

**MODELAMIENTO DE SISTEMAS DE CONTROL Y SINCRONIZACIÓN DE  
PROCESOS BASADOS EN REDES DE PETRI**

**MICHEL ANDRÉS MUÑOZ RODRÍGUEZ CÓDIGO: 0004006**

**GIANY GENARO RICCIO MOLINARES CÓDIGO: 0004030**

**UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR  
FACULTAD DE INGENIERÍAS  
DIRECCIÓN DE PROGRAMAS ELÉCTRICA Y ELECTRÓNICA  
MINOR EN AUTOMATIZACIÓN INDUSTRIAL  
CARTAGENA DE INDIAS D.T. Y C.**

**2005**

**MODELAMIENTO DE SISTEMAS DE CONTROL Y SINCRONIZACIÓN DE  
PROCESOS BASADOS EN REDES DE PETRI**

**MICHEL ANDRÉS MUÑOZ RODRÍGUEZ  
GIANY GENARO RICCIO MOLINARES**

**Monografía presentada como requisito para la aprobación del Minor en  
Automatización Industrial**

**Director**

**Eduardo Gómez**

**MAGÍSTER CIENCIAS COMPUTACIONALES**

**UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR  
FACULTAD DE INGENIERÍAS  
DIRECCIÓN DE PROGRAMAS ELÉCTRICA Y ELECTRÓNICA  
MINOR EN AUTOMATIZACIÓN INDUSTRIAL  
CARTAGENA DE INDIAS D.T. Y C.**

**2005**

NOTA DE ACEPTACIÓN

---

---

---

---

---

Firma del presidente del jurado

---

Firma del jurado

---

Firma del jurado

Cartagena, Octubre 21 del 2005

Cartagena, Octubre 21 del 2005

Señores

**COMITÉ DE REVISIÓN DE MONOGRAFÍA**  
**UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR**  
LA CIUDAD

Apreciados señores:

Por medio de la presente nos permitimos informarles que la monografía titulada **“Modelamiento De Sistemas De Control Y Sincronización De Procesos Basados En Redes De Petri”** ha sido desarrollada de acuerdo a los objetivos establecidos.

Como autores del proyecto consideramos que el trabajo es satisfactorio y amerita ser presentado para su evaluación.

Atentamente,

---

Michel Andrés Muñoz

Código 0004006

---

Giany Riccio Molinares

Código 0004006

Cartagena, Octubre 21 del 2005

Señores

**COMITÉ DE REVISIÓN DE MONOGRAFÍA**  
**UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR**  
LA CIUDAD

Apreciados señores:

Por medio de la presente nos permitimos informarles que la monografía titulada **“Modelamiento De Sistemas De Control Y Sincronización De Procesos Basados En Redes De Petri”** ha sido desarrollada de acuerdo a los objetivos establecidos.

Como director considero que el trabajo es satisfactorio y amerita ser presentado para su evaluación.

Atentamente,

---

Eduardo Gómez Vásquez  
INGENIERO ELECTRICISTA  
MAGÍSTER EN CIENCIAS COMPUTACIONALES MSC.

## AUTORIZACIÓN

Cartagena de Indias D.T. Y C.

Octubre 21 de 2005

Yo Michel Andrés Muñoz Rodríguez identificado con la cedula de ciudadanía numero 73.192.446 de la ciudad de Cartagena. Autorizo a la Universidad Tecnológica de Bolívar a hacer uso de mi trabajo de grado y publicarlo en el catalogo ON LINE de la biblioteca.

LA SUSCRITA NOTARIA QUINTA DEL CIRCULO  
DE CARTAGENA  
15 DIC. 2005  
MICHEL ANDRÉS MUÑOZ RODRÍGUEZ  
C.C. 73192446  
Y QUI (PERSONA) (2) (3) (4) (5) (6) (7) (8) (9) (10) (11) (12) (13) (14) (15) (16) (17) (18) (19) (20) (21) (22) (23) (24) (25) (26) (27) (28) (29) (30) (31) (32) (33) (34) (35) (36) (37) (38) (39) (40) (41) (42) (43) (44) (45) (46) (47) (48) (49) (50) (51) (52) (53) (54) (55) (56) (57) (58) (59) (60) (61) (62) (63) (64) (65) (66) (67) (68) (69) (70) (71) (72) (73) (74) (75) (76) (77) (78) (79) (80) (81) (82) (83) (84) (85) (86) (87) (88) (89) (90) (91) (92) (93) (94) (95) (96) (97) (98) (99) (100)  
DECLARANTE: Michel Andrés Muñoz



MICHEL ANDRÉS MUÑOZ RODRÍGUEZ

## **ARTICULO 105**

La Universidad Tecnológica de Bolívar, se reserva el derecho de propiedad intelectual de todos los trabajos de grados aprobados, y no se pueden ser explotados comercialmente sin previa autorización.

## **AGRADECIMIENTOS**

Los autores expresan su agradecimiento a:

A **DIOS**, por permitirnos finalizar con éxitos estos años de estudio, a nuestros Padres y Hermanos por su apoyo incondicional.

**MICHEL ANDRÉS MUÑOZ RODRÍGUEZ**

---

**GIANY GENARO RICCIO MOLINARES**

---



**LISTA DE FIGURAS****LISTA DE TABLAS****GLOSARIO****RESUMEN****INTRODUCCIÓN****1. FUNDAMENTOS** **21**

<b>1.1. LUGARES DE ENTRADA Y DE SALIDA</b>	<b>22</b>
<b>1.2. TIPOS DE NODOS DE LAS RDP</b>	<b>23</b>
<b>1.3. REDES DE PETRI ORDINARIAS Y GENERALIZADAS</b>	<b>24</b>
<b>1.4. MARCADO DE UNA RED DE PETRI</b>	<b>25</b>
<b>1.5. DISPARO DE UNA TRANSICIÓN</b>	<b>26</b>
<b>1.6. TIPOS DE REDES DE PETRI</b>	<b>27</b>
<b>1.7. PROPIEDADES DE LAS REDES DE PETRI</b>	<b>27</b>
1.7.1. <i>EL MARCADO</i>	27
<b>1.8. ECUACIÓN FUNDAMENTAL DE LAS REDES DE PETRI</b>	<b>29</b>
1.8.1. <i>VECTOR CARACTERÍSTICO</i>	29
1.8.2. <i>MATRIZ DE INCIDENCIA</i>	29
1.8.4. <i>RED DE PETRI LIMITADA</i>	31
1.8.5. <i>RED DE PETRI CONFORME</i>	32
<b>1.9. ARCOS INHIBIDORES</b>	<b>33</b>
<b>1.10. REDES DE PETRI NO – AUTÓNOMAS</b>	<b>34</b>
1.10.1 <i>REDES DE PETRI SINCRONIZADAS</i>	34
1.10.2. <i>REDES DE PETRI TEMPORIZADAS</i>	35
1.10.3. <i>REDES DE PETRI P - TEMPORIZADAS</i>	37
1.10.4 <i>REDES DE PETRI T – TEMPORIZADAS</i>	38
1.10.5. <i>MODELOS DE RDP PARA SISTEMAS DE COLAS</i>	38
<b>1.11. EJEMPLO DE APLICACIÓN I: (LÍNEA DE TRANSPORTE DE TRES ETAPAS)</b>	<b>41</b>
<b>1.12. INTERPRETACIÓN DE LAS REDES DE PETRI</b>	<b>44</b>

<b><u>2. MODELAMIENTO Y CONTROL DE SISTEMAS DE EVENTOS DISCRETOS</u></b>	<b>47</b>
2.1 ESTRUCTURAS JERÁRQUICAS DE CONTROL	47
2.2. MODELOS DE REDES DE PETRI VS. CONTROLADORES DE REDES DE PETRI	49
2.3 REDES DE PETRI VS. OTROS MODELOS EN MODELAMIENTO Y CONTROL	50
2.4 VENTAJAS DE LAS REDES DE PETRI	52
2.5. MÉTODOS DE ANÁLISIS.	54
2.5.1. <i>TÉCNICAS ENUMERATIVAS.</i>	54
2.5.2. <i>TÉCNICAS DE TRANSFORMACIÓN.</i>	54
2.5.3. <i>TÉCNICAS ESTRUCTURALES</i>	54
<b><u>3. MÉTODOS DE MODELADO</u></b>	<b>56</b>
3.1. PASOS PARA EL MODELADO.	56
3.2 EJEMPLO DE APLICACIÓN II: (SISTEMA DE MANUFACTURA)	57
3.2.1. <i>SIGUIENDO LA METODOLOGÍA</i>	58
<b><u>4. REDES DE PETRI INTERPRETADAS PARA CONTROLADORES LÓGICOS PROGRAMABLES.</u></b>	<b>62</b>
4.1. ESTRUCTURAS GENERALIZADAS	63
4.2. EQUIVALENCIA CON PLC´s.	69
4.3. EJEMPLO DE APLICACIÓN III: (INYECTORA DE PLÁSTICO)	70
<b><u>5. SIMULACIÓN</u></b>	<b>76</b>
5.1. EJEMPLO 1: SECUENCIA DE UNA RED DE PETRI.	77
5.2. EJEMPLO 2: RED DE PETRI CON EL FENÓMENO DE EXCLUSIÓN MUTUA	78
5.3 EJEMPLO 3: RED DE PETRI CON PARALELISMO.	80
5.4. EJEMPLO 4: CELDA DE UN SISTEMA DE MANUFACTURA CON RDP.	81
<b><u>CONCLUSIONES</u></b>	<b>85</b>
<b><u>BIBLIOGRAFÍA</u></b>	<b>87</b>
<b>ANEXOS</b>	



	<b>Pág.</b>
<i>Figura 1. Lugar, arco y transición</i> .....	22
<i>Figura 2. Lugares de entrada y de salida</i> .....	22
<i>Figura 3. Tipos de nodos</i> .....	24
<i>Figura 4. Red de Petri generalizada</i> .....	25
<i>Figura 5. Ejemplo del macado de una RdP</i> .....	26
<i>Figura 6. Varios casos de disparo de una transición</i> .....	27
<i>Figura 7. Ejemplo</i> .....	30
<i>Figura 8. RdP viva y RdP no-viva</i> .....	31
<i>Figura 9. Red de Petri no binaria</i> .....	32
<i>Figura 10. Red de Petri conforme</i> .....	33
<i>Figura 11. Arco inhibidor</i> .....	34
<i>Figura 12. Red de Petri sincronizada</i> .....	35
<i>Figura 13. Redes de Petri temporizadas.</i> .....	37
<i>Figura 14. RdP para sistemas de colas</i> .....	39
<i>Figura 15. Ejemplo sistema de colas</i> .....	40
<i>Figura 16. Ejemplo de sistema de colas</i> .....	41
<i>Figura 17. Línea transfer de tres etapas</i> .....	42
<i>Figura 18. Modelo de red de Petri temporizada, para el ejemplo I.</i> .....	43
<i>Figura 19. Arquitectura de control de un DES real</i> .....	48
<i>Figura 20. Redes de Petri como una representación de modelamiento, análisis y control</i> .....	53
<i>Figura 21. Sistema de manufactura</i> .....	57
<i>Figura 22. (a) Grafo de RdP para la secuencia del ejemplo del sistema de manufactura, (b) RdP con recursos, lugares, arcos y marca inicial</i> .....	59
<i>Figura 23. Sistema a eventos discretos</i> .....	62
<i>Figura 24. Esquema en RdP de la figura 23</i> .....	62
<i>Figura 25. Función de arranque y paro de un motor.</i> .....	63
<i>Figura 26. Estructura general de un proceso</i> .....	64
<i>Figura 27. Inyectora de plástico</i> .....	70
<i>Figura 28. Ciclo automático Inyectora de plástico</i> .....	72
<i>Figure 29. Programa de la Inyectora</i> .....	74
<i>Figura 30. Applet para simulación.</i> .....	76
<i>Figura 31. Secuencia para una red de Petri.</i> .....	77
<i>Figura 32. Transición t1, activada</i> .....	77
<i>Figura 33. Transición t2, activada</i> .....	78
<i>Figura 34. Transición t3, activada</i> .....	78
<i>Figura 35. Paralelismo en RdP</i> .....	79
<i>Figura 36. Transición t1, activada</i> .....	79
<i>Figura 37. Transición t2, activada</i> .....	79
<i>Figura 38. RdP con Paralelismo.</i> .....	80
<i>Figura 39. Transición t1, activada</i> .....	80
<i>Figura 41. Transición t1, activada</i> .....	81

<i>Figura 42. Transición t2, activada.....</i>	<i>82</i>
<i>Figura 43. Transición t3, activada.....</i>	<i>82</i>
<i>Figura 44. Transición t4, activada.....</i>	<i>83</i>
<i>Figura 45. Transición t5, activada.....</i>	<i>83</i>
<i>Figura 46. Transición t6, activada.....</i>	<i>84</i>
<i>Figura 47. Transición t7, activada.....</i>	<i>84</i>



<i>Tabla 1. Lugares y transiciones.....</i>	<i>45</i>
<i>Tabla 2. Dispositivos.....</i>	<i>70</i>
<i>Tabla 3. Dispositivo Simple efecto / sin confirmación.....</i>	<i>71</i>
<i>Tabla 4. Dispositivo Doble efecto / sin confirmación.....</i>	<i>72</i>
<i>Tabla 5. Dispositivo Doble efecto/doble confirmación.....</i>	<i>73</i>
<i>Tabla 6. Representación de la redes de Petri a RLL.....</i>	<i>74</i>
<i>Tabla 7. Componentes de la Inyectora de plástico.....</i>	<i>76</i>

### ACRÓNIMOS

**RdP** *Redes de Petri*

**RdPT** *Redes de Petri temporizadas*

**PLC** *Controladores lógicos programables*

**DES** *Sistemas de Eventos Discretos*

**DSP** *Procesadores digitales de señales*

**RLL** *Relay ladder logic*

### TÉRMINOS

**Lugares:** Se representan mediante círculos y representan esa fase “estable” por la que atraviesa el sistema entre dos “sucesos” consecutivos que acontecen.

**Transiciones:** Se representan por segmentos de recta, los cuales llevan asociados los eventos, cuya activación, provoca el disparo de la transición y por ende el marcado de uno o más lugares siguientes.

**Arcos:** Son segmentos de recta orientados que unen lugares y transiciones de forma alternativa. Cada arco, lleva asociado una función de peso  $w$ , que deberá ser un entero positivo  $\{0, 1, 2, \dots\}$ .

**Tokens:** Se representan por puntos en los lugares, y determinan la disponibilidad de los recursos o la ejecución de las actividades.

**Asincronismo:** El asincronismo es característico de los sistemas de eventos discretos, siendo esta una de sus más importantes propiedades, ya que los diferencia de los sistemas descritos por ecuaciones diferenciales. Para los sistemas de manufactura cada etapa de la producción está sincronizada por un reloj global, pero en realidad la ocurrencia de eventos se produce dentro de estos procesos de forma asíncrona.

**Relaciones secuenciales:** Para un grupo de eventos, pueden existir muchas relaciones de precedencia entre ellos. Lo que se dice es: que la relación secuencial entre dos eventos se cumple si un evento ocurre solo después de la ocurrencia del otro.

**Exclusión mutua:** se presenta cuando dos eventos requieren del mismo recurso para su ocurrencia y el sistema no puede realizar ambas tareas al mismo tiempo.

**Concurrencia:** Por concurrencia a este trabajo se refiere a que; entre dos eventos no existe ninguna relación secuencial entre ellos.

**No-determinismo:** Dos tipos de no-determinismo pueden ser observados. El primero, como el que se muestra en la concurrencia y se puede dar cuando dos eventos están listos para tomar lugar en el mismo momento. El segundo, está dado por los pequeños cambios en los parámetros del proceso y se presenta en las mínimas diferencias entre los tiempos de procesos entre dos máquinas diferentes.

Los sistemas de eventos discretos están compuestos por elementos que manejan entidades discretas, es decir enumerables y diferenciables entre sí (piezas, mensajes, archivos, etc.). Su funcionamiento está caracterizado por una sucesión de estados estables delimitados por eventos que ocurren; generalmente, de manera asíncrona. Como ejemplos de estos sistemas, se observa los sistemas operativos de computadoras y los sistemas de manufactura y sus controladores. Para el análisis de sistemas ya existentes, diferentes pruebas que incluyen la simulación pueden aplicarse al modelo con el fin de detectar problemas o bien para obtener información sobre el comportamiento del sistema o de una parte de él.

La legibilidad del modelo y las técnicas de análisis dependen de la herramienta o formalismo adoptado, por lo que es importante una adecuada selección de este. Es importante también, tener en cuenta si un formalismo tiene desarrollados métodos que conduzcan a la implementación del sistema. Entre los formalismos que existen actualmente, se destacan las redes de Petri por sus características surgidas de su naturaleza gráfica y su soporte matemático simple: con claridad en la descripción y facilidad para representar comportamientos complejos que incluyan secuencias, concurrencia, paralelismo, sincronizaciones e intercambios de información.


Este formalismo, definido por Carl A. Petri en 1962, fue utilizado inicialmente en la computación y rápidamente fue aplicado en otras áreas tales como la automatización, sistemas de manufacturas, informática, robótica, protocolos de comunicación y en la inteligencia artificial. Alrededor de este formalismo se



han desarrollado técnicas de análisis cualitativo que permiten detectar entre los problemas, bloqueos (dead-locks) y acumulación de material o de información (overflows), sin especificar la duración de las actividades representadas en el modelo. El análisis cuantitativo toma en cuenta la duración de las actividades y además de detectar los problemas mencionados, da información sobre las prestaciones o desempeño del sistema. El análisis con redes de Petri puede ser efectuado por simulación o empleado métodos que obtienen información a partir de la estructura de la red.

La simulación consiste en la ejecución de la red de manera autónoma. La ejecución autónoma requiere especificar en el modelo la duración de las operaciones y restricciones temporales sobre la disponibilidad de los recursos. En la ejecución interactiva es necesario indicar cada evento que sucede y observar la respuesta a tal evento. La extensión al formalismo original que incluye el tiempo son las redes de Petri temporizadas (RPT). Existen métodos de análisis que permiten evaluar el desempeño del sistema sin necesidad de ejecutar el modelo; sin embargo los resultados son válidos para modelos que cumplen con ciertas restricciones estructurales, las cuales no corresponden siempre a modelos de sistemas reales.

En el siguiente documento se mostrarán las redes de Petri como una alternativa para modelar sistemas, sus características hacen que para algunos problemas las redes de Petri funcionen de una manera natural. Las Redes de Petri son consideradas una herramienta para el estudio de los sistemas. Con su ayuda podemos modelar el comportamiento y la estructura de un sistema, y llevar el modelo a condiciones límite, que en un sistema real son difíciles de lograr o muy costosas. La teoría de red de Petri ha llegado a ser reconocida como una metodología establecida en la literatura de la robótica para modelar los sistemas de manufactura flexibles.



La automatización de procesos industriales requiere de la sistematización de procedimientos que faciliten el diseño, el desarrollo y la implementación de los programas que deben ejecutar los sistemas electrónicos donde residen los programas de mando. Estos programas adquieren diversas apariencias que dependen de la tecnología utilizada para implementar dicha automatización, es decir, de si se usan autómatas programables, PLC, microcontroladores, DSP u cualquier otra tecnología. El análisis y control de sistemas con tareas paralelas y concurrentes requiere de la construcción de modelos eficientes. El método de redes de Petri permite evaluar el paralelismo y la concurrencia eficientemente a través de un procedimiento sistemático.

Las redes de Petri (RdP en adelante) representan una alternativa para modelar sistemas, sus características hacen que, para algunos problemas las redes de Petri funcionen de una manera natural.

Las redes de Petri (RdP) fueron introducidas en la literatura en la tesis doctoral "kommunikation mit automaten" (Comunicación con autómatas), del alemán Carl Adam Petri en 1962, en esta se establece los fundamentos para el desarrollo teórico de los conceptos básicos de las redes de Petri, como una herramienta para simular las propiedades dinámicas de sistemas complejos mediante modelos gráficos de procesos concurrentes<sup>1</sup>.

