

UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA

DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA



PROYECTO FIN DE CARRERA

Simulación de la automatización de procesos Con Unity Pro

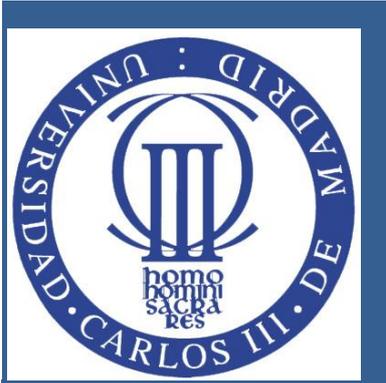
AUTOR: MARÍA MUÑOZ HORCAJUELO

TUTORA: MARÍA MALFAZ VÁZQUEZ

2012

Quisiera expresar mi profundo agradecimiento a mis padres y hermana por enseñarme a perseverar ante las dificultades, y a Ricardo por la motivación y el apoyo que me ha demostrado. Vosotros cuatro habéis sido imprescindibles para llegar hasta aquí.

También a Abueli, porque siempre cuidas de mí. Gracias.



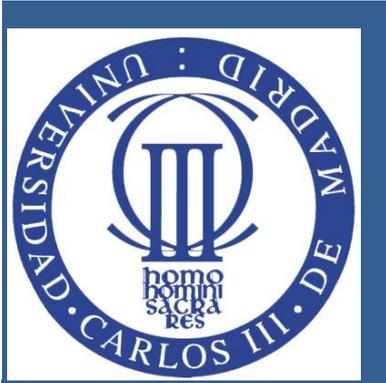
ÍNDICE DE CONTENIDOS

Simulación de la automatización de procesos con Unity Pro

Contenido

1	MOTIVACIÓN	11
1.1	Motivación del Proyecto	11
1.2	Objetivos	11
1.3	Estructura del documento.....	12
2	INTRODUCCIÓN	14
2.1	La automatización y su importancia en la docencia	14
2.2	Simulador: concepto y función	16
2.3	Uso de los simuladores	17
2.4	Aplicación de los simuladores en la docencia	18
2.5	Ventajas de los simuladores docentes	19
3	SOFTWARE UNITY PRO - SCHNEIDER	21
3.1	Descripción y características	21
3.2	Estandarización en la programación de control industrial	22
3.3	Tipos de variables, direcciones, bits y palabras de sistemas.....	25
3.3.1	Variables	25
3.3.2	Direcciones	26
3.3.3	Bits y palabras de sistema	26
3.4	El simulador de Unity Pro y comparativa con PLC	27
3.4.1	Simulador	27
3.4.2	Simulador Vs PLC.....	27
3.5	Diferencias entre un sistema Scada y un simulador	28
4	EJEMPLO DESARROLLADO.....	30
4.1	Presentación del problema	30
4.2	Configuración del entorno de trabajo y definición de variables necesarias	32
4.3	Programación de la solución del problema: diagrama SFC.....	39
4.3.1	Programación de etapas o pasos	42
4.3.2	Programación de Transiciones	44
4.3.3	Programación de Macroetapas	45
4.3.4	Lenguaje LD	47
4.3.5	Lenguaje ST.....	66
5	SIMULACIÓN DEL EJEMPLO PROGRAMADO	74
5.1	Pantallas del operador	74

5.1.1	Definición y creación	74
5.1.2	Librería y posibilidades en el desarrollo de elementos.....	77
5.1.3	Configuración del ejemplo desarrollado	91
5.2	Compilación y transferencia del Proyecto en modo simulación.....	106
5.3	EVOLUCIÓN DEL SISTEMA	110
5.3.1	Botones de comando	110
5.3.2	Tablas de animación.....	115
6	CONCLUSIONES Y APLICACIONES FUTURAS.....	122
6.1	CONCLUSIONES	122
6.2	APLICACIONES FUTURAS	123
7	BIBLIOGRAFÍA.....	125



ÍNDICE DE FIGURAS

Simulación de la automatización de procesos con Unity Pro

Figura 1: Autómata del laboratorio de Automatización de UC3M	15
Figura 2: Lenguajes IEC-1131-3	23
Figura 3: Representación bloque de función en LD y ST	24
Figura 4: Representación diagrama SFC.....	25
Figura 5: Plano de la instalación.....	30
Figura 6: Elección CPU en Unity Pro	32
Figura 7: Explorador de Proyectos en Unity Pro	33
Figura 8: Configuración del Rack	33
Figura 9: Variables de entrada	34
Figura 10: Variables de salida	34
Figura 11: Variables elementales	35
Figura 12: Tabla de Variables Elementales.....	36
Figura 13: Entradas Bastidor 1	36
Figura 14: Salidas Bastidor 1	37
Figura 15: Entradas Bastidor 2	37
Figura 16: Salidas Bastidor 2	38
Figura 17: Variables creadas para la simulación.....	38
Figura 18: Creación de una nueva sección SFC.....	39
Figura 19: Diagrama SFC	40
Figura 20: Elementos SFC	41
Figura 21: Elementos necesarios para crear el diagrama SFC.....	41
Figura 22: Programación de paso en SFC	42
Figura 23: Acción ligada a un sección en la creación de pasos.....	43
Figura 24: Acción ligada a una variable en la programación de pasos	43
Figura 25: Tipo de lenguaje	44
Figura 26: Condición de transición ligada a una variable.....	44
Figura 27: Condición de transición ligada a una sección.....	45
Figura 28: Tipo de lenguaje	45
Figura 29: SFC principal	46
Figura 30: Condición de transición ligada a una sección.....	47
Figura 31: Acción ligada a una sección.....	48
Figura 32: Alternativas de elementos para programación lenguaje LD	48
Figura 33: Etapa reposo	49
Figura 34: "Transición Producto_A/Producto_B/L"	49
Figura 35: "Transición Producto_A"	49
Figura 36: "Transición Producto_B"	50
Figura 37: "Transición Limpieza"	50
Figura 38: Propiedad del Paso "Dos_A"	50
Figura 39: Acción "Prod1"	51
Figura 40: Acción "V1_V2"	51
Figura 42: Macroetapa Mezcla	52
Figura 43: Macroetapa Mezcla-"Etapa Mezcla_IN".....	53
Figura 44: "Transición Calentar"	53
Figura 45: Acciones de macroetapa mezclar.....	53
Figura 46: Acción "Motor".....	54
Figura 47: Acción "Apagar_M"	54
Figura 54: "cerrar_valvulas"	57
Figura 55: Transición "Fin_llenado"	57
Figura 56: Macroetapa Limpieza.....	58
Figura 57: Propiedad del pasoIN "Limpiar_IN".....	59

Figura 58:Etapa LIMPIEZA_IN.....	59
Figura 59: Propiedad de paso "llenado_L"	60
Figura 60: Etapa Llenado.....	60
Figura 61: Transición "Depósito lleno"	60
Figura 62: Propiedad del paso "Agitar_L"	61
Figura 63: Transición "temp_1min"	61
Figura 64: Propiedad del paso: "Vaciar_L".....	61
Figura 65: sección "vaciado_agua"	62
Figura 66: sección "llenado_Tanque_A_mitad.....	62
Figura 67: "Etapa Vaciado"	62
Figura 68: "Transición "Depósito_vacio".....	62
Figura 69 :Propiedades de paso "Contador"	63
Figura 70: Etapa "Incrementar_contador".....	63
Figura 71: Transición "Contador_dos_veces".....	63
Figura 72: Transición "Contador_Menor_de_dos".....	63
Figura 73: Transición T_85	64
Figura 74:Etapa calentar_Agua.....	65
Figura 76: Alternativas de elementos para programación lenguaje ST	66
Figura 77: Código ST Dosificación Producto_B	68
Figura 78: Primera parte Código ST Dosificación Producto B comentado.....	69
Figura 79: Segunda parte Código ST Dosificación Producto B comentado.....	70
Figura 80: Código ST Dosificación Producto_B	71
Figura 81: Primera parte Código ST Limpieza comentado.....	72
Figura 82: Creación Pantalla del operador	75
Figura 83: Gráfico de la Instalación.....	76
Figura 84: Librería de Pantalla del operador.....	77
Figura 85: Desagrupar elemento motor	78
Figura 86: Agrupar figuras en Unity Pro.....	78
Figura 87: Tipos de elementos estáticos y dinámicos.....	89
Figura 88: Variables Simulación	90
Figura 89: Instalación	91
Figura 91: Variables Simulador para Depósito de mezcla	93
Figura 92: Variables INT "Dosificador_A" y "Llenar_producto_B"	94
Figura 93: Variables INT "Dosificador B" y "Llenar_producto_B"	94
Figura 94: Variables INT "Nivel_A" y "Llenar_producto_B2"	95
Figura 95: Variables INT "Nivel_B" y "Llenar_producto_A2".....	95
Figura 96: Variable INT "Agua_limpieza".....	96
Figura 98 : Propiedad del objeto: "pestaña Tipo de animación"	97
Figura 99: Propiedad del objeto: "pestaña Esquema".....	97
Figura 100: Variable INT "Tanque_A"	98
Figura 101: Variable INT "Tanque_vaciado"	99
Figura 102: Variable INT "Tanque_B"	99
Figura 103: Etiquetas para el cambiador de agua	100
Figura 104: Propiedades de los objetos: pestaña "Animación".....	100
Figura 105: Propiedades de los objetos: pestaña "Tipo de animación"	101
Figura 106: Propiedades de los objetos: pestaña "Esquema"	101
Figura 107: Variables "V1", "V2", "V3", "V4", "V5", "V5", "V6", "V7", "VA" y "VB"	102
Figura 108: Desactivación /activación Válvula 3 "V3"	103
Figura 110: Desactivación / activación Bomba 1 "B1"	104
Figura 111: Motor "M"	104

<i>Figura 112: Parada/puesta en marcha del motor</i>	105
<i>Figura 113: Tuberías de la instalación</i>	106
<i>Figura 115: Elección modo simulación</i>	107
<i>Figura 116: Establecer dirección</i>	107
<i>Figura 117: Conectar PLC</i>	108
<i>Figura 118: "Transferir proyecto a PLC"</i>	108
<i>Figura 119: Simulador minimizado</i>	108
<i>Figura 120: Simulador Unity Pro</i>	109
<i>Figura 121: Creación de Tabla de Animación</i>	110
<i>Figura 122: Selección "Botón de comando"</i>	111
<i>Figura 123: Propiedad "Botón de comando"</i>	111
<i>Figura 124: Habilitación variables de escritura</i>	112
<i>Figura 125: Aspecto de la Instalación en reposo</i>	112
<i>Figura 126: 33% Aditivo A</i>	113
<i>Figura 127: 66% Aditivo B</i>	113
<i>Figura 132: "Tabla de variables"</i>	115
<i>Figura 133: Señales que se pueden forzar</i>	116
<i>Figura 134: Habilitación "AC" y "TB"</i>	117
<i>Figura 143: Limpieza</i>	119
<i>Figura 144: Llenado "Depósito de agua"</i>	119
<i>Figura 146: Final de ciclo</i>	120

1. MOTIVACIÓN

**Simulación de la automatización de
procesos con Unity Pro**

1 MOTIVACIÓN

1.1 Motivación del Proyecto

El presente proyecto surge de la implementación de un nuevo software para los autómatas TSX Premium de Schneider del laboratorio de Automatización del departamento de Ingeniería de Sistemas y Automática. El funcionamiento de este software es muy similar a su predecesor, el PL7 Junior, pero tiene una nueva herramienta muy útil a la hora de trabajar sin el autómata. Esta herramienta es el simulador, y su estudio será el objeto de este proyecto fin de carrera. Este documento, o parte de él, pretende ser utilizado como manual de uso del simulador, ya que los alumnos no siempre tienen disponible el acceso a los laboratorios. El uso del simulador permite que el alumno pueda trabajar fuera de los laboratorios, en su casa, o en aulas informáticas, lo que facilita el estudio del manejo de los autómatas.

1.2 Objetivos

El objetivo de este proyecto es desarrollar un documento que sirva, en todo o en parte, como manual de uso del simulador del software Unity Pro a nivel de usuario. Este documento debe aportar toda la información necesaria para realizar una simulación completa de un proceso automatizado, desde la creación del proyecto, la programación, el diseño gráfico del sistema, hasta su simulación para observar el avance de los distintos estados del mismo. Dicho manual, irá orientado a todos los alumnos estudiantes de la asignatura de Automatización Industrial, y de todas las asignaturas en las que se utilice el PLC o autómata.

Para ello se deberán cumplir los siguientes sub-objetivos:

- Conocer la nueva herramienta de simulador y probar su funcionamiento.
- Analizar si cubre todas las necesidades de la asignatura, sustituyendo de manera absoluta al software anterior.
- Orientar el documento a un lenguaje docente de manera que los conceptos utilizados y los pasos a realizar queden completamente claros para los alumnos.

1.3 Estructura del documento

El documento está estructurado de la siguiente manera:

- El capítulo de la “Introducción”, que es el numerado como Capítulo 2, explica por qué la automatización es importante en el ámbito docente, además de presentar los autómatas y los medios de los que dispone la Universidad Carlos III para dar a conocer a sus alumnos este referente. Se define lo que es un simulador, sus funciones y aplicaciones. Para completar este capítulo, en él se presta atención al uso de los simuladores en la docencia y las ventajas que aportan.
- El capítulo siguiente, el Capítulo 3, es el llamado “Software Unity Pro”. En él se realiza un recorrido por el funcionamiento general del software. Se presenta una descripción del software y de sus características principales. Se hace un repaso de la estandarización de la programación del control industrial y del cumplimiento de la norma IEC 61131. A partir de esta norma se hace distinción de la programación que se considera lenguaje, tanto literal como gráfico, y de lo que simplemente es considerado una estructura organizacional. Esta aclaración es de gran importancia, porque no siempre ha sido igual debido a los cambios y fusiones que han surgido en los lenguajes de programación en la automatización industrial desde su nacimiento. El capítulo continúa con la presentación de las variables, direcciones y bits de palabras y sistemas disponibles en Unity Pro. A continuación, se describe el simulador y se detallan las opciones que ofrece tratar con un simulador. Para finalizar, se aclara la diferencia existente entre un sistema Scada y un simulador.
- El Capítulo 4 se llama “Ejemplo desarrollado”, donde se ha elegido un problema tipo de la automatización de un proceso, y se ha programado por completo con la herramienta de estudio. Tras la presentación del problema y la configuración del entorno de trabajo, se explica cómo se programa el ejemplo con dos soluciones diferentes.
- El Capítulo 5, “Simulación del ejemplo programado” se detalla la realización de una “pantalla del operador”. Todas las posibilidades de representación que existe en la librería y modo de programar cada tipo de elemento. Se expone un desarrollo de una simulación completa. Además de explicar cómo se realiza la compilación, la transferencia y el desarrollo del proyecto con el simulador de Unity Pro.
- En el Capítulo 6 se presentan las conclusiones y además posibles trabajos futuros relacionados con el tema de este proyecto.

2. INTRODUCCIÓN

**Simulación de la automatización de
procesos con Unity Pro**

2 INTRODUCCIÓN

Este capítulo introduce el término automatización, y se argumenta la importancia de la misma en el campo docente. Además se presenta un repaso breve al concepto y las funciones de los simuladores, su uso, ventajas y nuevamente, la aplicación de los mismos en la docencia.

2.1 La automatización y su importancia en la docencia

Según la Real Academia Española, la definición de automatización es aquel mecanismo que funciona en todo o en parte por sí solo. [1]

Para conocer de una manera más extensa el concepto de automatización, se va a definir como una disciplina de la ingeniería. La automatización es un concepto más amplio que un mero sistema de control. Abarca la instrumentación industrial, que incluye los sensores y transmisores de campo, los sistemas de supervisión, los sistemas de recolección de datos y las aplicaciones de software en tiempo real para controlar las operaciones de plantas o procesos industriales. [2]

Un autómata programable industrial (PLC) es un equipo electrónico de control con un cableado interno, denominado hardware, independiente del proceso a controlar, que se adapta a éste mediante un programa específico, software. Esta secuencia de operaciones se define sobre señales de entrada y salida al proceso, cableadas directamente en los bornes de conexión del autómata.

Las señales de entrada pueden proceder de elementos digitales, como finales de carrera y detectores de proximidad. O analógicos, como sensores de temperatura y dispositivos de salida en tensión o corriente. De la misma manera, las señales de salida pueden ser digitales, todo o nada, o señales analógicas, por ejemplo de tensión o corriente, que sirven a los elementos indicadores y actuadores del proceso.

El autómata gobierna las señales de salida según el programa de control previamente almacenado en una memoria, a partir del estado de las señales de entrada. Este programa se introduce en el autómata a través de la unidad de programación, que permite además funciones adicionales como depuración de programas, simulación, monitorización, control del autómata, etc.

Una característica diferenciadora del PLC frente a otros sistemas de control programables está en la estandarización de su hardware, que permite la configuración de sistemas de control adaptados a las necesidades estimadas de potencia de cálculo, y tipo de señales de entrada y salida. El autómata se configura alrededor de una unidad central o de control, que, unida por medio de buses internos a las interfaces de entrada y salida y a las memorias, define lo que se conoce como arquitectura interna del autómata.

Por este motivo, por sus especiales características de diseño, el PLC tiene un campo de aplicación muy extenso. La constante evolución del hardware y software amplía continuamente este campo para poder satisfacer las necesidades que se detectan en el espectro de sus posibilidades reales.

Su utilización se da fundamentalmente en aquellas instalaciones en donde es necesario un proceso de maniobra, control, señalización, etc. Por tanto, su aplicación abarca desde procesos de fabricación de cualquier tipo a transformaciones industriales hasta control de instalaciones.

El amplio concepto de automatización, hace realidad que en la docencia, los alumnos de Ingeniería Industrial, particularmente de la especialidad de Automática, tengan una asignatura dedicada al aprendizaje de su funcionamiento, pues va a ser un instrumento básico en sus horas de trabajo futuro.

El objetivo de las aulas de automatización y de los laboratorios donde los alumnos realizan prácticas en la Universidad Carlos III, es introducir al alumno al desarrollo de proyectos de automatización industriales, donde intervienen desde sistemas eléctricos cableados, pasando por control mediante PLC, programas SCADA y sistemas neumáticos.

El autómeta utilizado es el Premiun TSX de Scheneider, está compuesto de los siguientes elementos:

- Bastidor
- Fuente de alimentación
- Procesador
- Módulo de entradas digitales
- Módulo de salidas digitales
- Módulo de entradas analógicas
- Módulo de salidas analógicas
- Telefast
- Cable comunicación con PC

La configuración real variará en función al número de tarjetas que sean necesarias según los requisitos de los sensores y las comunicaciones. Una imagen real de dicho autómeta es la que aparece en la figura 1. [3]



Figura 1: Autómeta del laboratorio de Automatización de UC3M

2.2 Simulador: concepto y función

Un simulador es un sistema que reproduce el comportamiento de una máquina en ciertas condiciones. Permite que la persona que debe manejar dicho sistema pueda practicar sin necesidad de utilizar dicha máquina. Los simuladores, pueden encontrarse en el ámbito profesional o como un instrumento de ocio y entretenimiento. En el primer caso, estos dispositivos aparecen como indispensables para la formación de personas que tengan una gran responsabilidad a su cargo. Un simulador permite entrenamiento hasta adquirir la experiencia y la destreza necesarias para desempeñarse profesionalmente. [4]

Todo simulador tiene los siguientes atributos:

- Imita la realidad: la imitación de la realidad es lo que distingue un simulador de un juego.
- No es real en sí mismo: un simulador se usa para practicar algo, no sustituye experiencias reales. Rose Tremain escribió que "la vida no es un ensayo general". Si se comete un error durante un ensayo general no hay coste, solo se repite hasta que salga bien. Un simulador es una simplificación de la vida real. Los simuladores sencillos son, normalmente, mejores que los complicados. Los simuladores complejos pueden hacer que el usuario se pierda en los detalles y no pueda extraer el aprendizaje o las lecciones valiosas del simulador. [5]
- Puede ser cambiado por sus usuarios: las simulaciones son activas, fuerzan a los usuarios pasivos hacia un aprendizaje activo.

Actualmente, el modelo de simulaciones asistidas por ordenador está siendo utilizado con éxito en diversos sectores para el desarrollo de una amplia gama de competencias. Una simulación parte de una reconstrucción de modelos de actuación reales y permite tomar decisiones relacionadas con dicho modelo, minimizando el riesgo de cometer errores. De esta forma, el usuario aprende por la experiencia, con una base eminentemente práctica.

La formación basada en la simulación permite aprender haciendo, o lo que es lo mismo, tomando decisiones en escenarios reales. Es lo que se conoce como "learn by doing" es decir, aprender experimentando situaciones que parecen reales. Este tipo de aprendizaje facilita esa adhesión o retención de la información y permite aprender más rápido a la vez que facilita el desarrollo de una mayor intuición a la hora de tomar decisiones reales, tal y como se ha comentado con anterioridad [6].

La motivación en este tipo de aprendizaje está basada en los simuladores. El aprendizaje a través de las simulaciones es auto conducido. Un usuario que se sienta delante de un ordenador y comienza a conducir una simulación irá a través de un círculo de aprendizaje: reflexionar sobre el caso, elegir la estrategia, tomar decisiones y observar las consecuencias de esas decisiones. Sin este auto esfuerzo el simulador se convertirá en un simple juego.

Un simulador permite el desarrollo de un sistema. En realidad, reproducen sensaciones que no están sucediendo. Pretende reproducir tanto las sensaciones físicas (velocidad, aceleración, percepción del entorno) como el comportamiento de los equipos de la máquina que se

pretende simular. Para simular las sensaciones físicas se recurre a complejos mecanismos comandados por potentes ordenadores que mediante modelos matemáticos consiguen reproducir sensaciones de velocidad y aceleración entre otros.

2.3 Uso de los simuladores

Durante años, la simulación asistida por ordenador ha jugado un papel muy significativo en los programas de formación de importantes sectores de la economía como la aviación o la industria militar.

De hecho, los primeros simuladores surgen en la década de los 60 con el objetivo de reducir el nivel de error humano en los vuelos comerciales. Desde entonces el entrenamiento de los pilotos es impensable sin un simulador.

Las barreras técnicas y los costes de desarrollo, han sido los dos principales factores que, hasta ahora, han frenado la expansión de los simuladores como herramienta de aprendizaje. Pero las nuevas tecnologías, omnipresentes en la formación a través de Internet, así como la creación de herramientas de autor específicas para simulaciones, están abaratando significativamente los procesos de producción.

Así, podemos afirmar que a corto o medio plazo asistiremos al auge de los simuladores como herramienta de formación. [8]

Entre los muchos ejemplos, cabe destacar:

- Simulador de conducción: permiten a los alumnos de autoescuela, enfrentarse con mayor seguridad a las primeras clases prácticas.
- Simulador de carreras, con los que se puede conducir un automóvil, motocicleta, camión, etc.
- Simulador de vuelo o de aviones: permite dominar el mundo de la aviación y pilotar aviones, helicópteros...
- Simulador de trenes: permite controlar un tren.
- Simulador de negocio: permite simular un entorno empresarial.
- Simulador clínico médico: permite realizar diagnósticos clínicos sobre pacientes virtuales.
- Simulador musical: permite reproducir sonidos con un instrumento de juguete.
- Simulador termo-solar: permite analizar la influencia de la producción de electricidad en la modificación de ciertos parámetros en una central solar termoeléctrica.

2.4 Aplicación de los simuladores en la docencia

Los simuladores educativos son programas que permiten hacer prácticas simuladas, pero con las mismas características que en la realidad. Permiten ahorrar costes y reducir riesgos pero manteniendo la calidad en la formación.

En el campo de la docencia, las simulaciones hacen que los estudiantes sean responsables de su propio aprendizaje y su motivación consiste en la consecución de objetivos. Se pueden dar varios grados de implicación por parte de profesor y alumnos:

- Utilización por parte del profesor para ilustrar un procedimiento o proceso concreto.
- Utilización por parte del alumno sin guía del profesor, para tratar de descubrir cómo afectan distintas variables a un procedimiento o proceso.
- Utilización supervisada o guiada por el profesor, con el fin de que el alumno adquiera el suficiente dominio y comprensión de procedimientos y procesos.

Además, lleva como valor añadido los propios objetivos de la automatización, mejorar la calidad, productividad y seguridad, así como la reducción de costes en la producción. Simplifica las operaciones con la integración de gestión y producción.

La utilización de un simulador en el marco del constructivismo en la educación, para promover el desarrollo de habilidades y actitudes con los alumnos en clase, permite ampliar las técnicas y herramientas tecnológicas para un mayor aprendizaje [9].

Entre los retos a resolver en este siglo en la educación, destaca el gran reto de desarrollo e implementación de estrategias de enseñanza. Conseguir métodos dirigidos a desarrollar habilidades y actitudes en los estudiantes, para que adquieran distinciones y competencias. Éstas permiten estar funcionalmente activos en lo profesional y ser capaces de tomar decisiones que les lleven a resolver genuinamente los problemas. Además, demanda del alumno el ejercicio de compartir los recursos y conocimientos de que dispongan, a través de la práctica de aprendizajes colaborativos.

La función de los simuladores está basada en aprendizaje de tipo experimental, y en conjeturas, para llevar a cabo un aprendizaje basado en el descubrimiento. El alumno obtiene conocimiento por medio de la interacción con un mundo simulado logrando así controlar una situación del mundo real, mediante el cual se logrará observar diferentes situaciones, y aprenderá a tomar las decisiones.

La principal característica de los simuladores no es la de alcanzar la realidad, sino la de proveer situaciones llamativas y entretenidas para el usuario, con lo que se logra el aprendizaje de un determinado contexto [10].

2.5 Ventajas de los simuladores docentes

Las principales ventajas que ofrecen los simuladores docentes, son los mencionados a continuación:

- El estudiante es la fuente principal de aprendizaje, actuando de manera activa, debe resolver la situación simulada que se le presenta: procesar la información, tomar decisiones, y obtener los resultados.
- El problema llega a ser un reto para el alumno, con lo que el material logra la fase de motivación.
- Se logra cumplir con los requerimientos de los sistemas educativos, en cuanto a practicar y afinar lo aprendido en casa. Estas restricciones de operación hacen necesario que el alumno realice un esfuerzo para lograr las metas propuestas.
- Permite manipular al alumno un número reducido de variables dentro del proceso simulado, acotando y enfocando la situación al objetivo de la asignatura.
- Como ejemplo destacable, la consultora McKinsey en un artículo titulado “Is Simulation Better Than Experience?” [7] expone que las simulaciones pueden ser mejores que la experiencia real ya que comprimen el tiempo en lo rigurosamente importante y elimina detalles irrelevantes. Destaca que la utilización de un sistema simulador asegura un aprendizaje automático, y que éstos son creados para la asimilación de situaciones.

Una de las claves de éxito de un simulador docente es el realismo que genera, creando una experiencia más divertida, clara y educativa.

Uno de los principales problemas asociado al uso de un laboratorio tradicional en la Universidad, radica en la limitación de recursos de personas, tiempo y espacios restringidos, principalmente debido a la masificación de la educación superior y a los inherentes problemas presupuestarios asociados. La situación tradicional requiere la presencia física del estudiante y la supervisión del profesor. El inconveniente de no disponer de suficientes autómatas y laboratorios, muchos de ellos costosos, se transforma en el problema de encontrar software adecuado para simularlos. Contar con suficientes recursos informáticos para alcanzar el aprovechamiento del alumno y así elaborar prácticas con simuladores es el siguiente paso en que hay que trabajar.

3. SOFTWARE UNITY PRO SCHNEIDER

**Simulación de la automatización de
procesos con Unity Pro**

3 SOFTWARE UNITY PRO - SCHNEIDER

En este capítulo se representará brevemente el funcionamiento general del software Unity Pro de la versión 5.0 Se desarrollará su descripción y sus características principales. Se hace un repaso de la estandarización de la programación del control industrial y en el cumplimiento de la norma IEC 61131. También se comentará todas las posibilidades que ofrece la utilización de las variables, direcciones y bits de palabras y sistemas.

3.1 Descripción y características

El software de programación Unity Pro XL, es un software de última generación y de reciente implantación en el campo de la automatización industrial. Ante tal eventualidad, se va a entrar en detalles sobre su modo de operación en el capítulo 4.

Dentro del software Unity Pro XL podemos generar proyectos en varias familias de autómatas, como son Modicom M340, Premium, Quantum y Atrium. Todos ellos dentro de la marca Schneider Electric.

Las características del software industrial son las mencionadas a continuación:

- Está basado en formatos estándares, reduciendo sensiblemente las actividades de familiarización y capacitación.
- Presenta las herramientas necesarias para la creación, depuración y puesta en marcha de las aplicaciones.
- El entorno de ejecución de los programas es Windows 98/2000/NT/XP/7, estando adaptado a su funcionamiento gráfico y orientado a objetos.
- Su utilización es intuitiva.

Unity Pro aprovecha al máximo las ventajas de los interfaces gráficos y contextuales de Windows XP y Windows 2000:

- Acceso directo a las herramientas y a los datos.
- Configuración 100% gráfica.
- Barra de herramientas e iconos personalizables.
- Funciones avanzadas de “arrastrar y soltar” y zoom.
- Ventana de diagnóstico integrado

La principal ventaja de este software de programación es la estandarización de sus códigos gracias al cumplimiento de la norma IEC 61131-3, con la cual se consigue que Unity Pro proponga un conjunto completo de funcionalidades y de herramientas que permiten calcar la aplicación en la estructura del proceso o de la máquina.

3.2 Estandarización en la programación de control industrial

La norma IEC 61131-3 persigue la estandarización de los lenguajes en la programación del control industrial. [11] En la actualidad aún siguen persistiendo sistemas de control específicos del fabricante, con programación dependiente y conexión compleja entre distintos sistemas de control. Esto significa para el usuario costos elevados, escasa flexibilidad y falta de normalización en las soluciones al control industrial. Las implicaciones técnicas de la norma son altas, dejando bastante espacio para el crecimiento y la diferenciación. El uso de la norma proporciona numerosos beneficios para usuarios/programadores. Algunos de estos beneficios son:

- Se reduce el gasto en recursos humanos, formación, mantenimiento y consultoría.
- Evita las fuentes habituales de problemas por el alto nivel de flexibilidad y reusabilidad del software.
- Las técnicas de programación son utilizables en amplios sectores (control industrial en general).
- Combinan adecuadamente diferentes elementos que pueden provenir de diferentes fabricantes, programas, proyectos...
- Incrementa la conectividad y comunicación entre los distintos departamentos y compañías.

Bajo esta norma, se definen cuatro lenguajes de programación normalizados. Esto significa que su sintaxis y semántica ha sido definida, no permitiendo particularidades distintivas (dialectos).

Los lenguajes consisten en dos de tipo literal y dos de tipo gráfico:

Literales:

- Lista de instrucciones ("*Instruction List*", "*IL*").
- Texto estructurado ("*Structured Text*", "*ST*").

Gráficos:

- Diagrama de contactos ("*Diagram Ladder*", "*LD*").
- Diagrama de bloques funcionales ("*Function Block Diagram*", "*FBD*").

En la Figura 2, se muestran cuatro programas que describen la misma acción. El uso de uno u otro depende de los conocimientos del programador, el problema a tratar, el nivel de descripción del proceso o la estructura del sistema de control. Los cuatro lenguajes están interrelacionados y permiten su empleo para resolver conjuntamente un problema común según la experiencia del usuario.

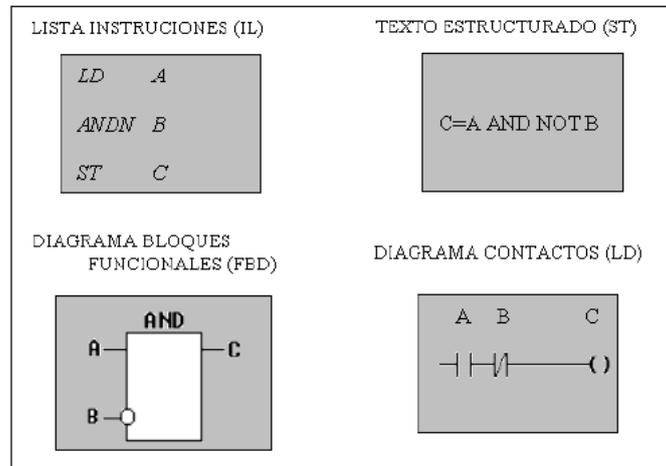


Figura 2: Lenguajes IEC-1131-3

- **“Lista de Instrucciones (IL)”** es el modelo de lenguaje ensamblador basado en un acumulador simple; procede del alemán *Anweisungsliste*, *AWL*.
- **“El lenguaje Texto estructurado (ST)”** es un lenguaje de alto nivel con orígenes en ADA, Pascal, y ‘C’; puede ser utilizado para codificar expresiones complejas e instrucciones anidadas; este lenguaje dispone de estructuras para bucles (*REPEAT-UNTIL*; *WHILE-DO*), ejecución condicional (*IF-THEN-ELSE*; *CASE*), funciones (*SQRT*, *SIN*, etc.)
- **“El Diagrama de contactos (LD)”** está basado en la presentación gráfica de la lógica de relés.
- **“El Diagramas de Bloques Funcionales (FBD)”** es muy común en aplicaciones que implican flujo de información o datos entre componentes de control. Las funciones y bloques funcionales aparecen como circuitos integrados y es ampliamente utilizado en Europa.

La norma también permite dos formas de desarrollar el programa de control: de arriba a abajo (*Top-down*) y de abajo a arriba (*bottom-up*). Se puede especificar inicialmente la aplicación completa y dividirla en partes, declarar las variables, etc. También se puede comenzar la programación desde abajo, por ejemplo, por medio de funciones y bloque funcionales.

En los lenguajes de programación *LD*, *ST*, *FBD* e *IL* es posible insertar bloques de función.

Centrando la atención en los lenguajes utilizados para la realización del proyecto, el bloque función aparece en formato de bloque en el lenguaje gráfico *LD* y en formato texto/estructura en el lenguaje de tipo de texto *ST*. Los bloques de función vienen de un conjunto de librerías y dentro de cada librería se encuentran familias.

En *LD*, un bloque función se representa como una trama de bloques con entradas y una salida. Las entradas siempre aparecen a la izquierda de la trama y las salidas a la derecha. El nombre de la función, por ejemplo el tipo de función, se muestra en el centro de la trama. En *ST*, un bloque función se representa como una estructura de datos con el nombre de la instancia y entre paréntesis sus entradas y sus salidas. La Figura 3 muestra una representación de cada uno de estos lenguajes:

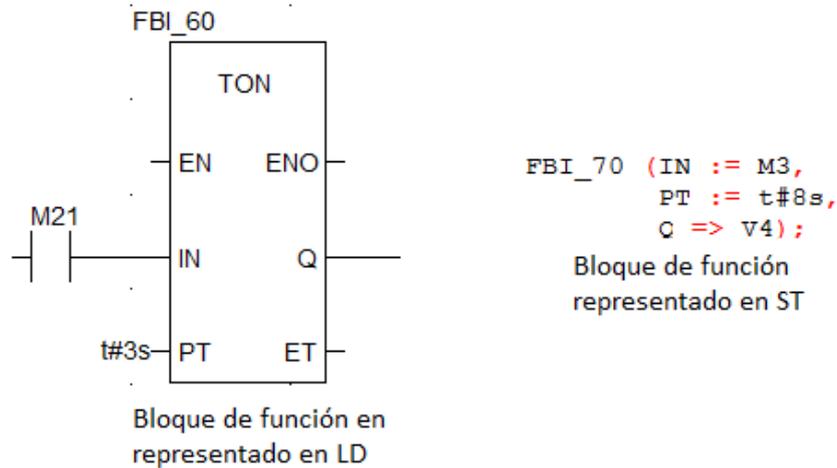


Figura 3: Representación bloque de función en LD y ST

Adicionalmente, el estándar *IEC 1131-3* incluye una forma de programación orientada a objetos llamada *Sequential Function Chart (SFC)*. *SFC* es a menudo categorizado como un lenguaje *IEC 1131-3*, pero éste es realmente una estructura organizacional que coordina los cuatro lenguajes estándares de programación (*LD*, *FBD*, *IL* y *ST*). La estructura del *SFC* tuvo sus raíces en el primer estándar francés de *Grafcet (IEC 848)*. Es un "lenguaje" gráfico que provee una representación diagramática de secuencias de control en un programa. Básicamente, *SFC* es similar a un diagrama de flujo, en el que se puede organizar los subprogramas o subrutinas (programadas en *LD*, *FBD*, *IL* y/o *ST*) que forman el programa de control. *SFC* es particularmente útil para operaciones de control secuencial, donde un programa fluye de un punto a otro una vez que una condición ha sido satisfecha (cierta o falsa).

