UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO

CARRERA DE INGENIERÍA ELECTRÓNICA Y AUTOMATIZACIÓN

TEMA:

Desarrollo de un Sistema de Control y Supervisión para la Gestión de Procesos de Llenado y Envasado en la Industria Alimentaria utilizando Software Ignition SCADA por OPC UA

AUTOR:

Carriel Sarmiento, Luis Fernando

Trabajo de Integración Curricular previo a la obtención de título de INGENIERO EN ELECTRÓNICA Y AUTOMATIZACIÓN

TUTOR:

Ing. Heras Sánchez, Miguel Armando

Guayaquil, Ecuador

20 de febrero del 2025

UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA ELECTRÓNICA Y AUTOMATIZACIÓN

CERTIFICACIÓN

Certificamos que el presente Trabajo de Titulación, fue realizado en su totalidad por **Luis Fernando Carriel Sarmiento** como requerimiento para la obtención del título de Ingeniero en Electrónica y Automatización

TUTOR:

Ing. Heras Sánchez, Miguel Armando

DIRECTOR DE LA CARRERA

Ing. Celso Bayardo Bohórquez Escobar, Ph.D.

Guayaquil, 20 de febrero del 2025

UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA ELECTRÓNICA Y AUTOMATIZACIÓN

DECLARACIÓN DE RESPONSABILIDAD

Yo, **Carriel Sarmiento, Luis Fernando**

DECLARO QUE:

El Trabajo de Titulación, **Desarrollo de un Sistema de Control y Supervisión para la Gestión de Procesos de Llenado y Envasado en la Industria Alimentaria utilizando Software Ignition SCADA por OPC UA** previo a la obtención del título de **Ingeniero en Electrónica y Automatización**, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

Guayaquil, 20 de febrero del 2025

EL AUTOR

Luis Fernando Carriel Sarmiento

UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA ELECTRÓNICA Y AUTOMATIZACIÓN

AUTORIZACIÓN

Yo, **Carriel Sarmiento, Luis Fernando**

Autorizo a la Universidad Católica de Santiago de Guayaquil a la publicación en la biblioteca de la institución del Trabajo de Titulación, **Desarrollo de un Sistema de Control y Supervisión para la Gestión de Procesos de Llenado y Envasado en la Industria Alimentaria utilizando Software Ignition SCADA por OPC UA**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, 20 de febrero del 2025

EL AUTOR

Luis Fernando Carriel Sarmiento

UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO CARRERA DE INGENIERÍA ELECTRÓNICA Y AUTOMATIZACIÓN

CERTIFICADO DE COMPILATIO


CERTIFICADO DE ANÁLISIS
magister

TESIS LCARRIEL

2% Textos sospechosos

2% Similitudes
0% similitudes entre comillas
< 1% entre las fuentes mencionadas
2% Idiomas no reconocidos (ignorado)
1% Textos potencialmente generados por la IA (ignorado)

Nombre del documento: TESIS LCARRIEL.pdf
ID del documento: 876db7841ddc8e6738973f7bdffab8f6a28c7345
Tamaño del documento original: 8,12 MB
Autores: []

Depositante: Ricardo Xavier Ubilla Gonzalez
Fecha de depósito: 13/2/2025
Tipo de carga: Interface
fecha de fin de análisis: 13/2/2025

Número de palabras: 17.864
Número de caracteres: 128.419

Ubicación de las similitudes en el documento:

Fuentes principales detectadas

N°	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	TIC-JR-II-2023-Final recibido 28-01-24, 21h00.docx | TIC-JR-II-2023-Final reci... #4e8203 — El documento proviene de mi grupo — 37 fuentes similares	1%		Palabras idénticas: 1% (208 palabras)
2	localhost | Evaluación a 5 años, de los efectos terapéuticos del metrotexato parenter... http://localhost:8080/xmlui/bitstream/3317/9365/3/T-UCSG-PRE-MED-634.pdf.txt — 34 fuentes similares	1%		Palabras idénticas: 1% (193 palabras)
3	repositorio.ucsg.edu.ec | Modelado e implementación de sistemas de control electr... http://repositorio.ucsg.edu.ec/bitstream/3317/1553/1/T-UCSG-PRE-TEC-IEM-259.pdf — 11 fuentes similares	< 1%		Palabras idénticas: < 1% (158 palabras)
4	repositorio.ucsg.edu.ec | Diseño y simulación de un sistema de control de riego, te... http://repositorio.ucsg.edu.ec/bitstream/3317/21682/1/T-UCSG-PRE-TEC-CIEA-11.pdf — 17 fuentes similares	< 1%		Palabras idénticas: < 1% (140 palabras)
5	www.controlsanitario.gob.ec https://www.controlsanitario.gob.ec/wp-content/uploads/downloads/2013/11/REGLAMENTO-DE-... — 1 fuente similar	< 1%		Palabras idénticas: < 1% (90 palabras)

Fuentes con similitudes fortuitas

N°	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	repositorio.ucsg.edu.ec | Gestión del proceso de selección e inducción del personal ... http://repositorio.ucsg.edu.ec/bitstream/3317/24136/1/UCSG-468-2365.pdf	< 1%		Palabras idénticas: < 1% (38 palabras)
2	repositorio.ucsg.edu.ec | Estudio de factibilidad para la creación de una empresa pr... http://repositorio.ucsg.edu.ec/bitstream/3317/2553/1/T-UCSG-PRE-ECO-ADM-92.pdf	< 1%		Palabras idénticas: < 1% (33 palabras)

Se revisó el Trabajo de Titulación, Desarrollo de un Sistema de Control y Supervisión para la Gestión de Procesos de Llenado y Envasado en la Industria Alimentaria utilizando Software Ignition SCADA por OPC UA presentado por el estudiante Luis Fernando Carriel Sarmiento, de la carrera de Ingeniería en Electrónica y automatización, donde obtuvo del programa COMPILATIO, el valor de 2% de coincidencias, considerando ser aprobada por esta dirección.

Certifica,

TUTOR:

Firmado electrónicamente por:
MIGUEL ARMANDO HERAS SANCHEZ

Ing. Heras Sánchez, Miguel Armando

DEDICATORIA

Dedico este trabajo a mis amigos y mi familia quienes pusieron su confianza en mí. Le dedico a mi madre quien no me dejo rendirme múltiples ocasiones donde mi confianza andaba por el suelo y ha sido un gran apoyo durante toda mi carrera.

Dedico este trabajo a los amigos y familiares que perdí este año que no pudieron verme convertido en un profesional.

AGRADECIMIENTO

Agradezco a dios por darme salud cuando necesite avanzar con la carrera, por darme la voluntad de seguir adelante pese a las múltiples adversidades. Agradezco a mi familia por apoyarme lo posible en el desarrollo de esta tesis.

Agradezco a los ingenieros de la facultad que proporcionaron sus experiencias y conocimientos a lo largo de mi carrera de ingeniería para formar lo que soy hoy.

Agradezco al Ingeniero Pablo Pincay, quien me ayudo a comprender mejor el software de este trabajo de titulación y me dio los apuntes necesarios para completar el diseño.

UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE EDUCACION TÉCNICA PARA EL DESARROLLO

CARRERA DE INGENIERIA EN ELECTRÓNICA Y AUTOMATIZACIÓN

TRIBUNAL DE SUSTENTACIÓN

f. _____

Ing. Celso Bayardo Bohórquez Escobar, Ph.D.

DIRECTOR DE CARRERA

f. _____

Ing. Ricardo Xavier Ubilla González, MSc.

COORDINADOR DEL ÁREA

f. _____

Ing. Daniel Bayardo Bohórquez Heras, MSc.

OPONENTE

Índice General

Índice de figuras

Índice de tablas

Resumen

El presente trabajo de titulación propone el diseño y simulación de un sistema SCADA para la gestión de llenado y envasados de productos alimenticios usando un software nuevo en la industria alimenticia del Ecuador llamado Ignition SCADA.

El objetivo del trabajo es presentar una alternativa a la automatización de estos procesos cruciales en la industria alimenticia del Ecuador mediante el diseño de un sistema SCADA en un programa tan versátil como Ignition que se podrá modificar o adaptar a las necesidades de diferentes empresas y simularlo para comprobar su eficiencia y la comunicación de los datos en tiempo real a un servidor principal que ofrece el software usando protocolo de comunicación OPC.

Este trabajo está dividido en 4 capítulos, el primer capítulo describe e introduce el trabajo de titulación en general, el segundo capítulo es el marco teórico la cual explica el concepto y funcionamiento de los procesos y herramientas que serán utilizados en el trabajo, el tercer capítulo detalla el diseño ejemplo en este software y todo lo necesario para llevar a cabo su función y su simulación, para así finalmente en el cuarto capitulo dar recomendaciones y conclusiones del trabajo de titulación presentado.

Palabras claves: Sistema SCADA, OPC UA, Ignition SCADA, Gestión de llenado y envasado de productos alimenticios, Industria alimenticia.

CAPITULO 1: DESCRIPCION GENERAL

1.1 Introducción

Hoy en día, la industria alimentaria se topa con retos considerables en cuanto a productividad, calidad y seguridad. La aplicación de tecnologías de vanguardia para la supervisión y el control de procesos se ha transformado en un elemento crucial para satisfacer las demandas en aumento del mercado y acatar las regulaciones internacionales.

El llenado y envasado de productos alimenticios son procesos de suma importancia en la industria alimenticia. Estas fácilmente impactan en la calidad, preservación y presentación las cuales determinaran si dicho producto termina siendo aceptado en el mercado y el consumidor. Sin embargo, estos procesos enfrentan diversos problemas que terminan por afectar la producción y la competitividad. Estos problemas suelen ser mal manejo de residuos, inactividad no controlada o planificada, fallos humanos, entre otros.

Por eso el propósito de este trabajo de titulación es de crear desde 0 un sistema SCADA para la gestión de los procesos de llenado y envasado de productos alimenticios usando un software nuevo en Ecuador de la plataforma Ignition SCADA con el estándar de comunicación OPC UA con el fin de innovar estos procesos que necesita la industria alimenticia y presentar Ignition como una idea a considerar para la automatización industrial.

1.2 Justificación

La industria alimentaria se enfrenta a estrictas normas de seguridad y trazabilidad junto con una creciente demanda de productos de alta calidad. Los procesos de llenado y envasado, esenciales para esta industria, requieren sistemas que garanticen eficiencia, precisión y cumplimiento de estándares internacionales.

El desarrollo de un sistema de supervisión y adquisición de datos usando Ignition concede a las empresas optimizar estos procesos implementando tecnologías avanzadas de automatización al minimizar errores, retrasos en la producción, mejorar la trazabilidad y la eficiencia en general.

Este trabajo de titulación no solo toma en cuenta las necesidades del sector industrial, sino inclusive sentar las bases para futuras implementaciones de esta tecnología en diferentes áreas que así lo requieran.

1.3 Definición del problema

Los procesos de llenado y envasado son pasos cruciales en la industria alimentaria teniendo un impacto directo en la satisfacción del cliente, la eficacia operativa y la calidad del producto. No obstante, muchas empresas enfrentan problemas que incluyen una mala integración de un sistema de control, tiempos de inactividad significativos, errores humanos y problemas con la captura y el monitoreo de datos en tiempo real.

Así mismo, conociendo como es el mercado altamente regulado y competitivo, habiendo una falta de un monitoreo centralizado dificulta la emisión de juicios y decisiones rápidas que requieren de información en caso de existir alguna falla. Esto fácilmente lleva a la elevación de costos operativos y reducción de la competitividad de la empresa.

Bajo este contexto, se hace imperativo desarrollar un sistema de control y supervisión eficaz y accesible que permita optimizar los procesos de llenado y envasado, mejore la trazabilidad, reduzca los desperdicios y asegure el cumplimiento de los estándares de calidad en la industria alimentaria.

1.4 Justificación del problema

El presente trabajo de titulación se desarrolla fundamentalmente en base a la necesidad de estos sistemas de supervisión y monitoreo en la industria alimentaria buscando generar ganancias al aumentar la producción, calidad de productos alimenticios, minimizar cualquier factor de error posible y tomar decisiones informadas en base a datos en tiempo real que puedan solucionar estos errores diligentemente.

Por ende, la idea de un diseño versátil y adaptable de un sistema SCADA que gestione estos procesos de llenado y envasado vino a la mente buscando solucionar estos problemas que enfrenta la industria alimentaria y beneficiarlas con lo posteriormente mencionado.

1.5 Objetivos

1.5.1 Objetivo general

Diseñar un sistema de control y supervisión basado en Ignition SCADA con el protocolo OPC UA para optimizar los procesos de llenado y envasado en la industria alimentaria, mejorando la eficiencia operativa, la toma de decisiones en tiempo real y demostrando la utilidad del programa moderno Ignition SCADA.

1.5.2 Objetivos específicos

- Definir los conceptos funcionales y técnicos referentes a un sistema de control y supervisión de gestión de llenado y envasado.
- Diseñar el sistema SCADA usando todo lo que ofrece Ignition a su disposición tomando él cuenta el uso del protocolo de comunicación OPC UA.
- Comprobar el sistema desarrollado mediante simulaciones evaluando su eficiencia y precisión en la gestión de estos procesos.
- Analizar los resultados obtenidas de las pruebas.

1.6 Metodología de investigación

La metodología de investigación informativa aplicada es el marco teórico y sistemático que se utiliza para resolver problemas específicos que afectan a la sociedad y a las personas basándose en el uso de conocimientos científicos y tecnológicos para encontrar soluciones prácticas. Se escogió este método por las siguientes razones:

- Se enfoca en la resolución de un problema real permitiendo abordar necesidades prácticas y generar soluciones concretas y funcionales.
- Permite tener más control del entorno real o simulado para realizar las pruebas que se necesita y garantizar que el sistema cumpla con los requisitos previo a su implementación.
- Al utilizar esta metodología que combina teoría y práctica, los resultados se pueden aplicar directamente, beneficiando a la industria alimentaria al mejorar la competitividad, reducir los costos operativos y optimizar procesos claves.

CAPITULO 2: MARCO TEORICO

2.1 Introducción a la Automatización Industrial

La automatización industrial a lo largo de la civilización humana fue tomando más fuerza y relevancia en base a la necesidad de buscar nuevas maneras de aumentar la producción de bienes para el consumo.

En la actualidad, la automatización industrial se volvió pieza fundamental en el desarrollo de empresas que ofrecen productos o servicios básicos como electricidad y agua, siendo la industria alimentaria una de las más beneficiadas por el cambio y aumento drástico de productividad que ofrecía la automatización industrial.

2.1.1 Concepto de la automatización industrial

La automatización industrial es un medio por el cual los procesos de producción utilizan el avance de la computación para controlar las variables de entrada, de proceso y de salida que gobiernan el sistema de producción. Esto implica que procesos, máquinas y materiales van a ser controlados por medio de programas cuyo objetivo es producir bajo los niveles de calidad y cantidad exigidos por el cliente o consumidor, de tal manera que se logren costos de operación competitivos. (Acuña, 1990)

Una definición más técnica dice que la automatización industrial es la tecnología basada en la aplicación de complejos sistemas mecánicos, electrónicos y computacionales a la operación y control de la producción. Esta tecnología incluye entre otras cosas maquinas, herramientas automáticas para

producir partes, sistemas automáticos de manejo de materiales, sistemas de control y retroalimentación de información de para la calidad y la producción de sistemas computarizados de control de procesos. Estos sistemas permiten tomar decisiones más acertadas con el apoyo de bases de datos construidas de acuerdo con las características del proceso de producción. (Acuña, 1990)

2.1.2 Aplicaciones de la automatización industrial

- **Aplicaciones directas** son aquellas en las que la computadora está directamente conectada a las líneas de producción, específicamente a líneas de transferencia, a los sistemas de manejo de materiales y a las maquinas, con el objetivo de dar instrucciones, ejecutar el control y efectuar la corrección de errores en forma directa mediante él envió de señales. (Acuña, 1990)
- **Aplicaciones indirectas** son aquellas en las cuales la computadora, con base en datos recolectados y en programas de uso especifico, genera bases de datos que se usan como un medio para controlar los componentes del proceso. Esta acción se lleva a cabo fuera de las líneas de producción lo que implica que la computadora no está directamente conectada a las maquinas, a las líneas de transferencia o a los sistemas de manejo de materiales. (Acuña, 1990)

2.1.3 Beneficios o ventajas de la automatización industrial

Ya definido lo que es la automatización industrial con sus aplicaciones tanto directas como indirectas, estos son los 6 principales beneficios de la automatización industrial:

1) **Productividad:** el incremento de la velocidad de producción sin necesidad de intervención humana con el uso eficiente de maquinaria logra incrementar drásticamente la producción.

2) **Calidad:** El uso de maquinarias y herramientas de primera con el control eficiente de una computadora logra que estos procesos de producción se acerquen cada vez más a los niveles de calidad fijados por el consumidor. Con la utilización de máquinas automáticas, la variabilidad es reducida significativamente.

3) **Desintegración de intervención humana en la producción:** La automatización aplicada directamente asegura una mínima intervención en la ejecución y gestión de las actividades que componen el sistema productivo. Esto supone una gran ventaja desde el punto de vista tecnológico, ya que las personas son la principal causa de variación en las líneas de producción. La mejor manera de alcanzar los niveles de producción y calidad necesarios para competir en mercados no tradicionales es reducir y, en muchos casos, eliminar la intervención humana en las operaciones de producción.

4) **Eficiente manejo de materiales:** Determinando cuantitativamente las materias primas y productos en un tiempo permanente en la línea de

producción, es fácil entender que más del 60% del tiempo total de producción es el trabajo de gestión de materiales. Utilizando sistemas de gestión de materiales computarizados y productos de mitad de producción, todo el proceso será más ágil y eficiente, lo que reducirá significativamente el tiempo de producción y el stock en el proceso.

5) Reducción del tiempo de preparación de maquina: Otra operación que consumo mucho tiempo es la preparación del equipo para la producción, especialmente si se fabrican varios productos. Usando sistemas informáticos, estas operaciones se pueden hacer fuera de la línea de producción reduciendo el tiempo de procesamiento.

(Acuña, 1990)

2.1.4 Desventajas de la automatización industrial

Aun con estas ventajas claras es importante recalcar que también conlleva desventajas que afectan a las compañías, tales como:

1) **Desempleo:** el hecho de utilizar mejores maquinas en los procesos con poca intervención del ser humano conlleva a una elevación de la tasa de desempleo, si el proceso de automatización se ejecuta en forma desorganizada y no planificada. Con la aplicación de la computadora a la programación y control de producción se origina desempleo de personal técnico.
2) **Relegación del ser humano a una maquina:** es un hecho que la sustitución de la mano de obra por máquinas de alguna manera significa un desplazamiento laboral del ser humano. Esta relegación puede ser positiva cuando se trata de actividades muy monótonas o riesgosas, pero también puede tener consecuencias funestas si el proceso de automatización no se planifica y si no se buscan nuevas una oportunidad de empleo en la que una persona no sólo se emplea para realizar un trabajo manual sino, lo que es más importante, para desarrollarse intelectualmente.
3) **Reducir el poder adquisitivo:** Tras desarrollar vastos productos de calidad, el mercado va a estar saturado de productos que, en su mayoría, van a incitar al consumismo y a una ardua lucha entre industriales del mismo ramo para abarcar mercados.

   (Acuña, 1990)

2.1.5 Automatización en la industria alimentaria

Definido lo que es la automatización industrial con sus beneficios y problemas, tomemos en cuenta lo que esto conlleva a la industria de alimentos.

Desde principios de la década de 1980, se anticipó que tres innovaciones importantes que tendrían un impacto en la industria alimentaria: la ingeniería genética, la biotecnología, la fabricación asistida por computadora y la robotización. En todos los casos se demostró que estas innovaciones contribuirán a una producción más competitiva y eficiente. Estas tecnologías se utilizan hoy en día o se propone su introducción en las distintas etapas del sistema alimentario. La siguiente figura muestra cómo opera usualmente una empresa con mano de obra humana en Ecuador.

Figura 1 Línea de procesamiento de atún por mano de obra humana

Fuente: https://www.primicias.ec/noticias/economia/industrias-alimentos-papel-plastico-crecimiento/

Ahora si tomamos datos económicos: La industria manufacturera de alimentos y bebidas es el sector más grande de Ecuador. Según el Banco Central del Ecuador, esta industria cuenta con una formación bruta de capital fijo (FBKF),

en miles de dólares, de $ 1 608 242 en el 2019 y con un Valor Agregado Bruto (VAB) del 5.21% en el 2020, con relación al Producto Interno Bruto (PIB), con cuyas cifras se posiciona a la cabeza de las demás industrias manufactureras. (BCE, 2020)

Con esto presente, vemos que a mayor demanda incrementa la necesidad de innovar las tecnologías que utilizan las empresas que se dedican a este sector. Por ende, es mayor inclusive la inversión requerida para modernizar el proceso de producción a gran escala. Las empresas hoy en día si quieren destacarse de otras que ofrecen los mismos productos necesitaran actualizar sus métodos de producción y la automatización brinda la innovación que necesitan para mantenerse a flote y garantizar un lugar en el mercado competitivo de comidas procesadas al aumentar la producción con mínimos errores humanos que a largo plazo logra reducir los costos y a su vez tener un mejor control de calidad en lo que conlleva todo el proceso. Un gran ejemplo de lo que es la automatización en esta industria se muestra en la figura 2:

Figura 2 Automatización en la industria alimentaria

Fuente: https://www.vld-eng.com/blog/automatizacion-en-industria-alimentaria/

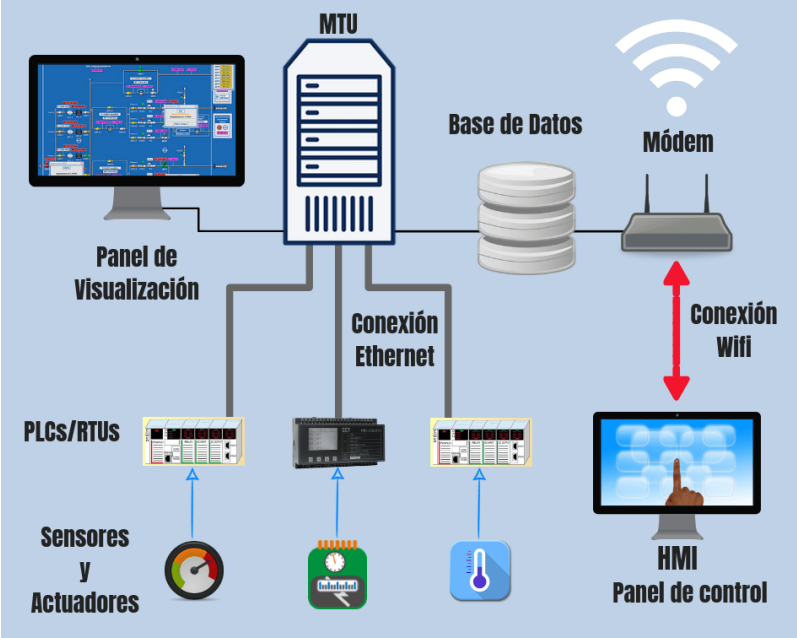
2.2 Sistemas SCADA

2.2.1 Definición de los sistemas SCADA

Los sistemas de Supervisión de Control y Adquisición de Datos (traducción más o menos aproximada de SCADA, *Supervisory Control And Data Acquisition*) permite gestionar y controlar cualquier sistema local o remoto con una interfase grafica que comunica al usuario con el sistema. (Penin, 2007)

Los sistemas SCADA son ampliamente utilizados en aplicaciones industriales y energía muy diversas, como el control de plantas en diferentes procesos, líneas de producción, centrales eléctricas, redes de transmisión o distribución eléctrica. (Laurente Raimundo, 2023)

Figura 3 Ejemplo de un diagrama básico con sistema SCADA



Fuente: https://www.cursosaula21.com/que-es-un-sistema-scada/

Aunque los sistemas SCADA puede variar en función de la aplicación, su objetivo principal es siempre el mismo: supervisar y controlar equipos y procesos remotos mediante la recopilación, transmisión y análisis de datos en tiempo real. (Laurente Raimundo, 2023)

Hay muchos tipos diferentes de sistemas SCADA variando según de su arquitectura y distribución. Los principales tipos de sistemas SCADA dependen de su distribución son las siguientes:

- **Sistemas distribuidos:** Los componentes están divididos y conectados físicamente por una red de comunicaciones.
- **Sistemas centralizados:** Los componentes se encuentran en un mismo lugar.
- **Sistemas híbridos:** Combina las características de ambos sistemas distribuidos y centralizados.

De acuerdo con su arquitectura se diferencian principalmente por ser:

- **Sistemas Clientes/Servidor:** en el que el proceso de adquisición y procesamiento es llevado a cabo por un servidor al que se vinculan los clientes. En este escenario, la interrupción del servicio de un cliente específico no impacta en el desempeño del sistema, mientras que, si el servidor se desconecta, todo el sistema se encuentra sin servicio. Para minimizar esto, se crean sistemas cliente/servidores redundantes, en los que, en caso de que uno de los servidores deje de funcionar, el otro asume el control sin causar interrupciones en el sistema.

- **Sistemas Standalone:** En esta arquitectura, todas las computadoras se encuentran vinculadas directamente a los controladores del proceso. Todos los dispositivos independientes operan en paralelo.

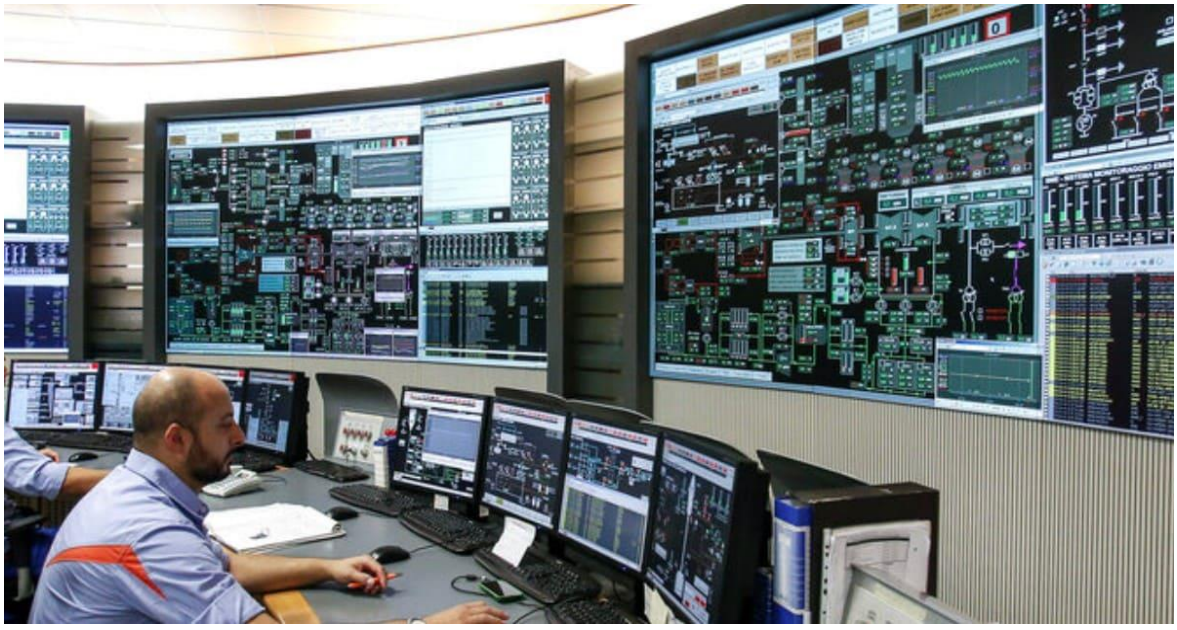
  (Laurente Raimundo, 2023)

Los sistemas SCADA se emplean en diversas industrias para supervisar y regular procesos, y en numerosas situaciones, posibilitan la supervisión de parámetros esenciales como la temperatura, la presión y las variables eléctricas. Los sistemas SCADA, al supervisar estos parámetros, ayudan a garantizar un funcionamiento seguro y eficaz de los procesos.

(Laurente Raimundo, 2023)

La siguiente figura indica como luce un sistema SCADA a escala industrial manejado por solo 2 operadores capacitados:

Figura 4 Monitoreo de un sistema SCADA avanzado



Fuente: https://mesautomation.com/que-es-un-sistema-scada-y-como-puede-ayudar-a-tu-fabrica/

2.2.2 Objetivos y funciones del sistema SCADA

La finalidad primordial de un sistema SCADA es supervisar y gestionar equipos y procesos tanto locales como remotamente. No obstante, estos sistemas también tienen la capacidad de desempeñar otras funciones y metas, como por ejemplo:

- **Supervisión en tiempo real:** El sistema SCADA tiene la capacidad de recolectar y enviar datos en tiempo real, facilitando así una vigilancia continua, un componente esencial en la supervisión de los operadores de los equipos o instalaciones.
- **Análisis de datos:** el sistema SCADA tiene la capacidad de examinar los datos obtenidos a través de la recolección de información para crear alertas o documentar sucesos.
- **Generación de alertas:** en caso de que se identifique un problema, el sistema SCADA tiene la capacidad de producir alertas para informar al operador.
- **Gestión de dispositivos:** los sistemas SCADA tienen la capacidad de gestionar directamente dispositivos y/o instalaciones a distancia, por ejemplo, mediante el activado o desactivado de motores o la apertura o cierre de interruptores.

  (Laurente Raimundo, 2023)
- **Economía:** resulta más sencillo observar lo que sucede en la instalación desde la oficina que mandar a un trabajador a llevar a cabo la labor. Algunas revisiones se tornarán inútiles.

- **Mantenimiento:** La obtención de datos materializa la oportunidad de recolectar información de un proceso, guardarla y presentarla de forma comprensible para un usuario no experto. Se puede configurar la misma aplicación para que nos informe cuando se aproximan las fechas de revisión o cuando una máquina presente más errores de los que se consideran normales.
- **Flexibilidad:** cualquier alteración en alguna de las propiedades del sistema de visualización no implica un desembolso en tiempo y recursos, ya que no existen alteraciones físicas que necesiten la instalación de un cableado o del contador.
- **Conectividad:** localiza sistemas de carácter abierto. La documentación de los protocolos de comunicación vigentes facilita la interconexión de sistemas de varios proveedores y previene la presencia de vacíos informativos que puedan provocar errores en la operación o en la seguridad.

  (Penin, Sistemas SCADA: guía práctica, 2007)

2.2.3 Componentes y estructuras principales del sistema SCADA

El sistema SCADA requiere de componentes vitales tanto de hardware como de software para que funcione de forma óptima y cumpla con sus funciones dadas, estos componentes se describen a continuación:

Hardware

- **Ordenador Central o MTU:** Es el equipo principal del sistema, que supervisa y recolecta datos de las demás subestaciones, ya sean otros equipos vinculados (en sistemas complejos) a los instrumentos de campo o directamente sobre estos mismos instrumentos. Normalmente, este equipo es un PC que soporta el HMI. (Pérez-López, 2015)
- **Ordenadores Remotos o RTU:** Estas computadoras se ubican en los puntos clave del sistema, administrando y controlando las subestaciones; captan las señales de los sensores de campo y instruyen los elementos finales de control mediante la ejecución del software de la aplicación SCADA. Están en la etapa intermedia o de automatización; el MTU se encuentra en una etapa superior y los diversos instrumentos de campo son los encargados de la automatización física del sistema, el control y la recolección de datos. Estos equipos no necesitan ser PC, dado que la demanda para soportar un HMI no es tan alta a este nivel, por lo que generalmente son computadoras industriales de tipo armarios de control, aunque en sistemas muy sofisticados pueden existir subestaciones intermedias en formato HMI.

  (Pérez-López, 2015)
- **Red de comunicación:** Este es el nivel que administra los datos que los instrumentos de campo transmiten a la red de computadoras desde el sistema. El tipo de BUS empleado en las comunicaciones puede fluctuar considerablemente dependiendo de las necesidades del sistema y del

software seleccionado para la implementación del sistema SCADA, dado que no todos los programas (ni los instrumentos de campo como PLC) son capaces de manejar todos los tipos de BUS. (Pérez-López, 2015)

- **Instrumentos de Campo:** Se refiere a todos aquellos que facilitan tanto la automatización o gestión del sistema (PLC, controladores de procesos industriales y actuadores en general) como aquellos responsables de recoger datos del sistema (sensores y alarmas). (Pérez-López, 2015)

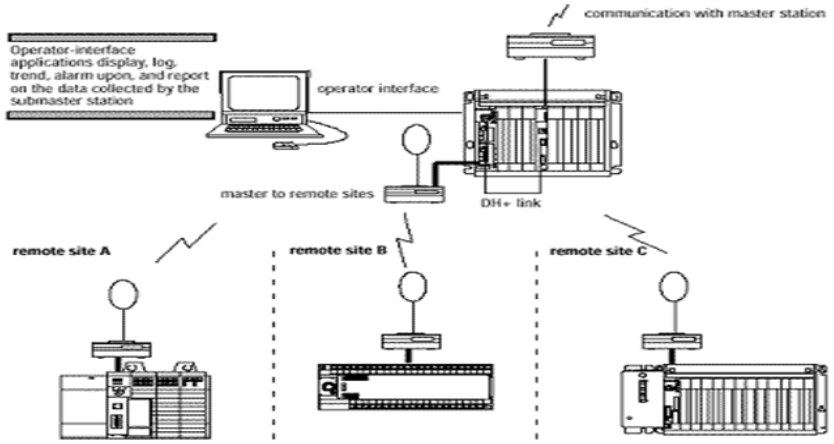
Software

- **Configuración:** facilita establecer el ambiente laboral de la aplicación de acuerdo con la organización de pantallas necesarias y los niveles de acceso para los diferentes usuarios. En este módulo, el usuario determina las pantallas de texto o gráficas que usará, importándolas de otra aplicación o creándolas en el SCADA mismo. En este sentido, se incluye un editor gráfico que facilita el diseño a nivel de píxel (punto de pantalla) o el uso de elementos estándar existentes, como líneas, círculos, textos o figuras, con funciones de edición habituales como copiar, mover, borrar, entre otros. A lo largo de la configuración también se eligen los drivers de comunicación que facilitarán el enlace con los elementos de campo y la conexión o no en red de estos últimos; se elige el puerto de comunicación en la computadora; se elige el puerto de comunicación en la computadora. y sus parámetros, etc. (Pérez-López, 2015)
- **Interfaz gráfica del operador:** ofrece al usuario las responsabilidades de control y supervisión de la planta. El procedimiento que se monitoreará se

ilustra a través de sinópticos gráficos guardados en el equipo y producidos desde el editor integrado en el SCADA o importados desde otra aplicación de uso general (Paintbrush, DrawPerfect, AutoCAD, etc.) durante la configuración del software. (Pérez-López, 2015)

- **Módulo de proceso:** aplica las acciones de control preestablecidas basándose en los valores presentes de las variables analizadas. Cada pantalla permite establecer relaciones entre variables del ordenador o del autómata que se realizan de manera constante mientras está en funcionamiento. La codificación se lleva a cabo a través de bloques de software en lenguaje de alto nivel (C, Basic, etc.). (Pérez-López, 2015)
- **Gestión y archivo de datos:** Se ocupa de guardar y procesar de manera organizada los datos, utilizando formatos entendibles para componentes periféricos de hardware (impresoras, registradores) o software (bases de datos, hojas de cálculo) del sistema, de manera que otra aplicación o dispositivo pueda acceder a estos. (Pérez-López, 2015)

Figura 5 Componentes en un sistema SCADA



Fuente: https://instrumentacionycontrol.net/componentes-en-un-sistema-scada/

2.2.4 Ejemplos de aplicaciones en la industria

Si bien los sistemas SCADA se definen como útiles en el manejo de una gestión de procesos, he aquí unos ejemplos que confirman el dicho:

Caso número 1-Foxboro en Kimberly Clark Costa Rica, planta Belén

Kimberly Clark Costa Rica, con su fábrica en Belén, es una compañía especializada en la producción de artículos de papel empleando como material de origen papel reciclado. Los procedimientos de producción de papel son intrincados y sutiles, por lo que la compañía optó por automatizar algunos de los más esenciales, empleando productos de la línea Foxboro. Cuando se realiza esta labor, se dispone de automatización en una planta de reciclaje (incluyendo el proceso de blanqueamiento), dos maquinarias de papel y la planta de tratamiento de aguas.