---

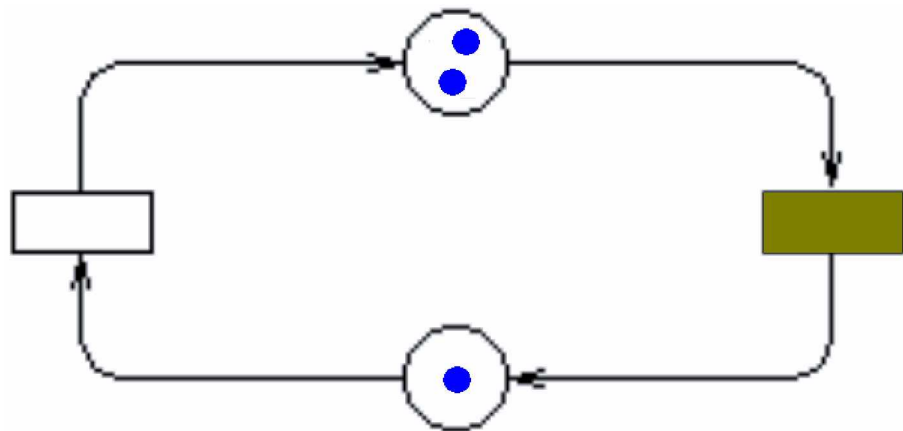
<sup>1</sup> LÓPEZ, Ernesto, Redes de Petri para el análisis temporal de sistemas, Ciencia UANL / Vol. I, No. 2, Abril– Junio 1998

Las redes de Petri son consideradas una herramienta para el estudio de los sistemas. Con su ayuda es posible modelar el comportamiento y la estructura de un sistema, y llevar el modelo a condiciones límite, que en un sistema real son difíciles de lograr o muy costosas.

La teoría la RdP ha llegado a ser reconocida como una metodología establecida en la literatura de la robótica para modelar los sistemas de manufactura flexibles. Comparada con otros modelos de comportamiento dinámico gráficos, como los diagramas de las máquinas de estados finitos, las RdP ofrecen una forma de expresar procesos que requieren sincronía. Y quizás lo más importante es que las RdP pueden ser analizadas de manera formal y obtener información del comportamiento dinámico del sistema modelado.

El modelado constituye una etapa importante, puesto que pruebas efectuadas sobre un modelo permiten detectar problemas de diseño debidas generalmente a que las especificaciones son incompletas o ambiguas; la aplicación iterada de ajustes y pruebas al modelo conducen a una implementación confiable. Por otro lado, para el análisis de sistemas ya existentes, diferentes pruebas que incluyen la simulación pueden aplicarse al modelo con el fin detectar problemas o bien para obtener información sobre el comportamiento del sistema o de una parte de el. La legibilidad del modelo y las técnicas de análisis dependen de la herramienta o formalismo adoptado, por lo que es importante una adecuada selección de este. La idea principal de este documento es adoptar el formalismo RdP como una herramienta útil en el modelado de sistemas.

# FUNDAMENTOS



## 1. FUNDAMENTOS

Una red de Petri (RdP), es un formalismo utilizado para modelizar, analizar, simular, controlar y evaluar el comportamiento de sistemas concurrentes y distribuidos. Este formalismo, sin embargo, no ofrece una manera directa de modelizar características tales como cambios dinámicos, modos de operación múltiples, etc. En términos básicos una RdP es un grafo orientado formado por elementos denominados lugares, las transiciones, y los arcos. Cada uno de estos elementos representa conceptos diferentes. Los lugares se representan mediante círculos y, en primera definición sin excesivo rigor y de una manera intuitiva, representan esa fase “estable” por la que atraviesa el sistema entre dos “sucesos” consecutivos que acontecen en el sistema. Las etapas llevan asociado el concepto de marcado del que hablaremos más adelante.

Las transiciones se representan por segmentos de recta, los cuales llevan asociados los eventos, cuya activación, debe provocar el disparo de la transición y por ende el marcado de uno o más lugares siguientes.

Los arcos, son segmentos orientados que unen lugares y transiciones de forma alternativa. Cada arco, lleva asociado una función de peso  $w$ , que deberá ser un entero positivo  $\{0, 1, 2, \dots\}$

De una manera formal, una red de Petri es una cuadrupla

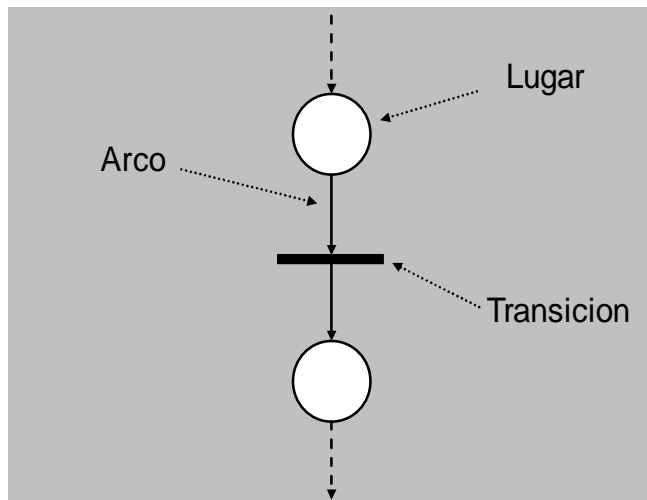
$(P, T, A, W)$  Donde

$P$  es un conjunto finito de lugares

$T$  es un conjunto finito de transiciones

$A$  es un conjunto de arcos

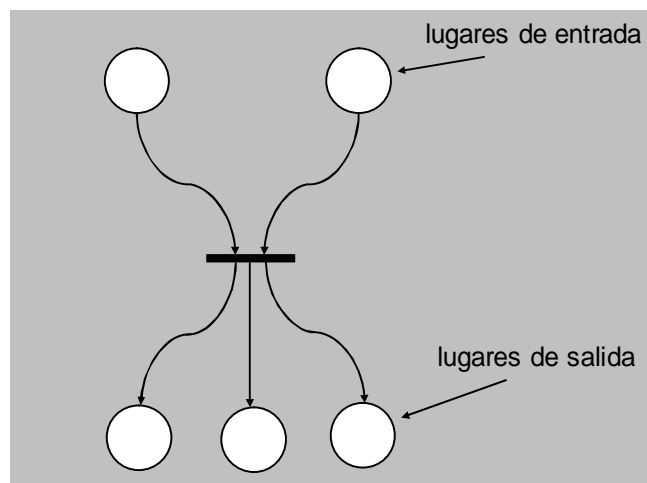
$w$  es una función peso,  $w: A \rightarrow \{1, 2, 3, \dots\}$



**Figura 1.** Lugar, arco y transición

*1.1. Lugares de entrada y de salida*

Un lugar se denomina de entrada a una transición, si existe un arco orientado desde un lugar hasta la citada transición. De la misma manera, un lugar se denomina de salida de una transición, si existe un arco orientado desde la transición a dicho lugar.



**Figura 2.** Lugares de entrada y de salida

## 1.2. Tipos de nodos de las RdP

Para la representación gráfica de los conceptos de concurrencia, sin sincronismo, exclusión de secuencias y otros, que trataremos con mayor profundidad mas adelante, las RdP poseen distintos tipos de nodos mediante los que se convienen dicha representación.

Un lugar que posee varios arcos de entrada y/o varios arcos de salida se denomina *nodo* OR. Pueden presentarse dos casos particulares

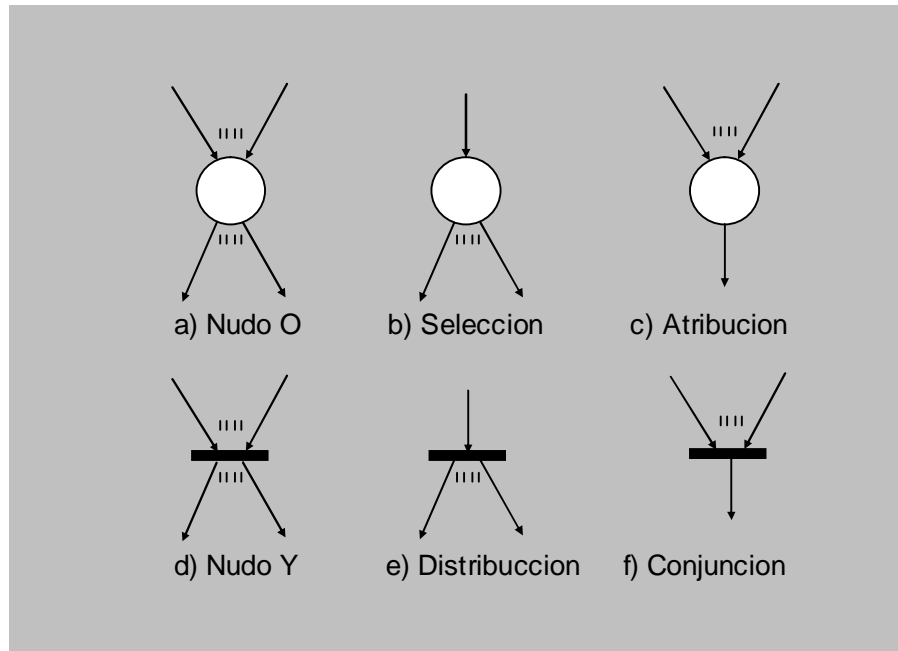
**Nodo de selección:** Es aquel nodo OR que únicamente posee un arco de entrada y varios de salida.

**Nodo de atribución:** Es aquel nodo OR que únicamente posee un arco de salida y varios de entrada.

Una transición que posee varios arcos de entrada y/o salida se denomina nodo AND. Al igual que en el caso anterior, se presentan dos casos particulares.

**Nodo de Distribución:** Es aquel nodo AND que presenta un único arco de entrada.

**Nodo de Conjunción:** Es aquel nodo AND que presenta un único arco de salida.



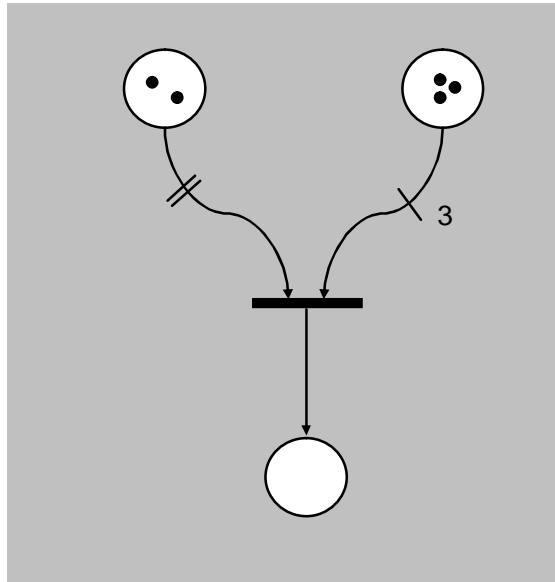
**Figura 3. Tipos de nodos**

### 1.3. Redes de Petri ordinarias y generalizadas

Redes de Petri ordinarias, son aquellas en las que los pesos asociados a los arcos toman como máximo un valor 1, es decir,  $w \in \{0,1\}$ . La representación grafica de este tipo de RdP., queda establecida con solo representar el arco en cuestión sin ningún otro tipo de notación (Véase la figura 2.)

Por otra parte, las RdP generalizadas son aquellas que pueden poseer arcos cuya función de peso sea  $w > 1$ . En este caso, su representación grafica deberá reflejar específicamente su peso, bien mediante un entero junto a un segmento que atraviesa el arco correspondiente, o bien únicamente con un numero de segmentos idéntico al peso del arco correspondiente.



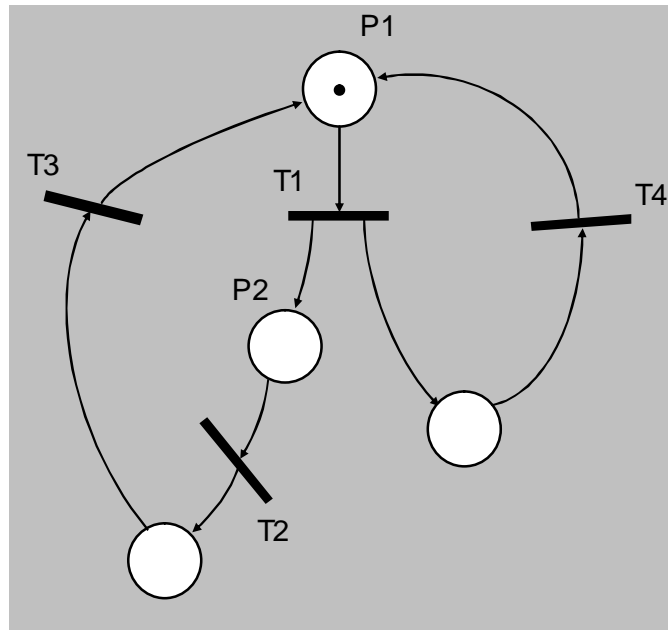


**Figura 4.** Red de Petri generalizada

#### 1.4. Marcado de una red de Petri

Cada lugar perteneciente a una RdP puede contener un número entero positivo, no necesariamente limitado de marcas, representadas por un punto en el interior del círculo representativo de cada lugar (*tokens*), gráficamente dibujados como puntos negros. El estado del sistema que la red esta modelando es representado con la asignación de estos tokens a los sitios o lugares.

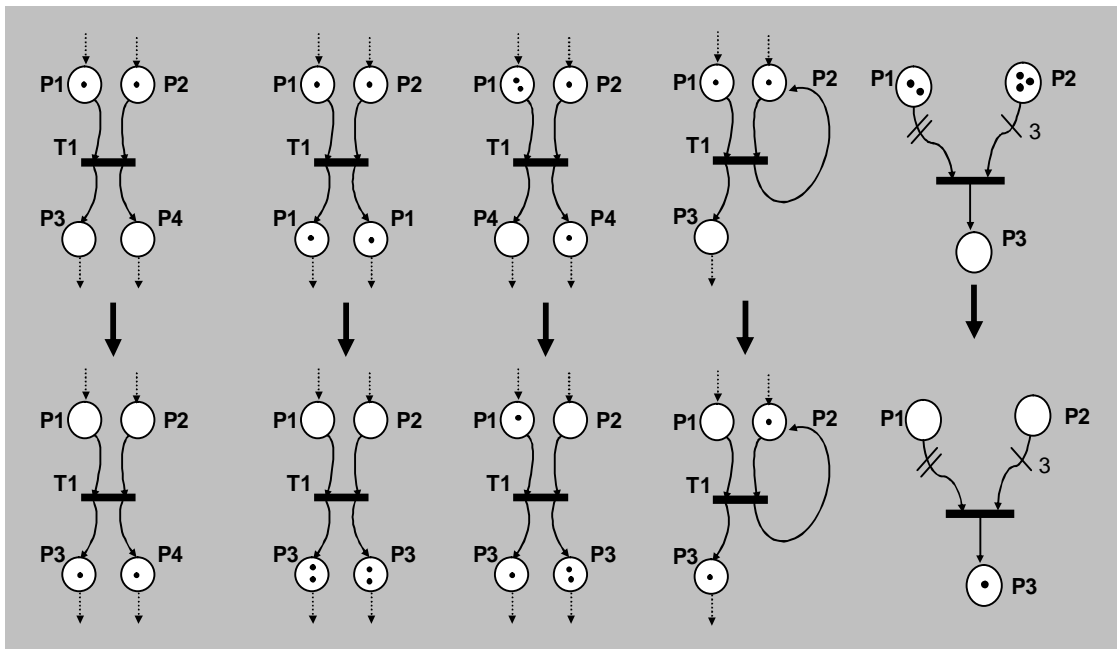
En la siguiente Figura 5., se observa que el lugar 1 posee una marca. Al conjunto de marcas de una RdP, es un instante considerado, se le denomina marcado de una RdP.



**Figura 5.** Ejemplo del macado de una RdP

### 1.5. Disparo de una transición

Para que una transición pueda dispararse, hecho que se asocia al franqueamiento de dicha transición, ha de estar validada; es decir, el lugar o lugares de entrada a dicha transición han de poseer al menos una marca en las RdP ordinarias, y un número idéntico al del peso del arco de salida en las RdP generalizadas. El disparo de la transición se produce cuando estando valida, acontezca el evento asociado a dicha transición, con lo cual deberá quitarse una marca de cada uno de los lugares de entrada a la transición, y añadirse una marca a cada uno de los lugares de la salida de la misma (caso de las RdP ordinarias). Si la RdP es generalizada el número de marcas, que se quitan en los lugares de entrada y se ponen en los lugares de salida, han de coincidir con los pesos de los arcos de entrada y salida respectivamente. En la figura 6, se contemplan distintos casos en el disparo de una transición.



**Figura 6.** Varios casos de disparo de una transición

### 1.6. Tipos de redes de Petri

Cuando los instantes de disparo de una RdP no se indican o se desconocen, se dice que dicha RdP es una red de Petri autónoma. Cuando la evolución de un sistema viene determinada por eventos externos o por la variable de tiempo, se dice que dicha RdP es no-autónoma. Las RdP no autónomas, a su vez, pueden ser sincronizadas y /o temporizadas.

Para el estudio de las propiedades de carácter cualitativo de las RdP, utilizaremos las RdP autónomas, haciendo extensivas dichas cualidades en el estudio posterior de las RdP no – autónomas.

### 1.7. Propiedades de las redes de Petri

#### 1.7.1. El marcado

El marcado de una RdP representa los lugares de la red, que en un instante considerado, poseen una o varias marcas (*tokens*). De una manera formal, el

marcado de las RdP se representa mediante un vector columna, aunque por motivos de simplicidad en la notación, se utiliza usualmente su forma transpuesta, la expresión correspondiente a la Fig. 5.

$$M_0 = (1,0,0,0) = [1 \ 0 \ 0 \ 0]^T = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Para el marcado  $M_0$  que se esta considerando, únicamente la transición  $T_1$  esta validada; por tanto es la única susceptible de ser disparada. El disparo de esta transición, partiendo del marcado inicial  $M_0$ , conduce al marcado  $M_1$ .

La notación formal de lo anterior es:

$$M_0 [T_1 > M_1 = (0, 1, 1, 0)$$

Una vez realizado el disparo en consideración, permanecerán validadas  $T_2$  y  $T_4$  pudiendo a su vez, producirse un nuevo disparo a partir de

$$M_1 [T_2 > M_2 = (0, 0, 1, 1)$$

$$M_1 [T_4 > M_3 = (0, 1, 1, 0)$$

La anterior notación, también puede utilizarse incluyendo una secuencia de disparo, donde puede verse afectada más de una transición. La notación para una secuencia de disparo se establece como:

$$S = T_1 T_2$$

Y el marcado alcanzable, desde un marcado inicial, aplicando la citada secuencia

$$M_0 [S < M_2$$

## 1.8. Ecuación fundamental de las redes de Petri

Definiremos previamente una serie de conceptos necesarios, para la resolución de la denominada ecuación fundamental de las RdP.

### 1.8.1. Vector característico

Sea la secuencia de disparo  $S = T_1T_2T_3T_2$ , donde se observa un determinado orden en el disparo de las transiciones implicadas en RdP de la Fig. 5 al tiempo que se observa el número de veces que se disparan las transiciones implicadas en dicha secuencia. Se denomina vector característico  $\underline{S}$  de dicha secuencia de disparo, al vector  $m$ -componente cuyo componente  $j$ , corresponde el número de veces que la transición  $T_j$  se ha disparado.

