



UNIVERSIDAD DE LOS ANDES  
DOCTORADO EN CIENCIAS APLICADAS  
FACULTAD DE INGENIERIA

Desarrollo de un Procedimiento para Diseño de Sistemas de Control en Procesos con un  
Modelo de Integración Basado en Holones.

Trabajo presentado como requisito final para optar al título de Doctor en Ciencias Aplicadas

Autor:  
Miguel Indriago

Tutor:  
Dr. Edgar Chacón

Mérida, Abril 2011

**acta**

**acta claude**

## Resumen

La Arquitectura de Integración para la Automatización de Complejos de Producción es una propuesta de arquitectura de referencia basada en sistemas de manufactura holónicos que divide los procesos en unidades de producción. Cada unidad de producción representa una parte del proceso (y su sistema de control) con autonomía y capacidad para negociar con las otras unidades de producción. Este trabajo presenta un método para modelar e implementar supervisores de eventos discretos en la unidad de producción. El supervisor se modela con una red de Petri a la que se le agrega una interfaz para poder leer los eventos y ejecutar las acciones del proceso que se quiere controlar. El método permite simular, probar y modificar el supervisor fácilmente esto hace que la unidad de producción sea flexible a los cambios en el producto o en el proceso.

## **Dedicatoria**

A Dios

A mis Padres, José y Mercedes

A mi esposa, Haydee

A mis hijos, Miguel, Luis Eduardo y Alejandra

## **Agradecimiento**

A Dios Todopoderoso.

Al Dr. Edgar Chacón, mi tutor quien me guió a lo largo de estos últimos años, sus consejos y sus recomendaciones siempre fueron de mucha ayuda. Además de toda la confianza que ha depositado en mí.

A los doctores Mariela Cerrada, Félix García, Miguel Ríos y Claude Martínez que con sus sugerencias enriquecieron a este trabajo. Muchas gracias por el tiempo que dedicaron para revisar este trabajo.

A Milagros por toda su ayuda, la cual facilitó enormemente el desarrollo de mi doctorado.

A mis colegas Oscar Lamedá, Luis Roberti y Willians Osal, quienes desde la dirección de Investigación de Postgrado de la Universidad Nacional Experimental “Antonio José de Sucre” vicerrectorado Barquisimeto, dieron su apoyo, el cual fue de muy útil para la culminación de mis estudios.

# Contenido

Resumen .....	2
Dedicatoria .....	5
Agradecimiento .....	6
Contenido .....	7
Indice de figuras .....	10
Indice de tablas .....	13
Indice de tablas .....	13
Capítulo 1. Introducción .....	14
Capítulo 2. Preliminares .....	21
2.1 Sistemas dinámicos .....	21
2.2 Modelo del sistema de eventos discreto .....	25
2.2.1 Grafos .....	25
2.2.2 Redes de Petri .....	26
2.3 Bases para el modelado de sistemas híbridos .....	29
2.4 Supervisores .....	31
2.5 Arquitecturas .....	32
2.6 Agentes .....	33

2.7	Sistemas de Manufactura Holónicos .....	35
2.8	Arquitectura de Integración para la Automatización de Complejos de Producción.	36
2.8.1	Unidad de producción.....	39
2.9	Conclusión al capítulo .....	42
Capítulo 3. Sistema Supervisor .....		44
3.1	El supervisor .....	44
3.1.1	Modelo del supervisor .....	44
3.2	Marco conceptual para la implementación del supervisor. ....	46
3.2.1	Actuadores .....	49
3.3	Síntesis de Supervisores .....	56
3.3.1	Síntesis de supervisores usando invarianza de lugares.....	58
3.3.2	Supervisores de condición - acción .....	61
3.4	Sistema de Coordinación.....	62
3.5	Conclusiones del capítulo.....	64
Capítulo 4. Implementación. ....		65
4.1	Resumen del método para la implementación del supervisor .....	65
4.2	Implementación en un computador personal.....	66
4.2.1	Simulación.....	72
4.3	Implementación en un controlador lógico programable.....	75



4.3.1	Implementación del modelo de la planta.....	75
4.3.2	Implementación de la función de control .....	77
4.4	Simulación de la dinámicas continuas en un PLC.....	78
4.5	Resumen del capítulo .....	79
Capítulo 5. Caso de estudio 1 .....		80
5.1	Modelo de la planta. ....	81
5.2	Diseño del supervisor. ....	81
5.3	Dinámicas continuas.....	84
5.4	Definición del modelo .....	85
5.5	Caso de estudio 2.....	87
5.5.1	Modelo de la planta .....	88
5.5.2	Supervisor.....	88
5.5.3	Dinámicas continuas.....	90
5.6	Caso de estudio 3.....	91
5.6.1	Arquitectura de Sistema .....	91
5.6.2	Filosofía de operación .....	93
5.6.3	Función de control.....	100
Conclusiones.....		104
Referencias bibliográficas .....		106

## Índice de figuras

Figura 2.1. Sistema .....	21
Figura 2.2. Sistema dinámico de variable continua.....	23
Figura 2.3. Sistema dinámico de eventos discretos.....	24
Figura 2.4. Sistema dinámico híbrido.....	25
Figura 2.5. Grafos.....	26
Figura 2.6. Red de Petri.....	26
Figura 2.7. Modelo de una planta híbrida.....	30
Figura 2.8. Agente reactivo basado en modelo. ....	34
Figura 2.9. Modelo de una arquitectura de empresas.....	37
Figura 2.10. Enfoque holónico.....	38
Figura 2.11. Enfoque clásico o jerárquico.....	38
Figura 2.12. Enfoque heterárquico.....	38
Figura 2.13. Entradas y salidas de una unidad de producción.....	39
Figura 2.14. Ciclo de toma de decisiones.....	40
Figura 2.15. Capas del sistema de decisión e información en conexión con el proceso.....	40
Figura 2.16. Esquema de negociación entre unidades de producción.....	41
Figura 3.1. Modelo del supervisor.....	46

Figura 3.2. Ejemplo de un tanque con dos válvulas. ....	48
Figura 3.3. Cuatro niveles inferiores del modelo físico en la norma s88.1. ....	48
Figura 3.5. Flujos de las variables del actuador. ....	53
Figura 3.6. Modelo del comportamiento de un actuador tipo 1. ....	55
Figura 3.7. Modelo del comportamiento de un actuador tipo 2. ....	55
Figura 3.8. Modelo del comportamiento de un actuador tipo 3. ....	55
Figura 3.9. Modelo del comportamiento de un actuador tipo 4. ....	56
Figura 3.10. Transformación de una transición en dos transiciones y un lugar. ....	60
Figura 3.11. Agente reactivo basado en modelo. ....	63
Figura 4.1. Diagrama de clases.....	68
Figura 4.2. Ejemplo de un tanque y una válvula. ....	69
Figura 4.3. Modelo de ejemplo de un tanque y una válvula.....	70
Figura 4.4. Diagrama de secuencias. ....	72
Figura 4.5. Sistema Planta - Supervisor. ....	73
Figura 4.6. Diagrama escalera de una transición.....	76
Figura 4.7. Diagrama escalera de modelos en redes de Petri segura y ordinaria. ....	77
Figura 4.8. Forma de las reglas de la función de control en lenguaje escalera ....	77
Figura 5.1. Sistema híbrido de dos tanques.....	80
Figura 5.2. Modelo detallado del sistema dos tanques, lazo abierto. ....	81

Figura 5.3. Modelo de modos del sistema de dos tanques, lazo abierto.....	82
Figura 5.4. Modelo del sistema de dos tanques.....	82
Figura 5.5. Modelo a lazo cerrado de sistema de dos tanques.....	83
Figura 5.6.- Función de control en diagrama escalera para el sistema de dos tanques .....	84
Figura 5.7. Graficas de los niveles del sistema de dos tanques.....	87
Figura 5.8. Sistema de tres tanques. ....	88
Figura 5.9. Red de Petri del sistema de 3 tanques. ....	89
Figura 5.10. Red de Petri del sistema de tres tanques con supervisor.....	90
Figura 5.11. Gráficas de los niveles del sistema de tres tanques.....	91
Figura 5.12. Proceso de transporte de harina. ....	92
Figura 5.13. Arquitectura del sistema de transporte de harina. ....	93
Figura 5.14. Función de control para el sistema de transporte de harina .....	102
Figura 5.15. Resultados del sistema de control para el transporte de harina.....	103

## Indice de tablas

Tabla 3.1. Variables asociadas a los actuadores.....	50
Tabla 4.1. Acciones y comando para cada actuador en modo automático.....	71
Tabla 4.2. Valores de las entradas en función de la salida de los actuadores.....	73
Tabla 4.3. Relación entre la red de Petri y la memoria del PLC .....	75
Tabla 5.1. Conjuntos del sistema.....	86
Tabla 5.2. Funciones del sistema.....	86

## Capítulo 1. Introducción

En las últimas décadas se ha buscado mejorar los procesos industriales de tal forma que los productos se elaboren en menor tiempo, a un menor costo y con mejor calidad. Una vía para lograr esto es tener procesos flexibles. Los procesos flexibles tiene la capacidad de reaccionar a cambios, planificados o inesperados, de tal forma que pueden ajustarse en corto tiempo y con poco esfuerzo, estos cambios pueden estar orientados hacia la elaboración de un producto diferente o por la necesidad de usar diferentes equipos para hacer la misma operación. La flexibilidad de los procesos se logra con tecnologías y estructuras organizacionales que permitan adaptarse a los cambios que exige el mercado. En este sentido, existen distintos enfoques como los jerárquicos (Williams, 1993; Li, 1994; ISA, 1995), los heterárquicos (Silveira y Combacau, 2003; Saad y otros, 1995) y los holónicos (Wyns, 1999).

El enfoque jerárquico divide los procesos en niveles apilados, la información del estado viaja del nivel inferior, donde se encuentra la planta, hacia los niveles superiores donde se toman las decisiones, las acciones van en sentido inverso, desde los niveles superiores hacia los niveles inferiores. El enfoque heterárquico divide el proceso en subprocesos, cada uno de los cuales mantiene independencia y emplean algún mecanismo de coordinación que les permite mantenerse dentro de las metas del conjunto. En el enfoque holónico, cada subprocesos es independiente y cooperante, al mismo tiempo, mantiene características jerárquicas en su interior. Dentro de los enfoques holónicos se encuentra la Arquitectura de Integración para la Automatización de Complejos de Producción (AIACP), en ella se basará nuestra propuesta (Chacón y otros, 2002).

Los sistemas de manufactura holónicos se originan de una teoría acerca de sistemas complejos adaptativos. La palabra holón fue introducida en los años sesenta por Arthur Koestler (Koestler, 1967), y sugiere un comportamiento doble, el holón se puede comportar como un todo, pero a su vez, también se puede comportar como una parte (efecto Janus). Los sistemas de manufactura holónicos están formados por comunidades de holones, cada

holón puede formar parte de otro holón que lo contiene y a su vez estar formado por otros holones. Los sistemas de manufactura holónicos aprovechan las ventajas de los sistemas jerárquicos y de los sistemas heterárquicos; en este sentido, un holón es un ente autónomo capaz de tomar sus propias decisiones y, a su vez, tiene la capacidad de negociar con otros holones con la finalidad de alcanzar las metas en común. La comunidad de holones relacionados entre sí es lo que se conoce como holarquía (Wyns, 1999). En esta tesis se explica y se verifica, como a través de casos industriales organizados según la AIACP, se lleva a cabo el manejo eficaz de un proceso industrial con respecto a los objetivos del mencionado proceso.

La AIACP divide cada proceso en subprocesos denominados unidades de producción (UP). Cada UP está formada por una parte física, que le da valor agregado al producto, y un sistema de control, que tiene entre sus funciones: mantener las variables continuas y el estado discreto en los valores que permitan alcanzar las metas fijadas, manejar las contingencias, registrar información para el mantenimiento, medir variables que tengan impacto económico y negociar con otras unidades de producción. En el proceso de negociación las UP revisan las posibilidades de alcanzar las metas propias y las metas comunes del conjunto según su estado actual, en caso de encontrar deficiencias se comienza el proceso de negociación para establecer un acuerdo con las otras UP, dentro de un espacio de acuerdos previstos que permitan re-planificar en función de cumplir los objetivos comunes. Las unidades de producción son autónomas.

Para que un proceso sea flexible, los cambios en el sistema de control deben ser poco complicados y rápidos; además la implementación debe ser sencilla y confiable. El diseño y puesta en funcionamiento del sistema de control involucra cuatro etapas: modelado de la planta, diseño del control, simulación e implementación. En el capítulo 2 se describe como realizar el modelo de la planta, el supervisor se describe en el capítulo 3, mientras que en el capítulo 4 se dan detalles de la simulación y puesta en funcionamiento o implementación.

El modelo de la planta depende de su dinámica, la cual puede ser continua, de eventos discretos o híbrida. Si la dinámica de la planta es de naturaleza continua se usarán ecuaciones diferenciales para la representación del mismo (Lennartson y otros, 1994). Si la planta tiene dinámicas discretas controladas por eventos entonces los modelos se construirán usando con redes de Petri (Polic y Jezernik, 2005; Moody y Antsaklis, 1998; Murata, 1989; Peterson, 1981). Los sistemas que presentan simultáneamente naturaleza continua y discretas controladas por eventos, son llamados sistemas híbridos, se usará una combinación de ecuaciones diferenciales y redes de Petri para la elaboración de los modelos de plantas con dinámicas híbridas (Moody y Antsaklis, 1998; Frey, 1998). Resulta útil incluir en el sistema de control un supervisor el cual se modela como un sistema de eventos discretos y tiene como objetivo mantener el estado de la planta dentro de una región deseada, esto se puede lograr ejecutando acciones de control según el estado actual del proceso. Algunos autores (Zapata y Carrasco, 2002; Frey, 1998) utilizan redes de Petri interpretadas para la construcción de este tipo de supervisores. Existen diversas técnicas que ayudan al diseño de supervisores, muchas de ellas están basadas en redes de Petri. La implementación del supervisor se debe hacer en un equipo de control y es común utilizar controladores lógicos programables (PLC) o computadores personales (PC) para este fin.

En este trabajo se divide el sistema de control de la UP en tres capas: una capa inferior con la función de control regulatorio, encargada de mantener las variables continuas en el valor deseado, una capa intermedia, donde se encuentra el supervisor que determina y controla el estado discreto y una capa superior denominada coordinador, que maneja las contingencias y se encarga de las negociaciones.

Este trabajo presenta un método para diseñar el supervisor de eventos discretos de la capa intermedia de la UP en la AIACP, el cual, se compone del modelo de la planta, basado en redes de Petri, una función de control, que ejecuta las acciones dependiendo del estado del modelo y una interfaz con el proceso, que convierte las entradas en eventos y las acciones en salidas. Además, en el diseño, se toma consideración la capacidad que debe tener la capa



intermedia de supervisión, para intercambiar información con las otras dos capas. También se muestra la forma de implementar el supervisor, en el equipo de control.

Entre los trabajos que contribuyeron a esta investigación se pueden citar los siguientes:

Chacón y Sarrazín (2002), definen los sistemas dinámicos de variable continua, de variable discreta controlada por eventos y de variables híbridas, así como también dan las definiciones de estado, espacio de estado y evento.

Chacón y otros (2002), hacen la propuesta de la Arquitectura Integración para la Automatización de Complejos de Producción, donde explican las funciones de los diferentes niveles que la forman. En este trabajo se definen tres ejes de las organizaciones, estos son: producción, soporte y decisión, además, se establecen las relaciones entre ellos. También se menciona el ciclo de toma de decisiones, se describe detalladamente y se presenta un mapa conceptual de los componentes de la UP.

Wyn (1999), describe una arquitectura de referencia, en este caso PROSA. En su trabajo se desarrolla esta arquitectura de referencia, la cual permite flexibilidad y reconfiguración. PROSA está inspirada en el paradigma de los sistemas de manufactura holónicos, y consiste de cuatro tipos de holones; el holón orden, responsable de los aspectos logísticos, el holón producto, el cual tiene asignado la responsabilidad del proceso y la elaboración del producto y el holón recurso, responsable del manejo de los recursos. El cuarto tipo de holón, llamado staff, es opcional, y avisa al resto de los holones acerca de ciertos aspectos del sistema de producción. PROSA es una generalización de los enfoques heterárquico y jerárquicos existentes.

Indriago y Lucena (2004), presentan un trabajo en el que se explican los detalles la función estándar de los actuadores, la cual sirve de soporte al supervisor, y la manera de implementarla en un PLC. Se define la función estándar como aquella encargada de ejecutar las acciones de los actuadores que son independiente del proceso al que pertenecen, entre estas acciones se encuentran: el cambio de modo manual o automático, paradas de

emergencia, alarmas por falta de la respuesta del proceso y parada por alarmas entre otras. En el trabajo se usa una subdivisión del proceso según el modelo físico de la norma ISA s88.

Indriago y Chacón (2010), dividen la unidad de producción de la AIAPC en tres capas, control regulatorio, supervisor y coordinador. Además, se presenta un modelo para los procesos híbridos y para la capa de supervisión, finalmente se implementa el supervisor en un computador personal (PC) y en un controlador lógico programable (PLC).

Moddy y Antsaklis (1998), presentan un método para la síntesis de supervisores basado en la invarianza de lugares en una red de Petri, la cual se diseña a partir del modelo de la planta y un conjunto de restricciones. En este trabajo se destaca la definición de la red de Petri usando las matrices de lugares de entrada y de salida, además de presentar otras definiciones importantes tales como: transiciones controlables y observables, supervisor y supervisor máximo permisivo.

Murata (1989), presenta un trabajo de especial importancia para las redes de Petri, es una recopilación de todos los aspectos teóricos para estas redes, entre los cuales se destaca: la definición de la red y sus propiedades. Además se muestran diversos ejemplos de modelos de sistemas, y se muestra la solución de algunos problemas típicos usando redes de Petri, que pueden ser de especial utilidad en el diseño de supervisores. También encontramos algunas definiciones importantes tales como: controlabilidad, trampas y sifones, entre otras.

A continuación se mencionan otros trabajos que se revisaron para el desarrollo de esta tesis:

Hellgren (2002), muestra una metodología para el desarrollo e implementación de supervisores como parte de CHAMP (Chalmers Architecture and Methodology for Flexible Production), y que se usa para el desarrollo del sistema de control en sistemas de manufactura flexibles. La metodología consta de cuatro fases que son: modelado de la planta, especificaciones de la conducta, síntesis de controladores e implementación. El enfoque para el modelado, llamado parametrización, usa un algoritmo de composición

sincrónica priorizada (PSC) de redes de Petri. El trabajo muestra un algoritmo para la síntesis de supervisores y finalmente se implementa sobre controladores lógicos programables usando cartas de función secuencial (SFC).

Zeigler y otros (1995), presentan un formalismo matemático para sistemas dinámicos de eventos discretos (SDED), el cual ofrece un marco para modelos, diseño, análisis y simulación de sistemas híbridos, que igualmente es usado para la síntesis de sistemas supervisorios.

Brinksma y otros (2002), reportan el uso de técnicas para examinar modelos con la finalidad de la verificación del programa para el control de un proceso híbridos y la derivación de una planificación para control óptimo. El caso de estudio fue una planta química y el controlador fue un PLC.

Polic y Jezernik (2005), dirigen su trabajo hacia el modelado, descripción dinámica, y control de sistemas de eventos discretos (SED) para sistemas de producción, y diseño lógico de control de máquinas. La lógica de control de tales sistemas son considerados como un SED, el cual es modelado por una representación matricial de una red de Petri, descrita por una ecuación algebraica del próximo estado y una ecuación que selecciona el evento que produce el disparo. El control de los SED es introducido por señales lógicas, las cuales representan condiciones para la ocurrencia de eventos. Tal representación facilita el modelo de lazo cerrado del SED, de utilidad para la simulación y análisis del sistema antes de la puesta en marcha.

El presente trabajo se organiza de la siguiente manera: El capítulo 1 es esta introducción. El capítulo 2 menciona los aspectos preliminares que van a ser de utilidad en el desarrollo del trabajo, tales como las definiciones de los sistemas dinámicos, sistemas continuos, sistemas de eventos discreto, sistemas híbridos, grafos, redes de Petri, arquitectura y agentes, además se dan detalles de la AIACP, la cual está basada en sistemas de manufactura holónicos, se define la unidad de producción y sus analogías con el holón; también se explica el modelo de tres capas del sistema de control de la UP. En capítulo 3 se define el supervisor como un

sistema de eventos discretos y a la función estándar, la cual sirve de apoyo al supervisor. El capítulo 4 muestra en que forma se puede implementar el supervisor en un computador y en un controlador lógico programable; además se resume el método con el cual se construye el supervisor. En el capítulo 5 se presentan tres casos de estudio, el primer caso es netamente académico y se trata del llenado de un par de tanques, el segundo caso es un sistema de tres tanques, el tercer caso es un sistema de mayor complejidad que se logró implementar en fábrica. En el capítulo 6 se dan las conclusiones.

## Capítulo 2. Preliminares

En este capítulo se presentan aspectos preliminares, los cuales fueron de utilidad en el método propuesto para implementar un sistema supervisor, que sirva para el control de un proceso industrial. Algunos de estos aspectos, son definiciones básicas de sistemas dinámicos y redes de Petri, también se mencionan los tipos de arquitecturas, los agentes y se da la base para el modelo de la planta.

### 2.1 Sistemas dinámicos

*Definición 2.1:* Un sistema  $S$  es una estructura en la que actúa una entrada ( $u$ ) y que responde con una salida ( $y$ ), en un intervalo de tiempo dado. Los sistemas pueden ser discretos, continuos o híbridos.

Cuando se conoce la estructura, los componentes y las leyes matemáticas internas es posible formular un modelo matemático intrínseco, en términos de las variables físicas o lógicas del sistema. A estas variables se le denominan estado.

*Definición 2.2:* Un estado  $x$  de un sistema  $S$  es una variable interna de  $S$  en el momento presente y que determina la salida. La figura 2.1 muestra un sistema y sus relaciones.

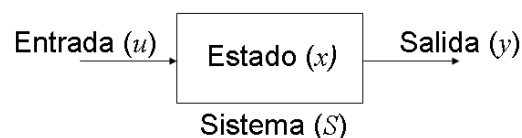


Figura 2.1. Sistema

*Definición 2.3:* Un Sistema Dinámico cambia su estado en función del tiempo y se define según la siguiente tupla (Zeigler et al, 1995)

$$SD = (T, U, Y, \Omega, X, \Delta, A) \quad (1)$$

donde:

$T$  es la base de tiempo,

$U$  es el conjunto de valores de entrada,

$Y$  es el conjunto de los valores de salida

$\Omega$  es el conjunto de segmento de entradas admisibles  $\omega: (t_1, t_2) \rightarrow X$  siendo  $(t_1, t_2)$  un intervalo en  $T$ ,

$X$  es el conjunto de estados,

$\Delta: X \times \Omega \rightarrow X$  es la función de transición de estado global,

$A: X \times U \rightarrow Y$  es la función de salida.

*Definición 2.4:* Un Sistema Dinámico de Variables Continuas (SDVC) es un sistema donde las variables de estado evolucionan continuamente en el tiempo (Chacón y Sarrazin, 2004).

La figura 2.2 muestra un sistema dinámico de variable continua que, por lo general, es modelado por un sistema de ecuaciones diferenciales de la forma

$$\dot{x}(t) = f(x(t), u(t), t) \quad (2)$$

donde:

$f$  es la función de transición de estado,

$x(t)$  el vector de variables de estado,

$u(t)$  es el vector de entradas,

$t$  es el tiempo.

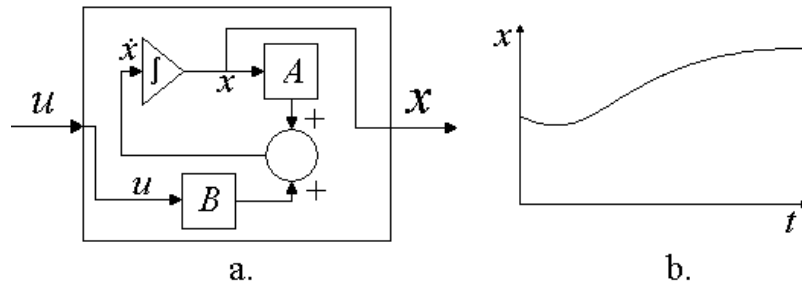


Figura 2.2. a. Sistema dinámico de variable continua. b. Variación del estado en función del tiempo.

*Definición 2.5:* Un evento es una ocurrencia instantánea que causa la transición de valor del estado del sistema (Chacón y Sarrazin, 2004).

Un evento se considera controlable si el mismo puede ser deshabilitado por un supervisor, con el fin de que no aparezca en el sistema, mientras que los eventos no controlables (incontrolables), aparecen espontáneamente en el sistema.

*Definición 2.6:* Un Sistema Dinámico de Eventos Discretos (SDED), es un sistema con un espacio de estado discreto, las transiciones de estados están determinadas por la ocurrencia de eventos, los cuales ocurren en instantes de tiempo que son impredecibles (Chacón y Sarrazin, 2004).

