

## SIMATIC

### Programar con STEP 7 V5.1

#### Manual

Este manual forma parte del paquete de documentación con la referencia:

**6ES7 810-4CA05-8DA0**

Edición 08/2000  
A5E00069876-03

#### Indicaciones importantes, Contenido

Presentación del producto e instalación	1
Instalación y autorización	2
Planificar una solución de automatización	3
Nociones básicas para diseñar la estructura del programa	4
Cómo arrancar y utilizar STEP 7	5
Elaboración de proyectos	6
Definir símbolos	7
Crear bloques y librerías	8
Crear bloques lógicos	9
Crear bloques de datos	10
Crear fuentes AWL	11
Mostrar datos de referencia	12
Comprobar coherencia del bloque y fecha y hora como propiedad del bloque	13
Configurar mensajes	14
Configurar variables para manejo y visualización	15
Establecer enlaces online y ajustar de la CPU	16
Cargar	17
Test con la tabla de variables	18
Test con el estado del programa	19
Test con el programa de simulación S7-PLSIM (software opcional)	20
Diagnóstico	21
Imprimir y Archivar	22
Configuración multiusuario en una red Windows	23
Trabajar con sistemas de automatización M7	24
Consejos y trucos	25
Anexo	A
Índice alfabético	

## Consignas de seguridad para el usuario

Este manual contiene las informaciones necesarias para la seguridad personal así como para la prevención de daños materiales. Las informaciones están puestas de relieve mediante señales de precaución. Las señales que figuran a continuación representan distintos grados de peligro:



### Peligro

Significa que, si no se adoptan las medidas preventivas adecuadas, **se producirá** la muerte, lesiones corporales graves o daños materiales considerables.



### Precaución

Significa que, si no se adoptan las medidas preventivas adecuadas, **puede producirse** la muerte, lesiones corporales graves o daños materiales considerables.



### Cuidado

Significa que, si no se adoptan las medidas preventivas adecuadas, pueden producirse lesiones corporales o daños materiales.

### Nota

Se trata de una información importante, sobre el producto o sobre una parte determinada del manual, sobre la que se desea llamar particularmente la atención.

## Personal cualificado

La puesta en funcionamiento y el servicio del equipo sólo deben ser llevados a cabo conforme con este manual. Sólo está autorizado a intervenir en este equipo el personal cualificado. En el sentido del manual se trata de personas que disponen de los conocimientos técnicos necesarios para poner en funcionamiento, conectar a tierra y marcar los aparatos, sistemas y circuitos de acuerdo con las normas estándar de seguridad.

## Uso conforme

Considere lo siguiente:



### Precaución

El equipo o los componentes del sistema solo se podrán utilizar para los casos de aplicación previstos en el catálogo y en la descripción técnica, y solo en unión de los equipos y componentes de proveniencia tercera recomendados y homologados por Siemens.

## Marca registrada

SIMATIC®, SIMATIC HMI® y SIMATIC NET® son marcas registradas por SIEMENS AG.

Los restantes nombres y designaciones contenidos en el presente impreso pueden ser marcas registradas cuya utilización por terceros para sus fines pueden violar los derechos de los propietarios.

### Copyright © Siemens AG 1998 All rights reserved

La divulgación y reproducción de este documento, así como el uso y la comunicación de su contenido, no están autorizados, a no ser que se obtenga el consentimiento expreso para ello. Los infractores quedan obligados a la indemnización de los daños. Se reservan todos los derechos, en particular para el caso de concesión de patentes o de modelos de utilidad.

Siemens AG  
Bereich Automatisierungs- und Antriebstechnik (A&D)  
Geschaeftsgebiet Industrie-Automatisierungssysteme (AS)  
Postfach 4848, D- 90327 Nuernberg

### Exención de responsabilidad

Hemos probado el contenido de esta publicación con la concordancia descrita para el hardware y el software. Sin embargo, es posible que se den algunas desviaciones que nos impiden tomar garantía completa de esta concordancia. El contenido de esta publicación está sometido a revisiones regularmente y en caso necesario se incluyen las correcciones en la siguiente edición. Agradecemos sugerencias.

© Siemens AG 1998  
Sujeto a cambios sin previo aviso.

