

# Evaluación del Desempeño de Procesos Industriales Utilizando Redes de Petri Simuladas Bajo el Formalismo DEVS

## Performance Evaluation of Industrial Processes Using Petri Nets Simulated Under DEVS Formalism

Carlos A. Parra Ortega  
Dpto. EEST. Facultad de Ingenierías y Arquitectura  
Universidad de Pamplona, Ciudadela El Buque, Pamplona – Colombia  
carapa@unipamplona.edu.co

Recibido para revisión 21 de Abril de 2008, Aceptado 19 de Mayo de 2008, Versión final 24 de Mayo de 2008

**Resumen**—Un aspecto fundamental de los sistemas de producción es la medida de su desempeño. Para obtener esta medida, se recurre frecuentemente a modelos matemáticos tales como redes de Petri, con el fin de validar su comportamiento concurrente, presencia/ausencia de bloqueos, sincronización de actividades, entre otros aspectos. A pesar de sus ventajas, el uso de redes de Petri no permite evaluar otros aspectos que son importantes en los procesos de manufactura, por tal motivo se recurre a la simulación estocástica para obtener otros índices de desempeño. En este documento se utiliza un método para llevar una representación en redes de Petri a sistemas estocásticos a eventos discretos, y se presenta su aplicación a un ejemplo académico de manufactura.

**Palabras claves:** Sistemas de Manufactura, Redes de Petri, Simulación Estocástica, DEVS.

**Abstract**—A fundamental issue of production systems is the measuring of performance. In order to obtain such measure, mathematical models are used such as Petri nets to validate concurrent behavior, presence/absence of blockings, activity synchronization, among other aspects of this industrial process. Despite of its advantages, the use of Petri nets do not allow for evaluate other important issues in manufacturing processes, for this reason stochastic systems simulation is used to obtain other performance measures. In this document a method for map Petri nets to stochastic system is used, and is showed an academic manufacture example.

**Keywords:** Manufacturing Systems, Petri Nets, Stochastic simulation, DEVS.

### I. INTRODUCCIÓN

Este documento se ubica dentro de los estudios sobre Control y Supervisión de Sistemas Industriales, donde se plantea un método para evaluar el desempeño de los procesos de manufactura modelados con Redes de Petri utilizando simulación estocástica de sistemas a eventos discretos. Más en detalle, se utiliza una especificación común para modelar sistemas discretos, y se utiliza un mecanismo para proyectar la Red de Petri a una red estocástica, donde las etapas y recursos del proceso industrial pueden verse como un conjunto de componentes que intercambian información. Por medio de esta simulación, se podrían evaluar otros aspectos del sistema que no alcanzan a evaluarse utilizando Redes de Petri solamente. Uno de estos aspectos puede ser el tiempo libre que tiene un recurso sin utilizarse, otro aspecto podría ser el tiempo de permanencia en el sistema de un insumo, hasta que es transformado en producto, otro sería si se represan insumos a procesar en una máquina y así sucesivamente.

El Formalismo DEVS propuesto por Zeigler et al [1] es la especificación más general para describir sistemas discretos, abarcando los Grafsets, StateCharts, Redes de Petri y Redes de Colas, entre otros. Si se tiene un proceso industrial modelado en Redes de Petri, se puede llevar a una representación a otros sistemas de eventos discretos a partir de la especificación más general. Según Chacón et al [2], la automatización de sistemas de producción muestran dos aspectos muy importantes para optimizar y mejorar un proceso de producción: características físicas del sistema a automatizar, y el modo de operación del sistema automatizado. La metodología que será utilizada para efectuar la evaluación de la alternativa de automatización se puede resumir de la siguiente manera:

- Aplicación de una arquitectura de referencia para automatización.
- Obtención de un modelo del comportamiento del proceso.

- Simulación numérica de este modelo como un sistema de eventos discretos.

Este trabajo asume que el proceso industrial ya está enmarcado dentro de una arquitectura de automatización, de manera que se concentrará en la evaluación del desempeño del modelo de comportamiento del proceso utilizando simulación. Varios trabajos bajo diversos enfoques se han llevado a cabo, como los de DiCesare y Desroches [3] enfocados hacia redes de Petri y mediciones de salida, así como Cembellín [4], que plantea una medida de desempeño basada en procesos markovianos. Es conveniente destacar que muchas aplicaciones se han hecho para evaluar sistemas de comunicaciones modelados con redes de Petri, los cuales razones de espacio no se describen en detalle.

## II. PROBLEMAS ASOCIADOS A LA REPRESENTACIÓN DE PROCESOS INDUSTRIALES CON REDES DE PETRI

Las redes de Petri son una herramienta matemática que permite hacer modelado matemático procesos de diversa naturaleza, en particular, procesos de manufactura concurrentes sincrónicos y asincrónicos. Las redes de Petri son una herramienta muy útil para validar el comportamiento de un modelo con respecto a la sincronización de sus actividades y al diseño de respuestas ante la ocurrencia de eventos, pero es difícil evaluar otros aspectos, como por ejemplo, cuando ocurrirá cada evento exactamente, o cual es la eficiencia del sistema en cuanto a utilización de los recursos. La simulación de sistemas de eventos discretos implementando una lista de eventos futuros propuesta por Law y Kelton [5] ofrece un complemento para medir la eficiencia del sistema en cuanto a la utilización de equipamiento, represamiento de insumos o productos intermedios, probabilidad de ocurrencia de un evento en un tiempo dado, entre otros. Para explicar mejor la descripción anterior, es necesario apoyarse en un caso de estudio de un problema típico de manufactura.

Una definición formal de red de Petri establece que tiene cuatro elementos: un conjunto de lugares  $P$ , un conjunto de transiciones  $T$ , una función de entrada  $I$ , y una función de salida  $O$ , las cuales relacionan transiciones y lugares. Una definición formal se le debe a Peterson [6], que establece que una red de Petri se define como una cuadrupla  $C$ , siendo  $C = (P, T, I, O)$ .

Donde:

- $P = \{p_0, p_1, p_2, \dots, p_n\}$  conjunto de lugares,  $n > 0$ .
- $T = \{t_0, t_1, t_2, \dots, t_m\}$  conjunto de transiciones,  $m > 0$ .
- $I: T \rightarrow P^\infty$  es la función de entrada, que proyecta de transiciones a lugares.
- $O: P \rightarrow T^\infty$  un mapeo de transiciones a grupos de lugares.

