

LABVIEW

FASE 2 DESARROLLAR SISTEMA DE SUPERVISIÓN Y
CONTROL

PRESENTADO POR
DÍDIER JAVIER RAMÍREZ HENAO

PRESENTADO A
ALEXÁNDER FLÓREZ

GRUPO
208055_1

OCTUBRE 03 DE 2021
MANIZALES – CALDAS

INTRODUCCIÓN

Gracias al avance tecnológico y científico, también se ha podido avanzar en los sistemas de control automático de los procesos, sistemas que con el paso del tiempo es más fácil de encontrar en los diversos ámbitos de nuestras vidas. Ejemplo de ello son los sistemas de control y supervisión de las naves que son enviadas al espacio o en la industria donde existen procesos de fabricación que son modernos, en sistemas de control de tráfico e incluso en el aumento considerable de los hogares que empiezan a convertir sus casas en hogares inteligentes y que desde su celular o computador puede saber cómo se encuentra e incluso pueden modificar dichas variables.

Ahora, el sistema de control y supervisión sería obsoleto si no se pudiera analizar en tiempo real la forma en que cambian las variables de interés, ya que, de ser así, las soluciones a implementar en muchos casos serían tardías y casi siempre se requiere tomar acciones en el momento indicado para evitar daños en las máquinas y equipos que hacen parte de un determinado proceso y/o en los productos con que se está trabajando.

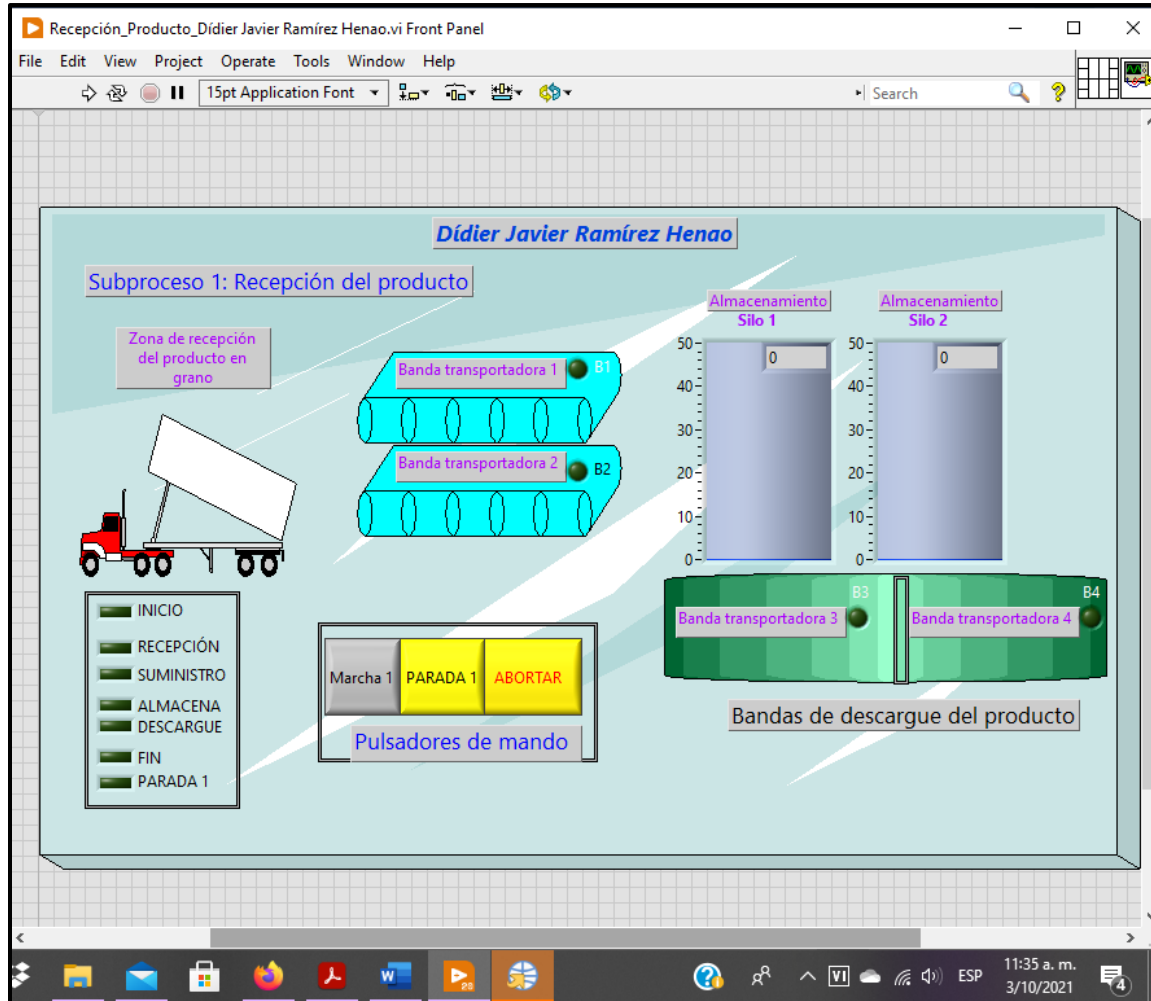
Debido al desarrollo en los sistemas SCADA, se tiene la posibilidad de supervisar e incluso controlar diversas variables dentro de un proceso, para lo cual es necesario contar con el hardware y software requeridos para interactuar tanto con el proceso mismo como para que el operario de la planta pueda observar el estado de las variables, observar su funcionamiento, darse cuenta de las alertas y si es el caso modificar los parámetros necesarios de forma remota para tener control sobre la planta.

