

# RECOPIACIÓN Y EJEMPLOS

---



# CAPÍTULO 11

## En este capítulo...

Introducción	10-2
Que es memoria V	11-2
Familiarización con <i>DirectSOFT</i>	11-4
Conexión al PLC con <i>DirectSOFT</i>	11-5
Crear o modificar un programa	11-6
Ejercicios para el nuevo programador	11-13
Resumen de las particularidades de los PLCs <i>DirectLOGIC</i>	11-15
Programación de entradas y salidas análogas con PLCs DL	11-15
Una palabra sobre números negativos	11-16
Una palabra sobre programación de PID	11-17
El PLC DL06 puede tener E/S remotas	11-17
Concepto de una instalación controlada por PLC's	11-18
Ejemplo 1: Triturador de minerales con E/S discretas	11-21
Ejemplo 2: Uso de contadores	11-50
Ejemplo 3: Uso de comparación	11-50
Ejemplo 4: Uso de una interface de operador	11-51
Ejemplo 5: Uso de E/S análogas	11-53
Ejemplo 6: Uso de comunicaciones seriales	11-55
Ejemplo 7: Uso de lazo de control PID	11-58

# Introducción al capítulo 11

En este capítulo revisaremos todos los conceptos de los otros capítulos. *Este capítulo no existe en el manual en inglés.* Una vez que se hayan repasados conceptos generales, que también se aplican a otras familias de PLCs *DirectLOGIC*, mostraremos una selección de ejemplos explicados para ayudar a entender la técnica de programación a los nuevos programadores.

Puesto que usted está leyendo este capítulo, debe significar que ha leído el resto del libro y ahora usted tiene cierta experiencia con la programación de PLCs. Pero, puede ser que no haya sido nuestro PLC con el cual usted haya trabajado. Hay algunas cosas sobre PLCs *DirectLOGIC* que son diferentes que otros PLCs, y este capítulo puede ayudarle a entender nuestra orientación. Puede ser también que haya repeticiones de conceptos.

## Que es memoria V

La memoria V en los PLCs *DirectLOGIC* son los registros, o palabras de 16 bits para todas nuestras familias de PLCs (excepto el 330/340 PLCs). Estos registros son referidos como dirección de sistema octal, es decir, no hay 8's o 9's. Por ejemplo, si usted trata de entrar la dirección de memoria V2019, usted tendrá un error.

Muchas direcciones de memoria V se pueden acceder también en los bits individuales de los cuales la palabra está constituida. El PLC DL06 puede utilizar lo que se llama **bit-of-word**, tal cual como las CPUs D0-05, D2-250-1, D2-260 o la D4-450. Esto significa que usted puede especificar el bit de una palabra particular para un dispositivo booleano, como un contacto normalmente abierto. De modo que el primer bit de V2000 sería mostrado como B2000.0.

**Nota:** El bit en la palabra no es octal, así que las opciones son B2000.0 hasta B2000.15. También, al entrar **bit-of-word** usted realmente entra V2000.0 y cuando usted presiona ENTER el código cambia a B2000.0, de modo que usted pueda decir que es una dirección de bit-of -word.

La configuración del PLC es hecha de preferencia con el programa *DirectSOFT*. Existe la posibilidad de bajar el programa desde nuestro sitio de Internet en forma gratuita, para efectos de demostración y éste trabaja normalmente con la única limitación de que sólo se puede programar hasta 100 palabras. El manual de *DirectSOFT* está incluido en el "folder" o carpeta HELP del programa de demostración y se encuentra cuando ya se haya instalado el programa.

**¡Las instrucciones por defecto en los PLCs *DirectLOGIC* usan números BCD!** El apéndice J detalla más sobre el sistema numérico BCD.

El concepto a recordar aquí es que usted debe usar instrucciones matemáticas que correspondan a sus datos. Ocurre frecuentemente que se tiene una instrucción LD, luego una operación matemática, y luego una instrucción OUT en otra dirección de memoria V. Muchos clientes llaman a apoyo técnico quejándose de que el resultado que están consiguiendo es el mismo número que fue cargado con LD originalmente. Esto es generalmente debido a que uno o más de sus números están en el formato incorrecto para la instrucción usada. En *DirectSOFT* la manera más rápida de determinar en qué formato están datos específicos de las direcciones de memoria V es abrir una ventana de **Data View** y ver el mismo dato en varios formatos hasta que



usted encuentra el correcto. Cualquier número que no sea una configuración de bits legítima en BCD será tratado como hexadecimal por el PLC para propósito de exhibición en el formato BCD/Hexadecimal.

Tenemos detallado más adelante un ejemplo para ver los datos en **Data View** en *DirectSOFT* en los varios formatos disponibles. Usted puede entrar la misma dirección de memoria V en varias líneas en **Data View** y cambiar cada uno a un formato diferente. Éste es un método excelente para entender las diferencias entre formatos.

*Los temporizadores y contadores en el PLC DL06 ven el tiempo y cuentan en formato BCD.*

El valor de referencia (Set point, en inglés) de PID, la variable de proceso y la salida son todos los números binarios/decimales.

Muchas direcciones de memoria V son direcciones dedicadas del sistema que no pueden ser cambiadas por código y tienen funciones especiales, como por ejemplo, hora y fecha. Un listado de estas direcciones de memoria comienza en la página 3-30 del manual.

### **Bits X (entradas)**

Los bits X corresponden a las entradas físicas en los módulos del PLC. Las entradas análogas pueden usar bits X para obtener datos traducidos a las direcciones de memoria V, dependiendo del tipo de PLC. Los bits X pueden también ser vistos o manipulados como palabras. Vea la página 3-32 del manual y las tablas de bits de X y de Y. Note que a la derecha de X0-X17 dice V40400. V40400 se compone de los bits de entradas X0-X17. Y también observe que estos bits son OCTALES, que no hay allí ningún ocho o nueve.

### **Bits Y (salidas)**

Los bits Y corresponden a las salidas físicas en los módulos del PLC. Éstos son iguales que los bits de X, excepto que la referencia de la dirección de memoria V comienza en V40500.

### **Bits C (relevadores de control)**

Los relevadores de control son los que utiliza el programador para los relevadores internos en sus programas. Este PLC tiene 1024 de éstos, lo que es más que suficiente para muchas de las aplicaciones. Vea la página 4-35 del manual. Aquí vemos la dirección para los bits C, notamos que son referidos en formato octal y que pueden también ser referidos por la dirección de memoria V comenzando con V40600. Esta funcionalidad es muy útil para muchas de nuestras interfaces de operador que escriben a datos a nivel de bits a las direcciones de palabra.

### **Bits SP (Estado)**

Todos nuestros PLCs, (excepto D3-330/340), comparten un sistema bastante estándar de bits de estado. Éstos tienen funciones que se extienden desde un bit siempre ON (SP1) a bits de estado para los puertos de comunicación de PLCs (SP112). Hay una lista de estos relevadores de propósitos especiales en el apéndice D de este manual.

### **Bits Tx y CTx**

Estos bits son contactos de se activan cuando los temporizadores y los contadores llegan al valor de actuación o valor predefinido. Tienen la misma definición que la instrucción de que se está manejando.

### Alimentación de energía

Le recomendamos que utilice una protección de sobretensión para el funcionamiento de la computadora con *DirectSOFT5*. Un protector de sobretensión protegerá su computadora contra la mayoría de las sobretensiones. Sin embargo una fuente de alimentación continua (UPS) proporcionará la mejor protección. Una UPS proporciona aislación completa entre la fuente de corriente alterna y la computadora y tiene reserva de batería para las condiciones de falta de energía y de baja tensión.

### Familiarización con *DirectSOFT*

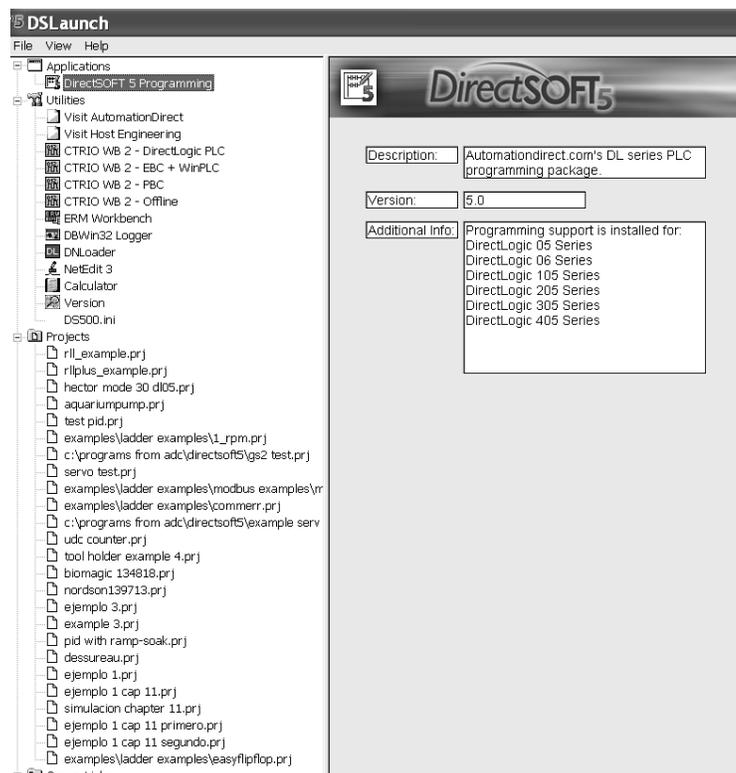
El software de programación *DirectSOFT5* funciona con el sistema de operación Windows 2000, XP y Vista y se usa para configurar el PLC DL06. Tome un momento para estudiar el manual de referencia de *DirectSOFT*. Verifique los requisitos al elegir su configuración de PC. Una vez que esté instalado el software de programación *DirectSOFT* en su computadora, usted deseará comenzar a usarlo. Le recomendamos leer el manual del programa, que está como archivo en formato pdf dentro del folder HELP en *DirectSOFT*.

Antes de comenzar a hacer o corregir un programa, usted necesita abrir *DirectSOFT*. Haga clic en Start (INICIAR) en la esquina más baja a la izquierda del monitor de la computadora. Luego vaya a **Programs**, coloque el cursor en *DirectSOFT5*, luego haga clic en **DSLlaunch** (con la figura de un cohete) en el submenú. Aparecerá la siguiente ventana de **DSLlaunch**. De esta ventana, se pueden partir utilidades adicionales, por ejemplo, CTRIO WB, ERM Workbench, etc., a partir de un lugar central. Este mismo lugar se utiliza para crear y para manejar programas del PLC y las comunicaciones entre su computadora personal y el PLC.

Note las diversas áreas en la ventana de la figura adyacente.

**Applications** Éstas son las aplicaciones instaladas actualmente en *DirectSOFT*. Son visibles en el menú bajo la carpeta (folder) o icono **Applications** y se conectan a las aplicaciones que se han diseñado para iniciarse desde *DirectSOFT*. Por ejemplo, para crear un nuevo programa haga clic doble en *DirectSOFT programming*.

**Utilities** Hay varias utilidades disponibles bajo el folder/carpeta **Utilities**. Algunas se pueden comprar en **AutomationDirect**,



por ejemplo, *KEPDirect*. Otras utilidades vendrán con el software de programación *DirectSOFT*. Estas utilidades son *ERM Workbench*, *CTRIO Workbench* y *NetEdit3*.

**Projects** Se crean proyectos en *DirectSOFT*. Un proyecto (también llamado un documento) es el nombre colectivo para su programa y toda su documentación. Cuando usted crea un nuevo proyecto, o trabaja en un proyecto existente, usted verá una enumeración en el menú bajo el folder/ícono **Projects** por nombre. Los documentos se enumeran en el orden "más recientemente usados".

**Comm Links** Los "enlaces" o links sirven para establecer parámetros para los puentes de comunicaciones entre su PC y uno o más PLCs. Los enlaces no son solamente para los programas de control. Realmente son puentes de comunicaciones (es decir, el enlace entre la computadora y la impresora). Cualquier aplicación puede usar el enlace. Cuando usted crea enlaces, aparecerán en el menú bajo el folder o ícono **Comm links** (enlaces de comunicación).

### Conexión al PLC desde una computadora con *DirectSOFT*.

Normalmente Ud. se conectará a una computadora con *DirectSOFT* con un cable serial D2-DSCBL, conectado al puerto 1 del PLC. (Puede ser hecho también a través de Ethernet). El switch de selección de modo en el PLC debe estar en **TERM**.

Para poder establecer la comunicación debe usar el **Link Wizard**, que se activa desde el diálogo **DS Launch**, haciendo clic en el archivo **COMM LINKS**, como se muestra en la figura adyacente.

Al hacer esto, aparece un pequeño mensaje que dice **ADD LINK** (que significa Agregue una conexión). Haga clic nuevamente sobre él y aparecerá el diálogo de la figura adyacente.

Esto le permite seleccionar el puerto de comunicación de la computadora. Normalmente es **COM1**. Luego de seleccionarlo, haga clic en **NEXT>**.

Si usa un adaptador de USB, puede ser que obtenga otro puerto en la computadora. Vea que es necesario hacer, en ese caso, en el manual del programa *DirectSOFT*.

Luego viene otro cuadro de diálogo, que le permite seleccionar la familia del PLC.

Seleccione el grupo de familias **DL 0/1/2/4/350 only** y luego haga clic en **NEXT>**.



Aparecerá el próximo cuadro de diálogo que le permite seleccionar el protocolo y la dirección del nodo.

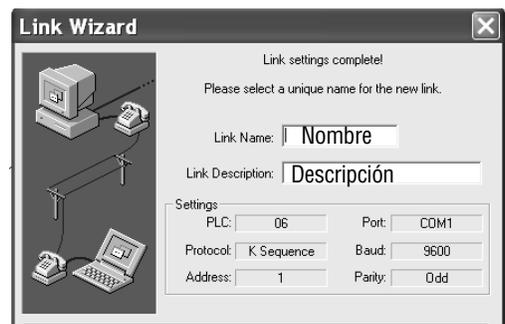
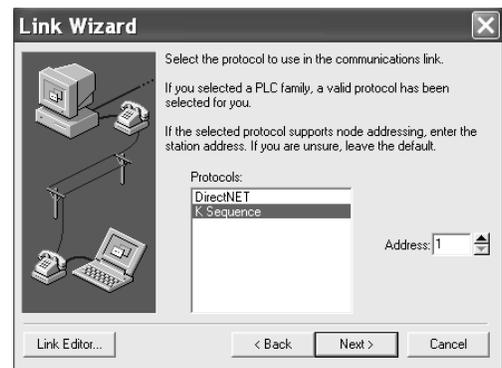
Seleccione **K-sequence** en este caso. Use el nodo 1 que es el valor por defecto y luego haga clic en **NEXT>**.

Esto hará que la computadora encuentre el PLC. Si hay errores, la computadora le informará sobre eso. Vea más detalles en el manual de *DirectSOFT*. Normalmente no hay ningún problema de conexión.

Por último, coloque un nombre al enlace como se muestra en la figura adyacente.

Cada enlace debe tener un nombre único. Puede tener hasta 16 caracteres. Haga clic en el botón **FINISH** para volver a la ventana de *DSLlaunch*. Es creado un archivo con el nombre que Ud. asignó, y queda debajo de la carpeta **Comm Link**.

Al lado izquierdo del archivo aparece una luz, que puede ser verde, amarilla o roja. Si está verde, el enlace está activo.



### Comenzar a crear o modificar un programa

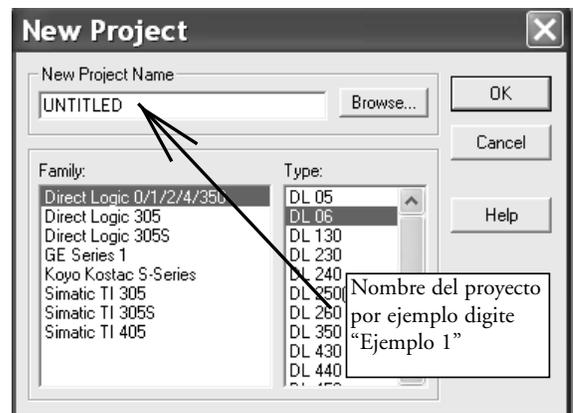
Los pasos siguientes le mostrarán los pasos básicos para modificar programas con *DirectSOFT*. Esto le dará los fundamentos para comenzar a usar de modo que usted pueda modificar un programa.

#### Paso 1: Entre en el modo Program

Para comenzar el programa *DirectSOFT*, haga clic en el icono *DSLlaunch5*; aparece un diálogo cuya parte izquierda muestra una lista de carpetas; Para comenzar un nuevo programa (proyecto), haga clic doble en *DirectSOFT Programming*, situado en la carpeta **Applications** del menú.

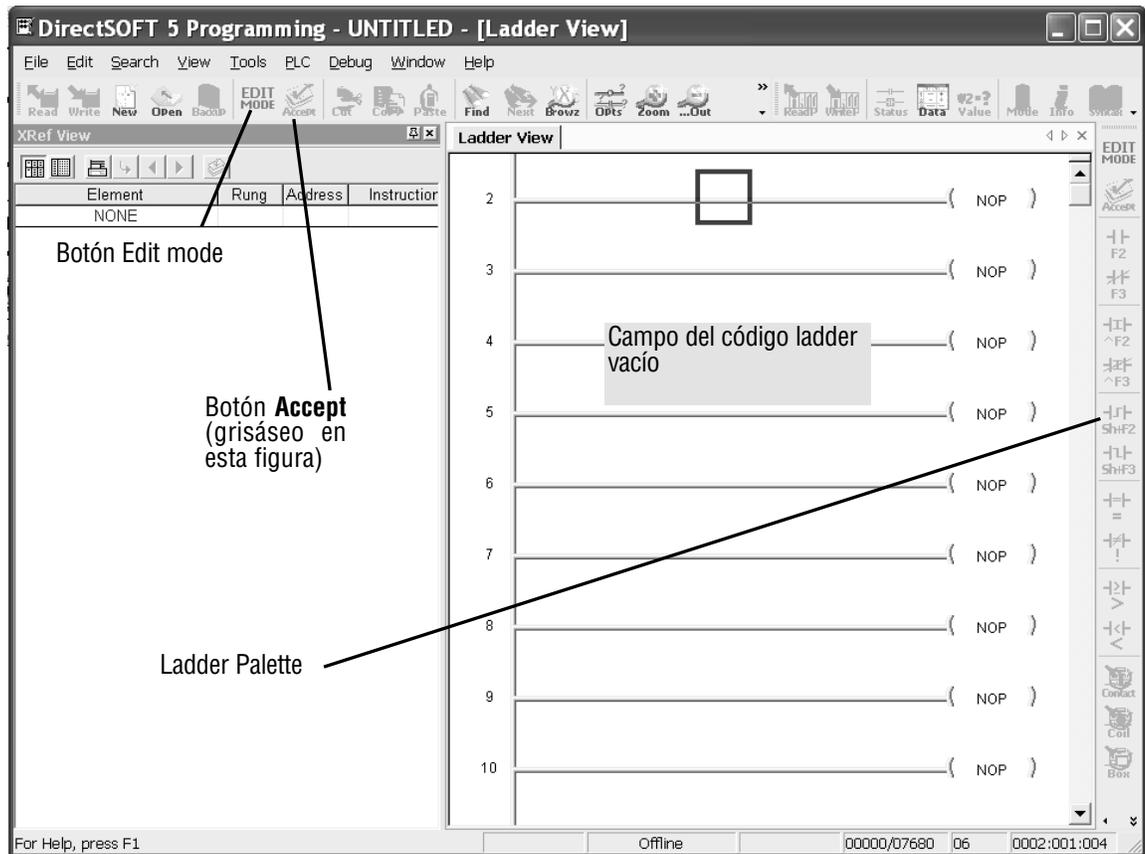
#### Paso 2: Comience un nuevo proyecto

Usted debe ahora ver la ventana **New Project**. Usted puede nombrar un proyecto usando cualquier combinación de 15 caracteres (incluyendo espacios). "EJEMPLO 1" es el nombre del proyecto a ser usado para este ejemplo. Mueva la barra de selección a la **PLC Family** (familia del PLC) y al **CPU Type** (tipo de CPU). Haga clic en **OK** después de usted ha hecho sus selecciones de familia y tipo. Para este ejemplo, use el grupo de PLCs que pertenece a las familias DL05/06/105/DL205/DL405. Tenga presente



que las mnemónicas disponibles, las reglas de proceso e incluso las características de la barra de la herramienta se adaptan a la selección de familia y tipo que usted hizo.

Después de hacer clic en OK para entrar el nombre del proyecto, usted verá los renglones de lógica ladder listos para ser modificados, tal como en la figura de abajo.



Éste es solamente el modo de leer en este punto (pero no modificar). En este modo, el cursor se presenta siempre vacío y no se puede hacer programación. Solamente se puede observar un proyecto. Si usted es programador "con experiencia", puede ser que no le guste la forma de presentación.

Esto sería un buen momento para seleccionar las opciones de colores. Por favor, vea al manual de usuario de programación del software *DirectSOFT*, para configurar el aspecto de la ventana de programación.

### Paso 3: Vaya al modo EDIT

El modo de modificación o **Edit Mode** se utiliza para escribir el programa de control. Usted tiene la opción de entrar el modo de modificar el programa de tres maneras, siendo lo más común hacer clic en el botón **Edit Mode** en la barra superior de herramientas. Será un botón con letras rojas y fondo transparente. Otra manera de cambiar el modo de modificación es hacer clic en la barra de menú superior en **EDIT** y luego seleccione **Edit Mode**. La última manera de entrar en el modo de corrección es apretar las teclas **Ctrl + E** (presione las teclas control y E simultáneamente).

*DirectSOFT* indicará el modo de modificación activo (Edit mode) cuando la caja del cursor se convierte en sólida y el botón **Edit Mode** se hace blanco. La herramienta **LADDER PALETTE** también aparecerá en la parte lateral de la ventana de programación, la cual muestra símbolos de programación, tal como un contacto normalmente abierto.

### Paso 4: Usando la Ladder Palette para entrar el primer elemento en el renglón

La **Ladder Palette** puede ser muy útil, especialmente al comienzo mientras aprende a usar el programa ladder con *DirectSOFT*. Más adelante, usted puede preferir utilizar las teclas especiales (**Hot Keys**). Estas se muestran en cada botón de símbolos y aparecen siempre cuando el cursor está en el botón de símbolos (por ejemplo, F3 coloca un contacto normalmente cerrado).

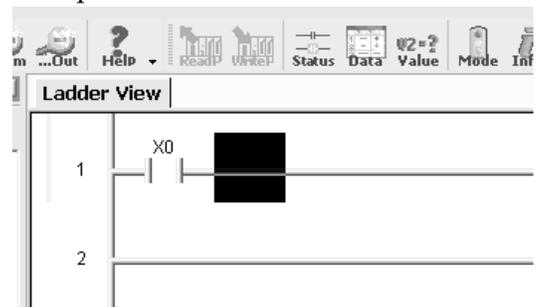
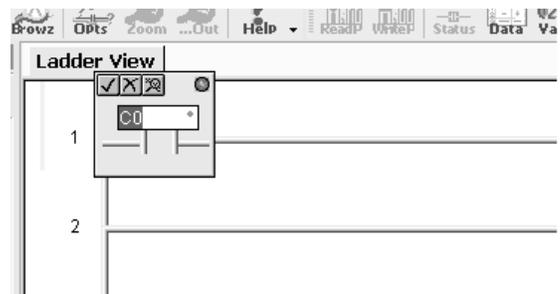
Vea el manual de usuario de programación del software *DirectSOFT* para más detalles. Use la **Ladder Palette** para incorporar la primera instrucción del programa. Primero, mueva el cursor a la localización deseada para el primer elemento. Esto se hace con el mouse o con las flechas en el teclado hacia arriba y hacia abajo. Al usar el mouse, coloque simplemente la flecha del mouse en el punto donde usted quiera que el elemento sea puesto y haga clic el botón de mouse izquierdo.

En este ejemplo, será colocado un contacto normalmente abierto en la primera posición respecto al renglón 1. Coloque el cursor al principio del renglón y haga clic en el símbolo “contacto normalmente abierto” en la **Ladder Palette**.

### Paso 5: Entrando entradas del tipo X

Usted verá que el cursor cambia a una caja de diálogo con un contacto abierto, una ventana con el cursor de texto que centellea en la dirección **C0** (destacada) y un indicador verde. Si el punto verde del elemento cambia al rojo, significa que la dirección es incorrecta, inválida o un carácter incorrecto. Por ejemplo, si usted teclea la letra **O** en vez del dígito **0**, el indicador se torna rojo y permanecería rojo hasta que usted corrija su error. Entre **X0** mientras **CO** está destacado. Después de que haya entrado la dirección y el indicador de error es verde, haga clic en la marca de verificación (✓) o presione la tecla **ENTER**.

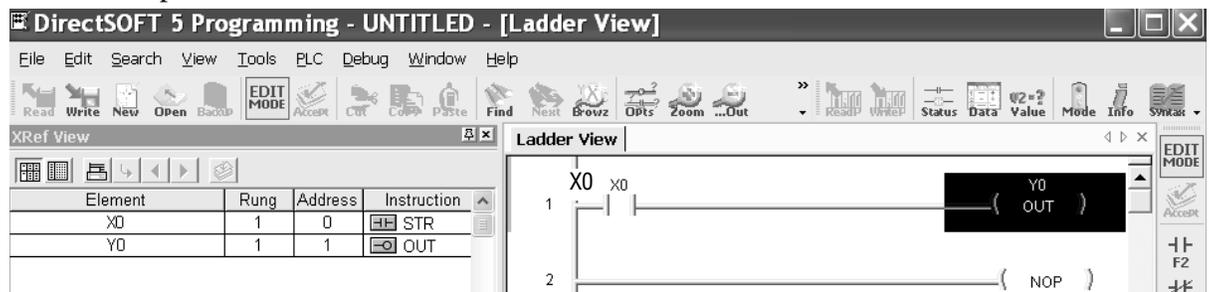
Se ha entrado la instrucción y el cursor se ha movido a la posición siguiente de entrada. Note la barra vertical amarilla que aparece al lado del renglón. Puede ser que éste no sea un manual en colores, pero se ve una barra vertical coloreada en el ejemplo de la pantalla. La barra amarilla indica que se ha entrado una instrucción o instrucciones, pero que el programa no se ha aceptado (compilado) de modo que usted pueda salvar el programa revisado al disco duro. Los renglones que se han aceptado ya en memoria compilada tendrán una barra verde en lugar de amarilla. Sin la compilación, usted no verá los iconos para **Save to Disk** (Salvar al disco duro) o **Save to PLC** (Salve al PLC).



Esto significa que para salvar su programa en el disco o en el PLC, usted tendrá que compilar el programa apretando **ACCEPT** primero. Por ejemplo, si usted desea parar de trabajar *DirectSOFT* ahora, usted primero debería compilar todos los renglones

### Paso 6: Entrando salidas del tipo Y

Después, mueva el cursor al extremo del renglón, usando las teclas **CTRL + la flecha ->** del teclado, sobre la instrucción **NOP**. Haga clic en el botón **BROWSE Coils** (Navegar las bobinas) en **Tool Palette**. El **instruction browser** aparecerá con la bobina estándar seleccionada por defecto. Haga clic en **OK** para entrar una bobina estándar.

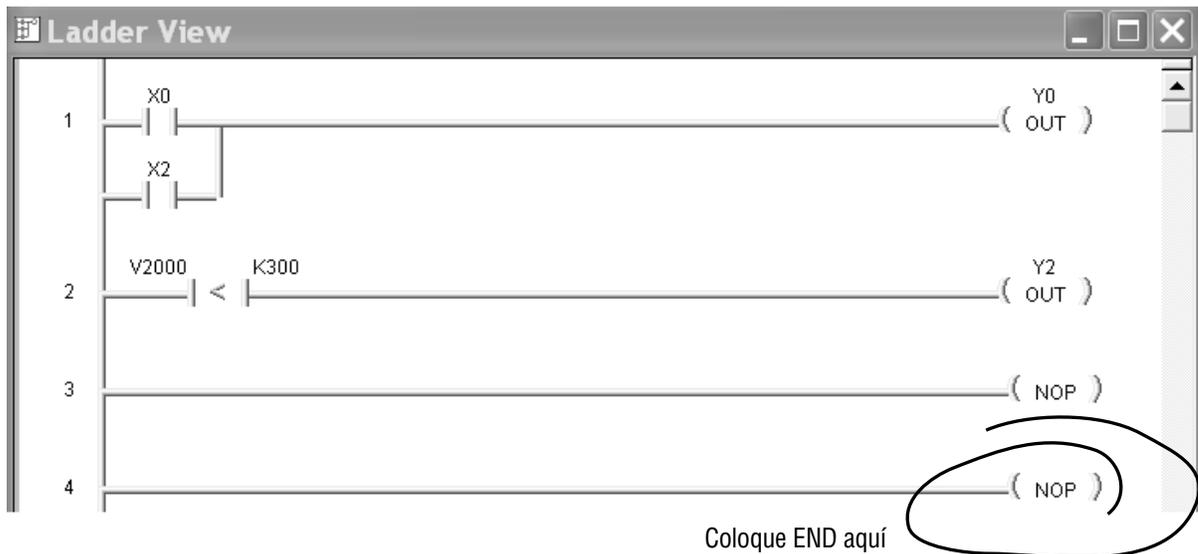


### Paso 7: Cuadro de diálogo de entrada del elemento

El navegador de instrucciones será substituido por la caja de la entrada del elemento. Note que la instrucción se llama **OUT**. Esta es la bobina **OUT** (Hay otra instrucción **OUT**, que es de caja). La dirección por defecto, **C0**, estará destacada. Teclee **Y0 > Enter**. Cuando la dirección es entrada correctamente, el indicador de error estará verde. Si quiere agregar otro contacto en paralelo con el primero, apriete la tecla **Enter** y se abrirá un espacio debajo de este renglón. Allí Ud. puede colocar otro contacto, como **X2**, de la misma forma. Luego cierre el circuito usando las teclas **CTRL + flecha para arriba** simultáneamente. Acaba de programarse el renglón 1. Ud. puede programar cuantos reglones sean necesarios para su programa. Apriete el botón **Accept**. De la misma forma se programan instrucciones caja, tales como **LD** o **MUL** u otras.

Digamos que quiere ahora entrar una comparación de un valor en **V2000** con una constante en el renglón 2. Ud. va colocar el cursor en el segundo renglón y hacer un clic en la **Ladder Palette**, en el símbolo de **menor que** (**-|<|-**) o apretar la tecla **<**. La caja de diálogo le presenta dos campos; en el primero Ud. coloca **V2000** (donde dice **TA0**) y en el segundo campo coloque **K300**, siendo **300** la constante. Mueva el cursor a la derecha y allí puede colocar una bobina **Y1**, por ejemplo, haciendo clic en donde dice **Coil**. Nuevamente aparecerá el diálogo **Instruction Browser**. Esta instrucción compara el contenido de **V2000** con la constante **3000**. Si la comparación es verdadera, la bobina **Y1** será activada.

Estos renglones se podrían descargar o bajar al PLC, excepto que falta un elemento. El programa **siempre** se debe terminar con un renglón con la bobina **END**. de otra forma, tendrá un error al compilar. Vea como queda la programación en la figura de la próxima página, antes de colocar la instrucción **END**.



### Paso 8: Entre la bobina END

Para programar este renglón, mueva el cursor de modo que quede sobre la instrucción **NOP** en el renglón 4, y haga clic en el botón **COIL**. Aparecerá la ventana **Instruction Browser**. Mueva las flechas hacia arriba o hacia abajo o use el mouse para seleccionar **Program Control** en la sección de **COIL CLASS** (clase de bobina) de la ventana. **END** estará al comienzo de la lista de las bobinas y estará destacado. Haga clic en **OK** para entrar el elemento. Acepte el programa. Hay ahora varios renglones programados en este ejemplo. Este programa se puede descargar a un PLC de la forma que está o, si desea, se pueden agregar renglones adicionales al programa. La bobina **END** debe estar al final del programa. Continúe practicando lo que se ha discutido antes de continuar. Esta es una de las formas de escribir el programa en *DirectSOFT*.

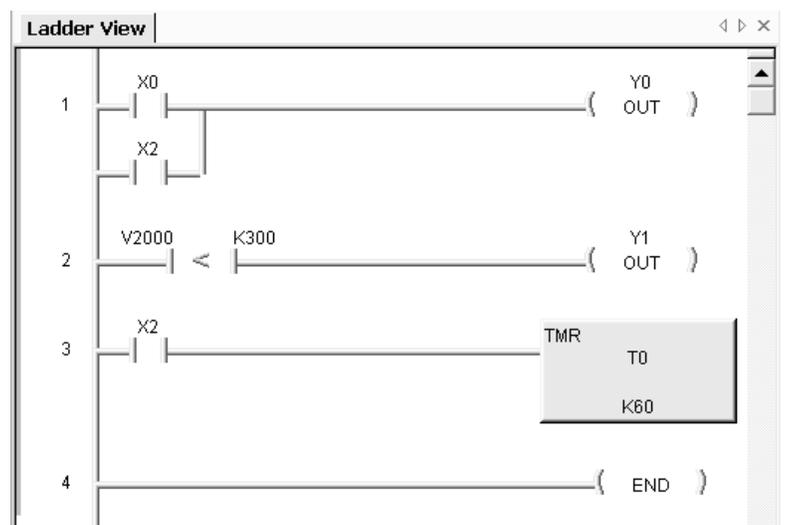
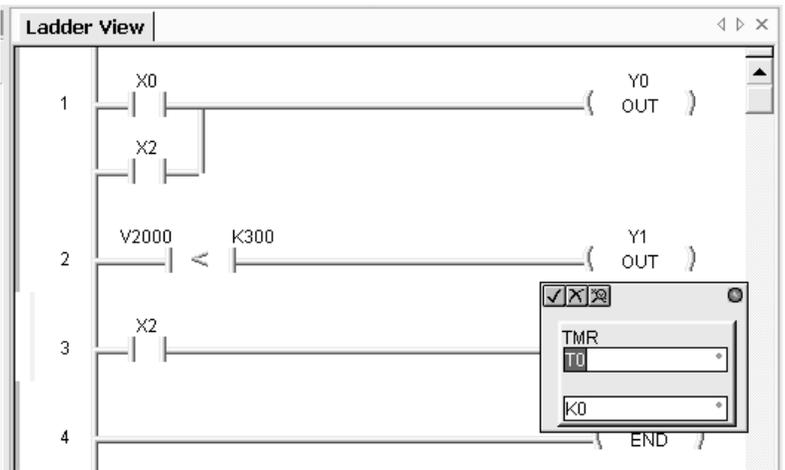
### Paso 9: Entre un temporizador

En este ejemplo, será colocado un contacto normalmente abierto en la primera posición del renglón 3, como condición para activar el temporizador. Coloque el cursor al principio del renglón y haga clic en el símbolo “contacto normalmente abierto” en la **Ladder Palette** (o apriete F2).

