

Ingeniería. Revista de la Universidad de

Costa Rica

ISSN: 1409-2441

marcela.quiros@ucr.ac.cr

Universidad de Costa Rica

Costa Rica

Álvarez Salaza, Johny; Mejía Arango, Juan Guillermo

Metodología de programación de autómatas programables a partir de Redes de Petri
Ingeniería. Revista de la Universidad de Costa Rica, vol. 29, núm. 2, julio-diciembre,

2019, pp. 32-43

Universidad de Costa Rica

Ciudad Universitaria Rodrigo Facio, Costa Rica

Disponible en: <https://www.redalyc.org/articulo.oa?id=44170543002>

- ▶ Cómo citar el artículo
- ▶ Número completo
- ▶ Más información del artículo
- ▶ Página de la revista en redalyc.org

Metodología de programación de autómatas programables a partir de Redes de Petri

Programming Methodology for Programmable Automatas from Petri Nets

Johny Álvarez Salazar, Instituto Tecnológico Metropolitano, Colombia.

Ingeniero Electromecánico, MSc. Automatización y Control Industrial.

Correo: johny.alvarez@itm.edu.co

<https://orcid.org/0000-0002-7041-8619>

Juan Guillermo Mejía Arango, Instituto Tecnológico Metropolitano, Colombia.

Ingeniero Electrico, MSc. Gestión Energética.

Correo: juanmejia@itm.edu.co

<https://orcid.org/0000-0003-1210-6550>

Recibido: 9 de marzo 2019

Aceptado: 17 de junio 2019

Resumen

Este trabajo tiene por objeto proponer una metodología para la programación de autómatas programables con TIA Portal a partir de Redes de Petri. Para esto se selecciona una aplicación industrial donde se requiere hacer una mezcla de pinturas; se presenta, entonces, un proceso simulado compuesto por tres tanques con sus correspondientes válvulas, las cuales se activarán por un tiempo determinado de acuerdo con el color requerido en la preparación. Tres pulsadores seleccionarán el color de la pintura que se quiere producir. Inicialmente, se plantea la red de Petri; luego, por medio de matrices se despliega la ecuación de estados para los disparos de todas las transiciones hasta regresar al estado inicial. La metodología establece que luego se deben asociar las salidas a los diferentes estados y las transiciones se asocian a entradas. La red de Petri es, entonces, traducida al lenguaje de programación, teniendo cuidado de activar el estado siguiente y desactivando el estado anterior según la transición que se dé. Finalmente, las salidas son activadas de acuerdo a los estados en que fueron ubicadas. Al realizar la simulación en el PLC, se pudo comprobar el comportamiento adecuado de la secuencia de acuerdo con lo que se tenía previsto, demostrándose así la efectividad de la metodología presentada.

Palabras clave:

TIA Portal, Red de Petri, Automatización, Ladder, Simatic

Abstract

This investigation looks to propose a methodology for the programming of programmable automatas with TIA Portal from Petri Nets. In order to do it, an industrial application that requires to mix paints is selected; then, a simulated process that involves three tanks with their corresponding valves that activate for each one of them during an amount of time according to the color required in the preparation. Three



Esta obra está bajo una Licencia de Creative Commons. Atribución - No Comercial - Compartir Igual

push buttons will select the color of the painting that you want to produce. Initially, the Petri net is considered and then using of matrices the states equation is displayed for the trigger of all the transitions until returning to the initial state. The methodology establishes that after it the outputs must be associated to the different states and the transitions are associated to inputs. The Petri network is, then, translated into the programming language, taking care to activate the next state and to deactivate the previous state according to the transition that occurs. Finally, the outputs are activated according to the states in which they were placed. When executing the simulation in the PLC, it was possible to verify the proper behavior of the sequence according to what was planned, thus demonstrating the effectiveness of the presented methodology.

Keywords:

TIA Portal, Petri Nets, Automation, Ladder, Simatic

1. INTRODUCCIÓN

Según Chmie (2019), un PLC está construido bajo el estándar IEC 61131, es un dispositivo con la capacidad de leer un entorno a través de sus entradas, y en el que se puede realizar un algoritmo de control para luego, a través de sus salidas, ejecutar unas acciones.

El algoritmo de control puede realizarse en diversos lenguajes de programación, siendo el más conocido y utilizado el ladder o lenguaje de escalera.