# Indicaciones importantes

## Objetivo del manual

Este manual pretende dar al lector una visión general de cómo programar con **STEP 7**. Entre otras cosas explica cómo instalar y arrancar el software, cómo crear programas y cuáles son los principales componentes de los programas de usuario.

Este manual está dirigido a aquellas personas que realizan soluciones de automatización con STEP 7 para los sistemas de automatización SIMATIC S7.

Recomendamos familiarizarse primero con los ejemplos descritos en el Getting Started de STEP 7 "Introducción y ejercicios prácticos". Estos ejemplos ayudan a iniciarse en el tema de la "Programar con STEP 7".

## Nociones básicas

Se requieren conocimientos básicos en el campo de la automatización

Además es necesario estar familiarizado con el uso de ordenadores o equipos similares a un PC (p. ej. unidades de programación) que funcionen en el entorno Windows 95/98/2000 o Windows NT.

## Objeto del manual

El software en el que se basan las indicaciones del manual es STEP 7 V5.1.

## Paquetes de documentación de STEP 7

Este manual forma parte del paquete de documentación "STEP 7 Información básica".

La tabla siguiente da una visión de conjunto de la documentación de STEP 7:

Manuales	Tema	Referencia
Información básica de STEP 7 compuesta por: <ul style="list-style-type: none"> <li>STEP 7 V5.1: Introducción y ejercicios prácticos, Getting Started</li> <li>Programar con STEP 7 V5.1</li> <li>Configurar el hardware y la comunicación con STEP 7 V5.1</li> <li>De S5 a S7, Guía para facilitar la transición</li> </ul>	Nociones básicas para el personal técnico. Describe cómo realizar soluciones de control con el software STEP 7 para los sistemas S7-300/400.	6ES7810-4CA05-8DA0
Información de referencia para STEP 7, compuesta por <ul style="list-style-type: none"> <li>Manuales KOP/FUP/AWL para S7-300/400</li> <li>Funciones estándar y funciones de sistema para S7-300/400</li> </ul>	Esta obra de consulta describe los lenguajes de programación KOP, FUP y AWL así como las funciones estándar y las funciones de sistema como complemento a la 'Información básica de STEP'.	6ES7810-4CA05-8DR0

Ayudas en pantalla	Tema	Referencia
Ayuda de STEP 7	Nociones básicas para diseñar programas y configurar el hardware con STEP 7. Disponible en forma de Ayuda en pantalla.	Componente del paquete de software STEP 7
Ayudas de referencia para AWL/KOP/FUP Ayudas de referencia para SFBs/SFCs Ayudas de referencia para los bloques de organización	Información de referencia sensible al contexto	Componente del paquete de software STEP 7

## Ayuda en pantalla

Como complemento al manual puede recurrir a la Ayuda en pantalla integrada en el software.

A la Ayuda que está integrada en el software se accede de distinta manera:

- El menú **Ayuda** ofrece varios comandos de menú: **Temas de Ayuda** abre el índice de la Ayuda de STEP 7.
- **Uso de la Ayuda** explica detalladamente cómo utilizar la Ayuda en pantalla.
- La Ayuda sensible al contexto ofrece información sobre el contexto actual, p. ej. sobre el cuadro de diálogo que esté abierto o sobre la ventana activa. Para acceder a esta ayuda pulse el botón de comando "Ayuda" o bien la tecla F1.
- También la barra de estado ofrece ayuda sensible al contexto. Si sitúa el puntero del ratón en un comando de menú aparecerá una breve explicación sobre dicho comando.
- Situando el puntero del ratón unos segundos sobre uno de los botones de la barra de herramientas aparecerá una breve explicación al respecto.

Si prefiere leer los textos de la Ayuda en papel, puede imprimir los temas o libros que le interesen o bien imprimir toda la Ayuda.

Este manual es un extracto de la Ayuda de STEP 7 basada en HTML. Si necesita instrucciones más detalladas, consulte la ayuda de STEP 7. Debido a que la estructura del manual se corresponde a grandes rasgos con la de la Ayuda en pantalla puede alternar la lectura del manual con la de la Ayuda en pantalla.