La figura 2.3 muestra un Sistema Dinámico de Eventos Discretos que, por lo general, es modelado con una ecuación de la forma

$$q_{k+1} = F(q_k, \sigma_k) \quad (3)$$

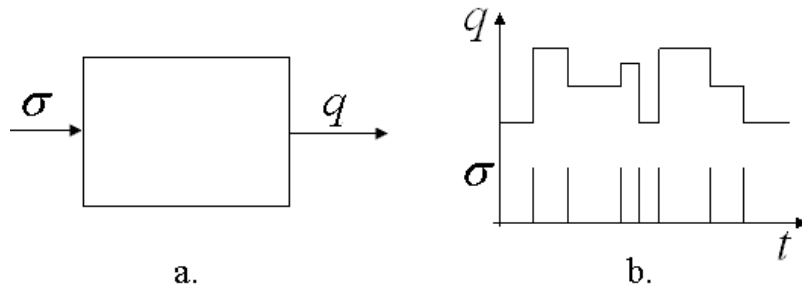
donde:

$F$  es la función de transición de estado,

$q_{k+1}$  es el próximo estado,

$q_k$  es el estado actual,

$\sigma_k$  es la secuencia de eventos de entradas hasta el k-ésimo evento ocurrido.



**Figura 2.3. a. Sistema dinámico de eventos discretos. b. Variación del estado en respuesta a los eventos.**

*Definición 2.7:* Un Sistema Dinámico Híbrido (HDS) es un sistema cuyo comportamiento es descrito por un sistema continuo y otro de eventos discretos, además de la interacción entre ambos (Chacón y Sarrazin, 2004). En otras palabras, a cada estado discreto del sistema, el cual ocurre en un intervalo de tiempo delimitado por un evento inicial y un evento final, le corresponde un conjunto de ecuaciones diferenciales que definen el estado continuo en dicho intervalo de tiempo.

Para modelar los sistemas híbridos, se usa una combinación de las ecuaciones (2) y (3) tal como se muestra en la siguiente ecuación.

$$\dot{x}(t) = H(q_k, \sigma_k, x(t), u(t), t) \quad (4)$$

donde:

$q_k$  es el estado discreto actual,

$\sigma_k$  es la secuencia de eventos de entradas hasta el  $k$ -ésimo evento ocurrido.

$x(t)$  el vector de variables de estado continuas,

$u(t)$  es el vector de entradas,

$t$  es el tiempo.

En la figura 2.4 se muestra un esquema del sistema híbrido.



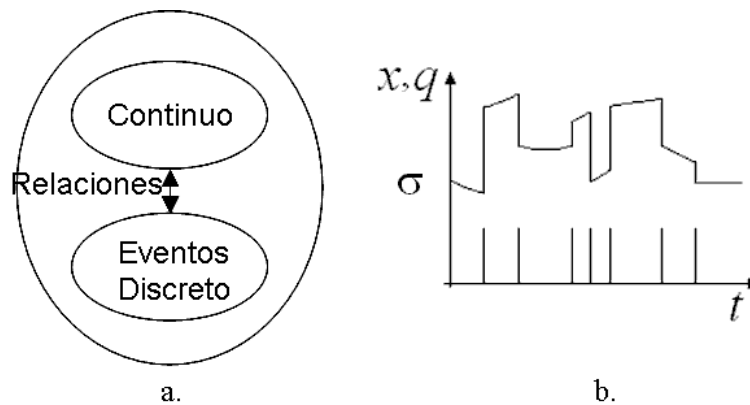


Figura 2.4. a. Sistema dinámico híbrido. b. Variación del estado en función del tiempo.

## 2.2 Modelo del sistema de eventos discreto

Para representar gráficamente un SDED, se utilizan distintas técnicas como los autómatas, redes de Petri, entre otros, de manera gráfica, estas representaciones tienen en común utilización de grafos.

### 2.2.1 Grafos

*Definición 2.8:* Un grafo es un par  $G = (V, A)$ , donde  $V$  es un conjunto de nodos, llamados vértices, y  $A$  es un conjunto de pares de vértices, llamados aristas. La arista  $a, b$  se denota  $ab$ , donde  $a$  y  $b$  son dos vértices pertenecientes a  $V$ , tal como se puede observar en la figura 2.5.a.

*Definición 2.9:* A un grafo dirigido se puede definir como un grafo que contiene aristas dirigidas. Una arista dirigida tiene un vértice origen y un vértice destino. La arista lleva una punta de flecha en el vértice destino, tal como se puede observar en la figura 2.5.b.

*Definición 2.10:* Un grafo  $G = (V, A)$  se llama bipartito si  $V$  está partido en dos subconjuntos  $P$  y  $T$  tal que para toda arista  $ab$  se cumpla que  $(a \in P \wedge b \in T) \vee (a \in T \wedge b \in P)$ , como se muestra en la figura 2.5.c.

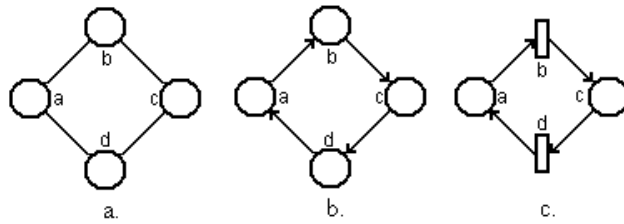


Figura 2.5. (a) grafo, (b) grafo dirigido y (c) grafo bipartito dirigido.

### 2.2.2 Redes de Petri

Una red de Petri es un grafo dirigido bipartito que tiene como vértices lugares  $p$  y transiciones  $t$ . Los lugares son dibujados como círculos y las transiciones como barras o rectángulos. Las aristas del grafo son conocidas como arcos. Estos tienen un peso específico entero positivo. Los arcos van desde un lugar  $p$  hacia una transición  $t$  o desde una transición  $t$  a un lugar  $p$ , nunca entre dos transiciones o entre dos lugares. De esta forma podemos tener lugares de entrada  $p_i$  a una transición  $t$  y lugares de salida  $p_o$  de una transición  $t$ . Cada arco tiene un peso  $k$ . Por simplicidad, el peso de los arcos no se indica cuando éste es igual a 1. Un arco que esté etiquetado con  $k$ , donde  $k \in \mathbb{Z}^+$ , puede ser interpretado como  $k$  arcos paralelos.

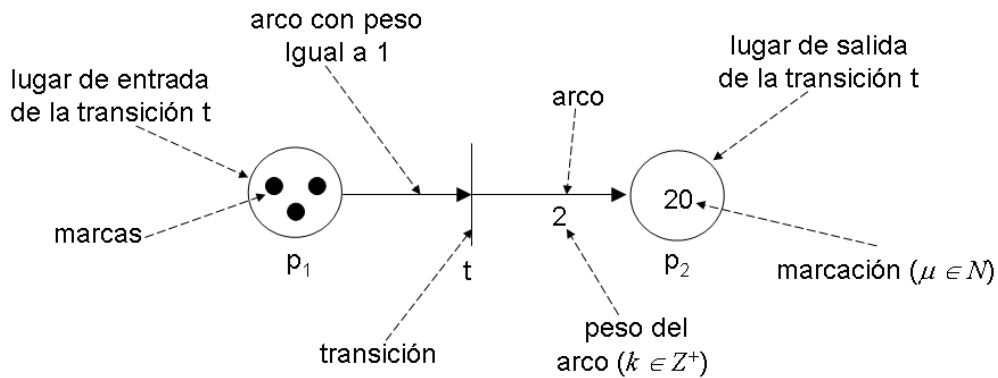


Figura 2.6. Red de Petri.

Se le asignan un número entero no negativo de marcas o fichas a los lugares. Esta asignación es conocida como marcación  $\mu$ , la cual es representada gráficamente mediante unos pequeños círculos negros, llamados marcas o fichas, dentro de un lugar  $p$ . Si el número de marcas es demasiado grande, las  $k$  marcas son representadas con un número no

negativo dentro del correspondiente lugar. Si asociamos los lugares de entrada  $p_i$  de una transición a pre condiciones, los lugares de salida  $p_o$  a pos condiciones, las transiciones  $t$  a eventos y las marcas a estado o recursos, entonces las redes de Petri son útiles para modelar sistemas dinámicos de eventos discretos. Las marcas pueden tener muchas interpretaciones, por ejemplo, cuando un lugar contiene una marca, éste puede representar que la correspondiente condición es verdadera, en otros casos,  $k$  marcas pueden representar  $k$  recursos. Debido a que las redes de Petri pueden modelar una gran variedad de sistemas dinámicos de eventos discretos, lo que los lugares, transiciones y marcas representen puede variar de una aplicación a otra. La figura 2.6 muestra una red de Petri con todos sus elementos.

Muchos autores han dado una definición para las redes de Petri (Murata, 1989; Peterson, 1981; Moody y Antsaklis, 1998), todas ellas equivalentes entre sí, la diferencia está en la forma de representar los arcos que relacionan los lugares con las transiciones y el peso de los mismos. Se usará la definición que da Moody y Antsaklis (1998), debido a que esta definición incluye las matrices de lugares de entrada y salida que serán de utilidad en el cálculo de supervisores.

*Definición 2.11:* Una Red de Petri (*RdP*) es una quintupla (Moody y Antsaklis, 1998),

$$RdP = (P, T, D^+, D^-, \mu_0) \quad (5)$$

donde:

$P = \{p_1, p_2, \dots, p_n\}$  es un conjunto finito de lugares,

$T = \{t_1, t_2, \dots, t_m\}$  es un conjunto finito de transiciones,

$D^+ \in \mathbb{Z}^{n \times m}$  es la matriz de los lugares de salida,

$D^- \in \mathbb{Z}^{n \times m}$  es la matriz de los lugares de entrada,

$\mu_0: P \rightarrow \{\mu_1, \mu_2, \mu_3, \dots, \mu_n\}$  es el vector marcación inicial con dimensión  $n \times 1$ , donde  $\mu_i$  es la marcación inicial de cada lugar  $p_i$ ,

$$P \cap T = \emptyset \text{ y } P \cup T \neq \emptyset.$$

Cada elemento  $d_{ij}^+$  de la matriz  $D^+$ , es el peso del arco del lugar de salida  $i$  desde la transición  $j$ , y cada elemento  $d_{ij}^-$  de la matriz  $D^-$ , es el peso del arco desde lugar de entrada  $i$  hacia la transición  $j$ . La matriz de incidencia  $D$  está formada por la diferencia de  $D^+$  y  $D^-$ , esto es

$$D = D^+ - D^- \quad (6)$$

### 2.2.2.1 Reglas de transición (disparos)

La marcación  $\mu$  en una  $RdP$  cambia de acuerdo a las siguientes reglas (de disparo) (Murata, 1989):

1. Una transición  $t_j$  está habilitada si cada lugar de entrada  $p_i$  de  $t_j$  está marcado con al menos  $d_{ij}^-$  marcas esto es equivalente a verificar la condición de la ecuación (7)

$$\mu \geq D^- q \quad (7)$$

donde  $q$  es un vector de enteros de dimensión  $m$  llamado vector de disparo; este vector contiene un 1 en el  $j$ -ésimo elemento, que corresponde a la transición que se intenta disparar, el resto de los elementos son ceros. Si la condición es válida, la transición  $t_j$  está habilitada y puede ser disparada.

2. Una transición habilitada puede o no puede dispararse, dependiendo si el evento asociado a la transición toma lugar o no.
3. Un disparo de una transición habilitada  $t_j$  remueve  $d_{ij}^-$  marcas de cada lugar de entrada  $p_i$  de  $t_j$  y agrega  $d_{kj}^+$  marcas a cada lugar de salida  $p_k$  de  $t_j$ .

Si ocurre un disparo en una transición habilitada la nueva marcación puede calcularse con la ecuación (8), llamada ecuación de estado.

$$\mu_k = \mu_{k-1} + Dq_k \quad (8)$$

### **2.2.2.2 Transiciones observables y controlables**

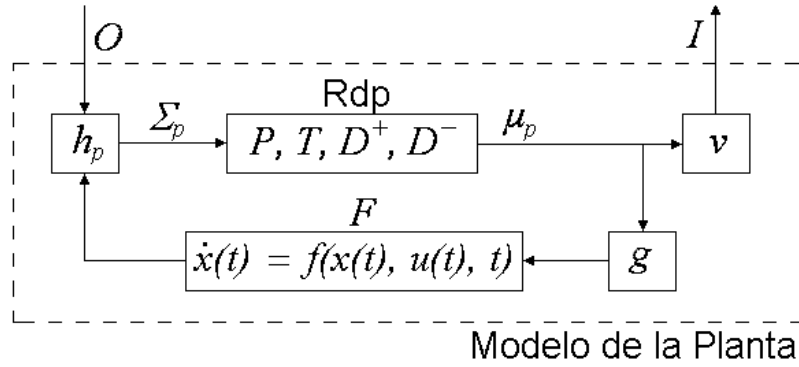
*Definición 2.12:* Una transición es observable si se conoce cuando la transición es disparada (Moody y Antsaklis, 1998).

*Definición 2.13:* Una transición controlable es una transición observable en la que se tiene control sobre la ejecución del evento asociado a la transición, en otras palabras una transición es controlable si el evento asociada a ella es controlable (Moody y Antsaklis, 1998).

## **2.3 Bases para el modelado de sistemas híbridos**

Existen varios tipos de sistemas híbridos, en este trabajo se tratarán los sistemas continuos controlados por entradas discretas. La definición (2.7) que se dio de sistemas híbridos se deriva directamente de la definición (2.4) de sistemas continuos y la definición (2.6) de sistemas de eventos discretos, como es de esperarse para modelar sistemas híbridos debe combinar técnicas de modelado de sistemas continuos y técnicas de modelado de sistemas de eventos discretos. Una herramienta para modelar sistemas de eventos discretos son las redes de Petri (definición 2.11), el estado de la red de Petri, indicado por la marcación  $\mu$ , cambia con la secuencia de eventos de entrada, donde cada evento está asociado a un transición, la evolución del sistema se puede determinar con la ecuación de estado (8). De forma análoga, en un sistema continuo, se representa el cambio de estado en función del tiempo con una ecuación diferencial, como la mostrada en la ecuación (2). El sistema híbrido se modela usando ecuaciones diferenciales y redes de Petri y agregando una función que relaciona ambas dinámicas, esto es lo que se denomina en este trabajo como

red de Petri híbrida. La figura 2.7 muestra un esquema del modelo de un proceso con naturaleza híbrida.



**Figura 2.7. Modelo de una planta híbrida.**

Para modelar sistemas de eventos discretos se pueden usar, además de redes de Petri, otras herramientas como los autómatas, por ejemplo. Cada técnica tiene sus ventajas y desventajas pero en general todas cumplen su función. La principal razón por la cual se usan redes de Petri para el modelo de los sistemas de eventos discretos, radica en el hecho de que, el modelo de la planta, es la unión de los modelos de los equipos que conforman el proceso, el cual, por lo general, es más sencillo y manejable al que resultan de otras técnicas, como por ejemplo los generadores usados con autómatas.

La definición formal para la red de Petri híbrida queda como sigue:

*Definición 2.14:* Una red de Petri híbrida (*RdPH*) está formada por la siguiente tupla (Indriago y Chacón, 2010):

$$RdPH = (P, T, D^+, D^-, \mu_p, F, g, \Sigma_p, h_p, O, I, v) \quad (9)$$

$P = \{p_1, p_2, \dots, p_n\}$  es un conjunto finito de lugares,

$T = \{t_1, t_2, \dots, t_m\}$  es un conjunto finito de transiciones,

$D^+ \in \mathbb{Z}^{n \times m}$  es la matriz de los lugares de salida,

$D^- \in \mathbb{Z}^{n \times m}$  es la matriz de los lugares de entrada,

$\mu_p$  es un vector que contiene la marcación de la red de Petri y tiene dimensión  $n \times I$ ,

$F$  es el conjunto de ecuaciones diferenciales que tienen la forma de la ecuación (2) y que rigen la dinámica continua del sistema híbrido,

$g: \mu_p \rightarrow F$  es la función que relaciona el modelo continuo con la red de Petri,

$\Sigma_p$  es el conjunto de los eventos asociados a las transiciones. Cada evento de  $\Sigma_p$  está relacionado con una transición de  $T$ , esta relación es biyectiva.

$h_p: x \times O \rightarrow \Sigma_p$  es una función que relaciona el estado continuo  $x$  y las acciones externas con los eventos de planta.

$O$ : es el conjunto de las acciones que puede recibir la planta.

$I$ : es el conjunto de todas las repuestas que da la planta para indicar su estado.

$v: \mu \rightarrow I$  es la función que convierte el estado de la planta en las respuesta de la misma.

Para que el modelo esté completo se debe conocer el valor inicial de la marcación ( $\mu_0$ ).

Esta definición de modelo para sistemas híbridos es muy útil al momento de simular los sistemas.

## **2.4 Supervisores**

*Definición 2.15:* Un supervisor es un dispositivo con la capacidad de mantener a un sistema de eventos discretos o a un sistema híbrido dentro de un conjunto de estados deseados.

*Definición 2.16:* Supervisor máximo permisivo es un supervisor que da a la planta el grado más alto posible de libertad durante su evolución (Ramadge and Wonham, 1987; Moody y Antsaklis, 1998).

Un supervisor máximo permisivo sólo deshabilitará un evento en la evolución de la planta cuando la ocurrencia del evento podría violar una restricción directamente o indirectamente a través del disparo de una transición no controlable. (Moody y Antsaklis, 1998).

## **2.5 Arquitecturas**

El término arquitectura se usará en este trabajo dentro de ámbito de la automatización industrial; en ese sentido se explicará el significado de arquitectura de sistema y arquitectura de referencia dentro de este ámbito (Wyns, 1999).

La *arquitectura del sistema* es el resultado de un proceso de diseño para un sistema específico, establece las funciones de los componentes, las interfaces, las interacciones y las restricciones, lo que viene a ser la base para el diseño detallado y la implementación. Además, proporciona los modelos, permite enfocar en las áreas que requieren el mayor cambio, indica los componentes vitales del sistema y las estructuras que no deben violarse al adaptar el sistema a los nuevos usos.

La *arquitectura de referencia* se refiere a un principio de diseño coherente, usado en un dominio específico. Ejemplos de tales arquitecturas son: PROSA (Product Resource Order Staff Architecture), AIACP (Arquitectura de Integración para la Automatización de los Complejos de Producción) AMRF (Automated Manufacturing Research Facility). La arquitectura de referencia describe los tipos de componentes del sistema, sus responsabilidades, dependencias, posibles interacciones y restricciones.

La arquitectura de referencia es la base para diseñar la arquitectura de sistema, además recoge los principios y reglas acerca del desarrollo del sistema en un dominio específico, para lograr una terminología unificada, sin ambigüedades y ampliamente entendible. Además de proporcionar simplicidad en el diseño de la arquitectura de sistema, sistemas de calidad superior, conexión y posibilidad de re-uso entre diferentes proyectos o generaciones del sistema, desarrollo o aplicación de tareas que pueden dividirse entre los equipos diferentes, seguimiento entre la solución de los requerimientos independiente y la realización final.



La arquitectura de referencia describe elementos de dominio del sistema, sus funciones e interacciones, no describe cómo ellos realmente funcionan y actúan recíprocamente en un sistema específico.

*Modelo* es una descripción abstracta de un sistema. El modelo puede tener una o más vistas que enfocan diferentes aspectos del sistema tales como estructura, componentes, relaciones, datos, funciones, interacciones entre otros.

*Marco de modelado* (Modelling framework) describen a los sistemas en varias etapas de su desarrollo descritos desde diferentes puntos de vista. El marco de modelado puede identificar una arquitectura de referencia adecuada. Ejemplo de marcos de modelado de empresas son CIMOSA (Computer Integrated Manufacturing Open System Architecture) y PERA (Purdue Enterprise Reference Architecture).

Las metas de un marco de modelado son similares a las metas de una arquitectura de referencia, asistir al desarrollo de una arquitectura de sistema. Mientras una arquitectura de referencia enfoca sobre una plantilla genérica, el marco de modelado se relaciona a una metodología, métodos, herramientas entre otras para desarrollar una arquitectura de sistema.

## **2.6 Agentes**

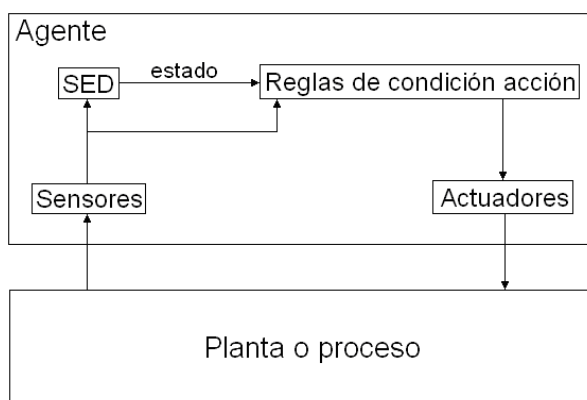
El término agente aparece a mediados de los años 70 viene del latín “*agere*” que significa hacer. Los agentes son usados por la inteligencia artificial, en el ámbito de la automatización industrial, para resolver problemas que son difíciles de solucionar por las teorías clásicas de control, esto ocurre cuando hay que tomar en cuenta muchas variables, existen muchas restricciones y el tiempo para la toma de decisión es muy pequeño.

*Definición 2.17.* “Los agentes son entidades que resuelven problemas claramente identificables con bondades e interfaces bien definidas, situados en un ambiente en particular, diseñados para cumplir un propósito específico, autónomo, reactivo y proactivo” (Wooldridge, 1997).

Esta definición muestra las características del agente que le permiten implementar a un sistema basado en holones. Estas características son:

1. Una interfaz (sensores y actuadores) con el ambiente en donde se encuentra el agente.
2. El agente debe identificar un problema y modificar el ambiente (proceso) para resolverlo.

Además, el agente está diseñado para realizar una tarea específica (metas), tiene autonomía para decidir como alcanzar su meta (reactivo), reacciona a los cambios de ambiente pero también razona para alcanzar estos cambios (pro-activo). Para cumplir con todo lo expuesto, el agente debe reaccionar rápidamente a los eventos y recordar el estado de donde viene, de esta manera puede tomar decisiones rápidas y adecuadas necesarias en el piso de planta de cualquier empresa. Un agente con estas características es denominado “agente reactivo basado en modelo” (Russell y Norvig, 2004) y se muestra en la figura 2.8.



**Figura 2.8. Agente reactivo basado en modelo.**

El sistema de eventos discreto del agente, se denominó modelo de modos y se deriva del modelo de eventos discretos del supervisor. Este modelo arroja información de cuál es el modo de operación en el cual se encuentra el sistema. Las reglas condición acción o reglas de producción (Kowalski, 2006) describen la conducta del agente y tienen la forma:

Si (estas) condiciones (se cumplen) entonces haga (estas) acciones.

Los estados del modelo de modos son condiciones de las reglas de producción, las cuales constituyen la base de conocimiento del agente y deben salir de la experiencia de los expertos en el sistema para el cual está siendo diseñado el agente. Los agentes basados en conocimiento son flexibles y pueden adaptarse a nuevas tareas, simplemente cambiando la base de conocimiento. En las reglas condición acción la inferencia (derivar la conclusión) es simple y se puede hacer rápidamente, lo que le permitirá al agente tomar decisiones en poco tiempo. El conjunto de las reglas de condición acción en unión con el supervisor componen el sistema decisorio del agente.

Para resolver los problemas que se presentan en los procesos industriales se necesitará más de un agente intercambiando información. Los sistemas donde existen más de un agente se denominan sistemas multiagente

## ***2.7 Sistemas de Manufactura Holónicos***

Los sistemas de manufactura holónicos (HMS) se originan de una teoría filosófica sobre la creación y evolución de sistemas adaptativos complejos en el mundo. Combina los conceptos naturales de sistemas jerárquicos y la integración de elementos autónomos en sistemas distribuidos. Una arquitectura basada en sistemas de manufactura holónicos es modelada usando relaciones todo-parte, básicamente, un sistema de manufactura holónico es una organización jerárquica y dinámica de agentes cooperativos y autónomos (Wyns, 1999).

La palabra holón fue introducida en los años sesenta por Arthur Koestler (Koestler, 1967), y sugiere un comportamiento doble, el holón se puede comportar como un todo, pero a su vez, también puede comportarse como una parte (efecto Janus). Los sistemas de manufactura holónicos están formados por comunidades de holones, cada holón puede formar parte de un holón que lo contiene y a su vez estar formado por otros holones. Los sistemas de manufactura holónicos aprovechan las ventajas de los sistemas jerárquicos y de los sistemas heterárquicos; en este sentido, un holón es un ente autónomo con capacidad de tomar sus propias decisiones y a su vez tiene la capacidad de negociar con otros holones en

pro de alcanzar una meta en común. Esta comunidad de holones relacionados entre sí es lo que se conoce como holarquía. Antes se mencionó la preocupación actual por mejorar la calidad, tiempos y costos de los productos en sistemas de producción, para esto existen varias propuestas, algunas de ellas basadas en sistemas de manufactura holónicos, tal es el caso de la “Arquitectura de Integración para la Automatización de Complejos de Producción” (Chacón et al., 2002) que será estudiada en la siguiente sección.

## ***2.8 Arquitectura de Integración para la Automatización de Complejos de Producción***

La Arquitectura de Integración para la Automatización de Complejos de Producción (Chacón et al., 2002), es una propuesta desarrollada en la Universidad de Los Andes, está basada en sistemas de manufactura holónicos y tiene como objetivo fundamental facilitar la implementación del sistema de control en complejos de producción (CP). Esta arquitectura divide a la empresa en cuatro niveles: nivel empresa, nivel complejo de producción, nivel unidad de producción y nivel equipo.

