



Universidad de  
Oviedo



# **ESCUELA POLITÉCNICA DE INGENIERÍA DE GIJÓN**

## **GRADO EN INGENIERÍA INFORMÁTICA EN TECNOLOGÍAS DE LA INFORMACIÓN**

**AUTOMATIZACIÓN DE PROCESOS DE CONTROL EN INDUSTRIA  
ALIMENTARIA Y FARMACÉUTICA**

**D. PABLO GORDO MARTÍN  
TUTOR: D. RUBÉN USAMENTIGA FERNÁNDEZ**

**FECHA: febrero 2022**

## CONTENIDO

1. INTRODUCCIÓN .....	8
1.1.- MOTIVACIÓN DEL PROYECTO .....	8
1.2.- VISIÓN GENERAL DEL PROYECTO .....	9
1.3.- OBJETIVOS.....	10
1.4.- EMPRESA .....	11
2. AUTOMATIZACIÓN INDUSTRIAL .....	14
2.1.- AUTOMATIZACIÓN INDUSTRIAL.....	14
2.2.- INDUSTRIA 4.0 .....	15
2.3.- SISTEMAS SCADA .....	17
2.4.- PROCESAMIENTO POR LOTES .....	19
2.5.- WONDERWARE INBATCH.....	21
2.6.- PRINCIPALES ELEMENTOS DE LA HERRAMIENTA .....	22
3. WONDERWARE .....	24
3.1.- WONDERWARE.....	24
3.2.- SYSTEM PLATFORM.....	25
3.3.- ARCHESTRA IDE.....	27
3.4.- INTOUCH OMI .....	30
3.5.- MICROSOFT SQL SERVER MANAGEMENT STUDIO .....	31
3.6.- SITUATIONAL AWARENESS .....	33
4. DESARROLLO SISTEMA DE AUTOMATIZACIÓN INDUSTRIAL .	35
4.1.- SUPUESTOS INICIALES .....	35
4.2.- CREACIÓN DE LA GALAXIA.....	47
4.3.- CREACIÓN E IMPLEMENTACIÓN DE LA PRIMERA VÁLVULA.....	49
4.4.- CREACIÓN DE NUEVOS COMPONENTES.....	52
4.5.- PRIMERA CONEXIÓN CON LA BASE DE DATOS .....	54
4.6.- CREACIÓN DE LAS TABLAS FINALES .....	57

4.7.-	COMIENZO DE LA EJECUCIÓN .....	59
4.8.-	CREACION DE LOS LAYOUT Y SUS SÍMBOLOS .....	65
4.9.-	ÚLTIMAS MODIFICACIONES .....	69
4.10.-	INCLUIR NUEVOS ELEMENTOS EN FUNCIÓN DE LA ESPECIFICACIÓN DE LA PLANTA .....	72
4.10.1.	CREACIÓN DE NUEVOS OBJETOS .....	72
4.10.2.	MODIFICACIÓN DE ELEMENTOS EXISTENTES .....	76
4.11.-	RECETA REAL EMPLEANDO LOS NUEVOS COMPONENTES .....	78
4.11.1.	CREACIÓN DE LA RECETA .....	78
4.11.2.	CREACIÓN DE LA FÓRMULA .....	80
4.11.3.	CREACIÓN OPERACIÓN NUEVA.....	81
4.11.4.	MODIFICAR GRÁFICOS.....	84
4.11.5.	PRUEBA .....	84
5.	PLANIFICACIÓN Y PRESUPUESTO .....	89
5.1.-	PLANIFICACIÓN .....	89
5.1.1.	PREPARACIÓN DEL PROYECTO .....	89
5.1.2.	INGENIERÍA.....	90
5.1.3.	DESARROLLO BASE .....	90
5.1.4.	EJECUCIÓN .....	90
5.1.5.	VALIDACIÓN Y PUESTA EN MARCHA .....	91
5.1.6.	CONTROL Y SUPERVISIÓN .....	91
5.2.-	PRESUPUESTO .....	93
5.2.1.	LICENCIAS NECESARIAS .....	93
5.2.2.	PERFILES NECESARIOS PARA EL DESARROLLO .....	94
5.2.3.	PRESUPUESTO FINAL.....	96
5.2.4.	PRESUPUESTO WONDERWARE INBATCH .....	98
6.	CONCLUSIONES .....	100
7.	REFERENCIAS.....	102
7.1.-	REFERENCIAS DE TEXTO.....	102
7.2.-	LICENCIAS A IMÁGENES .....	103

## ÍNDICE ILUSTRACIONES

Ilustración 1.1 .- Representación caja negra .....	9
Ilustración 1.2 .- Logo TSK .....	11
Ilustración 2.1 .- Automatización industrial.....	15
Ilustración 2.2 .- Máquina de vapor .....	15
Ilustración 2.3 .- Industria 4.0 .....	16
Ilustración 2.4 .- Operario usando SCADA .....	17
Ilustración 2.5 .- Componentes SCADA.....	19
Ilustración 2.6 .- Logo ISA.....	20
Ilustración 2.7 .- Esquema InBatch .....	21
Ilustración 3.1 .- Logo Wonderware .....	24
Ilustración 3.2 .- Logo System Platform .....	26
Ilustración 3.3 .- Esquema elementos de una galaxia .....	27
Ilustración 3.4 .- Ejemplo áreas de una galaxia.....	28
Ilustración 3.5 .- Creación de instancias .....	29
Ilustración 3.6 .- Ejemplo layout.....	30
Ilustración 3.7 .- Aplicación Intouch OMI.....	30
Ilustración 3.8 .- Servidor central SQL .....	31
Ilustración 3.9 .- Vista SQL Server Management Studio.....	32
Ilustración 3.10 .- SCADA con biblioteca SituationalAwarness .....	34
Ilustración 3.11 .- SCADA con elementos genéricos.....	34
Ilustración 4.1 .- Representación caja negra .....	35
Ilustración 4.2 .- Grafo .....	36
Ilustración 4.3 .- Script con el sistema de estados.....	37
Ilustración 4.4 .- Opciones de los Scripts.....	38
Ilustración 4.5 .- Opciones de los atributos .....	39
Ilustración 4.6 .- Lectura en tiempo de ejecución .....	39
Ilustración 4.7 .- Primera base de datos .....	39
Ilustración 4.8 .- Creación de símbolos en la plantilla.....	42
Ilustración 4.9 .- Opciones Wizard.....	43
Ilustración 4.10 .- Primera visualización.....	44
Ilustración 4.11 .- Visualización con dos tanques y dos válvulas .....	44

Ilustración 4.12 .- Primera vez que se añade un mixer.....	45
Ilustración 4.13 .- Visualización completa no final.....	46
Ilustración 4.14 .- Visualización final Factoría .....	46
Ilustración 4.15 Visualización final Lectura.....	47
Ilustración 4.16 .- Aplicación de escritorio .....	47
Ilustración 4.17 .- Creación de la galaxia .....	47
Ilustración 4.18 .- Plantillas creadas .....	48
Ilustración 4.19 .- Áreas creadas .....	49
Ilustración 4.20 .- Atributos de una plantilla .....	49
Ilustración 4.21 .- Creación de un objeto.....	50
Ilustración 4.22 .- Símbolo inicial .....	50
Ilustración 4.23 .- Símbolo inicial finalizado con éxito .....	51
Ilustración 4.24 .- Símbolo Inicial cancelado .....	51
Ilustración 4.25 .- Esquema actual de ejecución .....	52
Ilustración 4.26 .- Expresión para la ejecución del script.....	53
Ilustración 4.27 .- Visualización con dos tanques y dos válvulas .....	53
Ilustración 4.28 .- Visualización con mixer y tanque de salida.....	54
Ilustración 4.29 .- Variables para la conexión a SQL.....	55
Ilustración 4.30 .- Creación de atributos como arrays.....	56
Ilustración 4.31 .- Declaración y uso de una variable indirecta .....	56
Ilustración 4.32 .- Áreas de la galaxia .....	57
Ilustración 4.33 .- Comprobaciones de la fórmula .....	58
Ilustración 4.34 .- Si la fórmula no es correcta no permite la ejecución .....	58
Ilustración 4.35 .- Configuración script de factoría.....	59
Ilustración 4.36 .- Script restauración de valores .....	60
Ilustración 4.37 .- Código comprobación de la acción .....	60
Ilustración 4.38 .- Asignación de valores .....	60
Ilustración 4.39 .- Posibles movimientos de material.....	61
Ilustración 4.40 .- Comprobación primer elemento.....	61
Ilustración 4.41 .- Activación variable mediante variable indirecta.....	62
Ilustración 4.42 .- Atributo de control del objeto VI_1_2.....	62
Ilustración 4.43 .- Ejecución de una operación .....	63
Ilustración 4.44 .- Configuración objeto ViewApp.....	65

Ilustración 4.45 .- Plantilla Procesos .....	66
Ilustración 4.46 .- Insertar símbolo de un objeto propio .....	66
Ilustración 4.47 .- Creación símbolo final.....	67
Ilustración 4.48 .- Enlazar objeto de lectura para SQL .....	67
Ilustración 4.49 .- Visualización final factoría .....	68
Ilustración 4.50 .- Visualización final lectura .....	68
Ilustración 4.51 .- Modificación Trigger period para demorar la ejecución .....	69
Ilustración 4.52 .- Visualización Factoría con log .....	70
Ilustración 4.53.- Visualización Lectura final.....	71
Ilustración 8.1 .- Opciones wizard para la configuración de la planta .....	72
Ilustración 8.2 .- Configuración inicial de la planta.....	73
Ilustración 8.3 .- Boceto en papel de los componentes .....	73
Ilustración 8.4 .- Creación de nuevas instancias .....	74
Ilustración 8.5 .- Cambio de nombre y asignación del objeto a su área.....	74
Ilustración 8.6 .- Nuevos atributos en el área de los mixers.....	75
Ilustración 8.7 .- Nuevos atributos en el área de las válvulas .....	75
Ilustración 8.8 .- Duplicar símbolo S4 .....	76
Ilustración 8.9 .- Configuración ContentType .....	77
Ilustración 8.10 .- Configuración final Factoría.....	77
Ilustración 8.11 .- Esquema receta en papel.....	79
Ilustración 8.11 .- Crear nueva operación .....	82
Ilustración 8.12 .- Script operación Pasteurizador .....	82
Ilustración 8.13 .- Script incremento temperatura.....	83
Ilustración 8.14 .- Propagación de cambios a instancias.....	83
Ilustración 8.15 .- Nueva opción RadioButtonGroup .....	84
Ilustración 8.16 .- Pantalla Lecturas sin datos cargados .....	85
Ilustración 8.17 .- Pantalla Lecturas con dato cargados correctamente .....	85
Ilustración 8.18 .- Pantalla Lecturas con datos cargados erróneamente.....	86
Ilustración 8.19 .- Botón para comenzar la ejecución .....	86
Ilustración 8.20 .- Aumento de temperatura en el mixer 2.....	87
Ilustración 8.21 .- Movimiento de material entre el tanque 3 y el mixer 4.....	87
Ilustración 8.22 .- Almacenamiento producto final.....	88

## ÍNDICE TABLAS

Tabla 4-1 .- Base de datos Receta.....	40
Tabla 4-2 .- Base de datos Fórmula.....	40
Tabla 4-3 .- Asignación de los datos .....	41
Tabla 5-1 .- Planificación Preparación del Proyecto .....	89
Tabla 5-2 .- Planificación Ingeniería .....	90
Tabla 5-3 .- Planificación Desarrollo Base.....	90
Tabla 5-4 .- Planificación Ejecución .....	90
Tabla 5-5 .- Planificación Validación y Puesta en Marcha .....	91
Tabla 5-6 .- Planificación Control y Supervisión.....	91
Tabla 5-7 .- Planificación final .....	92
Tabla 5-8 .- Presupuesto licencias .....	94
Tabla 5-9 .- Presupuesto personal .....	94
Tabla 5-10 .- Presupuesto consultor de Operaciones .....	96
Tabla 5-11 .- Presupuesto Programador y Jefe de proyecto .....	97
Tabla 5-12 .- Cálculos finales presupuesto.....	97
Tabla 6-1 .- Coste licencias básicas.....	98
Tabla 6-2 2 .- Coste licencias InBatch.....	99
Tabla 8-1 .- Receta Yogurt de Arándanos .....	80
Tabla 8-2 .- Fórmula Yogurt Arándanos .....	81

# 1. INTRODUCCIÓN

## 1.1.- MOTIVACIÓN DEL PROYECTO

Se pretende dar una alternativa al software de Wonderware InBatch, que es una herramienta de Wonderware que trabaja en la automatización de procesos por lotes, empleando System Platform, de forma que se permita configurar mediante un criterio propio la herramienta. Así mismo, se evitará tener que emplear diferentes licencias propias de Wonderware InBatch para su uso, lo que supondrá un gran ahorro de dinero.

Este software se crea con la idea de ser comercializado en un futuro por la empresa Grupo TSK Electricidad y Electrónica donde me encuentro trabajando actualmente. Se creará una primera versión más simple donde no se incorporarán todas las características y configuraciones necesarias para la puesta en marcha real de una planta, pero si se hará de una forma simulada.

Por lo tanto, este proyecto tiene principalmente dos valores, primero el valor práctico, pues se permitirá crear la herramienta con unas configuraciones propias, mejorando posibles características de Wonderware InBatch y el segundo, y muy importante, el valor económico, pues se estima que se reducirá en una gran suma de dinero la venta del producto, principalmente por que únicamente se necesitará la licencia propia de Wonderware Platform, pero no las múltiples licencias necesarias para InBatch.

## 1.2.- VISION GENERAL DEL PROYECTO

Este proyecto se basa en el software de Wonderware InBatch, que es un sistema de gestión de recetas por lotes o cajas automatizado. Podría verse como una caja con dos entradas, un actuador y una cantidad, además, una salida con la siguiente acción a realizar, que será otra caja con los mismos parámetros. El sistema de automatización por lotes consiste en repetir este procedimiento hasta finalizar el proceso, lo que provoca que el fallo sea mínimo y así los resultados sean mejores.



*Ilustración 1.1 .- Representación caja negra*

La idea de partida será tener una planta industrial donde se establecerán los componentes que luego se emplearán en la receta, posteriormente se incorporará la receta, que será el esqueleto de nuestro proceso, es decir, los puntos por los que se va a avanzar en la ejecución, para finalmente introducir la fórmula, donde se especificarán las cantidades y los componentes finales sobre los que se actuará.

La herramienta que se va a implantar se creará con el software de Wonderware, en concreto en System Platform que proporciona un entorno de desarrollo de aplicaciones basado en objetos que permite construir, probar e implementar una amplia gama de aplicaciones industriales utilizando objetos de aplicación reutilizables. También será de gran utilidad InTouch OMI, pues mediante este motor de visualización de Wonderware se podrá tener una vista de la planta que se está implantando.

### 1.3.- OBJETIVOS

- Investigar situación actual de la informática en el sector de la industria, cómo funcionan los sistemas SCADA y los usos que tienen en los procesos industriales.
- Investigar acerca de Wonderware y sus herramientas, como son System Platform, Intouch OMI, AchestrA IDE...
- Investigar el software actual de Wonderware InBatch, de manera que cuando se implemente el software se tenga claro lo que se quiere hacer, mejorar o incluso obviar, pudiendo llegar a ser innecesario para el cliente que solicite la herramienta.
- Implementar la solución de manera que sea lo más versátil posible, esto es, permitir modificar los componentes de la planta en función de las características de la fábrica que se desea automatizar.
- Permitir incorporar numerosas recetas y fórmulas y ejecutarse sin interrupciones para cada tipo.
- Realizar un diseño de la parte visual de la planta lo más actual posible y que toda la información que recoja sea útil y de interés.
- Demostrar la utilidad de lo que se ha realizado mediante la simulación de una receta real en una planta configurada con unas características concretas.

## 1.4.- EMPRESA

El presente proyecto fin de carrera se desarrolla cursando las prácticas extracurriculares en la empresa TSK Electricidad y Electrónica S.A, tratando de cubrir las necesidades de la organización en base a lo que solicitan y exigen los clientes.



*Ilustración 1.2 .- Logo TSK*

Esta empresa posee una capacidad de operación a nivel global con clientes y proyectos en gran cantidad de países y continentes. El sector principal en el que opera es el industrial, destacando en infraestructuras eléctricas, plantas industriales y centrales de generación de energía. La organización está compuesta por más de 1.000 empleados presentes en algo más de 35 países. Se calcula que se generan unas cifras de negocio que llegan a alcanzar los 100 millones de euros.

Además, se encuentra ubicada entre las 10 mejores empresas españolas que trabajan con proyectos de ingeniería y construcción industrial, liderando sobre todo el sector en la automatización y control de procesos industriales.

Actualmente, la empresa tiene la capacidad de implantar distintos tipos de instalaciones industriales con personal altamente cualificado y equipos muy modernos y avanzados, permitiendo realizar grandes diseños y gran computación de cálculo.

Existen 7 líneas de negocio bien definidas, estas son:

- Energía:

Centrales flexibles y renovables para la generación de energía, principalmente solares, eólicas y de biomasa.

- Digital Innovation:

Proyectos basados en la digitalización industrial, en ellos la innovación es esencial. Se desarrollan soluciones que mejoran la cadena de valor de los clientes optimizando su rendimiento. Las tecnologías empleadas principalmente son BigData, IoT y Realidad Virtual.

- Infraestructuras Eléctricas:

La empresa es líder en el sector del equipamiento eléctrico. Desarrolla proyectos de potencia y control asociados a instalaciones industriales nuevas y ya existentes.

- Medio Ambiente:

Siendo conscientes de lo que la sociedad demanda, TSK ha decidió fomentar la conservación y preservación de los recursos naturales del planeta, principalmente se presta atención al desarrollo de tecnologías orientadas al abastecimiento de agua potable.

- Industria:

Diseño e implantación de diferentes plantas de proceso. En el ámbito del papel, minería, siderurgia y cementos se ofrecen equipos e instalaciones de almacenamiento, transporte e incluso plantas “llave en mano” completas. También, en los últimos años se están llevando a cabo proyectos agroindustriales, enfocándose en la sostenibilidad y optimización de costes.

- Gas to Power:

Construcción de vías para el transporte de gases o carburantes a través de diferentes instalaciones, almacenes de combustibles, principalmente gas natural y terminales GNL (Gas Natural Licuado).

- Handling:

Mediante PHB Weserhütte, empresa que fue adquirida en 1995 por TSK, se lidera el sector de los sistemas de almacenamiento, equipamiento de puertos y, principalmente, cintas transportadoras.

De entre todas las líneas de negocio, el presente trabajo se desarrolla en “Digital Innovation”, encargados de dar el servicio propio de un moderno departamento TIC a todas las empresas del grupo y desarrollar proyectos basados en las nuevas tecnologías como son:

- ❖ Proyectos sobre la Industria 4.0, especialmente sobre Wonderware.
- ❖ Soporte en sistemas MES, SCADA...
- ❖ Experiencia en ciberseguridad.
- ❖ Proyectos de IoT e IIoT.

Con relación a los proyectos mencionados previamente, especialmente con los primeros sobre la Industria 4.0 y Wonderware, TSK es System Integrator Partner Certified ArchestrA, lo que quiere decir, que se posee el nivel más alto de conocimientos en soluciones Wonderware, por lo que todo el personal que trabaje para la empresa ha de estar certificado, para así garantizar que la empresa tiene los conocimientos necesarios para el correcto uso de sus herramientas y asegurar el éxito. Otra ventaja de estar certificados es que se tiene acceso a los servicios de soporte técnico de la herramienta, lo cual aporta mayor calidad si es posible.

# 2. AUTOMATIZACIÓN INDUSTRIAL

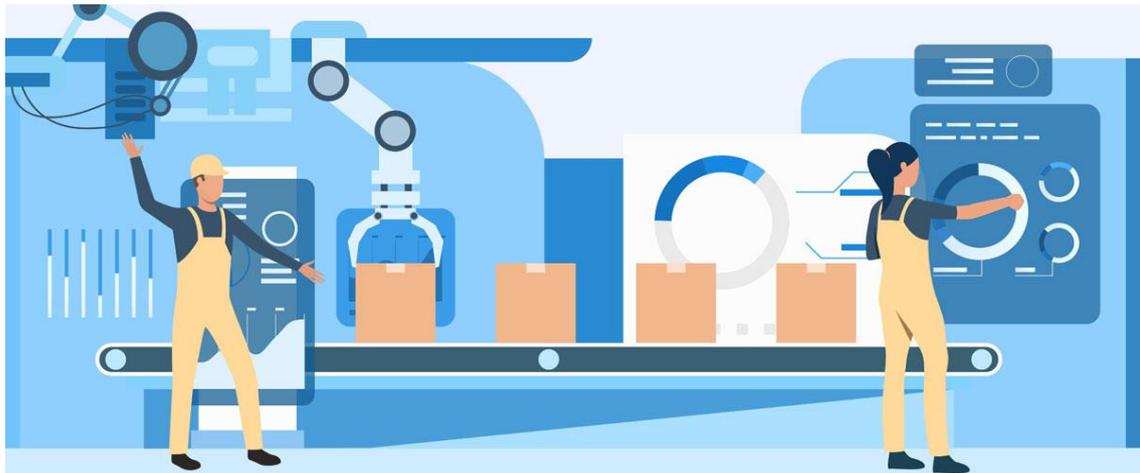
## 2.1.- AUTOMATIZACIÓN INDUSTRIAL

La automatización industrial consiste en un conjunto de herramientas combinadas con el fin de mejorar e impulsar la producción, además de aumentar la distribución y comercialización en las industrias. Hoy en día se ha convertido en parte indispensable de la fabricación de productos de alta calidad en tiempo récord.

En conjunto, se podría definir como la aplicación de diferentes tecnologías y máquinas con el fin de reducir el tiempo de fabricación de los productos y las cargas laborales, pues se pretende conseguir una mínima participación humana, todo esto con un menor margen de error.

Ofrece numerosos beneficios como son: [1]

- Además de ampliar la producción de las fábricas, conseguir que esta producción sea de la mayor calidad posible y sin fallos.
- La automatización es un proceso computarizado por lo que no sería necesario parar, permitiendo mantener la fabricación durante todo el día sin interrupciones.
- Conseguir reducir el tiempo de fabricación de los productos, los costes de materiales, aprovechando mejor las cantidades disponibles, y la mano de obra, lo que produce un ahorro económico final.
- Realizar actividades que para los humanos podrían ser de alto riesgo mientras que para las máquinas no.



*Ilustración 2.1 .- Automatización industrial*

## 2.2.- INDUSTRIA 4.0

Primero fue la máquina de vapor a mediados del siglo XVIII, posteriormente el descubrimiento de la electricidad para fines domésticos e industriales a finales del siglo XIX, para pasar a la automatización en el siglo XX, y ahora la industria se encuentra en una nueva etapa, la transformación digital.



*Ilustración 2.2 .- Máquina de vapor*

Consiste principalmente de la digitalización de las plantas industriales mediante el uso de nuevas tecnologías. El objetivo es hacer estas plantas más inteligentes, además de tratar que sean más respetuosas con el medio ambiente

Se basa en el uso principalmente de internet, para así tener todos los equipos y componentes siempre conectados y tener la capacidad de controlarlos de forma remota. De esta manera conseguiremos tener procesos óptimos y más eficientes para impulsar una mayor productividad.

Las nuevas tecnologías más empleadas son, IOT (Internet of Things), IA (Inteligencia Artificial), Big Data, Cloud Computing, Alanytics, Realidad Virtual y Aumentada, herramientas de Planificación de la Producción y Nanotecnología. Estas nuevas tecnologías están en constante evolución, por lo que será necesario mantenerse informado continuamente con el fin de no quedarse obsoletos.

Las empresas han de comprender en que aspectos deben avanzar, pues los cambios, aunque a veces pueden ser complicados, están pensados para ofrecer unos mejores resultados, lo que a nivel industrial se traduce en mejorar la calidad de los productos y los ingresos. Aquellas empresas que no consigan comprender los cambios necesarios perderán muchas oportunidades de mercado ya que estarán malgastando su tiempo y dinero. [2]



*Ilustración 2.3 .- Industria 4.0*

### 2.3.- SISTEMAS SCADA

Según la definición que propone Wonderware, *SCADA no es una tecnología concreta si no un tipo de aplicación.* [3]

El objetivo de estos sistemas es principalmente conseguir la excelencia operativa, esto es, aumentar la productividad de las plantas de producción manteniendo el nivel de calidad más alto posible, minimizando los costes y el tiempo de producción.

Las siglas SCADA vienen de “Supervisory Control And Data Acquisition”. Se trata del uso de diferentes aplicaciones informáticas en conjunto para asegurar el correcto funcionamiento en los procesos productivos. Inicialmente estaba únicamente destinado a al sector industrial, pero cada vez se está extendiendo a más ámbitos, como, por ejemplo, el farmacéutico y el alimentario.

Su función principal es supervisar y llevar un control constante de los componentes que se encuentran interconectados en nuestra planta, lo cual nos ayuda a subsanar, corregir posibles errores y, además, adelantarlos a lo que va a suceder mediante modelos predictivos. Este sistema se encuentra en una capa intermedia entre los sistemas de control (PLCs) y los MES.

Se emplea principalmente para:



*Ilustración 2.4 .- Operario usando SCADA*

- Supervisión en remoto.
- Asegurar una calidad mínima durante todo el proceso.
- Controlar y procesar gran cantidad de datos en tiempo real.
- Tener una visión gráfica de todo el proceso.
- Interactuar mediante alarmas ante los posibles errores.
- Almacenar los datos recogidos para poder realizar posteriormente un análisis de estos y predecir futuros errores.

Sus principales componentes son: [4]

- HMI (*Human Machine Interface*), se refiere a una pantalla mediante la cual el usuario controlador logra comunicarse con la máquina o software, mostrando los datos en tiempo real y permitiendo al usuario controlar las máquinas de manera remota.
- MTU (*Master Terminal Unit*), es elemento central de control de un sistema en el que se recogen datos de los RTUs y envía las acciones a realizar a través de líneas de comando.
- RTU (*Remote Terminal Units*), son microprocesadores que transmiten datos obtenidos remotamente para después ser procesados. Como se muestra en la *Ilustración 2.5*, se convierten las señales recibidas de los sensores en datos enviados al MTU.
- PLCs, son la mejor alternativa a RTU, comúnmente empleada en la industria por su bajo precio y versatilidad.
- Red, todos los componentes han de conectarse entre sí y la mejor forma de hacerlo será mediante Wifi, fibra óptica o Ethernet.
- Sensores, serán los dispositivos mediante los cuales se captan magnitudes físicas o de otro tipo que deseamos medir.

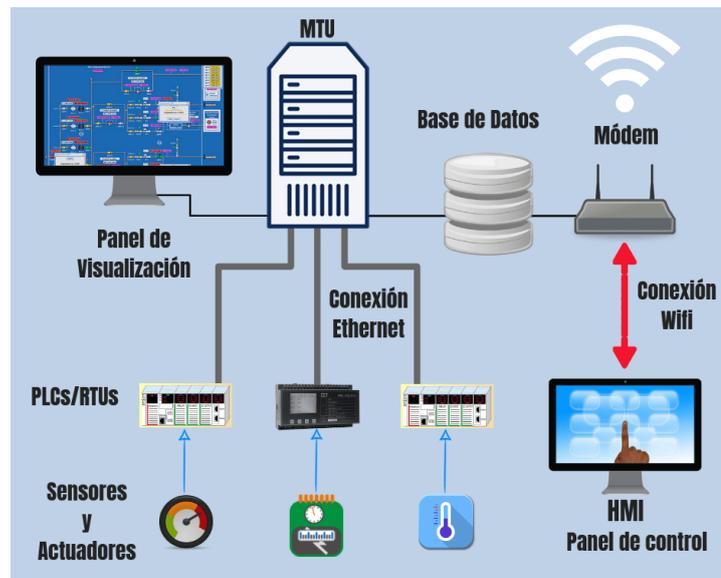


Ilustración 2.5.- Componentes SCADA

Wonderware, la herramienta que se empleará a lo largo de este proyecto, pues es uno de los principales sistemas SCADA.

