

LABVIEW

FASE 5: CONSTRUIR SISTEMA DE SUPERVISIÓN Y CONTROL

PRESENTADO POR

DÍDIER JAVIER RAMÍREZ HENAO

PRESENTADO A

ALEXÁNDER FLÓREZ

GRUPO

208055_1

DICIEMBRE 03 DE 2021

MANIZALES – CALDAS

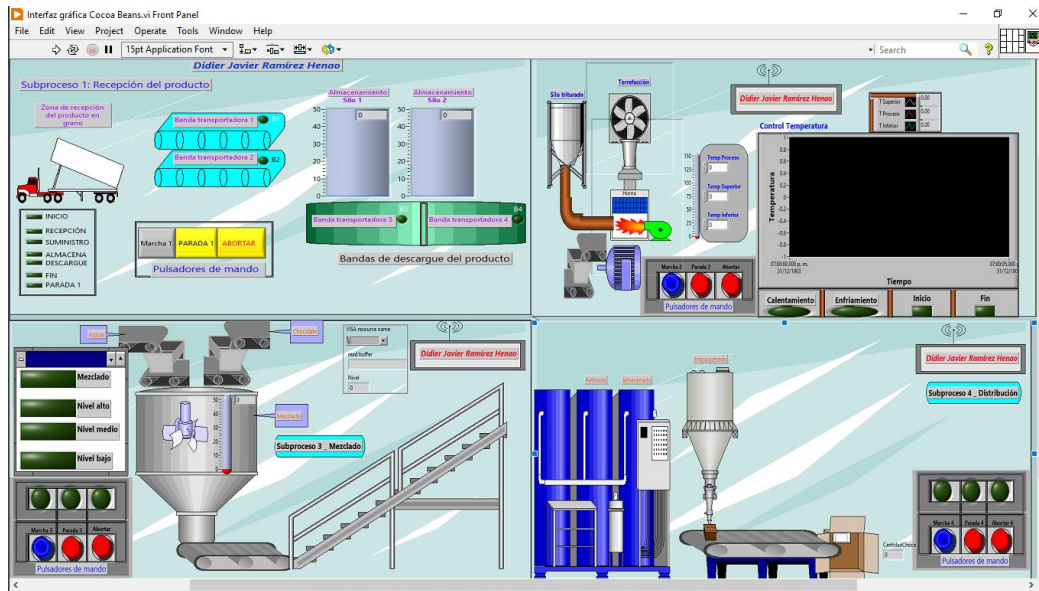
INTRODUCCIÓN

Con el desarrollo de la presente actividad se integran las interfaces gráficas de los subprocesos anteriores con la interfaz gráfica del subproceso de distribución, con lo que se puede visualizar de forma gráfica todos los procesos que ocurren dentro de la planta.

También se encuentra el desarrollo de la programación del subproceso de distribución de la empresa Cocoa Beans, cumpliendo con los requerimientos de la guía de actividades. Con la culminación de esta fase, se logra integrar cada una de las etapas, para que la empresa Cocoa Beans tenga un proceso de automatización acorde a sus necesidades.

