

# DESARROLLO DE UN SIMULADOR DIDÁCTICO DE REDES DE PETRI DE BAJO NIVEL

***Fernando Bautista Hernández***

Tecnológico Nacional de México, Instituto Tecnológico de Orizaba  
*fbautistah@gmail.com*

***Ignacio Herrera Aguilar***

Tecnológico Nacional de México, Instituto Tecnológico de Orizaba  
*iherrera@ito-depi.edu.mx*

***Oscar Osvaldo Sandoval González***

Tecnológico Nacional de México, Instituto Tecnológico de Orizaba  
*osandoval@ito-depi.edu.mx*

***Gerardo Águila Rodríguez***

Tecnológico Nacional de México, Instituto Tecnológico de Orizaba  
*gerardo\_aguila03@yahoo.com.mx*

***José Pastor Rodríguez Jarquin***

Tecnológico Nacional de México, Instituto Tecnológico de Orizaba  
*jprj\_02@hotmail.com*

## **Resumen**

Este documento representa una propuesta de un simulador didáctico que puede emplearse para el aprendizaje de redes de Petri de bajo nivel que permitan al estudiante adentrarse en el manejo de los conceptos básicos de las mismas y en la formación de los modelos matemáticos que permiten llevar a cabo la evolución del marcado del sistema; éste se ha desarrollado tomando en cuenta las reglas

propias del comportamiento dinámico de las redes de Petri, que son fundamentales para llevar a cabo las transiciones en éstas.

El resultado obtenido es un programa de cómputo funcional que, comparado con otras herramientas gráficas, brinda la posibilidad de comprender la evolución del mercado bajo una perspectiva matemática.

**Palabra(s) Clave(s):** Simulador, didáctico, redes de Petri, evolución de mercado.

## 1. Introducción

Las redes de Petri (RdP o rdP), permiten representar de forma natural un modelo de eventos discretos, en éstas los eventos están asociados a transiciones y las actividades a lugares.

Éstas han sido ampliamente difundidas como una herramienta de modelado gráfico para la descripción de sistemas, se han empleado indistintamente en sistemas químicos, informáticos, de control de producción, así como en sistemas de manufactura automatizados comandados por controladores lógicos [1] [2], se ha incrementado el interés por el desarrollo de sistemas de control haciendo uso de las redes de Petri [3].

Para la enseñanza del modelado empleando redes de Petri es necesario contar con herramientas computacionales que permitan al estudiante conocer y poner en práctica la operación de los elementos que conforman a la misma.

Existe una gran cantidad de herramientas computacionales producto de trabajos de investigación de diferentes instituciones que sirven para llevar a cabo el diseño, análisis, y simulación de redes de Petri simples y de alto nivel [4] [5], sin embargo éstas herramientas, por lo general gráficas, no hacen mención del desarrollo matemático necesario para la comprensión de la evolución del mercado en redes de Petri, en [6] se muestra un desarrollo completo pero como una metodología con ejemplos no es un software que permita ingresar los datos de la red y nos muestre la evolución de la misma, en este trabajo se mostrará un simulador desarrollado para la comprensión de las bases matemáticas que permiten simular redes de Petri simples.

## 2. Desarrollo

Matemáticamente una rdP puede definirse a partir de la ecuación 1 de 5 elementos.

$$rdP = (P, T, A, W, M_0) \quad (1)$$

Donde el significado de cada una de las siglas utilizadas es el siguiente:

- $P = \{P_1, P_2, P_3, \dots, P_{np}\}$ : Conjunto de nodos tipo lugar.
- $T = \{T_1, T_2, T_3, \dots, T_{nt}\}$ : Conjunto de nodos tipo transición.
- $A = \{A_1, A_2, A_3, \dots, A_{na}\}$ : Conjunto de arcos de la rdP. Es subconjunto del producto cartesiano de todos los nodos P y T.
- $W = A_i \mid \rightarrow \{1,2,3, \dots\} \forall A_i$ : Peso asociado a cada arco.
- $M_0 = P_i \mid \rightarrow \{1,2,3, \dots\} \forall P_i$ : Número de marcas o tokens iniciales en cada nodo tipo lugar (marcado inicial).

