

## ANEXO A. Programación en el PLC.

Este anexo contiene la programación realizada en el PLC, así como un detalle de las variables utilizadas en el programa. Se corresponde con el archivo entregado llamado **“Control de línea de producción.stx”**.

### A.1. Programación de PRL.

```
(*Adquisición de señales de mando procedentes del PC*)
SR0:(*Ejecuto el subprograma de adquisición de entradas*)

(*DETECCIÓN DE LOS POSIBLES FALLOS DE LA INSTALACIÓN AUTOMATIZADA.*)

(*FALLOS ENTRADAS/SALIDAS*)
F_tarjetas:=%I0.MOD.ERR OR %I1.MOD.ERR OR %I2.MOD.ERR OR %I3.MOD.ERR OR %I4.MOD.ERR OR NOT %S10;

(*FALLOS MAGNETOTÉRMICOS*)
(*Detección de fallos Magnetotérmicos motores. Como en la maqueta no tenemos,
los simulamos con bits de memoria*)
F_magneto:=Magcinta1 OR Magcinta2 OR Magcinta3 OR Magcinta4 OR Magmaq1 OR Magmaq2 OR Magselladelante
OR Magsellatras OR Magsel2adelante OR Magsel2atras;

(*FALLOS POR TIEMPOS EXCESIVOS DE CINTAS O SELECTORES*)
F_tiempos_sel1:=Tsegsellre.Q OR Tsegsellav.Q;
F_tiempos_sel2:=Tsegsel2re.Q OR Tsegsel2av.Q;
F_tiempos:=(Tsegc1.Q OR Tsegc2.Q OR Tsegc3.Q OR Tsegc4.Q OR F_tiempos_sel1 OR F_tiempos_sel2);

(*FALLOS POR SITUACIONES ILÓGICAS DE CAPTADORES*)
F_sitilogicas:=(%X22 AND Isensormaq1)OR(%X31 AND Isensormaq2)OR((Modo_automatgico OR Modo_finciclo)
AND(%X10 OR %X11 OR %X12)AND NOT Isellini)OR((Modo_automatgico OR Modo_finciclo)AND(%X40 OR %X41)AND
NOT Isel2ini)OR Modo_posicionamiento AND(NOT Isensorc1 OR NOT Isensormaq1 OR NOT Isensormaq2 OR NOT Isensorc4);

(*FALLOS POR PERDIDA DE COMUNICACIÓN ENTRE EL PLC Y EL SCADA*)
F_comunicacion:=%X10 AND(Modo_automatgico OR Modo_finciclo)AND(Señal_mando1<=250 OR Señal_mando2<=250);
(*Antes de recibir el valor con los tiempos compruebo que hay comunicación con el SCADA*)

(*FALLOS POR REARRANQUE EN CALIENTE. Si Se pasa el automata a STOP durante ciclos automaticos lo considerare
dentro de esta casuistica*)
IF %S1 OR %S13 AND(Modo_posicionamiento OR Modo_automatgico OR Modo_finciclo)THEN
SET F_rearraque_caliente;
END_IF;

(*CONJUNTO DE SITUACIONES (FALLOS y SETA) QUE GENERAN LA PARADA DE EMERGENCIA*)
Situacion_emergencia:=F_tarjetas OR F_magneto OR F_tiempos OR F_sitilogicas OR Seta_emergencia OR %S0 OR
F_rearraque_caliente OR F_comunicacion OR Error_foto;
```

Figura A.1. Programación de PRL primera parte.

```
(*VISUALIZACIÓN POR PANTALLA DEL MOTIVO DE LA PARADA. SI HAY VARIOS MOTIVOS, SE MUESTRA LA MÁS IMPORTANTE*)
(*Utilizo una señal de control (%QW0.10) para indicar al SCADA la alarma producida*)
(*%MW0: Establezco una prioridad para cada tipo de alarma para evitar que alguna alarma menos
importante enmascare una importante*)
IF Situacion_emergencia THEN
  RESET Rearme_memoria; (*Siempre que se de una situación de emergencia se resetea la marca de rearmado*)
  IF %S0 THEN (*Arranque Frio*)
    Señal_control:=1500;
    %MW0:=10;
  ELSIF F_rearraque_caliente THEN (*Arranque Caliente*)
    Señal_control:=2000;
    %MW0:=10;
  RESET F_rearraque_caliente;
  ELSIF F_tarjetas AND %MW0<9 THEN (*Fallo Tarjetas*)
    Señal_control:=2500;
    %MW0:=9;
  ELSIF F_comunicacion AND %MW0<8 THEN (*Fallo Comunicación*)
    Señal_control:=3000;
    %MW0:=8;
  ELSIF Error_foto AND %MW0<8 THEN (*Error SCADA*)
    Señal_control:=8000;
    %MW0:=8;
  ELSIF Seta_emergencia AND %MW0<7 THEN (*Seta Emergencia*)
    Señal_control:=3500;
    %MW0:=7;
  ELSIF F_magneto AND %MW0<6 THEN (*Magnetotérmico*)
    Señal_control:=4000;
    %MW0:=6;
  ELSIF F_sitilogicas AND %MW0<5 THEN (*Error Captadores*)
    Señal_control:=4500;
    %MW0:=5;
  ELSIF Tsegc1.Q AND %MW0<=4 THEN (*Texcesivo Cinta1*)
    Señal_control:=5000;
    %MW0:=4;
  ELSIF Tsegc2.Q AND %MW0<=4 THEN (*Texcesivo Cinta2*)
    Señal_control:=5500;
    %MW0:=4;
  ELSIF Tsegc3.Q AND %MW0<=4 THEN (*Texcesivo Cinta3*)
    Señal_control:=6000;
    %MW0:=4;
  ELSIF Tsegc4.Q AND %MW0<=4 THEN (*Texcesivo Cinta4*)
    Señal_control:=6500;
    %MW0:=4;
  ELSIF F_tiempos_sel1 AND %MW0<=4 THEN (*Texcesivo Selector1*)
    Señal_control:=7000;
    %MW0:=4;
  ELSIF F_tiempos_sel2 AND %MW0<=4 THEN (*Texcesivo Selector2*)
    Señal_control:=7500;
    %MW0:=4;
  END_IF;
  ELSIF NOT Memoria_emergencia THEN (*Indico al sistema que esta todo ok*)
    Señal_control:=500;
  END_IF;
```

Figura A.2. Programación de PRL segunda parte.

```
(*ACTIVACIÓN DE LA PARADA DE EMERGENCIA*)

IF(Situacion_emergencia AND NOT Memoria_emergencia)THEN
  SET Memoria_emergencia;(*Se almacena un bit para indicar que ha habido una parada de emergencia*)
  IF(Modo_posicionamiento OR Modo_automatico OR Modo_finciclo)THEN
    SET %S23;
  (*Se congelan los grafcet si se detecta que se ha producido algun motivo de parada en estado automatico*)
  END_IF;
END_IF;

(*GESTIÓN DEL REARME*)

(*Cuando se haya producido una emergencia y pulse rearme*)
IF Memoria_emergencia AND Rearme AND NOT Rearmando THEN
  START Trearme;(*Arranca el temporizador que detrermina la forma de rearmar*)
  SET Rearmando;
  END_IF;

(*Forma de rearmar*)
IF(Rearmando AND FE Rearme)THEN
  IF NOT %S23 OR Trearme.Q THEN
  (*Rearmo el GRAFCET a posición inicial si no estoy en estados automáticos o finaliza Trearme*)
    RESET Memoria_emergencia;
    RESET %S23;(*Descongelo el GRAFCET*)
    SET %S21;(*Inicializo el GRAFCET*)
  ELSIF NOT Situacion_emergencia THEN
  (*El sistema se rearma y el GRAFCET prosigue donde se quedó si se ha solucionado la emergencia*)
    RESET Memoria_emergencia;
    SET Rearme_memoria; (*Guardo una marca indicando que se ha rearmado el sistema*)
  END_IF;
  RESET Rearmando;(*Reseteo esta marca para poder usarla en el siguiente rearme*)
  %MWO:=0;
  DOWN Trearme;
  END_IF;

(*Para continuar se deberá de pulsar Marcha*)
IF Rearme_memoria AND Marcha THEN
  RESET %S23;(*Se descongela el GRAFCET*)
  RESET Rearme_memoria;
  END_IF;
```

Figura A.3. Programación PRL tercera parte.

## A.2. Programación de SR0.

```
(*TRANSFORMACIÓN DE ENTRADAS INTERNAS EN ENTRADAS EXTERNAS PARA TRATAMIENTO HOMOGENEO*)

(*Doy significado a las consignas recibidas como salidas digitales desde
el SCADA a entradas Analógicas en el PLC*)
IF Auto_manual_a>2000 THEN (*Automatico_Manual*)
SET Automatico;(*Automático*)
ELSE SET Manual;(*Manual*)
END_IF;
IF Marcha_a>2000 THEN (*Marcha*)
SET Marcha;
END_IF;
IF Rearme_a>2000 THEN (*Rearme*)
SET Rearme;
END_IF;
IF Foto_ok_a>2000 THEN (*Patron detectado*)
SET Foto_ok;
END_IF;
IF Error_foto_a>2000 THEN (*Error al tomar fotografía*)
SET Error_foto;
END_IF;

(*En caso de estar en modo Manual, compruebo si se pulsa algún mando manual*)
IF Modo_manual AND Señal_mando1>2750 AND Señal_mando1<=7750 THEN
IF Señal_mando1<3250 THEN (*Manual_C1*)
SET Manual_c1;
ELSIF Señal_mando1<3750 THEN (*Manual_sel1_ad*)
SET Manual_sel1_ad;
ELSIF Señal_mando1<4250 THEN (*Manual_C2*)
SET Manual_c2;
ELSIF Señal_mando1<4750 THEN (*Manual_sel2_ad*)
SET Manual_sel2_ad;
ELSIF Señal_mando1<5250 THEN (*Manual_C3*)
SET Manual_c3;
ELSIF Señal_mando1<5750 THEN (*Manual_sel1_atras*)
SET Manual_sel1_atras;
ELSIF Señal_mando1<6250 THEN (*Manual_C4*)
SET Manual_c4;
ELSIF Señal_mando1<6750 THEN (*Manual_sel2_atras*)
SET Manual_sel2_atras;
ELSIF Señal_mando1<7250 THEN (*Manual_m1*)
SET Manual_m1;
ELSIF Señal_mando1<7750 THEN (*Manual_m2*)
SET Manual_m2;
END_IF;
END_IF;
```

Figura A.4. Programación de SR0.

### A.3. Programación de Chart.

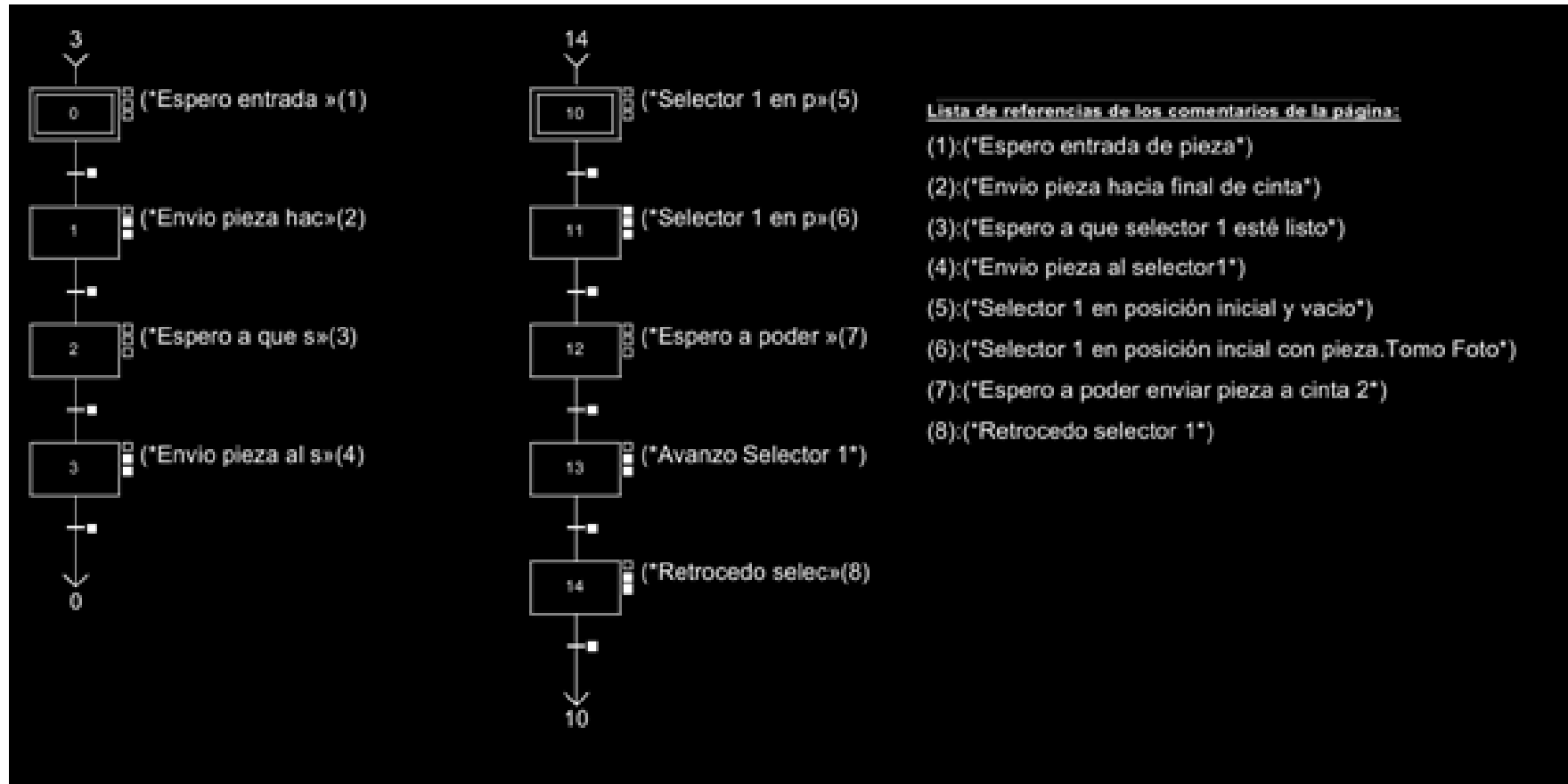


Figura A.5. Programación de Chart. Grafos 1 y 2.

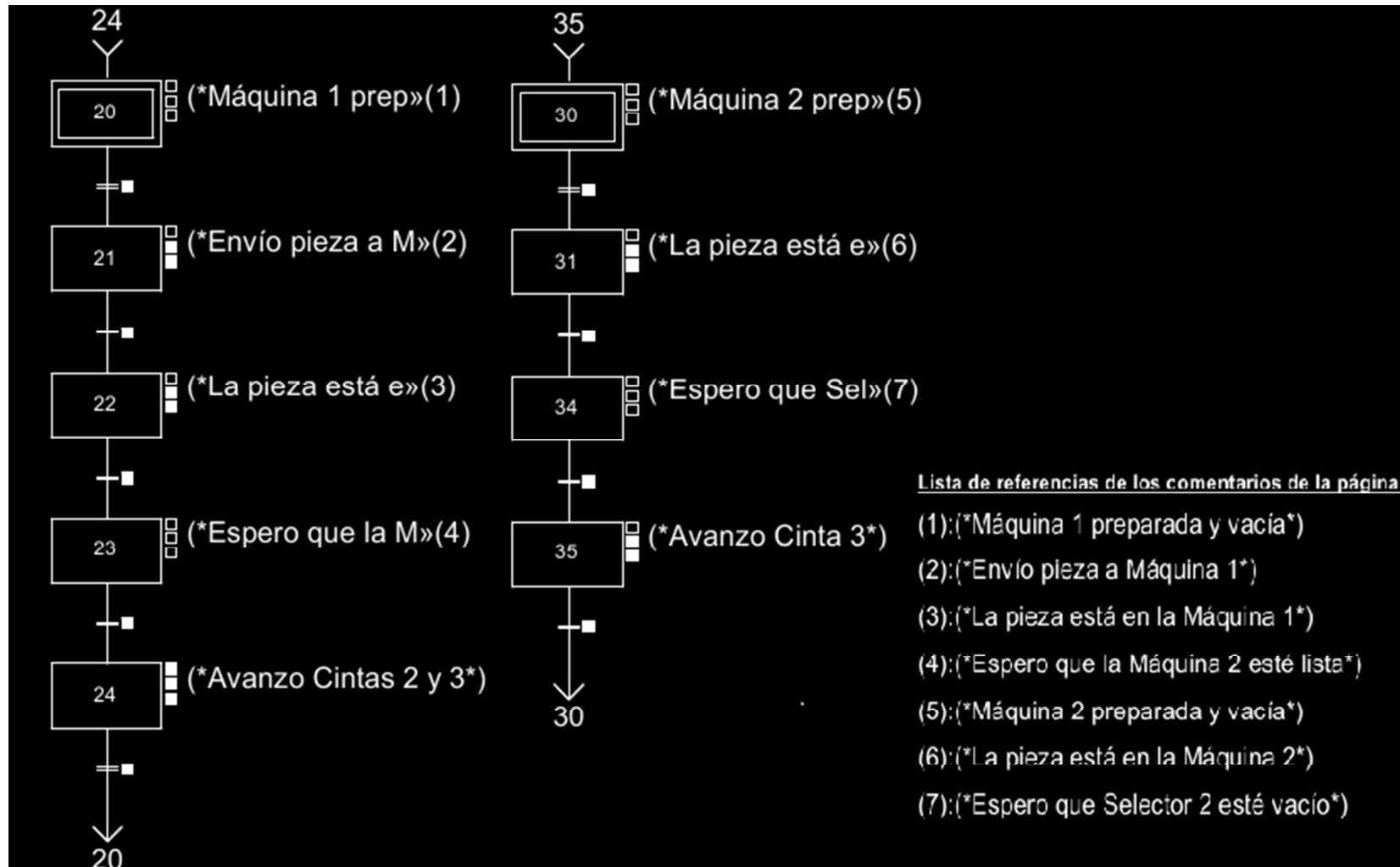


Figura A.6. Programación de Chart. Grafos 3 y 4.

