buque mediante intervenciones automatizadas.
- **Capítulo 4: Conclusiones y líneas futuras.** Se realiza un resumen global del trabajo y se incluyen líneas futuras sobre investigaciones con inteligencia artificial y plataformas avanzadas
- **Capítulo 5: Bibliografía.** Listado completo de referencias utilizadas para fundamentar el desarrollo del trabajo.
- **Capítulo 6: Anexos.** Información adicional sobre siglas, implicaciones éticas, sociales, ambientales y códigos fuente desarrollados.

2 ESTADO DEL ARTE

2.1 Métodos de representación del flujo en sistemas

En el ámbito de la programación y control de sistemas, la representación del flujo de información y procesos es fundamental para garantizar una gestión eficiente y precisa. Existen diversas metodologías que permiten modelar estos sistemas, proporcionando herramientas gráficas y matemáticas para su análisis y optimización, tanto en mecánica, electrónica y termodinámica. Este apartado revisa las principales técnicas utilizadas en la representación de flujos cada uno con sus ventajas, desventajas y ámbito de aplicación.

2.1.1 Diagramas de bloques funcionales

Los diagramas de bloques funcionales (FBD, *Función Block Diagram*) son un lenguaje de programación gráfico utilizado en sistemas de automatización industrial, particularmente en controladores lógicos programables (PLC, *Programmable Logic Controller*). Se basa en la representación de funciones mediante bloques interconectados que modelan la lógica del sistema [1]. Facilitan la comprensión de las interacciones entre diferentes componentes del sistema al representar las relaciones funcionales mediante bloques conectados por flechas, que indican la dirección del flujo de información [2].

Este método de representación, definido en la norma IEC 61131-3 [3], se utiliza para simplificar la interpretación de los circuitos eléctricos y electrónicos, permitiendo visualizar la función y posición relativa de cada componente sin necesidad de un diagrama detallado [2]. Además, los diagramas de bloques permiten derivar ecuaciones de estado a partir de ecuaciones diferenciales ordinarias facilitando la simulación numérica de números continuos, es decir, ayudan a convertir ecuaciones diferenciales en ecuaciones de estado, lo que permite analizar y simular sistemas dinámicos de manera más sencilla usando métodos numéricos en computadoras, que es clave en la automatización, control de procesos e ingeniería de sistemas [4]. La descomposición, o modularidad de estos diagramas facilita su reutilización en distintas aplicaciones, lo que los hace ideales para el diseño de sistemas de control industrial. No obstante, su aplicabilidad en sistemas con concurrencia avanzada es limitada, esto se debe a que cada subsistema debe considerar sus variables de interacción como entrada o salida, lo que puede generar conflictos de causalidad, es decir, problemas de sincronización temporal, cuando se intenta acoplar subsistemas, y por ello se ha impulsado la búsqueda de modelos más robustos capaces de gestionar múltiples procesos simultáneamente [5].



Figura 2-1 Representación básica de un Diagrama de Bloques Funcionales [2]

Como se ha explicado anteriormente, este tipo de diagramas se representa “*mediante bloques conectados por flechas, que indican la dirección del flujo de información*”. Su representación simbólica se basa en, por un lado, los bloques funcionales que se representan como rectángulos que simbolizan funciones lógicas o matemáticas, y cada bloque tiene entradas (a la izquierda) y salidas (a la derecha), se puede apreciar en la Figura 2-1. Estos bloques pueden ser, por ejemplo, un *AND*, *OR*, temporizadores, comparadores, sumadores, etc. Y, por otro lado, están las flechas que son la conexión entre los bloques y representan el flujo de datos y la dirección del proceso, que siempre es de izquierda a derecha. Cabe comentar que se pueden conectar varias entradas a un bloque, y una o varias salidas. Las entradas (a la izquierda) reciben información de sensores, botones y señales externas, sin embargo, las de salida (derecha) activan motores, luces, válvulas u otros procesos. También existen en este tipo de diagramas los círculos, o mejor dicho los puntos de conexión o estado, estos representan puntos de unión donde varias señales convergen, también pueden indicar variables de estado dentro de la lógica de control. Por último, también cabe mencionar los temporizadores y contadores, que se representan también como bloques rectangulares con parámetros internos.

2.1.2 Autómatas finitos

Los autómatas finitos constituyen una metodología matemática empleada en la modelización de sistemas secuenciales o dinámicos discretos. Permiten describir la evolución de estados en función de eventos discretos. Su estructura está basada en la existencia de estados definidos y transiciones entre ellos, activadas por eventos preestablecidos [6].

Esta técnica es ampliamente utilizada en el diseño de protocolos de comunicación y análisis de lenguajes formales, proporcionando un modelo riguroso para la gestión de procesos secuenciales. Gracias a su precisión y fiabilidad, los autómatas finitos son esenciales en aplicaciones donde la ejecución ordenada de eventos es prioritaria, como en sistemas de control de tráfico, autenticación de protocolos de seguridad y automatización de procesos en la industria manufacturera. Sin embargo, debido a sus limitaciones en la representación de concurrencia y sincronización de eventos, los autómatas finitos presentan dificultades al abordar sistemas complejos de automatización industrial que requieren la ejecución paralela de múltiples procesos [7].

Existen dos tipos de autómatas finitos, por un lado, los autómatas finitos deterministas (AFD) son un modelo matemático de computación que consiste en un conjunto finito de estados, un alfabeto de entrada, una función de transición que asigna un único estado siguiente a cada par formado por un estado y un símbolo de entrada, un estado inicial y un conjunto de estados finales. Su principal característica es que, dada una entrada específica, el autómata siempre sigue una única secuencia de estados bien definida, lo que lo hace completamente predecible y adecuado para modelar procesos que requieren una ejecución secuencial sin ambigüedad, un ejemplo es la Figura 2-2.

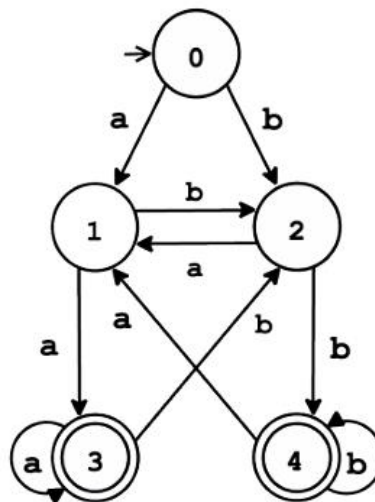


Figura 2-2 Ejemplo de la representación de un autómata finito determinista [8]

Por otro lado, un autómata finito no determinista (AFND) comparte la misma estructura general que el AFD, pero su función de transición permite que, para un mismo estado y símbolo de entrada, exista múltiples estados posibles a los que pueda transitar. En algunos casos, incluso puede realizar una transición sin consumir ningún símbolo de entrada. Esto significa que un AFND puede explorar diferentes caminos simultáneamente dentro de su estructura de estados, lo que lo hace útil en la representación de problemas de búsqueda y reconocimiento de lenguajes donde la incertidumbre o la ambigüedad están presentes.

La principal diferencia entre ambos tipos de autómatas radica en su capacidad de transición. Mientras que un AFD siempre tiene un único estado destino para cada combinación de estado y símbolo de entrada, un AFND puede tener múltiples opciones o incluso ninguna. A pesar de esta diferencia, se ha demostrado que ambos poseen la misma capacidad computacional, ya que cualquier AFND puede transformarse en un AFD equivalente, aunque con un posible incremento exponencial en el número de estados. Sin embargo, los AFND suelen ser más compactos y expresivos para el diseño y la especificación de lenguajes, aunque su implementación práctica requiere una conversión AFD.

Aunque los autómatas finitos son herramientas poderosas para el modelado de sistemas discretos, su aplicación en sistemas dinámicos continuos es limitada, lo que ha llevado a su integración con otras metodologías como las redes de Petri y los sistemas híbridos para mejorar su aplicabilidad en entornos más complejos [7].

Los elementos para su representación son: los estados, las transiciones, y las reglas de transición. Los estados están representados por círculos etiquetados con un identificador único, el estado inicial se denota con una flecha de entrada sin origen apuntando hacia él, los estados finales se representan con un doble círculo. En cuanto a las transiciones, estas son representadas con flechas dirigidas entre los estados, etiquetadas con los símbolos del alfabeto de entrada que provocan dichas transiciones. Y por último las reglas de transición, por ejemplo, si una transición lleva de un estado “q” a otro estado “q’” bajo la entrada “a”, se representa como una función:

$$f(q, a) = q'$$

Esto se representa con una flecha desde “q” hacia “q’”, con la etiqueta “a” sobre la flecha [9].

Para representar un autómata finito de manera efectiva, es esencial seguir una metodología sistemática que garantice precisión y claridad en su diseño. Para ello primeramente conviene llevar a cabo una identificación de condiciones relevantes, es decir, tratar de determinar las condiciones o situaciones clave que el autómata debe reconocer o procesar. Cada una de estas condiciones se asociará a un estado específico del autómata. Seguidamente la asignación de estados, que consiste en asociar cada

condición identificada en el paso anterior a un estado único del autómata. Es fundamental que cada estado represente una condición claramente definida y distinta. Se continua con la definición de transiciones, estableciendo las reglas de transición entre los estados en función de los símbolos de entrada. Para cada estado se determina como responde el autómata a cada símbolo del alfabeto, especificando el estado al que debe transitar. Por último, se determinan los estados iniciales y finales seleccionando el estado desde el cual el autómata comienza su procesamiento. Asimismo, se identifican los estados finales o de aceptación que indican cuándo una cadena de entrada es aceptada por el autómata. Este enfoque estructurado facilita el diseño de autómatas finitos, asegurando que cada estado y transición estén claramente definidos y que el autómata funcione de manera coherente y precisa [7].

2.1.3 Redes de Petri

Las redes de Petri han emergido como una alternativa eficiente para modelar sistemas dinámicos con concurrencia y sincronización, es decir una herramienta gráfica y matemática utilizada para modelar sistemas concurrentes y eventos discretos [10]. A diferencia de otros sistemas, las redes de Petri están más enfocadas en sistemas continuos, son especialmente útiles en sistemas en los que los eventos ocurren de manera asíncrona y en paralelo.

En el contexto de sistemas dinámicos, las redes de Petri permiten representar la dinámica de procesos industriales, logísticos y de comunicaciones, proporcionando una representación clara del flujo de eventos y de los estados del sistema. Un aspecto clave de este método es su capacidad para manejar estructuras complejas con restricciones lógicas, facilitando el análisis de rendimiento y la verificación de propiedades como la vivacidad y la ausencia de bloqueos [10].

Esta herramienta matemática se basa en una estructura visual de lugares, transiciones y *tokens*, permitiendo representar con precisión el comportamiento de sistemas distribuidos y como se ha mencionado anteriormente, concurrentes. Su flexibilidad ha hecho que sean ampliamente aplicadas en la automatización industrial, el modelado de sistemas de transporte y la optimización de procesos informáticos. En comparación con otros métodos de modelización, las redes de Petri permiten una mayor capacidad de análisis, facilitando la identificación de cuellos de botella, la optimización del rendimiento del sistema y la predicción de fallos operativos. Además, las herramientas de simulación y validación basadas en redes de Petri han evolucionado significativamente, proporcionando modelos más detallados y precisos para la industria de la automatización [10].

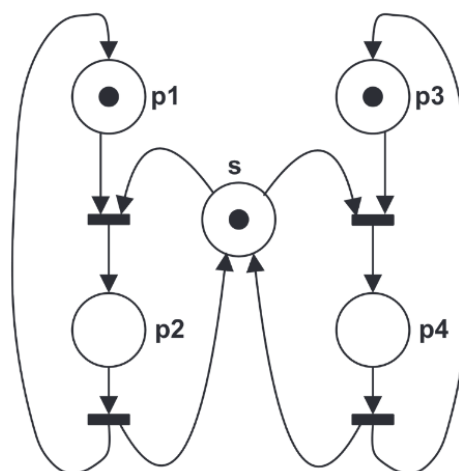


Figura 2-3 Ejemplo de la representación de una red de Petri [11]

Desde una perspectiva gráfica, una red de Petri se compone de varios elementos fundamentales. Los lugares, representados mediante círculos, simbolizan estados o condiciones del sistema. Estos lugares

pueden contener *tokens*, los cuales representan la disponibilidad de recursos o el estado actual del sistema en un momento dado. Por otro lado, las transiciones, representadas como rectángulos, denotan eventos o acciones que pueden modificar dichos estados. Para que una transición se active, debe verificarse la condición de habilitación, es decir, que todos los lugares de entrada contengan la cantidad suficiente de *tokens*. Una vez satisfecha esta condición, la transición se dispara, consumiendo los *tokens* de los lugares de entrada y generando nuevos *tokens* en los lugares de salida, alterando así el estado del sistema [10].

El comportamiento dinámico de una red de Petri se define a través de los arcos dirigidos, representados por flechas, los cuales establecen la relación entre lugares y transiciones. Estos arcos determinan la estructura del sistema y el flujo de eventos, permitiendo modelar escenarios en los que múltiples transiciones pueden ejecutarse simultáneamente si sus condiciones de habilitación se cumplen de manera independiente. Este aspecto es especialmente relevante en el modelado de sistemas industriales y de automatización, donde diversas operaciones pueden ejecutarse en paralelo y deben ser coordinadas para garantizar un funcionamiento eficiente y seguro.

En síntesis, las redes de Petri constituyen un modelo formal eficaz para la representación y análisis de sistemas con concurrencia y sincronización de eventos. Su capacidad para modelar la evolución de los estados del sistema en función de la activación de eventos las convierte en una herramienta valiosa en el diseño, validación, simulación y optimización de procesos automatizados, asegurando un control eficiente y una gestión precisa de recursos disponibles.

2.1.4 Diagramas de estado

Los diagramas de estado también conocidos como diagramas de transición de estados o máquinas de estados, son herramientas de modelado utilizadas en el desarrollo de *software* y sistemas. Su objetivo principal es representar los diferentes estados por los que pasa un objeto, clase o sistema a lo largo de su ciclo de vida, así como los eventos que provocan los cambios de estado. Estos diagramas son particularmente útiles en el modelado de sistemas reactivos, es decir, aquellos que responden a eventos externos o internos. Se emplean en diversos ámbitos de la informática, tales como sistemas en tiempo real, por ejemplo, *software* para monitorización de signos vitales, dispositivos electrónicos como cajeros automáticos o electrodomésticos inteligentes, videojuegos donde se modelan los comportamientos de personajes y entornos y sistemas basados en protocolos, tales como redes de comunicación y protocolos de seguridad [12].

Un diagrama de estado describe los estados posibles de un sistema y las transiciones entre ellos. Cada transición se activa por un evento y puede estar condicionada por ciertos factores. Los componentes básicos de un diagrama de estado son el estado, que representa una condición en la que se encuentra el sistema en un momento determinado, la transición, que indica el cambio de un estado a otro desencadenado por un evento, el evento que es el suceso que provoca una transición entre estados, la condición de guarda que es una expresión booleana que determina si la transición puede ejecutarse, y la acción que es la operación que se lleva a cabo al realizar una transición [12], [13].

Los diagramas de estado utilizan una notación estandarizada que incluye el estado inicial, representado por un círculo relleno que indica el punto de partida del sistema; el estado final, representado por un círculo relleno dentro de otro círculo que señala la terminación del proceso; los estados, representados por rectángulos redondeados que contienen el nombre del estado y, opcionalmente acciones internas; las transiciones, representadas por flechas que indican el cambio de un estado a otro; los eventos, que son etiquetas en las transiciones que especifican los sucesos que provocan cambios de estado, y las acciones en transiciones que se representan con sintaxis: evento/acción [14], [15]. En la Figura 2-4 se puede ver de manera representada todo lo que acaba de ser comentado.

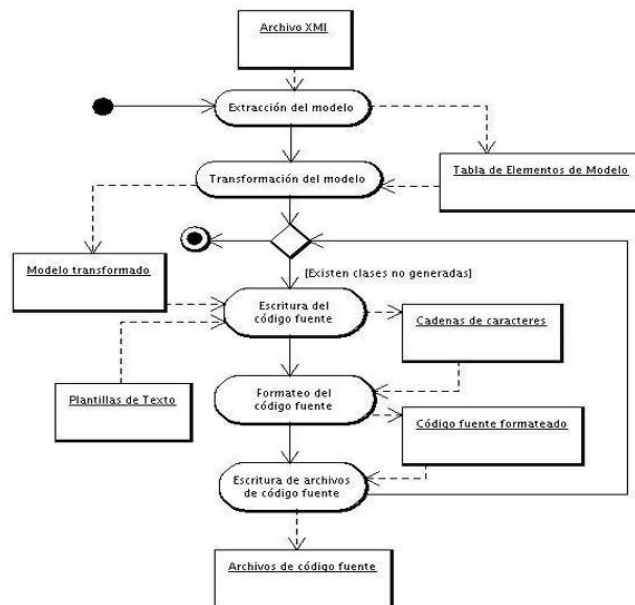


Figura 2-4 Ejemplo de la representación de un diagrama de estado [16]

Existen diversas variaciones de diagramas de estado [13], entre ellas el diagrama de estados simple que describe estados y transiciones de manera lineal, el diagrama de estados jerárquico que permite la anidación de estados dentro de otros estados para una mejor organización y el diagrama de estados paralelos que modela la concurrencia de varios estados simultáneos.

El uso de diagramas de estado ofrece diversos beneficios, como la claridad en la representación del comportamiento del sistema, la reducción de ambigüedades en el diseño de *software*, la facilidad para detectar errores en el flujo de estados y la documentación visual que mejora la comprensión del sistema. Sin embargo, también presentan algunas limitaciones ya que la complejidad crece exponencialmente en sistemas grandes, pueden volverse inmanejables sin un diseño adecuado y no siempre son la mejor herramienta para sistemas que no dependen de eventos. En conclusión, los diagramas de estado son herramientas fundamentales en el modelado de sistemas dinámicos y reactivos. Su aplicación permite una mejor organización y entendimiento del comportamiento del sistema, favoreciendo un desarrollo de *software* estructurado y eficiente. Sin embargo, es necesario un diseño cuidadoso para evitar diagramas excesivamente complejos y difíciles de interpretar [12].

2.1.5 Bond graphs

Los *bond graphs* (o diagramas de vínculos) constituyen una metodología del modelado gráfico utilizada para representar el comportamiento dinámico de sistemas físicos en múltiples dominios, tales como la mecánica, electricidad, hidráulica, térmica y acústica. Su principal ventaja radica en la posibilidad de describir estos sistemas de manera unificada a través del flujo de energía, facilitando así la identificación de interacciones y la derivación de ecuaciones matemáticas asociadas [17].

El enfoque de los *bond graphs* se basa en la representación de los sistemas físicos mediante enlaces denominados *bonds*, los cuales establecen una conexión entre los componentes que intercambian energía. Cada uno de estos enlaces transporta dos variables conjugadas, esfuerzo y flujo, cuyo producto representa la potencia transferida en el sistema. Esta propiedad permite modelar sistemas multidominio de manera homogénea, proporcionando una herramienta robusta para la simulación y el análisis de sistemas dinámicos complejos [18].

Los elementos fundamentales en la representación mediante *bond graphs* incluyen distintos tipos de componentes, cada uno de los cuales desempeñan una función específica en el intercambio de energía.

En primer lugar, los elementos de almacenamiento de energía comprenden los componentes de capacitancia (C), que almacenan energía en forma de esfuerzo acumulado, como los capacitores en circuitos eléctricos o los resortes en sistemas mecánicos, e inercia (I), que almacenan energía en forma de flujo acumulado, como las masas en mecánica o los inductores en circuitos eléctricos [17]. En segundo lugar, los elementos resistivos, representados por resistencias (R), disipan energía a lo largo del sistema, como ocurre con las resistencias eléctricas o los amortiguadores mecánicos. Además, existen las fuentes de energía o esfuerzo fuente (Se), que representan fuentes de esfuerzo constante, como una fuente de corriente o un caudal de fluido en un sistema hidráulico [19].

Otros elementos clave en los *bond graphs* incluyen los transformadores y giradores, que permiten la conversión de energía. Un transformador establece una relación proporcional entre esfuerzos y flujos en sistemas con relaciones fijas, como engranajes en mecánica o transformadores en electricidad, mientras que un girador convierte esfuerzos en flujos y viceversa, facilitando la conversión de energía entre dominios, como ocurre en los motores eléctricos que transforman voltaje en torque.

Uno de los aspectos más relevantes en la modelización con *bond graphs* es la causalidad, la cual define cómo se propaga la información dentro del sistema y es fundamental para la correcta derivación de ecuaciones matemáticas. Existen distintos tipos de causalidad: causalidad fija, que se aplica en fuentes (Se), donde la variable de salida está determinada de antemano; causalidad preferida, en los elementos de almacenamiento (C , I), donde se prioriza la integración numérica en lugar de la diferenciación para evitar inestabilidad en el sistema; causalidad indiferente, en resistencias (R), donde tanto el flujo como el esfuerzo pueden ser determinados indistintamente, y causalidad restringida, que se presenta en las *0-Junctions* y *1-Junctions*, donde la causalidad de un enlace impone restricciones en los demás [20].

El uso de *bond graphs* permite el modelado eficiente de sistemas dinámicos complejos en distintos ámbitos de la ingeniería. Por ejemplo, en representaciones resistencia, inductancia y capacitancia (RLC), los voltajes representados como esfuerzos y las corrientes como flujos. En sistemas mecánicos, un modelo que combine una masa, un resorte y un amortiguador puede representarse con los elementos I (inercia), C (constante elástica) y R (amortiguación), estableciendo conexiones mediante una *1-Junction* para indicar la velocidad común entre los componentes [20], [21].

En la Figura 2-5 se puede observar un ejemplo.

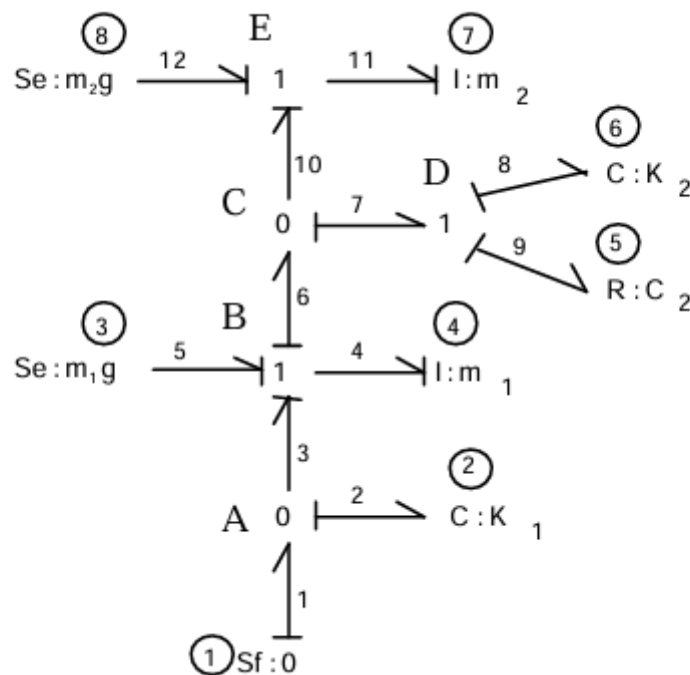


Figura 2-5 Ejemplo de una representación de *bond graphs* [22]

En conclusión, los *bond graphs* constituyen una herramienta fundamental para la modelización y simulación de sistemas físicos en distintos dominios de la ingeniería. Su enfoque basado en el flujo de energía permite la integración de múltiples disciplinas en una única representación gráfica, facilitando tanto el análisis como la implementación de soluciones en automatización, robótica, ingeniería mecánica, eléctrica y otras áreas afines. La capacidad de estos modelos para representar de manera precisa las interacciones dinámicas entre componentes y su aplicabilidad en el diseño y control de sistemas avanzados demuestran su relevancia y utilidad en ingeniería moderna.

2.1.6 Comparación entre métodos y justificación del uso de redes de Petri

En la Figura 2-1 se puede observar una comparativa entre los distintos tipos de métodos de representación del flujo en sistemas mencionados y explicados anteriormente.

Método	Ventajas	Desventajas	Aplicación en automatización
Diagramas de bloques funcionales	Intuitivos, fáciles de entender, visuales	Poca formalidad matemática, difíciles de analizar en sistemas complejos	Modelado de sistemas industriales y de control
Autómatas finitos	Buena formalización matemática, útiles para sistemas secuenciales	Crecen exponencialmente en complejidad, difíciles de escalar	Control de procesos industriales y sistemas de comunicación
Redes de Petri	Análisis riguroso, buena representación del paralelismo	Complejidad matemática alta, difícil interpretación visual en grandes sistemas	Modelado y análisis de sistemas concurrentes
Diagramas de estado	Claridad en la representación del comportamiento del sistema	Difícil de manejar en sistemas grandes con muchas transiciones	Modelado de sistemas embebidos y <i>software</i>
<i>Bond graphs</i>	Representación unificada de sistemas físicos	Requieren conocimientos avanzados en modelado físico	Modelado de sistemas dinámicos y energéticos

Tabla 2-1 Comparativa entre métodos de representación de flujos en sistemas.

2.2 Lenguajes para la implementación de redes de Petri

A lo largo del tiempo, las redes de Petri han evolucionado desde un modelo matemático teórico hacia una herramienta fundamental en el diseño y control de sistemas automatizados. Para su implementación en entornos reales, es necesario utilizar lenguajes de programación específicos que permitan la simulación, análisis y ejecución de estos modelos.

Uno de los lenguajes más utilizados en la automatización industrial es el Lenguaje *Ladder*, ampliamente empleado en contadores lógicos programables. Sin embargo, existen otras alternativas que amplían las capacidades de las redes de Petri, permitiendo una mayor flexibilidad y aplicabilidad en distintos entornos.

En esta sección, además del lenguaje *Ladder*, se presentan los principales lenguajes utilizados para la implementación de redes de Petri, analizando sus características, ventajas y casos de uso en sistemas de automatización y control.

2.2.1 Lenguaje *Ladder*

El lenguaje *Ladder* (LD), también conocido como *Ladder Diagram*, es un lenguaje de programación ampliamente utilizado en la automatización industrial para programar PLCs [23]. Este lenguaje se inspira en los diagramas eléctricos de relés, facilitando su comprensión a ingenieros y técnicos familiarizados con los esquemas de circuitos eléctricos [24]. Su estructura está basada en contactos y bobinas que representan entradas, salidas y operaciones lógicas dentro del sistema de control, permitiendo así la implementación de automatismos secuenciales y combinacionales.

El propósito principal del LD es la programación eficiente de PLCs en la industria, proporcionando un medio gráfico y visual para diseñar sistemas de control. Es utilizado en diversas aplicaciones industriales, como la automatización de procesos de manufactura, el control de maquinaria, la gestión de sistemas de transporte y la regulación de sistemas eléctricos complejos [24]. Su enfoque secuencial permite representar de manera clara el flujo de operaciones dentro de un proceso automatizado, lo que facilita el mantenimiento y la depuración de los programas [23].

Una de las principales ventajas del LD radica en su facilidad de aprendizaje y su similitud con los esquemas de control eléctrico tradicionales, lo que reduce la curva de aprendizaje para los operadores y técnicos encargados de la programación y el mantenimiento [23]. Asimismo, su representación gráfica permite una identificación intuitiva de los estados del sistema y de los posibles fallos, lo que resulta en una optimización del tiempo de desarrollo y puesta en marcha de los automatismos. Además, el uso de este lenguaje facilita la documentación del proceso de diseño y programación, garantizando una mayor trazabilidad entre el diseño del sistema y su implementación en el PLC [24].

No obstante, el LD también presenta ciertas limitaciones. Su simplicidad estructural puede dificultar la implementación de sistemas que requieran la gestión de eventos complejos, como la concurrencia, el sincronismo y las secuencias avanzadas. Además, el crecimiento en la complejidad de los sistemas automatizados ha generado la necesidad de métodos formales de diseño que reduzcan la dificultad de programación y permitan optimizar los algoritmos de control.

En este contexto, se han desarrollado metodologías como el modelado de automatismos secuenciales utilizando la teoría binodal. Esta teoría permite obtener ecuaciones de manera optimizada, lo que facilita la implementación de algoritmos en LD sin recurrir a procedimientos que dependen de la experiencia del programador. El uso de metodologías formales, como la teoría binodal, contribuye a la reducción del tiempo de desarrollo y los errores de implementación, ofreciendo mayor confiabilidad en los sistemas automatizados [23].

En conclusión, el LD sigue siendo una herramienta esencial en la automatización industrial debido a su facilidad de uso y compatibilidad con los estándares de programación PLC. No obstante, la creciente complejidad de los sistemas automatizados ha motivado la búsqueda de enfoques complementarios que permitan superar sus limitaciones. La integración de estos enfoques es el diseño de sistemas de control contribuye a la mejora de la eficiencia y fiabilidad de los procesos industriales.

2.2.2 Redes de Petri de alto nivel

Las redes de Petri de alto nivel (HLPN) representan una evolución del formalismo clásico de las redes de Petri, integrando estructuras matemáticas avanzadas para el modelado de sistemas complejos. Este formalismo ha sido ampliamente estudiado y aplicado en diversas áreas como la automatización, la gestión de redes de comunicación y la optimización de procesos industriales [25], [26].

El concepto de HLPN surge como una respuesta a las limitaciones de las redes de Petri de lugar-transición, las cuales, si bien proporcionan un marco sólido para la representación de sistemas de eventos

discretos, resultan poco eficientes en la modelización de sistemas a gran escala debido al crecimiento exponencial de los estados y transiciones. Para abordar esta problemática, las HLPN incorporan *tokens* con atributos, estructuras de datos complejas, y expresiones lógicas en las transiciones, lo que permite compactar la representación de los sistemas sin perder expresividad [27].

Desde un punto de vista formal, una HLPN se define como una tupla $HLPN=(P,T,A,C,E,G,I,M)$, donde P representa los lugares, T las transiciones, A los arcos que conectan elementos, C una función de colores que asigna tipos de datos a los *tokens*, E las expresiones que determinan la transformación de los *tokens* en los arcos, G las condiciones lógicas para la habilitación de transiciones, I la función de inscripción inicial y M el marcado inicial del sistema. Este formalismo ha sido estandarizado bajo la norma ISO/IEC 15909, garantizando su interoperabilidad en diferentes entornos de modelado y ejecución [25].

Una de las aplicaciones más destacadas de las HLPN es su empleo en la simulación y validación de flujos de trabajo en entornos industriales. En particular, en el marco de la ingeniería de control y la automatización, han sido utilizadas para modelar sistemas de manufactura, donde la representación de productos en diferentes etapas de procesamiento requiere la incorporación de atributos específicos en los *tokens* [25].

Además, las HLPN han demostrado ser particularmente efectivas en la modelización de protocolos de comunicación y en la evaluación de prestaciones de redes informáticas. Al permitir la representación de mensajes con múltiples atributos y definir condiciones de enrutamiento mediante expresiones lógicas, este formalismo facilita la simulación de redes de telecomunicaciones, garantizando una validación rigurosa de los protocolos antes de su implementación. En este contexto, se han desarrollado herramientas de simulación basadas en el HLPN que permiten analizar el comportamiento de protocolos de acceso al medio, como ALOHA y CSMA/CD, proporcionando métricas clave sobre eficiencia y confiabilidad [27].

Otra contribución relevante de las HLPN es su capacidad para integrarse con otros modelos matemáticos y sistemas de información, facilitando la interoperabilidad. Esta flexibilidad ha permitido su aplicación en el desarrollo de sistemas de *software* críticos, donde la verificación formal es un requisito fundamental para garantizar la corrección del sistema antes de su despliegue en entornos reales [26].

Desde el punto de vista computacional, la implementación de HLPN ha sido respaldada por diversas herramientas de modelado. Estas herramientas proporcionan un entorno gráfico para la definición de las redes, permitiendo la simulación y análisis de propiedades como la, alcanzabilidad y ausencia de bloqueos en los modelos diseñados. Adicionalmente, en el ámbito académico, las HLPN han sido incorporadas en programas de enseñanza para la formación en modelado de sistemas, ofreciendo a los estudiantes un marco riguroso para el análisis de sistemas concurrentes y distribuidos [25].

En conclusión, las HLPN constituyen una herramienta fundamental para el modelado y análisis de sistemas dinámicos en múltiples dominios. Su capacidad para representar estructuras complejas mediante la incorporación de atributos en los *tokens*, junto con la posibilidad de definir condiciones lógicas avanzada, las convierte en una alternativa eficiente frente a las redes lugar-transición tradicionales. Su aplicación en áreas como la automatización industrial, la simulación de redes de comunicación y el desarrollo de *software* crítico demuestra su versatilidad y relevancia en la ingeniería moderna.

2.2.3 Lenguaje Sequential Function Chart

El *Sequential Function Chart* (SFC) es un lenguaje utilizado en la especificación de programas para controladores lógico-programables. Su principal propósito es modelar de manera estructurada y jerárquica las secuencias de control de procesos industriales, permitiendo la implementación de sistemas de automatización en entornos de producción complejos. Su empleo es particularmente relevante en la

industria de procesos *batch*, para la representación de la parte procedimental de una receta de fabricación, facilitando la estructuración de procedimientos mediante la identificación de estados y transiciones. Esta característica lo convierte en una herramienta fundamental en la planificación y control de la producción, garantizando la ejecución de procesos de manera eficiente y optimizada [28],[29].

El lenguaje SFC se estructura en secuencias de estados, las cuales están compuestas por pasos, transiciones y enlaces. Los pasos representan estados específicos dentro de un proceso y pueden estar activo o inactivos, mientras que las transiciones determinan el paso de un estado a otro en función de condiciones lógicas predefinidas. Estas estructuras permiten la configuración de secuencias simples, donde los estados se suceden de manera lineal, así como secuencias paralelas, en las que múltiples procesos pueden ejecutarse simultáneamente. Asimismo, el SFC permite modular secuencias divergentes, en las cuales se elige una transición entre varias opciones posibles, y convergencias de secuencias, que integran múltiples caminos en una única trayectoria. También es posible la implementación de *loops*, mediante los cuales la ejecución de un paso puede retroceder a un estado previo, repitiendo un ciclo de trabajo cuando sea necesario [28].

En el contexto de la Industria 4.0, la integración del lenguaje SFC con tecnologías de gemelo digital y *Asset Administration Shell* (AAS) ha cobrado relevancia, ya que permite la digitalización y modelización de procesos industriales, asegurando la interoperabilidad entre diferentes sistemas de producción. Un gemelo digital es una representación virtual de un activo industrial que interactúa con su contraparte física en tiempo real, permitiendo la optimización del rendimiento y la toma de decisiones basada en datos. En este sentido, un AAS es un marco de referencia propuesto que define la estructura y el intercambio de información entre diferentes activos digitales. La integración del SFC en estos entornos digitales facilita la automatización y optimización de la producción (PPC), permitiendo a las empresas mejorar la flexibilidad y eficiencia en la gestión de procesos productivos [29].

Uno de los principales desafíos en la implementación de SFC es la verificación de la validez de los modelos de control, especialmente en lo que respecta a la detección de errores estructurales. Se necesita evitar dos tipos de problemas en la construcción de redes SFC: las redes inseguras, caracterizadas por la activación descoordinada de múltiples pasos de manera simultánea, y las redes inalcanzables, que contienen pasos o transiciones que nunca serán ejecutadas. Para mitigar estos problemas, se han desarrollado mecanismos de verificación formal que emplean herramientas de modelado. Estas herramientas permiten definir reglas y restricciones que garantizan la correcta estructuración de modelos, asegurando que las secuencias de control sean ejecutables y funcionales en entornos industriales [28].

La formalización del lenguaje SFC ha permitido el desarrollo de técnicas avanzadas de validación, como el uso de atributos derivados que permiten evaluar la coherencia estructural de una red. Estos atributos han sido implementados, asegurando la detección temprana de configuraciones incorrectas y la generación de modelos robustos para su uso en sistemas industriales [29].

En el ámbito de la producción por lotes, el uso de SFC representa una ventaja significativa al permitir la estructuración y estandarización de recetas de fabricación, asegurando la coordinación efectiva de equipos y recursos. La incorporación del SFC dentro del modelo de AAS facilita la interoperabilidad con otros sistemas digitales, promoviendo una gestión inteligente de la producción basada en modelos digitales estandarizados. La representación formal de las secuencias de producción mediante este lenguaje no solo mejora la eficiencia operativa, sino que también proporciona un marco estructurado para la optimización continua de procesos industriales [29] [28].

En conclusión, el SFC se ha consolidado con una herramienta esencial en la automatización industrial, proporcionando una representación clara y estructurada de secuencias de control. Su aplicación en la industria de procesos *batch*, junto con su integración en tecnologías digitales como DT y AAS, permite la optimización de procesos interoperabilidad. La formalización del SFC garantiza la validez de los modelos y evita errores estructurales, asegurando que los sistemas de control sean eficientes y seguros. Su empleo en la planificación y control de la producción sigue evolucionando,

impulsando la transformación digital y mejorando la capacidad de respuesta de las industrias ante los desafíos de la manufactura moderna.

2.2.4 Comparación de los lenguajes

Para evaluar los distintos lenguajes utilizados en la implementación de redes de Petri, se ha considerado varios criterios fundamentales como la aplicabilidad, facilidad de programación, capacidad de modelado, flexibilidad, soporte de concurrencia y dificultad de implementación con entornos industriales. A continuación, en la Figura 2-3 se presenta una comparación detallada entre el LD, HLPN y el lenguaje SFC.

Características	Lenguaje <i>Ladder</i> (LD)	Redes de Petri de Alto Nivel (HLPN)	Lenguaje SFC (<i>Sequential Function Chart</i>)
Aplicabilidad	Automoción industrial, control PLCs	Modelado matemático, simulación de procesos	Control de procesos industriales, producción por lotes
Facilidad de programación	Alta, basado en diagramas eléctricos	Moderada, requiere conocimientos matemáticos avanzados	Moderada, requiere estructuración secuencial
Capacidad de modelado	Baja, limitado a secuencias lógicas simples	Alta, permite modelar sistemas complejos con condiciones lógicas avanzadas	Alta, permite modelar secuencias, paralelismos y ciclos
Flexibilidad	Baja, difícil adaptación a sistemas complejos	Alta, integración con otros modelos matemáticos	Alta, adaptable a diferentes entornos industriales
Soporte de concurrencia	Limitado, requiere programación adicional para sincronización	Alto, modela eventos concurrentes de forma nativa	Medio, maneja procesos paralelos, pero con restricciones
Dificultad implementación	Baja, intuitivo para ingenieros y técnicos	Alta, requiere <i>software</i> especializado	Media, requiere validación formal para evitar errores

Tabla 2-2 Comparativa entre lenguajes para implementación de redes de Petri.

2.3 Proyectos de procesos automatizados

La automatización de procesos ha cobrado una relevancia creciente en el ámbito industrial y naval, impulsada por la necesidad de mejorar la eficiencia operativa, reducir costos y optimizar la producción. En el contexto de la denominada Industria 4.0, la integración de tecnologías digitales y la informatización de los sistemas productivos han permitido transformar los métodos tradicionales de fabricación y gestión, dando paso a procesos más eficientes y adaptables a las condiciones cambiantes del mercado. Dentro de esta transformación, la automatización de procesos se posiciona como en eje central para la competitividad y la sostenibilidad, tanto en la industria manufacturera como en otros

sectores, donde la complejidad de los sistemas y la necesidad de precisión requieren soluciones avanzadas [30].

La transición a entornos de producción más automatizados responde a la creciente demanda de soluciones inteligentes que optimicen la gestión de los recursos y permitan una mayor integración de las TIC en los procesos industriales. Este enfoque no solo implica la utilización de maquinaria automatizada, sino también la digitalización de los requisitos funcionales, la monitorización en tiempo real de los procesos y la utilización de inteligencia artificial (IA) para la toma de decisiones estratégicas [30].

La automatización en este contexto no solo tiene como objetivo aumentar la productividad, sino que también permite la personalización de productos según las necesidades específicas del cliente, optimizando los tiempos de producción y reduciendo el margen de error. En este sentido, el uso de técnicas avanzadas, como la integración de PLCs, la robótica industrial, el *Robot Process Automation* (RPA) y la IA, ha demostrado ser fundamental para la evolución de los sistemas productivos. La capacidad de estas tecnologías para gestionar de manera autónoma procesos repetitivos y para analizar grandes volúmenes de datos en tiempo real contribuye a una mejor toma de decisiones y una mayor eficiencia operativa [30].

En los siguientes apartados se analizarán proyectos enfocados en el mismo ámbito, la automatización de procesos, pero abordando diferentes enfoques aplicados tanto en la manufactura como en la gestión y análisis de datos. A través del estudio de casos concretos, se evidenciará cómo la implementación de sistemas automatizados ha permitido mejorar la eficiencia en diversas áreas industriales, desde la optimización y reducción de costos hasta la mejor en la precisión y la calidad de los productos finales. Estos ejemplos servirán para ilustrar el impacto de la automatización en la evolución de los procesos industriales y su papel en la transformación digital de los sectores productivos.

A través del análisis de estos dos ejemplos de procesos automatizados, se podrá evidenciar la gran diversidad existente en la automatización, no solo en el ámbito de los procesos físicos, como el embotellado industrial, sino también en la gestión de información, como en el caso del procesamiento automatizado de datos en la venta de entradas. Ambos proyectos muestran cómo la automatización se adapta a diferentes necesidades y sectores, optimizando tareas repetitivas y reduciendo la intervención humana en actividades que pueden ser gestionadas de manera más eficiente por sistemas programados.

En el caso del embotellado, la integración de sensores, motores y controladores programables ha permitido optimizar un proceso mecánico de producción, asegurando precisión y eficiencia en cada etapa. Por otro lado, en el caso de análisis de entradas deportivas, la aplicación de RPA ha demostrado como la automatización también puede aplicarse en la recopilación, transformación y visualización de datos para mejorar la toma de decisiones. Esto permite comprender que la automatización no se limita únicamente al a ejecución de tareas en entornos industriales, sino que también juega un papel fundamental en la clasificación y estructuración de información en entornos digitales.

Estos dos ejemplos muestran cómo la automatización ha trascendido su aplicación tradicional en fábricas y líneas de ensamblaje, abarcando sectores como la inteligencia de negocio, la minería de datos y la optimización de procesos administrativos. En este sentido, la diversidad de aplicaciones de la automatización evidencia su capacidad para mejorar la productividad, reducir errores y optimizar recursos, tanto en el mundo físico como en el digital. Con el avance de la tecnología, es previsible que esta tendencia continúe expandiéndose, ofreciendo nuevas oportunidades para mejorar la eficiencia en múltiples sectores y consolidando la automatización como una herramienta clave en la transformación digital.

A continuación, se expondrán a grandes rasgos un proyecto de embotellado y otro de extracción y análisis, ambos automatizados.

2.3.1 Proyecto de embotellado automatizado

El proyecto consiste en el desarrollo e implementación de un prototipo automatizado para el llenado y tapado de botellas de plástico, con el objetivo de mejorar la calidad del producto y aumentar la eficiencia del proceso envasado. La necesidad de optimizar la producción de agua embotellado y otros líquidos ha llevado a la adopción de tecnologías avanzadas de automatización que permiten reducir costos operativos y aumentar la productividad. Para ello, se diseñó un sistema compuesto por actuadores neumáticos, sensores, electroválvulas, solenoides, motores y un controlador lógico programable. Este sistema automatizado permite gestionar de manera eficiente el flujo de trabajo, eliminando los cuellos de botella y asegurando un llenado uniforme y preciso [31].

El funcionamiento del sistema se estructura en varias etapas. En primer lugar, las botellas vacías se colocan en una faja transportadora controlada por un motor. La velocidad de esta faja ha sido calibrada para garantizar un flujo constante y estable de botellas a lo largo del proceso. Posteriormente, las botellas pasan a la estación de llenado, donde una matriz con diez cavidades y un mecanismo de movimiento rotativo paso a paso, basado en una rueda de ginebra, permite posicionarlas adecuadamente bajo las válvulas de llenado. Un sensor óptico detecta la presencia de cada botella, deteniendo momentáneamente el sistema para permitir la apertura de la válvula solenoide, a través de la cual el líquido fluye desde un depósito de almacenamiento. Esta fase del proceso garantiza la dosificación exacta del contenido de cada botella, asegurando uniformidad en el llenado [31].

Una vez completada la fase de llenado, las botellas avanzan a la siguiente etapa donde se colocan las tapas. Para ello, se diseñó un dispensador de tapas que posiciona las mismas sobre las botellas a medida que estas se desplazan en la matriz giratoria. Dado que este paso requiere precisión, se utilizaron sensores ópticos para garantizar que las tapas sean correctamente colocadas en la boca de las botellas antes de proceder a la fase de roscado. En esta etapa, las botellas pasan bajo un motor equipado con un molde roscador que se ajusta a la tapa. A través de un cilindro neumático de doble efecto, el motor desciende y ejecuta el roscado de manera automática, asegurando que las tapas queden firmemente selladas sin generar daños en la botella. La detección y alineación del sistema es clave para evitar errores en esta fase, motivo por el cual se emplearon sensores magnéticos que sincronizan la operación con la posición exacta de la botella [31].