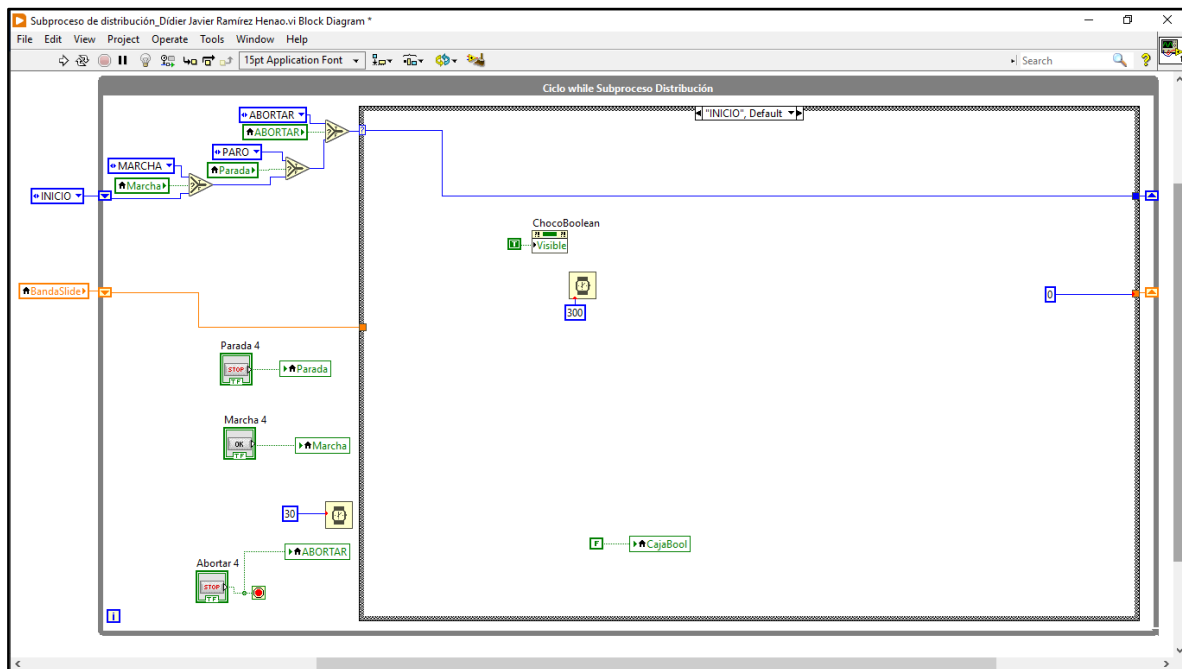
DESARROLLO DE LAS ACTIVIDADES

Figura 1. Interfaz gráfica Didier Javier Ramírez Henao.



Explicación del algoritmo subproceso de distribución:

Figura 2. Ciclo while del subproceso de distribución.

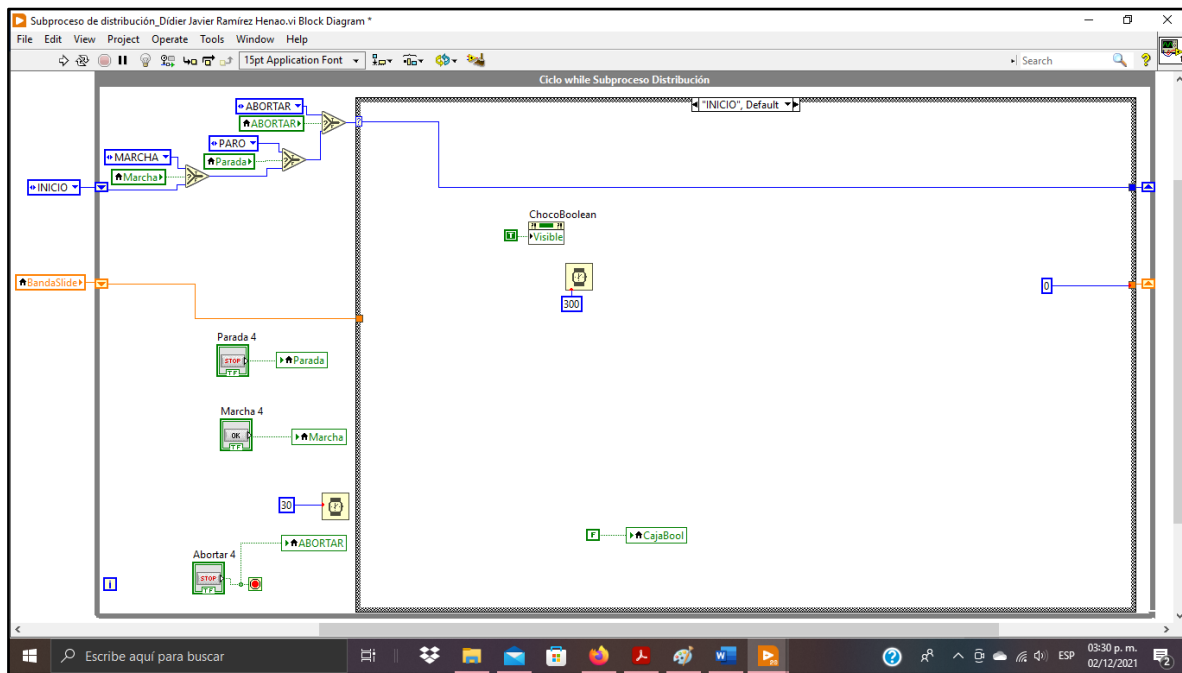


En la figura 2 se observa que en la parte externa del “while loop” se tienen dos registros. Un registro que tiene el propósito de que al salir de dicho ciclo se conserve el estado al cual se debe dirigir el programa (INICIO, FIN, MARCHA, ABORTAR, PARO). El otro registro se llama BandaSlide y allí se conserva el valor de la banda deslizante.