Usted verá nuevamente que el cursor cambia a una caja de diálogo con un contacto abierto, una ventana con el cursor de texto que centellea en la dirección **C0** (destacada) y un indicador verde. Entre **X2** donde **C0** está destacado. Después de que haya entrado la dirección y el indicador de error es verde, haga clic en la marca de verificación (✓) o presione la tecla **ENTER**.

Para programar un temporizador de 10 segundos, mueva el cursor de modo que quede sobre la instrucción **NOP** en el renglón, y haga clic en el botón **Box** en la **Ladder Palette**. Aparecerá la ventana **Instruction Browser**. Mueva las flechas hacia arriba o hacia abajo o use el mouse para seleccionar **Timer/Counter/SR** en la sección de **BOX CLASS** (clase de box) de la ventana. **TMR** estará en la lista de boxes; haga clic para destacarlo. Haga clic en **OK** para entrar el elemento. Aparecerá el diálogo mostrado en la figura de la próxima página.

Note que hay dos campos a ser llenados. El superior determina la identificación del temporizador (T0, en este caso). El campo inferior define el tiempo que toma en activarse el contacto T0. Cada unidad es de 0,1 segundos, de modo que para hacer 6 segundos debe escribirse allí el valor 60, o K60, siendo K el método normal de indicar una constante. Este campo también puede tener una variable, un registro tal como V4000. Después de escribir estos valores, acepte esta instrucción haciendo **Enter**. El programa quedará como en la figura adyacente. Haga clic en el botón **ACCEPT** en el menú barra de herramientas para compilar el programa. Este programa se puede descargar a un PLC de la forma que está o, si desea, se pueden agregar renglones adicionales al programa. Note que la instrucción **END** está colocada al final del programa.



Continúe practicando lo que se ha discutido antes de continuar. Esta es una de las formas de escribir el programa en *DirectSOFT*.

**Paso 10: Acepte para compilar y luego salve el programa**

Continuaremos con este ejemplo para hacer las cosas simples. El programa necesita ser aceptado para ser descargado al PLC. Note que los dos botones de diskette a la izquierda del menú barra de herramientas que permiten leer desde el disco (**READ FROM DISK**) o escribir al disco (**WRITE TO DISK**), no están "grises".



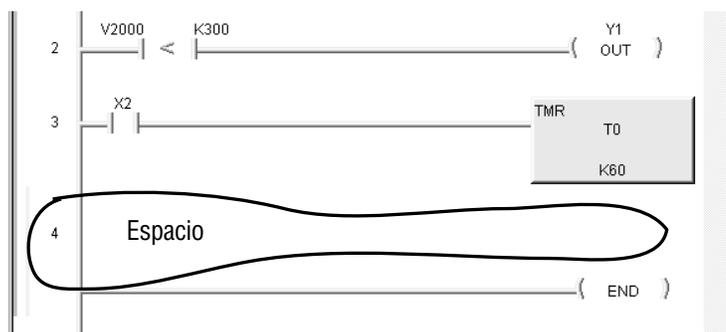
En este caso, usted deseará hacer clic el botón de escribir para salvar el programa al disco duro (no es necesario salvar el programa al disco para descargar el programa a un PLC). Es una buena práctica salvar su trabajo mientras modifica un programa. Puede haber una equivocación ocasionalmente y puede desear restaurar el programa al estado que estaba antes de que sucediera la equivocación. Para hacer esto, se puede hacer clic en el botón READ (leer) y el programa previamente salvado aparecerá en la pantalla y la programación puede continuar.

### Paso 11: Programe un contador

En este ejemplo, vamos a abrir un espacio entre el renglón 3 y el 4. Para eso, coloque el cursor al comienzo del renglón 4 y apriete la tecla **Enter**. Esta acción abre un espacio y muestra el número 4 arriba del renglón que tiene la instrucción END. Luego será colocado un contacto normalmente abierto en la primera posición del renglón 4, como el pulso para activar el contador. Coloque el cursor al principio del renglón y apriete **F2**.

Usted verá nuevamente que el cursor cambia a una caja de diálogo con un contacto abierto, una ventana con el cursor de texto que centellea en la dirección **C0** (destacada) y un

indicador verde. Entre **X3** donde está **CO**. Después de que haya entrado la dirección y el indicador de error es verde, presione la tecla **ENTER**.



Para programar un contador de hasta 9999 pulsos, haga una línea apretando las teclas **CTRL+flecha ->** simultáneamente y haga clic en el botón **Box** en la **Ladder Palette**. Aparecerá la ventana **Instruction Browser**. Mueva las flechas hacia arriba o hacia abajo o use el mouse para seleccionar **Timer/Counter/SR** en la sección de **BOX CLASS** (clase de box) de la ventana. **CNT** estará en el primer lugar de la lista de boxes; haga clic para destacarlo. Haga clic en **OK** para entrar el elemento. El contador tiene un contacto de reset. Use **X4** para volver el valor de conteo a cero. El valor corriente del contador se ve en **V1000**. para el contador **CT1** estará en **V1001**, y así sucesivamente. Esta acción programa el contador. Si Ud necesita contar más de 9999, considere usar el condador **UDC**, que puede contar hasta 99999999.

### Paso 12: Acepte para compilar y luego salve el programa

Continuaremos con este ejemplo para hacer las cosas simples. El programa necesita ser aceptado para ser descargado al PLC. Note que los dos botones de diskette a la izquierda del menú barra de herramientas que permiten leer desde el disco (**READ FROM DISK**) o escribir al disco (**WRITE TO DISK**), no están "grises". En este caso, usted deseará hacer clic el botón de escribir para salvar el programa al disco duro. Es una buena práctica salvar su trabajo mientras modifica un programa.

**Ejercicios para el nuevo programador:**

¿Ud. entiende el direccionamiento de PLCs, no es cierto? Que bien, porque ahora usted va a poder escribir un programa. Como primera acción, puede ser que sea provechoso leer algunas notas que tenemos para usar *DirectSOFT* y que se encuentran en nuestro sitio de Internet. Hay documentos en inglés que le permiten conocer más de *DirectSOFT*, entre ellos:

AN-DS-002 es un documento que se ocupa de varios problemas de localización de averías que hemos podido documentar para *DirectSOFT*.

AN-DS-003 es un documento que trata de como comenzar y como usar *DirectSOFT*. Revise este documento en su totalidad. Muchos de los trucos de allí le ahorrarán tiempo y le ayudarán a solucionar problemas más eficientemente.

Es necesario también que lea el manual de los módulos de señales análogas D0-OPTIONS-M-SP.

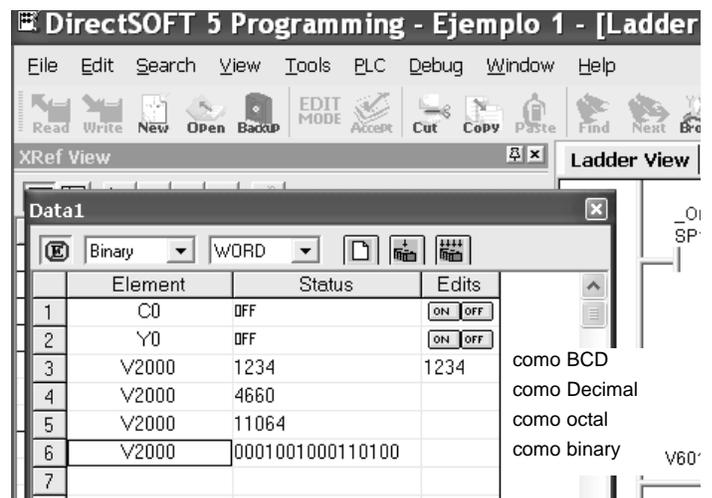
Ahora que usted está poco más familiarizado con el software, escribamos (o hagamos lógica) de lo siguiente (Damos una solución de algunas de estas tareas):

- Conéctese con el PLC y luego abra Data View, haciendo clic en el icono “Data”, y entre V2000 en varias líneas.

Cambie el formato en cada línea a un formato diferente y luego entre forzando datos a V2000 en varios formatos y observe las diferencias. Observe la figura adyacente.

Data View tiene normalmente 2 columnas, pero puede hacer que aparezca la columna 3 haciendo clic en el icono amarillo con una letra E.

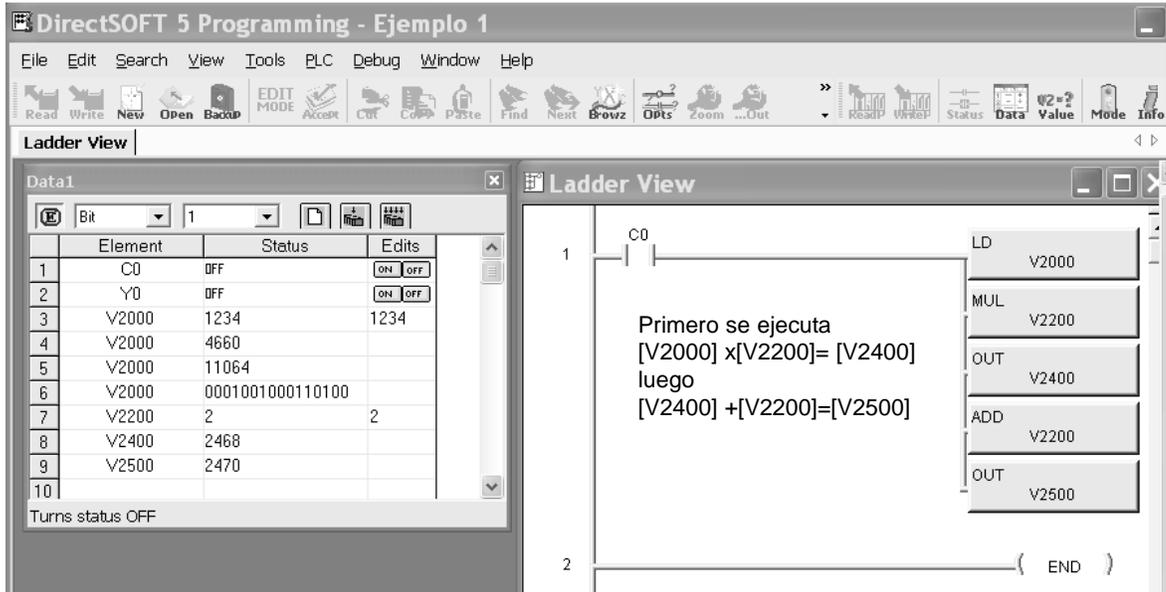
Como puede observar, el valor 1234 en BCD puede ser visto también como 4660 en formato decimal, o como 11064 en formato octal o aún en binario, como secuencia de ceros y unos.



- Haga la misma cosa para las instrucciones binarias/decimales y nuevamente para instrucciones reales (punto flotante).
- Escriba el código para cambiar el formato de datos de varios formatos uno a otro. Use las instrucciones BIN, BCD, BTOR, y RTOB para cambiar datos de formato a formato y viceversa.
- Escriba lógica usando instrucciones BCD tal como MUL y ADD y fuerce los valores V2000 y V2200 correctamente ajustados a formato BCD así como también datos incorrectamente ajustados a formato y vea los resultados. Accione C0 desde la columna Edit. Vea la figura de la página siguiente. Cada vez que acciona C0 para ON, el PLC ejecuta la operación.

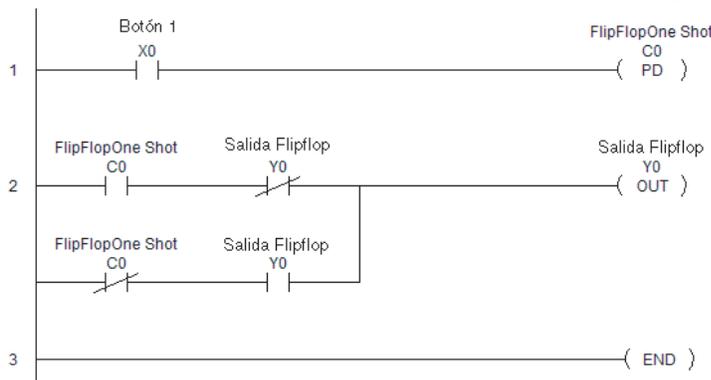
La operación funciona de la siguiente manera: cada vez que el renglón se hace verdadero, la instrucción LD (load) coloca el valor de V2000 en el acumulador del PLC. Luego la

instrucción MUL (multiplicación en BCD) toma el valor en el acumulador y lo multiplica por el argumento de la instrucción MUL (V2200 en este caso). Luego la instrucción OUT lee lo que está en el acumulador y lo copia el argumento de la instrucción (V2400 en este caso). Luego la instrucción ADD suma en BCD el valor del argumento (V2200 en este caso) con el contenido del acumulador y lo coloca nuevamente en el acumulador. Por último, la



instrucción OUT copia a la memoria del argumento de OUT lo que está en el acumulador

- Escriba el código para ejecutar un temporizador T6 en que la salida sea activada cuando la entrada C10 se activa. El temporizador debe comenzar a contar el tiempo cuando la entrada se apaga. La salida debe apagarse cuando el temporizador llega al tiempo prefijado. Este temporizador se conoce en inglés como OFF-DELAY TIMER. Hay dos ejemplos en el sitio de Internet de **AUTOMATIONDIRECT**>Tech support>Example programs> documento EP-MISC-012. Esta área es una buena fuente para obtener ejemplos.
- Escriba la lógica para hacer que un botón de entrada encienda una salida cuando se aprieta una vez y luego apague la salida cuando se apriete nuevamente. Esto es el circuito **flip flop**. Asegúrese de que sean eliminados los rebotes del contacto de modo que al apretar el botón no haga que el circuito complete un ciclo. Usted tal vez quiera usar la bobina PD (One shot) y contactos diferenciales (o un pulso en la transición de OFF para ON o “one shot”).



- Escriba el código para hacer que la instrucción Shift register (SR) cambie la posición de un bit repetidamente desde C0 hasta C16. Utilice el relevador especial SP4 (con ciclos de 1 segundo) para producir el cambio. Cada vez que un bit se cambia en C16, C0 debe activarse otra vez. Éste es un método de generar un circuito que completa un ciclo, útil para renglones de lógica de comunicaciones y otras características.
- Escriba la lógica usando un contador que vuelva a cero cada 16 conteos. Use el relevador especial SP4. Esta lógica se puede también usar como un circuito que completa un ciclo. Algunos clientes prefieren el SR y algunos prefieren contadores.

### Resumen de las particularidades de las familias de PLCs *Direct*LOGIC

- Las instrucciones aritméticas por defecto son BCD, y hay instrucciones que pueden modificar el número para otros formatos, tales como binario o real.
- Las funciones aritméticas trabajan con el acumulador de modo que usted tiene que cargar un valor en el acumulador antes de que usted pueda realizar una operación en ella en un bloque de aritmética. El resultado residirá en el acumulador, de modo que debe mover el resultado a una dirección de memoria con una instrucción OUT.
- Los valores prefijados de temporizadores y contadores deben tener el formato BCD. Además, los valores actuales de temporizadores y contadores están en formato BCD (incluso las constantes).
- Los temporizadores TMRA y contadores UDC toman (2) palabras consecutivas cada uno, de modo que si usted tiene un TMRA T0 usted no puede tener un T1.
- Las instrucciones DRUM (TAMBOR) toman cuatro direcciones consecutivas de contadores.
- Las instrucciones de temporizadores, contadores y tambores no se deberían usarse en lógica de subrutinas o de etapas. Su comportamiento puede ser comprometido por estar localizado en etapas o subrutinas.
- Las asignaciones de entradas y salidas en todos los PLCs (excepto la familia 305) serán asignadas de izquierda a derecha por el tipo de módulo. De modo no hay diferencia en la asignación para 2 módulos de entradas y luego 2 módulos de salidas versus entradas-salidas-entradas-salidas. *Por esta razón se recomienda NO dejar ranuras vacías a la izquierda de cualquier módulo de entradas y salidas*, ya que al agregar un módulo del mismo tipo a la izquierda de un módulo cambiará las asignaciones de entradas y salidas de todos los módulos que vienen a continuación.
- Los módulos análogos tendrán asignadas X e Y incluso si se programan usando el método del puntero.

### Programación de entradas y salidas análogas con PLCs *Direct*LOGIC

Ahora usaremos entradas y salidas análogas. Obtenga un módulo de entradas y de salidas de voltaje tal como el F0-2AD2DA-2. Éste es más fácil de usar para conectar uno al otro puesto que los módulos de voltaje no tienen tipo de salidas surtidoras/drenadoras. Usted no podría conectar una señal no aislada de 4-20 mA con una entrada de 4-20 mA en nuestros PLCs a menos que fuera un módulo aislado como algunos de nuestros módulos 4-20 mA. Esto puede también ser un problema al conectar con otros dispositivos, y nuestros clientes tienen con

frecuencia este problema.

La mayoría de nuestros módulos análogos requieren que se alimente 24VCC a dos terminales en el módulo.

**También, cada módulo de señales análogas debe ser configurado por ladder.**

Hay dos tipos de configuración para los módulos análogos en nuestros PLC's DL06; con IBOX y con el método del puntero.

La instrucción IBOX es la más fácil. También, el PLC DL06 usa el método del puntero. **¡El método del puntero también es fácil!**

Los módulos análogos reciben una asignación de bits X o Y de la misma forma que son hechos en los módulos discretos. Vea el manual de módulos análogos D0-OPTIONS-M-SP para más información.

En el método del puntero, la programación es simple. Usted programa un solo renglón para cada módulo. Este renglón contiene la información que le dice al PLC cuántos canales usted desea, donde usted desea los datos localizados y en qué formato usted desea los datos.

Una advertencia es que los números de la ranura comienzan en 1 y 1 es la ranura más a la izquierda en el PLC, mirando el PLC por la frente (en el PLC DL06; esto no es válido para PLCs de otras familias). Obviamente, si se selecciona la dirección dedicada incorrecta porque se tiene el módulo en una ranura diferente, las entradas no trabajarán.

### Una palabra sobre números negativos

*¡No hay números negativos en BCD!* En BCD los números negativos son representados generalmente activando el bit más significativo de los 16 bits para ese canal. Esto tiene el efecto de sumar una constante 8000 al número. De modo que 8035 es el negativo 35.

En números decimales (o binarios, como se define en el manual), los números negativos se indican con el complemento de 2, que es el número está invertido (todo los 1s cambian a ceros, a todos los ceros cambian a 1s) y luego se suma un valor binario de 1. Esta conversión es también necesaria si usted desea exhibir un número negativo en un panel táctil (o touch screen). Sepa que la función PID considera números negativos como magnitud más signo. En magnitud más signo, el bit más significativo de la palabra decimal que está encendido indica un número negativo. El rango es entonces -32767 a 32767.

Entonces, aquí están sus tareas sobre módulos análogos:

- Conecte un canal de salida de su módulo de salida de voltaje con un canal de entrada de su módulo de entradas de voltaje. Escriba un programa que permite que todos los canales en ambos módulos tengan los datos del BCD.
- Vaya a Data View en *DirectSOFT* y envíe los datos al canal de la salida que usted ha conectado al canal de entrada y observe los datos recibidos para ese módulo en **Data View**.
- Abra el circuito del canal o lleve la señal fuera del rango y observe el resultado.

## Una palabra sobre programación de PID

¿Qué hace el control PID? Aprenda sobre control PID en el capítulo 8 de este manual. No hay programación de PID en nuestros PLCs, ya que realmente no hay una instrucción lógica para el PID. En nuestros PLCs el PID es una función que está activada o desactivada. Usted puede enviarle datos de lógica y puede también activar el o los lazos con la programación.

Aquí están los principales asuntos para el PID:

- No se olvide de entrar un valor máximo de salida diferente de cero, que es el valor por defecto; si usted no lo hace, el PID parecerá que está trabajando, pero la salida nunca cambiará.
- No cambie los bits de cambio de modo manual/auto/cascade permanentemente. Estos bits necesitan ser encendidos con un pulso one shot. El código activa básicamente el bit SET y después el PLC cambia modo y resetea el bit.
- El lazo PID necesita que la variable de referencia (Set point) y de proceso estén en formato binario o decimal (magnitud más signo si es negativa). La salida PID está también en decimal. Cerciórese de que su código análogo esté instalado como decimal o agregue el código convertido desde BCD a decimal antes de enviar al PID.
- El modo **bumpless 1** cambiará el SP para hacerlo igual al PV al cambiar al modo de funcionamiento. Esto incomoda a clientes que no están escribiendo permanentemente a su registro de SP. Usted puede desactivar esto seleccionando el modo **bumpless 2**.
- **Bias Freeze** le dice al PID que pare de calcular un **bias** adicional si la salida está en el máximo o el mínimo (Esto debería estar activado por defecto). Y los clientes con lazos de control PID con una rampa de larga duración necesitan activar esto para estar en el lado seguro.
- Cada lazo **PID** necesita tener un valor específico de ganancia (GAIN) diferente de cero o de otra forma nunca funcionará. La ganancia debe ser ajustada durante la sintonización (tuning) junto con el valor de Reset y de Rate.
- Vea un ejemplo de PID en el ejemplo 7 de este capítulo.

## El PLC DL06 puede tener entradas y salidas remotas

El PLC DL06 tiene dos puertos seriales, que permiten hacer comunicación serial con el protocolo MODBUS RTU como maestro o esclavo. Esta función le permite añadir más de 2000 entradas y salidas remotas hasta una distancia de hasta 1000 metros con el sistema de cableado RS-422 or S-485, disponible en el puerto 2 del PLC. Evidentemente esto también se puede instalar en el mismo gabinete del PLC DL06. Un esclavo es, por ejemplo, otro PLC DL06 o un PLC DL05 o un variador de frecuencia GS2-25P0, que son declarados esclavos.

También **Automation Direct** tiene un producto modular llamado Terminator I/O, que es un sistema de entradas y salidas no inteligentes, pero que acepta el protocolo MODBUS como esclavo hasta una velocidad de transmisión de 38.4 KBaud y puede aceptar fácilmente hasta 90 nodos. Vea más informaciones de Terminator I/O en nuestro sitio de Internet.

# Concepto de una instalación controlada por PLC's.

El proyecto de control de una máquina o un sistema de equipos a ser controlado por uno o más PLCs es tarea de uno o más técnicos, quienes deben entender los detalles de operación de la máquina o el sistema. La programación del PLC es una tarea, entre muchas otras tareas, en el diseño de un sistema, que debe ser ejecutada para la implementación de un proyecto. Así como se requiere estudios para definir el proceso, para escoger la mejor solución, para seleccionar cada una de las piezas que componen un sistema, también es necesario planear la selección del sistema de control con la parte de operación (HMI), uno o más PLCs, sensores, contactores, transductores, válvulas solenoides, motores y actuadores entre otro equipo. y después como será alambrado el sistema, cuántas y qué tipo de entradas y salidas se requieren.

No es el propósito de este capítulo describir cada una de las etapas anteriores a la programación de un PLC; sin embargo, describiremos aquí en general acciones que es necesario implementar para tener un sistema de control con PLCs hecho en forma exitosa.

Cualquier sistema de control tendrá como entradas: comandos hechos por un operador, entre otras, que son hechos a través de sistemas llamados paneles de operador o, más en general, interfaces hombre máquina (H. M. I.), definición de los estados de la maquinaria o de un sistema, tales como el estado de motores (funcionando o parados), conmutadores para determinar si una determinada máquina puede funcionar antes que otra, sensores de proximidad que determinan la posición de una determinada parte de la maquinaria, sensores fotoeléctricos que detectan luz, para contar u otras funciones, dependiendo de lo que se cruce frente a estos sensores, conmutadores que indican si el nivel de agua está más bajo que un cierto nivel prefijado, confirmación de válvulas solenoides activadas, velocidad de la maquinaria, posición de ejes o de discos, temperaturas, presiones, niveles, corrientes, etc. y salidas discretas como señales para hacer funcionar un motor girando en un sentido o el otro, activar válvulas, lámparas de indicación de estado, sirenas para llamar la atención de un determinado paso o partida de motores o salidas análogas para actuación de variadores de frecuencia, válvulas para actuadores de control de flujo, aparatos de potencia de control de potencia para controlar calentamiento o refrigeración, e incluso transferencia en tiempo real de informaciones para sistemas que adquisición de datos.

La programación de un PLC es una de las últimas etapas de un proyecto de control, ya que primeramente se tiene que definir cada uno de los componentes del PLC y cada uno de los sensores, contactores, el alambrado a cada uno de los sensores, a los paneles de operador, a los centros de controles de motores, a los variadores de frecuencia, a los encoders para medir velocidad o posición etc.

Tomemos por ejemplo el caso de un sistema simple de control de una planta de cemento; una planta de cemento tiene típicamente unos 300 a 500 motores, unas 150 mediciones de temperatura, presión, flujo, unos 1000 switches de diferentes tipos, unas 200 válvulas solenoides y otros componentes los cuales deben ser comandados, debe verse el estado de la mayoría de ellos, y debe tenerse un sistema para eso. Aquí los PLCs juegan un papel importante en el control de este sistema. Cualquier defecto en una o más de las máquinas significa que hay que tomar acción inmediata mente para evitar parar la planta y perder producción o incluso dañar otras máquinas.

Considerando toda la información descrita anteriormente, estos son los pasos básicos para poder hacer un desarrollo del programa.



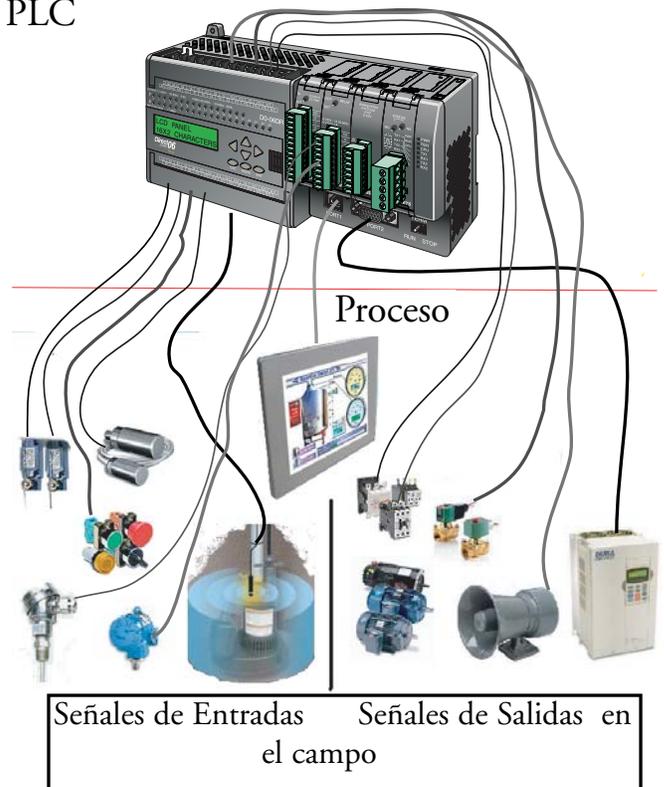
**ADVERTENCIA:** Es su responsabilidad diseñar un ambiente seguro de operación para el personal y el equipo y debe ser su meta primaria durante la planificación de sistema e instalación. Los sistemas de automatización pueden fallar y poder tener como resultado situaciones que pueden causar heridas graves al personal o daño al equipo. No confíe solamente en el sistema de automatización para hacer seguro el ambiente de operación. Usted debe usar artefactos electro-mecánicos externos, tales como conmutadores de límite, independientes de la aplicación del PLC para suministrar protección para cualquier parte del sistema que pueda causar heridas o daños personales. Cada aplicación de automatización es diferente, así que puede haber requerimientos especiales para su aplicación particular.

### Etapa 1 Descripción sumaria de alto nivel de la aplicación

Tenga un documento con una primera descripción de su aplicación; debe hacerse en los términos más sencillos posibles (descripción en texto, una lista de instrucciones, un diagrama de flujo o cualquiera que le sea cómodo). Evite la implementación de detalles específicos en este punto. Si usted no puede describir su aplicación, usted no puede PLC programarla. Si su aplicación es grande y compleja, desglóselas en partes más pequeñas.

Casi todos programas de aplicación tienen tres componentes: Entradas, Proceso, Salidas.

- ¿Cuales son sus entradas en la aplicación? En el puerto serial, entradas físicas, memoria del PLC.
- ¿Qué proceso se debe hacer? Analice la serie de datos, cálculo de valores, construya una cadena de enclavamiento, etc.
- ¿Dónde van las salidas de la aplicación? activan salidas como motores, válvulas solenoides, envían datos procesados por vías de comunicación, datos a ser enviados por el puerto serial, leen y escriben valores a memorias del PLC, imprimen datos a resultados de cálculos, etc.



### Etapa 2 Revise su descripción

Lea su primera descripción.

- ¿Le hace sentido a usted?
- ¿Tiene definidos los componentes correctos del PLC?
- ¿Se han tomado en cuenta consideraciones para hacer el sistema seguro para el personal y el equipo?
- ¿Es la velocidad de procesamiento un factor a considerar?

- ¿Dónde están los cuellos de botella en la aplicación?
- ¿Hay una manera más sencilla de alcanzar los mismos resultados?
- ¿Hay una manera más rápida de alcanzar los mismos resultados?
- ¿Cuales son las consideraciones o requisitos para esta aplicación?
- ¿Han sido alcanzadas las consideraciones o requisitos especiales para esta aplicación?

Si las preguntas han causado cambios en partes de la aplicación, repita los pasos 1 y 2 hasta que usted esté satisfecho y luego proceda a la etapa siguiente.

### **Etapas 3 Descripción de nivel eléctrico de la aplicación**

Refine su primera descripción de la aplicación con detalles específicos usando los términos más sencillos (descripción en texto, una lista de declaraciones, un diagrama de flujo o cualquier otro que haga que Ud. se sienta cómodo). Debe tener claro toda la parte de hardware que será usada.

### **Etapas 4 Codifique y arme una aplicación de prueba.**

En esta etapa Ud. comenzará a trabajar con el o los PLCs. Asegúrese que conoce bien el PLC y todas sus características y configuraciones. Escoja y verifique los módulos de entradas y salidas adecuados para cada función. Entienda todas las opciones de alambreado. Entienda las opciones para programación. Entienda la necesidad de hacer mantención y búsqueda de problemas en el sistema cuando éste ya esté trabajando.

Cada programa tiene varias instrucciones que se deben incluir. El programa puede trabajar sin estas instrucciones pero es buena práctica siempre que programe incluir estas instrucciones, porque le pueden salvar tiempo en la depuración de un programa durante la implantación. Documente la codificación, es decir, coloque apodos en cada entrada y salida, y haga comentarios en cada rung según la lógica que ud está usando.

### **Etapas 5 Escriba código para el próximo componente del programa de la aplicación.**

Mientras más sencillo es el componente, mejor. Documente el programa mientras lo ejecuta.

### **Etapas 6 Pruebe todas las combinaciones posibles de entradas y la salidas de los componentes de la aplicación.**

### **Etapas 7 Repita las etapas 5 y 6 para cada componente de la aplicación**

### **Etapas 8 Pruebe la aplicación completamente con cada combinación posible antes de colocar en funcionamiento el programa.**

Verifique que la documentación esté correcta, para que todas las personas que vayan a usarlo entiendan bien cada uno de los procedimientos.

### **Etapas 9 Colocación en funcionamiento:**

Esta es la etapa donde se descubrirá si se ha hecho un buen programa o todavía es necesario cambiar algunas partes del programa. Típicamente esta parte es muy esforzada porque el dueño, (llamemos dueño al que tiene más interés que el proceso esté funcionando), quiere tener la máquina o el conjunto de máquinas trabajando en el más corto tiempo posible. En este período es donde el ingeniero electricista sufre todas las culpas del atraso del proyecto, aunque no sea verdad. Es muy conveniente evitar esos problemas al máximo.

