TUTORIAL TIA

AUTOR

RUBÉN DARÍO VÁSQUEZ SALAZAR

DOCENTE DE TIEMPO COMPLETO

ÁREA INSTRUMENTACIÓN Y CONTROL

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

2012

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

TABLA DE CONTENIDO

LISTA DE FIGURAS

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

1. INTRODUCCIÓN

El entorno TIA Portal ha lanzado recientemente sus versiones 10 y 11. Ambas versiones son iguales en diseño y con muy pocas diferencias en diseño. Algunas imágenes de este tutorial fueron tomadas de las diferentes versiones, por lo cual puede haber algunas sutiles diferencias.

Las versiones Basic permiten únicamente programar y configurar los PLCs S7-1200 y sus respectivos paneles de operador, pero la versión Professional que ya está disponible para empresas, no para instituciones educativas por el momento, permite hacerlo también para los equipos de la gama S7-300 y S7-400, además posee simulador, lo que se convierte en un gran factor diferenciador, ya que permite evaluar la programación realizada sin necesidad de conectarse al PLC real.

Este programa lo denominan los de Siemens como un concepto que incluye nuevas tendencias y herramientas en cuanto a programación y configuración de Controladores Lógicos Programables (PLC por sus siglas en inglés). En este concepto se pretenden integrar diferentes herramientas de automatización en un solo paquete, por lo tanto el TIA Portal no solamente es útil para PLCs, sino para paneles de operador, redes de comunicación industrial y otros dispositivos.

La parte II del tutorial TIA Portal se ha enfocado más hacia funciones más avanzadas de los PLC Siemens S7-1200, incluyendo principalmente:

Comunicaciones industriales - Industrial Ethernet: Se incluyen ejercicios de comunicación con otras plataformas como Matlab y Labview, además de enlaces entre PLCs.

Comunicaciones industriales – Profibus: Se explica el procedimiento para configurar la red y establecer comunicación con sensores y actuadores que cumplen este protocolo.

WinCC Flexible: Diseño de Interfaces Hombre Máquina (HMI) para pantallas táctiles que permitan construir el sistema de supervisión de un proceso industrial.

En esta segunda parte del tutorial se contó con la auxiliar de docencia Tatiana López (tatin1530@gmail.com), quién ha sido un apoyo para las conexiones, simulaciones y documentación de los ejercicios y pruebas realizadas.

2. PROYECTO TIA CON MÁS DE 1 CPU

La idea de tener varias CPUs en un solo proyecto es para que, estando todas conectadas en red, se pueda acceder a cada una de ellas sin tener que abrir o cerrar proyectos. Así, se podrían manipular todos los PLCs de una empresa desde un único computador y único proyecto TIA.

Se requiere que al iniciar este procedimiento ya se cuente con un proyecto con una CPU correctamente configurada. Esto se realiza con los pasos explicados en la parte I del tutorial. El proyecto debe verse como se muestra en la Figura 1.

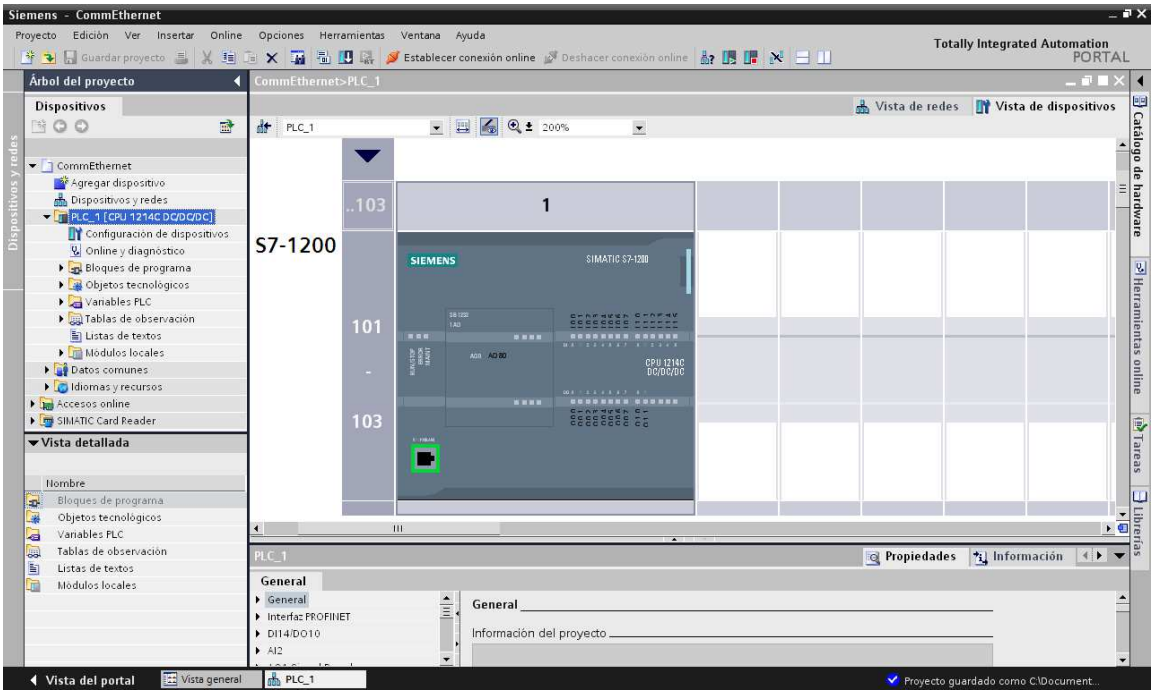

Figura 1. Proyecto con una CPU configurada

Si se desea añadir otro dispositivo entonces haga clic en “Agregar dispositivo” y siga los pasos que ya debe tener bien familiarizados.

Sin embargo, únicamente para el caso en que la otra CPU sea igual que la primera, entonces se pueden utilizar las opciones de copiar y pegar.

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

Seleccionamos el PLC_1 y seleccionamos la opción Copiar

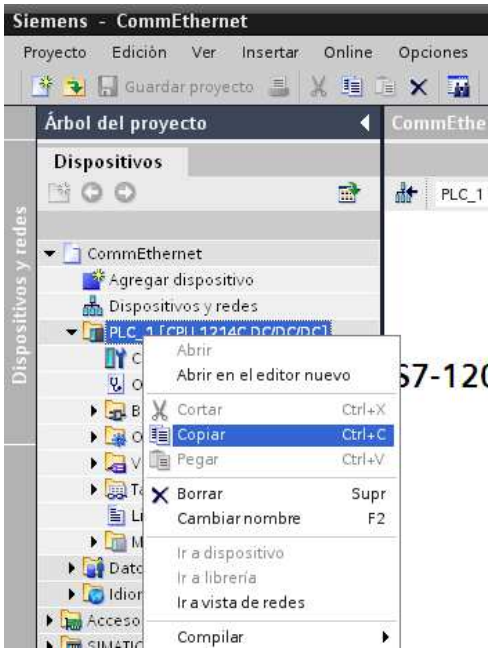

Figura 2. Copiar el PLC_1

Luego le damos clic derecho en el nombre del proyecto y le damos “Pegar”.

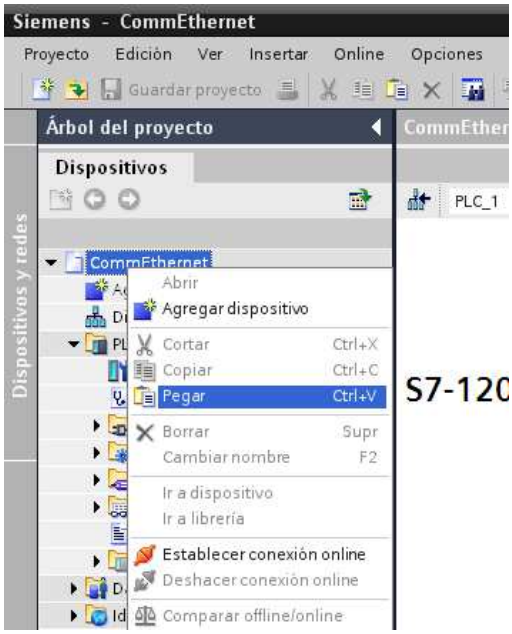

Figura 3. Pegar nuevo dispositivo

Con lo anterior obtenemos un dispositivo con las mismas características del PLC_1.

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

Ya se podrá ver el PLC en el proyecto. Puede modificarle el nombre, por ejemplo “PLC_2”.

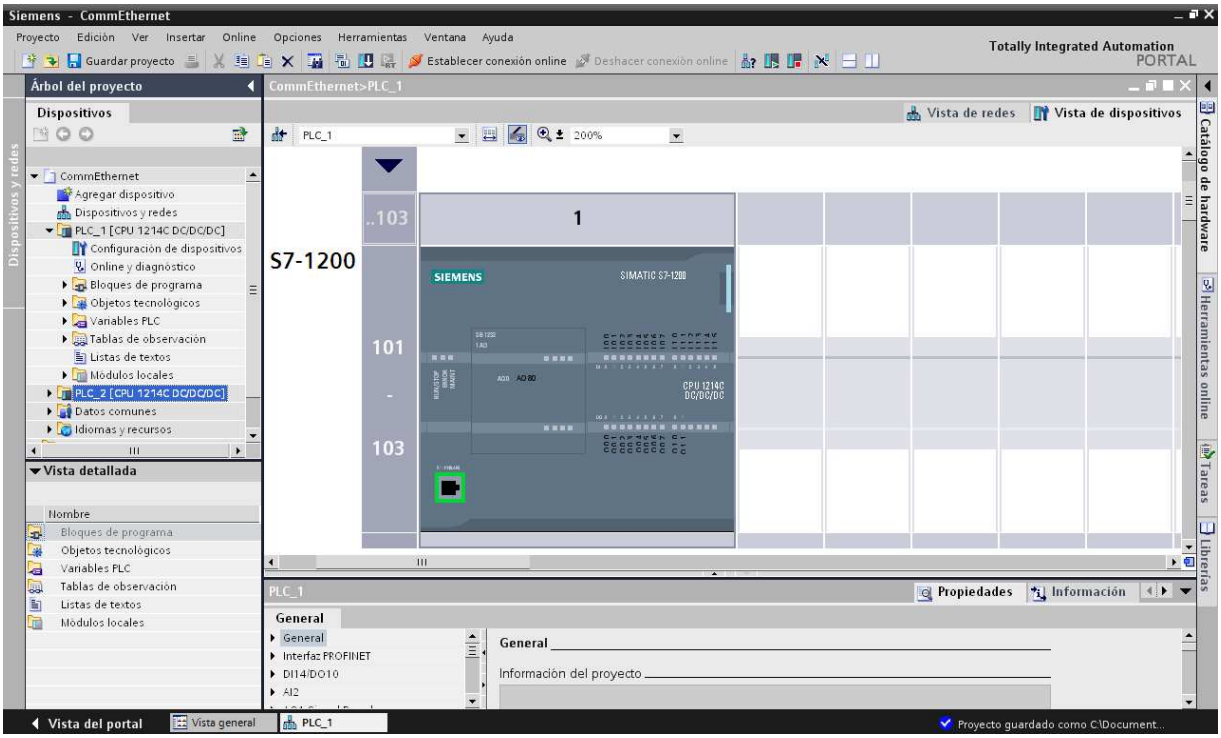

Figura 4. Obtención PLC_2

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

A continuación hacemos doble clic en la interface de comunicación Ethernet y vamos a “Direcciones Ethernet”.

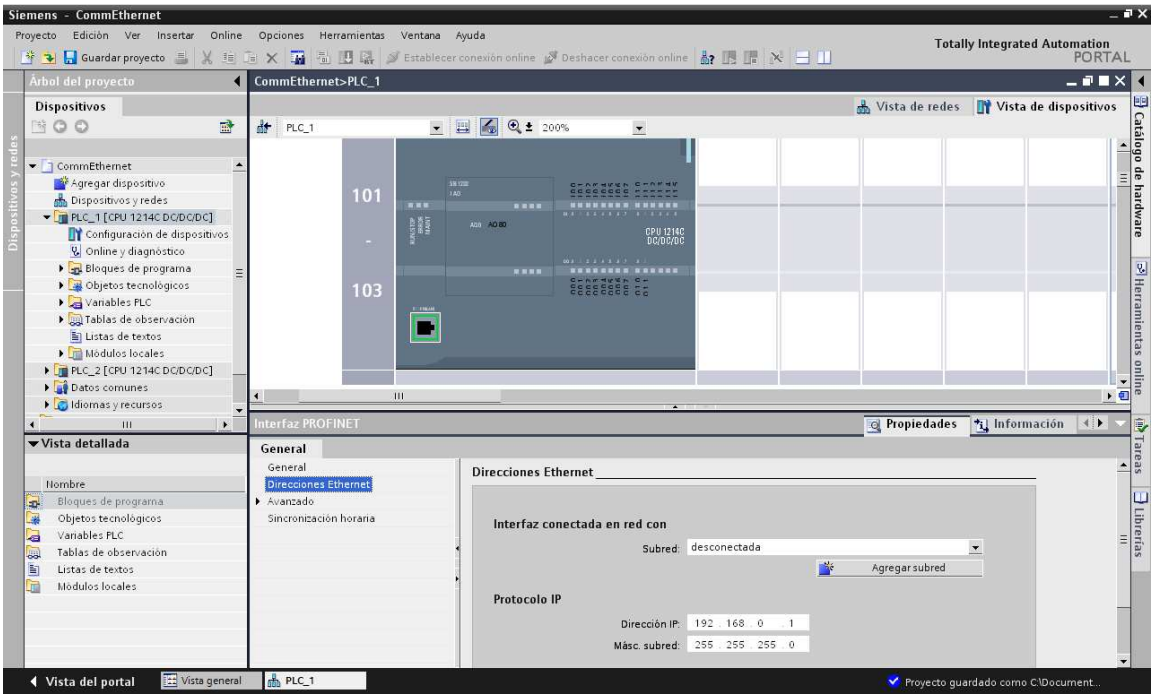

Figura 5. Direcciones Ethernet PLC_1

Para el PLC_1 se agrega la subred en la cual estarán los PLC y se asigna una dirección IP correspondiente a la CPU del dispositivo actual, de la siguiente manera:

Dar clic en el botón “Agregar subred”.

En el campo Subred aparecerá PN/IE_1, seleccionados esta opción.

Asignamos al PLC_1 la siguiente dirección IP: 192.168.0.1

Asignamos al PLC_1 la siguiente máscara de subred: 255.255.255.0

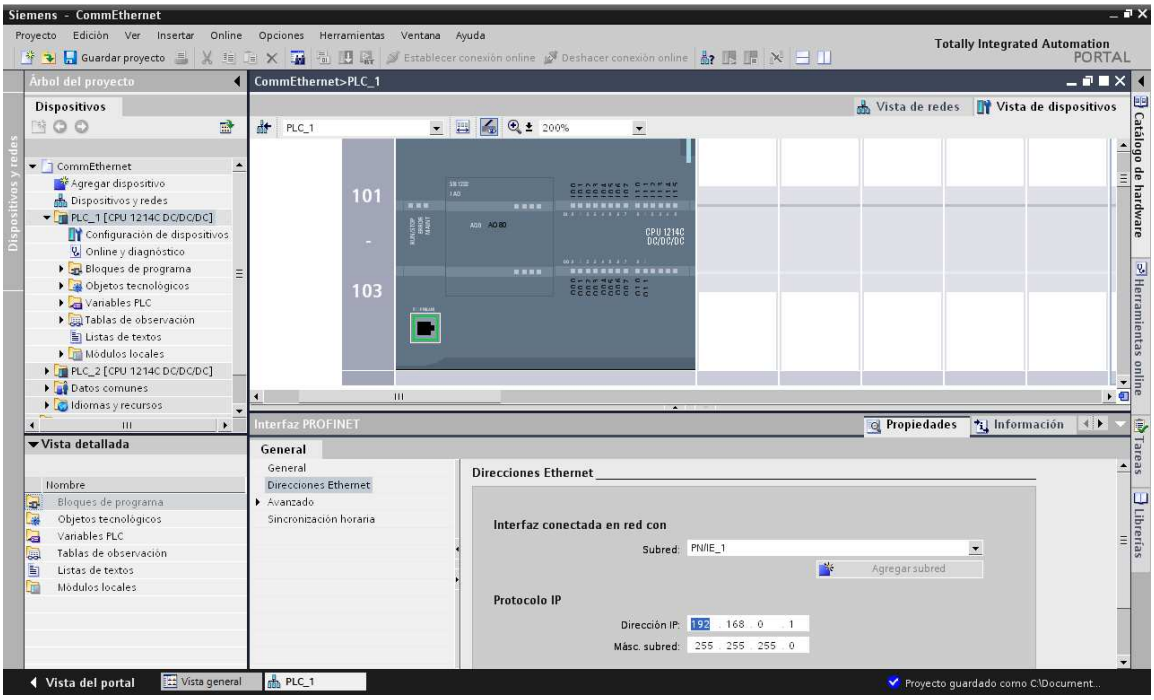

Figura 6. Agregar Subred y asignar dirección IP al PLC_1

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

Para el PLC_2 se realizan los mismos pasos de configuración del dispositivo, es decir, agregar el dispositivo a la misma subred pero teniendo en cuenta que esta CPU debe tener una dirección IP distinta a la del PLC_1. Por ejemplo asigne la 192.168.0.2

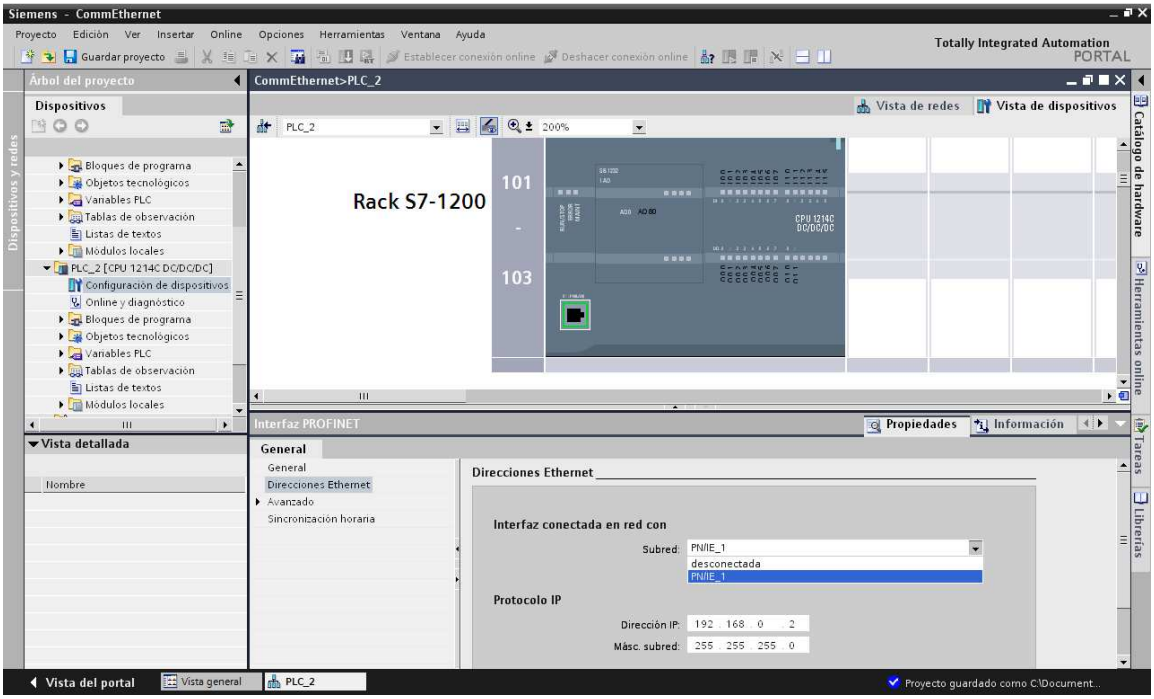

Figura 7. Agregar Subred y asignar dirección IP al PLC_2

Y listo, ya se tienen 2 PLCs en la misma subred y en el mismo proyecto. Se puede cargar el programa a cada PLC con conexión punto a punto entre el PC y el PLC. Una vez cargado se pueden pegar todos en red utilizando un suiche o un router y desde el mismo PC se podrá modificar el programa o ver en línea cualquier PLC perteneciente al proyecto.

3. COMUNICACIÓN ETHERNET ENTRE 2 PLCS

Asegúrese de contar con un proyecto TIA con 2 PLCs configuradas y conectadas a la misma subred, como se explicó en “PROYECTO TIA CON MÁS DE 1 CPUCOMUNICACIÓN ETHERNET ENTRE 2 PLCS”.

Lo primero que se va a hacer es enviar datos desde el PLC_1 hacia el PLC_2. Para ello sitúese en “Bloques de programa” del PLC_1, nos ubicamos en el Main [OB1] y buscamos en la parte lateral derecha el catálogo de “Instrucciones Avanzadas” donde encontraremos una carpeta llamada “Comunicación” de la cual podemos obtener un bloque llamado TSEND_C, función con la cual haremos el envío de datos al PLC_2.

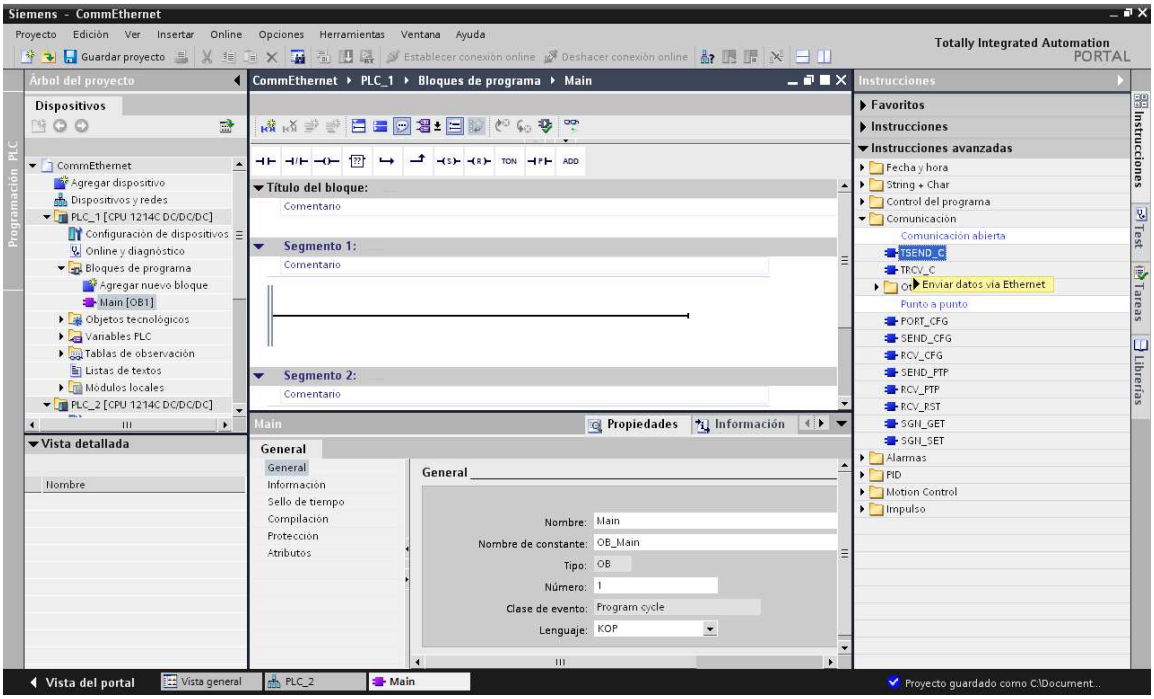

Figura 8. PLC_1 Main[OB1]

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

Arrastre el bloque TSEND hacia algún segmento de programación del OB1. Se abrirá un cuadro con las opciones de llamada.

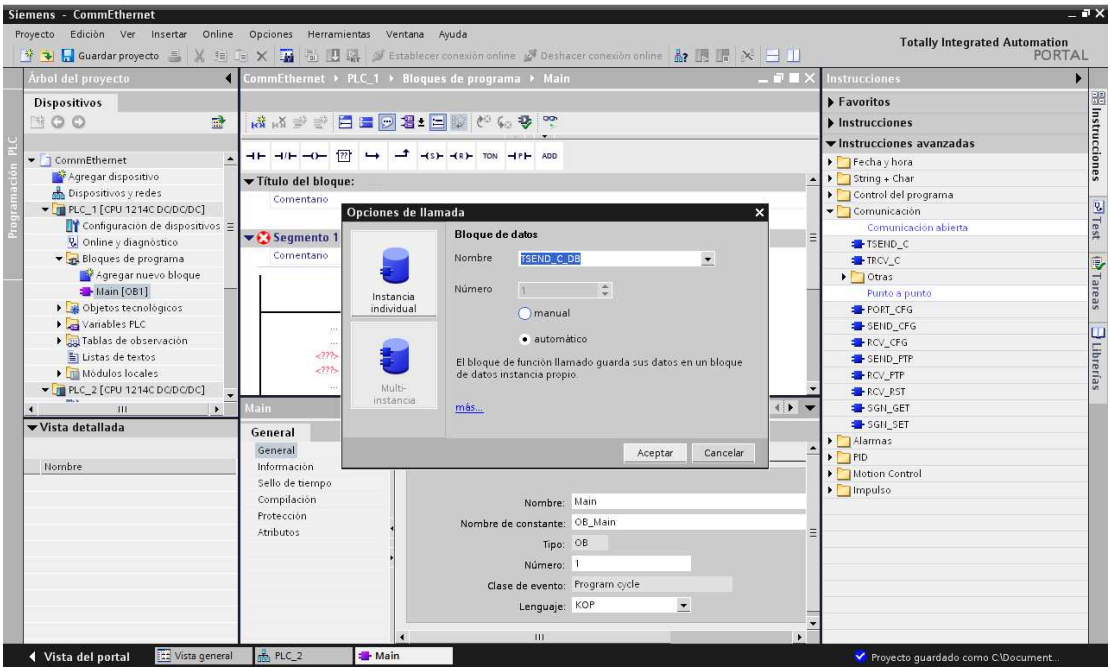

Figura 9. Insertar bloque TSEND_C

Asigne un nombre cualquiera al bloque, por ejemplo “Envío”.

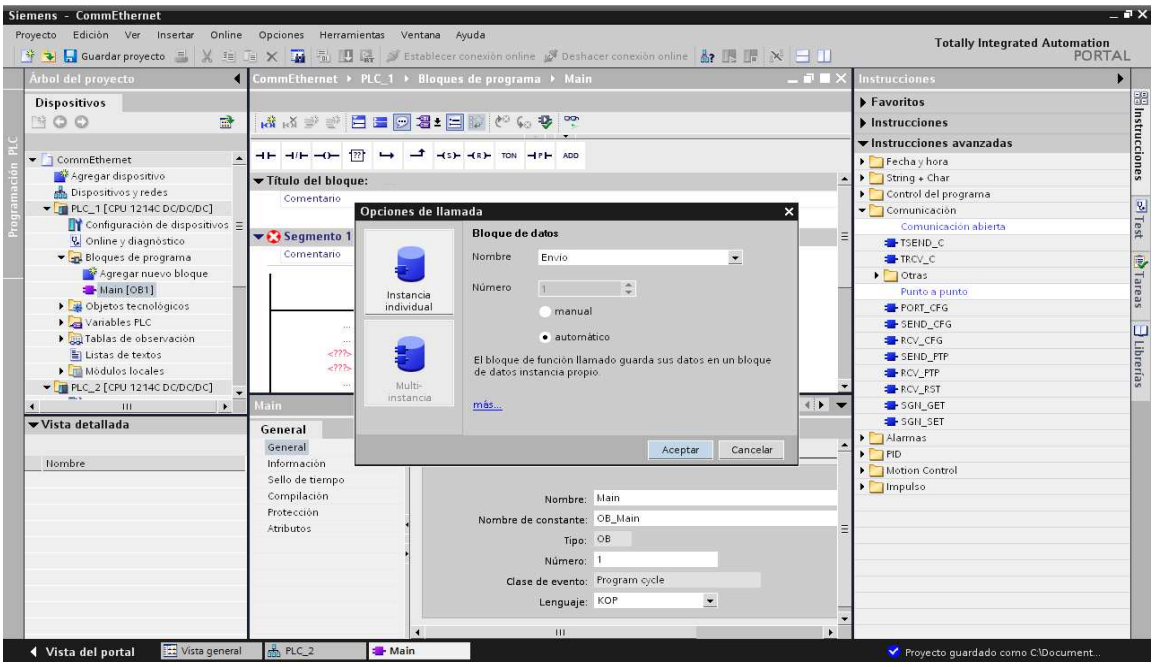

Figura 10. Cambiar nombre bloque TSEND_C

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

Una vez añadido el bloque de Envío podemos configurar los parámetros de la conexión. El campo “Local” va el PLC_1 del cual vamos a enviar los paquetes de datos, en “Subred” tenemos PN/IE_1 y en “Dirección” aparece la dirección IP correspondiente al PLC_1.

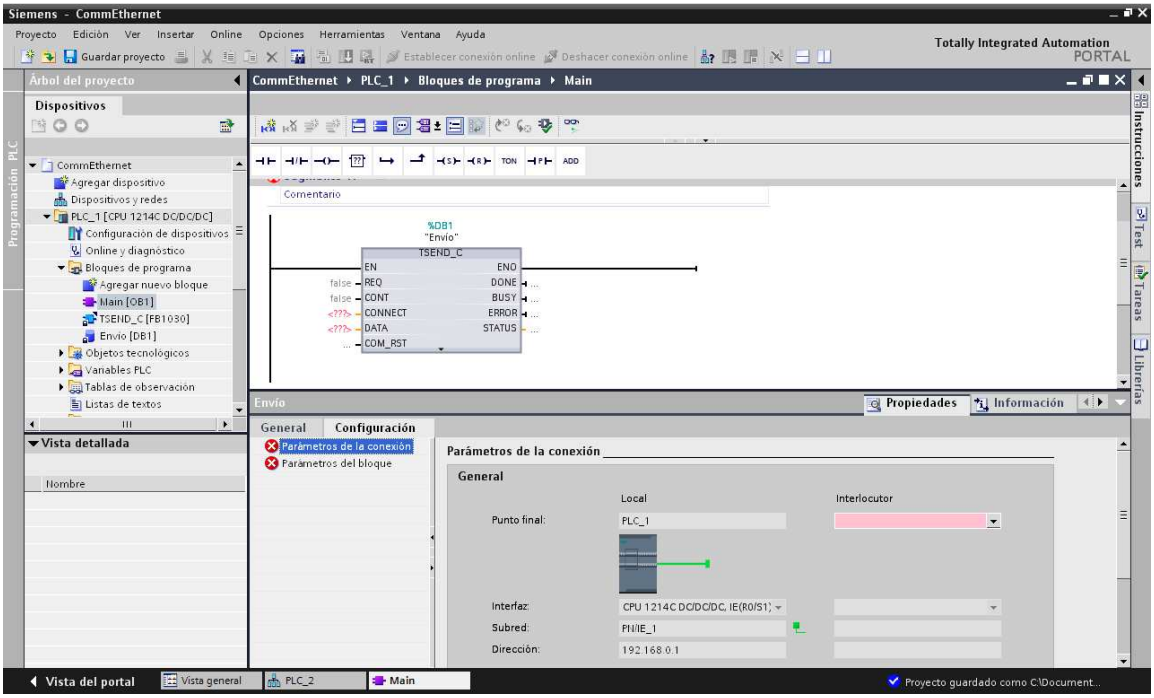

Figura 11. Parámetros de conexión bloque Envío

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

En el campo “Interlocutor” seleccionamos el PLC_2 al cual vamos a enviar los datos desde el PLC_1, se tiene la misma subred antes seleccionada y la correspondiente dirección IP del PLC_2.

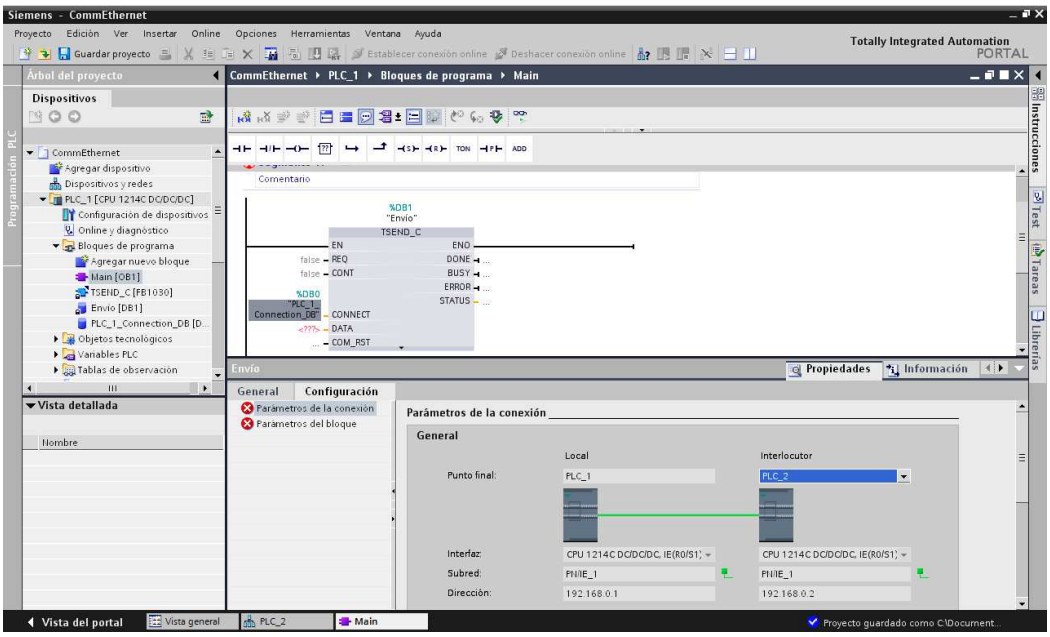

Figura 12. Configuración bloque de Envío.

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

Se debe determinar el tipo de conexión, en este caso “ISO on TCP”. En el campo “ID de conexión” elegir un número entero cualquiera, se sugiere uno pequeño. Este va a ser un identificador del enlace que se está realizando entre ambos PLCs, y debe ser único en el proyecto.

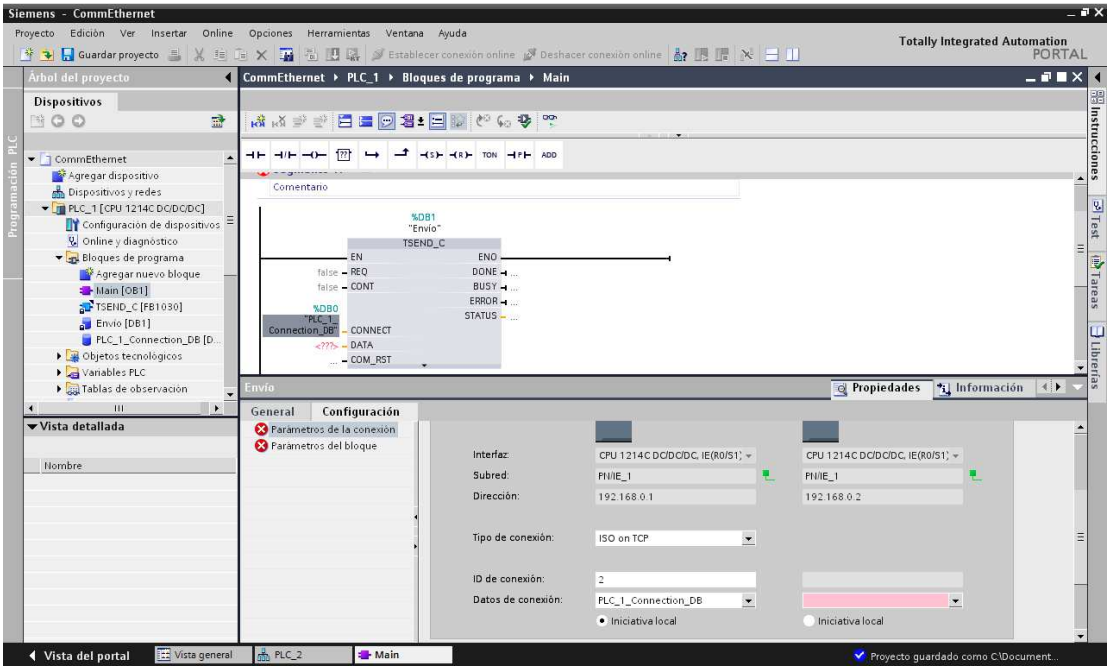

Figura 13. Configurar tipo de conexión PLC_1

Si se desea enviar la información a un PLC diferente a un S7-1200, entonces se debe elegir el interlocutor como “sin especificar” y simplemente escribir la IP del equipo que recibirá los datos. Esto se puede hacer únicamente en los siguientes casos:

El equipo interlocutor es otro PLC, por ejemplo un S7-300, que cumple el estándar Ethernet y se ha programado un bloque de recepción.

El equipo interlocutor es un computador u otra terminal compatible con el protocolo Ethernet y se han configurado correctamente las funciones para recibir o escuchar los datos, por ejemplo Labview o Matlab como se explicará más adelante en este tutorial.

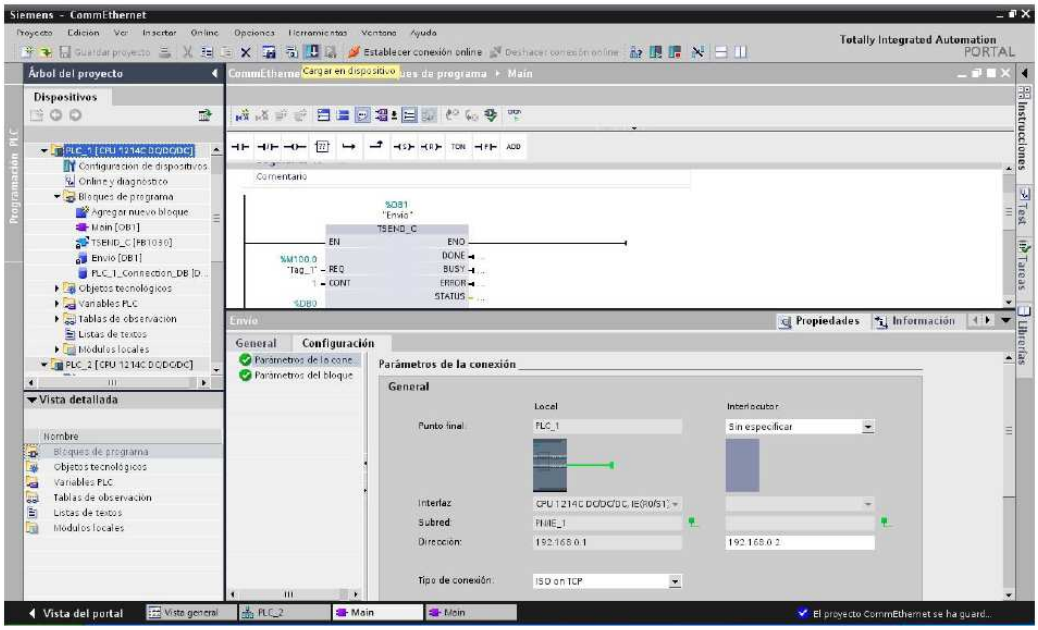

Figura 14. Interlocutor sin especificar

Siguiendo con los pasos, entonces en la pestaña de configuración nos situamos en la opción “Parámetros del bloque” y configuramos los parámetros:

“Inicio de la petición (REQ)”: La función de esta entrada es detectar un bit y, cada que se detecte un flanco positivo de éste, se hará el envío de datos. En este ejemplo asignamos una marca intermitente, que puede ser una marca de ciclo. Previamente en este proyecto se había configurado el byte de marcas 100 como marca de ciclo. Este procedimiento se explicó en la parte I del tutorial TIA. Al elegirse M100.0 se cuenta con una intermitencia de 10Hz.

En “Estado de la conexión (CONT)” en el campo CONT vamos a poner 1, para mantener la conexión y no desconectar automáticamente. Si en algún momento se desea cerrar la conexión, entonces aquí se debe ubicar una variable booleana sobre la que tengamos control, para encenderla o apagarla cuando se necesite.

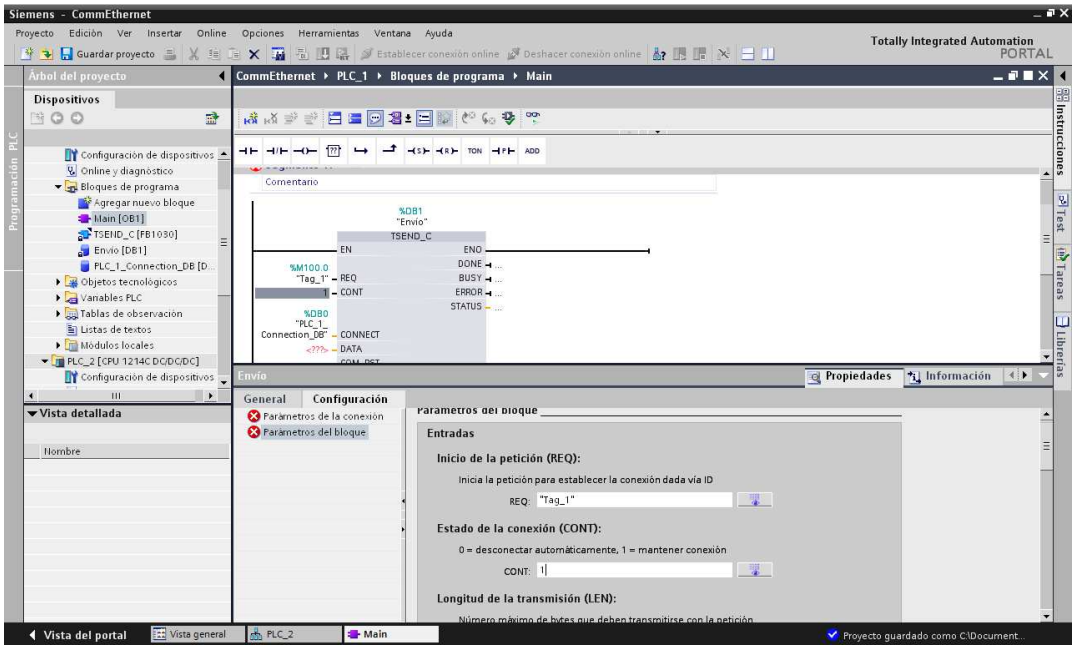

Figura 15. Configurar Estado de la Conexión PLC_1

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

Para nuestro caso vamos a enviar 2 bytes de datos del PLC_1 al PLC_2. En “Área de emisión (DATA)” en el campo “Inicio” ponemos una marca a partir de la cual se enviaran los datos, en “Longitud” se asigna el tamaño del dato y el tipo de dato, es decir, si serán bits, bytes, etc.

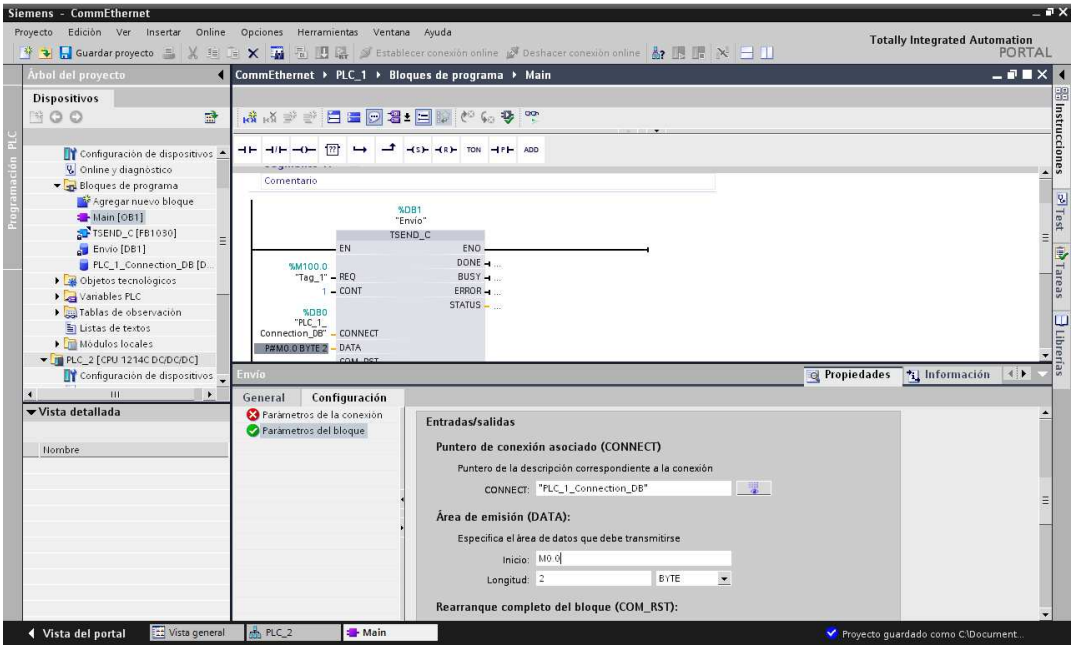

Figura 16. Dirección de los datos de envío

Ahora vamos a configurar la recepción de datos en el PLC_2. Vamos entonces a “Bloques de programa” del PLC_2, nos ubicamos en el Main [OB1] y buscamos en la parte lateral derecha el catálogo de “Instrucciones Avanzadas” en la carpeta llamada “Comunicación” obtenemos un bloque llamado TRCV_C, función con la cual haremos la recepción de los datos enviados desde el PLC_1.

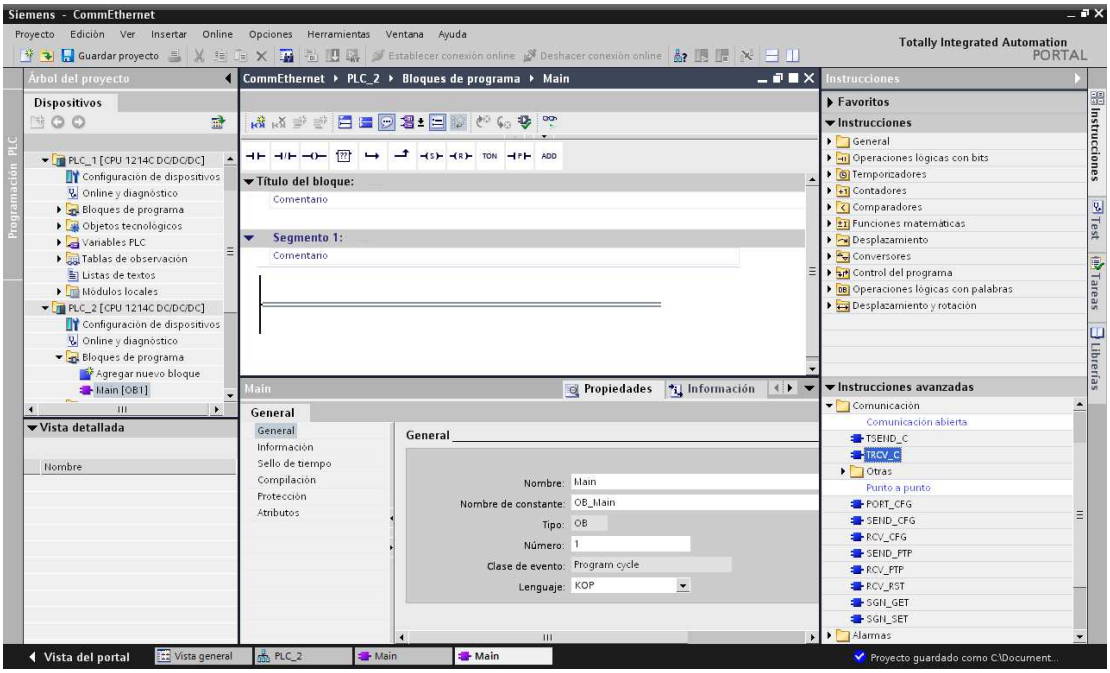

Figura 17. PLC_2 MAIN [OB1]

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

Arrastre el bloque de recepción TRCV a un segmento de programación del OB1.