## 2.4.- PROCESAMIENTO POR LOTES

Comúnmente se le conoce también como batching, es el proceso de realizar lotes en una cadena de producción, de forma simultánea y sin alteraciones. En términos de programación se basa en la ejecución de archivos de manera automática, sin intervención de un usuario, con la extensión .bat, que es un tipo de archivo de texto sin formato que contiene una secuencia de comandos. [5]

La ISA88 define un proceso batch como: *“un sistema que induce a la producción de cantidades finitas de material, sometiendo las cantidades de material de entrada a un conjunto ordenado de actividades de procesamiento sobre un periodo finito de tiempo, usando uno o más recursos”*.



*Ilustración 2.6 .- Logo ISA*

La idea del procesamiento por lotes surge en 1890 cuando Herman Hollerith desarrolló un sistema de tarjetas perforadas, el cual le ayudaba a procesar de forma más sencilla los datos obtenidos de los censos de Estados Unidos.

Ventajas:

- Estándares ISA-88 integrados.
- El usuario no participa una vez establecida la receta.
- Forma efectiva de procesar datos gran cantidad de datos simultáneamente.
- No hay variaciones en el proceso de producción.
- La gestión del proceso no precisa de grandes conocimientos.
- No es necesario un hardware muy complejo.
- Continúa en ejecución aún sin estar conectado a la red.

## 2.5.- WONDERWARE INBATCH

El software proporcionado por Wonderware es uno de los softwares de procesamiento por lotes más comercializados del mercado.

Este sistema puede ser empleado para tratar procesos muy complejos y que requieran de un nivel de flexibilidad muy alto. Además, la forma de interactuar con él es muy sencilla permitiendo modificar las fórmulas rápido añadir nuevas recetas sin necesidad de grandes conocimientos ni experiencia.

Como todos los softwares de procesamiento por lotes, este también cumple con las especificaciones de la ISA-88, pero a diferencia del resto, incluye más fases de transparencia. También permite la administración de rutas flexibles y ejecución de múltiples productos y lotes en varios flujos. [6]

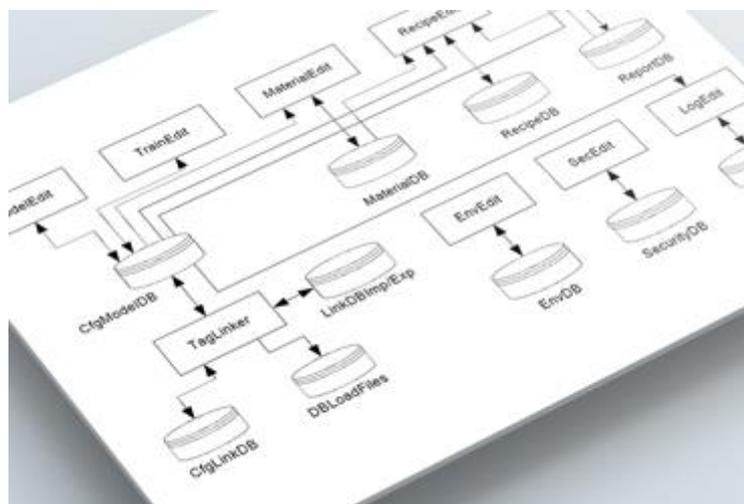


Ilustración 2.7.- Esquema InBatch

Otra característica de sistema es que permite administrar los materiales, llevando un conteo de la disponibilidad de estos, los movimientos, inventarios, etc. Esto reduce el costo para cumplir con las regulaciones actuales de la industria y se almacena en un Registro Electrónico de Batch completo (EBR). Posee una gran versatilidad en cuando a los servidores, permitiendo conectar varios a la misma base de datos y compartir información,

así como gran capacidad de trabajar con clientes potentes que demanden mucha capacidad de cómputo.

Se emplea principalmente en plantas industriales donde se realizan productos químicos y alimentarios. Proporciona una automatización de los procesos con el fin de establecer una alta calidad sin errores, además, también permite la historización de todos los equipos.

## **2.6.- PRINCIPALES ELEMENTOS DE LA HERRAMIENTA**

Para poder entender el proyecto es importante tener claros los elementos físicos que se incluyen en el mismo, ya que se hablará numerosas veces de ellos. Estos son los elementos principales que se encuentran en una planta industrial donde se pueden realizar medicamentos o diferentes tipos de alimentos:

- Tanques: En ellos se almacenan los materiales que se emplean, estos contienen material líquido, desde agua, leche, suero o cualquier otro producto que se necesitara. Habrá tanques de entrada en los cuales estará el material inicialmente, pero también habrá otros que serán de salida, los cuales inicialmente estarán vacíos, pues es donde se almacenará el producto una vez ha pasado por todo el proceso.
- Tolvas: Al igual que los tanques, es un componente de almacenamiento, pero en estos se encuentran materiales sólidos como son la sal, el azúcar, cacao, o cualquier otro material que se emplee en la planta.
- Válvulas: Son indispensables en una planta, pues con ellas se controlarán las cantidades que a mover. Si una válvula se abre, es que hay un movimiento de material entre dos o más componentes.

- Tuberías: Junto con las válvulas, son los componentes encargados de transportar el material entre los componentes de la planta.
- Motores: Serán necesarios para mover los materiales por las tuberías, pues para llevar un material desde el punto A hasta el B, lo normal será que la tubería no sea completamente vertical y vaya hacia abajo, sino que se necesite la ayuda de un motor para crear una presión y así conseguir que el material se desplace.
- Mixers: En estos componentes es donde se realizan las acciones principales, ya puede ser: alcanzar una temperatura, conseguir una presión concreta, arrancar un mezclador... es decir, cualquier tipo de acción que se deba realizar. El material llegará desde los tanques y tolvas mediante las tuberías y válvulas hasta estos elementos. Entonces, en este punto, se realizará una acción.

## 3. WONDERWARE

### 3.1.- WONDERWARE

Este software se fundó en 1987 de la mano de Dennis Morin y Phil Huber con la intención de ayudar a los operarios de las plantas industriales a monitorizar las operaciones que se producían de una manera sencilla. Tras el éxito que tuvieron, se unieron más socios: Cole Chevalier, Bill Urone y Jerry Cuckler.



*Ilustración 3.1 .- Logo Wonderware*

Wonderware surge en 1987, cuando se crea la primera interfaz hombre-máquina, InTouch HMI. Estaba basada en Microsoft Windows, lo cual fue algo totalmente novedoso para aquellos tiempos, pues ofrecía la posibilidad tanto a los operarios como a los directores de las plantas de interactuar con ellas. El crecimiento que tuvieron fue tan rápido, que en 1990 ya estaban operando en más de 100.000 fábricas del mundo.

InTouch HMI sigue siendo hoy en día una de las mayores innovaciones en el sector industrial, además, formando parte de la compañía energética y de ingeniería Schneider Electric, son el interfaz más utilizado para la supervisión y recolección de datos en los procesos industriales. Este software industrial posee como principales características y ventajas el ser abierto, escalable, fácil de usar, seguro y muy versátil, además actualmente están operativas más de 800.000 licencias de su software repartidas en más de 130 países.

Wonderware ha ampliado sus servicios, no solo recogiendo y mostrando datos, si no ayudando a aprovecharlos de la manera más eficiente posible. Además, ya se realizaban conexiones inteligentes dentro de las empresas antes de que a esto se le denominara Internet de las Cosas, y con la creación de Wonderware System Platform, mejoraron las comunicaciones abiertas y seguras dentro de las fábricas sincronizando las personas, procesos, equipos y aplicaciones. [7]

### 3.2.- SYSTEM PLATFORM

Wonderware System Platform, construido como un sistema operativo para aplicaciones industriales, proporciona una plataforma única y escalable, diseñada para integrar sus existentes y nuevos sistemas operacionales e IT en una visualización unificada, InTouch Operation Management Interface (OMI).

Este sistema operativo, ofrece una ventaja competitiva permitiendo una rápida gestión del cambio que se alinea con los constantes cambios de la tecnología y los negocios. Permite el cumplimiento de estándares entre procesos y equipos funcionales. Es una infraestructura común que puede proporcionar un contexto universal a los procesos en tiempo real, alarmas, eventos y datos históricos archivados a través de sistemas empresariales dispares. Es un modelo de información común que hace que el diseño y el mantenimiento del sistema sea más eficiente, flexible y seguro.

System Platform 2017 mejora la experiencia del usuario mediante una visualización estandarizada de contenidos y procesos lo que simplifica en gran medida la tarea de lograr una identificación óptima de los procesos industriales. Además, funciona con cualquier dispositivo con una navegación inteligente y pantallas capaces de ofrecer una gran experiencia en las operaciones.

Wonderware System Platform es una plataforma totalmente abierta que permite conectarse y recoger datos de innumerables dispositivos de control, equipos y sistemas de fabricación. Este producto de Wonderware es de los más utilizados por las grandes plantas industriales a nivel mundial.

Además, soporta todas las soluciones de Wonderware como Wonderware Skelta BPM, Wonderware MES Operations, Wonderware Intelligence, Wonderware IntelaTrac, Wonderware Dream Report, Wonderware InBatch, Wonderware Recipe Manager Plus and Wonderware Online. [8]

Las principales herramientas de las que se compone y además emplearemos en el desarrollo de nuestra solución son: ArchestrA IDE, principal herramienta donde se implementan todas las áreas, objetos, plantillas y visualizaciones, System Platform Manager, realiza las conexiones con los elementos a automatizar mediante drivers de comunicación y posee un Log Viewer donde se muestran los errores, Historian, combina técnicas avanzadas de almacenamiento y compresión de datos con una interfaz de consultas sencilla que permite analizar los valores y tomar acciones necesarias para evitar pérdidas de producción en tiempo real, (la solución propuesta no se incorpora este proyecto, pero será necesaria en el futuro) e Intouch OMI, que es el motor encargado de correr las visualizaciones.



*Ilustración 3.2 .- Logo System Platform*

### 3.3.- ARCHESTRA IDE

Es la arquitectura Tecnológica basada en .NET de Microsoft, desarrollada por Wonderware para facilitar e impulsar la Integración de Dispositivos y Sistemas a distintos niveles. Esta arquitectura de software informático está diseñada para integrar y extender la vida de los sistemas heredados, aprovechando las tecnologías de software y los estándares abiertos más avanzados de la industria. [9]

Es la herramienta principal de System Platform, y mediante su uso es posible ensamblar aplicaciones rápidamente por medio del uso de objetos de software, en lugar de “programarlas”. Es posible crear plantillas de objetos para casi cualquier propósito y luego utilizarlos para construir nuevas aplicaciones de manera sencilla a través del reensamble y la ligera modificación de estos objetos, ahorrando con ello tiempo y reduciendo los costos de desarrollo. [10]

Los proyectos se desarrollan en lo que ArchestrA IDE denomina *Galaxia*, aquí es donde se definen todos los elementos de la planta, se escriben los scripts, se realizan las visualizaciones y demás configuraciones que sean necesarias. Dentro de las galaxias existen una serie de componentes que conviene describir, pues se mencionaran con asiduidad:

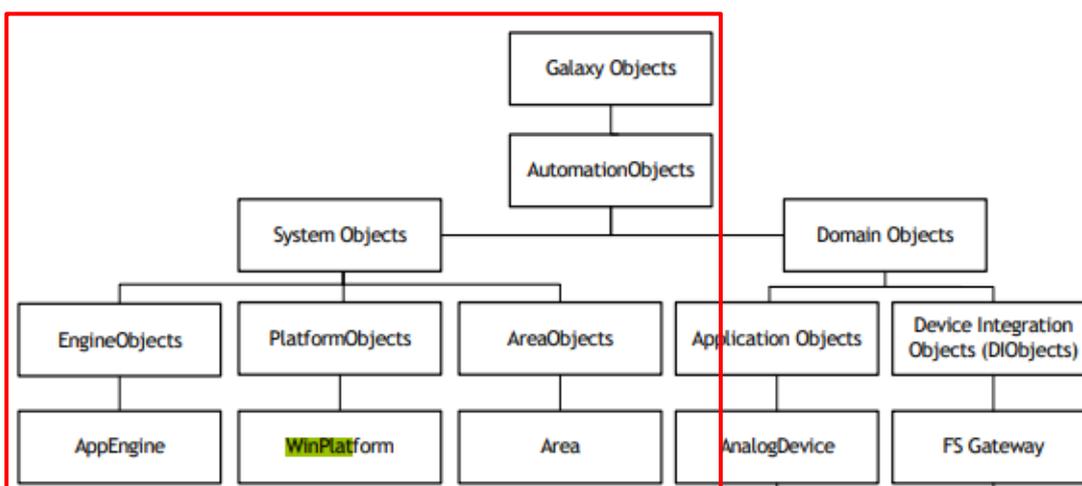
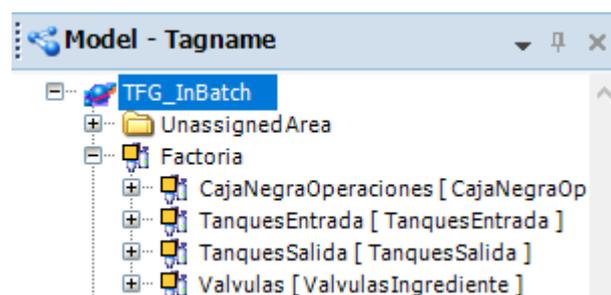


Ilustración 3.3 .- Esquema elementos de una galaxia

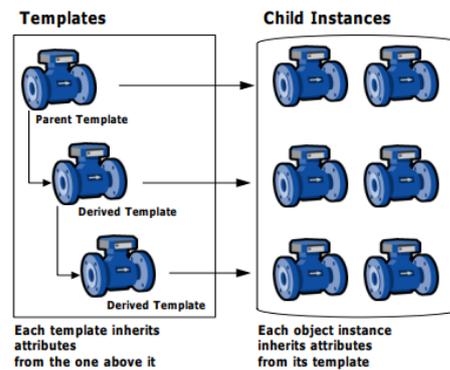
- WinPlatform (o plataformas): servidor donde se ejecutará la galaxia y en él se pueden definir ciertas configuraciones como por ejemplo si queremos historizar los datos, o parte de ellos, o el tiempo de refresco de la información. Puede existir más de uno en nuestra galaxia.
- AppEngine (o motores): Apuntan a su plataforma correspondiente y suelen existir tres, uno general con las instancias y áreas de la galaxia, otro para las comunicaciones y otro para la visualización.
- Áreas: En ellas se organizan los objetos para darle sentido a nuestra galaxia, se intenta replicar la distribución que habrá en la realidad dándole sentido a nuestra arquitectura. Todas las instancias han de estar en un área.



*Ilustración 3.4 .- Ejemplo áreas de una galaxia*

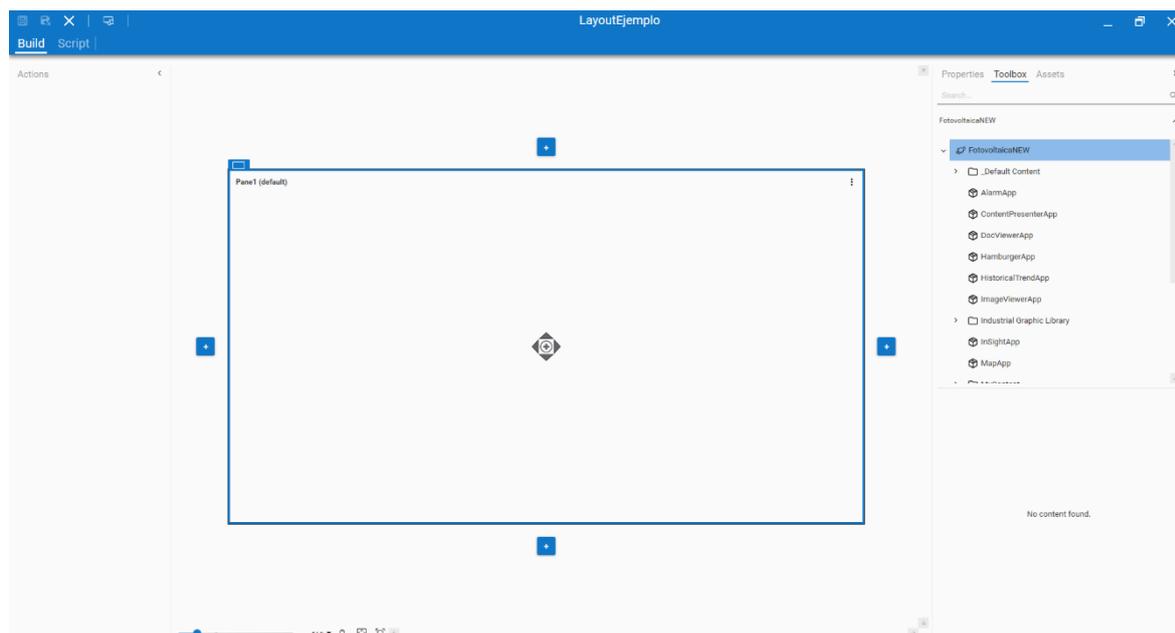
- Plantillas: En ellas se definen todas las configuraciones que queremos que tengan nuestras instancias, como pueden ser los atributos, scripts, alarmas, etc. Habrá una plantilla por cada tipo de objeto que tengamos en nuestra planta, así cuando queramos añadir nuevos elementos, desde la plantilla será muy rápido, pues las instancias heredaran todas sus características. La utilización de ellas permite tener un proceso estandarizado en el sistema, ya que, si queremos realizar un cambio en alguno de los elementos, con cambiarlo en la plantilla se aplicará a todas las instancias que hereden de esta, por lo que no será necesario tocar los elementos uno a uno.

- **Instancias:** Son los objetos en tiempo de ejecución creados a partir de las plantillas y habrá uno por cada elemento de nuestra planta. Heredaran todos los atributos de la plantilla, además de las configuraciones y símbolos que puedan existir en ella.



*Ilustración 3.5 .- Creación de instancias*

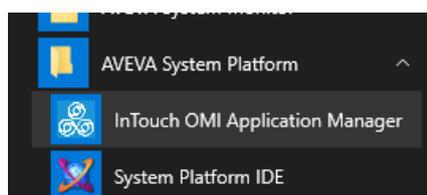
- **Atributos:** Se definen en las áreas, plantillas e instancias y existirán en todo momento, se podrá acceder a sus valores en todo momento.
- **Variables:** Similares a los atributos, pero estas se definen en los scripts, y únicamente se podrá acceder a ellos desde su propio script, por lo que no podremos emplear su valor en otro lugar de nuestra galaxia.
- **Símbolos:** Representación gráfica de los elementos que tenemos en nuestra galaxia, podremos crearlos libremente mediante líneas, curvas, etc. O bien podremos emplear símbolos ya creados en bibliotecas ya existentes. Los símbolos se añadirán a un layout.
- **Layout:** En él se distribuye la visualización final, escogiendo que símbolos se muestran en cada parte de la pantalla.



*Ilustración 3.6 .- Ejemplo layout*

### 3.4.- INTOUCH OMI

Es la herramienta que se encarga de correr las visualizaciones creadas en System Platform para formar el sistema de supervisión y control. Permite la visualización también a través de un navegador web, esto permite acceder al SCADA desde cualquier lugar y dispositivo mientras se disponga de conexión con el servidor. Por otro lado, se pueden configurar diferentes usuarios que tendrán acceso al sistema de supervisión, pero dependiendo del tipo de usuario se tendrá acceso a unas pantallas y funcionalidades distintas.



*Ilustración 3.7 .- Aplicación Intouch OMI*

El entorno de visualización es capaz de soportar la operación en varios monitores. Así mismo, se adapta dinámicamente la visualización de la aplicación a diferentes formatos y tamaños en función de las características de los dispositivos utilizados ya sean equipos de sobremesa, paneles industriales o bien dispositivos itinerantes como tabletas o móviles. Así mismo, soporta nativamente la operación en dispositivos táctiles con sus funciones de navegación y será capaz de utilizar paneles fijos o deslizantes. [11]

### 3.5.- MICROSOFT SQL SERVER MANAGEMENT STUDIO

Es uno de los programas de administración de servidores y gestión de bases de datos relacional más empleados en la actualidad. Creado y optimizado por Microsoft, nació en 1989 cuando solo se llamaba SQL y únicamente daba servicio a sistemas operativos Windows, pero desde hace 6 años también se puede incorporar a sistemas GNU/Linux y Docker. Los principales rivales de SQL Server son: MySql, MariaDB, Oracle, entre otros.

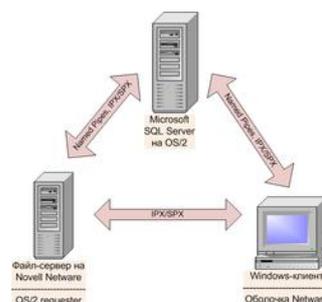


Ilustración 3.8 .- Servidor central SQL

Permite emplear numerosas instancias en el mismo servidor físico y sus usos principales para bases de datos son: [12]

- Cree y modifique bases de datos rápidamente;
- Agregar y modificar objetos de base de datos, incluidas tablas y vistas;
- Prueba de objetos de base de datos con herramientas de prueba externas;

- Implementación de bases de datos;
- Ejecución de una consulta en bases de datos;
- Optimizar las bases de datos para mejorar el tiempo de respuesta;
- Gestionar bases de datos como copias de seguridad y restauración;
- Importar y exportar datos.

También permite realizar las siguientes tareas administrativas del servidores SQL:

- Registro y conexión de servidores en la ubicación y de forma remota
- Gestionar usuarios para servidores y bases de datos
- Monitorear la actividad del servidor a través de registros
- Monitorear el desempeño de la base de datos y SQL Server
- Eliminar y desconectar bases de datos
- Herramientas para escribir guiones para realizar tareas administrativas

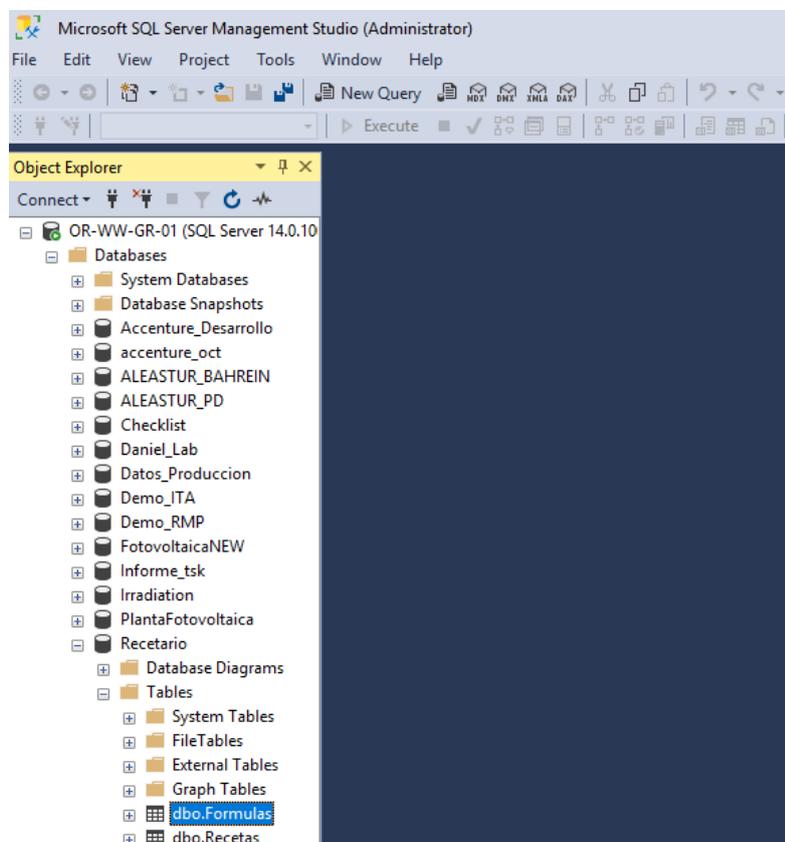


Ilustración 3.9 .- Vista SQL Server Management Studio

### 3.6.- SITUATIONAL AWARENESS

La Biblioteca de Símbolos de Wonderware Situational Awareness es algo más que unos sencillos gráficos y proporciona elementos de diseño muy valiosos y reutilizables que simplifican el diseño y el mantenimiento de HMI, permiten ahorrar tiempo, esfuerzo y coste.

Los diseños de HMI tradicionales a menudo complican las cosas al presentar demasiada información sin contexto, dejando así que los operarios interpreten qué es lo más importante. Wonderware Situational Awareness presenta al usuario información altamente contextualizada y fácil de interpretar que es específica para los aspectos más importantes del proceso, además de filtrar el ruido para que los operarios estén concentrados. Wonderware Situational Awareness ofrece las siguientes ventajas: [13]

- Identificación más rápida de problemas
- Maximización del tiempo de disponibilidad
- Plantilla empoderada
- Menos alarmas y reconocimiento acelerado de alarmas
- Mayor eficiencia operativa
- Menos errores

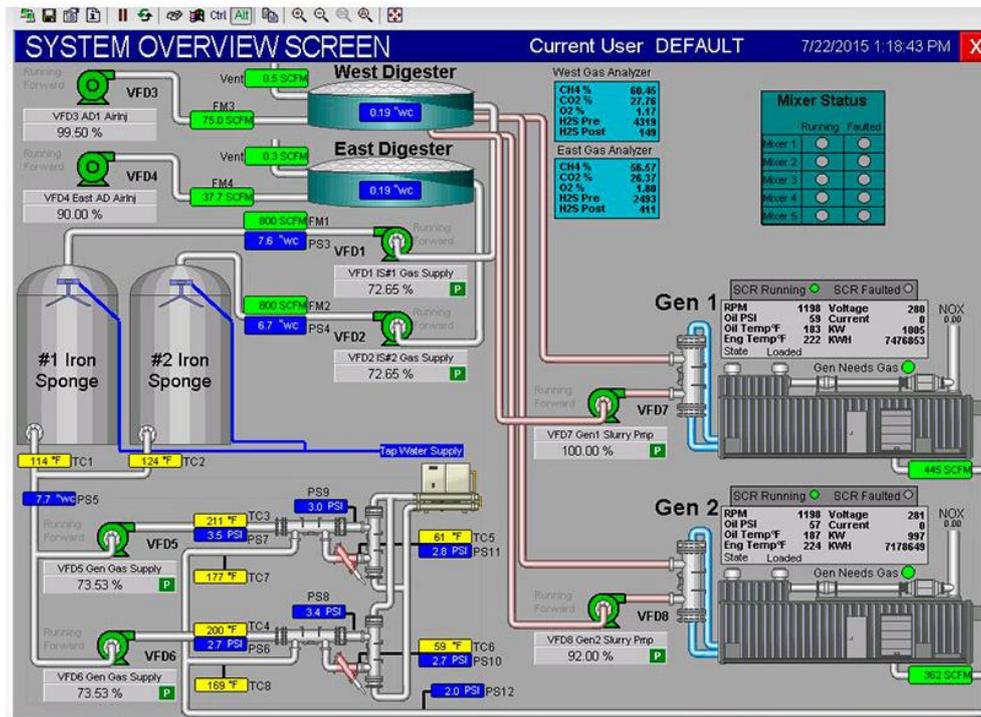


Ilustración 3.11.- SCADA con elementos genéricos



Ilustración 3.10.- SCADA con biblioteca SituationalAwareness

# 4. DESARROLLO SISTEMA DE AUTOMATIZACIÓN INDUSTRIAL

## 4.1.- SUPUESTOS INICIALES

Una vez se ha comprendido el funcionamiento de lo que será la solución del software InBatch, se analizarán los puntos críticos en la creación y desarrollo de esta herramienta que se va a implantar.

