

**SISTEMA DE SUPERVISIÓN PARA EL PROCESO DE MEDICIÓN
DINÁMICA DE HIDROCARBUROS PARA LA EMPRESA SWCOL
LTDA.**



**YENY CRISTINA ZEA YANZA
CRISTIAN ALFONSO CAMPO**

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

**SISTEMA DE SUPERVISION PARA EL PROCESO DE MEDICION
DINAMICA DE HIDROCARBUROS PARA LA EMPRESA SWCOL
LTDA.**

ANEXOS



**YENY CRISTINA ZEA YANZA
CRISTIAN ALFONSO CAMPO**

**Director
Ingeniero. OSCAR AMAURY ROJAS A.**

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

UNIVERSIDAD DEL CAUCA	1
UNIVERSIDAD DEL CAUCA	2
1. ANEXO A. CONCEPTOS GENERALES	5
□ DENSIDAD	5
□ PRESIÓN.....	5
□ TEMPERATURA.....	6
□ TIPOS DE VOLUMEN.....	6
1.2. COMPUTADOR DE FLUJO OMNI 3000	7
1.2.2. ARQUITECTURA COMPUTADOR DE FLUJO OMNI 3000.....	9
1.2.2.1. BACKPLANE:	9
1.2.2.2. TARJETA CENTRAL DE PROCESO	9
1.2.2.4. BORNERAS TERMINALES.....	10
1.2.2.5. TARJETA ENTRADA / SALIDAS.	11
1.2.2.6. TARJETA DE ALIMENTACIÓN (POTENCIA).	14
1.2.2.7. PANEL FRONTAL.....	15
1.3. RESTRICCIONES QUE INCLUYEN VECTOR DE DISPARO.....	15
2. ANEXO B. CADENA DE VALOR PARA LA EMPRESA SWCOL LTDA ...	17
2.1. PROCESO DE LOGÍSTICA Y MANTENIMIENTO	19
2.1.1. LOGÍSTICA INTERNA.....	20
2.1.1.1 PROCESO DE RECEPCIÓN DE COMBUSTIBLE	20
2.1.2. LOGÍSTICA EXTERNA	20
2.1.3. <i>Venta de combustible</i>	21
2.1.4. <i>Proceso de desarrollo de tecnología</i>	22
2.1.5. <i>Proceso de Administración de recursos humanos</i>	22
2.1.6. <i>Proceso de infraestructura empresarial</i>	22
3. ANEXO C. MANUAL DE USUARIO DEL SOFTWARE PIPE V4	24
3.1. PIPE	24
3.2. EJECUTAR.....	24
3.3. ENTORNO DE TRABAJO.....	24
3.4. CREAR UNA RED DE PETRI	25
3.4.1. <i>Crear lugar</i>	25
3.4.2. <i>Asignar nombre a un lugar</i>	25
3.4.3. <i>Asignar un marcado inicial a un lugar</i>	25
3.4.4. <i>Crear transición</i>	26
3.4.5. <i>Asignar nombre a una transición</i>	26
3.4.6. <i>Crear arcos entre los lugares y las transiciones</i>	26
3.5. SIMULAR RED DE PETRI CREADA	27
3.6. OBTENER LA MATRIZ DE INCIDENCIA Y EL MARCADO DE UNA RDP.....	28
4. ANEXO D. SUPERVISOR COMPLETO DEL CASO DE ESTUDIO	30
4.1. DESARROLLO DEL SUPERVISOR BASADO EN INVARIANTES DE LUGAR.....	30
4.1.1. <i>Restricciones del sistema</i>	30
4.1.2. <i>Código utilizado en Matlab</i>	31
4.2. DESARROLLO DEL SUPERVISOR CON RESTRICCIONES DE DISPARO.....	32
4.2.1. <i>Restricciones del sistema</i>	32
4.2.2. <i>Desarrollo del supervisor con restricción de disparo 2</i>	33
4.2.3. <i>Desarrollo del supervisor con restricción de disparo 3</i>	34
4.2.4. <i>Desarrollo del supervisor con restricción de disparo 4</i>	36
4.2.5. <i>Desarrollo del supervisor con restricción de disparo 5</i>	37
4.2.6. <i>Desarrollo del supervisor con restricción de disparo 6</i>	39

4.2.7. Desarrollo del supervisor con restricción de disparo 7.....	40
4.2.7. Desarrollo del supervisor con restricción de disparo 8.....	41
4.2.8. Desarrollo del supervisor con restricción de disparo 9.....	43
5. ANEXO E. CODIGOS DE LAS FUNCIONES EN MATLAB	45
5.1. FUNCIONES TOMADAS	45
6. ANEXO F. GENERACION DEL LADDER.....	47
6.1. EJECUTAR	47
6.2. ADICIÓN DE LA RDP	48
6.3. ADICIÓN DE RESTRICCIONES	50
6.4. GUARDAR TRABAJO	53
6.5. GENERAR UN ARCHIVO PLC	53
6.6. CONFIGURACIÓN DEL RSLOGIX 500.....	54
6.7. GENERAR CÓDIGO LADDER PARA UN PLC MICROLOGIX 1500	58
6.8. DEFINICIÓN DE I/O EN PIPE	60
6.8.1. Definir entradas	60
6.8.2. Definir salidas	60
6.9. ASIGNACIÓN DE ENTRADAS Y SALIDAS DEL PLC	60
7. ANEXO G. MANUAL DE USUARIO DEL SOFTWARE INTOUCH 10.1	62
7.2. INTOUCH PARA EL DISEÑO DE UN HMI.....	62
7.3. INTOUCH 10.1.....	63
7.3.1. Tipo de ventanas:	63
7.3.2. Creación de variables	63
7.3.3. Scripts.....	64
7.3.4. Animation Links.....	64
7.3.5. Direccionamiento de I/O en intouch.....	64
7.3.6. Wonderware Intouch Server	64
7.4. CARACTERÍSTICAS DE INTOUCH.....	64
7.5. ENTORNO INTOUCH 10.1.....	65
7.5.1. Elementos de WindowMaker	66
7.1. CONFIGURACIÓN DE UN NUEVO PROYECTO	68
7.2. ENTORNO DE TRABAJO DE INTOUCH “WINDOWMAKER”	71
7.2.1 Wizards	72
7.3. CREACIÓN DE UNA VENTANA	73
7.4. DEFINICIÓN DE TAGNAME	74
7.5. ANIMACIÓN LINKS	76
7.5.1. Animación objetos.....	77
7.6. ANIMACIÓN CON WIZARDS	80
7.7. ALARMAS.....	83
7.8. CURVAS HISTÓRICAS Y RALES	84
7.9. PROGRAMACIÓN DE LÓGICA SCRIPT	85
7.9.1. Tipos de Scripts	85
8. ANEXO H. COMUNICACIÓN ENTRE INTOUCH Y RSLOGIX 500.....	86
8.2. RSLINX.....	87
8.3. RSLOGIX EMULATE 500	92
8.4. DESCARGA DEL PROGRAMA AL PLC VIRTUAL.....	93
9. ANEXO I. CODIGO LADDER DEL PROYECTO.....	95

1. ANEXO A. CONCEPTOS GENERALES

2.1. Condiciones generales y vocabulario

- **Densidad**

Es la relación de la masa de un líquido con respecto a su volumen [21].

$$Densidad = \frac{Masa}{Volumen}$$

- **Densidad relativa o gravedad específica.**

La densidad relativa, es la relación de la densidad de un líquido con respecto a la densidad del agua a 4°C [11].

Realizar esta medición en el crudo, permite concluir que cuanto más liviano es el crudo y más alta es su gravedad API, mejor es su calidad y mayor es su precio [21].

$$Gravedad\ específica\ del\ fluido = \frac{Densidad\ del\ líquido}{Densidad\ H_2O}$$

$$Densidad\ del\ agua = 999.012 \frac{Kg}{m^3} \text{ a } 60 \text{ } ^\circ F$$

La muestra de combustible es adquirida a través de un tomador de muestra; este procedimiento debe seguir la norma API MPMS capítulo 3 sección 1.A [5].

- **Presión.**

Es la fuerza aplicada o distribuida sobre una superficie, medida como fuerza por unidad de área [20].

$$Presión = \frac{Fuerza}{Area}$$

- **Presión absoluta.**

Es la presión por encima del vacío perfecto (0 PSI). La presión atmosférica es siempre expresada como una presión absoluta [21].

- **Presión manométrica.**

Es la presión absoluta menos la presión atmosférica [21].

- **Temperatura.**

La temperatura afecta muchas de las propiedades de los hidrocarburos líquidos, tales como: gravedad, presión de vapor, compresibilidad y volumen.

La temperatura no cambia el peso de un líquido en un recipiente, pero si cambia su volumen [21].

- **Tipos de Volumen.**

- **VI (Volumen Indicado).** Es el volumen marcado, registrado o leído directamente en un medidor durante una entrega, recibo o despacho.

$$VI = \text{Lectura final} - \text{lectura inicial}$$

$$VI = \frac{\text{Acumulado de pulsos}}{KF}$$

KF = Pulsos por unidad de volumen generados por el medidor. Dato suministrado por el fabricante del medidor [21].

- **VB (Volumen Bruto).** Volumen real en el momento en que se mide. Es el VI, corregido por el factor de calibración del medidor MF (*Meter Factor*).

El Volumen Bruto, es la cantidad física real transportada y/o bombeada [20].

$$V_B = VI * MF$$

$$V_B = \frac{\text{Acumulado de pulsos}}{KF} * MF$$

- **Vs (Volumen estándar).** Volumen a condiciones estándar: 60 grados y 0 PSI. Es el equivalente del VB a condiciones base de temperatura y presión, sin descontar agua y sedimento [21].

$$V_S = V_B * CTL * CPL$$

$$V_S = \frac{\text{Acumulado de pulsos}}{KF} * MF * CTL * CPL$$

CTL: Factor de corrección de temperatura

CPL: Factor de Corrección de presión

Los factores de corrección de volumen de líquidos que han sido definidos por la API, son empleados para tomar en cuenta los cambios de volumen de los líquidos debido a cambios de temperatura y presión.

- **V_N (Volumen Neto).** Es el V_S , descontado el contenido de agua y sedimento [20]

$$V_N = V_S * CSW$$

$$V_N = \frac{\text{Acumuladodepulsos}}{KF} * MF * CTL * CPL * CSW$$

CTL: Factor de corrección de temperatura

CPL: Factor de Corrección de presión

CSW: Cantidad neto de crudo

En la Figura 1, se puede apreciar las condiciones reales del volumen dentro de un depósito cilíndrico, en donde dicho volumen, se encuentra conformado por el líquido y los sedimentos dentro del cilindro.

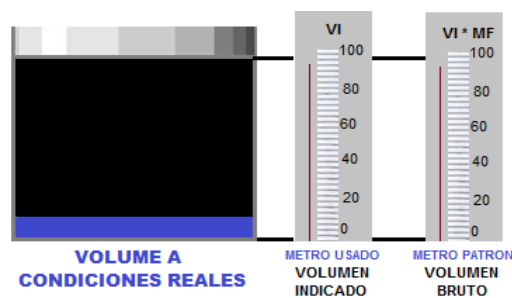


Figura 1. Volumen a condiciones reales

- **Caudal.**

El caudal es la cantidad de fluido que pasa en una unidad de tiempo. Normalmente, se identifica con el flujo volumétrico o volumen que pasa por un área dada en la unidad de tiempo; menos frecuentemente, se identifica con el flujo másico o masa que pasa por un área por unidad de tiempo [21].

1.2. Computador de flujo OMNI 3000

Un computador de flujo es un equipo diseñado y desarrollado para realizar todos los procedimientos y cálculos requeridos para la medición dinámica, este es uno de los equipos que se utiliza para certificar los volúmenes de producto que se recibe o se entrega en custodia ya sea para ser procesado y/o transportado. Básicamente es un Control lógico

programable desarrollado para cumplir con la tarea específica de la medición dinámica. Se compone de las siguientes partes:

- Unidad central de Procesos. CPU.
- Tarjetas E/S: en estas tarjetas se conectan físicamente toda la Instrumentación de cada uno de los brazos (Transmisor de flujo, transmisor de temperatura, transmisor de presión, densidad etc), la instrumentación del probador dependiendo del tipo de probador que se tenga etc.
- Tarjetas de Comunicación: se tienen protocolos de comunicación como Modbus, algunos puertos se pueden ser utilizados para comunicarse con otros dispositivos, tales como un HMI, PLC, etc.
- Display de visualización: se utiliza para visualizar todos los parámetros que posee este equipo.
- Teclado de operación.

1.2.1. Computador de flujo OMNI 3000

Del fabricante OMNI Flow Computers Inc, es un computador de flujo seguro, fácil de usar. Es programado, para poder tener una configuración de uno o varios Patines de Medición, dependiendo de su programación estos pueden medir diferentes productos como: crudos, refinados, gas natural, GLP, etc.

Su capacidad en cuanto a puertos de comunicaciones es de 2 puertos, las señales Entradas / Salidas dependiendo de la tarjeta son configurables como se detallara más adelante (se pueden conectar hasta dos brazos de medición):

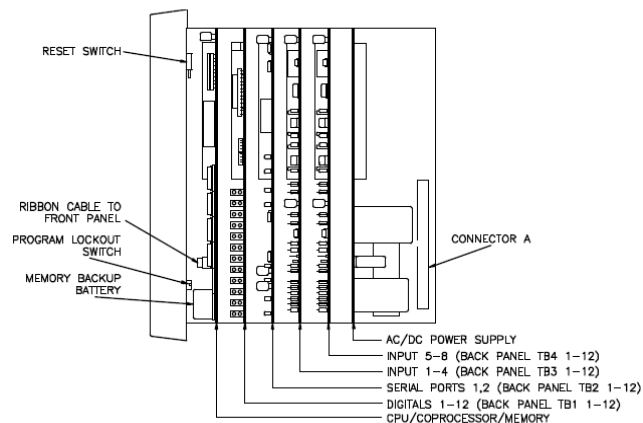


Figura 2. Esquema de OMNI 3000

1.2.2. Arquitectura computador de flujo OMNI 3000

El computador de flujo OMNI está compuesto por las siguientes partes:

- Backplane
- Tarjeta Central de Proceso
- Borneras Terminales
- Tarjetas Entradas / Salidas
- Tarjeta de Alimentación (Potencia)
- Panel Frontal

A continuación se realizara una descripción de cada una de las Partes:

1.2.2.1. Backplane:

El Backplane es la parte del computador de flujo donde se aloja la tarjeta principal (Mother Board), y sobre esta se instalan todas las tarjetas que requiere el computador de flujo, existen dos tipos de montaje de este Backplane:

- **Backplane compacto:** este tipo de montaje es cuando el panel frontal está conectado directamente con el backplane.
- **Backplane extendido:** este tipo de montaje es cuando el backplane no está conectado directamente con el panel frontal esta conexión se realiza a través de una cinta ribbon.

1.2.2.2. Tarjeta Central de Proceso

Esta tarjeta es el cerebro del computador de flujo ya que en ella se encuentra el microprocesador de 16/32 bits de Motorola que operan a 16MHz, un máximo de 512 Kbyte de memoria SRAM, 1 MB de memoria EPROM memoria del programa, coprocesador matemático y la hora. U3 y U4 contienen el programa de EPROMs. En estas memorias esta todo el Firmware que viene pre programado de fabrica.

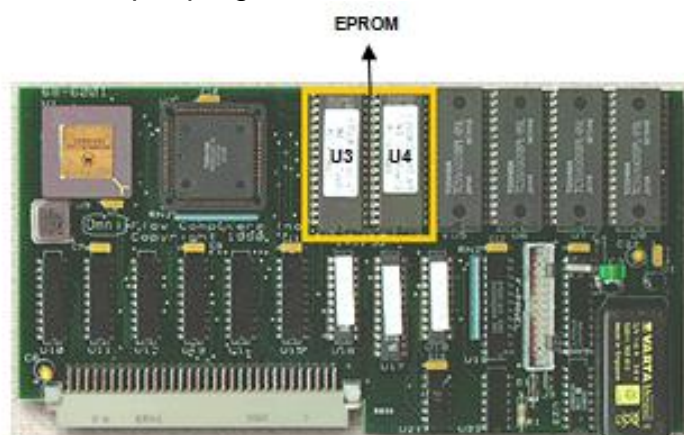


Figura 3. Tarjeta central de procesos

1.2.2.4. Borneras Terminales

Estas borneras son la conexión física de todos los instrumentos de campo que se conectan al computador de flujo, también se conectan los equipos que se comunican con el computador de flujo, alimentación para energizar el computador de flujo. Estas borneras tienen dos tipos de montaje:

- **Borneras Terminales compacto:** Este tipo de montaje se emplea cuando las borneras terminales están conectadas directamente con el backplane.

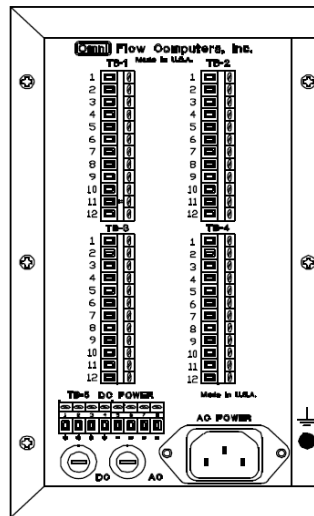


Figura 4. Borneras Terminales Compactas OMNI 3000

- **Borneras Terminales extendidas:** Este tipo de montaje es cuando el backplane no está conectado directamente con las borneras terminales esta conexión se realiza a través de una cinta ribbon.

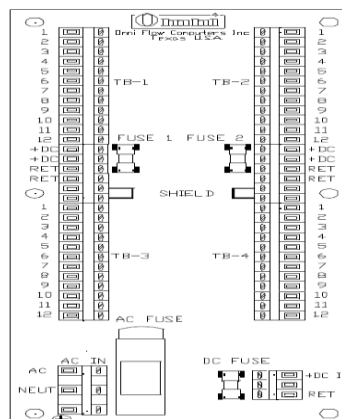


Figura 5. Borneras Terminales Extendidas OMNI 3000

1.2.2.5. Tarjeta Entrada / Salidas.

El computador de flujo Omni, tiene la posibilidad de montar hasta tres tipos de tarjetas Entradas / Salidas:

- Digital I/O Modules
- Serial I/O Modules
- Process I/O Combo Modules:

Casi cualquier combinación de Entrada / Salida puede tener la única limitante es el número de Slots de conexión en el Backplane, para el OMNI 3000 consta de 4 slots, pudiéndose conectar hasta 48 señales.

Para el OMNI 3000 que posee 4 slots, en el primero slot siempre se instala tarjeta central de proceso, la siguiente es numerada como slot #1 hasta slot #4 para instalar tarjeta entrada / salidas:

- Slot #1 Corresponde TB1 en borneras terminales.
- Slot #2 Corresponde TB2 en borneras terminales.
- Slot #3 Corresponde TB3 en borneras terminales.
- Slot #4 Corresponde TB4 en borneras terminales.

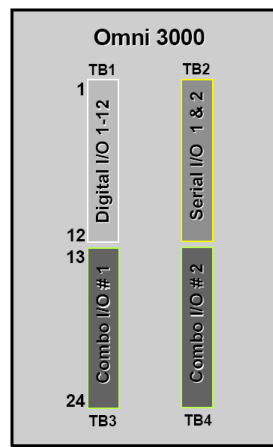


Figura 6. Ejemplo de posible distribución de Entradas / Salidas OMNI 300

- **Digital I/O Modulo:**

Este modulo tiene la posibilidad de por cada tarjeta poder conectar doce (12) señales discretas las cuales pueden ser configurables como entrada o salida. Cada señal tiene un fusible de protección y un LED indicador de que el punto esta activó. Por diseño de fábrica recomienda que el modulo digital siempre se instala en el slot 1 para el OMNI 3000.

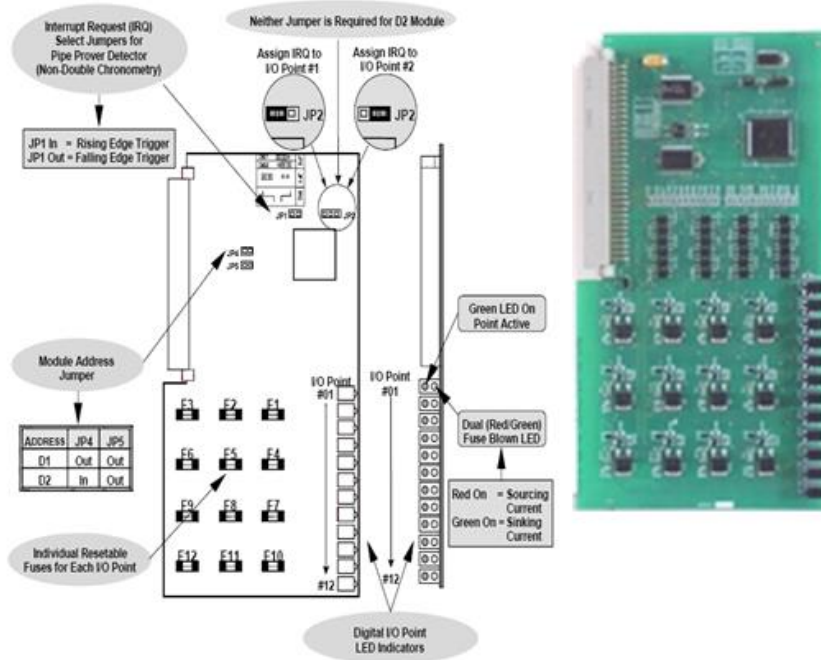


Figura 7. Digital I/O Modulo

- **Serial I/O Modules**

Dentro de los Módulos seriales se tienen diferentes tipos:

- RS-232/485 Serial I/O Module Model # 68-6205
- Serial\Ethernet I/O Modbus Mux Module Model # 68-6209

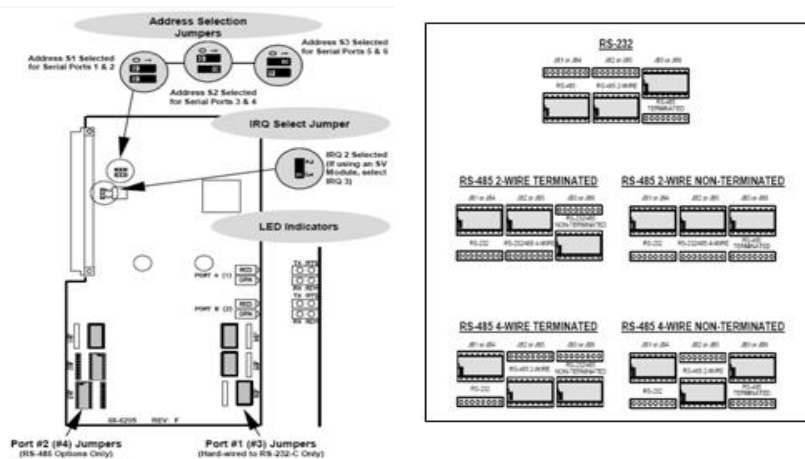


Figura 8. 232/485 Serial I/O Module

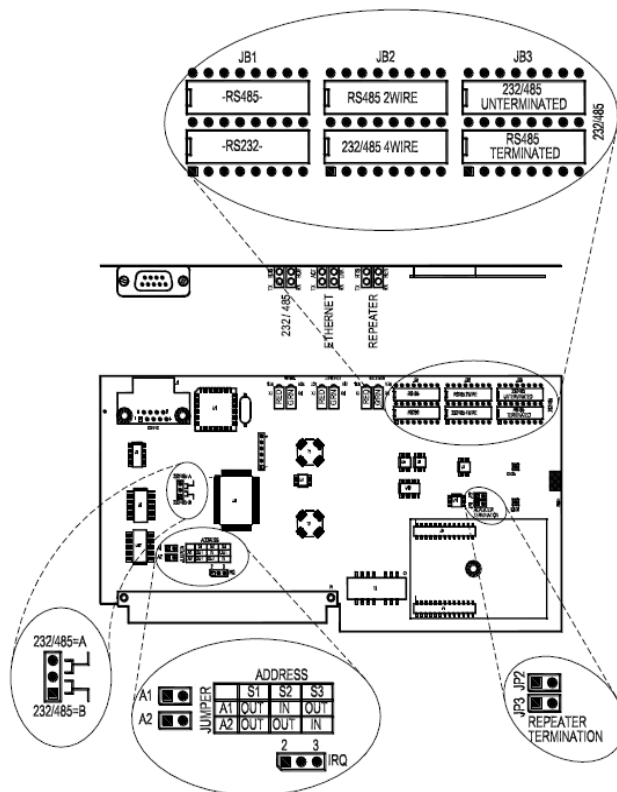


Figura 9. Serial\Ethernet I/O Modbus Mux Module

- **Process I/O Combo Modules**

Estos módulos tienen la posibilidad de capturar todas las señales de proceso tales como temperatura, presión, densidad y flujo. Cada módulo se puede encargar de 4 entradas de una variedad de tipos de señal y ofrece una o dos salidas analógicas 4-20 mA (Excepto el módulo SV, que cuenta con seis salidas de 4-20 mA). Estos módulos son llamados combos, existen siete tipos de combos disponibles: A, B, E, E/D, H, HV y SV.

Todos los módulos aceptan entradas analógicas y de frecuencia tipo pulso, a excepción de la H y HV módulos de interfaz digital para transmisores inteligentes de Honeywell y el modulo SV que es una interfaz serie RS-485 compatible para transmisores multivariable.

El combo A y B y el combo E y E/D son idénticos, a excepción de la posición de un puente de configuración que selecciona el tipo y la dirección de cada módulo.