Vale la pena destacar que existen otras definiciones formales equivalentes a la mencionada anteriormente, donde definen a la red de Petri desde el punto de vista matricial, de acuerdo al trabajo de Moody y Antsaklis [7], destinada a la síntesis de controladores de proceso.

La descripción de eventos discretos por redes de Petri se fundamenta en los conceptos de eventos y condiciones. Los eventos son asociaciones que ocurren en el sistema; y las condiciones son descripciones lógicas del estado de varias partes del sistema. Para que los eventos ocurran deben existir ciertas condiciones, llamadas precondiciones. Cuando ocurre un evento estas precondiciones generalmente cambian para dar lugar a otro conjunto de condiciones, llamadas post-condiciones. Así, las post-condiciones de un evento pueden ser precondiciones de otro evento.

Para construir una red de Petri el modelador debe estimar el dominio de interpretación del sistema físico etiquetando lugares y transiciones. Así, por medio de identificar los eventos y las condiciones del sistema, se puede modelar un sistema a eventos discretos de la siguiente manera:

1. Cada evento se representa y etiqueta como una transición.
2. Los lugares representan las condiciones.
3. Las precondiciones de un evento son los lugares de entrada a la transición de tal evento.
4. Las post-condiciones de un evento son los lugares de salida de la transición que representa tal evento.

Diversos investigadores han desarrollado metodologías para construir redes de Petri a partir de las especificaciones de un sistema de producción, destacándose el trabajo de Zhou y DiCesare [8] donde a partir de las especificaciones de un proceso de manufactura diseñan una red de Petri. Su propuesta se basa en la clasificación de los lugares de la RdP en tres tipos: lugares que representan el estado de una operación, lugares que representan entidades temporales, como lo son las piezas a procesar, y lugares que representan las entidades permanentes, como lo es el equipamiento. Charlotta Johnsson [9] propone una metodología para aplicar redes de Petri en el modelado de sistemas de producción por lotes. Para este trabajo, se asumirá que los procesos industriales a los cuales se les quiere evaluar desempeño están representados por medio de la metodología de Zhou y DiCesare.

En la figura 1 se muestra un proceso de producción simplificado, donde hay dos máquinas  $M1$  y  $M2$ , dos cintas transportadoras con una bandeja en cada una de ellas  $C1$  y  $C2$ . Las piezas a ser procesadas son de tipo  $A$  y  $B$ . El esquema se considera simplificado porque no se muestran detalles sobre como se proporcionarán las piezas a la máquina, o el mecanismo de coordinación entre máquina y la cinta. Obviamente existe una red de comunicación de campo, para transmitir señales y recibir comandos de los controladores. En la parte derecha de la figura 1 se puede observar la red de Petri asociada al proceso. La principal característica del sistema es que los recursos son compartidos (las dos máquinas), y bajo ciertas condiciones en la secuencia de producción, pueden operar simultáneamente. Los lugares  $p1$  y  $p2$ , por ejemplo, indican el estado en que se encuentra la máquina 1, mientras que los lugares  $p5$  y  $p6$  indican la posición del palet  $C1$ , que transporta piezas de tipo  $A$ .

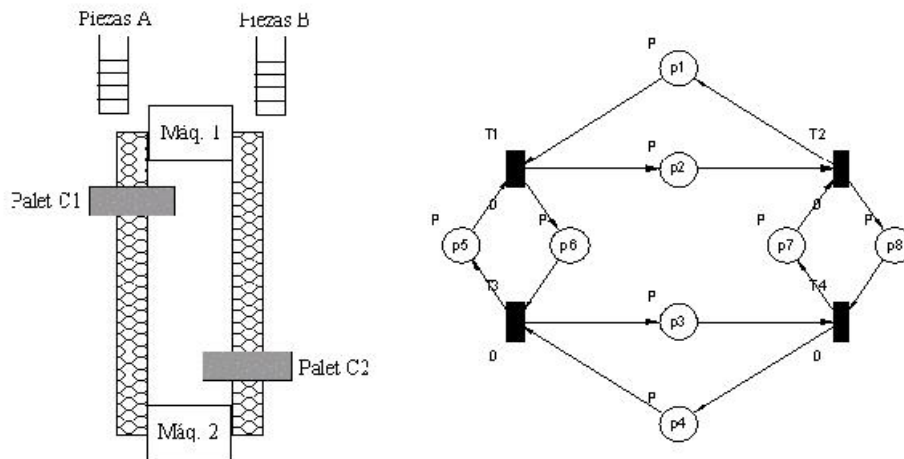


Figura 1. Representación de un proceso industrial sencillo utilizando redes de Petri

Existen muchos argumentos para estudiar sistemas dinámicos utilizando redes de Petri, algunos de ellos propuestos por Murata [10], ya que pueden proporcionar información sobre la estructura y comportamiento del sistema, y puede utilizarse esta información para evaluar el sistema modelado y sugerir mejoras o cambios. Las redes de Petri exhiben ciertas propiedades, que pueden agruparse en dos tipos: propiedades estructurales, que no dependen de la marcación inicial, y propiedades que dependen de la marcación inicial, las cuales se denominan propiedades de conducta. En la descripción de procesos industriales, es deseable que el comportamiento del proceso exhiba ciertas propiedades, las cuales se van a reflejar en las propiedades de la red de Petri. Según Dicesare y Desroches [3], estas propiedades deben ser: red de Petri limitada, viva, reversible, consistente y conservativa, lo que garantiza que el proceso industrial modelado por la red de Petri va a exhibir estas propiedades de la siguiente manera: si cumple la segunda propiedad indica que el sistema no se va a bloquear, en tanto que la tercera indica que el proceso se puede re-inicializar, puesto que hay una secuencia de transiciones que llevan la marcación a repetirse posteriormente.

Dada la descripción de un proceso industrial con redes de Petri, es necesario establecer mecanismos que permitan evaluar el desempeño del proceso. El análisis y evaluación de redes de Petri permiten describir como funciona el proceso, y a que posibles estados se puede llegar en un futuro, pero por sí solos no bastan para explicar el grado de utilización de un recurso o equipamiento determinado, la salida (throughput) de material producido, entre otros aspectos. Se han desarrollado trabajos con RdP temporizadas y cadenas de Markov, para evaluar aspectos como la medición de salida de productos, como el trabajo de Cembellín [4], por mencionar a uno de ellos. Por lo tanto, se requiere disponer de algún método que permita calcular estos indicadores, sin perder la representación del proceso industrial, para así complementar las propiedades de los procesos industriales.