La herramienta se basará en la solución de Wonderware InBatch, y será un sistema de automatización por lotes o cajas negras automatizado, el cual consiste la repetición de unos procesos concretos hasta llegar a un resultado esperado.

Esta herramienta nueva se creará mediante System Platform y ArcestrA donde se crearán las plantillas y se definirá el esqueleto de la planta y se visualizará mediante Intouch OMI. Además, SQL Management Server tomará un papel importante a la hora de declarar las recetas y fórmulas que se quieran incluir.

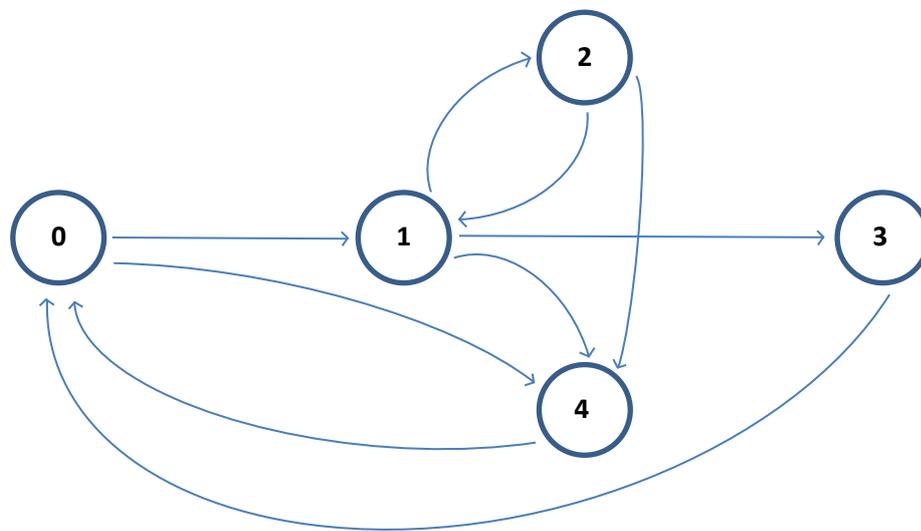


*Ilustración 4.1 .- Representación caja negra*

Los puntos en donde a priori se espera tener problemas son los siguientes:

- ¿Como controlar los movimientos de material, procesos y demás acciones?

Uno de los principales problemas será como actuar en las cajas negras. La primera versión se realizará mediante un sistema de estados como el que se muestra en la figura siguiente, simulando el movimiento de material en una válvula.



*Ilustración 4.2 .- Grafo*

Inicialmente se encuentra en el estado “Vacío” (0) y pulsando sobre el botón de arranque comenzará la ejecución pasando al estado “Arranque” (1), aquí pueden suceder varias cosas, que se presione el botón de pausa, como su propio nombre indica se pausará la ejecución guardando los valores en dicho momento de ejecución, que se produzca algún error, o se pulse sobre el botón de cancelado, lo cual lo llevaría a dicho estado “Cancelado” (4) o bien que finalice con éxito y vaya al último estado llamado “Fin” (3). En caso de pausarse se podrá continuar desde el momento en el que se detuvo la ejecución o bien se podrá cancelar y pasar al estado cancelado. Una vez la ejecución llega al estado “Cancelado” solo existe la posibilidad de volver al inicio restaurando la válvula.

De cara a implementarlo mediante script se realizará de la siguiente manera:

```
4 if Me.Estado == "Arranque" then
5     'Variable para comprobar ultimo este
6     x = 1
7     'Comprobar nombre y cantidad
8     'Comprobar si es posible
9     'Añadir un temporizador
10    'Enviar señal de fin
11    'Mostrar el resultado
12    'Notificar a la siguiente caja que j
13
14    'Enviar al estado de FIN
15 endif
16 if Me.Estado == "Pausa" then
17     if x == 1 then
18         'Guardar temporizador
19         'Parar el tiempo
20     endif
21     else
22         'mostrar error de que no se ha j
23 endif
24
25 if Me.Estado == "FIN" then
26     'Incluir valores en las salidas
27     'Salida de actuacion
28     'Salida de siguiente caja negra
29
30 endif
31
32 if Me.Estado == "Cancelado" then
```

Line: 1 Col: 1

*Ilustración 4.3 .- Script con el sistema de estados*

Este sistema de estados o grafo sirvió para realizar la primera caja negra con su correcto funcionamiento, pero una vez analizado, se llegó a la conclusión de que, si se pretende ampliar a un número infinito de acciones, sería más conveniente hacerlo de otra forma, pues habría que modificar el script en función del número de componentes que se quisiera incluir.

Para aprovechar las características de la aplicación que se emplea, ArchestrA, se hará uso de booleanos, pues tiene una característica la cual es muy empleada a lo largo del desarrollo de la herramienta, y es que en los scripts que se incorporan, se permite escoger en base a que parámetros interesa que se ejecute, esto es, en la declaración de los scripts existen las siguientes opciones, como se muestra en la siguiente imagen.

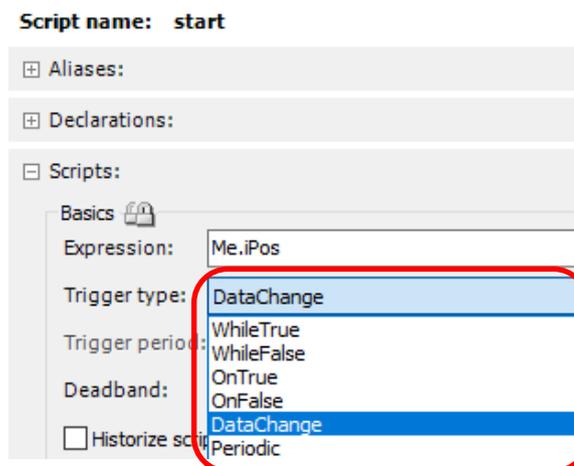
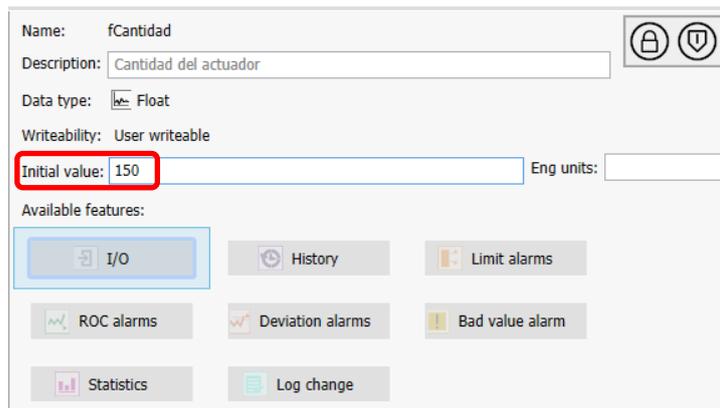


Ilustración 4.4 .- Opciones de los Scripts

Por ejemplo, en el caso actual, en el área Factoría, existe un atributo *iPos*, definido como *Integer*, el cual se emplea para controlar la ejecución del programa. Pues bien, siempre que el valor de *iPos* cambie, el script se ejecutará. Como muestra la imagen, hay otras opciones de *Trigger*, las más utilizadas en este proyecto son *DataChange* y *OnTrue*, pues como se menciona previamente y de cara a controlar los cambios de estado y de secuencia, se emplearán booleanos en prácticamente todos los scripts. Con esta segunda opción, únicamente se ejecutarán cuando el valor de la variable booleana sea cierto.

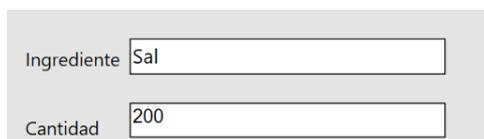
- ¿Como añadir en ArchestrA los datos correspondientes a los ingredientes, sus cantidades y los componentes sobre los que se actúa?

Se sabe que lo más adecuado para añadir de una forma rápida y eficaz todos los datos será a través una base de datos, pero inicialmente y con el fin de comprender el funcionamiento del sistema, se realizará de una forma rudimentaria y poco eficaz, que será añadiendo los datos a mano en los atributos del objeto, por lo tanto, estos valores serán fijos, como se muestra en la imagen siguiente.



*Ilustración 4.5 .- Opciones de los atributos*

En la modificación siguiente se realizó un cambio, pasando a introducir los datos mediante una entrada en tiempo de ejecución, a través un cuadro de texto en la visualización, esta solución permite modificar los componentes y sus valores, pero no es la forma más correcta, pues tendría que estar un operario siempre disponible para realizar esta acción, y eso no es lo que se busca, por lo que se decide, una vez se tenga claro cómo se va a trabajar con los datos, obtener los mismos de la base de datos.



*Ilustración 4.6 .- Lectura en tiempo de ejecución*

Con la conexión con la base de datos realizada, y de cara a realizar pruebas, se creó una tabla con 3 campos, como la siguiente. En este caso no se controlan los componentes sobre los que se actúa, únicamente se establece si se trata un movimiento de ingredientes o un proceso, y las cantidades correspondientes, pues solo se disponía dos tanques y un mixer.

I/P	$\Sigma$	Valor	$\Sigma$	Accion	$\Sigma$
<input checked="" type="checkbox"/>	I	10		leche	
<input type="checkbox"/>	I	10		Sal	
<input type="checkbox"/>	P	60		Mezclar	

*Ilustración 4.7 .- Primera base de datos*

Una vez se comprobó que funcionaba correctamente y con el fin de ampliar la planta, se crearon dos tablas que sustituían a la que había en ese momento, se corresponderán con la receta y la fórmula que habrá finalmente. En la receta se controla el componente que se va a realizar, la descripción, los componentes disponibles sobre los que actuará, el tipo de operación y las cantidades mínimas y máximas, entre otros atributos.

id	idReceta	Secuencia	Solapamiento	Descripcion	Componentes	Tipo	Actuador	CantidadMin	CantidadMax	Unidad
1	R001	10	no	Movimiento	T1,T2,M1	Entrada	Agua	0	100	L
2	R001	20	no	Operacion	M1,M2,M3	Start/Stop	Start			
3	R001	30	no	Movimiento	T1,T2,T3,M1,M2,M3	Entrada	Leche	0	300	L
4	R001	40	no	Operacion	M1,M2,M3	Consigna	Temperatura	0	200	°C
5	R001	50	no	Movimiento	T1,T2,T3,M1,M2,M3	Entrada	Sal	0	100	Kg
6	R001	60	no	Movimiento	T1,T2,T3,M1,M2,M3	Entrada	Leche	0	300	L
7	R001	70	no	Operacion	M1,M2,M3	Start/Stop	Stop			
8	R001	80	no	Movimiento	M1,M2,M3,TS1,TS2,TS3	Entrada	Vaciado	-1	-1	L
10	R001	90	no	Operacion	M1,M2,M3	Consigna	Presion	0	300	Var
11	R001	100	no	Movimiento	M1,M2,M3,TS1,TS2,TS3	Entrada	Vaciado	-1	-1	L

Tabla 4-1 .- Base de datos Receta

Mientras tanto, en la fórmula se establecerán los valores finales que tendrá cada elemento de la receta, si en esta se indicaba que el material a obtener podría estar en los tanques 1 o 2, en la fórmula se especificará que es en el 2, de la misma manera con el mixer, que dirá cual es concretamente y lo mismo con la cantidad del ingrediente, en vez de dar un rango de valores, nos mostrará el valor final.

id	idFomula	idReceta	Descripcion	Secuencia	ComponenteInicial	ComponenteFinal	Cantidad
1	F001	R001	Añadir agua	10	T1	M1	100
2	F001	R001	Start	20	M1		
3	F001	R001	Añadir leche	30	T2	M1	80
4	F001	R001	Temperatura	40	M1		180
5	F001	R001	Añadir sal	50	T3	M2	50
6	F001	R001	Añadir leche	60	T2	M2	100
7	F001	R001	Stop	70	M1		
8	F001	R001	Vaciado	80	M1	M2	-1
10	F001	R001	Presion	90	M2		150
11	F001	R001	Vaciado	100	M2	TS1	-1

Tabla 4-2 .- Base de datos Fórmula

Además, de esta manera será posible incorporar más de una fórmula para cada receta, pues estas van identificadas por un id único. Cuando se realiza la lectura se filtrará este id para así obtener fórmula correcta.

- ¿Cómo tratar los datos que se obtienen de la base de datos?

De cara a trabajar con los datos una vez se han leído existen 2 opciones, la primera será asignar por orden a los componentes cada valor, esto es, se comprobará si se trata de un movimiento de material o de un proceso, una vez sabe que es, se asignará por orden al primer componente, y así sucesivamente.

La segunda opción será crear *arrays* dinámicos de cada atributo y almacenar todos los valores en el área de lectura, donde posteriormente se irán asignando a los objetos correspondientes, pero en la aplicación de System Platform no es posible realizar *arrays* dinámicos como sí puede ser en otros lenguajes de programación, por lo que se realizarán dos lecturas a la base de datos, una primera mediante la cual se obtiene el número de líneas de la tabla, lo que indica el número de secuencias a realizar, que será la dimensión final de los *arrays*, y en la segunda lectura se asignará cada línea de la base de datos a una posición de los *arrays*.

Esta segunda opción será la que más adelante se empleará, y mediante variables indirectas se activarán los componentes necesarios. A lo largo del documento se mostrará la forma de realizar esto.

	id	idFormula	idReceta	Descripcion	Secuencia	ComponenteInicial	ComponenteFinal	Cantidad
1	1	F001	R001	Añadir agua	10	T1	M1	100
2	2	F001	R001	Start	20	M1		
3	3	F001	R001	Añadir leche	30	T2	M1	80

iSecuencia [1] = 10

sComponenteInicial [1] = T1

sComponenteFinal [1] = M1

fCantidad [1] = 100

iSecuencia [2] = 20

sComponenteInicial [2] = M1

sComponenteFinal [2] = vacío

fCantidad [2] = vacío

iSecuencia [3] = 10

sComponenteInicial [3] = T1

sComponenteFinal [3] = M1

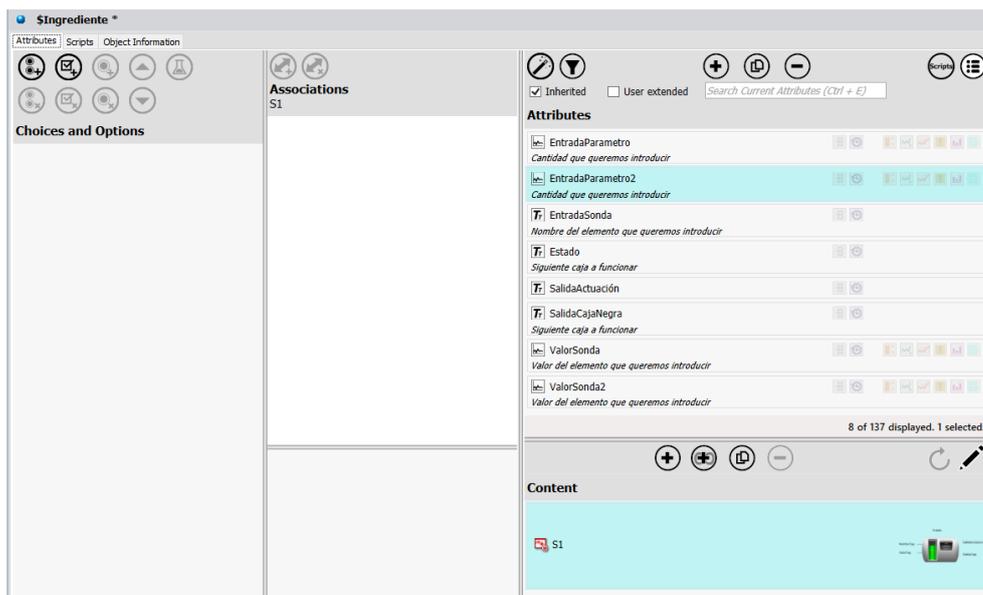
fCantidad [3] = 100

Tabla 4-3 .- Asignación de los datos

- ¿Cómo se diseñará la parte visual?

La visualización de la planta se ha realizado a la par que se iban incluyendo mejoras en la solución propuesta. La primera pantalla que se creó fue la de una válvula, donde se simulaba el sistema de estamos mencionado previamente.

Para esta primera pantalla, se creó un nuevo símbolo donde se añadieron los componentes necesarios a mano. Más adelante se comprobó que no es la forma más correcta de hacerlo, pues de cara a incorporar un elevado número de componentes, sería muy lento hacerlo de esa forma, por lo que se decide crear los símbolos en las plantillas para así cuando se creen nuevas instancias, estas ya tengan incluido su símbolo correspondiente.



*Ilustración 4.8 .- Creación de símbolos en la plantilla*

Además, se crearon símbolos tipo Wizard, los cuales son altamente modificables permitiendo cambiar ciertas características de manera muy rápida. Por ejemplo, en la imagen que se muestra a continuación, hay 2 opciones, la primera para escoger el número de tanques y la segunda para el número de mixers. En el símbolo habrá creados un número finito de estos elementos, pero mediante estas opciones se permitirá escoger si se desea mostrar todos o solo unos pocos.

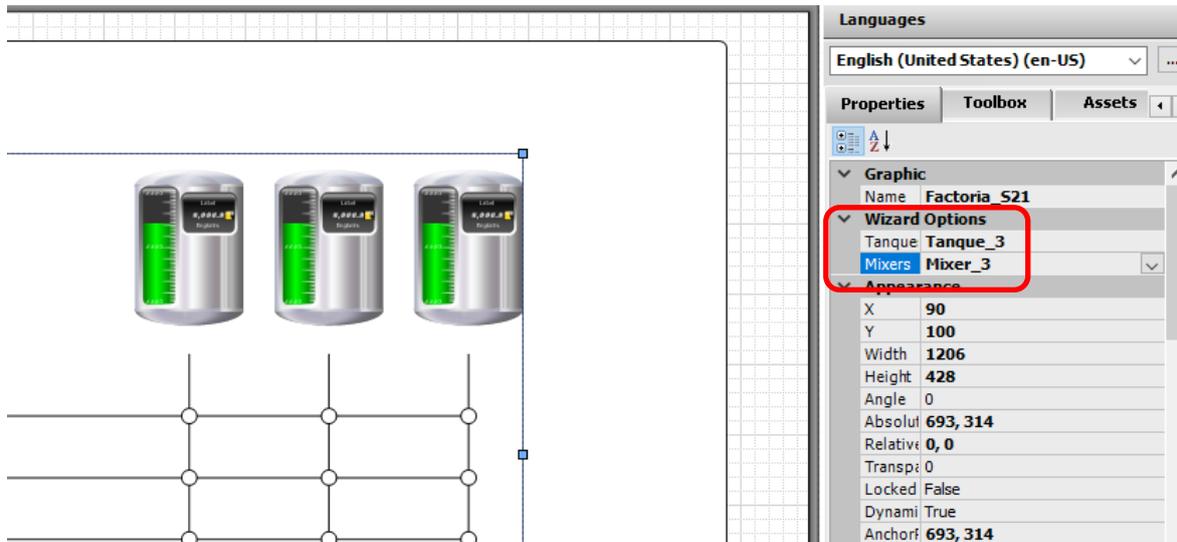
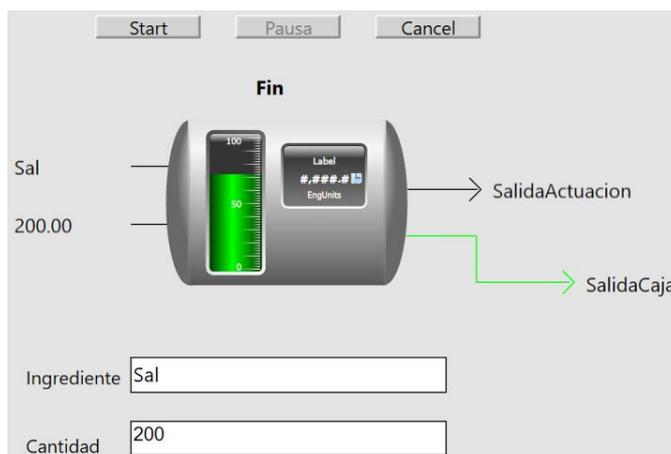


Ilustración 4.9.- Opciones Wizard

También surgió la duda sobre qué tipo de símbolos emplear, pues en ArcestrA existen numerosas bibliotecas con infinidad de opciones. Inicialmente no se tiene en cuenta esta cuestión y se crearon visualizaciones como la anterior, con elementos en 3D y muy coloridos, pero con el paso del tiempo se comprueba que lo mejor será emplear la biblioteca Situational Awareness, la cual se ha descrito previamente en el punto 3.6.- de la memoria. De esta forma estarán todos los símbolos acordes unos con otros manteniendo un mismo formato.

Las visualizaciones por las que se ha ido pasando son las siguientes:

- En primero lugar solo existía una válvula, en la cual se incorporaban los datos de manera manual mediante lo *TextBox*.



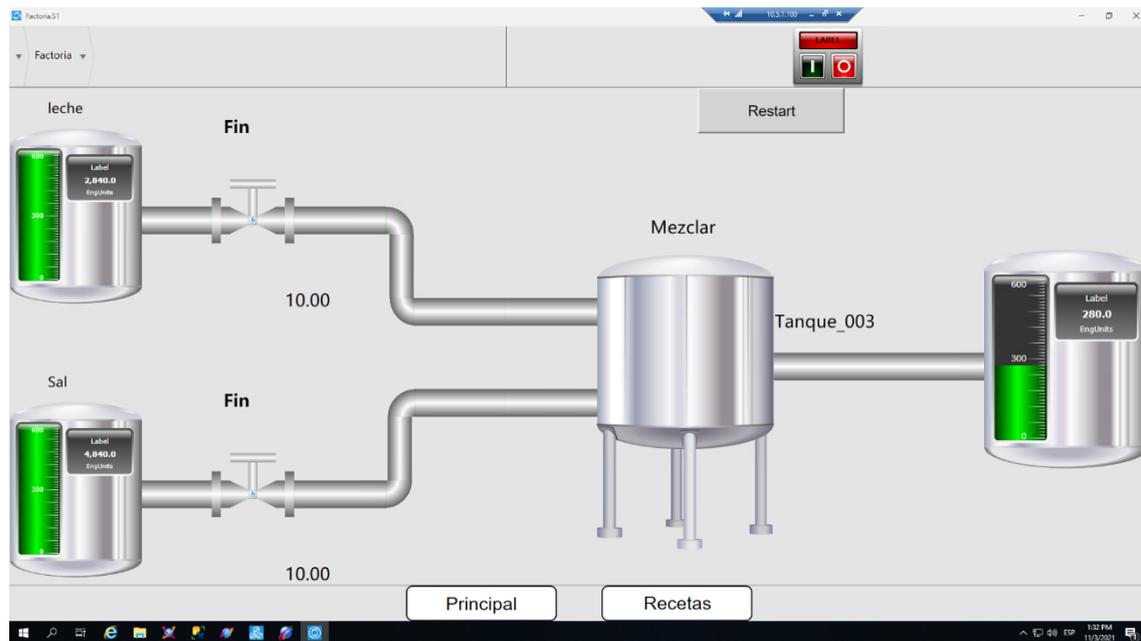
*Ilustración 4.10 .- Primera visualización*

El siguiente paso fue incorporar dos tanques de entrada y dos válvulas, las mismas que se habían incorporado previamente, pues en este momento únicamente se podía obtener material de ambos tanques y hacerlo pasar por sus válvulas correspondientes, siendo los valores siempre fijos.



*Ilustración 4.11 .- Visualización con dos tanques y dos válvulas*

Posteriormente se consiguió incorporar un mixer donde se realizaba la primera y única operación. Además, se cambió el símbolo correspondiente a las válvulas, pues ya dejó de ser la única acción que se realizaba y también se añadió un tanque de salida, donde se almacenaba todo el material resultante.



*Ilustración 4.12 .- Primera vez que se añade un mixer*

La siguiente visualización se acerca mucho a la final, pues en ella se incorporan ya todos los elementos, incluyendo varios de cada tipo. En este punto se recogían ya recetas de todo tipo, permitiendo escoger entre que tanques queríamos obtener el material, a que mixers queríamos que se añadiera, así como la operación que se iba a realizar, y por último donde deseábamos almacenar el producto resultante.

También se incorporó otra pestaña, Lectura, donde se mostraba la base de datos correspondiente a la receta establecida.