Tabla 1. Detalle de Entradas / Salidas de Cada Combo

INPUT/OUTPUT CAPABILITIES AND FEATURES OF EACH I/O COMBO MODULE TYPE							
TYPE	INPUT #1	INPUT #2	INPUT #3	INPUT #4	ANALOG OUTPUTS	LEVEL A FIDELITY	DOUBLE CHRONOMETRY PROVING
A	1-5v; 4-20mA; RTD		1-5v; 4-20mA; Flow Pulses		Two 4-20mA	No	No
B	1-5v; 4-20mA; RTD		1-5v; 4-20mA Flow Pulse	Frequency Density	One 4-20mA	No	No
E/D	1-5v; 4-20mA; RTD		Frequency Density		Two 4-20mA	No	No
E	1-5v; 4-20mA; RTD		Flow Pulses		Two 4-20mA	Yes	Yes
H	Honeywell DE Protocol				Two 4-20mA	No	No
HV	Honeywell Multivariable DE Protocol				Two 4-20mA	No	No
	PORT #1		PORT #2				
SV	RS-485 Multi-drop to Various Multivariable Transmitters				Six 4-20mA	No	No

1.2.2.6. Tarjeta de Alimentación (Potencia).

El computador de flujo OMNI 3000, tiene la posibilidad de ser energizado con AC o DC.

Cuando se energiza con corriente alterna, se le debe aplicar 120 VAC 50 Watts, esta se debe conectar en los terminales identificados para esto en las borneras terminales ya sea compacta o externa. Cuando es alimentado de esta forma, la fuente de regulada debe tener de 500 mA a 24 VDC.

Esta tarjeta de alimentación ocupa el último slot en el Backplane, en cualquier modelo ya sea OMNI 3000 u OMNI 6000

Cuando se energiza con corriente directa, se le debe aplicar de 18 a 30 VDC VAC, 50 Watts, esta se debe conectar en los terminales identificados para esto en las borneras terminales ya sea compacta o externa.

1.2.2.7. Panel frontal

El Panel frontal del computador de flujo OMNI 3000 cuenta con cuatro partes básicas a saber.

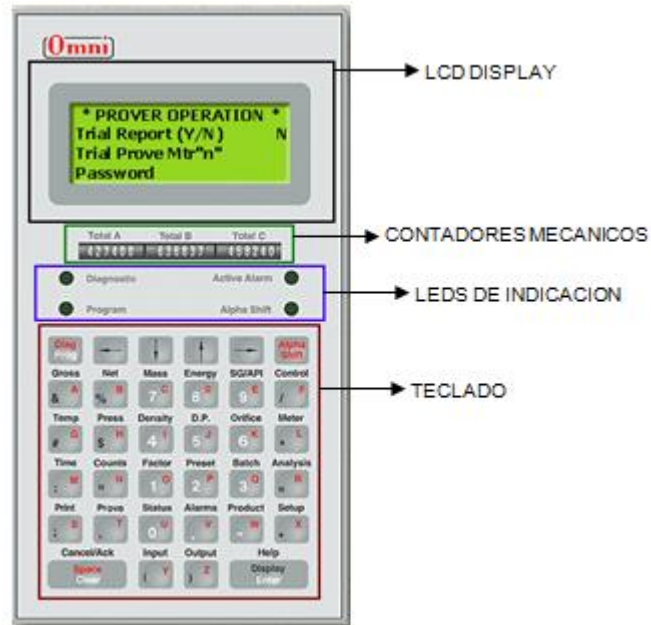


Figura 10. Panel frontal OMNI 3000

1.3. Restricciones que incluyen vector de disparo

Si se necesita que dos transiciones no se disparen simultáneamente, o que una transición no pueda dispararse cuando unos determinados lugares tengan marcas, estas restricciones pueden ser descritas de la siguiente forma [27].

$$\mu_i - q_j \leq 1$$

Si se asume que la planta debe satisfacer la anterior ecuación, la interpretación directa de esta restricción implica que la transición j no debe ser disparada si el lugar i tiene una marca. Para escribir esta restricción de forma que solamente contenga elementos del vector de disparo, la planta se transforma de la siguiente manera: la transición j es remplazada por dos transiciones y un lugar en medio de ellas, como se muestra en la Figura 11. La transición es artificial y no afectará el modelo de las Redes de Petri del sistema, su único propósito es introducir un lugar μ_j , el cual, almacenará el disparo de la transición t_j , luego de que el supervisor sea calculado, el sistema volverá a ser transformado a su estado original [27].

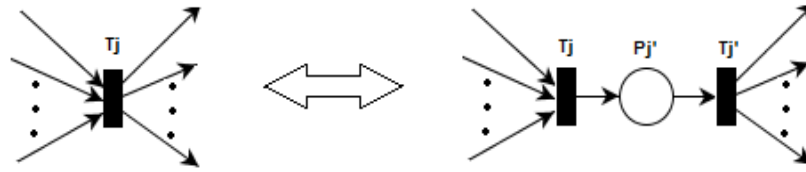


Figura 11. Transformación de una transición

El marcado μ_j' de P_j' reemplaza a q_j en la ecuación $\mu_i - q_j \leq 1$, la cual, se convierte en:

$$\mu_i - u_j' \leq 1$$

La restricción ahora sólo contiene μ_s y el controlador puede ser hallado a través de la forma descrita para invariantes de lugar. Luego de que éste sea calculado, las dos transiciones y el lugar son contraídos al lugar original, mientras se mantiene el cumplimiento de la restricción original.

Los arcos que salen del controlador irán normalmente conectados a la transición t_j y los arcos que entran al controlador irán conectados a la transición t_j' . El acto de contraer la estructura transformada, volverá la RDP del sistema a su forma original y los arcos de entrada y salida irán conectados a la transición t_j .

En resumen, dada la planta (D_p, μ_{p_0}) , siendo la matriz de incidencia del sistema transformado $D_p' \in Z^{n+1 \times m+1}$, el supervisor basado en RDP $D_c \in Z^{1 \times m}$, con marcado inicial $\mu_{c_0} \in Z^{n_c}$ puede ser calculado así:

$$D_c = -LD_p'$$

$$\mu_{c_0} = b - L\mu_{p_0}'$$

2. ANEXO B. CADENA DE VALOR PARA LA EMPRESA SWCOL LTDA

El uso de la cadena de valor sirve fundamentalmente, para la descripción del modo en cómo se desarrollan la dinámica de las acciones dentro de una empresa, iniciando desde la recepción y manejo de materias primas, pasando por la producción de un determinado producto, hasta llegar a su distribución y venta.

El estudio de la estructura de la cadena de valor, permite mejorar el desarrollo del proceso productivo, ya que se puede analizar detalladamente el funcionamiento y dinámica de la empresa paso a paso, permitiendo así establecer la reducción de costos, generando el aumento en la calidad y cantidad de la producción.

El modelo de la cadena de valor propuesto por Michael Porter, tiene como objetivo establecer detalladamente las distintas actividades que se realizan dentro de la empresa, permitiendo descomponer horizontalmente a la organización en sus partes constitutivas como: la dinámica de los flujos de conocimientos, productos, e informaciones entre las distintas unidades de producción del mismo nivel de la empresa, lo cual, conlleva a satisfacer óptimamente las necesidades del cliente y cumplir con los propósitos y metas establecidas por la organización.

La cadena de valor genérica se constituye por dos elementos básicos:

- Las actividades primarias.
- Las actividades de soporte a las actividades primarias.

El modelo basado en la cadena de valor, debe tener en cuenta el flujo de información que existe entre las distintas unidades de producción, con el fin de mantener y llevar un reporte sobre el producto a medida que éste se desplaza a través de las distintas unidades, permitiendo establecer su comportamiento y cambios a través de éstas. Dichos cambios, son seguidos por el nivel de coordinación, quien proporcionará el estado actual de cada unidad de producción [29]. En la Figura 12, se representa la composición de una unidad de producción.

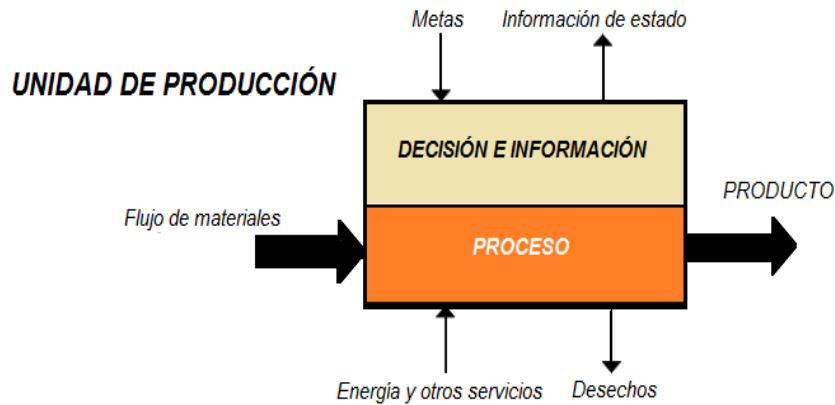


Figura 12. Representación de una unidad de producción

En la Figura 13, se muestra la cadena de valor para la empresa SWCOL Ltda., seguido de una descripción detallada de los distintos componentes que hacen parte de esta.

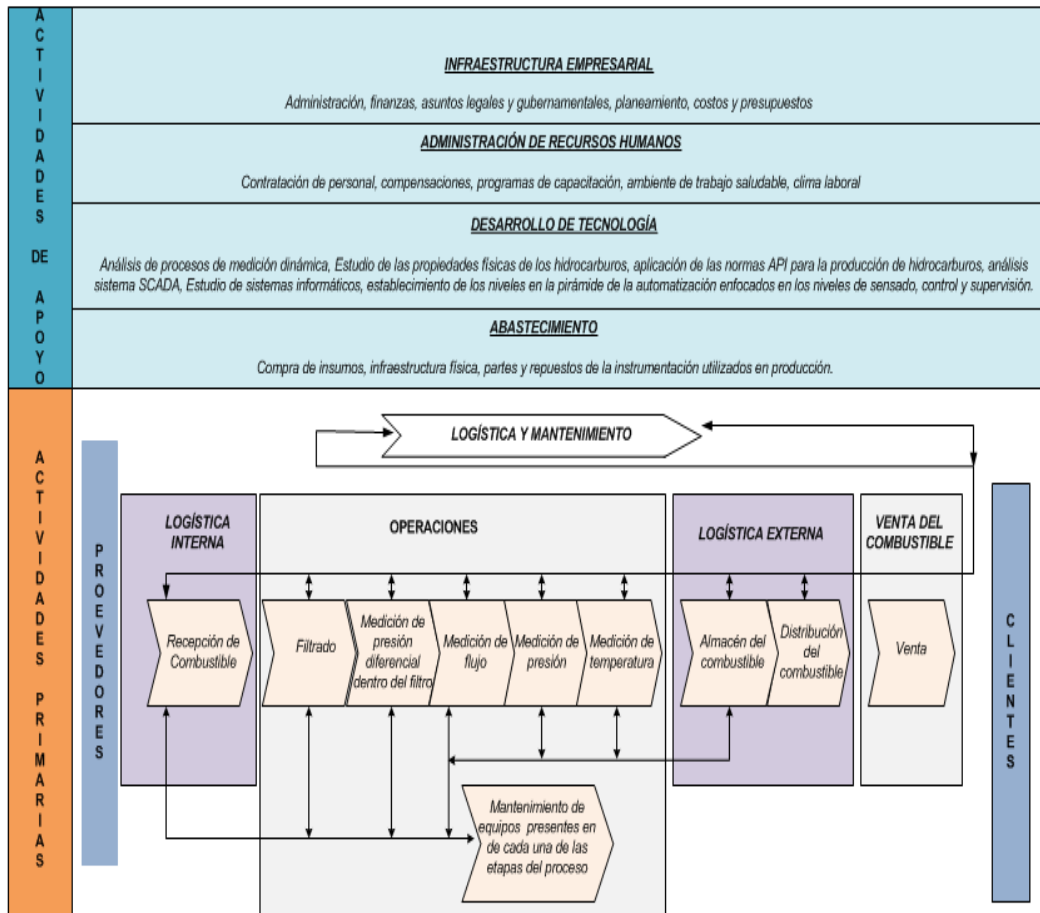


Figura 13. Cadena de valor de la empresa SWCOL Ltda.

A continuación, se describirá detalladamente cada eslabón perteneciente a la cadena de valor de la Figura 13.

2.1. Proceso de logística y mantenimiento

Este proceso perteneciente a la cadena de valor de la empresa SWCOL Ltda., cobra gran importancia, ya que sobre éste se apoyan y realizan las actividades de producción, a través del reporte de cada una de sus etapas. Cada etapa nace y se comporta como una unidad de producción en donde entra una determinada materia prima, se establecen condiciones y objetivos sobre cada una de éstas, para luego obtener una salida que resultaría siendo el producto final.

Como puede observarse en la Figura 13, el proceso de logística y mantenimiento se relaciona directamente con la gran mayoría de actividades primarias entre ellas: logística interna, operaciones y logística externa.

- Recepción de combustible.
- Filtrado.
- Medición de presión diferencial dentro del filtro.
- Medición de flujo.
- Medición de presión.
- Medición de temperatura.
- Almacén de combustible.
- Distribución del combustible.

Objetos de negocios para el proceso de logística y mantenimiento de la cadena de valor de la empresa SWCOL Ltda.

- Estado de órdenes de mantenimiento, tanto de los instrumentos de medición, como de los equipos y la infraestructura del complejo industrial.
- Estado de órdenes de trabajo, al momento de realizar una actividad específica para la medición dinámica de hidrocarburos líquidos.
- Estado de órdenes del crudo que será extraído desde los yacimientos.
- Estado de órdenes de los barriles de crudo previamente medidos volumétricamente.
- Estado de órdenes de los barriles de crudo depositados en los silos de distribución y venta.
- Estado de órdenes de despacho del crudo que previamente será distribuido.
- Reporte de fallas dentro del complejo industrial.
- Reporte de tiempos utilizados en cada etapa del proceso de medición dinámica de hidrocarburos.
- Planes de mantenimiento.
- Reporte y seguimiento del estado actual de cada uno de los instrumentos de medición y equipos dentro complejo industrial.
- Estado del proceso de mantenimiento y reparación de la instrumentación.

- Reporte estadístico de mantenimiento.
- Reporte de la recepción de combustible.
- Estado final del filtrado del hidrocarburo.
- Reporte de la medición de presión y temperatura del hidrocarburo con el fin de evitar mediciones erróneas o accidentes.

2.1.1. Logística interna

La logística interna hace parte de las actividades primarias de la cadena de valor, en ésta se describen las actividades relacionadas con la recepción, almacenamiento y distribución de materias primas hacia la etapa de operaciones, donde se hará posteriormente la producción o manejo del producto.

2.1.1.1 Proceso de Recepción de combustible

Este proceso corresponde a una etapa importante, ya que es desde este punto donde empieza el proceso de la medición dinámica del hidrocarburo, consiguiendo así, un producto final libre de impurezas que pueden afectar su composición y por ende su calidad.

Los hidrocarburos obtenidos en las áreas de producción son recolectados, procesados y transferidos hacia los patios de tanques, mientras que el gas es separado y enviado a las plantas de compresión para su posterior utilización.

La medición tiene una relación inmediata con todas las negociaciones que realiza la industria petrolera, sus clientes reciben el crudo y/o producto basado en las medidas realizadas, las cuales, deben ser efectuadas bajo las normas API y ASTM desarrolladas para tal caso. Para este proyecto en particular, el proceso se amplía a dos brazos de medición (patín de medición).

2.1.2. Logística Externa

En la logística externa, se encuentran las actividades relacionadas con la recopilación, almacenamiento y distribución física del producto final hacia los clientes y compradores.

2.1.2.1. Proceso de almacén del combustible

Una vez se haya filtrado y medido el combustible en movimiento, será almacenado en uno o más silos de gran tamaño para el depósito de grandes cantidades de barriles de crudo, esta actividad se realiza mediante la utilización de una bomba, la cual, se encarga de llevar el hidrocarburo desde la tubería de producción hacia los silos de almacenaje, estos volúmenes de combustible no permanecen gran tiempo sobre dichos silos, ya que serán entregados rápidamente hacia el proceso de venta.

Objetos de negocios para el proceso de almacén:

- Bomba auxiliar para el almacenaje del crudo.
- Válvulas reguladoras de caudal.
- Válvula de los silos de almacenaje.
- Nivel de los silos de almacenamiento.
- Temperatura del crudo almacenado.
- Presión del crudo almacenado.
- Informe y reporte de volumen almacenado.

2.1.2.2. Proceso de distribución del combustible

Una vez se ha almacenado el crudo bajo condiciones ideales, será entregado y distribuido hacia las terminales y medios de transporte, para luego ser entregado hacia los respectivos clientes.

Objetos de negocios para el proceso de distribución del combustible:

- Terminales de distribución
- Vehículos transportadores
- Anuncio de transporte
- Tiempos de distribución
- Estado del producto final
- Reporte sobre el número de barriles distribuidos

2.1.3. Venta de combustible

2.1.3.1. Proceso de venta

En el proceso de venta de combustibles, hacen parte las actividades que proporcionan un medio, por el cual, los clientes y compradores de combustibles puedan comprar el producto e inducir a hacerlo. En este proceso, también se realizan las actividades relacionadas con la tramitación y pedidos por parte de los respectivos clientes.

Objetos de negocio para el proceso de ventas de hidrocarburos:

- Clientes.
- Estado y reporte de facturas.
- Pedios y exigencias del cliente.
- Ofertas para los clientes.
- Información y reporte de inventario.
- Órdenes de expedición.
- Facturas.
- Reclamos e inconformidades.

2.1.4. Proceso de desarrollo de tecnología

Este proceso hace parte de las actividades de apoyo de la cadena de valor, cada actividad de valor representa tecnología, sea conocimientos, procedimientos, o tecnología dentro del proceso.

Objeto de negocios para el proceso de desarrollo de tecnología:

- Análisis de procesos de medición dinámica.
- Estudio de las propiedades físicas de los hidrocarburos.
- Aplicación de las normas API para la producción de hidrocarburos.
- Análisis sistema SCADA.
- Estudio de sistemas informáticos.
- Establecimiento de los niveles en la pirámide de la automatización enfocados en los niveles de sensado, control y supervisión.

2.1.5. Proceso de Administración de recursos humanos

Este proceso hace parte de las actividades de apoyo de la cadena de valor; tiene como fin, realizar y definir la búsqueda, contratación, entrenamientos y desarrollo de todo tipo de personal perteneciente a la empresa.

Objetos de negocio para el proceso de administración de recursos humanos:

- Estado de la contratación del recurso humano.
- Estado y reporte del comportamiento de cada trabajador.
- Registro de la cantidad de trabajadores.
- Registro de las tareas designadas a cada trabajador.
- Tiempo estimado para la realización de cada tarea indicada a cada trabajador.
- Reporte de evaluación de tareas implementadas por cada trabajador.
- Capacitación al personal.
- Clima laboral.

2.1.6. Proceso de infraestructura empresarial

Este proceso hace parte de las actividades de apoyo de la cadena de valor, y consiste en varias actividades como: administración general, planeación de finanzas, contabilidad, asuntos legales, planificación de producción, programación de la producción y lanzamiento de órdenes de trabajo.

Este proceso, se encarga de apoyar normalmente a toda la cadena de valor y no a actividades individuales, suministrando a los demás procesos

de la cadena de valor de la empresa la información de los planes de producción, mantenimiento y órdenes de trabajo que son interpretadas por cada una de las unidades de producción.

Objetos de negocios para el proceso de infraestructura empresarial:

- Planes de producción.
- Órdenes de producción.
- Órdenes de mantenimiento.
- Órdenes de flujo de trabajo.
- Órdenes de distribución.
- Órdenes de transporte.
- Órdenes de ventas.
- Órdenes de expedición.
- Reporte de los estados de los equipos.
- Planes de mantenimiento.
- Datos y registros estadísticos de producción.

Una vez explicado el concepto de unidades de producción y la cadena de valor propuesta para la empresa SWCOL Ltda., se realizará a continuación los aspectos más relevantes sobre las características del combustible y las exigencias del proceso de medición dinámica de hidrocarburos fundamentada bajo una serie de recomendaciones propuestas por la norma API.

3. ANEXO C. MANUAL DE USUARIO DEL SOFTWARE PIPE V4

3.1. PIPE

PIPE (Platform Independent Petri Net Editor) es una herramienta software libre desarrolla para el modelado, simulación y el análisis cualitativo de las Redes de Petri.

3.2. Ejecutar.

Primero se debe ejecutar el archivo llamado pipe.bat, con lo cual no hay necesidad de instalarlo o configurarlo. PIPE está ubicado en la carpeta donde se encuentra la aplicación pipe v4.



Figura 14. Icono de para ejecutar a PIPE

3.3. Entorno de trabajo

Al ejecutar PIPE aparecerá una ventana como se muestra en la Figura 15.

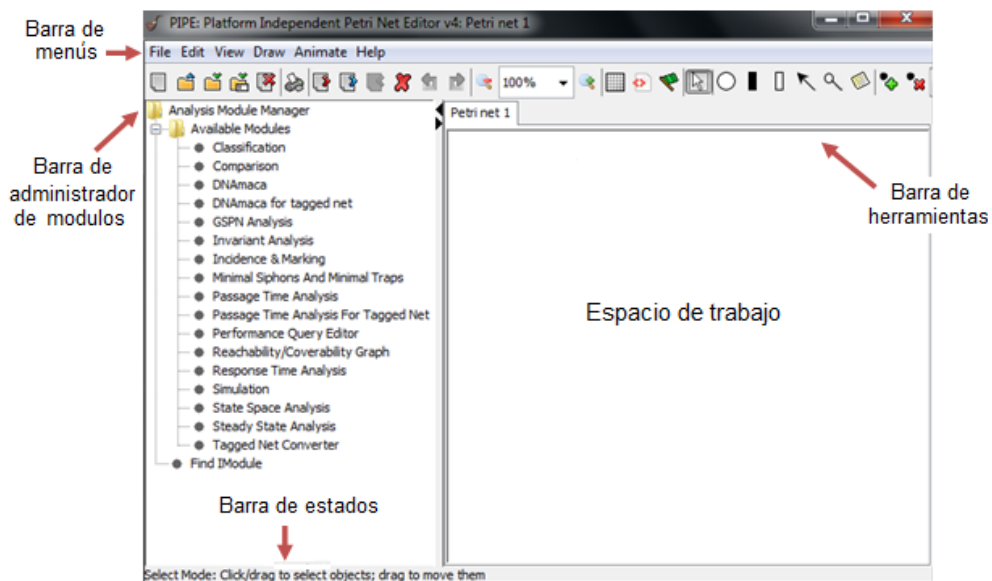


Figura 15. Entorno Software PIPE

3.4. Crear una Red de Petri

3.4.1. Crear lugar

Para crear un lugar dar clic en “add a place”, que se encuentra ubicado en la barra de herramientas, agregar la cantidad de lugares deseados en el espacio de trabajo. Como se muestra en la Figura 16.

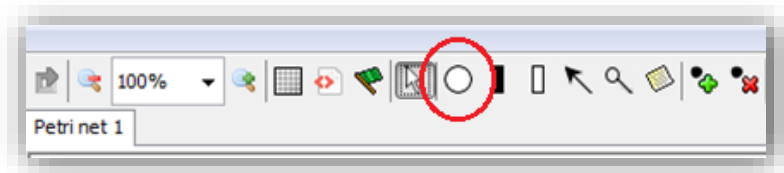


Figura 16. Añadir lugares

3.4.2. Asignar nombre a un lugar

Si desea cambiarle el nombre al lugar dar doble clic en el respectivo lugar, se desplegará una ventana donde se configura el nombre, el marcado y la capacidad como se muestra en la Figura 17.

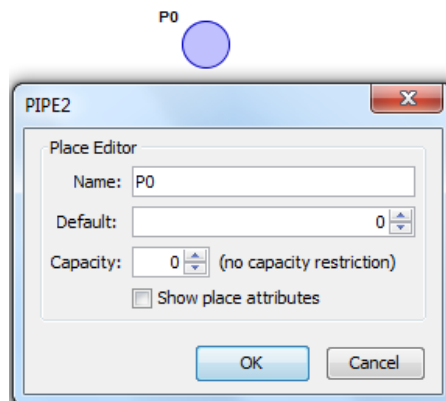




Figura 17. Configuración de un lugar

3.4.3. Asignar un marcado inicial a un lugar

Para asignar un marcado inicial a un lugar de una RdP en PIPE, se debe presionar clic sobre el botón  de la barra de herramientas, luego se presiona clic las veces necesarias sobre el lugar al cual se le desean agregar marcas. Si se desea eliminar marcas de un lugar se presiona clic sobre el botón  de la barra de herramientas y luego se presiona clic sobre el lugar en donde se desea eliminar las marcas.

3.4.4. Crear transición

Para agregar una transición dar clic en “add an immediate transition”, que se encuentra ubicado en la barra de herramientas, agregar la cantidad de transiciones necesarias en el espacio de trabajo. Como se muestra en la Figura 18.

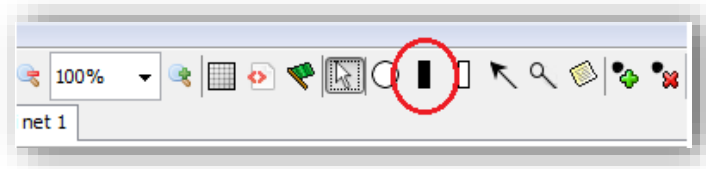


Figura 18. Añadir transiciones

3.4.5. Asignar nombre a una transición

Si desea cambiarle el nombre a la transición dar doble clic en la respectiva transición, se desplegará una ventana donde se configura el nombre, la capacidad y da la opción de escoger si dicha transición es inmediata o temporizada, como se muestra en la Figura 19.