## Sugerencias para mejorar la documentación

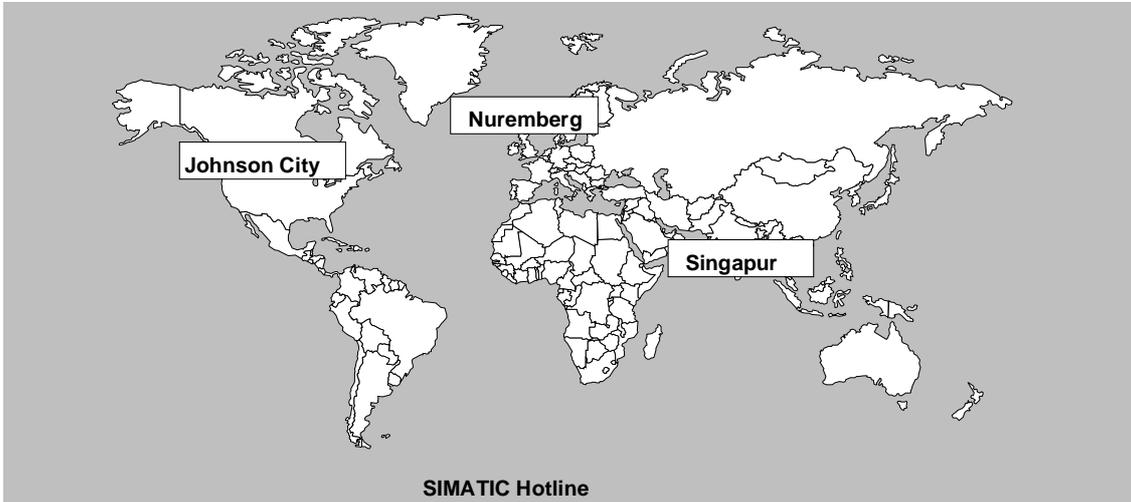
Para poder ofrecerle a Ud. y a los futuros usuarios una documentación de calidad necesitamos su colaboración. Si desea hacer sugerencias relacionadas con el presente *manual* o con la *Ayuda en pantalla* rogamos rellene el formulario que encontrará en las últimas páginas del manual y remitirlo a la dirección indicada. Le agradeceríamos asimismo que nos comunique la opinión que le merece el manual.

## Centro de entrenamiento SIMATIC

Para iniciarse en el mundo de los sistemas de automatización SIMATIC S7, ofrecemos distintos cursillos. Si tiene interés póngase en contacto con su centro de entrenamiento regional o con el centro principal en D 90327 Nürnberg.  
Teléfono: +49 (911) 895-3200.

## Servicio de Asistencia al Cliente SIMATIC (Hotline)

Estamos a su disposición en cualquier lugar del mundo las 24 horas del día:



### Worldwide (Nuremberg) Technical Support

(FreeContact)

Hora: lunes a viernes 7:00 a 17:00  
Teléfono: +49 (180) 5050-222  
Fax: +49 (180) 5050-223  
E-Mail: techsupport@  
ad.siemens.de  
GMT: +1:00

### Worldwide (Nuremberg) Technical Support

(a cargo del cliente, sólo con  
SIMATIC Card)

Hora: lunes a viernes 0:00 a 24:00  
Teléfono: +49 (911) 895-7777  
Fax: +49 (911) 895-7001  
GMT: +01:00

### Europe / Africa (Nuremberg) Authorization

Hora: lunes a viernes 7:00 a 17:00  
Teléfono: +49 (911) 895-7200  
Fax: +49 (911) 895-7201  
E-Mail: authorization@  
nbgm.siemens.de  
GMT: +1:00

### America (Johnson City) Technical Support and Authorization

Hora: lunes a viernes 8:00 a 19:00  
Teléfono: +1 423 461-2522  
Fax: +1 423 461-2289  
E-Mail: simatic.hotline@  
sea.siemens.com  
GMT: -5:00

### Asia / Australia (Singapur) Technical Support and Authorization

Hora: lunes a viernes 8:30 a 17:30  
Teléfono: +65 740-7000  
Fax: +65 740-7001  
E-Mail: simatic.hotline@  
sae.siemens.com.sg  
GMT: +8:00

El personal que atiende las hotlines SIMATIC habla por regla general alemán e inglés; el de la hotline para cuestiones relacionadas con las autorizaciones también habla español, francés e italiano.