En Kimberly Clark, se ha alcanzado el nivel de utilización del SCADA, también conocido como DCS para Foxboro; este sistema es una interfaz que posibilita al operador no solo examinar diversas variables, sino también interactuar con el proceso mediante modificaciones desde el ordenador en su sala de control. El sistema DCS ha brindado a los operadores una mayor gestión de su funcionamiento y manejo de gráficos de tendencias, que se actualizan de manera constante, facilitando la toma de datos y la toma de decisiones a tiempo.

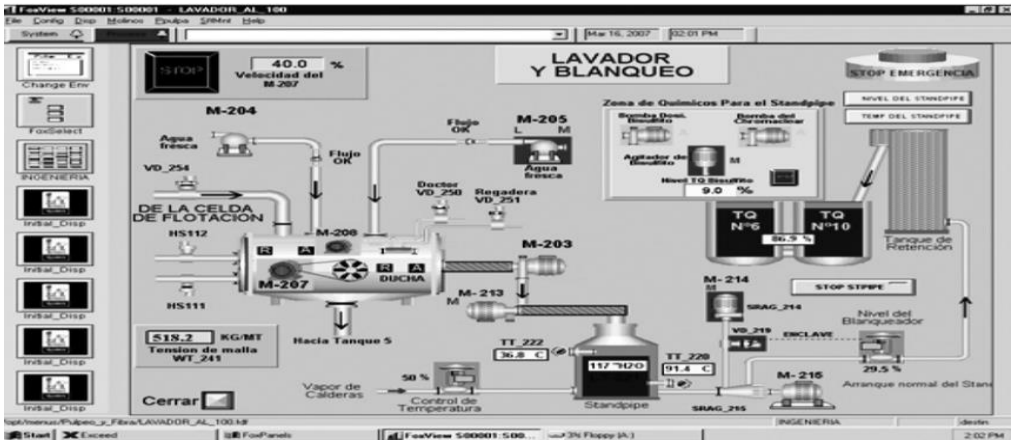
Una de las principales ventajas de la puesta en marcha de este sistema son los ahorros que surgen al tener una mejor comprensión de los flujos de uso

de sustancias químicas, además de mejoras en la calidad del papel generadas al tener una mejor comprensión de las diferentes etapas del proceso.

Como se emplean sustancias químicas, el proceso también ha obtenido un gran beneficio ya que se ha reducido al mínimo el contacto con estas sustancias, salvaguardando así a los trabajadores implicados.

(Pérez-López, 2015)

Figura 6 Pantalla de un SCADA en Kimberly Clark Costa Rica



Fuente: (Pérez-López, 2015)

Caso número 2-Pepsi Garner, Carolina del Norte (EUA)

La fábrica de Pepsi, situada en Garner, Carolina del Norte, se especializa en la producción de productos de consumo masivo en el sector de las bebidas gaseosas. Pepsi Bottling Ventures (PBV) es conocida como la principal fábrica de embotellado en América del Norte. PVB se ha transformado en el tercer mayor productor y distribuidor de Pepsi-Cola en América del Norte, gestionando 27 instalaciones de embotellamiento y distribución en seis estados (Estados Unidos). PBV produce y comercializa más de 100 marcas y sabores distintos.

PBV Garner ha expandido su producción y distribución a 189 productos distintos, alcanzando hoy más de 500. Los cambios de botella y a envases secundarios ocurren diariamente en la planta, y para poder ajustarse a estos cambios, se necesita un nivel singular de flexibilidad en la producción, atención al detalle y acceso a los datos de producción.

Dado que las demandas de producción aumentaban y la tecnología progresaba, PBV buscó una solución de software que permitiera evaluar el desempeño, el tiempo de parada y la eficacia total del equipo. Tras realizar numerosas investigaciones y evaluar diversos sistemas, PBV decidió implementar una solución de rendimiento basada en el software de Gestión de Operaciones de Wonderware Invensys. La solución de Wonderware demostró ser una sólida propuesta de valor para PBV, con el objetivo de examinar de manera exacta sus datos y determinar lo que verdaderamente impacta la producción.

(Pérez-López, 2015)

Figura 7 Pepsi Bottling Ventures de Garner, North Carolina

Fuente: https://www.aveva.com/en/perspectives/success-stories/pepsi-bottling-ventures-of-garner-north-carolina/

2.3 Software Ignition

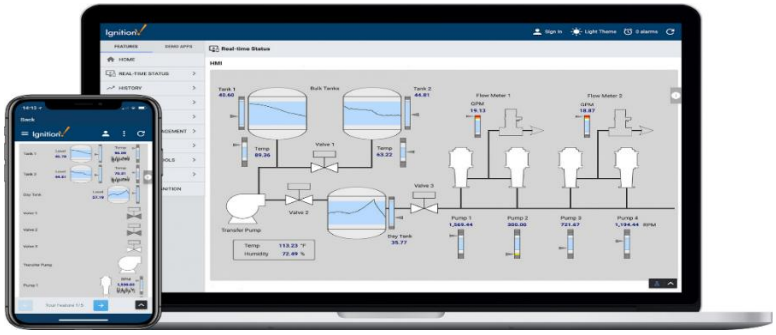
2.3.1 Descripción del software

Ignition SCADA es una solución totalmente integrada que consta de la plataforma Ignition y potentes módulos que agregan funcionalidad a la plataforma. Un paquete estándar de Ignition SCADA consta de estos módulos: Vision, OPC UA (con controladores), Tag Historian, notificación de alarma, informes y módulos de puente SQL (Structured Query Language). (Inductive Automation, 2010)

Ignition es un software SCADA flexible que reúne sistemas de hardware de diferentes proveedores, como como PLC Siemens y Allen-Bradley, en una única estructura de control. (Racharla, Gummadi, & Rawashdeh, 2024)

Ignition funciona como el núcleo de todo en su sistema, recolectando información sin contratiempos y creando cualquier tipo de aplicación industrial de manera sencilla, desde el equipo de producción hasta las bases de datos SQL, reduciendo la brecha entre la producción y la Tecnología de la Información. (Inductive Automation, 2010)

Figura 8 Aplicación de Ignition SCADA



Fuente: https://inductiveautomation.com/distributor-landing/espanol

2.3.2 Características principales

Ignition SCADA representa un cambio de paradigma en el ámbito de software de automatización industrial. A diferencia de los sistemas SCADA tradicionales, Ignition se basa en tecnologías modernas como Java, bases de datos SQL (Structured Query Language) y arquitecturas basadas en web, estas ofrecen escalabilidad, flexibilidad y facilidad de uso incomparables. Ignition se destaca por sus capacidades en interactuar con PLC y facilitar la adquisición de datos visuales y el control. (Racharla, Gummadi, & Rawashdeh, 2024)

Si enlistamos sus características principales, tenemos estas que destacan:

- **Licencia Ilimitada:** Tanto sus clientes, tags, conexiones y escalabilidad serán ilimitados.
- **Implantación basada en Web:** Basado en navegador, java y en base de datos SQL con un servidor centralizado, cuya instalación no se demora más de unos minutos.
- **Monitorización y control en tiempo real:** Control de estado en tiempo real con su propio sistema de alarmas, capacidad de crear reportes y adquirir datos en tiempo real para la toma de decisiones.
- **Estabilidad y seguridad:** Sistema redundante con tecnología SSL (Secure Sockets Layer) y auditoria en tiempo real.
- **Rápido desarrollo:** Se puede desarrollar varios clientes a la vez en un entorno orientado a objetos con herramientas de Dibujo e importación de gráficos. Incluye lenguaje para scripts.

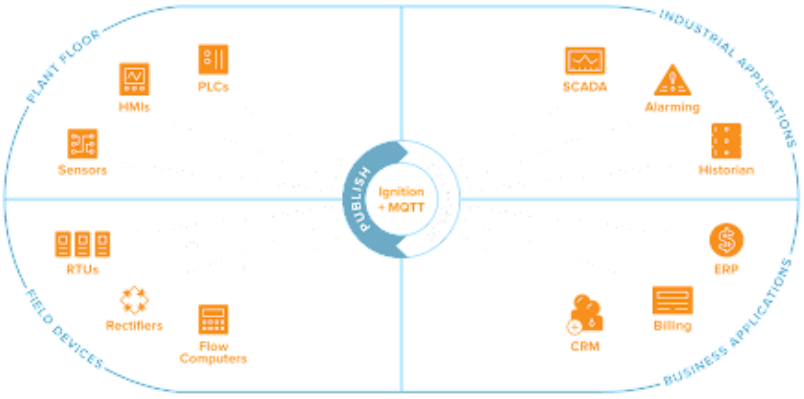
- **Plataforma universal:** Se puede aplicar cualquier tipo de modulo necesario para el desarrollo del sistema, todo en una sola plataforma.

2.3.3 Aplicaciones de Ignition SCADA

Ignition es un poderoso ambiente de desarrollo que abarca todo lo requerido para desarrollar prácticamente cualquier tipo de aplicación industrial, todo en una única plataforma. Estas aplicaciones pueden ser:

- SCADA (Supervisory Control and Data Acquisition):
- IIOT (Internet Industrial de las Cosas)
- MES (Manufacturing Execution System)
- HMI (Interfaz Hombre-Maquina)
- Alarmas
- Reportes
- Computación Edge
- Empresarial
- Móvil

Figura 9 Aplicaciones de Ignition



Fuente: https://www.nvtecnologias.com/ignition-iiot

2.4 Protocolo OPC UA

2.4.1 Definición y objetivos del estándar OPC UA

Open Productivity Collaboration Unified Architecture (OPC UA) fue desarrollado con el objetivo de superar las limitaciones que presentaba su antecesor OPC Clásico. El mismo que fue diseñado para funcionar únicamente con sistemas Operativos Windows. (Montalvo, Encalada, Miranda, Garcia, & Garcia, 2020)

En la actualidad, OPC UA es una plataforma autónoma de los fabricantes y por ende abierta, ya que garantiza el intercambio de datos entre componentes de diferentes proveedores, siendo la OPC Fundation la entidad responsable de la creación de este estándar. OPC UA es una tecnología creada para transmitir información de manera segura y funcional entre aplicaciones en el sector de la automatización y en otros campos. (Montalvo, Encalada, Miranda, Garcia, & Garcia, 2020)

La nueva especificación OPC UA se ha creado debido a que cada vez son más desafíos en el modelado y seguridad de datos. En cambio, era imprescindible un protocolo más compatible con otros sistemas operativos y, por ende, aplicable a tecnologías futuras. (Montalvo, Encalada, Miranda, Garcia, & Garcia, 2020)

2.4.2 Principales características

OPC UA cuenta con dos destacables características:

- Sustituye el protocolo COM y DCOM, particular de Windows, por protocolos abiertos e independientes que puedan funcionar en otros sistemas operativos con una implantación significativa no solo en equipos de computación como Windows, Linux, Mac, etc., sino también en dispositivos móviles como Android y variados tipos de controles y dispositivos de vigilancia, sensores, controladores, entre otros. Que se relacionan con el mundo real incluyendo también mecanismos de protección extra.
- Incluye el modelo orientado a objetos de información que integra las funcionalidades convencionales de OPC (tales como acceso a datos, registro, alertas, eventos, condiciones, etc.) y otras novedosas e innovadoras enfocadas en los tipos de datos y métodos.

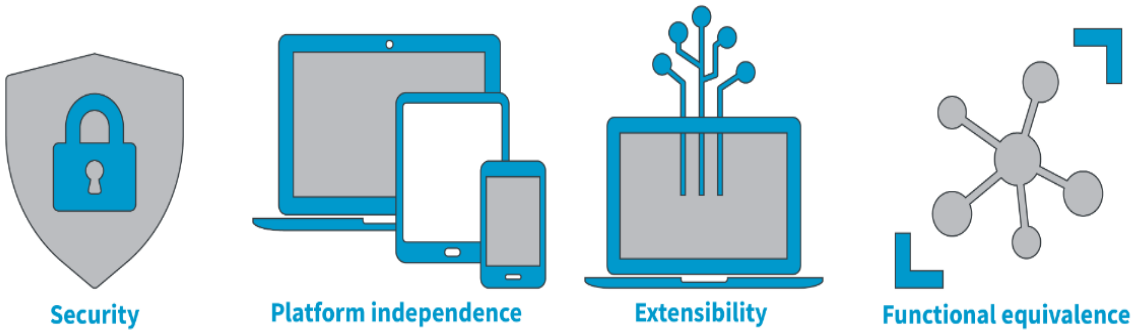
  (Montalvo, Encalada, Miranda, Garcia, & Garcia, 2020)

El hecho de que en los servidores OPC UA la estructura de direccionamiento sea orientada a objetos y que la interfaz para acceder a dicha estructura sea completamente genérica hace que OPC UA no solo sea visto como una pasarela de comunicación entre dos equipos, sino también como un lenguaje de programación con habilidades para comunicarse a través de redes. (Montalvo, Encalada, Miranda, Garcia, & Garcia, 2020)

2.4.3 Ventajas del uso de OPC UA en sistemas de automatización

- **Seguridad:** OPC UA es una plataforma que funciona en compatibilidad con cortafuegos, proporcionando varios niveles de cifrado, autenticación, auditoría y gestión de usuarios.
- **Independiente de cualquier plataforma:** OPC UA no tiene relación con ningún sistema operativo o lenguaje de codificación específico. Es compatible con varias plataformas de hardware (como PC, servidores de Internet, etc.) y sistemas operativos (como Windows, Apple OSX, Android, entre otros).
- **Capacidad de extensión:** OPC UA es un sistema orientado hacia el futuro, diseñado para incorporar la tecnología futura. En resumen, está concebido para ajustarse a tecnologías que todavía no han sido creadas, manteniendo compatibilidad con iteraciones previas de productos ya existentes, incluso a medida que estos progresan.
- **Equivalencia funcional:** OPC UA es una ampliación de la tecnología OPC previa que todavía puede incorporarse de manera sencilla en el nuevo sistema optimizado.

Figura 10 Ventajas que ofrece OPC UA



Fuente: https://www.atlascopco.com/es-ec/compressors/wiki/compressed-air-articles/opc-ua-and-its-benefits

2.5 Procesos de Llenado y Envasado en la Industria Alimentaria

2.5.1 Descripción de los procesos de llenado y envasado

El llenado y envasado en la industria alimentaria se define como el conjunto de procesos por la cual se garantiza la calidad y seguridad de los productos alimenticios, para que estos sean trasportados a los diferentes mercados de forma que no afecten su estado. Dependiendo del producto estos envases pueden ser de distintos materiales y formas, así como su forma de llenarlas suelen variar.

Desde luego, no solo se realizan estos procesos por cuestión de calidad y seguridad, sino inclusive en la etiqueta, presentación e identidad del producto. Como se presenta el producto ante el mercado es muy importante para determinar si este logra ser adquirido por los consumidores y generar ganancias.

Sin embargo, estos procesos principalmente tienen de objetivo proteger el producto de las condiciones adversas ambientales como la temperatura, humedad, luz y otros contaminantes tanto naturales como artificiales. Adicionalmente, estos envases logran conservar la calidad de su producto por varios años como en el caso de frutos secos y granos.

Las etapas importantes que conllevan estos procesos son:

1) **Preparación del envase:** Es requerido tenerlos alineados y asegurados como corresponde para evitar riegos o algún error de cálculo en caso de usar maquinarias o sistemas.

2) **Llenado con la cantidad precisa del producto:** Así evitas perdidas y aseguras que cada producto tenga la misma cantidad especificada la cual ayuda a evitar problemas de subllenado o sobrellenado que afectan a la consistencia y calidad del producto.
3) **Sellado del envase:** Con esto aseguras el envase y proteges el producto de las condiciones adversas que podrían afectar su calidad.
4) **Etiquetado del envase:** Paso final que distingue la marca de tu producto de otros que hacen competencia en el mercado y ayuda a especificar detalles que conllevan el envase como la cantidad de gramos, litros o cantidad de calorías, etc....

Estas etapas son ilustradas en la siguiente figura como un ejemplo simple de este proceso en una industria:

Figura 11 Máquinas de llenado y cierre de envases

Fuente: https://www.jbtc.com/foodtech/es/products-and-solutions/solutions/filling-and-closing/

2.5.2 Normativas y Estándares en la Industria Alimentaria

Como toda industria, existen normativas y estándares sujetadas a la buena práctica de la elaboración de alimentos ya que estas afectan a la salud del consumidor. Si no se sigue a palabra estos estándares, existirían mayores repercusiones a la salud pública y surgirían problemas legales con la institución gubernamental quien dispuso de estas normativas, sin hablar de la mala imagen que recibiría la compañía. Con la involucración de la tecnología a los procesos de manufacturación de alimentos, se tuvieron que actualizar las normativas y estándares con visión al futuro.

La constitución política del Ecuador firmo un decreto con el reglamento de buenas prácticas para alimentos procesados, en las cuales sus artículos exclaman, con respecto a estos procesos:

- **Art. 8.-** La selección, fabricación e instalación de los equipos deben ser acorde a las operaciones a realizar y al tipo de alimento a producir. El equipo comprende las máquinas utilizadas para la fabricación, llenado o envasado, acondicionamiento, almacenamiento, control, emisión y transporte de materias primas y alimentos terminados.
- **Art. 9.-** MONITOREO DE LOS EQUIPOS: Condiciones de instalación y funcionamiento.
    1. La instalación de los equipos debe realizarse de acuerdo con las recomendaciones del fabricante.
    2. Toda maquinaria o equipo debe estar provista de la instrumentación adecuada y demás implementos necesarios para su operación, control

y mantenimiento. Se dispondrá de un sistema de calibración que posibilite garantizar que tanto los equipos y maquinaria, como los instrumentos de control, ofrezcan mediciones fiables.

Además, el funcionamiento de los equipos contempla lo siguiente: que todos los componentes del equipo que se encuentren en contacto con materias primas y alimentos durante el proceso deben ser limpios para prevenir contaminaciones.

- **Art. 41.-** Todos los alimentos deben ser embalajes, rotulados y empaquetados de acuerdo con las normas técnicas y reglamentarias correspondientes.
- **Art. 42.-** El diseño y los materiales de embalaje deben proporcionar una defensa apropiada para los alimentos con el fin de minimizar la contaminación, prevenir perjuicios y permitir un etiquetado acorde a las regulaciones técnicas correspondientes. Cuando se emplean materiales o gases para el embalaje, estos no deben ser tóxicos ni constituir un peligro para la seguridad y la aptitud de los alimentos bajo las condiciones especificadas para su almacenamiento y uso.
- **Art. 51.-** Para evitar que las partículas del empaque contaminen los alimentos, las operaciones de llenado y empaquetado deben llevarse a cabo en zonas distintas.

(Organo del gobierno del Ecuador-Tribunal Constitucional, 2002, 04 de noviembre)

2.5.3 Problemas comunes en procesos manuales o poco automatizados

Como se sabe, la automatización brinda beneficios que ayudan a las empresas dedicadas a esta industria de la alimentación a elevar sus ganancias al aumentar la producción de sus productos. Sin embargo, no todas las empresas pueden darse el lujo de automatizar sus procesos ya que esto requiere de una gran inversión en capital con sus maquinarias, sistemas, materiales electrónicos y hasta para capacitar el personal o contratar personal especializado. Por ende, estas empresas deciden seguir con la contratación de personal humano para llevar a cabo estos procesos de forma manual. Esta decisión hace que surja problemas las cuales se pudiesen evitar con la automatización, las cuales son:

- **Inconsistencia en la cantidad del llenado:** Esto puede causar que varíen en la cantidad de producto debido a la falta de precisión o de herramientas de medición, lo cual llevaría a una inconsistencia que afecta a la satisfacción del cliente e incumplimiento de normativas.
- **Contaminación del producto:** El ser humano es capaz de transportar miles de bacterias en el cuerpo, como se tiene que manipular con las manos el producto, la falta de higiene no solo afectaría la calidad sino también podría ser un peligro para la salud.
- **Desperdicio:** La errónea manipulación de los productos o líquidos puede llegar a ensuciar el ambiente de trabajo y generar perdidas de tiempo y dinero con solo la limpieza, considerando también la pérdida de la materia que pudo haberse usado para envasar más productos.

- **Sellado defectuoso:** El no sellar adecuadamente o de forma precisa puede llevar a que el producto se exponga a las condiciones adversas del ambiente la cual terminara afectado la calidad final y pone en peligro a la salud pública.
- **Baja productividad:** Se vuelve difícil de cumplir la demanda por ser un proceso manual no optimizado.
- **Variación en la presentación del producto:** Por la inconsistencia y desigualdad en el envase lleva al incumplimiento de estándares de presentación.
- **Problemas Ergonómicos:** El ser humano no es una máquina, repetir el mismo proceso por horas lleva al cansancio y la fatiga mental lo que aumenta la probabilidad de errores humanos y en el peor de los casos, accidentes laborales.
- **Incumplimiento de las normativas legales:** Esto lleva a múltiples multas las cuales se vuelven difíciles de costear, en el peor de los casos, tu producto termina siendo retirado del mercado y eventualmente la empresa es clausurada.

Estos problemas comunes pueden ser evitados con la implementación de la automatización, pero eso no quita que algunas empresas han logrado subsistir y prosperar sin necesidad de implementarlas. Tras capacitar al personal, proveer de uniformes pulcros, limpiar con frecuencia el ambiente de trabajo, cumplir con las normativas legales y proporcionar descanso a los trabajadores o cambiar de

turnos para evitar la fatiga, logras evitarte estos problemas de manera legal pero también cuenta con su costo de tiempo y dinero distinto.

Figura 12 Seguridad industrial en el sector de la alimentación

Fuente: https://www.grupo-tice.com/seguridad-industrial-en-el-sector-de-la-alimentacion/

CAPITULO 3: ANALISIS DEL DISEÑO Y RESULTADOS

3.1 Diseño del sistema SCADA para el llenado y envasado en Ignition

3.1.1 Consideraciones previo al diseño del sistema SCADA

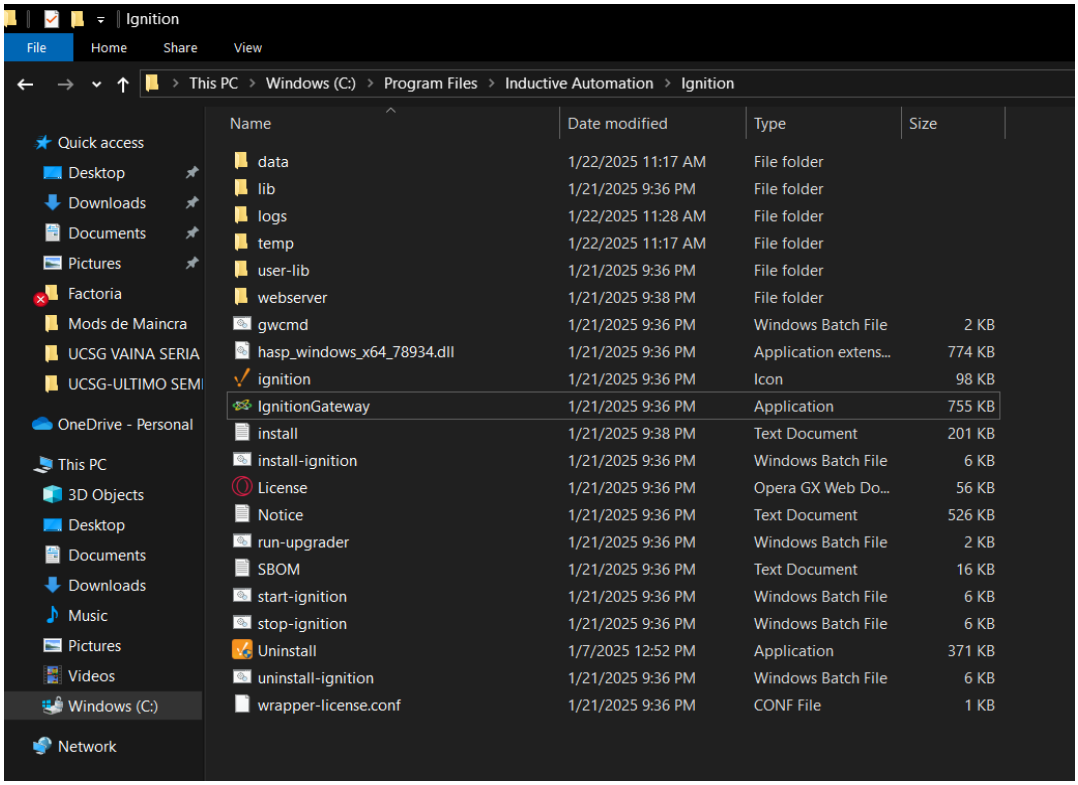
Como se explicó en el capítulo 2, en la industria se ha reconocido a Ignition SCADA como una nueva alternativa al diseño SCADA y conexión de diferentes bases de datos y protocolos de comunicación eficiente. Por ese motivo, se escogió este software para diseñar el sistema SCADA de un llenado y envasado de productos alimenticios, en este caso botellas de bebidas. Sin embargo, una ventaja que ofrece Ignition y los sistemas SCADA en sí, es su versatilidad y flexibilidad lo que permite cambiar el diseño y acomodar las mismas funciones para diferentes productos.

Antes de empezar, es crucial que tengas descargado en tu ordenador dos softwares de Ignition: el Ignition Gateway y el Ignition Designer. El Gateway te permite conectar a un servidor local en la web de tu preferencia para que puedas controlar y supervisar tus procesos una vez creados, conectar a diferentes bases de datos y hasta PLCs de forma gratuita y accesible. Esta incluye también la función de guardar tu progreso en cuanto empieces a diseñar como una nube. El Designer, como su nombre lo dice, es el software principal que te ayudara con el diseño del sistema que deseas crear con diferentes tipos de conexiones si así se desea.

Para que puedas empezar a diseñar, necesitas tener activado los servicios de Ignition Gateway todo el tiempo que dure tu proceso y reactivarlo cuando reanudes tu progreso. Es un proceso sencillo que consta de solo buscar en tus

archivos de computadora: C:\Program Files\Inductive Automation\Ignition, haces doble clic en “IgnitionGateway” y con eso activas los servicios.

Figura 13 Ubicación de los archivos de Ignition



Fuente: El autor

Una vez activado, se registra en la página de Ignition localhost (en este caso 8088), creando un usuario con contraseña y eso sería todo con respeto al Gateway.

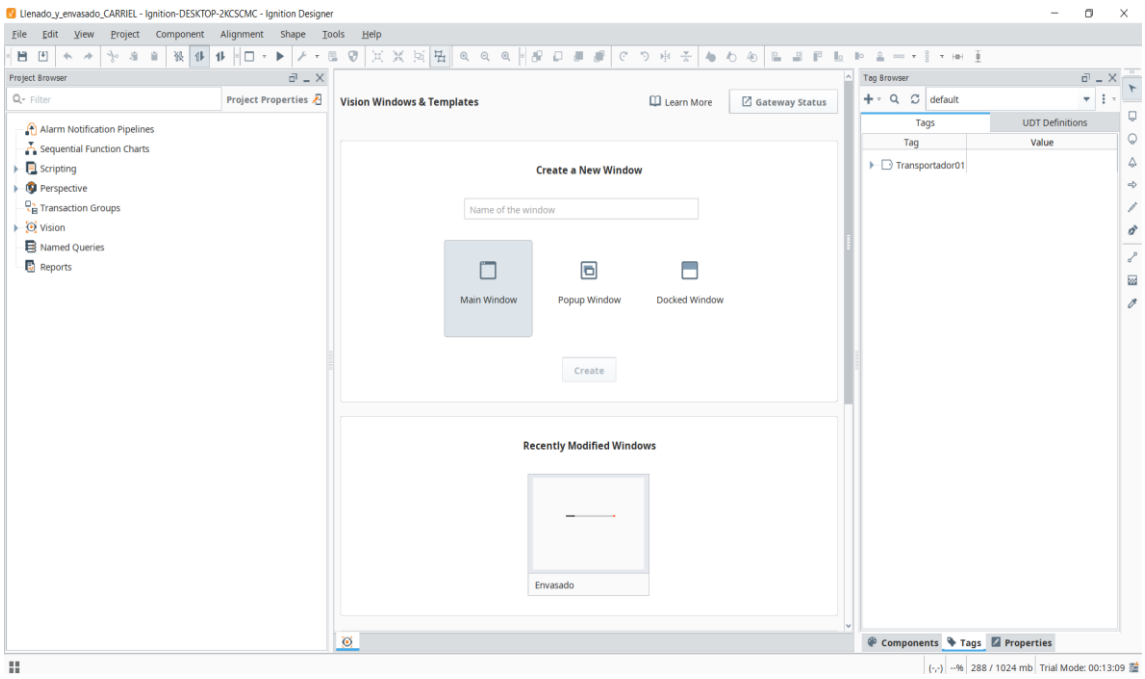
Continuando con el Designer de Ignition, una vez instalado te pide que escribas la misma cuenta que creaste para el Gateway con su contraseña y crees un proyecto. Se le puede poner cualquier nombre que el usuario quisiese, siendo todo un proceso sencillo y rápido de entender para cualquier persona que desee iniciar sus proyectos con Ignition.

3.1.2 Procedimiento del diseño SCADA y sus componentes principales

A continuación, se explicará detalladamente el proceso arduo que tomo este diseño de un sistema SCADA de llenado y envasado de botellas de colas:

En el entorno de trabajo del diseñador tenemos 3 partes importantes que son el árbol de trabajo (Izquierda), la ventana principal donde se hace el diseño (Medio) y la zona de los tags, componentes y propiedades (Derecha).

Figura 14 Entorno de trabajo de Ignition Designer



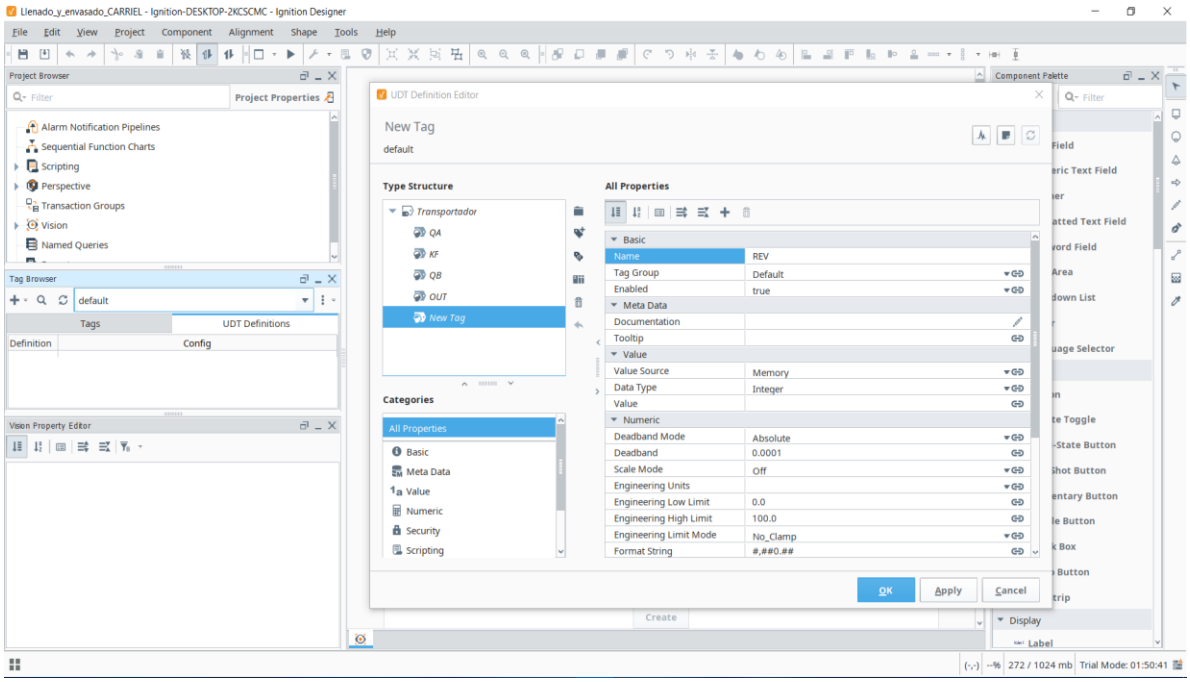
Fuente: El autor

El primer paso consiste en crear lo que se conoce como “tags”. Los tags son básicamente objetos que contienen datos, atributos y valores según su tipo de data que pueden ser integrales, booleanos, flotantes, etc… Es la principal forma de mover estos valores o datos entre los proyectos de Ignition, estas pueden venir de cualquier fuente como la dirección OPC, PLCs, entre otros.