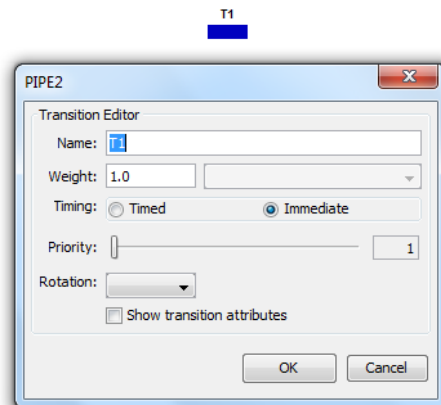


Figura 19. Configuración de una transición

3.4.6. Crear arcos entre los lugares y las transiciones

Después de haber creado los lugares y las transiciones se debe realizar las conexiones.

3.4.6.1. Agregar un arco

Para agregar un arco a la Red de Petri dar clic en “add an arc” que se encuentra ubicado en la barra de herramientas, debe llevar el cursor al lugar en el que se quiere que empiece el arco, donde se da un clic y luego sobre la transición en la que se desea que termine, PIPE por defecto asigna un peso de 1, si se desea modificar dicho peso, se presiona clic izquierdo sobre el arco y se selecciona la opción *Edit Weight* y se abrirá una ventana en la cual se puede ingresar el peso deseado, también es posible agregar puntos para dar movimiento al arco o por último borrarlo.



Figura 20. Añadir arcos

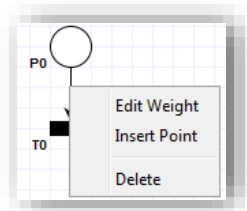


Figura 21. Configuración de un arco

3.5. Simular Red de Petri creada

Dar clic en “toggle animation mode” en la barra de herramientas. Ver Figura 22.

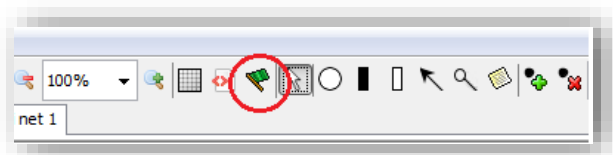


Figura 22. Simulación de la Red de Petri

Las transiciones activas cambiaran a color rojo, indicando que estas pueden ser disparadas, dar clic sobre ellas, una vez disparadas se moverán las marcas entre los lugares. Ver Figura 23.

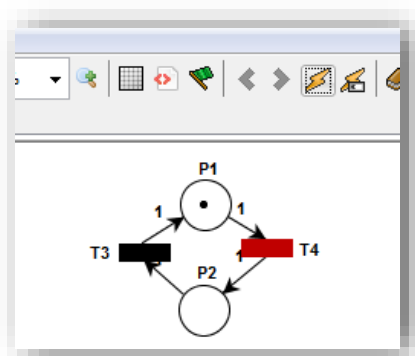


Figura 23. Cambio de color de las transiciones activas en modo de simulación

Para que cualquiera de las transiciones activas se dispare aleatoriamente dar clic en “randomly fire a transition”, como se muestra en la Figura 24.

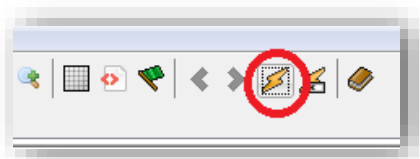


Figura 25. Simulación por selección de transición aleatoria

Existe además la posibilidad de disparar un número de transiciones deseado, presionando el botón “randomly fire a number of transition”, el cual abrirá una ventana donde se ingresa el número de disparos a realizar, luego de presionar *aceptar* se abrirá otra ventana en la cual se configura un tiempo de disparo entre las transiciones, finalmente al oprimir *aceptar* se inicia la serie de disparos. Ver Figura 26.

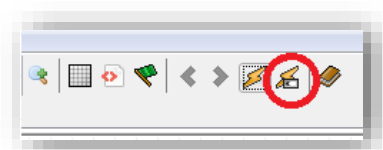


Figura 26. Simulación para disparar un número de transiciones deseado

3.6. Obtener la matriz de incidencia y el marcado de una RdP

En nuestro proyecto es imprescindible obtener la matriz de incidencia del sistema para el cálculo del supervisor. Dentro del administrador de módulos de análisis dar clic en una herramienta llamada Incidence & Marking, la cual permite calcular la matriz de incidencia y el marcado de la Red de Petri.

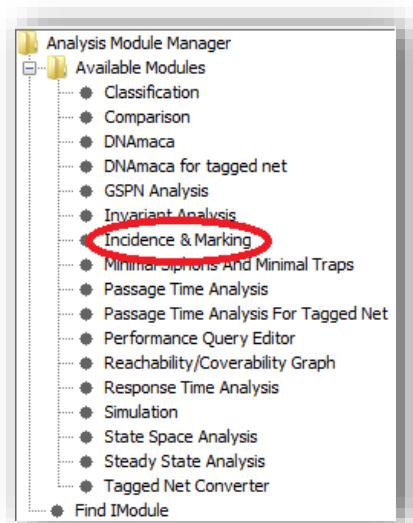


Figura 27. Administrador de herramientas de análisis

Se abrirá una ventana en blanco sobre la cual se debe presionar el botón *Calculate*. Como se muestra en la Figura 28

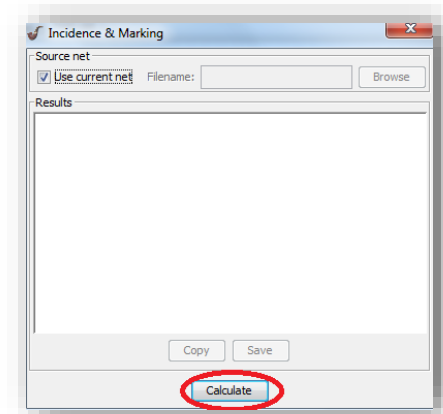


Figura 28. Herramienta de análisis Incidence & Marking

Esta herramienta calcula:

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

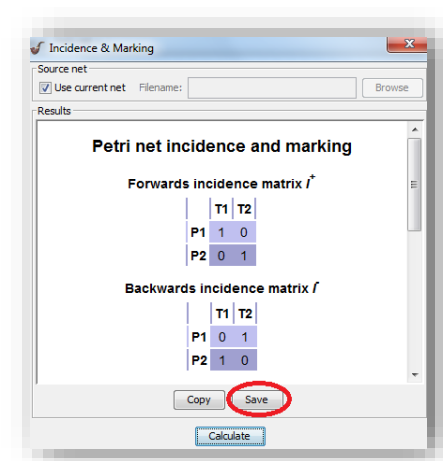


Figura 29. Resultado obtenido con la herramienta Incidence & Marking

Para copiar estas matrices como una imagen o guardarlas en una extensión .html dando clic en "Save" para poder ser leídas por otros programas. Ver figura 29.

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

4. ANEXO D. SUPERVISOR COMPLETO DEL CASO DE ESTUDIO

El objetivo del supervisor es asegurar que el filtro no se llene, ya que si esto sucede se detiene el paso del crudo en la tubería impidiendo la medición dinámica.

En este anexo se muestra el desarrollo del supervisor basado en invariantes de lugar de las restricciones 2, 3, 4, 5 y 6, la restricción 1 está desarrollada en la monografía del proyecto. Asimismo en el ítem 2.2 se muestra el desarrollo del supervisor con las restricciones de disparo 2, 3, 4, 5, 6, 7, 8 y 9.

4.1. Desarrollo del supervisor basado en invariantes de lugar

4.1.1. Restricciones del sistema

Para realizar el supervisor de este modelo se definieron las siguientes restricciones:

1. No puede estar el crudo circulando por el filtro y/o filtro llenándose y la válvula de eliminación de partículas cerrada al mismo tiempo.

$$u_5 + u_6 + u_8 \leq 1 \longrightarrow u_5 + u_6 + u_8 + u_{c2} \leq 1$$

2. No puede estar el crudo circulando por el filtro y/o filtro llenándose y la válvula de eliminación de partículas cerrada al mismo tiempo.

$$u_5 + u_6 + u_8 \leq 1 \longrightarrow u_5 + u_6 + u_8 + u_{c2} \leq 1$$

3. No puede estar filtro lleno y la válvula de eliminación de partículas abierta al mismo tiempo.

$$u_7 + u_9 \leq 1 \longrightarrow u_7 + u_9 + u_{c3} \leq 1$$

4. No puede estar filtro lleno y la válvula de del desairador abierta al mismo tiempo.

$$u_7 + u_{11} \leq 1 \longrightarrow u_7 + u_{11} + u_{c4} \leq 1$$

5. No puede estar la válvula manual abierta y la válvula de eliminación de partículas cerrada al mismo tiempo.

$$u_2 + u_8 \leq 1 \longrightarrow u_2 + u_8 + u_{c5} \leq 1$$

6. No puede estar la válvula manual abierta y la válvula del desairador cerrada al mismo tiempo.

$$u_2 + u_{10} \leq 1 \longrightarrow u_2 + u_{10} + u_{c6} \leq 1$$

Siguiendo la metodología para hallar el supervisor, se tiene que:

$$b = [1 \ 1 \ 1 \ 1 \ 1]^T$$

Y

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

Después de tener la matriz de incidencia del modelo calculada en Pipe v4, y guardarla en un archivo con extensión *.html*, podemos calcular en Matlab las ecuaciones del controlador:

$$D_c = -LD_p$$

$$b\mu_{c0} = b - L\mu_{p0}$$

4.1.2. Código utilizado en Matlab

A continuación se muestra el código utilizado en Matlab para las restricciones mencionadas:

```
*****
close all
clear all
clc

%Restricciones:

% Obtener matrices de PIPE:
[MIp,MIA,up0]=lectura('matriz_de_incidencia_modelo.html');

% Cálculo de la matriz de incidencia de la planta:
Dp=MIp-MIA;

% Descripción de la restricción:
L = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0;
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1;
     0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0;
     0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0;
     0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
```


7. No se puede apagar medidor de presión si la válvula manual está abierta

$$u_2 + q_{17} \leq 1$$

8. No se puede apagar sensor de temperatura si la válvula manual está abierta

$$u_2 + q_{21} \leq 1$$

9. No se puede apagar medidor de presión del filtro si la válvula manual está abierta

$$u_2 + q_{31} \leq 1$$

4.2.2. Desarrollo del supervisor con restricción de disparo 2

Se agrega al caso de estudio la restricción número 2:

$$u_1 + q_6 \leq 1$$

Al aplicar la transformación de la ecuación $\mu_i - q_j \leq 1$ se obtiene:

$$u_1 + u_{30} \leq 1$$

u_{30} es el lugar que se introduce, el cual almacenara el disparo de la transición t_6 , luego que el supervisor sea calculado, el supervisor volverá a ser transformado a su estado original.

La transformación hecha a la Red de Petri del filtro se muestra en la Figura 31.

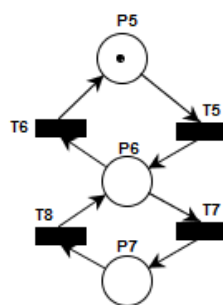


Figura 30. RDP original del filtro

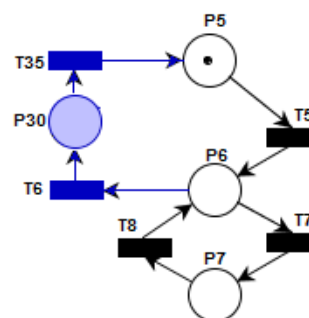


Figura 31. RDP transformada

2.2.2.1. Código utilizado en Matlab para la restricción de disparo 2

A continuación se muestra el código utilizado en Matlab para el cálculo del supervisor.

La transformación hecha a la Ted de Petri del filtro en la Figura 34.

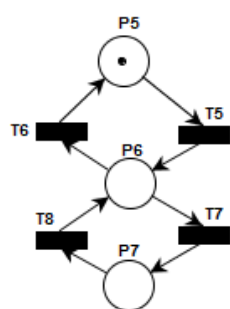


Figura 33. RDP original del filtro

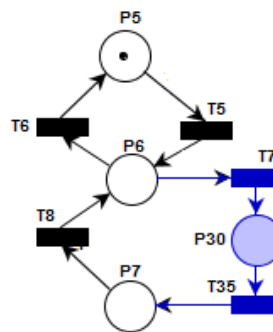


Figura 34. RDP transformada del filtro

4.2.3.1. Código utilizado en Matlab para la restricción de disparo 3

```

*****
close all
clear all
clc

%Disparo 3

% Obtener matrices de PIPE:
[MIp,M Ia,up0]=lectura('Disparo_3.html');

% Cálculo de la matriz de incidencia de la planta:
Dp=MIp-MIa;

% Descripción de la restricción:
L=[1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0];
b=[1];

% Cálculo del supervisor:
Dc=-L*Dp
uco=b-(L*up0)

*****

```

Ejecutando el archivo en Matlab se obtiene:

$$Dc = [1 0 0 0 0 0 0 0 0 0 0 -1 0 1 0 0 0 -1 0 0]$$

Con marcado inicial:

$$uc0 = [0]$$

El lugar de control descrito se muestra en la Figura 35.

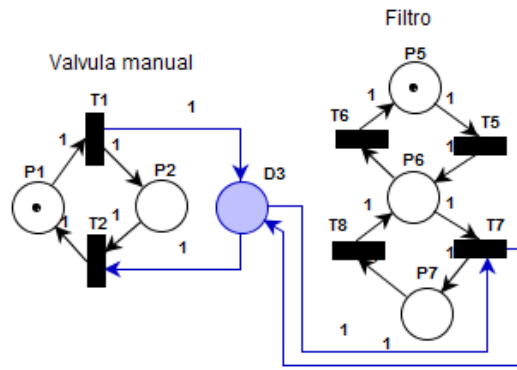


Figura 35. RDP supervisada de la restricción 3

4.2.4. Desarrollo del supervisor con restricción de disparo 4

Se agrega al caso de estudio la restricción número 4:

$$u_1 + q_8 \leq 1$$

Al aplicar la transformación de la ecuación $\mu_i - q_j \leq 1$ se obtiene:

$$u_1 + u_{30} \leq 1$$

La transformación hecha a la Red de Petri del filtro se muestra en la Figura 37.

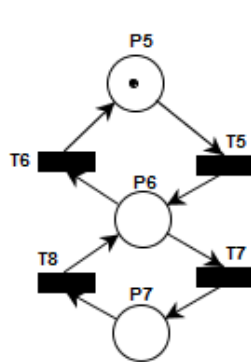


Figura 36. RDP original del filtro

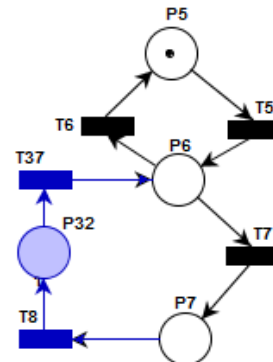


Figura 37. RDP transformada

4.2.4.1. Código utilizado en Matlab para la restricción de disparo 4

```

*****
close all
clear all
clc

%Disparo 4

% Obtener matrices de PIPE:
[MIp,MIA,up0]=lectura('Disparo_4.html');

% Cálculo de la matriz de incidencia de la planta:

```


4.2.6. Desarrollo del supervisor con restricción de disparo 6

Se agrega al caso de estudio la restricción número 6:

$$u_2 + q_{13} \leq 1$$

Al aplicar la transformación de la ecuación $\mu_i - q_j \leq 1$ se obtiene:

$$u_2 + u_{30} \leq 1$$

La transformación hecha a la Red de Petri del medidor de flujo se muestra en la figura 42.

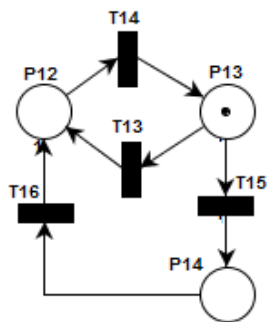


Figura 41. RDP original del medidor de flujo

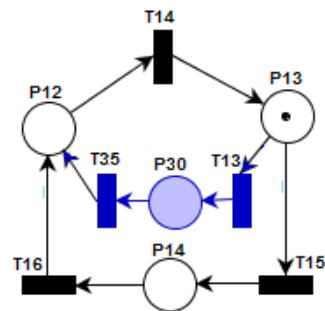


Figura 42. RDP transformada del medidor de flujo

4.2.6.1. Código utilizado en Matlab para la restricción de disparo 6

```
*****
close all
clear all
clc
%Disparo 6

% Obtener matrices de PIPE:
[MIp,MIa,up0]=lectura('Disparo_6.html');

% Cálculo de la matriz de incidencia de la planta:
Dp=MIp-MIa;

% Descripción de la restricción:
L=[0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
b=[1];

% Cálculo del supervisor:
Dc=-L*Dp
uco=b-(L*up0)
*****
```

Ejecutando el archivo en Matlab se obtiene:

$$Dc = [-1 0 0 0 -1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0]$$

Con marcado inicial:

$$uc0 = [1]$$

El lugar de control descrito se muestra en la figura 43

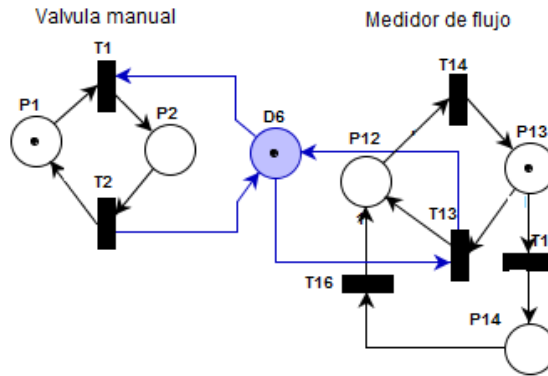


Figura 43. RDP supervisada de la restricción 6

4.2.7. Desarrollo del supervisor con restricción de disparo 7

Se agrega al caso de estudio la restricción número 7:

$$u_2 + q_{17} \leq 1$$

Al aplicar la transformación de la ecuación $\mu_i - q_j \leq 1$ se obtiene:

$$u_2 + u_{30} \leq 1$$

La transformación hecha a la Red de Petri del medidor de flujo se muestra en la figura 45.

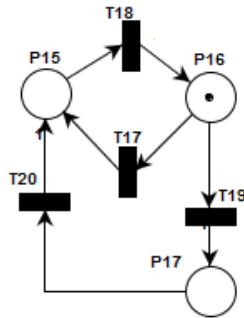


Figura 44. RDP original del sensor de presión

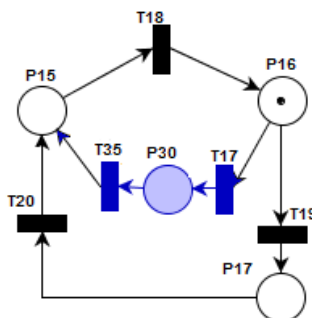


Figura 45. RDP transformada del sensor de presión

4.2.6.1. Código utilizado en Matlab para la restricción de disparo 7

```
*****
close all
clear all
clc

%Disparo 7
```



```

% Obtener matrices de PIPE:
[MIp,MIA,up0]=lectura('Disparo_7.html');

% Cálculo de la matriz de incidencia de la planta:
Dp=MIp-MIA;

% Descripción de la restricción:
L=[0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0];
b=[1];

% Cálculo del supervisor:
Dc=-L*Dp
uco=b-(L*up0)

*****

```

Ejecutando el archivo en Matlab se obtiene:

$$Dc = [-1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ -1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$$

Con marcado inicial:

$$uc0 = [1]$$

El lugar de control descrito se muestra en la Figura 46.

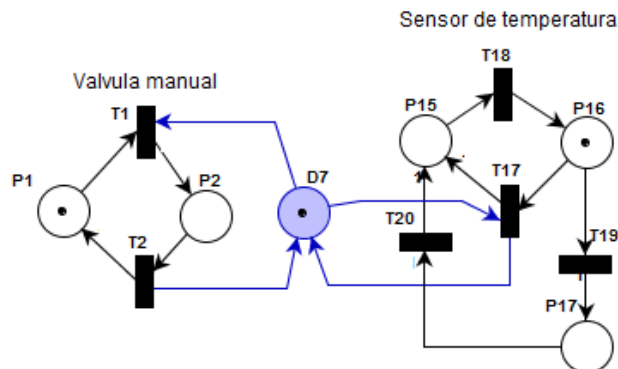


Figura 46. RDP supervisada de la restricción 7

4.2.7. Desarrollo del supervisor con restricción de disparo 8

Se agrega al caso de estudio la restricción numero 8:

$$u_2 + q_{21} \leq 1$$

Al aplicar la transformación de la ecuación $\mu_i - q_j \leq 1$ se obtiene:

$$u_2 + u_{30} \leq 1$$

La transformación hecha a la Red de Petri del medidor de flujo se muestra en la figura 48.

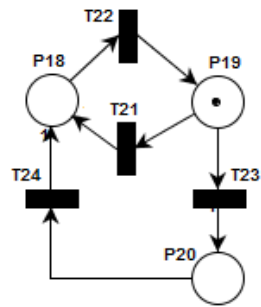


Figura 47. RDP original del sensor de temperatura

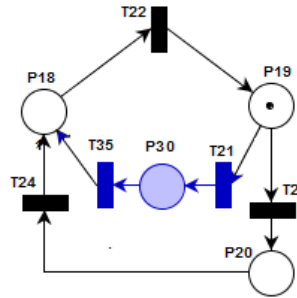


Figura 48. RDP transformada del sensor de temperatura

4.2.7.1. Código utilizado en Matlab para la restricción de disparo 8

```

*****
close all
clear all
clc

%Disparo 8

% Obtener matrices de PIPE:
[MIp,MIa,up0]=lectura('Disparo_8.html');

% Cálculo de la matriz de incidencia de la planta:
Dp=MIp-MIa;

% Descripción de la restricción:
L=[0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0];
b=[1];

% Cálculo del supervisor:
Dc=-L*Dp
uco=b-(L*up0)

*****

```

Ejecutando el archivo en Matlab se obtiene:

$$Dc = [-1 0 0 0 0 0 0 0 0 0 0 1 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0]$$

Con marcado inicial:

$$uco = [1]$$

El lugar de control descrito se muestra en la figura 49

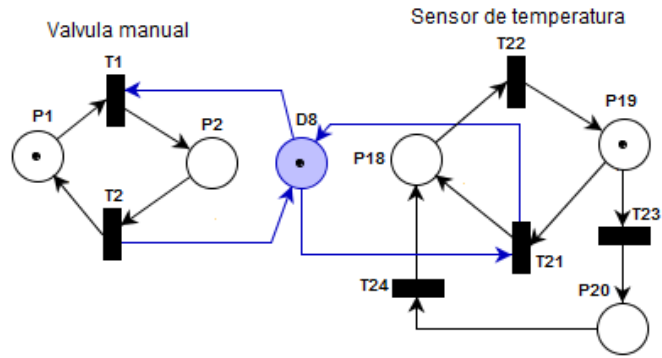


Figura 49. RDP supervisada de la restricción 8

4.2.8. Desarrollo del supervisor con restricción de disparo 9

Se agrega al caso de estudio la restricción numero 9:

$$u_2 + q_{31} \leq 1$$

Al aplicar la transformación de la ecuación $\mu_i - q_j \leq 1$ se obtiene:

$$u_2 + u_{30} \leq 1$$

La transformación hecha a la Red de Petri del medidor de flujo se muestra en la figura 51.