Finalmente, las botellas llenas y tapadas se trasladan a la última fase del proceso, donde son distribuidas para posterior etiquetado y empaquetado. La automatización de este sistema ha permitido reducir en un 36 % el tiempo de embotellado en comparación con los métodos manuales tradicionales. Además, la implementación de este sistema ha demostrado un incremento significativo en la calidad del producto final, cierre de las botellas. Gracias a esta mejora en la precisión, se ha logrado reducir en un 90% las devoluciones de botellas defectuosas por parte de los consumidores, lo que ha facilitado la posibilidad de expansión a nuevos mercados [31].

El desarrollo de este proyecto implicó varias etapas de diseño, ensamblaje y pruebas. Inicialmente se llevó a cabo el diseño preliminar del sistema utilizando *software* CAD (*Computer Aided Design*), asegurando que todos los componentes estuvieran correctamente dimensionados y alineados con los requerimientos del proceso. Posteriormente, se adquirieron los materiales necesarios y se procedió con el ensamblaje de la estructura del prototipo, seguido de la instalación de sensores, cilindros neumáticos y solenoides. Una vez finalizado el cableado del PLC, se desarrolló la programación necesaria para coordinar todas las fases del proceso de manera automática. Las pruebas iniciales permitieron verificar el correcto funcionamiento del sistema y realizar ajustes para optimizar su rendimiento. Finalmente, se realizó una prueba final en la que se evaluó la eficiencia del sistema en condiciones de operación real [31].

En este primer proyecto se realiza la automatización mediante un PLC, y la programación se lleva a cabo utilizando el LD, una de las metodologías más utilizadas en entornos industriales para automatización de maquinaria. En el inicio del proceso se configura una entrada digital que activa la

cinta transportadora cuando las botellas vacías son detectadas por un sensor óptico, además también se establecen temporizadores para garantizar que el sistema espere el tiempo necesario para posicionar las botellas en la zona de llenado. Ya en la fase de llenado, una vez la botella está posicionada, el PLC acciona una válvula solenoide, la cual se abre para permitir el flujo de líquido hasta que un sensor de nivel indique que la botella está llena, se implementan interruptores de presión para evitar derrames y controlar la cantidad exacta de líquido en cada botella. En la colocación de la tapa y roscado, se activa un mecanismo neumático que posiciona la tapa sobre la botella utilizando sensores de proximidad, se emplea un motor que desciende mediante un cilindro neumático y ajusta la tapa a través de un molde roscador. La programación LD controla la secuencia de operación, asegurando que el motor no actúe hasta que la tapa esté correctamente alineada. Por último, para su distribución, una vez finalizado el roscado, el PLC se activa nuevamente la cinta transportadora para trasladar las botellas al área de almacenamiento, se han programado alarmas y fallos en caso de que un sensor no detecte la tapa o si hay un error en la fase de llenado [31].

2.3.2 Proyecto de extracción y análisis automatizado

El proyecto se centra en la implementación de un sistema automatizado para la toma de decisiones en un entorno digital, utilizando tecnologías avanzadas como RPA y herramientas de análisis de datos como *Tableau*. El objetivo principal es optimizar el proceso de extracción y análisis de información, permitiendo la automatización de tareas repetitivas que antes requerían intervención manual [32].

El estudio se basa en la automatización del proceso de adquisición y análisis de datos provenientes de una web de venta de entradas para eventos deportivos. A través de *UiPath*, se han desarrollado robots desatendidos que diariamente acceden al portal de venta de entradas, extraen información sobre disponibilidad, precios y tendencias de compra, y la almacenan en bases de datos estructuradas. Estos datos son posteriormente analizados y visualizados en *Tableau*, donde se generan *dashboards* interactivos que facilitan la interpretación de la información en tiempo real [32].

El flujo de trabajo del sistema automatizado consta de varias fases. En primer lugar, los robots de *UiPath* ejecutan tareas de *web scraping*, que consiste en una técnica informática que extrae información automáticamente desde páginas web, recopilando información sobre eventos disponibles, número de entradas vendidas y precios por zona dentro del recinto deportivo. Este proceso es dinámico, ya que las plataformas de venta de entradas pueden modificar su estructura y presentación de datos, lo que exige una adaptación constante de algoritmos de extracción [32].

Una vez extraídos los datos, estos son transformados y organizados en estructuras de almacenamiento adecuadas para su análisis. Se aplican procesos de limpieza y consolidación para garantizar que la información sea precisa y relevante. Posteriormente generan indicadores clave de rendimiento que permiten visualizar tendencias de compra, ocupación de asientos y fluctuaciones en los precios de las entradas a lo largo del tiempo [32].

El análisis de los resultados obtenidos permite comprender el comportamiento de los usuarios en la compra de entradas, identificar patrones de demanda y prever posibles estrategias de optimización de precios y gestión de aforo. Además, el modelo desarrollado es escalable y adaptable a otros entornos digitales, lo que abre la posibilidad de aplicarlo a distintos sectores donde la toma de decisiones basada en datos sea un factor determinante [32].

En términos de impacto, este proyecto demuestra como la automatización y la inteligencia de negocio pueden transformar procesos convencionales en sistemas altamente eficientes y adaptables. La implementación de RPA no solo reduce la carga de trabajo manual, sino que también mejora la precisión y rapidez en la toma de decisiones, ofreciendo una ventaja competitiva en entornos donde el acceso inmediato a información estructurada es clave [32].

Este enfoque, además de optimizar la gestión operativa, proporciona una base sólida para futuras investigaciones y desarrollos en el ámbito de la automatización y analítica de datos. Con la evolución de

la tecnología y el crecimiento del *Big Data*, este tipo de soluciones continuará siendo fundamental para mejorar la eficiencia y la competitividad en múltiples industrias [32].

En este proyecto se ha utilizado *UiPath* para programar robots que automatizan la extracción, transformación y análisis de datos de una web de venta de entradas. La programación del proceso automatizado se basa en la metodología RPA y sigue un flujo de ejecución bien estructurado. Primero, se realiza la extracción de datos (*Web Scraping*), se programa un robot desatendido que se ejecuta diariamente y accede a la web de venta de entradas, se usa la actividad “*Extract Structured Data*” un *UiPath* para obtener información sobre disponibilidad de entradas, precios y ubicaciones, y se han programado condicionales para manejar cambios en la estructura de la web y evitar fallos en la extracción de datos. Seguidamente, se realiza la transformación de los datos, una vez extraídos, los datos se procesan utilizando scripts en *UiPath Studio* para limpiar, estructurar y formatear la información, se aplican reglas de filtrado y clasificación para determinar qué sectores tienen más demanda y cuales están quedando vacíos, se han integrado expresiones regulares para la normalización de los datos extraídos. Después se realiza el almacenamiento y procesamiento, los datos se guardan en archivos Excel y bases de datos SQL (*Structured Query Language*), permitiendo su consulta posterior, se han desarrollado los scripts de automatización que verifican diariamente si hay cambios en los datos históricos. A continuación, se realiza la visualización y análisis en *Tableau*, se ha configurado una interfaz de programación en aplicaciones (API) en *UiPath* que envía automáticamente los datos procesados a *Tableau* para la generación de *dashboards* interactivos, se han programado conectores en *UiPath* que permiten la actualización en tiempo real de las visualizaciones en función de los nuevos datos recopilados. Y, por último, la ejecución y monitorización del robot, se ha programado una ejecución automática y programada desde *UiPath Orchestrator*, donde los robots se ejecutan sin supervisión, además se han integrado *logs* de ejecución que almacenan información sobre posibles fallos y permiten su corrección en tiempo real [32].

3 DESARROLLO DEL TRABAJO

El Trabajo de Fin de Grado se estructura en proyectos, cada uno compuesto por tres módulos que corresponden a las prácticas de laboratorio desarrolladas durante la asignatura Fundamentos de Automática en el cuarto curso. Esta organización tiene como objetivo proporcionar un aprendizaje progresivo, donde los módulos dentro de cada proyecto aumentan en complejidad, integrando nuevos conceptos y herramientas estudiadas en las clases teóricas y seminarios. De este modo, los estudiantes pueden asimilar de manera gradual los fundamentos de la automatización y el control de sistemas. Además, se busca dotar el aprendizaje de un enfoque aplicado a su futura profesión, asegurando que los proyectos estén alineados con las actividades realizadas en la Armada. Así, los estudiantes no solo adquieren conocimientos teóricos y prácticos, sino que también desarrollan habilidades directamente relacionadas con el ámbito profesional en el que desempeñarán su labor.

Cada módulo está diseñado para guiar al estudiante a través de un proceso de aprendizaje progresivo, en lo que respecta a la programación de redes de Petri en lenguaje LD. En una primera etapa, se introduce un nivel básico de comprensión del sistema; posteriormente, se incorporan funciones adicionales que amplían su alcance; y, finalmente, se integran elementos avanzados que permiten una gestión completa y eficiente del sistema en estudio. Esta metodología escalonada favorece un aprendizaje estructurado, permitiendo a los alumnos desarrollar y consolidar sus habilidades técnicas de manera progresiva.

Es importante señalar que los sistemas modelados en este trabajo no representan fielmente los sistemas reales empleados en la industria naval. Una simulación completamente precisa aumentaría significativamente la complejidad del estudio, lo que dificultaría su aplicación en el entorno académico. No obstante, se ha procurado preservar la esencia de los procesos reales, de modo que los estudiantes puedan familiarizarse con la terminología, la metodología y los principios fundamentales del sector en el que desarrollarán su carrera profesional. De esta forma, los contadores y temporizadores utilizados en los proyectos no reflejan valores exactos de los sistemas reales, sino que han sido ajustados para facilitar su visualización y ejecución en las prácticas de laboratorio. Esta adaptación permite a los estudiantes centrarse en la lógica de funcionamiento sin las restricciones de tiempos y parámetros industriales, optimizando así su proceso de aprendizaje.

Cabe destacar que el alcance de este trabajo se ha enfocado exclusivamente en la programación del sistema, dado que la parte visual ya estaba previamente diseñada. En este sentido, el desarrollo se ha orientado a la implementación de la lógica de operación mediante el uso de herramientas de automatización.

El objetivo final de estos proyectos es que los alumnos no solo comprendan el funcionamiento teórico y práctico de los proyectos desarrollados, sino que también sean capaces de aplicar estos conocimientos en un entorno estructurado y controlado. Esto les proporcionará una base sólida para

abordar sistemas más complejos en su trayectoria profesional, asegurando un aprendizaje significativo y alineado con las exigencias y necesidades de la industria naval.

A continuación, se exponen los proyectos que han sido desarrollados durante este trabajo.

3.1 Proyecto 1: Automatización operativa y preventiva de un sistema de propulsión híbrido naval.

El sistema de control de propulsión de un barco equipado con tecnología de propulsión combinada, que integra fuentes de energía eléctrica y a gas, debe ser capaz de gestionar de manera autónoma, eficiente y segura la transición entre los diferentes modos de operación: eléctrico, gas o híbrido. Este proceso de gestión dinámica depende de las condiciones detectadas en tiempo real por un conjunto de sensores distribuidos, así como de la demanda de potencia requerida por el sistema en cada momento.

Durante estas transiciones, es esencial garantizar la continuidad operativa, evitando interrupciones que puedan afectar la funcionalidad del barco. Al mismo tiempo, se busca optimizar el consumo energético, reduciendo el uso innecesario de recursos y minimizando cualquier impacto negativo en la eficiencia, la confiabilidad y la vida útil de los componentes del sistema. Este equilibrio entre rendimiento y sostenibilidad resulta crucial para cumplir con los exigentes estándares de la industria marítima moderna, que cada vez otorga mayor prioridad a la eficiencia energética y a la reducción de emisiones.

El sistema también debe ser capaz de supervisar y gestionar un conjunto de subsistemas auxiliares vitales, como los sistemas de enfriamiento, limpieza de turbinas y lubricación. Estos subsistemas desempeñan un papel crítico en el mantenimiento de las condiciones óptimas de operación y en la protección de los componentes clave del sistema de propulsión. La activación o desactivación de estos subsistemas debe realizarse de forma automática y precisa, en respuesta a condiciones críticas detectadas en tiempo real, como variaciones en la temperatura, presión, niveles de vibración u otros parámetros que puedan comprometer la operación segura.

Además de su capacidad para reaccionar ante condiciones críticas, el sistema debe anticiparse a escenarios potenciales de riesgo, ajustando de manera proactiva las configuraciones de operación para prevenir fallos o deterioros prematuros de los componentes. Esto requiere no solo un monitoreo constante, sino también algoritmos avanzados de toma de decisiones que integren análisis predictivo y control adaptativo.

En conjunto, este enfoque permite no solo una operación más eficiente y segura, sino también una mejora en la sostenibilidad ambiental y económica del barco, reduciendo tanto los costos de operación como el impacto ambiental asociado al consumo de combustible y las emisiones.

Para abordar este desafío, el modelado del sistema utilizando redes de Petri ofrece una herramienta poderosa para diseñar y analizar la lógica de operación requerida. Este enfoque posibilita la implementación de una gestión optimizada de recursos, asegurando que las decisiones del sistema sean automáticas, coordinadas y adaptativas a los cambios dinámicos del entorno operativo.

El objetivo es modelar una simplificación del sistema de control de propulsión de un barco con redes de Petri, integrando entradas y salidas, temporizadores, contadores de ciclos, y recursos compartidos. Esto permitirá simular secuencias lógicas de funcionamiento y gestionar las operaciones de manera eficiente y segura.

El desarrollo del proyecto se lleva a cabo mediante tres etapas, las cuales se detallan a continuación:

1. Gestión de la propulsión:

La gestión de los modos de propulsión híbrida en un barco que combina dos motores eléctricos y una turbina de gas requiere una integración sofisticada de varios sistemas. Estos sistemas

permiten transiciones fluidas entre los diferentes modos de operación (eléctrico, gas o híbrido), optimizando la eficiencia energética y garantizando la seguridad y confiabilidad en tiempo real.

En este módulo se desarrolla el funcionamiento general del sistema de propulsión híbrida en base al modo de operación seleccionado y la necesidad de potencia requerida, teniendo en cuenta que el sistema se compone de dos baterías, dos motores eléctricos y una turbina de gas que se encargan de rotar el eje de transmisión que transmite el movimiento a las hélices del barco.

2. Gestión del mantenimiento preventivo:

La gestión del mantenimiento preventivo en un sistema de propulsión híbrida que combina tecnología de gas y eléctrica es una tarea fundamental para garantizar la operatividad, seguridad y eficiencia del sistema. Este tipo de propulsión, al integrar dos fuentes de energía complementarias, presenta una mayor complejidad técnica, lo que requiere un enfoque estructurado y proactivo en la planificación y ejecución del mantenimiento.

De esta forma, en este módulo se añade la gestión de tareas relativas al mantenimiento preventivo de la instalación, tales como la limpieza de la turbina de gas cada cierto número de ciclos de uso, la recarga de las baterías cuando su nivel sea bajo, la recarga del depósito de gas que alimenta la turbina o la necesidad de controlar y recargar el lubricante empleado para minimizar el desgaste de los componentes móviles.

3. Gestión de alarmas y fallos:

La gestión de alarmas y fallos en un sistema de propulsión híbrida (gas-eléctrica) es una tarea crucial para garantizar la operación segura, eficiente y confiable del barco. Esta gestión se compone de varios sistemas interconectados que monitorean, detectan, analizan y responden ante condiciones anómalas o fallos potenciales.

De esta forma, en este módulo se añade la necesidad de detectar y actuar frente a sobrecargas eléctricas, sobrecalentamientos y fallos de aislamiento en el sistema eléctrico, pérdida de eficiencia por obstrucción o residuos en la caldera de gas, aparición de temperaturas extremas en la caldera de la turbina de gas o al aumento de vibraciones en el eje de transmisión.

3.1.1 Descripción del sistema de propulsión híbrida

El sistema de propulsión híbrida que se busca automatizar está compuesto por una combinación de elementos mecánicos y tecnológicos avanzados que trabajan en conjunto para garantizar un funcionamiento eficiente, flexible y seguro. Este sistema integra los siguientes componentes principales:

1. Componentes mecánicos:

- a. **Hélice:** responsable de generar el empuje necesario para mover el barco. Su diseño y eficiencia son cruciales para maximizar el rendimiento del sistema de propulsión.
- b. **Eje de transmisión:** conecta los motores eléctricos y la turbina de gas con la hélice, transmitiendo la potencia generada de manera eficiente. Este eje puede estar equipado con acoplamientos flexibles o embragues para permitir transiciones suaves entre los diferentes modos de operación.
- c. **Otros elementos mecánicos:** incluyen cojinetes, soportes y sistemas de sellado diseñados para soportar las cargas dinámicas y evitar pérdidas de eficiencia durante la operación.

2. Componentes eléctricos y de almacenamiento de energía:

- a. **Dos baterías de alta capacidad (B1 y B2):** actúan como fuente de energía principal para los motores eléctricos en el modo eléctrico y como respaldo en el modo híbrido.

Diseñadas para ofrecer alta densidad energética y ciclos de carga/descarga eficiente, minimizando el deterioro a largo plazo. Equipadas con sistemas de monitoreo que controlan la temperatura, estado de carga y salud de las celdas, asegurando un rendimiento óptimo.

- b. **Dos motores eléctricos (E1 y E2):** responsables de proporcionar la potencia necesaria para impulsar el barco en modo eléctrico. Pueden operar de manera independiente o conjunta, dependiendo de la demanda de potencia y las condiciones operativas. Controlados por sistemas electrónicos que regulan la velocidad y el par motor para optimizar la eficiencia energética.

3. Componentes de Combustible y Generación de Energía:

- a. **Depósito de gas:** almacena el combustible necesario para la operación de la turbina de gas en modo gas o híbrido. Diseñado para cumplir con estrictas normativas de seguridad, incluyendo sistemas de válvulas de alivio, monitoreo de presión y mecanismos de detección de fugas.
- b. **Turbina de gas:** actúa como la fuente principal de energía en el modo gas y como apoyo en el modo híbrido, generando energía mecánica que puede ser utilizada directamente para la propulsión o convertida en electricidad mediante generadores. Optimizada para operar en un amplio rango de velocidades y condiciones, con sistemas avanzados de control de combustión para maximizar la eficiencia y reducir las emisiones.

A continuación, se exponen los 3 módulos que componen el proyecto, de forma que para cada módulo se expone su enunciado, la tabla de variables (entradas, salidas, temporizadores y contadores) empleada en la programación, las redes de Petri que modelan el comportamiento del sistema automático definido, y una descripción de la programación del sistema, así como de la visualización disponible para cada módulo.

3.1.2 Módulo 1: Gestión y control de un sistema de propulsión híbrida

3.1.2.1 Enunciado

Para realizar la automatización del sistema de propulsión descrito se deben considerar los siguientes aspectos:

Los motores eléctricos se encienden ejecutando las instrucciones **ELECTRIC1** y **ELECTRIC2**.

- La turbina de gas se enciende ejecutando la instrucción **GAS**.
- Inicialmente se asume que el sistema está apagado, con las baterías totalmente cargadas, el depósito de gas lleno y en modo de operación manual.
- Si se pulsa el botón de parada el sistema se apaga.
- El sistema debe emplear los siguientes contadores generales:
 - **CP:** Número de veces que se produce un cambio en las condiciones de potencia requerida.
 - **CE1:** Número de veces que se activa el motor eléctrico 1.
 - **CE2:** Número de veces que se activa el motor eléctrico 2.
 - **CG:** Número de veces que se activa la turbina de gas.
- El panel de control y seguimiento se compone de los siguientes elementos:
 - Un botón de arranque que genera la señal de encendido del sistema **START**.

- Un botón de parada que genera la señal de apagado del sistema **STOP**. El sistema solo puede apagarse si los dos motores y la turbina no están en uso, en cuyo caso se desestima la señal de parada.
- Un selector del porcentaje de potencia requerida que genera las siguientes señales:
 - **POT25**: Se activa cuando el selector de potencia está por encima del 25%.
 - **POT50**: Se activa cuando el selector de potencia está por encima del 50%.
 - **POT75**: Se activa cuando el selector de potencia está por encima del 75%.
- Un selector del modo de operación que genera la señal **MODE**, cuyo valor indica lo siguiente:
 - **MODE==0**: Modo de operación manual.
 - **MODE==1**: Modo de operación automático.
- El funcionamiento de la propulsión, en función del modo y la potencia requerida, es el siguiente:
 - Modo Manual: (**MODE=0**): En el modo manual se asume que el operador controla cada sistema de propulsión de forma manual, para lo cual dispone de tres mandos de control (**ON_ELE1**, **ON_ELE2**, **ON_GAS**) que directamente activan directamente las instrucciones de salida relacionadas con el encendido de estos sistemas de propulsión.
 - Modo Automático: (**MODE=1**):
 - Si la potencia es inferior al 25% se activa el motor eléctrico 1 solo si está activado el mando de control **ON_ELE1**.
 - Si la potencia está entre el 25% y el 50% se debe activar los dos motores eléctricos.
 - Si la potencia está entre el 50% y el 75%:
 - Se activan los dos motores eléctricos.
 - Se activa la turbina de gas en ciclos de 2 segundos encendida y 5 segundos apagada.
 - Si la potencia es superior al 75% se activan los dos motores eléctricos y la turbina de gas simultáneamente.

3.1.2.2 Tabla de variables

En el presente módulo del proyecto de automatización de sistemas de propulsión híbrida en barcos, se establecen distintas variables que intervienen en la operatividad del sistema. Estas variables se agrupan en cuatro categorías principales: entradas, salidas, temporizadores, y contadores, cada una desempeñando un rol fundamental en el control y supervisión del sistema de propulsión.

En la Figura 3-1 las entradas representan todos los dispositivos de detección y activación que permiten la interacción con el sistema. En este caso, se dispone de elementos como botones de control, con las entradas **START** y **STOP**, que permite la activación o puesta en marcha del sistema y ordena la detención del sistema, siempre y cuando se cumplan las condiciones para un apagado seguro respectivamente. También existe el selector de modo de operación que define el régimen de funcionamiento del sistema, que puede ser manual o automático. A su vez, encontramos los mandos de activación individual de los motores **ON_ELE1**, **ON_ELE2**, **ON_GAS**, que controlan la activación del motor eléctrico uno, dos y habilita la puesta en funcionamiento de la turbina de gas. Y por último el selector de potencia, este componente, representado por una barra deslizante, permite definir la potencia requerida por el usuario, su configuración se traduce en los estados de entrada **POT25**, **POT50**, **POT75**, que se activan cuando la potencia requerida es superior al veinticinco, cincuenta y setenta y cinco por ciento.

Las salidas del sistema a diferencia de las entradas representan el estado operativo de los distintos componentes de propulsión en función de la configuración establecida y las condiciones de operación.

En este módulo solo existen definidas tres salidas, *ELECTRIC1*, *ELECTRIC2* y *GAS*, que indica que el motor uno o dos está en funcionamiento, o si la turbina de gas está encendida y operativa.

Para el monitoreo y control de los procesos operativos del sistema de propulsión tenemos los contadores, que registran eventos específicos dentro del funcionamiento del sistema. Entre ellos podemos encontrar *CP*, que registra la cantidad de cambios realizados en la potencia requerida, *CE1* que lleva un control del número de veces que se ha activado el motor número 1, *CE2* realiza el mismo seguimiento, pero aplicado al motor dos, y por último *CG*, que registra la cantidad de veces que se ha activado la turbina de gas.

Para finalizar, en la cuarta columna de la Tabla 3-1 se encuentran los temporizadores *TA1* y *TA2*, que regulan el tiempo de encendido y apagado de la turbina.

Entradas	Salidas	Contadores	Temporizadores
START	ELECTRIC1	CP	TA1
STOP	ELECTRIC2	CE1	TA2
POT25	GAS	CE2	
POT50		CG	
POT75			
MODE			
ON_ELE1			
ON_ELE2			
ON_GAS			

Tabla 3-1 Variables del análisis de gestión y control de un sistema de propulsión híbrido.

3.1.2.3 Red de Petri del análisis de gestión y control

En esta sección se presenta la modelización del sistema de propulsión híbrida de una embarcación mediante la red de Petri de la **¡Error! No se encuentra el origen de la referencia.**, la cual permite representar de manera gráfica y estructurada su funcionamiento. Este enfoque facilita el análisis del comportamiento del sistema en función de las condiciones establecidas y las transiciones posibles entre sus diferentes estados.

Inicialmente, el sistema se encuentra en estado apagado, lo cual se refleja con la marca en el lugar *PA1*. A partir de este estado, existe un único camino posible para que el sistema cambie de estado, representado mediante una transición denominada *TA1*. Las condiciones necesarias para que esta transición se lleve a cabo se encuentran indicadas junto a *TA1*, incluyendo las condiciones lógicas y temporales que deben cumplirse. Además, se especifican los contadores o temporizadores que deben ser inicializados al producirse la transición, los cuales se anotan después de una barra y un asterisco.

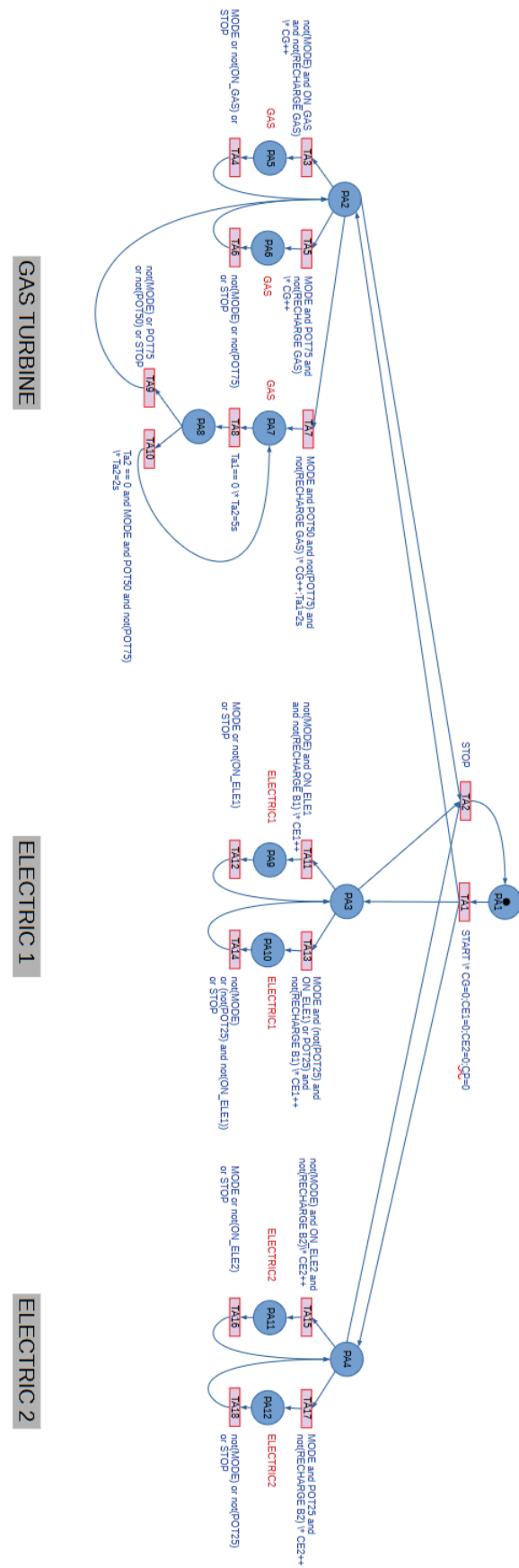


Figura 3-1 Red de Petri del análisis de gestión y control de un sistema de propulsión híbrido. Fuente propia.

Para que la transición *TA1* ocurra, la condición necesaria es la activación de la entrada *START*. Una vez cumplida esta condición, todos los contadores del sistema se inicializan en cero. Como resultado de esta transición, la marca en *PA1* desaparece y los nuevos lugares marcados son *PA2*, *PA3* y *PA4*. El comportamiento del sistema sigue el mismo principio descrito anteriormente, pero con condiciones específicas para cada caso particular.

Es relevante señalar que algunas transiciones incluyen la operación de incremento de contadores. Por ejemplo, en la transición *TA5*, si se cumplen las condiciones establecidas, se incrementa en una unidad el contador *CG*, lo cual se presenta como *CG++* en el modelo.

Por otra parte, las condiciones que deben cumplirse para que se produzca una transición no se limitan exclusivamente a la activación de entradas, sino que también pueden depender de temporizadores. Un ejemplo de esta situación se observa en la transición *TA8*, donde la única condición para que el sistema pase al estado *PA8* es que el temporizador *TA1* haya alcanzado su fin. Esto se representa en la red de Petri como *TA1=0*, dicho temporizador habiendo sido inicializado previamente en el lugar *PA7*.

Finalmente, es importante destacar que algunas condiciones pueden requerir la negación de una entrada para permitir la transición de estado. Un caso ilustrativo de este mecanismo se encuentra en la transición *TA12*, en la que se exige una entrada *ON ELE1* no esté activa para que la transición pueda ejecutarse.

En síntesis, la red de Petri utilizada en este estudio proporciona una representación formal y estructurada del funcionamiento del sistema de propulsión híbrida, permitiendo una mejor comprensión de sus estados, transiciones y condiciones necesarias para el cambio de estado. Este modelo facilita el análisis y optimización del sistema, asegurando un diseño eficiente y adecuado a los requerimientos operativos de la embarcación.

3.1.2.4 Programación y visualización

En este apartado se busca demostrar de manera gráfica la implementación práctica de la lógica de control desarrollada para el sistema de propulsión híbrida. La intención es evidenciar, mediante las figuras que van desde la Figura 3-2 hasta la Figura 3-8, como se traduce el modelo teórico, basado en redes de Petri, a una programación en lenguaje LD y cómo esta programación se refleja en una interfaz de usuario que permite la monitorización en tiempo real del sistema. Además, para poder observar cómo funciona de manera viva, se ha realizado un video donde se muestra el funcionamiento del sistema, este enlace se encuentra al final de este apartado y de la misma manera se encontrará en el resto de los módulos y proyectos.

En primer lugar, las figuras de la programación en lenguaje LD muestran la configuración de las variables e instrucciones básicas. Se puede observar cómo se asignan y utilizan comandos fundamentales tales como *ELECTRIC1*, *ELECTRIC2* y *GAS* que activan motores y la turbina de gas en función de las señales generadas por el panel de control. Dichas señales incluyen los botones de arranque (*START*) y de parada (*STOP*), los mandos de activación individual (*ON_ELE1*, *ON_ELE2*, *ON_GAS*) y los selectores de potencia (*POT25*, *POT50*, *POT75*). Esta estructura de programación se ha diseñado para reflejar fielmente la lógica definida en el modelo de red de Petri, integrando contadores y temporizadores que permiten registrar y sincronizar los eventos y transiciones del sistema. De esta manera, se garantiza que cada acción programada se ejecute conforme a las condiciones estipuladas, asegurando una operación coherente y predecible.

Por otro lado, las figuras de la interfaz de usuario evidencian la capacidad del sistema para mostrar de manera intuitiva el estado de cada uno de sus elementos. La visualización en tiempo real permite al operador observar cómo se activan los motores y la turbina, y cómo evolucionan los temporizadores y contadores a medida que el sistema responde a las variaciones en la potencia requerida y al cambio de modo de operación. Esta herramienta gráfica es fundamental para la validación y el diagnóstico del

sistema, ya que facilita la detección de posibles anomalías y permite al usuario interactuar de forma directa con el proceso de control.

En conjunto, este apartado no solo ilustra la implementación técnica de la lógica de control a través del lenguaje LD, sino que también demuestra la importancia de contar con una visualización efectiva para supervisar y validar el comportamiento del sistema en entornos simulados. La integración de ambos aspectos -la programación y la visualización- resulta crucial para garantizar que el sistema de propulsión híbrida opere de manera segura, eficiente y acorde a los requerimientos establecidos, consolidando así la metodología aplicada en este trabajo y proporcionando una base sólida para el aprendizaje de técnicas avanzadas de automatización en un contexto naval.

En la Figura 3-2, se presenta un fragmento de código dentro del bloque *HMI (Human-Machine Interfaz)*, que forma parte de un sistema de control diseñado para la gestión de un sistema de propulsión híbrida en barcos. Este sistema integra tanto motores eléctricos como una turbina de gas, lo que permite la optimización del consumo energético y una transición eficiente entre los distintos modos de operación.

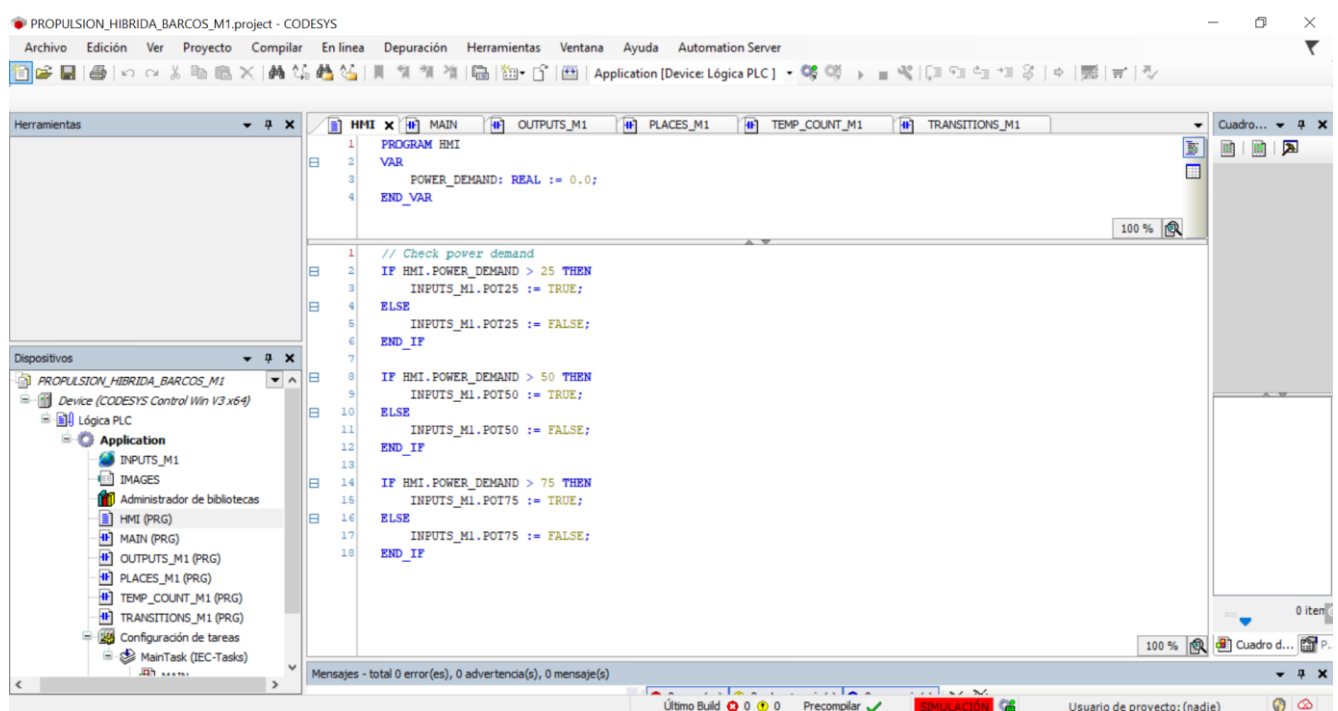


Figura 3-2 Programación Codesys del sistema de gestión de propulsión híbrida, bloque HMI.

El código que se visualiza tiene la función de gestionar la demanda de potencia en función de ciertos umbrales establecidos. Para ello, se define una variable denominada *POWER_DEMAND*, la cual representa la demanda de potencia del sistema y se inicializa con un valor de cero. Posteriormente se implementa una serie de sentencias condicionales, “*IF...THEN...ELSE...END_IF*” que determinan la activación o desactivación de distintas señales en función de la potencia requerida.

El primer bloque de código evalúa si la variable *POWER DEMAND* supera el valor de veinticinco. En caso afirmativo, se activa la señal *POT25*, lo que indica que la demanda de potencia ha sobrepasado este umbral. Si la condición no se cumple, la señal permanece desactivada. Un segundo bloque de código realiza una comprobación similar, pero en este caso con el umbral de cincuenta, activando o desactivando la señal *POT50* en consecuencia. Finalmente, el último bloque evalúa si la demanda de potencia supera el 75%, activando la señal *POT75* si esta condición se cumple y desactivándola en caso contrario.

Este enfoque permite una gestión eficiente del sistema de propulsión híbrida, ya que garantiza que los motores y la turbina se activen de manera progresiva según la potencia requerida en cada momento. A través de esta lógica, se pueden implementar estrategias de control que optimicen el consumo

energético y minimicen el desgaste de los componentes mecánicos y eléctricos. Además, la estructura del código en lenguaje estructurado proporciona claridad y modularidad, facilitando la integración con otros sistemas dentro del mismo entorno de programación.

El uso de este tipo de programación en un sistema de propulsión híbrida es fundamental para garantizar la eficiencia y seguridad operativa del barco. La activación progresiva de los motores eléctricos y la turbina de gas en función de la demanda de potencia permite no solo una mejor gestión del consumo de energía, sino también una reducción de las emisiones contaminantes y un menos desgaste de los componentes mecánicos. De este modo, la implementación de este código dentro del sistema de control contribuye a mejorar el rendimiento global del sistema, garantizando una operación segura y optimizada en diversas condiciones de navegación. Este fragmento representa una parte clave dentro de un sistema de automatización más amplio.

La Figura 3-3 muestra la edición del programa principal (*MAIN*). Posteriormente se identifican tres bloques de función denominados *TRANSITIONS_M1*, *TEMP_COUNT_M1* y *PLACES_M1*, los cuales representan diferentes módulos o funciones dentro del sistema de control, cabe a mencionar que dentro de este apartado existen más bloques de función que no se muestran en la imagen.

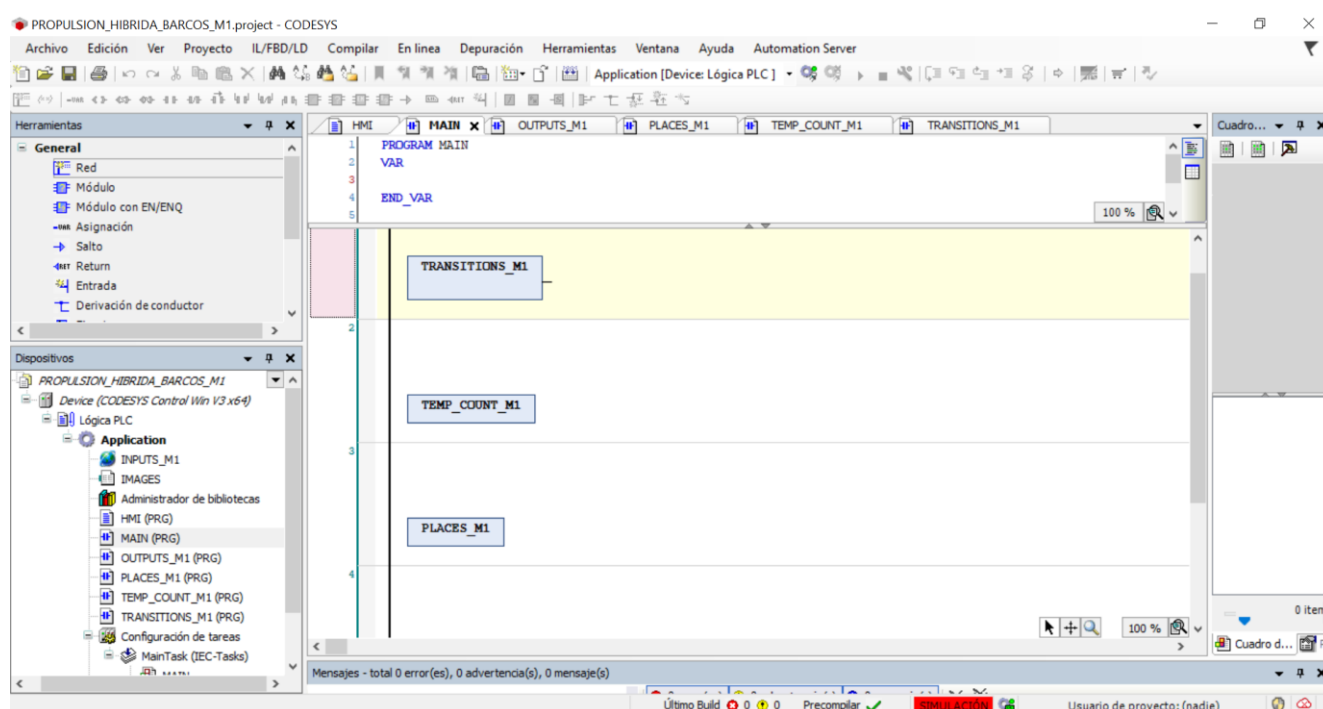


Figura 3-3 Programación Codesys del sistema de gestión de propulsión híbrida, bloque MAIN.

En la Figura 3-4 se muestra el bloque *OUTPUTS_M1*, encargado de gestionar las salidas del sistema. Dentro de este programa, se utiliza una representación gráfica estructurada del sistema mediante bloques funcionales. En la parte superior de la pantalla, específicamente en el área destinada al código del programa, se puede observar cómo se declaran las variables con la estructura *VAR...END_VAR*. En esta sección se definen variables de tipo booleano (*BOOL*) relacionadas directamente con los actuadores principales del sistema de propulsión: *ELECTRIC1*, *ELECTRIC2* y *GAS*. Estas variables se inicializan con el valor *FALSE*, lo cual indica que, al momento de iniciar el sistema, tanto los motores eléctricos como la turbina de gas se encuentran apagados por defecto.

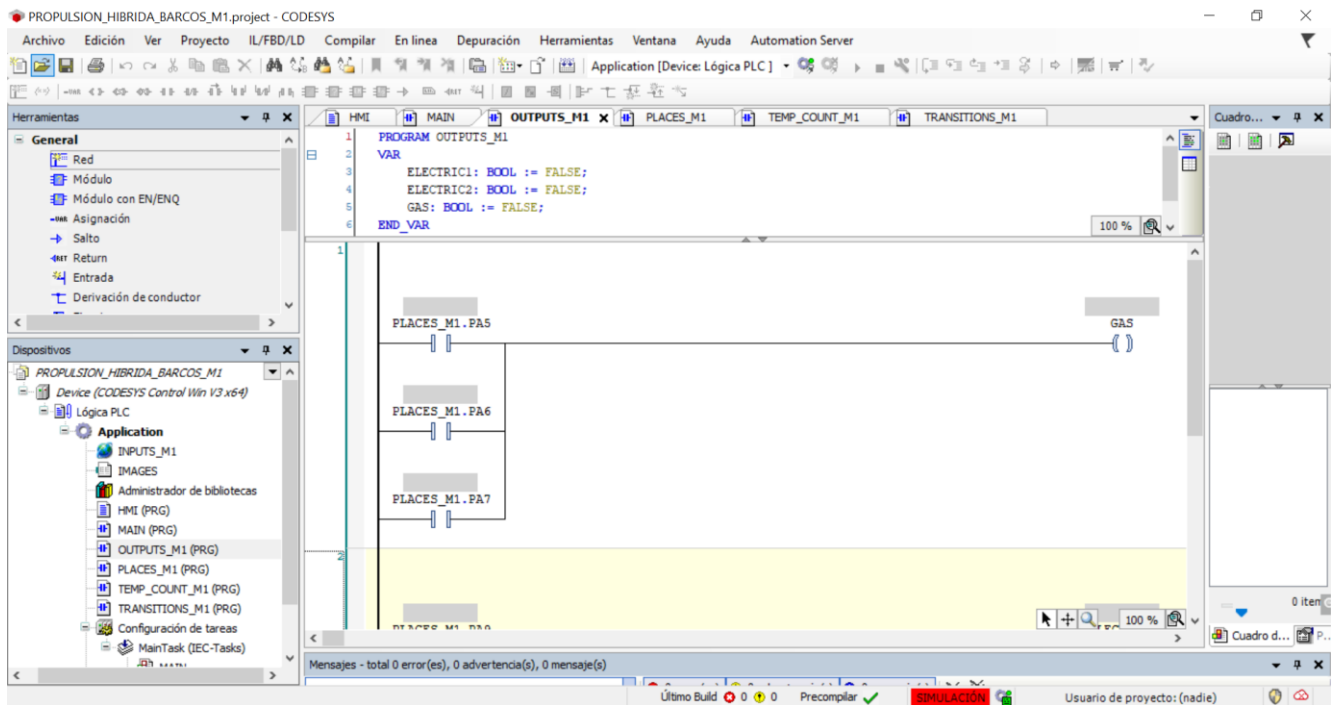


Figura 3-4 Programación Codesys del sistema de gestión de propulsión híbrida, bloque OUTPUTS.

En la sección gráfica del lenguaje LD de la Figura 3-4, se representan tres contactos normalmente abiertos asociados a las señales *PLACES_M1.PA5*, *PLACES_M1.PA6* y *PLACES_M1.PA7*. Estos contactos actúan como condiciones necesarias para la activación de la salida *GAS*. En términos de funcionamiento, esto significa que la turbina de gas solo se activará si se cumplen simultáneamente las condiciones establecidas en estas tres posiciones de la red de Petri, lo que refleja la modelización de un sistema de control basado en eventos discretos. Así se van definiendo una a una todas las salidas del sistema con los condicionantes que las activan de manera precisa y coordinada, mejorando así la seguridad y eficiencia del sistema de propulsión de la embarcación.

