

# PLC

## Conceptos Básicos de GX Works2

Este curso de capacitación (aprendizaje en línea) está diseñado para quienes utilicen el software GX Works2 por primera vez para crear programas de secuencia.

## Introducción **Objetivo del curso**

Este curso brinda un conocimiento básico sobre el uso del software GX Works2 para programación, depuración y verificación de la operación de un controlador programable (PLC). Este curso está destinado para quienes creen programas de secuencia para los controladores de las series MELSEC-Q, MELSEC-L y MELSEC-F.

## Introducción Estructura del curso

El contenido de este curso es el siguiente.  
Le recomendamos comenzar desde el Capítulo 1.

### Capítulo 1 - Método de control del sistema de PLC

El software y el lenguaje usados para la programación se presentan aquí.

### Capítulo 2 - Diseño de programas

Aprenderá a diseñar un programa en base a los elementos de control y configuración del hardware.

### Capítulo 3 - Programación

Aprenderá a programar usando el software dedicado GX Works2

### Capítulo 4 - Depuración

Aprenderá a escribir programas de secuencia al módulo de CPU y a depurarlos.

### Capítulo 5 - Prueba final

Calificación para aprobar: 60% o superior.

## Introducción **Cómo usar esta herramienta de aprendizaje en línea**



Ir a la página siguiente		Ir a la página siguiente.
Regresar a la página anterior		Regresar a la página anterior.
Ir a la página deseada		Se visualizará el "Índice", lo que le permitirá navegar a la página deseada.
Salir del aprendizaje		Salir del aprendizaje. El aprendizaje y las ventanas como la pantalla de "Contenidos" se cerrarán.

## Introducción Precauciones del uso



### Precauciones de seguridad

Cuando aprenda usando productos reales, lea con cuidado las precauciones de seguridad ubicadas en los manuales correspondientes.

### Precauciones en este curso

- Es posible que las pantallas visualizadas de la versión del software que use sean diferentes a las de este curso.

# Capítulo 1 Método de control del sistema de PLC

Este curso está destinado para personas que trabajen con un software de ingeniería. Cubre algunos de los conceptos básicos de manejar los sistemas de la series MELSEC-Q, L y F.

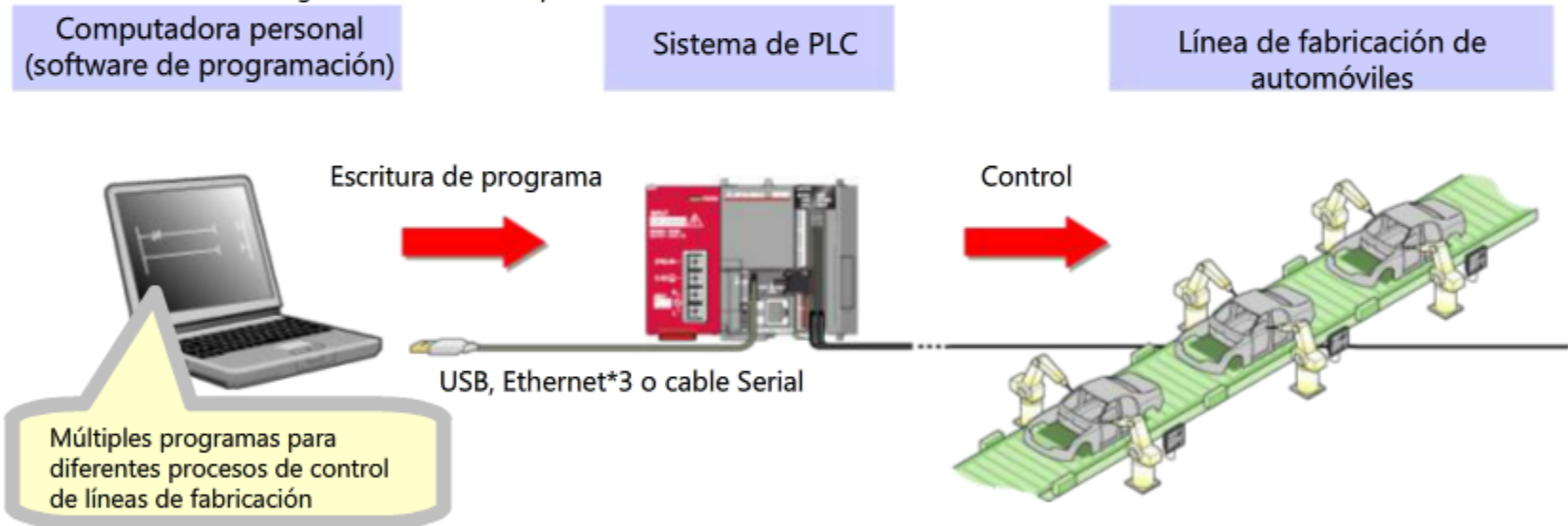
GX Works 2 (GXW2) utiliza lenguajes de programación estandarizados a nivel internacional, incluyendo el lenguaje de Diagrama de función secuencial (SFC, Sequential Function Chart), Lista de Instrucciones (IL, Instruction List)\*1, Lógica en Escalera (Logic Ladder), Diagrama de Bloque de Función (FBD, Function Block Diagram)\*2 y Texto Estructurado (ST, Structured Text).

Los programas se desarrollan usando una computadora personal con un "software de ingeniería" en ejecución, GX works2, y generalmente se escribe al CPU del controlador programable mediante un USB, cable Ethernet\*3 o cable Serial. El módulo de CPU puede reprogramarse cuantas veces sea necesario para adaptarse a cualquier cambio requerido en el control deseado.

\*1 Plan futuro para GX works2.

\*2 Actualmente llamado Escalera Estructurada (Structured Ladder) en GX works2, cumplimiento con IEC previsto.

\*3 Ethernet es una marca registrada de Xerox Corp.



En este curso, la lógica en escalera (uno de los lenguajes de programación de PLC más populares) se utiliza en el programa de ejemplo. Si bien los ejemplos utilizan un PLC de la Serie L, los contenidos de este curso se aplican equitativamente a los Sistemas de la Serie Q.

El método de control básico también es el mismo para la serie MELSEC-F, pero algunas de las operaciones y funciones son diferentes.

Este curso de aprendizaje en línea cubre los pasos de diseño de software (mostrados en verde) necesarios para la implementación de un sistema de controlador programable.

### Diseño de hardware

(1) Diseño del sistema ..... Curso de MELSEC-Q/MELSEC-L Básico

(2) Selección del producto ..... Curso de MELSEC-Q/MELSEC-L Básico

(3) Preparación de avance ..... Curso de MELSEC-Q/MELSEC-L Básico

(4) Instalación y cableado ..... Curso de MELSEC-Q/MELSEC-L Básico

(5) Verificación del cableado .. Curso de MELSEC-Q/MELSEC-L Básico

### Diseño de software

(6) Diseño de programas ..... Capítulo 2

(7) Programación ..... Capítulo 3

(8) Depuración ..... Capítulo 4

(9) Operación

**Alcance de este  
curso**

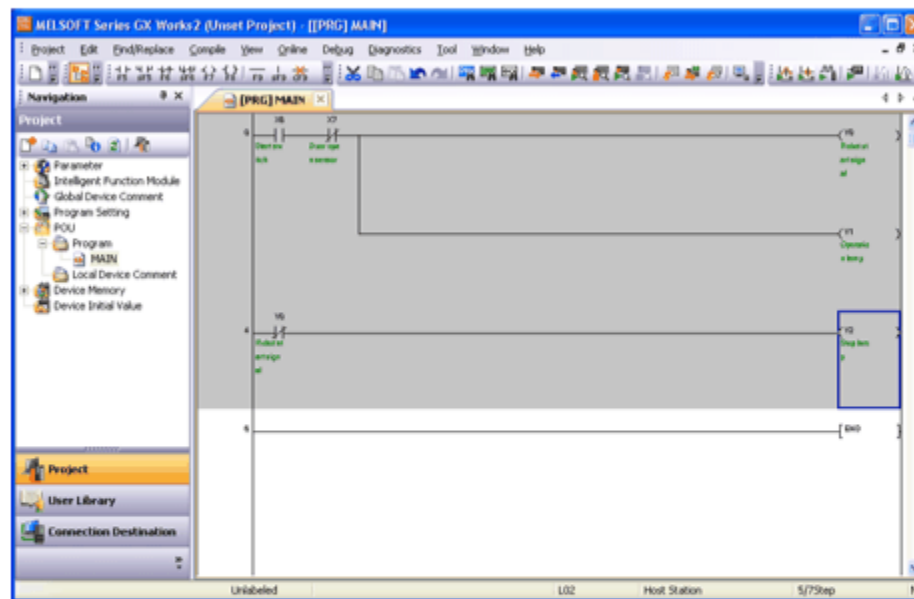
## 1.2

## Requisitos para la programación

En este curso, nos enfocamos en cómo usar el software de ingeniería de controlador programable GX Works2 para desarrollar el programa del sistema de ejemplo.

Unas cuantas funciones principales del GX Works2 se mencionan a continuación.

- Memoria y gestión de archivos
- Desarrollo de programas de controlador programable
- Gestión de la documentación del programa (comentarios, etc.)
- Lectura de los datos de escritura (especialmente los programas) desde/hasta el módulo de CPU
- Verificación de la operación del programa
  - Estimulación del software del hardware del PLC
  - Forzar apagado o encendido de E/S
  - Monitoreo del E/S y estado de dirección de memoria
- Realización de las tareas de mantenimiento y solución de problemas





## 1.3

## Configuración de pantalla de GX Works2

La configuración de pantalla de GX Works2 se muestra a continuación.

Coloque el mouse en un marco rojo para visualizar la función respectiva.