## Ejemplos de programas ladder con PLC DL06

### Ejemplo 1.

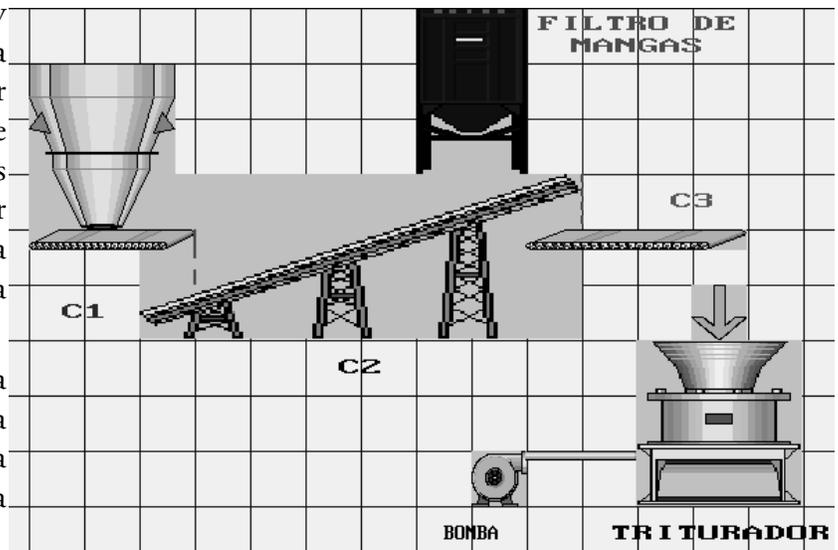
A continuación presentamos como hacer un proyecto que debe ser definido por un técnico que entienda el proceso a ser controlado. Hay muchas etapas iniciales antes de hacer el sistema de control y de ejecutar la programación del PLC:

Como ejemplo de la metodología para poder hacer un proyecto industrial o comercial con un PLC mostraremos un ejemplo simple de un sistema de molienda de mineral con 3 correas transportadoras. En los próximos ejemplos trabajaremos sobre el mismo concepto.

**Etapas 1.** Este control debe tener un enclavamiento de partida de tres correas transportadoras de mineral que deben partir una luego de otra con un retraso entre cada una para alimentar un triturador de mineral y

además, al ser activada una parada de emergencia por cuerda, todas las correas que están en el lado “aguas arriba” van a parar inmediatamente. Vea la figura adyacente con la idea descrita.

Cada correa transportadora C1, C2 o C3 tiene asociada una parada de emergencia por cuerda a lo largo de la correa.



En la partida, el operador apretará un botón que hará sonar una sirena por 45 segundos, para alertar a las personas que el sistema va a partir. Al final de los 45 segundos parte la bomba de lubricación del triturador y al mismo tiempo el ventilador de un filtro de mangas para sacar el polvo que será creado en el transporte y en la molienda, si el aire para limpiar las mangas tiene la presión suficiente.

Si la presión de lubricación llega al valor mínimo aceptable y el ventilador está corriendo, parte el triturador de mineral y después de un cierto tiempo partirán las otras correas para eliminar cualquier material que pueda haber quedado en la parada anterior en la secuencia C3, C2 y C1 con un intervalo de tiempo de cada una.

La parada normal hará que la correa C1, pare primeramente y permitirá que las otras queden vacías antes de parar. Cada correa tiene un motor de inducción de una velocidad solamente, en este ejemplo. Por definiciones del proceso, el tiempo de parada está definido con un cierto tiempo entre correas (puede ser otro valor cualquiera diferente que el que se ha programado). Luego se para el triturador y después de un cierto tiempo, la bomba de lubricación del triturador.

### Etapa 2 Revisemos la descripción:

- Lea la descripción. *Falta definir el lugar del pupitre de control; Eso debe ser definido por la conveniencia de la operación y las distancias envueltas.*
- ¿Le hace sentido a usted? *Parece ser correcto pero falta saber qué viene después del triturador. Tendría sentido colocar un enclavamiento para el sistema que va después del triturador. Sería interesante mostrar al operador qué equipo está funcionando.*
- ¿Tiene definidos los componentes correctos del PLC? *NO, los definiremos mas adelante:*
- ¿Se han tomando en cuenta consideraciones para hacer el sistema seguro para el personal y el equipo? *Se han considerado paradas de emergencia en la correas; no se han considerado protecciones en el triturador ni en el filtro de mangas. Posiblemente existan protecciones mecánicas para impedir el acceso. Debe haber alarmas cuando falla la presión de aire o de lubricación. Si para el triturador, deben parar inmediatamente las correas, Si se para el ventilador, debe pararse la correa C1 y luego las otras hasta parar el triturador vacío.*
- ¿Es la velocidad de procesamiento un factor a considerar? *No en este momento.*
- ¿Dónde están los cuellos de botella en la aplicación? *Posiblemente en la posibilidad de atorar el triturador con material. El triturador tiene una capacidad de 300 Toneladas por hora y si se sobrepasa este límite el motor puede parar y lo peor es que el triturador quede lleno y al quedar lleno no puede partir tan fácilmente.*
- ¿Hay una manera más sencilla de alcanzar los mismos resultados? *Bueno, el sistema está ya definido; no hay por ahora opciones en este caso. Sin embargo, la solución de control puede tener más de una alternativa. De hecho existen otras alternativas, pero no las discutiremos aquí.*
- ¿Hay una manera más rápida de alcanzar los mismos resultados? *Este no es un sistema en que la velocidad de procesamiento del control juegue un papel importante, por lo menos en esta etapa.*

### Etapa 3 Descripción de nivel eléctrico de la aplicación

Este es una definición de las necesidades para la parte eléctrica de potencia:

Alimentación trifásica 380 Volt, 50 Hz, 20 MVA de potencia de cortocircuito

Motor de la correa alimentadora C1	20 HP, 1500 rpm
Motor de la correa transportadora C2	50 HP, 1500 rpm
Motor de la correa transportadora C3	10 HP, 1500 rpm
Motor de la bomba	3 HP, 1000 rpm
Motor del triturador	500 HP, 1500 rpm con 6,6 kV; control separado
Motor del ventilador del filtro de mangas	30 HP, 1500 rpm

Necesitamos de un centro de control de motores con 5 partidores en 380 Volt; el control del triturador en media tensión, los fusibles o interruptores necesarios, todo instalado en uno o más gabinetes metálicos, de acuerdo a normas preestablecidas de seguridad, una sirena de alarma, un pupitre de control con botones y lámparas, el PLC para hacer el control, incluyendo el programa de control, un diagrama de alambrado de cada componente, instrucciones de instalación y de

operación, después de tener claro el concepto de control. El control del triturador solamente necesita un contacto que se cierra para partir, y se para al abrir este contacto.



*Nota: Este es solamente un ejemplo. AUTOMATIONDIRECT no toma ninguna responsabilidad en la implementación de un sistema similar. Este ejemplo solamente sirve para mostrar la metodología.*

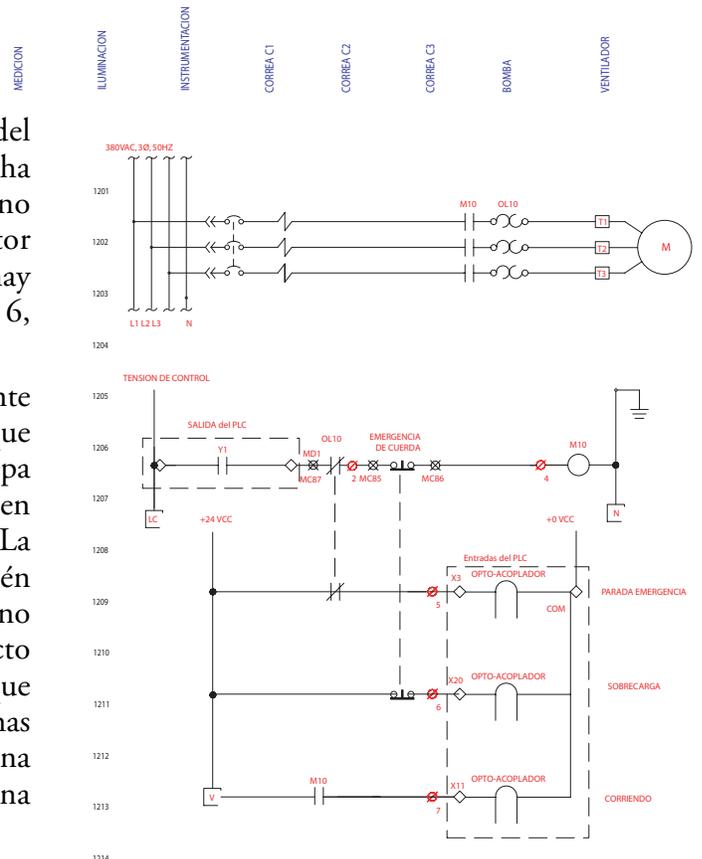
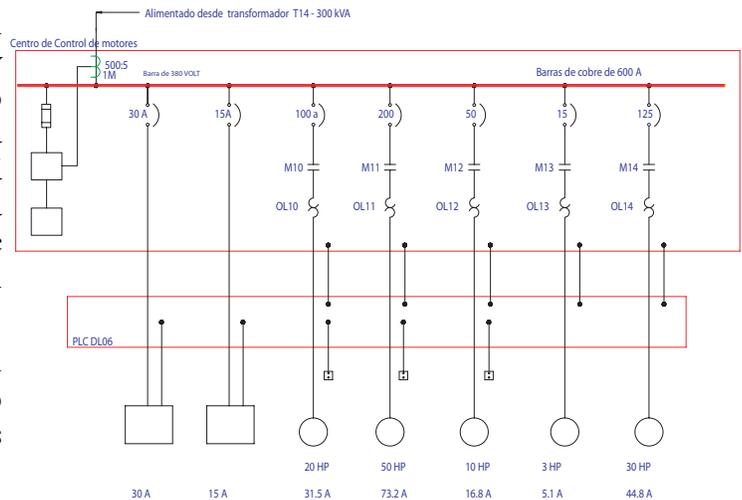
En la figura adyacente se muestra un diagrama unifilar de potencia y como se ha desarrollado el cableado de control de cada motor; esto da una idea de como será interconectado el PLC a cada entrada y cada salida. Naturalmente es necesario más documentación que ésta.

El botón de emergencia principal debe entrar en el PLC y al mismo tiempo debe desenergizar todas las salidas del PLC.

En la segunda figura de esta página se muestra una conexión típica de cada partidor de motor de 380 Volt.

Note que el relevador de sobrecarga OL10 abre el circuito de la bobina del partidor físicamente, es decir, se ha optado en este caso que el PLC no intervenga en la apertura del contactor del partidor del motor; sin embargo, hay aún una señal de sobrecarga en el borne 6, que va al PLC.

Esta señal se usa en este caso solamente como indicación para el operador, lo que puede ser conveniente para que éste sepa que el motor ha sido sobrecargado y en este caso no es un problema eléctrico. La parada de emergencia por cuerda también se ha implementado en el cableado y no en el PLC. Por último, el contacto auxiliar del contactor M10 confirma que el contactor se ha cerrado. Hay muchas otras soluciones de control. Ésta es una muy simple y no constituye una recomendación.



## Capítulo 11: Recopilación y ejemplos

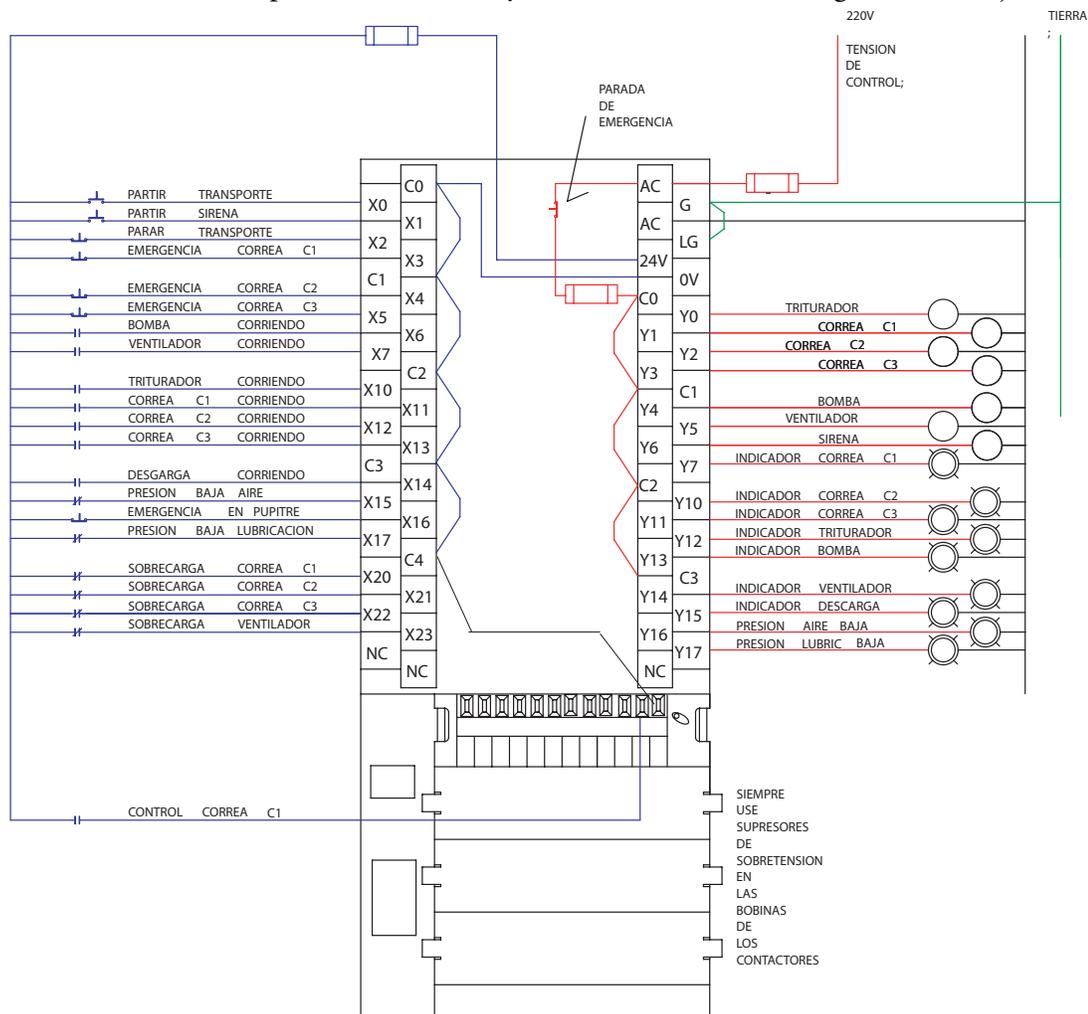
Los diagramas anteriores se deben ejecutar realmente lo más detallados posibles, para evitar que en el futuro los electricistas que deban mantener el sistema se confundan con el concepto. En los treinta años de experiencia del autor se ha visto mucho estas situaciones y vale la pena recalcar mucho este asunto. Sin embargo, no es el propósito de este manual entrar a discutir estos detalles.

En primera instancia escogeremos el PLC D0-06AR para este ejemplo. Este PLC tiene una alimentación de 100 a 240 VAC, 43 a 63 Hz y entradas de 90 a 120 Volt así como salidas entre 17 a 240 VCA.

La primera consideración a ser hecha es escoger 110 o 220 Volt para el voltaje de control. Ya que un voltaje de 380 Volt nos entrega 220 Volt entre una fase y tierra, parece ser más lógico escoger 220 Volt, pero tenemos la limitación que las entradas sólo pueden llegar hasta 120 Volt.

En ese caso, preferiremos hacer un sistema con entradas de 24 Volt CC y salida de 220 Volt. El PLC tiene una fuente de poder de 24 Volt, 300 mA. Cada entrada llega a 8 mA cuando está activada y ya que tenemos 20 entradas usaremos solamente 160 mA en el peor de los casos. De modo que usaremos el PLC D0-06DR. Puede considerarse un módulo de entrada.

Haremos el cableado de potencia, entradas y salidas de acuerdo al diagrama de abajo.



## Asignación de las estradas discretas

Las entradas serán asignadas por una convención arbitraria, tal como la tabla de abajo:

En general se recomienda dejar algunas entradas disponibles para el futuro, en caso de modificación del proyecto, adición de una máquina en el futuro, etc. En este caso, si fueran necesarias otras entradas o salidas, se puede adicionar otro módulo de acuerdo a la necesidad, ya que el PLC DL-06 tiene 4 ranuras para módulos opcionales.

Se ve la necesidad de instalar los botones en un lugar conveniente. Decidimos en este ejemplo que un operador tendrá un pupitre en una localización central, con los botones X0, X1, X2 y X100, además de indicaciones de funcionamiento a ser definidas de acuerdo entre el técnico de proceso y el diseñador del sistema de control. En este caso, agregaremos el módulo D0-08TR.

Note que el proceso no siempre es bien definido por el ingeniero de proceso y por lo tanto es necesario tener siempre una realimentación. Hay varias preguntas que el programador debe hacer al “dueño”, para evitar responsabilidades después que el proceso está funcionando y no está adecuadamente diseñado. No se puede culpar al ejecutor del sistema de control de las fallas de diseño para el operador, que no permitan que el operador esté en control del sistema.

Terminal	Apodo	Descripción
X0	Partir	Botón de partida del operador en el pupitre de control N.A
X1	Sirena	Botón para activar la sirena para comunicación N.A.
X2	Parar normal	Botón de parada en el pupitre de control N.C.
X3	Emergencia de C1	Conmutador de emergencia en la correa C1, N.C
X4	Emergencia de C2	Conmutador de emergencia en la correa C2 N.C
X5	Emergencia de C3	Conmutador de emergencia en la correa C3 N.C
X6	Bomba corriendo	Contacto auxiliar del contacto del motor de la bomba N.A
X7	Ventilador corriendo	Contacto auxiliar del contacto del motor de la bomba N.A.
X10	Triturador corriendo	Contacto auxiliar del contacto del motor de la bomba N.A
X11	C1corriendo	Contacto auxiliar del contactot del motor de la correa C1 N.A.
X12	C2 corriendo	Contacto auxiliar del contacto del motor de la correa C2 N.A
X13	C3 corriendo	Contacto auxiliar del contacto del motor de la correa C3 N.A
X14	Enclavamiento después del triturador	Contacto auxiliar N.C.de un contacto del sistema de descarga
X15	Falta de presión de aire del filtro	Presostato de baja presión de aire.(N.C está normal)
X16	Parada de emergencia principal	Botón de emergencia en el pupitre
X17	Presión de lubricación normal	Presión del aceite de lubricación del triturador (N.C. está normal)
X20	Sobrecarga de la correa C1	Contacto abierto del disparador térmico de sobrecorriente
X21	Sobrecarga de la correa C2	Contacto abierto del disparador térmico de sobrecorriente
X22	Sobrecarga de la correa C3	Contacto abierto del disparador térmico de sobrecorriente
X23	Sobrecarga del ventilador	Contacto abierto del disparador térmico de sobrecorriente
X100	Control de la correa C1	Botón en el pupitre para parar temporalmente las correas
X101	Sobrecarga de la bomba	Contacto abierto del disparador térmico de sobrecorriente
X102	Sobrecarga del triturador	Contacto abierto del disparador térmico de sobrecorriente

### Asignación de salidas discretas

Las salidas serán asignadas por convención como la tabla de abajo:

Terminal	Apodo	Descripción
Y0	Triturador parte	Esta salida parte y hace funcionar el triturador
Y1	C1 activada	Esta salida parte y hace funcionar la correa C1
Y2	C2 activada	Esta salida parte y hace funcionar la correa C2
Y3	C3 activada	Esta salida parte y hace funcionar la correa C3
Y4	Bomba activada	Esta salida parte y hace funcionar la bomba
Y5	Ventilador parte	Esta salida parte y hace funcionar el ventilador
Y6	Sirena activada	Esta salida parte y hace funcionar la sirena
Y7	Indicador de C1	Prende la lámpara C1
Y10	Indicador de C2	Prende la lámpara C2
Y11	Indicador de C3	Prende la lámpara C3
Y12	Indicador de triturador	Prende la lámpara del triturador
Y13	Indicador de bomba	Prende la lámpara de la bomba
Y14	Indicador de ventilador	Prende la lámpara del ventilador
Y15	Indicador descarga OK	Lámpara indicando que la descarga del triturador está corriendo
Y16	Presión baja de aire	Prende la lámpara para indicar presión baja
Y17	Lubricación sin presión	Prende la lámpara para indicar falla en la lubricación
Y100	Indicador sobrecarga C1	Lámpara indicando que la correa C1 paró por sobrecarga
Y101	Indicador sobrecarga C2	Lámpara indicando que la correa C2 paró por sobrecarga
Y102	Indicador sobrecarga C3	Lámpara indicando que la correa C3 paró por sobrecarga
Y103	Indicador sobrecarga bomba	Lámpara indicando que la bomba paró por sobrecarga
Y104	Indicador sobrecarga ventilador	Lámpara indicando que el ventilador paró por sobrecarga
Y105	Indicador sobrecarga Triturador	Lámpara indicando que el triturador paró por sobrecarga

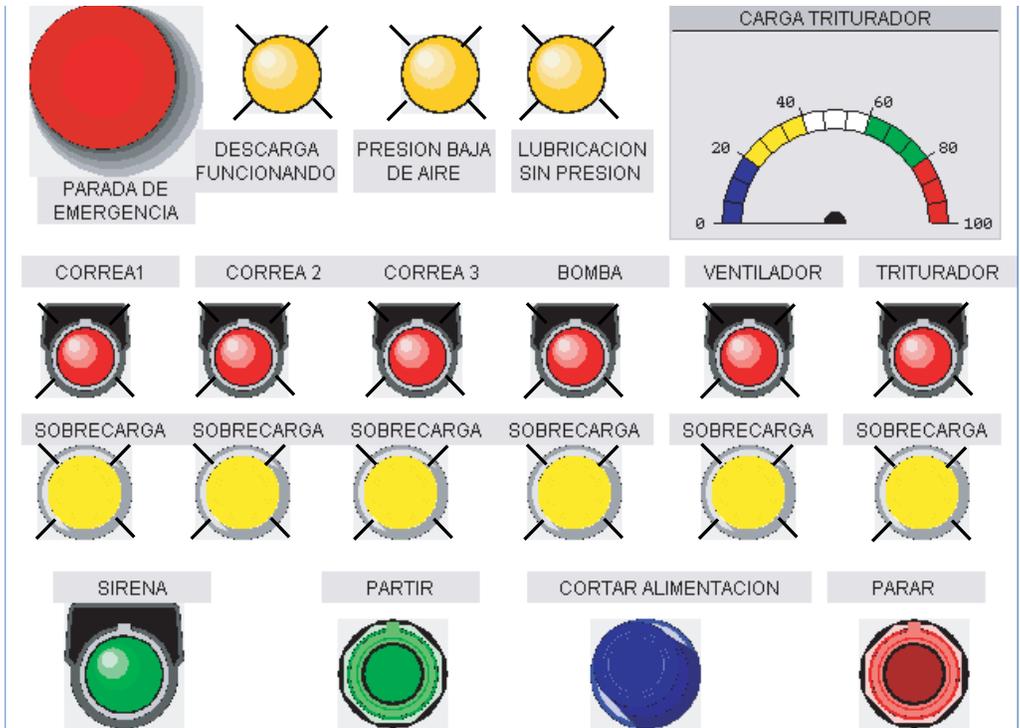
En general se recomienda dejar algunas salidas disponibles para el futuro, en caso de modificación del proyecto, adición de una máquina en el futuro, etc. En este caso, si fueran necesarias otras entradas o salidas, se puede adicionar uno o más módulos de acuerdo a la necesidad, ya que el PLC DL-06 tiene 4 ranuras para módulos opcionales. De hecho, en los siguientes ejemplos, agregaremos más módulos.

Como definimos en la página anterior, un operador tendrá en el pupitre las indicaciones de funcionamiento (definidas de acuerdo entre el técnico de proceso y el diseñador del sistema de control). Digamos que las asignadas arriba fueron ya definidas.

Es muy importante definir estas indicaciones basadas en experiencias anteriores. No hay nada mejor que en un proyecto se use la experiencia de personal que haya trabajado en sistemas similares. Evidentemente es necesario ser inventivo en el caso de un sistema completamente nuevo.

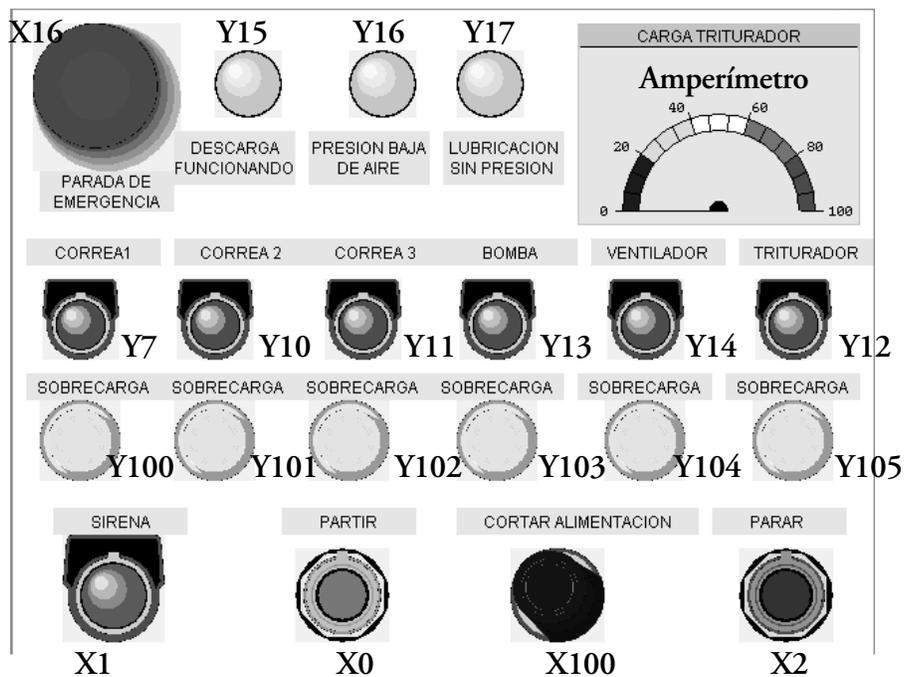
El pupitre de control estará en una parte central de la planta, donde el operador pueda tener acceso al personal de apoyo, si fuera necesario. Vea en la próxima página más informaciones sobre el pupitre.

Todos los indicadores iluminados tienen el símbolo mostrado arriba. Lo que no está indicado significa un botón o selector.



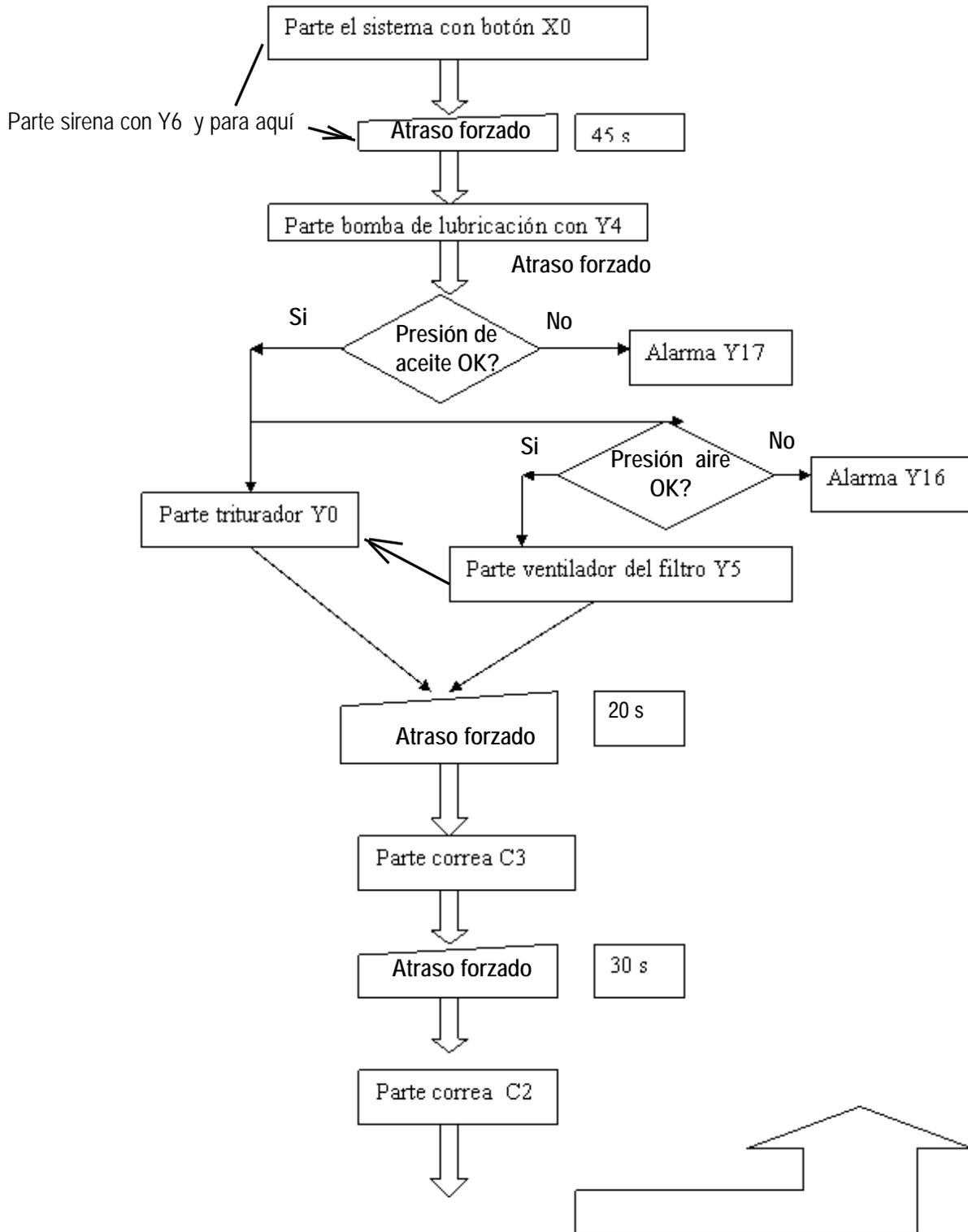
Vista frontal del pupitre

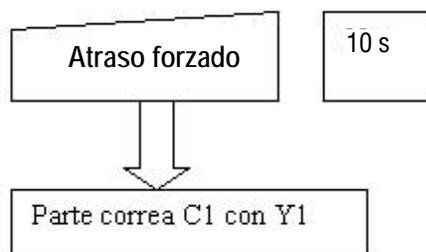
El pupitre tiene asociadas las entradas y salidas mostrada en la figura de abajo. Note que necesitaremos más salidas de relevadores que las previstas, para activar la indicaciones Y100 hasta Y105. Agregaremos el módulo D0-08TR en la ranura 2.



Definición de entradas y salidas del PLC

Diagrama de bloque de la operación





El diagrama de bloques muestra que el sistema está funcionando en este momento.

### Observaciones para el desarrollo del programa

Note los siguientes puntos cuando se observa el diagrama:

- La bomba necesita de unos segundos para crear presión de aceite de lubricación.
- La presión de aire es generada por un compresor fuera del sistema, pero lo interesante es que haya presión de aire necesario para la limpieza de las mangas del filtro. No es necesario saber si el compresor está funcionando. Esto enclava la partida del ventilador del filtro de mangas.
- Aquí no aparece el diagrama de operación de parada normal, de paradas por cuerda de las correas transportadoras en emergencia o en caso de falla de cualquier motor por sobrecarga y será por lo tanto necesario hacer otro diagrama; en ese momento, haremos lo definido en el diagrama, que es típico en el desarrollo del programa ladder. Nada errado con tratar varias veces. La experiencia naturalmente ayuda a hacer el desarrollo lo más simple posible, pero en esta descripción de la ejecución del programa mostraremos cada etapa y como se desarrolla el proyecto.

Observe en las próximas páginas la primera tentativa de ejecutar el programa.

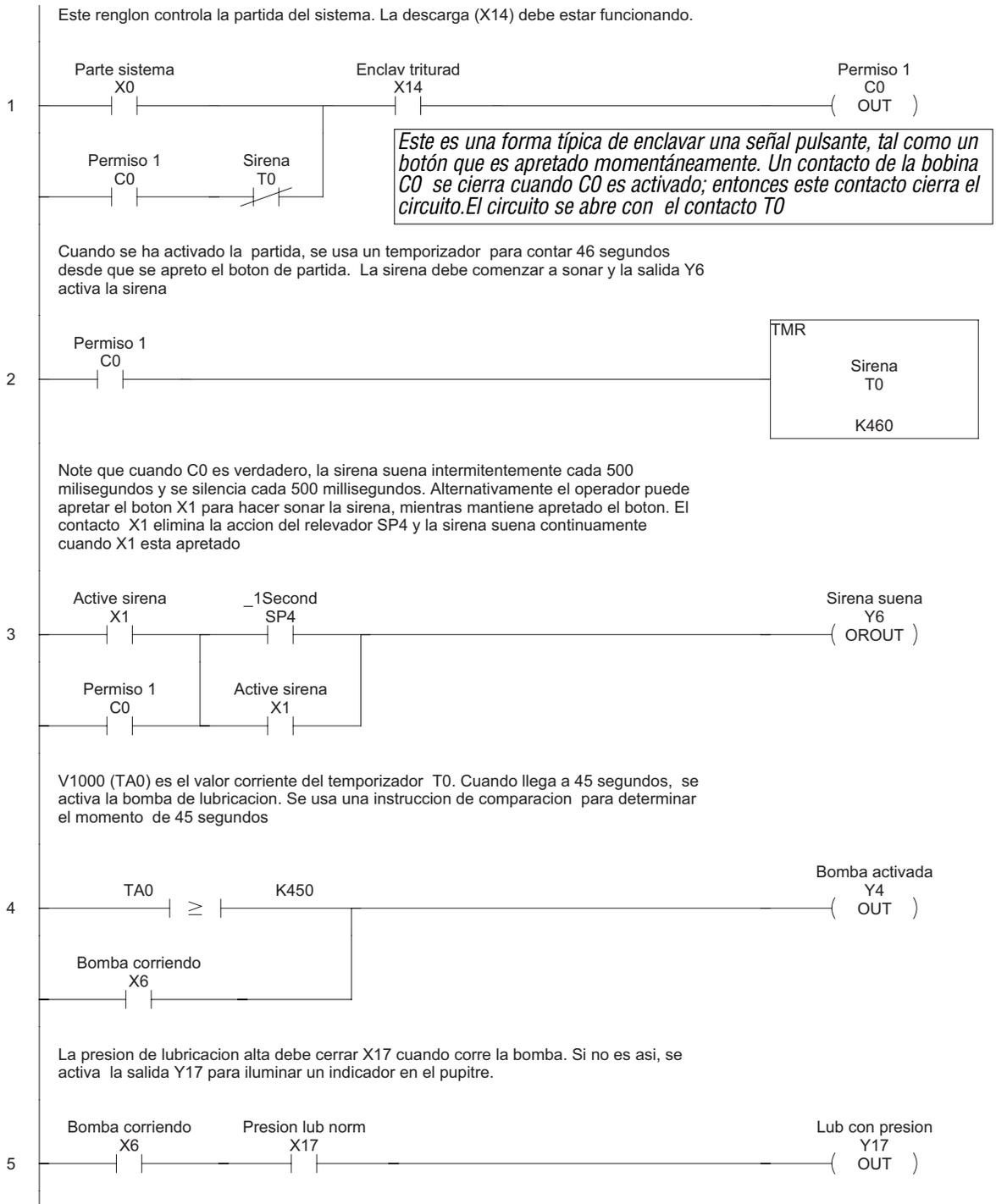
Note que cada partidador de motor tiene un contacto auxiliar que informa al PLC que realmente el contactor se ha cerrado. Esto es necesario para estar seguro que el motor está corriendo. Hay también otras formas de tener la confirmación de que el motor corre normalmente. Esta operación es remota y por lo tanto el operador no vé si una de las correas transportadoras o la bomba realmente están funcionando.