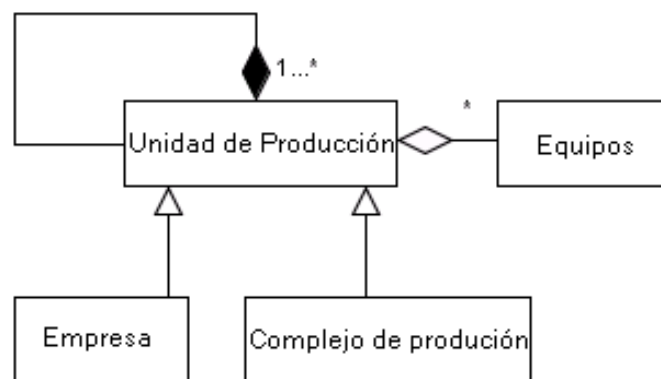
Nivel empresa. En este nivel son ejecutadas las funciones de planificación estratégica y el manejo centralizado de la empresa, el cual decide acerca de las finanzas, recursos humanos, mercado, investigación y actividades de desarrollo. El nivel empresa coordina las actividades de producción ejecutadas en el nivel complejo de producción. El nivel empresa está formado por uno o más sistemas complejos de producción.

Nivel complejo de producción (CP). Este nivel está orientado a coordinar las unidades de producción con el propósito de optimizar los recursos y plan de operaciones. Además, se establecen las cantidades de producción, se envían las órdenes, ejecutan funciones de contabilidad e inventario, se realizan actividades de mantenimiento y se establecen requerimientos de materiales. Un complejo de producción está formado por una o más unidades de producción.

Nivel unidad de producción (UP). En este nivel se ejecutan los procesos de producción. Una UP ejecuta funciones de regulación y control, “conoce” su estado, sus capacidades, sus restricciones y cómo alcanzar sus metas de producción. La UP tiene un alto grado de autonomía y son capaces de negociar con otras UP. Una UP está formada por uno o varios equipos o por otras UP.

Nivel equipo. Los equipos de proceso, los sensores, los equipos de control, los actuadores y las redes de comunicación, están ubicados en este nivel.

Es importante resaltar que el concepto de UP está basado en el concepto de holón, ésta posee autonomía y capacidad de negociación, además cumple con la característica de agregación (puede formar parte de otras unidades de producción) y especialización (puede estar formado por otras unidades de producción). En este mismo sentido, observamos que tanto la empresa como los complejos de producción también pueden ser considerados UP, la figura 2.9 muestra un diagrama UML donde se ilustra esta relación.



**Figura 2.9. Modelo de una arquitectura de empresas.**

Las unidades de producción se integran para formar un complejo de producción, y varios complejos de producción conforman una empresa. En la figura 2.10 se muestra el enfoque dado por la AIACP, el cual difiere del enfoque clásico como el que se muestra en la figura 2.11 y del enfoque heterárquico (Silveira y Combacau, 2003; Saad y otros, 1995) que se muestra en la figura 2.12.

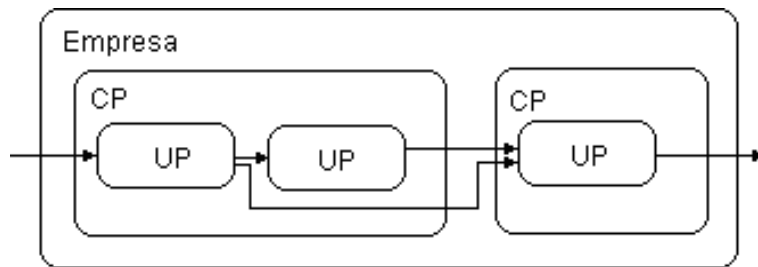


Figura 2.10. Enfoque holónico.

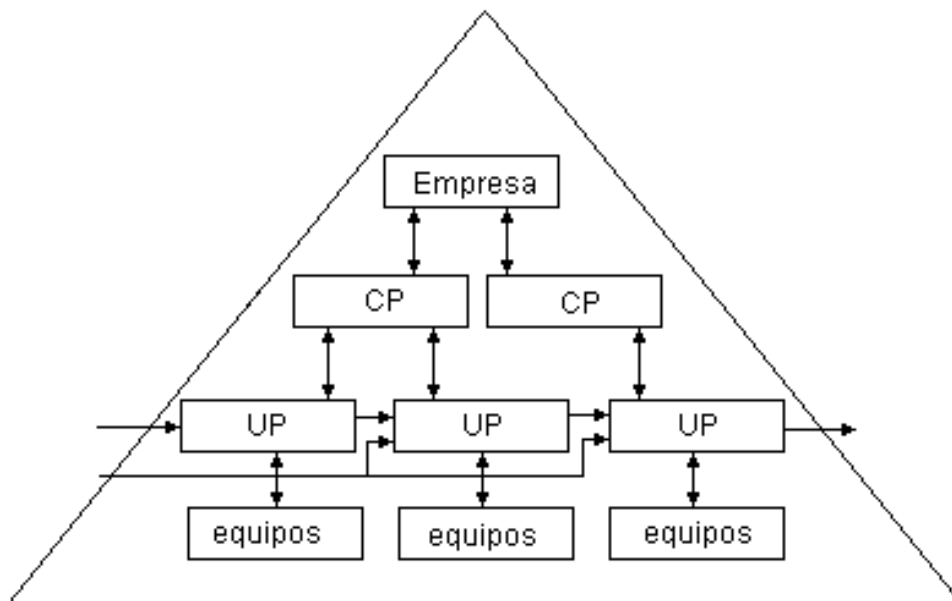


Figura 2.11. Enfoque clásico o jerárquico.

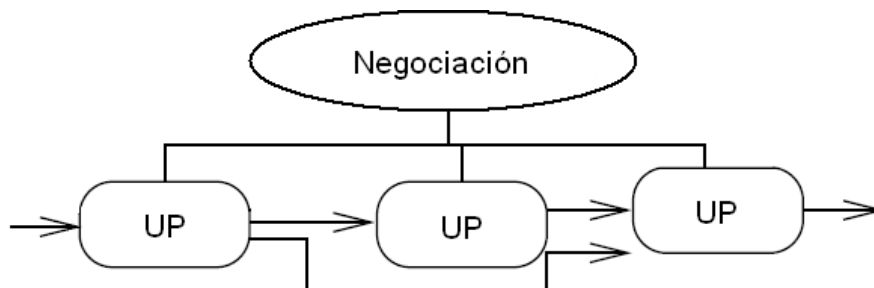
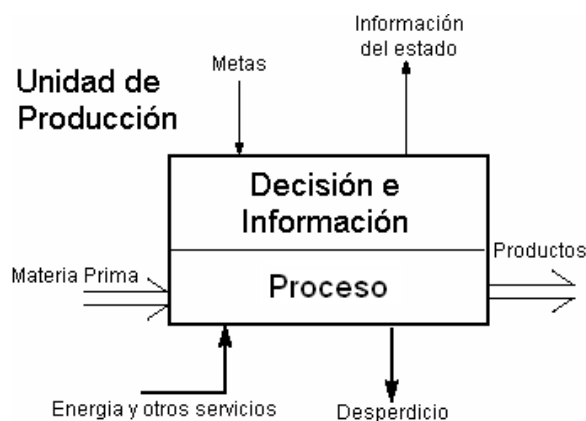


Figura 2.12. Enfoque heterárquico.

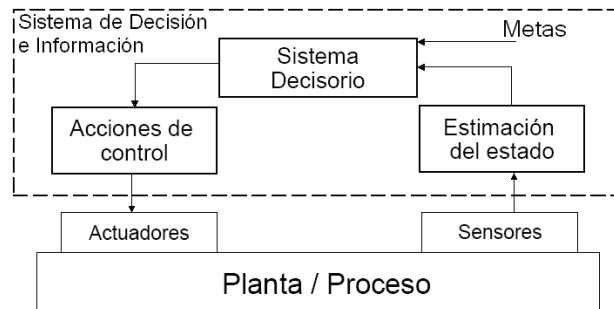
### 2.8.1 Unidad de producción

Dos elementos principales forman a la UP, éstos son el proceso y el sistema de información y toma de decisiones (SI&TD). El proceso contiene los elementos físicos y/o procesos químicos que transforman los materiales de entrada en otros productos. El sistema de información y toma de decisiones se encarga del control y regulación del proceso e intercambia información con otros sistemas para tener el control y regulación global del complejo. La autonomía de las UP implica que éstas están en capacidad de tomar decisiones con respecto a su misión, tales como: cambiar los procedimientos para elaborar el producto, detectar si una meta ha sido modificada e informar si esta meta puede ser aceptada. Además debe informar si una meta ha sido completada o si no puede ser alcanzada debido a alguna situación anormal. La materia prima, las metas, la energía y otros servicios son entradas de la UP, mientras que los productos, información del estado y desperdicio son salidas. La figura 2.13 muestra un esquema de los elementos, las entradas y las salidas de una UP.



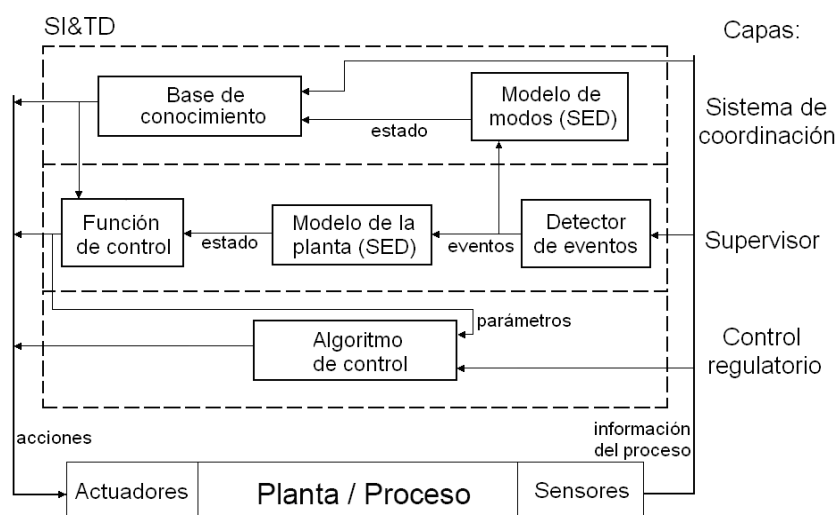
**Figura 2.13. Entradas y salidas de una unidad de producción**

El SI&TD interactúa con el proceso, determinando el estado del mismo a través de sensores, esta medición del estado es comparada con el estado que debe tener el proceso para permitir alcanzar las metas. El resultado de la comparación se evalúa para decidir las acciones que son llevadas al proceso a través de actuadores, esto es lo que se conoce como ciclo de toma de decisiones. La figura 2.14 muestra el ciclo de toma de decisiones.



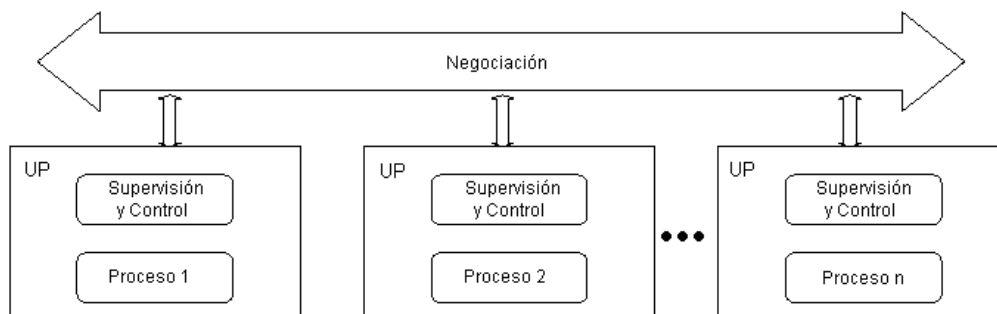
**Figura 2.14. Ciclo de toma de decisiones.**

Para cumplir con todas las funciones, este enfoque divide al SI&TD en tres capas como se muestra en la figura 2.15. A cada capa se le asignan determinadas funciones, haciendo más sencillo el diseño. La primera capa es el control regulatorio, encargado del manejo de las variables analógicas mediante algoritmos de control tales como lazos PID. La segunda capa es el supervisor de eventos discretos, donde se determina el estado del proceso y se establecen las acciones que permiten al proceso mantenerse en un estado deseado. La capa superior es un sistema de coordinación, conformado con agentes capaces de manejar las restricciones, conocer los cambios de metas, cambiar los procedimientos para elaborar un producto, informar si una meta ha sido alcanzada y negociar con otras UP. El proceso de negociación le permite a la UP cooperar para alcanzar la meta en común del sistema al que pertenece. La figura 2.16 muestra un esquema de negociación entre UP.



**Figura 2.15. Capas del sistema de decisión e información en conexión con el proceso.**





**Figura 2.16. Esquema de negociación entre unidades de producción.**

Cada capa cumple sus funciones e intercambia información con las otras dos. Del control regulatorio se extraen los eventos que sirven de entrada al supervisor y al sistema de coordinación, a su vez las acciones que se generan en el supervisor cambian los parámetros en los algoritmos del control regulatorio. De manera similar, el sistema de coordinación maneja un modelo de eventos discretos que indica el modo de operación en el que se encuentra la planta, los eventos de entrada a este modelo provienen del modelo de la planta que se encuentra en el supervisor. Además, el sistema de coordinación genera acciones que modifican la conducta del supervisor y del control regulatorio.

El intercambio de información entre las capas puede ser de diferentes formas dependiendo, de la arquitectura donde se implemente el SI&TD; cada capa puede residir en equipos diferentes o en el mismo equipo de control. Durante la etapa de diseño se debe definir el intercambio de información entre las diferentes capas para incluir la estructura de datos necesaria para este fin. En el sistema de información y toma de decisiones no es obligatorio la existencia de las tres capas, se puede prescindir de una o dos capas para aplicaciones simples.

El ciclo de toma de decisiones de la figura 2.14 se repite en todas las capas del SI&TD (Chacón et al., 2008). El sistema decisorio para el control regulatorio es un algoritmo de control con una variable del proceso como entrada, la cual se quiere mantener igual a una consigna o referencia, en el supervisor es un modelo de eventos discretos y una función de control que tienen como entrada la secuencia de eventos del proceso y para el coordinador

es un conjunto de reglas lógicas con los valores de las premisas como entrada. En cada capa se pueden presentar situaciones no deseadas, en el control regulatorio una diferencia muy grande entre la variable de proceso y la consigna o referencia genera una alarma, en el supervisor determinados estados indican condiciones de operación fuera de la región permitida y el coordinador combina la información de las capas inferiores con la información que llega (o no llega) de otras unidades de producción para determinar condiciones de fallas locales o problemas en las negociaciones.

## **2.9 Conclusión al capítulo**

Para comprender mejor los procesos industriales se modelan éstos como sistemas dinámicos y se aprovecha todo el estudio al respecto para su análisis y control. Los sistemas dinámicos de nuestro interés son los de variable continua, los de variables discreta controlados por eventos, y los que contienen ambas dinámicas los cuales se denomina híbridos. Las dinámicas continuas son modeladas con ecuaciones diferenciales y las dinámicas discretas gobernadas por eventos son modeladas con redes de Petri. Los modelos son usados para estudiar los procesos y diseñar los supervisores, los cuales son integrados bajo los lineamientos de una arquitectura de referencia para formar una arquitectura de sistema. El supervisor se vale de los actuadores para modificar la conducta del proceso.

Para modelar sistemas híbridos se combinan técnicas de modelado de sistemas continuos y técnicas de modelado de sistemas de eventos discretos, en este trabajo el sistema híbrido se modela usando ecuaciones diferenciales y redes de Petri y agregando una función que relacione ambas dinámicas discreto con los eventos. Esta definición de modelo para sistemas híbridos es muy útil al momento de realizar simulaciones.

Los sistemas de manufactura holónicos (HMS) combinan los conceptos naturales de sistemas jerárquicos y la integración de elementos autónomos en sistemas distribuidos. La Arquitectura de Integración para la Automatización de Complejos de Producción es una propuesta desarrollada en la Universidad de Los Andes y está basada en sistemas de manufactura holónicos y tiene como objetivo fundamental facilitar la implementación del

sistema de control en complejos de producción continuos (CP). Está formada por tres niveles, el nivel empresa, el nivel complejo de producción y el nivel unidad de producción. Dos elementos principales forman a la UP estos son el proceso y el sistema de información y toma de decisiones (SI&TD). El SI&TD contiene un control regulatorio, un supervisor y un sistema de coordinación.

## **Capítulo 3. Sistema Supervisor**

La capa de supervisión de la unidad de producción detecta los eventos que suministran información y permiten determinar el estado del proceso. Si el estado resulta fuera de la región deseada se ejecutan las acciones necesarias para corregirlo. Algunos autores (Frey, 1998; Zapata y Carrasco, 2002) utilizan redes de Petri interpretadas, debido a que éstas incluyen en su definición un conjunto de acciones que se ejecutan según el estado de la red.

Para diseñar un supervisor se necesita una descripción formal (modelo) del proceso y algún método para síntesis de supervisores. El modelo de la planta está relacionado con el tipo de dinámica de la misma, una planta con dinámicas continuas es modelada por ecuaciones diferenciales (Tittus et al., 1994), una planta con dinámica de eventos discretos es modelada con redes de Petri (Polic y Jezernik, 2005; Zapata y Carrasco, 2002) y una planta con dinámicas híbridas es modelada con ecuaciones diferenciales y redes de Petri (Frey, 1998). En este trabajo se estudian con más detalles las plantas con dinámicas híbridas.

### **3.1 El supervisor**

El diseño del supervisor depende de las características de la planta, su dinámica, el modelo que se derive de ella, los eventos que se puedan observar y los eventos que se puedan controlar. Son bien conocidos los trabajos de síntesis de supervisores para sistemas de eventos discretos basados en autómatas (Ramadge y Wonham, 1987), donde el supervisor mantiene la conducta del proceso en una región deseada habilitando y deshabilitando eventos según el estado en que se encuentre. Para los supervisores desarrollados con redes de Petri, en la mayoría de los casos el diseño del supervisor resulta en agregar arcos y lugares al modelo de la planta. A continuación se describe el modelo para el supervisor propuesto.

#### **3.1.1 Modelo del supervisor**

El supervisor propuesto está formado por un detector de eventos, un modelo de la planta y una función de control. El ciclo de decisión se realiza dentro del supervisor de la siguiente

forma: las señales provenientes de los sensores son transformadas en eventos que sirven de entrada al modelo de la planta en donde se determina el estado; a su vez, la función de control ejecuta las acciones dependiendo del estado del modelo.

De este ciclo se deriva la siguiente definición:

*Definición 3.1:* Un supervisor ( $S$ ) está formado por la siguiente tupla (Indriago y Chacón, 2010):

$$S = (P, T, D^+, D^-, \mu_S, \Sigma_S, I, O, h_S, y) \quad (10)$$

$P = \{p_1, p_2, \dots, p_n\}$  es un conjunto finito de lugares,

$T = \{t_1, t_2, \dots, t_m\}$  es un conjunto finito de transiciones,

$D^+ \in \mathbb{Z}^{n \times m}$  es la matriz de los lugares de salida,

$D^- \in \mathbb{Z}^{n \times m}$  es la matriz de los lugares de entrada,

$\mu_S$  es un vector que contiene la marcación de la red de Petri y tiene dimensión  $n \times 1$ ,

$\Sigma_S = \Sigma_{ST} \cup \Sigma_{Sc}$  es el conjunto de los eventos, formado por los eventos asociados a las transiciones ( $\Sigma_{ST}$ ) y los eventos que producen el resto de las condiciones ( $\Sigma_{Sc}$ ). Cada evento de  $\Sigma_{ST}$  está relacionado con una transición de  $T$ , esta relación es biyectiva.

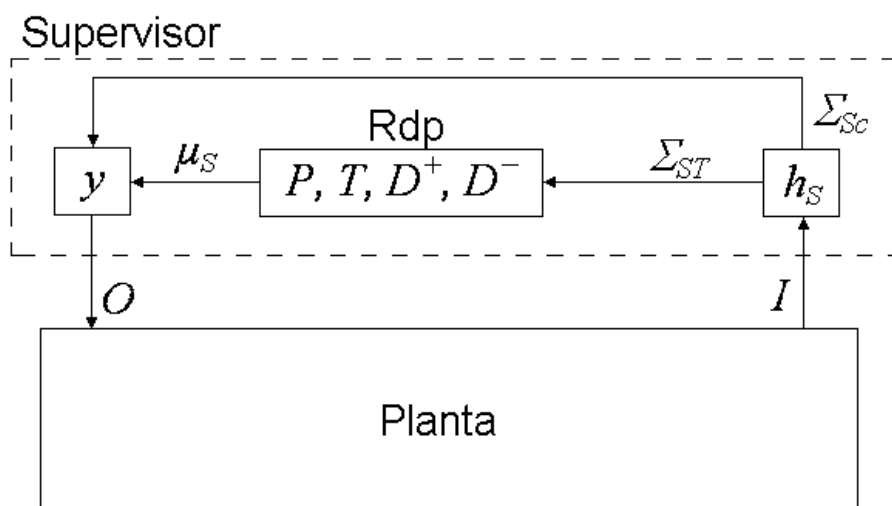
$I$ : es el conjunto de las entradas al supervisor,

$O$ : es el conjunto de salida del supervisor,

$h_S: I \rightarrow \Sigma_S$  es el detector de eventos donde se relacionan las entradas con los eventos,

$y: \mu_S \times \Sigma_{Sc} \rightarrow O$  es una función que relaciona al estado de la red de Petri con la salida.

Para que el modelo esté completo se debe contar con el estado inicial del mismo que se representa con la marcación inicial ( $\mu_{S0}$ ). La figura 3.1 muestra un ejemplo del modelo del supervisor conectado a una planta.



**Figura 3.1. Modelo del supervisor.**

Como ya se mencionó, el detector de eventos realiza las operaciones necesarias sobre las entradas para obtener los eventos. Cada evento puede provenir de una o más entradas, diferentes eventos pueden estar relacionados con las mismas entradas.

### **3.2 Marco conceptual para la implementación del supervisor.**

Para la implementación del supervisor en un proceso industrial se debe contar con una estructura que sirva para empotrar el supervisor, esta estructura involucra aspectos tales como los equipos, los tipos de actuadores, las variables que se van a intercambiar entre la planta y el controlador, entre otros. Los equipos que forman parte del proceso se mencionan a continuación:

Equipos de control, donde residen los algoritmos y estrategias para controlar el proceso. Ejemplo de equipos de control son los controladores lógicos programables, computadores personales, sistemas de control distribuidos, entre otros.

Interfaz de operación, se usan para que el operador observe el estado, indique acciones o cambie los parámetros del sistema. Algunas estaciones tienen la capacidad para almacenar datos. Las estaciones de operación intercambian datos con los equipos de control. Ejemplos de interfaz de operación son: luces pilotos y pulsadores, pantallas alfanuméricas, pantallas gráficas, entre otros

Equipos de medición o sensores, pasan información a los equipos de control del estado del proceso. Ejemplo de estos equipos son: sensores de posición (detectores de proximidad), transmisores (presión, temperatura, flujo, nivel), entre otros.

Equipos de proceso, tienen como función darle valor agregado al producto. Ejemplo de equipos de proceso son los tanques, reactores, intercambiadores, entre otros.

Actuadores, las acciones del control se ejecutan en los actuadores y tienen como consecuencia un cambio de estado en el proceso. Ejemplo de actuadores son los motores y las válvulas.

La figura 3.2 muestra un ejemplo del sistema de control de nivel de un tanque en el que se pueden observar los siguientes equipos: un equipo de control (controlador lógico programable YC2), una interfaz gráfica (LI101), dos sensores (detectores de nivel LSH101 y LSL101), un equipo de proceso (Tanque1) y dos actuadores (válvulas YV101A y YV101B).

Una lista que enumere los equipos y una lista con todas las señales de entradas y salidas del controlador deben formar parte de las especificaciones para el diseño del supervisor. Los equipos de proceso se valen de los actuadores para cambiar su estado. Los actuadores tienen funciones equivalentes a los módulos de control que se encuentran en el modelo físico de la norma s88.1 de la ISA (ISA, 1995) que se muestra en la figura 3.3. La UP tiene funciones similares a una Unidad en este modelo.

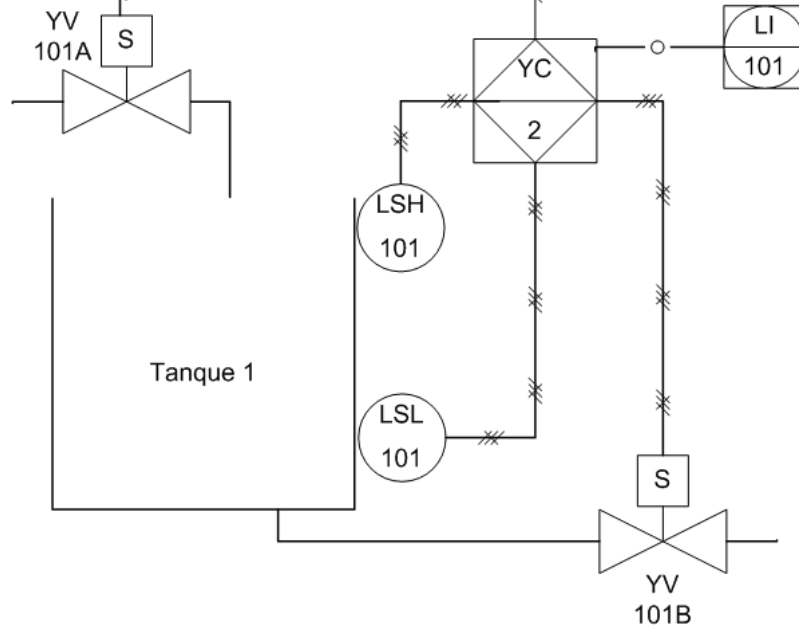


Figura 3.2. Ejemplo de un tanque con dos válvulas.