Dentro del “while loop principal” se tiene:

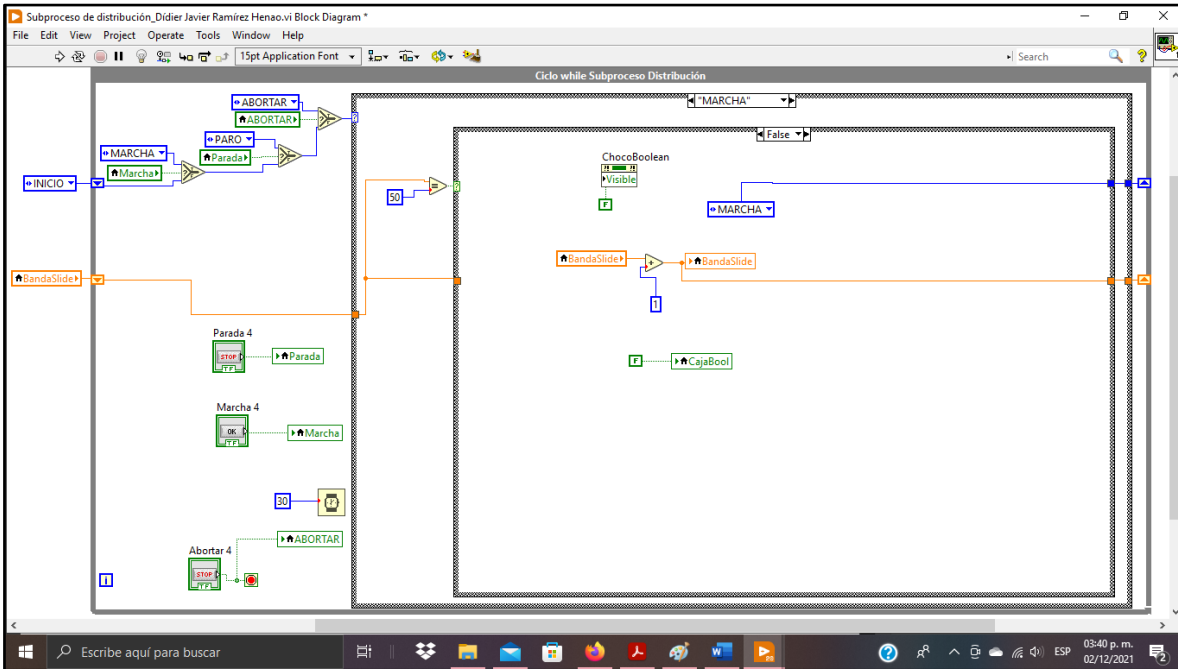
- ❖ El stop, el cual para el programa y cumple la función de “Abortar 4” conectado al Stop, se tiene un indicador booleano denominado ABORTAR, el cual se activa o desactiva dependiendo del estado de Abortar 4. Al botón de Parada 4 está conectado el indicador booleano de PARADA, el cual se activa o desactiva dependiendo del estado de Parada 4. Al botón de Marcha 4 está conectado el indicador booleano de marcha, el cual se activa o desactiva dependiendo del estado del botón de marcha.
- ❖ Tres selectores que vamos a denominar selector 1, selector 2 y selector 3. El selector 1 depende del indicador de Marcha; si el botón de marcha está pulsado el selector 1 es verdadero y selecciona el estado de MARCHA y si es falso se selecciona el estado que va a través del registro. El resultado de la selección va al selector 2.
- ❖ El selector 2 depende del indicador booleano de Parada; si este indicador está activo, el selector 2 toma la opción como verdadera y envía el estado de PARO; si es falso, se selecciona el estado que viene del selector 1.
- ❖ El selector 3 depende del indicador booleano de ABORTAR; si este indicador está activo, quiere decir que el botón de abortar está presionado y el selector 3 toma el valor verdadero seleccionando el estado de ABORTAR y si es falso, toma el estado que viene del selector 2.

Figura 3. Estado INICIO.



