

HERRAMIENTA BASADA EN REDES DE PETRI PARA DISEÑO DE  
SUPERVISORES DE SISTEMAS DE EVENTOS DISCRETOS

ANEXOS

Cristhian David Buchely  
Fausto Ruiz Coque

UNIVERSIDAD DEL CAUCA  
FACULTAD DE INGENIERÍA ELECTRÓNICA Y ELECOMUNICACIONES  
DEPARTAMENTO DE ELECTRÓNICA, INSTRUMENTACIÓN Y CONTROL  
POPAYÁN  
2012

HERRAMIENTA BASADA EN REDES DE PETRI PARA DISEÑO DE  
SUPERVISORES DE SISTEMAS DE EVENTOS DISCRETOS













ANEXOS

Cristhian David Buchely  
Fausto Ruiz Coque

Director  
PhD. Carlos Alberto Gaviria López

UNIVERSIDAD DEL CAUCA  
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES  
DEPARTAMENTO DE ELECTRÓNICA, INSTRUMENTACIÓN Y CONTROL  
POPAYÁN  
2012

## Contenido

1 ANEXO A: MANUAL DE USUARIO DE LA HERRAMIENTA CRP INTEGRADA CON PIPE 2.0.....	2
1.1 Lanzamiento de la aplicación.....	2
1.2 Edición de redes de petri .....	5
1.2.1 Lugar  :.....	5
1.2.2 Transición inmediata  :.....	6
1.2.3 Transición temporizada  :.....	8
1.2.4 Arco  :.....	8
1.2.5 Seleccionar  :.....	9
1.2.6 Adicionar una marca  :.....	9
1.2.7 Eliminar una marca  :.....	9
1.2.8 Anotación  :.....	9
1.3 Celda de pulido y pintado de relojes.....	9
1.3.1 Volver  :.....	15
1.3.2 Continuar  :.....	15
1.3.3 Disparo aleatorio  :.....	15
1.3.4 Animar  :.....	15
1.3.5 Módulos de PIPE.....	16
1.4 Uso de CRP (Control redes de petri) .....	18
1.4.1 Abrir, guardar y cerrar un trabajo .....	20
1.4.2 Creación del supervisor para la celda de pulido y pintado de relojes. 21	
1.4.3 Análisis de bloqueo de la red. ....	30
1.4.4 Generación de código ladder para la celda de pulido y pintado de relojes 34	
1.5 Generar una especificación de un dispositivo (archivo PLC).....	47

2	ANEXO B: DESCRIPCION DE LOS FICHEROS DE PIPE .....	55
3	ANEXO C. METODOLOGÍAS PARA HALLAR SIFONES MÍNIMOS Y TRAMPAS MÍNIMAS EN UNA PN.....	63
4	ANEXO D. DESCARGA E INSTALACIÓN DE NETBEANS Y DEV C...	68
4.1	Descarga e Instalación de NetBeans.....	68
4.2	Descarga e instalación de DEV C.....	71

# 1 ANEXO A: MANUAL DE USUARIO DE LA HERRAMIENTA CRP INTEGRADA CON PIPE 2.0

En este anexo se explicará en detalle todas las opciones y comandos de la aplicación CRP y PIPE.

## 1.1 Lanzamiento de la aplicación.

La aplicación se lanza mediante el archivo ejecutable CRP.exe. Para que esta aplicación pueda funcionar correctamente es necesario que en su directorio se encuentren los siguientes elementos:

- CRP.exe
- RunGui2.jar
- Carpeta xlst
- Carpeta Images
- Carpeta cfg

Opcionalmente se encuentran las carpetas Example nets, conteniendo los ejemplos de PIPE y la carpeta Docs, con la documentación de este programa.

Una vez lanzado, la ventana que aparecerá en pantalla es la siguiente.

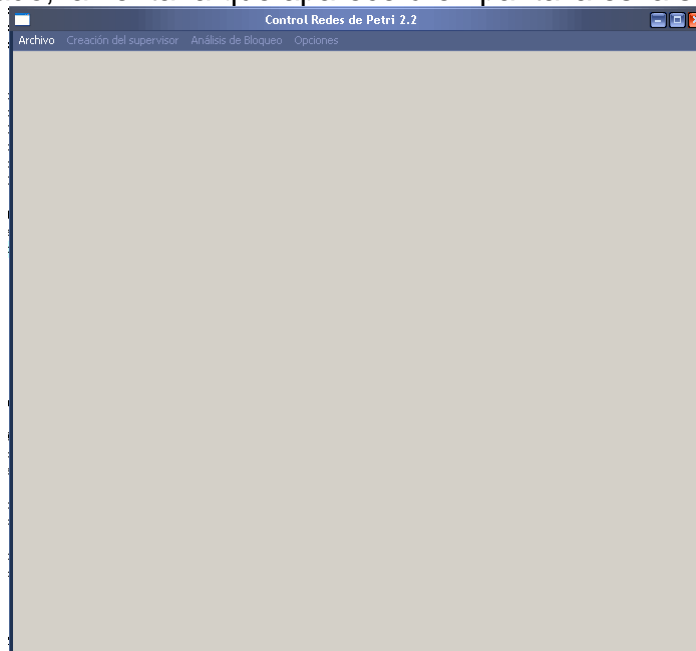


Figura 1. Interfaz de inicio de CRP

El menú archivo es el único menú disponible al inicio de la aplicación, este contiene las siguientes opciones.

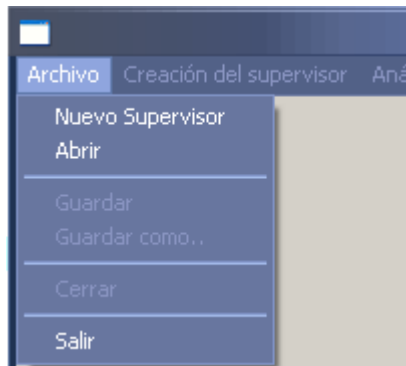


Figura 2. Menú archivo de la aplicación al inicio del programa

Las dos opciones disponibles son crear un nuevo supervisor o abrir otro desde un archivo existente. Por ahora se explicara desde la opción crear supervisor, ya que para la segunda es necesario primero tener un archivo en el cual se haya guardado un trabajo previo.

Una vez seleccionada la opción “Nuevo Supervisor”, la herramienta PIPE será lanzada y CRP cambiará su interfaz para brindar los comandos necesarios para la edición del supervisor y generación de ladder.

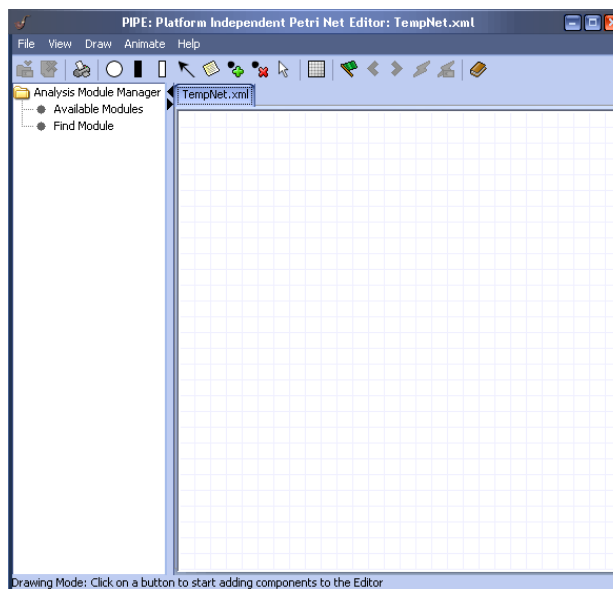


Figura 2. Interfaz de PIPE

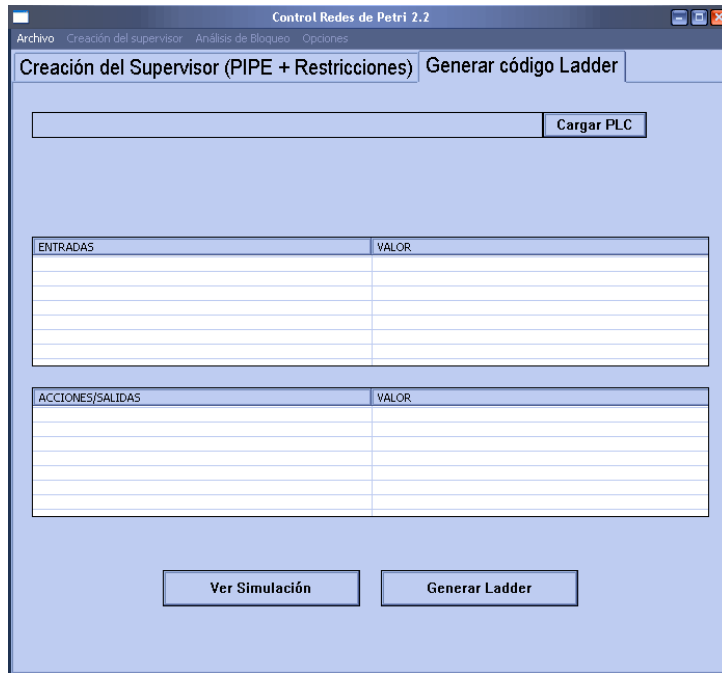
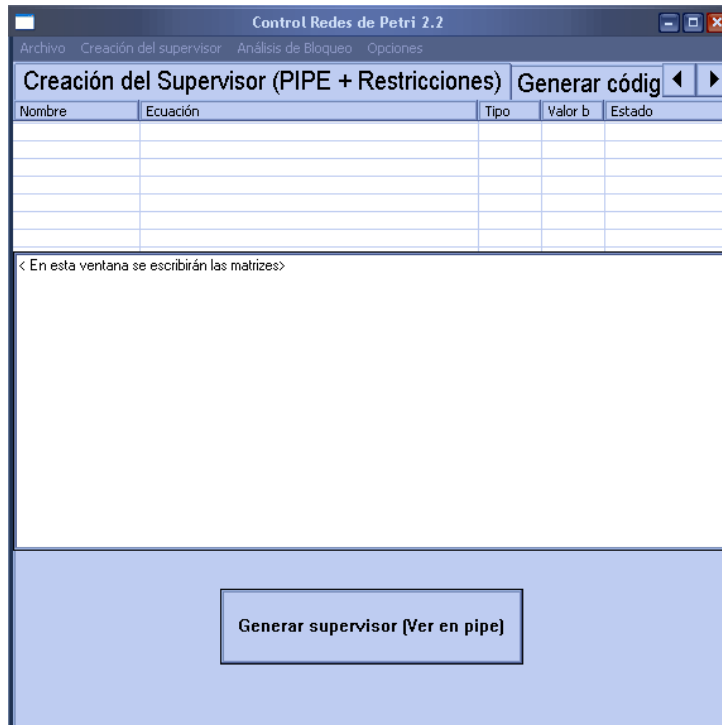


Figura 3. Interfaz de CRP cuando la aplicación es lanzada.

Como se puede observar en la figura tres, todos los controles de CRP se encuentran vacíos debido a que no se ha editado aun la red de petri, en el

momento en que se empiece a realizar esto PIPE y CRP se comunicarán para empezar la creación del supervisor. Por ahora no es necesario enfocarse en CRP, antes se debe manejar PIPE para crear un modelo de una red de petri.

## 1.2 Edición de redes de petri

PIPE usa permite la edición de lugares, transiciones, transiciones temporizadas y arcos de cualquier peso, esto se realiza mediante los siguientes comandos.

### 1.2.1 Lugar :

Adiciona un lugar en la red de petri. Para usarlo se debe seleccionar y posteriormente hacer clic en la coordenada donde se adicionara.

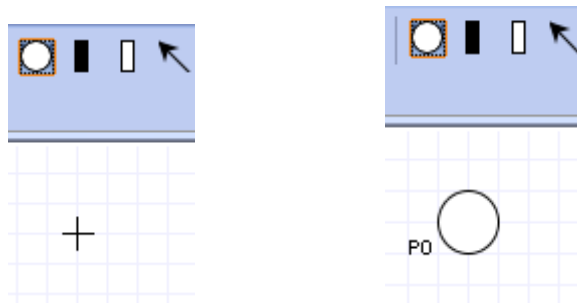


Figura 4. Edición de un lugar

Una vez creado, al hacer clic derecho en el lugar es posible editar el nombre, marcado inicial y la acción de este elemento. Saldrá el siguiente menú con las opciones disponibles.



Figura 5. Menú de edición de lugares

La opción “Delete” eliminará el lugar, la opción “Edit label” editará el nombre del lugar, se desplegará el siguiente cuadro de diálogo donde se editará el nuevo nombre.





Figura 6. Edición del nombre del lugar

Las opciones “Edit Tokens” y “Editar acción” despliegan cuadros iguales para editar las marcas y el nombre de la acción del lugar.

Se recuerda al lector que *acción* se toma en esta tesis como una salida del PLC, la cual es activada cuando algún lugar que la tenga asignada tenga al menos una marca.

### 1.2.2 Transición inmediata :

Adiciona una transición inmediata (no temporizada) en la red de petri. Para usarlo se debe seleccionar y posteriormente hacer clic en la coordenada donde se adicionara.

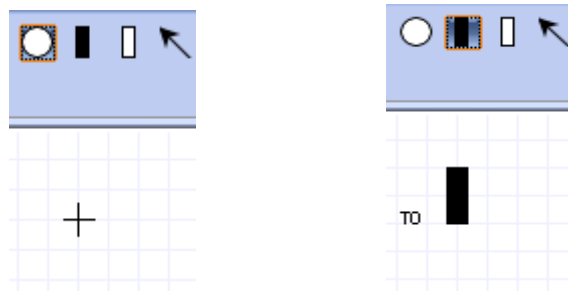


Figura 7. Edición de una transición

Haciendo clic derecho sobre el elemento es posible editar las propiedades de la transición

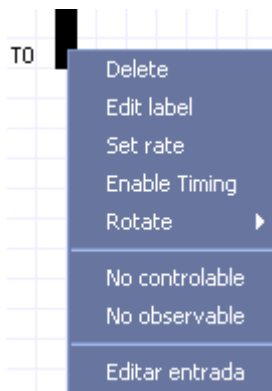


Figura 8. Edición de los atributos de la transición

La opción “Delete” elimina la transición. La opción “Edit label” es igual que en el caso de los lugares, “Set rate” despliega el mismo cuadro de diálogo, en este caso es necesario pasar un número real. Si se selecciona la opción “Enable Timing” la transición se tornará temporizada.

Las opciones “No controlable” y “No observable” editan la controlabilidad y observabilidad de la transición, respectivamente. Si una transición es no controlable se visualizará de color morado, en caso de ser no observable se visualizará de color verde, si alguna transición es tanto no controlable y no observable se editará de color naranja. En caso de ser controlable y observable, se editará negra.



Figura 9. Transiciones inmediatas con distintos atributos de controlabilidad y observabilidad

La opción “Editar entrada” permite adicionar un evento a la transición. Este es un nombre que representa la entrada.

Una vez se selecciona esta opción saldrá un cuadro de diálogo igual al de la edición de los parámetros anteriores. Si se coloca una cadena no nula en este cuadro de edición, se podrán editar dos banderas más para esta transición:

- **Activar una vez:** Indica que el evento se genera solo una vez cuando la señal se activa.
- **Invertir:** Indica que el evento se debe generar si la señal está apagada.



Figura 10. Diálogo de edición del evento.

Las últimas opciones de edición de la transición están más relacionadas con la generación de ladder que con la teoría tradicional de DES, una explicación de estos atributos se puede hallar en la metodología para la generación de ladder expuesta en la monografía de esta tesis.

### 1.2.3 Transición temporizada :

Adiciona una transición temporizada en la red de petri. Para usarlo se debe seleccionar y posteriormente hacer clic en la coordenada donde se adicionara.

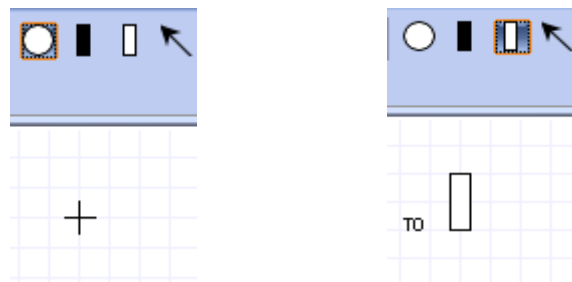


Figura 7. Edición de una transición temporizada

Las opciones de edición para la transición temporizada, son iguales que para la transición inmediata. La única diferencia radica en que en vez de habilitar el timing, ahora es posible deshabilitarlo.

### 1.2.4 Arco :

Crea un arco entre un lugar y una transición. Para usarlo, se debe seleccionar y posteriormente hacer clic en el elemento de origen y después en el de destino.

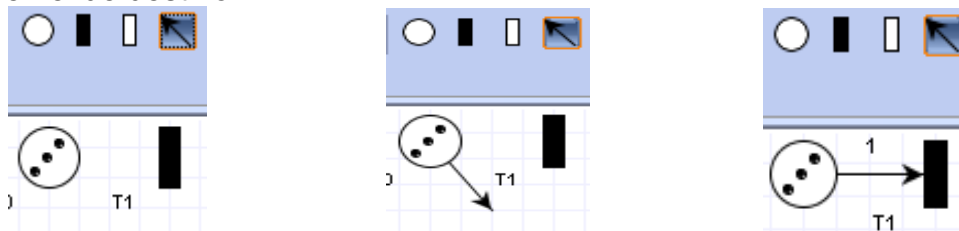


Figura 9. Creación de un arco entre elementos.

### 1.2.5 Seleccionar

Permite usar el cursor para seleccionar uno o varios elementos.

### 1.2.6 Adicionar una marca

Adiciona una marca a un lugar sin necesidad de editar propiedades. Para usarse se debe seleccionar y hacer clic en el lugar a editar.

### 1.2.7 Eliminar una marca

Elimina una marca de un lugar sin necesidad de editar propiedades. Para usarse se debe seleccionar y hacer clic en el lugar a editar.

### 1.2.8 Anotación

Crea un cuadro de texto para colocar etiquetas en una red de petri. Para usarse se debe seleccionar y hacer el clic en la coordenada donde se creara el cuadro.

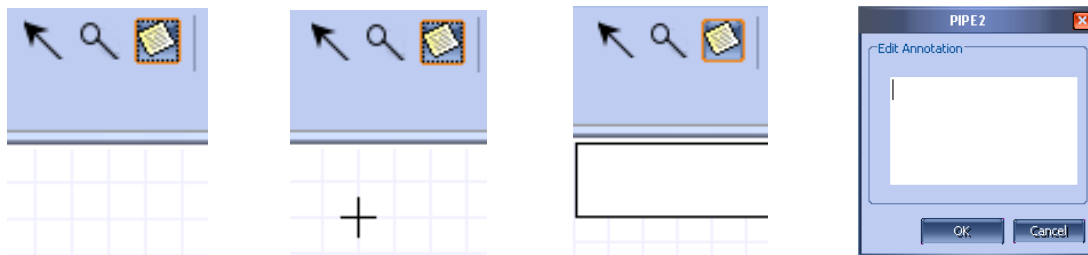
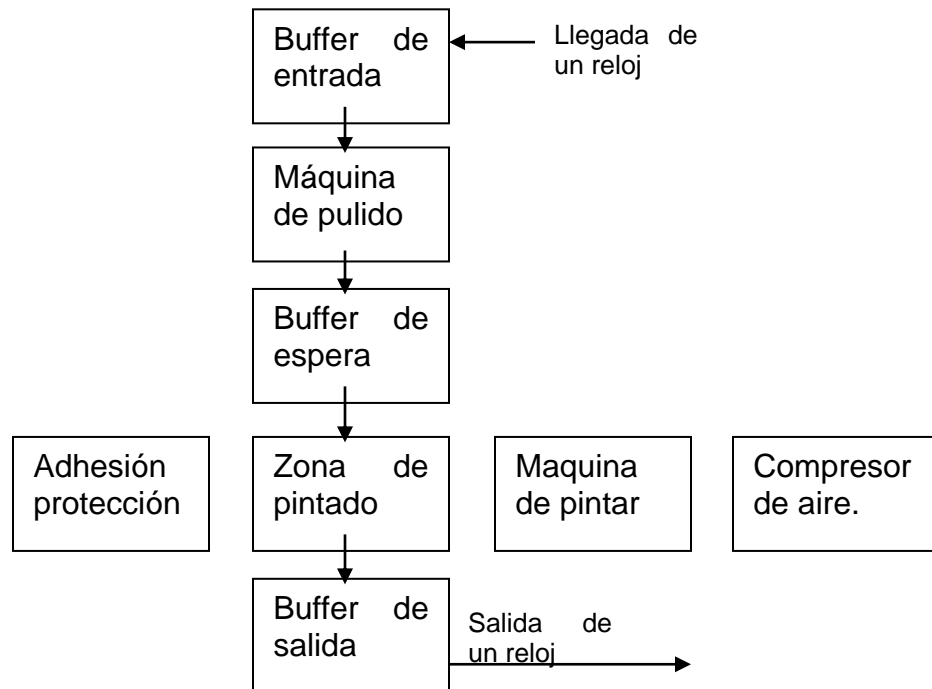


Figura 10. Edición de una anotación.

Con los comandos explicados anteriormente, es posible la edición de cualquier red de petri general. Para la explicación en este manual, se partirá de un ejemplo, el cual ha sido llamado “Celda de pulido y pintado de relojes”, en primer lugar se dará una descripción de este ejemplo y después se mostrará cómo crear una red de petri que lo modele. En la sección de CRP, se creará un supervisor para la celda de relojes y se mostrará en simulación, por último se generará código ladder para programar un PLC Rockwell, que implementa este supervisor.

## 1.3 Celda de pulido y pintado de relojes

Considérese una celda donde se pulen y pintan relojes, para esta celda se ha propuesto la siguiente estructura.



Esta celda está basada en una configuración en línea, en el buffer de entrada llegarán desde otro proceso todos los relojes que necesitan ser pulidos y después pintados, este buffer se considerará de tamaño infinito. Los relojes son pasados uno por uno a la máquina de pulido, esta solo puede trabajar con un reloj a la vez, una vez termina esta máquina transporta el reloj al buffer de espera de la zona de pintado, este buffer tiene capacidad máxima tres relojes. Un sistema de fijación toma un reloj del buffer de espera y lo pasa a la zona de pintado, una a la vez en esta zona, antes de pintar se deben adherir protectores a las zonas del reloj que no deberían ser pintadas, tales como el vidrio donde se visualiza la hora y el logotipo de la empresa, esto lo realiza un mecanismo automático, posteriormente se empieza a pintar el reloj, se dispone de una maquina de pintar la cual funciona mediante aire en un compresor, este último puede quedarse sin aire en cualquier momento, cuando esto ocurre se debe apagar la maquina (de pintar) y encender el compresor 10 minutos para recargar, la operación de pintado debería reiniciarse después de un apagado forzoso para evitar fallas de calidad. Después de que el reloj ha sido pintado pasa a un buffer de salida en el cual esperará a que sea llevado a la zona de empaque, este buffer también se considera de capacidad infinita. Se supone que el PLC que implementa el supervisor será avisado por medio de señales lógicas cuando alguno se los siguientes sucesos ocurran:

1. Un reloj llego al buffer de entrada.
2. La máquina de pulido terminó de procesar un reloj.
3. El sistema automático termino de colocar las etiquetas.

4. La máquina de pintar terminó de pintar un reloj.
5. El compresor se quedo sin aire.

El trabajo del supervisor, será encargarse de activar los elementos de la celda cuando corresponda para que esta funcione correctamente, estos elementos son:

1. La máquina de pulido. Cuando esta se activa, toma un reloj del buffer de entrada y solo se apaga cuando termina la operación, automáticamente transportando el reloj al buffer de espera de la zona de pintado.
2. Sistema de fijación, cuando este se activa toma un reloj del buffer de espera y lo fija en la zona de pintado. Al desactivarse pasa el reloj al buffer de salida.
3. La máquina de pintar, cuando esta se activa inicia la operación de pintado y se desactivará automáticamente cuando termine indicando el fin del pintado. También es posible apagarse de manera forzosa, antes de terminar el pintado.
4. Sistema de adhesión de protecciones. Cuando este sistema se activa comienza a adherir las protecciones, se desactivará automáticamente al terminar indicando el fin de la operación.
5. Compresor de aire. Este es un compresor normal de aire, el supervisor lo puede activar o apagar a placer para mantener la máquina de pintado abastecida de aire. No se debe mantener prendido permanentemente este elemento.

En el anterior proceso es muy sencillo de modelar y supervisar, en seguida se muestra el procedimiento para crear la red que lo modela. El lector no hallará mayor problema en crear el modelo en PIPE, con la explicación de los comandos ya hechos arriba.

Primero será necesario crear un modelo para la línea de operaciones, en el cual se modele la llegada de un reloj al buffer de entrada y todo su proceso hasta llegar pulido y pintado al buffer de salida. Estas operaciones se pueden modelar con la siguiente red de petri.

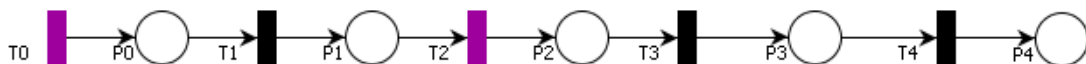


Figura 11. Red de petri de la línea de operaciones, para el caso del pulido y pintado de relojes.

P0	Buffer de entrada
P1	Maquina de pulido
P2	Buffer de espera para zona de pintado
P3	Zona de pintado
P4	Buffer de salida de relojes terminados

T0	Llegada de un reloj al buffer de entrada, esta transición es no controlable, pues no se puede evitar recibir una reloj.
T1	Paso de un reloj del buffer de entrada a la máquina de pulido.
T2	La máquina de pulido termina de procesar una pieza y la pasa al buffer de la zona de pintado. Esta transición es no controlable, pues una vez empieza no es posible detener la máquina de pulido.
T3	Un reloj pasa de la zona del buffer de espera a la zona de pintado
T4	Un reloj es terminado y pasa al buffer de salida

La red anterior modela el flujo de relojes mediante las marcas, una marca representa un reloj, y según el lugar donde se encuentre indica el lugar

Aún falta modelar 3 componentes que no dependen directamente de la línea de producción, estos son la máquina de pintar, el compresor y el sistema de adhesión de protecciones.

Para la máquina de pintar se tiene el siguiente modelo.

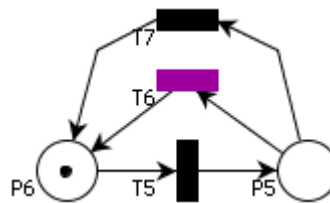


Figura 12. Modelo de la máquina de pintar en red de petri

P5	Máquina de pintar es encuentra operando
P6	Máquina de pintar está apagada

T5	La máquina se pasa a operar.
T6	La máquina termina de pintar un reloj y pasa

	automáticamente ha apagado.
T7	La máquina se apaga de forma forzosa, en cualquier etapa del proceso de pintado.