El marco de programación de *SFC* contiene dos principales elementos que organizan el programa de control:

- Pasos (etapas)
- Transiciones (condiciones)

El programa irá activando cada una de las etapas y desactivando la anterior conforme se vayan cumpliendo cada una de las condiciones. Las acciones se realizarán en función de la etapa activa a la que están asociadas. Por ejemplo, en la Figura 4, la etapa 1 activa tras arrancar el programa, al cumplirse la "Condición 1", se activará la etapa 2, se desactivará la 1, y se realizará la "Acción 1":

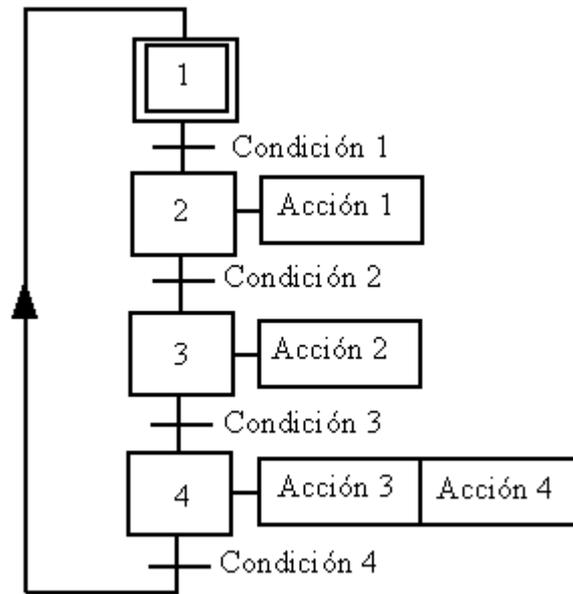


Figura 4: Representación diagrama SFC

3.3 Tipos de variables, direcciones, bits y palabras de sistemas

3.3.1 Variables

Los tipos de variable que se pueden definir en Unity Pro son diversos. Es necesario definir las variables tanto de entrada como de salida. Las posibilidades de variables más destacadas entre las que se puede elegir son las siguientes:

- **BOOL**: Sigla del tipo Booleano. Se trata del elemento de datos de base informática. Una variable de tipo BOOL posee uno de estos valores: 0 (FALSE) o 1 (TRUE). Un Bit extraído de la palabra es de tipo BOOL, por ejemplo: %MW10.4.
- **EBOOL**: Sigla del tipo Extended BOOLean (booleano extendido). Una variable de tipo EBOOL posee el valor 0 (FALSE) o 1 (TRUE), pero igualmente los flancos ascendentes o descendentes y las funciones de forzado. Una variable de tipo EBOOL ocupa un byte de memoria.
- **INT**: Sigla del formato single INTeger (entero simple) (codificado en 26 bits). Los límites inferior y superior figuran a continuación: de -2^{15} a 2^{15} .

3.3.2 Direcciones

Cuáles son los tipos y características de cada una de las direcciones son las que se van a enumerar a continuación. De esta manera se puede llegar a implementar mejor el desarrollo del programa de forma que se pueda reducir el tiempo de ciclo y por tanto optimizar el rendimiento de la instalación.

- %I: Indica un objeto de lenguaje de entrada binario.
- %W: Indica un objeto de lenguaje de entrada analógico.
- %KW: Indica un objeto de lenguaje de palabra constante.
- %M: Indica un objeto de lenguaje de bit de memoria.
- %MW: Indica un objeto de lenguaje de palabra de memoria.
- %Q: Indica un objeto de lenguaje de salida binaria.
- %QW: Indica un objeto de lenguaje de salida analógica.

Lo primero que cabe destacar es la nomenclatura a seguir para definir las direcciones. Todas empiezan por el símbolo % seguidas de una letra inicial según el tipo de dirección que se vaya a gestionar. Todos los tipos de variables están definidas según la norma IEC 61131-3, como anteriormente se ha detallado.

Se puede observar que el grupo de tipos de variables se distinguen básicamente en dos según el área de memoria que necesita el autómatas para procesarla. Por una parte están las direcciones puramente digitales cuyos valor es binario, solo puede tener dos posiciones 1 ó 0. Y por otro lado están las palabras cuya dimensión es de 16 bits, las cuales pueden ser por ejemplo una entrada analógica para expresa un valor numérico.

3.3.3 Bits y palabras de sistema

Existen unos bits y palabras del sistema que son de gran utilidad e importancia a la hora tanto de programar como de depurar la programación en busca de errores y estados del autómatas. Los autómatas Modicon M340, Premium, Atrium y Quantum utilizan bits de sistema %S_i que indican los estados del autómatas o que permiten controlar el funcionamiento de éste. Dichos bits pueden probarse en el programa del usuario con el fin de detectar cualquier evolución de funcionamiento que conlleve un procedimiento de procesamiento establecido. Algunos de estos bits deben volver a su estado inicial o normal por programa. No obstante, los bits de sistema que vuelven a su estado inicial o normal a través del sistema, no deben hacerlo a través del programa ni del terminal.

Los bits de sistema, funcionan como un diagnóstico de errores. Entre otros, el bit de arranque en frío %S0, está normalmente en 0. Este bit se define a 1 durante el primer ciclo completo de restauración del PLC en modalidad RUN o STOP, pero no en caso de modo simulación. Es el botón "Reinicializar" del simulador, el cual puede restablecer el PLC simulado; exactamente equivale a un arranque en frío del PLC, se interrumpe la conexión entre Unity Pro y el

simulador, se reinician todas las variables del proyecto y el simulador pasa al estado RUN (si el arranque automático está activado) o STOP (si el arranque automático está desactivado). En realidad, el botón Reiniciar hace el papel del botón de restablecimiento de una CPU real. Este es uno de los bits de sistema que encuentran su función limitada al funcionar con el simulador, y es una clara desventaja frente al uso de PLC.

De entre todos los bits de sistema, uno de los más prácticos y usado por su potente funcionalidad es el *"watchdog"*. Es un concepto de protección usado para volver a reiniciar el programa cuando éste "se pierde" o realiza una acción no prevista. Su funcionamiento es exactamente igual al de un microprocesador: cuenta cada cierto pulso de reloj en un determinado tiempo esperando algún evento generado por el programa. Si no le llega tal, el watchdog se activa y hace que todo empiece de nuevo. Si le llega el evento, no hace nada porque todo funciona como se espera.

3.4 El simulador de Unity Pro y comparativa con PLC

3.4.1 Simulador

Una novedad dentro del software de programación de Schneider Electric es un simulador del autómatas integrado. Reproduce fielmente el comportamiento del programa en el PC. Gran parte de las herramientas de puesta a punto se pueden utilizar en simulación para aumentar la calidad antes de la instalación. Por tanto, gracias a este simulador se puede llevar el código pre-depurado a la instalación, con lo cual se reduce el tiempo de puesta en marcha del proceso. Entre las funciones que se pueden desarrollar con el simulador, destaca:

- Ejecución del programa paso a paso.
- Punto de parada y de visualización
- Animaciones dinámicas para visualizar el estado de las variables y la lógica que se está ejecutando.

3.4.2 Simulador Vs PLC

Si se compara el simulador con un PLC real, se observan algunas diferencias descritas a continuación:

- El simulador del PLC simula un proyecto completo con todas sus tareas de usuario. No obstante, el comportamiento de tiempo de ejecución del simulador no es equiparable al comportamiento de tiempo de ejecución de un PLC real.
- El simulador de PLC no admite ninguna forma de Entrada/Salida (E/S). Aunque la simulación contiene los componentes del proyecto para E/S, el simulador de PLC no los procesa. A las entradas y salidas sólo se puede acceder desde el proyecto o a través de las funciones online de Unity Pro (leer, escribir, forzar, animar, etc.).

- El simulador no soporta algunos bits de sistema como el PLC. Entre otros, el bit de arranque en frío %S0, está normalmente en 0. Este bit, como se ha comentado anteriormente, se define a 1 durante el primer ciclo completo de restauración del PLC en modalidad RUN o STOP, pero no en caso de modo simulación.
- En cuanto a los servicios del SO del PLC, el simulador de PLC admite la mayoría de los servicios de sistemas operativos de PLC en diversas plataformas. Dichos servicios sólo se aplican como servicios ficticios. Esto significa que pueden utilizarse funciones y bloques de funciones en el proyecto cargado, pero no funcionan como deberían o devuelven un mensaje de error. Esto afecta principalmente a las funciones y bloques de funciones que accedieron a plataformas especiales, como bloques de E/S y funciones específicas de hardware y comunicación.

3.5 Diferencias entre un sistema Scada y un simulador

Un sistema Scada (Supervisory Control And Data Acquisition) se define como un sistema, basado en computadores, que permite supervisar y controlar variables de proceso a distancia, proporcionando comunicación con los dispositivos de campo. Además, controla el proceso de forma automática por medio de un software especializado. A diferencia de un simulador que reproduce un sistema que no está ocurriendo en tiempo real, el sistema Scada provee de toda la información que se genera en el proceso a diversos usuarios, tanto del mismo nivel como de otros usuarios supervisores dentro del sistema.

La mayor diferencia entre ambos, es que en el simulador se simulan las variables configuradas y durante la simulación, es posible manipular, activar y desactivar dichas variables. En el Scada se puede supervisar las tareas de otros individuos, aunque esto último no significa que tenga el control sobre ellos, sino que el objetivo principal es correctivo. Para que dicha modificación sea posible, además, el sistema Scada puede tener realimentación. El proceso de compartir observaciones, preocupaciones y sugerencias, con la intención de recabar información, a nivel individual o colectivo, puede mejorar o modificar diversos aspectos del funcionamiento de una organización. La realimentación tiene que ser bidireccional de modo que la mejora continua sea posible.

La manera de integrar un sistema Scada en Unity Pro, es comunicando un sistema de control mediante redes o buses. Se trata de conseguir una aplicación gráfica que reproduzca en la pantalla el proceso que se va a automatizar, distinta de la que puede ofrecer el simulador mediante la herramienta "Pantalla del operador" como se verá más adelante. Una manera posible es mediante "Entorno Vijeo Citect". [12]

En el caso que se quisiera unir el PLC con el Scada, sería necesaria previamente configurar el HW necesario que realizará la configuración de la red Ethernet a la que estará conectada el PC o PLC. Por lo tanto habría que realizar la definición de la red Ethernet, así como su asignación al puerto Ethernet de la CPU.

4. EJEMPLO DESARROLLADO: PROGRAMACIÓN

**Simulación de la automatización de
procesos con Unity Pro**

4 EJEMPLO DESARROLLADO

En este capítulo se va a mostrar un problema práctico de cómo simular la automatización de un sistema físico propuesto. En primer lugar, este ejemplo se programará con un diagrama SFC en lenguaje LD, y a continuación, directamente en lenguaje ST.

Para conseguir un diagrama gráfico SFC, se programará las etapas y transiciones. Si la complicación de una etapa es elevada, se recurre a una herramienta disponible en Unity Pro denominada macroetapa. Cada una de estas etapas serán programadas en lenguaje LD. Para mostrar las posibilidades ofertadas por el software, se programará el mismo sistema en lenguaje ST.

4.1 Presentación del problema

Se trata de automatizar un proceso en el que se producen dos tipos de compuestos a partir de dos aditivos. El proceso consta de unos depósitos de aditivos, un depósito de mezcla, un cambiador de calor, un tanque para almacenar el compuesto A, un tanque para almacenar el compuesto B y un depósito de agua caliente. El proceso se va a controlar con un autómatas que tiene entradas y salidas en un segundo bastidor. Debe controlarse tanto el preparado del compuesto pedido como el mantenimiento de la temperatura del depósito de agua caliente. La instalación completa se representa en la Figura 5:

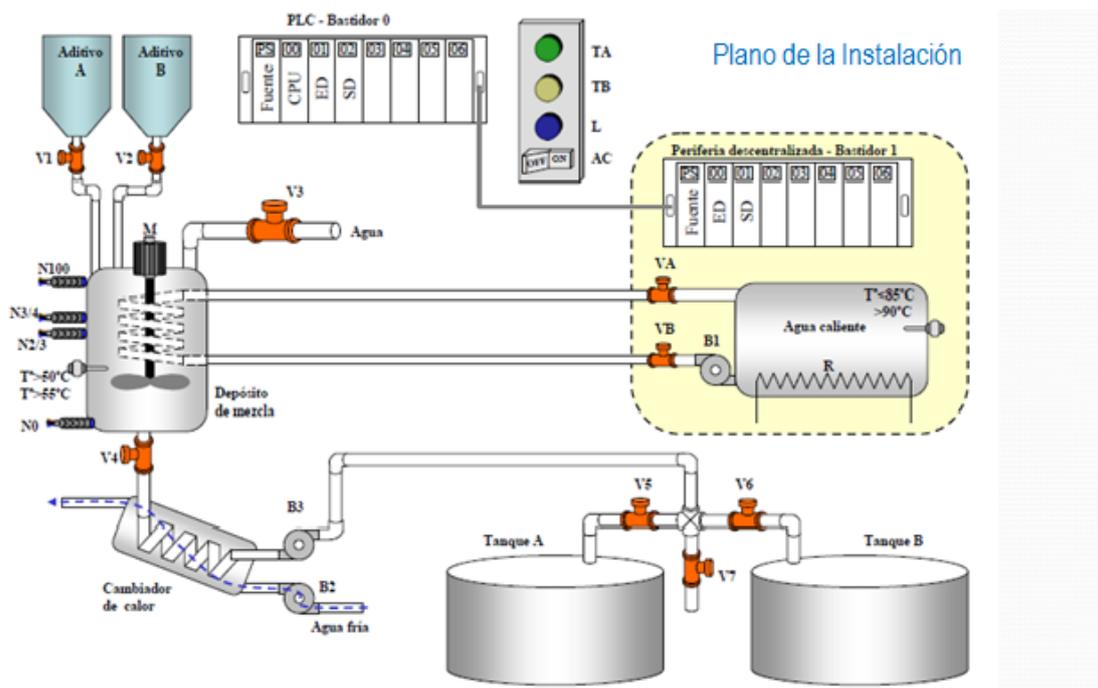


Figura 5: Plano de la instalación

El proceso se basa en cinco etapas, descritas a continuación:

OPERACIÓN:

- Si se pulsa el botón TA, se debe producir una mezcla con $1/3$ de aditivo A y $2/3$ de aditivo B y almacenarse en el tanque A.
- Si se pulsa el botón TB, se debe producir una mezcla con $3/4$ de aditivo de A y $1/4$ de aditivo de B y almacenarse en el tanque B.
- Si se pulsa el botón L, se procede a limpiar el depósito de mezcla

DOSIFICACIÓN:

Ayudándose de los sensores de nivel que indican que el depósito está lleno según los porcentajes en que se ajusta la composición de la mezcla:

- N0: Vacío
- N $_{1/3}$: $1/3$ de su ocupación
- N $_{3/4}$: $3/4$ de su ocupación
- N100: depósito lleno

MEZCLA:

- En el depósito de mezcla, después de añadirse los aditivos deben calentarse la mezcla hasta 50°C en el caso de Mezcla A y 55°C en el caso de Mezcla B.
- En el depósito de mezcla se calienta mediante agua caliente. El agua se mantiene caliente en un depósito aparte, y debe controlarse con el autómata que se mantenga entre 85°C y 90°C . La mezcla se agitará durante 1 minuto.

LLENADO DE TANQUES:

- Mediante la activación de las correspondientes válvulas y bombas, se deposita la mezcla en un tanque correspondiente.
- Después de realizar la mezcla, se realiza un proceso de limpiado del depósito de mezcla.

LIMPIEZA:

La limpieza consiste en llenar el depósito de agua mediante la válvula V3, agitar la mezcla durante 1 minuto y vaciar el depósito.

4.2 Configuración del entorno de trabajo y definición de variables necesarias

Lo primero que hay que hacer para comenzar a programar una vez planteado el problema es especificar con qué modelo de autómatas se va a trabajar. Dentro de las tres familias de PLC's que posee la gama de Schneider Electric, se va a seleccionar la Premium. Ha sido elegida porque es la disponible en el laboratorio de Automatización Industrial.

Una vez abierto el programa Unity Pro, se crea un nuevo proyecto. La manera de crear un nuevo proyecto es en el menú Fichero, seleccionar "Nuevo". Aparecerá un cuadro como el que se muestra en la Figura 6. En él se elegirá la familia seleccionada, el "modelo TSX P57 104M 0260 57-1":

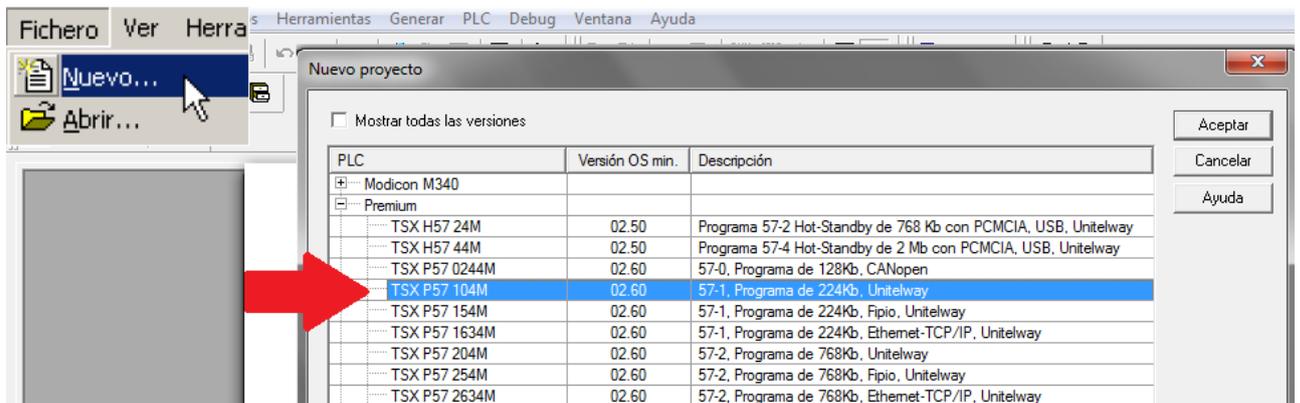


Figura 6: Elección CPU en Unity Pro

De esta manera se genera automáticamente el entorno de trabajo. Su aspecto es el mostrado en la Figura 7:

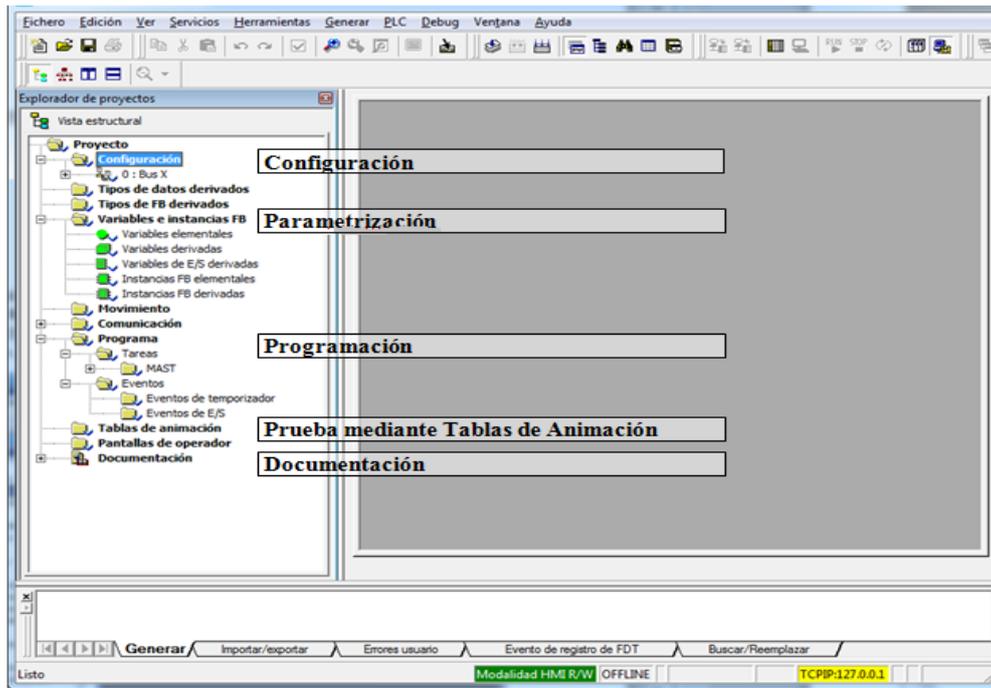


Figura 7: Explorador de Proyectos en Unity Pro

Dentro del Explorador de Proyectos, mostrado en la Figura 7, el primer paso es definir el Hardware con el que se va a trabajar. Se necesita instalar los diferentes módulos en el Rack donde está colocada la CPU y una Fuente de Alimentación instalada automáticamente por el programa con un Rack.

Se puede observar en la Figura 8, que el Rack está dividido en varios Slots. La primera posición del Rack (la posición cero) estará ocupada por la CPU y a continuación se disponen las tarjetas necesarias. Este orden siempre debe ser seguido en la configuración del hardware, ya que si no se tiene en cuenta al compilar la aplicación, podría ocurrir un error.

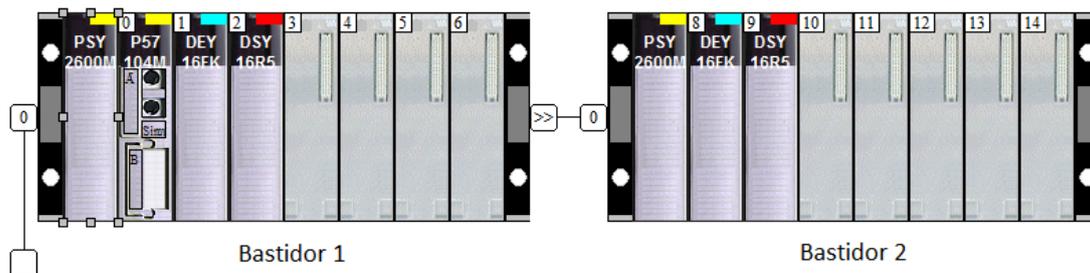


Figura 8: Configuración del Rack

El primero de los bloques de cada uno de los dos bastidores que contiene el ejemplo de la Figura 8, es el asignado con las siglas “PSY” corresponde a la Fuente de Alimentación de 100/240 VCA 26W. El bloque denominado “P57” es el procesador CPU y sólo se encuentra en el bastidor 1, situado a la izquierda. Las entradas digitales están contenidas en el bloque “DEY”, y las salidas digitales en el bloque “DSY”, necesarias en ambos bastidores.

Se puede modificar esta configuración inicial en todos los aspectos. Añadiendo un Rack mayor, un segundo Rack, módulos binarios, analógicos, conteo o de comunicación.

De entre todas las variables creadas (Figura 12), se puede hacer una clasificación especial según el uso que se les dé:

- Hay algunas que son utilizadas para la programación del sistema, y por tanto también se aprovecharán para la simulación. Dichas variables corresponden con las entradas del sistema (Figura 9) y las salidas del sistema (Figura 10).
- Sin embargo, hay algunas variables que exclusivamente han sido creadas para la simulación, y que más adelante se detallará su origen. Dichas variables son mostradas más adelante (Figura 17), y no se utilizan en código de la programación.

Las variables de entrada son las que muestran la Figura 9:

Nombre	\	T...	Comentario
AC	EBOOL		Alimentación
TA	EBOOL		Mezcla Compuesto A
TB	EBOOL		Mezcla compuesto B
L	EBOOL		Limpieza
N2_3	EBOOL		Sensor 2/3% Depósito de mezcla
N3_4	EBOOL		Sensor 3/4% Depósito de mezcla
N_0	EBOOL		Sensor 0% Depósito de mezcla
N_100	EBOOL		Sensor 100% Depósito de mezcla
T_50	EBOOL		Sensor 50°C Depósito de mezcla
T_55	EBOOL		Sensor 55°C Depósito de mezcla
T_85	EBOOL		Sensor agua caliente 85°C
T_90	EBOOL		Sensor agua caliente 90°C

Figura 9: Variables de entrada

Las variables de entrada del calentador, que son las T_85 y las T_90, mostradas en color negro en la Figura 9, se han considerado en un segundo bastidor para crearlo en un sistema independiente.

Y las variables de salida son las detalladas en la Figura 10:

Nombre	\	T...	Comentario
V1	EBOOL		Válvula Aditivo A
V2	EBOOL		Válvula Aditivo B
V3	EBOOL		Válvula agua Depósito mezcla
V4	EBOOL		Válvula cambiador de calor Depósito mezcla
V5	EBOOL		Válvula Tanque A
V6	EBOOL		Válvula Tanque B
V7	EBOOL		Salida
B2	EBOOL		Agua fría
B3	EBOOL		Mezcla A o B
Motor	EBOOL		Motor
VA	EBOOL		Válvula retomo agua del calentador
VB	EBOOL		Válvula salida agua del calentador
B1	EBOOL		Bomba del calentador
R	EBOOL		Resistencia

Figura 10: Variables de salida

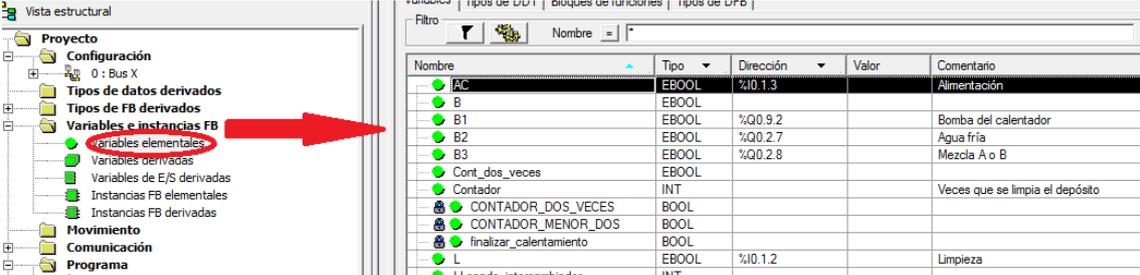
Con el fin de conseguir un control independiente, las salidas del calentador correspondientes al segundo bastidor son mostradas en color negro en la Figura 10.

Una vez contempladas las variables, hay de definir las. La manera de realizarlo, es escribiendo el nombre de cada variable, sin dejar espacios, e ir añadiendo los siguientes parámetros:

- Nombre
- Tipo
- Dirección
- Valor
- Comentario

Dichas variables se crean dentro del Explorador de Proyectos, en el apartado de “Variables e instancias FB” y una vez aquí, en variables elementales. El sistema permite hacer distinción entre “variables derivada”, “variables derivadas de E/S”, “Instancias FB elementales” e “Instancias FB derivadas”.

La tabla de variables elementales queda definida de la manera que se observa en la Figura 11:



The screenshot shows the 'Vista estructural' (Structural View) of a project. In the left pane, the 'Variables e instancias FB' folder is expanded, and the 'Variables elementales' sub-folder is highlighted with a red circle and a red arrow pointing to the table on the right. The table lists the following variables:

Nombre	Tipo	Dirección	Valor	Comentario
fAC	EBOOL	%I0.1.3		Alimentación
B	EBOOL			
B1	EBOOL	%Q0.9.2		Bomba del calentador
B2	EBOOL	%Q0.2.7		Agua fría
B3	EBOOL	%Q0.2.8		Mezcla A o B
Cont_dos_veces	EBOOL			
Contador	INT			Veces que se limpia el depósito
CONTADOR_DOS_VECES	BOOL			
CONTADOR_MENOR_DOS	BOOL			
finalizar_calentamiento	BOOL			
L	EBOOL	%I0.1.2		Limpieza
Llanado_internabrador	INT			

Figura 11: Variables elementales

En la Figura 12, se observa la tabla de todas las variables elementales, donde aparece el nombre, tipo y dirección de cada una de ellas:

Nombre	Tipo	Dirección	Nombre	Tipo	Dirección
AC	EBOOL	%I0.1.3	N100	EBOOL	%I0.1.7
Agua_limpieza	INT		Nivel_A	INT	
B1	EBOOL	%Q0.9.2	Nivel_B	INT	
B2	EBOOL	%Q0.2.7	prod_A	BOOL	
B3	EBOOL	%Q0.2.8	Prod_B	BOOL	
Calentar	BOOL		Producto	DWORD	%MW0
cont	DINT	%MW2	R	EBOOL	
cont2	BOOL		T_1min	BOOL	
cont_menor_2	BOOL		T_50	EBOOL	%I0.1.8
Deposito_lleno	BOOL		T_55	EBOOL	%I0.1.9
Deposito_vacio	BOOL		T_85	EBOOL	%I0.8.2
Dosificador_A	INT		T_90	EBOOL	%I0.8.1
Dosificador_B	INT		TA	EBOOL	%I0.1.0
fin_de_mezcla	BOOL		Tanque_A	INT	
Fin_llenado	BOOL		Tanque_B	INT	
L	EBOOL	%I0.1.2	Tanque_vaciado	INT	
Limpieza	BOOL		TB	EBOOL	%I0.1.1
llenado_intercambiador	INT		temp_1min	BOOL	
Llenar_producto_A	INT		V1	EBOOL	%Q0.2.0
Llenar_producto_A2	INT		V2	EBOOL	%Q0.2.1
Llenar_producto_B	INT		V3	EBOOL	%Q0.2.2
Llenar_producto_B2	INT		V4	EBOOL	%Q0.2.3
M	EBOOL	%Q0.2.9	V5	EBOOL	%Q0.2.4
N0	EBOOL	%I0.1.4	V6	EBOOL	%Q0.2.5
N2_3	EBOOL	%I0.1.5	V7	EBOOL	%Q0.2.6
N3_4	EBOOL	%I0.1.6	VA	EBOOL	%Q0.9.0
N100	EBOOL	%I0.1.7	VB	EBOOL	%Q0.9.1

Figura 12: Tabla de Variables Elementales

Hay que señalar, que las variables que tienen un candado a la izquierda, se generan automáticamente por el software, pues corresponden a transiciones ligadas a una sección, y es el nombre que se les pone en la programación.

Se puede observar en la figura anterior, que las entradas “T_85” y “T_90”, y las salidas “B1”, “V1” y “V2” pertenecen al segundo rack, mientras que todas las demás entradas y salidas pertenecen al primero. Esto se observa en la nomenclatura de su dirección. La manera de configurar ambos bastidores, es asociando cada una de las variables creadas al bloque que pertenezca, conociendo que ambos bastidores tienen 16 bits para entradas y salidas respectivamente. Si se hace doble clic en el bloque “DEY 16FK” del primer bastidor de la Figura 8, se obtiene la siguiente imagen de la Figura 13:

	Símbolo	Filtro	Función
0	TA	4 ms	
1	TB	4 ms	
2	L	4 ms	
3	AC	4 ms	
4	N0	4 ms	
5	N2_3	4 ms	
6	N3_4	4 ms	
7	N100	4 ms	
8	T_50	4 ms	
9	T_55	4 ms	
10		4 ms	
11		4 ms	
12		4 ms	
13		4 ms	
14		4 ms	
15		4 ms	

Figura 13: Entradas Bastidor 1

Las 16 entradas del bastidor 1, quedan definidas a través de dos canales de 1 byte cada uno.

Haciendo doble clic en el bloque “DSY 16R5” del primer bastidor de la Figura 8, se obtienen 16 salidas distribuidas en los Canales 0 y 8 de 1 byte cada uno. Dichos 16 bits aparecen en la Figura 14:

Índice	Símbolo	Valor de retorno
0	V1	0
1	V2	0
2	V3	0
3	V4	0
4	V5	0
5	V6	0
6	V7	0
7	B2	0
8	B3	0
9	M	0
10		0
11		0
12		0
13		0
14		0
15		0

Figura 14: Salidas Bastidor 1

Análogamente, haciendo doble clic en el bloque “DEY 16FK” del segundo bastidor de la Figura 8, se obtiene la siguiente imagen de la Figura 15:

Índice	Símbolo	Filtro	Función
0		4 ms	
1	T_90	4 ms	
2	T_85	4 ms	
3		4 ms	
4		4 ms	
5		4 ms	
6		4 ms	
7		4 ms	
8		4 ms	
9		4 ms	
10		4 ms	
11		4 ms	
12		4 ms	
13		4 ms	
14		4 ms	
15		4 ms	

Figura 15: Entradas Bastidor 2

En este caso, sólo se han hecho uso de dos bits de los 16 disponibles, pues sólo se han asociado dos entradas.

Haciendo doble clic en el bloque “DSY 16R5” del segundo bastidor de la Figura 8, se obtienen 16 salidas distribuidas en los Canales 0 y 8 de 1 byte cada uno. Dichos 16 bits aparecen en la Figura 16, y sólo se hacen uso de tres, pues son éstas las salidas asociadas:

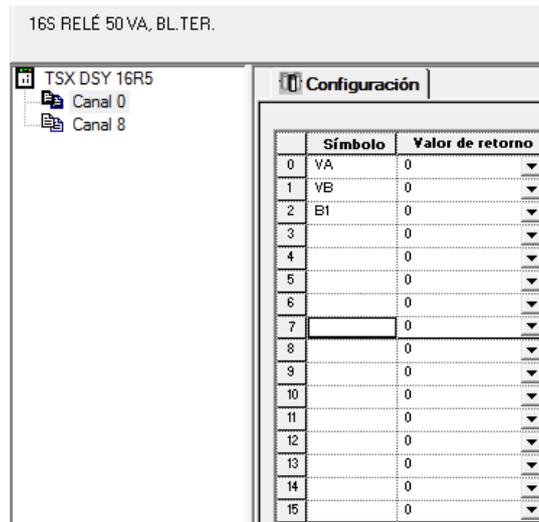


Figura 16: Salidas Bastidor 2

No es estrictamente necesario establecer las variables al comienzo del proyecto, ni definir las direcciones. Es posible ir incluyendo variables a la vez que se van necesitando en la programación. De la misma forma, se puede definir las direcciones de las variables al concluir la programación y antes de realizar la transferencia al autómatas.

Entre todas las variables elementales, las que han sido creadas exclusivamente para la simulación, son las mostradas a continuación, en la Figura 17:

Nombre	Tipo
● Agua_limpieza	INT
● Dosificador_A	INT
● Dosificador_B	INT
● llenado_intercambiador	INT
● Llenar_producto_A	INT
● Llenar_producto_A2	INT
● Llenar_producto_B	INT
● Llenar_producto_B2	INT
● Nivel_A	INT
● Nivel_B	INT
● Tanque_A	INT
● Tanque_B	INT
● Tanque_vaciado	INT

Figura 17: Variables creadas para la simulación

Como se puede comprobar y se explicará en el siguiente capítulo, donde desarrolla la simulación, el tipo de cada una de éstas variables es dependiente de la función que desempeñe el objeto al que esté asociado. En este caso son todas de tipo entero, "INT", pues van a representar niveles de capacidad.

4.3 Programación de la solución del problema: diagrama SFC

En esta sección, se programa el sistema automatizado mediante un diagrama SFC. Es el método de descripción de procesos, mediante un gráfico funcional interpretable. Tal gráfico permite normalizar la forma de descripción del proceso. Por esta razón, no siempre es estrictamente necesario, pudiéndose prescindir de él y abordar directamente la programación. En este documento se desarrollará un diagrama SFC programado en LD y además el mismo ejemplo programado en ST sin SFC, para mostrar la versatilidad de Unity Pro.

Para definir un diagrama SFC, se comienza como muestra la Figura 18, creando una nueva sección SFC:

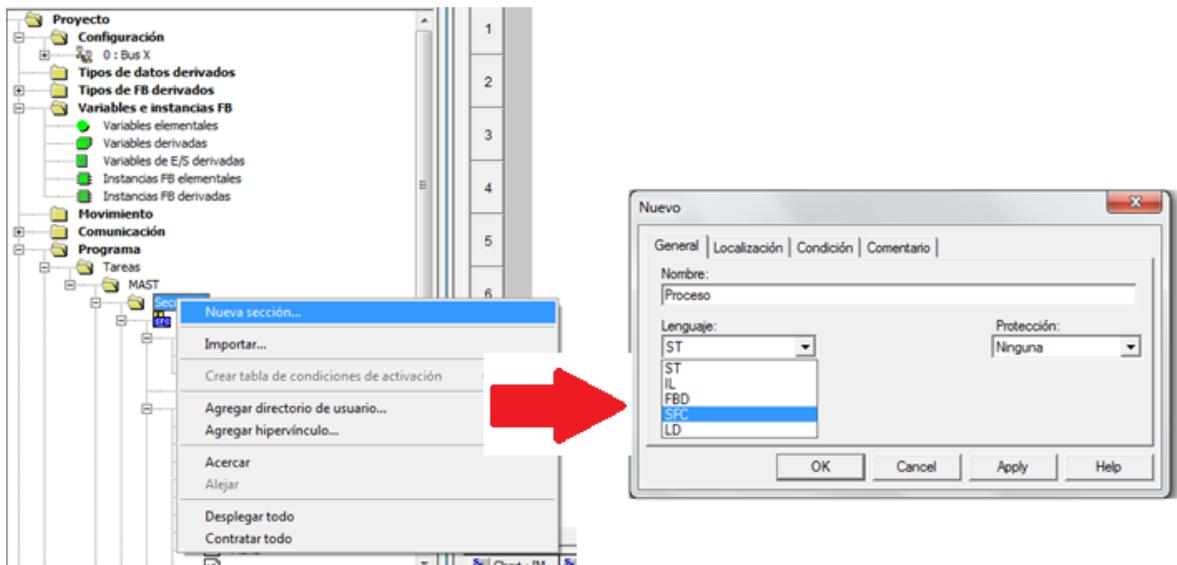


Figura 18: Creación de una nueva sección SFC

En el ejemplo programado, el diagrama SFC final es el que se indica en la Figura 19. En él se observan definidas todas y cada una de las etapas que son necesarias para su ejecución.