Sabiendo lo que son los tags, creamos en UDT definitions, presionando el símbolo de **+**, un grupo de instancia que conformaran las funciones y datos de una cinta transportadora la cual llamaremos “Transportador01”, entre sus tags tenemos los siguientes:

- **QA:** Seleccionador del campo, data booleana (bool)
- **KF/RUNNING:** Señal de confirmación de encendido, dato booleano (bool)
- **QB:** Señal del guardamotor, data booleana (bool)
- **OUT:** Señal de encendido, data booleana (bool)
- **REV:** Señal de reversa, data booleana (bool)
- **FOR:** Señal de delante, data booleana (bool)
- **FLT:** Falla, data booleana (bool)
- **ST:** Estatus del motor [0,1,2,3], data entera (int)

Figura 15 Creación de los tags de la cinta transportadora



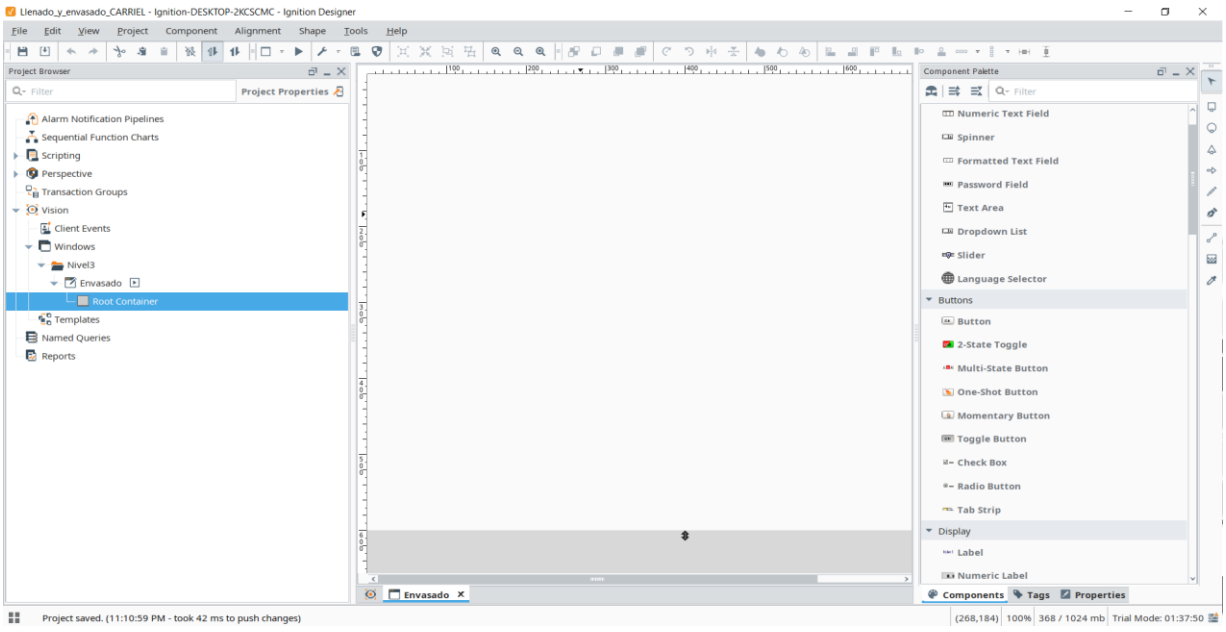
Fuente: El autor

Una vez creados los tags, en el árbol de trabajo por debajo de visión hacemos clic derecho en Windows para crear una nueva carpeta la cual llamaremos Nivel 3.

Nota: *Usualmente en las industrias existen 3 niveles en un sistema SCADA; Nivel 1 Inventario, Nivel 2 información de la planta y Nivel 3 Pantallas de control de instrumentos. El diseño solo abarcara el nivel 3.*

En la carpeta “Nivel 3” creamos un template la cual servirá como nuestro proceso de envasado y lo llamamos “Envasado”. En ese template es donde colocaremos nuestros componentes necesarios y diseñaremos el sistema en base a lo que se necesite.

Figura 16 Creación de la carpeta “Nivel 3” y del template “Envasado”

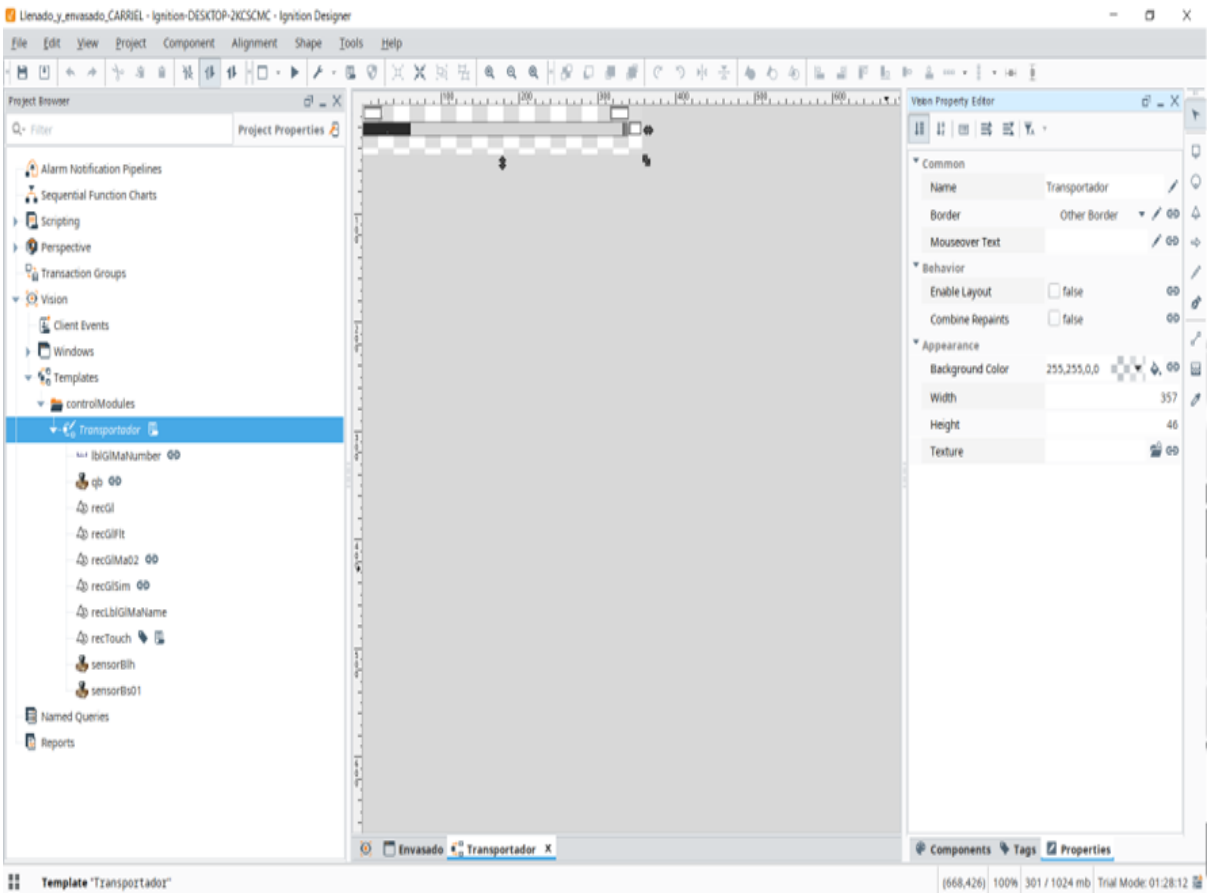


Fuente: El autor

Siguiendo con el diseño, si no podemos conseguir los componentes de la lista por defecto en Ignition, no hay problema ya que este permite importarlos de

bibliotecas o hasta crear tus propios componentes. Importamos los componentes de una cinta transportadora y lo llevamos al área de trabajo, creando así también un nuevo template donde podemos configurar sus parámetros y darle valores.

Figura 17 Cinta transportadora importada



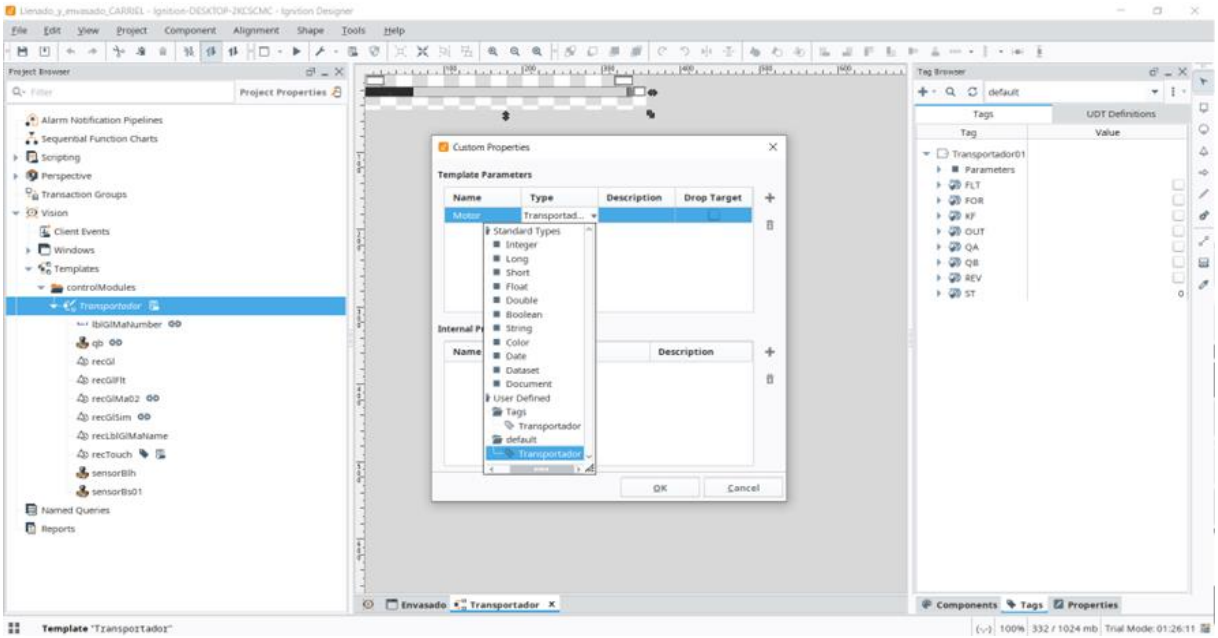
Fuente: El autor

Como se observa en la figura 17, hemos importado una cinta transportadora con sus propios componentes y parámetros ya establecidos. Estos componentes y parámetros necesitamos editarlos para que funcionen en base a nuestros tags, para eso hacemos lo que se conoce como “binding”.

A lo que nos referimos con binding es un tipo de enlace que vincula propiedades del componente a un tag. Esencialmente cambiamos sus propiedades y su comportamiento en el editor para que estos actúen cuando el tag vinculado se activa o desactiva.

Por el momento, solo le cambiamos los parámetros del template de la cinta transportadora para que esta actúe en cuanto uno de los tags del grupo de instancia “Transportador01” se active.

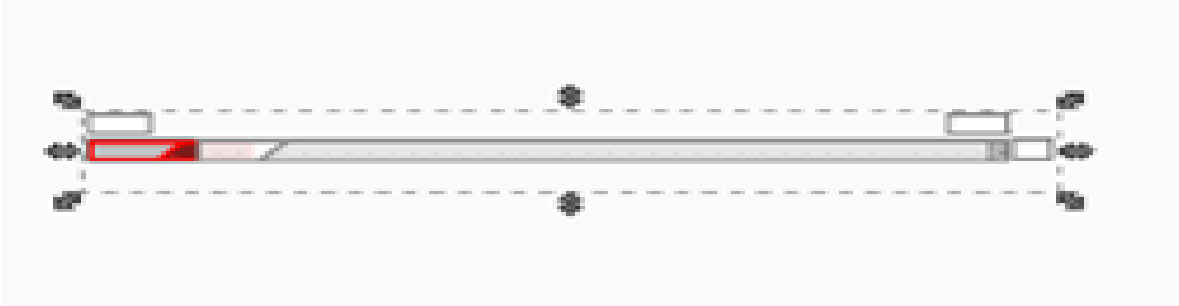
Figura 18 Cambio de parámetros



Fuente: El autor

La cinta transportadora la arrastramos a la ventana de Envasado y nos da como resultado nuestra primera cinta transportadora en el diseño como tal:

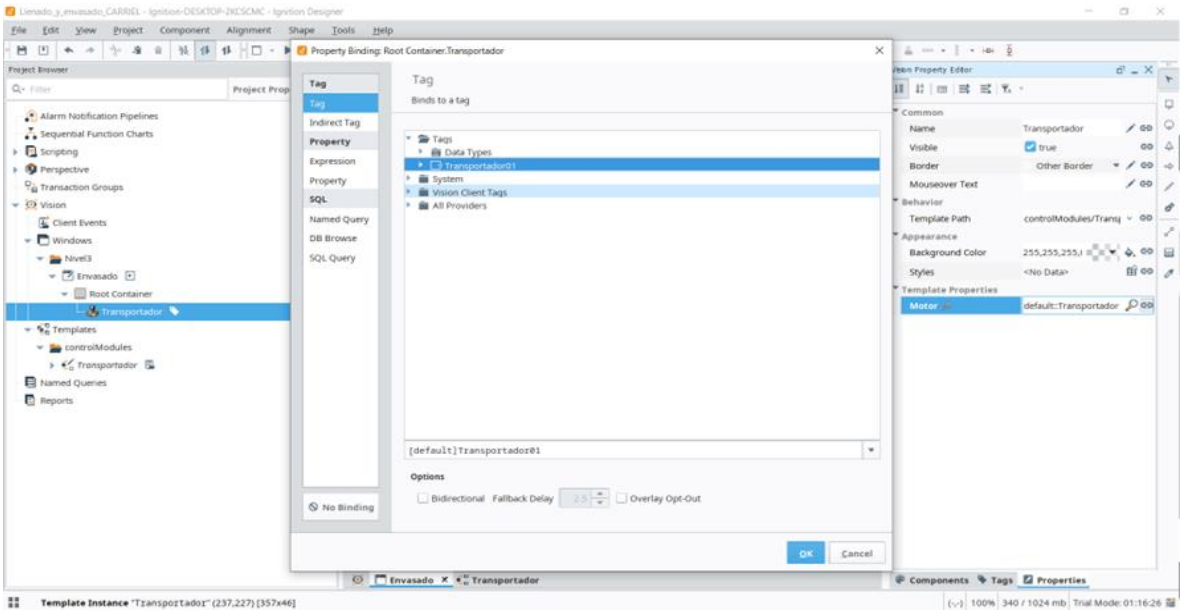
Figura 19 Cinta transportadora en la ventana de envasado



Fuente: El autor

Para hacer el binding como tal, en la ventana derecha de propiedades nos dirigimos a las propiedades del template (Template Properties) y hay un símbolo que parece un doble clip alado de la lupa, al hacer clic se nos abre la ventana de “Property Binding” y vinculamos la cinta transportadora con el grupo de instancia llamado “Transportador01”

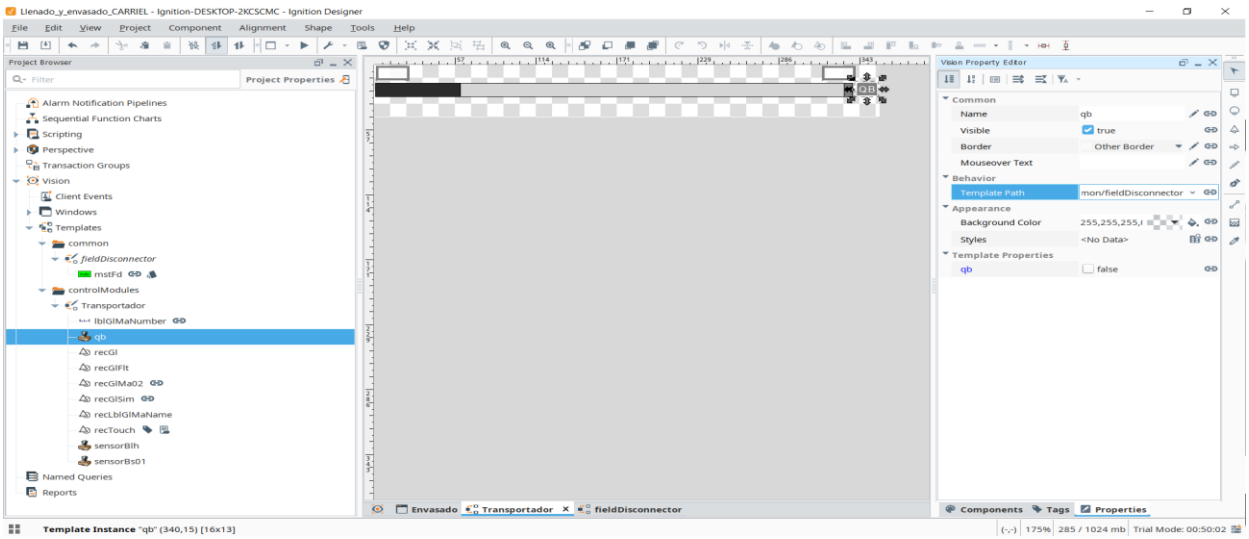
Figura 20 Binding de la cinta transportadora al grupo de instancia



Fuente: El autor

Explicado lo que es el binding, empezamos a vincular los parámetros y propiedades de los demás componentes con nuestros tags, empezando con “qb”, que si bien recordamos es el guardamotor de nuestra cinta transportadora. El guardamotor se encargará de que en el caso de que existiese un sobrecalentamiento en el motor, este se activa para detener el proceso y proteger el motor. Funciona similarmente a un relé térmico.

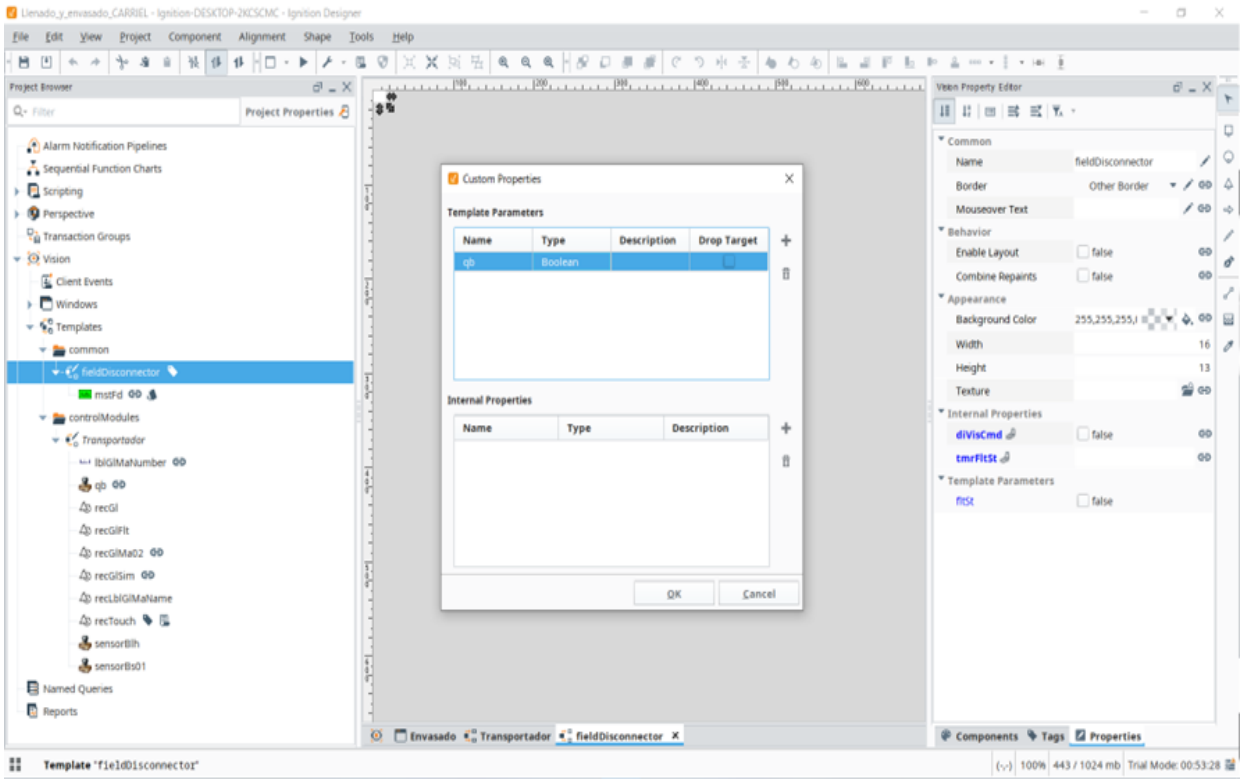
Figura 21 “Qb” componente guardamotor de la cinta transportadora



Fuente: El autor

Cabe mencionar que antes de vincular qb, se requirió importar el componente como tal, la cual se lo conoce como “fieldDisconnector”. Lo importamos a nuestro diseño y lo movemos a la carpeta “common” de los templates para que se integre a nuestro diseño. Cambiamos sus parámetros para que funcionen en base al grupo de estancia creado como se hizo con la cinta transportadora.

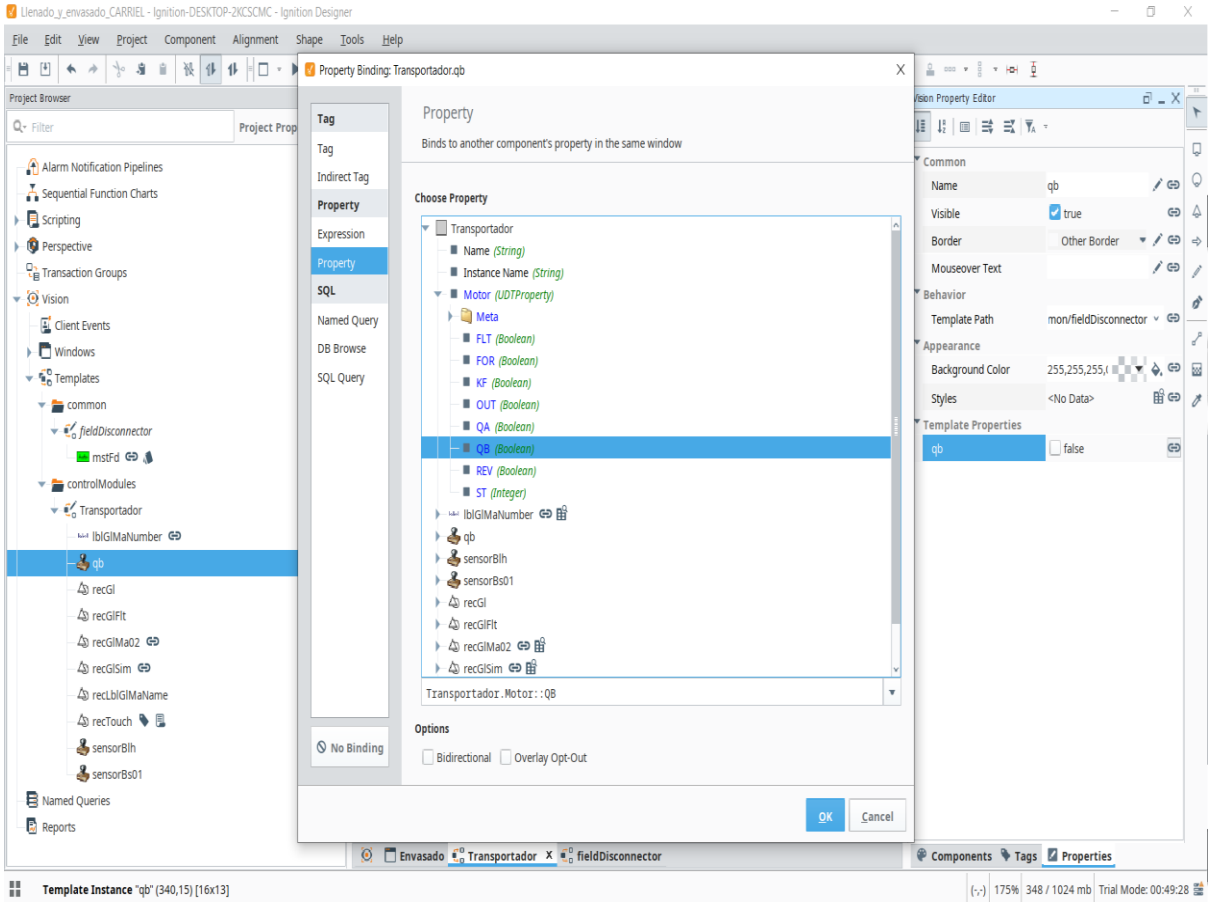
Figura 22 Cambio de parámetros del "fieldDisconnector" al tag QB



Fuente: El autor

Proseguimos con la vinculación del componente del guardamotor (QB) y notamos el "Multi-State Indicator" o Indicador de múltiples estados. Este indicador funciona similar a un LED que nos indica cuando el guardamotor este activado. Es muy importante ya que sirve como aviso al usuario de que algo salió mal con el motor que se tuvo que activar para detener el proceso y evitar daños en la producción. Vinculamos el componente con nuestro tag y editamos sus parámetros una vez más y con eso ya tendríamos listo el guardamotor de nuestra cinta transportadora en el sistema de envasado.

Figura 23 Vinculación del componente qb al tag QB

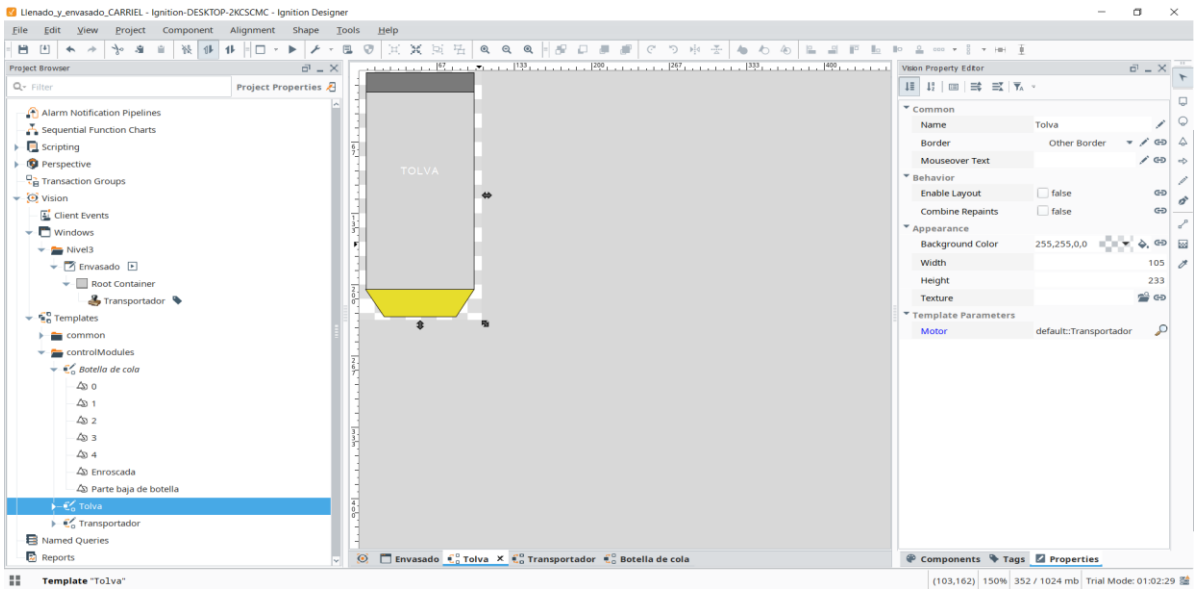


Fuente: El autor

Continuando con el diseño, un sistema de llenado y envase necesita de varios componentes necesarios como Tolvas, que almacenan líquidos, válvulas que permiten el flujo de estos líquidos almacenados en la tolva y las botellas en sí.

Ignition Designer ofrece una biblioteca de símbolos que contiene más componentes de las que normalmente ofrece la ventana. Usando símbolos básicos podemos dibujar nuestros componentes y luego darle cada parte una función. Como por ejemplo esta tolva en la figura 24.

Figura 24 Tolva creada con figuras básicas



Fuente: El Autor

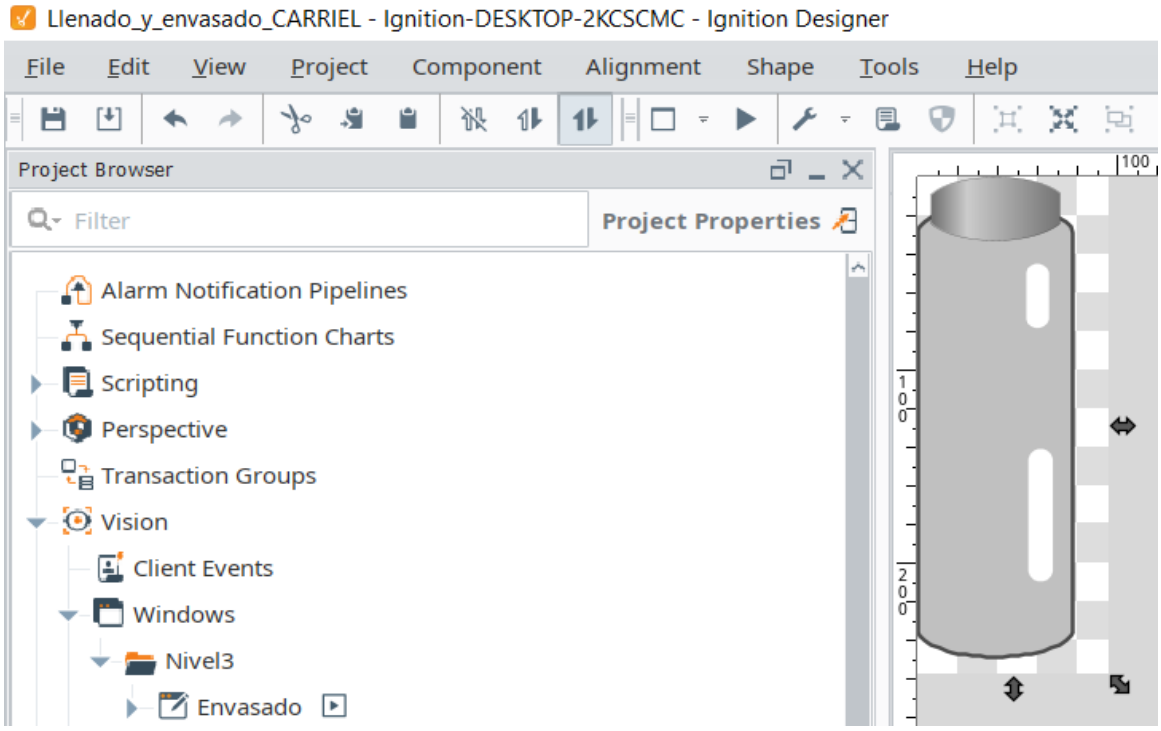
En el diseño principal pondremos 4 de esta Tolva que servirá como almacenamiento del líquido, considerando que en una industria se requerirá llenar múltiples botellas, alcanzando cifras de miles de botellas por día, tener múltiples Tolvas permitirá que la producción no se detenga para llenar una sola tolva.

Se diseña la botella usando figuras básicas como con la tolva. Sin embargo, a la botella se le añade 5 capas distintas del cuerpo. Estas 5 capas representan el estado de llenado de la botella con:

- 0=0 %
- 1=30%
- 2=50%
- 3=75%
- 4=100%

La idea es animar el llenado de las botellas en cuando se corra el programa entero. ***Nota:*** *Es importante agrupar cada parte de la botella para que funcionen como una sola figura en cualquier otro template.*

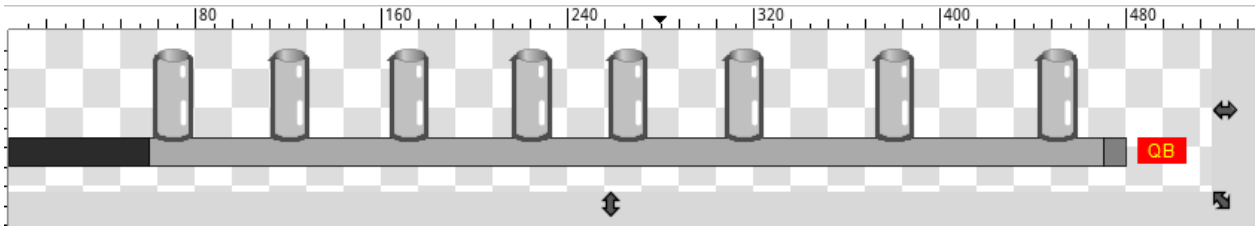
Figura 25 Botella de cola



Fuente: El autor

Una vez diseñadas las botellas, se copia y pega 8 figuras del template de la botella al template de la cinta transportado de tal manera que se muestra a continuación:

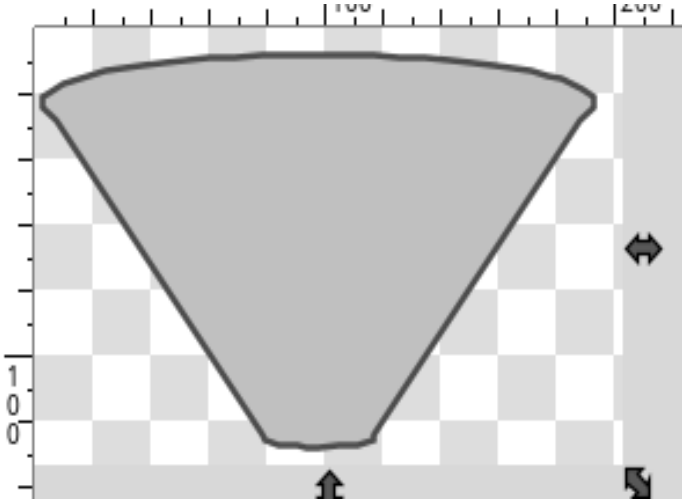
Figura 26 Botellas de cola en el template de la cinta transportadora



Fuente: El autor

Luego se diseña una válvula, así como los demás componentes usando figuras básicas, la cual esta se abrirá o cerrará para permitir el flujo de la formula hacia la maquina dispensadora que llenará las botellas de cola. Para simular el encendido o apagado de esta válvula se añade la misma figura sobre la otra de un color distinto (en este caso amarillo) y se desactiva su visibilidad.