En la **¡Error! No se encuentra el origen de la referencia.** se puede encontrar la edición del programa *PLACES_M1*. En la parte superior se han definido las variables booleanas que representan los lugares en la red de Petri, *PA1*, *PA2*, *PA3*, *PA4*..., los cuales representan diferentes estados dentro del sistema de propulsión. Inicialmente *PA1* se encuentra en estado *TRUE*, mientras que las demás están en *FALSE*, lo que indica que el sistema arranca desde una condición específica en la cual únicamente *PA1* está activo.

En la parte gráfica se observa la relación entre los lugares y las transiciones de la red. Se han presentado transiciones, como *TRANSITIONS_M1.TA1* y *TRANSITIONS_M1.TA2* que actúan como condiciones necesarias para el cambio de estado dentro del sistema. La activación de estas transiciones permite modificar el estado de los lugares, lo que genera un comportamiento dinámico dentro del modelo de control del sistema de propulsión híbrida. Estas transiciones garantizan que los motores eléctricos y la turbina de gas se activen y desactiven de manera sincronizada.

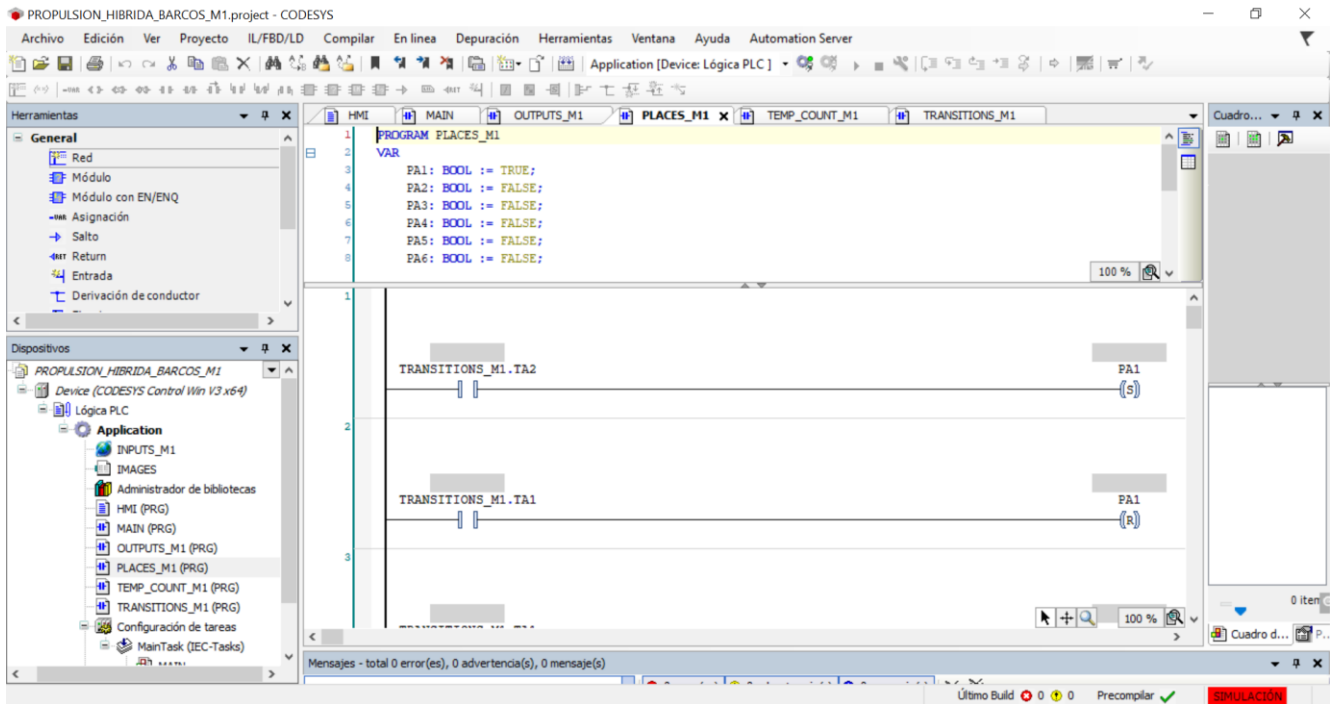


Figura 3-5 Programación Codesys del sistema de gestión de propulsión híbrida, bloque PLACES.

En la **Error! No se encuentra el origen de la referencia.** se encuentra la edición del módulo *TEMP_COUNT_M1*, que se encarga del control de temporizadores y contadores del sistema, asegurando una sincronización precisa en la ejecución de las distintas operaciones.

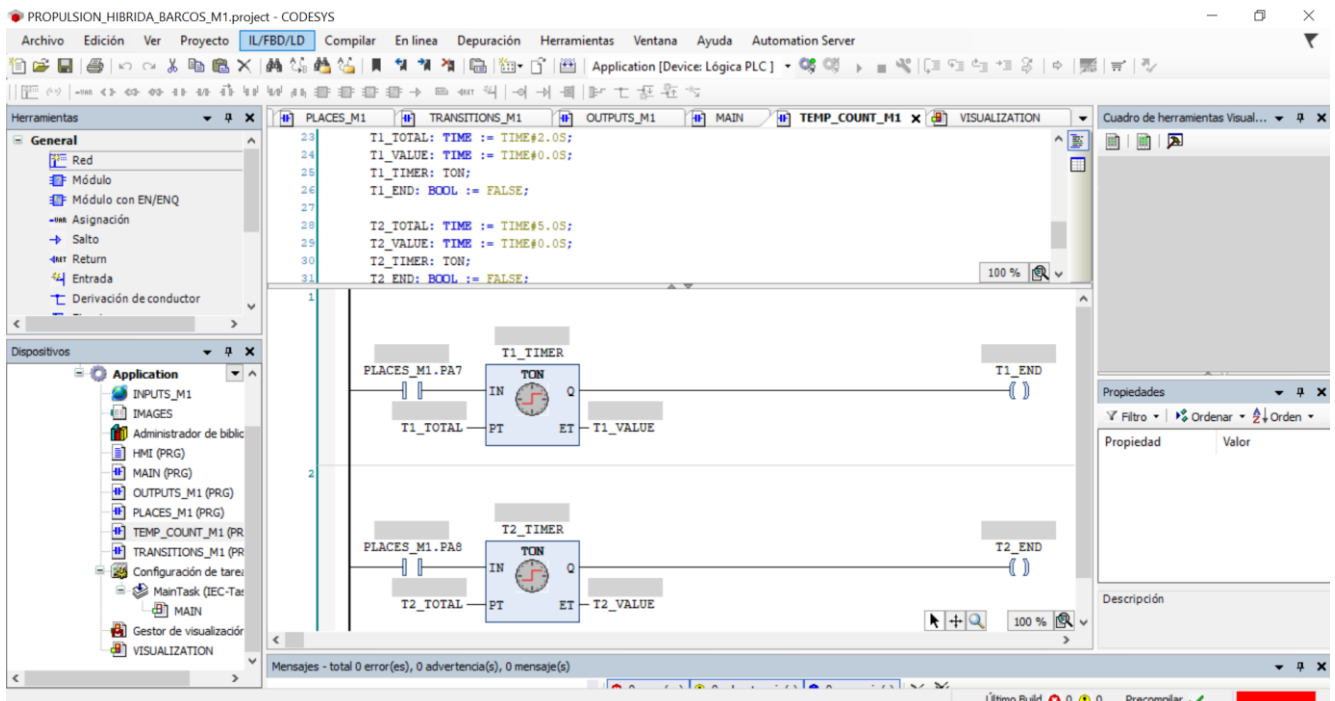


Figura 3-6 Programación Codesys del sistema de gestión de propulsión híbrida, bloque TEMP_COUNT.

También se han definido dos temporizadores de retardo (*TON*), denominados *T1_TIMER*, *T2_TIMER*, que permiten establecer retardos en la activación de ciertos procesos del sistema. Cada temporizador cuenta con variables auxiliares de tipo *TIME*, tales como *T1_TOTAL* y *T1_VALUE* en el caso del primer temporizador, y *T2_TOTAL* y *T2_VALUE* en el segundo. Estas variables determinan los

tiempos máximos y actuales de operación de cada temporizador, respectivamente. También se han definido variables booleanas como *T1_END* y *T2_END*, que indican cuando los temporizadores han alcanzado su tiempo máximo establecido, permitiendo así la transición a la siguiente etapa del proceso automatizado.

En la sección gráfica del lenguaje LD de la **¡Error! No se encuentra el origen de la referencia.**, se observan dos líneas de control donde se han implementado los temporizadores. En la primera línea, el temporizador *T1_TIMER* es activado cuando la condición *PLACES_M1.PA7*, se cumple, lo que sugiere que este temporizador depende de un estado previo. Una vez activado, el temporizador inicia la cuenta regresiva de acuerdo con el valor establecido en *T1_TOTAL*. Cuando el tiempo especificado se cumple se almacena el valor en *T1_VALUE* y se activa la salida *T1_END*, permitiendo la transición a la siguiente fase del sistema de propulsión. De manera análoga, en la segunda línea de control se observa la implementación del temporizador *T2_TIMER* de la misma manera que el anterior. La correcta sincronización de estos tiempos es esencial para garantizar un funcionamiento eficiente del sistema.

En este bloque además de definir las variables de los temporizadores, también se definen los contadores de la misma manera.

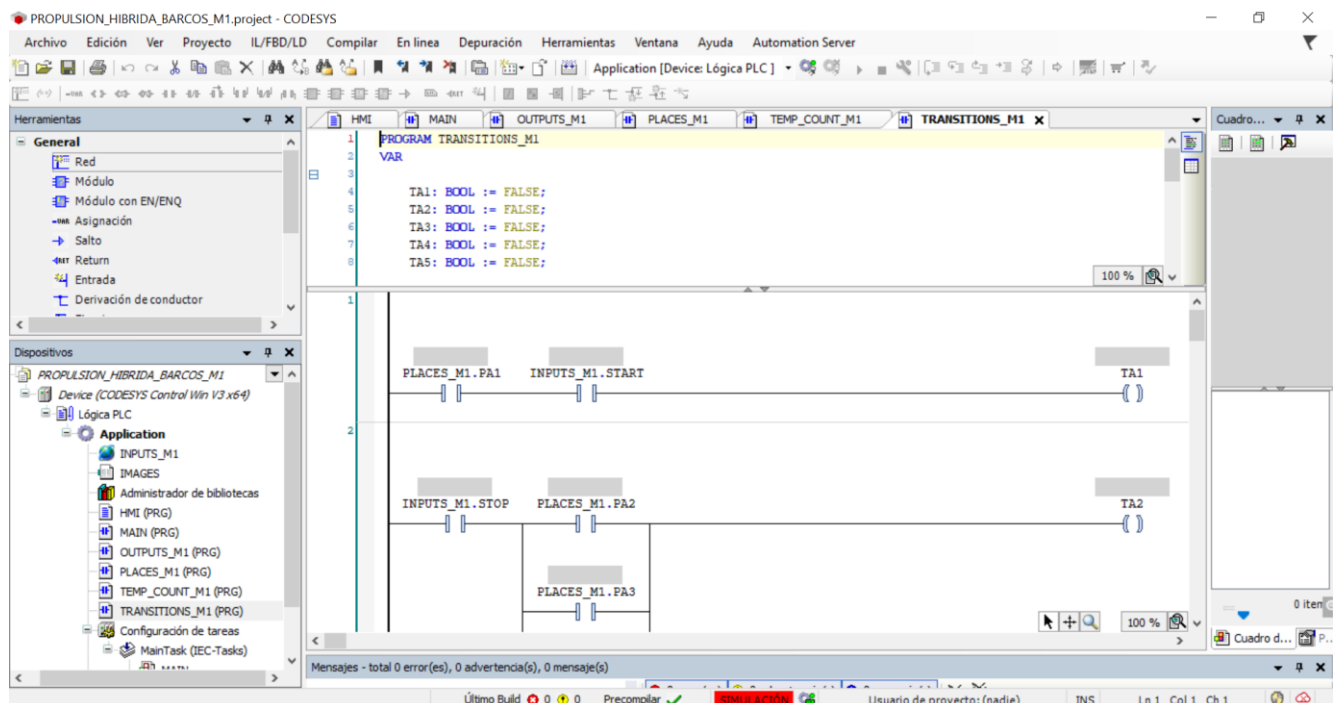


Figura 3-7 Programación Codesys del sistema de gestión de propulsión híbrida, bloque TRANSITIONS.

En este caso (Figura 3-7), se encuentra la edición del módulo *TRANSITIONS_M1*, encargado de definir las condiciones que permiten el cambio de estado entre diferentes lugares de la red, asegurando el correcto funcionamiento.

En la parte superior como en los anteriores casos se han definido las variables booleanas *TA1*, *TA2*, *TA3*, *TA4*... Todas ellas inicializadas en *FALSE*. Estas variables representan las transiciones dentro del sistema y se activarán cuando se cumplan las condiciones lógicas establecidas en la programación. El uso de estas transiciones dentro de la red de Petri permite gestionar la evolución del sistema garantizando su fluidez.

En las primeras líneas se muestran la implementación de las transiciones *TA1* y *TA2*. La primera se activará cuando la condición *PLACES_M1.PA1* esté presente y se reciba una señal de inicio *INPUTS_M1.START*. Esto indica que el sistema solo avanzará a la siguiente etapa si el lugar *PA1* está activo y la entrada de inicio ha sido pulsada. De manera similar, en la segunda línea *TA2* depende de la activación simultánea de *INPUTS_M1.STOP*, *PLACES_M1.PA2* y *PLACES_M1.PA3*, lo que indica que

para que esta transición se active, es necesario que el sistema se encuentre en los estados *PA2* y *PA3* y que se haya recibido una orden de parada. Así se establecen de la misma manera el resto de las condiciones para que se vayan produciendo las transiciones correspondientes en cada instante.

En la Figura 3-8 se observa un panel de visualización que representa los elementos clave del sistema, permitiendo a los operadores interactuar con el modelo de control mediante una serie de botones, interruptores e indicadores visuales.

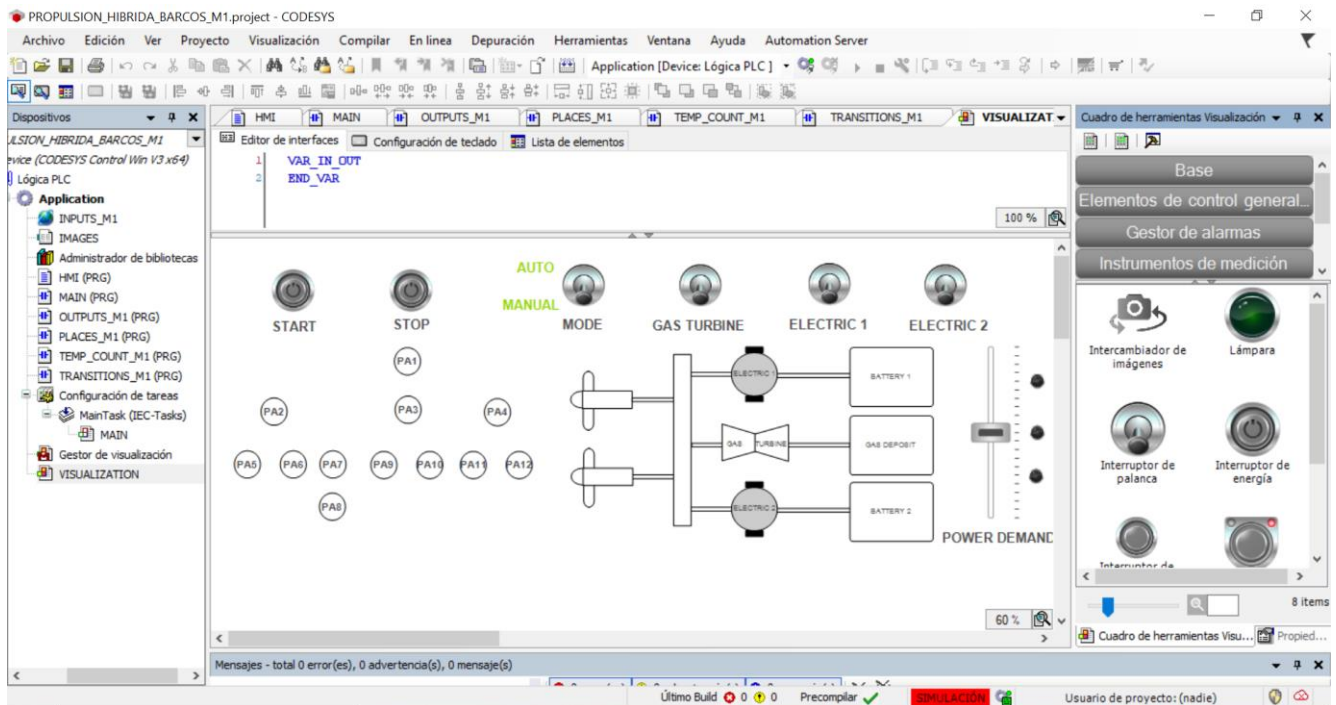


Figura 3-8 Programación Codesys del sistema de gestión de propulsión híbrida, bloque VISUALIZACIÓN.

En la parte superior del interfaz se encuentran los controles principales para la activación y desactivación del sistema, con botones etiquetados como *START* y *STOP*. Estos permiten iniciar y detener la operación del sistema de propulsión, respectivamente. Adyacente a estos controles, se muestra un selector de modo con dos opciones: *AUTO* y *MANUAL*, lo que indica que el sistema puede operar en un modo completamente automatizado o permitir la intervención manual por parte del usuario.

En el centro de la pantalla, se representan gráficamente los distintos componentes del sistema, destacando los motores eléctricos, la turbina de gas, y los sistemas de almacenamiento de energía, que incluyen dos baterías y un depósito de gas. La representación esquemática muestra como estos elementos están interconectados, permitiendo al usuario visualizar el flujo de energía dentro del sistema. Se incorpora además una barra de desplazamiento etiquetada como *POWER DEMAND*, la cual permite ajustar la demanda de potencia del sistema, afectando la activación de los motores eléctricos y la turbina de gas en función de los requerimientos operativos.

Además, en la parte inferior a los botones de entrada *START* y *STOP*, se encuentran varios indicadores circulares, los cuales corresponden a los diferentes lugares dentro de la red de Petri que modela el comportamiento del sistema. Estos lugares representan estados dentro del modelo de control, indicando la activación o inactividad de ciertos procesos y facilitando la supervisión.

A la derecha de la interfaz, se visualiza el cuadro de herramientas de visualización, que contiene diversos elementos gráficos que pueden ser utilizados para enriquecer la interfaz. Entre ellos se incluyen, por ejemplo, interruptores de palanca o lámparas indicadoras, los cuales son usados para mejorar la parte visual del sistema.

En todas las figuras, también se puede distinguir el panel izquierdo de la interfaz que muestra la estructura del proyecto, dentro del cual se organizan los diferentes módulos y subprogramas. Se observa la existencia de módulos dedicados a la gestión de entradas (*INPUTS_M1*), la interfaz hombre-máquina (*HMI*), las transiciones de estados (*TRANSITIONS_M1*), la administración de temporizadores y contadores (*TEMP_COUNT_M1*), y la representación de los lugares dentro de la red de Petri (*PLACES_M1*). Esto sugiere una arquitectura modular que permite una clara segmentación de las diferentes funciones dentro del sistema de control, facilitando su gestión, mantenimiento y escalabilidad.

Además, también en la parte inferior de la pantalla, se visualiza la barra de estado del sistema, la cual indica que no existen errores, advertencias, ni mensajes pendientes en la compilación del programa, lo que sugiere que la lógica de control ha sido correctamente implementada. Además, se observa que la simulación está activada, lo que permite verificar el comportamiento del sistema sin necesidad de una implementación física, asegurando una validación previa del código antes de su despliegue en hardware real.

En las figuras (de la Figura 3-9 a la Figura 3-11) se muestran ejemplos del funcionamiento del interfaz visual. Estas representan los distintos modos de operación del sistema y se pueden observar modos de funcionamiento: manual y a automático. En modo manual, la activación de los motores eléctrico o la turbina de gas depende directamente del operador, quien decide qué componentes encender o apagar. En modo automático, el sistema ajusta automáticamente la activación de los motores y turbina según el nivel de demanda de potencia establecido en la barra de control. Concretamente, en la Figura 3-9, el sistema se encuentra en modo manual, con el botón *STOP* activado y el lugar *PA1* resaltado en rojo, indicando que el sistema está apagado o en estado de reposo. No hay motores ni turbina en funcionamiento.

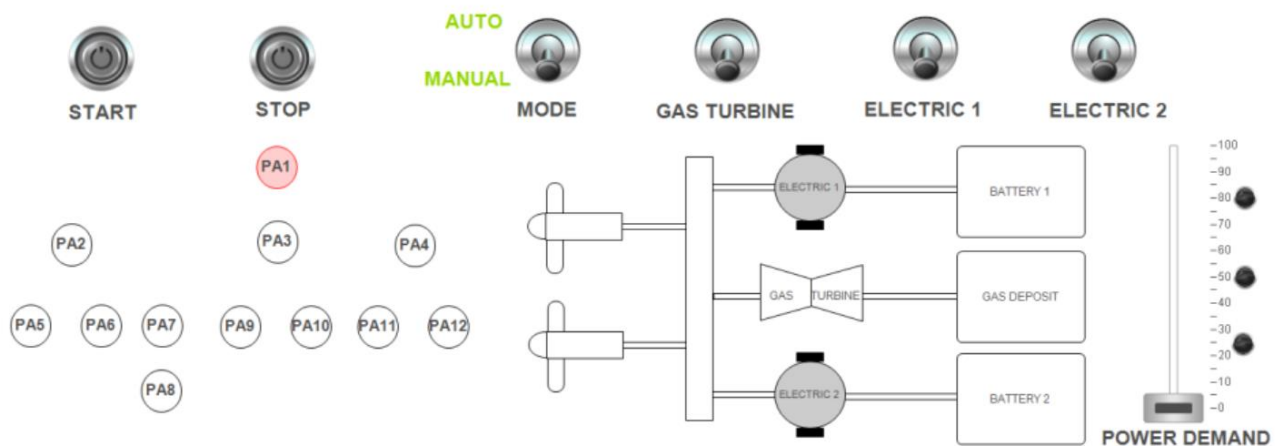


Figura 3-9 Representación visual del modelo del sistema de gestión de propulsión híbrida apagado.

Luego, en la Figura 3-10, el sistema sigue en modo manual, y el operador ha activado el motor *ELECTRIC1* (resaltado en verde) y la turbina de gas (resaltado en rojo), lo que indica que ambas fuentes de energía están operando en paralelo. Varios lugares de la red de Petri (*PA5*, *PA9*, *PA4*) han cambiado a rojo, lo que representa la transición a un estado de operación.

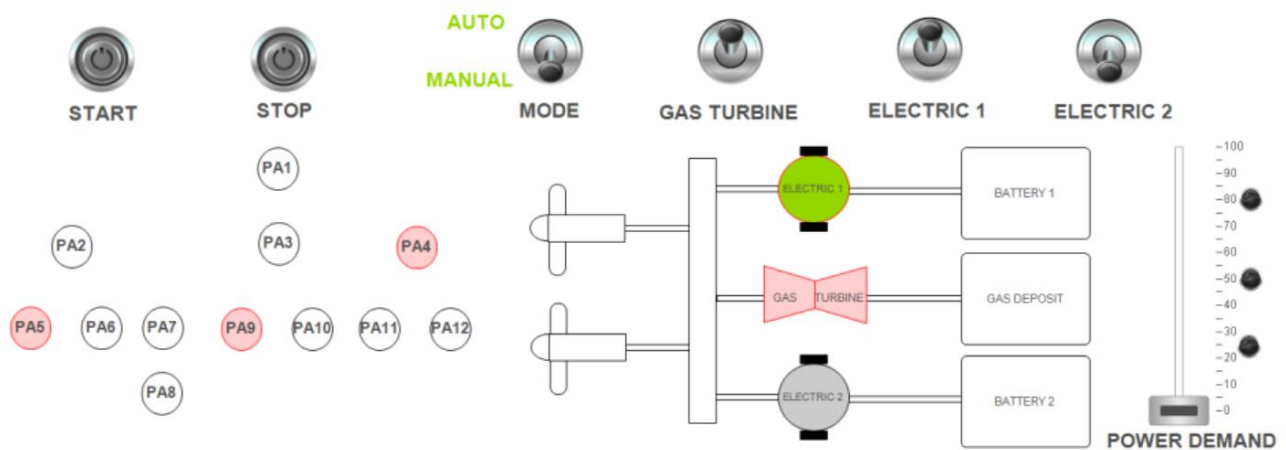


Figura 3-10 Representación visual del modelo del sistema de gestión de propulsión híbrida encendido, en modo manual, con la turbina de gas y motor eléctrico 1 activado.

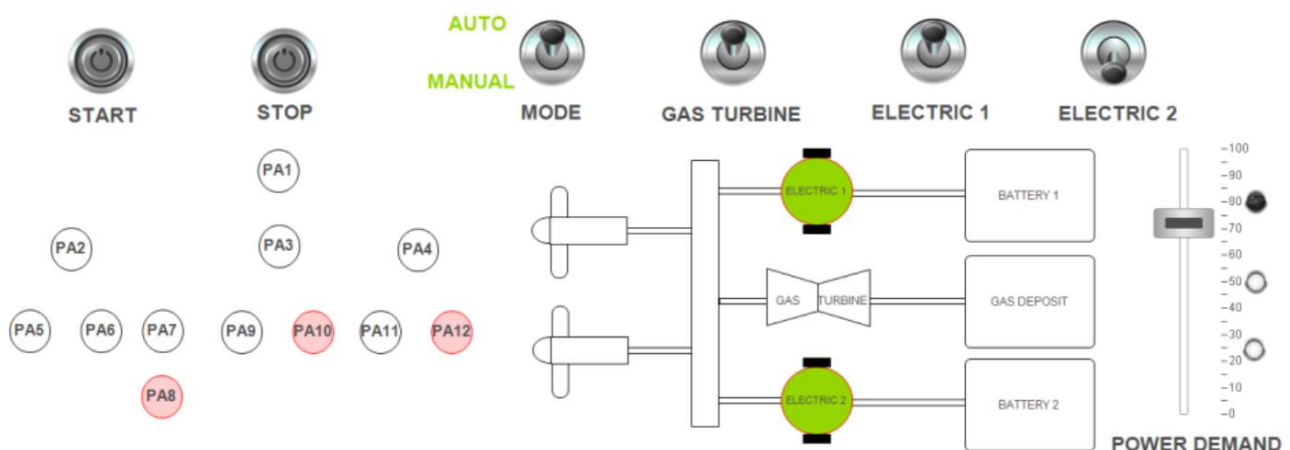


Figura 3-11 Representación visual del modelo del sistema de gestión de propulsión híbrida encendido, en modo automático, con una prestación del 75 por ciento de potencia.

Finalmente, en la Figura 3-11, el sistema ha cambiado a modo automático, lo que significa que la activación de los motores depende del porcentaje de potencia configurado en la barra. En este caso, la barra de *POWER DEMAND* ha aumentado hasta un 75%, lo que ha llevado al sistema a activar automáticamente ambos motores eléctricos, resaltados en verde, mientras la turbina de gas está en modo intermitente en función de los temporizadores asignados a este, así optimizando el consumo de combustible. Los lugares de la red de Petri PA8, PA10 y PA12 han cambiado a rojo, lo que indica la transición exitosa a este estado de operación.

Existen más estados posibles del sistema; estos son solo algunos ejemplos. Se ha realizado un video mostrando el funcionamiento del sistema, se puede acceder usando el siguiente enlace:

<https://drive.google.com/file/d/1KdK5o5y5q-aoxLMEf5Dv8uNohItvnkDk/view?usp=sharing>

3.1.3 Módulo 2: Gestión del mantenimiento preventivo del sistema de propulsión híbrido

El mantenimiento preventivo es un aspecto esencial en la operación de los sistemas de propulsión híbrido en barcos, ya que permite garantizar su fiabilidad, eficiencia y seguridad operativa. En este módulo, se desarrolla un sistema automatizado para la gestión del mantenimiento preventivo, con el objetivo de optimizar la disponibilidad y prolongar la vida útil de los componentes clave del sistema.

A diferencia del mantenimiento correctivo, que reacciona ante fallos ya ocurridos, el mantenimiento preventivo busca anticiparse a estos mediante tareas programadas de inspección, limpieza y recarga de recursos energéticos. En el contexto de los sistemas híbridos de propulsión, donde la integración de motores eléctricos y turbinas de gas conlleva una mayor complejidad técnica, es fundamental establecer protocolos estructurados que permitan mantener en óptimas condiciones los elementos mecánicos, eléctricos y de almacenamiento de energía.

El sistema automatizado de mantenimiento preventivo implementado en este módulo incorpora procesos de limpieza de los motores eléctricos y la turbina de gas, recarga de baterías y repostaje de la caldera de gas en función del número de ciclos de operación registrados. Para ello, se emplean sensores y contadores que monitorizan el estado de los componentes y activan los procedimientos de mantenimiento de manera autónoma, minimizando la intervención humana y reduciendo el riesgo de fallos inesperados.

A través del modelado con redes de Petri, se representa el comportamiento del sistema de mantenimiento y su integración con el sistema de propulsión, asegurando que todas las tareas se ejecuten en el momento adecuado y bajo las condiciones óptimas. La automatización de estos procesos no solo mejora la eficiencia operativa del barco, sino que también contribuye a la sostenibilidad ambiental al evitar consumos innecesarios de energía y reducir el desgaste prematuro de los equipos.

En este módulo, se analizarán en detalle los procedimientos de mantenimiento implementados, la lógica de operación de las redes de Petri utilizadas y la estructura del sistema automatizado, proporcionando una solución efectiva y adaptable a distintos entornos de operación naval.

3.1.3.1 Enunciado

Para realizar la automatización del sistema de propulsión descrito se deben considerar los siguientes aspectos:

- El sistema de mantenimiento comienza estando apagado y se activa una vez se pulsa el botón **START**, a la vez que el propio sistema de propulsión.
- Un botón de parada que genera la señal de apagado del sistema **STOP**. En caso de recibir esta señal se debe producir un apagado general, es decir, el de ambas redes de Petri. Para ello:
 - Debe modificarse la red del módulo 1 para que se apague incluso si alguno la turbina o alguno de los motores está en uso.
 - Si es necesario, se debe esperar a que finalice un proceso de recarga (de una batería o de la caldera de gas) y a continuación apagar la red del módulo 2.
- Nada más encender el sistema, automáticamente se produce un lavado simultáneo tanto de los motores eléctricos como de la turbina de gas. Para ello se deben ejecutar a la vez los siguientes procesos simultáneos:
 - Lavar los motores eléctricos empleando las instrucciones **CLEAN E1** y **CLEAN E2**. Este proceso dura 5 segundos.
 - Lavar la turbina de gas ejecutando la instrucción **CLEAN TURBINE**. Existe un sensor de la finalización de limpiado de la turbina que genera la señal **TURBINE CLEANED** cuando ha finalizado esta limpieza.

- Una vez terminada la limpieza de los motores eléctricos y la turbina de gas, el sistema quedará a la espera para realizar la siguiente acción dependiendo del número de ciclos de uso de cada uno de los elementos de la propulsión, por medio del control del valor de los contadores **CG**, **CE1**, y **CE2** definidos en el primer módulo.
- A continuación, se deben gestionar los siguientes procesos:
 - **RECARGA DE GAS:** Una vez la turbina haya realizado 6 ciclos de encendido, el propio sistema, de manera automática, debe realizar la recarga de la caldera que alimenta la turbina de gas, para lo cual debe ejecutar la instrucción **RECHARGE GAS**, terminando la acción cuando un indicador activa la señal **GAS RECHARGED**. Cuando sucede el proceso de recarga el contador de ciclos de la turbina se reinicia, de forma que se indique que vuelve a estar cargada la caldera.
 - **RECARGA DE MOTORES ELÉCTRICOS:** En primer lugar, hay que tener en cuenta que el mecanismo que se encarga de realizar la recarga de los motores eléctricos es compartido por ambas baterías, de forma que no se permite la carga de ambas baterías a la vez. Es decir, en caso de necesidad de recarga de los dos motores solo uno de ellos podrá recargarse, y el otro tendrá que esperar a que finalice la recarga del primero para poder comenzar con la suya. Por otra parte, en caso de que ambos motores quieran iniciar una recarga en el mismo instante, el motor eléctrico 1 tiene preferencia de acceso sobre el motor eléctrico 2. Los procesos de recarga de las baterías son los siguientes:
 - **RECARGA DE LA BATERÍA DEL MOTOR ELÉCTRICO 1:** La carga de la batería 1 se debe ejecutar cada 10 encendidos del motor eléctrico 1, para lo cual hay que ejecutar la instrucción **RECHARGE B1**. La recarga de la batería 1 finaliza cuando se active la señal **B1 RECHARGED**. En este mismo momento el contador de ciclos del motor eléctrico 1 se reiniciará para indicar que la batería vuelve a estar cargada.
 - **RECARGA DE LA BATERÍA DEL MOTOR ELÉCTRICO 2:** La carga de la batería 2 se debe ejecutar cada 8 encendidos del motor eléctrico 2, para lo cual hay que ejecutar la instrucción **RECHARGE B2**. La recarga de la batería 2 finaliza cuando se active la señal **B2 RECHARGED**. En este mismo momento el contador de ciclos del motor eléctrico 2 se reiniciará para indicar que la batería vuelve a estar cargada.

3.1.3.2 Tabla de variables

En el módulo de gestión de mantenimiento preventivo, la automatización del sistema requiere el uso de diversas variables que permiten supervisar y ejecutar las tareas de mantenimiento de manera eficiente. Estas variables se organizan en cuatro categorías principales representadas en la Tabla 3-2: entradas, salidas, contadores y temporizadores, cada una con un rol específico en el control del mantenimiento preventivo del sistema de propulsión híbrida.

Las entradas incluyen señales que indican eventos clave en el mantenimiento, como el inicio y la detención del sistema (**START** y **STOP**), la detección de la finalización del lavado de la turbina (**TURBINE CLEANED**), y la confirmación de procesos de recarga de gas y baterías (**GAS RECHARGED**, **B1 RECHARGED** y **B2 RECHARGED**). Estas señales aseguran que el sistema responda adecuadamente a las condiciones operativas establecidas.

Por otro lado, las salidas activan las acciones necesarias para ejecutar el mantenimiento preventivo. Entre ellas se encuentran las instrucciones de limpieza de los motores eléctricos (**CLEAN E1** y **CLEAN**

E2) y de la turbina de gas (*CLEAN TURBINE*), así como los comandos de recarga de combustible y baterías (*RECHARGE GAS*, *RECHARGE B1* y *RECHARGE B2*), que garantizan la disponibilidad energética del sistema.

Los contadores (*CG*, *CE1* y *CE2*) registran el número de ciclos de operación de la turbina de gas y los motores eléctricos, lo que permite programar tareas de mantenimiento basadas en su uso acumulado. Finalmente, los temporizadores (*TB1*) regulan la duración de ciertos procesos de mantenimiento, como el tiempo de limpieza de los motores y la turbina, asegurando su correcta ejecución.

El uso combinado de estas variables permite una gestión automatizada y eficiente del mantenimiento preventivo, optimizando una gestión automatizada y eficiente del mantenimiento preventivo, optimizando la operatividad del sistema y minimizando el riesgo de fallos inesperados.

Entradas	Salidas	Contadores	Temporizadores
START	CLEAN E1	CG	TB1
STOP	CLEAN E2	CE1	
TURBINE CLEANED	CLEAN TURBINE	CE2	
GAS RECHARGED	RECHARGE GAS		
B1 RECHARGED	RECHARGE B1		
B2 RECHARGED	RECHARGE B2		

Tabla 3-2 Variables del sistema de gestión de mantenimiento preventivo.

3.1.3.3 Red de Petri del mantenimiento predictivo

En esta sección, se presenta la red de Petri del sistema de gestión del mantenimiento preventivo de la propulsión híbrida en la **¡Error! No se encuentra el origen de la referencia..** Esta red modela de manera gráfica y estructurada el comportamiento del sistema en relación con las tareas de limpieza, recarga de gas y recarga de baterías. A diferencia del módulo anterior, donde se gestionaba exclusivamente la propulsión, en esta red se introduce una nueva característica clave: la implementación de recursos compartidos, lo que añade un nivel adicional de complejidad en la gestión de las transiciones.

Una de las principales innovaciones en este módulo es la gestión de la recarga de baterías mediante un recurso compartido, lo que significa que ambas baterías no pueden recargarse simultáneamente, ya que dependen de un mismo mecanismo de carga. Este recurso compartido se modela mediante lugares y transiciones que regulan el acceso a la recarga, garantizando que cuando una batería está en proceso de carga, la otra debe esperar a que finalice para poder iniciar su propio proceso.

El control del recurso compartido se representa en la red mediante plazas intermedias (*PB9*, *PB10*, *PB12* y *PB13*) y transiciones que establecen la prioridad de carga mediante arcos inhibidores. En este caso, la batería del motor eléctrico 1 tiene prioridad sobre la batería del motor eléctrico 2, lo que se traduce en la estructura de la red de Petri. Si ambas baterías necesitan ser recargadas al mismo tiempo, el sistema concede acceso al motor eléctrico 1 primero y solo cuando este ha finalizado su proceso (*B1 RECHARGED*), se habilita la transición que permite el inicio de la recarga del motor eléctrico 2.

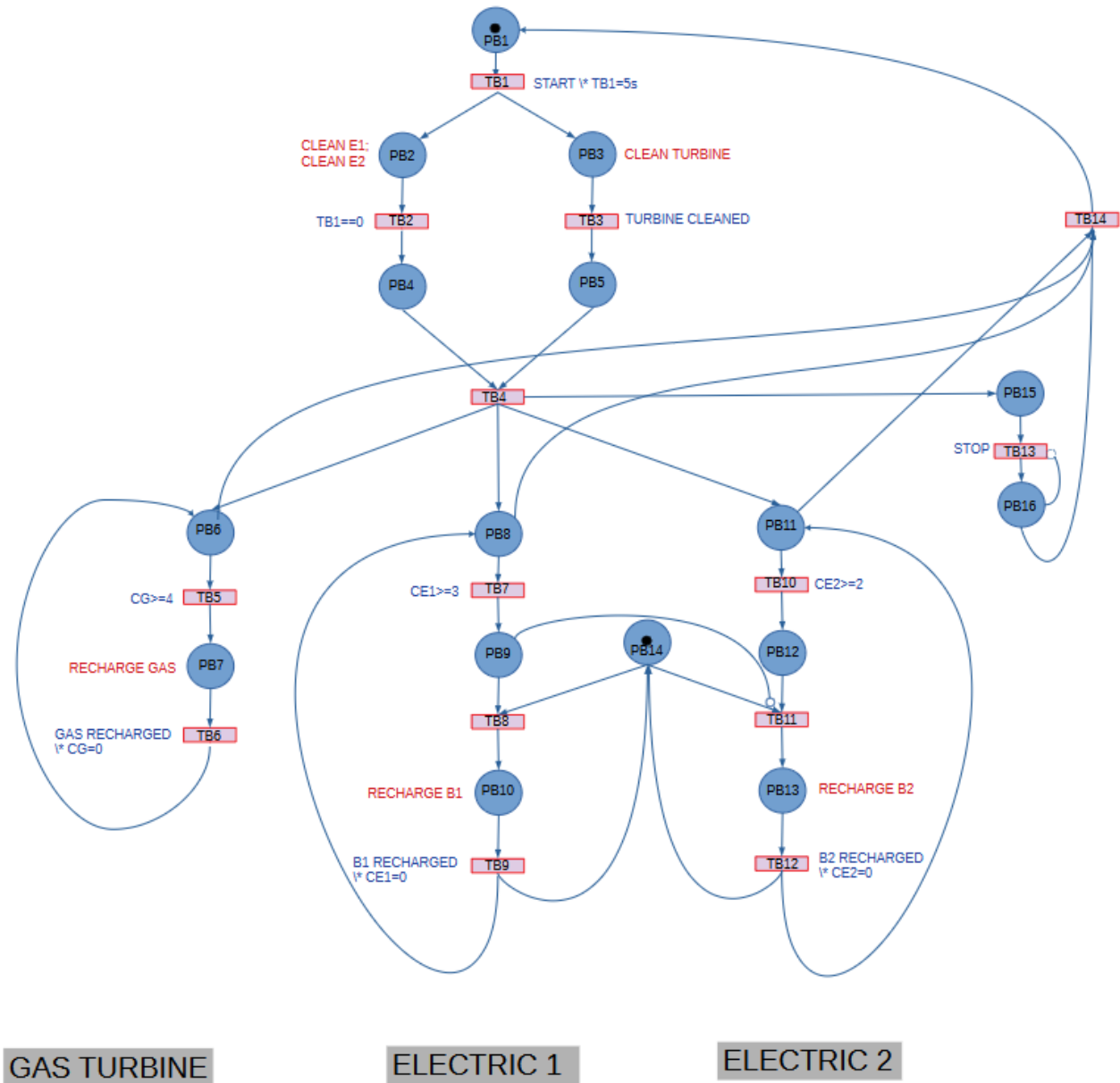


Figura 3-12 Red de Petri del sistema de gestión del mantenimiento preventivo de la propulsión híbrida.

Con este modelo permite una gestión eficiente del recurso de carga, evitando situaciones de conflicto y optimizando el uso de los sistemas de almacenamiento energético del barco.

En cuanto al flujo de operación de la red, el sistema se activa con la señal *START*, iniciando la limpieza simultánea de los motores eléctricos (*CLEAN E1* y *CLEAN E2*) y de la turbina de gas (*CLEAN TURBINE*). Esta fase está regulada por un temporizador (*TB1*), que asegura que el proceso dure un tiempo determinado antes de permitir la siguiente transición. Una vez finalizado el proceso de limpieza (*TURBINE CLEANED*), la red de Petri evalúa el número de ciclos de operación (*CG*, *CE1* y *CE2*) para determinar si es necesario proceder con la recarga de gas o de baterías. Si la turbina ha alcanzado el número determinado de ciclos establecidos, que son 4 ($CG \geq 4$), se activa el proceso *RECHARGE GAS*. La red de Petri controla que este proceso finalice correctamente antes de continuar con la siguiente tarea. Si el número de ciclos de uso del motor eléctrico 1 es mayor o igual a 3 ($CE \geq 3$), se inicia la recarga de la batería 1 (*RECHARGE B1*), bloqueando temporalmente la posibilidad de recargar el motor eléctrico 2. Cuando *B1 RECHARGED* se activa, la red habilita la transición del motor eléctrico 2. Y si $CE2 \geq 2$, se ejecuta *RECHARGE B2*, asegurando que la batería 2 se recarga solo después de que la 1 haya

terminado. Una vez todas las tareas de mantenimiento han sido ejecutadas, el sistema vuelve a un estado de espera hasta la siguiente activación.

La red de Petri de mantenimiento preventivo no solo permite modelar el comportamiento general del sistema, sino que introduce la gestión eficiente de recursos compartidos, en este caso, el sistema de recarga de baterías. Esta implementación evita conflictos en la asignación de energía, optimizando el rendimiento del sistema y asegurando una operación confiable en el tiempo. Esta metodología puede extrapolarse a otros procesos donde múltiples sistemas dependan de un mismo recurso, como la gestión de combustible o la distribución de energía en embarcaciones híbridas.

3.1.3.4 Programación y visualización

En la Figura 3-13 se muestra una parte del bloque de programación *TRANSITIONS_M2*, específicamente la lógica implementada en la transición *TB11*, la cual regula la activación del proceso de recarga del motor eléctrico 2. En este fragmento de código en lenguaje LD, se puede observar cómo se traduce el arco inhibidor de la red de Petri al esquema de programación, utilizando un contacto negado que restringe la ejecución de la recarga del motor eléctrico 2 mientras el motor eléctrico 1 se encuentre en proceso de carga.