El estado del sistema queda totalmente determinado por el número de marcas en cada nodo tipo lugar y puede describirse matemáticamente por el vector  $P$ .

Para representar y analizar la rdP es conveniente definir la función  $E(T_j)$  que representa el conjunto de lugares de entrada a la transición  $T_j$ . De forma similar, se puede definir la función  $S(T_j)$  para indicar los lugares de salida de la transición  $T_j$  :, ecuaciones 2 y 3.

$$E(T_j) = \{P_i \in P, (P_i, T_j) \in A\} \quad (2)$$

$$S(T_j) = \{P_i \in P, (T_j, P_i) \in A\} \quad (3)$$

Las reglas para simular el comportamiento dinámico de rdP son:

- Una transición  $T_i$  esta activada si cada uno de los nodos  $P_j$  tipo lugar que están conectados a la entrada contienen al menos  $W(P_j, T_i)$  marcas.  $W(P_j, T_i)$  representa el peso del arco que une el nodo  $P_j$  con la transición  $T_i$

(si en un arco no aparece su peso, se maneja el valor de 1), de forma resumida se dice que una transición  $T_i$  (ecuación 4):

$$M(P_j) \geq W(P_j, T_i) \forall P_j \in E(T_i) \quad (4)$$

Donde  $M(P_j)$  es el número de marcas en el nodo tipo lugar  $P_j$ .

- Una transición activada puede dispararse en cualquier momento.
- Como resultado de disparar una transición activada, se eliminan  $W(P_j, T_i)$  marcas de cada nodo  $P_j$  a la entrada de  $T_i$  y se añaden  $W(T_i, P_k)$  marcas a cada nodo  $P_k$  de salida de  $T_i$ .

Por lo anterior se consideró importante desarrollar una herramienta que permita comprender la evolución del marcado en una red de Petri, ecuación 5.

$$M_k = M_{k-1} + A^T \mu_k \quad k = 1, 2, 3, \dots, n \quad (5)$$

Donde:

- $M_k$  Estado que se desea conocer.
- $M_{k-1}$  Vector de marcas del estado actual del sistema.
- $A^T$  Matriz de incidencia.
- $\mu_k$  Vector de transiciones.
- $k$  Número entero positivo.

La matriz de incidencia de ecuación 5 obtiene a través de ecuación 6.

$$A^T = a_{ij}^+ - a_{ij}^- \quad (6)$$

Donde:

- $a_{ij}^+$  Matriz de peso de arcos que entran al lugar  $i$  desde la transición  $j$ .
- $a_{ij}^-$  Matriz de peso de arcos que salen del lugar  $i$  y llegan a la transición  $j$ .

El diagrama a bloques del simulador desarrollado se muestra en figura 1.

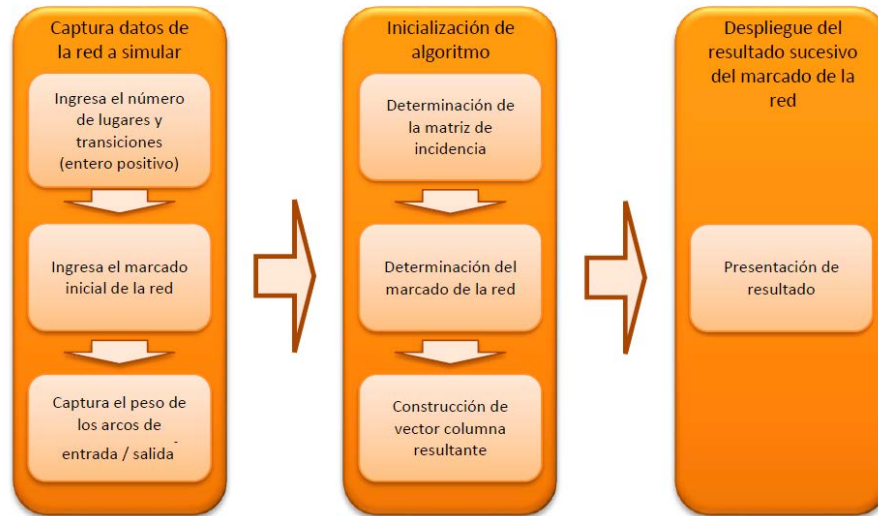


Figura 1 Diagrama a bloques del simulador desarrollado.