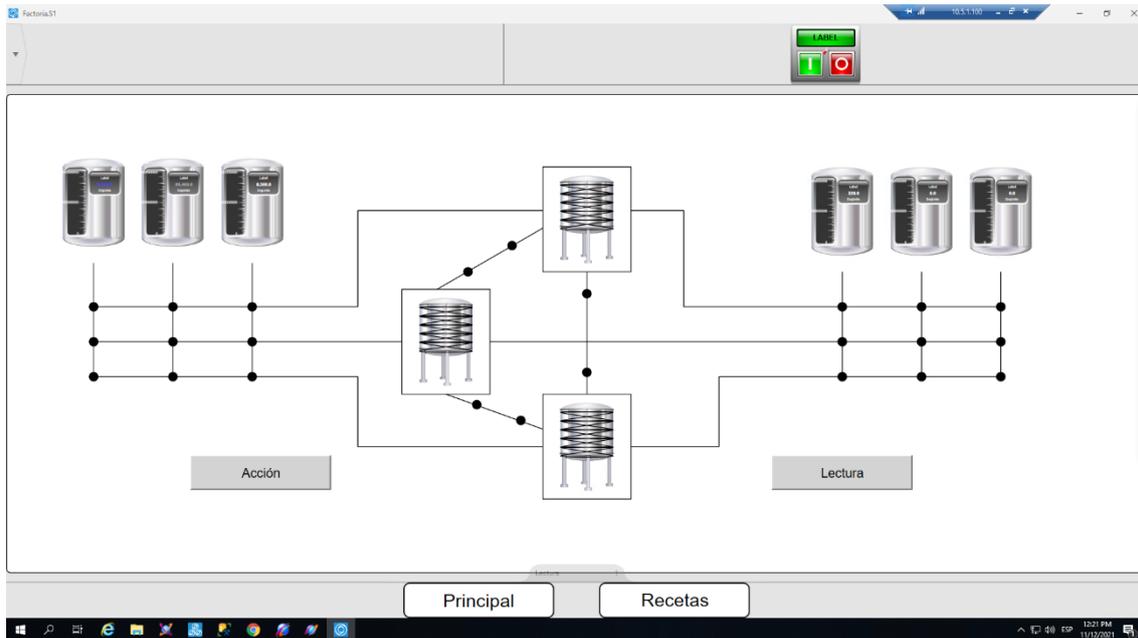


Ilustración 4.13 .- Visualización completa no final

El último cambio que se realizó fue la estandarización de los símbolos a la biblioteca Situational Awareness mencionada previamente y el cambio de la ventana de Lectura, donde se permite escoger si queremos visualizar la receta o la fórmula en la base de datos

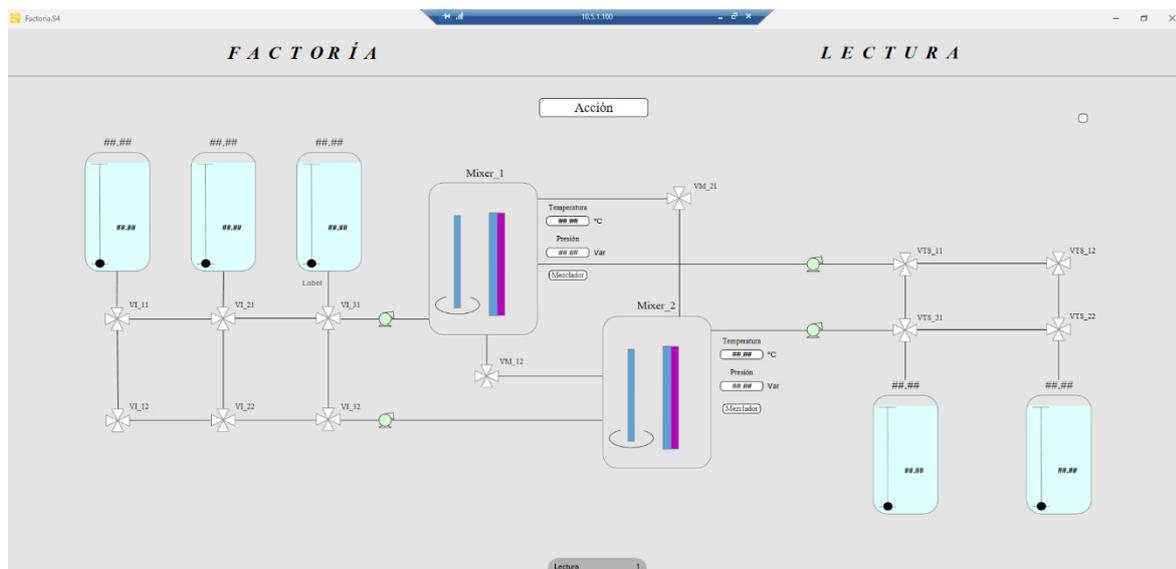


Ilustración 4.14 .- Visualización final Factoría

FACTORIA										LECTURA			
Id	idReceta	Secuencia	Solapamiento	Descripcion	Componentes	Tipo	Actuador	CantidadMin	CantidadMax	Unidad			
1	R001	10	no	Movimiento	T1.T2.M1	Entrada	Agua	0	100	L			
2	R001	20	no	Operacion	M1.M2.M3	Start/Stop	Start						
3	R001	30	no	Movimiento	T1.T2.T3.M1.M2...	Entrada	Leche	0	300	L			
4	R001	40	no	Operacion	M1.M2.M3	Consigna	Temperatura	0	200	°C			
5	R001	50	no	Movimiento	T1.T2.T3.M1.M2...	Entrada	Sal	0	100	Kg			
6	R001	60	no	Movimiento	T1.T2.T3.M1.M2...	Entrada	Leche	0	300	L			
7	R001	70	no	Operacion	M1.M2.M3	Start/Stop	Stop						
8	R001	80	no	Movimiento	M1.M2.M3.TS1...	Entrada	Vaciado	-1	-1	L			
10	R001	90	no	Operacion	M1.M2.M3	Consigna	Presion	0	300	Var			
11	R001	100	no	Movimiento	M1.M2.M3.TS1...	Entrada	Vaciado	-1	-1	L			

Ilustración 4.15 Visualización final Lectura

## 4.2.- CREACIÓN DE LA GALAXIA

Lo primero que se ha de realizar será abrir el programa que en el que se trabajará, System Platform, y lo primero será crear una nueva galaxia vacía, la cual se denominará TFG\_InBatch.

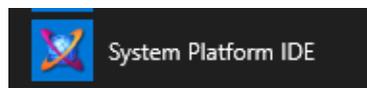


Ilustración 4.16 .- Aplicación de escritorio

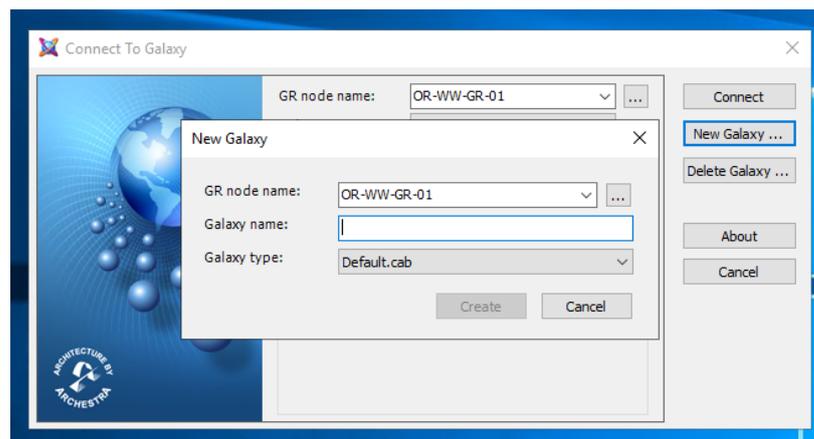
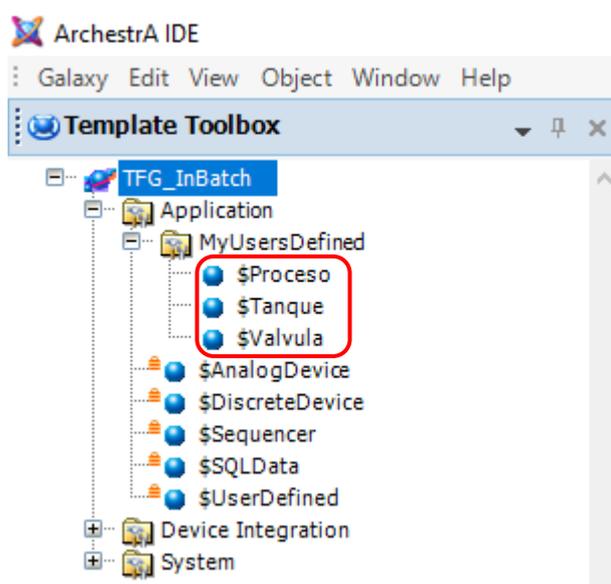


Ilustración 4.17 .- Creación de la galaxia

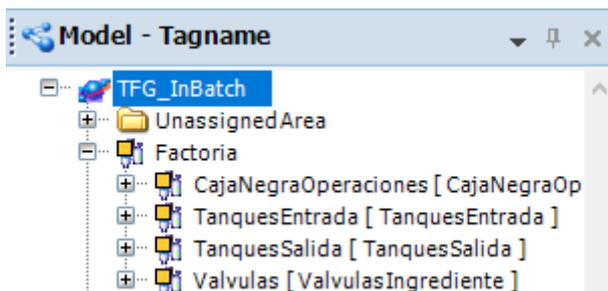
Inicialmente la galaxia estará compuesta por las plantillas predefinidas \$Area y \$UserDefined, que serán las que se usarán para realizar toda la configuración correspondiente a la planta industrial, además de \$AppEngine y \$WinPlatfor, pues son los motores de la herramienta, y por último \$ViewApp y \$ViewEngine, necesarios para implementar y mostrar las visualizaciones. De todas estas plantillas se crearán instancias u otras plantillas, pues estas están protegidas y no es posible realizar las configuraciones en ellas.



*Ilustración 4.18 .- Plantillas creadas*

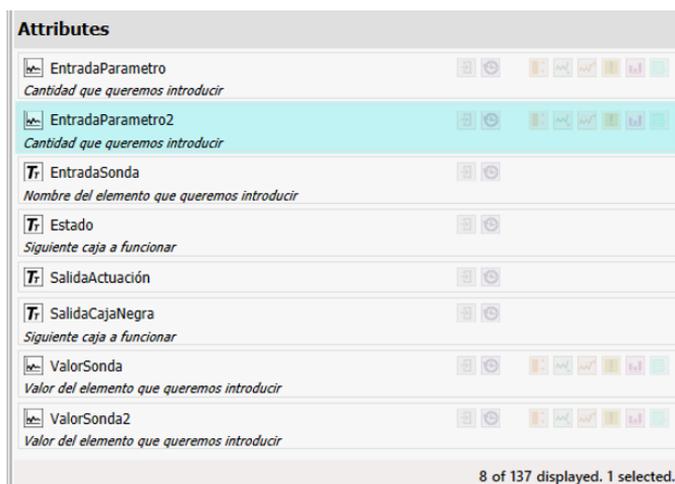
De cara a crear los componentes que se van a emplear habrá que realizar nuevas plantillas derivadas de \$UserDefined, una por componente, por lo que se creará \$Valvula, \$Tanque y \$Mixer. Así cada vez que se quiera incorporar un nuevo componente de este tipo, solo será necesario crear una instancia de la plantilla correspondiente, la cual ya dispondrá de las configuraciones que se hayan definido en ella.

También es necesario definir nuevas áreas, estas establecerán la ubicación física de los componentes en la planta. Se creará una donde se almacenarán todos los componentes, Factoría, y dentro de ella se incorporarán otras 4, una para agrupar las válvulas, otras para los tanques de entrada, para los de salida y por último otra para los mixers, por lo que la configuración final será como se muestra en la ilustración siguiente.



*Ilustración 4.19 .- Áreas creadas*

Una vez se han definido las plantillas correctas, habrá que configurarlas, creando los atributos necesarios en cada caso, además, se han de incorporar los scripts correspondientes donde se establecerá el comportamiento que realizará posteriormente la instancia correspondiente.



*Ilustración 4.20 .- Atributos de una plantilla*

### 4.3.- CREACIÓN E IMPLEMENTACIÓN DE LA PRIMERA VÁLVULA

Una vez se crearon las plantillas y las áreas, se simuló el funcionamiento del primer componente creado, una válvula, permitiendo establecer el nombre del material, su cantidad y la siguiente acción a realizar. La forma de crear dicho componente será haciendo una instancia de la plantilla e incorporándola a su área correspondiente.

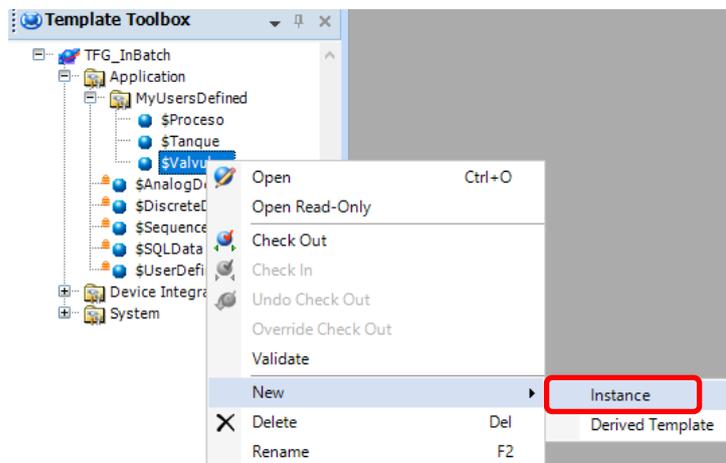


Ilustración 4.21 .- Creación de un objeto

Como se explicó previamente, se realizó la implementación de un sistema de estados, ver *Ilustración 4.3*. Con este script completo, se conseguía simular el movimiento de material en una única válvula, estableciendo el nombre del material mediante una caja de texto. En ella, mediante script, se realizaban las siguientes comprobaciones:

- Que el ingrediente exista en nuestra lista.
- Que disponemos en nuestro almacén de la cantidad solicitada de ese ingrediente.
- Que el tanque almacén no se va a quedar completamente vacío.
- Que la capacidad del mixer es superior a la cantidad que vamos a introducir.

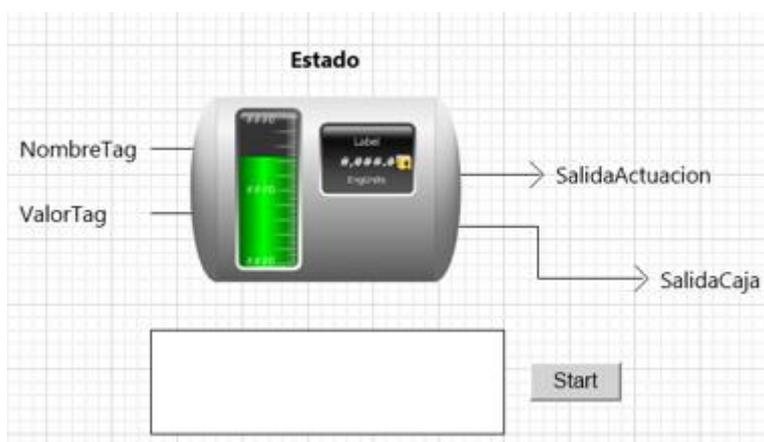
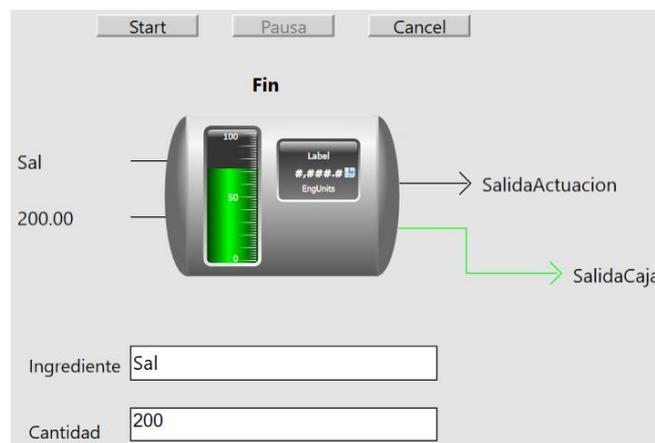


Ilustración 4.22 .- Símbolo inicial

Además, se realizó un primer símbolo donde se incorporó un cilindro que simulaba la válvula, con sus entradas y salidas y un nivel para ver la cantidad que circulaba por ella. Ver *Ilustración 4.23*. También se añadió un botón “Start”, a través del cual comenzaba la ejecución.

En este momento, cuando se pulsa sobre el botón “Start”, comienza la ejecución, realizando las comprobaciones definidas previamente, si todo es correcto finalizará correctamente, en caso de no ser así, se mostrará por pantalla que ha ocurrido un error y se cancelará la ejecución.



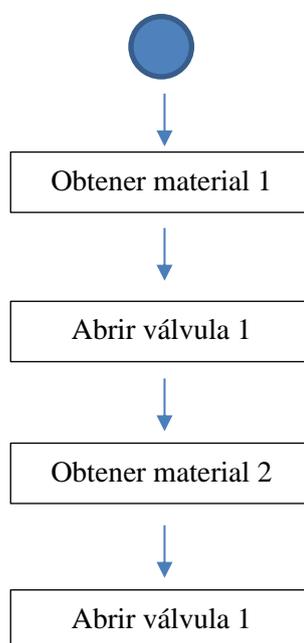
*Ilustración 4.23 .- Símbolo inicial finalizado con éxito*



*Ilustración 4.24 .- Símbolo Inicial cancelado*

#### 4.4.- CREACIÓN DE NUEVOS COMPONENTES

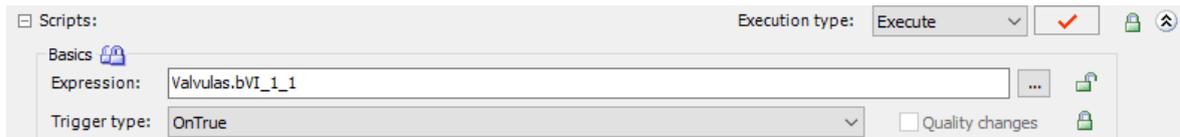
Una vez se comprobó el funcionamiento correcto de lo realizado hasta el momento, se añadieron más componentes a la planta. Dos tanques, únicamente de entrada y otra válvula. Con estos componentes se pretende simular el esquema mostrado en la *Ilustración 4.26*, donde se realiza lo siguiente, obtener el material del primer tanque, enviarlo por la primera válvula y que cuando este primer proceso finalice, obtener el material del segundo tanque y abrir la segunda válvula.



*Ilustración 4.25 .- Esquema actual de ejecución*

El punto crítico de este paso se encuentra en conseguir que cuando la primera válvula finalice, comience la segunda automáticamente. Esto se definió en el área Factoría mediante un script que controlaba el comienzo y cambio de una válvula a otra.

Cuando la primera válvula mostraba la flecha verde, indicando que había finalizado correctamente, comenzaba la extracción de material del segundo tanque hacia la segunda válvula. La Factoría controlaba los cambios mediante booleanos, los cuales se empleaban en las expresiones que daban pie a la ejecución de los scripts.



*Ilustración 4.26 .- Expresión para la ejecución del script*

Pero existían varios problemas, y es que el programa solo permitía tener estos cuatro elementos y siempre seguía la misma secuencia, obtener material del tanque 1, abrir válvula 1, obtener material del tanque 2 y abrir válvula 2, si se pretendía añadir algún componente más daría problemas, pues el sistema no estaba preparado para ello. Además, aún no se realizaba ninguna acción, lo cual es parte indispensable en la planta. Y, por último, las cantidades y materiales que actuaban eran fijos, no era posible modificarlos de una manera sencilla.

Por lo tanto, no se podía quedar así, aunque se habían conseguido algunos avances, pues ya se sabía cómo realizar los cambios de secuencia, por lo que, de cara a añadir más componentes y secuencias ya se sabía cómo realizarlo.

Así mismo, se realizó la visualización correspondiente, pero esta vez se crearon los símbolos en las plantillas, pues había dos componentes de cada tipo así, cuando creamos la pantalla final, lo único que hizo falta fue incorporar los que necesitábamos, buscándolos en los objetos.

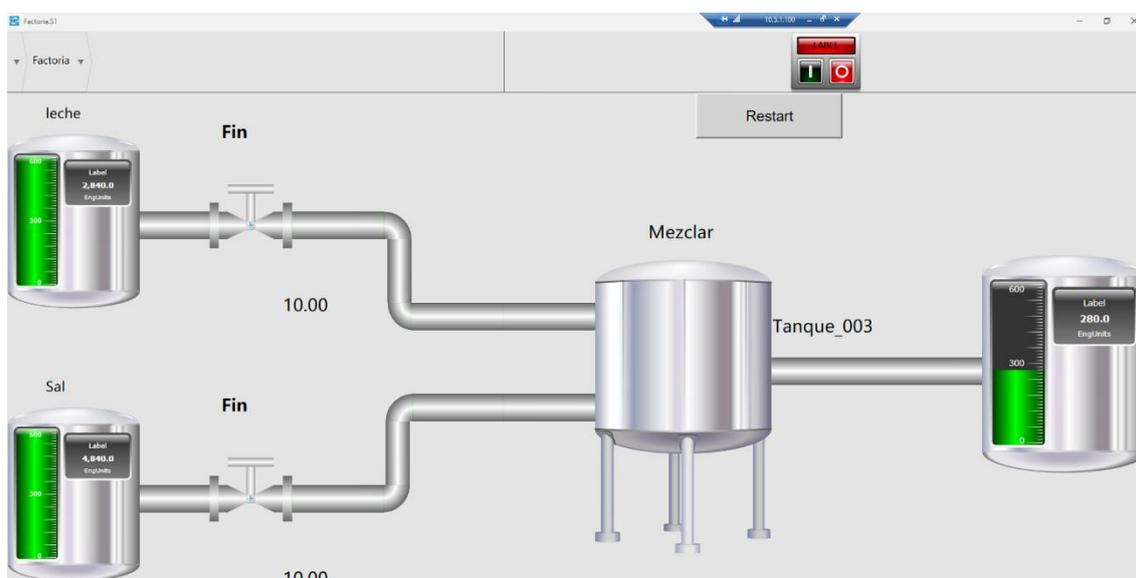


*Ilustración 4.27 .- Visualización con dos tanques y dos válvulas*

El siguiente paso realizado fue incorporar el primer mixer y tanque de salida, así se tendría una receta prácticamente completa, aunque fija, la cual se ampliará en el futuro.

Para ello se crearán las instancias correspondientes: un mixer, que derivará de la plantilla \$Proceso y un tanque, de \$Tanque, y al igual que antes, el cambio de una secuencia se controlaba mediante booleanos en el área Factoría, donde se comprobaba si lo que se iba a hacer era un movimiento de material, donde actuarían los tanques de entrada y válvulas, o bien un proceso, donde entraría en acción el mixer.

Por lo tanto, la visualización correspondiente quedó de esta manera, ver *Ilustración 4.28*.



*Ilustración 4.28 .- Visualización con mixer y tanque de salida*

## 4.5.- PRIMERA CONEXIÓN CON LA BASE DE DATOS

Una decisión muy importante que se toma es la de que las recetas y por ello, los datos de los ingredientes, cantidades, operaciones, etc. Han de añadirse mediante una base de datos. De esta manera será mucho más sencillo insertar nuevas recetas o fórmulas, así como editar las existentes. Para realizar la lectura, se creó una nueva área, llamada Lectura, que estará al mismo nivel que Factoría, y todo lo realizado estará en ella.

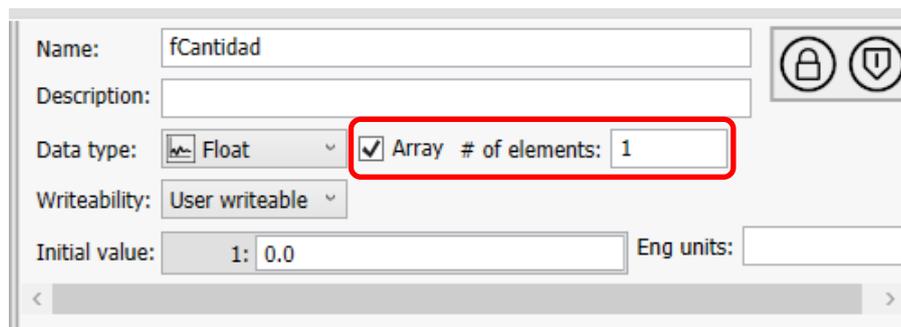
La primera receta que se incorporó iba asociada al planteamiento que había en ese momento, correspondiente a la *Ilustración 4.28*, por lo que se incluirían dos ingredientes con sus cantidades y una operación a realizar. Únicamente estaría formada por 3 secuencias, la receta correspondiente se puede ver en la *Ilustración 4.7*.- Primera base de datos.

La forma de conectar SQL con ArchestrA es la siguiente y se realiza por script: habrá que crear 3 variables propias del sistema que heredan de la biblioteca de SQL ya incorporada en la herramienta. Se creará la conexión con la primera de las variables, *connection*, pasándole los datos de base nuestra base de datos. Posteriormente se enviará la consulta que se desea realizar, mediante *command*, y finalmente se obtendrá el resultado mediante la variable *reader*. Si el resultado de la lectura es más de un elemento habrá que leerla a través de un bucle *while* para obtener línea a línea.

```
1 dim i as integer;
2
3 'Crear conexiones con la base de datos
4 dim connection as System.Data.SqlClient.SqlConnection;
5 dim command as System.Data.SqlClient.SqlCommand;
6 dim reader as System.Data.SqlClient.SqlDataReader;
7
8 dim connection1 as System.Data.SqlClient.SqlConnection;
9 dim command1 as System.Data.SqlClient.SqlCommand;
10 dim reader1 as System.Data.SqlClient.SqlDataReader;
11
12 dim DBServer as String;
13 dim DBName as String;
14 dim DBUser as string;
15 dim DBPwd as string;
16
17 dim MyQuery as string;
18 MyQuery = "";
19
20 DBName = "localhost";
21 DBName = "Recetario";
22 DBUser = "sa";
23 DBPwd = "Aeioul234";
24
25
26 'Consulta para obtener el numero de elementos de una receta
27 connection1 = new System.Data.SqlClient.SqlConnection("Server=" + DBServer + ";Database=" + DBName + ";UID=" + DBUser +
28 connection1.Open();
29 command1 = new System.Data.SqlClient.SqlCommand("SELECT Count(*) FROM Formulas WHERE idFormula = 'F001'",connection1);
30 reader1 = command1.ExecuteReader();
31 reader1.Read();
```

*Ilustración 4.29*.- Variables para la conexión a SQL

Para almacenar los datos leídos se crearon atributos tipo array en el área con la misma dimensión que secuencias tuviera la receta, así cuando se realiza la lectura, se añadirá a cada posición del array el valor correspondiente.



*Ilustración 4.30 .- Creación de atributos como arrays*

En este punto se tienen los datos con lo que se trabajará almacenados en atributos de la galaxia, pero será necesario asignarlos a los objetos para poder trabajar con ellos, para eso se creó un *array* con el nombre de los objetos existentes y empleando la función *bindTo* propia de VisualBasic.net se permite asignar los valores de los atributos a los objetos correctos.

```
dim ptr as indirect;
ptr.BindTo(Me.sComponenteF+".fCantidad");
cantidadF = ptr;
```

*Ilustración 4.31 .- Declaración y uso de una variable indirecta*

En este caso que se muestra en la imagen superior, el script se encuentra declarado en una plantilla, así que cuando se llame a *Me.sComponenteF*, se refiere al atributo *sComponenteF* correspondiente al objeto *Me*. Por lo que modificará el valor sin necesidad de incluir su nombre y se podrá reutilizar para un número infinito de componentes.

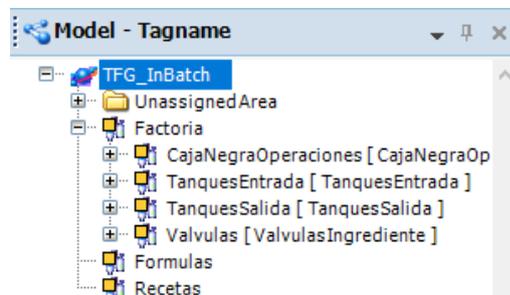
Llegados a este punto se tienen dos hitos conseguidos, como son la capacidad de introducir mediante SQL los datos de la planta, lo cual indirectamente se refleja en que no es necesario un operario que se encuentre constantemente en la planta introduciendo los valores, si no que una vez que se introducen los datos, la planta podrá comenzar la ejecución cuando se requiera. Y, además, esos valores leídos de la base de datos, se asignan de forma automática a los objetos que los van a emplear, mediante la función *bindTo* mencionada previamente.

En el siguiente paso se crearán nuevas tablas con la información realmente relevante y detallada.

#### 4.6.- CREACIÓN DE LAS TABLAS FINALES

De cara a implantar esta solución en una planta real será necesario conocer exactamente las secuencias que se van a realizar, así como las cantidades que se manejarán y los componentes sobre los que se actuará, para ello se crearon 2 nuevas tablas, las cuales ya se mencionaron previamente en el punto 4.1.- de la memoria.

Para manejar todos estos datos se emplean 2 áreas, una para cada tabla, Fórmulas y Recetas. El proceso de lectura será el mismo que se ha mencionado en el punto anterior, mediante scripting, pero modificando la consulta para así obtener los datos de la tabla correcta, como se muestra en la *Ilustración 4.32*.



