Sistema de desarrollo para la programación de PLC usando Redes de Petri en el modelo de Control

Pedro Mateo Hinojosa Miranda Jorge Antonio Nava Amador

Abstract

El artículo describe el diseño y funcionamiento de un sistema de desarrollo para la programación del Controlador Lógico Programable (PLC), mediante el modelado conceptual del proceso, por Redes de Petri Interpretadas para Control (RPIC), las cuales son una extensión de las Redes de Petri (RP), cuyas principales características incluyen la representación de señales de entrada y salida, arcos habilitadores y temporización. Este sistema de desarrollo provee una fácil alternativa para el diseño de programas de control de Sistemas a Eventos Discretos (DES), debido al uso de RP en el modelado de la lógica de control, aspecto que simplifica la descripción del comportamiento dinámico del sistema. Por otro lado, al ser las RP una herramienta de modelado cuya representación gráfica es sencilla, se tiene un herramienta con poca demanda de aprendizaje y alta comprensión respecto a otras técnicas de desarrollo de programas para PLC.

El sistema de desarrollo se implementó, complementando una plataforma ya existente, usando Java como lenguaje de programación; Se proporciona una interfaz gráfica para el modelado del proceso mediante RPIC, una descripción matemática del modelo mediante arreglos matriciales y una interfaz para la visualización del programa de control generado, programa que consiste en Diagramas Ladder (LD).

Keywords – Redes de Petri; Automatas Programables; PLC; Sistemas a Eventos Discretos.

I. Introducción

El Controlador Lógico Programable - PLC, es un dispositivo ampliamente empleado en la automatización de procesos industriales, y aunque ha evolucionado constantemente desde su aparición, los lenguajes de programación no han cambiado; lenguajes definidos por el estándar IEC 61131 – 3, que define 5 lenguajes: Diagramas Ladder, Lista de Instrucciones, Diagramas de Funciones, Texto Estructurado y Grafcet. De los cuales los Diagramas Ladder son los más utilizados por su característica gráfica que facilita el desarrollo de sistemas de control simples, pero se ven limitados cuando se trabajan con sistemas más complejos.

En la actualidad los procesos exigen que se gobiernen distintas actividades y tareas, que además deben realizar acciones coordinadas, por lo tanto muestran elevada complejidad a la hora de pretenderse abordar con LD, puesto que se requiere la sincronización y la ejecución de tareas paralelas, las cuales son complicados de abordar con este lenguaje.

Por otro lado las RP constituyen un excelente método para el modelado e implementación de sistemas a eventos discretos, puesto que describen de manera clara y detallada, el comportamiento dinámico del sistema, además constituye una excelente herramienta para el modelado procesos tales como el paralelismo y sincronización, los cuales se encuentran presentes en gran número de procesos industriales. Sin embargo, teniendo en cuenta que la implementación de sistemas de control está condicionada a la interacción del sistema con elementos externos, se hace evidente la necesidad de extender las funcionalidades de las Redes de Petri, puesto que si bien estas describen claramente la dinámica de un sistema, no proveen información del momento en que el sistema evoluciona de un estado a otro, y tampoco proveen herramientas para la representación de los elementos de periferia. Es así que para el desarrollo del sistema se hace uso de RPIC.

Bajo esa lógica se propone un Sistema de Desarrollo que a partir del modelado del proceso con RPIC, genere el Diagrama Ladder para incorporarlo como lógica de control en un autómata, simplificando con ello el dificultoso proceso de programar directamente con diagramas Ladder.

II. Planteamiento del Problema

El constante crecimiento de la complejidad de los sistemas automatización, hace que el desarrollo de programas de control con Diagramas Ladder se torne cada vez más complicada. Además la falta de métodos formales para el diseño de controladores mediante LD se realiza con métodos heurísticos que se apoyan en gran medida en las habilidades y pericia del programador. Como resultado no se obtienen soluciones rápidas y los programas generados normalmente no son fáciles de comprender.

Por otro lado se tiene que el lenguaje gráfico simbólico conocido como Diagramas Ladder, es el más utilizado en el desarrollo de programas para autómatas; este método sobresale por su simplicidad y similitud a los diagramas de circuitos eléctricos de contactos, sin embargo esta simplicidad no se ve reflejada cuando se desarrollan sistemas de control complejos que implican ciertos eventos específicos tales como

la concurrencia, el sincronismo y la secuencialidad, que aunque no son imposibles de implementar en Diagramas Ladder, resulta bastante complicado para el diseñador concebir y ajustar un diseño de forma ágil y sencilla de este tipo de controladores.

Adicionalmente, el diseño de Diagramas Ladder al no estar apoyado en métodos formales, se realiza mediante acondicionamiento y ajuste de configuraciones básicas hasta alcanzar el desempeño deseado, lo cual produce:

- Soluciones altamente dependientes del razonamiento lógico del programador.
- Alto tiempo de desarrollo y ajuste del controlador.
- Dificultad en el mantenimiento y reconfiguración del controlador.

Por otro lado, si bien ciertas marcas de autómatas permiten emplear herramientas gráficas más estructuradas para la implementación de los sistemas secuenciales, como el Grafcet, rara vez van más allá, y no se encuentran aplicaciones para trabajar directamente con otras herramientas de más alto nivel, como son las RP, y aunque existen diversas aplicaciones informáticas para el diseño, edición y análisis de RP que especifican sin ambigüedades las propiedades estructurales y el comportamiento dinámico del sistema que se desea controlar y están provistas de herramientas de análisis mediante la validación de sus propiedades, lo que garantiza la robustez del diseño, estas aplicaciones no cuentan con módulos que permitan la relación directa de los diseños elaborados a un lenguaje normalizado de programación de PLC.

III. Alcance

- Desarrollar una interfaz gráfica con elementos suficientes para el modelado de sistemas a eventos discretos mediante RPIC.
- Desarrollar un módulo para la representación matemática de las RP, la cual describe la dinámica del sistema a eventos discretos.
- Desarrollar un módulo para la generación automática de Diagramas Ladder a partir del modelo del sistema realizado con RPIC

Dada la extensión del alcance propuesto, se pretendió realizar un prototipo del sistema de desarrollo, que contemple las funcionalidades mínimas que permitan la generación de programas de control a partir del modelado del proceso con RPIC.