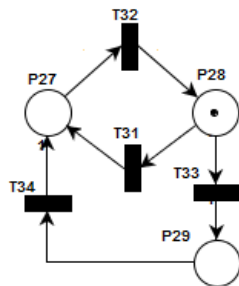


Figura 50. RDP original del sensor de presión del filtro

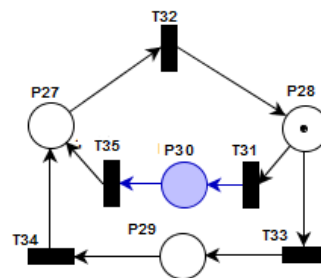


Figura 51. RDP transformada del sensor de presión del filtro

4.2.8.1. Código utilizado en Matlab para la restricción de disparo 9

```

*****
close all
clear all
clc

%Disparo 9

% Obtener matrices de PIPE:
[MIp,MIA,up0]=lectura('Disparo_9.html');

% Cálculo de la matriz de incidencia de la planta:
Dp=MIp-MIA;
% Descripción de la restricción:

```

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

```
% Cálculo del supervisor:
```

```
Dc=-L*Dp
```

```
uco=b-(L*up0)
```

```
*****
```

Ejecutando el archivo en Matlab se obtiene:

```
Dc = [-1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 1 0 0 0 0 0]
```

Con marcado inicial:

```
uc0 = [1]
```

El lugar de control descrito se muestra en la figura 52

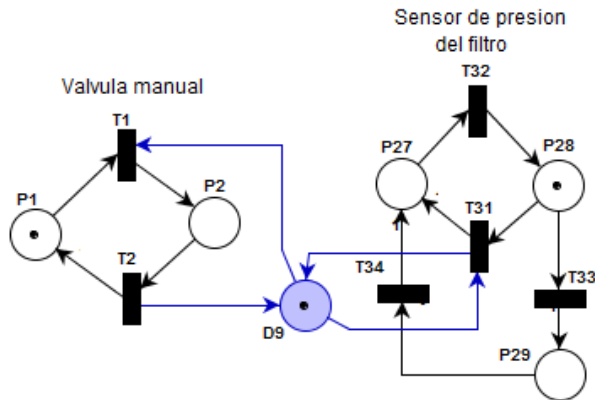


Figura 52. RDP supervisada de la restricción 8

5. ANEXO E. CODIGOS DE LAS FUNCIONES EN MATLAB

5.1. Funciones tomadas

a. Lectura.m

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

- Mlpost → matriz de incidencia posterior
- Mlant → matriz de incidencia anterior
- MI → marcado inicial

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

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

for i = 1:numlugares
    posval(i,:) = strfind(texto(pos(i):pos(i+1)),'<tdclass="cell">')+pos(i)+15;
end
s = size(posval);
numtransiciones = s(1,2);
Mlpost = zeros(numlugares,numtransiciones);
for i = 1:numlugares
    for j = 1:numtransiciones
        Mlpost(i,j) = str2double(texto(posval(i,j)));
    end
end
pos = strfind(texto(posincant:posmarc),'<tdclass="cell">')+posincant+15;
Mlant = zeros(numlugares,numtransiciones);
cont=2;
for i = 1:numlugares
    for j = 1:numtransiciones
        Mlant(i,j) = str2double(texto(pos(cont)));
```

```

        cont = cont+1;
    end
end
pos = strfind(texto(posmarc:poscurr),'<tdclass="cell">')+posmarc+15;
MI = zeros(numlugares,1);
cont=1;
for i = 1:numlugares
    MI(i,1) = str2double(texto(pos(cont)));
    cont = cont+1;
end
fclose(fid);

```

b. Leer_col.m

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

```

function [A]=leer_col(nombre)