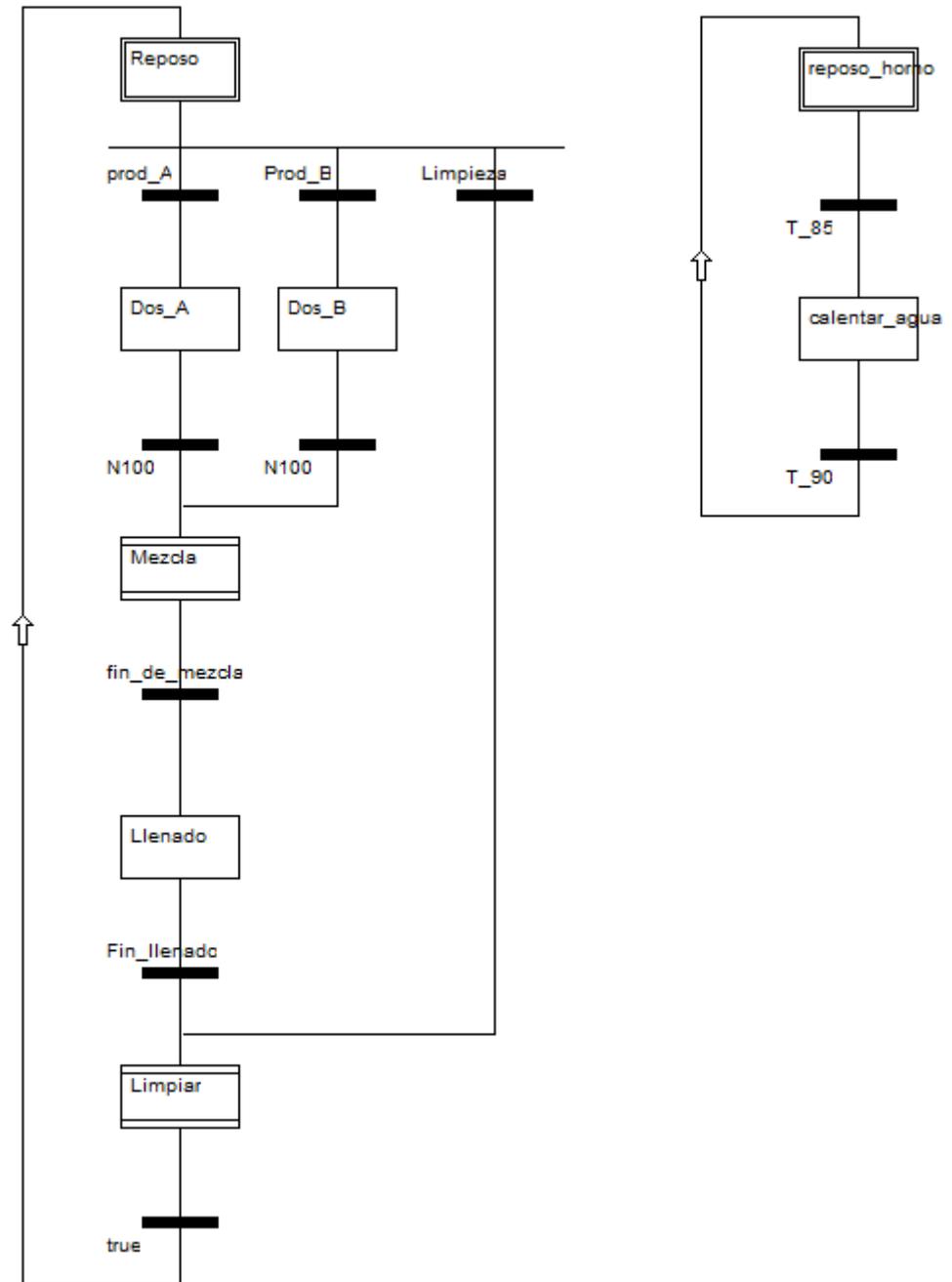


Figura 19: Diagrama SFC

Cuando se selecciona en el software Unity Pro una representación SFC, la pantalla presenta el siguiente aspecto de la Figura 20:

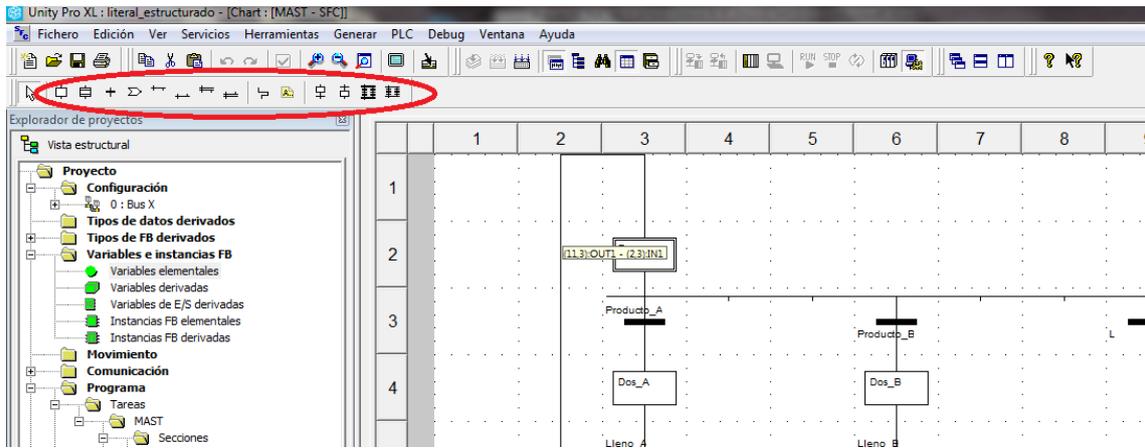


Figura 20: Elementos SFC

Y la barra de herramientas que aparece en la pantalla, es la siguiente mostrada en la Figura 21:

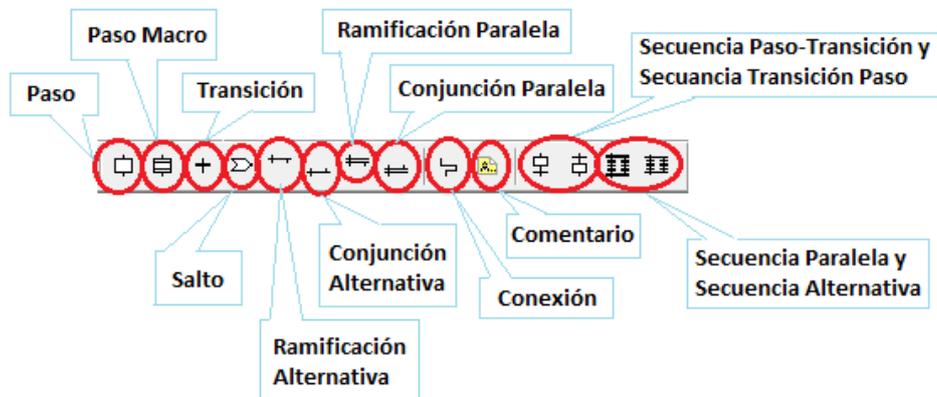


Figura 21: Elementos necesarios para crear el diagrama SFC

4.3.1 Programación de etapas o pasos

Para la programación de las etapas o pasos, todos los elementos se insertan arrastrando con el ratón uno a uno los símbolos necesitados. Una vez situados en la zona de trabajo (formada por una cuadrícula), se nombran y se programan las propiedades de cada paso o etapa, como muestra la Figura 22:

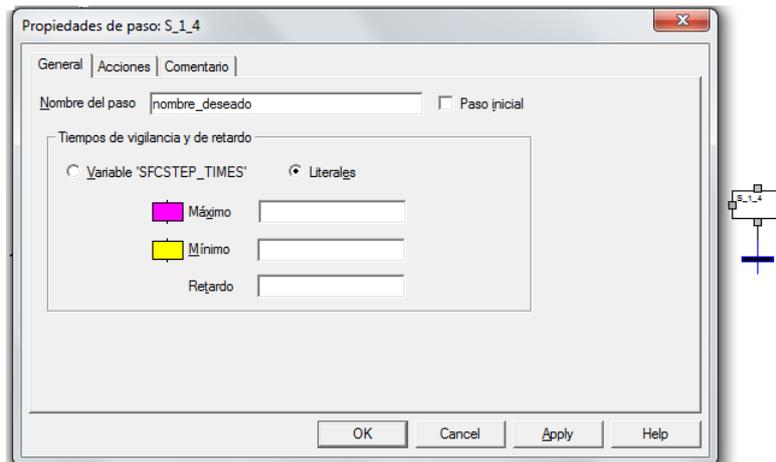


Figura 22: Programación de paso en SFC

En la pestaña Acciones, es el lugar donde se programan las acciones que se deberá ejecutar cuando la etapa esté activa. Existen dos maneras de programar las acciones de los pasos:

- Acción ligada a una variable
- Acción ligada a una sección

Cada acción dispone de un descriptor que es el que determina, o bien en qué momento se ejecuta la acción, o qué acción se va a ejecutar. A continuación, se presentan algunos de los descriptors más utilizados:

- N: Se ejecuta la acción programada en la sección (o pone a "1" la variable asignada) durante el tiempo que está activa la etapa.
- P1: Se ejecuta la acción programada en la sección (o pone a "1" la variable asignada) cuando la etapa pasa de inactiva a activa (acción al activar).
- P0: Se ejecuta la acción programada en la sección (o pone a "1" la variable asignada) cuando la etapa pasa de activa a inactiva (acción al desactivar).
- S: Pone a "1" la variable asignada.
- R: Pone a "0" la variable asignada

Para programar una **acción ligada a una variable** se selecciona “Variable” en el recuadro acción de la pestaña “Acciones” y se selecciona dicha variable a través del desplegable que aparece al pulsar sobre el cuadrado gris con tres puntos, o se escribe directamente. También hay que definir uno de los descriptores que se ajuste a la acción que se quiere ejecutar en esa etapa. La Figura 23 muestra un ejemplo en que se ha elegido la variable “V1” y el descriptor “P0”, pues sólo se quiere que se ejecute cuando la etapa pasa de activa a inactiva:

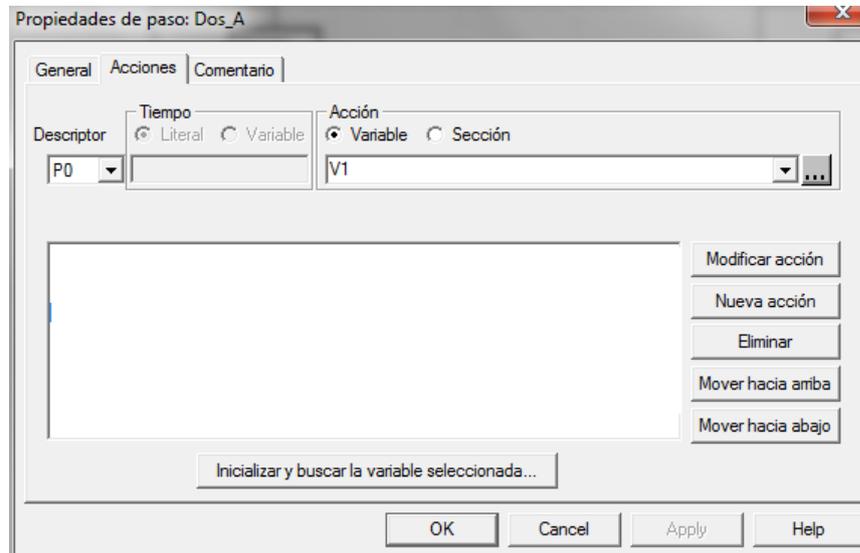


Figura 23: Acción ligada a un sección en la creación de pasos

Para determinar una **acción ligada a una sección**, se elige un descriptor que mejor se ajuste al programa, se pone nombre a la sección y se selecciona “Nueva acción”. Tras pulsar el botón inferior de “Editar sección de acciones” mostrado en la Figura 24, el software mostrará una pantalla, en la que se podrá elegir uno de los lenguajes entre los cuatro disponibles, y en el que se quiere programar, tal y como se muestra en la Figura 24:

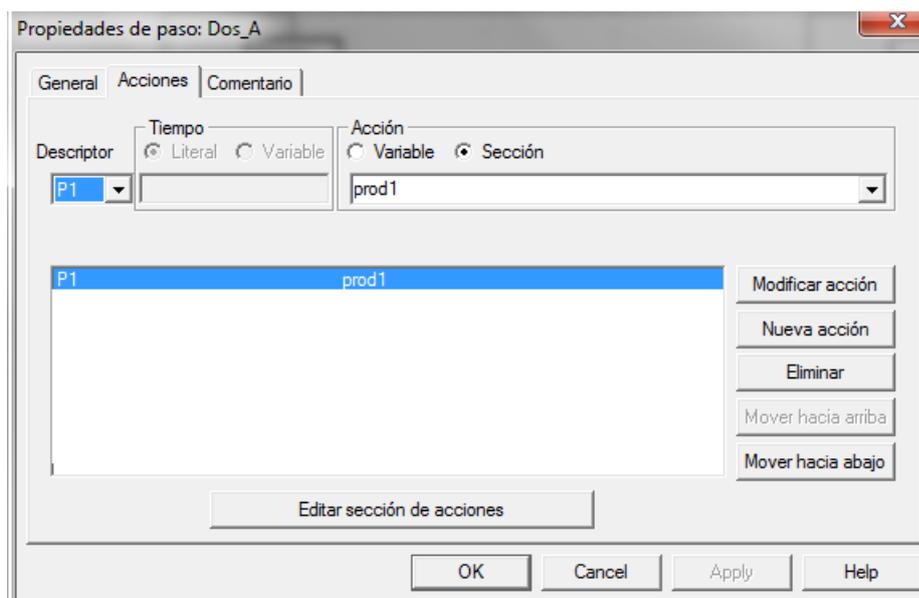


Figura 24: Acción ligada a una variable en la programación de pasos



Figura 25: Tipo de lenguaje

4.3.2 Programación de Transiciones

Para programar las transiciones, existen dos formas posibles:

- La condición de transición ligada a una variable
- La condición de transición ligada a una sección

Para programar una **condición de transición ligada a una variable**, como muestra la Figura 26, es necesario seleccionar "Variable" en el tipo de condición de transición, y elegir la dirección, valor o variable Bool que hace referencia la transición:

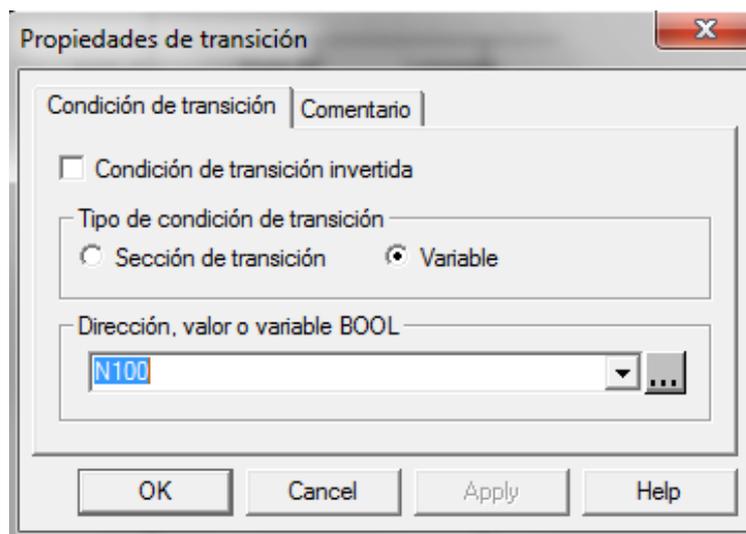


Figura 26: Condición de transición ligada a una variable

Para programar la **condición de transición ligada a una sección**, es necesario seleccionar y poner el nombre en la “Sección de transición”, como se visualiza en la Figura 27, y a continuación, pulsar “Editar”. Aparece en la Figura 28, la pantalla donde se especifica el lenguaje de programación:

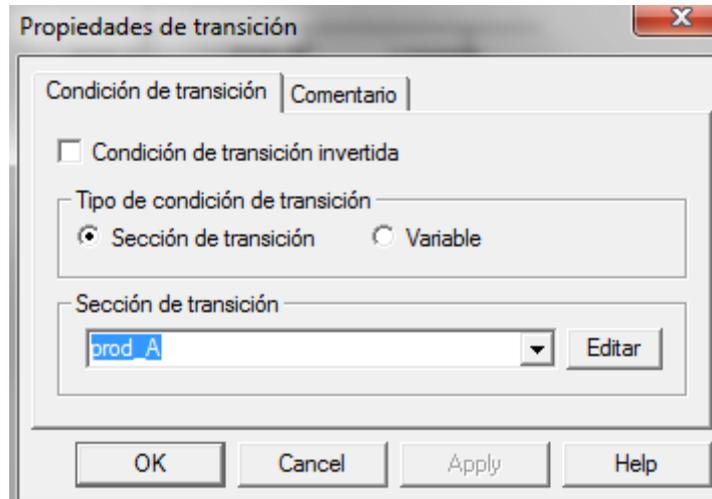


Figura 27: Condición de transición ligada a una sección



Figura 28: Tipo de lenguaje

4.3.3 Programación de Macroetapas

Antes de continuar, hay que detenerse en el concepto de macroetapa. Al programar en el lenguaje SFC, con el fin de simplificar la programación, es posible agrupar subprocesos complejos en grandes bloques de acciones llamados macroetapas.

En el ejemplo desarrollado, van a ser las etapas de mezclado y limpieza las que van a estar programadas por macroetapas, tal y como se observa en la Figura 29:

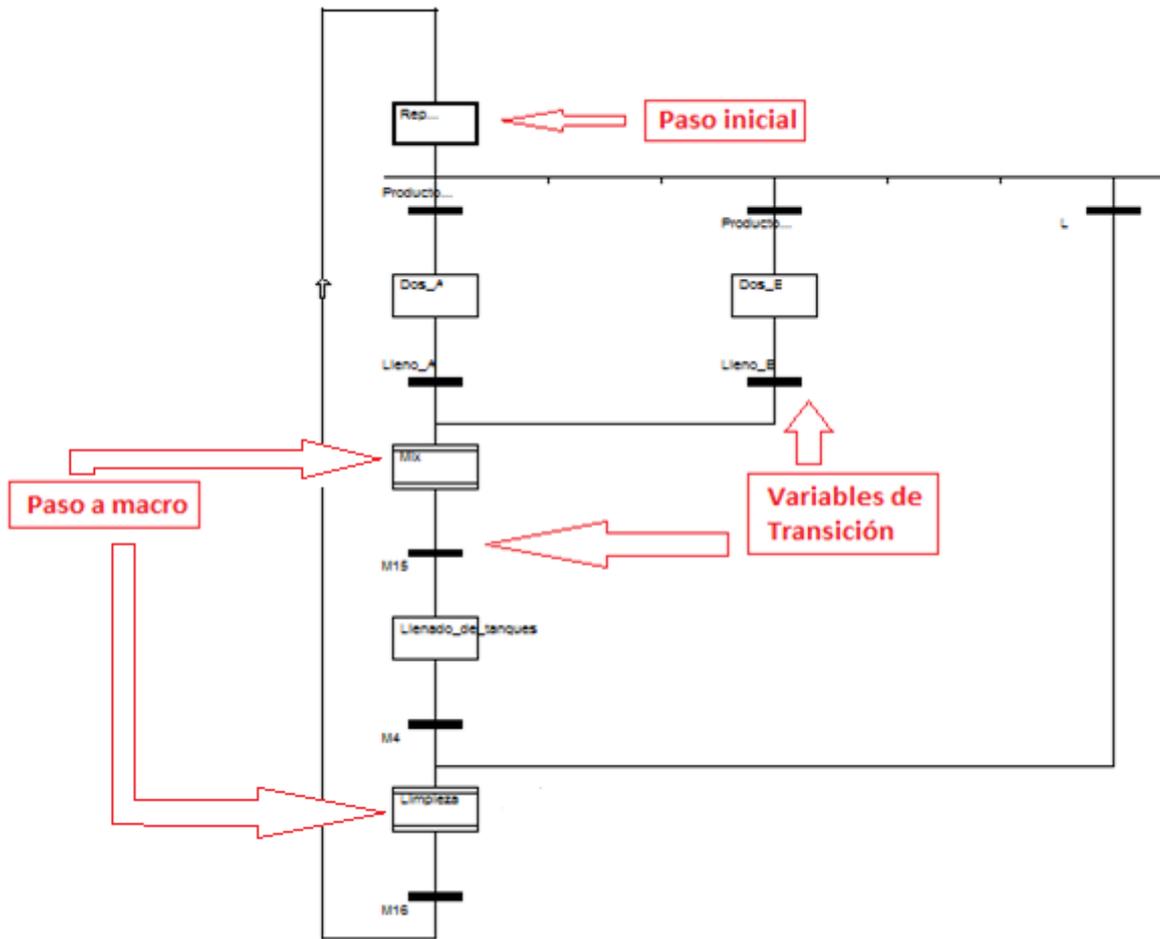


Figura 29: SFC principal

Estas macroetapas pueden ser divididas a su vez en operaciones más elementales, hasta conseguir que las acciones a realizar en cada una de ellas solamente dependan de relaciones combinatorias entre entradas y salidas. Cada una de las etapas elementales llevará asociada una variable de estado.

Por tanto, las macroetapas representan “tareas” y equivalen a lo que en algunos lenguajes se define abreviadamente como “macros”. Desde un punto de vista formal, una macroetapa no es más que un conjunto de etapas, valga la redundancia, agrupadas que se definen posteriormente de forma individual, en lo que se denomina en detalle o expansión de la macroetapa.

En esencia, la finalidad de la macroetapa es la de permitir una aproximación progresiva y estructurada al proceso en todas las fases. Se puede partir de una definición en líneas generales (macroetapas) del proceso y posteriormente descomponer cada macroetapa en las acciones simples correspondientes.

La expansión de una macro debe tener una única etapa inicial y una única etapa final, sin que ello condicione a la macro a una estructura lineal. En efecto, dentro de la macro pueden existir estructuras de cualquier complejidad e incluso otras macros “anidadas”. Es decir, la expansión de una macroetapa puede, a su vez, utilizar otras macroetapas, fragmentando así el problema

global en “tareas” que normalmente se hacen corresponder a partes del proceso tecnológicamente completas. Esta herramienta va a ser muy útil en la etapa limpieza.

Una idea muy importante, es que el franqueo de la transición inmediatamente anterior a la macro, activa la etapa de entrada de la misma y, la activación de la etapa de salida de la macro valida la transición siguiente de la misma.

La utilidad de las macroetapas se potencia notablemente, de esta forma, pueden definirse como auténticas “subrutinas” o “procedimientos” que pueden ser llamados desde cualquier punto del programa, con la única condición de que no sean llamadas mientras se están ejecutando. Lo anterior permite una programación más estructurada de las tareas de un proceso y acorta la longitud del programa.

Una vez aclarado el concepto macroetapa, se va a ir analizando una a una cada una de las etapas en cada uno de los lenguajes escogidos.

4.3.4 Lenguaje LD

En esta sección se va a describir como se programa en lenguaje LD aquellas acciones y/o transiciones ligadas a secciones del diagrama SFC.

La Figura 30, recuerda los pasos a seguir para llegar a programar en lenguaje LD, para la condición de transición ligada a una sección. Tras este paso, aparecerá una pantalla en la que es posible escribir en LD.

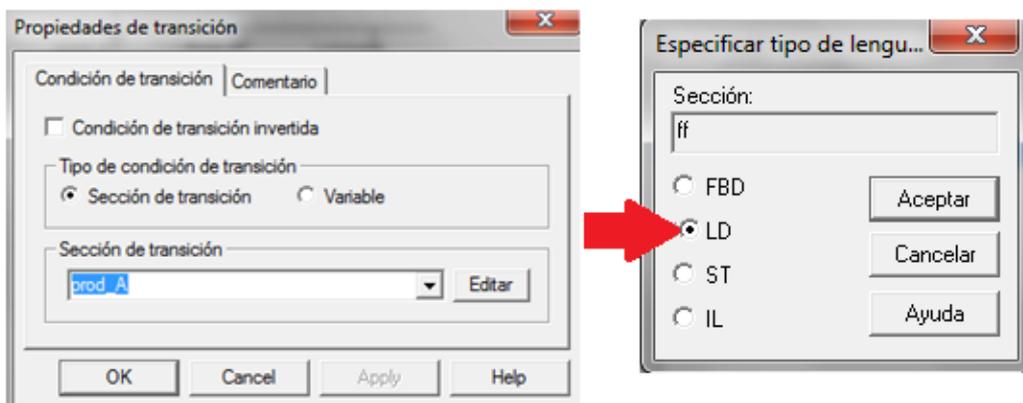


Figura 30: Condición de transición ligada a una sección

La Figura 31, recuerda los pasos para alcanzar a definir una acción ligada a una sección:

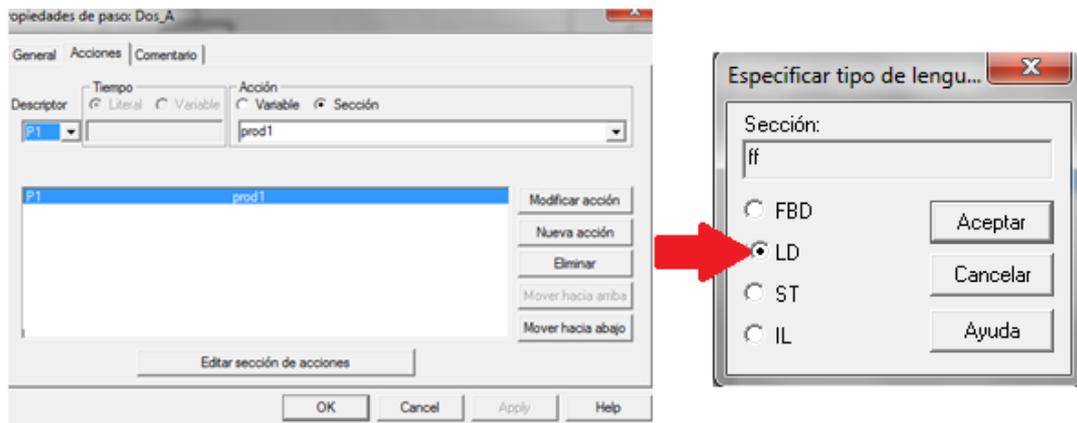


Figura 31: Acción ligada a una sección

Ahora ya es posible comenzar a escribir en lenguaje LD, y las herramientas disponibles para este lenguaje, son las mostradas en la Figura 32:

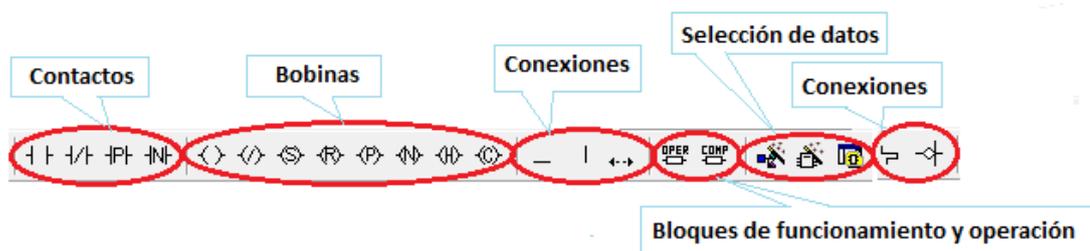


Figura 32: Alternativas de elementos para programación lenguaje LD

Para facilitar la comprensión del diagrama, se va a recorrer una a una las etapas que componen el sistema. En cada una de ellas, va a aparecer a la izquierda marcada mediante un círculo rojo la etapa en cuestión, para que el seguimiento de la explicación sea más sencillo.

Al constar de dos diagramas SFC, se comienza por el diagrama principal donde suceden las etapas de dosificación, mezcla y limpieza, y se continuará por el diagrama relacionada con el calentador de agua.

4.3.4.1 Programación SFC principal

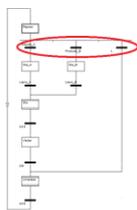
El recorrido de la programación, será desde la etapa inicial a la etapa de finalización de ciclo. A continuación se mostrará el desarrollo de todas las variables utilizadas. Tanto las de programación del sistema estrictamente, como las creadas para la simulación.

“Etapa reposo”: La etapa de inicialización es la denominada “reposo”. Por ser el inicio del diagrama SFC, su representación es de doble rectángulo.

Durante toda la programación se va a hacer uso de la variable “Producto”. Esta variable ha sido definida para recibir valores mediante el operador. Si el usuario elige el “Producto A (TA)”, el valor de “Producto” es “1”. Si es “Producto B (TB)” el valor de “Producto” es “2”. Al comienzo del sistema, dicha variable recibe el valor de “0”, como muestra la Figura 33:



Figura 33: Etapa reposo



“Transición Producto A/ Producto B/L”: Para que el diagrama siga su curso, se debe cumplir una de las tres transiciones posibles. Las tres son transiciones ligadas a una sección. Cada una de ellas, son las que a continuación se describen:

- Producto_A.
- Producto_B.
- Limpieza

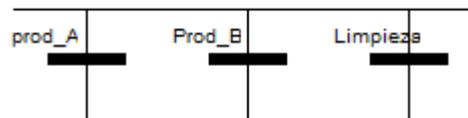


Figura 34: “Transición Producto_A/Productivo_B/L”

“Producto_A”: cuando se dan las condiciones de que el usuario activa la “alimentación (AC)” y el “producto A (TA)”, se cumple la transición “prod_A”.

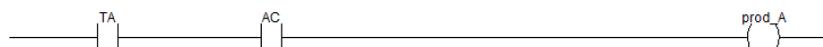


Figura 35: “Transición Producto_A”

“Producto_B”: cuando se dan las condiciones de que el usuario activa la “alimentación (AC)” y el “producto B (TB)”, se cumple la transición “prod_B”.



Figura 36: “Transición Producto_B”

“Limpieza”: las condiciones de “alimentación (AC)” y “limpieza (L)”, dan lugar a la limpieza del tanque.



Figura 37: “Transición Limpieza”

En el caso de que el usuario elija el producto A, debe cumplirse la condición “Producto_A”. El diagrama va a continuar por esta rama.



“Etapa dosificación A”: La etapa de “dosificación_B” será completamente igual, pero cambiando las condiciones exigidas de niveles de producto, para una mayor brevedad del documento, se centra la atención en la “Dosificación_A”.

En la “Etapa dosificación A” va a haber tres acciones ligadas a una sección: “prod1”, “V1_V2” y “The_end”. La Figura 38 muestra la pantalla de las “Propiedades de pasos”:

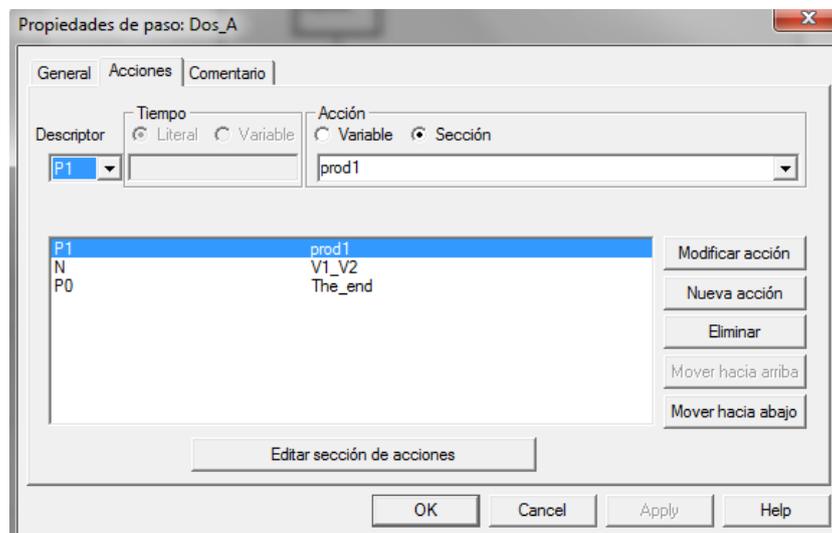


Figura 38: Propiedad del Paso “Dos_A”

En la acción "prod1", se le asigna el valor "1" a la variable "Producto". La sección a la que está ligada se muestra en la Figura 39:



Figura 39: Acción "Prod1"

En la acción "V1_V2", cuando todos los sensores de nivel están desactivados indicando que el "depósito de mezcla" está vacío, o como se requiere un 33% de "Producto A", si el sensor indicador del 33% del tanque de mezcla "N2_3" no está activo, pero sí el sensor de 0 % de nivel, se abre la válvula "V1". Los valores de "Llenar_producto_A" y "Dosificador_A" pasan a ser "1000", tomando su máximo valor.

Cuando los sensores que indican 0% y 33% están activos y desactivados los que indican 75% y 100%, o cuando los sensores que indican 0%, 33% y 75% están activos y el que indica 100% desactivado, la válvula "V2" se abre. Los valores de "Dosificador_B" pasará a ser 1000, y el de "Dosificador_A" será 0. La sección a la que está ligada, se muestra en la Figura 40:

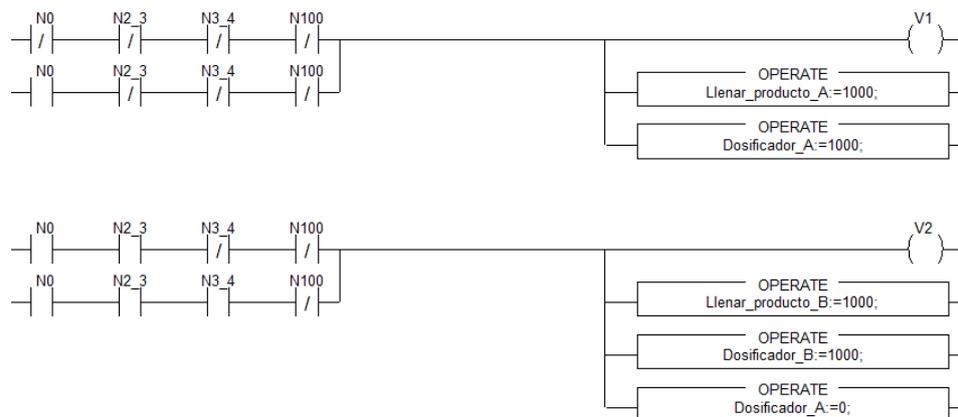


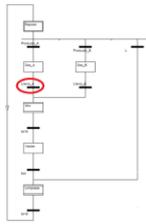
Figura 40: Acción "V1_V2"

En la acción "The_end", si el sensor 100% está activo, el nivel de la variable "Dosificador_B" es cero. La acción a la que está ligada se muestra en la Figura 41:

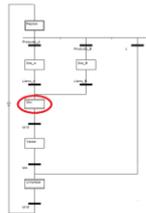


Figura 41: Acción "The_end"

Para mayor seguridad en el funcionamiento del sistema, se puede añadir una acción ligada a una sección en la que se reseteen las válvulas abiertas en la etapa, en este caso "V1" y "V2". Sin embargo esta acción sería redundante porque la etapa se ha programado asociada al descriptor "N". Sería aplicable a cada una de las etapas.



“Transición N100”: Se trata de la transición ligada a una variable. Ocurre si está activo el sensor N100.



“Macroetapa Mezcla”: al ser una macroetapa, está compuesta de etapa de inicio: **“Mezcla_IN”**, etapa principal de la macroetapa: **“Agitar”**, y etapa final: **“Mezcla_OUT”**.