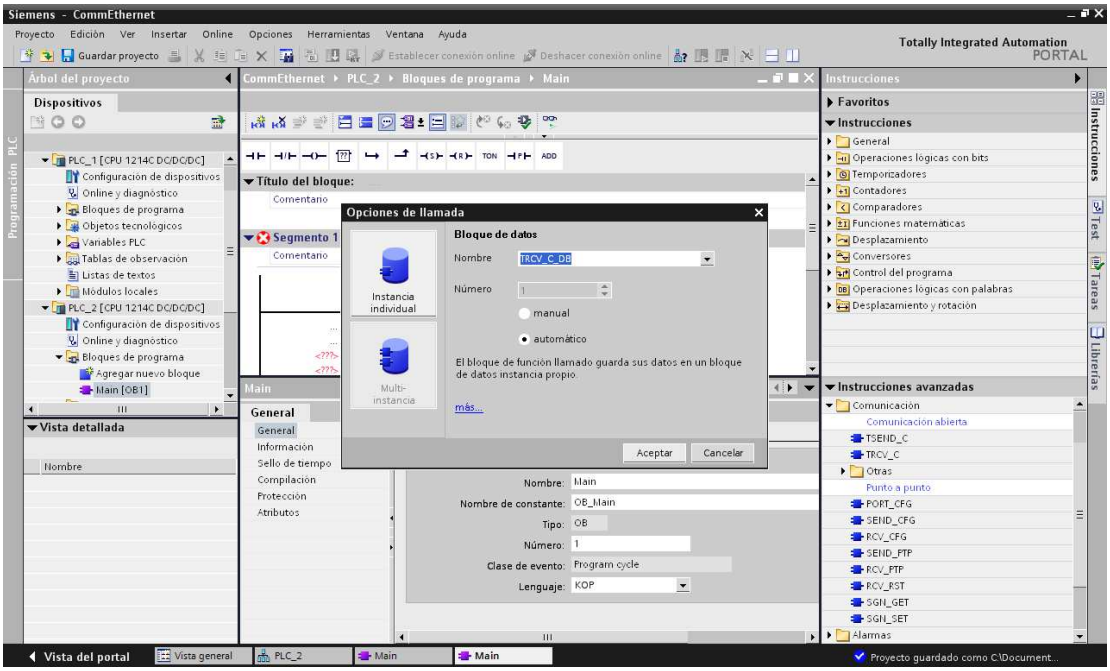

Figura 18. Insertar bloque TRCV_C

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

Asigne un nombre, por ejemplo “Recepción”

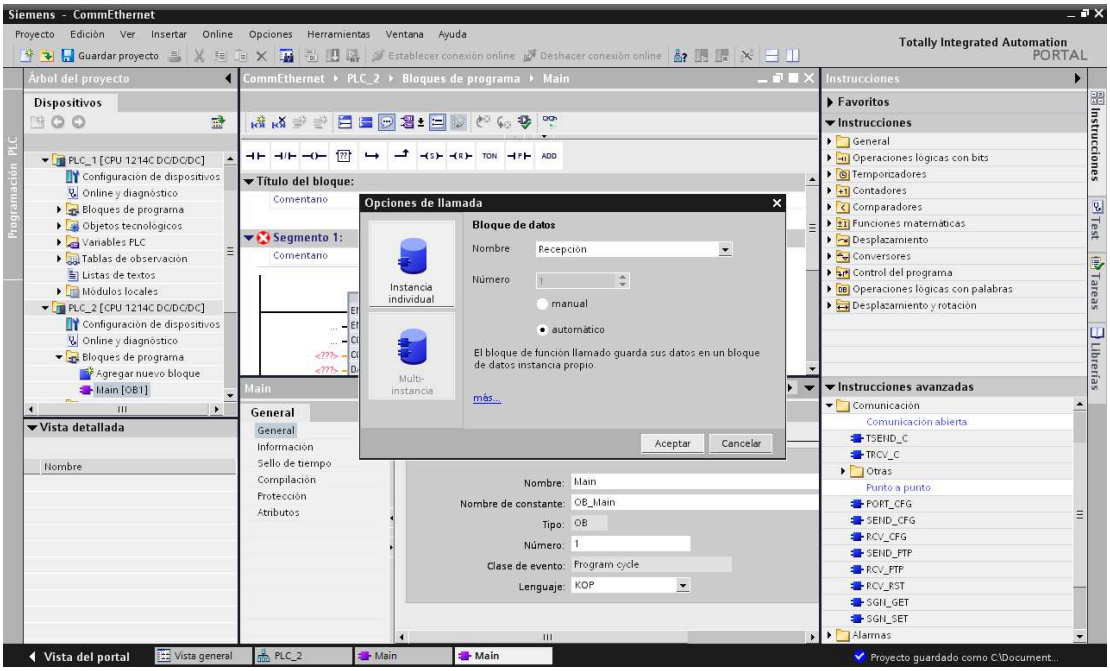

Figura 19. Cambiar nombre bloque TRCV_C

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

Seleccionando el bloque de Recepción podemos configurar los parámetros de la conexión. El campo “Local” va el PLC_2 en el cual vamos a recibir los paquetes de datos, el “Interlocutor” será el PLC_1 quien será en “Subred” tenemos la que ha sido asignada anteriormente y en “Dirección” aparece la dirección IP correspondiente al PLC_2. Recordemos que el “Tipo de conexión” para nuestro caso es “ISO on TCP”.

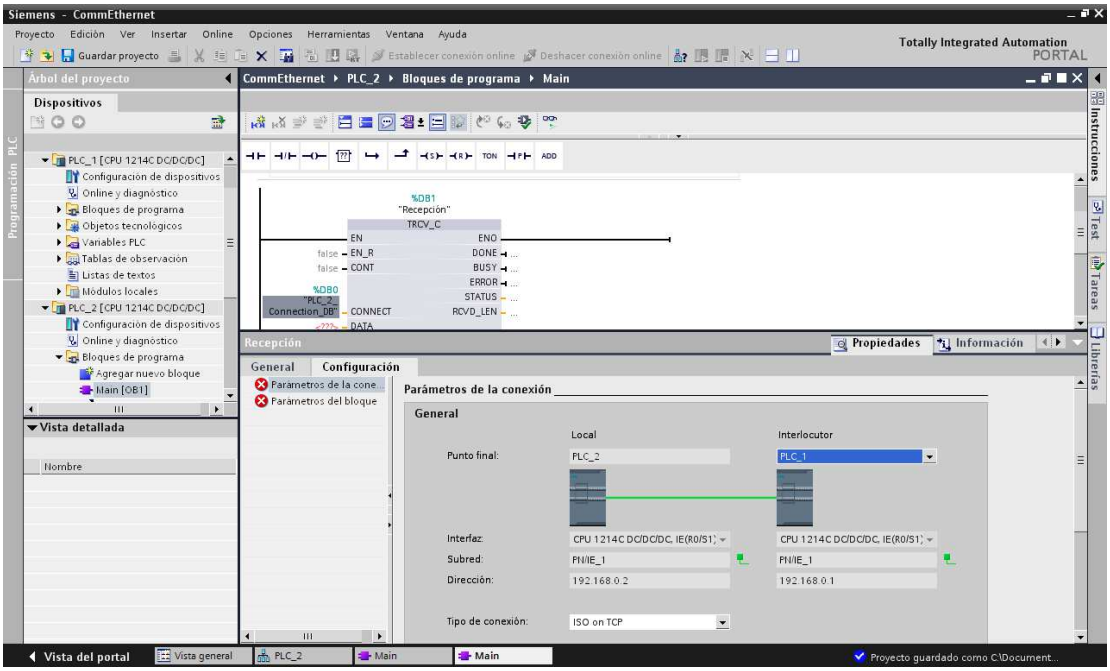

Figura 20. Configurar bloque de Recepción

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

En el ID de conexión escriba el mismo número que se había escrito en el mismo campo pero en el PLC_1, esto con el fin que haya unicidad en el ID en ambos extremos del enlace.

Elija los bloques de datos de ambos PLCs.

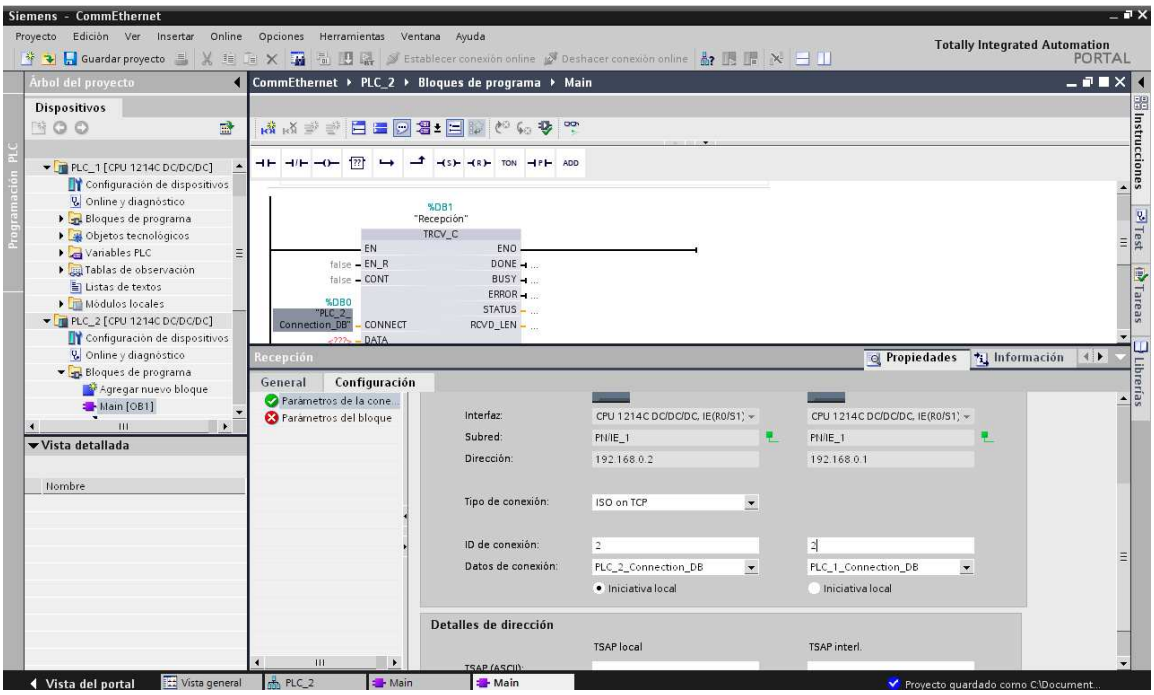

Figura 21. Configurar tipo de conexión PLC_2

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

En la pestaña de configuración nos situamos en la opción “Parámetros del bloque”. En el parámetro EN_R ubicamos una marca de ciclo. Lo más adecuado es que tenga la misma frecuencia que se configuró en el envío. Para el PLC_2 también se había configurado el byte 100 de marcas como marca de ciclo, por lo tanto lo más adecuado es utilizar también M100.0

En el campo CONT vamos a poner 1 igual que para el PLC_1, para mantener la conexión y no desconectar automáticamente.

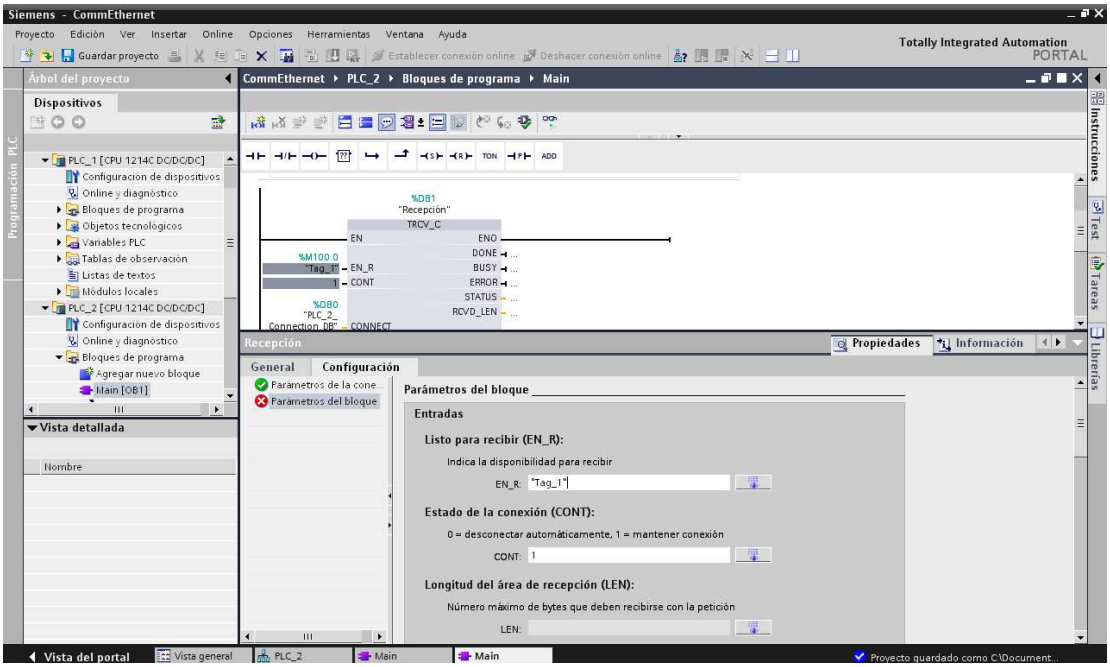

Figura 22. Configurar Estado de la Conexión PLC_1

Para nuestro caso vamos a recibir en el PLC_2, 2 bytes de datos provenientes del PLC_1. En “Área de emisión (DATA)” en el campo “Inicio” ponemos una marca a partir de la cual se recibirán los datos, en “Longitud” se asigna el tamaño del dato y el tipo de dato, es decir, si serán bits, bytes, etc. Para nuestro caso 2 bytes, como ya se configuró en el envío del PLC_1.

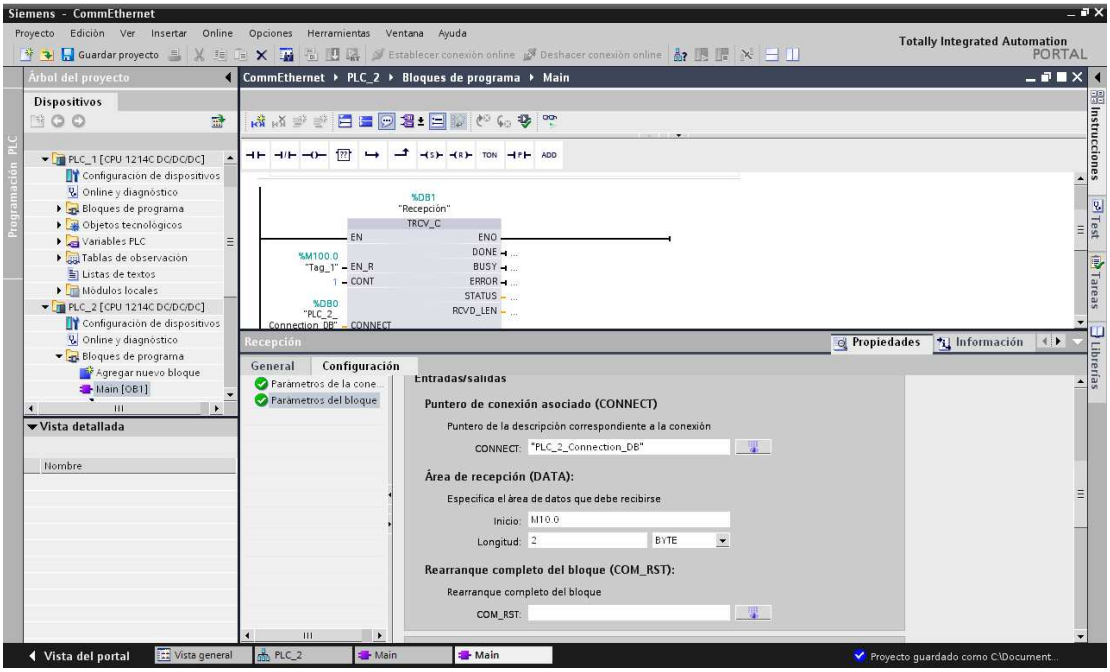

Figura 23. Dirección de los datos de recepción

El siguiente paso es guardar el proyecto para que, si ocurre algún bloqueo o corte eléctrico, la información quede respaldada.

Luego cargar a ambos PLCs y probar los resultados.

4. COMUNICACIÓN DE PLC CON MATLAB USANDO ETHERNET

Asegúrese de contar con un proyecto TIA con un PLC configurado correctamente y una dirección IP asignada, por ejemplo 192.168.0.1 la cual trae por defecto el PLC.

Configure el bloque de envío de datos TSEND como se mostró en la Figura 9, pero no defina un interlocutor, únicamente asigne una IP para el envío de datos como se mostró en la Figura 14 en este tutorial.

Defina la IP del computador que tiene Matlab instalado, por ejemplo la 192.168.0.2. Por el panel de control de ese computador y en las opciones de conexión de área local, configure esta IP para la tarjeta de red.

Para configurar la comunicación desde el software Matlab utilizaremos el comando **tcpip ,** esta función nos construye un objeto TCPIP, se asocia el host remoto que en este caso es el PLC y configuramos el valor del puerto remoto. Por último se habilita el soporte para sockets de servidor. Un socket (enchufe), es un método para la comunicación entre un programa del cliente y un programa del servidor en una red.

A continuación se muestra un ejemplo de la utilización del comando **tcpip**:

```
>> conection=tcpip('192.168.0.1',2000,'NetworkRole','Server')
```

Con lo anterior se iniciar el servidor TCP / IP y se crea un objeto TCPIP. El primer parámetro de entrada es la dirección IP del PLC con el cual nos vamos a comunicar.

El segundo parámetro de entrada es el valor del puerto remoto. Para entender este concepto debemos saber que en las redes que utilizan los protocolos TCP/IP, cuando un programa cliente necesita de un servicio particular de un servidor, además del tipo de servicio y localización del servidor, debe indicar el puerto por el que se establecerá la conexión. Un puerto es un extremo de una conexión lógica, son indicados por números. Dichos números se asignan dependiendo de los siguientes rangos:

Puertos bien conocidos ("Well known ports"), comprendidos entre 0 y 1023. Estos 1024 (2^{10}) puertos pueden ser representados con 10 bits y son reservados para servicios conocidos.

Puertos registrados ("Registered ports"). 48127 puertos comprendidos entre 1024 y 49151.

Puertos dinámicos y privados. Los comprendidos entre los números 49152 y 65535.

Para poder utilizar un servicio de un servidor es necesario que el puerto correspondiente sea el correcto y que esté habilitado, así podemos decir que el servidor debe estar "escuchando" por dicho puerto. En este caso se ha asignado el puerto 2000 que corresponde a un puerto registrado.

En el tercer parámetro de entrada tenemos la propiedad NetworkRole, como se dijo anteriormente, esta propiedad configura el computador como cliente o servidor de la red, en este caso el valor de la propiedad es 'Server', es decir que el computador operará como el servidor.

Luego de definir el objeto TCP/IP, se procede a conectar el objeto con el host, por medio del siguiente comando:

```
>> fopen(conection)
```

El paso a seguir luego de configurar la comunicación es saber si existen datos disponibles en el buffer, para que no se generen errores al momento de la lectura. Lo anterior se hace preguntando por el valor de la propiedad **BytesAvailable** que indica el número de bytes disponibles en la actualidad para ser leídos de la entrada del buffer. El valor de la propiedad se actualiza continuamente cuando el buffer de entrada se llena, y se pone a 0 después de que la función fopen se emite. El valor de BytesAvailable está entre un rango desde cero hasta el tamaño de la entrada del buffer. Si se quisiera saber el tamaño se utiliza la propiedad InputBufferSize para especificar el tamaño de entrada del buffer. Utilice la propiedad ValuesReceived para devolver el número total de valores leídos. Pero para nuestro caso solo necesitamos saber si hay datos disponibles para la lectura, entonces con la propiedad BytesAvailable basta. Para nuestro ejemplo se usará como se indica en la siguiente sentencia:

```
>> if conection.BytesAvailable~=0
```

Luego de saber si hay datos disponibles, procedemos a leer dichos datos del buffer, para esto utilizaremos el comando **fread** que lee el número de elementos de la variable 'conection', especificados por el segundo parámetro de entrada, que es el tamaño, en este caso 1, de la siguiente manera:

```
>> fread(conection,1)
```

Para nuestro ejemplo el dato que nos envía el PLC está en formato decimal, nosotros lo convertiremos en formato binario, para lo que se empleará el comando **de2bi** en el que especificamos cuántos dígitos queremos que nos arroje, para nuestro ejemplo esperamos un byte, entonces en el segundo parámetro de entrada, ponemos '8', es decir 8 bits, así:

```
>> dato=de2bi(fread(conection,1),8)
```

Rotamos el vector para hacer que el LSB (Bit menos significativo) pase a ser el MSB (Bit más significativo), debido a que el dato llega invertido.

```
>> dato=dato(end:-1:1)
```

Como la comunicación para el tipo de configuración de hardware hecha en el PLC es bidireccional, es necesario también enviar datos para que sean recibidos en la CPU del S7-1200. Usando el comando **fwrite** escribimos al objeto TCP/IP definido como 'conection', en este caso enviaremos un dato aleatorio con valores entre 0 y 255, convirtiéndolo previamente en enteros sin signo de 8 bits, de la siguiente manera:

```
>> fwrite(conection,uint8(randi([0 255])))
```

A continuación se presenta el código completo con sus respectivos comentarios, en el cual se lee un byte de datos enviado desde el PLC y se envía un dato de valor aleatorio de 0 a 255.

```
Instrreset  %Desconecta y elimina todos los objetos creados. Si los datos

% se escriben o se leen de forma asíncrona, la operación

% asincrónica se detiene.

conection=tcpip('192.168.0.1',2000,'networkrole','server')% Construye un
%objeto TCPIP, asociado a un host remoto al cual pertenece %
la dirección IP seleccionada y se configura el valor del puerto remoto.  % Por último se habilita el soporte para sockets de servidor.

fopen(conection) %Conectar el objeto TCP/IP con el host remoto

while(1)

if conection.BytesAvailable~=0    % Validar si existen datos

            disponibles en el buffer para la lectura

    dato=de2bi(fread(conection,1),8); % Leer datos del buffer, convertirlos a formato binario tomando 8 bits para la lectura.

    dato=dato(end:-1:1);  % Rotar el vector dato, el LSB pasa a ser el MSB, debido a que el dato leído en el buffer llega invertido.

    fwrite(conection,uint8(randi([0 255])));  % Escribir datos al host remoto, en este caso un valor aleatorio entre 0 y 255.

  end

  drawnow(); % Actualiza los valores en cada ciclo del while, evitando que se bloquee la aplicación por ejecutarse en períodos muy cortos de tiempo.

end
```

5. COMUNICACIÓN DE PLC CON MATLAB USANDO ETHERNET CON ENTORNO GRÁFICO

En el siguiente ejemplo se hará la misma conexión que se realizó en el capítulo anterior usando un computador con el software Matlab y un PLC S7-1200 de Siemens.

A continuación se presenta el código correspondiente:

```
function varargout=prueba_plc(varargin)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if ~isempty(findobj('tag','tcp'))                    % Encontrar el objeto con la propiedad
                                                     % especifica, en este el tag de la figura para
                                                     % saber si ya se encuentra abierta

warndlg('Señor usuario la aplicación esta ejecutada','Advertencia')    % De ser asi mostrar una ventana de advertencia, para decir
                                                     % al usuario que la aplicación ya se ejecutó.

  return

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

hg.fig(1)=figure('Name','Matlab_TCP/IP_PLC','menubar','none',...     % Crear la figura principal especificando caracteristicas como el titulo, posición, tamaño, color, entre otras

'Resize','Off','numbertitle','off','position',[0 0 500 300],...

'color',[1 1 1],'tag','tcp','CloseRequestFcn',@closereq);

 movegui(hg.fig(1),'center');                        % Centrar la figura principal

hg.bot(1)=uicontrol('parent',hg.fig(1),'style','pushbutton',...     % Crear boton que permita hacer la conexión con el PLC, especificar posicion, tamaño, etiqueta, entre otras

'position',[30 250 100 20],'string','Conectar','backgroundcolor',...
```

```
[125 221 255]./256,'callback',@conectar);

 %Interfaz Lectura

hg.text(1)=uicontrol('parent',hg.fig(1),'style','text','position',...     % Crear texto de indicación para los bits de lectura

[30 200 100 20],'string','Lectura','backgroundcolor',[125 221 255]./256);

for l=1:8

  hb.frm(l)=  uicontrol('parent',hg.fig(1),'style','text',...                % Crear 8 cuadros de texto que indicarán el estado de los bits que serán leídos, deshabilitados para los usuarios.

  'Backgroundcolor',[125 221 255]./256,'position',...

  [10+(l-1)*60 150 50 20],'string',num2str(l),'Enable','off');

end

hg.read(1)=uicontrol('parent',hg.fig(1),'style','text','position',...     % Crear texto de indicación del valor de lectura en formato decimal

[150 200 100 20],'backgroundcolor',[1 1 1],'Enable','Off');

 %Interfaz Escritura

hg.text(2)=uicontrol('parent',hg.fig(1),'style','text','position',...     % Crear texto de indicación para los bits de escritura

[30 100 100 20],'string','Escritura','backgroundcolor',...

[125 221 255]./256);

for k=1:8

  hb.push(k)= uicontrol('parent',hg.fig(1),'style','togglebutton',...   % Crear botones que se pulsarán para enviar bit a bit el dato que se transmitirá al PLC

  'Backgroundcolor',[125 221 255]./256,'position',...

  [10+(k-1)*60 50 50 20],'string',num2str(k));

end

hg.write(1)=uicontrol('parent',hg.fig(1),'style','text','position',...     % Crear texto de indicación del valor de escritura en formato decimal

[150 100 100 20],'backgroundcolor',[1 1 1],'Enable','Off');
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%

 clc                                            % Clear command window

 instrreset                                     % Desconectar y eliminar todos los objetos previamente creados

 conection=tcpip('192.168.0.1',2000,'networkrole','server')      % Construye un objeto TCPIP, asociado a un host remoto al cual pertenece

                                                % la direccion IP seleccionada y se configura el valor del puerto remoto.

                                                % Por último se habilita el soporte para sockets de servidor.

 fopen(conection)                               % Conectar el objeto TCP/IP con el host remoto


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%

%%

function varargout=conectar(varargin)           % Rutina que se ejecuta cuando el objeto está siendo activado, en este caso el boton conectar

    while(1)

     if conection.BytesAvailable~=0             % Validar si existen datos disponibles para la lectura en el buffer

%Lectura

      dato=de2bi(fread(conection,1),8);         % Leer datos del buffer, convertirlos a formato binario tomando 8 bits para

                                                % la lectura.

      dato_read=bi2de(dato);                    % Convertir el dato leído de formato binario a decimal

      set(hg.read(1),'string',double2str(dato_read))    % Definir el string del texto visualizacion de lectura con el valor decimal correspondiente
```

```
    dato=dato(end:-1:1);                % Rotar el vector dato, el LSB pasa a ser el MSB,
debido a que el dato leido
                                        % en el buffer llega invertido.
   for n=1:8                            % Ciclo For


    lcolor=dato(n);                     % Obtener valor del bit correspondiente al ciclo
actual de la estructura For
    if lcolor                           % Si este valor es '1'
      set(hb.frm(n),'backgroundcolor','b');          % Definir la propiedad color como azul,
del bloque correspondiente al bit actualmente leído en el For
    else                                % De lo contrario
      set(hb.frm(n),'backgroundcolor','c') ;         % Definir la propiedad color como cyan,
del bloque correspondiente al bit actualmente leído en el For
    end                                 % Fin del if
   end                                  % Fin del for
%Escritura
    dat_bin=cell2mat(get(hb.push(1:end),'value'))';  % Obtener los estados de todos los
botones correspondientes a los bits de escritura,
                                        % convertirlos en una matriz ya que se obtiene una celda
inicialmente
    dato_write=bi2de(dat_bin(end:-1:1));            % Convertir de formato bintario a
decimal
    set(hg.read(1),'string',double2str(dato_read))  % Definir el string del texto
visualizacion de escritura con el valor decimal correspondiente
    fwrite(conection,uint8(dato_write));            % Escribir datos al host remoto, en este
caso uel valor leido de los botones.


    end
```

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

```
      drawnow();                              % Refresca los eventos pendientes en el entorno
gráfico, para evitar posibles bloqueos

    end

   end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%

function varargout=closereq(hObject,~)                  % Rutina para cerrar forzosamente la
interfaz desde el Exit principal

     close force

     close all

end


end
```

La interfaz que se crea con el código anterior es la que se puede observar en la

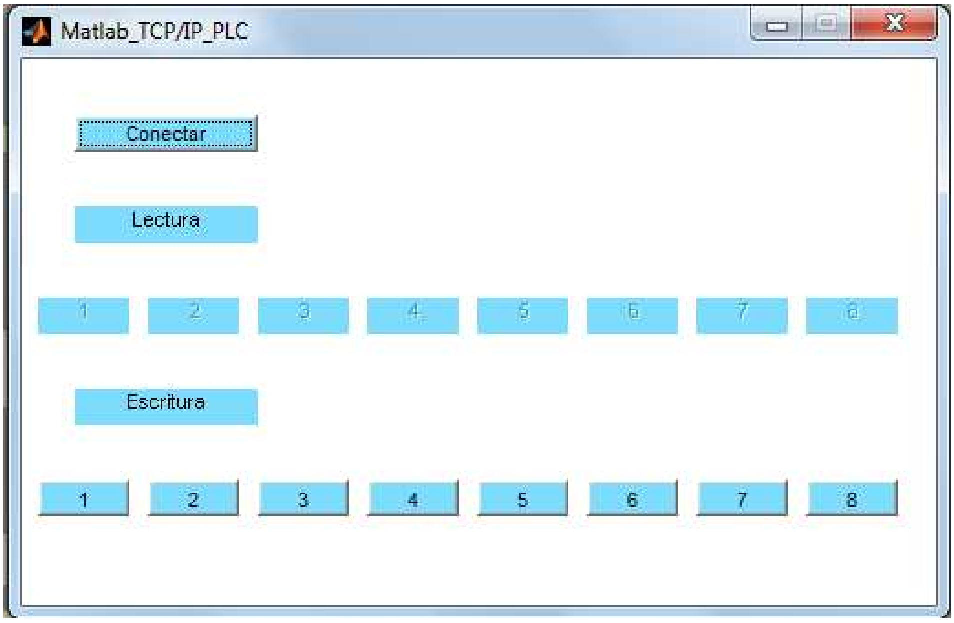

Figura 24. Interfaz Matlab

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

6. COMUNICACIÓN DE PLC CON LABVIEW USANDO ETHERNET

Como se observó en el Capítulo 4. COMUNICACIÓN DE PLC CON MATLAB USANDO ETHERNET se configura el envío y recepción del PLC S7-1200, se asigna la correspondiente dirección IP a la CPU. Recordar que en este caso el Interlocutor es un dispositivo *"Sin especificar".*

El primer paso que se debe llevar a cabo es crear un nuevo VI.

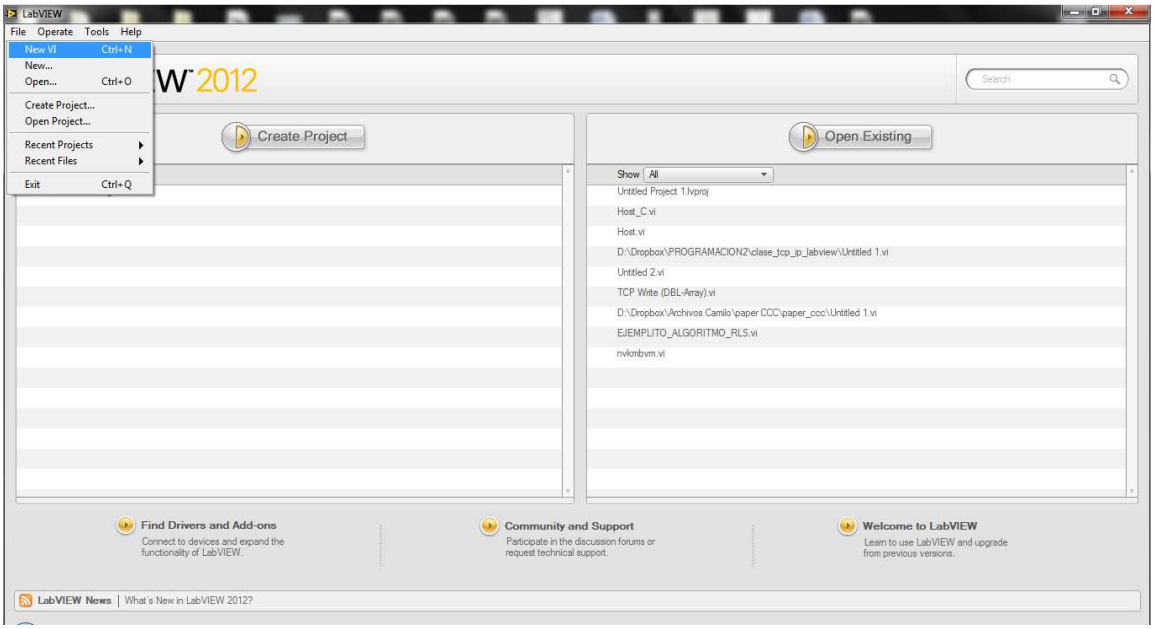

Figura 25. Insertar Nuevo VI

Abrir las librerías de comunicación "*Data communication*" seleccionamos "*Protocolos*" seleccionamos la librería protocolo "*TCP*", dejamos anclado este espacio de la librería para utilizar posteriormente los bloques que sean necesarios.

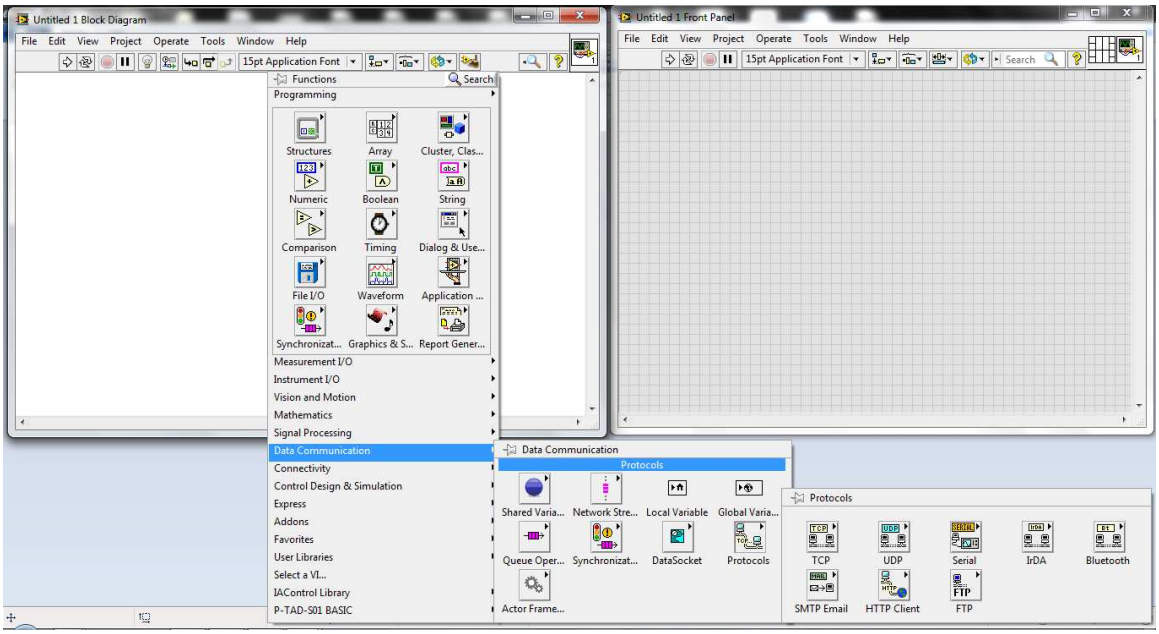

Figura 26. Librería TCP

Agregar una estructura “While Loop”, para que nuestro programa esté en un ciclo en el cual un conjunto de instrucciones se ejecutan mientras la condición del While permanezca como verdadera en el momento en que la condición se convierte en falsa el ciclo termina.

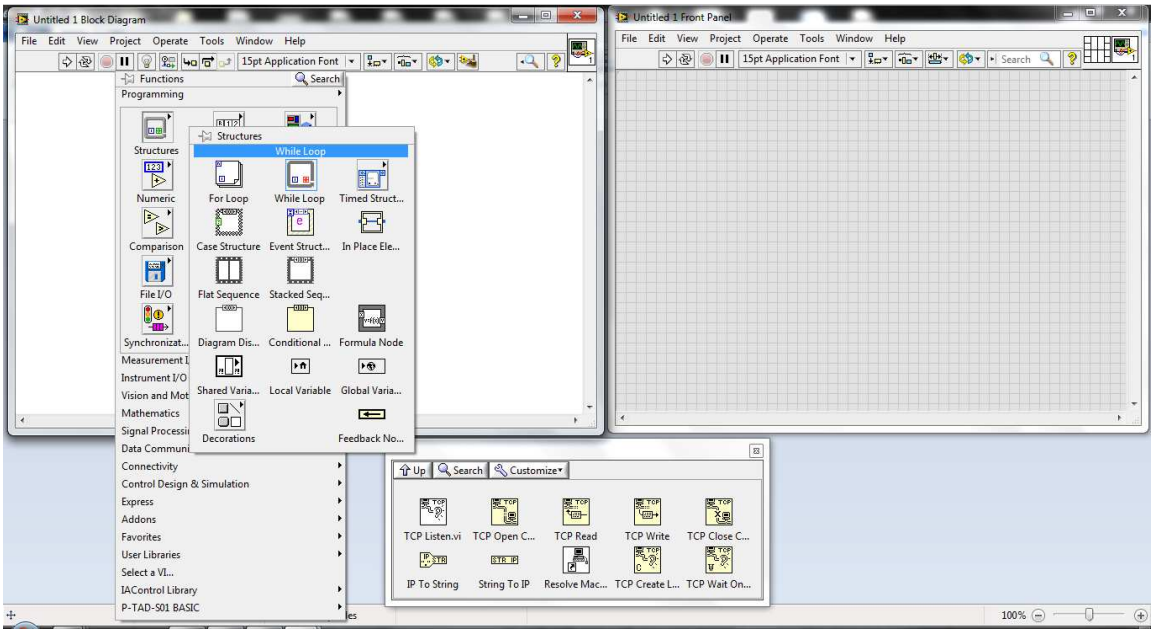

Figura 27. While Loop

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

El ciclo While ejecuta el código que contiene hasta la terminal condicional, una terminal de entrada que recibe un valor Booleano específico, en este caso FALSE con la finalidad de que no sea el usuario quién detenga el ciclo, sino que se ejecute indefinidamente.

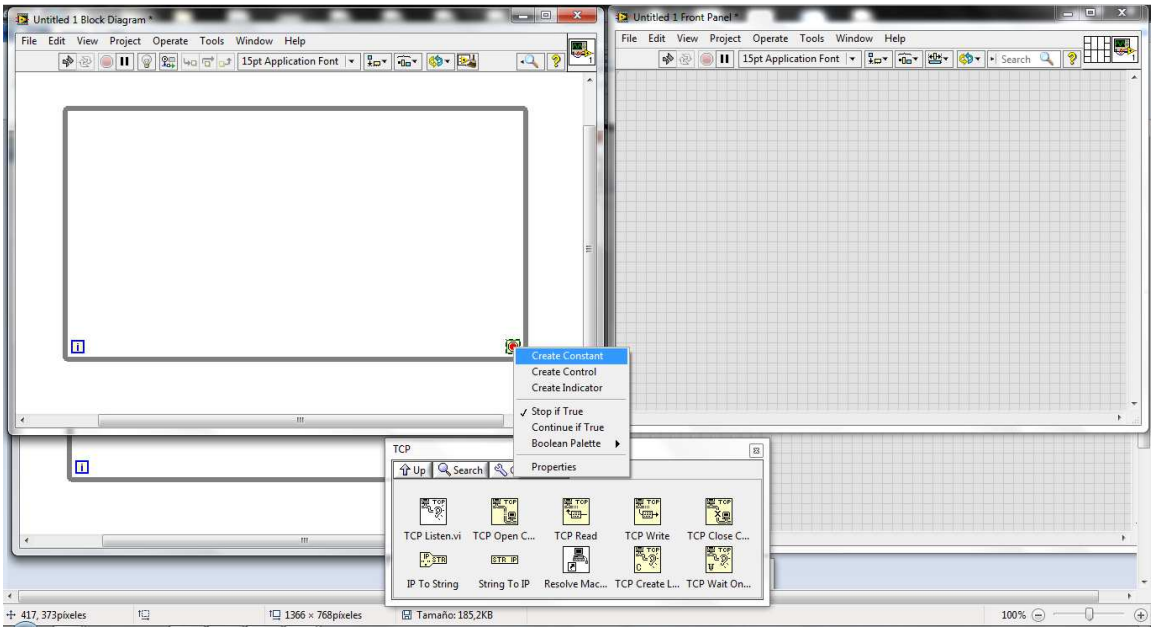

Figura 28. Condicional While Loop

Agregamos el bloque *"TCP Listen"* que crea un agente de escucha y espera una conexión de red TCP aceptada en el puerto especificado.

La entrada *port* por medio de la cual se establece el puerto a través del cual se realizara la comunicación con la otra estación o dispositivo, no se debe olvidar que ambas estaciones deben tener configurado el mismo número de puerto, la salida *conecction ID* nos suministra un identificador para hacer accesos a esta conexión una vez establecida, esta salida debe estar cableada a todos los bloques TCP involucrados en la comunicación con el fin de garantizar claridad respecto a cual canal de comunicación usar, la salida *error out* informa errores surgidos en el proceso de conexión TCP, finalmente tenemos la entrada *timeout ms* por medio de la cual se establece el lapso de tiempo durante el cual se esperará por una conexión TCP entrante, si transcurrido el tiempo programado no se ha realizado una llamada de otra estación este conector generará un error.

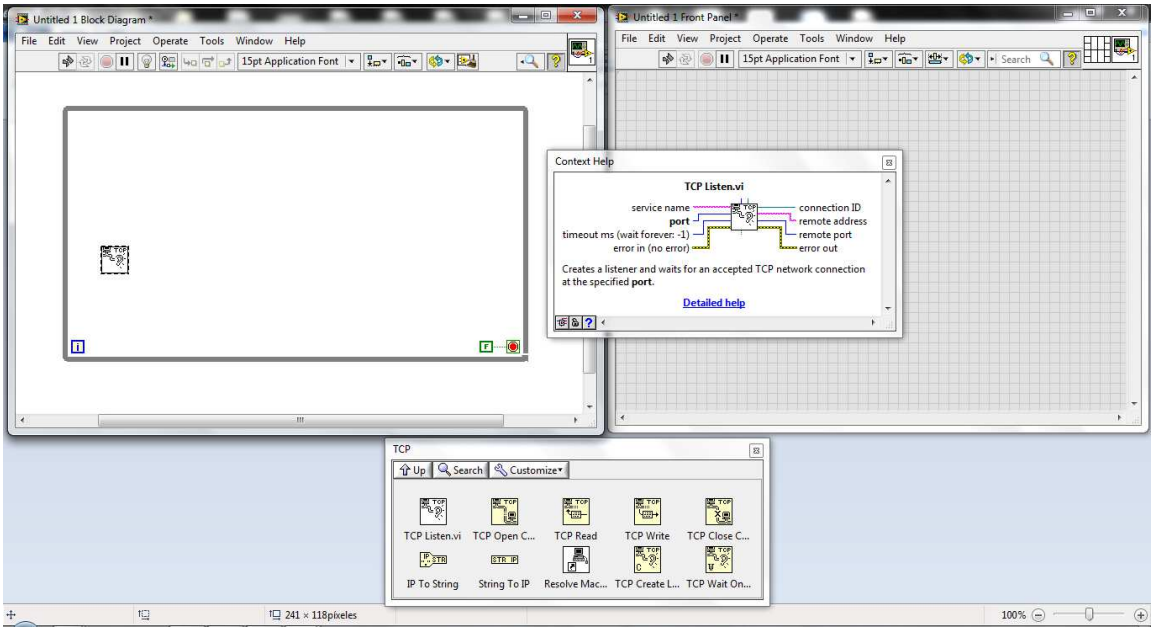

Figura 29. Insertar Bloque TCP Listen

Definir puerto como una constante 2000. En la sección COMUNICACIÓN DE PLC CON MATLAB USANDO ETHERNET de este tutorial se explicó de forma detallada porque se escoge este valor para el puerto.