IV. Definiciones Básicas

En esta sección se presentan las definiciones básicas de las RP y RPIC, las cuales han sido utilizadas en el desarrollo del proyecto, y que para entender el trabajo es necesario hacer referencias a ellas.

A. Redes de Petri

Se sabe que la teoría inicial de las RP fue postulada por el alemán Carl Adam Petri, la cual consiste en una herramienta gráfica para la descripción formal de sistemas, además permite la representación matemática de los mismos y esta provista de métodos para el análisis de las propiedades dinámicas y estructurales del sistema especificado. La principal aplicación de las redes de Petri es el modelado y análisis de sistemas.

1) Representación Gráfica

La representación gráfica de las RP es bastante útil para la comprensión del proceso que está siendo especificado. Una RP está formada por dos tipos de elementos básicos, los cuales son las transiciones que son elementos activos y por otro lado tenemos, los lugares, que corresponden a elementos pasivos. Los lugares representan a las variables de estado que conforman un sistema, y las transiciones representan los eventos y acciones que son ejecutadas por el sistema. Para la ejecución de una acción se deben cumplir con ciertos requisitos o precondiciones, las cuales están asociadas a los estados del sistema; por tanto, se observa que existe una relación entre los lugares y las transiciones que posibilitan la ejecución de la acción. Asimismo, después de la ejecución de una acción la información de algunos lugares también cambiara debido a que corresponden a post condiciones de la acción. Gráficamente los lugares están representados por círculos y las transiciones por líneas o figuras rectangulares.

2) Representación Matricial

La representación matricial de las RP posibilita el análisis formal de las propiedades estructurales y dinámicas del sistema. Según este punto de vista la RP está formado por un conjunto de lugares P, un conjunto de Transiciones T, una matriz de entrada a las transiciones I y una matriz de salida de las transiciones O, los cuales reciben

el nombre de matriz de incidencia previa y matriz de incidencia posterior respectivamente. Ambas matrices son arreglos bidimensionales en los que las columnas representan las transiciones, las filas los lugares y las celdas la conexión entre ambas. Las matrices de entrada y salida representan las precondiciones y post condiciones del sistema respectivamente, en ellas se observa las condiciones necesarias para realizar el disparo de una transición.

3) Red de Petri Marcada

La marcación es un concepto fundamental en las RP, la cual consiste en una función que asigna a cada lugar un valor entero no negativo (marca). Las marcas son informaciones atribuidas a cada lugar, y el número y distribución de marcas representa la marcación (estado) de la red en un determinado momento. El marcado gráficamente se representa mediante puntos en los lugares y matricialmente son representados por un vector cuyo número de componentes corresponde al número de lugares de la red. Asimismo es importante señalar que una Red de Petri Marcada debe contar con una marcación inicial (M0), para que a partir de este se puedan encontrar nuevos marcados con cada uno de los disparos de las transiciones.

La presencia de marcas en un determinado lugar se pude interpretar como la disponibilidad de recursos de un determinado tipo. Vale la pena resaltar que el marcado de la Red de Petri representa el estado en que esta se encuentra.

4) Dinámica de las Redes de Petri

La dinámica de transición de estados en las RP viene representada por el cambio en la posición de las marcas de la red, este cambio hace que el marcado de la red cambie consecuentemente también cambiará el estado de la misma. El marcado de la red cambia con el disparo de las transiciones, asimismo una transición puede disparar cuando esta se encuentra sensibilizada vale decir que en los lugares de entrada a esta transición deben existir igual o mayor número de marcas que el valor correspondiente al peso de los arcos que unen estos lugares con la transición. Es importante señalar que la marcación habilita una transición para disparar pero no la obliga.

El disparo de una transición retira marcas de los lugares origen y deposita marcas en los lugares de destino, lo cual modifica la distribución de marcas de la red y consecuentemente el estado en que se encuentra.

B. Redes de Petri Interpretadas para Control

Las redes de Petri Autónomas son bastante útiles para modelar la dinámica de los DES, no obstante en este tipo redes no se definen elementos para interactuar con elementos de periferia, ni incluyen conceptos temporales, por tanto no son óptimas para el modelado adecuado en "control" cuando se tienen sistemas que se encuentran condicionados o por el tiempo o por la ocurrencia de un evento. Para poder modelar sistemas controlados por autómatas, las RP han sido extendidas, para ser capaces de procesar eventos lógicos que condicionan el disparo de las transiciones, así como también para definir atributos temporales y funciones asociadas a los lugares, lo cual permite interactuar con los elementos periféricos al proceso. Este tipo de Redes de Petri constituyen una de las extensiones de las RP y son denominadas en general como Redes de Petri Interpretadas, de las cuales se tienen una amplia variedad. Estas redes se caracterizan por que los eventos y las condiciones se definen sobre variables booleanas.

Las RPIC son un tipo particular de Redes de Petri Interpretadas, las cuales fueron propuestas por David & Alla. Estas Redes han sido extendidas para interactuar con información externa al sistema los cuales son captados por sensores y se ejecutan por elementos actuadores. La información con la que el sistema puede interactuar debe ser de naturaleza booleana, lo cual representa a un conjunto de eventos que son condiciones asociadas a las transiciones, las cuales condicionan la evolución de la red. Por otro lado, esta red también es capaz de enviar información a los dispositivos responsables de ejecutar las operaciones en el medio externo al sistema. También están diseñadas para poder operar, procesar e interpretar señales sin duración temporal tal como los cambios de nivel, los cuales están presentes por lo general en muchos sistemas de control diseñados.

Las RPIC, poseen las siguientes características:

- Sincronizadas por eventos externos
- Segura
- Determinística

En este tipo de redes el disparo de las transiciones no está condicionado únicamente a las marcas que contienen los lugares que forman parte del conjunto de entradas a la red, sino también se deben cumplir con todas las condiciones asociadas a las transiciones las cuales representan eventos externos al sistema obtenidas por medio de sensores, estas condiciones pueden ser de nivel o de flanco.

Como las RPIC son seguras, el número máximo de marcas asignados a los lugares no puede ser más de uno, por otro lado se tiene que, al ser una red determinística, esta debe estar libre de conflictos los cuales pueden resueltos mediante: La asignación de condiciones mutuamente excluyentes entre las transiciones en conflicto, mediante la asignación de prioridades en el disparo de las transiciones o mediante el uso de arcos inhibidores en las transiciones en conflicto.