Para el compresor, se ha diseñado el modelo de la figura trece, nótese que en este modelo está contemplada la temporización de 10 minutos que se debe mantener encendido el elemento hasta recargar el aire.

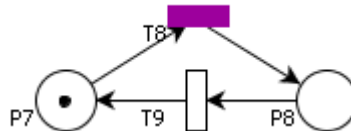


Figura 13. Modelo en red de petri del compresor.

P7	El compresor está apagado y contiene suficiente aire
P8	El compresor esta activo para recargar el aire

T8	El compresor se quedó sin aire y pasa a un estado encendido para recargar. Esta transición es incontrolable, pues no es posible anticipar este suceso.
T9	El compresor ha sido recargado, después de 10 minutos de estar encendido y se apaga. Para simulación, solo se pondrán 5 segundos de temporización.

Finalmente solo falta el sistema automático de adhesión de protecciones, este sistema tiene el siguiente modelo en red de petri.

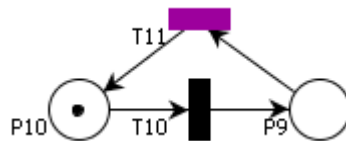


Figura 14. Modelo del sistema de adhesión de protecciones

P9	El sistema está activo y colocando etiquetas al reloj en la zona de pintado.
P10	Es sistema está inactivo.

T10	El sistema pasa a estado activo.
T11	El sistema termina la operación de adhesión y pasa a apagado. Esta transición es no



	controlable pues una vez activo no es posible detener este sistema.
--	---

Con la explicación de los comandos para editar redes de petri, el usuario no encontrará mayor dificultad para construir el modelo en red de petri de la celda de pulido y pintado de relojes. A continuación se muestra el modelo completo en una misma red de petri.

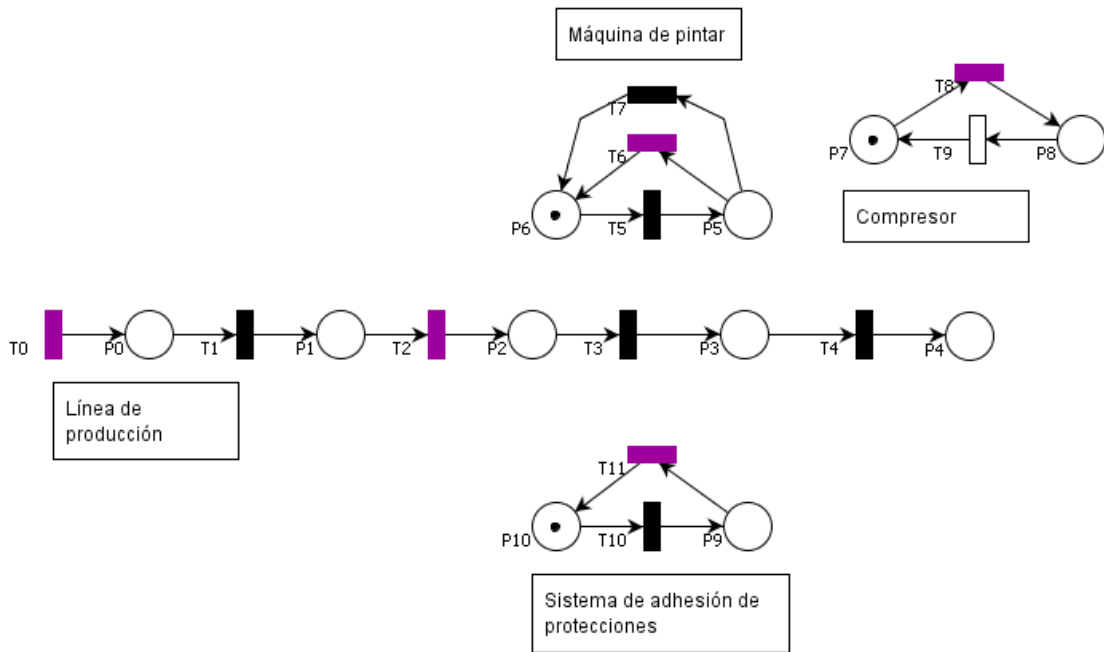



Figura 15. Modelo de la red de petri para la celda de pulido y pintado de relojes

Una vez realizado el modelo en red de petri, es posible utilizar PIPE para animar la red de petri, buscar sifones y trampas mínimas, calcular matrices de incidencia, pre, post y vector de marcado. Estas opciones ayudan a verificar el modelo, detectar bloqueos y visualizar la evolución de la red.

La animación de la red se puede realizar mediante el botón animación el cual tiene la figura , al presionar esta opción los comandos de edición de la red de petri se desactivarán y los comandos de animación se habilitarán. En rojo se resaltarán las transiciones habilitadas para dispararse.

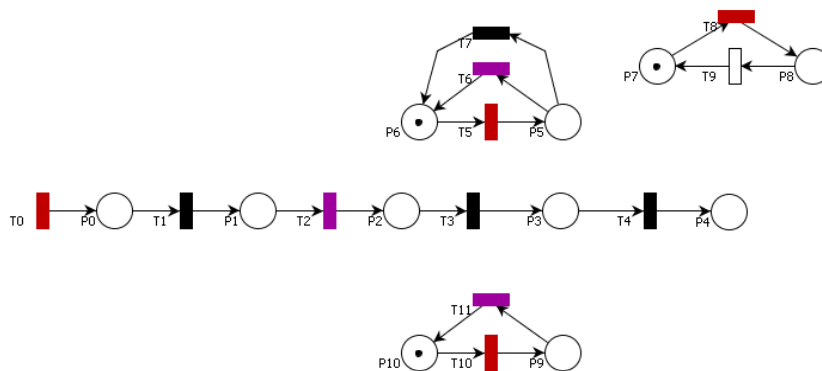


Figura 16. Red del ejemplo en el estado inicial, T0, T5 y T10 están habilitadas.

En la figura 16 se pueden observar el estado inicial de la red de petri, después de presionar la opción animación. Es posible disparar cualquier transición habilitada haciendo clic izquierdo encima de ella, inmediatamente después PIPE actualizará el marcado de la red y cualquier transición que deje de estar habilitada será mostrada en su color original. El usuario puede hacer evolucionar la red a placer, además también cuenta con las siguientes opciones.

### 1.3.1 Volver :

Vuelve la red al estado anterior. Es válido solo en modo animación, una vez se ha realizado un disparo.

### 1.3.2 Continuar :

Deshace los cambios de la opción anterior.

### 1.3.3 Disparo aleatorio :

Realiza un disparo aleatorio. Solo es válido en modo animación.

### 1.3.4 Animar :

Realiza un número determinado de disparos aleatorios. La cantidad de disparos se deben especificar en el cuadro de diálogo que se despliega.

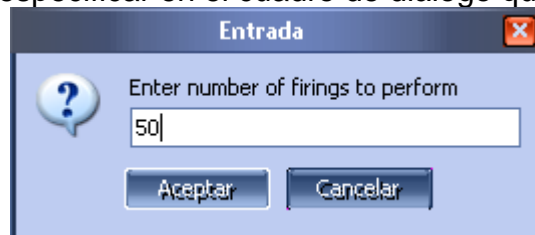


Figura 17. Número de disparos aleatorios.

A medida que el usuario realiza la animación, PIPE lleva todo el registro de las transiciones que han sido disparadas en la zona inferior izquierda de la ventana.

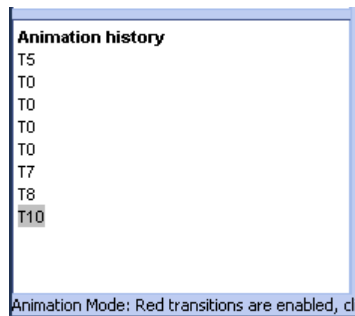


Figura 18. Historial de la animación.

Las opciones de animación también están disponibles en el menú “Animate” de PIPE, como se observa en la siguiente figura.

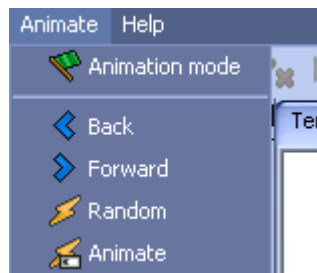


Figura 19. Menú “Animate”

Para salir de la animación se debe seleccionar la opción “Animation mode” nuevamente, una vez realizado esto PIPE devolverá la red de petri a su marcado inicial y reactivará los comandos de edición y deshabilitará los de animación.

### 1.3.5 Módulos de PIPE

Otras opciones disponibles son los módulos, en estos PIPE implementa diversas herramientas, la descripción de estos módulos y la teoría que representan se puede encontrar en la dirección <http://pipe2.sourceforge.net/docs.html>. En este manual se explicarán los módulos relacionados con la teoría trabajada en esta tesis. Estos módulos son:

- **Clasificación:** Determina si la red de petri pertenece a una o más de las siguientes clases:
  - Máquina de estados
  - Grafo marcado.
  - De libre elección
  - De libre elección extendida
  - Red simple

- Red simple extendida
- **Análisis Invariante:** Calcula todos los P-Invariantes y T-Invariantes linealmente independientes de la red.
- **Matriz de incidencia y marcado:** calcula la matriz de incidencia, pre, post y el vector de marcado de la red.
- **Sifones y trampas mínimas:** Calcula los sifones y trampas mínimas de la red.
- **Análisis de espacio de estado:** determina si la red es limitada, segura y si esta contiene algún bloqueo.

La descripción de otros módulos puede encontrarse en la dirección de arriba y la documentación de PIPE en la carpeta Docs. Para acceder a cualquier módulo, se debe seleccionar del control desplegable en la parte izquierda de PIPE.

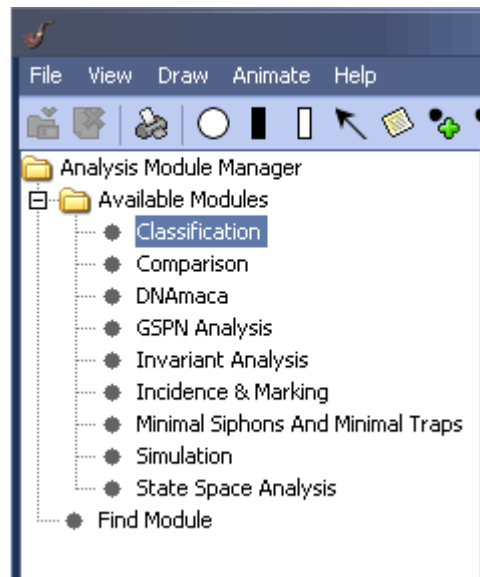


Figura 20. Módulos de la herramienta PIPE

Al presionar sobre el módulo que se desea ejecutar, se desplegará la ventana con los comandos de este, en la mayoría de los casos, solo será necesario presionar el botón "Calculate" o "Generate" para que el módulo aplique la operación deseada sobre la red actual. Además es posible aplicar los módulos a una red en un fichero XML indicando el directorio del archivo.

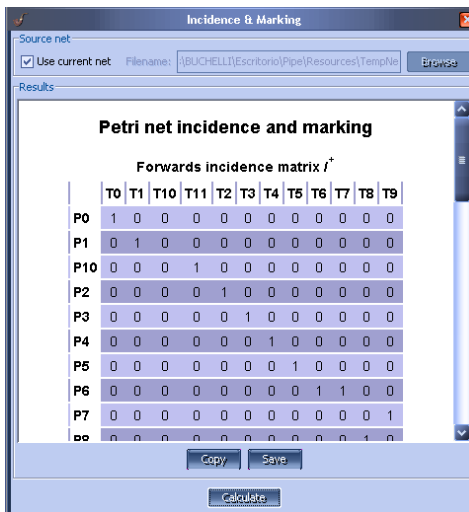


Figura 21. Módulo de incidencia y marcado aplicado a una red de petri.

Estos módulos sirven para observar las características de la red y junto a la simulación, resolver algunos problemas simples de bloqueo, en este último caso, CRP dispone de una herramienta más potente en casos más complejos que no puedan ser solucionados con una simple inspección.

Ya se ha creado el modelo de la celda de pulido y pintado de relojes en red de petri, ahora se desea crear un supervisor que haga cumplir los requerimientos para la celda. Este procedimiento será realizado en la siguiente sección mediante el uso de CRP.

#### 1.4 Uso de CRP (Control redes de petri)

En CRP es posible crear supervisores mediante la metodología de restricciones generales, como se expone en la monografía de esta tesis y genera ladder mediante la metodología Token Passing Logic, como también fue expuesto en este trabajo de grado.

CRP presenta una interfaz para cada una de estas funciones, las cuales se pueden seleccionar mediante las pestañas superiores, en la pestaña "Creación del supervisor" se muestran todos los comandos necesarios para esta función, de una manera similar, la pestaña "Generar código ladder", solo muestra los comandos suficientes para la implementación en PLC del supervisor. Cuando CRP está configurado para generar ladder, PIPE también es ocultado para comodidad.

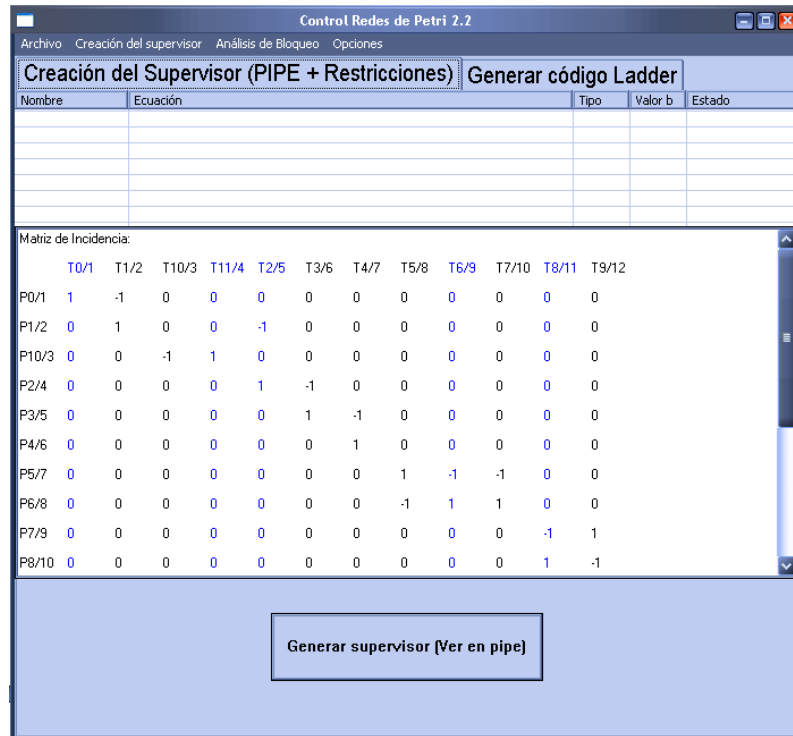


Figura 22. Interfaz para la creación del supervisor.

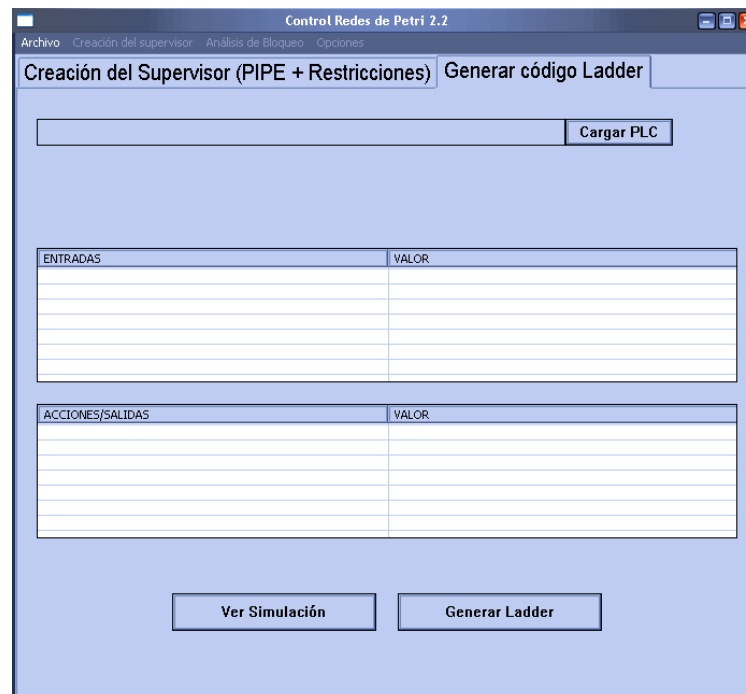


Figura 23. Interfaz para la creación de ladder.

Nótese que en la interfaz de creación del supervisor, CRP ya ha calculado la matriz de incidencia de la red de petri editada en PIPE. Esta matriz muestra

las transiciones controlables y observables en color negro, las no controlables en azul, las no observables en verde y las no controlables y no observables en rojo, fueron cambiados los colores usados en PIPE debido a que estos colores no se diferencian bien en un fondo blanco y el color azul se puede confundir en PIPE con una selección, así como el rojo en modo animación.

#### 1.4.1 Abrir, guardar y cerrar un trabajo

Usando el menú “Archivo” de CRP es posible guardar, abrir o cerrar en cualquier etapa del trabajo que el usuario está realizando.

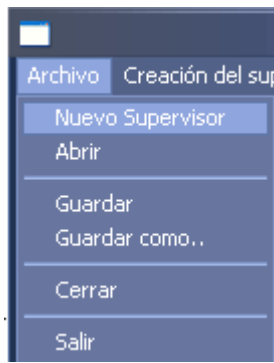


Figura 24. Menú archivo de CRP

La explicación de estos comandos se presenta a continuación:

- **Nuevo supervisor:** Lanza a PIPE 2.0 con contenido vacío y se crean toda la interfaz grafica necesaria para que el usuario empiece un nuevo trabajo. Si se está trabajando un proyecto antes de esta operación, es cerrado y liberada toda la memoria asociada.
- **Abrir:** Abre un trabajo previamente guardado en un archivo XML.
- **Guardar:** Guarda todos los datos que el usuario a insertado en la aplicación en un archivo XML. Los datos incluyen la red de petri, restricciones y entradas o salidas del PLC asignadas a entradas y acciones. Si aún no existe un fichero donde previamente se ha recuperado los datos o se hallan guardado, se debe elegir la ubicación donde será creado el fichero, una vez hecho esto, se guardará el camino para realizar la operación automáticamente.
- **Guardar como:** Realiza la misma operación anterior, pero creando siempre un nuevo archivo.
- **Cerrar:** Cierra a PIPE y toda la interfaz gráfica para trabajar en CRP.

- **Salir:** Cierra a PIPE y toda la interfaz gráfica para trabajar en CRP, y posteriormente termina la aplicación.

#### 1.4.2 Creación del supervisor para la celda de pulido y pintado de relojes.

Por ahora, en el ejemplo de este manual es necesario crear un supervisor para la celda de pulido y pintado de relojes, así es seleccionada la pestaña “Creación del Supervisor (PIPE + Restricciones)”. En esta pestaña se activan todos los menús de la aplicación (excepto “Archivo”, el cual siempre está activo), en la etapa del supervisor, se tienen que realizar, básicamente, dos operaciones:

1. Cumplir los requerimientos del proceso, esto se logra mediante la adición de restricciones que hacen que el proceso se comporte de la manera deseada.
2. Evitar algún bloqueo que pueda presentarse. Esto se logra mediante la búsqueda de sifones, si se topa con algún sifón no controlado, la red es propensa a bloquearse.

CRP separa los comandos para realizar las anteriores operaciones en dos menús diferentes: El menú “Creación del supervisor” y “Análisis de bloqueo”, los cuales se observan en la siguiente imagen.

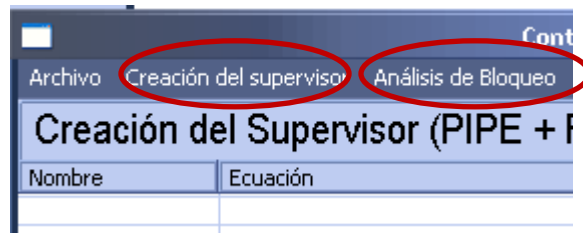


Figura 25. Menús de creación del supervisor y análisis de bloqueo

En el referido a la creación del supervisor, este será creado mediante la adición de restricciones  $Lu + Hq + Cv \leq b$  o  $Lu + Hq + Cv \geq b$ , según la metodología de restricciones generales.



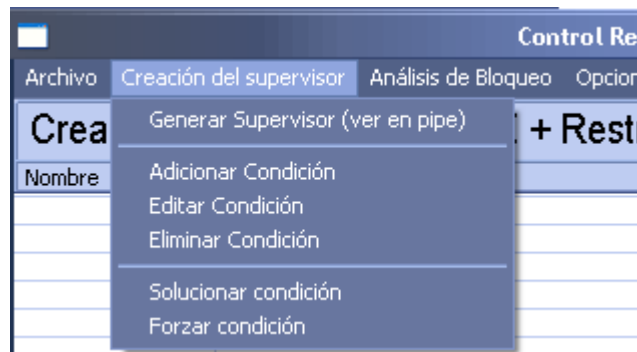


Figura 26. Menú creación del supervisor.

Para agregar una restricción, se debe seleccionar la opción “Adicionar restricción”, en el menú “Creación del supervisor”, inmediatamente se desplegará la siguiente ventana.

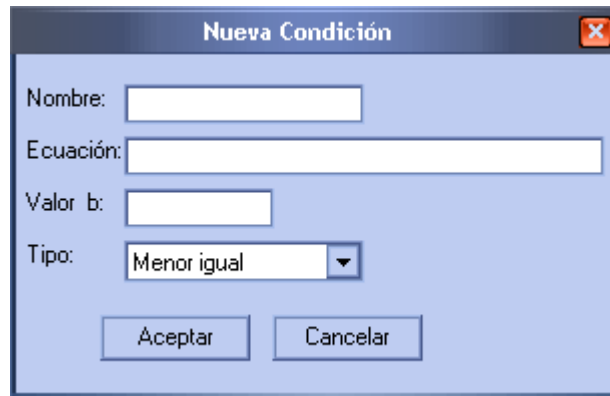


Figura 27. Ventana de adición de restricción.

En esta ventana se debe agregar un nombre a la restricción, no se permitirá agregar una restricción sin nombre. En el campo “Ecuación” se debe escribir la ecuación de la restricción, esta ecuación se escribe haciendo referencia al vector de marcado, el vector de disparo o el vector de Parikh de los distintos elementos, se usa el carácter  $u$  para referenciar el vector de marcado, el carácter  $q$  para referenciar el vector de disparo y el carácter  $v$  para referenciar el vector de Parikh. Después del carácter *clave*, se debe escribir el nombre del elemento al cual se está referenciando.

Opcionalmente puede ir un coeficiente entero antes del carácter clave, este valor se multiplica por el elemento, en caso de no existir un coeficiente, se supone que este es uno. La forma de escribir no difiere de manera significativa con respecto a una ecuación cualquiera, como ejemplo a lector considere la siguiente restricción que se desea imponer, la red de petri es el ejemplo que se está trabajando, la celda de pulido y pintado de relojes.

$$u_0 - u_1 + 2u_5 + 3q_2 + 2q_5 + 2v_2 - 10v_9 \leq 8$$

Esta restricción se usa solo como propósito ilustrativo de la sintaxis, no se pretende cumplir ninguna especificación con ella y no es una de las restricciones que comprende el supervisor para este ejemplo. Para escribir esta restricción, se debería escribir en la ventana de “Nueva Condición” el siguiente texto:

$$uP0 - uP1 + 2uP5 + 3qT2 + 2qT5 + 2vT2 - 10vT9$$

Además, en el campo “Valor b” se debe colocar el valor 8 y seleccionar “Menor igual” en la pestaña “Tipo”, como se observa en la siguiente figura.

Figura 28. Edición de la restricción de ejemplo.

Se sugiere al lector analizar la anterior parte, en caso de dudas, antes de continuar con la lectura de este manual.

Al presionar el botón “Aceptar”, se agrega la restricción, si hay algún error en la sintaxis, será indicado. El botón “Cancelar” cancela la operación y cierra la ventana. A medida que las restricciones son agregadas, estas se van escribiendo en la lista superior de CRP, como se observa en la siguiente figura.

Nombre	Ecuación	Tipo	Valor b	Estado
Cap Maq 3	uP13+uP14	<=	1	<Inadmisible>
Cap Maq 4	uP21+uP22	<=	1	<Inadmisible>
Cap Buff C	uP17	<=	1	<Inadmisible>
Seq A 1	vT24-vT19	<=	0	<Admisible>
Seq A 2	vT19-vT10	<=	0	<Admisible>
Seq A 3	vT10-vT7	<=	0	<Admisible>

Figura 29. Lista de restricciones.

En esta lista se muestran los atributos editados en la ventana “Nueva Condición”, además se calcula la admisibilidad de estas restricciones en el campo estado. En estado pueden hallarse los siguientes valores:

- **Inadmisible:** La restricción no es admisible, debe resolverse por algún método o eliminarse, en caso extremo, ser *forzada*. En las siguientes hojas se extenderá sobre el tratamiento de este tipo de restricciones.
- **Admisible:** La restricción es admisible.
- **Inválida:** El usuario ha eliminado uno o más elementos en la red de petri que ha hecho que esta restricción no tenga sentido.
- **Nolicial:** El usuario ha cambiado el marcado inicial de la red de petri que ha hecho esta restricción sea imposible de reforzar y por tanto inválida.
- **Forzada:** Esta es una restricción antes no admisible, para la cual se hace una excepción en el criterio de admisibilidad y es implementada pese a cualquier error. En las siguientes hojas se extenderá sobre el tratamiento de restricciones inadmisibles.

Una vez las restricciones están en la lista superior, hay cuatro operaciones que se pueden realizar sobre estas, como lo indica el menú “Creación del supervisor”, en la figura 26. Estas operaciones se describen a continuación. Para aplicar una operación a una restricción, primero es necesario seleccionarla de la lista superior, de tal manera que queda resaltada en el color de selección, como lo indica la figura 30.

Nombre	Ecuación	Tipo	Valor b	Estado
Cap Maq 3	uP13+uP14	<=	1	<Inadmisible>
Cap Maq 4	uP21+uP22	<=	1	<Inadmisible>
Cap Buff C	uP17	<=	1	<Inadmisible>
Seq A 1	vT24-vT19	<=	0	<Admisible>
Seq A 2	vT19-vT10	<=	0	<Admisible>
Seq A 3	vT10-vT7	<=	0	<Admisible>

Figura 30. Selección de una restricción, en este caso la llamada “Cap Buff C”

- **Editar condición:** Lanza nuevamente la ventana de la figura 27 con los atributos de la restricción para modificarlos. Al presionar aceptar se remplazan los antiguos atributos por los nuevos.