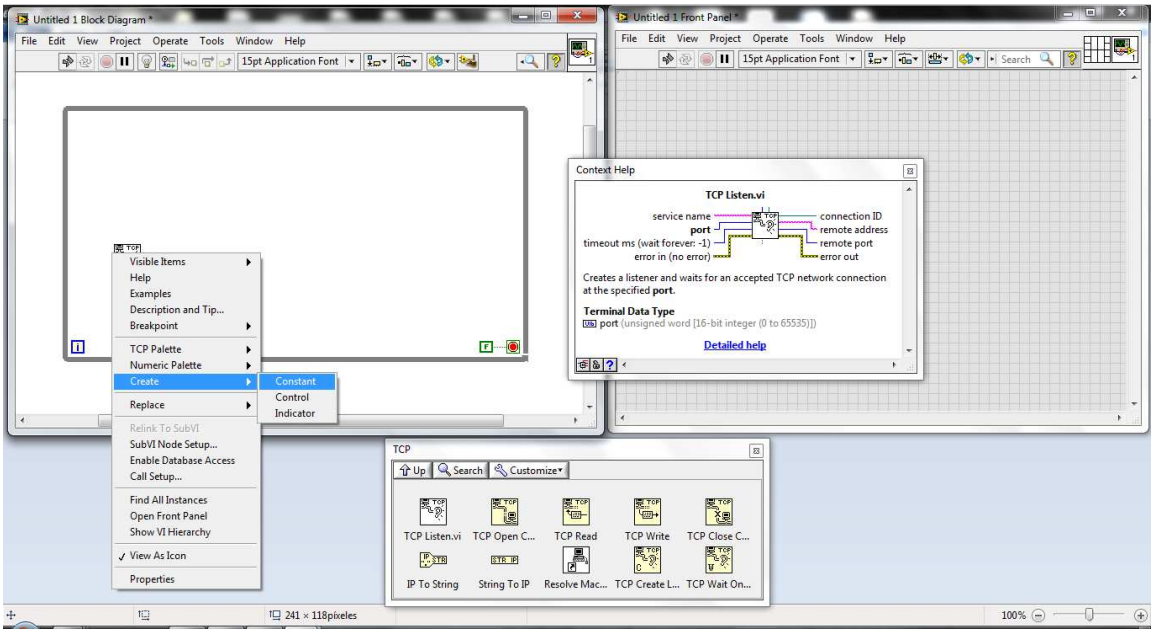

Figura 30. Configurar Puerto

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

Insertamos una estructura tipo "*Case Structure*", con la que validaremos si existen errores de comunicación, por medio de la línea *error out d*el bloque *TCP Listen.*

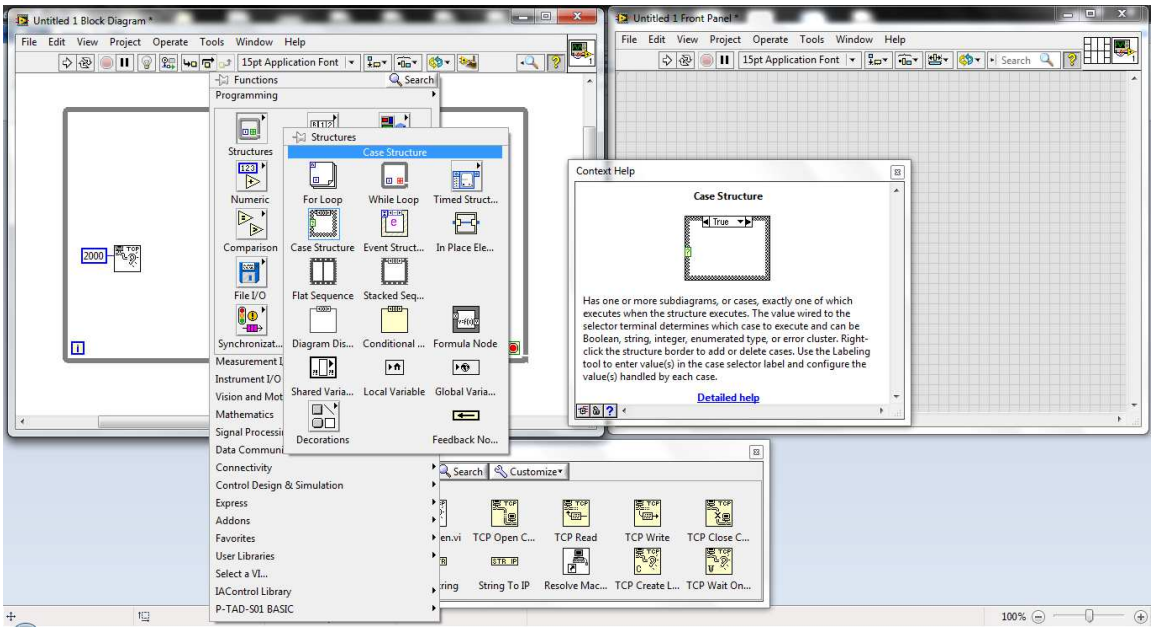

Figura 31. Estructura tipo *"Case"*

Se conecta el *Cluster* de error, es decir su línea correspondiente a la estructura *Case* por si existe un error en la lectura del puerto.

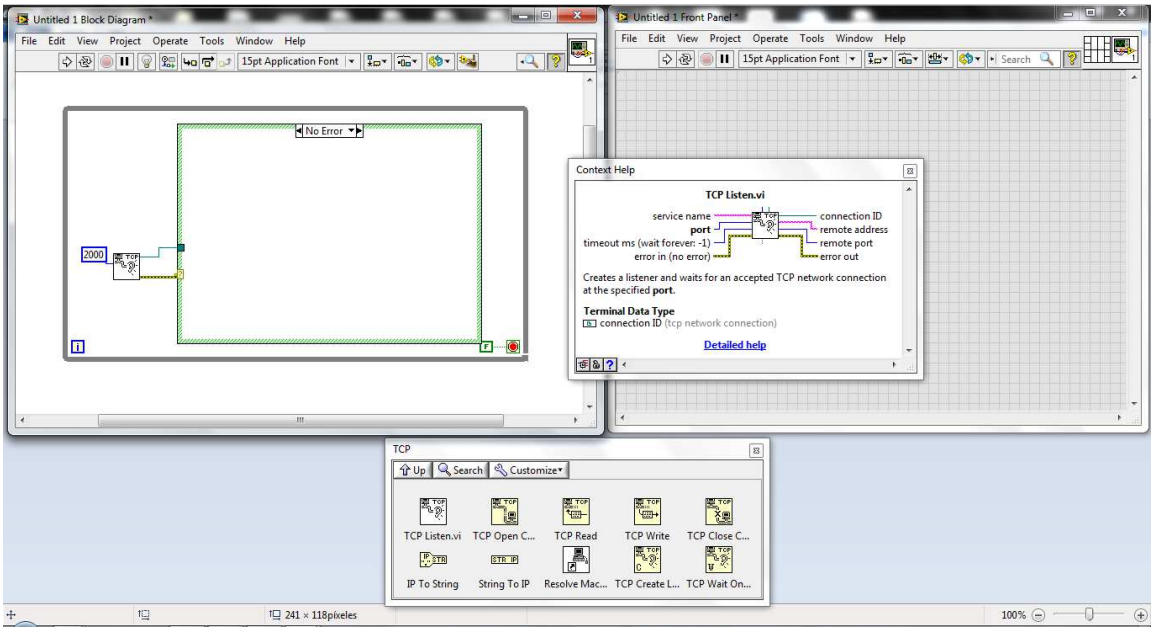

Figura 32. Cluster de error

Agregamos un ciclo while dentro de la estructura *Case* en el caso de que no existan errores de comunicación, es decir en *No Error,* ya que cumpliéndose esta condición se puede ejecutar el programa constantemente.

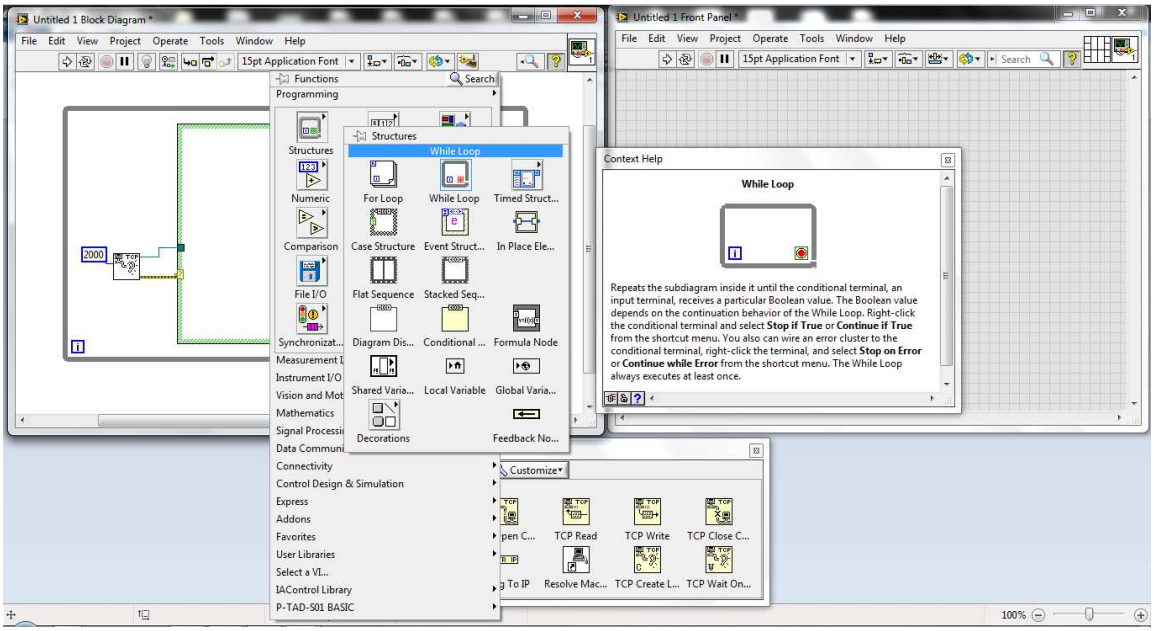

Figura 33. Agregar Estructura While

Como se hizo anteriormente en la estructura *Case* se conecta el cluster del error al ciclo while agregado.

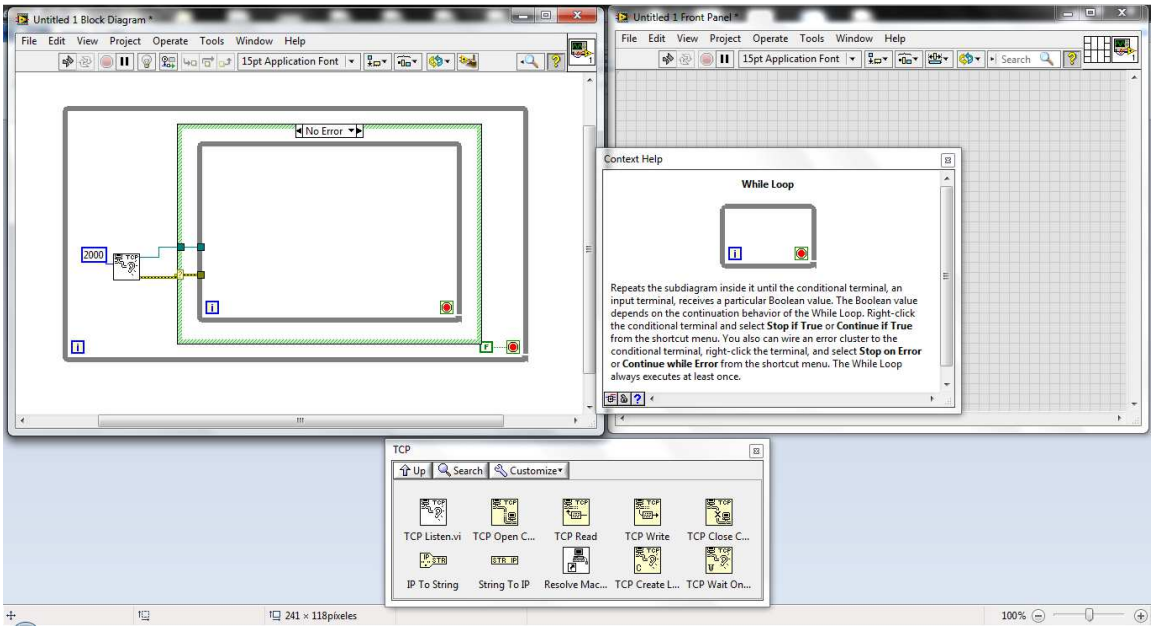

Figura 34. Cluster de Error While Loop

Si se produce un error en la comunicación se cierra el puerto mediante el bloque "*TCP Close Connection*"

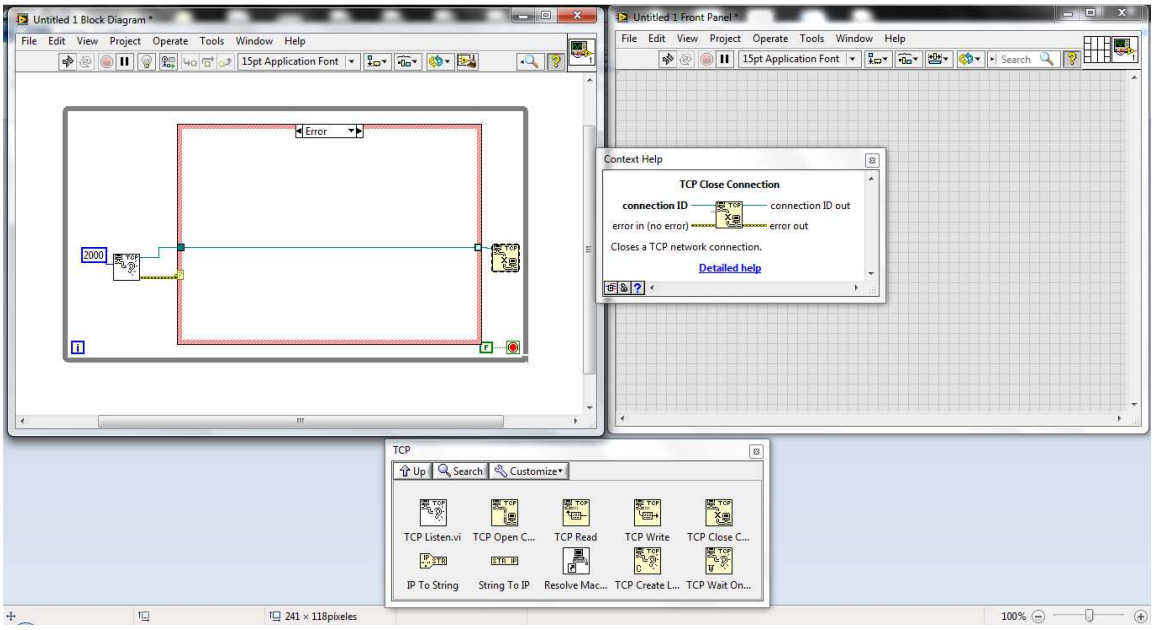

Figura 35. TCP *Close Connection*

En caso de no haber errores de comunicación se procede a escribir mediante el bloque "*TCP Write*" (Escribe datos en una conexión de red TCP) y realizar lectura mediante el bloque "*TCP Read*" (Lee el número de bytes de la conexión de red TCP, devolviendo los resultados los datos recibidos).

Desde que no exista un error de comunicación no se romperá el ciclo while en el cual se está llevando a cabo la escritura y la lectura, en caso de que suceda un error inmediatamente se detiene la comunicación cerrando la conexión de red TCP.

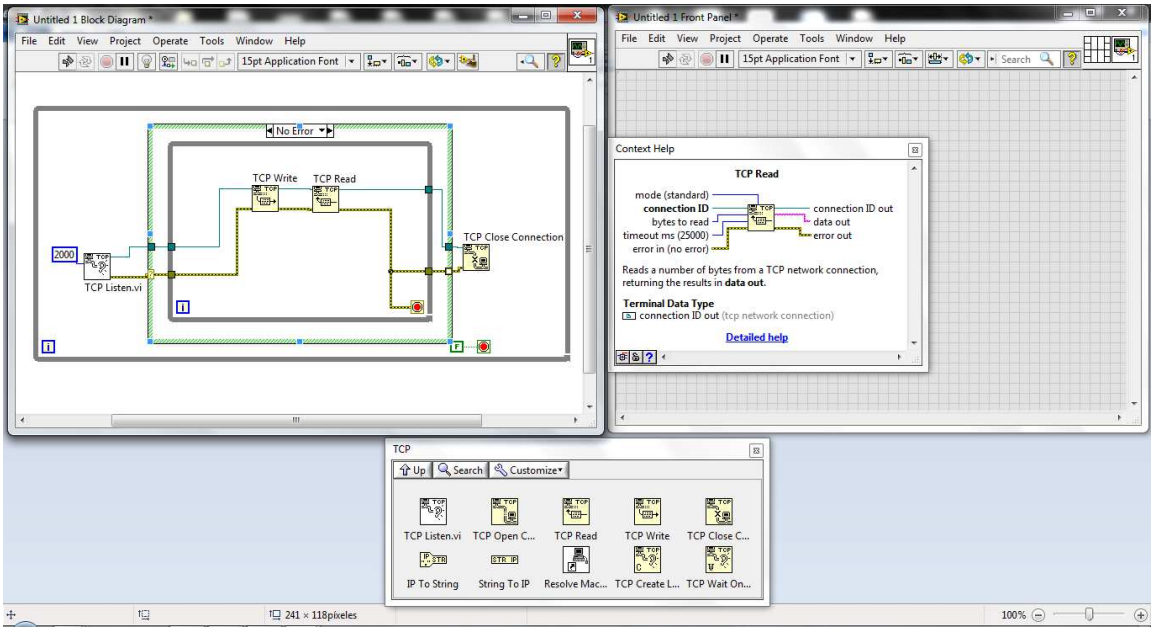

Figura 36. Bloque *TCP Read*

En el Front Panel nos ubicamos en Controles Modernos, librería de Arreglos, Matrices y Clusters y seleccionamos un "*Array*". El arreglo será booleano y será el dato que enviará al PLC.

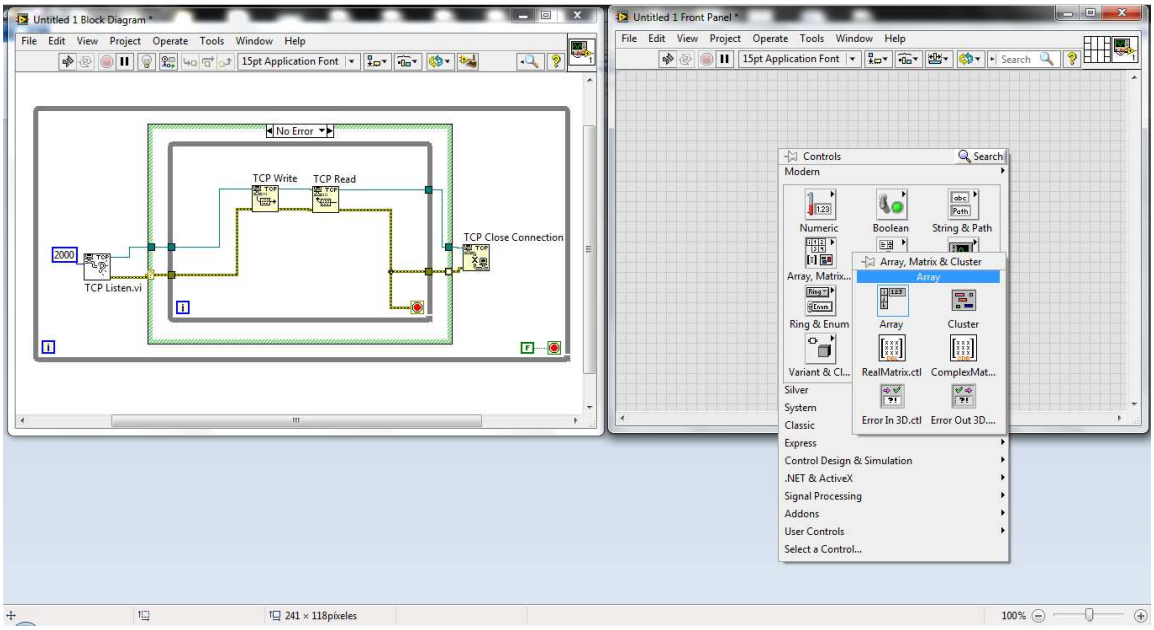

Figura 37. Insertar Array

Estando en Controles Modernos agregamos un *Switch* tipo "*Vertical Toggle*" de la librería "Boolean".

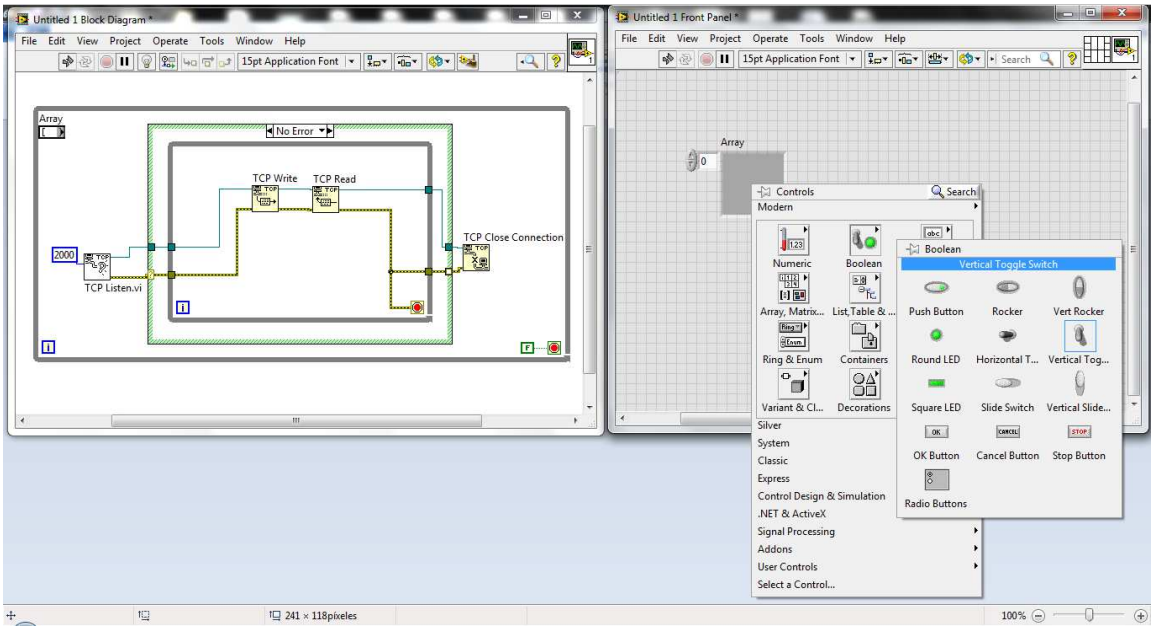

Figura 38. Insertar *Vertical Toggle*

Extendemos el arreglo hasta obtener el numero de *Switches* necesarios, en este caso utilizamos 16, porque se van a transmitir 2 Bytes hacia el PLC entonces, cada Switch representa un bit.

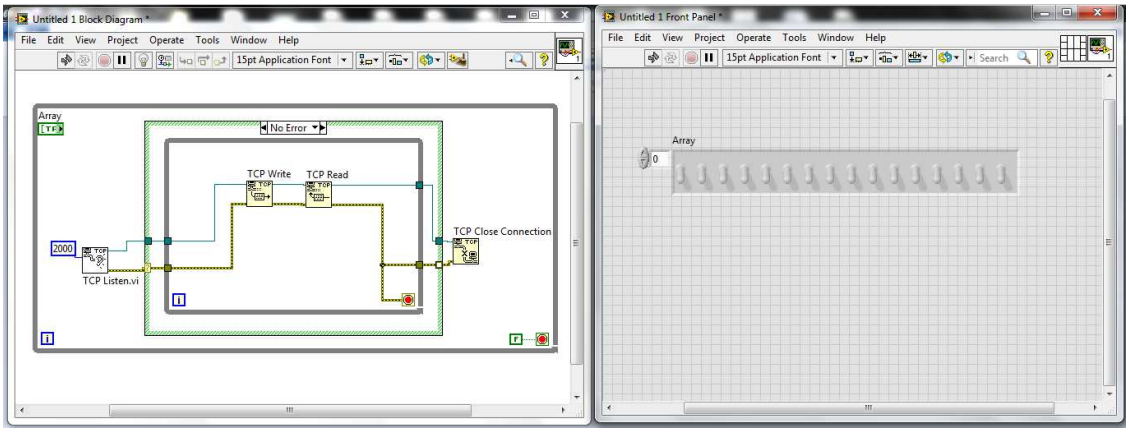

Figura 39. Obtener 16 *Switches*

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

En el Block Diagram ubicamos la librería "*Programming", "Boolean"* y agregamos un bloque llamado "*Array to Num*" que convierte un arreglo booleano en número entero mediante la interpretación del arreglo como una representación binaria del número.

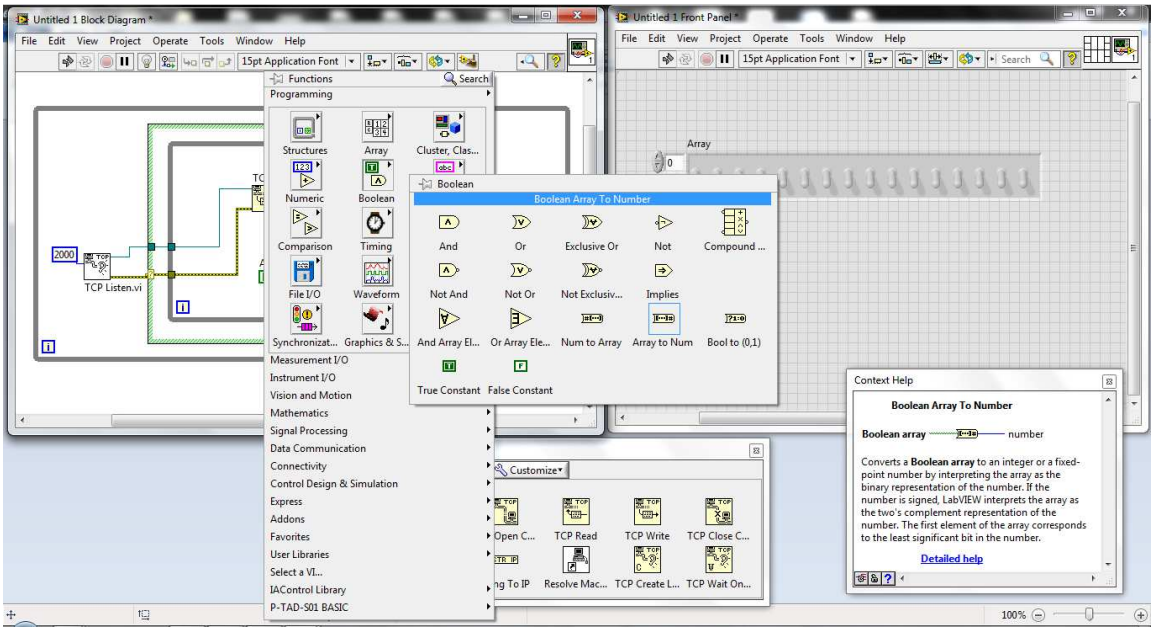

Figura 40. Agregar bloque *Array to Num*

El bloque *Array to Num* se conecta al arreglo booleano previamente hecho. Este bloque funciona de la siguiente manera: si se cablea un arreglo de dos valores booleanos para esta función y los dos valores son FALSE ([0 0]), la función devuelve un valor de 0. Si el primer valor booleano es TRUE ([1 0]), la función devuelve 1. Si el valor del segundo booleano es TRUE ([0 1]), la función devuelve 2. Si los dos valores booleanos son TRUE ([1 1]), la función devuelve 3.

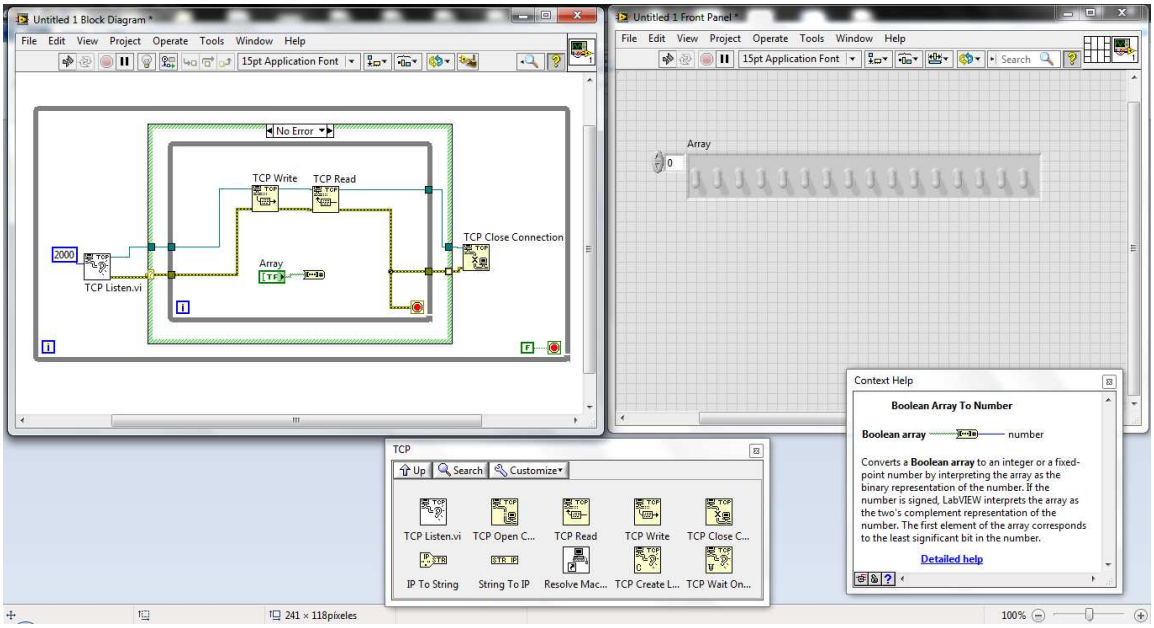

Figura 41. *Array to Num*

Vamos a la paleta de conversiones y seleccionamos “To Unsigned Word Integer Function” que convierte un número a un entero sin signo de 16-bit en el rango de 0 a 65.535.

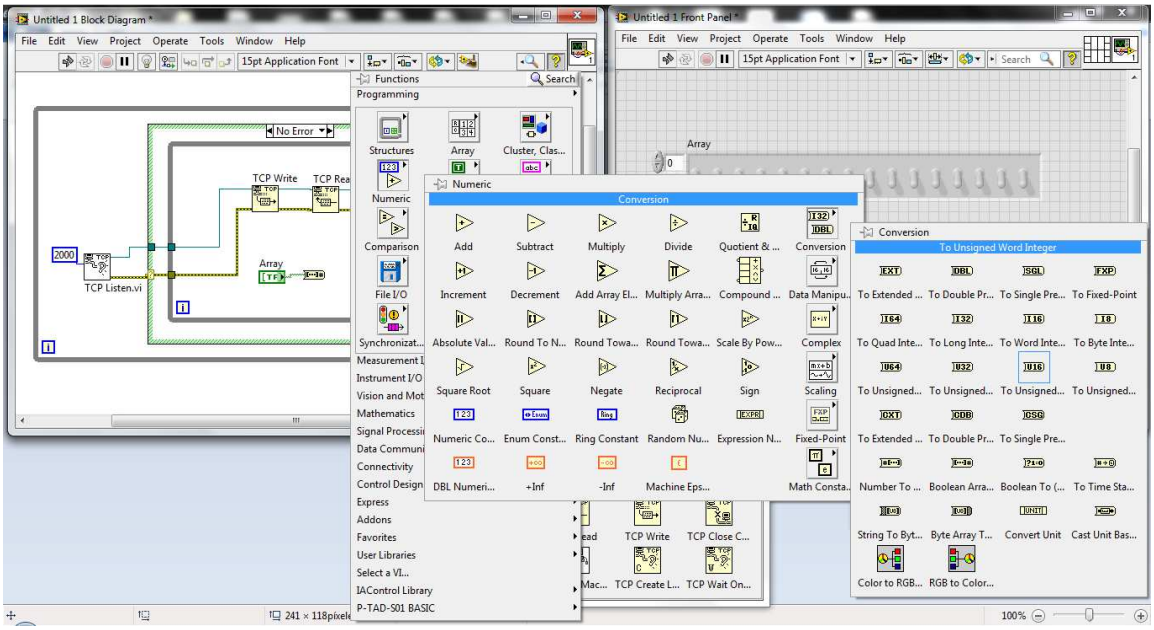

Figura 42.Bloque *To Unsigned Word Integer*

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

Enlazamos el bloque *To Unsigned Word Integer* al arreglo previamente convertido de formato booleano a formato numérico.

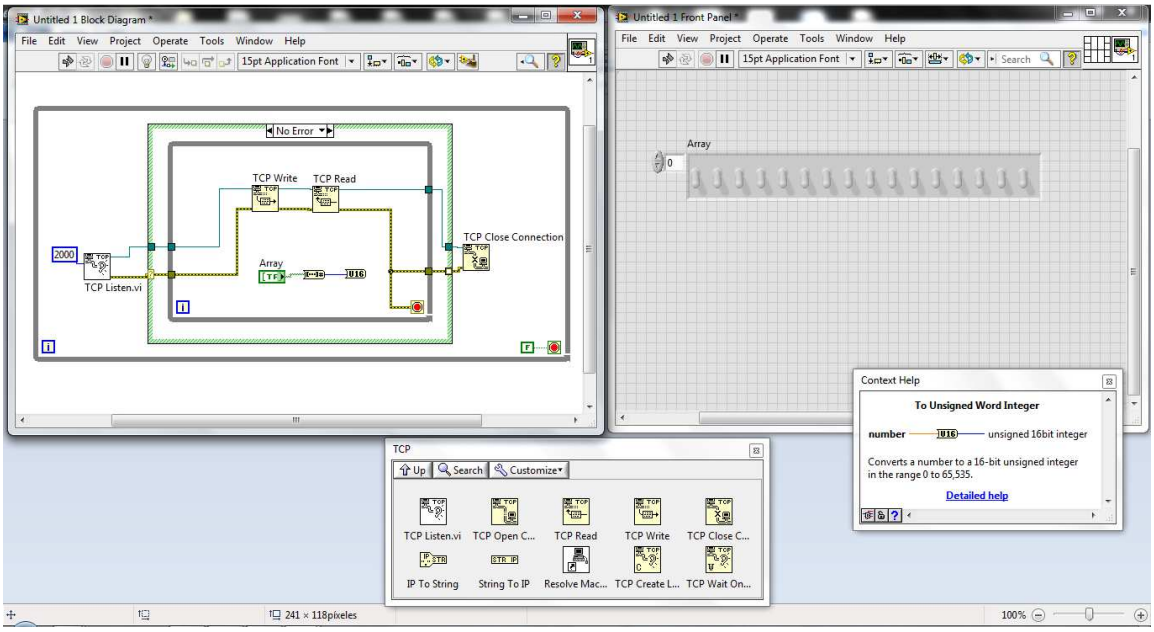

Figura 43. Conectar Bloque *To Unsigned Word Integer*

De la librería "*Numeric", "Data Manipulation" s*eleccionamos el bloque *Type Cast* que convierte entre tipos de datos, para nuestro caso de formato numérico a string, ya que el bloque *TCP Write* en su entrada *data in* solo recibe datos con formato string.

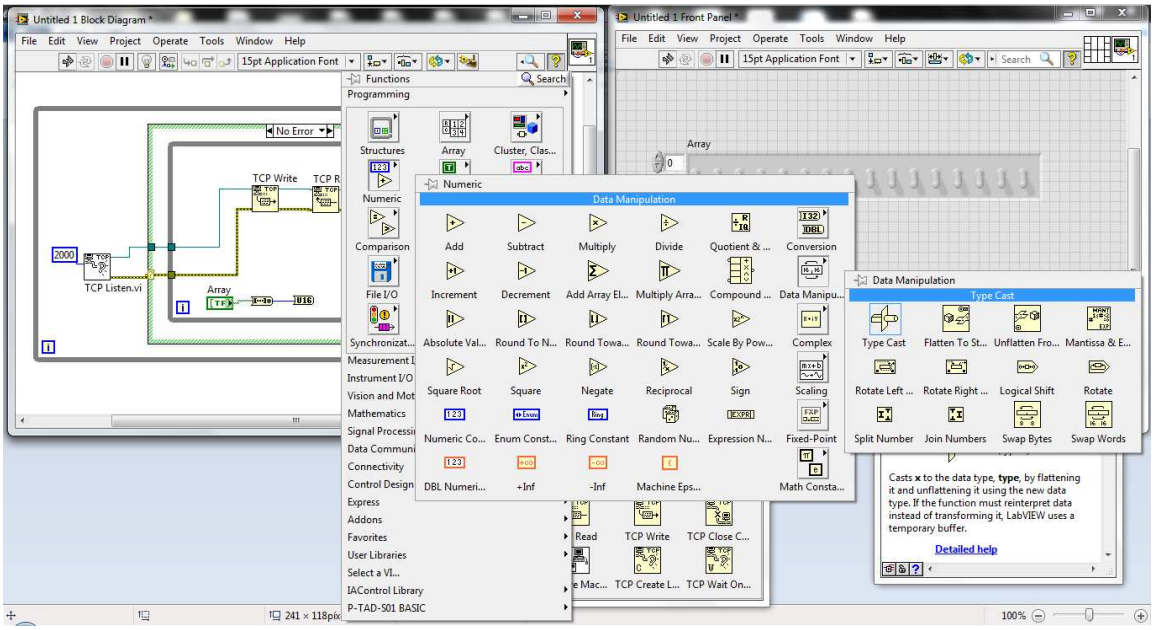

Figura 44. Librería *Data Manipulation*

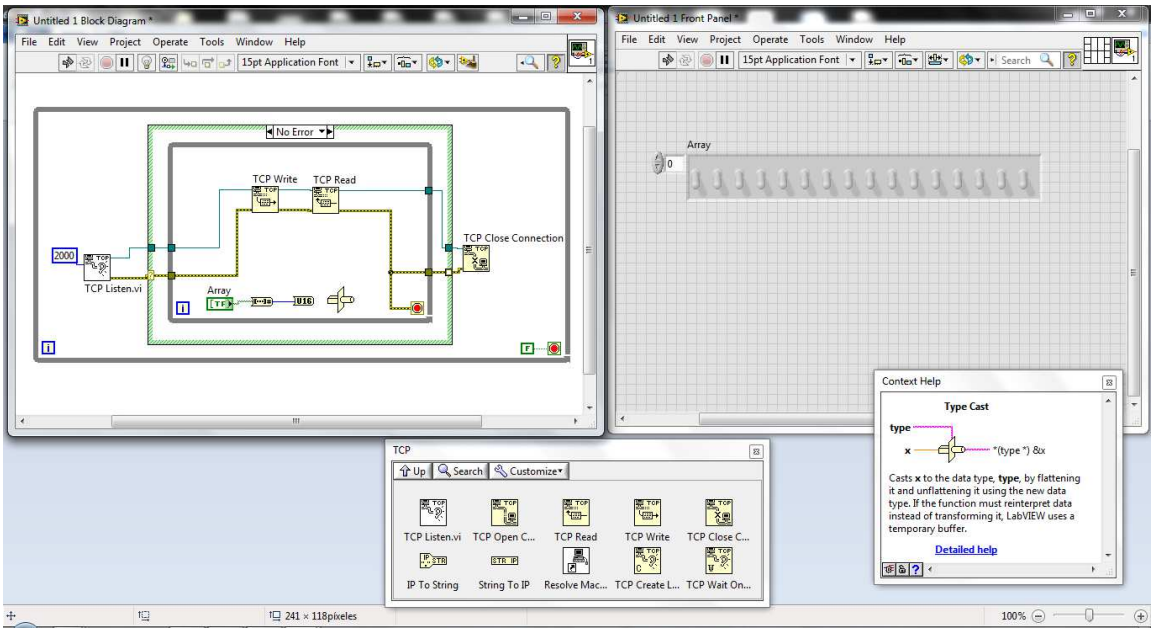

Figura 45. Insertar *Type Cast*

Vamos a convertir el dato numérico del arreglo a un S*tring* que será escrito en el bloque *TCP Write,* simplemente conectamos la salida del bloque *Type Cast* en la entrada "*Data in*" del bloque *TCP Write.*

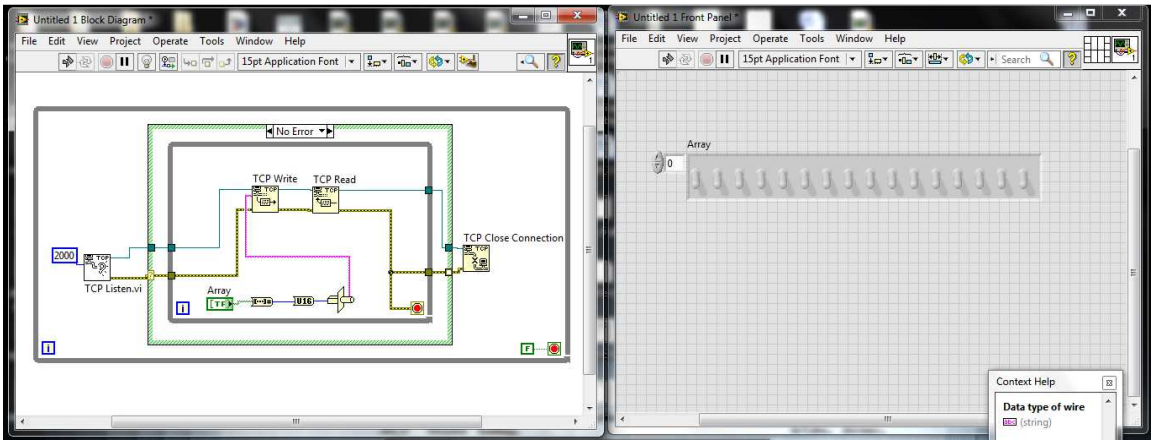

Figura 46. Conectar *Type Cast* a *Data in*

Tomamos lo que se lee en el bloque "*TCP Read*" y lo convertimos en tipo de dato Unsigned Word Integer.

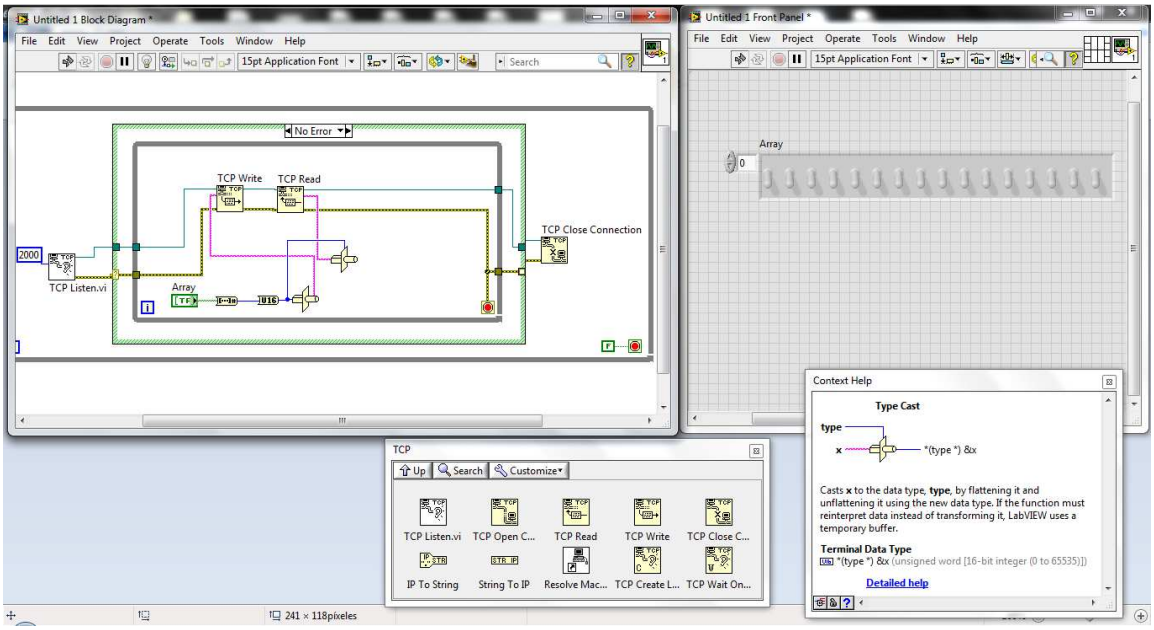

Figura 47. Convertir a Unsigned Word Integer.

En la paleta de booleanos tomamos el bloque "*Num to array*" (Función que convierte de un número a un arreglo booleano).

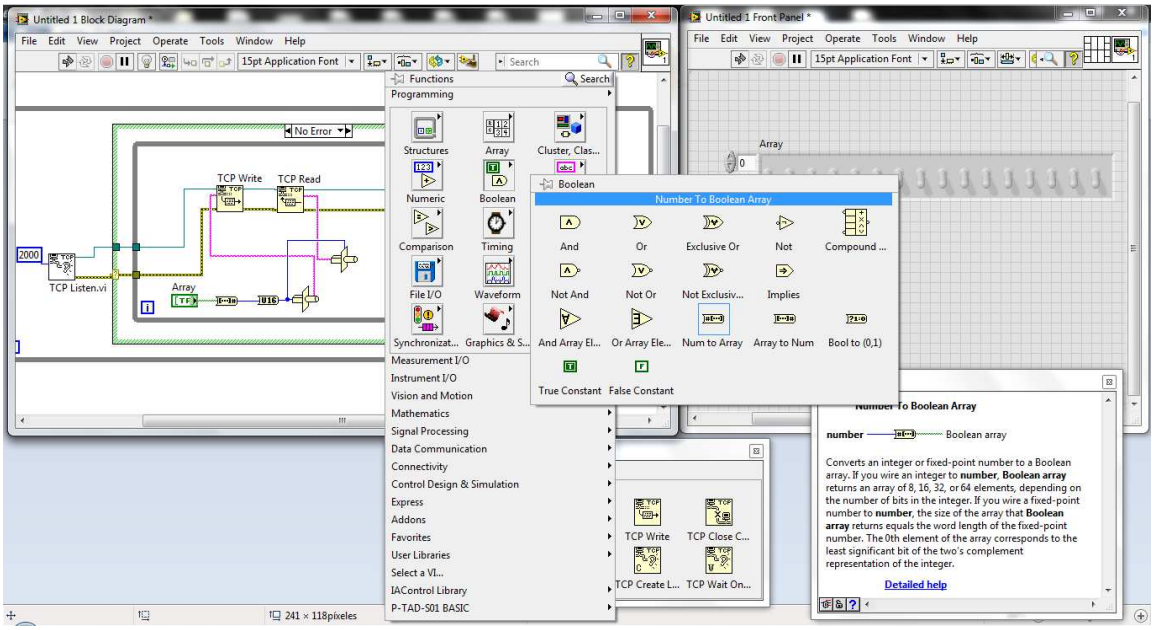

Figura 48. Bloque *Num To Array*

Lo conectamos al dato leído del bloque *TCP Read* con este bloque vamos a convertir un número entero en un arreglo booleano. Si se cablea un número entero, el arreglo booleano retorna un arreglo de 8, 16, 32, o 64 elementos, en función del número de bits en el entero.