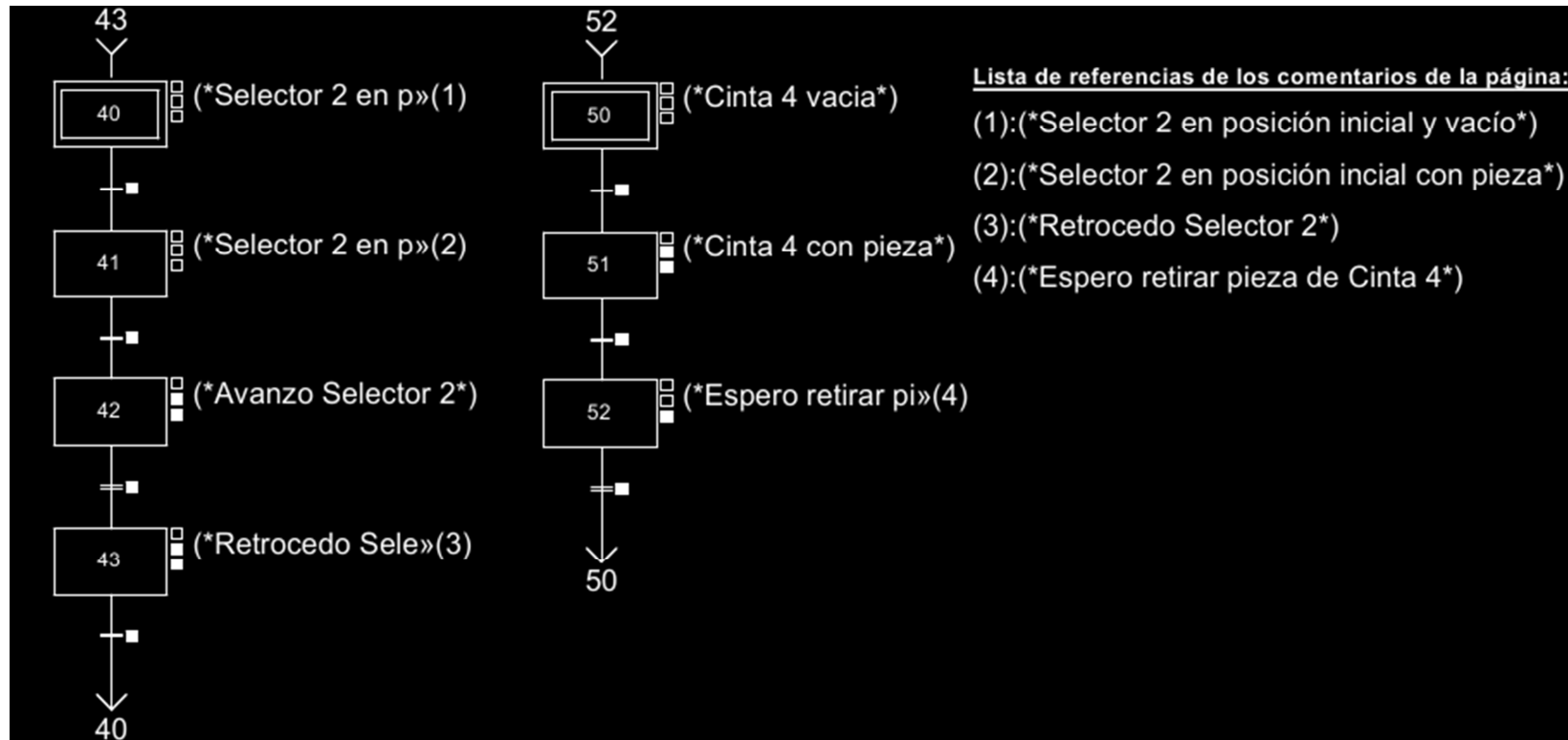


Figura A.7. Programación de Chart. Grafos 5 y 6.

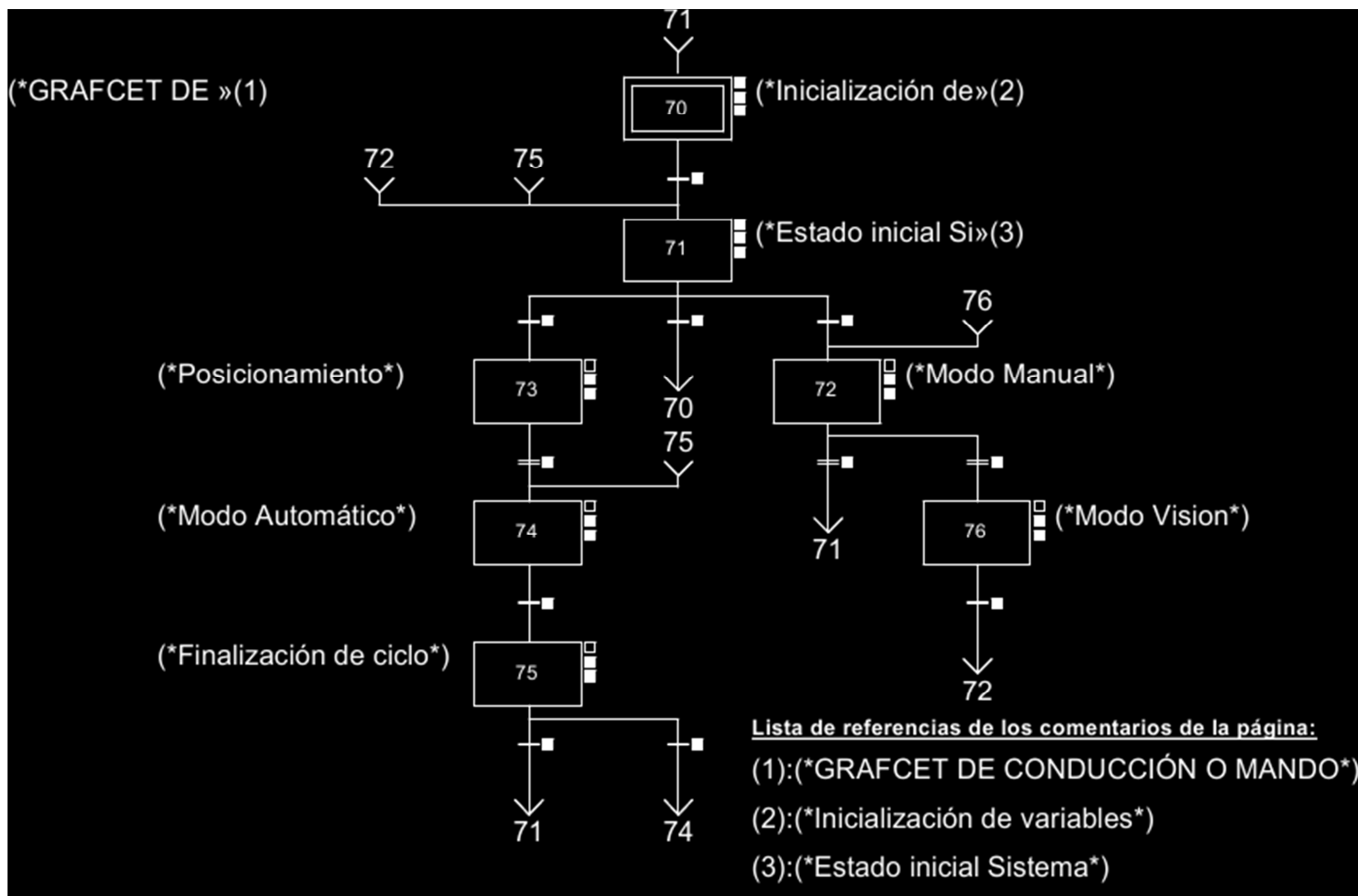


Figura A.8. Programación de Chart. Grafjet de Conducción o Mando.



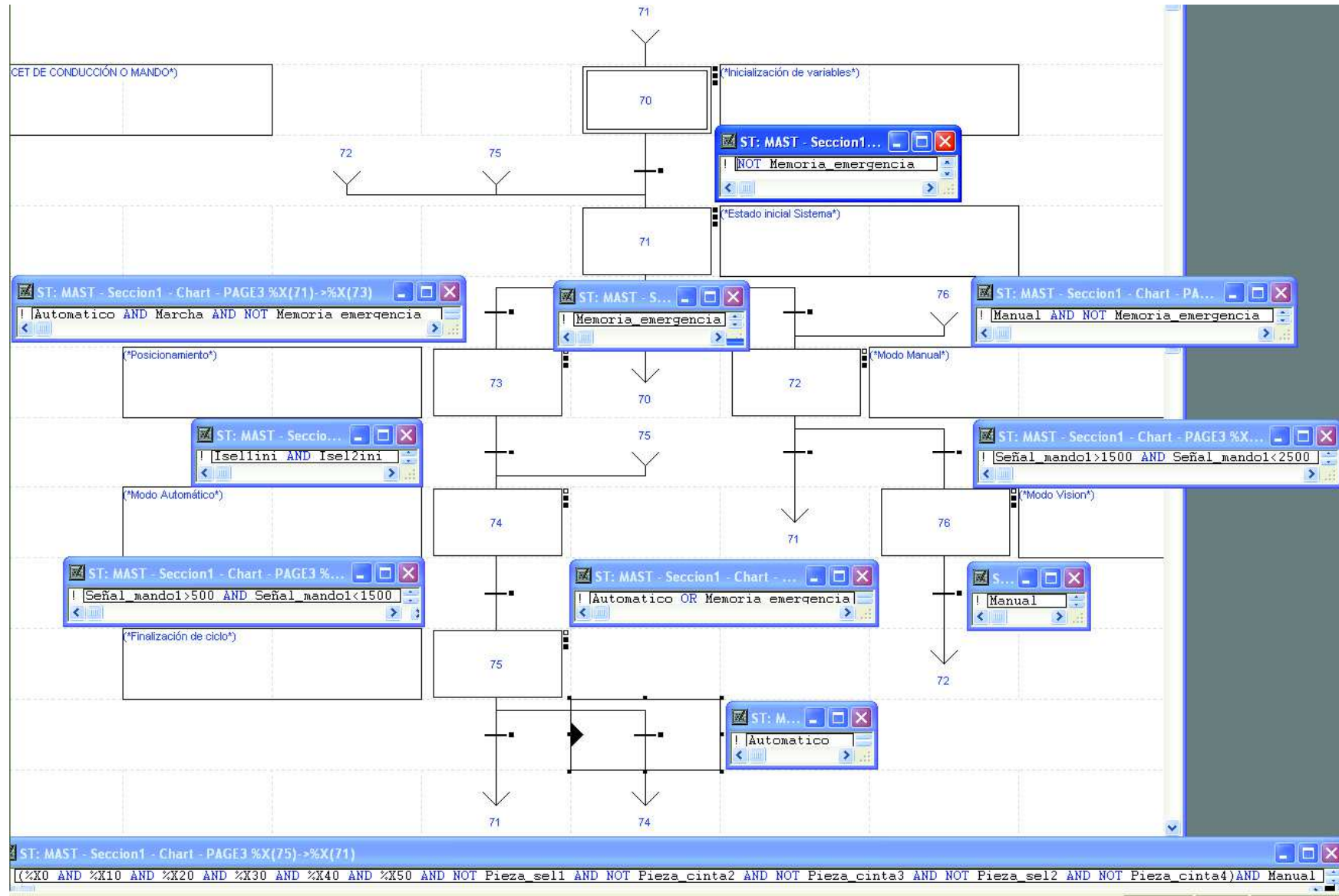


Figura A.9. Grafset de Conducción o Mando con transiciones (misma figura que figura 4.13 de la memoria).

#### A.4. Programación del POST.

Se han programado las salidas en este apartado.

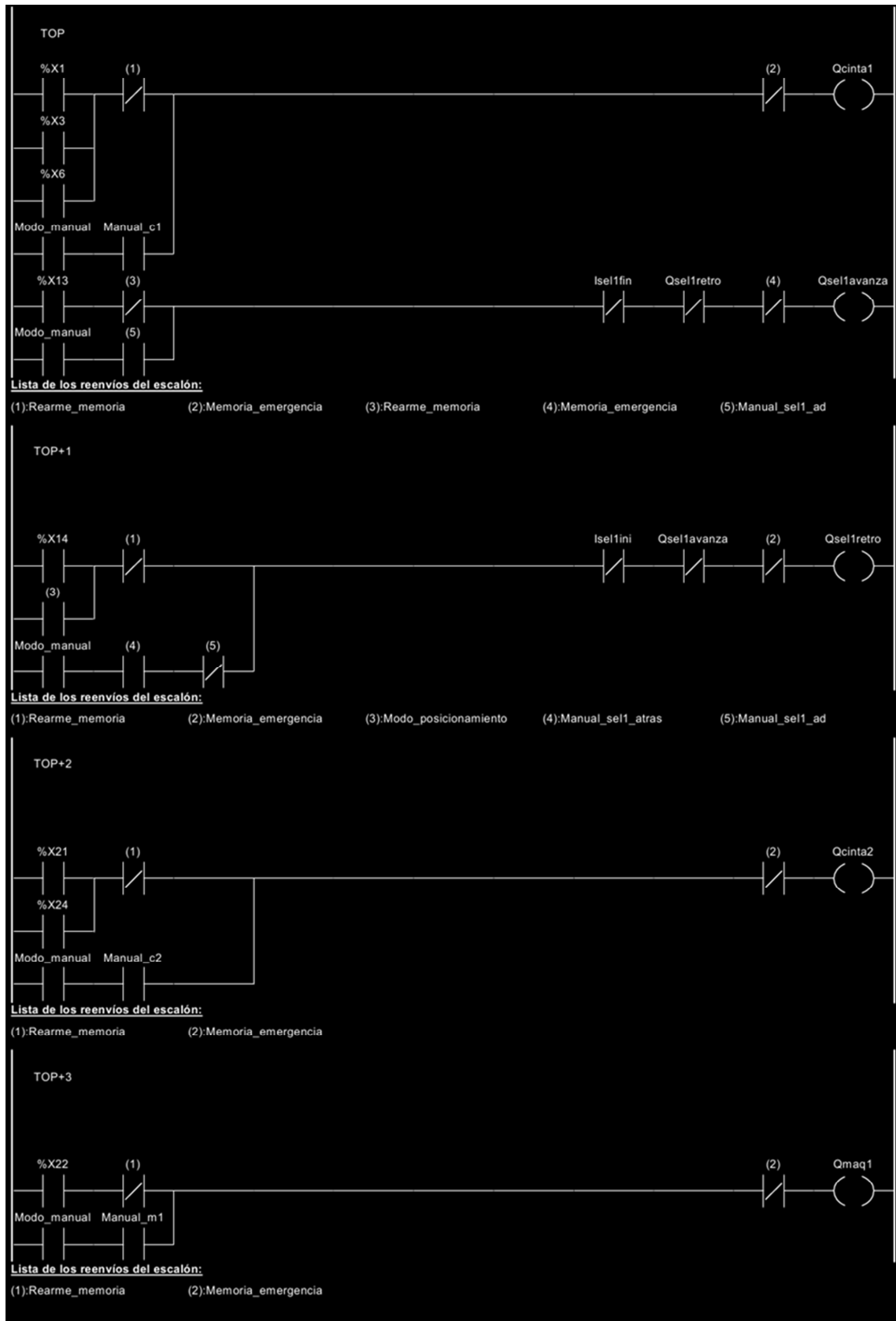


Figura A.10. POST primera parte.

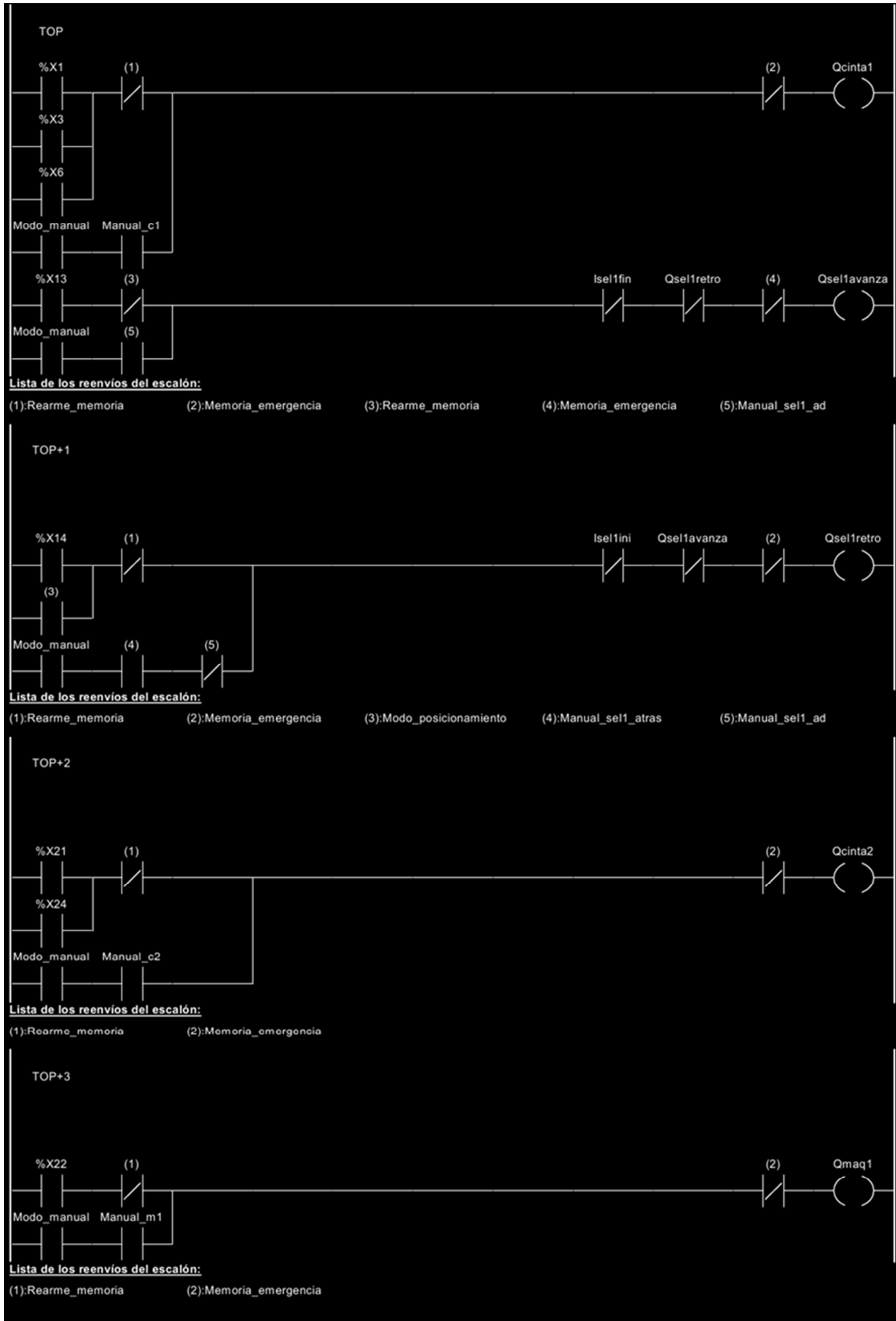


Figura A.11. POST Segunda parte.