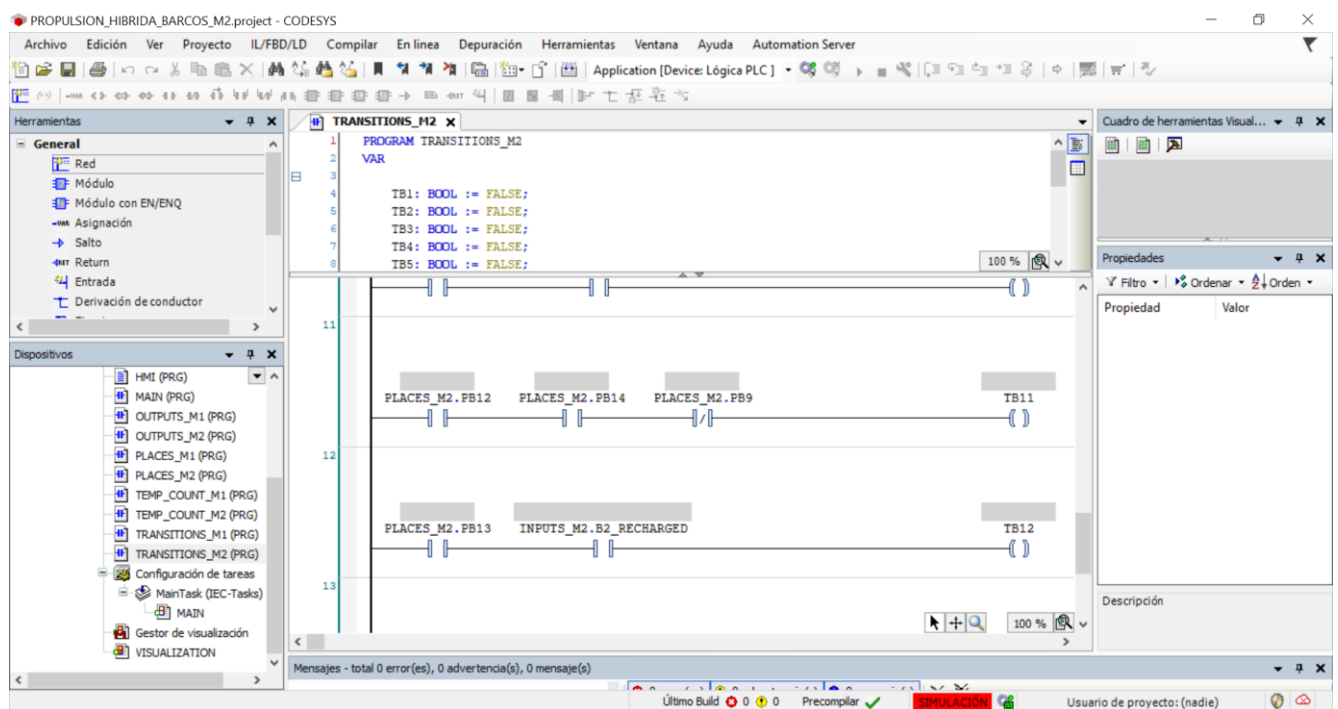


Figura 3-13 Programación Codesys del sistema de gestión de mantenimiento, bloque TRANSITIONS.

Como se ha mencionado anteriormente el arco inhibidor se implementa en LD mediante un contacto negado asociado al lugar de espera del motor eléctrico 1 (*PLACES_M2.PB9*). Esta condición asegura que mientras el motor eléctrico 1 esté a la espera de ser recargado, la transición *TB11* no puede activarse, impidiendo que el motor eléctrico 2 inicie su proceso de carga hasta que el recurso compartido esté disponible.

Por tanto, la línea 11 del código es la relevante, siendo *PLACES_M2.PB12* el lugar que representa el estado de preparación del motor eléctrico 2 para su recarga, *PLACES_M2.PB14*, el lugar que representa el recurso compartido (es decir, el sistema de recarga de baterías), asegurando que este debe estar disponible para que pueda activarse la carga del motor eléctrico correspondiente y *PLACES_M2.PB9* (un contacto negado), el lugar que indica la espera del motor eléctrico 1 para su recarga, que al estar negado impide la activación de *TB11* si el motor eléctrico 1 aún está en espera de carga, siendo *TB11* la transición que, al activarse, inicia la recarga del motor eléctrico 2.

En definitiva, se trata de una lógica en la que es necesario que se cumplan simultáneamente las condiciones descritas en el anterior párrafo: *PLACES_M2.PB12* debe estar activo, *PLACES_M2.PB14* debe estar disponible, y *PLACES_M2.PB9* debe estar desactivado. Es decir, *B2* está listo para la recarga, garantizando que no hay otro proceso en curso y que el motor eléctrico 1 haya finalizado su carga y no está esperando para acceder al recurso compartido.

En este sentido, la Figura 3-14 representa la interfaz del sistema de gestión y mantenimiento preventivo de la propulsión híbrida en barcos. Se observa una representación visual de los diferentes estados del sistema, incluyendo la red de Petri, los motores eléctricos, la turbina de gas, los contadores de ciclos y los nuevos interruptores de activación del mantenimiento. A continuación, se analiza el estado actual del sistema teniendo en cuenta el recurso compartido, las nuevas implementaciones de los indicadores de contadores, y los interruptores recientemente añadidos.

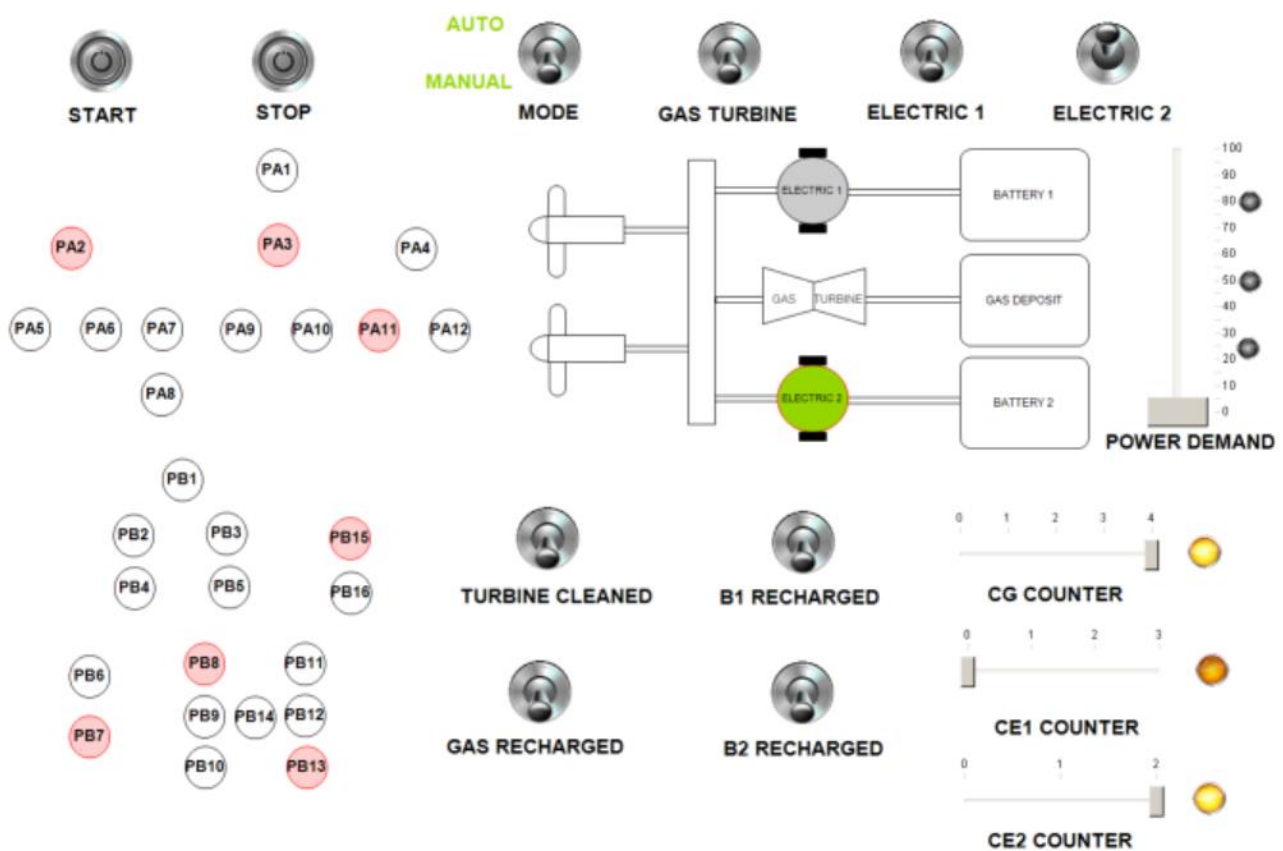


Figura 3-14 Representación visual del modelo del sistema de gestión de mantenimiento recargando la turbina de gas y el motor eléctrico 2.

Se observa a la izquierda la red de Petri con *PB13* activado, lo que indica que la recarga del motor eléctrico 2 aún no ha finalizado y dado que el sistema de recarga de baterías es un recurso compartido, la batería del motor eléctrico 1 no podrá iniciar su recarga hasta que el 1 no finalice la suya. Además, también podemos ver *PB7* activo, lo cual indica la recarga de gas de la turbina.

A la derecha de la Figura 3-14, se han añadido nuevos indicadores de contadores, los cuales permiten visualizar los ciclos de operación de los motores y turbina de gas. Actualmente *CG COUNTER* se encuentra en 4 ciclos, lo que indica que se ha cumplido el número de ciclos requeridos para que la turbina necesite una recarga de gas, por ello se encuentra iluminada la lámpara que se encuentra al lado. Además, *CE1 COUNTER* marca 0 ciclos, lo que sugiere que para que necesite una recarga y la lámpara se ilumine

le quedan un restante de 3 ciclos. Y, por último, el *CE2 COUNTER* indica 2 ciclos, con la lámpara encendida, lo que implica que el motor eléctrico 2 necesita una recarga.

Como se ha mencionado anteriormente se han agregado cuatro nuevos interruptores, *TURBINE CLEANED*, que permite indicar que la turbina ha sido limpiada. *B1 RECHARGED*, *B2 RECHARGED* y *GAS RECHARGED*, que confirman que los motores eléctricos uno, dos y la turbina de gas han sido recargados respectivamente. Estos componentes permiten validar el estado del mantenimiento de cada componente del sistema, asegurando que las tareas de recarga y limpieza sean ejecutadas correctamente antes de habilitar nuevas transiciones en la red de Petri.

Existen más estados posibles del sistema; estos son solo algunos ejemplos. Se ha realizado un video mostrando el funcionamiento del sistema, se puede acceder usando el siguiente enlace:

<https://drive.google.com/file/d/1CSUNRjzTkYTbu2lw-fHYhIT2C97M9RFu/view?usp=sharing>

3.1.4 Módulo 3: Gestión de alarmas y fallos del sistema de propulsión híbrida

En los dos primeros módulos del sistema de automatización propuesto, se han presentado aspectos relacionados con la programación de gestión de procesos que operan en paralelo, programación de secuencias básicas, o el uso de temporizadores y contadores en el primer módulo, así como la sincronización de procesos simultáneos o el uso de recursos compartidos en el segundo módulo. En el tercer y último módulo, se incluirán formas de comunicación entre redes de Petri para conseguir elaborar sistemas modulares colaborativos. Para realizar esta tarea se van a emplear dos métodos diferentes para comunicar dos redes de Petri entre sí:

- Emplear variables de estado de lugares de una red como entradas en otra red.
- Emplear arcos que conecten diferentes elementos de ambas redes.

3.1.4.1 Enunciado

Para realizar la automatización del sistema de gestión de fallos y alarmas se deben considerar los siguientes aspectos:

- En primer lugar, se requiere que los motores eléctricos y la turbina de gas no puedan ser encendidos hasta que el proceso de limpieza inicial presentado en el módulo 2 haya finalizado. Para corregir esta situación, es necesario modificar la red del módulo 1 incluyendo un nuevo lugar de espera (*PA13*) que durante el arranque espere a permitir activar el encendido de motores eléctricos o la turbina hasta que el proceso de limpieza expuesto en el módulo 2 haya finalizado, evento que puede ser indicado mediante la activación del lugar *PB15* en la red del módulo 2, es decir, se empleará el estado del lugar *PB15* como una nueva entrada en la red del módulo 1.
- A continuación, es necesario desarrollar una red para la gestión de los fallos en la turbina, para lo cual se dispone de un sensor de vibraciones que en caso de un fallo de funcionamiento genera la señal ***TURBINE FAIL***. En este caso, siempre que no se esté ejecutando un proceso de recarga de la caldera de gas, se inicia el proceso de reparación ejecutando la instrucción ***REPAIR TURBINE*** que finaliza al recibir la señal ***TURBINE REPAIRED***. Para evitar que se active la reparación mientras se está cargando la caldera, se emplea un arco inhibidor que conecta el lugar *PB7* de la red del módulo 2 con la transición *TC2* de la nueva red del módulo 3, de forma que mientras se está recargando la caldera (*PB7* activo) no se lance el proceso de reparación de la turbina. Por otra parte, también es necesario bloquear el acceso a la recarga de la caldera de gas si ya se está realizando una reparación, para lo cual, en este caso, se empleará el estado del lugar

PC2 de la nueva red del módulo 3, como una entrada en la red del módulo 2, para evitar que se inicie la recarga si *PC2* no se encuentra activo. Para finalizar, es necesario modificar la red del módulo 1 para evitar que se encienda la turbina de gas si se está ejecutando un proceso de reparación, para lo cual se empleará el estado del lugar *PC2* de la nueva red del módulo 3 como una entrada en la red del módulo 1. De esta forma es necesario, por una parte, evitar que se pueda activar el encendido de la turbina si este lugar no se encuentra activo, y, por otra parte, apagar el encendido de la turbina si se detecta un fallo que activa su reparación, y por lo tanto desactivando el lugar *PC2*.

- A continuación, es necesario desarrollar dos redes independientes para gestionar los fallos, de forma similar a la turbina, en cada uno de los motores eléctricos. Para ello se dispone de los sensores térmicos que emiten la señales ***ELE1 FAIL*** y ***ELE2 FAIL***, que indican la presencia de un fallo en los motores eléctrico 1 y 2, en cuyo caso se deberían activar las instrucciones de reparación ***REPAIR ELE1*** y ***REPAIR ELE2*** que finalizan al recibir la señales ***ELE1 REPAIRED*** y ***ELE2 REPAIRED***, respectivamente. De igual forma que en el caso de la turbina se establece que:
 - El proceso de reparación de uno de los motores eléctricos no puede empezar si ya se encuentra realizando una recarga de su batería, para lo cual se empleará un arco inhibidor desde los lugares *PB10* y *PB13* a las transiciones *TC6* y *TC10*, respectivamente. Por otra parte, tampoco debe ser posible iniciar la recarga de una de las baterías si el sistema se encuentra realizando una reparación en su motor, para lo cual se empleará el estado de los lugares *PC5* y *PC9* para bloquear el acceso a la recarga en la red del módulo 2.
 - Por último, se empleará el estado de los lugares *PC5* y *PC9* para bloquear el acceso al encendido de los motores en la red del módulo 1, así como para establecer condiciones de apagado de los motores si se produce un fallo mientras están en funcionamiento.
- Por último, se quieren implementar medidas de seguridad adicionales, para lo cual debe incorporarse la siguiente lógica al sistema desarrollado:
 - Mientras se realiza la recarga de una de las baterías o de la caldera de gas, los motores eléctricos y la turbina deben apagarse y no permitir su activación hasta finalizar la recarga. Para solucionar esta medida se emplearán los estados de los lugares *PB7*, *PB8* y *PB11* en la red que gestiona la activación de los motores asociada al módulo 1.
 - Si se produce un fallo en un motor o la turbina mientras se está recargando, al finalizar el proceso de recarga se debe entrar en el modo de gestión del fallo de forma automática.

3.1.4.2 Tabla de variables

Como se ha mencionado anteriormente este módulo se centra en la gestión de alarmas y fallos en el sistema de propulsión híbrida, asegurando que los motores eléctricos y la turbina de gas operen de manera segura y eficiente. La Tabla 3-3 representa las entradas, salidas, contadores y temporizadores que intervienen en este proceso, permitiendo la implementación de mecanismos de detección, respuesta y recuperación ante fallos en los componentes clave del sistema.

Las entradas desempeñan un papel fundamental, ya que, incluyen señales que activan o desactivan el sistema, así como las señales *START* y *STOP*, las cuales permiten la activación y desactivación del sistema, asegurando que todos los procesos de detección de fallos y alarmas funcionen correctamente en el modo automático. Posteriormente, se identifican las señales específicas para la detección de fallos en los componentes clave del sistema de propulsión. *TURBINE FAIL* indica la detección de un fallo en la turbina de gas mediante un sensor de vibraciones, lo que activa el protocolo de reparación. De manera

similar, *ELE1 FAIL* y *ELE2 FAIL* corresponden a fallos detectados en los motores eléctricos 1 y 2, respectivamente, a través de sensores térmicos que supervisan su estado de funcionamiento. Complementando estos indicadores, se incluyen las señales *TURBINE REPAIRED*, *ELE1 REPAIRED* y *ELE2 REPAIRED*, las cuales confirman la finalización del proceso de reparación de cada componente y permiten restablecer su operatividad.

En correspondencia con estas entradas, se definen las salidas que ejecutan acciones en respuesta a los fallos detectados, activando los protocolos de reparación de los distintos elementos del sistema. *REPAIR TURBINE* se encarga de iniciar el proceso de reparación de la turbina de gas cuando se detecta un fallo en su funcionamiento. Del mismo modo, *REPAIR ELE1* y *REPAIR ELE2* activan las reparaciones correspondientes en los motores eléctricos 1 y 2 cuando se ha identificado un problema en alguno de ellos. Estas salidas garantizan que el sistema responda de manera eficiente y automática en fallos, evitando que los motores o la turbina operen en condiciones inadecuadas que puedan averiar en fallos mayores o daños irreparables.

A diferencia de módulos anteriores, esta tabla no contempla el uso de contadores ni temporizadores, lo que indica que el sistema de gestión de fallos se basa principalmente en eventos y señales de estado en lugar de procesos dependientes de ciclos o tiempos de ejecución. No obstante, en una implementación más avanzada, se podrían incorporar para establecer tiempos mínimos de reparación o prevenir intentos de encendido prematuro tras un fallo, así como contadores que registren la cantidad de fallos detectados en cada componente para llevar un control predictivo del mantenimiento.

Entradas	Salidas	Contadores	Temporizadores
START	REPAIR TURBINE		
STOP	REPAIR ELE1		
TURBINE FAIL	REPAIR ELE2		
ELE1 FAIL			
ELE2 FAIL			
TURBINE REPAIRED			
ELE1 REPAIRED			
ELE2 REPAIRED			

Tabla 3-3 Variables del sistema de gestión de alarmas y fallos.

3.1.4.3 Red de Petri del sistema de gestión de alarmas y fallos.

En el módulo 3 del sistema de propulsión híbrida, se introduce un aspecto clave en la automatización: la integración de múltiples redes de Petri dentro de un mismo sistema, permitiendo la interconexión entre los distintos procesos para una gestión colaborativa de alarmas y fallos. A diferencia de los módulos anteriores, donde cada red de Petri operaba de manera más independiente, en esta implementación las redes están diseñadas para comunicarse entre sí, estableciendo una relación de dependencia mutua que permite coordinar la activación, mantenimiento y recuperación de los motores eléctricos y la turbina de gas.

La primera red de Petri de la Figura 3-15 representa el sistema principal de control, encargado de gestionar el funcionamiento de los motores eléctricos y la turbina de gas. En esta estructura, se han incorporado mecanismos que garantizan que estos elementos no puedan activarse hasta que los procesos de limpieza y mantenimiento preventivo, definidos en el módulo 2, se hayan completado. Para ello, se ha agregado un nuevo lugar de espera (*PA13*), el cual bloquea la activación de los motores y la turbina hasta que el lugar *PB15* de la segunda red de Petri indique que el proceso de limpieza ha finalizado. Esta interconexión entre redes permite que el sistema de propulsión no inicie su operación si las condiciones no son óptimas, evitando fallos derivados de un encendido prematuro.

Por otro lado, la segunda red de Petri de la Figura 3-16 representa el sistema de gestión de fallos y alarmas, el cual supervisa el estado de los motores eléctricos y la turbina de gas, activando procesos de reparación cuando se detectan anomalías. En esta estructura, los sensores térmicos y de vibraciones juegan un papel crucial, generando señales de fallo (*ELE1 FAIL*, *ELE2 FAIL* y *TURBINE FAIL*) que activan los procesos de reparación (*REPAIR ELE1*, *REPAIR ELE2* y *REPAIRE TURBINE*) hasta recibir una confirmación de recuperación. Sin embargo, para evitar conflictos en la ejecución de estos procesos, se han implementado condiciones de bloqueo y dependencia entre las redes.

Uno de los aspectos más relevantes en esta integración es la restricción en la recarga de la caldera de gas y de las baterías cuando un proceso de reparación está en curso. Esto se logra mediante el uso de arcos inhibidores, como el que conecta el lugar *PB7* de la segunda red con la transición *TC2* de la red de gestión de fallos. Gracias a este mecanismo, el sistema evita iniciar la reparación de la turbina si aún se está realizando la recarga de gas, asegurando que no haya interferencias entre ambos procesos. De manera similar, la recarga de las baterías de los motores eléctricos queda bloqueada cuando se está ejecutando una reparación en el respectivo motor, lo que se implementa mediante el uso de los lugares *PC5* y *PC9* en la segunda red, los cuales inhiben el acceso a la recarga en la primera red.

Además, se ha incorporado un mecanismo de seguridad adicional que impide la activación de los motores y la turbina mientras se está llevando a cabo una recarga. Esto se logra mediante la supervisión de los lugares *PB7*, *PB8* y *PB11*, asegurando que, si alguno de estos está activo, el sistema bloqueará la activación de los motores hasta que finalice el proceso. Esta medida previene fallos operacionales y garantiza que la distribución de energía dentro del sistema sea adecuada en todo momento.

Finalmente, otra interconexión fundamental entre ambas redes es la sincronización del modo de fallo automático. En caso de que se produzca un fallo en un motor o la turbina mientras se está llevando a cabo una recarga, al finalizar dicho proceso el sistema entra automáticamente en modo de gestión de fallo, asegurando que la reparación se ejecute inmediatamente sin necesidad de intervención manual. Esto refuerza la autonomía del sistema y minimiza los tiempos de inactividad de los componentes.

En conclusión, la principal innovación introducida en este módulo es la comunicación bidireccional entre redes de Petri, permitiendo una integración modular que mejora la eficiencia y seguridad del sistema de propulsión híbrida. Al establecer dependencias entre procesos, se logra una gestión optimizada del mantenimiento y la recuperación de fallos, asegurando que ningún componente opere en condiciones inadecuadas. Gracias a este enfoque, el sistema no solo es capaz de detectar y responder a fallos de manera automática, sino que también garantiza una coordinación eficiente entre los distintos módulos, consolidando un diseño robusto y adaptable a distintos escenarios operativos.



Figura 3-15 Red de Petri del sistema de gestión de alarmas y fallos (1).

3.1.4.4 Programación y visualización

En la Figura 3-17, se puede observar una parte del código del módulo 3 implementado en *Codesys*, en este caso se trata del bloque de transiciones, donde en la línea 19 se encuentra la transición *TA19*, la cual representa una interconexión entre redes de Petri. Este tipo de implementación es una de las principales innovaciones, y en este caso la transición *TA19* está condicionada por los lugares *PLACES_M1.PA13* y *PLACES_M2.PB15*, lo que indica que para que esta transición se active es necesario que ambas condiciones se cumplan simultáneamente. *PA13* pertenece a la red del módulo 1, mientras que *PB15* se encuentra en la red del módulo 2, lo que confirma la comunicación directa entre módulos. Esta conexión tiene como objetivo garantizar que el sistema de propulsión no pueda iniciar su operación hasta que el proceso de limpieza del módulo 2 haya finalizado correctamente.

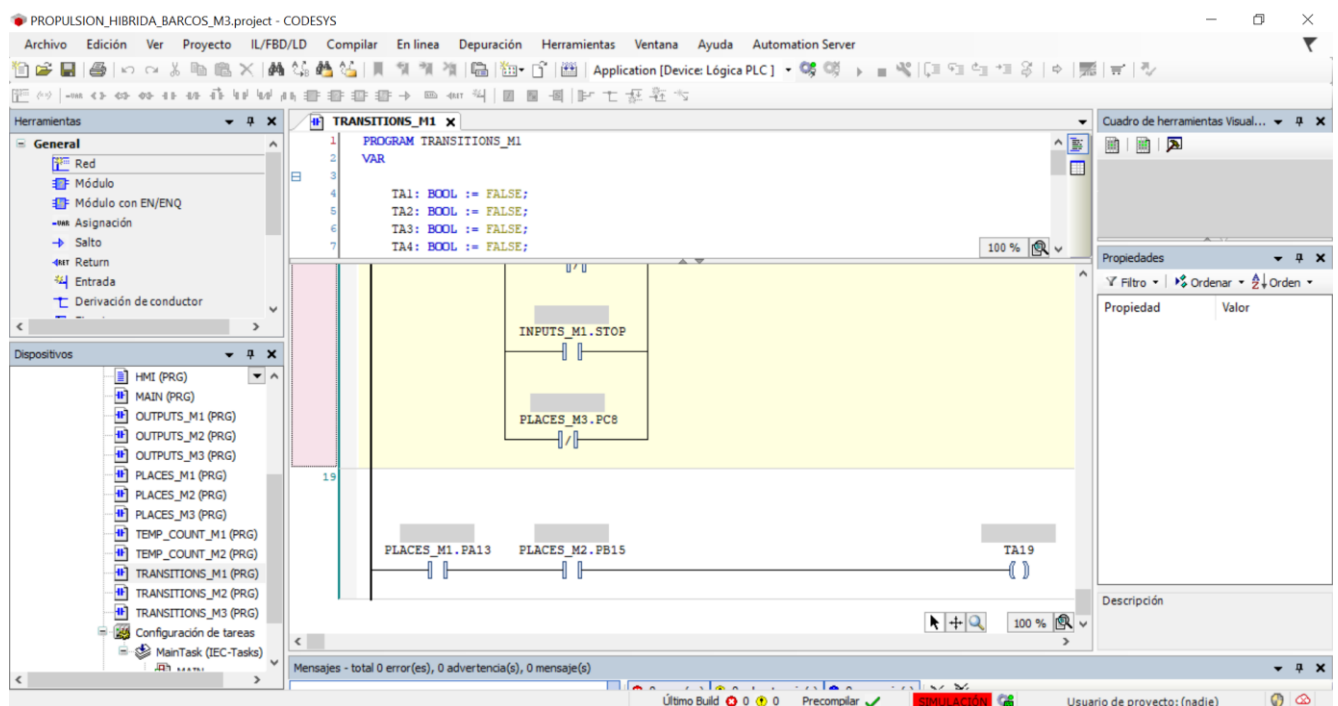


Figura 3-17 Programación Codesys del sistema de gestión de alarmas y fallos, bloque TRANSITIONS_M1.

Esta imagen representa un claro ejemplo de la interconexión entre redes de Petri en el módulo 3, asegurando que los procesos de mantenimiento y reparación se sincronicen adecuadamente con la operación del sistema de propulsión híbrida. Gracias a esta implementación, se logra una gestión más robusta y segura, evitando la activación de motores y turbinas en condiciones inadecuadas y garantizando que todos los procesos previos necesarios hayan sido completados antes de la puesta en marcha.

En la Figura 3-18 se observa un fragmento de la programación donde se puede diferenciar dos transiciones clave, *TC17* y *TC18*, cada una condicionada por diferentes lugares dentro de la red, lo que permite coordinar la activación de procesos en función del estado del sistema y de los eventos de fallo detectados.

En primer lugar, la transición *TC17* se encuentra condicionada por cuatro elementos: *PLACES_M3.PC8*, *PLACES_M2.PB13*, *PLACES_M3.PC12* y la señal de entrada *INPUTS_M3.ELE2_FAIL*. Este conjunto de condiciones indica que la transición solo podrá activarse si todas estas señales están en el estado adecuado. *PC8* y *PC12* pertenecen a la red del módulo 3, lo que sugiere que están asociados al control de fallos y a la coordinación entre módulos. Por su parte, *PB13* pertenece a la red del módulo 2, lo que significa que esta transición también depende de la gestión de los recursos compartidos y de los procesos en curso en la red de mantenimiento. Finalmente, la señal

ELE2_FAIL representa un fallo detectado en el motor eléctrico 2, lo que implica que el sistema ha identificado un mal funcionamiento y debe inicial el protocolo de reparación.

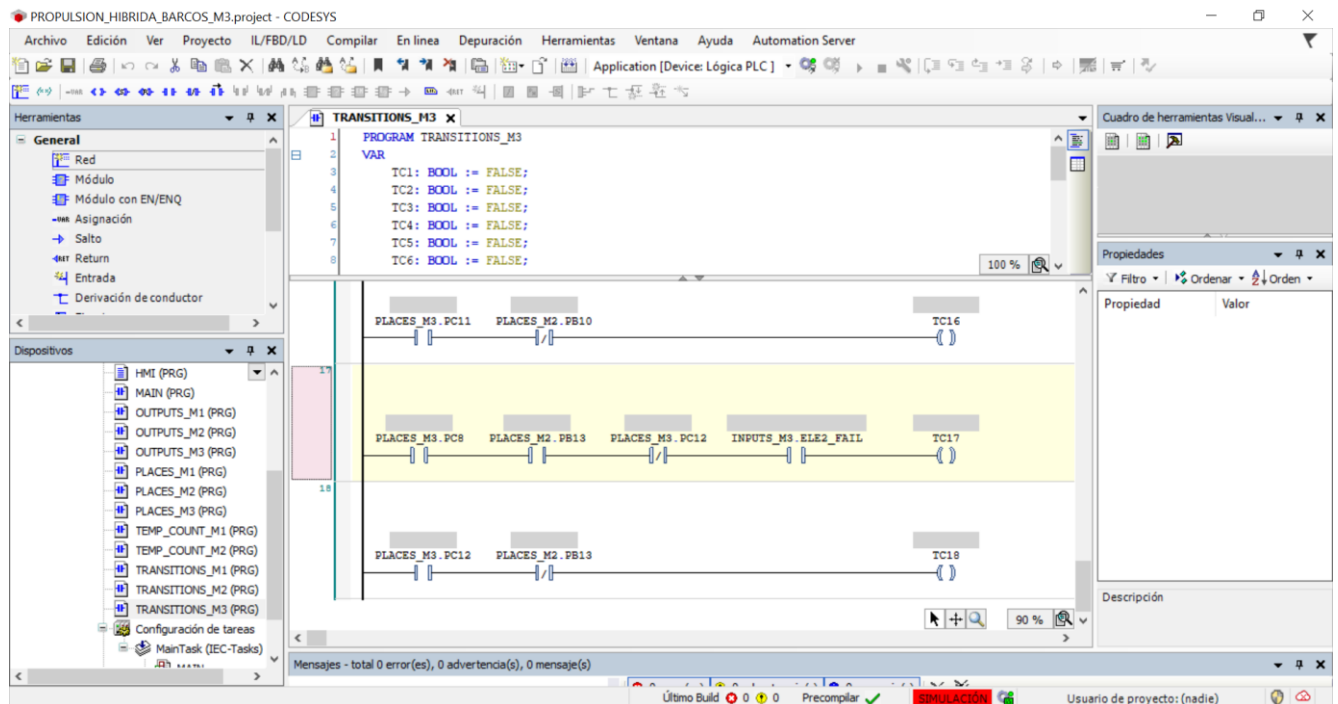


Figura 3-18 Programación Codesys del sistema de gestión de alarmas y fallos, bloque TRANSITIONS_M3.

El hecho de que *PC12* esté incluido dentro de las condiciones sugiere que este lugar actúa como una validación dentro del módulo 3, asegurando que el sistema de gestión de fallos se encuentra en un estado que permite la ejecución de la reparación. Asimismo, la presencia de *PB13* indica que el proceso de recarga de la batería del motor eléctrico 2 no debe estar en curso, ya que en este módulo se ha establecido que la recarga y la reparación no pueden ocurrir simultáneamente. De este modo, si *PB13* está activo, la transición *TC17* no podrá ejecutarse, lo que impide que la reparación del motor eléctrico 2 comience hasta que la batería haya completado su ciclo de carga.

Por otro lado, la segunda línea del código muestra la transición *TC18*, la cual está condicionada por *PLACES_M3.PC12* y *PLACES_M2.PB13*. En este caso, la lógica indica que la activación de *TC18* requiere que *PC12* esté en un estado que permita continuar con la operación, pero también depende de *PB13*, lo que refuerza la restricción de que la recarga y la reparación no deben ocurrir simultáneamente. Esta transición está asociada con la finalización del bloqueo y la habilitación de la recarga.

En la Figura 3-19, se representa el sistema de gestión de alarmas y fallos en un estado donde la turbina de gas ha presentado un fallo, sin embargo, el sistema aún no ha iniciado su proceso de reparación debido a que la recarga de gas en la turbina sigue en curso. Esto se evidencia en la activación del lugar *PB7*, el cual representa el proceso de recarga de la caldera de gas, así como en la presencia activa de la señal *TURBINE FAIL*, indicando que la turbina ha sido detectada en estado de fallo. La interconexión entre las redes de Petri en este módulo impide que se inicie el proceso de reparación mientras la recarga de gas no haya finalizado, lo que se logra mediante un arco inhibidor que bloquea la transición correspondiente. De esta manera, el sistema prioriza la recarga de combustible antes de proceder con la reparación, asegurando que una vez completado el proceso, la turbina cuente con los recursos necesarios para volver a operar correctamente.

Adicionalmente, el lugar *PC3* se encuentra inactivo, lo que confirma que la reparación aún no ha comenzado.

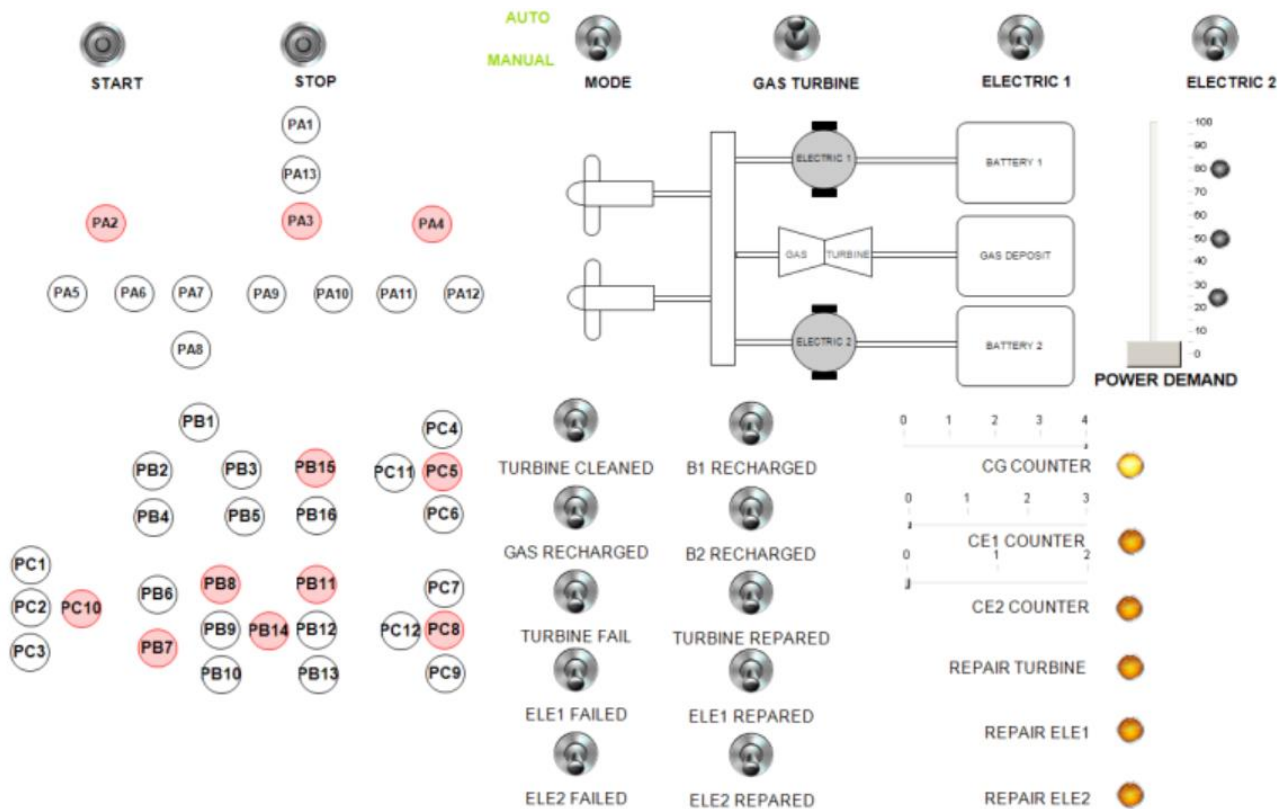


Figura 3-19 Representación visual del sistema de gestión de alarmas y fallos, en espera de la finalización de la recarga de gas en la turbina para comenzar con la reparación de esta.

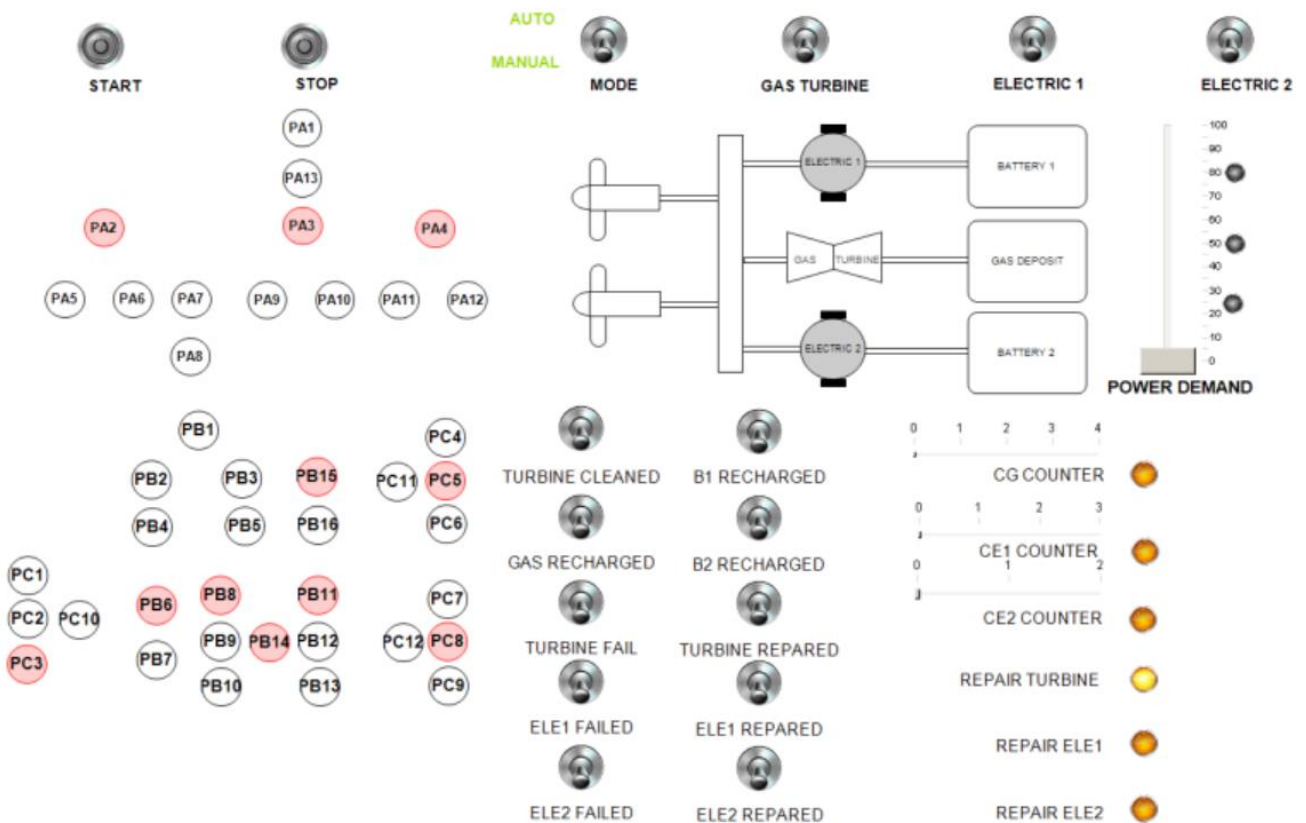


Figura 3-20 Representación visual del sistema de gestión de alarmas y fallos, reparando la turbina.

En la Figura 3-20 el sistema ha cambiado de estado y ahora la reparación de la turbina ha sido iniciada. Esto se observa en la activación del lugar *PC8*, lo que indica que la transición correspondiente se ha habilitado tras la finalización de la recarga de gas. A diferencia de la Figura 3-19, el lugar *PB7* ya no está activo, lo que significa que la caldera de gas ha completado su proceso de llenado, permitiendo así que la red de gestión de fallos proceda con la reparación de la turbina. Además, la señal *TURBINE REPAIRED* aún no está activa, lo que confirma que el proceso de reparación sigue en curso y no ha sido finalizado. La correcta secuenciación de estos eventos demuestra la funcionalidad del sistema de automatización basado en redes de Petri, donde las dependencias entre módulos aseguran que cada proceso ocurra en el orden adecuado, evitando interferencias o fallos en la operatividad del sistema de propulsión híbrida. La implementación de estas restricciones garantiza que la turbina no vuelva a ponerse en funcionamiento hasta que la reparación haya sido completada exitosamente, asegurando así un sistema más seguro y eficiente.

Además, se han incluido nuevos interruptores y lámparas con respecto a los módulos anteriores, los cuales han sido añadidos para la gestión de alarmas y fallos del sistema. Estos nuevos elementos permiten una mayor supervisión y control del estado del sistema.

En cuanto a los nuevos interruptores, se han agregado seis elementos todos situados en la parte inferior derecha de la interfaz seguidos unos de otros. *TURBINE FAIL*, *ELE1 FAIL* y *ELE2 FAIL*, que permite visualizar el fallo de la turbina o motores y su activación indica que se ha producido un fallo y debe procederse con su reparación, o en caso de los motores reacción ante anomalías térmicas iniciando los procesos de reparación también. Además, *TURBINE REPAIRED*, *ELE1 REPAIRED* y *ELE2 REPAIRED*, indicando la finalización del proceso de reparación ante un fallo de cualquiera de los tres elementos.

También se ha añadido nuevas lámparas indicadoras en el sistema para proporcionar información visual sobre el estado de las reparaciones en curso, son: *REPAIR ELE1*, *REPAIR ELE2* y *REPAIR TURBINE* a la derecha de los nuevos interruptores descritos anteriormente. Estas lámparas se activan cuando se ha iniciado el proceso de reparación y está en curso.

Estas nuevas implementaciones pueden mejorar la visualización del estado del sistema y facilitan la interacción con el operador, garantizando que cualquier fallo detectado sea gestionado correctamente y que las tareas de reparación sean llevadas a cabo sin interferencias con otros procesos en curso. La inclusión de estas nuevas señales visuales y de control refuerza la seguridad y la confiabilidad del sistema, permitiendo que cada componente de la propulsión híbrida funcione bajo condiciones óptimas y con una supervisión adecuada.

Existen más estados posibles del sistema; estos son solo algunos ejemplos. Se ha realizado un video mostrando el funcionamiento del sistema, se puede acceder usando el siguiente enlace:

<https://drive.google.com/file/d/1S-hv4P3awvO4ybXp4nAOMF95ULUzQgXr/view?usp=sharing>

3.2 Proyecto 2: Automatización logística y energética de un muelle de carga.

3.2.1 Descripción del sistema de carga automatizada

El proyecto contempla la implementación de un muelle de carga diseñado para la operativa de dos lanchas cada una con un sistema de carga diferenciado. Una de las lanchas está equipada con un sistema de carga interna automatizado, mientras que la segunda requiere una carga manual.

Para optimizar la lógica, se integra una plataforma robótica autónoma que transporta las cargas desde un almacén central hasta la posición de carga de una de las embarcaciones. Este sistema garantiza una transferencia eficiente de materiales, reduciendo los tiempos de espera y mejorando la operatividad general del muelle. Además, el diseño incluye un sistema de recarga de baterías tanto para la plataforma robótica que transporta la carga como para el sistema automatizado de la lancha con carga interna. Este sistema de recarga asegura la operatividad continua de ambos dispositivos, minimizando interrupciones y optimizando la eficiencia energética del proceso. Con esta infraestructura, se logra un sistema coordinado y automatizado que mejora la gestión de carga y distribución en el muelle, adaptándose a distintos métodos de carga según la embarcación.

3.2.2 Módulo 1: Lancha con plataforma de carga automática

El objetivo es el modelado y programación de una plataforma de carga automatizada con el objetivo de disponer de sistemas rápidos de carga que permitan reducir los tiempos de respuesta frente a una emergencia. La representación del tipo de sistema a modelar es una lancha donde se dispone de 4 zonas de almacenamiento (*P1*, *P2*, *P3* y *P4*) y una zona de recogida de carga en la popa, en donde puede verse la plataforma, en color verde, encargada de recoger la carga y depositarla de forma ordenada en las zonas de almacenamiento. En la proa se dispone de un conjunto de luces que indican el estado del funcionamiento de la plataforma, y se dispone de un panel de control que permite encender y apagar el sistema, y visualizar si se encuentra la plataforma en operación y el número de elementos de almacenamiento que ya han sido cargados.