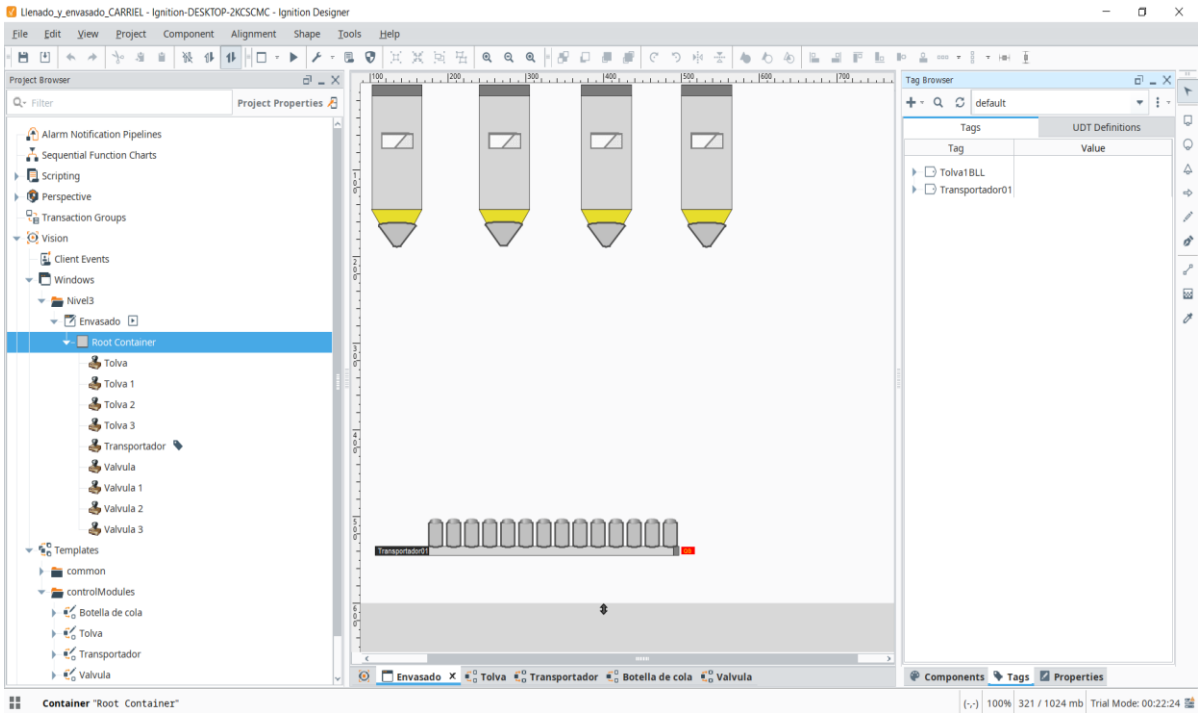
Figura 27 Válvula de la tolva



Fuente: El autor

Pasamos las Tolvas, válvulas y la cinta transportadora con las botellas de cola a nuestro diseño de envase y llenado y nos da como resultado lo que se presencia en la figura 28. Cuatro tolvas por encima de la cinta transportadora y debajo de ellas las válvulas que se abrirán o cerrarán a comando del usuario una vez que las vinculemos con sus respectivos tags.

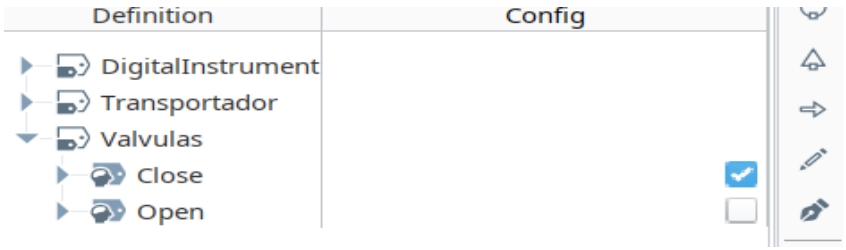
Figura 28 Template principal del sistema de gestión de llenado y envasado con componentes necesarios



Fuente: El autor

Como mencionado anteriormente, las válvulas requieren de dos tags booleanos para simular el cerrado y abierto de estas. Para esto se crea un nuevo grupo de estancia en UDT definitions la cual se llamará “Válvula” y creamos dos tags booleanos, la cual llamaremos “Open” y “Close”. ***Nota:*** *Un pequeño ajuste que se realiza es que el tag “Close” en vez de tener valor por defecto false, este será true.*

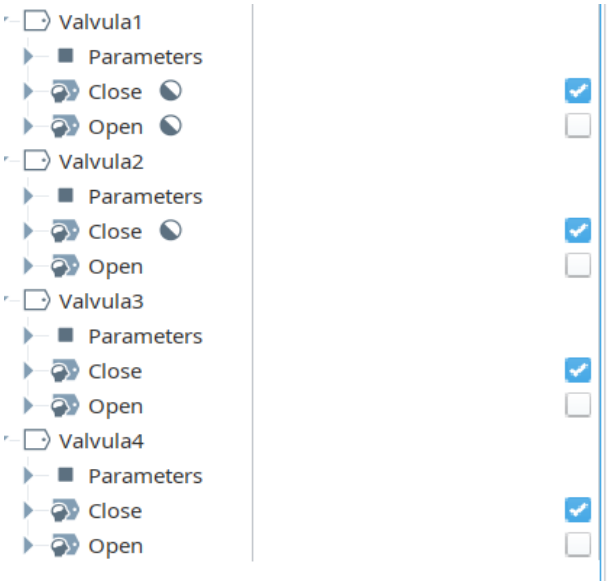
Figura 29 Grupo de estancia “Válvula” en UDT definitions



Fuente: El autor

Considerando que existen cuatro válvulas, una por cada tolva, si se quisiera se podría hacer que cada válvula funcione al mismo tiempo con un solo tag de Open y Close, pero usualmente en las industrias no pueden parar la producción por la falta de líquido en una tolva así que, en la pestaña de tags, cada válvula tendrá sus propios tags de abierto y cerrado para dejar fluir el liquido de su tolva correspondiente o cerrar el flujo para dejar que la tolva con nivel bajo de líquido se rellene eventualmente como se muestra en la figura 30.

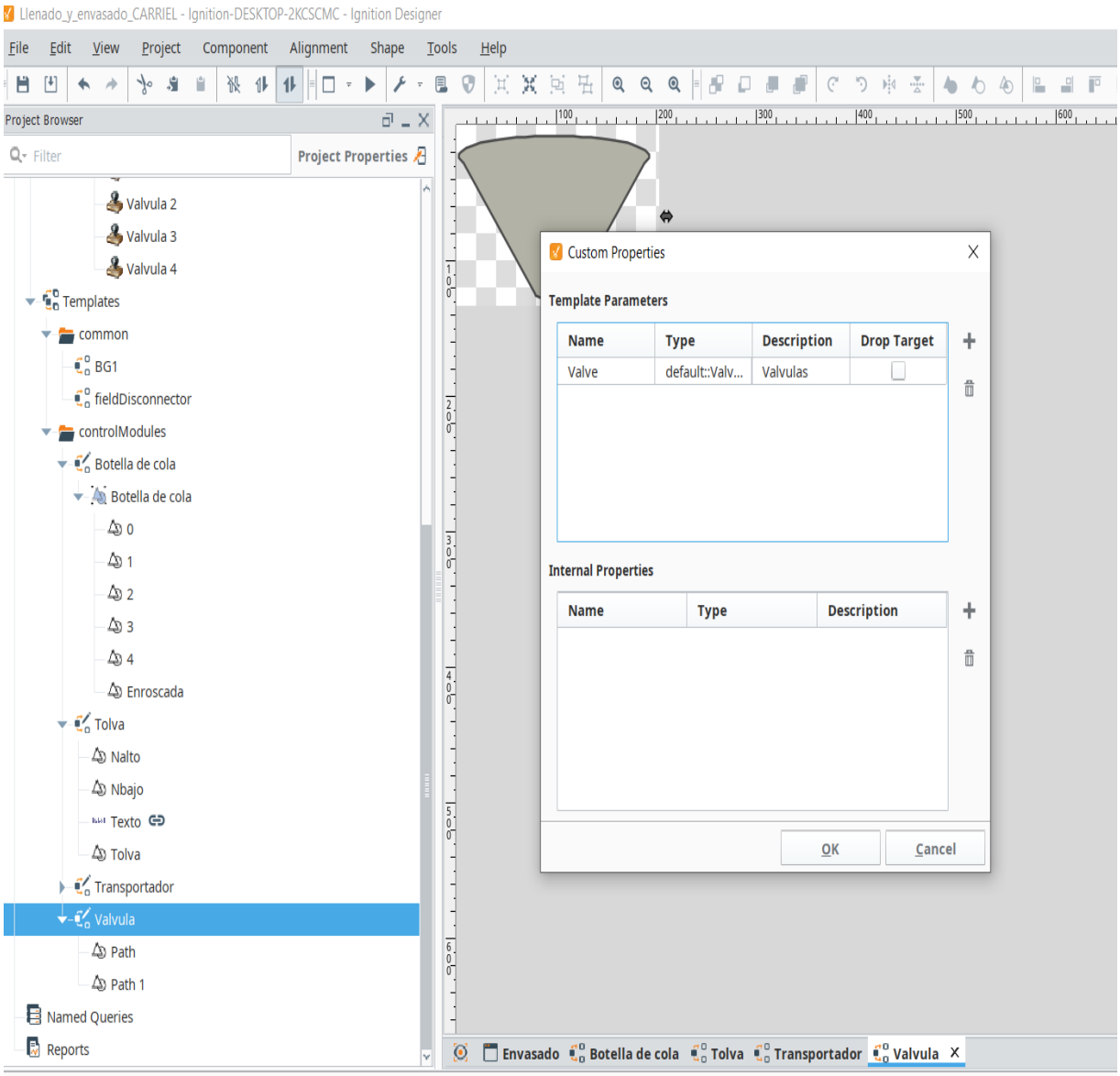
Figura 30 Tags de las cuatro Válvulas del diseño



Fuente: El Autor

Volviendo al template de la válvula, haciendo clic derecho en él y seleccionando “Customize Properties”, así mismo como se hizo con la cinta transportadora se puede configurar los parámetros del template “Válvula” para que estas funcionen de acuerdo con el grupo de estancia “Válvulas” como se muestra en la siguiente figura.

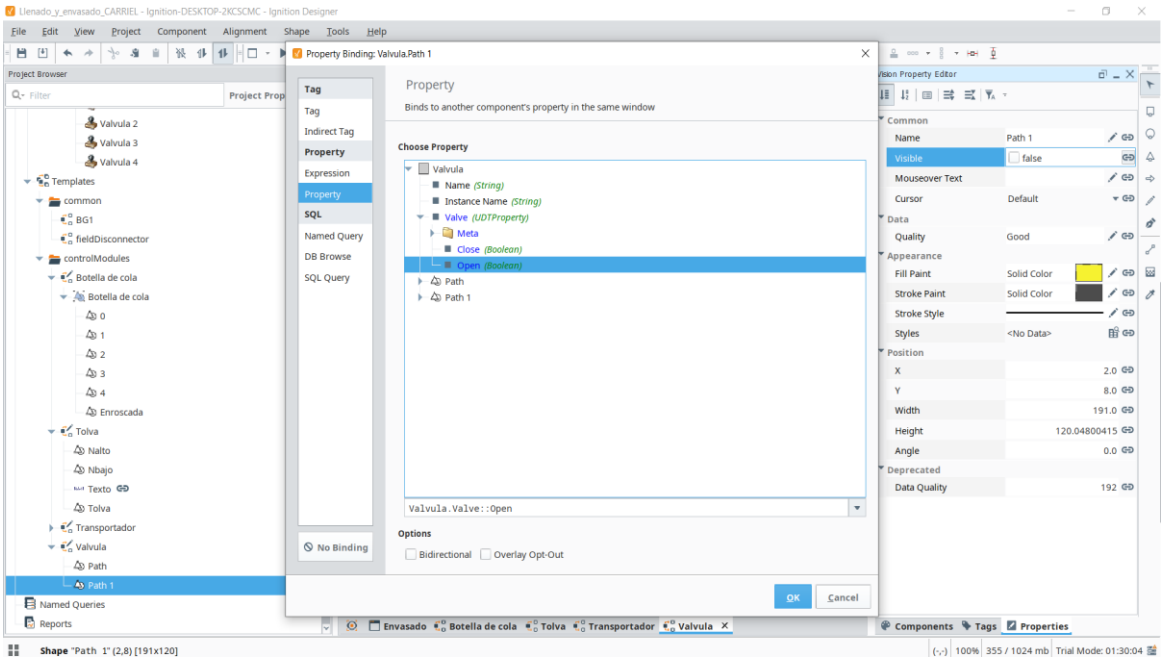
Figura 31 Parámetros configurados del template "Válvula"



Fuente: El Autor

Cabe recordar que el template de la válvula cuenta con dos figuras ("Path y Path 1"), como se muestra en la figura 31, Path es de color gris para simular que está cerrado mientras que Path 1 es de color amarillo para simular que este se encuentra abierto. Así mismo estas figuras se las vincula (binding) respectivamente con los tags "Open" y "Close" en su visibilidad.

Figura 32 Vinculación de las partes del template "Válvula" a sus respectivos tags

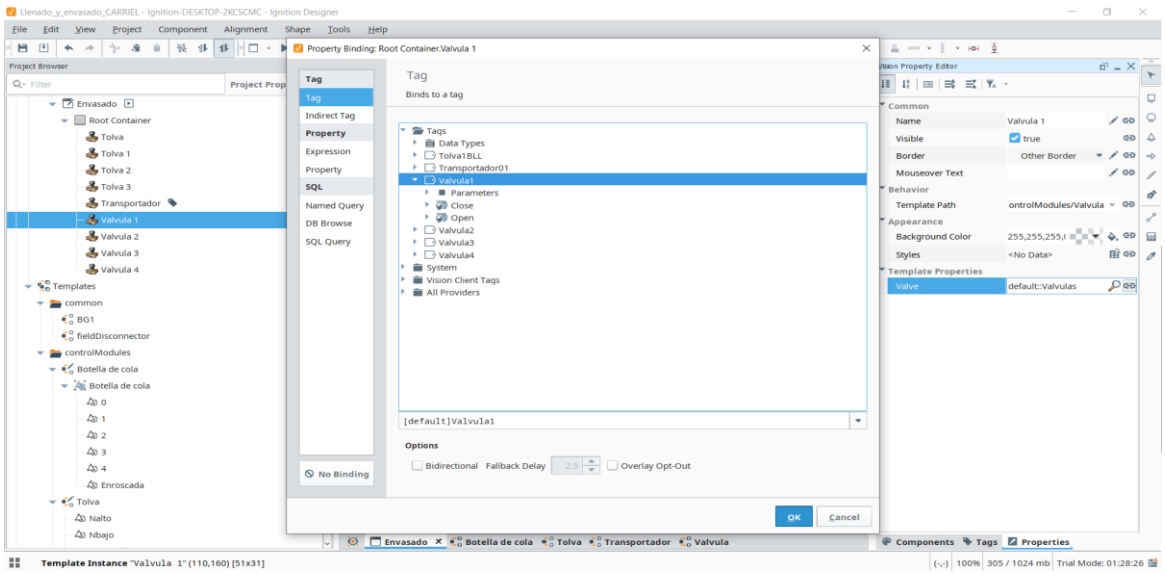


Fuente: El Autor

Ahora que el template de la válvula está configurado correctamente con sus parámetros y tags respectivos, volviendo al diseño principal, a cada válvula se le asigna un número y sus respectivos tags de izquierda a derecha (1 al 4) y hacemos sus respectivas vinculaciones a los tags creados anteriormente (revisar figura 30). De esa manera el usuario tendrá control respectivo de cada una de la válvula que permitirán o no el flujo de la fórmula de la bebida.

Para hacer el binding directamente del componente individual en el diseño principal se lo selecciona, se dirige a la pestaña de Properties y en "Template Properties" se vincula el componente no a los tags individuales sino al grupo que contiene los tags de abrir y cerrado. Tal como se muestra en la siguiente figura:

Figura 33 Binding de cada válvula a sus respectivos grupos de tags



Fuente: El autor

Prosiguiendo con el diseño, ahora el enfoque se dirige hacia las tolvas que almacenaran la fórmula para ser distribuida a las botellas. Estas tolvas cuentan con 3 partes principales: Un sensor de nivel alto de líquido, el cuerpo en si y el sensor de nivel bajo. Esta también cuenta con un cuadro de texto para identificar la tolva, no es necesario es más una cuestión de estética.

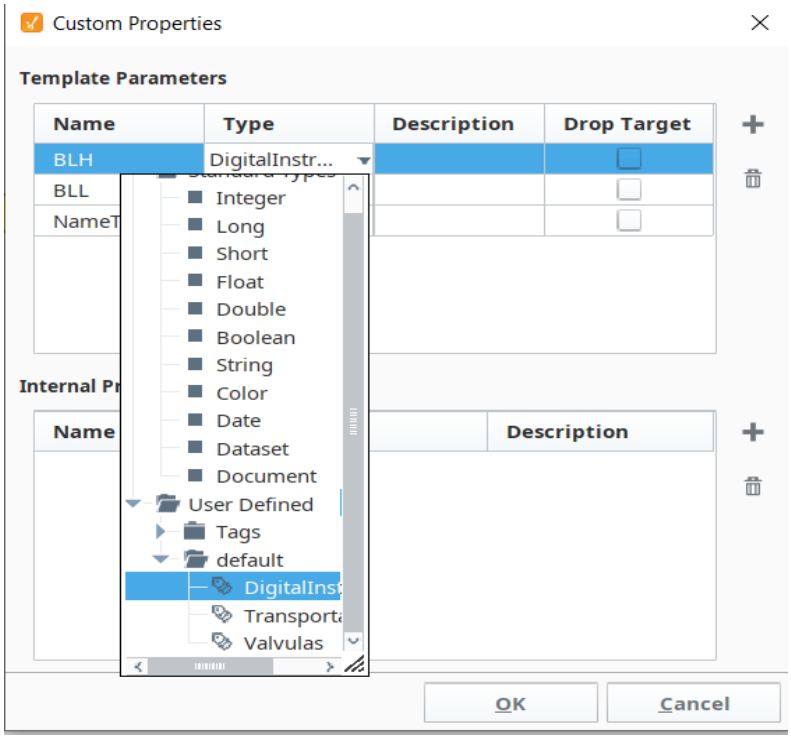
Antes de empezar con los cambios necesarios en sus parámetros, es importante crear un grupo de estancia para estos sensores. El grupo tendrá como nombre “DigitalInstrument” y tendrá un simple tag booleano “LevelSensor”.

Figura 34 Grupo de estancia “DigitalInstrument”

Fuente: El autor

Así mismo como con el template de la válvula, vamos a “Customize Properties” y añadimos 3 parámetros las cuales son: BLL (Nivel bajo de líquido), BLH (Nivel alto de líquido) y NameTolva (Etiqueta de la tolva). El NameTolva será un tipo de dato String, es decir que se podrá escribir cualquier texto que el usuario desee, en este caso lo usare para identificar cada tolva. El BLL y BLH será del mismo tipo de dato creado en el grupo de instancia “DigitalInstrument” como se puede revisar en la siguiente figura:

Figura 35 Customización del tipo de dato del parámetro BLH



Fuente: El autor

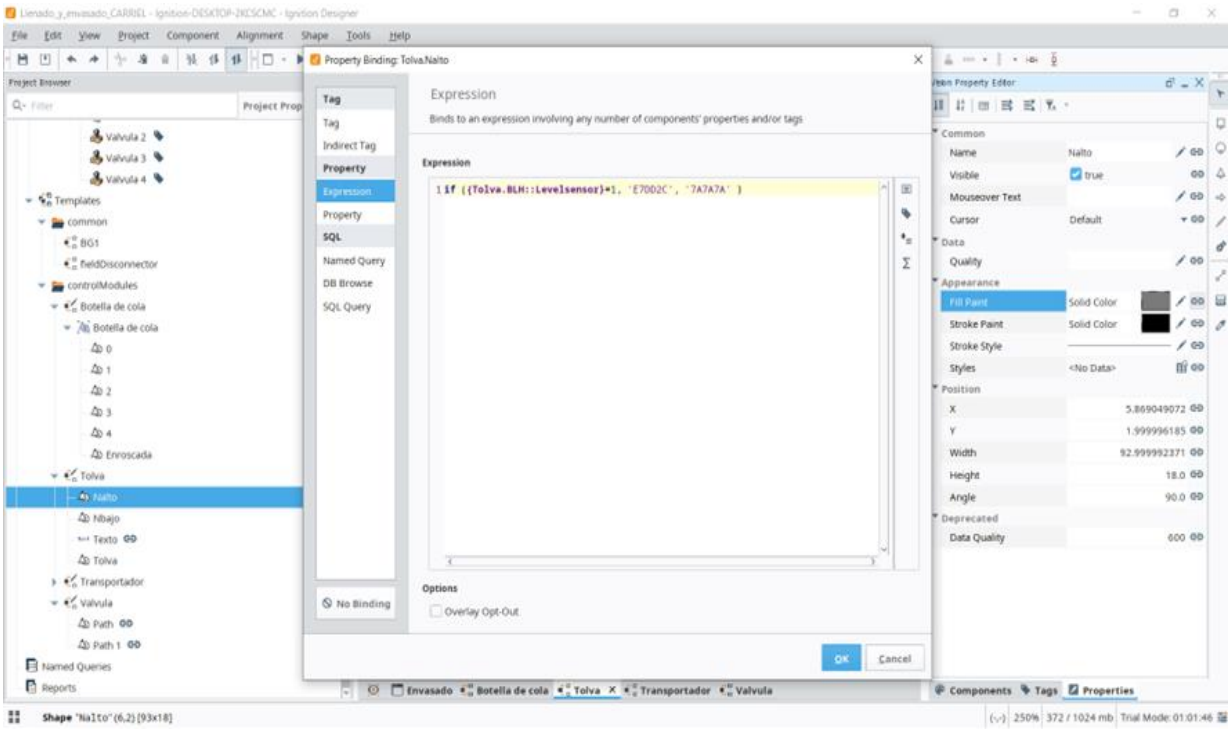
Una vez creados los parámetros, hacemos unas vinculaciones o binding a cada parte de la tolva, el más sencillo de vincular es el cuadro de texto la cual como es un tipo de data String, se vincula el texto al parámetro “NameTolva” esto hace que cualquier texto que se escriba será visible en el diseño final.

Los niveles altos y bajos de la tolva tienen un distinto proceso de vinculación. Estas en vez de simplemente vincularlas con un tag o un parámetro, se vinculaba sus apariencias, en particular su color de fondo, escribiendo una línea de código en Property-Expression. Estas líneas son:

- **Para Nalto/BLH:** if ({Tolva.BLH::Levelsensor}=1, 'E7DD2C', '7A7A7A')
- **Para Nbajo/BLL:** if ({Tolva.BLL::Levelsensor}=1, '7A7A7A', 'E7DD2C')

La idea detrás de esto es que cuando la tolva tuviese un nivel alto de líquido, el sensor se activaría cambiando de color de gris (apagado/7A7A7A) a amarillo (activado/E7DD2C) y viceversa para cuando tuviese un nivel bajo de líquido. Esto se aprecia mejor en la figura 36.

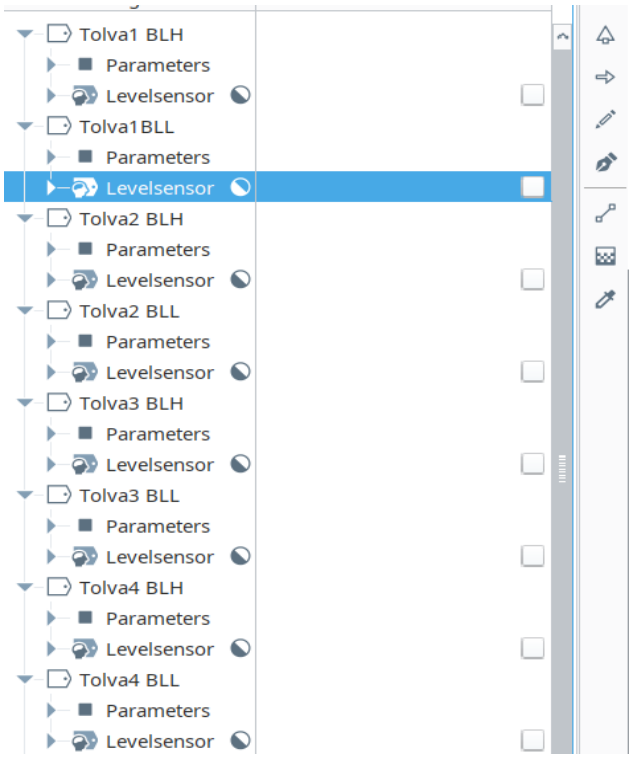
Figura 36 Vinculación de la apariencia del sensor alto de la tolva



Fuente: El autor

Ahora si configurado y vinculado sus parámetros y partes de la tolva, volviendo al diseño principal, creamos varios grupos de tags correspondientes a cada sensor tanto alto como bajo de cada tolva. Esto se presencia mejor en la siguiente figura.

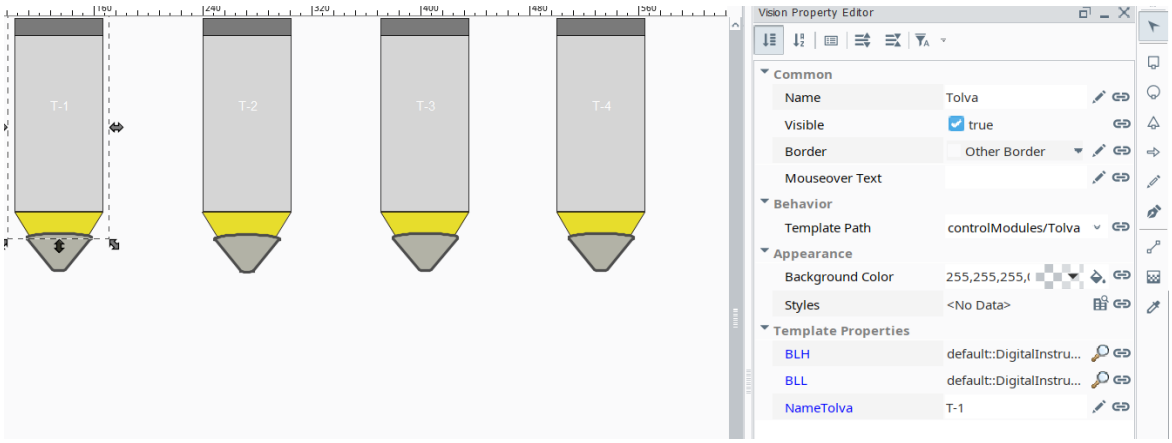
Figura 37 Grupos de instancias creados para los sensores de cada tolva usando el tag LevelSensor



Fuente: El autor

Se selecciona la tolva del diseño principal y se nota en la ventana de propiedades “Template Properties” y se encuentran enlistados los parámetros que se configuraron anteriormente. En cada parámetro lo vinculamos con su respectivo tag y grupo a excepción del NameTolva que al ser un dato String simplemente se escribe en ese cuadro de texto la etiqueta de la tolva en este caso T-1 hasta T-4. Así sucesivamente hasta vincular todos los sensores de cada tolva.

Figura 38 Template Properties de las Tolva del diseño

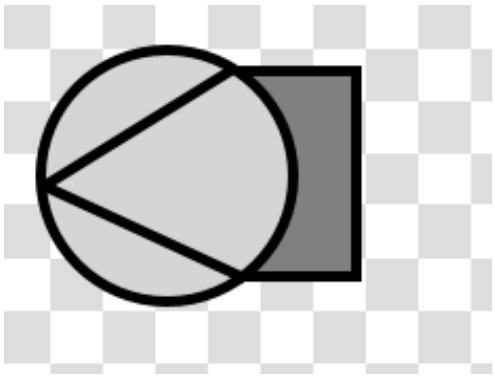


Fuente: El autor

Uno de los últimos componentes cruciales para nuestro diseño es una bomba con un sistema de tuberías que llevara el líquido de las tolvas hacia las botellas de la cinta transportadora.

Para realizar la bomba se podría hacer lo mismo que con la tolva, botellas o válvula, usar figuras simples, pero al final se importó de unos archivos la bomba ya armada para ahorrar tiempo. Esta trajo consigo algunos componentes ya vinculados que se tuvo que eliminar y modificar acorde a nuestro diseño. La bomba es tal como se muestra en la siguiente figura:

Figura 39 Bomba importada al diseño SCADA



Fuente: El autor

Una vez importada la bomba, vino con dos partes principales cuyos nombres son: “mcFixMa02” y “mcHqHwMa01” la cual juntaremos para que se complementen y formen un solo componente.

Haciendo el mismo proceso que con las demás figuras, configuramos sus parámetros a datos “Integer” o enteros a ambos templates, siendo para “mcFixMa02” el parámetro de “Bomba” (Revisar la figura 40) y para “mcHqHwMa01” el parámetro “ST” (Revisar figura 41).

Figura 40 Parámetro Bomba para mcFixMa02

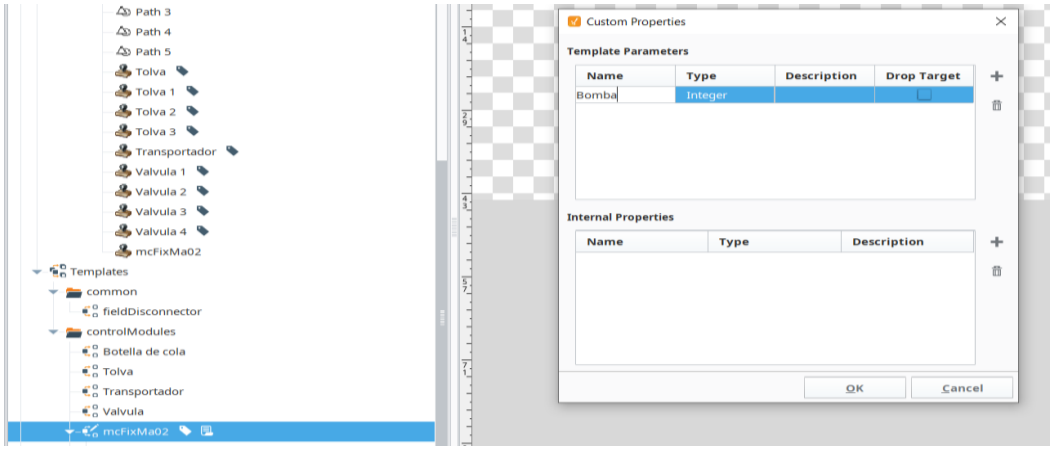
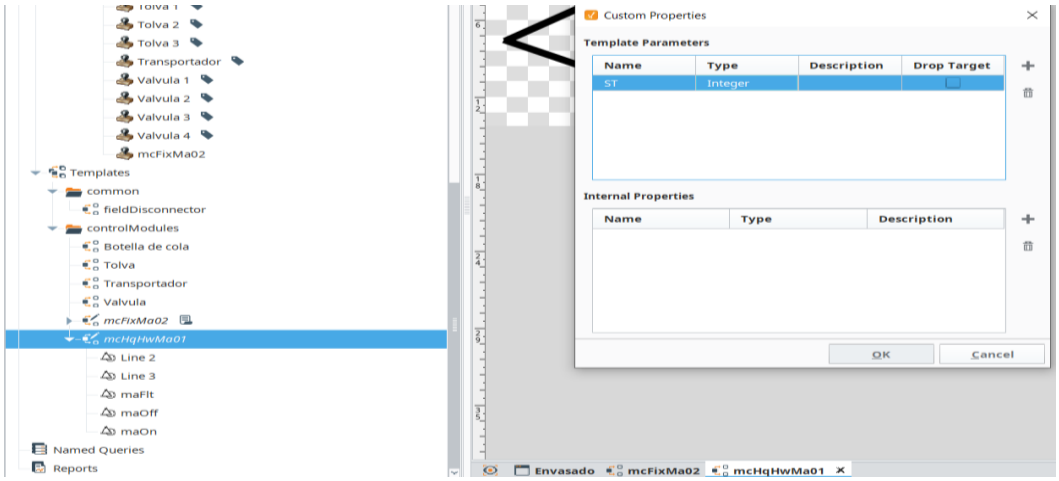


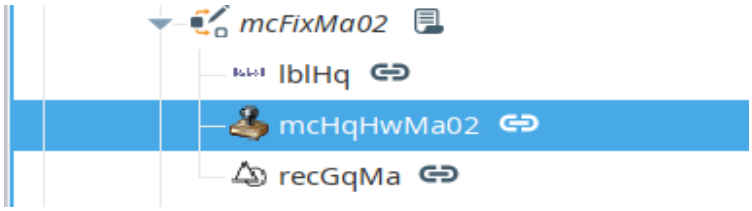
Figura 41 Parámetro ST para mcHqHwMa01



Fuente: El autor

Cabe aclarar que “mcHqHwMa01” se lo traspasa al template “mcFixMa02” y este queda como un componente de ese template que no se puede modificar ahí sino en su template original y este queda con el nombre “mcHqHwMa02”.

Figura 42 Traspaso del componente de la señal de bomba al template completo de la bomba



Fuente: El autor

Volviendo al template “mcHqHwMa01”, este cuenta con 3 partes: “maFlt, maOff y maOn” cada una son de diferente color en apariencia para simbolizar cuando la bomba esta activada, desactivada o cuando esta presenta un error.