Con el desarrollo de esta tarea, nos introducimos en el desarrollo de elementos SCADA como son la interfaz de usuario y la programación por medio de máquina de estados, para lograr observar y controlar el subproceso 1 llamado “recepción del producto”.

CONTENIDO

Interfaz gráfica individual

Figura 1. Pantallazo interfaz gráfica Didier Javier Ramírez Henao.



Interfaz gráfica colaborativa

Teniendo en cuenta que solo yo voté por el aplicativo para trabajar de forma colaborativa y ningún otro compañero lo hizo y que además la actividad vence el día de hoy (03/10/2021), he tomado la decisión de continuar de forma individual. Anexo pantallazos donde se evidencia que voté por el aplicativo para trabajar de forma colaborativa y donde informo al grupo que voy a trabajar de forma individual:

Figura 2. Selección del aplicativo para trabajar de forma colaborativa.

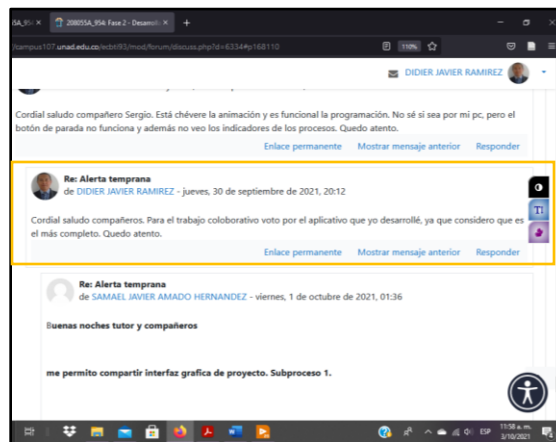
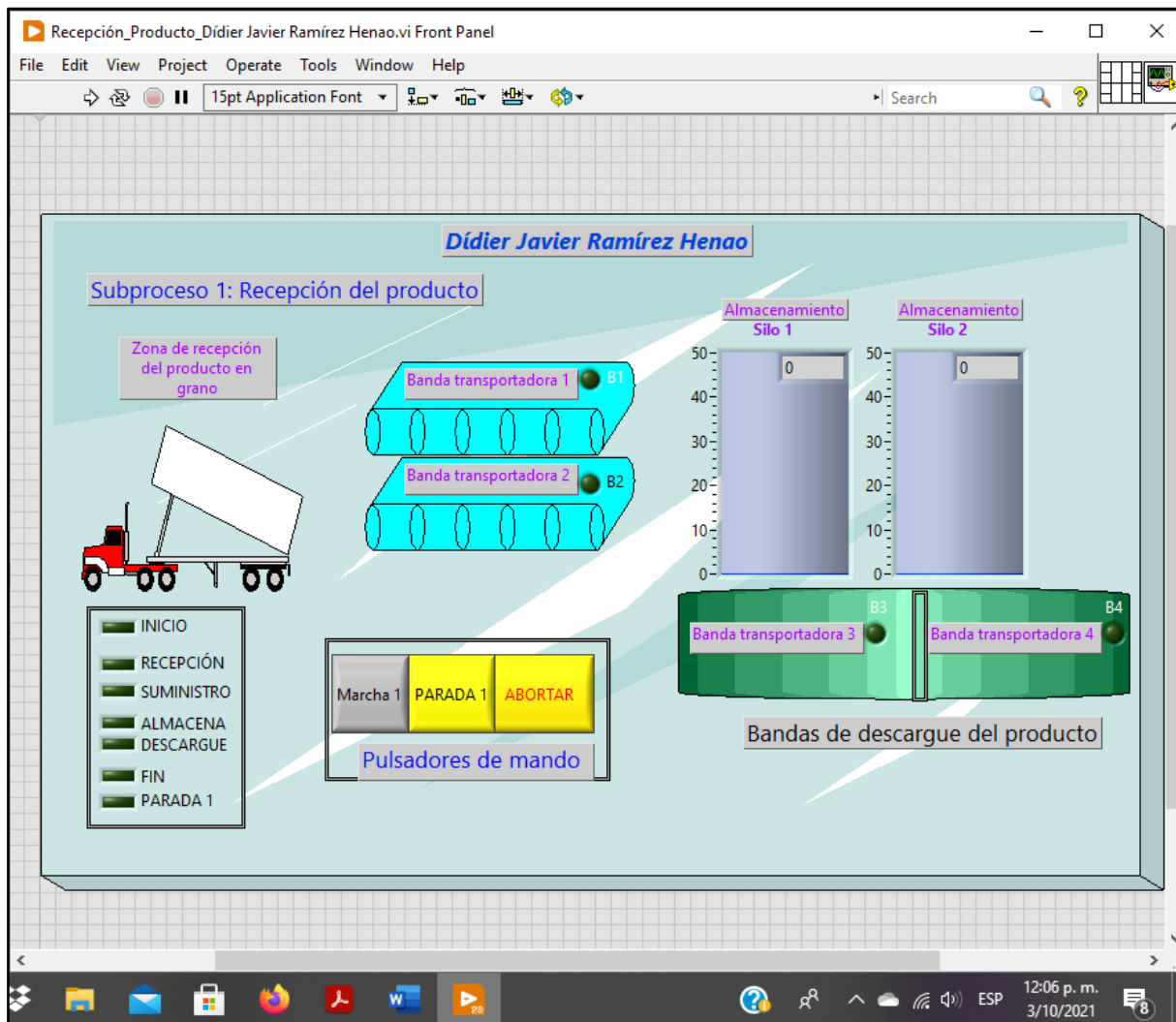


Figura 3. Se informa al grupo que voy a continuar de forma individual.



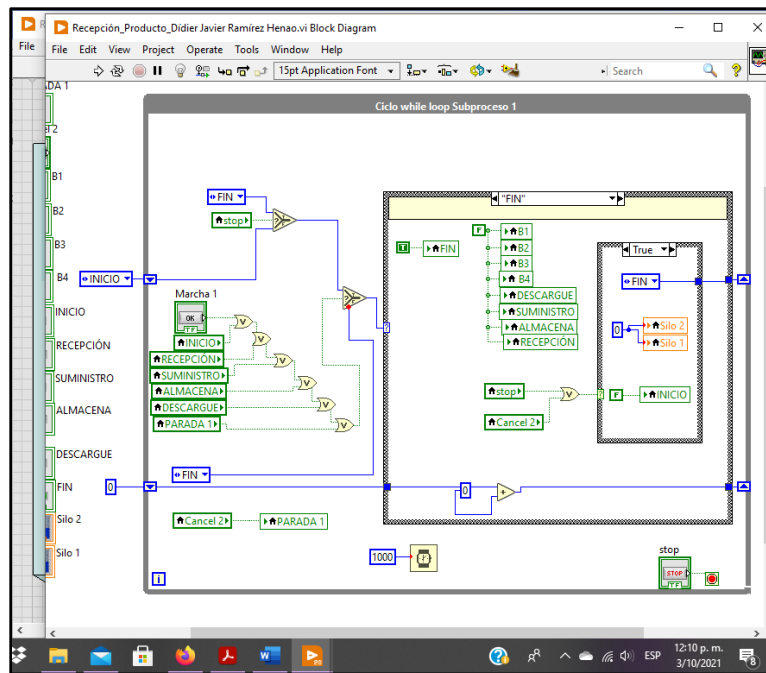
Debido a lo anterior, el aplicativo seleccionado para trabajar en esta actividad es el que yo realicé.

Figura 4. Interfaz gráfica seleccionada.