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

```

6. ANEXO F. GENERACION DEL LADDER

A continuación se hace el desarrollo para la generación del ladder del modelo en Redes de Petri.

Los requerimientos para el CRP siguientes:

- Sistema operativo: Windows7/Vista/WindowsXP/Windows2003 Server/Windows 2000/ Windows 98, otras plataformas Windows pueden funcionar, más no están verificadas.
- Utiliza en promedio 11MB de RAM, bajo uso típico (debe sumarse los requerimientos del SO, este valor puede subir para redes muy grandes)
- 2.5MB de espacio en disco
- Según la versión de Windows y el tipo de procesador, instalar Java con el JRE.

6.1. Ejecutar

Creamos una carpeta cuyo nombre en nuestro caso se llamara “*supervisorio*”. En esta carpeta guardaremos la aplicación del CRP y el archivo de Micrologix 500.

En la carpeta llamada “*LADDER_PETRI*”, abrir la carpeta llamada “*CRP full*”, dar clic en *CPR*. Se podrá visualizar la Figura 52.

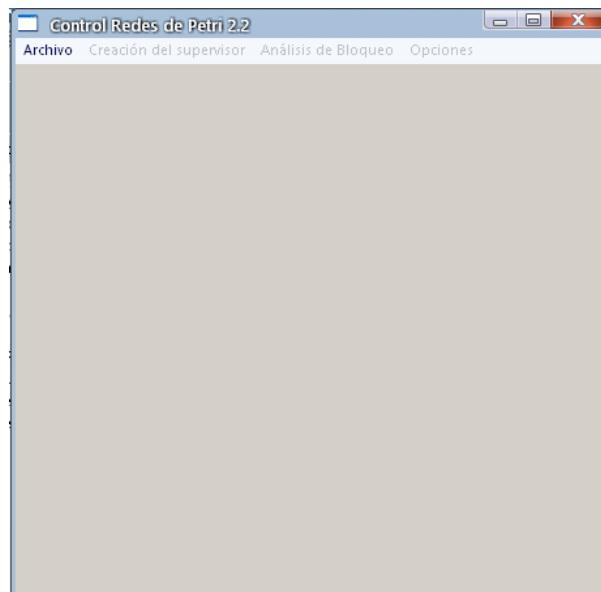


Figura 52. Interfaz inicio del CRP

6.2. Adición de la RDP

Dar clic *archivo*, elegir la opción *nuevo supervisor* (Figura 53)

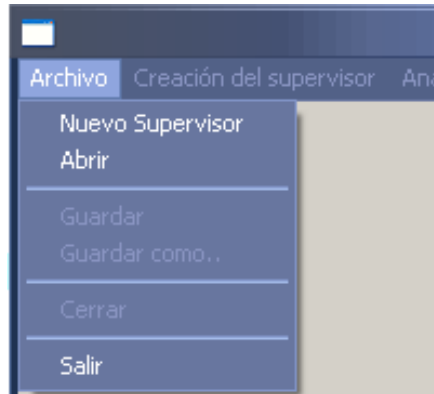


Figura 53. Menú archivo

Automáticamente se abre la plataforma de PIPE (ver figura 54), la ventana del CRP cambia su interfaz como se muestra en la Figura 55. Para agregar una RDP ver anexo A.

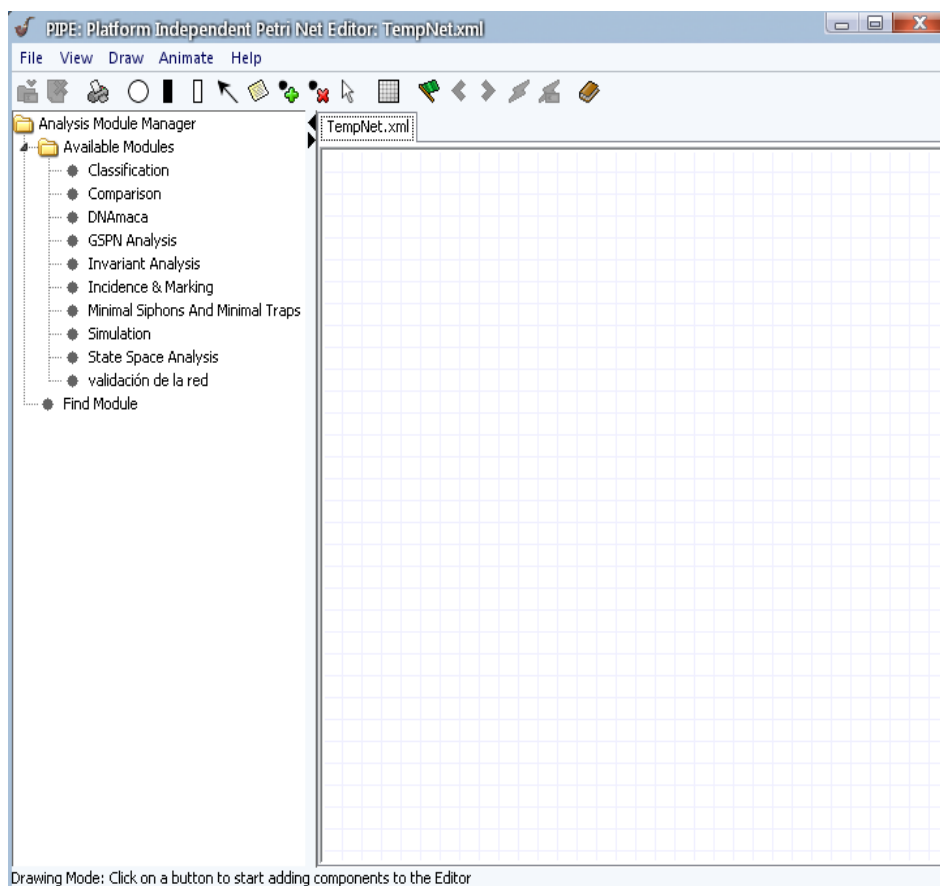


Figura 54. Interfaz PIPE

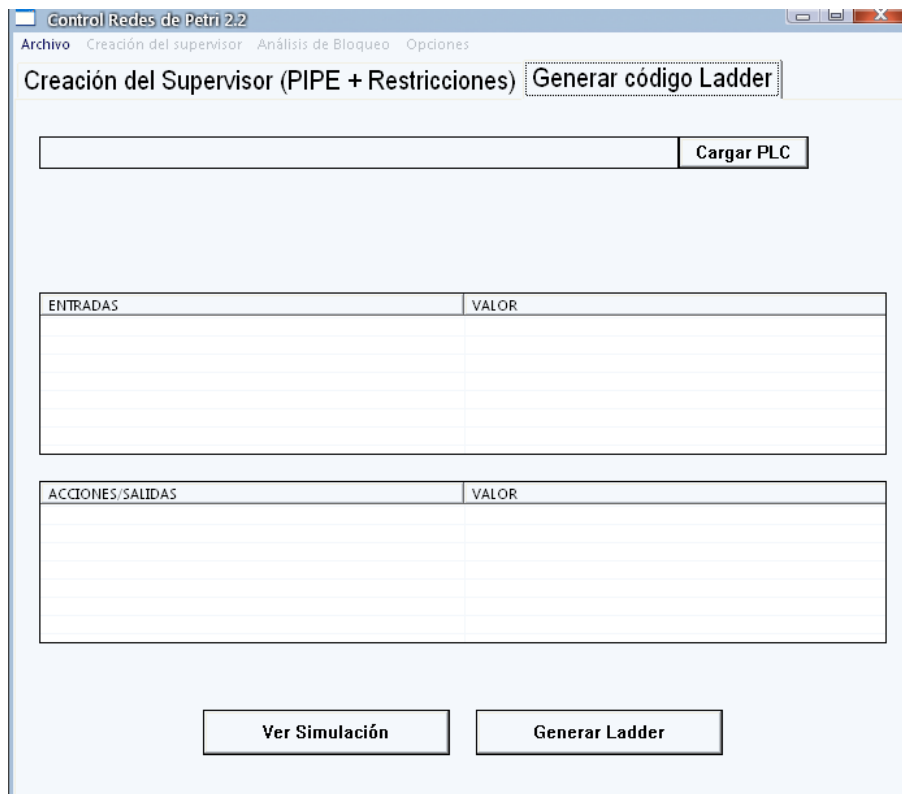
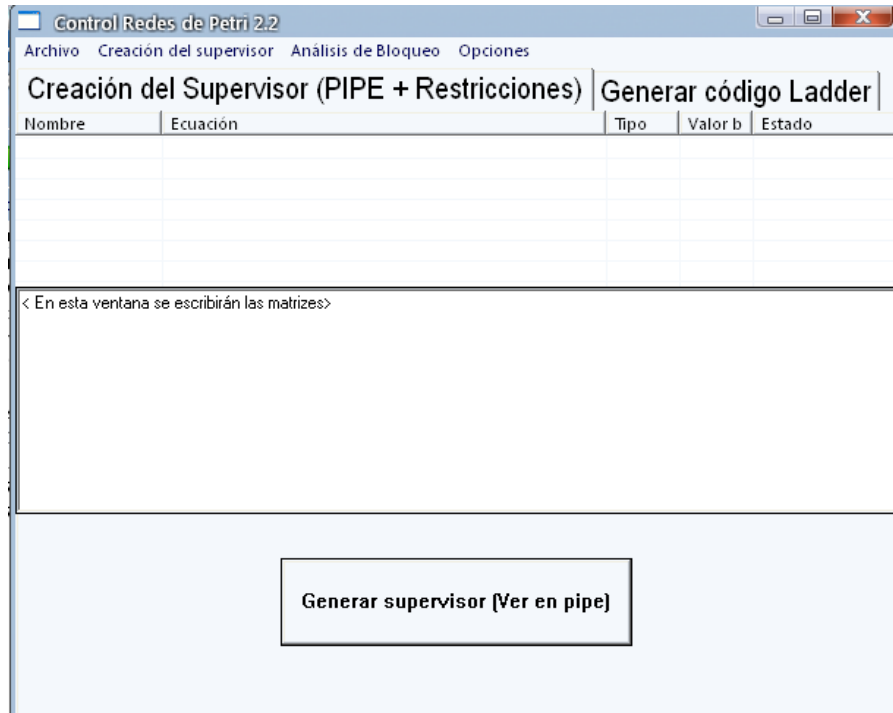


Figura 55. Interfaz del CRP

6.3. Adición de restricciones

Después de crear la RDP en PIPE, diríjase a la ventana del CRP para proceder a adicionar las restricciones del supervisor. Para ello dar clic en *Creación del supervisor* (ver figura 56), elegir la opción *Adicionar condición* y se mostrara la ventana de la Figura 57.

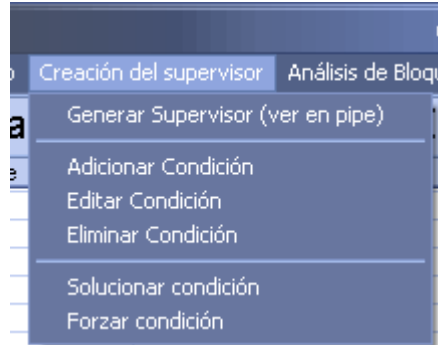


Figura 56. Menú creación supervisor

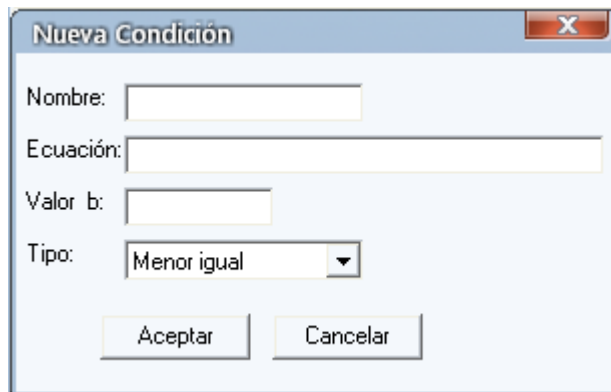


Figura 57. Ventana para la adición de una restricción

En la figura 57 se debe adicionar el nombre de la restricción, la ecuación de la restricción y el valor de b.

Para la adición de la ecuación de la restricción se usa el carácter u para referenciar el vector de marcado, el carácter q para referenciar el vector de disparo. Se da un ejemplo de la restricción que se desea adicionar en nuestro proyecto, en este caso se desea adicionar la siguiente restricción de invariante de lugar:

$$u_5 + u_6 + u_8 \leq 1$$

La anterior restricción en la ventana de *nueva condición* (Figura 57) queda de la siguiente forma (Ver figura 58):

$$uP_5 + uP_6 + uP_8 \leq 1$$

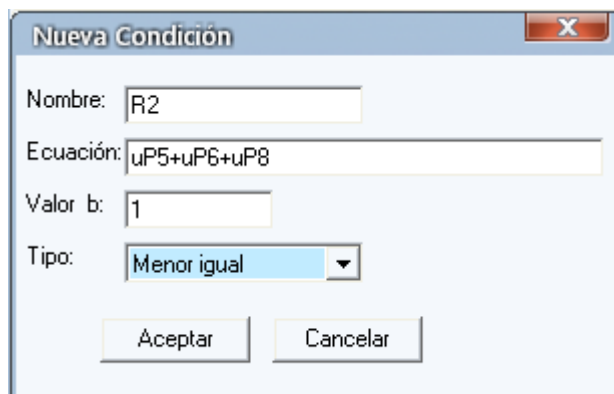


Figura 58. Ejemplo de la adición de la restricción de invariante de lugar

También se da un ejemplo de la restricción con vector de disparo, se adiciona la siguiente restricción:

$$u_1 + q_5 \leq 1$$

La anterior restricción disparo en la ventana de *nueva condición* (Figura 57) queda de la siguiente forma (Ver Figura 59):

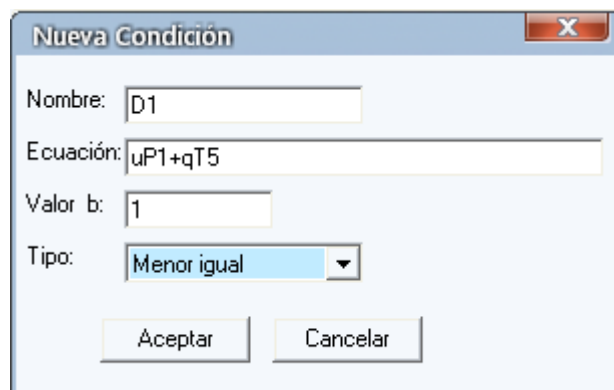


Figura 59. Ejemplo de la adición de la restricción de vector de disparo.

Después de presionar “*aceptar*” en la ventana de *nueva condición* se agrega la restricción al CRP. Si la restricción se guardo correctamente, es escrita en la lista superior del CRP, como se muestra en la Figura 60.

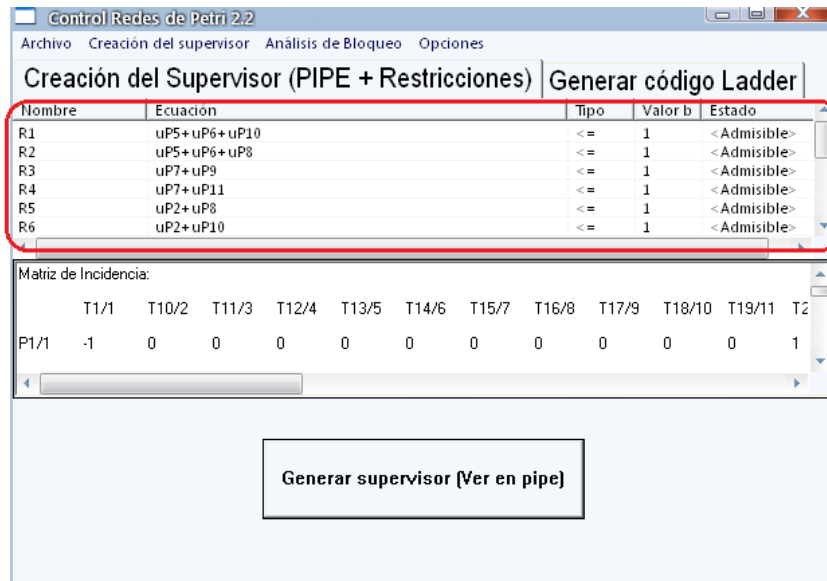


Figura 60. Lista de restricciones.

Una vez se han agregado todas las restricciones y todas estas sean admisibles, es posible implementar el supervisor. Esto se realiza de la siguiente forma:

Clic en el comando *Generar supervisor* del menú *Creación del supervisor*. Como se muestra en la figura 61.

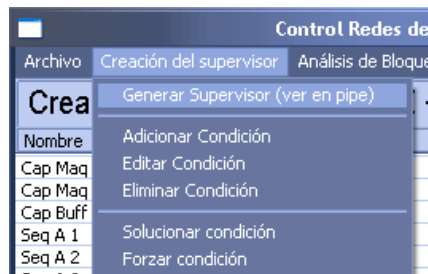


Figura 61. Generar supervisor

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.
- **No Inicial:** 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.

6.4. Guardar trabajo

Usando el menú *Archivo* del CRP dar clic en guardar (ver Figura 62), este archivo lo guardamos en la carpeta llamada “*supervisorio*” anteriormente creada.

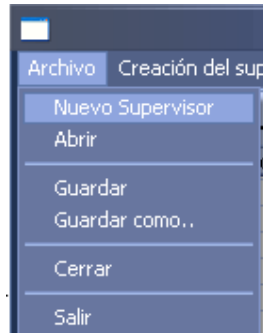


Figura 62. Guardar trabajo

Después de haber adicionado todas las restricciones del supervisor en CRP y de haber guardado el trabajo, procedemos a hacer lo siguiente:

6.5. Generar un archivo PLC

El archivo PLC sirve para indicar al CRP la configuración de un dispositivo y poder generar el código ladder para este. Para generar una especificación en un archivo PLC se utiliza la herramienta “PLC Detector” encontrada en la carpeta “*LADDER_PETRI*”

Abrimos el RsLogix 500, adicionamos al menos una línea de código cualquiera, esta será usada como referencia por el “PLC Detector” como se muestra en la Figura 63.

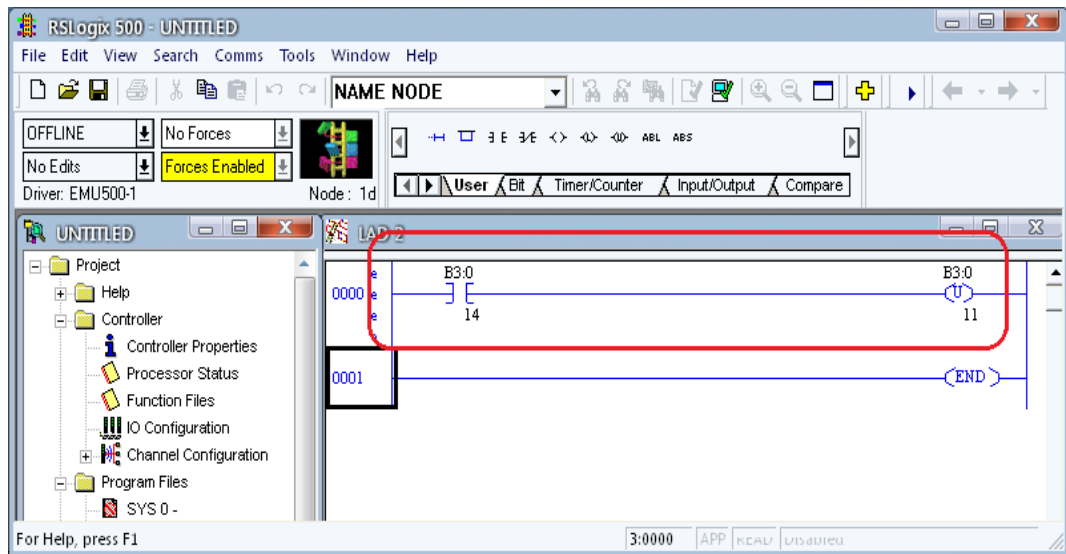


Figura 63. Adición de una línea de código cualquiera

6.6. Configuración del RsLogix 500

Para hacer la configuración del RsLogix 500 dar clic en *Controller Propiedades* como se muestra en la Figura 64.

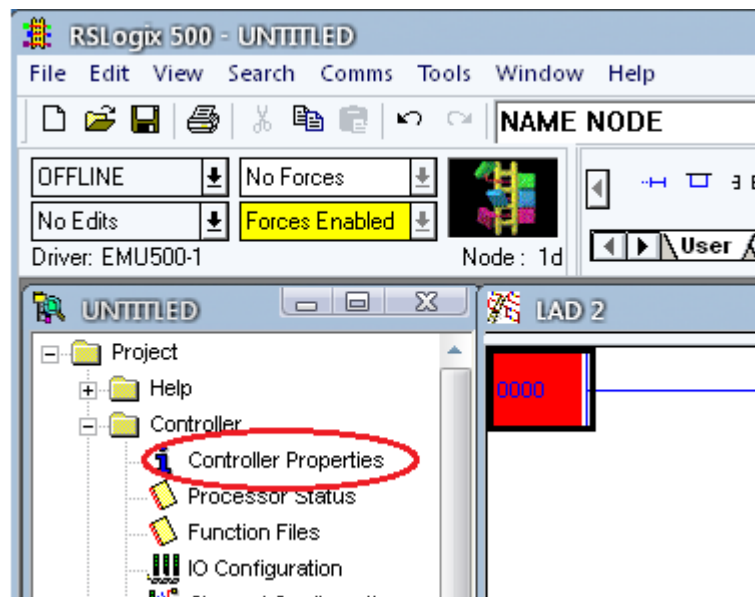


Figura 64. Configuración del RsLogix 500

Se despliega un ventana como se muestra en la Figura 65. En *Processor Type* elegir el Micrologix 1500 LSP Serie A. Clic en *aplicar*.

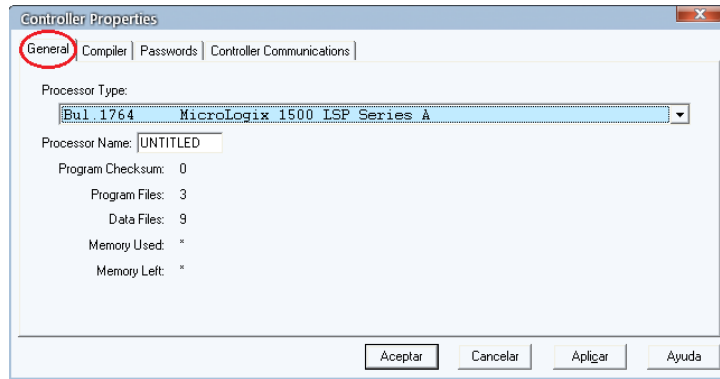


Figura 65. Ventana de Controller properties

Después de haber configurado el ladder damos clic en *guardar*. Guardamos en archivo en la carpeta “supervisorio”

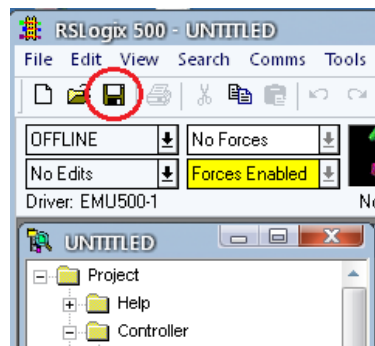


Figura 66. Guardar línea de código

El proyecto creado se debe guardar en formato *.SLC* como se indica en la Figura 67.

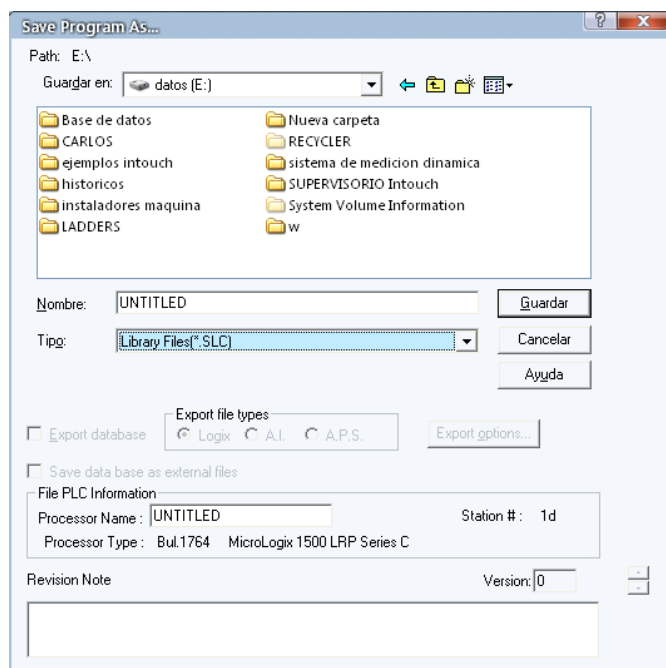


Figura 57. Guardar línea de código con extensión *.SLC*

Después de crear el archivo *.SLC*, dar clic en “PLC Detector” que se encuentra en la carpeta “*LADDER_PETRI*”. Posteriormente se abre la ventana de la Figura 68.

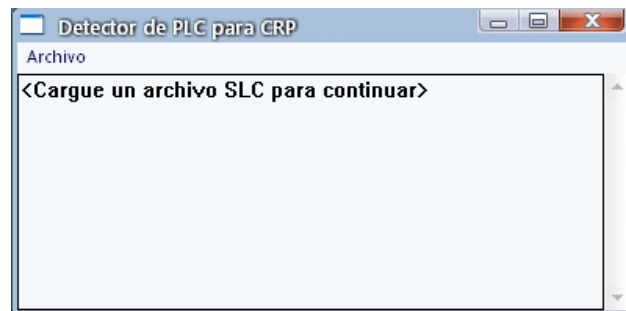


Figura 68. Detector de PLC para el CRP

Clic en *Archivo*, elegir la opción *Cargar* (Figura 69), posteriormente buscamos el archivo *.SLC* creado anteriormente y lo cargamos como se muestra en la figura 57.

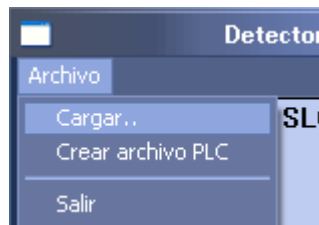


Figura 69. Opción cargar archivo *.SLC*

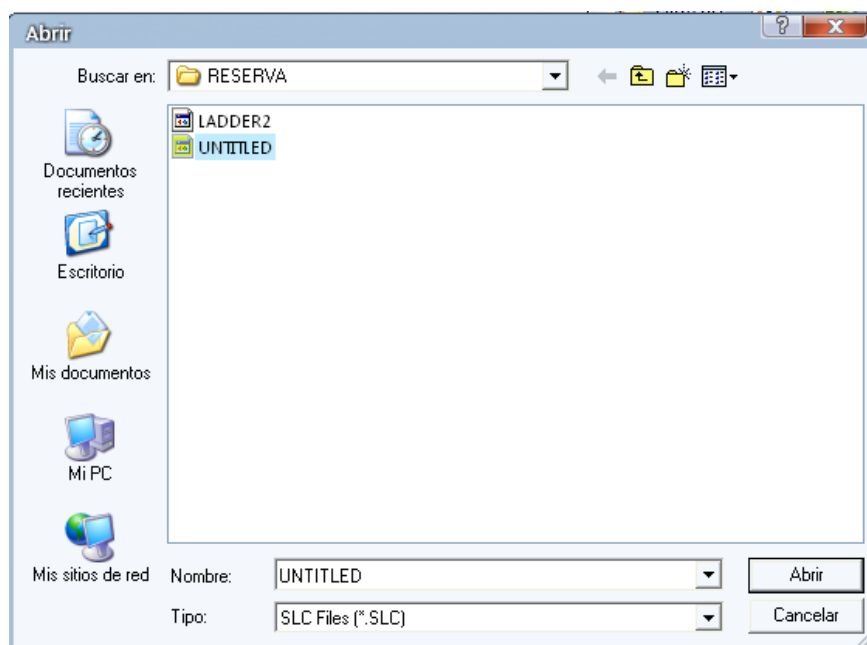


Figura 70. Cargar archivo *.SLC*

Si “PLC Detector” puede determinar la configuración del PLC se debe mostrar la siguiente ventana (Figura 71)

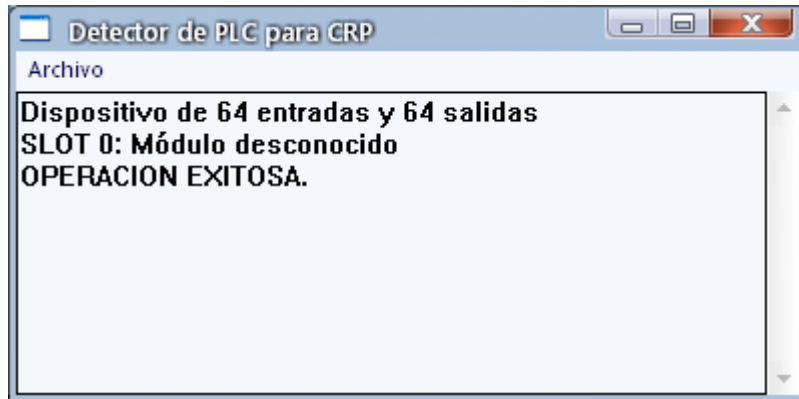


Figura 71. Detección exitosa del dispositivo

Si la detección es exitosa, se procede a crear el archivo PLC:

Clic en archivo, seleccionar la opción *Crear archivo PLC* como se muestra en la Figura 71.

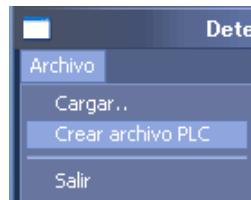


Figura 71. Opción crea crear archivo PLC

Automáticamente se muestra una ventana que permite ubicar donde se guardará el archivo PLC. Lo guardamos con el nombre "micrologix 1500" en la carpeta "supervisorio"

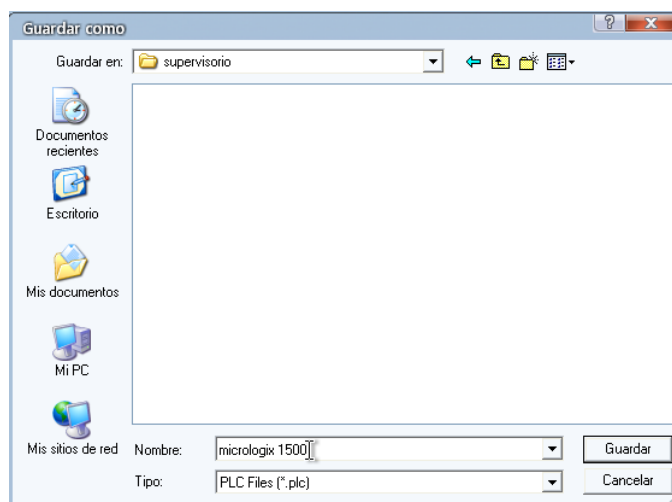


Figura 72. Carga del archivo PLC

Una vez creado, ya es posible utilizar el archivo PLC, el cual contiene la información necesaria para generar ladder utilizando CRP.

6.7. Generar código ladder para un PLC micrologix 1500

A continuación se generará el ladder para el controlador Micrologix 1500 serie A, del cual ya se creó el archivo *PLC* correspondiente.

En la interfaz del CRP, clic en la parte superior en *Generar código ladder*, como se muestra en la Figura 73.

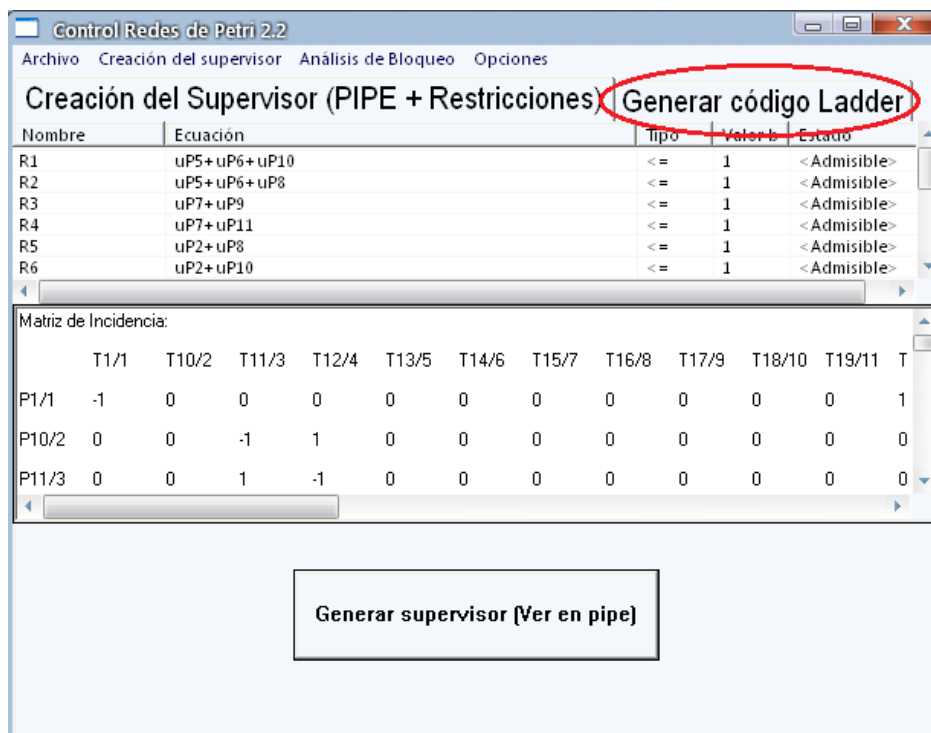


Figura 73. Icono generar código

El archivo *.PLC* puede ser cargado mediante el botón *Cargar PLC* en la interfaz de la Figura 74.

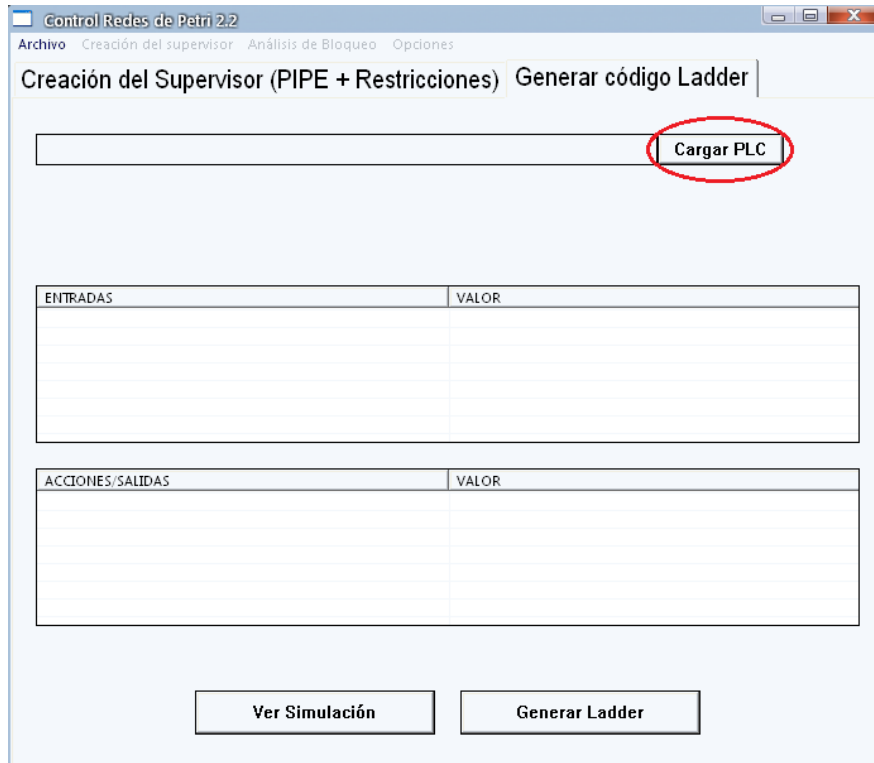


Figura 74. Comando de cargar PLC

Seguidamente se muestra una ventana que permite seleccionar el archivo *.PLC* que se va a cargar, que tiene por nombre “micrologix 1500” y se encuentra ubicado en la carpeta “supervisorio”. Ver Figura 75.

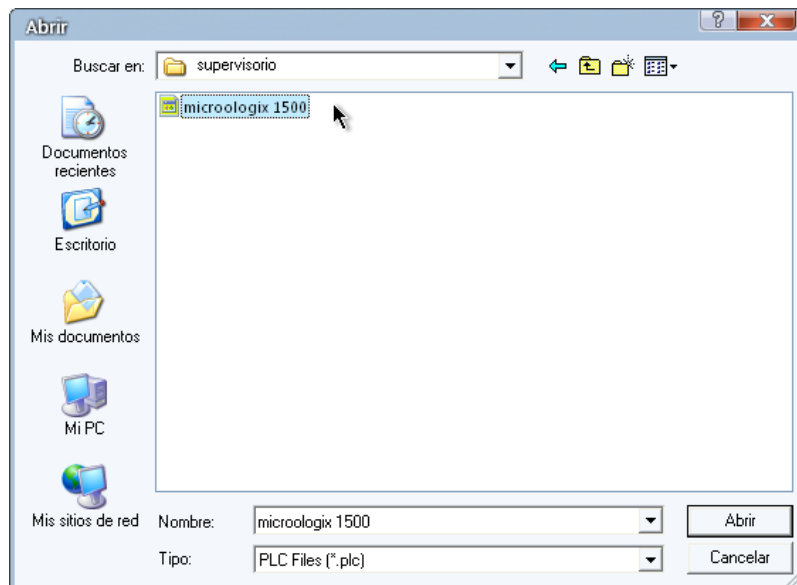


Figura 76. Carga del archivo PLC

Si este último procedimiento se hizo de la forma correcta, aparecerá una etiqueta que dice “PLC CARGADO” en la parte superior de la interfaz, como se aprecia en la Figura 77.

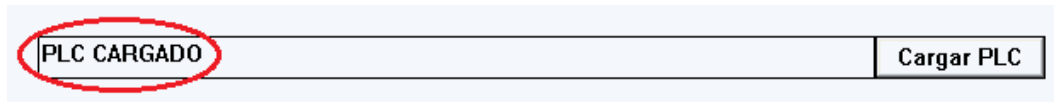


Figura 77. Archivo PLC cargado por el CRP

6.8. Definición de I/O en PIPE

6.8.1. Definir entradas

Para asignar una entrada a una transición seleccionamos la transición, damos clic derecho a dicha transición y escogemos la opción *Editar entrada*.

Una vez seleccionada saldrá un cuadro de dialogo como se muestra en la Figura 78, en el espacio asignamos el nombre de la entrada, para más facilidad en nuestro proyecto a cada entrada se le asigno el nombre de cada transición.

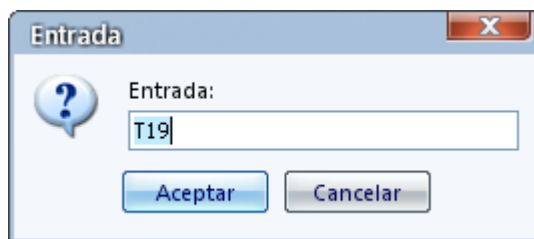


Figura 78. Edición de la entrada

6.8.2. Definir salidas

Para asignar una salida a un lugar seleccionamos el lugar, damos clic derecho a dicho lugar y escogemos la opción *Editar acción*.

Una vez seleccionada saldrá un cuadro como se muestra en la Figura 79, en el espacio asignamos el nombre de la salida, en nuestro proyecto para más facilidad a cada salida se le asigno el nombre de cada lugar.

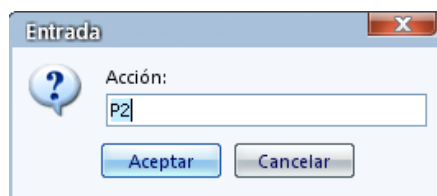


Figura 79. Edición de la acción

Una vez asignadas todas las entradas y salidas, estas se pueden ver en la interfaz del CRP en la pestaña *Generar código ladder*.

6.9. Asignación de entradas y salidas del PLC

Debemos asignar las entradas y salidas reales del PLC, para esto se hace clic sobre cada entrada o salida en el cuadro de dialogo del CRP. 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 y el usuario debe remplazar este valor en RsLogix 500. Ver Figura 80.

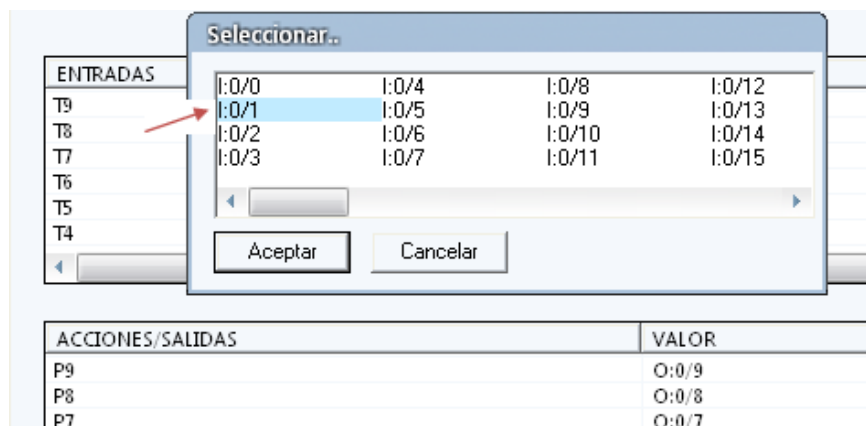


Figura 80. Asignación de I/O del PLC

Por último para generar el ladder, presionar el botón *Generar ladder* en la parte inferior del CRP, se desplegará una ventana donde se especifica la ubicación del ladder (archivo *.SLC*) creado. El archivo *.SCL* contiene el código ladder y se puede abrir en RsLogix 500.



Figura 81. Comando para generar el ladder.

7. ANEXO G. MANUAL DE USUARIO DEL SOFTWARE INTOUCH 10.1

El programa Intouch Wonderware es un software de visualización que permite crear aplicaciones de interfaz entre hombre maquina (HMI) para la automatización industrial.

Los requerimientos del sistema son los siguientes:

- Mínimo 500 Mb de disco duro
- Mínimo 64 MB de RAM
- Microsoft Windows W95/98 SE o NT.

Para que un HMI sea una herramienta de control supervisorio se requiere: Adquirir datos, visualizar y supervisar la secuencia del proceso, todo esto se realizará desde el monitor de un computador. Para el desarrollo de este proyecto se utilizó:

- Inouch 10.1. de Wonderware: Creación de la interfaz Hombre-Máquina (HMI).
- Wonderware I/O server: Comunicación entre el PLC e Intouch.
- Rslogix 500: Programación del PLC

7.2. Intouch para el diseño de un HMI

El programa Intouch, es un software de visualización que permite crear aplicaciones HMI (Interfaz Hombre-Máquina) para la automatización industrial, control de procesos y monitoreo. Intouch es el primer paquete SCADA que utiliza el sistema operativo Windows como plataforma. Aparece en 1989 cuando los sistemas de monitorización utilizan DOS como sistema operativo [32].