Dependiendo del fabricante, se pueden tener alternativas de programación como por ejemplo Grafset y programación estructurada, en Chmie (2019) también se define que estos lenguajes de programación deben estar bajo el estándar IEC 61131-3. Y de acuerdo a Álvarez Salazar y Mejía Arango (2017), los Controladores Lógicos Programables (PLC) se han convertido en una de las herramientas más importantes para soportar los proyectos de automatización que requieren las empresas.

Para Oliveira, Dias Da Silva, Gorgônio y Perkusich (2011), una Red de Petri permite representar datos complejos mediante estados y marcas, posibilitando que los modelos puedan ser descritos de una forma más concisa mediante lugares-transiciones. Una Red de Petri está compuesta por: lugares, transiciones, marcas y arcos. Los lugares representan condiciones, actividades o recursos; las marcas representan disponibilidad de recursos; las transiciones representan eventos; y los arcos indican lugares de entrada/salida de las transiciones.

Los elementos de la red de Petri permiten realizar un paralelo con las variables físicas del PLC, las salidas de este se asocian a los estados y las entradas a las transiciones, permitiendo esto, realizar la solución de automatización, para luego pasar a implementarla en un algoritmo dentro del autómata programable.

Este artículo está organizado en cuatro secciones incluyendo esta. La segunda sección presenta los materiales y métodos utilizados, correspondientes a las definiciones, ecuaciones y planteamientos matemáticos de las redes de Petri. En la tercera sección se presenta la metodología propuesta en el desarrollo de un problema. Finalmente en la sección 4 se concluye este artículo.

2. MATERIALES Y MÉTODOS

Silva (1985) define una Red de Petri Ordinaria (RdPO), N, como un cuádruplo:

$$N = \langle P, T, Pre, Post \rangle,$$

donde:

$P = \{P_1, P_2, \dots, P_m\}$: es un conjunto finito y no vacío de lugares

$T = \{t_1, t_2, \dots, t_n\}$: es un conjunto finito y no vacío de transiciones

$Pre: P \times T \rightarrow \{0,1\}$: es el conjunto de lugares de entrada a T

$Post: T \times P \rightarrow \{0,1\}$: es el conjunto de lugares de salida de T

Una RdPO marcada es un par $Nm = \langle N, M_0 \rangle$, en el cual N es una Red de Petri Ordinaria y M_0 es el marcado inicial.

Una Red de Petri controlada por pesos de los arcos está definida por una tupla:

$$\varrho = (P, T, W, C, B),$$

donde:

P:	es un conjunto de lugares
T:	es un conjunto finito de transiciones
$W: (P \times T) \cup (T \times P) = N$:	es una función que indica el peso de los arcos conectores entre los lugares y las transiciones (donde N es un entero no negativo del conjunto de los enteros)
C:	es un conjunto finito de lugares de control
$B: (C \times T) = \{0,1\}$:	es el peso binario de un arco dirigido; asociando los lugares con las transiciones

El peso de un arco desde un lugar P_i hasta una transición T_j es definido por $w(P_i, T_j)$ y el peso desde un arco desde T_j hasta P_i por $w(T_j, P_i)$.

Cuando una transición dispara, toma la cantidad de marcas requeridas para la salida y distribución hacia los siguientes lugares según su peso del arco. Si los pesos de los arcos son iguales para todos los lugares de salida, las marcas se mueven a través de la transición. Si existe alguna diferencia en los pesos de los arcos, las marcas pueden desaparecer o ser creadas según el lugar de salida al que estas se dirijan. De hecho, una transición al disparar lo hace produciendo y eliminando marcas, con lo que se puede decir que la transición trabaja en función del peso del arco.

De acuerdo a Murillo (2008), las Redes de Petri son nombradas según sus características, es ordinaria si el peso de sus arcos es siempre uno. Entre sus principales propiedades dinámicas se encuentra la alcanzabilidad, la cual consiste en que cada disparo de una transición habilitada modifica la distribución de los marcados dentro de la red, de acuerdo con las reglas de disparo. Una secuencia de disparos generará una secuencia de marcados. Se dice que un marcado M_n es alcanzable desde el marcado M_0 si, y sólo si, existe una secuencia de disparos que transforme M_0 en M_n . La secuencia de disparos se denota por sigma: $\sigma = M_0 t_1 M_1 t_2 \dots M_n t_n$

Además, una Red de Petri es reversible si para cada marcado M_n existente, M_0 es alcanzable desde M_n . De esta forma, una Red de Petri reversible es aquella donde siempre es posible alcanzar nuevamente el marcado inicial o estado inicial del sistema.