Para el ejemplo de la figura 5, el vector característico correspondiente es

$$\underline{S} = (2, 1, 1, 0)$$

### 1.8.2. Matriz de incidencia

- Matriz de incidencia de entrada:

Cada uno de los elementos de la matriz de incidencia de entrada de una RdP, informa del valor de la aplicación de incidencia previa del lugar  $P_m$  a la transición  $T_n$ , lo cual, implícitamente, informa de la existencia de un arco de entrada desde el lugar a la transición, así como del peso de dicho arco. La matriz de incidencia previa se representa por

$W^-$

- Matriz de incidencia de salida

Cada uno de los elementos de la matriz de incidencias de salida de una RdP, informa del valor de la aplicación de incidencia de salida de la transición  $T_n$  al lugar  $P_m$ , lo cual, implícitamente, informa de la existencia de un arco de salida desde la transición al lugar, así como del peso de dicho arco. La matriz de incidencia de salida se representa por

$W^+$

La matriz de incidencia de una RdP, cuya expresión viene dada por

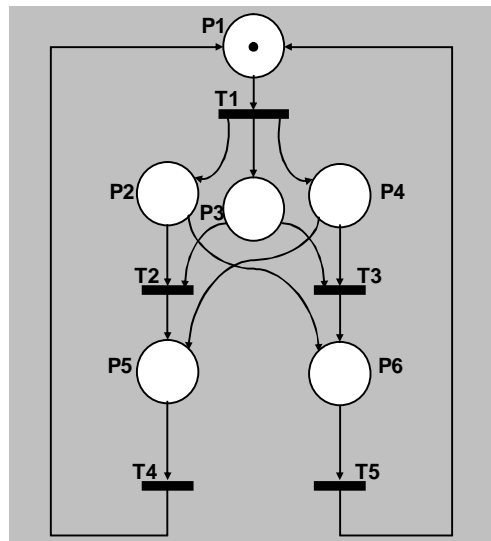
$$W = w^+ - w^-$$

Da idea del balance final en el número de marcas que se producen en los lugares de la misma, cuando se realiza el disparo de las transiciones implicadas. Específicamente, una columna de la matriz de incidencia muestra la modificación que se produce en todos los lugares de la RdP, por el disparo de la transición correspondiente.

Dado un marcado inicial  $M_i$ , una secuencia de disparo representada por su vector característico correspondiente  $\underline{S}$ , y una determinada matriz de incidencia  $W$ , es posible conocer el marcado alcanzable de una RdP, mediante simples operaciones de calculo matricial, resolviendo la siguiente expresión:

$$M_f = M_i + W \cdot \underline{S}$$

Véase el ejemplo de la siguiente Figura 7.

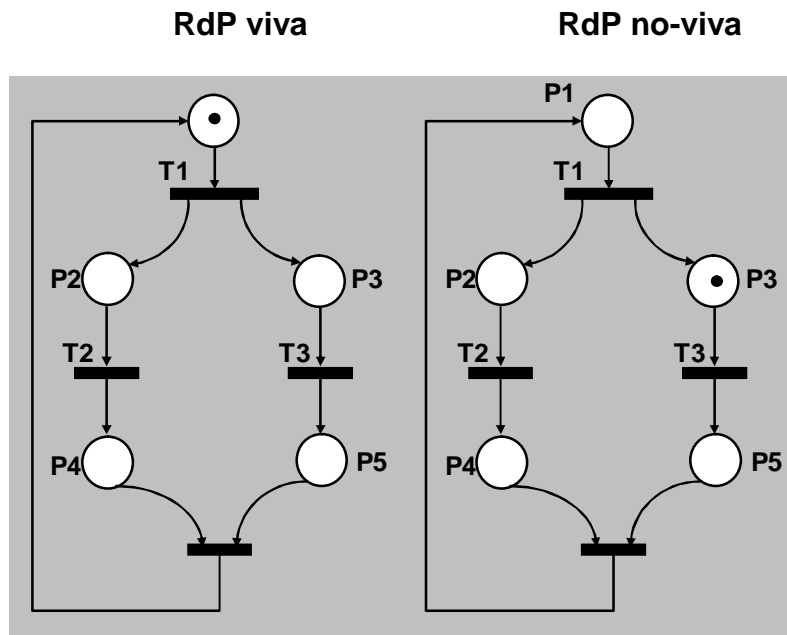


**Figura 7. Ejemplo**

### 1.8.3. Red de Petri viva

El concepto de vivacidad de una RdP es sumamente importante, desde el momento en que el modelo de RdP que presente esta propiedad y que represente a un sistema en particular, por ejemplo un automatismo, va a suponer que este no presentara bloqueos estructurales en su diseño. Lo contrario, es decir, cuando el modelo representativo del sistema presenta transiciones que no pueden ser disparadas a partir de un cierto momento, supone la existencia del concepto de bloqueo.

Una definición formal de lo anterior expone que: una transición  $T_j$  es viva, para un marcado inicial  $M_0$ , si para cada marcado alcanzable  $M_i$  existe una secuencia de disparo  $S$ , desde  $M_i$ , que contenga  $T_j$ .



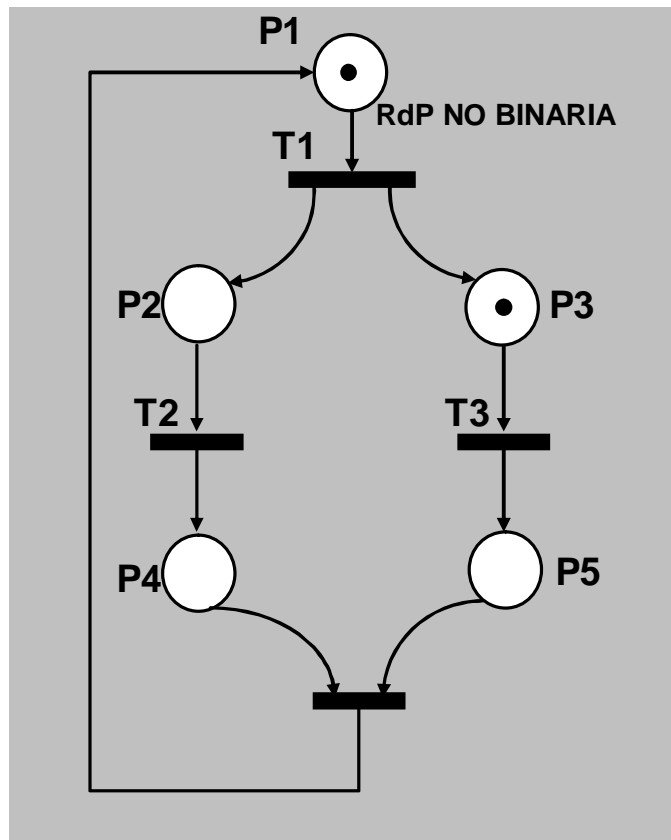
**Figura 8.** RdP viva y RdP no-viva

### 1.8.4. Red de Petri limitada

Para un marcado inicial, un lugar se denomina limitado si existe un entero natural  $k$ , tal que para cualquier marcado alcanzable desde  $M_0$ , el número de

marcas no es mayor que  $k$ . Si lo anterior se cumple, se dice que el lugar es  $k$ -limitado. Una red de Petri es limitada para un marcado inicial  $M_0$ , si todos los lugares son limitados  $M_0$ .

Un caso particular de las RdP limitadas son aquellas que únicamente poseen una marca. En tal caso este tipo de RdP reciben el nombre de binarias y sus derivados, tales como el Grafcet, merecen atención especial puesto que su uso se ha generalizado en el diseño de automatismos basados en Autómatas Programables, al igual en lógica cableada mediante módulos secuenciadores.

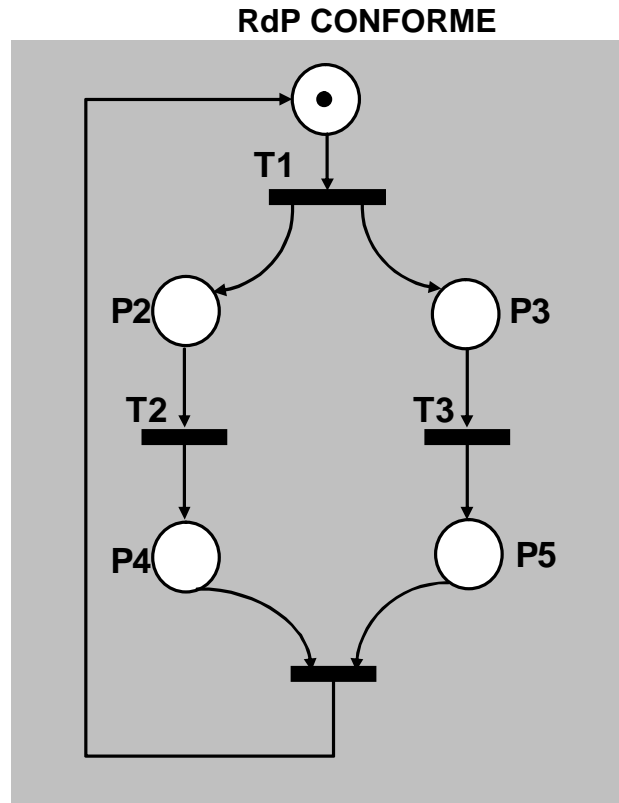


**Figura 9.** Red de Petri no binaria

#### 1.8.5. Red de Petri conforme

Para un marcado inicial  $M_0$  dado, se dice que una RdP es conforme si es binaria y viva, como podemos observar en la figura a continuación.



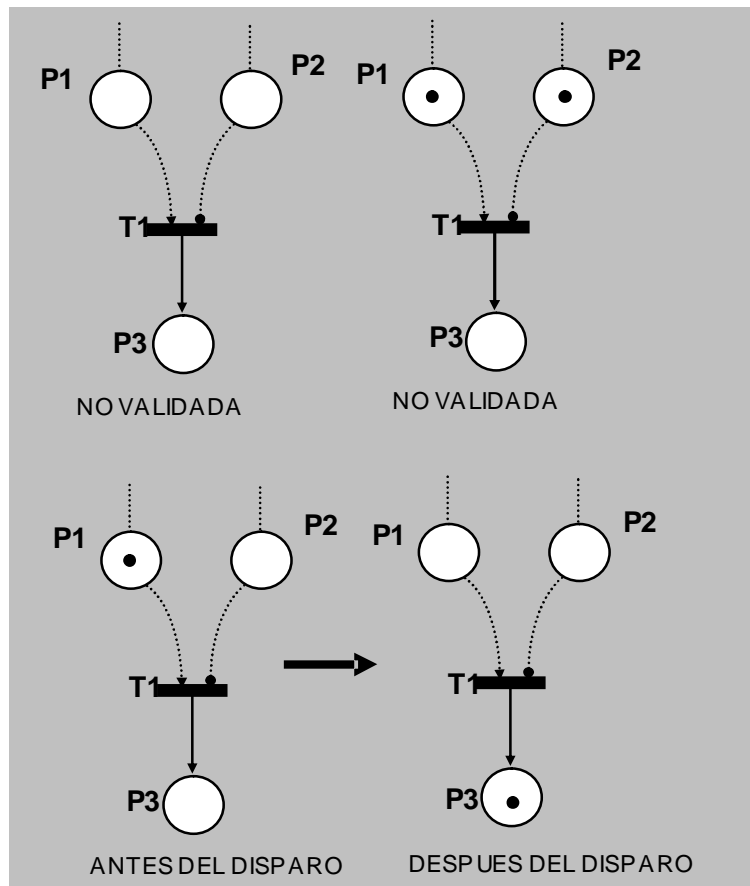


**Figura 10.** Red de Petri conforme

### 1.9. Arcos Inhibidores

Los arcos inhibidores son arcos dirigidos que unen exclusivamente lugares con transiciones. En el extremo del arco, junto con la transición Fig. 11. Se coloca un pequeño círculo para denotar su característica inhibidora. Su denominación viene dada por el hecho de que la validación de la transición implicada, se dará cuando en el lugar de entrada a dicha transición, conectado por el arco inhibidor, no exista marca alguna.

En el ejemplo de la figura 11, se vera la validación de una transición.



**Figura 11.** Arco inhibitor

### 1.10. Redes de Petri no – autónomas

Hasta el momento se han estudiado algunas de las propiedades de las RdP desde el punto de vista cualitativo. A partir de este apartado, se describirán las RdP, cuya evolución queda condicionada por eventos externos al sistema descrito por estas, así como por la variable tiempo.

#### 1.10.1 Redes de Petri sincronizadas

Las RdP sincronizadas son aquellas en las que cada transición T tiene asociado un evento, teniendo lugar al disparo de dicha transición cuando el

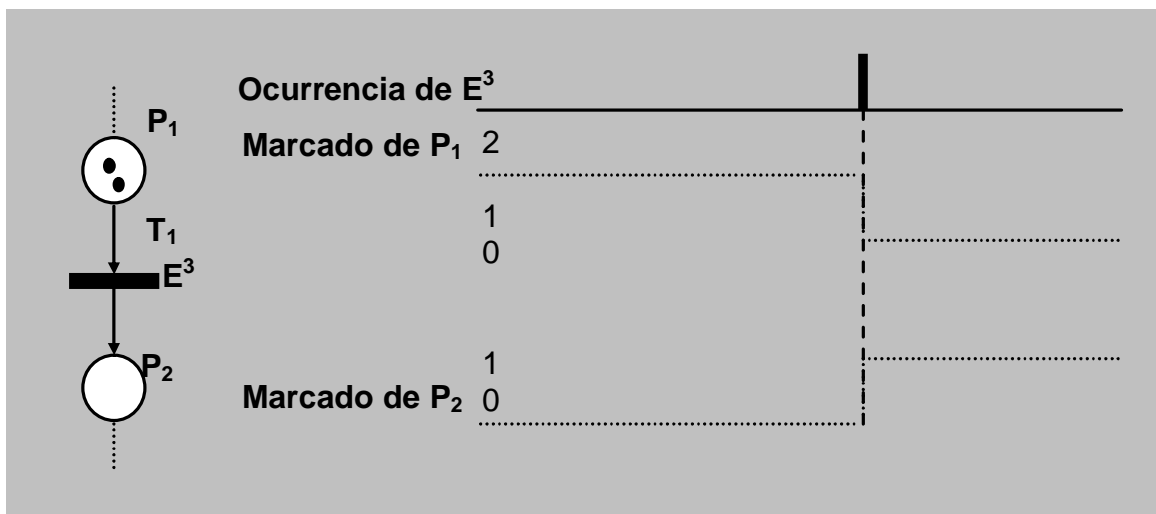
evento ocurre. Los eventos externos se corresponden con cambios de estado del mundo exterior.

En una definición formal, una RdP sincronizada, es una tripleta

$$\langle R, E, \text{Sync} \rangle$$

Donde R es una RdP marcada, E es un conjunto de eventos externos, Sync es una función del conjunto T de transiciones de R a  $E \cup \{e\}$ , donde e, es el evento de siempre concurrencia.

El conjunto  $E = \{E^1, E^2, \dots\}$ , es el conjunto de eventos externos, donde la notación se corresponde con el nombre del número de un evento externo, y la notación  $E_j$  se corresponde con el evento asociado con la transición  $T_j$ .



**Figura 12.** Red de Petri sincronizada

### 1.10.2. Redes de Petri temporizadas

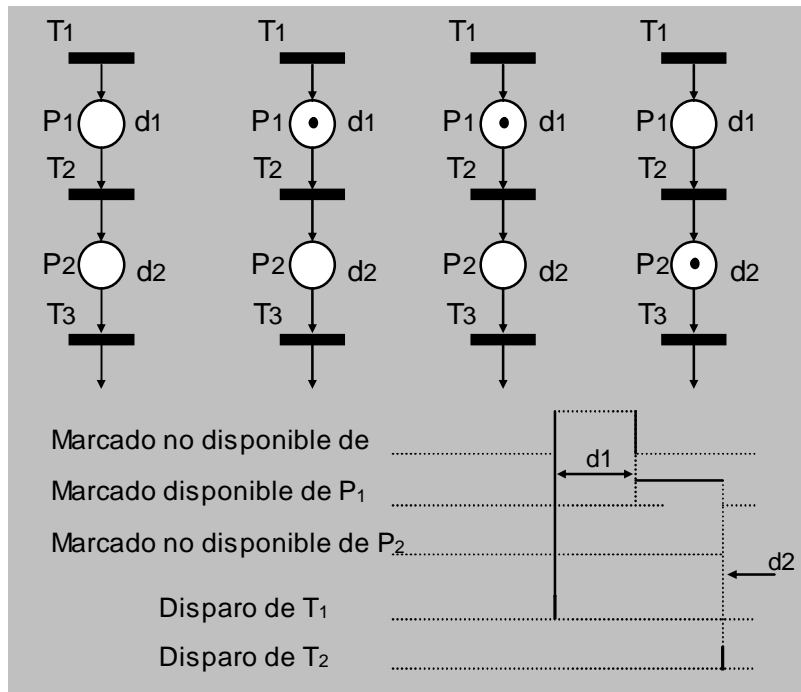
En los sistemas automatizados la variable tiempo interviene de forma casi constante, en el sentido de que el inicio, final y duración de toda una serie de operaciones esta condicionada por esta variable. Además, independientemente del modelado cualitativo de estos sistemas, la variable tiempo es indispensable para el análisis de la evaluación en ejecución de los

sistemas automatizados de producción. En este tipo de análisis, a grosso modo, de lo que se persigue es poder minimizar el tiempo que el sistema productivo dedica a las operaciones que no ofrecen valor añadido al producto susceptible de ser fabricado.

Algunos de los parámetros utilizados para el anterior análisis son:

- Tiempo principal de fabricación (Manufacturing Lead Time)
- Trabajo en curso (Work – in - process)
- Utilización de máquina (Machina utilization)
- Inicio de producción (Throughput)
- Capacidad (capacity)
- Flexibilidad (Flexibility)
- Ejecutabilidad (Performability)
- Calidad (Quality)

Existen dos tipos de RdP temporizadas, las P-temporizadas, y las T-temporizadas, en las que el concepto de temporizadas esta referido a los lugares y las transiciones, respectivamente.



**Figura 13.** Redes de Petri temporizadas.

### 1.10.3. Redes de Petri P - temporizadas

En una definición formal, una RdP P-temporizadas es un par  $\langle R, \text{Tempo} \rangle$  tal que: R es una RdP marcada, y Tempo es una función del conjunto P de lugares y del conjunto de números racionales positivos o cero.

La notación  $\text{Tempo}(P_j) = d_j$ , se refiere a la temporización asociada a un lugar  $P_j$ .

El principio de funcionamiento de una RdP temporizada es el siguiente: cuando una marca se deposita en un lugar  $P_i$ , la marca debe permanecer en este, al menos durante el tiempo  $d_i$ . Este periodo de tiempo se denomina periodo de no disponibilidad de la marca. Cuando el periodo de un tiempo ha transcurrido, solo entonces la marca esta disponible ante un eventual disparo de la transición. El caso particular en el que la transición se dispara tan pronto como queda disponible, se define como *funcionamiento a la máxima velocidad*.

#### 1.10.4 Redes de Petri T – Temporizadas

En el caso de las RdP temporizadas, la temporización queda asociada a la transición  $T_i$ . Una marca puede tener dos estados: puede estar *no – reservada*.

En este caso, una determinada transición  $T_1$  puede dispararse y una marca ser depositada en un cierto lugar de salida  $P_1$ , con lo cual la transición de salida  $T_2$  de este queda validada. A partir de esta situación se puede decidir el disparo de  $T_2$ . Cuando se decide el disparo se efectúa la “reserva” de la marca para dicho disparo, la cual, pasado el periodo de tiempo  $d_2$ , se deposita en  $P_2$  con el estado de no – reservada.

#### 1.10.5. Modelos de RdP para sistemas de colas

Un sistema de colas, puede ser definido de una manera simple atendiendo a ejemplos de la vida real tal como la cola de clientes que eventualmente pueden formarse en un comercio, donde podemos distinguir una serie de eventos tales como:

- a: llegada de clientes
- b: comienzo de servicio
- c: servicio completo y salida del cliente

Obsérvese, que este ejemplo puede extenderse a sistemas típicos de los sistemas automatizados de fabricación donde tenemos:

- Llegada de piezas
- Comienzo en el proceso de la pieza
- Tarea completa y salida de la pieza

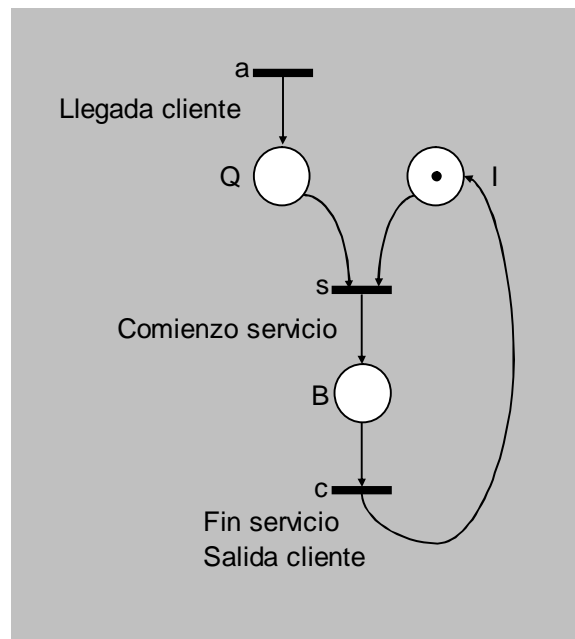
Además, en este tipo de sistemas se han de considerar, como parámetros a tener en cuenta, el tiempo promedio de servicios y la capacidad de encolamiento.