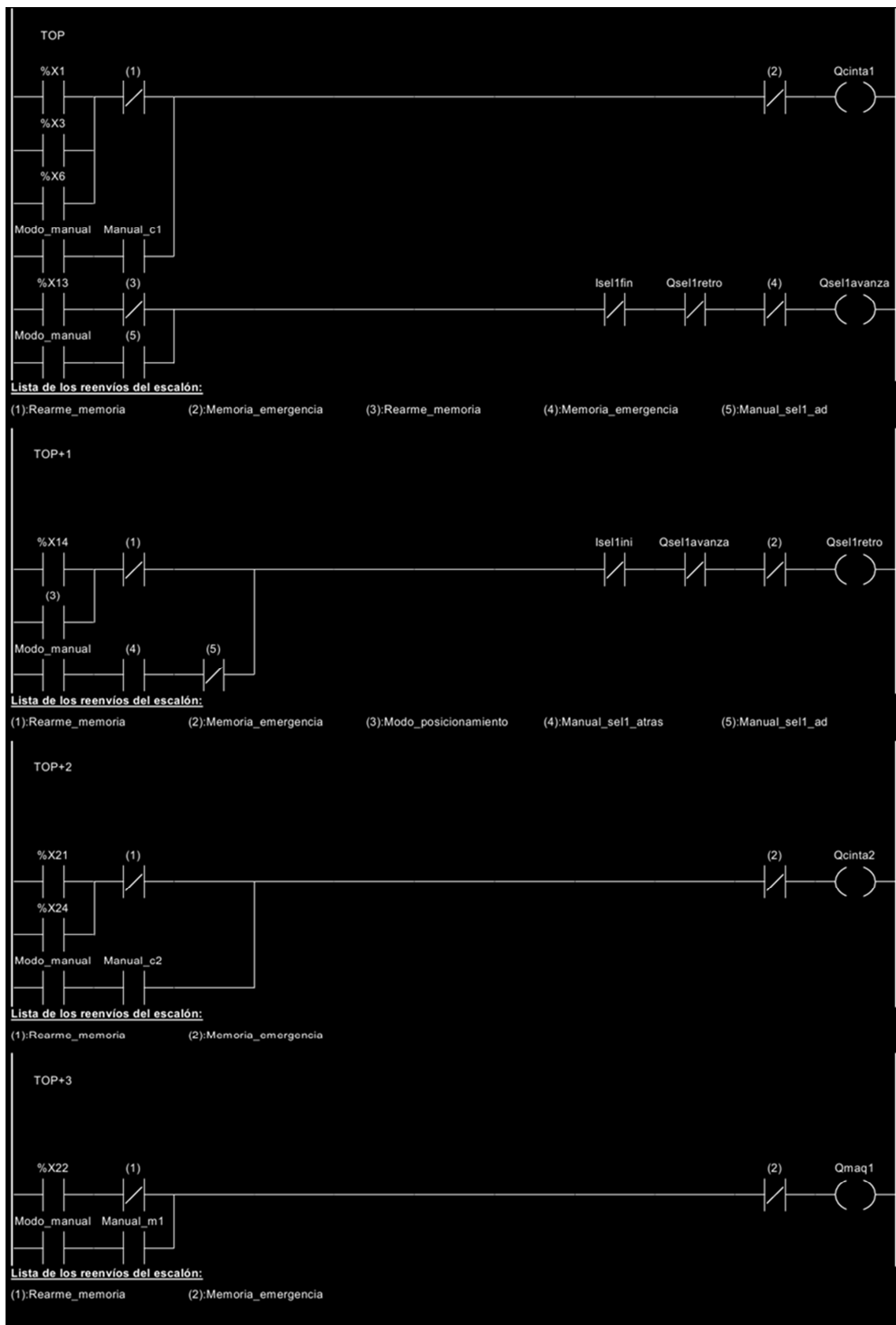


Figura A.12. POST Tercera parte.

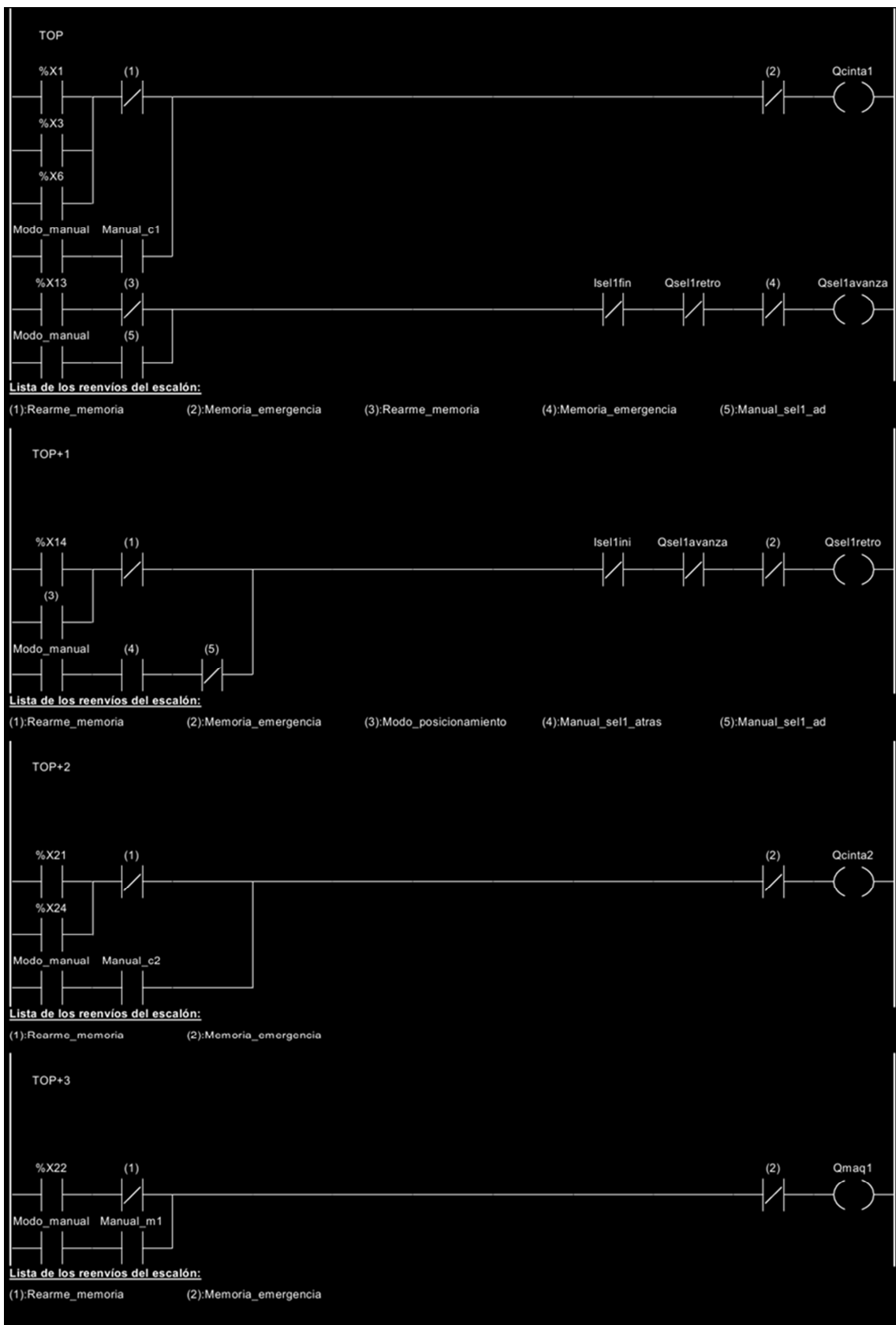


Figura A.13. POST Cuarta parte.

## A.5. Variables utilizadas.

Simbolo	Variable	Comentario
lsel1fin	%I1.0	(*Selector 1 en posición final*)
lsensorc1	%I1.1	(*Sensor presencia final cinta 1*)
lsensorcarga	%I1.2	(*Sensor presencia de nuevo producto*)
lsensormaq1	%I1.3	(*Sensor presencia máquina 1*)
lsensormaq2	%I1.4	(*Sensor presencia máquina 2*)
lsel1ini	%I1.5	(*Selector 1 en posición inicial*)
lsensorc4	%I1.6	(*Sensor presencia final cinta 4*)
lsel2fin	%I1.7	(*Selector 2 en posición final*)
Qsel1avanza	%Q2.0	(*Selector 1 avanza*)
Qsel1retro	%Q2.3	(*Selector 1 retrocede*)
Qsel2avanza	%Q2.4	(*Selector 2 avanza*)
Qsel2retro	%Q2.7	(*Selector 2 retrocede*)
Qcinta1	%Q2.5	(*Cinta 1 avanza*)
Qcinta2	%Q2.6	(*Cinta 2 avanza*)
Qmaq1	%Q2.1	(*Maquina 1 trabaja*)
Qmaq2	%Q2.8	(*Maquina 2 trabaja*)
Qcinta3	%Q2.2	(*Cinta 3 avanza*)
Qcinta4	%Q2.10	(*Cinta 4 avanza*)
Tsegc1	%TM1	(*Tiempo de seguridad de funcionamiento de Cinta1*)
Tsegc2	%TM2	(*Tiempo de seguridad de funcionamiento de Cinta2*)
Tsegc4	%TM4	(*Tiempo de seguridad de funcionamiento de Cinta4*)
Tsegc3	%TM3	(*Tiempo de seguridad de funcionamiento de Cinta3*)
Cpiezas_entrada	%C0	(*Contador de piezas en Cinta1*)
Tmaq2_pieza_a	%TM32	(*Tiempo de trabajo de la Máquina 2 para la pieza A*)
Tmaq2_pieza_b	%TM33	(*Tiempo de trabajo de la máquina 2 para la pieza B*)
Tmaq1_pieza_b	%TM24	(*Tiempo de trabajo de la Máquina 1 para la pieza B*)
Tmaq1_pieza_a	%TM23	(*Tiempo de trabajo de la Máquina 1 para la pieza A*)
T_m2_sel2	%TM35	(*Tiempo que cuesta pasar la pieza de la Máquina 2 al Selector 2*)
Tsegsel2re	%TM43	(*Tiempo de seguridad de funcionamiento del Selector 2 hacia detras*)
Tsegsel2av	%TM42	(*Tiempo de seguridad de funcionamiento del Selector 2 hacia adelante*)
Tsegsel1av	%TM13	(*Tiempo de seguridad de funcionamiento del Selector 1 hacia adelante*)
Seta_emergencia	%I1.9	(*Pulsador de Seta de Emergencia*)
lsel2ini	%I1.11	(*Selector 2 en posición inicial*)
F_tarjetas	%M60	(*Fallos de tarjetas Entrada/salida*)
F_tiempos	%M61	(*Fallos por tiempos excesivos*)
Situacion_emergencia	%M63	(*Marca que indica que se ha dado una situación de emergencia*)
F_sitilogicas	%M62	(*Fallos por situaciones ilógicas de los captadores*)
Memoria_emergencia	%M64	(*Indica que se ha producido una situación de emergencia*)
Trearme	%TM10	(*Tiempo de rearme*)
Magcinta1	%M21	(*Magnetotérmico Cinta1*)
Magcinta2	%M22	(*Magnetotérmico Cinta2*)
Magcinta3	%M23	(*Magnetotérmico Cinta3*)
Magcinta4	%M24	(*Magnetotérmico Cinta4*)
Magmaq1	%M25	(*Magnetotérmico Maquina1*)



Magmaq2	%M26	(*Magnetotérmico Maquina2*)
Magsel1adelante	%M27	(*Magnetotérmico Seleccionador1 adelante*)
Magsel1ras	%M28	(*Magnetotérmico Seleccionador2 ras*)
Magsel2adelante	%M29	(*Magnetotérmico Seleccionador2 adelante*)
Magsel2ras	%M30	(*Magnetotérmico Seleccionador2 ras*)
F_magneto	%M20	(*Marca que indica que hay algún magnetotérmico disparado*)
T_c1_sel1	%TM6	(*Tiempo que cuesta pasar la pieza de la Cinta 1 al Selector 1*)
Run_stop	%I1.8	(*Habilitación del Automata*)
F_tiempos_sel1	%M65	(*Fallo por tiempo excesivo Selector1*)
F_tiempos_sel2	%M66	(*Fallo por tiempo excesivo Selector2*)
Marca_x72	%M72	(*Indicador de sistema en estado %X72*)
Marca_x71	%M71	(*Indicador de sistema en estado %X71*)
Marca_x73	%M73	(*Indicador de sistema en estado %X73*)
Marca_x74	%M74	(*Indicador de sistema en estado %X74*)
Marca_x75	%M75	(*Indicador de sistema en estado %X75*)
Señal_mando1	%IW0.2	(*Señal de Mando1*)
E_situacion_emergencia	%Q4.0	(*Señal de comunicación de estado de situación de Emergencia*) (*Señal de comunicación de estado de situación de Memoria de Emergencia*)
E_memoria_emergencia	%Q4.1	
E_inicial	%Q4.2	(*Señal de comunicación de estado Inicial*)
E_manual	%Q4.3	(*Señal de comunicación de estado Manual*)
E_posicionamiento	%Q4.4	(*Señal de comunicación de estado Posicionamiento*)
E_automatico	%Q4.5	(*Señal de comunicación de estado Automático*)
E_fin_ciclo	%Q4.6	(*Señal de comunicación de estado Fin de Ciclo*)
E_rearme_memoria	%Q4.7	(*Señal de comunicación de estado de sistema rearmado*)
E_deteccion	%Q4.8	(*Señal de comunicación de estado en proceso de detección*)
Marca_x70	%M70	(*Indicador de sistema en estado %X70*)
Rearme	%I3.0	(*Pulsador para rearmar el sistema*)
Marcha	%I3.1	(*Pulsador de marcha*)
Manual_m1	%I3.2	(*Pulsador para mover la Maquina 1 de forma manual*)
Manual_m2	%I3.3	(*Pulsador para mover la Maquina 2 de forma manual*)
Manual_c1	%I3.4	(*Pulsador para mover la Cinta 1 de forma manual*)
Manual_c2	%I3.5	(*Pulsador para mover la Cinta 2 de forma manual*)
Manual_c3	%I3.6	(*Pulsador para mover la Cinta 3 de forma manual*)
Manual_c4	%I3.7	(*Pulsador para mover la Cinta 4 de forma manual*)
Manual_sel1_ad	%I3.8	(*Pulsador para mover el Selector 1 adelante de forma manual*)
Manual_sel2_ad	%I3.10	(*Pulsador para mover el Selector 2 adelante de forma manual*)
Manual_sel1_ras	%I3.9	(*Pulsador para mover el Selector 1 ras de forma manual*)
Manual_sel2_ras	%I3.11	(*Pulsador para mover el Selector 2 ras de forma manual*)
Foto_ok	%I1.10	(*La foto ha sido tomada*)
Señal_control	%QW0.10	(*Señal de control*)
Pieza_sel1	%M100	(*Marca para comunicar los grafcet. Indico que la pieza está en el selector 1*)
Pieza_cinta2	%M101	(*Marca para comunicar los grafcet. Indico que la pieza está en cinta2*)
Pieza_cinta3	%M102	(*Marca para comunicar los grafcet. Indico que la pieza está en cinta3*)
Pieza_sel2	%M103	(*Marca para comunicar los grafcet. Indico que la pieza está en el selector 2*)
Pieza_cinta4	%M104	(*Marca para comunicar los grafcet. Indico que la pieza está en cinta4*)
Rearme_memoria	%M6	(*Se ha producido un rearme de forma correcta*)
Marca_x11	%M11	(*Marca para indicar al PC que se puede Tomar foto en esta etapa*)



F_comunicacion	%M69	(*Fallo debido a perdida de comunicacion en tre PLC y SCADA*)
T_maq1_sel1	%TM50	(*Tiempo de trabajo de la Maquina1 tomado en el selector1*)
T_maq2_sel1	%TM51	(*Tiempo de trabajo de la Maquina2 tomado en el selector1*)
Automático	%I3.12	(*Selector en Modo Automático*)
Manual	%I3.13	(*Selector en Modo Manual*)
T_maq1_m1	%TM52	(*Tiempo de trabajo de la maquina1 tomado en la máquina1*)
T_maq2_m1	%TM53	(*Tiempo de trabajo de la maquina2 tomado en la máquina1*)
T_maq2_m2	%TM54	(*Tiempo de trabajo de la maquina2 tomado en la máquina2*)
Señal_mando2	%IW0.7	(*Señal de Mando 2*)
Foto_ok_a	%IW0.3	(*Señal analógica de comunicación de que foto se ha tomado*)
Error_foto_a	%IW0.4	(*Señal analógica de comunicación de que ha ocurrido error en foto*)
Rearme_a	%IW0.5	(*Señal analógica de comunicación de Rearme*)
Marcha_a	%IW0.6	(*Señal analógica de comunicación de Marcha*)
Auto_manual_a	%IW0.8	(*Señal analógica de comunicación de Manual o Auto*)
Rearmando	%M7	(*Marca que indica que el sistema se está rearmando*)
Error_foto	%I3.14	(*Error al tomar la Fotografía*)
Modo_manual	%X72	(*Funcionamiento Modo Manual*)
Modo_posicionamiento	%X73	(*Funcionamiento Modo Posicionamiento*)
Modo_automatiko	%X74	(*Funcionamiento Modo Automático*)
Modo_finciclo	%X75	(*Funcionamiento Fin de Ciclo*)
Tsegsel1re	%TM14	(*Tiempo de seguridad de funcionamiento del Selector 1 hacia detrás*)
Marca_x76	%M76	(*Funcionamiento Modo Vision*)
F_rearraque_caliente	%M59	(*Fallo producido por un re arranque en caliente*)
E_rearmando	%Q4.9	(*Señal de comunicación de estado sistema rearmando*)
E_modovision	%Q4.10	(*Señal de comunicación de estado Modo Vision*)

**A.1. Variables de trabajo.**





**Bucle de ejecución principal:** Se ejecuta un bucle (ver figura B.2), en cuanto se abre el archivo, que solo finaliza cuando el usuario pulsa **“EXIT LABVIEW”**, con el sistema en STOP. En cada iteración que se lleva a cabo, se realizan lecturas de los pulsadores de mando, especialmente importante el de **“STOP/RUN”**. En la figura se muestra la ejecución realizada cuando el sistema está en STOP (reinicio de indicadores y señales de mando y control, actualización de pantalla de información).

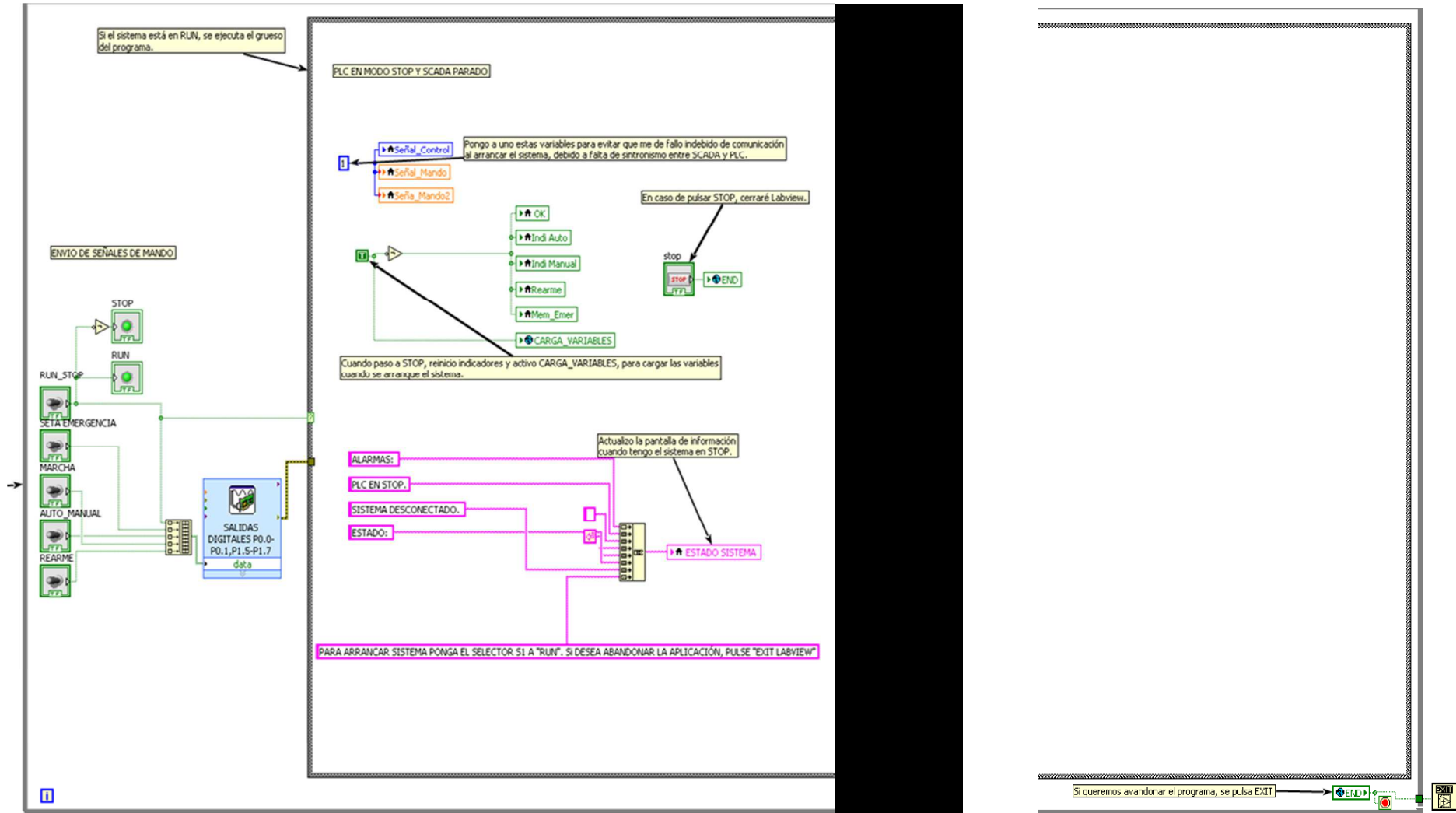


Figura B.2. Ejecución del bucle principal y funciones ejecutadas con el Sistema en STOP.