### III. CONSIDERACIONES PARA EVALUAR DESEMPEÑO DE SISTEMAS DE MANUFACTURA UTILIZANDO SIMULACIÓN ESTOCÁSTICA

Los modelos que representan a sistemas de eventos discretos son dinámicos, estocásticos y discretos, donde las variables de estado cambian de valor en instantes no predecibles de tiempo. Estos instantes de tiempo corresponden a un evento, por lo tanto, según Guasch et al [11] “un evento se define como una acción instantánea que puede cambiar el estado del sistema”. La simulación de sistemas de eventos discretos por medio de una lista de eventos futuros [3] ofrece un complemento para medir eficiencia del sistema en cuanto a la utilización de equipamiento, represamiento de insumos, probabilidad de ocurrencia de un evento en un tiempo dado, y otros. Una especificación formal que cubre todos los aspectos de la simulación estocástica y las redes de Petri es el formalismo DEVS [1], este formalismo para modelado y simulación establece que un sistema complejo, como lo es un proceso industrial, puede considerarse como una red de componentes que intercambian mensajes, de acuerdo a una ocurrencia aleatoria de eventos. Los formalismos exhibirían la siguiente relación, que se puede apreciar en la figura 2.

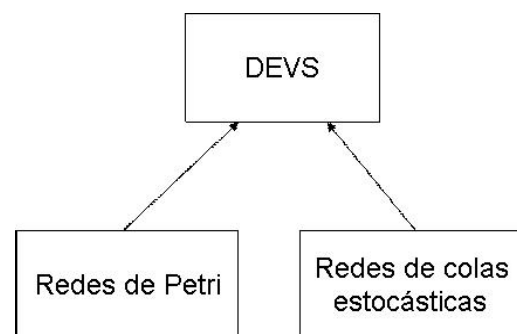


Figura 2. Diagrama que muestra la relación entre formalismos de modelado

Resumiendo a varios autores, los elementos de interés de los modelos de eventos discretos son:

- **Actividades:** tareas que se desarrollan en el sistema, delimitadas por dos eventos: uno que la inicia y uno que la termina.
- **Entidades:** son los objetos que circulan por el sistema. Pueden ser recursos que utiliza el sistema (permanentes) u objetos que se procesan en el sistema (temporales).
- **Atributos:** son las propiedades de los recursos y de los objetos temporales del sistema.

Estos elementos pueden representarse gráficamente, como una red de componentes que intercambian mensajes en eventos cuya ocurrencia se determina estocásticamente. Arena y Automod [12,13] son ejemplos de plataformas de simulación que han definido y extendido esta simbología hasta convertirla en un estándar.

**A. Generalidades de las redes de nodos que procesan eventos discretos de forma estocástica**

Existen muchas formas de implementar un sistema basado en DEVS para simular fenómenos de diversa naturaleza. Un ejemplo de lenguaje de simulación basado en DEVS es Glider [14], basado en la plataforma de simulación ARENA [12], donde se representa un sistema de eventos discreto por medio de una red de nodos intercambiando mensajes. Un resumen de las características de los tipos de nodos predefinidos se muestra en la tabla 1.

**Tabla 1.** Tipos de nodos que implementa el lenguaje de simulación Glider, basado en DEVS.

Nodo tipo	Símbolo	Función
Input		Generan mensajes que entran al sistema para ser procesados, enviándolos a otros nodos.
Line		Reorganiza una cola de eventos, según la disciplina que se quiera implementar.
Gate		Detiene o permite que los mensajes circulen por la red.
Resource		Simula los recursos utilizados por los mensajes, pueden representar también etapas en un proceso.
Decisión		Se utiliza para dirigir mensajes entre nodos (enrutamiento).
Exit		Destruye los mensajes. Simula las entidades que abandonan el sistema.
Continuous		Resuelve sistemas de ecuaciones diferenciales de primer orden.
Autonomous		De propósito general, maneja variables globales, envía mensajes.

**B. Proyección de la red de Petri hacia un sistema estocástico de eventos discretos**

Considerando que las redes de Petri son una herramienta útil para evaluar la lógica de procesos industriales concurrentes, y que las redes estocásticas de componentes permiten la evaluación del desempeño global del sistema, es necesario disponer de algún método que permita obtener tanto la lógica

del proceso como la medida del desempeño. Parra y Chacón [15] propusieron una metodología para llevar de una forma de representación a otra, de una manera sistemática utilizando el formalismo DEVS, y que se resume de la siguiente manera:

1. El punto de partida es la representación en redes de Petri del proceso industrial a evaluar, según la propuesta de Zhou y Dicesare.
2. Llevar esa representación formal hacia la especificación DEVS general. Uno o más lugares de la Red de Petri pueden corresponder a uno o más componentes DEVS atómicos.
3. De la especificación DEVS llevar a la representación de la red estocástica. Estos eventos generan mensajes que circulan a través de la red. Estos nodos pueden ser de varios tipos: entrada, recurso del sistema, salida del sistema, reguladores del paso de mensajes, y coordinador de las activaciones de los demás nodos así como sincronizar los mensajes que representan al flujo de insumos que se transforman para obtener un producto terminado.

La proyección de una Red de Petri a una Red de Nodos se puede resumir así:

- Lugar que representa a un insumo físico resulta en un nodo de tipo entrada (input).
- Equipamiento disponible resulta en un nodo con variables globales, que activa los eventos de los nodos de entrada.
- Las etapas del proceso resulta en un nodo de tipo recurso (resource).
- Los lugares que llevan a cabo la coordinación del proceso se proyectan hacia un nodo con variables globales, y uno o varios nodos que representan compuertas (gates) para regular el paso de mensajes.

Entre los beneficios de la simulación estocástica de procesos industriales descritos mediante redes de Petri y proyectados a DEVS se pueden destacar la medida de desempeño del proceso, en cuanto a los aspectos de utilización de recursos (el tiempo en que un recurso está trabajando en el proceso, respecto al tiempo total), throughput (cantidad de producto final obtenido con respecto a la cantidad de insumos procesados), eficiencia (cantidad de recursos usados y tiempo necesario para obtener una unidad de producto final) y costo de obtener una unidad de producto final, sumando costo de insumo y costo de operación del equipamiento. La evaluación del desempeño de estos procesos en cuanto a los aspectos antes mencionados brindará más elementos de juicio para tomar decisiones respecto al diseño de procesos industriales y la logística necesaria.