The screenshot displays the MELSOFT Series GX Works2 software interface. The main window shows a ladder logic program for a sub-program labeled [PRG] SUB. The program consists of three rungs:

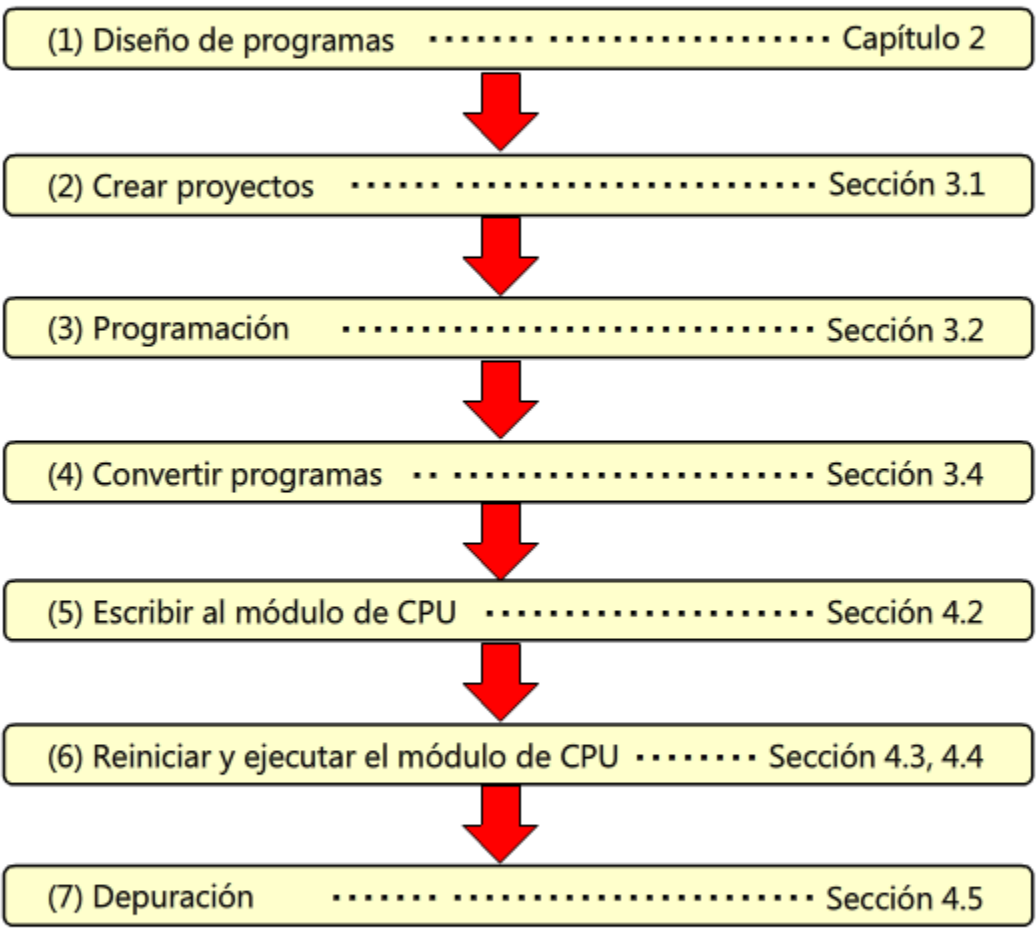
- Rung 0: A normally open contact labeled X6 (Stop Sw) and a normally closed contact labeled X7 (Door open sensor) are connected to output Y0 (Robot start signal).
- Rung 4: A normally open contact labeled Y0 (Robot start signal) is connected to output Y1 (Operation lamp).
- Rung 6: A normally open contact labeled Y0 (Robot start signal) is connected to output Y2 (Stop lamp).

A red rectangular box highlights the [PRG] MAIN window, which is a smaller view of the same ladder logic program. The [PRG] MAIN window shows the same three rungs as the [PRG] SUB window, but with a blue rectangular box highlighting the output Y2 (Stop lamp) connection on rung 6.

The interface includes a menu bar (Project, Edit, Find/Replace, Compile, View, Online, Debug, Diagnostics, Tool, Window, Help), a toolbar, and a navigation pane on the left. The navigation pane shows the project structure, including Parameter, Intelligent Function Module, Global Device Comment, Program Setting, POU, Program (MAIN, SUB), Local Device Comment, Device Memory, and Device Initial Value. The status bar at the bottom indicates the current step is 5/7 Step.

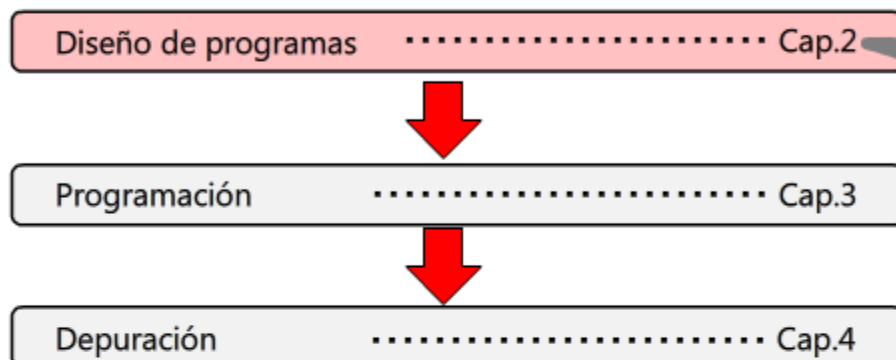
# 1.4 Procedimiento para la creación de un programa de secuencia

Cree un programa de secuencia de acuerdo al siguiente procedimiento.



## Capítulo 2 Crear datos de pantalla

En el Capítulo 2, aprenderá a diseñar programas, incluso definir los contenidos de control y convertirlos a un programa.



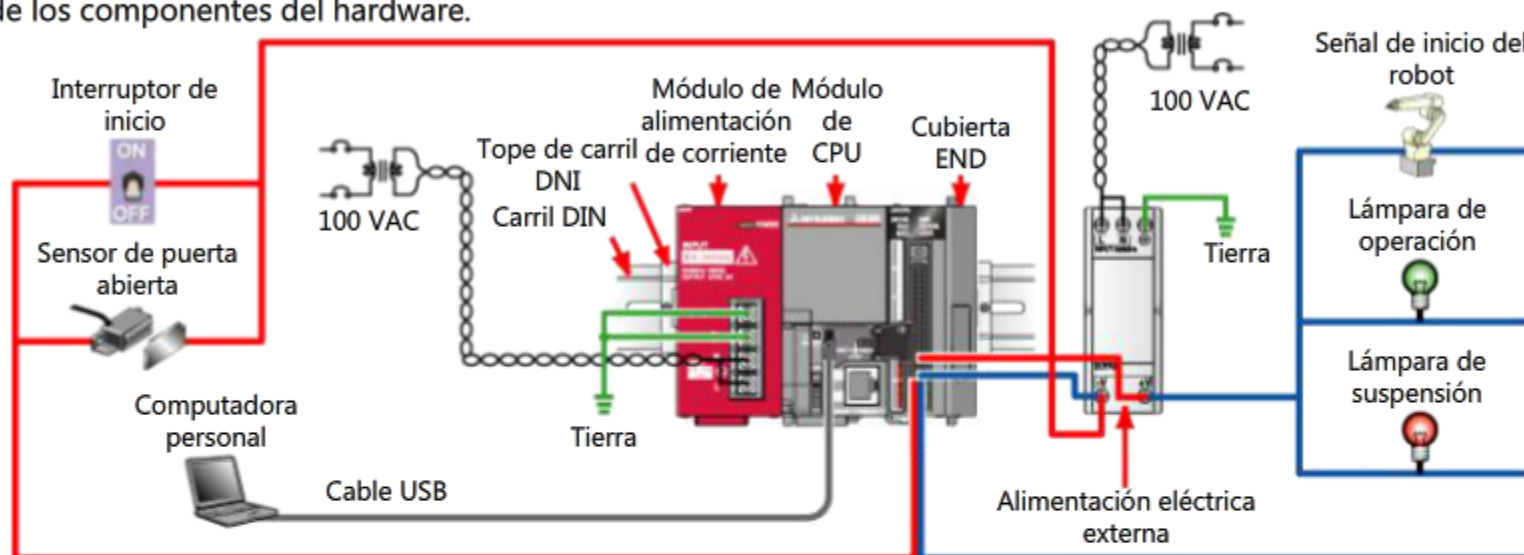
### Pasos para el aprendizaje en el Capítulo 2

- 2.1 Configuración de hardware del ejemplo  
Sistema usado para aprender
- 2.2 Definir los elementos de control
- 2.3 Creación de una tabla de correspondencia de E/S  
Dispositivos y números de dispositivo
- 2.4 Diseño de un programa

## 2.1 Configuración de hardware del sistema de ejemplo usado para el aprendizaje

En este curso, usted construirá un sistema de PLC (en adelante llamado "sistema de ejemplo"), el cual inicia al robot siguiendo un procedimiento.

A continuación, se muestra un diagrama de la configuración de hardware del sistema de ejemplo junto con una lista de los componentes del hardware.



Elemento	Componente	Modelo	Descripción
Sistema de PLC	Módulo de alimentación eléctrica	L61P	Suministra energía a los módulos incluyendo el módulo de CPU y módulo E/S.
	Módulo de CPU	L02CPU	Controla el sistema de PLC.
	Cubierta END	L6EC	Conectado en el lado derecho del bloque del sistema.
	Cable USB	MR-J3USBCBL3M	Conecta la computadora personal, en la cual GX Works2 está instalado, al módulo de CPU.
	Computadora personal	—	Funciona con el GX Works2 instalado.
Alimentación eléctrica externa	—	—	Suministra energía a los dispositivos E/S externos.
Equipo E/S externo	Interruptor	—	Fijado en ON (ENCENDER) para iniciar el control.
	Sensor	—	Detecta si la puerta está abierta o cerrada.
	Robot	—	Opera de acuerdo a las señales de control.
	Dos lámparas	—	Iluminan de acuerdo al estado de operación.

