

**SUPERVISORES DE SECUENCIAS DE EVENTOS  
DISCRETOS DISEÑADOS CON REDES DE PETRI  
BAJO EL SOFTWARE ARENA**



**María Teresa Cárdenas Galvis  
Víctor Gabriel Sandoval Ramos**

*Universidad del Cauca*

**Facultad de Ingeniería Electrónica y Telecomunicaciones  
Departamento de Electrónica, Instrumentación y Control**

**Línea de Control**

Popayán, Junio de 2012

**SUPERVISORES DE SECUENCIAS DE EVENTOS  
DISCRETOS DISEÑADOS CON REDES DE PETRI  
BAJO EL SOFTWARE ARENA**

**ANEXOS**

**María Teresa Cárdenas Galvis  
Víctor Gabriel Sandoval Ramos**

Director:  
PhD. Carlos Alberto Gaviria López

*Universidad del Cauca*

**Facultad de Ingeniería Electrónica y Telecomunicaciones  
Departamento de Electrónica, Instrumentación y Control  
Línea de Control  
Popayán, Junio de 2012**

## CONTENIDO

A.1. PIPE .....	3
A.1.1. Ejecutar Programa.....	3
A.1.2. Entorno de trabajo de PIPE .....	3
A.1.3. Crear una RdP.....	4
Crear un lugar.....	4
Asignar un marcado inicial a un lugar .....	4
Crear una transición .....	4
Crear una transición temporizada .....	5
Crear arcos entre los lugares y las transiciones.....	5
A.1.4. Simular una RdP.....	5
Simulación por selección de transición .....	6
Simulación por selección de transición aleatoria.....	6
A.2. Obtener la matriz de incidencia y el marcado de una RdP .....	7
ANEXO B. MANUAL DE USUARIO DEL SOFTWARE ARENA.....	10
B.1. Configuración de los bloques y módulos.....	10
Módulo Create .....	10
Módulo Process.....	11
Módulo Decide.....	12
Módulo Record .....	13
Modulo de datos Resource.....	14
Modulo de datos Set.....	14
Módulo Delay.....	15
Módulo Hold .....	15
Módulo Match .....	16
Modulo de datos Statistic .....	17
Bloque Duplicate .....	18
B.2. Configuración del tiempo de simulación y cantidad de replicas.....	19
B.3. Ejecución de una simulación.....	20
B.3.1. Ejecución pasó a paso .....	20
B.3.2. Ejecución continúa .....	20

B.4.	Exportar datos de Tallys usando la herramienta <i>Output Analyzer</i> .....	21
ANEXO C.	EJEMPLO COMPLETO DE UNIDAD DE DOSIFICACIÓN.....	23
C.1.	Descripción del ejemplo unidad de dosificación.....	23
C.2.	Restricciones del sistema.....	24
C.3.	RdP resultante al aplicar todas las restricciones.....	25
ANEXO D.	CODIGOS EN MATLAB.....	26
D.1.	Funciones creadas.....	26
a.	transformB.m.....	26
b.	lectura.m.....	27
c.	leer_col.m.....	28
D.2.	Códigos para el ejemplo de la unidad de dosificación.....	29
a.	TK_gela2_2.m.....	29
b.	Transformación_disparo1.m.....	29
c.	Transformación_disparo2.m.....	30
d.	Transformación_disparo3.m.....	31
e.	Transformación_disparo4.m.....	31
f.	Transformación_disparo5.m.....	32
g.	Transformación_disparo6.m.....	32
D.3.	Códigos para el caso de estudio 1.....	33
a.	robotsymesa.m.....	33
b.	análisis_mesa_no_sup.m.....	34
c.	análisis_mesa_sup.m.....	35
D.4.	Códigos para el caso de estudio 2.....	36
a.	Maquinas4_3robot.m.....	36
b.	análisis_FMS_no_sup.m.....	37
c.	análisis_FMS_sup.m.....	38
D.5.	Códigos para el caso de estudio 3.....	40
a.	AGVs.m.....	40
b.	análisis_FMS_no_sup.m.....	40
c.	análisis_FMS_no_sup.m.....	41

## ANEXO A. MANUAL DE USUARIO DEL SOFTWARE PIPE

### A.1. PIPE

PIPE (Platform Independent Petri Net Editor 2.5) es un recurso abierto, sirve para crear y analizar redes de Petri, incluyendo redes de Petri estocásticas generalizadas (GSPN). El software PIPE está construido bajo la plataforma java, su entorno gráfico es muy simple y fácil de usar.

#### A.1.1. Ejecutar Programa

Para iniciar el software PIPE solo se debe ejecutar el archivo llamado *pipe.bat* ubicado en la carpeta donde se encuentra la aplicación *pipe25\_rc5.*, ya sea desde su disco duro o desde una unidad de Cd que contenga el ejecutable de la herramienta.



Figura 1. Icono de acceso a PIPE

#### A.1.2. Entorno de trabajo de PIPE

Después de iniciar el ejecutable aparecerá una ventana como la que se muestra en la figura 2. En la cual se encuentra el entorno de trabajo de PIPE el cual se compone por los menús, la barra de herramientas, la barra de Administrador de Módulos de Análisis (*Analysis Module Manager*) y la ventana de trabajo.

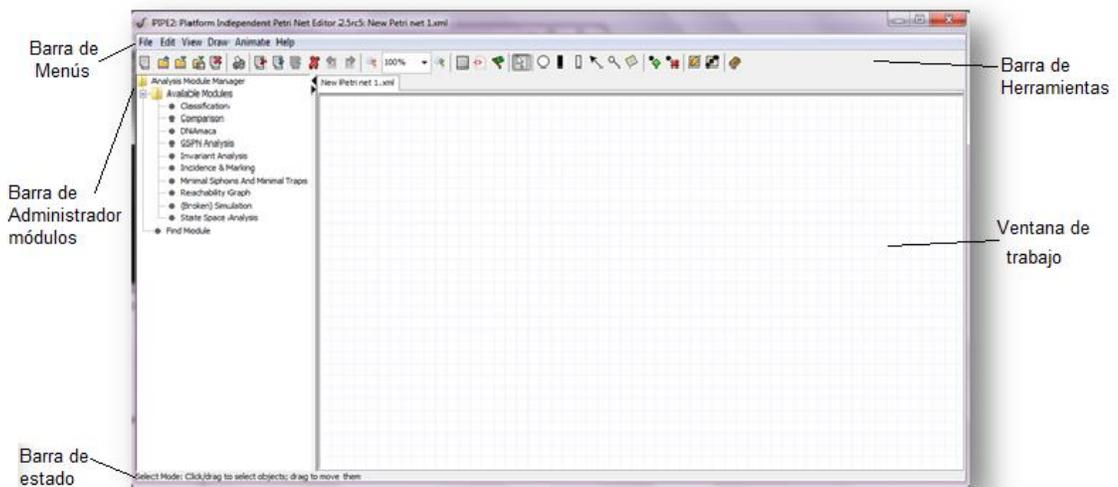


Figura 2. Entorno Software PIPE

### A.1.3. Crear una RdP

Para crear una nueva RdP se debe agregar los lugares, arcos y transiciones que se deseen a la ventana de trabajo. A continuación se realiza una descripción detallada de cada parte que permiten construir una RdP.

#### Crear un lugar

Para agregar un lugar a la RdP se debe hacer clic sobre el botón  de la barra de herramientas y luego poner la cantidad de lugares que se desee en el espacio de trabajo, PIPE por defecto asigna un nombre al lugar empezando desde P0, luego P1 y así sucesivamente.

#### Asignar un marcado inicial a un lugar

Para asignar un marcado inicial a un lugar de una RdP en PIPE, se debe presionar clic sobre el botón  de la barra de herramientas, luego se presiona clic las veces necesarias sobre el lugar a el cual se le desean agregar marcas. Si se desea eliminar marcas de un lugar se presiona clic sobre el botón  de la barra de herramientas y luego se presiona clic sobre el lugar en donde se desea eliminar las marcas. Otra forma de agregar el marcado a un lugar es dando doble clic sobre el lugar y aparecerá una ventana donde se configura el nombre, el marcado y la capacidad.

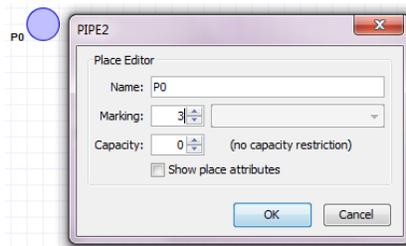


Figura 3. Configuración de un lugar

#### Crear una transición

Para agregar un transición a la RdP se debe hacer clic sobre el botón  de la barra de herramientas y luego poner la cantidad de transiciones que se desee en el espacio de trabajo, PIPE por defecto asigna un nombre a la transición empezando desde T0, luego T1 y así sucesivamente.

## Crear una transición temporizada

Para agregar una transición temporizada a la RdP, se debe hacer clic sobre el botón  de la barra de herramientas y luego poner la cantidad de transiciones que se desee en el espacio de trabajo, PIPE por defecto asigna un nombre desde T0, luego T1 y así sucesivamente. El usuario debe configurar nombre, el tiempo de dispar, el tipo de transición (Temporizado o Inmediata) y la rotación de la barra que representa la transición.

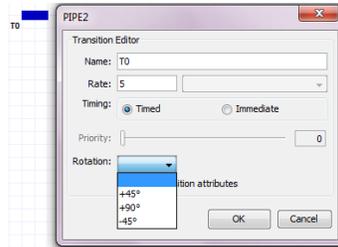


Figura 4. Configuración de la transición

## Crear arcos entre los lugares y las transiciones

Después de que el usuario ha creado sus lugares y transiciones debe realizar las conexiones, para agregar un arco a la RdP se da un clic sobre el botón  de la barra de herramientas, a continuación debe llevar el cursor al lugar en el que se quiere que empiece el arco, donde se da un clic y luego sobre la transición en la que se desea que termine, PIPE por defecto asigna un peso de 1, si se desea modificar dicho peso, se presiona clic izquierdo sobre el arco y se selecciona la opción *Edit Weight* y se abrirá una ventana en la cual se puede ingresar el peso deseado, también es posible agregar puntos para dar movimiento al arco o por último borrarlo.