**IV. CASO DE ESTUDIO**

El caso particular de producción que se quiere analizar es un ejemplo académico, que consiste en una celda de producción cuyo equipamiento lo conforman dos robots con instrumentos manipuladores, y una mesa giratoria. Los insumos son los elementos A, B y C. El primer robot toma los elementos A y B y los ensambla para producir el subproducto AB, lo coloca

en la mesa, la cual gira y entonces el segundo robot toma los elementos AB y C, y los ensambla para producir ABC y luego lo coloca en la cinta transportadora. La mesa es un recurso compartido, y mientras un robot está trabajando sobre ella esta no debe girar, así que se requiere un mecanismo de coordinación, al igual que un mecanismo que no permita que circulen insumos de más en el proceso.

Un esquema simplificado que muestra el anterior proceso se muestra en la figura 3, se considera simplificado porque no se muestran detalles sobre como los robots percibirán la presencia o ausencia de piezas en su entorno, ni quien se las proporciona. Obviamente deben disponer de una red de comunicación de campo, para transmitir

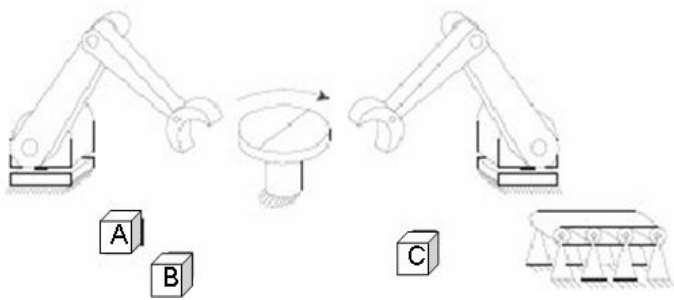


Figura 3. Esquema del proceso de fabricación.

Un diagrama de cadena de valor que muestra este proceso se puede apreciar en la figura 4. Obsérvese que el proceso inicia cuando están los insumos A y B disponibles, luego del primer proceso de ensamblado se obtiene el subproducto AB, que luego se le une el producto C, para un posterior ensamblado, finalmente se obtiene ABC y se distribuye, en este caso para una cinta transportadora.



Figura 4. Cadena de valor del proceso de manufactura de los dos robots.

La principal característica del sistema es que los recursos son compartidos (la mesa es un recurso utilizado por los dos brazos), y los brazos bajo ciertas condiciones en la secuencia de producción, operan a la vez. Esto presenta alguna dificultad para la obtención del modelo del proceso de producción.

Una descripción del proceso de manufactura, más en detalle, podría ser la siguiente:

- El brazo #1 toma un elemento A y lo coloca sobre su lado de la mesa.

- El brazo #1 toma un elemento B, lo coloca sobre su lado de la mesa y ensambla las piezas, obteniendo el subproducto AB.

- La mesa gira para que el subproducto quede del lado del brazo #2.

- El brazo #2 toma un elemento C, lo coloca sobre su lado de la mesa y ensambla para obtener el producto terminado ABC.

- El brazo #2 toma la pieza ABC ensamblada y la lleva a una cinta transportadora.

Este proceso tiene a su vez las siguientes restricciones para su funcionamiento:

- Ambos brazos sólo pueden tomar un elemento a la vez.

- El brazo #1 sólo coloca el elemento A, cuando su lado de la mesa está vacío.

- El brazo #1 sólo coloca el elemento B, luego de colocar el elemento A.

- La mesa solo gira cuando el robot #1 ensambló AB, y el lado del brazo #2 esta vacío.

- El brazo #2 sólo coloca el elemento C cuando la pieza AB está de su lado de la mesa.

#### A. Red de Petri del proceso industrial.

La red de Petri que representa al proceso de producción que se está analizando se muestra en la figura 5, y se obtuvo aplicando refinamientos sucesivos a partir de la red de Petri que describe al proceso global, cuyas etapas corresponden a la cadena de valor representada en la figura 4, y luego sustituyendo las transiciones por sub-redes, hasta obtener un modelo más detallado. Además de los lugares asignados a cada etapa del proceso y del estado de los recursos, se muestran lugares y transiciones adicionales para representar al mecanismo de control, los cuales se obtuvieron por medio de la aplicación de la teoría de Moody y Antsaklis sobre síntesis de supervisores [7], basada en resolver restricciones lineales para algunas marcaciones de la red, de manera que se obtengan lugares y transiciones “de holgura”, cuya función es prevenir que ocurran marcaciones no deseadas en la red. Obsérvese que la red se ha dividido en cuatro zonas: supervisor, robot 1, robot 2 y mesa. Es decir, cada elemento del proceso industrial está reflejado por un conjunto de lugares y transiciones de la red de Petri. Las interacciones entre ellos están representadas por los eventos que son habilitados por lugares y los que son activados por las transiciones. Incluso algunos lugares se encargan de inhibir la ocurrencia de ciertos eventos, como el supervisor impidiendo que la mesa gire cuando alguno de los robots esté ensamblando las piezas sobre ella.

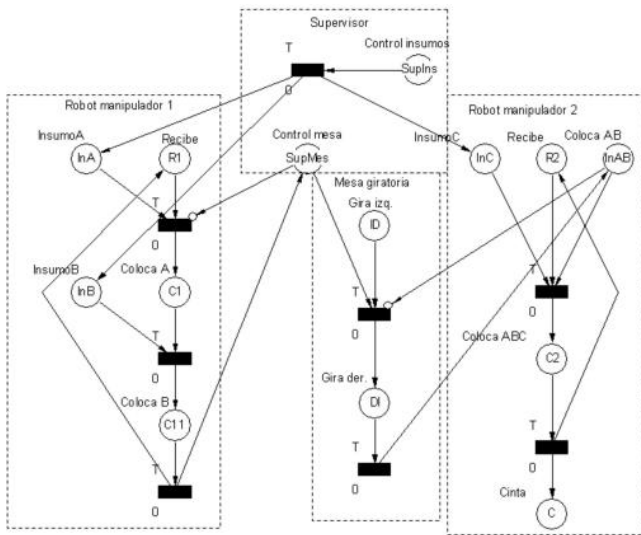


Figura 5. Red de Petri que representa al proceso de manufactura supervisado.

**Lugares y transiciones asociados al robot 1.** Los lugares indican el estado en que se encuentra el robot 1, y las transiciones indican la ocurrencia de eventos que modifican un estado, o tareas que debe efectuar el robot 1 para alcanzar un estado determinado. La siguiente tabla describe cada uno de ellos.

Tabla 2. Descripción de lugares y transiciones para el robot 1.