En este estado, el estado que llega por medio del selector 3 es enviado directamente al registro de estado. La variable BandaSlide se pone en cero a través del registro. La variable CajaBool se pone en falso, ya que es una variable booleana donde la caja está abierta. La variable ChocoBoolean se pone en valor verdadero para que sea visible y se observe al inicio del proceso un empaque de color verde. Finalmente se da un tiempo de espera de 300 ms para poder evidenciar el cambio de color en el empaque cuando se dé Marcha 4.

Figura 4. Estado MARCHA, con Case Structure falso.



Se tiene el estado de marcha (Figura 4), un Case Structure que depende de la condición de si BandaSlide es igual a 50, que es valor máximo que puede tomar la banda que transporta las barras de chocolate. Si la BandaSlide no ha llegado a 50, el Case Structure es falso y sucede lo siguiente:

La visibilidad de ChocoBoolean es falsa con lo que no se vé, quedando el empaque que está debajo, generando que de color verde pasa a ser color chocolate, simulando que se inyectó chocolate al paquete. Por medio del registro se envía al estado de MARCHA. Se va sumando una unidad a BandaSlide y este valor es enviado al registro. La caja está abierta por medio del valor falso que se pone a CajaBool.

[illegible]

La BandaSlide toma el valor de cero y ChocoBoolean se hace visible. Se regresa al estado de INICIO.

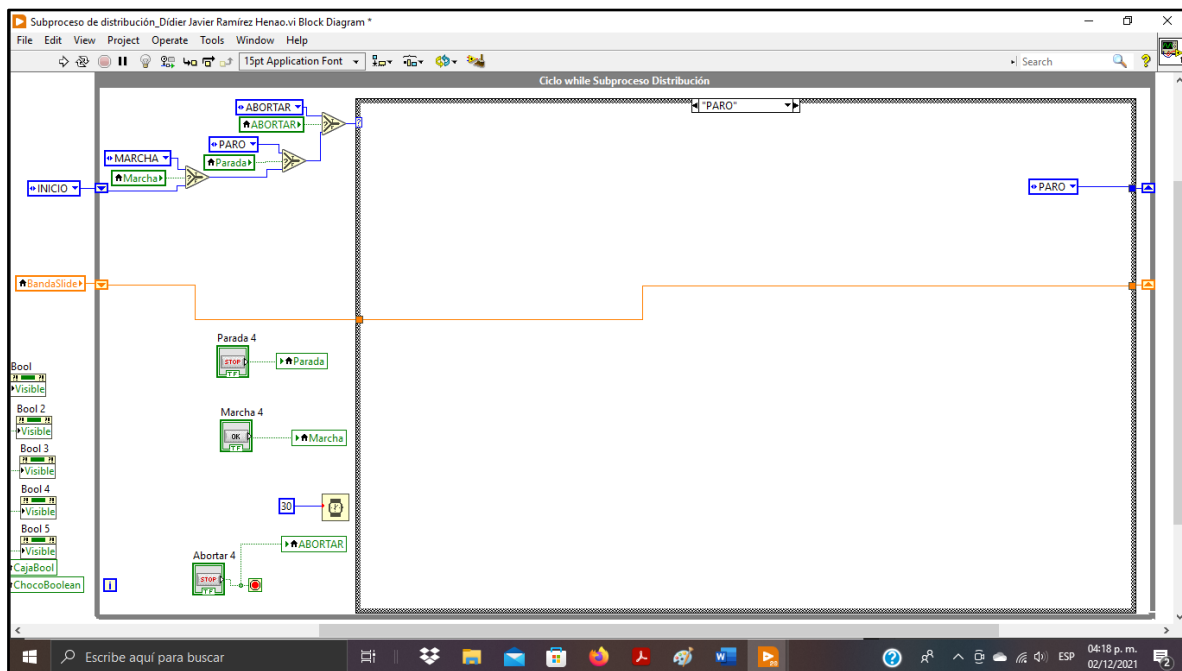
Se adiciona una unidad a la variable CantidadChoco. Se compara el valor de cantidad de choco. Si CantidadChoco vale 1, se hace visible la variable Bool para mostrar que hay un paquete en la caja. Si CantidadChoco vale 2, se hace visible la variable Bool 2 para mostrar que hay dos paquetes en la caja. Si CantidadChoco vale 3, se hace visible la variable Bool 3 para mostrar que hay tres paquetes en la caja. Si CantidadChoco vale 4, se hace visible la variable Bool 4 para mostrar que hay cuatro paquetes en la caja. Si CantidadChoco vale 5, se hace visible la variable Bool 5 para mostrar que hay cinco paquetes en la caja.