## Servicios online del SIMATIC Customer Support

La línea directa SIMATIC Customer Support le ofrece además una información completa acerca de los productos SIMATIC:

- Para recibir información general de actualidad consulte
  - en **Internet** bajo <http://www.ad.siemens.de/simatic>
- Si requiere las informaciones del producto y los downloads más actuales consulte:
  - en **Internet** bajo <http://www.ad.siemens.de/simatic-cs>
  - vía **Bulletin Board System** (BBS) en Nuremberg (SIMATIC Customer Support Mailbox) marcando el número +49 (911) 895-7100.

Para acceder al mailbox utilice un módem de hasta V.34 (28,8 kbit/s), ajustando sus parámetros como sigue: 8, N, 1, ANSI, o bien acceda por RDSI (x.75, 64 kbits).

- Consulte la persona de contacto para Automation & Drives de su localidad en nuestra base de datos:
  - en **Internet** bajo <http://www3.ad.siemens.de/partner/search.asp?lang=en>



# Contenido

<b>1</b>	<b>Presentación del producto e instalación</b>	<b>1-1</b>
1.1	Guía de orientación de STEP 7.....	1-1
1.2	El software estándar STEP 7.....	1-5
1.3	Novedades de la versión 5.1 de STEP 7.....	1-9
1.4	Posibilidades de ampliar el software estándar STEP 7.....	1-13
1.1.1	Posibilidades de ampliar el software estándar STEP 7.....	1-13
1.1.2	Herramientas de ingeniería.....	1-14
1.1.3	Software Runtime.....	1-15
1.1.4	Interfaces hombre-máquina.....	1-16
<b>2</b>	<b>Instalación y autorización</b>	<b>2-1</b>
2.1	Autorización.....	2-1
2.1.1	Autorización.....	2-1
2.1.2	Instalar y desinstalar la autorización.....	2-1
2.1.3	Reglas para el empleo de autorizaciones.....	2-4
2.2	Instalar STEP 7.....	2-7
2.2.1	Instalar STEP 7.....	2-7
2.2.2	Procedimiento de instalación.....	2-8
2.2.3	Ajustar el interface PG/PC.....	2-11
2.3	Desinstalar STEP 7.....	2-13
2.3.1	Desinstalar STEP 7.....	2-13
<b>3</b>	<b>Planificar una solución de automatización</b>	<b>3-1</b>
3.1	Procedimiento básico para planificar una solución de automatización.....	3-1
3.2	Subdividir el proceso en tareas y áreas.....	3-2
3.3	Describir el funcionamiento en diversas áreas.....	3-4
3.4	Listado de entradas y salidas.....	3-6
3.5	Crear un diagrama de E/S para los motores.....	3-6
3.6	Crear un diagrama de E/S para las válvulas.....	3-7
3.7	Definir los requerimientos de seguridad.....	3-8
3.8	Describir los elementos necesarios para manejo y visualización.....	3-9
3.9	Crear un esquema de configuración.....	3-10
<b>4</b>	<b>Nociones básicas para diseñar la estructura del programa</b>	<b>4-1</b>
4.1	Programas de una CPU.....	4-1
4.2	Bloques del programa de usuario.....	4-2
4.2.1	Bloques del programa de usuario.....	4-2
4.2.2	Bloques de organización y estructura del programa.....	4-3
4.2.3	Jerarquía de llamada en el programa de usuario.....	4-8
4.2.4	Tipos de bloques y ejecución cíclica del programa.....	4-10
4.2.5	Bloques de organización para la ejecución controlada por alarmas.....	4-22