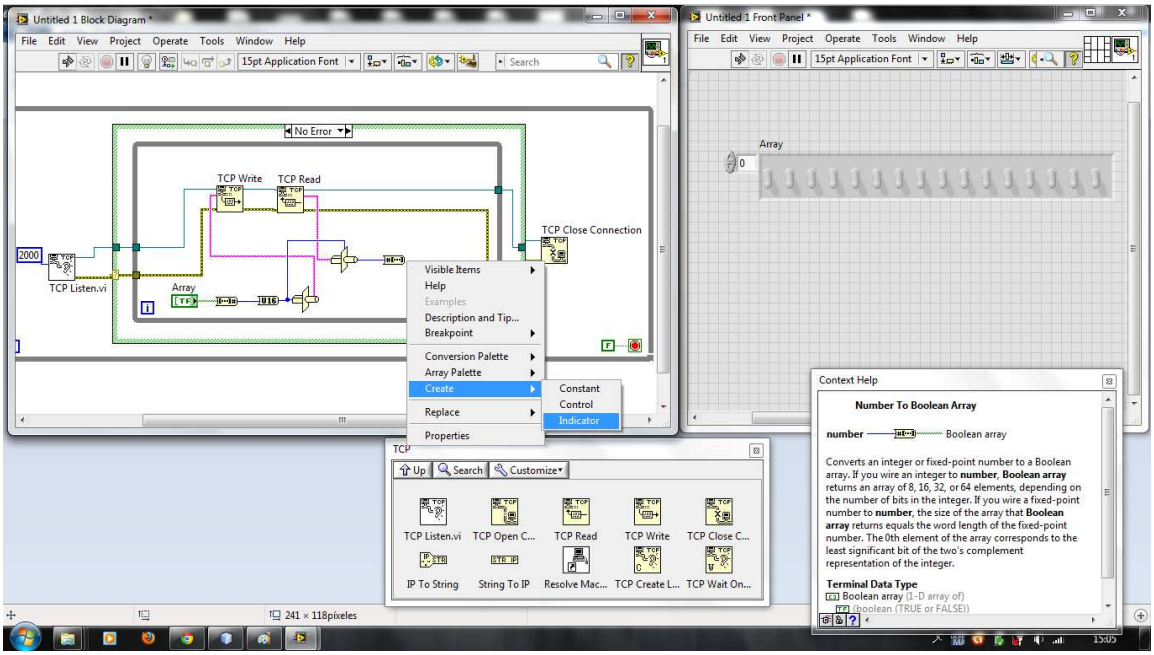

Figura 49. *Num to Array* conectado a *TCP Read*

De esta forma obtenemos un arreglo booleano de los 16 bits de datos que van a ser leídos.

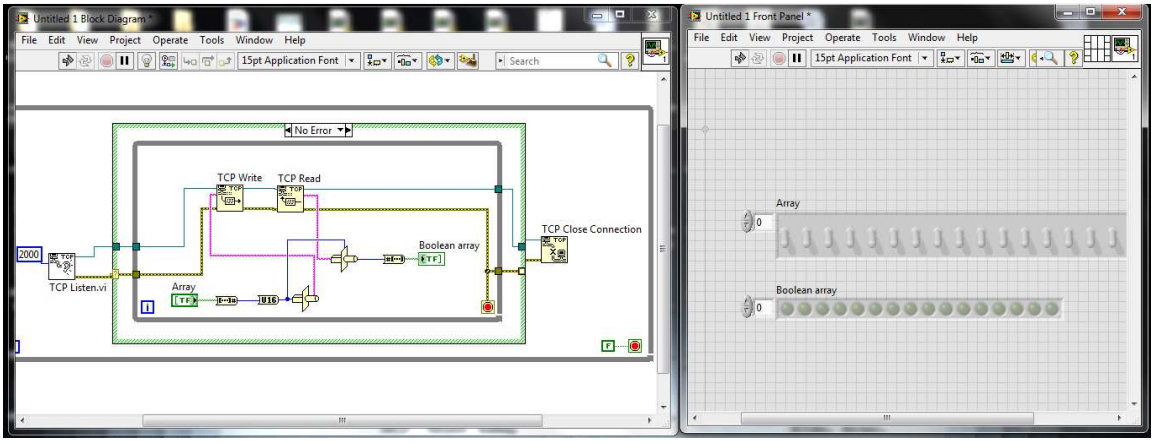

Figura 50. *TCP Read* Arreglo Booleano

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

En la librería Programming, Dialog&User Interface buscamos el bloque "*Clear Errors*" *que b*orra los diálogos de error que se puedan producir en la comunicación, para que no se visualicen posteriormente.

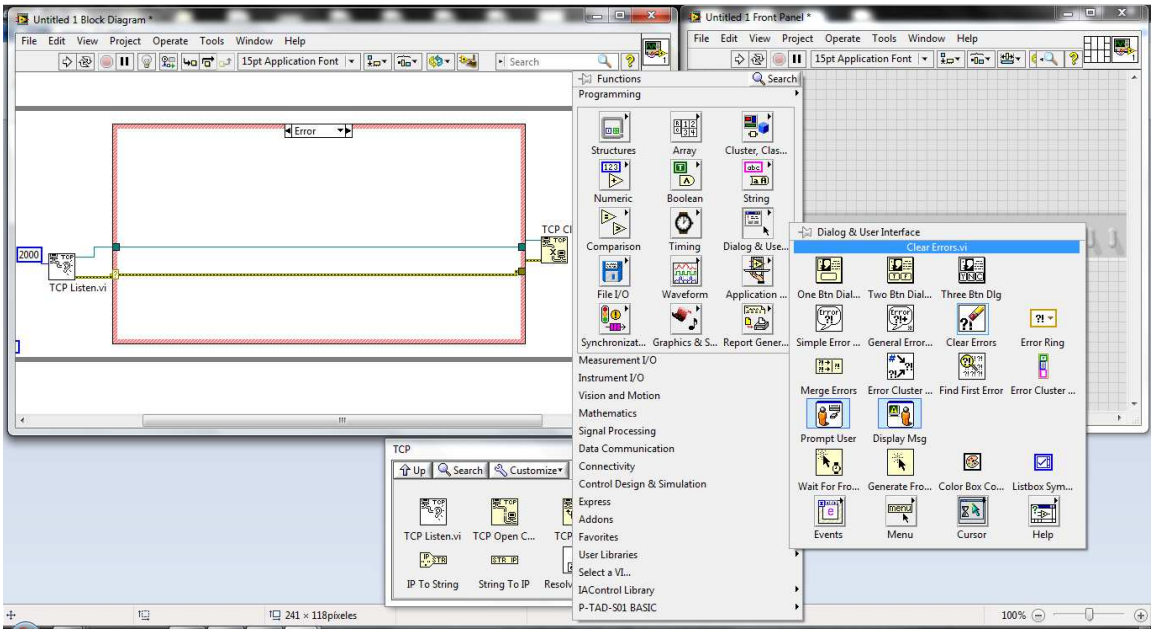

Figura 51. Bloque *Clear Errors*

Se conecta luego del bloque que cierra la conexión TCP, para que en el futuro cuando se vuelva a correr el programa no se carguen diálogos de errores de comunicación anteriores.

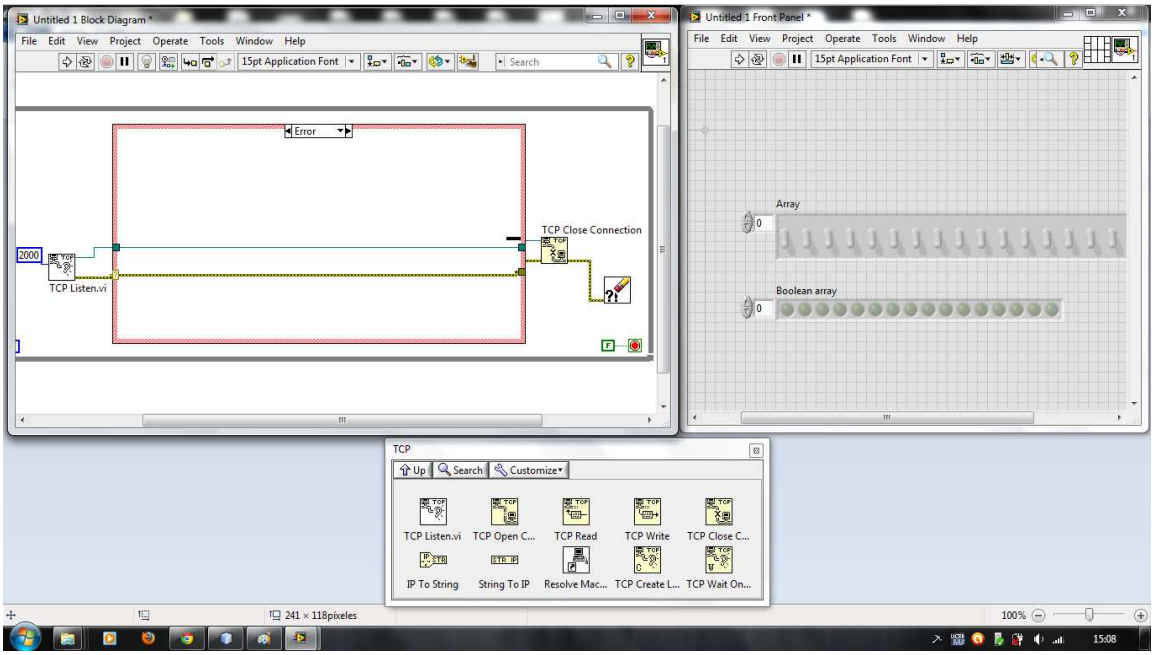

Figura 52. Conectar Bloque *Clear Errors*

Se agrega a la entrada *timeout ms* del bloque *TCP Listen* un tiempo de 1 ms, es el tiempo en el que se espera que haya conexión, sino se muestra un error y cierra la conexión como se dijo anteriormente, luego de lo cual se trata de establecer la conexión nuevamente.

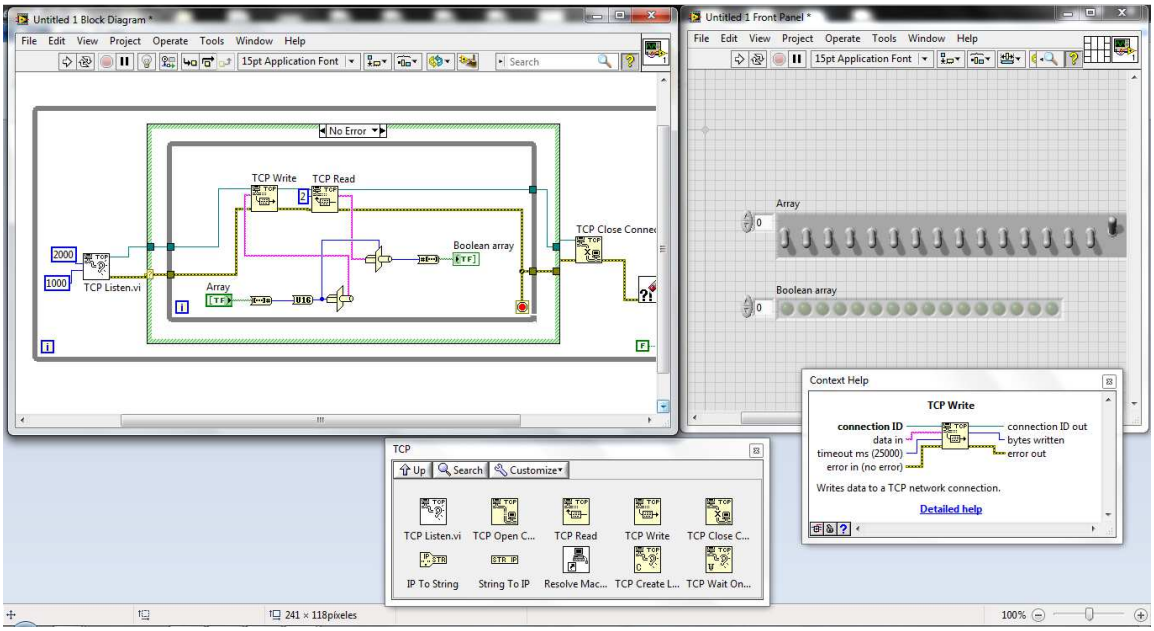

Figura 53. *Timeout* de 1 milisegundo

Especificamos en el bloque *TCP Read* cuántos bytes se van a leer, en este caso se leerán 2 bytes.

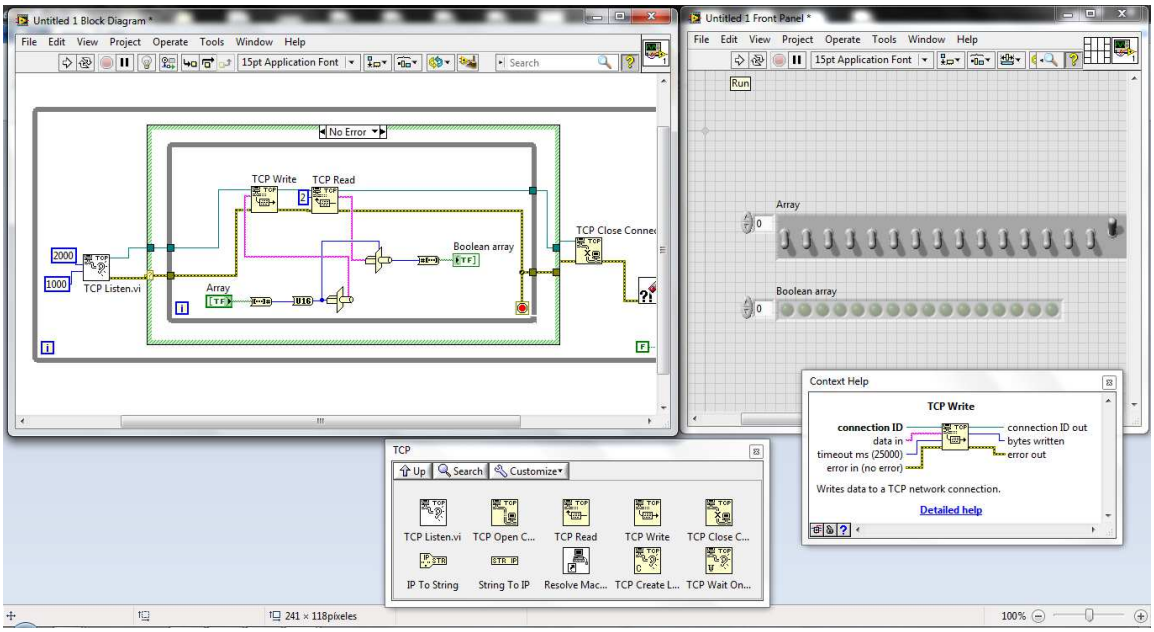

Figura 54. Cantidad Bytes a leer

Procedemos a asignar la dirección IP del PC y la máscara de subred, la misma que se puso en el Interlocutor *Sin Especificar* en los respectivos bloques de TSEND_C y TRCV_C del PLC.

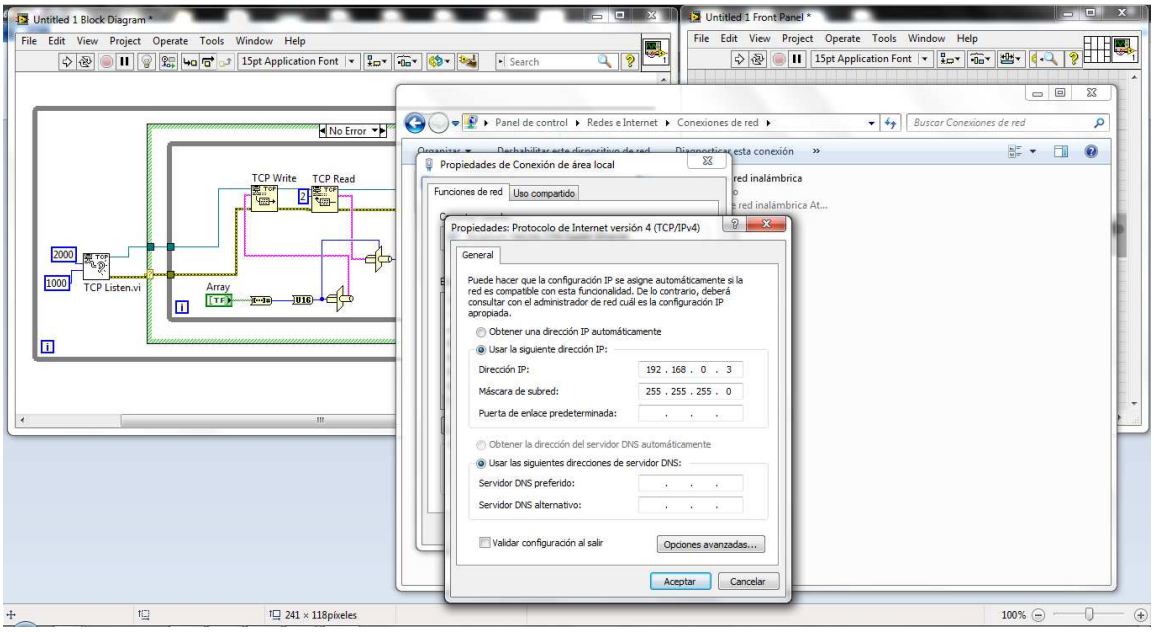

Figura 55. Asignar Dirección IP y Máscara de Subred.

Ahora se puede proceder a correr el programa. En este caso ponemos en alto los bits 15,14 y 13.

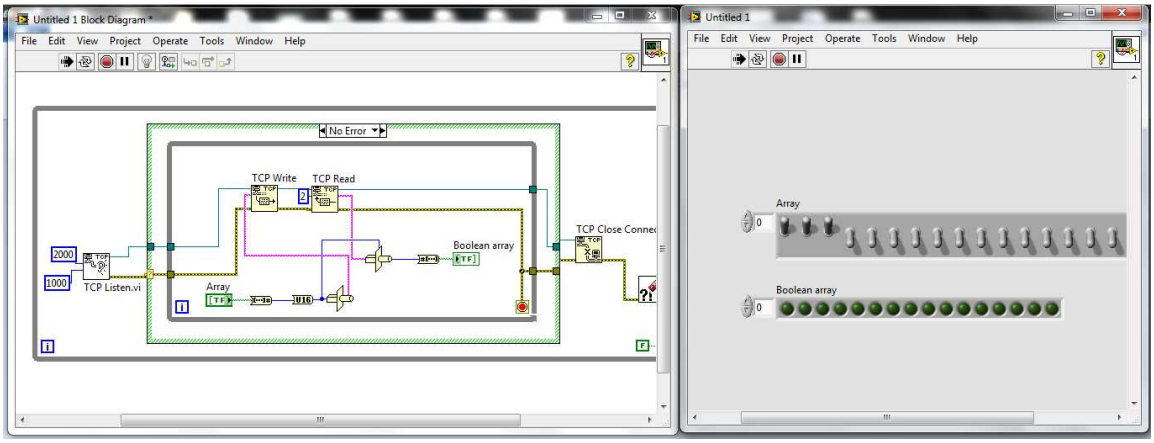

Figura 56. Correr el programa

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

En MB2 y MB3 se visualiza en dato recibido del computador pero como se puede observar se visualiza invertido esto es debido a que el bit menos significativo para el PLC es el bit 15 del arreglo booleano de Labview.

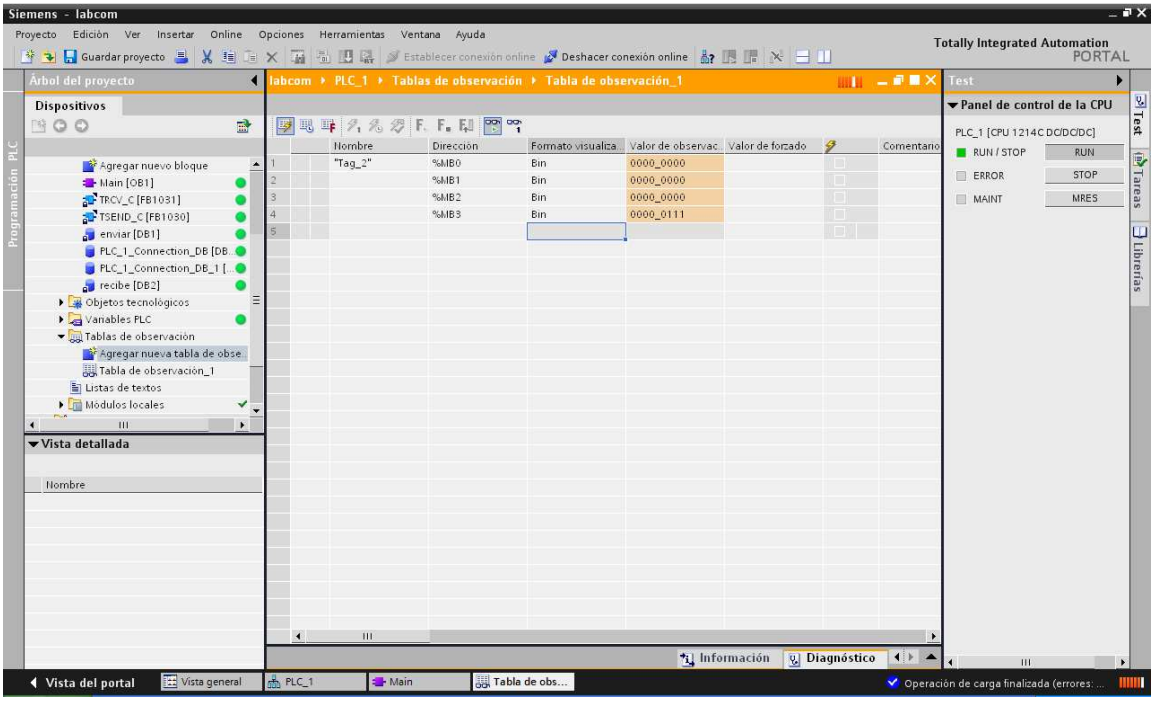

Figura 57. Visualización Bytes enviados al PLC (MB2 y MB3)

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

Luego de esto observamos un Byte enviado desde el PLC al computador por medio del arreglo booleano destinado para dicha finalidad agregado en la Figura 50.

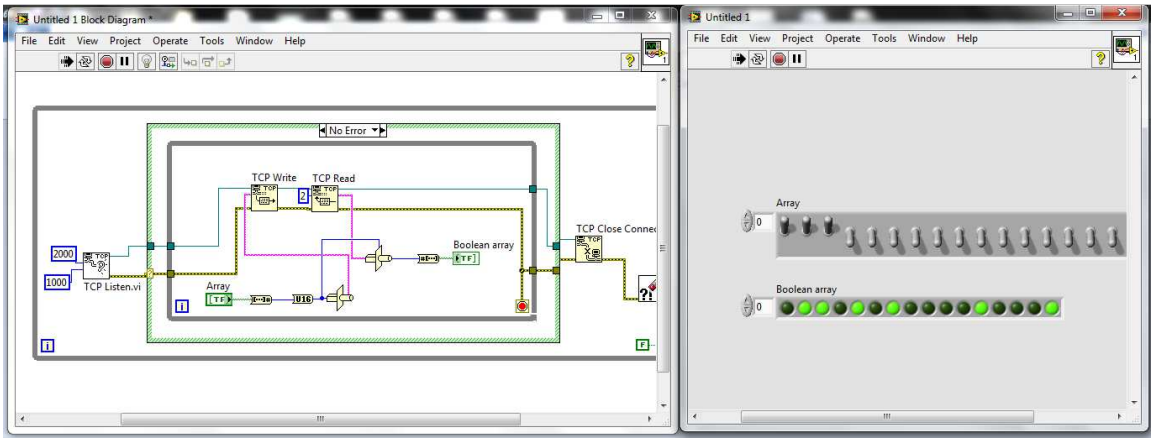

Figura 58. Lectura Bytes recibidos del PLC

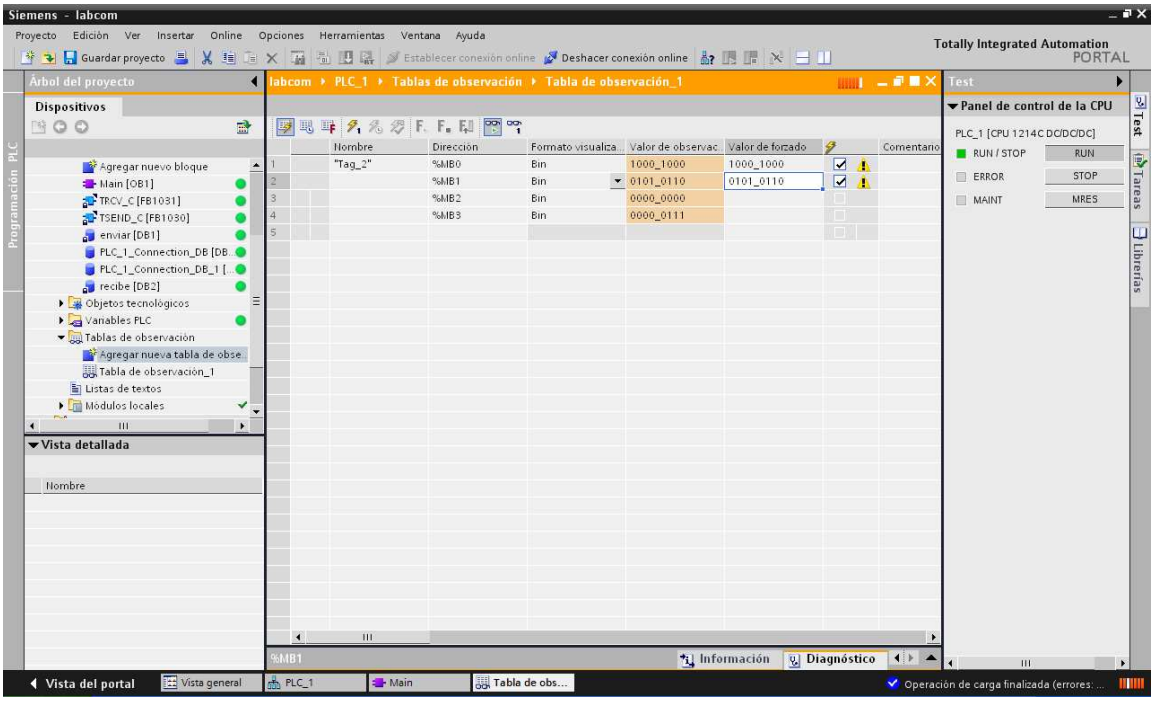

Figura 59. Bytes enviados desde el PLC (MB0 y MB1)

Como se explicó anteriormente los datos enviados y recibidos se visualizan invertidos, entonces, para evitar confusiones procederemos por software a invertir los datos de forma que se visualicen en el mismo orden en el computador y en el PLC.

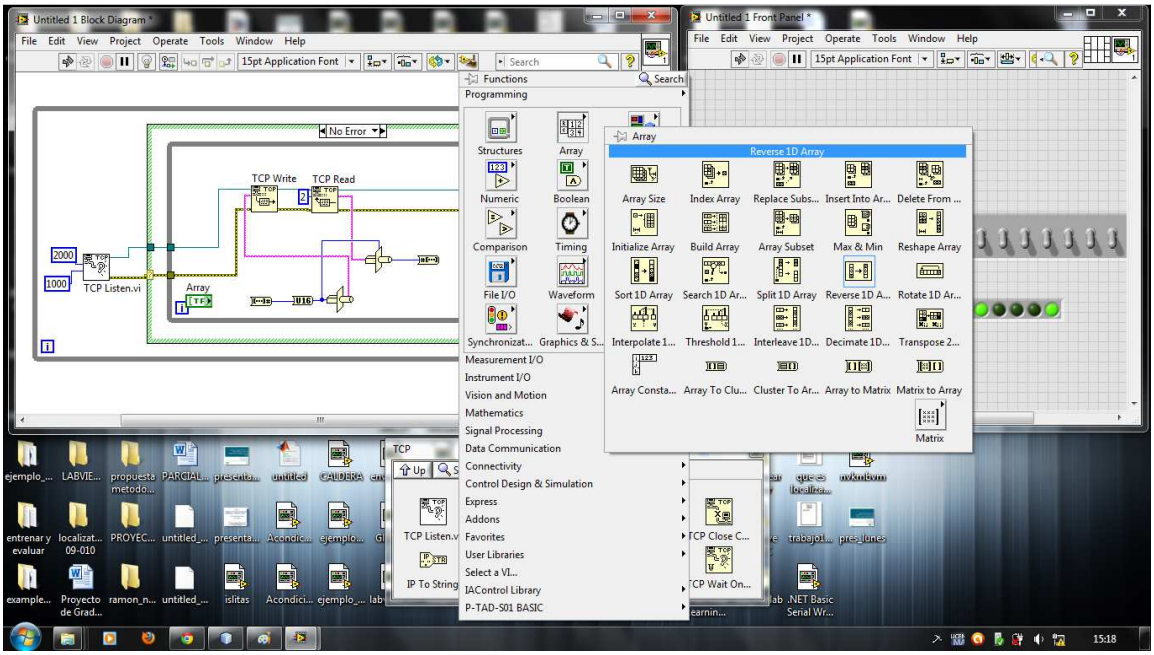

Figura 60. Insertar bloque *Reverse 1D Array*

Este bloque se puede encontrar en Programming, en la librería Array. Este bloque invierte los elementos de un arreglo.

Insertamos el bloque *Reverse 1D Array* antes del arreglo booleano de visualización.

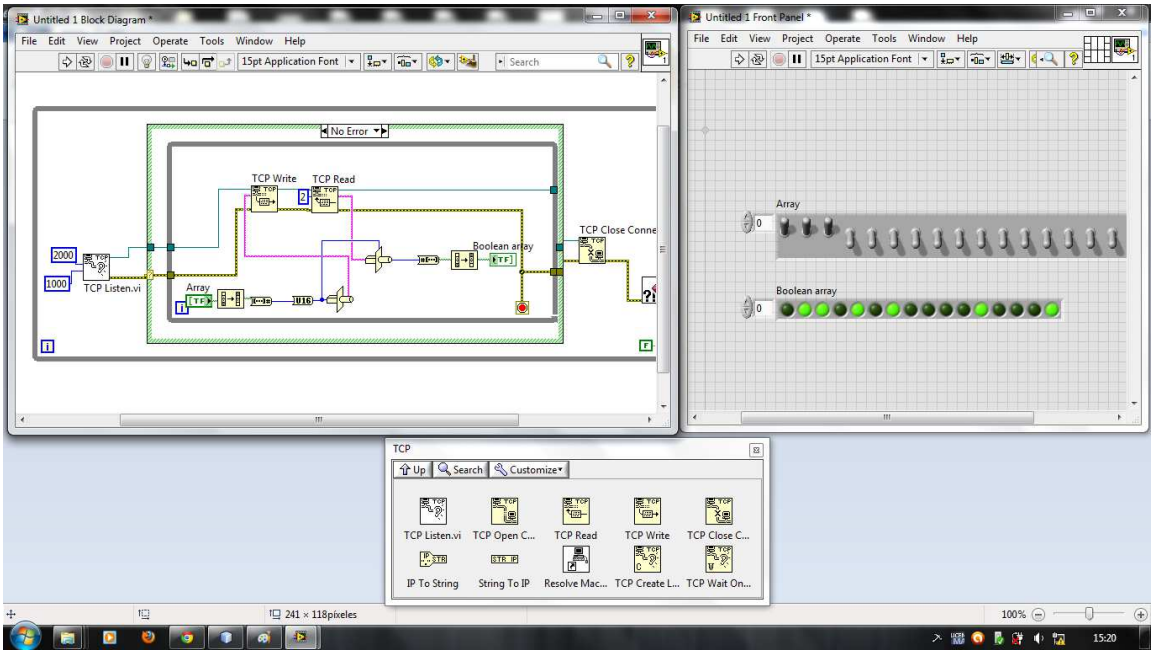

Figura 61. Conectar *Reverse 1D Array*

Ahora se puede visualizar el dato tal cual como se envió desde el PLC.

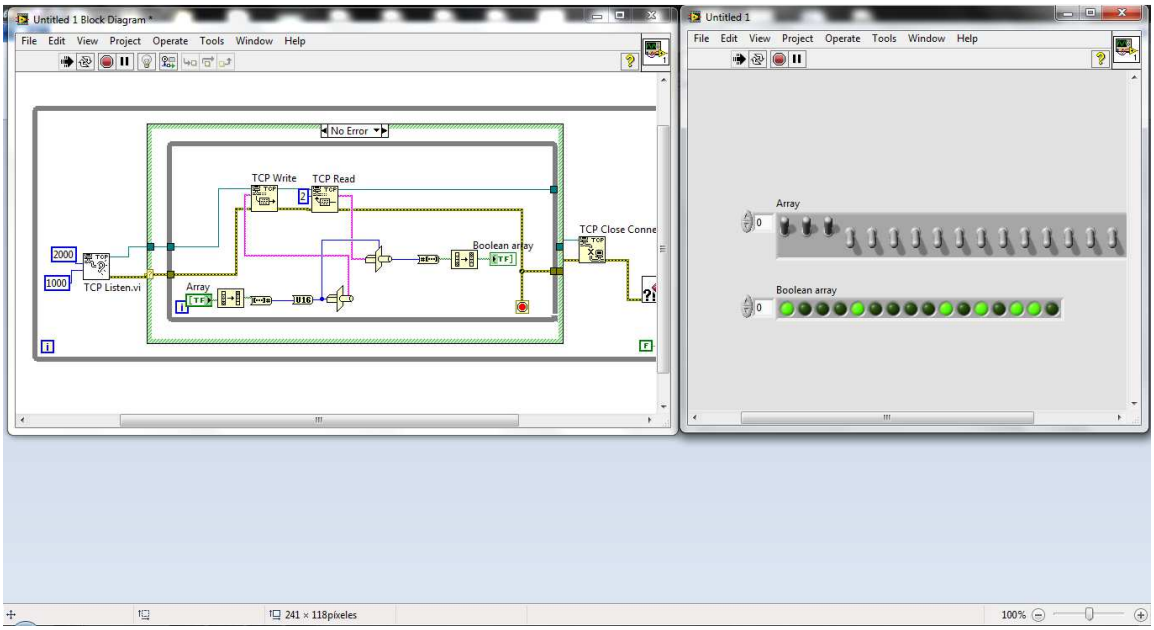

Figura 62. Visualizar dato correctamente

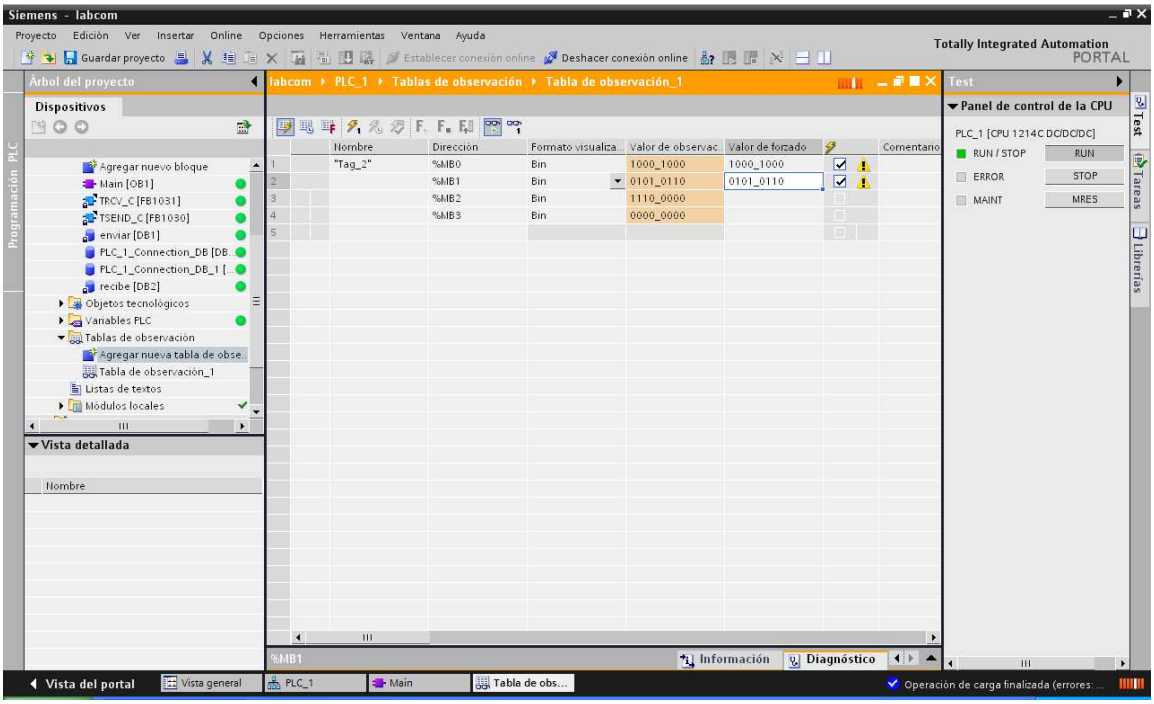

Figura 63. Confirmar dato enviado desde el PLC

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

7. DISEÑO DE HMIs CON WINCC FLEXIBLE

Un sistema HMI representa la interfaz entre el usuario y el proceso. En el proceso las operaciones son controladas principalmente por el PLC, pero el usuario puede utilizar el panel de operador o pantalla HMI para monitorear el proceso o intervenir en el proceso en ejecución.

Las funciones de la HMI son manejar y visualizar el funcionamiento de las máquinas e instalaciones, específicamente mostrar los procesos, operar los procesos, dar avisos cuando algo específico pase en el proceso, modificar las distintas variables del proceso, entre otros. Existen objetos predefinidos disponibles para crear estas pantallas, se puede utilizar estos objetos para simular las máquinas, mostrar los procesos y definir valores a las variables del proceso.

Para iniciar la creación de una interfaz de usuario con WinCC flexible es necesario que se tenga un proyecto ya creado con un PLC añadido, conectado a una subred Ethernet. Si no ha realizado esto configure esta subred como se explicó en la sección COMUNICACIÓN ETHERNET ENTRE 2 PLCS. La vista de redes debería estar como en la Figura 64.

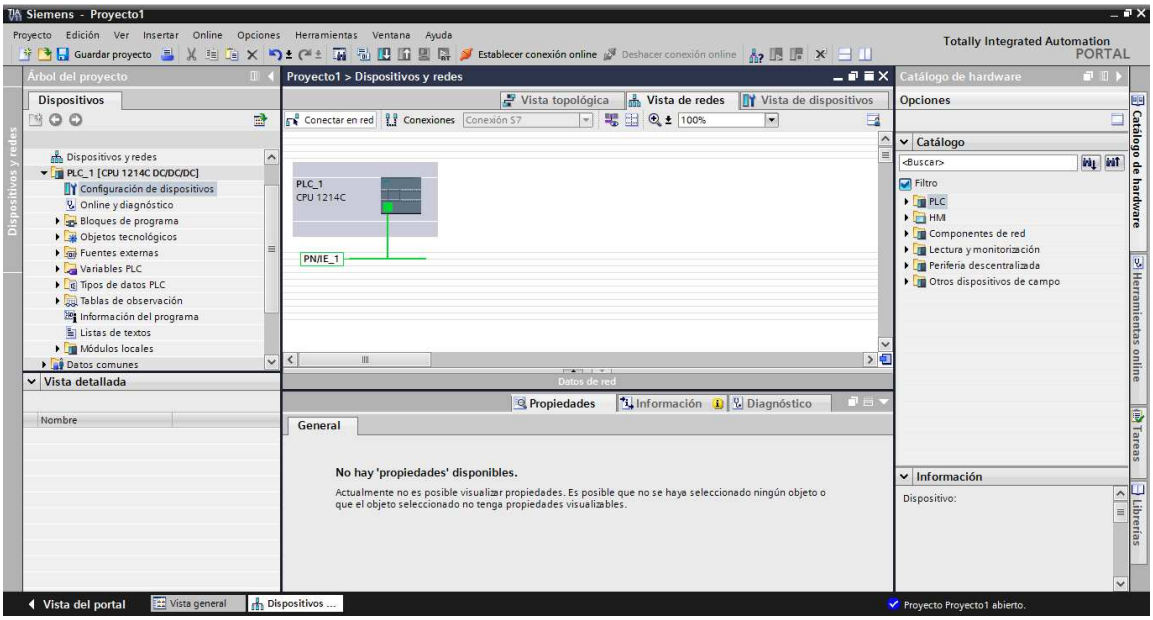

Figura 64. Vista de redes con PLC y subred configurada

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

Una vez se cuente con estas condiciones se hace clic en "*Agregar dispositivo*" en el árbol del proyecto.

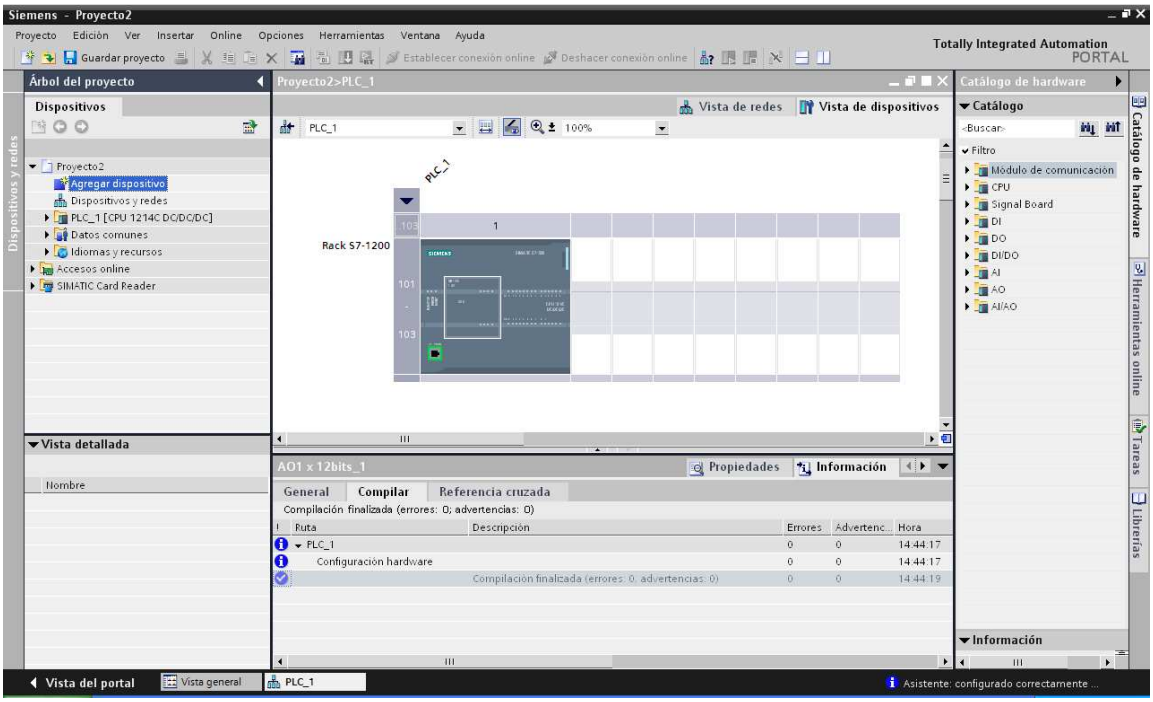

Figura 65. Agregar nuevo dispositivo

Seleccione la opción "*SIMATIC HMI",* escoger la pantalla que se tiene para la aplicación, en este caso es una pantalla de 6" entonces desplegamos la librería "*6" Display*" y se elige la referencia correspondiente al dispositivo, en nuestra aplicación es una KTP600 PN. Por último asígnele un nombre al dispositivo, deje seleccionada la casilla de verificación "*Iniciar el asistente de dispositivos*" que se encuentra en la esquina inferior izquierda y de clic en aceptar.

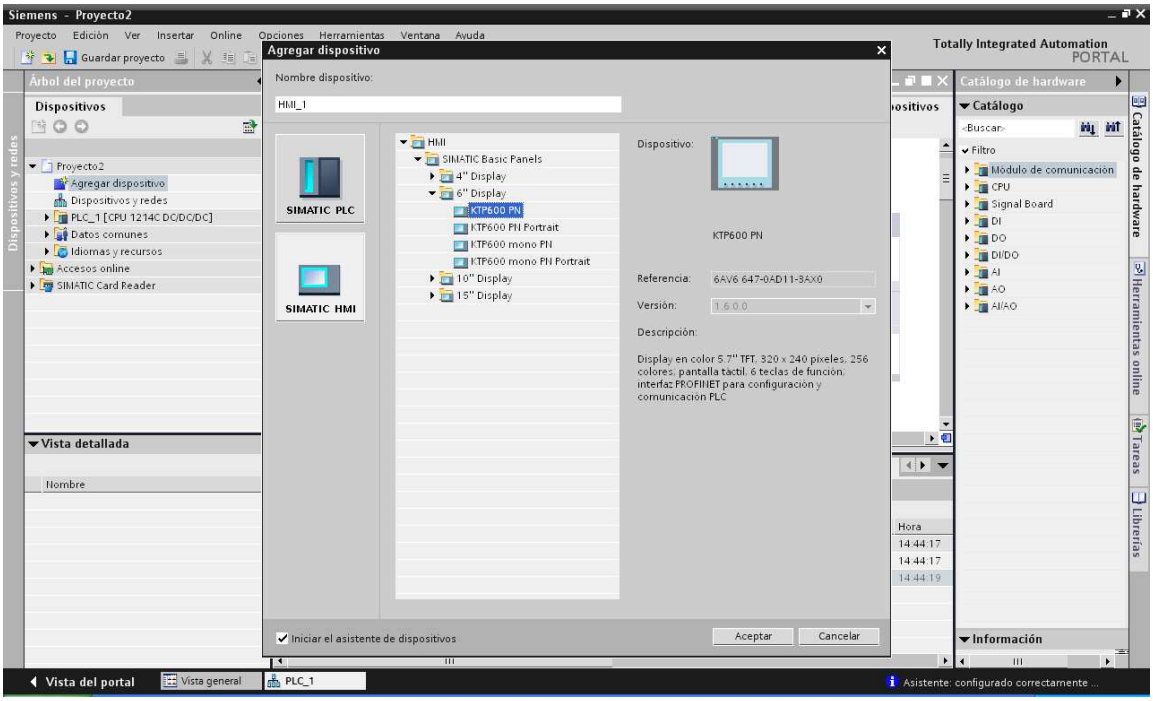

Figura 66. Seleccionar dispositivo HMI

Si no se tiene seleccionada esta casilla "*Iniciar el asistente de dispositivos*" entones la adición de la pantalla al programa será más complicada porque se debe realizar manualmente. Por lo tanto se sugiere utilizar el asistente. Asegúrese de tener activada la casilla para continuar.

El asistente de dispositivos se abre después de haber creado la nueva HMI, inicia con el cuadro de "Conexiones de PLC" de diálogo, aquí se define la configuración de la conexión del PLC y la HMI, esta conexión también se puede configurar en Dispositivos y redes. Si configura la conexión en este cuadro de diálogo, se creará automáticamente.

Despliegue la pestaña "*Examinar*" y seleccione el PLC con el cual se hará la conexión, en este caso PLC_1.

Si no se realiza este procedimiento luego se debería crear la conexión manualmente, no se recomienda hacer ese procedimiento por ser más largo. El asistente es la mejor opción para configurar la pantalla HMI.