3.2.2.1 Enunciado

Para realizar la automatización del sistema de propulsión descrito se deben considerar los siguientes aspectos:

- La plataforma se enciende al recibir la señal **START**.
- La plataforma dispone de 3 sensores de posición, que generan las señales **SP0** (para la posición de recogida de la carga), **SP1** (para las posiciones de descarga *P1* y *P2*) y **SP2** (para las posiciones de descarga *P3* y *P4*).
- Se debe emplear un contador **CL** para codificar la posición actual en la que debe realizarse cada nueva carga disponible.
- Mientras se mantenga encendida debe mantenerse encendida la luz verde activando la instrucción **GREEN**.
- En el momento de encenderse el sistema, el primer paso consiste en abrir la tapa de la plataforma que debe comenzar a abrirse ejecutando la instrucción **OPEN** hasta recibir la señal **OPENED** generada por un sensor que indica que la plataforma está completamente disponible. En ese momento queda a la espera (recordando, manteniendo encendida la luz verde) de dos tipos de instrucciones:
 - Si se recibe la señal **STOP**, la plataforma se debe apagar, pero antes debe ser cerrada ejecutando la instrucción **CLOSE** (manteniendo encendida la luz verde) hasta que la señal **OPENED** se encuentre negada.

- Si se recibe la señal **LOAD PA** se inicia el proceso de carga en la plataforma que se describe a continuación:
 - En primer lugar, se carga la plataforma mediante la instrucción **CHARGE**, que dura **0.5 segundos** y durante la cual deben mantenerse encendidas las luces verde y azul (esta última se activa con la instrucción **BLUE**).
 - A continuación, la plataforma debe avanzar hasta alcanzar la posición de descarga correcta de las cuatro disponibles, teniendo en cuenta que la carga debe realizarse en el orden correcto (*P1*, *P2*, *P3* y *P4*) para evitar oscilaciones peligrosas en la lancha. Para lograr esto se debe activar la instrucción **FORWARD** y activar la luz roja empleando la instrucción **RED**.
 - Una vez que alcanza la posición adecuada para realizar la descarga, se deben ejecutar los dos siguientes procesos simultáneos:
 - Para realizar la descarga hay que emplear la orden **UNCHARGE**, la cual debe activarse durante 0.5 segundos, manteniendo activada la luz azul durante el proceso de descarga.
 - Para evitar problemas de estabilización se debe ejecutar la orden **STABILIZE**, que se encarga de corregir la descarga para evitar oscilaciones en la lancha, la cual finaliza al recibir la señal **IS STABLE** indicando que las oscilaciones han sido eliminadas.
 - A continuación, la plataforma debe retroceder hasta alcanzar la posición de carga activando la instrucción **REVERSE** y las luces roja y verde.
 - Al regresar a la zona de recogida de carga la plataforma debe apagarse automáticamente si se ha finalizado la carga de las 4 posiciones, o de lo contrario quedar a la espera que una nueva carga que llevar a su posición dentro de la lancha.
 -

3.2.2.2 Tabla de variables

En este apartado se presentan las variables utilizadas en el modelo de la lancha con plataforma de carga automática, en la Tabla 3-4 . La definición de estas variables es clave para estructurar el funcionamiento del sistema automatizado, ya que permite controlar los estados y transiciones dentro de la red de Petri. Se incluyen tanto las variables de entrada, que reflejan la detección de eventos externos, como las variables de salida, que activan los actuadores responsables del movimiento y posicionamiento de la carga. Además, se consideran las variables de control internas que facilitan la sincronización de los procesos y garantizan el correcto flujo de operaciones dentro del sistema.

Entradas	Salidas	Contadores	Temporizadores
START	BLUE	CL	T1
STOP	RED		
OPENED	GREEN		
LOAD_PA	CLOSE		
IS_STABLE	OPEN		
SP0	UNCHARGE		
SP1	CHARGE		
SP2	FORWARD		
	REVERSE		
	STABILIZE		

Tabla 3-4 Variables de lancha con plataforma de carga automática.

3.2.2.3 Red de Petri de lancha con carga automática

En este apartado se describe la representación mediante una red de Petri del funcionamiento de la lancha con plataforma de carga automática, representada en la Figura 3-21. Cabe destacar que se ha implementado una estructura en la que aparecen procesos paralelos, lo que significa que varios eventos, en concreto dos eventos (*UNCHARGE* y *STABILIZE*), pueden desarrollarse simultáneamente dentro del sistema, pero, sin embargo, se ha definido de tal manera que hasta que ambos no hayan terminado el sistema no podrá continuar con su siguiente etapa (*REVERSE*).

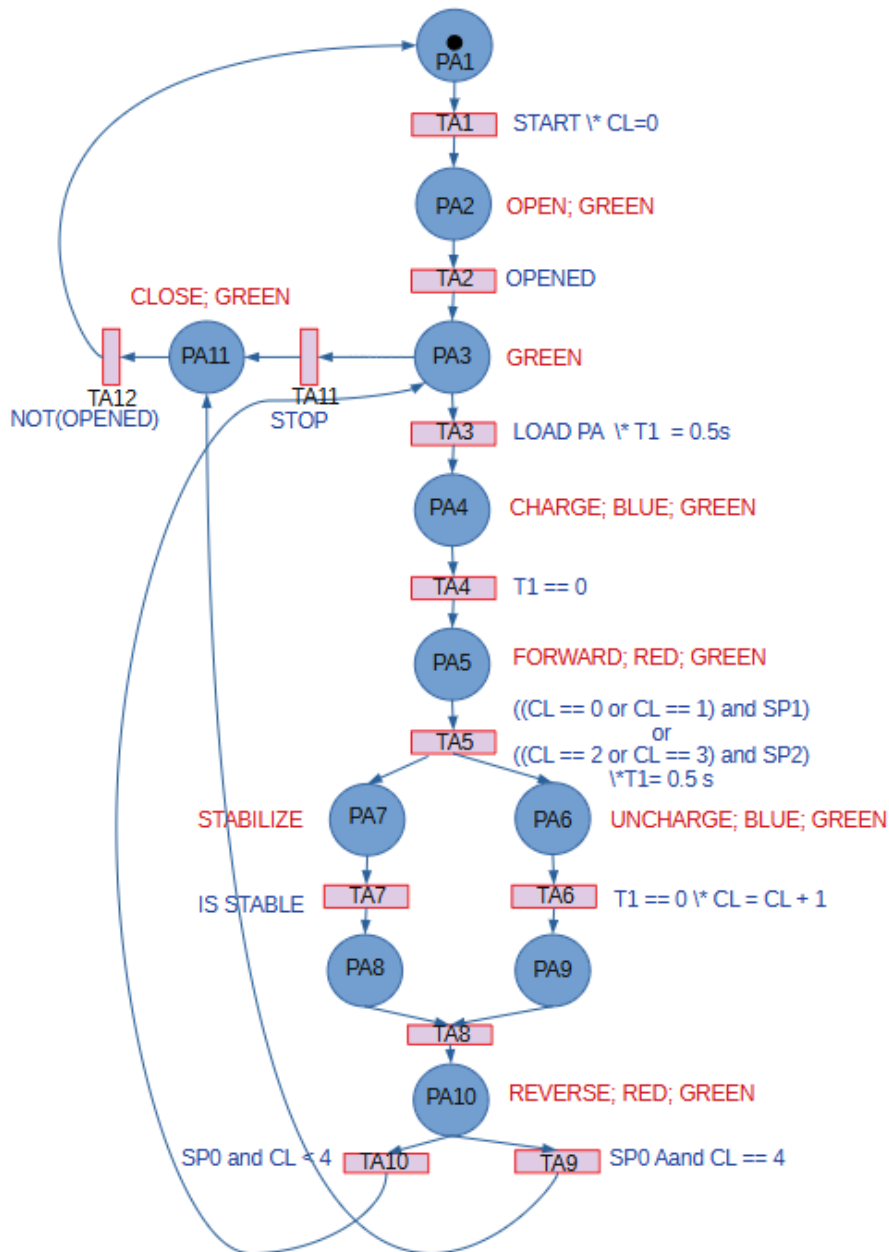


Figura 3-21 Red de Petri de lancha con plataforma de carga automática.

3.2.2.4 Programación y visualización

La programación de este módulo se ha realizado siguiendo la misma metodología aplicada en los módulos anteriores, utilizando el lenguaje LD para representar la lógica de funcionamiento de la red de Petri. Se han definido las condiciones de transición mediante comparaciones entre entradas y valores de variables de control.

Sin embargo, en este caso se quiere destacar la programación de *TA5*, ya que su implementación tiene particularidades importantes. En esta transición se ha establecido una lógica condicional que evalúa el valor de *CL_VALUE* y el estado de los sensores de posición *SP1* y *SP2*, determinando así el momento en el que la transición puede activarse.

En la Figura 3-22 y Figura 3-23 se especifica la dependencia de las variables *C1*, *C2*, *C3* y *C4*.

Se realiza una evaluación de los sensores *SP1* y *SP2*

- Se verifica la activación de *INPUT3_M1.SP1* o *INPUT3_M1.SP2* en función del valor del contador *TEMP_COUNT_M1.CL_VALUE*.
- Se han definido condiciones en las que si el contador *CL_VALUE* toma los valores 0 o 1, la transición será activada cuando *SP1* esté activo.
- Si el contador *CL_VALUE* toma los valores 2 o 3, la transición se activará cuando *SP2* esté activo.

También se hace uso de un temporizador para establecer el retardo de 0,5 segundos.

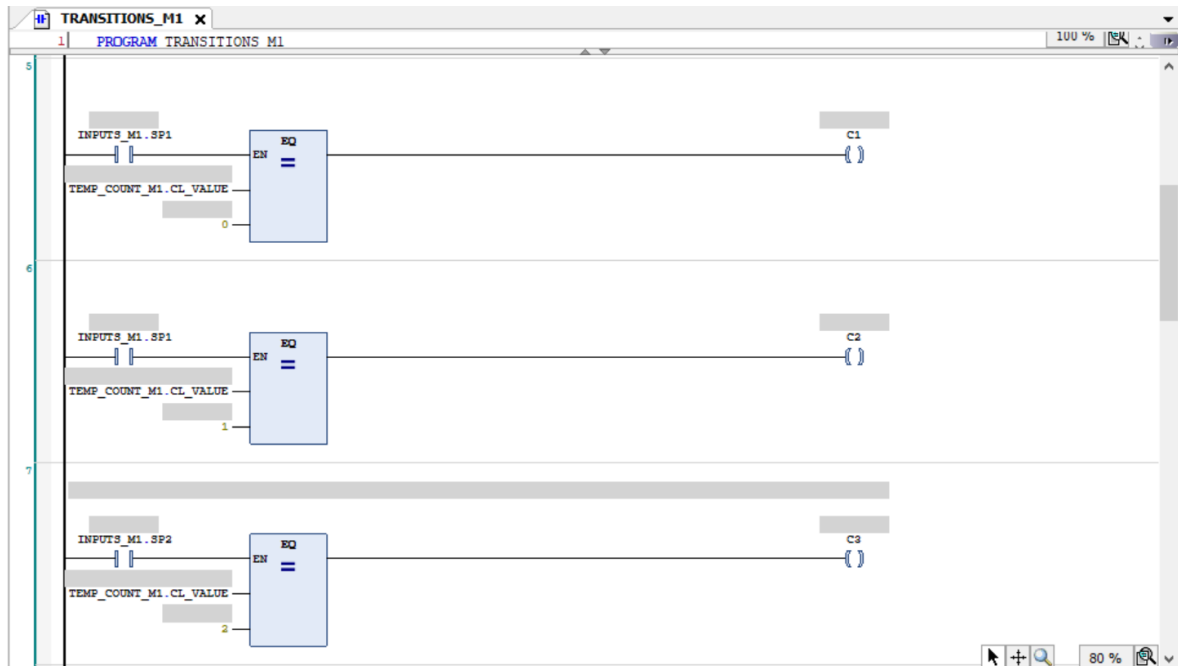


Figura 3-22 Programación Codesys del sistema del muelle de carga automatizada, bloque TRANSITIONS_M2 (1).

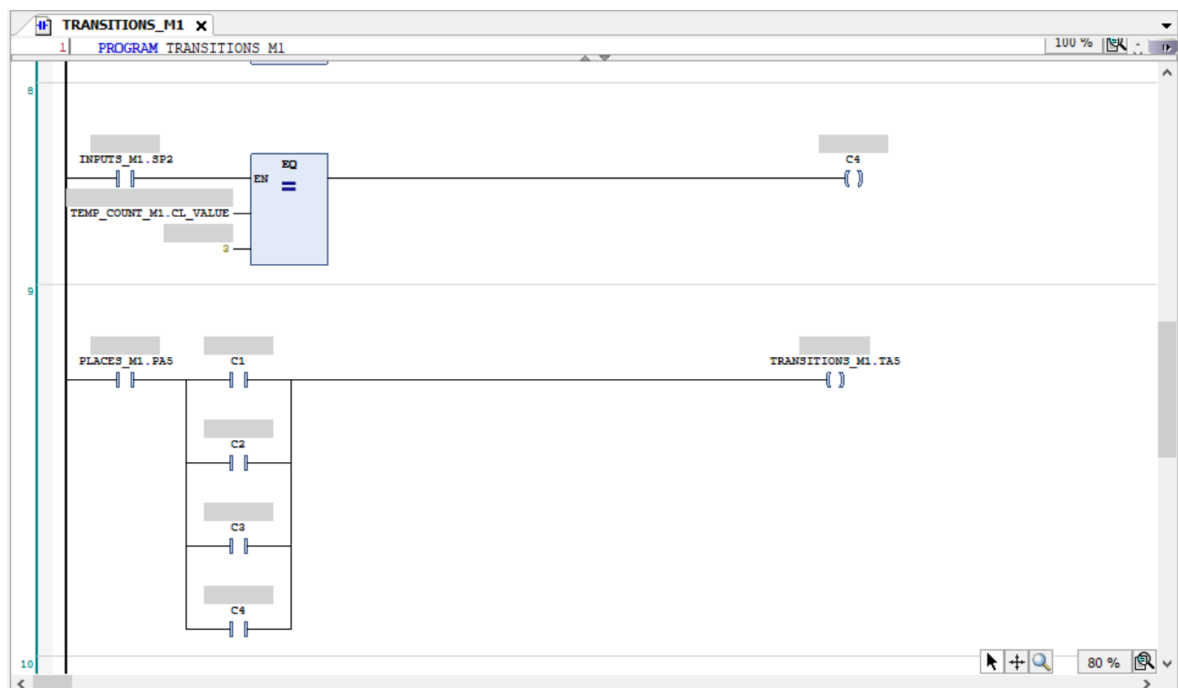


Figura 3-23 Programación Codesys del sistema del muelle de carga, bloque TRANSITIONS_M2 (2).

En la Figura 3-23 se observa como la activación de *TA5* depende de la activación del lugar *PA5* y de algunas de las variables (*C1*, *C2*, *C3* y *C4*) que se han determinado anteriormente.

En este apartado además se presenta la interfaz de programación utilizada para modelar y simular el comportamiento del sistema automatizado. En la pantalla se pueden observar los distintos elementos que componen la interfaz de usuario, donde se incluyen los controles principales situados en la parte superior, los indicadores de estado en la zona central representados con lámparas de luz, y la representación de la red de Petri en la zona izquierda. También se puede observar la embarcación con las distintas zonas de descarga debajo de los controles principales en la zona derecha de la interfaz.

Cada imagen representa una fase específica del proceso de carga y descarga, permitiendo visualizar de una manera clara el flujo de trabajo del sistema. Se identifican los momentos en los que la lancha está en espera, en movimiento, realizando la carga o finalizando el proceso. Esta representación gráfica facilita la comprensión del funcionamiento del modelo y ayuda a detectar posibles mejoras en la operativa del sistema automatizado.

En la Figura 3-24 se representa el sistema en su estado inicial, es decir, apagado. Se puede verificar esta condición debido a la ausencia de indicadores luminosos activados en la interfaz.

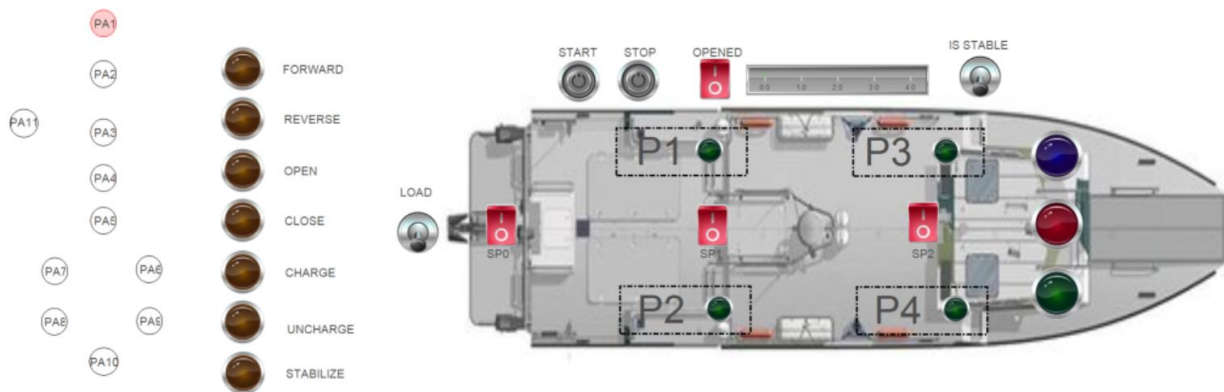


Figura 3-24 Representación visual del sistema del muelle de carga automatizada desactivado

En primer lugar, ninguna de las luces situadas junto a la red de Petri está encendida, lo que confirma que el sistema no está ejecutando ninguna acción en ese momento. Asimismo, los puntos de descarga de la lancha (*P1*, *P2*, *P3* y *P4*) no presentan iluminación, lo que indica que no se ha realizado ninguna operación de descarga en dichas áreas.

Por otro lado, los sensores se encuentran desactivados lo que sugiere que la plataforma automática no está posicionada en ninguno de los tres puntos clave del sistema (*SP0*, *SP1* y *SP2*). Finalmente, se puede concluir que el sistema permanece apagado, ya que la luz verde de la proa tampoco se encuentra encendida, lo que demuestra la inactividad del sistema en esta fase.

En la Figura 3-25 se observa el sistema en una fase de operación activa. A diferencia del estado inicial, en esta imagen se pueden identificar varias señales que indican que la lancha y su plataforma de carga automatizada están ejecutando ciertos procesos.

En primer lugar, se aprecia que las luces correspondientes a los puntos de descarga *P1*, *P2* y *P4* están encendidas en color verde, lo que indica que estos puntos están activos y han recibido carga. Sin embargo, *P3* no muestra iluminación, ya que no se ha recibido carga en esa área.

En cuanto a los sensores en la plataforma, se observa el sensor *SP3* activado, se puede apreciar que la plataforma de descarga se encuentra en esa zona y acaba de descargar en *P3*, pero está esperando a recibir la señal de estabilidad de la embarcación para poder regresar a *SP0* a recoger más carga, por ello

la lámpara con la salida *STABILIZE* se encuentra encendida, para asegurar la estabilidad antes de continuar con la siguiente fase de operación (*REVERSE*).

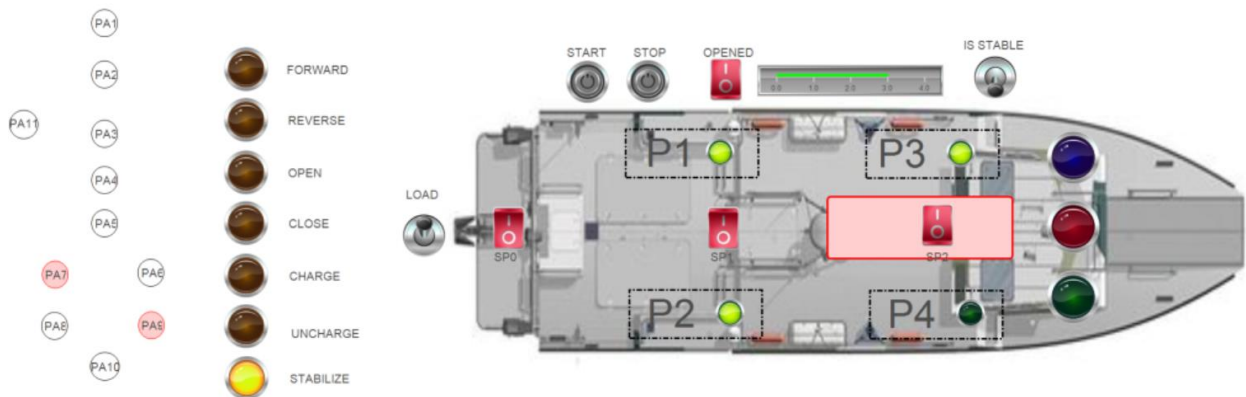


Figura 3-25 Representación visual del sistema del muelle de carga automatizada, realizando descarga en P3

Estas dos imágenes representan dos estados del sistema, sin embargo, existen muchos más estados en los que se puede encontrar y se pueden observar en la interfaz visual. Por este motivo, se ha realizado un video mostrando el funcionamiento del sistema, se puede acceder usando el siguiente enlace:

https://drive.google.com/file/d/13INCBIiLREObHbAYMAyAu1isyRbPi_PuB/view?usp=sharing

3.2.3 Módulo 2: Gestión de la carga y almacenamiento en el muelle automatizado

El objetivo de este módulo es modelar y programar la gestión de la carga y almacenamiento en el muelle automatizado. En este sistema, la carga se distribuye entre dos zonas de almacenamiento denominadas A y B (las dos embarcaciones). La plataforma de carga automática debe clasificar y transportar la carga de manera eficiente, asegurando que se almacene correctamente en una de las dos áreas, dependiendo de su tipo y disponibilidad.

En el sistema a modelar, la carga es procesada en diferentes etapas hasta su almacenamiento final. El sistema inicia con la activación del proceso de carga y permite la distribución en función de las condiciones de almacenamiento disponibles.

3.2.3.1 Enunciado

Para realizar la automatización del sistema de carga y almacenamiento se deben considerar los siguientes aspectos:

- El sistema se enciende al recibir la señal **START**.
- Si se recibe la señal **STOP**, la plataforma se debe apagar, pero es necesario que se encuentre en la posición del almacén en espera de recibir una petición de carga de una de las dos lanchas.
- Debe modificarse la red de Petri del módulo 1 para que, al finalizar la carga de una lancha automática, comience automáticamente la carga de otra lancha automática nueva.
- Existen dos zonas para realizar la carga de las lanchas: la zona A (carga automatizada) y la zona B (carga manual) y una zona del almacén general (S)
- Inicialmente, se asume que las áreas de almacenamiento A y B están vacías de carga y que la plataforma de carga está en la posición del almacén general (S).

- Se deben emplear los contadores **CA**, **CB** y **CS** para registrar la cantidad de cargas almacenadas en cada una de las áreas (A y B), y la cantidad total de cargas extraídas del almacén (CS).
- Si la carga ha sido procesada y depositada en su área de almacenamiento correspondiente, se actualizan los contadores y la plataforma del muelle, una vez que ha regresado al almacén central, queda en espera de una nueva petición de carga en una de las lanchas.
- Si estando en espera de recibir una petición de carga, se recibe la señal **STOP**, el sistema debe detenerse, es decir, estando seguro de que todas las operaciones en curso sean finalizadas adecuadamente.

Secuencia de operación

- Solicitud de carga:
 - La lancha automatizada situada en la zona A debe estar operativa con la plataforma abierta para indicar que se encuentra disponible para recibir una carga. Por otra parte, la plataforma de carga de la lancha automática ya no recibe la señal empleada en el módulo 1 denominada **LOAD PA**, de forma que el paso a realizar la carga depende de que la plataforma del almacén se encuentre en modo de descarga en la posición A.
 - La lancha automatizada situada en la zona B activa la señal **LOAD PB** para indicar que se encuentra disponible para recibir una carga.
 - La plataforma de transporte del muelle es compartida por ambas zonas, y en caso de conflicto la plataforma manual situada en B tiene preferencia de acceso a la plataforma A.
- Transporte de la carga:
 - Si la carga debe ir a la zona A, se debe ejecutar la instrucción **PLOAD TO A** para que la plataforma de transporte del muelle se desplace a esa zona.
 - Si la carga debe ir a la zona B, se debe ejecutar la instrucción **PLOAD TO B** para que la plataforma de transporte del muelle se desplace a esa zona.
- Descarga en la zona de almacenamiento:
 - Una vez que la plataforma de transporte llega a la zona A o B dos sensores generan las señales **PLOAD IN A** o **PLOAD IN B** para indicar la posición de la plataforma de transporte en la zona correspondiente.
 - Se debe considerar que el proceso de transferencia de la carga se ejecuta mediante la instrucción **PLOAD UNCHARGE**, el cual, independientemente de la zona A o B, dura como mínimo 0,5 segundos. Además, en el caso de la zona A la descarga solo puede finalizar cuando la plancha automatizada haya finalizado el proceso de carga.
- Actualización contadores:
 - Una vez que la carga ha sido correctamente almacenada, se incrementan los contadores **CA** o **CB** dependiendo de la zona de almacenamiento utilizada.
 - Una vez que la plataforma de transporte regresa al almacén, se debe incrementar el contador total de cargas transportadas **CS**.
- Regrese a la posición inicial:
 - Una vez completado el proceso de carga y almacenamiento, el sistema retorna a su estado inicial, para lo cual se debe ejecutar la instrucción **PLOAD TO STORAGE** hasta llegar al punto (S) de almacenamiento recibiendo la señal **PLOAD IN STORAGE** quedando a la espera de una nueva solicitud de carga por parte de una de las dos lanchas.

3.2.3.2 Tabla de variables

En este apartado se presentan las variables utilizadas en la modelización del sistema de gestión de carga y almacenamiento en el muelle automatizado. Estas variables incluyen las entradas y salidas del sistema los temporizadores utilizados en la sincronización de las transiciones y los contadores que permiten el control de los eventos del sistema. Esta tabla de variables define los elementos clave que intervienen en el proceso de carga de las lanchas, diferenciando entre la lancha con carga automatizada y la lancha con carga manual. También se incluyen los eventos asociados al movimiento de la plataforma robótica, la disponibilidad de carga en el almacén y la activación de los procesos de carga y descarga.

Estas variables permiten modelar el funcionamiento del sistema de carga y almacenamiento, asegurando que los procesos de transporte, carga y sustitución de baterías se realicen de manera eficiente y sin interrupciones.

Entradas	Salidas	Contadores	Temporizadores
LOAD_PB	PLOAD_TO_A	CS	T2
PLOAD_IN_A	PLOAD_TO_B	CA	
PLOAD_IN_B	PLOAD_UNCHARGE	CB	
PLOAD_IN_STORAGE	PLOAD_TO_STORAGE		

Tabla 3-5 Variables de la gestión de carga y almacenamiento en el muelle.

3.2.3.3 Red de Petri de la gestión de carga y almacenamiento en el muelle automatizado

A continuación, se presenta la red de Petri que modela la gestión de carga y almacenamiento en el muelle automatizado. En esta red se presentan las transiciones entre los distintos estados del sistema, los eventos que activan dichas transiciones y los recursos compartidos entre las dos lanchas.

La red de Petri de la Figura 3-27 modela el funcionamiento de un muelle de carga automatizado, donde la mercancía es distribuida entre dos destinos principales (A y B) o enviada al almacenamiento central. El sistema inicia en *PB1*, activado por *TB1* (*START*), y permite detenerse en cualquier momento con *TB12* (*STOP*). La carga ingresa al muelle por dos rutas paralelas, representadas por *PB3* y *PB8*, desde donde se decide su destino. Si la carga se dirige a A, pasa por *PB4*, luego a *PB5* (*PLOAD TO A*) y se descarga en *PB6*, incrementando el contador *CA*. Si va a B, sigue un proceso similar mediante *PB9*, *PB10* y *PB11*, aumentando *CB*. En caso de que la carga no tenga un destino inmediato, se almacena en *PB7* o *PB12*, aumentando el contador *CS*. El modelo asegura una distribución ordenada mediante tiempos de espera y contadores que permiten registrar la cantidad de carga gestionada en cada destino.

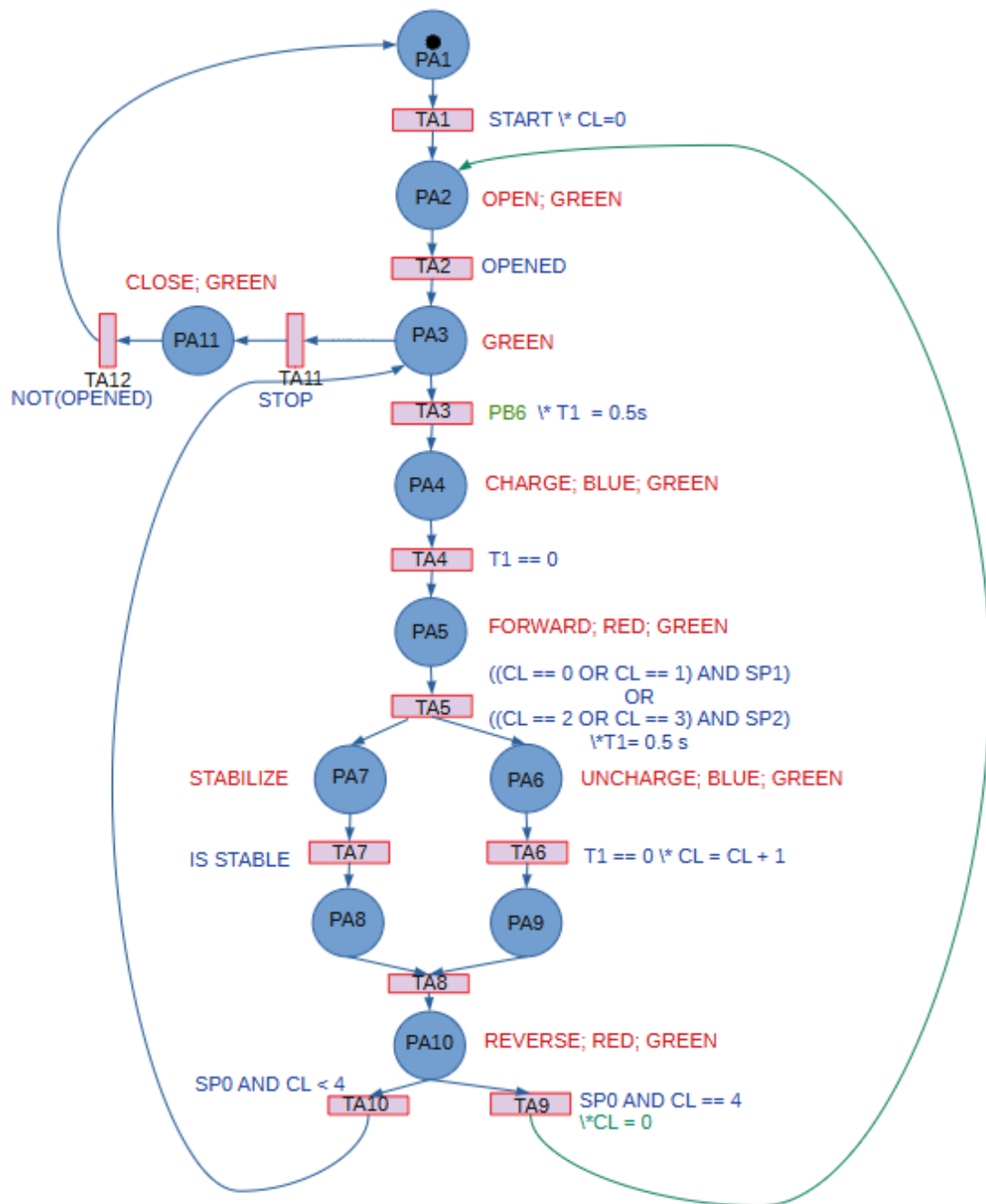


Figura 3-26 Red de Petri de la gestión de carga y almacenamiento en el muelle (1).

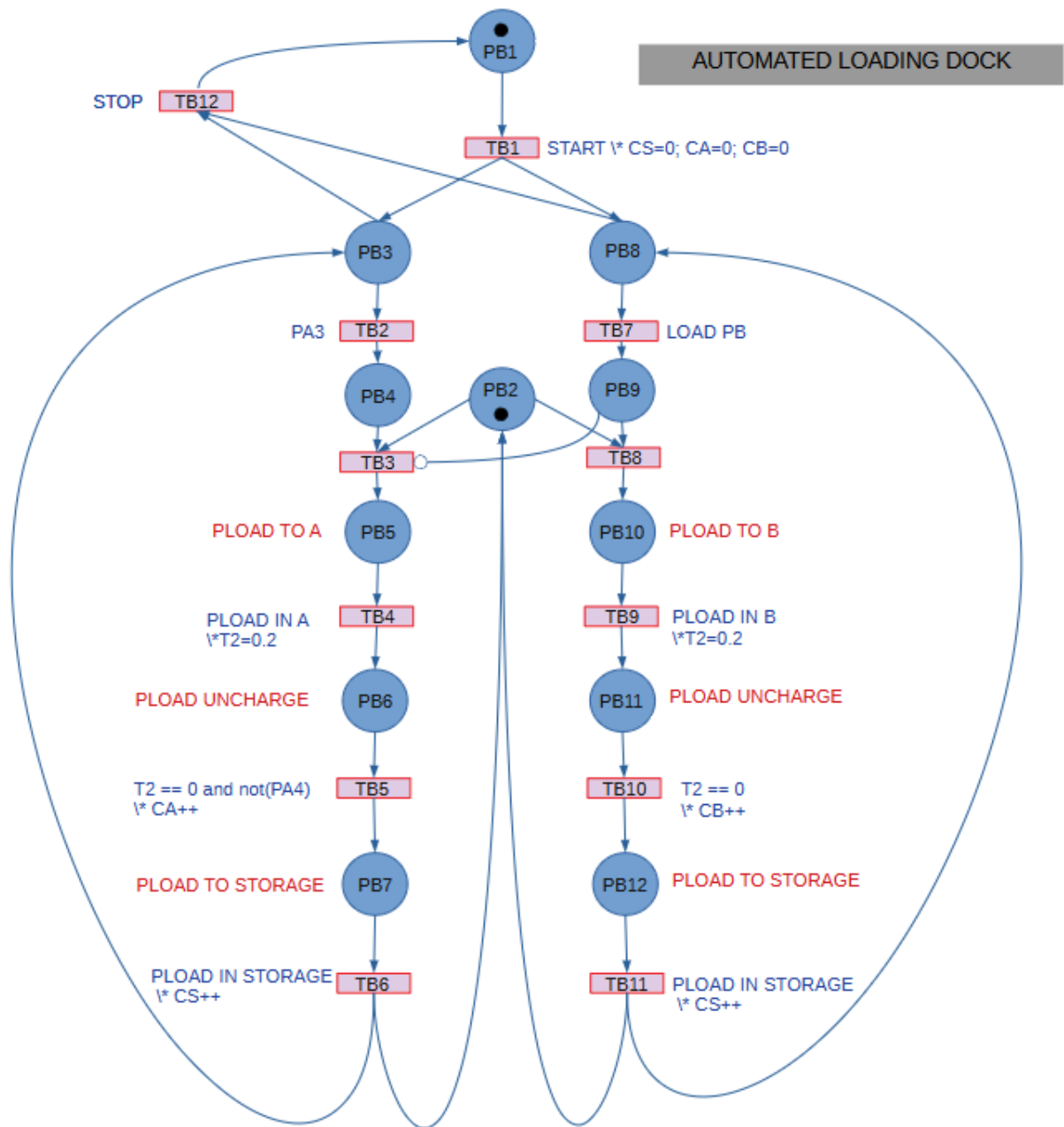


Figura 3-27 Red de Petri de la gestión de carga y almacenamiento en el muelle (2).

3.2.3.4 Programación y visualización

La programación de este módulo se ha realizado siguiendo la misma metodología aplicada en los módulos anteriores, utilizando el lenguaje LD para representar la lógica de funcionamiento de la red de Petri.

La Figura 3-28 muestra el bloque *PLACES_M2*, destacando la implementación del recurso compartido. Este recurso se gestiona mediante lugares específicos como *PB2* que controlan la prioridad de la carga para evitar conflictos en el uso del muelle. De esta forma, se asegura un flujo ordenado y sin interferencias entre los distintos procesos de carga y almacenamiento, mediante la imposición de transiciones, en este caso, *TB3* y *TB8*.

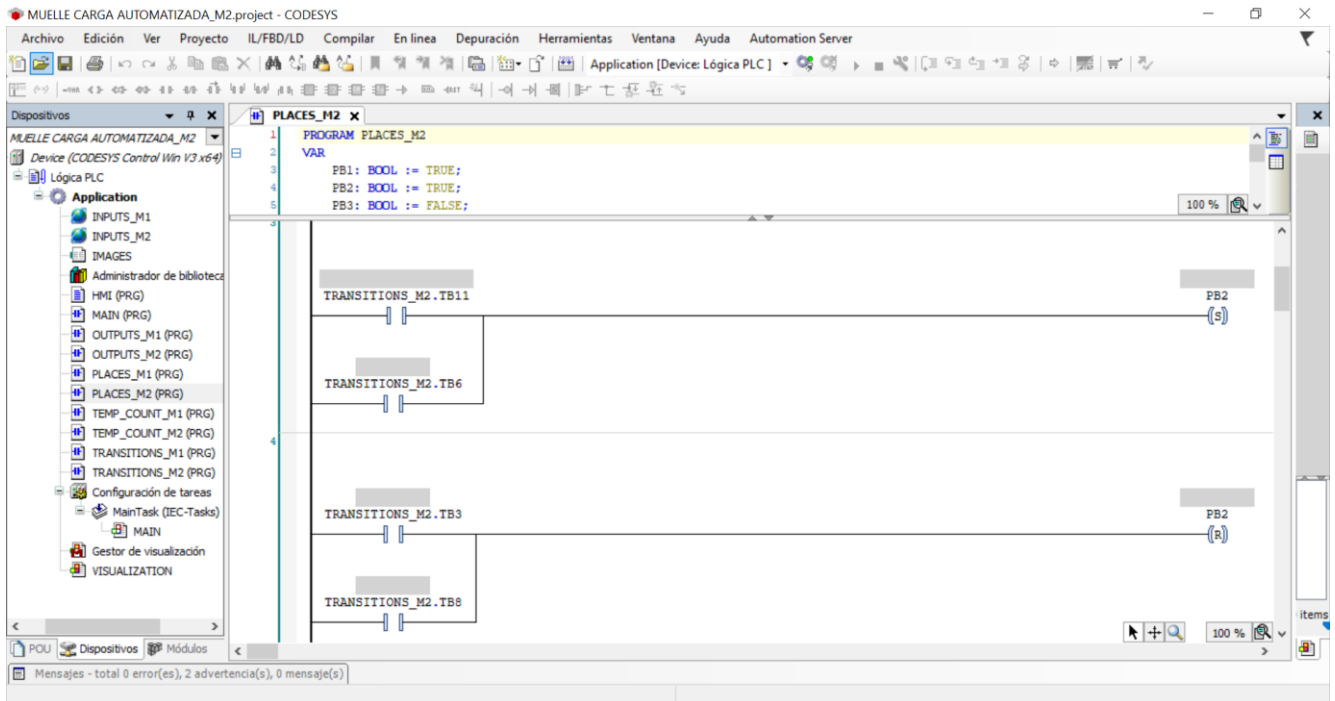


Figura 3-28 Programación Codesys del sistema de gestión de carga y almacenamiento en el muelle, bloque PLACES_M2.

En la Figura 3-29, se presenta el bloque *TRANSITIONS_M2*, con un énfasis particular en la transición *TA3*. Aquí, se observa cómo la programación establece una preferencia para la activación del lugar *PB3* sobre otros lugares competidores, mediante una lógica condicional en la que la transición solo puede activarse cuando se cumple que *PB2* está disponible y las condiciones previas han sido satisfechas. Esta estructura asegura que el sistema opera eficientemente, dando prioridad adecuada según las necesidades del proceso.

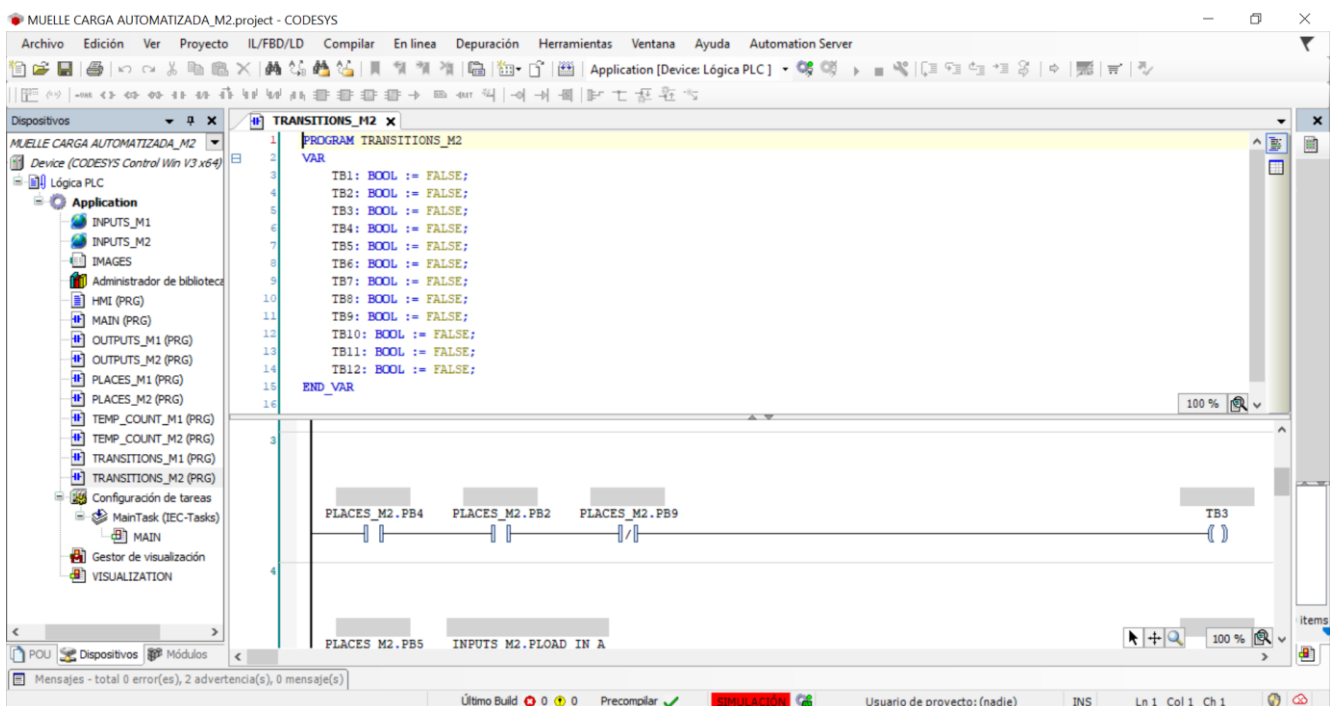


Figura 3-29 Programación Codesys del sistema de gestión de carga y almacenamiento en el muelle, bloque TRANSITIONS_M2.

La visualización del sistema desde la Figura 3-30 a la Figura 3-32 permite comprender claramente la operativa del muelle en diferentes estados. Concretamente, en la Figura 3-30, se observa el sistema en estado desactivado, donde todas las señales luminosas se encuentran apagadas, indicando que no hay actividad alguna en el muelle, y solo la luz del lugar de almacenamiento encendida, en su estado de espera.

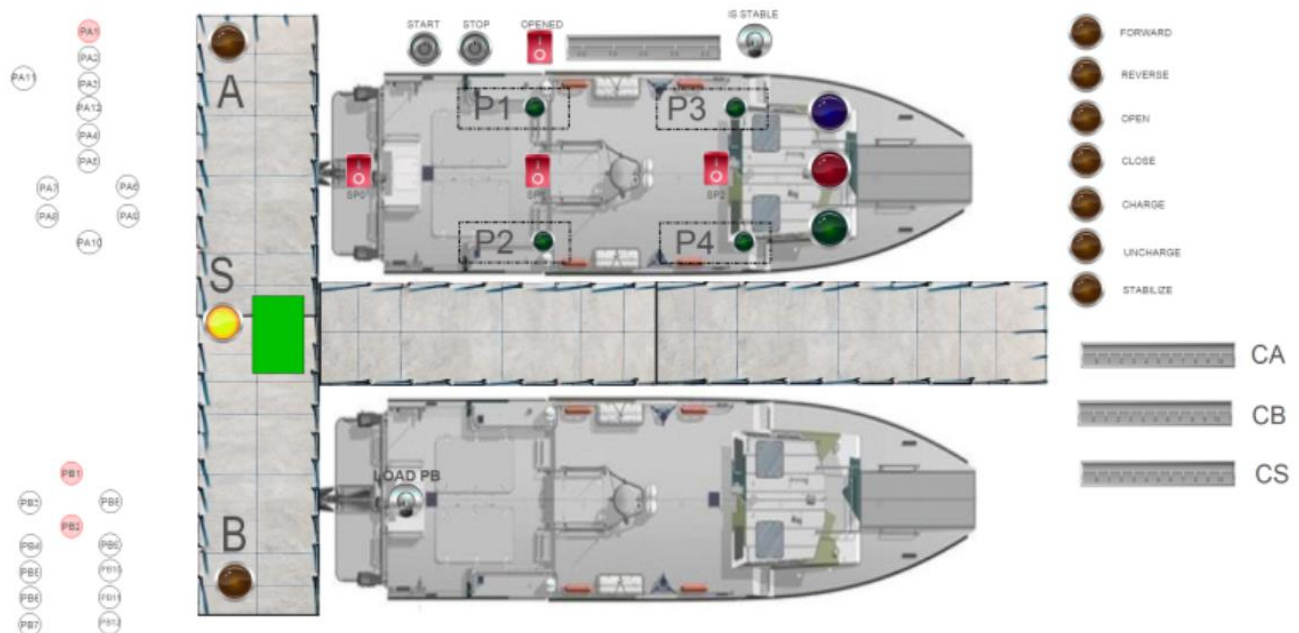


Figura 3-30 Representación visual del sistema de gestión de carga y almacenamiento en el muelle desactivado

En contraste, la Figura 3-31 muestra una situación activa, específicamente en la fase de descarga hacia la zona A de manera automática. Se activan las señales luminosas correspondientes que indican claramente que la plataforma se encuentra en operación, descargando en la zona de almacenamiento designada.