**Secuencia primera:** Con el sistema en RUN, se ejecutan tres secuencias. La figura B.3 muestra la primera de estas secuencias. Se leen las señales de estado procedentes del PLC y la “*Señal\_Control*” (alarmas). Si es el primer ciclo tras STOP o se ha producido alguna acción que hace necesaria actualización de variables con valores almacenados en archivos de registro, se realizará la ejecución del subVI “*Carga\_Archivos.vi*”, con el que se cargarán los datos en el sistema. Destacar también el control de posibles fallos en la DAQ, y su comunicación al PLC en caso de darse dichos fallos.

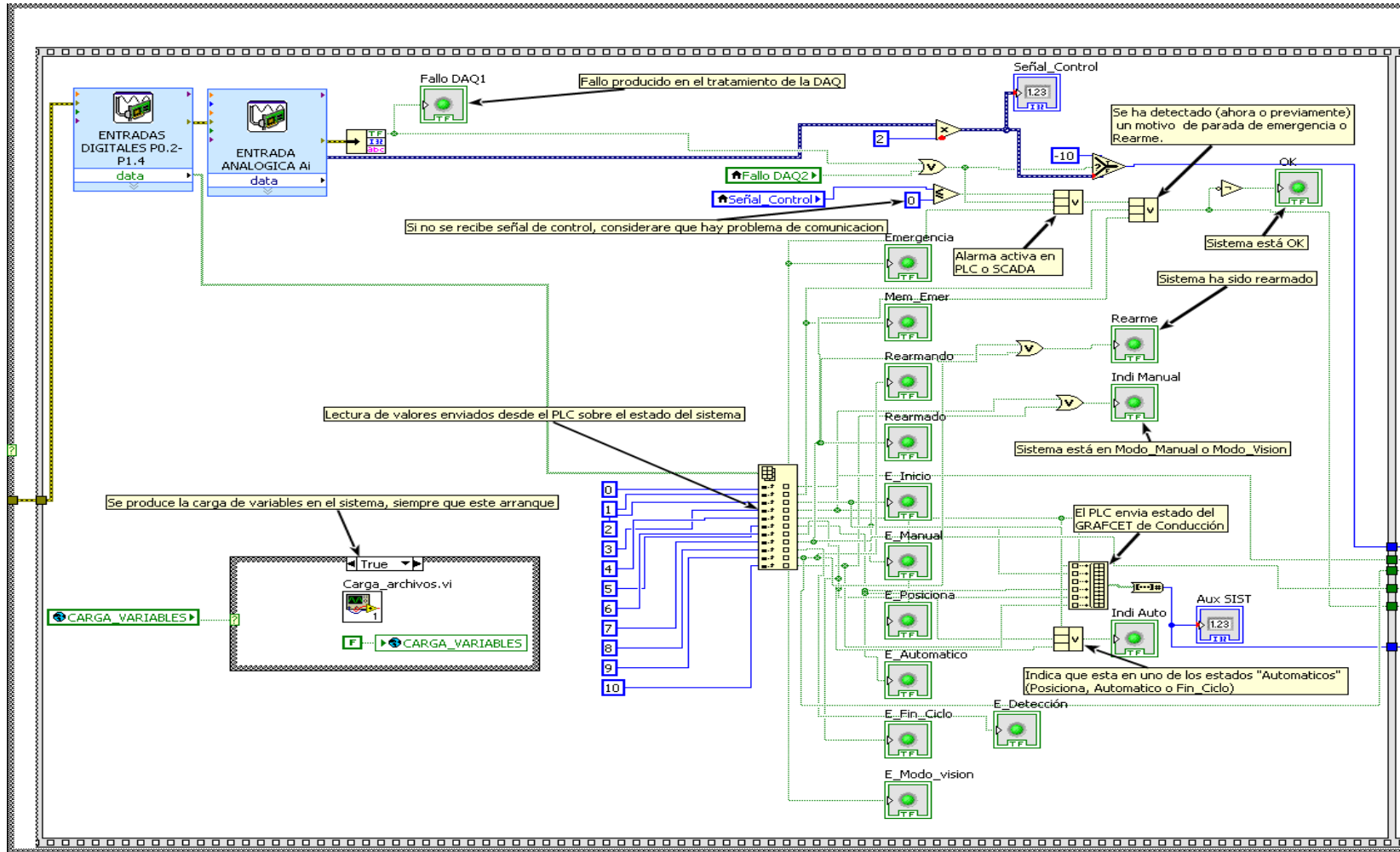


Figura B.3. Secuencia primera en RUN.

**Carga\_archivos.vi:** En la figura B.4 se observa cómo se realiza la carga de los archivos de registro mencionados en el apartado 3.2.6 de la memoria. Se limita el número de elementos que deben tener las variables que son de tipo array. Se ejecuta una subVI, “*Carga\_Im\_patrones.vi*” que realizará la carga de las imágenes para visualizar en el sistema. Se observa que todos archivos se encuentran en la misma ubicación, la carpeta llamada “**Archivos SCADA**” comentada en el apartado 3.6 de la memoria.

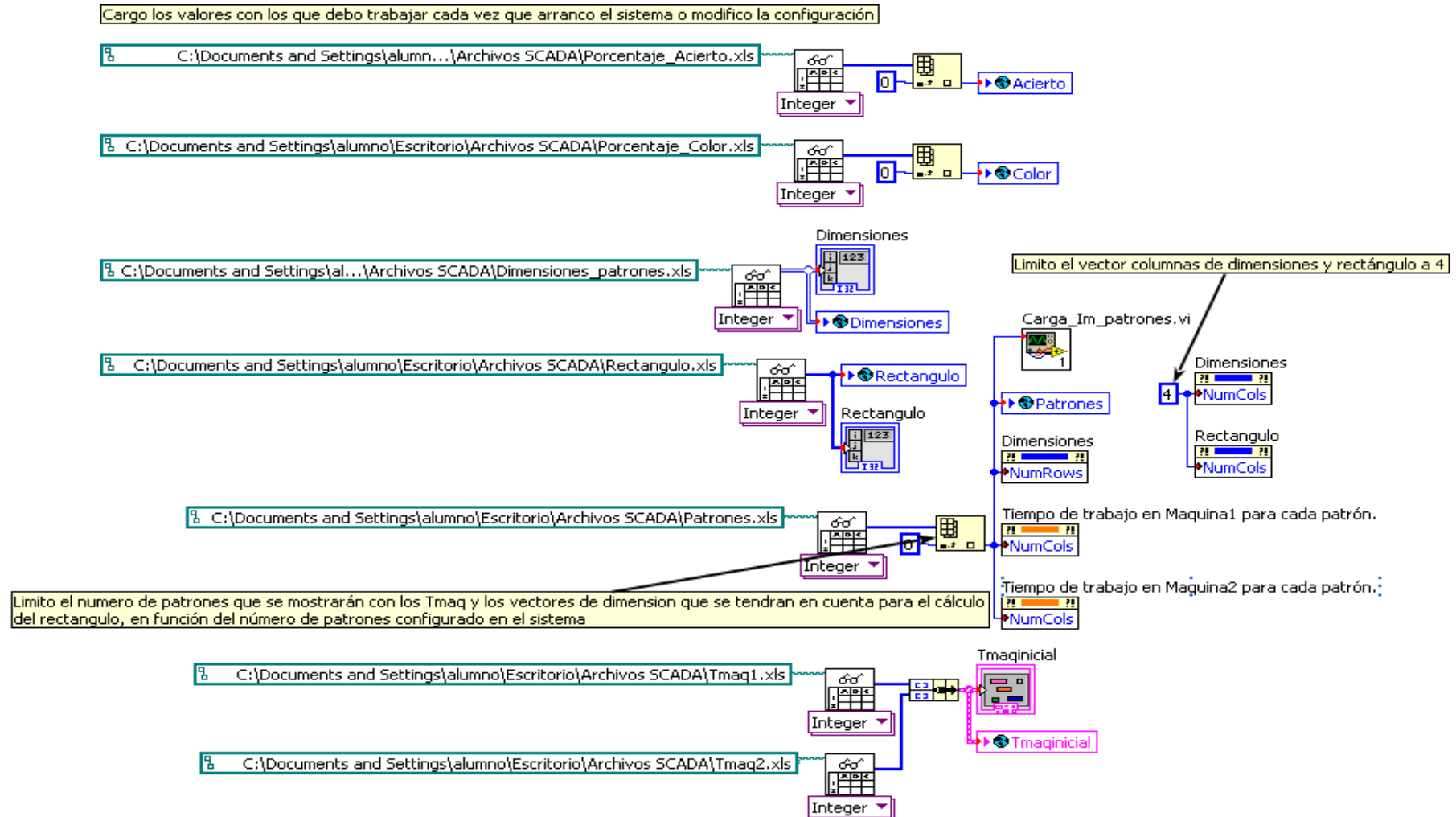


Figura B.4. Carga\_archivos.vi.

**Carga\_Im\_patrones.vi:** Su función es cargar en el sistema las imágenes de los patrones con los que se está trabajando. Como se aprecia en la figura B.5.a, todas las imágenes se vuelven a guardar en la carpeta “**Archivos SCADA**”. Se carga también la imagen que sacará el sistema cuando no detecte ninguno de los patrones disponibles, que será la que se observa en la figura B.5.b.

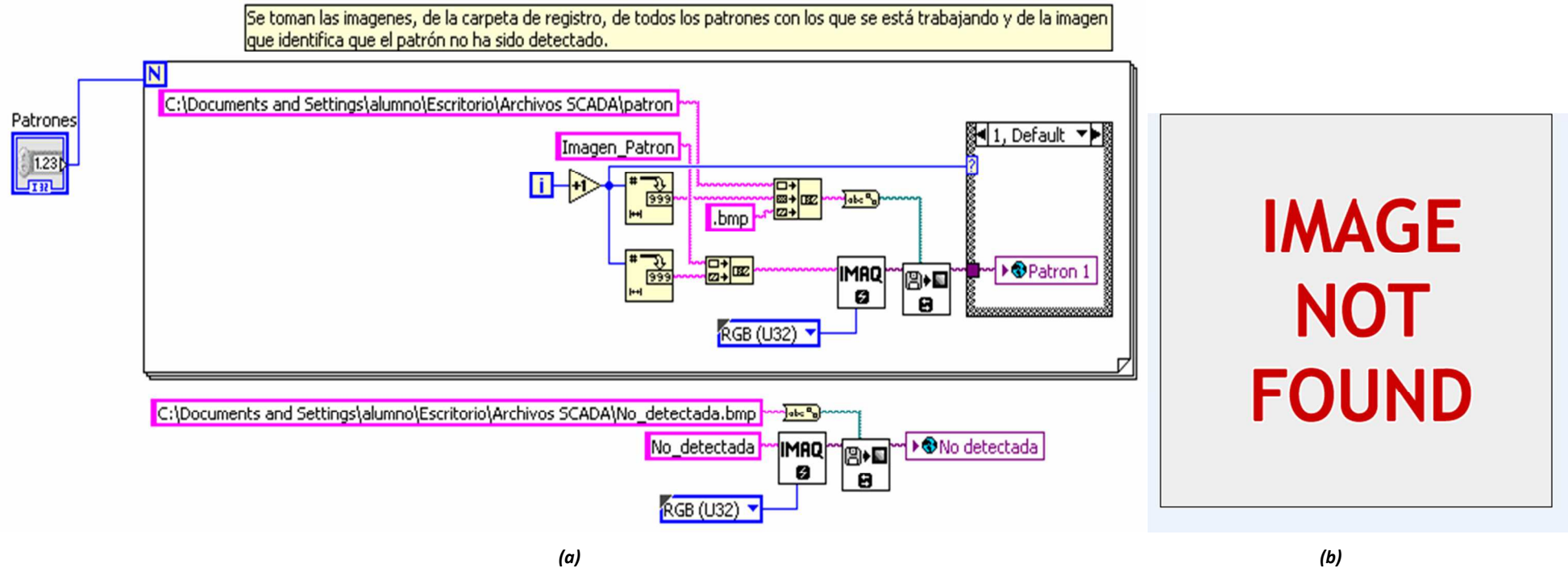


Figura B.5. Carga de imágenes de patrones en el sistema. (a)Carga\_Im\_patrones.vi. (b) Imagen que indica que no se ha detectado ningún patrón.

**Secuencia Segunda:** La figura B.6.a muestra la segunda secuencia, en la que únicamente se actualiza la pantalla de información del SCADA en función de las señales que recibe del PLC. Este proceso se realiza a través de la subVI **"Alarmas\_estado.vi"** que se ve en la figura B.6.b. Se dividen los comentarios en 4 partes, que muestran el modo de funcionamiento, las posibles alarmas producidas y otra información útil para el usuario en función de las variables de estado recibidas desde el PLC.

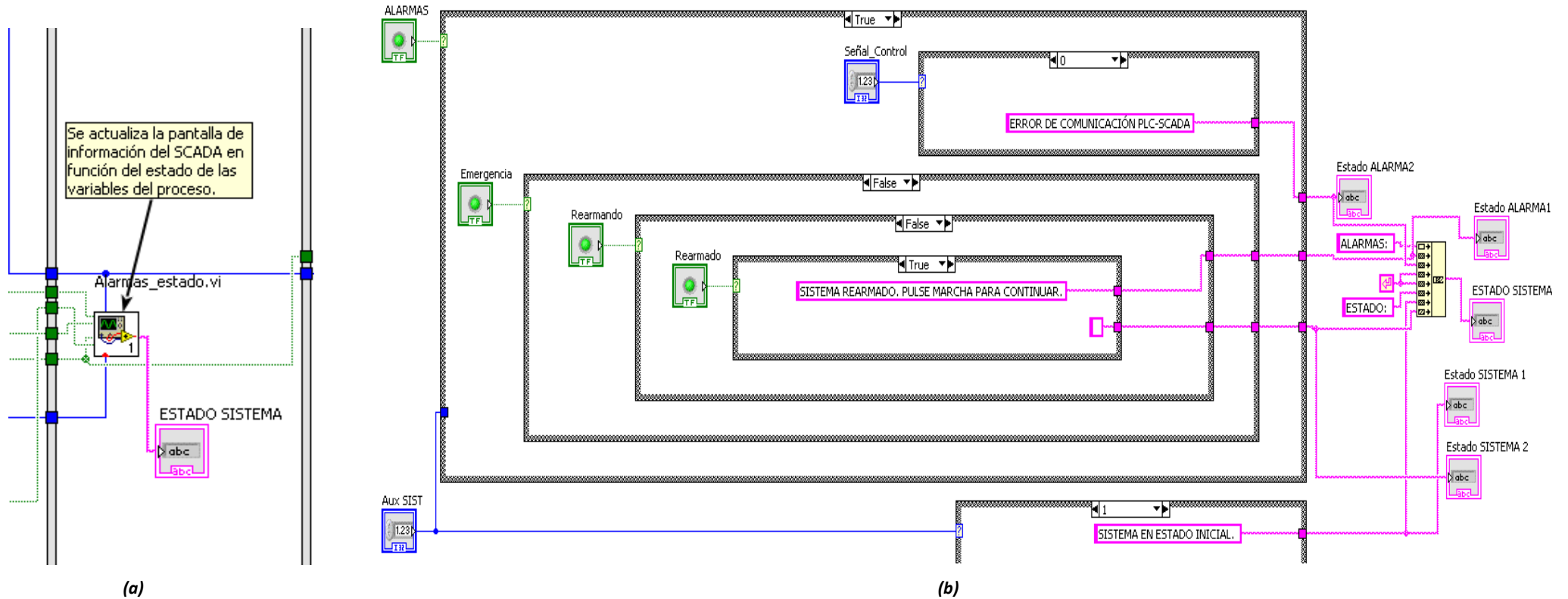


Figura B.6. Actualización de alarmas de estado. (a) Secuencia segunda donde se realiza la actualización. (b) SubVI que realiza dicha actualización.



**Secuencia Tercera Emergencia:** La figura B.7 muestra la ejecución de la tercera secuencia cuando se ha producido algún motivo de parada de emergencia. Destacar que si la “*Señal\_Control*” es menor o igual a 0 (se ha perdido comunicación por el cable de comunicación de dicha señal), se realiza un envío de 0v por la “*Señal\_Mando*” que originará, de acuerdo a lo programado en el PLC, que éste genere una parada informando al SCADA de pérdida de comunicación entre PLC-SCADA. Hasta que no se recupere la comunicación, no se podrá continuar el proceso.