V. Metodología de Conversión

Diversas metodologías han sido propuestas para la generación de código para autómatas programables a partir del modelado conceptual de un proceso mediante RP, no

obstante se hizo uso de la metodología propuesta por Moreira, Botelho y Basilio, la cual hace uso de RPIC para la generación de Diagramas Ladder, método que consiste en cinco módulos para garantizar un adecuado desempeño del controlador.

A. Módulo de Inicialización

Se hace uso del vector de marcado inicial M0 para formar la primera línea del LD, que consiste en las condiciones iniciales asociadas al controlador. Este módulo consiste en un contacto normalmente cerrado (NC) asociada con una variable binaria interna del PLC por tanto en este ciclo se energizarán todas las bobinas asociadas con los lugares que inicialmente poseen una marca. Una vez concluido el primer ciclo de barrido del autómata el contacto normalmente cerrado será abierto, lo cual producirá el efecto deseado de no volver a ejecutar nuevamente ese ciclo dado que únicamente son condiciones iniciales.

B. Módulo de Eventos

Este módulo está relacionado con la ocurrencia de eventos que son captados por los sensores, estos eventos son interpretados como cambios de nivel en las señales que recibe el controlador, los cuales pueden ser de subida o bajada. También es necesario señalar que estos eventos son parte de las condiciones que habilitan el disparo de una transición. Se utiliza un contacto normalmente abierto (NA) para describir que el evento corresponde a un borde de subida y un contacto normalmente cerrado para representa bordes de bajada. También se utiliza un contacto con la instrucción OSR, cuya función es la de detectar cambios de nivel en las señales de entrada del controlador.

C. Módulo de Condiciones

Este módulo representa en LD todas las condiciones asociadas al disparo de transiciones las cuales están formadas por el conjunto de eventos, condiciones, representadas por variables booleanas definidas sobre las transiciones, y la matriz de incidencia, la cual representa las condiciones propias de la dinámica del sistema.

Para transiciones que tienen asociadas retraso, se utilizan temporizadores en el cual el valor de retardo en el "PRESET" es el mismo que el retardo asignado a la transición. Por tanto es evidente que el tiempo es también una condición de disparo. Lo cual produce que el temporizador energice una bobina cuando haya alcanzado el tiempo de retardo.

D. Módulo de la Dinámica de la Red de Petri

Este módulo se encarga de la actualización del estado de la RP tras el disparo de una transición. Lo cual se describe con la ecuación de estado de la red. Razón por la cual, la implementación en LD de este módulo se hace uso de la matriz de incidencia.

Este módulo energiza las bobinas correspondientes a las los lugares que son post condiciones de la transición habilitada para disparar y des energiza las bobinas asociadas los lugares que corresponden a las precondiciones de la transición.

E. Módulo de Acciones

Este módulo hace una revisión del conjunto de lugares para determinar las funciones asociadas a los mismos los cuales pueden ser de nivel o impulsionales, para posteriormente asignar una bobina de salida a cada lugar que tenga definido una acción.

VI. Especificación de Requerimientos

Para lograr el alcance propuesto, resulta necesario definir el conjunto de funcionalidades con las que el sistema estará provisto. Estas se describen a continuación:

- Crear Red de Petri. Consiste en la construcción de la representación gráfica de la misma, vale decir dibujar los lugares, transiciones, arcos, asignar marcas y funciones a los lugares y definir temporización y condiciones en las transiciones.
- Editar Red de Petri. Posibilidad el ajuste de la Red de Petri previamente creada, lo cual incluye: mover elementos, borrar elementos y redefinir marcas, funciones, temporización y condiciones.
- Guardar Red de Petri. Permite guardar el diseño elaborado, en un archivo con formato XML.
- Abrir Red de Petri. Permite abrir un diseño previamente guardado, con el formato especificado.
- Generar Arreglos Matriciales. Permite estructurar y crear los diferentes arreglos matriciales que forman parte de la representación matricial de la Red de Petri, los cuales son: Vector de marcado inicial, matriz de incidencia previa, matriz de incidencia posterior y matriz de incidencia.
- Generar Tablas. Permite generar dos tablas, una que contienen la información de cada uno de los lugares creados y sus funciones asociadas y otra que contiene la información de las transiciones definidas junto a su temporización y condiciones asociadas.
- Generar Código. Permite la generación de Diagramas Ladder, a partir del modelado conceptual de un proceso con el editor gráfico implementado.

VII. Arquitectura del Sistema

La figura 1 muestra la arquitectura del sistema, la cual consiste en tres capas, en la cual los bloques que lo componen representan los servicios que forman parte del sis-

tema de desarrollo, en los cuales se implementan todas las funcionalidades que forman parte del análisis de requerimientos.

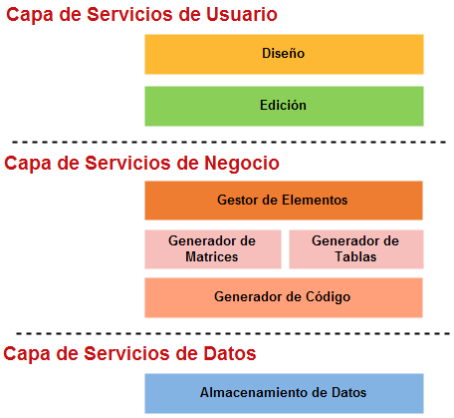


Figura 1: Arquitectura del Sistema

A. *Diseño*

Este servicio permite la interacción del usuario con el sistema para realizar el modelado de sistemas mediante RP, para esto el sistema contempla todos los escenarios previamente analizados en la especificación de requerimientos.

B. *Edición*

Este servicio permite al usuario modifica o ajustar el diseño previamente definido, entre las funcionalidades que posee se encuentra la opción de mover elementos, borrar Elementos, reasignas marcas y funciones a lugares y finalmente la reasignación de temporización y condiciones a las transiciones.

C. *Gestor de Elementos*

Este servicio es el encargado de realizar la obtención, actualización y la retención temporal de los datos que están siendo usados en la aplicación.

D. *Generador de Matrices*

Este servicio tiene el objetivo de generar todos los arreglos matriciales que representan la forma matemática de las RP.