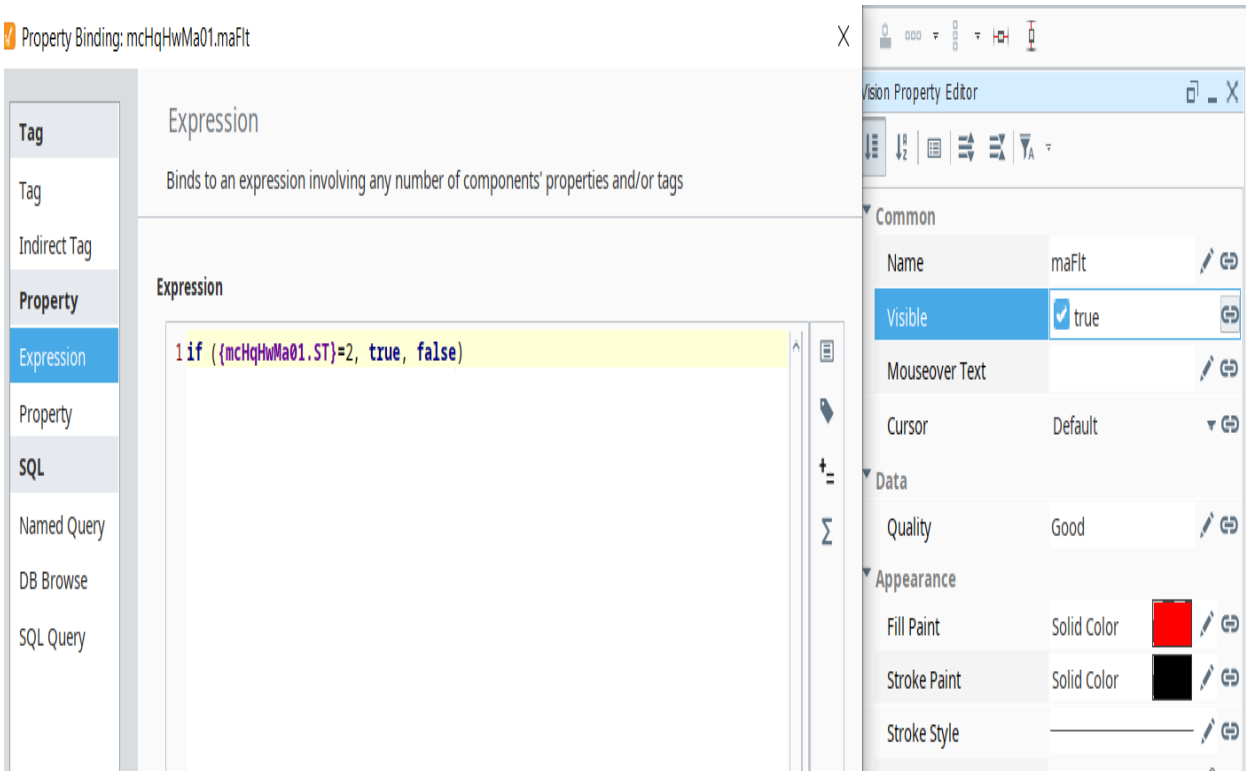
A cada una de estas partes se le hace un binding en cuanto a su visibilidad en el diseño. Tomando en cuenta que el template tiene como parámetro un valor ST de tipo de data entero (int), se tiene que hacer binding en su Property-expression y escribir los siguientes códigos:

- **Para maFlt:** if ({mcHqHwMa01.ST}=2, true, false)
- **Para maOff:** if ({mcHqHwMa01.ST}=0, true, false)
- **Para maOn:** if ({mcHqHwMa01.ST}=1, true, false)

La idea de esto es que cuando el valor entero del tag “ST” sea 2, la señal de error se activará y la bomba se pondrá roja deteniendo el proceso. Cuando sea 1, la señal estará verde simbolizando el correcto funcionamiento de la bomba y cuando este en 0, se pondrá en gris lo que significa que esta apagado. Una

vista más clara con respeto al binding se la puede presenciar en la siguiente figura:

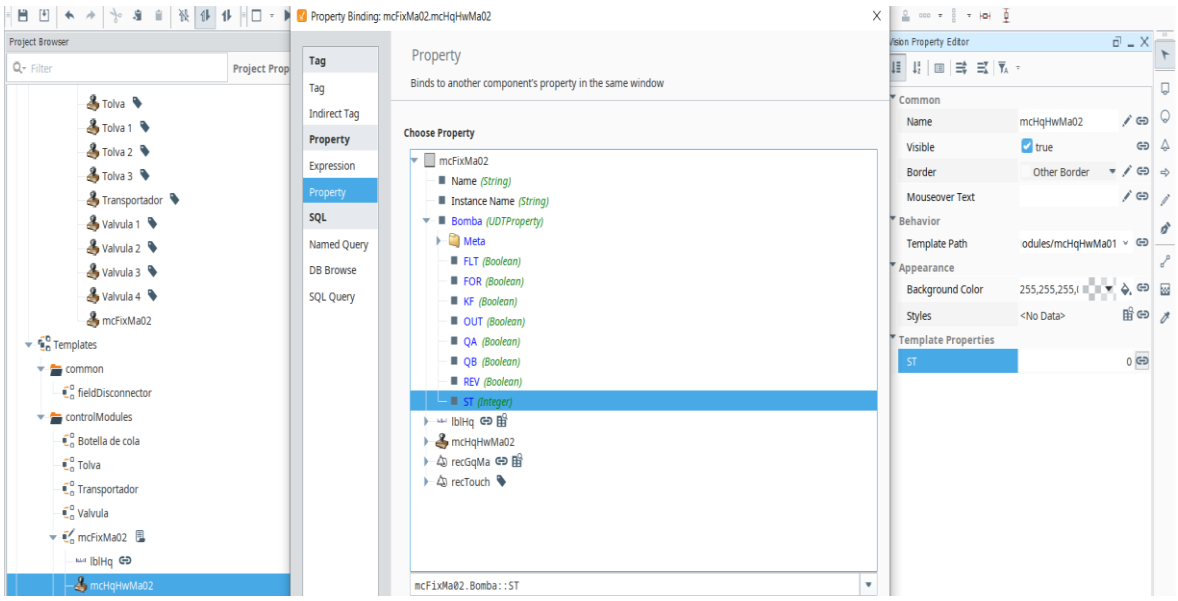
Figura 43 Property-expression binding de la señal de la bomba (maFlt)



Fuente: El autor

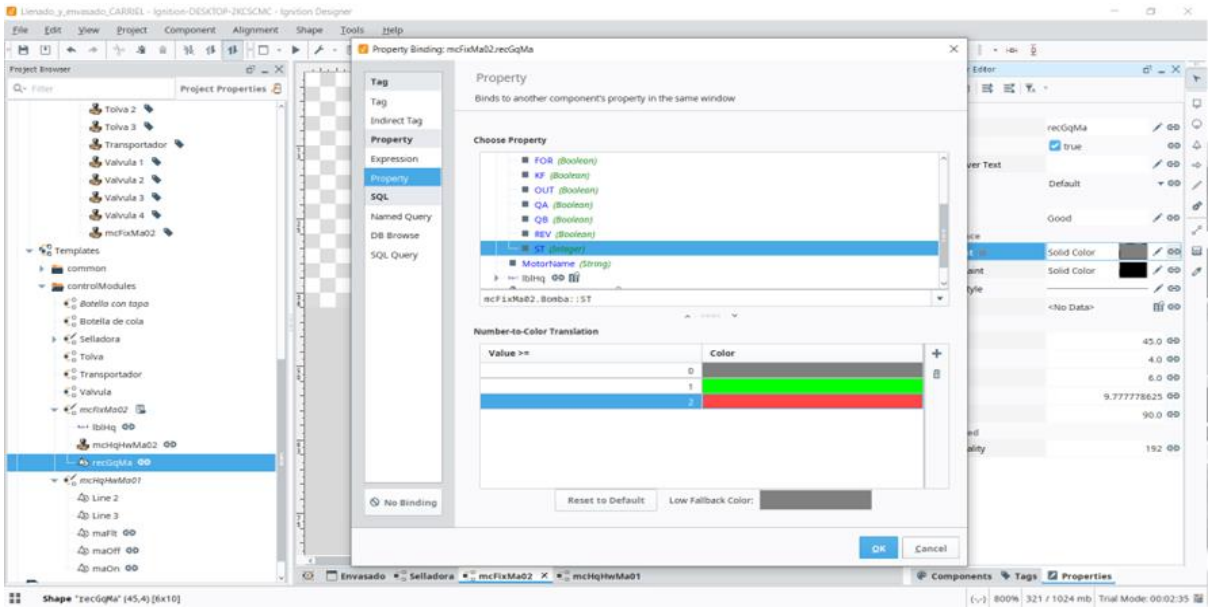
Una vez hechos los vínculos, creamos un grupo nuevo de instancia en la pestaña tags con los mismos tags que el del transportador ya que usan lo mismo que sería un motor. A este grupo lo llamaremos “Bomba” y finalmente vinculamos el componente de la bomba “mcHqHwMa02” al tag ST (que es un tipo de data entera o int) y vinculamos una parte de la bomba llamada “recGqMA” también al tag ST que simboliza la abertura de la bomba y el fluido de los líquidos hacia las botellas con presión. Tal cuales se presentan en las siguientes figuras:

Figura 44 Vinculación de las señales de la bomba al tag ST



Fuente: El autor

Figura 45 Vinculación de la abertura de la bomba al tag ST

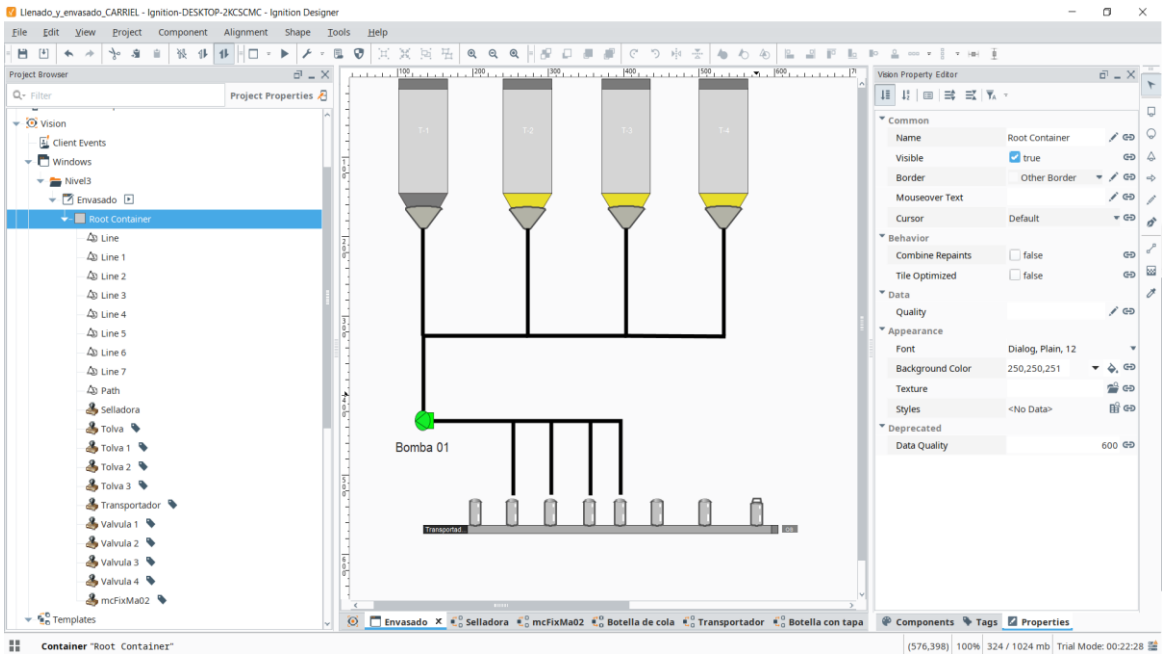


Fuente: El autor

Activamos los tags en el diseño principal de la bomba para comprobar si funciona y nos da como resultado que cuando se prende se pone tanto la bomba

como su señal en verde, cuando se desactivan estos se ponen en gris y si se presenta un error, la bomba con su señal se pondrán rojos.

Figura 46 Bomba del diseño activado



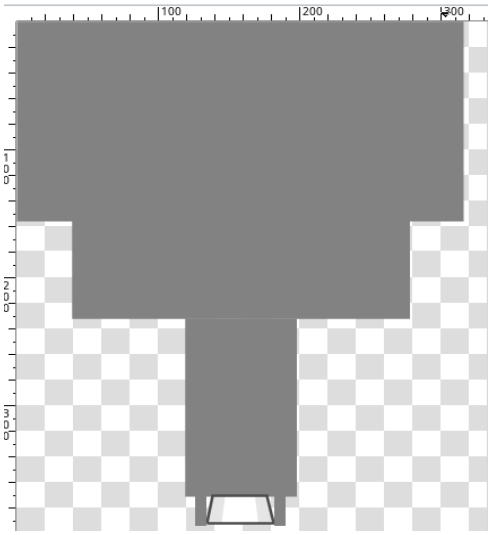
Fuente: El autor

Nota: *Para hacer las "tuberías" se usó la herramienta de pluma que se encuentra en el costado derecho del programa y se customizo sus colores como quise.*

Por último, en el diseño se requiere una maquina selladora que les pondrá las tapas a las botellas. Se crea un nuevo template con el nombre "Selladora" y usando figuras simples se le da forma a la máquina.

Para la tapa de la botella simplemente se saco de una botella de plástico de la biblioteca de símbolos de Ignition descartando el cuerpo y solo dejando la tapa para ajustarla como uno desee la cual da como resultado la siguiente figura:

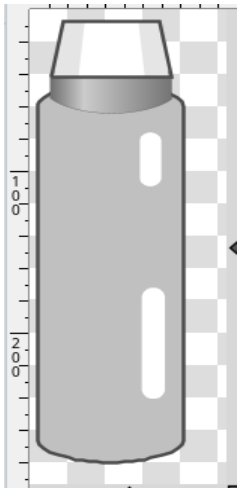
Figura 47 Template de la selladora de botellas



Fuente: El autor

Antes de terminar con el diseño, se hace una pequeña mejora estética a las botellas de plástico y creamos un template de la botella de plástico con su tapa una vez sellada.

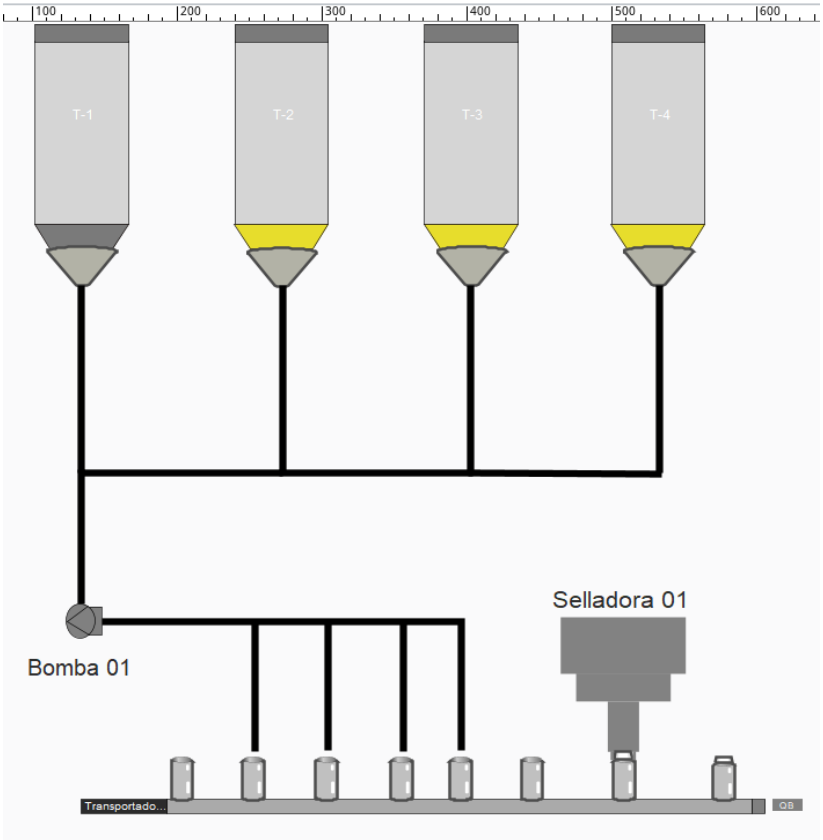
Figura 48 Template de la botella de plástico con su tapa



Fuente: El autor

Ahora que se tiene todos los componentes, se los pasa al diseño principal y nos da como resultado lo que se muestra a continuación:

Figura 49 Diseño base del llenado y envasado de botellas de cola



Fuente: El autor

Con esto se podría dar por terminado el diseño base del sistema SCADA de llenado y envasado de botellas de cola.

3.1.3 Ajustes y últimos detalles previos a la codificación

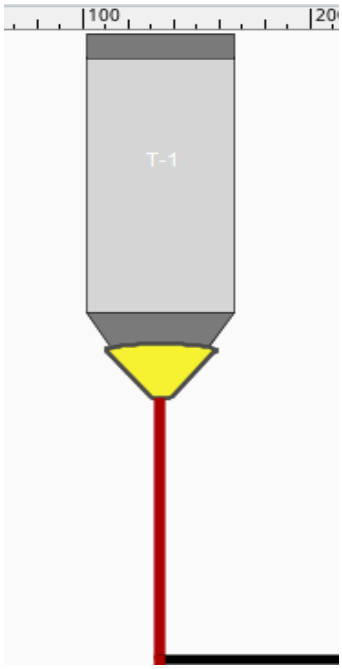
A lo que se diseñó el sistema de tuberías usando la herramienta de pluma, estas se volvieron partes del diseño y con todo el sentido ya que estas transportan el líquido desde las tolvas hasta las botellas de la cinta trasportadora.

La idea con estos tubos es que estos se activen cambiando de color al mismo que del líquido de la cola para simbolizar que está fluyendo el líquido por este sistema de tuberías. Este sistema estará dividido en dos secciones: Las

tuberías conectadas desde las válvulas de cada tolva hacia a la bomba y las tuberías conectadas desde la bomba hacia las botellas (Revisar figura 49).

La idea es que la primera sección de tuberías tendrá un binding con sus respectivas válvulas y que cuando el tag del abierto de cualquier válvula (Open) se active, estas tuberías cambien de color al mismo del líquido de las colas (AC0000) para simbolizar su flujo como se muestra en la siguiente figura:

Figura 50 Tubería conectada a la válvula de la Tolva1 activada



Fuente: El autor

Para lograr esto es necesario hacerles binding en la apariencia de cada una de las líneas de la primera sección. No tenemos que hacerles binding directo a un tag creado, sino escribir los siguientes códigos condicionales en las propiedades de expresión para hacer los bindings:

Primero con los tubos verticales tenemos:

- **Tubo de Válvula1:** if ({[default]Valvula1/Open}=1, 'AC0000', '000000')
- **Tubo de Válvula2:** if ({[default]Valvula2/Open}=1, 'AC0000', '000000')
- **Tubo de Válvula3:** if ({[default]Valvula3/Open}=1, 'AC0000', '000000')
- **Tubo de Válvula4:** if ({[default]Valvula4/Open}=1, 'AC0000', '000000')

Con los tubos horizontales el código es distinto ya que unos tubos son un poco más complejos que otros, por ende, el código usara operadores lógicos (En este caso **OR "| |"**) para cumplir su propósito de transportar el fluido. De izquierda a derecha los códigos son los siguientes:

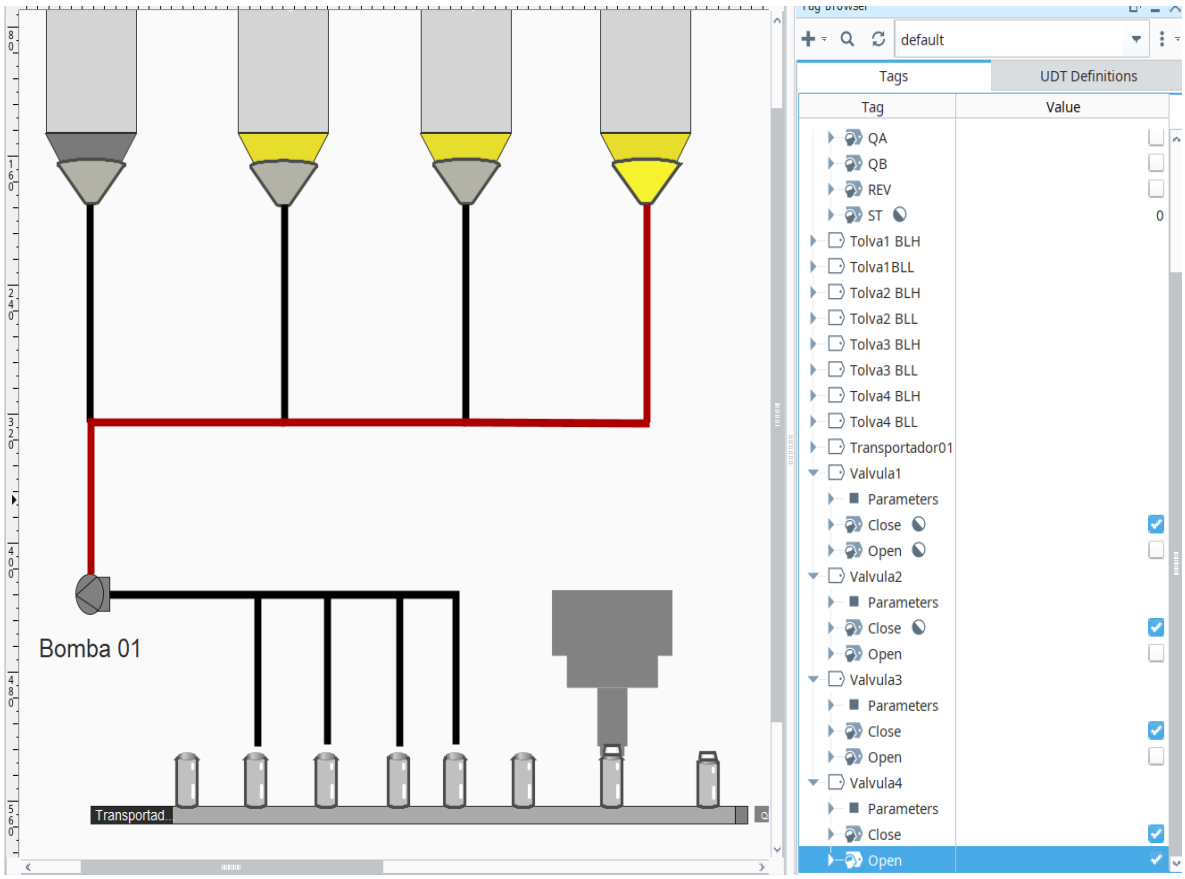
- **Tubo de la izquierda:** if ({[default]Valvula4/Open}=1 || {[default]Valvula3/Open}=1 || {[default]Valvula2/Open}=1, 'AC0000', '000000')
- **Tubo del medio:** if ({[default]Valvula4/Open}=1 || {[default]Valvula3/Open}=1, 'AC0000', '000000')
- **Tubo de la derecha:** if ({[default]Valvula4/Open}=1, 'AC0000', '000000')

Por último, tenemos un tubo que conecta directamente con la bomba y el resto de los tubos de la primera sección, este tendrá el siguiente código:

if ({[default]Valvula4/Open}=1 || {[default]Valvula3/Open}=1 || {[default]Valvula2/Open}=1 || {[default]Valvula1/Open}=1, 'AC0000', '000000')

Una vez configurados lo ponemos a prueba y nos da como resultado lo siguiente cuando abrimos el paso de llave de la válvula 4:

Figura 51 Demostración del flujo de líquido desde la Tolva 4 hacia la bomba



Fuente: El autor

Como quedo demostrado, si se activa el tag Open de la Válvula4, el sistema de tuberías reaccionara con respeto a cuál válvula este abierta en ese momento por el usuario.

Ahora para el segundo sector es mas sencillo ya que este trabajara en conjunto. La idea es la misma que la del primer sector que cuando la válvula se abra los tubos cambien de color para simular el flujo de líquido, sin embargo, hay que añadirle una condicional extra usando el ordenador lógico **AND (&&)** de cuando la bomba se prenda, es decir cuando el tag ST de la bomba sea 1, esta se prende y pasa liquido por esos tubos a presión.

Para lograr eso escribimos el siguiente código en su binding:

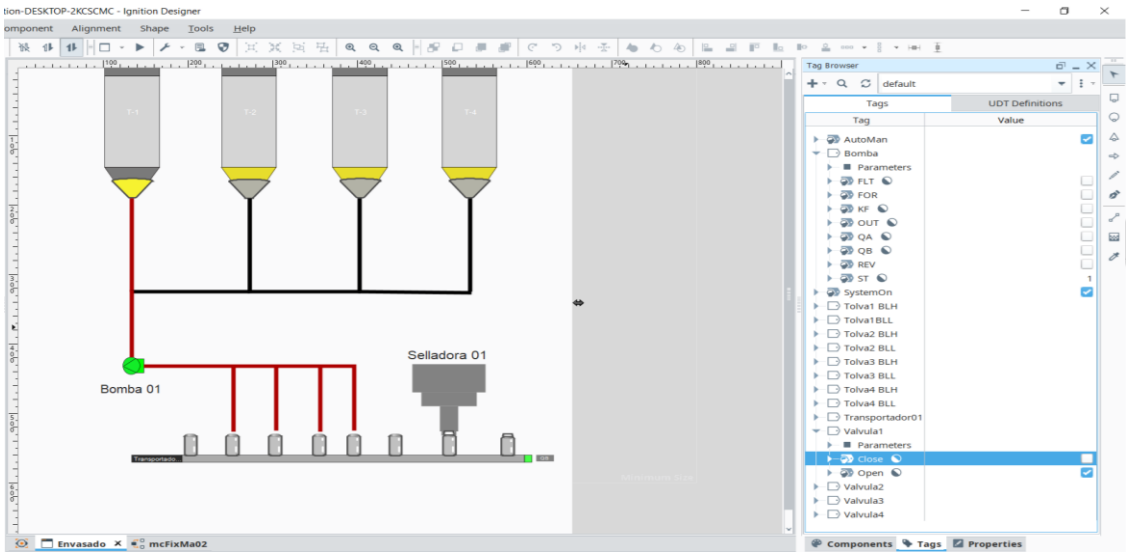
if ({[default]Bomba/ST}=1 && ({[default]Valvula1/Open}=1 ||
{[default]Valvula2/Open}=1 || {[default]Valvula3/Open}=1 ||
{[default]Valvula4/Open}=1), 'AC0000', '000000')

Figura 52 Binding de expression a la segunda sección de tuberias

Fuente: El autor

Poniéndolo a prueba tenemos como resultado que cuando el tag ST de la bomba sea 1 y cualquier válvula en este caso T-1 este activada, las tuberías funcionan correctamente como su código especifica.

Figura 53 Prueba de funcionamiento de las válvulas de la segunda sección

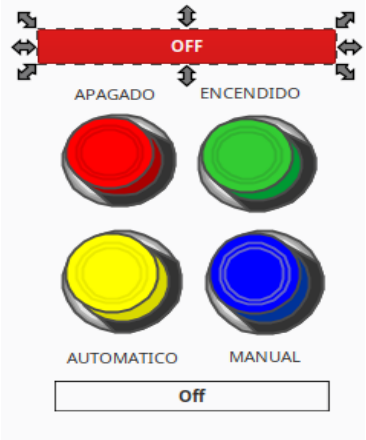


Fuente: El autor

Expandimos nuestro diseño para añadir 6 cosas extras. No son componentes importantes como tal para el sistema, pero servirán como controladores e indicadores del sistema en sí.

Añadimos al costado derecho del sistema: 4 botones booleanos, 1 indicador de doble sentido y 1 indicador de estado múltiple tal como se muestra en la siguiente figura:

Figura 54 Botones e indicadores complementarios del diseño

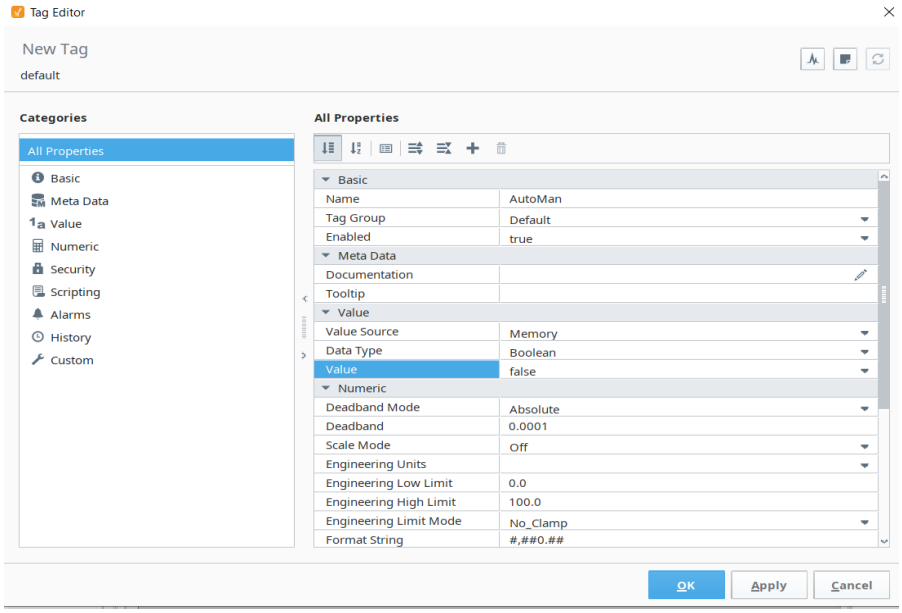


Fuente: El autor

La idea es que los botones influyan en el estado del sistema en general, como su encendido (verde), su apagado (rojo), si el sistema está en automático (amarillo) y/o manual (azul). El indicador superior avisara al usuario si el sistema este encendido o apagado mientras que el inferior indicara si esta manual o automático.

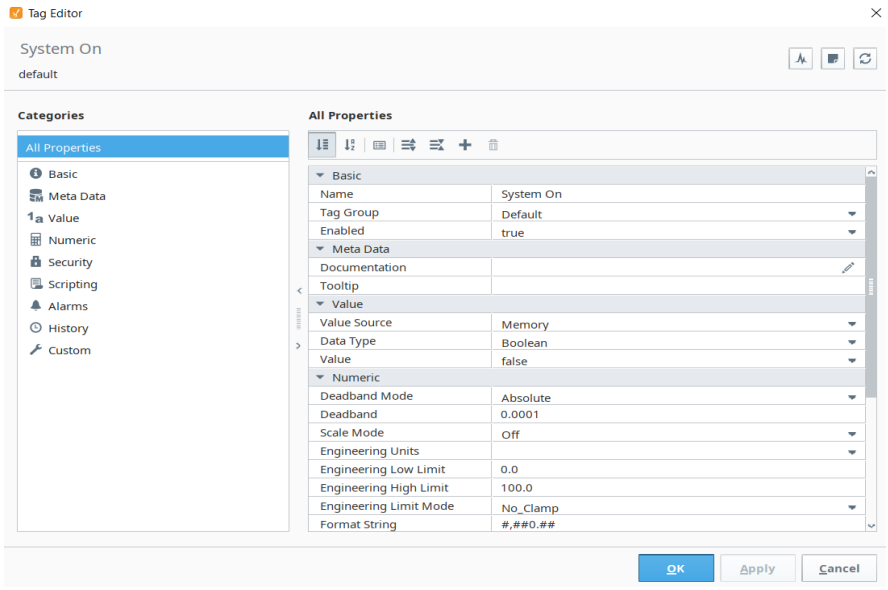
Para darle funcionamiento a estos botones se tienen que crean nuevos tags de tipo de dato booleano llamados “System On” y “AutoMan” con su valor en false por predeterminado como muestran las siguientes figuras.

Figura 55 Creación del tag "AutoMan"



Fuente: El autor

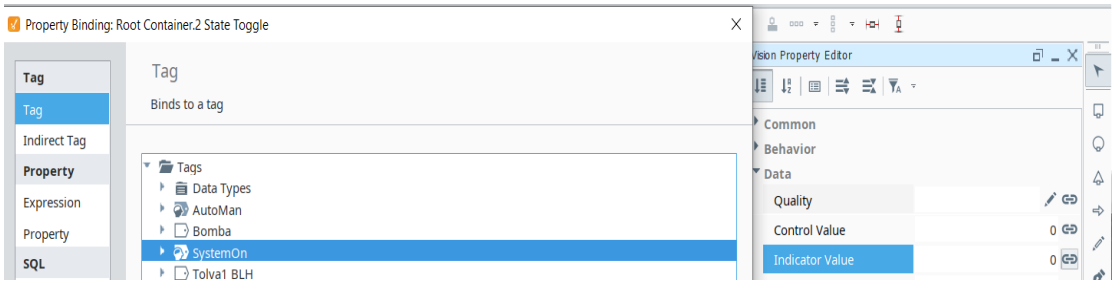
Figura 56 Creación del tag "System On"



Fuente: El autor

Una vez creados los tags al indicador superior se vincula en sus propiedades "Data-Indicator Value" al tag System On. Esto se muestra mejor en la siguiente figura:

Figura 57 Vinculación del indicador al tag System On



Fuente: El autor

Una vez vinculamos editamos su apariencia en los estilos del indicador en las propiedades de apariencia para que en vez de solo decir “ON” y “OFF” este diga “SYSTEM ON” o “SYSTEM OFF”

Figura 58 Edición del estilo del indicador “SYSTEM ON/OFF”

Fuente: El autor

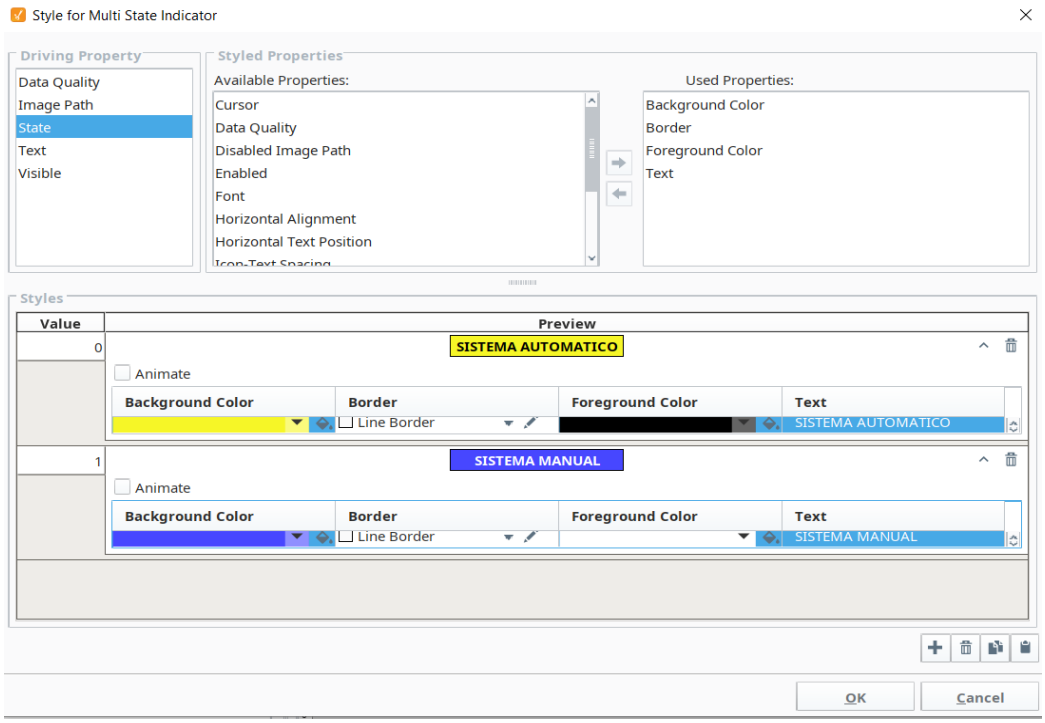
Para el indicador inferior del Sistema Automático/Manual, a este le vinculamos en su propiedad Data-State el tag AutoMan.