Cada una de ellas están programadas mediante código LD y entre ellas las transiciones correspondientes. El diagrama de la **“Macroetapa Mezcla”**, se muestra en la Figura 42:

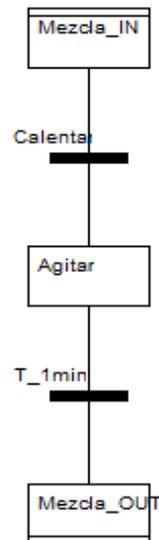
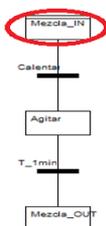


Figura 42: Macroetapa Mezcla



“Macroetapa Mezcla”-“Etapa Mezcla IN”: El **“depósito de mezcla”** se calienta mediante agua caliente. El agua como se mantiene caliente en un depósito aparte, es necesario activar las válvulas **“VA”**, **“VB”** y la bomba **“B1”** para que se produzca la mezcla. Las tres acciones están asociadas al descriptor **“N”**, pues es necesario que estén activas durante la duración de la etapa. La transiciones ligadas a las variables correspondientes se muestran en la Figura 43:

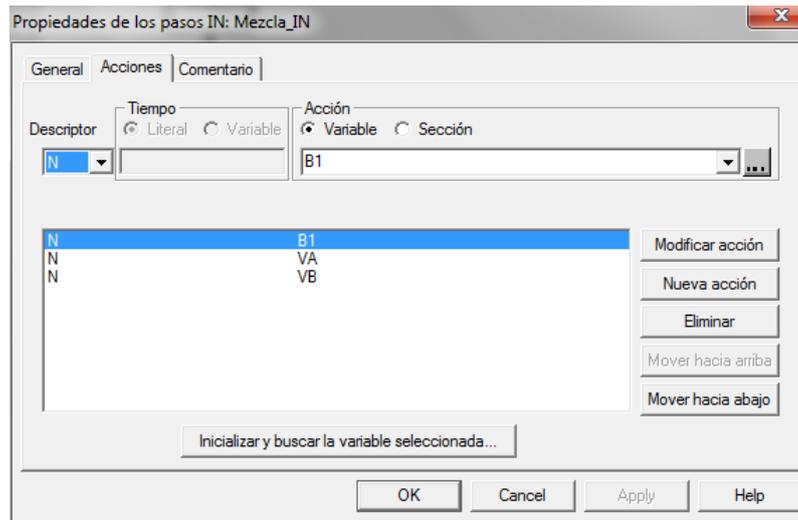
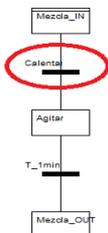


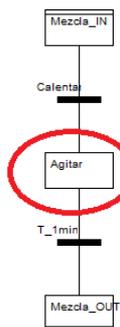
Figura 43: Macroetapa Mezcla-“Etapa Mezcla_IN”



“Macroetapa Mezcla”-“Transición Calentar”: Es una transición ligada a una sección. Si el usuario ha elegido el “Producto A” y se ha alcanzado la Temperatura de 50°C ó si ha elegido el “Producto B” y la Temperatura es 55°C, da paso a la siguiente etapa para comenzar a calentar la mezcla. La sección de transición se muestra en la Figura 44:



Figura 44: “Transición Calentar”



“Macroetapa Mezcla”-“Etapa Agitar”: Está formada por dos acciones ligada a una sección como muestra la Figura 45:

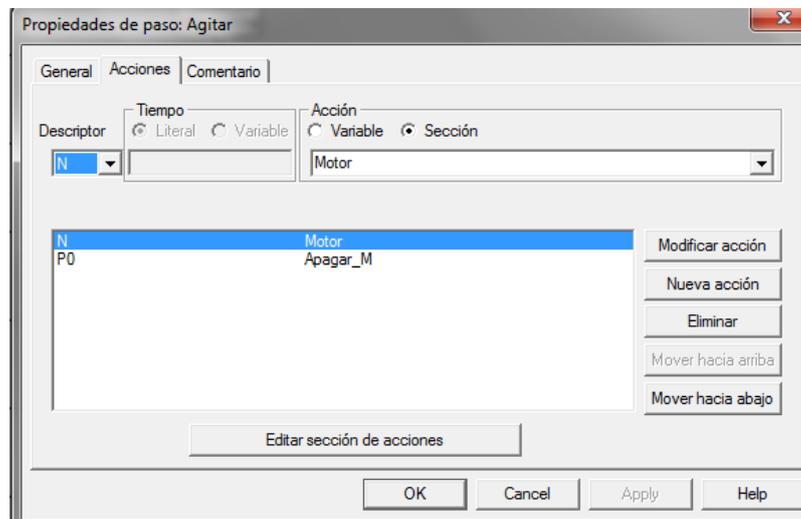


Figura 45: Acciones de macroetapa mezclar

En la sección “Motor”, dependiendo si se ha alcanzado la temperatura de 50º para el “Producto_A” o de 55º para el “Producto_B”, se activa el motor. La sección es mostrada en la Figura 46:



Figura 46: Acción “Motor”

En la sección “Apagar_M”, se resetea la variable motor “M”, como muestra la Figura 47:

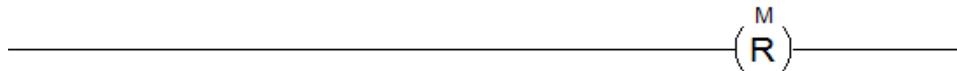


Figura 47: Acción “Apagar_M”

“Macroetapa Mezcla”-“Transición T_1min”: Se ejecuta la etapa “agitar” durante 1 minuto, como muestras la Figura 48:

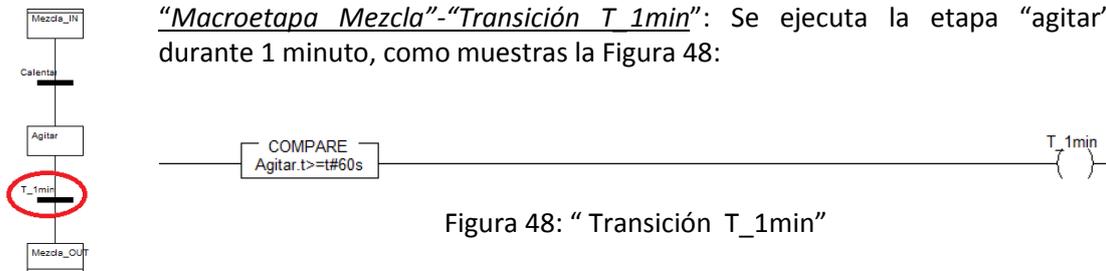
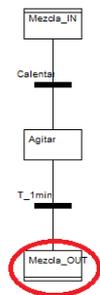


Figura 48: “ Transición T_1min”

“Macroetapa Mezcla”-“Mezcla OUT”: Tras producirse la mezcla durante un tiempo, se abandona la macroetapa, y el sistema continua por el diagrama principal.



Lo que está ocurriendo realmente se puede observar de una manera clara, en la Figura 49:

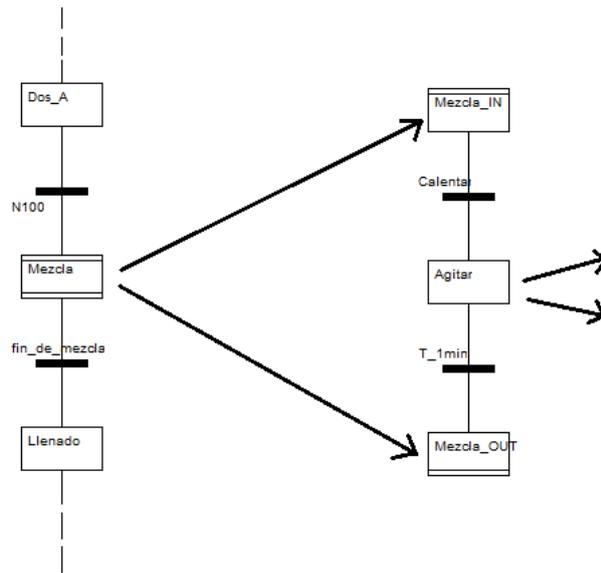
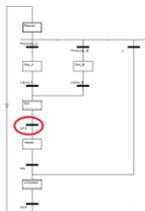


Figura 49: Explicación de macroetapa Mezclar

Dentro de la macroetapa “Mezcla”, se produce un diagrama SFC formado por tres etapas que son la de entrada, la de salida y una intermedia como se ha comentado anteriormente.

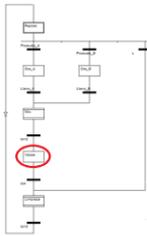


“Transición Fin de Mezcla”: Es una transición ligada a una sección. Después de estar el motor activo durante un tiempo, y antes de pasar a la “etapa de llenado”, los niveles de ambos productos se igualan a cero. Cada una de estas variables que se han igualado a cero, se explicarán en el capítulo de simulación el modo de programarlas.

La sección se muestra en la Figura 50:



Figura 50: Transición “Fin de Mezcla”:



“Etapa Llenado”: La etapa llenado, consta de tres acciones ligadas a una variable y tres acciones ligadas a una sección, como muestra la Figura 51:

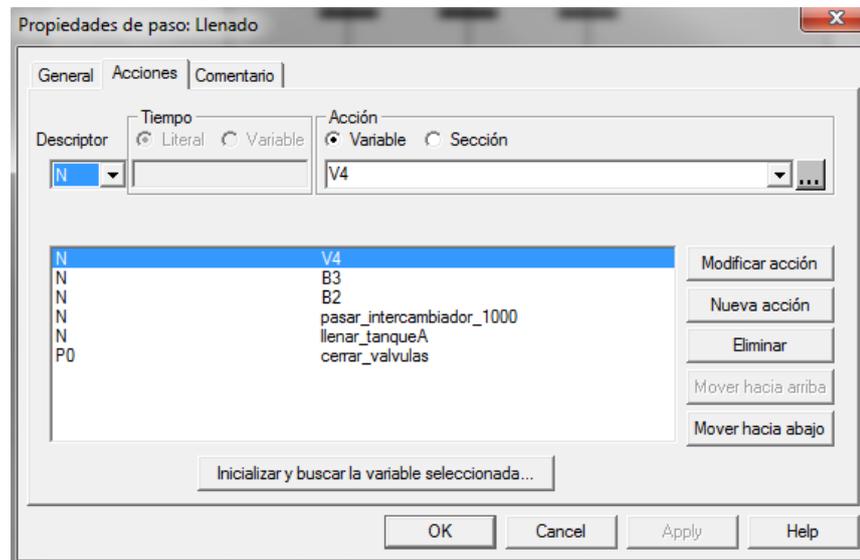


Figura 51: Propiedades de paso “Llenado”

En las tres primeras, acciones ligadas a una variable, dichas variables son “V4”, “B2” y “B3”. La acción es programada durante el tiempo que está activa la etapa, mediante el descriptor “N”.

En la acción ligada a una sección denominada “pasar_intercambiador_1000”, se asigna el valor “llenado_intercambiador=1000”, como muestra la Figura 52:



Figura 52: Acción “pasar_intercambiador_1000”

En la acción ligada a una sección llamada “llenar_Tanque_A”, se determina la variable “Producto=1” para “Producto_A”. La “V5” se activa y el valor del nivel “Tanque_A=1000”.

En la segunda línea, la variable “Producto=2” para “Producto_B”, en cuyo caso la “V6” se activa y el valor del nivel “Tanque_B=1000”. La sección se observa en la Figura 53:

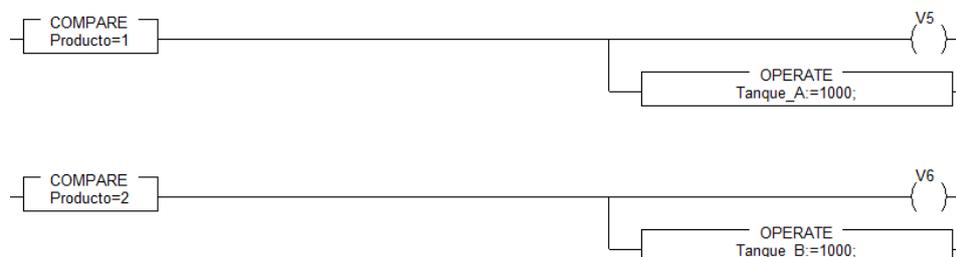


Figura 53: “llenar_Tanque_A”

En la acción ligada a una sección llamada “cerrar_valvulas”, en el caso de Producto A, la variable “Producto=1”, se resetea la válvula “V5”. En caso de que sea el Producto B, será “Producto=2” y se reseteará la variable “V6”, como muestra la Figura 54:

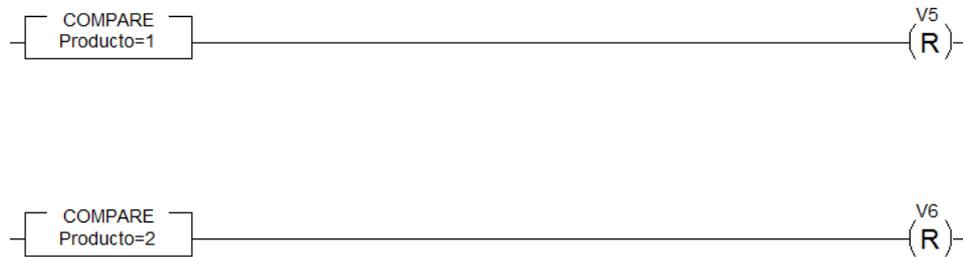
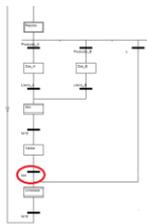


Figura 54: “cerrar_valvulas”



“Transición Fin llenado”: Es una transición ligada a una sección. Si los sensores de capacidad se encuentran desactivados, se da el valor de “llenado_intercambiador=0”. La sección se puede observar en la Figura 55:

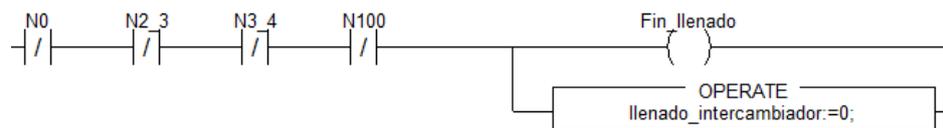
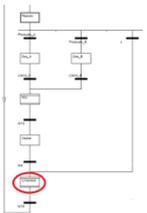


Figura 55: Transición “Fin llenado”



“Macroetapa Limpieza”: como se puede apreciar en la Figura 56 ,dicha macroetapa consta de etapa inicial: “LIMPIEZA_IN”, etapas y transiciones intermedias y etapa final: “LIMPIEZA_OUT”:

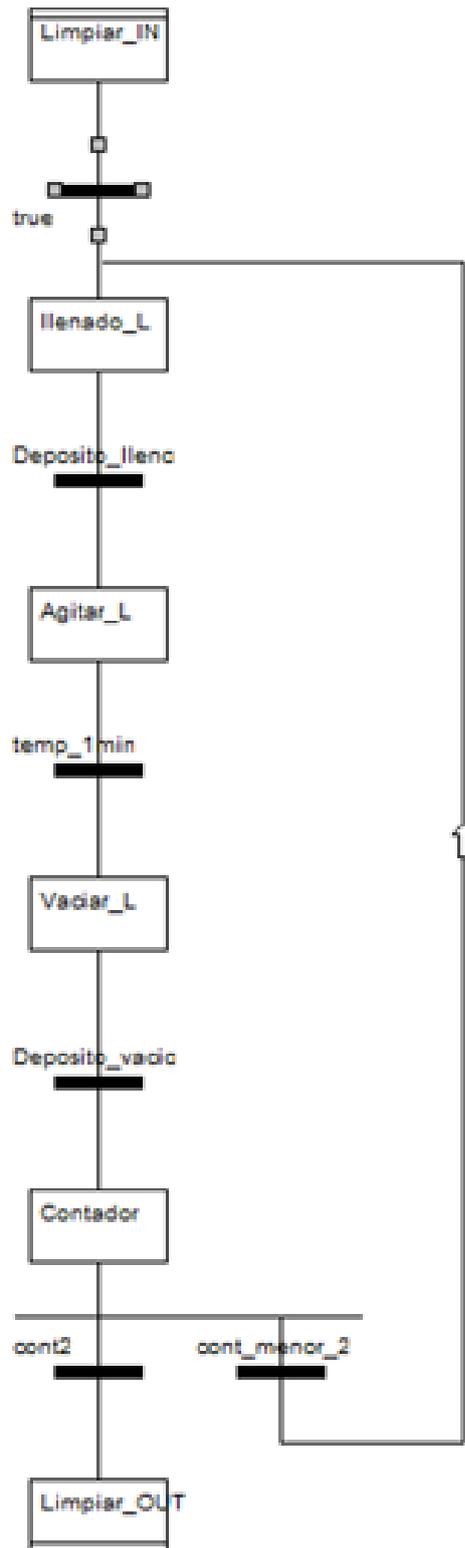
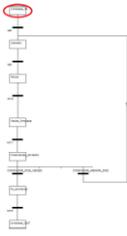


Figura 56: Macroetapa Limpieza



“Macroetapa Limpieza”-“Etapa LIMPIEZA_IN”: Se utiliza un contador para saber cuántas veces se ha realizado la limpieza. El ejemplo requiere dos turnos de limpieza en cada operación de Producto A o Producto B. Está programado mediante una acción ligada a una sección mostrada en la Figura 57, en la que se asigna el valor cero a la variable “cont”, observado en la Figura 58:

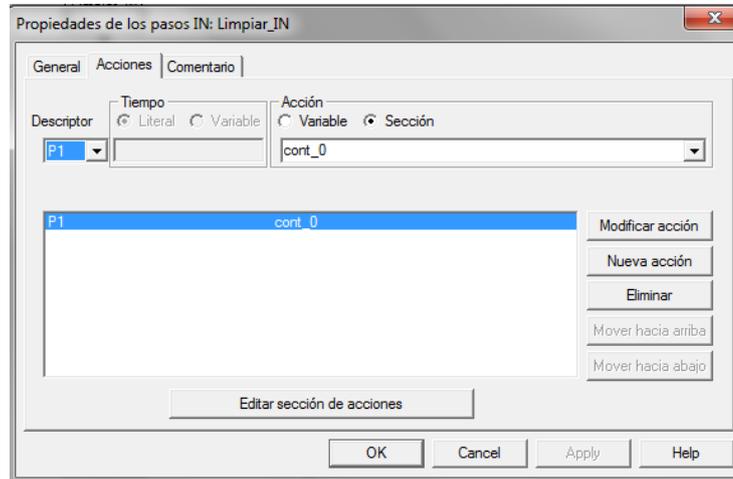


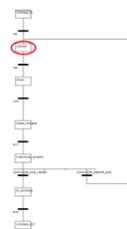
Figura 57: Propiedad del pasoIN “Limpiar_IN”



Figura 58:Etapa LIMPIEZA_IN



“Macroetapa Limpieza”-“Transición true”: Transición que da paso directamente a la siguiente etapa. Es siempre verdadera.



“Macroetapa Limpieza”-“Etapa Llenado L”: Es la etapa que corresponde al llenado de del agua del tanque de limpieza. Está programada mediante una acción ligada a una variable: “V3”, y mediante la acción ligada a una sección “agua”, como se visualiza en la Figura 59:

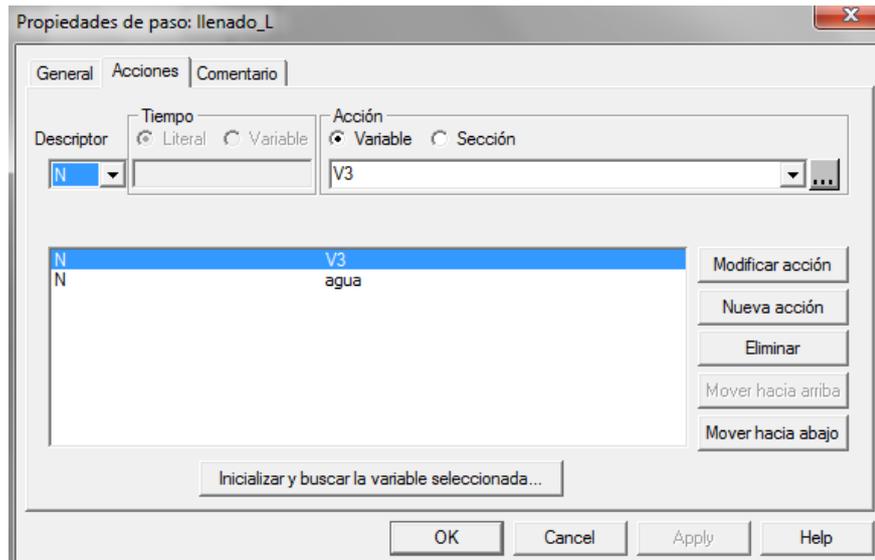
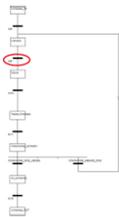


Figura 59: Propiedad de paso “llenado_L”

En la acción ligada a una sección “agua”, cuando la variable “V3” está activada, el valor de “Agua_limpieza=1000” su máximo valor porque está lleno. La acción se observa en la Figura 60:



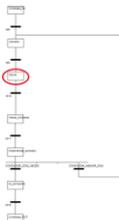
Figura 60: Etapa Llenado



“Macroetapa Limpieza”-“Transición Depósito Lleno”: Es una transición ligada a una sección. Cuando todos los sensores de nivel del “depósito de mezcla” están activados, se da paso a la etapa, como se observa en la Figura 61:



Figura 61: Transición “Depósito Lleno”



“Macroetapa Limpieza”-“Etapa Agitar L” : Es una acción ligada a una variable. Se activa mediante la variable “M” que corresponde al motor. Se ejecuta la acción durante el tiempo que está activa la etapa, descriptor “N”. La acción se observa en la Figura 62:

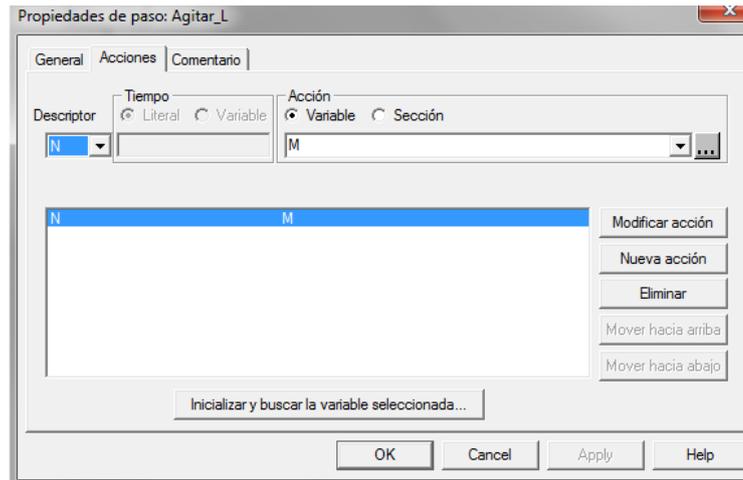
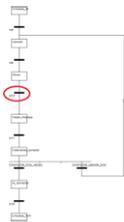


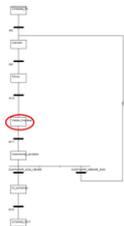
Figura 62: Propiedad del paso "Agitar_L"



"Macroetapa Limpieza"-"Transición temp_1min": Se trata de una transición ligada a sección. Como se observa en la Figura 63, transcurrido el tiempo de un minuto, se pasa a la siguiente atapa:



Figura 63: Transición "temp_1min"



"Macroetapa Limpieza"-"Etapa Vaciar L": Consta de tres acciones ligadas a una variable "V4", "V7"y "B3". En todas ellas, se ejecuta la acción ligada a la sección durante el tiempo que está activa la etapa, mediante el descriptor "N".

También está compuesta de tres acciones ligadas a una sección, las cuales son denominadas: "vaciado_agua", "llenado_Tanque_A_mitad", "Llenado_intercambiador_limpieza". El aspecto que muestra se observa en la Figura 64:

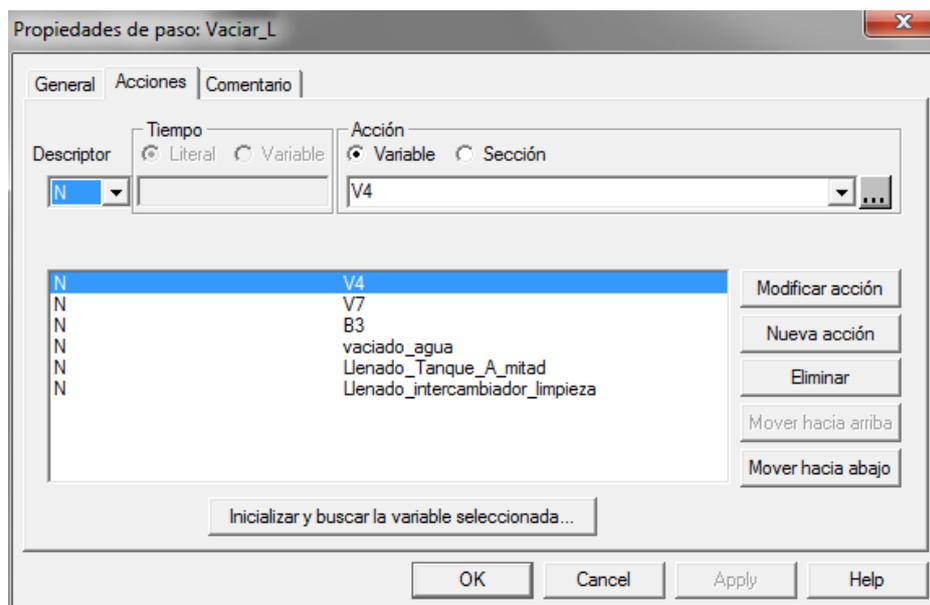


Figura 64: Propiedad del paso: "Vaciar_L"

En la acción ligada a la sección “*vaciado_agua*”, cuando el motor “M” deja de funcionar, el nivel “*Agua_limpieza=0*”. Se observa en la Figura 65:

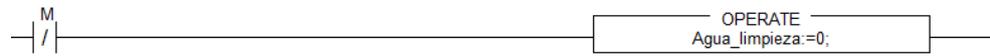


Figura 65: sección “*vaciado_agua*”

En la acción ligada a la sección “*llenado_Tanque_A_mitad*”, como se ha realizado la primera limpieza, el valor de “*Tanque_vaciado=500*” que corresponde a la mitad de su valor. Se puede comprobar en la Figura 66:

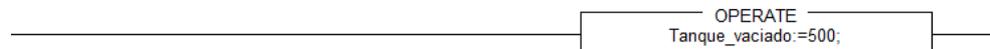
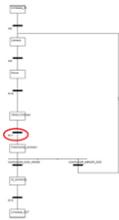


Figura 66: sección “*llenado_Tanque_A_mitad*”

En la acción ligada a la sección “*Llenado intercambiador limpieza*”, como ya se ha realizado las dos limpiezas requeridas, “*llenado_intercambiador:1000*” que corresponde a su valor máximo, como se observa con la Figura 67:



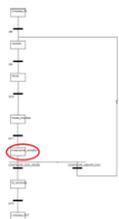
Figura 67: “Etapa Vaciado”



“*Macroetapa Limpieza*”-“*Transición Depósito vacío*”: Es una transición ligada a una sección. Si los sensores de nivel de capacidad del “depósito de mezcla” se encuentran desactivados, se da paso a la etapa siguiente, como se comprueba en la Figura 68:



Figura 68: “Transición “Depósito_vacio”



“*Macroetapa Limpieza*”-“*Etapa Contador*”: Se ejecuta la acción ligada a una sección durante el tiempo que está activa la etapa de la sección “*cont_mas*”, como se muestra en la Figura 69. En ella, se incrementa en “1” la variable “*contador*” que se comprueba en la Figura 70:

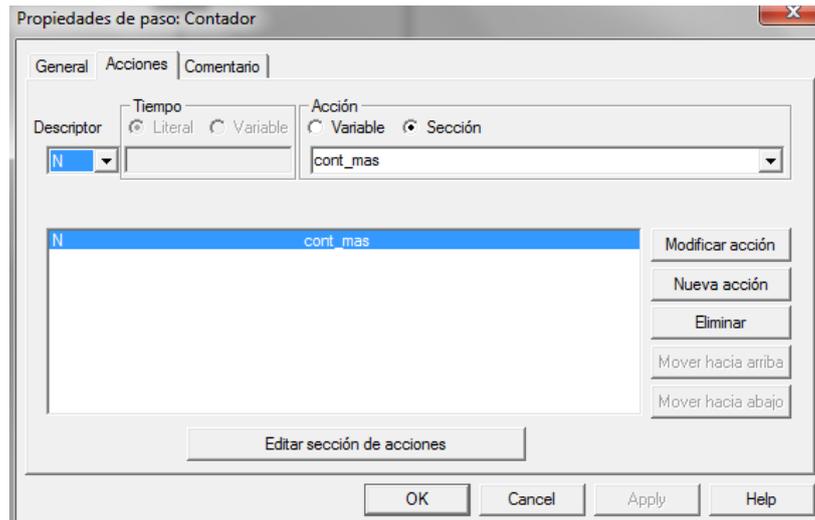
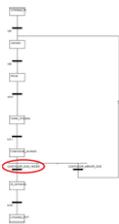


Figura 69 :Propiedades de paso “Contador”



Figura 70: Etapa “Incrementar_contador”



“Macroetapa Limpieza”-“Transición cont2”: Es una transición ligada a una sección. Cuando el contador ya está en valor 2, o para cubrir todos los casos posibles es mayor que este valor, el nivel de “llenado:_intercambiador=0” y “Tanque_vaciado=1000”, su máximo valor. La sección se observa en la Figura 71:



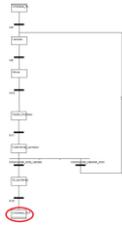
Figura 71: Transición “Contador_dos_veces”



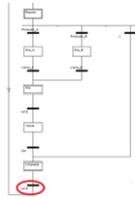
“Macroetapa Limpieza”-“Transición cont menor 2”: es una transición ligada a una sección. Cuando la variable “contador” todavía no ha alcanzado el valor 2, y por tanto su valor es menor, el nivel del “llenado_intercambiador=0” se da paso a etapa que repite el proceso, como se comprueba en la Figura en la Figura 72:



Figura 72: Transición “Contador_Menor_de_dos”



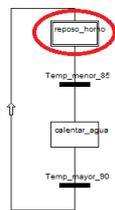
“Macroetapa Limpieza”-“Etapa LIMPIAR OUT”: Alcanzada esta etapa, la “macroetapa limpieza”finaliza, y el sistema continua por el diagrama SFC principal.



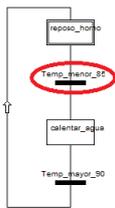
“Transición true”: Es una transición ligada a una variable que siempre ocurre. El sistema retorna al inicio, a la etapa “Reposo”.

4.3.4.2 Diagrama SFC calentador

El diagrama alternativo para el calentador consta de dos etapas y dos transiciones como las que acontecen a continuación:



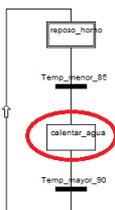
“Etapa reposo_horno”: Es una etapa que el sistema permanece estable hasta que no ocurra ninguna condición idónea.



“Transición menor 85”: Es una transición ligada a una sección. Se activa si el sistema está alimentado: “AC”, y si la temperatura es menor que 85°C: “T_85” y por lo tanto menor que 90°C: “T_90”, como muestra la Figura 73:



Figura 73: Transición T_85



“Etapa calentar Agua”: Es una acción ligada a una variable. En ella se activa la variable “R”., asociada al descriptor “N”, como muestra la Figura 74:

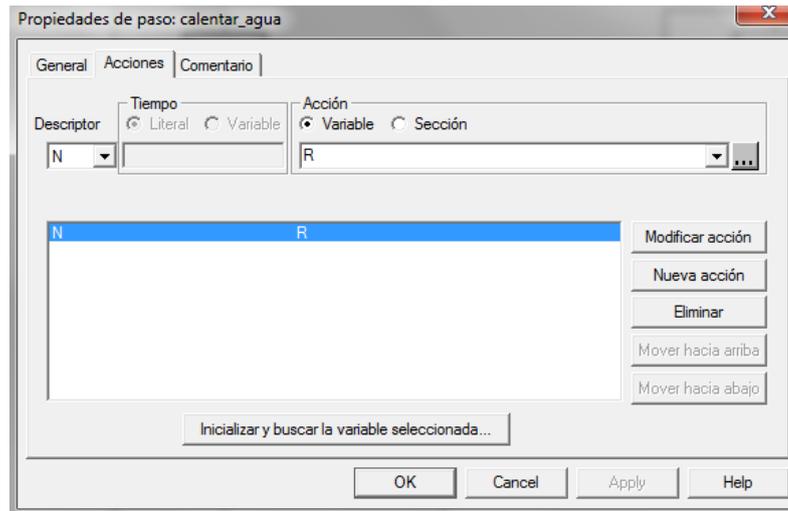
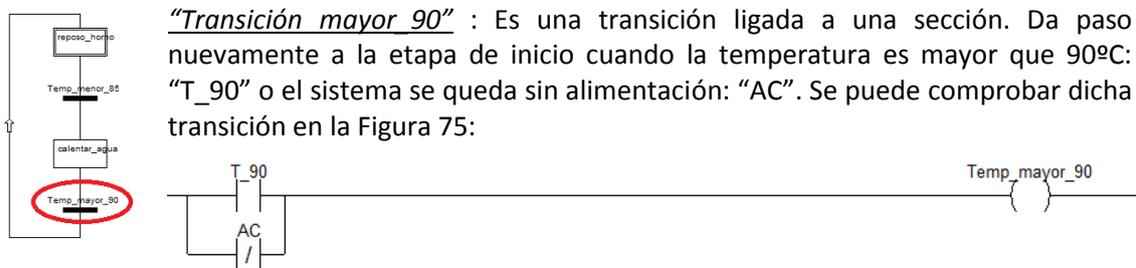


Figura 74:Etapa calentar_Agua



“Transición mayor 90” : Es una transición ligada a una sección. Da paso nuevamente a la etapa de inicio cuando la temperatura es mayor que 90°C: “T_90” o el sistema se queda sin alimentación: “AC”. Se puede comprobar dicha transición en la Figura 75:

Figura 75:_Transición mayor_90

4.3.5 Lenguaje ST

La principal característica del lenguaje estructurado (ST) es su gran potencia y versatilidad, siendo el más apropiado para funciones complejas y aplicaciones en las cuales sea necesario un programa optimizado. Es un lenguaje parecido al C, por tanto cualquier persona con conocimientos de programación informática es capaz de realizar aplicaciones complejas mediante el software Unity Pro.

En este apartado, se desea mostrar las pautas de programación en lenguaje ST. No se ha incluido la programación mediante diagrama SFC, ya que se ha intentado variar el método de programación y así ampliar los conocimientos en Unity Pro.

El primer paso antes de comenzar a programar en ST es conocer la nomenclatura y funciones propias del lenguaje, siendo esto la parte más compleja de la programación.

- Operador; Define la operación a realizar.
- Operando; Objeto sobre el que actúa el operador.
- Instrucción de Control; Orden del programa (ej. ifthenelse).
- Instrucción; Finalizado con punto y coma.
- Comentario; Información adicional cuya sintaxis es (*...*).
- Etiqueta; Indica un bloque de programa (acabado por dos puntos).

