



Informe del proyecto: AUTOMATIZACIÓN DE LÍNEA DE EMBOTELLADO DE CERVEZAS

DIEGO CARRILLO SANTAMARÍA

Tabla de contenidos

Tabla de Figuras.....	2
1. PRÓLOGO.....	3
2. OBJETIVOS.....	4
2.1 General.....	4
2.2 Específicos.....	4
3. SELECCIÓN Y JUSTIFICACIÓN DEL PROCESO.....	5
4. SOFTWARE UNITY PRO Y CONFIGURACIÓN DEL PLC	6
4.1 Software UNITY PRO XL.....	6
4.2 Configuración del PLC.....	6
5. DISEÑO DEL PROCESO.....	8
5.1 Estación de lavado (ST1)	9
5.2 Estación de secado (ST2).....	10
5.3 Estación de llenado (ST3)	10
5.4 Estación de etiquetado (ST4).....	10
5.5 Estación de Calidad (ST5).....	11
5.6 Estación de taponado (ST6).....	12
5.7 Estación de reproceso (ST7)	12
5.8 Tanque de abastecimiento.....	13
6. DESARROLLO DEL CÓDIGO	14
6.1 Main	15
6.1.1 Configuración del Proceso MAIN	18
6.1.2 Uso de Contadores.....	19
6.1.3 Uso de Temporizadores	19
6.2 Tank.....	21
6.3 Emergency STOP.....	21
6.4 Reset CTD's.....	23
7. PANEL DE CONTROL DE OPERADOR.....	24
7.1 Pantalla principal de operador.....	25
7.2 Pantalla de operación del tanque de abastecimiento	26
8. VARIABLES GENERADAS	27
9. CONCLUSIONES	30
10. BIBLIOGRAFÍA	30

Tabla de Figuras

Figura 1: PLC Modicon TSX Premium P57 104M de Schneider Electric Industries (Fuente: Schneider Electric). ...3	3
Figura 2: Formas de envasado. Distribución porcentual. (Fuente: INEGI, México)5	5
Figura 3: Logo Unity Pro XL. (Fuente: Schneider Electric)6	6
Figura 4: Configuración del X Bus en Unity Pro. a) Rack con sus respectivos módulos b) Módulo del procesador P57 104M7	7
Figura 5: Diseño de planta. Línea de embotellado de cerveza.8	8
Figura 6: Diseño de planta en modo simulación del Unity Pro.....9	9
Figura 7: Operación de la ST1. a) Estado inactivo, b) Estado activo9	9
Figura 8: Estación de etiquetado (ST4). 11	11
Figura 9: Estación ST5 de verificación de etiqueta. 11	11
Figura 10: Estación ST7 de taponado. 12	12
Figura 11: Modo de simulación del Tanque de abastecimiento, con su respectivo panel de control. 14	14
Figura 12: SFC de la función MAIN. 15	15
Figura 13: MACROSTEPS para procesos A y B. (Representación simbólica). 16	16
Figura 14: MACROSTEP para el proceso A 17	17
Figura 15: Acciones programadas en la sección MAIN. 18	18
Figura 16: Transiciones programadas en la sección MAIN 18	18
Figura 17: Función contadora CONT_A..... 19	19
Figura 18: Función Fin_Proc_A..... 19	19
Figura 19: Variables temporizadas para las acciones. 19	19
Figura 20: Acción de retardo temporizado para la salida Q_Secado..... 20	20
Figura 21: Acción temporizada en una Transición. 20	20
Figura 22: Diagrama SFC operación del tanque. 21	21
Figura 23: SFC para el manejo del paro de emergencia. 22	22
Figura 24: Acción Emergency_Act..... 22	22
Figura 25: Acción de reinicio de las variables de salida 23	23
Figura 26: Acción Restart_Release. 23	23
Figura 27: Acción Reset_Conts para reinicio de los contadores del panel principal de operación. 24	24
Figura 28: Acción RST_TANK_CTD..... 24	24
Figura 29: Panel de Control de Operador 24	24
Figura 30: Panel de operación del tanque de abastecimiento. 25	25
Figura 31: Opción de inicio y selección en pantalla. 25	25
Figura 32: Panel de visualización de material procesado y de reproceso. 26	26
Figura 33: Modo de operación del tanque de abastecimiento. A) Conteo de ciclos de operación B) Mensaje de Fin de ciclo..... 27	27
Figura 34: Entradas del sistema. 27	27
Figura 35: Salidas del sistema. 28	28
Figura 36: Variables del tipo INT utilizadas en el programa. 28	28
Figura 37: Etapas del sistema..... 29	29

1. PRÓLOGO

El presente informe tiene como finalidad mostrar el trabajo realizado a nivel de diseño y programación de la automatización de una línea de proceso de embotellado de cervezas, así como mostrar los resultados experimentales que se obtuvieron en la ejecución de dicho programa. El desarrollo mencionado anteriormente hace uso del dispositivo electrónico para automatización de procesos llamado *PLC* por sus siglas en inglés de *Programmable Logic Controller*, también llamado *Autómata Programable*.

Desde que en 1968 apareció el primer PLC, como propuesta de reemplazo electrónico de sistemas de lógica cableada cuya programación era realizada por personal con nivel medio de formación en electricidad, empleando un método basado en diagramas eléctricos utilizados en automatización industrial y mediante el uso de ordenadores industriales hasta hoy, el autómata programable ha ido evolucionando gracias a los avances producidos en el campo de la electrónica, ofreciendo día a día mayores prestaciones. (Antúñez S. F., 2016)

Como es conocido por muchos, la automatización juega un papel de vital importancia en la industria y hasta en el hogar desde hace varios años, y aún a la fecha se siguen utilizando dispositivos que fueron creados en la década de los 60's como lo es el controlador lógico programable (PLC), también conocido como Autómata Programable, el cual fue el controlador que se utilizará para realizar el proyecto.

En el caso nuestro se ha utilizado el PLC *TSX Premium* de Schneider (Figura 1) mismo que se encuentra en el laboratorio de Automática de la UC3M y que utiliza el software *Unity Pro XL* propiedad de *Schneider Electric Industries*, el cual no solo nos permite crear el código de programación si no que también nos da la opción de simular el proceso gracias a sus propiedades y características propias de entorno de programación.



Figura 1: PLC Modicon TSX Premium P57 104M de Schneider Electric Industries (Fuente: Schneider Electric).

2. OBJETIVOS

2.1 General

- Automatizar el proceso de una línea de producción de embotellado de cervezas haciendo uso del PLC Modicom TSX Premium, así como el software de edición Unity PRO XL, aplicando el lenguaje de programación SFC (Grafcet) y el modo de simulación propio del software, así como las funciones propias que permitan optimizar el proceso al máximo.

2.2 Específicos

- Diseñar el proceso utilizando el lenguaje SFC (Sequential Function Chart) definido por la norma IEC 61131-3 para programación basada en GRAFCET.
- Desarrollar una aplicación automática que controle el proceso de fabricación descrito.
- Simular el proceso haciendo uso de las herramientas gráficas tanto de programación como de animación de los procesos diseñados.
- Desarrollar habilidades y conocimiento de las herramientas y funciones que ofrece el programa Unity Pro.
- En la simulación, lograr hacer el proceso de envasado de cerveza tal y como se haría en un proceso industrializado real.
- Reproducir los resultados de los procesos en interfaces visuales que permita observar el estado a lo largo de la evolución sobre la puesta en marcha del mismo.

3. SELECCIÓN Y JUSTIFICACIÓN DEL PROCESO

Para este trabajo hemos seleccionado el proceso de embotellado de cerveza el cual es el último que se realiza antes de salir a la venta al mercado, por lo cual se ha estudiado primeramente el orden secuencial del proceso de embotellamiento de la cerveza en una fábrica comercial que suele manejar estándares de producción auditados y certificados, en donde la ingeniería aplicada se ve plasmada en dichos procesos. Es importante destacar que los procesos pueden variar dependiendo de ciertos criterios técnicos, sin embargo, nuestro modelo se basa en un prototipo estándar.

La razón por la cual se ha escogido este proceso, es principalmente porque puede tener un sin número de variables a lo largo de toda la línea de producción y es precisamente esa opulencia y versatilidad que nos llamó la atención ya que es posible hacer un diseño que a pesar que tiene respetar ciertas condiciones, no nos limita para aplicar toda la creatividad y variabilidad que queramos.

En países con gran número de población como lo es caso de México se han hecho estudios por parte del Instituto Nacional de Estadística, Geografía e Informática (INEGI) para analizar la estadística de producción de éste popular producto, en donde ha quedado en evidencia que, de todas las formas de envasado de cerveza, el embotellado de cristal no retornable representa 35%, siendo así la segunda forma de envasado más popular, como observamos en la siguiente figura. (INEGI, 2017)

Formas de envasado de la cerveza

Principales formas de envasado de la cerveza, 2016 Porcentajes respecto al valor total de la producción	
	Valor
La principal forma de envasado de la cerveza es en lata, seguido de la cerveza en botella no retornable	100.0
	En lata 36.3
	En botella no retornable 35.0
	En botella retornable 28.7

Fuente: INEGI. Encuesta Mensual de la Industria Manufacturera.

Figura 2: Formas de envasado. Distribución porcentual. (Fuente: INEGI, México)

Es por esta popularidad que se ha mencionado que muchas personas el día de hoy se animan a realizarla en sus casa, el tema es que el embotellamiento juega un papel fundamental en la producción de esta popular bebida, y en fábricas de producción masiva se requiere minimizar todo lo posible el error humano, aumentar la productividad, y hacer una correcta gestión energética para que los equipos no operen en el momento que no se está produciendo, o se restrinja una estación en la que no haya producto para procesar.