E. *Generador de Tablas*

Este servicio es el encargado de ordenar los lugares y las transiciones definidas en tablas, para así mostrar de forma más ordenada las funciones y condiciones asignadas a los elementos creados.

F. *Generador de Código*

Este servicio consiste en la generación de cinco módulos independientes, con una parte de código que se pretende generar en cada uno, estos módulos se forman tanto de los arreglos matriciales como de las tablas previamente generadas.

G. *Generador de Código*

El almacenamiento de los datos del sistema se lo realizará en archivos XML con el objetivo de hacer más portables los diseños realizados con esta herramienta.

VIII. Implementación del Prototipo

El prototipo de la aplicación informática previamente diseñada, fue implementado utilizando Java como lenguaje de programación, por ser esta una herramienta que cuenta con excelentes recursos para la programación orientada objetos. Por otro lado, el formato utilizado para el almacenamiento de los datos definidos en la aplicación, es el XML.

IX. Descripción del Sistema

El sistema de desarrollo está compuesto por la barra menú, la barra de herramientas y un conjunto de pestañas en las cuales se visualiza, la representación gráfica de la RPIC, la representación matricial, las tablas de condiciones y funciones y los LD generados. La figura 2 Muestra la interfaz gráfica desarrollada.

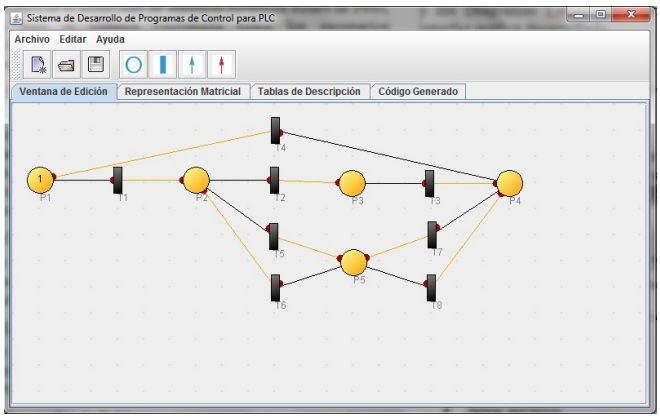


Figura 2: Interfaz Gráfica

A. *Barra de Herramientas*

La barra de herramienta se muestra en la Fig. 3, la cual contiene los elementos suficientes para el diseño de una RPIC.

Figura 3: Barra de Herramientas

Entre las funciones que contiene se encuentra:

- Crear Nuevo Archivo
- Guardar Archivo
- Abrir Archivo
- Definir Lugar
- Definir Transición
- Definir Arco Simple
- Definir Arco Habilitador

B. Ventana de Edición

La ventana de edición consiste en el área donde el usuario podrá realizar la representación gráfica de la RP, esta se muestra en la figura 2, en la cual se encuentra un gráfico diseñado con esta herramienta.

C. Representación Matricial

La pestaña de representación matricial, genera todos los arreglos matriciales que representan a la RP como se muestra en la figura 4.

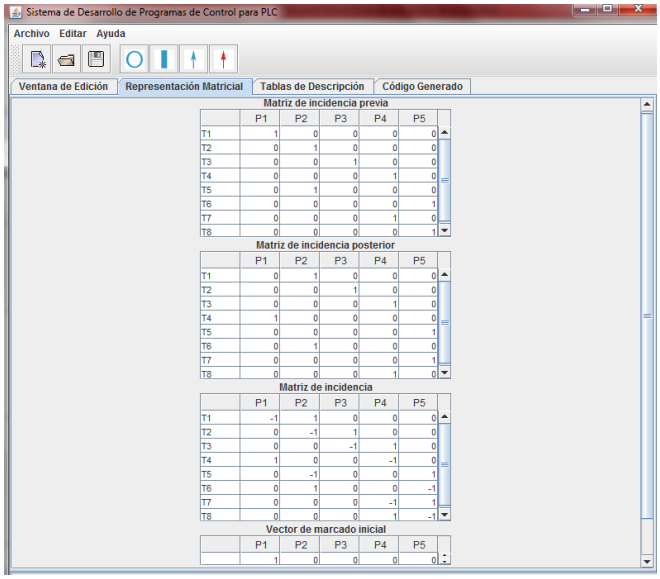


Figura 4: Representación Matricial

D. Tablas de Descripción

En esta pestaña se muestra de manera ordenada, las condiciones de disparo definidas sobre las transiciones y las funciones definidas en los lugares figura 5

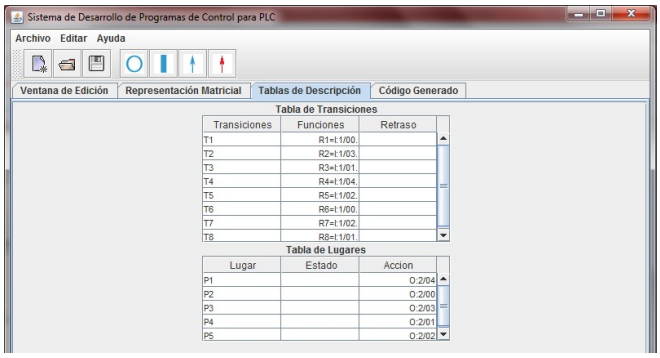


Figura 5: Tablas de Descripción

E. Código Generado

El código generado corresponde al lenguaje de Diagramas Ladder, el cual se visualiza en la última pestaña de la interfaz gráfica, La figura 6 muestra un segmento del programa generado.