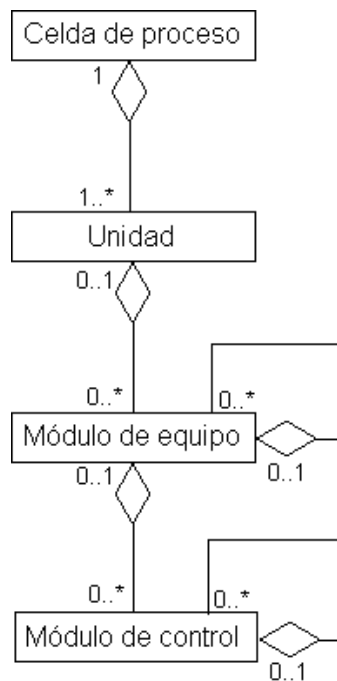


Figura 3.3. Cuatro niveles inferiores del modelo físico en la norma s88.1.



### 3.2.1 Actuadores

Existe una gran variedad de actuadores que pueden estar presentes en los procesos; en este trabajo se consideran cuatro tipos de uso muy común (Indriago y Lucena, 2004) que se diferencian entre ellos por el comportamiento de las entradas y salidas (se denominan entradas a las señales que van desde el actuador hacia el controlador y salidas las que van desde el controlador hacia el actuador). Los tipos de actuadores se describen a continuación:

Tipo 1. Una entrada y una salida. El controlador envía un comando de salida y el actuador responde confirmando que lo ha recibido a través de una entrada. Un ejemplo típico de este tipo de equipos son los motores.

Tipo 2. Dos entradas y una salida. Este tipo de actuadores tiene dos posiciones, cada entrada se activa para indicar que el actuador se encuentra en alguna de las dos posiciones. El controlador da un comando de salida para ir de una posición a otra y lo quita para regresar al estado inicial. Un ejemplo de estos actuadores son las válvulas con un solenoide de activación y reposición por resorte.

Tipo 3. Dos entradas y dos salidas. Son muy similares al anterior, con la diferencia que para ir de una posición a otra se activa una de las salidas y para regresar se activa la otra salida. Un ejemplo de estos tipos de actuadores son las válvulas de dos solenoides.

Tipo 4. Una entrada y dos salidas. Estos actuadores tienen dos modos de operación, cada modo es activado por una de las salidas. La confirmación de operación de ambos modos se recibe por la misma entrada. Un ejemplo de estos actuadores son los motores reversibles.

Además de la lista de equipos y la lista de entradas y salidas, también debe haber una filosofía de operación como parte de las especificaciones, en ella se indica las condiciones que debe haber para ejecutar determinada acción.

Cada actuador tiene asociado un conjunto de variables lógicas mediante las cuales se conoce el estado del actuador o se ejecuta una acción sobre él. La tabla 3.1 describe estas variables y en que tipo de actuador aparecen.

**Tabla 3.1. Variables asociadas a los actuadores.**

Nombre	Siglas	Descripción	Tipo de modulo de control			
			1	2	3	4
Modo	<i>MO</i>	Cambia el modo del actuador entre automático (verdadero) y manual (falso)	si	si	si	si
Arranque manual	<i>AM</i>	Se coloca en verdadero para un arranque en manual	si	si	si	si
Arranque automático	<i>AA</i>	Se coloca en verdadero para un arranque en automático	si	si	si	si
Dirección	<i>DI</i>	Indica la dirección de acción en los actuadores de dos salidas	no	no	si	si
Comando de salida	<i>CS</i>	Manipulando esta variable se logra la acción sobre el actuador	si	si	si	si
Comando de salida inverso	<i>CI</i>	Manipulando esta variable se logra la acción sobre los actuadores de dos salidas	no	no	si	si
Entrada 1	<i>NA</i>	Indica el estado del actuador	si	si	si	si
Entrada 2	<i>NC</i>	En conjunto con la entrada 1 indica el estado del actuador en los actuadores de dos entradas	no	si	si	no
Permisivos	<i>PE</i>	Agrupar el estado de todos los permisivos del actuador, en condiciones de operación tiene un valor de verdadero	si	si	si	si
Alarmas	<i>AI</i>	Agrupar el estado de todas las alarmas del actuador excepto la de falla de respuesta al comando de salida, en condiciones de operación tiene un valor de verdadero	si	si	si	si
Falla de respuesta al comando de salida	<i>FR</i>	Se coloca en verdadero cuando hay una inconsistencia entre los comandos de salidas y las entradas	si	si	si	si
Alarma general	<i>AL</i>	Se coloca en verdadero cuando hay una alarma o falla de la respuesta al comando de salida	si	si	si	si
Borrar alarmas	<i>BO</i>	Restablece los valores de <i>AI</i> , <i>FR</i> y <i>AL</i>	si	si	si	si

Las variables de modo, arranque manual, dirección y borrar normalmente son manipuladas en la interfaz de operación, la cual puede leer todas las demás variables con fines de visualización. Las variables de arranque en automático y dirección cambian de estado en la

función de control. El valor de las entradas 1 y 2 depende directamente del estado del actuador. Las variables comando de salida, comando de salida inverso, falla de respuesta de marcha, permisivos, alarmas y alarma general son determinadas por una función denominada función estándar. La función estándar es intrínseca del actuador, está programada en el controlador y es independientemente del proceso. La función estándar se encarga de ejecutar la acción, limpiar alarmas, verificar la respuesta de marcha y desactivar el actuador cuando ocurre una alarma, una parada de emergencia o un cambio de modo. Una de las características de la función estándar, es que permite manejar el proceso en modo manual, incluso antes del diseño del supervisor. A continuación se muestran las reglas que forman la función estándar expresadas en premisas lógicas para un actuador tipo 1 (una entrada y una salida) con  $j$  permisivos y  $k$  alarmas:

1.  $((AM \wedge \neg MO) \vee (AA \wedge MO)) \wedge \neg AL \wedge PE \Leftrightarrow CS$
2.  $\neg AL \wedge (CS \wedge \neg NA) \vee (\neg CS \wedge NA) \Rightarrow FR$
3.  $FR \vee \neg AI \Rightarrow AL$
4.  $BO \Rightarrow \neg AL$
5.  $BO \Rightarrow \neg FR$
6.  $BO \Rightarrow AI$
7.  $MO \vee AL \vee PARADA\ EMERGENCIA \Rightarrow \neg AM$
8.  $\neg MO \vee AL \vee PARADA\ EMERGENCIA \Rightarrow \neg AA$
9.  $\neg PERMISIV\ O_1 \vee \dots \vee \neg PERMISIV\ O_j \Leftrightarrow PE$
10.  $\neg ALARMA_1 \vee \dots \vee \neg ALARMA_k \Leftrightarrow AI$

Para un actuador tipo 2 (dos entradas y una salida) con  $j$  permisivos y  $k$  alarmas, el conjunto de premisas es el siguiente:

1.  $((AM \wedge \neg MO) \vee (AA \wedge MO)) \wedge \neg AL \wedge PE \Leftrightarrow CS$
2.  $\neg AL \wedge ((CS \wedge (\neg NA \vee NC)) \vee (\neg CS \wedge (NA \vee \neg NC))) \Rightarrow FR$
3.  $FR \vee \neg AI \Rightarrow AL$

4.  $BO \Rightarrow \neg AL$
5.  $BO \Rightarrow \neg FR$
6.  $BO \Rightarrow AI$
7.  $MO \vee AL \vee PARADA EMERGENCIA \Rightarrow \neg AM$
8.  $\neg MO \vee AL \vee PARADA EMERGENCIA \Rightarrow \neg AA$
9.  $\neg PERMISIV O_1 \vee \dots \vee \neg PERMISIV O_j \Leftrightarrow PE$
10.  $\neg ALARMA_1 \vee \dots \vee \neg ALARMA_k \Leftrightarrow AI$

Para un actuador tipo 3 (dos entradas y dos salidas) con  $j$  permisivos y  $k$  alarmas, las premisas de la función estándar quedan como se muestra a continuación.

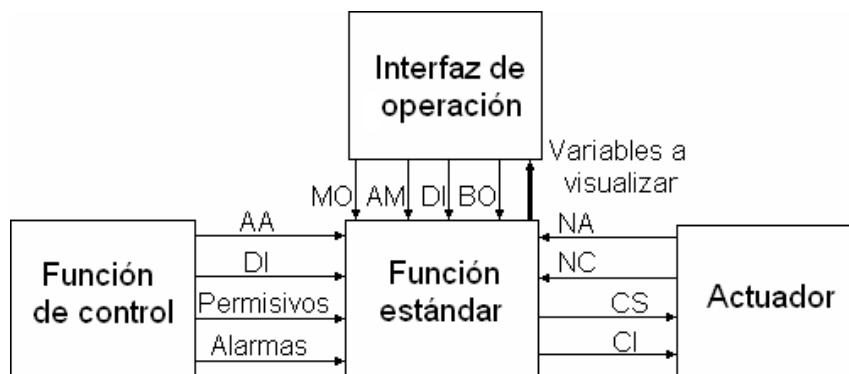
1.  $((AM \wedge \neg MO) \vee (AA \wedge MO)) \wedge \neg AL \wedge PE \wedge DI \wedge \neg NA \Leftrightarrow CS$
2.  $((AM \wedge \neg MO) \vee (AA \wedge MO)) \wedge \neg AL \wedge PE \wedge \neg DI \wedge \neg NA \Leftrightarrow CI$
3.  $\neg AL \wedge ((\neg NA \wedge \neg NC) \vee (NA \wedge NC) \vee CS \vee CI) \Rightarrow FR$
4.  $FR \vee \neg AI \Rightarrow AL$
5.  $BO \Rightarrow \neg AL$
6.  $BO \Rightarrow \neg FR$
7.  $BO \Rightarrow AI$
8.  $MO \vee AL \vee PARADA EMERGENCIA \vee (DI \wedge NA) \vee (\neg DI \wedge NC) \Rightarrow \neg AM$
9.  $\neg MO \vee AL \vee PARADA EMERGENCIA \vee (DI \wedge NA) \vee (\neg DI \wedge NC) \Rightarrow \neg AA$
10.  $\neg PERMISIV O_1 \vee \dots \vee \neg PERMISIV O_j \Leftrightarrow PE$
11.  $\neg ALARMA_1 \vee \dots \vee \neg ALARMA_k \Leftrightarrow AI$

Para un actuador tipo 4, las premisas se muestran a continuación:

1.  $((AM \wedge \neg MO) \vee (AA \wedge MO)) \wedge \neg AL \wedge PE \wedge DI \Leftrightarrow CS$
2.  $((AM \wedge \neg MO) \vee (AA \wedge MO)) \wedge \neg AL \wedge PE \wedge \neg DI \Leftrightarrow CI$
3.  $\neg AL \wedge ((CS \vee CI) \wedge \neg NA) \vee (\neg CS \wedge \neg CI \wedge NA) \Rightarrow FR$
4.  $FR \vee \neg AI \Rightarrow AL$

5.  $BO \Rightarrow \neg AL$
6.  $BO \Rightarrow \neg FR$
7.  $BO \Rightarrow AI$
8.  $MO \vee AL \vee PARADA EMERGENCIA \Rightarrow \neg AM$
9.  $\neg MO \vee AL \vee PARADA EMERGENCIA \Rightarrow \neg AA$
10.  $\neg PERMISIV O_i \vee \dots \vee \neg PERMISIV O_j \Leftrightarrow PE$
11.  $\neg ALARMA_i \vee \dots \vee \neg ALARMA_k \Leftrightarrow AI$

La figura 3.4 muestra el flujo de las variables entre la función de control, la interfaz de operación, la función estándar y el actuador.



**Figura 3.4. Flujos de las variables del actuador.**

Para ilustrar analizaremos la acción de abrir y la detección de una alarma en la válvula YV101A del ejemplo de la figura 3.2. Si tenemos en consideración que las válvulas YV101A y YV101B son de dos solenoides, entonces se clasifican como actuadores de dos entradas (NA=1: cerrado y NC=1: abierto) y dos salidas (CS=1: cerrar y CI=1: abrir), tipo 3. Tanto la función de control, como la función estándar, están programadas dentro del PLC YC2. Si se quiere abrir la válvula YV101A en forma automática, la función de control coloca un valor de verdadero en la variable “arranque en automático” (YV101A\_AA=1) para indicar al actuador que accione, además coloca un valor de verdadero en la variable “dirección” (YV101A\_DI=1), para indicar que la acción va en el sentido de abrir la válvula.

La función estándar responde colocando un valor de verdadero en la variable “comando de salida” (YV101A\_CS=1), esto hace que el actuador se desplace, y cuando llega a su destino responde colocando un valor de verdadero en la variable “entrada 1” (YV101A\_NA=1) y un valor de falso en la variable “entrada 2” (YV101A\_NA=0); en este punto, la función estándar coloca un valor de falso en la variable “comando de salida” (YV101A\_CS=0) para detener la acción. Si la válvula pierde posición, por ejemplo, que indique cerrada y abierta simultáneamente (YV101A\_NA = YV101A\_NC = 1), la función estándar coloca a verdadero las variables “falla de respuesta al comando de salida” (YV101A\_FR = 1) y “alarma general” (YV101A\_AL = 1), además de desactivar las salidas (YV101A\_CS = YV101A\_CS = 0), opcionalmente, se puede mostrar esta información en la interfaz gráfica LI101.

El modelo en el supervisor está compuesto por los modelos de los equipos proceso y actuadores que forman parte de la planta. Para este trabajo ninguna transición es controlable, en el sentido de la definición 2.13, debido a que el supervisor genera una acción que eventualmente produce el evento y dispara la transición; sin embargo, el hecho de que se genere la acción no garantiza la ocurrencia del evento. Las figuras de la 3.5 a la 3.8 muestran el modelo en redes de Petri de los actuadores tipo 1, 2 3 y 4 que se mencionaron antes. En el actuador tipo 1, la transición  $t_2$  se dispara cuando la entrada está presente y la transición  $t_1$  se dispara cuando la entrada está ausente. La transición  $t_4$  se dispara cuando ocurre una falla mientras que la transición  $t_3$  se dispara cuando la falla desaparece. Si el actuador se trata de un motor y se quiere poner éste en marcha, no se puede disparar  $t_2$  a voluntad, en su lugar se ejecuta una acción que activa la salida del actuador para energizar el arrancador que coloca en marcha al motor, si todas las condiciones están dadas; como consecuencia de la acción, se activa la entrada del actuador y esto genera el evento que dispara la transición  $t_2$  (en este punto se puede ver el concepto de lazo cerrado en el actuador). Si al pasar un tiempo determinado no se recibe la confirmación de que el motor está en marcha, entonces, se genera el evento que dispara  $t_4$  indicando que el motor está en falla. En este sentido se puede ver que ninguna transición del sistema es controlable, las que existen son todas observables.

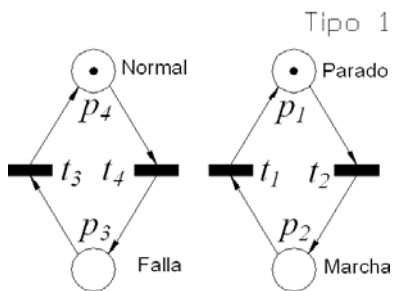


Figura 3.5. Modelo del comportamiento de un actuador tipo 1.

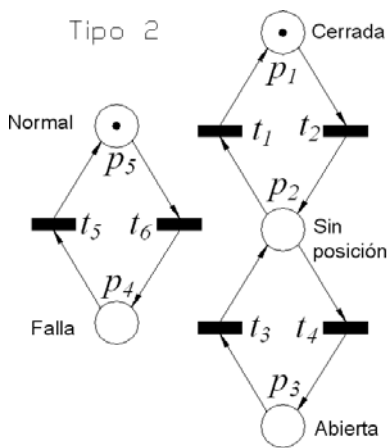


Figura 3.6. Modelo del comportamiento de un actuador tipo 2.

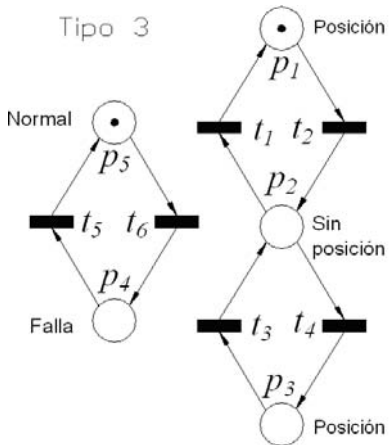


Figura 3.7. Modelo del comportamiento de un actuador tipo 3.

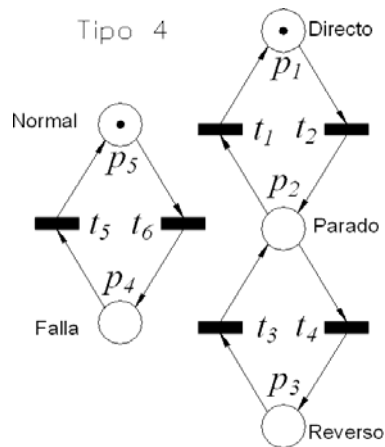


Figura 3.8. Modelo del comportamiento de un actuador tipo 4.

### 3.3 Síntesis de Supervisores

El modelo de la planta dentro del equipo de control es un modelo de lazo abierto (modelo de la planta sin tomar en cuenta el supervisor), el cual cambia de estado con la información proveniente del detector de eventos, mientras la función de control ejecuta acciones para modificar el estado de la planta a través de los actuadores. El conjunto formado por el modelo de lazo abierto de la planta y la función de control, forman el modelo a lazo cerrado de la planta, tal como lo indica la ecuación (11).

$$M_{la} + M_y = M_{lc} \quad (11)$$

Donde

$M_{la}$ : modelo de lazo abierto.

$M_y$ : modelo en redes de Petri de la función de control.

$M_{lc}$ : modelo de lazo cerrado.

Si al modelo de lazo abierto se le aplica algún método de síntesis de supervisores encontraremos un modelo de lazo cerrado que es una solución para el sistema de control.



Dos métodos para síntesis de supervisores que mencionaremos son el método de invarianza de lugares y el método de condición - acción. En el método de invarianza de lugares, al modelo de lazo abierto de la planta se le aplica una serie de restricciones, que indican lo que la planta no debe hacer, y de esta manera obtener el modelo de lazo cerrado. En el método de condición – acción, al modelo de la planta se le aplica un conjunto de reglas, que indican lo que la planta debe hacer, para obtener el modelo de lazo cerrado del proceso.

Si al modelo de lazo cerrado se le quita la parte del modelo de lazo abierto, lo que queda es el equivalente en red de Petri de la función de control.

$$M_y = M_{lc} - M_{la} \quad (12)$$

Las reglas y restricciones del proceso se colocan en la filosofía de operación, de allí se deriva la función de control, la cual está compuesta por una serie de premisas lógicas que tienen como argumentos la marcación del proceso ( $\mu$ ) y las condiciones de entradas ( $\Sigma_{Sc}$ ) y como salidas las acciones que se ejecutan sobre los actuadores ( $O$ ). Cada acción que se ejecuta depende del tipo de actuador, para los actuadores tipo 1 se ejecutan las acciones de marcha y parada, para los tipo 2 y 3 se ejecutan las acciones de abrir y cerrar o la acción de cambio de posición y para los tipo 4 se ejecutan las acciones de marcha directa, marcha reversa y parada. En la función de control también se especifican los permisivos y alarmas de cada actuador. A continuación se muestra la estructura de una función de control para un sistema con actuadores:

*actuador<sub>1</sub> (Tipo 1):*

- *condición de marcha  $\Rightarrow AA$*
- *condición de parada  $\Rightarrow \neg AA$*
- *lista de permisivos*
- *lista de alarmas*

*actuador<sub>2</sub>*

...

*actuador<sub>n</sub>*

La condición de marcha y la condición de parada son variables lógicas obtenidas como resultado de operaciones que tienen como argumentos la marcación de la red de Petri ( $\mu$ ) y elementos del conjunto de los eventos de condiciones ( $\Sigma_{Sc}$ ). Cada permisivo de la lista es una variable lógica, si algún permisivo de la lista toma el valor de verdadero entonces la variable *PE* adquiere un valor de falso indicando que el actuador no tiene permiso. Las alarmas funcionan de forma similar, si alguna de las alarmas adquiere el valor de verdadero la variable *AI* toma el valor de falso, esto causa que el equipo se detenga.

En el proceso de diseño del supervisor se debe analizar el modelo de lazo cerrado de la planta para determinar estados indeseados como bloqueos, las propiedades de alcance y vivacidad de las redes de Petri son útiles para este fin. El análisis del modelo permite mejorar la filosofía de operación, lo que se traduce en una función de control más eficiente.

### **3.3.1 Síntesis de supervisores usando invarianza de lugares**

Existen diferentes propuestas para el desarrollo de supervisores con redes de Petri, Murata (1989), por ejemplo, resuelve una serie de casos típicos, cuales se presentan frecuentemente en el diseño de supervisores. En otro enfoque se modela la planta usando algebra (MAX, +) y se aplican una serie de restricciones lo que permite encontrar una expresión para el supervisor (Martínez, 2008). En general, cualquier herramienta que permita diseñar supervisores modelados con redes de Petri sirve para esta propuesta. A continuación se presenta la síntesis de supervisores para sistemas de eventos discretos, modelados con redes de Petri, basados en el método de invarianza de lugares propuesto por Moody y Antsaklis (1998).

Las restricciones que se toman en cuenta para el diseño del supervisor usando invarianza de lugares, indican que el número de marcas en determinados lugares, no debe ser mayor que un vector *b*, esto se puede escribir de la siguiente forma:

$$L\mu_p \leq b \quad (13)$$

donde  $\mu_p$  es la marcación de la planta,  $L \in \mathbb{Z}^{n_c \times n}$  es una matriz de enteros,  $b \in \mathbb{Z}^{n_c}$ ,  $n$  es el número de lugares y  $n_c$  es el número de restricciones. La desigualdad (13) se puede escribir de la forma:

$$L\mu_p + \mu_c = b \quad (14)$$

donde  $\mu_c \in \mathbb{Z}^{n_c}$  es un vector de enteros que representa la marcación de los lugares del supervisor o controlador. El controlador también tiene una matriz de incidencia denominada  $D_c$  que contiene arcos que conectan los lugares del controlador con las transiciones de la planta. La matriz de incidencia total (planta más controlador)  $D \in \mathbb{Z}^{(n+n_c) \times m}$  está dado por

$$D = \begin{bmatrix} D_p \\ D_c \end{bmatrix} \quad (15)$$

la marcación del conjunto planta - controlador  $\mu \in \mathbb{Z}^{n+n_c}$  y la marcación inicial  $\mu_0$  son:

$$\mu = \begin{bmatrix} \mu_p \\ \mu_c \end{bmatrix}, \quad \mu_0 = \begin{bmatrix} \mu_{p_0} \\ \mu_{c_0} \end{bmatrix} \quad (16)$$

La ecuación (14) denota invarianza, el sistema formado por las ecuaciones (15) y (16) debe satisfacer la siguiente ecuación.

$$X^T D = 0 \quad (17)$$

lo que garantiza la invarianza en la red de Petri. La ecuación (15) puede escribirse de la forma

$$X^T D = [LI] \begin{bmatrix} D_p \\ D_c \end{bmatrix} \quad (18)$$

que a su vez se puede escribir como

$$D_c = -LD_p \quad (19)$$

y de la ecuación (14) se puede deducir la marcación inicial de los lugares del controlador

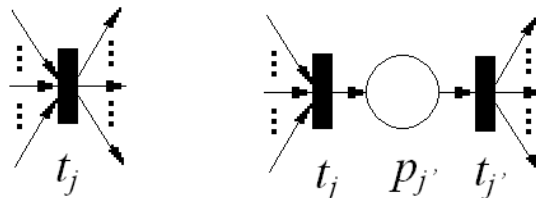
$$\mu_{co} = b - L\mu_{po} \quad (20)$$

Las ecuaciones (19) y (20) describen el controlador con su marcación inicial para restricciones de la forma (13), sin embargo se pueden escribir las restricciones de la forma

$$\mu_i + q_j \leq 1 \quad (21)$$

lo que indica que la transición  $j$  debe estar deshabilitada si el lugar  $i$  contiene una marca. Este tipo de restricciones se puede llevar a la forma de la restricción (13), transformando la transición  $j$  como un lugar  $p_j$  entre dos transiciones  $t_j$  y  $t_j'$  tal como muestra la figura 3.9. Esta transformación sólo se hace para calcular el supervisor, los lugares del supervisor estarán conectados a transiciones, así que la transformación se puede llevar a su forma original. Lo más importante de aplicar la transformación es que la restricción (21) se puede escribir como

$$\mu_i + \mu_{j'} \leq 1 \quad (22)$$



**Figura 3.9. Transformación de una transición en dos transiciones y un lugar.**

que es la misma forma de la restricción (13) y de esta manera se puede calcular el supervisor por la misma vía que antes. Es posible que una restricción genere un supervisor que tenga unas condiciones iniciales que violen la restricción o que inhiba a una transición no controlable, en este caso dicha restricción es inadmisibles. En Moody y Antsaklis, (1998) se puede encontrar la forma de estudiar la admisibilidad y como transformar la restricciones inadmisibles en restricciones admisibles.