De igual manera, se requieren controles de calidad para tener toda la certeza y seguridad para que el producto que se entregará al mercado cumpla todas las especificaciones en cada botella, y así poder conseguir todos los permisos sanitarios que se requieren para la distribución y por supuesto el consumo. Así que sin mas preámbulo explicamos el diseño en la siguiente [Sección 4](#).

4. SOFTWARE UNITY PRO Y CONFIGURACIÓN DEL PLC

Esta sección provee información del software que se ha utilizado para el diseño y configuración del proyecto que se mencionó anteriormente en la [Sección 3](#). De igual manera se dará información fundamental acerca de la configuración del PLC que se utilizó, así como los detalles de los ajustes realizados.

4.1 Software UNITY PRO XL

Unity Pro es el software de programación, depuración y operación común para las gamas Modicon Premium, Atrium and Quantum PLC. Con sus cinco lenguajes IEC61131-3, todas las herramientas de depuración y diagnóstico, Unity Pro está hecho para aumentar su productividad de desarrollo y la facilidad de mantenimiento. Con Unity Pro, su inversión de software se optimiza, los costos de capacitación se reducen y aprovecha un potencial de desarrollo inigualable. (Schneider, 2020)

Como bien se describe en el sitio web oficial de Schneider, el software no solo nos permite programar nuestro proceso, sino que también nos permite ejecutar el modo de simulación, lo que nos facilita ampliamente el monitorear el proceso, detectar fallas y ver el desempeño en tiempo real de la ejecución del programa diseñado. En el caso nuestro se utiliza el *Unity PRO XL, v13*, un software propiedad de Schneider Electric Industries SAS.



Figura 3: Logo Unity Pro XL. (Fuente: Schneider Electric).

4.2 Configuración del PLC

En este proyecto se debe resaltar que se ha utilizado el modo simulación que ofrece el software Unity PRO XL, ya que por disposiciones generales no se habilitó el acceso al laboratorio, sin embargo, gracias a la función de simulación del software fue posible hacer la configuración tal y como si se tratase de una práctica de laboratorio. Dicho lo anterior, se ha utilizado el PLC Modicon TSX Premium P57 104M de Schneider Electric Industries tal y como se ha mencionado en la [Sección 1](#), este PLC tiene una gran variedad de características las cuales proveen una serie de beneficios al trabajar con éste dispositivo, según el manual de usuario, mismo que podemos consultar en el sitio web oficial del Schneider Electric se mencionan las principales particularidades:

- 5 idiomas IEC como estándar: LD, ST, FBD, SFC, IL.
- CPU de alto rendimiento con 37 ns por instrucción y hasta 7 Mb de programa
- Sistema multitarea de alto nivel
- Un amplio catálogo de módulos para aplicaciones específicas (seguridad, procesamiento de reflejos, contador, control de posición, movimiento, peso, almacenamiento de datos)
- Nuevos procesadores de gama alta
- Todos servicios Transparent Ready de Ethernet TCP/IP: Escaneo de E/S, datos globales, servidor Web, mensajes de correo electrónico, acceso directo a bases de datos, abierto a TCP, Network Time Protocol, etc.

(Schneider E. I., 2020)

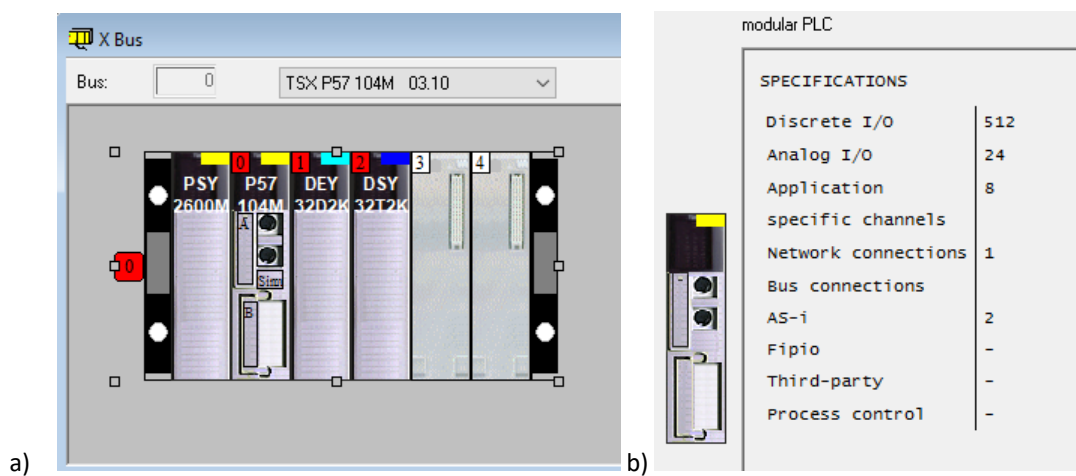


Figura 4: Configuración del X Bus en Unity Pro. a) Rack con sus respectivos módulos b) Módulo del procesador P57 104M

En la Figura 4 podemos observar los módulos que utilizamos en este PLC, se tratan de:

- Una fuente de alimentación: PSY2600M
- Procesador modular: TSX P57 104M
- Modulo de entradas: TSX DEY32D2K (32I 24VDC SINK CONN)
- Modulo de salidas: TSX DSY32T2K (32Q 24VDC 0.1A CONN)

Estos son en realidad los equipos con los que cuenta el laboratorio de automática de la universidad, por lo tanto, hemos decido utilizar los mismos módulos para acercar a la realidad la ejecución del proyecto.

5. DISEÑO DEL PROCESO

Para el diseño de nuestra planta, se han estudiado diferentes procesos de embotellado, en esta ocasión se ha elegido una configuración de línea de 7 estaciones, cada una de las cuales cumple con una función específica, razón por la cual se han instalado siguiendo un orden secuencial y ordenado. Dichas estaciones tienen independencia operacional, pero a su vez cada estación depende de la anterior para poder seguir avanzando hasta llegar al final de la línea. La siguiente Figura 5 muestra el diagrama operacional de la I línea.

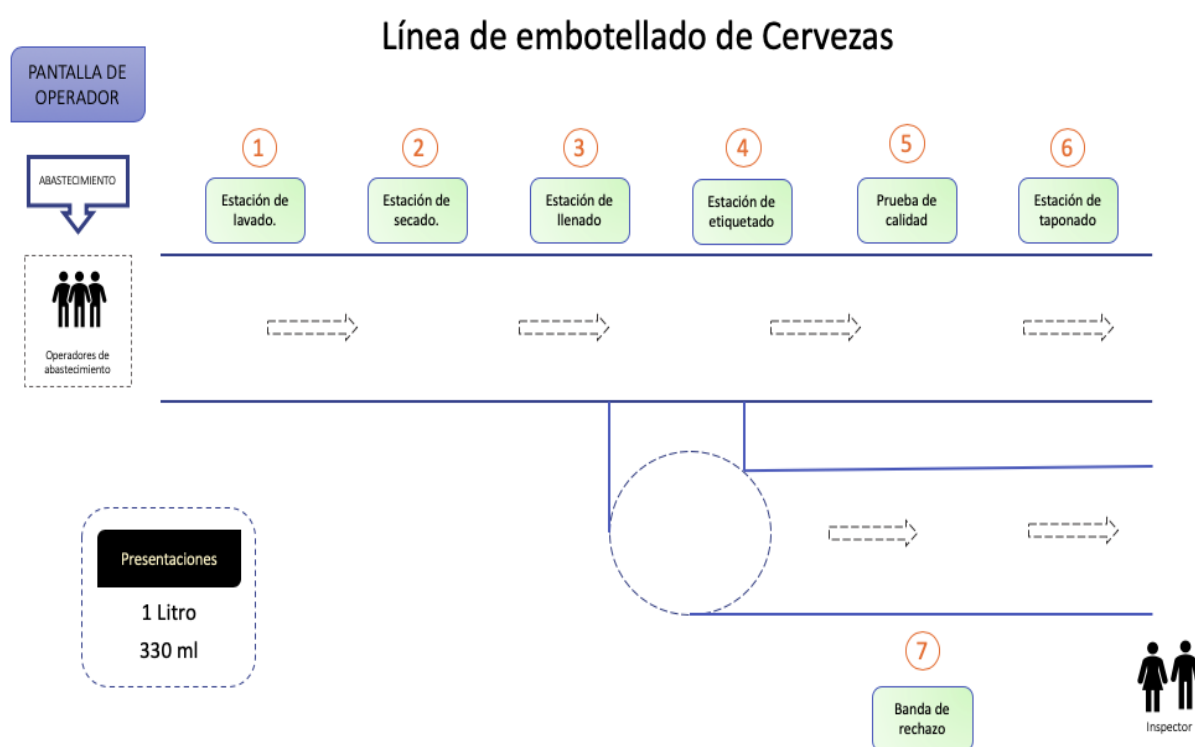


Figura 5: Diseño de planta. Línea de embotellado de cerveza.

Como podemos ver en la Figura 5, el diseño de la planta sigue un modelo de línea donde cada estación de trabajo depende de la anterior para poder llevar a cabo su función, por otro lado, en la Figura 6 podemos observar la configuración en modo de simulación para la planta, en esta interfaz utilizamos las facilidades de edición del software Unity PRO, para simular el proceso lo mas cercano a la realidad.