En una primera descripción del sistema, tenemos un conjunto de eventos  $T = \{a, s, c\}$  asociados a diferentes transiciones.

La transición  $a$ , es de carácter espontáneo y no requiere condiciones es decir, no requiere lugares de entrada. La transición  $s$  depende de dos condiciones:

- De la presencia de clientes en la cola
- De que el servidor este ocioso

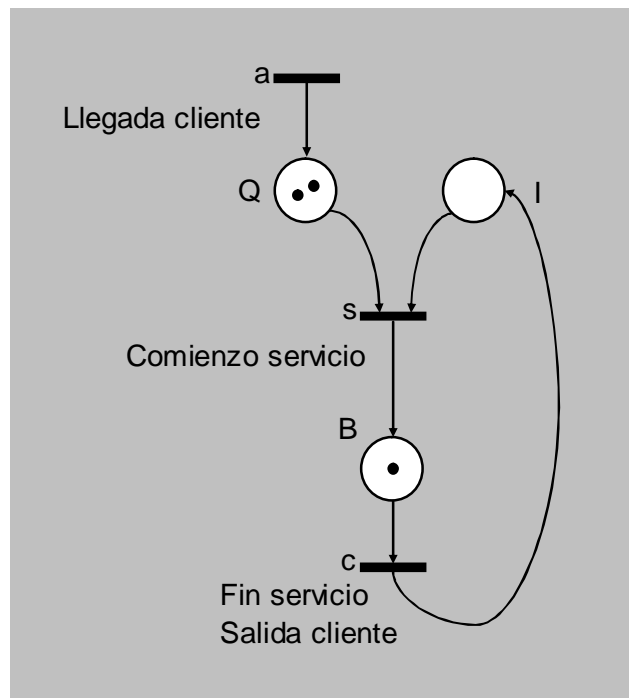
Estas dos condiciones pueden representarse mediante dos lugares de entrada a dicha transición, el lugar  $Q$  (cola), y el lugar  $I$  (ocioso). Por ultimo la transición  $c$  necesita que el servidor este ocupado; por tanto, se introduce un lugar de entrada  $B$  (ocupado). Finalmente el conjunto de lugares finales es  $P = \{Q, I, B\}$ .



**Figura 14.** RdP para sistemas de colas

El grafo de la Figura 14., se muestra el modelo de RdP para el sistema de colas descrito. En el, inicialmente, se ha puesto que la cola vacía al no tener marcas en Q. También, el servidor esta ocioso al estar el lugar I con una marca. Este estado inicial define el marcada  $[0, 1, 0]$ . Debido a que la transición a esta siempre validada, se puede generar distintos caminos de muestreo.

Por ejemplo; la figura 15, muestra el caso en que dos clientes se encuentran en una cola y uno esta siendo atendido; dicho estado se corresponde con el marcado  $[2, 0, 1]$ , resultando de la secuencia de disparo  $\{a, s, a, a, c, s, a\}$ .



**Figura 15.** Ejemplo sistema de colas

Podemos introducir un modelo con un nivel de detalle más elevado, incluyendo una transición adicional D: partida del cliente  
La cual requiere la condición F (cliente finalmente atendido).



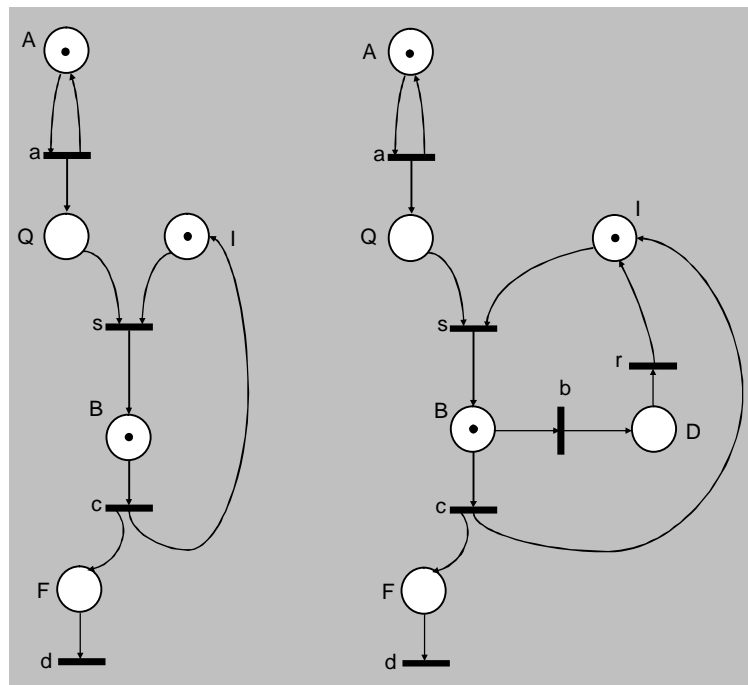
Adicionalmente, el proceso de llegada externo puede representarse mediante un lugar A de entrada a la transición a, de forma que siempre se mantendrá una marca en dicho lugar manteniendo la transición a validada.

Este nuevo modelo, se puede observar en la Fig. 16., en donde se tendrá

$$T = \{a, s, c, d\} \text{ y } P = \{A, Q, I, B, F\}.$$

Aun mas, al nuevo modelo se le puede incorporar la posibilidad de contemplar la posibilidad de b: avería y r: reanudación tras un eventual reparación, introduciendo un nuevo lugar D (caída de servidor). Por tanto, tendremos

$$T = \{a, s, c, d, b, r\} \text{ y } P = \{A, Q, I, B, F, D\}.$$

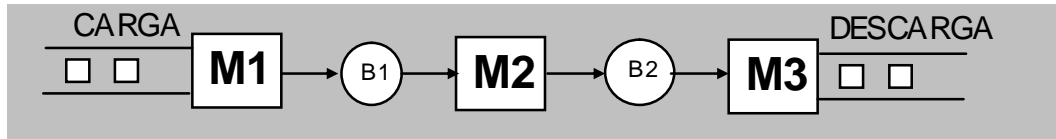


**Figura 16.** Ejemplo de sistema de colas

1.11. Ejemplo de Aplicación I: (Línea de transporte de tres etapas)

Consideremos la línea de transporte de tres etapas, mostrada en la Fig. 17, con tres máquinas y dos puestos de almacenamiento (buffer). Una máquina

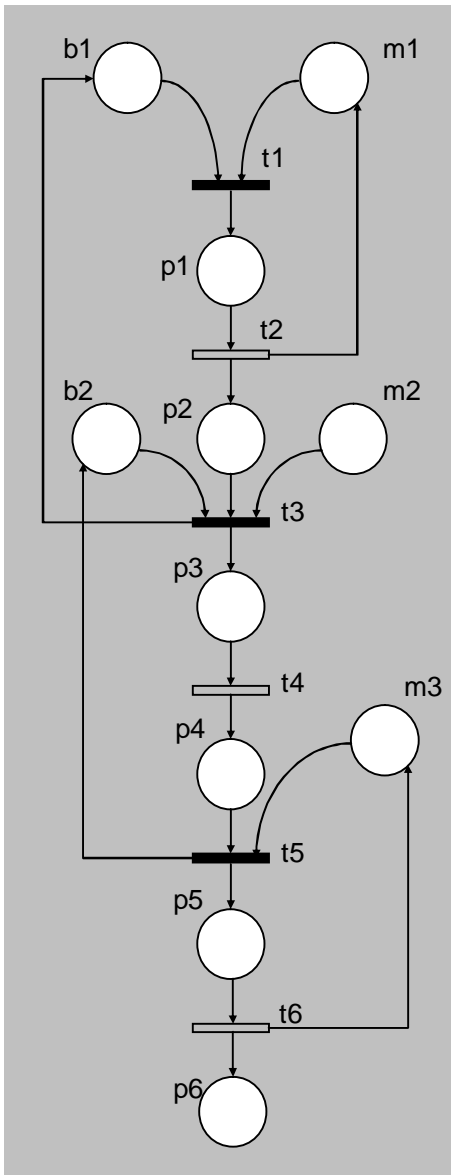
comienza el procesamiento si hay, por lo menos, una pieza en el buffer de entrada y si hay una ranura vacía en el buffer de salida. Suponemos que hay un espacio ilimitado en el buffer de entrada para M1 y un espacio ilimitado en el buffer de salida M3. Se supondrá que el mecanismo de transporte utiliza una cantidad de tiempo insignificante para transportar una pieza desde la máquina al buffer de entrada y del buffer de salida a la máquina.



**Figura 17.** Línea transfer de tres etapas

El modelo de red de Petri temporizada para esta línea de transporte se muestra en la Fig. 18., la interpretación de lugares y las transiciones se muestra en la Tabla 1. Lugares y transiciones.

<b>Lugares</b>	
$B_i$	ranuras disponibles en el buffer $B_i$ ; $i = 1, 2$
$M_i$	$M_i$ disponible, $i=1,2,3$
$p_1$	pieza siendo procesada por M1
$p_3$	pieza siendo procesada por M2
$p_5$	pieza siendo procesada por M3
$p_2$	Piezas que esperan en el buffer de $B_i$
$p_4$	Piezas que esperan en el buffer de $B_2$
$p_6$	Piezas que salen del sistema
<b>Transiciones</b>	
$t_1$	comienzo de procesado de M1 (Inmediata)
$t_2$	final de procesado de M1 (Temporizada)
$t_3$	comienzo de procesado de M2 (Inmediata)
$t_4$	final de procesado de M2 (Temporizada)
$t_5$	comienzo de procesado de M3 (Inmediata)
$t_6$	final de procesado de M3 (Temporizada)



**Figura 18.** Modelo de red de Petri temporizada, para el ejemplo I.

El objetivo es usar este modelo para preceder la ocurrencia de diversos eventos discretos en el sistema, comenzando desde una marca inicial determinada  $M_0$ . Esto se realiza mediante la ejecución de la red de Petri; es decir, disparando todas las transiciones validadas que comienzan desde  $M_0$ . Cualesquiera transiciones conflictivas se resuelven exactamente como en el sistema original y suponemos que existe una interfaz libre de fallos, entre el modelo y el sistema real, para comunicar las decisiones de más alto nivel.

El modelo de la red de Petri de la figura 18, tiene dos tipos de transiciones. Las transiciones inmediatas que emplean tiempo cero para su disparo y las transiciones que se toman un tiempo de duración finita para

dispararse. Generalmente en los estudios de evaluación de ejecución, esta duración de tiempo se toma como una variable aleatoria con distribución exponencial.

### 1.12. Interpretación de las redes de Petri

- ◆ Una RdP interpretada (RdPI) se define por:
  - Una RdP marcada  $\langle R, M_0 \rangle$ ;
  - Una aplicación de T en E que asocia a cada Transición  $T_j$  un evento  $E_j$ ;
  - Una aplicación de T en C que asocia a cada Transición  $T_j$  una condición externa  $C_j$ ;
  - Una aplicación de T en Z, que a las transiciones les asocia salidas impulsionales.
  - Una aplicación  $P \times C$  en Y, que a los lugares les asocia salidas a nivel, eventualmente condicionadas.
  
- ◆ Regla de evolución del marcado de una RdPI.
  - Regla n°1: El disparo de una transición  $T_i$  sensibilizada solo se realiza si se verifica  $C_i E_i$ . En estas condiciones se dice que  $T_i$  es receptiva a la condición y al evento
  - Regla n°2: Cuando una transición es disparada, todas las salidas impulsionales asociadas a  $T_i$  son generadas.
  - Regla n°3: Para un marcado dado, las salidas a nivel asociadas a los lugares marcados son generadas si se verifican las condiciones externas que eventualmente le pueden estar asociadas. Si  $Y_j$  es una salida asociada a un lugar  $P_i$  condicionada por  $C_{ij}$ , se dirá que la salida es sensible a  $C_{ij}$ , cuando  $P_{ij}$  esta marcado.

Para la implementación de automatismos mediante redes de Petri interpretadas nos centraremos fundamentalmente en las RdP binarias, como

las más adecuadas para la modelización de los controladores lógicos programables (Autómatas Programables).

La idea fundamental consiste en implementar cada lugar de marcado binario mediante una memoria binaria (biestable) y establecer la correspondencia entre la puesta a "1" de la memoria, y el marcado del lugar correspondiente.

Obviamente para la implementación de RdP no binaria se deberá utilizar, para cada lugar que pueda contener  $k$  marcas, un contador modulo  $k + 1$  o mayor.

Como se ha visto, en la interpretación de las RdP se realiza una distinción en la naturaleza de las salidas asociadas a lugares y transiciones.

Se pueden clasificar dos tipos de salidas:

- Salidas de nivel.
- Salidas de impulsionales.

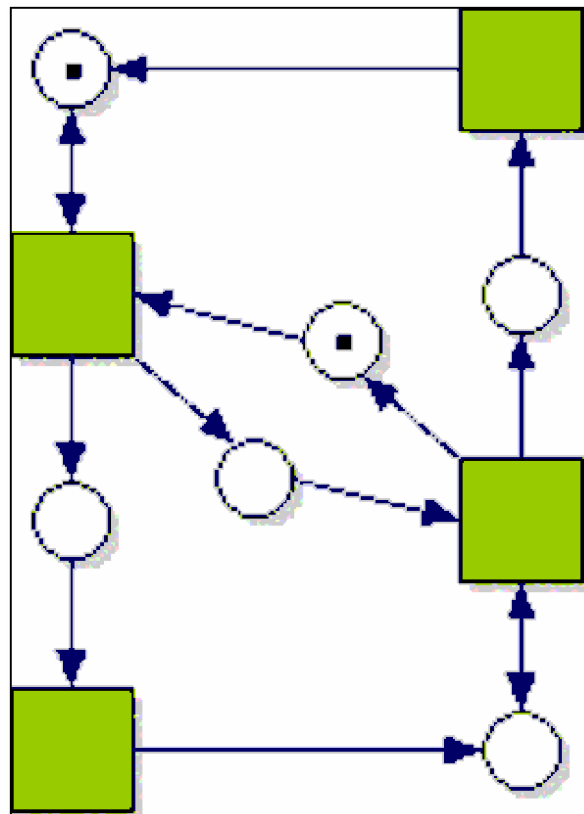
Las salidas a nivel pueden ser modeladas mediante una variable booleana.

Tendrán una duración finita en el tiempo. Por ejemplo:

- Movimiento a la derecha.
- Apertura de válvula V.

Las salidas impulsionales no tienen duración; en la práctica se asociaran a flancos de subida o bajada de variables. Estarán especialmente indicadas para ejecutar órdenes en pre-accionadores de tipo biestable.

MODELAMIENTO Y CONTROL DE SISTEMAS DE EVENTOS DISCRETOS



## 2. MODELAMIENTO Y CONTROL DE SISTEMAS DE EVENTOS DISCRETOS

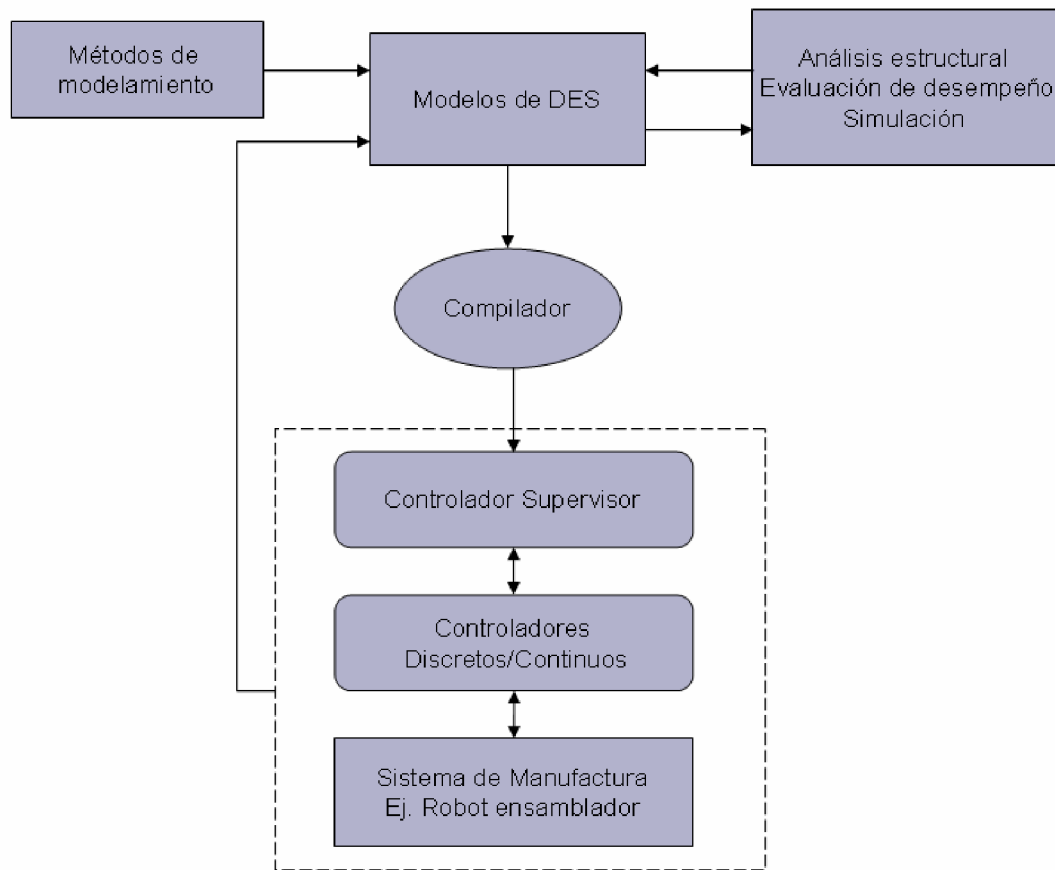
Los sistemas de eventos discretos exhiben muchas características tales como; la concurrencia, el no determinismo, el asincronismo entre otros. Para involucrar estas propiedades, muchos mecanismos han sido propuestos y desarrollados para el modelamiento de tales sistemas. Dentro de los cuales están las: máquinas de estado, las redes de Petri, los procesos secuenciales de comunicación e infinidad de recursos de procesos. Para poder conducir un análisis de desempeño a este tipo de sistemas han sido formulados y aplicados métodos tales como el análisis de perturbación y los procesos de Markov entre otros, para poder diseñar controladores supervisores y modelos basados en autómatas para tales sistemas de eventos.

### *2.1 Estructuras Jerárquicas De Control*

Para los sistemas de manufacturas y otros sistemas distribuidos, la estructura de controles jerárquicos ha sido discutida en la literatura de Menon<sup>2</sup>. En este trabajo las estructuras de tiempo real de sistemas de *eventos manejados* son vistas como estructuras de dos niveles. El primer nivel de control es un control supervisorio o control de relación de eventos y el segundo nivel es control de procesos el cual se puede descomponer más, y es frecuentemente implementado en retroalimentación de lazo cerrado continua o en controladores discretos.

---

<sup>2</sup> MENON, S. R., QUINN, P. M., KAPOOR, S. G. Coordination control of flexible manufacturing systems using colored Petri Nets. 1988. Pág. 317-326.



**Figura 19.** Arquitectura de control de un DES real

Tales estructuras de control pueden ser observadas refiriéndose a un sistema de manufactura con arquitectura como la mostrada en la figura 19. Por ejemplo, cuando se da un comando “mover el robot del punto A al punto B” esto es abordado por el controlador supervisor el robot comienza a moverse desde el punto A al punto B. el proceso “mover” puede ser un proceso continuo el cual es controlado por un controlador especial. En ambiente computarizado, tal sistema de control jerárquico puede ser implementado de dos métodos básicos: control centralizado y control distribuido.