### 3.3.2 Supervisores de condición - acción

Los supervisores condición - acción se derivan directamente de las reglas condición - acción de la lógica proposicional. En las reglas condición - acción, si las condiciones son ciertas entonces se ejecuta la acción. En un modelo con redes de Petri en ocasiones queremos una conducta como el de las reglas condición - acción, las condiciones en la red de Petri vienen dadas por la marcación y las acciones se traducen en la habilitación de una transición. En otras palabras, cuando hay una marca en un lugar, y esto indica que la condición es cierta, entonces se debe habilitar una transición, lo que implica que debe ir un arco desde el lugar hacia la transición. Si la transición es disparada entonces eliminaría a la marca del lugar, en la práctica esto no es necesariamente cierto, así que para que la marcación no cambie debe haber un arco desde la transición hacia el lugar. Dicho de otra manera, una regla de condición - acción, aplicada a una red de Petri, introduce un autolazo que va desde el lugar que es condiciones, hacia la transición que implica la acción.

Una regla de condición - acción se escribe de la siguiente manera:

$$\mu_i \rightarrow q_j \quad (23)$$

Una sentencia de ese tipo se puede representar con un uno en la posición  $ij$  así, la matriz de reglas, va a tener tantos elementos diferentes de cero como reglas existan.

### **3.4 Sistema de Coordinación**

Este trabajo sugiere que la capa de coordinación de la unidad de producción, contenga dos agentes, un agente operador y un agente negociador.

El agente operador tiene como objetivo manejar todas las situaciones que no están contempladas en el supervisor. El agente operador vigila que el proceso permanezca en un estado permitido, cuando no es así, actúa de manera de llevarlo a un estado permitido o en su defecto detenerlo para evitar daños en los equipos o pérdida de material. El agente recibe este nombre debido a que cumple algunas funciones de la que se encargan los operadores en los procesos. Una función muy importante del agente operador es informar al agente negociador la condición del proceso, esto es, si el proceso está funcionando o en falla, cuál es la capacidad de producción disponible y la capacidad ocupada, qué recursos está usando o requiriendo en su funcionamiento y si es posible alcanzar la meta.

El agente negociador debe su nombre a las negociaciones que hace con otras UP que tienen la misma estructura y su principal objetivo es coordinar las acciones que permitan alcanzar las metas del sistema entero. En el proceso de negociación se necesita que dos o más agentes autónomos cooperen entre ellos mostrando opciones, ofreciendo concesiones y aceptando acuerdos que le permitan cumplir con las metas individuales de cada agente así como la meta común de todo el conjunto.

Como muestra Jenninig (2000), el proceso de negociación está compuesto por tres tópicos que se describen a continuación:

1. Protocolo de negociación: es un conjunto de reglas que rigen la interacción entre agentes, el cual cubre los tipos y número de participantes, el estado de la negociación, los eventos que causan cambios de estado en la negociación y las acciones válidas de cada participante en cada estado.
2. El Objeto de la negociación: el conjunto de temas sobre el cual se debe alcanzar un acuerdo, puede ser desde uno hasta cientos de temas. Los participantes pueden aceptar o

rechazar el tema o pueden cambiar el valor del tema en el objeto de la negociación haciendo contrapropuesta o asegurándose de que el acuerdo encaje mejor en su objeto de negociación. Los participantes también pueden alterar dinámicamente la estructura del objeto de negociación. Para la UP algunos ejemplos de objeto de negociación son: los recursos que se necesita y la meta de producción que se desea alcanzar.

3. Modelo de toma de decisiones de agentes: Es el aparato que los participantes utilizan para alcanzar sus objetivos. Las decisiones tomadas son influenciadas por el protocolo de negociación, el objeto de la negociación y por las operaciones que puedan ser ejecutadas por él. En la UP el modelo de toma de decisiones está orquestada por el coordinador.

El proceso de la negociación se hace sobre un espacio de acuerdos cuya dimensión está definida por el rango de los objetos de negociación. La dirección de búsqueda para el acuerdo la definen el modelo de toma de decisiones y el protocolo de negociación.

Dos formas de interacción de agentes están presentes en una comunidad de UP. Cada UP es un sistema multiagente ya que en ella residen e interactúan los agentes operador y negociador, los cuales intercambian información del estado de la UP y las nuevas metas a alcanzar. El sistema que contiene a todas la UP es otro nivel de sistemas multiagente. Los agentes negociadores cooperan entre si para alcanzar la meta común. La figura 3.10 muestra este esquema de sistema multiagentes.

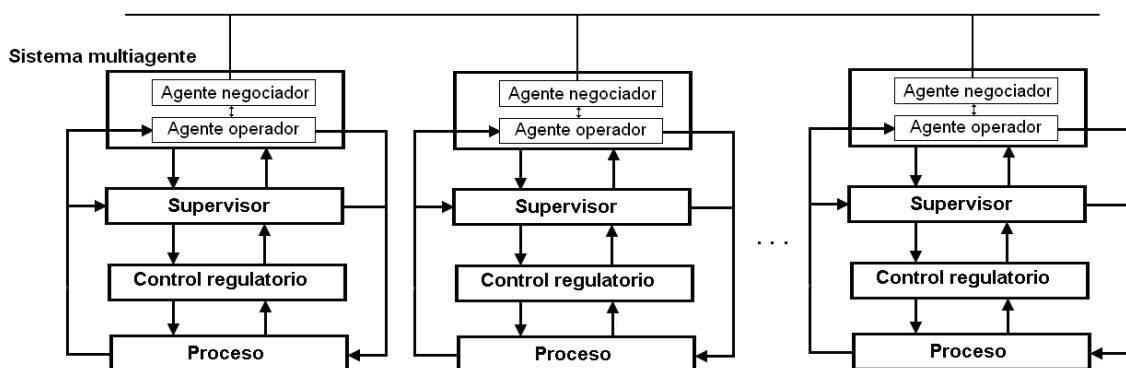


Figura 3.10. Agente reactivo basado en modelo.

### **3.5 Conclusiones del capítulo**

Para el caso de supervisores desarrollados con redes de Petri, en la mayoría de los casos el diseño del supervisor resulta en agregar arcos y lugares al modelo de la planta. El supervisor propuesto está formado por un detector de eventos, un modelo de la planta y una función de control. Las señales provenientes de los sensores son transformadas en eventos que sirven de entrada al modelo de la planta en donde se determina su estado, a su vez la función de control ejecuta las acciones dependiendo del estado del modelo.

El modelo en el supervisor está compuesto por los modelos de los equipos proceso y actuadores que forman parte de la planta. El modelo de lazo abierto de la planta y la función de control forman el modelo a lazo cerrado de la planta. Para el diseño de supervisores se usan dos métodos, el de invarianza de lugares y el de condición acción.



## Capítulo 4. Implementación.

El supervisor que se propone se implementa del modelo a lazo abierto y la función de control, debido a que la ejecución se hace sobre la acción que genera el evento y no directamente sobre el evento; esto ocurre para todos los actuadores descritos en este trabajo. El modelo de lazo cerrado se usa para verificar la conducta del supervisor, encontrar el modelo en redes de Petri de la función de control y analizar el supervisor de tal forma de poder determinar estados indeseados, vivacidad, controlabilidad, trampas y sifones entre otros.

### **4.1 Resumen del método para la implementación del supervisor**

En la implementación del supervisor se sigue método descrito a continuación, el cual requiere de la lista de equipos y la filosofía de operación

1. Se hace el modelo de la planta ( $M_{la}$ ) con la información en la lista de equipos.
2. Se convierten las premisas de la filosofía de control en reglas y/o restricciones según el/los métodos que se vayan a usar para síntesis de supervisor.
3. Se aplica el método para el diseño del supervisor y se obtiene el modelo de lazo cerrado del sistema planta supervisor ( $M_{lc}$ ).
4. Se usan los métodos de simplificación de redes de Petri para eliminar lugares redundantes.
5. Se ejecuta todo el análisis pertinente en el modelo de lazo cerrado para verificar el funcionamiento deseado del supervisor.
6. Se extrae del modelo a lazo cerrado el modelo de lazo abierto y lo que queda es el modelo en red de Petri de la función de control.
7. Se programa la función de control en el equipo de control.

8. Finalmente se hacen la pruebas y puesta en marcha

No es la finalidad de este trabajo obligar a una arquitectura de sistema específica, la idea que se persigue es dar lineamientos que sirva de guía para la implementación del supervisor, la razón principal para esto es la flexibilidad del método propuesto, en el cual el supervisor puede estar distribuido entre diferentes equipos. La figura 4.1 muestra cuatro configuraciones probadas en donde los elementos del supervisor residen en diferentes equipos de control.

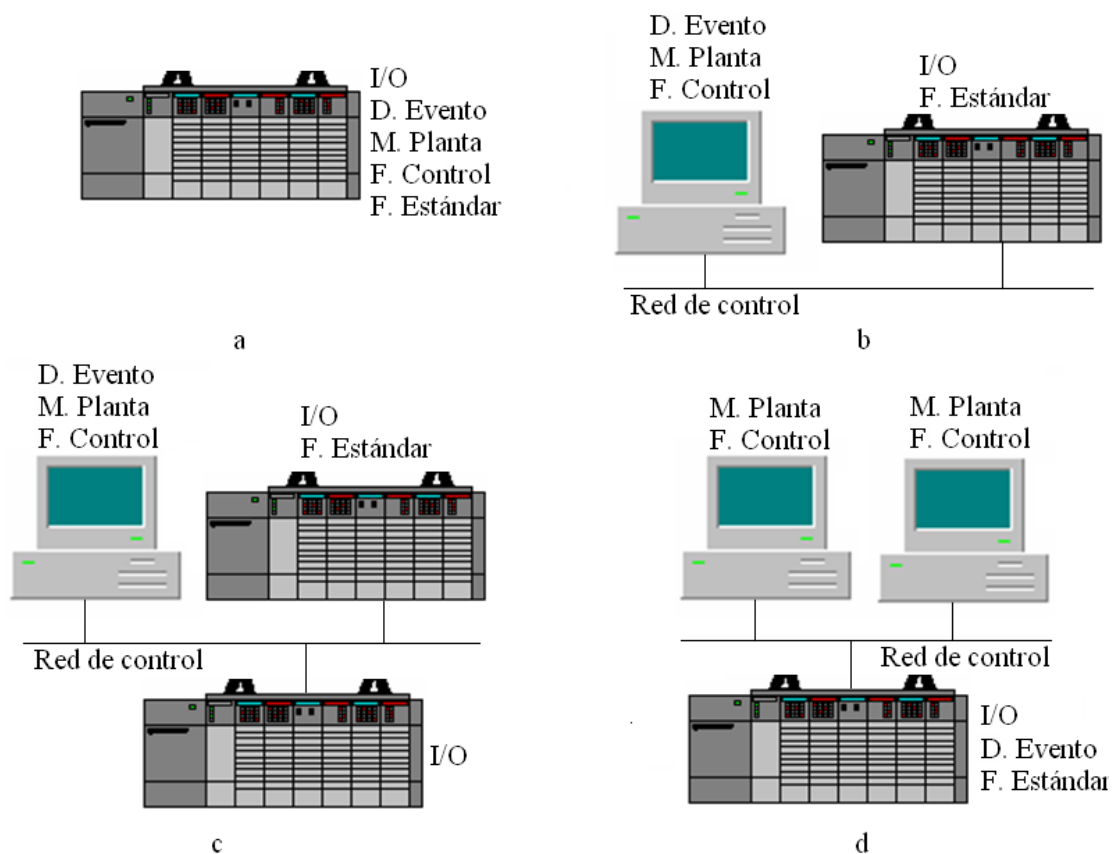


Figura 4.1. Diferentes arquitecturas de sistemas para la implementación del supervisor.

## 4.2 Implementación en un computador personal.

El supervisor se debe programar en un equipo de control como, por ejemplo, un computador personal o controlador lógico programable. Cuando la implementación se hace

en un computador, se usa programación orientada a objetos. En la programación orientada a objetos se modela software haciendo analogías con los objetos en el mundo real (Meyer et al, 1995; Booch et al, 1999). Los objetos pertenecen a una clase. Cada objeto tiene su propia identidad, datos asociados a él por medio del cual se determina su estado y presenta un comportamiento contenidos en un conjunto de métodos.

Para el supervisor el modelo de la planta es un objeto de una clase llama “RDP” (redes de Petri) la cual está formada a su vez por objetos de la clase “Lugar” y objetos de la clase “Transición”. Cada objeto de clase “Transición” tiene asociado un evento. Los eventos provienen de otro objeto de la clase “Detector\_Eventos”; que tiene las funciones de convertir las entradas en eventos y las acciones en salidas.

La conexión del objeto de la clase “Detector\_Eventos” al proceso se puede hacer por diferentes vías, por ejemplo, se puede comunicar con un servidor que contenga los datos del proceso, dentro de mismo computador; esto se puede hacer por medio de socket, el cual es un método para la comunicación entre un programa del cliente y un programa del servidor en una red; en este caso el socket debe apuntar a la dirección del host local. Si el sistema operativo es Windows, se puede usar OPC (OLE for process control, OLE para control de proceso, en español) o DDE (Dynamic Data Exchange, intercambio dinámico de datos, en español).

La comunicación también puede ser directa con una unidad terminal remota (RTU) o un chasis de entrada y salida remoto a través de alguna interfaz del computador como lo pueden ser los puertos RS232, USB o ethernet. En el caso de ethernet también se usan los socket, pero para este caso, se apunta a la dirección del dispositivo dentro de la red donde residan los datos.

Independientemente que la comunicación sea con un servidor de datos o con un chasis de entrada y salida, se necesita algún protocolo de comunicación, que va a depender del fabricante del dispositivo con el que se esté comunicando el supervisor.

La figura 4.2 muestra como se relacionan los elementos del supervisor. En el diagrama de clases se puede ver como atributos o métodos todos los elementos presentes en la definición del supervisor. Uno de los métodos del objeto de la clase RDP es la función de control.

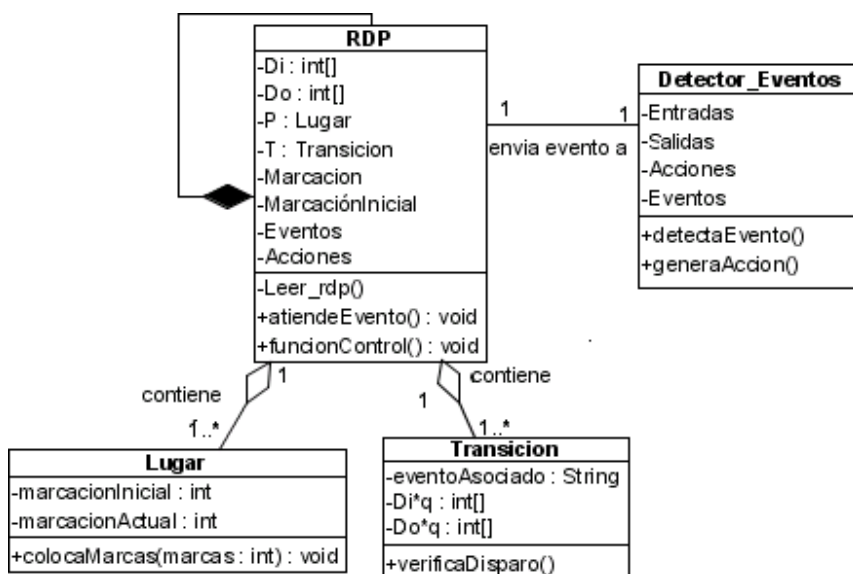


Figura 4.2. Diagrama de clases.

La figura 4.3 muestra un ejemplo de un tanque que es llenado a través de la válvula YV-101 y es controlado por el PLC YC2. El controlador tiene tres entradas,  $I_1$ : YV-101\_NA,  $I_2$ : YV-101\_NC;  $I_3$ : LSL-101; y una salida  $O_1$ : YV-101\_CS.

La válvula de ejemplo es un actuador tipo 2, el cual tiene un modelo como el mostrado en la figura 3.6. El modelo del tanque está formado por un par de lugares y un par de transiciones. La figura 4.4 muestra el modelo de eventos discretos del ejemplo, el cual puede escribirse de la siguiente manera:

$$P = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7\};$$

$$T = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8\};$$

$$D^+ = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}; D^- = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}; \mu_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

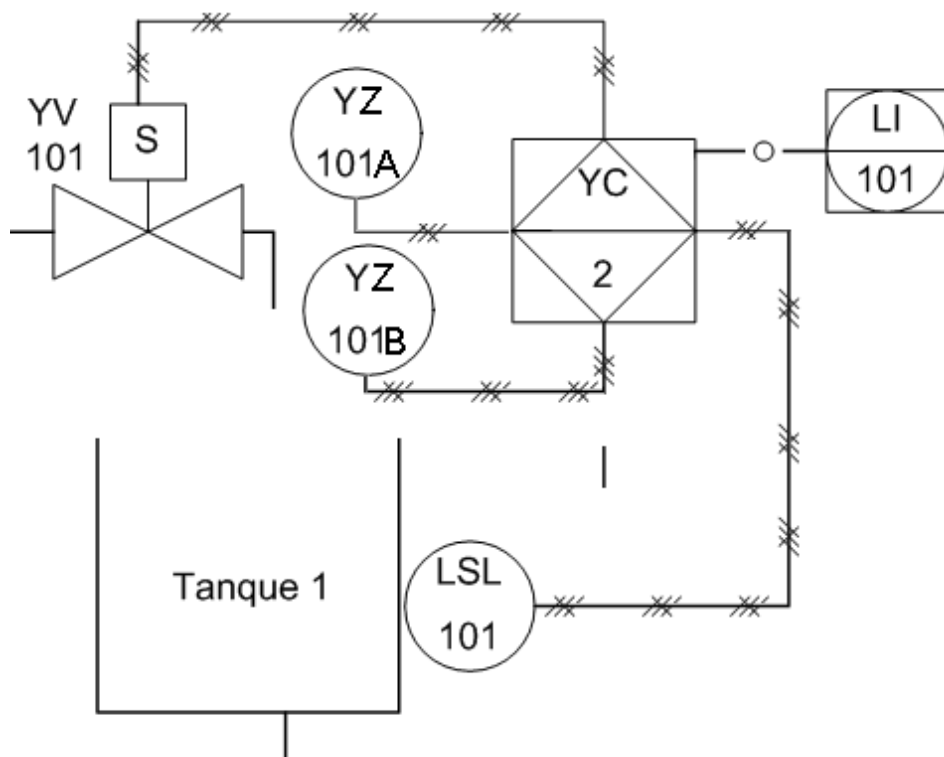
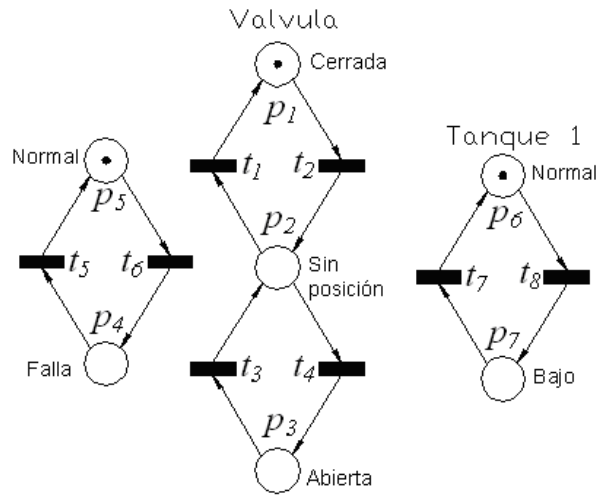


Figura 4.3. Ejemplo de un tanque y una válvula.



**Figura 4.4. Modelo de ejemplo de un tanque y una válvula.**

Cada vez que se detecta un evento se comprueba si ocurre un disparo, eso se hace verificando dos condiciones. Lo primero que se comprueba es, si el evento está asociado a alguna transición; luego se verifica si la transición, a la que pertenece el evento, está habilitada (ecuación 7). Si las dos condiciones son ciertas, ocurre un disparo de la transición asociada al evento. Una vez confirmado el disparo se calcula la nueva marcación (ecuación 8) y se ejecuta la función de control para determinar la acción, en caso de haya alguna.

Las reglas en la función de control están representadas por un conjunto de  $k$  vectores ( $v$ ), donde  $k$  es el número de reglas:

$$v = \begin{bmatrix} v_1 \\ \vdots \\ v_k \end{bmatrix} \quad (24)$$

Cada vector  $v_i$  ( $i=1, \dots, k$ ) tiene tantos elementos como lugares haya en el modelo, es decir:

$$Dim(v_i) = dim(P) \quad (25)$$

Cada vez que se dispara una transición el método función de control de la clase RDP compara los  $k$  vectores de  $v$  con la marcación actual  $\mu$ , para verificar si alguna regla se cumple, en caso de que la regla  $j$ -ésima se cumpla, se ejecuta acción  $j$ -ésima del vector de acciones ( $a$ ):

$$a = \begin{bmatrix} a_1 \\ \vdots \\ a_j \\ \vdots \\ a_k \end{bmatrix} \quad (26)$$

En la tabla 4.1 se muestra un resumen de las acciones que se pueden ejecutar, a que tipo de actuador están asociadas y cual es el valor de la salida para cada acción. La figuras 4.5 muestra la conducta del supervisor ante la llegada de los eventos.

**Tabla 4.1. Acciones y comando para cada actuador en modo automático (MO=1).**

Actuador	Acción	Comando	Salida
Tipo 1	Marcha	AA	CS
	Parada	$\neg$ AA	$\neg$ CS
Tipo 2	Abrir	AA	CS
	Cerrar	$\neg$ AA	$\neg$ CS
Tipo 3	Hacia posición 1	AA, DI	CS, $\neg$ CI
	Hacia posición 2	AA, $\neg$ DI	$\neg$ CS, CI
Tipo 4	Marcha dirección 1	AA, DI	CS, $\neg$ CI
	Marcha dirección 2	AA, $\neg$ DI	$\neg$ CS, CI
	Parada	$\neg$ AA	$\neg$ CS y $\neg$ CI

Para el ejemplo del tanque con la válvula de la figura 4.3, se pueden colocar dos reglas para la función de control:

*Si el tanque está en bajo abre la válvula*

*Si el tanque está normal cierra la válvula*

Los vectores de reglas y el vector de acciones se muestran a continuación:

$$v = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$a = \begin{bmatrix} YV-101\_AA \\ \neg YV-101\_AA \end{bmatrix}$$

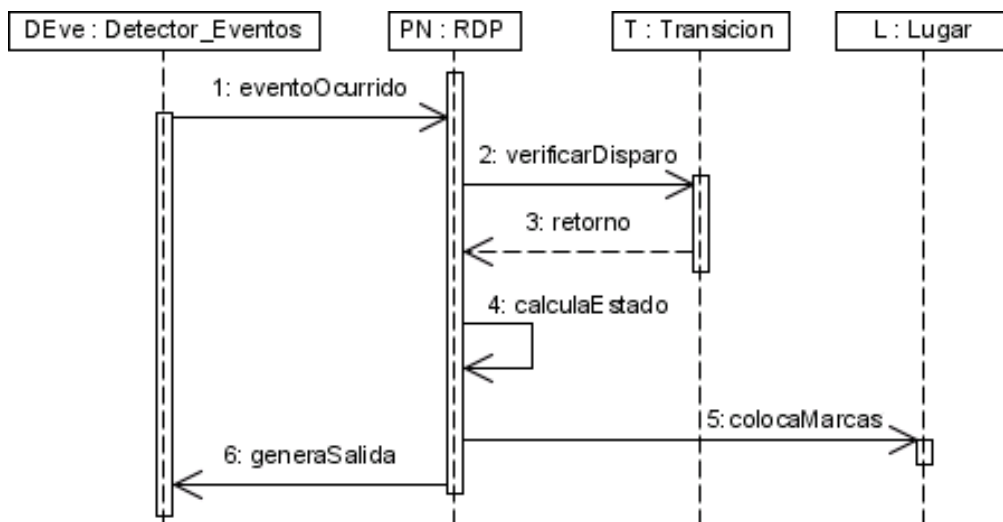


Figura 4.5. Diagrama de secuencias.

#### 4.2.1 Simulación

Para probar el supervisor, antes de la puesta en marcha final, se hace una simulación, la cual consiste en reemplazar la planta con un modelo de la misma, que tiene la forma descrita por la ecuación (9) de la definición 2.14, para procesos con dinámicas híbridas. Durante la simulación los eventos que llegan al supervisor, provienen del modelo de la planta.