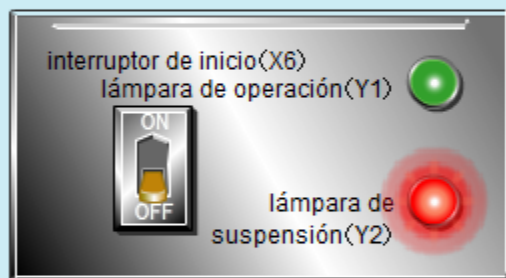
El primer paso para diseñar un programa es identificar los dispositivos que se deben controlar y los dispositivos E/S necesarios para el control deseado. En el sistema de ejemplo, se realiza el control de la operación de inicio y de suspensión de un robot. El robot no se iniciará si la puerta de la valla de seguridad está abierta, y se detendrá si la puerta está abierta durante la operación.

Vea la siguiente animación para entender mejor cómo operará el sistema de ejemplo.

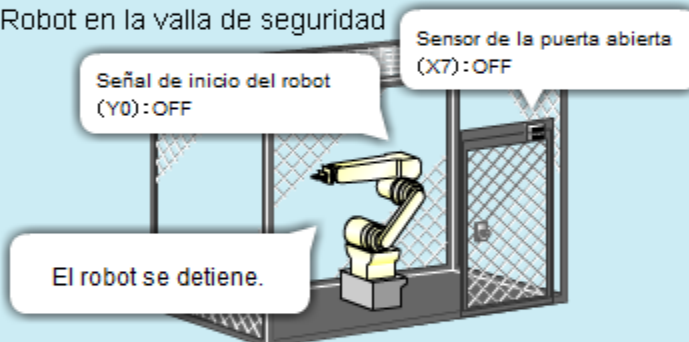
### Operación del sistema de ejemplo

 Haga clic dentro del círculo rojo

Panel de control del robot



Robot en la valla de seguridad



Cuando usted mueva el **interruptor de inicio (X6)** a la posición OFF, la **señal de inicio del robot (Y0)** se apaga para detener la operación del robot. Simultáneamente, la **lámpara de operación (Y1)** en el panel de control se apaga, y la **lámpara de suspensión (Y2)** se enciende.

Volver a reproducir



Anterior

## 2.3 Crear una tabla de correspondencia de dispositivos E/S y números de dispositivo

Es una buena idea crear una tabla que incluya a todos los dispositivos E/S y registros usados en un PLC junto con su información correspondiente para cualquier programa que se cree. Eso reduce la probabilidad de cometer errores durante el proceso de diseño y programación, y ayuda a incrementar la eficiencia de la programación. Si el sistema ya cuenta con una tabla de correspondencia, por ejemplo, una creada por la persona que configuró el hardware, úsela.

La tabla siguiente es una tabla de correspondencia para el sistema de ejemplo usado en este curso

Nombre de dispositivo E/S	No. de dispositivo	Tipo de E/S	Tipo de dispositivo	Descripción
Interruptor de inicio	X6	Entrada	Bit	Este interruptor inicia o detiene la operación del robot.
Sensor de puerta abierta	X7	Entrada	Bit	Este sensor verifica si la puerta de la valla de seguridad del robot está abierta. Cuando la puerta se abre, el sensor se enciende. Cuando la puerta se cierra, el sensor se apaga.
Señal de inicio del robot	Y0	Salida	Bit	Cuando esta señal se enciende, el robot inicia la operación.
Lámpara de operación	Y1	Salida	Bit	Esta lámpara se ilumina mientras el robot está en operación.
Lámpara de suspensión	Y2	Salida	Bit	Esta lámpara se ilumina mientras el robot está detenido.

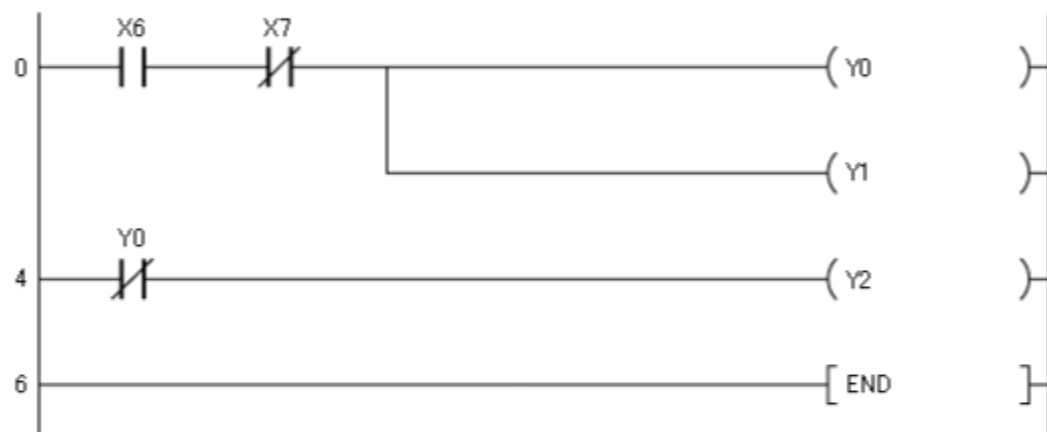
\* Si se usan datos de palabras, en la tabla se debe incluir el valor inicial, el rango de configuración (límites superior e inferior), el tipo de datos (con signo, real, etc.), y el comentario. Esta información será útil para diseñar y modificar programas.

## 2.4

## Diseñar programas

Diseñe un programa usando el lenguaje de lógica en Escalera (Ladder Logic) en base a los elementos de control y la tabla de correspondencia de E/S. El programa en escalera y la tabla de correspondencia de E/S diseñados para el sistema de ejemplo se muestran a continuación.

## Programa en escalera



## Tabla de correspondencia E/S

Nombre de dispositivo E/S	Tipo	No. de dispositivo
Interruptor de inicio	Entrada	X6
Sensor de puerta abierta	Entrada	X7
Señal de inicio del robot	Salida	Y0
Lámpara de operación	Salida	Y1
Lámpara de suspensión	Salida	Y2

## Capítulo 3 Programación

En el Capítulo 3, aprenderá a programar el programa diseñado usando GX Works2.

Diseño de programas ..... Cap.2



Programación ..... Cap.3



Depuración ..... Cap.4

### Pasos para el aprendizaje en el Capítulo 3

- 3.1 Crear proyectos
- 3.2 Crear programas
- 3.3 Hacer que los programas sean fáciles de entender
- 3.4 Convertir programas a una forma ejecutable
- 3.5 Guardar proyectos



## 3.1

## Crear proyectos

El primer paso para escribir un programa es crear un proyecto.

El proyecto es una colección de datos que GX Works2 utiliza para manejar los programas.

La tabla siguiente contiene los componentes principales de un proyecto.

Tipo de datos	Descripción
Programa	El código de fuente y código compilado para las operaciones secuenciales de CPU.
Comentario	Un tipo de documentación visualizada dentro del programa. Vea la Sección 3.3 "Hacer que los programas sean fáciles de entender" para tener más detalles.
Parámetro	Contiene la mayor parte o toda la información de ajuste y configuración para un sistema.
Configuración de transferencia	La información de ruta de conexión necesaria para establecer las comunicaciones entre el sistema que opera GX Works2 y el módulo de CPU.

### Programa en escalera

GX Works2 le permite seleccionar los dos tipos de proyecto siguientes.

El programa de ejemplo en este curso utiliza el tipo "proyecto simple".

Tipo de proyecto	Descripción
Proyecto simple	Este tipo de proyecto tiene compatibilidad regresiva con proyectos de GX Developer. Los proyectos simples después pueden ser convertidos en proyectos estructurados, pero no a la inversa.
Proyecto estructurado	Estos proyectos son capaces de usar un lenguaje de programación adicional llamado Escalera Estructurada. Adicionalmente, los programas se pueden separar en muchas partes pequeñas y las piezas de código utilizadas con frecuencia se pueden modularizar y volver a usar mediante una biblioteca de usuario. Las etiquetas pueden ser modularizadas de manera similar para volver a usarlas con facilidad. Esto puede mejorar la eficiencia de programación y depuración, especialmente para proyectos muy grandes.

### Etiquetas

Las etiquetas son nombres creados por los usuarios que se convierten en un nombre alternativo para las direcciones de dispositivos. Se pueden usar a nivel global, local o a nivel de sistema cuando se implementan junto con MELSOFT Navigator. Los proyectos simples se pueden crear con o sin la capacidad de usar etiquetas. Para el proyecto de ejemplo, no se utilizarán las etiquetas.

Para comenzar a crear el proyecto de ejemplo, realice las configuraciones siguientes.

Antes de crear un proyecto, se debe conocer el nombre del modelo y de la serie del controlador programable, al igual que el tipo de proyecto que se utilizará.

Elemento	Descripción
Tipo de proyecto	El tipo de proyecto determina las características que están disponibles al escribir los programas. Para este ejemplo, escoja "proyecto simple".
Etiqueta de uso	Si se requiere la capacidad de escribir programas usando etiquetas, marque este elemento. El programa de ejemplo no usa etiquetas. Por lo tanto, deje esta casilla sin marcar.
Serie PLC	La serie PLC determina los modelos disponibles a ser seleccionados en la lista desplegable del tipo de PLC. Para este ejemplo, escoja "LCPU".
Tipo de PLC	El tipo de PLC determina cómo el compilador convierte los programas del usuario a código de máquina. Escoja el modelo de PLC que se programará, en este caso, "L02".
Lenguaje de programación	El lenguaje de programación determina el tipo de programa del primer programa creado automáticamente (MAIN). Luego se pueden añadir programas adicionales usando lenguajes diferentes. Para este ejemplo, escoja "Ladder" (Escalera).

Revise la página siguiente, donde se simulan los procesos para crear un proyecto nuevo.

## 3.1

## Crear proyectos

MELSOFT Series GX Works2

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help



Navigation

Project



Project

User Library

Connection Destination

New Project

Project Type:

Simple Project

OK

Cancel

 Use Label

PLC Series:

LCPU

PLC Type:

L02

Language:

Ladder

Ahora se creó un nuevo proyecto.