Descripción de cada estado del algoritmo

Figura 5. Pantalla inicial.



En la figura 5 se observa que antes de ingresar al ciclo “while” se tienen dos registros; uno de ellos es con el fin de que al salir de dicho ciclo se conserve el estado al cual se debe dirigir el programa (inicio, fin, pausa, descargue, almacenamiento, suministro o recepción) y el otro para conservar de llenado de los silos.

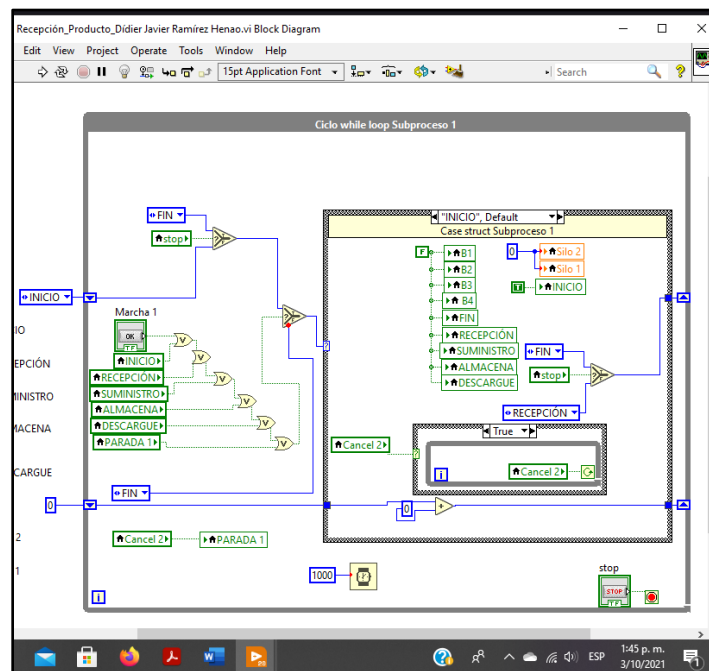
Dentro del “while loop” se tiene:

- ❖ El stop, el cual para el programa y cumple la función de “abortar”.
- ❖ Dos selectores que vamos a denominar selector 1 y selector 2.
- ❖ El selector 1 depende del stop (abortar). En caso de que el Stop esté presionado, selecciona el estado “Fin” y si no se presiona Stop, continúa el estado que viene por fuera del “while loop” a través del registro. El resultado de la selección va al selector 2.
- ❖ El selector 2 depende de varias variables booleanas (Marcha 1, inicio, recepción, suministro, almacena, descargue, parada 1); todas estas variables van unidas a través de

compuestas OR, lo que quiere decir que, si al menos una de estas variables booleanas está activa, se toma la selección como verdadera y en caso de ser verdadera, se envía el estado que fue seleccionado a través del selector 1. Si ninguna de las variables booleanas está activa el selector 2 toma la opción de falso y toma el estado “Fin”. La decisión a través del selector se envía a un “case structure” donde según el estado se cumplen determinados procesos para el correcto funcionamiento de lo que se requiere del programa.

❖ En el “while loop” está la variable booleana “cancel 2”, la cual depende del botón “parada 1”. Cuando el botón de “parada 1” esté activo, se va a activar la variable booleana “parada 1” a través de “cancel 2”. También dentro del “while” se tiene un tiempo de espera de 1 segundo.

Figura 6. Estado "INICIO"



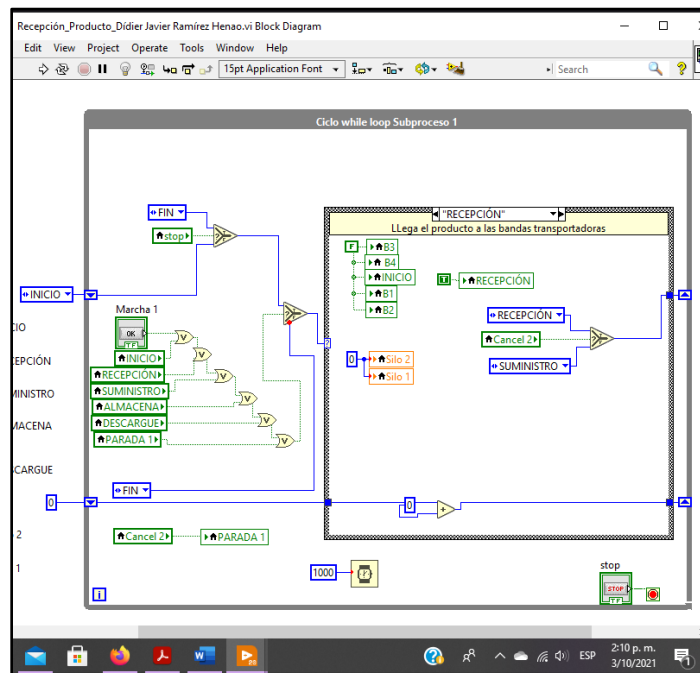
Cuando dentro del “case struct” se selecciona el estado “INICIO” por medio del selector 2, se tiene la siguiente programación:

❖ Las variables booleanas B1, B2, B3, B4 (que corresponden a los indicadores de las bandas), Fin, Recepción, Suministro, Almacena, Descargue (que son indicadores booleanos del proceso en el cual está la planta), se les asigna el valor de “falso”, los silos se ponen en cero y la variable booleana “inicio” se pone como verdadera.

❖ Se tiene un “case Struct” que depende de la variable “cancel 2”, la cual cuando el botón de parada 1 está activo, hace que la selección sea verdadera y entre a un “while loop” y allí se mantiene hasta que se desactiva el botón de parada 2 o hasta que se oprima el botón de “abortar 1”. La función de este “while loop” es que cuando “parada 1” esté activo, se realice el paro del sistema. Si el “case Struct” es falso (botón de “parada 1” no está activo no pasa nada.

❖ Se tiene también un selector que vamos a denominar selector 3, el cual depende del botón “abortar 1”. Si “abortar 1” no está presionado, del estado de “INICIO”, se va a pasar al de “RECEPCIÓN”, pero si está presionado el botón de “abortar 1”, el estado seleccionado será el de “FIN”.

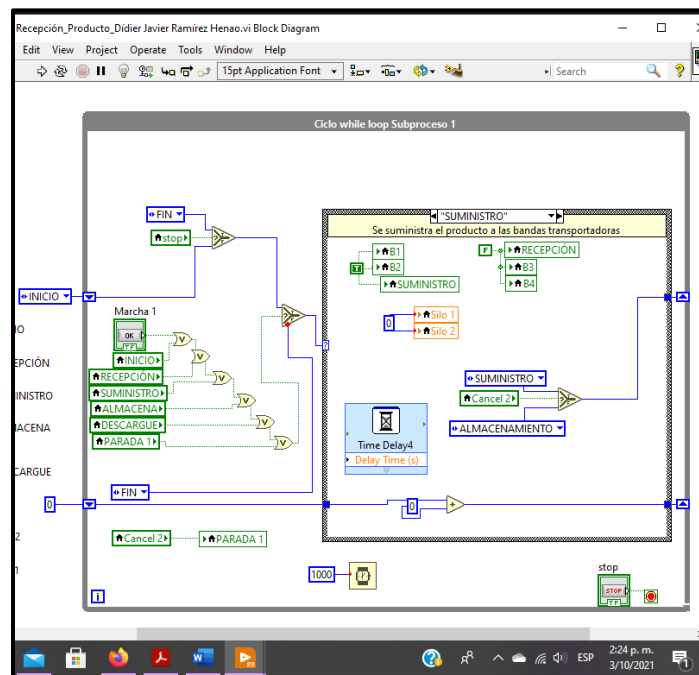
Figura 7. Estado "RECEPCIÓN"



Cuando dentro del “case struct” se selecciona el estado “RECEPCIÓN” por medio del selector 2, se tiene la siguiente programación:

- ❖ B1, B2, B3 y B4 que son los indicadores de las variables booleanas de las bandas transportadoras toman el valor de falso, al igual que el indicador booleano de “inicio”. La variable booleana “recepción” toma el valor de verdadero, para indicar al operario que se encuentra en el proceso de recepción. Los silos toman el valor cero.
- ❖ Se tiene un selector que se va a denominar selector 4 el cual depende del botón de parada 1. Si el botón de “parada 1” está activo, entonces se selecciona el estado “RECEPCIÓN”, es decir, continúa en el estado de “RECEPCIÓN” hasta que se deje de oprimir el botón de “parada1”. Si el botón de “parada 1” no está activo, se avanza al estado de “SUMINISTRO” por medio del registro.