Al funcionar sobre Windows, aprovecha las capacidades gráficas de este sistema operativo: los procesos son más fáciles de documentar, el entorno gráfico es ideal para la representación de esquemas y valores y las aplicaciones son más flexibles y fáciles de implementar e interpretar.

El software se desarrolla bajo el sistema operativo Windows 95/98/NT/2000, el cual, da el acceso de usuarios a un paquete completo de las herramientas de automatización, entre las cuales, se pueden mencionar las siguientes:

- Base de datos de la fábrica (Servidor industrial SQL).
- Supervisión de producción.
- Gerencia flexible de la jornada (In Batch).
- Visualización en Internet (Web Server de Factory Suite).

- Conectividad I/O Servers.

Intouch permite crear aplicaciones con características completas, estas incluyen el intercambio dinámico de datos DDE, enlace de objetos e incrustaciones (OLE), gráficos y más. Al paquete se lo puede ampliar agregando asistentes personalizados, objetos genéricos y extensiones de script. Intouch tiene grandes aplicaciones en el área industrial y abarca una multitud de mercados tales como el alimenticio, automotor, farmacéutico, petrolero, pulpa y papel entre otros [32].

7.3. Intouch 10.1

Intouch 10.1, proporciona una sola visión integrada de todos sus recursos de control e información, consta básicamente de dos elementos: Windowmaker y Windowviewer.

Windowmaker es el sistema de desarrollo de las aplicaciones, el que permite utilizar todas las herramientas para la creación de ventanas animadas e interactivas conectadas a otros sistemas de entrada/salida externos u otras aplicaciones de MS Windows. Windowviewer es el sistema Runtime utilizado para poder ejecutar las aplicaciones desarrolladas con el sistema Windowmaker [33]. Se muestran los íconos correspondientes al Windowmaker y Windowviewer, respectivamente.

Este es el icono de Windowmaker o creador de aplicaciones:



Este es el icono de Windowviewer o Runtime:



7.3.1. Tipo de ventanas:

Intouch trabaja con ventanas o pantallas. Antes de empezar a dibujar, es necesario definir la ventana sobre la que se va a trabajar. Éstas pueden ser de tres tipos:

- Replace: Cierra cualquier otra ventana cuando aparece en pantalla.
- Overlay: Aparece sobre la ventana que está siendo desplegada.
- Popup: Esta ventana siempre queda por encima de las demás[33].

7.3.2. Creación de variables

Las variables y sus nombres son creadas usando el *diccionario de variables*. Para ello, se ingresa en el menú *Special/TagNameDictionary*,

entonces se despliega una ventana en la que se tienen varias opciones de las que se puede valer para crear diferentes tipos de variables, ya sean locales o relacionadas con una fuente de datos externa como por ejemplo un PLC [33].

7.3.3. Scripts

Son herramientas de programación de Intouch, que permiten ejecutar comandos y operaciones lógicas cuando se cumplen condiciones iniciales especificadas. Se tienen básicamente tres tipos de scripts:

- Window Scripts: Funcionan dentro de una ventana específica.
- Condition Scripts: Están relacionados con una variable discreta o con una expresión que cumpla la condición *verdadero* o *falso*.
- Data change Script: Están relacionados solamente con una variable. Son ejecutados una vez que la variable cambia por un valor mayor que el valor de banda muerta definido en el *diccionario de variables*.

7.3.4. Animation Links

Gracias a estos *enlaces de animación* se puede crear un gráfico y éste puede ser animado mediante las *Animation Links*. Las *Animation Links* provocan que el objeto cambie de apariencia reflejando cambios en los valores de la base de datos [33].

7.3.5. Direccionamiento de I/O en intouch

Intouch identifica un elemento de datos en un programa Servidor de I/O, usando una nomenclatura de tres partes en la que se incluye *applicationname*, *topicname* e *ítem name*. Para obtener datos de otra aplicación, el programa cliente (Intouch) abre un canal al programa servidor especificando los tres ítems antes mencionados.

Para que Intouch pueda adquirir datos de otra aplicación, debe conocer el tipo de dato, sea éste discreto, entero, real (punto flotante), o mensaje. Esta información determina el tipo de I/O para el *tagname* cuando éste es definido en la base de datos de Intouch. Ver configuración del direccionamiento de I/O en Intouch en el anexo G.

7.3.6. Wonderware Intouch Server

Wonderware I/O Servers son aplicaciones de Microsoft Windows, mediante las cuales, otras aplicaciones de Windows como Intouch, Excel, etc., pueden acceder a datos de un mundo real que están almacenados en un PLC, RTU, etc [33].

7.4. Características de Intouch

El paquete Intouch incluye las siguientes características:

- **Sistemas de alarmas distribuidas:** El nuevo sistema distribuido, soporta múltiples servidores de alarmas de forma concurrente, proporcionando a los operadores la capacidad de visualizar y reconocer simultáneamente información de alarmas desde múltiples ubicaciones remotas.
- **Historial distribuido:** El sistema de tendencias históricas distribuidas, le permite especificar de forma dinámica una fuente de datos de archivos históricos diferente para cada pluma de un gráfico.
- **Conversión de resolución dinámica:** Puede desarrollar aplicaciones con una resolución de pantalla y ejecutarlas en otras, sin afectar a la aplicación original. Las aplicaciones se ejecutan con una resolución definida por el usuario, en lugar de la resolución de visualización.
- **Direccionamiento de referencias dinámicas:** Las referencias de fuente de datos se pueden modificar de forma dinámica para direccionar múltiples fuentes de datos con una única etiqueta.
- **Desarrollo de la aplicación en red:** Las nuevas características de desarrollo remoto, dan cabida a grandes instalaciones multinodo, incluyendo la actualización de todos los nodos de una red, desde una única estación de desarrollo.
- **FactoryFocus:** FactoryFocus es una versión sólo de visualización de la ejecución de Intouch 10.1. Permite a los administradores y supervisores visualizar un proceso continuo de aplicación [32].

Otras características y ventajas que incluye Intouch son:

- Solución de visualización de procesos de bajo costo a un precio significativamente menos que el de un HMI completo.
- Visualización de procesos de aplicación en tiempo real.
- Fácil conexión con Wonderware NetDDE [32].

7.5. Entorno Intouch 10.1.

En la Figura 82, se puede observar, el entorno básico de trabajo del WindowMaker del software Intouch 10.1.

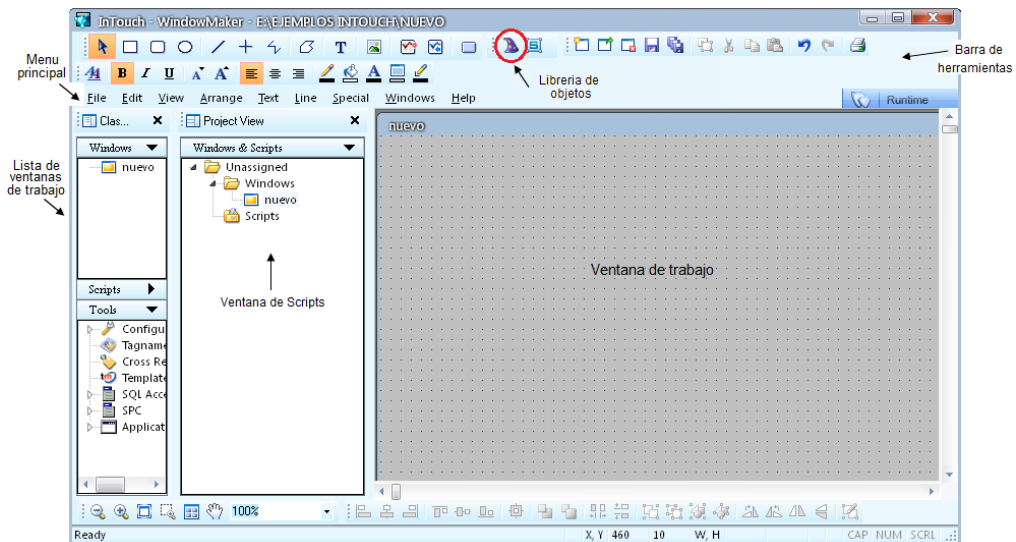


Figura 82. Entorno de trabajo de Intouch 10.1.

A continuación se describen cada uno de los elementos del WindowMaker.

7.5.1. Elementos de WindowMaker

File: Manejo de ficheros y ventanas. Contiene los siguientes elementos:

Tabla 2. Opciones menú *File*

New	Crear una nueva ventana
Open Window	Abrir ventana existente
SaveWindow	Guardar una ventana
CloseWindow	Cerrar ventana
DeleteWindow	Borrar una ventana
SaveWindow as	Guardar una ventana con un nombre distinto
SaveallWindow	Guardar todas las ventanas abiertas
WindowViewer	Saltar o ejecutar el programa WindowViewer
Print	Permite imprimir
ExportWindow	Exportar ventana a otra aplicación Intouch
Import	Importar ventana de otra aplicación Intouch

Edit: Contiene una serie de comandos para editar los objetos de la ventana. Con estas funciones, se puede editar los objetos que se encuentren seleccionados.

Tabla 3. Opciones menú *Edit*

Undo	Permite deshacer la última acción de edición
------	--

Nothingto redo	Rehace la última acción deshecha
Duplicate	Duplica el/los objeto/s seleccionados
Cut	Cortar al portapapeles
Copy	Copiar al portapapeles
Paste	Pegar la portapapeles
Erase	Borrar
Selectall	Seleccionar todo
Reduce radius	Reducir el radio de una curva del objeto seleccionado
Addpoint	Añadir punto
Del point	Borrar punto
Symbol Factory	Llamada al programa Symbol Factory

View: Con el menú *VIEW*, se define qué utilidades o elementos de *WindowMaker* se quieren tener visibles mientras se programa.

Tabla 4. Opciones menú *View*

Arrange	Este menú contiene comandos que permiten "arreglar" los objetos seleccionados. Se puede por ejemplo, alinear objetos, rotar o superponer	
	Make Symbol	Este comando permite combinar varios objetos seleccionados y convertirlos en un sólo objeto llamado "símbolo". Un símbolo puede estar compuesto por varios símbolos y/o múltiples objetos.
	Break Symbol	Permite dividir un símbolo previamente creado en los objetos originales
	Makecell	Este comando permite combinar objetos individuales y/o símbolos en una única unidad, llamada "celda". A una celda no se le pueden asignar <i>animation links</i>
	Break cell	Permite dividir una celda previamente creada en los objetos originales
Text	Permite modificar la fuente del objeto texto seleccionado, así como ponerlo en negrita, subrayado, itálica, justificarlo y modificar su tamaño	
Line	Permite modificar la forma de la línea del objeto seleccionado	
Special	Contiene menús muy importantes de WM, que son explicados posteriormente	
Windows	Permite moverse rápidamente de una a otra ventana activa	
Help	Permite acceder a una ayuda sensitiva en cualquier momento de la programación	
RunTime	Permite un acceso directo y rápido a la emulación de la aplicación	

7.1. Configuración de un nuevo proyecto

Para crear un nuevo proyecto es necesario entrar en intouch desde Windows, clic dos veces en el icono de Intouch.



Figura 83. Icono de intouch 10.1

En la pantalla aparecerá la ventana de *Application Manager* (Figura 84). Desde esta ventana podemos abrir proyectos previamente creados, o crear un proyecto nuevo.

Para crear un proyecto nuevo, seleccionar *File; New* en el menú principal.

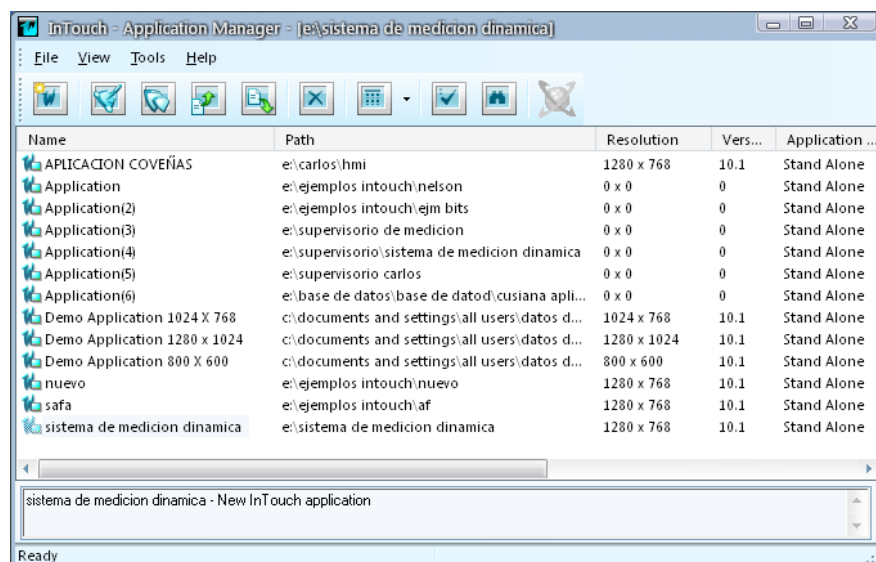


Figura 84. Ventana del Application Manager

Inmediatamente aparecerá una ventana (ver Figura 85) donde se debe definir la ruta donde se guardará el proyecto. Si se desea cambiar el directorio del proyecto seleccionar *Browse*, de otra manera el sistema elige el directorio por defecto. Presionar *Siguiente*.

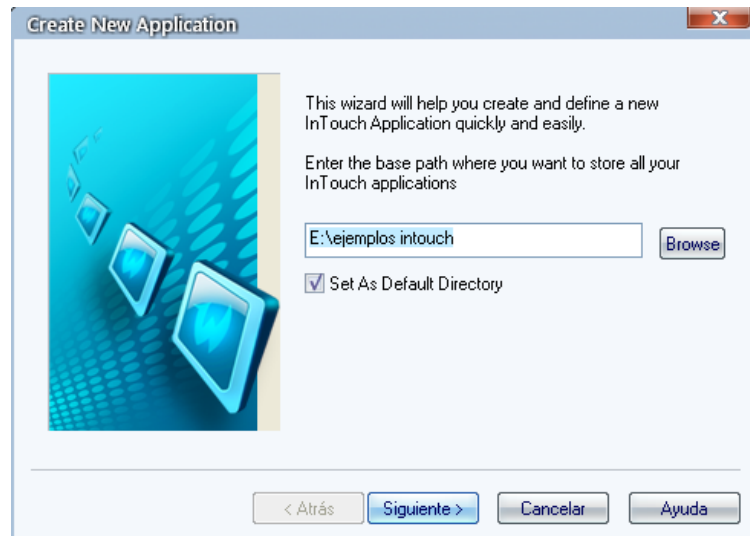


Figura 86. Definir directorio

Introducir el nombre de la carpeta donde se creara el proyecto, para nuestro proyecto la llamaremos “Sistema de medición dinámica” (Figura 87). Posteriormente clic en *Siguiente*.

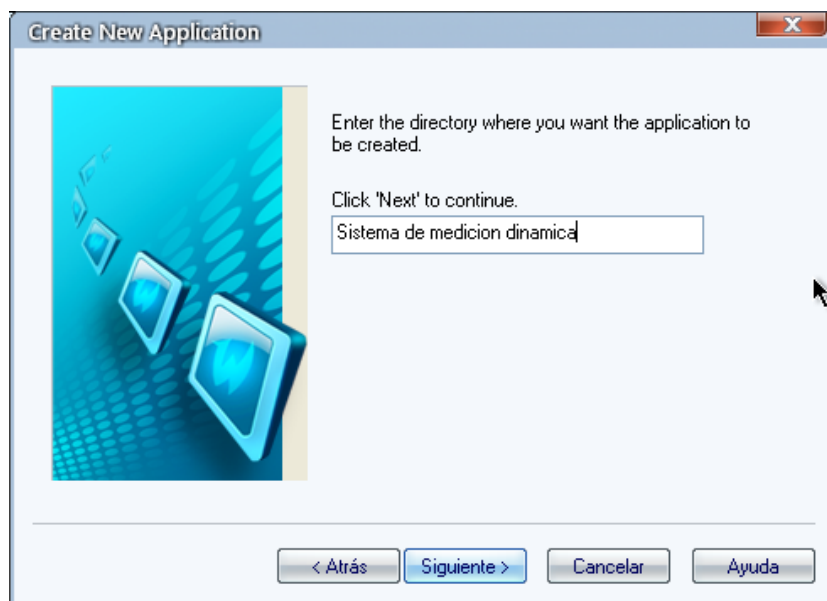


Figura 87. Definir nombre de la carpeta del proyecto

En el campo *Name*, digitar el nombre del proyecto (Sistema de medición dinámica). Ver Figura 88

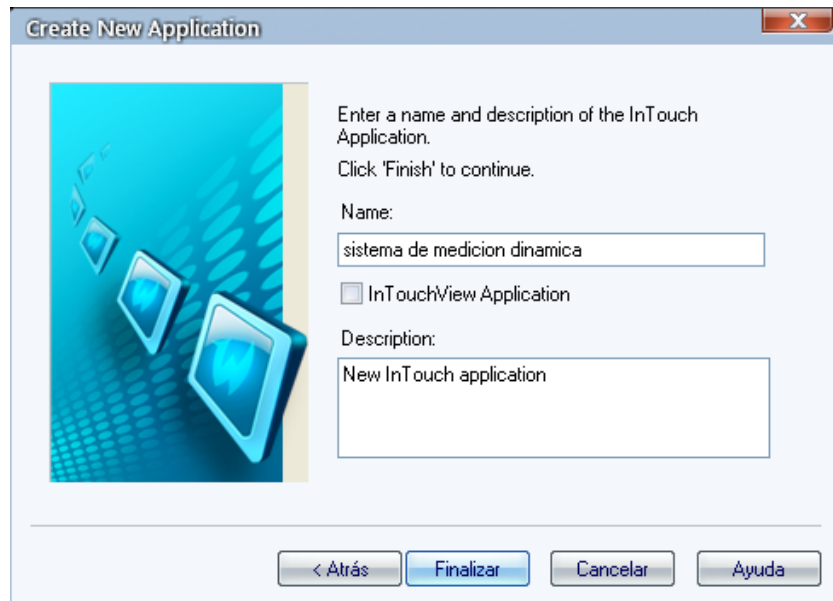


Figura 88. Definir nombre del proyecto

Si se siguieron todos los pasos correctamente, el proyecto creado se debe aparecer en la ventana de proyectos, como se muestra en la Figura 89.

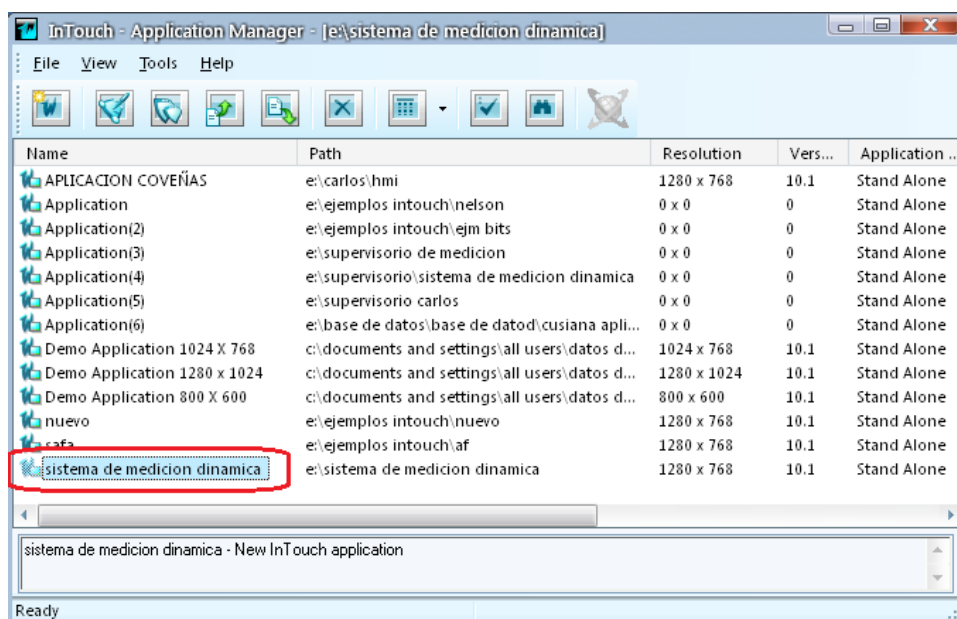


Figura 89. Ventana de proyectos del Application Manager

Doble clic en el nombre del proyecto creado para poder acceder a la interfaz de trabajo de intouch llamada WindowMaker. Seguidamente se despliega la siguiente ventana