Figura 59 Vinculación del indicador Automático/Manual al tag AutoMan

Fuente: El autor

Como es un indicador de múltiples estados para cambiar su apariencia uno le hace doble clic a la figura para que se abra una ventana de estilos en donde se le podrá configurar a gusto sus estados y borrar los que no sirvan como se muestra en la figura:

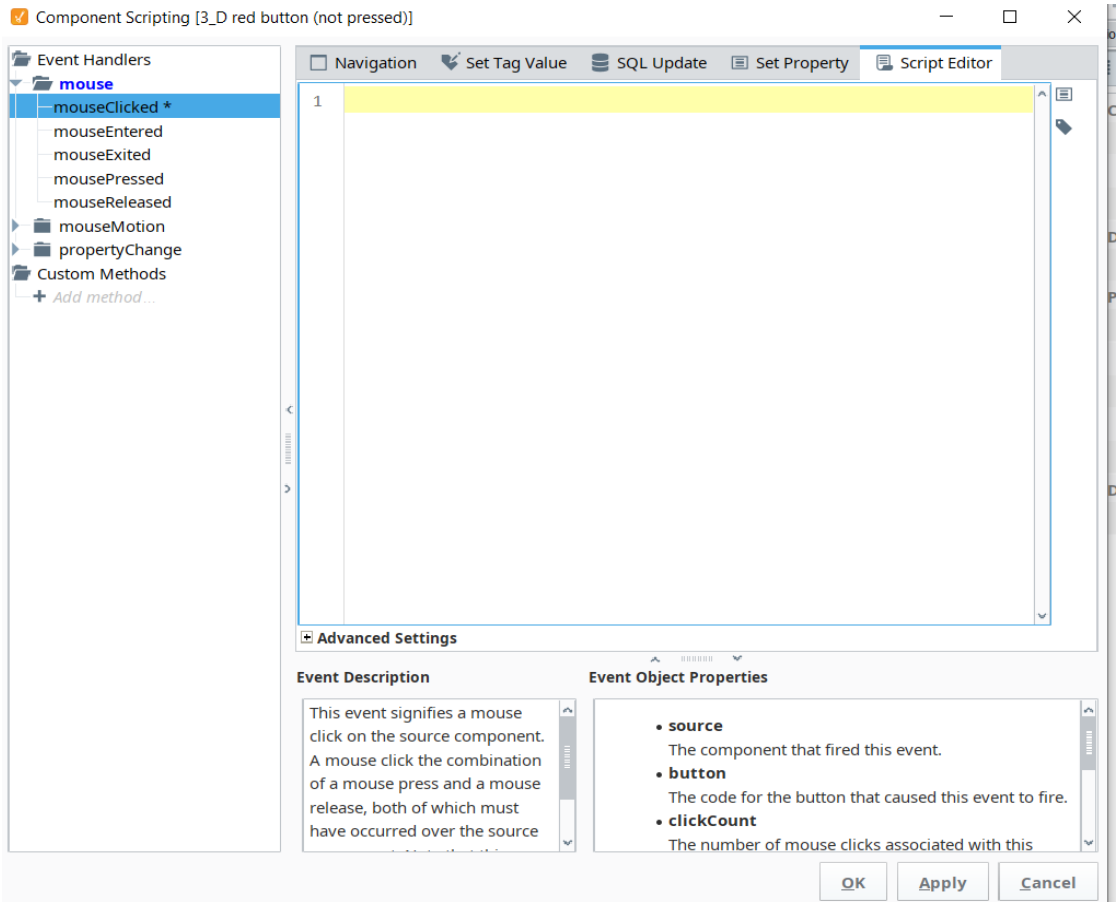
Figura 60 Editor de estilos del Indicador "Automático/Manual"



Fuente: El autor

Ahora pasando a los botones booleanos, para configurarlos rápidamente se selecciona cualquier botón, se presiona CTRL+J y se abrirá una ventana llamada "Component Scripting". Se selecciona en la carpeta de eventos "mouse/mouse clicked" (Esto aplica para todos los botones) se dirige al editor de Script y aparece lo siguiente:

Figura 61 Ventana "Component Scripting" abierto



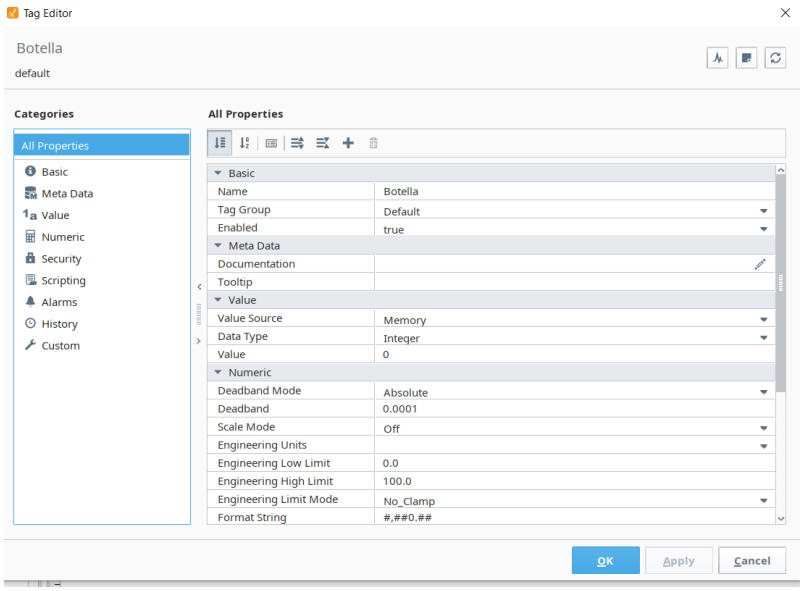
Fuente: El autor

En ese editor para cada botón se escribirán los siguientes códigos:

- **Botón Rojo:** system.tag.writeBlocking('[default]SystemOn', [0])
- **Botón Verde:** system.tag.writeBlocking('[default]SystemOn', [1])
- **Botón Amarillo:** system.tag.writeBlocking('[default]AutoMan', [0])
- **Botón Azul:** system.tag.writeBlocking('[default]AutoMan', [1])

Continuando, se les hace ajustes a las botellas de cola en la cinta transportadora. Para esto se crea un nuevo tag con el nombre "Botella" la cual tendrá un tipo de dato integrer (entero).

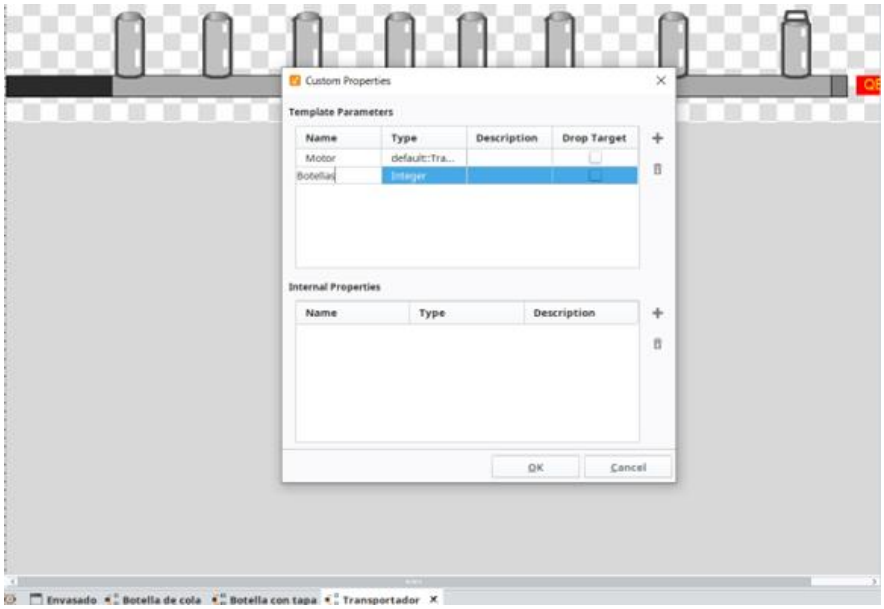
Figura 62 Creación del Tag “Botella”



Fuente: El autor

Una vez creado el tag, en el template de la cinta transportador se le añade un parámetro de Botella de tipo Integrer la cual eventualmente estará vinculada con el tag Botella.

Figura 63 Parámetro de botella (int) en el template de Transportador



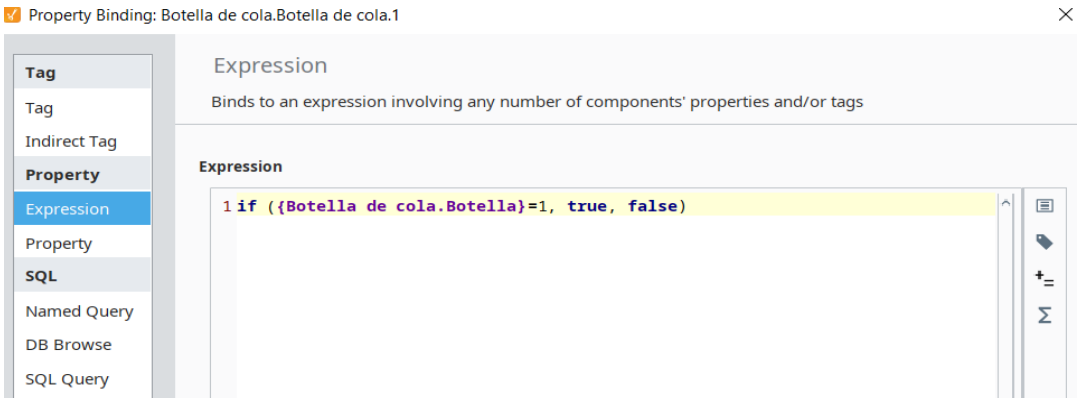
Fuente: El autor

Adicionalmente, en el template de botella de cola se incluye un parámetro de Botella (int) en sus propiedades. Esto con el fin de que según vaya cambiando de entero el tag, como del 1 al 4, estas botellas se irán llenando y pasando al siguiente estado.

Como se menciona anteriormente, el template de la botella cuenta con diferentes partes en su figura. Del 0 al 4 son los diferentes estados de llenado de la botella, siendo 0 que la botella está vacía y gradualmente se ira llenando hasta alcanzar 4 que es donde la botella se termina de llenar. Para conseguir lo dicho, se tiene que hacer binding a la visibilidad de cada una de las partes con el tag Botella, a diferencia de lo usual, este se le hace binding en Property-Expression y se escribe lo siguiente en cada parte de la figura que se requiera:

- **1:** if ({Botella de cola.Botella}=1, true, false)
- **2:** if ({Botella de cola.Botella}=2, true, false)
- **3:** if ({Botella de cola.Botella}=3, true, false)
- **4:** if ({Botella de cola.Botella}=4, true, false)

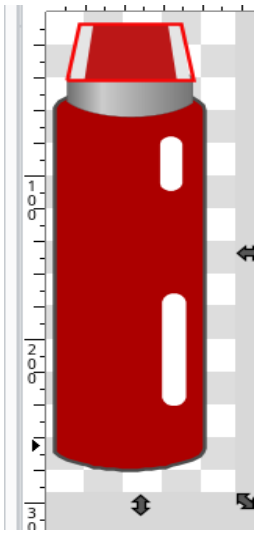
Figura 64 Property-Expression binding de la visibilidad de cada parte de Botella



Fuente: El autor

Se hace unos cambios visuales a la botella terminada, como añadirle color a su tapa y que siempre este llena ya que ese es su estado final tras llenarse y envasarse. Adicionalmente, se crea un template con la botella llena, pero sin la tapa.

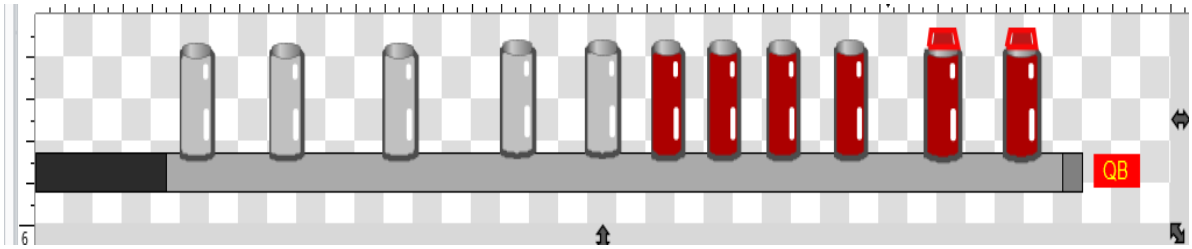
Figura 65 Botella terminada



Fuente: El autor

Volviendo a la cinta, en vez de pasar las botellas como si fuesen figuras, las pasamos al template como componentes. Lo que hace que actúen en base a los bindings hechos en sus propios parámetros sin alterar la cinta como tal. Las ubicamos a ojo y nos da como resultado lo siguiente:

Figura 66 Cinta transportadora con sus respectivas botellas corregidas



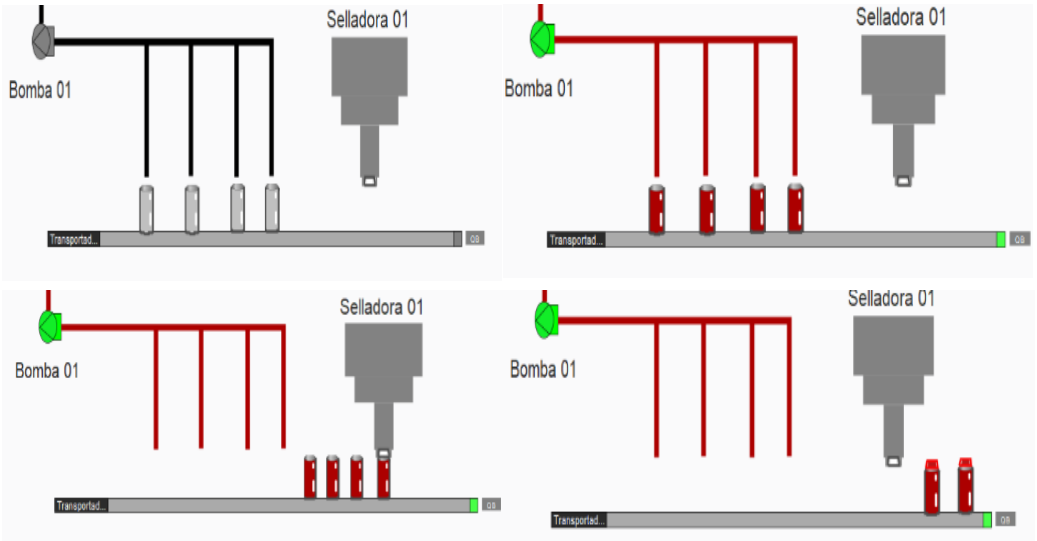
Fuente: El autor

En esa cinta tenemos 3 templates de botellas distintas, sin embargo, cada una va a actuar de acuerdo con el tag Botella siendo este entero. En cada uno de los templates se le hace un binding en cuanto a su visibilidad en el diseño. Los bindings se realizan en Property-Expression, las cuales son:

- **Botella de cola:** if ({Botella de cola.Botella}=5 || {Botella de cola.Botella}=6 , false, true)
- **Botella de cola llena:** if ({Botella de cola llena.Botella}=5, true, false)
- **Botella con tapa:** if ({Botella con tapa.Botella}=6, true, false)

La idea es que cuando el tag Botella marque los números 1,2,3,4 las primeras cuatro botellas se irán llenando respectivamente y en cuanto el contador este en 5 o 6 desaparecen para dar paso al siguiente estado de las botellas recorriendo la cinta para toparse con la maquina selladora (5) y al final salir de la cinta con la bebida llenada y envasada (6).

Figura 67 Estados del llenado y envasado de las botellas



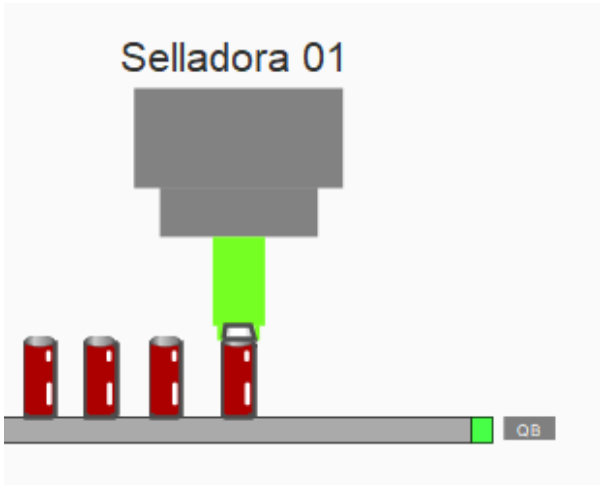
Fuente: El autor

Para concluir, tenemos la selladora la cual solo se activará en cuanto la botella pase por su área. Para esto simplemente vinculamos partes de su cuerpo para que cambien de color a verde al tag Botella, creando sus parámetros en el template y vinculando en su apariencia de color la siguiente expresión en las 3 partes inferiores de la selladora:

if ({Selladora.Sellado}=5, '76FF22', '828282')

Luego en el árbol de trabajo, vinculamos el componente de sellado y su parámetro “Sellado” al tag de Botella lo que nos da como resultado lo que se propuso:

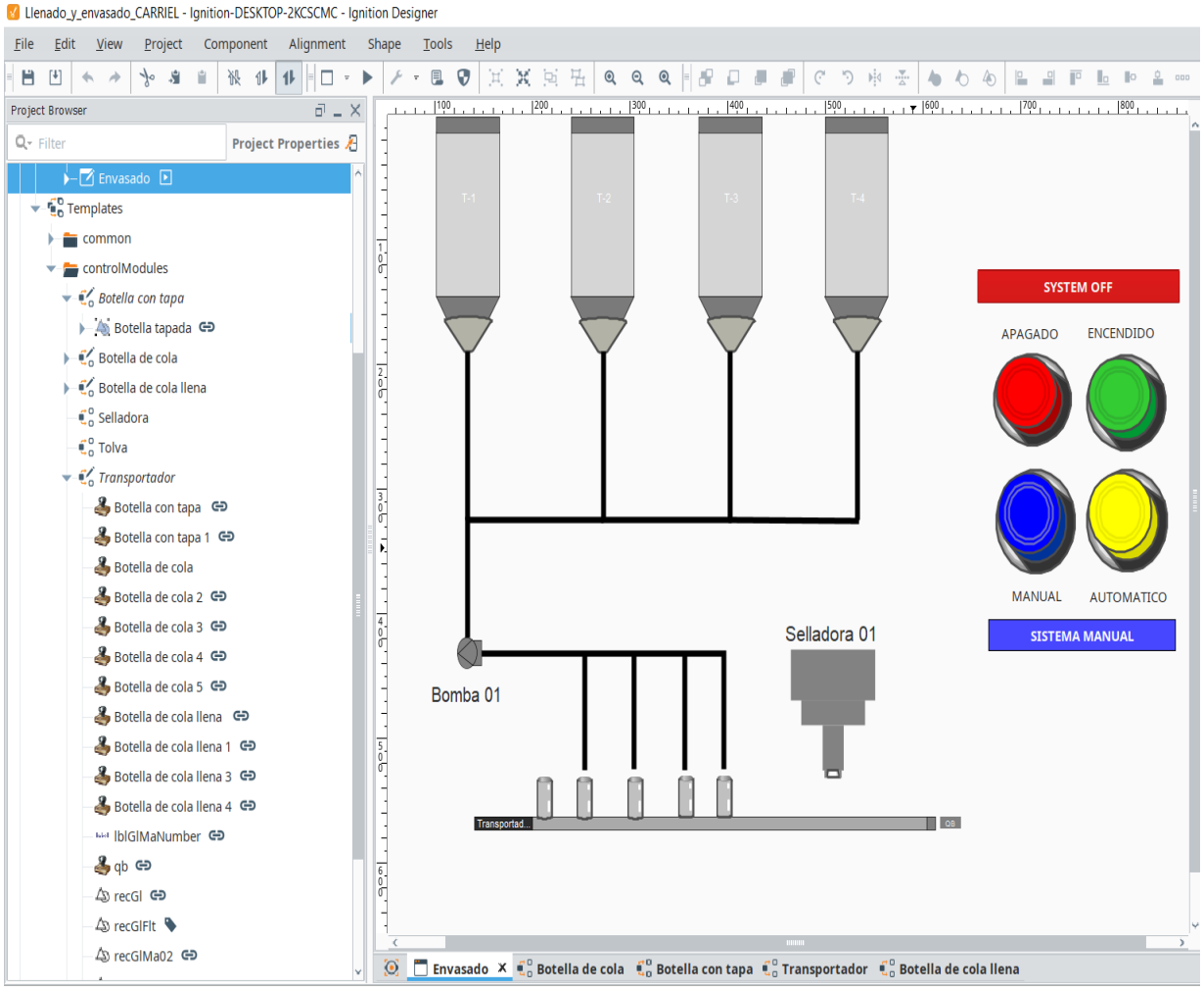
Figura 68 Encendido de la Selladora



Fuente: El autor

Ya con todos esos ajustes realizados, se puede dar por finalizado el apartado del diseño del sistema SCADA, sus tags y sus componentes principales. Por el momento, es un diseño relativamente básico, pero Ignition ofrece la posibilidad de mejorarlo por su flexibilidad. Nos da como resultado la siguiente imagen:

Figura 69 Diseño terminado del sistema SCADA de llenado y envasado de botellas de cola



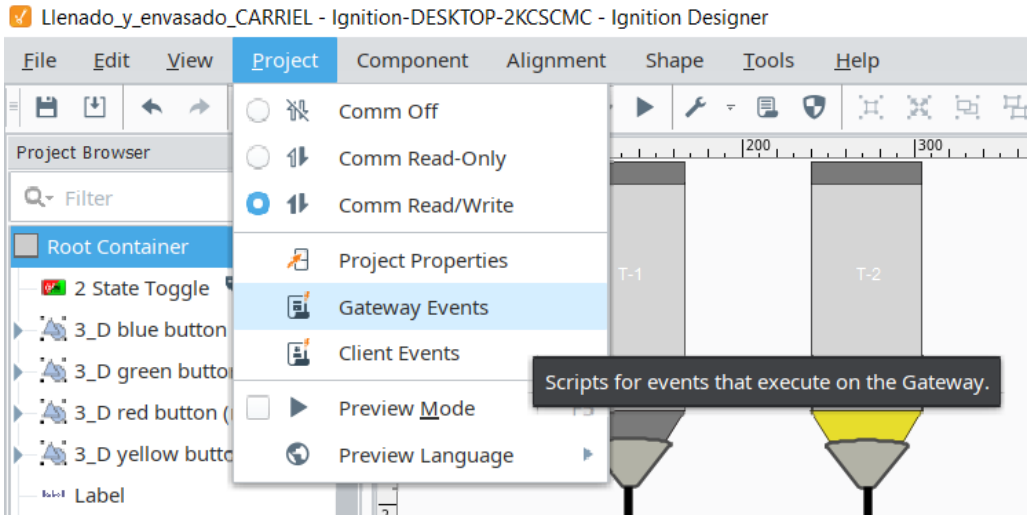
Fuente: El autor

3.2 Creación de scripts y eventos del diseño SCADA

Los scripts en Ignition son básicamente líneas de código que conllevan funciones primordiales para ejecutar tu diseño a través de eventos. Estos eventos pueden ser para el cliente o para el Gateway. Como estamos usando el protocolo de comunicación OPC UA por defecto en Ignition, la mejor opción para crear scripts y que estos terminen registrados en el Gateway es Gateway Events. Esto con el fin de analizarlos en la página que ofrece Ignition su estado y eventos.

Para esto en la parte superior izquierda del programa, se hace clic en el apartado de Project y clic al Gateway Events.

Figura 70 Ubicación de Gateway Event



Fuente: El autor

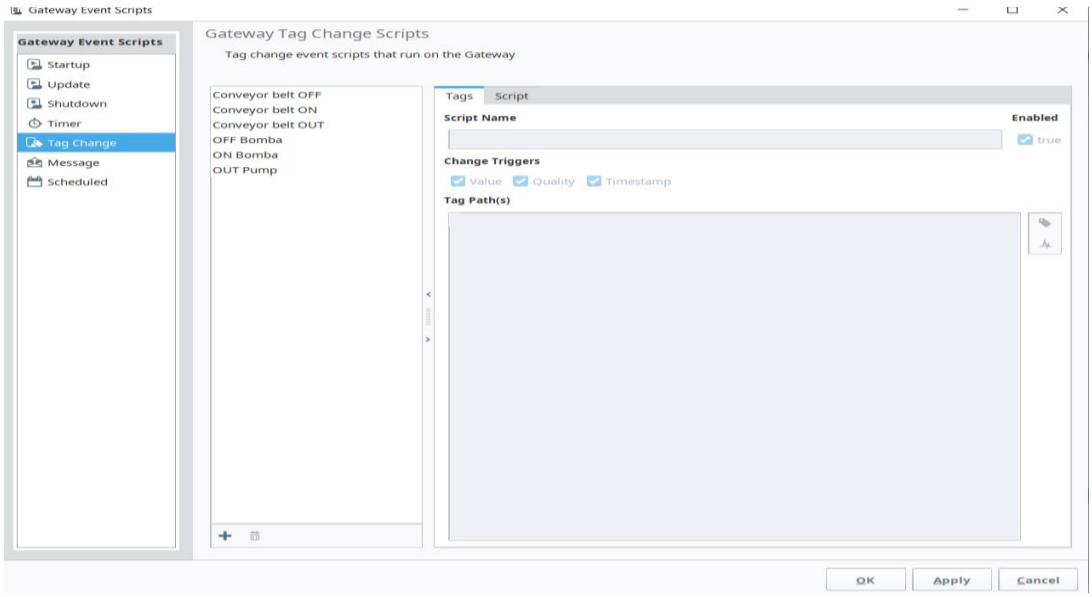
Este abrirá una ventana llamada Gateway Event Scripts la cual se usará para escribir las diferentes líneas de código que dará funciones a los componentes y tags del diseño.

Esta ventana ofrece diferentes opciones de tipos de eventos como Timer, Startup, Update, Shutdown, Tag Change, Message y Scheduled. Cada una tiene su función distinta a pesar de tener el mismo método de codificación. El evento por usar será el “Tag Change”.

El Tag Change es un cambio de etiqueta que tiene sus valores y parámetros propios determinados en el diseño. Se tiene que escribir condiciones en el script para lograr dicho cambio, como por ejemplo si se activa un botón de encendido al sistema, la señal de la bomba debería encenderse o viceversa.

Para realizar un Tag Change, se tiene que primero añadir el script en el símbolo de **+** en la parte inferior de la ventana, darle cualquier nombre que se desee y luego en Tag Path(s) copiar la ubicación del tag que deseas que interactúe con tu script o añadirlo usando el botón de etiqueta en la parte derecha del Tag Path(s). La ventana luce de esta manera:

Figura 71 Ventana Gateway Event Scripts



Fuente: El autor

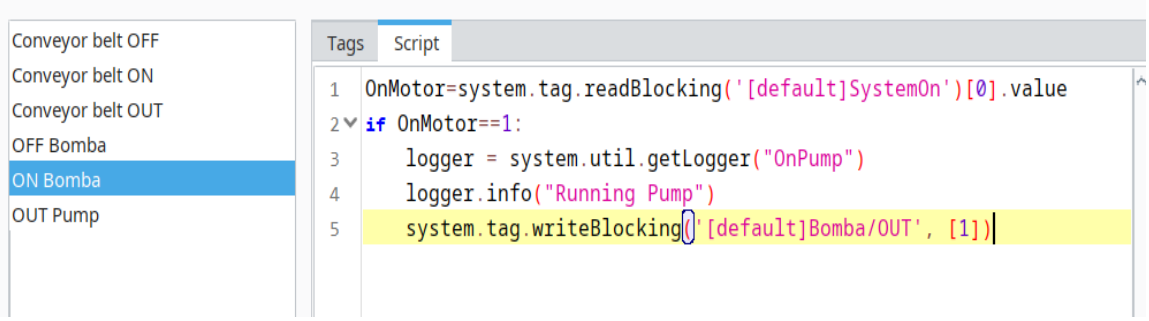
Sabiendo lo que es un script y en donde realizarlos, se procede con la codificación de los scripts tal cual necesite el diseño del sistema SCADA.

3.2.1 Script change ON Bomba

La idea general de este script es cuando el tag ‘SystemOn” se active (pulsando el botón de encendido), identificado por el indicador “System On/OFF”, el motor de la bomba se encienda junto a su señal.

En el Tag Path(s) se añade la ubicación del tag “SystemOn” que sería ‘[default]SystemOn’ y pestaña de Script añadimos el siguiente código:

Figura 72 Script de ON Bomba



Fuente: El autor

Antes de explicar el script es importante resaltar lo que significa “logger”. Logger es un comando de data String/texto que se envía hacia la base de datos. Este mensaje puede ser cualquier texto que se escriba en comillas como en este caso es “Running Pump” o en español “Encendiendo la bomba”

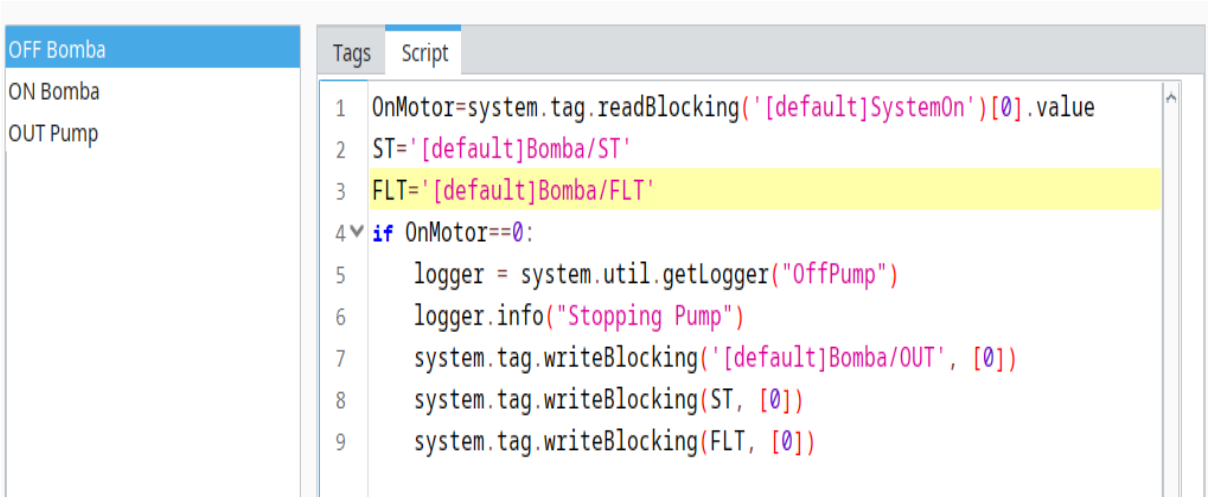
Primero se declara “OnMotor” como un valor ligado a la lectura del tag “SystemOn” dándole un valor de 0 por defecto. La condición es que, si el tag SystemOn se activa, el tag de la Bomba OUT también se activara. Este tag será importante en otro Script.

3.2.2 Script change Off Bomba

La idea es que cuando se presione el botón de apagado (que está ligado al tag SystemON) la bomba se detenga y se apague su señal.

Se añade la misma ubicación '[default]SystemOn' el Tag Path(s) y en la pestaña script se escribe el siguiente código:

Figura 73 Script de OFF Bomba



Fuente: El autor

Se define 3 variables: OnMotor, ST y FLT. OnMotor está vinculado al tag SystemOn asi mismo como con el script de ON Bomba. ST y FLT son los tags del grupo de bomba. El mensaje que aparecera en la base de datos sera "Stopping Pump" o "Apagando Bomba".

Cuando el motor de la bomba se apague o el tag SystemOn se desactive, los tags booleanos de la Bomba OUT y FLT se desactivarán. ST siendo un tag integrer o entero, su valor se volvera 0 lo que significa que la señal de la bomba y el motor se apagaran.

3.2.3 Script Change OUT Bomba

Este script se usará para simular cuando la bomba este experimentado una falla debido a la falta del activado de los tags QA o QB que en este caso serian como los disyuntores de la bomba.

A diferencia de las anteriores el Tag Path sera distinto: '[default]Bomba/OUT'

Figura 74 Script de OUT Bomba

Tags | Script

```
OnMotor=system.tag.readBlocking('[default]Bomba/OUT')[0].value
if OnMotor==1:
    logger = system.util.getLogger("OUTPump")
    logger.info("Out Pump")
    FLT='[default]Bomba/FLT'
    ST='[default]Bomba/ST'
    QA=system.tag.readBlocking('[default]Bomba/QA')[0].value
    QB=system.tag.readBlocking('[default]Bomba/QB')[0].value
    if QA==0 and QB==0:
        system.tag.writeBlocking(FLT, [1])
        system.tag.writeBlocking(ST, [2])
    else:
        system.tag.writeBlocking(ST, [1])
        system.tag.writeBlocking(FLT, [0])
```

Fuente: El autor

3.2.4 Script Change Cinta ON

La idea es la misma que con el encendido de la bomba, se podría decir que hasta su código son similares, solo que en se cambia el Tag Path y la ubicación de los tags de bomba a "Transportador01". Su código es el siguiente:

Figura 75 Script de Cinta ON

Cinta belt OUT
Cinta OFF
Cinta ON
OFF Bomba
ON Bomba
OUT Bomba
Valvula 1 OPEN

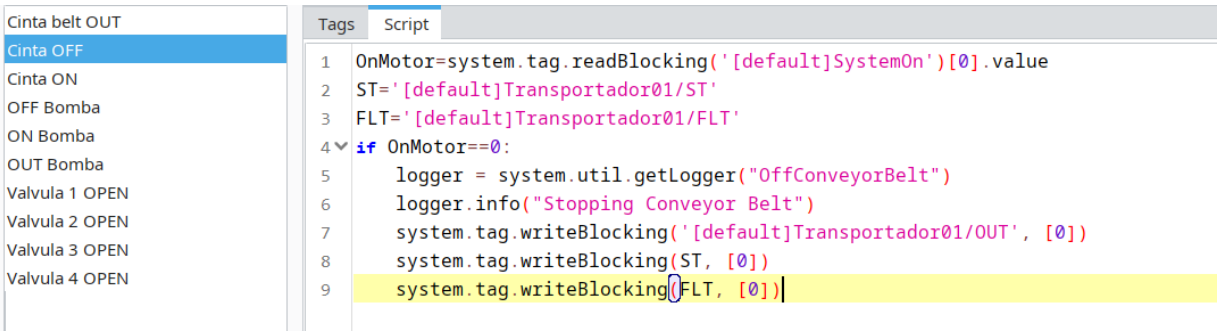
Tags | Script

```
OnMotor=system.tag.readBlocking('[default]SystemOn')[0].value
if OnMotor==1:
    logger = system.util.getLogger("OnConveyorBelt")
    logger.info("Running Conveyor Belt")
    system.tag.writeBlocking('[default]Transportador01/OUT', [1])
```

3.2.5 Script Change Cinta OFF

Similar al apagado de la bomba. Solo se cambia los tags a Transportador01 y su Tag Path(s) al del transportador, pero la estructura del código se mantiene igual.