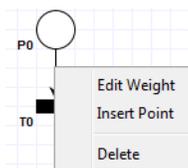


Figura 5. Configuración de un arco

### A.1.4. Simular una RdP

Para iniciar el modo de simulación se debe presionar el botón  de la barra de herramientas, luego de esto la barra cambia de forma como se muestra en la figura.

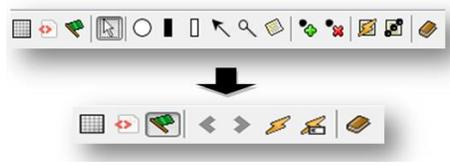


Figura 6. Barra de herramientas de simulación

Las transiciones activas cambian a color rojo, indicando que estas pueden ser disparadas.

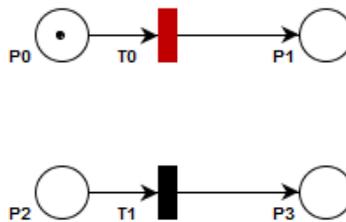


Figura 7. Cambio de color de las transiciones activas en modo de simulación

### Simulación por selección de transición

La simulación por selección de transición consiste en escoger la transición que se desea disparar, luego de ir a modo de simulación las transiciones activas cambian de color estas pueden ser disparadas presionando clic sobre ellas, una vez disparadas se moverán las marcas entre los lugares.

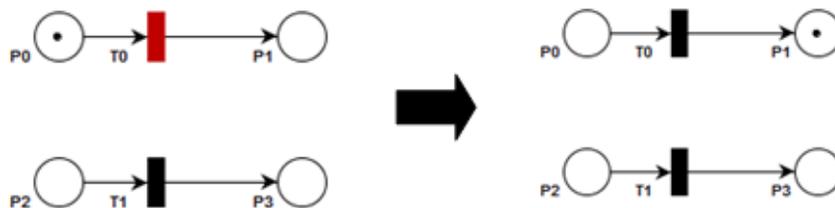


Figura 8. Cambio de marcas debido al disparo de una transición

### Simulación por selección de transición aleatoria

La simulación por selección de transición aleatoria, consiste en que las transiciones se disparan al azar, es decir, si hay varias transiciones activas la probabilidad de que se

dispare alguna es la misma, para esto se presiona  y cualquiera de las transiciones activas se disparará.

Existe además la posibilidad de disparar un número de transiciones deseado, presionando el botón , el cual abrirá una ventana donde se ingresa el número de disparos a realizar, luego de presionar aceptar se abrirá otra ventana en la cual se configura un tiempo de disparo entre las transiciones, finalmente al oprimir *aceptar* se inicia la serie de disparos.

## A.2. Obtener la matriz de incidencia y el marcado de una RdP

Dentro del administrador de módulos de análisis se encuentra una herramienta llamada *Incidence & Marking*, la cual permite obtener la matriz de incidencia y el marcado de la red de Petri modelada en el espacio de trabajo, para esto presionamos doble clic sobre *Incidence & Marking*.

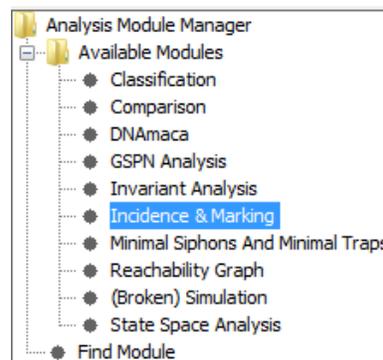


Figura 9. Administrador de herramientas de análisis

Se abrirá una ventana en blanco sobre la cual se debe presionar el botón *Calculate*.

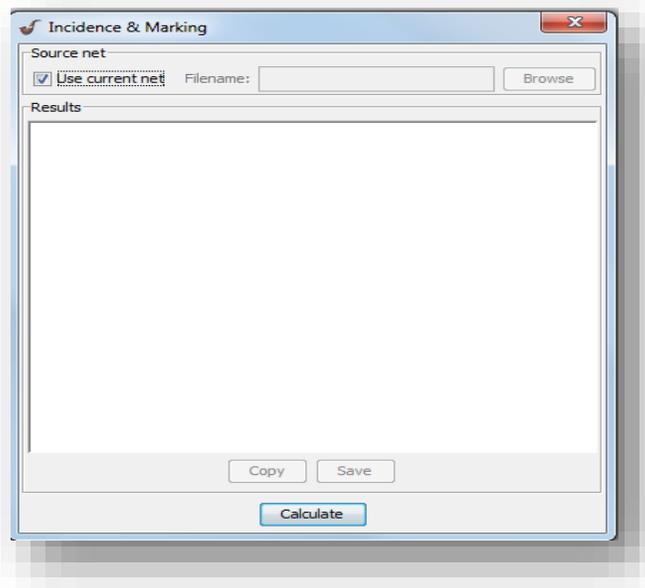


Figura 10. Herramienta de análisis Incidence & Marking

Esta herramienta entrega:

- *Forward incidence matrix* (matriz de incidencia posterior).
- *Backwards incidence matrix* (matriz de incidencia anterior).
- *Combined incidence matrix* (matriz de incidencia combinada).
- *Inhibition matrix* (matriz de inhibición).
- *Marking* (marcado).
- *Enabled transitions* (transiciones activas).

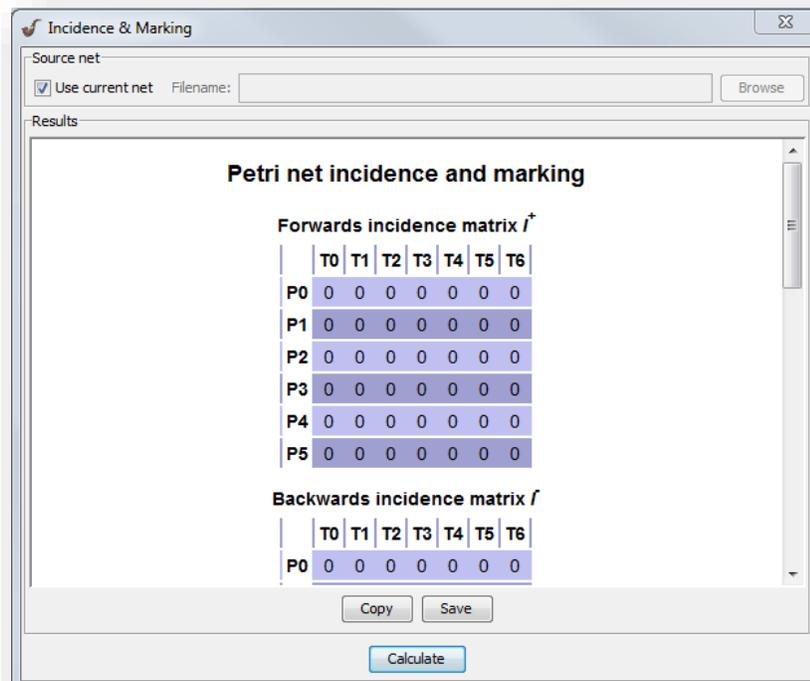


Figura 11. Resultado obtenido con la herramienta Incidence & Marking

Si se desea es posible copiar estas matrices como una imagen o guardarlas en una extensión *.html* presionando el botón *Save* para ser leídas en otros programas.

**Nota:** En este Anexo se explica únicamente las funciones de la herramienta usadas en el desarrollo del proyecto, las funciones adicionales de PIPE pueden ser abordadas en la carpeta *Docs* incluida en la carpeta completa donde se ubica el ejecutable del software.

## ANEXO B. MANUAL DE USUARIO DEL SOFTWARE ARENA

### B.1. Configuración de los bloques y módulos

Los diferentes bloques y módulos del software ARENA pueden ser configurados de diversas formas dependiendo de la necesidad de simulación que se tenga.

#### Módulo Create

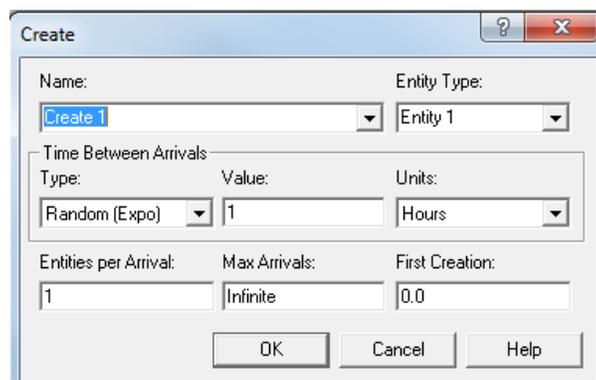


Figura 12. Ventana de configuración del módulo *Create*

Tabla 1. Configuraciones del bloque *Create*

Entrada	Configuración	Descripción	
Name	El nombre de cada módulo debe ser único y si es posible debe asociarse al ejercicio que se esta realizando.	Por defecto se le asigna el nombre del módulo y número de veces que se ha instanciado. Ejemplo <i>Create 1</i>	
Entity type	Nombre del tipo de entidad que se desea generar.	Por defecto Arena asigna crear entidades del tipo <i>Entity 1</i> y una imagen por defecto.	
Type	Flujo de llegada	Random	Distribución exponencial y el usuario especifica la media.
	Schedule	Distribución exponencial y la media está determinada por un bloque <i>Schedule</i> especificado.	
	Constant	Valor constante especificado por el usuario.	
	Expression	Despliega una lista de varias distribuciones.	
Value	Valor en Random	Determina la media de la distribución.	
	Valor en Constant	Determina el tiempo entra llegadas.	
Units	Unidad de tiempo usada entre llegadas.	Puede ser configurada como segundos, minutos, horas, días.	

Entities per Arrival	Número de entidades que llegaran al sistema por cada llegada	Se puede escribir una expresión o un valor constante.
Max arrivals	Máximo número de entidades que el módulo genera.	Por defecto el programa asigna un número infinito como máximo.
First creation	Tiempo en el que inicia la primera creación de las entidades.	Por defecto el programa asigna la primera llegada en el tiempo 0.