Nombre	Tipo	Descripción
R1	Lugar	Robot 1 dispuesto a manipular insumo A.
InA	Lugar	Insumo A en posición para ser tomado por el robot 1.
Ensambla	Transición	El robot 1 toma insumo A y lo coloca sobre la mesa
C1	Lugar	El robot 1 ha colocado insumo A sobre la mesa
InB	Lugar	Insumo B en posición para ser tomado por el robot 1.
T2	Transición	El robot 1 toma insumo B y lo coloca sobre la mesa
C11	Lugar	El robot 1 ha colocado insumo B sobre la mesa
Tm	Transición	El robot 1 ensambló los insumos A y B, obteniendo AB.

**Lugares y transiciones asociados al robot 2.** El robot 2 también tiene sus lugares y transiciones que indican en que estado se encuentra, que tareas debe ejecutar para cambiar su estado. En la tabla 3 se describen cada uno de ellos.

Tabla 3. Descripción de lugares y transiciones para el robot 2.

Nombre	Tipo	Descripción
InC	Lugar	Insumo C en posición para ser tomado por el robot 2.
InAB	Lugar	Subproducto AB en posición para ser tomado por el robot 2.
R2	Lugar	Robot 2 listo para manipular.
EnsamblaABC	Transición	El robot 2 toma insumo y subproducto y los ensambla
C2	Lugar	El robot 2 ha ensamblado producto ABC
T2	Transición	El robot 2 lleva producto ABC a la cinta transportadora
Cinta	Lugar	El robot 2 ha colocado ABC en la cinta transportadora

**Lugares y transiciones asociados a la mesa y al supervisor.** Aplicando las mismas definiciones para la mesa y para el supervisor, se resumen las características de los lugares y las transiciones para ambos.

Tabla 4. Descripción de lugares y transiciones relacionadas con la mesa.

Nombre	Tipo	Descripción
ID	Lugar	Subproducto AB se encuentra sobre la mesa.
Mid	Transición	Gira la mesa de izquierda a derecha, para llevar subproductos al robot 2.
DI	Lugar	Subproducto ABC está sobre la mesa.
Mdi	Transición	Gira la mesa de derecha a izquierda, para retornar a su estado inicial.

Tabla 5. Descripción de lugares y transiciones relacionadas con el supervisor.

Nombre	Tipo	Descripción
SupIns	Lugar	Cantidad de unidades de producto ABC a producir (equivalente a cantidad de unidades de insumos A, B y C a producir). Restricción: estos valores son iguales.
SupMes	Lugar	La mesa no se puede girar (inhibidor)
TP1	Transición	Envía insumos A, B y C a sus correspondientes posiciones para que la tomen los robots.

**B. Precondiciones para la ocurrencia de eventos, y condiciones que generan la ocurrencia de tales eventos**

Para que ocurra una transición, es necesario que se hayan alcanzado ciertos lugares en la red. A su vez, la ocurrencia de esta transición habilita que se activen otros lugares, pudiéndose determinar el estado del sistema hallando los lugares donde se encuentran ubicados los mensajes de sincronización (las fichas). La siguiente tabla establece condiciones previas para las transiciones, y los eventos que desencadena.

Tabla 6. Pre y Post condiciones para cada uno de las transiciones de la red de Petri.

Eventos	Pre-condiciones	Post-condiciones
Tp1	SupIns	InA, InB, InC
Ensambla	InA, R1, no SupMes	C1
T2	InB, C1	C11
Tm	C11	R1, ID
Mid	ID, SupMes	DI
Mdi	DI	InAB
ensamblaABC	InC, R2, InAB	C2
Tc	C2	C, R2

**C. El árbol de alcance**

La evolución del proceso puede seguirse a través de los cambios que ocurren en los lugares cuando ocurre una transición. Un método habitual para hacer seguimiento a esta evolución es el del árbol de alcance, que consiste en representar todos los lugares de la red como un vector, asignándoles el valor de 0 o 1 si está presente o no un mensaje o ficha. En ciertas ocasiones, puede tener valores mayores, si en un lugar confluyen varias fichas en la medida en que el sistema evoluciona. Los valores del vector cambian cada vez que ocurre una transición, ya que la ficha se mueve de un lugar a otro, y así el sistema va cambiando de estado. Este método facilita un seguimiento sistemático y su implementación por medio de algún lenguaje de programación de alto nivel. Incluso puede proyectarse a lenguajes de bajo nivel, y así obtener la implementación de un control supervisorio basado en redes de Petri. Para el caso de

estudio de los dos robots, el supervisor y la mesa, tenemos la siguiente estructura del vector de lugares:

[SupIns SupMes R1 InA C1 InB C11 ID DI R2 InAB InC C2 C]t

El estado inicial queda representado entonces por los siguientes valores:

[1 0 1 0 0 0 0 0 1 0 0 0 0]t

Que se interpretaría de la siguiente manera: el Supervisor va a producir una unidad de ABC, la mesa puede girar, el robot 1 está listo para trabajar, no hay insumo A, no hay insumo A en la mesa, no hay insumo B, no hay subproducto AB en la mesa, la mesa no ha girado, la mesa no ha retornado, el robot 2 está listo para trabajar, no hay insumo AB, no hay insumo C, los insumos y subproducto no están en la mesa, no hay producto ABC en la cinta transportadora. Cuando empiezan a ocurrir los eventos, el sistema evoluciona de acuerdo al siguiente diagrama, que se muestra en la figura 6.

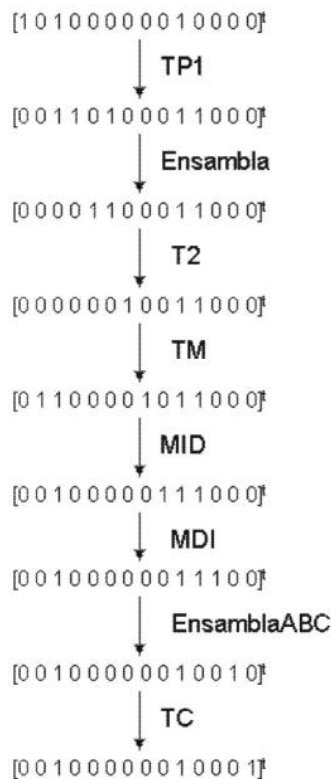


Figura 6. Árbol de alcance de la red de Petri que representa el proceso de ensamblaje de piezas, para una sola pieza.