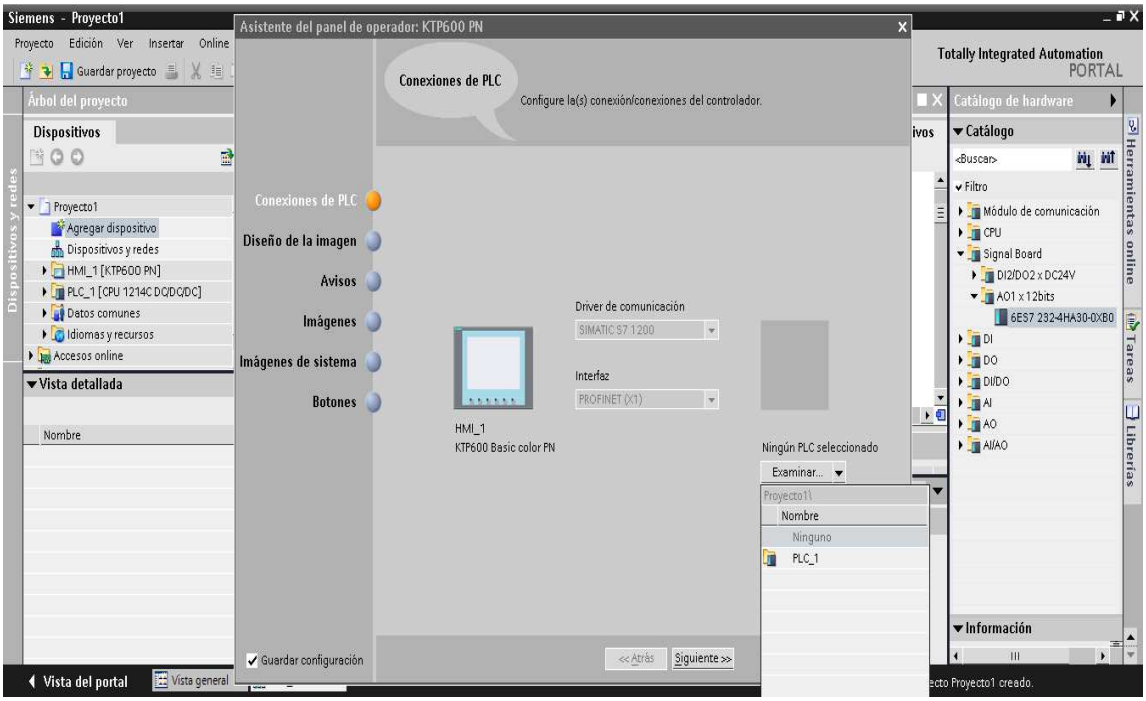

Figura 67. Conexión con el PLC

Después de configurar la conexión se observará la HMI conectada con el PLC, como se observa en la Figura 68.

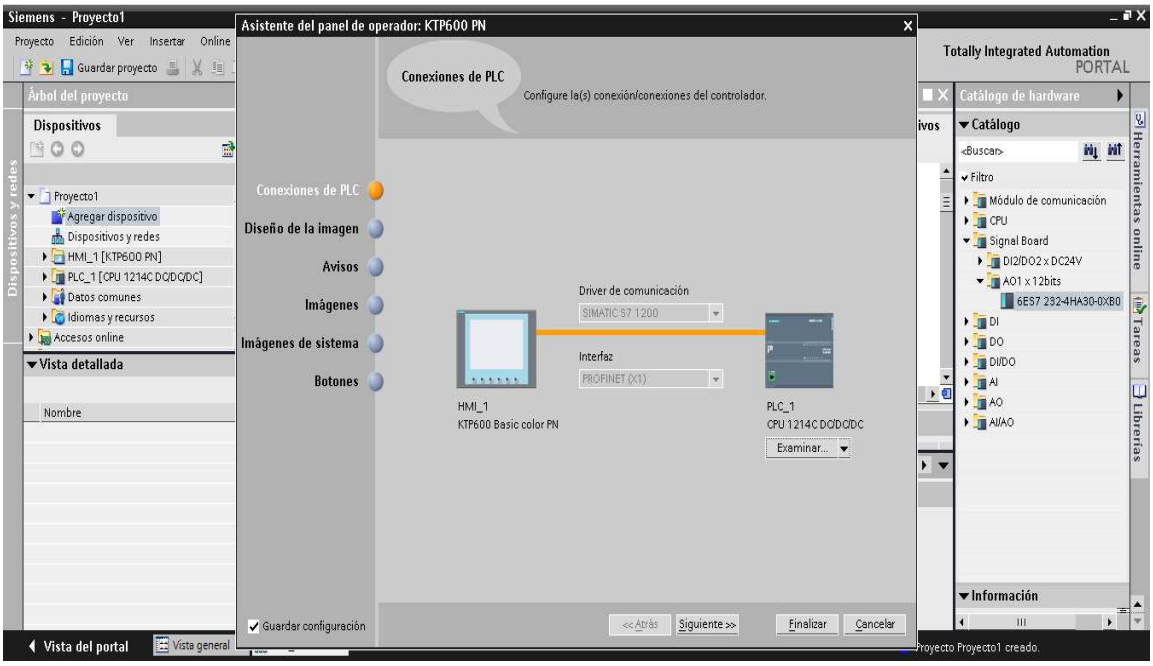

Figura 68. Conexión HMI y PLC

Haga clic en “Siguiente” para configurar paso a paso la pantalla. El siguiente paso es la presentación de la imagen. Se puede elegir el color de fondo y un encabezado con logotipo, fecha y hora. Se sugiere elegir el fondo blanco y deshabilitar el encabezado.

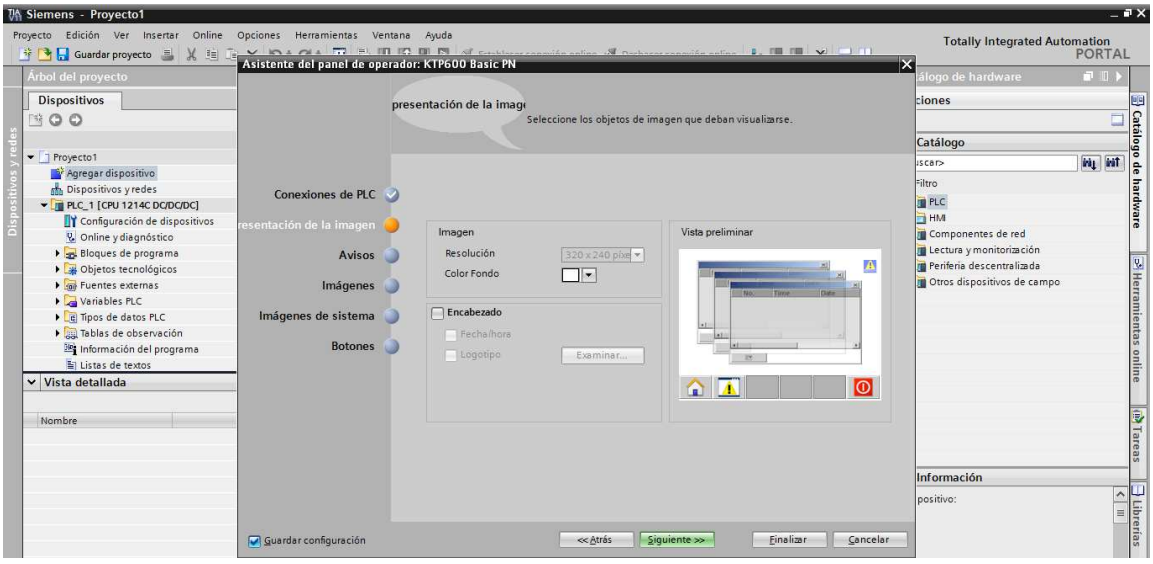

Figura 69. Presentación de la imagen

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

Haga clic en “Siguiente” y se pasará a la configuración de avisos. Se sugiere deshabilitar las tres opciones para tener más limpia la imagen.

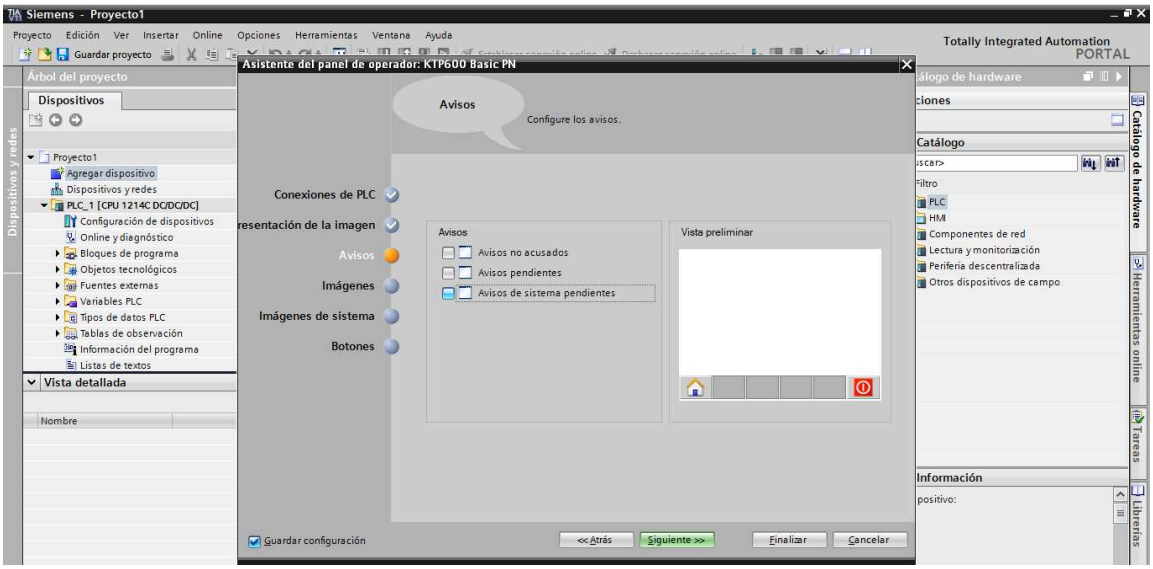

Figura 70. Configuración de avisos

El siguiente paso es la navegación de imágenes. Si ya se tiene definido cuántas imágenes tendrá el proyecto, entonces se pueden añadir desde esta opción. Si no se tiene definido, entonces se puede dejar únicamente la imagen raíz (principal) y luego se añadirán manualmente.

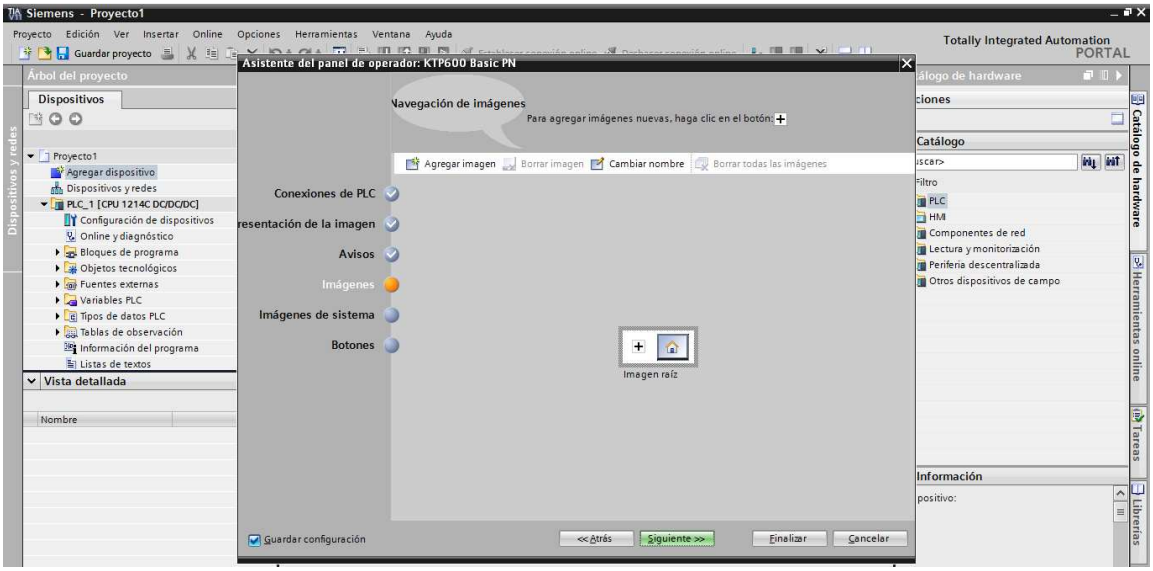

Figura 71. Navegación de imágenes

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

El siguiente paso es la elección de imágenes de sistema. Si no es un usuario avanzado no las necesitará, por lo tanto no cambie ninguna configuración y haga clic en “Siguiente”.

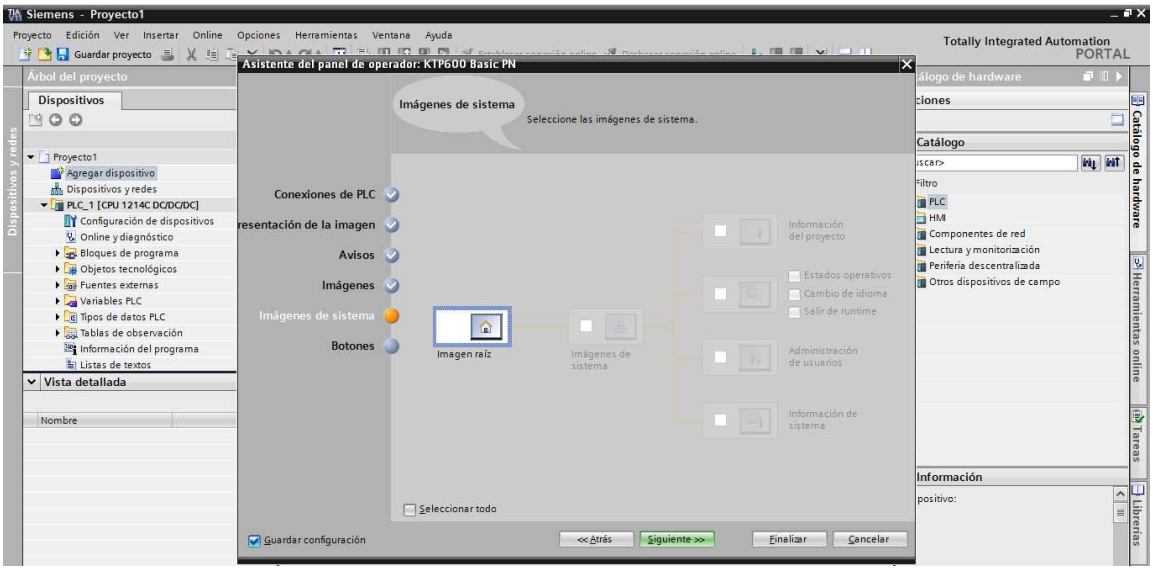

Figura 72. Imágenes de sistema

La última opción es la configuración de botones de sistema, los cuales permiten apagar la pantalla, o salir al panel de control, o regresar a la imagen principal. Éstos se pueden crear luego manualmente, pero si quiere seleccione algunos de ellos y arrástrelos a su posición deseada. Haga clic en “Finalizar” para terminar de configurar la pantalla y añadirla al proyecto según las elecciones realizadas.

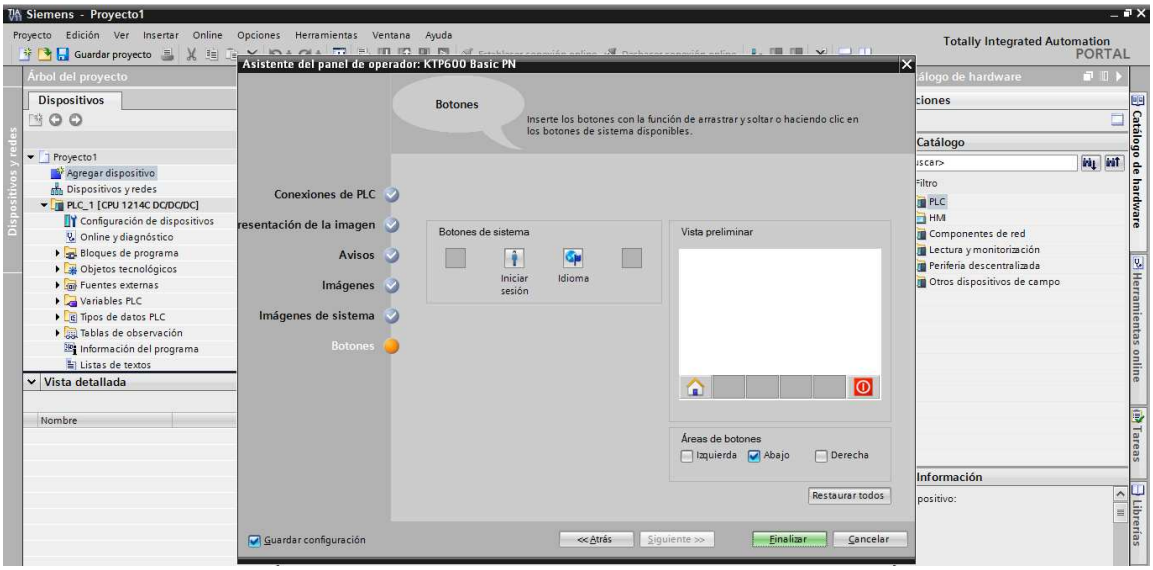

Figura 73. Añadir botones a las imágenes

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

Se creará entonces una pantalla según la referencia y preferencias elegidas.

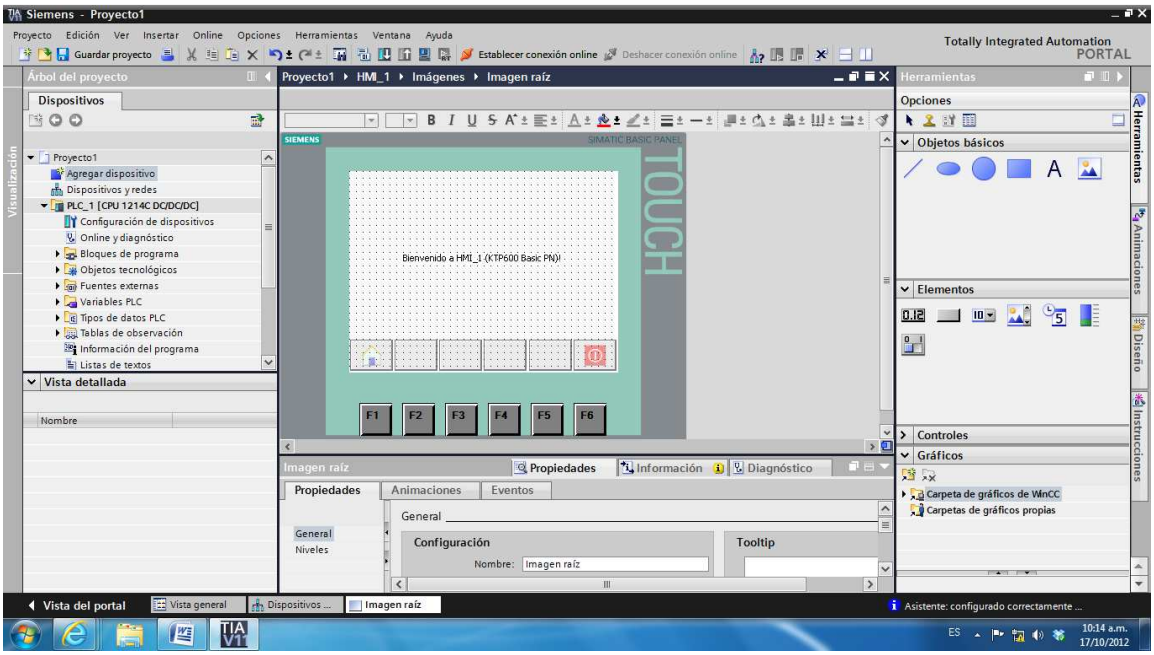

Figura 74. Imagen Básica

Note que aparece el texto “Bienvenido a HMI_1 (KTP600 Basic PN)!”, este texto se puede eliminar haciendo clic y luego presionando la tecla “Supr” del teclado, o bien, se puede modificar haciendo clic derecho y luego seleccionar “Propiedades”, para cambiar su formato, ubicación, tamaño y texto.

Para empezar la edición de una imagen desde cero, agreguemos una nueva imagen en la pestaña imágenes, clic en "*Agregar imagen*".

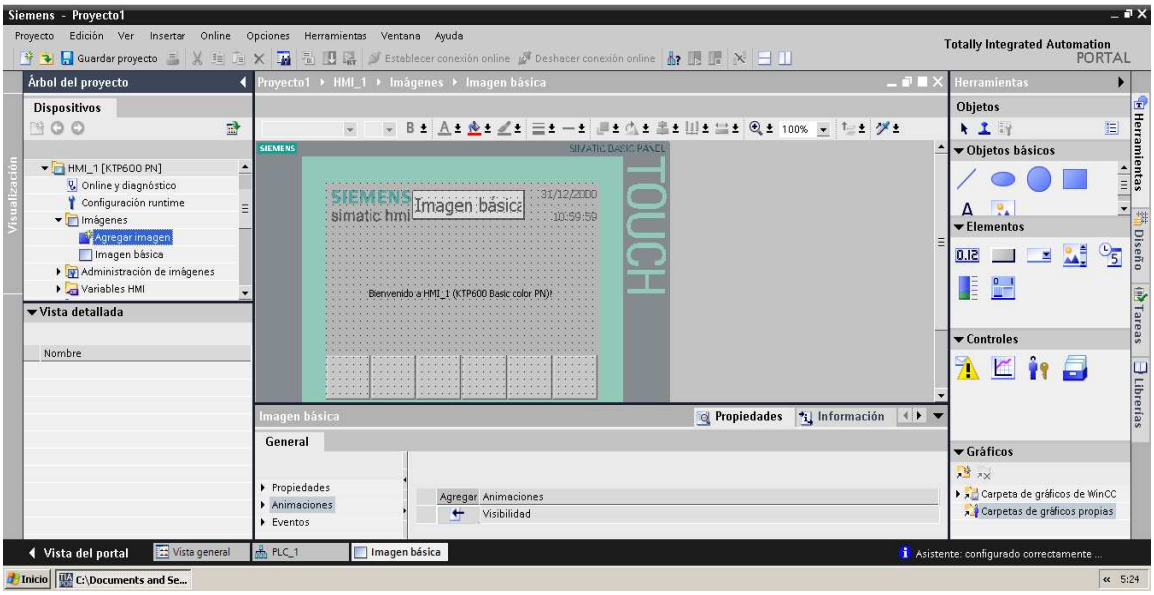

Figura 75. Agregar nueva imagen

Aparece una nueva imagen en la vista de proyecto, se asigna un nombre automáticamente "Imagen_1". Como se observa en la Figura 76 es una pantalla que no contiene objetos gráficos.

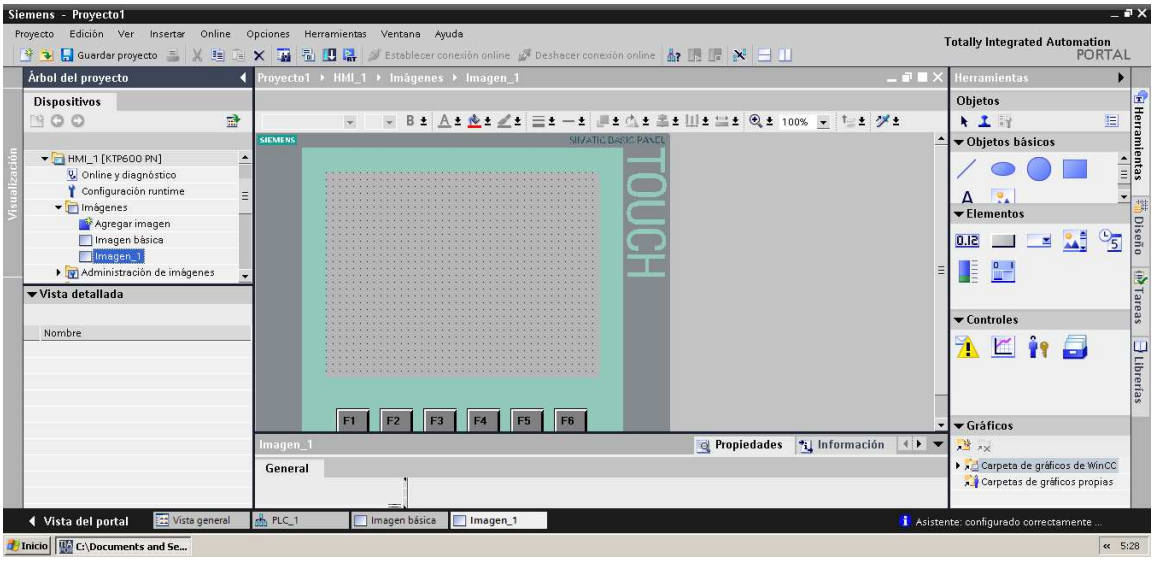

Figura 76. Imagen_1

Cualquier imagen se puede importar directamente desde el portapapeles del computador. Se busca la imagen y se copia, luego se viene a WinCC y se pega.

Figura 77. Imagen de fondo

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

En la parte inferior podemos encontrar las diferentes propiedades de los objetos gráficos, en este caso seleccionamos la Imagen_1 y cambiaremos el color de fondo. Desplegamos la pestaña "*Propiedades*" y damos clic en "*General*", aquí podemos cambiar el nombre de la imagen actual, el color de fondo, el color de la cuadricula, entre otras opciones. En este el color de fondo y la cuadrícula serán blancos debido a que la imagen de fondo escogida es de este color.

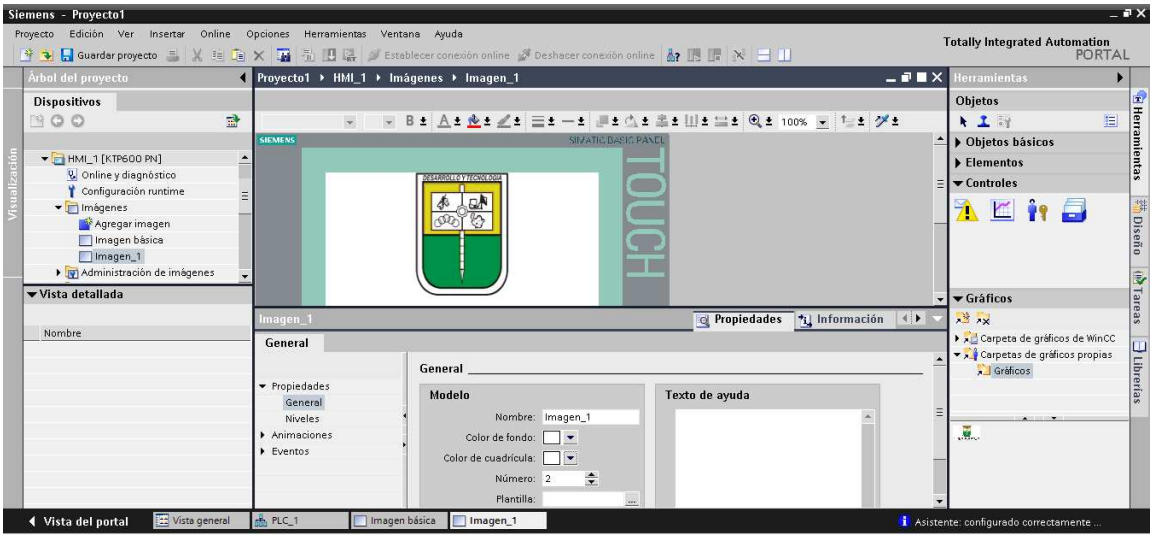

Figura 78. Propiedades generales de la imagen

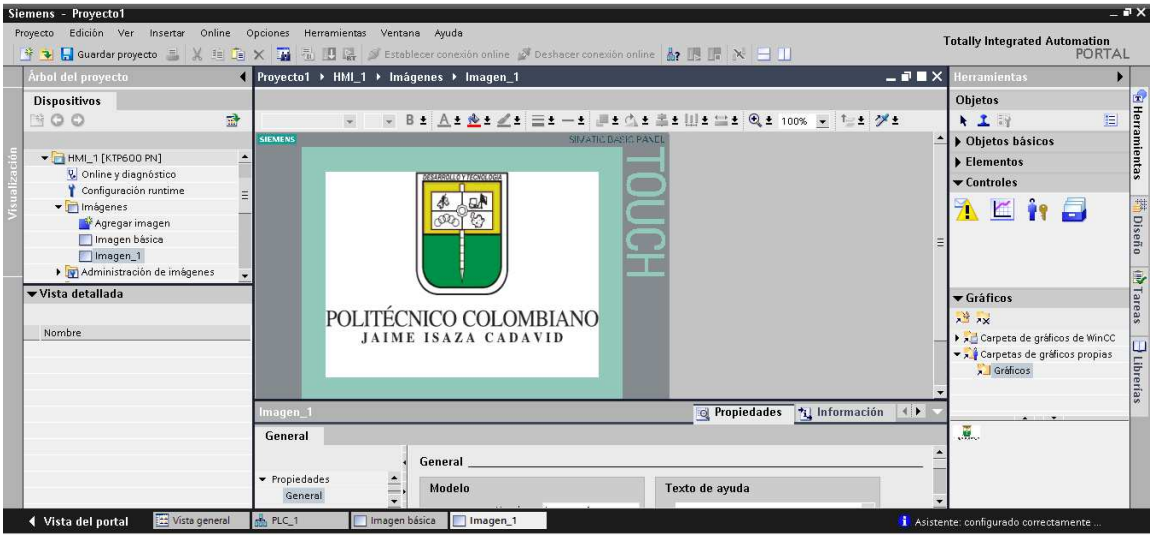

Figura 79. Cambiar color de fondo

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

Ahora insertaremos una nueva imagen, que por defecto se llamará Imagen_2, pero seguiremos ubicados en la Imagen_1, con el fin de insertar un botón que, al presionar, nos lleve a la Imagen_2. En la parte lateral derecha, del catálogo "*Herramientas*", desplegamos la pestaña "*Elementos*", seleccionamos el elemento "*Botón*" y lo llevamos a la Imagen_1.

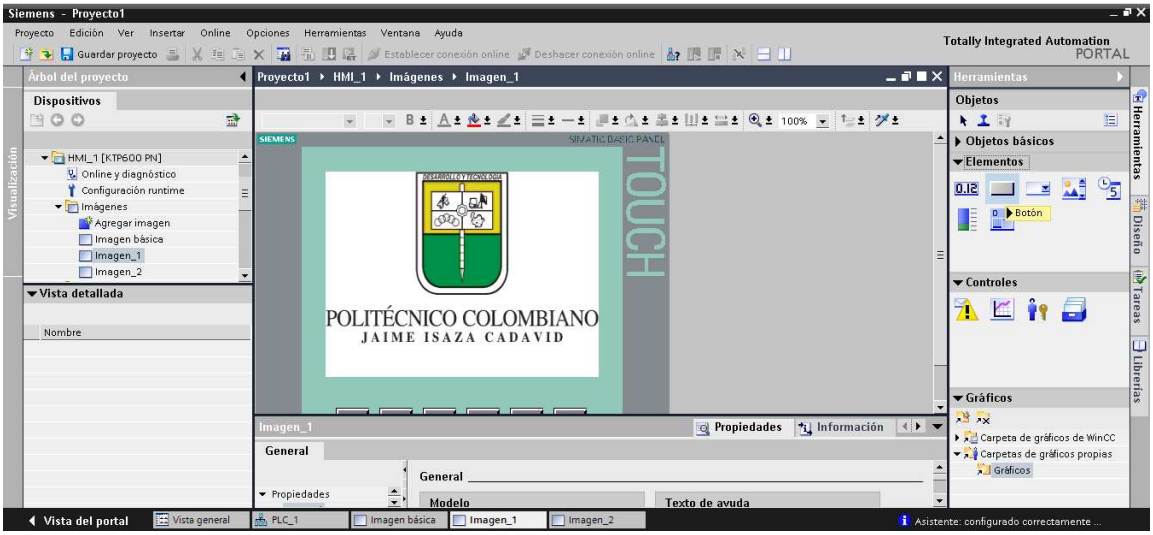

Figura 80. Elemento "*Botón*"

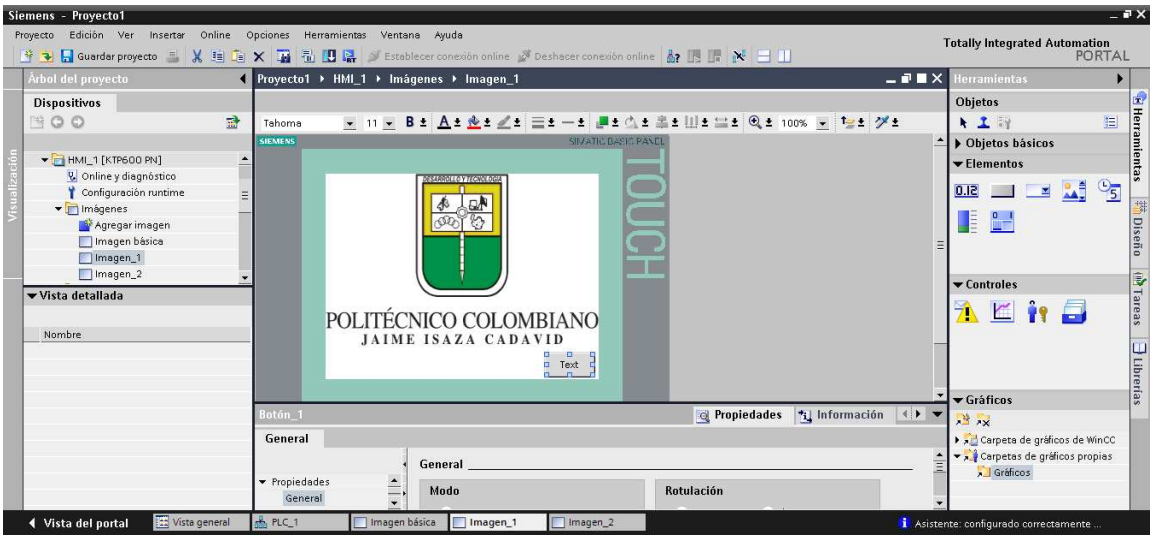

Figura 81. Insertar Botón

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

Si seleccionamos el botón que se insertó, podemos ver en la parte inferior sus propiedades y cambiar el texto que se imprime sobre el botón cuando aún no se ha presionado y cuando se presiona, en las propiedades generales, en “Rotulación” se encuentra como “Soltado” y “Pulsado” respectivamente. Como la finalidad de este botón será llevarnos a la siguiente imagen, el texto para cada opción será “Siguiente”.

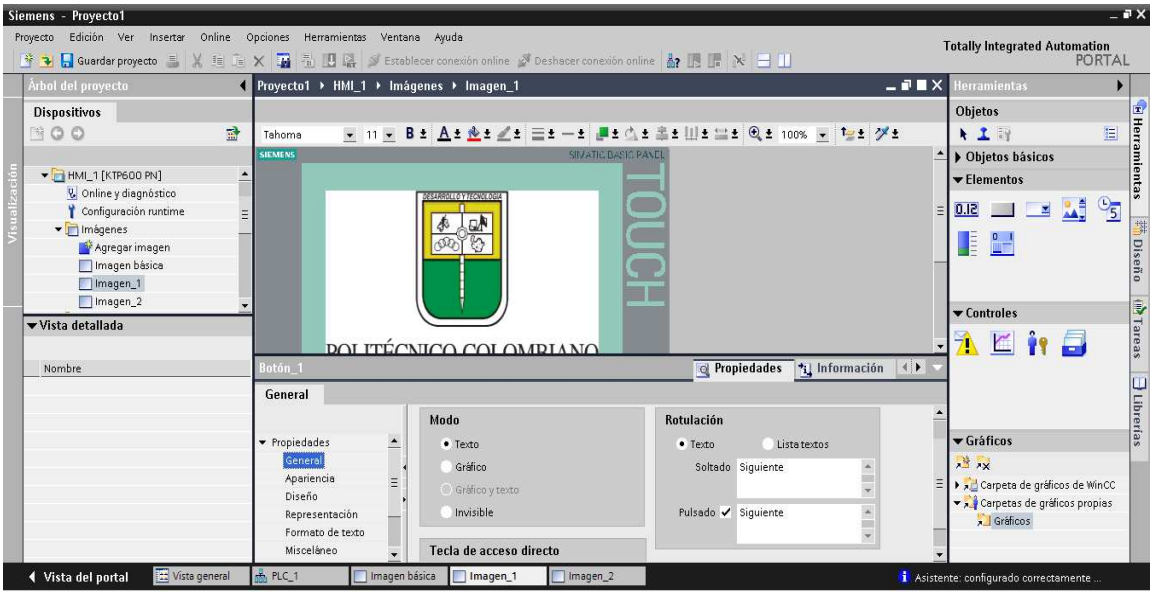

Figura 82. Texto del botón

En la configuración “Modo” se puede poner un gráfico al botón en lugar de texto. Explore estas opciones y utilice la que desee. Para este ejemplo dejaremos la rotulación con texto.

En la pestaña propiedades se pueden configurar otro tipo de características del objeto como la apariencia, diseño, representación, formato de texto, entre otras, en las cuales se puede configurar el color de fondo, el color y tipo de letra, elegir si tiene efecto 3D, la posición y el tamaño del objeto, etc. Explore estas opciones y configure el objeto con las características requeridas.

Seleccionemos de las opciones “Formato de texto”, aquí se ajusta la posición y el tamaño del objeto, pero aquí haremos énfasis en que es importante que en “Adaptación del tamaño” se seleccione la opción “Adaptar objeto al contenido” para ajustar automáticamente el tamaño del botón a la longitud del texto.

Puede utilizar esta función en particular cuando se trabaja en proyectos futuros con selección de idioma para las pantallas HMI. Según el idioma seleccionado, el texto traducido puede ser más corto o más largo que el original. Utilice esta función para asegurarse de que las etiquetas de los botones no se truncan. El tamaño del botón se ajustará automáticamente en caso de cambios en el texto en el original.

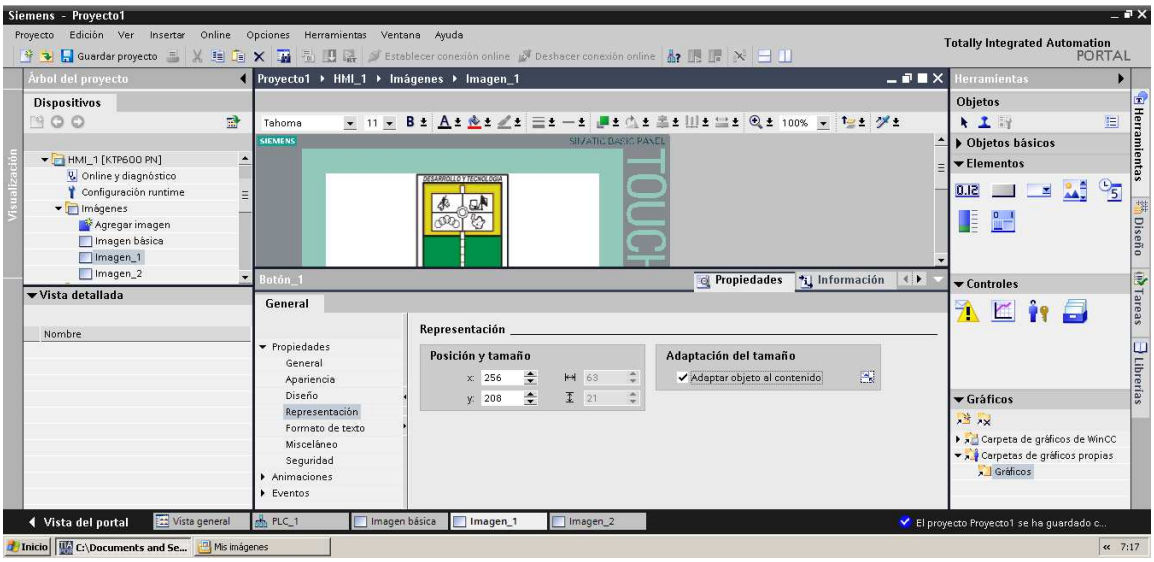

Figura 83. Adaptar objeto al contenido

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

En las opciones desplegamos la pestaña “Eventos” donde se escoge la función que queremos que botón realice cuando en éste se haga un clic, cuando sea pulsado, cuando se suelte, entre otras. Para nuestro caso queremos que pase a la siguiente imagen cuando el botón sea soltado, entonces damos clic en la opción “Soltar.

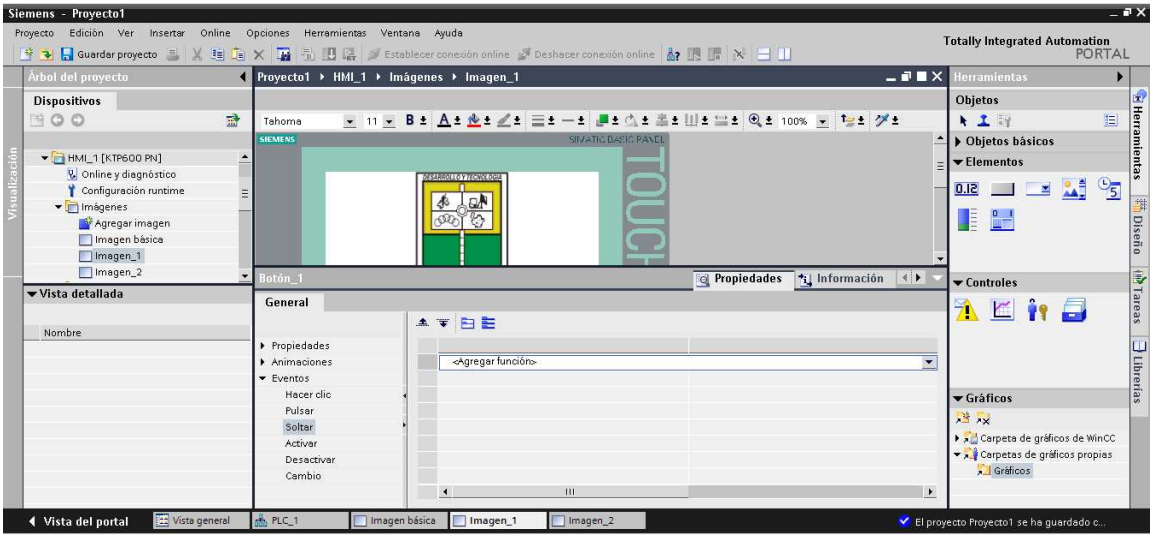

Figura 84. Agregar evento

En el panel desplegamos la lista de “Agregar función” y se observarán todas las funciones posibles que puede realizar el objeto.

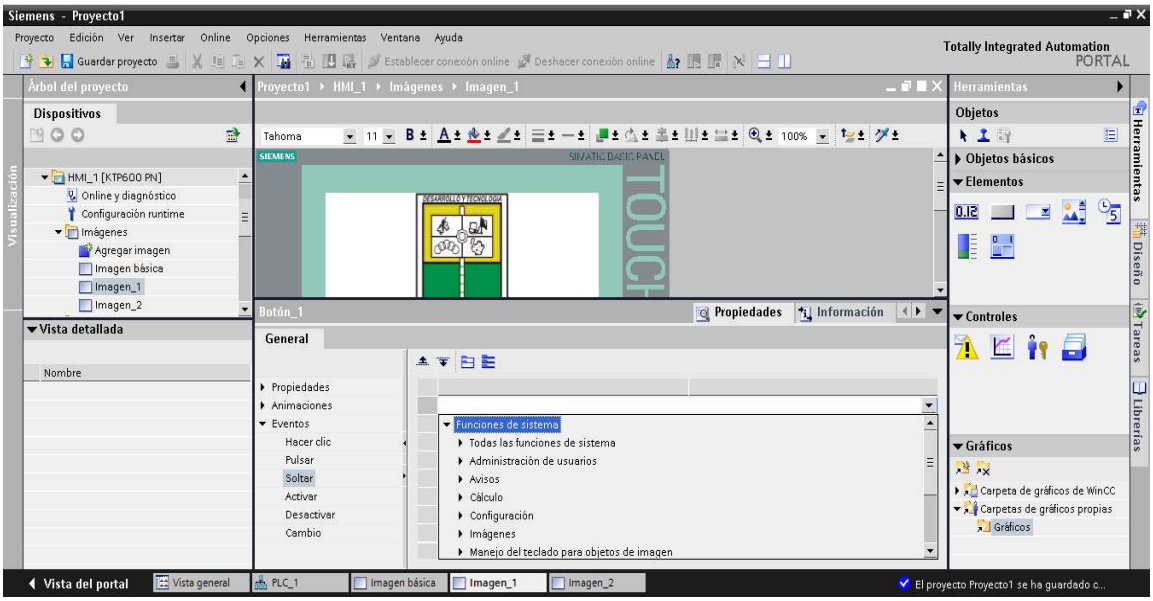

Figura 85. Lista de funciones

Damos clic en la opción “*Imágenes*” y seleccionamos “*Activar imagen*”.

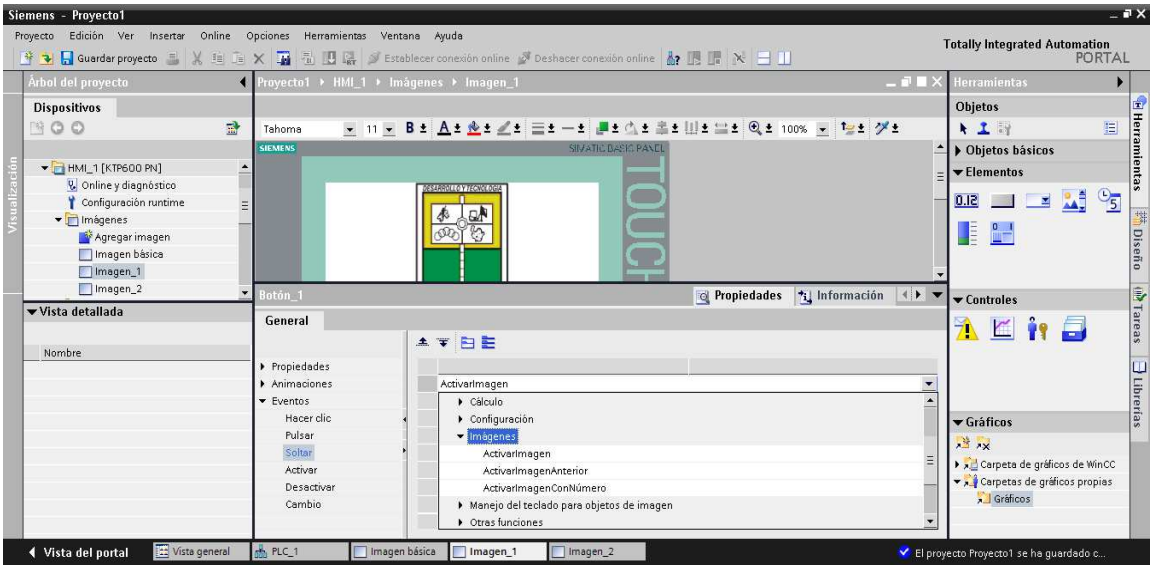

Figura 86. Eventos de imágenes

Después de seleccionar la opción "*Activar imagen*" se observa en la pantalla lo que se ve en la Figura 87.