## Módulo Process

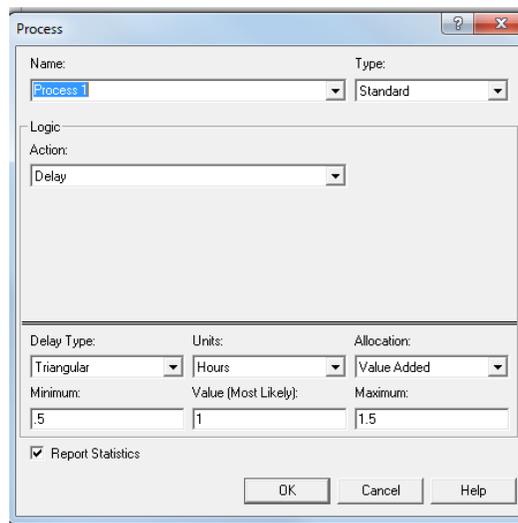


Figura 13. Ventana de configuración del módulo Process.

Tabla 2. Configuraciones de módulo Process

Entrada	Configuración	Descripción
Name	El nombre de cada modulo debe ser único	Por defecto se le asigna el nombre del modulo y número de veces que se ha instanciado.
Action	Delay	Demora del proceso sin recursos.
	Seize Delay	Indica que alguno recursos serán capturados en este modulo con una demora.
	Seize Delay Release	Indica que algunos recursos serán capturados en este módulo con una demora y posteriormente serán liberados.
	Delay Release	Indica que algún recurso capturado

		previamente será demorado y liberado.
Delay type	Tipo de distribución o método que especifica el tipo de retardo.	Puede ser configurado como: Constant, Normal, Triangular, Uniform, Expression.
Units	Unidades de tiempo para el retardo generado.	Puede ser configurada como segundos, minutos, horas, días.
Resources	Lista de recursos usados para el procesamiento de la entidad.	no aplica cuando el bloque sea del tipo Delay

### Módulo Decide

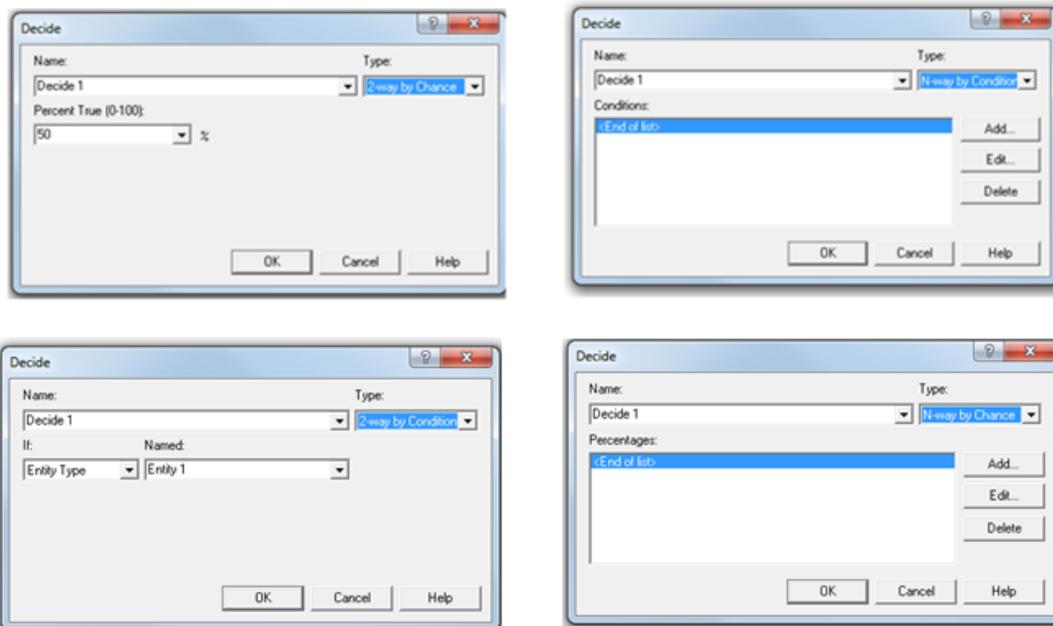


Figura 14. Ventanas de configuración del módulo Decide

Tabla 3. Configuraciones del módulo Decide

Entrada	Configuración	Descripción
Name	El nombre de cada modulo debe ser único.	Por defecto se le asigna el nombre del modulo y número de veces que se ha instanciado.
Type	2-way by chance	2 salidas por un porcentaje definido en <i>Percent true</i> .
	2-way by condition	2 salidas basadas en la condicion <i>if</i> .
	N-way by chance	<i>N</i> salidas basadas en los porcentajes especificados en <i>Percentages</i> .
	N-way by condition	<i>N</i> salidas basadas en las condiciones

		especificadas en <i>Conditions</i> .
Conditions	Define una o varias condiciones usadas para dirigir las entidades.	Se crean tantas salidas como condiciones sean agregadas.
Percentages	Define una o varios porcentajes usados para dirigir las entidades.	Se crean tantas salidas como porcentajes sean agregados.
If	Condicion para la salida de las entidades.	Si se cumple las entidades salen por la rama True, si no por False.
Percent true	Valor que será revisado para determinar el porcentaje de entidades que saldrán por la rama true.	Por defecto el programa asigna una probabilidad de 50-50.

### Módulo Record

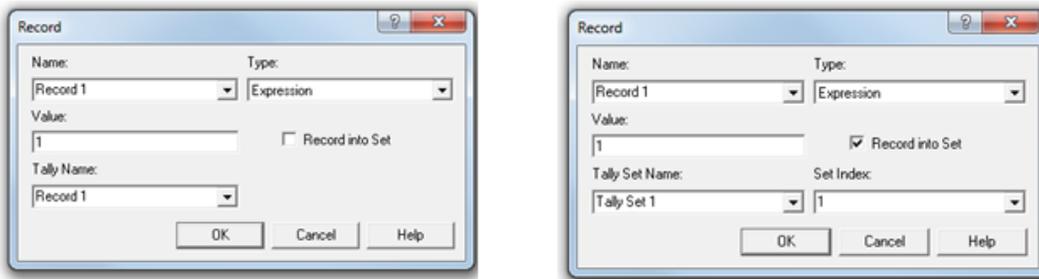


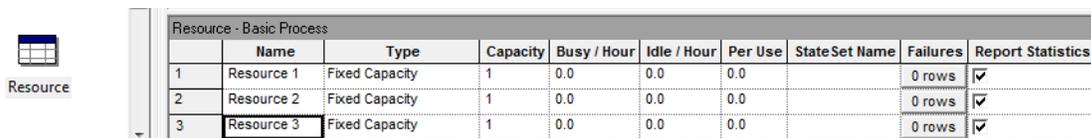
Figura 15. Ventana de configuración del módulo Record

Tabla 4. Configuraciones del módulo Record

Entrada	Configuración	Descripción
Name	El nombre de cada modulo debe ser único.	Por defecto se le asigna el nombre del modulo y el número de veces que se ha instanciado.
Value	Valor que será grabado	Puede ser un número o una expresión con variables definidas por el usuario o propias de ARENA.
Tally name	Define el nombre de la lista ( <i>tally</i> ) en la cual serán grabadas las observaciones (datos).	Por defecto el programa le asigna el mismo nombre del modulo <i>Record</i> .
Record into	Especifica que un	Se deberá definir un modulo <i>Set</i> del tipo <i>Tally</i>

Set	conjunto tipo <i>Tally</i> será usado para almacenar los datos.	para poder usarlo y crear las columnas que sean necesarias.
Tally set name	Nombre del conjunto de Tallys donde se almacenaran los datos.	Por defecto el programa crea <i>Tally set</i> llamado <i>Tally set1</i> , pero se deben agregar las columnas necesarias.
Set index	Índice del tally set donde se desean guardar los datos.	Se debe configurar el índice siempre la y columna creada en el Set cuando exista.

### Modulo de datos Resource



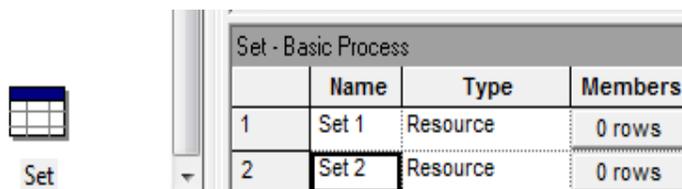
	Name	Type	Capacity	Busy / Hour	Idle / Hour	Per Use	State Set Name	Failures	Report Statistics
1	Resource 1	Fixed Capacity	1	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
2	Resource 2	Fixed Capacity	1	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
3	Resource 3	Fixed Capacity	1	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>

Figura 16. Hoja de cálculo del modulo Resource

Tabla 5. Configuraciones de modulo Resource

Entrada	Configuración	Descripción
Name	El nombre de cada modulo debe ser único.	Por defecto se le asigna el nombre del módulo y número de veces que se ha instanciado.
Type	Determina la capacidad del recurso.	Puede ser configurada de capacidad fija o basada en calendario.
capacity	Número de recursos o unidades disponibles en el sistema.	El programa por defecto pone una capacidad fija por cada recurso agregado.

### Modulo de datos Set



	Name	Type	Members
1	Set 1	Resource	0 rows
2	Set 2	Resource	0 rows

Figura 17. Hoja de cálculo del modulo Set

Tabla 6. Configuraciones de del modulo Set

Entrada	Configuración	Descripción
Name	El nombre de cada modulo debe ser único.	Por defecto se le asigna el nombre del modulo y el número de veces que se ha instanciado.
Type	Resource	Conjunto de recursos.
	Counter	Conjunto de contadores.
	Tally	Conjunto de Tallys.
	Entity Type	Conjunto de tipos de entidades.
Members	Lista los recursos, contadores, listas (tallys) o tipos de entidad que pertenezcan al set.	Se crearan tantas columnas como miembros contenga el conjunto.