*Ilustración 4.32 .- Áreas de la galaxia*

Para almacenar los datos se emplearán *arrays* dinámicos, como se explicó en el punto 4.1.- de la memoria. Una vez los datos están su área correspondiente, se realizarán una serie de comprobaciones para verificar que la receta y la fórmula guardan relación, pues todos los componentes de la fórmula han de estar incluidos en la receta, así como las cantidades que se manejan, en la receta se establecerá un rango de valores y en la fórmula habrá un único valor, el cual ha de estar comprendido en ese rango.

```

for i = 1 to Me.x
    if Recetas.iSecuencia[i] <> Me.iSecuencia[i] then ←
        Factoria.bFormulaCorrecta = false;
        LogMessage("Error en la secuencia ["+i+"]");
    else;
        'Valor correcto de la secuencia
    endif;

    comp = Recetas.sComponentes[i];
    if comp.Contains(Me.sComponenteInicial[i]) and comp.Contains(Me.sComponenteFinal[i]) then
        'Valor correcto de los componentes ↑
    else;
        Factoria.bFormulaCorrecta = false;
        LogMessage("Error en los componentes ["+i+"]");
    endif;

    if Me.fCantidad[i] <= Recetas.fCantidadMax[i] then ←
        'Valor correcto de la cantidad max

        if Me.fCantidad[i] >= Recetas.fCantidadMin[i] then ←
            'Valor correcto de la cantidad min

        else;
            Factoria.bFormulaCorrecta = false;
            LogMessage("Error en la cantidad min ["+i+"]");
        endif;

    else;

```

*Ilustración 4.33 .- Comprobaciones de la fórmula*

En caso de producirse un error, el sistema mostrará una serie de mensajes informando donde se ha producido el problema y no se permitirá comenzar la ejecución hasta que este no se corrija.

```

'Si la fórmula es modificada y no es correcta, no podremos accionar el proceso.
if Me.bFormulaCorrecta == true then

```

*Ilustración 4.34 .- Si la fórmula no es correcta no permite la ejecución*

Como el número de componentes sobre lo que se actuará puede variar en función de la planta en la cual se implante la solución, para esta simulación, únicamente se crearon 3 tanques de entrada y de salida, 2 mixers y 12 válvulas, las cuales interconectan todos los componentes entre ellos.

Con esto se finaliza la configuración de la planta, permitiendo incluir recetas y fórmulas de longitudes variables y finalizando de manera correcta.

#### 4.7.- COMIENZO DE LA EJECUCIÓN

Una vez se han realizado las comprobaciones correspondientes a la lectura de los datos introducidos y se ha mostrado que todo es correcto, se permitirá comenzar la ejecución del sistema. El principal elemento de control de la ejecución será un atributo de tipo *Integer* declarado en el área Factoría, este será *iPos*, es lo más útil ya que al trabajar con *arrays* es necesario acceder a sus posiciones en un orden concreto.



*Ilustración 4.35 .- Configuración script de factoría*

Cuando la ejecución está lista para comenzar y se pulse sobre el botón correspondiente al arranque, el valor de este atributo cambiará, pudiendo establecerse un valor negativo en caso de que surja algún imprevisto, o bien, con el valor 1 si es correcto. De esta forma comenzará la ejecución, además en este mismo instante, se restaurarán los valores de la planta, por ejemplo, la cantidad almacenada en los mixers o en los tanques de salida se inicializada a 0, simulando un vaciado, y así disponer de todo el almacenamiento posible.

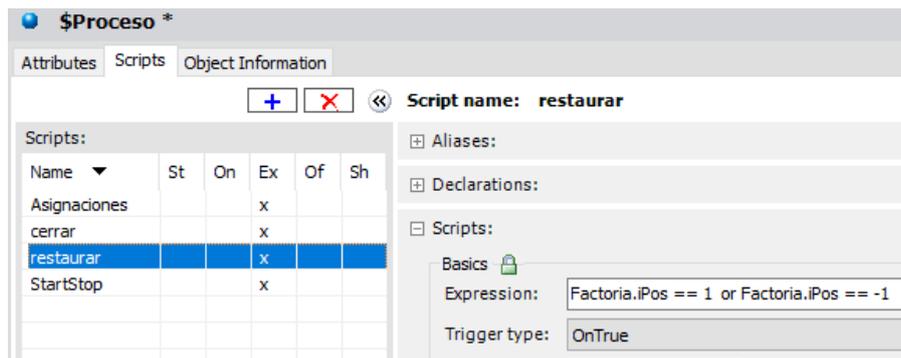


Ilustración 4.36 .- Script restauración de valores

Si no surge ningún error se comprobará en Factoría que tipo de acción se va a realizar, siendo posible hacer un movimiento o una operación, una vez se sabe cuál va a ser la acción, se activará el booleano correspondiente y llevará la ejecución a su área correspondiente.

```

if Recetas.sDescripcion[Me.iPos] == "Movimiento" then
    LogMessage("Movimiento de material.");
    Me.bIngrediente = true;
else;
    if Recetas.sDescripcion[Me.iPos] == "Operacion" then
        LogMessage("Operación.");
        Me.bOperacion = true;
    else;
        endif;
    endif;
endif;

```

Ilustración 4.37 .- Código comprobación de la acción

En caso de ser un movimiento de material, se abrirá la válvula que conecta los componentes que intervienen en la secuencia, de forma que deje pasar una cantidad concreta de material. Para ello, primero que se hace será asignar los valores correspondientes a la línea de ejecución que vamos a realizar a los atributos del área válvulas.

```

'Asignamos los valores de la secuencia actual.
Me.x = Factoria.iPos;
Me.sEntrada = Formulas.sComponenteInicial[Me.x];
Me.sSalida = Formulas.sComponenteFinal[Me.x];
Me.fCantidad = Formulas.fCantidad[Me.x];
Me.sActuador = Recetas.sActuador[Me.x];

```

Ilustración 4.38 .- Asignación de valores

Así, a través de variables indirectas, se abrirá la válvula que conecte el componente del cual va a salir el material, que estará en el atributo *Me.sEntrada* y el que lo va a recibir, *Me.sSalida*, estas serán las posibilidades de movimiento entre componentes:

POSIBILIDADES:	
1. Tanque Entrada X	- Mixer X
2. Mixer X	- Mixer X
3. Mixer X	- Tanque Salida SX

*Ilustración 4.39 .- Posibles movimientos de material*

A continuación se mostrará un ejemplo para entenderlo mejor: Se supone que en la variable *Me.sEntrada* se encuentra el valor T1 y en *Me.sSalida* M2. En ArchestrA no es posible recorrer un atributo, por lo que se asignarán ambos valores a variables locales creadas en el script y se comprobará cual es el primer elemento de la entrada, *Me.sEntrada.Char(0)*, será 'T', por lo que ya se sabe que el material saldrá de un tanque, y como se muestra en la Ilustración 4.39, la única posibilidad es enviarlo a un mixer, por lo que estamos en el primer caso.

```
if entrada.Chars(0) == "T" then
```

*Ilustración 4.40 .- Comprobación primer elemento*

Ahora mediante la variable indirecta se activará el objeto correspondiente a la válvula que conecta ambos componentes.

```

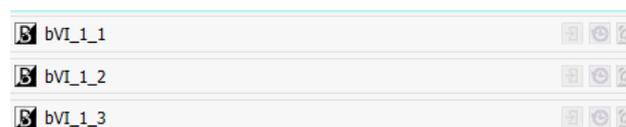
if entrada.Chars(0) == "T" then
    'Conexión Tanque - Mixer
    ptr.BindTo("Me.bVI_" + entrada.Chars(entrada.Length-1) + "_" + salida.Chars(salida.Length-1));
else;
    if salida.Chars(0) == "M" then
        'Coneion Mixer - Mixer
        ptr.BindTo("Me.bVM_" + entrada.Chars(entrada.Length-1) + "_" + salida.Chars(salida.Length-1));
    else;
        'conexión Mixer - Tanque
        ptr.BindTo("Me.bVMS_" + entrada.Chars(entrada.Length-1) + "_" + salida.Chars(salida.Length-1));
    endif;
endif;

'Activamos la válvula correspondiente.
ptr = true;

```

*Ilustración 4.41 .- Activación variable mediante variable indirecta*

En este momento la ejecución se encuentra en el área Válvulas, donde hay definido un atributo booleano por cada válvula creada, por lo que a través del comando *Me*. permitirá activar la válvula especificada en cada caso. De la misma forma que se obtuvo el primer elemento de la variable de entrada, será obtendrá el número de componente al cual va dirigido el material. En el supuesto anterior sería tal que así, el material parte del tanque 1, ya que  $entrada.Chars(entrada.Length-1) = 1$ , y se almacenará en el mixer 2,  $salida.Chars(salida.Length - 1) = 2$ . Así que se activará la variable booleana correspondiente a *Me.bVI\_1\_2*.



*Ilustración 4.42 .- Atributo de control del objeto VI\_1\_2*

Cuando esta variable se establece como cierta, enviará la ejecución al objeto VI\_1\_2, el cual ejecutará el script que se encuentra en la plantilla de válvulas. Este obtiene el material del tanque para introducirlo en el mixer correspondiente. Además, al igual que en los inicios, se realizarán las mismas comprobaciones sobre los tanques y las cantidades a mover.

Una vez que finaliza el movimiento del material, se cerrará la válvula de la misma manera que se abrió y se incrementará la variable *iPos*, de manera que comience la ejecución de la siguiente línea de nuestra tabla, llevando la ejecución de nuevo al área factoría.

Al incrementarse el valor de *iPos*, la ejecución volverá al área factoría, *Ilustración 4.37*, y de nuevo comprobará si la siguiente acción a realizar se trata de un movimiento o una operación, ver *Ilustración 4.39*. Si la segunda secuencia se tratara de una operación seguiría el mismo proceso que el movimiento de material, pero esta vez activando el booleano de operaciones, llevando la ejecución a su área correspondiente, en este caso será *CajaNegraOperaciones*.

Actualmente hay definidas 3 operaciones en nuestra planta, Start/Stop, la cual activa o desactiva el proceso de mezclado del mixer, y las otras 2 son consignas, de temperatura y presión, que finalizan cuando el mixer alcanza la cantidad introducida. Como cada planta puede realizar diferentes acciones no se han creado más operaciones, pero se realizaría de una manera muy sencilla, pues, al estar en la plantilla \$Proceso, todo lo que se incluya en ella se propagará a las instancias de esta, que son los mixers.

```
'Indicar el tipo. Si es consigna o start/Stop.  
  
if Me.sTipo == "Start/Stop" then  
    'Si es de tipo Start/Stop  
  
    ptr.BindTo("Mixer_"+componente.Chars(componente.Length-1)+".bStartStop");  
    'Si estaba en Stop se pone a Start, y viceversa.  
    if ptr == false then  
        ptr = true;  
    else;  
        ptr = false;  
    endif;  
  
else;  
    'Es otro tipo, por lo que se activa el objeto mixerX y se ejecuta.  
    ptr.BindTo("Me.bMixer_"+componente.Chars(componente.Length-1));  
    ptr = true;  
  
endif;
```

*Ilustración 4.43 .- Ejecución de una operación*

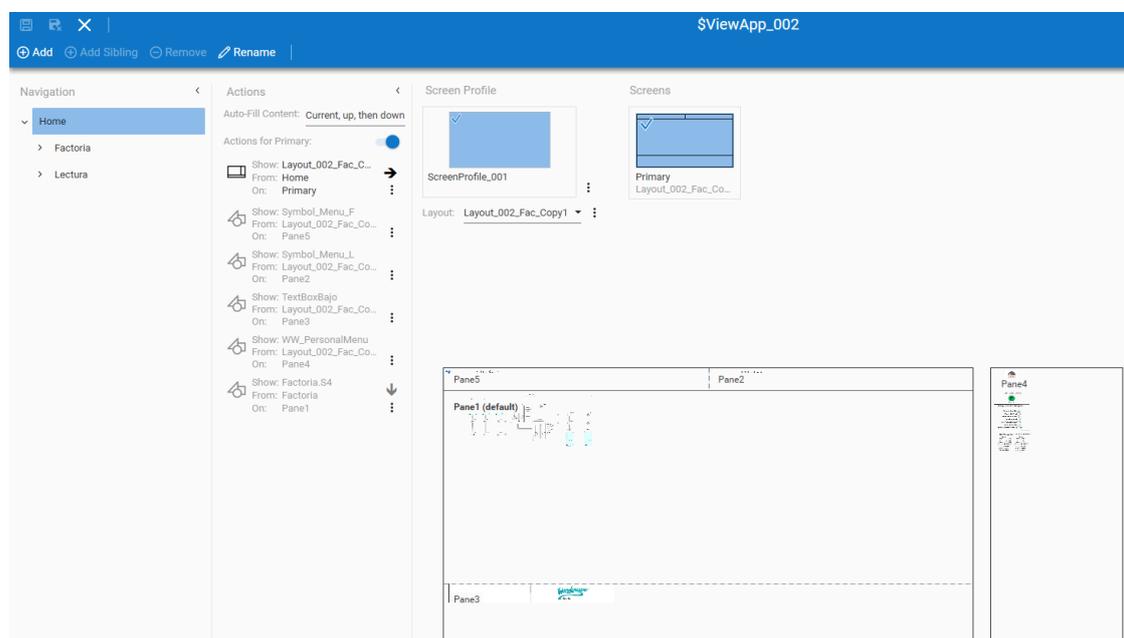
Al igual que en las válvulas, será necesario obtener el componente sobre el cual se actual y activarlo a través de la variable indirecta. En este caso solo se accionará uno ya que las operaciones se realizan únicamente en los mixers. En la simulación, las consignas se realizarán mediante un bucle el cual se incrementará hasta alcanzar el valor requerido, pero de cara a implantarlo en una planta real esto se controlaría a través de PLCs los cuales estarían constantemente obteniendo datos reales.

Una vez que finaliza la operación, de nuevo se incrementará el valor de *iPos*, por lo que la ejecución volverá al área factoría y se repetirá el mismo proceso una y otra vez hasta finalizar la fórmula.

Con esto se alcanza el final del desarrollo del software, pues ya es posible incluir recetas y fórmulas variables consiguiendo finalizar con éxito con sus respectivas comprobaciones.

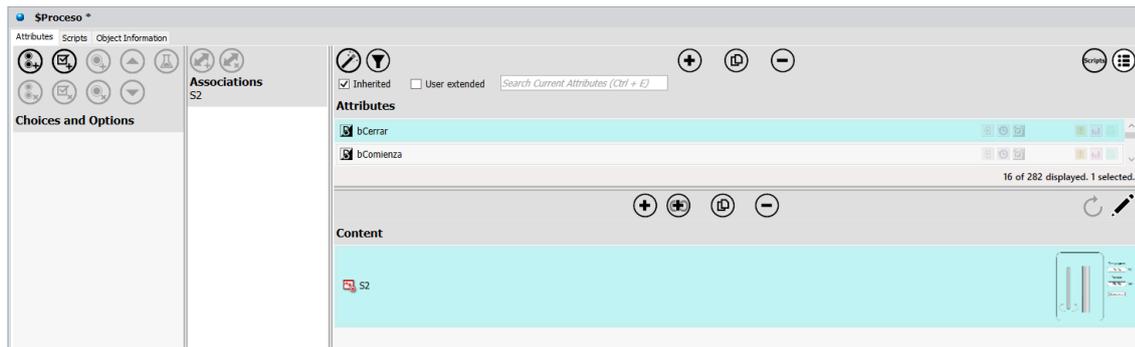
## 4.8.- CREACION DE LOS LAYOUT Y SUS SÍMBOLOS

Parte importante del software es la visualización, pues permitirá observar lo que sucede en tiempo de ejecución y llevar un control más exhaustivo, y para eso será necesario crear los siguientes elementos: un layout por cada vista que se quiera incorporar, en total habrá 2, uno para la factoría y otro para la lectura, y un objeto del tipo ViewApp, donde se establecerá el orden de las visualizaciones.



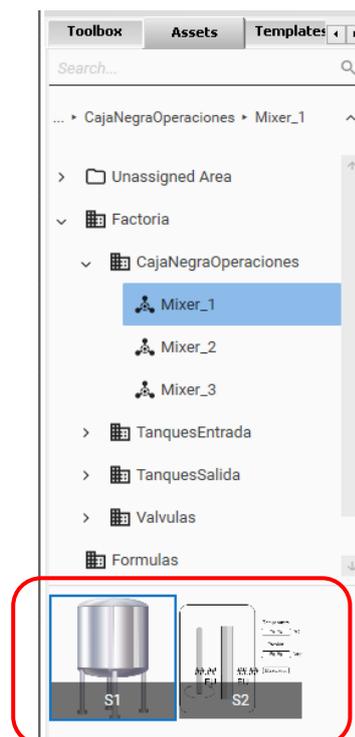
*Ilustración 4.44 .- Configuración objeto ViewApp*

En los layout se enlazarán los símbolos correspondientes a cada área y en ellos se incorporarán todos los componentes que estén en la planta y demás detalles, como podrían ser líneas de conexión de los elementos. Una ventaja de tener los símbolos de los componentes creados en las plantillas es la siguiente, si en el futuro se quieren incorporar nuevos elementos del mismo tipo, no sería necesario crearlos desde cero, pues una vez que se ha realizado la instancia, esta ya incorporará todas las configuraciones de la plantilla, así como los símbolos que esta posea.



*Ilustración 4.45 .- Plantilla Procesos*

ArchestrA ofrece la posibilidad de añadir infinidad de elementos de diferentes tipos según la biblioteca que se escoja, pero para incluir los símbolos de los objetos ya creados se hará de la siguiente manera, ver Ilustración 4.46, y seleccionando en la parte inferior el símbolo que se requiera.



*Ilustración 4.46 .- Insertar símbolo de un objeto propio*

Una vez se han añadido todos los componentes y se han realizado las conexiones entre ellos, así es como queda el símbolo correspondiente a la factoría.

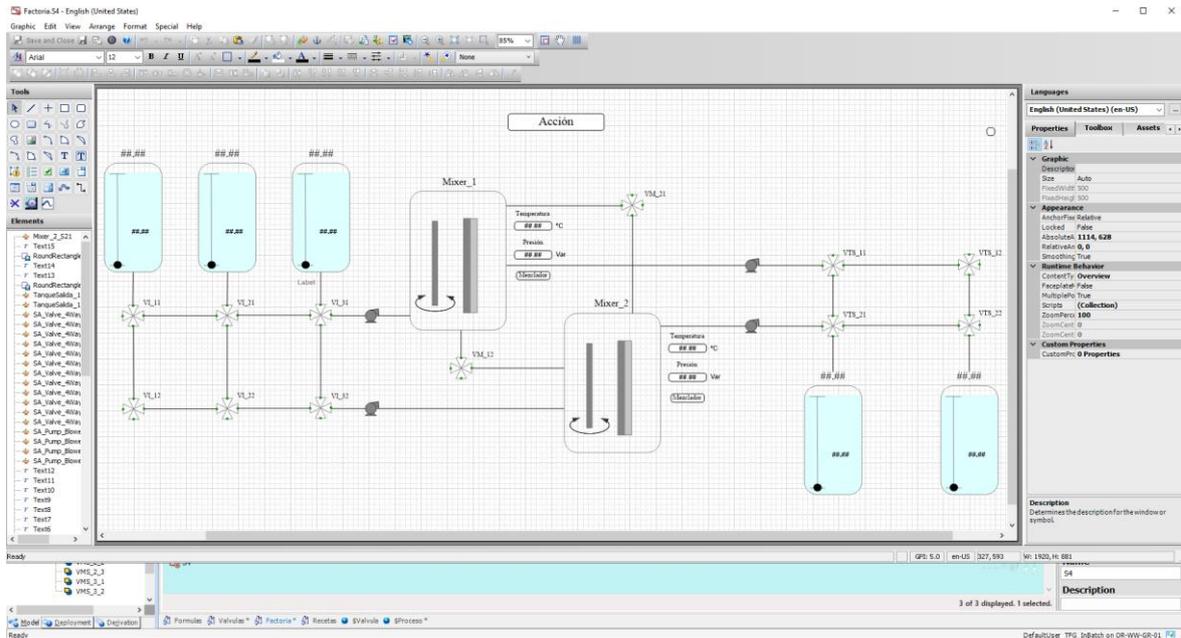


Ilustración 4.47.- Creación símbolo final

La visualización de Lectura se hizo de una forma diferente, pues lo único que se pretendía mostrar es la base de datos, por lo que, en el layout, en vez de enlazar un símbolo de Lecturas, se enlazó un objeto propio de System Platform que trabaja con SQL. Este, a través de script se enlaza con la base de datos de SQL Managements Studio ya creada y muestra todos los datos que haya en ella.

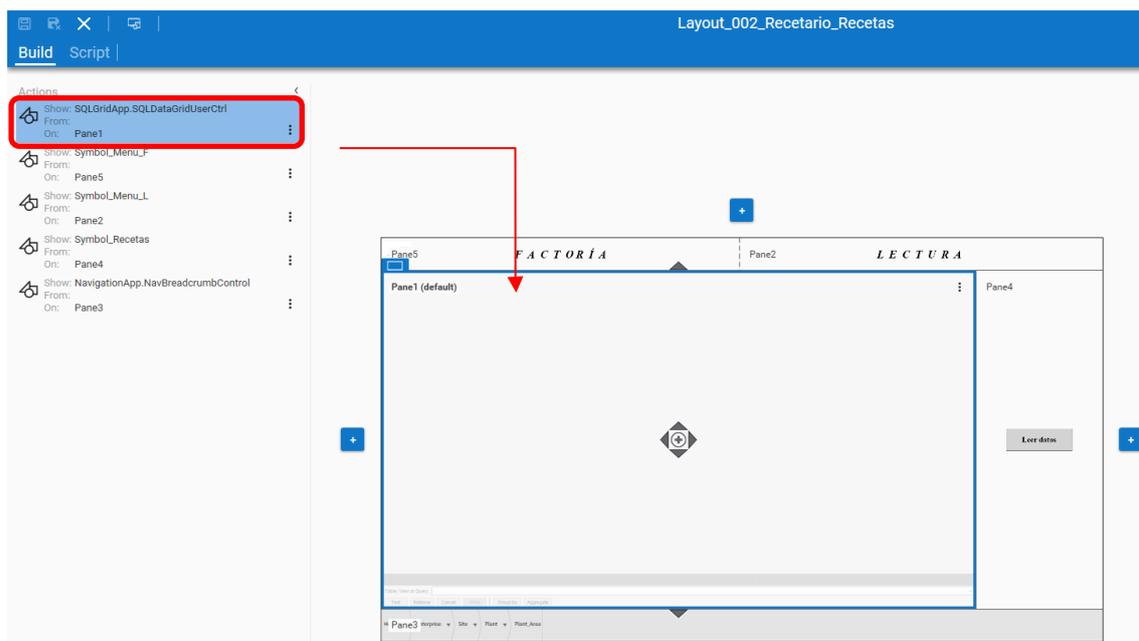


Ilustración 4.48.- Enlazar objeto de lectura para SQL

Por lo tanto, la visualización final quedaría de la siguiente forma: una vez se arrancar el visualizador IntouchOMI, se mostrará la factoría, y mediante el botón superior que pone “Lectura”, se accederá a la otra ventana, mostrando la base de datos correspondiente a la receta.

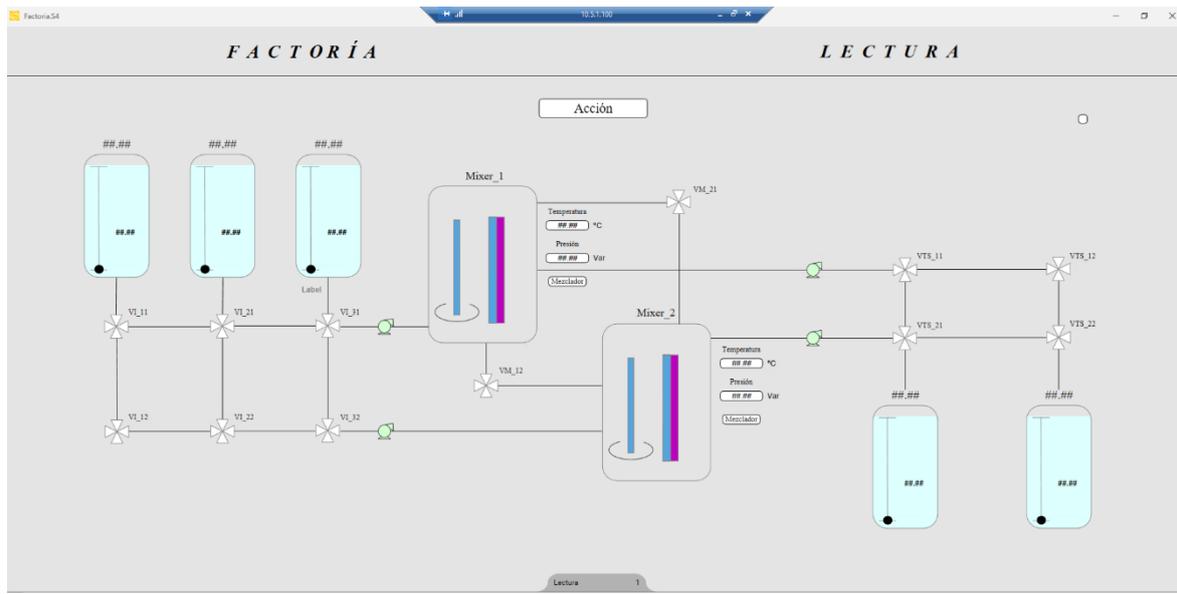


Ilustración 4.49 .- Visualización final factoría

FACTORIA										LECTURA	
id	IdReceta	Secuencia	Solapamiento	Descripcion	Componentes	Tipo	Actuador	CantidadMin	CantidadMax	Unidad	
1	R001	10	no	Movimiento	T1,T2,M1	Entrada	Agua	0	100	L	
2	R001	20	no	Operacion	M1,M2,M3	Start/Stop	Start				
3	R001	30	no	Movimiento	T1,T2,T3,M1,M2	Entrada	Leche	0	300	L	
4	R001	40	no	Operacion	M1,M2,M3	Consigna	Temperatura	0	200	°C	
5	R001	50	no	Movimiento	T1,T2,T3,M1,M2	Entrada	Sal	0	100	Kg	
6	R001	60	no	Movimiento	T1,T2,T3,M1,M2	Entrada	Leche	0	300	L	
7	R001	70	no	Operacion	M1,M2,M3	Start/Stop	Stop				
8	R001	80	no	Movimiento	M1,M2,M3,TS1...	Entrada	Vaciado	-1	-1	L	
10	R001	90	no	Operacion	M1,M2,M3	Consigna	Presion	0	300	Var	
11	R001	100	no	Movimiento	M1,M2,M3,TS1...	Entrada	Vaciado	-1	-1	L	

Ilustración 4.50 .- Visualización final lectura

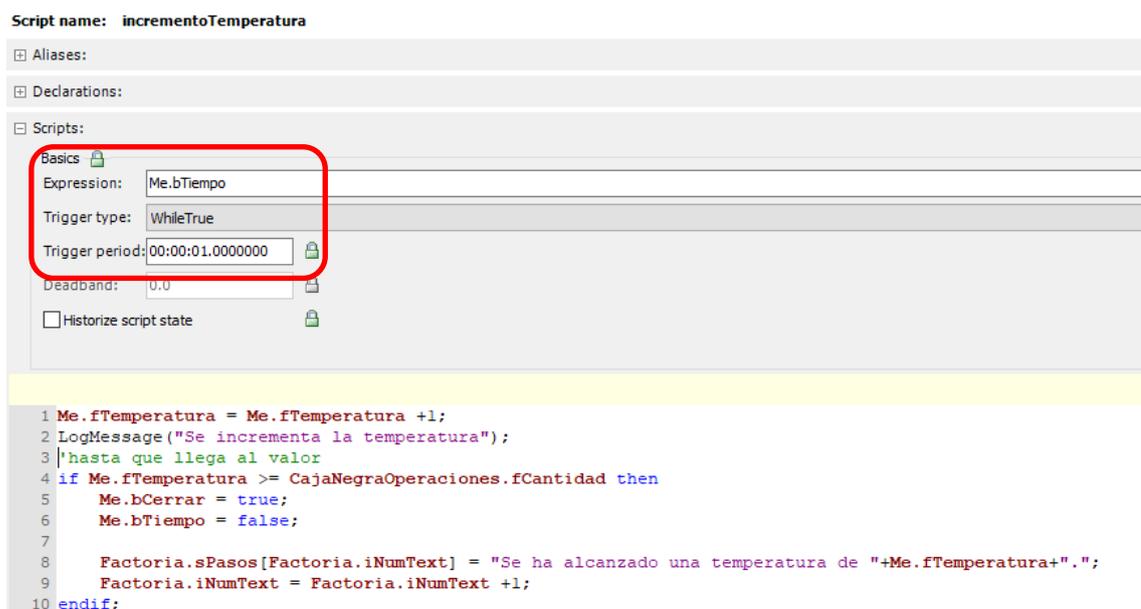
## 4.9.- ÚLTIMAS MODIFICACIONES

Las últimas modificaciones realizadas se hicieron con el fin de mejorar algunos detalles que no se habían tenido en cuenta previamente, así como indicaciones por parte de la empresa.