Adentro del Case Structure se tiene otro Case Structure que depende de la variable CantidadChoco. Si CantidadChoco es menor a 5, la variable CajaBool se pone en falso, con

lo que la caja se ve abierta. Si Cantidad de Choco es mayor o igual a 5, se tiene un Flat Sequence Structure que tiene la siguiente secuencia:

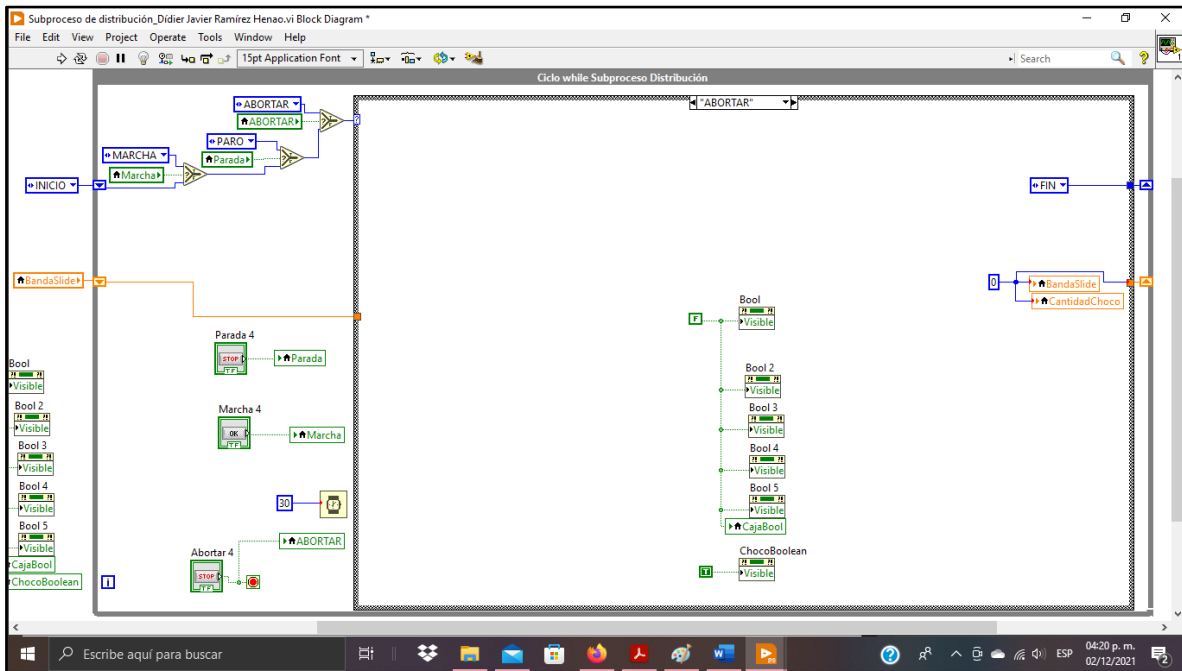
En la primera secuencia hay un retardo de 1 segundo para que se pueda visualizar el proceso de la caja con los paquetes adentro. Luego las variables Bool, Bool 2, Bool 3, Bool 4 y Bool 5 se pone su visibilidad en falso con lo que en la caja no se ven los paquetes, aunque esto no lo idéntica el operario porque inmediatamente después sigue la otra secuencia. La última secuencia consiste en poner la variable CajaBool en verdadero, con lo que se observa que la caja se cierra, dando la apariencia que la caja se llenó de los paquetes de chocolate; la variable CantidadChoco se pone en cero y se genera un retardo de 1 segundo para que el proceso pueda ser evidente.

Figura 6. Estado PARO.