Haga clic en  para continuar.

## 3.2

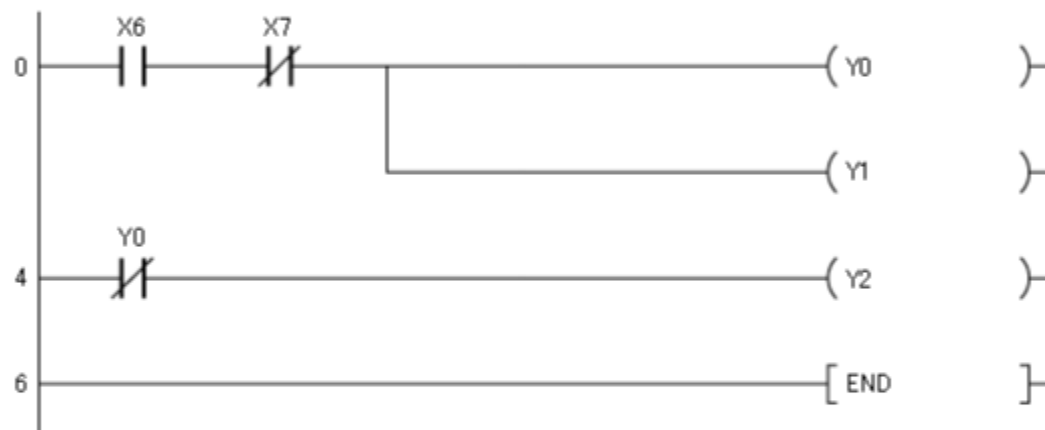
## Crear programas

Luego de crear un proyecto, creemos un programa.

Cree el siguiente programa y conozca las operaciones básicas (entrada de instrucción, cambiar, eliminar, copiar y pegar, e ingresar/eliminar línea de regla).

A continuación, se muestra el programa diseñado para el sistema de ejemplo en el Capítulo 2.

### Programa para el sistema de ejemplo



En la página siguiente, intente crear este programa usando la ventana simulada.

## 3.2

## Crear programas

MELSOFT Series GX Works2 (Unset Project) - [[PRG] MAIN]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

Navigation [PRG] MAIN

Project

- Parameter
- Intelligent Function Module
- Global Device Comment
- Program Setting
- POU
  - Program
    - MAIN
    - Local Device Comment
- Device Memory
- Device Initial Value


Unlabeled

L02

Host Station

0/15Step

NL

Ahora el programa de circuito de Escalera está completo.  
Haga clic en  para continuar.

## 3.3

## Hacer que los programas sean fáciles de entender

En su estado actual, la representación visual del programa solo contiene dispositivos, instrucciones, líneas y números de pasos. Cuando se ve un programa complejo, puede ser difícil determinar lo que hace el programa.

- Es difícil encontrar errores de programación como las instrucciones o los números de dispositivo incorrectos.
- En general, es difícil realizar el análisis operativo, la depuración y la expansión de programa.
- Si el desarrollador del programa original ya no puede mantener el programa, la tarea de saber cómo opera el programa puede ser abrumadora y quizá imposible para cualquier persona.

### Contra medidas

Incluyen **documentación** en el programa, para que todos puedan entender rápidamente cómo funciona el programa. Como buena práctica, todos los programadores deben añadir comentarios detallados a sus programas para permitirse a sí mismos y a otros entender mejor el programa.



GX Works2 permite que se usen tres tipos diferentes de comentarios. Para más detalles, consulte el manual de Proyectos Simples de GX Works2.

Tipo de comentario	Alcance del comentario
Comentario del dispositivo	Entrada de hasta 32 caracteres que se visualizarán bajo el dispositivo seleccionado (E/S u otra dirección de memoria).
Declaración	Entrada de hasta 64 caracteres por declaración que se añadirán a la parte superior del bloque de escalera seleccionado (encima del número de paso). Cada bloque de escalera puede tener declaraciones múltiples.
Nota	Entrada de hasta 32 caracteres que se visualizarán encima de la bobina seleccionada o instrucción de aplicación.

La página siguiente simula el proceso de añadir comentarios de dispositivo al programa de ejemplo.

## 3.3

## Hacer que los programas sean fáciles de entender

The screenshot shows the MELSOFT Series GX Works2 software interface. The title bar reads "MELSOFT Series GX Works2 (Unset Project) - [[PRG] MAIN]". The menu bar includes Project, Edit, Find/Replace, Compile, View, Online, Debug, Diagnostics, Tool, Window, and Help. The navigation pane on the left shows the project structure, including Parameter, Intelligent Function Module, Global Device Comment, Program Setting, POU, Program (MAIN), Local Device Comment, Device Memory, and Device Initial Value. The main workspace displays a ladder logic program with the following components:

- Step 0: A normally open contact labeled "Start switch" (X6) and a normally closed contact labeled "Door open sensor" (X7) are connected to a coil for output Y0, labeled "Robot start signal".
- Step 4: A normally open contact labeled "Robot start signal" (Y0) is connected to a coil for output Y2, labeled "Stop lamp".
- Step 6: The program ends with an [END] instruction.

A tooltip message is displayed at the bottom right of the workspace, stating: "La entrada del comentario de dispositivo está completa. Haga clic en [Play icon] para continuar." (The device comment input is complete. Click on [Play icon] to continue.)

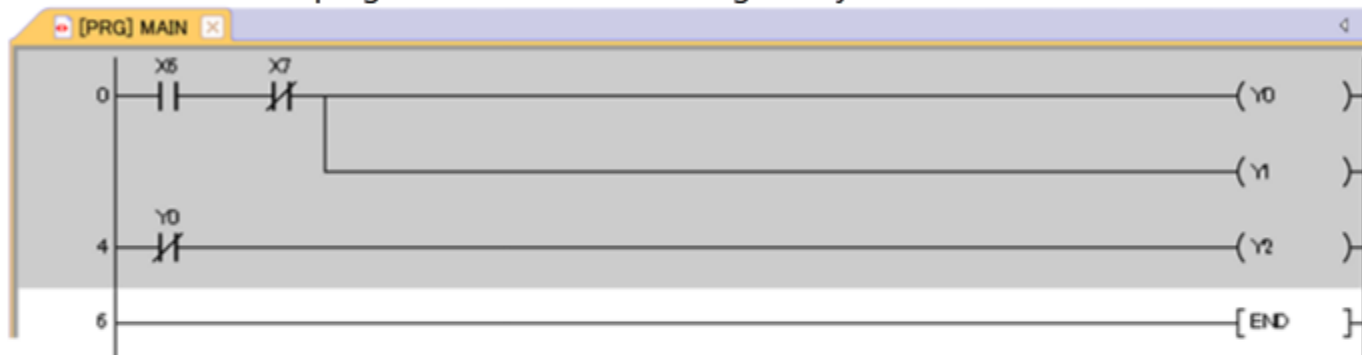
At the bottom of the software window, the status bar shows "Unlabeled", "L02", "Host Station", "5/75Step", and "NL".



## 3.4 Convertir programas a una forma ejecutable

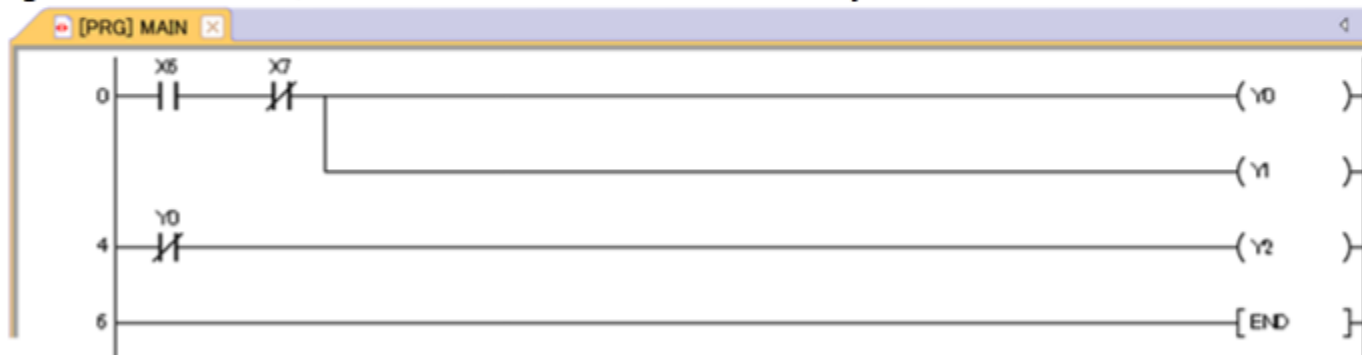
Luego de completar el programa, necesita convertirlo a una forma que pueda ejecutarse en el módulo de CPU. Los programas no convertidos no pueden ejecutarse o guardarse.

El color de fondo de los programas no convertidos es gris tal y como se muestra a continuación.



Convertir

Luego de la conversión, el color del fondo cambia a blanco tal y como se muestra a continuación.



En la página siguiente, intente convertir el programa usando la ventana simulada.



## 3.4

## Convertir programas a una forma ejecutable



MELSOFT Series GX Works2 (Unset Project) - [[PRG] MAIN]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

Navigation [PRG] MAIN

Project

- Parameter
- Intelligent Function Module
- Global Device Comment
- Program Setting
- POU
  - Program
    - MAIN
  - Local Device Comment
- Device Memory
- Device Initial Value

Unlabeled

L02

Host Station

5/75Step

NL


0 X6 X7 Y0 Y1 Y2

Start switch Door open sensor Robot start signal Operation lamp

4 Y0 Robot start signal

6 [END]

Quando el programa se convierta, el color del fondo cambia de gris a blanco.

El programa se ha convertido.  
Haga clic en  para continuar.