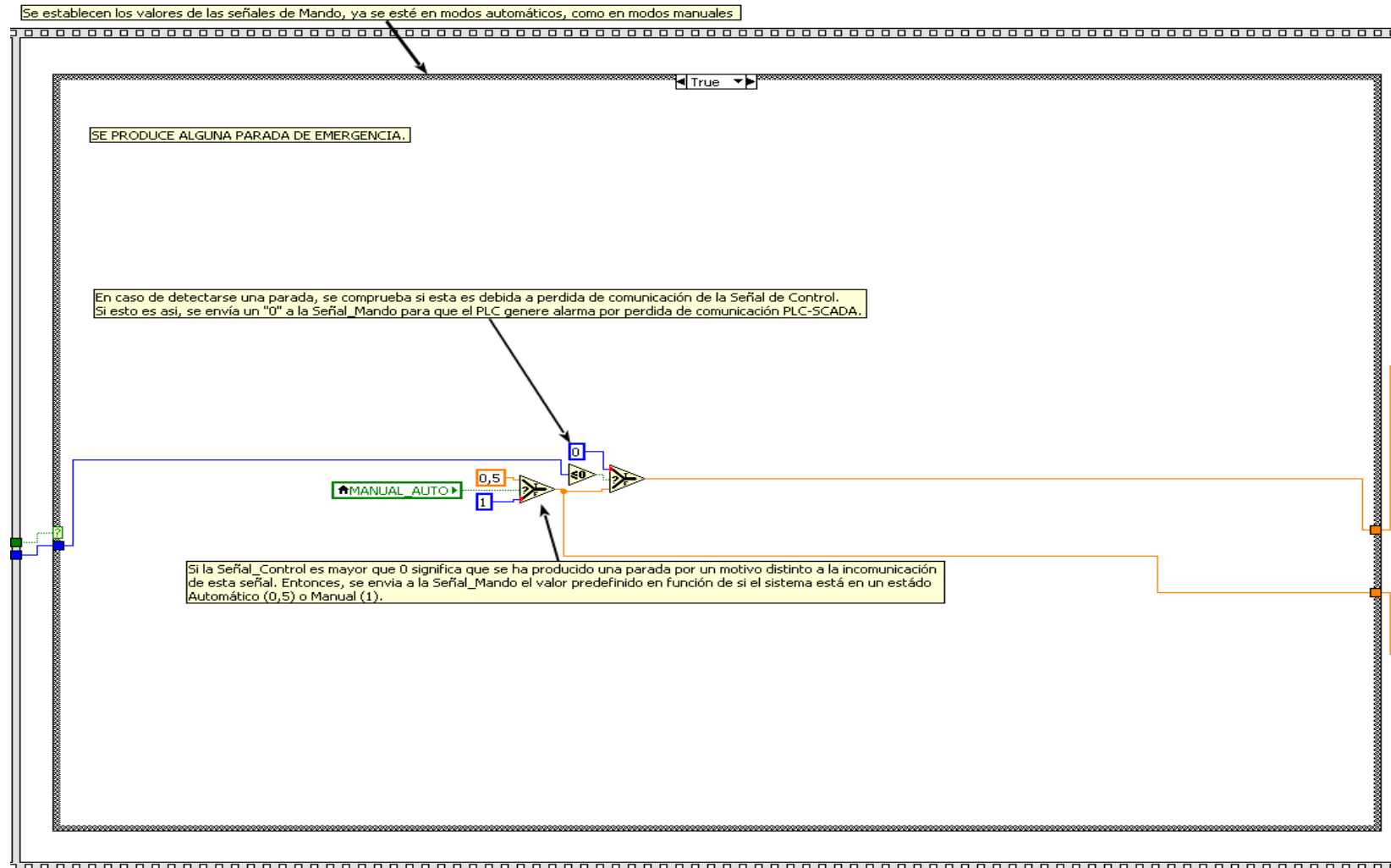


Figura B.7. Ejecución de Secuencia Tercera Emergencia

**Secuencia Tercera sin Emergencia y modo Automático distinto a detección:** La figura B.8 muestra la ejecución de la tercera secuencia cuando no se ha producido parada de emergencia, se está en modo Automático y en un estado distinto del estado de “*E\_Detección*”. En este caso, lo único que hace el SCADA es enviar 0,5v a las señales de mando indicando al PLC que todo está correcto. La variable “*Patron\_detectado*” se pone a falso por temas de sincronismo.

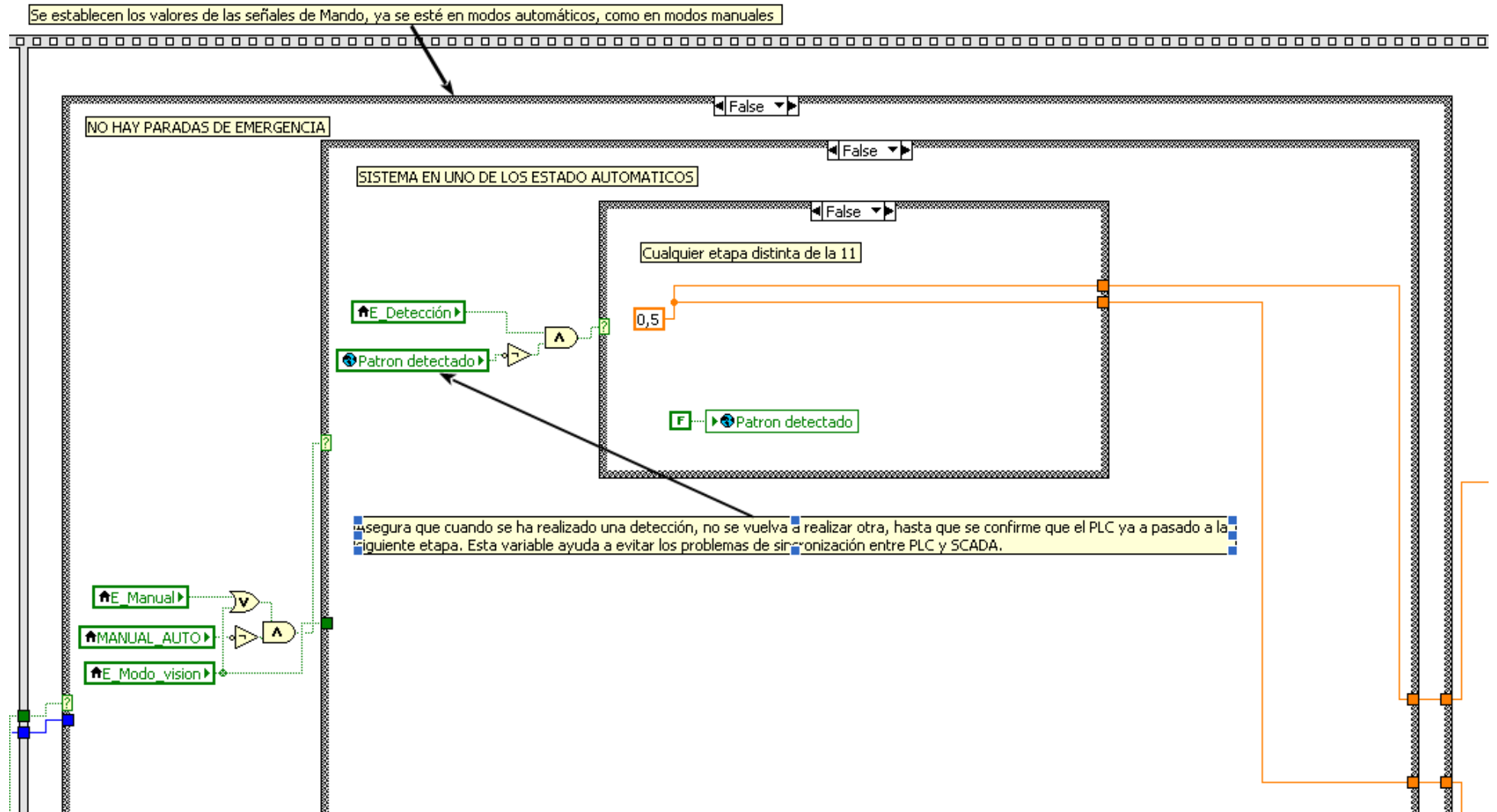


Figura B.8. Ejecución de Secuencia Tercera sin Emergencia y modo Automático distinto a detección.



**Secuencia Tercera sin Emergencia y modo Automático en la etapa de detección:** La figura B.9 muestra la ejecución de la tercera secuencia cuando no se ha producido parada de emergencia, se está en modo Automático y en estado de detección “*E\_Detección*”. En este punto se realizará la ejecución del algoritmo detector de patrones mediante el subVI “*Detector\_patrones.vi*”. Se verá esta configuración más adelante en este anexo.

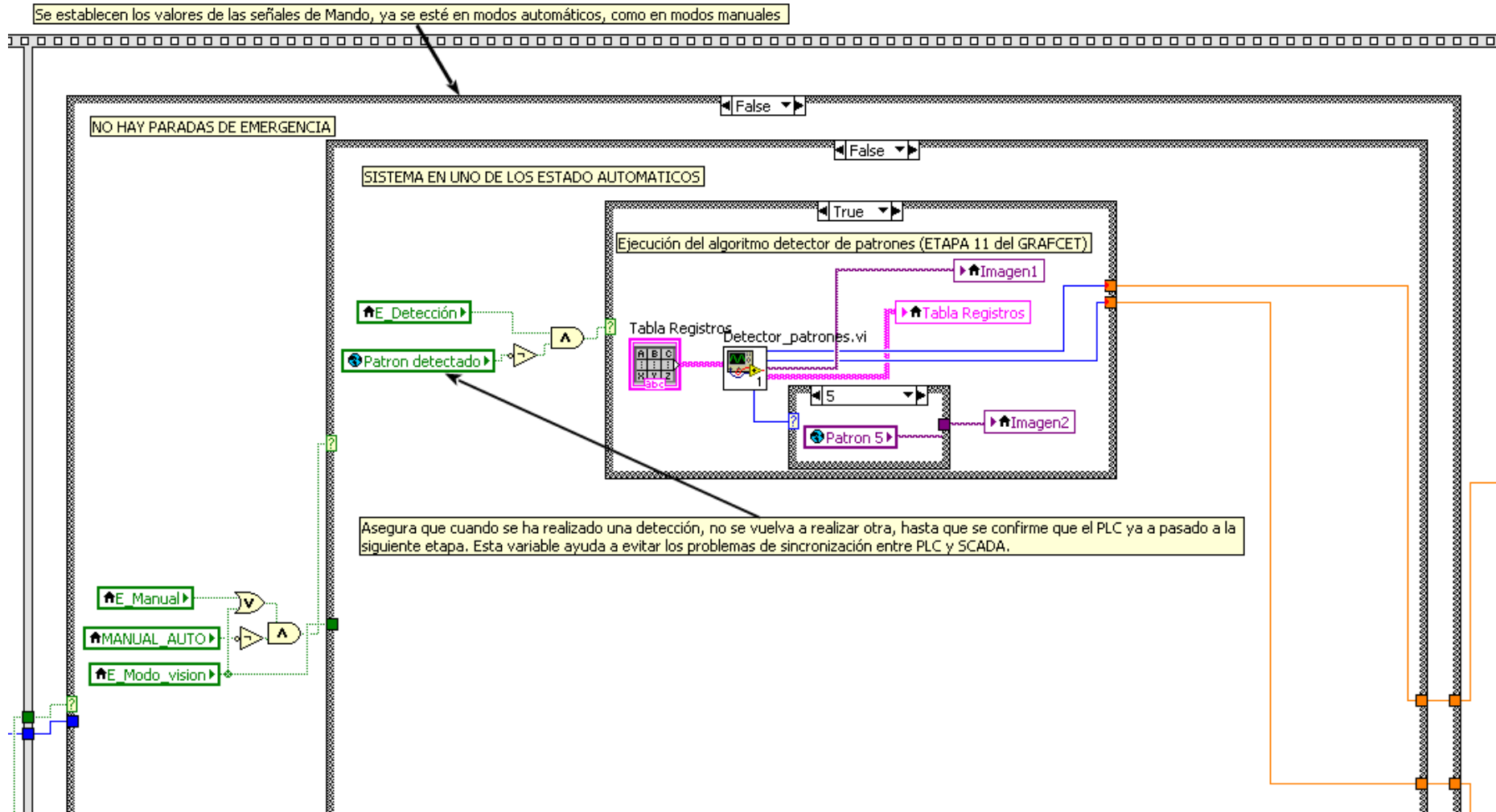


Figura B.9. Ejecución de Secuencia Tercera sin Emergencia y modo Automático en estado de detección.

**Secuencia Tercera sin Emergencia y modo Manual:** La figura B.10 muestra la ejecución de la tercera secuencia cuando no se ha producido parada de emergencia y se está en modo Manual, a la espera de que algún mando manual sea pulsado para habilitar el dispositivo respectivo. El sistema está diseñado, ante las limitaciones del número de señales de comunicación, de forma que solo uno de estos dispositivos sea movido a la vez. Por eso, se ha creado una estructura de prioridades donde "Sel1" tiene prioridad sobre todos y "Sel2" sobre el resto. El hecho de que la pulsación sobre los mandos sea mediante ratón, hace que no haya problemas de pulsaciones simultáneas que pudieran alterar el movimiento del resto de mandos. Aquí se ve la forma en que se generan las señales vistas en la figura 5.1 (en el PLC) de la memoria. También se observa que se tiene el selector para elegir el patrón a mostrar por pantalla.

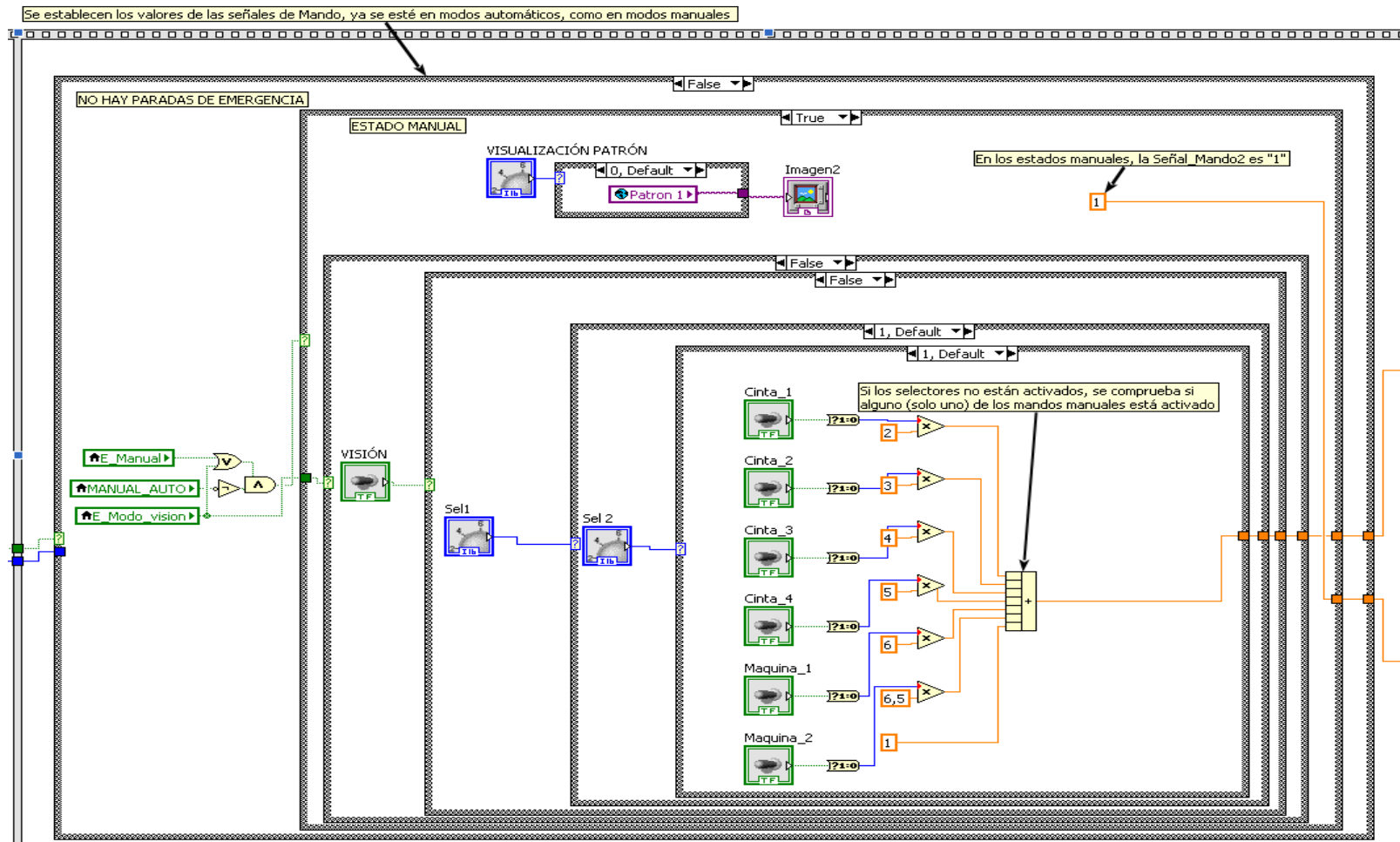


Figura B.10. Ejecución de Secuencia Tercera sin Emergencia y modo Manual.

**Secuencia Tercera sin Emergencia y modo Visión:** La figura B.11 muestra la ejecución de la tercera secuencia cuando no se ha producido parada de emergencia y se está en modo Visión. En este momento, se detiene el VI principal pasando a ejecutarse **"Menu\_Configuracion.vi"**, que mostrará por pantalla las distintas opciones de configuración como se vio en la figura 3.4 de la memoria. La configuración de este subVI y el de las opciones que presenta se verán en el apartado B.2 de este anexo.

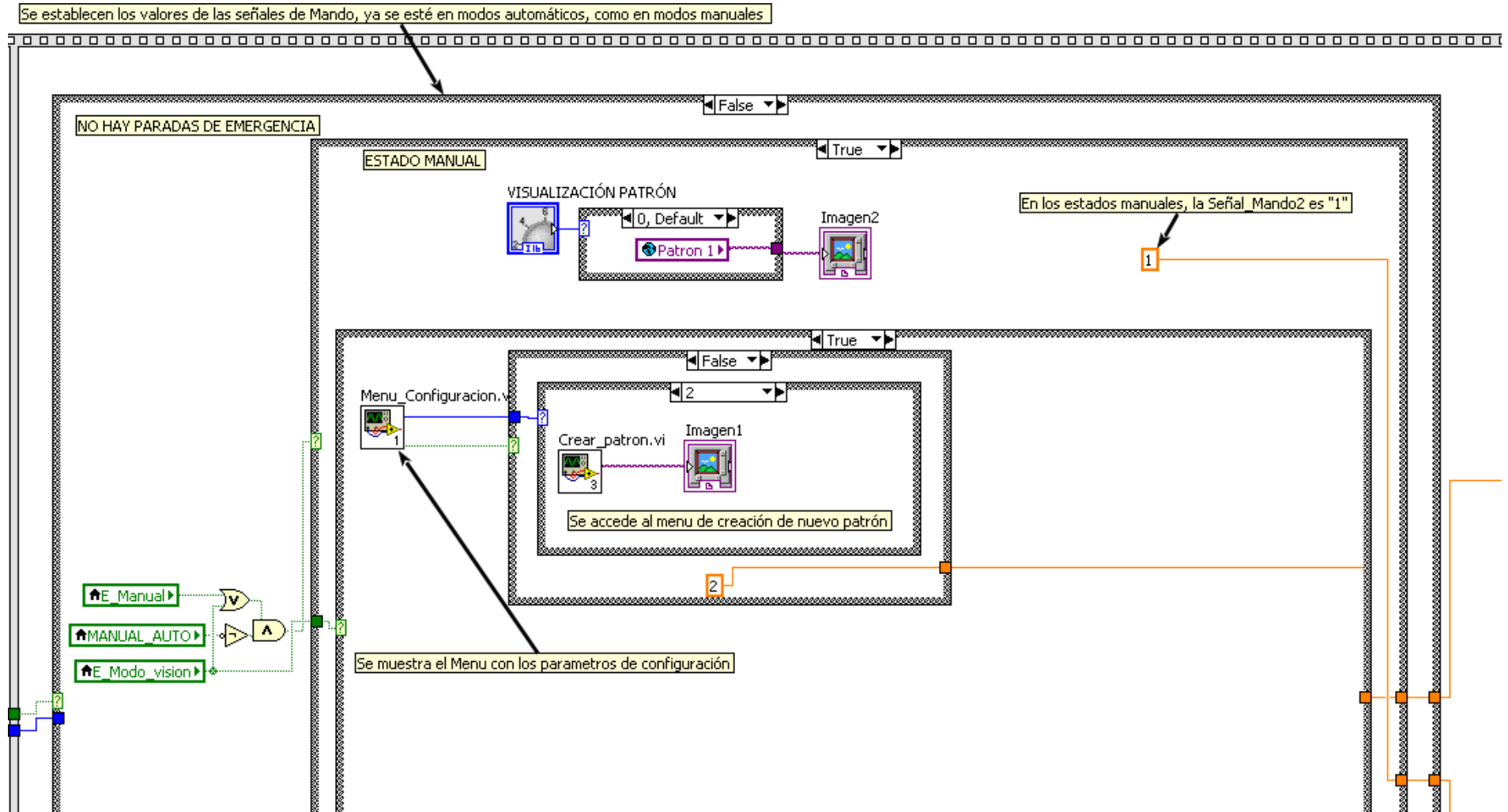


Figura B.11. Ejecución de Secuencia Tercera sin Emergencia y modo Visión.