- **Eliminar condición:** Elimina esta restricción de la lista superior.

Las siguientes operaciones son válidas solo para restricciones no admisibles.

- **Solucionar condición: (Solo para restricciones no admisibles)**  
Aplica los métodos de solución seleccionados para buscar soluciones a esta restricción. en caso de encontrar al menos una, se desplegará una ventana donde se puede elegir una restricción para remplazar la original.

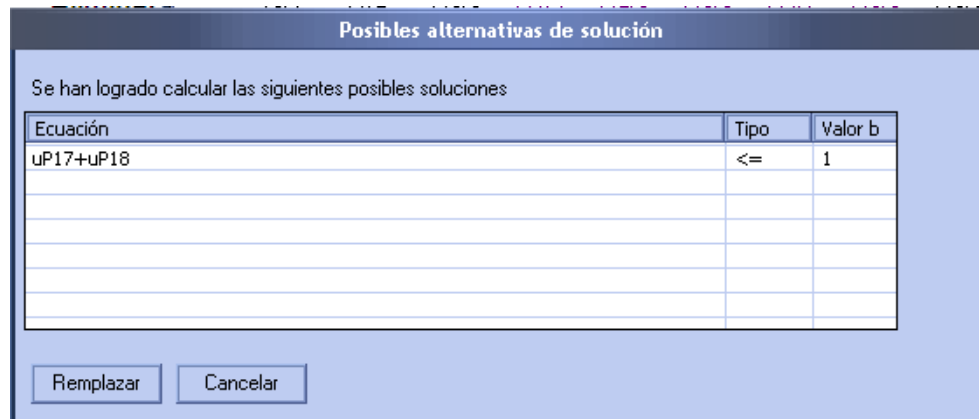


Figura 31. Ventana de visualización de soluciones

CRP dispone de tres métodos para solucionar restricciones no admisibles, el método del núcleo, operaciones por fila de matriz y programación lineal entera. Estos métodos se pueden activar en el menú “Opciones” y la opción “Opciones del solver”, al realizar esto se desplegará la siguiente ventana.

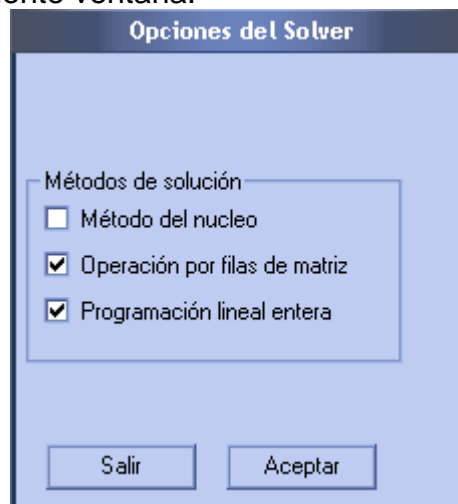


Figura 31. Opción por defecto para los métodos de solución.

Cada vez que se lleve cabo la operación “Solucionar Condición”, se aplicarán todos los métodos cuya casilla de verificación este seleccionada.

- **Forzar condición:** cambia la restricción de un estado inadmisibles a un estado forzado, la aplicación hará una excepción al criterio de admisibilidad e implementará esta restricción pese a cualquier error en el supervisor. Cuando el usuario pida esta operación sobre una restricción, se lanzara un mensaje confirmando proceder con ella.

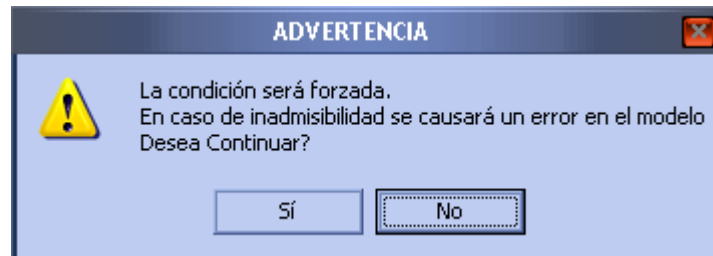


Figura 32. Confirmar operación de forzado.

En caso de seleccionar “Sí” se procederá con la operación, en caso contrario se abortará.

Una vez se han agregado todas las restricciones y todas estas sean admisibles o forzadas, es posible implementar el supervisor. Esto se puede realizar mediante el comando “Generar supervisor (ver en pipe)” del menú “Creación del supervisor” o del botón inferior del mismo nombre.

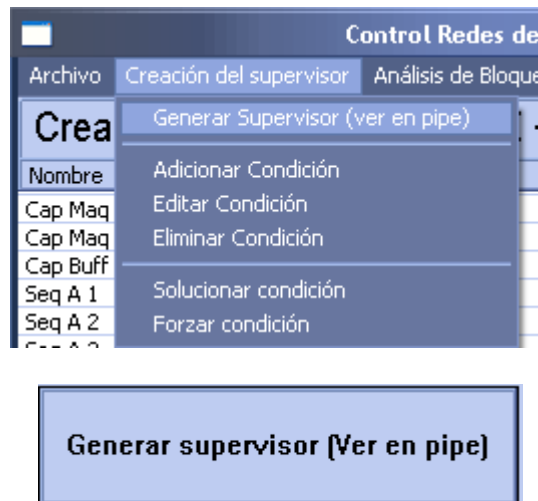


Figura 33. Comandos para la implementación del supervisor

Si la operación tiene éxito, inmediatamente después se mostrará en otra pestaña de PIPE el modelo de la red con las restricciones impuestas.

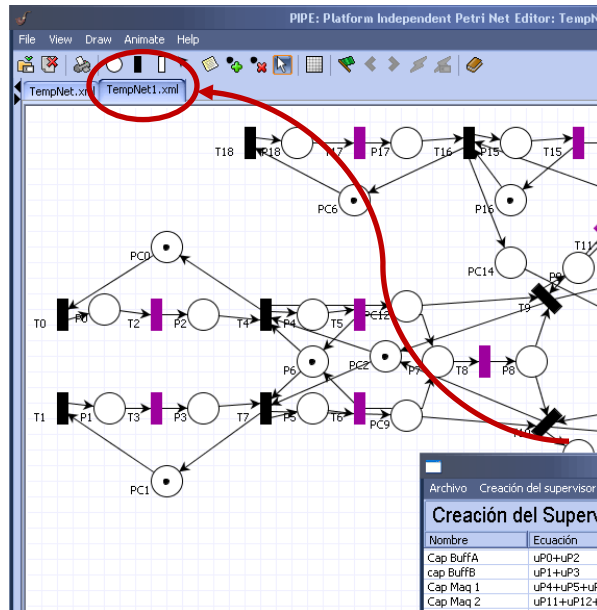
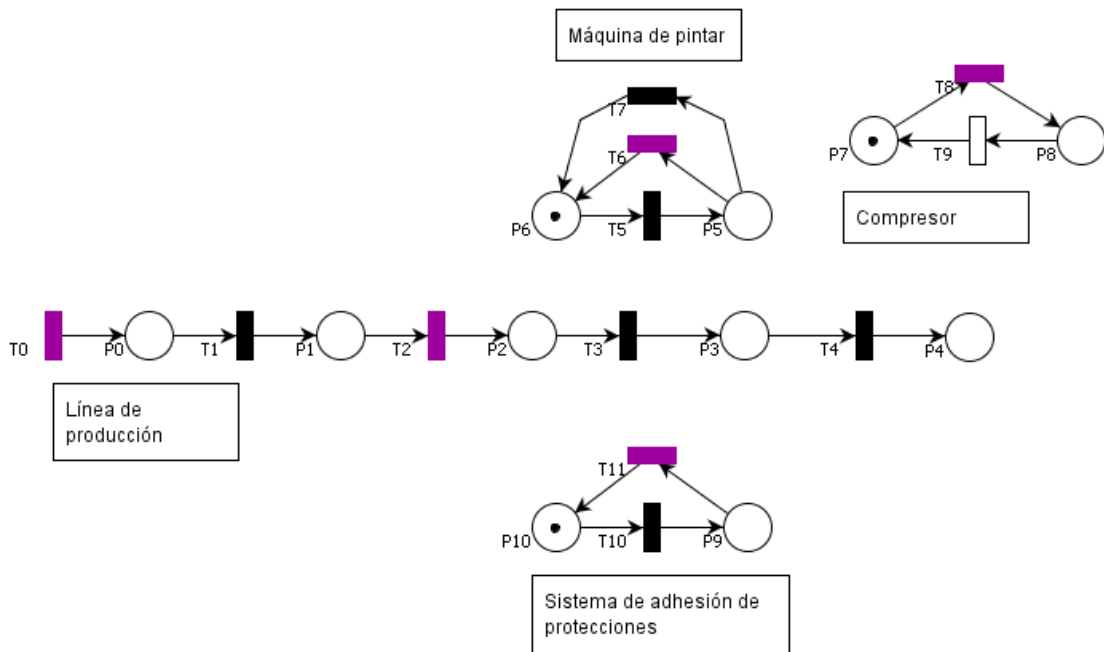


Figura 34. Nueva pestaña de PIPE con la red supervisada. Explicado la parte de creación de supervisor usando CRP, se procede a realizar el supervisor para el ejemplo de este manual, para esto, se lista los requerimientos del proceso.



Los requerimientos que fueron descritos para este proceso son:

1. La máquina de pulido solo puede trabajar con una pieza a la vez.
2. El buffer de espera a la zona de pintado alberga máximo tres relojes.
3. La zona de pintado puede albergar un solo reloj.
4. El pintado y colocación de las etiquetas protectoras, se pueden llevar a cabo cuando hay un reloj en la zona de pintado, y antes de pintar, se deben colocar las etiquetas.
5. En caso que el compresor se quede sin aire, se debe apagar la máquina de pintado y esperar a que se recargue, la operación debe reiniciarse después de esto.
6. Obviamente, se debe retirar el reloj después de terminadas todas las operaciones.

En el caso del primer requerimiento, la restricción  $u_1 \leq 1$  sobre el modelo será suficiente para cumplirlo. En el caso del segundo, la restricción  $u_2 \leq 3$ . En el caso del tercero, la restricción  $u_3 \leq 1$  cumplirá con el requisito.

En el cuarto requerimiento, se tienen dos requisitos, para el primero, que ese coloquen las etiquetas e inicie la máquina de pintar solo cuando haya un reloj en la zona de pintado, se puede cumplir con la siguiente restricción:

$$-u_3 + q_5 + q_{10} \leq 0$$

El segundo requisito, realizar la colocación de la etiquetas protectoras antes, se puede lograr con la siguiente restricción.

$$v_5 - v_{11} - v_7 \leq 0$$

Para el quinto requisito, se puede lograr con dos restricciones:

La máquina se apaga:

$$-u_8 + q_7 \leq 0$$

La maquina reinicia la operación:

$$-u_7 + q_5 \leq 0$$

El último requisito, retirar el reloj, se logra con las siguientes restricciones.

$$\begin{aligned} 2v_4 - v_6 - v_{11} &\leq 0 \\ v_{10} - v_4 &\leq 1 \end{aligned}$$

Ahora se adiciona estas restricciones a la lista superior de CRP, para ser impuestas al proceso.

Nombre	Ecuación	Tipo	Valor b	Estado
Requisito uno	$uP1$	$\leq$	1	<Admisible>
Requisito dos	$uP2$	$\leq$	3	<Inadmisible>
Requisito tres A	$-uP3+qT5+qT10$	$\leq$	0	<Admisible>
Requisito tres B	$vT5-vT11-vT7$	$\leq$	0	<Admisible>
Requisito cuatro A	$-uP8+qT7$	$\leq$	0	<Admisible>
Requisito cuatro B	$-uP7+qT5$	$\leq$	0	<Admisible>
Requisito quinto A	$2vT4-vT6-vT11$	$\leq$	0	<Admisible>

Figura 35. Todas las restricciones adicionadas a la lista

Como se puede observar, solo la condición dos resulta inadmisibile, en este caso es una restricción que cualquier método de solución puede solucionar de manera sencilla. Al aplicar la operación “Solucionar condición”, se halla la siguiente solución a la restricción, la cual es usada para remplazar la original.

Se han logrado calcular las siguientes posibles soluciones

Ecuación	Tipo	Valor b
$uP1+uP2$	$\leq$	3

Remplazar    Cancelar

Figura 36. Solución a la restricción inadmisibile.

Al realizar la implementación del supervisor en CRP, la red supervisada que se obtiene en PIPE, es el de la figura 37.



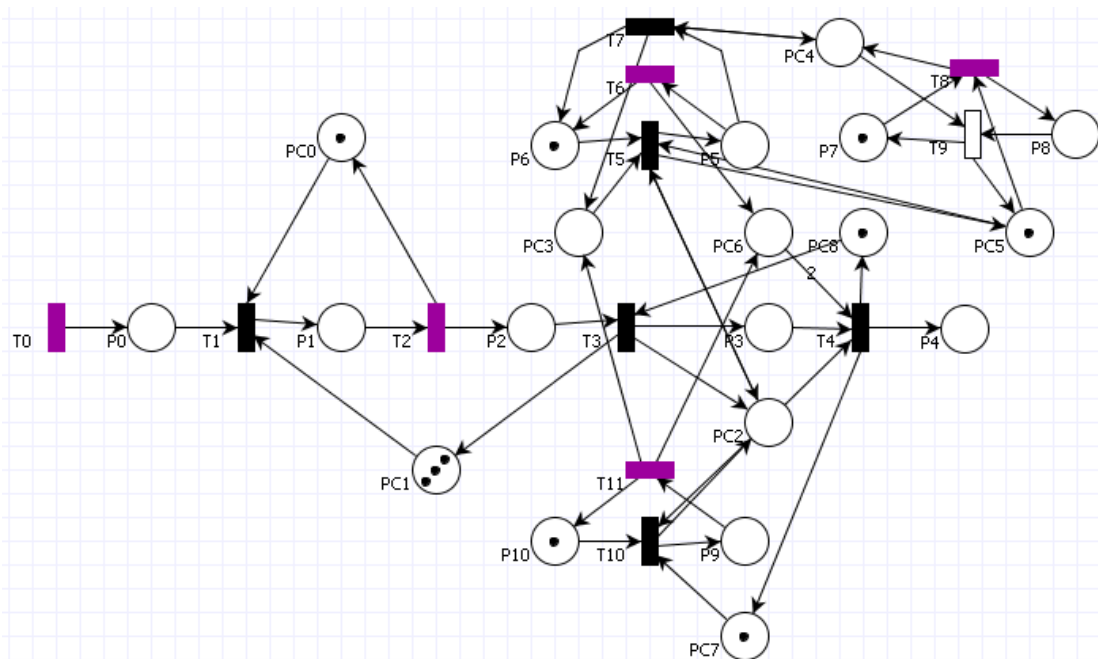


Figura 37. Red supervisada para el ejemplo de pulido y pintado de relojes.

Si el lector ha realizado el modelo de la red correctamente y ha insertado las restricciones correctas, puede ver mediante una sencilla simulación en PIPE que la red supervisada cumple con los requerimientos para el proceso. El modelo en PIPE para la red supervisada es temporal, este puede ser guardado o cerrado mediante las opciones “Save as” (Guardar como) y “Close” (Cerrar) que se activan en PIPE.

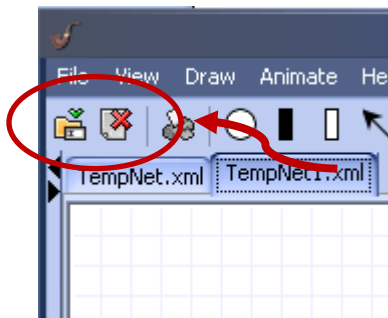


Figura 38. Guardar o cerrar el modelo supervisado

Si no se guarda este modelo se borrará, será necesario generar el supervisor nuevamente en CRP para crear otro modelo, si se necesita recuperarlo.

### 1.4.3 Análisis de bloqueo de la red.

La prevención del bloqueo es un tema importante en la creación de supervisores mediante sistemas a eventos discretos. El bloqueo se presenta

cuando algunas transiciones o todas dejan de estar habilitadas para cualquier marca alcanzable desde la actual, es decir, a partir del marcado actual no es posible que estas vuelvan a disparar. La detección de estos casos se realiza mediante la búsqueda de sifones, un sifón es un conjunto de lugares, el cual al perder todas sus marcas ya no vuelve a estar marcado, por tanto las transiciones con arcos de entrada desde estos lugares quedan inhabilitadas permanentemente desde ese marcado. Es vital asegurar que todos los sifones no puedan perder todas sus marcas.

Para este análisis, la primera herramienta que se recomienda usar son los módulos de sifones y espacio de estados de PIPE, así como la simulación en este software. El primero calcula los sifones y trampas presentes en la red, de esta manera se puede realizar una primera idea del estado de estos elementos en la red de petri, cuáles de estos pueden provocar un bloqueo y cuáles están controlados.

La segunda herramienta recomendada es el módulo de espacio de estados, este módulo permite conocer propiedades que ayudan a determinar la posibilidad de bloqueo de la red, estas son:

- Calcula si la red es acotada
- Calcula si la red es segura
- Calcula si existe un *bloqueo*.

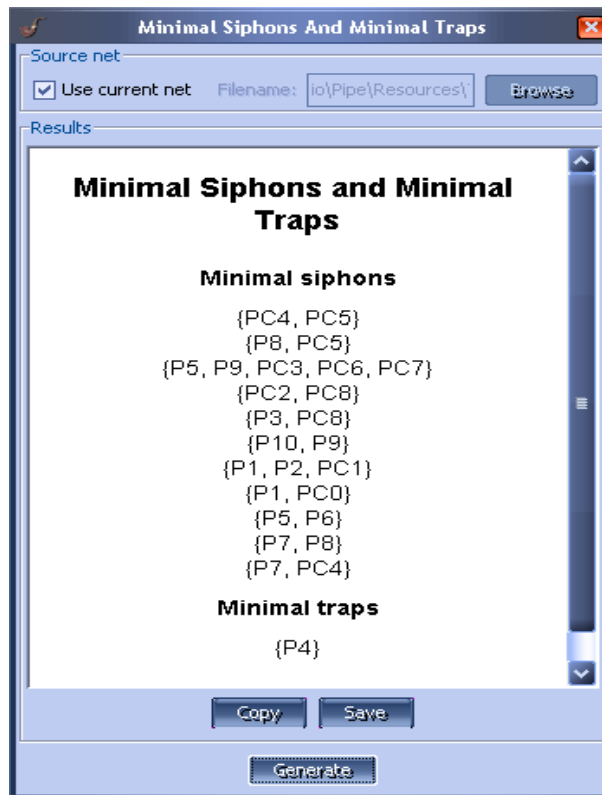


Figura 39. Módulo de sifones aplicado al modelo supervisado.

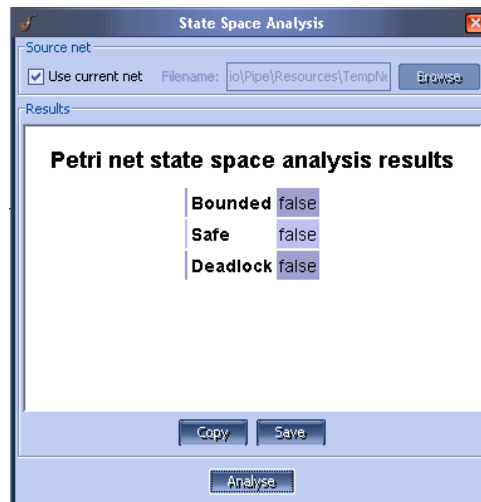


Figura 40. Módulo de espacio de estados aplicado al modelo supervisado.

La dificultad con este módulo, es que pueda que estas propiedades puedan no ser suficientes para determinar si la red sufre un bloqueo, este módulo considera que existe un bloqueo cuando todas las transiciones de la red están inhabilitadas, sin embargo, en muchos casos se buscará resolver un

problema de bloqueo *parcial*, esto es solo algunas transiciones de la red quedan inhabilitadas.

Aún es posible realizar una búsqueda del bloqueo mediante la simulación automática en PIPE, este método es muy efectivo en la mayoría de los casos

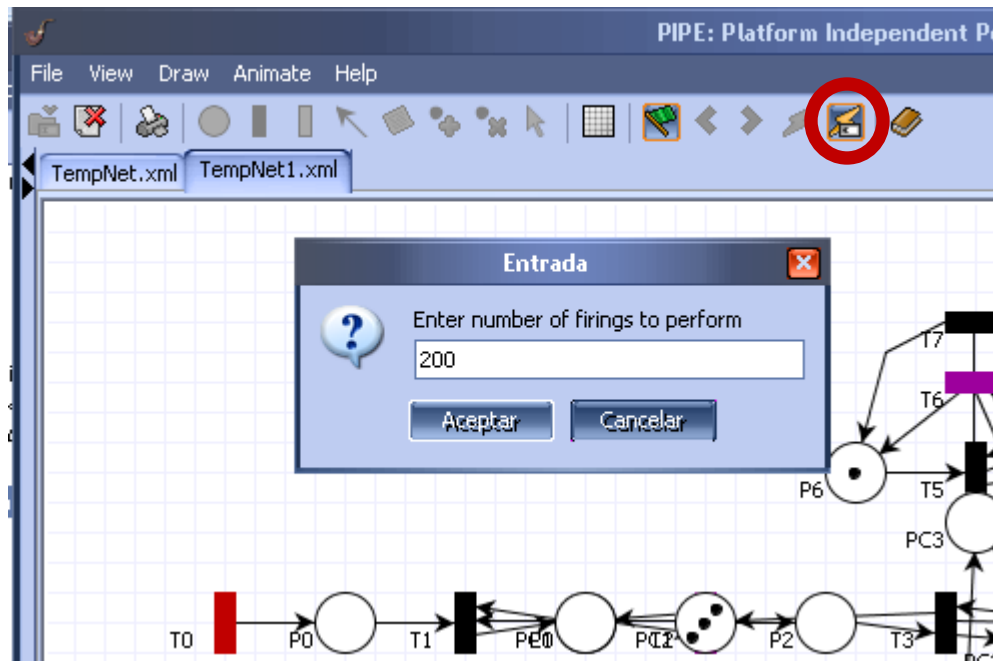


Figura 41. Simulación automática de 200 disparos en la red supervisada

Sin embargo, hasta este método puede fallar en la búsqueda del bloqueo, y en caso de hallar un bloqueo, resulta difícil localizar cuando y donde este ocurrió, información vital para resolver el problema.

Cuando las herramientas de PIPE fallan o son insuficientes, debería utilizarse la funcionalidad de análisis de bloqueo de CRP. Este análisis está basado en la búsqueda del bloqueo mediante la búsqueda de sifones. La ventaja de este análisis es que además de buscar sifones, determina si estos están controlados por alguna marca, invariante, ambos o no están controlados del todo. De esta manera, localizando los sifones no controlados, es posible detectar cualquier bloqueo de la red, de esta manera es posible resolver el bloqueo.

Sin embargo, la funcionalidad de CRP también tiene la desventaja en el gasto computacional necesario, que en redes complejas puede llegar a ser considerable. Por el motivo anterior, se recomienda usar este módulo, cuando las herramientas de PIPE han fallado, el problema no es simple de resolver o la red no es muy compleja. A continuación se describe las opciones que se disponen para el análisis de bloqueo.

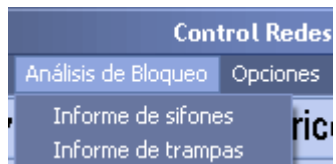


Figura 42. Opciones del menú análisis de bloqueo.

- **Informe de sifones:** Calcula todos los sifones de la red de petri, y busca si estos están controlados por alguna trampa o invariante. En caso contrario indica que cuales no están controlados.
- **Informe de trampas:** Calcula todas las trampas de la red e indica cuales están marcadas y cuáles no.

El problema en este manual no contiene un bloqueo. Como el lector puede comprobarlo mediante PIPE o CRP, un caso de bloqueo se trabaja en los casos de estudio de esta tesis.

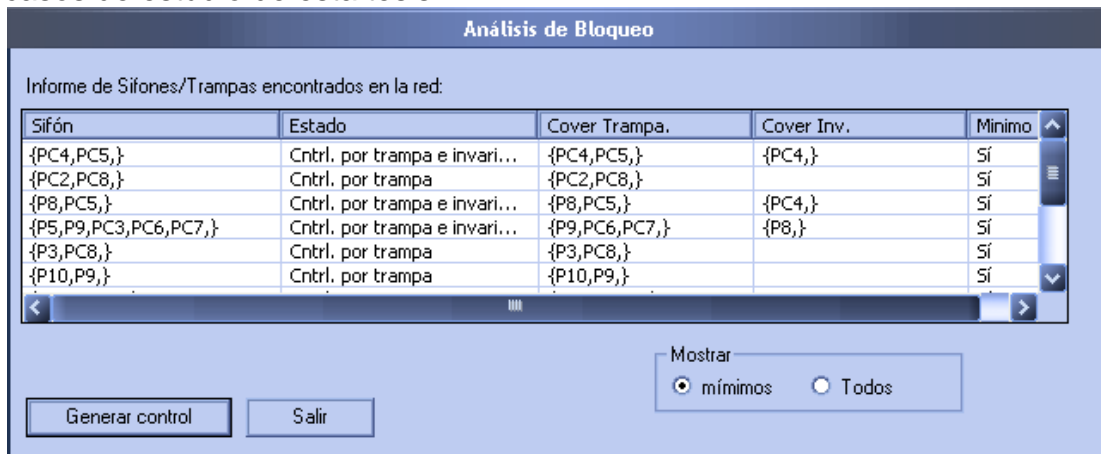


Figura 43. Celda de relojes con los tres sifones mínimos controlados

#### 1.4.4 Generación de código ladder para la celda de pulido y pintado de relojes

CRP puede generar código ladder desde cualquier red de petri creada en PIPE. En la metodología usada por CRP, el usuario no debe preocuparse por asignar variables al ladder, esto es realizado automáticamente por la aplicación, solo es precisa la asignación de los las entradas y salidas que se han asignado a la red.

Las entradas son asignadas a las transiciones, cuando una transición tiene asignada una entrada, esta se supone que se puede disparar únicamente cuando la entrada está activa, alternativas a estas opciones son las banderas

“activar una vez” e “Invertir o negar”, al activar estas opciones se realizan los siguientes cambios:

- **Activar una vez:** La transición se dispara solo una vez cuando la señal esta activa. Esta opción es útil solo cuando la transición contiene suficientes marcas para dispararse múltiples veces cuando alguna señal se activa, de otra manera es innecesaria. Para que la transición pueda volver a dispararse, es necesario que la señal se apague y vuelva a activarse.
- **Invertir o negar:** Se invierte el significado de la señal, es decir, la transición puede dispararse solo si la señal está apagada.