En la barra de herramientas del programa de la Figura 76, se encuentran todos los iconos para insertar en el código:

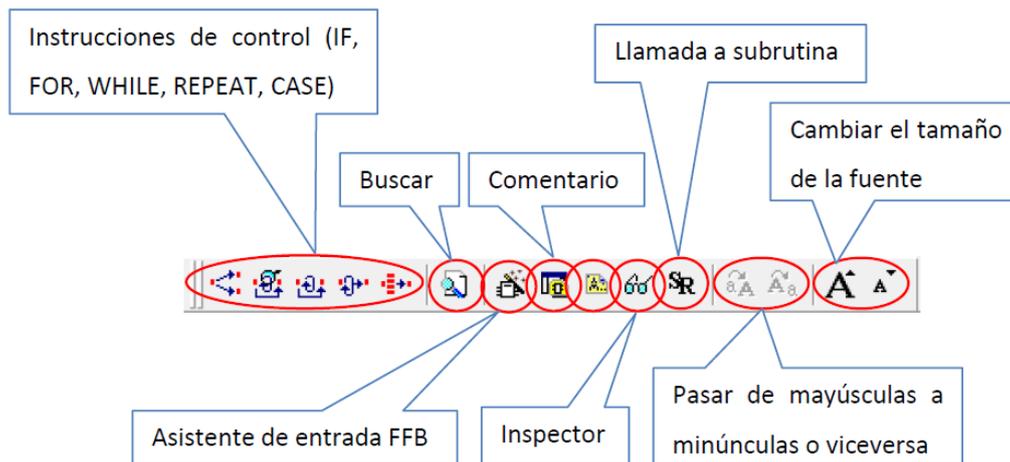


Figura 76: Alternativas de elementos para programación lenguaje ST

La parte más importante de la barra de herramientas son las instrucciones de control. Gracias a ellas podemos ir creando la aplicación paso a paso. A continuación quedan definidas:

- IF: La instrucción IF determina que una instrucción o un grupo de instrucciones se ejecuten sólo si la expresión booleana correspondiente tiene el valor 1. Si la condición es 0, la instrucción o el grupo de instrucciones no se ejecuta. "IF...THEN...END_IF,ELSE y ELSIF...THEN"
- FOR: La instrucción FOR repite una secuencia de instrucciones hasta la instrucción END_FOR. La cantidad de repeticiones se determina mediante el valor inicial, el valor final y la variable de control. "FOR...TO...BY...DO...END_FOR"
- WHILE: La instrucción WHILE provoca la ejecución repetitiva de una secuencia de instrucciones hasta que la expresión booleana correspondiente sea 0. Si la expresión es falsa desde el principio, no se ejecuta el grupo de instrucciones." WHILE...DO...END WHILE".
- REPEAT: La instrucción REPEAT provoca la ejecución repetitiva de una secuencia de instrucciones (al menos una vez) hasta que la condición booleana correspondiente sea 1. "REPEAT...UNTIL...END_REPEAT"
- CASE: La instrucción CASE está compuesta por una expresión del tipo datos INT (el "selector") y una lista de grupos de instrucciones. Cada grupo provisto de una etiqueta que está compuesta por uno o más números enteros (INT,DINT, UINT,UDINT) o rangos de valores enteros. Se ejecuta el primer grupo de instrucciones cuya etiqueta contenga el valor calculado del selector. De lo contrario, no se ejecuta ninguna de las instrucciones. "CASE...OF...END_CASE"

A partir de ahora, la programación en lenguaje ST, va a ser el centro de análisis. Este caso se va a desarrollar la programación para la elección del producto B. En la Figura 77 se observa el código completo que se necesita para una dosificación de producto B:

```
IF AC AND TB THEN

    IF N3_4 AND N_100 THEN Motor:=1;

        M3:=1;

        FBI_70 (IN := M3,
                PT := t#8s,
                Q => V4);

        IF V4 THEN Motor:=0;
        V4:=1;
        END_IF;
    END_IF;

    IF NOT N3_4 AND NOT N_100 AND V4 THEN
        LLenado_intercambiador:=1000;
        M4:=1;
        FBI_71 (IN := M4,
                PT := t#3s,
                Q => B3);

        IF B3 THEN M5:=1;
        V4:=0;
        LLenado_intercambiador:=0;
        END_IF;
        M5:=0;
        V6:=1;
    END_IF;

    IF V6 THEN Llenado_Tanque_B:=1000;
        FBI_72 (IN := V6,
                PT := t#4s,
                Q => B2);
    END_IF;

    IF B2 THEN V6:=0;
    END_IF;

END_IF;
```

Figura 77: Código ST Dosificación Producto_B

Siguiendo la programación, y observando la Figura 78 que incluye el código comentado, se observa que si se cumplen las condiciones de alimentación “AC” y producto B “TB”, se produce el llenado del depósito de mezcla hasta el valor del sensor 75% del aditivo A, y hasta el nivel del 100% del aditivo B. De esta manera, y alcanzadas las condiciones necesarias de 55 °C de temperatura, se produce la activación del motor para que se produzca la mezcla con la activación del motor, durante unos segundos. Cumplido este tiempo, el temporizador activa la variable V4 (válvula que da paso al vaciado del depósito de mezcla y llenado del intercambiador) y se da la instrucción de desactivación del motor.

```
(*PRODUCTO B*)

IF AC AND TB THEN                                     (* Si el usuario pulsa alimentación y producto se va
                                                       a realizar un ciclo de dosificación de producto B*)

    IF N3_4 AND N_100 THEN Motor:=1; (* Si además los sensores que indican los niveles del
                                       tanque de dosificación 66% para producto B y el resto
                                       hasta llegar a 100% para producto A están activados, se
                                       pone en marcha el motor*)

        M3:=1;                                       (*Se activa una variable de memoria M3*)

        FBI_70 (IN := M3,
                PT := t#8s,
                Q => V4);                             (*Se temporiza 3 segundos, pasado este tiempo se activa
                                                       V4,válvula de salida del depósito de mezcla*)

        IF V4 THEN Motor:=0; (*Si V4 está abierta, el motor se para*)
        V4:=1;
        END_IF;
    END_IF;
```

Figura 78: Primera parte Código ST Dosificación Producto B comentado

Cumplidas estas condiciones, sale del bucle, y continúa con la siguiente instrucción que corresponde al vaciado. Como está activada V4, el depósito de mezcla se va vaciando, y esto da el resultado que los sensores de nivel dejan de estar activos. Los primeros en notar este efecto son los sensores de 100% de nivel y 75%. Cuando no están ambos activos, el intercambiador de calor se empieza a llenar, se le asigna su valor máximo 1000 durante 3 segundos, pasado este tiempo, se abre la válvula de salida de líquido del intercambiador de calor. Mientras ocurre esto se da la instrucción de cerrar la válvula V4 y la variable llenado_intercambiador recibe el valor nulo y se activa la válvula V6 que es la del tanque B, pasado un tiempo se cierra dicha válvula y se completa el ciclo. A continuación observar la Figura 79:

```
IF NOT N3_4 AND NOT N_100 AND V4 THEN
  LLenado_intercambiador:=1000;
  M4:=1;
  FBI_71 (IN := M4,
  PT := t#3s,
  Q => B3);
  (*Cuando el depósito de mezcla empieza a vaciarse,
  el nivel del intercambiador se pone al máximo
  y esto dura un tiempo de 3 segundos.Pasado este tiempo, se
  abre la bomba de salida del intercambiador*)

  IF B3 THEN M5:=1;
    V4:=0;
    LLenado_intercambiador:=0;(*Cuando la bomba del intercambiador está abierta,
    se cierra la válvula del depósito de mezcla, y se
    abre la válvula V6 del depósito de vaciado de producto B *)

  END_IF;
  M5:=0;
  V6:=1;
END_IF;

IF V6 THEN Llenado_Tanque_B:=1000; (*El tanque del producto B se llena*)
  FBI_72 (IN := V6,
  PT := t#4s,
  Q => B2);
END_IF;

IF B2 THEN V6:=0;
END_IF;
(*Se cierra la válvula del tanque B*)
END IF;
```

Figura 79: Segunda parte Código ST Dosificación Producto B comentado

Finalizada la dosificación de Producto_B, se va a completar la explicación con un ciclo de limpieza. La Figura 80 muestra el código completo:

```
(*LIMPIEZA*)

IF AC AND L THEN V3

    IF N_100 THEN Motor:=1;

        M3:=1;

        FBI_70 (IN := M3,
                PT := t#8s,
                Q => V4);

        IF V4 THEN Motor:=0;
        V4:=1;
        END_IF;
    END_IF;

IF NOT N3_4 AND NOT N_100 AND V4 THEN

    LLenado_intercambiador:=1000;
    M4:=1;
    FBI_71 (IN := M4,
            PT := t#3s,
            Q => B3);
WHILE M5<=2 DO

    IF B3 THEN M5:=1;

        V4:=0;
        LLenado_intercambiador:=0;
    END_IF;
    M5:=0;
    V7:=1;
END_IF;

IF V6 THEN Llenado_Tanque_B:=1000;
    FBI_72 (IN := V7,
            PT := t#4s,
            Q => B2);
END_IF;

IF B2 THEN V7:=0;
END_IF;

END_IF;

END_WHILE;

END_IF;
```

Figura 80: Código ST Dosificación Producto_B

Para que se produzca la limpieza deben darse las condiciones de alimentación y haber pulsado botón L. Cuando el depósito de mezcla está lleno de agua indicado por el sensor de 100% de nivel, se activa el motor. Pasado un tiempo se abre la válvula V4 de descarga y se para el motor. Cuando la operación de vaciado está activa, se pone al máximo el nivel del intercambiador y se vuelve a temporizar otro tiempo, para a continuación activar la bomba de salida del intercambiador B3.

```
(*LIMPIEZA*)

IF AC AND L THEN V3 (*Se dan las condiciones de limpieza con la activación de L*)

  IF N_100 THEN Motor:=1; (*Cuando el depósito de mezcla esté lleno completamente, se activa el
                           motor durante un tiempo*)

    M3:=1;

    FBI_70 (IN := M3,
           PT := t#8s,
           Q => V4); (*Pasado ese tiempo, se para el motor y se abre válvula de salida*)

    IF V4 THEN Motor:=0;
    V4:=1;
    END_IF;
  END_IF;

IF NOT N3_4 AND NOT N_100 AND V4 THEN (*Durante el vaciado del depósito de mezcla,
                                       se llena el intercambiador de calor*)
  LLenado_intercambiador:=1000;
  M4:=1;
  FBI_71 (IN := M4,
         PT := t#3s,
         Q => B3);
```

Figura 81: Primera parte Código ST Limpieza comentado

Tras el vaciado del intercambiador de calor se llena el tanque de vaciado mediante la activación de la válvula V7, pasado un tiempo que sea suficiente para su llenado, se cierra la válvula V7. Este proceso se repite dos veces.

5. SIMULACIÓN DEL EJEMPLO DESARROLLADO

**Simulación de la automatización de
procesos con Unity Pro**

5 SIMULACIÓN DEL EJEMPLO PROGRAMADO

5.1 Pantallas del operador

5.1.1 Definición y creación

Para observar de manera gráfica la simulación de todo el proceso descrito anteriormente, existe en Unity Pro la posibilidad de representar mediante gráficos los procesos que están ocurriendo cuando está en marcha el simulador en el caso de trabajar de manera offline.

Una vez que se tiene creado el proyecto y programado el sistema que se quiere simular, el siguiente paso es crear una imagen de aquello que se ha programado.

Para conseguir la simulación de un sistema, se necesita un diseño gráfico, que ayude a interpretar de manera animada la evolución del sistema. Los elementos gráficos se crean mediante “Pantalla del operador”. Se trata de una herramienta que aporta Unity Pro, en la que el usuario puede crear pantallas, en las que es posible insertar objetos estáticos o dinámicos que pueden ser botones, indicadores, textos, números, barras o casilla de verificación. En definitiva son imágenes que desde una librería de pantallas de operador o desde el propio disco, se pueden ir insertando. El diseño es totalmente libre, no hay que seguir ninguna regla. Se basa en herramientas tradicionales como copiar, cortar, pegar y el ratón para desplazar los objetos o acceder a las propiedades (mediante un doble clic). De esta manera el manejo de dicha herramienta es muy fácil e intuitivo. La librería de operador permite utilizar numerosos objetos gráficos ya existentes que facilitan la creación de pantallas de operador.

La manera de crear una pantalla es desde la ventana de “Explorador de Proyecto”, seleccionando con el botón derecho sobre “Pantallas del Operador”, y nueva pantalla. La Figura 82 muestra la manera de realizar la operación:

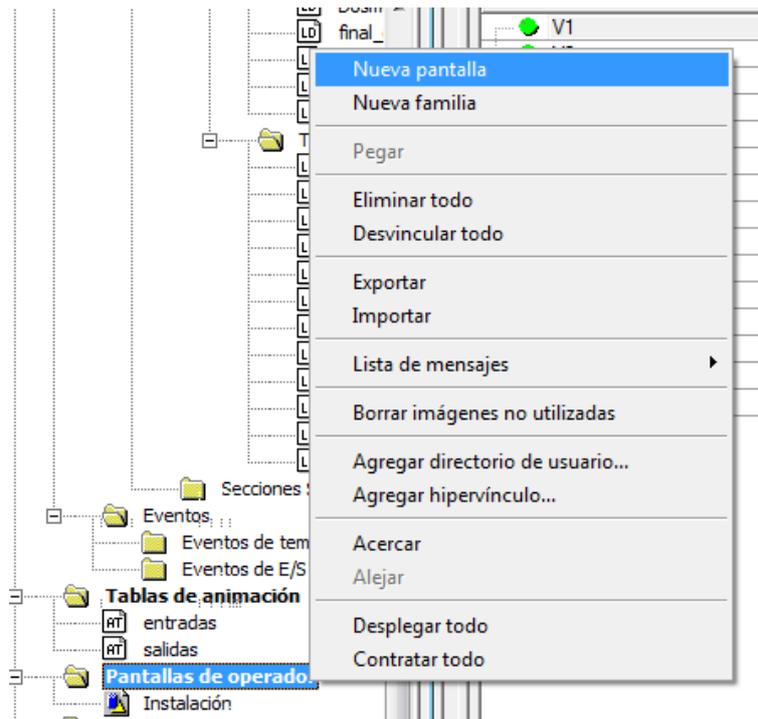


Figura 82: Creación Pantalla del operador

De esta forma aparecerá una pantalla en blanco preparada para ser rellena por los elementos necesarios. El aspecto final que muestra la pantalla que responde a la simulación de la instalación programada es la siguiente, mostrado en Figura 83:

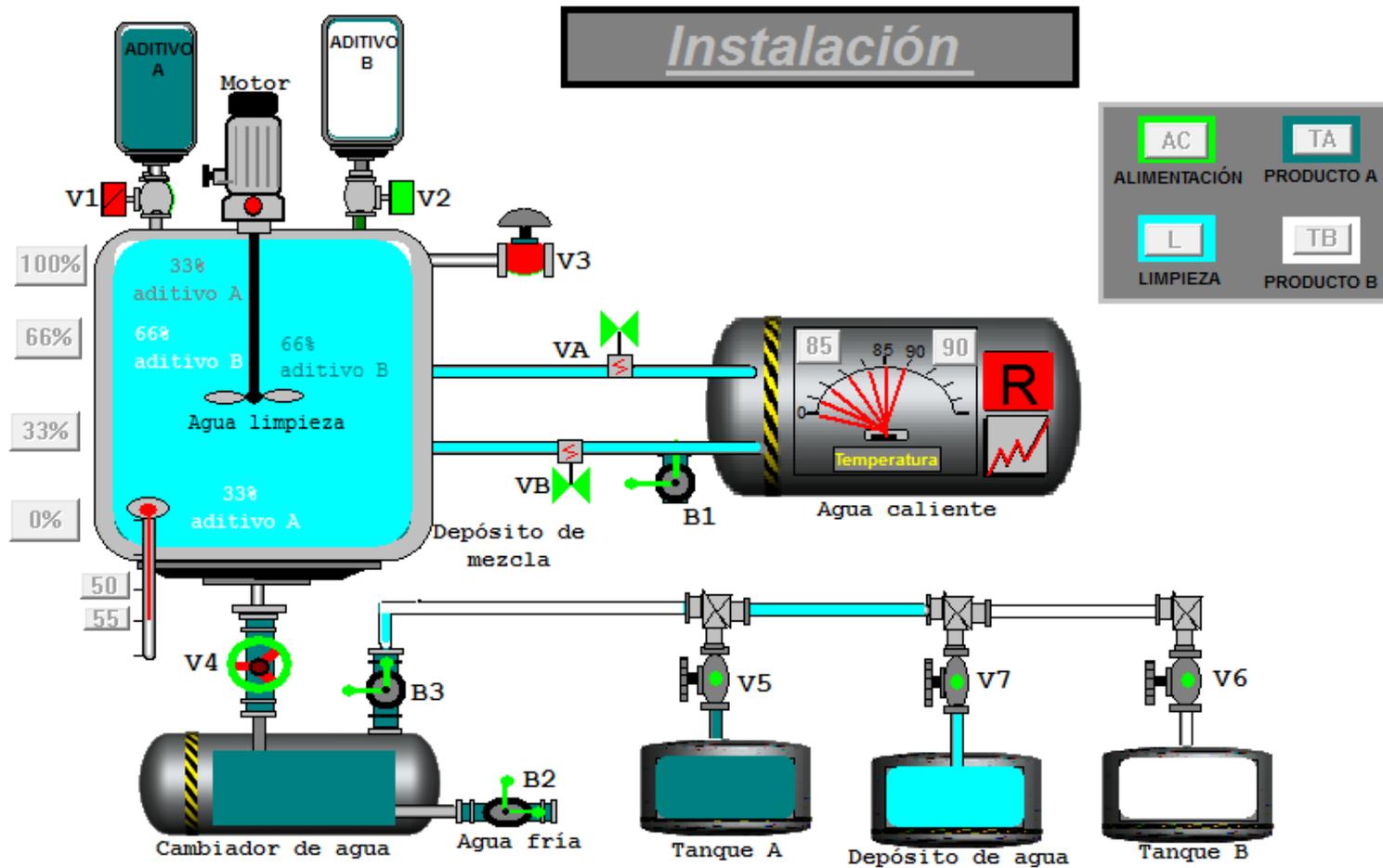


Figura 83: Gráfico de la Instalación

5.1.2 Librería y posibilidades en el desarrollo de elementos

A continuación se va a ir explicando la manera en que se va insertando cada elemento. Para añadir elementos animados, se puede hacer uso del conjunto de librerías que el software incorpora. Estas librerías, así como las familias, funciones y variables (estructuras de datos de E/S) se pueden utilizar para desarrollar un proyecto de automatización. También es posible añadir librerías nuevas, familias nuevas y gestionar las versiones de los bloques con la función de insertados en la librería. El acceso es mediante la barra de herramienta. Aparecerá una nueva ventana en la que se podrá desplegar conjuntos animados de distintas temáticas. Se muestra en la siguiente Figura 84, cómo acceder a la librería:

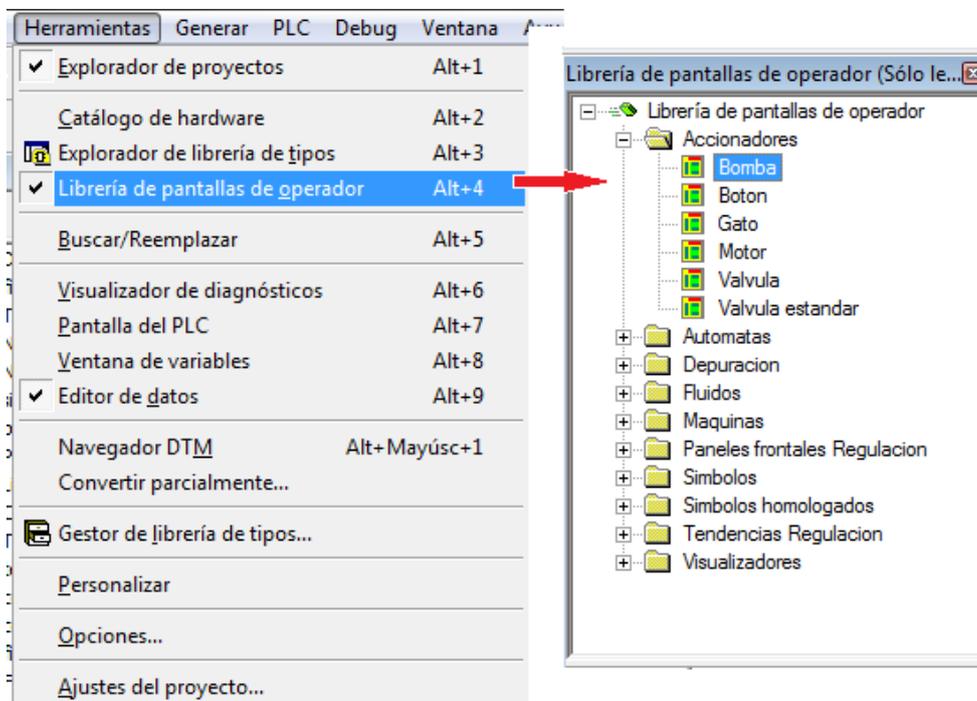


Figura 84: Librería de Pantalla del operador

El simulador de Unity Pro permite seleccionar dentro de un catálogo amplio, que es la "librería de pantallas del operador", numerosos objetos gráficos que pueden configurarse. Todos los objetos pueden desagruparse para configurar una animación a una parte del objeto. Es decir, si se selecciona un motor, se puede desagrupar para programar cada elemento que lo compone. Al desagruparlo, se obtiene cada parte: las aspas, el led de encendido, el tornillo sin fin que realiza el movimiento de giro...por separado, tal y como muestra la siguiente Figura 85:

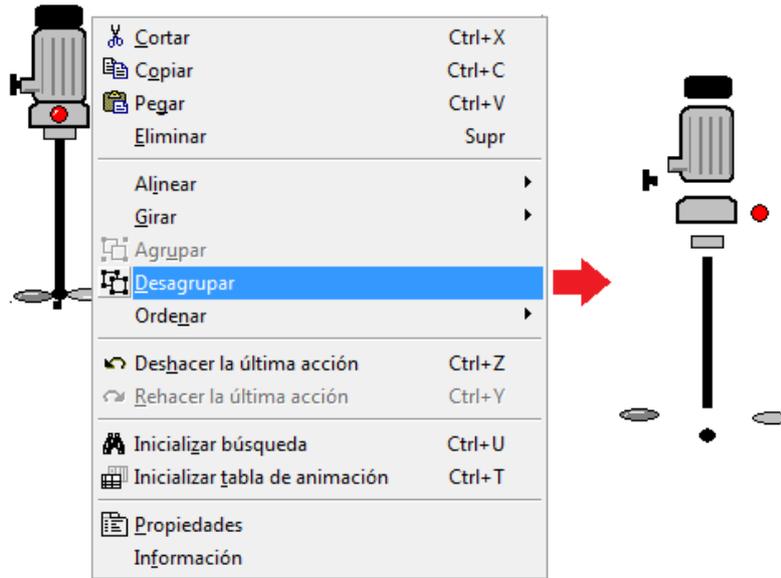


Figura 85: Desagrupar elemento motor

Cada uno de los elementos es el producto de la “agrupación” de figuras simples como círculos y polígonos de diferentes lados que en su conjunto dan vida a una figura. También existe la posibilidad de que el usuario combine a su necesidad un objeto desde formas simples y pulsando sobre el botón derecho para que las agrupe, tal y como muestra la siguiente Figura 86:

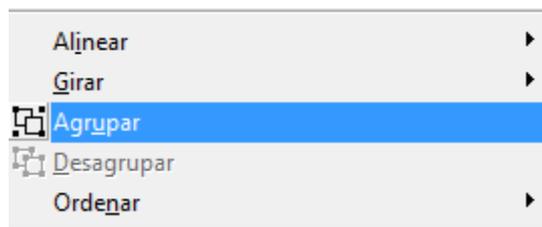
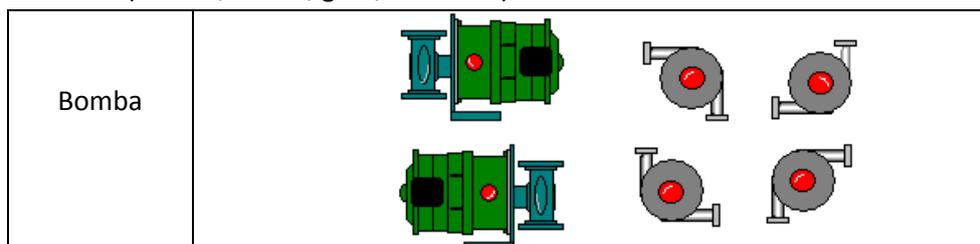
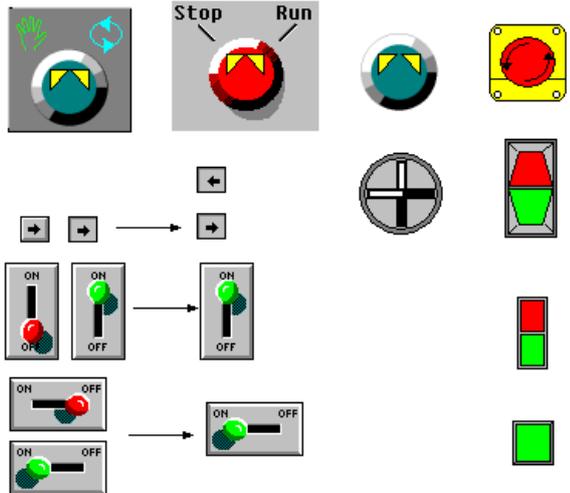
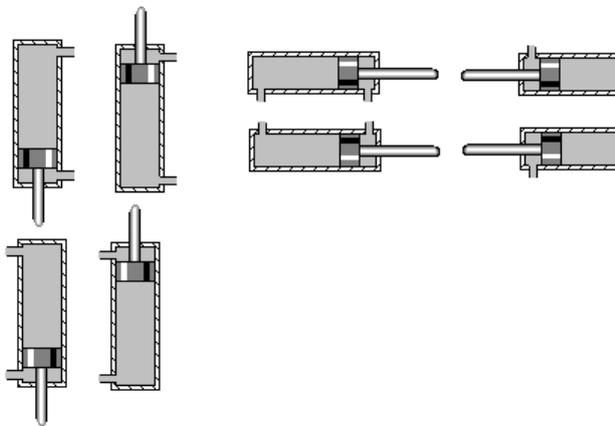
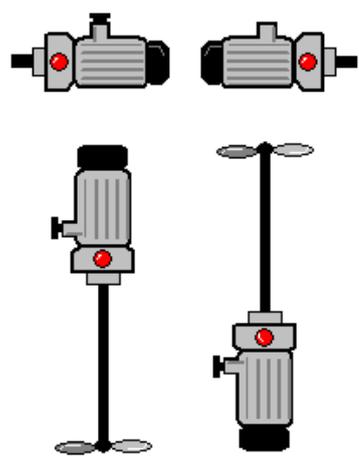


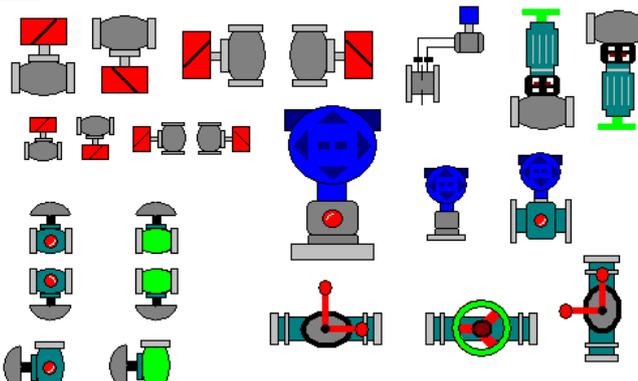
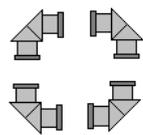
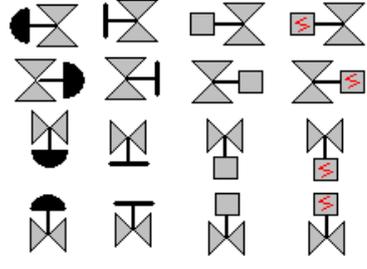
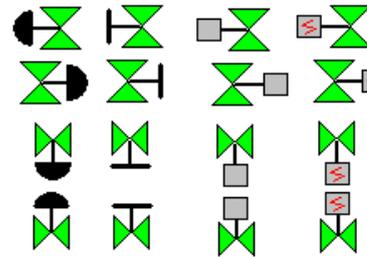
Figura 86: Agrupar figuras en Unity Pro

Entre todas las posibilidades que ofrece están algunos ejemplos registrados en la librería:

Accionamientos (bomba, motor, gato, válvula....)

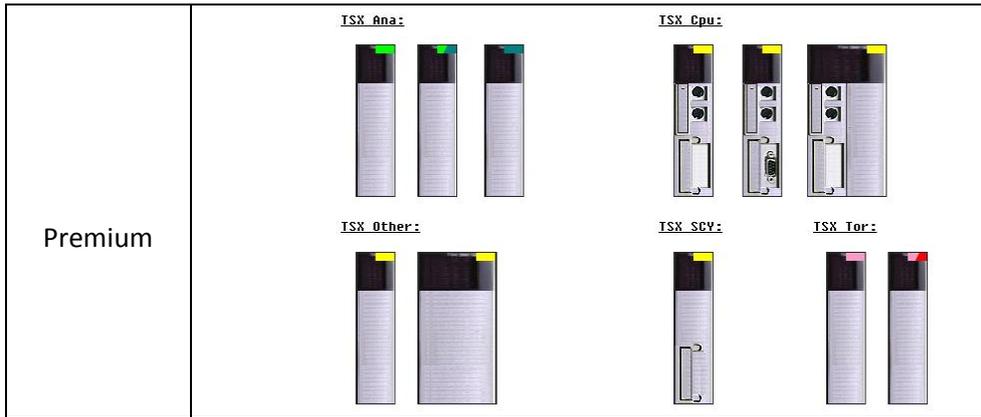


<p>Botón</p>	<p>Dynamic</p> 
<p>Gato</p>	<p>Dynamic</p> 
<p>Motor</p>	<p>Dynamic</p> 

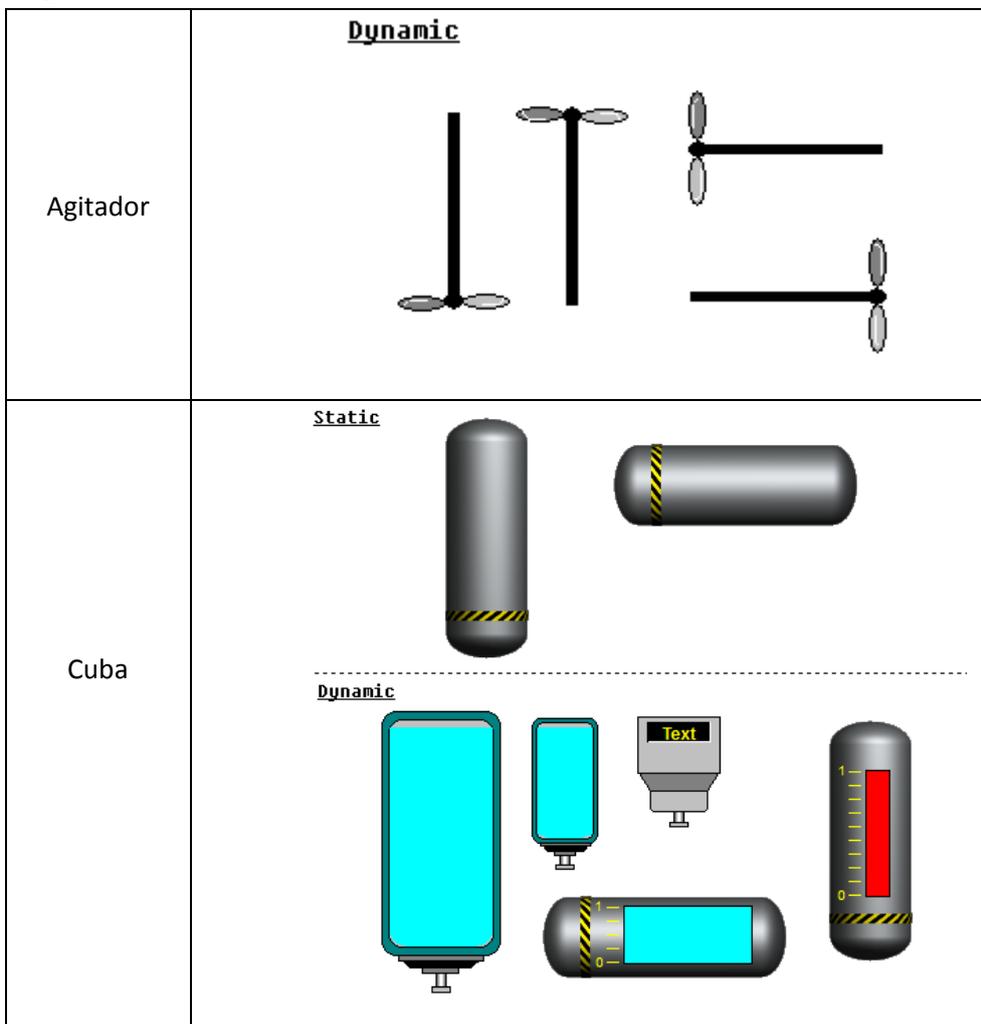
<p>Válvula</p>	<p>Dynamic</p>  <p>Static</p> 
<p>Válvula estándar</p>	<p>Static</p>  <p>Dynamic</p> 

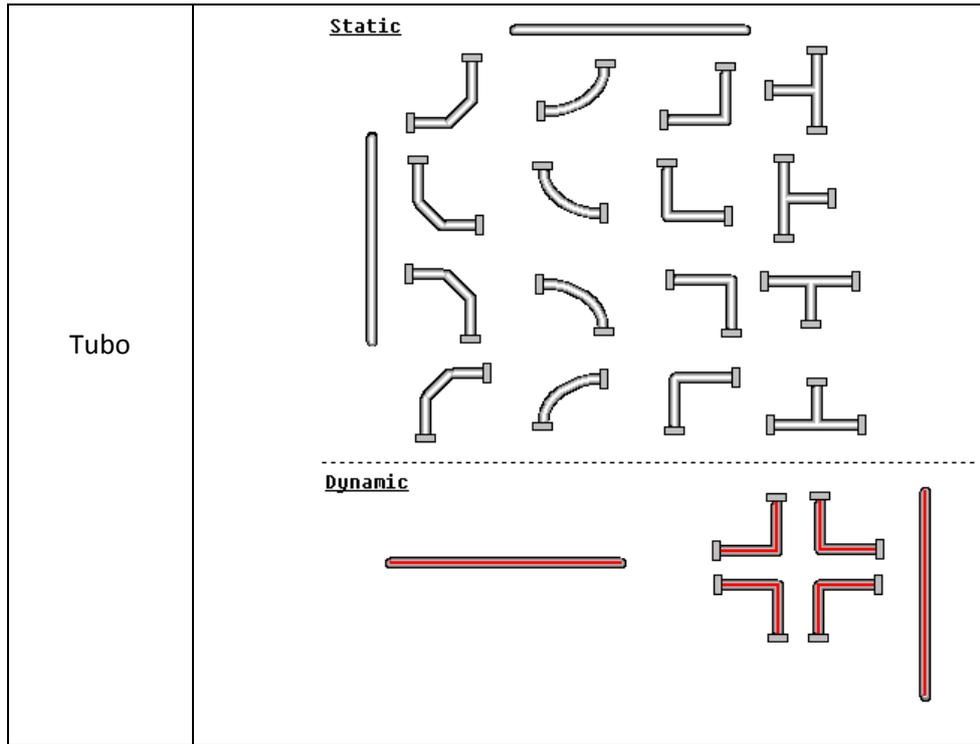
Autómatas (CANopen, CCX, Magelis, Micro, Modicon M340...)

<p>CANopen</p>	<p>Drives</p> <p>ATV31 ATV61 ATV71</p>  <p>Servo drives</p> <p>Ic1A Lexium 05</p>  <p>Remote I/Os</p> <p>Advantys FTB Advantys OTB</p>  <p>Safety module</p> <p>Preventa XPS MC</p>  <p>Absolute multi turns rotary encoder</p> <p>Gsicoder</p> 
<p>CCX</p>	<p>Static</p> 
<p>Modico M340</p>	<p>Modicon M340</p>  <p>Analog modules</p> <p>Inputs Inputs+ Outputs</p> <p> outputs</p>  <p>AC digital modules</p> <p>Inputs Inputs+ Outputs</p> <p> outputs</p>  <p>DC digital modules</p> <p>Inputs Inputs+ Outputs Inputs+ Outputs</p> <p> outputs relays</p> 

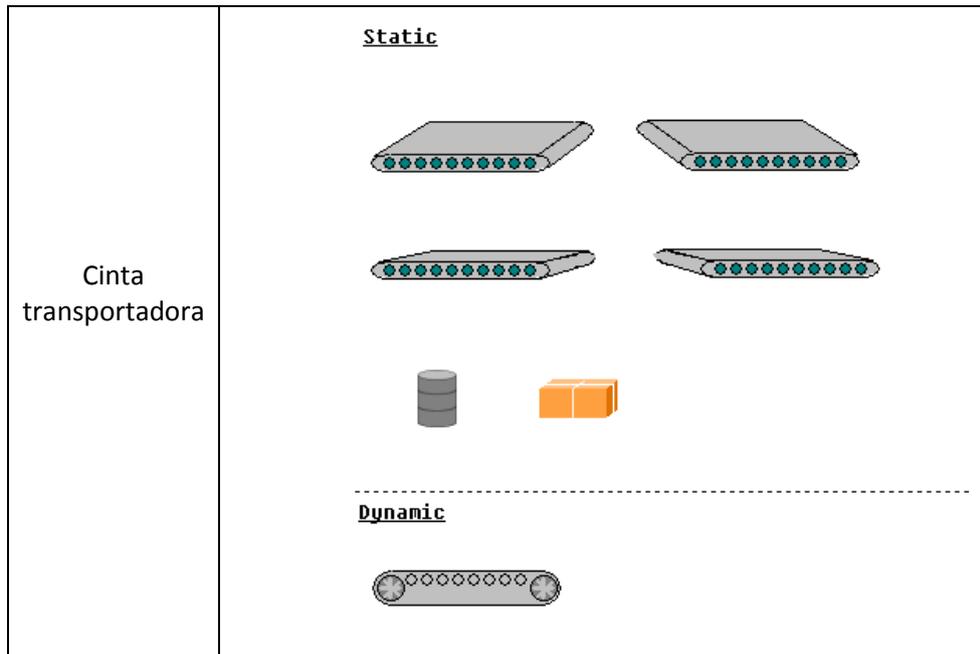


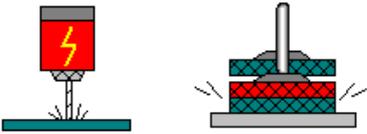
Fluidos (agitador, Cubo, Tubo...)