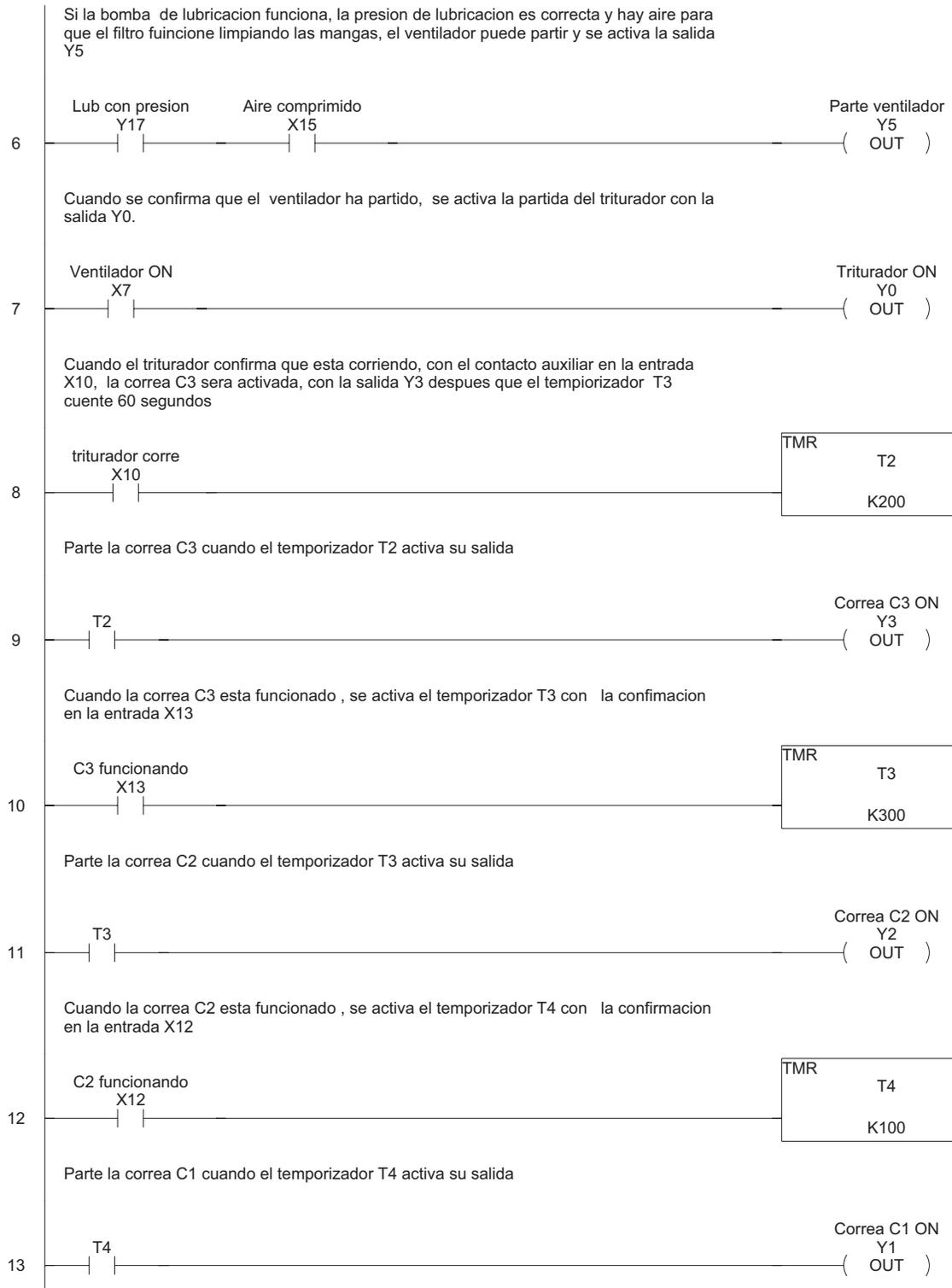
Note también que el pupitre tiene indicaciones sobre el funcionamiento de cada elemento de la operación y hay que asociarles una entrada o una salida.

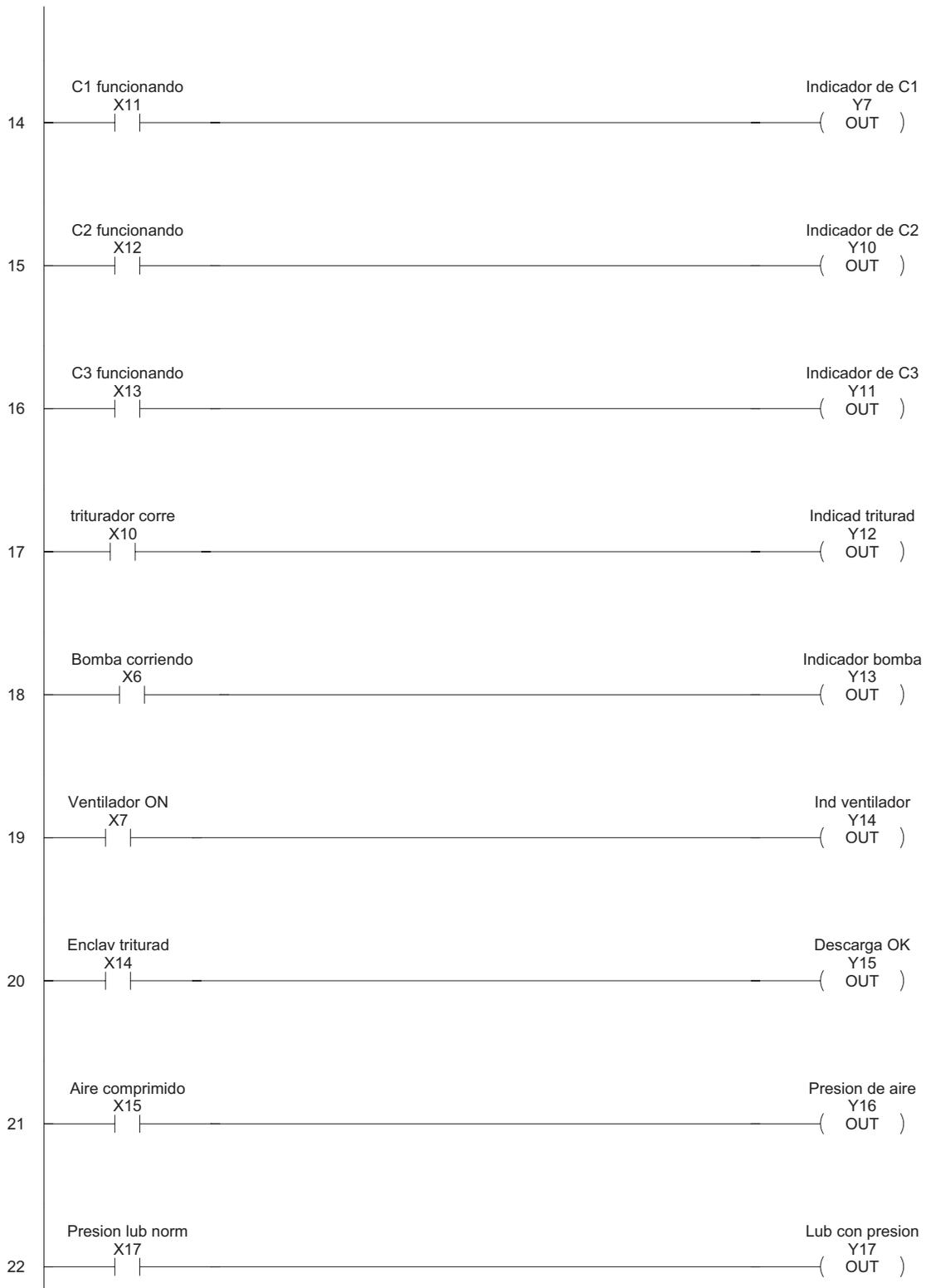
En general, no se tiene disponible durante la programación el centro de control de motores, el pupitre, los presostatos y ninguno de los elementos que completan el circuito de control.

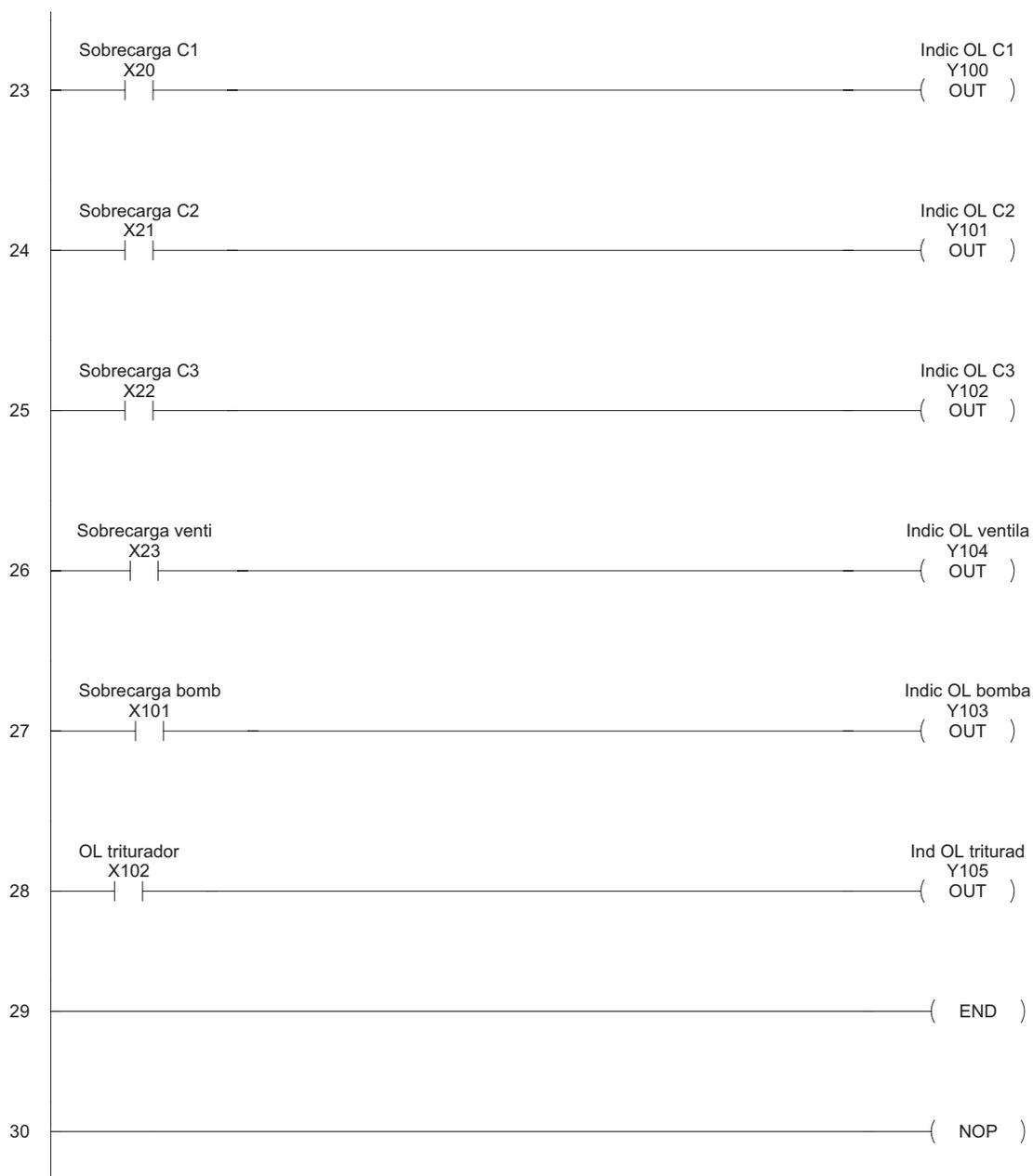
Por eso es muy deseable tener herramientas que puedan hacer la simulación del sistema. En el caso de este ejemplo, se conectó un segundo PLC DL06 con un módulo simulador (D0-08SIM) que permitió simular el proceso. Si Ud tiene solamente un PLC, puede hacer la simulación directamente en el mismo PLC, o parcialmente con **Data View**, y luego, cuando todo funciona ya en la planta, se borra el programa de simulación.

A continuación presentamos el diagrama ladder para esta parte. Hay explicaciones en cada rung. Otras explicaciones pueden ser encontradas en cuadros específicos.









El concepto de control del sistema y el correspondiente programa ladder en el PLC debe entenderse bien, entenderse como se ejecuta éste en el PLC, debe simularse y posiblemente corregir errores que puedan existir y luego obtener que éste funcione, antes de ir a una segunda revisión.

Como habíamos dicho, no están programadas las diversas paradas ni las protecciones, etc. Eso es lo que veremos en las próximas páginas. Note que estamos haciendo el programa basado en el diagrama de bloques creado inicialmente.

El programa puede ser modificado para aceptar las nuevas condiciones de parada.

El sistema debe causar paradas de los equipos por parada normal, por sobrecarga del triturador, por falla del sistema de descarga, por parada de emergencia desde el pupitre o por las cuerdas de emergencia al lado de las correas transportadoras o por defecto de uno de los motores de los equipos involucrados.

Analicemos cada caso:

### 1. Parada normal (cuando se aprieta el botón asociado a la entrada X2 en el pupitre)

Al apretar el botón X2 debe suceder que:

- a) Se para la correa C1 inmediatamente, al momento de apretar el botón
- b) Se espera un tiempo para que se vacíe de material la correa C1 (digamos 12 segundos) y se para la correa C2.
- c) Se espera un tiempo para que se vacíe de material la correa C2 (digamos 25 segundos) y se para la correa C3.
- d) Se espera un tiempo para que se vacíe de material el triturador (digamos 15 segundos) y se para el triturador y luego de 30 segundos la bomba de lubricación.
- e) Se espera que la lubricación pierda la presión para parar el ventilador del filtro de mangas.

### 2. Parada por triturador cargado sobre el límite (cuando la carga pasa de 95%)

Se espera que el triturador trabaje entorno del 90% de la carga máxima. El operador de la planta debe observar esta condición con el amperímetro del pupitre para determinar como está la carga del triturador. El PLC no puede tomar ninguna acción aquí porque no se ha definido ningún criterio para hacer ésto. En un ejemplo posterior veremos como automatizar este evento, lo que libera tareas del operador. Cuando la carga llegue a 90%, el operador debe parar la alimentación de material y para eso debe apretar un botón de parada de alimentación que debe parar las correas C1, C2 y C3 por un tiempo adecuado, que estimamos ahora en 60 segundos. Después de ese período, la correa C3 parte, luego la C2 y por ultimo la C1.

Este botón no había sido previsto al comienzo del desarrollo del proyecto y como sabemos que podemos aumentar la cantidad de entradas y salidas, lo haremos agregando un módulo D0-10ND3 y la entrada a ser usada será X100.

### 3. Parada de emergencia desde el pupitre.

En ese caso, el operador puede hacer que todas las máquinas paren inmediatamente, por alguna causa de peligro a una persona o por posibilidad de destrucción de l equipo.

En ese caso, todos los motores deben parar. Recuerde que ésta es una operación simple para mostrar como se hace la programación de un PLC en un proyecto de control. Talvez, dependiendo de otras condiciones definidas por el personal de proceso, el ventilador y la bomba del triturador no sean necesarias que paren. Para este caso simple. también se pararán estos motores.

4. **Parada por falla de la descarga.** Esto es lo mismo que una parada de emergencia. En este caso se cubre también una falla de energía eléctrica.

4. **Parada por accionamiento de la emergencia de cuerda de las correas.**

En ese caso, el PLC recibe la información de que se ha parado una de las correas y las correas “aguas arriba” deben parar inmediatamente. Será necesario que se inspeccione la causa de la parada y se tomen las medidas necesarias para reiniciar la operación nuevamente.

5. **Parada por defecto de uno o mas equipos** (típicamente una desconexión por sobrecarga)

En este ejemplo, tenemos las siguientes condiciones:

- Cada uno de los relevadores de sobrecarga de los partidores de motores.
- Si falla el aire comprimido para funcionamiento del filtro de mangas.
- Si falla la presión de lubricación del triturador.

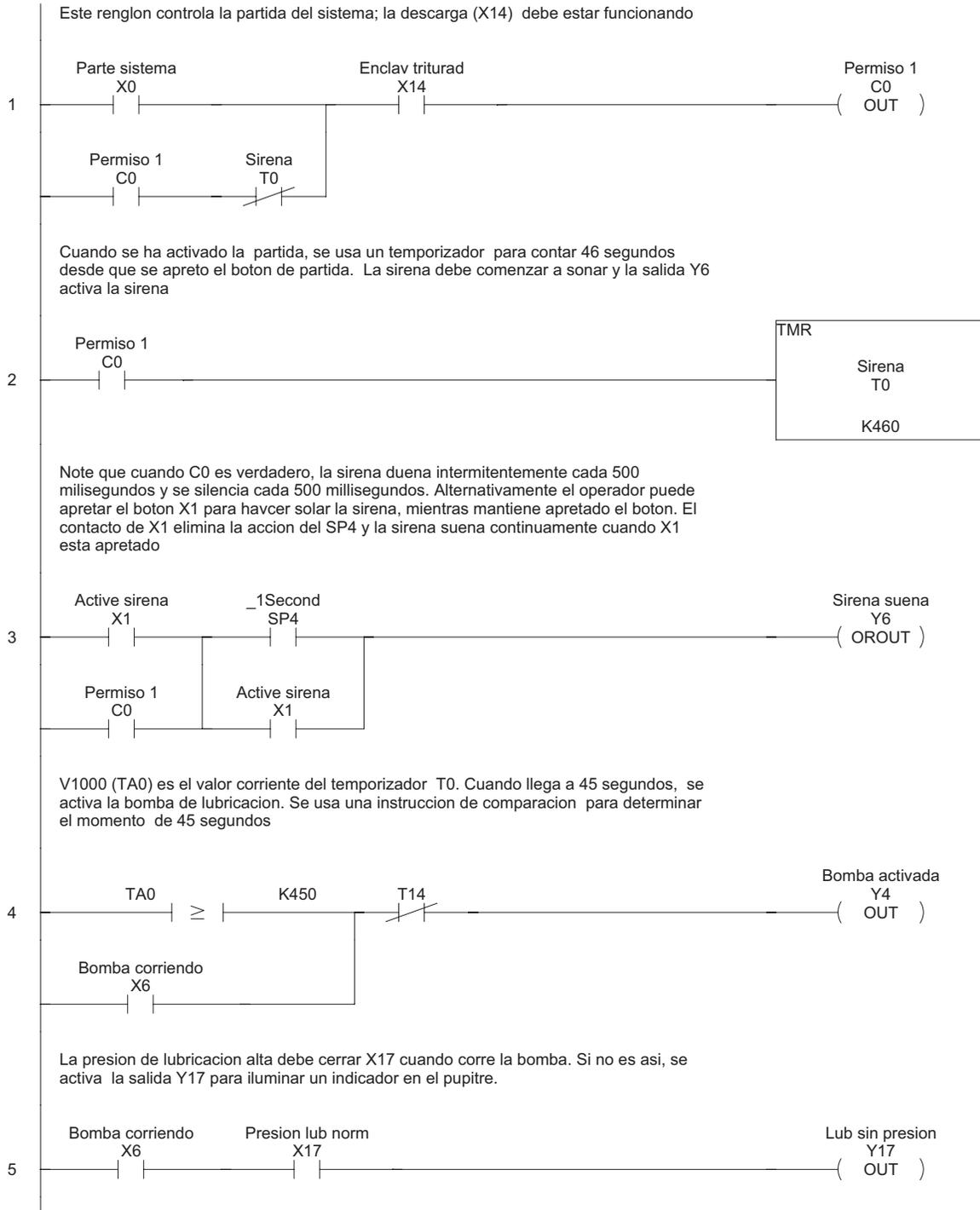
Veamos la condición en cada una de las posibilidades:

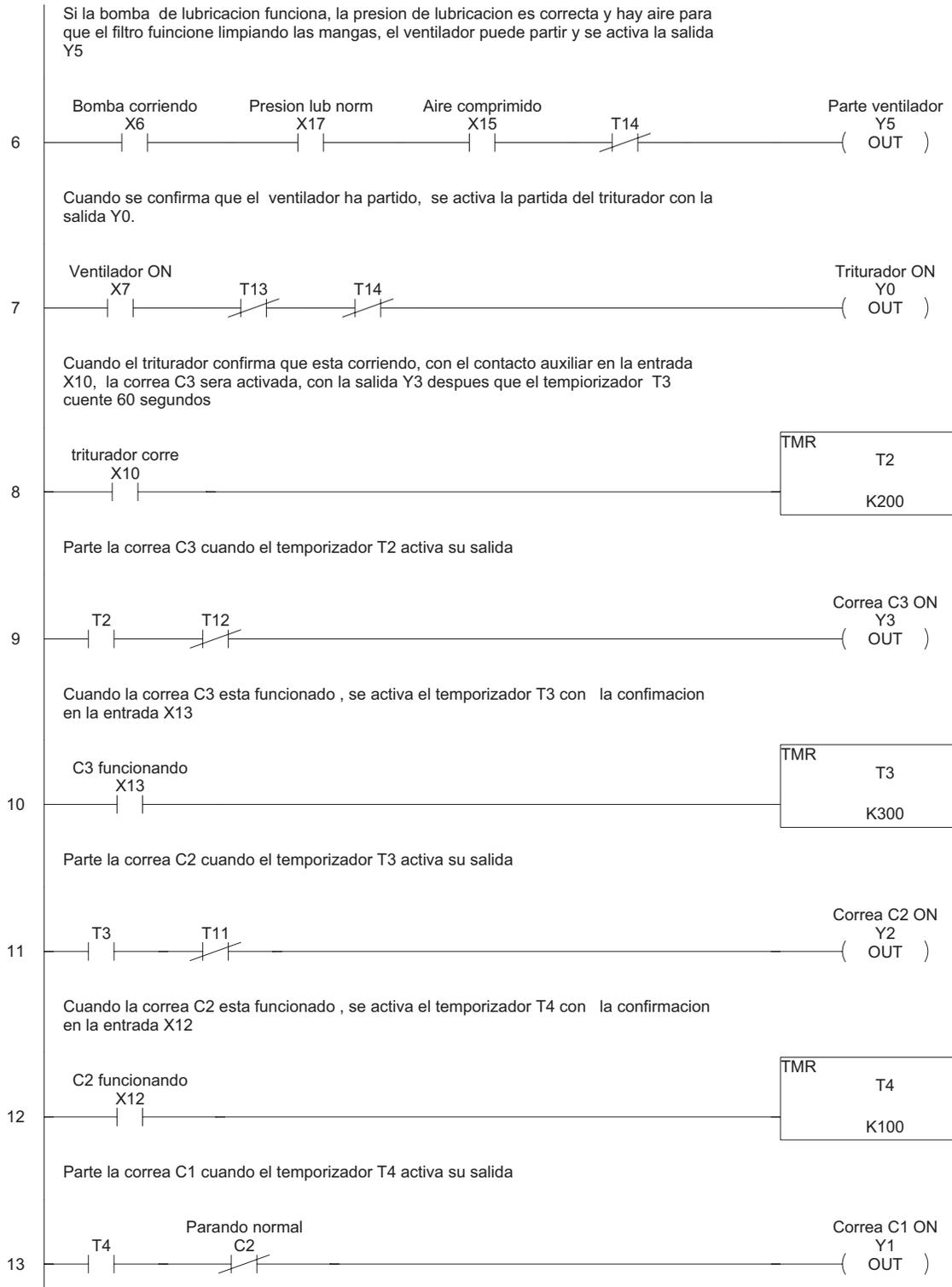
- **Desconexión por sobrecarga del triturador:** Al parar el triturador, deben parar todos los equipos “aguas arriba” inmediatamente, excepto el ventilador del filtro y la bomba de lubricación.
- **Desconexión por sobrecarga del ventilador:** Al parar el ventilador, se pierde la acción de limpieza del filtro y por lo tanto deben parar todos los equipos inmediatamente.
- **Desconexión por sobrecarga de la bomba de lubricación:** Al parar el triturador, deben parar todos los equipos “aguas arriba” inmediatamente.
- **Desconexión por sobrecarga una o más correas:** Al parar una de las correas, deben parar las correas “aguas arriba”.
- **Desconexión por falla de presión de aire:** Al no tener aire, se para la acción del filtro de mangas. Deben parar todos los equipos “aguas arriba”.

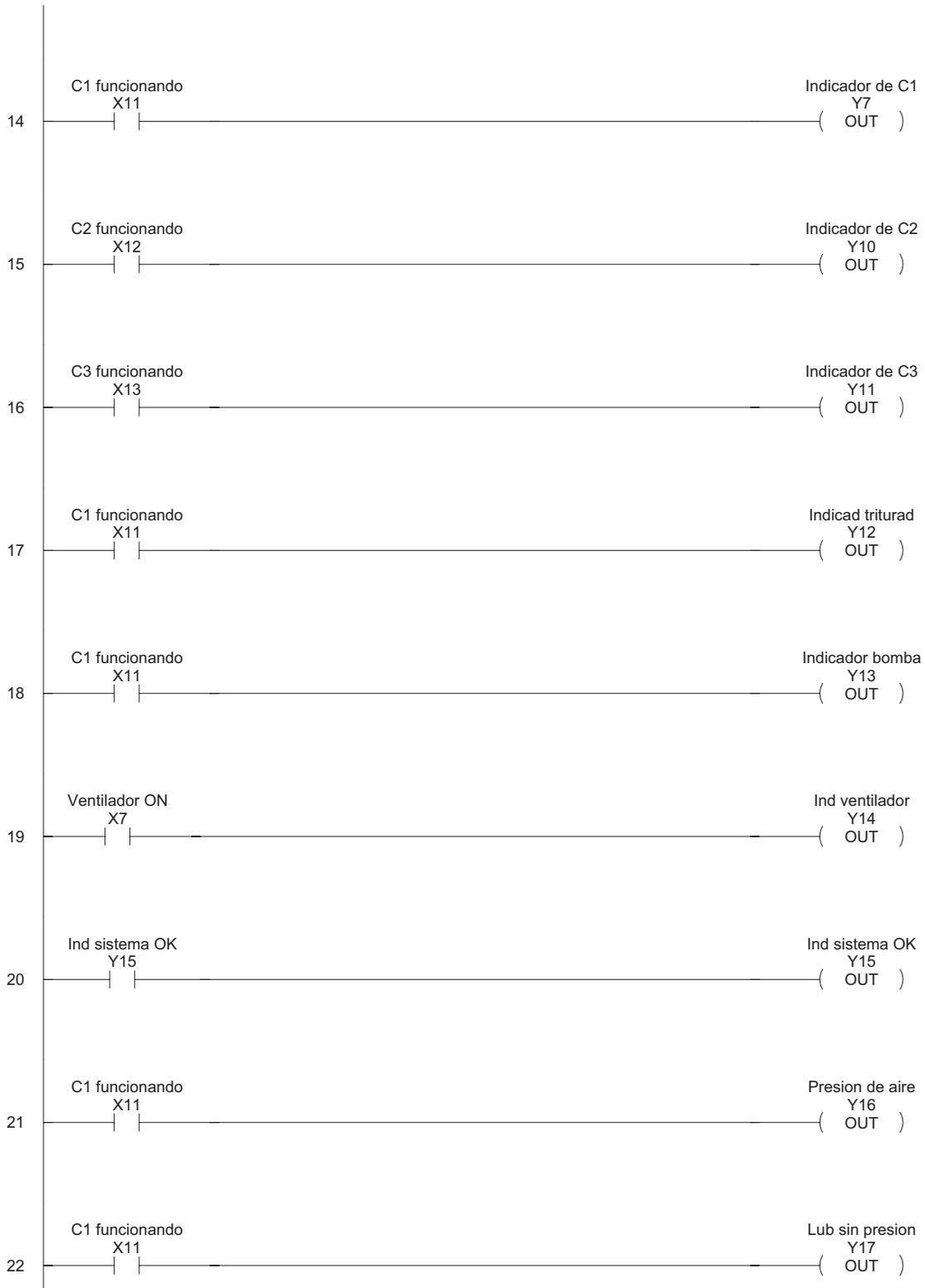
La primera corrección al programa hecho inicialmente tiene que ver con la parada normal. Crearemos un bit C16, que se hace ON al apretar el botón de parada del pupitre X2 y cuando éste esté activado, podrán ejecutarse las acciones de parada de los motores.

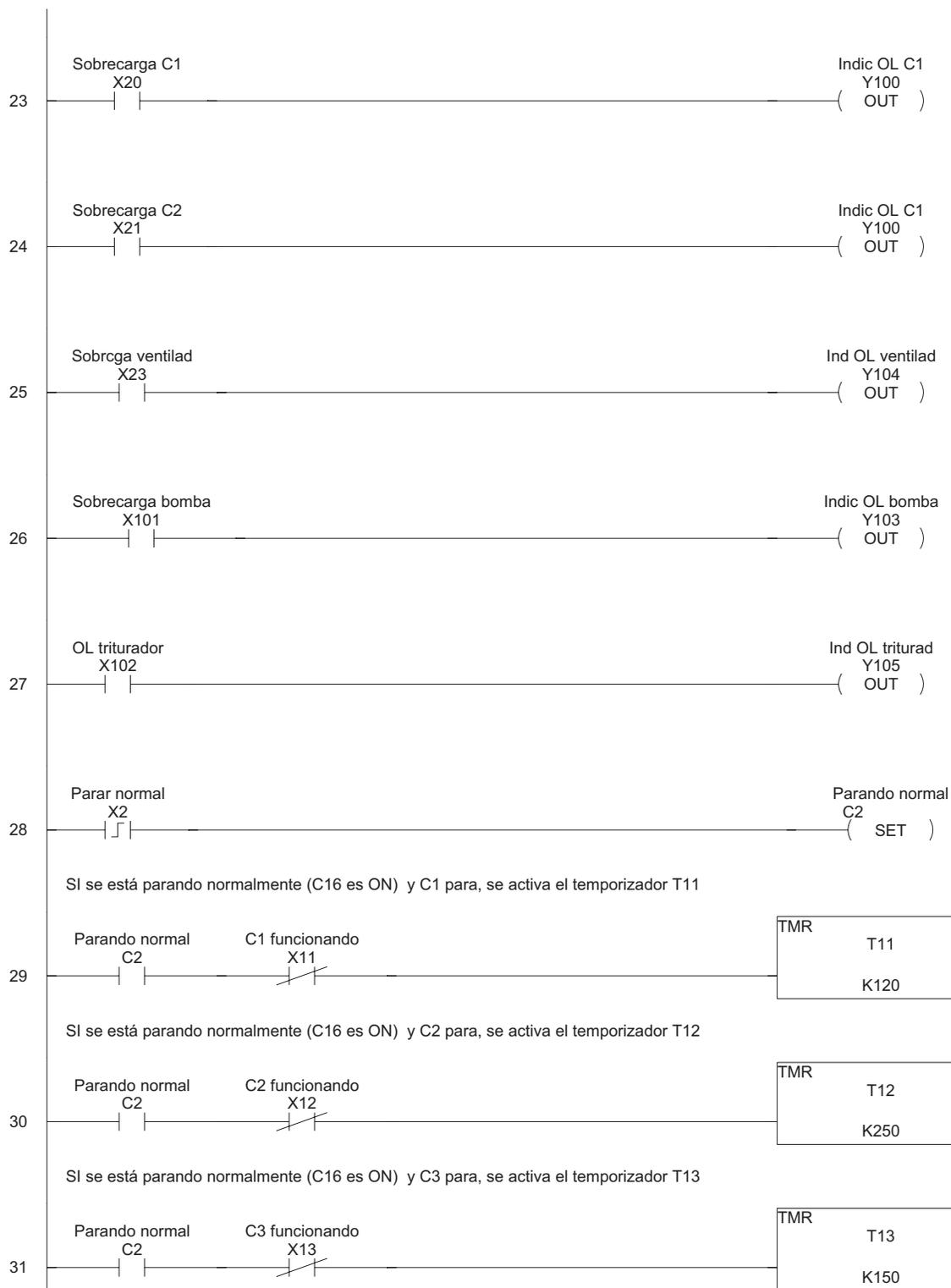
En las próximas páginas mostramos como ha sido modificado el programa para cubrir esta parte.