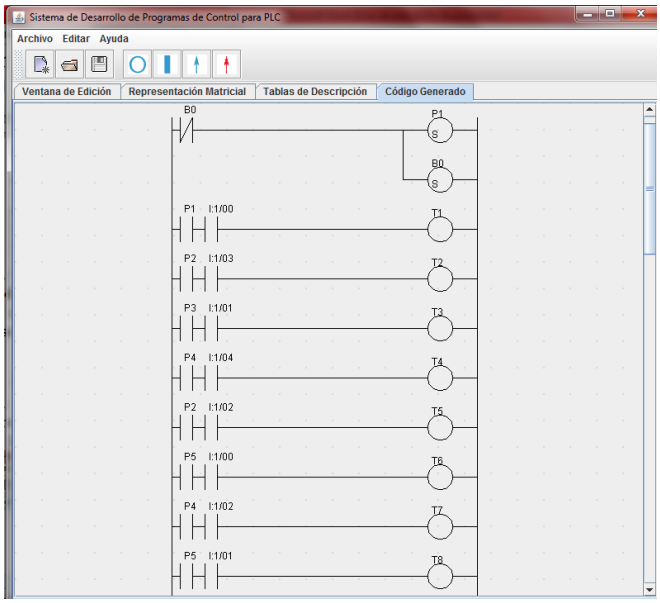


Figura 6: Código Generado

X. Apliación

En este punto se muestra la aplicación de esta herramienta, para el modelado de un sistema automatizado para la destilación de agua por termo-comprensión; proceso que consiste en la separación de contaminantes disueltos mediante la evaporación del agua, que posteriormente es condensada para obtener agua destilada.

A. Descripción de los Componentes del Sistema

El sistema está compuesto por un conjunto de elementos que hacen posible la interacción de tres procesos diferentes: Control de nivel del agua en el tanque, control

de temperatura, y finalmente el control de vacío. La figura 7 muestra un diagrama con todos los elementos que forman parte de este sistema.

1) Control de Nivel

El control de nivel se realiza mediante los siguientes elementos:

- Sensor de Nivel. Este sensor consiste en un flotador que entrega un nivel lógico alto, cuando el agua alcanza la altura de referencia, y nivel lógico bajo cuando la altura del agua está por debajo del nivel de referencia.
- Bomba de Agua. La bomba de agua inicia su funcionamiento con un nivel lógico alto de tensión. La cual es enviada por una de las salidas digitales del PLC.

2) Control de Temperatura

El control de temperatura se realiza mediante la interacción de los siguientes elementos:

- Sensor de Temperatura. Es necesario un circuito de acondicionamiento para establecer la temperatura de referencia, para el cual el sensor entregará un nivel lógico alto.
- Resistencias Eléctricas. Consiste en un conjunto de tres resistencias que deben ser alimentadas. Por otro lado dos de estas resistencias están en funcionamiento continuo, y la tercera se pone en funcionamiento cuando la temperatura del sistema es menor a la temperatura de referencia.

3) Control de Vacío

El control de vacío se realiza con los siguientes elementos:

- Sensor de Vacío. Consiste en un vacuostato, el cual entrega un nivel lógico alto a la entrada del PLC cuando se alcanza el valor de referencia.
- Bomba de Vacío. La bomba de vacío estará en funcionamiento durante todo el proceso, simplemente se ajustará a dos velocidades distintas dependiendo de la señal recibida por el sensor de vacío, lo cual se conseguirá mediante la conexión de un variador de frecuencia a este dispositivo.

B. Modelado del Controlador

El sistema de control está constituido por un conjunto sensores que se traducen en condiciones asociadas a transiciones que sirven para modificar el comportamiento del proceso.

1) Descripción de las Señales de Entrada

Las señales de entrada, proveniente de los sensores incorporados, son asignadas a como condiciones de disparo a las transiciones, las cuales se especifican en la tabla 1.

Condiciones	Descripción
S	Switch de encendido
N	Sensor de nivel
T1	Sensor de temperatura baja
T2	Sensor de temperatura alta
V	Sensor de vacío

Tabla 1: Especificación de los marcadores.

2) Descripción de las Señales de Salida

Las señales de salida son las responsable de poner en funcionamiento los elementos actuadores del sistemas, estas se describen en la tabla 2.

Funciones	Descripción
BA	Encender bomba de agua
R1	Encender resistencia eléctrica 1
R2	Encender resistencia eléctrica 2
BL	Encender bomba de vacío en velocidad baja
BH	Encender bomba de vacío en velocidad alta
VS	Abrir válvula de salida

Tabla 2: Descripción de las Condiciones.

3) Descripción de los Estados del Sistema (Lugares)

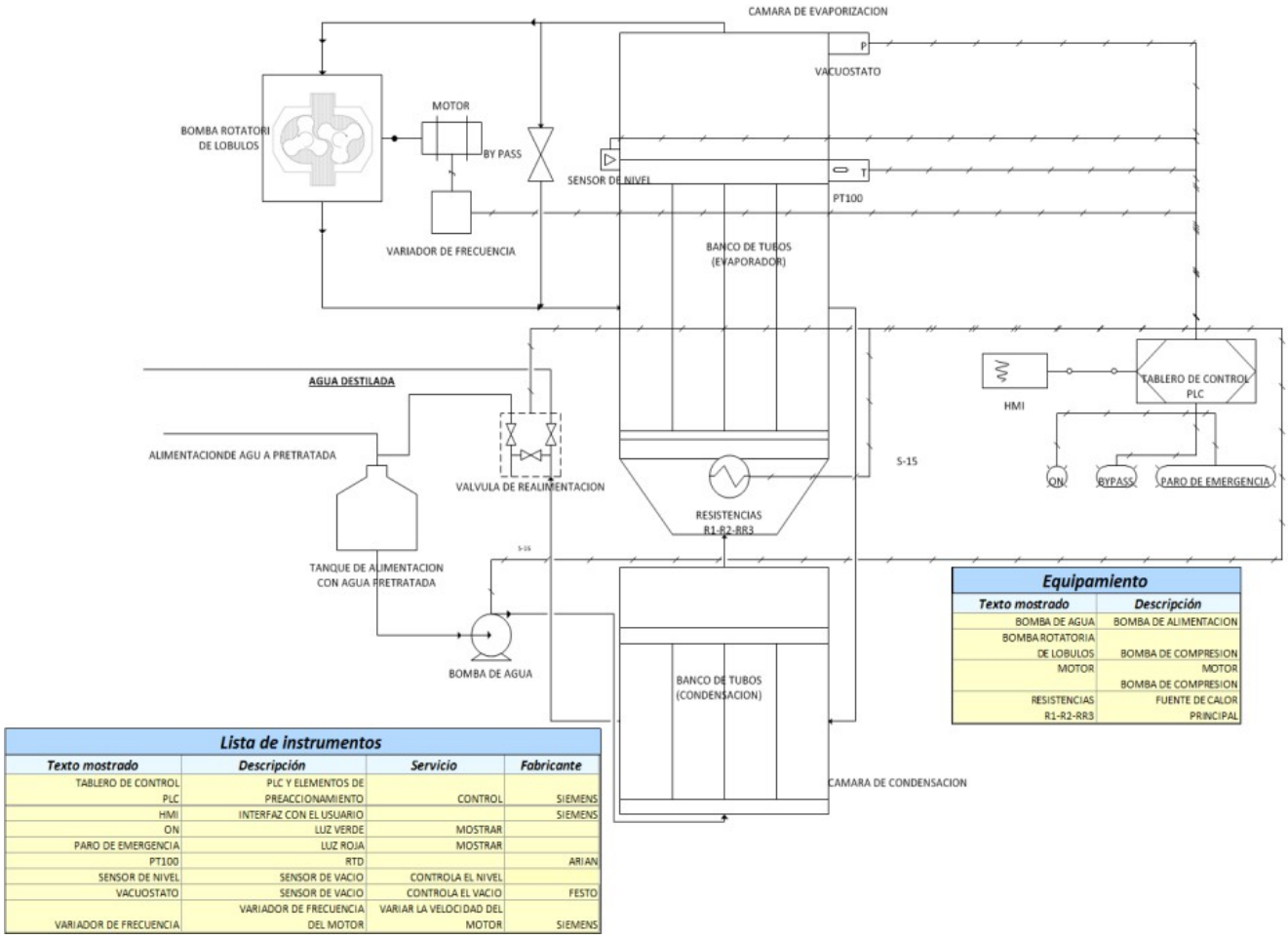
La tabla 3 muestra la descripción de los diferentes lugares definidos para modelar el sistema mediante RPIC, así como también el estado que cada uno de estos lugares representa y las funciones asignadas, funciones que son representadas mediante variables de tipo booleano.