**Secuencia Cuarta:** En realidad, la mayor parte de la programación vista en las imágenes anteriores correspondiente a la Secuencia 3 tiene como fin asignar un valor a las señales de mando, principalmente a **“Señal\_Mando”**. Se vio en el apartado 5.3.1 de la memoria que esta señal será la encargada de comunicar al PLC la activación de alguno de los elementos si el sistema está en modo Manual. También se encarga de detectar la posible pérdida de comunicación con el SCADA. Por otro lado, las señales de mando, en la etapa de **“E\_deteccion”** comunicarán al PLC el valor del tiempo que utilizarán las máquinas para la pieza que se ha detectado. Como se dijo en el apartado 5.3.1 de la memoria, la DAQ no permite enviar dos señales analógicas a la vez, de ahí la programación mostrada. En la última secuencia del **“flat sequence”** se tiene en envío de la señal que indica al PLC que la detección se ha realizado o el posible error que haya podido ocurrir durante la detección. Destacar también el control de fallos que se realiza de la DAQ.

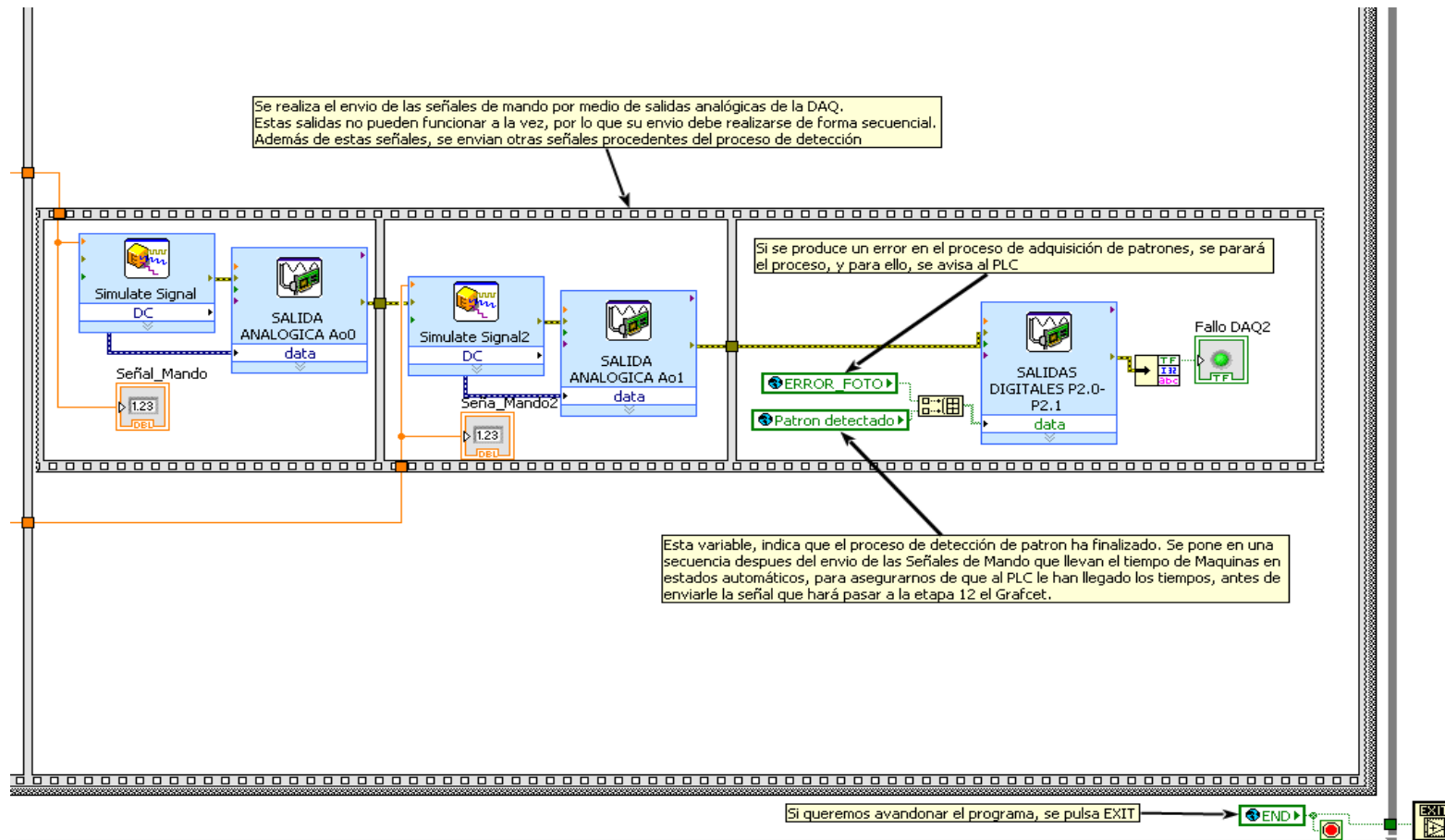


Figura B.12. Ejecución de Secuencia Tercera sin Emergencia y modo Visión.

## B.2. SubVIs utilizados para los procesos de aprendizaje y detección de patrones.

**Menu\_configuración.vi:** La figura B.13 muestra el menú con las opciones de configuración. Hasta que no se pulse una de las opciones, se repetirá continuamente el bucle. Las asignaciones de la izquierda de la figura se realizan para mostrar por pantalla los valores configurados actuales de forma correcta. La pantalla que se muestra durante su ejecución se vio en la figura 3.4 de la memoria.

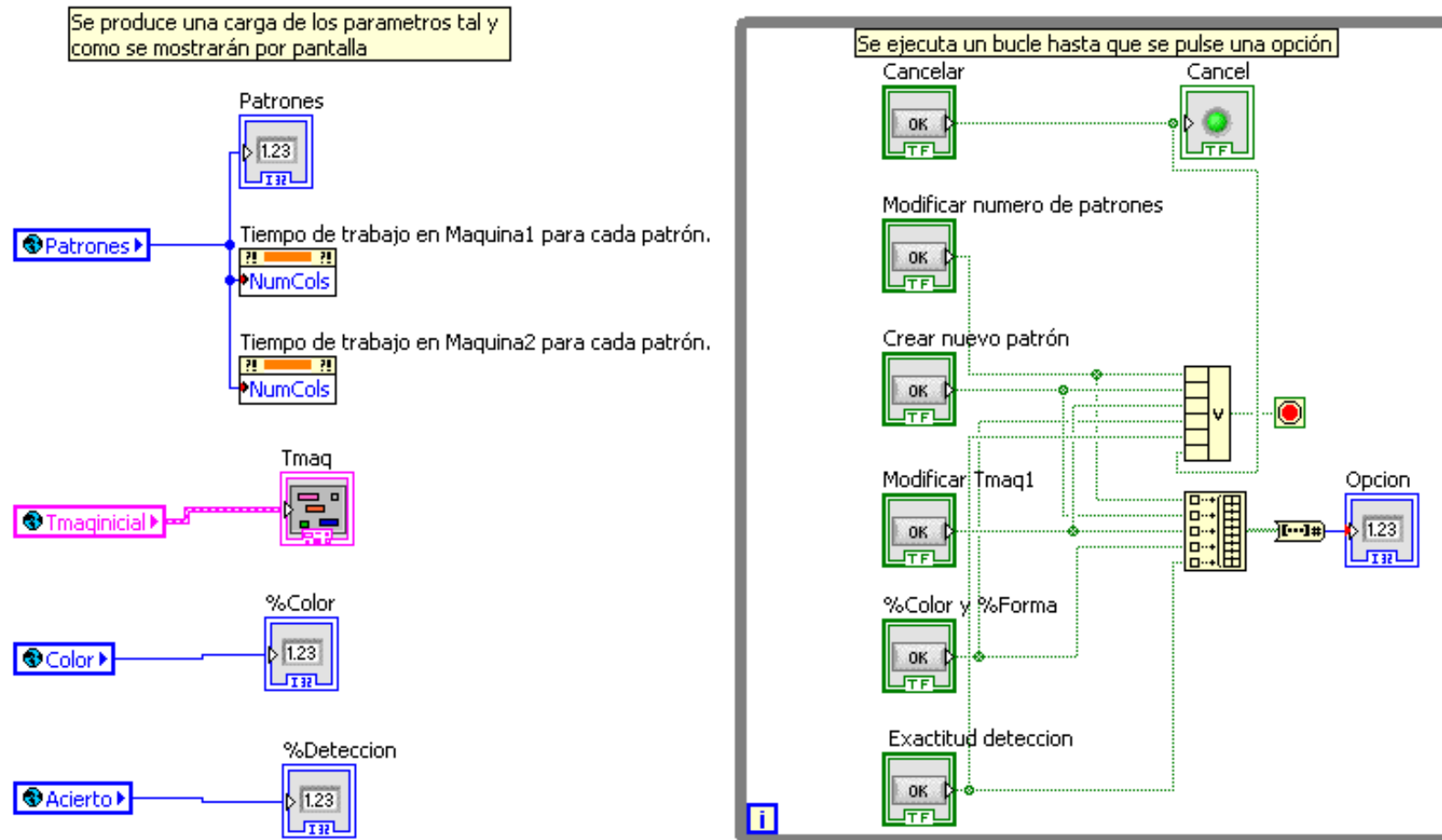


Figura B.13. Menu\_configuración.vi

**Sel\_patrones.vi:** En la figura B.14 se muestra este subVI que permite al usuario seleccionar el número de patrones con los que trabajará el sistema. La figura 3.5 de la memoria muestra el menú que le aparecerá al usuario en la pantalla. Destacar que cada cambio queda almacenado en el archivo de registro y que después de la elección del número de patrones se debe realizar un nuevo cálculo del rectángulo de trabajo como ya se comentó en el apartado 6.4.2 de la memoria. Se verá el programa de **“Rectangulo.vi”** poco más adelante.

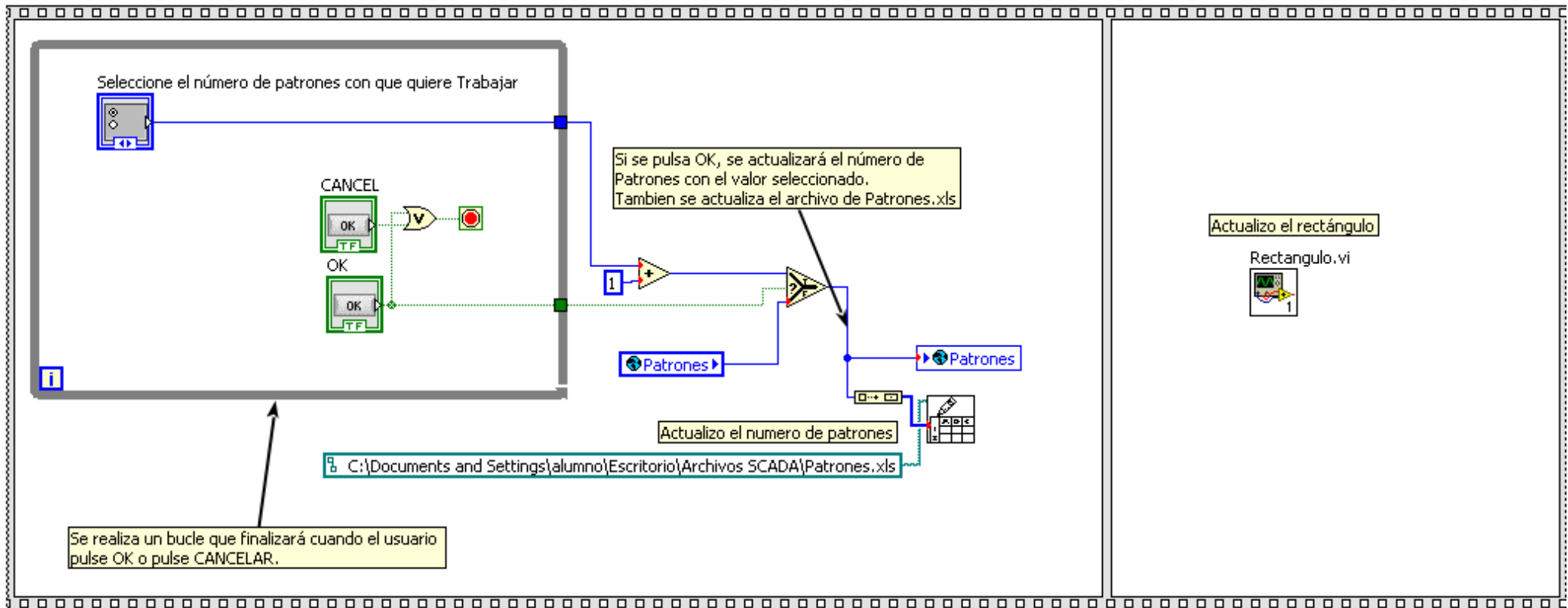


Figura B.14. Sel\_patrones.vi

**Creacion\_patrones.vi parte 1:** Como se vio en el apartado 3.5.2 de la memoria, se producen dos pasos previos, como se aprecia en la figura B.15, antes de pasar a la creación del patrón. Lo más interesante de esta figura es que en función del número de patrones que haya en el sistema, se ejecutará un determinado “*Carga\_xpatrones.vi*” donde x será el número de patrones. La figura 3.6 de la memoria muestra las ventanas que se mostrarán al usuario por pantalla.

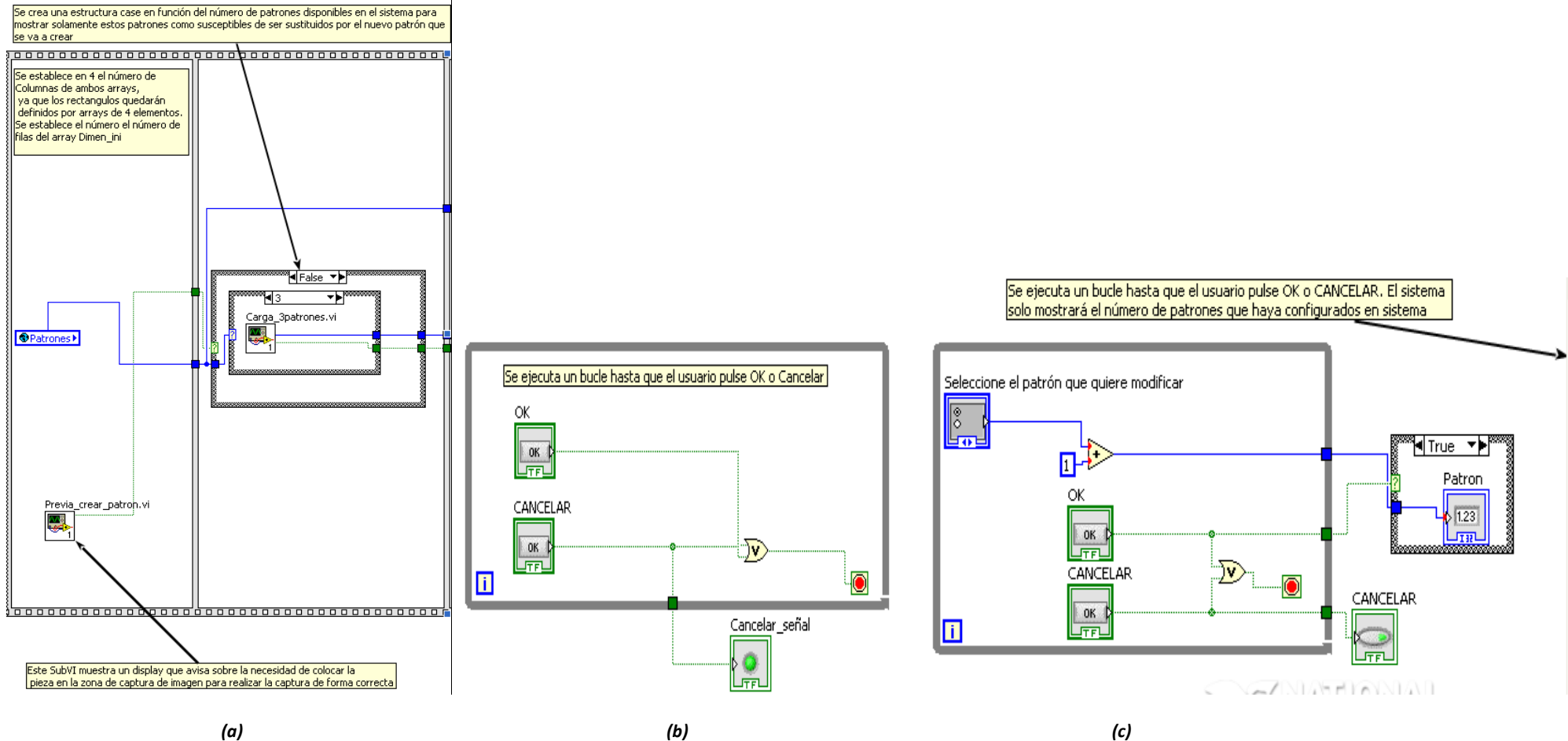


Figura B.15. Pasos previos de creación de patrones (a) Crear\_patrones.vi parte 1. (b)Previa\_crear\_patron.vi. (c)Carga\_xpatrones.vi

**Creacion\_patrones.vi parte 2:** La figura B.16 es la misma a la expuesta en el apartado 6.5.4.1 de la memoria, donde se explicó este algoritmo. Se remite al lector a este apartado de la memoria para ver su explicación.