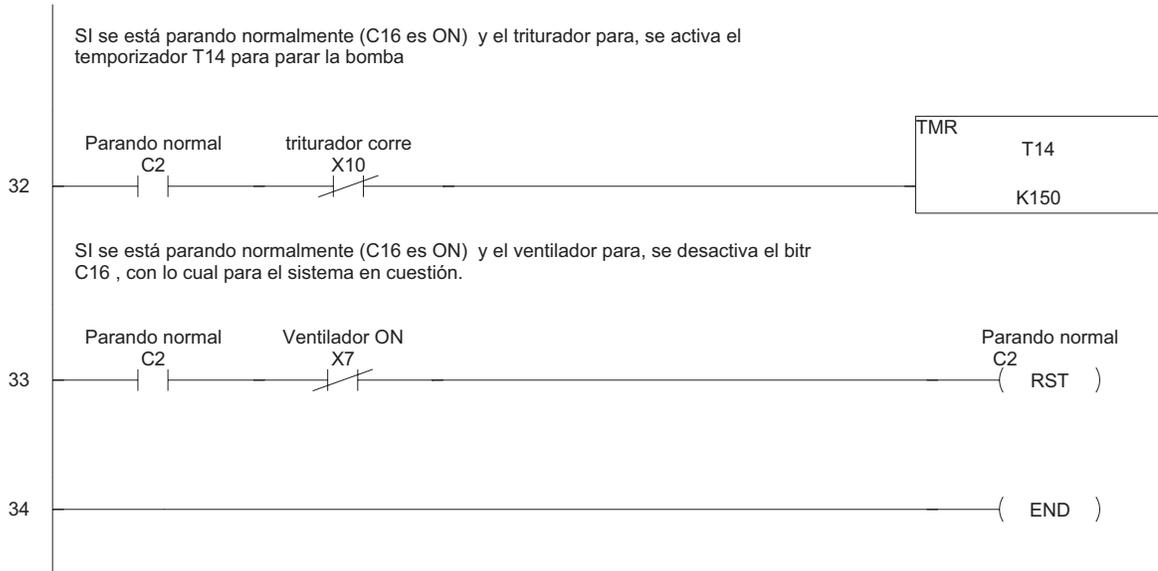
Esta parte define como fue modificado el programa ladder para incluir la parada normal.





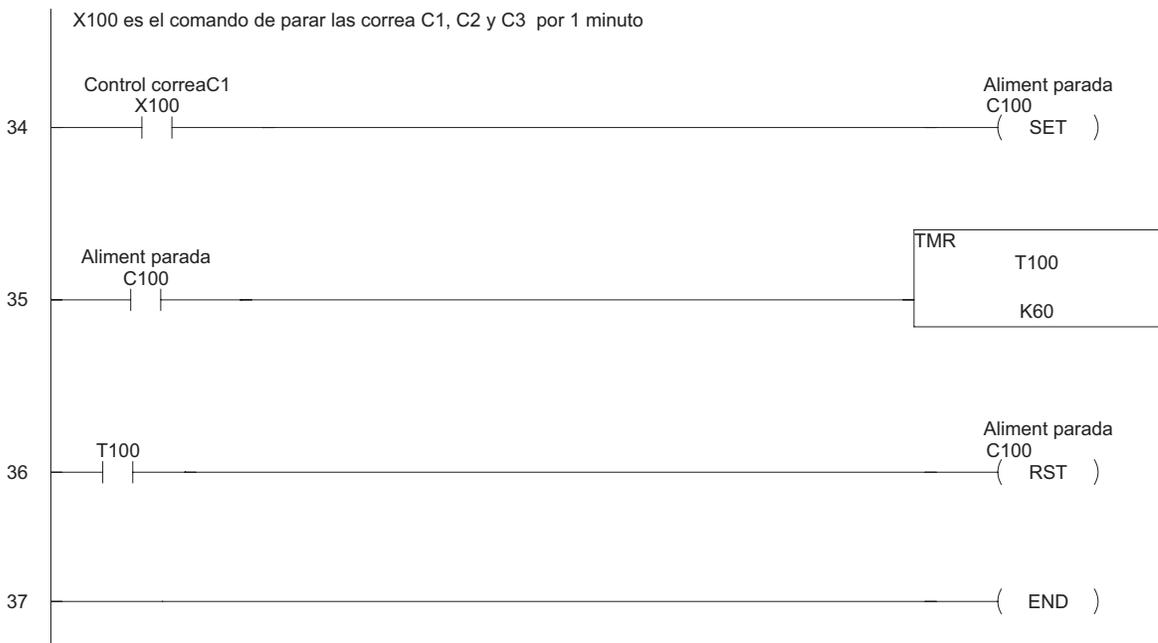






## 2. Parada por triturador sobrecargado.

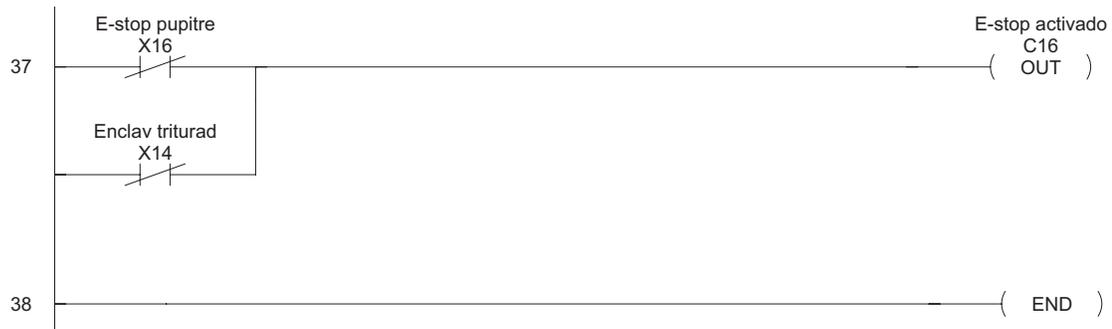
Esta condición debe bajar temporalmente el flujo de alimentación de material al triturador. Las correas C1, C2 y C3 deben parar por un cierto tiempo para que el triturador consuma el material que está procesando. Posiblemente 1 minuto puede obtener el resultado esperado, pero naturalmente este valor se podrá ajustar durante la operación, cambiando el valor en el programa. Vea qué renglones son necesarios en el próximo segmento de programa.



C100 enclava las salidas de las correas en los renglones 9, 11 y 13. No es necesario mostrar todo el programa nuevamente.

### 3. Parada de emergencia desde el pupitre o falla de la descarga

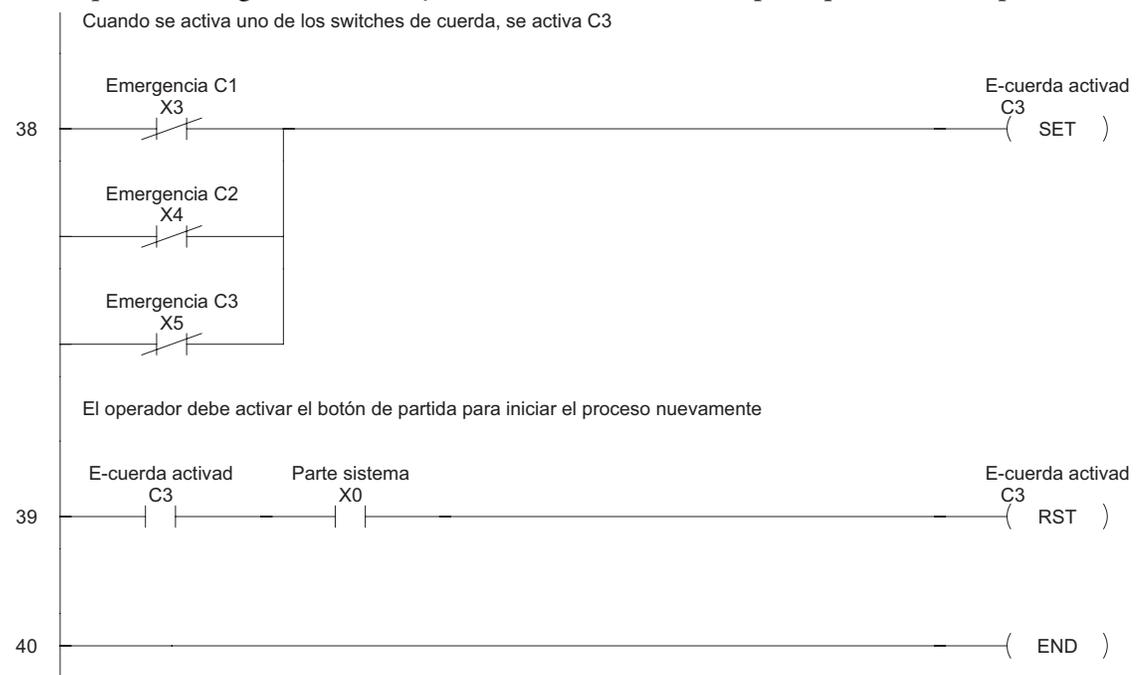
Vea que solamente es necesario colocar una condición de parada causada por el botón X16 o el botón X14. Uno de los botones activa la bobina de relevador interno C16, que abre el circuito en el renglón 4. No es necesario mostrar todo el programa nuevamente.



### 4. Parada de emergencia de cuerda de las correas transportadoras

En este caso, las entradas X3, X4 y X5 generan la condición de parada de cada una de las correas. La señal X3 para la correa C1. La señal X4 para la correa C2. La señal X5 para la correa C3.

Note que los renglones 8, 10 y 12 fueron cambiados para poder hacer parar la correa



correspondiente. No es necesario mostrar todo el programa nuevamente.

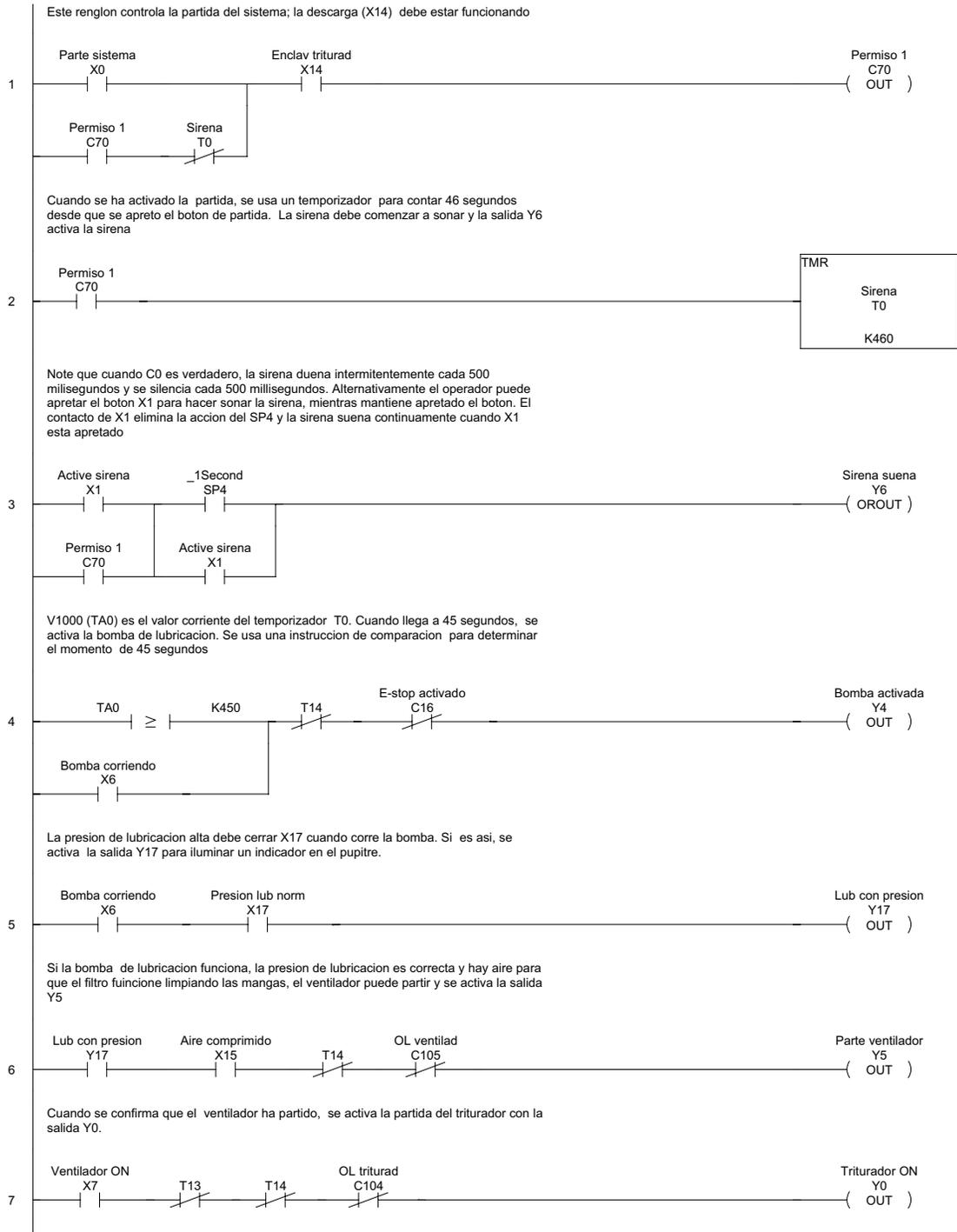
### 5. Parada por defecto de uno o más equipos

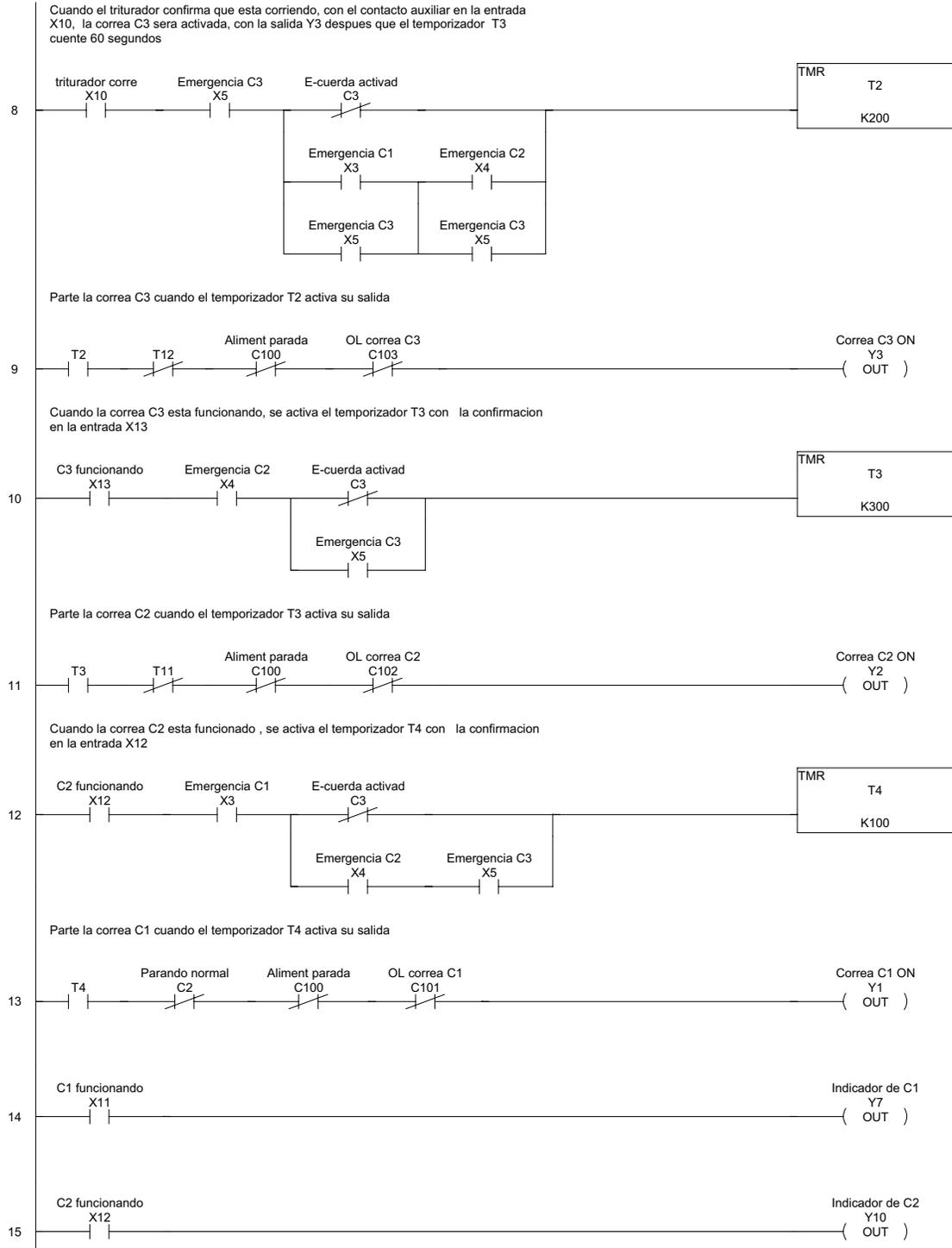
Vea en el diagrama de las próximas páginas como fueron implementados estos eventos.

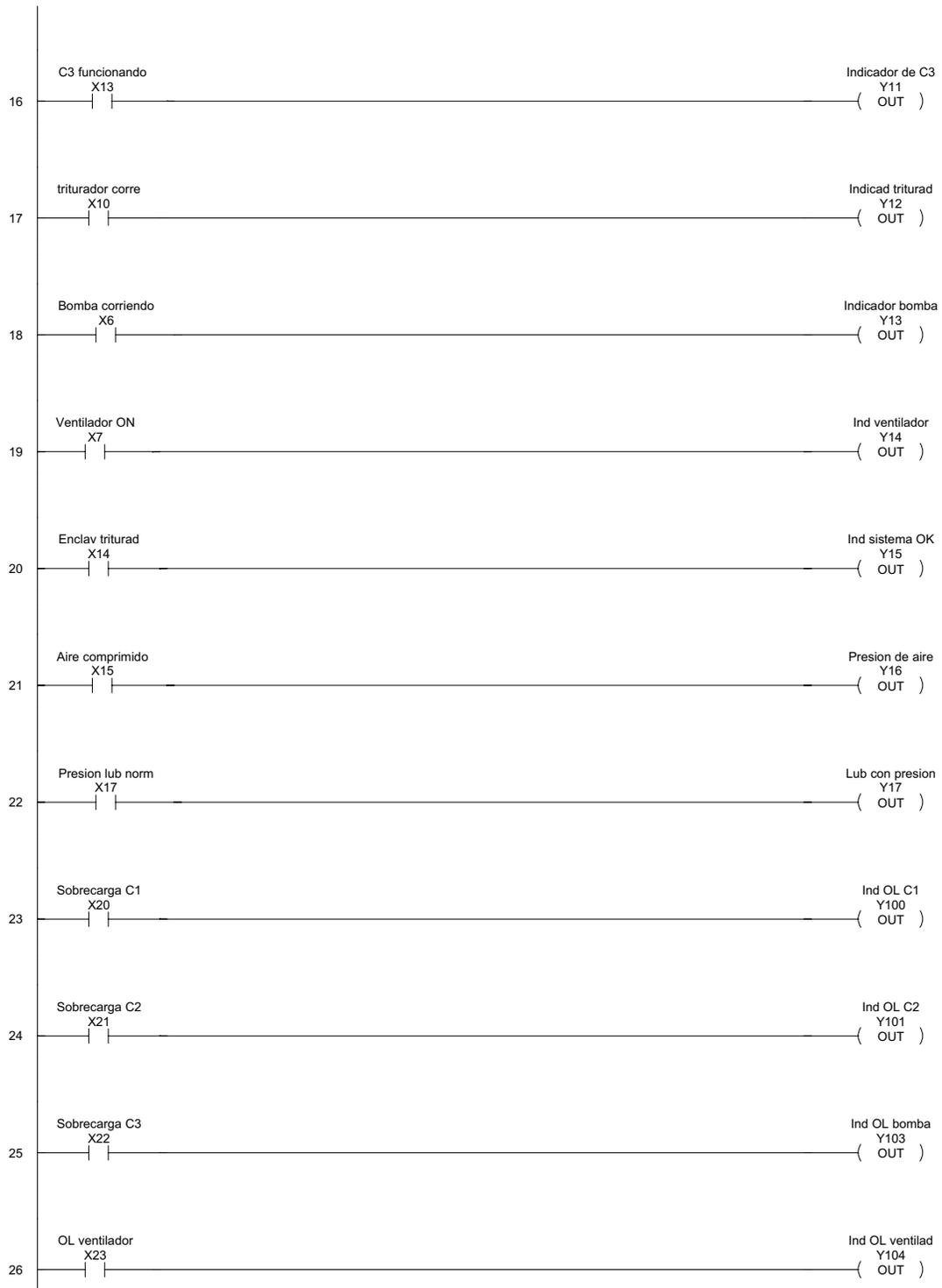
Éstos son:

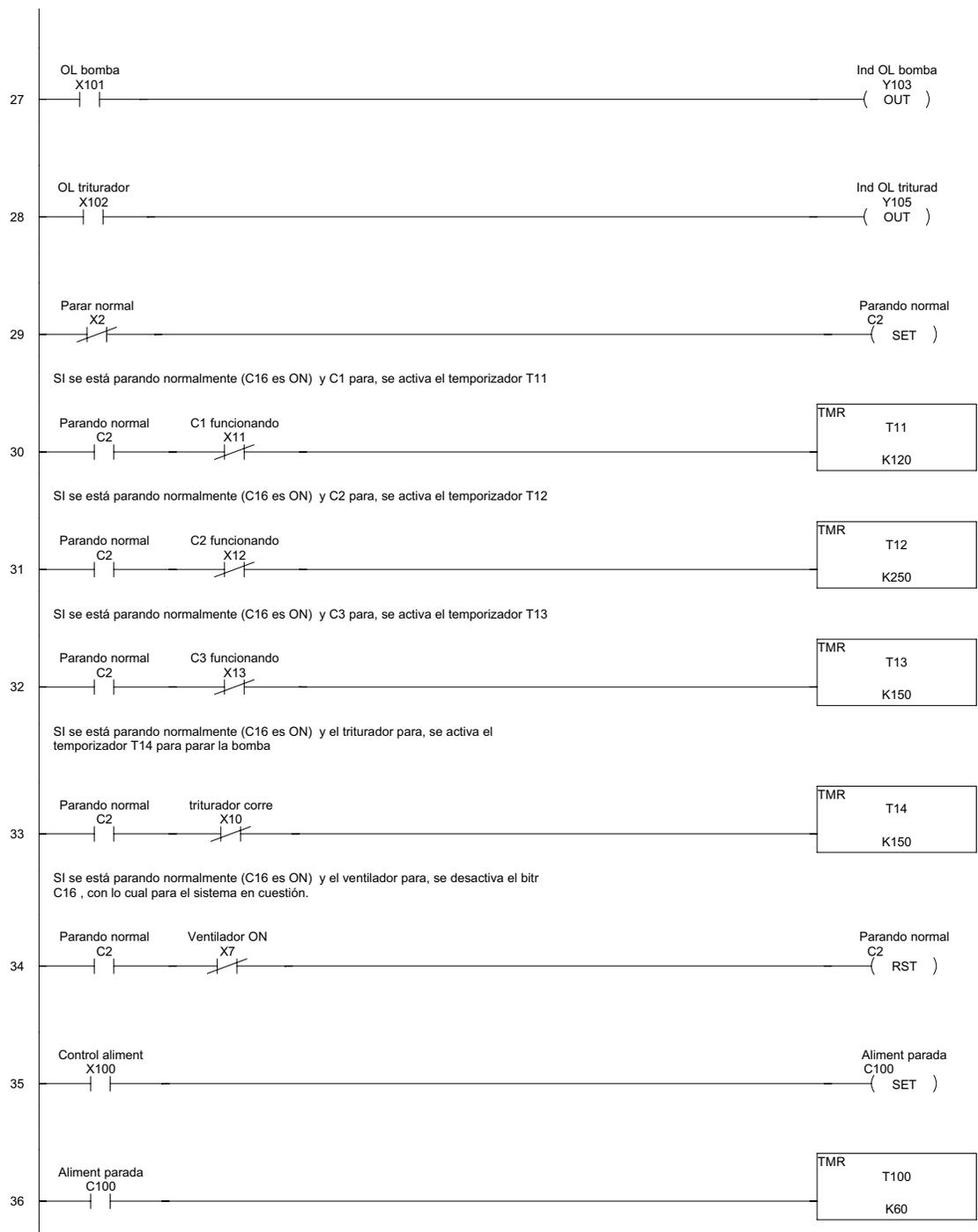
## Capítulo 11: Recopilación y ejemplos

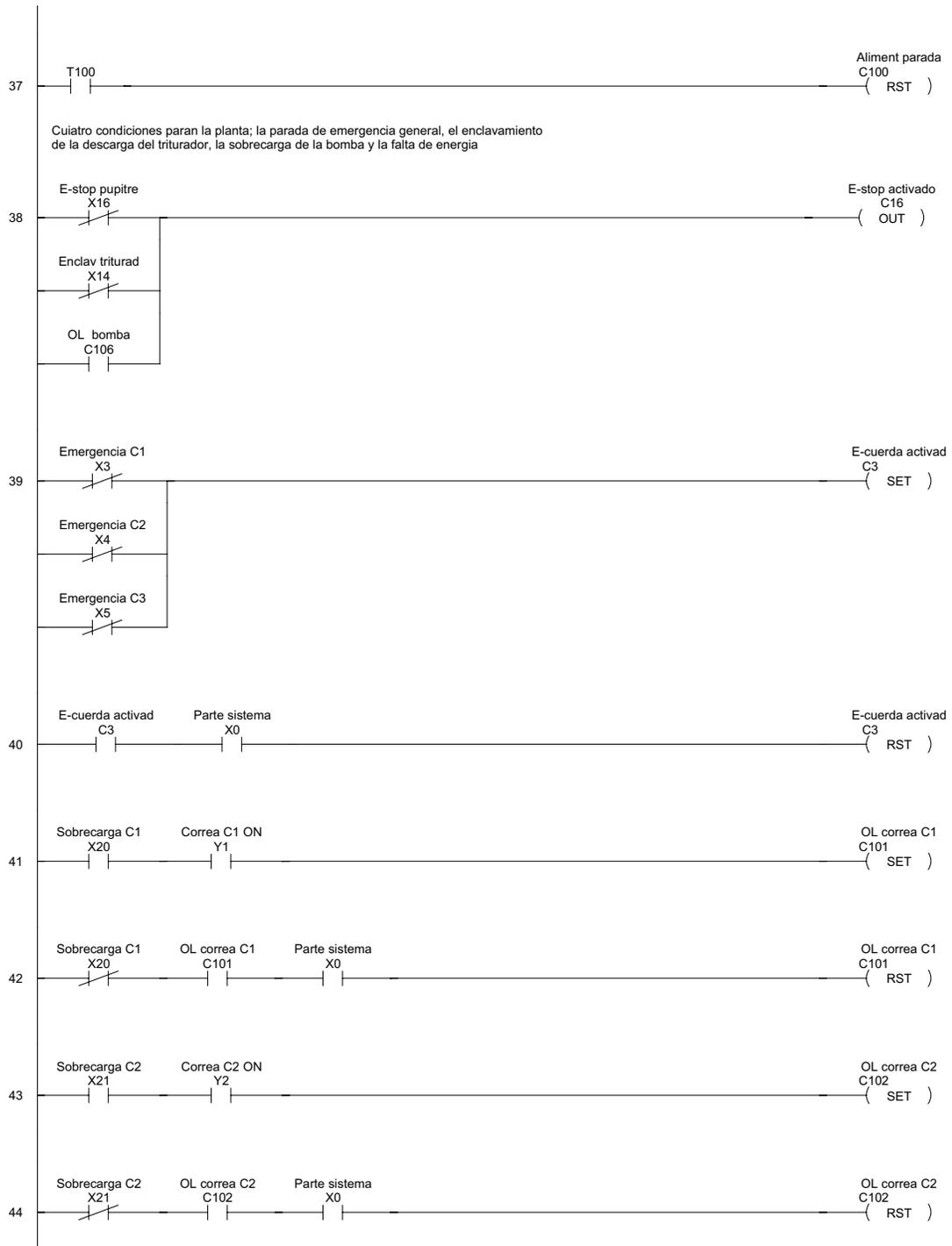
- sobrecarga de cada uno de los motores; en los motores cada confirmación debe causar un enclavamiento con el sistema y parar los demas equipos.
- falla del aire comprimido para la limpieza de las mangas del filtro. Debe causar la parada general de la planta.