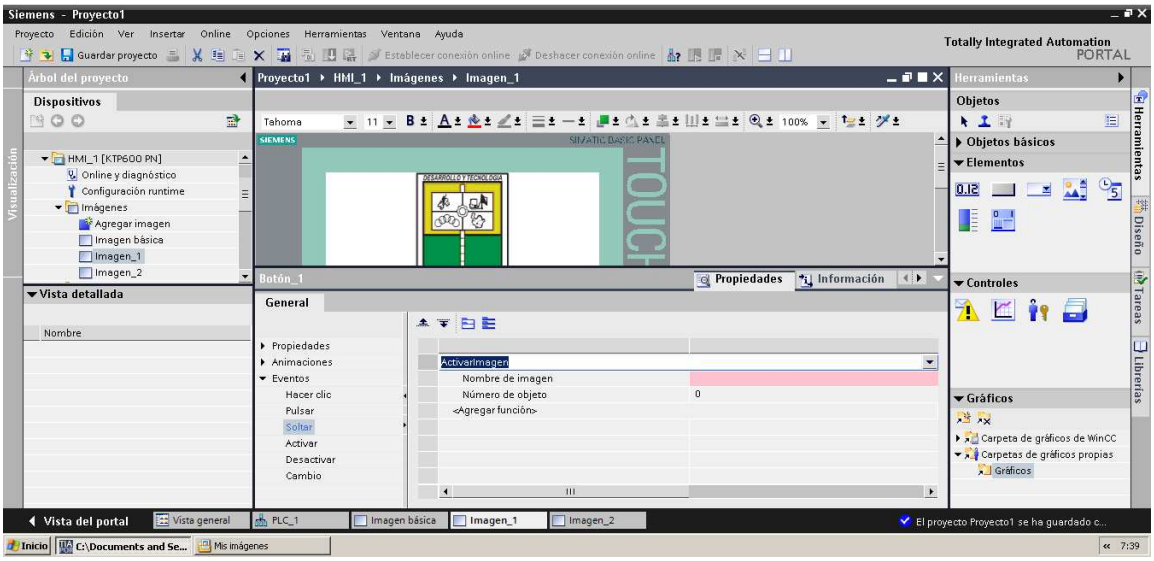

Figura 87. Activar imagen

Dar clic sobre el campo que se resalta en color rojo correspondiente a la pestaña "*Nombre de imagen*", en el cual seleccionaremos la imagen que queremos que se active cuando suceda el evento correspondiente al botón.

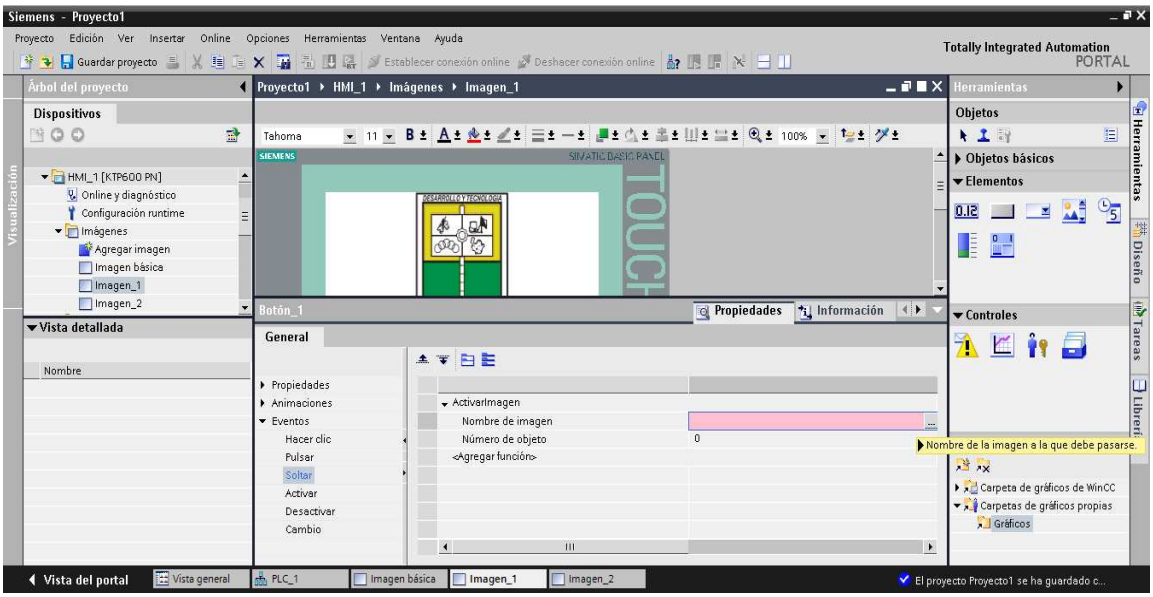

Figura 88. Nombre imagen que debe activarse

Se abre una nueva ventana en la cual se selecciona la imagen que se desea activar, luego damos clic en el botón  para aplicar la configuración y cerrar el cuadro de diálogo.

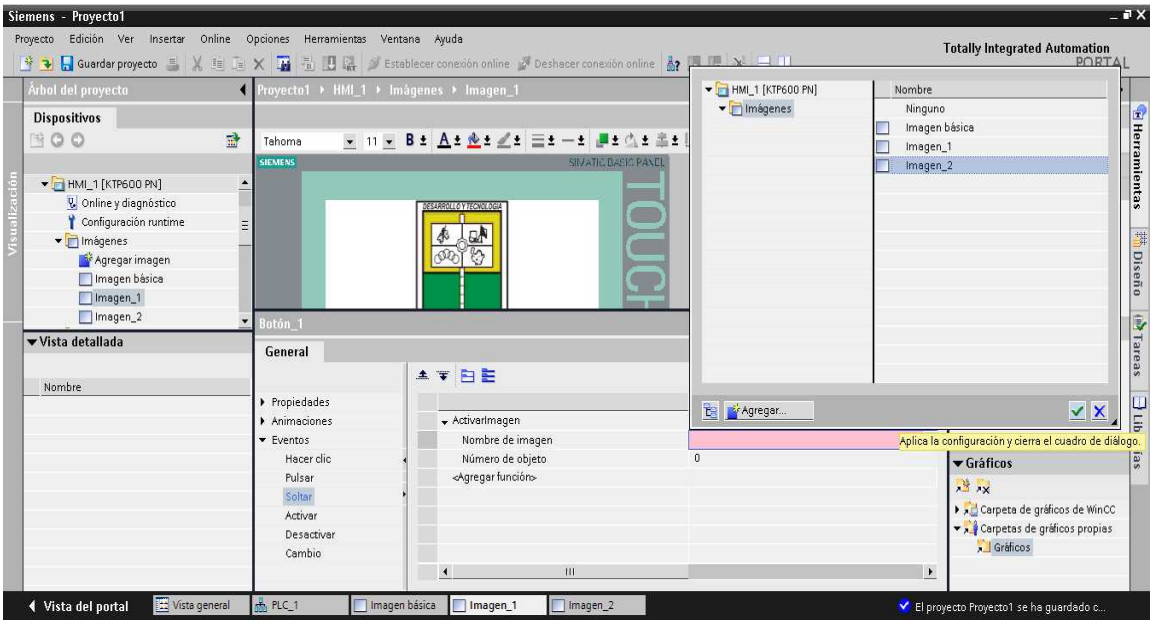

Figura 89. Seleccionar imagen

Cuando la HMI fue añadida, la primera imagen que se configuró automáticamente fue “Imagen básica”, por lo tanto cuando la pantalla se encienda ésta será la primera imagen que se muestra. Sin embargo esta opción se puede cambiar si deseamos que sea otra imagen la que aparezca inicialmente. Por ejemplo configuremos “Imagen_1” como la imagen de arranque.

Damos doble clic a "*Configuración runtime*" en el árbol del proyecto.

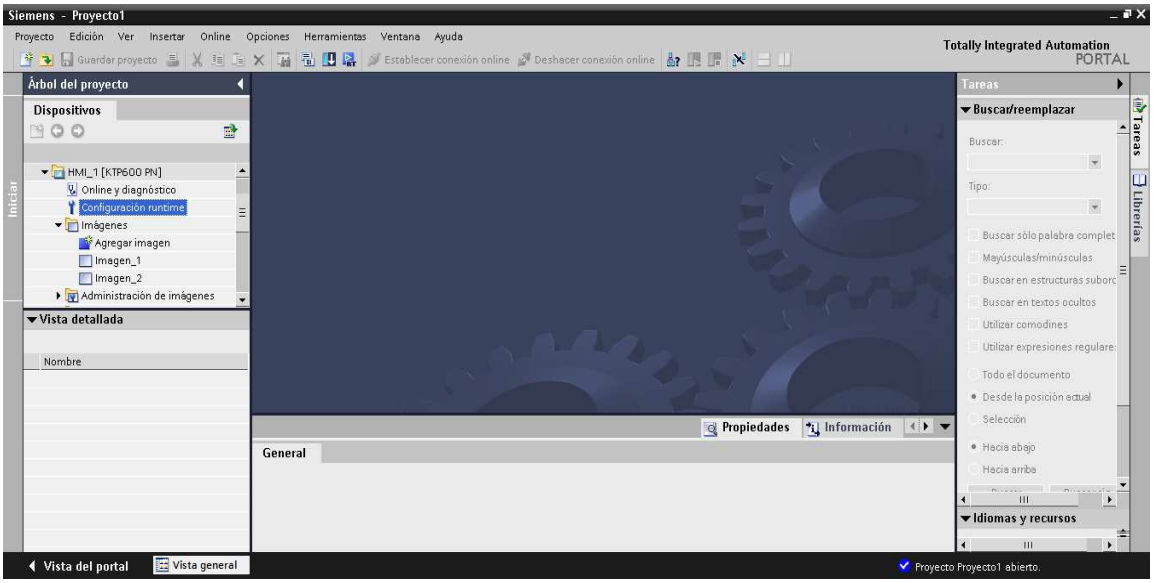

Figura 90. Configuración runtime

Se abre una nueva ventana, en el panel lateral izquierdo nos ubicamos en la pestaña "*Imágenes*", en el campo "*Imagen inicial*" se da clic en el ícono y en la lista de imágenes desplegada seleccionamos la que deseamos sea la imagen inicial de la HMI, en este caso la Imagen_1.

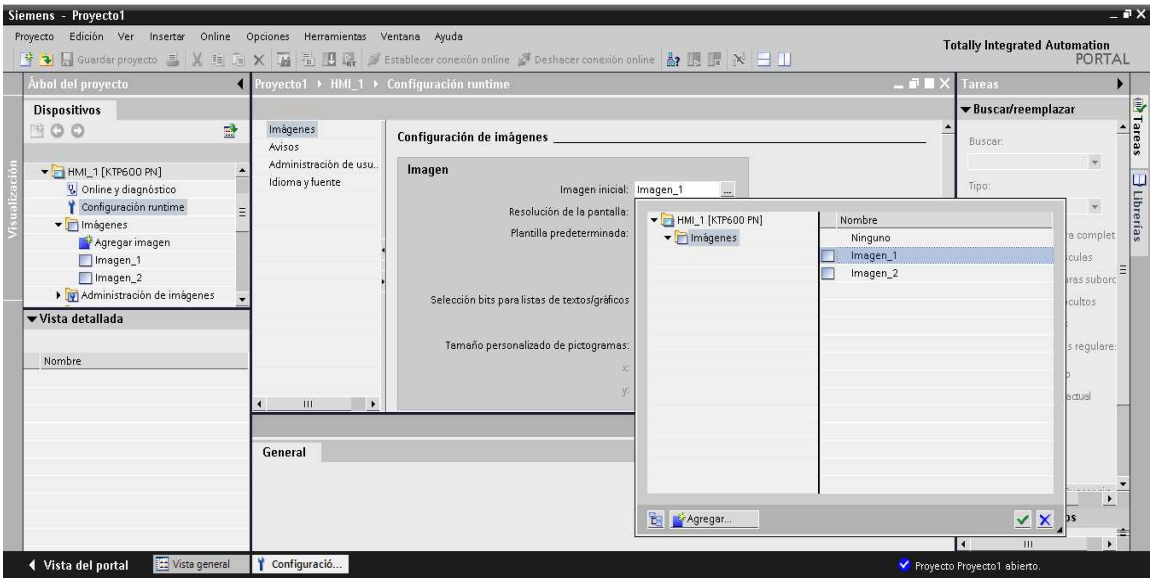

Figura 91. Seleccionar imagen inicial

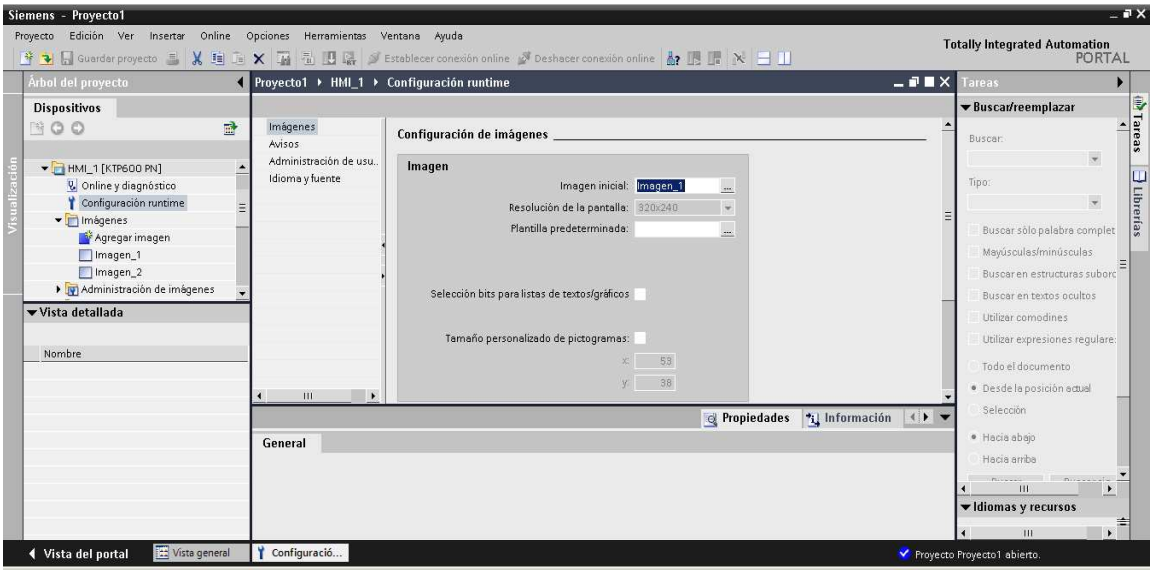

Figura 92. Imagen inicial

Después de tener configurada la imagen inicial, nos ubicamos en la Imagen_2 y agregamos un botón que al ser pulsado nos devuelva a la imagen inicial. Cambiamos el color de fondo y de la cuadrícula de la Imagen_2 por blanco. Con estos cambios observamos la Imagen_2 como se muestra en la Figura 93.

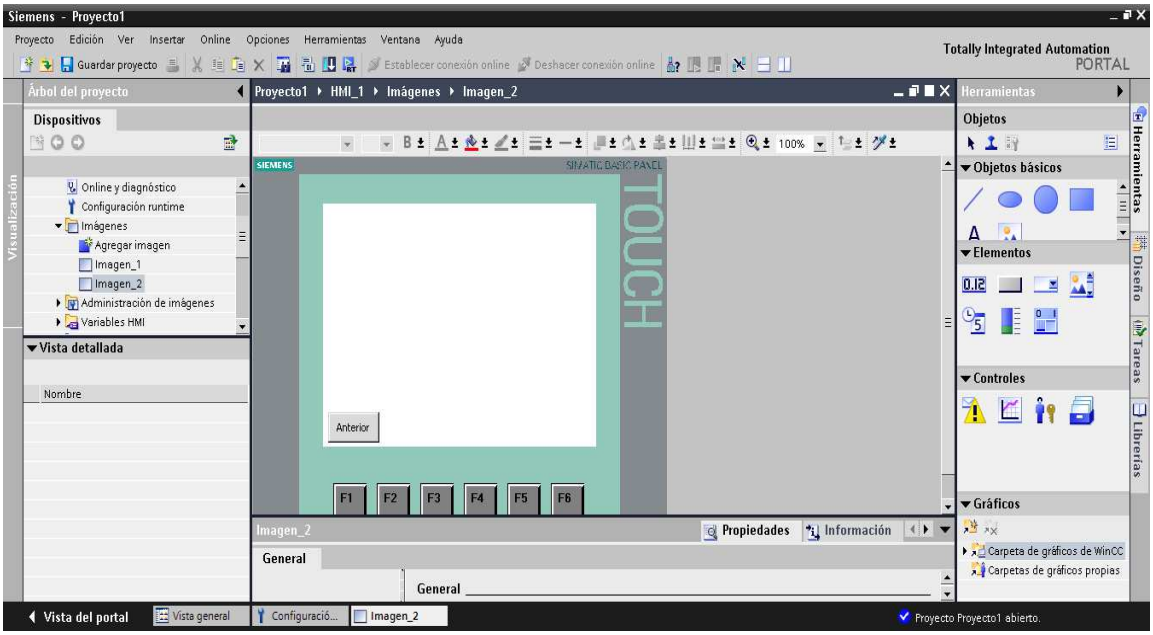

Figura 93. Configurar Imagen_2

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

Ahora vamos a añadir un pulsador que encienda y apague un led. Antes de diseñar la imagen en la HMI debemos hacer el código en el PLC que nos relacione las variables que serán asignadas a los objetos gráficos de la HMI. En el panel lateral izquierdo desplegamos la pestaña correspondiente al PLC_1.

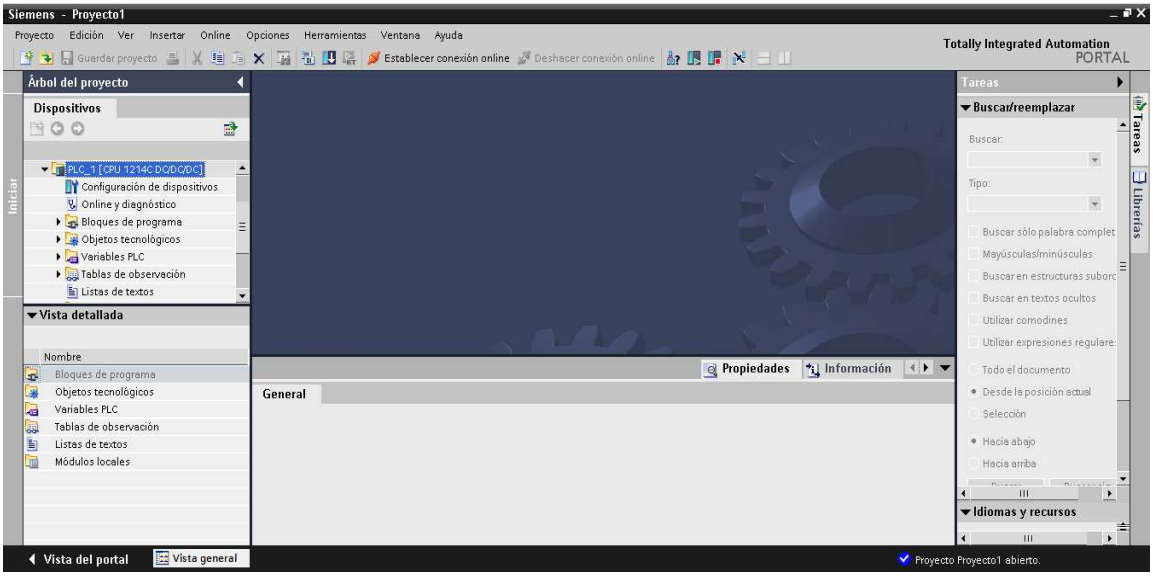

Figura 94. Seleccionar PLC_1

De la lista que se desplegó damos clic a la pestaña "*Bloques de programa*" y luego seleccionamos la opción "*Main [OB1]*", para escribir el código que simulará el encendido y apagado de un led en la HMI.

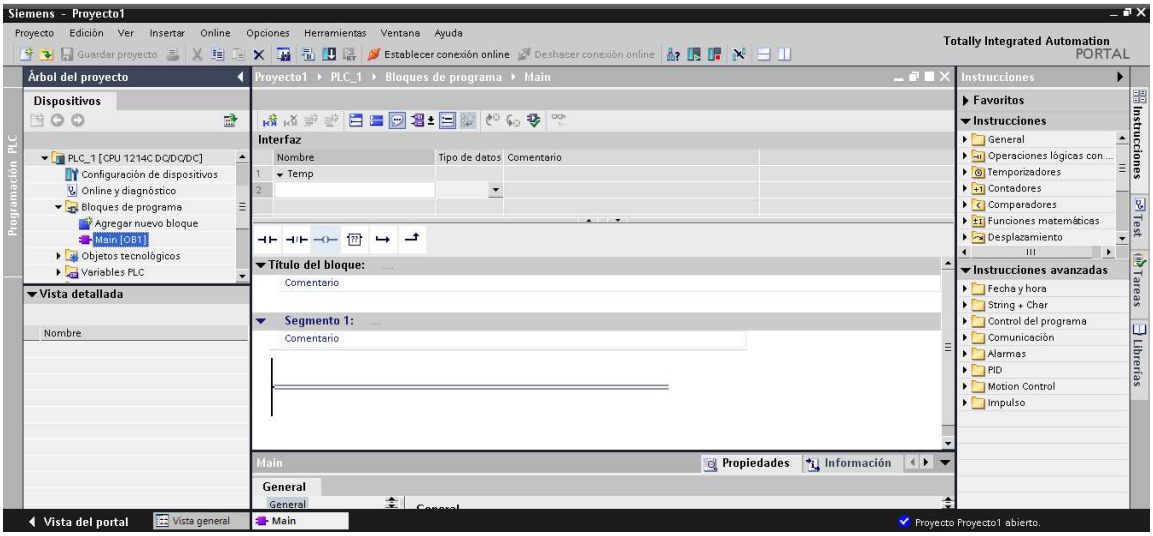

Figura 95. Main [OB1]

Vamos a asignar las direcciones del PLC que se utilizarán y sus respectivas etiquetas, esto lo hacemos en "*Variables PLC*". La dirección M0.0 corresponde al interruptor y la Q0.0 la correspondiente al led. Estos tipos de datos solo tomarán valores lógicos, es decir 0 ó 1, por lo que el tipo de dato será booleano. Observe esta asignación en la Figura 96.

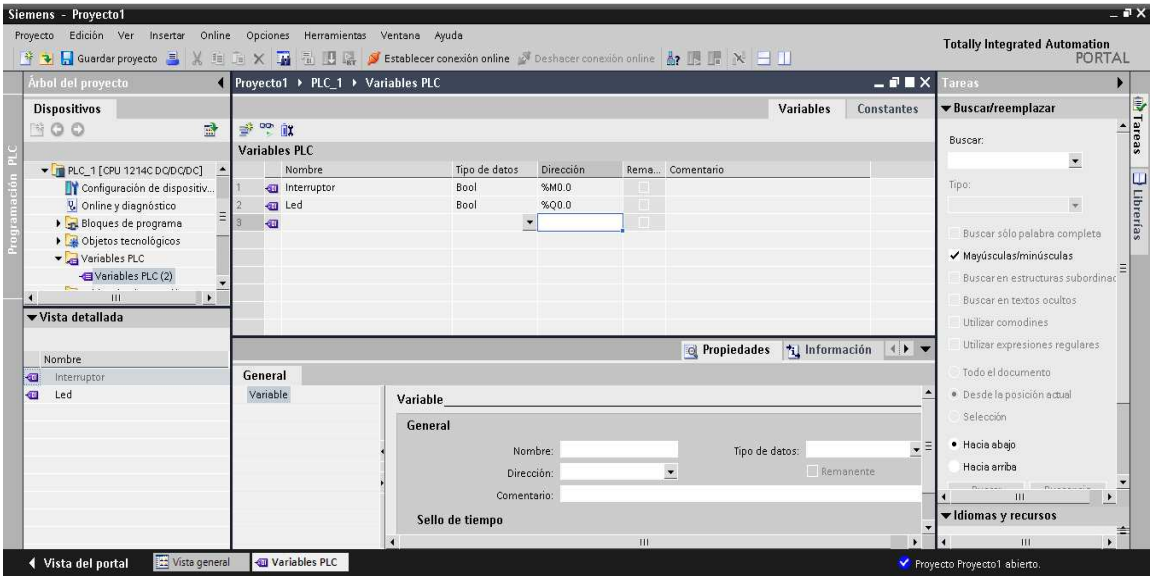

Figura 96. Variables del PLC

Volviendo al Main, agregamos un contacto normalmente abierto en el segmento 1, su dirección será la correspondiente al interruptor, es decir la M0.0 y al final del segmento insertamos una bobina, su dirección es la Q0.0, es decir la que activará el led de la HMI. Observe la Figura 97.

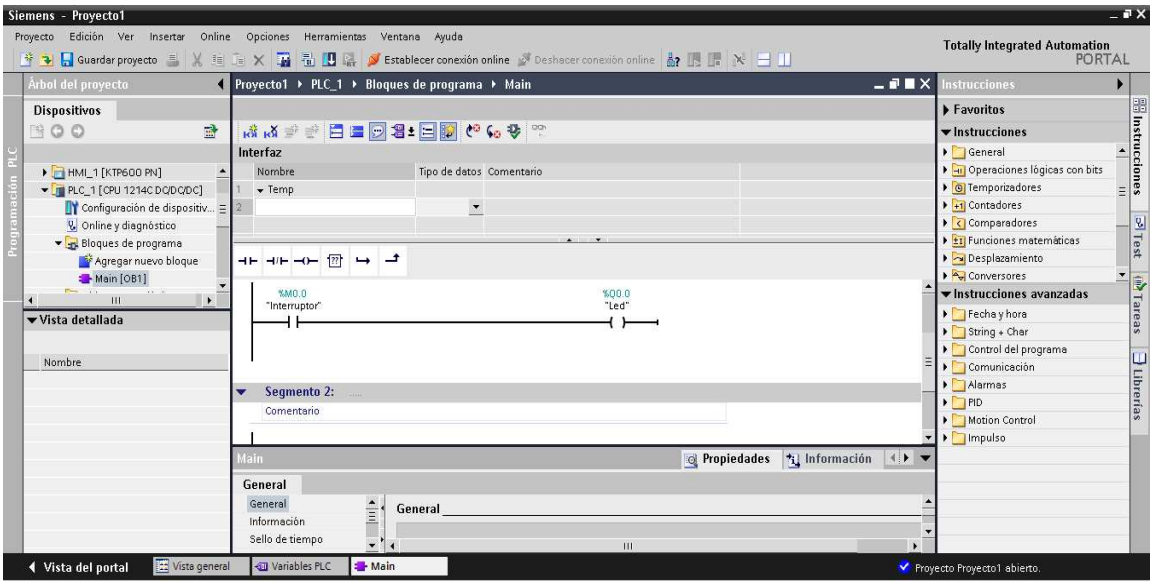

Figura 97. Insertar contacto NA y Bobina

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

Luego de esto cargamos el programa al PLC, para que cuando diseñemos la HMI corra simultáneamente el programa que acabamos de hacer.

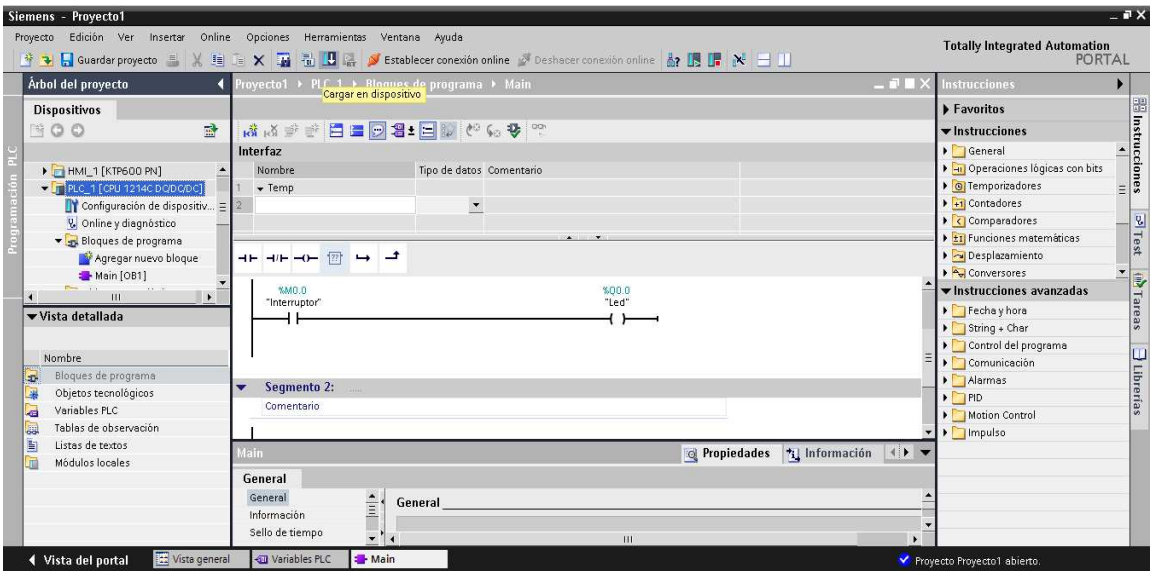

Figura 98. Cargar programa PLC

Regresamos a las imágenes de la HMI, seleccionamos la imagen_2, que es en la cual vamos a crear la interfaz que simule el encendido y apagado de un led. De la librería elementos, seleccione un interruptor y arrástrelo hasta la imagen_2 con clic sostenido.

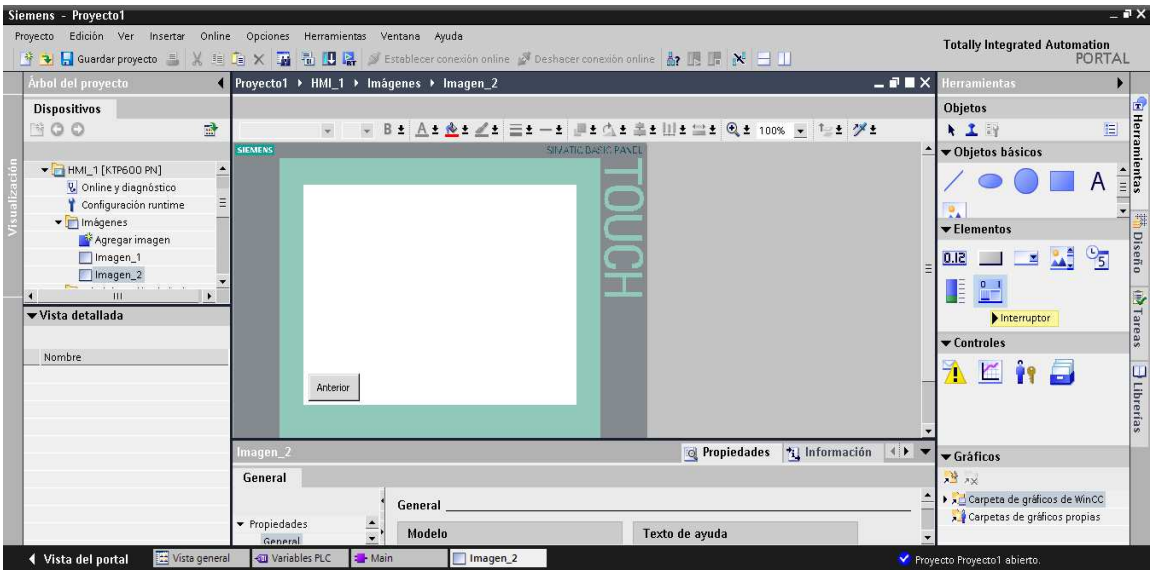

Figura 99. Seleccionar interruptor

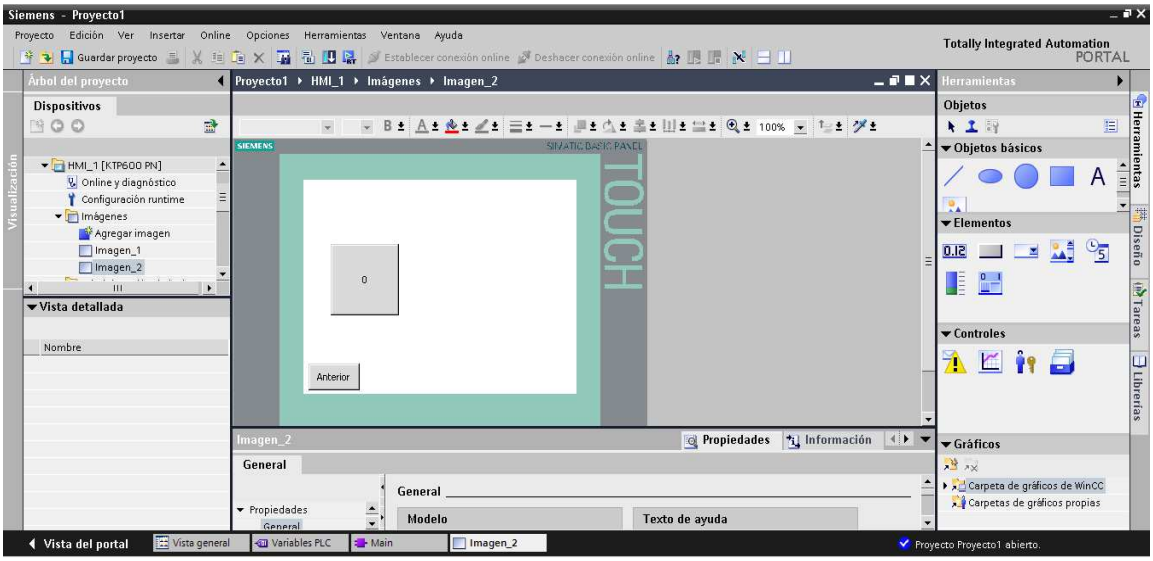

Figura 100. Arrastrar interruptor a la imagen

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

Seleccionamos el interruptor y en la parte inferior, desplegamos la pestaña "*Propiedades*", elegimos la opción "*General*". En la ventana "*Proceso*", en el campo "*Variable",* definimos la variable del PLC al que estará asociado el interruptor, en este caso la marca M0.0 que es la activa la bobina correspondiente al led.

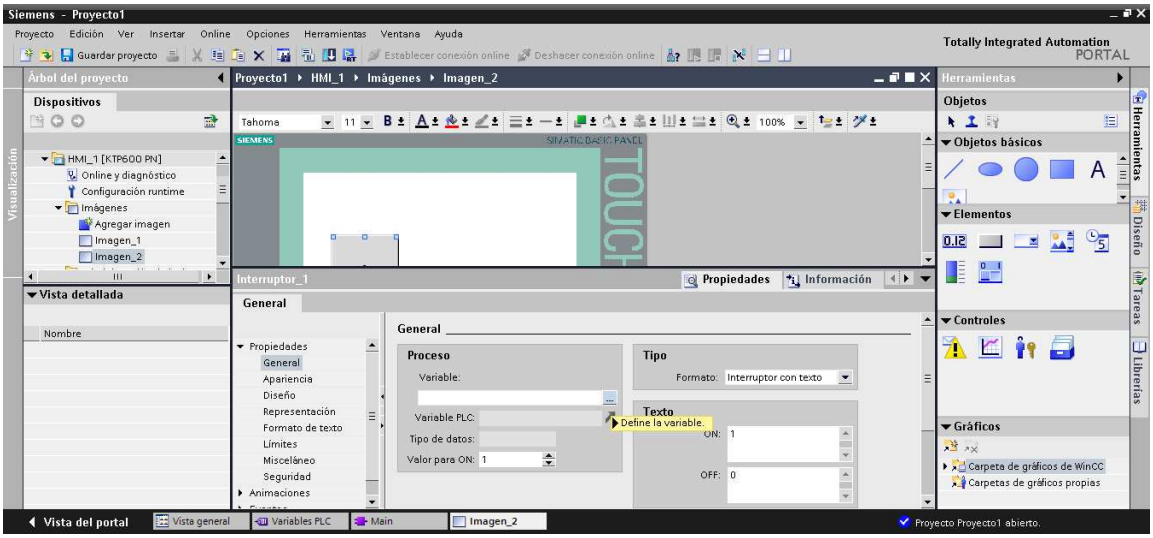

Figura 101. Definir variable interruptor

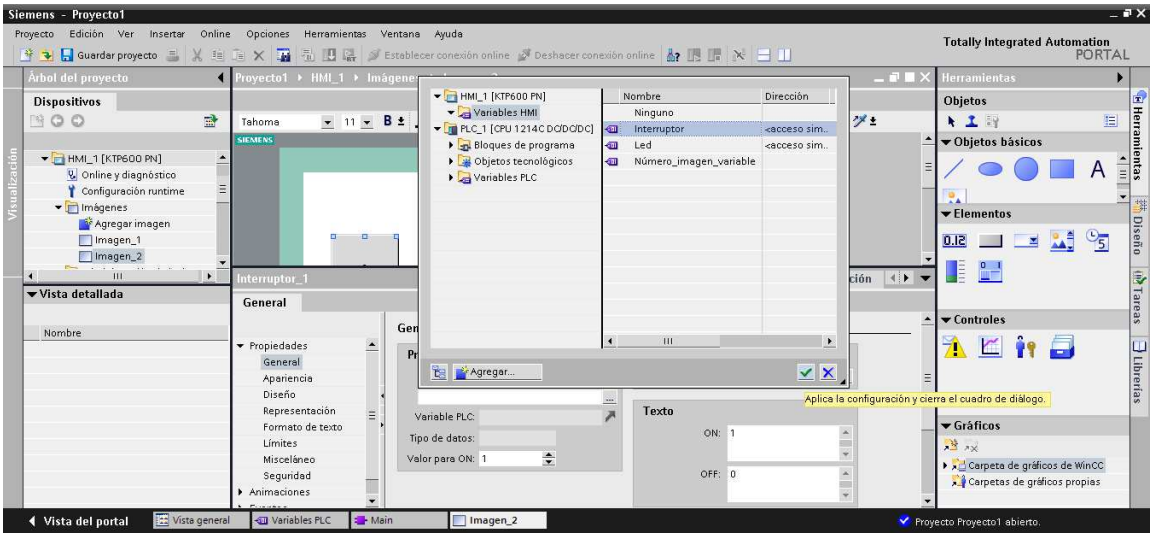

Figura 102. Seleccionar M0.0 Interruptor

Ahora en el panel "*Tipo*", en el campo "*Formato*" seleccionamos la opción "Interruptor con gráfico", esto con la finalidad de poner un gráfico que reemplace la forma que tiene por defecto el elemento interruptor.

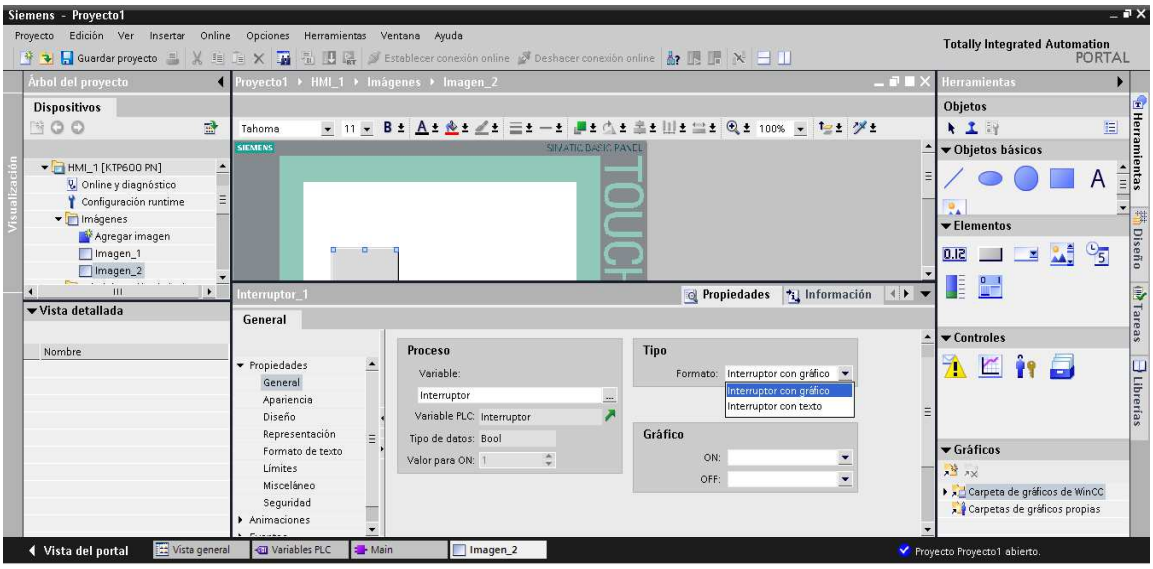

Figura 103. Interruptor con gráfico

En el panel "*Gráfico*" podemos observar los dos estados posibles para el interruptor (ON, OFF) si desplegamos la pestaña correspondiente a cada uno podemos seleccionar el gráfico que se visualizará cuando se presente el respectivo estado lógico. Pero para que en este catalogo de imágenes salga el gráfico deseado primero tenemos que agregarlo a la imagen directamente desde la librería de gráficos que está ubicada en el panel lateral derecho. Nos ubicamos en la siguiente ruta "*Carpeta de gráficos de WinCC\Symbol Factory Graphics\ SymbolFactory 256 Colors\3-D Pushbuttons Etc*", seleccionamos y arrastramos hasta la imagen un interruptor presionado y uno que este sin presionar. Como se muestra en la Figura 104.

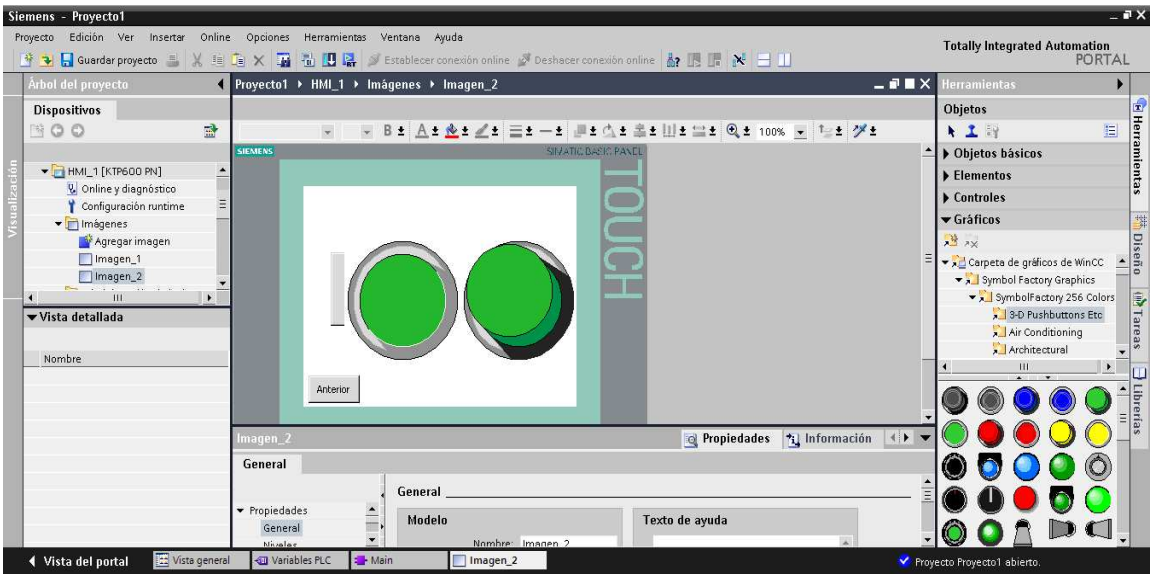

Figura 104. Insertar interruptores gráficos

Luego simplemente los seleccionamos y los eliminamos de la imagen. Como se dijo anteriormente, esto se hace para que estos aparezcan en el catálogo de imágenes que se desplegará para cada uno de los estados lógicos.

Seleccionamos nuevamente el interruptor y en el panel "*Gráfico*" de las propiedades generales, desplegamos la pestaña para el estado "*ON*" y seleccionamos de la lista el interruptor presionado y aplicamos cambios, como se muestra en la Figura 105.

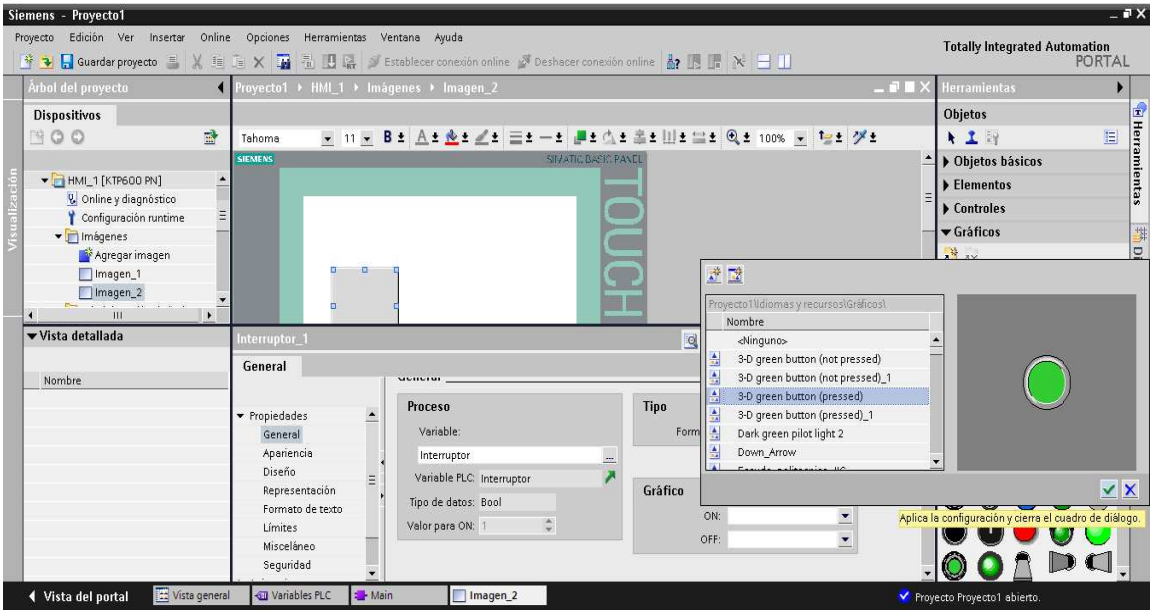

Figura 105. Seleccionar gráfico interruptor ON

Para el estado "*OFF*" hacemos exactamente lo mismo, pero seleccionamos el interruptor que no está presionado. Como se muestra en

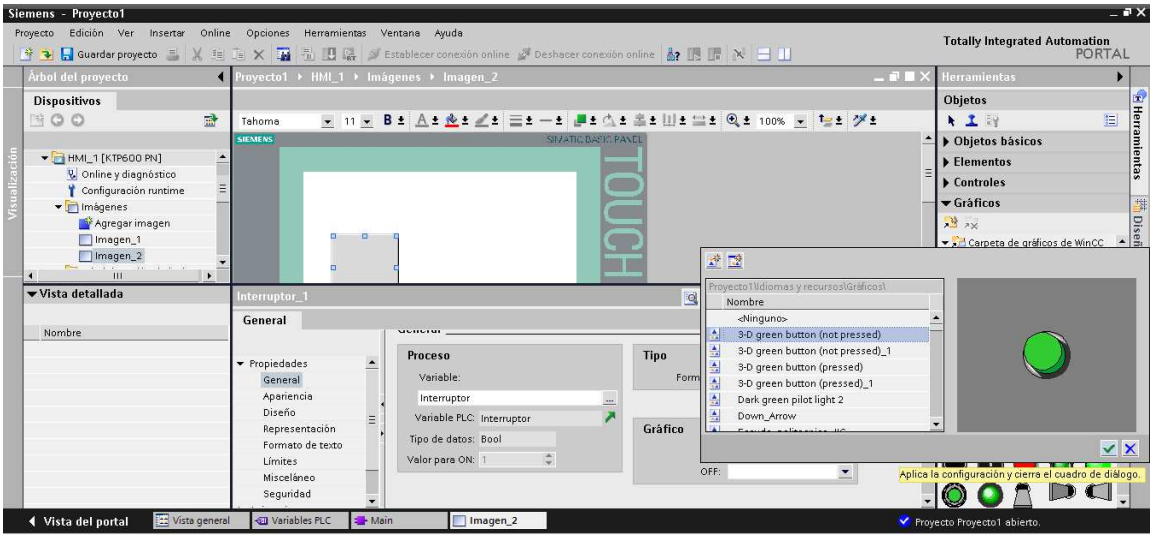

Figura 106. Seleccionar gráfico interruptor OFF

Ahora podemos observar que el interruptor ha cambiado de apariencia, ha tomado la forma del gráfico que seleccionamos en el paso anterior. Seleccionamos el interruptor y en propiedades, "*Apariencia*", seleccionamos blanco para el color de fondo, con la finalidad de que sea el mismo color del fondo de la imagen y no se observe el marco que encierra el interruptor.