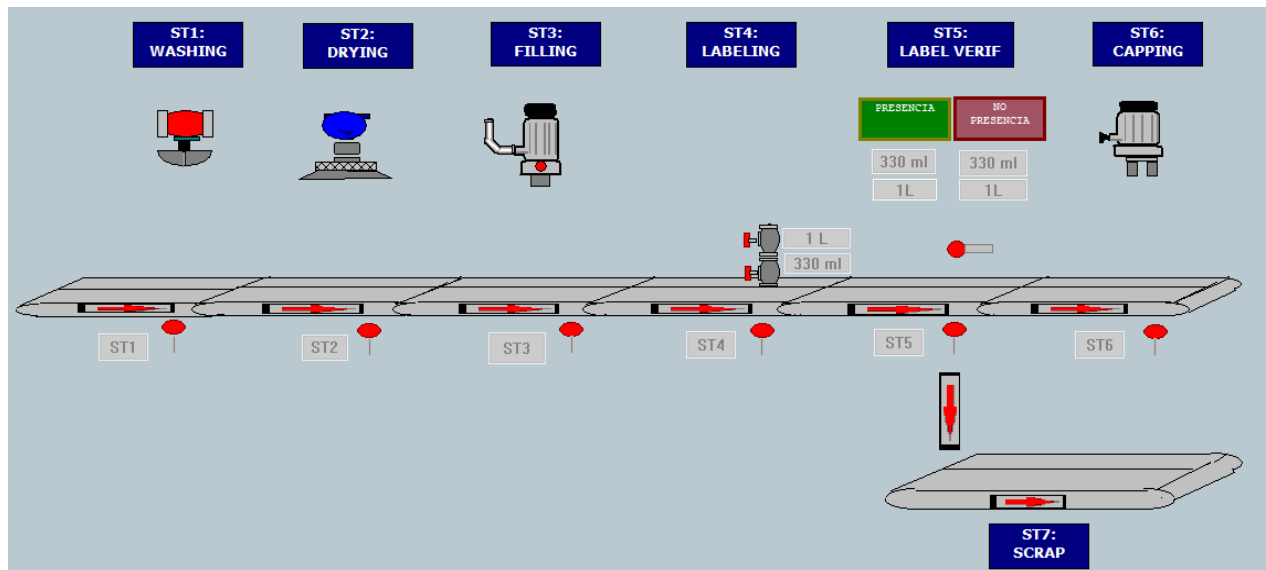


Figura 6: Diseño de planta en modo simulación del Unity Pro.

5.1 Estación de lavado (ST1)

Esta estación es la primera de la línea y se abastece de botellas manualmente según la presentación que se elige en la pantalla de mando, cuando el usuario elige el tipo de botella se inicia con el transporte de la misma por la Banda 1, hasta que el sensor ST1 detecta la presencia de la misma para detener la banda y luego de un retardo de 1 s, se procede con el lavado de la misma, acción que es controlada por una electroválvula cuyo estado activo es temporizado y depende del tamaño de la botella, a mayor tamaño mayor es el tiempo en que la electroválvula se encontrará en estado activo. Cuando se cumple este procedimiento, se procede al transporte de la botella por la siguiente Banda 2 para iniciar con el siguiente paso.

Es importante mencionar que se le han asignado el color rojo para indicar estado inactivo = 0, y color verde para indicar estado activo = 1, como vemos a continuación:

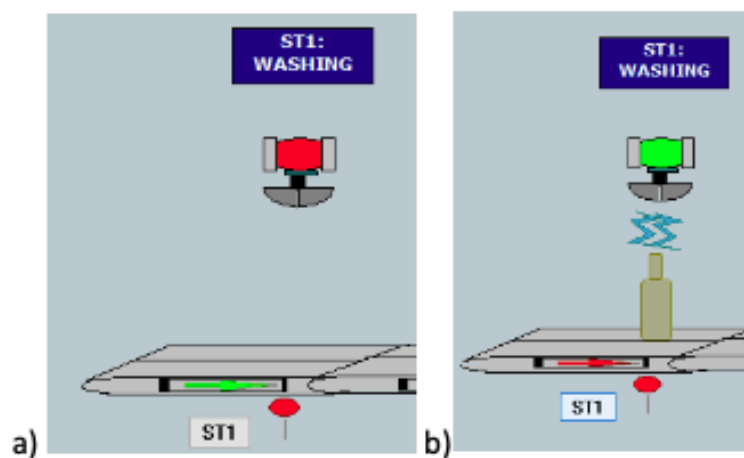


Figura 7: Operación de la ST1. a) Estado inactivo, b) Estado activo

5.2 Estación de secado (ST2)

Como hemos elegido en el diseño que cada estación tenga su propia banda transportadora para hacer el proceso mas ordenado y controlado, la Banda 2 es la encargada de llevar el producto hasta el secador que no es mas que un soplador de aire caliente, el cual tiene como función secar la botella después del proceso de lavado para que llegue no solo desinfectada sino que seca a la estación de llenado, esto porque se deben cumplir las medidas de seguridad y los protocolos sanitarios de calidad que se requieren para poder comercializar el producto.

Una vez que el sensor ST2 detecta la presencia de la botella, detiene la banda para que el secador se active por un tiempo y pueda secar la botella de manera adecuada.

5.3 Estación de llenado (ST3)

Esta es una de las estaciones mas importantes de todo el proceso pues abastece líquido a cada botella que llegue hasta este punto, evidentemente, el tiempo de llenado dependerá del tamaño de la botella que llegue hasta esa estación pues las presentaciones son de 1 Litro y 330 ml, así que los tiempos de apertura de válvula será mayor o menor según la presentación que se haya elegido. Se debe cumplir un tiempo de retardo de 1 s desde que la botella es detectada por el sensor y la apertura de la válvula de llenado para evitar que el líquido se derrame y provoque un

5.4 Estación de etiquetado (ST4)

Una vez que se han llenado las botellas del producto, se pasa a la siguiente estación que es donde se procede con el etiquetado de las mismas, la Banda 4 las transporta hasta la estación, cuando el sensor ST4 detecta la presencia de la botella, un actuador se activa para realizar el etiquetado correspondiente por medio de un cilindro neumático. Un detalle importante a resaltar acá es que dependiendo de cuál botella se está procesando en el momento, se activará uno u otro actuador, esto se desarrolló colocando 2 válvulas colocadas a diferentes alturas para que, según la botella, se active uno u otro actuador. El programa debe ser capaz de actuar según el tamaño de botella para que la orientación de la etiqueta sea siempre en la misma posición, como se aprecia en la siguiente figura.

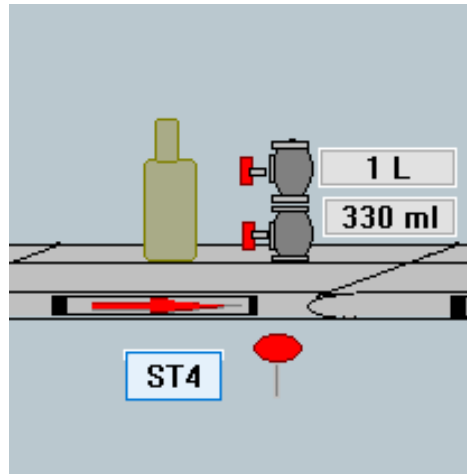


Figura 8: Estación de etiquetado (ST4).

5.5 Estación de Calidad (ST5)

La estación de calidad, tiene como función detectar la presencia o no de la etiqueta en la botella de manera correcta, para ello nos valemos de 2 sensores de presencia y otros 2 de NO presencia, 1 para cada presentación de botella. Si hace una detección correcta de la etiqueta, permite que el proceso continúe a la siguiente estación, caso contrario, si no la detecta activa un actuador de rechazo para colocar la botella en una banda de reproceso de material. La manera en la que se ha decidido configurar en la planta para poder realizar la simulación, es como podemos observar en la Figura 9, de esta manera que se puede hacer una sencilla representación gráfica para habilitar una u otra condición.



Figura 9: Estación ST5 de verificación de etiqueta.

5.6 Estación de taponado (ST6)

Esta estación solo entrará en funcionamiento si la estación de calidad previamente ha permitido el paso de la botella a esta estación, lo cual quiere decir que ha llegado a este punto porque ha detectado de manera correcta la etiqueta y se ha habilitado el paso para que se proceda con la estación final que es la de taponado. Acá, una banda transporta la botella hasta que el sensor ST6 detecta la presencia de la botella y dependiendo de que cual presentación de botella se esté procesando, se activará uno u otro actuador.

Para las botellas pequeñas se requiere que el pistón al ser accionado, recorra mas distancia que la botella grande, por lo que se han independizado para no provocar ningún accidente a la hora de realizar el taponado.

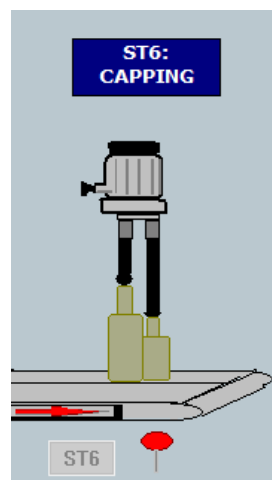


Figura 10: Estación ST7 de taponado.

5.7 Estación de reproceso (ST7)

Esta estación pasará a estado activo si la estación #5 de calidad, no ha detectado la presencia de la botella de manera correcta, por lo que rechaza por medio de un actuador que lleva la botella hasta esta banda. Esta estación realiza un conteo del número de botellas que ha sido reprocesadas, y emite una alerta en el momento que una cierta cantidad de botellas ha sido rechazada en la estación ST5, acción que evita el excesivo reproceso por alguna falla ya que detiene el proceso. Esto permite que los operarios de la línea puedan volver a llevar dicho material de rechazo a iniciar de nuevo el proceso desde el inicio.

5.8 Tanque de abastecimiento

Adicionalmente, se ha diseñado el proceso de abastecimiento del líquido por medio de un tanque con control de mando para el usuario, en la que se ha habilitado dos modos de operación:

- A) **Modo llenado:** Este modo inicia su funcionamiento en el momento que el operador presione el botón de inicio. En el primer instante abre una electroválvula por lo que el tanque se empieza a llenar con el líquido hasta que el sensor de nivel alto (del tipo ultrasónico) detecte que el tanque se ha llenado. Es importante mencionar que este sensor se ha colocado de manera estratégica, de modo solo permita al tanque llenarse al 90% de su capacidad, esto para evitar un posible derrame de líquido o algún problema similar.