En Murata (1989) se propone la ecuación de estados de una Red de Petri (1) como una ecuación matricial que depende del marcado inicial y de una secuencia de disparos de estados habilitados.

Por otro lado, el TIA PORTAL (Totally Integrated Automation Portal) es considerado como

una herramienta que integra diferentes softwares de SIEMENS como son: el SIMATIC STEP 7, el WinCC y componentes y módulos para soluciones de automatización basadas en PC con SIMATIC. Por su parte, la ejecución del controlador Lógico Programable está dividida en ciclos de scan. Durante un ciclo, el flujo de energía pasa de izquierda a derecha y de arriba abajo cada segmento a la vez. Cuando el ciclo llega al final del diagrama, este inicia nuevamente. En el comienzo del ciclo, todas las entradas son leídas y almacenadas. Durante la ejecución, los datos que van a ser usados como salidas pueden ser modificados por la lógica de programación.

$$M_d = M_0 + A^T \sum_{k=1}^d U_k \quad (1)$$

donde:

M_0 : es el marcado inicial

M_d : es un vector columna de $m \times 1$

U_k : es el vector de disparo

A^T : es la matriz de incidencia, denota como cambiará el marcado

En la Figura 1 se presenta en forma esquemática el proceso que se sigue en la aplicación de la metodología planteada.

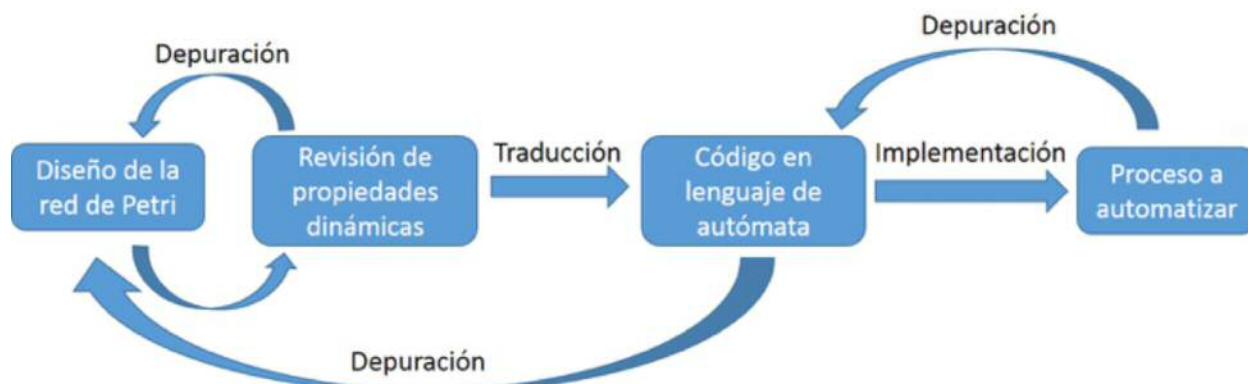


Figura 1. Relación entre la Red de Petri y el lenguaje del autómata.

3. RESULTADOS

Para un proceso simulado de una mezcladora de pinturas, en la que se tienen tres tanques: uno con pintura amarilla (Válvula Q0.0), otro con azul (Válvula Q0.1) y un tercer tanque con pintura roja (Válvula Q0.2). Se requiere realizar unas mezclas teniendo en cuenta que cuando se abre

la válvula de uno de los tanques salen 10 gramos por cada segundo. Las mezclas que se pueden obtener son las siguientes:

- verde: 50 gramos de amarillo (5 segundos de Q0.0) y 50 gramos de azul (5 segundos de Q0.1)
- púrpura: 40 gramos de azul (4 segundos de Q0.1) y 60 gramos de rojo (6 segundos de Q0.2)
- naranja: 30 gramos de rojo (3 segundos de Q0.2) y 70 gramos de amarillo (7 segundos de Q0.0)

Se tiene, entonces, que son tres procesos independientes y que se debe partir de un estado de marcado inicial para realizar cualquiera de ellos, y que van a depender del vector de disparo con las transiciones sensibilizadas T1, T2, T3. Así entonces teniendo en cuenta la ecuación (1) planteada en Murata (1989) después del disparo de la transición sensibilizada T1 el marcaje de la red se presenta en A:

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \end{bmatrix} * \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad \text{A}$$

Si el disparo se da en T2, el marcaje es presentado en B:

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \end{bmatrix} * \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad \text{B}$$

Si el disparo se da en T3, el marcaje es presentado en C:

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \end{bmatrix} * \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad \text{C}$$

Se observa así la total dependencia del estado sensibilizado de las transiciones T1, T2 y T3, conservándose la matriz de incidencia y el vector de disparo iguales para los tres casos con el estado de marcado inicial.