- Tiempo de ejecución en los componentes.

Este cambio solo se realizará para la simulación. Hasta el momento, la ejecución se realizaba siguiendo el reloj interno de la computadora, por lo que las operaciones eran instantáneas y para visualizar la ejecución era un tanto incómodo, pues era demasiado rápido. Por ejemplo, cuando se realizaba una operación de consigna de temperatura, el proceso de hasta alcanzar una cierta temperatura era automática, por lo que no daba tiempo a apreciarlo.

Ahora, mediante las opciones de ejecución de los scripts, se consiguió que la ejecución se demorara un tiempo, para así poder visualizarlo mejor. El tiempo establecido es completamente inventado, pues en el futuro cuando se implante en una planta real, habrá diferentes sensores que controlen los datos y no será necesario incluir este tiempo.



Script name: incrementoTemperatura

Aliases:

Declarations:

Scripts:

Basics 

Expression: Me.bTiempo

Trigger type: WhileTrue

Trigger period: 00:00:01.00000000 

Deadband: 0.0 

Historize script state 

```
1 Me.fTemperatura = Me.fTemperatura +1;
2 LogMessage("Se incrementa la temperatura");
3 |'hasta que llega al valor
4 if Me.fTemperatura >= CajaNegraOperaciones.fCantidad then
5     Me.bCerrar = true;
6     Me.bTiempo = false;
7
8     Factoria.sPasos[Factoria.iNumText] = "Se ha alcanzado una temperatura de "+Me.fTemperatura+";
9     Factoria.iNumText = Factoria.iNumText +1;
10 endif;
```

Ilustración 4.51 .- Modificación Trigger period para demorar la ejecución

- Añadir log informativo.

Contrastando lo realizado en la solución software con lo que quería la empresa, sugirieron introducir un cuadro de texto en la parte inferior donde se muestren mensajes a la par que la ejecución vaya avanzando, así, además de visualizar la simulación a través de los componentes, se pueda llevar un control más detallado de todas las acciones que van sucediendo, permitiendo también comprobar todos los mensajes una vez a finalizado la ejecución.

Para ello se insertó un *TextBox* que estará presente en ambas pantallas. El *TextBox* está asociado a un *array* definido en el área Factoría, donde se irán insertando mensajes según la ejecución va realizándose. Además, siempre que la ejecución comience de nuevo o bien se cargue una nueva fórmula, este *array* se vaciará.

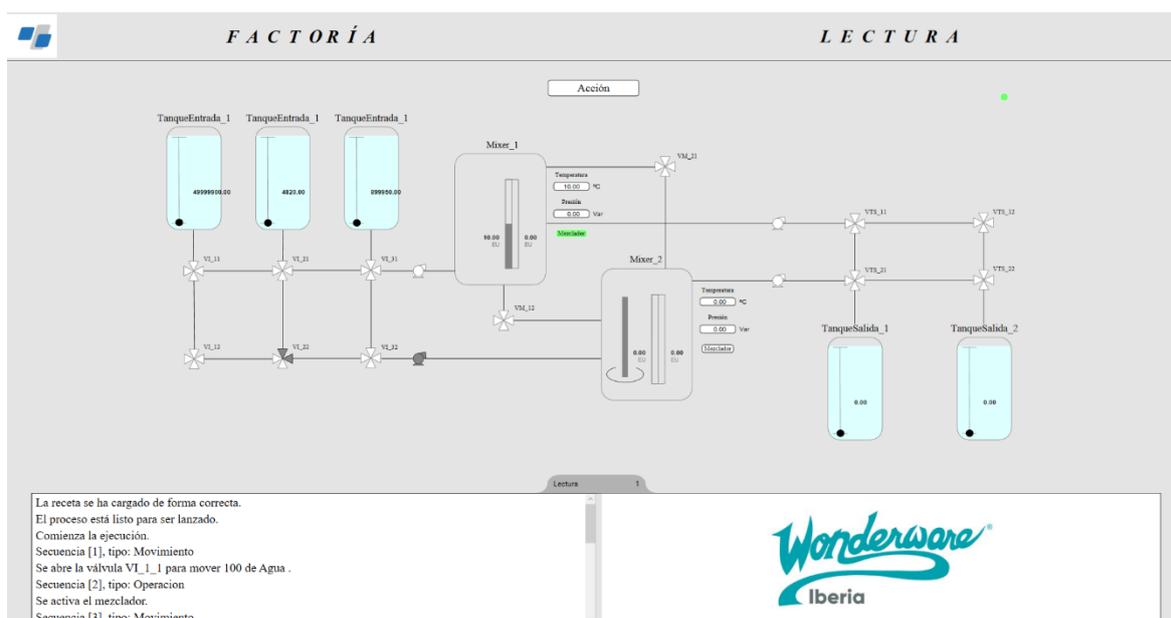


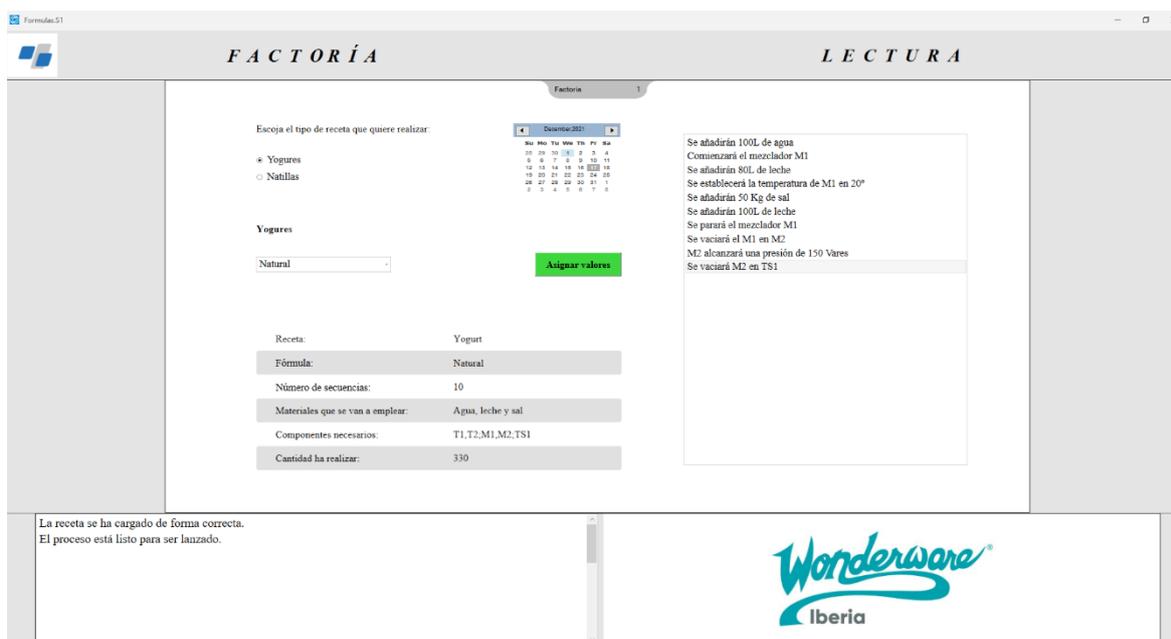
Ilustración 4.52 .- Visualización Factoría con log

- Cambio visualización lectura.

Al igual que en el caso anterior, se sugirió cambiar la visualización de la ventana de lecturas, era correcta, pero mostraba información irrelevante, además, no permitía escoger el producto a realizar desde ella, por lo que se modifica, de manera que ahora se tenga la posibilidad de seleccionar entre las recetas que estén incluidas en la base de datos y posteriormente la fórmula, en función de la receta seleccionada.

Se crearon *RadioButtons*, tantos como diferentes recetas existan y un desplegable, en cual mostrará las fórmulas asociadas a cada receta. Una vez se ha seleccionado lo que se quiere realizar, mediante el botón “Asignar valores” se cargarán los valores en el sistema desde la base de datos, además de realizarse las comprobaciones pertinentes. En caso de ser haber algún error, se mostrará a través de la pantalla cambiando el color del botón que se ha pulsado previamente e insertando en el *TextBox* informativo el problema que ha surgido.

También al pulsar sobre el botón previamente mencionado, se mostrará un resumen de la fórmula seleccionada en los cuadros de texto que se muestra en la ilustración siguiente, *Ilustración 4.53*.



*Ilustración 4.53.- Visualización Lectura final*

#### 4.10.- INCLUIR NUEVOS ELEMENTOS EN FUNCIÓN DE LA ESPECIFICACIÓN DE LA PLANTA

De cara a demostrar la escalabilidad de esta herramienta, se establecerá una nueva configuración de la planta, simulando una solicitud real de un cliente. Esta nueva planta constará de 3 tanques de entrada, 4 mixers y de nuevo otros 3 tanques de salida. Por lo que habrá que modificar el esquema actual.

Estas modificaciones se realizarán de una forma muy rápida gracias a las posibilidades de Archestra y los símbolos Wizard incluidos previamente, en concreto para los mixers y tanques de salida, pues son los tipos de componentes que se han de incorporar.

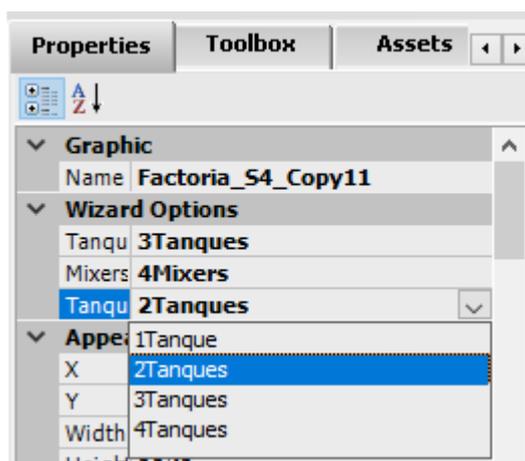


Ilustración 4.54 .- Opciones wizard para la configuración de la planta

##### 4.10.1. CREACIÓN DE NUEVOS OBJETOS

Lo primero que se hará será observar que objetos existen en la planta y cuáles será necesario crear. Como se observa en la *Ilustración 4.55*, ya se dispone de 3 tanques de entrada, 2 mixers y 2 tanques de salida, por lo que para realizar la configuración solicitada habrá que crear 2 mixers nuevos y un tanque de salida más.

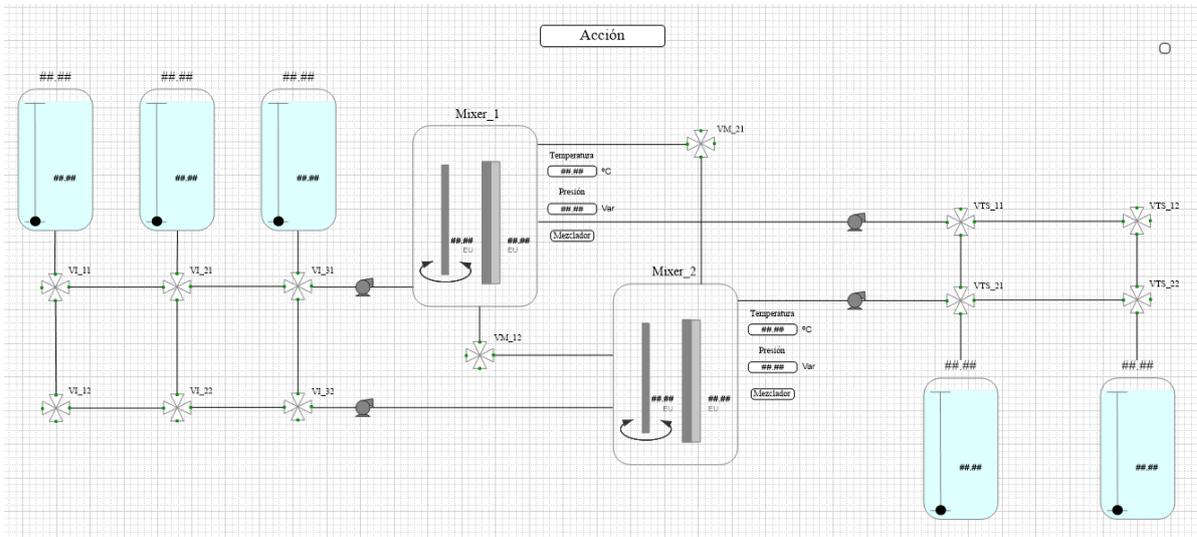


Ilustración 4.55 .- Configuración inicial de la planta

Indirectamente será necesario crear válvulas que permitan el movimiento de material entre los tanques existentes y los nuevos, para así interconectar todos los elementos. Donde antes había 12 válvulas, ahora se crearán otras 24, por lo que en total la planta dispondrá de 36 válvulas que interconectarán todos los componentes finales. Se realizará un boceto en papel de cara a comprobar cuál puede ser la mejor configuración posible.

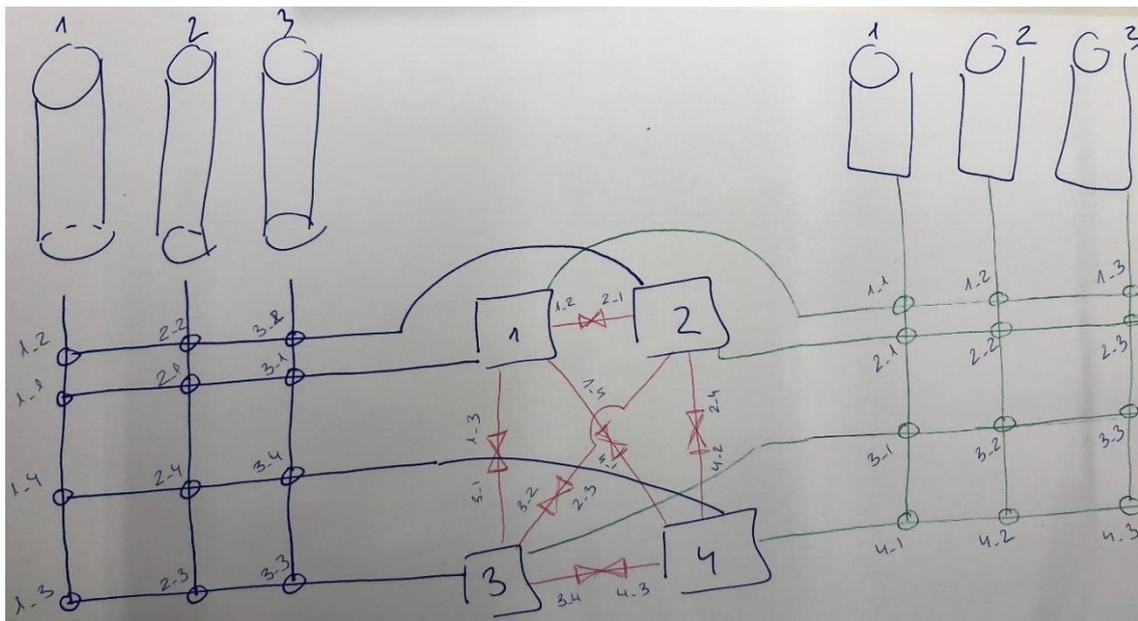


Ilustración 4.56 .- Boceto en papel de los componentes

Para crear los nuevos componentes se realizará de la misma forma que se hizo inicialmente, creando instancias de las plantillas que existen ya definidas y configuradas, pues así se ahorrará una gran cantidad de tiempo y todos los componentes serán iguales.

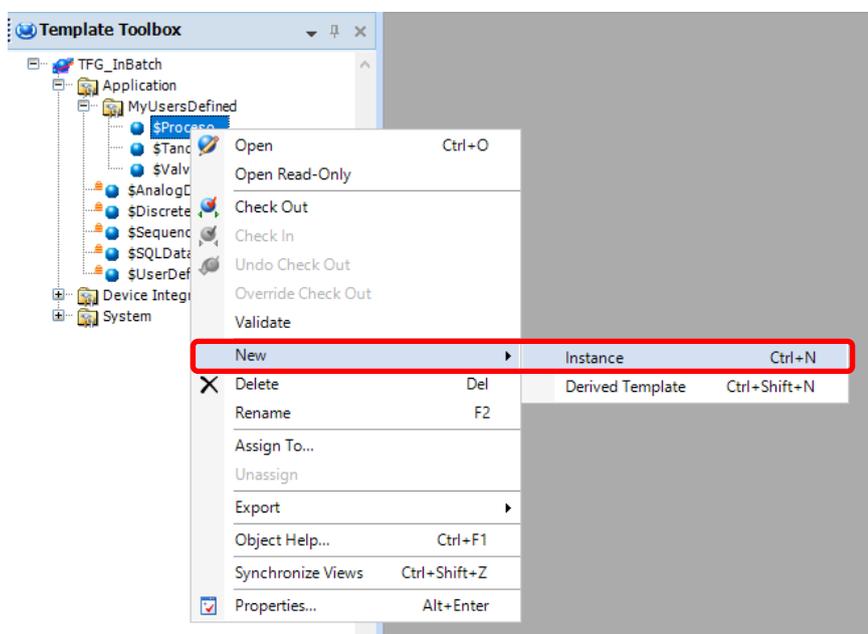


Ilustración 4.57 .- Creación de nuevas instancias

Cada instancia nueva creada habrá que moverla a su área correspondiente y se cambiará el nombre para así tener todos los elementos de la planta definidos con el mismo formato, ya que, para acceder a los objetos será necesario que el nombre siga una regla, por lo que esto será imprescindible.

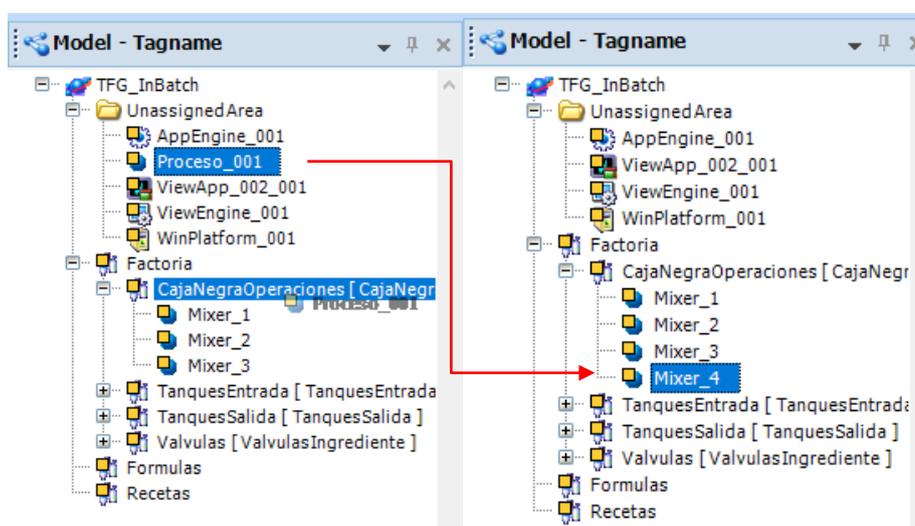
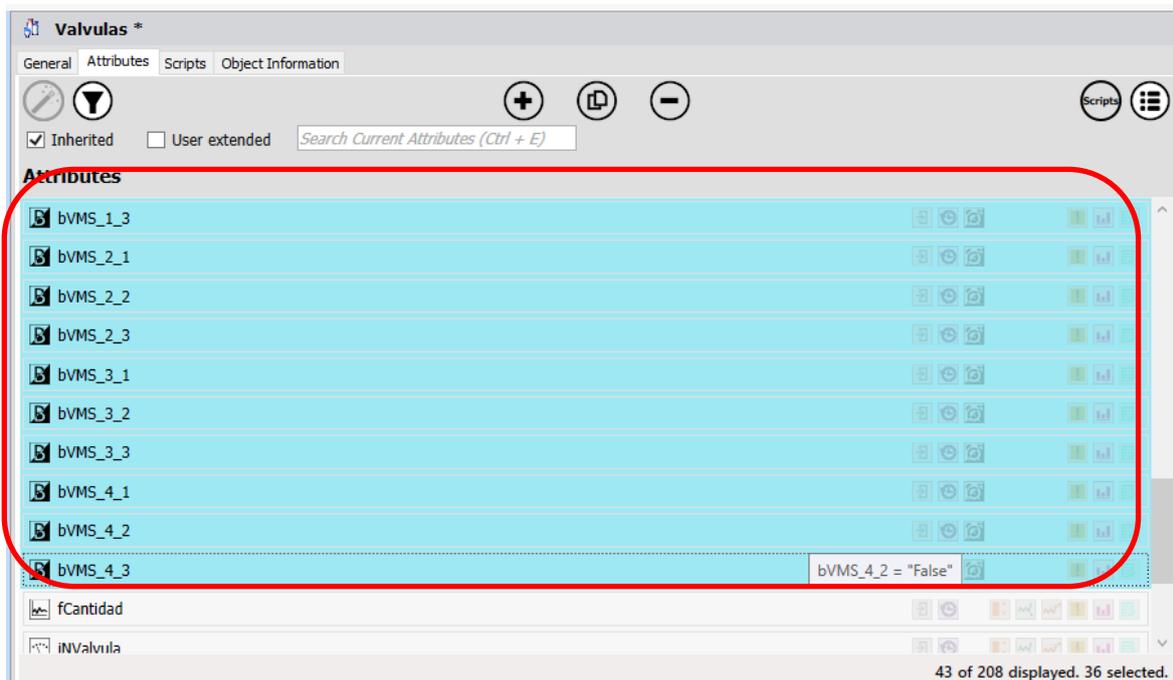


Ilustración 4.58 .- Cambio de nombre y asignación del objeto a su área

El sistema se controla mediante booleanos, por lo que para poder manejar los nuevos componentes creados habrá que añadir nuevos atributos a cada área correspondiente, concretamente en el área donde se ubican los mixers se crearán dos booleanos *bMixer\_3* y *bMixer\_4*, y en el área de las válvulas se crearán 18 nuevos atributos, pues algunos ya existían previamente.



*Ilustración 4.60 .- Nuevos atributos en el área de las válvulas*

En función a como está todo programado por detrás el sistema, no será necesario modificar nada en los scripts de los componentes nuevos o en las áreas en la cuales se incluyen estos elementos nuevos, por lo que para finalizar la nueva configuración solo habrá que modificar el símbolo correspondiente a la factoría, añadiendo ahí tanto los elementos nuevos como las conexiones necesarias para su correcta visualización.

### 4.10.2. MODIFICACIÓN DE ELEMENTOS EXISTENTES

Como se mencionó previamente, los símbolos se añadirán rápidamente mediante las opciones wizard que existen ya definidas, por lo que únicamente será necesario escoger el número exacto de elementos que se desean de cada tipo, como se muestra en la Ilustración 4.54.

Para no modificar el símbolo existente se creará un duplicado de este, pues en caso de cambiar algo de manera incorrecta no será necesario volver a crear desde cero todos los gráficos.



*Ilustración 4.61 .- Duplicar símbolo S4*

Una vez seleccionados los símbolos que se quieren incorporar, se interconectarán mediante líneas y válvulas y se desplazarán para que la planta quede visualmente atractiva a la vista y no sea un caos. Además, será necesario establecer el nuevo símbolo como la pantalla principal el layout correspondiente a la Factoría. Mediante el ContentType, se establece que símbolo irá ubicado en los huecos del layout creados, de forma que coincidan.

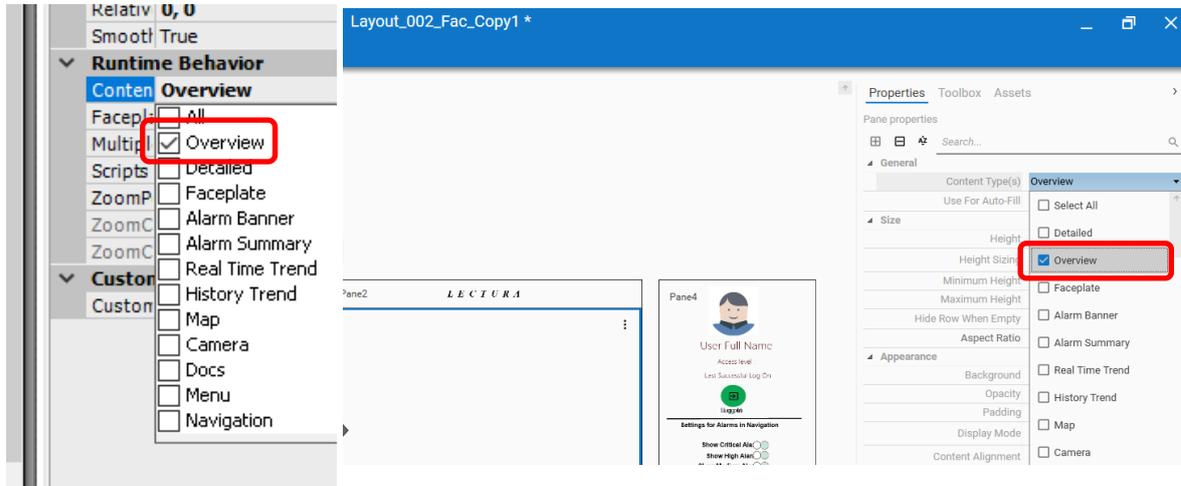


Ilustración 4.62 .- Configuración ContentType

Finalmente la configuración de los elementos de la planta quedará de la siguiente manera como se muestra en la Ilustración 4.63

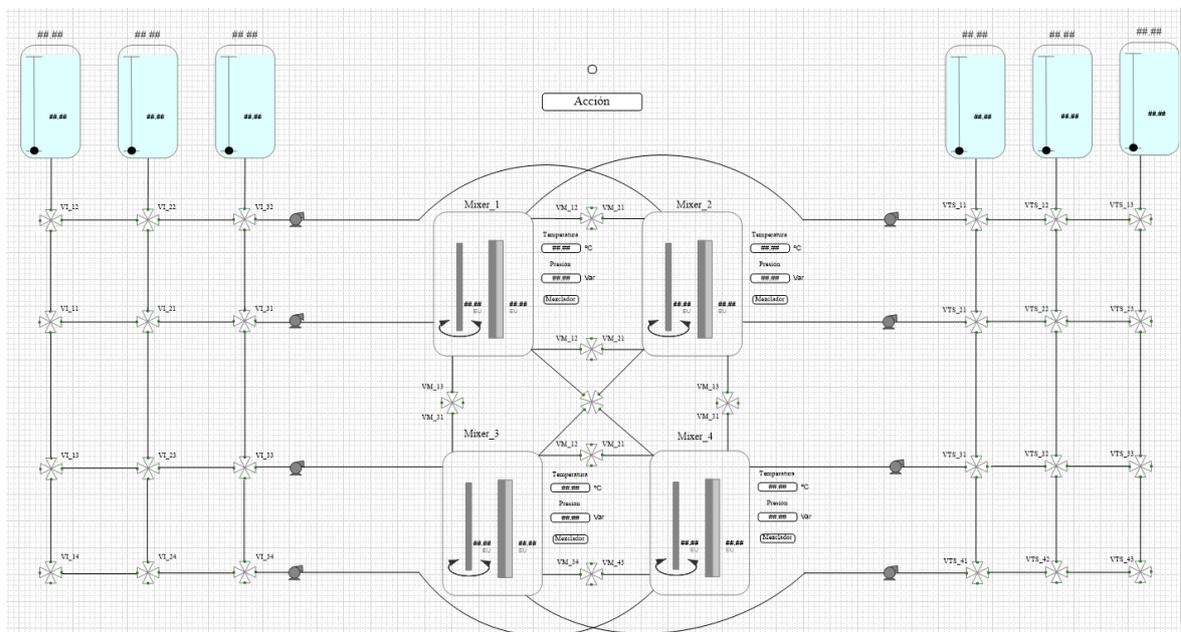


Ilustración 4.63 .- Configuración final Factoría

## **4.11.- RECETA REAL EMPLEANDO LOS NUEVOS COMPONENTES**

Se pretende incorporar una receta a modo de ejemplo para demostrar que se pueden realizar productos reales. La receta que se establecerá será la del yogurt de arándanos, en esta intervendrán 4 ingredientes, que estarán ya almacenados en los tanques de entrada.