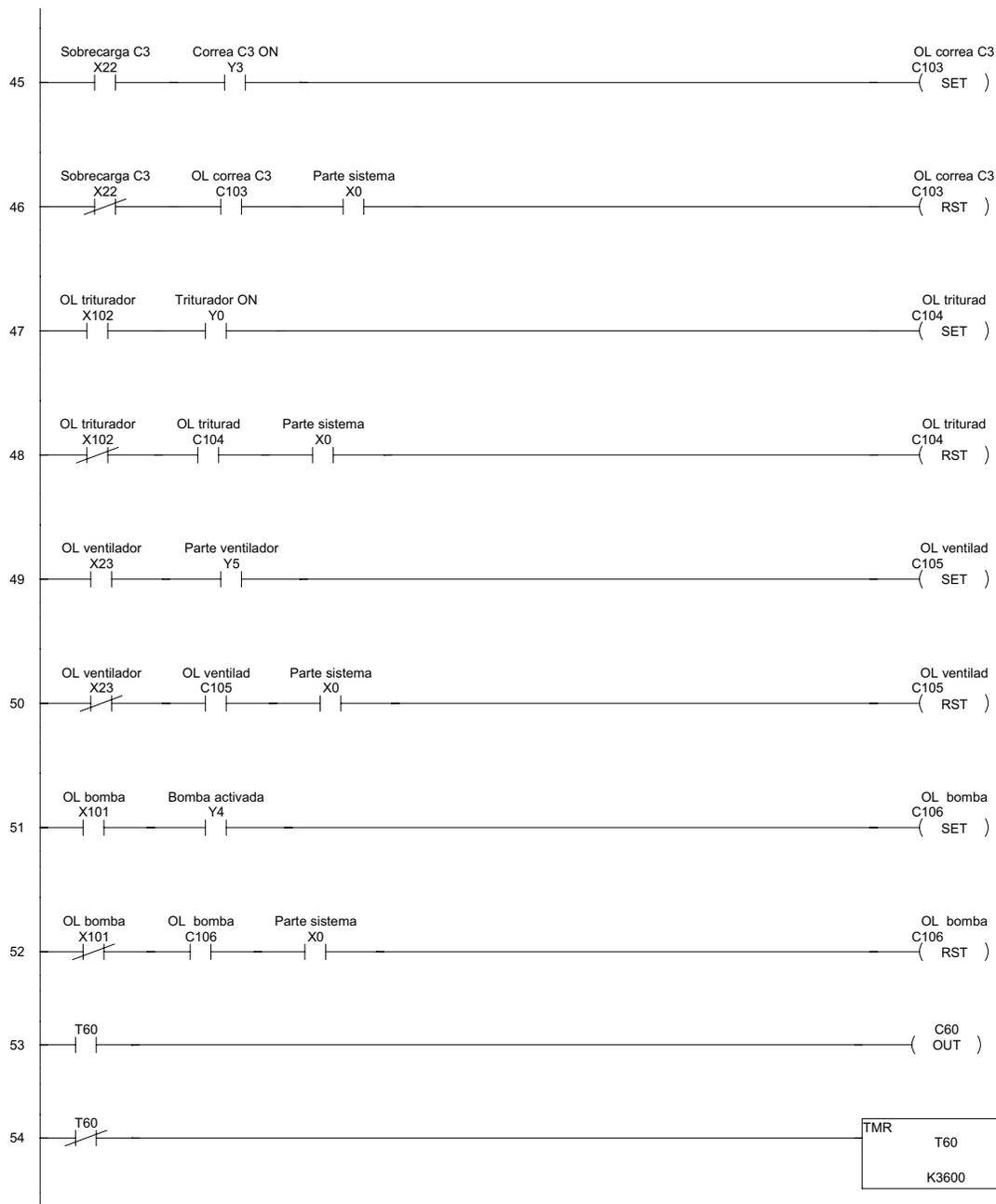


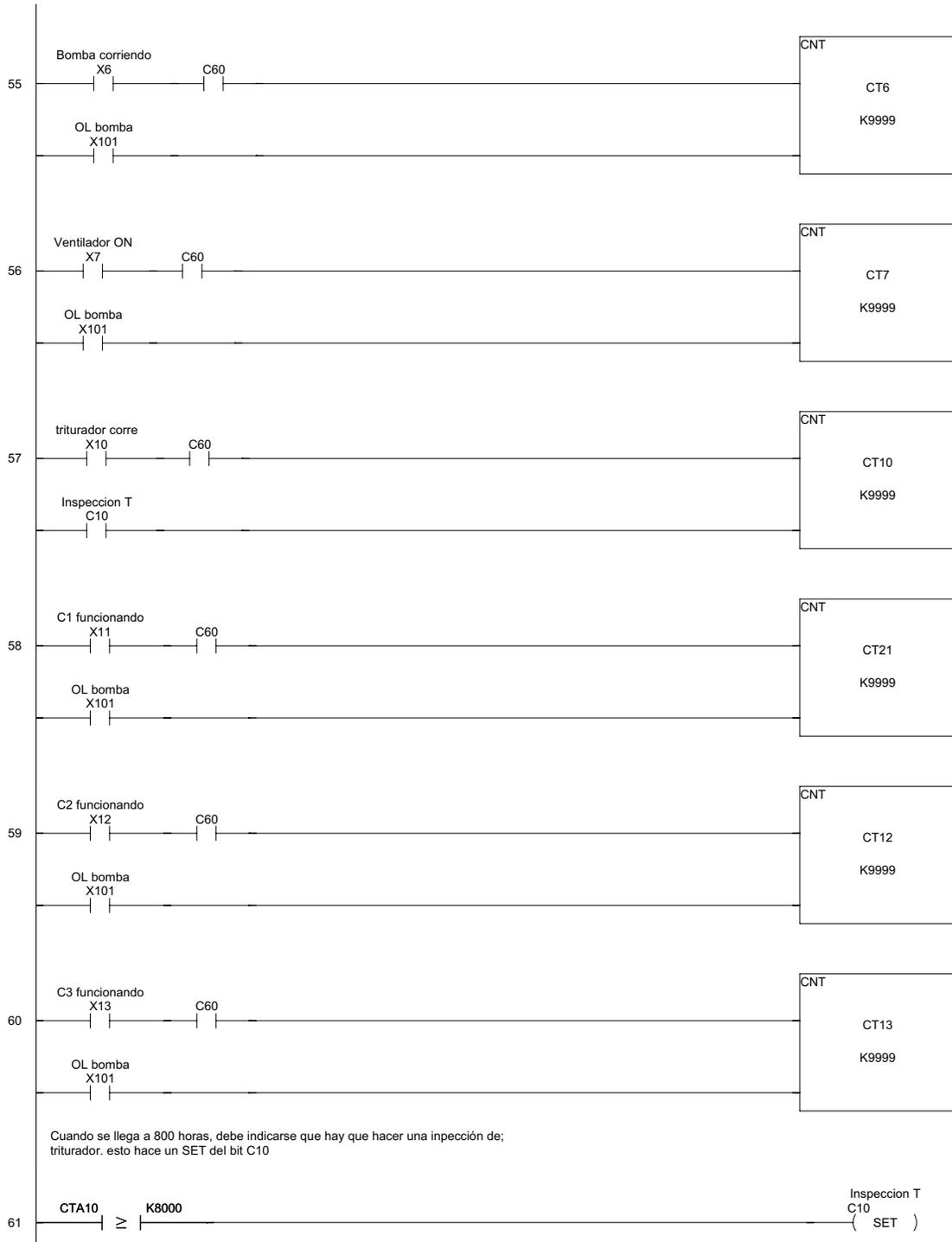


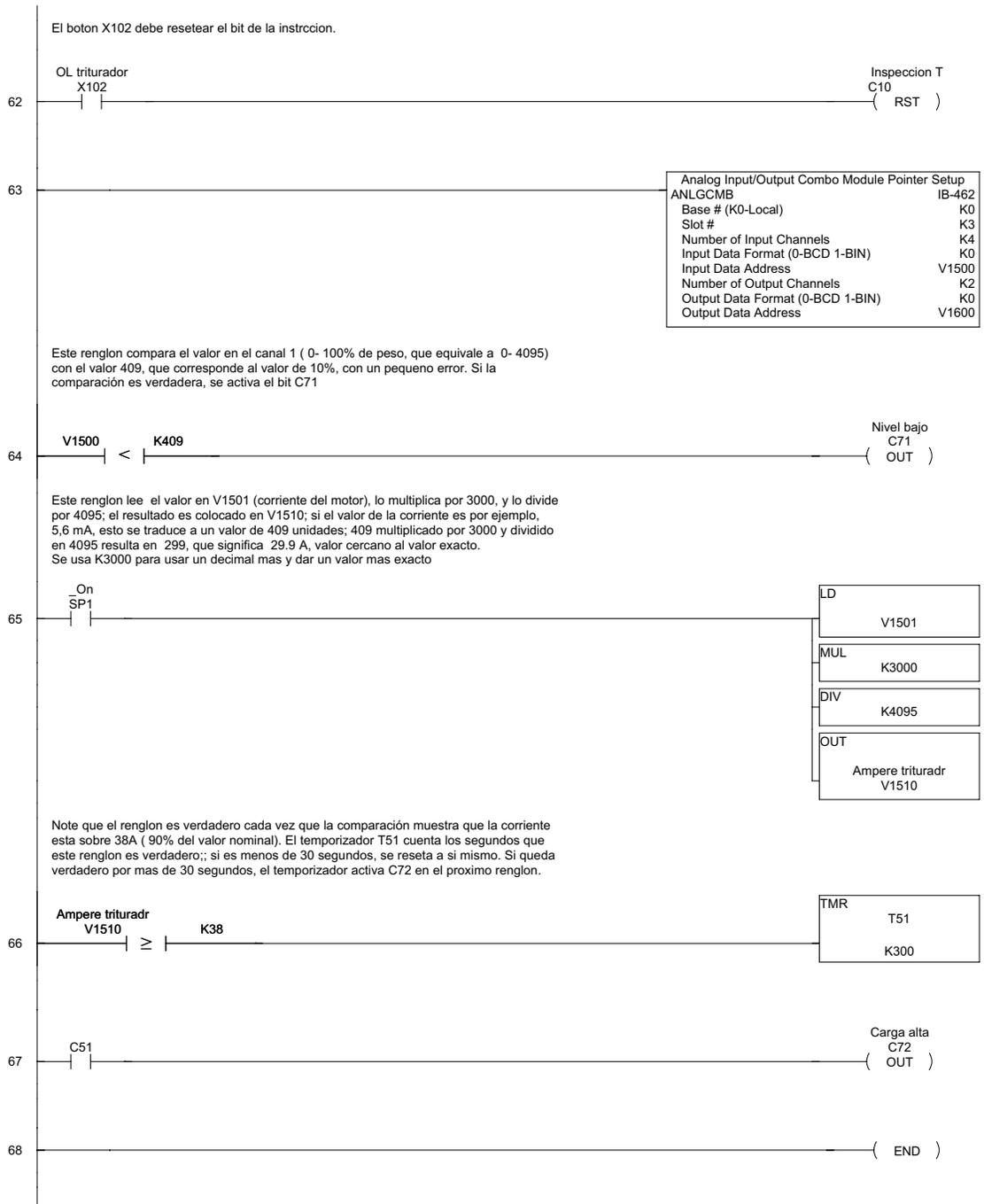












## Capítulo 11: Recopilación y ejemplos

Pasamos ahora a desarrollar otros conceptos requeridos en la implementación de programas con el PLC DL06.

**Ejemplo 2:** Consideremos usar contadores en el mismo ejemplo. Una función bastante normal es medir las horas que un motor ha funcionado, por ejemplo, para ayudar al departamento de mantención a saber cuando es necesario revisar la grasa en los rodamientos de los motores, o para determinar cuando es el día que corresponde a una inspección después de 1000 horas, por ejemplo.

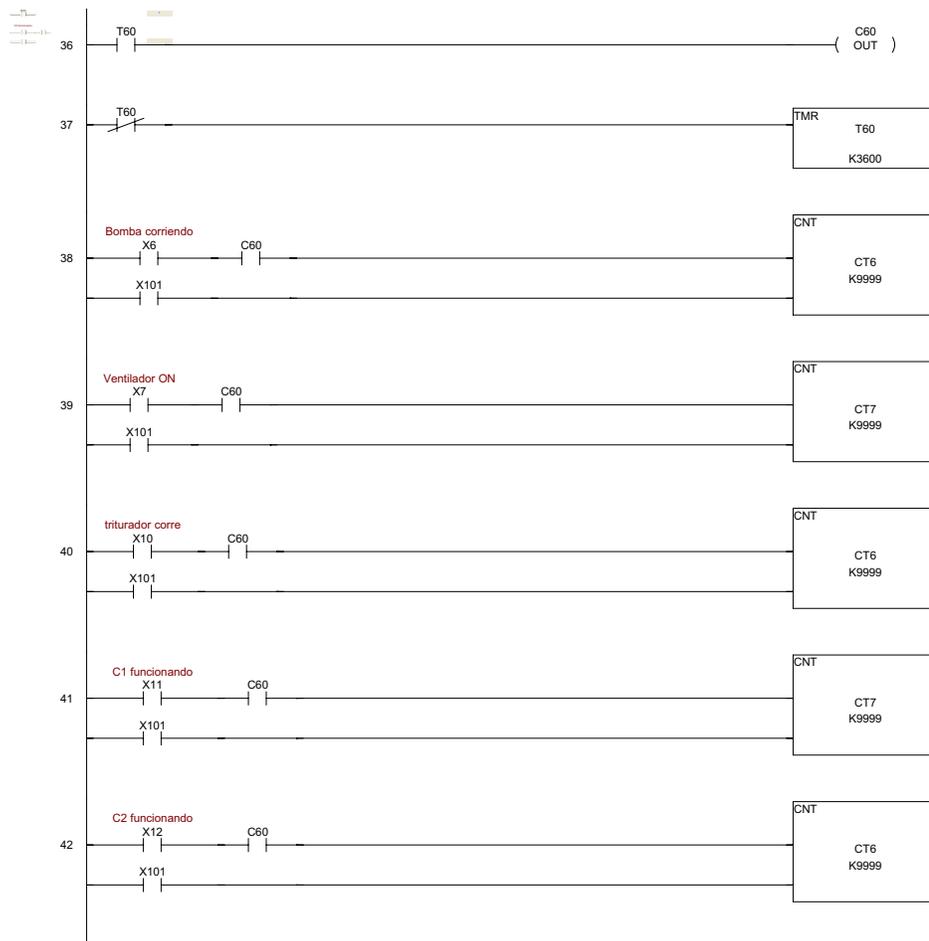
Esto no es un valor exacto hasta el segundo de modo que no sería necesario contarlos, y en ese caso usaremos cada décimo de hora como unidad.

Cada confirmación de funcionamiento de un motor tendrá asociado un temporizador. Recuerde que hay 256 temporizadores y 128 contadores en este PLC. Hasta ahora se han usado solamente unos 10 temporizadores y ningún contador.

Crearemos primero un generador de un pulso con C60 a cada 6 minutos (1 décimo de hora) con el temporizador T60. Luego cada entrada de confirmacion de motor corriendo recibe un contador que cuenta en décimos de hora;

Si el contador CT6 cuenta 134 pulsos, esto significa que el motor de la bomba ha corrido 13.4 horas. y así sucesivamente.

Note que podemos contar solamente hasta 999 horas y 9 décimos; esto equivale a unos 41 días;

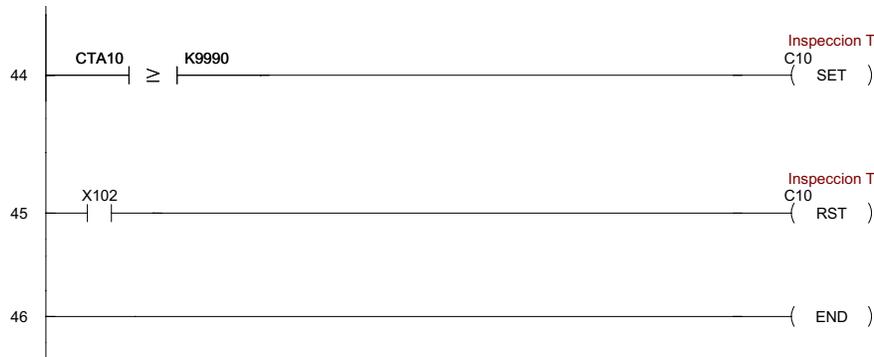


posiblemente sea interesante tener más horas. En ese caso se puede usar el contador UDC, que permite contar hasta 99999999 pulsos. Esto se ha implementado en el caso del triturador.

Note que deberíamos crear un nuevo botón, el botón X101, no previsto en la especificación inicial, para llevar a cero los contadores. También se podría hacer un “reset” del valor cada Lunes a las 6 :00 de la mañana, con el reloj que tiene el PLC. Dejaremos esto para el lector.

**Ejemplo 3:** Usando la cantidad de horas acumuladas, podemos hacer comparaciones, para mostrar como se operan con esas instrucciones.

El bit C10 puede ser conectado a una salida real (que no ha sido considerada) o a la sirena, con un pulso de 100 ms, cada minuto, para avisar que es tiempo de inspeccionar el triturador. Cuando eso haya sido hecho, el electricista (no el operador) debe apretar el botón X102 para resetear el bit y parar el aviso. Hay varias otras formas. Solamente se muestra una posibilidad.



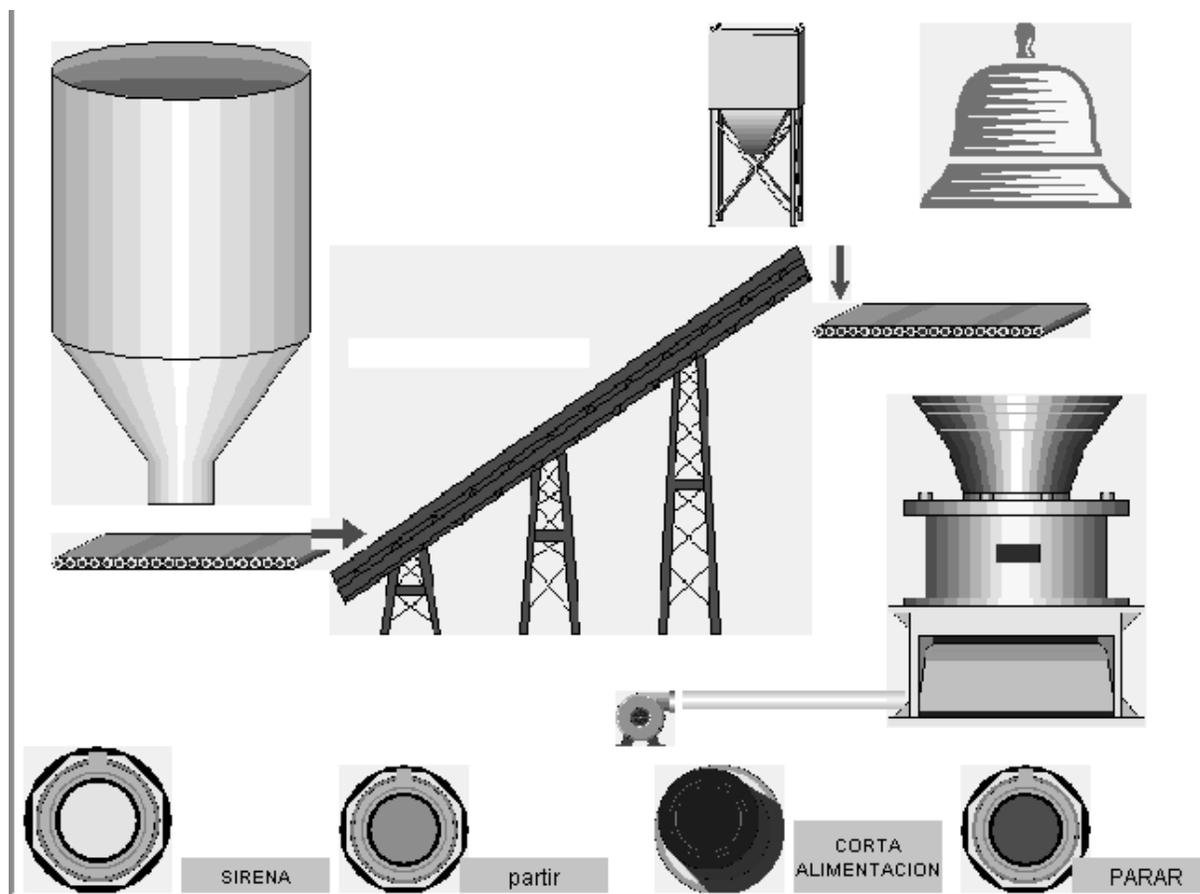
**Ejemplo 4;** Digamos que se requiere que el operador tenga más informaciones del comportamiento del sistema y para eso instalaremos una interface de operador C-more de 10 pulgadas EA7-T10C, lo que permite tener mas adquisición de datos; los botones e indicadores con lámparas serán reemplazados en ese panel. Observaremos que esta acción libera ahora entradas y salidas físicas del PLC.

En este caso, crearemos una pantalla en el panel con las siguientes asociaciones:

- X0-cambiando a---> C0=====>Objeto bitmap botón “Partir”.
- X1-cambiando a---> C1 =====>Objeto bitmap botón “Sirena”.
- X2-cambiando a---> C2 =====>Objeto bitmap botón “Parar”.
- X6----->Objeto bitmap =====>Bomba corriendo.
- X7----->Objeto bitmap =====>Ventilador corriendo
- X10-----> Objeto bitmap =====>”Triturador corriendo”.
- X11-----> Objeto bitmap =====>” Correa C1 funcionando”.
- X12- ----->Objeto bitmap =====>.”Correa C2 funcionando”.
- X13----->Objeto bitmap =====>”Correa C3 funcionando”.
- X14----->Objeto bitmap =====>”Descarga funcionando”.

- X15----->Objeto bitmap =====>"Presión de aire correcta".
- X17----->Objeto bitmap =====>"Presión de lubricación correcta".
- X20----->Objeto bitmap =====>"Sobrecarga Correa C1".
- X21----->Objeto bitmap =====>"Sobrecarga Correa C2".
- X22----->Objeto bitmap =====>"Sobrecarga Correa C3".
- X23----->Objeto bitmap =====>"Sobrecarga ventilador".
- X100----->Objeto bitmap =====>"Corta alimentación". (un botón)
- X101----->Objeto bitmap =====>"Sobrecarga de bomba".
- X102----->Objeto bitmap =====>"Sobrecarga del triturador".

Ésto se muestra en la figura a continuación:



Como puede notar, el uso de un panel de interface de operador puede mostrar el proceso de una forma mucho más clara que usando solamente indicadores, no se necesitan usar tantas salidas físicas para encender las lámparas de los indicadores, se puede conectar solamente un cable a PLC en vez de hacer un alambrado punto a punto. la pantalla ocupa en espacio menor. Se deja todavía el botón de parada de emergencia principal por razones de seguridad.

La conexión al PLC es hecha a través del puerto 2, como conexión serial RS-232. Se pueden crear más pantallas con gráficos de tendencia, alarmas, etc.

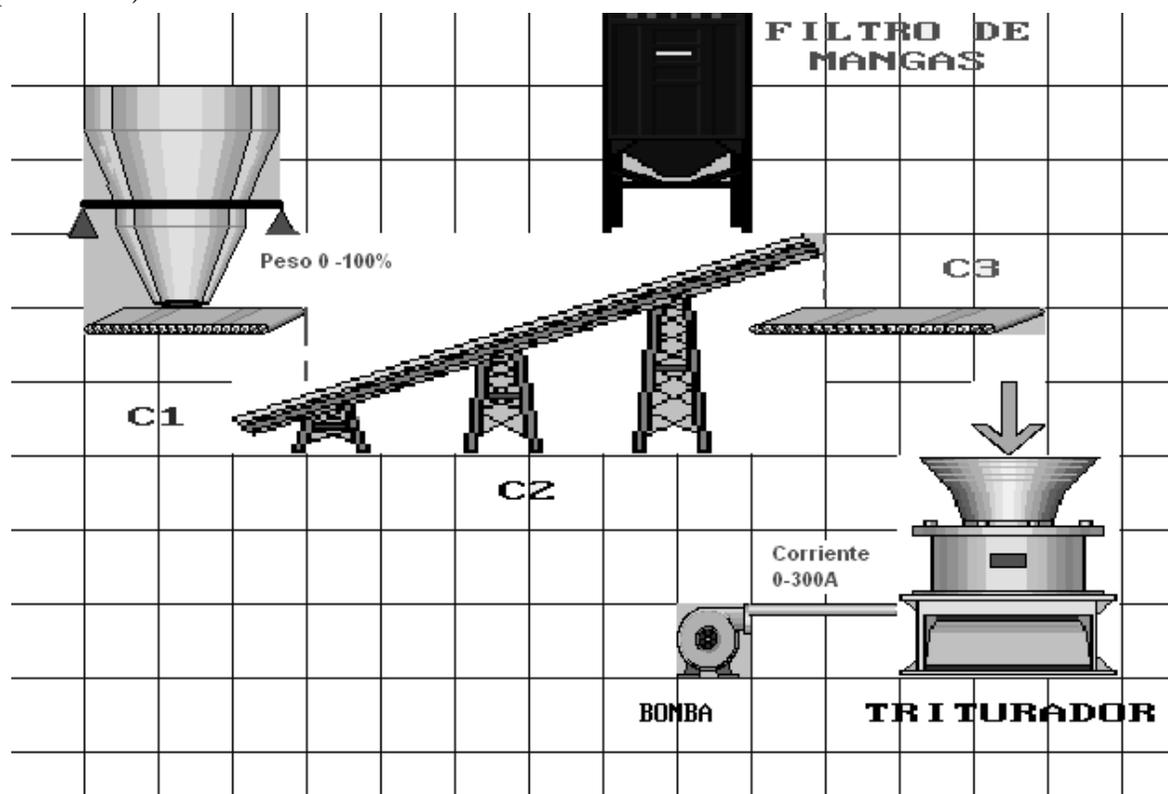
Vea más detalles en el manual del panel C-More.

**Ejemplo 5:** Hagamos ahora un ejemplo en que queremos analizar el comportamiento de señales análogas. Utilizando un transductor de corriente podemos ver el valor de corriente en el motor del triturador y determinar el nivel de mineral en el “Depósito de mineral” en la alimentación para evitar que las correas y el triturador corran vacíos. En ese caso, aceptaremos una señal de menos de 10% del nivel como depósito de mineral vacío y la corriente del motor por 30 segundos menor que 50 % de la corriente nominal indicará que el sistema está corriendo sin mineral. Debe ser generada una alarma. Veamos primero que necesitamos en el sistema.

- Un transductor con señal de 4-20 mA que indique el peso del material en el depósito de mineral (u otro similar tal como nivel). 20 mA indica 100% del peso de mineral.
- Un transductor de corriente 4-20 mA que indique la corriente del motor del triturador, como medio de indicar el consumo. El motor tiene una corriente nominal de 42 A y el partidor usa un transductor de corriente con relación 0-300A/4-20 mA.

A continuación tenemos un diagrama, que muestra el arreglo de este ejemplo.

Los transductores serán conectados a un módulo de entradas análogas instalado en una de las ranuras del PLC DL06, tal como el F0-04AD2DA-1 (Salidas a ser usadas por otras aplicaciones).



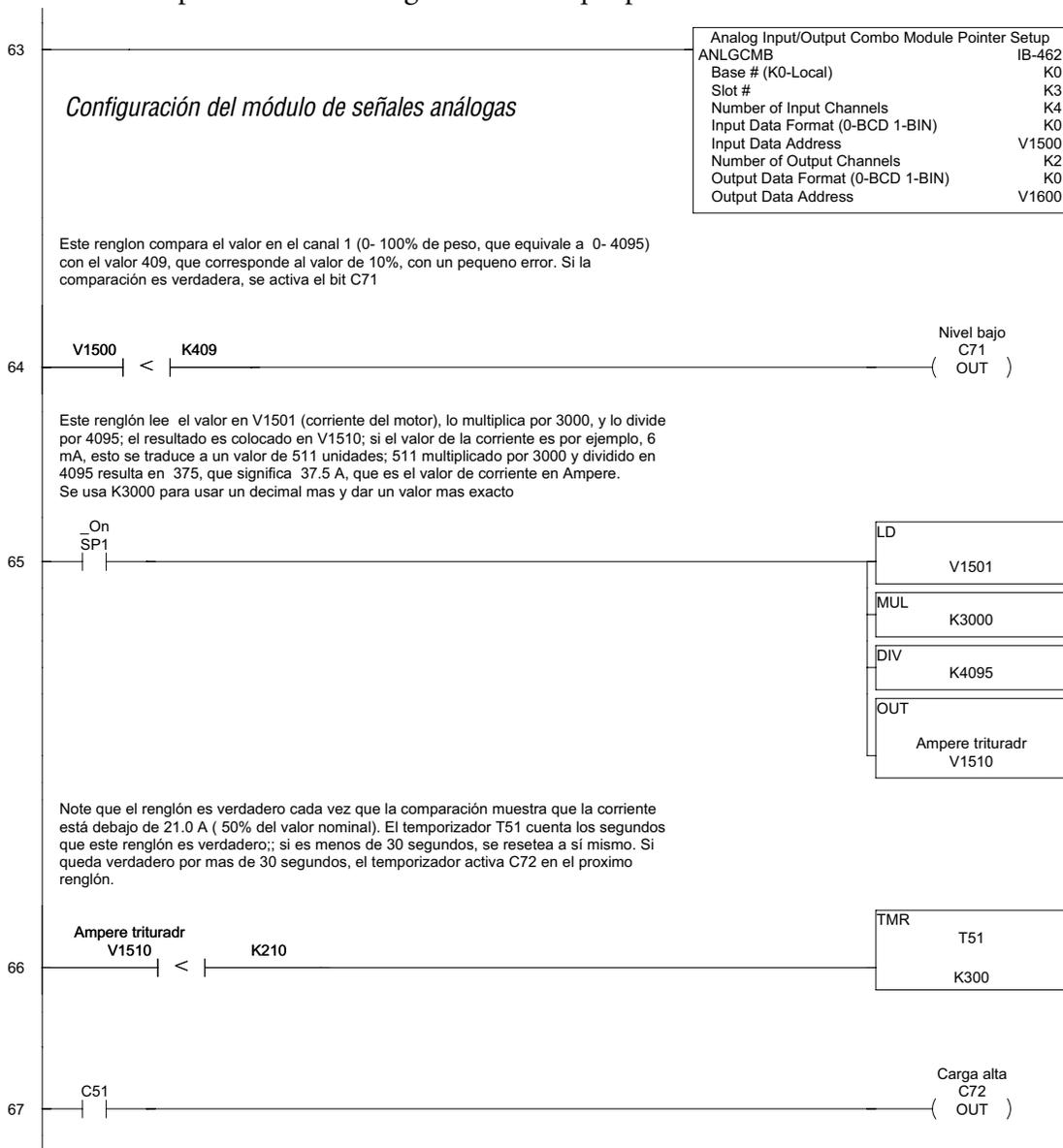
## Capítulo 11: Recopilación y ejemplos

Este módulo debe ser configurado en el PLC para que el PLC sepa en qué ranura está colocado, en que registros se van a leer los datos y en qué formato van a ser leídos los niveles de señal.

Seleccionaremos el rango V1500 hasta V1503, correspondiente a los canales 1, 2, 3 y 4 para leer los datos en BCD. Vea más explicaciones en el programa. V1500 es el nivel de material y V1501 es la corriente del motor

Usamos una comparación para determinar que el nivel está a 10% o menor para activar una alarma. Esta salida es directa; en el caso de la corriente del motor del triturador, es necesario colocar un temporizador que cuente al menos 30 segundos para generar esa alarma. Usamos también una codificación para hacer escala para presentar el valor de corriente directamente en décimos de Ampere.

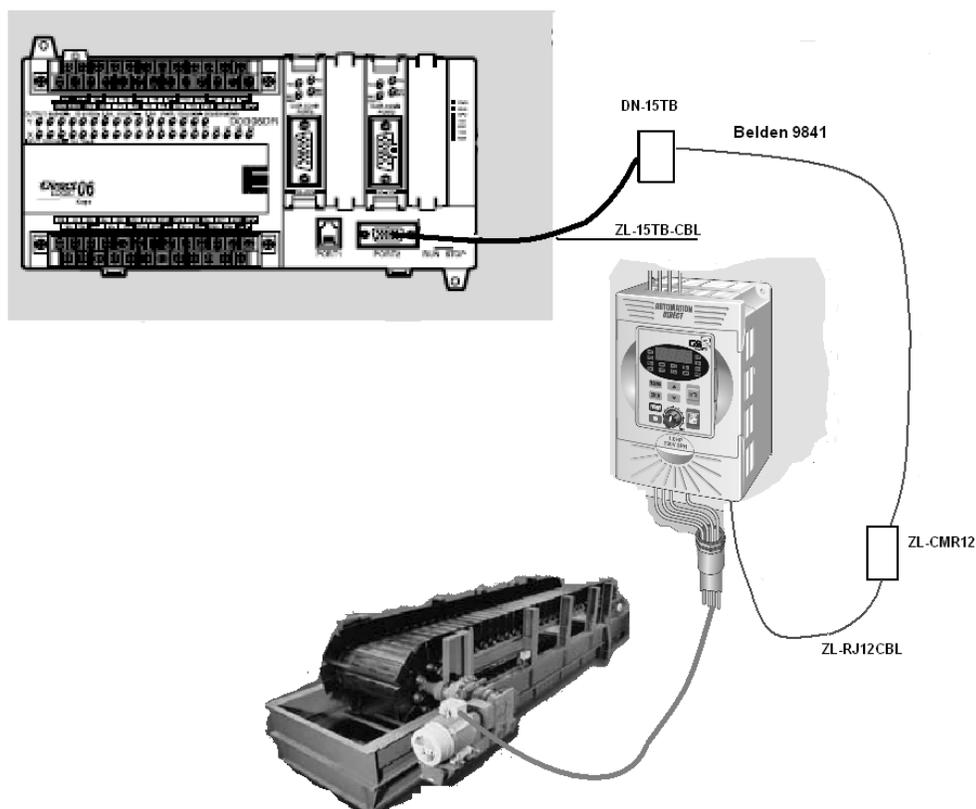
A continuación presentamos el diagrama ladder que permite hacer este control.



**Ejemplo 6:** Digamos que queremos controlar el flujo de material de la correa alimentadora C1 ajustando la velocidad en un rango de velocidad de 40 a 100% y para eso colocaremos un controlador en el motor de la correa. Éste tendrá un variador de frecuencia GS2-23P0 en el motor de la correa.

Este variador de frecuencia será conectado como esclavo en una red MODBUS RS-485. El PLC usará el puerto 2 para transmitir la información. Instalaremos el variador de frecuencia cerca del motor para evitar problemas en la transmisión de datos. Si se ha ocupado el puerto 2 del PLC DL06 para otro uso, se podría usar un módulo D0-DCM para usarlo como maestro en la red al variador. Vea mas datos del variador de frecuencia en el manual que existe en español en nuestro sitio de Internet.

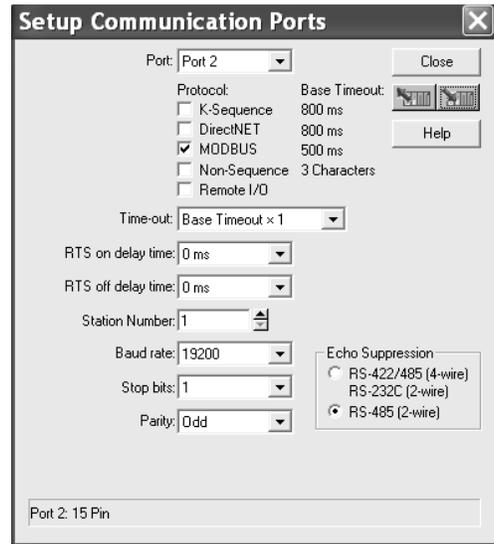
Necesitamos escribir a los registros de la referencia de velocidad, dirección 42331, y al registro de comando de partir y parar, en la dirección 42332. Los otros parametros tales como aceleracion, datos del motor, etc, se colocan directamente en el variador. Solamente le informaremos al variador cuando partir y parar y a qué velocidad. El maestro es el PLC DL06



y el puerto 2 debe ser configurado para trabajar con el protocolo MODBUS RTU y debe definirse la velocidad de transmisión, el tratamiento de la falla de comunicación y en general los demás datos de comunicación. Esto se hace con el programa *DirectSOFT*, como se muestra a continuación.

La configuración del puerto 2 se podría hacer por código ladder, para evitar que se pueda perder esta configuración si faltara la energía eléctrica por más de 4 días y no se ha instalado una batería en el PLC.

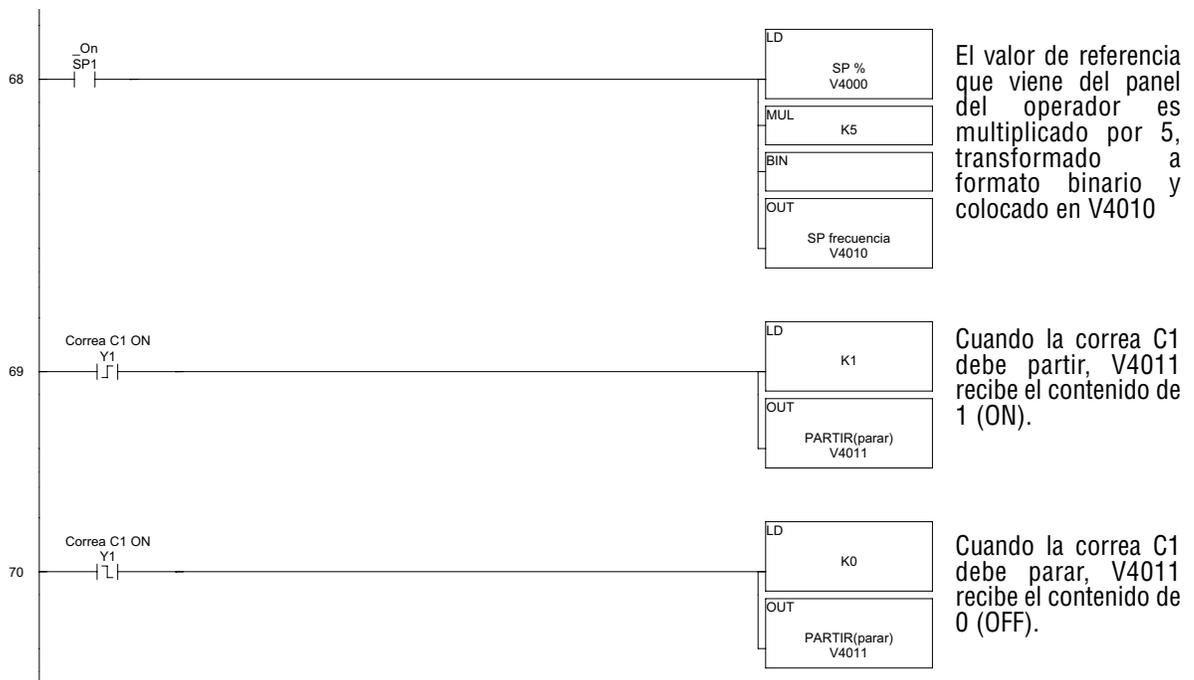
Estando conectado el PLC a la PC con *DirectSOFT*, vaya al menú **PLC**, luego **SETUP** y por último “**Setup sec. comm port...**”. Aparecerá el diálogo mostrado en la figura adyacente.