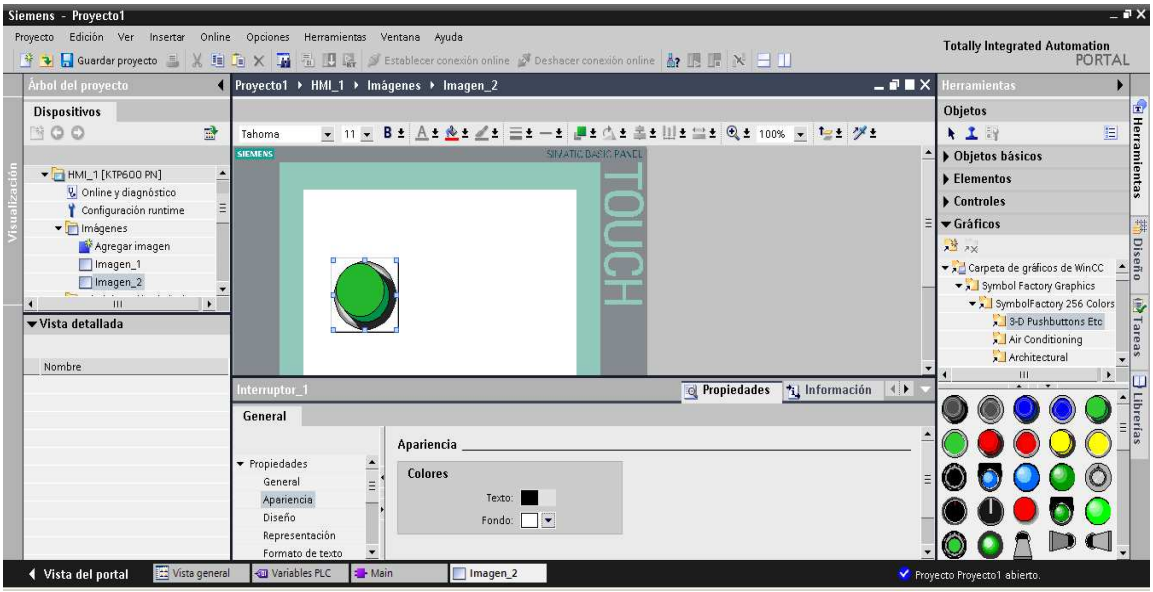

Figura 107. Cambiar color de fondo interruptor

Ahora seleccionamos la opción "*Diseño*" de las propiedades generales y desactivamos la casilla "*Efecto 3D*" para que no se vea el objeto resaltado en la imagen.

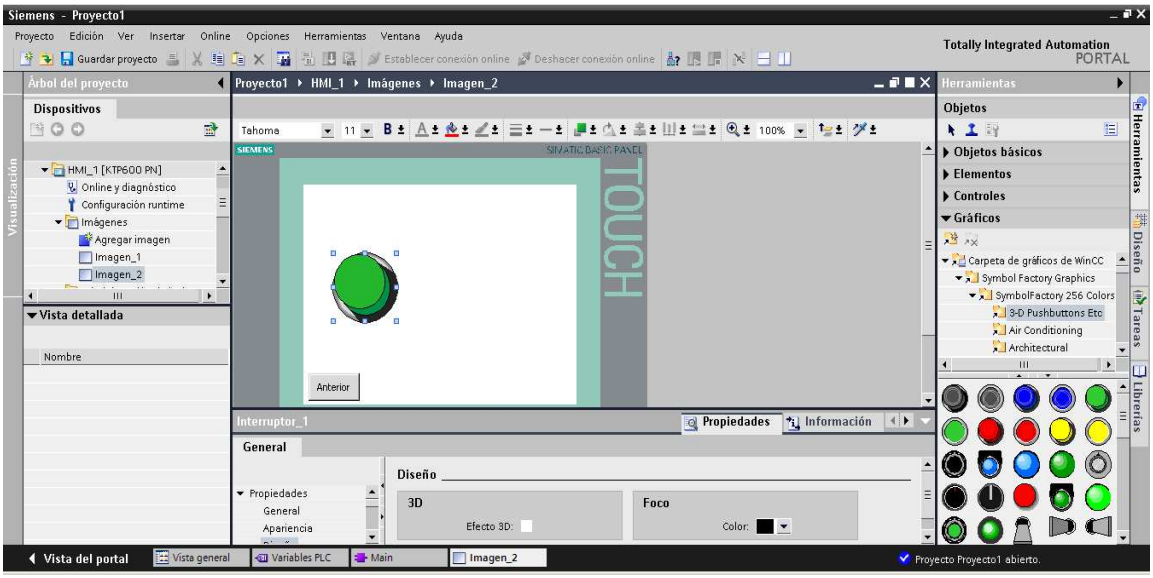

Figura 108. Desactivar efecto 3D

Ahora nos falta agregar el gráfico que simule el led que será activado por el interruptor. De la misma librería de gráficos de WinCC seleccionamos un led verde que será el que se visualice en estado ON del interruptor y un led rojo que se visualice para el estado OFF del interruptor.

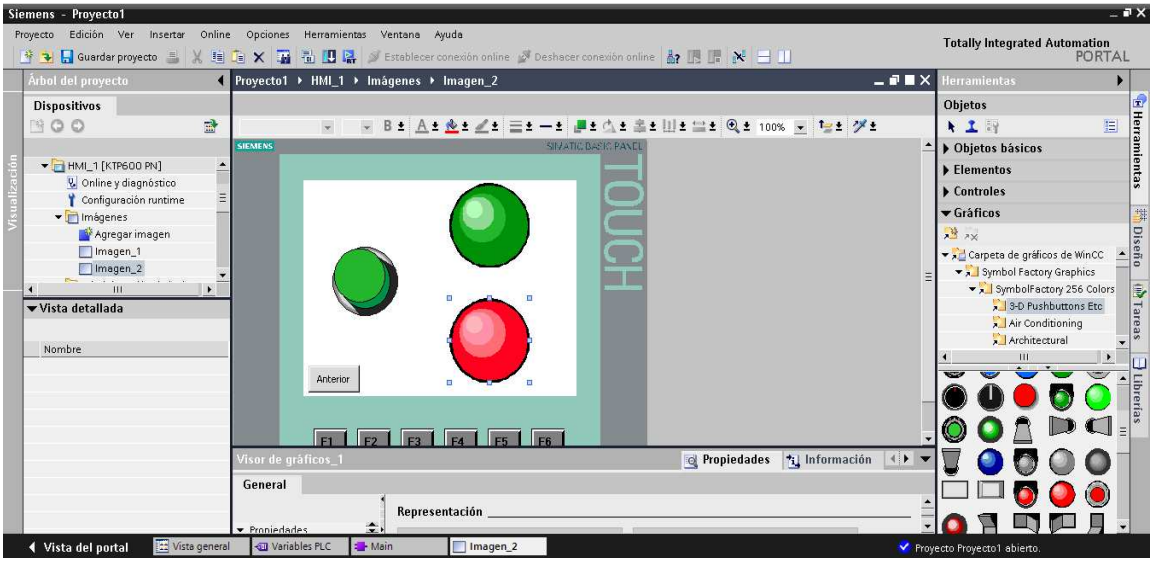

Figura 109. Insertar led

Seleccionamos el led de color verde y en la parte inferior desplegamos la pestaña “*Animaciones*”, elegimos la opción “*Nueva animación*” y damos doble clic en la opción “*Visibilidad*”.

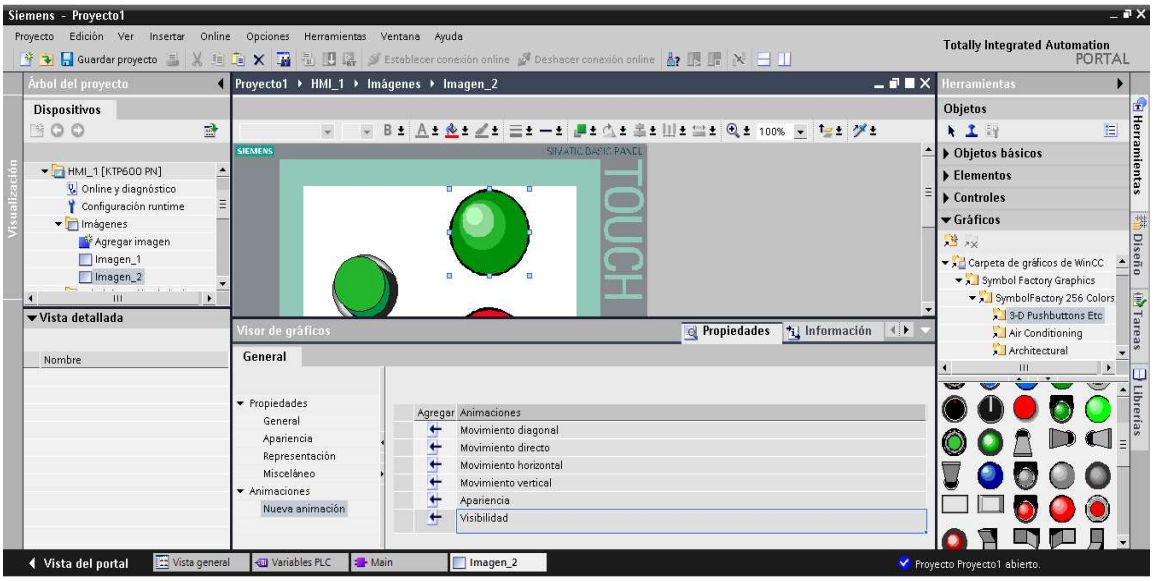

Figura 110. Nueva animación, Visibilidad

Seleccionamos la variable del PLC correspondiente al led y el rango en el cual el grafico estará visible será, para el led verde, el estado lógico alto, es decir, cuando la variable tome el valor de 1. La configuración queda como se observa en la

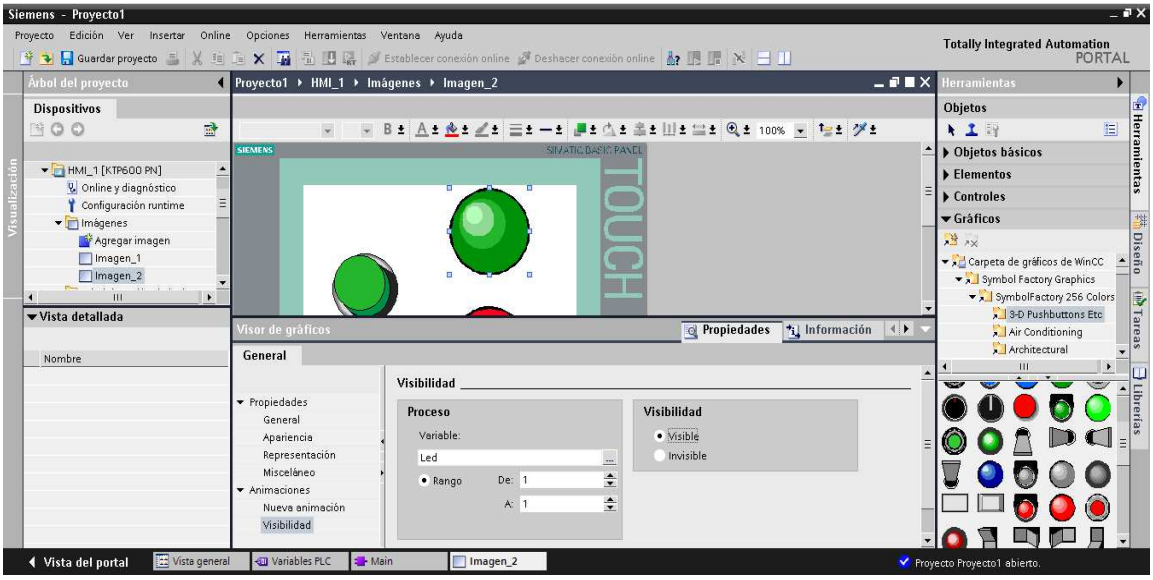

Figura 111. Configurar animación led verde

Ahora seleccionamos el led rojo, insertamos una animación de visibilidad como se hizo con el led verde, la diferencia es que para este caso el rango en el que estará visible el objeto gráfico es cuando la dirección Q0.0 tenga un nivel lógico bajo, es decir un 0.

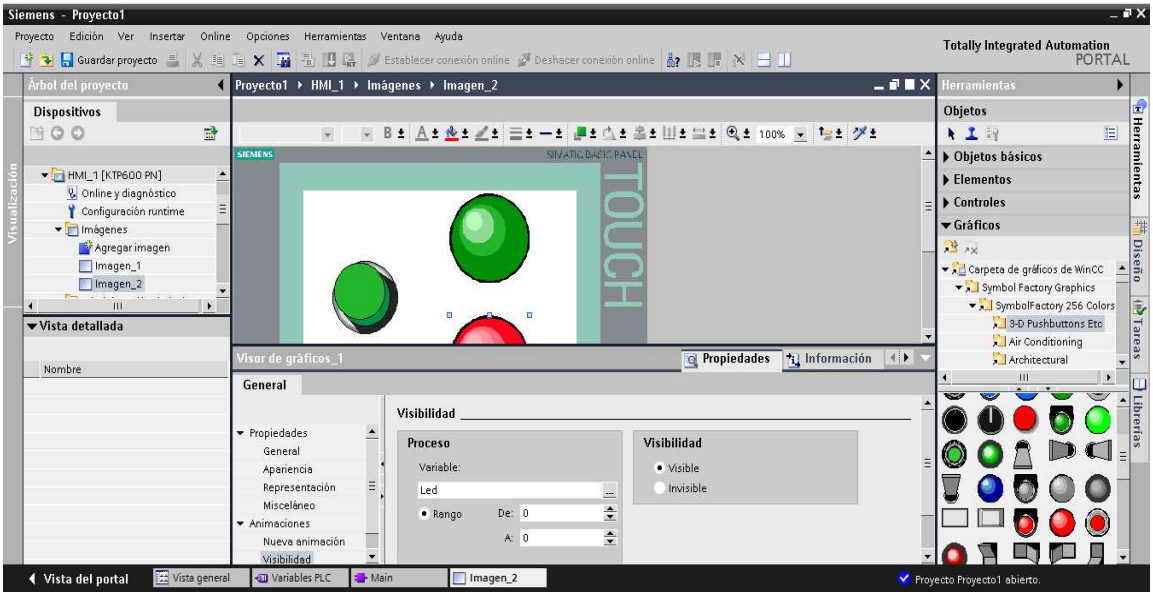

Figura 112. Configurar animación led rojo

Por último ubicamos los led en la misma posición sobre la imagen, la finalidad es que se vea uno o el otro dependiendo del estado lógico del interruptor, así creará el efecto de que cambia de color cada que el interruptor es activado o desactivado.

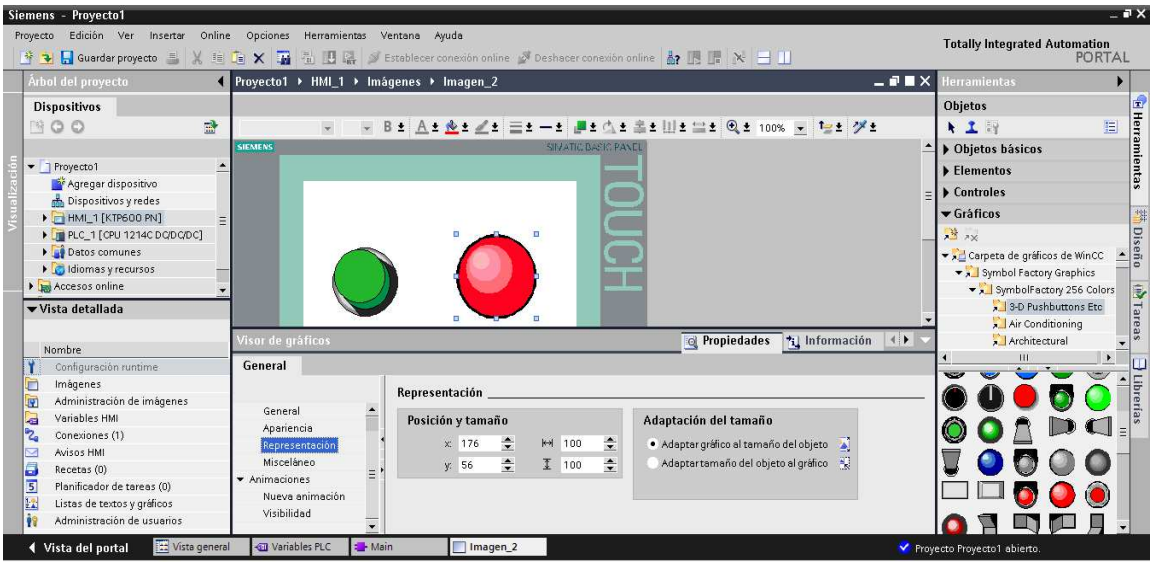

Figura 113. Configurar posición led rojo

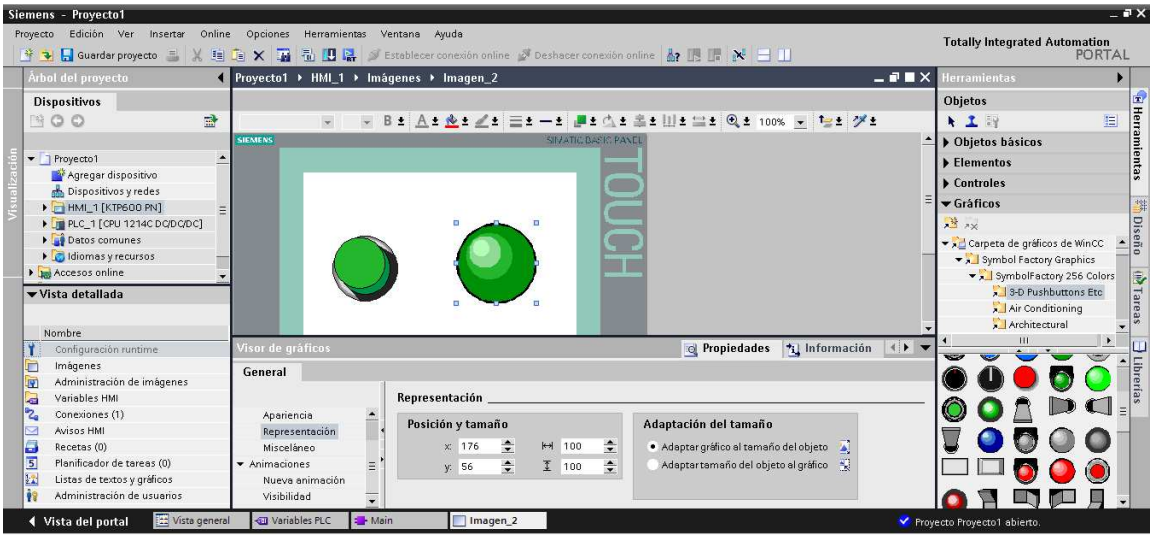

Figura 114. Configurar posición led verde

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

Ahora vamos a cargar la configuración a la pantalla HMI para proceder a visualizar el proceso. En el árbol de proyecto seleccionamos la HMI y luego damos clic en el botón "*Cargar en dispositivo*".

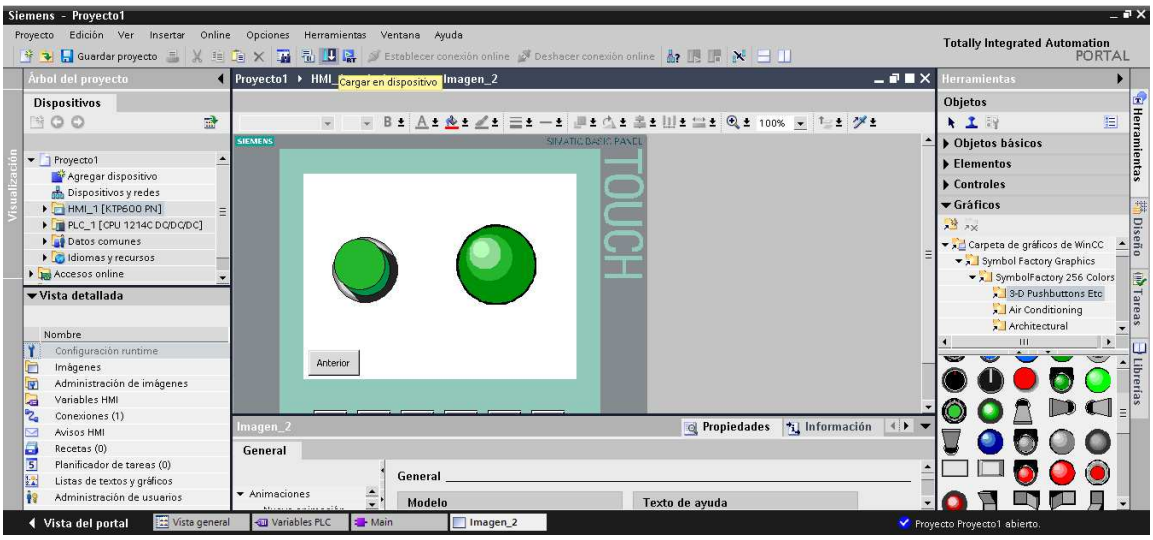

Figura 115. Cargar dispositivo

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

Ahora podemos ver en pantalla que cuando el interruptor no está presionado se observa el led rojo (apagado) como se puede ver en la Figura 116 y cuando se presiona el interruptor se observa el led verde (encendido) como se observa en la Figura 117.

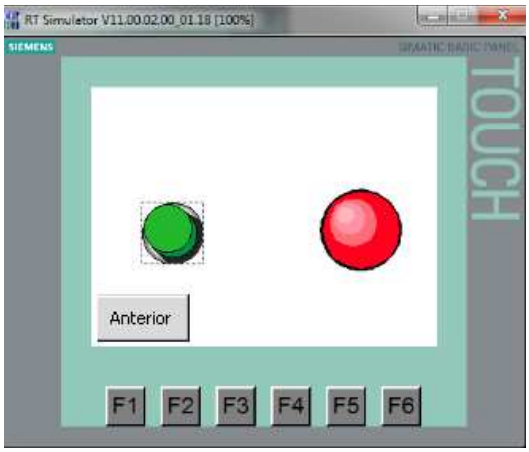

Figura 116. Interruptor OFF, Led apagado

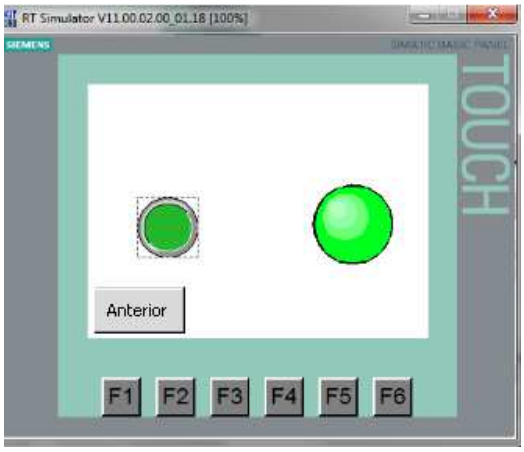

Figura 117. Interruptor ON, Led encendido

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

8. PROFIBUS

Process Field BUS, PROFIBUS es un estándar de bus de campo abierto, independiente del fabricante, permite que los dispositivos de distintos fabricantes certificados en PROFIBUS se puedan comunicar sin la necesidad de utilizar interfaces; está diseñado para satisfacer las necesidades de comunicación de los procesos industriales discretos, continuos y procesos distribuidos.

En las fábricas muchos componentes como (válvulas, actuadores, accionamiento, transmisores etc.), generalmente operan en lugares alejados de las computadoras o autómatas. Por esto, se instalan unidades periféricas descentralizadas (estaciones remotas de entradas y salidas) en área de campo (espacio físico donde se efectúa el proceso de la fábrica), estas estaciones remotas deben comunicarse a través de un bus de comunicación con los computadores ubicados en las diferentes salas de control, para así conocer como está funcionando la planta. PROFIBUS cumple con los requisitos necesarios para realizar este tipo de comunicaciones, ofreciendo distintas interfaces de usuario tanto para comunicaciones rápidas con dispositivos de campo, por ejemplo estaciones periféricas o descentralizadas o accionamientos, como para un amplio intercambio de dato entre equipos maestros.

A continuación veremos cómo crear una red PROFIBUS con un PLC S7-1200.

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

Lo primero que debemos hacer es crear un nuevo proyecto. Agregando un PLC sin especificar para que el software determine el hardware automáticamente. Como se observa en la Figura 118.

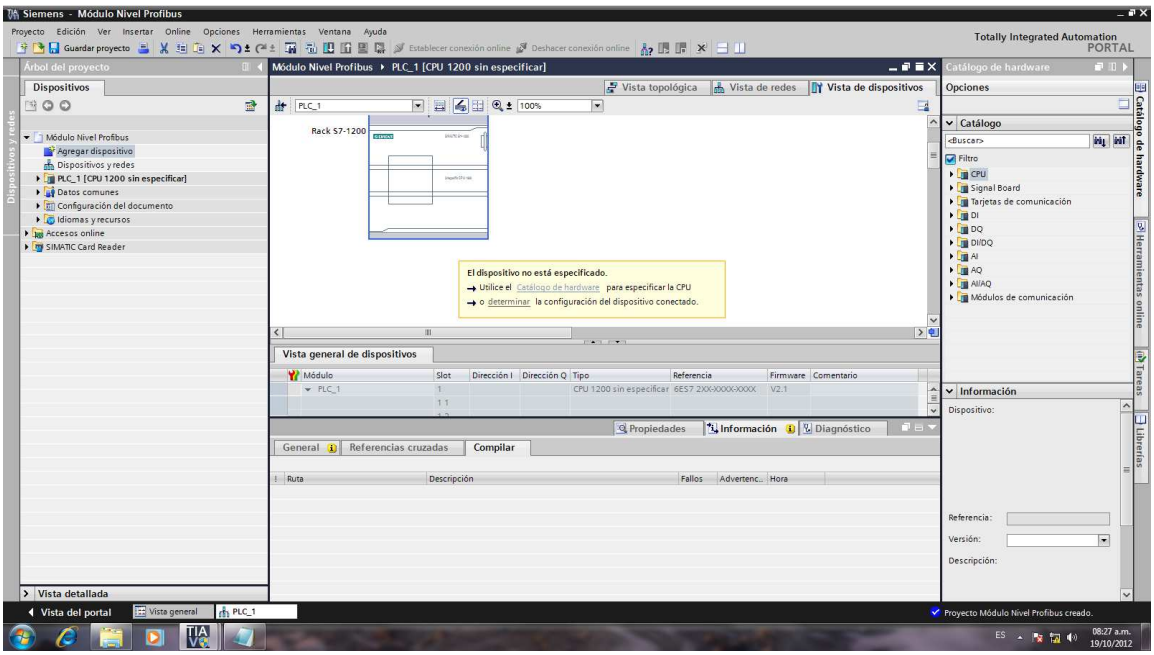

Figura 118. Determinar dispositivo

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

Luego de determinar el dispositivo, podemos observar sus características dando clic sobre cada módulo del hardware. Para este caso tenemos un módulo Profibus, un módulo de 2 salidas analógicas. Es importante tener presente las direcciones de entradas y salidas analógicas para facilitar la programación de nuestras aplicaciones.

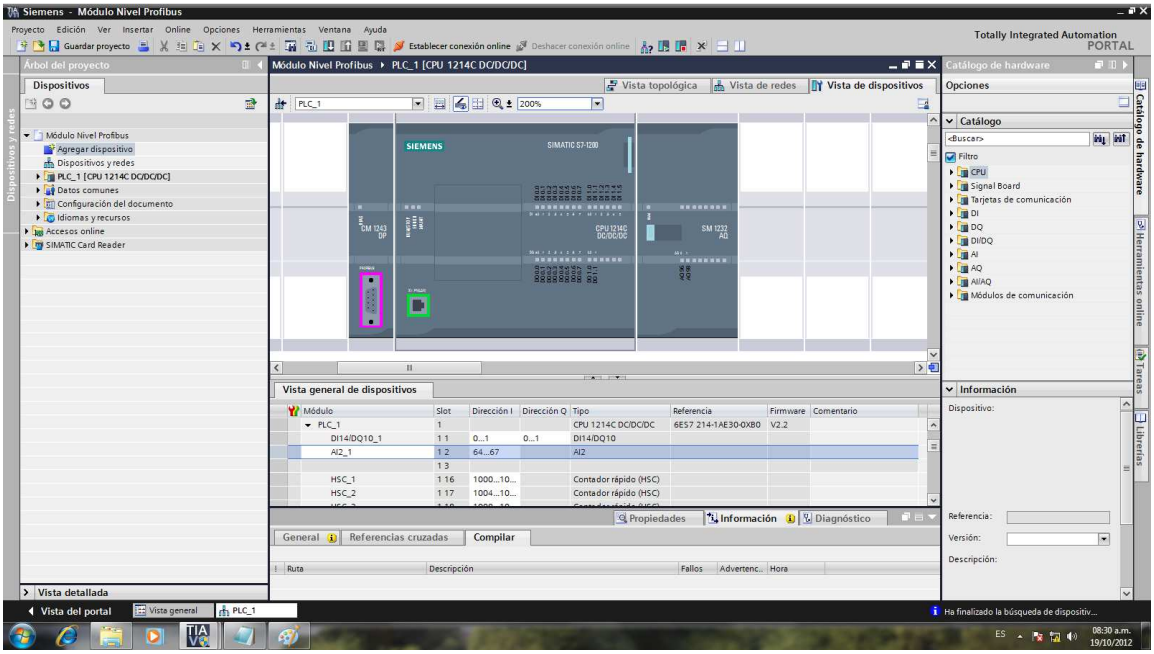

Figura 119. PLC determinado

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

Para este caso se puede notar que las direcciones de las salidas analógicas del módulo que contiene 2 de ellas tiene la dirección desde el byte 96 hasta el 99. Como una salida analógica es tipo entero (INT) y utiliza 2 bytes, entonces la dirección de ambas salidas es QW96 (que utiliza los bytes 96 y 97) y QW99 (que utiliza los bytes 98 y 99)

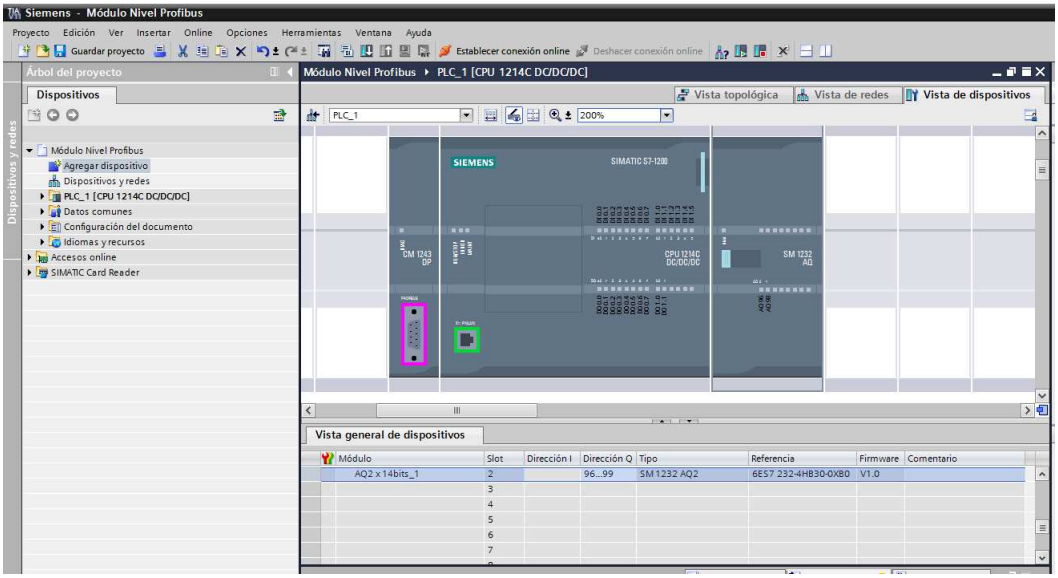

Figura 120. Salidas analógicas

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

Hacemos clic en el módulo PROFIBUS y agregamos una subred y la respectiva dirección del dispositivo, en este caso del PLC. En este tipo de comunicación las direcciones de cada dispositivo se configuran directamente en el dispositivo, no se hace desde el software, es importante saber cuál dirección ha sido asignada a cada uno de los elementos que van a conformar la red, para una correcta configuración. Para nuestro caso el PLC tiene la dirección 2.

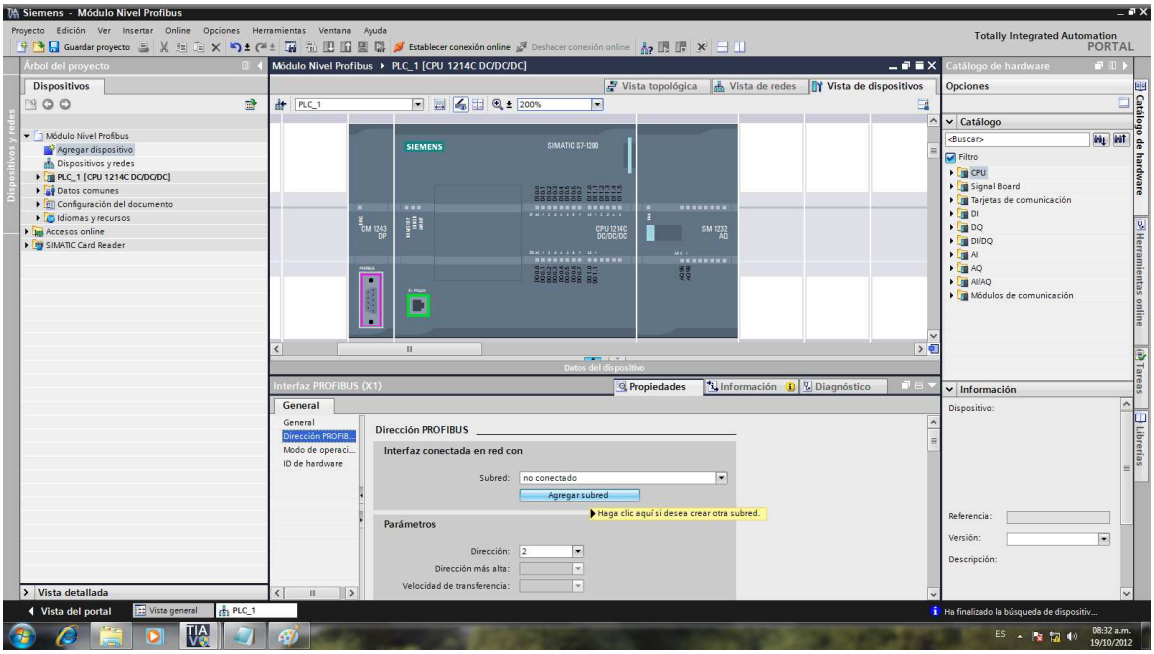

Figura 121. Agregar subred y dirección del PLC

Ahora podemos ir a "*Vista de redes*", donde se puede ver todos los dispositivos que se han agregado hasta el momento y la respectiva conexión al bus de comunicación de la red previamente creada.

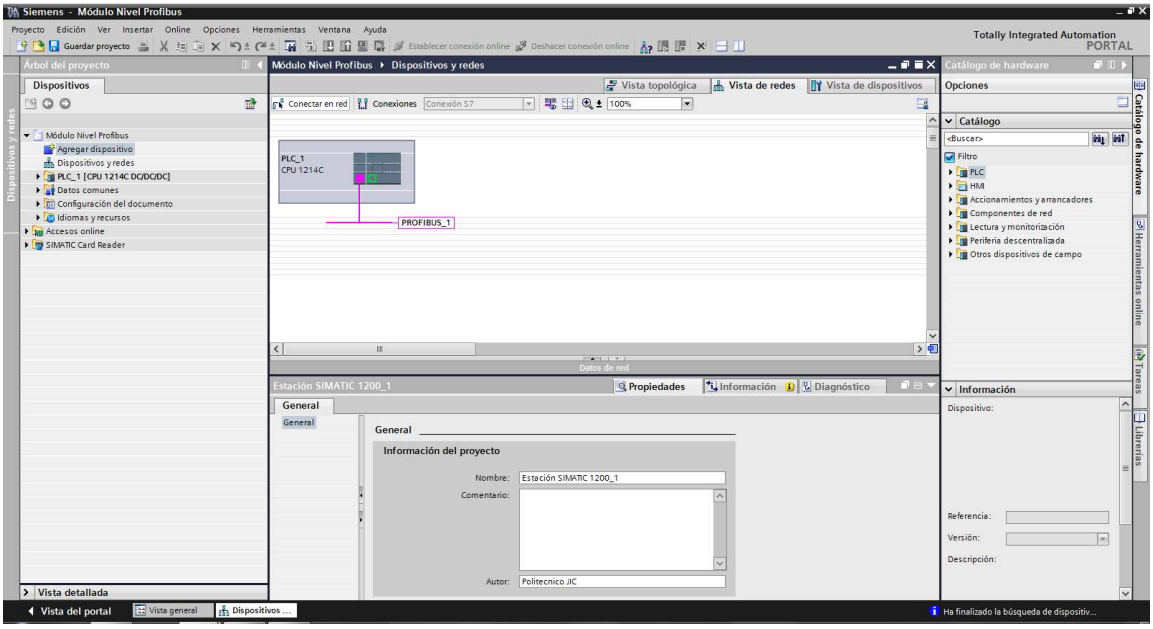

Figura 122. Vista de redes

Damos clic en la línea de red (color rosa) para configurar la velocidad de transferencia y el perfil PROFIBUS correspondiente. En este caso el perfil es DP, **Decentralised Periphery**. Esta versión de PROFIBUS está diseñada básicamente para la comunicación entre los sistemas automáticos de control (autómata programable, robot, sistema de control numérico, etc.) y los equipos periféricos distribuidos a nivel de campo (Decentralised Periphery), compuesto por los dispositivos sensores y actuadores. Dejar la velocidad de transferencia en el valor por defecto, como se muestra en la Figura 123.

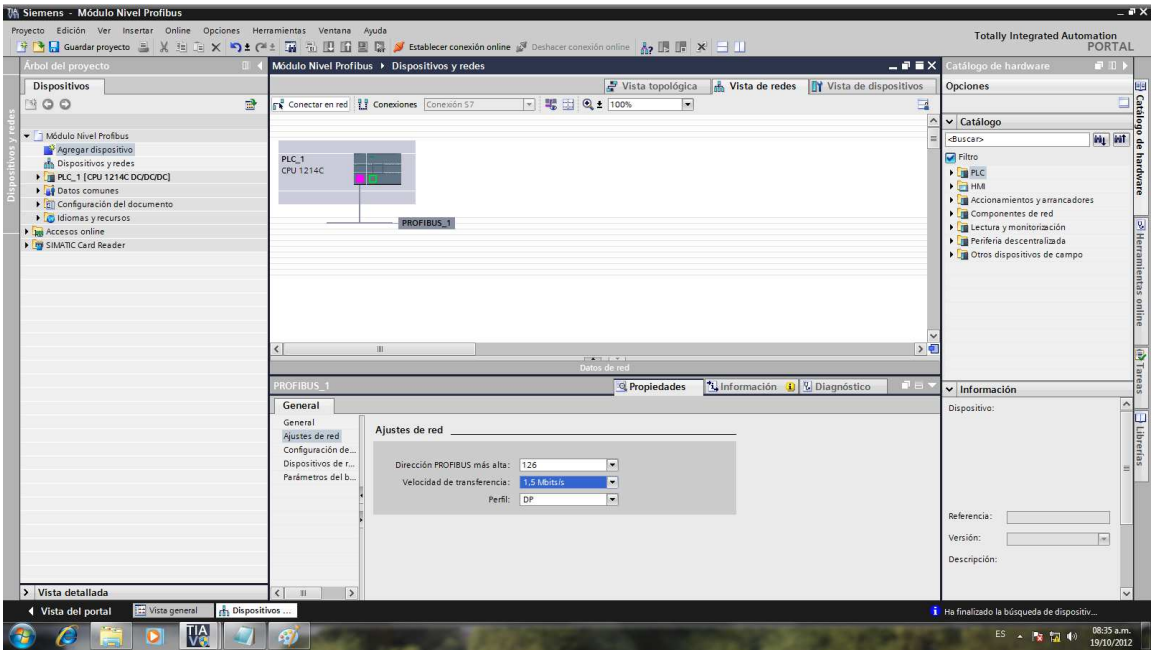

Figura 123. Velocidad de transferencia y perfil DP

Para proceder a agregar los demás dispositivos que harán parte de la red y que conforman la estructura física del proceso, es necesario instalar el correspondiente archivo GSD de cada dispositivo. Un archivo GSD es un archivo de texto que contiene las características y las opciones de configuración del dispositivo al que representan. Estos archivos son importados por el Configurador y posteriormente agregados para el dispositivo maestro.

Los archivos GSD son frecuentemente suministrados con los dispositivos por el fabricante, pero pueden descargarse de Internet de la siguiente página: http://www.profibus.com

Los archivos GSD son estructurados por el fabricante individualmente para cada tipo de dispositivo, de acuerdo con un formato fijo. Algunos parámetros son obligatorios, incorporan valores por defecto y otros son opcionales. Estos archivos están divididos en las siguientes partes: Especificaciones generales, como nombre del fabricante, referencia del dispositivo, versión hardware y software, tipo de estación, nº de identificación, protocolo y velocidades soportadas.

Vamos a instalar el archivo GSD de un transmisor de flujo de SIEMENS con referencia SITRANS F M MAGFLO - MAG6000_7ME6920-01AA30-1AA0, cuyo archivo GSD se encuentra con el siguiente nombre **si060649.gsd.**

En la barra de herramientas vamos a "*Opciones*" y de la lista desplegada seleccionamos "*Instalar archivo de descripción del dispositivo*"

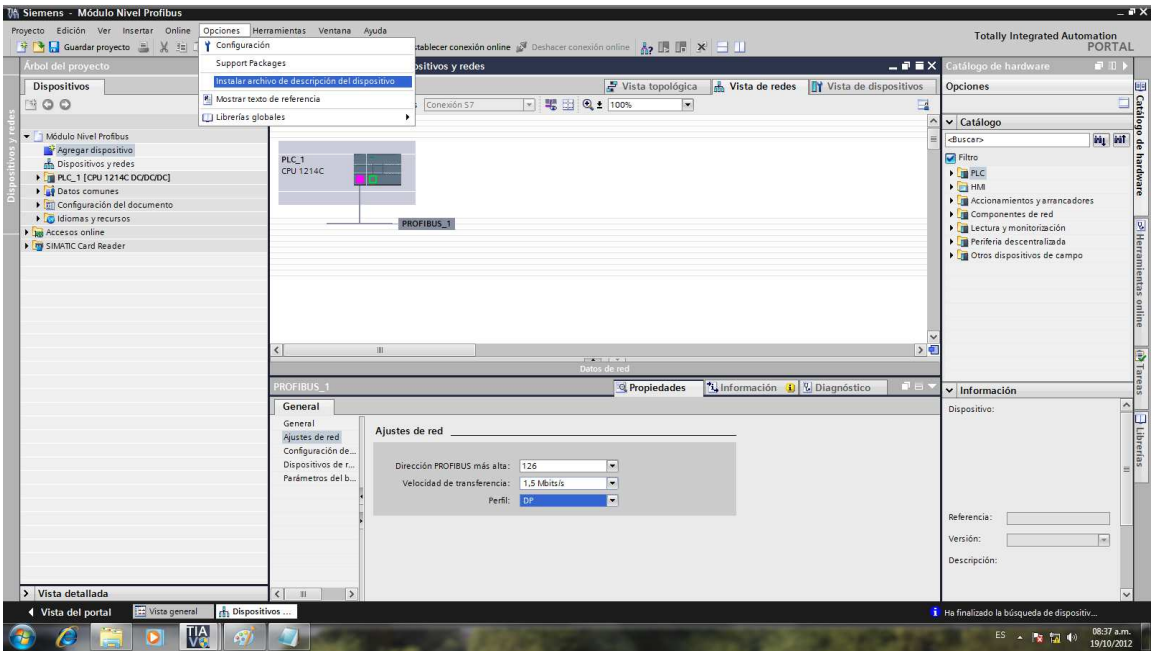

Figura 124. Instalar archivo GSD

En el campo ruta de origen nos ubicamos en la carpeta que contiene el archivo GSD que se va a instalar y seleccionamos el archivo respectivo que en este caso para el transmisor de flujo es **si060649.gsd**, posteriormente damos clic en el botón "*Instalar".* Es necesario tener el software TIA abierto sólo una vez, de no ser así, el programa no permite realizar la instalación.