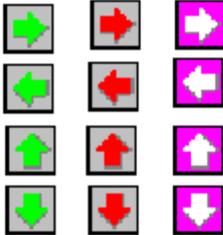


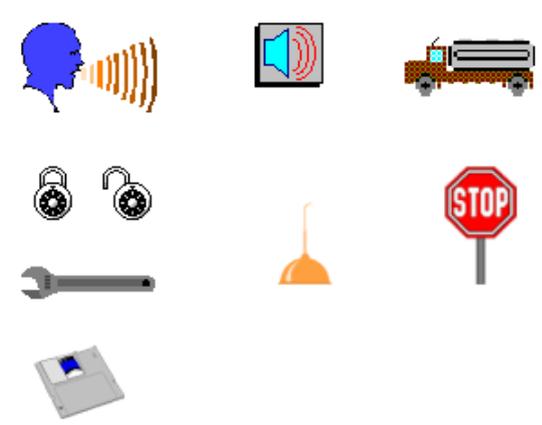
Máquinas (Cinta transportadora)



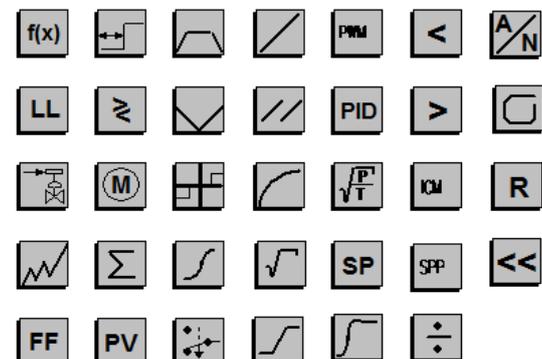
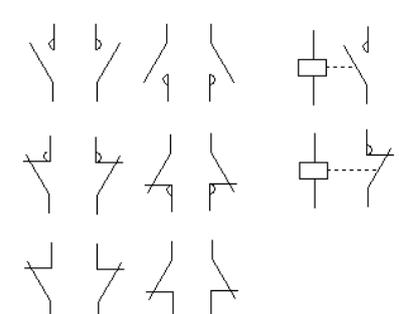
Máquina	<p style="text-align: center;"><u>Static</u></p> 
	<p style="text-align: center;">-----</p> <p style="text-align: center;"><u>Dynamic</u></p> 

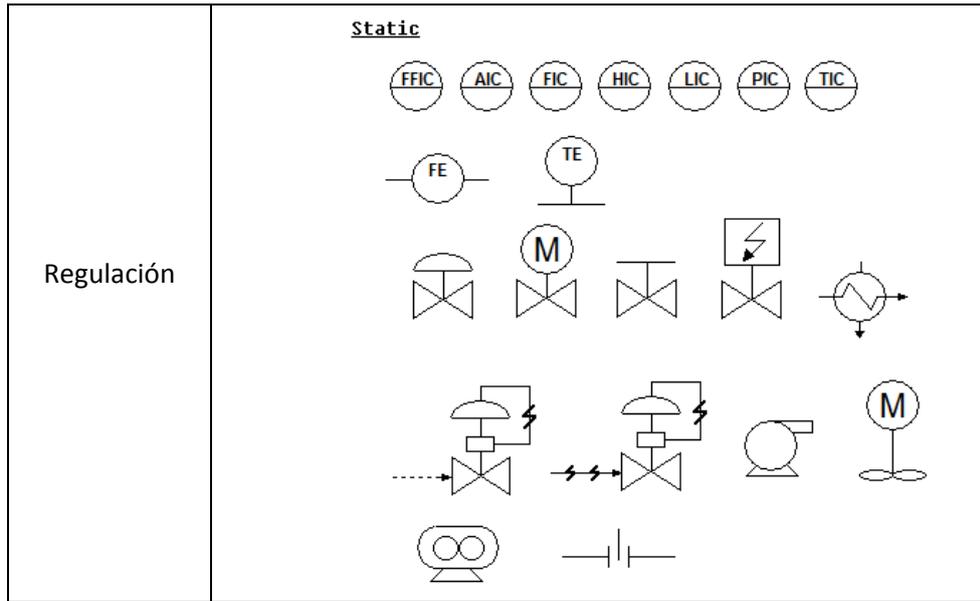
Símbolos (flecha, pictograma...)

Flecha	<p style="text-align: center;"><u>Static</u></p> 
Pictograma	<p style="text-align: center;"><u>Static</u></p>  <p style="text-align: center;">-----</p> <p style="text-align: center;"><u>Dynamic</u></p> 

Varios	<p style="text-align: center;">Static</p> 
--------	---

Bloques homologados, (de función, de regulación, eléctrico...)

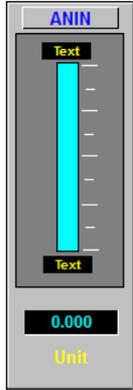
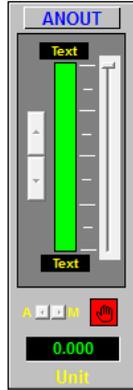
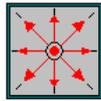
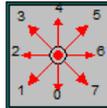
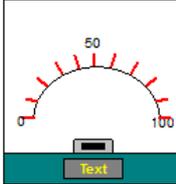
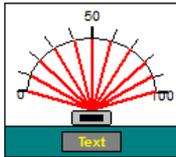
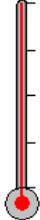
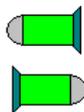
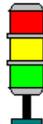
Bloque de función	<p style="text-align: center;">Static</p>  <p style="text-align: center;">Dynamic</p> 
Eléctrico	<p style="text-align: center;">Static</p> 

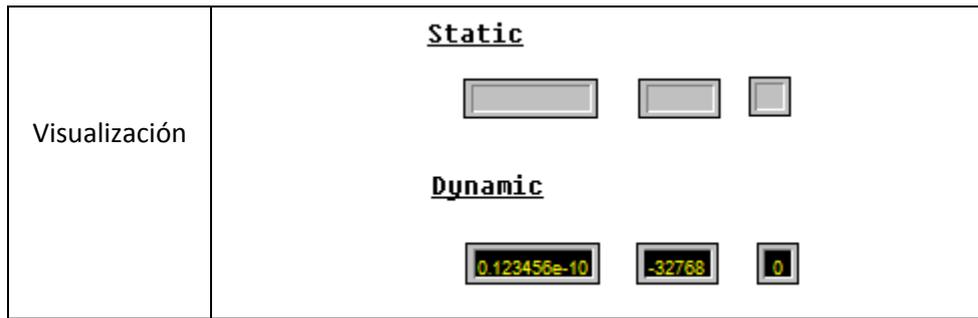


Tendencias de regulación (PID, IMC...)

IMC	<pre> %M0 : L/R state "_sts_l_r" %M1 : A/M state "_sts_m_a" %M2 : Alarms "_sts_alarms" %NF0 : PU "_pv" %NF2 : SP "_sp" %NF4 : OUT "_out_nan" %NF6 : PU_SUP "_pv_sup" %NF8 : PU_INF "_pv_inf" %NF10 : Alarm H "_pv_h" %NF12 : Alarm L "_pv_l" %NF20 : Ks "_ks" %NF22 : I Open loop "_o1_time" %NF24 : I Delay "_t_delay" %NF26 : Te "_t_ech" %NF28 : Perf "_c1_perf" %NF30 : DBAND "_dband" </pre>	<table border="1" style="margin-top: 10px;"> <tr> <td>Ks</td><td>0.000</td> <td>Te (s.)</td><td>0.000</td> <td>PV</td><td>0.000</td> <td>Unit</td><td></td> </tr> <tr> <td>Tdo (s.)</td><td>0.000</td> <td>Perf</td><td>0.000</td> <td>SP</td><td>0.000</td> <td></td><td></td> </tr> <tr> <td>Del (s.)</td><td>0.000</td> <td>DBand</td><td>0.000</td> <td>OUT</td><td>#XXXX</td> <td></td><td></td> </tr> </table>	Ks	0.000	Te (s.)	0.000	PV	0.000	Unit		Tdo (s.)	0.000	Perf	0.000	SP	0.000			Del (s.)	0.000	DBand	0.000	OUT	#XXXX		
Ks	0.000	Te (s.)	0.000	PV	0.000	Unit																				
Tdo (s.)	0.000	Perf	0.000	SP	0.000																					
Del (s.)	0.000	DBand	0.000	OUT	#XXXX																					

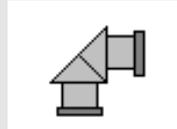
Visualizadores (indicador, medidor visual....)

<p>ANIN ANOUT</p>	<p style="text-align: center;"><u>Dynamic</u></p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">  <p>%MF2 : ANIN %MF6 : MAX %MF8 : MIN %MF10 : Alarm H %MF12 : Alarm L</p> </div> <div style="text-align: center;">  <p>%MF2 : ANOUT %MF6 : MAX %MF8 : MIN %MF10 : Alarm H %MF12 : Alarm L %M1 / %M5: A/M</p> </div> </div>
<p>Medidor visual</p>	<div style="display: flex; flex-wrap: wrap; justify-content: space-around;"> <div style="text-align: center;">  </div> <div style="text-align: center;">  </div> <div style="text-align: center;">  </div> <div style="text-align: center;">  </div> <div style="text-align: center; margin-top: 20px;">  </div> </div>
<p>Indicador</p>	<p style="text-align: center;"><u>Dynamic</u></p> <div style="display: flex; flex-direction: column; align-items: center; justify-content: center;">  <div style="display: flex; justify-content: space-around; width: 100%;">    </div> <div style="display: flex; justify-content: center; margin-top: 20px;">  <div style="display: flex; gap: 10px; margin-left: 20px;">   </div> </div> </div>

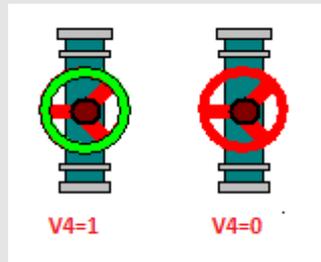


Cada uno de los anteriores conjuntos, están asociados a una temática que incorpora elementos estáticos y elementos dinámicos de dos tipos. A continuación una clasificación según el movimiento que se transmita:

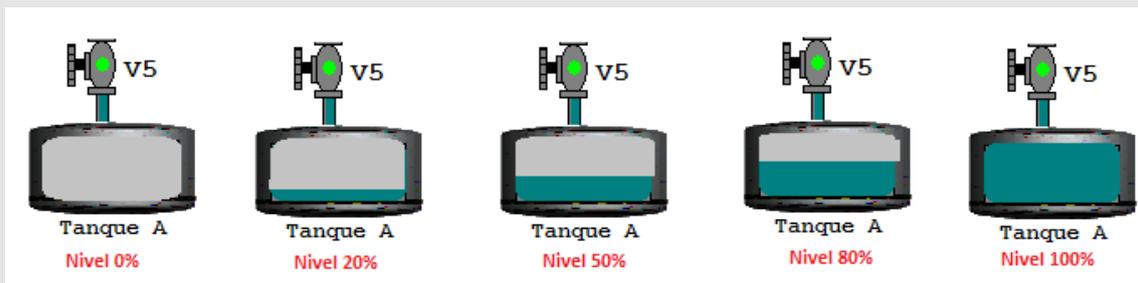
- Los **elementos estáticos**, como su nombre indica, aparecen en la simulación representando a un objeto concreto pero no ofrece ningún movimiento en su simulación. Puede aparecer durante todo el tiempo de la duración de la simulación, o si están asociados a una variable booleana, el máximo “dinamismo” (no olvidando que se habla de elemento estático), que puede ofrecer, es que aparezca cuando esté activa la variable y desaparezca cuando esté inactiva, o viceversa. Es el caso de una tubería. El elemento tubería es representado por el primer objeto que se muestra en la Figura 87, en donde no se produce ningún movimiento.
- Los **elementos dinámicos asociados a variable booleana**, producen movimiento en el transcurso de la simulación. El dinamismo, va a ir en función del objeto. El segundo de los elementos mostrados en la Figura 87, es una válvula. En ella se puede observar que se va a producir dos situaciones diferentes cuando la variable sea valor “FALSE” y cuando sea valor “TRUE”.
- Los **elementos dinámicos con variables tipo entero (INT)** asociada, van a producir un dinamismo más real. El tercer objeto de la Figura 87, representa un nivel de un tanque. El movimiento es la subida ascendente de nivel de capacidad del depósito.



Elemento estático tubería: no ofrece ningún movimiento.



Elemento dinámico, asociado a la válvula V4 de tipo EBOOL.
Cuando la variable está activa muestra color verde, significa válvula abierta. Cuando la variable está desactivada, la válvula está cerrada mostrando el color es rojo.



Elemento dinámico asociado a la variable Nivel de Tanque A, de tipo INT. El dinamismo consiste en que va subiendo el nivel de capacidad del tanque.

Figura 87: Tipos de elementos estáticos y dinámicos

Los objetos utilizados en la simulación del sistema desarrollado, son de estos tres tipos. La combinación entre ellos, hace un sistema final dinámico y progresivo.

Para explicar el término “sistema dinámico”, se va a seguir paso a paso el ejemplo desarrollado. Sin embargo, antes hay que recordar las variables que fueron creadas estrictamente para la simulación. En la Figura 88 aparecen y son todas aquellas que son de tipo INT:

Nombre	Tipo	Nombre	Tipo
● Agua_limpieza	INT	● Llenar_producto_A2	INT
● Calentar	BOOL	● Llenar_producto_B	INT
● cont	DINT	● Llenar_producto_B2	INT
● cont2	BOOL	● Nivel_A	INT
● cont_menor_2	BOOL	● Nivel_B	INT
● Deposito_lleno	BOOL	● prod_A	BOOL
● Deposito_vacio	BOOL	● Prod_B	BOOL
● Dosificador_A	INT	● Producto	DWORD
● Dosificador_B	INT	● R	EBOOL
● fin_demezcla	BOOL	● T_1min	BOOL
● Fin_llenado	BOOL	● Tanque_A	INT
● Limpieza	BOOL	● Tanque_B	INT
● llenado_intercambiador	INT	● Tanque_vaciado	INT
● Llenar_producto_A	INT	● temp_1min	BOOL

Figura 88: Variables Simulación

Además es muy importante tener en cuenta el significado de cada uno de los colores en la simulación. Cada objeto va a llevar etiquetas de colores que va a indicar si se encuentra vacío, lleno de aditivo A, de aditivo B o de agua. Por tanto, el color de cada elemento, indica un estado diferente en cada momento. La Tabla 1 contiene el resumen de colores:

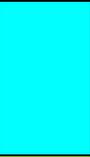
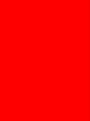
Nombre color	Color	Etiqueta	Objeto afectado	Significado
Verde oscuro		Aditivo A Producto A	Tanque Aditivo A Depósito de mezcla Cambiador de agua Tanque A	Nivel de llenado
Blanco		Aditivo B Producto B	Tanque Aditivo B Depósito de mezcla Cambiador de agua Tanque B	Nivel de llenado
Gris		El fondo de material ("metálico") de cada elemento	Tanque Aditivo A/B Depósito de mezcla Cambiador de agua Tanque A/B Depósito de agua	El objeto está vacío: exento de cualquier líquido
Azul claro		Agua	Depósito de mezcla Cambiador de agua Depósito de agua	Nivel de llenado
Verde claro		Abierto Activado ON	Válvulas, Bombas y Motor	Válvula abierta Bomba encendida Motor en marcha
Rojo		Cerrado Desactivado OFF	Válvulas, Bombas y Motor	Válvula cerrada Bomba apagada Motor parado

Tabla 1: Significado colores en la simulación

5.1.3 Configuración del ejemplo desarrollado

Con la combinación de todos los objetos de la librería, se puede llegar a la configuración que muestra la Figura 89:

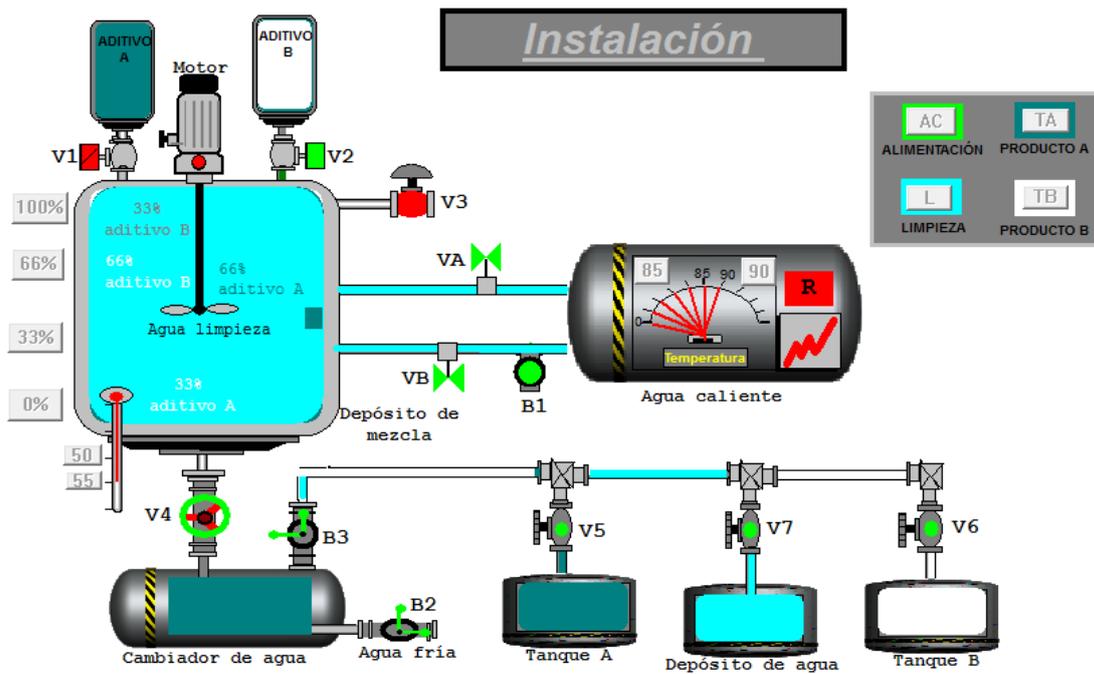


Figura 89: Instalación

Cada uno de los colores va a ir cambiando a lo largo del proceso, indicando un estado diferente de cada uno de los objetos. La Figura 89 intenta mostrar cómo se puede crear un sistema en dos dimensiones de una estación real. Es decir, las posibilidades que ofrece Unity Pro en la combinación de objetos para representar un sistema completo de tres dimensiones. No hay que olvidar que la Figura 89 es una figura estática plana, y que hay etiquetas de la Tabla 1 se encuentran solapadas unas sobre otras. El sentido de los colores aparece durante la simulación, cuando el sistema está en proceso. La obtención de una foto de un “vídeo” puede llevar a confusiones en los colores. No obstante, a lo largo del capítulo, se mostrará una serie de imágenes que puedan detallar con claridad paso a paso el desarrollo de la simulación, de duración cuatro minutos aproximadamente.

Para proceder a la explicación de cómo se han creado cada una de las etiquetas comentadas en el dibujo de la Figura 89, se va a ir analizando los objetos por separado:

➤ Cubas dinámicas: Tanque aditivo A, Tanque aditivo B y Depósito de mezcla

En toda la simulación predomina la utilización de cubas dinámicas. Tanto para los tanques, donde se encuentran los aditivos A y B, como para el depósito de mezcla se han utilizado dichos elementos encontrados en la librería fluidos. Cada uno de ellos lleva asociada una variable tipo entero "INT" para el indicador del nivel, mostrado anteriormente en la Figura 86.

De todas las variables INT, las cuatro siguientes se utilizan en **etiquetas creadas en los tanques de aditivos A y B**:

- "Dosificador_A"
- "Dosificador_B"
- "Nivel_A"
- "Nivel_B"

El Producto A, está formado de 33% de Aditivo A y 66% de Aditivo B.

Para el producto A, el aditivo A es identificado con una etiqueta verde oscuro a la que se le asocia la variable "**Dosificador_A**", y el aditivo B es identificado con otra etiqueta verde oscuro que se le asocia la variable "**Dosificador_B**".

El Producto B, está formado por 75% de Aditivo A y 25% de Aditivo B.

Para el Producto B, el aditivo A es identificado con una etiqueta blanca a la que se le asocia la variable "**Nivel_A**", y el aditivo B es identificado con otra etiqueta blanca que se le asocia la variable "**Nivel_B**".

La Figura 90, recoge la relación que existe entre las cuatro variables INT:



Figura 90: Variables Simulador para Tanques de Aditivos

Además, las cuatro variables tipo INT siguientes, se utilizan en **etiquetas creadas para el depósito de mezcla**:

- "Llenar_producto_A"
- "Llenar_producto_B"
- "Llenar_producto_A2"
- "Llenar_prodcuto_B2"

Como el Producto A, está formado de 33% de Aditivo A y 66% de Aditivo B, para este producto, el depósito de mezcla es identificado hasta el 33% con una etiqueta verde oscuro a la que se le asocia la variable "**Llenar_producto_A**", y para el 66% restante con una etiqueta de color verde oscuro a la que se le asocia la variable "**Llenar_producto_B**".

Como el Producto B, está formado por 75% de Aditivo A y 25% de Aditivo B, para este producto, el depósito de mezcla es identificado hasta el 75% con una etiqueta blanca a la que se le asocia la variable "**Llenar_prodcuto_B2**", y para el 25% restante con una etiqueta blanca a la que se le asocia la variable "**Llenar_producto_A2**".

La Figura 91, recoge la relación que existe entre las cuatro variables INT:

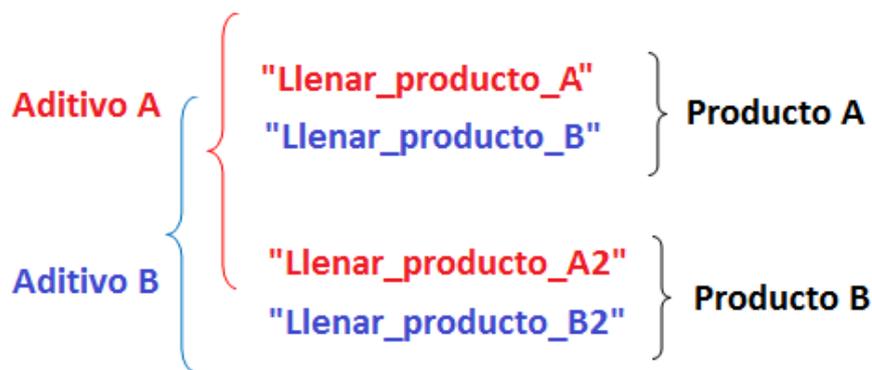


Figura 91: Variables Simulador para Depósito de mezcla

Si se elige "Producto A":

- Para el 33% de aditivo A, hay que activar las etiquetas asociadas a la variable "**Dosificador_A**" para el tanque de aditivo A, y a la variable "**Llenar_producto_A**" para el depósito de mezcla, como muestra la Figura 92.
- Para 66% de aditivo B, hay que activar las etiquetas asociadas a la variable "**Dosificador_B**" para el tanque de aditivo B, y a la variable "**Llenar_producto_B**" para el depósito de mezcla, como muestra la Figura 93.

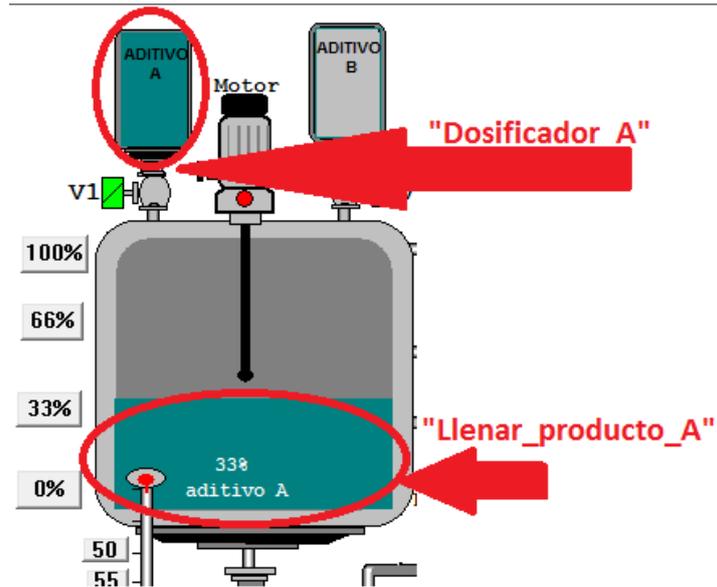


Figura 92: Variables INT "Dosificador_A" y "Llenar_producto_B"

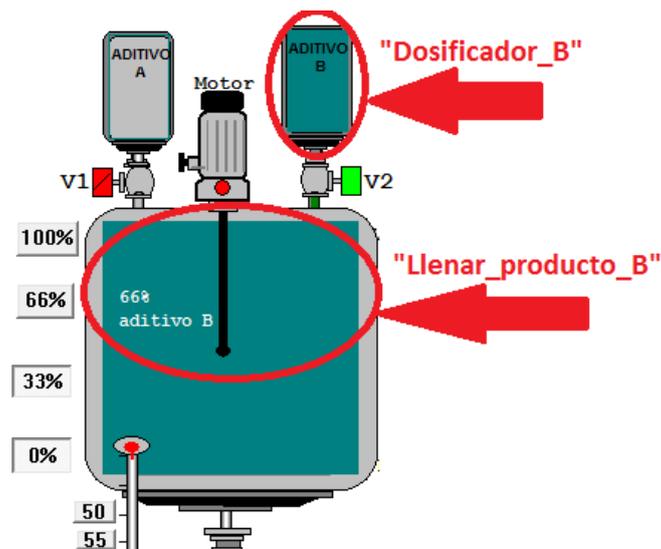


Figura 93: Variables INT "Dosificador_B" y "Llenar_producto_B"

Si se elige "Producto B":

- Para el 75% de aditivo A, hay que activar las etiquetas asociadas a la variable "Nivel_A" para el tanque de aditivo A, y a la variable "Llenar_producto_B2" para el depósito de mezcla, como muestra la Figura 94.
- Para 25% de aditivo B, hay que activar las etiquetas asociadas a la variable "Nivel_B" para el tanque de aditivo B, y a la variable "Llenar_producto_A2" para el depósito de mezcla, como muestra la Figura 95.

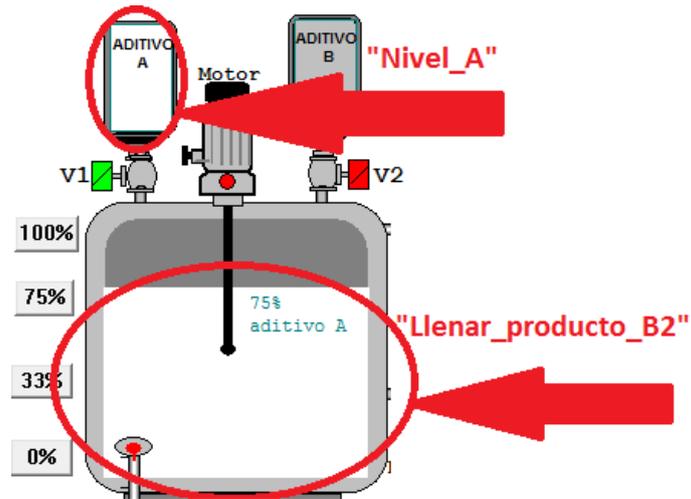


Figura 94: Variables INT "Nivel_A" y "Llenar_producto_B2"

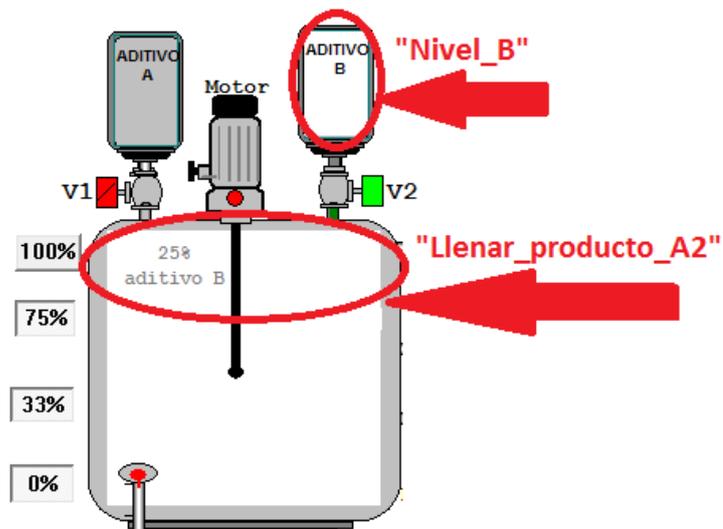


Figura 95: Variables INT "Nivel_B" y "Llenar_producto_A2"

Para simular el agua en el depósito de mezcla, se ha elegido una etiqueta de color azul claro asociada a la variable "Agua_limpieza" como muestra la Figura 96:

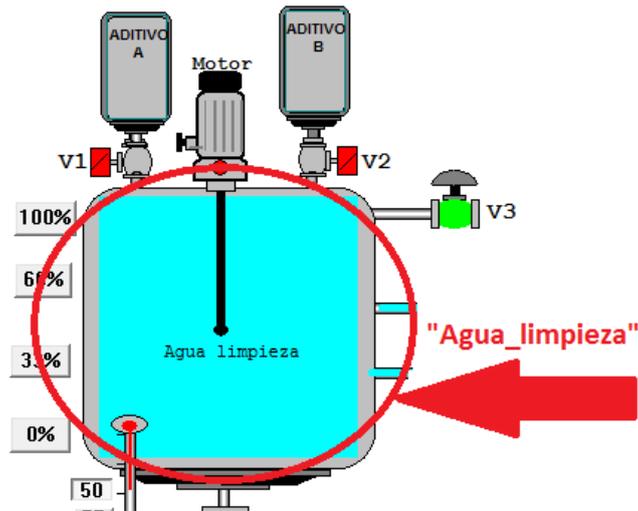


Figura 96: Variable INT "Agua_limpieza"

La manera de programar éstas etiquetas con las variables es como sucede a continuación:

Haciendo doble clic sobre uno de los objetos dinámicos como son las tanques de los aditivos o el depósito de mezcla, aparece el cuadro mostrado en la Figura 97:

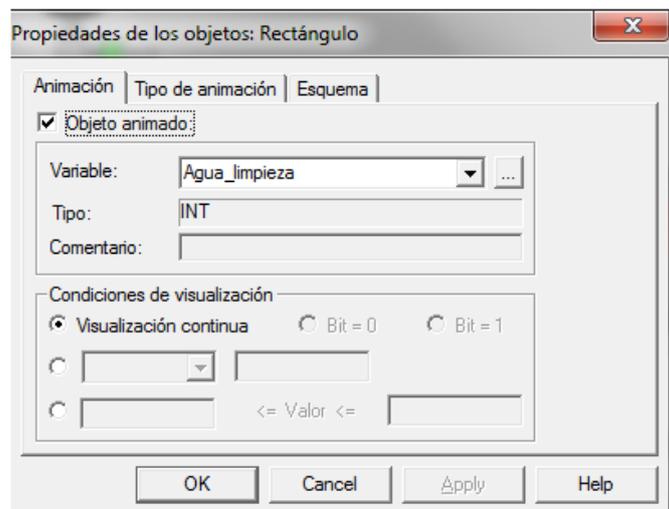


Figura 97 : Propiedad del objeto: "pestaña Animación"

En la pestaña "Animación", se marca "Objeto animado" y se indica mediante el menú desplegable aquella variable ya creada que se haga referencia. Se indica qué tipo de variable es. Y además se hace clic en la condición de visualización deseada. En el caso del ejemplo de la Figura 97, se ha elegido "Visualización continua".

En la siguiente pestaña, “*Tipo de animación*” se elige la función “Gráfico de barras”, y pulsando sobre la flecha aparece la imagen de la derecha. En ella, se decide “*Tipo de gráfico de barras*”, los “*umbrales*”, los “*valores y colores predeterminados*”, como muestra la Figura 98:

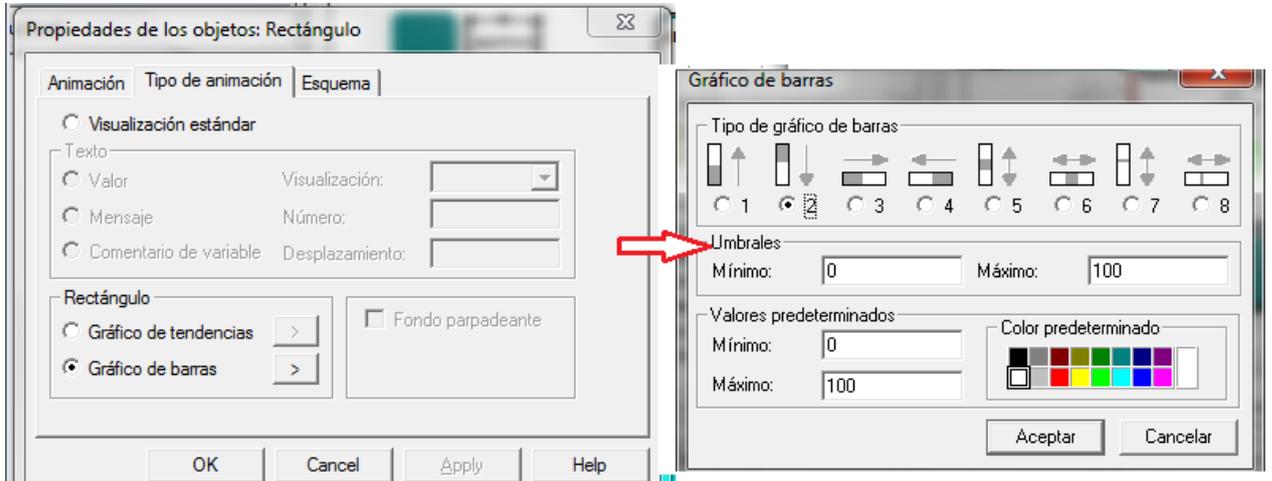


Figura 98 : Propiedad del objeto: “pestaña Tipo de animación”

Para finalizar con este nivel, en la pestaña “*Esquema*”, mostrada en la Figura 99, se eligen cuestiones más estéticas sobre los colores del fondo, de la línea o la forma del objeto.

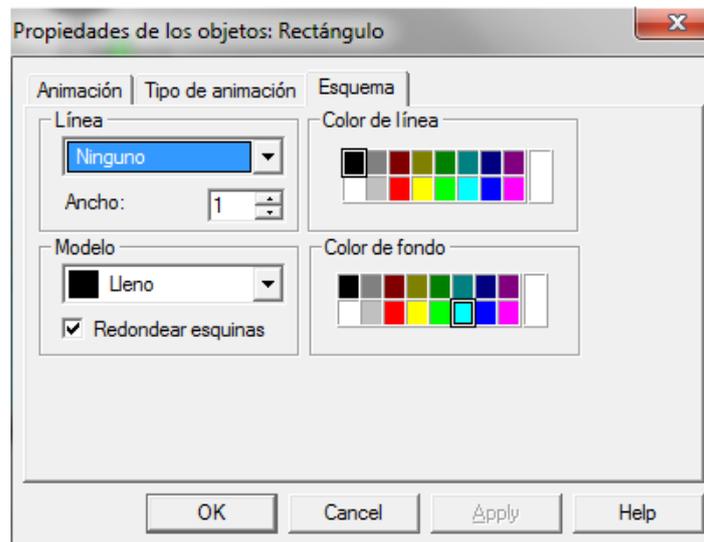


Figura 99: Propiedad del objeto: “pestaña Esquema”

➤ [Otras cubas dinámicas: Tanque A, Tanque B y Depósito de agua](#)

Como ya se ha comentado anteriormente, en toda la simulación predomina la utilización de cubas dinámicas. Tanto para los tanques donde se depositan Producto A y Producto B, como para el depósito de agua tras la limpieza, también se han utilizado dichos elementos encontrados en la librería fluidos. De la misma manera que las cubas dinámicas anteriores, cada uno de ellos lleva asociada una variable tipo entero "INT" para el indicador del nivel, mostrado anteriormente en la Figura 88.