## 3.5 Guardar proyectos

Luego de terminar la conversión del programa, guarde el proyecto incluyendo los programas. Si se termina GX Works2 sin guardar el proyecto, los programas asociados serán descartados, así que debe guardar su proyecto cada cierto tiempo. Al guardar un proyecto nuevo, especifique los siguientes tipos de información del proyecto. (Esto no se requiere para guardar sobrescritura.)

Debe incluir la información que haga que sea fácil para los demás entender los contenidos del control del programa, el nombre del sistema, etc.

Elemento	Requerido	Descripción
Ruta de destino para guardar	✓	Especifique la carpeta donde un espacio de trabajo está asignado.
Lista del espacio del trabajo/proyecto		Si ya existe uno o más espacios de trabajo en la carpeta especificada en "Save destination path" (Ruta de destino para guardar), los espacios de trabajo existentes aparecerán.
Nombre del espacio de trabajo	✓	Especifique un nombre para el espacio de trabajo con hasta 128 caracteres.
Nombre de proyecto	✓	Especifique un nombre para el proyecto con hasta 128 caracteres.
Título		Especifique un título para el proyecto con hasta 128 caracteres. Este parámetro es útil cuando quiere asignar un nombre largo que no entra en el "Nombre del proyecto".

**El espacio de trabajo** es una carpeta para gestionar proyectos múltiples.

Un ejemplo sobre el uso de un espacio de trabajo se muestra a continuación. (Los proyectos se gestionan para cada tipo de vehículo en la línea de fabricación de automóviles.)

Nombre del espacio de trabajo	Nombre de proyecto	Título
Línea de fabricación de automóviles	Línea de fabricación de tipo-A	Programa operativo normal para controlar la línea de fabricación de tipo-A
	Línea de fabricación de tipo-B	Programa operativo normal para controlar la línea de fabricación de tipo-B
	Línea de fabricación de tipo-C	Programa operativo normal para controlar la línea de fabricación de tipo-C

### Notas:

- Si se guarda un proyecto que contiene un programa no convertido, solo se descartará el programa no convertido. Antes de guardar un proyecto, realice la conversión del programa tal y como lo aprendió en la Sección 3.4.
- Especifique la ruta de destino para guardar, el nombre del espacio de trabajo, y el nombre del proyecto de manera que el número total de caracteres no se exceda de 150.

En la página siguiente, intente guardar el proyecto usando la ventana simulada.

## 3.5

## Guardar proyectos

MELSOFT Series GX Works2 C:\SequenceProgram\Learning\Robot\_Control - [[PRG] MAIN]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

Navigation [PRG] MAIN

Project

- Parameter
- Intelligent Function Module
- Global Device Comment
- Program Setting
- POU
  - Program
    - MAIN
  - Local Device Comment
- Device Memory
- Device Initial Value


Y0 Robot start signal

Y1 Operation lamp

Y2 Stop lamp

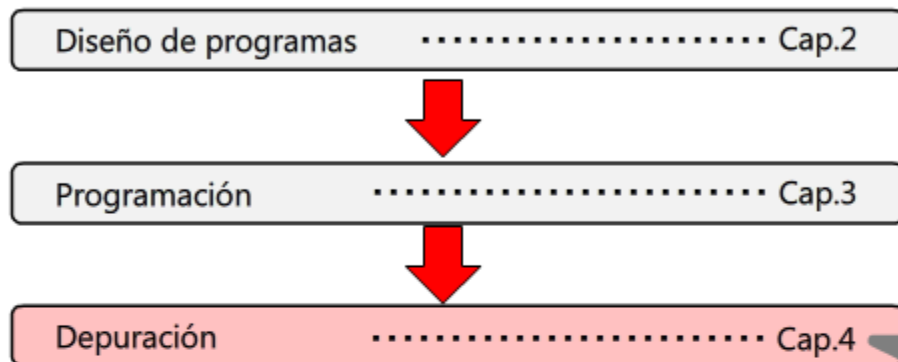
END

Unlabeled L02 Host Station 6/75Step

El proyecto se ha guardado.  
Haga clic en  para continuar.

## Capítulo 4 Depuración

En el Capítulo 4, aprenderá a escribir programas de secuencia al módulo de CPU y a depurarlos.



### Pasos para el aprendizaje en el Capítulo 4

#### 4.1 Depuración

- 4.1.1 Depurar un programa sin usar el módulo de CPU
- 4.1.2 Cambiar el estado de un dispositivo E/S
- 4.1.3 Supervisar el estado del dispositivo

#### 4.2 Escribir programas al módulo de CPU

#### 4.3 Habilitar programas escritos

#### 4.4 Ejecutar programas

#### 4.5 Depurar programas

#### 4.6 Verificar la operación del sistema de PLC

#### 4.7 Operar el sistema de PLC

#### 4.8 Conclusión

Una vez que se ha escrito un programa o un segmento de programa, es necesario probar el código para asegurarse de que funcione como se espera.

Los errores de software (cuando el código escrito no funciona como debería) se llaman "bugs", y el proceso para encontrar la causa del funcionamiento no deseado y corregirlo se conoce como "debugging" (depuración).

Probar y depurar son pasos esenciales en la creación de programas.

En especial en los controladores programables porque si hay bugs presentes, estos podrían ocasionar la suspensión del sistema, daños al equipo, u otros accidentes.

La tabla siguiente contiene algunas de las funciones en GX Works2 que pueden ayudar con el proceso de depuración.

Nombre de la función	Descripción
Simulador	Esta función se usar para simular la ejecución de un programa incluso sin un módulo de CPU. Esta función se puede usar para depurar en un entorno en el cual no haya un módulo de CPU disponible.
Supervisión	Esta función permite supervisar el estado de ejecución y el estado de cada dispositivo durante la ejecución del módulo de CPU. Las múltiples funciones de supervisión están disponibles según la aplicación, como supervisar en la escalera, supervisar solo los dispositivos registrados y supervisar todos los dispositivos en un batch.
Cambiar el valor actual	Esta función puede cambiar a la fuerza el estado del dispositivo (bit; ON ↔ OFF, palabra: valor actual) durante la ejecución del módulo de CPU. Esta función es útil para cambiar el valor actual de un dispositivo de palabras o el estado de un relé interno.
Registro/cancelación de entrada salida forzada	Esta función puede cambiar a la fuerza el estado (ON ↔ OFF) de un dispositivo E/S registrado durante la ejecución del módulo de CPU. Para verificación de operación o depuración solo con un módulo de CPU, esta función se puede usar como sustituto de un interruptor.

Estas funciones se explican más detalladamente con respecto al proceso de depuración en lo que resta del capítulo.

### Notas sobre depuración

No realice tareas de depuración mientras el controlador programable está conectado a dispositivos E/S físicos.

Los errores de software en el programa, dispositivos E/S forzados, o cambios en el valor de palabras podrían ocasionar daños en el equipo externo o algo peor.

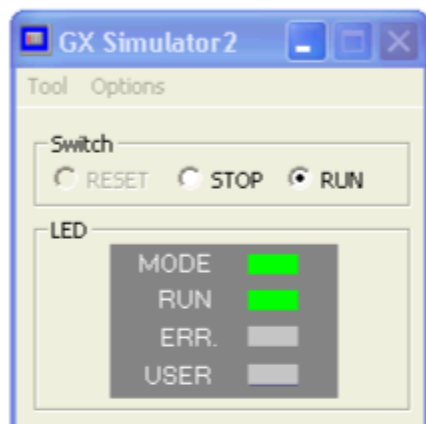
Si un sistema de PLC desconectado no está disponible, use la función de simulador.


## 4.1.1


## Depurar un programa sin usar el módulo de CPU

Si un módulo de CPU no está disponible para depuración, use la **función de simulador**.

Un programa puede funcionar en un módulo de CPU virtual provisto por el software sin usar el módulo de CPU real.



 On (encendido)

 Off (apagado)

Elemento	Estado	Description
Switch (interruptor)	RUN (EJECUTAR)	Ejecuta el módulo de CPU virtual.
	STOP (ALTO)	Detiene el módulo de CPU virtual.
	RESET (REINICIAR)	Reinicia el módulo de CPU virtual. (Habilitado solo en el estado STOP)
LED	MODE (MODO)	Indica el estado MODE del CPU virtual.
	RUN (EJECUTAR)	Indica el estado de ejecución del CPU virtual. •On (encendido): estado RUN •Off (apagado): estado STOP
	ERR	Indica el estado de error del módulo de CPU virtual. Si hay un error, el LED se encenderá o parpadeará.
	USER (USUARIO)	Indicar si se ha cometido un error de usuario en el CPU virtual. Se enciende o parpadea cuando ocurre un error.

### Notas sobre el uso de la función de simulador

- Depurar usando la función de simulador no garantiza que el programa secuencial funcione correctamente luego de la depuración.
- La función de simulador ejecuta la entrada/salida de datos con módulos de E/S usando la memoria de simulación. La función no soporta algunas instrucciones, funciones, y memoria de dispositivo. Por lo tanto, los resultados de la operación con la función de simulador pueden ser diferentes de los del módulo de CPU real.

En la página siguiente, intente utilizar la función de simulador con la ventana simulada.



## 4.1.1

## Depurar un programa sin usar el módulo de CPU



MELSOFT Series GX Works2 C:\SequenceProgram\Learning\Robot\_Control - [[PRG] MAIN]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

Navigation [PRG] MAIN

Project

- Parameter
- Intelligent Function Module
- Global Device Comment
- Program Setting
- POU
  - Program
    - MAIN
  - Local Device Comment
- Device Memory
- Device Initial Value

Unlabeled

L02

Host Station

6/75Step

NL

```
graph TD
    subgraph Step0 [0]
        X6[Start Switch] --- X7[Door open sensor]
    end
    subgraph Step4 [4]
        Y0[Robot start signal]
    end
    subgraph Step6 [6]
        END[END]
    end
    X6 --> Y0
    X7 --> Y0
    Y0 --> END
```

Ahora ya sabe cómo usar la función de simulador.  
Haga clic en para continuar.