4) Descripción de Eventos del Sistema (Transiciones)

En la tabla 4 se muestra el conjunto de transiciones, las cuales por un lado corresponden al conjunto de todos los escenarios que se pueden dar en el sistema, y por otro lado corresponden a los eventos que permitirán al sistema evolucionar de un estado a otro.

5) Representación Gráfica del Modelo del Controlador

El diseño de la RPIC que describe el controlador del sistema se realiza en base al sistema de desarrollo implementado, en el cual se recogen todas las consideraciones



Equipamiento

Texto mostrado	Descripción
BOMBA DE AGUA	BOMBA DE ALIMENTACION
BOMBA ROTATORIA DE LOBULOS	BOMBA DE COMPRESION
MOTOR	MOTOR BOMBA DE COMPRESION
RESISTENCIAS R1-R2-RR3	FUENTE DE CALOR PRINCIPAL

Lista de instrumentos

Texto mostrado	Descripción	Servicio	Fabricante
TABLERO DE CONTROL PLC	PLC Y ELEMENTOS DE PREACCIONAMIENTO	CONTROL	SIEMENS
HMI	INTERFAZ CON EL USUARIO		SIEMENS
ON	LUZ VERDE	MOSTRAR	
PARO DE EMERGENCIA	LUZ ROJA	MOSTRAR	
PT100	RTD		ARIAN
SENSOR DE NIVEL	SENSOR DE VACIO	CONTROLA EL NIVEL	
VACUOSTATO	SENSOR DE VACIO	CONTROLA EL VACIO	FESTO
VARIADOR DE FRECUENCIA	VARIADOR DE FRECUENCIA DEL MOTOR	VARIAR LA VELOCIDAD DEL MOTOR	SIEMENS

Figura 7: Diagrama Técnico del Destilador

Lugar	Función	Descripción
P1		Sistema apagado
P2	BA	Llenando tanque
P3		Tanque lleno
P4		Resistencias Apagadas
P5	R1	Resistencia eléctrica R1 encendida
P6	R2	Resistencia eléctrica R2 encendida
P7		Bomba de vacío apagada
P8	BL	Bomba de vacío encendida velocidad baja
P9	BH	Bomba de vacío encendida velocidad alta
P10		Válvula de salida cerrada
P11	VS	Válvula de salida abierta

Tabla 3: Lugares y Funciones asociadas

realizadas previamente. La figura 8 muestra el diseño realizado en el cual se definieron:

- Lugares, representados por los círculos.
- Transiciones, representados por los rectángulos.
- Arcos Simples, representados por las líneas con círculos rojos al final.
- Arcos Habilitadores, representados por las líneas con círculos azules al final.

Para la definición de las condiciones de disparo en la transiciones y asignación de funciones y marcado inicial en los lugares, el sistema está provisto de ventanas auxiliares que son activadas mediante un click derecho en cada uno de los elementos.

La Condiciones son definidas mediante la asignación de variables en las transiciones, los cuales pueden ser de 4 tipos: Flanco de subida, flanco de bajo, variables de nivel alto y variables de nivel alto. La figura 9 muestra las opciones para definir las condiciones

Las funciones son definidas mediante la asignación de variables en los lugares, los cuales son únicamente de nivel alto. Las marcas que corresponden a las condiciones iniciales del sistema se definen mediante la asignación de un número, el cual por ser una RP segura solo puede ser uno o cero. La figura 10 muestra la ventana para realizar estas asignaciones.

Transición	Condición	Descripción
t1	S	Encender sistema
t2	N	Apagar bomba de agua
t3	!N	Encender bomba de agua
t4	!T1!T2	Encender resistencia eléctrica R1
t5	↓T2	Encender resistencia eléctrica R2
t6	↓T1	Encender resistencia eléctrica R1
t7	T2	Apagar resistencia eléctrica R2
t8	T1	Encender resistencia eléctrica R2
t9	T1	Encender bomba de vacío en velocidad baja
t10	!T1	Apagar bomba de vacío
t11	!V	Aumentar velocidad de la bomba de vacío
t12	V	Disminuir velocidad de la bomba de vacío
t13	↓T1	Apagar bomba de vacío
t14	T1	Abrir válvula de salida
t15	↓T1	Cerrar válvula de vacío
t16, T17, T18, T19, T20, T21, T22	!S	Apagar el sistema

Tabla 4: Transiciones y condiciones asociadas

6) *Representación Matricial del Modelo del Controlador*

La representación matricial está compuesta por un conjunto de 4 arreglos matriciales, los cuales son mostrados en las figuras 11, 12, 13, 14.