Una vez que alcanza este nivel, deshabilita la válvula de llenado, el líquido se comienza a consumir hasta que un sensor de nivel bajo, el cual se ha colocado de manera estratégica para que cuando alcance el 20% del tanque, vuelva a abrir la válvula de llenado hasta que nuevamente alcance el nivel alto del tanque.

Después de 3 ciclos de operación donde se llena y se vacía el tanque, se termina el ciclo para proceder con una tarea de mantenimiento, esto por la naturaleza de la cerveza que después de un tiempo podría generar algún sedimento dentro del tanque, y es algo que se debe evitar para asegurar la calidad del producto.

Todos los sistemas y circuitos de cerveza requieren de un mantenimiento y limpieza para evitar contaminaciones por levadura o acumulación de sedimentos, además de mantener siempre la frescura y sabor originales que provienen del barril de cerveza. La suciedad provoca la alteración del sabor real de la cerveza que se dispensa, por tanto, cuanto más limpio mantengamos el circuito por donde pasa la cerveza, más puro será el sabor que proviene del barril. (Marimon S, 2015)

- B) **Modo de vaciado:** Este modo de operación se utiliza si se quiere vaciar el tanque en algún momento en que se requiera alguna labor de mantenimiento correctivo o similar, en la que se debe detener el proceso de manera inmediata. Este ciclo activa una válvula de vaciado la cual tiene un indicador visual en pantalla para que el operador sepa que está activo este modo.

Para que el modo de vaciado pueda habilitarse, se debe cumplir que el tanque tenga líquido mayor al 20% y menor o igual al 90 % del tanque. El proceso de vaciado termina en el momento que un sensor detecta que el tanque tiene un 0% de líquido en su interior, el cual se ha colocado de manera estratégica en esta posición.

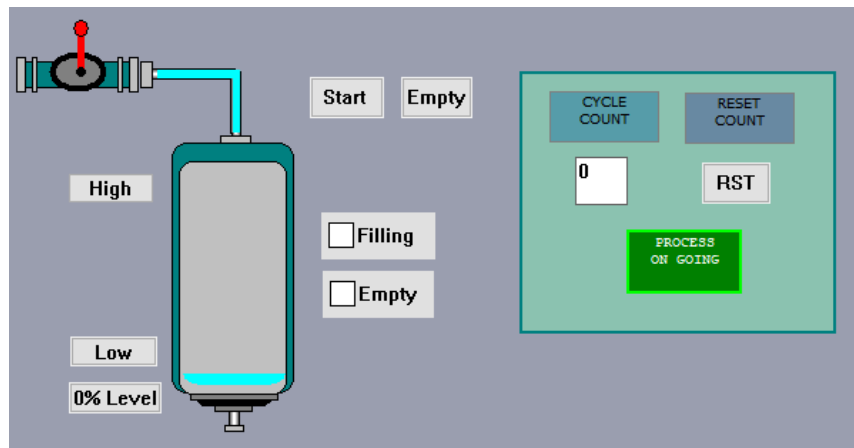


Figura 11: Modo de simulación del Tanque de abastecimiento, con su respectivo panel de control.

6. DESARROLLO DEL CÓDIGO

Esta sección tiene como finalidad mostrar algunos detalles importantes del desarrollo del código que se ha hecho. Como se había mencionado en la sección 2, el objetivo era desarrollar el código en lenguaje SFC (Sequential Function Chart) el cual cumple con la norma IEC 61131-3, y que nos facilita de gran manera la implementación del proceso diseñado. Tal y como lo asegura Álvaro Orozco G y Mauricio Holguín L. en su trabajo de investigación sobre SFC para simulación y automatización de procesos: *“Dentro de las técnicas de mayor prestación en automatización se encuentra SFC, la cual gracias a funcionalidades tales como descripción en forma de eventos discretos, acciones de forzado, control de acciones concurrentes y divergentes se ha convertido en estándar”*. (Orozco G., 2009)

Nuestro programa consta de 4 secciones de SFC los cuales están integrados en el modo de operación MutiToken, el cual nos permite tener habilitados varios procesos al mismo tiempo, estas secciones son:

1. **Main:** Representa el ciclo de trabajo principal, en el se encuentran contenidos todos los procesos de la línea.
2. **Tank:** Proceso de operación del tanque, el cual se opera de manera independiente al Main y tiene su propio control con interfaz de usuario propia.
3. **Emergency:** Este ciclo SFC es el que contiene las acciones a realizar en el momento en que el botón de paro de emergencia es activado.
4. **Reset_CTDS:** Este SFC tiene como función principal la acción de reinicio en el momento que se activa el botón de RESET vía manual.

6.1 Main

Primeramente, se debe mencionar que el proceso tiene una variación en el tipo de botella que se va a procesar, en donde, por un lado, tenemos la presentación de 1L al que llamaremos *Proceso A* y por otro la de 330 ml que sería el *Proceso B*, ambos procesos deben utilizar el mismo sensor de detección, pero al mismo tiempo el programa debe ser capaz de interpretar cual botella se está procesando en el momento, gracias a la opción de selección que se habilita en la pantalla de operador.

Este proceso utiliza la función MACROSTEP que nos permite encapsular procesos lineales, con el fin de que nuestro SFC se configure en una manera mas ordenada y permita de una manera mas sencilla la detección de fallas. Por otro lado, se debe destacar que este SFC cuenta con todas las funciones de decisión que el programa debe tomar según la información de entrada que provean los sensores.

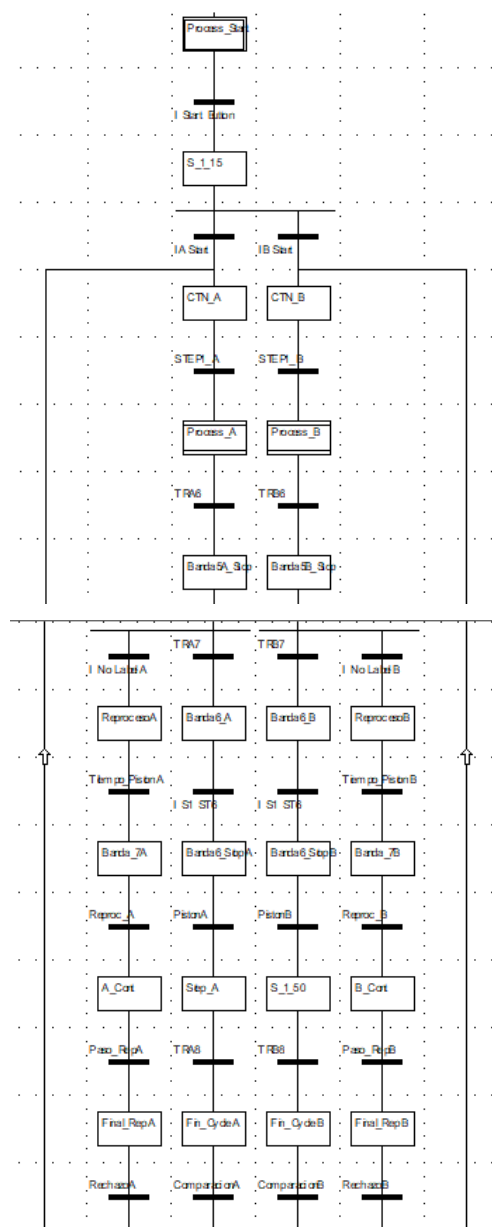


Figura 12: SFC de la función MAIN.

También, como se ha mencionado anteriormente se ha hecho uso de las funciones *MACROSTEP* para programar una sección de código que incluye un proceso lineal entre las 1 y 4 de la planta. Esto hace mucho mas sencillo el proceso a nivel visual y secuencial, de manera que podemos detectar de una manera mucho mas fácil una posible falla. El proceso, lo observamos en la Figura 13.

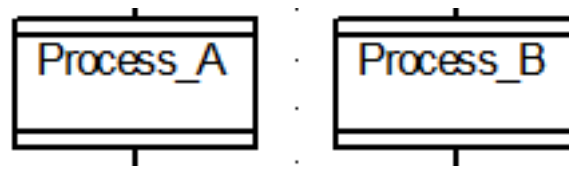


Figura 13: MACROSTEPS para procesos A y B. (Representación simbólica).

Mientras que Figura 14 muestra el contenido del MACROSTEP el cual es igual tanto para el proceso A como B, como vemos a continuación.

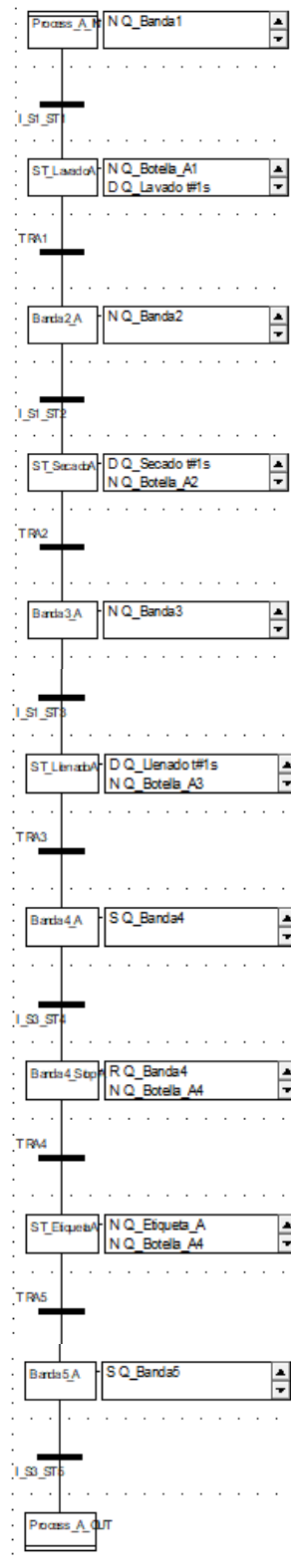


Figura 14: MACROSTEP para el proceso A

6.1.1 Configuración del Proceso MAIN

Para poder dotar de funcionalidad operativa al proceso principal se han configurado tanto acciones como transiciones. Cada acción representa una actividad que se ejecuta en el momento que el paso está habilitado. Según la configuración que se ha programado, el programa nos muestra la siguiente estructura de acciones (ver Figura 15) y transiciones (ver Figura 16):

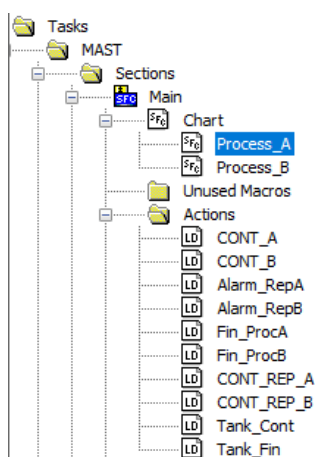


Figura 15: Acciones programadas en la sección MAIN.