## 4.1.2

## Cambiar el estado de un dispositivo E/S

Al depurar un programa de secuencia con un módulo de CPU al cual ningún dispositivo E/S está conectado o al usar la función de simulador, utilice la función **Forced Input Output Registration/Cancellation (Registro/cancelación de entrada salida forzada)** para cambiar el estado ON/OFF (ENCENDIDO/APAGADO) de un dispositivo E/S.

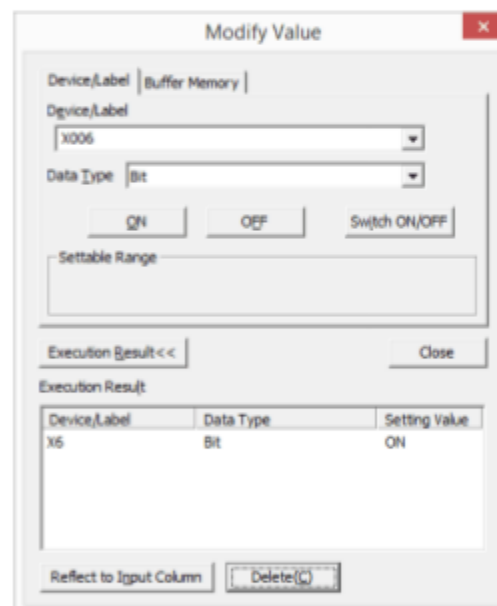
El estado de los dispositivos E/S registrados se pueden cambiar a la fuerza a ON u OFF con el software.

(Series MELSEC-Q y MELSEC-L): En la pantalla "Forced Input Output Registration/Cancellation" (Registro/Cancelación de Entrada/Salida Forzada)

(Serie MELSEC-F): De la pantalla "Modify Value" (Modificar Valor)



Pantalla Forced Input Output Registration/Cancellation (Registro/Cancelación de Entrada/Salida Forzada) (Series MELSEC-Q y MELSEC-L)



Pantalla Modify Value (Modificar Valor) (Serie MELSEC-F)

### Para cambiar los estados de otros dispositivos

Para cambiar el dispositivo actual de un dispositivo de palabras o el estado ON/OFF de un relé interno, use la **función de cambio de valor actual**.

Para más detalles, consulte el manual.



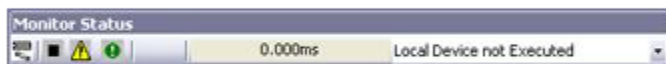
## 4.1.3 Supervisar el estado del dispositivo

Cuando se inicie la simulación, la supervisión comenzará automáticamente. Para ingresar al modo de supervisión cuando se está conectado a un CPU de controlador programable, simplemente haga clic en Online (En línea), Monitor (Supervisión), y luego Start Monitoring (Iniciar supervisión). O use el acceso directo en el teclado, F3.

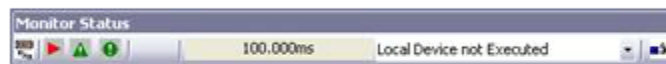
Durante el modo de supervisión se puede ver que los valores y estados de todos los dispositivos en el programa se superponen al código del programa. Esto le permite al usuario ver los valores que cambian incluyendo los efectos de usar la función "Registro/cancelación de entrada salida forzada".

Además, la barra del Estado de Supervisión aparece e incluye información básica para determinar el estado de CPU o CPU virtual. Consulte la tabla siguiente para entender la información provista por la barra del Estado de Supervisión.

**Al estar conectado al módulo de CPU**



**Al usar la función de simulador**



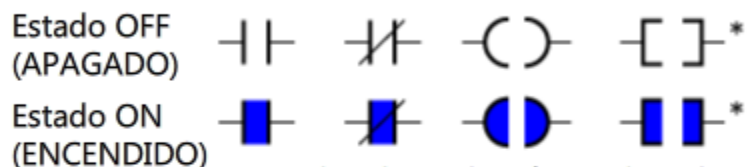
Estado	Ícono/indicación	Descripción
Estado de conexión	Al estar conectado al módulo de CPU	Visualiza el estado de la conexión con el módulo de CPU o la función de simulador.
	Al usar la función de simulador	
Estado RUN/STOP (EJECUTAR/ALTO)	RUN (EJECUTAR)	Visualiza el estado ejecutar del CPU (RUN o STOP).
	STOP (ALTO)	
Estado ERR.	ERR. apagado	Visualiza el estado de error del módulo de CPU.
	ERR. encendido	
	ERR parpadeando	
Estado USER (USUARIO)	USER apagado	Visualiza el estado de usuario del módulo de CPU.
	USER encendido	
	USER parpadeando	
Tiempo de escaneo		Visualiza el tiempo de escaneo máximo del módulo de CPU que se está supervisando.
Estado de presencia/ausencia de instrucción sin soporte	Existe instrucción sin soporte.	Visualiza si existe una instrucción sin respaldo cuando se ejecuta la función de simulador.
	No existe instrucción sin soporte.	Hacer clic en el ícono abre la ventana de Dispositivo/Instrucción Sin Respaldo.

## 4.1.3 Supervisar el estado del dispositivo

Durante el modo de supervisión, el estado actual de todos los dispositivos en el programa se vuelve visible.

### Visualización del estado de dispositivo bit (ON/OFF)

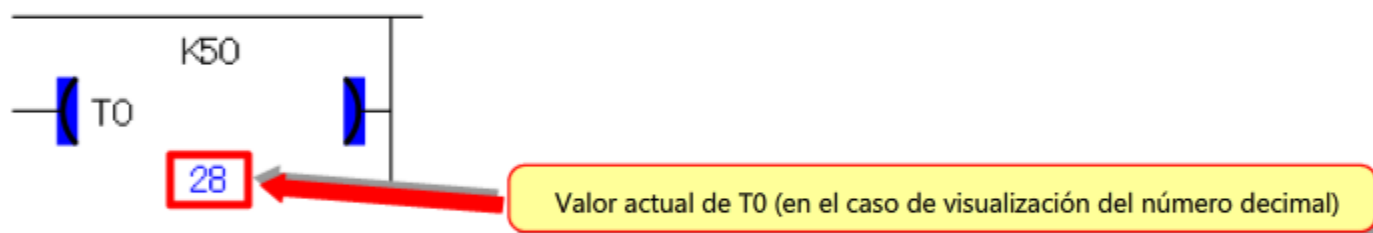
El estado ON/OFF (ENCENDIDO/APAGADO) se visualiza durante la supervisión tal y como se muestra a continuación.



\* Esta clase de visualización se aplica solo a SET, RST, PLS, PLF, SFT, SFTP, MC, e instrucciones de comparación de tipo de contacto. Note que para la instrucción RST, solo se visualiza el estado ON/OFF.

### Visualización de valor actual del dispositivo de palabras (visualización del número decimal/hexadecimal)

El valor actual durante la supervisión se visualiza tal y como se muestra a continuación.



### Supervisar solo dispositivos específicos

Al supervisar un programa muy largo o complejo, puede ser beneficioso supervisar solamente ciertos dispositivos de interés. Para lograr esto, GX Works2 incluye ventanas de observación que permiten al usuario añadir fácilmente los dispositivos en los que están interesados, ver su estado actual, y modificar sus valores durante la supervisión. Para más detalles, consulte el Manual de Operación de GX Works2 (común).

Watch 1					
Device/Label	Current Value	Data Type	Class	Device	Comment
X7	--	Bit		X7	Door open sensor
Y0	--	Bit		Y0	Robot start signal
Y1	--	Bit		Y1	Operation lamp
Y0	--	Bit		Y0	Robot start signal
Y2	--	Bit		Y2	Stop lamp
Y0	--	Bit		Y0	Robot start signal

## 4.2

## Escribir programas al módulo de CPU

Antes de depurar algo usando un módulo de CPU real, coloque el CPU en **modo ALTO**, asegúrese de que se haya establecido una conexión al CPU, y escriba los programas y parámetros a la memoria de programa.

Así como se ve en la captura de pantalla siguiente, las funciones principales de la ventana **Write to PLC (Escribir a PLC)** permiten que el usuario seleccione los archivos deseados que se van a escribir, que elija su ubicación, y que confirme la capacidad de memoria del CPU. Los tres botones encima de la lista de archivos le permiten al usuario seleccionar rápidamente los archivos deseados que se van a escribir. El más común, el cual se usa en la simulación siguiente, es **"Parameter+Program (Parámetro+Programa)."**

Lista de archivos

Capacidad de memoria

Module Name/Data Name	Title	Target	Detail	Last Change	Target Memory	Size
Robot_Control						
PLC Data					Program Memory/D...	
Program(Program File)		<input checked="" type="checkbox"/>				
MA2N		<input checked="" type="checkbox"/>		2010/08/05 15:22:24		2256 Bytes
Parameter		<input checked="" type="checkbox"/>				
PLC(Network/Remote Password/Switch Setting)		<input checked="" type="checkbox"/>		2010/08/05 15:22:24		2936 Bytes
Global Device Comment		<input type="checkbox"/>				
COMMENT		<input type="checkbox"/>		2010/08/05 15:25:26		
Device Memory		<input type="checkbox"/>				
MA2N		<input type="checkbox"/>		2010/08/05 15:22:25		

Necessary Setting:  No Setting /  Already Set /  Set if it is needed /  No Setting /  Already Set

Writing Size: 5,192bytes | Free Volume: 76,728 | Use Volume: 5,192bytes | Refresh

Related Functions: c.c

Execute Close

En la página siguiente, intente escribir el módulo de CPU usando la ventana simulada.