#### 1.4.4.1 Adición de entradas y salidas para la celda de pulido y pintado de relojes.

La adición de las entradas en PIPE ya fue explicada en la primera sección de este anexo, esta operación se realiza mediante un nombre que el usuario desee darle a cada entrada, se recomienda no utilizar un valor de una entrada real del PLC, sino uno que indique el nombre o propósito de esta. En el ejemplo que se ha trabajado en este anexo, se deben tener en cuenta de los siguientes sucesos.

1. Un reloj llego al buffer de entrada. Esto se realiza mediante la adición de la entrada es asignada la transición T0, se edita esta entrada con el nombre “LlegadaReloj”, como se observa en la figura.

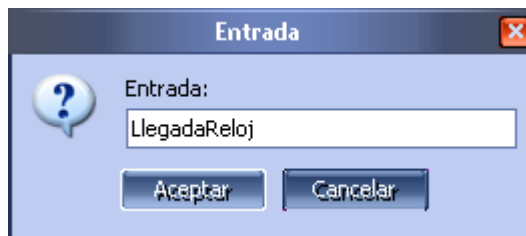


Figura 44. Edición de la entrada LlegadaReloj

Para esta entrada es necesario activar la opción “ActivarUnaVez”, ya que el otro proceso advertirá la llegada de un reloj mediante un pulso hacia el PLC.

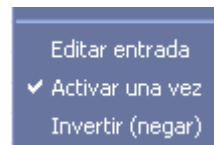


Figura 45. Opción “Activar una vez” para esta opción

2. La máquina de pulido terminó de procesar un reloj. Esto se realiza mediante la adición de la entrada “PulidoTerminado” a la transición T2.

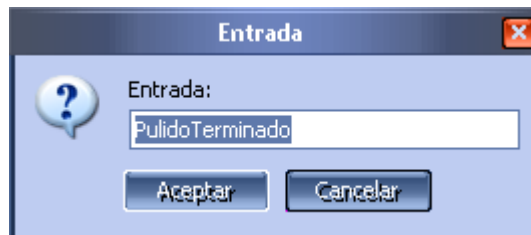


Figura 46. Edición de la entrada “PulidoTerminado”

3. El sistema automático termino de colocar las etiquetas. Para esto se adiciona la entrada “EtiquetasColocadas” a la transición T11.

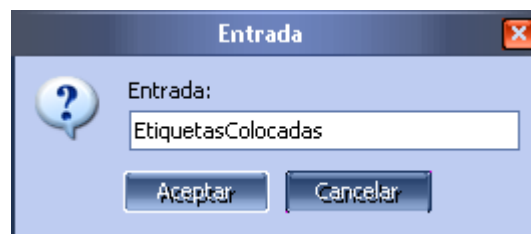


Figura 47. Edición de la entrada “EtiquetasColocadas”

4. La máquina de pintar terminó de pintar un reloj. Para esto se adiciona la entrada “RelojPintado” a la transición T6.

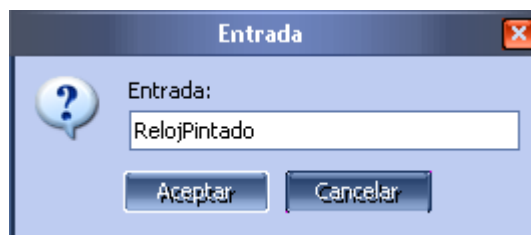


Figura 48. Edición de la entrada “RelojPintado”

5. El compresor se quedo sin aire. Para esto se adiciona la entrada “CompresorSinAire” a la transición T8. Para esta última entrada también se activa la opción activar una vez

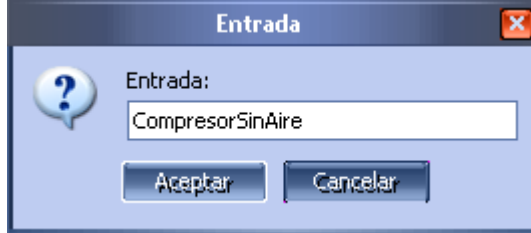


Figura 49. Edición de la entrada “CompresorSinAire”

Asignadas las entradas, solo falta la asignación de salidas, esto se hace mediante los lugares de la red de petri. Si algún lugar contiene una salida, esta se activará si el lugar tiene al menos una marca, ya que es posible asignar una misma salida a varios lugares, esta se activará si al menos uno de ellos contiene al menos una marca. En este manual se utiliza el término acción como sinónimo de salida frecuentemente. En el ejemplo se necesitan activar o desactivar los siguientes elementos:

1. La máquina de pulido. Para esto se adicionará la acción “ActivarMaquinaPulido” en el lugar P1 de la red de petri.

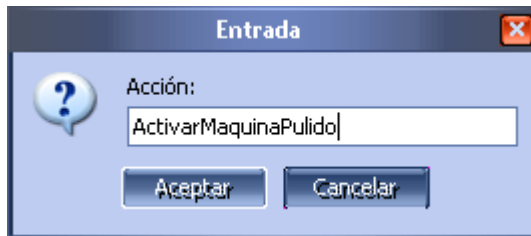


Figura 50. Edición de la acción “ActivarMaquinaPulido”

2. El sistema de fijación. Para esto se adicionará la acción “ActivarSistemaFijacion” en el lugar P3 de la red de petri.

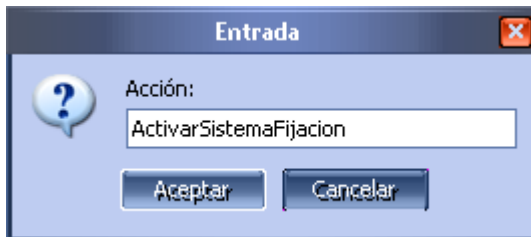


Figura 51. Edición de la acción “ActivarSistemaFijacion”

3. La máquina de pintar. Para esto se adicionará la acción “ActivarMaquinaPintar” en el lugar P5 de la red de petri.





Figura 52. Edición de la acción "ActivarMaquinaPintado"

4. El sistema de adhesión de protecciones. Para esto se adicionará la acción "ActivarSistemaAdhesion" en el lugar P9 de la red de petri.

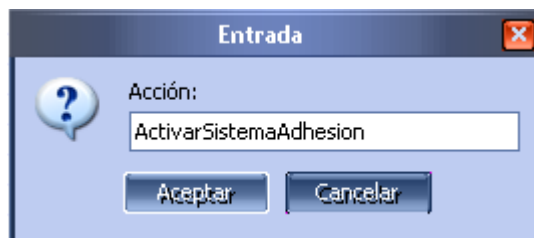


Figura 53. Edición de la acción "ActivarSistemaAdhesion"

5. El compresor de aire. Para esto se adicionará la acción "ActivarCompresor" en el lugar P8 de la red de petri.



Figura 54. Edición de la acción "ActivarCompresor"

Una vez terminado la asignación de entradas y salidas, estas se pueden observar en la interfaz de la pestaña "Generar código ladder" de la aplicación CRP. Desde aquí, es posible acceder a dos opciones, la simulación y la generación de ladder. A estas opciones se puede acceder mediante los botones inferiores de la ventana.



Figura 55. Interfaz para la generación de código ladder con todas las entradas y salidas asignadas.

#### 1.4.4.2 Simulación del supervisor en CRP

Antes de generar ladder para el ejemplo trabajado en este capítulo, se desea realizar una simulación antes de descargarlo en un PLC, la simulación realizada en CRP no simula como tal la red de petri, como lo hace PIPE, en cambio simula el código ladder generado a partir de la red en un PLC virtual, este dispositivo virtual se configura con un tiempo de escaneo de 200 milisegundos, tiempo ideal para que el usuario pueda ver la evolución del programa y este no se retrase demasiado de una implementación real. CRP lee la memoria del PLC virtual, y a partir de esta crea un bosquejo del

marcado de la red de petri para el usuario pueda visualizarla. Para el PLC virtual se crearan todas las entradas y salidas que fueron editadas en PIPE.

Simulando en CRP, es posible detectar errores previos antes de generar el código ladder, esto ahorra mucho tiempo de diseño, ya que de otra forma sería necesario generar el fichero de ladder y usar la simulación en Rockwell cada vez que alguna cosa salga mal, esto consumiría mucho tiempo al usuario. Sin embargo, una última etapa de simulación en Rockwell usando el dispositivo a usar es recomendada, y la simulación en CRP no puede tomarse como un remplazo a esta última fase. Antes de realizar una simulación, se puede elegir la red desde la cual el código ladder es generado: la original o la red supervisada.



Figura 56. Elección de la red para simular

Para el ejemplo de la celda de pulido y pintado de relojes, la interfaz de simulación se puede ver en la figura 54, esta corresponde a la red supervisada. En esta ventana se observan las entradas asignadas en la red de petri en la parte inferior izquierda y las acciones en la parte inferior derecha, en la parte central se observa el bosquejo de la red de petri para la simulación. Verificando la casilla en la parte izquierda de cada entrada, se puede simular la activación de esta, de manera análoga se puede ver que salidas ha activado el PLC virtual mediante el campo estado de la lista acción/salida.

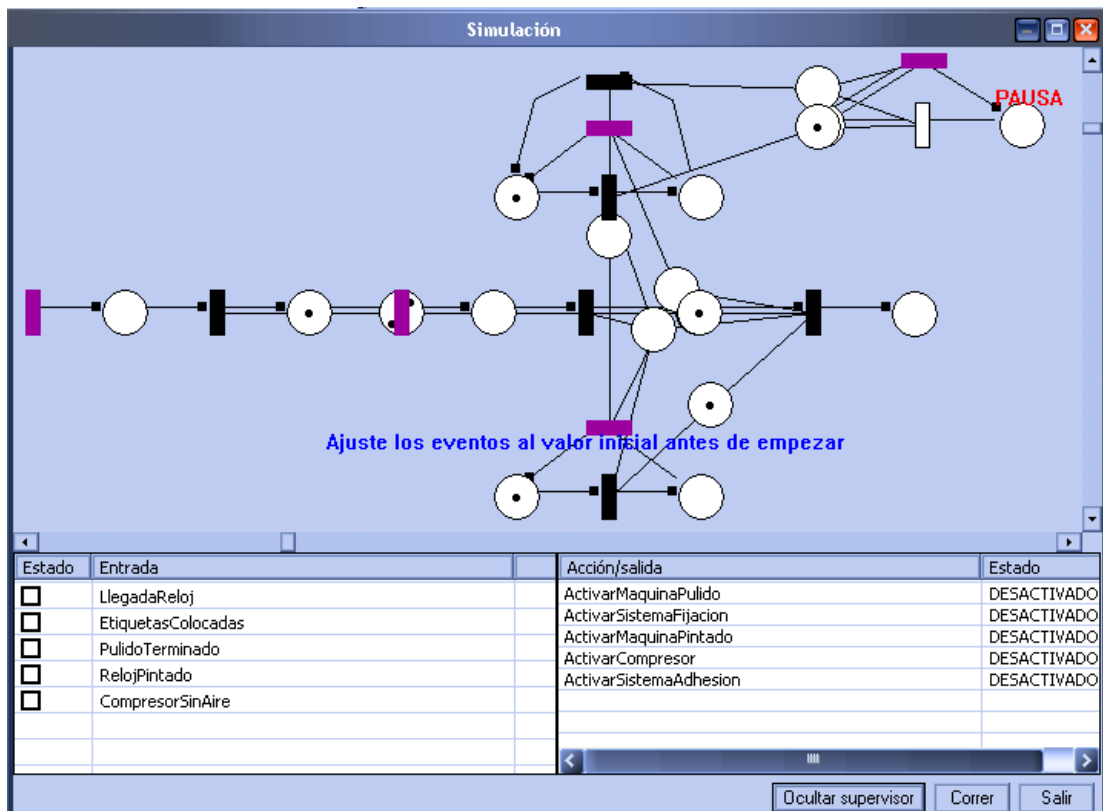


Figura 57. Ventana de simulación del código ladder.

Esta simulación puede ser pausada en cualquier momento. Adicionalmente es posible ocultar los lugares de control, ya que esta información no es de mucha utilidad. Estas opciones están disponibles en los botones inferiores de la ventana, así como la opción salir.

Por extensión, en este manual, se simulará la llegada de un único reloj a la celda hasta que este se encuentre en el buffer de salida, el lector podrá simular varias alternativas a este camino. El supervisor será ocultado por comodidad.

Primero, es simulado que un reloj llega a la celda, cuando esto sucede, es enviada al PLC una señal por la entrada denotada nombrada "LlegadaReloj", se simula esto verificando la casilla de esta entrada. En seguida se muestra el nuevo la reacción del PLC ante este suceso.

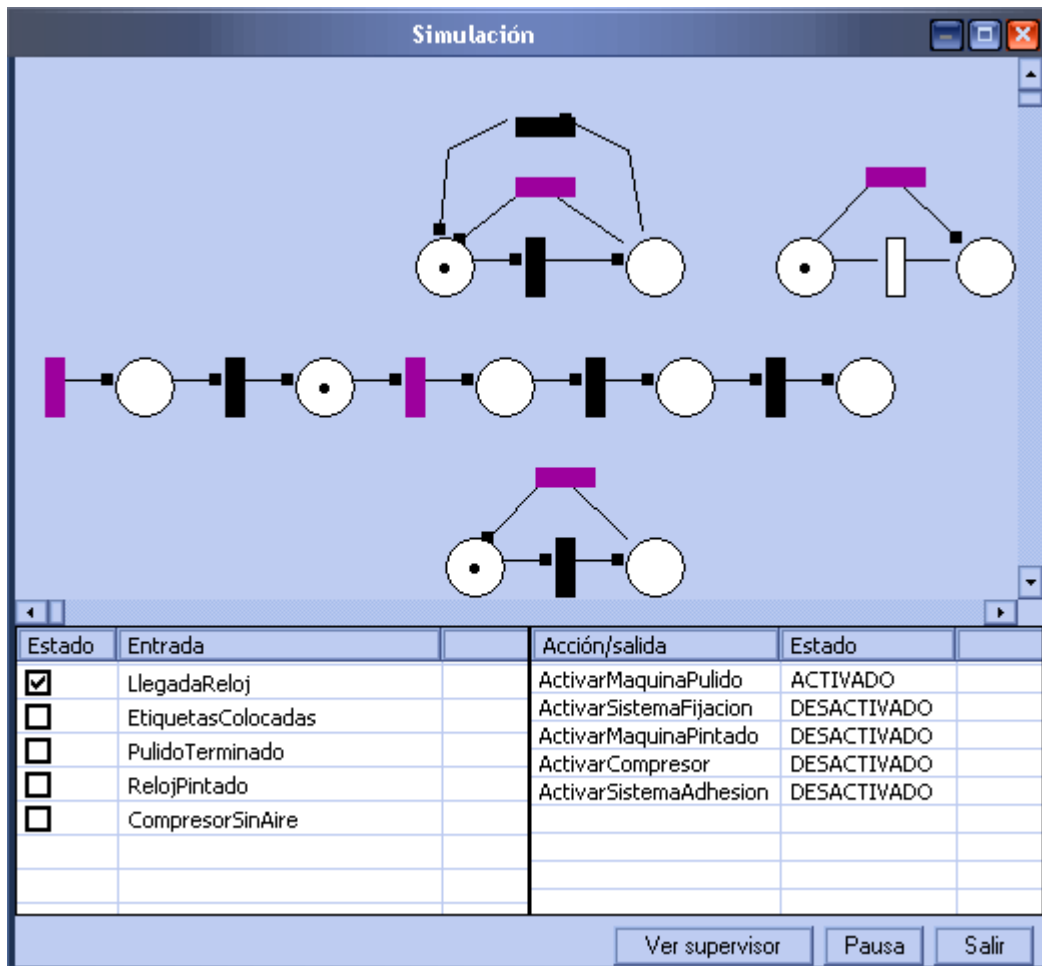


Figura 58. Llegada de un reloj a la celda.

Se observa que una vez ha llegado el reloj, el PLC ha activado la entrada “ActivarMaquinaPulido”, la cual enciende la máquina de pulido, esto sucede ya que no existe ningún reloj en el buffer de espera, por tanto es posible empezar de inmediato con esta operación. El PLC ahora espera la señal de la máquina que indique la terminación de operación de pulido.

Una vez la máquina termino la operación, mandará una señal que el PLC leerá mediante la entrada “PulidoTerminado”, para continuar con la simulación se simula esta señal con la casilla correspondiente. En la figura 59 se muestran las acciones que el PLC ha tomado a este suceso, se ha desactivado la máquina de pulido y el sistema de fijación ha sido activado, pues no existía un reloj en la zona de pintado que impidiera realizar esta acción, al realizar esto, el sistema de adhesión de etiquetas también fue activado, ahora el PLC está a la espera de la señal de terminación de este sistema.

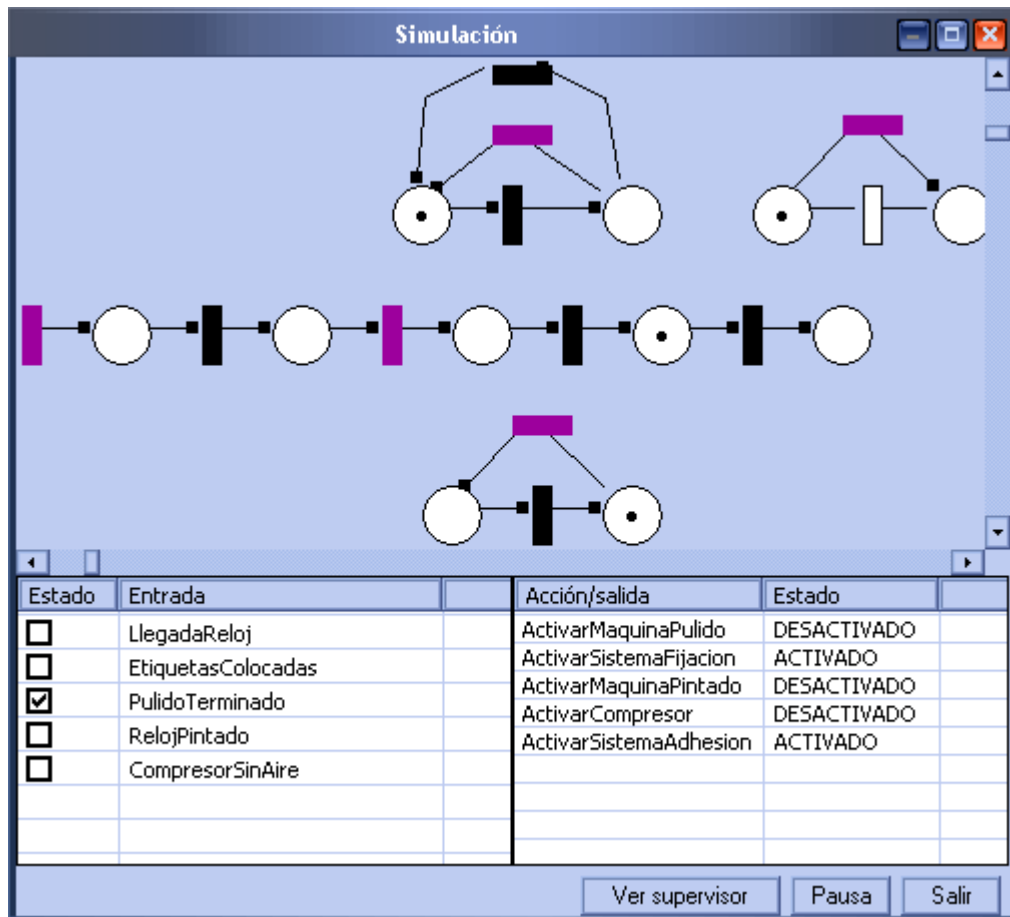


Figura 59. Terminación de la operación de pulido.

En seguida es simulada la señal de terminación del sistema de adhesión de protecciones, esta señal es leída por la entrada “EtiquetasColocadas” del PLC, este reacciona desactivando el sistema de adhesión de protecciones y activando la máquina de pintado, como se observa en la figura 60. Cuando la máquina de pintado esta activa, el compresor pudiese quedarse sin aire en cualquier momento, cuando esto sucede el compresor envía un señal que el PLC puede leer mediante la entrada “CompresorSinAire”, cuando esto ocurre el PLC apaga temporalmente la máquina de pintar y enciende el compresor, esto se puede ver en la figura 61. Ocurridos los 5 segundos, que fueron puestos a la transición temporizada (originalmente 10 minutos, se reduce a 5 segundos por simulación) el PLC vuelve a encender la máquina de pintar y apagar el compresor, como se observa en la figura 60.

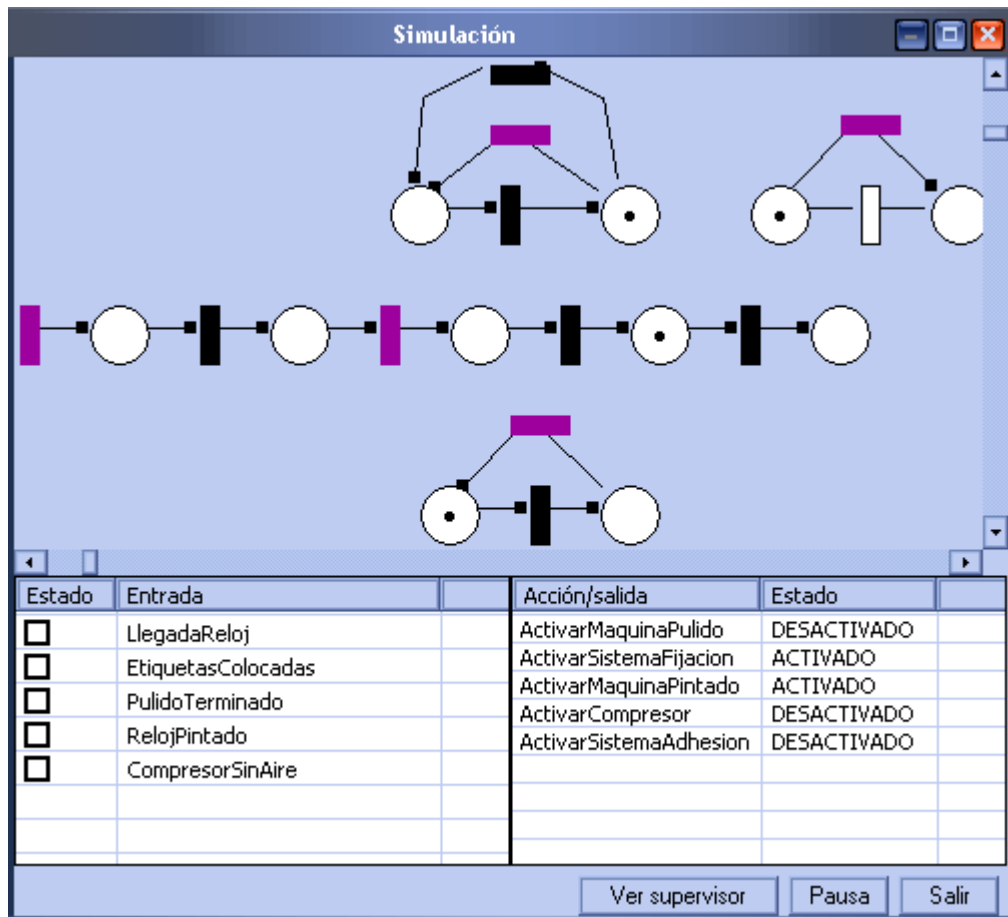


Figura 60. Terminación del colocado de etiquetas

Finalmente el PLC espera la señal de terminación de la operación de pintado, cuando esto ocurre, el PLC apaga todos los componentes a la espera de la llegada a de un nuevo reloj a la celda. Nótese, que no existe nada que evite que otro reloj llegase a la celda mientras el primero aún está siendo procesado, en este caso el PLC probablemente hubiese reaccionado de muy distintas maneras, estas son omitidas por extensión en este manual, se el lector podrá realizar más pruebas con la simulación para comprobar lo flexible que es el supervisor implementado mediante este método.

En la siguiente sección, se presentaran los comandos para generar ladder para un PLC Rockwell real. Este procedimiento se realiza mediante la generación de un fichero que contiene el ladder, y puede ser abierto en RSLogix 500 para descarga en el PLC.

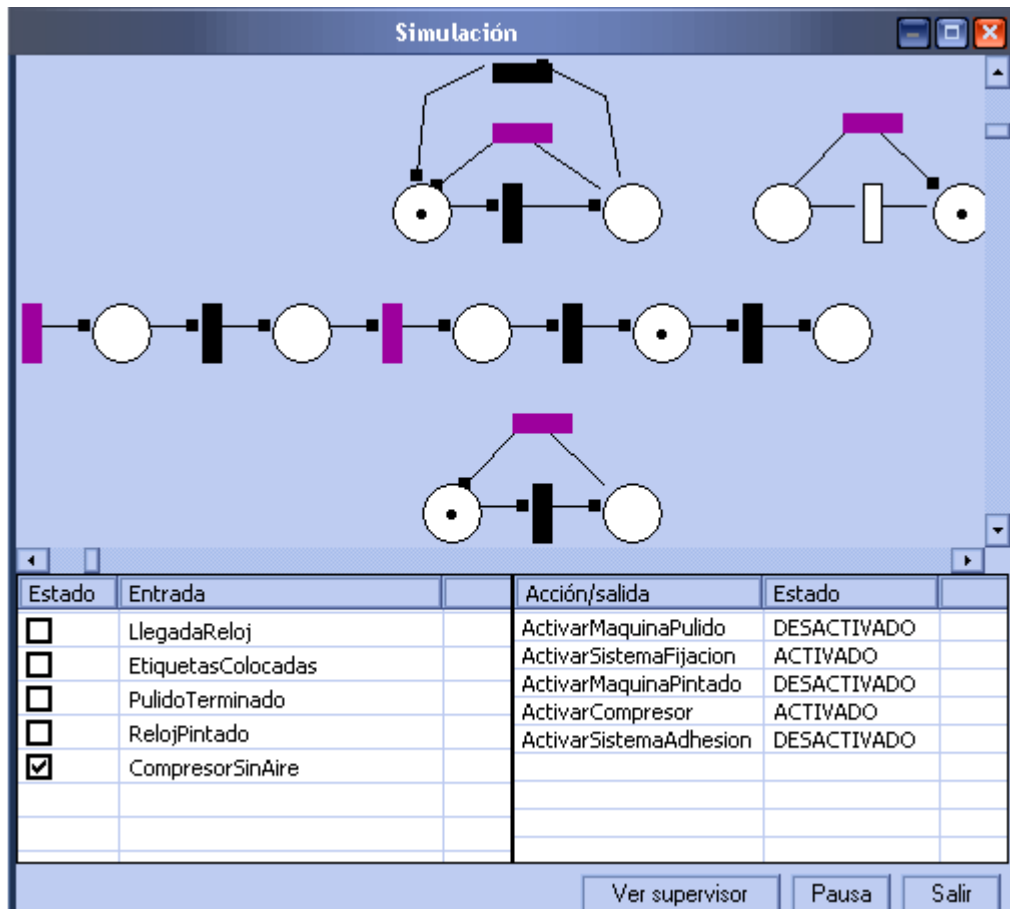


Figura 61. El compresor ha enviado la señal de falta de aire

#### 1.4.4.3 Generar código ladder para un PLC Rockwell SLC-500.

Para generar ladder para un PLC real, primero se debe cargar la especificación de este dispositivo a CRP. Esta especificación es un archivo *\*plc*, el cual contiene toda la información necesaria para generar código ladder para el controlador y puede ser cargado mediante el botón “Cargar PLC” de la interfaz de CRP.