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11
	1	0	0	1	0	0	1	0	0	1	0

Figura 14: Ventana para Definición de Marcas y Funciones

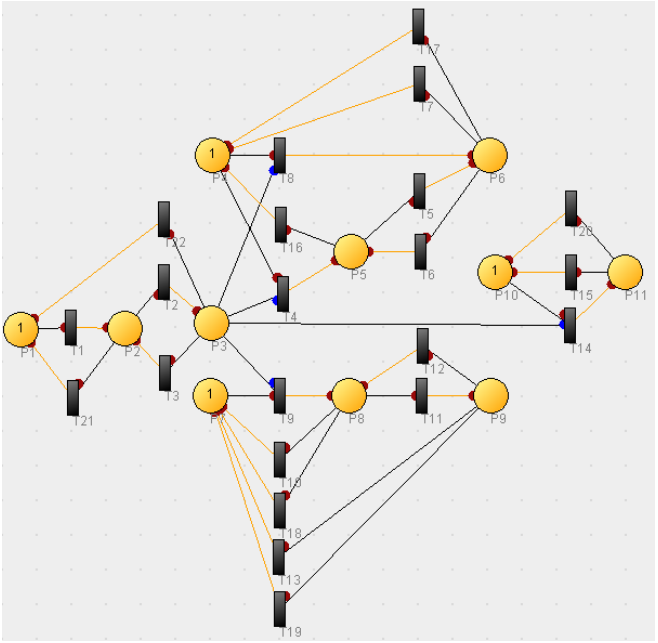


Figura 8: Diagrama Técnico del Destilador

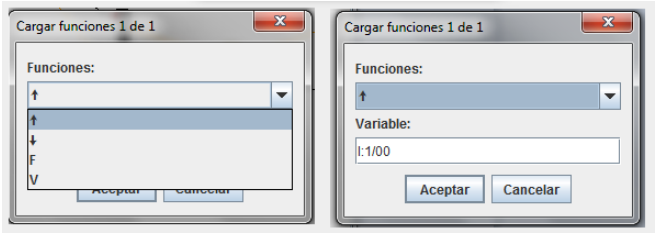


Figura 9: Ventanas para Definición de Condiciones

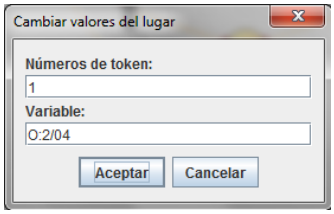


Figura 10: Ventana para Definición de Marcas y Funciones

7) *Código Generado*

En la figura 15 se muestra parte código generado que gobierna la lógica de control del PLC, a partir del modelo realizado con RIPC.

XI. Conclusión

Se realizó el diseño de un sistema de desarrollo para la programación de PLC, en el cual el modelado del controlador se lo realiza mediante Redes de Petri, para posteriormente generar de manera automática los Diagramas Ladder que serán implementados en el PLC, lo

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11
T1	1	0	0	0	0	0	0	0	0	0	0
T2	0	1	0	0	0	0	0	0	0	0	0
T3	0	0	1	0	0	0	0	0	0	0	0
T4	0	0	1	1	0	0	0	0	0	0	0
T5	0	0	0	0	1	0	0	0	0	0	0
T6	0	0	0	0	0	1	0	0	0	0	0
T7	0	0	0	0	0	1	0	0	0	0	0
T8	0	0	1	1	0	0	0	0	0	0	0
T9	0	0	1	0	0	0	1	0	0	0	0
T10	0	0	0	0	0	0	0	1	0	0	0
T11	0	0	0	0	0	0	0	1	0	0	0
T12	0	0	0	0	0	0	0	0	1	0	0
T13	0	0	0	0	0	0	0	0	1	0	0
T14	0	0	1	0	0	0	0	0	0	1	0
T15	0	0	0	0	0	0	0	0	0	0	1
T16	0	0	0	0	1	0	0	0	0	0	0
T17	0	0	0	0	0	1	0	0	0	0	0
T18	0	0	0	0	0	0	0	1	0	0	0
T19	0	0	0	0	0	0	0	0	1	0	0
T20	0	0	0	0	0	0	0	0	0	0	1
T21	0	1	0	0	0	0	0	0	0	0	0
T22	0	0	1	0	0	0	0	0	0	0	0

Figura 11: Ventana para Definición de Marcas y Funciones

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11
T1	0	1	0	0	0	0	0	0	0	0	0
T2	0	0	1	0	0	0	0	0	0	0	0
T3	0	1	0	0	0	0	0	0	0	0	0
T4	0	0	0	0	1	0	0	0	0	0	0
T5	0	0	0	0	0	1	0	0	0	0	0
T6	0	0	0	0	1	0	0	0	0	0	0
T7	0	0	0	1	0	0	0	0	0	0	0
T8	0	0	0	0	0	1	0	0	0	0	0
T9	0	0	0	0	0	0	0	1	0	0	0
T10	0	0	0	0	0	0	1	0	0	0	0
T11	0	0	0	0	0	0	0	0	1	0	0
T12	0	0	0	0	0	0	0	1	0	0	0
T13	0	0	0	0	0	0	1	0	0	0	0
T14	0	0	0	0	0	0	0	0	0	0	1
T15	0	0	0	0	0	0	0	0	0	1	0
T16	0	0	0	1	0	0	0	0	0	0	0
T17	0	0	0	1	0	0	0	0	0	0	0
T18	0	0	0	0	0	0	1	0	0	0	0
T19	0	0	0	0	0	0	1	0	0	0	0
T20	0	0	0	0	0	0	0	0	0	1	0
T21	1	0	0	0	0	0	0	0	0	0	0
T22	1	0	0	0	0	0	0	0	0	0	0