Debido a la acción del mecanismo de control, algunos eventos se inhiben, y por lo tanto no se observa paralelismo cuando evoluciona el sistema, a menos que se produzca más de una unidad, lo que permite que el robot 1 quede habilitado para trabajar y con insumos cuando el robot 2 está ensamblando.

#### D. Proyección de la red de Petri a Componentes DEVS

Las redes de Petri son una herramienta para validar el comportamiento de un modelo con respecto a la sincronización

de sus actividades y al diseño de respuestas ante la ocurrencia de eventos, pero no establece una pauta sobre como se comportará el sistema en el tiempo (cuando ocurrirá cada evento, exactamente), ni tampoco establece una medida de eficiencia en el sistema. Por ejemplo, para el caso de estudio, una pregunta que podría hacerse es cuanto tiempo libre tiene cada brazo de robot, en términos porcentuales. La simulación de sistemas de eventos discretos por medio de una lista de eventos futuros [3] ofrece un complemento para medir eficiencia del sistema en cuanto a la utilización de equipamiento, represamiento de insumos, probabilidad de ocurrencia de un evento en un tiempo dado, y otros.

**Aplicación al caso de estudio.** Para proceder a realizar su simulación, es necesario entender y analizar el sistema. El desarrollo de las definiciones propuestas por Banks et Al [16] aparece a continuación, aplicándose al sistema objeto de estudio.

**Entidades.** Pueden clasificarse en dos grupos: recursos con que cuenta el sistema para desarrollar su misión, como lo son los dos robots y la mesa giratoria; y el material que circula por el sistema, en este caso los insumos A, B y C, subproducto AB y producto ABC.

**Atributos:** Se pueden considerar como atributos el tiempo promedio que le toma ensamblar las piezas. Para el robot 1, podemos considerar el tiempo que toma ensamblar las piezas A y B como una variable que se distribuye como una variable aleatoria gaussiana (normal truncada, es decir, no produce valores negativos), con una media y desviación determinada. Lo mismo podemos considerar para el robot 2, el cual tiene que ensamblar el subproducto AB con la pieza C. Para los insumos y subproductos que están circulando, el número de orden es importante, ya que el insumo A #1 solo puede ensamblarse con el insumo B #1, para dar origen al subproducto AB #1, el cual solo se puede ensamblar con el insumo C #1.

**Actividades:** Toda acción que represente un período de tiempo es una actividad. El proceso de ensamblado de los insumos A y B, la colocación de los insumos A y B en la mesa, del lado del robot 1, la colocación de insumos C y subproducto AB en la mesa, del lado del robot 2 y el ensamble de los insumos C con AB son las actividades que ocurren en este sistema.

**Eventos:** los eventos considerados son: cada vez que llega un insumo A o B a la mesa, previa orden del supervisor. El proceso de ensamble de A y B, el giro de la mesa para colocar AB en el lado del robot 2, el proceso de ensamble de AB con C, y la colocación del insumo C en la mesa.

**Variables de estado:** la cantidad de producto terminado colocado en la cinta transportadora, el estado de los robots: desocupado, ensamblando. El estado de la mesa: puede girar o no puede girar. Estas variables se pueden proyectar hacia el vector que contiene los lugares de la red de Petri.

Una vez determinados los elementos del sistema, y basándose en la Red de Petri de la figura 5, se diseñó una red de nodos que representan a las entidades mencionadas previamente,

donde cada nodo representa o bien una entidad o un evento. Para obtener la red de nodos, se aplicaron los pasos que se describieron en la sección anterior, y se muestra el diagrama de la red en figura 7.

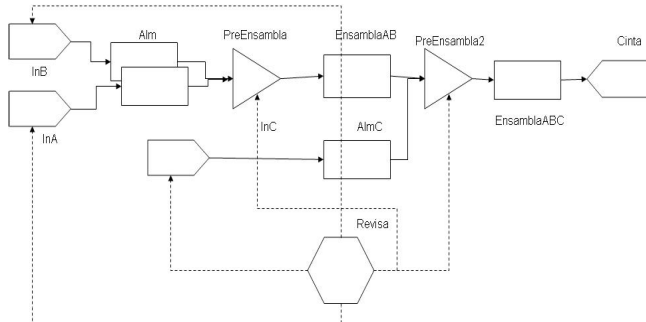


Figura 7. Red de nodos para simular el problema de los dos robots.

**Descripción de los nodos.**

- InA: es un nodo de tipo Input, el mensaje que genera representa al insumo A cuando se da la orden de que se coloque en la mesa para que trabaje el robot 1.
- InB: es un nodo de tipo Input, el mensaje que genera representa al insumo B cuando se da la orden de que se coloque en la mesa para que trabaje el robot 2.
- Alm: es un nodo de tipo Resource, tiene capacidad 2 y representa la puesta en la mesa de los insumos A y/o B, del lado más cercano al robot1.
- PreEnsambla: nodo de tipo Gate, simula la sincronización de los insumos A y B para su ensamblado. El robot 1 toma ambos insumos.
- EnsamblaAB: nodo de tipo Resource, simula el proceso de ensamblado de los insumos A y B para obtener el producto intermedio AB. Tiene una duración que se distribuye estadísticamente como una normal truncada.
- InC: es un nodo de tipo Input, el mensaje que genera representa al insumo C, cuando se da la orden de que se coloque en la mesa para que la trabaje el robot 2.
- AlmC: es un nodo de tipo Resource, y representa la puesta en la mesa del insumo C, más cercano al robot 2.
- PreEnsambla2: nodo de tipo Gate, simula la sincronización de los insumos AB y C para su ensamblado. El mensaje que representa al insumo C ya se ha colocado en la mesa, proveniente del nodo AlmC, y se ubica junto con el producto intermedio AB, que proviene del nodo EnsamblaAB.
- EnsamblaABC: nodo de tipo Resource, simula el proceso de ensamblado de los insumos AB y C para obtener el producto final ABC. La duración de este proceso se distribuye como una variable normal truncada.
- Cinta: nodo de tipo Exit, los mensajes que llegan a este nodo representan la puesta del producto ABC en la cinta transportadora para su despacho.

- Revisa: es un nodo de tipo Autónomo, que en este caso representa a un objeto que está chequeando el proceso y permitiendo/inhabilitando el paso de insumos, a manera de un supervisor.