### Módulo Delay

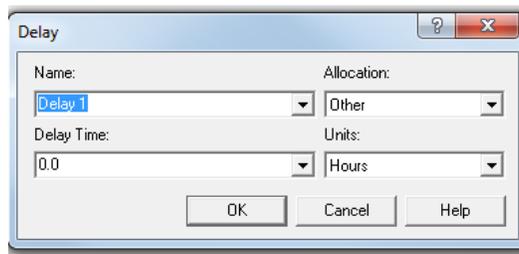


Figura 18. Ventana de configuración del módulo Delay

Tabla 7. Configuraciones del módulo Delay.

Entrada	Configuración	Descripción
Name	El nombre de cada modulo debe ser único.	Por defecto se le asigna el nombre del modulo y número de veces que se ha instanciado.
Delay Time	Determina el valor del retardo de la entidad.	Por defecto se configura en retardo de 0.
Units	Unidad de tiempo para el tiempo de retardo.	Puede ser configurada como segundos, minutos, horas, días.

### Módulo Hold

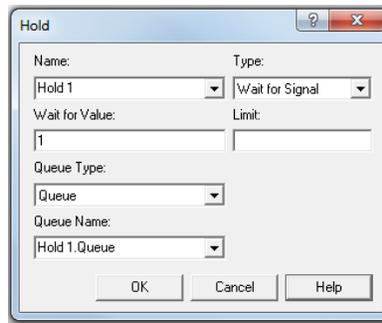


Figura 19. Ventana de configuración del módulo Hold.

Tabla 8. Configuraciones del módulo Hold.

Entrada	Configuración	Descripción
Name	El nombre de cada modulo debe ser único.	Por defecto se le asigna el nombre del modulo y el número de veces que se ha instanciado.
Type	Wait for signal	Mantiene las entidades hasta que un módulo señal active el mismo valor.
	Scan for condition	Mantiene la entidad hasta que la condición especificada sea verdadera.
Wait for value	Debe ser una constante	Se asigna a un módulo <i>Signal</i> una constante, la cual al pasar una entidad envía una señal para que el módulo <i>Hold</i> que espera ese valor libere las entidades.
Condition	Condición que será evaluada para mantener las entidades	Mientras la condición sea falsa las entidades permanecerán en el modulo.
Queue name	Define el nombre de la fila	Por defecto tendrá en nombre del módulo separado por un punto de la palabra Queue.

### Módulo Match

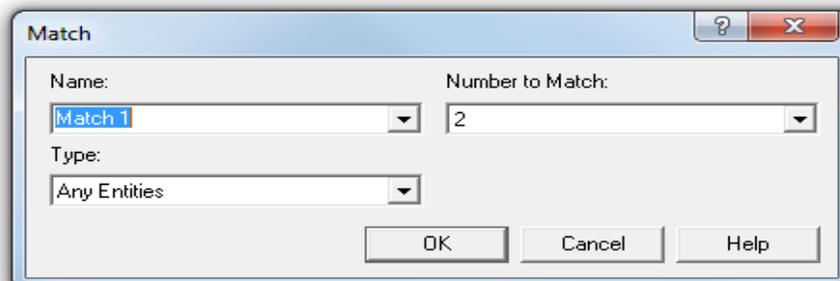


Figura 20. Ventana de configuración del módulo Match

Tabla 9. Configuraciones del módulo Match

Entrada	Configuración	Descripción
Name	El nombre de cada modulo debe ser único.	Por defecto se le asigna el nombre del modulo y número de veces que se ha instanciado.
Number to match	Número de entidades a emparejar	Debe ser un entero entre 2 y 5.
Type	Método para emparejar las entidades que entran	Si es de tipo Any Entities una entidad estará en cada cola del match. Si es basada en atributo una entidad debe estar en cada fila con el mismo atributo.
Attribute name	Nombre del atributo	Atributo para identificar las llegadas de las entidades.

### Modulo de datos Statistic

Statistic - Advanced Process				
	Name	Type	Tally Name	Tally Output File
1	Statistic 1	Tally	Tally 1	C:\Users\Victor\Desktop\archivo_1
2	Statistic 2	Tally	Tally 2	C:\Users\Victor\Desktop\archivo_2

Figura 21. Ventana de datos del modulo Statistic

Tabla 10. Configuraciones del modulo Statistic

Entrada	Configuración	Descripción
Name	El nombre de cada modulo debe ser único.	Por defecto se le asigna el nombre del modulo y número de veces que se ha instanciado.
Type	Tipo de estadístico que se desea crear	Puede ser configurado como <i>Tally</i> , <i>Counter</i> , <i>Output</i> , <i>Frecuency</i> .
Tally Name	Define el nombre de la lista de la cual se desean reportar las estadísticas	Por defecto se le asigna un valor de <i>Tally 1</i>
Tally output File	Especifica el archivo en el cual se van a escribir los datos	No tiene un valor por defecto, por tanto se debe configurar la ubicación y el nombre del archivo.

## Bloque Duplicate

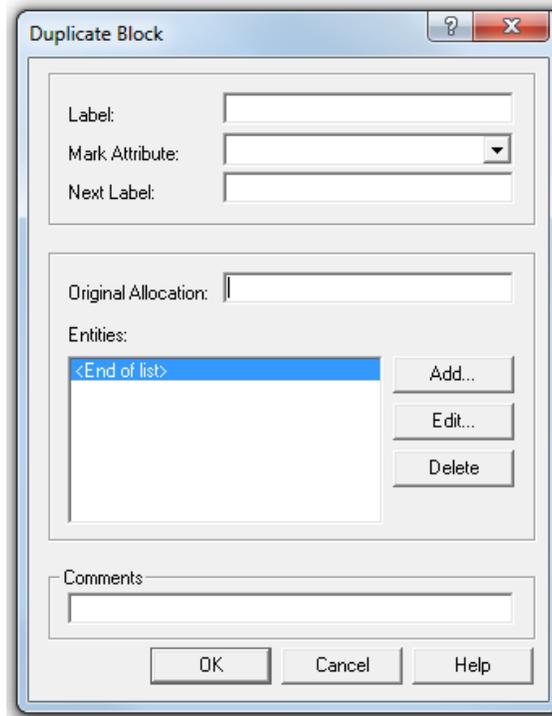


Figura 22. Ventana de configuración del bloque Duplicate

Para la configuración de un bloque Duplicate se debe presionar el botón Add para agregar ramas a la salida, luego es necesario configurar la cantidad a duplicar, es decir cuántas entidades saldrán por cada una que entra al módulo, además es necesario agregar un peso que indica el costo de los duplicados.

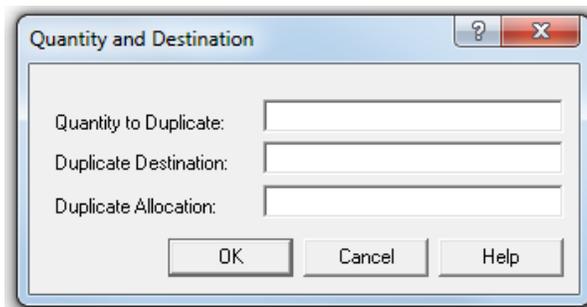


Figura 23. Configuración del módulo Duplicate.

## B.2. Configuración del tiempo de simulación y cantidad de replicas

Para la configuración del tiempo y la cantidad de replicas que se desean simular se debe presionar clic sobre el menú *Run* luego sobre *Setup...*

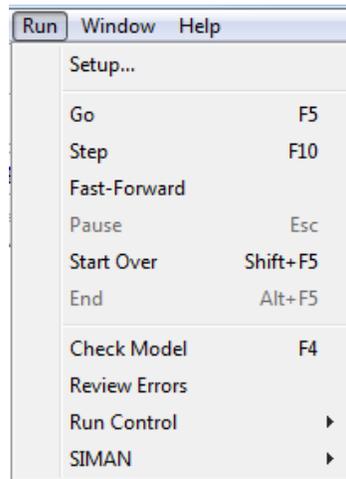


Figura 24. Menú Run.

A continuación se despliega la ventana donde se puede configurar en la pestaña *Replication Parameters* la cantidad de replicas, el tiempo en el que se desea iniciar la simulación, el periodo de calentamiento, la longitud de la réplica y las horas de trabajo al día.

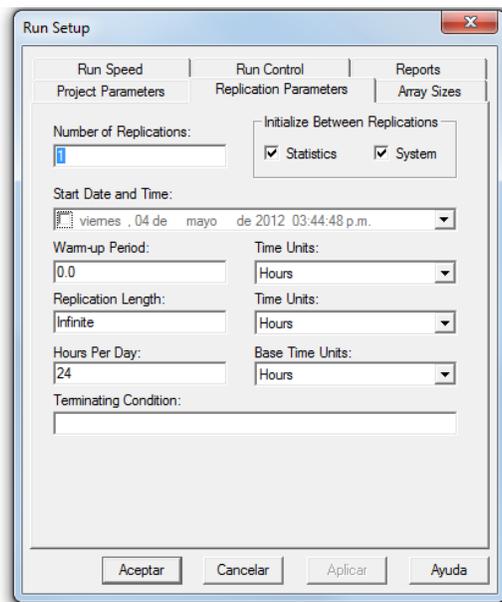


Figura 25. Configuración de la simulación

Tabla 11. Configuraciones de la simulación

Entrada	Configuración	Descripción
Number of Replications	Cantidad de replicas a simular	Se debe ingresar un número entre 1 y 5.
Warm-up Period	Cantidad de tiempo de calentamiento	Tiempo que se adiciona al inicio de la simulación para tomar datos de prueba.
Replication Length	Longitud de la réplica en las unidades de tiempo dadas	Se debe configurar un valor entre 1 e infinito (Infinite).
Hours Per Day	Cantidad de horas al día que se desea simular	Se debe ingresar un número entre 1 y 24.

### B.3. Ejecución de una simulación

La simulación de un modelo se puede iniciar de dos formas: una es usando la barra de simulación y la otra es a través del menú *Run*.