Figura 8. Estado de "SUMINISTRO"

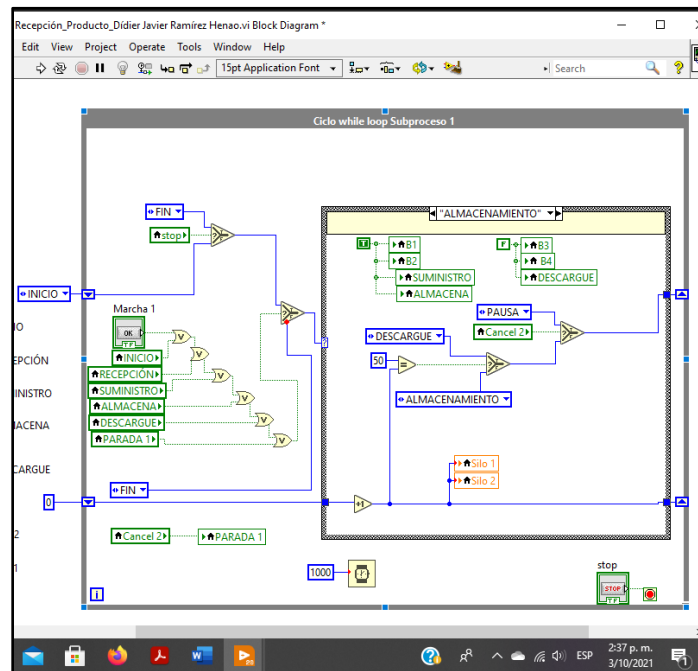


Cuando dentro del “case struct” se selecciona el estado “SUMINISTRO” por medio del selector 2, se tiene la siguiente programación:

- ❖ Las variables booleanas de las bandas transportadoras B1 y B2 se ponen en valor verdadero para indicar que se está transportando el producto por dichas bandas. La variable booleana de “suministro” también se pone verdadera para indicar al operario de la planta que se está en dicho proceso.
- ❖ La variable booleana “recepción” se pone en falso, para indicar que este proceso ha terminado. Los silos toman el valor numérico de cero. Se pone un “time delay” para esperar dos segundos mientras el producto llega de las bandas transportadoras desde la recepción del producto al almacenamiento.
- ❖ Se tiene un selector que se va a denominar selector 5 el cual depende del botón de parada 1. Si el botón de “parada 1” está activo, entonces se selecciona el estado “SUMINISTRO”, es decir, continúa en el estado de “SUMINISTRO” hasta que se deje de

oprimir el botón de “parada1”. Si el botón de “parada 1” no está activo, se avanza al estado de “ALMACENAMIENTO” por medio del registro.

Figura 9. Estado de "ALMACENAMIENTO"

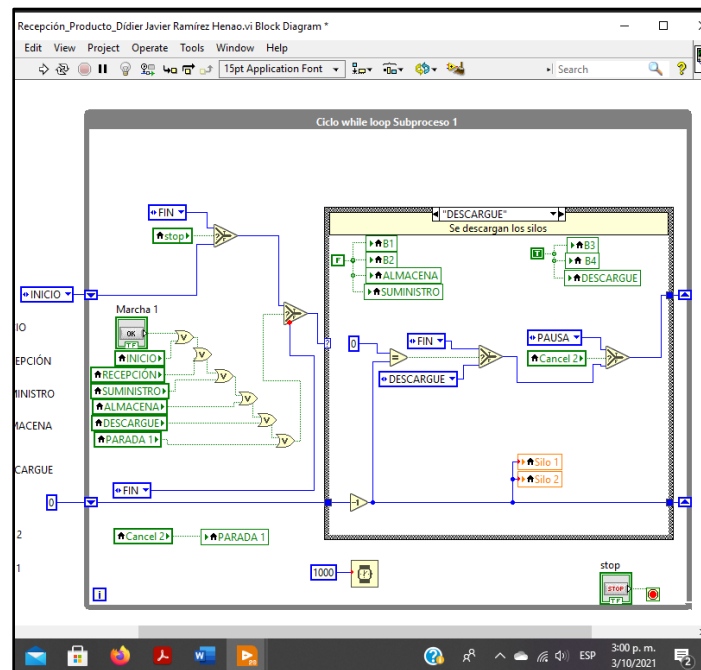


- ❖ Los indicadores booleanos B1 y B2 continúan en verdadero indicando que se está suministrando producto a los tanques de almacenamiento. Como hay suministro de producto y además se está almacenando en los silos, los indicadores booleanos de “suministro” y “almacenamiento” se ponen en verdadero para que el operario pueda ver el proceso en el que va la planta.
- ❖ Los indicadores booleanos “descargue”, B3 y B4 se ponen en falso, ya que el producto se está almacenando y no se está descargando. Los silos se empiezan a llenar.
- ❖ Se tiene dos selectores que se van a denominar selector 6 y selector 7.
- ❖ El selector 6 es falso siempre y cuando el registro para los silos sea menor a 50; en este caso se continúa en el estado “ALMACENAMIENTO” y los silos se siguen llenando.