Antes de iniciar la operación del simulador, es necesario ingresar los datos de la red a simular, (ver figura 2). A y B son números enteros positivos obtenidos de la red de Petri que desea simular, C es un vector columna en el que debe de ingresar el marcado inicial de la red ( $M_0$ ), D y E son los pesos de las matrices de entrada y salida respectivamente, estos serán ingresados en forma de una matriz  $m \times n$  donde las filas (m) serán los lugares y las columnas (n) serán las transiciones existentes en la red.

```
Número de lugares  
P = (A)  
Número de Transiciones  
T = (B)  
Matriz Inicial  
M = (C)  
Peso de Arcos de entrada  
a = (D)  
Peso de arcos de salida  
b = (E)
```

Figura 2 Interfaz de captura de datos de la red de Petri deseada.

Al ejecutar la simulación, la evolución del marcado será presentada como se muestra (ver figura 3), en F el estudiante podrá observar el total de iteraciones en

los que la evolución del marcado de la red es completada y en G tendrá la oportunidad de observar cómo se llevó a cabo dicha evolución, esto es, el desplazamiento de los tokens de acuerdo al peso de los arcos y su ubicación, paso a paso, en los diferentes lugares que conformar la red de Petri que este desee simular.

```

Iteracion: 1
Iteracion: 2
Iteracion: 3
Iteracion: 4
Iteracion: 5
Iteracion: 6
Iteracion: 7
NumIter =
      7 (F)
Evolucion =
  4   3   3   2   1   1   0
  1   2   0   1   2   0   1
  0   0   1   1   1   2   2 (G)
    
```

Figura 3 Despliegado de la solución de la red de Petri simulada.

### 3. Resultados

Para verificar el funcionamiento del simulador desarrollado, se demostró la evolución del marcado de una red de Petri y con uso de una herramienta grafica de simulación de redes de alto nivel como lo es CPN Tools [3] se corroboró el resultado obtenido, la red a simular tendrá las siguientes características:

Sea una operación de producción (Ejemplo 1) que para obtener una unidad de producto terminado hace uso de dos tipos de materias primas llamadas MP1 y MP2, de MP1 se requiere una unidad y existen en total 5 unidades y de MP2 se requieren dos unidades y en total existen 10 unidades, estas son procesadas de manera simultánea por una máquina.

Con la información anterior y empleando el simulador gráfico podemos dibujar la red correspondiente (figura 4).

De la simulación presentada se observa que existen 5 lugares (P1 a P5) y 2 transiciones (T1 y T2), los arcos que conectan a los lugares P1 y P2 con T1 tienen pesos diferentes, en el caso de P1 el peso es de 1 y para P2 el peso es de 2, el

peso del resto de los arcos de la red es de 1, la cantidad de tokens en cada lugar está determinada de acuerdo a tabla 1.

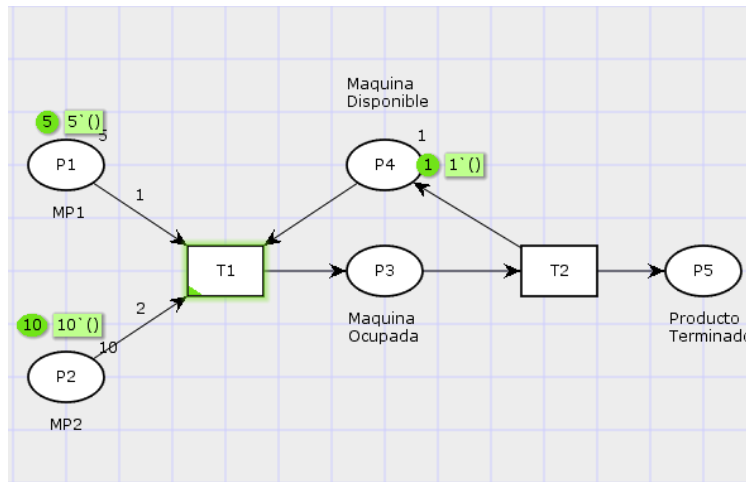


Figura 4 Simulación en CPN Tools de un proceso de producción.