Figura 76 Script de

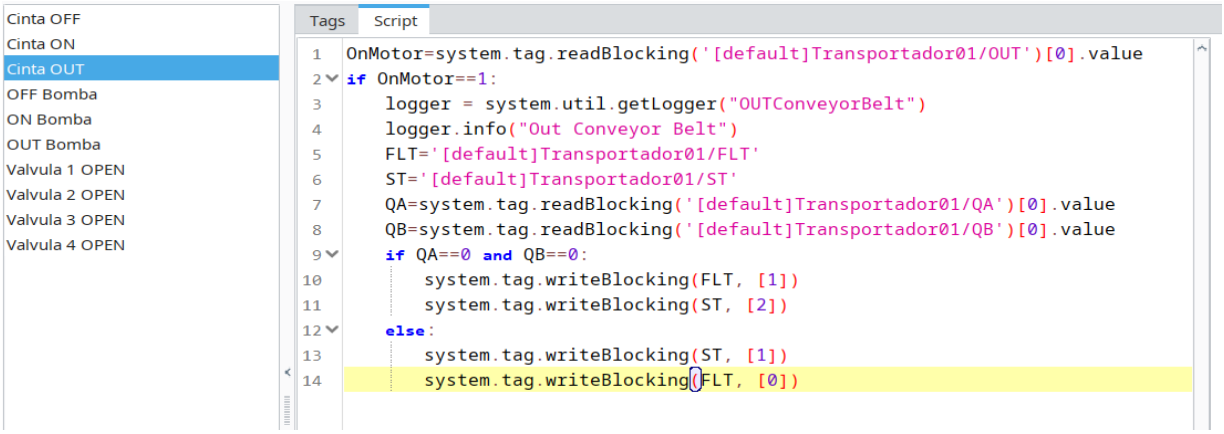


Fuente: El autor

3.2.6 Script Change Cinta OUT

Similar al OUT de la bomba. Solo se cambia los tags a Transportador01 y su Tag Path(s) al del transportador, pero la estructura del código se mantiene igual.

Figura 77 Script de Cinta OUT



Fuente: El autor

3.2.7 Script Change para las Válvulas (1-4) OPEN

La idea de estos scripts es que cuando se cumplan las condiciones de un nivel alto de líquido en la tolva, que el sistema se encuentre en un estado automático y que el sistema este encendido, cada respectiva válvula se abrirá dejando fluir el líquido de las tolvas hacia las botellas.

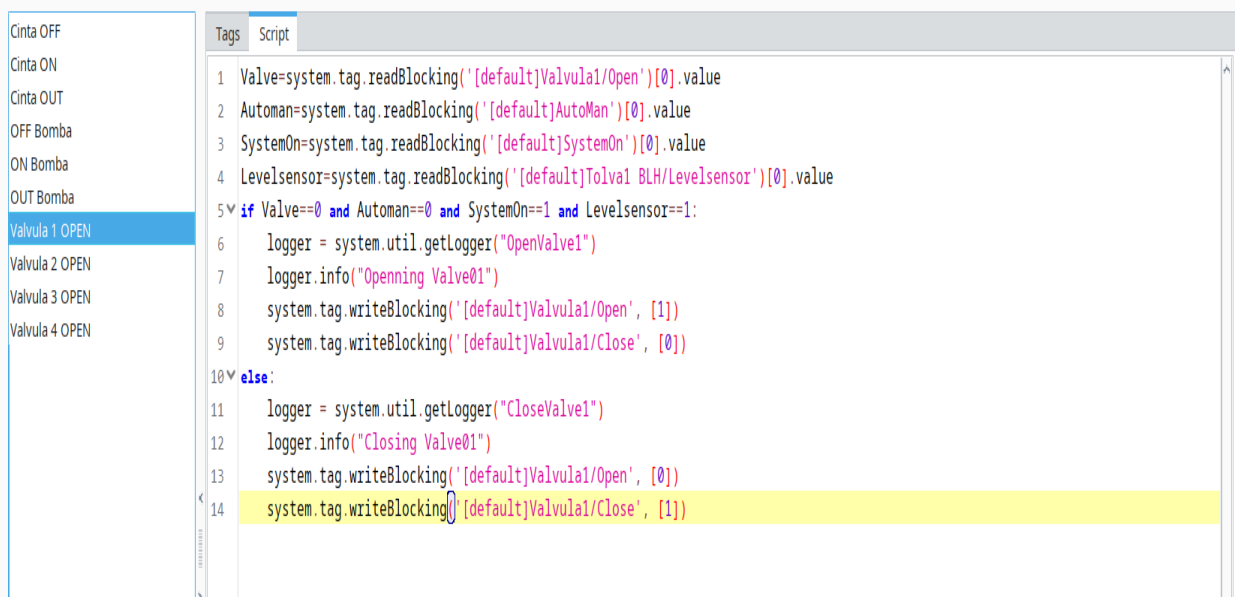
A cada una de las válvulas se les asigna un distinto Tag Path(s) las cuales son:

- **Válvula 1 OPEN:** [default]Tolva1 BLH/Levelsensor
- **Válvula 2 OPEN:** [default]Tolva2 BLH/Levelsensor
- **Válvula 3 OPEN:** [default]Tolva3 BLH/Levelsensor
- **Válvula 4 OPEN:** [default]Tolva4 BLH/Levelsensor

Se procede a escribir los siguientes códigos:

3.2.7.1 Válvula 1

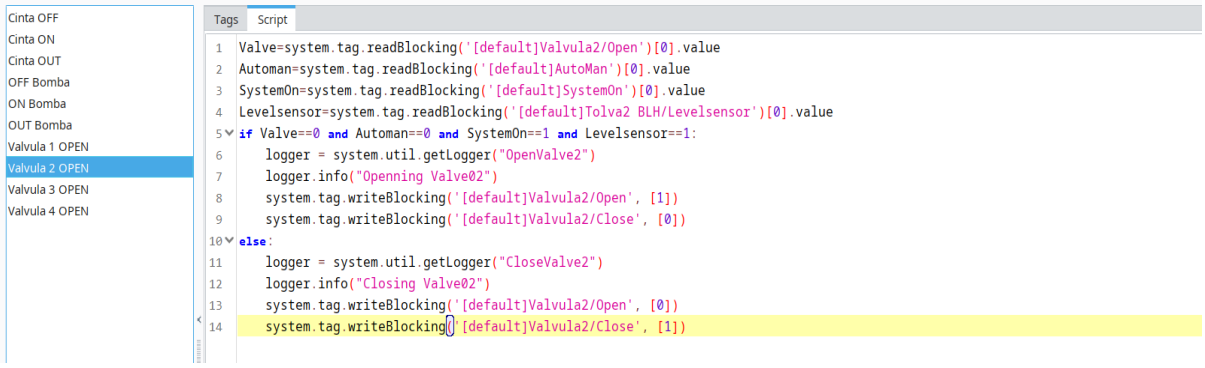
Figura 78 Script de Válvula1 OPEN



Fuente: El autor

3.2.7.2 Válvula 2

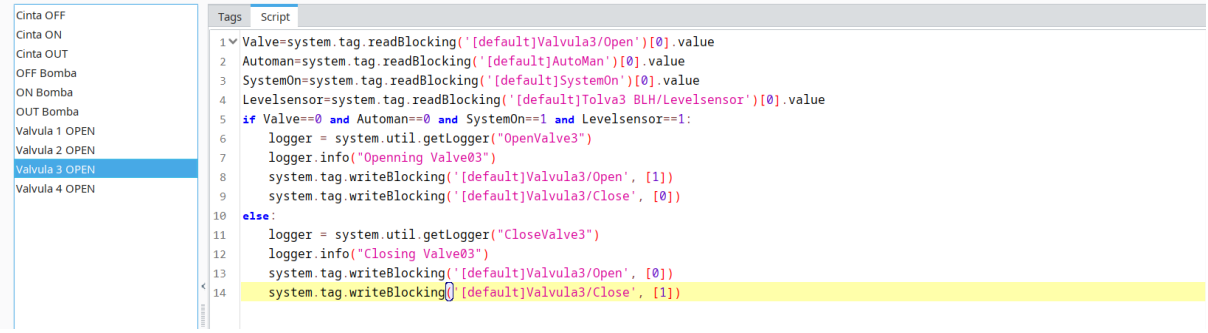
Figura 79 Script de Válvula2 OPEN



Fuente: El autor

3.2.7.3 Válvula 3

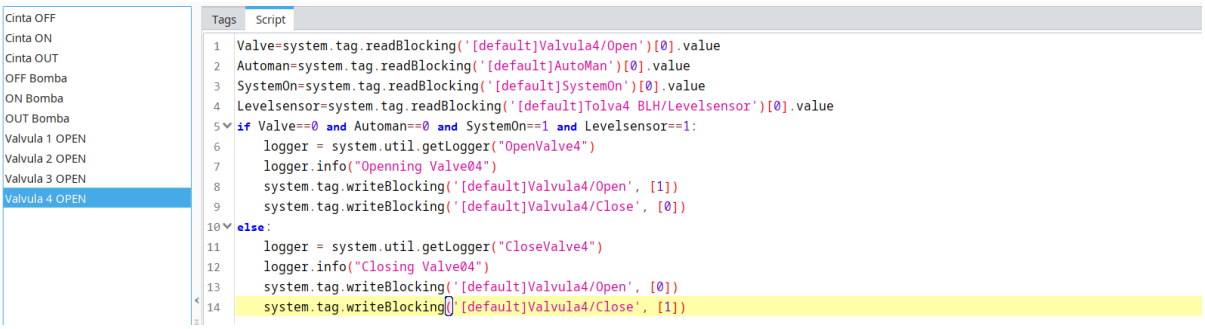
Figura 80 Script de Válvula3 OPEN



Fuente: El autor

3.2.7.4 Válvula 4

Figura 81 Script de Válvula4 OPEN



Fuente: El autor

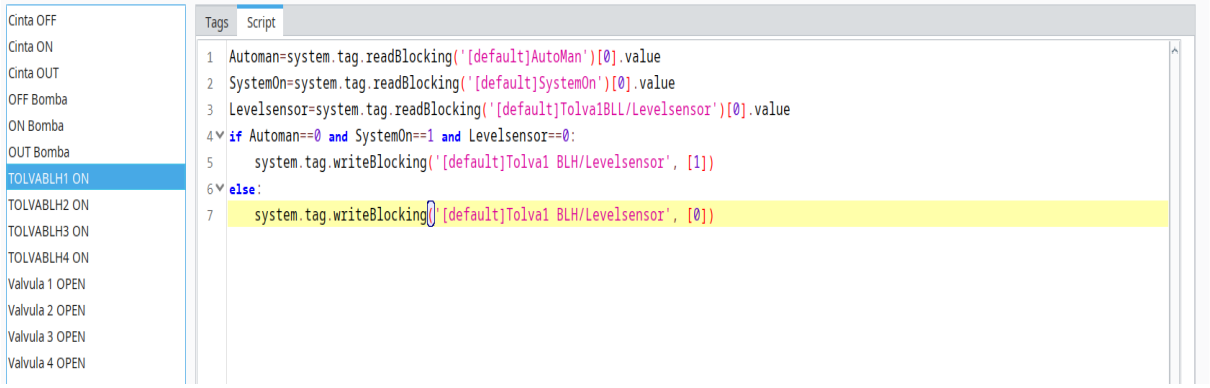
3.2.8 Script change para TolvaBLH (1-4) ON

La idea de estos Script changes de los sensores de nivel alto de cada Tolva es que se activen en cuanto cumplan las condiciones, caso contrario estas se apaguen:

- **Tolva1 BLH**
  - ✓ Sistema automático
  - ✓ Sistema encendido
  - ✓ Sensor Nivel bajo de la Tolva1 apagado
- **Tolva2 BLH**
  - ✓ Sistema automático
  - ✓ Sistema encendido
  - ✓ Sensor Nivel bajo de la Tolva2 apagado
  - ✓ Sensor Nivel alto Tolva1 apagado
  - ✓ Sensor Nivel alto Tolva3 apagado
  - ✓ Sensor Nivel alto Tolva4 apagado
- **Tolva3 BLH**
  - ✓ Sistema automático
  - ✓ Sistema encendido
  - ✓ Sensor Nivel bajo de la Tolva3 apagado
  - ✓ Sensor Nivel alto Tolva1 apagado
  - ✓ Sensor Nivel alto Tolva2 apagado
  - ✓ Sensor Nivel alto Tolva4 apagado
- **Tolva4 BLH**
  - ✓ Sistema automático
  - ✓ Sistema encendido
  - ✓ Sensor Nivel bajo de la Tolva4 apagado
  - ✓ Sensor Nivel alto Tolva1 apagado
  - ✓ Sensor Nivel alto Tolva2 apagado
  - ✓ Sensor Nivel alto Tolva3 apagado

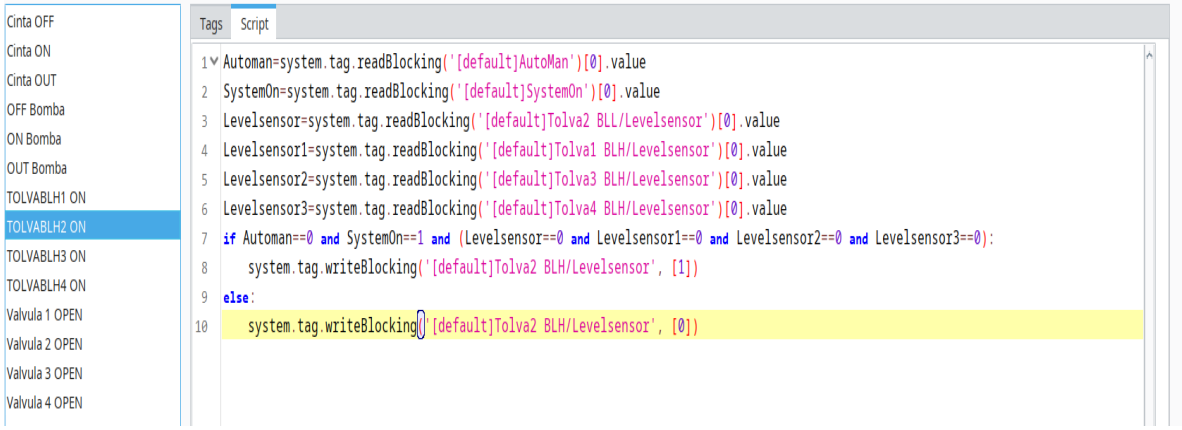
Los códigos scripts para cada Tolva son los siguientes:

Figura 82 Script TOLVA1BLH ON



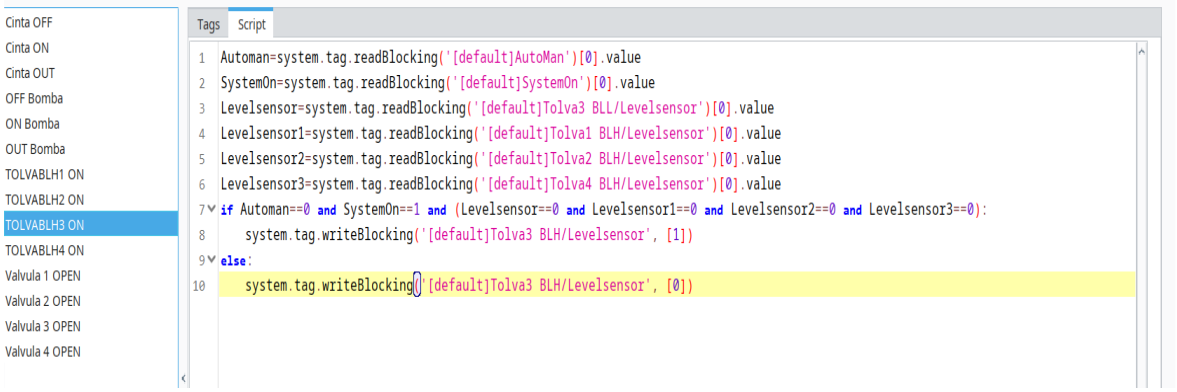
Fuente: El autor

Figura 83 Script TOLVA2BLH ON



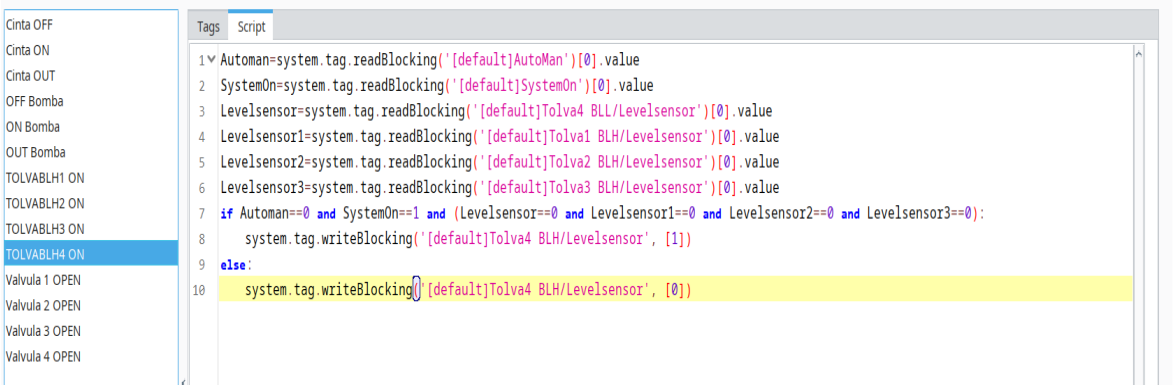
Fuente: El autor

Figura 84 Script TOLVA3BLH ON



Fuente: El autor

Figura 85 Script TOLVA4BLH ON



Fuente: El autor

3.2.9 Timers de eventos del sistema SCADA

Los Timers son scripts que se usan para repetir eventos de cualquier tipo en un mismo código. Este script se le puede dar una condición para que cumpla en su debido tiempo o simplemente repetirse cuantas veces se desee según el usuario. Se encuentra ubicado en la misma ventana de Gateway Events Scripts y a diferencia del Tag change, no necesitas ubicarle un Tag Path para accionarlo, solo escribir su script y determinarle un tiempo para su activación.

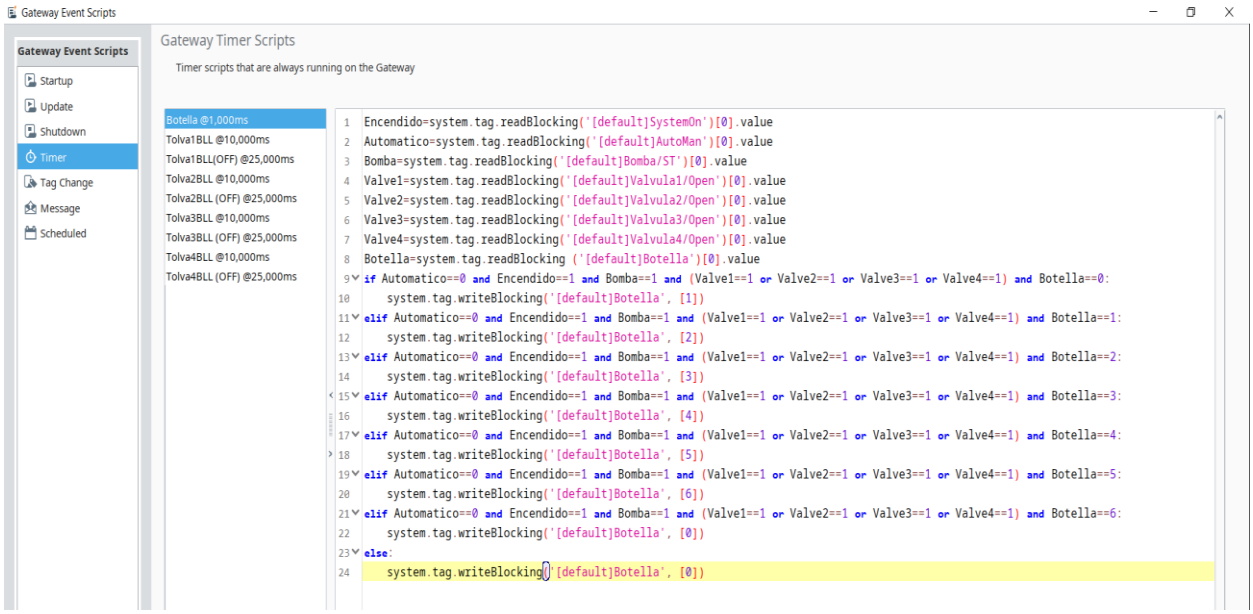
3.2.9.1 Timer de botellas

La idea de este Timer es que se repita el proceso de llenado y envasado de la botella de cola en cuando se cumplan las siguientes condiciones:

- ✓ El sistema debe estar encendido
- ✓ El sistema debe estar automático
- ✓ La Bomba debe estar encendida
- ✓ Cualquier Válvula debe estar abierta

Para lograr esta propuesta, se tiene que usar el tag de Botella, que en el previo subcapítulo se explicó que es un tag integrer, y escribir el siguiente código que se muestra a continuación:

Figura 86 Script del Timer Botella



Fuente: El autor

En fin, esto hará que el tag Botella cambie de estado cada segundo si se cumplen sus condiciones y afectara a las botellas de la cinta transportadora.

3.2.9.2 Timer de TolvaBLL ON/OFF (1 al 4)

La idea general es que pasado un tiempo el sensor de nivel bajo de la Tolva se active y eso influye en el script change de cada una de las tolvas (Revisar capítulo 3.2.8) ya que, si se nota un nivel bajo de líquido, el sensor BLH de su respectiva tolva se desactivara lo que cerrara la válvula y dejara de fluir la fórmula de esa tolva mientras se rellena con el tiempo.

Para esto cada sensor BLL debe tener dos Timers, una para que se active (Nivel bajo de fluidos, se necesita rellenar) y la otra para que se desactive (Tolva rellenada). Se tiene que primero cumplir estas condiciones para que se activen los Timers:

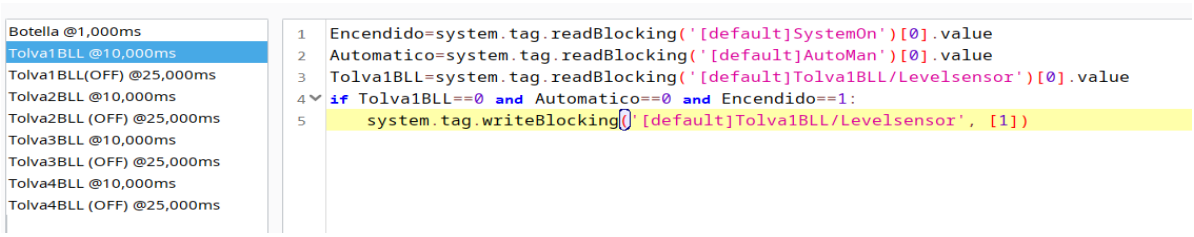
- ✓ Sistema encendido
- ✓ Sistema automático
- ✓ Tag de sensor BLL apagado

Nota: *En el caso de desactivado de los sensores el Tag del sensor BLL de su respectiva Tolva debe estar encendido.*

Los scripts son los siguientes:

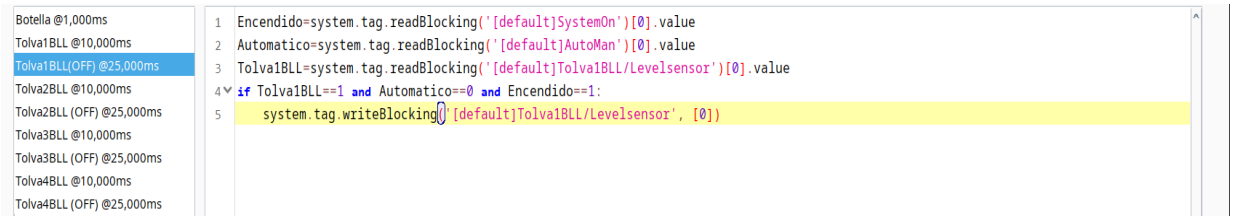
TolvaBLL1 ON/OFF

Figura 87 Script Timer Tolva1BLL



Fuente: El autor

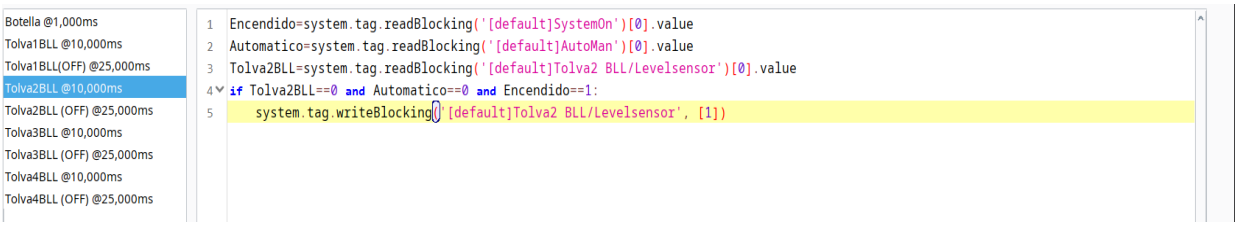
Figura 88 Script Timer Tolva1BLL (OFF)



Fuente: El autor

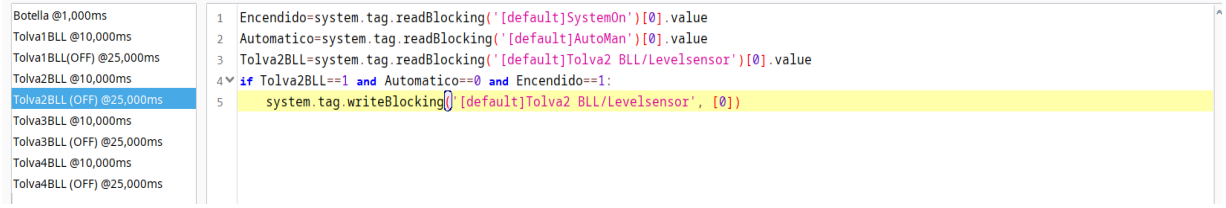
TolvaBLL2 ON/OFF

Figura 89 Script Timer Tolva2BLL



Fuente: El autor

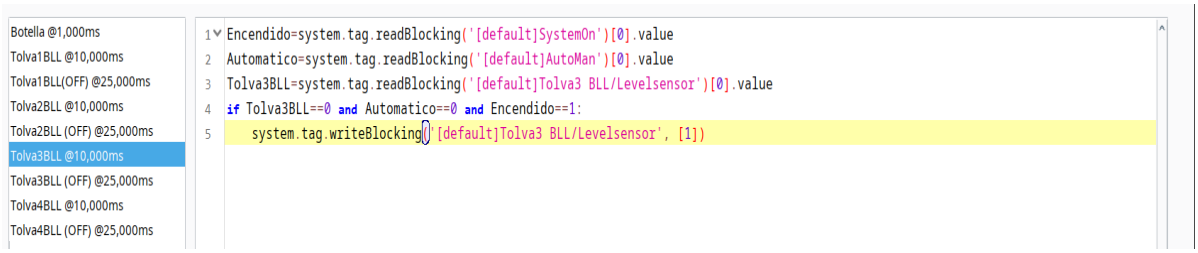
Figura 90 Script Timer Tolva2BLL (OFF)



Fuente: El autor

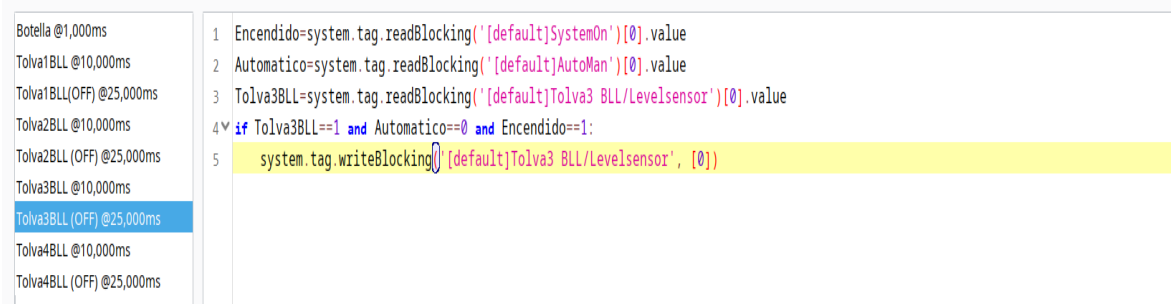
TolvaBLL3 ON/OFF

Figura 91 Script Timer Tolva3BLL



Fuente: El autor

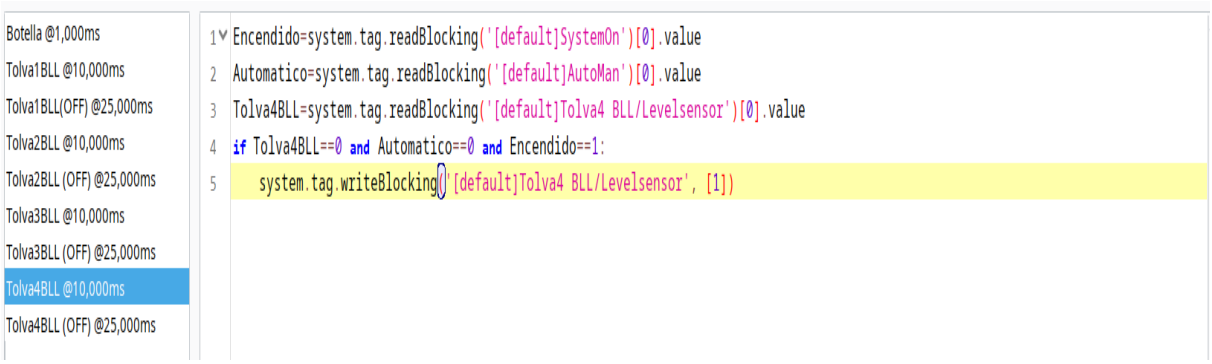
Figura 92 Script Timer Tolva3BLL (OFF)



Fuente: El autor

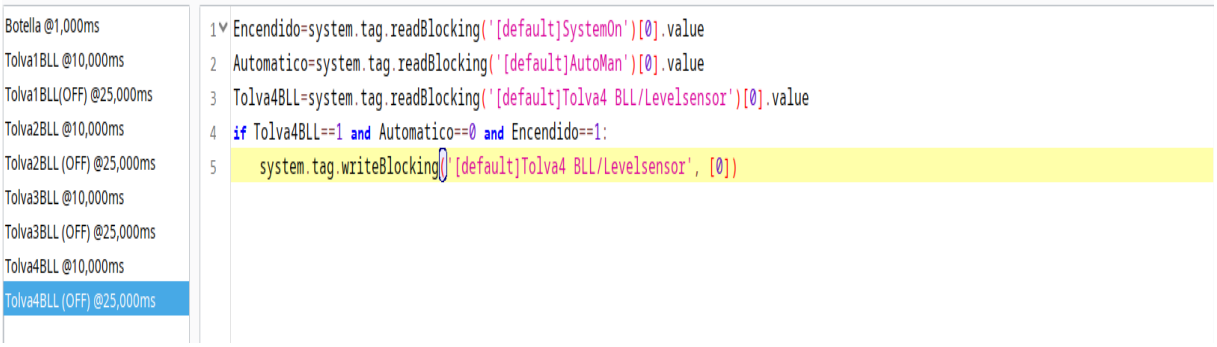
TolvaBLL4 ON/OFF

Figura 93 Script Timer Tolva4BLL



Fuente: El autor

Figura 94 Script Timer Tolva4BLL OFF



Fuente: El autor

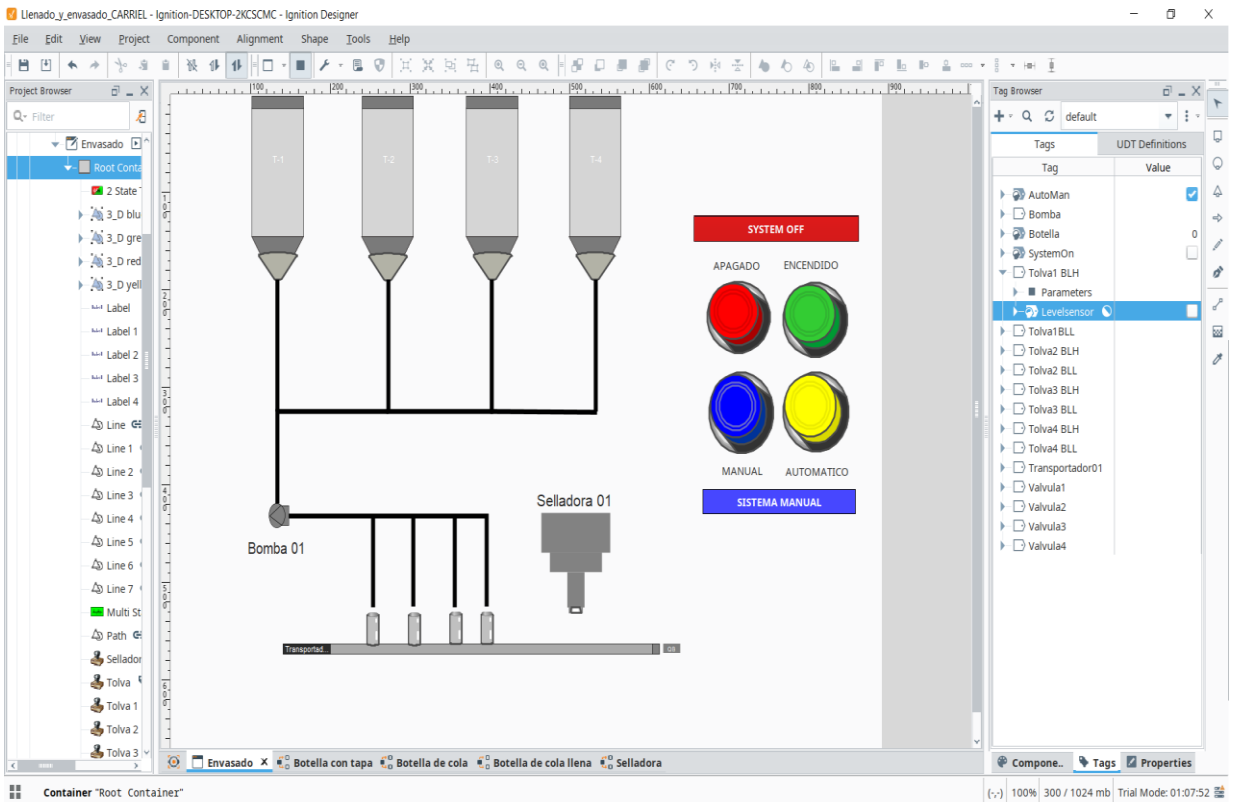
Con esto desarrollado, tenemos completado nuestro diseño y su codificación del sistema SCADA. Existe el espacio para mejorar considerando que este diseño no se ha implementado a una escala real, pero eso no quita su validez como un prototipo de un sistema SCADA del llenado y envasado de botellas de colas.