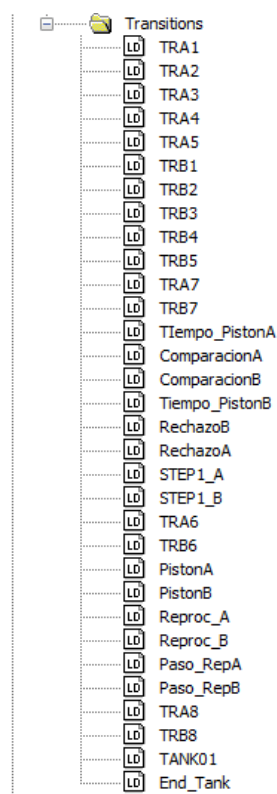


Figura 16: Transiciones programadas en la sección MAIN

La mayoría de transiciones representan entradas que se producen por alguna combinación de sensores y/o pulsadores manuales. Mientras que las acciones principales, están ligadas a tareas de conteo, activación de alarmas y condiciones de final de proceso.

6.1.2 Uso de Contadores

A manera de ejemplos didácticos esta sección describe algunas disposiciones importantes con respecto a la programación de contadores, los cuales se usan para poder tener registro de la cantidad de producto que se elabora, misma manera que se hace con el reproceso, el cual debe activar una alarma cuando se ha superado el límite de material a reprocesar, ya que nos indica que hay alguna anomalía que se detecta en la estación de calidad que hace que el material sea rechazado. Estos contadores se han programado con la función OPERATE como lo vemos en la siguiente Figura 17:

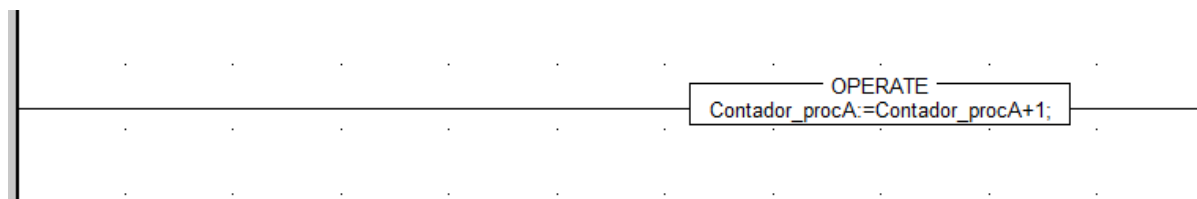


Figura 17: Función contadora CONT_A

En el caso de CONT_A, se finaliza el conteo cuando la función comparadora detecta que cierta cantidad se ha cumplido y se detiene el proceso. Estas cantidades deben ser elegidas según la cantidad de botellas que se quieran procesar. El proceso real de embotellado tiene presentaciones estándar comerciales las cuales representan cajas con 6, 12 y 24 cervezas, lo anterior lo vemos en la siguiente Figura 18:

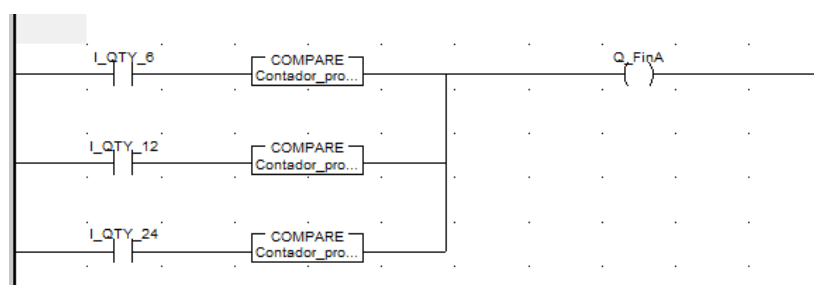


Figura 18: Función Fin_Proc_A

6.1.3 Uso de Temporizadores

En este caso Unity PRO nos permite hacer uso de diferentes funciones temporizadas, las cuales las podemos definir en los cada uno de las etapas del SFC, por medio de funciones de retardo declaradas directamente sobre cada acción que se realice. Para ello el programa nos facilita las siguientes variables:

- **N:** Se ejecuta la acción programada en la sección (o pone a "1" la variable asignada) durante el tiempo que está activa la etapa.
- **P1:** Se ejecuta la acción programada en la sección (o pone a "1" la variable asignada) cuando la etapa pasa de inactiva a activa (acción al activar).
- **P0:** Se ejecuta la acción programada en la sección (o pone a "1" la variable asignada) cuando la etapa pasa de activa a inactiva (acción al desactivar).
- **S:** Pone a "1" la variable asignada.
- **R:** Pone a "0" la variable asignada.
- **L:** Acción con tiempo limitado, sólo se ejecuta durante el tiempo definido.
- **D:** Acción retrasada, se ejecutará un tiempo después de activada la etapa.

Figura 19: Variables temporizadas para las acciones.

En nuestro código se ha utilizado en algunas funciones como la que vemos la Figura 20, en la que se hace un retardo temporizado sobre la acción *Q_Secado*, lo cual quiere decir que una vez que se active la etapa *ST_SecadoA*, la *Q_Secado* deberá transcurrir un tiempo de 1 s para que se pueda habilitar dicha salida.

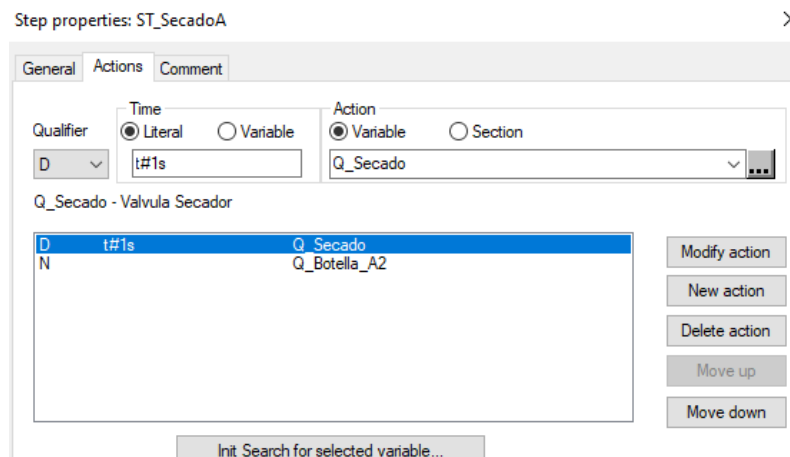


Figura 20: Acción de retardo temporizado para la salida *Q_Secado*.

Otra forma en la que se ha utilizado la función temporizada es en las transiciones, como lo vemos en la Figura 21 la transición se activará cuando la etapa anterior haya sido activada por un tiempo que definimos por medio de una acción compradora. En este caso la transición llamada *ComparacionA* se activará hasta que transcurran 500ms. Esto lo declaramos por medio de la expresión *Fin_CycleA.t>t#500ms*

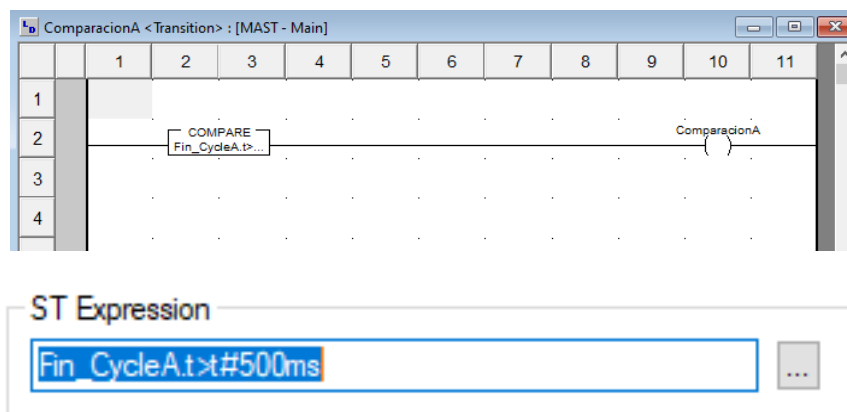


Figura 21: Acción temporizada en una Transición.

Existen otras formas de temporizar una acción por medio del uso de bloques FFB (Function Blocks) de temporizadores de cuenta regresiva o progresiva, así mismo como dentro de la configuración de etapas podemos definir acciones de Máximo, Mínimo y Retardos.

6.2 Tank

Como se explicó al inicio de esta sección, se ha programado un SFC exclusivo para la operación del tanque, el cual es el encargado de abastecer de cerveza en la estación de llenado. Es la [Sección 5.8](#) se había detallado todo lo correspondiente a la operación de este proceso, sin embargo, en esta sección se detallan circunstancias ligadas a la programación.