Tabla 1 Relación de tokens asignados por lugar.

Lugar	Tokens
P1	5
P2	10
P3	0
P4	1
P5	0

Por lo que se puede deducir la matriz de tokens inicial como en la ecuación 7.

$$M_0 = \begin{pmatrix} 5 \\ 10 \\ 0 \\ 1 \\ 0 \end{pmatrix} \quad (7)$$

Para el caso de la matriz de incidencia se deben conocer  $aij^+$  y  $aij^-$ , por lo tanto, ecuación 8.

$$aij^+ = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix} \quad aij^- = \begin{pmatrix} 1 & 0 \\ 2 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 0 \end{pmatrix} \quad (8)$$

Cuando se ingresan los datos previamente obtenidos de la red a la pantalla de captura de red del simulador, estos quedan expresados de acuerdo a figura 5.

```

Número de lugares
P = 5
Número de Transiciones
T = 2
Matriz Inicial
M = [5 10 0 1 0]'
Peso de Arcos de entrada
a = [0 0; 0 0; 1 0; 0 1; 0 1]
Peso de arcos de salida
b = [1 0; 2 0; 0 1; 1 0; 0 0]
    
```

Figura 5 Captura de datos de la red de Petri en el simulador.

Cuando se inicializa el algoritmo del simulador, este determina las condiciones de inicialización de la red, así como los datos necesarios para llevar a cabo la evolución de la misma, por lo que la matriz de incidencia expresada en la ecuación 9 y la matriz de transiciones (Ec.10) son determinadas de manera automática.

$$A^T = \begin{pmatrix} -1 & 0 \\ -2 & 0 \\ 1 & -1 \\ -1 & 1 \\ 0 & 1 \end{pmatrix} \quad (9)$$

La matriz de transiciones  $\mu_k$ , inicializa en 0 y conforme se va realizando la evolución del marcado los valores de las transiciones se van alternando entre transición no activada (0) y transición activada (1), por lo que para este caso la matriz está dada por ecuación 10:

$$\mu_k = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (10)$$

El simulador lleva a cabo la solución de evolución del marcado vista previamente en la ecuación 5, por lo que, para el caso planteado, la evolución obtenida para el marcado ( $M_1$ ), estaría conformada por ecuaciones 11 y 12.

El resultado obtenido en  $M_1$  respecto a la evolución de los tokens se puede corroborar en la simulación realizada con el software CPN Tools (ver figura 6).



$$M_1 = \begin{pmatrix} 5 \\ 10 \\ 0 \\ 1 \\ 0 \end{pmatrix} + \begin{pmatrix} -1 & 0 \\ -2 & 0 \\ 1 & -1 \\ -1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad (11)$$