3.3 Simulación y adquisición de datos

Para simular proyectos en Ignition hay un botón en la parte superior que tiene de símbolo el botón de play. Al hacer clic, el script que codificamos empezara a actuar en base a lo que se haga en el diseño. Para eso se crearon los botones rojo, verde, amarillo y azul, estos deciden el estado de la maquina y como actuara el diseño.

Para encenderlo, se presiona el botón verde, para apagarlo el botón rojo, para cambiarlo a automático el botón amarillo y para manual el botón azul. En si los scripts y Timers usan de base en que el sistema debería estar en automático para que se ejecuten por ende eso es lo que se simulara.

Figura 95 Sistema SCADA completo y codificado



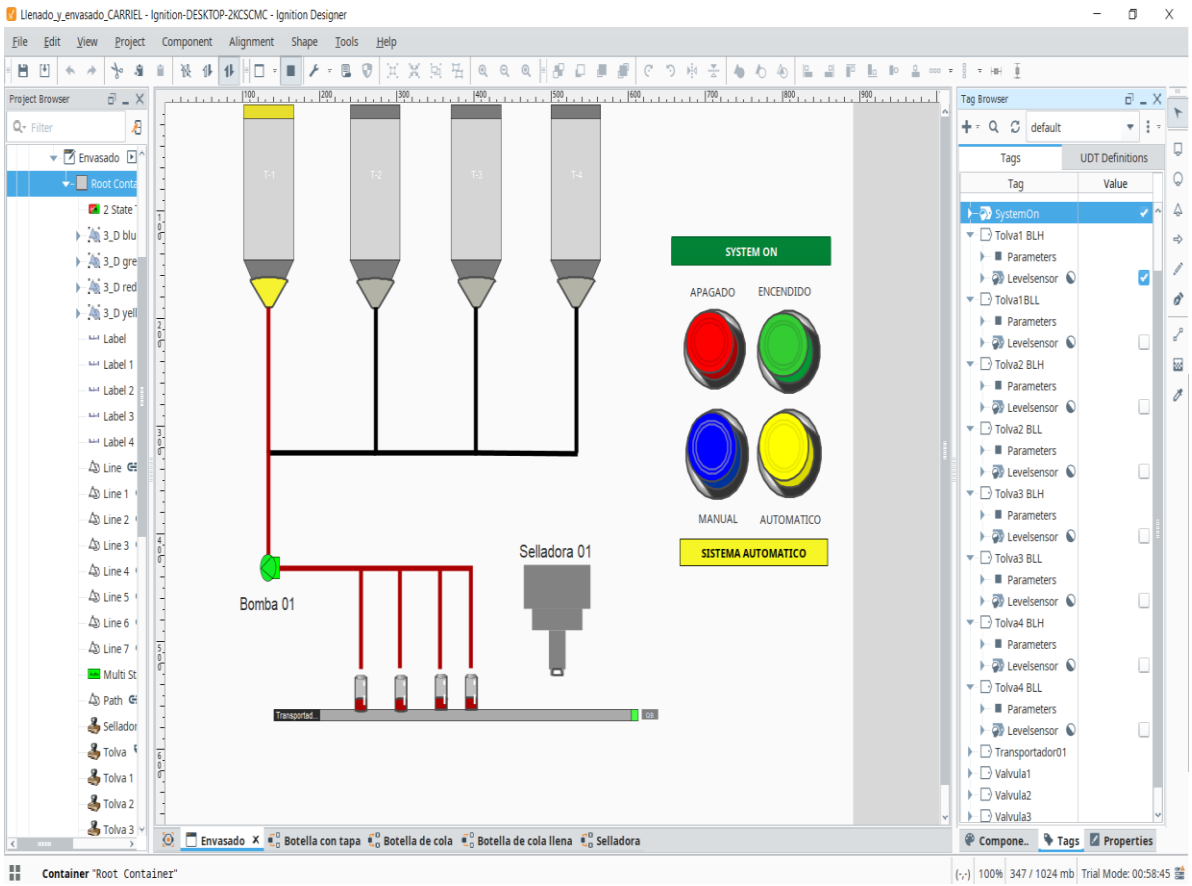
Fuente: El autor

La simulacion se divide en 4 etapas en cada etapa 1 sola valvula conectada a la tolva debe de funcionar mientras la cinta lleva las botellas. Lo que se quiere lograr son dos objetivos principales:

- ✓ Que las botellas se llenen y salgan envasadas
- ✓ Que despues de un tiempo, las tolvas se iran vaciando y cambiando de valvula para continuar con la produccion.

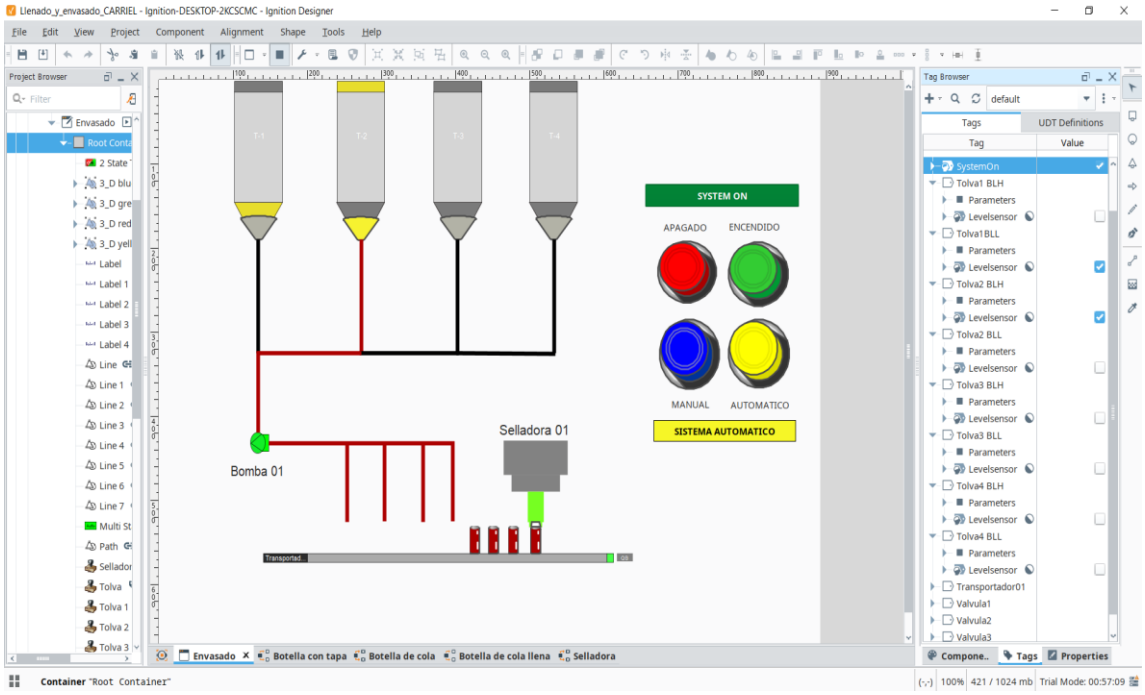
Al ejecutar el programa y presionar los botones de encendido y automatico nos dio como resultado las siguientes imágenes:

Figura 96 Primer estado de simulación



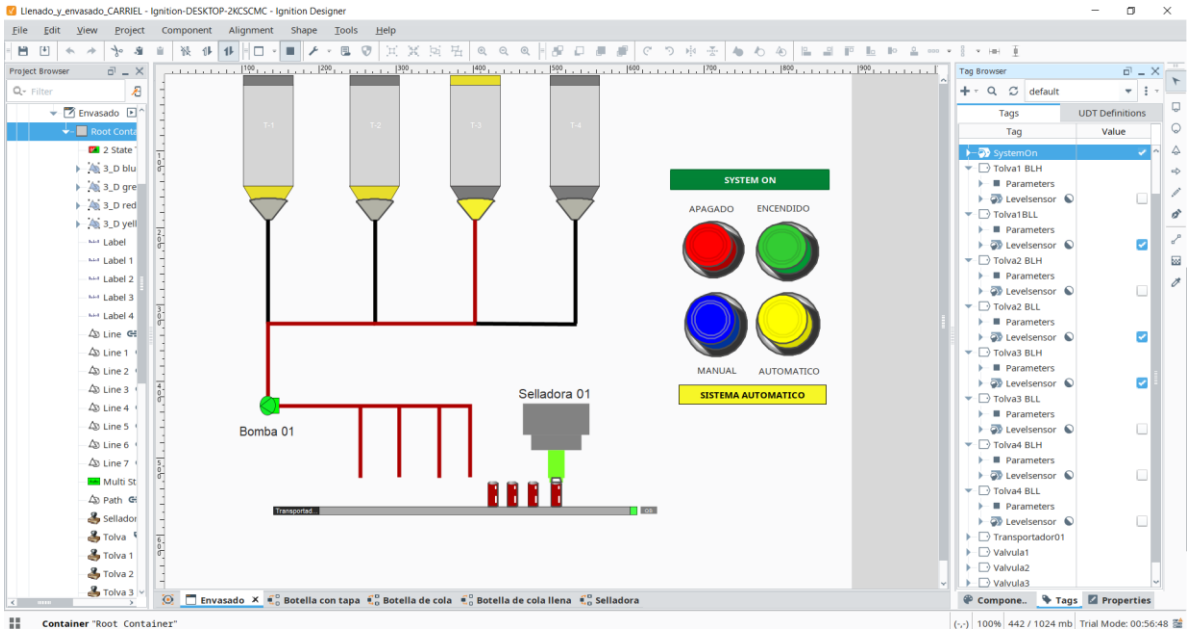
Fuente: El autor

Figura 97 Segundo estado de simulación



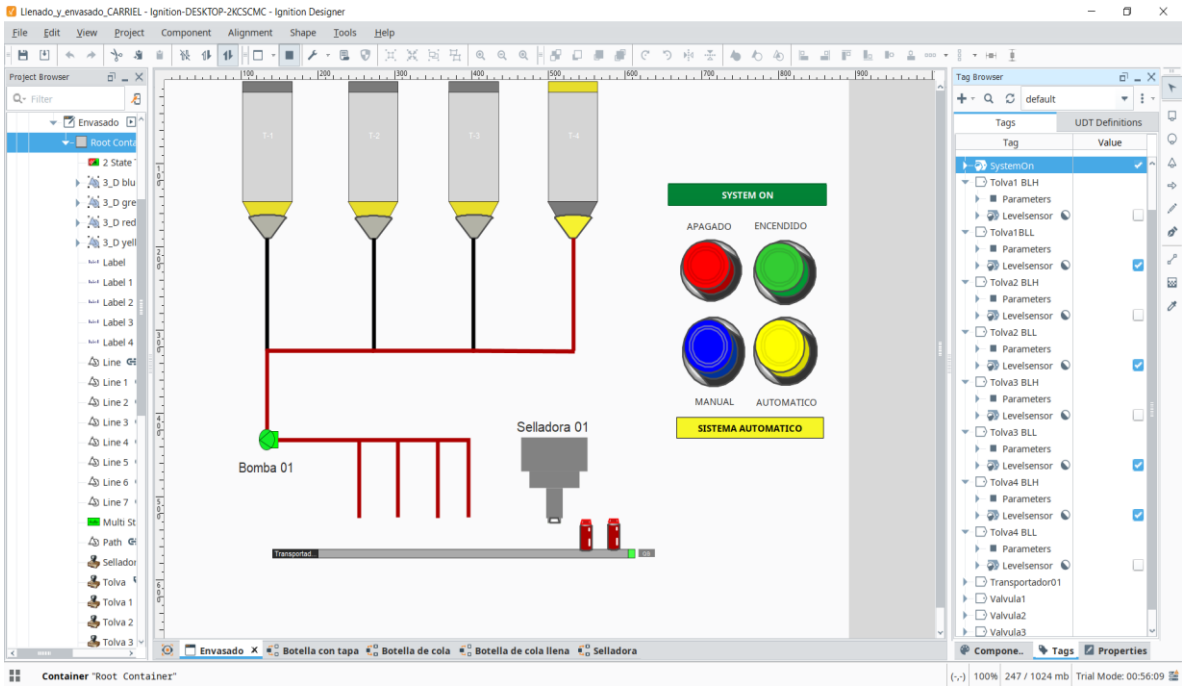
Fuente: El autor

Figura 98 Tercer estado de simulación



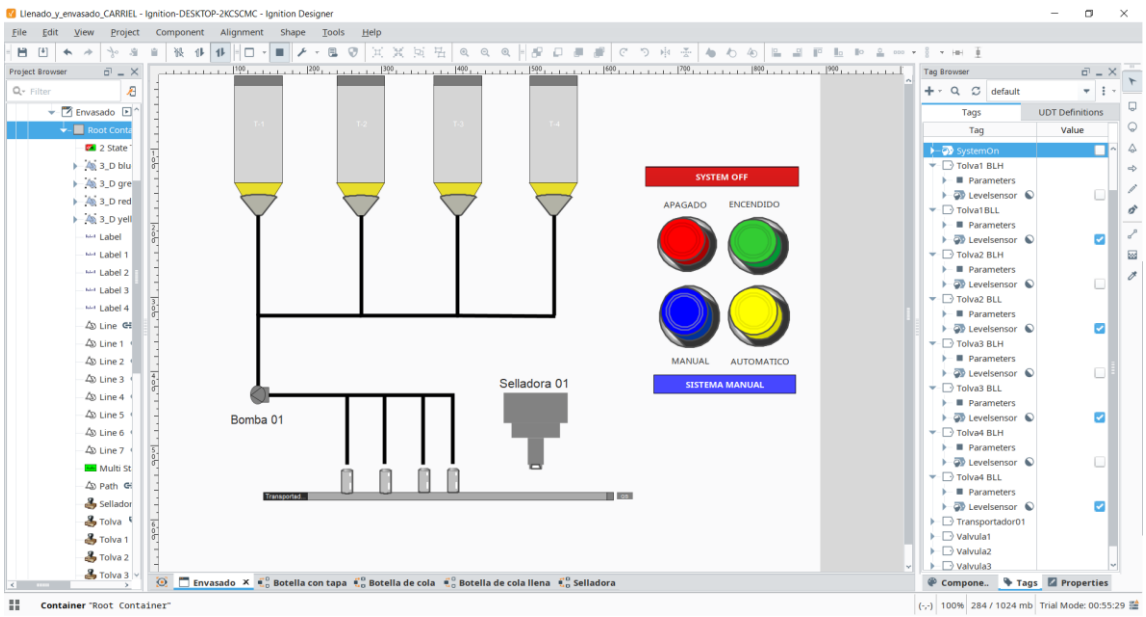
Fuente: El autor

Figura 99 Cuarto estado de simulación



Fuente: El autor

Figura 100 Apagado del sistema



Fuente: El autor

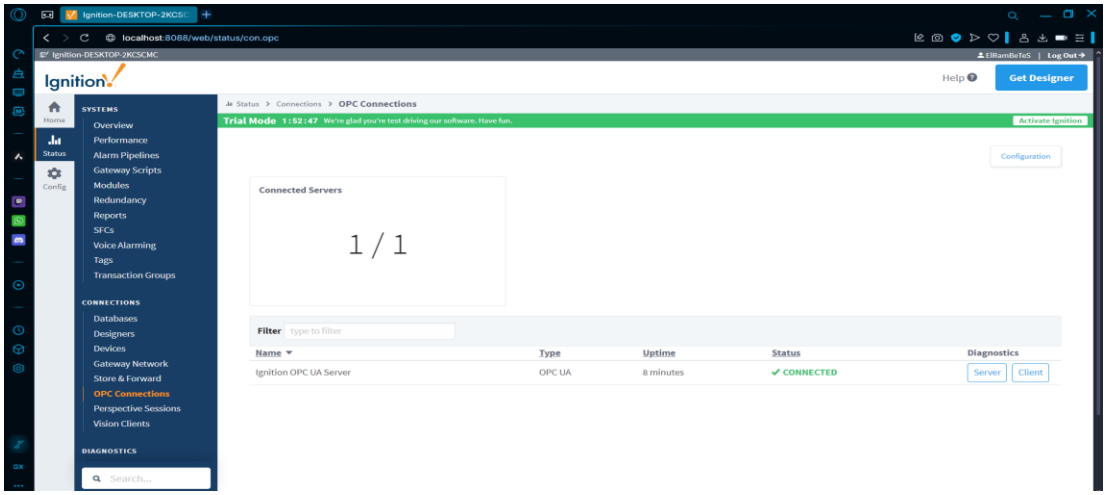
Como se puede observar en las figuras anteriores, la simulación es un éxito y se procede a verificar en el Gateway de Ignition si este adquirió los datos que se le pidió que enviara a la base de datos mediante los scripts. Para eso, abriendo en nuestro navegador ponemos el siguiente enlace:

http://localhost:8088/web/home?9

Este enlace es el “Gateway” gratuito que ofrece Ignition. Sirve como la base de datos donde se guardarán tus proyectos, como una especie de nube. Permite revisar datos históricos, conectar a diferentes bases de datos, revisar las conexiones, tags, dispositivos, entre otros.

Lo que interesa primero es revisar el tipo de conexión que uso nuestro proyecto. Se dirige a Status y OPC Connections y como se había mencionado, Ignition por defecto usa el protocolo de comunicación OPC UA. La siguiente imagen confirma lo dicho:

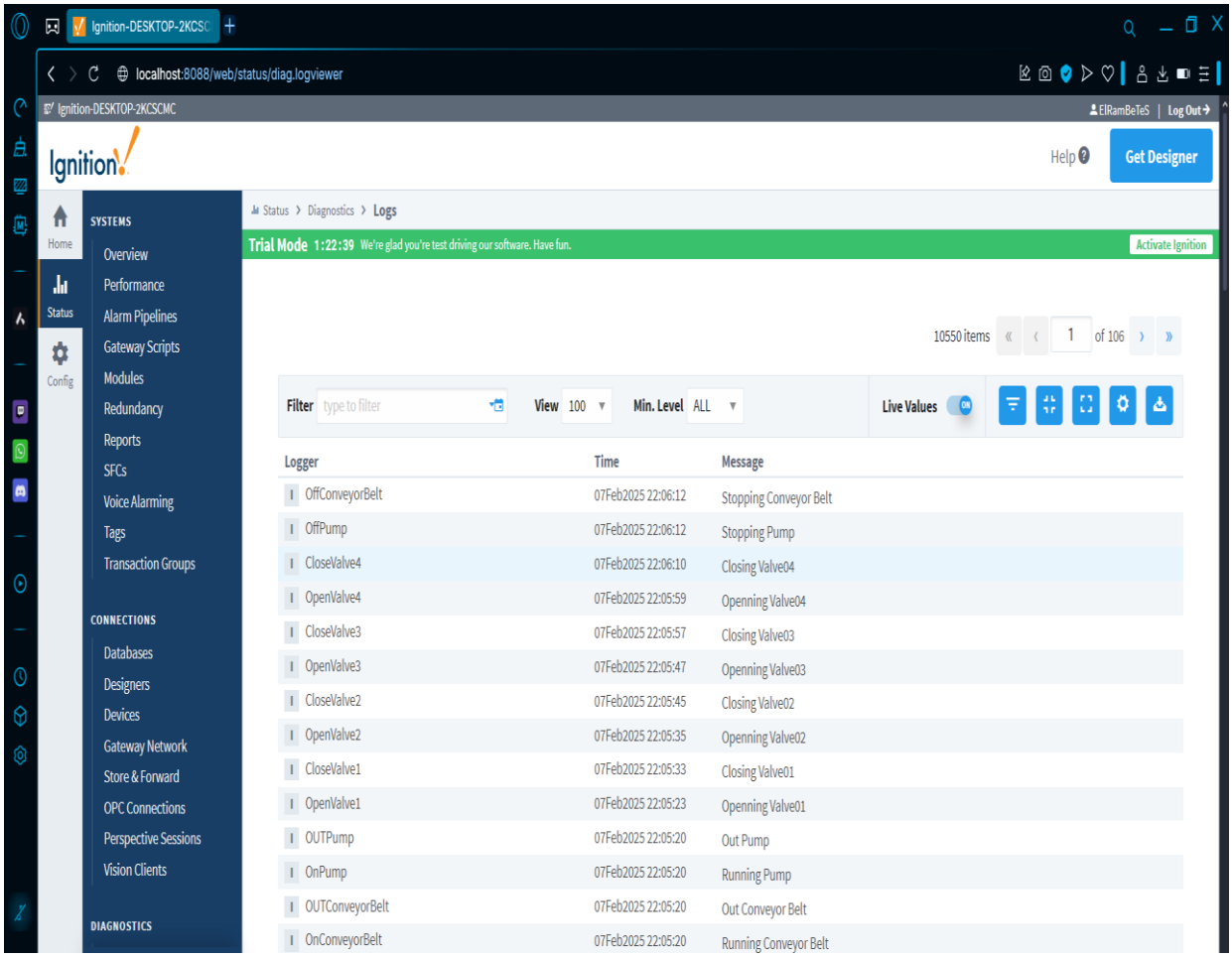
Figura 101 Verificación de conexión a servidor OPC UA



Fuente: El autor

En “Logs” Ignition te resume todos los eventos ocurridos en el tiempo que se tuvo encendido el dispositivo, incluyendo los mensajes que se enviaron al Gateway al usar “logger” en los scripts. Una vez terminada la simulación, se confirma en la siguiente figura que el Gateway adquiere los datos de que se activa, que se desactiva y cuánto tiempo duro todo el proyecto.

Figura 102 Logs de los últimos eventos del proyecto



Fuente: El autor

3.4 Análisis de datos adquiridos y datos financieros

3.4.1 Análisis eficiencia según datos logs

Tal como muestra la figura 102, los últimos logs del evento confirman que el sistema funciona, y que todo un ciclo dura exactamente 52 segundos en donde las válvulas permanecen abiertas por 49 segundos. Tomando en cuenta que cada 6 segundos 4 botellas de cola son llenadas y envasadas, haciendo los siguientes cálculos tenemos:

49/6=8.16 (Redondeando da 8)

8*4=**32 Botellas**

En menos de 50 segundos se producen 32 botellas listas para su distribución en base al prototipo de sistema SCADA. En un promedio por minuto se tienen listas **40 botellas** de cola. Si lo pasamos a la hora serán **2400 botellas** de cola y considerando una jornada laboral de 8 horas al día tenemos que se producen **19200 botellas** de cola al día en un sistema de llenado y envasado básico.

Cabe recalcar, esto es un caso hipotético y aún hay espacio para mejorar ya que este sistema no se ha implementado en un sistema industrial real donde la escala es mucho mayor, pero sirve como un ejemplo a lucir del potencial de Ignition en la industria ecuatoriana.

3.4.2 Análisis financiero

Ignition es prácticamente gratuito, sin embargo, es gratuito y accesible porque ofrece a cualquier persona dos horas gratuitas en la que la conexión al Gateway estará disponible. Esto a escala industrial no es conveniente ya que una vez que se terminen las dos horas el sistema entero se caerá hasta que se vuelva a reestablecer la conexión. Conviene más comprar la licencia ya que este es para siempre en un solo pago.

Volviendo al proyecto, estos serían los costos en caso de que se desee implementarlos en una industria de bebidas. El costo puede variar con el tiempo según la inflación o el fabricante, esto son precios estimados en base al mercado:

Tabla 1 Presupuesto de la implementación del sistema SCADA

MATERIALES	COSTO UNITARIO	COSTO ACTUAL
Licencia de Ignition SCADA	$ 8,000.00	$ 8,000.00
Tolva (4)	$ 4,845.23	$ 19,380.92
Válvulas (4)	$ 48.00	$ 192.00
Bomba	$ 376.95	$ 376.95
Cinta transportadora	$ 2,327.00	$ 2,327.00
Maquina selladora	$ 1,850.00	$ 1,850.00
Sensores de nivel (8)	$ 200.00	$ 1,600.00
TOTAL	**$ 17,647.18**	**$ 33,726.87**

CAPITULO 4: CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

La propuesta del uso de un software emergente para la supervisión y gestión de procesos cruciales como el llenado y envasado de productos alimenticios es un gran paso a la innovación de la automatización en las industrias alimenticias del Ecuador. En cuanto las empresas empiecen a adoptar este software, estas observaran mejoras en su producción, adquisición de datos y estarán mejor preparados para afrontar cualquier error que se presente en la producción sin necesidad de intervención humana.

En la actualidad, el sector industrial en general ha desarrollado una dependencia a la automatización y de gente capacitada con conocimiento profundo en la automatización con sus métodos y softwares que ofrece este servicio. Ignition siendo tan versátil y accesible cuenta con todo lo que un ingeniero electrónico necesita para diseñar, supervisar, controlar y gestionar cualquier proceso automatizado garantizado la eficiencia, trazabilidad, operabilidad y estabilidad que cualquier empresa requiere para continuar con su funcionamiento.

En conclusión, este trabajo ofrece una guía básica sobre como diseñar un prototipo de sistema SCADA usando de ejemplo el llenado y envasado de botellas de cola, utilizando un software no frecuentemente usado en las industrias de Ecuador como lo es Ignition en comunicación OPC UA usando simulaciones para probar su eficiencia, adaptabilidad y compatibilidad.

4.2 Recomendaciones

- **Implementación a Escala Real:** Es recomendable llevar a cabo una evaluación integrando el sistema SCADA a una línea de producción real de esta manera se valida su desempeño ante verdaderas condiciones adversas en la industria, detectar errores y empezar a mejorar el sistema.
- **Capacitación:** Al ser un software accesible y nuevo para la gente que desee utilizarlo, se requerirá de capacitación ya que es fundamental para el funcionamiento del programa y su implementación en la industria.
- **Reforzar la Ciberseguridad:** Respecto al protocolo OPC UA, este permite la comunicación en redes Wifis, por ende, es crucial reforzar la seguridad informática en orden para prevenir ataques DDoS, accesos no autorizados, proteger la información del usuario y conservar la integridad de la base de datos.
- **Mejoramiento continuo:** Al ver tus datos en la implementación real, deja mucho espacio para el mejoramiento de tu diseño. Con cada prueba, es crucial tomar nota e ir mejorando tu sistema para llegar al óptimo funcionamiento que se desee.

REFERENCIAS

Acuña, J. (1990). Automatización industrial: Definición y conceptos. (E. T. Rica, Ed.) *Tecnología en Marcha, 10*(1), 27-30.

BCE. (2020). *Formación Bruta de Capital Fijo*. Obtenido de Cuentas Nacionales: https://contenido.bce.fin.ec/documentos/PublicacionesNotas/Catalogo/CuentasNacionales/Anuales/Dolares/IndiceCtasNac.htm

Inductive Automation. (2010). *Innovative SCADA Software for Process Control and Data Management*. Obtenido de Inductive Automation: https://inductiveautomation.com/distributor-landing/espanol

Laurente Raimundo, Y. J. (2023). Implementacion de sistema SCADA aplicado a la operación de transmision de energía eléctrica para el centro de control de la empresa Conelsur. Obtenido de http://hdl.handle.net/20.500.12894/9878

Montalvo, W., Encalada, P., Miranda, A., Garcia, C. A., & Garcia, M. V. (Febrero de 2020). Implementación de OPC UA en una Plataforma Web para la integración de comunicación en el área de producción. *Revista Ibérica de Sistemas e Tecnologias de Informação*(E26), 667–680.

Organo del gobierno del Ecuador-Tribunal Constitucional. (2002, 04 de noviembre). *REGLAMENTO DE BUENAS PRACTICAS PARA ALIMENTOS PROCESADOS.* Quito: Agencia Nacional de Regulacion, Control y Vigilancia Sanitaria. Obtenido de https://www.controlsanitario.gob.ec/wp-content/uploads/downloads/2013/11/REGLAMENTO-DE-BUENAS-PRACTICAS-PARA-ALIMENTOS-PROCESADOS.pdf

Penin, A. R. (2007). *SISTEMAS SCADA.* (E. T. MARCOMBO, Ed.) Alfaomega .

Penin, A. R. (2007). *Sistemas SCADA: guía práctica.* Barcelona: Marcombo.

Pérez-López, E. (2015). Los sistemas SCADA en la automatización industrial. *Tecnología en Marcha, 28*(4), 8-13.

Racharla, K. R., Gummadi, S., & Rawashdeh, N. (2024). Ignition SCADA System for a Programmable Logic Controller Mechatronics System. *22nd International Conference on Research and Education in Mechatronics (REM)*, (págs. 126-131). Amman, Jordan. doi:10.1109/REM63063.2024.10735624

GLOSARIO

SCADA	Supervisión, control y adquisición de datos
OPC UA	Arquitectura unificada de comunicaciones de plataforma abierta
MTU	Unidad maestra
HMI	Interfaz hombre-maquina
RTU	Unidad remota
DCS	Sistema de control distribuido
PLC	Controlador lógico programable
SQL	Lenguaje de Consulta Estructurada
SSL	Capa de sockets seguros
IIOT	Internet Industrial de las Cosas
MES	Sistema de Ejecución de Fabricación
COM/DCOM	Modelo de Objetos Componentes Distribuidos’
DDoS	Ataque de denegación de servicio distribuido

DECLARACIÓN Y AUTORIZACIÓN

Yo, **Luis Fernando Carriel Sarmiento** con C.C: # 092497281-3 autor del Trabajo de Titulación: **Desarrollo de un Sistema de Control y Supervisión para la Gestión de Procesos de Llenado y Envasado en la Industria Alimentaria utilizando Software Ignition SCADA por OPC UA** previo a la obtención del título de **INGENIERO EN ELECTRONICA Y AUTOMATIZACION** en la Universidad Católica de Santiago de Guayaquil.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, 20 de febrero del 2025

f. _____

Nombre: Luis Fernando Carriel Sarmiento

C.C: 092497281-3

REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA			
FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN			
TÍTULO Y SUBTÍTULO:	Desarrollo de un Sistema de Control y Supervisión para la Gestión de Procesos de Llenado y Envasado en la Industria Alimentaria utilizando Software Ignition SCADA por OPC UA		
AUTOR(ES)	Carriel Sarmiento, Luis Fernando		
REVISOR(ES)/TUTOR(ES)	Ing. Heras Sánchez, Miguel Armando		
INSTITUCIÓN:	Universidad Católica de Santiago de Guayaquil		
FACULTAD:	Facultad de Educación Técnica para el Desarrollo		
CARRERA:	Ingeniería en Electrónica y Automatización		
TITULO OBTENIDO:	Ingeniero en Electrónica y automatización		
FECHA DE PUBLICACIÓN:	20 de febrero del 2025	**No. DE PÁGINAS:**	108
ÁREAS TEMÁTICAS:	Sistemas SCADA, Diseño Electrónico		
PALABRAS CLAVES/ KEYWORDS:	Sistema SCADA, OPC UA, Ignition SCADA, Gestión de llenado y envasado de productos alimenticios, Industria alimenticia.		

El presente trabajo de titulación propone el diseño y simulación de un sistema SCADA para la gestión de llenado y envasados de productos alimenticios usando un software nuevo en la industria alimenticia del Ecuador llamado Ignition SCADA.

El objetivo del trabajo es presentar una alternativa a la automatización de estos procesos cruciales en la industria alimenticia del Ecuador mediante el diseño de un sistema SCADA en un programa tan versátil como Ignition que se podrá modificar o adaptar a las necesidades de diferentes empresas y simularlo para comprobar su eficiencia y la comunicación de los datos en tiempo real a un servidor principal que ofrece el software usando protocolo de comunicación OPC.

Este trabajo está dividido en 4 capítulos, el primer capítulo describe e introduce el trabajo de titulación en general, el segundo capítulo es el marco teórico la cual explica el concepto y funcionamiento de los procesos y herramientas que serán utilizados en el trabajo, el tercer capítulo detalla el diseño ejemplo en este software y todo lo necesario para llevar a cabo su función y su simulación, para así finalmente en el cuarto capitulo dar recomendaciones y conclusiones del trabajo de titulación presentado.

ADJUNTO PDF:	☒ SI	☐ NO
CONTACTO CON AUTOR/ES:	**Teléfono:** +593 99 468 8401	E-mail: luisfer1400@gmail.com o luis.carriel02@cu.ucsg.edu.ec
CONTACTO CON LA INSTITUCIÓN: COORDINADOR DEL PROCESO DE UTE	**Nombre:** Heras Sánchez, Miguel Armando	
	Teléfono: +593 99 747 9664	
	E-mail: miguel.heras@cu.ucsg.edu.ec	

SECCIÓN PARA USO DE BIBLIOTECA	
Nº. DE REGISTRO (en base a datos):	
Nº. DE CLASIFICACIÓN:	
DIRECCIÓN URL (tesis en la web):	