**E. Simulación estocástica del sistema a eventos discretos**

El diagrama de la red de componentes estocástica expuesto en la figura anterior se codificó con el lenguaje Glider [14], desarrollado en la Universidad de los Andes de Mérida (Venezuela), el cual fue desarrollado para simular sistemas de eventos discretos y continuos, como una agrupación de subsistemas que intercambian información de diferentes maneras, frecuentemente como paso de mensajes entre nodos. Las siguientes condiciones iniciales y parámetros del sistema se mencionan a continuación. El modelo se programó para solicitarle al usuario cuantas unidades de los insumos A, B y C estaban disponibles para el proceso de manufactura. De la simulación se obtuvieron una traza y tres resúmenes de resultados.

**Análisis de trazas.** Se realizó una traza corta, donde se manufacturaron un total de seis piezas. La siguiente tabla muestra un fragmento de la misma, y luego se resumen todos los eventos que ocurrieron, ordenados por tiempo.

Tabla 6. Fragmento de una traza de simulación

```

EXAM.LIST IL_ALMC 1 SEEK EXIT TIME 6.056
EXT.MESS.# 1 GEN INC FROM IL_ALMC 1 LL 1 TIME 6.056
ADD MESS.# 1 GEN INC TO EL_PREENSAMBLA2 LL 1 TIME 6.056
2*****XXXX*****
NODE PREENSAMBLA2[1]
EXA.MESS.# 1 GEN INB
SEEK TO SYNCHRO 2 WITH CONDITION 2
EXA.MESS.# 1 GEN INB
2 RECORDS SYNCHRONIZED
EXA.MESS.# 1 GEN INB
EXT.MESS.# 1 GEN INB FROM EL_PREENSAMBLA2 LL 2 TIME 6.056
ADD MESS.# 1 GEN INB TO EL_ENSAMBLAABC LL 0 TIME 6.056
1*****XXXX*****
EXT.MESS.# 1 GEN INC FROM EL_PREENSAMBLA2 LL 1 TIME 6.056
DELETED MESS# 1 GENERATED IN INB AT 1.000
DELETED MESS# 1 GENERATED IN INB AT 1.000
NODE ENSAMBLAABC[1]
EXA.MESS.# 1 GEN INB
EXT.MESS.# 1 GEN INB FROM EL_ENSAMBLAABC LL 1 TIME 6.056
ADD TO FEL EV 9 IEV 1 TIME 8.255
ADD MESS.# 1 GEN INB TO IL_ENSAMBLAABC LL 0 TIME 6.056
1*****XXXX*****
    
```

La tabla 7 muestra un resumen de los eventos que ocurrieron, hasta el tiempo 50.



**Tabla 7.** Trazo de eventos resumida

Tiempo	Evento
0	Activación de paso de insumos A y B
1	Insumo A habilitado
1	Insumo B habilitado
1	Revisión del sistema
1.997	El robot1 toma el insumo B
2	Revisión del sistema
2.166	El robot1 toma el insumo A
3	Revisión del sistema
3.845	Se ensambló y se obtuvo AB y la mesa giró, el robot 2 tomó AB
4	Revisión del sistema
5	Insumo C habilitado
5	Revisión del sistema
6	Revisión del sistema
6.056	El robot2 toma el insumo C
7	Revisión del sistema
8	Revisión del sistema
8.255	Se ensambló y se obtuvo ABC, el robot2 colocó en la cinta a AB.
9	Revisión del sistema
10	Insumo A habilitado
10	Insumo B habilitado
10	Revisión del sistema
10.979	El robot1 toma el insumo B
11.050	El robot1 toma el insumo A
.....	
49	Revisión del sistema
49.142	Se ensambló y se obtuvo ABC, el robot2 colocó en la cinta a AB.
50	Revisión del sistema....

**Análisis de resultados.** Para las posteriores simulaciones, se hicieron ocho replicaciones, donde se simuló el ensamblado de 25 unidades. Una salida típica arrojada por el lenguaje de simulación GLIDER tiene la siguiente forma, mostrando resultados de simulación de la cuarta replicación en la tabla 8:

**Tabla 8.** Salida de GLIDER mostrando resumen de una replicación.

**Basic Experiment**

Time 208.00 Time Stat. 208.00 Replication 4 4/4/2008 8h 37m 48s

Elapsed time 0h 0m 10.500s

Nod/Ind Ant/Li #Ent Lgth Max Mean Dev MaxSt MeanSt Dev T.Free

INA

1 # Gen.	25								
INB									
1 # Gen.	25								
ALM									
1 EL	25	0	1	0.0	0.0	0.0	0.0	0.0	208.000
IL	25	0	1	0.12147	0.32667	1.19882	0.97505	0.28400	182.734
2 EL	25	0	1	0.0	0.0	0.0	0.0	0.0	208.000
IL	25	0	1	0.12150	0.32671	1.18857	0.96334	0.32892	182.728
PREENSAMBLA									
1 EL	50	0	2	0.01853	0.13486	0.34835	0.07709	0.10276	204.146
ENSAMBLAAB									
1 EL	25	0	1	0.0	0.0	0.0	0.0	0.0	208.000
IL	25	0	1	0.18554	0.38873	1.91368	1.49072	0.44731	169.409
INC									
1 # Gen.	25								
ALMC									
1 EL	25	0	1	0.0	0.0	0.0	0.0	0.0	208.000
IL	25	0	1	0.11909	0.32390	1.09009	0.95268	0.27753	183.229
PREENSAMBLA2									
1 EL	50	0	2	0.28358	0.45073	2.85259	1.17967	1.18888	149.016
ENSAMBLAABC									
1 EL	25	0	1	0.0	0.0	0.0	0.0	0.0	208.000
IL	25	0	1	0.22585	0.41814	2.41087	1.79383	0.61937	161.022
SALEABC									
1 EL	25	0	1	0.0	0.0	0.0	0.0	0.0	208.000
Time in System: Mean 6.86996 Dev. 0.25567 Max. 7.32155 Min. 6.18712									
REVISAR									
Var/Ind									
U_ALM 1	1			0.12147	0.32667	1.00000	0.0	0.50000	0.50000
U_ALM 2	2			0.12150	0.32671	1.00000	0.0	0.50000	0.50000
U_ALMC 1	1			0.11909	0.32390	1.00000	0.0	0.50000	0.50000
U_ENSAMBLAAB 1	1			0.18554	0.38873	1.00000	0.0	0.50000	0.50000
U_ENSAMBLAABC 1	1			0.22585	0.41814	1.00000	0.0	0.50000	0.50000

Revisando las replicaciones y promediando sus datos, observamos que producir 25 unidades toma 213 unidades de tiempo como promedio, no se presentan colas en ninguno de los recursos, y esto se debe a que el mecanismo de control no habilita paso de nuevos insumos hasta que la mesa esté

disponible. El robot 1 tiene una utilización del 41.27% como promedio, y esto se debe a que debe esperar a que los insumos le lleguen, a tenerlos a ambos en la mesa, y luego esperara que el robot 2 complete su tarea. Por su parte, el robot 2 se ocupa un 35.39% del tiempo, por la misma razón. Un proceso típico desde los insumos A y B hasta obtener el producto final toma como promedio 6.32 unidades de tiempo. La tabla 9 muestra un resumen y promedio global de las diferentes replications, midiendo tiempo de simulación, tiempo de proceso por pieza y porcentaje de uso de ambos robots.