Así mismo se realizará sobre la planta configurada en el anexo anterior, para así comprobar que se ha realizado correctamente, demostrando la escalabilidad de la herramienta, permitiendo incorporar los componentes que se requieran.

Se creará una nueva receta y una fórmula que opere sobre los nuevos componentes, así como los que ya existían, además para finalizar se mostrará cómo se va simulando la ejecución.

### **4.11.1. CREACIÓN DE LA RECETA**

La receta del yogurt está formada por 4 ingredientes, además de varias operaciones de mezclado, presión, temperatura y pasteurización, la cual habrá que incluir en el sistema. Se realizará un esquema en papel de cómo será la ejecución, para posteriormente cargar instrucciones en la base de datos SQL Server.

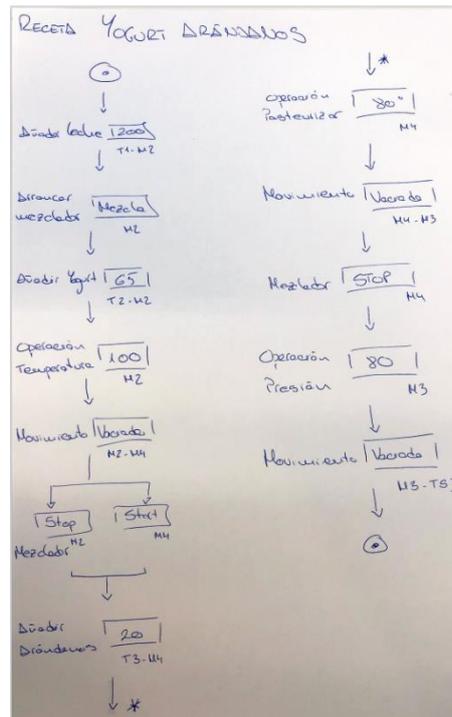


Ilustración 4.64 .- Esquema receta en papel

La nueva receta tendrá como id R003, pues convivirá en la base de datos con las anteriores ya creadas y estará formada por 13 instrucciones, entre las que existirán operaciones y movimientos de material. Se establecerá cuáles serán los componentes que van a actuar en cada instrucción, si será un movimiento de entrada o vaciado, y la operación que se realizará, pudiendo ser una consigna o un Start/Stop del mezclador. Así mismo se especificará el rango de cantidades que se podrán mover o consignas que se van a alcanzar, todo con sus respectivas unidades.

Movimiento	T1,T2,T3,T4, M1, M...	Entrada	Leche	0	300	L
Operacion	M1,M2,M3,M4	Start/Stop	Start			
Movimiento	T2,T3,M1,M2	Entrada	Yogurt	0	100	Gr
Operacion	M1,M2,M3,M4	Consigna	Temperatura	0	150	°C
Movimiento	M1,M2,M3,M4	Entrada	Vaciado	-1	-1	
Operacion	M1,M2,M3,M4	Start/Stop	Stop			
Operacion	M1,M2,M3,M4	Start/Stop	Start			
Movimiento	T1,T2,T3,M1,M2,M...	Entrada	Arandanos	0	50	Gr
Operacion	M1,M2,M3,M4	Consigna	Pasteurizador	0	30	s
Movimiento	M1,M2,M3,M4	Entrada	Vaciado	-1	-1	
Operacion	M1,M2,M3,M4	Start/Stop	Stop			
Operacion	M1,M2,M3,M4	Consigna	Presion	0	300	Var
Movimiento	M1,M2,M3,TS1,TS2...	Entrada	Vaciado	-1	-1	

*Tabla 4-4 .- Receta Yogurt de Arándanos*

#### 4.11.2. CREACIÓN DE LA FÓRMULA

La fórmula, como se explica en la memoria dictará cuales son los componentes y cantidades finales para emplear. También incluye una descripción que será el mensaje que se mostrará por pantalla en tiempo de ejecución.

La fórmula siguiente corresponde a la receta previamente creada, con id R003 y será la primera creada para esta receta, por eso se identificará como F001.

F001	R003	Se añaden 200L de leche	10	T1	M2	200
F001	R003	Comenzará el mezclador M2	20	M2		
F001	R003	Se añaden 65Kg deYogurt	30	T2	M2	65
F001	R003	Se establecerá una temperatura de 10...	40	M2		100
F001	R003	Se moverá el contenido de M2 en M4	50	M2	M4	-1
F001	R003	Parará el mezclador en M2	60	M2		
F001	R003	Comenzará el mezclador en M4	70	M4		
F001	R003	Se añaden 20Kg de Arándanos	80	T3	M4	20
F001	R003	Se pasteurizará el contenido de M4	90	M4		30
F001	R003	Se moverá el contenido de M4 en M3	100 ...	M4	M3	-1
F001	R003	Parará el mezclador en M4	110 ...	M4		
F001	R003	Se alcanzará una presión de 80 Vares	120 ...	M3		80
F001	R003	Se vaciará el contenido de M3 en TS3	130 ...	M3	TS3	-1

*Tabla 4-5 .- Fórmula Yogurt Arándanos*

### 4.11.3. CREACIÓN OPERACIÓN NUEVA

Para esta nueva receta se incorporó una nueva operación, la cual será necesario implantar en la galaxia. El proceso se realizará de la siguiente manera; desde la plantilla *Proceso*, mediante script se añadirá una nueva opción que corresponda al nombre de “Pasteurizador”.

La operación “Pasteurizador” realizar un incremento de temperatura durante un periodo de tiempo y posteriormente un enfriamiento rápido, su finalidad es destruir los gérmenes y prolongar la conservación del alimento.

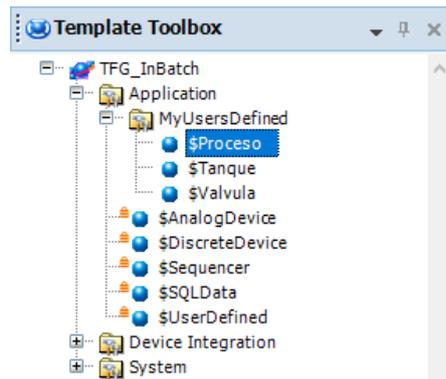


Ilustración 4.65 .- Crear nueva operación

En el script principal se añade la nueva opción correspondiente a esta operación, activará un booleano que mediante otro script enlazado incrementará la temperatura del mixer hasta los 80° y automáticamente otro script decrementará la temperatura hasta llegar a los 0°.

```
if CajaNegraOperaciones.sActuador == "Pasteurizador" then
    'Pasteurizador
    LogMessage("Pasteurizador durante: "+CajaNegraOperaciones.fCantidad+" minutos.");
    Me.bPasteurizacionInq = true;
else;
```

Ilustración 4.66 .- Script operación Pasteurizador

Para que la temperatura aumente de manera repetitiva, no será necesario crear ningún bucle, pues la opción de ejecución *WhileTrue* provoca que el script esté en constante ejecución hasta que la variable que lo controla esté a *false*.

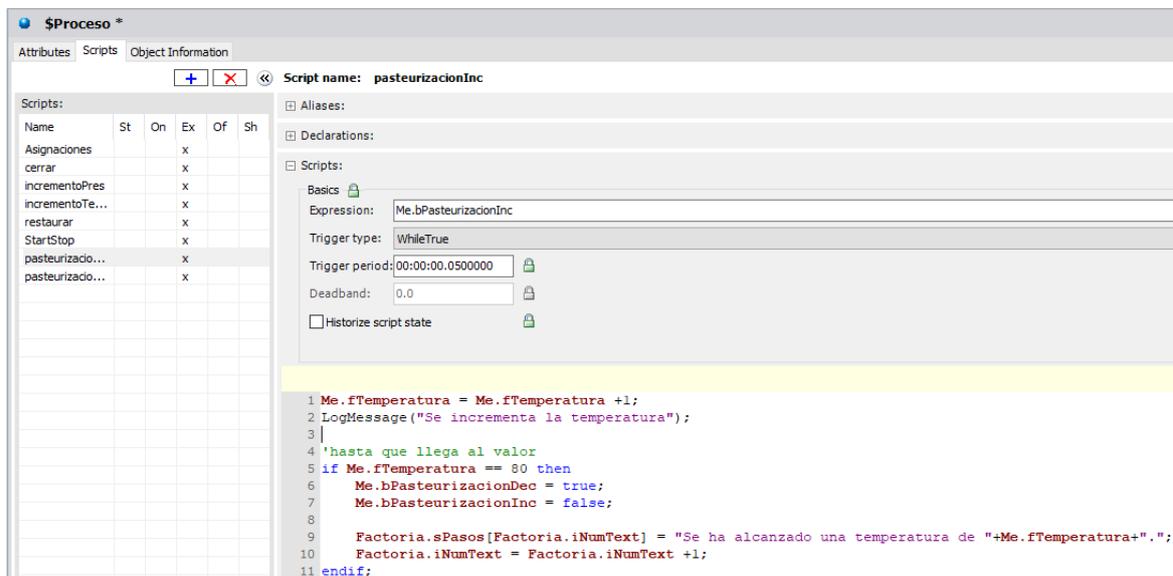


Ilustración 4.67.- Script incremento temperatura

La operación de decrementar temperatura será análoga a la anterior a excepción que en vez de aumentar la temperatura, está disminuirá con el paso de las ejecuciones.

Con estas operaciones realizadas en la plantilla no será necesario tocar nada más, pues una vez que se guarda se propagarán los cambios a las instancias que tenemos creadas de manera automática.

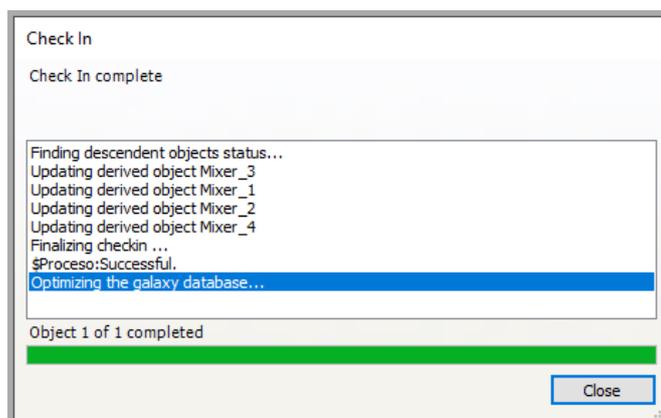


Ilustración 4.68.- Propagación de cambios a instancias

#### 4.11.4. MODIFICAR GRÁFICOS

En los gráficos correspondientes a la lectura será necesario incluir la nueva receta junto con algunos elementos, primero se añadirá una nueva opción en el *RadioButtonGroup* y seguidamente se creará una nueva *ListBox* con las fórmulas que existan con relación a la receta correspondiente.

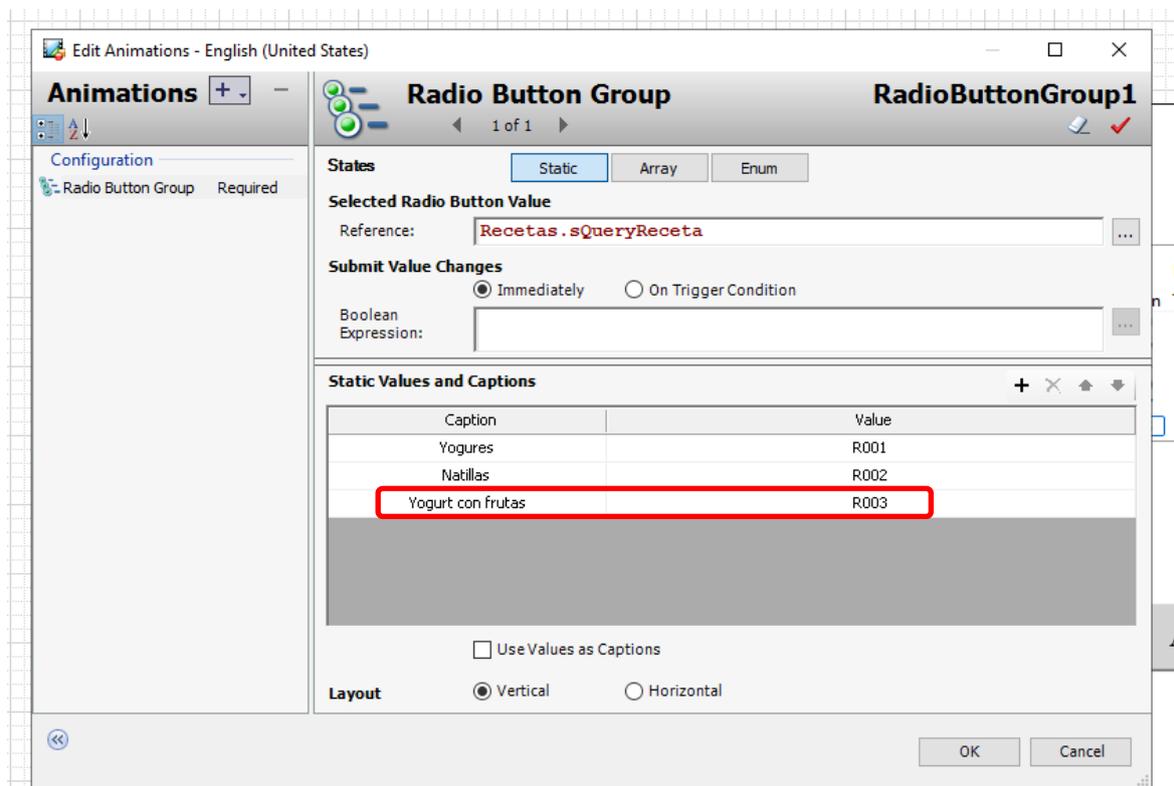


Ilustración 4.69 .- Nueva opción *RadioButtonGroup*

#### 4.11.5. PRUEBA

En este último paso se cargará tanto la receta como la fórmula en el sistema y se simulará la ejecución para ir viendo paso a paso como es el proceso que se va produciendo, detallando con presión todos los comportamientos que sucedan.

Inicialmente será necesario cargar la receta seleccionando la opción “Yogurt con frutas”, lo que provocará que el desplegable cambie y se muestren las opciones que tenemos

de esta receta. Pulsando sobre el botón *Asignar valores* se cargarán todos los valores declarados en la base de datos en la planta y de la misma manera se realizarán las comprobaciones necesarias.

Escoja el tipo de receta que quiere realizar:

Yogures  
 Natillas  
 Yogurt con frutas

December, 2021

Su	Mo	Tu	We	Th	Fr	Sa
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

Natillas

Suero

Asignar valores

Receta:

Fórmula:

Número de secuencias:

Materiales que se van a emplear:

Componentes necesarios:

Cantidad ha realizar:

Ilustración 4.70 .- Pantalla Lecturas sin datos cargados

Factoria 1

Escoja el tipo de receta que quiere realizar:

Yogures  
 Natillas  
 Yogurt con frutas

December, 2021

Su	Mo	Tu	We	Th	Fr	Sa
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

Yogurt con frutas

Arándanos

Asignar valores

Receta: Yogurt

Fórmula: Arándanos

Número de secuencias: 13

Materiales que se van a emplear: Leche, yogurt, arándanos

Componentes necesarios: T1,T2,T3, M1,M2,M3,M4; TS1,TS2,TS3

Cantidad ha realizar: 285

Se añaden 200L de leche  
Comenzará el mezclador M2  
Se añaden 65Kg deYogurt  
Se establecerá una temperatura de 100°C  
Se moverá el contenido de M2 en M4  
Parará el mezclador en M2  
Comenzará el mezclador en M4  
Se añaden 20Kg de Arándanos  
Se pasteurizará el contenido de M4  
Se moverá el contenido de M4 en M3  
Parará el mezclador en M4  
Se alcanzará una presión de 80 Vares  
Se vaciará el contenido de M3 en TS3

La receta se ha cargado de forma correcta.  
El proceso está listo para ser lanzado.

Ilustración 4.71 .- Pantalla Lecturas con dato cargados correctamente

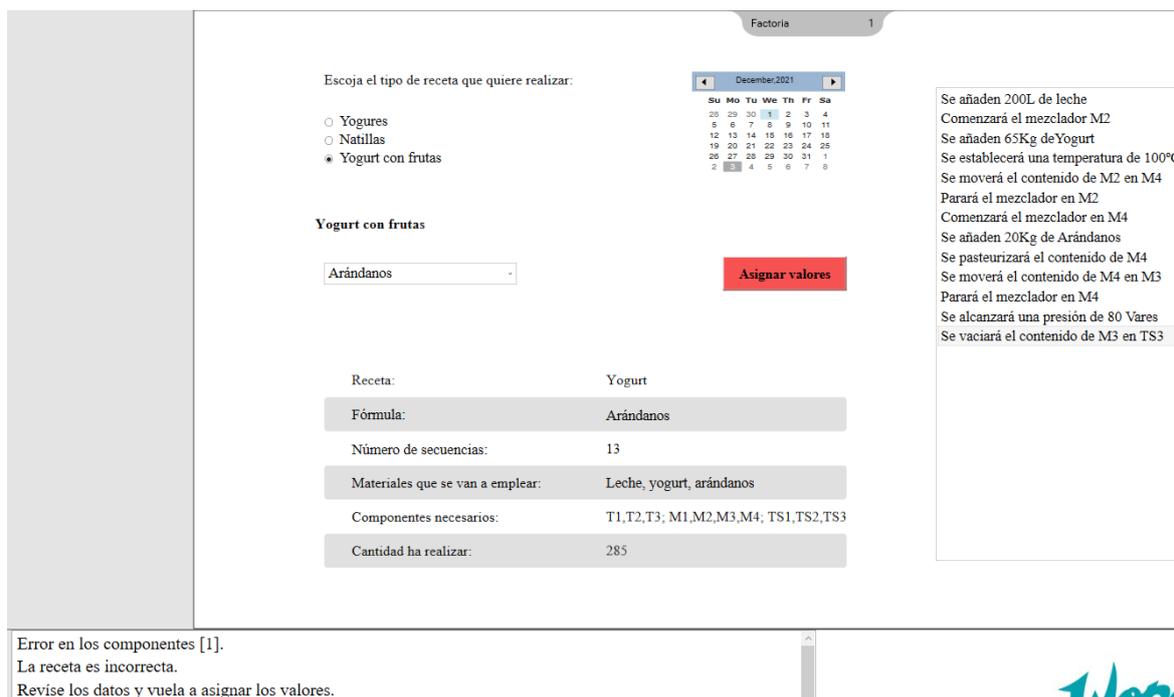


Ilustración 4.72 .- Pantalla Lecturas con datos cargados erróneamente

Una vez se han cargado los datos correctamente habrá que volver a la ventana correspondiente a la *Factoría* y comenzar la ejecución pulsando sobre el botón *Acción*.

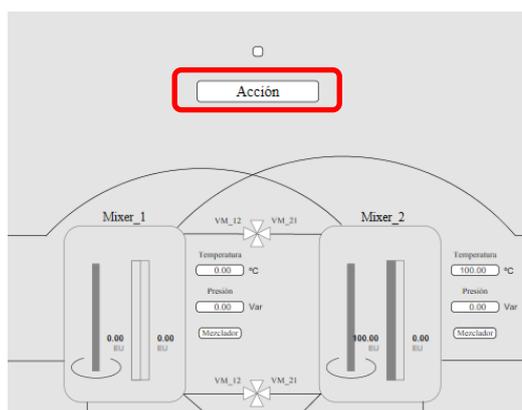
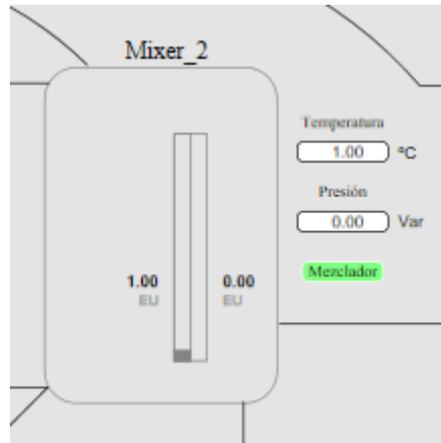


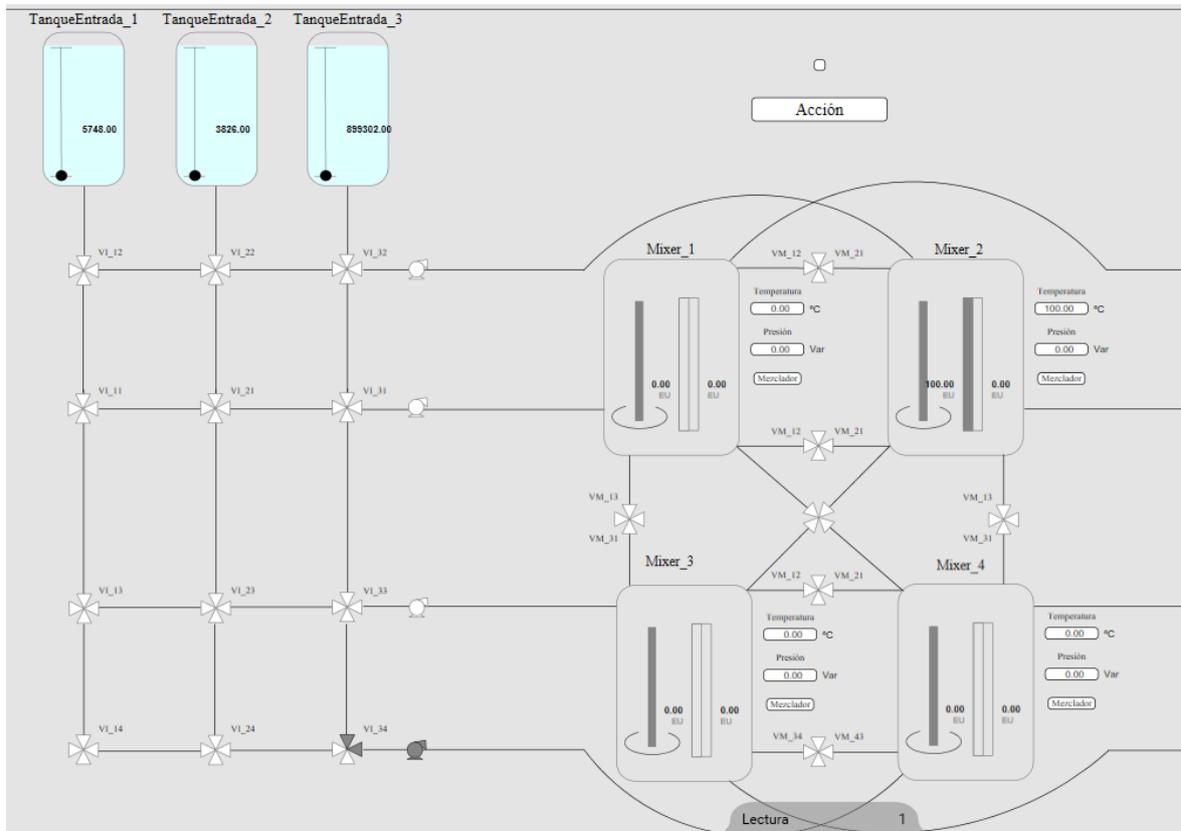
Ilustración 4.73 .- Botón para comenzar la ejecución

Se irá visualizando los cambios de color en las válvulas y en los mixers, lo que representará que se está realizando una acción entre esos componentes, por ejemplo en la *Ilustración 4.74* se muestra como el mixer tiene el mezclador activado y a su vez se está realizando una operación de aumento de temperatura, mientras que en la siguiente

ilustración, *Ilustración 4.75*, se está realizando un movimiento de material entre en tanque de entrada 3 y el mixer 4 como se puede ver en la válvula que conecta ambos componentes.

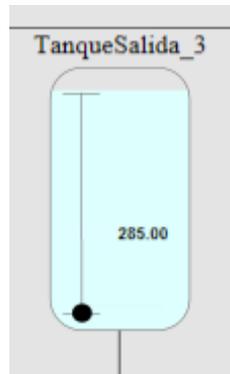


*Ilustración 4.74 .- Aumento de temperatura en el mixer 2*



*Ilustración 4.75 .- Movimiento de material entre el tanque 3 y el mixer 4*

Una vez que ha finalizado el proceso, el material resultante se almacenará en el tanque de salida 3 como se había especificado en la fórmula.



*Ilustración 4.76 .- Almacenamiento producto final*

# 5. PLANIFICACIÓN

## PRESUPUESTO

Y

Para comprender el proyecto y como será llevado a cabo todo lo necesario para su implantación, es necesario realizar una planificación de este, así se visualizará la repartición del trabajo durante los meses que dure el proyecto, se representará como un diagrama de GANTT. Además, también es necesario realizar un presupuesto para estimar los costes que supondrán la investigación e implantación de la herramienta.

### 5.1.- PLANIFICACIÓN

Para la correcta implantación de la herramienta se requiere un total de 16 semanas y media, entre las cuales se desarrollan todas las actividades necesarias. Esta es la planificación final del proyecto, y en los siguientes subapartados se describirá cada una de las principales fases que tienen lugar.

#### 5.1.1. PREPARACIÓN DEL PROYECTO

En esta fase se producen dos hitos principales como son los estudios y análisis previos, así como la familiarización con Wonderware.

Id	Nombre de tarea	Duración	Comienzo	Fin	Nombres de los recursos	tiembre 2021					octubre 2021	
						04	09	14	19	24	29	04
2	Preparación del Proyecto	15 días	lun 13/09/21	vie 01/10/21								
3	Estudios y Análisis Previos	15 días	lun 13/09/21	vie 01/10/21								
4	Familiarización con Wonderware	10 días	mar 14/09/21	lun 27/09/21								

Tabla 5-1 .- Planificación Preparación del Proyecto

### 5.1.2. INGENIERÍA

Se contemplarán cuales son los requisitos que se deben cumplir, así como estudiar los elementos involucrados y se realizará el diseño de las recetas y fórmulas.