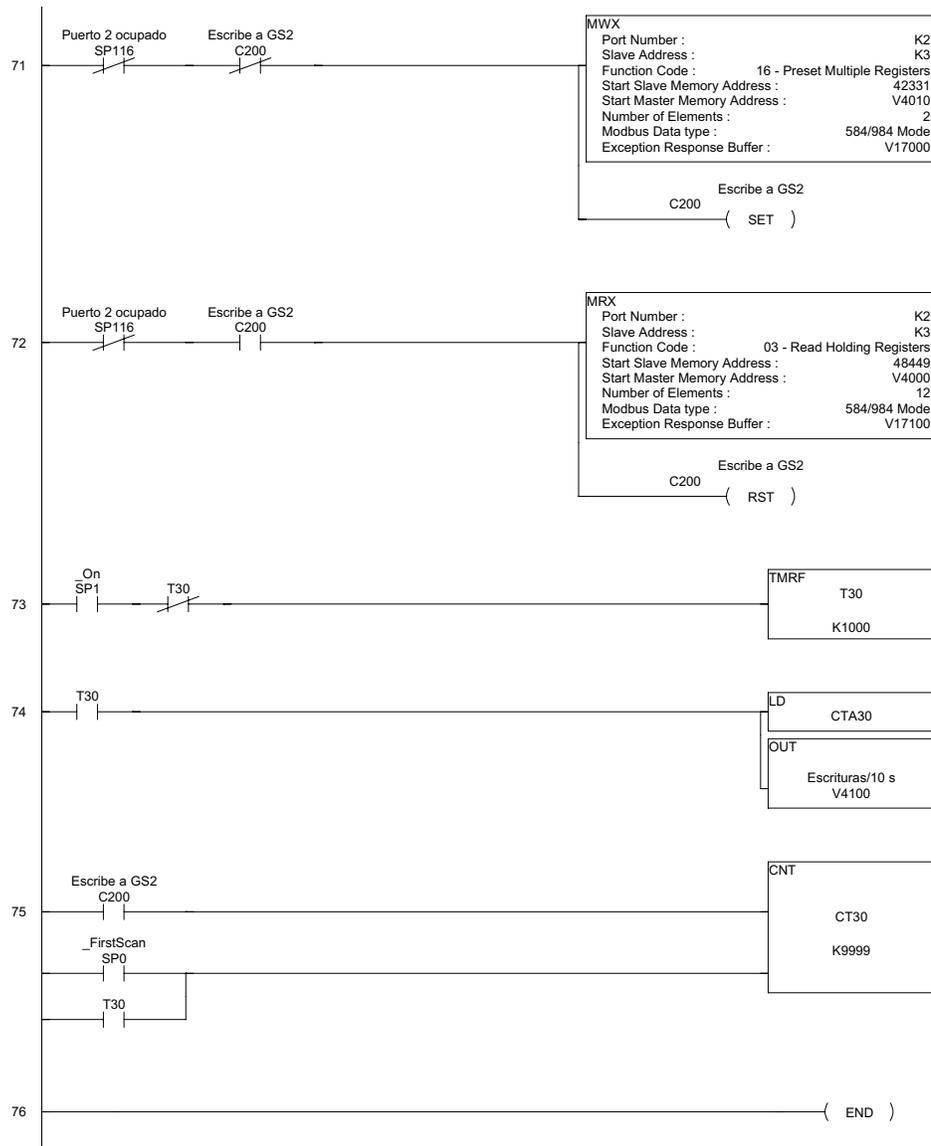


Coloque los datos como mostrado en la figura y asegúrese que los mismos datos está colocados en el variador de frecuencia. Luego es necesario crear el código ladder para permitir que la información sea transmitida. El valor de referencia para velocidad es dado por el panel de pantalla táctil C-more, en % en el registro V4000, como valor BCD de 16 bits. Debemos transformar ésto a frecuencia. También, el mínimo valor es 40%. Eso será ejecutado en el Panel C-more y solamente esperamos un número entre 40 a 100%. Esto corresponde a frecuencias entre 20 a 50Hz. El variador de frecuencia acepta valores decimales (no BCD) y en décimos de Hz. Por lo tanto debemos hacer una operación en el PLC para transformar el valor en % en frecuencia. Eso es fácil multiplicando el valor por 5 ( $100 \times 5 = 500$ , que es la frecuencia máxima). Colocaremos ese valor en V4010.

Recuerde que habíamos controlado el motor con la salida Y1. Esta salida aún puede ser usada para colocar los datos en el puerto 2 para la transmisión. Cuando se activa, el variador de frecuencia debe partir y cuando se desactiva debe parar. Este bit debe colocar un 1 en la memoria designada para tener este dato, cuando se parte, que por conveniencia lo haremos en V4011. Cuando debe parar, el valor en V4011 debe ser 0.

Vea en el código a continuación como ésto fue implementado en *DirectSOFT*.





MWX escribe dos registros desde el PLC en formato binario : . . .  
 V4010=> V42331  
 V4011=> V42332  
 El bit C200 se hace ON cuando la instrucción comienza a ser ejecutada. Esto dura mas de un barrido

La instrucción MRX lee desde los registros de GS2 hacia el PLC  
 48449=> V4010  
 48450=> V4011  
 por 12 registros consecutivos. Vea abajo

Este temporizador crea un pulso cada 10 segundos

Cada 10 segundos se transfiere el conteo del contador CT30 a V4100

El contador CT30 cuenta cuantas veces cada 10 segundos se cierra C200. Esto permite saber cuantas veces por segundo se escribe al aparato GS2 y determinar el periodo de la transacción

Este simple programa permite escribir datos al esclavo 142 veces cada 10 segundos, o 14,2 veces por segundo, cada 70 milisegundos. Esta es la tasa de transferencia con una velocidad de transmisión de 19,2 kbps, para este ejemplo específico.

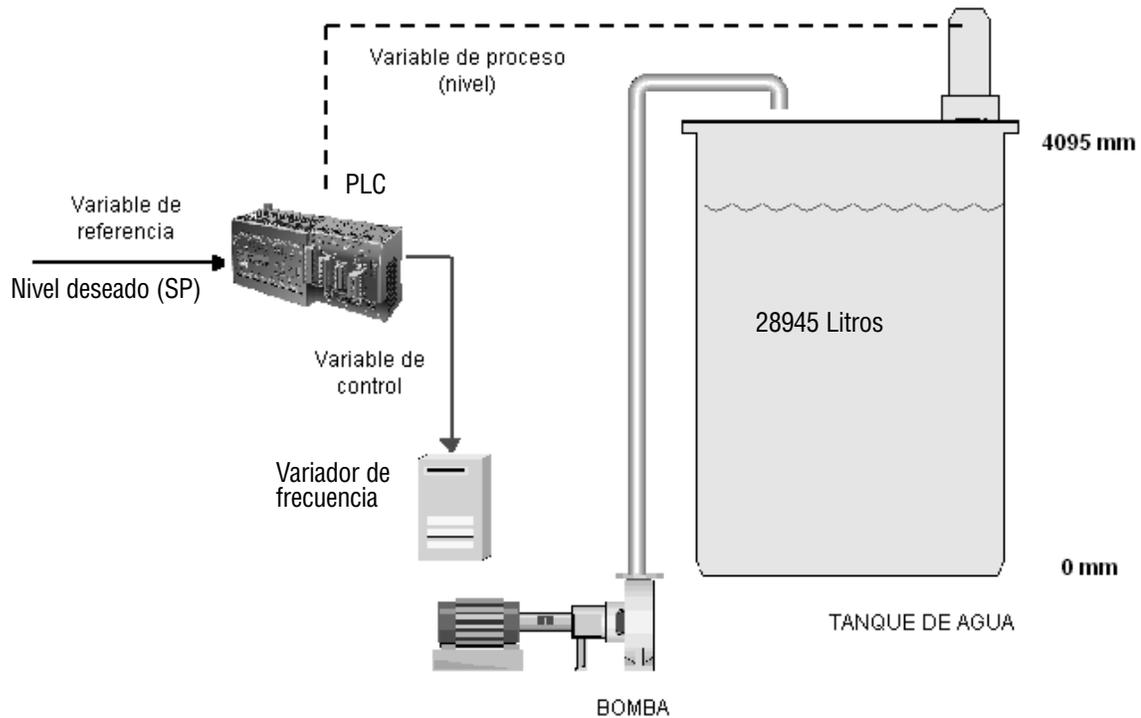
Por otra parte, el PLC tiene un barrido promedio de 4 ms con 76 renglones, como hecho arriba. Aquí se puede ver que las transacciones duran más de un barrido del PLC.

Las lecturas colocan los datos del estado del variador de frecuencia GS2, así como también la frecuencia de referencia y de salida, la corriente de salida, el voltaje de la barra de corriente continua, el voltaje de salida, el valor estimado de rpm del motor, un valor a escala de la frecuencia, el ángulo del factor de potencia y el porcentaje de carga estimado.

Estos valores se pueden mostrar en el panel del operador o hacer uso de los valores para otras funciones.

### Ejemplo 7: Control de un nivel de un tanque de agua usando simulación con números reales.

Este ejemplo considera un lazo de control PID para controlar el nivel de un tanque de agua cilíndrico, que tiene un consumo variable, y una alimentación con una bomba centrífuga activada por un variador de frecuencia. El nivel se mide con un transductor de salida 0-10 Volt.



En este ejemplo se ha simulado el nivel del lazo de control usando código ladder.

La variable de proceso es el nivel del tanque, definido como 0-4095 unidades en V7003 (mm). La referencia de nivel es dado en V7002, en mm, las mismas unidades que la variable de proceso. La salida del PID es expresada en 0-4095 unidades (pero 1750 es 100% de velocidad).

El consumo o perturbación (cantidad de agua que sale del tanque) está definida en V6010, en el rango 120-620 litros por minuto; es decir, siempre hay un consumo variable de por lo menos 120 L/m. El tanque tiene un diámetro de 3 metros y contiene hasta 28945 litros. Por lo tanto, la altura máxima es de aproximadamente 4.09 metros (o 4095 milímetros).

La bomba puede entregar hasta 1200 litros por minuto, de modo que se llena en 26.8 minutos si se comienza desde que el tanque esté vacío y el consumo sea 120 litros por minuto. La bomba gira a un máximo de 1750 rpm.

La relación adaptada entre el caudal de la bomba y la velocidad del motor es:

$$Q [\text{litros/minuto}] = 1200 * n^2 / 1750^2 \text{ siendo } n \text{ la velocidad de la bomba en rpm.}$$

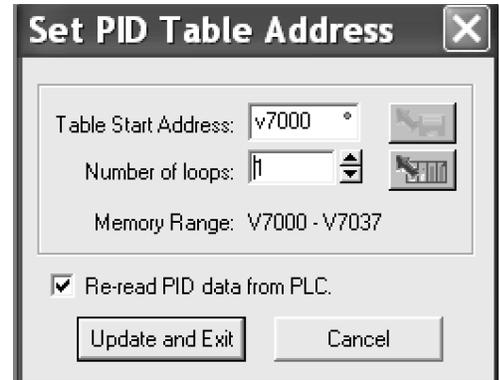
La tabla de PID está en V7000. La tabla de ramp/soak está en el rango V5000 hasta V5037

The PLC tiene un módulo de entradas y salidas como definido anteriormente, con señales de 0- 10 Volt. El transductor de presión entrega una señal de 0 -10 Volt, que corresponde a una señal interna en V2000, de 0- 4095. La salida también será de 0-10 Volt, y para eso es necesario

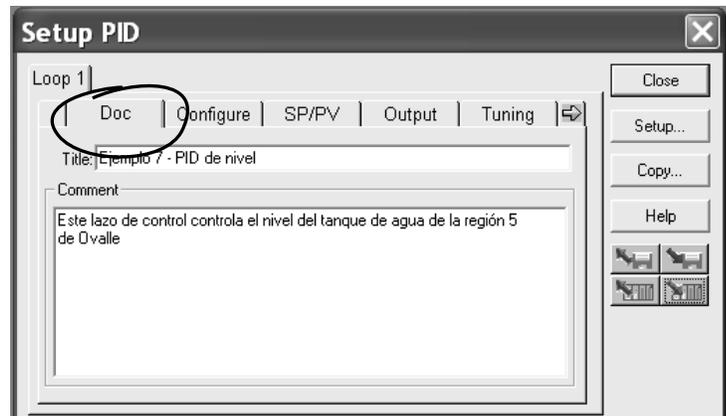
colocar una señal del control de la velocidad del motor escribiendo en la memoria V2010. Antes de ejecutar el programa, es necesario configurar el lazo PID, de la siguiente forma:

### Abra el diálogo de setup PID

Conéctese al PLC. Haga clic en el menú PLC, luego en **Setup** y por último en **PID**. Aparece el cuadro de diálogo **PID Table address**, como se muestra en la figura adyacente. Este diálogo le permite definir en que registro comenzará el grupo de parametros de los lazos PID. Cuando haya definido los dos valores que necesita, haga clic en el botón **Update and Exit**, lo que salva la localización de los parámetros y la cantidad de lazos. En este caso elegiremos V7000 y 1 lazo de control.



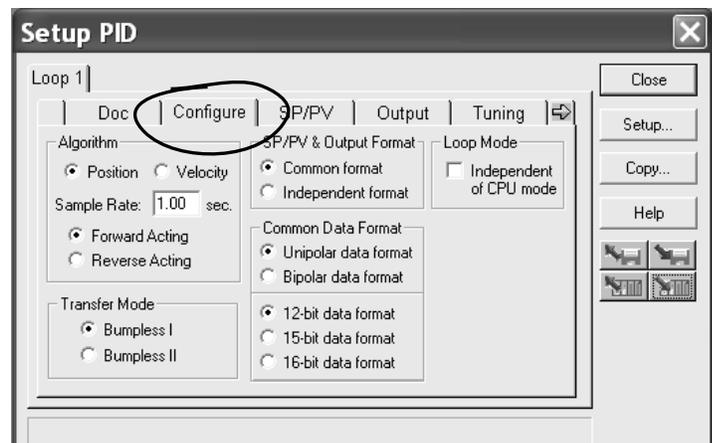
Cuando hace clic en el botón indicado, aparecerá el cuadro de diálogo como se muestra en la figura adyacente. Observe que hay “lengüetas” que seleccionan diversos campos. Por defecto viene en **Doc**. Coloque el título del lazo en el campo **Title**, y comentarios en el campo **Comments**. Esto es opcional.



Luego hay otra lengüeta llamada **Configure**. Este diálogo le muestra como seleccionar las características del algoritmo PID a ser usado.

En este caso, sabemos que debe ser colocado como se muestra en la figura adyacente. Vea más explicaciones en el capítulo 8.

En este caso, la acción directa (**Forward acting**) es necesaria. Si el error (SP-PV) aumenta, la variable de proceso PV aumentará. Usaremos un tiempo de muestreo (**sampling time**) de 1 segundo, ya que no es necesario hacer el lazo más rápido. Usamos también **common format**, ya que la entrada y salida tienen el mismo formato (12 bits, BCD).



Luego debe seleccionar la lengüeta **SP/PV**. Note que las direcciones de estos valores han sido ya seleccionadas.

Aquí, en ese caso, no es necesario cambiar ningún valor. Note que es posible seleccionar la raíz cuadrada (SQUARE ROOT) de la variable de proceso. Esto se hace en el caso de, por ejemplo, placas orificio, que miden presión para determinar flujo.

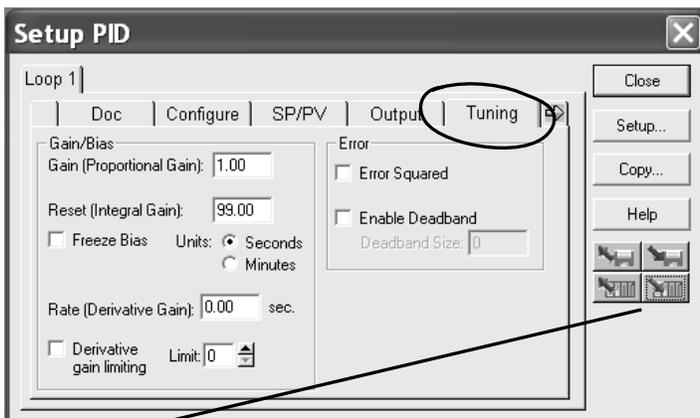
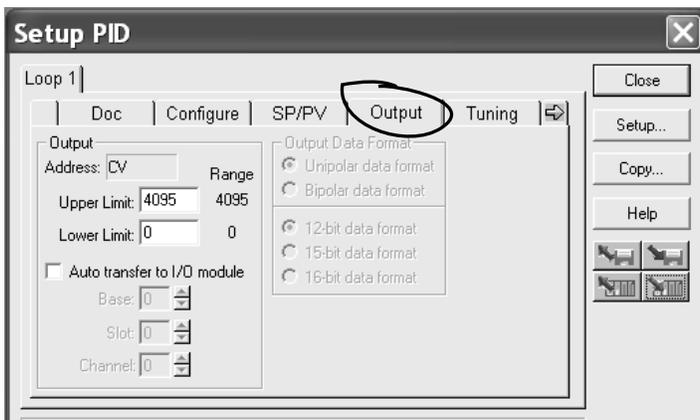
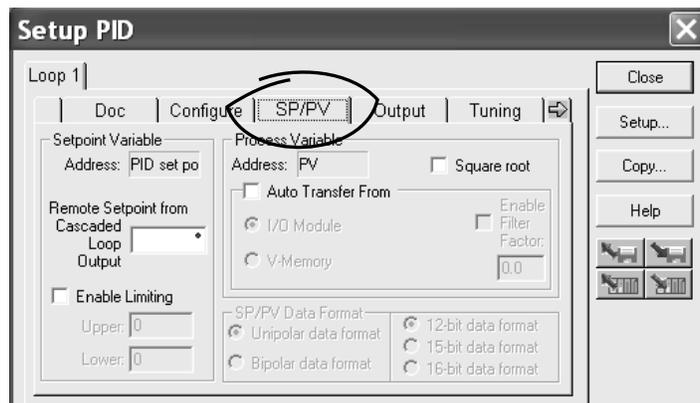
La próxima lengüeta es la lengüeta **Output** (Salida de control) que configura como será el valor numérico de la salida del controlador PID. En el campo **Upper limit** seleccionaremos el valor 4095, en este caso, para limitar la salida al valor máximo que acepta el módulo de salida.

Puede ser que se quiera que la velocidad mínima de la bomba sea de 20%; en ese caso seleccionaríamos un límite inferior. Pero no es necesario en este caso.

No hay otro ajuste que hacer en este caso. Note que siempre el valor límite máximo es 0, por defecto, y si no se coloca un valor diferente, la salida del lazo PID no trabajará.

La próxima lengüeta es la lengüeta **Tuning** (sintonización del lazo) que configura los valores de ganancia proporcional (Gain) y el valor integral (Reset), que son determinados posteriormente, cuando ya esté conectado el sistema de bombeo al PLC. Por ahora, colocaremos el valor 1 en la ganancia y 99 segundos en el reset. No usaremos el factor derivativo. Se puede considerar el uso de freeze bias, pero no lo usaremos en este caso. El lector puede probar el efecto de este acción. Como se hará una simulación, mostraremos esto como si fuera una salida física. Salve los resultados haciendo clic en el botón indicado aquí.

La próxima lengüeta es la lengüeta **Alarms** (alarmas del lazo) que configura los valores límites alarma para este lazo específico, Cada alarma genera un bit activado en uno de los registros de



parámetros del lazo PID. Vea mas detalles en el capítulo 8. No cubriremos ésto en este ejemplo. Por último viene la lengüeta **Ramp/Soak**, que es la programación de hasta 8 segmentos de valores de referencia. Analizaremos este asunto más adelante. Usaremos esto solamente para mostrar como funciona el ramp/soak.

A continuación mostraremos como se hace el programa ladder para interactuar con el lazo PID y como se hace la simulación; V2000 lee los rpm del variador de frecuencia. V2010 es el valor que genera la referencia de frecuencia en el variador de frecuencia. Usaremos un panel de interface de operador EA7-T10C, para entregar el nivel deseado en V7002 (SP), que es la variable en el lazo PID que acepta el valor de referencia, así como también los comandos de partir y parar la bomba y de como cambiar del modo Manual a Automático o viceversa.

En este ejemplo, usamos números de punto flotante, conocidos como números reales. Recordemos que, en realidad, es una representación de un número real, pero éste tiene algún error de truncado, que es pequeño, pero hace que el cálculo no sea exacto. En este caso, el error no es importante.

Lo que se hace primeramente en el programa es configurar el módulo de señales análogas con la instrucción ANLGCMB en el renglón 1.

El renglón 2 define constantes en el primer barrido del PLC, usando SP0.

El renglón 3 genera un pulso cada segundo, para minimizar el tiempo de barrido promedio. Note que se hace un enclavamiento con el momento en que el algoritmo del lazo PID se ejecuta. Luego se programa en el renglón 4 el consumo de agua en litros por minuto, ya que la entrada está en el rango 0-4095, se multiplica por el factor  $500/4095 = 0.1221$  para obtener litros por minuto, y luego se le agrega 60, para poder considerar que el consumo será al menos 60 litros por minuto.

El renglón 5 lee los rpm desde el variador de frecuencia, como señal 0-10 Volt, que es leído en el rango 0-4095, lo transforma en rpm con el factor  $1750/4095 = 0.42735$ . Luego es elevado al cuadrado para considerar la fórmula de flujo, y se obtiene el flujo en litros por minuto en la memoria V6000, como número de punto flotante.

El renglón 6 calcula el aporte de agua neto, ya que se alimenta el tanque con la bomba, pero al mismo tiempo hay agua saliendo del tanque. El aporte se coloca en V6016, en litros por minuto, pero se transforma en litros por segundo, para permitir el cálculo del volumen de agua en el tanque, a ser hecho posteriormente.

El renglón 7 calcula el volumen del tanque en litros. El aporte se coloca en V6016, en litros y por facilidad de supervisión con Data View, se coloca también en V6032.

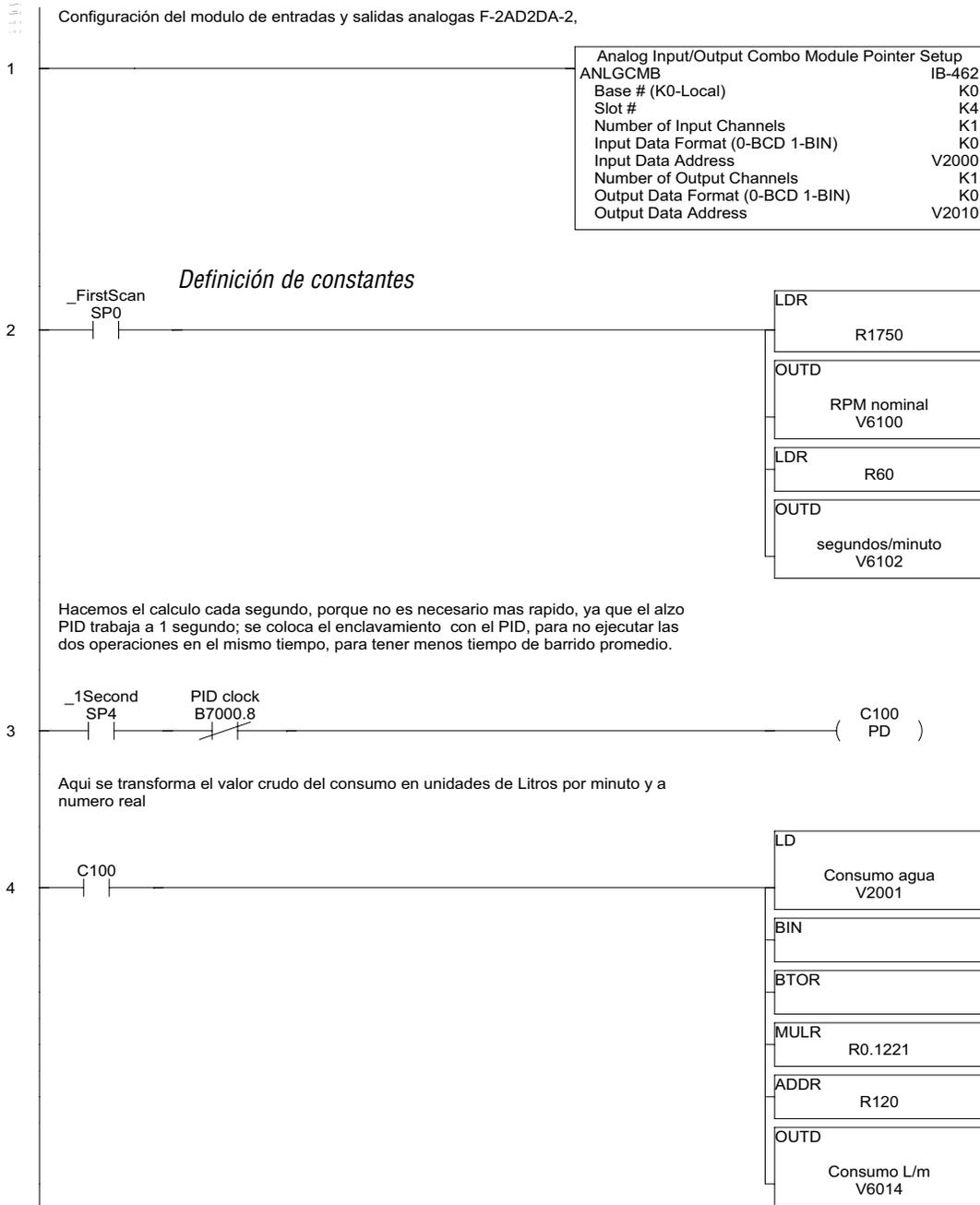
El renglón 8 compara el volumen de agua con la capacidad del tanque.

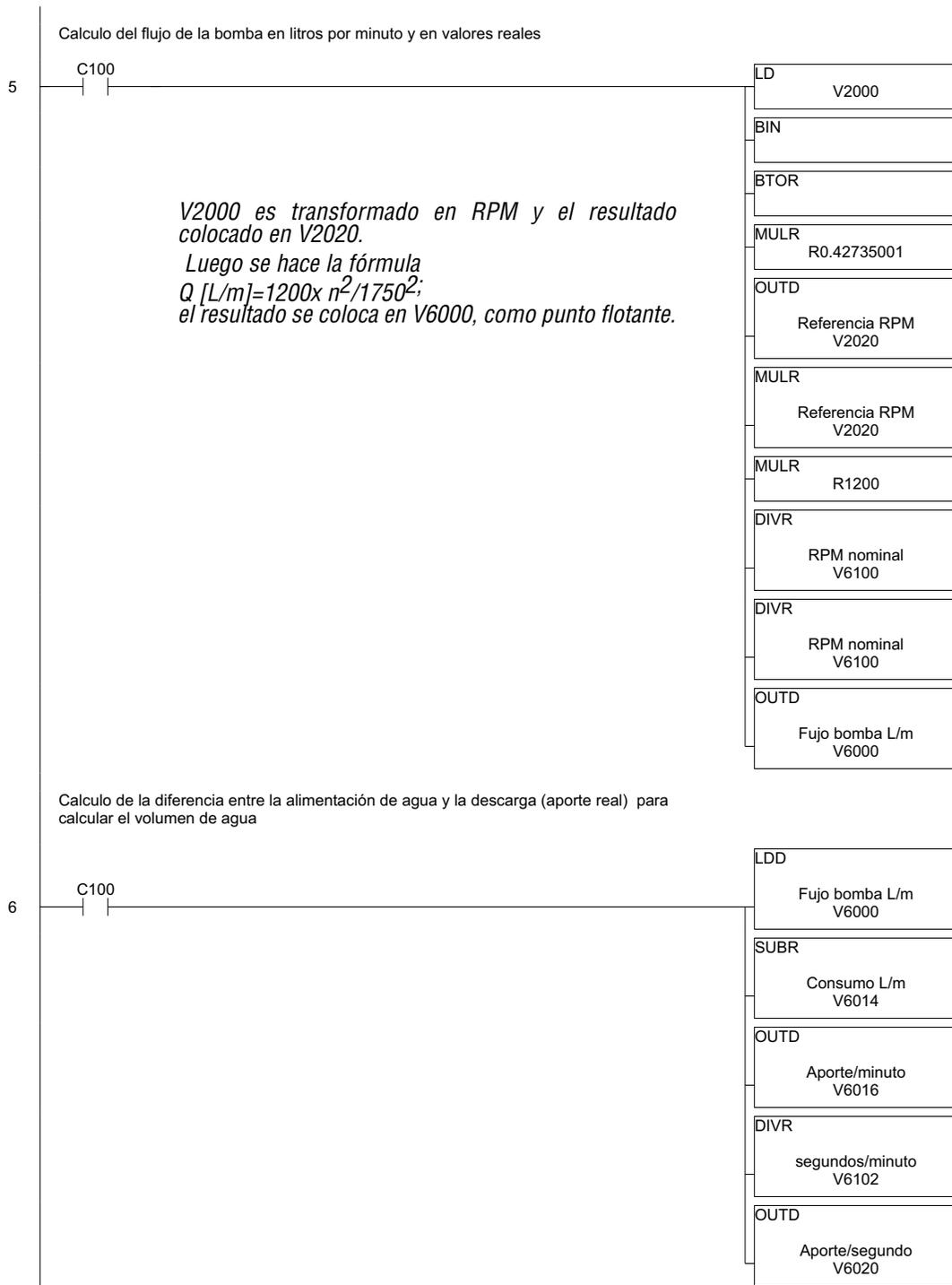
El renglón 9 recibe el resultado de la comparación y se coloca en la memoria el máximo valor que el tanque puede contener, para simular rebalse del tanque.

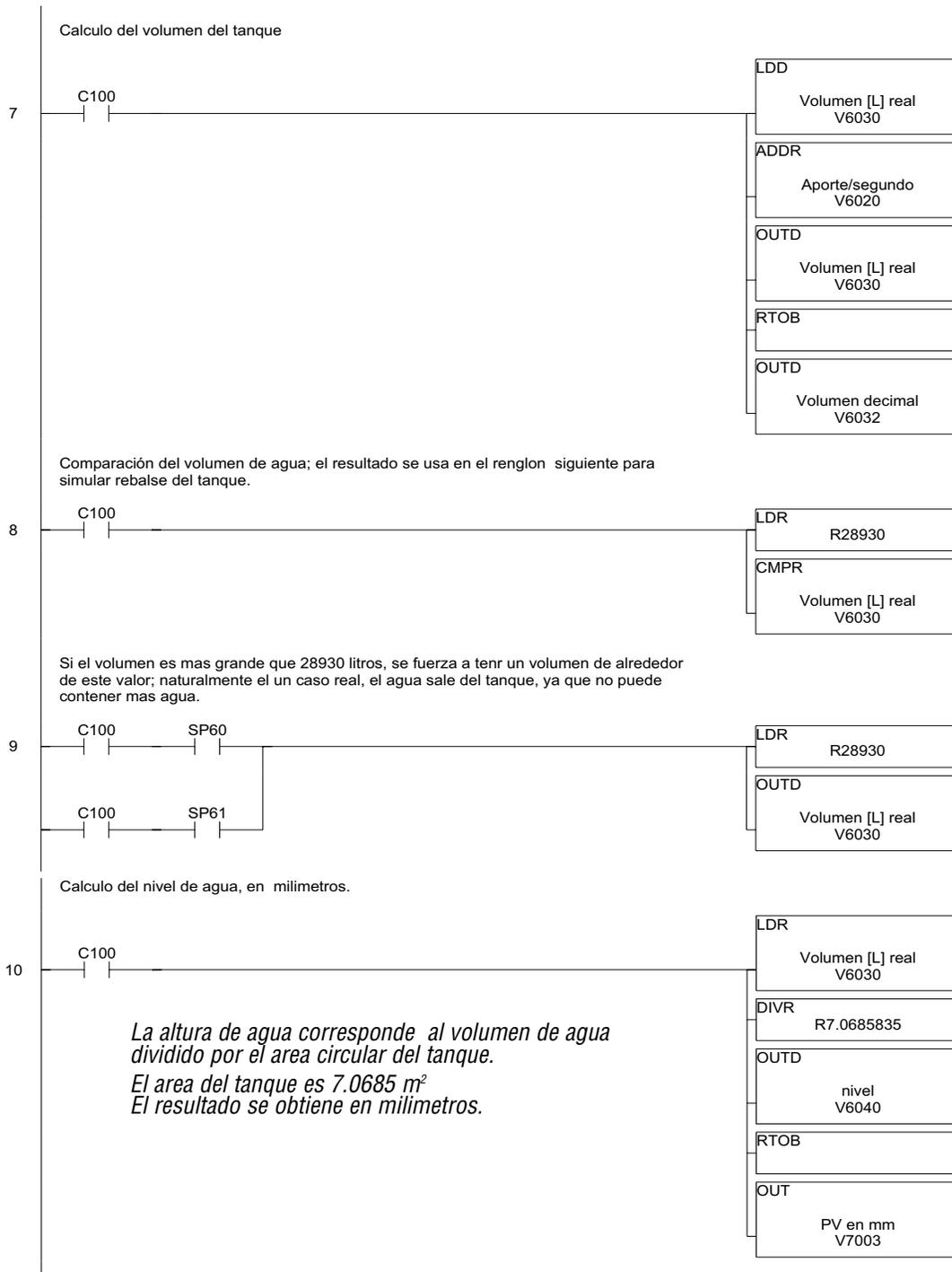
El renglón 10 hace el cálculo del nivel directamente en milímetros y el resultado es colocado en V6040; este valor tiene el formato de número de punto flotante. Este valor se transforma en valor decimal con la instrucción RTOB y se coloca directamente en V7003, la dirección del lazo PID que acepta la variable de proceso.