De todas las variables INT, las tres siguientes se utilizan en etiquetas creadas en el Tanque A, Depósito de agua y Tanque B respectivamente:

- "Tanque_A"
- "Tanque_vaciado"
- "Tanque_B"

Las Figuras 100, 101 y 102 muestran las etiquetas asociadas a cada una de estas variables en el tanque correspondiente:

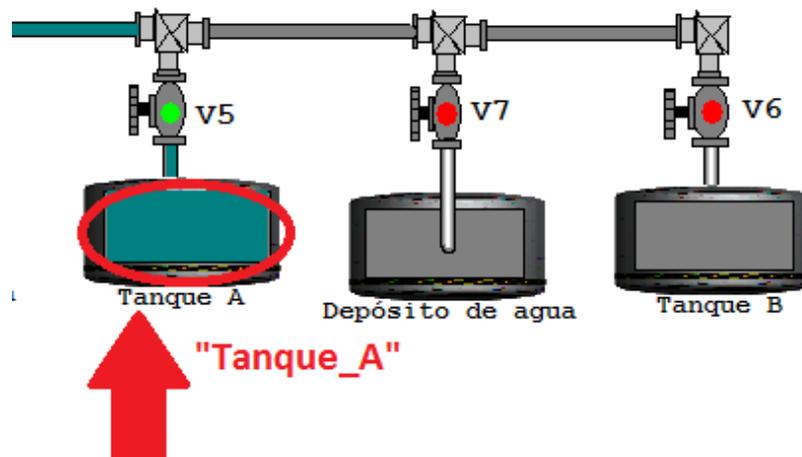


Figura 100: Variable INT "Tanque_A"

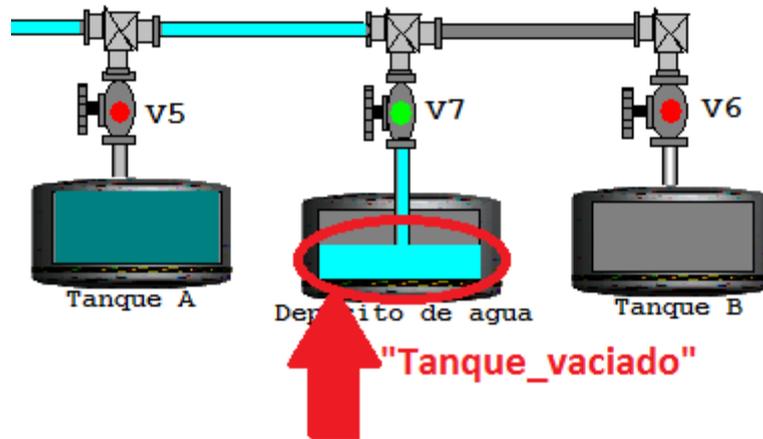


Figura 101: Variable INT "Tanque_vaciado"

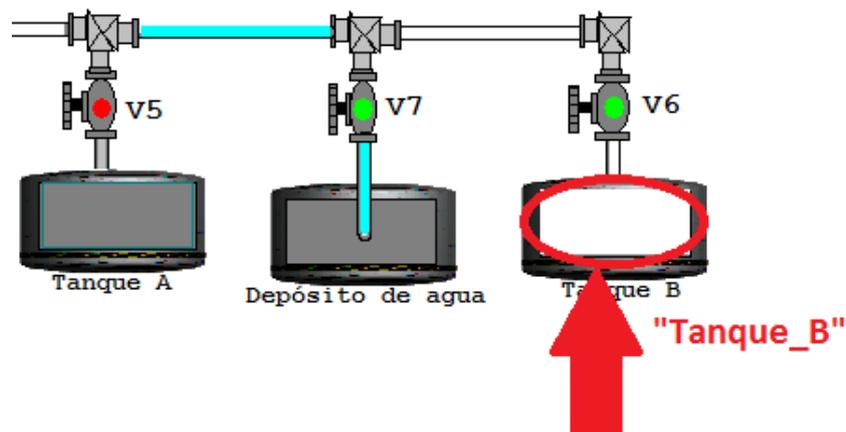


Figura 102: Variable INT "Tanque_B"

Para programar éstas etiquetas con la variable INT correspondiente, se realiza de la misma manera que en las anteriores cubas dinámicas explicadas. Los pasos a seguir se muestran en las Figuras 97, 98 y 99.

➤ Cambiador de agua

El cambiador de agua es una cuba dinámica, pero su explicación se realiza aparte, pues merece una atención especial. Aunque en un primer momento pudiera parecer lógico utilizar una variable tipo INT para su programación, no es posible porque su estado debe cambiar según pase por él el producto A (color verde oscuro), el producto B (color blanco) o el agua (color azul claro). Se necesitan cuatro estados diferentes (vacío y los tres tipos de líquidos) para un mismo objeto de intercambiador de agua. Se han creado tres etiquetas en las que cada una de ellas lleva asociada una variable de salida creada para la programación. A continuación se desglosa qué variable va asociada con cada etiqueta de color.

- Para la etiqueta verde oscuro, se le asocia la variable "V5", para el paso de producto A.
- Para la etiqueta azul claro, se le asocia la variable "V7", para el paso de agua.
- Para la etiqueta blanca, se le asocia la variable "V6", para el paso de producto B.
- Cuando las etiquetas no aparecen, el estado del intercambiador es de color gris, y por tanto vacío.

En la Figura 103, se muestra los colores y variables asociadas a cada una de las etiquetas:

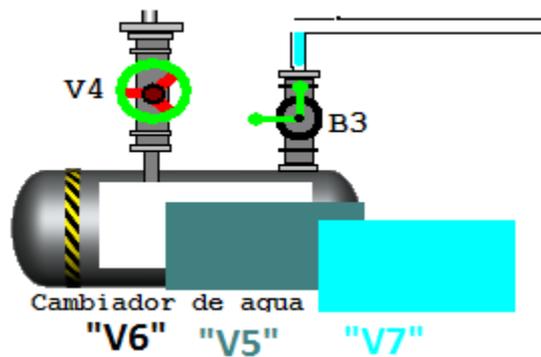


Figura 103: Etiquetas para el cambiador de agua

Para programar el cambiador de agua se ha realizado de la manera que se explica a continuación:

Si se hace doble clic sobre una de ellas, aparece la pantalla que aparece en la Figura 104:

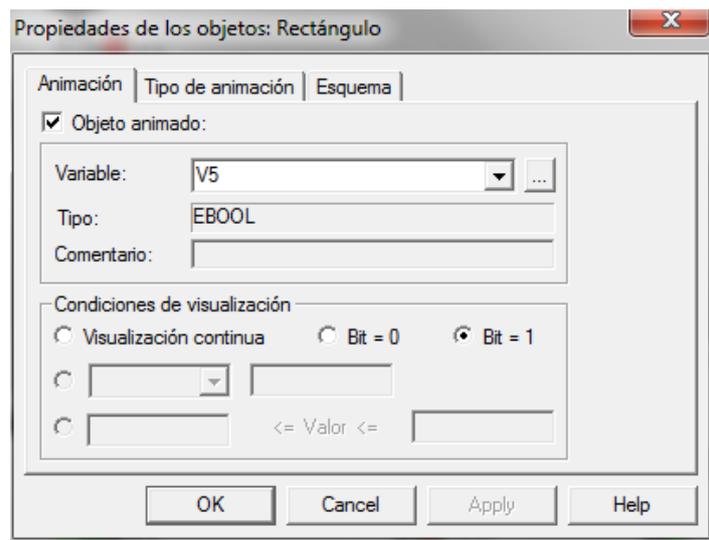


Figura 104: Propiedades de los objetos: pestaña "Animación"

En la pestaña “Animación”, se selecciona “Objeto animado”, la variable mediante el menú desplegable o escribiéndola y el “Tipo” de variable que es. También se elige que se muestre cuando el bit está a “1”, para que siempre que esté activa la variable “V5” de la Figura 104, permanezca visible el objeto.

Si se marca el bit a “0” ocurrirá lo contrario, el objeto se visualizará cuando la variable “V5” permanezca inactiva. Si se pulsa “visualización continua”, la variable asociada permanecerá visible durante toda la simulación.

La siguiente pestaña es la denominada “Tipo de animación”, y mostrada en la Figura 105:

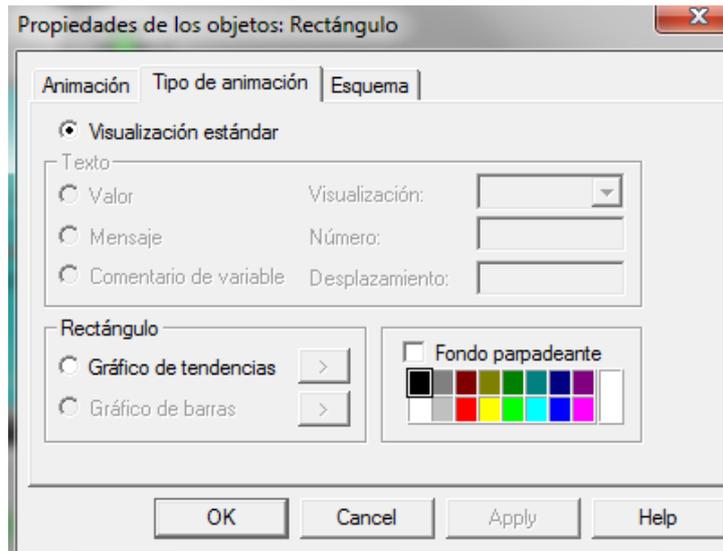


Figura 105: Propiedades de los objetos: pestaña “Tipo de animación”

En ella, hay que señalar que es “Visualización estándar” y el color deseado.

Para finalizar de completar las “Propiedades de los objetos”, en la Figura 106 se observa como en la pestaña “Esquema” se seleccionan cuestiones estéticas de colores y formas:

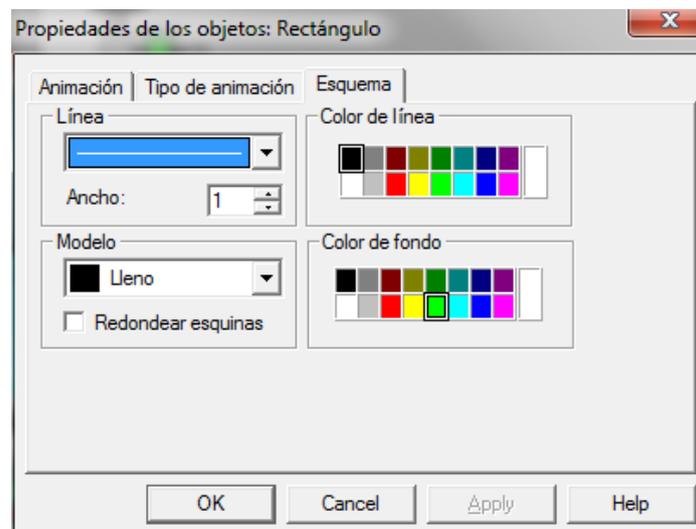


Figura 106: Propiedades de los objetos: pestaña “Esquema”

Finalmente, se han unificado todas las etiquetas haciendo doble clic sobre ellas unas vez alineadas correctamente, y se han agrupado.

➤ Válvulas

Cada una de las válvulas, "V1", "V2", "V3", "V4", "V5", "V5", "V6", "V7", "VA" y "VB" indica mediante color rojo que se encuentran cerradas, o mediante color verde claro que se encuentran abiertas. Estas variables han sido creadas tanto para la programación como para la simulación, y su aspecto se observa en la Figura 107:

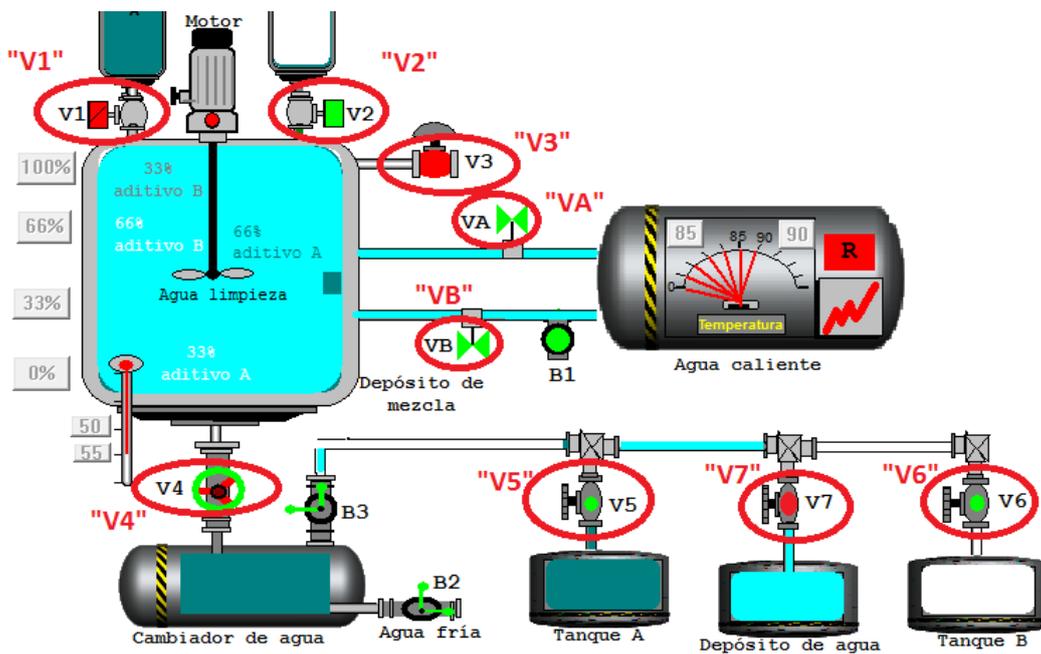


Figura 107: Variables "V1", "V2", "V3", "V4", "V5", "V5", "V6", "V7", "VA" y "VB"

La Figura 108, muestra el cambio de color de la Válvula 3 "V3", según su estado sea activado o desactivado:

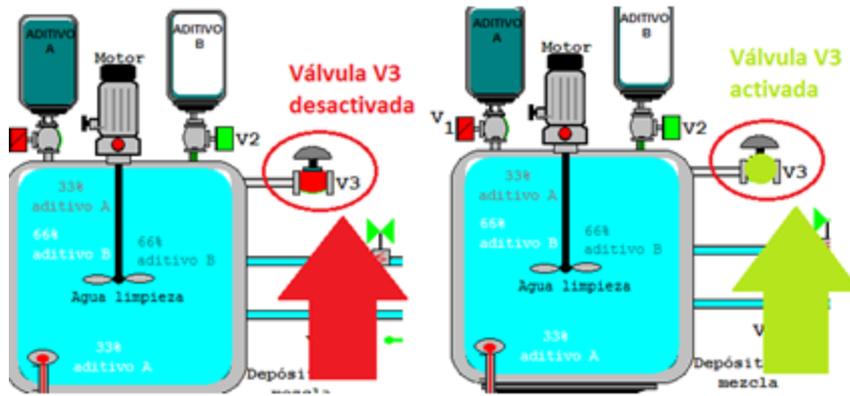


Figura 108: Desactivación /activación Válvula 3 “V3”

Para llevar a cabo su programación, al tratarse de variables tipo EBOOL, se han seguido los pasos mostrados en las Figuras 104, 105 y 106.

► [Bombas](#)

Cada una de las bombas “B1”, “B2” y “B3” indica mediante color rojo que se encuentran desactivadas, o mediante color verde claro que se encuentran activadas. Estas variables han sido creadas tanto para la programación como para la simulación, y su aspecto se observa en la Figura 109:

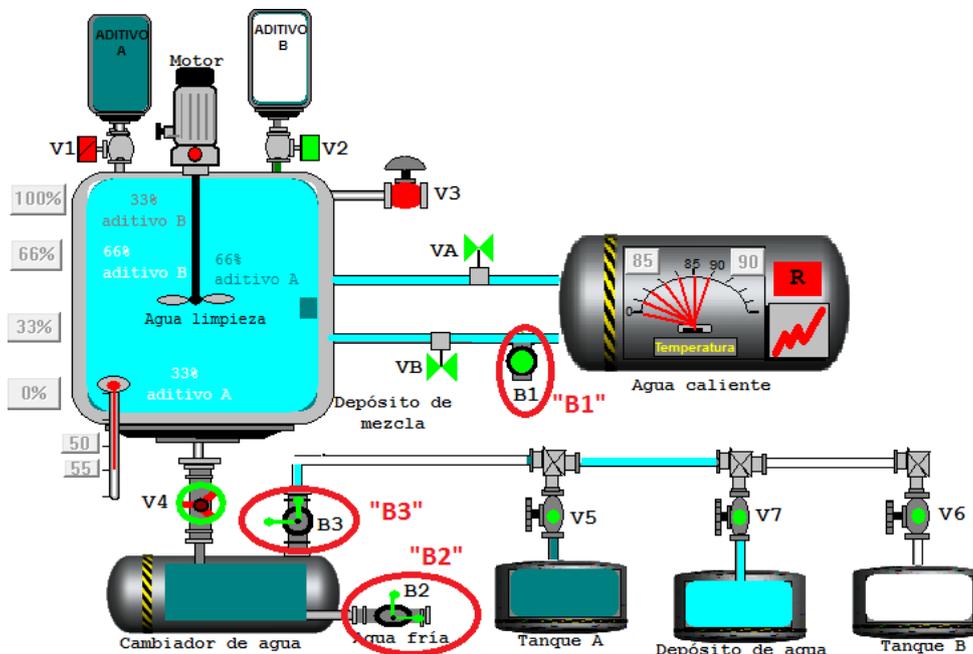


Figura 109: Bombas “B1”, “B2” y “B3”

La Figura 110, muestra el cambio de color de la Bomba 1 “B1”, según su estado sea activado o desactivado:

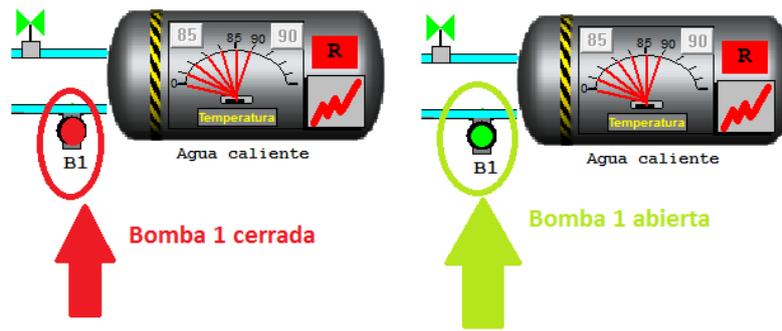


Figura 110: Desactivación / activación Bomba 1 “B1”

Si se hace doble clic sobre una de ellas, el modo de programación es igual que los pasos mostrados en las Figuras 104, 105 y 106, pues nuevamente se trata de objetos dinámicos asociados a variables EBOOL.

➤ Motor

La variable “M” del motor, es creada para la programación por ser una entrada del sistema, y es válida para la simulación. Es una variable asociada a objeto dinámico de tipo EBOOL, como las válvulas.

El motor está programado para que cuando la variable “M” esté activada se ponga en verde claro el punto central del elemento motor y las aspas aparezcan simulando el giro. El caso contrario es el motor desactivado, con el punto central rojo y las aspas inmóviles. El objeto motor “M” se observa en la Figura 111:

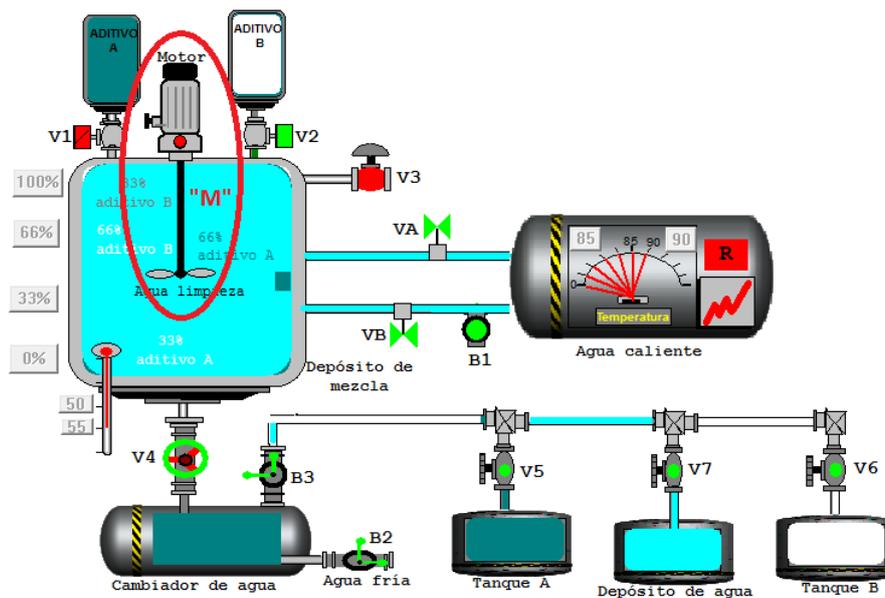


Figura 111: Motor “M”

La Figura 112, muestra el cambio de la variable Motor “M”, según su estado sea en marcha o parado:

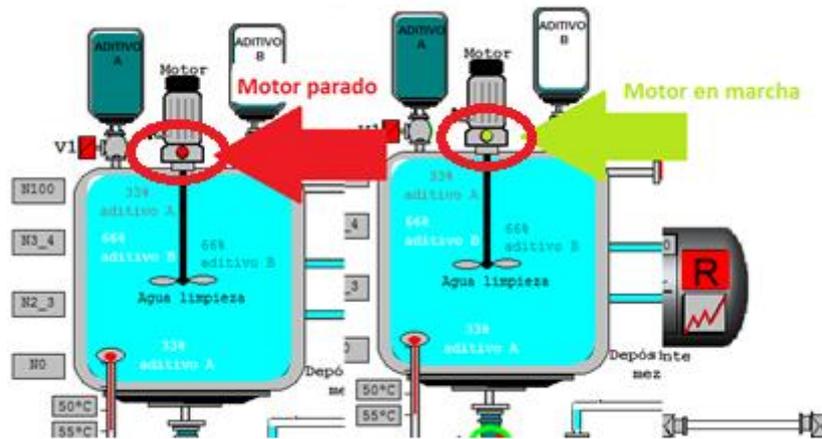


Figura 112: Parada/puesta en marcha del motor

➤ Tuberías

Para dar más dinamismo y realidad la simulación, todas las tuberías se muestran del color correspondiente al líquido que está siendo atravesado. El efecto es conseguido mediante asociación de cada objeto tubería a la salida que lleva asociada el líquido correspondiente.

A continuación, se explica cada tubería a la variable asociada:

- Tubería de agua caliente, asociada a la variable "B1". El agua siempre tiene que estar caliente para que se produzca la mezcla en el depósito, manteniendo la temperatura entre 85º y 90º en el depósito de agua caliente. Cuando la bomba 1 está activada, significa que el agua está pasado hacia el depósito de mezcla.
- Tubería del producto A, asociada a la variable "V5", que da paso al producto A
- Tubería de agua, asociada a la variable "V7", que da paso al agua de la limpieza
- Tubería del producto B, asociada a la variable "V6", que da paso al producto B

La Figura 113, muestra a las tuberías a las que se hace referencia:

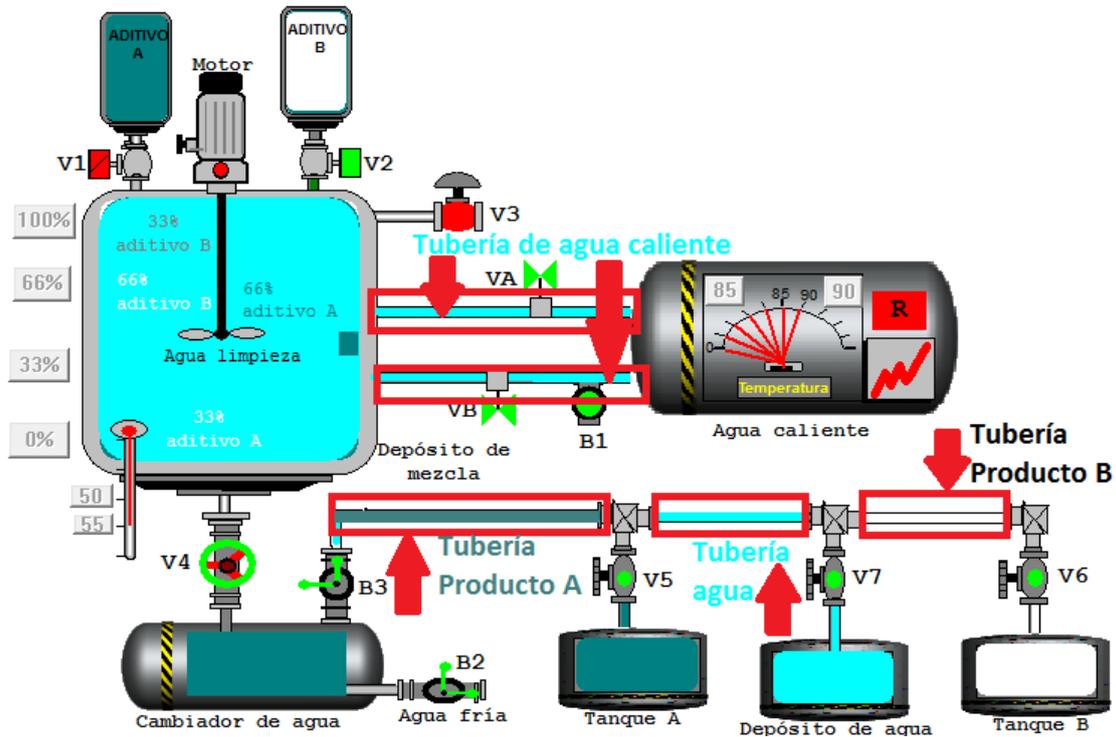


Figura 113: Tuberías de la instalación

5.2 Compilación y transferencia del Proyecto en modo simulación

Una vez que se ha programado y se ha llegado a una solución para el problema planteado, es momento de compilar el sistema para comprobar los errores que pudiera contener y habilitar la transferencia en el simulador.

El modo de actuar para iniciar la compilación es a través de la barra herramientas, seleccionar "Generar: generar todo el proyecto" corresponde al análisis y la generación del código ejecutable, como muestra la Figura 114:

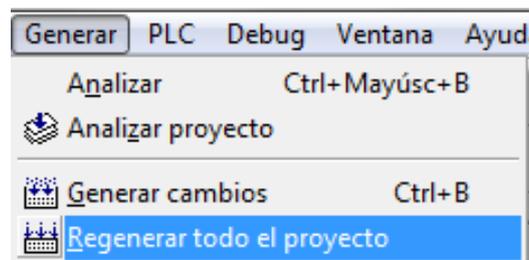


Figura 114: Generación de Proyecto

En el caso de tener errores, aparecerá la descripción de los errores en la ventana de resultados y haciendo un doble clic sobre la línea, el programa llevará directamente a la parte del proyecto que contiene el error.

Cuando se ha conseguido eliminar los errores en el caso de haberlos, se pone en modo simulación desde la barra de herramientas, como muestra la Figura 115:

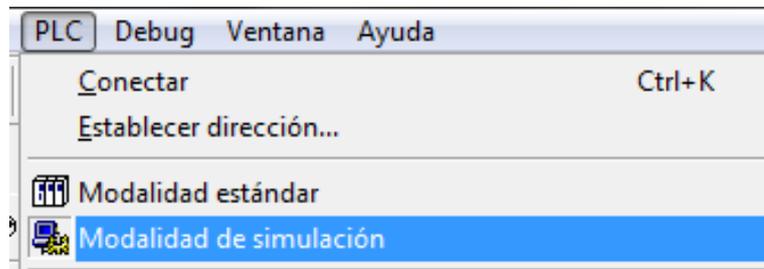


Figura 115: Elección modo simulación

A continuación, antes de realizar la conexión hay que establecer la dirección de la manera que indica la configuración en la Figura 116, y pulsar Aceptar:

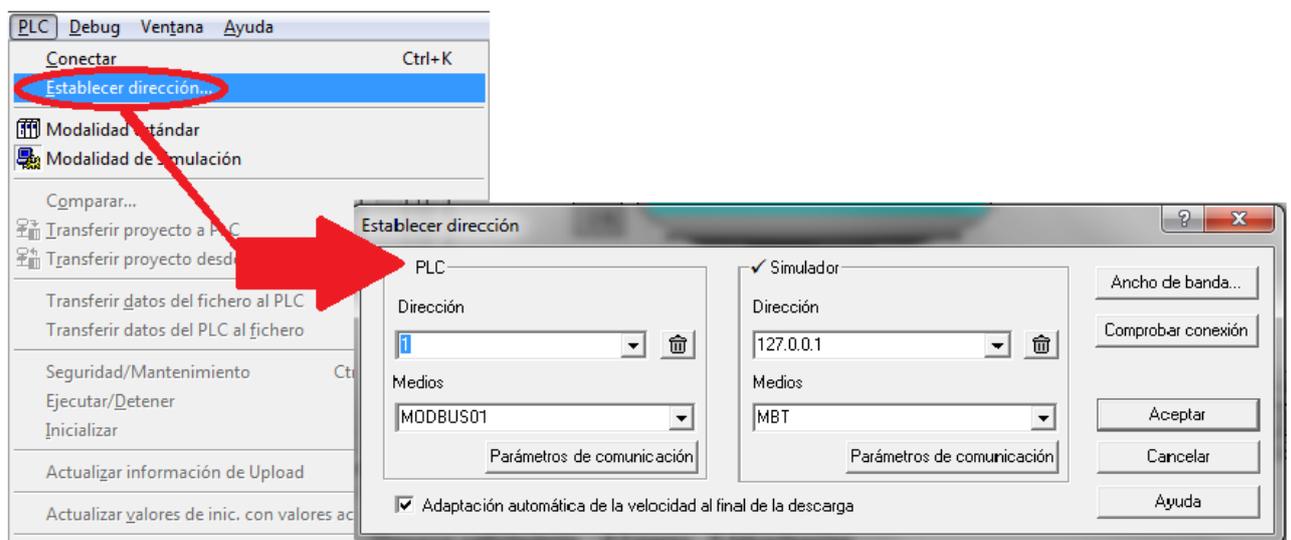


Figura 116: Establecer dirección

La siguiente tabla, muestra los contenidos que hay que rellenar para el correcto funcionamiento del PLC en modo simulador:

	PLC	SIMULADOR
Dirección	1	127.0.0.1
Medios	Modbus	MTB

Tabla 2: Direcciones de simulador

Después de haber establecido la anterior configuración, y conectado de la forma que se muestra en la Figura 117, hay que transferir el proyecto como indica la Figura 118:

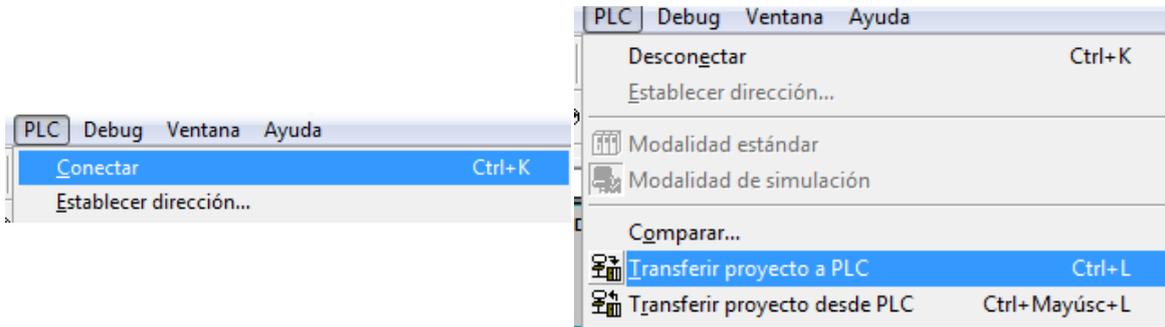


Figura 117: Conectar PLC

Figura 118: "Transferir proyecto a PLC"

En este momento, el software estará preparado para funcionar con el simulador mediante la barra de herramienta dando a PLC/ Ejecutar.

De esta manera el simulador estará arrancado. Si se hace doble clic en la parte inferior derecha de la pantalla del ordenador, en el icono verde de la Figura 119:

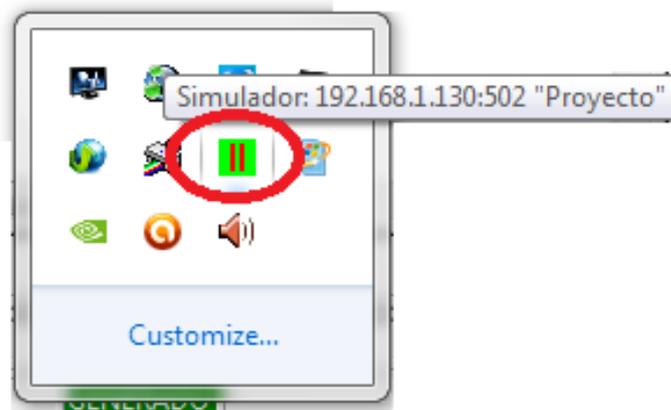


Figura 119: Simulador minimizado

El simulador de PLC mostrará el aspecto que muestra la Figura 120:

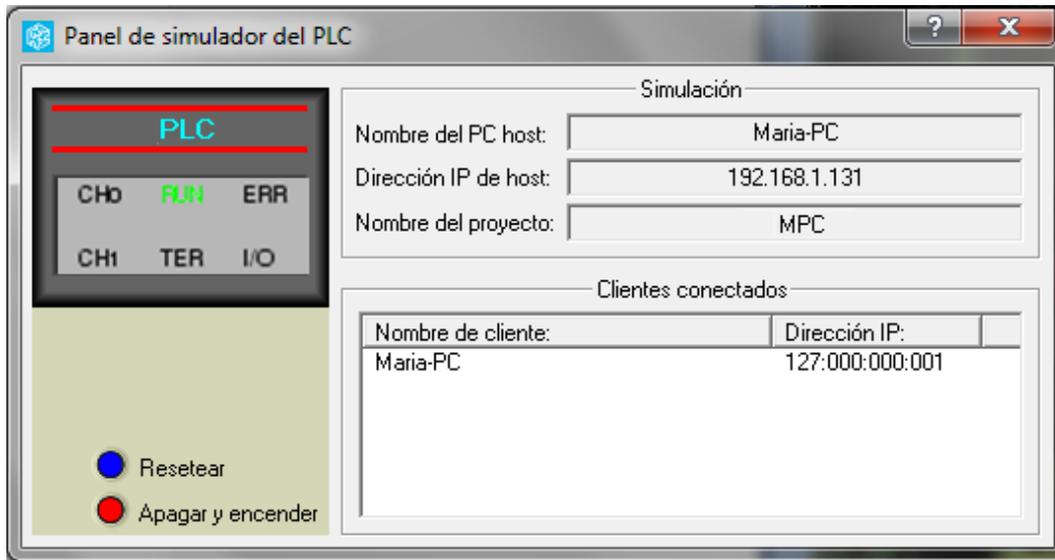


Figura 120: Simulador Unity Pro

Como se observa en la Figura 120, el manejo es muy sencillo, está basado en dos botones de resetear (botón de color azul) y apagar y encender (botón de color rojo).

En el cuadro de la parte superior izquierda del simulador, se muestra el estado del PLC simulado. Cuando todo es correcto, se enciende el piloto RUN. Si presenta algún error en la configuración aparecerá encendido en rojo el indicador ERR.

En el cuadro superior derecho del simulador, hace referencia a la IP y a los datos del proyecto cargado.

En la parte inferior derecha, se muestran los clientes conectados. Aparecerán varios en el caso de estar varios ordenadores conectados al mismo simulador. En la Figura 120 se observa la dirección IP del único ordenador conectado y que anteriormente se ha configurado satisfactoriamente.

Es a partir de ahora, cuando el simulador está cargado con el programa del proyecto que se quiere simular.