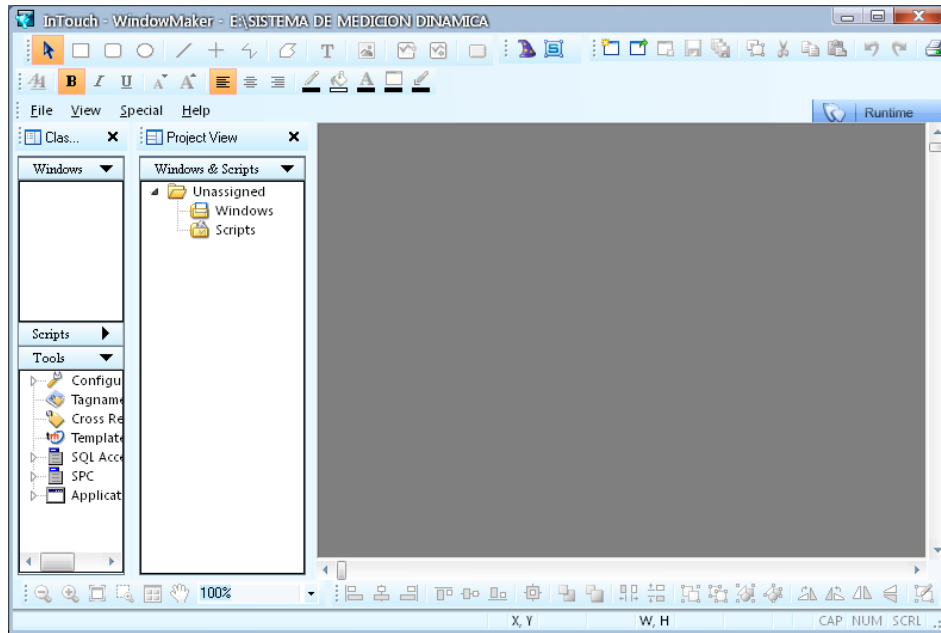


Figura 90. Ventana de trabajo WindowMaker

7.2. Entorno de trabajo de intouch “WindowMaker”

WindowMaker (WM) de InTouch es una herramienta de dibujo basada en gráficos por objetos, en lugar de gráficos por pixels. Básicamente podemos decir que creamos objetos (círculos, rectángulos, etc.) independientes unos de otros. Ello facilita la labor de edición del dibujo y, lo que es más importante, permite una enorme sencillez y potencia en la animación de cada uno de los objetos, independientemente o por grupos.

InTouch 10.1, tiene las siguientes barras de herramientas, que son el entorno de trabajo del proyecto.

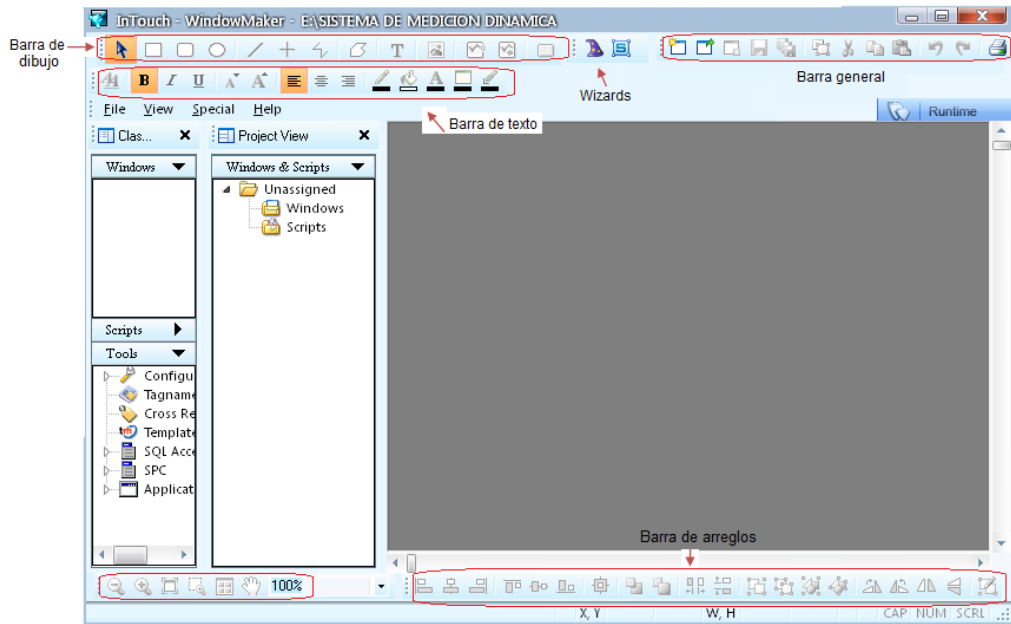


Figura 91. Entorno de trabajo WindowMaker

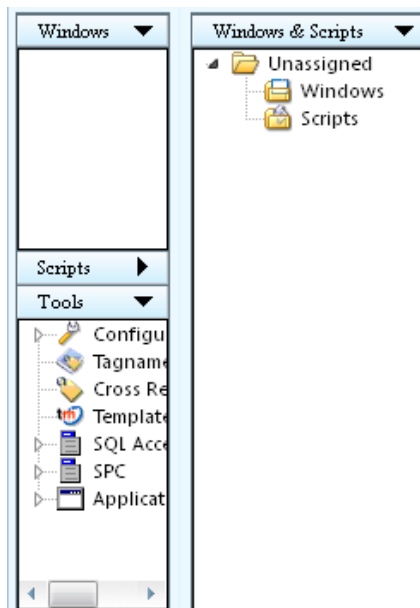


Figura 92. Campo para administración de ventana, scripts y carpetas

7.2.1 Wizards

En su más básico concepto, podría ser definido como "elementos inteligentes" que permiten que las aplicaciones **InTouch** puedan ser generadas de un modo más rápido y eficiente.

La versión 10.1 de **InTouch** dispone de los elementos WIZARDS que permiten crear rápidamente un objeto en la pantalla. Haciendo doble click sobre el objeto podemos asociarle links (animación), asignarlo a tagnames o incluso incluir una lógica en ese objeto. Si agrupamos varios

de estos objetos, podemos crear un elemento completo, acabado y programado, que lo podemos utilizar tantas veces como queramos.

7.3. Creación de una ventana

En InTouch se crean ventanas que permitirán la supervisión y el mando remoto del proceso, dichas ventanas deben estar configuradas en tamaño, tipo, color de fondo, tipo de trabajo, etc.

En el menú principal de la ventana de *WindowMaker* clic en *File; New Window*. En la ventana que aparece (Figura 93) defina las siguientes propiedades:

- **Name:** Nombre de la ventana
- **Comment:** Comentario de la ventana
- **Window color:** color de fondo de la ventana
- **Window type:**
 - Replace: La ventana aparecerá sobre la ventana activa solamente cuando se hace clic sobre su botón de activación
 - Overlay: La ventana permanecerá sobre cualquier ventana activa
 - Popup: Aparecerá cuando se hace clic sobre su botón de activación
- **Dimensions:**
 - X Location: Posición de inicio en el eje X
 - Y Location: Posición de inicio en el eje Y

Luego de haber definido todas las propiedades de la ventana clic en *OK*.

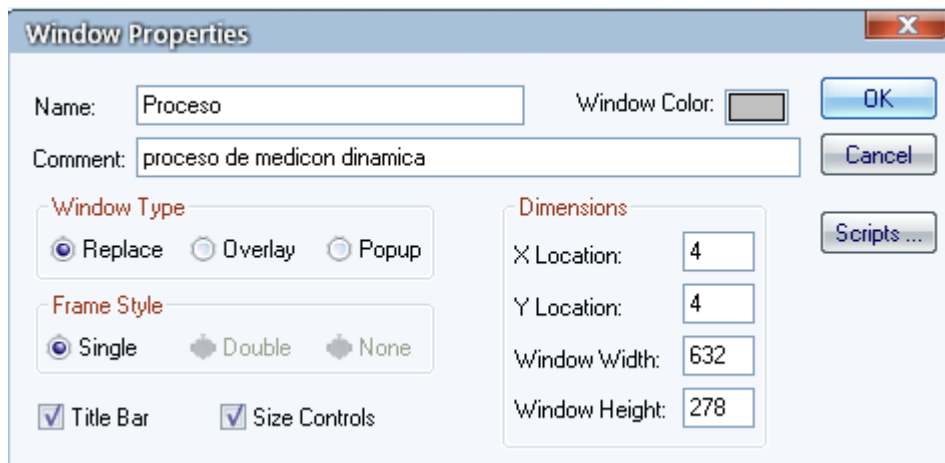


Figura 93. Propiedades de una ventana

Se podrá visualizar la siguiente ventana de trabajo, que se encuentra lista para definir los objetos y tagnames:

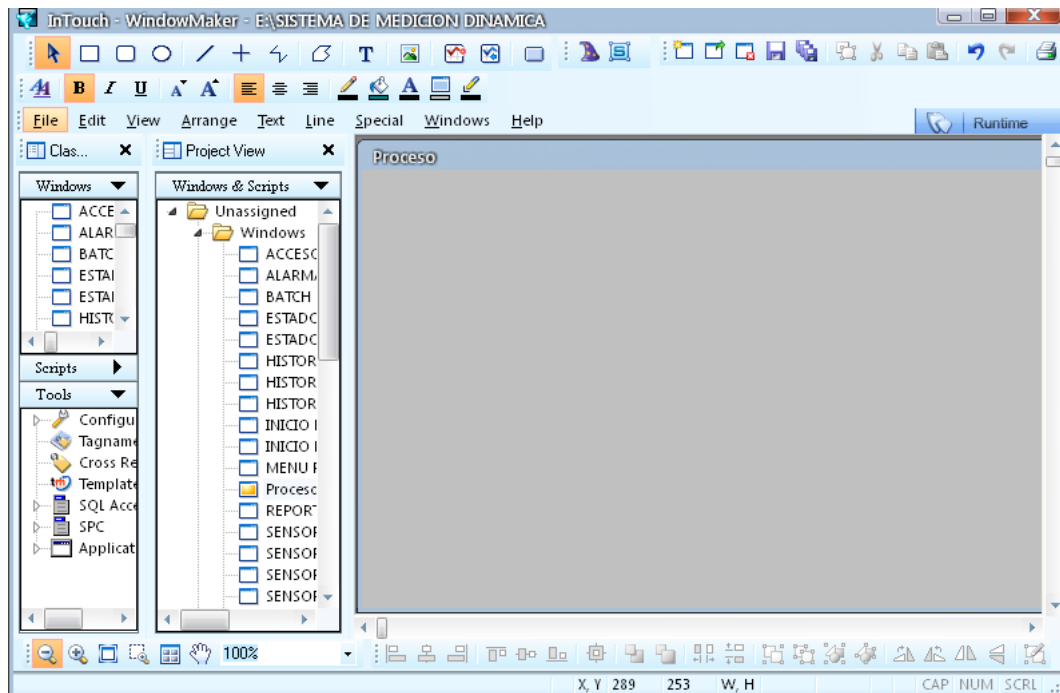


Figura 94. WindowMaker con la nueva ventana creada

7.4. Definición de Tagname

El diccionario de tagnames es el núcleo de **InTouch**. Durante el *runtime*, este diccionario contiene todos los valores de los elementos en la base de datos.

Para crear esa base de datos, **InTouch** necesita saber qué elementos la van a componer. Se debe, por lo tanto, crear una base de datos con todos aquellos datos que se necesitan para la aplicación.

A cada uno de estos datos (tags) se le asigna un nombre, y una dirección de I/O. Al final, dispondremos de un diccionario con todos los tagnames creados. Para crear tagname clic en *Especial* que se encuentra en la barra del menú (Figura 95), inmediatamente se desplegará un menú, clic en tagname Dictionary.

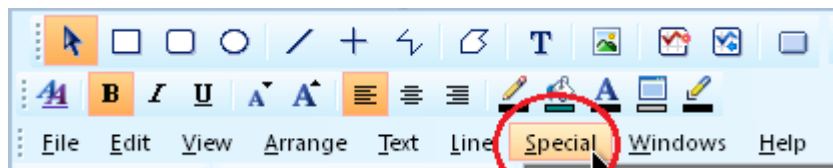


Figura 95. Menú *Especial*

En la Figura 96 clic en *New*, posteriormente se despliega una ventana como se muestra en la figura 97

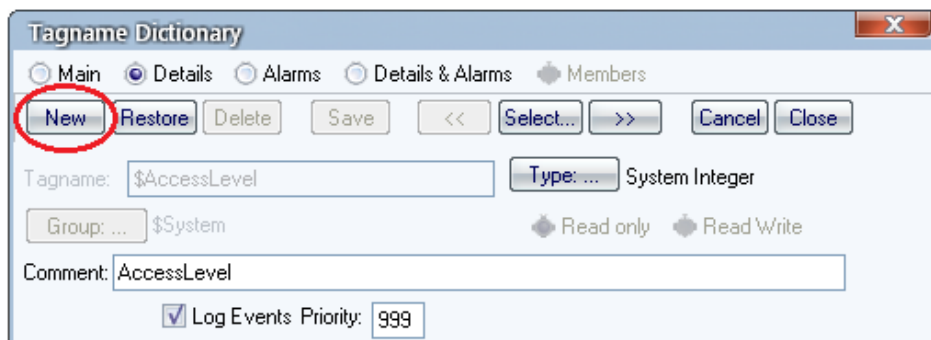


Figura 96. Tagname Dictionary

Desde la ventana “*Tagnames Dictionary*” definimos las tagnames y sus características. Existen diversos tipos de tagnames, según su función o características. Básicamente se dividen en:

- **Memory:** Tags registros internos de **InTouch**
- **I/O:** Registros de enlace con otros programas
- **Indirect:** Tags de tipo indirecto

Para los tres primeros tipos, disponemos de:

- **Discrete:** Puede disponer de un valor 0 ó 1
- **Integer:** Tagname de 32 bits con signo. Su valor va desde - 2.147.483.648 hasta 2.147.483.647
- **Real:** Tagname en coma flotante. Su valor va entre $\pm 3.4e38$. Todos los cálculos son hechos en 64 bits de resolución, pero el resultado se almacena en 32 bits
- **Message:** Tagname alfanumérico de hasta 131 caracteres de longitud

Rellenar los campos de ese tagname

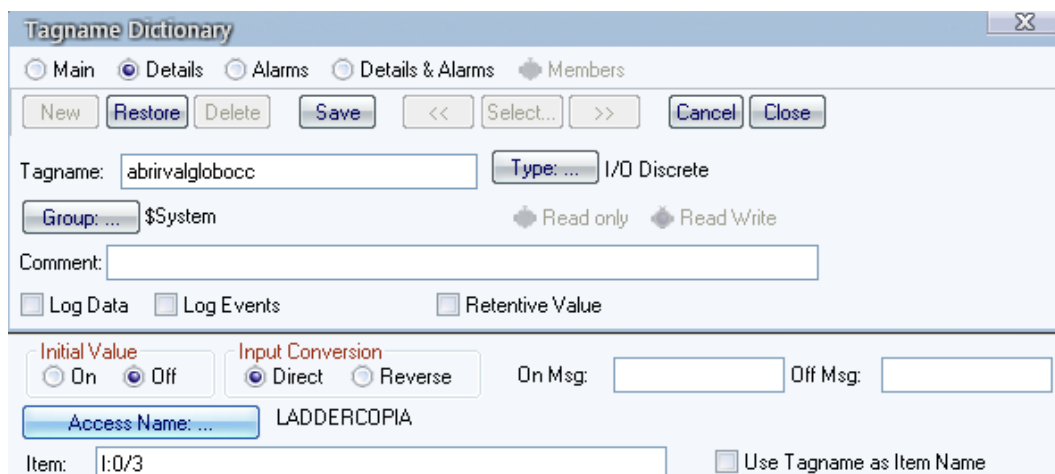


Figura 97. Tagname Dictionary New

Como se puede observar en la Figura 97 se creó un tagname llamado I/O de tipo *I/O Discrete*. I/O significa se va a utilizar para entrada y salida de datos y el tipo I/O Real porque se utiliza para entrada y salida de datos pero de punto flotante.

Una vez creado el tagname se necesita asociarlo con un nombre de acceso (*Access Name*). El Nombre de Acceso contiene la información que se usa para que InTouch se comunice con otra fuente I/O de datos en nuestro caso RsLinx. Incluyendo el nombre del nodo (node name), el nombre de aplicación (application name) y el nombre del tema (topic name).

- **Access Name:** Nombre de enlace. Puede ser un nombre aleatorio (se recomienda utilizar el mismo que el tópico).
- **Node Name:** Nombre del Nodo. Sólo hay que rellenarlo si vamos a leer datos de otro PC. En caso que los datos sean del mismo PC, dejarlo en blanco
- **Application Name:** Nombre de la aplicación de la que queremos leer (pe Excel, Siemens, etc.), en nuestro caso RsLinx
- **Topic Name:** Nombre del tópico del que queremos leer (pe. Libro1.xls, PLC1, etc.)
- **Which protocol To Use:** DDE para enlace DDE

Para especificar todo esto se ingresa a la ventana del nombre de acceso dando un click en el botón *Access Name*. A continuación se dá un clic en el botón *Add* para ingresar un nuevo nombre de acceso. De esta manera se recibe los datos en InTouch. (Ver figura 98).



Figura 98. Ventana del *Access Name*

7.5. Animación LINKS

Los enlaces de animación pueden combinarse para ofrecer movimientos, colores, cambios de tamaño y/o posición a los objetos. Incluyendo entradas de contacto discretas y analógicas; deslizadores horizontales,

verticales; pulsadores discretos y de acción; pulsadores para mostrar y aculatar ventanas, enlaces de color de línea, relleno y texto para valores y alarmas discretos y analógicos; enlaces de altura y ancho de objetos, enlaces de posición horizontal y vertical, y más.

Una vez un objeto o símbolo ha sido creado, este puede ser animado eligiendo los correctos enlaces de animación. Los enlaces de animación provocan que los objetos, símbolos cambien su apariencia por medio de la variación del valor de la base de los tagnames.

Para asignar una animación *Link* a un objeto, este deberá estar seleccionado. Haciendo dos veces clic sobre el objeto o símbolo deseado entramos directamente en el menú de *Animation Links* (Figura 99).

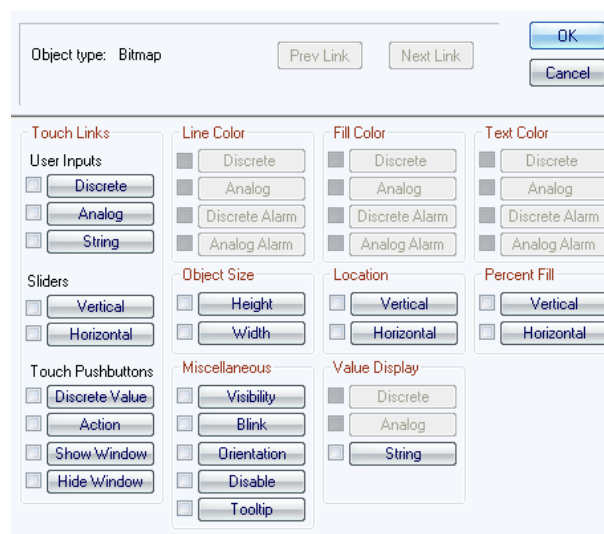


Figura 99. Ventana *Animation Links*

7.5.1. Animación objetos

Cada una de las funciones de *Animation Links* dispone de un submenú que deberemos rellenar. Se trata de las características propias de cada comando de animación. Los comandos de animación son los siguientes:

7.5.1.1. User Inputs

Al pulsar sobre *User Input*, intouch nos permite modificar un valor discreto, analógico (ya sea *Memory Type* o *I/O Type*) y permite modificar una cadena alfanumérica (Figura 100):

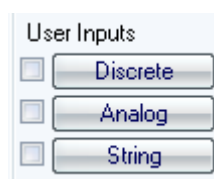


Figura 100. Comando *User inputs*

7.5.1.2. Value Sliders

Permite crear una barra de desplazamiento vertical u horizontal para seleccionar valores (Figura 101):

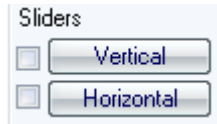


Figura 101. Comando *Sliders*

7.5.1.3. Touch Pushbutton

- **Discrete:** A diferencia de *Touch Value*, actúa como un pulsador, directamente sobre una señal 0/1
- **Action:** Permite ejecutar una lógica de acciones (llamada a otros programas, impresiones, etc).
- **Show/Hide Window:** Permite llamar a otras pantallas o hacerlas desaparecer del monitor.



Figura 102. Comando *Touch Pushbuttons*

7.5.1.4. Line Color

Permiten animar el color de línea de un objeto. Este cambio de color puede depender de un valor discreto/analógico o asociarse a una alarma de tipo discreto o analógico (Figura 103):



Figura 103. Comando *Line Color*

7.5.1.5. Fill Color

Permiten rellenar un objeto de un color. Este cambio de color puede depender de un valor discreto/analógico o asociarse a una alarma de tipo discreto o analógico (Figura 104):



Figura 104. Comando *Fill Color*

7.5.1.6. Text Color

Permiten cambiar el color de un texto. Este cambio de color puede depender de un valor discreto/análogo o asociarse a una alarma de tipo discreto o analógico (Figura 105):



Figura 105. Comando *Text Color*

7.5.1.7. Object Size

Permite asociar el tamaño vertical/horizontal de un objeto a un registro (Figura 106):



Figura 106. Comando *Object Size*

7.5.1.8. Miscelaneus

- **Visibility:** Permite que un objeto aparezca/desaparezca de la pantalla
- **Blink:** Intermitencia del objeto
- **Orientation:** Modifica orientación del objeto
- **Disable:** Hace que un objeto "táctil" deje de serlo



Figura 107. Comando *Miscelaneus*

7.5.1.9. Location

Permite modificar la posición del objeto (Figura 108):

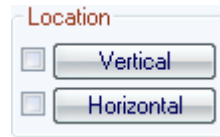


Figura 108. Comando *Location*

7.5.1.10. Value Display

Se utiliza para visualizar un valor discreto, analógico o alfanumérico (figura 109):



Figura 109. Comando *Value Display*

7.5.1.11. Percent Fill

Permite asociar un registro tagname a una barra grafica, por ejemplo para hacer el llenado o vaciado de un tanque (Figura 110):

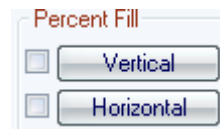


Figura 110. Comando *Percent Fill*

7.6. Animación con Wizards

Para hacer una animación con *Wizards* clic en el icono *Wizards* (en el sombrero del mago, Figura 111). Aparece una ventana llamada *Wizards Selection* (Figura 112), donde se tiene las librerías de objetos de alarmas, botoneras, medidores, paneles, etc.



Figura 111. Icono de *Wizards*

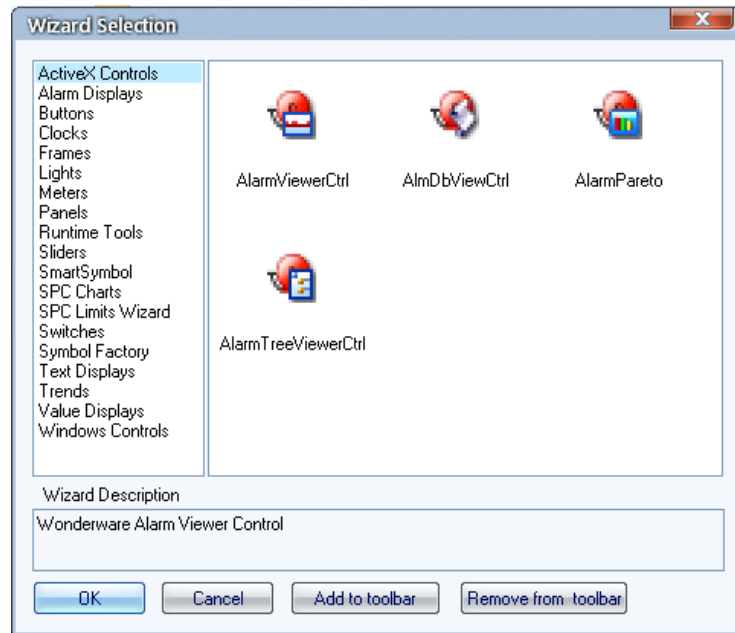


Figura 112. Ventana para seleccionar el Wizard

De estas librerías la más importante es la opción *Symbol Factory*, donde se encuentra otras librerías de objetos más especializados y de aplicación para procesos de ingeniería. Ver figura 113. Hacer clic sobre *Symbol Factory*, dentro escoger *Symbol Factory* y clic en *OK*

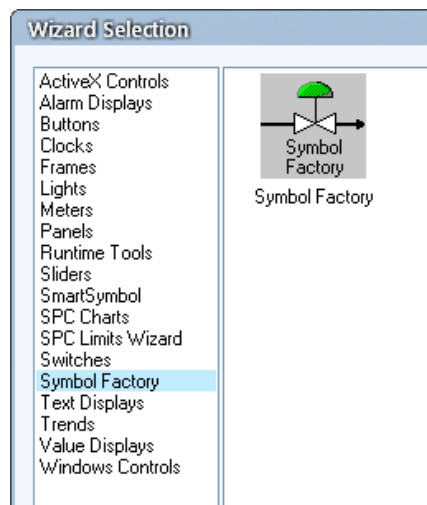


Figura 113. Icono *Symbol Factory*

Aparentemente desaparece la ventana de objetos, clic sobre la ventana de trabajo y aparece la ventana de la figura 114 “*Symbol Factory by Reichard Software*”



Figura 114. Librería *Symbol Factory*

Se puede elegir categorías de objetos como se menciona anteriormente más especializada y de aplicación de ingeniería, por ejemplo la categoría de sensores como se muestra en la figura 115.

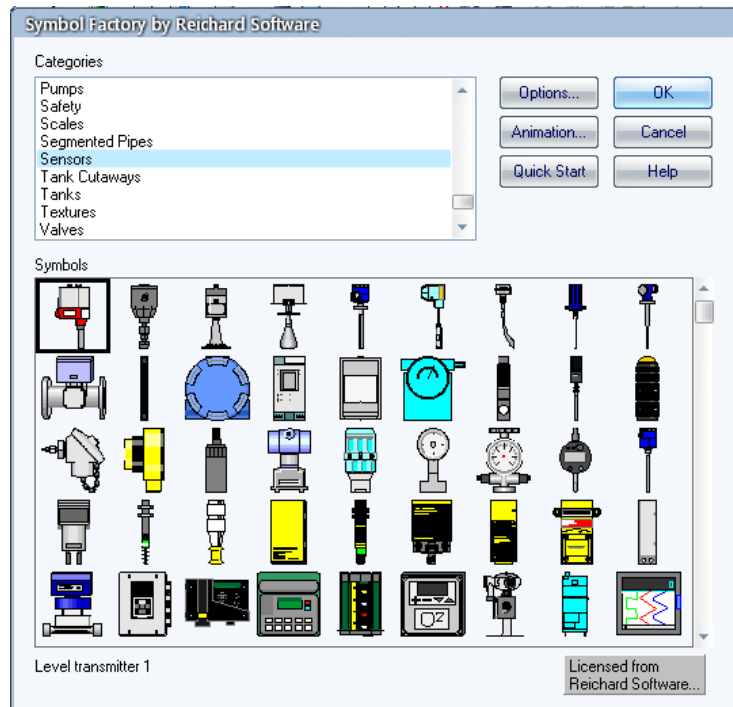
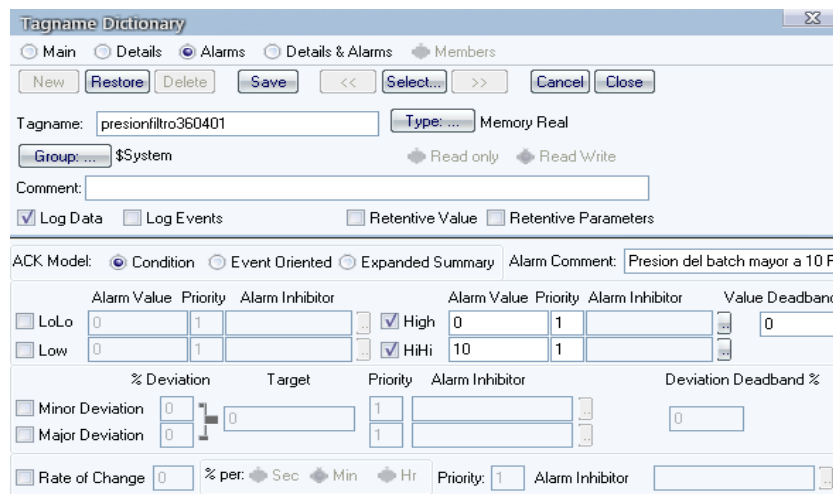


Figura 115. Categoría *Sensor*

7.7. Alarmas