Figura 26. Barra de simulación

#### B.3.1. Ejecución pasó a paso

Para iniciar el modo de simulación, se debe presionar el botón ▶ de la barra de simulación; para simular paso a paso se debe presionar el botón ▶▶ las veces que se desee. Para finalizar la simulación se puede presionar el botón ■.

La barra deslizante permite dirigir la velocidad en que se desea ver el recorrido de las entidades en el modelo, a la izquierda menor velocidad y a la derecha mayor velocidad.

También se puede iniciar el modo de simulación en modo *Step* presionando (*F10*) del menú *Run*, luego para simular paso a paso se debe presionar nuevamente *Step* (*F10*) el botón las veces que se desee. Para finalizar la simulación se puede presionar el botón *End* (*alt + F5*).

#### B.3.2. Ejecución continúa

Para iniciar el modo de simulación se debe presionar el botón ▶ de la barra de simulación, luego para simular de manera continua se debe presionar nuevamente el botón ▶. Para finalizar la simulación se puede presionar el botón ■.

También se puede iniciar el modo de simulación se debe presionar la opción *Go (F5)* del menú *Run*; para finalizar la simulación se puede presionar el botón *End (alt + F5)*.

La simulación se puede realizar de manera acelerada presionando el botón  o haciendo clic sobre la opción *Fast-Forward* del menú de *Run*.

#### B.4. Exportar datos de Tallys usando la herramienta *Output Analyzer*

Para exportar datos obtenidos de las simulaciones en ARENA a archivos de extensión *.dax* a los cuales se pueda acceder desde otro programa (Excel, matlab, etc). Se debe abrir el *Output Analyzer* que se encuentra en la misma carpeta en la que se instaló ARENA.

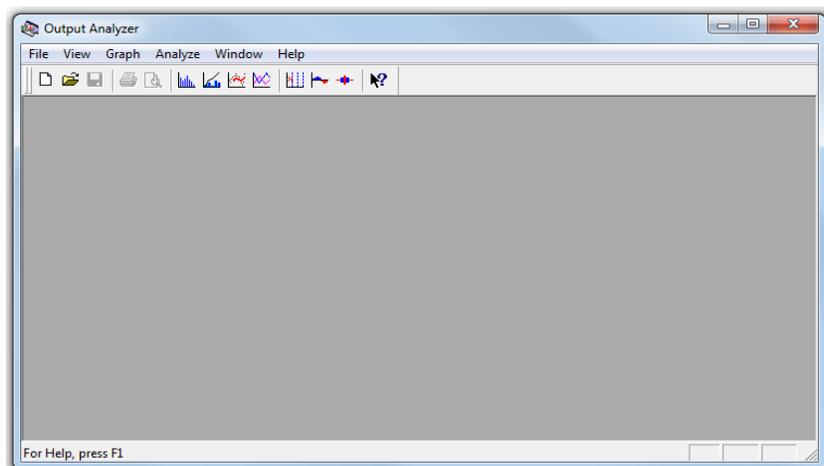


Figura 27. Entorno de la herramienta Output Analyzer

Luego de iniciado el programa se presiona clic sobre el menú *File*, seguidamente sobre *Data File* y después sobre la opción *Export...* se debe abrir una ventana como la de la figura siguiente.



Figura 28. Ventana Export Data File

En esta ventana se debe configurar el archivo creado por ARENA a partir del módulo Statistic en la pestaña *Data File*, luego sobre la pestaña *Export to File* se selecciona la ruta y el nombre del archivo al cual deseamos exportar los datos. Después de presionar aceptar el programa reportará la cantidad de datos escritos en el nuevo archivo.

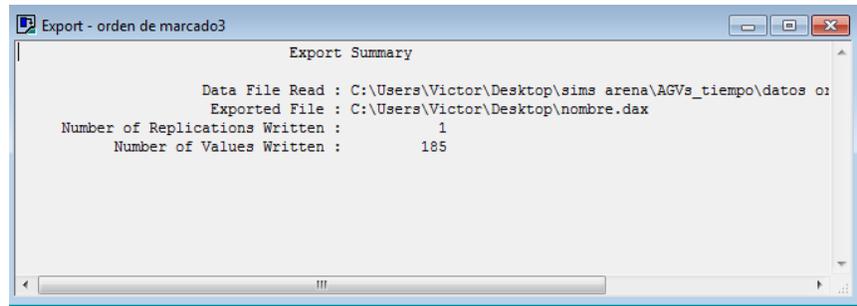


Figura 29. Reporte de archivo exportado en el Output Analyzer

## ANEXO C. EJEMPLO COMPLETO DE UNIDAD DE DOSIFICACIÓN

### C.1. Descripción del ejemplo unidad de dosificación

La unidad de dosificación de la figura 30. Consiste de un tanque, una válvula de alimentación A, una válvula de salida B, y dos sensores, que indican cuando el tanque de dosificación está lleno o vacío. Los dos sensores, L1 en el fondo, y L2 en la superficie del tanque, los cuales pueden sentir encendido o apagado. Cuando ambos están apagados, el tanque está vacío. El tanque esta en un estado intermedio cuando el sensor L1 esta encendido y el sensor L2 está apagado. El tanque está lleno cuando ambos sensores están encendidos. Los eventos de los sensores son no controlables. Inicialmente el tanque esta vacío. Las válvulas A y B pueden estar abiertas o cerradas. Los eventos para abrir y cerrar las válvulas son controlables. El fluido debe ser agitado ya que podría volverse gelatina, detectado por el evento no controlable "gel", cuando la sustancia no está en movimiento. El agitador puede estar encendido o apagado, eventos que son controlables. Inicialmente el agitador está apagado. El proceso de gelatinización puede ocurrir cuando ambas válvulas están cerradas, el tanque lleno, y el agitador no está en marcha. Lo que ocurre cuando la substancia se vuelve gelatina no se modela ya más adelante se desea construir un supervisor que evite el proceso de gelatinización.

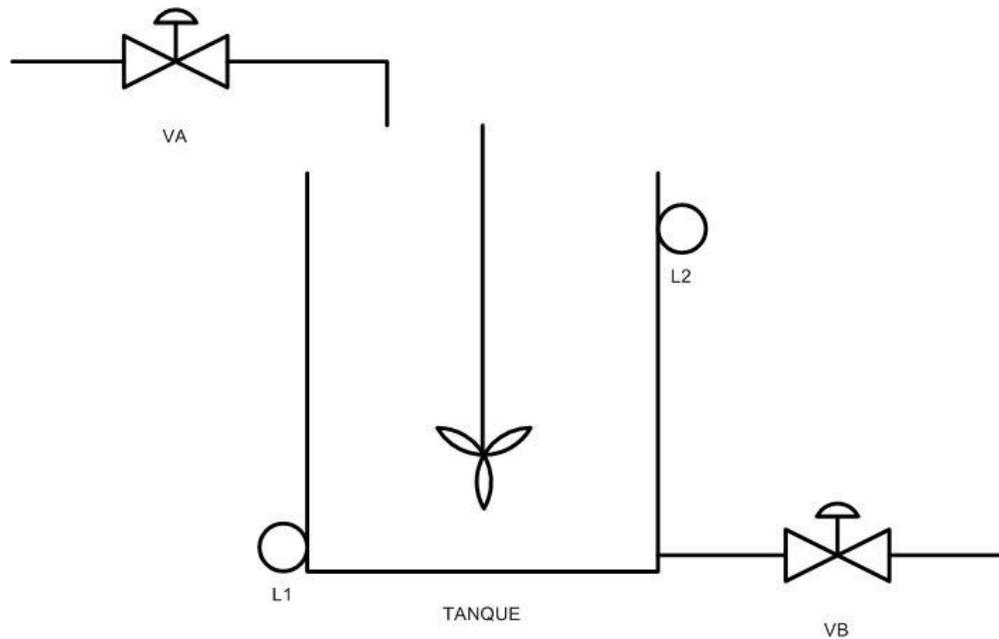


Figura 30. Unidad de dosificación

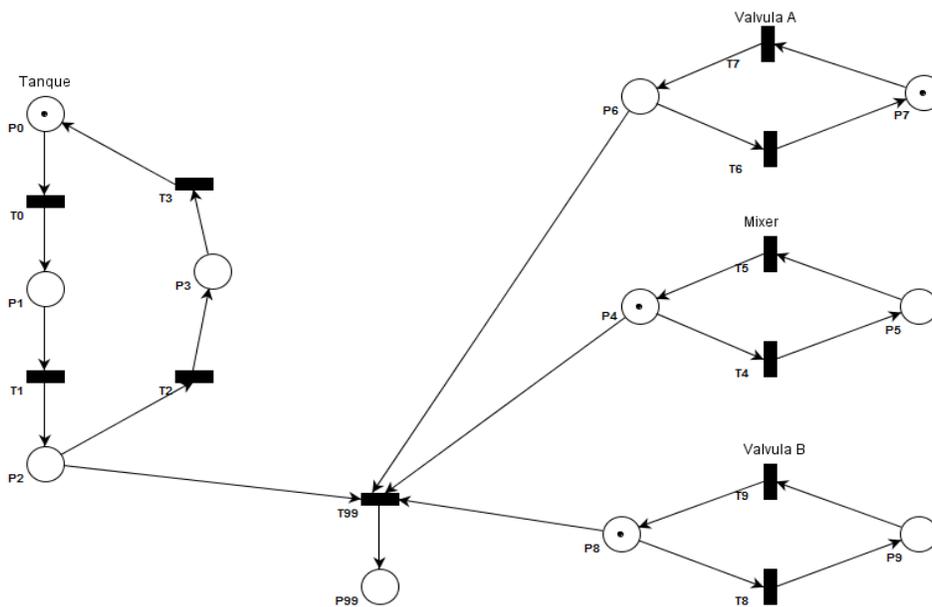


Figura 31. Modelo en RdP del sistema

### C.2. Restricciones del sistema

- No se puede abrir la válvula Va y la válvula Vb al mismo tiempo.