Figura 12: Ventana para Definición de Marcas y Funciones

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11
T1	-1	1	0	0	0	0	0	0	0	0	0
T2	0	-1	1	0	0	0	0	0	0	0	0
T3	0	1	-1	0	0	0	0	0	0	0	0
T4	0	0	-1	-1	1	0	0	0	0	0	0
T5	0	0	0	0	-1	1	0	0	0	0	0
T6	0	0	0	0	1	-1	0	0	0	0	0
T7	0	0	0	1	0	-1	0	0	0	0	0
T8	0	0	-1	-1	0	1	0	0	0	0	0
T9	0	0	-1	0	0	0	-1	1	0	0	0
T10	0	0	0	0	0	0	1	-1	0	0	0
T11	0	0	0	0	0	0	0	-1	1	0	0
T12	0	0	0	0	0	0	0	1	-1	0	0
T13	0	0	0	0	0	0	1	0	-1	0	0
T14	0	0	-1	0	0	0	0	0	0	-1	1
T15	0	0	0	0	0	0	0	0	0	1	-1
T16	0	0	0	1	-1	0	0	0	0	0	0
T17	0	0	0	1	0	-1	0	0	0	0	0
T18	0	0	0	0	0	0	1	-1	0	0	0
T19	0	0	0	0	0	0	1	0	-1	0	0
T20	0	0	0	0	0	0	0	0	0	1	-1
T21	1	-1	0	0	0	0	0	0	0	0	0
T22	1	0	-1	0	0	0	0	0	0	0	0

Figura 13: Ventana para Definición de Marcas y Funciones

cual simplifica el dificultoso proceso de diseño del controlador directamente con Diagramas Ladder, el cual no presenta la misma simplicidad en el diseño de sistemas de control complejos como las Redes de Petri, debido a

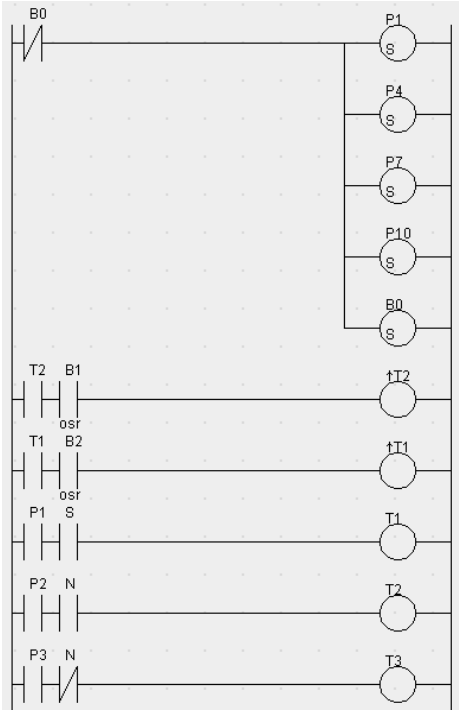


Figura 15: Ventana para Definición de Marcas y Funciones

la falta de una metodología clara en el proceso de diseño. Lo cual entre las produce altos tiempos de desarrollo y gran dependencia a la lógica del programador.

Por otro lado entre las ventajas del modelado de sistemas de control a eventos discretos mediante Redes de Petri se tienen:

- Facilidad en la especificación formal del proceso.
- Cuenta con elementos gráficos para representar fácilmente la dinámica de un proceso.
- Provee métodos de análisis del sistema mediante la validación de sus propiedades. Lo cual garantiza el apropiado desempeño del controlador.
- Facilita el mantenimiento y reconfiguración del controlador debido a la fácil comprensión que se tiene del modelo.

El sistema desarrollado genera Diagramas Ladder como lógica de control para la programación del PLC, dado que este lenguaje tiene un amplio uso en el desarrollo de proyectos con Autómatas Programables, además de ser el lenguaje que todo fabricante de PLC proporciona para la programación de sus dispositivos.

Entre las restricciones que presenta el diseño de sistemas de control con esta herramienta se tienen:

- El prototipo implementado no considera el diseño de sistemas que contengan elementos temporizadores y contadores.
- No se consideraron procesos que involucren operaciones aritméticas en el diseño del sistema.

Sin embargo el prototipo implementado permite cubrir las principales funcionalidades consideradas en el análisis de requerimientos, y constituye una opción más simple para el desarrollo de sistemas de control en Controladores Lógicos Programables.

Referencias

[1] Juan Felipe Medina Arboleda. *Manual técnico de programación de micro PLC's Siemens S7-200 basado en modelos de Redes de Petri bajo el lenguaje de programación KOP*. 2011.

[2] M. Baigorri, E. Jiménez y E. Baigorri. *Análisis y aplicación informática para implementación sobre PLC de sistemas automáticos desarrollados con herramientas de alto nivel*. 2003.

[3] M. Brito y J. Giraldo. *Metodología para diseño de circuitos Ladder con base en sistemas secuenciales y combinacionales*. 2010.

[4] J. Cardoso y R. Valette. *Redes de Petri*. 1997.

[5] José Eduardo Ribeiro Cury. *Teoria de controle supervisório de sistemas a eventos discretos*. 2001.

[6] R. David y H. Alla. *Discrete, Continuous, and Hybrid Petri Nets*. 2005.

[7] H. Gutiérrez y A. Muñoz. *Generación de diagramas Ladder mediante el uso de redes de Petri difusas*. 2013.

[8] G. Klein y M. Frey. *PLC programming with signal interpreted Petri nets*. 2003.

[9] Germán Darío Zapata Madrigal. *Diseño de automatismos secuenciales para controladores lógicos programables*. 2007.

[10] P. Martins, R.Dueire y P. Freire. *Intordução às Redes de Petri e aplicações*. 1996.

[11] M. Moreira, S. Botelho y J. Basilio. *Ladder Diagram implementation of Control Interpreted Petri Nets: A state equation approach*. 2009.

[12] A. Orozco, C. Guarnizo y M. Olguín. *Automatismos industriales*. 2008.

[13] C. Robles y D. Jaimes. *Revisión bibliográfica de las técnicas que relacionan la lógica de programación de red de Petri con la lógica de programación de escalera para la implementación de sistemas automatizados de control*. 2011.

[14] R. Sanchis, J. Romero y C. Ariño. *Automatización industrial*. 2010.

[15] Rafael Pentagna Silvestre. *Implementação em Ladder de sistemas de automação descritos por redes de Petri interpretadas para controle*. 2010.

[16] M. Uzam y A. Jones. *Discrete event control system design using Automation Petri Nets and their Ladder Diagram implementation*. 1998.

[17] G. Zapata y E. Carrasco. *Estructuras generalizadas para controladores lógicos modeladas mediante redes de Petri*. 2002.