Ya para el caso de retorno al estado inicial una vez terminado cada proceso, se presentan los siguientes casos:

Con marcaje en el estado dos y disparo en la transición cuatro se presenta D.

$$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \end{bmatrix} * \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{D}$$

Con marcaje en el estado tres y disparo en la transición cinco se presenta E.

$$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \end{bmatrix} * \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{E}$$

Con marcaje en el estado tres y disparo en la transición seis se presenta F.

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} -1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \end{bmatrix} * \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad \mathbf{F}$$

Pasos para obtener el código Ladder

Paso cero, Red de Petri.

Depende de los diferentes estados de la máquina y la interacción entre los mismos. Se deben colocar las direcciones al lado de cada estado, teniendo en cuenta el orden de la secuencia del proceso. Para el caso de las transiciones, se colocan en forma cronológica en número arábigo anteponiendo la letra T. En la Figura 2 se presenta la Red de Petri. Donde M0.0 es el estado inicial (estado uno) del proceso, M0.1 es el proceso de fabricación del color Naranja (estado dos), M0.2 es el proceso de fabricación del color Verde (estado tres), M0.3 es proceso de fabricación del color Púrpura (estado cuatro) y T1, T2, T3, T4, T5, T6 y T7 las transiciones del sistema.

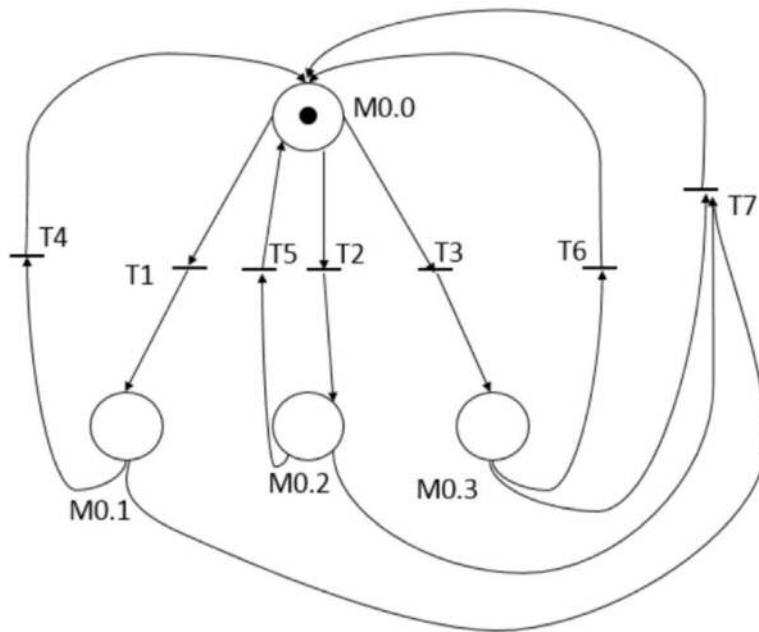


Figura 2. Red de Petri Mezcladora de pinturas.

Paso uno, tabla con estados-salidas y Transiciones-entradas

Se realiza una tabla especificando las relaciones existentes entre los estados con las salidas del PLC y las transiciones con las entradas del PLC. En la Tabla 1 se presenta la información suministrada de acuerdo al problema propuesto. Para el caso de las reversiones al estado inicial, en este caso se realizan con flancos negativos de la salida que tiene el tiempo más extenso, de acuerdo a cada proceso de fabricación.