Con el fin de seguir visualmente los cambios en los valores que se producen en las señales, y en concreto de las entradas del sistema, se creará una tabla de animación, tal y como indica la Figura 121:

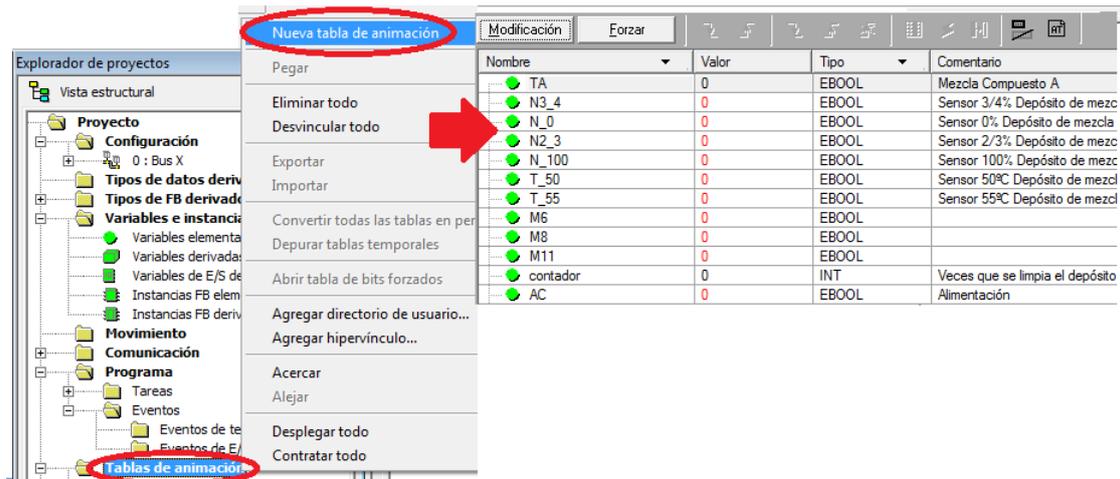


Figura 121: Creación de Tabla de Animación

En el siguiente apartado, se entrará en detalle en el concepto de “Tabla de Animación” y como basar en ella la evolución del sistema.

5.3 EVOLUCIÓN DEL SISTEMA

Para que el sistema siga su curso, y vaya avanzando en el paso de las etapas y transiciones, Unity Pro ofrece dos maneras de poder realizarlo. Una de ellas es a través de “Botones de comando” y la otra es a través de “Tablas de animación”.

5.3.1 Botones de comando

Los botones de comando es la solución más práctica y sencilla de las dos existentes para observar el recorrido del sistema en la simulación. La solución planteada por “botones de comando” consiste en simular las entradas del sistema de ambos bastidores, mediante unos botones presentes en la pantalla, y que activen dicha señal al pulsar el ratón sobre él.

Hay que tener especial atención para que el simulador funcione de esta manera, porque las **entradas, no llevarán asociada ninguna dirección %I en la tabla de variables**. Es decir, al hora de crear la “Tabla de variables elementales” el campo de dirección permanece vacío.

Para realizar la correspondencia de las entradas con dichos botones, se seleccionará el icono de botón de comando que aparece en la Figura 122, y se arrastrará hasta la pantalla:

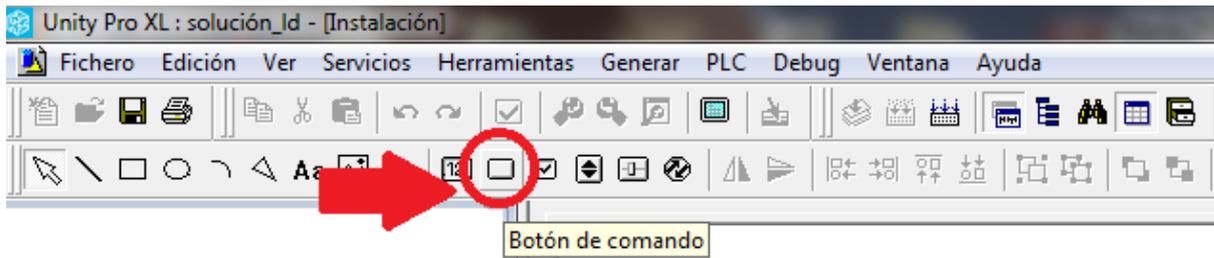


Figura 122: Selección “Botón de comando”

Una vez arrastrado el símbolo de “Botón de comando”, y haciendo doble clic sobre él, aparecerá una ventana como la que se observa en la Figura 123, donde se ha seleccionado la variable Alimentación, “AC” que corresponde con una entrada del sistema:

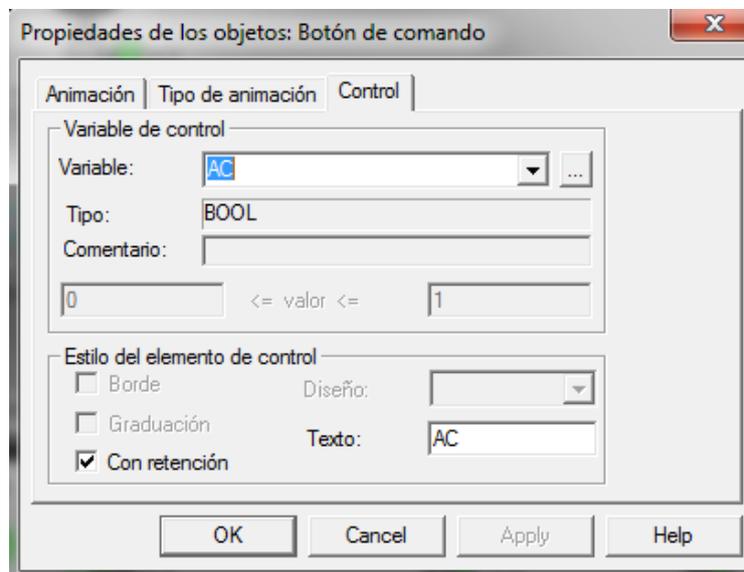


Figura 123: Propiedad “Botón de comando”

En la pestaña “Control” de la Figura 123, se selecciona en el menú desplegable la variable y el tipo de variable que es. Se escribe el texto que se desea que aparezca escrito, y se señala “Con retención”, para que al pulsar sobre él en la simulación, permanezca activado hasta que no se vuelva a pulsar de nuevo. En el caso de pulsar dos veces, el botón se desactivará.

Como ya se ha dicho, cuando se desea que las entradas del sistema sean activadas mediante estos botones, no hay que incluir las direcciones de éstas en la tabla de variables elementales. Es debido a que las entradas se van a activar directamente por el usuario mediante un clic en el “Botón de comando” y no mediante el forzado o modificación de la señal desde la “Tabla de animación”.

Es muy importante a la hora de la simulación, que el botón “habilitar variables de escritura” que se encuentra en la barra de herramientas, permanezca activo. La Figura 124 así lo muestra:

Las bombas “B1”, “B2”, “B3” también se encuentran grises color indicador de que están desactivadas.

El motor se encuentra parado y por eso presenta en su parte superior un punto de color rojo, y las aspas están inactivas.

Los termómetros están desactivados y es por ello que no marcan la temperatura.

Las tuberías están vacías de líquidos y muestran un color gris.

Los botones de comando, al no haber sido pulsados, están inactivos.

Si el usuario pulsa el botón de alimentación “AC” y de producto A “TA”, el sistema comienza a funcionar:

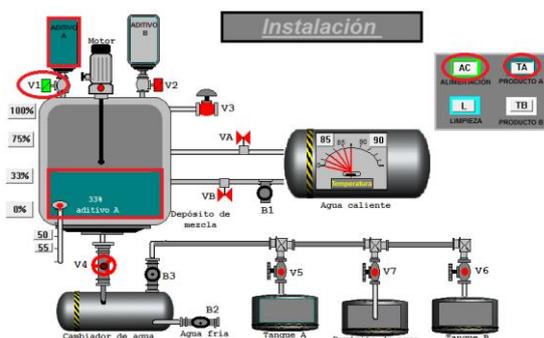


Figura 126: 33% Aditivo A

Al pulsar “AC” y “TA” la válvula “V1” se activa, y es por ello que muestra color verde claro, y se dispensa 33% de “aditivo A”. Se observa que el Tanque de “Aditivo A” y “Depósito de mezcla” muestran color verde oscuro, tal y como señala la Figura 126.

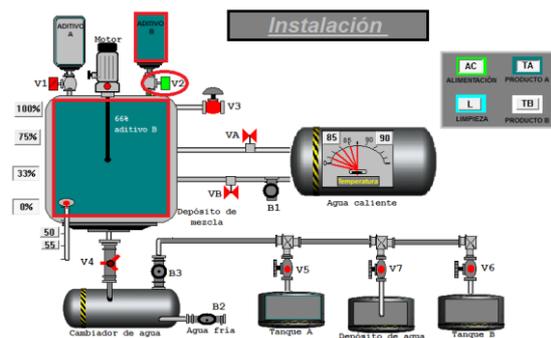


Figura 127: 66% Aditivo B

Tras pulsarse los botones de comando correspondiente a los sensores 0% y 33%, se cierra válvula “V1” (rojo), se abre válvula “V2” (verde claro), se da paso al 66% de “Aditivo B” y el “depósito de mezcla” se llena como cita el enunciado.

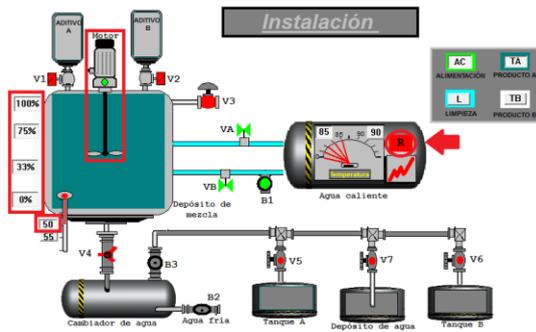


Figura 128: Giro motor mezcla "Producto A"

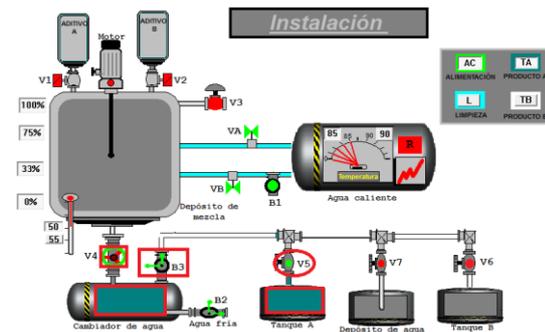


Figura 129: Llenado "Tanque A"

Alcanzado el nivel 100% del "Depósito de mezcla", y tras conseguir la temperatura necesaria de "50°C", se activa el motor. Esta temperatura se consigue gracias a la activación de la resistencia en el depósito de agua caliente. Se puede observar que se da paso a una luz verde claro en la parte superior del mismo, y que sus aspas comienzan a girar. El tiempo que permanece esta situación es un minuto.

Pasado el tiempo temporizado, se activa la válvula "V4" y las bombas "B2" y "B3". Se vacía el "Depósito de mezcla", y por tanto los sensores de capacidad. El líquido pasa por el "calentador de agua" y a continuación, con la activación de la válvula "V5", se llena el "Tanque A".

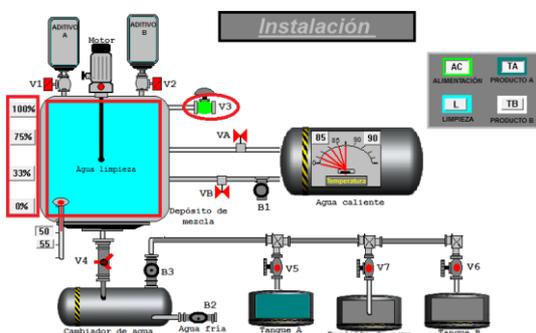


Figura 130: Limpieza "Depósito de mezcla"

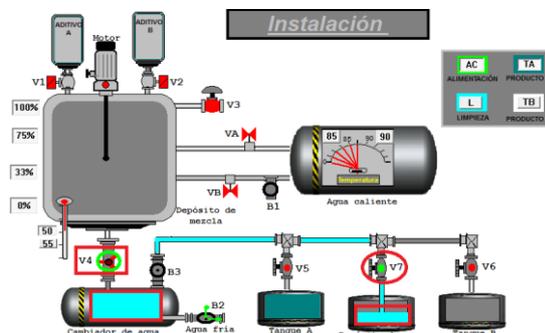


Figura 131: Llenado "Depósito de agua"

Tras esta operación, la válvula "V3" se activa y da paso al agua que hace que el "Depósito de mezcla" se llene. Cuando los cuatro sensores de nivel estén activos, la válvula V3 parará de llenar el tanque y el motor "M" agitará la mezcla un tiempo de un minuto.

Para vaciar el "Depósito de mezcla", se activa "V4" y se desactivan los sensores de nivel. El agua pasa por el "intercambiador de agua", y llega hasta el "Depósito de agua" tras activarse "V7". Ese proceso de repite dos veces, hasta completar el nivel de capacidad del depósito de agua. Concluye la evolución del sistema volviendo al estado de reposo.

5.3.2 Tablas de animación

Otra manera de que las etapas vayan avanzando es mediante “Tablas de animación”. Dichas tablas, son ventanas en las que se puede visualizar el valor de las variables (0 ó 1 si son señales digitales, valor numérico si son enteros o reales). Además, para operaciones de puestas en marcha o diagnóstico, es posible desde la tabla de animación.

Si es a través de la “tabla de animación”, como se desea modificar el valor de una variable, hay que hacerlo mediante el forzado de las señales con direccionamiento, con las entradas %I.

Hay que tener presente que una tabla de animación **siempre** facilitará el trabajo para el seguimiento de señales en sus cambios de valor. Pero si además, la evolución del sistema se realiza con “Tabla de animación”, es de doble utilidad pues es a través de ella es donde se pueden controlar dichos cambios. Las variables que se introducen en la tabla de animación, normalmente son aquellas que son necesarias activar para el recorrido del programa, es decir, todas las variables que son las entradas en un PLC real. Este método es más tradicional y algo más largo a la hora de desarrollarlo.

Tanto las entradas como las salidas deben estar definidas previamente en la tabla de variables elementales de la Figura 132:

Nombre	Tipo	Dirección
AC	EBOOL	%I0.1.3
Agua_limpieza	INT	
B1	EBOOL	%Q0.9.2
B2	EBOOL	%Q0.2.7
B3	EBOOL	%Q0.2.8
Calentar	BOOL	
cont	DINT	%MW2
cont2	BOOL	
cont_menor_2	BOOL	
Deposito_lleno	BOOL	
Deposito_vacio	BOOL	
Dosificador_A	INT	
Dosificador_B	INT	
fin_de_mezcla	BOOL	
Fin_llenado	BOOL	
L	EBOOL	%I0.1.2
Limpieza	BOOL	
llenado_intercambiador	INT	
llenar_producto_A	INT	
llenar_producto_A2	INT	
llenar_producto_B	INT	
llenar_producto_B2	INT	
M	EBOOL	%Q0.2.9
N0	EBOOL	%I0.1.4
N2_3	EBOOL	%I0.1.5
N3_4	EBOOL	%I0.1.6
N100	EBOOL	%I0.1.7

Nombre	Tipo	Dirección
Nivel_A	INT	
Nivel_B	INT	
prod_A	BOOL	
prod_B	BOOL	
Producto	DWORD	%MW0
R	EBOOL	
T_1min	BOOL	
T_50	EBOOL	%I0.1.8
T_55	EBOOL	%I0.1.9
T_85	EBOOL	%I0.1.12
T_90	EBOOL	%I0.1.13
TA	EBOOL	%I0.1.0
Tanque_A	INT	
Tanque_B	INT	
Tanque_vaciado	INT	
TB	EBOOL	%I0.1.1
temp_1min	BOOL	
V1	EBOOL	%Q0.2.0
V2	EBOOL	%Q0.2.1
V3	EBOOL	%Q0.2.2
V4	EBOOL	%Q0.2.3
V5	EBOOL	%Q0.2.4
V6	EBOOL	%Q0.2.5
V7	EBOOL	%Q0.2.6
VA	EBOOL	%Q0.9.0
VB	EBOOL	%Q0.9.1

Figura 132: “Tabla de variables”

En una “Tabla de animación” es muy importante saber si el PLC contiene bits forzados. Si aparecen unos mensajes que indican que hay bits forzados, como no se indica cuáles, es necesario seleccionar la pestaña de información general de la tabla de animación, para descubrir a cuáles hace referencia. Las señales se pueden modificar mediante el botón que

aparece en la parte superior izquierda de “Modificación” o forzar la señal mediante el botón “Forzar”.

Como se observa en la Figura 133, para forzarlo puede ser como flanco de subida o de bajada. Es decir hay que pulsar las flechas de descenso (roja) o ascenso (verde) de la barra superior.

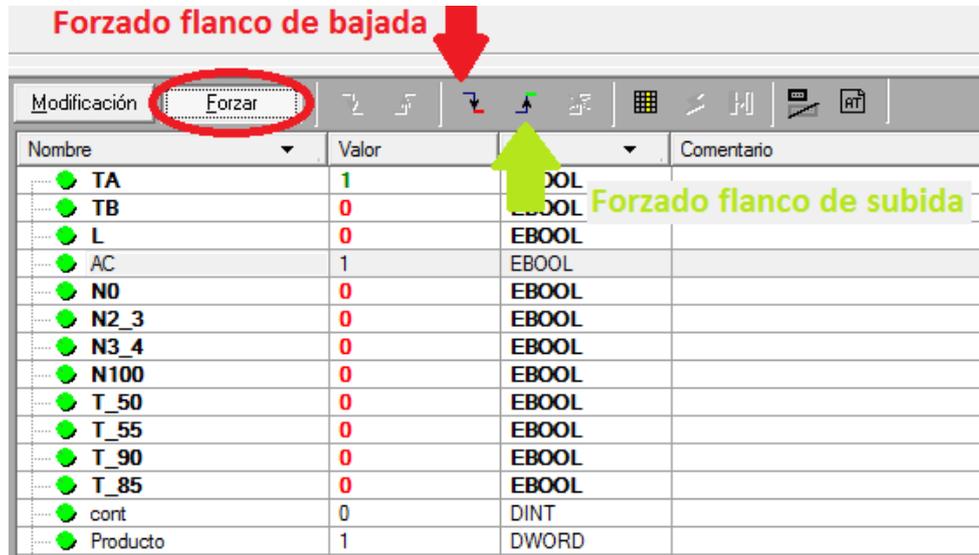


Figura 133: Señales que se pueden forzar

También es posible visualizar un valor utilizando otro formato de visualización: binario, decimal, hexadecimal, ASCII.

Para que el proceso programado vaya avanzando, es necesario ir forzando los bits que sea o sean necesarios en cada instante. En este caso funcionará tal y como se ha programado y en función de la opción que elija el usuario.

Para observar la evolución del sistema mediante “Tablas de animación”, se va a desarrollar el caso en que el usuario pidiese producto “TB”:

Nombre	Valor	Tipo
TA	0	EBOOL
TB	1	EBOOL
L	0	EBOOL
AC	1	EBOOL
N0	0	EBOOL
N2_3	0	EBOOL
N3_4	0	EBOOL
N100	0	EBOOL
T_50	0	EBOOL
T_55	0	EBOOL
T_90	0	EBOOL
T_85	0	EBOOL
cont	0	DINT
Producto	2	DWORD

Figura 134: Habilitación “AC” y “TB”

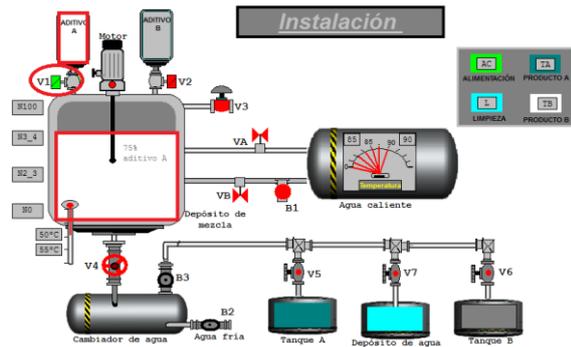


Figura 135: 66% Aditivo A

Se fuerzan las señales “AC” y “TB”. Se observa en la Figura 132 que ese instante la variable “Producto” toma el valor “2”, por tratarse de Producto B.

Al pulsar “AC” y “TB” la válvula “V1” se activa, y se dispensa 75% de “aditivo A”. Se observa que el Tanque de “Aditivo A” y los 3/4 del “Depósito de mezcla” muestran color blanco, tal y como señala la Figura 135.

Nombre	Valor	Tipo
AC	1	EBOOL
cont	0	DINT
L	0	EBOOL
N0	0	EBOOL
N2_3	1	EBOOL
N3_4	1	EBOOL
Producto	2	DWORD
T_50	0	EBOOL
T_55	0	EBOOL
T_85	0	EBOOL
T_90	0	EBOOL
TA	0	EBOOL
TB	1	EBOOL

Figura 136: Activación “N0”, “N2_3” y “N3_4”

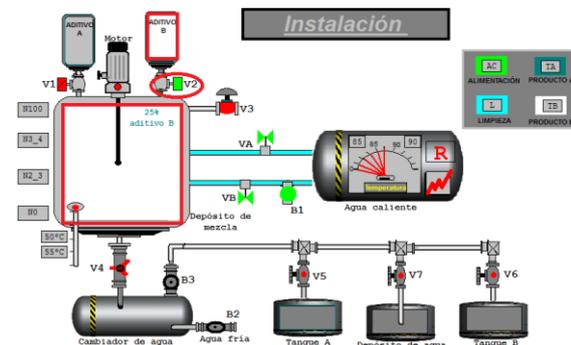


Figura 137: 33% Aditivo B

Se activan las entradas de los sensores del 0%: “N0”, 33%: “N2_3” y 75%: “N3_4”.

Tras estar activo el sensor 75%, y por tanto también el de 0% y 33%, se cierra válvula “V1” (rojo), se abre válvula “V2” (verde claro), se da paso al 25% de “Aditivo B” y el “depósito de mezcla” se llena completamente.

Nombre	Valor	Tipo
AC	1	EBOOL
cont	0	DINT
L	0	EBOOL
N0	1	EBOOL
N2_3	1	EBOOL
N100	1	EBOOL
N3_4	1	EBOOL
Producto	2	DWORD
T_50	1	EBOOL
T_55	1	EBOOL
T_85	0	EBOOL
T_90	0	EBOOL
TA	0	EBOOL
TB	1	EBOOL

Figura 138: Activación "N100" y "T_55"

Se activan los sensores de nivel porque el tanque está lleno, y la temperatura necesaria de 55°C, correspondiendo con "T_55".

Como se observa en la Figura 138, al llegar a 55°C obviamente los 50°C también se han alcanzado.

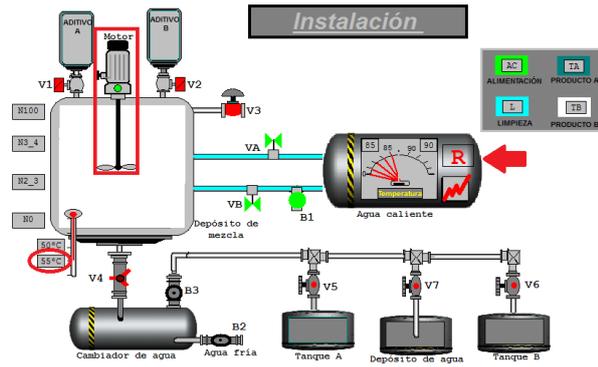


Figura 139: Giro motor mezcla Producto B

Alcanzado el nivel 100% del "Depósito de mezcla", y por tanto todos los sensores de nivel encendidos. Tras conseguir la temperatura necesaria de 55°C, se activa el motor. Se puede observar que se da paso a una luz verde claro en la parte superior del mismo, y que sus aspas comienzan a girar. El tiempo que permanece esta situación es un minuto.

Nombre	Valor	Tipo
AC	1	EBOOL
cont	0	DINT
L	0	EBOOL
N0	0	EBOOL
N2_3	0	EBOOL
N100	0	EBOOL
N3_4	0	EBOOL
Producto	2	DWORD
T_50	1	EBOOL
T_55	1	EBOOL
T_85	0	EBOOL
T_90	0	EBOOL
TA	0	EBOOL
TB	1	EBOOL

Figura 140: Vaciado depósito de mezcla

Se desactivan los cuatro sensores de nivel para el vaciado del depósito de mezcla del producto B.

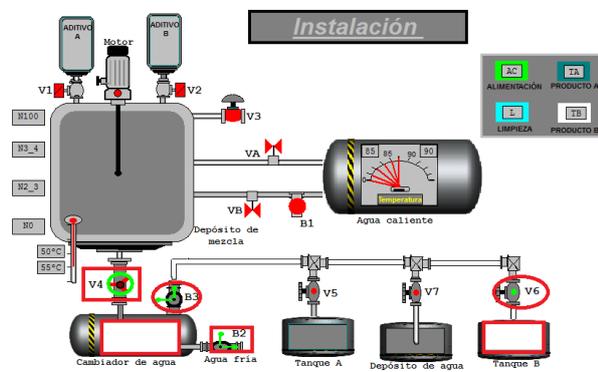


Figura 141: Llenado "Tanque B"

Pasado el tiempo temporizado, se activa la válvula "V4" y las bombas "B2" y "B3". Se vacía el "Depósito de mezcla", el líquido pasa por el "calentador de agua" y a continuación, con la activación de la válvula "V6", se llena el "Tanque B".

Nombre	Valor	Tipo
AC	1	EBOOL
cont	0	DINT
L	0	EBOOL
N0	1	EBOOL
N2_3	1	EBOOL
N100	1	EBOOL
N3_4	1	EBOOL
Producto	2	DWORD
T_50	1	EBOOL
T_55	1	EBOOL
T_85	0	EBOOL
T_90	0	EBOOL
TA	0	EBOOL
TB	1	EBOOL

Figura 142: Llenado “Tanque B”

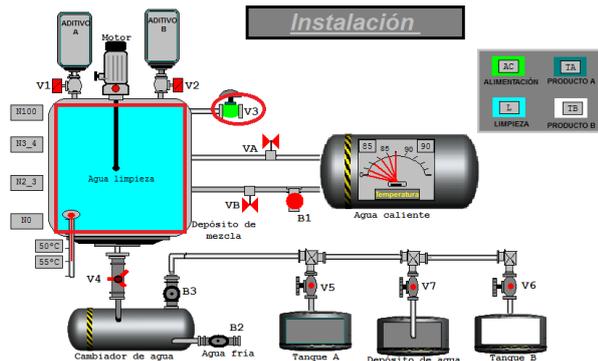


Figura 143: Limpieza

Al activarse la válvula “V3”, el depósito se llena de agua, volviendo a activarse los cuatro sensores de nivel.

Tras esta operación, la válvula “V3” se activa y da paso al agua que hace que el “Depósito de mezcla” se llene. El motor “M” agita la mezcla un tiempo de un minuto.

Nombre	Valor	Tipo
AC	1	EBOOL
cont	0	DINT
L	0	EBOOL
N0	0	EBOOL
N2_3	0	EBOOL
N100	0	EBOOL
N3_4	0	EBOOL
Producto	2	DWORD
T_50	1	EBOOL
T_55	1	EBOOL
T_85	0	EBOOL
T_90	0	EBOOL
TA	0	EBOOL
TB	1	EBOOL

Figura 144: Llenado “Depósito de agua”

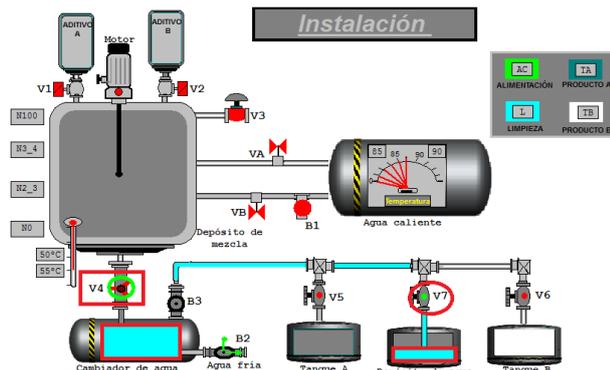


Figura 145: Llenado “Depósito de agua”

Al activarse la válvula “V4”, nuevamente se desactivan los cuatro sensores de nivel para el vaciado del depósito de mezcla, pero esta vez de agua.

De la misma manera que la mezcla de aditivos, para vaciar el “Depósito de mezcla”, se activa “V4”. El agua pasa por el “intercambiador de agua”, y llega hasta el “Depósito de agua” tras activarse “V7”

Para finalizar, se repite este último paso y cuando el "Depósito de agua" queda lleno, tras dos veces de repetición de la limpieza, termina el proceso. El sistema permanece en el estado de reposo.

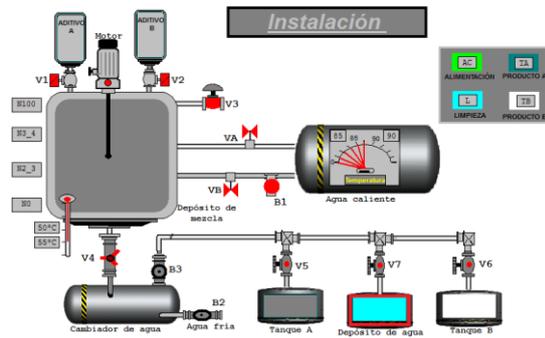


Figura 146: Final de ciclo

6. CONCLUSIONES Y APLICACIONES FUTURAS

**Simulación de la automatización de
procesos con Unity Pro**

6 CONCLUSIONES Y APLICACIONES FUTURAS

6.1 CONCLUSIONES

Desde el punto de vista de simulador como herramienta de apoyo al desarrollar un trabajo, son muchas las ventajas que se adquieren. No sólo por lo fácil que es adquirirlo e implementarlo en un sistema real y porque se desarrolla en un ambiente que brinda mucha sencillez en su uso, sino por lo rápido que se adquieren las destrezas para sacar el máximo partido de esta herramienta.

Unity Pro es una potente herramienta de programación, con la que se puede optimizar el funcionamiento de un sistema de control. Incorporar un buen simulador como es el de Unity Pro, aporta la gran ventaja de poder comprobar señales sin correr el menor riesgo a que estén mal desarrolladas, y traiga consigo incidentes no deseados en grandes instalaciones, y también importante ahorro de dinero, pues se pone en juego un sistema ficticio con idénticas propiedades que el real.

Para analizar el resultado de este documento, hay que comprobar que el objetivo en el punto de partida, era realizar un documento para la creación de un manual de la herramienta del simulador. En base a esto, se ha hecho un recorrido por la simulación completa de un ejemplo, para analizar paso a paso como se puede aprovechar el simulador del software. El manual contiene las pautas a seguir para la creación de la “pantalla del operador”, la utilización de la librería y forma de programación según el tipo de simulación que se quisiera conseguir y los dos métodos de desarrollo de ésta: “Botón de comando” y “Tabla de animación”.

En todo momento se ha intentado que quede claro al usuario cada uno de los aspectos desarrollados. También cabe esperar que el desarrollo de soluciones a medida, como son las que ofrece el software estudiado, resulte fácil de alcanzar y muy satisfactorio para el alumno por dos motivos:

- Por la sencillez de manejo y adaptación del entorno.
- Por la fidelización conseguida en el modelo.

La aportación del simulador a la Universidad, es solucionar los problemas de espacio para distribuir a los alumnos en las horas reglamentarias de realización de prácticas en los laboratorios de automatización, apareciendo la posibilidad de poder trabajar desde casa También, optimizar la enseñanza en las asignaturas relacionadas con la automatización, con el uso del modelo descrito, comprobar los errores de manera segura y visualizar gráficamente el estado del sistema representado.

6.2 APLICACIONES FUTURAS

La amplia disponibilidad y la capacidad de estas nuevas facilidades de comunicación, combinadas con el rápido desarrollo de las tecnologías Web, y la generalización de componentes de uso común para la adquisición de datos y el control de procesos industriales, permiten incorporar estos recursos a procesos de enseñanza-aprendizaje en docencia universitaria evitando la asistencia a los laboratorios tradicionales.

Una de las misiones de la Escuela Politécnica Superior es proporcionar una buena formación a sus alumnos. En esa formación, indudablemente la parte práctica juega un papel principal. Es a través de ella como el alumno fijará conceptos teóricos y adquirirá procedimientos fundamentales del área de conocimiento que esté estudiando. En este contexto los simuladores docentes desempeñan una misión muy importante. Como se ha visto, el uso de simuladores puede resultar complementario con la experimentación real, ya que los alumnos adquieren conceptos acerca de los sistemas de forma que cuando están realizando los experimentos in situ puedan realizar un mayor aprovechamiento didáctico de las horas del laboratorio y se centren en resolver problemáticas específicas de la interacción directa con los equipos como es la puesta a punto de los mismos, cableado, etc.

A nivel de la Educación, en el simulador que ofrece Unity Pro en concreto, es muy potente pero se podría mejorar añadiendo más dinamismo en la funciones. Es decir, se echa en falta en algunas ocasiones elementos que complementen a los existentes para que dicha herramienta cuente con unas posibilidades de simulación en las que la realidad esté presente. Existen algunos elementos de la librería que quedan limitados a la hora de representar diferentes aspectos de la cotidianidad, como es un nivel o un giro de un motor porque son excesivamente estáticos, y no tienen expresión de movimiento cuando entran a formar parte del juego. De esta manera y desde mi punto de vista, se obtiene un resultado con un toque infantil, que a su vez no se encuentra reñido con la sencillez y claridad de conceptos, que en definitiva es el objetivo fundamental.

El trabajo que queda por realizar por parte de la Universidad, es habilitar más recursos informáticos para el acceso a los simuladores, aún siendo los existentes numerosos en comparación con otros centros docentes.

7.BIBLIOGRAFÍA

Automatización de procesos con Unity Pro

7 BIBLIOGRAFÍA

- [1] Real Academia Española de la Lengua. Diccionario de la Real Academia Española. 22ª ed., Espasa_Calpe, Barcelona, 2001.
- [2] Wikipedia, La enciclopedia libre [en línea] <http://es.wikipedia.org/wiki/Automatizaci%C3%B3n_industrial> [Consulta: 22 de septiembre de 2012]
- [3] BLANCO, Dolores; MALFAZ, María. Práctica 1 de autómatas programables entorno de programación Unity Pro. Departamento de Ingeniería de Sistemas y Automática Universidad Carlos III . Año 2012.
- [4] España. Real Decreto 1631/2006, de 29 de diciembre, por el que se establecen las enseñanzas mínimas correspondientes a la Educación Secundaria Obligatoria. 5 de enero de 2007, núm. 5, páginas 677 a 773 (97 págs.)
- [5] PANIAGUA, Soraya. Artículos de la Sociedad de la Información: Aprender haciendo, formación basada en simuladores [en línea]. Disponible en Web: < <http://sociedadinformacion.fundacion.telefonica.com/url-direct/pdf-generator?tipoContenido=articulo&idContenido=2009100116300118> > [Consulta: 21 de abril de 2012]
- [6] Danish Technology Institute, Study of the e-learning suppliers' "market" in Europe. Disponible en Web: <http://ec.europa.eu/education/archive/elearning/doc/studies/market_study_en.pdf> [Consulta: 21 de septiembre de 2012]
- [7] GLASS-HUSAIN, Will. What is a Simulation?. The Drive to Learn, Foro Business Simulations. [Consulta: 15 de Junio de 2012]
- [8] MORA VILLAVICENCIOS, Roberto Carlos. Simulador para la estación neumática PN-2800 del laboratorio del CIM 2000 [en línea] [Sangolquí Ecuador] 2010. Disponible en Web: <<http://repositorio.espe.edu.ec/bitstream/21000/235/6/T-ESPE-027368.pdf>>
- [9] SANCHEZ BENÍTEZ, Gema. Las estrategias de aprendizaje a través de componente lúdico [en línea] [Alcalá de Henares, España]. Disponible en Web: < <http://marcoele.com/descargas/11/sanchez-estrategias-ludico.pdf>> [Consulta: 10 de septiembre de 2012]
- [10] OXFORD, Rebecca L *Language Learning Strategies: What every teacher should know..* Heinle Elt. 1990.
- [11] INTERNATIONAL ELECTROTECHNICAL COMMISSION. Estandarización en la programación del control industrial. IEC 61131-3. 2003

[12] BRAVO, Ignacio. Entornos de automatizaciones industriales [en línea] [Universidad de Alcalá, España]. Disponible en Web: <http://193.146.57.132/depeca/repositorio/asignaturas/201608/T3_VIJE0_2_comunic_PLC.pdf> [Consulta: 12 de mayo de 2012]