Cuando el valor del registro para los silos es igual a 50, el estado seleccionado es el de “DESCARGUE”, ya que los silos están llenos. El resultado del selector 6 va al selector 7.

❖ El selector 7 depende del botón de “parada 1”. Si se activa el botón de parada 1, entonces el estado seleccionado es el de “PAUSA”. Si el botón de “parada 1” no está oprimido, la selección realizada es la que viene del selector 6.

Figura 10. Estado de "DESCARGUE"



❖ Los indicadores booleanos B1, B2, “suministro” y “almacenamiento” pasan a estado falso para indicar al operario que estos procesos ya sucedieron. Los indicadores booleanos B3, B4 y “descargue” se ponen en verdadero para que el operario se de cuenta que las bandas de descargue están funcionando y el proceso de descargue están aconteciendo.

❖ Por medio del registro para los silos se empieza a decrementar su valor.

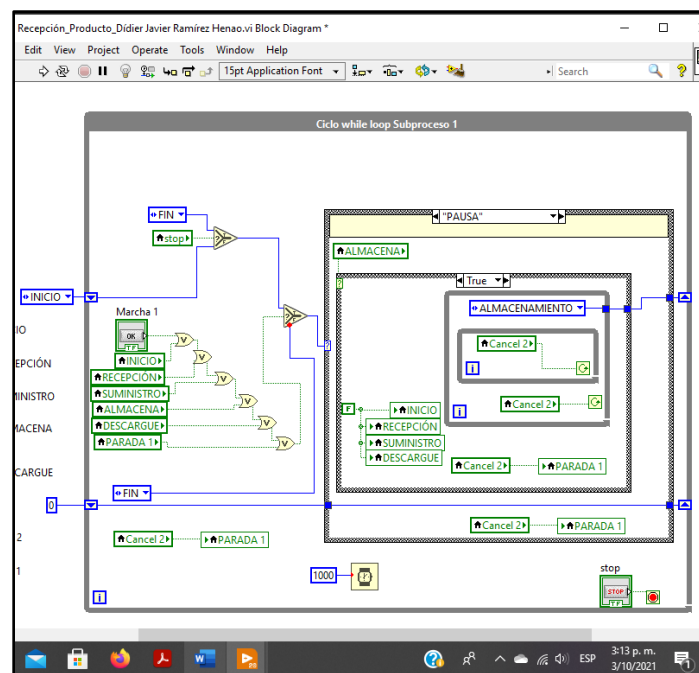
❖ Se tiene dos selectores que se van a denominar selector 8 y selector 9.

❖ El selector 8 es toma la opción de falso siempre y cuando el registro para los silos sea mayor a cero; en este caso se continúa en el estado “DESCARGUE” y los silos se siguen

desocupando. Cuando el valor del registro para los silos es igual a cero, el estado seleccionado es el de “FIN”, ya que los silos están desocupados. El resultado del selector 8 va al selector 9.

- ❖ El selector 9 depende del botón de “parada 1”. Si se activa el botón de parada 1, entonces el estado seleccionado es el de “PAUSA”. Si el botón de “parada 1” no está oprimido, la selección realizada es la que viene del selector 8.

Figura 11. Estado de "PAUSA"