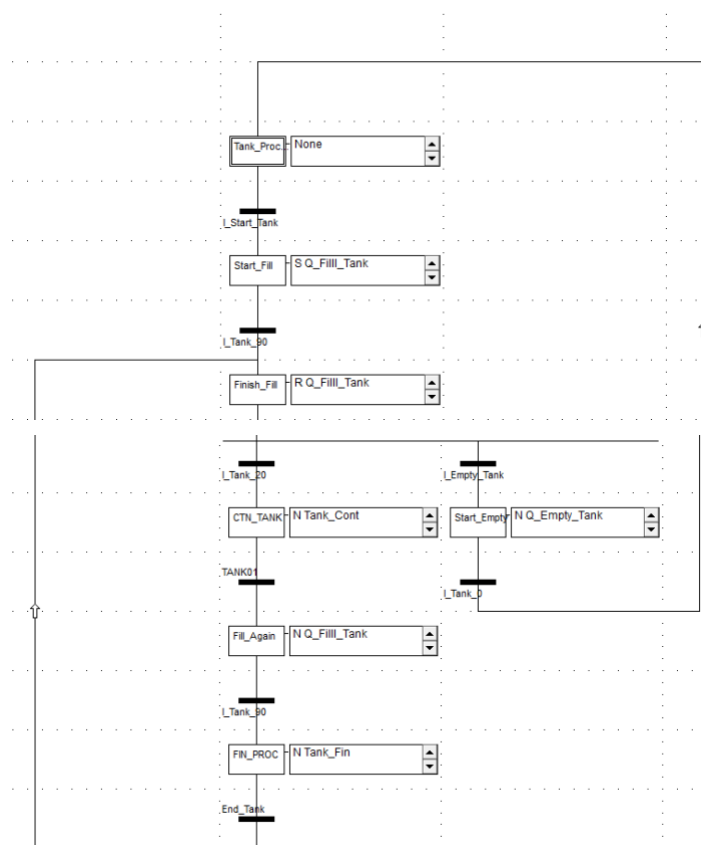


Figura 22: Diagrama SFC operación del tanque.

En la Figura 22, podemos apreciar que se establecieron dos procesos, uno para llenar el tanque y mantenerlo con líquido durante 3 ciclos, y otro modo para proceder con el vaciado anticipado para cuando se requiera hacer alguna labor de mantenimiento en el que se deja sin líquido el interior del tanque. De igual manera se han programado tareas que requieren contadores y temporizadores como en el caso de la sección MAIN.

6.3 Emergency STOP

El paro de emergencia se activa al presionar la seta y esto hace que todo el proceso se detenga, lo cual significa que todas las variables de salida, actuadores, motores, válvulas y cilindros neumáticos se desenergicen de inmediato una vez presionada la seta. Esto también tiene una implicación cuando se vuelve a habilitar la seta, y es que debe permitir que se inicie el ciclo desde el principio del proceso.

Es importante mencionar que hemos preparado una nueva sección dentro de la tarea maestra “MAST” a la que hemos llamado EMERGENCY, la cual esta dedicada al manejo de esta tarea, así como la de Reset de los Contadores que analizaremos en la siguiente sección. A continuación, observaremos la programación en la siguiente Figura 23:

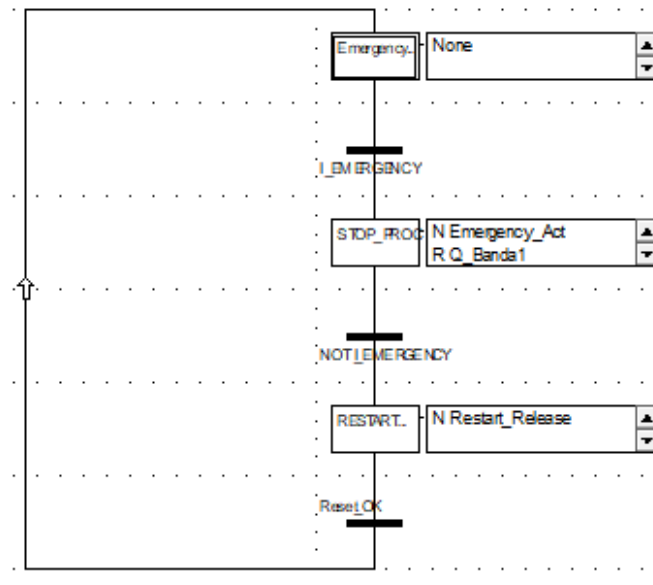


Figura 23: SFC para el manejo del paro de emergencia.

Como podemos apreciar, se trata de un ciclo sencillo en el que habilita la acción a la que hemos llamado *Emergency_Act* en el momento que se pulse la seta de emergencia. Esta tarea se ha configurado de la siguiente manera:

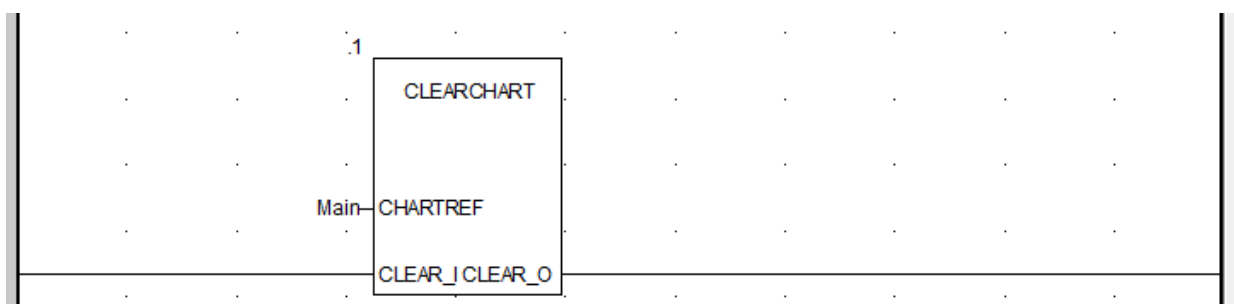


Figura 24: Acción *Emergency_Act*

Donde se utiliza una función llamada *CLEARCHART*, la cual nos permite detener una sección o una secuencia cuando se pone a 1 la señal, para el caso nuestro hemos puesto que el *CHARTREF* detenga nuestra función *MAIN*, el cual contiene la programación de nuestro proceso. Al mismo tiempo que habilitamos una función *R* (reset) para todas las variables de salida, como lo vemos en la siguiente Figura 25:

R	Q_Banda1
R	Q_Banda2
R	Q_Banda3
R	Q_Banda4
R	Q_Banda5
R	Q_Banda6
R	Q_Banda7
R	Q_Empty_Tank
R	Q_Etiqueta_A
R	Q_Etiqueta_B
R	Q_Fill_Tank
R	Q_Lavado
R	Q_Llenado
R	Q_Piston_Rechazo
R	Q_Secado
R	Q_TaponA
R	Q_TaponB

Figura 25: Acción de reinicio de las variables de salida

Posteriormente, cuando se vuelva a lanzar la seta de emergencia, se habilita la acción llamada *Restart_Release* la cual contiene un bloque de *INITCHART* encargado de volver a iniciar la sección que se seleccione, en nuestro caso sería la función principal *MAIN*, mientras que el *SETSTEP* nos permite escoger en cual etapa queremos que se inicie la puesta en marcha, en donde elegimos nuestra etapa inicial llamada *Process_Start*, lo anterior lo podemos observar en la siguiente Figura 26:

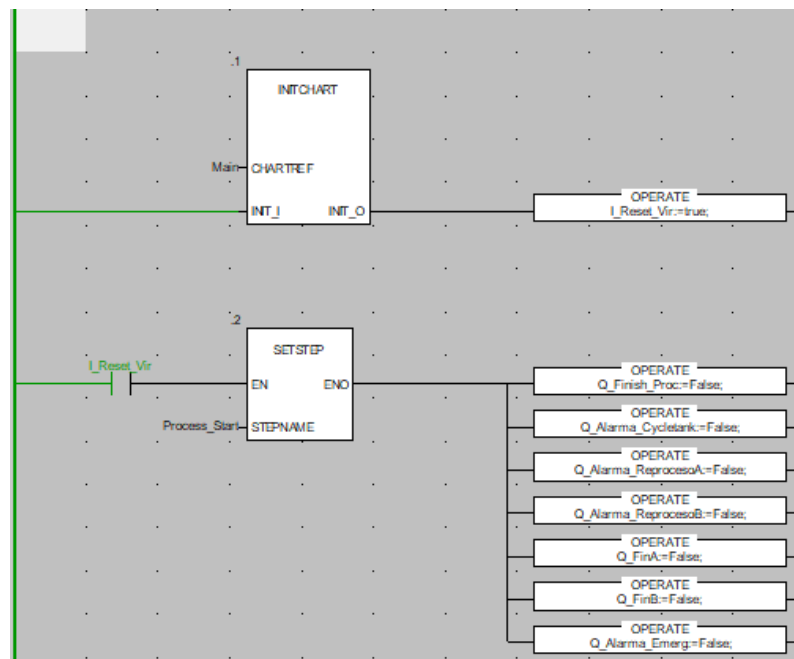


Figura 26: Acción *Restart_Release*.

6.4 Reset CTD's

Por último, tenemos nuestro SFC para la acción reiniciadora de todos los contadores con que cuenta el sistema, que básicamente controlamos por medio de la función *OPERATE*, en donde establecemos cuales son los contadores que queremos poner a 0. Se debe resaltar que existen dos Resets del sistema, uno para el proceso principal y otro para el contador del tanque. La anteriormente dicho lo hemos configurado de la siguiente manera:

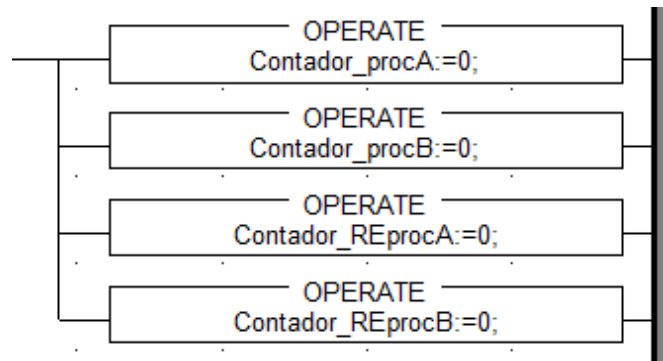


Figura 27: Acción Reset_Cnts para reinicio de los contadores del panel principal de operación.