Tabla 1. *Estados-salidas y Transiciones-entradas.*

Estado	Salidas	Transición	Entradas
M0.0	Ninguna	T1	I0.0 (Naranja)
M0.1	Q0.0 (Amarillo)	T2	
	T. Impulso 7s		I0.1 (Verde)
	Q0.2 (Rojo)	T3	I0.2 (Púrpura)
	T. Impulso 3s		
M0.2	Q0.0 (Amarillo)	T4	N de Q0.0
	T. Impulso 5s		
	Q0.1 (Azul)	T5	N de Q0.1
	T. Impulso 5s		
M0.3	Q0.1 (Azul)		
	T. Impulso 4s	T6	N de Q0.2
	Q0.2 (Rojo)		
	T. Impulso 6s		
		T7	I0.3 (stop)

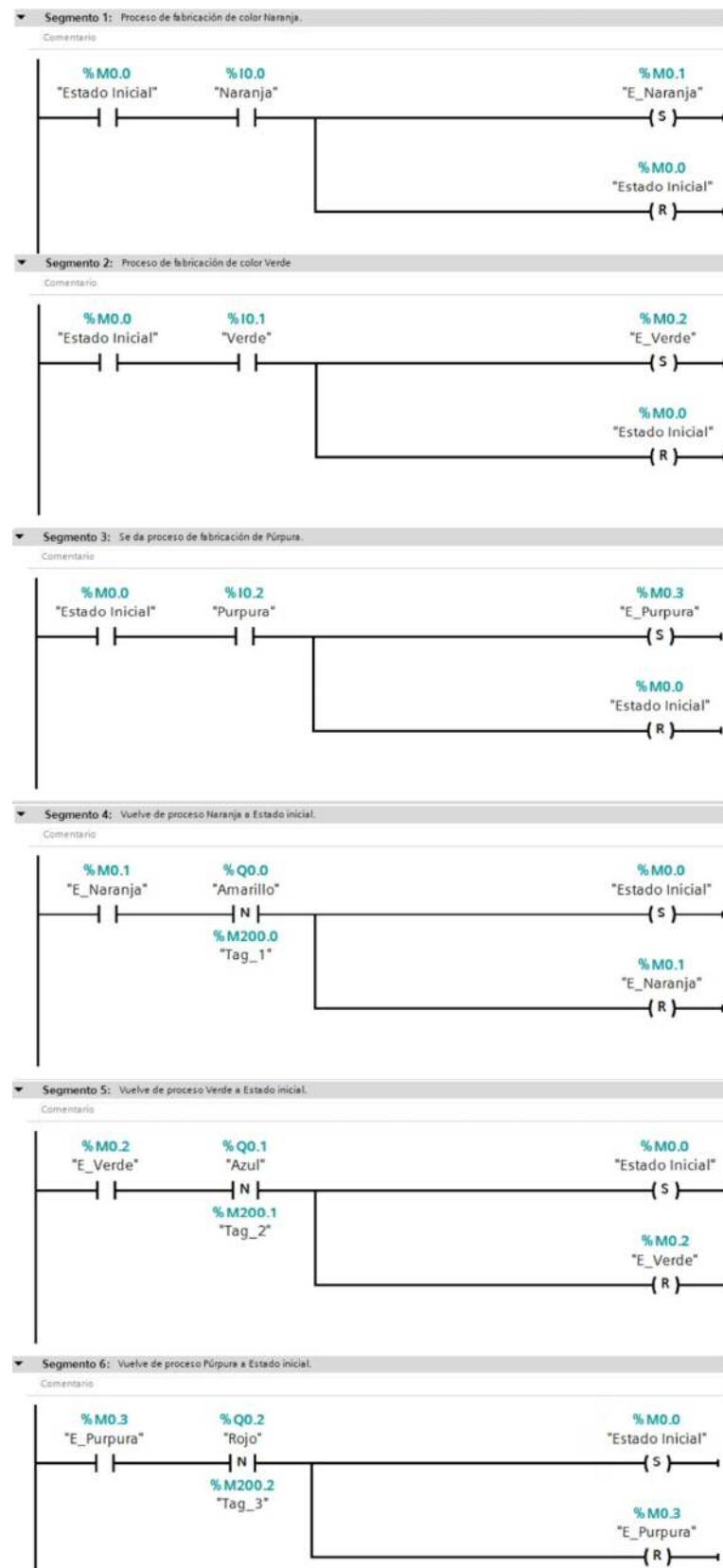


Figura 3. Código ladder paso 2, de acuerdo a la Red de Petri.

Paso dos, conversión de la red de Petri a lenguaje de PLC

La cantidad de segmentos utilizados corresponde a la misma cantidad de transiciones existentes. Como se tienen 6 transiciones, se realizan 6 segmentos de código (ver Figura 3).