En el método de control centralizado, un computador huésped (host) es usado para realizar y sincronizar todas las actividades de eventos relacionados y controlar todos los procesos de control de bajo nivel.



En el método de control distribuido, los componentes del sistema pueden ser subsistemas independientes basados en procesadores, la comunicación entre estos se vuelve muy importantes. Un controlador supervisor necesita ser diseñado para la coordinación de las tareas asincrónicas entres estos componentes distribuidos para satisfacer los requerimientos del sistema.

De la figura 19, podemos ver que los modelos de DES pueden ser construidos usando aproximaciones de modelamiento y herramientas para DES, entonces, tales modelos pueden ser usados para diseñar controladores supervisores ya sea a partir de un compilador o de un algoritmo. Otras aplicaciones de estos modelos incluyen análisis estructural, evaluación de desempeño y simulación para DES. Se puede también apreciar que un modelo para DES puede directamente convertirse en un controlador supervisor, como se discute a continuación.

## *2.2. Modelos de redes de Petri vs. Controladores de redes de Petri*

Los modelos de redes de Petri para sistemas de manufactura han sido ampliamente usados para su validación; la lógica de control, la simulación, y análisis de desempeño. Estos también pueden ser implementados como controladores supervisorios de un sistema, ya que estos absorben los procesos de bajo nivel como lugares o transiciones en los modelos de redes de Petri. Estos procesos de bajo nivel pueden ser representados como componentes de software/hardware<sup>3</sup>.

Los controladores están representados como códigos de control en un ambiente computarizado. Tal controlador, es llamado controlador de red de Petri y puede ser definido como un tipo de control lógico de una red dada, para sistemas de eventos discretos. El flujo de tokens a través de los lugares está

---

<sup>3</sup> ZHOU, MengChu. DICESARE, Frank. Petri Net Synthesis for Discrete Event Control of Manufacturing Systems. Kluwer Academic Publisher. Londres 1993.

regulado por las transiciones de disparo en la red dada. Es este flujo de tokens lo que define el control supervisorio de acciones.

Los componentes de software son diferentes de los módulos o de las subrutinas en programación modular. Estos exponen tres características básicas:

- Una interfase publica bien definida
- Una implementación interna que es accesible al usuario
- Ambas, la parte visible y la inaccesible deben ser compiladas de forma separada de los componentes del programa que se use.

Estas características permiten la prevención de errores de programación, extensos chequeos de errores en la compilación, una mayor confiabilidad en el programa y hace que los componentes del software sean de conexión compatible. Los componentes de software/hardware son la extensión del concepto de componentes de software que incluye los componentes de manufactura como robots o máquinas.

La idea básica de un “compilador” puede ser desarrollada de tal forma que la red de Petri sea transformada en un programa digamos por ejemplo C++ que incluya los componentes software/hardware

### *2.3 Redes de Petri vs. Otros modelos en modelamiento y control*

Un sistema de *manejo eventos* puede ser extraído como en una máquina de estados, en la cual el estado cambia cuando el evento ocurre. La máquina de estado finita o el modelo de autómatas, resulta cuando el total de números de estados en el sistema es finito. La máquina de estado está bien establecida como el modelo fundamental para la computación y las máquinas de computación. Sin embargo, cuando son usadas para modelar DES en modo

de línea recta o hacia delante, el incremento exponencial en el número de estados hace muy difícil la implementación de DESs complejos. La representación grafica es prácticamente imposible y su visualización casi no se puede realizar.

Otros métodos para el modelado y el control de sistemas de eventos discretos, han sido desarrollados de forma reciente, dentro de los cuales se encuentran los lenguajes basados en la teoría de control supervisorio y una infinidad de procesos recursivos. En el control supervisorio, la teoría es elegante y es independiente de los modelos usados para la aplicación. En la mayoría de aplicaciones, cada evento discreto procesado, se asume será modelado por un autómata o una máquina de estado y su comportamiento descrito totalmente por el lenguaje generado por el autómata.

Todos los requerimientos del sistema o sus especificaciones, se asume serán especificadas como lenguajes, por eso, un problema de diseño para los controladores supervisorios de sistemas de eventos discretos será: encontrar un autómata el cual sea un controlador supervisorio, tal que, el autómata combinado para el controlador del sistema, genere el lenguaje especificado. Un controlador supervisor controla un DESs, habilitando o deshabilitando un evento controlado<sup>4</sup>.

La aplicabilidad en el mundo real de sistemas distribuidos esta limitada por el uso de la representación de la máquina de estado; por consiguiente cuando estas son usadas para describir un sistema complicado, el problema de diseño puede volverse fácilmente inmanejable. En adición a que especificar el lenguaje para un sistema no es realmente fácil.

---

<sup>4</sup> ZHOU, MengChu. DICESARE, Frank. Petri Net Synthesis for Discrete Event Control of Manufacturing Systems. Kluwer Academic Publisher. Londres 1993.

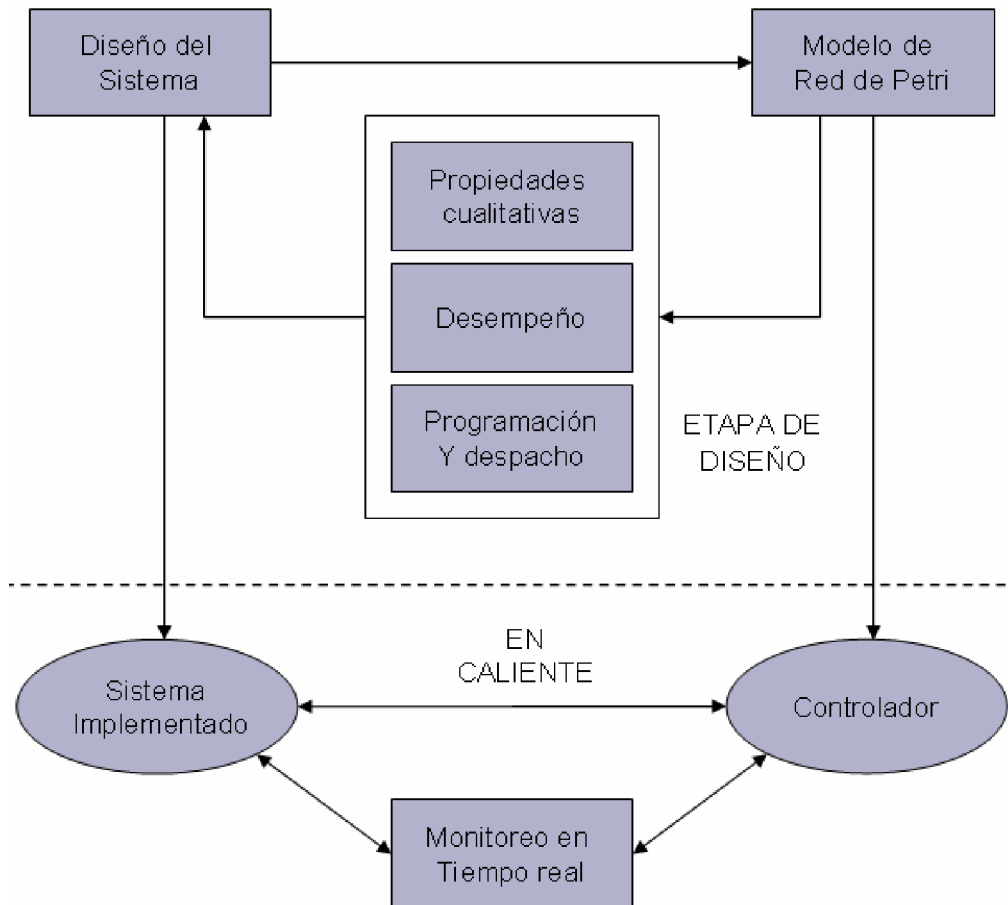
Los procesos recursivos finitos (FRP) han sido propuestos para describir sistemas de eventos discretos o modelos de eventos discretos (DEM). Los FRP están principalmente basados en los procesos de comunicación secuencial de Hoare`s. en la formulación del FRP, dados una serie de eventos, un proceso es definido como una tripleta de elementos: un conjunto de trazas, en que el proceso puede ejecutar un evento, una función o la terminación de una función. Uno de sus importantes rasgos es que cada proceso puede ser descrito como un conjunto de ecuaciones recursivas. Esto implica que la descripción de un sistema puede ser implementada usando formas de ecuaciones. No obstante muchos problemas aún se mantienen sin resolver: por ejemplo, el uso de tales ecuaciones para diseñar controladores supervisorios para sistemas de tiempo real.

#### *2.4 Ventajas de las Redes de Petri*

Las redes de Petri son una herramienta grafica que provee un método unificado para el diseño de DES desde descripciones de sistemas jerárquicos hasta sus realizaciones físicas. Comparados con otros métodos discutidos estas tienen las siguientes ventajas:

- Facilidad para modelar las características de los DES, tales como: asincronismo, concurrencia, sincronismo, conflictos, exclusión mutua, relaciones precedentes, no-determinismo y las parálisis del sistema.
- Excelente visualización de las dependencias del sistema.
- Focalización de la información local.
- Su diseño se puede realizar de abajo a arriba o viceversa.
- Habilidad para generar códigos para control supervisor, directamente de las graficas de Petri.
- Habilidad de chequear el sistema por propiedades indeseable, tales como: parálisis e inestabilidad.

- Análisis del desempeño para muchos sistemas sin necesidad de simulación. También las tasas de producción y utilización de recursos pueden ser evaluadas.
- La simulación de eventos discretos puede ser manejada desde el modelo.
- Permite ver el estatus en tiempo real de la información.



**Figura 20.** Redes de Petri como una representación de modelamiento, análisis y control.

Una simple representación de la herramienta es mostrada en la figura 20, las redes de Petri se añaden en: modelamiento, análisis, validadas, simuladas, verificadas y evaluadas en la etapa de diseño. Una vez el sistema demuestre el comportamiento deseado, la red se puede convertir en un control y un monitor de operaciones en tiempo real. Por eso, las redes de Petri pueden ser

tanto una herramienta matemática poderosa como una herramienta grafica de diseño para varios sistemas de eventos discretos.

### *2.5. Métodos de análisis.*

Las técnicas para el análisis de RdP (análisis cualitativo de redes) se clasifican normalmente en tres grupos<sup>5</sup>.

#### *2.5.1. Técnicas Enumerativas.*

En primer lugar se tienen las técnicas enumerativas estas se basan en la generación del grafo de alcanzabilidad para sistemas limitados o del grafo de cobertura para sistemas no limitados. Estas técnicas se pueden aplicar en teoría, pero en la práctica están limitadas a sistemas pequeños debido a su elevada complejidad computacional. En una red no limitada, el número de señales en un lugar puede ser infinito. Un lugar que puede llegar a tener un número infinito de señales se representa por  $\omega$ . El grafo resultante es llamado grafo de cobertura.

#### *2.5.2. Técnicas de Transformación.*

En segundo lugar se tienen las técnicas de transformación, en este grupo de técnicas el objetivo es reducir el tamaño de los modelos mediante reglas de reducción que preserven las propiedades que se quieren estudiar. Con este conjunto de reglas es posible reducir la complejidad del cálculo de la vivacidad y limitación de un sistema.

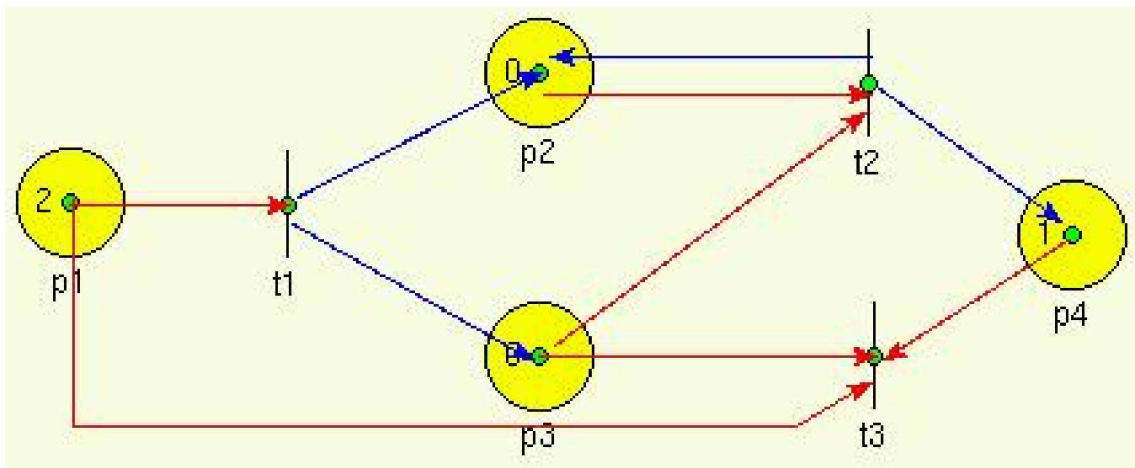
#### *2.5.3. Técnicas Estructurales*

En tercer lugar se tienen las técnicas estructurales. En este grupo de técnicas el objetivo es obtener la máxima información del modelo utilizando conceptos básicos.

---

<sup>5</sup> CERVANTES, Jair. Representación y aprendizaje de conocimiento con Redes de Petri difusas. México. 2005.

## MÉTODOS DE MODELADO



### 3. MÉTODOS DE MODELADO

Los métodos de modelado son la llave para la aplicación de las redes de Petri para los sistemas de manufactura. Debido a que con estas es que se logra modelizar y poner en uso esta herramienta.

Para realizar la modelación se debe tener las especificaciones de los sistemas de manufactura, ya que con base a estas es que se implementa la herramienta. Las especificaciones pueden ser consideradas como un conjunto de recursos, un conjunto de operaciones y sus relaciones precedentes.

#### 3.1. Pasos para el modelado.

En el siguiente método se utilizará las especificaciones para crear un modelo de red de Petri para un sistema determinado.

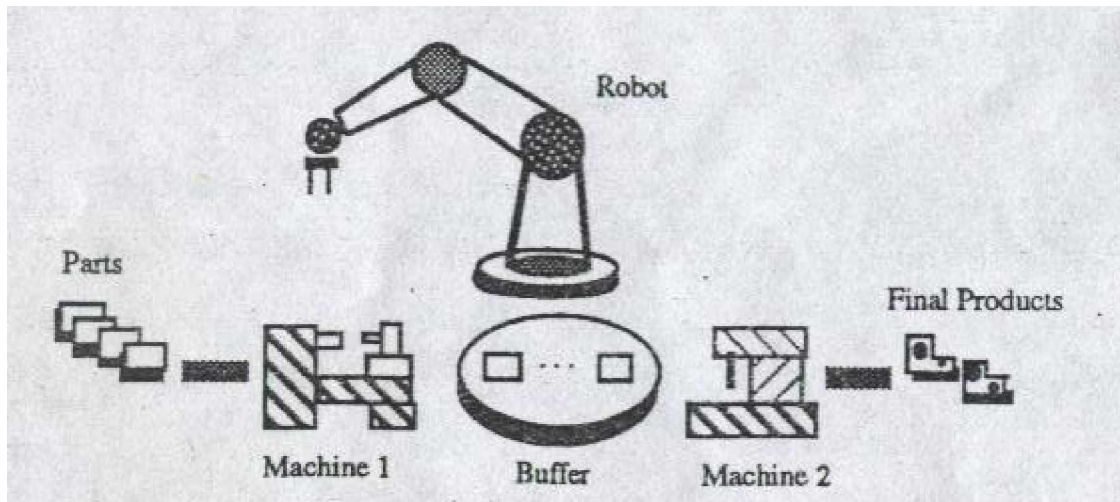
1. Identificar las actividades y recursos necesarios para la producción de un ítem de cada producto.
2. Ordenar las actividades por relaciones precedentes de acuerdo al plan de procesos.
3. Para cada actividad siga: cree y etiquete un lugar que represente el estatus de esa actividad; añada una transición (start activity) con arco(s) de salida hacia el lugar (o lugares); añada una transición (stop activity) con arco(s) de entrada del lugar (o lugares) de actividad. En general, la transición de parada para una actividad será la misma que comience la próxima actividad. Un token es un lugar de actividad que indica que la actividad esta tomando lugar cuando la red es ejecutada. Varios tokens indica que ocurren múltiples actividades.



4. Para cada actividad siga; si un lugar no ha sido creado, créelo y etiquételo para cada recurso, el cual debe estar disponible para comenzar la actividad. Conecte apropiadamente todos los lugares de recursos disponibles con arcos, como cada entrada al inicio de la transición para la actividad. Cree arcos de salida para conectar la transición de parada, siguiendo la actividad hacia cualquier lugar de recurso que represente recursos que deben quedar disponibles una vez se complete la actividad.
5. Especifique la marca inicial para el sistema.

### 3.2 Ejemplo de Aplicación II: (Sistema de Manufactura)

Para ilustrar la metodología del modelo, tomaremos un sistema simple de manufactura como se muestra en la siguiente figura.



**Figura 21.** Sistema de manufactura

El sistema consiste de dos máquinas, M1 y M2 de carga automática, un robot compartido, R para la descarga y un almacenamiento intermedio o buffer con dos espacios, B, para el almacenamiento de las partes intermedias.

El plan de procesos para cualquier parte es hacia delante pasando por M1 y después por M2. Las partes entran al sistema en M1, cada una es automáticamente cargada en M1 para su procesamiento, después de ser procesada las partes son descargadas por el robot R, en el buffer de almacenamiento intermedio, siendo las partes intermedias cargadas de forma automática en M2, para ser procesadas. Cuando M2 termina de procesar una parte; el robot R, descarga el producto final lo libera de la estiba y regresa a fijar más en M1. Se asume que las partes en la entrada están siempre disponibles a ser fijada a una estiba y que el producto final debe ser liberado de ella.

### *3.2.1. Siguiendo la metodología*

1. Las actividades requeridas son: maquinaria (fijación de estibas, cargar y procesar), almacenaje y descarga. Los recursos son M1, M2, R, B, estibas y partes

2. El orden de las actividades es:

M1P: M1 carga, fija y procesa la parte estibada.

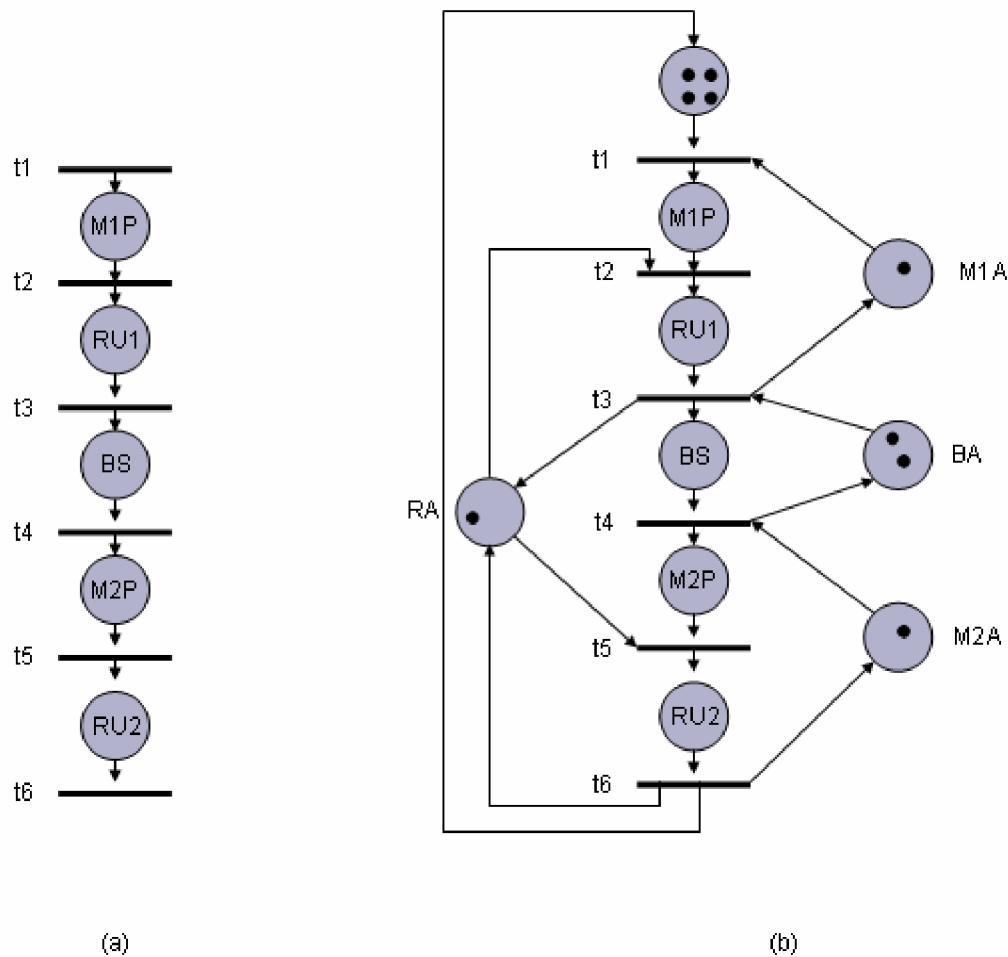
RU1: R descarga una parte intermedia en el buffer.