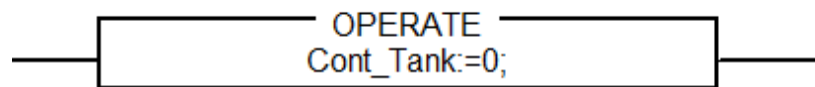


Figura 28: Acción RST_TANK_CTD

La acción de reinicio de contadores del tanque, actúa directamente sobre la variable entera de que cuenta la cantidad de ciclos de llenado que realiza el tanque.

7. PANEL DE CONTROL DE OPERADOR

En este proyecto se han habilitado dos paneles de operador, uno para el proceso principal y otro para el proceso del tanque de abastecimiento, como podemos observarlos la Figura 29 y 30 tienen la siguiente forma:

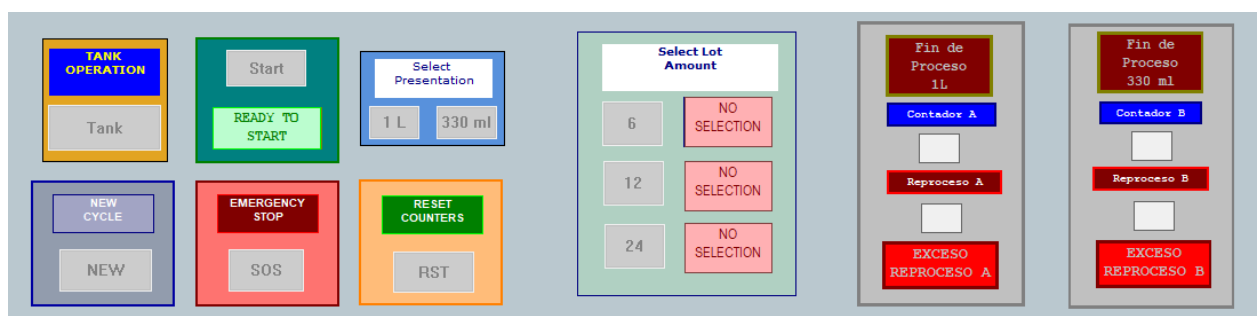


Figura 29: Panel de Control de Operador

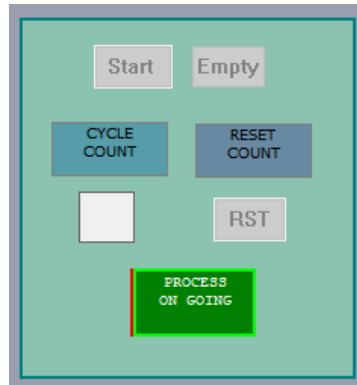


Figura 30: Panel de operación del tanque de abastecimiento.

Para el caso del panel principal, debemos señalar que es el control de mando que tiene el operador, el cual permite operar el sistema según la necesidad.

7.1 Pantalla principal de operador

El procedimiento para poner la puesta en marcha del sistema utilizando la pantalla de operador, se ha diseñado de manera que siga un patrón muy intuitivo, en donde cada uno de los elementos de esta pantalla permitan seleccionar las características del proceso que se quiera aplicar, en la Figura 31, vemos cuatro elementos que corresponden a: Botón de Inicio de Ciclo, Selección de Presentación de Botella, Selección de Cantidad de botellas y acceso a la pantalla del tanque.

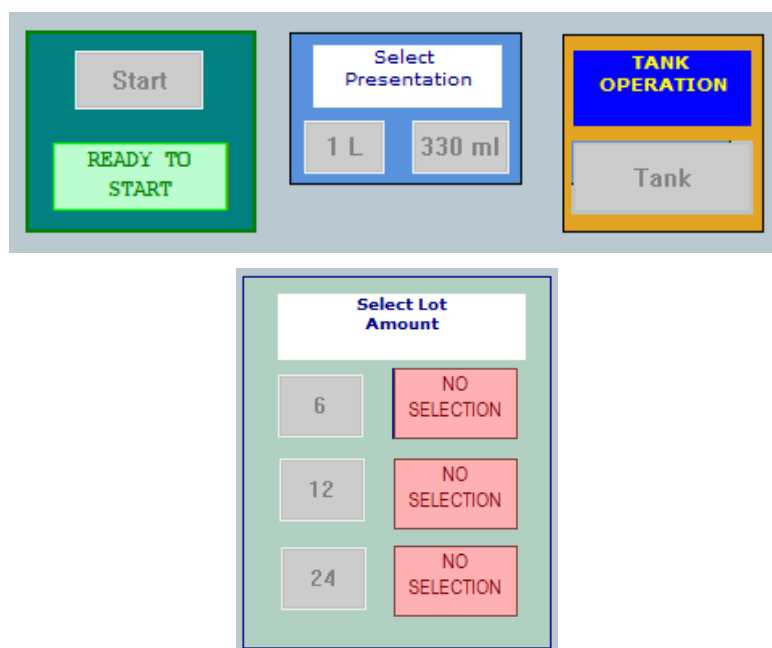


Figura 31: Opción de inicio y selección en pantalla.

La selección de la cantidad de botellas que se quieren procesar se ha descrito en la [sección 6.1.2](#), en donde se había dado énfasis en que comercialmente se ha normalizado esas presentaciones de cajas de cervezas en los establecimientos comerciales. Por otro lado, tenemos el panel para visualizar la cantidad de producto procesado, así mismo como la cantidad de reproceso, mismo que podemos apreciar en la Figura 32;

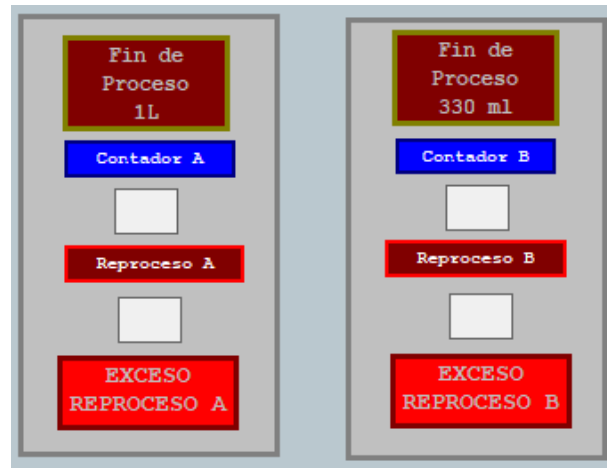


Figura 32: Panel de visualización de material procesado y de reproceso.

Otra función que podemos ver en esta sección son los indicadores visuales para dos acciones:

- Cuando termina un ciclo de proceso.
- Cuando hay exceso de reproceso.

7.2 Pantalla de operación del tanque de abastecimiento

Esta pantalla nos posibilita monitorear la actividad del tanque de abastecimiento de cerveza que permite el paso de líquido en la estación de llenado (ST3). En ella, podemos encontrar los botones de Inicio de sesión que se habían estudiado en la sección 6.2, en donde después de algunos ciclos de funcionamiento el proceso se detiene para permitir las rutinas de mantenimiento programado.

Mientras que, nos da la opción visual de monitorear la cantidad de ciclos que se han cumplido, y junto a este visualizador se encuentra botón de RST que permite el reinicio de la cuenta descrita en la sección 6.4. De igual manera se muestra una opción visual para poder mirar cuando el proceso está en marcha o bien cuando se ha detenido.

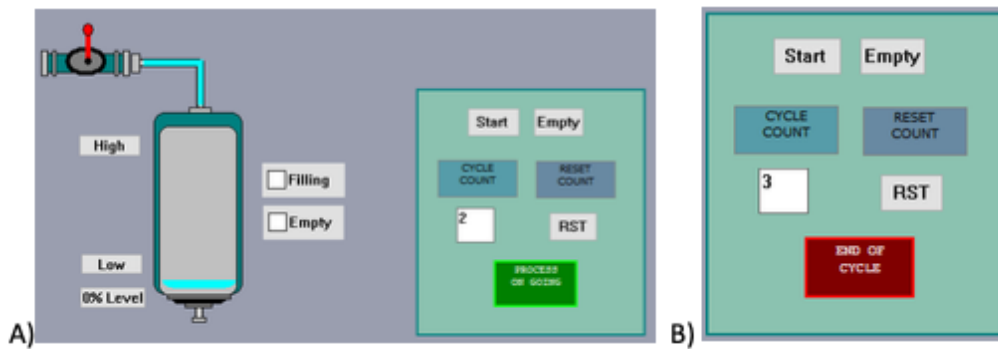


Figura 33: Modo de operación del tanque de abastecimiento. A) Conteo de ciclos de operación B) Mensaje de Fin de ciclo.