Paso tres, salidas del PLC

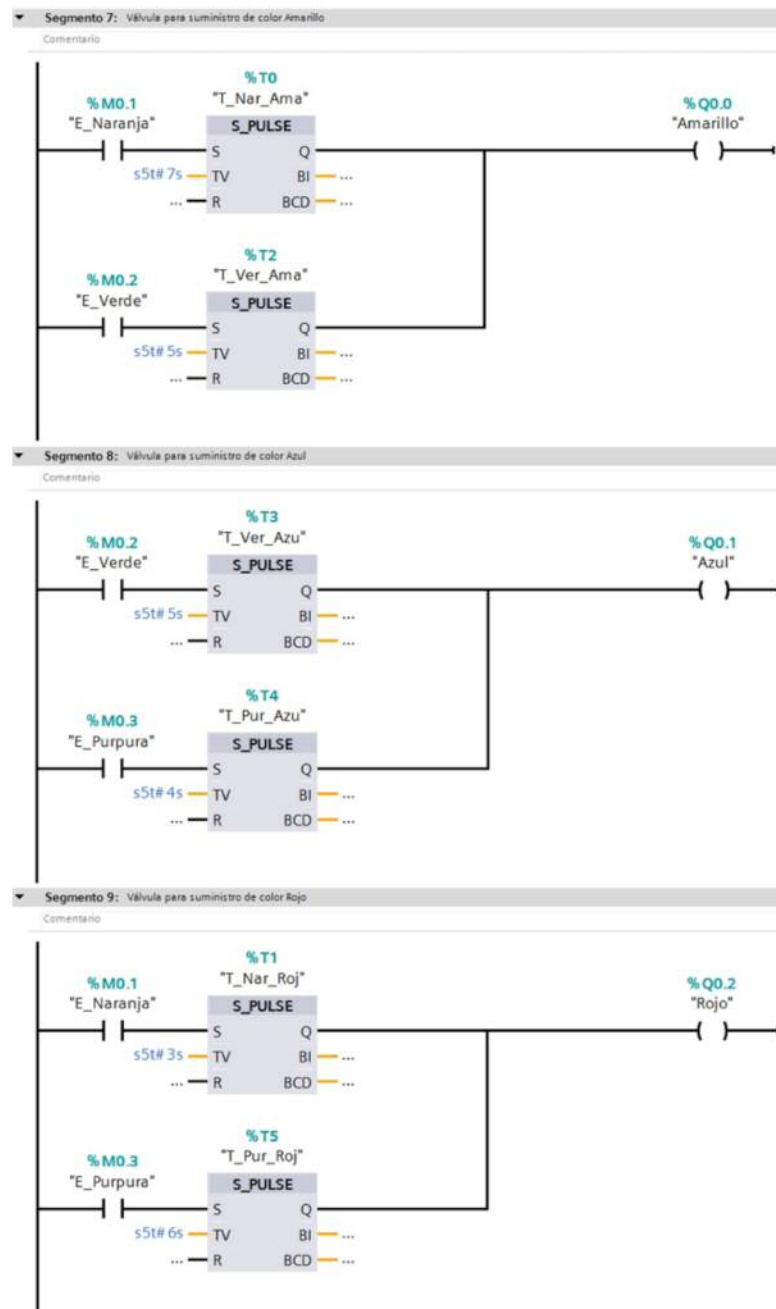


Figura 4. Código ladder paso 3, de acuerdo a la Red de Petri.

Se colocan todas las salidas utilizadas del PLC en segmentos independientes. Luego se procede a enlazarlas con los estados de las cuales dependen, en paralelo. De acuerdo a la Tabla 1, se tienen tres salidas (Q0.0, Q0.1, Q0.2). Dado que el problema planteado enuncia que se tiene una relación entre el tiempo y los gramos de cada color suministrado, se utilizan temporizadores al trabajo (ver Figura 4).

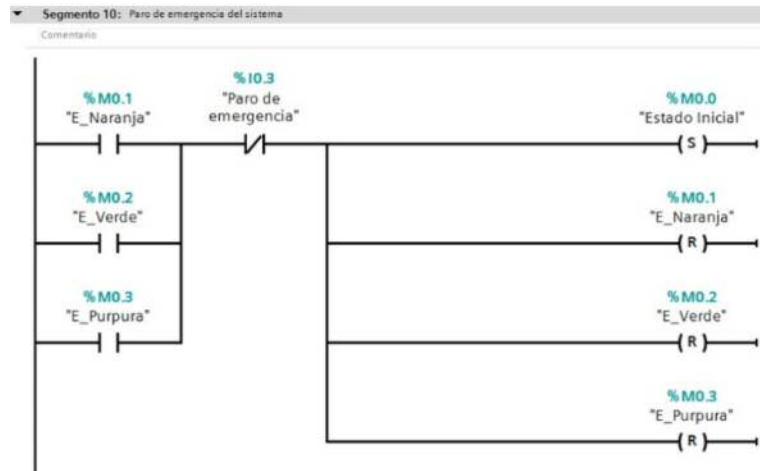


Figura 5. Paro de emergencia del sistema de acuerdo a la Red de Petri.