$$\mu_7 + \mu_9 \leq 1$$

- No deben estar el tanque lleno, mixer apagado y las válvulas cerradas.

$$\mu_2 + \mu_4 + \mu_6 + \mu_8 \leq 3$$

- No abrir la válvula Vb hasta que el tanque este lleno.

$$\mu_0 + \mu_1 + q_8 \leq 1 \rightarrow \mu_0 + \mu_1 + \mu_{999} \leq 1$$

- No abrir la válvula Va hasta que el tanque este vacío.

$$\mu_2 + \mu_3 + q_6 \leq 1 \rightarrow \mu_2 + \mu_3 + \mu_{999} \leq 1$$

- El nivel no debe subir si la válvula Va está cerrada.

$$\mu_6 + q_0 \leq 1 \rightarrow \mu_7 + \mu_{999} \leq 1^{**}$$

$$\mu_6 + q_1 \leq 1 \rightarrow \mu_7 + \mu_{999} \leq 1^{**}$$

- El nivel no debe bajar si la válvula Vb está cerrada.

$$\mu_8 + q_2 \leq 1 \rightarrow \mu_1 + \mu_{999} \leq 1 \text{ **}$$

$$\mu_8 + q_3 \leq 1 \rightarrow \mu_1 + \mu_{999} \leq 1 \text{ **}$$

\*\* Varían las ampliaciones de la RdP

### C.3. RdP resultante al aplicar todas las restricciones

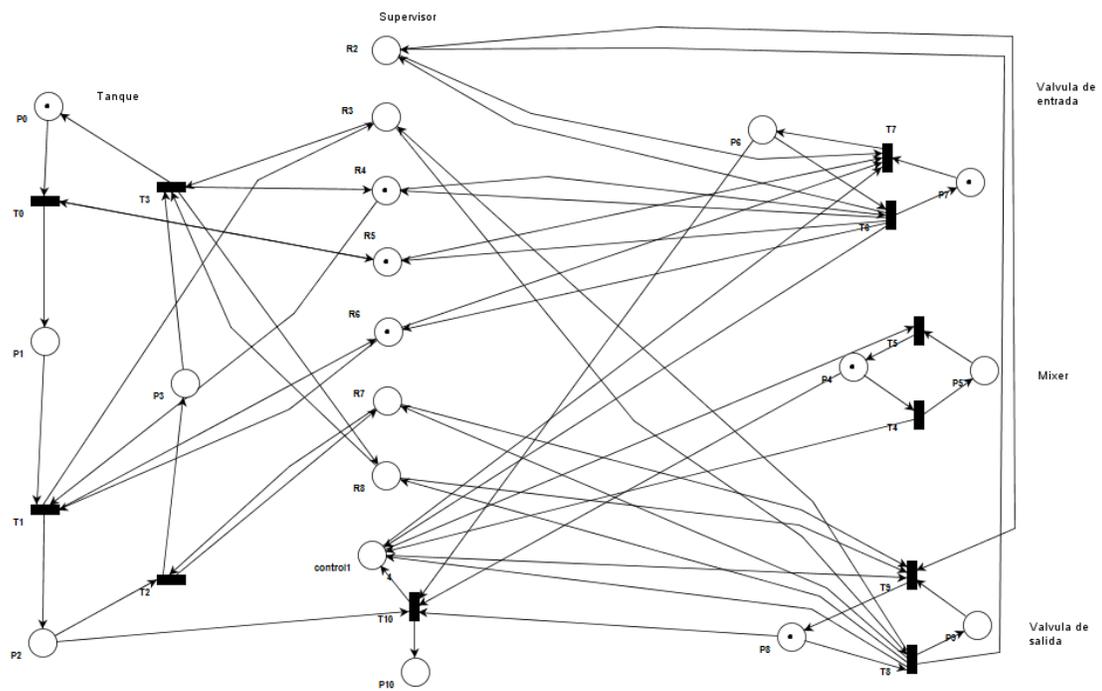


Figura 32. RdP del sistema supervisado.

## ANEXO D. CODIGOS EN MATLAB

### D.1. Funciones creadas

#### a. transformB.m

Esta función realiza una transformación de las restricciones que no cumplen con las condiciones de controlabilidad y observabilidad, los argumentos necesarios son:

- Matriz L → matriz de definición de restricciones.
- Matriz b → matriz de definición de restricciones.
- Matriz Duc → matriz de la parte no controlable (filas no controlables de la matriz de incidencia del sistema).
- Matriz Duo → matriz de la parte no observable controlable (filas no observables de la matriz de incidencia del sistema).
- Matriz up0 → vector del marcado inicial.

La función retorna dos matrices: Lp y bp con las cuales se puede realizar el cálculo del nuevo supervisor que cumple con las restricciones y no se conecta con las transiciones no controlables y no observables.

```
function [Lp,bp] = transformB(L,b,Duc,Duo,upo)
[nc,n]=size(L);
s = size(Duo);
nuo = s(2);
s = size(Duc);
nuc = s(2);
clc;
compLDuc = L*Duc <= 0;
compLDuo = L*Duo == 0;

if isequal(compLDuc,ones(nc,nuc)) && isequal(compLDuo,ones(nc,nuo))
    R1 = zeros(nc,n);
    R2 = eye(nc);
    error = zeros(1,nc);
else
    M = [compLDuc compLDuo];
    for j=1:nc
        if M(j,:)
            R1(j,1:n) = zeros(1,n);
            r2(j) = 1;           % La restricción x es admisible
            error(j)=0;
        else
            lf = L(j,:);
            bf = b(j);
            f = [upo;lf*upo-bf-1;zeros(nuc,1)];
            Aeq = [Duc' (lf*Duc)' eye(nuc,nuc);
                  Duo' (lf*Duo)' zeros(nuo,nuc)];
            beq = -[Duc';Duo']*lf';
```

```

        x =
bintprog(f,zeros(nuc+nuo,n+1+nuc),zeros(nuc+nuo,1),Aeq,beq);
        if x'*f > b-lf*upo
            error(j) = 1;
        else
            error(j) = 0;
        end
        R1(j,1:n) = x(1:n)';
        r2(j) = 1+x(n+1);
    end
    end
    R1
    R2 = diag(r2)
end
Lp = R1+R2*L;
bp = R2*(b+ones(nc,1))-ones(nc,1);
if isequal(error,zeros(1,nc))
    disp('Transformación encontrada');
else
    disp('Aunque se encontró una transformación R1 y R2, el control no es
realizable. ');
    disp(strcat('Las siguientes restricciones no satisfacen Lp*upo <= bp:
',num2str(find(error'))));
end
end

```

## b. lectura.m

Esta función es la encargada de leer las matrices de incidencia de los archivos de extensión *.html* creados en PIPE, para esto se usan comandos de búsqueda de texto propios de Matlab, la función tiene por argumento el nombre del archivo *.html* y retorna las matrices:

- MIpost → matriz de incidencia posterior
- MIant → matriz de incidencia anterior
- MI → marcado inicial

```

function [MIpost,MIant,MI] = lectura(archivo)
fid=fopen(archivo);
%ffid = fopen('intermedio.txt', 'w');
texto = fscanf(fid,'%s');
posincpost = strfind(texto,'Forwardsincidencematrix');
posincant = strfind(texto,'Backwardsincidencematrix');
posmarc = strfind(texto,'Initial');
poscurr = strfind(texto,'Current');

pos =
strfind(texto(posincpost:posincant), '<tdclass="rowhead">')+posincpost;
s = size(pos);
numlugares = s(1,2);
pos(1,numlugares+1)= posincant;

```

```

for i = 1:numlugares
    posval(i,:) =
strfind(texto(pos(i):pos(i+1)), '<tdclass="cell">')+pos(i)+15;
end
s = size(posval);
numtransiciones = s(1,2);
MIpost = zeros(numlugares,numtransiciones);
for i = 1:numlugares
    for j = 1:numtransiciones
        MIpost(i,j) = str2double(texto(posval(i,j)));
    end
end
pos = strfind(texto(posincant:posmarc), '<tdclass="cell">')+posincant+15;
MIant = zeros(numlugares,numtransiciones);
cont=2;
for i = 1:numlugares
    for j = 1:numtransiciones
        MIant(i,j) = str2double(texto(pos(cont)));
        cont = cont+1;
    end
end
pos = strfind(texto(posmarc:poscurr), '<tdclass="cell">')+posmarc+15;
MI = zeros(numlugares,1);
cont=1;
for i = 1:numlugares
    MI(i,1) = str2double(texto(pos(cont)));
    cont = cont+1;
end
fclose(fid);

```

### c. leer\_col.m

Esta función sirve para leer la segunda columna de los archivos de extensión *.dax* creados con el *Output Analyzer* de ARENA y así conseguir el marcado de un lugar en diferentes instantes; la función tiene por argumento el nombre del archivo *.dax* que se desea leer y retorna un vector columna.

```

function [A]=leer_col(nombre)

DELIMITER = ' ';
HEADERLINES = 4;
% Import the file
str = importdata(nombre, DELIMITER, HEADERLINES);
col=str.data;
A= col(:,2);

```

## D.2. Códigos para el ejemplo de la unidad de dosificación

### a. TK\_gela2\_2.m

En este archivo se realiza el cálculo por invariantes de lugar del supervisor para las restricciones 1 y 2 descritas en el anexo C, además se hace uso de las funciones lectura.m para obtener la matriz de incidencia del archivo de PIPE *tk\_gel.html* y transformB.m para transformar una de las restricciones que genera arcos sobre transiciones no controlables.

```
close all
clear all
clc
%obtener matrices de PIPE
[MIp,MIa,up0]=lectura('tk_gel.html');
Dp=MIp-MIa

L = [0 0 1 0 1 0 1 0 1 0 0;
      0 0 0 0 0 0 0 1 0 1 0];
b=[3;1];

%MATRICES Duc Y Duo
Duc=Dp(:,1:4); % Columnas de transiciones no controlables
Duo=[0;0;0;0;0;0;0;0;0;0;0];

%calculo del supervisor
Dc1=-L*Dp
uc0=b-(L*up0)

[Lp,bp] = transformB(L,b,Duc,Duo,up0);

%RESULTADO NUEVO CONTROLADOR

disp('supervisor original');
Dc1
uc0
disp('supervisor transformado');

Dc=-Lp*Dp
uc0=bp-(Lp*up0)
```