La simulación de los actuadores, los cuales tienen dinámicas de eventos discretos, reproduce la conducta del actuador. Para el caso de un actuador tipo 1, por ejemplo, cuando se ejecuta la acción de marcha en la función de control y están dadas todas las condiciones, el supervisor coloca la variable “comando de salida” (CS) en verdadero. El actuador

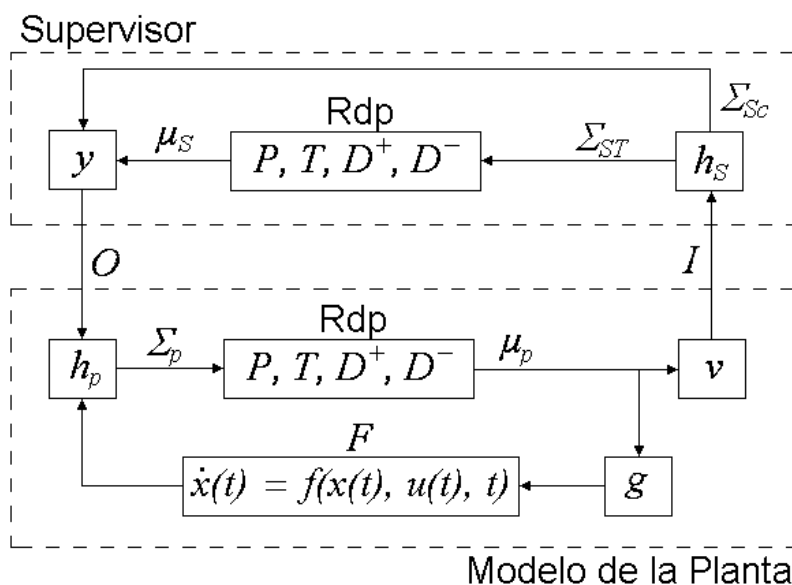


responde colocando la “entrada 1” (NA) a verdadero. Un simulador para los actuadores tipo 1 coloca la “entrada 1” (NA) en verdadera cuando la variable “comando de salida” (CS) es verdadera. La siguiente tabla muestra los valores de entradas en función de los valores de las salidas para los actuadores descritos. Cualquier combinación que no esté en la tabla produce un error en el actuador correspondiente.

**Tabla 4.2. Valores de las entradas en función de la salida para la simulación de los actuadores.**

Actuador	Salida	Entrada
Tipo 1	CS	NA
	$\neg$ CS	$\neg$ NA
Tipo 2	CS	NA, $\neg$ NC
	$\neg$ CS	$\neg$ NA, NC
Tipo 3	CS	NA, $\neg$ NC
	CI	$\neg$ NA, NC
Tipo 4	CS	NA
	CI	NA
	$\neg$ CS y $\neg$ CI	$\neg$ NA

El modelo de la planta se vale de las funciones  $h_p$ ,  $v$  y la red de Petri ( $P$ ,  $T$ ,  $D^+$ ,  $D^-$ ,  $\mu$ ) para suministrar la respuesta al controlador, como se puede observar en la figura 4.6.



**Figura 4.6. Sistema Planta - Supervisor.**

Para simular la dinámica continua de la planta, se transforma cada las ecuación diferencial (2) a una ecuación en diferencia aplicando el algoritmo de Euler hacia delante; así, la ecuación obtenida es:

$$x(t + \Delta t) = x(t) + \dot{x}(t)\Delta t \quad (27)$$

Esta ecuación se puede colocar en un método que se ejecute cíclicamente con un periodo de tiempo igual a  $(\Delta t)$ . Se supone que al ir de un conjunto de ecuaciones en diferencia a otro, según la función  $g$  de la definición 2.14, el valor de  $x$  se conserva, esto es  $x(t_{f_{\mu k}}) = x(t_{i_{\mu k+1}})$ , donde  $x(t_{f_{\mu k}})$  es el valor final del vector de estado proporcionado por las ecuaciones en diferencia en la marcación, antes del disparo de la transición y  $x(t_{i_{\mu k+1}})$  es el valor inicial del vector de estado proporcionado a las ecuaciones en diferencia en la marcación, después del disparo de la transición.

Para el ejemplo de la figura 4.3, el modelo de la planta teniendo en cuenta que, el evento  $e_i$  está asociado a la transición  $t_i$ ,  $r$  es la altura del sensor en el tanque,  $w$  es el flujo de entrada y  $v$  es el flujo de salida del tanque y el área del tanque es igual a uno, queda como sigue:

<i>Función</i>	<i>Si</i>	<i>Entonces</i>
$h_p: x \times O \rightarrow \Sigma_p$	$YV-101\_CS;$ $\neg YV-101\_CS;$ $x > r;$ $x \leq r;$	$e_2$ para $(\mu_1=1)$ y $e_4$ para $(\mu_2=1);$ $e_3$ para $(\mu_3=1)$ y $e_1$ para $(\mu_2=1);$ $e_7;$ $e_8;$
$g: \mu \rightarrow F$	$\mu_1=1$ $\mu_3=1$	$\dot{x} = -v$ ( $x_{k+1}=x_k-v\Delta t$ ) $\dot{x} = w - v$ ( $x_{k+1}=x_k+(w-v)\Delta t$ )
$v: \mu \rightarrow I$	$\mu_1=1$ $\mu_3=1$ $\mu_6=1$ $\mu_7=1$	$\neg YV-101\_NA, YV-101\_NC$ $YV-101\_NA, \neg YV-101\_NA$ $LSL-101;$ $\neg LSL-101$

### **4.3 Implementación en un controlador lógico programable.**

En la simulación e implementación de supervisores para sistemas de eventos discretos son muy usados los controladores lógicos programables (PLC). Cuando se trata de llevar una red de Petri a un controlador lógico programable es muy común utilizar gráficos de función secuencial (Brinksma et al., 2002; David, 1995; Hellgren, 2002); sin embargo, no todos los controladores lógicos programables cuentan con gráficos de función secuencial, por esta razón se presenta una propuesta para implementar una red de Petri en un controlador lógico programable usando diagrama escalera. El diagrama escalera se asemeja a los diagramas elementales usados para planos eléctricos y a cada línea del programa escalera se le denomina peldaño asemejando a las escaleras que utilizan los bomberos. La mayoría de los controladores lógicos programables que existen en el mercado pueden ser programados usando diagramas escalera.

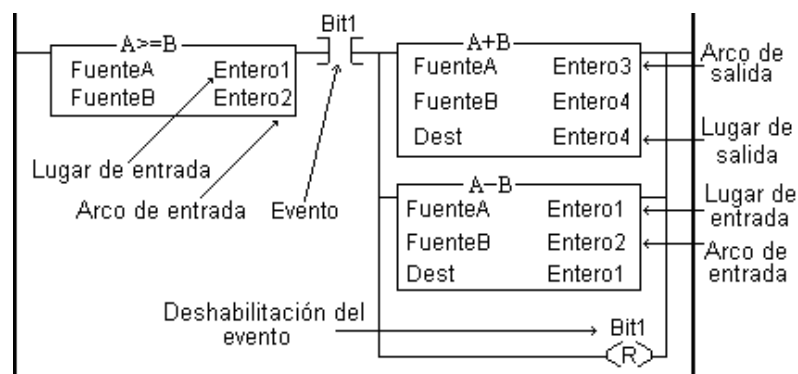
#### **4.3.1 Implementación del modelo de la planta**

Para la implementación el modelo de la planta como una red de Petri en el controlador lógico programable se asignó una relación entre la red de Petri y la memoria del PLC, tal como muestra la tabla 4.3.

**Tabla 4.3. Relación entre la red de Petri y la memoria del PLC.**

<i>Elemento de la RDP</i>	<i>Memoria del PLC</i>
<i>Lugar</i>	<i>Variable entera cuyo valor indica la cantidad de marcas actual en el lugar</i>
<i>Arco</i>	<i>Variable entera cuyo valor representa el peso del arco</i>
<i>Transición</i>	<i>Peldaño del programa que contiene instrucciones de comparación, sumas, restas, e instrucciones de bit</i>
<i>Evento</i>	<i>El evento no tiene representación en la memoria del controlador, sin embargo, hay una variable binaria asociada a cada transición que indica que el evento ha ocurrido</i>

Con esta asignación una transición  $t_j$  corresponde a un peldaño del programa escalera, el cual tendrá como instrucciones de entrada tantas instrucciones de comparación del tipo “mayor o igual que” ( $\geq$ ) como lugares de entrada tenga la transición, además una instrucción “contacto normalmente abierto” (-) [-], la cual estará asignado el bit que indica si el evento asociado con la transición ocurrió. Las instrucciones de salida del peldaño del programa escalera que representa la transición  $t_j$  incluirán tantas instrucciones de suma como lugares de salida tenga la transición, tantas instrucciones de resta como lugares de entrada tenga la transición, una instrucción de restablecimiento  $(-R-)$  para borrar el evento que disparó la transición. Las instrucciones de suma agregan tantas marcas al lugar de salida como peso tiene el arco que lo une a la transición y las instrucciones de resta eliminan tantas marcas del lugar de entrada como peso tenga el arco que lo une a la transición. La figura 4.7 ilustra una transición en lógica escalera.



**Figura 4.7. Diagrama escalera de una transición con un lugar de entrada y un lugar de salida.**

Además se debe incluir en el programa escalera una rutina para la detección de eventos. En esta rutina se asocian los elementos del conjunto  $I$  con los elementos del conjunto  $\Sigma_S$ , tal como indica la función  $h_S$ . El programa también debe contar con una rutina en donde las salidas del conjunto  $O$  sean asignadas según la función  $y$ . En cada controlador puede haber más de una red de Petri. Todos los elementos usados para implementar el modelo de eventos discreto en el controlador lógico programable (diagrama escalera, instrucciones de suma, resta, comparación, contacto normalmente abierto, bobina de restablecer, bobina de colocar y temporizadores) están contemplados en la norma IEC 61131-3 (ISO, 2001).

Si la red de Petri del modelo de la planta es segura (*I*-acotada) y ordinaria (todos los pesos de los arcos son iguales a uno), cada marcación de los lugares puede representarse con un bit, en vez de usar un entero, y se reemplaza la instrucción de comparación con una instrucción de contacto abierto (-) [-], esto ahorra tiempo de ejecución, memoria de programa y memoria de datos. En la figura 4.8 puede verse la representación escalera de un modelo seguro.

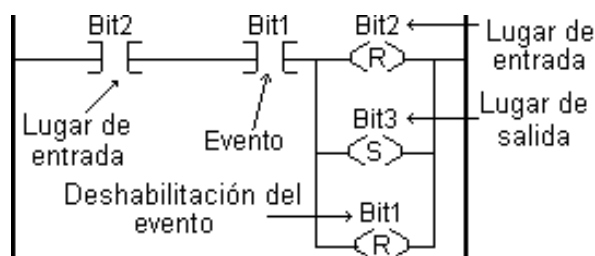


Figura 4.8. Diagrama escalera de modelos en redes de Petri segura y ordinaria.

### 4.3.2 Implementación de la función de control

La función de control es una rutina, en la cual cada peldaño corresponde a una regla. Cada regla tiene como instrucciones de entrada la marcación determinada, y como instrucciones de salida las acciones que son provocadas por la dicha marcación. La figura 4.9 muestra la estructura de una regla en la función de control expresada en lenguaje escalera, cual tiene *j* condiciones y ejecuta *k* acciones.

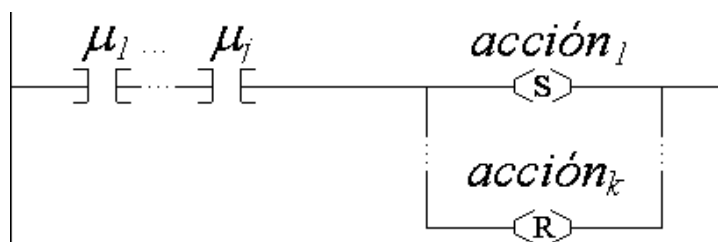


Figura 4.9. Forma de las reglas de la función de control en lenguaje escalera.

Como se mencionó antes, el supervisor está formado por el modelo de la planta y la función de control, pero para que pueda funcionar de forma adecuada, también debe existir la función estándar que se mencionó en el capítulo 3. No es necesario que el supervisor y la función estándar estén en el mismo equipo de control.

Otro aspecto a tomar en cuenta es la estrecha relación que existe entre el modelo en redes de Petri con el conjunto de entrada y salidas, por ejemplo para el actuador tipo 1 se puede verificar la siguiente relación

$$\mu(1) = -NA \quad (28)$$

$$\mu(2) = NA \quad (29)$$

Si se conocen todas las relaciones entre las marcaciones de la red de Petri (modelos de actuadores y equipos de proceso) y las entradas del supervisor, entonces, opcionalmente se puede prescindir del modelo de la planta.

#### **4.4 Simulación de la dinámicas continuas en un PLC**

El procedimiento para la simulación de las dinámicas continuas es el mismo usado para la simulación en computador personal, se transforma la ecuación diferencial (2) a una ecuación en diferencia aplicando el algoritmo de Euler hacia delante para obtener la ecuación 27.

La implementación de esta ecuación en un controlador lógico programable es bastante sencilla. Un aspecto importante que hay que tomar en consideración es la variación de tiempo ( $\Delta t$ ), ésta debe ser la misma en cada muestra, por esta razón es recomendable, siempre que se pueda, utilizar una interrupción de tiempo que ejecute la rutina en donde se va a colocar la ecuación en diferencia, en vez de usar temporizadores que estén dentro del ciclo de barrido del controlador lógico programable, ya que el tiempo de ejecución de cada ciclo no es constante y esto hace que el tiempo proporcionado por el temporizador tenga pequeñas diferencias entre un ciclo y el siguiente. También se supone, de la misma forma que en la implementación en el pc, que al ir de un conjunto de ecuaciones en diferencia a otro, según la función  $g$  de la definición 2.14, el valor de  $x$  se conserva.

#### ***4.5 Resumen del capítulo***

El modelo a lazo cerrado de la planta se usa para verificar la conducta del supervisor. La implementación se hace llevando el modelo de lazo abierto y la función de control a un equipo de control. Los equipos de control pueden ser controladores lógicos programables, computadores personales, sistemas de control distribuidos entre otros. Usando el modelo de la planta y el modelo del supervisor se puede simular el proceso, esto permite comprobar la conducta del sistema.

## Capítulo 5. Caso de estudio 1

El siguiente ejemplo (Bak and Izadi-Zamanabadi, 2004) sirve para ilustrar como se puede modelar, diseñar y simular un sistema híbrido con supervisor. En la figura 5.1 se muestra el sistema conformado por dos tanques. Hay una válvula de dos vías que dirige el fluido a los tanques de tal forma que siempre está llenando a uno y sólo uno de los dos tanques en un momento determinado. La cantidad de fluido que se dirige a cada tanque es  $w$ . El nivel del tanque 1 es  $x_1$  y el nivel del tanque 2 es  $x_2$ , el flujo de salida del tanque 1 es  $v_1$  y el flujo de salida del tanque 2 es  $v_2$ , supondremos que los flujos de salida son constantes (esto se puede lograr si la salida de cada tanque está conectado a una bomba de desplazamiento positivo). Además, hay dos sensores, uno en cada tanque que indican los niveles  $r_1$  y  $r_2$  de los tanques 1 y 2 respectivamente. El objetivo del control es lograr  $x_1 > r_1$  y  $x_2 > r_2$ . Con este objetivo, una solución estable existe siempre que  $w = v_1 + v_2$  y con condiciones iniciales  $x_1(t_0) \geq r_1$  y  $x_2(t_0) \geq r_2$ , de otra manera se desbordaría alguno de los tanques o el sistema se puede quedar sin fluido. En este caso de estudio se hará el modelo de la planta, la cual es un sistema híbrido, se calculará el supervisor y se hará una simulación del sistema planta supervisor.

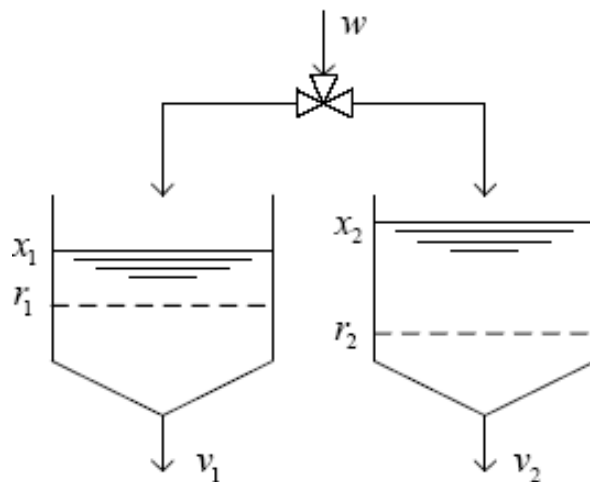


Figura 5.1. Sistema híbrido de dos tanques.



### 5.1 Modelo de la planta.

En el modelo detallado la válvula se modela como un actuador tipo 2, tal como se muestra en la figura 5.2, sin embargo, en el modelo de modos para este ejemplo, sólo importa saber cuál es el tanque que se está llenando, así que el modelo se puede simplificar al modelo que se observa en la figura 5.3. Cada tanque se modela con dos lugares y dos transiciones. Los lugares indican que el tanque está en estado normal ( $x_i \geq r_i$ ) o que el tanque está en estado bajo ( $x_i < r_i$ ). Cada tanque pasa de estado normal a estado bajo cuando ocurre el evento  $x_i < r_i$  con  $i = 1, 2$  y de estado bajo a estado normal cuando  $x_i \geq r_i$ . También existen dos eventos asociados a la válvula (llenar tanque 1 o llenar tanque 2). El modelo de la válvula, además, contiene un par de lugares, una marca dentro de cada lugar indica la dirección de llenado de la válvula.

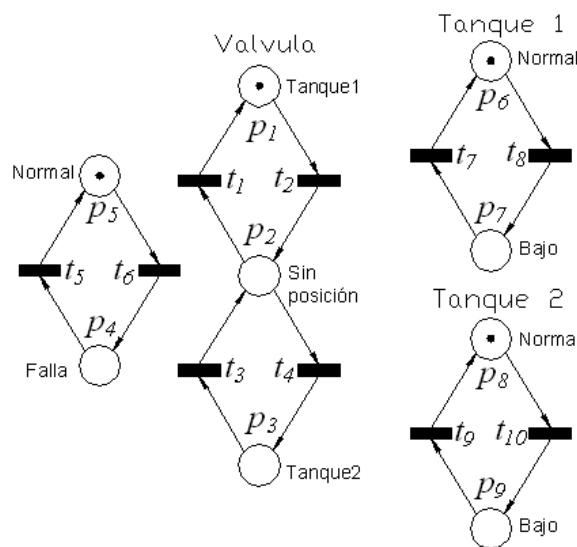


Figura 5.2. Modelo detallado del sistema dos tanques, lazo abierto.

### 5.2 Diseño del supervisor.

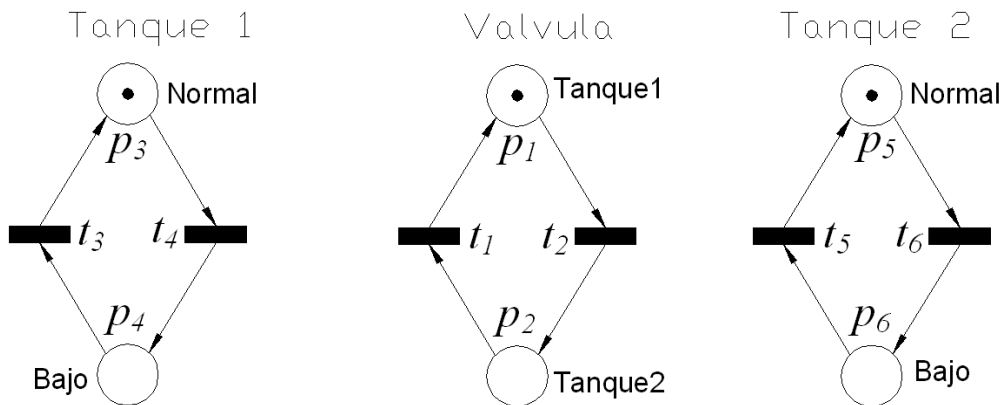
La primera restricción tiene como objetivo evitar que el tanque 1 se llene cuando el estado del tanque 1 está en normal, esto se puede escribir como

$$q_1 + \mu_3 \leq 1 \quad (30)$$

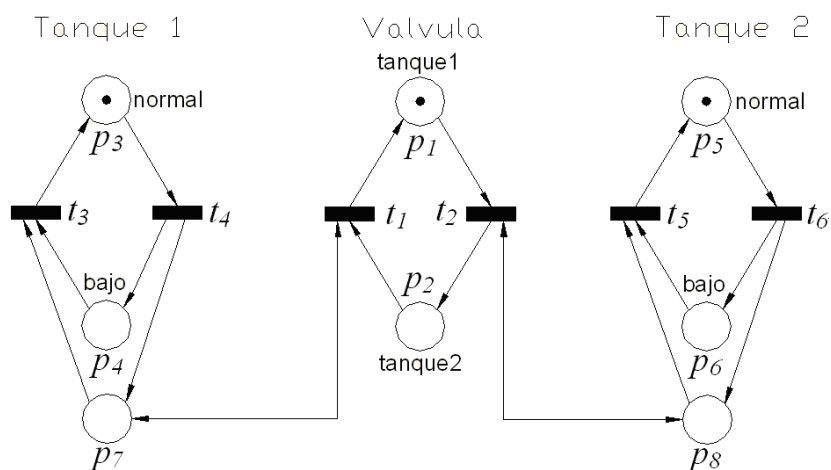
donde  $\mu_i$  es la marcación del lugar  $i$ . La restricción que mantiene el control sobre el tanque 2 se obtiene de manera similar de la restricción (30).

$$Q_2 + \mu_5 \leq 1 \quad (31)$$

Las dos restricciones son admisibles (Moody and Antsaklis, 1998) y por lo tanto se puede implementar el supervisor. La red de Petri agregando los lugares y arcos generados por las restricciones se puede observar en la figura 5.4.



**Figura 5.3. Modelo de modos del sistema de dos tanques, lazo abierto.**



**Figura 5.4. Modelo del sistema de dos tanques, lazo cerrado usando invarianza de lugares.**

El modelo de red de Petri de la figura 5.4 se puede reducir al modelo de la figura 5.5 (Murata, 1989). Como se puede observar en esta figura, las restricciones (30) y (31) se pueden escribir como reglas condición acción:

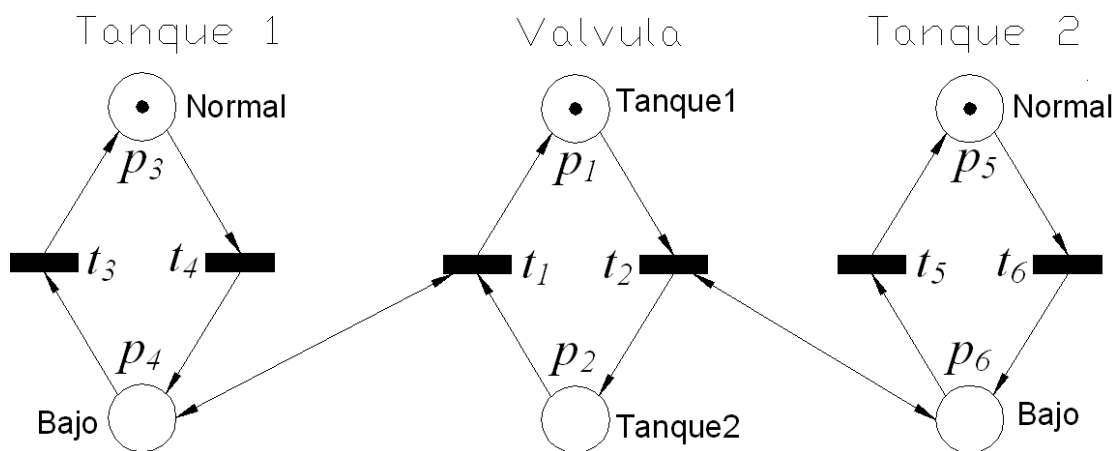
$$\mu_4 \rightarrow q_1 \quad (32)$$

$$\mu_6 \rightarrow q_2 \quad (33)$$

Estas reglas pueden interpretarse como las siguientes premisas respectivamente:

*Si el tanque 1 está vacío entonces llene el tanque 1*

*Si el tanque 2 está vacío entonces llene el tanque 2*



**Figura 5.5. Modelo a lazo cerrado de sistema de dos tanques usando supervisores de condición acción.**

La función de control en lógica escalera para el modelo de la figura 5.5 se muestra en la figura 5.6.

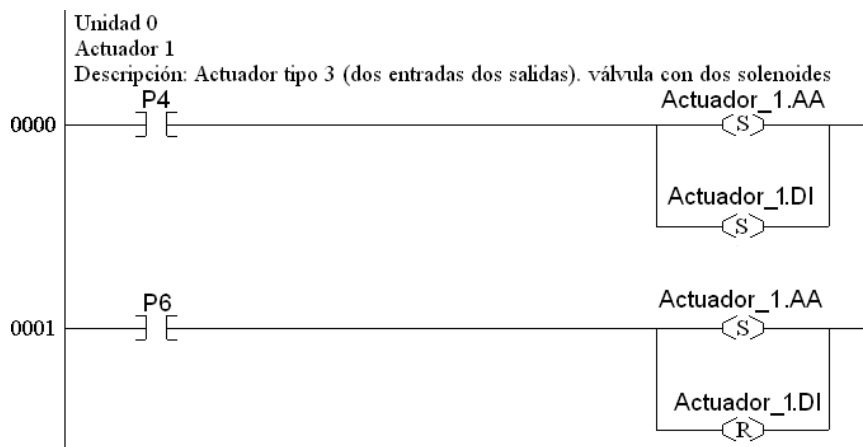


Figura 5.6.- Función de control en diagrama escalera para el sistema de dos tanques

### 5.3 Dinámicas continuas

El nivel  $x_i$  del tanque  $i$  ( $i = 1, 2$ ), suponiendo el área de la base igual a uno, viene dado por la ecuación (34), cuando el tanque se está llenando y por la ecuación (35) cuando el tanque se está vaciando

$$\dot{x}_i = w - v_i \quad (34)$$

$$\dot{x}_i = -v_i \quad (35)$$

Al combinar las ecuaciones (34) y (35) con la ecuación (27) se obtiene el siguiente conjunto de ecuaciones:

$$x_1(t + \Delta t) = x_1(t) + (w - v_1) \Delta t \quad (36)$$

$$x_1(t + \Delta t) = x_1(t) - v_1 \Delta t \quad (37)$$

$$x_2(t + \Delta t) = x_2(t) + (w - v_2) \Delta t \quad (38)$$

$$x_2(t + \Delta t) = x_2(t) - v_2 \Delta t \quad (39)$$

Las ecuaciones (36) y (37) corresponden a las ecuaciones del tanque 1, llenándose y vaciándose, respectivamente, y las ecuaciones (38) y (39) corresponden a las ecuaciones del tanque 2, llenándose y vaciándose, respectivamente.