**Tabla 9.** Resumen de resultados de las ocho replications.

Replicación	Tiempo de simulación	Tiempo de Proceso	Desviación Estándar	Uso de robot1	Uso de robot 2
1	212	6,26234	0,374646	41,20%	35,58%
2	219	6,86456	0,438244	40,00%	36,06%
3	216	6,28603	0,408356	40,09%	35,84%
4	208	6,17562	0,391426	42,85%	34,49%
5	213	6,29963	0,406358	41,46%	35,76%
6	216	6,28185	0,401472	40,50%	35,31%
7	212	6,17433	0,395878	41,45%	34,49%
8	208	6,2377	0,394366	42,58%	35,62%
<b>Promedio</b>	<b>213</b>	<b>6,3227575</b>	<b>0,40134325</b>	<b>41,27%</b>	<b>35,39%</b>

## V. CONCLUSIONES

Durante el proceso de plantear el problema, estudiar los formalismos de redes de Petri, DEVS y simulación estocástica, así como la experimentación y análisis de resultados, se obtuvieron algunas experiencias que se enuncian en las siguientes conclusiones:

- Se realizó la evaluación de desempeño de un proceso industrial combinando dos formalismos: redes de Petri y sistemas estocásticos basados en DEVS, lo que posibilita disponer de un mecanismo de optimización para procesos de producción que incorporan cualidades deseables a partir de su representación en redes de Petri.

- A pesar de que existen herramientas matemáticas para validar estructura y conducta de redes de Petri, requiere complementarse con simulación estocástica medir su eficiencia. En particular, para medir el grado de utilización de los recursos dentro de un sistema, el tiempo de proceso total para los insumos que se transforman a lo largo de proceso industrial. Lo que permite una optimización posterior del proceso de manufactura.

- La medición del grado de uso de recursos, el tiempo de proceso de insumos, los costos asociados a la operación de máquinas e insumos y la eficiencia de mecanismos de control son indicadores del desempeño de un proceso industrial. Por tal motivo se requiere conocer como será el comportamiento de un proceso antes de su implantación física, y la simulación es una herramienta adecuada para validar el diseño de procesos de manufactura, puesto que brinda capacidades de experimentación a partir de modelos matemáticos que describen a los elementos de un sistema de manufactura y las relaciones entre ellos.

- Para profundizar más en este tipo de trabajos se requiere estudiar los procesos continuos de producción,

los cuales tienen un modelado más complejo, pero que se puede enriquecer combinando diversos formalismos para tratar variables continuas del proceso físico, y etapas lógicas modeladas con redes de Petri.

- Con el fin de proporcionar mayor robustez al modelado de procesos con redes de Petri, se recomienda extender este modelado con el uso de redes de Petri coloreadas, las cuales permiten trabajar además con modelos estocásticos y existen varias herramientas computacionales para su modelado, simulación y análisis matemático.

## REFERENCIAS

- [1] Zeigler B., Praehofer H., Kim T. G. (2000), Theory of Modeling and Simulation. 2nd. Edition. New York: Academic Press.
- [2] Chacón E., Besemmel I., Hennem J. (2001), Automatización Industrial. Mérida: Universidad de Los Andes.
- [3] DiCesare F., Desroches A. (1991), Modeling, Control and Performance Analysis of automated systems using Petri Nets. En: Control and Dynamic Systems, Vol. 47, p. 121-172.
- [4] Cembellín A. (2003), Simulación de sistemas de control híbrido. En: Memorias de la XXIV Jornadas de Automática, comité español de automática.
- [5] Law A., Kelton D. (2000), Simulation Modeling and Analysis. McGraw-Hill series in Industrial Engineering and Management Science.
- [6] Peterson J. (1981), Petri Net Theory and the modeling of systems. Englewood cliffs: Prentice-Hall International.
- [7] Moody J., Antsaklis P. (1998), Supervisory Control of Discrete Event Systems using Petri Nets. Kluwer Academic Publishers.
- [8] Zhou M., DiCesare F. (1993), Petri Net Synthesis for Discrete Event Control of Manufacturing Systems. Kluwer Academic Publishers.
- [9] Johnsson C. (1999), A Graphical Language for Batch Control. PhD. Thesis. Lünd Institute of Technology. Sweden.
- [10] Murata T. (1989), Petri nets, properties, analysis and applications. En: Proceedings of the IEEE, Vol. 77.
- [11] Guasch A., Piera M., Casanovas J., Figueras J. (2005), Modelado y Simulación. Aplicación a procesos logísticos de fabricación y servicios. Ciudad de México: Alfaomega.
- [12] Kelton D., Sadowski P., Sturrock T. (2003), Simulation with ARENA. McGraw-Hill.
- [13] Banks J., Bountiful T. (2000), Getting started with Automod. Brooks Automation. Autosimulation Division.
- [14] CESIMO & IEAC. (1996), GLIDER reference manual. Mérida: Universidad de Los Andes.
- [15] Parra C., Chacón E. (2006), Evaluación del desempeño de procesos de manufactura proyectando redes de Petri hacia redes de Eventos Discretos. En: Proceedings of the XII Latin-American Congress on Automatic Control.
- [16] Banks J., Carson J., Nelson B., Nicol D. (2000), Discrete-event system simulation. Upper saddle river: Prentice-Hall.

**Carlos Arturo Parra Ortega:** nació en Medellín, Colombia, en 1969. Obtuvo el grado de Ingeniero de Sistemas de la Universidad Industrial de Santander, Bucaramanga, en 1993. La maestría en informática la obtuvo en 1997, en la Universidad Industrial de Santander. Actualmente es estudiante de doctorado en la Universidad de Los Andes en Mérida, Venezuela. Es profesor asistente en la Universidad de Pamplona, Colombia. Su área de investigación está relacionada con modelado y simulación, automatización industrial y sistemas multi-agente.