Cargar PLC

Figura 62. Comando de carga del archivo PLC.

En este ejemplo, se generará ladder para el controlador “Micrologix 1100 Serie B”, del cual ya se dispone el archivo PLC correspondiente, este archivo fue creado usando “PLC Detector”, en la sección 1.5 se detalla cómo usar esta aplicación para crear dicho archivo.



Posteriormente, se procede a cargar este archivo mediante el botón “Cargar PLC” de la figura 62.

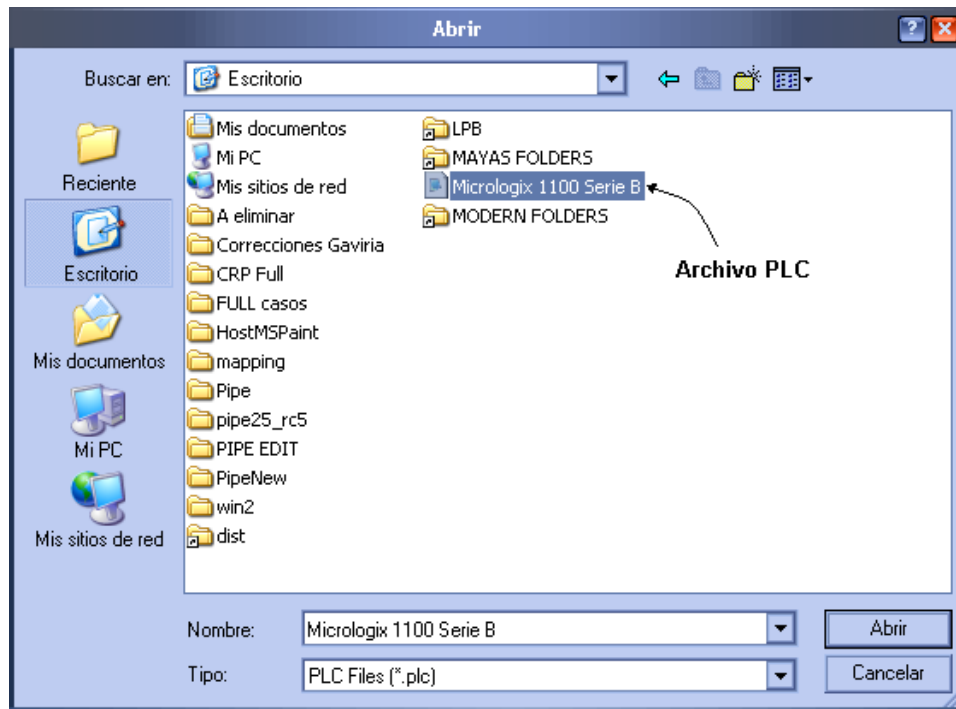


Figura 63. carga del archivo PLC.

Aparecerá un cuadro de diálogo en el cual se debe especificar la ubicación del archivo. Si esta operación tiene éxito, se mostrará una etiqueta que dice “PLC CARGADO” en la parte superior de la aplicación.



Figura 64. PLC cargado por CRP

Cuando esto aparece, ya es posible generar el código ladder para este dispositivo, solo falta asignar las entradas y salidas definidas en la red de petri por entradas y salidas reales del PLC, para esto, se hace doble clic sobre cada entrada o salida y en el cuadro de diálogo es posible elegir una. CRP remplazará todas las etiquetas por la entrada o salida real, en caso de no asignar una, el ladder se creará con el nombre editado en la red de petri, el usuario debe remplazar este valor en RSLogix500 después.

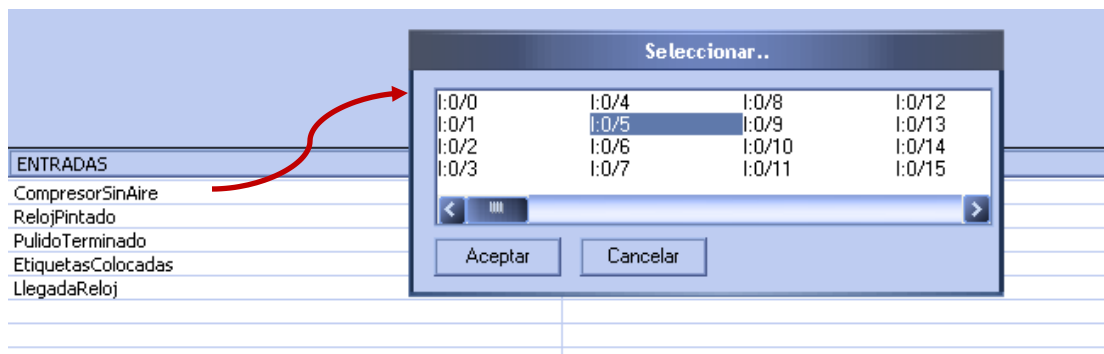


Figura 65. Elección de una entrada, un proceso similar se sigue con la elección de salidas.

ENTRADAS	VALOR
CompresorSinAire	I:0/5
RelojPintado	I:0/0
PulidoTerminado	I:0/14
EtiquetasColocadas	I:0/7
LlegadaReloj	I:0/2

ACCIONES/SALIDAS	VALOR
ActivarSistemaAdhesion	O:0/0
ActivarCompresor	O:0/1
ActivarMaquinaPintado	O:0/2
ActivarSistemaFijacion	O:0/3
ActivarMaquinaPulido	O:0/4

Figura 66. Todas las entradas y salidas.

Después de presionar el botón “Generar ladder”, se desplegará una ventana en la cual es posible especificar la ubicación donde el archivo SLC será creado. Este nuevo archivo contiene el ladder, y se puede abrir en RSLogix 500 listo para descargar al PLC, ya sea en un dispositivo real o en una simulación de rockwell.

### 1.5 Generar una especificación de un dispositivo (archivo PLC)

El archivo PLC sirve para indicar a CRP la configuración de un dispositivo, de esta manera es posible generar código ladder para este. Para generar una especificación en un archivo PLC se utiliza la herramienta “PLC Detector”, la cual se anexa con CRP.

“PLC Detector” tiene la siguiente interfaz de usuario.

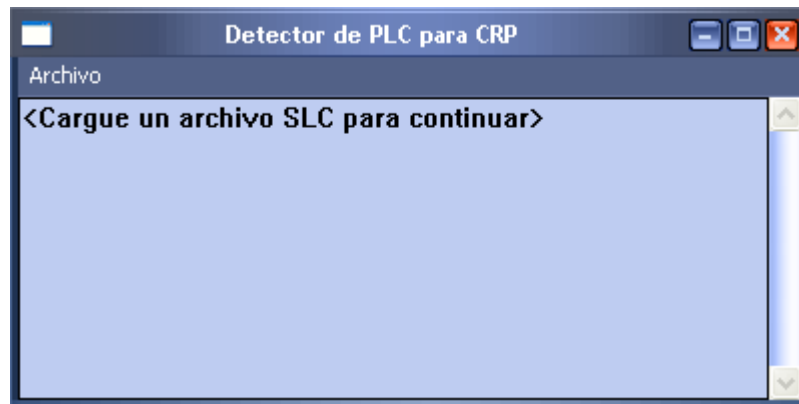


Figura 67. Interfaz de "PLC detector"

Junto con la configuración del dispositivo, el archivo PLC contiene información del número de entradas y salidas del dispositivo y los módulos que contiene. Este programa puede generar configuraciones para los siguientes dispositivos.

1747-L552 5/05 CPU - 32K mem. OS501  
 1747-L551 5/05 CPU - 16K mem. OS501  
 1747-L553 5/05 CPU - 64K mem. OS500  
 1747-L552 5/05 CPU - 32K mem. OS500  
 1747-L551 5/05 CPU - 16K mem. OS500  
 1747-L543C 5/04 CPU - 64K mem. OS401 Series C FRN 10 y posteriores  
 1747-L542C 5/04 CPU - 32K mem. OS401 Series C FRN 10 y posteriores  
 1747-L541C 5/04 CPU - 16K mem. OS401 Series C FRN 10 y posteriores  
 1747-L543C 5/04 CPU - 64K mem. OS401 Series C FRN 3-9  
 1747-L542C 5/04 CPU - 32K mem. OS401 Series C FRN 3-9  
 1747-L541C 5/04 CPU - 16K mem. OS401 Series C FRN 3-9  
 1747-L543 5/04 CPU - 64K mem. OS401  
 1747-L542B 5/04 CPU - 32K mem. OS401  
 1747-L541 5/04 CPU - 16K mem. OS401  
 1747-L542A 5/04 CPU - 24K mem. OS400  
 1747-L533E 5/03 CPU - 32K mem. OS302 Series C FRN 10 y posteriores  
 1747-L532E 5/03 CPU - 16K mem. OS302 Series C FRN 10 y posteriores  
 1747-L531E 5/03 CPU - 8K mem. OS302 Series C FRN 10 y posteriores  
 1747-L532E 5/03 CPU - 16K mem. OS302 Series C FRN 3-8  
 1747-L531E 5/03 CPU - 8K mem. OS302 Series C FRN 3-8  
 1747-L532C/D 5/03 CPU - 16K mem. OS302  
 1747-L531 5/03 CPU - 8K mem. OS302  
 1747-L532B 5/03 CPU - 16K mem. OS301  
 1747-L532 5/03 CPU - 16K mem. OS300  
 1747-L524 5/02 CPU - 4K mem.  
 1747-L514 5/01 CPU - 4K mem.  
 1747-L511 5/01 CPU - 1K mem.  
 Bul.1764 Micrologix 1500 LRP Series C  
 Bul.1764 Micrologix 1500 LRP Series B

Bul.1764	Micrologix 1500 LSP Series C
Bul.1764	Micrologix 1500 LSP Series B
Bul.1764	Micrologix 1500 LSP Series A
Bul.1766	Micrologix 1400 Series A
Bul.1762	Micrologix 1200 Series C
Bul.1762	Micrologix 1200 Series B
Bul.1762	Micrologix 1200 Series A
Bul.1763	Micrologix 1100 Series B
Bul.1763	Micrologix 1100 Series A
Bul.1761	Micrologix 1000 DH-485/HDslave
Bul.1761	Micrologix 1000
1747-L40A	24-115 VAC In. 16-RLY Out
1747-L40B	24-115 VAC In. 16-TRIAC Out
1747-L40C/F	24-DC SNK In. 16-RLY Out
1747-L40E	24-DC SNK In. 16-TRANS Out
1747-L40L	24-DC SRC In. 16-TRANS Out
1747-L40P	24-230 VAC In. 16-TRIAC Out
1747-L30A	18-115 VAC In. 12-RLY Out
1747-L30B	18-115 VAC In. 12-TRIAC Out
1747-L30C	18-DC SNK In. 12-RLY Out
1747-L30D	18-DC SNK In. 12-TRIAC Out
1747-L30L	18-DC SRC In. 12-TRANS SNK Out
1747-L30P	18-230 VAC In. 12-TRIAC Out
1747-L20A	12-115 VAC In. 8-RLY Out
1747-L20B	12-115 VAC In. 8-TRIAC Out
1747-L20C/F	12-DC SNK In. 8-RLY Out
1747-L20D	12-DC SNK In. 8-TRIAC Out
1747-L20E/G	12-DC SNK In. 8-TRANS SRC Out
1747-L20L/N	12-DC SRC In. 8-TRANS SNK Out
1747-L20P	12-240 VAC In. 8-TRIAC Out
1747-L20R	12-240 VAC In. 8-RLY Out

Para crear un archivo PLC para un dispositivo, se siguen los siguientes pasos.

1. Crear un nuevo proyecto en RsLogix500 con el dispositivo, y configurarlo usando las utilidades de ese software.

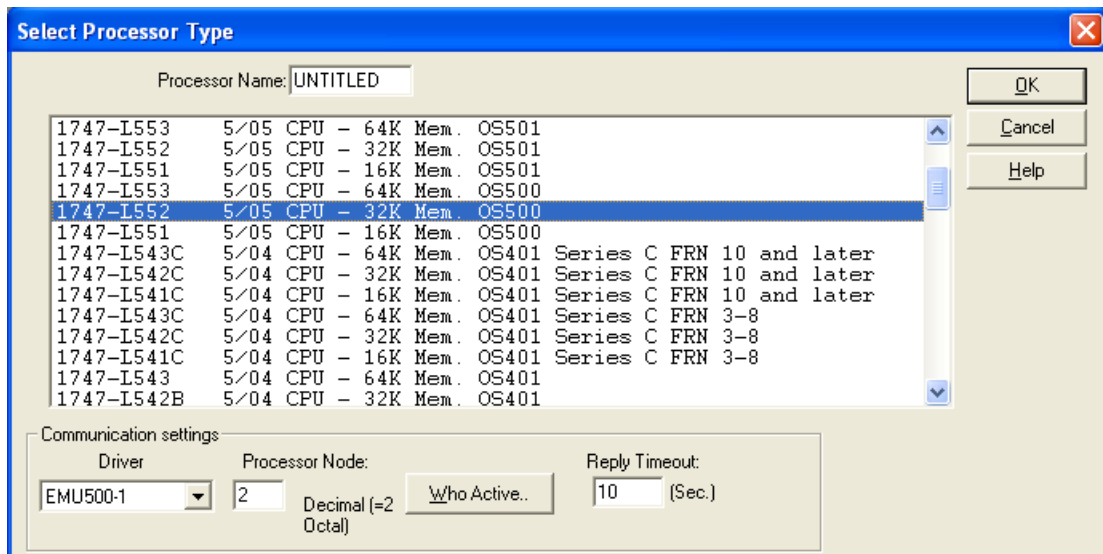


Figura 68. Creación de un nuevo proyecto en RsLogix500. Se utiliza un dispositivo 1747-L552

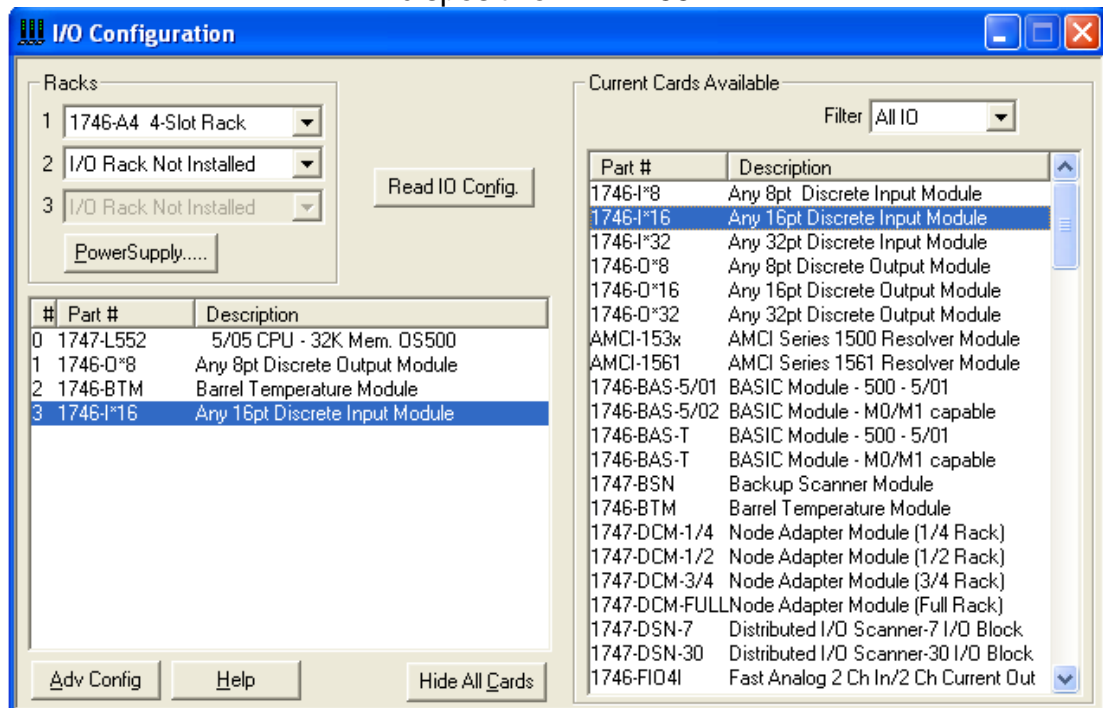


Figura 69. Configuración de módulos del dispositivo

En las dos figuras anteriores, se muestra la creación de un nuevo proyecto usando un dispositivo 1747-L522, como ejemplo. Este dispositivo contiene un Rack de cuatro slots, y contiene los siguientes módulos:

SLOT 1: Módulo de ocho salidas discretas.

SLOT 2: Módulo de temperatura de barril

SLOT 3: Módulo de 16 entradas discretas.

Finalmente, se debe crear al menos una línea de ladder en el proyecto, esta será usada como referencia por “PLC Detector”.

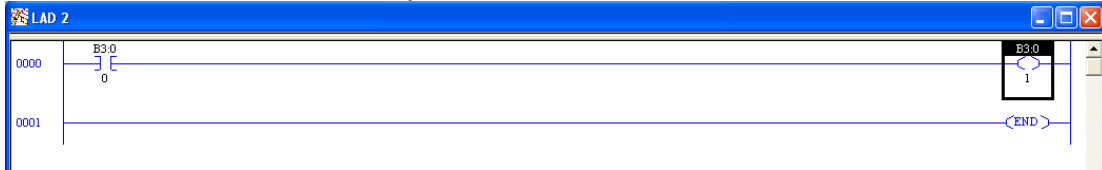


Figura 70. Creación de una línea de ladder de referencia.

Cuando lo anterior esta realizado, el proyecto se debe guardar en formato SLC, como se indica en la figura inferior. Este es el formato de exportar en RsLogix500, este pedirá la información a exportar.

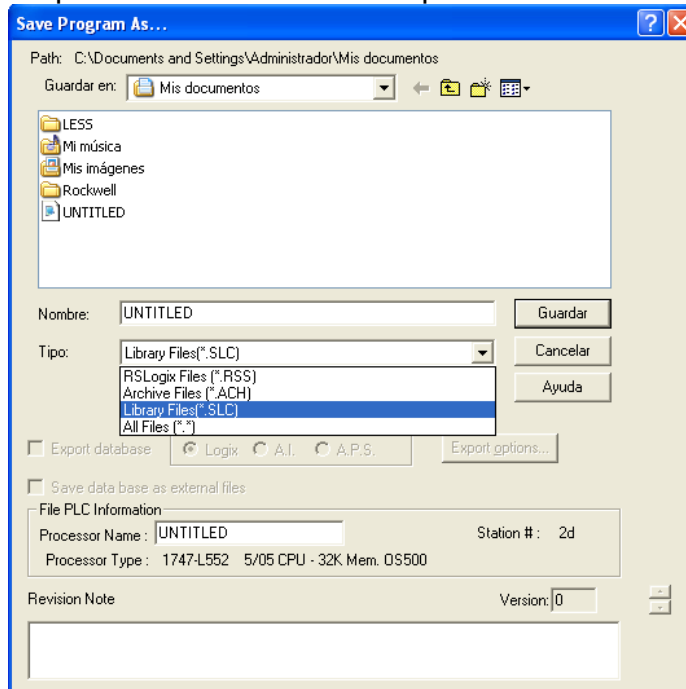


Figura 71. Guardar el proyecto en formato SLC.

La información a exportar se indica en el siguiente cuadro de diálogo de RsLogix500.

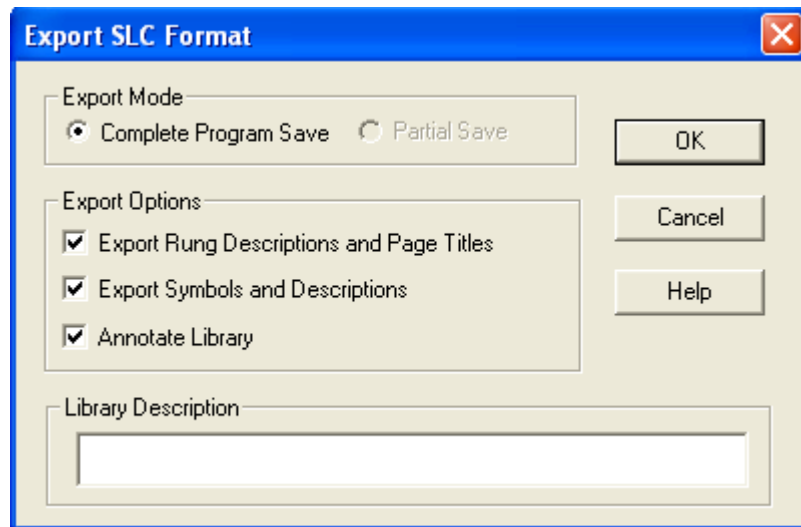


Figura 72. Información a exportar.

Se debe escoger el guardado completo del programa, y activar todas las opciones de exportación.

2. Una vez creado el archivo SLC, se debe abrir con "PLC Detector" y generar el archivo PLC.

Seleccionando la opción "Cargar" del menú "Archivo" de PLC detector, se desplegará un diálogo para buscar el archivo SLC proveniente de RsLogix500.



Figura 73. Opción cargar.

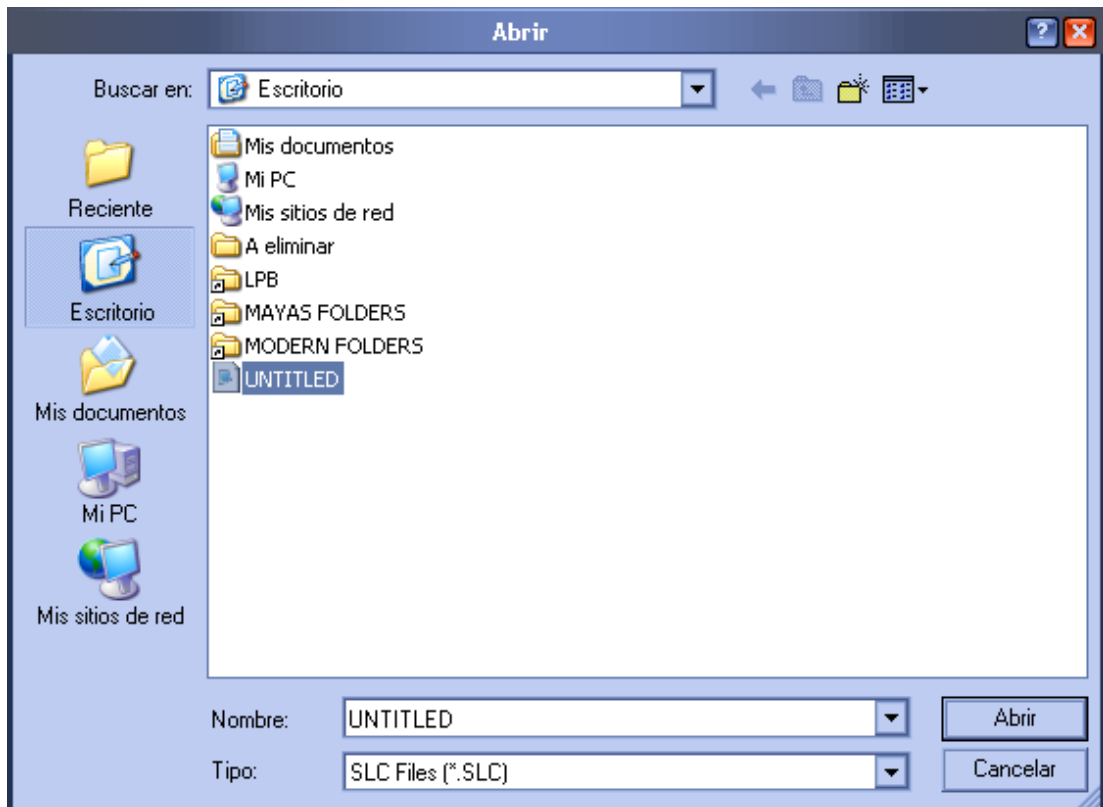


Figura 74. Cargar el archivo SLC a “PLC Detector”

Si “PLC Detector” puede determinar la configuración del PLC lo indicara en el cuadro de diálogo de la interfaz.

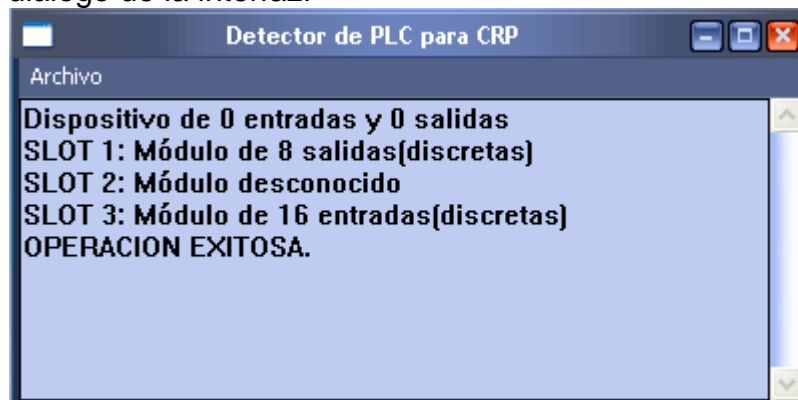


Figura 75. Detección exitosa del dispositivo.

Si la detección es exitosa, es posible crear el archivo PLC mediante la opción “Crear archivo PLC” del menú “Archivo”.

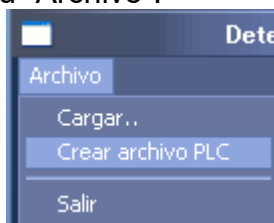




Figura 76. Opción crear el archivo PLC.

Se desplegará un cuadro de diálogo para establecer la ubicación donde se creara el archivo PLC.