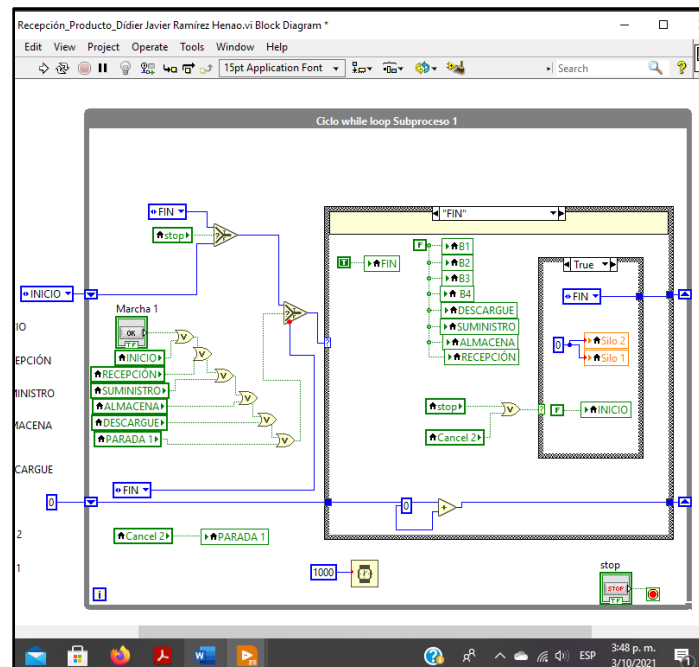
- ❖ Dentro del estado de pausa, cuando se oprime el botón de “parada 1”, se activa el indicador booleano “parada 1”, para indicar al operario que se ha pulsado este botón.
- ❖ Se encuentra un “case Struct” que depende de la variable booleana “almacena”.
- ❖ Cuando la variable booleana “almacena” es verdadera (lo que quiere decir que el paro se realizó cuando los silos estaban almacenando), las variables booleanas “inicio”, “recepción”, “suministro” y “descargue” se ponen en falso, para evitar que al momento del

paro alguno de estos indicadores quede activo. El indicador booleano “parada 1” se pone activo hasta que se deje de oprimir el botón de “parada 1”.

❖ Dentro del “case Struct” de pausa se encuentran dos “while loop” que dependen ambos del botón de “parada 1”; cuando dicho botón está activo se continúa en el “while loop” interno hasta que se deje de oprimir este botón. Cuando se deja de oprimir el botón de “parada 1”, el “while loop” también se deshabilita y continúa al estado de almacenamiento.

❖ Cuando el “case Struct” de “PAUSA” es falso, sucede lo mismo que cuando es verdadero, lo único que cambia es que envía al estado de “DESCARGUE” cuando se deja de oprimir el botón de “parada 1”, puesto que la parada del sistema se hizo cuando estaba en el estado de “DESCARGUE” y que en vez de poner en false la variable booleana “descargue”, se pone en false la variable booleana “almacenamiento”.

Figura 12. Estado "FIN"



- ❖ En el estado “FIN” la variable booleana “fin” se pone con valor verdadero, mientras que las variables booleanas B1, B2, B3 y B4, “descargue”, “suministro”, “almacena” y “recepción” se ponen en valor falso.
- ❖ Se cuenta con un “case Struct” que es verdadero cuando alguna de las variables booleanas de “stop” (abortar 1) o “cancel 2” (parada 1) está activo. Si esto sucede los silos se aseguran en cero, la variable booleana “inicio” se pone en falso y continúa en el estado de “FIN” por medio del registro. Cuando ninguno de estos botones está habilitado continúa en el estado “INICIO”.

CONCLUSIONES

- ❖ Por medio de LABVIEW se pueden crear interfaces gráficas que representan los procesos que están involucrados en una planta y que son fácilmente interpretados por los operarios.
- ❖ Con el desarrollo del código de programación de LABVIEW, se pueden recrear las interfaces gráficas, de tal manera que represente fidedignamente lo que sucede dentro de la planta y así se pueda llevar control y tomar decisiones respecto de lo que pasa en la fábrica.
- ❖ La implementación de programación por máquina de estados es muy eficiente cuando se pueden identificar los estados de la aplicación a programar. En su implementación un estado conduce a otro, los cuales dependen del diseño del programa mismo y de las entradas que realice el usuario.

REFERENCIAS BIBLIOGRÁFICAS

- ❖ Julian, E., y Almidon, A. (2019). Manual de programación LabVIEW 9.0. Páginas 22-36. Recuperado <https://doi-org.bibliotecavirtual.unad.edu.co/10.5281/zenodo.2557815>
- ❖ Flórez Martínez, A. (26,01,2017). Unidad1_Estructuras y tipos de datos en Labview. [Archivo de video]. Recuperado de: <http://hdl.handle.net/10596/10879>
- ❖ Yang, Y. (2014). LabVIEW Graphical Programming Cookbook. Packt Publishing. Páginas 59-70. Recuperado de https://bibliotecavirtual.unad.edu.co/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=690400&lang=es&site=eds-live&scope=site&ebv=EB&ppid=pp_59