$$M_1 = \begin{pmatrix} 4 \\ 8 \\ 1 \\ 0 \\ 0 \end{pmatrix} \quad (12)$$

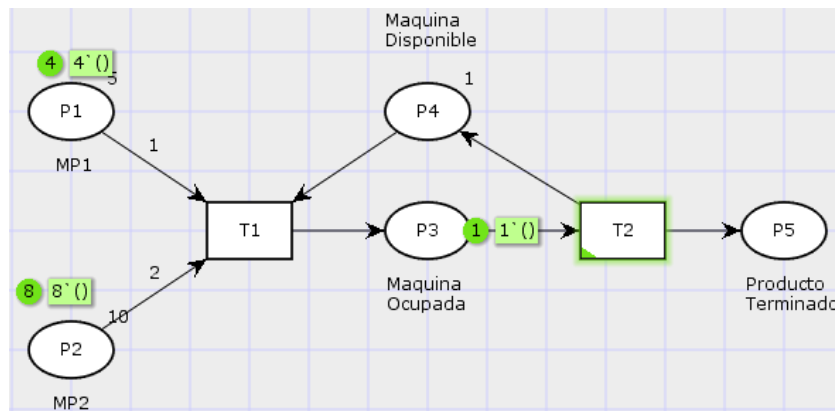


Figura 6 Evolución del marcado en la Red de Petri del ejemplo 1.

Se observa que hasta esta primera iteración no se ha conseguido producir una sola unidad de producto terminado, los tokens en los diferentes lugares de P1 a P4 se han modificado de acuerdo al peso asociado por cada arco, sin embargo, es necesario que el simulador lleve a cabo una nueva operación que permita continuar con la evolución de la red, por lo que la nueva operación a realizar es ecuación 13 y 14.

$$M_2 = \begin{pmatrix} 4 \\ 8 \\ 1 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} -1 & 0 \\ -2 & 0 \\ 1 & -1 \\ -1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad (13)$$

$$M_2 = \begin{pmatrix} 4 \\ 8 \\ 0 \\ 1 \\ 1 \end{pmatrix} \quad (14)$$

En esta iteración se puede observar que una unidad de producto terminado fue realizada, lo cual se corrobora en la simulación contenida (figura 7).

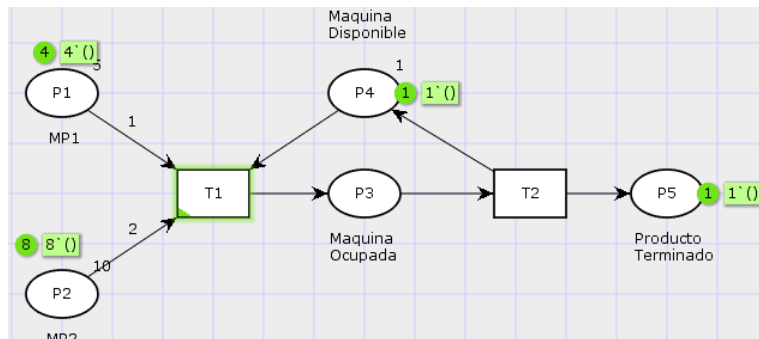


Figura 7 Evolución de la red de Petri y obtención de un producto terminado del ejemplo 1.

Se realizan las operaciones necesarias hasta obtener el total de productos terminados en el lugar 5, lo cual ocurre cuando ecuación 15:

$$M_{10} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 5 \end{pmatrix} \quad (15)$$

Lo que significaría que el total de materias primas fueron utilizadas para la elaboración de 5 unidades de producto terminado como lo podemos observar en el lugar 5 de la red de Petri.

El comportamiento de la misma red en el software CPN tools nos confirma la operación mencionada anteriormente (figura 8).

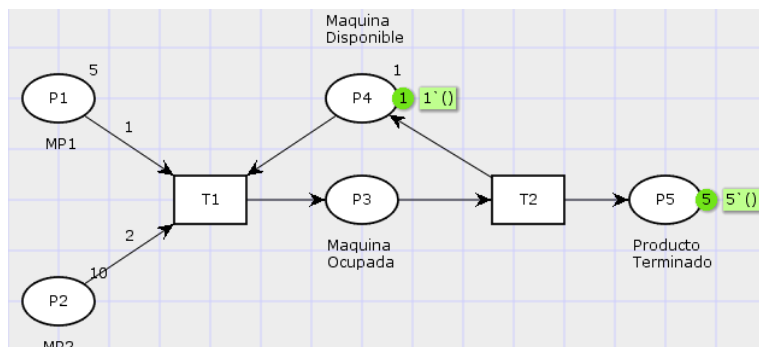


Figura 8 Evolución final de la red de Petri del ejemplo 1.