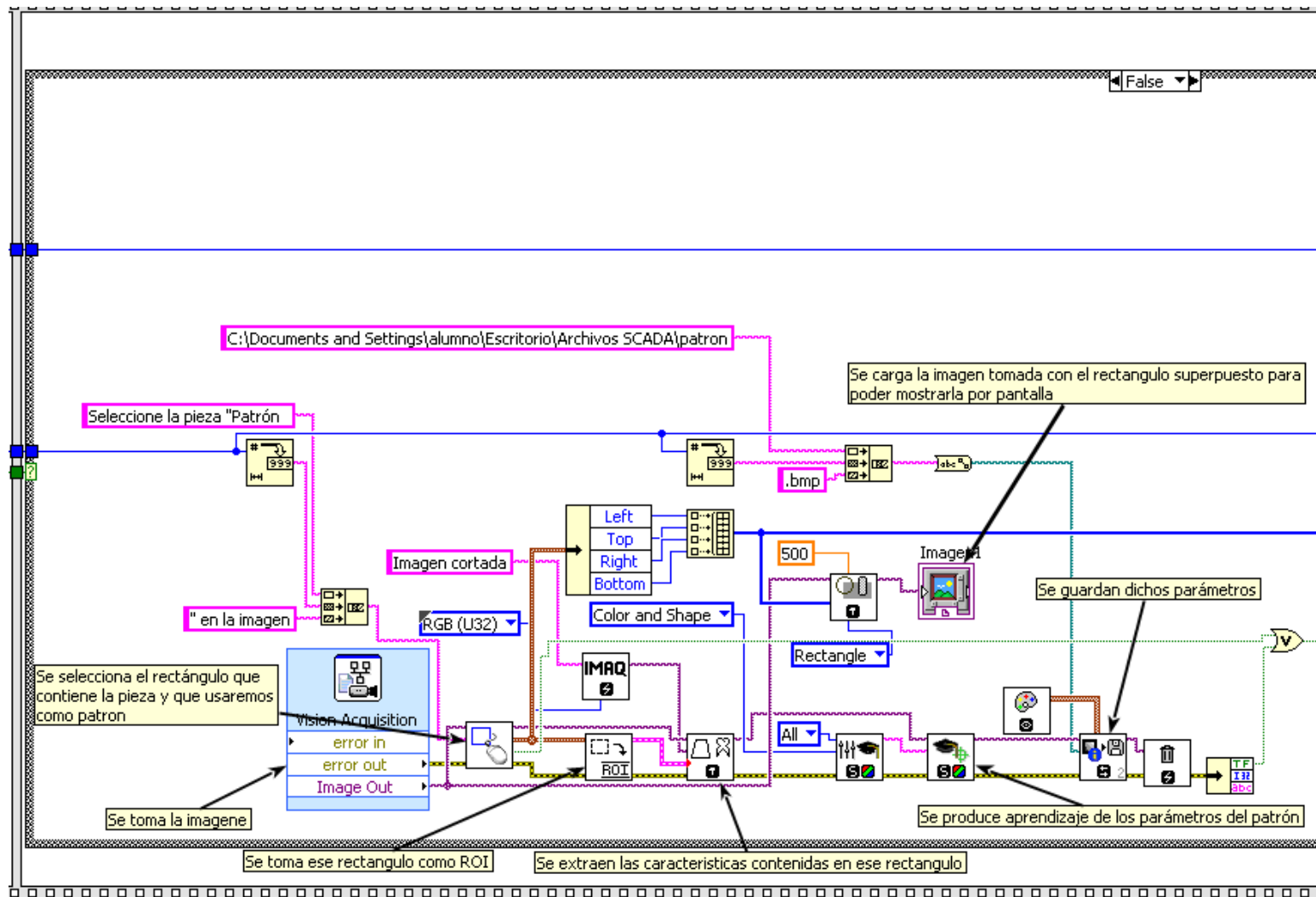


Figura B.16. Creacion\_patrones.vi parte 2



**Creacion\_patrones.vi parte 3:** La figura B.17 muestra la forma en que se almacenan las dimensiones del rectángulo correspondiente al patrón seleccionado y como vuelve a ser necesaria la ejecución de *“Rectangulo.vi”* para recalcular las dimensiones del rectángulo de trabajo. Tanto si el proceso se ejecuta correctamente como si no, se indica el resultado por pantalla.

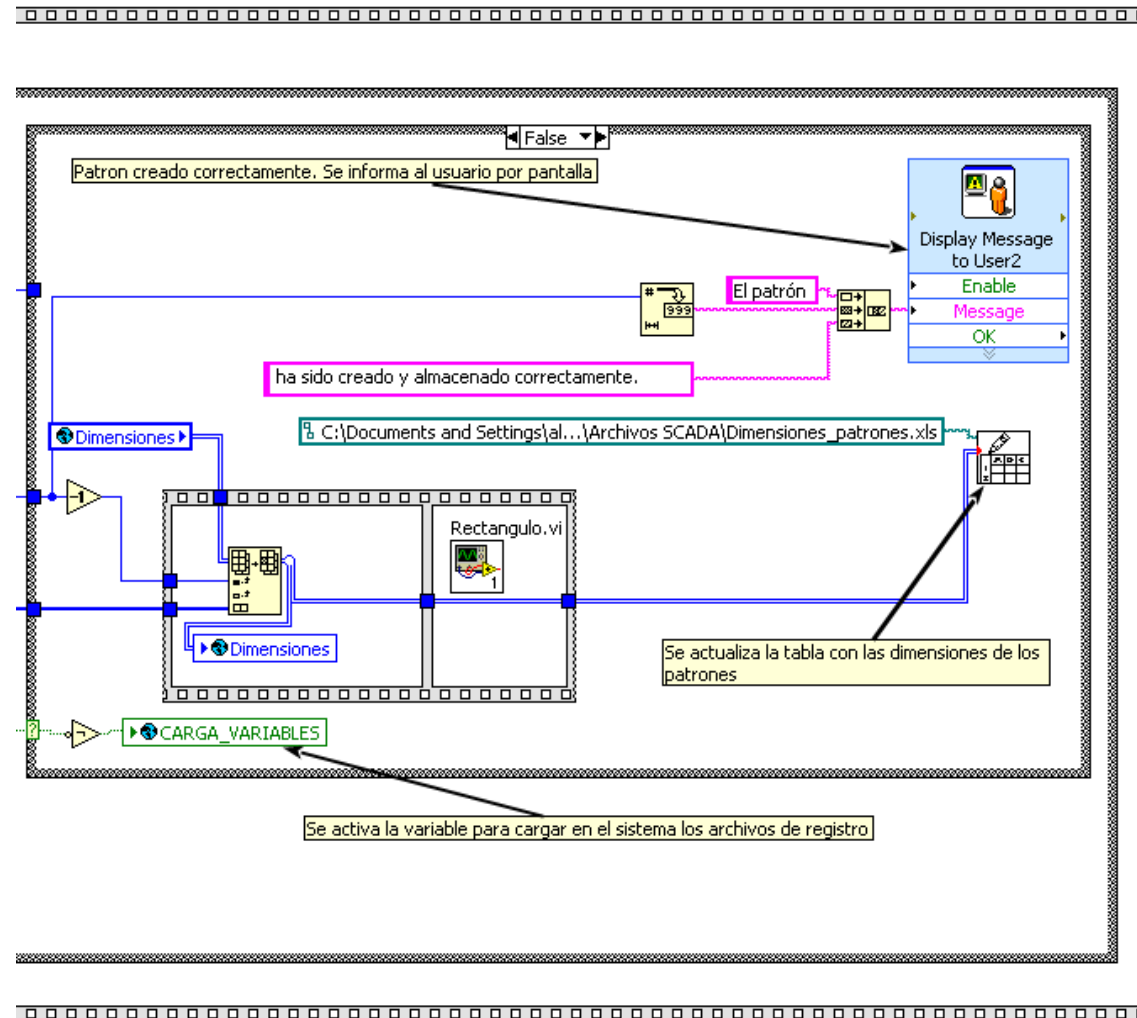


Figura B.17. Creacion\_patrones.vi parte 3.

**Mod\_TMaq.vi modificación de tiempos:** La modificación de tiempos utiliza varios SUBViS. Primero se pregunta al usuario con “Reinicio\_tiempos.vi” si desea reiniciar todos los tiempos o modificar los de un patrón en concreto. La figura B.18 muestra la programación para este subVI en el caso de que se escoja la modificación de tiempos de un patrón. Se pregunta al usuario por el patrón cuyos tiempos se variarán y después se da la opción de que elija dichos tiempos (de 0 a 9) de forma particular para cada máquina. Por último se almacenan los valores elegidos y se informa al usuario.

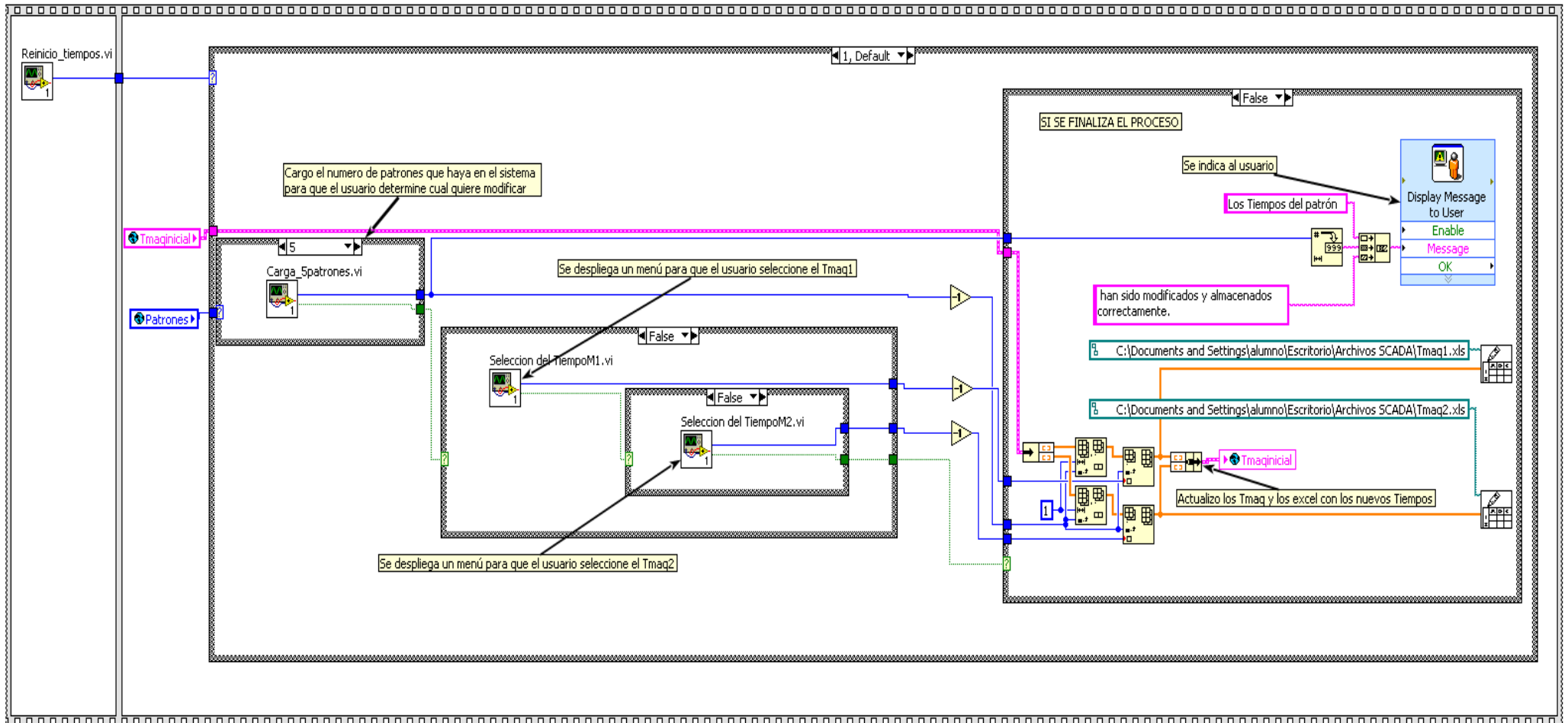


Figura B.18. Mod\_TMaq.vi

**Reinicio de tiempos:** Se produce un reinicio a 0 de todos los tiempos de trabajo de las máquinas. La figura B19.a muestra el bucle mediante el que se pregunta al usuario si desea modificar los tiempos de un patrón o restablecer los tiempos de todos. La figura B19.b muestra la forma en que se reinician todos los tiempos y se guardan en memoria y en los respectivos archivos de registro.

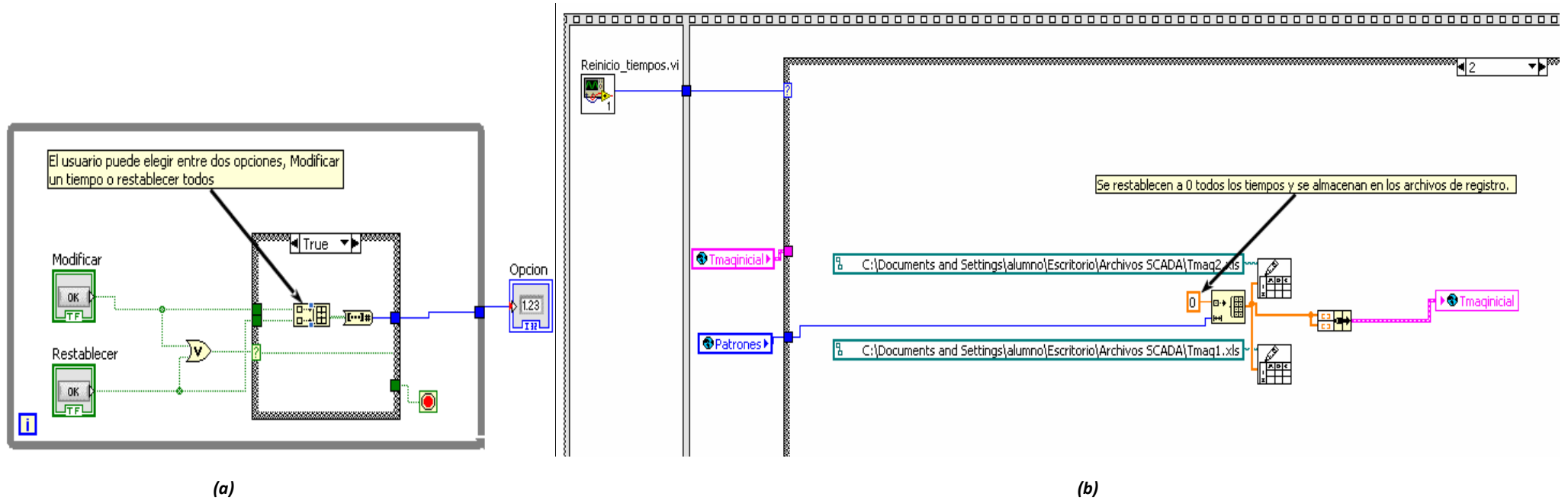
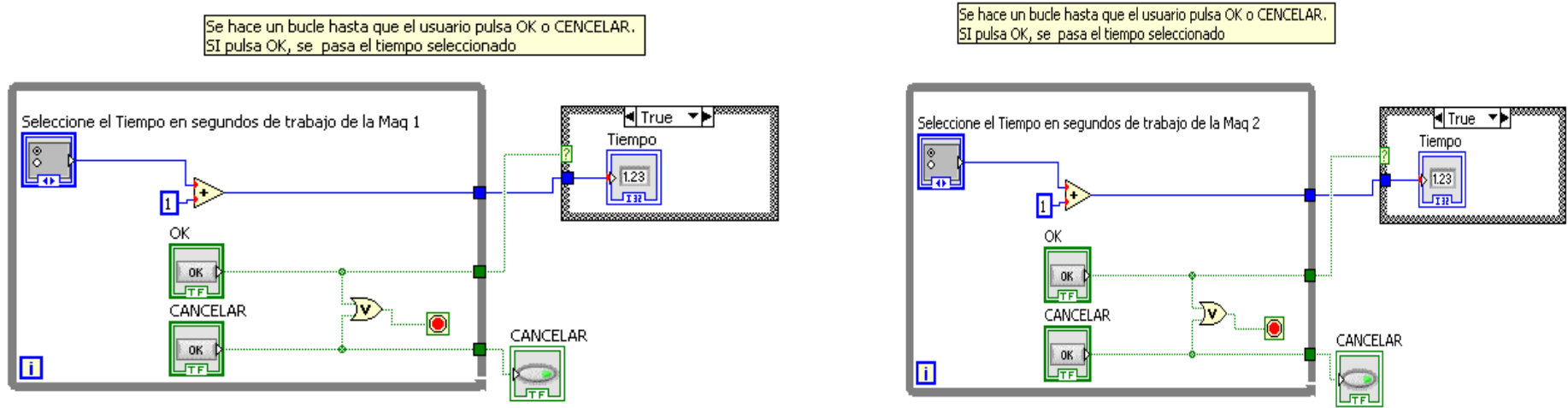


Figura B.19. Reinicio de tiempos. (a) Reinicio\_tiempos.vi. (b) Mod\_TMaq.vi reinicio de tiempos.

**Selección del tiempo Maquinas:** Como se ha mencionado, se da al usuario la opción de seleccionar el tiempo de trabajo de cada máquina para cada patrón. La programación (figura B.20) se realiza mediante un bucle que muestra casillas de tiempo de 0 a 9. Mientras el usuario no seleccione una y pulse OK o cancele la operación, se mostrará por pantalla el menú de la figura 3.7.c de la memoria para Tmaq1 o 3.7.d para Tmaq2.



(a)

(b)

Figura B.20. Selección de tiempos de trabajo de las máquinas. (a) Selección del tiempoM1.vi. (b) Selección del tiempoM2.vi.

**Porcentajes de color y score:** Se permite al usuario elegir el porcentaje de puntuación del color respecto a la forma en la figura B.21.a, mientras que en la B.21.b se permite elegir el porcentaje de score necesario para dar como válida una detección. Lo más importante a destacar de estos subVIs, es que para acceder a estos menús (vistos en figura 3.8 y 3.9 de la memoria), se debe introducir una contraseña. Esto otorga seguridad a estos parámetros debido a su criticidad en el sistema.

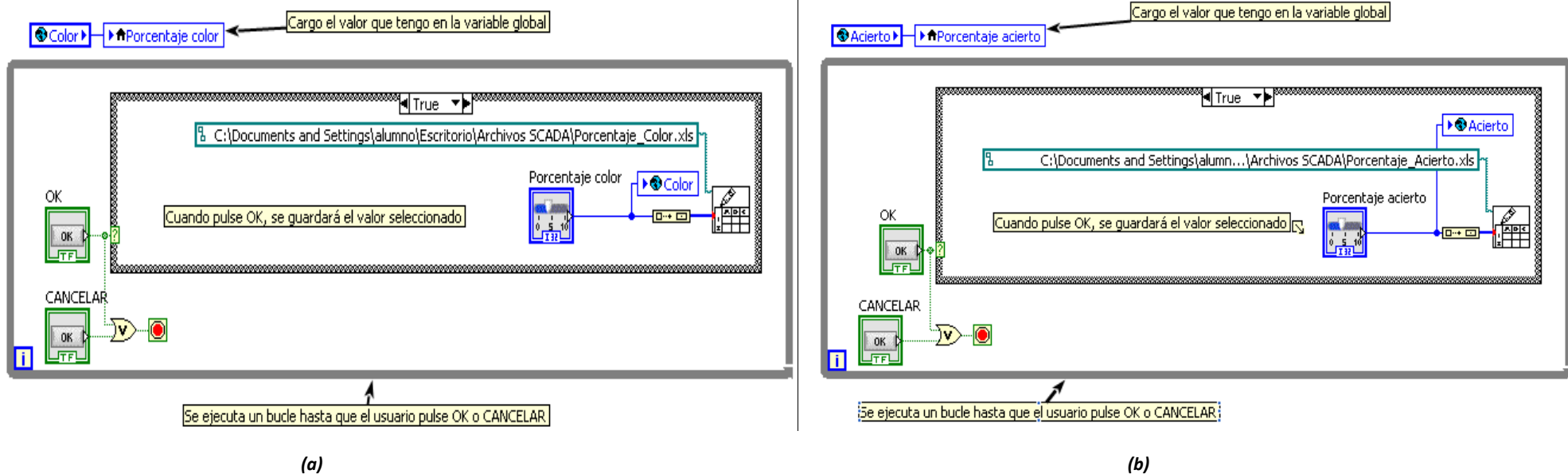


Figura B.21. Porcentajes (a) Porcentaje Color.vi. (b) Porcentaje Acierto.vi.

**Password.vi:** Como se ha mencionado, este subVI se ejecuta previamente a los mencionados **“Porcentaje Color.vi”** y **“Porcentaje Acierto.vi”**. Ejecuta un bucle con 3 iteraciones, permitiendo al usuario 3 oportunidades para introducir la contraseña correcta. En caso de intentar estas tres oportunidades con resultados negativos o cancelar, se vuelve al menú de Configuración.