BS: B almacena una parte intermedia.

M2P: M2 carga y procesa una parte intermedia.

RU2: R descarga el producto final de M2, libera la estiba y la regresa.

3. Como se muestra en la figura 22. (a), los lugares M1P, RU1, BS, M2P, y RU2 son creadas para modelar la secuencia para una parte. La transición t1 modela el comienzo de la actividad M1P; t2 modela la parada de la actividad M1P y el comienzo de la actividad RU1; t3 la parada de RU1 y el comienzo de la actividad de almacenamiento BS; t4 la parada de BS y el comienzo de la actividad M2P. De igual forma t5, modela la parada de la actividad M2P y el comienzo de RU2 y t6, la parada de RU2.



**Figura 22.** (a) Grafo de RdP para la secuencia del ejemplo del sistema de manufactura, (b) RdP con recursos, lugares, arcos y marca inicial.

4. Para la actividad MP1 se requiere una estiba y la máquina M1 se encuentren disponibles. En la figura 22. (b), se muestra un modelo de red de Petri con un lugar PA que representa las estibas disponibles, y el lugar M1A que representa la disponibilidad de la máquina M1. Cada una tiene un arco de entrada a la transición t1. En la siguiente se considera a t2 como la transición de parada para la actividad M1P. Cuando la actividad de maquinaria esta completa, ningún recurso puede ser liberado ya que la máquina debe ser descargada antes de estar disponible y una estiba será usada durante todo el proceso de manufactura.

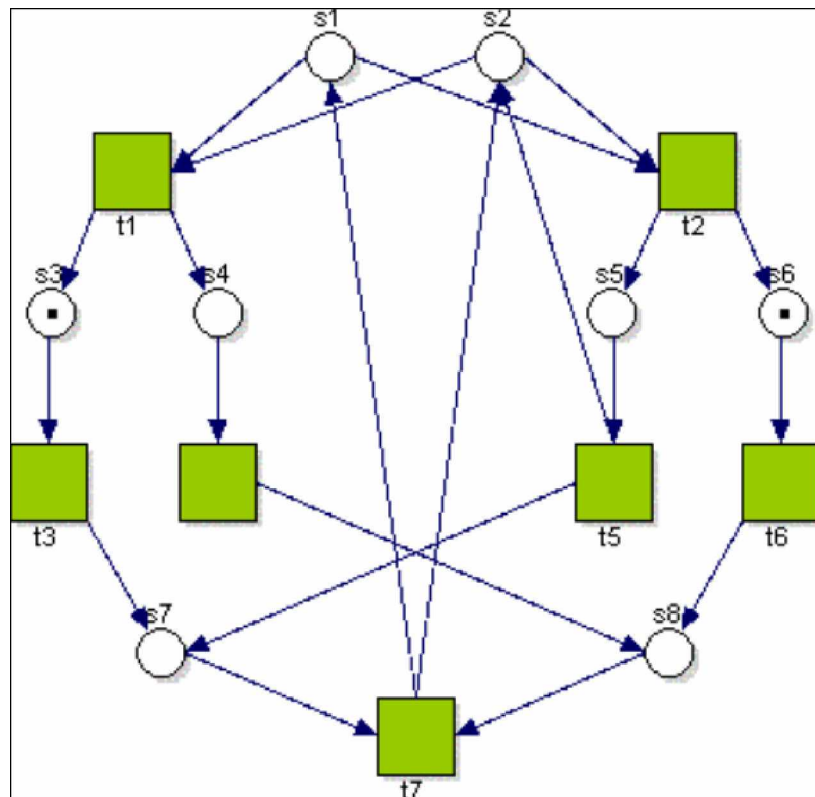
Considerando  $t_2$  como el comienzo de la descarga de M1 por el robot, actividad RU1, el único recurso adicional requerido es el robot R. Esto es representado por el lugar RA que nos dice que el robot esta disponible haciendo un arco desde RA hasta  $t_2$ . Siendo  $t_3$  la transición parada para RU1, la liberación de M1 se modela como un arco de salida desde  $t_3$  hacia M1A y la liberación de R como un arco de salida desde  $t_3$  hacia RA.

Luego consideramos que la actividad del buffer BS. Para poder almacenar las partes intermedias en el buffer, este debe espacios vacíos. Esto es representado por el lugar BA en el cual el número de tokens indica el número de lugares vacíos, y los arcos desde BA hacia  $t_3$  la transición de comienzo de BS. La finalización de la actividad de almacenamiento de una parte y la liberación de un espacio esta representada por la transición  $t_4$  y el arco de  $t_4$  hacia BA.

Las actividades y recursos para M2P y RU2, están modeladas de la misma forma como M1P y RU1; exceptuando la salida de  $t_6$  a PA. El cual representa el retorno de una estiba vacía después de que M2 es descargado y el producto le es quitado la estiba.

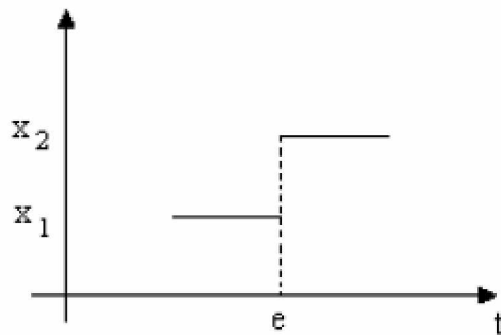
5. La marca inicial esta dada por el startup del sistema. En la cual se encuentran cuatro estibas disponibles para partes, esto representado por cuatro tokens en PA; ambas máquinas y el robot se encuentran disponibles como se muestra (cada una con un solo token M1A, M2A y RA respectivamente). Existen dos lugares vacíos para partes en el buffer de almacenamiento intermedio. Nótese que en la marca inicial solo la transición  $t_1$  se encuentra habilitada.

# REDES DE PETRI INTERPRETADAS PARA CONTROLADORES LÓGICOS PROGRAMABLES



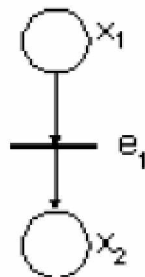
#### 4. REDES DE PETRI INTERPRETADAS PARA CONTROLADORES LÓGICOS PROGRAMABLES.

El fundamento de la modelación de los sistemas secuenciales mediante RdP es concebir éstos como sistemas a eventos discretos, en los cuales el sistema evoluciona de un estado a otro por la ocurrencia de un evento.



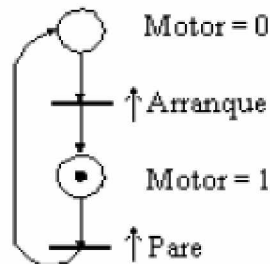
**Figura 23.** Sistema a eventos discretos.

En la figura 23, el sistema evoluciona del estado  $x_1$  al estado  $x_2$  por la ocurrencia del evento  $e_1$ . El modelo que representa esta situación es el siguiente:



**Figura 24.** Esquema en RdP de la figura 23.

Una RdP modela un DES, asociando los estados a los lugares y los eventos a transiciones. Por ejemplo, el modelo de la figura 25, representa la función de memoria set/reset aplicada al arranque y paro de un motor:

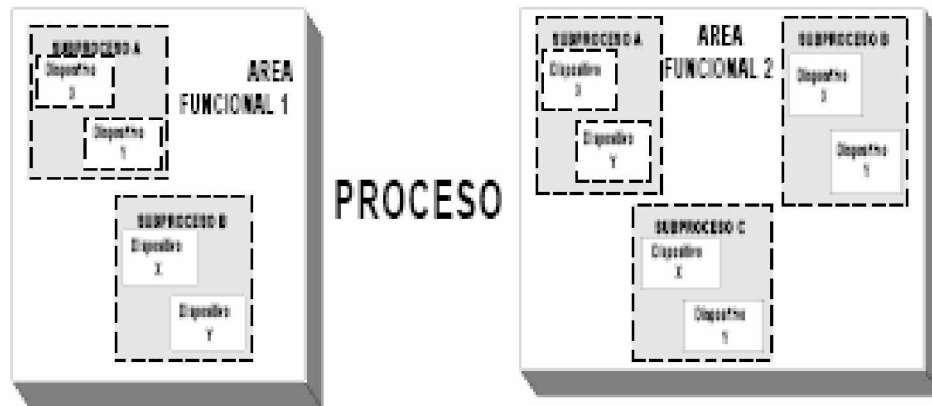


**Figura 25.** Función de arranque y paro de un motor.

El marcaje indicará el estado actual del sistema. En la figura anterior, el marcaje indica que el motor está encendido. Una vez presentado el marco conceptual, se procede a presentar el diseño de las estructuras, en las que se empleó como fundamento, la teoría expuesta de las RdP.

#### 4.1. Estructuras Generalizadas

La metodología que se propone para la solución de sistemas secuenciales utilizados en la automatización de procesos industriales identifica elementos comunes en los procesos, denominados dispositivos. Estos dispositivos interrelacionados conforman subprocesos, los que a su vez hacen parte de las áreas funcionales de los procesos, como se muestra en la figura 26.



**Figura 26.** Estructura general de un proceso

Los dispositivos o estructuras generalizadas se modelan mediante RdP, se simulan y se validan mediante una herramienta formal de análisis, como los Grafos de ocurrencia (Murata, 1989) y (Kurt, 1997)<sup>6</sup>. Posteriormente son codificadas en lenguajes de controladores lógicos como bloques funcionales parametrizables.

Los dispositivos son los diferentes actuadores utilizados en los procesos industriales, los cuales en su forma más básica poseen una señal de mando y una señal de salida que maneja un elemento final de control, existiendo gran cantidad de variantes dependiendo del tipo y la forma como operan.

Los más utilizados se enumeran en la Tabla 2.

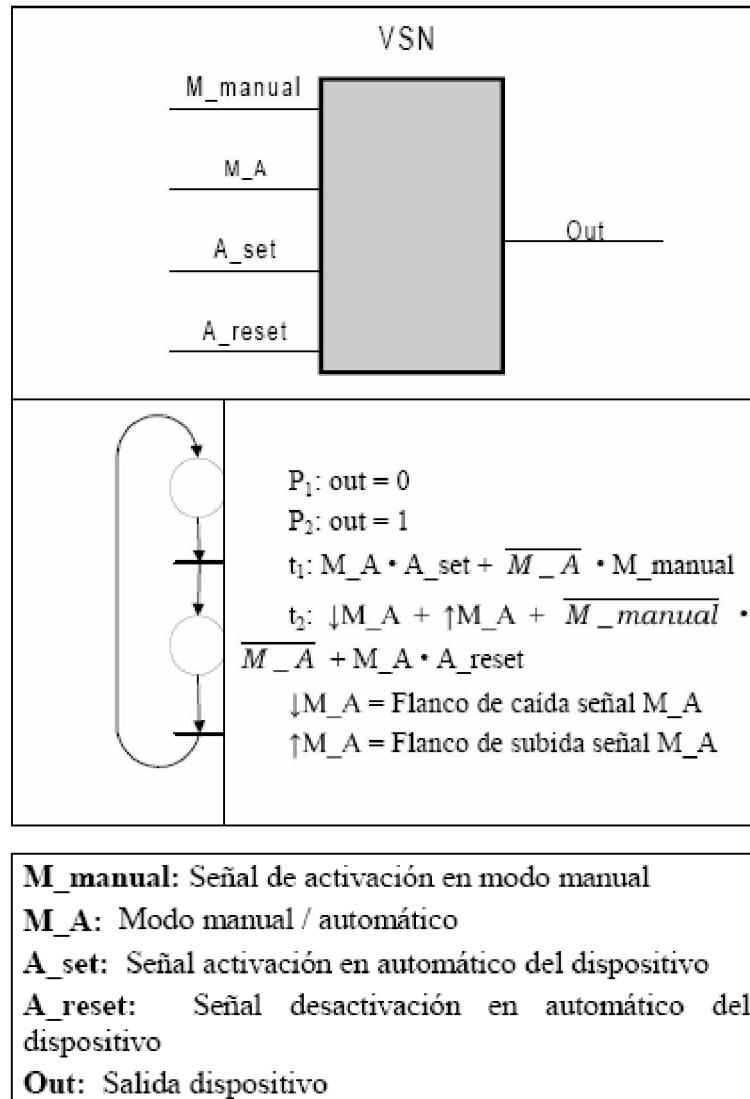
<sup>6</sup> ZAPATA German, CARRASCO Eliana. Estructuras Generalizadas para Controladores Lógicos modelados mediante Redes de Petri. UNAL Medellín 2002.



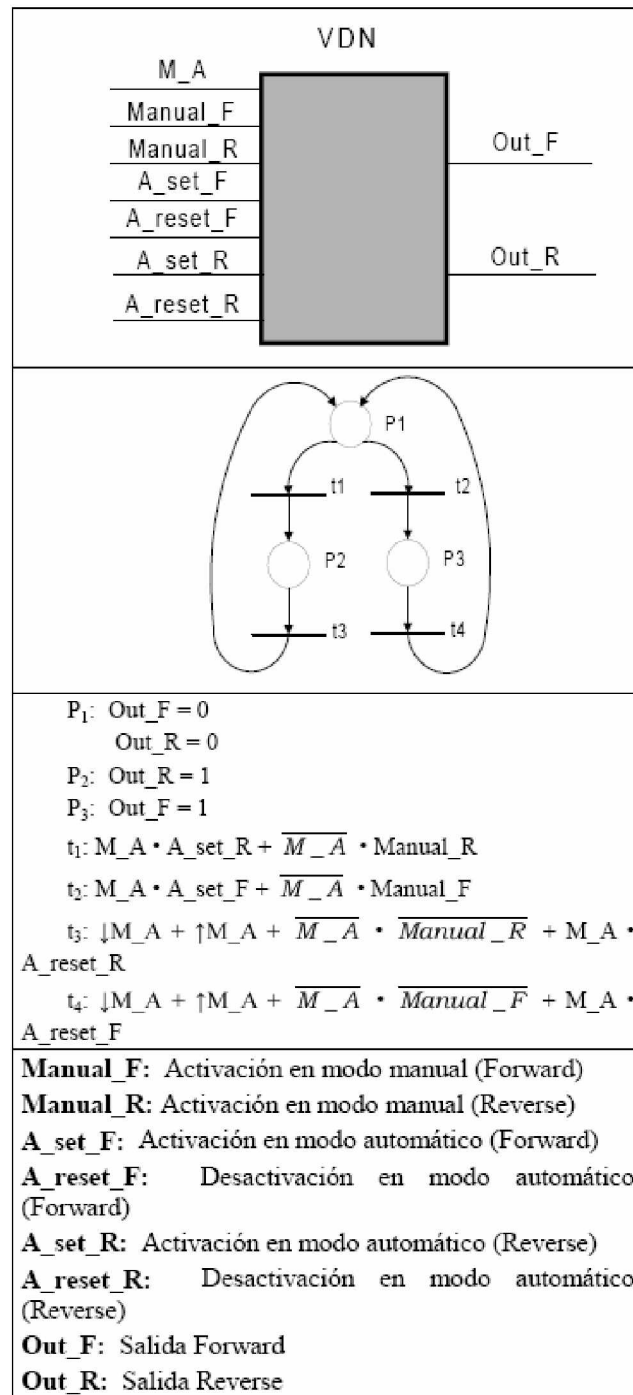
<b>DISPOSITIVO</b>	<b>SIM</b>	<b>ACCIÓN</b>
<b>Válvulas</b>	VSN	Simple efecto/sin confirmación
	VSS	Simple efecto/simple confirmación
	VSD	Simple efecto/doble confirmación
	VDN	Doble efecto/sin confirmación
	VDD	Doble efecto/doble confirmación
<b>Motores</b>	MSN	Simple mando/sin realimentación
	MSS	Simple mando/con realimentación
	MSR	Reversible forward/reverse
<b>Cilindros</b>	CSD	Actuador sencillo/doble realimentación. Existen dos variantes -Estado energizado extendido -Estado energizado retraído

**Tabla 2. Dispositivos**

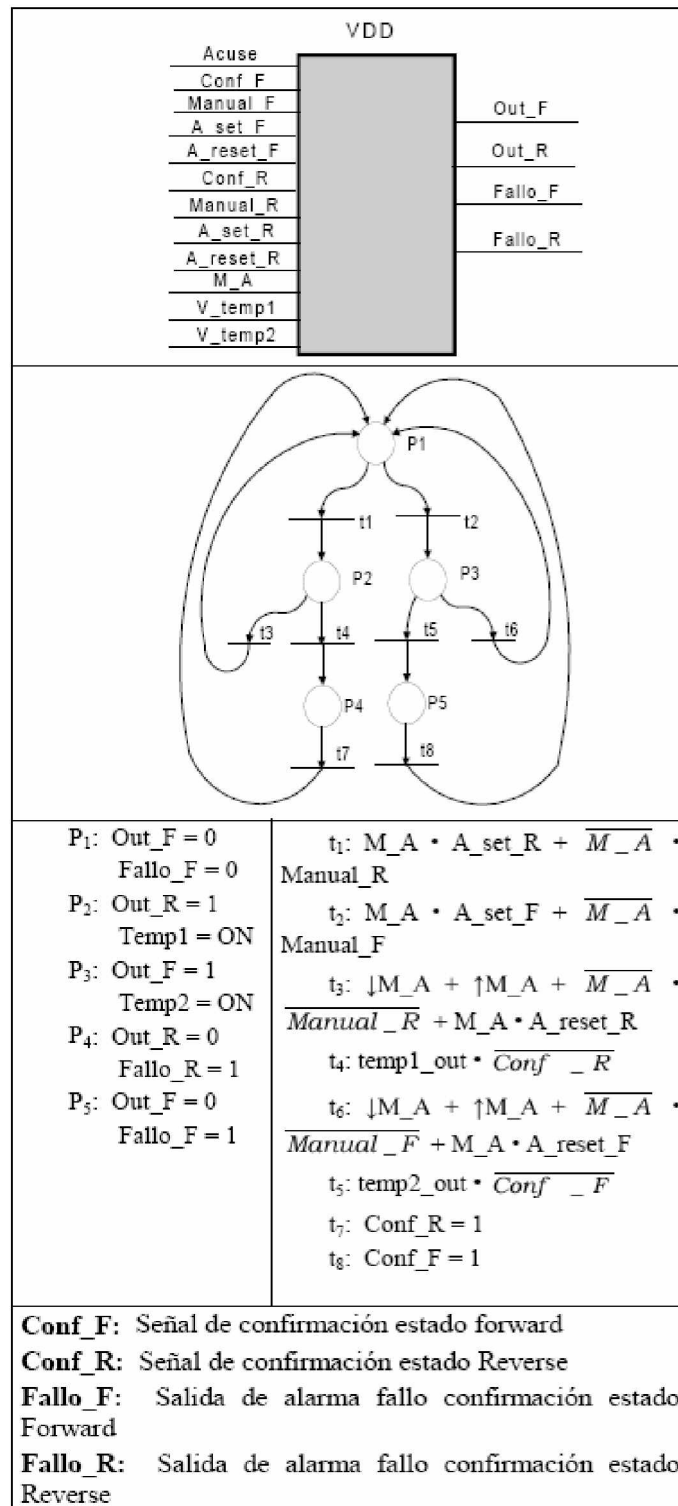
La importancia del diseño estructurado permite considerar los mandos o modos de marcha, las seguridades y el control supervisorio. Los modelos y la representación como estructura generalizada de algunos de estos dispositivos se muestran en las Tablas 3, 4 y 5.