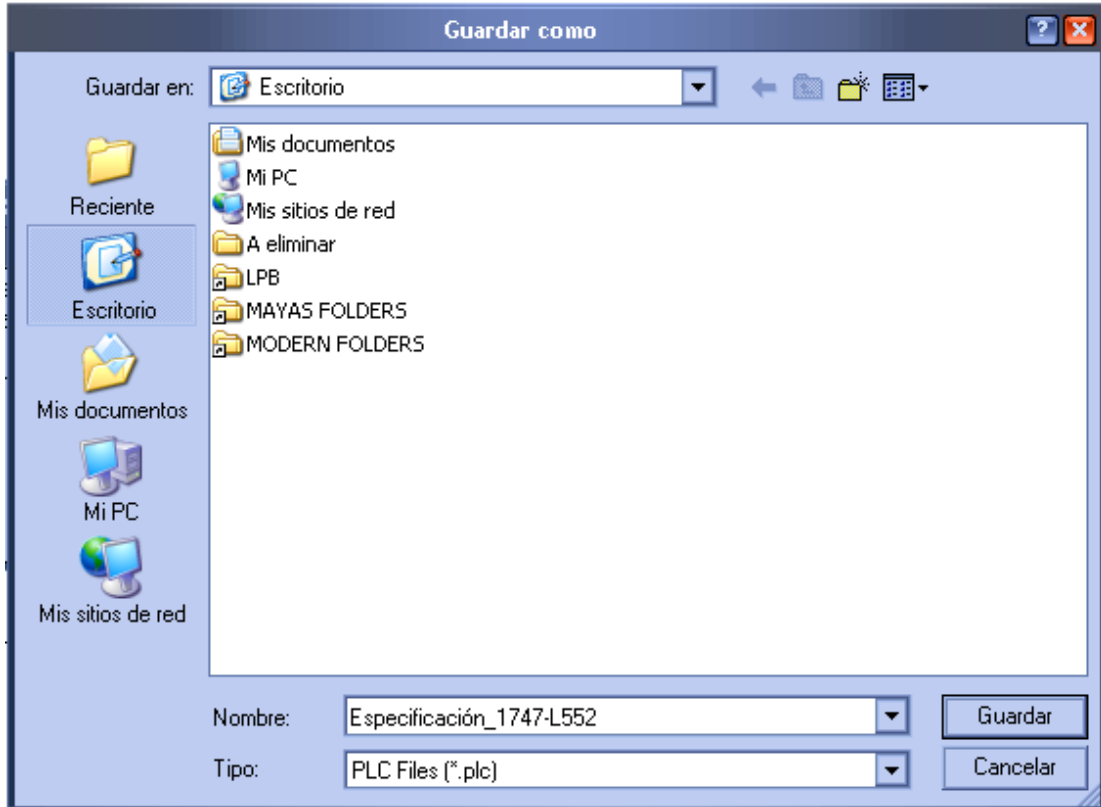
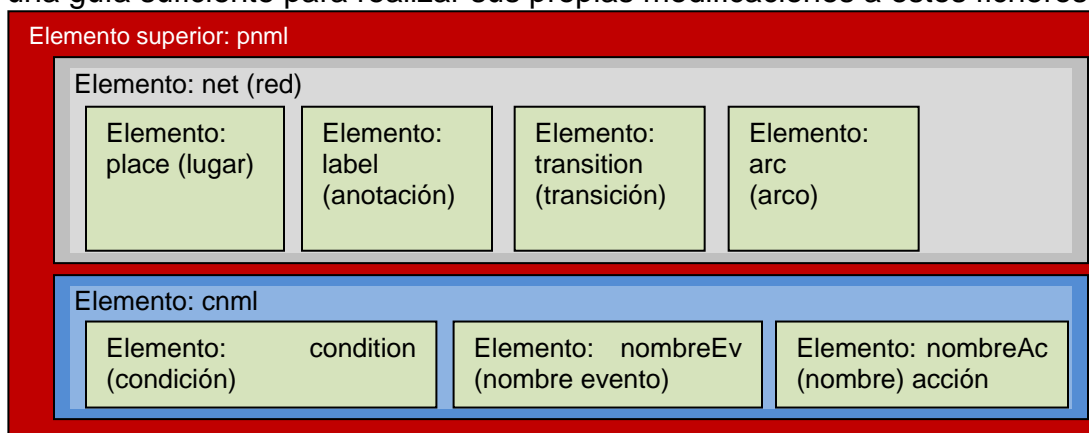


Figura 77. Guardar el archivo PLC.

Una vez creado, ya es posible utilizar el archivo PLC para generar ladder para este dispositivo utilizando CRP.

## 2 ANEXO B: DESCRIPCION DE LOS FICHEROS DE PIPE

Los archivos usados por la herramienta están escritos en lenguaje XML (Extensible Markup Language), no es necesario un conocimiento avanzado de este formato para manejarlos. En el siguiente esquema se denotan los elementos existentes en el fichero y su jerarquía. En este formato, todos los datos y valores se escriben como una cadena. En este anexo se explicarán los formatos de manera general, sirviendo de guía para lectores que desconozcan XML, para aquellos que manejen el lenguaje, este anexo será una guía suficiente para realizar sus propias modificaciones a estos ficheros.



PIPE se encarga de guardar el elemento net y todos sus elementos, el elemento cnml es adicionado por CRP para guarda las restricciones adicionadas por el usuario y las entradas o salidas asignadas a acciones y eventos.

La edición de estos elementos se declara con la palabra clave que aparece en el esquema, y termina con la misma palabra precedida por el carácter '/', en medio de estas dos instrucciones se declaran todos los atributos del elemento. Todas las etiquetas e instrucciones deben estar entre los caracteres '<' y '>'.

Al ser 'pnml' el elemento superior, entre las instrucciones de <pnml> y </pnml> se encuentra el contenido de todo el fichero.

```
<pnml>
...CONTENIDO DE TODO EL FICHERO
</pnml>
```

Después de la anterior definición, sigue la del elemento "net", para esta solo se definen dos atributos "id" y "type", el segundo siempre debe ser "P/T net" el primero puede ser cualquiera. Por ejemplo:

```

<pnml>
  <net id="Net-One" type="P/T net">
    ...CONTENIDO DE LA RED
  </net>
</pnml>

```

En medio de la declaración de “net”, están todas las declaraciones de lugares transiciones, arcos y anotaciones.

En primera instancia, se escriben las declaraciones de los lugares, cada lugar contiene los siguientes atributos.

- “id” , es la identificación única del lugar usada por PIPE
- “name” es le nombre del lugar
- “initialMarking” marca inicial del lugar.
- “graphics” el cual contiene a su vez el atributo “position”, la posición del lugar, este atributo contiene dos atributos a su vez “x” y “y”, que son las posiciones del lugar en el correspondiente eje.
- “accion” atributo que indica el nombre de la acción, si existe. Este atributo es opcional.

En caso de la versión modificada de PIPE, se ha hecho que “id” y “name” contengan siempre el mismo valor, esto se hizo para solucionar problemas cuando CRP y PIPE comparten datos, ya que dos lugares podrían tener el mismo nombre siempre y cuando el “id” sea diferente, esto genera confusiones ya que no es posible observar el “id” en el editor.

El atributo “id” es escribe con la instrucción de inicio del lugar, el atributo “position” se escribe en una sola instrucción en la declaración de “graphics” haciendo referencia a “x” y “y”, “name” se declara, y su valor se escribe entre dos clausulas “value”, de la misma manera se declara “initialMarking” y “accion”. Véase la siguiente ejemplo.

```

<place id="P0">
  <graphics>
    <position x="300.0" y="75.0" />
  </graphics>
  <name>
    <value>P0</value>
  </graphics />
</name>
<initialMarking>
  <value>0</value>
</graphics>
</initialMarking>
<accion>

```

```

    <value>V1On</value>
  </accion>
</place>

```

Las sentencias resaltadas en amarillo, son posibles valores de visualización para el nombre y el marcado inicial, sin embargo, PIPE 2.0 no genera ninguna opción alternativa de visualización, por lo que el atributo “graphics” de “name” siempre será una instrucción sobrante, así también el atributo “graphics” de “initialMarking”. Es necesaria la adición de estos elementos para que PIPE pueda leer correctamente el fichero. En verde han sido resaltados los valores de los atributos, seis atributos en total.

Las transiciones contienen de ocho a once atributos, estos son los siguientes

- “id” identificación única de la transición
- “graphics” que contiene “position” y este a “x” y “y”. La posición de la transición en el plano.
- “name” nombre de la transición
- “orientation” el grado de inclinación de la transición
- “rate” usado para la temporización de la transición
- “timed” indica si la transición es temporizada
- “uncontrol” indica si la transición no es controlable
- “unobser” indica si la transición no es observable
- “evento” indica el evento en esta transición. Opcional.
- “edged” indica si el evento se debe generar solo una vez con la activación de la señal. Solo se escribe si existe un evento para esta transición.
- “negate” indica si el evento se debe generar si la señal está apagada. Solo se escribe si existe un evento para esta transición.

Los atributos “id” y “graphics” se escribe de manera similar a la forma que fueron escritos para los lugares, todos los restantes se escriben colocando sus valores entre clausulas “value”. Véase el siguiente ejemplo.

```

<transition id="T0">
  <graphics>
    <position x="255.0" y="120.0" />
  </graphics>
  <name>
    <value>T0</value>
    <graphics />
  </name>
  <orientation>
    <value>0</value>
  </orientation>

```

```

    <rate>
    <value>1.0</value>
  </rate>
  <timed>
  <value>false</value>
</timed>
  <uncontrol>
  <value>false</value>
</uncontrol>
  <unobser>
  <value>false</value>
</unobser>
  <evento>
  <value>EventoUno</value>
</evento>
  <edged>
  <value>false</value>
</edged>
  <negate>
  <value>false</value>
</negate>
</transition>

```

La sentencia resaltada en amarillo es una sentencia sobrante. La parte resaltada en gris solo es escrita si la transición contiene un evento, en otro caso no se escribe.

Para los arcos se deben escribir los siguientes atributos.

- “id” identificación única del arco
- “source” id del elemento de salida
- “target” id del elemento de entrada
- “inscription” es el valor del peso del arco
- “arcpath” indica un punto en el camino del arco, se usa para graficar arcos quebrados. Este atributo contiene a su vez los atributos “id” el cual identifica el punto de quiebre dentro del arco, “x” y “y” que indican la posición del punto en el plano y “curvePoint” que es una bandera que indica si el arco se dibuja con una línea recta o una línea curva.

Para escribir los atributos anteriores, “id”, “source” y “target” se escriben en la declaración del arco, el atributo “inscription” se escribe entre clausulas “value” y el atributo “arcpath” se declara en una sentencia asignando el valor a todos los atributos. Véase el siguiente ejemplo.

```

<arc id="P0 to T0" source="P0" target="T0">
  <graphics />
  <inscription>
  <value>1</value>

```

```

<graphics />
</inscription>
<arcpath id="000" x="205" y="119" curvePoint="false" />
<arcpath id="001" x="260" y="131" curvePoint="false" />
</arc>

```

Para escribir cualquier valor en arcos, lugares y transiciones, PIPE hace uso de las plantillas encontradas en la carpeta *x/st* de su directorio. En caso de necesitar editar algún otro atributo para estos elementos, simplemente es preciso modificar estas plantillas. En el trabajo de esta tesis, se modificó la plantilla de las transiciones y lugares para editar los eventos, acciones y la controlabilidad y observabilidad, a continuación se muestran los cambios realizados.

La siguiente es la plantilla es usada por pipe para guardar los lugares, en rojo se resalta las adiciones realizadas. Todo esto se encuentra en el archivo "GeneratePNML.xls".

```

<xsl:template match="net/place">
  <xsl:element name="place">
    <xsl:call-template name="place-transition"/>
    <xsl:call-template name="initialMarking"/>
    <xsl:if test="(string-length(@Accion) > 0)">
      <xsl:call-template name="acciontmp"/>
    </xsl:if>
  </xsl:element>
</xsl:template>

```

En líneas generales, las instrucciones en rojo llaman a la plantilla "acciontmp" si el atributo "Accion" contiene algún valor, de ser así, la siguiente plantilla es llamada.

```

<xsl:template name="acciontmp">
  <xsl:element name = "accion">
    <xsl:element name = "value">
      <xsl:value-of select="@Accion"/>
    </xsl:element>
  </xsl:element>
</xsl:template>

```

Esta plantilla escribe el elemento acción y su valor en medio de clausulas "value", como se observa en el código del lugar en la parte superior.

Con respecto a las transiciones, esta es la plantilla con las modificaciones.

```

<xsl:template match="net/transition">
  <xsl:element name="transition">
    <xsl:call-template name="place-transition"/>
  </xsl:element>
</xsl:template>

```

```

        <xsl:call-template name="orientation"/>
        <xsl:call-template name="myrate"/>
        <xsl:call-template name="timed"/>
        <xsl:call-template name="control"/>
        <xsl:call-template name="observ"/>
        <xsl:if test="(string-length(@Evento) > 0)">
            <xsl:call-template name="putevento"/>
            <xsl:call-template name="putedged"/>
            <xsl:call-template name="putnegate"/>
        </xsl:if>
    </xsl:element>
</xsl:template>

```

Las plantillas “control” y “observ” son llamadas después de que se a escrito la información con respecto a la temporización. Si el atributo “Evento” contiene algún valor, se llama a las plantillas “putevento” “putedged” y “putnegate”. Todas estas plantillas son similares, declaran el elemento y escriben su valor en medio de clausulas “value”, el resultado se puede observar en el código escrito al principio del anexo. En seguida se muestran todas estas plantillas.

Plantilla “control”:

```

<xsl:template name="control">
    <xsl:element name = "uncontrol">
        <xsl:element name = "value">
            <xsl:value-of select="@nocontrol"/>
        </xsl:element>
    </xsl:element>
</xsl:template>

```

Plantilla “observ”:

```

<xsl:template name="observ">
    <xsl:element name = "unobser">
        <xsl:element name = "value">
            <xsl:value-of select="@noobser"/>
        </xsl:element>
    </xsl:element>
</xsl:template>

```

Plantilla “putedged”:

```

<xsl:template name="putedged">
    <xsl:element name = "edged">
        <xsl:element name = "value">
            <xsl:value-of select="@Edged"/>
        </xsl:element>
    </xsl:element>

```

```
</xsl:template>
```

Plantilla “putnegate”:

```
<xsl:template name="putnegate">
  <xsl:element name = "negate">
    <xsl:element name = "value">
      <xsl:value-of select="@Negate"/>
    </xsl:element>
  </xsl:element>
</xsl:template>
```

Plantilla “putevento”:

```
<xsl:template name="putevento">
  <xsl:element name = "evento">
    <xsl:element name = "value">
      <xsl:value-of select="@Evento"/>
    </xsl:element>
  </xsl:element>
</xsl:template>
```

En el archivo “GenerateObjectList.xls” de la carpeta *x/st* de PIPE, se encuentran las plantillas usadas para la lectura de los ficheros. En estas plantillas solo será necesario adicionar unas cuantas líneas de código para leer los atributos.

Plantilla de lectura para transiciones:

```
<xsl:template match="transition">
  <xsl:element name="transition">
    <xsl:attribute name="rate"><xsl:value-of
select="rate/value"/></xsl:attribute>
    <xsl:attribute name="timed"><xsl:value-of
select="timed/value"/></xsl:attribute>
    <xsl:attribute name="angle"><xsl:value-of
select="orientation/value"/></xsl:attribute>
    <xsl:attribute name="nocontrol"><xsl:value-of
select="uncontrol/value"/></xsl:attribute>
    <xsl:attribute name="noobser"><xsl:value-of
select="unobser/value"/></xsl:attribute>
    <xsl:attribute name="Evento"><xsl:value-of
select="evento/value"/></xsl:attribute>
    <xsl:attribute name="Edged"><xsl:value-of
select="edged/value"/></xsl:attribute>
    <xsl:attribute name="Negate"><xsl:value-of
select="negate/value"/></xsl:attribute>
```



```

        <xsl:call-template name="place-transition"/>
    </xsl:element>
</xsl:template>

```

Plantilla para la lectura de lugares:

```

<xsl:template match="place">
    <xsl:element name="place">
        <xsl:call-template name="place-transition"/>
        <xsl:attribute name="initialMarking">
            <xsl:value-of select="initialMarking/value"/>
        </xsl:attribute>
        <xsl:attribute name="markingOffsetX">
            <xsl:value-of
select="initialMarking/graphics/offset/@x"/>
        </xsl:attribute>
        <xsl:attribute name="markingOffsetY">
            <xsl:value-of
select="initialMarking/graphics/offset/@x"/>
        </xsl:attribute>
        <xsl:attribute name="Accion"><xsl:value-of
select="accion/value"/></xsl:attribute>
    </xsl:element>
</xsl:template>

```

Con las anteriores plantillas PIPE puede escribir y leer archivos de redes de petri con los nuevos atributos que se adicionaron. PIPE utiliza las clases:

```

org.w3c.dom.Attr;
org.w3c.dom.DOMException;
org.w3c.dom.Document;
org.w3c.dom.Element;
org.w3c.dom.Node;
org.w3c.dom.NodeList;
org.xml.sax.SAXException.

```

Para la creación de todos los elementos y sus atributos que llevará el archivo XML, estas clases generan los elementos que van a llevar el fichero. La descripción de estas clases se encuentra en la documentación oficial suministrada por Oracle, por tanto no se extenderá en este anexo.

### 3 ANEXO C. METODOLOGÍAS PARA HALLAR SIFONES MÍNIMOS Y TRAMPAS MÍNIMAS EN UNA PN

Varios algoritmos para calcular sifones y trampas han sido propuestos en la literatura. En los trabajos de Ezpeleta, presenta una aproximación algebraica basada en la transformación de la matriz de incidencia de la PN, el mismo es aplicado en la disertación presentada por Moody, dicho método se explica rápidamente abajo. Una definición muy completa es encontrada en el trabajo “*Incidence matrix based methods for computing repetitive vectors and siphons of petri nets*”, aunque en este trabajo los autores plantean un método ligeramente diferente al implementado en esta tesis.

A continuación se da una definición del método.

**Definición 1.1** Sea  $x$  un vector de números enteros correspondiente a los lugares de una PN. El **soporte** de  $x$ , denotado como  $\|x\|$ , se refiere al conjunto de lugares tales  $x_i \neq 0$ . Los sifones y trampas son conjuntos de lugares y no vectores, la definición de soporte es usada para ir entre estas dos representaciones [13].

**Proposición 1.1 Caracterización necesaria y suficiente de sifones.** Sea una PN con matriz de incidencia  $D$ , un vector  $x$ ,  $x \geq 0$  tal que:

$$x^T D' \leq 0$$

Donde  $D'$  es la transformación de la matriz de incidencia  $D$ , entonces  $\|x\|$  es un sifón en la PN con matriz de incidencia  $D$ .

La matriz  $D'$ , es la transformación de la matriz original  $D$ ,  $D'$  puede ser calculada de la siguiente manera:

$$D'^+ = D^+ \\ \forall t, \forall p \in \bullet t, D'^-(p, t) = k D^-(p, t)$$

Donde:

$$k = \sum_{p' \in \bullet t} D^+(p', t) \\ D' = D'^+ - D'^-$$

Para encontrar trampas, se aplica el método con la inversión de todos los arcos de la red, es decir, encontrar los sifones de la red de Petri con matriz de incidencia  $-D$ .

Otro enfoque trabajado por distintos investigadores], es el presentado en el trabajo “*Detections of Deadlocks and Traps in Petri Nets by Means of Thelen’s Prime Implicant Method*” este se basa en la representación de la PN en una ecuación lógica cuyas raíces conforman un sífon en la misma. Este enfoque se explica a continuación.

**Definición 1.2 Literal lógico** es una variable la cual puede tomar dos valores, cero o uno, el primer valor se entiende como *falso* y el segundo *verdadero*, respectivamente.

**Definición 1.3 Expresión lógica o booleana** es una función  $f(k_0, k_1 \dots k_n) \rightarrow 1,0$  donde  $k_0, k_1 \dots k_n$  son literales lógicos, toda la expresión puede tomar el valor de cero o uno.

**Definición 1.4 Un implicante**  $k$  de una expresión lógica  $F$  es una conjunción de literales (lógicos) tales que  $k \rightarrow F$ .

**Definición 1.5 Un implicante**  $k$  de una expresión lógica  $F$  es **primario**, si al remover cualquier literal de  $k$ , este deja de ser un implicante de  $F$ .

Para que un conjunto  $S$  sea un sífon debe cumplir  $\forall t \in T: t_i \bullet \cap S \neq \emptyset \Rightarrow \bullet t_i \cap S \neq \emptyset$ , de esta manera es posible representar la red en una expresión lógica [16], sea  $X = \{x_1, x_2 \dots x_m\}$  un conjunto de variables booleanas donde  $m = |P|$  y  $x_i = 1$  si y solo si  $p_i$  pertenece a un conjunto de lugares, se denota como  $X_i, X_i^*$ . Los subconjuntos en  $X$  correspondientes a  $\bullet t_i$  y  $t_i \bullet$  respectivamente, y a  $\vee [X_i]$  y  $\vee [X_i^*]$  las disyunciones de las variables correspondientes a los dos conjuntos. Entonces la condición para la transición  $t_i$  puede ser representada por la siguiente expresión [21]:

$$\vee [X_i^*] \rightarrow \vee [X_i] = 1$$

Utilizando la fórmula para todas las transiciones de la red, los sífonos de la PN son determinados por las raíces de la ecuación booleana [21] y [22]:

$$\wedge_{i=1}^n (\vee [X_i^*] \rightarrow \vee [X_i]) = 1$$

Donde  $n$  es el número de transiciones de la PN.

Hallar las raíces de una ecuación como la anterior es una tarea que puede requerir recursos computacionales considerables, y la cantidad de tiempo tiende a crecer de manera exponencial con respecto a los componentes de la red de Petri. En el trabajo “*Heuristics for Thelen’s Prime Implicant Method*” se propone un método para el cálculo de ecuaciones lógicas, el cual permite la

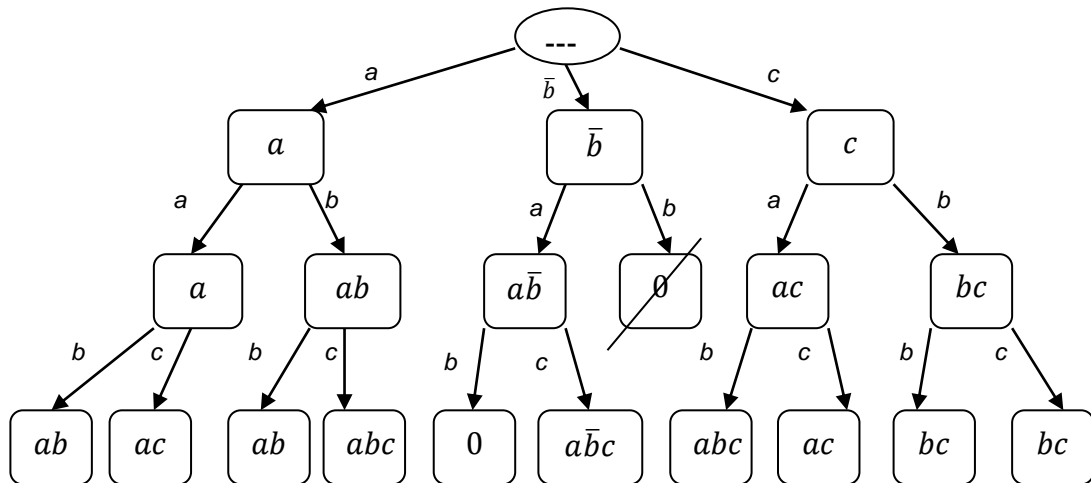
optimización del cálculo mediante reglas que disminuyen la extensión del problema., este algoritmo se aplica a expresiones en la forma normal conjuntiva de manera que es preciso la transformación de la ecuación a esta estructura [21]:

$$\begin{aligned} \vee [X_i^*] \rightarrow \vee [X_i] &= \overline{\vee [X_i^*]} \vee (\vee [X_i]) \\ &= \wedge_{i=1}^m \overline{x_i^{j^*}} \vee (\vee [X_i]) \\ &= \wedge_{i=1}^m (x_i^{j^*} \vee (\vee [X_i])) \end{aligned}$$

Al realizar la transformación anterior, resulta en una inversión el significado de la pertenencia.

El método propuesto en el anterior trabajo consiste en la construcción de un árbol el cual genera una rama por cada literal de cada disyunción de la expresión, teniendo en cuenta que esta se encuentra en forma normal conjuntiva. A continuación se muestra un ejemplo, donde se soluciona la siguiente expresión:

$$f = (a \vee \bar{b} \vee c)(a \vee b)(b \vee c)$$



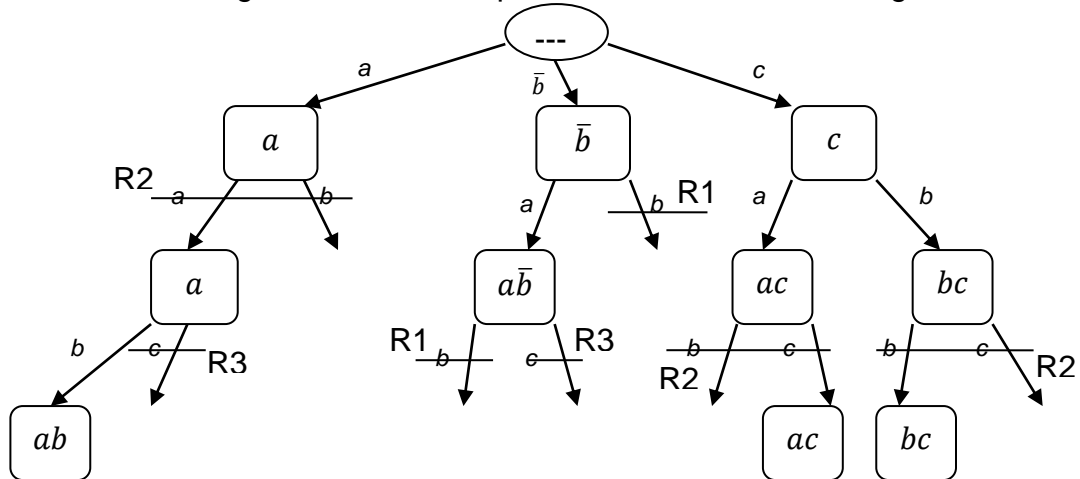
En el último nivel del árbol anterior se encuentran todas las soluciones de la expresión  $f$ , se observa como muchas de ellas se repiten y solo algunas son mínimas, realizando una revisión se puede determinar que solo existen tres soluciones básicas:  $ab, ac$  y  $bc$ .