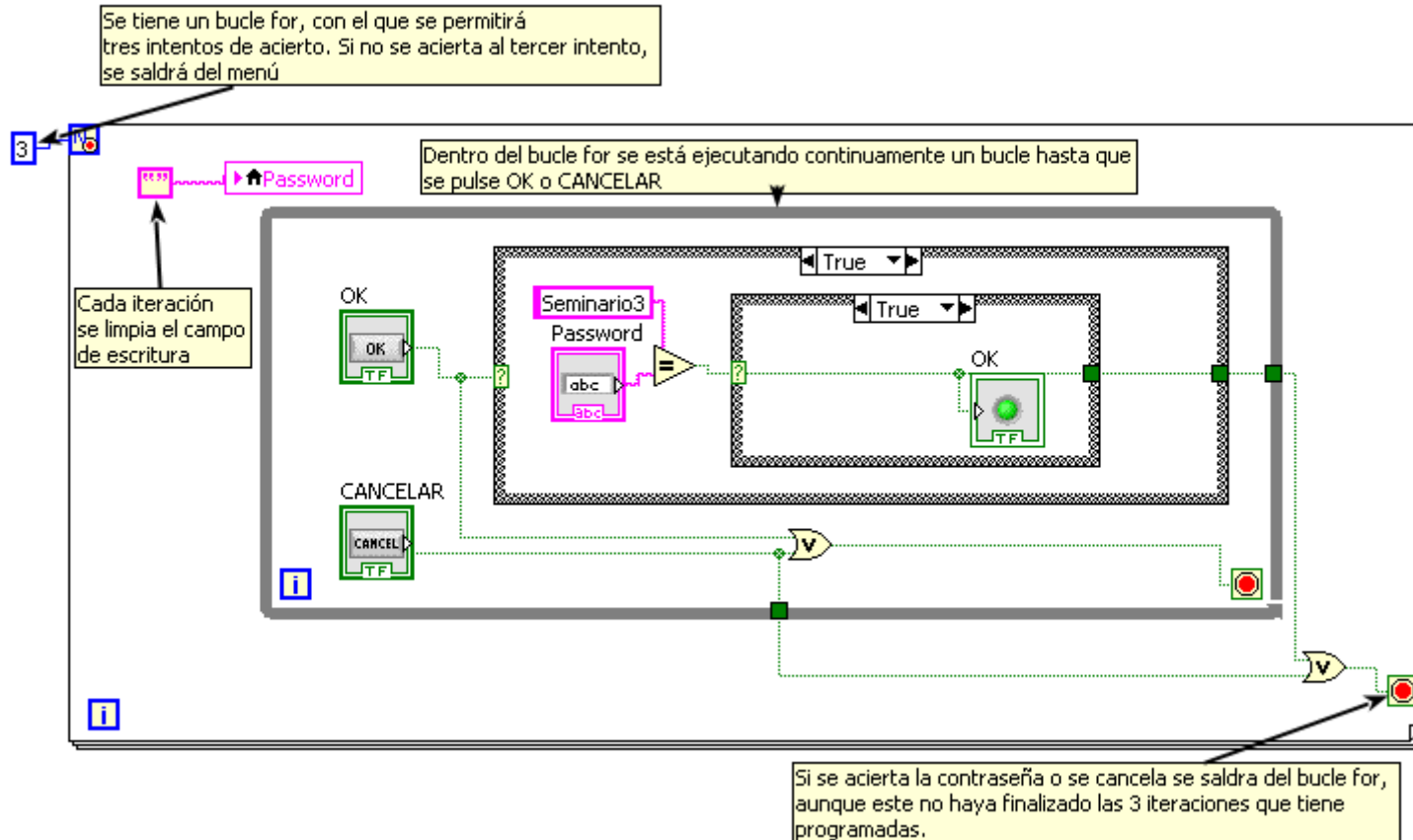


Figura B.22. Password.vi.

**Rectángulo.vi.** Se ha hecho referencia en numerosas ocasiones a este subVI, debido a su importancia en el sistema. Lo que se hace en este subVI es comparar los respectivos vértices de los patrones configurados en el sistema y elegir el mayor de ellos, para Right y Bottom, y el menor de ellos, para Top y Left, de forma que se consiguen 4 vértices de un rectángulo (rectángulo de trabajo o de referencia nombrado en multitud de ocasiones en la memoria y este anexo) que contiene a todos los patrones del sistema. Se aumenta en 10 pixeles la medida de cada vértice para evitar problemas de procesamiento.

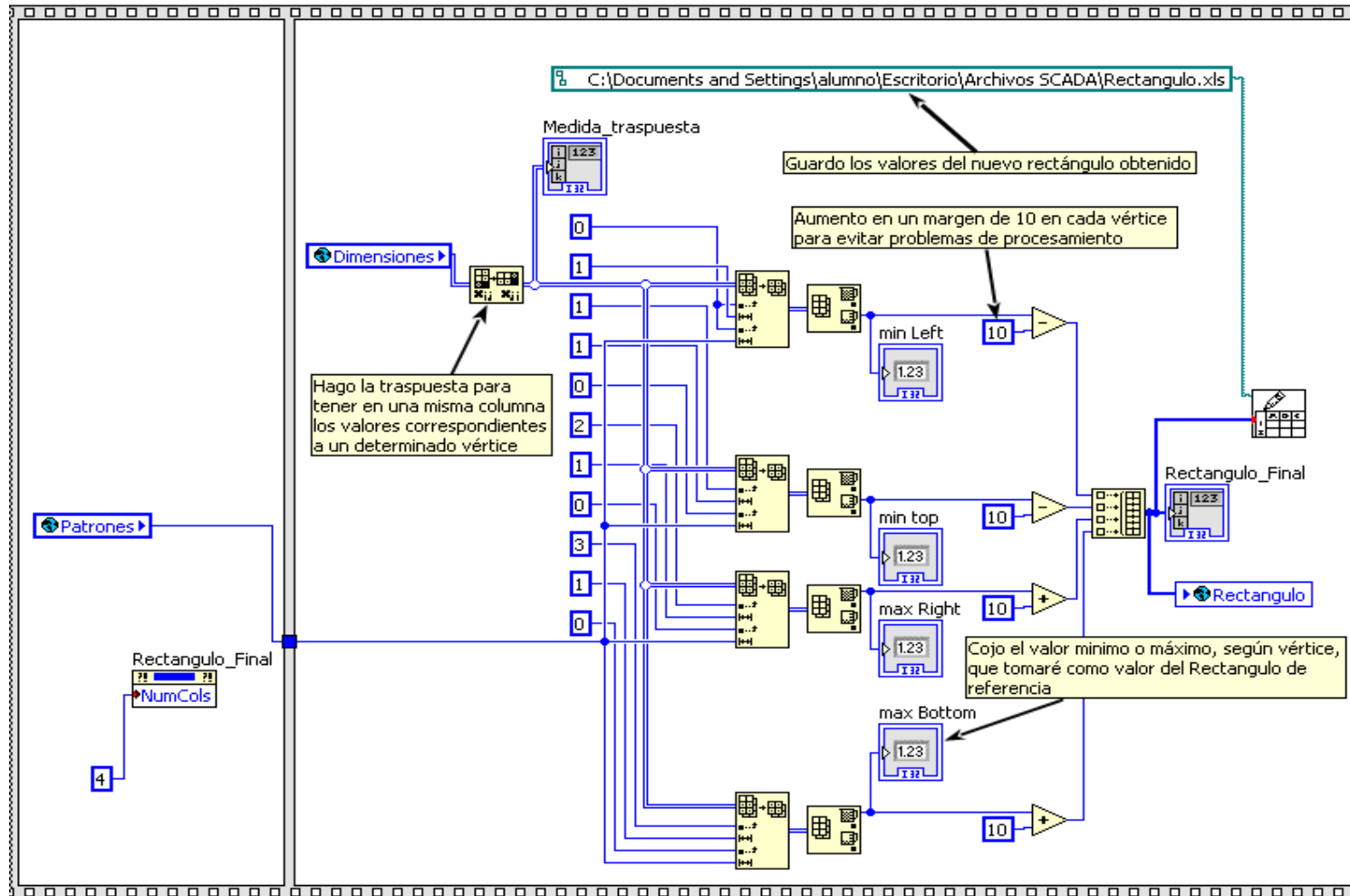


Figura B.23. Password.vi.

La figura B.24 muestra la forma en que se referencian los píxeles en Labview.

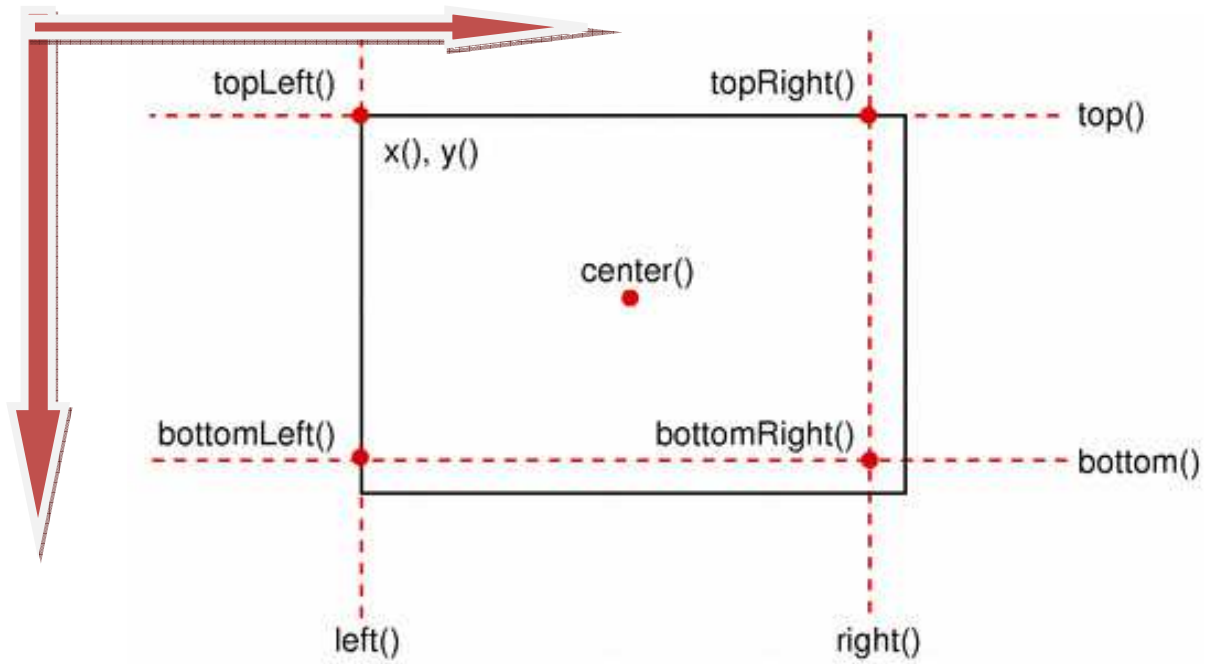


Figura B.24. Coordenadas de Labview.



**Detector\_patrones.vi parte 1.** Este subVI, de gran importancia, se ha dividido en tres imágenes. Fue explicado en la memoria en el apartado 6.5.4.2. La figura B.25 muestra el algoritmo ya explicado en dicho apartado.

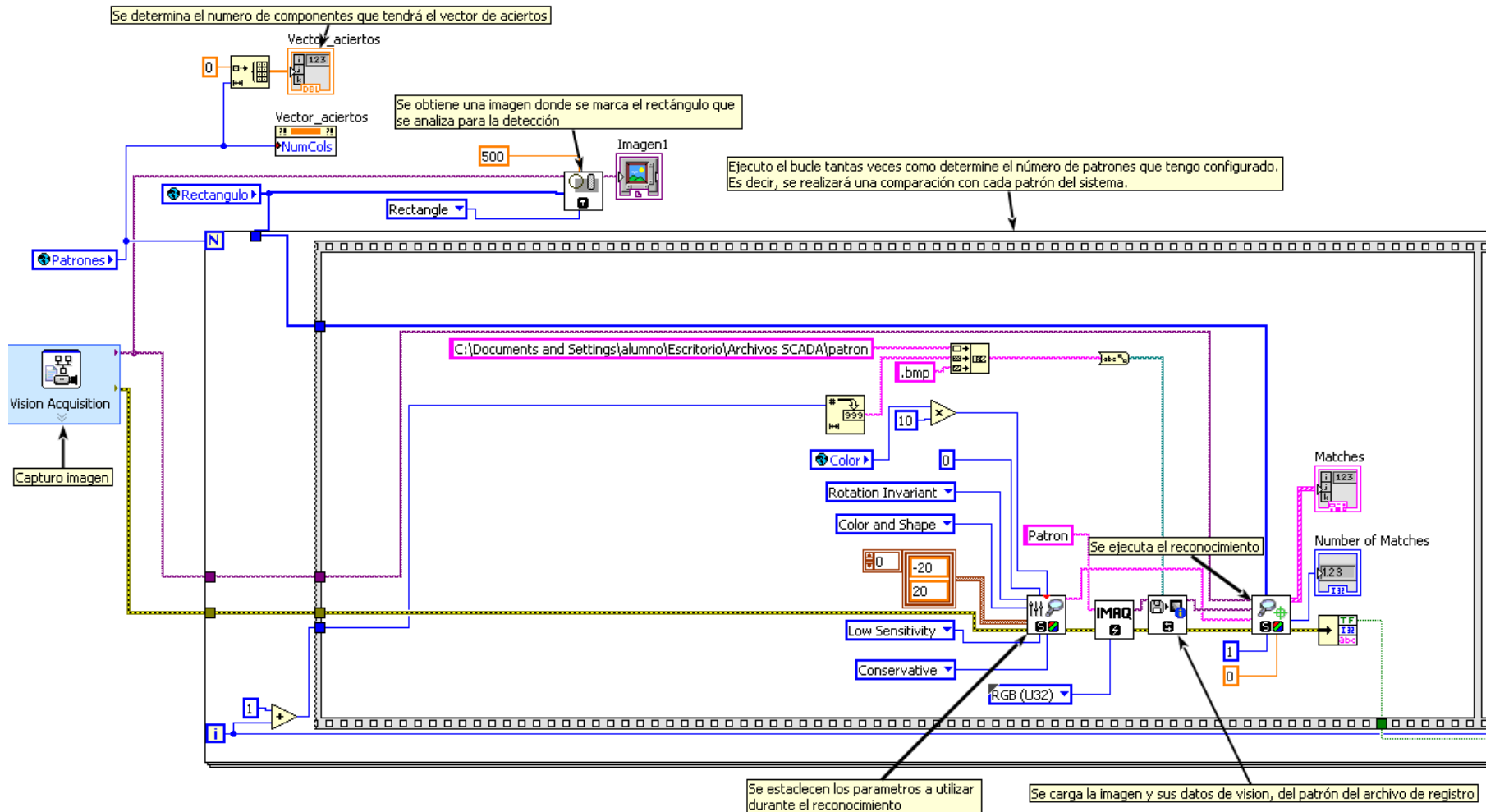


Figura B.25. Detector\_patrones.vi parte 1

Detector\_patrones.vi parte 2. La figura B26 muestra lo ya explicado en el apartado 6.5.4.2 de la memoria.

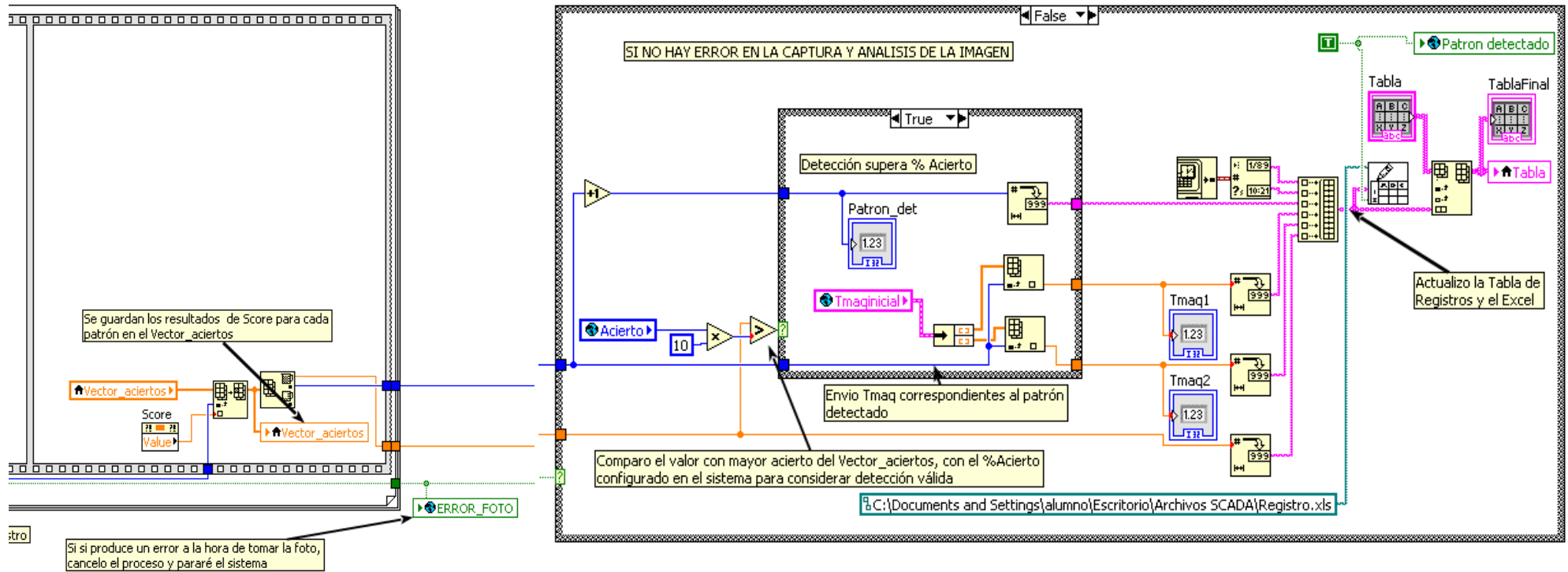


Figura B.26. Detector\_patrones.vi parte 2

### B.3. Variables globales.

En la programación se ha podido apreciar la presencia de variables globales para facilitar la transmisión de información entre subVIs. Todas estas variables han sido almacenadas en un subVI llamado "**Parametros\_globales.vi**". La tabla B.1 resume todas las variables globales utilizadas:

Variable	Tipo	Comentario
Patrones	Entero	Variable con el número de patrones que hay en el sistema.
Patron1	Imagen	Variable con la imagen del patrón 1.
Patron2	Imagen	Variable con la imagen del patrón 2.
Patron3	Imagen	Variable con la imagen del patrón 3.
Patron4	Imagen	Variable con la imagen del patrón 4.
Patron5	Imagen	Variable con la imagen del patrón 5.
No detectada	Imagen	Variable con la imagen del patrón no detectado.
Color	Entero	Variable con el porcentaje de color utilizado en la detección.
Acierto	Entero	Variable con el porcentaje de acierto necesario para una detección válida.
Dimensiones	Array	Variable con las dimensiones de los rectángulos que contienen a los distintos patrones del sistema.
Rectangulo	Array	Variable con las dimensiones del rectángulo de referencia.
Tmaquinas	Cluster de Arrays	Variable con los tiempos de trabajo en máquinas de cada patrón.
ERROR_FOTO	Booleana	Variable que indica si se produce un error a la hora de realizar la captura de imagen.
CARGA_VARIABLES	Booleana	Variable que indica que debe realizarse la carga de archivos desde la carpeta de registro.
END	Booleana	Variable que indica que se debe finalizar la ejecución del programa.
Patron detectado	Booleana	Variable que indica que la fase de detección del patrón ya ha concluido.

**Tabla B.1. Variables globales**