## 4.2

## Escribir programas al módulo de CPU

MELSOFT Series GX Works2 C:\SequenceProgram\Learning\Robot\_Control - [[PRG] MAIN]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

Navigation [PRG] MAIN

Project

- Parameter
- Intelligent Function Module
- Global Device Comment
- Program Setting
- POU
  - Program
    - MAIN
  - Local Device Comment
- Device Memory
- Device Initial Value

Project


User Library

Connection Destination

```

graph TD
    subgraph Step0 [0]
        X6[Start Switch] --- X7[Door open sensor]
    end
    subgraph Step4 [4]
        Y0[Robot start signal]
    end
    subgraph Step6 [6]
        END[END]
    end
    X6 --> Y0
    X7 --> Y0
    Y0 --> Y1[Operation lamp]
    Y0 --> Y2[Stop lamp]
  
```

Unlabeled L02 Host Station 6/75Step NL

El programa se ha escrito al módulo de PLC.  
Haga clic en  para continuar.

## 4.3 Habilitar programas escritos

(Serie MELSEC-F): La siguiente operación no es necesaria.

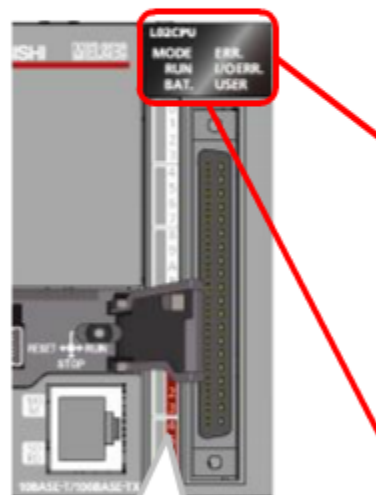
(Series MELSEC-Q y MELSEC-L): La siguiente operación es necesaria.

Luego de escribir un programa al módulo de CPU, **reinicie** el módulo de CPU.

Los programas escritos no se habilitan a menos que el módulo de CPU se reinicie.

\* Esta operación no se requiere si la función de simulador se usa para depuración.

Reinicie el módulo de CPU de la siguiente manera:



Interruptor  
REINICIAR/  
ALTO/EJECUTAR

- Presione y mantenga presionado el interruptor RESET/STOP/RUN (REINICIAR/ALTO/EJECUTAR) en el panel delantero del módulo de CPU hacia la posición RESET (durante 1 segundo o más).

[Reinicio en progreso]

L02CPU		MODE :	Encendido en verde
MODE	ERR.	RUN	: Apagado
RUN	I/OERR.	ERR.	: Parpadeando
BAT.	USER		

Presione durante 1  
segundo o más.



- Suelte el interruptor luego de que la luz de MODE LED y el parpadeo de ERR. LED se apaguen.

[Reinicio completo]

L02CPU		MODE :	Encendido en verde
MODE	ERR.	RUN	: Apagado
RUN	I/OERR.	ERR.	: Apagado
BAT.	USER		

- El interruptor regresa a la posición de ALTO para completar el reinicio.



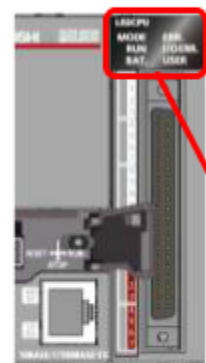
# 4.4 Ejecutar programas

## Serie MELSEC-Q y MELSEC-L

Luego de completar el reinicio, ejecute el programa.  
Coloque el módulo de CPU en el estado RUN (EJECUTAR) de la siguiente manera para ejecutar el programa.

\* Esta operación no se requiere si la función de simulador se usa para depuración.

- (1) Mueva el interruptor RESET/STOP/RUN (REINICIAR/ALTO/EJECUTAR) en el panel delantero del módulo de CPU a la posición RUN.



Visualización LED en estado STOP

L02CPU		
MODE <input type="checkbox"/>	ERR.	MODE : Encendido en verde
RUN <input type="checkbox"/>	I/OERR.	RUN : Apagado
BAT. <input type="checkbox"/>	USER	



- (2) Si el LED de RUN (EJECUTAR) se enciende en verde, el programa está funcionando con normalidad.

Visualización LED en estado RUN

L02CPU		
MODE <input type="checkbox"/>	ERR.	MODE : Encendido en verde
RUN <input checked="" type="checkbox"/>	I/OERR.	RUN : Encendido en verde
BAT. <input type="checkbox"/>	USER	

## Serie MELSEC-F

Luego de escribir un programa en la unidad principal, coloque la unidad principal en el estado RUN (EJECUTAR) de la siguiente manera para ejecutar el programa. (No es necesario reiniciar.)

- (1) Mueva el interruptor RUN/STOP (EJECUTAR /ALTO) en el panel delantero de la unidad principal a la posición RUN (EJECUTAR).



Visualización LED en estado STOP

POWER	<input checked="" type="checkbox"/>
RUN	<input type="checkbox"/>
BATT	<input type="checkbox"/>
ERROR	<input type="checkbox"/>

Visualización LED en estado STOP

- (2) Si el LED de RUN (EJECUTAR) se enciende, el programa está funcionando con normalidad.

Visualización LED en estado RUN

POWER	<input checked="" type="checkbox"/>
RUN	<input checked="" type="checkbox"/>
BATT	<input type="checkbox"/>
ERROR	<input type="checkbox"/>

Visualización LED en estado RUN

## 4.5

## Depurar programas

Luego de ejecutar el módulo de CPU, use la función de registro/cancelación entrada salida forzada para cambiar el estado de cada dispositivo y supervisar el resultado (salida) en la escalera.

(Ejemplo de pantalla de las series MELSEC-Q y MELSEC-L)

The screenshot shows the MELSOFT GX Works2 interface. A dialog box titled "Forced Input Output Registration/Cancellation" is open, displaying a table for device registration. The table has columns for "No.", "Device", and "ON/OFF". The first two rows are filled: No. 1 with Device X6 and ON/OFF ON; No. 2 with Device X7 and ON/OFF OFF. The rest of the rows are empty. Buttons for "Register FORCE ON", "Register FORCE OFF", "Cancel Registration", "Update Status", "Batch Cancel Registration", and "Close" are visible. In the background, a ladder logic diagram is partially visible, showing a network with inputs X6 and X7, and outputs Y0, Y1, and Y2.

No.	Device	ON/OFF	No.	Device	ON/OFF
1	X6	ON	17		
2	X7	OFF	18		
3			19		
4			20		
5			21		
6			22		
7			23		
8			24		
9			25		
10			26		
11			27		
12			28		
13			29		
14			30		
15			31		
16			32		

En la página siguiente, intente depurar el programa usando la ventana simulada.

## 4.5

## Depurar programas

MELSOFT Series GX Works2 C:\SequenceProgram\Learning\Robot\_Control - [[PRG] MAIN]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

1.000ms Local Device not Executed

Navigation

Project

- Parameter
- Intelligent Function Module
- Global Device Comment
- Program Setting
- POU
  - Program
    - MAIN
- Local Device Comment
- Device Memory
- Device Initial Value

Project

User Library

Connection Destination

```
graph TD
    subgraph Step_0 [0]
        X6[Start Switch] --- AND
        X7[Door open sensor] --- AND
        AND --- Y0[Robot start signal]
        AND --- Y1[Operation lamp]
    end
    subgraph Step_4 [4]
        Y0[Robot start signal] --- Y2[Stop lamp]
    end
    subgraph Step_6 [6]
        END[END]
    end
```

Unlabeled L02 Host Station 6/75Step N/A

La depuración del programa está completa.  
Haga clic en para continuar.



## 4.6

## Verificar la operación del sistema de PLC

Luego de completar la depuración del programa, escriba el programa al sistema de PLC real para finalizar de verificar la operación.

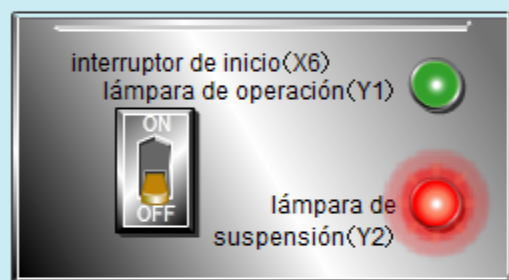
Opere el equipo E/S real para confirmar que funciona según lo designado.

Incluso al operar el equipo E/S, el estado de cada dispositivo se puede verificar usando la función de supervisión de GX Works2.

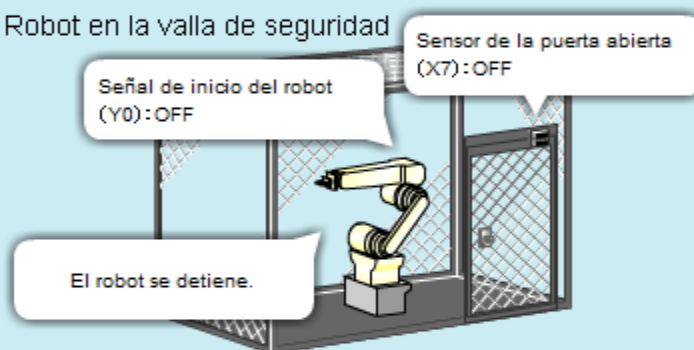
## Operación del sistema de ejemplo

 Haga clic dentro del círculo rojo

Panel de control del robot



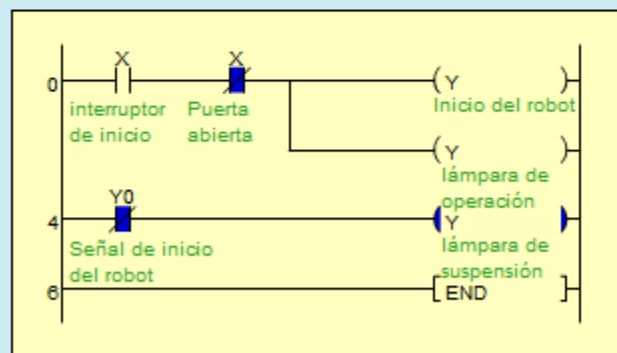
Robot en la valla de seguridad



Cuando usted mueva el **interruptor de inicio (X6)** a la posición OFF, la **señal de inicio del robot (Y0)** se apaga para detener la operación del robot. Simultáneamente, la **lámpara de operación (Y1)** en el panel de control se apaga, y la **lámpara de suspensión (Y2)** se enciende.