Las tres reglas para eliminar ramas innecesarias en el proceso de solución son:

- R1: Eliminar la rama si en el nodo predecesor se encuentra el complemento del literal.

- R2: se descarta una disyunción si contiene un literal que ya existe en el nodo predecesor.
- R3: se descarta una rama si en un nivel más alto existe un nodo aun no expandido con el mismo literal.

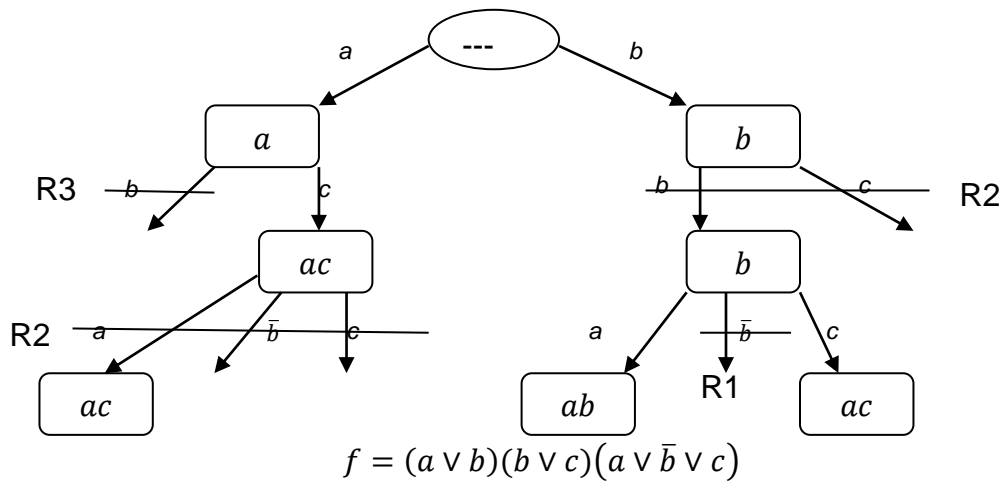
Utilizando las reglas anteriores, el problema se reduce en el siguiente árbol:



Una mayor reducción es posible aplicando las siguientes reglas heurísticas:

- H1: Ordenar las disyunciones desde la cual contenga menor número de literales a la que más cantidad de literales posea.
- H2: Escoger la disyunción con la menor cantidad de literales ya existentes en las disyunciones escogidas anteriormente.
- H3: Ordenar los literales de cada disyunción en dos grupos, el primer grupo compuesto de literales que no aparecen en las disyunciones siguientes, y el segundo ordenado según la frecuencia de aparición en las disyunciones siguientes.

Utilizando las anteriores reglas, la expresión  $f$  se reescribe:



Otro enfoque, consiste en el cálculo de únicamente los sifones mínimos, como se realiza en el artículo "*Some results on the computation of minimal siphons in petri nets*", en este trabajo, se realiza una reducción recursiva de la red hasta encontrar todos los sifones de la red. En términos generales, el algoritmo propuesto en este trabajo es:

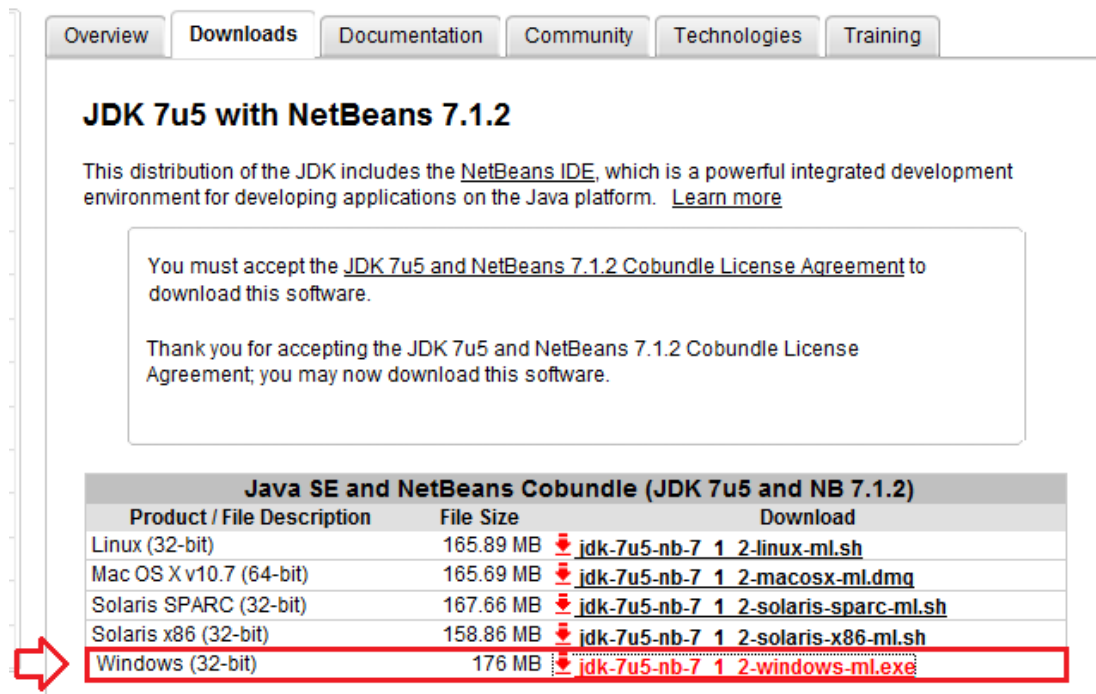
1. Encontrar un sifón genérico en la red.
2. Encontrar un sifón mínimo en el sifón genérico.
3. Descomponer la red para cada lugar del sifón mínimo y para cada problema volver al paso uno.

Este enfoque es bastante efectivo en comparación con los anteriores, tanto en calidad de los sifones encontrados como tiempo de cálculo. No se realiza una descripción detallada de este método ya que es posible encontrar y descargar el trabajo original en internet sin costo.

## 4 ANEXO D. DESCARGA E INSTALACIÓN DE NETBEANS Y DEV C

### 4.1 Descarga e Instalación de NetBeans

1. Ingrese al link:  
<http://www.oracle.com/technetwork/java/javase/downloads/jdk-7-netbeans-download-432126.html>, y descargue los paquetes NetBeans y JDK compatibles con su ordenador. En este manual, se descargará una versión para windows XP (sistema operativo de 32 bits).



Overview Downloads Documentation Community Technologies Training

### JDK 7u5 with NetBeans 7.1.2

This distribution of the JDK includes the [NetBeans IDE](#), which is a powerful integrated development environment for developing applications on the Java platform. [Learn more](#)

You must accept the [JDK 7u5 and NetBeans 7.1.2 Cobundle License Agreement](#) to download this software.

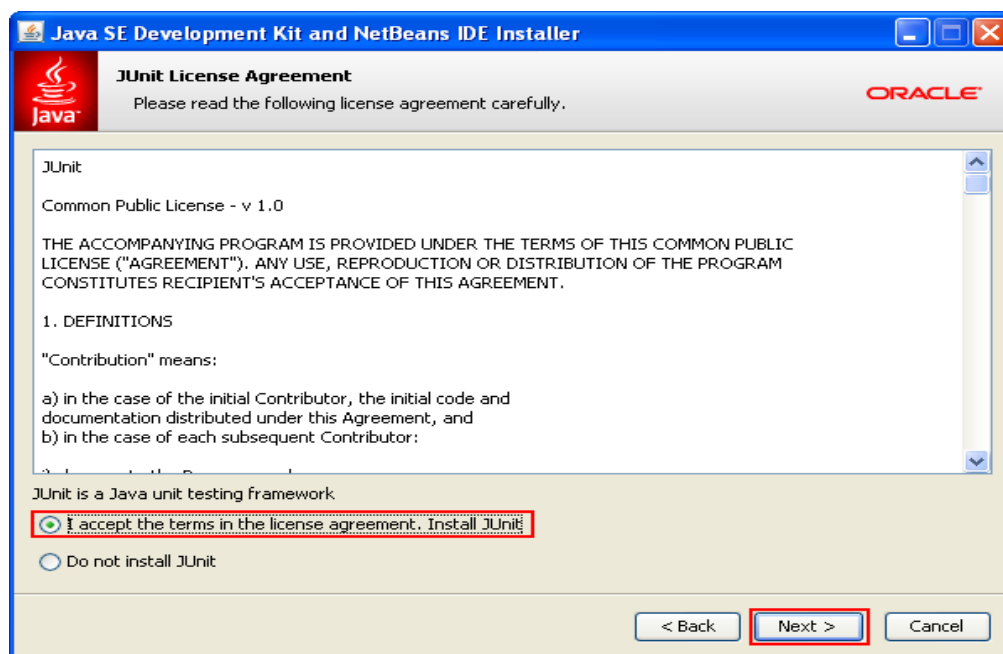
Thank you for accepting the JDK 7u5 and NetBeans 7.1.2 Cobundle License Agreement; you may now download this software.

Java SE and NetBeans Cobundle (JDK 7u5 and NB 7.1.2)		
Product / File Description	File Size	Download
Linux (32-bit)	165.89 MB	<a href="#">jdk-7u5-nb-7_1_2-linux-ml.sh</a>
Mac OS X v10.7 (64-bit)	165.69 MB	<a href="#">jdk-7u5-nb-7_1_2-macosx-ml.dmg</a>
Solaris SPARC (32-bit)	167.66 MB	<a href="#">jdk-7u5-nb-7_1_2-solaris-sparc-ml.sh</a>
Solaris x86 (32-bit)	158.86 MB	<a href="#">jdk-7u5-nb-7_1_2-solaris-x86-ml.sh</a>
Windows (32-bit)	176 MB	<a href="#">jdk-7u5-nb-7_1_2-windows-ml.exe</a>

2. Cuando haya descargado el archivo ejecútelo dando doble clic sobre él. De clic sobre el botón Next de la ventana de instalación del programa como se indica a continuación:

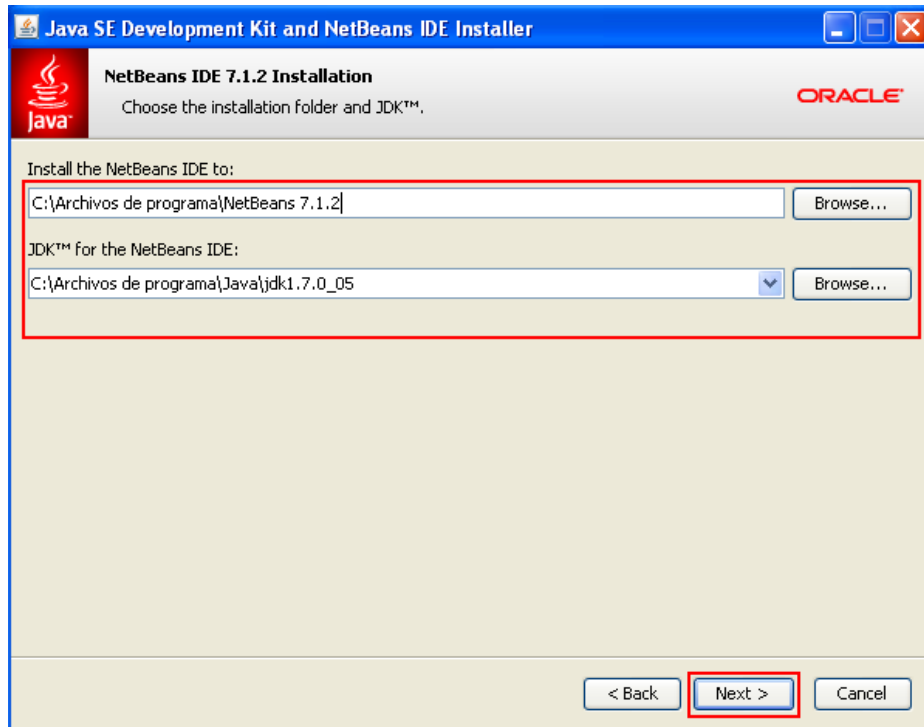


3. En la siguiente pantalla de clic en aceptar los términos de licencia de y luego en Next.

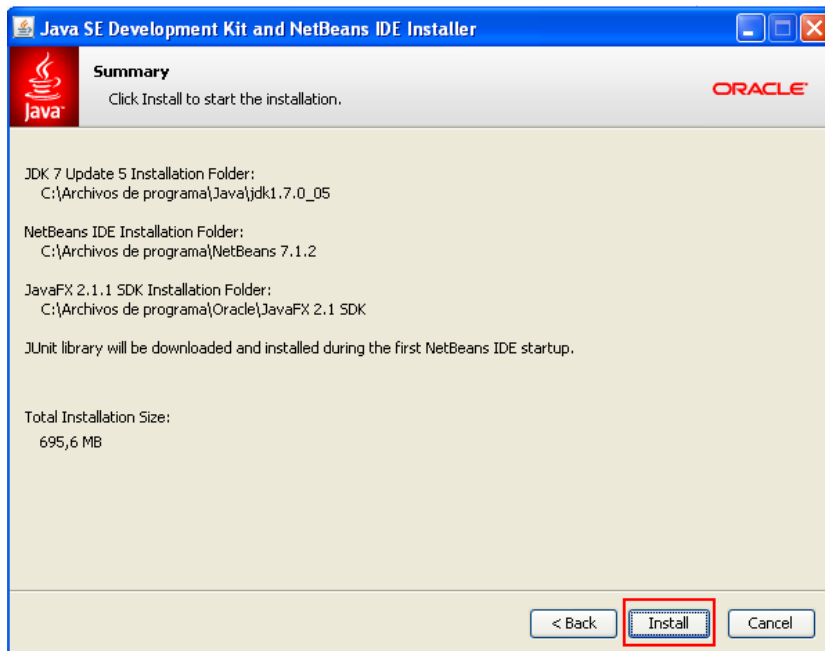


4. En la siguiente pantalla elija los directorios de instalación para NetBeans y JDK.





5. La pantalla siguiente muestra un resumen de las aplicaciones que se van a instalar, de clic en Install.



6. Al terminar la instalación de clic en Finish y listo.

## 4.2 Descarga e instalación de DEV C

1. Ingrese a la siguiente dirección: <http://www.bloodshed.net/dev/devcpp.html>, cuando haya ingresado a la página diríjase a la parte inferior de la página hasta encontrar las descargas disponibles, elija la descarga que incluye el compilador Mingw dando clic en SourceForge como se indica en la figura.



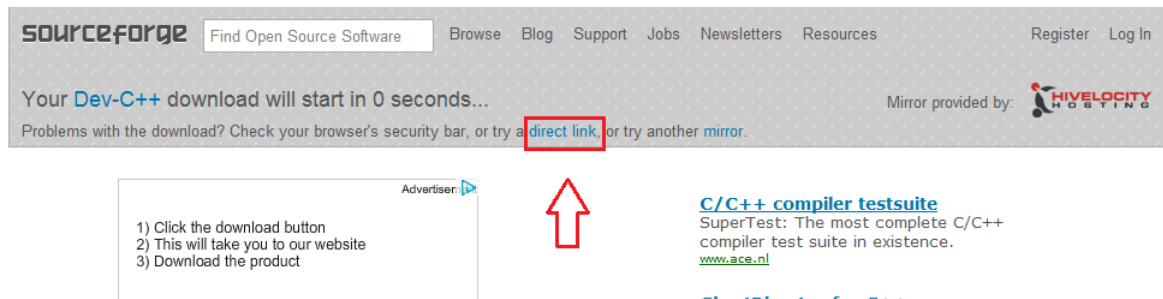
**Downloads**

 **Dev-C++ 5.0 beta 9.2 (4.9.9.2) (9.0 MB) with Mingw/GCC 3.4.2**  
Dev-C++ version 4.9.9.2, includes full Mingw compiler system with GCC 3.4.2 and GDB 5.2.1 See [NEWS.txt](#) for changes in this release.


Download from:

- [SourceForge](#)


2. La descarga debe iniciarse automáticamente, si no, dar clic en direct link, donde se indica por medio del rectángulo y la flecha:



**SOURCEFORGE** Find Open Source Software Browse Blog Support Jobs Newsletters Resources Register Log In

Your Dev-C++ download will start in 0 seconds... Mirror provided by: 

Problems with the download? Check your browser's security bar, or try a [direct link](#), or try another mirror.

Advertiser: 

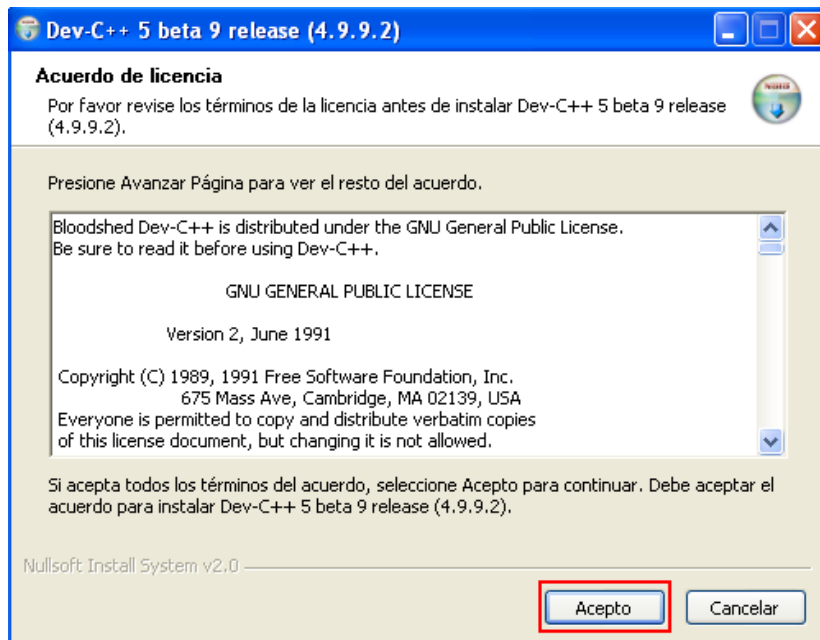
- 1) Click the download button
- 2) This will take you to our website
- 3) Download the product

[C/C++ compiler testsuite](#)  
SuperTest: The most complete C/C++ compiler test suite in existence.  
[www.ace.nl](http://www.ace.nl)

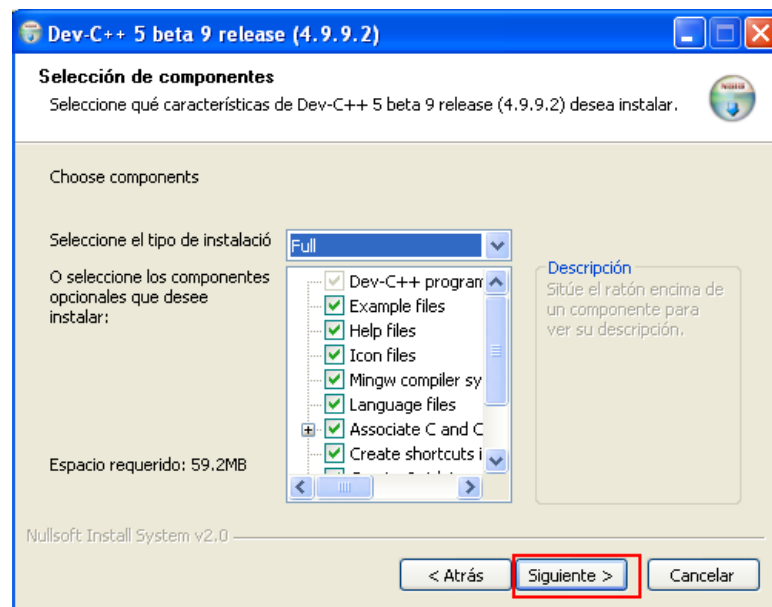
3. Cuando haya finalizado la descarga, ejecute el archivo dando doble clic sobre este.
4. Elija el idioma de instalación y clic en OK



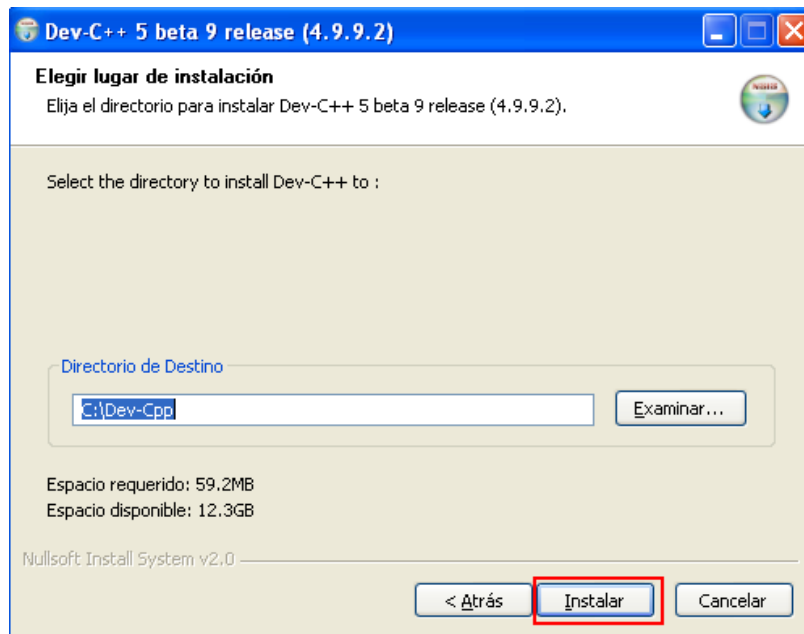
5. En la siguiente ventana de clic en acepto, para aceptar el acuerdo de licencia.



6. En la siguiente ventana clic en siguiente.



7. En la siguiente ventana clic en Instalar.



8. Clic en terminar y listo.

## 5 ANEXO E: DESCRIPCIÓN EN UML DE LA HERRAMIENTA CRP – PIPE.

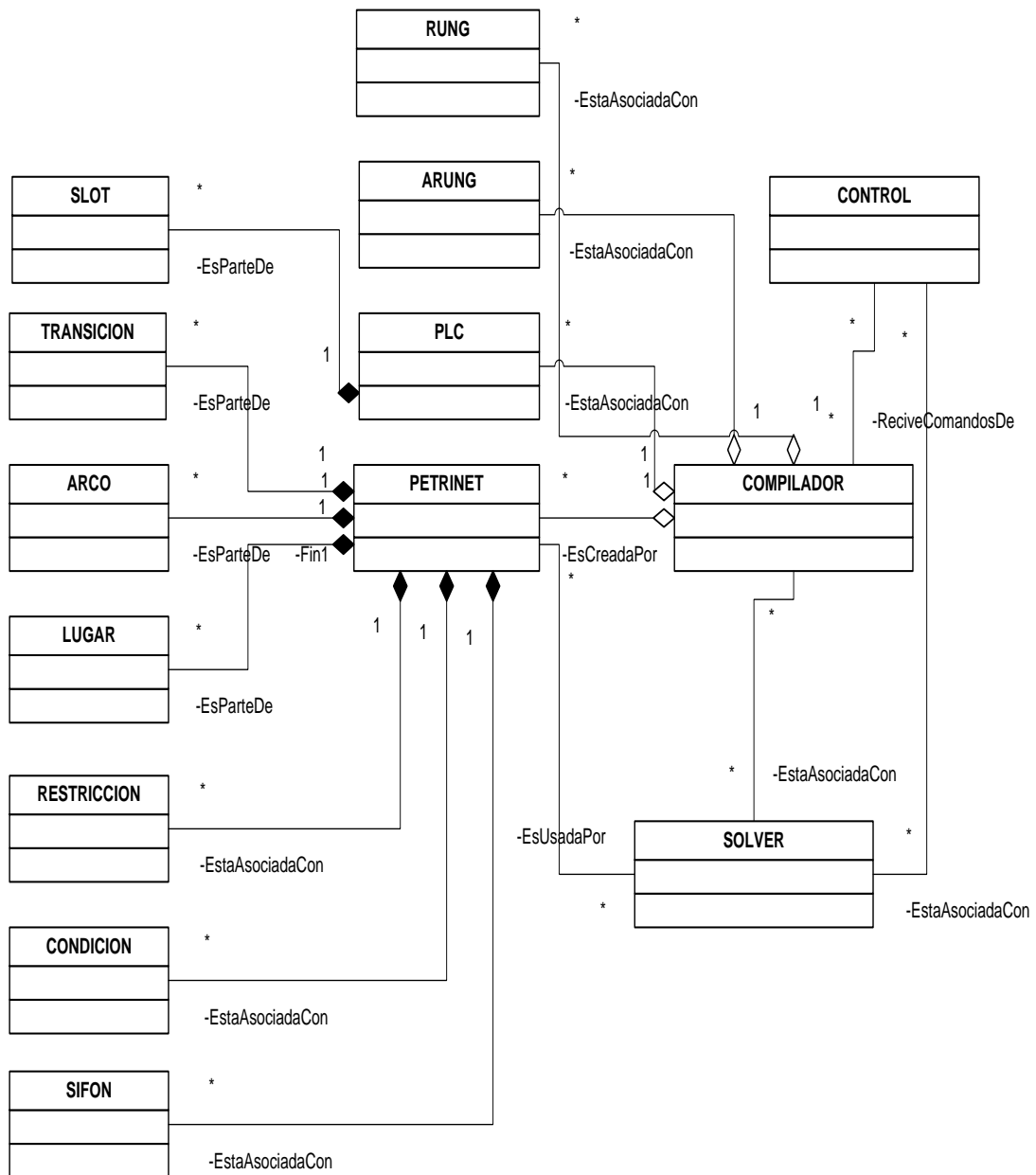
La herramienta CRP fue desarrollada usando un enfoque orientado a objetos en lenguaje C++, PIPE ha sido implementado en esta metodología. CRP implementa las siguientes clases.