#### 5.4 Definición del modelo

En las ecuaciones (40), (41) y (42) se pueden observar la matriz de incidencia de salida, la matriz de incidencia de entrada y el vector marcación inicial, respectivamente, para la red de Petri de la figura 5.5, la cuales son obtenidas del modelo de modos. Las tablas 5.1 y 5.2 contienen los conjuntos y las funciones que forman parte del modelo que describen al sistema de dos tanques según la definición 3.1.

$$D^+ = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (40)$$

$$D^- = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (41)$$

$$\mu_0 = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad (42)$$

**Tabla 5.1. Conjuntos del sistema**

<i>Conjuntos</i>	<i>Elementos</i>
$P$	$p_1, p_2, p_3, p_4, p_5, p_6$
$T$	$t_1, t_2, t_3, t_4, t_5, t_6$
$F$	$x_1(t + \Delta t) = x_1(t) + (w - v_1) \Delta t,$ $x_1(t + \Delta t) = x_1(t) - v_1 \Delta t,$ $x_2(t + \Delta t) = x_2(t) + (w - v_2) \Delta t,$ $x_2(t + \Delta t) = x_2(t) - v_2 \Delta t$
$\Sigma_S$	$e_1, e_2, e_3, e_4, e_5, e_6$
$I$	$i_1(x_1 \geq r_1), i_2(x_2 \geq r_2), i_3(NA), i_4(NC)$
$O$	$o_1(CS)$

**Tabla 5.2. Funciones del sistema**

<i>Función</i>	<i>si</i>	<i>Entonces</i>
$g : \mu_p \rightarrow F$	$\mu_1 = 1$	$x_1(t + \Delta t) = x_1(t) + (w - v_1) \Delta t$
	$\mu_1 = 0$	$x_1(t + \Delta t) = x_1(t) - v_1 \Delta t$
	$\mu_2 = 1$	$x_2(t + \Delta t) = x_2(t) + (w - v_2) \Delta t$
	$\mu_2 = 0$	$x_2(t + \Delta t) = x_2(t) - v_2 \Delta t$
$h : I \rightarrow \Sigma_S$	$i_3$	$e_1$
	$i_4$	$e_2$
	$i_1$	$e_3$
	$\neg i_1$	$e_4$
	$i_2$	$e_5$
	$\neg i_2$	$e_6$
$y : \mu_S \times \Sigma_{Sc} \rightarrow O$	$\mu_1 = 1$	$o_1 = 1, \text{ hacia tanque 1}$
	$\mu_2 = 1$	$o_1 = 0, \text{ hacia tanque 2}$

Los resultados de la simulación tomando en cuenta los siguientes valores de los parámetros  $x_{10} = 1,8$  mts,  $x_{20} = 1,7$  mts,  $w = 0,05$  mts<sup>3</sup>/s,  $v_1 = 0,03$  mts<sup>3</sup>/s,  $v_2 = 0,02$  mts<sup>3</sup>/s,  $r_1 = 1,6$  mts,  $r_2 = 1,6$  mts, área de la base del tanque =  $1 \text{ m}^2$ , se pueden ver en la figura 5.7.

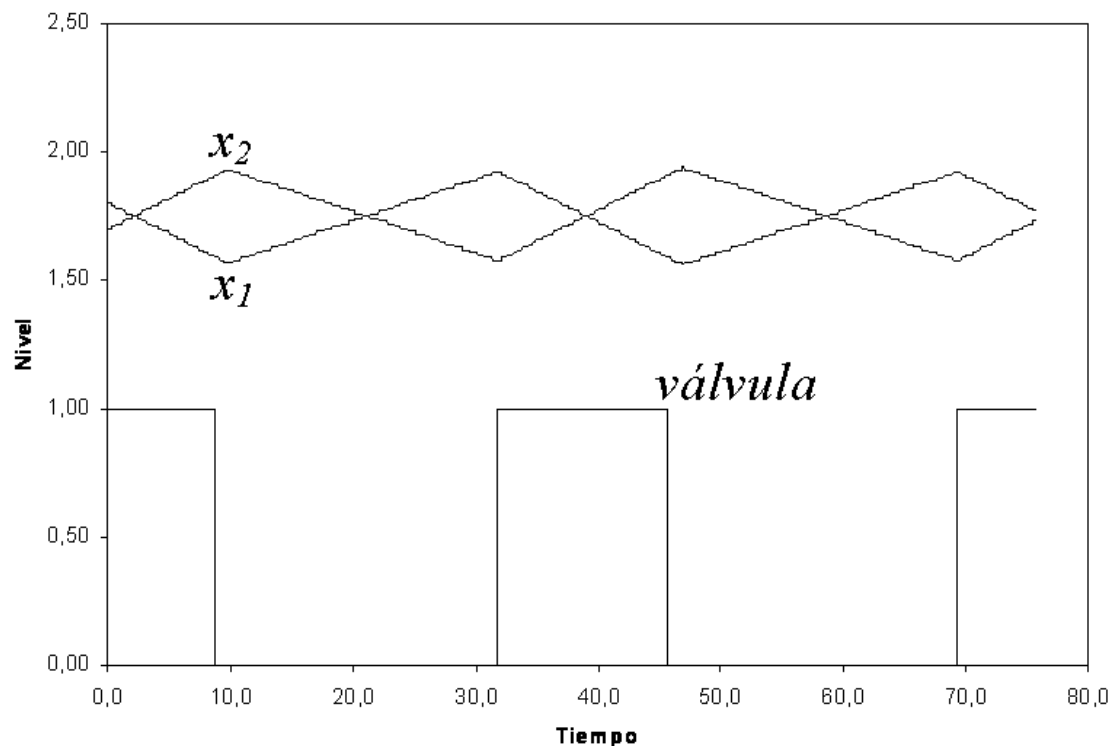
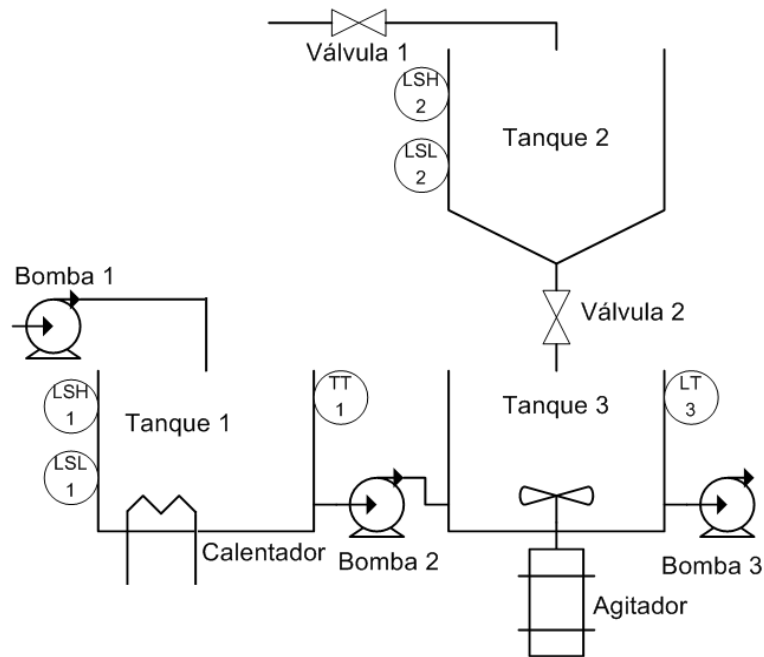


Figura 5.7. Graficas de los niveles del sistema de dos tanques. Un valor de 0 en la válvula indica que se llena el tanque 1 y un valor de 1 indica que se llena el tanque 2.

## 5.5 Caso de estudio 2

En la figura 5.8 se muestra un sistema conformado por tres tanques. Hay dos válvulas, una para llenar el tanque dos y la otra para vaciarlo. Además hay tres bombas: la primera llena el tanque uno, la segunda vacía el tanque 1 y llena el tanque 3 y la tercera vacía el tanque 3. El sistema, además, cuenta con un agitador en el tanque 3 y un calentador en el tanque 1. El tanque 1 y el tanque 2 tienen interruptores de nivel alto y nivel bajo, mientras que el tanque 3 tiene un transmisor analógico de nivel. Además, el tanque 1 tiene un transmisor analógico de temperatura. El objetivo del proceso es calentar el líquido del tanque 1 a  $70 \text{ }^\circ\text{C}$  para luego mezclar con el líquido del tanque 2 en el tanque 3.



**Figura 5.8. Sistema de tres tanques.**

### 5.5.1 Modelo de la planta

Las válvulas, las bombas, el agitador y el calentador se modelan con dos lugares y dos transiciones, donde la marcación indica si el elemento está activado o no. Cada tanque se modela con tres lugares y 4 transiciones, la marcación de cada tanque indica si el tanque está en nivel bajo, medio o lleno. La figura 5.9 muestra la red de Petri del sistema.

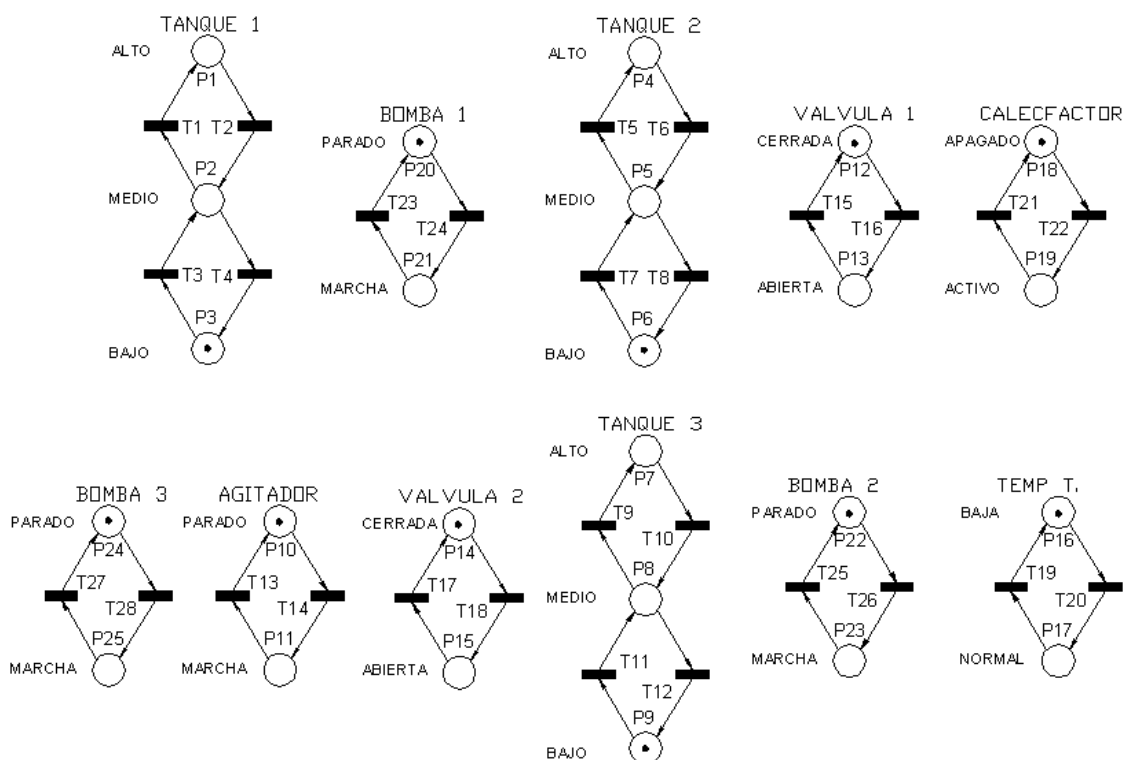
### 5.5.2 Supervisor

El supervisor se diseñó bajo las siguientes reglas:

1. La bomba 1 se pone en marcha cuando el tanque 1 está en nivel bajo y apaga cuando el tanque 1 está en nivel alto.
2. La bomba 2 se pone en marcha cuando la temperatura del líquido en el tanque 1 llega a 70 °C y el tanque 3 está vacío y se apaga cuando el tanque 3 llega a nivel medio.
3. La bomba 3 se pone en marcha 3 segundos después que el tanque tres llega a nivel alto y se apaga cuando se vacía el tanque 3.



4. La válvula 1 abre cuando el nivel del tanque 2 está en bajo y cierra cuando el nivel del tanque 2 está alto.
5. La válvula 2 abre cuando el tanque 3 llega a nivel medio y la bomba 3 está parada y cierra cuando el tanque 3 llega a nivel alto.
6. El agitador se pone en marcha cuando la válvula 2 está abierta y se apaga cuando la válvula 2 está cerrada.
7. El calefactor se activa cuando la temperatura es menor o igual a 70 °C y se apaga cuando la temperatura es mayor o igual a los 73 °C.



**Figura 5.9. Red de Petri del sistema de 3 tanques.**

La figura 5.10 muestra la red de Petri de la planta con el supervisor que se deriva directamente de las reglas condición - acción.

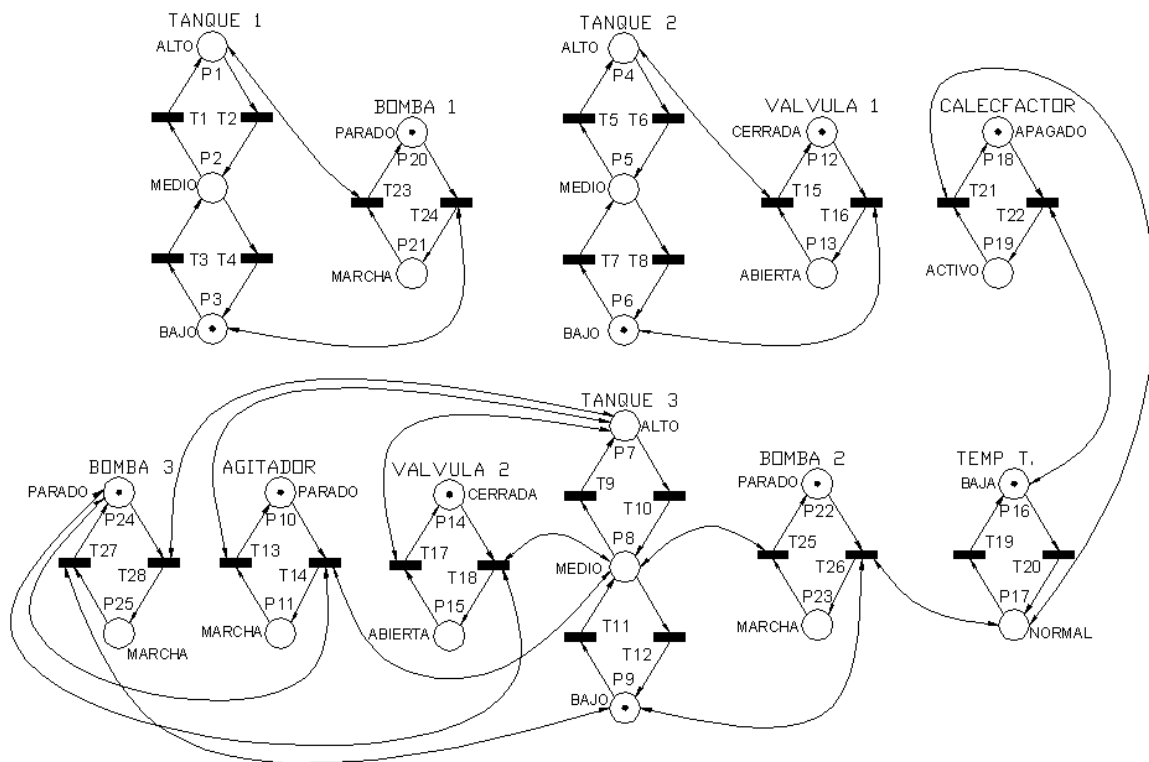


Figura 5.10. Red de Petri del sistema de tres tanques con supervisor.

### 5.5.3 Dinámicas continuas

El nivel  $x_i$  del tanque  $i$  ( $i = 1, 2, 3$ ) viene dado por la ecuación (43),

$$\dot{x}_i = w_i - v_i \quad (43)$$

donde  $w$  es el flujo de entrada del tanque y  $v$  es el flujo de salida del tanque, se supone el área de la base del tanque igual a uno. La ecuación que rige el cambio de temperatura del líquido está dada por:

$$\dot{t} = 1 - 0,01t \quad (44)$$

Donde  $t$  es la temperatura.

La gráfica de la simulación correspondiente a los niveles de los tanques se muestra en la figura 5.11.

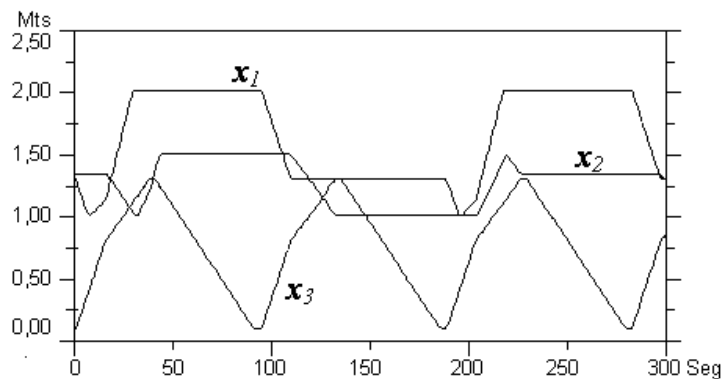


Figura 5.11. Gráficas de los niveles del sistema de tres tanques.

## 5.6 Caso de estudio 3

El caso de estudio se tomó de una fábrica de galletas donde se implementó un supervisor como el propuesto, la figura 5.12 muestra el diagrama de proceso del sistema que tiene como objetivo transportar harina desde un silo hasta los recipientes que alimentan las líneas de fabricación. Para su traslado se deja caer la harina en una tubería a la que se inyecta aire con un soplador, la harina es trasladada a una tolva que alimenta a un cernidor y de allí a un ciclón para luego ser transportada por otro soplador hasta las cajas de harina.

### 5.6.1 Arquitectura de Sistema

Para manejar el sistema se cuenta con un controlador lógico programable como equipo de control, éste se comunica por medio de una red serial (ethernet) a una interfaz gráfica de operación, la cual cuenta con las siguientes funciones:

1. Seleccionar las cajas de harinas que van a ser llenadas.
2. Mostrar el estado de los equipos que forman parte del sistema.
3. Mostrar las alarmas.

Los motores de los sopladores M30 y M31 y el ciclón M50 son manejados con arrancadores suaves, el resto de los motores son manejados con arrancadores electromecánicos a los que se les adiciona un circuito de protección térmica con capacidad

de entrada/salida. Tanto los arrancadores suaves como los circuitos de protección térmica tienen capacidad para conectarse a una red de campo (DeviceNet). El resto de las señales de campo tales como sensores de presión, nivel, solenoides de las válvulas, botones para la operación local de los motores, entre otras, llegan por medio de tarjetas de entradas y salidas montadas en el bastidor local del controlador lógico programable. La figura 5.13 muestra la arquitectura de control del sistema de transporte de harina.

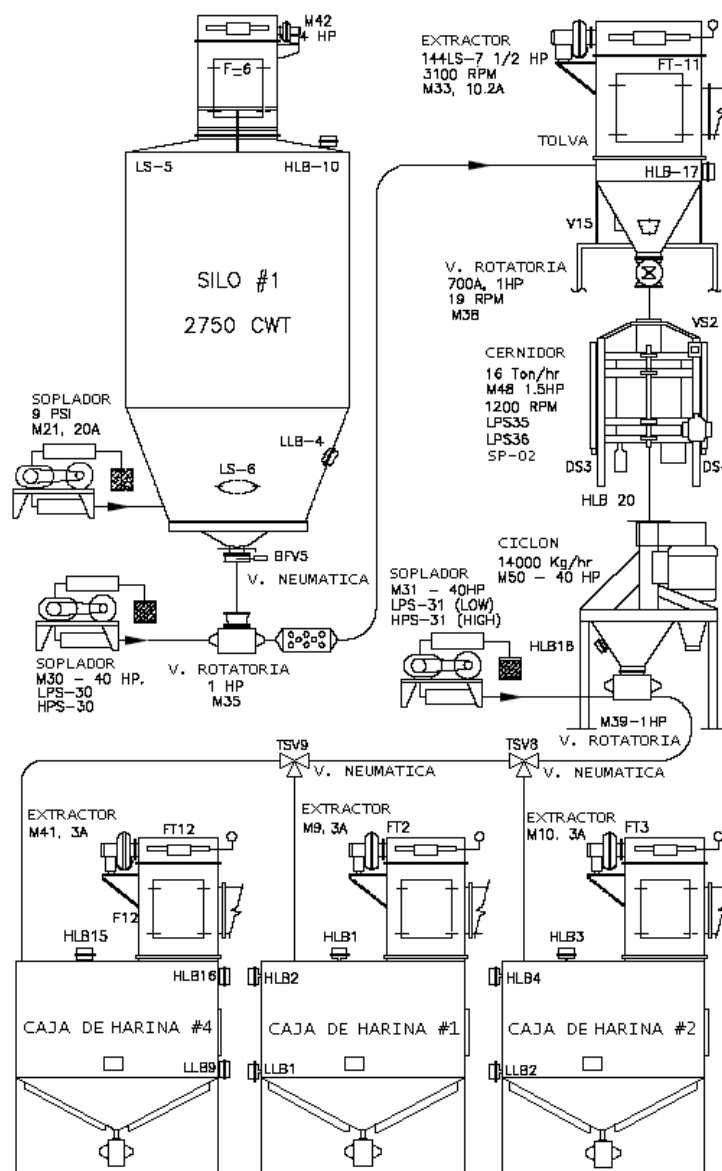
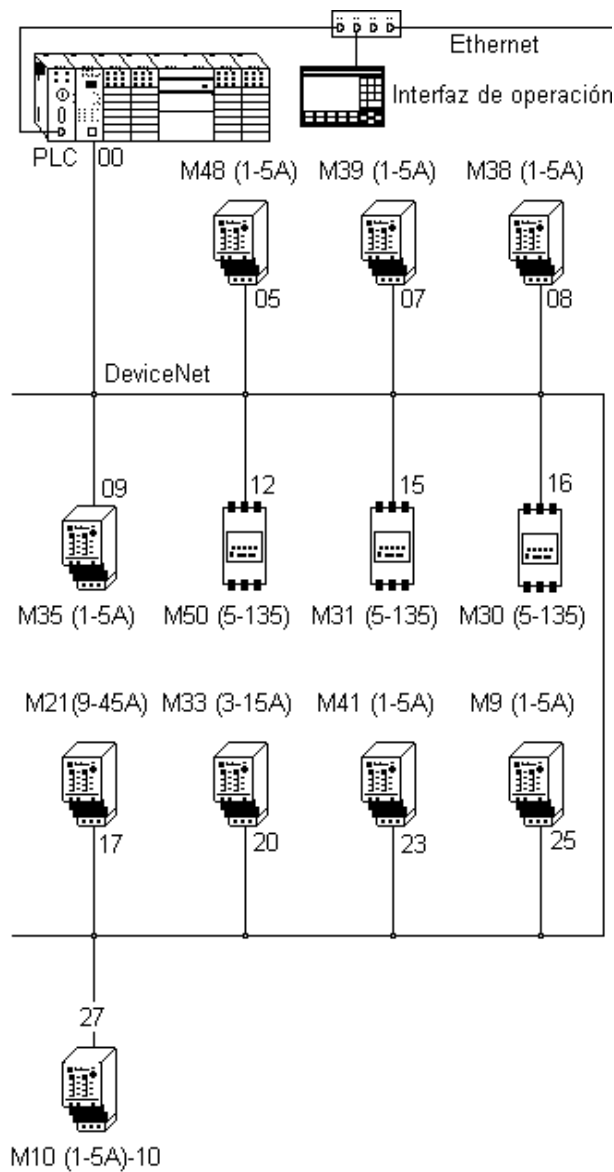


Figura 5.12. Proceso de transporte de harina.



**Figura 5.13. Arquitectura del sistema de transporte de harina.**

### 5.6.2 Filosofía de operación

Para la operación del sistema existen tres selectores, uno por caja, cada selector indica si la respectiva caja se va a llenar o no. La petición de llenado para cada caja aparece cuando el selector de la caja indica que se va a llenar y la misma tiene nivel bajo. La petición de llenado desaparece si el selector respectivo indica que la caja no debe llenarse o si la caja tiene un nivel alto de harina. No se pueden llenar dos cajas de harina al mismo tiempo.