<b>5</b>	<b>Cómo arrancar y utilizar STEP 7</b>	<b>5-1</b>
5.1	Cómo arrancar y utilizar STEP 7.....	5-1
5.1.1	Cómo iniciar STEP 7 .....	5-1
5.1.2	Arrancar STEP 7 con los parámetros de arranque estándar.....	5-2
5.1.3	Cómo acceder a las funciones de ayuda.....	5-3
5.2	Objetos y su jerarquía.....	5-5
5.2.1	Objetos y su jerarquía.....	5-5
5.2.2	Objeto "Proyecto" .....	5-6
5.2.3	Carpeta de objetos "Librería" .....	5-7
5.2.4	Objeto "Equipo" .....	5-8
5.2.5	Objeto "Módulo programable" .....	5-9
5.2.6	Objeto "Programa S7/M7" .....	5-11
5.2.7	Objeto "Carpeta de bloques".....	5-12
5.2.8	Objeto "Carpeta de fuentes".....	5-14
5.2.9	Programa S7/M7 sin equipo ni CPU.....	5-15
5.3	Interface de usuario .....	5-17
5.3.1	Manejo orientado a objetos.....	5-17
5.3.2	Estructura de la ventana .....	5-17
5.3.3	Elementos de los cuadros de diálogo.....	5-18
5.3.4	Crear y manejar objetos.....	5-19
5.3.5	Elegir objetos en cuadros de diálogo.....	5-24
5.3.6	Memoria de sesión .....	5-25
5.3.7	Cambiar la organización de las ventanas de la tabla de símbolos .....	5-25
5.3.8	Guardar y restablecer la organización de las ventanas.....	5-26
5.4	Manejo de las teclas .....	5-27
5.4.1	Manejo de las teclas .....	5-27
5.4.2	Combinaciones de teclas para comandos de menú.....	5-27
5.4.3	Combinaciones de teclas para desplazar el cursor.....	5-29
5.4.4	Combinaciones de teclas para seleccionar textos .....	5-30
5.4.5	Combinaciones de teclas para acceder a la Ayuda en pantalla .....	5-30
5.4.6	Combinaciones de teclas para cambiar de una ventana a otra.....	5-31
<b>6</b>	<b>Elaboración de proyectos</b>	<b>6-1</b>
6.1	Estructura de los proyectos.....	6-1
6.2	Crear proyectos .....	6-3
6.2.1	Crear un proyecto .....	6-3
6.2.2	Insertar equipos.....	6-5
6.2.3	Insertar un programa S7 o M7 .....	6-6
6.3	Elaborar un proyecto .....	6-8
6.3.1	Elaborar un proyecto .....	6-8
6.3.2	Gestionar textos en varios idiomas .....	6-9
<b>7</b>	<b>Definir símbolos</b>	<b>7-1</b>
7.1	Direccionamiento absoluto y simbólico.....	7-1
7.2	Símbolos globales y locales.....	7-3
7.3	Representación de símbolos globales y locales .....	7-4
7.4	Ajustar la preferencia de operandos.....	7-5
7.5	Tabla de símbolos para los símbolos globales .....	7-6
7.5.1	Tabla de símbolos para los símbolos globales .....	7-6
7.5.2	Estructura y componentes de la tabla de símbolos.....	7-6
7.5.3	Direcciones y tipos de datos admisibles en la tabla de símbolos .....	7-8
7.5.4	Símbolos incompletos y ambiguos en la tabla de símbolos .....	7-9
7.6	Métodos para introducir símbolos globales .....	7-10
7.6.1	Métodos para introducir símbolos globales .....	7-10
7.6.2	Observaciones generales para introducir símbolos .....	7-10
7.6.3	Introducir símbolos globales en un cuadro de diálogo .....	7-11
7.6.4	Introducir símbolos globales en la tabla de símbolos.....	7-12
7.6.5	Mayúsculas y minúsculas en los símbolos .....	7-13
7.6.6	Exportar e importar tablas de símbolos .....	7-15