### b. Transformación\_disparo1.m

En este archivo se realiza el cálculo del supervisor para la restricción 3 (transformación de disparo) descrita en el anexo C, además se hace uso de las funciones lectura.m para obtener la matriz de incidencia del archivo *tk\_geIRD1.html* de PIPE.

```
%RESTRICCION 1 DE DISPARO R3
```

```
close all  
clear all  
clc
```

```
%obtener matrices de PIPE
```

```
[MIp,MIa,up0]=lectura('tk_gelRD1.html');  
Dp=MIp-MIa
```

```
%RESTRICCION:
```

```
% 1. no abrir vb hasta que el tanque este lleno  
% se puede escribir como que la transicion T9 no se dispare si p1 o p2  
% contienen marcas.  
%  $u_1 + u_2 + t_8 \leq 1 \quad \rightarrow \quad u_1 + u_2 + u_{999} \leq 1$   
%matrices L y b
```

```
L = [1 1 0 0 0 0 0 0 0 0 0 1];  
b = [1];  
Dc=-L*Dp  
uc0=b-(L*up0)
```

### c. Transformación\_disparo2.m

En este archivo se realiza el cálculo del supervisor para la restricción 4 (transformación de disparo) descrita en el anexo C, además se hace uso de las funciones lectura.m para obtener la matriz de incidencia del archivo *tk\_gelRD2.html* de PIPE.

```
%RESTRICCION 2 DE DISPARO R4
```

```
close all  
clear all  
clc
```

```
%obtener matrices de PIPE
```

```
[MIp,MIa,up0]=lectura('tk_gelRD2.html');  
Dp=MIp-MIa;
```

```
%RESTRICCION:
```

```
% 1. no abrir vA hasta que el tanque este VACIO  
% se puede escribir como que la transicion T7 no se dispare si p3 o p4  
% contienen marcas.  
%  $u_3 + u_4 + t_7 \leq 1 \quad \rightarrow \quad u_3 + u_4 + u_{12} \leq 1$ 
```

```
%matrices L y b
```

```
L = [0 0 1 1 0 0 0 0 0 0 0 1];  
b = [1];  
Dc=-L*Dp  
uc0=b-(L*up0)
```

#### d. Transformación\_disparo3.m

En este archivo se realiza el cálculo del supervisor para la restricción 5 (transformación de disparo) descrita en el anexo C, además se hace uso de las funciones lectura.m para obtener la matriz de incidencia del archivo *tk\_gelRD3.html* de PIPE.

```
%RESTRICCION 3 DE DISPARO R5
close all
clear all
clc

%obtener matrices de PIPE
[MIp,MIa,up0]=lectura('tk_gelRD3.html');
Dp=MIp-MIa;

%RESTRICCION:
% el nivel no sube si va esta cerrada
% se puede escribir como que la transicion T1 y T2 no se dispare si p7
% contienen marcas.
%          u7 + t1 <=1    -->  u7+u12 <=1
%          u7 + t2 <=1    -->  u7+u12 <=1

%matrices L y b

L = [0 0 0 0 0 0 0 1 0 0 0 0 1];
b = [1];

Dc=-L*Dp
uc0=b-(L*up0)
```

#### e. Transformación\_disparo4.m

En este archivo se realiza el cálculo del supervisor para la restricción 6 (transformación de disparo) descrita en el anexo C, además se hace uso de las funciones lectura.m para obtener la matriz de incidencia del archivo *tk\_gelRD4.html* de PIPE.

```
%RESTRICCION 4 DE DISPARO R6
clear all
clc
%obtener matrices de PIPE

[MIp,MIa,up0]=lectura('tk_gelRD4.html');
Dp=MIp-MIa;

%RESTRICCION:
% el nivel no sube si va esta cerrada
% se puede escribir como que la transicion T1 y T2 no se dispare si p7
% contienen marcas.
%          u7 + t1 <=1    -->  u7+u12 <=1
%          u7 + t2 <=1    -->  u7+u12 <=1
```

```

%matrices L y b

L = [0 0 0 0 0 0 1 0 0 0 0 1];
b = [1];

Dc=-L*Dp
uc0=b-(L*up0)

```

#### f. Transformación\_disparo5.m

En este archivo se realiza el cálculo del supervisor para la restricción 7 (transformación de disparo) descrita en el anexo C, además se hace uso de las funciones lectura.m para obtener la matriz de incidencia del archivo *tk\_gelRD5.html* de PIPE.

```

%RESTRICCION 5 DE DISPARO R7

close all
clear all
clc

%obtener matrices de PIPE
[MIp,MIA,up0]=lectura('tk_gelRD5.html');
Dp=MIp-MIA

%RESTRICCION:
% el nivel no baja si vb esta cerrada
% se puede escribir como que la transicion T3 y T4 no se dispare si p9
% contienen marcas.
%          u8 + t1 <=1    -->    u7+u12 <=1
%          u7 + t2 <=1    -->    u7+u12 <=1

%matrices L y b

L = [0 0 0 0 0 0 0 0 1 0 0 1];
b = [1];

Dc=-L*Dp
uc0=b-(L*up0)

```

#### g. Transformación\_disparo6.m

En este archivo se realiza el cálculo del supervisor para la restricción 7 (transformación de disparo) descrita en el anexo C, además se hace uso de las funciones lectura.m para obtener la matriz de incidencia del archivo *tk\_gelRD5.html* de PIPE.

```

%RESTRICCION 5 DE DISPARO R8
close all
clear all
clc

%obtener matrices del PIPE

[MIp,MIA,up0]=lectura('tk_gelRD6.html');
Dp=MIp-MIA

%RESTRICCION:
% el nivel no baja si vb esta cerrada
% se puede escribir como que la transicion T3 y T4 no se dispare si p9
% contienen marcas.
%          u8 + t1 <=1    -->    u7+u12 <=1
%          u7 + t2 <=1    -->    u7+u12 <=1
%matrices L y b

L = [0 0 0 0 0 0 0 0 1 0 0 1];
b = [1];
Dc=-L*Dp
uc0=b-(L*up0)

```

### D.3. Códigos para el caso de estudio 1

#### a. robotsymesa.m

En este archivo se realiza el cálculo del supervisor del caso de estudio 1 descrito en el Capítulo 5 de la monografía, el cual cuenta con 2 restricciones: una con invariantes de lugar y otra que incluye el vector de disparo, por eso es necesario leer dos archivos en *html* y calcular los supervisores por aparte.

```

close all
clear all
clc

% RESTRICCION DE DISPARO no debe desocuparse el lado derecho de la mesa
% hasta que se transporte ABC a la cinta
%obtener matrices de PIPE

[MIp,MIA,up0]=lectura('robots_mesaRD.html');
Dp=MIp-MIA;

L1=[0 0 0 0 0 0 1 0 0 0 0 0 0 1];
b=[1];

Dc=-L1*Dp
uc0=b-(L1*up0)

```

```
%RESTRICCION DE CONTROL no avance el material hasta que se haya
completado el ciclo
```

```
[MIp1,MIa1,up01]=lectura('robots_mesa.html');
Dp1=MIp1-MIa1;
```

```
L2=[0 0 1 1 0 0 0 0 0 0 0 0 1];
b1=[3];
Dc1=-L2*Dp1
uc01=b1-(L2*up01)
```

## **b. análisis\_mesa\_no\_sup.m**

En este archivo se realiza el análisis de los marcados alcanzados en la simulación de la RdP del caso de estudio1 sin supervisor implementada en ARENA, por eso es necesario usar la función leer\_col.m; para leer las columnas de los archivos .dax correspondientes al marcado de cada lugar, luego se realiza una comparación de cada fila de la matriz de marcados para observar si los estados no deseados son alcanzables sin supervisor.

```
%NOTA:recuerde incluir la carpeta "datos marcado para matlab_no_sup" que
%se encuentra en sims arena\mesa_robot\ en el path
%funcion para obtener los datos de marcado de los archivos de texto.
```

```
close all
clear all
clc
```

```
a=leer_col('p0.dax');
b=leer_col('p1.dax');
c=leer_col('p2.dax');
d=leer_col('p3.dax');
e=leer_col('p4.dax');
f=leer_col('p5.dax');
g=leer_col('p6.dax');
h=leer_col('p7.dax');
i=leer_col('p8.dax');
j=leer_col('p9.dax');
k=leer_col('p10.dax');
l=leer_col('p11.dax');
m=leer_col('p12.dax');
```

```
marcado1=[a b c d e f g h i j k l m];%matriz de marcados
[sizefila, sizecol]=size(marcado1);
```

```
%busqueda de un marcado prohibido
```

```
disp('restriccion de control')
for z = 1:sizefila
```

```

%RESTRICCION DE CONTROL no avance el material hasta que se haya
%completado
% SE ESCRIBIO COMO QUE LA SUMATORIA DE MARCAS EN LOS LUGARES 9,10y 11 NO
%SEA MAYOR QUE 3 EN UN INSTANTE DADO

if  marcado1(z,9)+marcado1(z,10)+marcado1(z,11)>3;
    disp('')
    disp(['marcado Prohibido Encontrado  En Fila ', num2str(z)])
end
end

disp('restriccion de disparo')
for z = 1:sizefila
% RESTRICCION DE DISPARO no debe desocuparse el lado derecho de la mesa
% hasta que se transporte ABC a la cinta SE INTERPRETA COMO QUE CADA VEZ
% QUE P5 CONTENGA UNA MARCA P8 DEBE TENER UNA no debe darse P5=1 y P8=0

if  [1 0]==[marcado1(z,6) marcado1(z,9)];
    disp('')
    disp(['marcado Prohibido Encontrado  En Fila ', num2str(z)])
end
end