Cada caja de harina puede tener nivel alto, normal y bajo. Los niveles se determinan por medio de dos sensores, uno colocado para indicar el 90% de la capacidad de llenado y el otro para indicar el 50% de la capacidad de llenado. Los equipos del sistema arrancan de forma secuencial, primero los equipos que están cerca de las caja de harina y de último los equipos que están cerca del silo. El objetivo principal del sistema de control es mantener a las tres cajas en nivel normal. Si ocurre una parada imprevista en algún equipo todos los equipos que están hacia el silo de harina deben detenerse de forma inmediata, para evitar derrame de la materia prima. La operación de los equipos se describe a continuación:

#### **Extractor del filtro de la caja de harina 1 (M9)**

Condiciones de marcha:

- Petición de llenado de la caja 1.

Condición de parada:

- No hay petición de llenado de la caja 1 y han pasado 3 segundos que la válvula TSV9 se colocó en la posición para llenar la caja 4.

#### **Extractor del filtro de la caja de harina 2 (M10)**

Condición de marcha:

- Petición de llenado de la caja 2.

Condición de parada:

- No hay petición de llenado de la caja 2 y han transcurrido 3 segundos que la válvula TSV8 se colocó en la posición para llenar la caja 4.

#### **El extractor del filtro de la caja de harina 4 (M41)**

Condición de marcha:

- Petición de llenado de la caja 4.

Condición de parada:

- No hay petición de llenado de la caja 4 y han transcurrido 3 segundos después de se detuvo M33.

### **Válvula neumática de la caja de harina 1 (TSV9)**

En condiciones de reposo se encuentra hacia la dirección de la caja 4.

Condición para posición hacia la caja 1:

- Petición de llenado en la caja 1.

Condición para posición hacia la caja 4:

- No hay petición de llenado de la caja 1 y han transcurrido tres segundos desde que paró el extractor del filtro de la tolva superior o
- No hay petición de llenado de la caja 1 y hay una petición de llenado de la caja de harina 2 o
- No hay petición de llenado de la caja 1 y hay una petición de llenado de la caja de harina 4.

### **Válvula neumática de la caja de harina 2 (TSV8)**

En condiciones de reposo se encuentra hacia la dirección de la caja 4.

Condición para posición hacia la caja 2:

- Petición de llenado de la caja 2.

Condición para posición hacia la caja 4:

- No hay petición de llenado de la caja 2 y han transcurrido tres segundos desde que paró el extractor del filtro de la tolva superior o
- No hay petición de llenado de la caja 2 y hay una petición de llenado de la caja de harina 1 o
- No hay petición de llenado de la caja 2 y hay una petición de llenado de la caja de harina 4.

### **Extractor del filtro de la tolva superior (M33)**

Condiciones de marcha:

- La caja 1 tiene petición de llenado, TSV9 está dirigida hacia la caja 1, TSV8 está dirigida hacia la caja 4 y M09 está en marcha o
- La caja 2 tiene petición de llenado, TSV8 está dirigida hacia la caja 2 y M10 está en marcha o
- La caja 2 tiene petición de llenado, TSV9 y TSV8 están dirigidas hacia la caja 4 y M41 está en marcha.

Condición de parada:

- No hay petición de llenado de ninguna caja y han transcurrido tres segundos de haberse detenido el soplador M31.

### **Soplador (M31)**

Condición de marcha:

- Hay petición de llenado y han pasado tres segundos que el extractor del filtro de la tolva superior (M33) se puso en marcha.



Condiciones de parada:

- No hay petición de llenado y han transcurrido tres segundos de haberse detenido la válvula rotatoria del ciclón.

### **Válvula rotatoria del ciclón (M39)**

Condiciones de marcha:

- Hay petición de llenado y hay presión en el soplador M31.

Condición de parada:

- Si no hay petición de llenado y hace tres segundos que el ciclón (M50) se ha detenido o
- El soplador M31 está detenido.

### **Ciclón (M50)**

Condición de marcha:

- Hay petición de llenado y han pasado 5 segundos luego que la válvula rotativa (M39) se puso en marcha.

Condición de parada:

- No hay petición de llenado y han transcurrido tres segundos de haberse detenido el cernidor (M48) o
- La válvula rotatoria del ciclón está parada.

### **Cernidor (M48)**

Condición de marcha:

- Hay una petición de llenado y han transcurrido 5 segundos desde que el ciclón (M50) se ha puesto en marcha.

Condición de parada:

- No hay petición de llenado y han transcurrido 60 segundos de haberse detenido la válvula rotatoria de la tolva superior (M38) o
- El ciclón (M50) está detenido o
- Se ha activado el interruptor de vibración del cernidor.

### **Válvula rotatoria de la tolva superior (M38)**

Condición de marcha:

- Hay una petición de llenado y han transcurrido tres segundos de la puesta en marcha del cernidor (M48).

Condición de parada:

- No hay petición de llenado y han transcurrido 3 segundos de haberse parado el soplador M30 o si el cernidor (M31) está detenido.

### **Soplador del silo (M30)**

Condición de marcha:

- Hay una petición de llenado y han pasado 3 segundos de haberse puesto en marcha la válvula rotatoria de la tolva superior (M38).

Condición de parada:

- No hay petición de llenado y pasaron 20 segundos de haberse detenido la válvula rotatoria del silo M35 o

- Tolva superior está llena y han pasado 20 segundos de haberse parado M35.

### **Soplador (M21)**

Condición de marcha:

- Hay una petición de marcha y han transcurrido tres segundos de haberse puesto en marcha el soplador M30.

Condición de parada:

- No hay petición de llenado y la válvula rotativa del silo (M35) está parada o cuando la tolva superior está llena.

### **Válvula rotativa del silo (M35)**

Condiciones de marcha:

- Hay una petición de llenado y los sopladores M30 y M21 están en marcha.

Condiciones de parada:

- Han transcurrido cinco segundos después de que cerró la válvula BFV5 y no hay petición de llenado o
- Han transcurrido cinco segundos después de que cerró la válvula BFV5 y se llenó la tolva superior o
- El soplador M31 está detenido.

Permisivos

- 7 segundos dosificando y 7 segundos sin dosificar.

### **Válvula neumática (BFV5)**

Condiciones para abrir:

- Hay petición de llenado y han pasado tres segundos de la puesta en marcha de la válvula rotatoria M35.

Condiciones para cerrar:

- No hay petición de llenado o
- El soplador M30 está parado.

Además las siguientes señales hacen que desaparezca la petición de llenado de las cajas lo que provoca una parada secuencial del sistema.

1. Nivel alto en la tolva superior (HLB-17).
2. Interruptor de vibración del cernidor (VS2).
3. Interruptor de presión de aire para instrumento del cernidor (SP-02).
4. Interruptor de alta presión del soplador M30 (HPS-30).
5. Interruptor de baja presión del soplador M30 (LPS-30).
6. Interruptor de alta presión del soplador M31 (HPS-30).
7. Interruptor de baja presión del soplador M31 (LPS-30).
8. La señal de alarma general (AL) de cualquier actuador del sistema.

### **5.6.3 Función de control**

La función de control para los actuadores M35 y TSV9 se muestra en la figura 5.14, en ella se puede observar la estrecha relación que existe con la filosofía de operación, se agregaron temporizadores para lograr los tiempos de la secuencia de arranque.

Las etiquetas relacionadas con las variables de los actuadores tienen el formato *An\_nombre\_descripción*, donde *n* es el número del actuador, nombre se refiere al nombre del actuador y la descripción está asociada a las variables del actuador, por ejemplo la etiqueta A1\_M9\_AA se refiere a la variable arranque en automático del motor M9, el cual es el primer actuador en la función de control.

La gráfica de la figura 5.15 muestra la operación de los equipos del proceso, en ella se puede observar como el sistema de control responde para mantener las tres cajas en el nivel normal. Cuando una caja alcanza el nivel bajo las válvulas TSV8 y TSV9, se acomodan para llenar esta caja. Una altura de 50% para el sensor inferior garantiza para el caso en que haya más de una caja de harina con nivel bajo que el sistema pueda llenarlas antes de que alguna de ellas se quede sin harina. En la figura también se puede ver el arranque secuencial de los equipos en el siguiente orden: extractor del filtro de la caja a llenar (M9, M10 o M31), válvulas de posición (TSV8 y TSV9), extractor del filtro de la tolva superior (M33), válvula rotatoria del ciclón (M39), soplador del ciclón (M31), ciclón (M50), cernidor (M48), válvula rotatoria de la tolva superior (M38), soplador de transporte del silo (M30), soplador de fluidización del silo (M21), válvula rotatoria del silo (M35) y válvula de salida del silo (BFV5), la secuencia de parada ocurre en orden inverso cuando los tres silos tienen nivel normal. Además, en la figura 5.15 se observa como responde el supervisor ante una falla (indicada con el evento F) en la válvula TSV9 mientras se llena la caja de harina 1. La falla tiene como resultado una parada secuencial y cuando es corregida (evento B) se produce un arranque secuencial para llenar la caja de harina 2 debido a que ésta indicaba nivel bajo.

Este caso de estudio no tiene elementos en la capa de control regulatorio, si se quisiera agregar algún elemento a esta capa, un lazo de control PID por ejemplo, éste podría programarse en el PLC y la comunicación con el supervisor se haría directamente a través de la memoria de datos del controlador, una marcación determinada puede tener como acción el cambio del algún parámetro en el algoritmo de control, por ejemplo cambiar la consigna del lazo PID. La comunicación con el computador donde se ejecuta el sistema

multiagente se hace a través de la red ethernet, los datos enviados tienen información referente a la cantidad de cajas de harina que están en operación, la corriente de los motores (usada para el mantenimiento predictivo) y el estado del sistema (falla, parado o en operación). El sistema multiagente puede colocar o quitar la petición de llenado de cualquiera de las cajas.

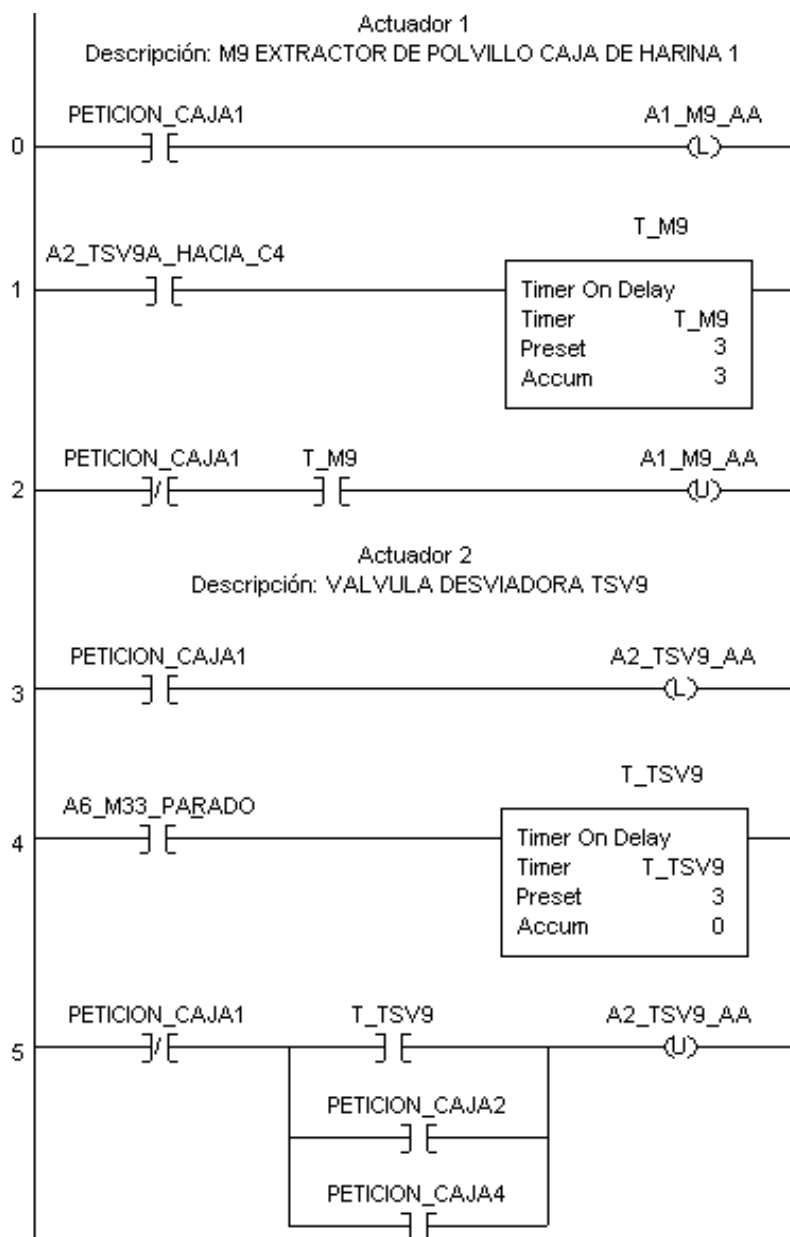


Figura 5.14. Función de control para los actuadores 1 y 2 del sistema de transporte de harina

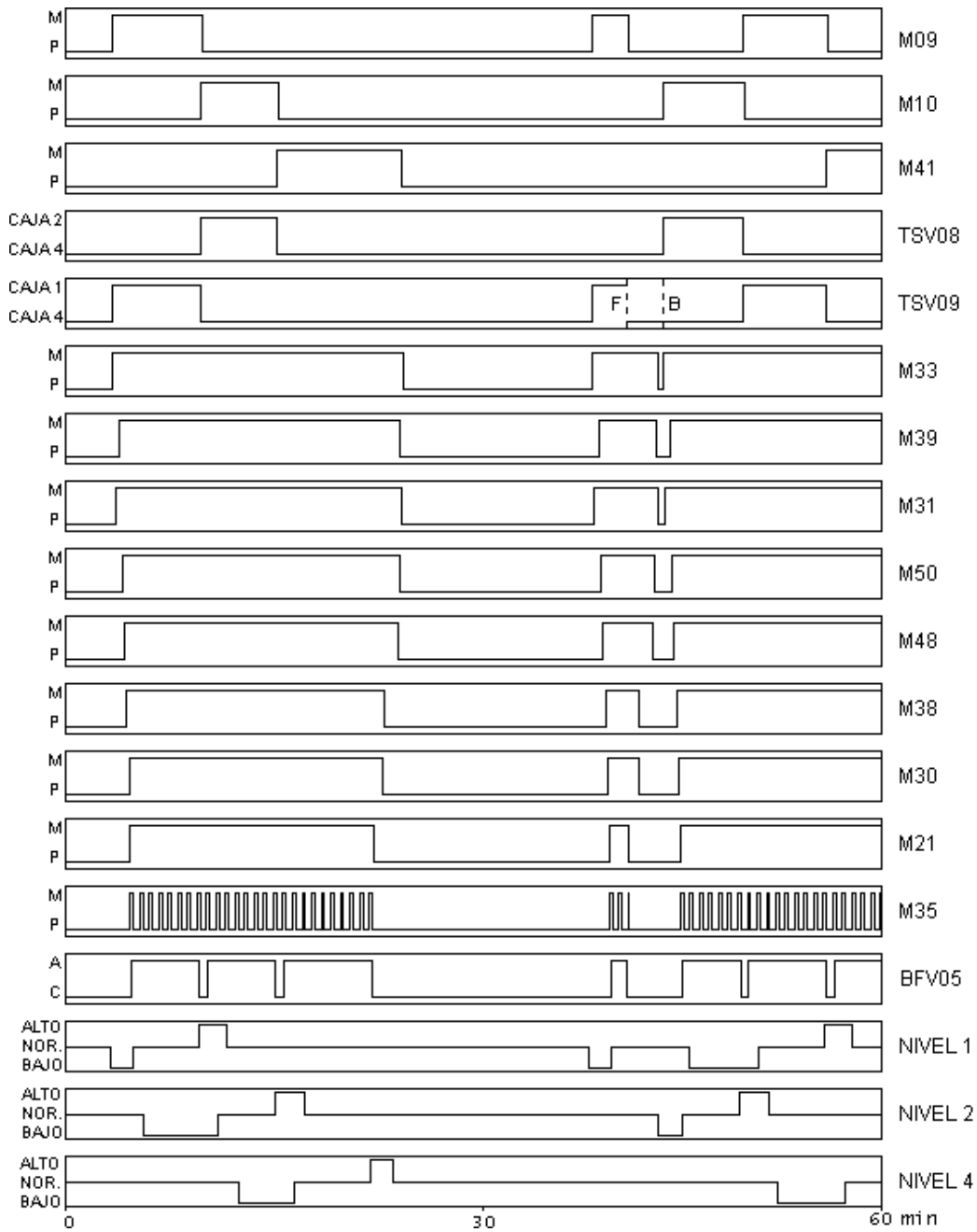


Figura 5.15. Resultados del sistema de control para el transporte de harina. M: marcha, P: parado, A: abierta, C: cerrada.

## Conclusiones

Entre los principales aportes de este trabajo se encuentran:

La definición de un modelo para procesos con dinámicas híbridas, la cual está basada en redes de Petri. En la definición se contempla una interfaz de entrada/salida, de tal forma, que se puede manipular a través de ésta, el estado del modelo de la planta. Por medio de la interfaz de entrada/salida se conecta el equipo de control para la operación del sistema modelo de la planta/supervisor. Este modelo es usado, principalmente, para hacer simulaciones de tal forma que se verifique la conducta del supervisor.

La unidad de producción de la Arquitectura de Integración para la Automatización de Complejo de Producción, se divide en tres capas, control regulatorio, supervisor y coordinador; esto permite simplificar el diseño del sistema de información y toma de decisiones.

Se define el supervisor, para la Arquitectura de Integración para la Automatización de Complejos de Producción, el cual contempla un detector de eventos, un modelo de eventos discretos de la planta y una función de control. El modelo de la planta en el supervisor es una red de Petri y se deriva directamente de la lista de equipos del proceso que se desea automatizar. La función de control está compuesta de una serie de reglas, cada regla permite ejecutar acciones dependiendo del estado de la planta. La función de control se deriva de la filosofía de operación del proceso que se desea automatizar. El detector de eventos sirve de interfaz de entrada/salida para comunicarse con el proceso. Esta definición del supervisor facilita la implementación en el equipo de control, permite la conexión con el proceso, y puede ser diseñado con cualquier herramienta para redes de Petri. Además se muestra como implementar el supervisor en un computador personal usando orientación a objetos, y en un controlador lógico programable, usando diagrama escalera.

La función estándar, la cual se ejecuta junto con el supervisor, se encarga de aquellas acciones que son independientes del proceso. La función estándar simplifica el diseño de



supervisor, permite la operación en manual del sistema, aún en ausencia del supervisor, el código que la compone es reusable y está probado.

La Arquitectura de Integración para la Automatización de Complejos de Producción, hasta el desarrollo de este trabajo no contaba con un método para el desarrollo e implementación de supervisores. El método propuesto permite el ahorro de tiempo en el diseño del supervisor, suministra una vía que permite el desarrollo sistemático del código mediante el uso de un computador personal, además, incluye los modelos necesarios para probar el supervisor antes de la instalación, lo que permite reducir los tiempos para la puesta en marcha. El método permite reproducir la conducta descrita en la filosofía de operación y mejorarla según los resultados encontrados en la simulación.

Una diferencia con otros métodos radica en el hecho de que la implementación en controladores lógicos programables se hace con lógica escalera y no con cartas de función secuencial, esto permite reducir costos en la implementación, debido a que los controladores que usan cartas de función secuencial, normalmente son más costosos.

De las tres capas del sistema de información y toma de decisiones de la unidad de producción, en este trabajo sólo se detalla el supervisor, se deja para trabajos posteriores los aspectos relacionados con el coordinador. Para el diseño e implementación del sistema regulatorio, se puede aprovechar las numerosas investigaciones que existen al respecto. El uso de la función estándar implica que se incluyan todas acciones definidas en ella, se usen o no, esto tiene la desventaja de mayor uso memoria de programa y de datos en el equipo de control.

Este trabajo sirve como base para el desarrollo de una herramienta que permita modelar, hacer síntesis de supervisores, simular y generar el código para controladores lógicos programables en un sólo ambiente. Esto que se traduce en ahorro de tiempo y dinero en el desarrollo de la automatización y reconfiguración del sistema de control en procesos industriales.

## Referencias bibliográficas

Bak, T. and R Izadi-Zamanabadi (2004). Lecture notes - hybrid systems. Technical report. Aalborg University Department of Control Engineering. Aalborg, Dinamarca.

Booch, G., Rumbaugh, J., Jacobson I (1999). “El Lenguaje Unificado de Modelado”. Addison Wesley Iberoamericana. Primera edición.

Brinksma, E., A. Mader and A. Fehnker (2002). Verification and optimization of a plc control schedule. International Journal on Software Tools for Technology Transfer 4(1), 21–53.

Chacón, E. y G. Sarrazin (2004). Sistemas dinámicos en la automatización de sistemas industriales. Technical report. Universidad de los Andes. Mérida, Venezuela.

Chacón, E., I. Besembel, F. Narciso, J. Montilva and E. Colina (2002). An intregation architecture for the automation of a continuous production complex. ISA transaction 41, 95– 113.

Chacón, E., I. Besembel, M Rivero and J. Cardillo (2008). Robotics, Automation and Control. Chap. The Holonic Production Unit: an Approach for an Architecture of Embedded Production Process. 1ra ed. I-Tech. Education and Publishing.

David, R. (1995). Grafcet: A powerful tool for specification of logic controllers. IEEE Trans. on Control Systems Technology 3(3), 253– 268.

Frey, G (1998). Simulation of hybrid systems based on interpreted Petri net. In: Proceedings of the IEE International Conference on Simulation - Innovation Throught Simulation. Vol. 30.09. York, pp. 168–175.

Hellgren, A. (2002). On the implementation of discrete event supervisory control with focus on flexible manufacturing systems. Technical Report 422. Control and Automation

Laboratory, Department of Signals and Systems, Chalmers University of Technology. Suecia.

Indriago, M. y Chacón, E. (2010). Implementación de Supervisores en una Arquitectura de Referencia Basadas en Sistemas de Manufactura Holónicos. *Revista RIAI* 7(3), (pp. 72-82).

Indriago, M. y Z. Lucena (2004). Optimización en el desarrollo de un programa para PLC. *Revista Ingeniería UC*. 11, 70–78.

ISA (1995). ANSI/ISA-S88.01-1995. Batch Control Part 1: Models and Terminology.

ISO (2001). Programmable Logic Controllers - Part 3. International standard IEC 61131-3 (final draft). 2nd ed.

Jennings, N. (2000). On agent-based software engineering. *Artificial Intelligence*, 117:277–296.

Koestler A. (1967), *The Ghost in the machine*, Arkana Paris.

Kowalski, R. (2006). *How to be Artificially Intelligent – the Logical Way*. <http://www.doc.ic.ac.uk/~rak/>. Dept. of Computing, Imperial College, London.

Li, H. (1994). *A Formalization and Extension of the Purdue Enterprise Reference Architecture and the Purdue Methodology*. Ph.d. thesis. Purdue University, West Lafayette. USA.

Martínez, C., Atto, A. y Amari, S. (2008). Supervision of an industrial plant subject to a maximal duration constraint. In: *Proceedings of the 9th International Workshop on Discrete Event Systems*. Göteborg, Sweden. pp. 254–259.

Mayer, R. J., et al, (1995). *IDEF4 Object-Oriented Design Method Report Version 2.0*. Jan 1995.

- Moody, J. y P. Antsaklis (1998). Supervisory control of discrete event systems using Petri net. 1ra ed.. Kluwer Academic Publisher. Norwell, USA.
- Murata, T. (1989). Petri nets: Properties, analysis and applications. IEEE, Proc 77(4), 541–580.
- Peterson, J. (1981). Petri Net Theory and the Modeling of Systems. 1ra ed. Perntice Hall. USA.
- Polic, A. y K. Jezernik (2005). Closed-loop matrix based model of discrete event systems for machine logic control design. IEEE Transactions on Industrial Informatics 1(1), 39–46.
- Ramadge, P. J. and W. M. Wonham (1987). Supervisory control of a class of discrete event processes. SIAM Journal of Control and Optimization 25(5), 206–230.
- Russell, S. y Norvig, P. (2004). Inteligencia artificial un enfoque moderno. 2da ed. Prentice Hall. USA
- Saad, A., K. Kawamura and G. Biswas (1995). Evaluation of contract net-based heterarchical scheduling for flexible manufacturingsystems. In: In Proceedings of the 1995 International Joint Conference on Artificial Intelligence (IJCAI'95), Workshop on Intelligent Manufacturing. pp. 310–321.
- Silveira, M. y M. Combacau (2003). Supervision and control of heterarchical discrete event systems: The laas approach. In: Journal of SBA - Sociedade Brasileira de Automática.
- Tittus, M., B. Egardt y B. Lennartson (1994). Hybrid systems in process control. Technical report. Control Engineering Lab, Chalmers University of Technology. Suecia.
- Williams, T.J., (1993) The Purdue Enterprise Reference Architecture, Information Infrastructure Systems for Manufacturing, IFIP Transactions B-14, Amsterdam, Elsevier Science B.V. (North Holland).

Wooldridge, M. (1997). Agent-based software engineering. IEE Proc. Software Engineering, 144(1):26–37.

Wyns, J. (1999). Reference Architecture for Holonic Manufacturing Systems. Phd thesis. Katholieke Universiteit Leuven, Departamento de Ingeniería Mecánica. Leuven, Bélgica.

Zapata, G. y E. Carrasco (2002). Estructuras generalizadas para controladores lógicos modeladas mediante redes de Petri. DYNA (135), 65–74.

Zeigler, B., Sang, H., Gon, T., y Praehofer, H (1995). DEVS Framework for Modelling, Simulation, Analysis, and Design of Hybrid Systems, In Proceedings of HSAC. 529 - 551. Springer - Verlag.