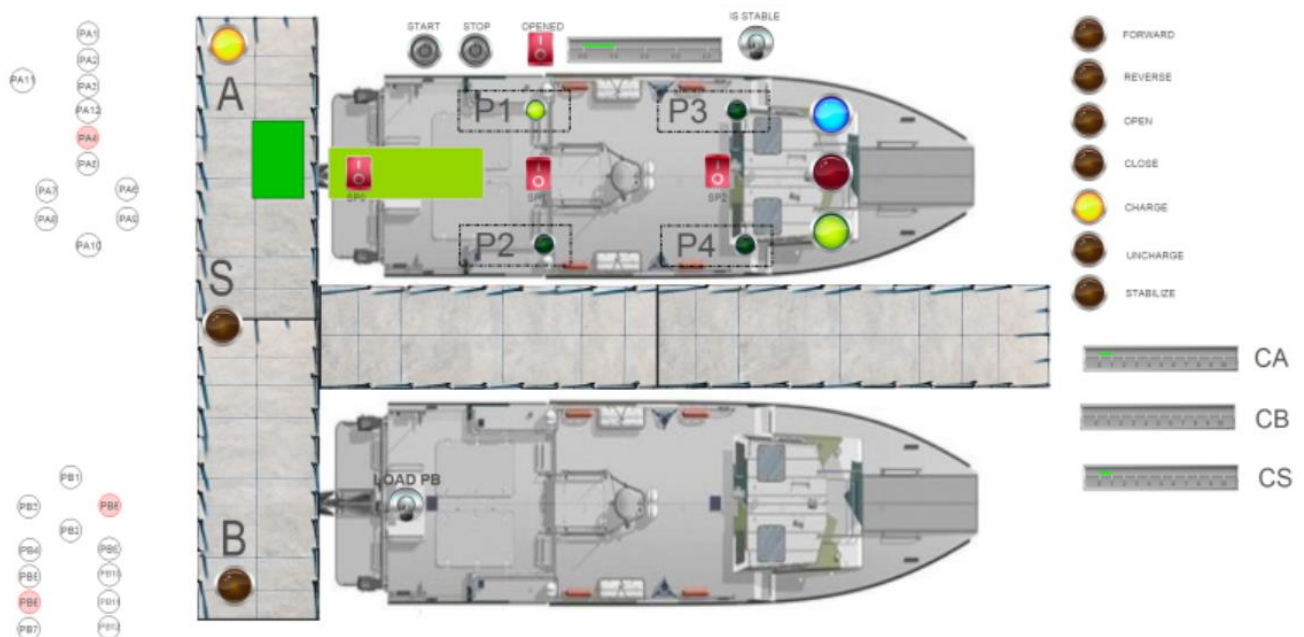


Figura 3-31 Representación visual del sistema de gestión de carga y almacenamiento en el muelle, descargando en A

Finalmente, en la Figura 3-32, se visualiza el proceso de descarga hacia la zona B activada de manera manual mediante un interruptor. En este estado, las señales indicadoras permiten identificar rápidamente el proceso activo, garantizando la correcta supervisión y facilitando la identificación de posibles problemas durante el proceso. Esta interfaz visual, clara y sencilla, facilita el monitoreo constante del sistema, asegurando que la gestión de carga y almacenamiento se realice con precisión y eficiencia.

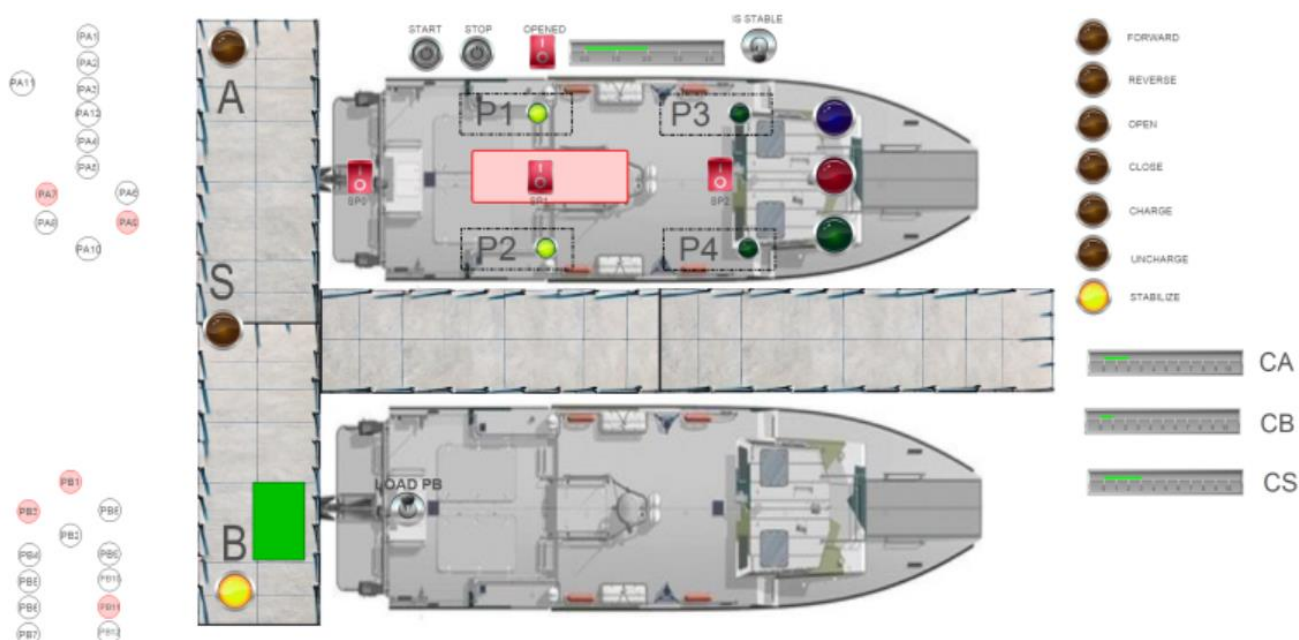


Figura 3-32 Representación visual del sistema de gestión de carga y almacenamiento en el muelle, descargando en B

Existen más estados posibles del sistema; estos son solo algunos ejemplos. Se ha realizado un video mostrando el funcionamiento del sistema, se puede acceder usando el siguiente enlace:

<https://drive.google.com/file/d/1U58t1OUOUNZ93noztUiXmeDFPEE7H2-j/view?usp=sharing>

3.2.4 Módulo 3: Gestión de energía y recarga de baterías

Este módulo se centra en la gestión energética y recarga de baterías de los dispositivos involucrados en el sistema de carga automatizada del muelle. La plataforma robótica encargada del transporte de carga y la lancha con sistema de carga interna dependen de baterías recargables, por lo que es fundamental garantizar su suministro continuo de energía que permita su correcto funcionamiento sin interrupciones operativas.

El sistema contempla la recarga de batería mediante dos procesos principales:

1. Sustitución de la batería de la grúa de carga: Cuando el nivel de carga de la plataforma cae por debajo de un umbral crítico, se activa el proceso de recarga empleando el acceso a una estación de carga compartida.
2. Recarga de la batería de la plataforma robótica: En este caso el proceso de carga consta de la extracción e inserción de un total de 3 baterías, del que se encarga la misma estación de carga compartida del punto anterior.

El sistema opera de manera autónoma, asegurando que los ciclos de carga se realicen de forma eficiente, minimizando tiempos de inactividad y evitando situaciones en las que las plataformas queden sin energía.

3.2.4.1 Enunciado

Para la automatización del proceso de gestión de energía y recarga de baterías, se deben considerar los siguientes aspectos a considerar:

- Inicio del sistema:
 - La operación comienza con la señal **START**, activando la supervisión de los niveles de batería.
- Monitoreo de la carga:
 - Si el nivel de carga de la grúa de carga es superior a un umbral crítico ($CB > 5$), se habilita la transición para el proceso de recarga de la batería de la grúa ().
 - Para la plataforma de carga, se inicia el procedimiento de sustitución de las baterías cada dos viajes completos que realice, y debe realizar siempre cuando la plataforma esté en reposo en el almacén.
- Proceso de recarga de la grúa de carga:
 - La batería se carga ejecutando la instrucción **CHARGE CRANE BATTERY**.
 - La batería se recarga en un tiempo determinado ($T3 = 0.5$), tras lo cual se considera que ha sido cargada.
- Procedimiento de sustitución de batería en la plataforma de carga necesita ejecutar 3 veces el siguiente ciclo de recarga:
 - Ejecutar la instrucción **REMOVE PLATFORM BATTERY** para eliminar una batería de la plataforma. Este proceso tarda 0.5 segundos en ejecutarse.
 - Una vez extraída, se inserta una nueva batería ejecutando la instrucción **INSERT PLATFORM BATTERY**, que también dura 0,5 segundos.
- Finalización del proceso:
 - Si la batería ha sido recargada correctamente, el sistema vuelve al estado de operación de espera a que se necesite recargar nuevamente.
 - Si estando en espera de necesitar una recarga, se recibe la señal **STOP**, el sistema se detiene y vuelve a su estado inicial.

3.2.4.2 Tabla de variables

En este apartado se presentan las variables utilizadas en la modelización del sistema de gestión de energía y recarga del muelle. Las variables de la Tabla 3-6 incluyen las entradas y salidas del sistema los temporizadores utilizados en la sincronización de las transiciones y los contadores que permiten el control de los eventos del sistema.

Define los elementos clave que intervienen en el proceso de carga y energía de las lanchas. Estas variables permiten modelar el funcionamiento del sistema asegurando que los que se realicen de manera eficiente y sin interrupciones.

Entradas	Salidas	Contadores	Temporizadores
	CHARGE CRANE BATTERY	C	T3
	REMOVE PLATFORM BATTERY		
	INSERT PLATFORM BATTERY		

Tabla 3-6 Variables del sistema de gestión de energía y recarga de baterías

3.2.4.3 Red de Petri de la gestión de energía y recarga de baterías

En la Figura 3-33 se presenta la red de Petri que describe el proceso automatizado de gestión de energía y recarga de baterías en el muelle. Inicialmente el sistema comienza desactivado, representado por el lugar *PB*. Al activarse la señal *START*, la transición *TB1* Es habilitada y el sistema pasa a monitorear continuamente los niveles de batería.

Cuando el nivel de carga de la grúa supera el umbral crítico ($CB > 5$), se habilita la transición correspondiente para recargar la batería de la grúa, activándose la instrucción *CHARGE CRANE BATTERY*, la cual se ejecuta por un periodo definido por el temporizador *T3*. Una vez que esta acción es completada, la transición vuelve al estado de espera *PB4*.

Por otro lado, la plataforma robótica activa su procedimiento de sustitución de baterías cada dos viajes completos, verificado mediante el contador interno. El ciclo de recarga consta de retirar (*REMOVE PLATFORM BATTERY*) y posteriormente insertar una nueva batería (*INSERT PLATFORM BATTERY*). Este proceso se repite tres veces, asegurado por la estructura de ciclos generada mediante lugares intermedios (*PB7* a *PB9*).

Si en cualquier momento se activa la señal *STOP* mientras el sistema espera una recarga o mientras no se realiza ninguna operación específica, el sistema vuelve al estado inicial *PB1*, garantizando que todos los procesos de recarga y gestión energética se detengan de forma segura y controlada.

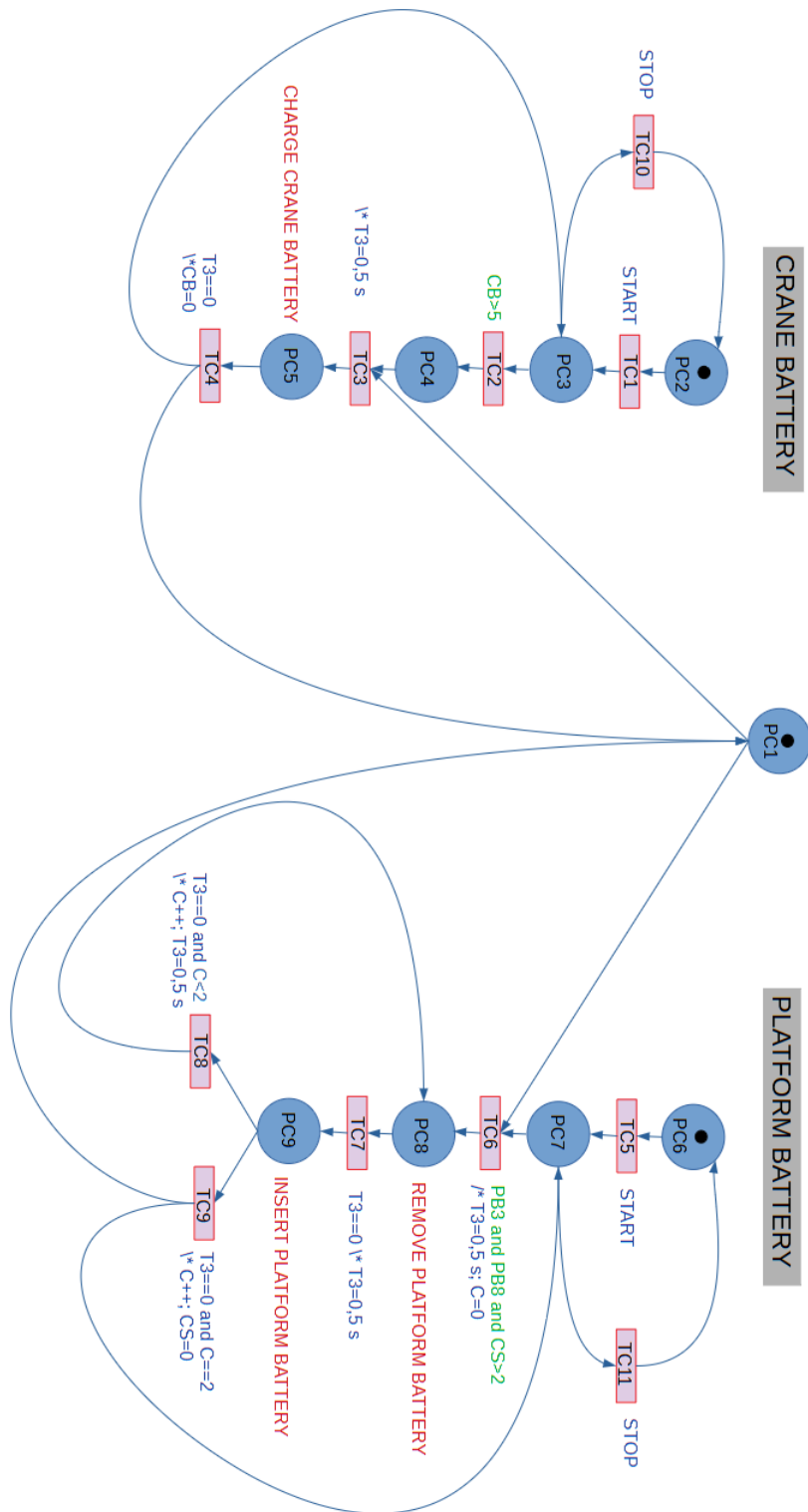


Figura 3-33 Red de Petri de la gestión de energía y recarga de baterías

3.2.4.4 Programación y visualización

La programación se ha realizado de la misma manera que en los anteriores módulos, mediante el lenguaje LD, siguiendo el esquema previamente planteado en la Figura 3-33.

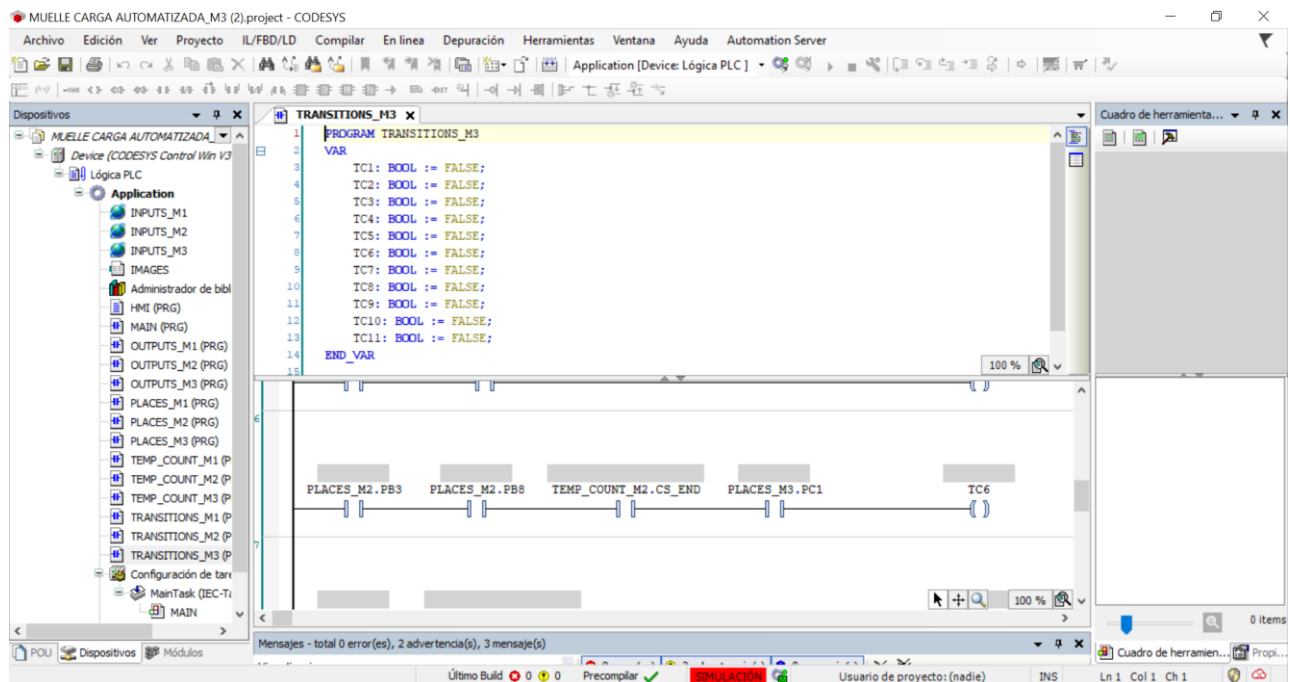


Figura 3-34 Programación Codesys del sistema de gestión de energía y recarga de baterías, bloque TRANSITIONS_M3.

En la Figura 3-34 , que muestra la implementación de *TRANSITIONS_M3*, se puede observar un ejemplo concreto en la transición *TC6*. Aquí, se verifica el estado actual del sistema mediante la evaluación simultánea de múltiples condiciones: una de ellas que el recurso compartido *PC1* este marcado, y además que dos lugares de espera de la red de Petri del segundo módulo estén marcados (*PB3* y *PB8*) y el contador *CS* sea mayor que 2.

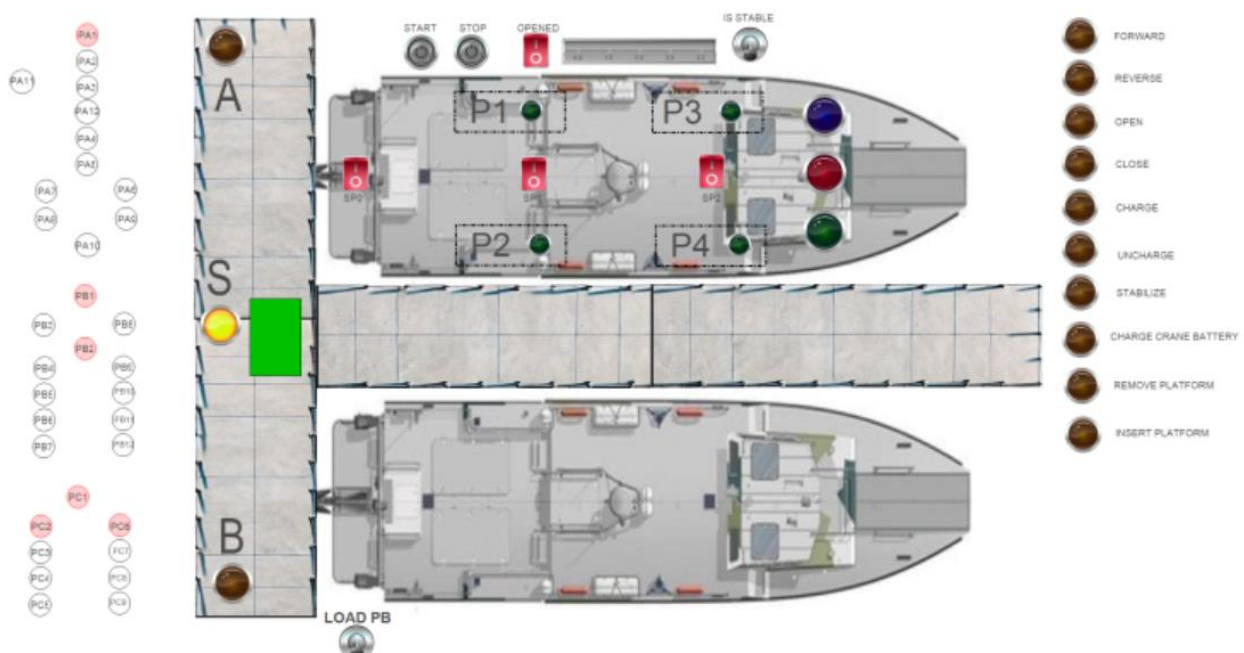


Figura 3-35 Representación visual de la gestión de energía y recarga de baterías, desactivado

La visualización del sistema, presentada en las Figura 3-35 a Figura 3-37, refleja claramente el estado operativo en tiempo real. En la Figura 3-35, se observa el sistema completamente desactivado, sin luces indicativas encendidas, mostrando que no hay operaciones en curso. En cambio, en la Figura 3-36, se visualiza el momento específico en que la plataforma robótica está en proceso de retirada de la batería (*REMOVE PLATFORM BATTERY*), indicado claramente mediante luces señalizadoras en el panel de control del sistema.

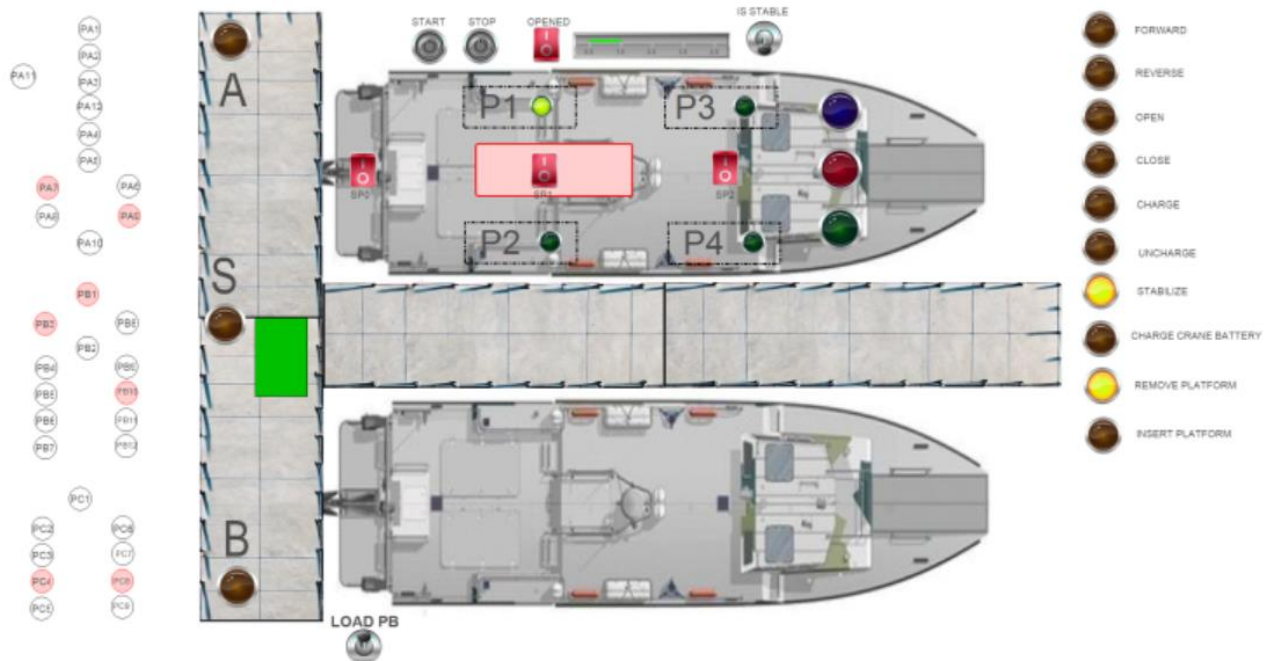


Figura 3-36 Representación visual de la gestión de energía y recarga de baterías, retirando la plataforma

Finalmente, en la Figura 3-37, se observa la fase en la que la batería de la grúa está siendo recargada, mostrando la activación de la luz indicadora correspondiente y la activación del lugar en la red de Petri. Este sistema visual permite un seguimiento fácil y eficiente del estado de carga y gestión energética, asegurando que cada proceso ocurra en el momento adecuado y en condiciones seguras para el personal y equipos involucrados.

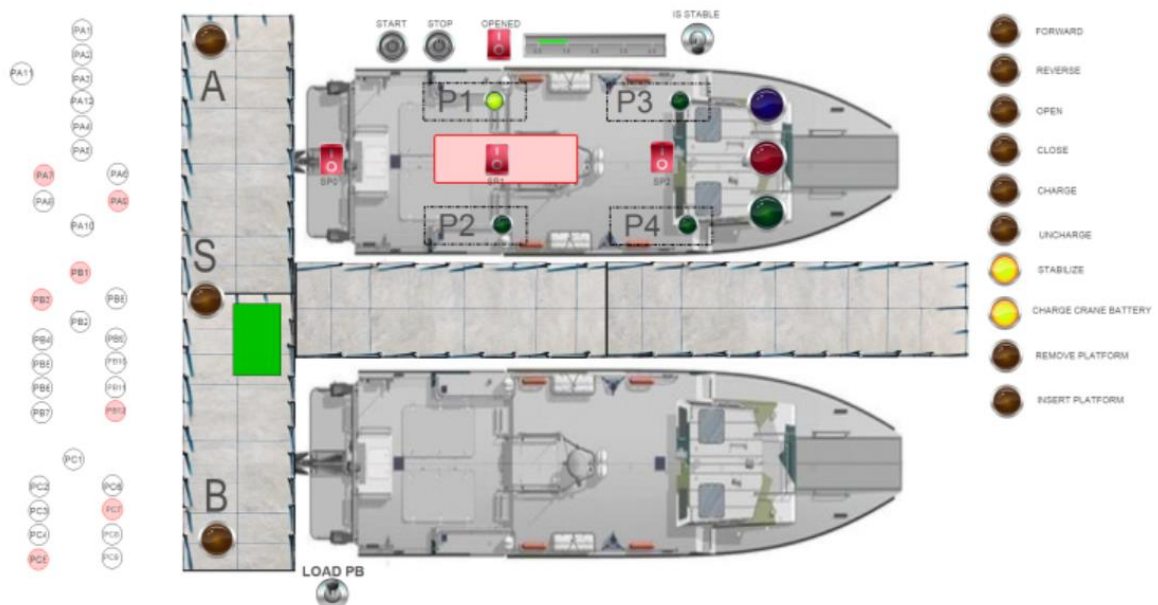


Figura 3-37 Representación visual de la gestión de energía y recarga de baterías, cargando la batería de la grúa

Existen más estados posibles del sistema; estos son solo algunos ejemplos. Se ha realizado un video mostrando el funcionamiento del sistema, se puede acceder usando el siguiente enlace:

<https://drive.google.com/file/d/16hzbNiRhY0p1yNNPPSD8sf97kz5W8NWJ/view?usp=sharing>

3.3 Proyecto 3: Automatización de la seguridad operativa y mantenimiento general en buques navales.

3.3.1 Descripción del sistema

Este proyecto aborda la automatización del plan de seguridad de un barco mediante el modelado con redes de Petri, centrándose en tres aspectos cruciales para la seguridad y la gestión de emergencias. En primer lugar, se desarrolla un sistema de detección y extinción de incendios para tres salas automatizadas del barco. Este sistema, en modo automático, monitoriza la temperatura en cada sala y activa automáticamente los extintores en caso de incendio, incorporando temporizadores para la duración de la activación y contadores para el seguimiento del uso de cada extintor. En modo manual, que será empleado para hacer las revisiones de mantenimiento, los extintores se activan de forma manual.

En segundo lugar, se implementa un sistema de conteo de personas y control de puertas, encargado de realizar un seguimiento del número de personas presentes en cada sala y gestionar la apertura automática de las puertas de una sala en caso de incendio una vez que se confirma la necesidad de realizar una evacuación.

Finalmente, se integra un sistema de mantenimiento automatizado encargado de supervisar el estado de los componentes críticos de la red de seguridad. Este sistema se encarga de programar el reemplazo de los extintores que hayan superado un límite de uso, verificar la funcionalidad de los sistemas de conteo y alarma mediante pruebas simuladas, y monitorizar el estado de las comunicaciones entre los diferentes subsistemas.

En conjunto, este proyecto busca mejorar significativamente la seguridad a bordo, optimizar la respuesta ante emergencias y facilitar las tareas de mantenimiento de los sistemas de seguridad del barco.

3.3.2 Módulo 1: Sistema de detección y extinción de incendios

Esta parte del proyecto se centra en la automatización de la detección y extinción de incendios en tres salas automatizadas del barco. Cada sala cuenta con un sensor de temperatura y extintores automáticos que deben ser activados en caso de detectarse un incendio.

1. Sensores y extintores:

- c. **Tres sensores de temperatura:** Programados para activar las señales **TEMP1**, **TEMP12** y **TEMP3** en caso de detectar un incendio en cada una de las 3 salas, respectivamente,
- d. **Tres extintores:** Responsables de apagar el fuego detectado en cada sala se activan ejecutando las instrucciones **EXTIN1**, **EXTIN2** y **EXTINT3**.

2. Otros elementos:

- a. **Luces de emergencia:** Empleadas para mostrar al personal si ha producido una alarma de incendio o si se va a realizar una prueba de mantenimiento que active los extintores.
- b. **Panel de control:** En donde se encuentran los mandos generales de control que permiten, por ejemplo, cambiar del modo automático al modo de pruebas manuales.

3.3.2.1 Enunciado

El sistema se enciende con la señal de entrada **START** y se apaga con la entrada **STOP** siempre y cuando no se encuentre activo ninguno de los sistemas de extinción.

Se dispone de un selector de modo de operación que puede seleccionar el modo automático (**MODE=0**) o el modo manual (**MODE=1**), que son los modos de operación del sistema y que se describen a continuación.

- **Modo Automático:**

- Si el sistema está activado en modo automático (**MODE=0**), los sensores de temperatura **TEMP1**, **TEMP2** y **TEMP3** monitorean continuamente cada sala y se deben activar automáticamente los extintores mediante la activación de las señales **EXTIN1**, **EXTIN2** o **EXTIN3** cuando cada uno de los sensores indica que hay fuego en una de las salas.
- Al activar un extintor se debe encender la luz roja activando la señal **LR1**, **LR2** o **LR3** según la sala en la que se active el extintor.

- **Modo Manual:**

- En este modo (**MODE=1**), los extintores solo pueden activarse desde el panel de control mediante los selectores que generan las señales **ACTIVATE1**, **ACTIVATE2** y **ACTIVATE3**, lo que permite realizar pruebas de mantenimiento o simulaciones de emergencia sin que los sensores de temperatura automaticen la respuesta.
- En este modo de operación se establece un temporizador de 0,5 segundos para controlar la duración de la activación de cada uno de los extintores, asegurando que el sistema vuelva a su estado estable una vez finalizada la operación.
- Al activar un extintor se debe encender la luz azul activando la señal **LB1**, **LB2** o **LB3** según la sala en la que se active el extintor.

Cabe destacar que, en caso de que se cambie el modo de funcionamiento (automático/manual) durante el uso de un extintor, se volverá de inmediato a su estado de espera. El sistema debe emplear los **contadores** (**CE1**, **CE2**, **CE3**) para registrar el número de activaciones de cada extintor.

3.3.2.2 Tabla de variables

En este apartado se presentan las variables involucradas en el funcionamiento del sistema de detección y extinción de incendios a bordo del barco. Estas variables se dividen en entradas, salidas, contadores y temporizadores, siendo esenciales para gestionar adecuadamente la lógica automatizada del sistema.

La Tabla 3-7 muestra claramente cada uno de los componentes del sistema. Las entradas incluyen señales críticas para el control del sistema, tales como **START** y **STOP** para encender y apagar el sistema, y **MODE** para seleccionar entre operación automática y manual. Asimismo, se incluyen sensores de temperatura (**TEMP1**, **TEMP2**, **TEMP3**) que permiten detectar incendios en las diferentes salas del barco y señales manuales para la activación directa de los extintores (**ACTIVATE1**, **ACTIVATE2**, **ACTIVATE3**).

Las salidas corresponden a las acciones que ejecuta el sistema ante determinadas condiciones. Aquí se encuentran los comandos de activación para los extintores automáticos (**EXTINT1**, **EXTINT2**, **EXTINT3**), así como luces indicadoras que informan visualmente del estado del sistema durante una emergencia o durante una prueba manual (**LR1**, **LR2**, **LR3** y **LB1**, **LB2**, **LB3**).

Por otro lado, los contadores (**CE1**, **CE2**, **CE3**) permiten llevar un registro preciso del número de activaciones realizadas por cada extintor, facilitando tareas de mantenimiento preventivo. Finalmente, los temporizadores (**TE1**, **TE2**, **TE3**) controlan la duración específica de activación de los extintores, asegurando un uso eficiente de los recursos disponibles.

La adecuada definición y utilización de estas variables garantizan una respuesta inmediata y precisa del sistema ante cualquier evento, incrementando la seguridad general a bordo.

Entradas	Salidas	Contadores	Temporizadores
START	EXTINT1	CE1	TE1
STOP	EXTINT2	CE2	TE2
MODE	EXTINT3	CE3	TE3
TEMP1	LR1		
TEMP2	LR2		
TEMP3	LR3		
ACTIVATE1	LB2		
ACTIVATE2	LB2		
ACTIVATE3	LB3		

Tabla 3-7 Variables del sistema de detección y extinción de incendios

3.3.2.3 Red de Petri

La Figura 3-38 muestra la red de Petri desarrollada para el sistema de detección y extinción de incendios. Este modelo ilustra el comportamiento automatizado del sistema, mostrando claramente los diferentes estados (lugares) y las condiciones para pasar de un estado a otro (transiciones).

La red inicia en un estado inactivo representado por el lugar inicial *P1*, en espera de la activación del sistema mediante la señal *START*. Una vez activo, el sistema monitorea constantemente la temperatura en las diferentes salas. Si algún sensor (*TEMP1*, *TEMP2* o *TEMP3*) detecta un nivel de temperatura que indica un incendio, se habilitan las transiciones correspondientes (*T3*, *T7* o *T11*) para activar automáticamente los extintores en la sala afectada (*EXTINT1*, *EXTINT2* o *EXTINT3*).

En modo manual, las transiciones *T5*, *T9* y *T13* permiten activar manualmente los extintores mediante los comandos específicos (*ACTIVATE1*, *ACTIVATE2*, *ACTIVATE3*). Este modo emplea temporizadores para gestionar el tiempo exacto de activación de los extintores y para retornar al estado de espera al finalizar la intervención manual.

Finalmente, la red incorpora transiciones que permiten regresar al estado de monitoreo una vez que la condición crítica ha sido resuelta, ya sea automática o manualmente. La inclusión de contadores en cada transición de activación permite además llevar un control riguroso del número de usos de los extintores, facilitando su mantenimiento.

Esta representación gráfica mediante la red de Petri proporciona una herramienta valiosa para visualizar, analizar y verificar el comportamiento del sistema, asegurando así una operación segura y efectiva ante situaciones de emergencia a bordo del barco

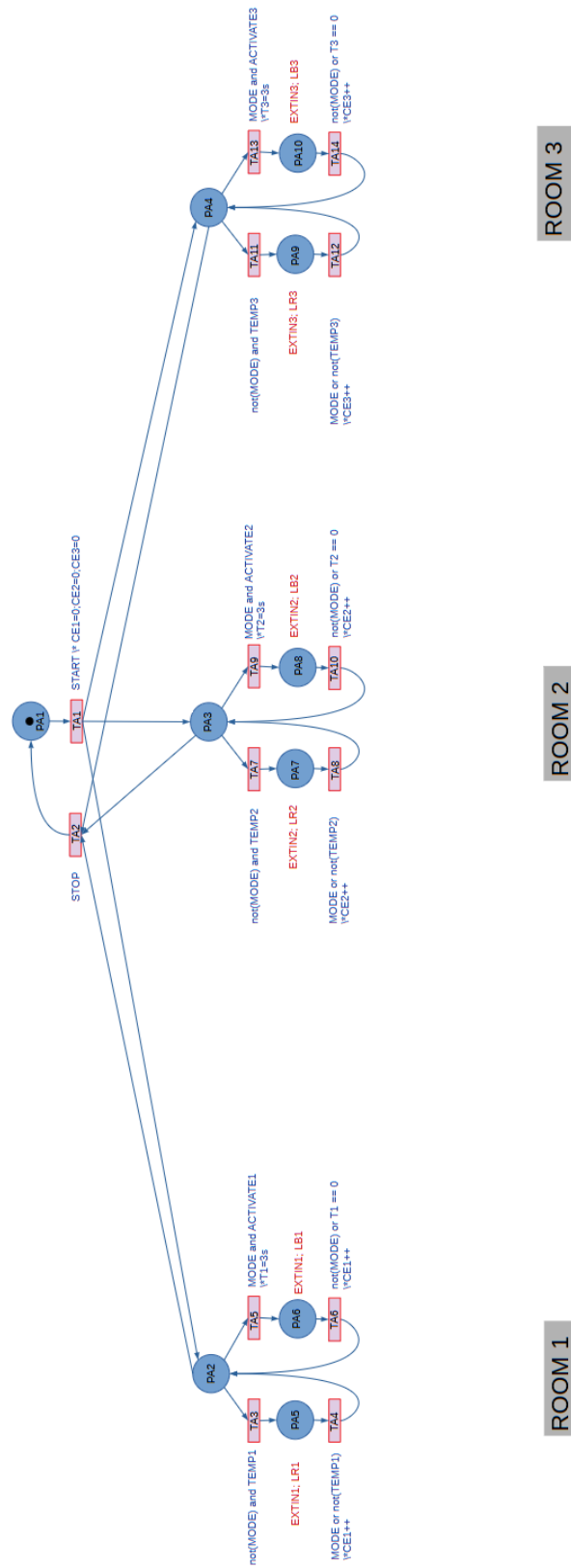


Figura 3-38 Red de Petri del sistema de detección y extinción de incendios

3.3.2.4 Programación y visualización

La implementación práctica del sistema de detección y extinción de incendios, representado mediante la red de Petri, se ha llevado a cabo empleando lenguaje LD dentro del entorno de desarrollo Codesys, tal como muestra desde la Figura 3-39 a la Figura 3-41 presentadas.

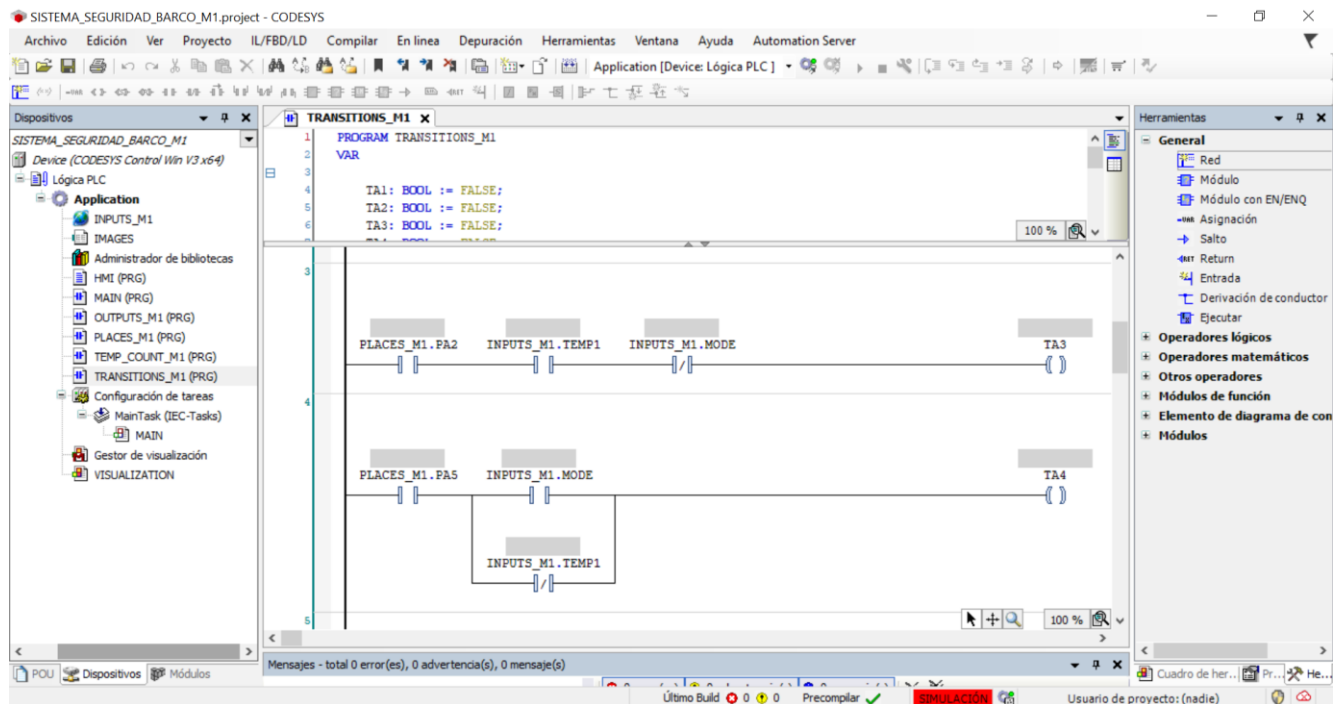


Figura 3-39 Programación Codesys del sistema de detección y extinción de incendios, bloque TRANSITIONS_M1

La Figura 3-39 correspondiente al bloque *TRANSITIONS_M1* muestra claramente la programación de la transición *TA3*, que es activada cuando el sistema está en modo automático (*INPUTS_M1.MODE*) y detecta una temperatura elevada (*INPUTS_M1.TEMP1*). Al cumplirse estas condiciones, se ejecuta la transición *TA3* y el sistema pasa al lugar *PA5*, activando el extintor *EXTINT1* y la luz indicadora *LR1*. Posteriormente, cuando la temperatura regresa a valores seguros o el sistema sale del modo automático, la transición *TA4* se activa retornando al sistema al lugar *PA2* para continuar monitoreando.