Volver a reproducir

 Anterior



## 4.7

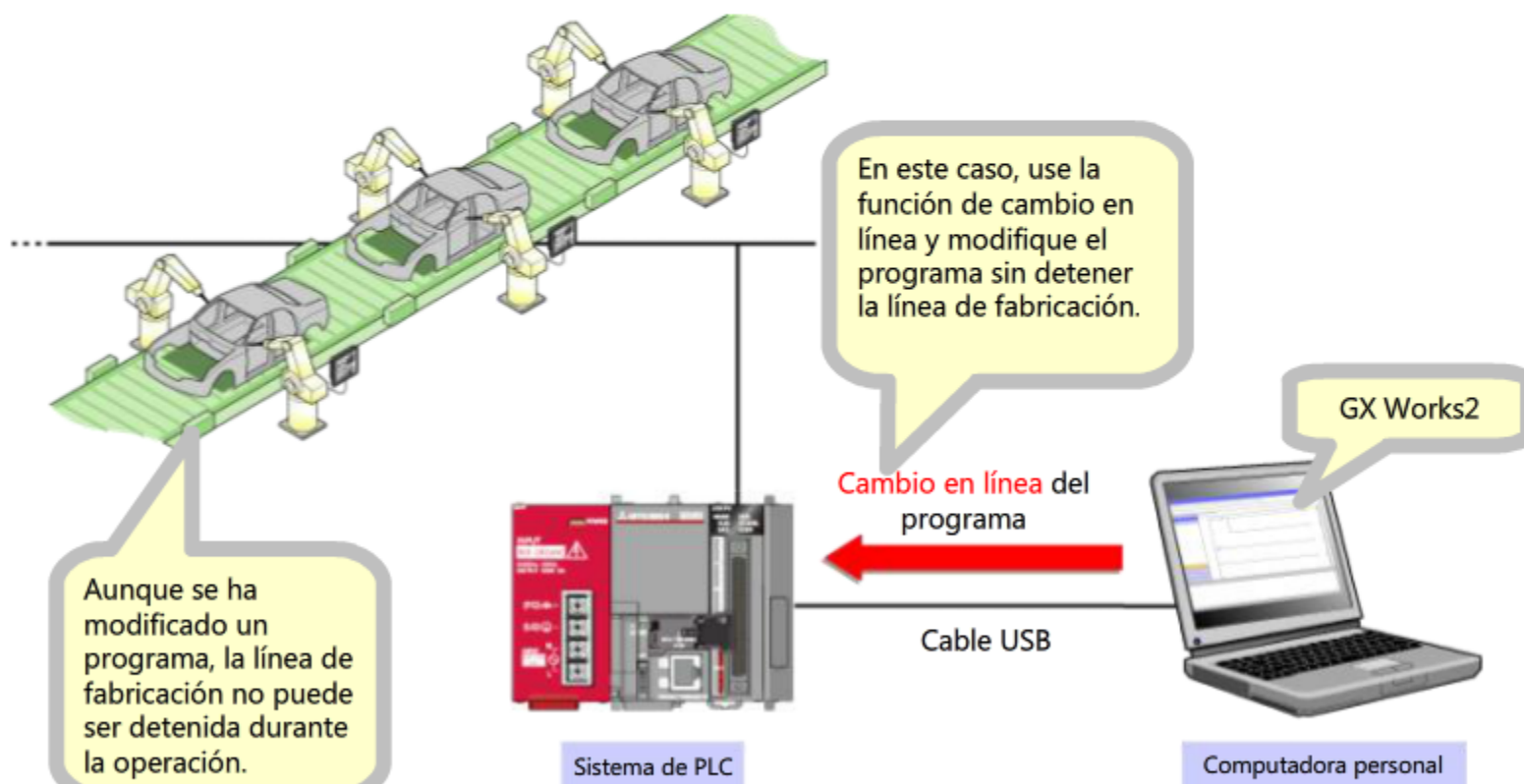
## Operar el sistema de PLC

Luego de completar la verificación de la operación, ejecute el sistema de PLC para iniciar la operación.

### Si el programa necesita ser modificado en el sistema en ejecución

Es posible que se requiera la modificación del programa como la corrección de un error de software o la expansión del sistema luego de iniciar la operación del sistema. Normalmente, se necesita detener el sistema (módulo de CPU) para escribir un programa modificado, pero no siempre es posible. Para resolver este problema, GX Works brinda una función de cambio en línea, la cual se usa para escribir programas sin detener el módulo de CPU en ejecución.

### Ejemplo: Línea de fabricación de automóviles de 24 horas



En la página siguiente, pruebe la función de cambio en línea con la ventana simulada.

## 4.7

## Operar el sistema de PLC



MELSOFT Series GX Works2 C:\SequenceProgram\Learning\Robot\_Control - [[PRG] MAIN]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help



1.000ms

Local Device not Executed

Navigation

Project

- Parameter
- Intelligent Function Module
- Global Device Comment
- Program Setting
- POU
  - Program
    - MAIN
- Local Device Comment
- Device Memory
- Device Initial Value

Project

User Library

Connection Destination

[[PRG] MAIN



El cambio en línea del programa modificado se ha completado.

Haga clic en  para continuar.

Unlabeled

L02

Host Station

2/75Step

NL

**4.8****Conclusión**

Esto completa la explicación básica sobre el diseño de software de controlador programable.

En este curso usted aprendió:

- Los elementos básicos para la programación de un sistema de PLC
- Algunas pautas básicas para el diseño de programas incluyendo el uso de comentarios
- Cómo usar GX works2 para realizar tareas básicas de programación de PLC
- Unas cuantas técnicas usadas para depurar programas de PLC

## Prueba Prueba final

Ahora que ha completado todas las lecciones sobre **PLC Conceptos Básicos de GX Works2**, está listo para tomar la prueba final. Si no tiene claro alguno de los temas cubiertos, tome esta oportunidad para revisar esos temas.

**Hay un total de 5 preguntas (15 áreas) en esta Prueba Final.**

Puede tomar la prueba final las veces que desee.

### Cómo calificar la prueba

Luego de seleccionar la respuesta, asegúrese de hacer clic en el botón **Respuesta**. Su respuesta se perderá si no hace clic en el botón Respuesta. (Se considerará como pregunta sin respuesta.)

### Resultados de la calificación

El número de respuestas correctas, el número de preguntas, el porcentaje de respuestas correctas, y el resultado sobre si aprobó o no aparecerá en la página de calificación.

Respuestas correctas: **2**

Total de preguntas: **9**

Porcentaje: **22%**

Para aprobar la prueba, debe responder correctamente el **60%** de las preguntas.

Continuar

Revisar

Volver a intentar

- Haga clic en el botón **Continuar** para salir de la prueba.
- Haga clic en el botón **Revisar** para revisar la prueba. (Verificar la respuesta correcta)
- Haga clic en el botón **Volver a intentar** para tomar la prueba nuevamente.

## Prueba Prueba Final 1

El programa del cual estaba a cargo lo asumió otra persona, a quien le pareció difícil entender los elementos de control para el programa. ¿Cuál es la contramedida apropiada para evitar este problema?

- Usando la función de comentarios de GX Works2, darle al programa un título y una explicación adecuados.
- Explicar oralmente los elementos de control a la persona nueva.
- Evitar asumir un programa largo y complejo.
- Transferir la tabla de correspondencia para dispositivos E/S y números de dispositivos junto con el programa.

Puntuación

Regresar

# Prueba Prueba Final 2



Complete el procedimiento de programación correcto.

Paso 1 Diseño de programas

Paso 2 ( Q1  )

Paso 3 ( Q2  )

Paso 4 Convertir programas

Paso 5 Guardar proyectos

Paso 6 ( Q3  )

Paso 7 ( Q4  )

Paso 8 Ejecutar el módulo de CPU (RUN)

Paso 9 ( Q5  )

Paso 10 Verificar la operación del sistema de PLC

Puntuación

Regresar

## Prueba Prueba Final 3

Llene los espacios en blanco para completar la explicación de lo que se necesita hacer luego de completar un programa.

Una vez que se ha escrito el programa, se debe probar para asegurar que funcione como lo esperado.

Un(a) () (cuando un código escrito no funciona como es debido) se llama

un(a) () y el proceso para encontrar la causa y corregirlo se llama

()

Este proceso es un paso esencial en la creación de programas.

Puntuación

Regresar



## Prueba Prueba Final 4



Seleccione la aplicación apropiada de cada función de GX Works2.

Función	Aplicación
Simulación	--Select--
Registro/cancelación de entrada salida forzada	--Select--
Cambiar el valor actual	--Select--
Supervisión de escalera	--Select--
Observación	--Select--

Puntuación

Regresar

## Prueba Prueba Final 5

Seleccione la descripción correcta de la función de cambio en línea.

- La función detiene automáticamente el CPU, escribe un programa al CPU, y luego ejecuta automáticamente el CPU.
- La función compara el programa en el módulo de CPU en ejecución con el programa abierto por GX Works2.
- La función puede escribir un programa al módulo de CPU después de detener el módulo de CPU en ejecución con seguridad.
- La función puede escribir un programa al módulo de CPU en ejecución sin detenerlo.

Puntuación

Regresar

## Prueba Calificación de la prueba

Ha completado la Prueba Final. Sus resultados son los siguientes.  
para terminar con la Prueba Final, vaya a la página siguiente.

Respuestas correctas: 0

Total de preguntas: 5

Porcentaje: 0%

Continuar

Revisar

Volver a intentar

**No ha aprobado la prueba.**

Ha completado los **PLC Conceptos Básicos de GX Works2.**

Gracias por tomar este curso.

Esperamos que haya disfrutado las lecciones y que la información recibida en este curso le sea útil en el futuro.

Puede revisar el curso las veces que desee.

**Revisar**

**Cerrar**