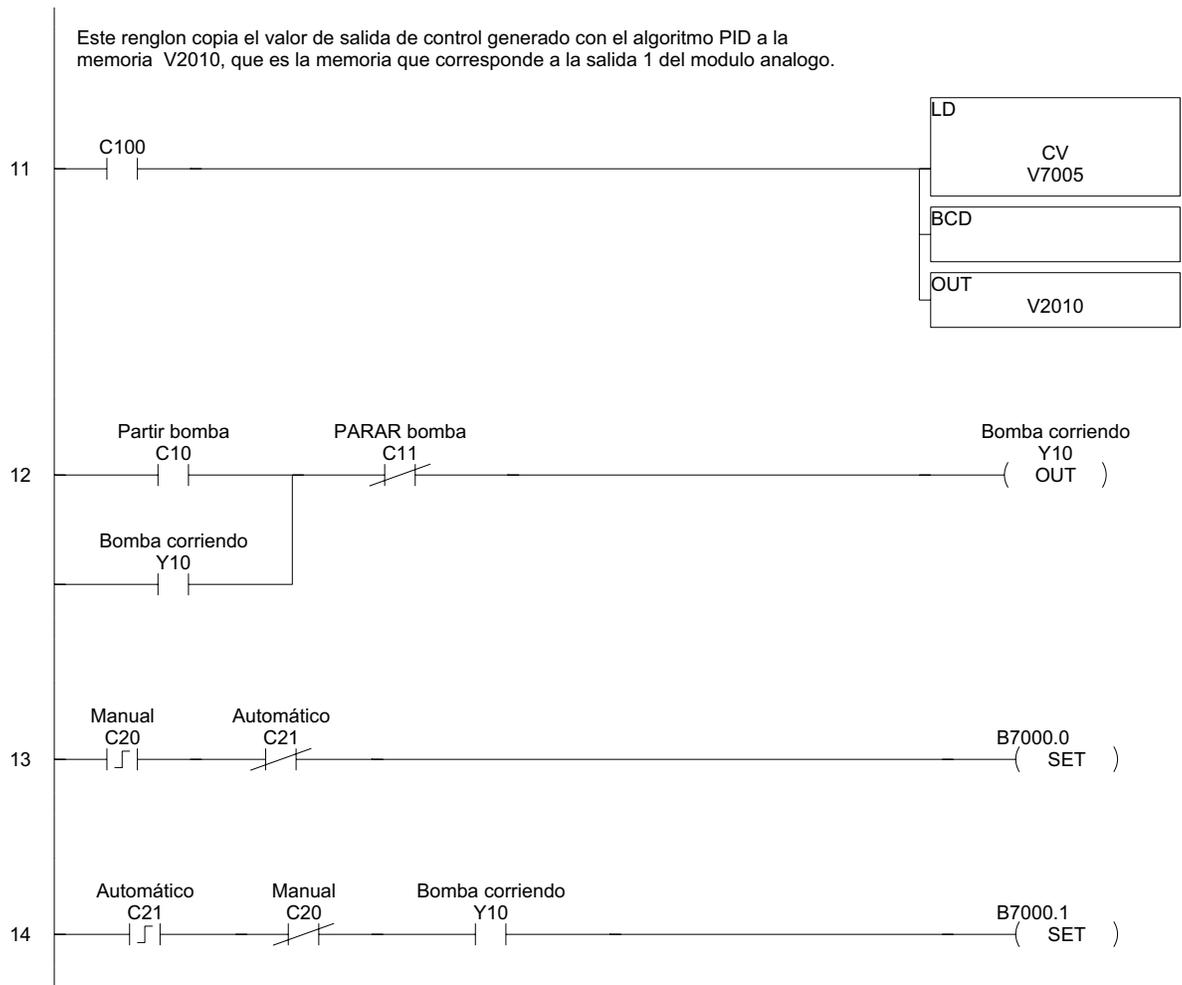
Otro asunto a considerar es el comando de partida y parada de la bomba, y también como comandar cuando el lazo de control va al modo Manual o al Automático. En Manual, el algoritmo que calcula la salida del lazo de control no es ejecutado. En Automático, la salida debe afectar el lazo de acuerdo a lo programado y hay una salida de control que depende del error (SP-PV) y de los valores de ganancia y de reset.

Usaremos botones en el panel de interface del operador para partir y parar la bomba, así como también para pasar de Manual a Automático. Se muestra en el diagrama ladder como se programa esta función. Vea a continuación el diagrama ladder correspondiente:









No se ha colocado la instrucción END porque el programa sigue posteriormente para mostrar el uso de la función ramp/soak.

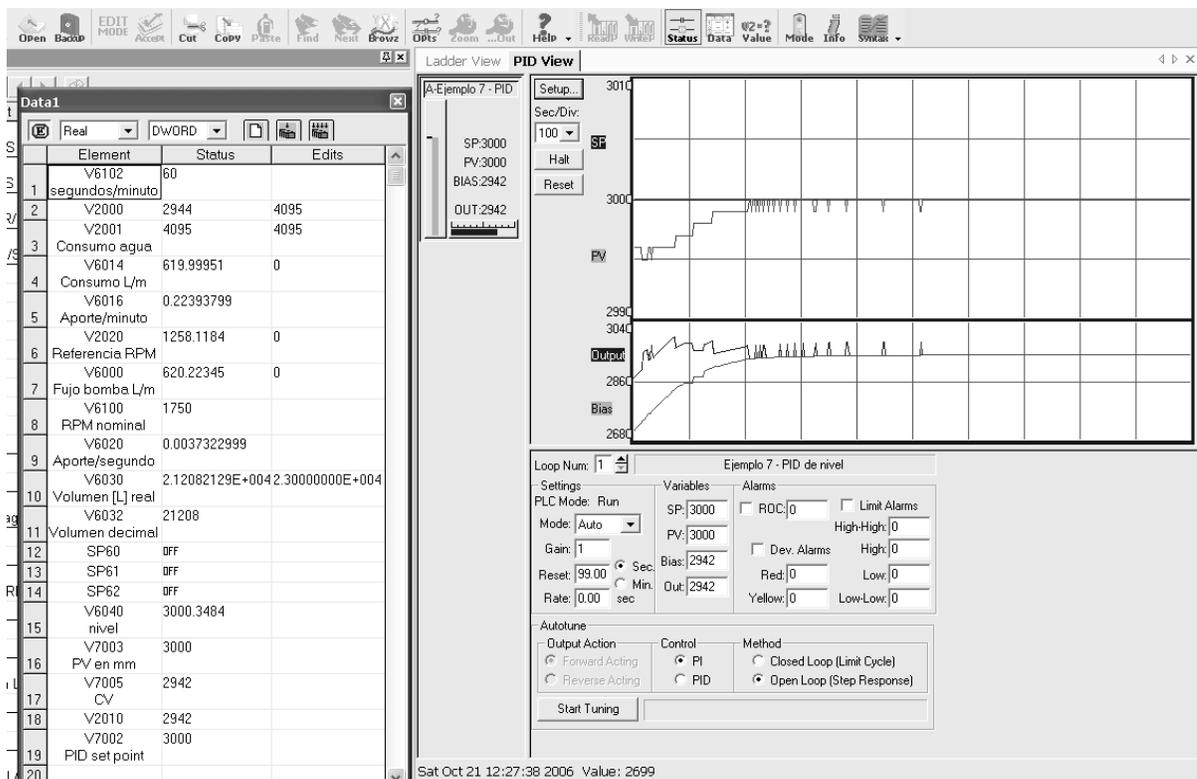
Note que hicimos, de propósito, cálculos sin usar Iboxes, para demostrar como se hace el cálculo con versiones de *DirectSOFT* anteriores a version 5. Naturalmente los cálculos pueden ser hechos con Iboxes. Se puede ver que el tiempo de barrido promedio es de 3 ms, con lo cual conseguimos el objetivo. Ésto no es importante en este caso, pero sirve para mostrar trucos para acortar el tiempo de barrido.

Teniendo el programa hecho, asumiremos que la salida del módulo de señales análogas y la entrada son equivalentes en RPM y por lo tanto conectaremos la salida 0-10 Volt a la entrada 0 -10 Volt. En la práctica, esto se conectará directamente al variador de frecuencia.

## Sintonización (Tuning)

Corresponde hacer la sintonización (tuning); haremos el método de sintonización automática. Para eso colocaremos el valor de la variable de proceso (PV) lo mismo que el valor de referencia (SP), por ejemplo, en 3000 mm, usando los valores de ganancia y de reset que habíamos definido anteriormente. Usamos PID View y DataView para ayudar al proceso.

Colocamos en funcionamiento la bomba y cambiamos el modo desde **Manual** (que es el valor por defecto) a **Automático (Auto)**, con ayuda del panel de interface de operador, si es que está conectado, o con Data View, o aún con la ventana de PID View. Ud puede, en este caso, cambiar el modo directamente en el campo **Mode** haciendo clic en la flecha de menú al lado derecho del campo **Mode**.

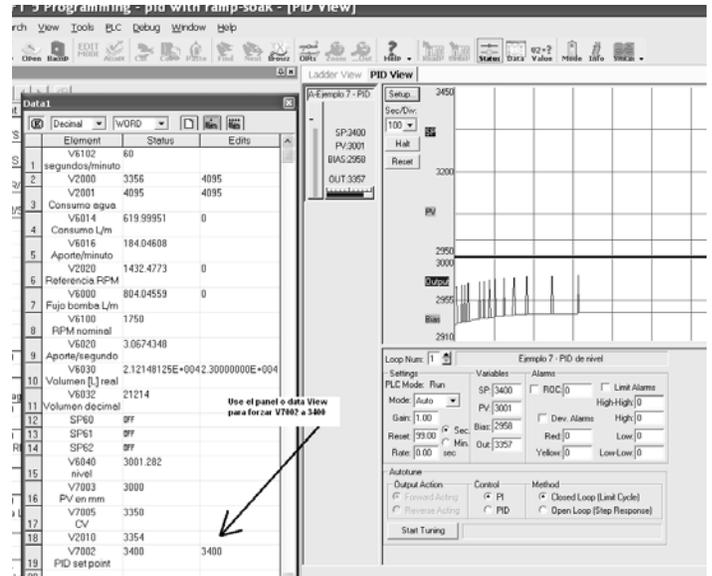


Como se ve en la figura de arriba, Data View puede colocar los valores que interesan para observar el proceso.

Con Data View se pueden forzar otros valores, se puede cambiar el estado de los bits C, etc.

El diálogo PID View permite visualizar la variable de proceso PV, el valor de referencia (SP), la salida de control CV y el bias, convenientemente en varios rangos de tiempo, hasta 200 segundos por división. También es posible cambiar valores y modos y es una herramienta indispensable para sintonización.

La próxima acción es activar el “autotuning” o sintonización automática; para eso usamos el campo **Autotune**; seleccionamos el método closed loop, el modo PI (no es necesario el efecto derivativo en este caso), colocamos una referencia más alta que la variable de proceso, por ejemplo, 3400 y se hace clic en el botón **Start Tuning** para que que el PLC comience a determinar los valores de ganancia y de reset aplicando escalones de referencia. Esto toma cierto tiempo. En este caso tomó unos 5 a 8 minutos debido a que el actuador (la bomba) es relativamente pequeña. El diálogo de PID View avisa cuando al proceso acaba diciendo “Autotune complete, check PID parameters”.

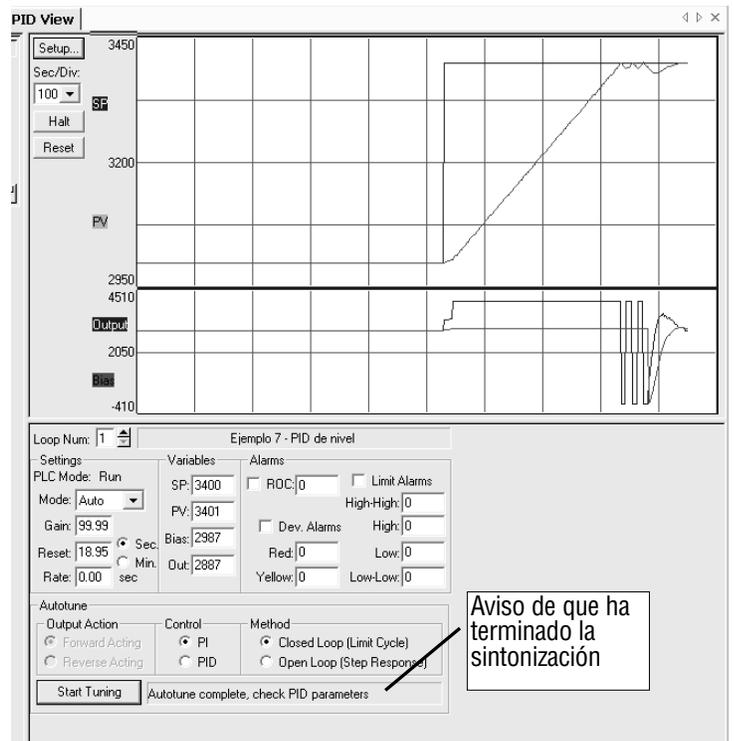


En general, la determinación de los valores de ganancia y reset son adecuados para la operación estable del control. Esta determinación es un valor calculado por el PLC considerando el atraso de la señal y la inclinación de la rampa producida por la variable de proceso al imponer referencias escalón por el PLC, durante este procedimiento.. Este valor no es el óptimo, sino es el que el PLC cree que es el mejor valor. Otras veces es necesario hacer una sintonización “manual”, para mejorar esos valores. El procedimiento para hacer esta operación está explicado en el capítulo 8 de este manual.

Observe las otras figuras que se muestran a continuación.

En esta figura se vé cuando el proceso de determinación de los valores ha terminado y aparece al lado del botón **Start Tuning** el mensaje que ha terminado la sintonización.

La próxima acción es verificar que el lazo de control trabaja en todo el rango. Por eso se han probado otros

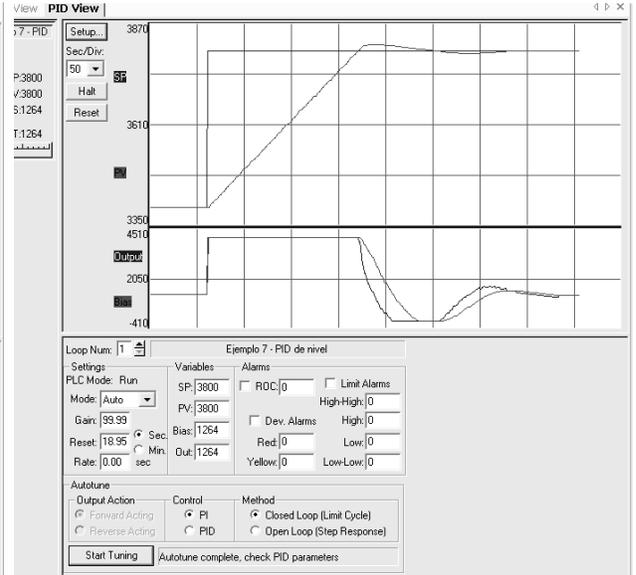
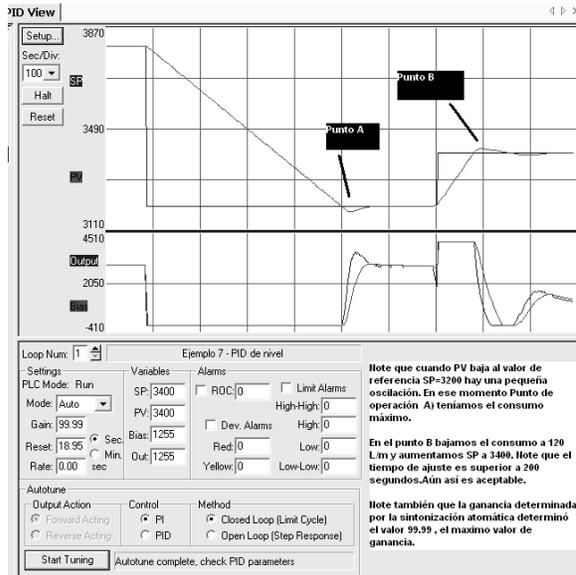
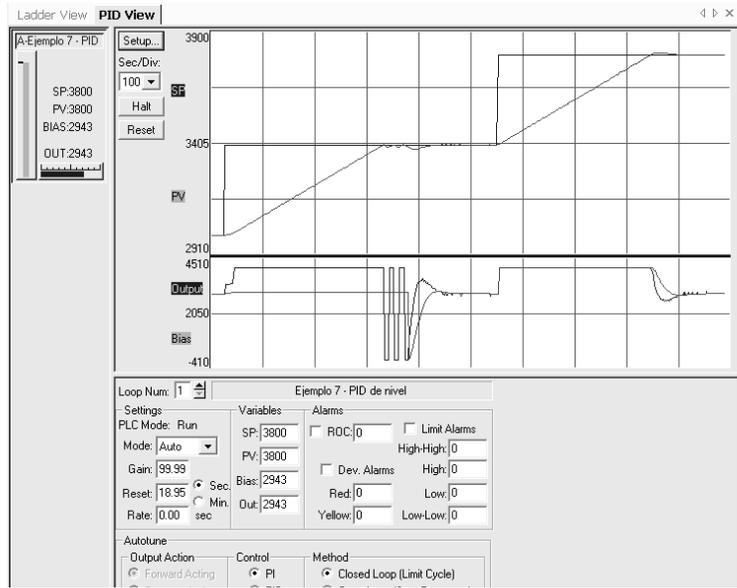


puntos de referencia y se ha cambiado el consumo de agua, del máximo al mínimo.

Se espera que el nivel del tanque se mantenga en aproximadamente 3800 mm.

Por eso se ha probado en el entorno de este valor con valores de 3200, 3400 y 3800 mm.

Las figuras muestran que el comportamiento es razonable. Aún existe “overshoot” y la variable de proceso se estabiliza en unos 200 segundos. En este caso no tiene

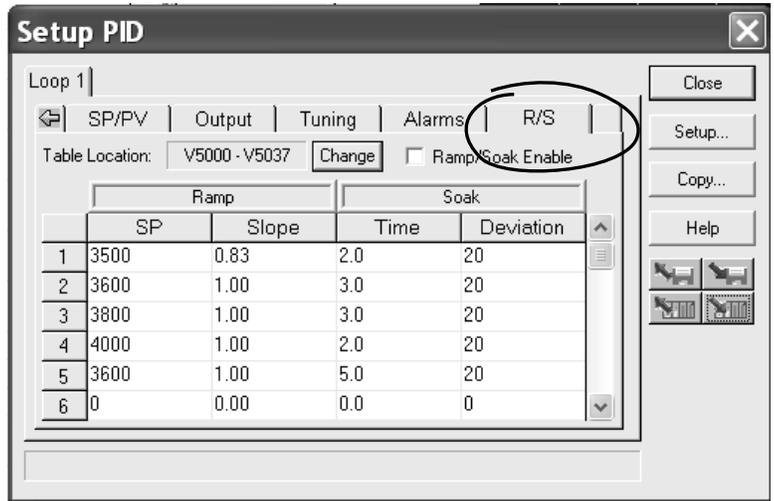


importancia, pero puede haber casos en que sea necesario no tener ningún “overshoot”. También, en otros casos, la respuesta debe ser más rápida y para eso, naturalmente, el actuador debe proporcionar mas corrección (la bomba debe tener mas capacidad de bombeo, es este caso).

El muestreo del control PID puede ser ajustado hasta 50 milisegundos, (lo que permite hacer que el PLC DL06 calcule la salida hasta 20 veces por segundo).

### Uso de ramp/soak

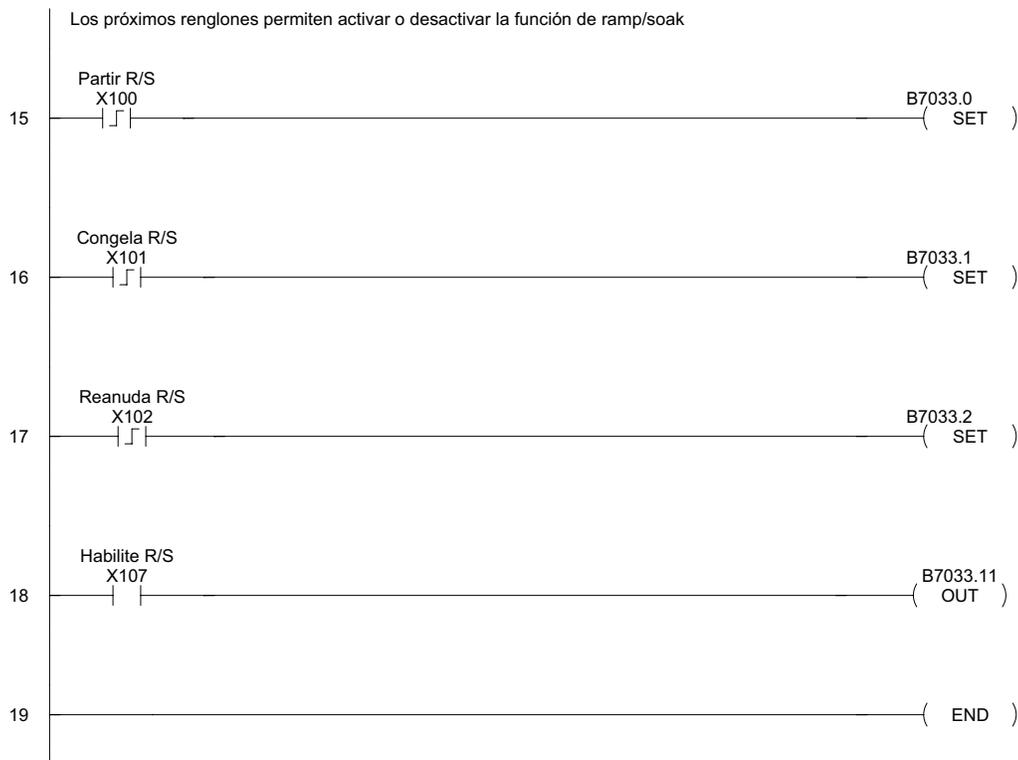
Veamos ahora un ejemplo de uso de ramp/soak. recordemos que esta función es la programación de la referencia del lazo PID variable durante el tiempo. Para programar estos tiempos, se usa el diálogo de **Setup PID**, en la lengüeta de R/S. Vea la figura adyacente para las próximas explicaciones:



Consideraremos en la primera etapa que la primera rampa es de 50 segundos (0.83 minutos) yendo de donde se encuentre la variable de proceso hasta 3600 mm. Luego tendremos un valor

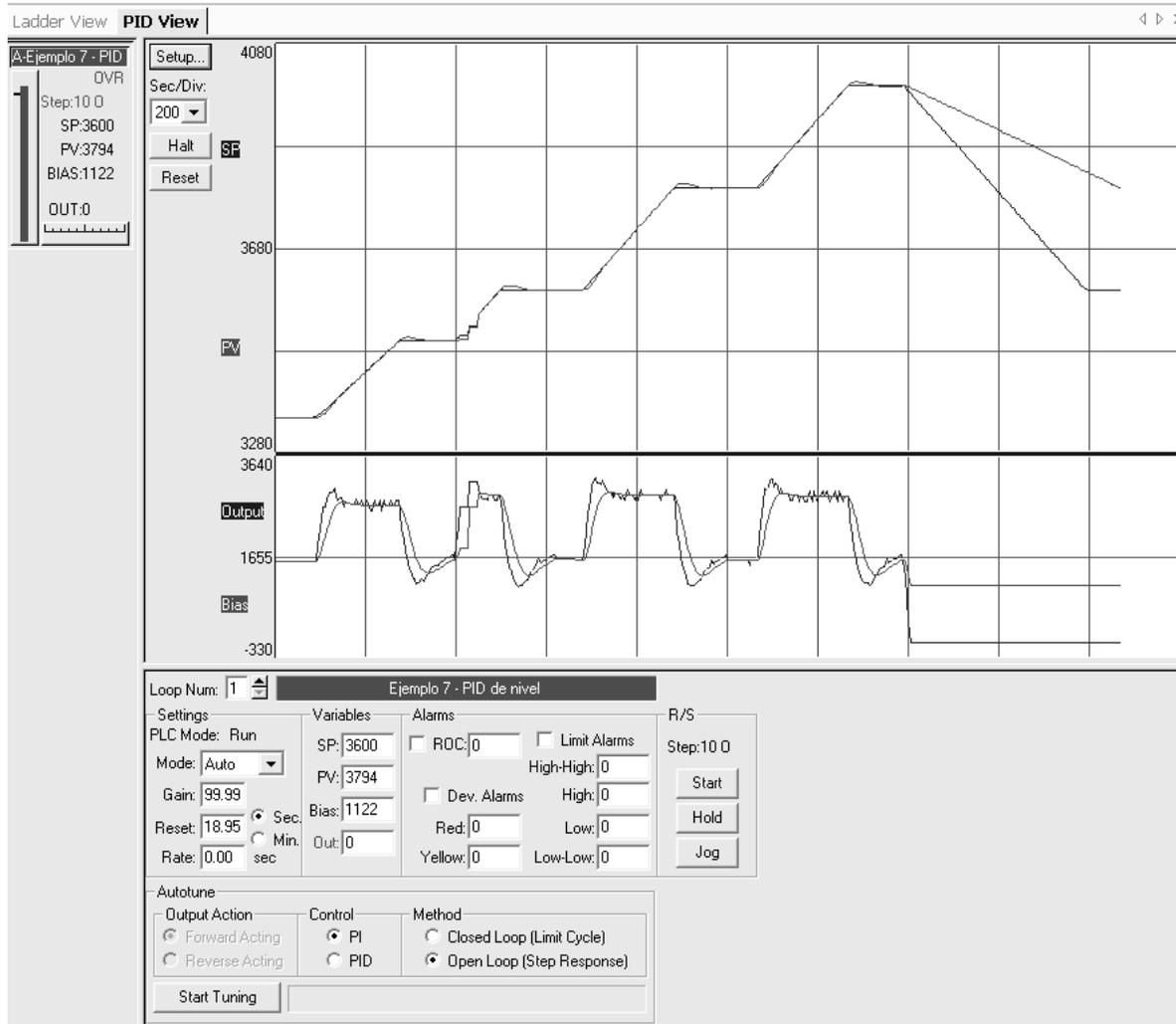
fijo de 2 minutos (soak), Luego viene la segunda etapa, con 1 minuto de rampa y 3 minutos de soak en el valor 3800 . Y así sucesivamente.

Es necesario hacer código ladder para poder partir y parar esta función. Eso se hace de la forma mostrada en el diagrama siguiente:



Note que hemos usado las entradas X100 hasta X107. La razón de eso es que hemos usado el módulo D0-08SIM, que permite simular entradas. Naturalmente pueden ser usadas las entradas X0 hasta X23, u otras del tipo C0, que hayan sido consideradas con el panel de interface de operador.

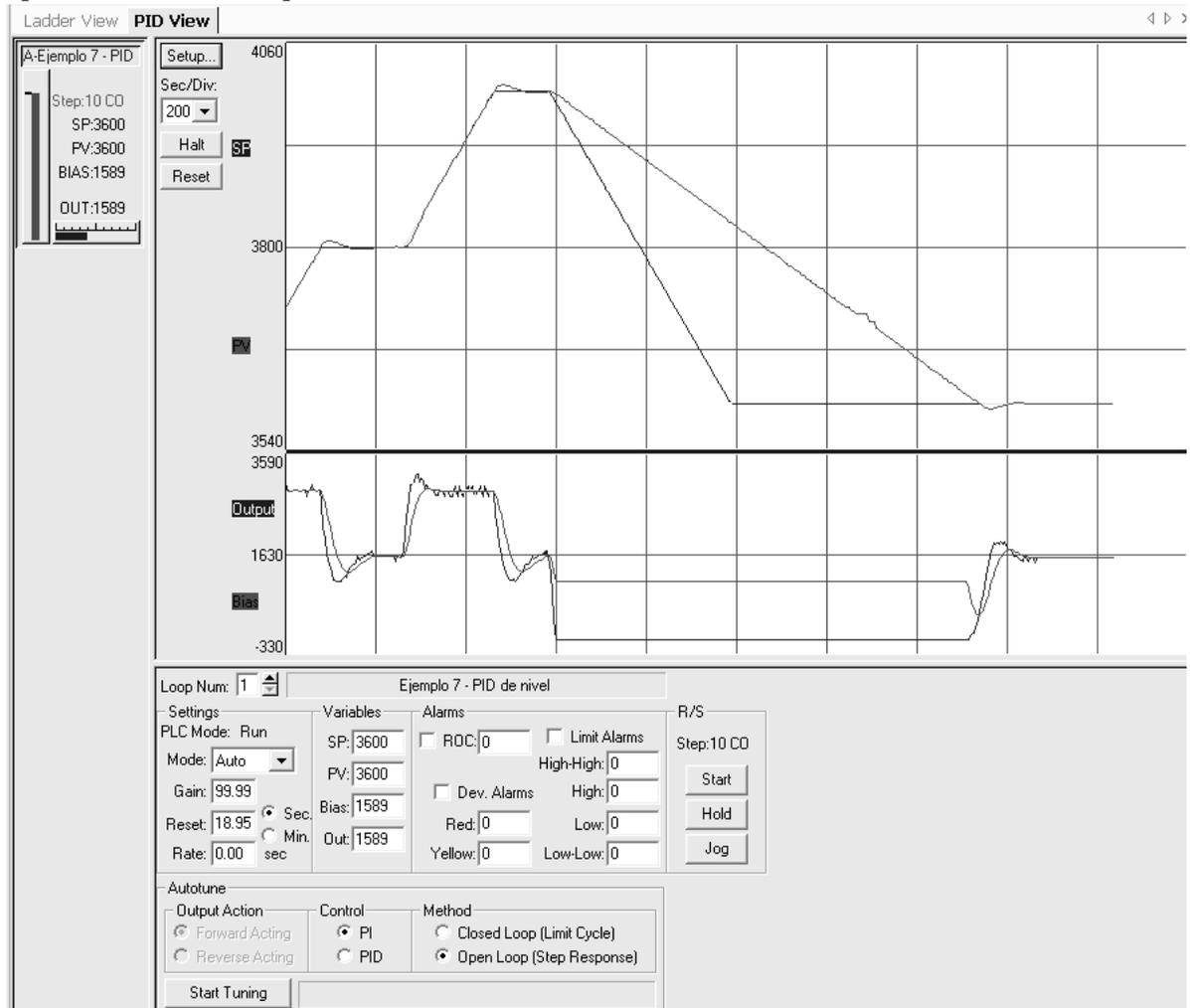
Vea en la próxima figura el resultado de esta función. Más explicaciones pueden ser encontradas en el capítulo 8.



### Observaciones de la figura de arriba, con ramp/soak:

Note que la referencia sigue la programación de la variable de proceso. Inicialmente el valor de referencia estaba en 3350 mm. Cuando se activó la función ramp/soak la salida cambió para activar la bomba para dar mas flujo para que la variable de proceso siga la referencia. Eso es posible sin problemas en la subida. En la disminución del valor de referencia, el consumo, aunque sea el máximo, no permite que la variable de proceso siga la referencia, porque, aunque la salida se vaya a cero (bomba parada) el volumen del tanque no disminuye tan rápido como es deseado. Esta es una consideración importante en los lazos de control PID.

Note también, en la próxima figura, que cuando la variable de proceso llega al valor de referencia, la bomba comienza a girar a la velocidad tal que compensa el consumo, es decir, 1589 rpm equivalentes ( $1859/4095 \times 1750$ ). Esta es una excelente solución, económica, para mantener el nivel del agua en el tanque a la altura deseada en función del tiempo. Este proceso de control PID también es una mucha mejor solución que hacer que la bomba parta y pare en función de 2 presostatos en el tanque.



Esto termina los ejemplos que han sido desarrollados en este capítulo. Esperamos que sean de utilidad para aprender a programar el PLC DL06. Depende de la inteligencia del programador hacer el programa de modo que haga las funciones deseadas, que se pueda entender, dependiendo de cuán bien sea documentado el diagrama. Existen muchas posibilidades de programación y un programa es válido cuando hace las funciones que se necesitan, con el mínimo de codificación, con un tiempo de barrido aceptable. Es muy conveniente que se pueda entender por otras personas. Por eso debe colocarse información en el diagrama. Note, por otra parte, que la documentación no se almacena en el PLC; solamente es almacenado el diagrama ladder en el PLC. Toda la documentación se almacena en la memoria del disco de la computadora.

Página dejada en blanco intencionalmente