**Tabla 3.** Dispositivo Simple efecto / sin confirmación



**Tabla 4.** Dispositivo Doble efecto / sin confirmación




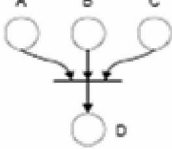


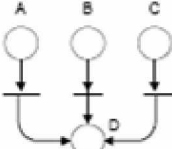
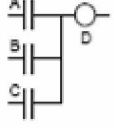
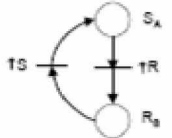
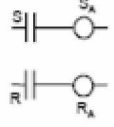


**Tabla 5.** Dispositivo Doble efecto/doble confirmación

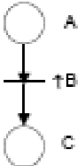
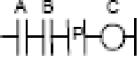
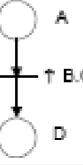
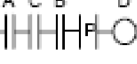
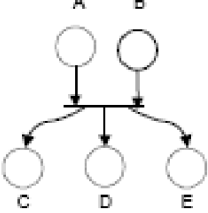
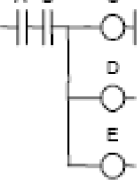
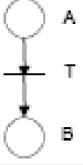
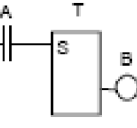
#### 4.2. Equivalencia con PLC's.

Los procedimientos de diseño de sistemas secuenciales mediante RdP deben contar además con una metodología para su implementación en equipos digitales. A nivel industrial, el equipo de uso más extensivo es el PLC (Programmable Logic Controller), por sus características de robustez, flexibilidad y su concepción como dispositivo de propósito general.

Para la traducción del modelo del sistema a un lenguaje de PLC normalizado se tendrán en cuenta las representaciones de la tabla 6:

CONSTRUCCION LÓGICA	RdP	RLL (Relay Ladder Logic)
Condición o estado de un elemento del sistema	 Lugar	Sin representación
Actividad, acción o evento	 Transición	Sin representación
Elemento activo	 Marca	Sin representación
Función lógica "y"	 	
Función lógica "O"		
Función memoria (S/R)		

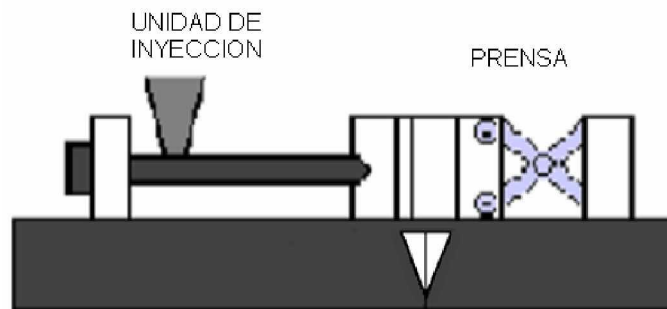
**Tabla. 6.** Representación de la redes de Petri a RLL.

CONSTRUCCION LÓGICA	RdP	RLL (Relay Ladder)
Flanco		
Receptividad		
Concurrencia		
Retardo		

**Tabla. 6.** Representación de la redes de Petri a RLL.

#### 4.3. Ejemplo de Aplicación III: (Inyectora de Plástico)

La metodología de diseño estructurado fue aplicada al automatismo simplificado de una Inyectora de plástico, mostradas en la figura 27.



**Figura 27.** Inyectora de plástico.

Las unidades funcionales son las especificadas en la Tabla 7.

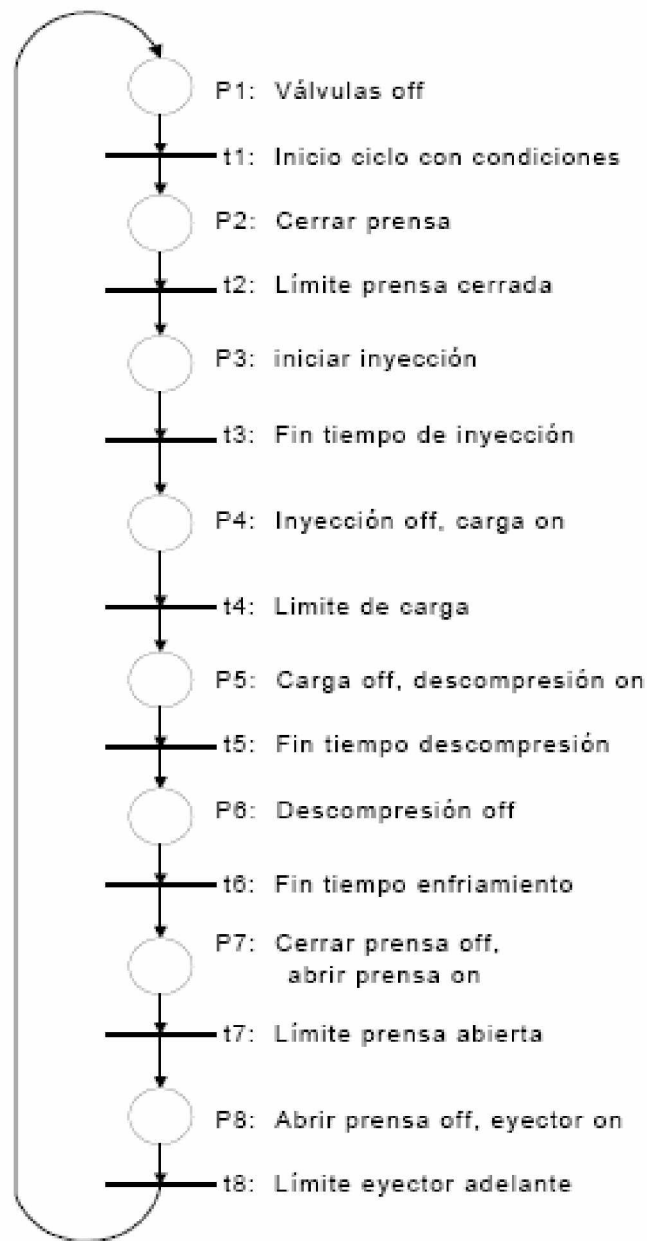
<b>AREA FUNCIONAL</b>	<b>ACCIÓN</b>	<b>DISPOSITIVO</b>
PRENSA	Abrir / cerrar	VDS
UNIDAD INYECCIÓN	Inyectar / descomprimir	VDS
	Cargar	VDN

**Tabla 7.** Componentes de la Inyectora de plástico

La Prensa realiza el cierre a alta presión del molde que aloja las cavidades a ser inyectadas. La unidad de inyección realiza la transformación, de las propiedades del polipropileno mediante calentamiento, para permitir el proceso de inyección de una colada caliente. El eyector realiza la tarea de expulsión de la pieza inyectada.

Tabla de señales

La red de la figura 28, especifica el automatismo, el cual se resuelve con el programa que se ilustra en la tabla 8, implementado mediante las estructuras generalizadas. El programa final se muestra en la figura 29.

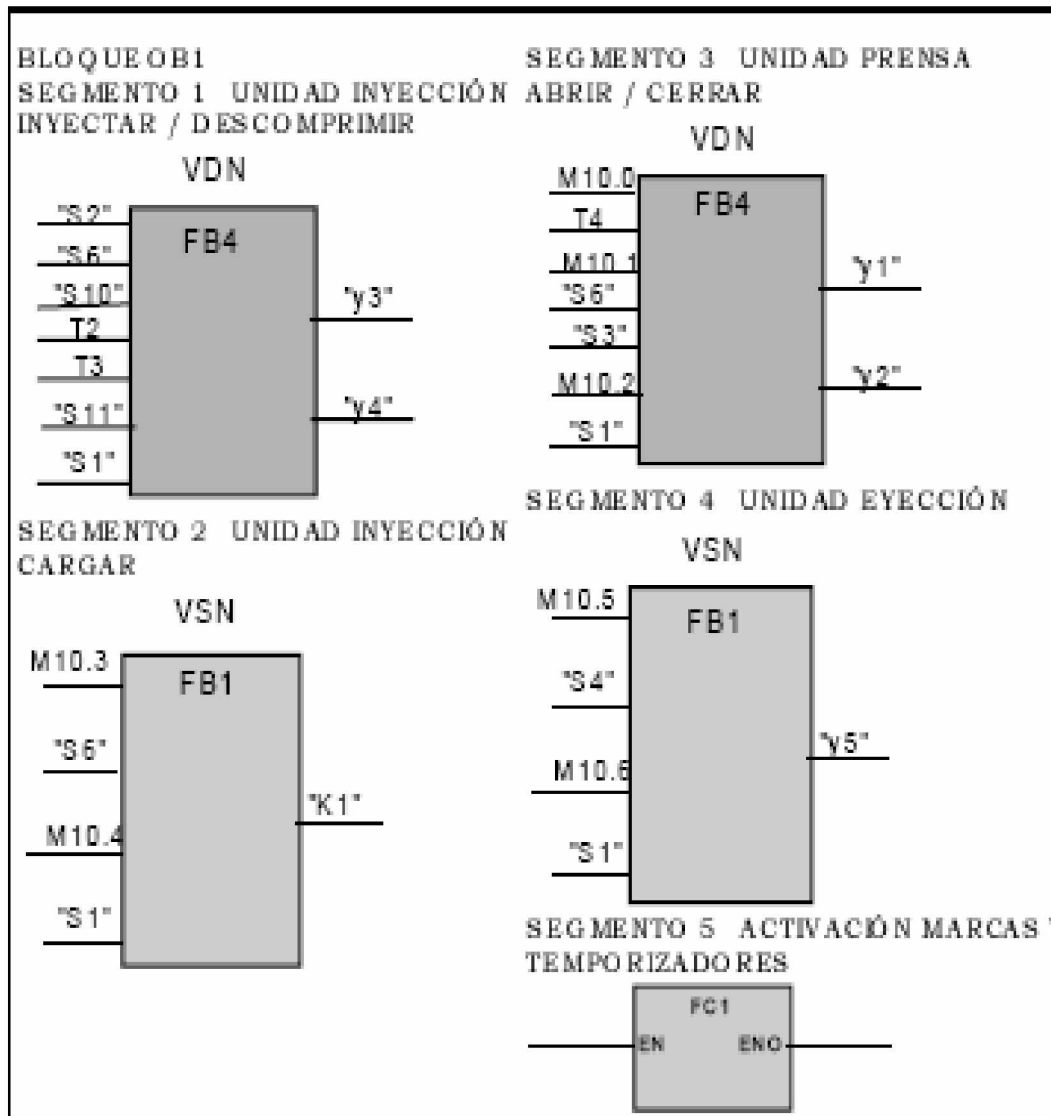


**Figura 28.** Ciclo automático Inyectora de plástico



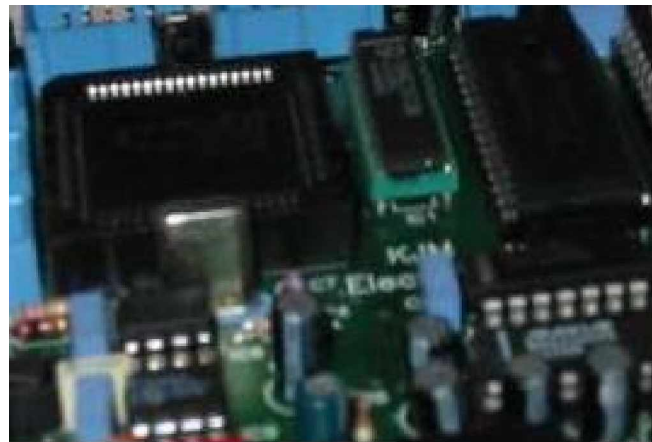
SÍMBOLO	DIRECCIÓN	COMENTARIO
Y1	A 124.0	Solenoides cerrar prensa
Y2	A 124.1	Solenoides abrir prensa
Y3	A 124.2	Solenoides inyectar
Y4	A 124.3	Solenoides descomprimir
Y5	A 124.4	Solenoides eyector adelante/atrás
K1	A 124.5	Contactores motor carga
S0	E 124.0	Pulsador inicio de ciclo
S1	E 124.1	Selector manual/automático
S2	E 124.2	Microsuiche límite de carrera prensa cerrada
S3	E 124.3	Microsuiche límite de carrera prensa abierta
S4	E 124.4	Microsuiche límite de carrera eyector adelante
S5	E 124.5	Microsuiche límite de carrera eyector atrás
S6	E 124.6	Microsuiche límite de carga
S7	E 124.7	Microsuiche puerta cerrada
S8	E 125.0	Pulsador mando manual cerrar prensa
S9	E 125.1	Pulsador mando manual abrir prensa
S10	E 125.2	Pulsador mando manual inyectar
S11	E 125.3	Pulsador mando manual descomprimir
S12	E 125.4	Pulsador mando manual cargar
S13	E 125.5	Pulsador mando manual eyector
	T 1	Tiempo de pausa
	T 2	Tiempo de inyección
	T 3	Tiempo de descompresión
	T 4	Tiempo de enfriamiento
	M10.3	Marca habilitar cargar en automático
	M10.4	Marca habilitar cargar modo manual hasta límite de recorrido
	M10.0	Marca habilitar cerrar prensa en automático
	M10.1	Marca habilitar cerrar modo manual hasta límite de recorrido
	M10.2	Marca habilitar abrir modo manual hasta límite de recorrido
	M10.5	Marca habilitar eyector en automático
	M10.6	Marca habilitar eyector modo manual hasta límite de recorrido

**Tabla 8.** Inventario de señales



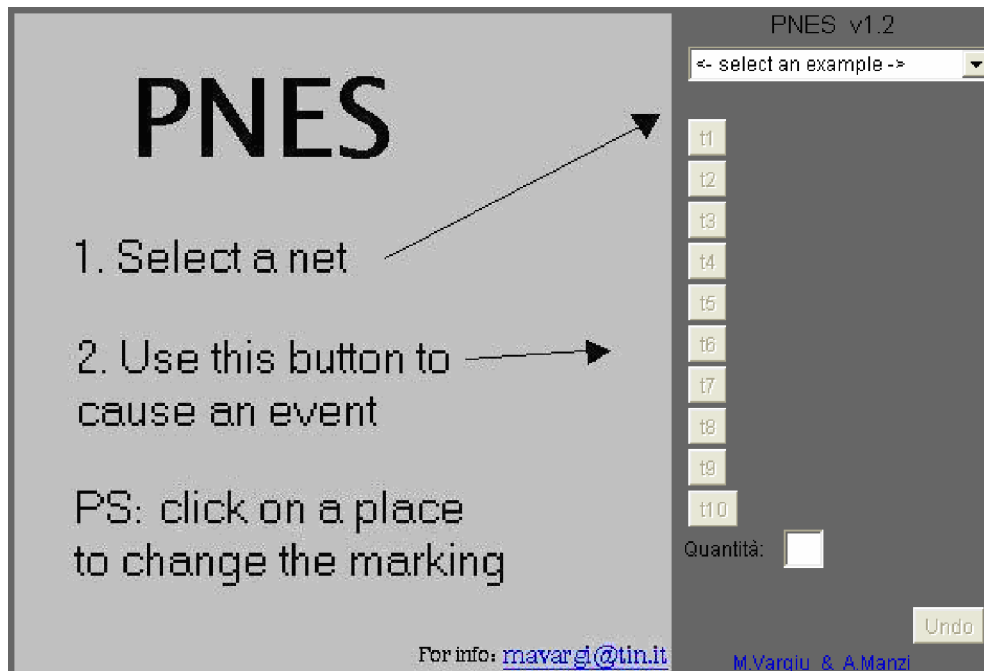
**Figura 29.** Programa de la Inyectora

# SIMULACIÓN



## 5. SIMULACIÓN

Para la simulación de las redes de Petri, se utilizó un applet de Java, desarrollado por el curso de software de ingeniería de la Universidad Degli Studi Di Cagliari, en los años 1999-2000<sup>7</sup>. El cual permite conocer los lugares, tokens y también entender de forma didáctica, como se ejecutan las transiciones en estas redes. Figura 30.



**Figura 30.** Applet para simulación.

Este programa permite seleccionar una variada serie de ejemplos donde se presenta las formas básicas en las que se puede dar una transición con uno o más tokens involucrados en la misma.

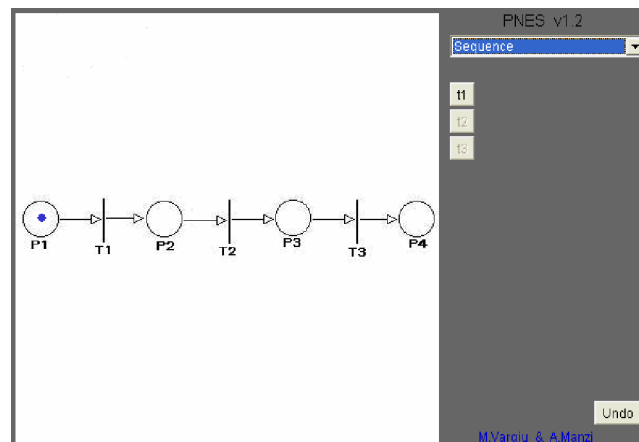
<sup>7</sup> En Internet: <http://web.tiscali.it/marconthenet/Pnes.Pnes.html>

Comenzaremos la demostración para estas redes desde su parte mas básica; como lo es ejemplificar la dinámica de su secuencia hasta uno de mayor complejidad como lo es, una celda de un sistema de manufactura.

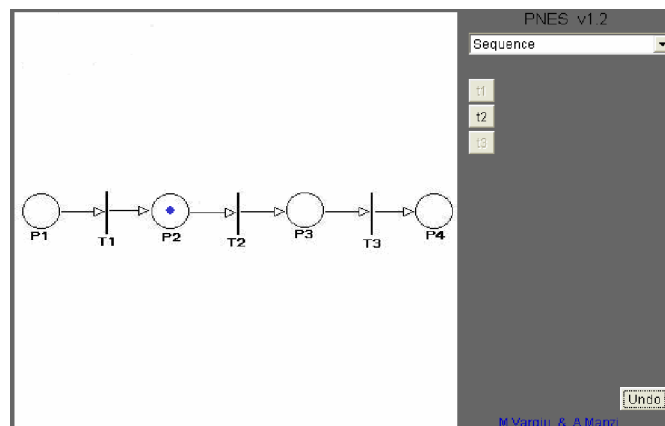
### 5.1. Ejemplo 1: Secuencia de una Red de Petri.

En la figura 31, se muestra una red de Petri hacia delante, en la que se ejemplifica como recorre los lugares de actividad a medida que se validan las condiciones para las transiciones.

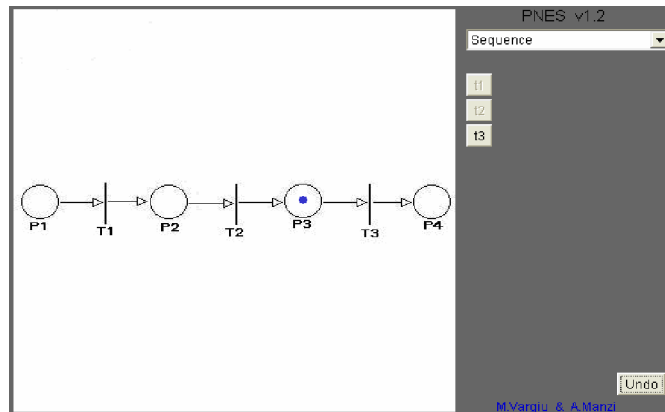
Para las figuras 32, 33 y 34, se presenta como se ejecutan todas y cada una de las transiciones al validarse las condiciones para t1, t2 y t3 respectivamente.



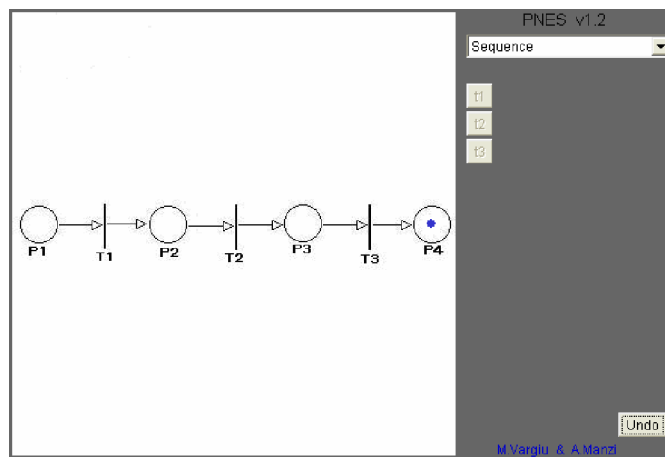
**Figura 31.** Secuencia para una red de Petri.