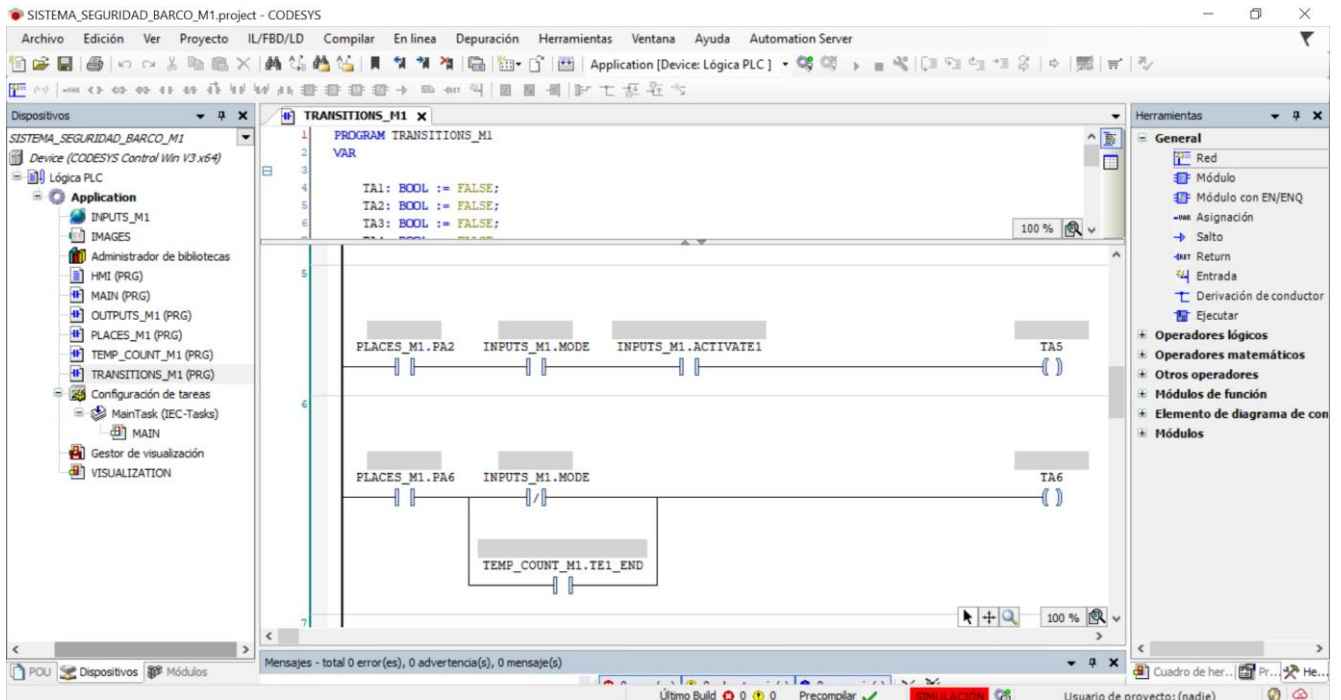


Figura 3-40 Programación Codesys del sistema de detección y extinción de incendios, bloque TRANSITIONS_M1

En el modo manual, la transición *TA5* es activada mediante la señal directa *INPUTS_M1.ACTIVATE1* junto con la confirmación del modo manual, también reflejada en la Figura 3-40 correspondiente al bloque *TRANSITIONS_M1*. Al activarse, lleva el sistema al lugar *PA6*, desde donde el extintor es activado manualmente. La duración de activación es controlada por el temporizador *TEMP_COUNT_M1.TE1_END*, que al cumplirse permite la activación de la transición *TA6*, devolviendo el sistema nuevamente al lugar *PA2*, listo para seguir supervisando.

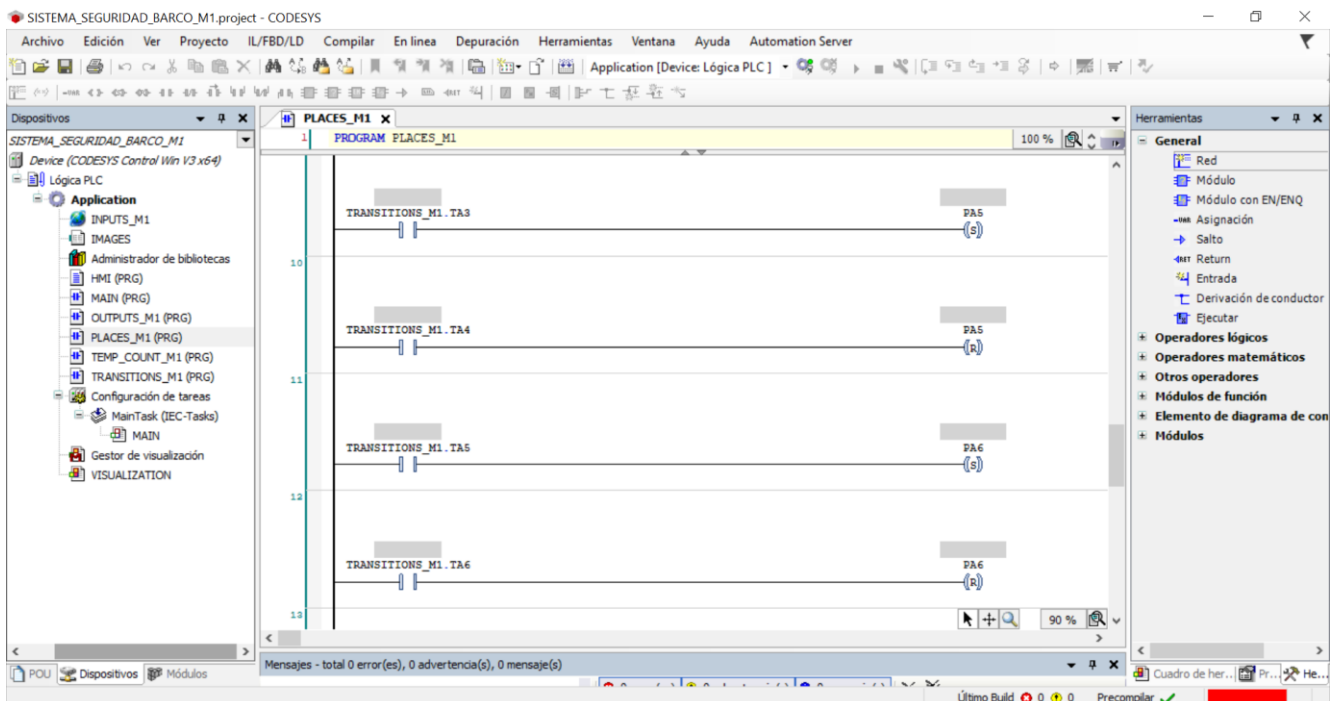


Figura 3-41 Programación Codesys del sistema de detección y extinción de incendios, bloque PLACES_M1

El bloque *PLACES_M1* mostrado en la Figura 3-41 confirma visualmente esta lógica de operación. Por ejemplo, el lugar *PA5* se activa (*set*) mediante la transición *TA3* y se desactiva (*reset*) mediante la transición *TA4*, indicando claramente la gestión del estado operativo del sistema durante un evento.

Cabe destacar que las otras dos habitaciones, *ROOM 2* y *ROOM 3*, operan bajo una lógica y programación idéntica, variando únicamente en los sensores de temperatura y los elementos de activación específicos para cada habitación. Esto permite que el sistema sea uniforme, simplificando el mantenimiento y asegurando la eficacia en la respuesta ante emergencias.

En la Figura 3-42, se observa que el sistema se encuentra en un estado totalmente desactivado. El lugar inicial (*PA1*) está activado, lo que indica que el sistema está en reposo, esperando una orden de inicio. Los botones *START* y *STOP* están en posición de reposo, y el selector de modo indica *AUTO*, pero no está operativo debido a la desactivación general del sistema. La temperatura en las tres habitaciones (*TEMP1*, *TEMP2*, *TEMP3*) está por debajo del umbral crítico de 60 grados, representado en las escalas visuales. Ninguna señal visual en las habitaciones está activada, confirmando que no existe ningún evento ni acción en curso.

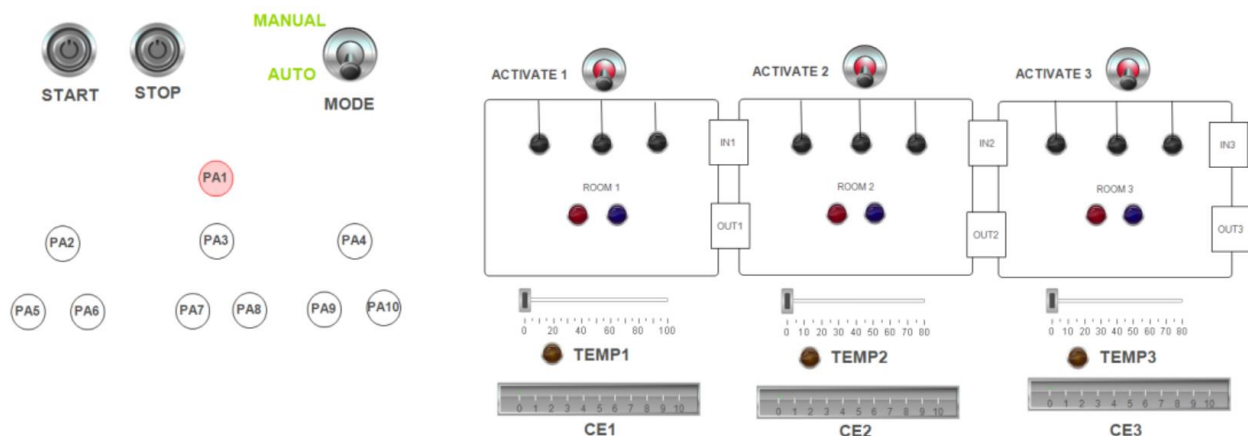


Figura 3-42 Representación visual del sistema de detección y extinción de incendios, desactivado

En la Figura 3-43, el sistema está activo en modo manual, reflejado claramente en el selector de modo. Se observa que los lugares activos son *PA3*, *PA4* y *PA6*, indicando que hay una intervención manual en curso en la *ROOM 1*. Esto es confirmado visualmente por la activación de la luz azul correspondiente a la extinción manual en la *ROOM 1*. La temperatura sigue en niveles normales, y el contador de eventos (*CE1*) muestra una activación registrada. Las otras habitaciones no presentan actividad alguna, permaneciendo con temperaturas normales y sin indicaciones visuales encendidas.

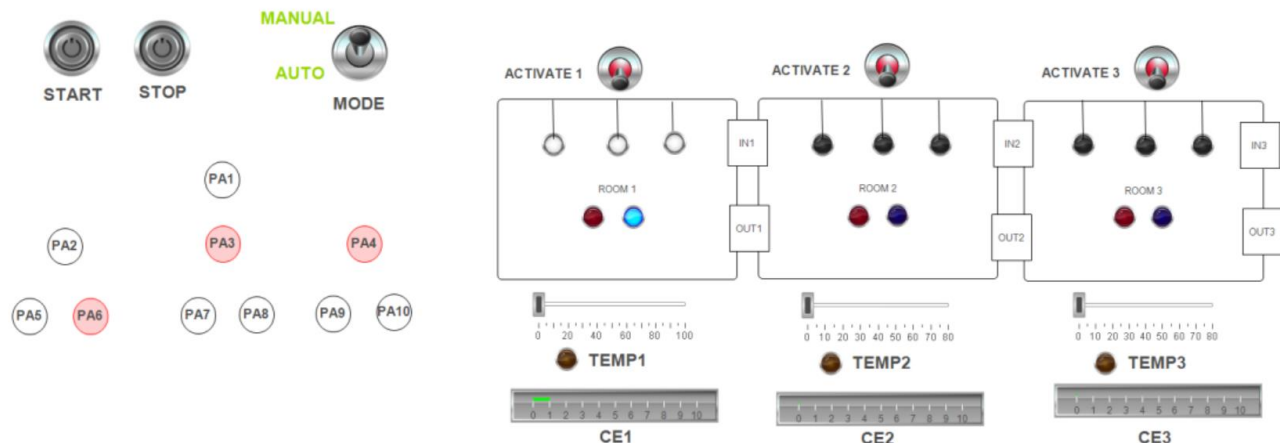


Figura 3-43 Representación visual del sistema de detección y extinción de incendios, modo manual

La Figura 3-44 muestra al sistema en funcionamiento en modo automático. En este modo, la temperatura umbral crítica para activar la extinción automática ha sido establecida en 60 grados. Claramente se observa una emergencia en la *ROOM 2*, ya que la temperatura ha sobrepasado el umbral crítico, indicado por el medidor de temperatura *TEMP2* resaltado en amarillo. La luz roja en la *ROOM 2* indica una activación automática del sistema de extinción. Los lugares *PA2*, *PA4* y *PA7* están activos, evidenciando gráficamente el estado operativo de emergencia. Además, el contador *CE2* refleja una activación registrada por el evento actual. Las demás habitaciones se mantienen en condiciones normales y sin acciones registradas, excepto que el extintor en la *ROOM 1* ha sido usado en una ocasión, representado en el contador *CE1*.

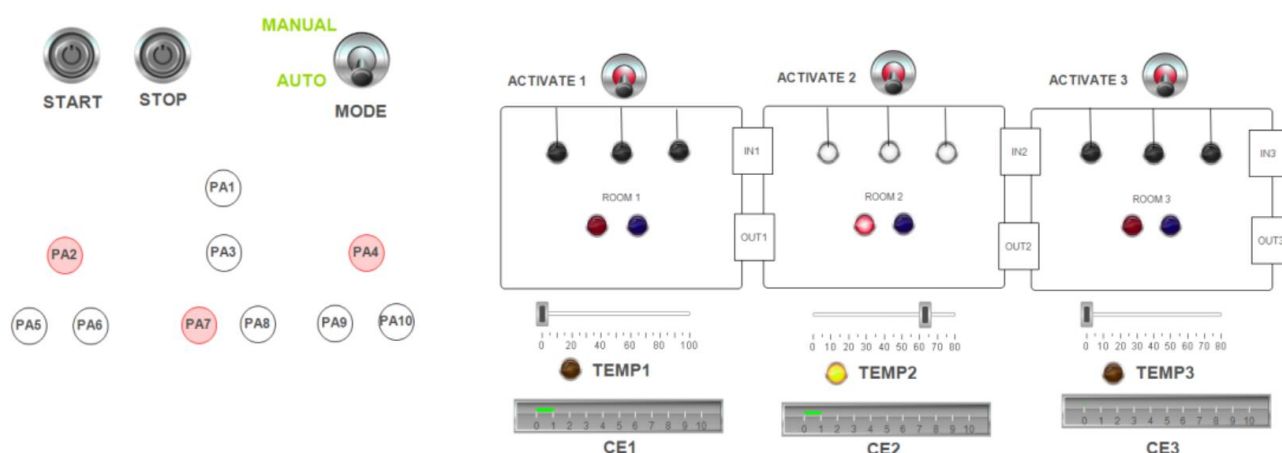


Figura 3-44 Representación visual del sistema de detección y extinción de incendios, automático

Estos análisis visuales permiten corroborar en tiempo real el estado de cada componente y habitación del sistema, facilitando el seguimiento inmediato del proceso automático o manual durante una emergencia

De nuevo, existen más estados posibles del sistema; estos son solo algunos ejemplos. Se ha realizado un video mostrando el funcionamiento del sistema, se puede acceder usando el siguiente enlace:

<https://drive.google.com/file/d/1IPcCx-NCOXV0hFKl09dggCzKQppGiIO9/view?usp=sharing>

3.3.3 Módulo 2: Sistema de conteo de personas y control de puertas.

Esta parte del proyecto se centra en la automatización del proceso de seguimiento de las personas que se encuentran en cada sala, teniendo en cuenta que cada sala tiene una capacidad máxima para 4 personas. Aparte de los elementos expuestos en el módulo 1, en este módulo se añaden los siguientes elementos

1. Contaje y control de personas:

- Control de acceso:** Existen sensores que se activan las señales *IN1*, *IN2* e *IN3* cuando una persona entra en una sala, así como *OUT1*, *OUT2* y *OUT3* cuando una persona sale de una sala.
- Bloqueo de acceso:** El acceso de cada sala puede ser bloqueado activando las señales *BIN1*, *BIN2* y *BIN3*, para el bloqueo de acceso de entradas, y *BOUT1*, *BOUT 2* y *BOUT3* para el bloqueo de acceso de salidas de las salas.

3.3.3.1 Enunciado

De igual forma al módulo 1, el sistema se enciende con la señal de entrada **START** y se apaga con la entrada **STOP** siempre y cuando se encuentre en una situación con todas las salas cerradas. Se debe asumir que inicialmente todas las salas deben estar vacías y cerradas.

El sistema debe permitir la entrada en una sala únicamente si aún existe aforo, de forma que en caso de que la sala esté llena, el acceso debe permanecer en modo de bloqueo. Cada vez que se produzca un evento de este tipo, la red debe modificar el valor de los contadores necesarios para mantener el control del número de personas en cada sala.

Por otra parte, estando en modo automático, en caso de que exista un fuego en alguna de las salas, se deben abrir todos los accesos de entrada y salida de las salas para favorecer su evacuación. Esto implica la necesidad de modificar la red de Petri del módulo 1 para establecer una condición que indique si estando en modo automático se ha detectado fuego en alguna de las salas.

Por último, estando en modo manual no debe permitirse el accionamiento del extintor de una sala si hay personas en su interior. En modo automático, los extintores deben funcionar sin importar si hay personas en la sala

3.3.3.2 Tabla de variables

El sistema implementado para el conteo de personas y control de puertas está basado en un conjunto específico de entradas, salidas y contadores que permiten el monitoreo y la gestión del acceso a las salas. Como se puede observar en la Tabla 3-8, las variables de entrada *IN1*, *IN2* e *IN3* representan los sensores que detectan la entrada de personas en las respectivas salas, mientras que las variables *OUT1*, *OUT2* y *OUT3* se encargan de registrar las salidas. Las salidas *BLOCK IN* y *BLOCK OUT* están destinadas a bloquear los accesos en situaciones específicas, dependiendo del aforo registrado en cada sala por los contadores *CP1*, *CP2* y *CP3*. Por otro lado, las variables *OPEN IN* y *OPEN OUT* posibilitan la apertura de accesos cuando el sistema determina que es seguro permitir la entrada o salida de personas.

Entradas	Salidas	Contadores	Temporizadores
IN1	BLOCK IN1	CP1	
IN2	BLOCK IN2	CP2	
IN3	BLOCK IN3	CP3	
OUT1	BLOCK OUT1		
OUT2	BLOCK OUT2		
OUT3	BLOCK OUT3		
	OPEN IN1		
	OPEN IN2		

OPEN IN3

OPEN OUT1

OPEN OUT2

OPEN OUT3

Tabla 3-8 Variables del sistema de control de personas y control de puertas

3.3.3.3 Red de Petri

Para representar el funcionamiento del sistema descrito, se han empleado las redes de Petri mostradas en la Figura 3-45 y Figura 3-46. Estas redes permiten visualizar claramente las condiciones de activación o bloqueo de entradas y salidas mediante lugares y transiciones.

La red de Petri de la Figura 3-45 experimenta modificaciones con la adición de dos lugares adicionales: *PA11* y *PA12*. Estos nuevos lugares representan específicamente la presencia o ausencia de fuego o incendio en las salas. El lugar *PA11* indica la ausencia total de incendios, mientras que *PA12* señala que existe fuego activo en una o más salas, permitiendo así gestionar automáticamente la respuesta del sistema en situaciones de emergencia.

La Figura 3-46 muestra tres secciones, una para cada sala (*ROOM 1*, *ROOM 2* y *ROOM 3*). Cada sala tiene lugares y transiciones específicas que permiten la entrada y salida (*OPEN IN* y *OPEN OUT*) o el bloqueo de estas (*BLOCK IN* y *BLOCK OUT*). Por ejemplo, el lugar *PB2* o *PB4* para *ROOM 1* se activan sin ninguna condición el sistema se enciende con todo bloqueado. La transición *TB4* y *TB2* abre el acceso, que bloqueado cuando el contador *CP1* indica aforo completo. El funcionamiento es similar en las otras dos salas, con lugares equivalentes y transiciones asociadas a cada contador y sensor.

Además, el lugar *PA12* aparece como condición de activación de distintas transiciones, debido a que la condición de existencia de fuego influye directamente en el funcionamiento del sistema, determinando si los accesos deben permanecer bloqueados o abiertos para facilitar la evacuación inmediata en caso de incendio.

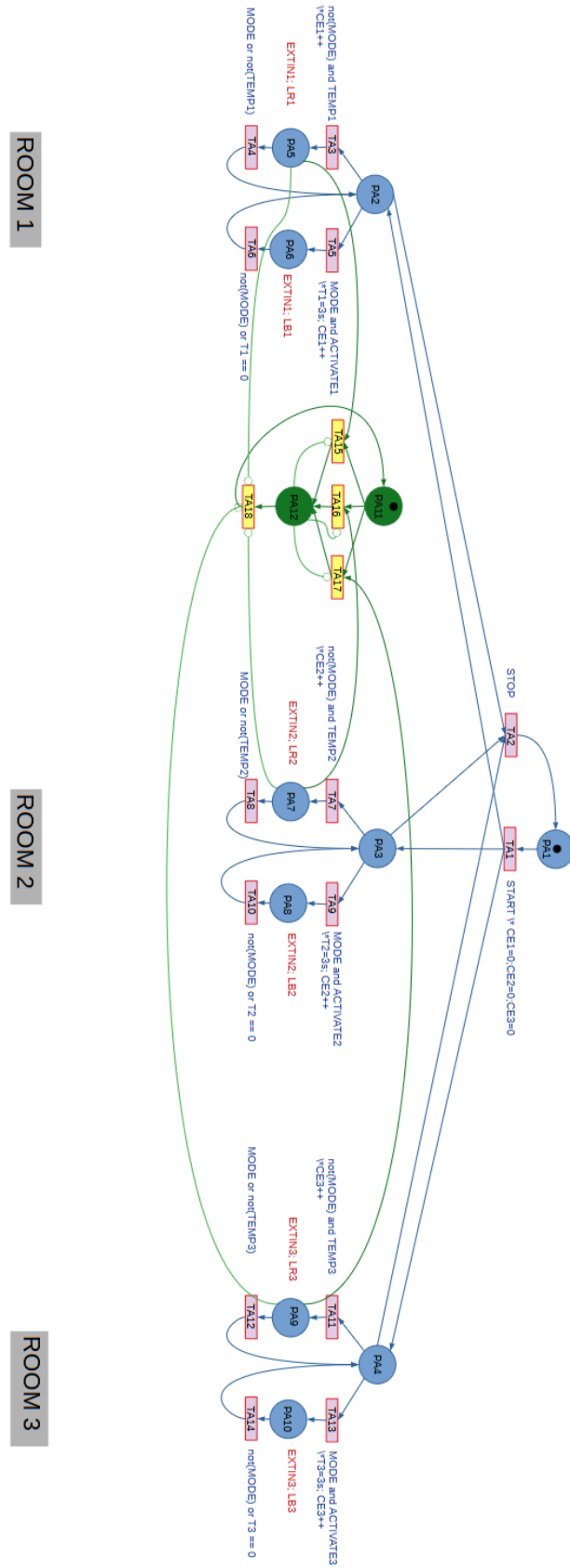


Figura 3-45 Red de Petri del sistema de control de personas y control de puertas (1)

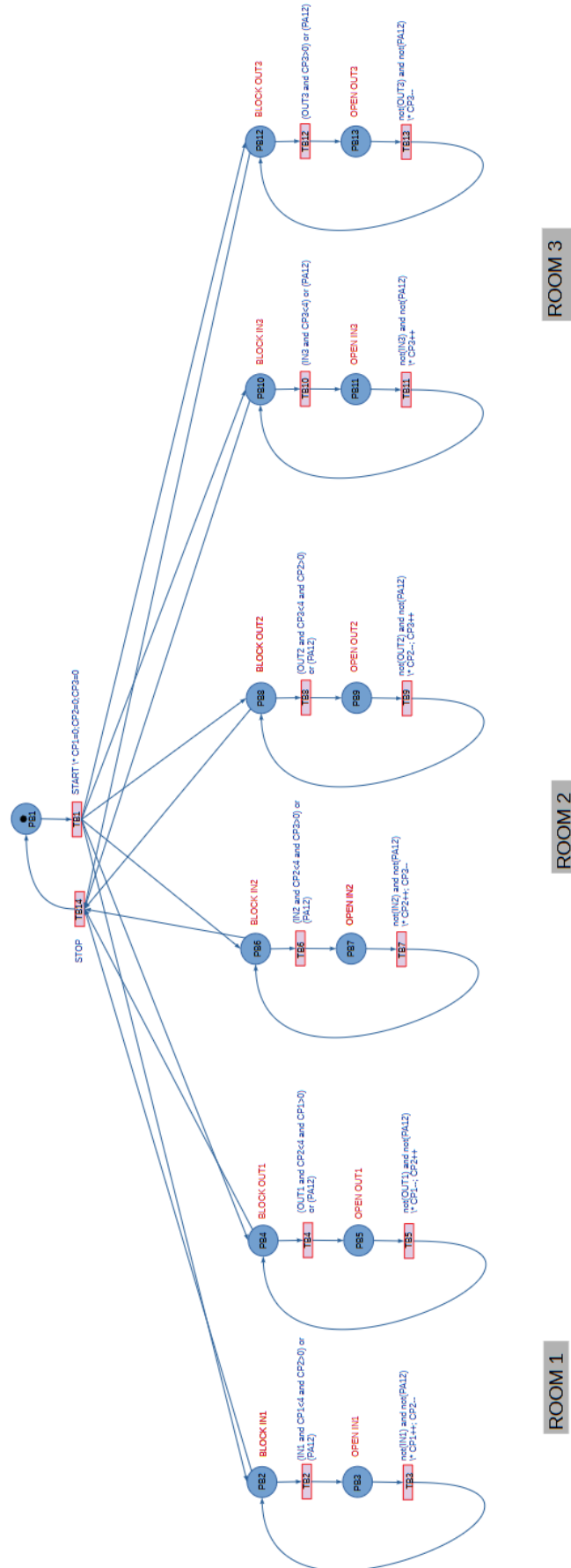


Figura 3-46 Red de Petri del sistema de control de personas y control de puertas (2)

3.3.3.4 Programación y visualización

Tal como se muestra desde la Figura 3-47 a la Figura 3-49, se presenta el bloque *PLACES_M2*. Centrándose en una de las habitaciones del sistema, *ROOM 1*, la lógica programada refleja claramente las condiciones necesarias para activar o desactivar los lugares específicos (*PB2*, *PB3*, *PB4*, *PB5*). Por ejemplo, *PB2* se activa mediante la transición *TB1* cuando se cumplen ciertas condiciones de entrada, mientras que se desactiva mediante *TB14*. De forma similar, el lugar *PB3* controla otras condiciones específicas al activarse por *TB2* y desactivarse por *TB3*. Este enfoque permite al sistema manejar eficazmente el control de acceso según las condiciones actuales en la sala.

Aunque este análisis detalla la *ROOM 1*, la lógica de funcionamiento y programación de las otras dos salas (*ROOM 2* y *ROOM 3*) sigue un patrón similar. La estructura lógica es consistente en todas las salas, variando únicamente en las variables y condiciones específicas asociadas a cada habitación, lo que facilita la uniformidad en el desarrollo, implementación y mantenimiento.

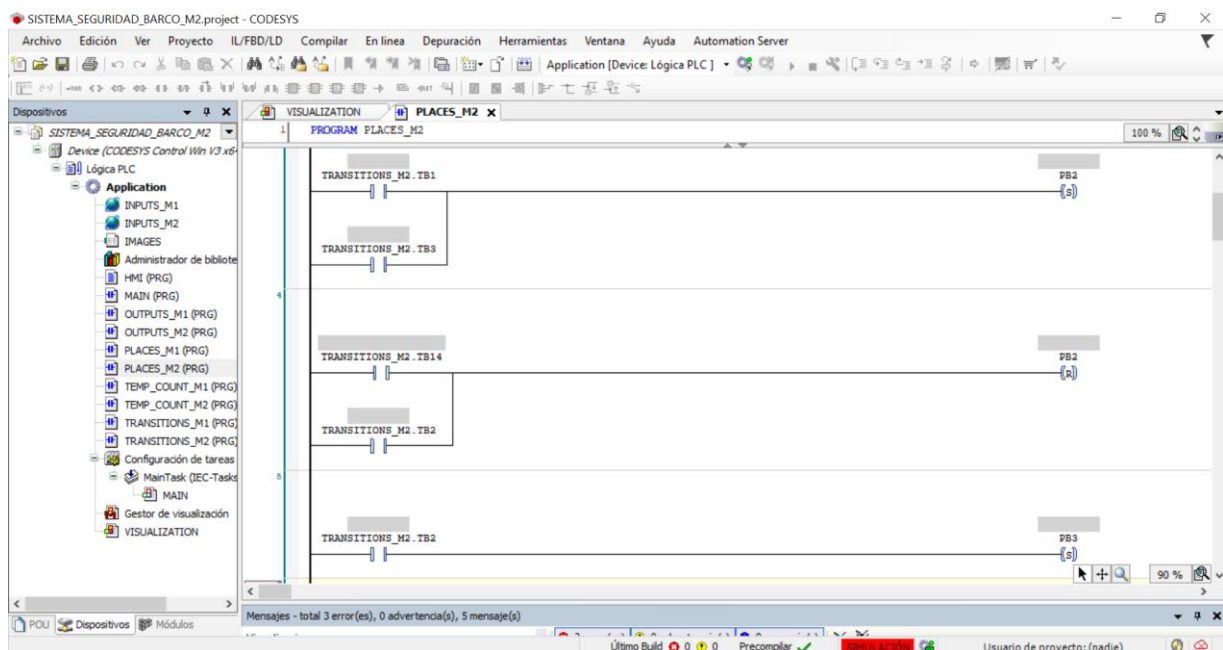


Figura 3-47 Programación Codesys del sistema de control de personas y control de puertas, bloque *PLACES_M2* (1)

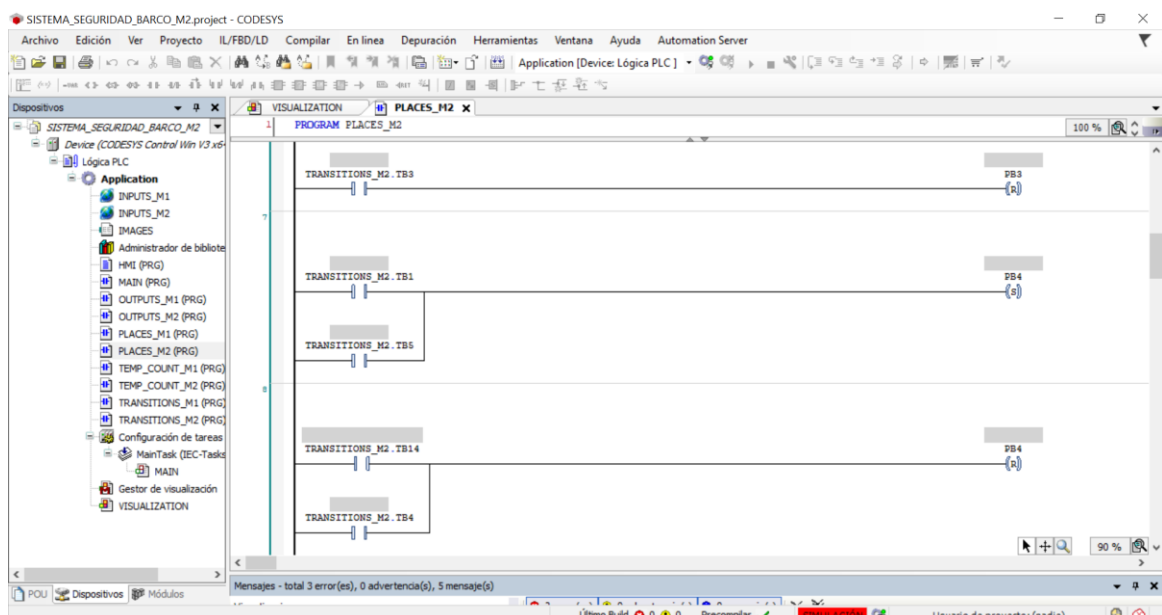


Figura 3-48 Programación Codesys del sistema de control de personas y control de puertas, bloque *PLACES_M2* (2)

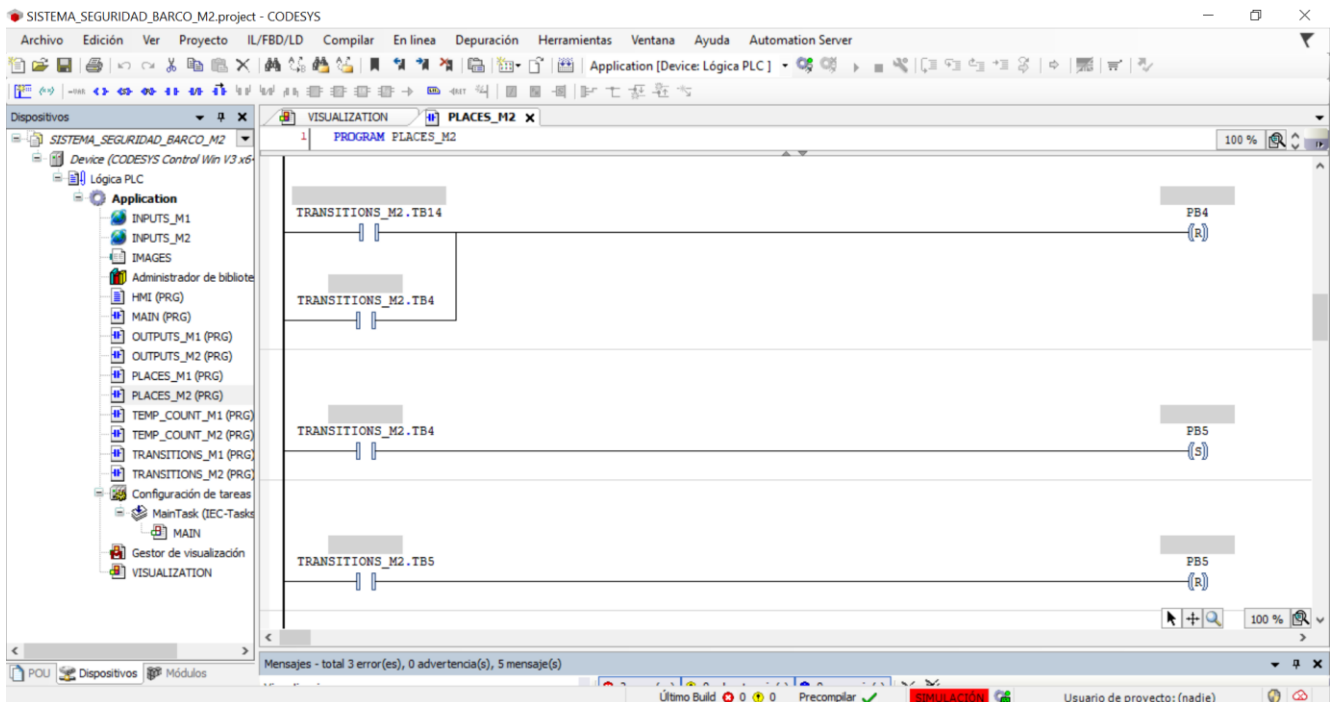


Figura 3-49 Programación Codesys del sistema de control de personas y control de puertas, bloque PLACES_M2 (3)

La programación mediante lenguaje LD, según muestran la Figura 3-50 y la Figura 3-51 proporcionadas, refleja la lógica del sistema descrito previamente. Tomando como referencia una de las habitaciones (ROOM 1), se observa cómo las transiciones *TB1*, *TB2* y *TB3* controlan estados específicos definidos por lugares como *PB2*, *PB3* y *PB3*. La transición *TB1* activa el lugar *PB2* cuando se reciben las señales adecuadas (*INPUTS_M1.START*) bloqueando la puerta, mientras que la transición *TB2* gestiona condiciones adicionales relacionadas con el contador *CP1*, *CP2* y la entrada *IN1*, o también el lugar *PA12* que representa la condición crítica de fuego activo. La transición *TB3* se habilita cuando no hay fuego activo y permite condiciones seguras de acceso.

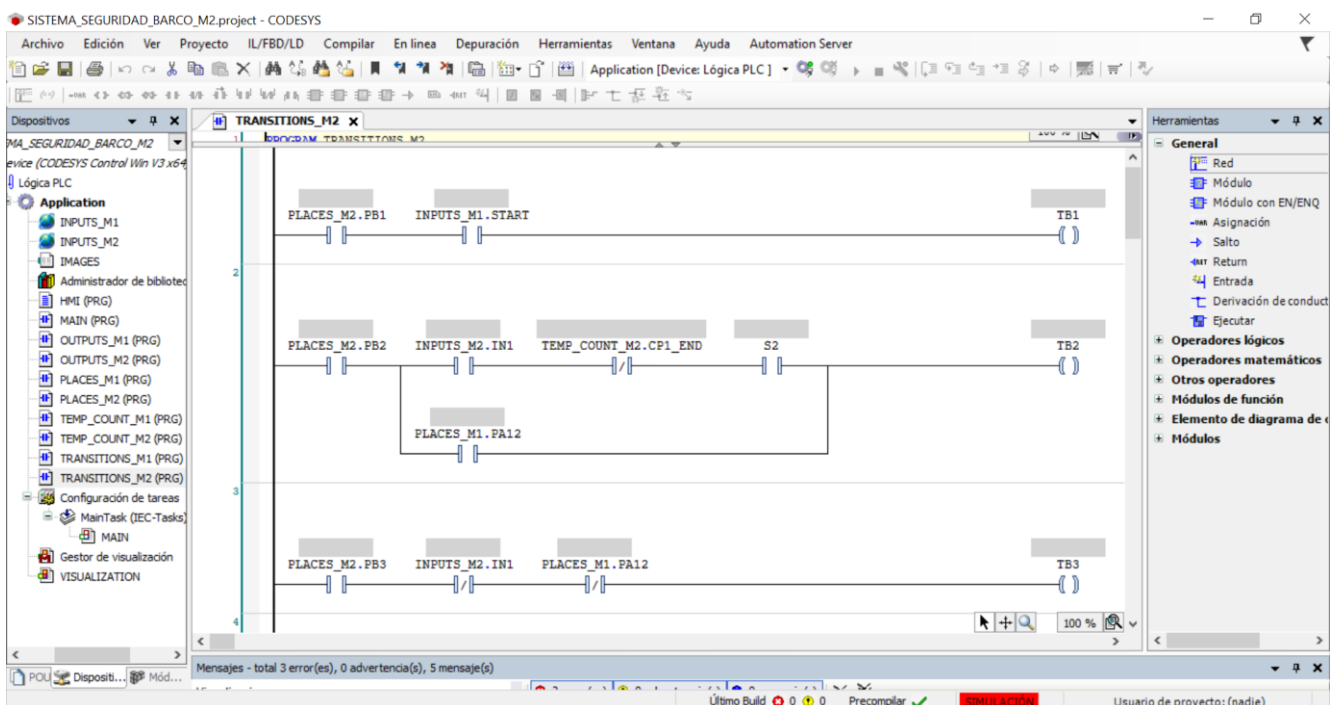


Figura 3-50 Programación Codesys del sistema de control de personas y control de puertas, bloque TRANSITIONS_M2 (1)

Este mismo patrón lógico se observa con otras transiciones como *TB4* y *TB5*, las cuales manejan otras variables para garantizar el correcto manejo de salidas y la seguridad en situaciones de emergencia.

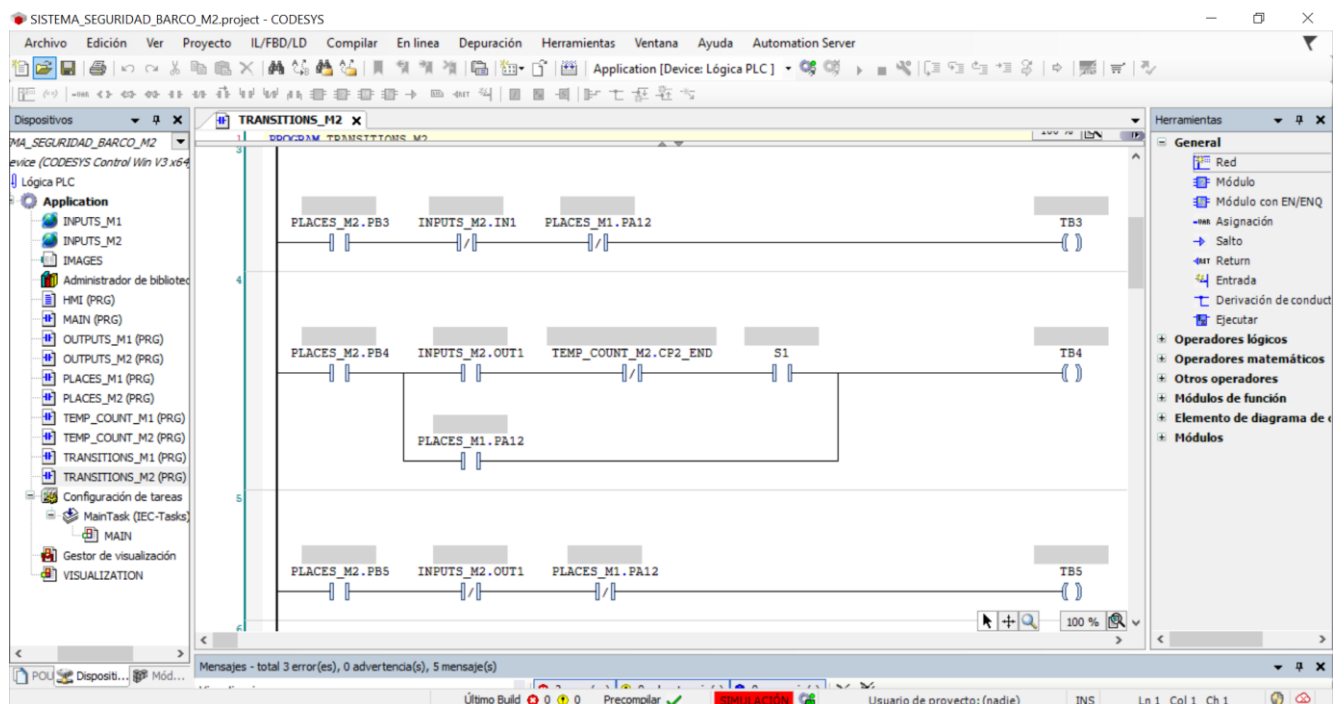


Figura 3-51 Programación Codesys del sistema de control de personas y control de puertas, bloque TRANSITIONS_M2 (2)

La Figura 3-52 y Figura 3-53 mostradas ilustran la interfaz gráfica correspondiente a este módulo. En la Figura 3-52, el sistema se encuentra en estado desactivado, lo cual puede evidenciarse al observar que la mayoría de los indicadores gráficos aparecen sin resaltar. Por el contrario, la Figura 3-53 muestra el sistema activado, diferenciándose claramente mediante los círculos resaltados en color rojo que indican los estados activos de diversas variables y lugares (como por ejemplo *PA2*, *PA11*, *PB2*, *PB4*, entre otros).

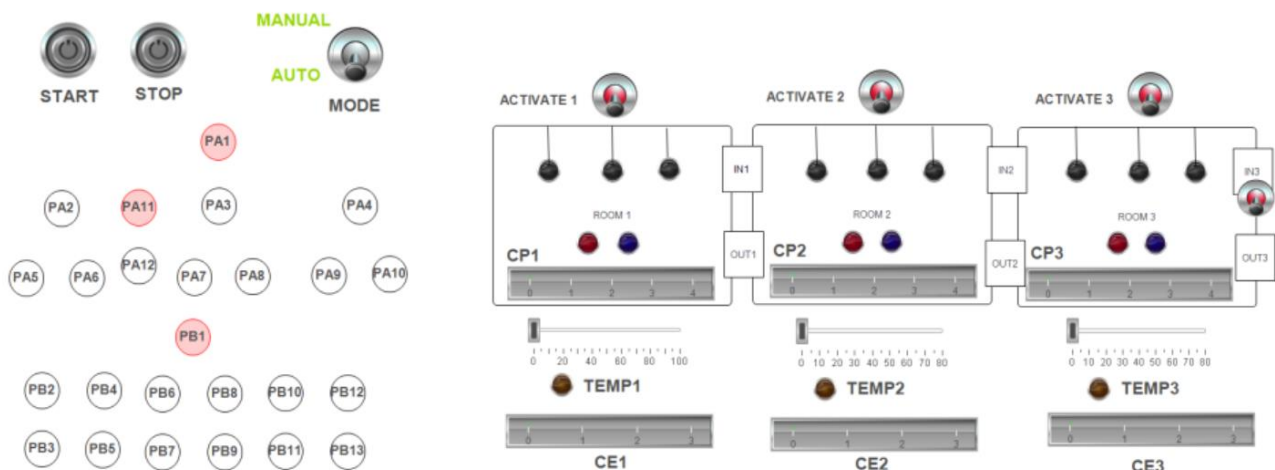


Figura 3-52 Representación visual del sistema de control de personas y control de puertas, desactivado

Adicionalmente, se han incorporado en la interfaz los interruptores que simbolizan los sensores de entrada (*IN1*, *IN2*, *IN3*) y salida (*OUT1*, *OUT2*, *OUT3*) ubicados en las puertas de cada sala, indicando si el acceso está habilitado o bloqueado en función del conteo de personas y condiciones establecidas

por el sistema. También es visible que el contador *CE1* registra varias activaciones, mostrando gráficamente el número de personas que han ingresado a la sala 1.