8. VARIABLES GENERADAS

El programa Unity PRO, guarda un registro con las variables que generamos, estas tienen que ver con las entradas, salidas, etapas, transiciones, entre otras. En este proyecto las variables que generaron fueron identificadas con la inicial I_ para las entradas (Ver Figura 34), Q_ para las salidas (Ver Figura 35):

● I_EMERGENCY	EBOOL	Boton de emergencia
● I_Empty_Tank	EBOOL	Boton vaciar tanque
● I_No_LabelA	EBOOL	No detecta etiqueta A
● I_No_LabelB	EBOOL	No detecta etiqueta B
● I_QTY_6	EBOOL	Selección 6 unidades
● I_QTY_12	EBOOL	Selección 12 unidades
● I_QTY_24	EBOOL	Selección 24 unidades
● I_Reset_ContTank	EBOOL	Boton para resetear cntd tank
● I_Reset_CTD	EBOOL	Reset Contadores
● I_S1_ST1	EBOOL	Sensor 1 ST 1
● I_S1_ST2	EBOOL	Sensor 1 ST 2
● I_S1_ST3	EBOOL	Sensor 1 ST 3
● I_S1_ST4	EBOOL	Sensor 1 ST 4
● I_S1_ST5	EBOOL	Sensor 1 ST 5
● I_S1_ST6	EBOOL	Sensor 1 ST 6
● I_S2_ST4	EBOOL	Sensor 2 ST 4
● I_S2_ST5	EBOOL	Sensor 2 ST 5
● I_S2_ST6	EBOOL	Sensor 2 ST 6
● I_S3_ST4	EBOOL	Sensor 3 ST 4
● I_S3_ST5	EBOOL	Sensor 3 ST 5
● I_Start_Button	EBOOL	Boton de Inicio de Sistema
● I_Start_Tank	EBOOL	Boton Inicio de Tanque
● I_Tank_0	EBOOL	Sensor level 0%
● I_Tank_20	EBOOL	Sensor level 20%
● I_Tank_90	EBOOL	Sensor level 90%
● IA_Start	EBOOL	Inicio proceso A
● IR_Start	FROOI	Inicio Proceso R

Figura 34: Entradas del sistema.

● Q_Alama_Cycletank	EBOOL	Alama p/ termino el ciclo tanque
● Q_Alama_Emerg	EBOOL	Alama de paro de emergencia
● Q_Alama_ReprocesoA	EBOOL	Exceso de reproceso A
● Q_Alama_ReprocesoB	EBOOL	Exceso de reproceso B
● Q_Banda1	EBOOL	Motor Banda 1
● Q_Banda2	EBOOL	Motor Banda 2
● Q_Banda3	EBOOL	Motor Banda 3
● Q_Banda4	EBOOL	Motor Banda 4
● Q_Banda5	EBOOL	Motor Banda 5
● Q_Banda6	EBOOL	Motor Banda 6
● Q_Banda7	EBOOL	Motor Banda 7
● Q_Botella_A1	EBOOL	Simulacion Botella A1
● Q_Botella_A2	EBOOL	Simulacion Botella A2
● Q_Botella_A3	EBOOL	Simulacion Botella A3
● Q_Botella_A4	EBOOL	Simulacion Botella A4
● Q_Botella_A5	EBOOL	Simulacion Botella A5
● Q_Botella_A6	EBOOL	Simulacion Botella A6
● Q_Botella_A7	EBOOL	Simulacion Botella A7
● Q_Botella_B1	EBOOL	Simulacion Botella B1
● Q_Botella_B2	EBOOL	Simulacion Botella B2
● Q_Botella_B3	EBOOL	Simulacion Botella B3
● Q_Botella_B4	EBOOL	Simulacion Botella B4
● Q_Botella_B5	EBOOL	Simulacion Botella B5
● Q_Botella_B6	EBOOL	Simulacion Botella B6
● Q_Botella_B7	EBOOL	Simulacion Botella B7
● Q_Empty_Tank	EBOOL	Valvula Vaciado
● Q_Etiqueta_A	EBOOL	Piston de Etiqueta A
● Q_Etiqueta_B	EBOOL	Piston de Etiqueta B
● Q_Fill_Tank	EBOOL	Valvula Llenado
● Q_FinA	EBOOL	Fin Proceso A
● Q_FinB	EBOOL	Fin Proceso B
● Q_Finish_Proc	EBOOL	Mensaje tanque vacio
● Q_Lavado	EBOOL	Valvula Lavado
● Q_Llenado	EBOOL	Valvula Llenado
● Q_Piston_Rechazo	EBOOL	Piston rechazo
● Q_Process_Ready	EBOOL	Mensaje proceso listo para iniciar
● Q_Secado	EBOOL	Valvula Secador
● Q_TaponA	EBOOL	Taponadora A
● Q_TaponB	EBOOL	Taponadora B

Figura 35: Salidas del sistema.

Las variables del tipo INT (entero), fueron las que se han utilizado para el manejo de contadores, las mismas las podemos observar en la siguiente Figura 36:

● Cont_Tank	INT	Ciclos de llenado del tanque
● Contador_procA	INT	Cuenta proceso A
● Contador_procB	INT	Cuenta proceso B
● Contador_REprocA	INT	Contador de reproceso A
● Contador_REprocB	INT	Contador de reproceso B

Figura 36: Variables del tipo INT utilizadas en el programa.

Por otro lado, también se ha generado una serie de etapas cada una con función dedicada a activar alguna salida ante una combinación de entrada, o un pulso de sensor. Éstas fueron:












































+		A_Cont	SFCS...
+		B_Cont	SFCS...
+		Banda2_A	SFCS...
+		Banda2_B	SFCS...
+		Banda3_A	SFCS...
+		Banda3_B	SFCS...
+		Banda4_A	SFCS...
+		Banda4_B	SFCS...
+		Banda4_StopA	SFCS...
+		Banda4_StopB	SFCS...
+		Banda5_A	SFCS...
+		Banda5_B	SFCS...
+		Banda5A_Stop	SFCS...
+		Banda5B_Stop	SFCS...
+		Banda6_A	SFCS...
+		Banda6_B	SFCS...
+		Banda6_StopA	SFCS...
+		Banda6_StopB	SFCS...
+		Banda_7A	SFCS...
+		Banda_7B	SFCS...
+		CTN_A	SFCS...
+		CTN_B	SFCS...
+		CTN_TANK	SFCS...
+		EMERGENCY	SFCC...
+		Emergency_Proc	SFCS...
+		Fill_Again	SFCS...
+		Fin_CycleA	SFCS...
+		Fin_CycleB	SFCS...
+		FIN_PROC	SFCS...
+		Final_RepA	SFCS...
+		Final_RepB	SFCS...
+		Finish_Fill	SFCS...
+		Main	SFCC...
+		Process_A	SFCS...
+		Process_A_IN	SFCS...
+		Process_A_OUT	SFCS...
+		Process_B	SFCS...
+		Process_B_IN	SFCS...
+		Process_B_OUT	SFCS...
+		Process_Start	SFCS...
+		ReprocesoA	SFCS...
+		ReprocesoB	SFCS...
+		Reset_act	SFCS...
+		Reset_ctd	SFCS...
+		Reset_CTDS	SFCS...
+		RESTART_PROC	SFCS...
+		S_1_15	SFCS...
+		S_1_50	SFCS...
+		ST_EtiquetaA	SFCS...
+		ST_EtiquetaB	SFCS...
+		ST_LavadoA	SFCS...
+		ST_LavadoB	SFCS...
+		ST_LlenadoA	SFCS...
+		ST_LlenadoB	SFCS...
+		ST_SecadoA	SFCS...
+		ST_SecadoB	SFCS...
+		Start_Empty	SFCS...
+		Start_Fill	SFCS...
+		Step_A	SFCS...
+		STOP_PROC	SFCS...
+		Tank_Process	SFCS...
...			

Figura 37: Etapas del sistema

9. CONCLUSIONES

El desarrollo de este proyecto ha supuesto una amplia investigación en el campo y por supuesto, en el uso del software Unity PRO XL, en donde se pudo corroborar la gran cantidad de funciones, operaciones y facilidades que ofrece este software de automatización. Con la finalización del proyecto podemos decir que hemos alcanzado los objetivos que nos hemos propuesto.

- En primer lugar, se ha alcanzado el principal objetivo que nos hemos proyectado, el cual era automatizar el proceso de línea de embotellado de cervezas haciendo uso del PLC Modicom TSX Premium y el software de edición Unity PRO XL y su modo de simulación, aplicando el lenguaje de programación SFC (Grafcet), haciendo uso de las funciones propias de dicho software para la optimización de dicho proceso.
- Se consigue con éxito el desarrollo de una aplicación automática que controle el proceso de fabricación descrito, dicha aplicación con su respectivo modo de operación y control por pantalla de usuario, para que el mismo tenga la posibilidad de tener el control absoluto de lo que sucede en la línea de producción.
- Se logra implementar el modo de simulación del proceso haciendo uso de las herramientas gráficas tanto de programación como de animación de los procesos diseñados.
- Se adquieren destrezas en el desarrollo tanto en el diseño como en programación, gracias al conocimiento obtenido de las herramientas y funciones que ofrece el programa Unity Pro.
- Se reproduce los resultados de los procesos en interfaces visuales que permiten observar el estado del proceso a lo largo de la evolución sobre la puesta en marcha del mismo.

10. BIBLIOGRAFÍA

1. Antúnez , F. S. (2016). Puesta en Marcha de Sistemas de Automatización Industrial. Málaga, España: IC Editorial.
2. INEGI. (2017). Estadísticas a propósito de la actividad de la elaboración de cerveza. Ciudad de México: INEGI.
3. Marimon S, C. (9 de Setiembre de 2015). Obtenido de Install Beer: <https://installbeer.com/blogs/diariocervecero/51334533-manual-de-instrucciones-de-limpieza-con-tanqueta>
4. Orozco G., A. H. (2009). Descripción e Implementación de SFC para Simulación de Automatismos. Pereira, Colombia: Universidad Tecnológica de Pereira.
5. Schneider. (13 de May de 2020). Schneider Electric. Obtenido de Manual de Usuario: <https://www.se.com/cl/es/product-range-presentation/548-unity-pro/>
6. Schneider, E. I. (14 de May de 2020). Life is On. Obtenido de Modicon Premium PLC: <https://www.se.com/co/es/product-range-presentation/537-modicon-premium/>