En la presentación del simulador desarrollado, se observa que los resultados obtenidos coinciden con lo plasmado gráficamente (figura 9), con lo que se puede asegurar que el funcionamiento de éste no presenta desviación alguna al comparar sus resultados con otro simulador, como se ha venido mostrando en las figuras 6 a 8.

```
Iteracion: 1
Iteracion: 2
Iteracion: 3
Iteracion: 4
Iteracion: 5
Iteracion: 6
Iteracion: 7
Iteracion: 8
Iteracion: 9
Iteracion: 10
NumIter =
    10
Evolucion =
    4  4  3  3  2  2  1  1  0  0
    8  8  6  6  4  4  2  2  0  0
    1  0  1  0  1  0  1  0  1  0
    0  1  0  1  0  1  0  1  0  1
    0  1  1  2  2  3  3  4  4  5
```

Figura 9 Evolución final de la red de Petri utilizando simulador desarrollado.

#### 4. Discusión

En [7,8,9] se presentan herramientas que permiten el aprendizaje y comprensión de redes de Petri, estas requieren de software especializado que no es de uso libre y no profundizaba en la composición de las ecuaciones matemáticas que permiten determinar la evolución del marcado, por lo que se decidió desarrollar un simulador didáctico que permite a los estudiantes profundizar en el uso de las operaciones del diseño de redes de Petri.

Se ha realizado una prueba con una muestra de 30 estudiantes de la asignatura Controladores Lógicos Programables del Instituto Tecnológico de Orizaba durante el semestre enero–junio 2016, en la cual se han obtenido diversos resultados, la mayoría aceptables por parte de los usuarios quienes han expresado que el uso de la herramienta les es de fácil comprensión y de ayuda para el aprendizaje de la evolución del marcado (figura 10).

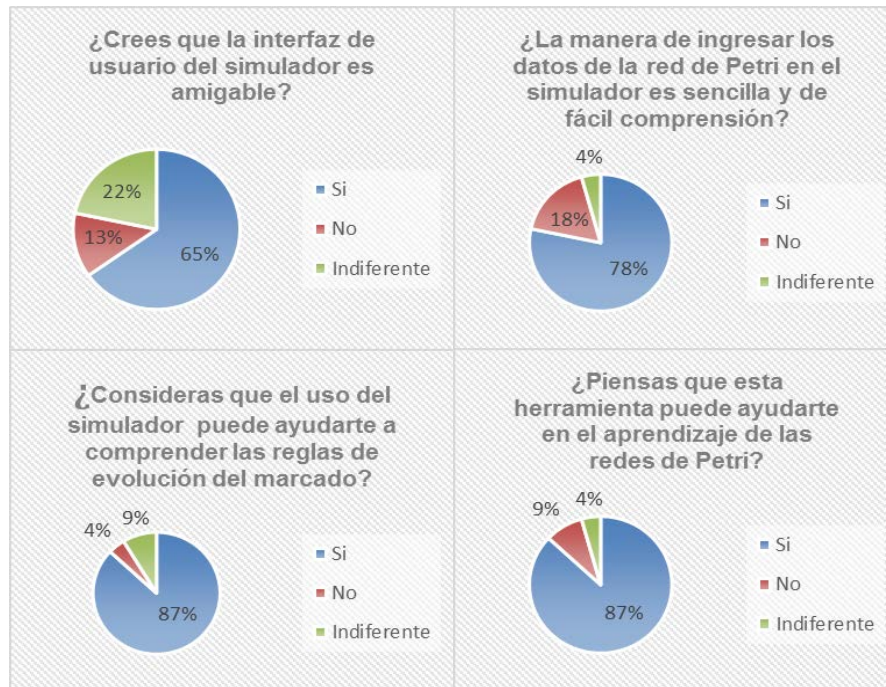


Figura 10 Resultados de la encuesta realizada a estudiantes del I.T.O.