- ARCO
- SIFON (BUCLE)
- COMPILADOR
- CONDICION
- RESTRICCION (GNERESTRICT)
- LUGAR
- TRANSICION
- PETRINET
- PLC
- SOLVER
- RUNG
- ARUNG

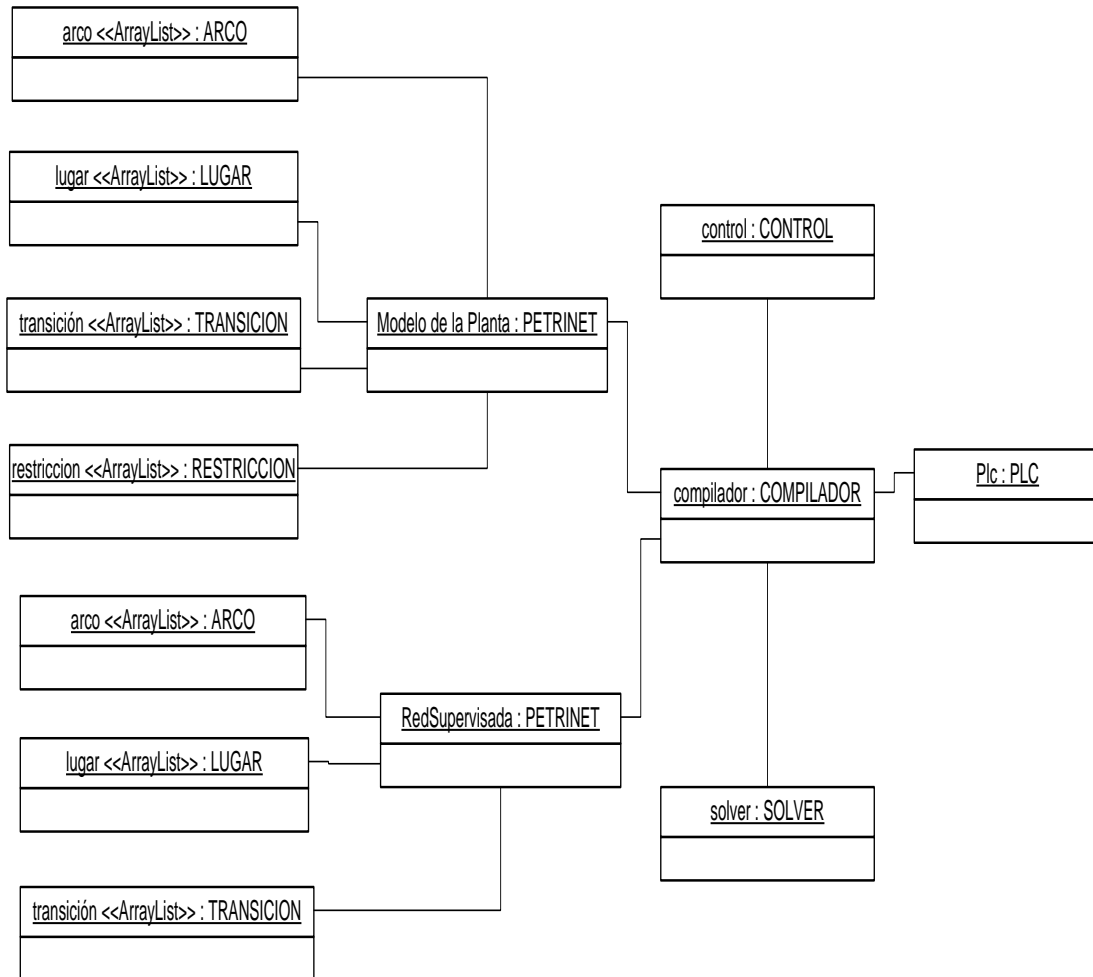
La descripción de estas clases con todos sus atributos y funciones se encuentra en el manual de funciones, el cual se encuentra en el CD de esta tesis.

A continuación se hace una descripción de las funcionalidades de la herramienta en diagramas de UML, esta se realiza para los pertenecientes a CRP, la implementación en clases del software PIPE puede encontrarse en: <http://pipe2.sourceforge.net/docs.html>.

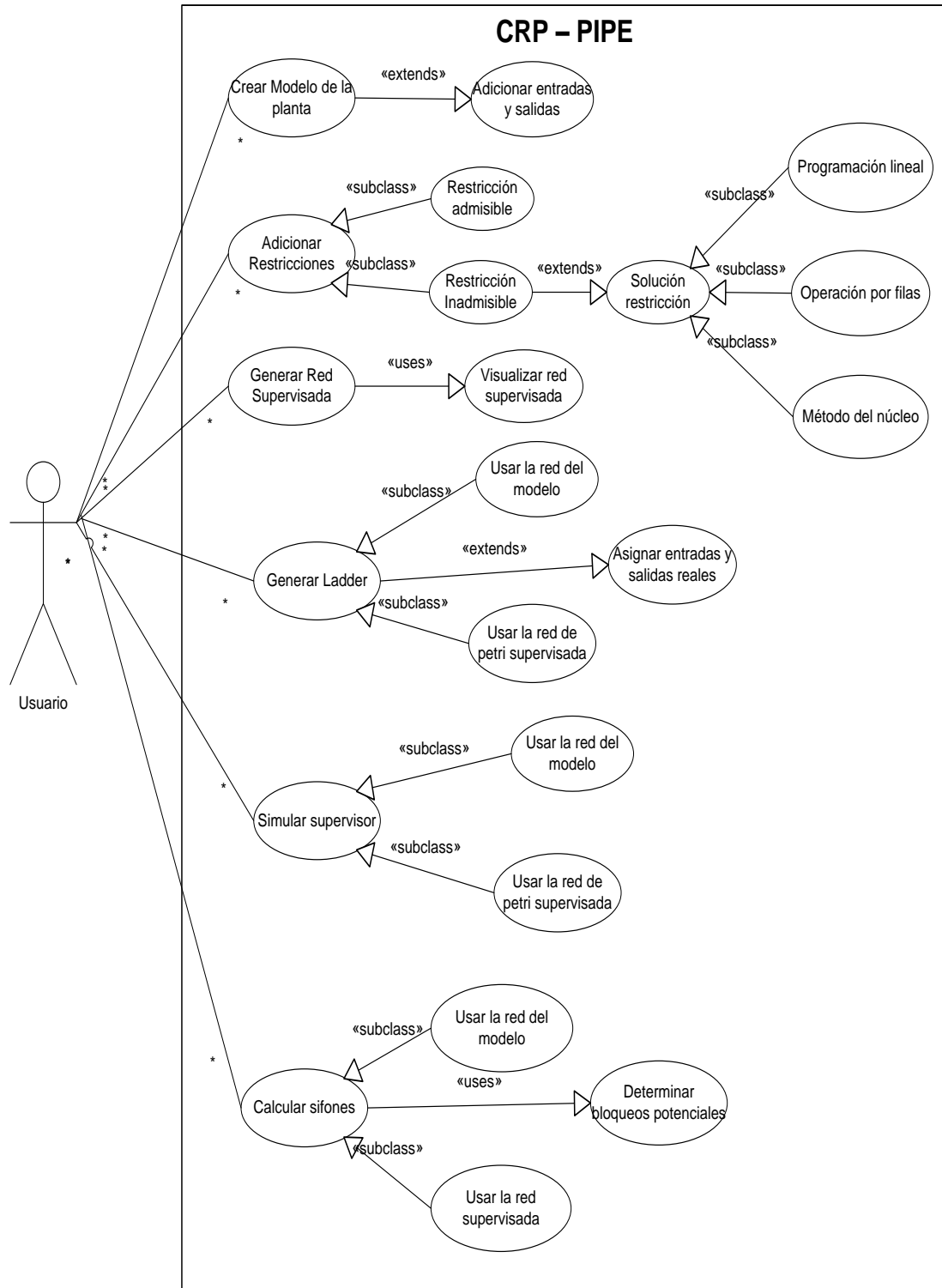
## Diagrama de Clases.



## Diagrama de Objetos.



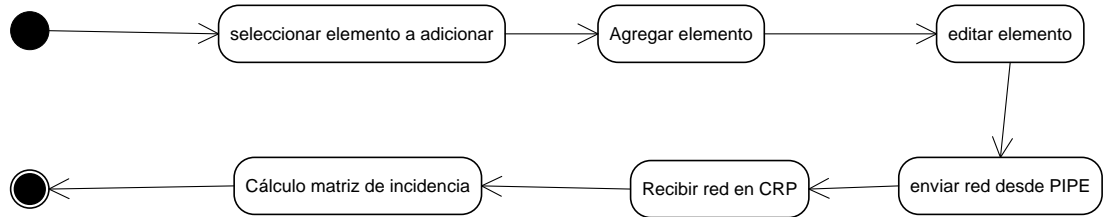
## Diagrama de Casos de uso.



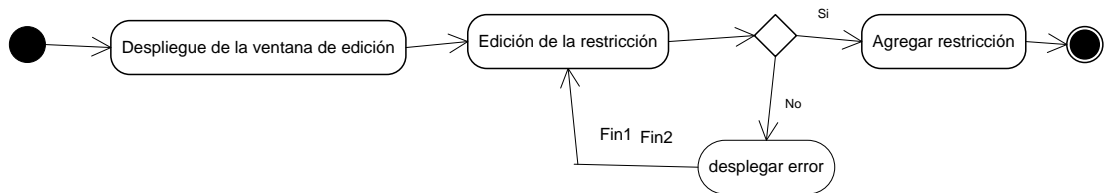


## Diagramas de actividades.

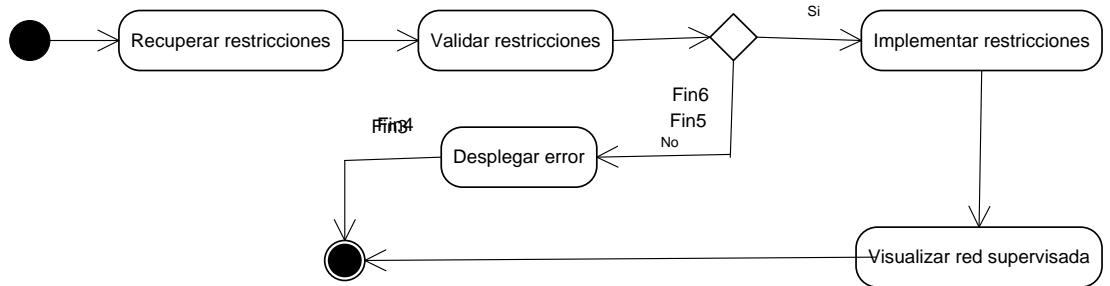
- Creación del modelo de la planta



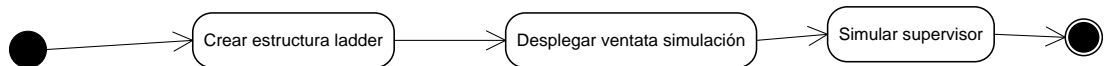
- Adicionar restricciones.



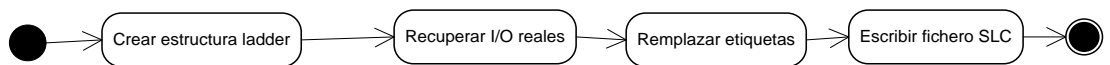
- Generar supervisor



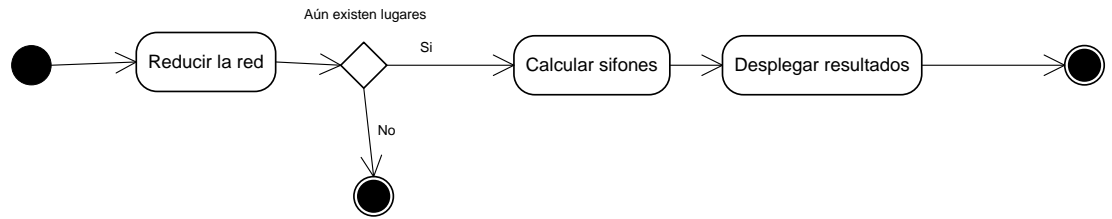
- Simular supervisor.



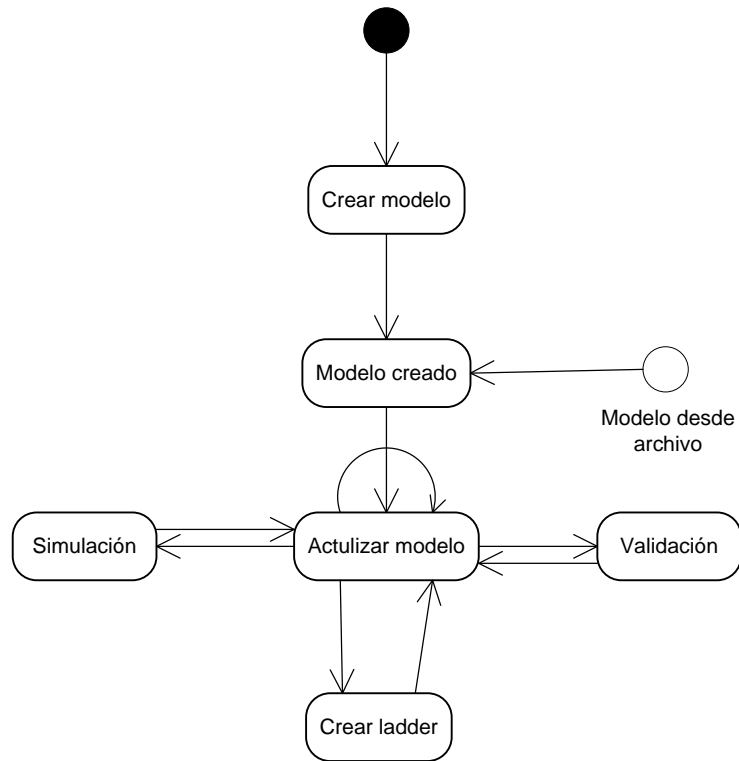
- Crear código ladder.



- Calcular sifones.

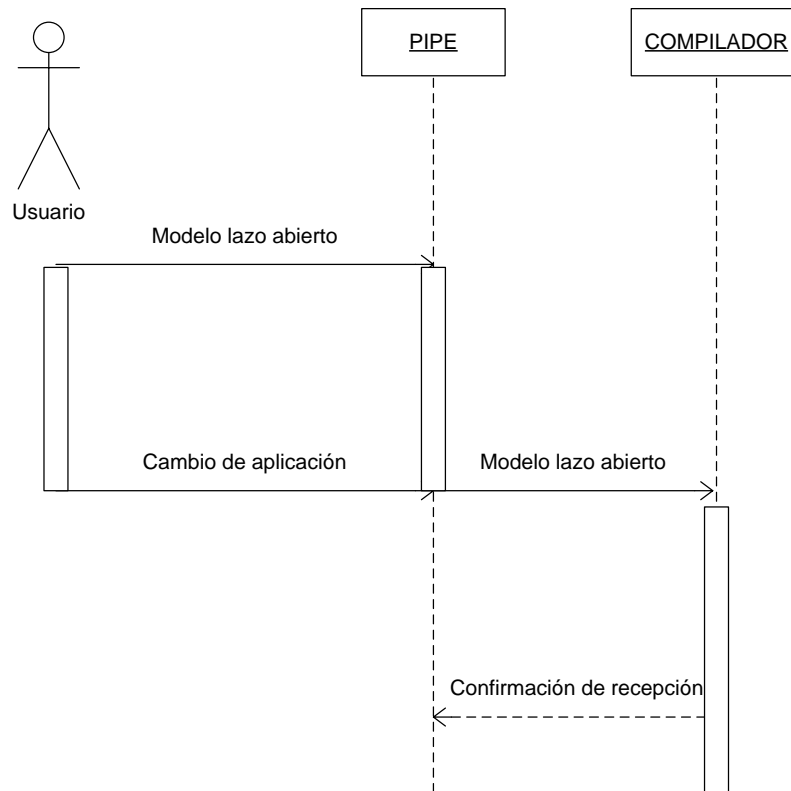


**Diagrama de estados de la aplicación.**

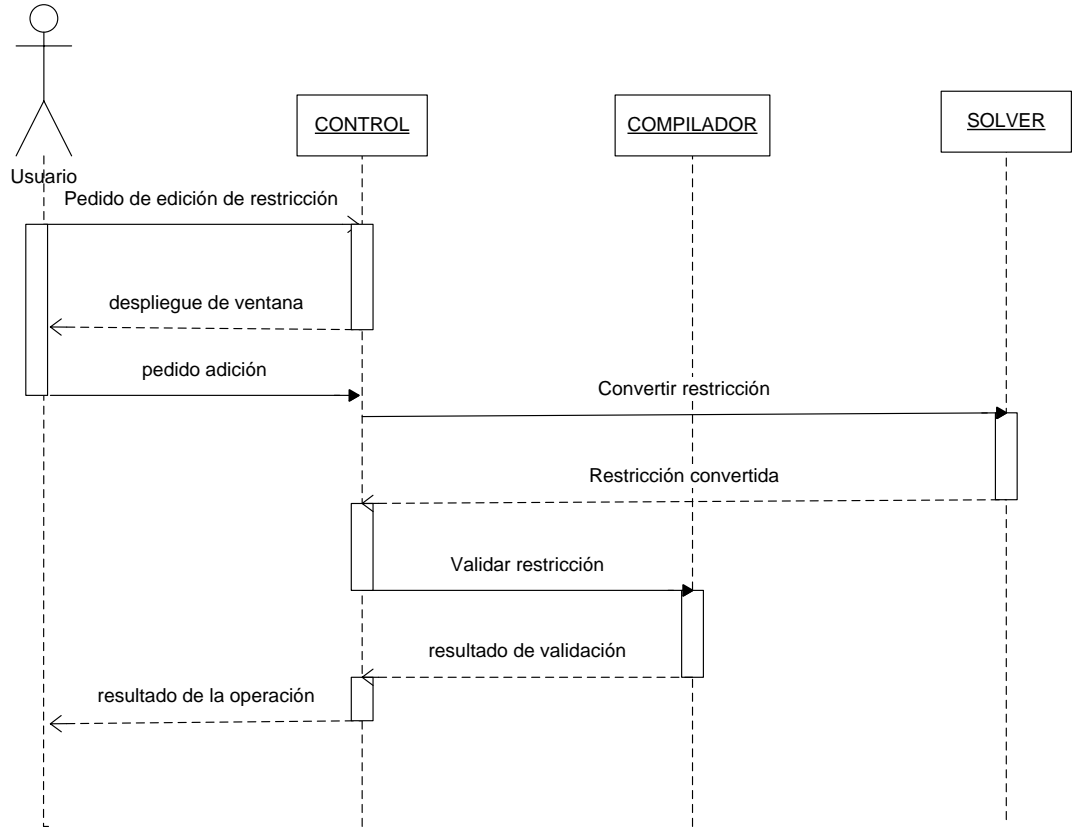


## Diagramas de secuencia.

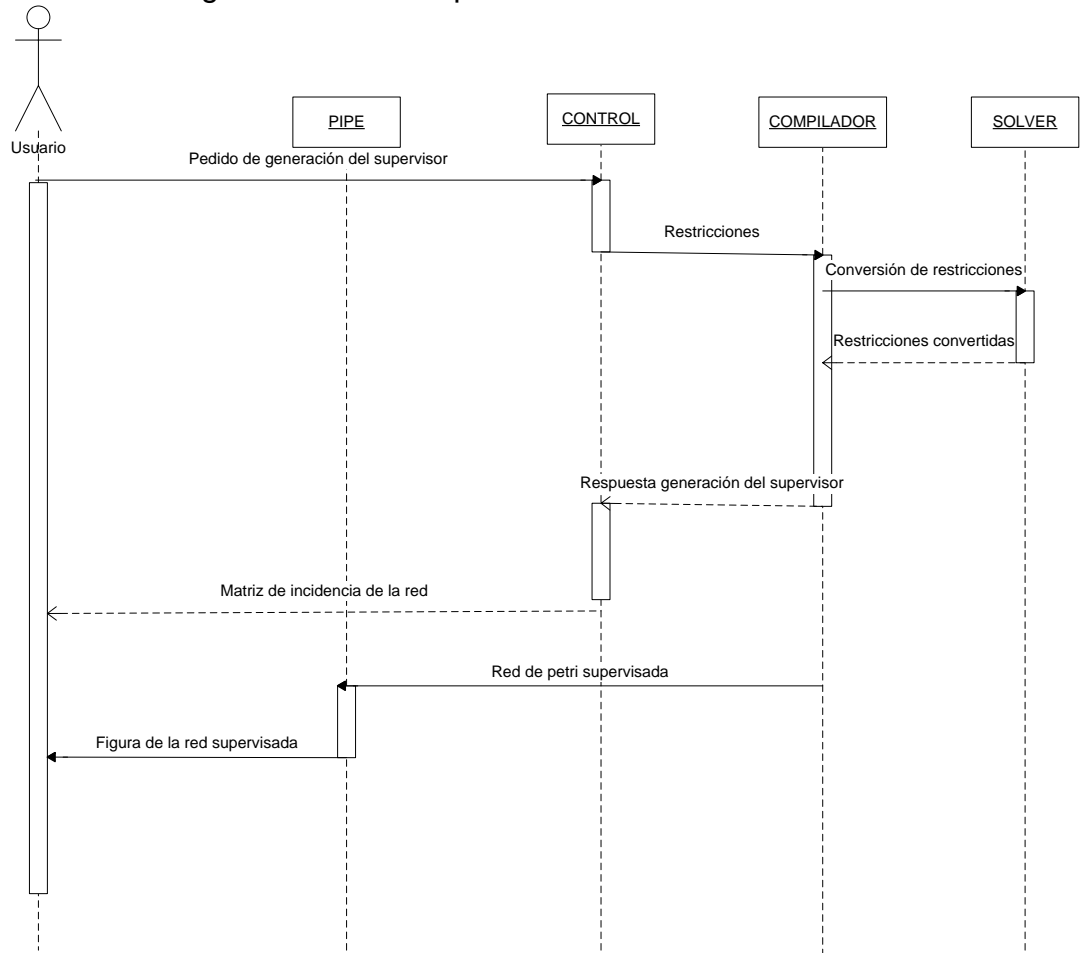
- Secuencia de intercambio de redes de Petri.



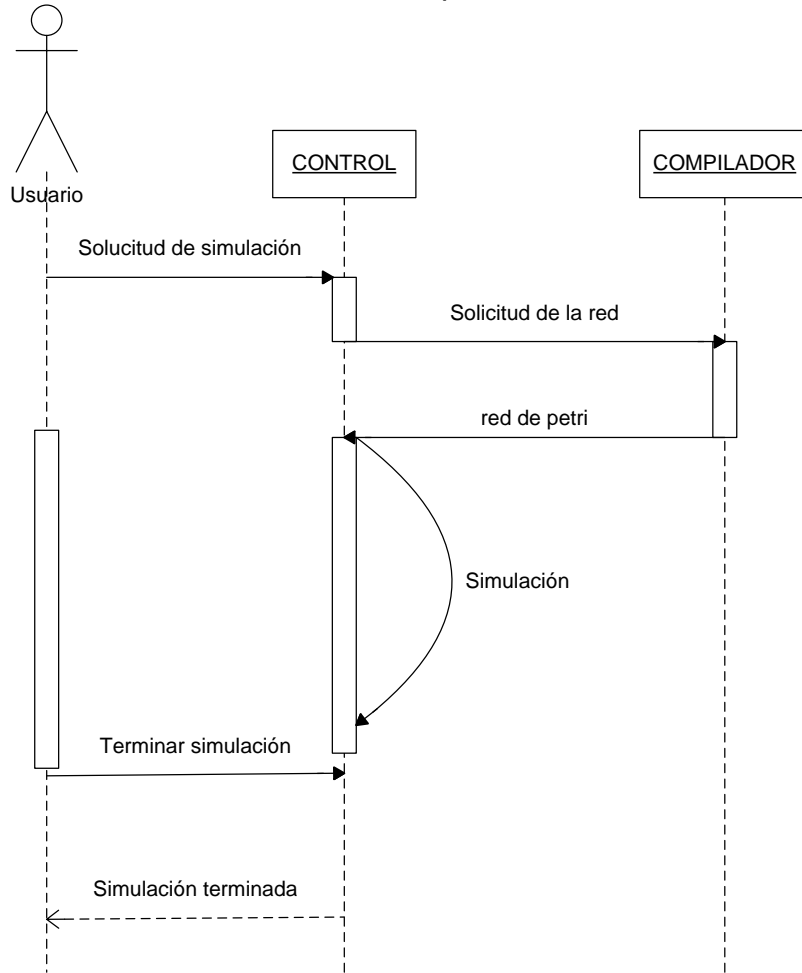
- Secuencia de adición de restricciones



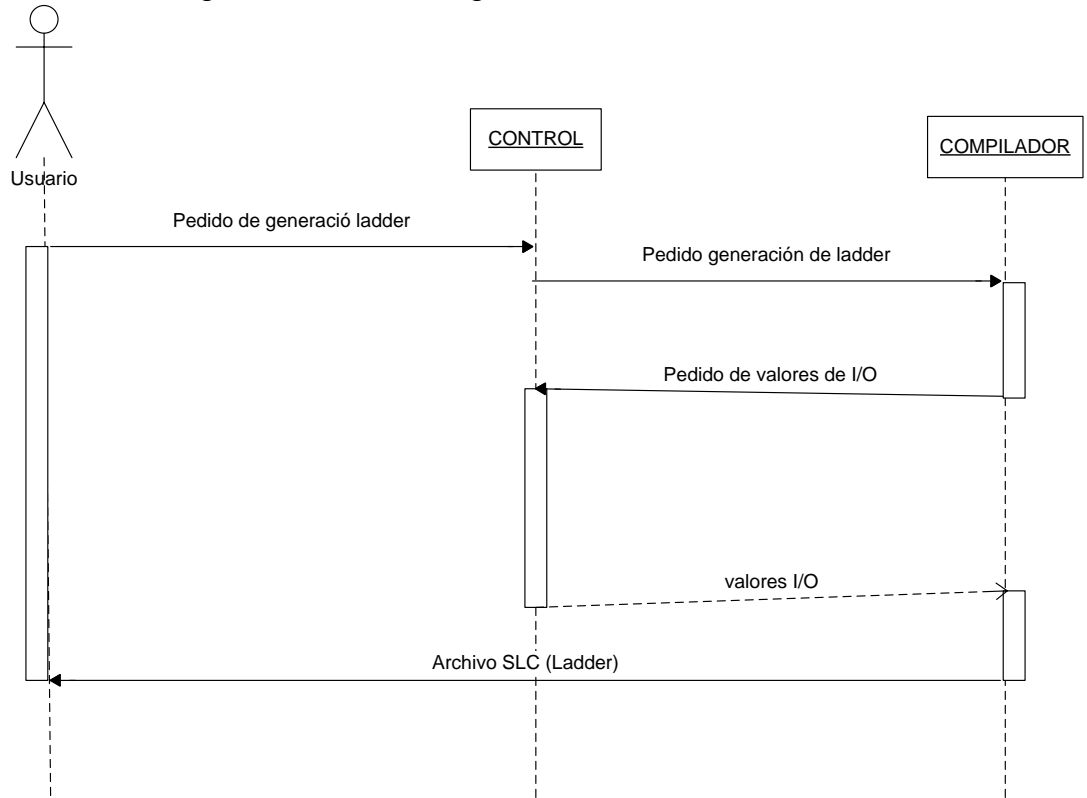
- Secuencia de generación del supervisor



- Secuencia de simulación del supervisor



- Secuencia de generación de código ladder



- Secuencia de cálculo de sifones.