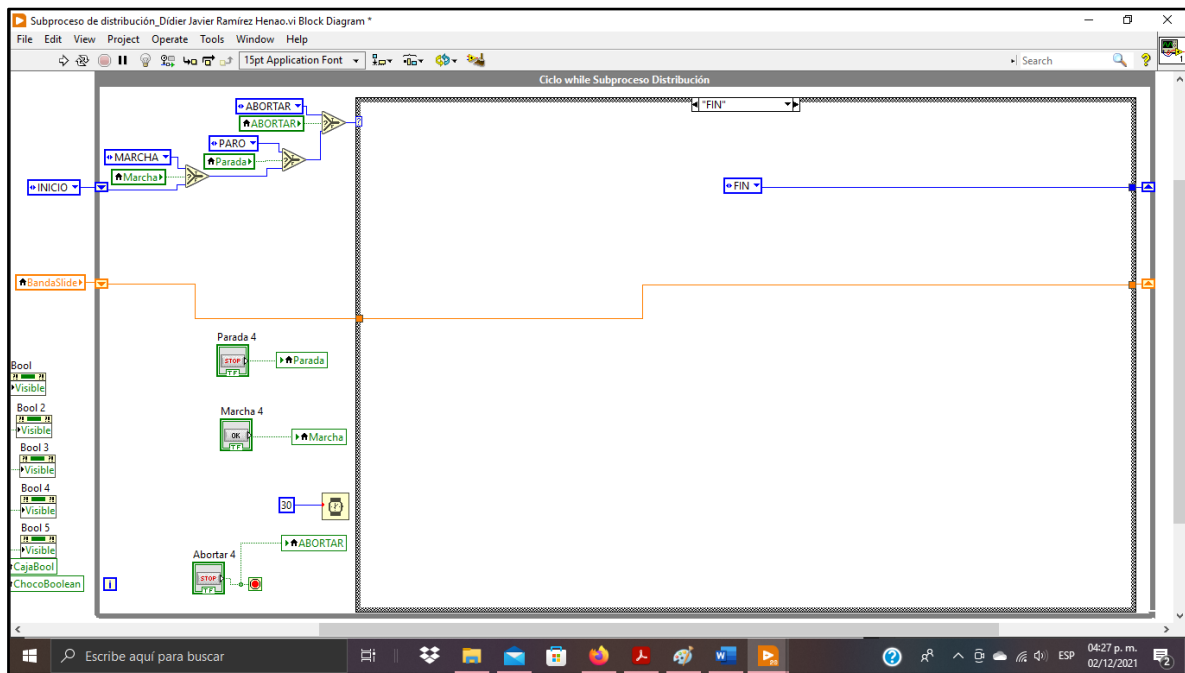
Este estado ocurre (figura 6) cuando el botón de Parada 4 está activo y lo único que sucede es que se envía al estado de PARO a través del registro indefinidamente hasta que se desactive el botón de Parada 4 o se active el botón de Abortar 4.

Figura 7. Estado ABORTAR.



Este estado ocurre (figura 7) cuando el botón de Abortar 4 está activo y lo que sucede es que se envía al estado de FIN a través del registro indefinidamente hasta que se desactive el botón de Abortar 4. A las variables Bool, Bool 2, Bool 3, Bool 4 y Bool 5 se les pone la visibilidad en falso, BandaSlide y CantidadChoco se ponen en cero; lo anterior para volver las variables a condiciones iniciales.

Figura 8. Estado FIN.



En el estado FIN, se envía el al estado FIN por medio del registro.

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.

La librería datalogging and supervisory control (DSC) de LabVIEW contiene diversos instrumentos virtuales y tienen la ventaja de que se les puede agregar animación, para obtener una interfaz gráfica más real de los procesos que suceden dentro de una planta.

REFERENCIAS BIBLIOGRÁFICAS

Información General sobre NI-VISA. Visión general. Información suplementaria de NI. Recuperado de: <https://www.ni.com/es-co/support/documentation/supplemental/06/ni-visa-overview.html>

JKI VI Package Manager. NI Engineer Ambitiously. Recuperado de: <https://www.ni.com/es-co/shop/software/products/jki-vi-package-manager.html>

Yang, Y. (2014). LabVIEW Graphical Programming Cookbook. Packt Publishing. Páginas 155-161. 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_155

Julián, E., y Almidón, A. (2019). Manual de programación LabVIEW 9.0. Páginas 45-48. Recuperado de <https://doi-org.bibliotecavirtual.unad.edu.co/10.5281/zenodo.2557815>

Lajara Vizcaíno, J. R. (2008). LabVIEW: entorno gráfico de programación. Marcombo. Páginas 205-208. Recuperado de <https://elibro-net.bibliotecavirtual.unad.edu.co/es/ereader/unad/35715?page=216>

Flórez Martínez, A. (26,01,2017). Unidad2_Sistema de Supervisión y Control. [Archivo de video]. Recuperado de <http://hdl.handle.net/10596/10878>