## 5. Conclusiones

A pesar de que existen herramientas de simulación que permiten llevar a cabo el modelado en redes de Petri de procesos extensos y complejos, se deben de fincar las bases que posibiliten a los interesados estudiar sobre este tema comprender la formación del modelo matemático de una red, la correcta distribución del mercado inicial y la asignación del peso de los arcos en las matrices correspondientes, por lo que este simulador persigue el objetivo de que un principiante en el tema se involucre en el correcto llenado de las matrices de peso de arcos y comprenda la evolución del mercado de una red a modelar, los resultados obtenidos en el estudio realizado con estudiantes del I.T.O. muestra que esta herramienta puede ser de gran ayuda en el aprendizaje de redes de Petri, la muestra tomada en el estudio conoce sobre el tema y pudo aportar ideas para el mejor desarrollo del simulador, durante los siguientes semestres se continuará con la realización de pruebas con otros estudiantes para seguir enriqueciendo el simulador y poder obtener mejores resultados y mayor aceptación en la comunidad estudiantil.

## 6. Bibliografía y Referencias

- [1] T. Murata, "Petri Nets: Properties, Analysis and Applications", Proceedings of the IEEE. Vol. 77. No. 4. April 1989. Pp 541 – 580.
- [2] R. David, H. Alla, "Petri nets for modeling of dynamic systems: A survey", *Automatica*. Vol. 30. No. 2. February 1994. Pp 175 – 202.
- [3] J.S. Lee, P. L. Hsu, "An improved evaluation of ladder logic diagrams and Petri nets for the sequence controller design in manufacturing systems", *The International Journal of Advanced Manufacturing Technology*. Vol. 24. No. 3. 08-2004. Pp 279-287.
- [4] Petri Nets Tool Database Quick Overview, <http://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/quick.html>. May 2016.
- [5] F. A. Candelas, J. Sánchez, "Recursos Didácticos Basados en Internet Para el Apoyo de la Enseñanza de Materias del Área de Ingeniería de Sistemas y Automática", *Revista Iberoamericana de Automática e Informática Industrial*. ISSN 1697-7912. Vol. 2 Núm. 2. Abril 2005. Pp. 93-101.
- [6] L. R. Baquero, L. E. Argota, O. Rodríguez, "Método para el Modelado y Prueba de Diagramas de Actividades Mediante Redes de Petri". *Revista Latinoamericana de Ingeniería de Software*. ISSN 2314-2642, Vol. 3. Núm. 5. 2015. Pp. 206-212.
- [7] M. H. Matcovschi, C. Mahulea, C. Lefter, O. Pastravanu, "Petri Net Toolbox in Control Engineering Education". *Proc. IEEE Conference on Computer Aided Control Systems Design*. October 2006. Pp. 2298–2303.
- [8] G. Music, I. Hafner, F. Breiteneker, A. Körner, "Learning Petri Net Dynamics Through a Matlab Web Interface". 8<sup>th</sup> EUROSIM Congress on Modelling and Simulation. Cardiff. September 2013. Pp. 324-329.
- [9] J. Figueras i Jové, A. Guasch i Petit, P. Fonseca i Casas, J. Casanovas-Garcia, "Teaching System Modelling and Simulation Through Petri Nets and Arena". *Proceedings of the 2014 Winter Simulation Conference 2014*. December 2014. Pp. 3662-3673.

## **7. Autores**

Ing. Fernando Bautista Hernández, es ingeniero en electrónica por el Instituto Tecnológico de Orizaba.

Dr. Ignacio Herrera Aguilar, obtuvo su título de PhD en Sistemas Automáticos por la Université Paul Sabatier en el año 2007.

Dr. Oscar Osvaldo Sandoval González, obtuvo su título de PhD en Robótica en la Scuola Superiore Sant'Anna en el año 2010.

Dr. Gerardo Águila Rodríguez, obtuvo su título de Doctor en Ingeniería Electrónica en la SEES del CINVESTAV - I.P.N. en el año 2008.

Dr. José Pastor Rodríguez Jarquín, obtuvo su título de Doctor en Ciencias en Ingeniería Electrónica en el Centro Nacional de Investigación y Desarrollo Tecnológico en al año 2016.