Id	Nombre de tarea	Duración	Comienzo	Fin	Nombres de los recursos	
5	Ingeniería	20 días	lun 20/09/21	vie 15/10/21		
6	Análisis de Requisitos	10 días	lun 20/09/21	vie 01/10/21		
7	Estudio de Elementos Involucrados	15 días	jue 23/09/21	mié 13/10/21		
8	Diseño del sistema de Recetas y Formulas	10 días	lun 04/10/21	vie 15/10/21		

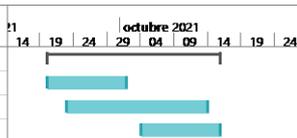


Tabla 5-2 .- Planificación Ingeniería

### 5.1.3. DESARROLLO BASE

Esta fase se solapa con la fase de Ingeniería donde se comienza con los diseños del sistema sobre como insertar las Recetas y Fórmulas. Se comenzará a crear la galaxia, sus plantillas y la BBDD definitiva.

Id	Nombre de tarea	Duración	Comienzo	Fin	Nombres de los recursos	
9	Desarrollo Base	25 días	lun 04/10/21	vie 05/11/21		
10	Definición de la Galaxia	6 días	lun 04/10/21	lun 11/10/21		
11	Creación de Plantillas	15 días	mié 06/10/21	mar 26/10/21		
12	Modelado de BBDD	5 días	lun 11/10/21	vie 15/10/21		
13	Creación de Motor de Recetas y Formulas	15 días	lun 18/10/21	vie 05/11/21		

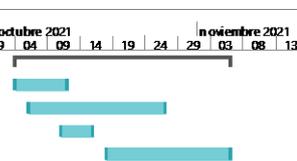


Tabla 5-3 .- Planificación Desarrollo Base

### 5.1.4. EJECUCIÓN

Es la fase que más tiempo lleva del proyecto, pues es donde se crean las variables, símbolos, recetas y sobre todo, se comienza con la configuración del software del proceso productivo y la creación de los símbolos que se van a incluir en las visualizaciones.

Id	Nombre de tarea	Duración	Comienzo	Fin	Nombres de los recursos	
14	Ejecución	40 días	lun 25/10/21	vie 17/12/21		
15	Creación de Variables	14 días	lun 25/10/21	jue 11/11/21		
16	Creación de Símbolos Gráficos	20 días	lun 08/11/21	vie 03/12/21		
17	Parametrización de Recetas y Formulas	25 días	lun 01/11/21	vie 03/12/21		
18	Implementación del Proceso Productivo	25 días	lun 08/11/21	vie 10/12/21		
19	Desarrollo de Objetos de Visualización	20 días	lun 22/11/21	vie 17/12/21		

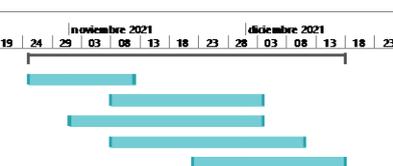


Tabla 5-4 .- Planificación Ejecución

### 5.1.5. VALIDACIÓN Y PUESTA EN MARCHA

Durante esta fase se producen dos hitos importantes:

- La aprobación de las pruebas de integración finales, que dan el paso a activar el inicio de la puesta en producción
- La validación de arranque

Id	Nombre de tarea	Duración	Comienzo	Fin	Nombres de los recursos
20	<b>Validación y Puesta en Marcha</b>	<b>11 días</b>	<b>lun 20/12/21</b>	<b>lun 03/01/22</b>	
21	Pruebas de comunicaciones	5 días	lun 20/12/21	vie 24/12/21	
22	Pruebas de integración	5 días	lun 27/12/21	vie 31/12/21	
23	Validación	1 día	lun 03/01/22	lun 03/01/22	
24	Despliegue definitivo	0 días	lun 03/01/22	lun 03/01/22	

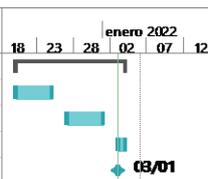


Tabla 5-5 .- Planificación Validación y Puesta en Marcha

### 5.1.6. CONTROL Y SUPERVISIÓN

La fase de control comprende la totalidad del plazo que ocupa el proyecto pues en ella se incluyen las reuniones periódicas que tienen lugar cada 15 días, así como el control por parte del jefe de proyecto.

Id	Nombre de tarea	Duración	Comienzo	Fin	Nombres de los recursos
25	<b>Control y Supervisión</b>	<b>83 días</b>	<b>lun 13/09/21</b>	<b>mié 05/01/22</b>	
26	Seguimiento del proyecto	21 días	lun 13/09/21	lun 03/01/22	
27	Control de Calidad	75 días	lun 20/09/21	vie 31/12/21	
28	Reunion de Seguimeinto 1	1 día	jue 30/09/21	jue 30/09/21	
29	Reunion de Seguimeinto 2	1 día	jue 14/10/21	jue 14/10/21	
30	Reunion de Seguimeinto 3	1 día	jue 28/10/21	jue 28/10/21	
31	Reunion de Seguimeinto 4	1 día	jue 11/11/21	jue 11/11/21	
32	Reunion de Seguimeinto 5	1 día	jue 25/11/21	jue 25/11/21	
33	Reunion de Seguimeinto 6	1 día	jue 09/12/21	jue 09/12/21	
34	Reunion de Seguimeinto 7	1 día	jue 23/12/21	jue 23/12/21	
35	Cierre del proyecto	1 día	mié 05/01/22	mié 05/01/22	

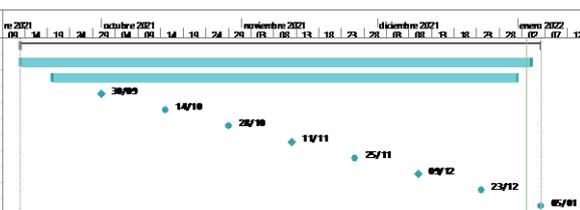


Tabla 5-6 .- Planificación Control y Supervisión

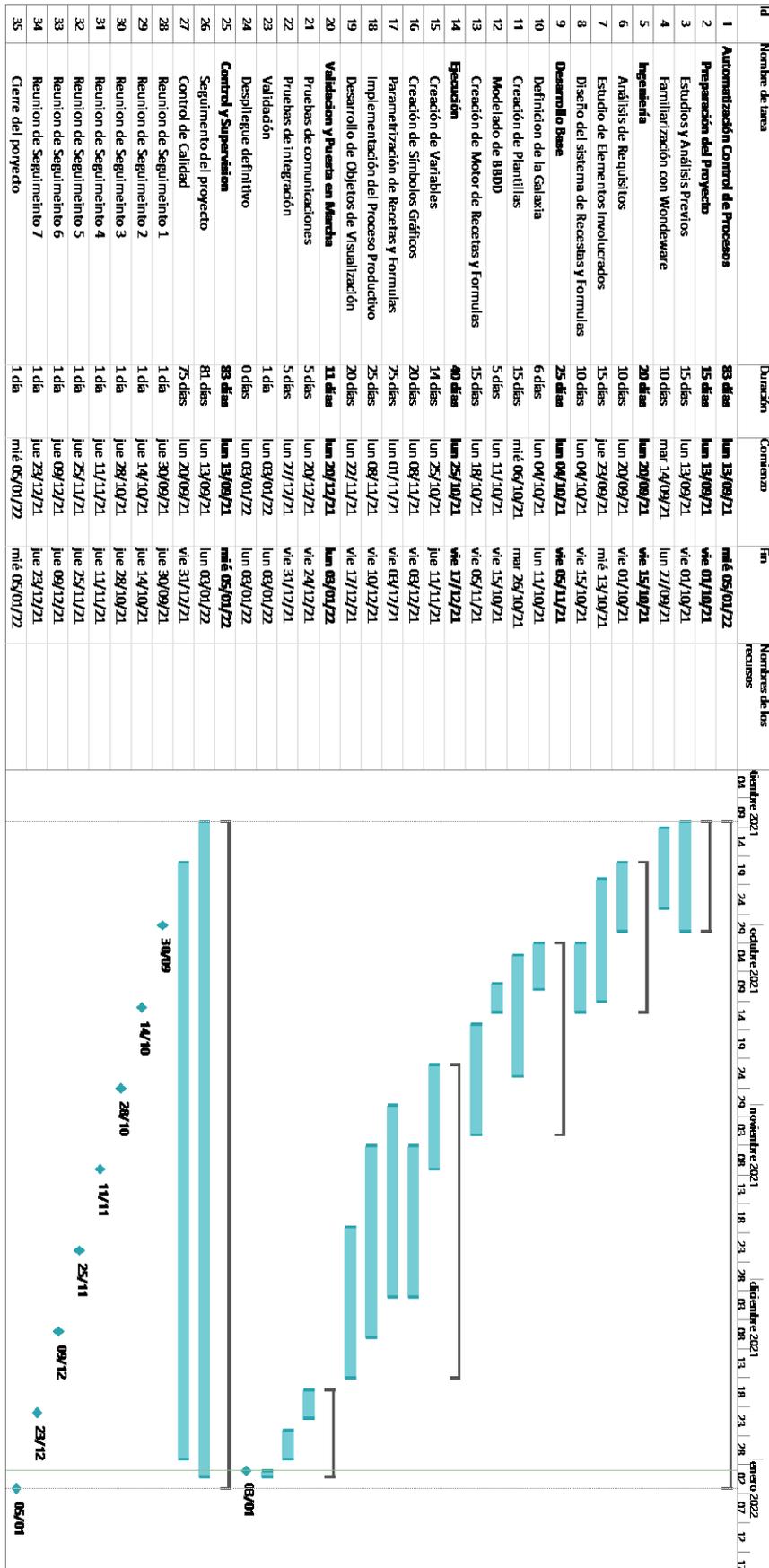


Tabla 5-7.- Planificación final

## 5.2.- PRESUPUESTO

En todos los proyectos será necesario realizar un presupuesto, pues es indispensable conocer lo que puede costar la correcta implantación de la herramienta, así como por todo lo que está compuesta. A continuación se mostrarán los cálculos realizados para el presupuesto actual.

### 5.2.1. LICENCIAS NECESARIAS

Para la implantación será necesario contar con las siguientes licencias propias de Wonderware, en concreto serán tres las que se emplearán. La primera de ellas será la necesaria para crear la galaxia, sus variables y poder usar System Platform, así como el Historian si fuera necesario, y la segunda será la necesaria para correr las visualizaciones mediante InTouch OMI y finalmente los clientes para poder correr el sistema. A todo esto hay que sumar la tarifa “Customer First” que Wonderware incorpora a todos sus productos la primera vez que se obtienen, se trata de un servicio de mantenimiento y garantía.

- AVEVA System Platform 2020, 5K IO/1K History - Application Server 5K IO, Historian Standard 1K Tag, 2 Communication Drivers Standard, 1 Historian Client Web.

Su PVP es de 28.388 €, se calcula que una planta media puede tener aproximadamente 5.000 tags, por lo que se propone emplear esta licencia, aunque en caso de ser más pequeña se podría realizar con una licencia de la mitad de las variables.

- AVEVA InTouch HMI 2020 Workstation 3K Tag with I/O

Con un PVP de 5.761 € se empleará una licencia de 3.000 tags suponiendo que no todas las variables serán de entrada/salida, por lo que esta cantidad será suficiente.

- AVEVA Supervisory Client with Historian Client Desktop 2020, no MSCAL

Su PVP es de 5.006 € y es necesario para:

- Customer first.

Para las licencias de System Platform es del 17%

Código licencia	Descripción	Precio	UD	Precio total
SysPlt-05-N-20	AVEVA System Platform 2020, 5K IO/1K History - Application Server 5K IO, Historian Standard 1K Tag , 2 Communication Drivers Standard, 1 Historian Client Web.	28.388 €	1	24.130 €
InTch-05-N-20	AVEVA InTouch HMI 2020 Workstation 3K Tag with I/O	5.761 €	1	4.897 €
SupClit-03-N-20	AVEVA Supervisory Client with Historian Client Desktop 2020, no MSCAL	5.006 €	1	4.255 €
	Customer First	0 €		4.825 €
	TOTAL			38.107 €

*Tabla 5-8 .- Presupuesto licencias*

## 5.2.2. PERFILES NECESARIOS PARA EL DESARROLLO

Perfiles	Precio por hora
Consultor de operaciones (CN)	30,00 €
Analista programador (PR)	50,00 €
Jefe de proyecto (JP)	60,00 €

*Tabla 5-9 .- Presupuesto personal*

Como en todos los proyectos será necesario contar con personal para realizar la implantación y mantenimiento, estos se clasificarán de la siguiente manera:

- **Consultor de operaciones:**

Aquella persona que realizará los estudios pertinentes de cara a aportar ese conocimiento a la correcta y ágil implantación del sistema. Organizará que tareas deben realizarse y cuál será el momento idóneo. Se especializará en Wonderware, System Platform, Intouch y SQL Server. Indicará como debe ser la creación de la galaxia, las plantillas, atributos y objetos necesarios. También decidirá cómo será el esquema que seguirán las Recetas y las Fórmulas.
- **Programador:**

Implantará con detalle todo lo que el consultor de operaciones ha diseñado y redactado, esto es, creará la galaxia y todos sus componentes de la forma que se le indica, pues se supone que será la más indicada. Así mismo realizará la programación necesaria y la creación de los símbolos gráficos y la base de datos. Es el encargado de realizar el sistema como tal.
- **Director de proyecto.**

Será el responsable de establecer y negociar la planificación del proyecto, organizar y participar en las reuniones previstas en el plan, realizar tareas de gestión de la documentación, definir y gestionar los recursos humanos del proyecto, asegurar la ejecución de las tareas del proyecto definidas en los planes, asegurar el cumplimiento del contrato y negociar las modificaciones eventuales, informar de los posibles cambios en las especificaciones técnicas, y finalmente de realizar las tareas de gestión de la documentación.

### 5.2.3. PRESUPUESTO FINAL

El presupuesto final se calculará en base a los días y el porcentaje de dedicación de la persona que realiza esa tarea, pues habrá varias de las tareas que se solapan por lo que el trabajador que la realice no invertirá el 100% de su tiempo siempre en todas ellas. Además, se calcularán los gastos generales que será un 5% de más, así como el beneficio industrial que obtendrá la empresa que será de un 15%.

Se mostrarán los cálculos en base a los operarios que realizan las acciones:

PERFIL	TAREAS	DÍAS	DEDICACIÓN	TOTAL COSTE	GASTOS GENERALES	BENEFICIO
CN	Estudioy Análisis Previos	15	40%	3.600,00 €	3.780,00 €	4.347,00 €
CN	Análisis y Estudio de Elementos Involucrados	10	60%	1.600,00 €	1.680,00 €	1.932,00 €
CN	Diseño del sistema de Recetas y Fórmulas	10	35%	1.600,00 €	1.680,00 €	1.932,00 €
CN	Creación de Plantillas	15	20%	1.200,00 €	1.260,00 €	1.449,00 €
CN	Modelado BBDD	20	30%	2.400,00 €	2.520,00 €	2.898,00 €
CN	Creación de Variables	15	20%	1.200,00 €	1.260,00 €	1.449,00 €
CN	Creación de Símbolos Gráficos	20	40%	3.200,00 €	3.360,00 €	3.864,00 €
CN	Desarrollo de Objetos de Visualización	20	10%	800,00 €	840,00 €	966,00 €
CN	Pruebas y validación	5	50%	1.000,00 €	1.050,00 €	1.207,50 €

*Tabla 5-10 .- Presupuesto consultor de Operaciones*

En la *Tabla 5-10* se muestran las tareas realizadas por el consultor de operaciones que se encarga principalmente de los análisis y estudios, así como de decidir qué atributos serán necesarios, qué símbolos serán los más correctos, dónde ubicar los símbolos, etc. Pero no realizará la implantación de estos. También tomará parte en la fase final donde se realizan las pruebas finales, validación y puesta en marcha junto con el programador.

El programador se encargará de realizar todas las operaciones e implantará el software necesario para desarrollar la herramienta basándose en lo que el consultor de operaciones le ordene, pues este es el que ha realizado los estudios y sabe cómo debe ser cada tarea, es por es por lo que en cada tarea ambos invierten parte de su tiempo.

Por otro lado también se incluye al jefe de proyecto en la siguiente tabla, este dedica todos los días que dura el proyecto, pero con una implicación del 20% ya que su función

principal es la de comprobar que las tareas se van realizando de la forma correcta en las reuniones periódicas y tomar las decisiones más adecuadas en cada situación.

PERFIL	TAREAS	DÍAS	DEDICACIÓN	TOTAL COSTE	GASTOS GENERALES	BENEFICIO
PR	Definición de la Galaxia	6	100%	1.440,00 €	1.512,00 €	1.738,80 €
PR	Creación de Plantillas	15	80%	2.880,00 €	3.024,00 €	3.477,60 €
PR	Modelado BBDD	5	70%	840,00 €	882,00 €	1.014,30 €
PR	Creación de Motor Recetas y Fórmulas	15	100%	3.600,00 €	3.780,00 €	4.347,00 €
PR	Creación de Variables	14	70%	2.352,00 €	2.469,60 €	2.840,04 €
PR	Creación de Símbolos Gráficos	20	60%	2.880,00 €	3.024,00 €	3.477,60 €
PR	Parametrización de Recetas y Fórmulas	25	100%	6.000,00 €	6.300,00 €	7.245,00 €
PR	Implementación del Proceso Productivo	25	100%	6.000,00 €	6.300,00 €	7.245,00 €
PR	Desarrollo de Objetos de Visualización	20	60%	2.880,00 €	3.024,00 €	3.477,60 €
PR	Pruebas y validación	5	50%	600,00 €	630,00 €	724,50 €
JP		81	20%	7.776,00 €	8.164,80 €	9.389,52 €

*Tabla 5-11 .- Presupuesto Programador y Jefe de proyecto*

Por lo tanto, realizando la suma de todas las tareas más las licencias, el presupuesto total es de ciento veinticuatro mil doscientos setenta y cuatro con noventa y un euros.

	TOTAL COSTE	GASTOS GENERALES	BENEFICIO
Tareas	54.448,00 €	57.170,40 €	65.745,96 €
Licencias	38.939,65 €	40.886,63 €	47.019,62 €
<b>Total proyecto</b>	<b>93.387,65 €</b>	<b>98.057,03 €</b>	<b>112.765,58 €</b>
		4.669,38 €	14.708,55 €

*Tabla 5-12 .- Cálculos finales presupuesto*

El beneficio industrial se calcula sobre el 15% del coste total del proyecto, por lo que la empresa generaría un beneficio industrial de 14.712,33 €.

## 5.2.4. PRESUPUESTO WONDERWARE INBATCH

Se analizará cual sería el coste de contratar la herramienta de Wonderware InBatch con todas sus licencias y mantenimientos.

Será necesario contratar las siguientes licencias:

- Mismas licencias que para la herramienta previamente creada, es decir, System Platform, InTouch y client.

**PVP: 38.107 €**

Código licencia	Descripción	Precio	UD	Precio total
SysPlt-05-N-20	AVEVA System Platform 2020, 5K IO/1K History - Application Server 5K IO, Historian Standard 1K Tag , 2 Communication Drivers Standard, 1 Historian Client Web.	28.388 €	1	24.130 €
InTch-05-N-20	AVEVA InTouch HMI 2020 Workstation 3K Tag with I/O	5.761 €	1	4.897 €
SupClt-03-N-20	AVEVA Supervisory Client with Historian Client Desktop 2020, no MSCAL	5.006 €	1	4.255 €
	Customer First	0 €		4.825 €
	<b>TOTAL</b>			<b>38.107 €</b>

*Tabla 5-13 .- Coste licencias básicas*

Además, Wonderware posee varias licencias propias de la solución InBatch como son:

- AVEVA Batch Management Server 2020, Medium (16-40 units)

**PVP: 43.214 €**

- AVEVA Batch Management Redundant Server 2020, Medium

**PVP: 21.620 €**

- AVEVA Batch Management Development Client 2020

**PVP: 6.505 €**

- Mom client MOM Client 2020, 5 Client Connections

**PVP: 7.053 €**

- Customer First, para las licencias del InBatch es del 22,5%

El problema que tienen las licencias de InBatch es que son únicas, esto es, si se pretende controlar el sistema desde varios equipos será necesario contratar dos licencias tipo *Batch-12-N-20*, para lo clientes pasará lo mismo, si van a existir más de 5 habrá que contratar 2. Por lo que, en caso de querer incorporar este software en una gran planta, o en más de una, con numerosos clientes, el precio se multiplicaría drásticamente y es donde se notaría la diferencia de precio, pues si sólo se quisiera controlar una pequeña planta controlada desde un equipo, con un único sistema de visualización, el precio, aunque sigue siendo superior, no es tanta la diferencia.

Código licencia	Descripción	Precio	UD	Precio total
Batch-02-N-20	AVEVA Batch Management Server 2020, Medium (16-40 units)	43.214 €	1	36.732 €
Batch-12-N-20	AVEVA Batch Management Redundant Server 2020, Medium	21.620 €	2	36.754 €
MOMcli-72-N-20	MOM Client 2020, 5 Client Connections	7.053 €	2	11.990 €
	AVEVA Batch Management Development Client 2020	6.505 €	2	11.059 €
	Customer First	0 €		19.232 €
	TOTAL			115.767 €

*Tabla 5-14 2 .- Coste licencias InBatch*

Por lo tanto, sumando todos los datos y suponiendo que se van a emplear dos licencias de cada tipo, a excepción de la primera, se tiene un coste total de: **154.706 €**.

## 6. CONCLUSIONES

Cuando se comenzó el proyecto la idea inicial era crear una herramienta software alternativa a la de Wonderware InBatch que realizara las mismas actividades, o en su caso, similares, esto es, controlar la automatización de procesos de control en la industria, concretamente alimentaria y farmacéutica, en función de lo que el cliente solicitara, pues en numerosas ocasiones el software de Wonderware InBatch ofrece cantidad de ventajas que son irrelevantes para alguna plantas industriales y no es posible no contratarlas con el fin de disminuir su alto precio.

Para ello ha sido necesario estudiar todo lo relacionado con este software desde la parte más baja, como es la automatización industrial, haciendo hincapié en el control de procesos por lotes, lo cual es muy empleado en industrias de múltiples sectores, especialmente alimentario y farmacéutico.

Una vez se tuvieron claros los conceptos teóricos acerca del software que se iba a realizar, fue necesario conocer las herramientas mediante las cuales se iba a implementar dicha solución. Se ha investigado acerca de Wonderware, lo que es y cuáles son sus principales usos. También se han conocido en profundidad sus herramientas, especialmente System Platform y ArchestrA, en donde se ha realizado toda la configuración del sistema haciendo uso de sus múltiples ventajas, pues permiten realizar softwares complejos sin necesidad de unos conocimientos previos muy avanzados de programación y diseño. Otra herramienta muy útil fue IntouchOMI, que es el motor encargado de correr las visualizaciones creadas previamente en ArchestrA.

Además, se ha conocido como deben de crearse las visualizaciones de una forma correcta siguiendo patrones como los que propone la biblioteca Situational Awareness, la cual emplea símbolos planos y muy limpios de cara a que visualmente sea muy rápido conocer que elementos de la planta pueden estar en funcionamiento o causando problemas. Así mismo también se ha trabajado con bases de datos, en concreto con SQL Server Management para incorporar los datos a nuestra planta, de esta forma será realmente sencillo

modificar los datos introducidos y permitirá que la gran mayoría del personal conozca como hacer esta actividad sin necesidad de tocar el software más complejo.

Uno de los objetivos al crear esta herramienta era la de realizarla con un coste inferior al que ofrece Wonderware, para ello se analizó en el *punto 5.2.4* cual sería el coste de contratar la propia herramienta ya existente.

El software que se desarrolla en este proyecto tiene un coste total del **93.411,65 €** como se calculó previamente en el *punto 5.2.3* de la memoria. También será necesario destacar que, aunque la solución propuesta puede ser implantada en una industria real, no cumple con los mismos requisitos ni tiene las ventajas que posee la herramienta propia de Wonderware. Lo que se ha realizado es una primera versión de simulación donde ya se conocía que no iba a cumplir con las mismas características, pero si bastante similares y con la capacidad de mejorarse en versiones posteriores, pues la idea por parte de la empresa es llegar a comercializar dicha herramienta como una gran solución.

Aun así, se ha cumplido con varios de los requisitos, especialmente con la rebaja de factor económico, lo cual para la gran mayoría de las empresas es algo crucial.

Otro objetivo que se ha cumplido es la realización de un prototipo real a modo de simulación, creando una planta con una configuración concreta e introduciendo una receta y una fórmula real. De esta forma se demuestra la utilidad del software que se ha creado, así como su escalabilidad.

Por último, hay que destacar la experiencia de formar parte de un equipo de trabajo en una empresa como TSK, donde se ha conocido como se trabaja en una organización que opera a nivel global atendiendo las necesidades de los clientes y cumpliendo los plazos establecidos y la presión que ello conlleva.

Además, aprender a manejar estas herramientas ha sido de gran utilidad, pues son muy empleadas en el sector industrial y las opciones de trabajar con ellas en el futuro son muy altas.

# 7. REFERENCIAS

## 7.1.- REFERENCIAS DE TEXTO

- [1] Beneficios automatización industrial: ¿Qué es la Automatización Industrial y para qué sirve? (edsrobotics.com) 25 abril, 2020 – *EDS ROBOTICS*
- [2] Información industria 4.0: ¿Qué es la Industria 4.0? | Deloitte España - Industria 4.0 la cuarta revolución industrial inteligente (cic.es) 16 Ene 2017 - *Ramón López Trueba Director General CIC.*
- [3] Definición SCADA Wonderware: ¿Qué es SCADA? - *Wonderware Iberia*
- [4] Componentes SCADA: Qué es un sistema SCADA, para qué sirve y cómo funciona | Aula21 (cursosaula21.com) - *SICMA21*
- [5] Que es batching: Procesamiento por lotes - Pragma Engineering | Soluciones en automatización y control (pragma-a.com) 2019 - *PRAGMA ENGINEERING*
- [6] Wonderware InBatch - *Wonderware Iberia*
- [7] Wonderware: Toda una historia potenciando las operaciones - *Wonderware Iberia*
- [8] Historia System Platform: AVEVA System Platform | AVEVA *Wonderware Iberia*  
Wonderware System Platform | Highly scalable SCADA Platform *Wonderware Iberia*
- [9] Definición ArchestrA: redindustria: ¿Qué es ArchestrA? 2008 - *Fernando Sevillano*
- [10] Usos ArchestrA: ArchestrA IDE | PDF (scribd.com)
- [11] InTouch OMI: Wonderware System Platform – Novedades - *Wonderware Iberia*
- [12] Principales usos bases de datos y principales tareas administrativas servidores: <https://softtrader.es/blog-microsoft/que-es-sql-server-management-studio/> - 2021  
*Antio, Director Gerente SoftTrader*
- [13] Wonderware Situational Awareness - *Wonderware Iberia*

## 7.2.- LICENCIAS A IMÁGENES

Ilustración 2.1	Imagen obtenida de: Automatización Industrial   ATRIA Innovation
Ilustración 2.2	Imagen sin copyrighth obtenida de pexels
Ilustración 2.4	Imagen obtenida de: Sistemas SCADA en el punto de mira de los Ciberataques - infoPLC
Ilustración 2.5	Componentes SCADA, imagen de Qué es un sistema SCADA, para qué sirve y cómo funciona   Aula21 (cursosaula21.com)
Ilustración 2.7	Imagen obtenida de: Wonderware InBatch
Ilustración 3.1	Logo Wonderware: LOGO WONDERWARE - ASSI
Ilustración 3.2	Logo System Platform: Software Autosave para Wonderware System Platform - Logitek
Ilustración 3.8	Imagen obtenida de: Microsoft SQL Server - Wikipedia, la enciclopedia libre
Ilustración 3.11	Tecnología SCADA: Supervisión Control y Adquisición de Datos - Vester training
Ilustración 3.12	Situational Awareness Actionable Alarming Revisited - YouTube