**Figura 32.** Transición t1, activada.



**Figura 33.** Transición *t2*, activada.

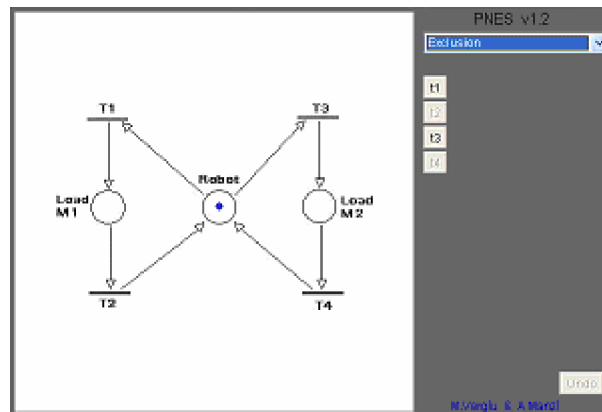


**Figura 34.** Transición *t3*, activada.

### 5.2. Ejemplo 2: Red de Petri con el fenómeno de exclusión mutua

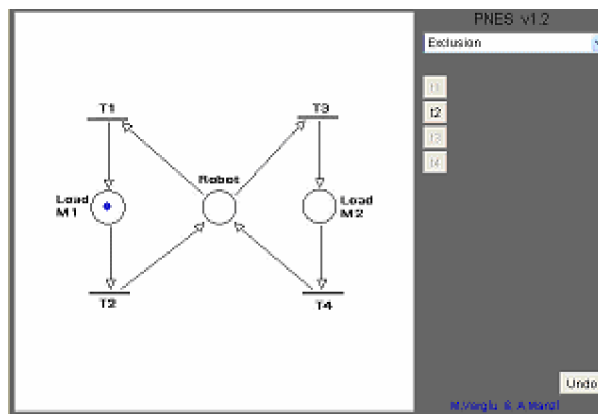
Este ejemplo plantea que un robot debe realizar dos cargas. Una en M1 y la otra en M2, sabiendo de solo puede efectuar una carga a la vez, se procede con la ejecución de una de las dos actividades.

Como se muestra en la figura 35.



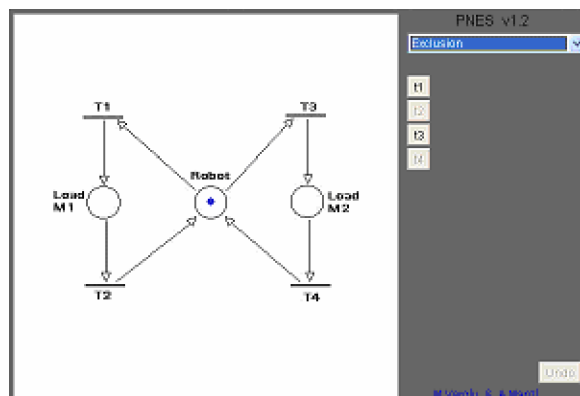
**Figura 35.** Paralelismo en RdP.

Dado el plan de proceso para este sistema, solo es posible la transición de t1 o de t3. De la cual se selecciono presionar la primera, t1. figura 36.



**Figura 36.** Transición t1, activada.

Observando el sistema se nota que la única transición posible es la transición t2, por lo tanto esta es la que se activa. Ver figura 37.

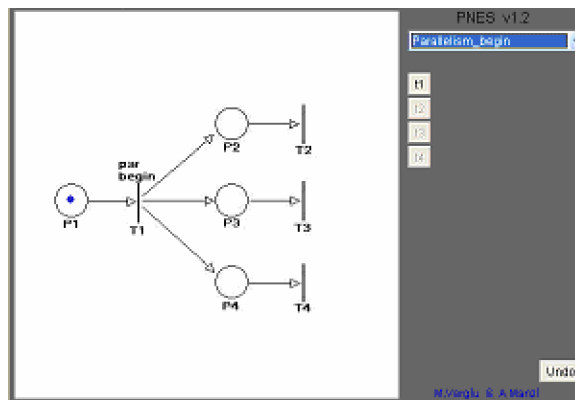


**Figura 37.** Transición t2, activada.

Al realizarse la transición t2 el robot, vuelve a quedar libre y nuevamente se presentan las dos opciones de actividad; la transición t1 o la transición t3. Si se decide activar la transición t3, ocurrirá algo muy similar a lo mostrado, cuando al comienzo se activo t1; solo que el camino a seguir será hacia t4 y luego quedara en la posición inicial. Demostrando con esto que se presenta la exclusión mutua de estas dos actividades. Como se ilustra en la figura 37.

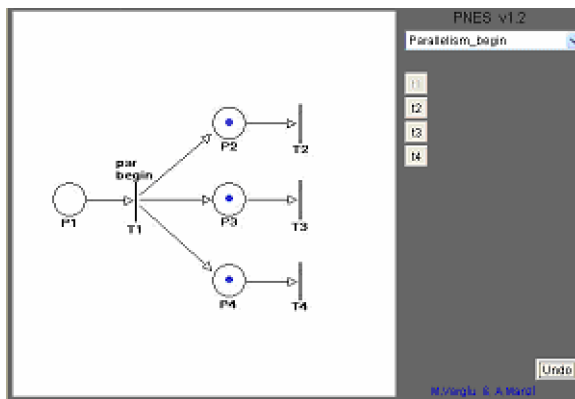
### 5.3 Ejemplo 3: Red de Petri con paralelismo.

Para esta Red mostrada en la figura 38, se tiene el lugar inicial P1, conectada a la transición T1 que luego se dividirá en tres caminos posibles con tres condiciones diferentes a cumplir o activar.



**Figura 38.** RdP con Paralelismo.

Se activa la transición t1 como se muestra en la figura 39.



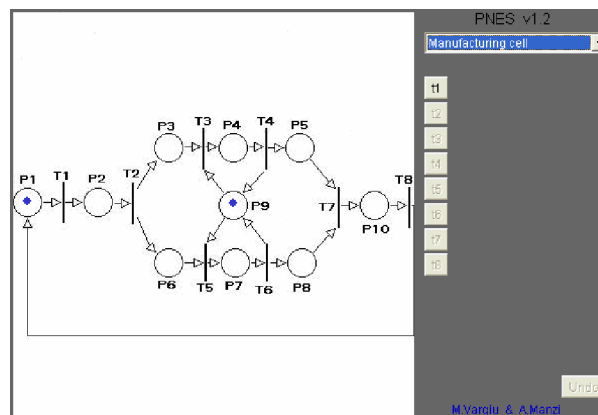
**Figura 39.** Transición t1, activada.



De esta forma se muestra que el mismo token o marca toma los tres caminos y no uno solo manifestándose así el fenómeno a demostrar; el paralelismo.

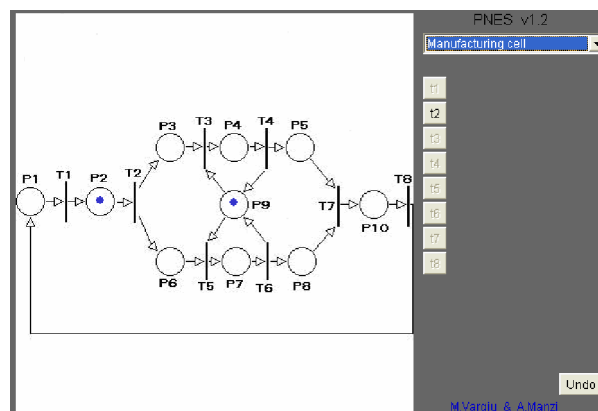
#### 5.4. Ejemplo 4: Celda de un sistema de manufactura con RdP.

Para una celda de un sistema de manufactura como el mostrado en la figura 40. En el cual se presenta todos los ejemplos anteriores, pero ya unificados en un solo sistema. Que puede representar la fabricación de una cualquier parte en un sistema de manufactura. Se comienza con la marca inicial P1.

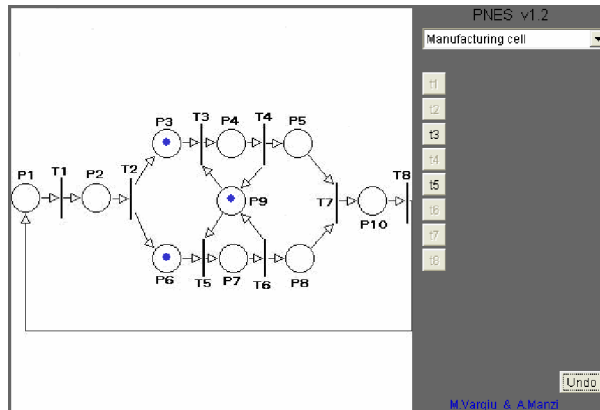


**Figura 40.** Celda de manufactura.

Al validarse la transición t1, ver figura 41, se muestra la secuencia para un grafo de RdP. Cuando se activa la transición t2 ocurre el fenómeno de paralelismo. Ver figura 42.

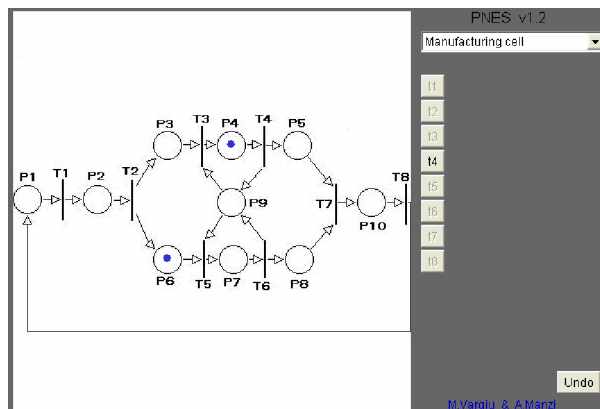


**Figura 41.** Transición t1, activada.

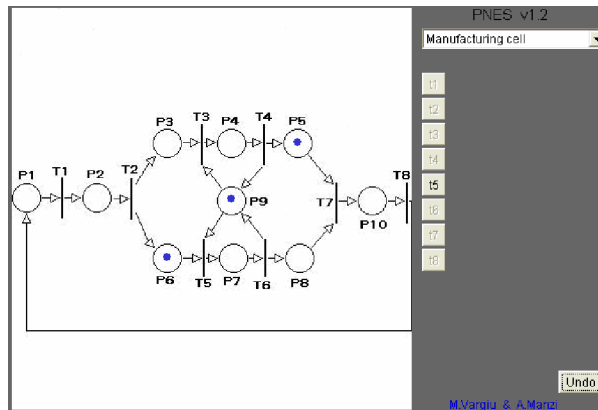


**Figura 42.** Transición t2, activada.

En la figura 43, se muestra el fenómeno de la exclusión mutua, al presentarse que dos actividades requieren de un mismo recurso. Representado como por el lugar P9, cumpliéndose la condición de activación para t3, quedando así el recurso P9, deshabilitado hasta que se termine la actividad en P4, con la activación de la transición t4. Como se muestra en la figura 44.

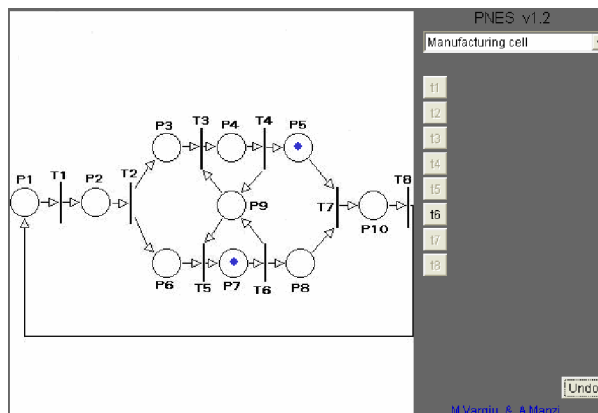


**Figura 43.** Transición t3, activada.

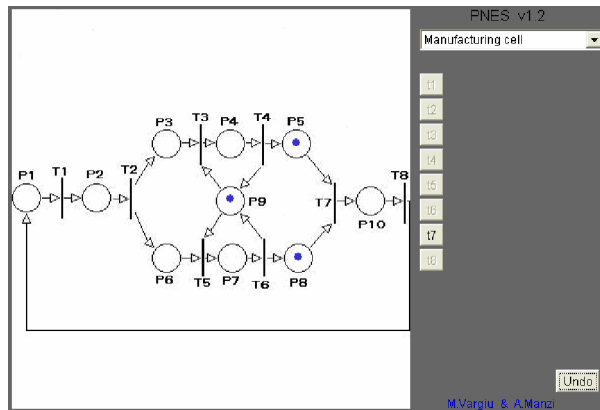


**Figura 44.** Transición *t4*, activada.

En las figuras 45 y 46, se ejecuta en mismo procedimiento explicado anteriormente; solo que en estas ocurren las transiciones *t5* y *t6*, que son las únicas posibles, ya que para que *t7* pueda ser activada debe cumplirse que los tokens indiquen actividad en los lugares *P5* y *P8*, y estas terminen. Nótese que al activarse *t7*, el recurso *P9* queda nuevamente libre. Como se muestra en la figura 46.

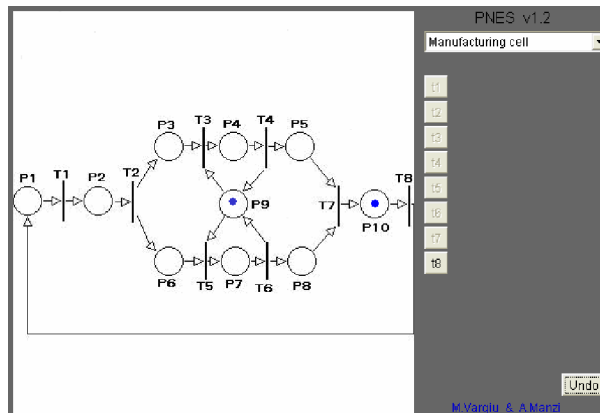


**Figura 45.** Transición *t5*, activada.



**Figura 46.** Transición t6, activada.

Por ultimo en la figura 47, se observa el final de la manufactura para determinada parte, con la activación de la transición t8, que a su vez es el comienzo de un nuevo proceso.



**Figura 47.** Transición t7, activada.

## CONCLUSIONES



En el desarrollo de proyectos de automatización industrial, el equipo de control más utilizado ha sido indudablemente el controlador lógico programable. Los avances en la industria del software han permitido que la aplicación de metodologías de diseño estructurado y los paradigmas de la programación orientada a objetos puedan ser llevadas a los lenguajes de programación de estos controladores. Estos lenguajes de programación tienen como fundamento las RdP, y otros métodos como los diagramas de estado-transición, cartas de estado, máquinas de estados finitos, siendo estos casos particulares de RdP.

La red de Petri es una herramienta para el modelado de sistemas de información que son considerados no-determinísticos, concurrentes, paralelos, asíncronos, distribuidos y/o estocásticos, se representa gráficamente por un grafo dirigido bipartito. Los dos tipos de nodos, lugares y transiciones representan, las variables que definen el estado del sistema (lugares) y a sus transformadores (transiciones). Los lugares se representan por círculos, las transiciones por barras, estos son conectados por arcos que van de lugares a transiciones y de transiciones a lugares. La presencia o ausencia de una señal dentro de un lugar puede indicar cuando una condición asociada con este lugar es verdadera o falsa. Las RdP son consideradas una herramienta para el estudio de los sistemas. Con su ayuda podemos modelar el comportamiento y la estructura de un sistema, y llevar el modelo a condiciones límite, que en un sistema real son difíciles de lograr o muy costosas.

Hoy en día existen abundantes teorías para el diseño de sistemas de control de bajo nivel discretos y continuos. Sin embargo, el estudio de la teoría de control y la implementación de los sistemas de eventos discretos (DES) ha emergido como un foco de investigación.

La teoría de RdP ha llegado a ser reconocida como una metodología establecida en la literatura de la robótica para modelar los sistemas de manufactura flexibles. Comparada con otros modelos de comportamiento dinámico gráficos, como los diagramas de las máquinas de estados finitos, las RdP ofrecen una forma de expresar procesos que requieren sincronía. Y quizás lo más importante es que las RdP pueden ser analizadas de manera formal y obtener información del comportamiento dinámico del sistema modelado.

Para modelar un sistema se usan representaciones matemáticas logrando una abstracción del sistema, esto es logrado con las RdP, que además pueden ser estudiadas como autómatas e investigar sus propiedades matemáticas. Podemos concluir diciendo que las Redes de Petri son una alternativa de modelado de sistemas, aplicados principalmente hacia el control de procesos, por su facilidad de manejo en el problema de la sincronización de procesos.

Es por esto que se ha realizado este recurso de información, con el ánimo de proponer esta valiosa y útil herramienta para ser tomada en cuenta y aplicada en los estudios de pregrado para los estudiantes de nuestra Universidad, dado que las aplicaciones de este modelo no está solo circunscrito a la ingeniería, sino a muchas otras áreas del saber. Como trabajo futuro se puede aplicar esta técnica de modelación de sistemas al PLC de la serie S7-300, cuya traducción a su lenguaje de programación, se encuentra ejemplificado en la tabla 6., que se encuentra en los laboratorios de la Universidad Tecnológica de Bolívar.

## BIBLIOGRAFÍA



- GARCÍA MORENO, Emilio. Automatización De Procesos Industriales. México D. F. Alfaomega. 2001. Pág. 204.
- ZHOU, MengChu. DICESARE, Frank. Petri Net Synthesis for Discrete Event Control of Manufacturing Systems. Kluwer Academic Publisher. Londres 1993.
- JOHNSONBAUGH, Richard. Matemáticas discretas. México D. F. Grupo Editorial Iberoamérica. 1988
- SABIGUERO, Ariel. Nomenclatura y definiciones básicas de Redes de Petri. Instituto de Computación. Facultad de Ingeniería.
- AJMONE Marsan, G. Balbo, G. Conte – Performance Models of Multiprocessor Systems. 1986.
- CERVANTES, Jair. Representación y aprendizaje de conocimiento con Redes de Petri difusas. México 2005.
- MENON, S. R., QUINN, P. M., KAPOOR, S. G. Coordination control of flexible manufacturing systems using colored Petri Nets.1988. Pág. 317-326.

- LINDEMANN, Christoph. Performance Modelling with Deterministic and Stochastic Petri Nets” 1998.
- ZAPATA German, CARRASCO Eliana. Estructuras Generalizadas para Controladores Lógicos modelados mediante Redes de Petri. UNAL Medellín 2002.
- LLORENS Marisa, OLIVER Javier. Redes Reconfigurables Controladas por Mercado: Redes de Petri con cambios dinámicos estructurales. Valencia, España 2004.
- Kurt, J. Coloured Petri Nets. Springer Verlag, 1997.





ANEXOS

## HERRAMIENTAS DE INVESTIGACIÓN

### PAGINAS WEB DE INTERÉS:

- Comunidad internacional de Redes de Petri.  
<http://www.informatik.uni-hamburg.de/TGI/PetriNets/>
- Instituto Internacional de Ingenieros Eléctricos Y Electrónicos.  
<http://www.ieee.org>
- Simuladores virtuales de RdP.  
[http://www.cs.ucr.edu/~vahid/courses/269\\_f98/Petri/Petri.html](http://www.cs.ucr.edu/~vahid/courses/269_f98/Petri/Petri.html)  
<http://www.filebot.org/Education/Science/item8333-3.htm>  
<http://web.tiscali.it/marconthenet/Pnes.Pnes.html>
- Asociación Europea del software la ciencia y la tecnología.  
<http://www.easst.org/>
- Definiciones formales y ejemplos.  
<http://www.fismat.umich.mx/~crivera/tesis/node25.html>
- Taller internacional sobre aplicaciones practicas del modelamiento estocástico.  
<http://www.doc.ic.ac.uk/pasm2004/>
- Software y sistemas de modelamientos.  
<http://www.sosym.org/>