```

### c. análisis\_mesa\_sup.m

En este archivo se realiza el análisis de los marcados alcanzados en la simulación de la RdP del caso de estudio 1 incluyendo supervisor implementada en ARENA, es necesario usar la función leer\_col.m para leer las columnas de los archivos .dax correspondientes al marcado de cada lugar, luego se realiza una comparación de cada fila de la matriz de marcados para observar si los estados no deseados son alcanzables con el supervisor.

```

%NOTA:recuerde incluir la carpeta "datos marcado para matlab_sup" que se
%encuantra en sims arena\mesa_robot\ en el path y que no se encuentre
%agregada otra carpeta agregada que contenga archivos con los nombres
%"p0_sup, p1_sup...p12_sup"

```

```

close all
clear all
clc

a=leer_col('p0_sup.dax');
b=leer_col('p1_sup.dax');
c=leer_col('p2_sup.dax');
d=leer_col('p3_sup.dax');
e=leer_col('p4_sup.dax');
f=leer_col('p5_sup.dax');
g=leer_col('p6_sup.dax');
h=leer_col('p7_sup.dax');
i=leer_col('p8_sup.dax');
j=leer_col('p9_sup.dax');
k=leer_col('p10_sup.dax');

```

```

l=leer_col('p11_sup.dax');
m=leer_col('p12_sup.dax');

marcadol=[a b c d e f g h i j k l m];%matriz de marcados
[sizefila, sizecol]=size(marcadol);

%busqueda de un marcado prohibido

disp('restriccion de control')

for z = 1:sizefila

%RESTRICCION DE CONTROL no avanze el material hasta que se haya
%completado
%el ciclo SE ESCRIBIO COMO QUE LA SUMATORIA DE MARCAS EN LOS LUGARES 9 10
%Y 11 NO SEA MAYOR QUE 3 EN UN INSTANTE DADO

if  marcadol(z,9)+marcadol(z,10)+marcadol(z,11)>3;
    disp('')
    disp(['marcado Prohibido Encontrado  En Fila ', num2str(z)])
end
end

disp('restriccion de disparo')
for z = 1:sizefila
% RESTRICCION DE DISPARO no debe desocuparse el lado derecho de la mesa
% hasta que se transporte ABC a la cinta SE INTERPRETA COMO QUE CADA VEZ
% QUE P5 CONTENGA UNA MARCA P8 DEBE TENER UNA no debe darse P5=1 y P8=0

if  [1 0]==[marcadol(z,6) marcadol(z,9)];
    disp('')
    disp(['marcado Prohibido Encontrado  En Fila ', num2str(z)])

end
end

```

#### **D.4. Códigos para el caso de estudio 2**

##### **a. Maquinas4\_3robot.m**

Este archivo realiza el cálculo del supervisor basado en invariantes de lugar para las restricciones descritas para este caso, por tanto se hace necesario usar la función lectura.m para obtener la matriz de incidencia y el marcado inicial del modelo RdP del archivo 3rob4maq.html.

```

close all
clear all
clc

```



```

%RESTRICCION DE CONTROL no avanze el material hasta que se haya
completado
%el ciclo SE ESCRIBIO COMO QUE LA SUMATORIA DE MARCAS EN LOS LUGARES 9 10
Y
%11 NO SEA MAYOR QUE 3 EN UN INSTANTE DADO

if  marcado1(z,5)+marcado1(z,6)>1;
    disp(['marcado Prohibido Encontrado  En Fila ', num2str(z)])
end
end

disp('restriccion 2')
for z = 1:sizefila
%RESTRICCION DE CONTROL no avanze el material hasta que se haya
completado
%el ciclo SE ESCRIBIO COMO QUE LA SUMATORIA DE MARCAS EN LOS LUGARES 9 10
Y
%11 NO SEA MAYOR QUE 3 EN UN INSTANTE DADO

if  marcado1(z,3)+marcado1(z,4)>1;
    disp(['marcado Prohibido Encontrado  En Fila ', num2str(z)])
end
end

disp('restriccion 3')
for z = 1:sizefila
%RESTRICCION DE CONTROL no avanze el material hasta que se haya
completado
%el ciclo SE ESCRIBIO COMO QUE LA SUMATORIA DE MARCAS EN LOS LUGARES 9 10
Y
%11 NO SEA MAYOR QUE 3 EN UN INSTANTE DADO

if  marcado1(z,1)+marcado1(z,2)>1;
    disp('')
    disp(['marcado Prohibido Encontrado  En Fila ', num2str(z)])
end
end

```

### **c. análisis\_FMS\_sup.m**

En este archivo se realiza el análisis de los marcados alcanzados en la simulación de la RdP del caso de estudio 2 con supervisor implementada en ARENA, por eso es necesario usar la función leer\_col.m para leer las columnas de los archivos .dax correspondientes al mercado de cada lugar, luego se realiza una comparación de cada fila de la matriz de marcados para observar si los estados no deseados son alcanzables con el supervisor.

%NOTA:recuerde incluir la carpeta "datos para matlab\_sup"ubicada el la  
%carpeta \sims arena\FMS\ en el path

%funcion para obtener los datos de marcado de los archivos de texto.

```

close all
clear all
clc

a1=leer_col('p6_sup.dax');
a2=leer_col('p7_sup.dax');
a3=leer_col('p9_sup.dax');
a4=leer_col('p10_sup.dax');
a5=leer_col('p11_sup.dax');
a6=leer_col('p12_sup.dax');

marcado_sup=[a1 a2 a3 a4 a5 a6];%matriz de marcados
[sizefila, sizecol]=size(marcado_sup);

%busqueda de un mercado prohibido
disp('restriccion 1')

for z = 1:sizefila

if marcado_sup(z,5)+marcado_sup(z,6)>1;
    disp(['mercado Prohibido Encontrado En Fila ', num2str(z)])
end

end

disp('restriccion 2')
for z = 1:sizefila

if marcado_sup(z,3)+marcado_sup(z,4)>1;
    disp(['mercado Prohibido Encontrado En Fila ', num2str(z)])
end

end

disp('restriccion 3')
for z = 1:sizefila

if marcado_sup(z,1)+marcado_sup(z,2)>1;
    disp('')
    disp(['mercado Prohibido Encontrado En Fila ', num2str(z)])
end

end

end

```

## D.5. Códigos para el caso de estudio 3

### a. AGVs.m

En este archivo se realiza el cálculo del supervisor basado en invariantes de lugar para las restricciones descritas para este caso, por tanto se hace necesario usar la función lectura.m para obtener la matriz de incidencia y el marcado inicial del modelo RdP del archivo AGV3Maq2Prod.html.

```
close all
clear all
clc

[MIp,MIA,M0]=lectura('AGV3Maq2Prod.html');

Cp = MIp-MIA;% Matriz de Incidencia
%   0 1 1011123 4 5 6 172 3 4 5 6 7 8 9
L = [0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0; %si la maquina3 esta libre
deben evitarse marcas en P11 y P14 dejando solo la opcion P10
    0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0]; %si la maquina2 esta libre
deben evitarse marcas en P11 y P12 dejando solo la opcion P13

b = [1;1];

Cc = -L*Cp % control supervisor
Mc = b-L*M0 % Marcado inicial Control Supervisor
```

### b. análisis\_FMS\_no\_sup.m

En este archivo se realiza el análisis de los marcados alcanzados en la simulación de la RdP del caso de estudio 3 sin supervisor implementada en ARENA, por eso es necesario usar la función leer\_col.m para evaluar las columnas de los archivos .dax correspondientes al marcado de cada lugar, luego se realiza una comparación de cada fila de la matriz de marcados para observar si los estados no deseados son alcanzables sin supervisor.

```
%NOTA:recuerde incluir la carpeta "datos para matlab_no_sup"ubicada el la
%carpeta \sims arena\AGVs_tiempo\ en el path
```

```
%funcion para obtener los datos de marcado de los archivos de texto.
```

```
close all
clear all
clc

a1=leer_col('p11.dax');
a2=leer_col('p12.dax');
a3=leer_col('p14.dax');
a4=leer_col('p15.dax');
```

```

marcado=[a1 a2 a3 a4];%matriz de marcados
[sizefila, sizecol]=size(marcado);

%busqueda de un marcado prohibido

disp('restriccion 1')
for z = 1:sizefila

if marcado(z,3)+marcado(z,4)>1;
    disp(['marcado Prohibido Encontrado En Fila ', num2str(z)])
end
end

disp('restriccion 2')
for z = 1:sizefila

if marcado(z,1)+marcado(z,2)>1;
    disp(['marcado Prohibido Encontrado En Fila ', num2str(z)])
end
end

```

### **c. análisis\_FMS\_no\_sup.m**

En este archivo se realiza el análisis de los marcados alcanzados en la simulación de la RdP del caso de estudio 3 con supervisor implementada en ARENA, es necesario usar la función leer\_col.m para leer las columnas de los archivos .dax correspondientes al marcado de cada lugar, luego se realiza una comparación de cada fila de la matriz de marcados para observar si los estados no deseados son alcanzables con el supervisor.

%NOTA:recuerde incluir la carpeta "datos para matlab\_sup"ubicada en la carpeta \sims arena\AGVs\_tiempo\ en el path

%funcion para obtener los datos de marcado de los archivos de texto.

```

close all
clear all
clc

a1=leer_col('p11_sup.dax');
a2=leer_col('p12_sup.dax');
a3=leer_col('p14_sup.dax');
a4=leer_col('p15_sup.dax');

marcado_sup=[a1 a2 a3 a4];%matriz de marcados
[sizefila, sizecol]=size(marcado_sup);

```

```
%busqueda de un marcado prohibido
disp('restriccion 1')
for z = 1:sizefila

if  marcado_sup(z,1)+marcado_sup(z,3)>1;
    disp(['marcado Prohibido Encontrado  En Fila ', num2str(z)])
end
end

disp('restriccion 2')
for z = 1:sizefila

if  marcado_sup(z,1)+marcado_sup(z,2)>1;
    disp(['marcado Prohibido Encontrado  En Fila ', num2str(z)])
end
end
```