Intouch permite configurar y establecer prioridades de alarmas rápidamente. Hasta 999 prioridades diferentes, cambios de color de acuerdo con el estado de la alarma y hasta 8 niveles de jerarquía entre los grupos de alarma con posibilidad hasta de 16 subgrupos para cada uno de ellos. No hay límite de número de alarmas. Se puede visualizar todas o un extracto de ellas de forma histórica o en tiempo real y grabar en un disco o imprimir en diferentes formatos personalizables.

Para crear una alarma se debe crear la tagname y en seleccionarla como *Log Data*, digitar la prioridad y el valor mínimo y máximo de la alarma. Ver Figura 116.



The screenshot shows the 'Tagname Dictionary' dialog box with the 'Alarms' tab selected. The configuration is as follows:

- Tagname: presionfiltro360401
- Type: Memory Real
- Group: \$System
- Comment: (empty)
- Log Data: (checked)
- Log Events: (unchecked)
- Retentive Value: (unchecked)
- Retentive Parameters: (unchecked)
- ACK Model: Condition (selected)
- Event Oriented: (unchecked)
- Expanded Summary: (unchecked)
- Alarm Comment: Presion del batch mayor a 10 P
- LoLo: (unchecked), Value: 0, Priority: 1, Alarm Inhibitor: (empty)
- Low: (unchecked), Value: 0, Priority: 1, Alarm Inhibitor: (empty)
- High: (checked), Value: 0, Priority: 1, Alarm Inhibitor: (empty), Value Deadband: 0
- HiHi: (checked), Value: 10, Priority: 1, Alarm Inhibitor: (empty), Value Deadband: (empty)
- Minor Deviation: (unchecked), % Deviation: 0, Target: 0, Priority: 1, Alarm Inhibitor: (empty), Deviation Deadband %: (empty)
- Major Deviation: (unchecked), % Deviation: 0, Target: (empty), Priority: 1, Alarm Inhibitor: (empty), Deviation Deadband %: (empty)
- Rate of Change: (unchecked), % per: (empty), Unit: (empty), Priority: 1, Alarm Inhibitor: (empty)

Figura 111. Definición de alarma

Después de definir la alarma como *Log Data*, dentro del icono del *Wizards* de la caja de herramientas encontraremos el objeto *Alarmas* (Figura 117). Para crear una ventana de alarmas, basta con seleccionar el objeto como si se tratara de un rectángulo o un círculo. Definimos su tamaño, y la ventana de alarmas quedará creada.

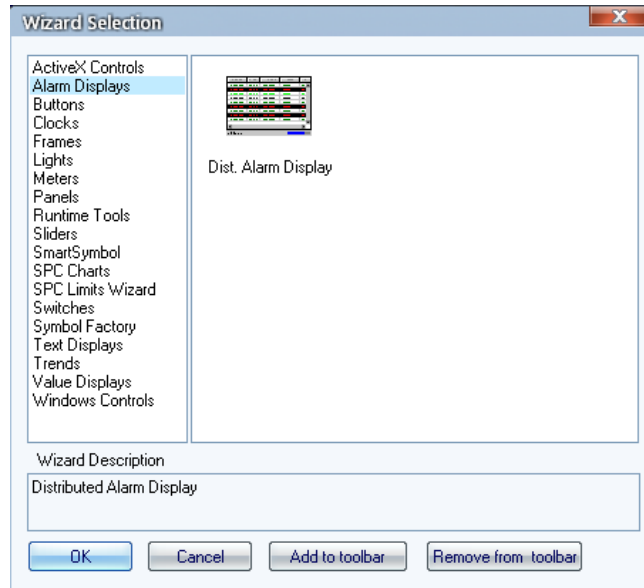


Figura 117. Display de alarmas

7.8. Curvas históricas y rales

El paquete de software **InTouch** permite desplegar curvas y tendencias en pantalla tanto en tiempo real como de valores históricos.

Para la creación de una curva real se dispone de una herramienta en la barra de herramientas que permite crear una curva en tiempo real. Para ello, se selecciona y se crea un rectángulo del tamaño que se desea para el gráfico. Cada gráfico puede visualizar hasta 4 lápices. (Figura 118)



Figura 118. Icono de Alarmas reales

Para alarmas histórica InTouch permite la visualización de históricos distribuidos (gestión de los históricos bajo una estructura cliente/servidor en una red de ordenadores). Las curvas históricas permiten visualizar la evolución con respecto al tiempo de un dato en forma de curva o tendencia. Este dato debe haber sido almacenado previamente, por lo que el tagname visualizado en este tipo de curvas debe haber sido previamente definido como del tipo logged.



Figura 119. Icono de Alarmas históricas

7.9. Programación de lógica Script

El lenguaje Script de Intouch es flexible y fácil de utilizar. El *Wonderware Scripts* incrementa la capacidad de Intouch de proveer la habilidad de ejecutar comandos y operadores lógicos.

Utilizando los *Scripts*, una gran variedad de sistemas personalizados y automáticos pueden ser creados.

7.9.1. Tipos de Scripts

- **Application Scripts:** Enlace se uso enteros.
- **Winsows Scripts:** Enlace de ventanas específicas.
- **Key Scripts:** Enlace de una llave específica o a una combinación dominante en el teclado.
- **Condition Scripts:** Enlace de un Tagname a una expresión directa.
- **Data Change Scripts:** Enlace de un Tagname y/o a un dotfield del tagname solamente.

Para programar en *Scripts* clic en *Special*, seguidamente en *Scripts*, escoger el tipo de script deseado. Ver Figura 120.

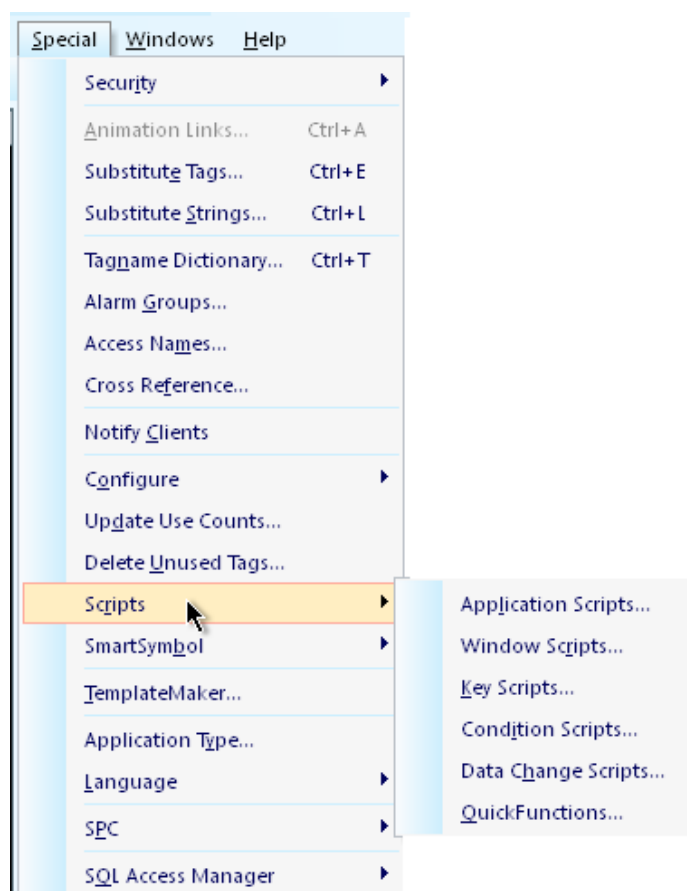


Figura 120. Tipos de *Scripts*

8. ANEXO H. COMUNICACIÓN ENTRE INTOUCH Y RSLOGIX 500

En nuestro proyecto se cuenta con dos brazos de medición, con lo cual cada brazo tiene un respectivo PLC. Para el brazo número 350501 se tomo el nodo 1, para brazo número 350601 se tomo el nodo 2.

Después de obtener el ladder del sistema de medición dinámica, se verifica que todas las direcciones de las marcas auxiliares coincidan con las direcciones utilizadas en los tagnames introducidas en Intouch, se continúa a configurar el enlace con Intouch.

Direccionamiento de I/O en intouch

Para que los datos provenientes desde el convertidor de frecuencia puedan ser leídos por Intouch, es necesario asignar un tagname a cada variable a ser monitoreada. Un tagname es la manera como reconoce Intouch a los datos con los que trabaja; los cuales deben ser, para este caso, de tipo I/O.

El siguiente paso en este proceso, es configurar InTouch. Un "Access Point" y crear "Tag Names" para permitir la visualización gráfica en el programa y para que pueda realizarse la comunicación entre el Intouch y el Rslogix 500. Un *Access Name* sirve para asociar a cada uno de los I/O tagnames de Intouch. *Access Name* es aquel que contiene la información del nombre de la aplicación (*Application Name*), y el nombre del tópico (*Topic Name*).

En el cuadro de diálogo de la Figura 121, se debe llenar toda la información requerida. El *Access Name* es arbitrario y para simplicidad debería coincidir con el *Application Name*. El *Application Name* debe ser "RSLINX", ya que se está usando el I/O Server con dicho nombre. El *Topic Name* debe ser el mismo que se configuró en el I/O Server "*Topic Definition*". Se debe asegurar que la opción DDE esté activa para permitir una correcta comunicación.

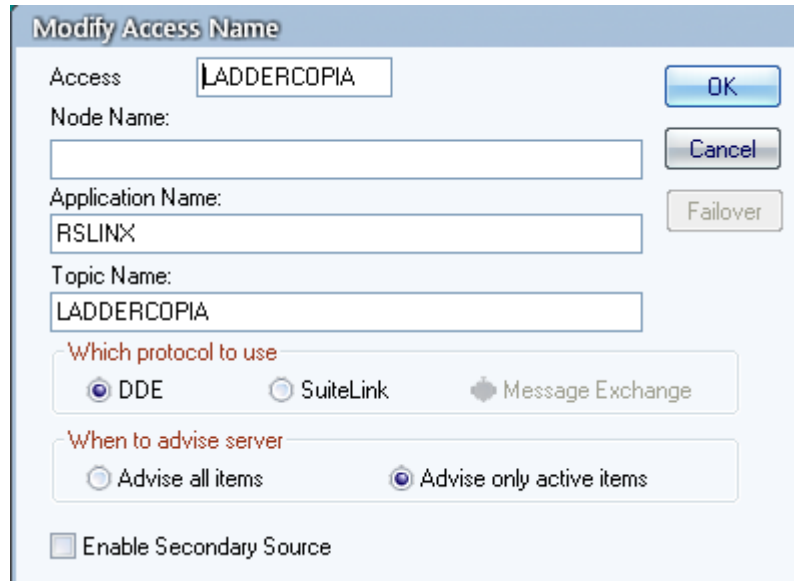


Figura 121. Ventana del *Access Name*

8.2. RsLinx

En este ítem se describe el proceso de configuración del driver necesario para realizar la comunicación con el PLC.

Para empezar clic en la *Barra de inicio*, inmediatamente aparecerá la ventana de la figura 122. Seleccionar *RsLinx Classic*.

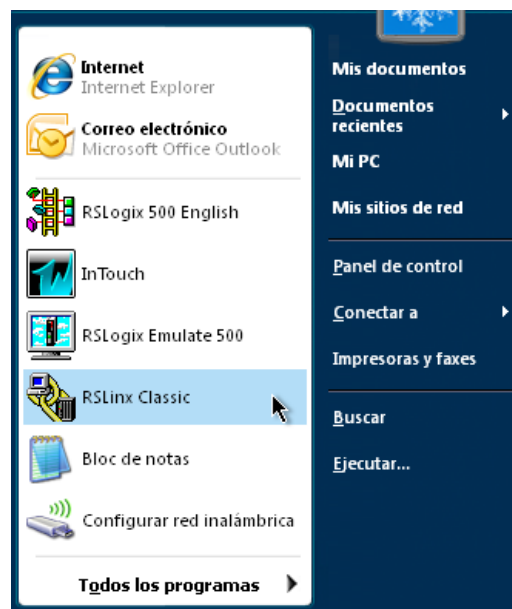


Figura 122. Barra de inicio

Automáticamente aparecerá una ventana como se muestra en la figura 123, ventana principal de *RsLinx Classic Gateway*

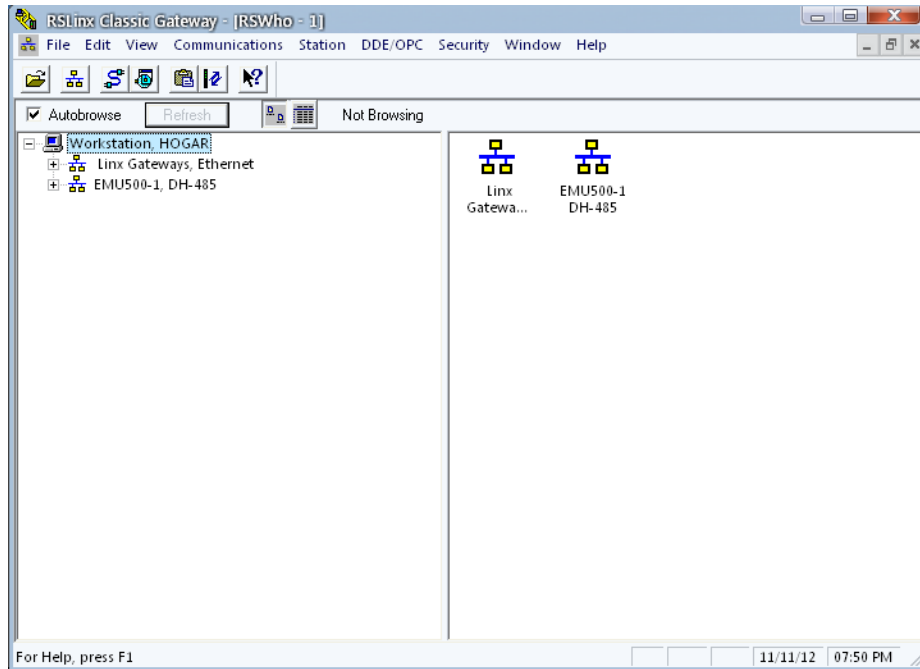


Figura 123. Ventana principal de *RsLink Classic Gateway*

Al no contar con un PLC real, primero se configura la emulación es decir que Intouch, trabajara en el *PC* sin la necesidad de contar con un PLC real.

En el menú *Communications* en la ventana principal seleccione *Configure Drivers* como se muestra en la figura 124.

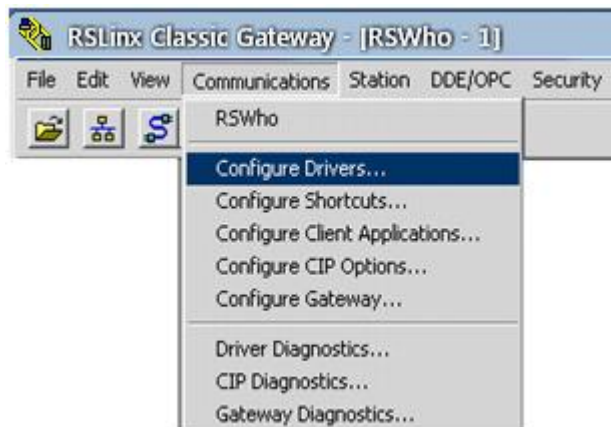


Figura 124. Menú *Communications*

En la ventana *Configure drivers*, clic izquierdo sobre la lista de los drivers disponibles, seleccione *SLC 500 (DH485) Emulator driver*, clic en *Add new*. Ver figura 125.

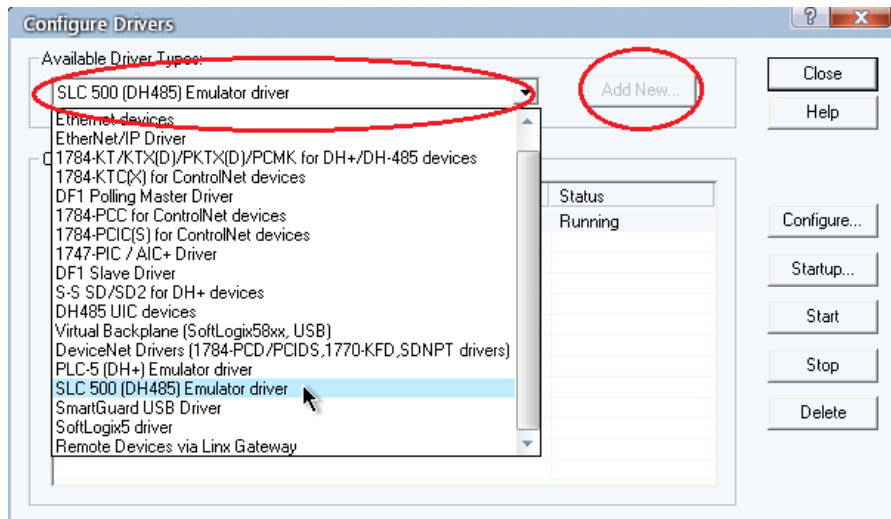


Figura 125. Configuración del *Driver*

Se desplegará una ventana donde se solicita al nombre del driver, dejar el nombre que aparece por defecto. Presionar **OK**. Ver figura 126.

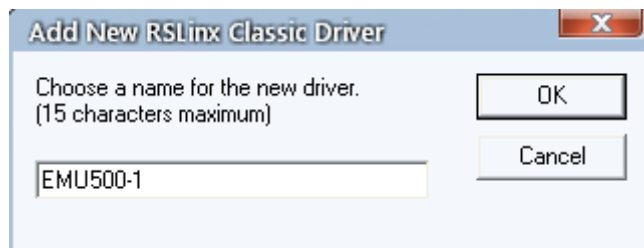


Figura 126. Ventana *Add new RSLinx Classic Driver*

Después aparece una ventana *Configure SLC 500 (DH485) Emulate driver*, clic en **OK**.

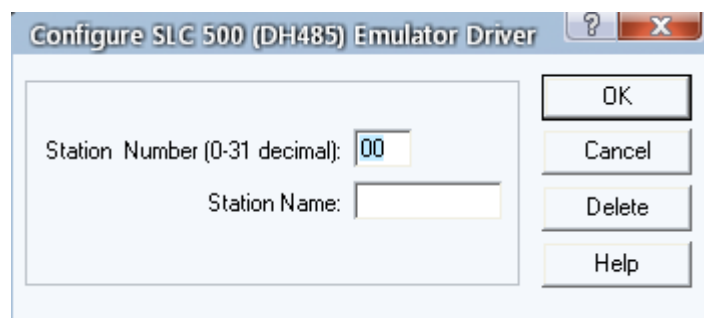


Figura 127. *Configure SLC 500 (DH485) Emulator river*

Si la configuración se hizo correctamente, el driver configurado se muestra con *Status Running* (Ver Figura 128), clic en *Close*

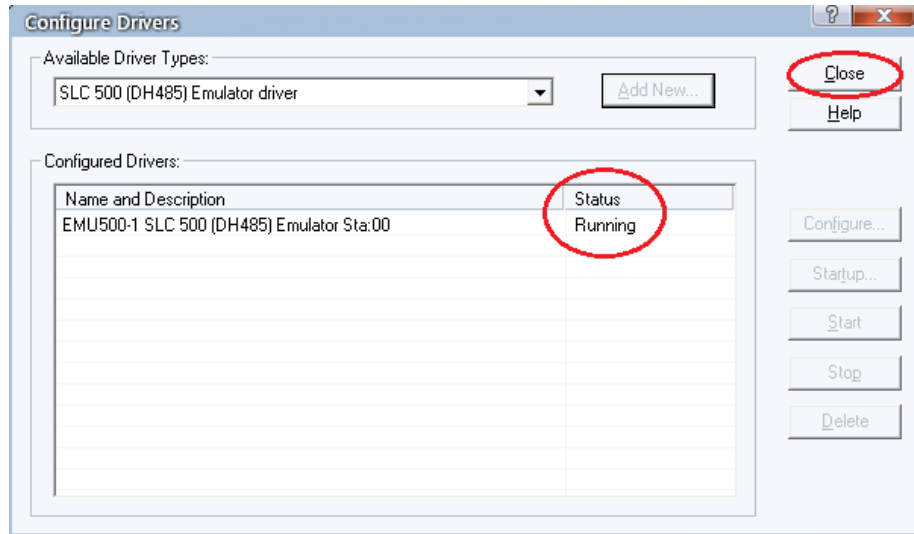


Figura 128. Driver creado en modo *Running*

En el menú principal elija la pestaña *DDE/OPC*, seguidamente clic en *Topic Configuration*.

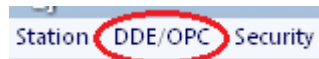


Figura 129. Menú DDE/OPC

Se desplegará una ventana como se muestra en la figura 130, clic en *New*, escribir el mismo nombre del código ladder, en nuestro caso el ladder del brazo 350501 se llama *LADDER2* y el ladder del brazo 350601 se llama *LADDERCOPIA*. Para el enlace es indispensable que el nombre del *Topic List*, debe ser el mismo nombre de nuestro código ladder, de no ser así no funcionará la emulación con Inotuch.

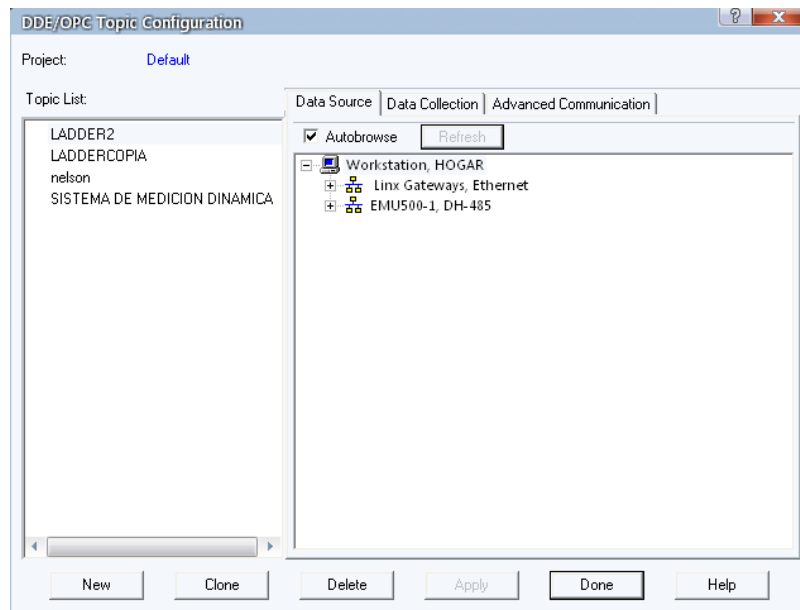


Figura 130. Configuración DDE/OPC

En el icono *Data Collection*, *Processor type*, elegir la opción *SLC 503+* (Figura 131).

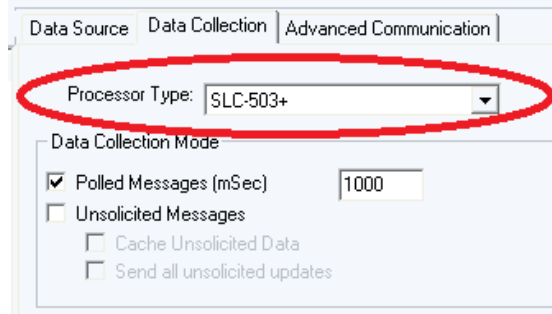


Figura 131. Processor Type

Luego en la pestaña *Advanced Communication*, en *Communication Driver* elegir *EMU 500-1 SLC 500 (DH485) Emulator ST. 00*, el cual ya fue configurado al elegir el driver de emulación. Para nuestro proyecto, en el campo *Station (decimal)*, para la comunicación del ladder llamado *LADDER2* del brazo 350501 la estación es la número 1 (Ver Figura 132), para la comunicación del ladder llamado *LADDERCOPIA* del brazo 350601 la estación es la número 2 (ver figura 133)

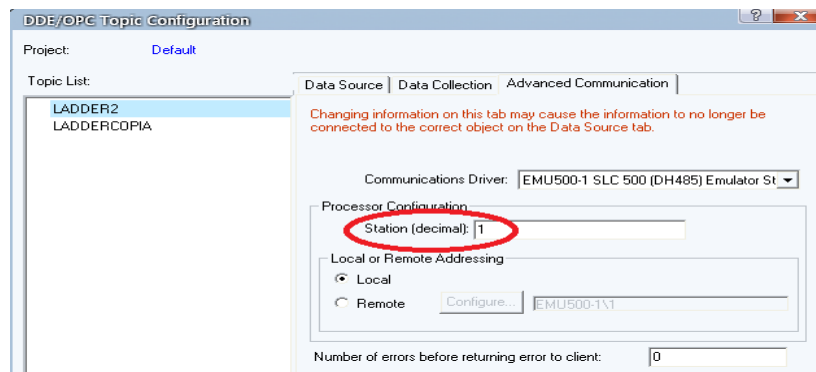


Figura 132. Advanced Communication del LADDER2

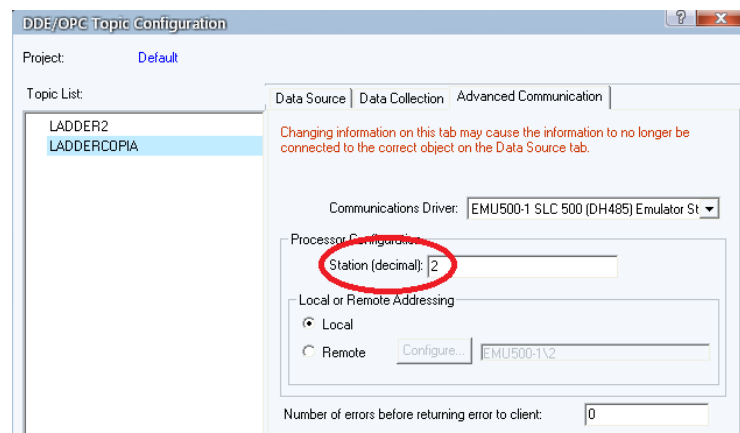


Figura 133. Advanced Communication del LADDERCOPIA

8.3. RsLogix Emulate 500

Para la simulación del código ladder ingresar a RsLogix Emulate 500.

Clic en abrir archivo y cargar el primer código ladder llamado LADDER2 (figura 134)

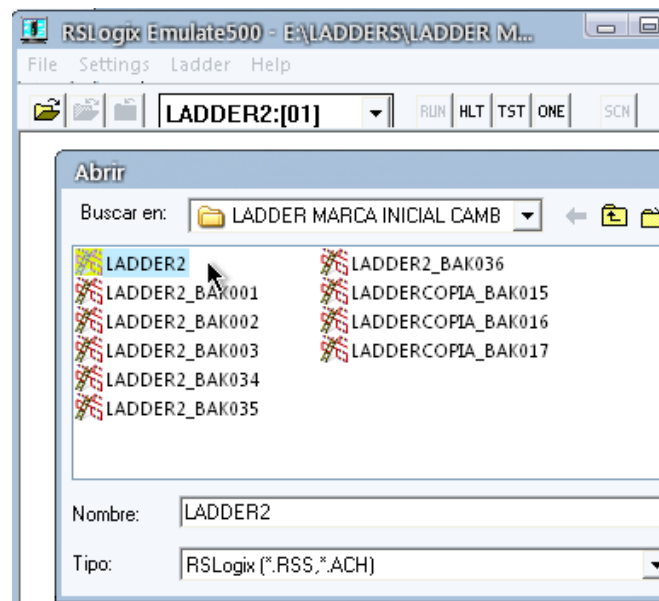


Figura 134. Carga del LADDER2

Después de cargar el código ladder aparecerá la ventana de configuración en la cual debemos elegir "1" para el LADDER2 (Ver Figura 135).

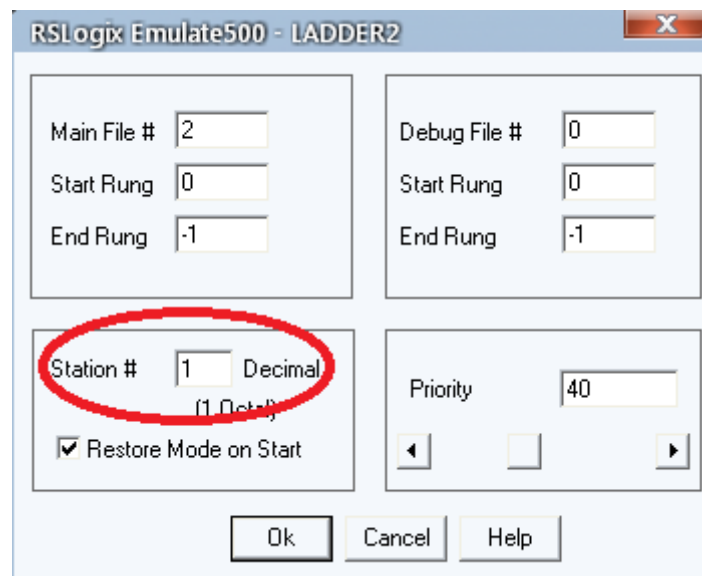


Figura 135. Configuración de la estación del LADDER2

En seguida se procede a cargar el segundo ladder llamado LADDECOPIA.

Al cargar el ladder aparecerá la ventana de configuración en la cual debemos elegir "2" para el LADDECOPIA (Ver Figura 136).

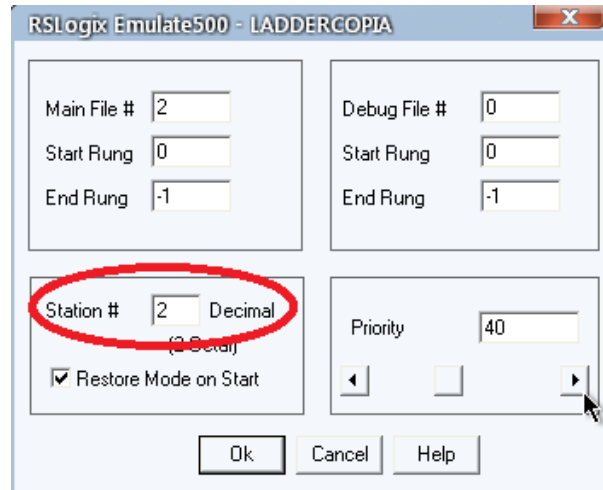


Figura 136. Configuración de la estación del LADDERCOPIA

Si la configuración no arroja ningún error, deberán aparecer los nombres de los códigos ladder en el campo de programas de simulación como se muestra en la Figura 137.

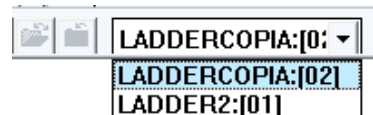


Figura 137. Ladder correctamente cargados y configurados

De esta forma ya están listos los proyectos para la descarga a la memoria virtual del *Emulator*.

8.4. Descarga del Programa al PLC virtual

Ir nuevamente al RsLogix 500, en el menú *Comms* seleccionar *System Comms*. Si todos los pasos se siguieron de la manera correcta debe aparecer la ventana de la figura 138. Todos los campos enmarcados con color rojo son los que se configuraron.

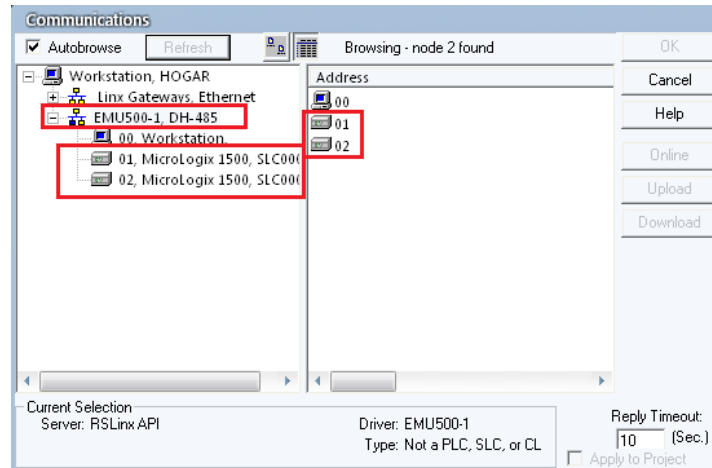


Figura 138. Ladder correctamente cargados y configurados

Para finalizar clic en *Download*, presionar *OK*.

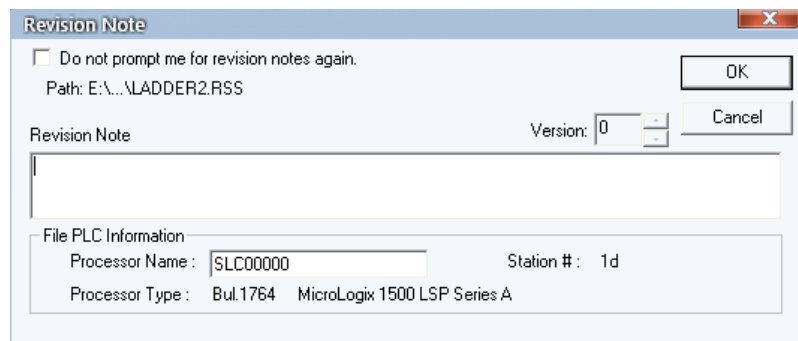


Figura 139. *Download*

Clic en la siguiente ventana (figura 140)

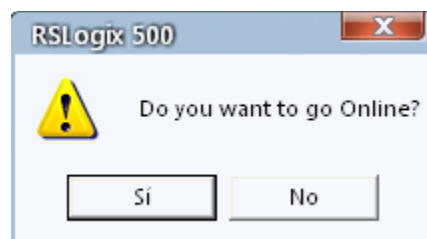


Figura 140. *Online*

Cambiar el modo del PLC a *Run* y con esto el código ladder está listo para la emulación con Intouch.

9. ANEXO I. CODIGO LADDER DEL PROYECTO