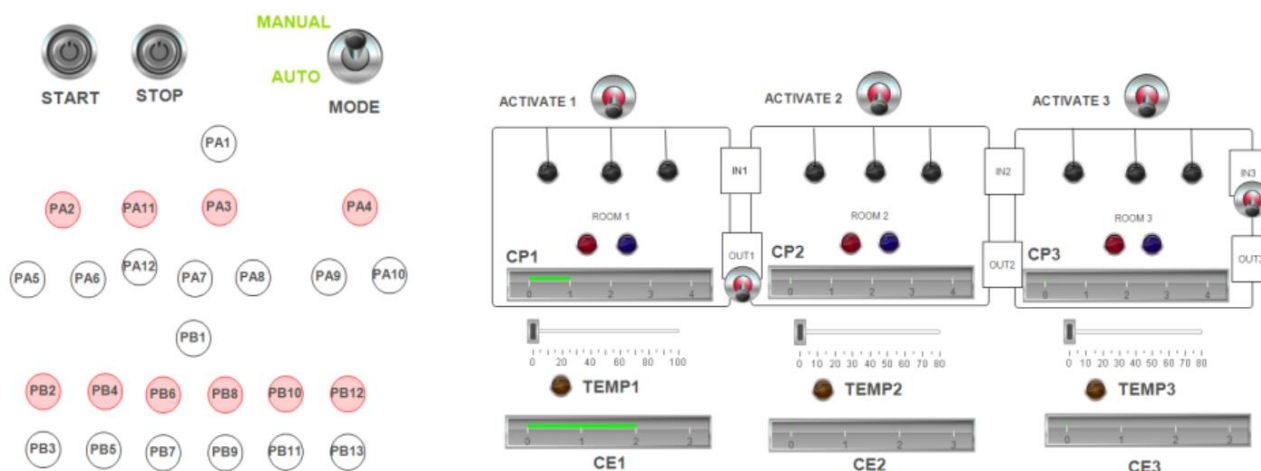


Figura 3-53 Representación visual del sistema de control de personas y control de puertas, activado

Debido a que el comportamiento dinámico y las transiciones entre los estados son más claramente perceptibles en movimiento, se recomienda la visualización del video disponible en el enlace proporcionado a continuación, donde se aprecia de forma más efectiva y detallada el funcionamiento real del módulo.:

<https://drive.google.com/file/d/1lXgWLYup0ucYuOkpWKWuX66S2lTD0ltb/view?usp=sharing>

3.3.4 Módulo 3: Sistema de mantenimiento automatizado

En este último módulo se pretende gestionar el sistema automatizado de recarga de los extintores, el cual es compartido por las tres salas. Los extintores de cada sala deben cambiarse cada vez que hayan sido activados 3 veces en la sala. La sala 3 tiene preferencia de acceso sobre la sala 2, y la sala 2 tiene preferencia de acceso sobre la sala 1.

Además, estando en modo automático, en caso de que exista un fuego en alguna de las salas debe bloquearse el accionamiento de la recarga de los extintores de esa sala, ya que en estas situaciones los extintores operan con el remanente de seguridad del de disponen cuando van a ser cambiados.

Por otra parte, si se está realizando una en recarga y en modo automático se produce un incendio, se debe dejar la recarga si ya se ha cargado como mínimo un extintor de los 3 totales.

3.3.4.1 Enunciado

De igual forma a los módulos anteriores, el sistema se enciende con la señal de entrada **START** y se apaga con la entrada **STOP** siempre y cuando se encuentre en una situación en la que no está realizando una recarga.

En caso de que se detecte que una de las salas ha activado sus extintores 3 o más veces, se debe realizar su recarga, para lo cual se activan las instrucciones **CHARGE1**, **CHARGE2** Y **CHARGE3** que finalizan al recibir las señales **END CHARGE1**, **END CHARGE2** Y **END CHARGE3**. El proceso de recarga debe repetirse 3 veces para considerar que los extintores de una sala se encuentran recargados.

Tal y como se comentó, el proceso de recarga debe abandonarse si estando en modo automático se produce un incendio en la sala en donde se está produciendo la recarga y ya se ha recargado como mínimo un extintor.

3.3.4.2 Tabla de variables

En este apartado se presenta un análisis detallado de las variables que intervienen en el sistema propuesto, las cuales han sido resumidas en la Tabla 3-9 adjunta. Las variables definidas corresponden tanto a entradas como salidas y estados intermedios del proceso, permitiendo así un control adecuado y efectivo del mismo.

Las señales de entrada *END CHARGE1*, *END CHARGE2* y *END CHARGE3* indican que el proceso de recarga de los extintores de cada sala ha finalizado. Cuando estas señales se activan, las salidas correspondientes *CHARGE E1*, *CHARGE E2* y *CHARGE E3* se encargan de iniciar un nuevo ciclo de recarga para cada extintor. Los contadores *C1*, *C2* y *C3* registran cuántos ciclos de carga se han realizado, permitiendo gestionar la necesidad de mantenimiento preventivo y asegurar así que cada extintor mantenga su funcionalidad de manera óptima.

Entradas	Salidas	Contadores	Temporizadores
END CHARGE1	CHARGE E1	C1	
END CHARGE2	CHARGE E2	C2	
END CHARGE3	CHARGE E3	C3	

Tabla 3-9 Variables del sistema de mantenimiento automatizado

3.3.4.3 Red de Petri

La Figura 3-54 ilustra la red de Petri del sistema de mantenimiento automatizado. Este modelo permite observar claramente la lógica operativa mediante lugares y transiciones. Inicialmente, el sistema se encuentra en estado de espera en el lugar inicial (*PC1*), pendiente de la señal *START* para comenzar la operación. Cuando alguno de los contadores (*C1*, *C2* o *C3*) indica que se han completado tres activaciones del extintor correspondiente, se activan transiciones específicas junto con más condiciones y lugares que llevan al sistema desde hacia los lugares *PC4*, *PC7* o *PC10*, iniciando así los procesos de recarga representados por las salidas *CHARGE E1*, *CHARGE E2* y *CHARGE E3* respectivamente.

Una vez iniciado, el proceso avanza repitiéndose tres veces hasta completarse totalmente. La evolución del proceso está regulada por las señales *END CHARGE*, que habilitan transiciones entre estos lugares para continuar con el siguiente ciclo o regresar al estado de espera *PC1*. Adicionalmente, la red presenta condiciones específicas en lugares de control intermedios que permiten abandonar la recarga en caso de detectarse un incendio en modo automático, siempre que se haya recargado al menos un extintor, reflejando una prioridad absoluta en situaciones de emergencia.

La estructura lógica de esta red, asociada claramente a los lugares mencionados, asegura tanto la continuidad del mantenimiento preventivo como una respuesta rápida y efectiva ante emergencias, incrementando notablemente la seguridad y confiabilidad del sistema.

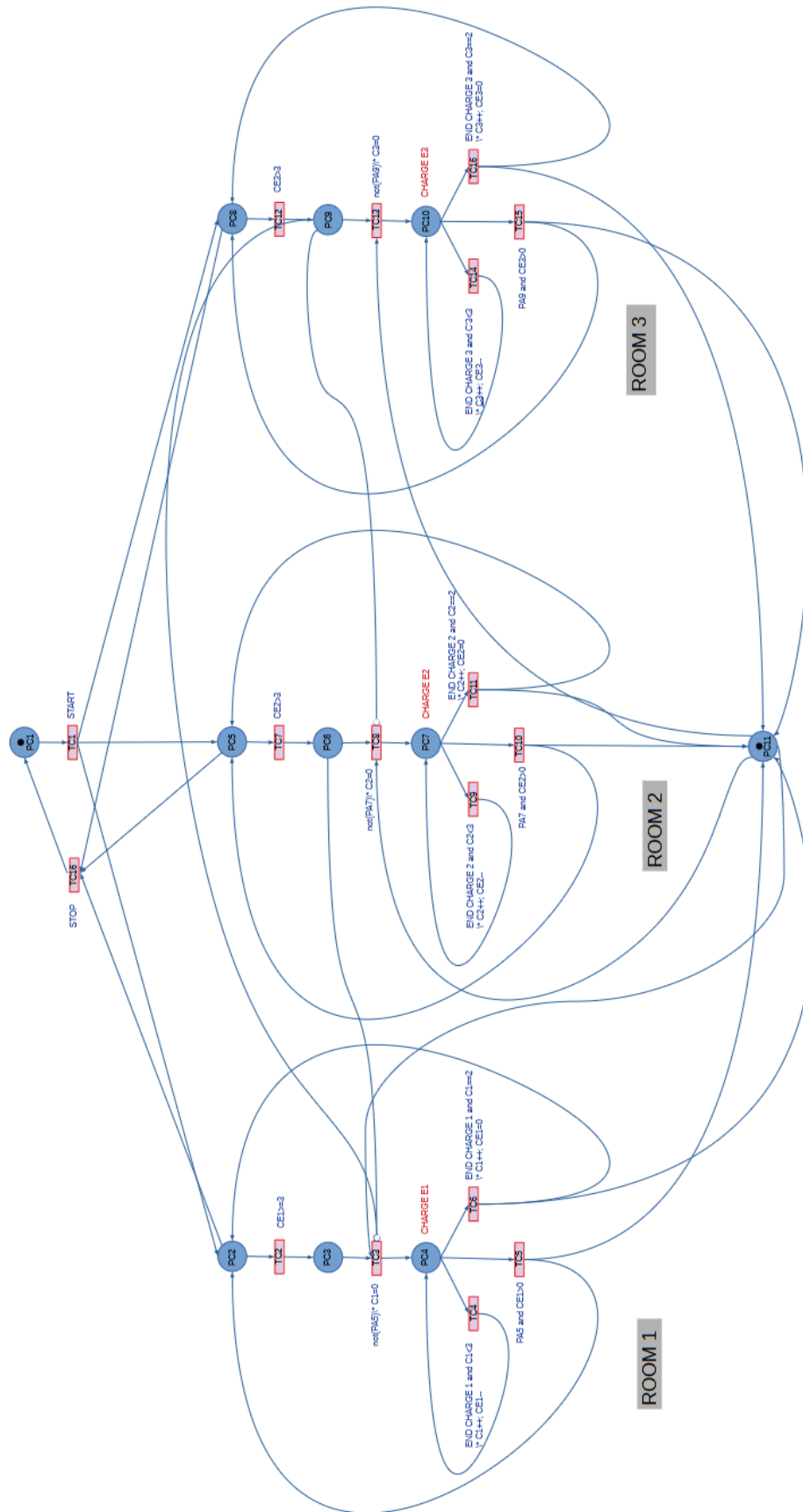


Figura 3-54 Red de Petri del sistema de mantenimiento automatizado

3.3.4.4 Programación y visualización

Desde la Figura 3-55 hasta la Figura 3-57 se muestra la programación del sistema automatizado de mantenimiento mediante lenguaje LD, específicamente enfocándose en la habitación 1 (*ROOM1*), aunque las demás habitaciones presentan una lógica similar. A través de esta programación, se observan claramente los lugares (*PC2*, *PC3*, *PC4*, etc.) que representan los estados posibles del sistema, ya sea en reposo o durante el proceso de carga de las baterías de los dispositivos correspondientes.

Cada lugar está condicionado por transiciones específicas (*TC1*, *TC2*, *TC3*, etc.), las cuales se activan mediante entradas y señales generadas por contadores que regulan el flujo y el avance del proceso. Concretamente, en cada habitación se distinguen tres lugares asociados a diferentes etapas del ciclo de mantenimiento, estableciendo así un control eficaz y secuencial de las operaciones. De esta forma, cuando se cumple la condición necesaria establecida por las entradas y contadores, el sistema pasa del estado de reposo al estado activo de carga y posteriormente retorna al reposo, garantizando una gestión ordenada y eficiente del mantenimiento preventivo en cada habitación.

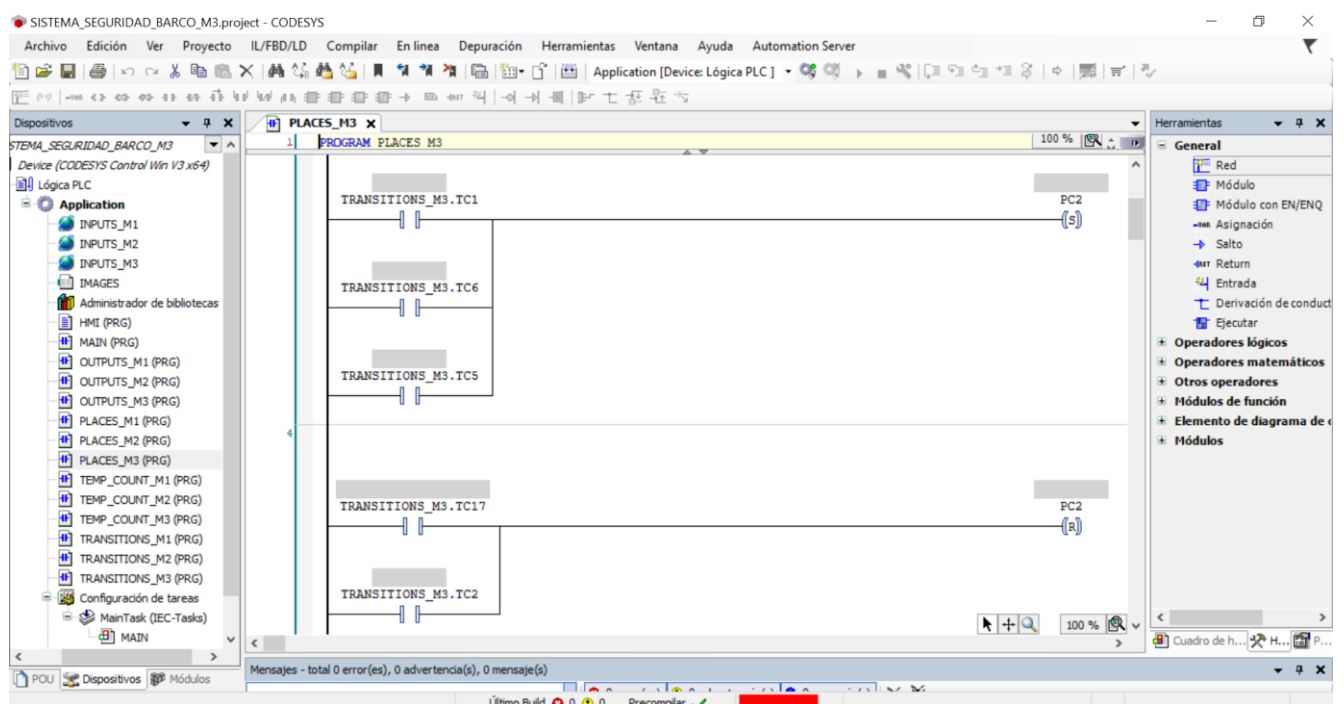


Figura 3-55 Programación Codesys del sistema de mantenimiento automatizado, PLACES_M3 (1)

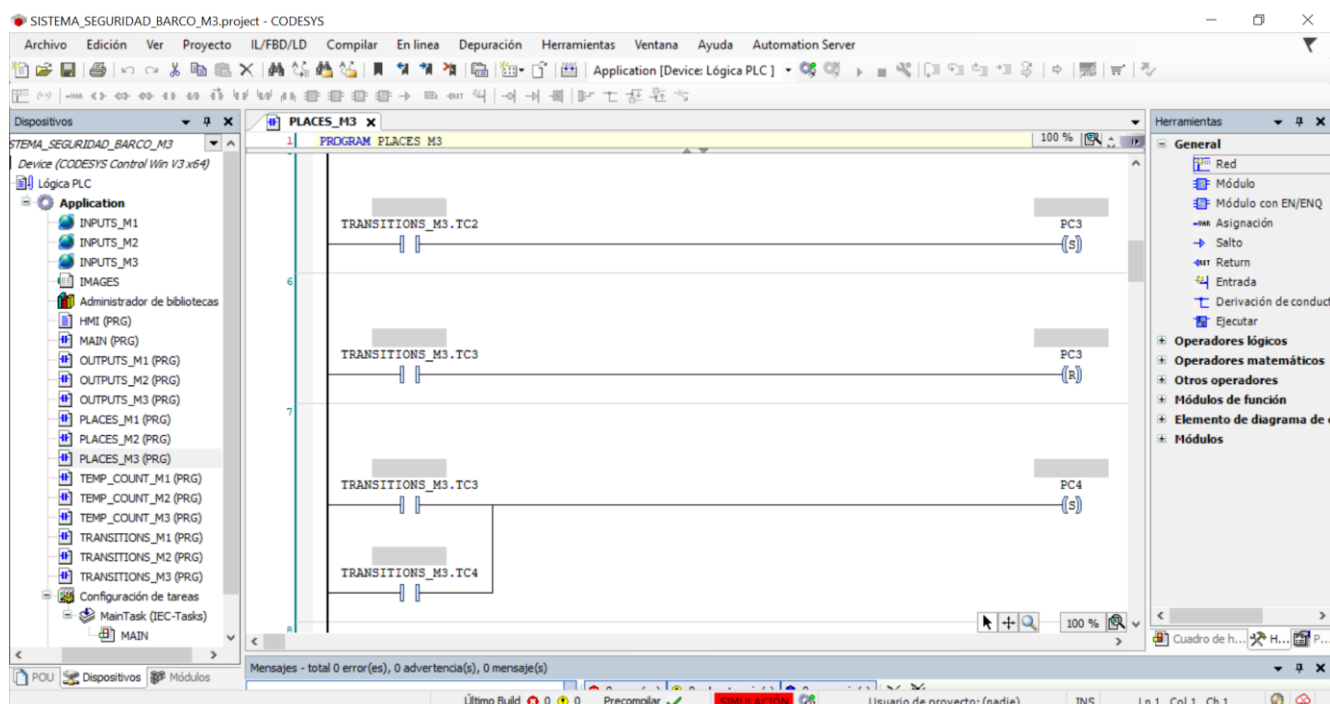


Figura 3-56 Programación Codesys del sistema de mantenimiento automatizado, PLACES_M3 (2)

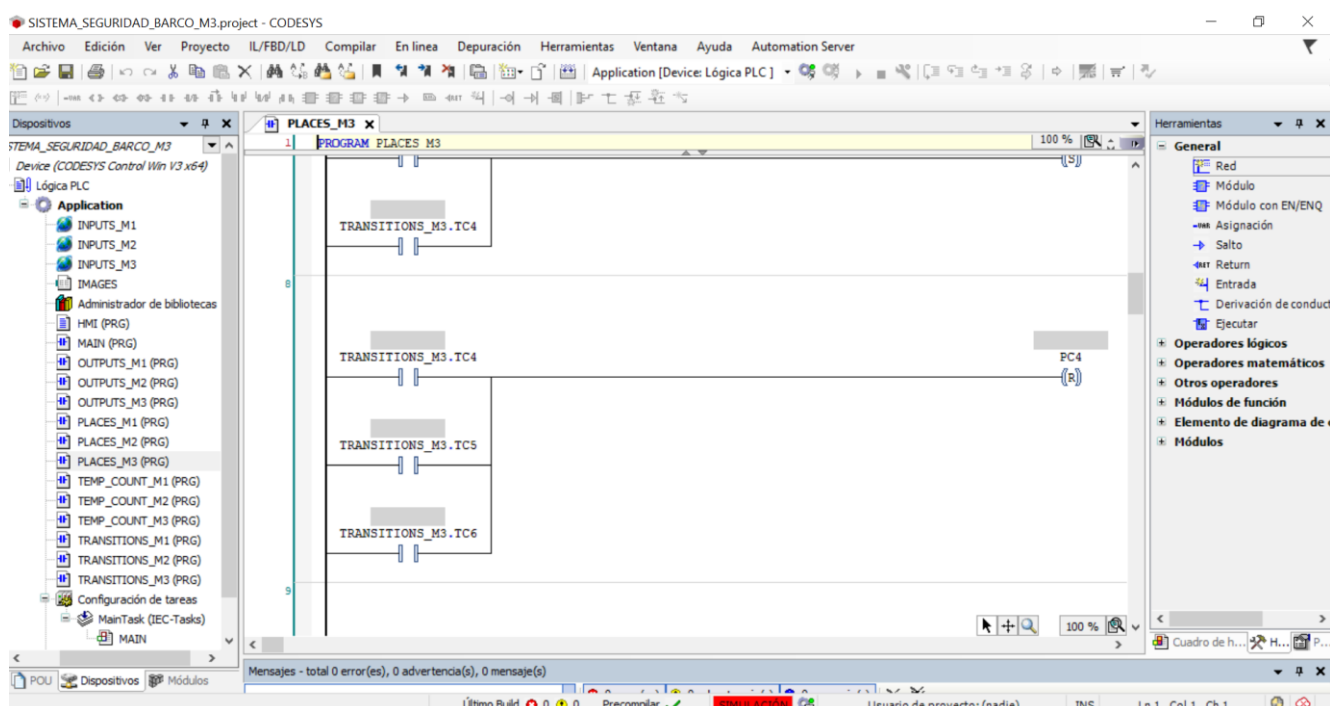


Figura 3-57 Programación Codesys del sistema de mantenimiento automatizado, PLACES_M3 (3)

En la Figura 3-58 y Figura 3-59 se muestra la interfaz visual diseñada para supervisar y controlar el sistema automatizado de mantenimiento de extintores. En ambas imágenes, se han integrado elementos visuales específicos, tales como indicadores luminosos que señalan de forma clara y directa cuándo los extintores están en proceso de carga.

En la Figura 3-58 se observa el sistema en estado desactivado, destacando que la mayoría de los indicadores permanecen apagados o sin resaltar, indicando que ninguna acción de mantenimiento está siendo ejecutada en este estado. Las luces indicadoras de carga permanecen inactivas, confirmando visualmente que ningún extintor está siendo sometido al proceso de recarga.

Por el contrario, en la Figura 3-59 el sistema se encuentra activado, lo que se evidencia mediante la iluminación y resaltado en color rojo de diversos lugares y variables clave (PA2, PA6, PB2, PB4, entre otras). Además, los indicadores luminosos correspondientes a la carga de los extintores están activados, mostrando claramente que el mantenimiento se encuentra en curso en algunas salas. Este estado permite visualizar fácilmente qué dispositivos específicos están siendo recargados, reforzando así el control y seguimiento del proceso en tiempo real.

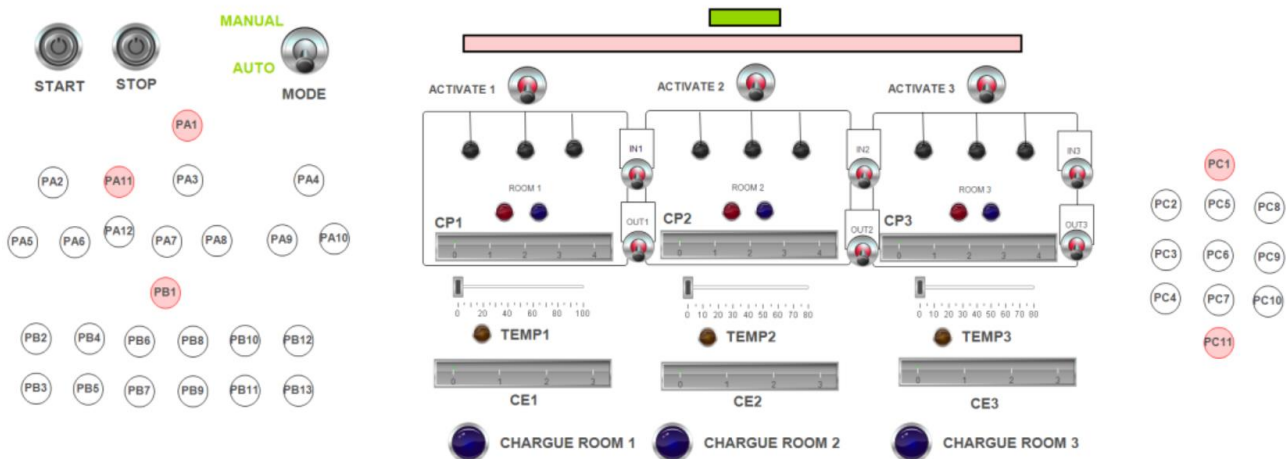


Figura 3-58 Representación visual del sistema de mantenimiento automatizado, desactivado

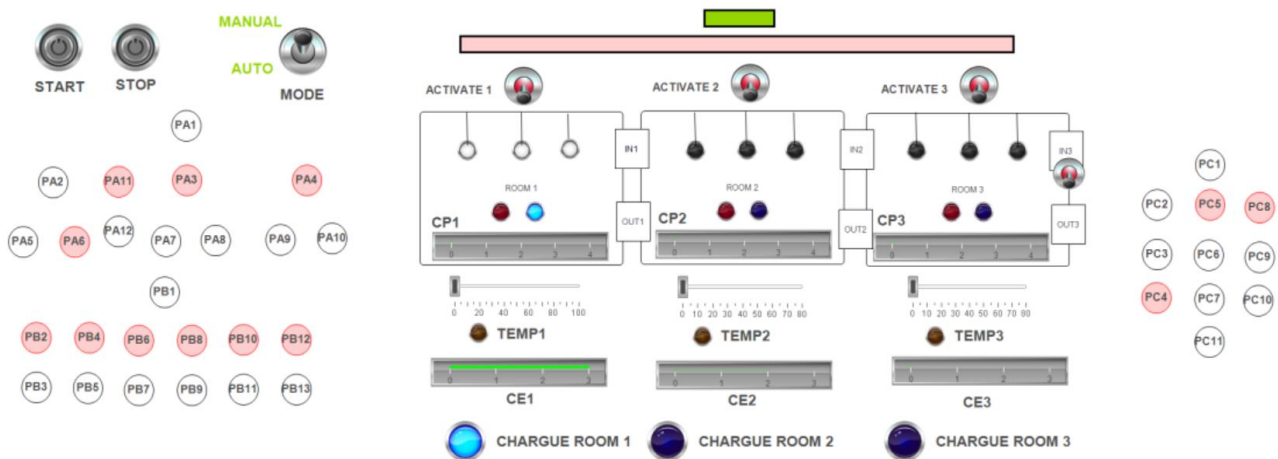


Figura 3-59 Representación visual del sistema de mantenimiento automatizado, desactivado

En resumen, la interfaz gráfica presentada en estas imágenes proporciona una visión integral y eficiente del estado operativo del sistema, facilitando la supervisión continua del proceso de mantenimiento automatizado.

Se incluye un enlace en el que se puede apreciar el funcionamiento del sistema en un video:

<https://drive.google.com/file/d/1kgMa6qgF09H5C182mGCKWQ7W9Dwepgjk/view?usp=sharing>

4 CONCLUSIONES Y LÍNEAS FUTURAS

El trabajo presentado demuestra que la aplicación conjunta de redes de Petri y el lenguaje LD permite desarrollar proyectos virtuales de automatización con éxito en el ámbito naval, mejorando significativamente la comprensión técnica y práctica de los estudiantes sin necesidad de recurrir a equipamiento físico. Se ha logrado alcanzar el objetivo principal, proporcionando una metodología eficaz para la enseñanza práctica y aplicada de automatización, aportando a los futuros oficiales navales competencias clave para afrontar desafíos tecnológicos.

Se han alcanzado satisfactoriamente todos los objetivos planteados al inicio de este Trabajo Fin de Grado. Se ha llevado a cabo el desarrollo estructurado y secuencial de tres proyectos independientes, cada uno compuesto por tres módulos específicos, diseñados de forma progresiva para incrementar su complejidad técnica y funcional.

En cada módulo se ha cumplido con éxito el objetivo de realizar una descripción técnica detallada y una adecuada representación lógica mediante el uso riguroso de Redes de Petri. Esto ha permitido modelar con precisión el comportamiento dinámico y la gestión de eventos concurrentes característicos de los sistemas propuestos.

Adicionalmente, se ha procedido a implementar estos modelos utilizando el lenguaje Ladder en el entorno de programación Codesys, asegurando una correcta traducción desde la representación teórica hasta su aplicación práctica y operativa en sistemas virtuales de automatización naval.

Finalmente, se ha logrado diseñar y montar una interfaz gráfica de usuario intuitiva y visualmente clara, capaz de reflejar fielmente el comportamiento y las secuencias previamente modeladas en cada uno de los módulos desarrollados. Esta interfaz no solo facilita la visualización en tiempo real del funcionamiento del sistema, sino que también proporciona una herramienta eficaz para la validación y verificación de los procesos automatizados.

De este modo, se confirma la validez de la metodología propuesta, demostrando que la integración de Redes de Petri, lenguaje Ladder y una interfaz gráfica de usuario contribuye significativamente a mejorar el aprendizaje práctico de la automatización en el ámbito naval, optimizando tiempos, minimizando errores operativos y ofreciendo una sólida base técnica para futuros desarrollos e investigaciones.

La implementación de los tres proyectos descritos: control de propulsión híbrida, automatización del muelle de carga naval, y seguridad operativa en buques, evidencia ventajas notables en eficiencia operativa, reducción de errores y tiempos de respuesta ante emergencias, reforzando la preparación técnica del alumnado. Este enfoque también resalta la versatilidad y la utilidad del modelado con redes

de Petri para representar procesos dinámicos complejos y facilitar su posterior programación en lenguaje LD.

Además, se ha evidenciado que la integración de herramientas virtuales y métodos de simulación facilita el aprendizaje práctico y optimiza significativamente los tiempos de diseño y desarrollo, fortaleciendo las habilidades necesarias para la automatización industrial aplicada al sector marítimo.

Para futuras investigaciones se propone explorar el potencial de integrar tecnologías avanzadas como Inteligencia Artificial y *Machine Learning* con las redes de Petri y lenguaje LD, con el fin de desarrollar sistemas híbridos inteligentes capaces de realizar análisis predictivos más complejos y mejorar aún más la eficacia operativa y seguridad en aplicaciones navales.

Otra línea de investigación sería la extensión y aplicación de estos métodos a otros ámbitos industriales y logísticos distintos del naval, probando así la flexibilidad y adaptabilidad de las metodologías propuestas. También se recomienda profundizar en el desarrollo de plataformas virtuales interactivas más completas, que permitan simulaciones aún más realistas y enriquecidas con interfaces gráficos avanzados, potenciando así un aprendizaje más intuitivo e inmersivo para los alumnos.

Finalmente, es fundamental contemplar aspectos éticos, sociales y ambientales asociados con la automatización, considerando estrategias para mitigar posibles impactos negativos como la reducción del empleo manual y el manejo responsable de los residuos tecnológicos derivados de estos procesos de automatización.

5 BIBLIOGRAFÍA

- [1] «Guia-Control-procesos-automatizacion.pdf». Accedido: 3 de marzo de 2025. [En línea]. Disponible en: <https://tecnologicosucre.edu.ec/web/wp-content/uploads/2024/09/Guia-Control-procesos-automatizacion.pdf>
- [2] D. Jensen, «DIAGRAMAS DE BLOQUES: ALGUNOS CONCEPTOS», 2004.
- [3] «intro_iec_61131_3_spanish.pdf». Accedido: 11 de febrero de 2025. [En línea]. Disponible en: https://www.plcopen.org/sites/default/files/downloads/intro_iec_61131_3_spanish.pdf
- [4] E. Kofman, «Modelado y Simulación de Sistemas Dinámicos: Métodos, Algoritmos y Herramientas - Sistemas Continuos», n.o 1.
- [5] J. E. Hopcroft, «Introduccion a la teoria de automatas lenguajes y computacion».
- [6] «eventos.pdf». Accedido: 11 de febrero de 2025. [En línea]. Disponible en: https://www.uaeh.edu.mx/docencia/P_Presentaciones/huejutla/sistemas/lenguajes_automatas/eventos.pdf
- [7] «Automatas y Lenguajes - Ramon Brena», studylib.es. Accedido: 11 de febrero de 2025. [En línea]. Disponible en: <https://studylib.es/doc/9124168/automatas-y-lenguajes---ramon-brena>
- [8] O. Vilca, «HERRAMIENTAS PARA ABORADAR AUTÓMATAS FINITOS», en Ciência da Computação: avanços e tendências em pesquisa, 1.a ed., Editora Científica Digital, 2023, pp. 69-80. doi: 10.37885/230914216.
- [9] E. Adrián, «08 Automatas finitos».
- [10] «BOOK-2018-014.pdf». Accedido: 12 de febrero de 2025. [En línea]. Disponible en: <https://zagan.unizar.es/record/70370/files/BOOK-2018-014.pdf>
- [11] «Vista de Redes de Petri: Modelado e implementación de algoritmos para autómatas programables». Accedido: 24 de marzo de 2025. [En línea]. Disponible en: https://revistas.tec.ac.cr/index.php/tec_marcha/article/view/231/228
- [12] S. Segura, «Tema 8: Técnicas de diagramas de transición de estados».
- [13] I. L. Francucci, «Tutorial: Diagramas de Estado (Statecharts)».
- [14] E. Vitoria, «5. DIAGRAMAS DE ESTADO».
- [15] «diagrmas-de-actividad-y-estado.pdf». Accedido: 15 de febrero de 2025. [En línea]. Disponible en: <https://diazsasite.wordpress.com/wp-content/uploads/2017/01/diagrmas-de-actividad-y-estado.pdf>

- [16] M. R. Nigro y J. A. Castro, «Generación Automática de Código a Partir de Máquinas de Estado Finito», vol. 14, n.o 4, 2011.
- [17] «document.pdf». Accedido: 1 de marzo de 2025. [En línea]. Disponible en: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=d3dd832ef0b8c9a3ad2adc30abf6a49107a10ab3>
- [18] A. K. Samantaray y B. O. Bouamama, Model-based Process Supervision: A Bond Graph Approach. Springer Science & Business Media, 2008.
- [19] J. F. Broenink, «Introduction to Physical Systems Modelling with Bond Graphs», 1999.
- [20] «05200512.pdf». Accedido: 12 de febrero de 2025. [En línea]. Disponible en: <https://oa.upm.es/89/1/05200512.pdf>
- [21] «powerdynamo.pdf». Accedido: 12 de febrero de 2025. [En línea]. Disponible en: <https://www.fceia.unr.edu.ar/~kofman/seminario/powerdynamo.pdf>
- [22] J. A. L. Ruiz, «SIMULACIÓN DE MECANISMOS».
- [23] H. G. R y Á. G. Muñoz, «Centro de Investigación de la Universidad Distrital Francisco José de Caldas», Rev. Vínculos, vol. 10, n.o 2, Art. n.o 2, dic. 2013, doi: 10.14483/2322939X.6523.
- [24] H. D. Páez-Logreira, R. Zamora-Musa, y J. Bohórquez-Pérez, «Programación de Controladores Lógicos (PLC) mediante Ladder y Lenguaje de Control Estructurado (SCL) en MATLAB», Rev. Fac. Ing., vol. 24, n.o 39, pp. 109-119, may 2015.
- [25] J. A. S. P. Miralles, «GHENeSys, uma rede unificada e de alto nível», 2012.
- [26] «tese_juancarlosvidal_20210618112106766_e90214140b.pdf». Accedido: 9 de marzo de 2025. [En línea]. Disponible en: https://citius.gal/static/0b837d06e12163672df885ce952844f8/tese_juancarlosvidal_20210618112106766_e90214140b.pdf
- [27] A. de, «Facultad de Informática Universidad Nacional de La Plata».
- [28] J. A. A. Domínguez, S. Gonnet, y M. M. Vegetti, «Asset Administration Shell para representación de procedimientos de procesos batch», JAIIO Jorn. Argent. Informática, vol. 9, n.o 15, Art. n.o 15, jul. 2023.
- [29] J. Alvarado Domínguez, M. Vegetti, y S. Gonnet, Formalización del lenguaje gráfico Sequential Function Chart para definir un submodelo de Asset Administration Shell en la industria de procesos. 2023.
- [30] «AT07-003_20.pdf». Accedido: 10 de marzo de 2025. [En línea]. Disponible en: http://dspace.aepro.com/xmlui/bitstream/handle/123456789/2549/AT07-003_20.pdf?sequence=5&isAllowed=y
- [31] J. V. Costa y J. G. Prado, «Prototipo automatizado para el llenado y tapado de botellas de plástico», Ind. Data, vol. 20, n.o 1, pp. 125-130, 2017.
- [32] «TFM-G1800.pdf». Accedido: 10 de marzo de 2025. [En línea]. Disponible en: <https://uvadoc.uva.es/bitstream/handle/10324/63047/TFM-G1800.pdf?sequence=1&isAllowed=y>

ANEXO I: DICCIONARIO DE SIGLAS, ACRÓNIMOS Y ABREVIATURAS

En la siguiente tabla se recoge el significado de las siglas, acrónimos y abreviaturas empleadas a lo largo del documento. Esto facilita la comprensión y lectura rápida del texto.

SIGLAS/ABREVIATURA	SIGNIFICADO
FBD	<i>Function Block Diagram</i>
PLC	<i>Programmable Logic Controller</i>
AFD	Autómata Finito Determinista
AFND	Autómata Finito No Determinista
C	Capacitancia
I	Inercia
R	Resistencia
Se	Source Effort
TF	Transformador
GY	Girador
RLC	Resistencia, Inductancia y Capacitancia
Ce	Constante Elástica
Ra	Amortiguación
LD	Lenguaje Ladder
HLPN	High Level Petri Net
SFC	Sequential Function Chart
TIC	Tecnologías de la Información y la Comunicación
IA	Inteligencia Artificial
RPA	Robot Process Automation
CAD	Computer Aided Design
SQL	Structured Query Language
API	Application Programming Interface
HMI	Human-Machine Interface

Tabla 0-1 Siglas, acrónimos y abreviaturas

ANEXO II: IMPLICACIONES SOCIALES, ECONÓMICAS, Y AMBIENTALES

El desarrollo de sistemas automatizados basados en redes de Petri y lenguaje LD conlleva una serie de repercusiones en distintos ámbitos que deben ser consideradas para evaluar su impacto en la sociedad, la economía y el medioambiente. La automatización de procesos en sectores como la industria naval y la manufactura transforma el mercado laboral, reduce costos operativos y optimiza la eficiencia energética, generando tanto oportunidades como desafíos que requieren un análisis integral.

Desde el punto de vista social, la automatización modifica las dinámicas laborales al sustituir tareas manuales y repetitivas por sistemas controlados de manera automática. Esto puede generar preocupación respecto a la pérdida de empleos en ciertos sectores, aunque al mismo tiempo fomenta la creación de nuevas oportunidades en el ámbito de la programación, el mantenimiento y la supervisión de sistemas tecnológicos. Además, contribuye a mejorar la seguridad en entornos de trabajo peligrosos, reduciendo la exposición de los trabajadores a situaciones de riesgo. La necesidad de formación especializada se convierte en un aspecto clave, impulsando la capacitación y la adaptación de los profesionales a un entorno cada vez más digitalizado.

En el ámbito económico, la automatización permite una notable reducción de costos operativos al minimizar errores humanos, optimizar el uso de materiales y mejorar la productividad. Al operar de manera continua, los sistemas automatizados incrementan la eficiencia en la producción y reducen los tiempos de inactividad, lo que se traduce en un retorno de inversión favorable a largo plazo. En el sector naval, la aplicación de estos sistemas mejora la gestión de operaciones y el consumo energético de las embarcaciones, permitiendo una mayor rentabilidad y eficiencia logística. A pesar de la inversión inicial requerida para implementar tecnologías de automatización, los beneficios a largo plazo compensan los costos, proporcionando una ventaja competitiva a las organizaciones que los adoptan.

El impacto ambiental de la automatización depende del tipo de tecnología empleada y su eficiencia en la gestión de recursos. La optimización de procesos contribuye a una reducción en el consumo de energía, disminuyendo la huella de carbono de las operaciones industriales y navales. Asimismo, la automatización facilita un uso más eficiente de materiales y minimiza la generación de residuos, reduciendo el impacto ambiental de la producción. En el caso del sector naval, la implementación de sistemas de propulsión híbrida automatizados permite una mejor regulación del consumo de combustibles y de las emisiones contaminantes, favoreciendo la sostenibilidad del transporte marítimo. Sin embargo, el incremento en la demanda de dispositivos electrónicos y sistemas automatizados también plantea el desafío de gestionar adecuadamente los residuos tecnológicos para evitar consecuencias negativas en el medioambiente.

En conclusión, la automatización mediante redes de Petri y lenguaje LD ofrece múltiples ventajas en términos de eficiencia, seguridad y sostenibilidad, pero su implementación debe ser acompañada por estrategias que mitiguen los efectos adversos en el ámbito social y ambiental. La adaptación del mercado laboral, la optimización de los recursos y la integración de soluciones sostenibles serán factores determinantes para que la automatización contribuya de manera equilibrada al desarrollo económico e industrial sin comprometer el bienestar social y ecológico.

ANEXO III: REFLEXIONES ÉTICAS Y SOCIALES

La automatización de procesos mediante redes de Petri y lenguaje LD plantea diversas reflexiones éticas y sociales que deben ser consideradas en el desarrollo y aplicación de estas tecnologías. La creciente integración de sistemas automatizados en la industria, el transporte y otros sectores genera impactos que van más allá de la eficiencia operativa, afectando el mercado laboral, la responsabilidad en la toma de decisiones y la seguridad de las personas.

Uno de los principales aspectos éticos en la automatización es la sustitución de la mano de obra humana por sistemas tecnológicos. Si bien estos avances permiten aumentar la eficiencia y reducir costos, también generan preocupación sobre la posible pérdida de empleos en sectores donde las tareas manuales están siendo reemplazadas por sistemas automáticos. La transformación del mercado laboral exige una mayor capacitación y reconversión profesional, lo que plantea el reto de garantizar que los trabajadores puedan adaptarse a las nuevas demandas tecnológicas sin quedar desplazados. Este cambio estructural debe abordarse con programas de formación y políticas que fomenten la empleabilidad en un entorno cada vez más digitalizado.

Otro aspecto relevante es la responsabilidad en la toma de decisiones automatizadas. Los sistemas programados mediante redes de Petri y lenguaje LD operan siguiendo instrucciones predefinidas, lo que plantea preguntas sobre quién es responsable en caso de fallos o consecuencias inesperadas. En entornos donde la seguridad es crítica, como la industria naval o la automatización de infraestructuras, la implementación de estos sistemas debe ir acompañada de protocolos de supervisión y mecanismos de intervención humana que garanticen la fiabilidad y seguridad de los procesos. Además, es fundamental establecer marcos normativos que regulen el uso de la automatización, asegurando que la responsabilidad de las decisiones no recaiga únicamente en la tecnología, sino también en las personas y organizaciones que la desarrollan y gestionan.

Desde una perspectiva social, la automatización puede contribuir a la mejora de la calidad de vida al reducir la exposición de los trabajadores a entornos peligrosos, minimizar la carga laboral en tareas repetitivas y permitir una mayor precisión en la ejecución de procesos críticos. Sin embargo, también puede generar desigualdades si su implementación no se lleva a cabo de manera equitativa. Es necesario garantizar que los beneficios de la automatización sean accesibles para diferentes sectores de la sociedad y no se conviertan en un factor que amplíe la brecha tecnológica entre países, empresas y trabajadores con diferentes niveles de formación.

Por otro lado, el impacto ético de la automatización también se extiende a la transparencia y equidad en la implementación de estos sistemas. La programación de sistemas automatizados debe realizarse con criterios de equidad, evitando sesgos que puedan generar discriminación o favorecer intereses particulares. La transparencia en la toma de decisiones automatizadas es clave para garantizar la confianza en la tecnología y su aceptación por parte de la sociedad.

En conclusión, la automatización mediante redes de Petri y lenguaje LD ofrece numerosas oportunidades para mejorar la eficiencia y la seguridad en distintos sectores, pero también plantea desafíos éticos y sociales que no pueden ser ignorados. La adaptación del mercado laboral, la regulación de la responsabilidad en la toma de decisiones, la equidad en la implementación y la transparencia en el desarrollo de estos sistemas son aspectos clave para garantizar un uso ético y socialmente responsable de la automatización. La tecnología debe estar al servicio del bienestar humano, asegurando que su desarrollo e implementación contribuyan a una sociedad más justa y equitativa.

ANEXO IV: CÓDIGO DEL PROYECTO 1, 2 Y 3

En este anexo se incluyen enlaces a documentos PDF con el código completo de cada módulo. Se optó por este formato debido a la considerable extensión del contenido. Esto facilita la revisión y manejo individualizado de cada módulo.

PROYECTO	MÓDULO	NOMBRE	ENLACE
1	1	<i>Gestión y control del sistema de propulsión híbrida</i>	https://drive.google.com/file/d/1T5tiBc5-7xoqh4lUzA40-a0BZsorYG18/view?usp=drive_link
1	2	<i>Gestión del mantenimiento preventivo del sistema de propulsión híbrida</i>	https://drive.google.com/file/d/1Mr9Z-IWe-2T7U6Cp2v_yyABKS4hTSDAV/view?usp=drive_link
1	3	Gestión de alarmas y fallos del sistema de propulsión híbrida	https://drive.google.com/file/d/1wylW21lYA_mPYhATEuml8bNW6GiKrhE5/view?usp=drive_link
2	1	Lancha con plataforma de carga automática	https://drive.google.com/file/d/1l4o-ZHc2xTD1RR0qmWt9dtaAlsFqkl_B/view?usp=drive_link
2	2	Gestión de la carga y almacenamiento en el muelle automatizado	https://drive.google.com/file/d/1GE3LkG5coGlp-t1PboeqP4sols8t6nZ0/view?usp=drive_link
2	3	Gestión de energía y recarga de baterías	https://drive.google.com/file/d/1Tjplhcj9Ww0hxoNzVG5kqt1LBmR6ARRj/view?usp=drive_link
3	1	Sistema de detección y extinción de incendios	https://drive.google.com/file/d/1v2o8Zcx2OmDo1LOAnX35QPuuQw78RfPz/view?usp=drive_link
3	2	Sistema de conteo de personas y control de puertas	https://drive.google.com/file/d/1YWacF3ocDib6dorC8S-tIFVaLHdbLVJc/view?usp=drive_link
3	3	Sistema de mantenimiento automatizado	https://drive.google.com/file/d/1GimD90pYhtmoxDJdJCXp71aU7xZcvhh8/view?usp=drive_link

Tabla 0-1 Códigos de los distintos proyectos