Figura 6. Código Ladder en bloque Complete Restart.

El paro de emergencia es presentado en la Figura 5. Estando el proceso en cualquiera de los estados, este regresa a las condiciones iniciales.

Para garantizar el marcaje del estado inicial se utiliza el bloque de organización OB100 (Complete Restart), y se coloca un contacto cerrado que se activa por una única vez durante el arranque del PLC el estado M0.0 (ver Figura 6).

4. CONCLUSIONES

En este trabajo se ha demostrado el correcto desempeño de una metodología para automatizar procesos industriales, específicamente en una planta de preparación de pinturas. Dicha metodología permite desarrollar un algoritmo de programación de un controlador lógico programable a partir del planteamiento de una red de Petri que describe la secuencia del proceso.

En el desarrollo del trabajo fue factible graficar una red de Petri que describe la secuencia y rutas del proceso a partir del modelo matricial. Combinando el marcado inicial, la matriz de incidencia y el vector de disparo. Se obtuvo el marcaje de la red de cada transición disparada hasta que se retornó al estado inicial.

Una vez determinada la red de Petri y comprobado su funcionamiento matricial para el proceso requerido, es relativamente sencillo y repetitivo el proceso de elaborar un código para la programación del PLC. Este último aspecto puede ser considerado como una desventaja, no de la metodología, pero sí del proceso, puesto que se requiere una acción manual del programador, no sistematizada para transformar la información gráfica a un código para el PLC.

Para la solución de problemas de automatización de eventos lógicos y secuenciales, frecuentemente se utilizan métodos heurísticos fundamentados en la experiencia del programador que no responden a un formalismo, resultando en soluciones donde no es posible sistematizar la metodología de diseño. El formalismo permite obtener algoritmos validados, verificados; resolviendo los problemas en forma sistemática. Las redes de Petri responden a este formalismo para definir sistemas de control; sin embargo, requieren una adaptación, así sea simple, por parte del programador para implementarlos en un PLC; convertir lo observado en un gráfico (Red de Petri) en instrucciones entendibles por el dispositivo controlador. Los fabricantes de PLC han intentado introducir sistemas basados en Redes de Petri para implementarlos directamente en sus equipos, tal es el caso del GRAFCET (gráfico funcional de etapas y transiciones) bajo la norma IEC 60848, ofreciendo un método gráfico de modelación y diseño eficiente pero que aún carece de algunos elementos como la definición de requerimientos, los factores de aceptación y los métodos de verificación del algoritmo. Lo anterior resulta en una buena oportunidad para adelantar proyectos de investigación que ayuden a obtener una formalización más completa.

REFERENCIAS

- Álvarez Salazar, J. A. y Mejía Arango, J. G. (2017). *TIA PORTAL. Aplicaciones de PLC*. Medellín: Fondo Editorial ITM.
- Chmie, M. (2019). FPGA-based implementation of bistable function blocks defined in the IEC 61131. (Elsevier, Ed.) *Microprocessors and Microsystems*, 65, 37-46. Recuperado de <http://www.sciencedirect.com/science/article/pii/S0141933117302387>
- Murata, T. (1989). Petri nets: Properties, Analysis and application. *Proceedings of the IEEE*, 77(4), 541-580.
- Murillo, L. D. (2008). Redes de Petri: Modelado e implementación de algoritmos para autómatas programables. *Tecnología en Marcha*, 21(4), 102-125.
- Oliveira, E. A., Dias Da Silva, L., Gorgônio, K. y Perkusich, A. M. (2011). CONVERTENDO DIAGRAMAS LADDER EM MODELOS DE REDES DE PETRI COLORIDAS. *Simpósio Brasileiro de Automação Inteligente*.
- Silva, M. (1985). *Las Redes de Petri y la informática*. Madrid: Editorial AC.