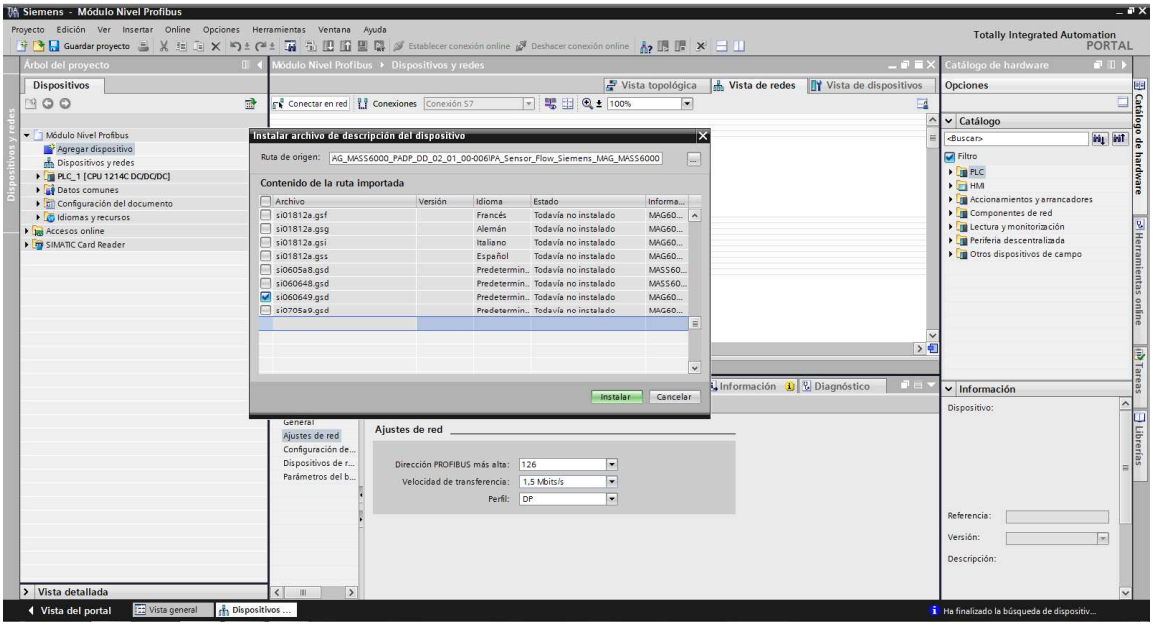

Figura 125. Seleccionar archivo GSD

Una vez instalado el archivo GSD podemos ver en la ventana actual, en la columna "*Estado*" indicando que el archivo ya está instalado. Procedemos luego a cerrar esta ventana.

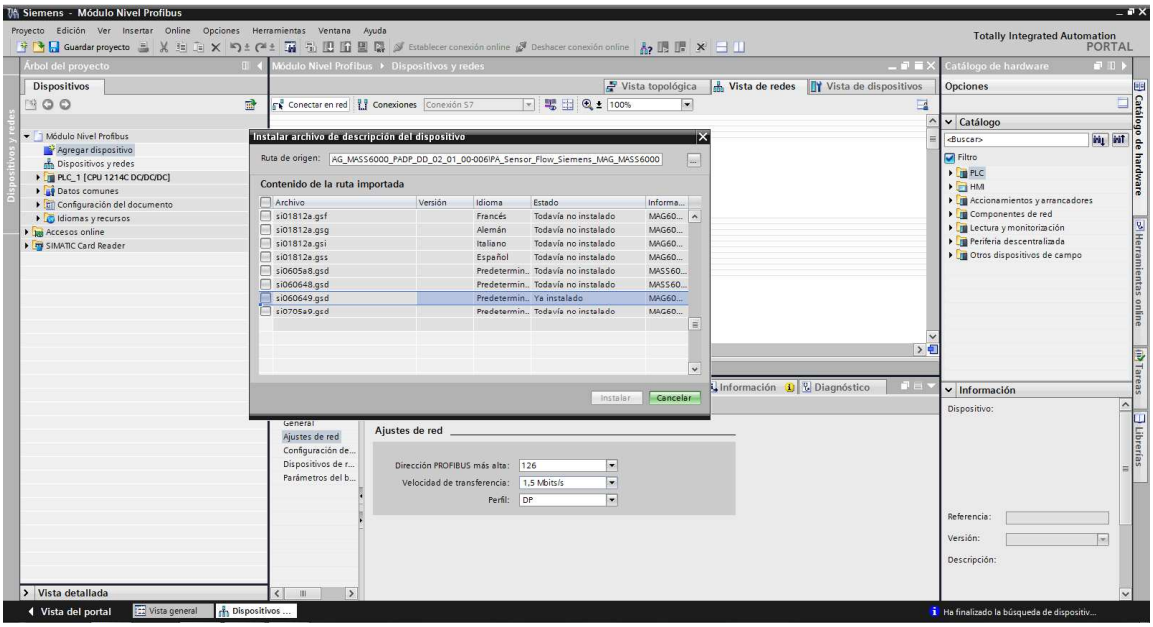

Figura 126. Archivo GSD instalado

Ahora en las librerías de dispositivos de campo podemos encontrar el respectivo transmisor de flujo que hemos instalado. Nos ubicamos en el catálogo ubicado en la parte lateral derecha de nuestro proyecto. El dispositivo se puede encontrar en la siguiente ruta: "*Otros dispositivos de campo \ PROFIBUS DP \ Perfil PROFIBUS PA \ Siemens AG \ MAG6000 PA Extended*"

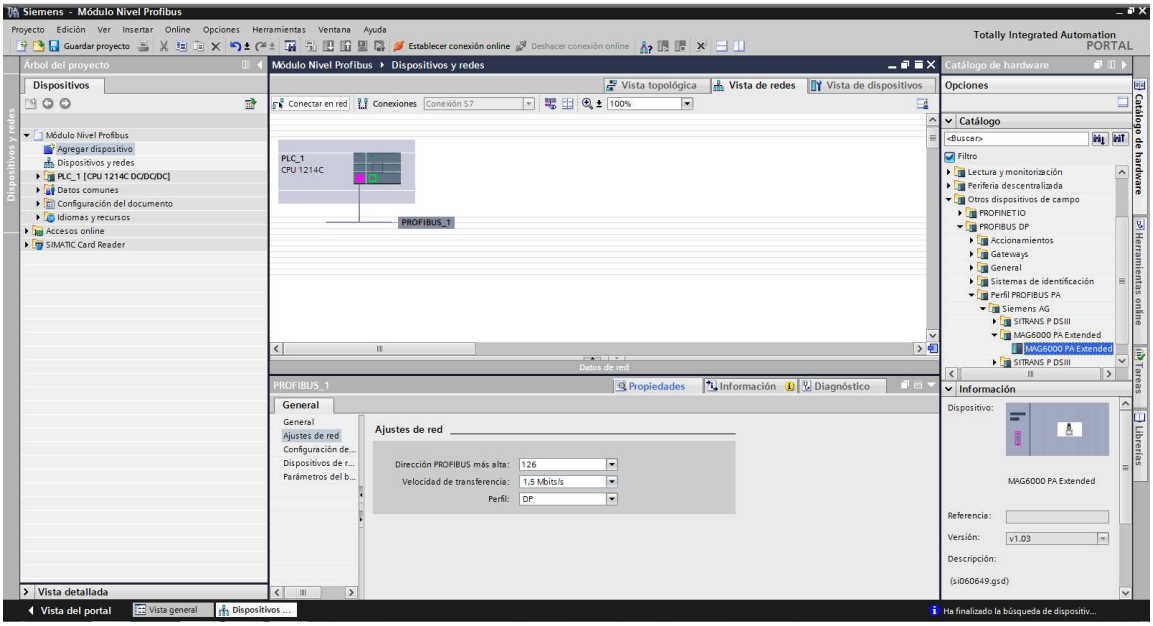

Figura 127. MAG6000 PA Extended

Arrastramos el dispositivo hasta la vista de redes.

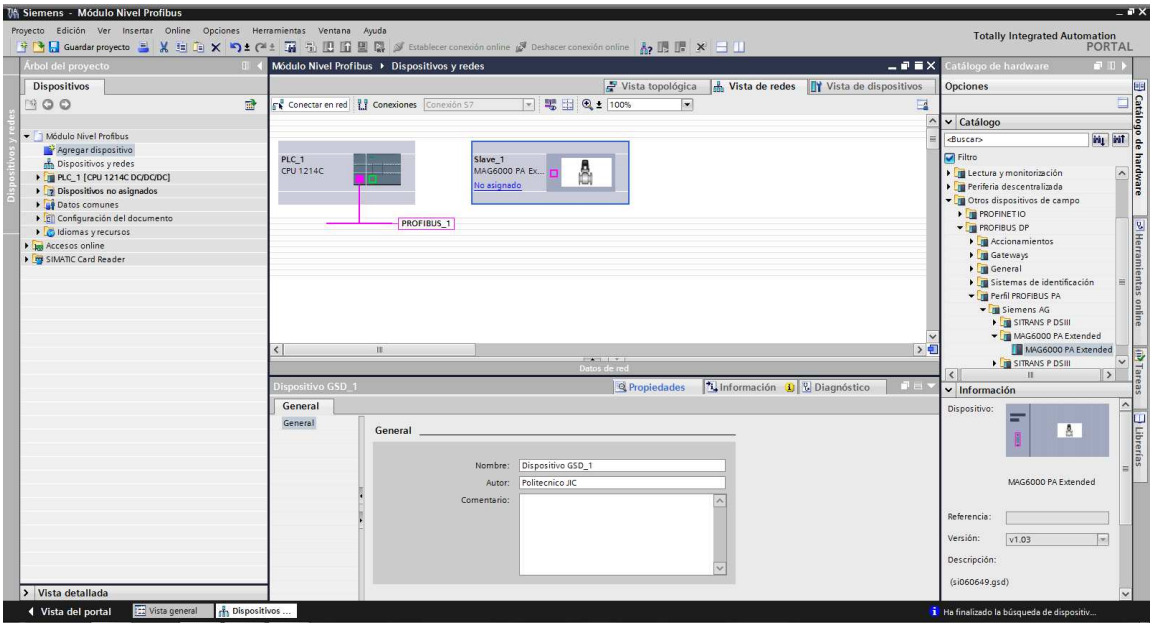

Figura 128. Agregar dispositivo a la red

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Tomar el conector color rosa del transmisor y se conecta a la red, dando clic sostenido y arrastrándolo hasta el bus PROFIBUS.

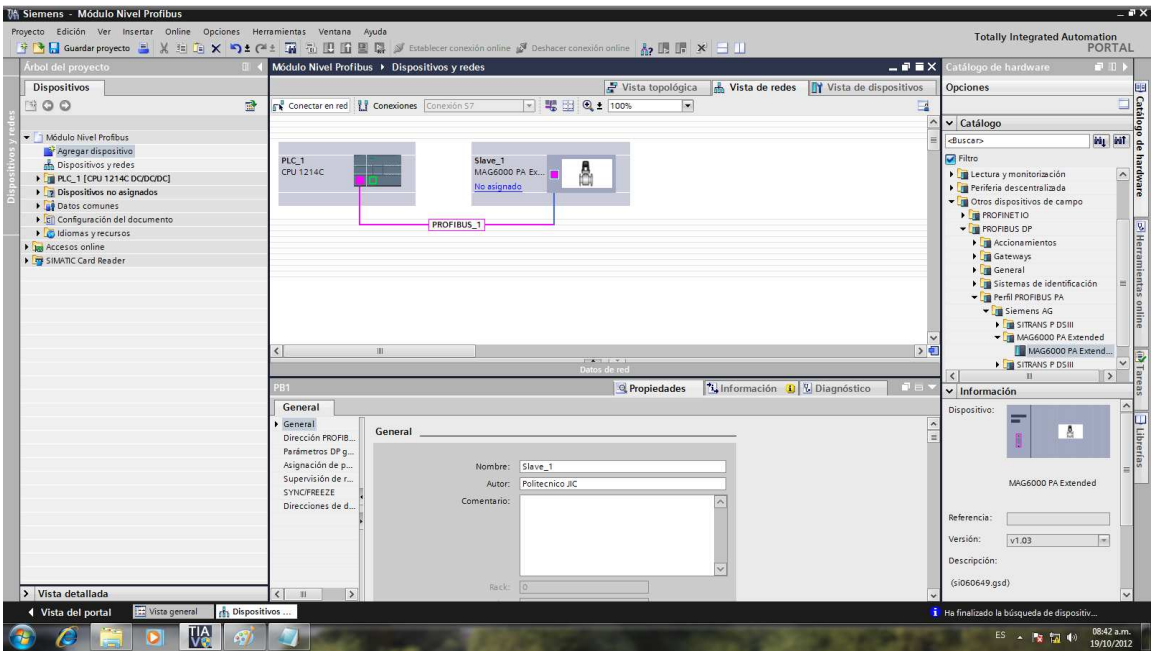

Figura 129. Conectar dispositivo a la red

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

En el bloque del transmisor se observa un link que dice “No asignado”, dando clic ahí se selecciona el dispositivo maestro, que para nuestro caso es el PLC.

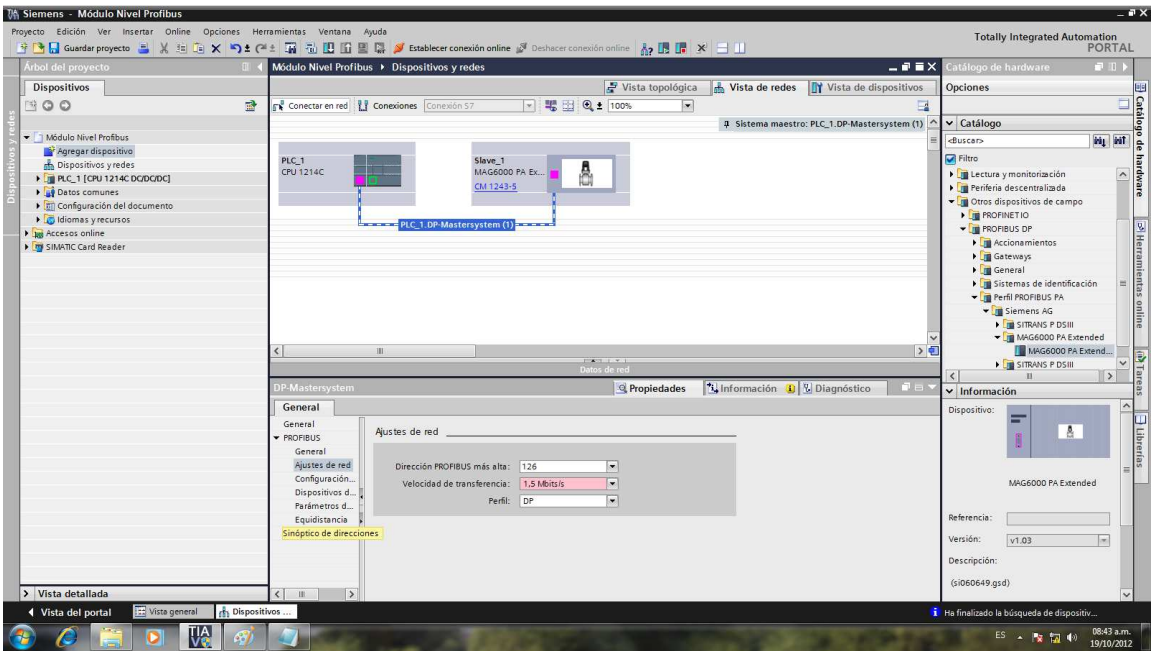

Figura 130. Seleccionar dispositivo maestro

Ahora en el espacio que antes decía No asignado aparece una referencia que corresponde al módulo PROFIBUS instalado físicamente y la línea color rosa ha cambiado su apariencia, indicando cual es el dispositivo maestro.

Si damos doble clic sobre el transmisor podemos cambiar diferentes opciones. En este caso cambiaremos el nombre para identificar el dispositivo más fácilmente. Lo llamaremos FIT_ Transmisor Indicador de Flujo.

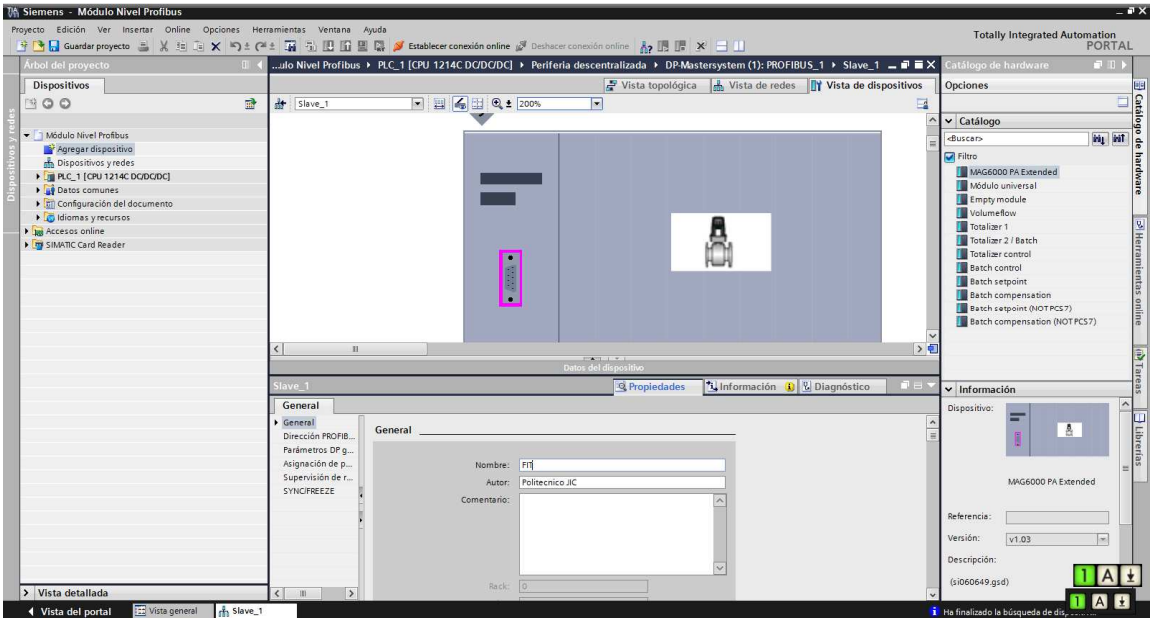

Figura 131. Cambiar nombre al dispositivo

Volver a la vista de redes.

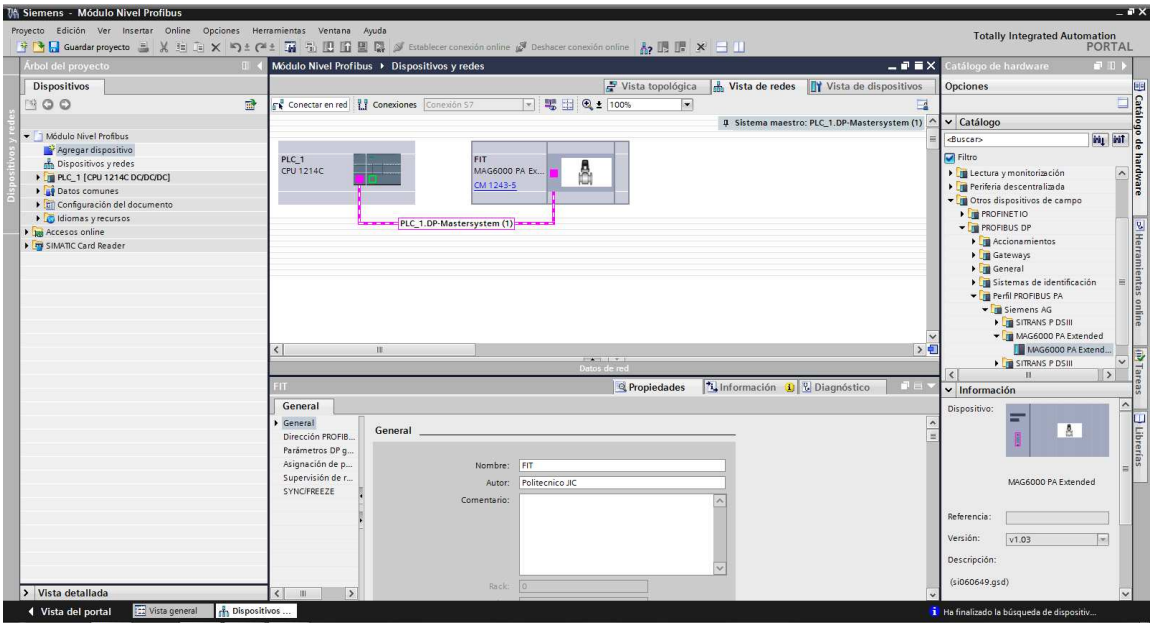

Figura 132. Vista de redes

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

Seleccionar el transmisor y asignar la correspondiente dirección PROFIBUS del dispositivo, que pare este caso es la 4.

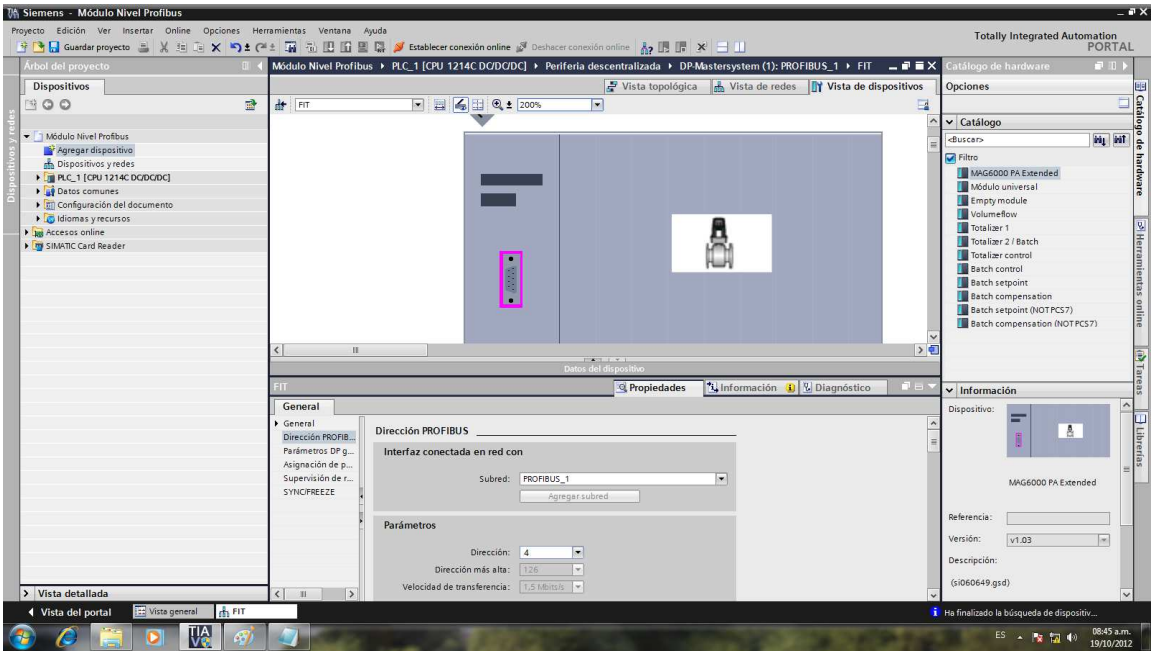

Figura 133. Configurar dirección del FIT

En la parte superior de la ventana se puede activar la opción "*Mostrar direcciones*", con lo que se puede visualizar la dirección de cada uno de los dispositivos en la red.

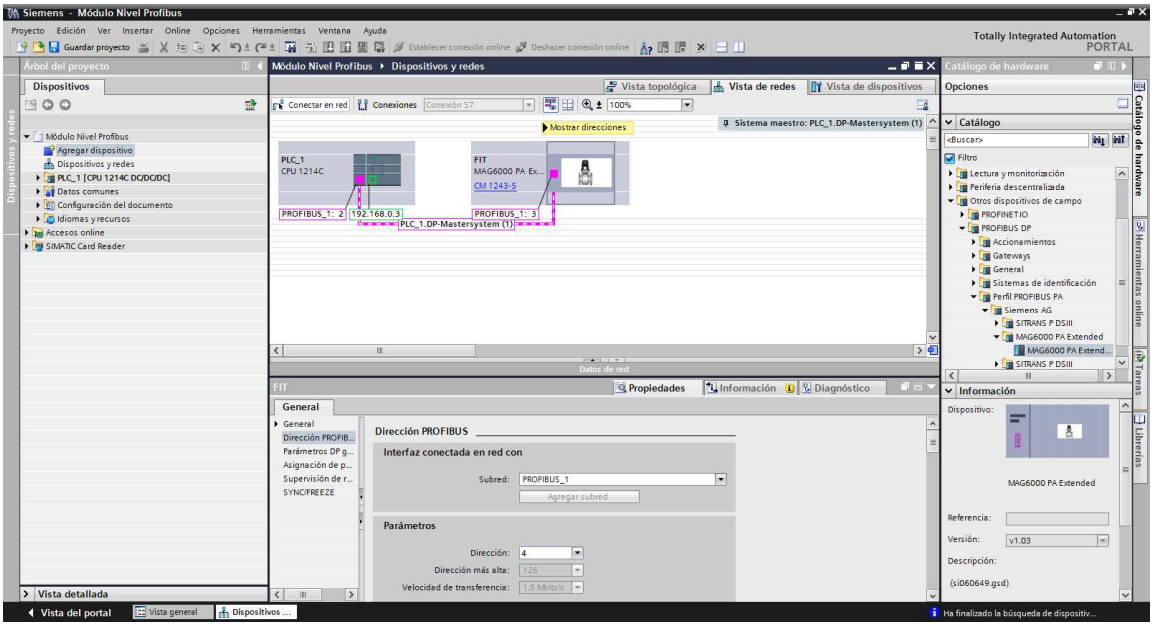

Figura 134. Mostrar direcciones

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

De la misma forma se agregan todos los dispositivos esclavos que son necesarios para nuestra aplicación. En nuestro caso un transmisión de presión y una válvula como se muestra en la Figura 135.

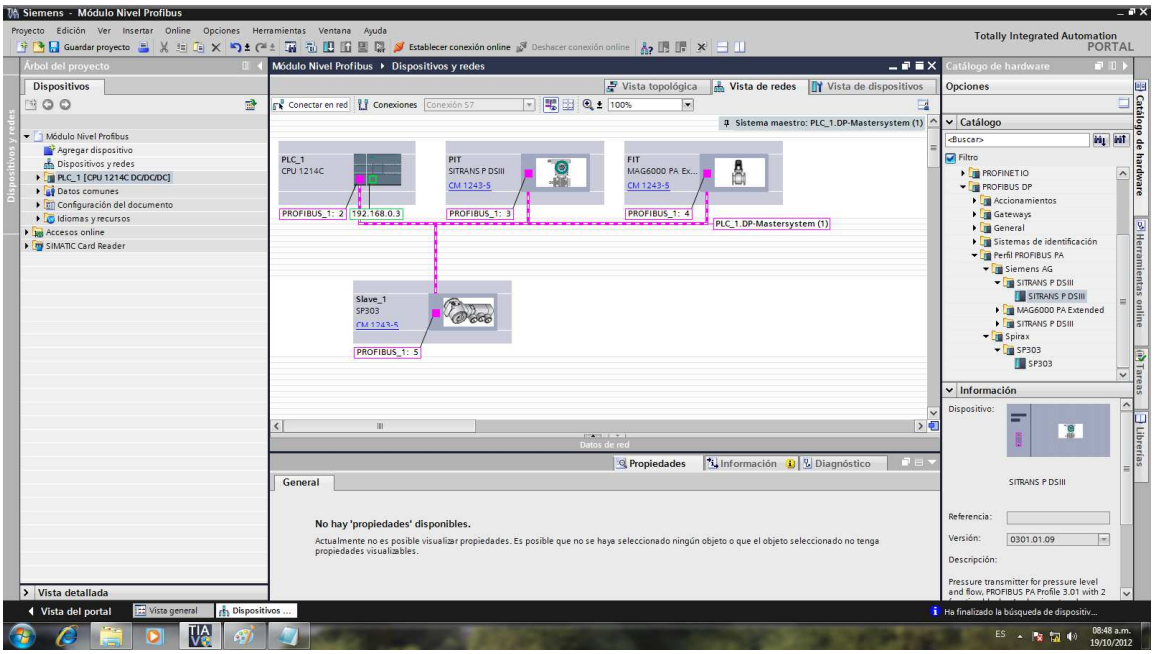

Figura 135. Agregar otros dispositivos

Con este protocolo de comunicación se le asignan ciertas entradas o salidas a los dispositivos de campo, direcciones en las cuales la señal medida llega en formato real, es decir que los datos ocupan 4 bytes, para las cuales las direcciones en el PLC son tipo ID ó QD si son entradas o salidas respectivamente. Como vemos en la hay un byte de más en las direcciones del transmisor de presión, este byte da un diagnostico del sensor y se utiliza para monitorear el estado del mismo.

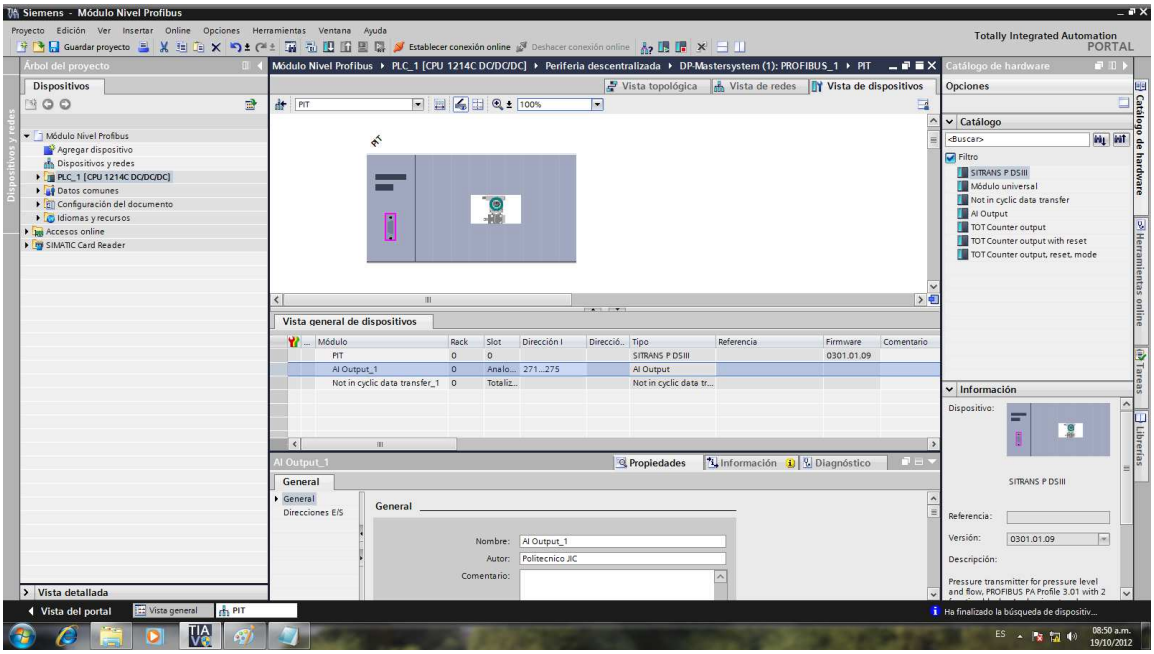

Figura 136. Direcciones entrada analógica del PIT

Como se observa la dirección ID 271 entregará la señal de presión ya escalizada y la dirección IB275 que es el bit de diagnostico del sensor. Así mismo se puede ver para cada dispositivo.

Para esta válvula hay que añadir el archivo correspondiente al posicionador como se muestra en la Seleccionando la válvula podemos ver sus direcciones de salida. Para nuestro caso la QD64 nos indica el porcentaje de posición de apertura y/o cierre de la válvula y la dirección QB68 hace referencia al control binario de la válvula (poniendo todos los bits en alto se pueden realizar acciones de apertura o cierre en la válvula).

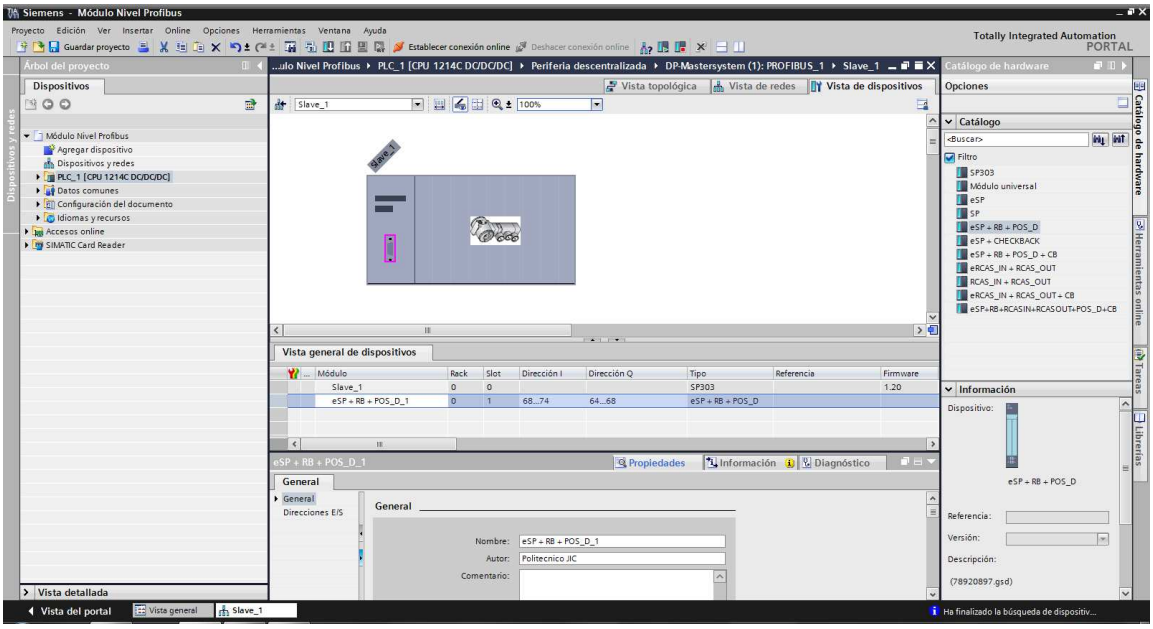

Figura 137. Direcciones de salida de la válvula

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

Compilamos el proyecto.

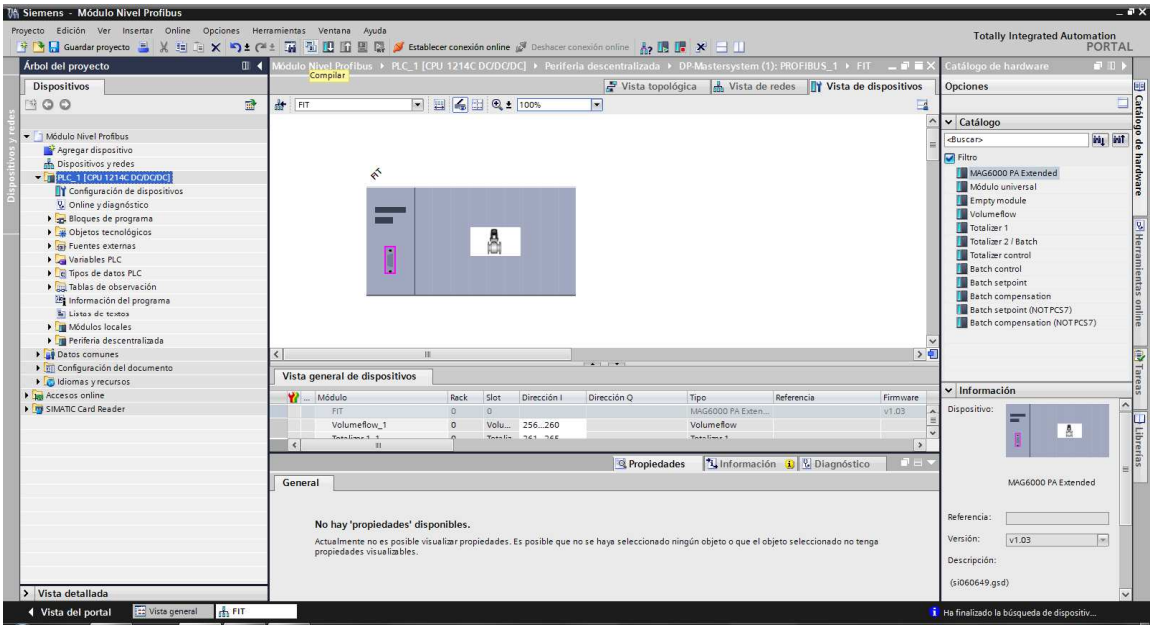

Figura 138. Compilar proyecto

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

Al compilar se observa un error que indica que algunos módulos no soportan la velocidad de transmisión que se escogió (1,5Mbit/s). Esto se debe a que al tener una red la velocidad de transferencia del bus debe ser soportada por el dispositivo más lento.

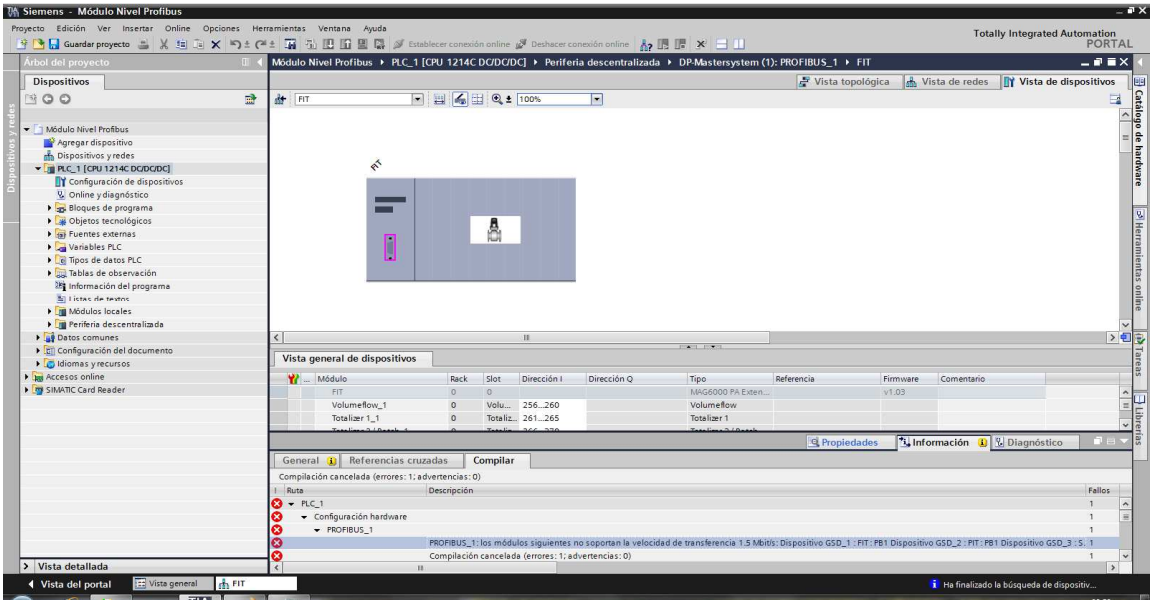

Figura 139. Error en la velocidad de transferencia

En vista de redes seleccionamos el bus de comunicación (línea rosa) y en ajustes de red cambiamos la velocidad de transferencia como se indica en la Figura 140.

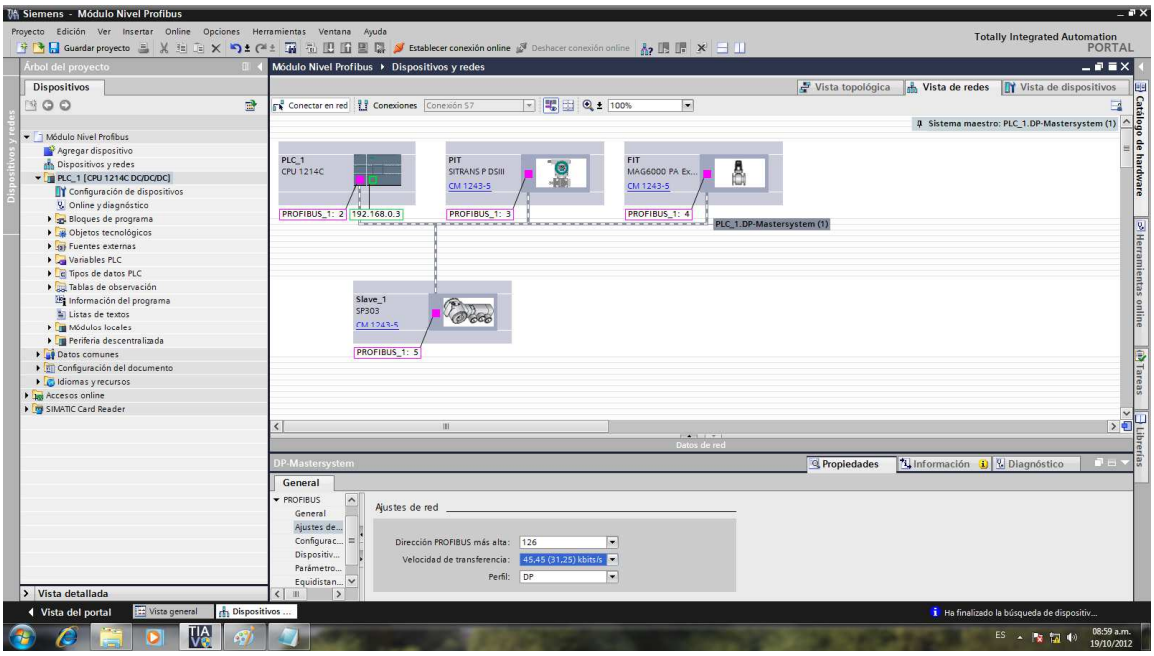

Figura 140. Cambiar velocidad de transferencia

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

Volvemos a compilar y observamos que ya no sale ningún error en el proyecto.

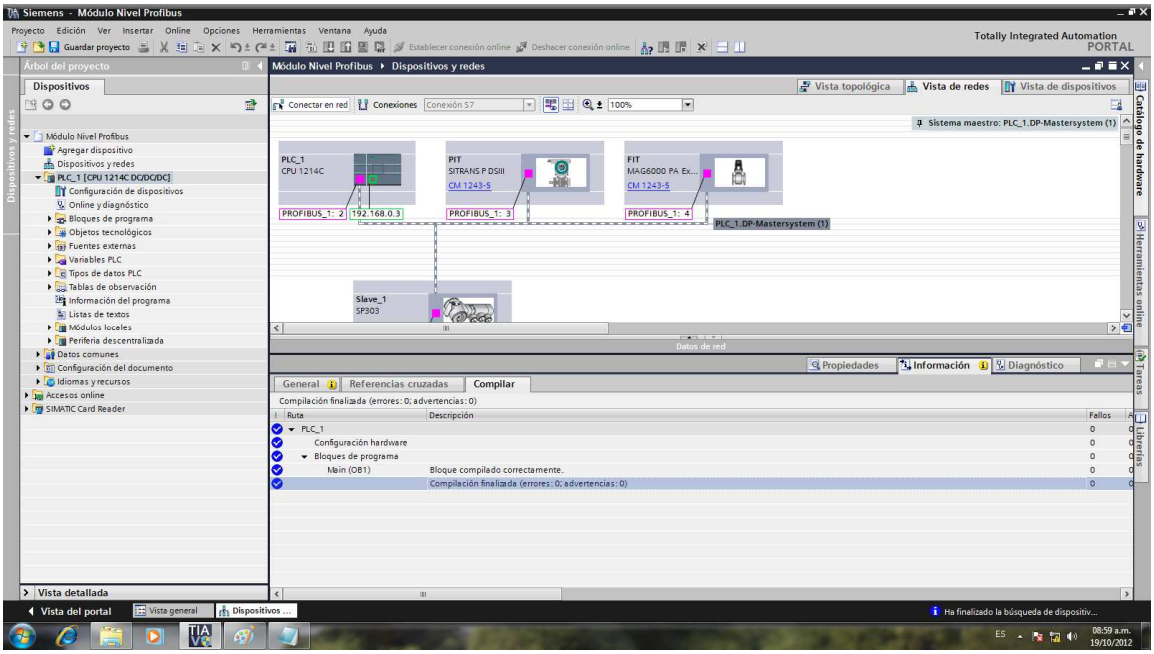

Figura 141. Compilación exitosa

Teniendo todo bien configurado y sin errores, podemos cargar la configuración al PLC.

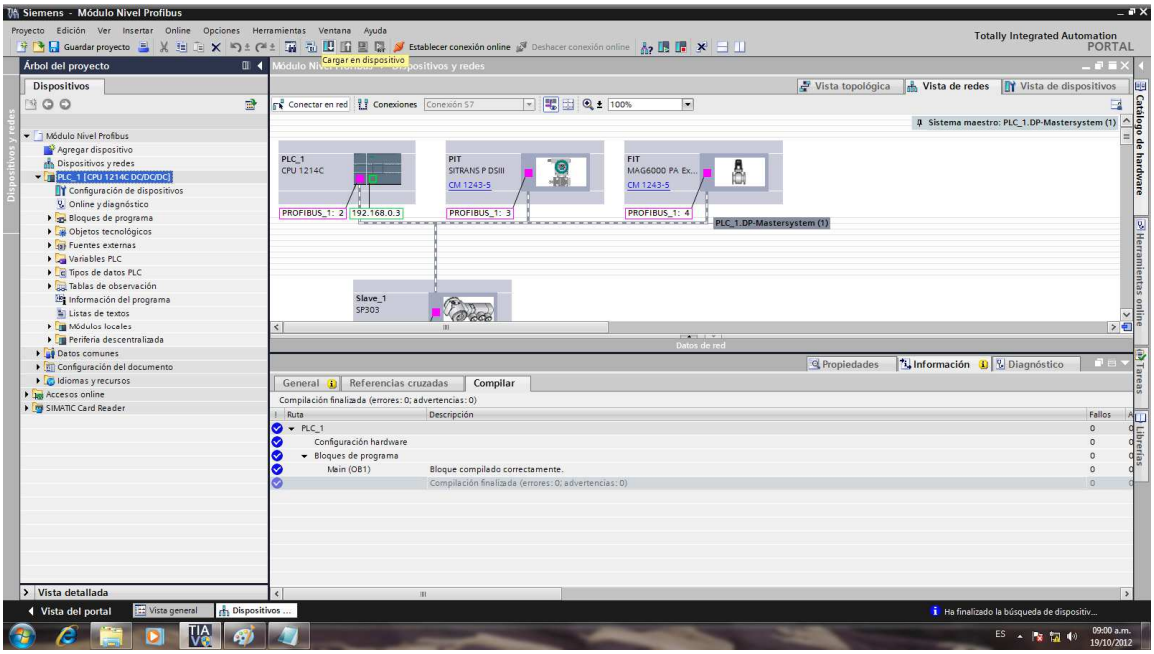

Figura 142. Cargar configuración al PLC

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Rubén Darío Vásquez Salazar
Docente de tiempo completo
Área de Instrumentación y Control

BIBLIOGRAFÍA

[1] Vásquez, R. (2010). Controladores Lógicos Programables. ISBN 978-958-8351-95-7. Fondo editorial ITM.

[2] Siemens AG, (2009). Manual de sistema – Controlador Lógico Programable S7-1200. A5E02486683-02. Noviembre de 2009.

[3] Siemens AG, (2012). Sitio web de soporte de Siemens. http://support.automation.siemens.com. Fecha de última consulta: 23 de octubre de 2012.

[4] Vásquez, R. (2012). Tutorial TIA Portal Parte 1.