7.6.7	Formatos de archivos para importar/exportar una tabla de símbolos .....	7-15
<b>8</b>	<b>Crear bloques y librerías</b>	<b>8-1</b>
8.1	Seleccionar el método de creación .....	8-1
8.2	Elegir el lenguaje de programación .....	8-2
8.2.1	Elegir el lenguaje de programación .....	8-2
8.2.2	Lenguaje de programación KOP (esquema de contactos) .....	8-4
8.2.3	Lenguaje de programación FUP (diagrama de funciones) .....	8-4
8.2.4	Lenguaje de programación AWL (lista de instrucciones) .....	8-5
8.2.5	Lenguaje de programación S7-SCL .....	8-6
8.2.6	Lenguaje de programación S7-GRAPH (control secuencial) .....	8-7
8.2.7	Lenguaje de programación S7-HiGraph (grafo de estado) .....	8-8
8.2.8	Lenguaje de programación S7-CFC .....	8-9
8.3	Crear bloques .....	8-10
8.3.1	Carpeta de bloques .....	8-10
8.3.2	Tipos de datos de usuario (UDT) .....	8-11
8.3.3	Propiedades del bloque .....	8-11
8.3.4	Visualizar longitudes de bloques .....	8-13
8.3.5	Recablear .....	8-14
8.3.6	Atributos de bloques y parámetros .....	8-14
8.4	Trabajar con librerías .....	8-15
8.4.1	Trabajar con librerías .....	8-15
8.4.2	Estructura jerárquica de las librerías .....	8-16
8.4.3	Librerías estándar disponibles .....	8-17
<b>9</b>	<b>Crear bloques lógicos</b>	<b>9-1</b>
9.1	Nociones básicas para crear bloques lógicos .....	9-1
9.1.1	Procedimiento básico para crear bloques lógicos .....	9-1
9.1.2	Preajustes para el editor de programas KOP/FUP/AWL .....	9-2
9.1.3	Derechos de acceso a bloques y fuentes .....	9-3
9.1.4	Instrucciones del catálogo de elementos de programa .....	9-3
9.2	Editar la tabla de declaración de variables .....	9-4
9.2.1	Utilización de la declaración de variables en los bloques lógicos .....	9-4
9.2.2	Interacción entre la tabla de declaración de variables y el área de instrucciones .....	9-5
9.2.3	Estructura de la tabla de declaración de variables .....	9-5
9.2.4	Observaciones generales sobre las tablas de declaración de variables .....	9-7
9.3	Multiinstancias en la tabla de declaración de variables .....	9-8
9.3.1	Utilización de multiinstancias .....	9-8
9.3.2	Reglas para declarar multiinstancias .....	9-9
9.3.3	Introducir la multiinstancia en la tabla de declaración de variables .....	9-9
9.4	Indicaciones generales para introducir instrucciones y comentarios .....	9-10
9.4.1	Estructura del área de instrucciones .....	9-10
9.4.2	Procedimiento para introducir instrucciones .....	9-10
9.4.3	Introducir símbolos globales en un programa .....	9-12
9.4.4	Títulos y comentarios de bloques y segmentos .....	9-12
9.4.5	Función de búsqueda de errores en el área de instrucciones .....	9-13
9.5	Editar instrucciones KOP en el área de instrucciones .....	9-14
9.5.1	Ajustes para el lenguaje de programación KOP .....	9-14
9.5.2	Reglas para introducir operaciones KOP .....	9-14
9.5.3	Interconexiones no admisibles en KOP .....	9-16
9.6	Editar instrucciones FUP en el área de instrucciones .....	9-17
9.6.1	Ajustes para el lenguaje de programación FUP .....	9-17
9.6.2	Reglas para introducir operaciones FUP .....	9-17
9.7	Editar instrucciones AWL en el área de instrucciones .....	9-20
9.7.1	Ajustes para el lenguaje de programación AWL .....	9-20
9.7.2	Reglas para introducir instrucciones AWL .....	9-20
9.8	Actualizar llamadas a bloques .....	9-21
9.8.1	Actualizar llamadas a bloques .....	9-21
9.9	Guardar bloques lógicos .....	9-22

9.9.1	Guardar bloques lógicos .....	9-22
9.9.2	Corregir los interfaces en una FC, un FB o un UDT .....	9-22
9.9.3	Evitar errores al llamar bloques.....	9-23
<b>10</b>	<b>Crear bloques de datos</b>	<b>10-1</b>
10.1	Nociones básicas para crear bloques de datos .....	10-1
10.2	Ver declaración de bloques de datos .....	10-2
10.3	Ver datos de los bloques de datos .....	10-3
10.4	Editar bloques de datos y guardar.....	10-4
10.4.1	Introducir la estructura de los bloques de datos globales.....	10-4
10.4.2	Introducir / visualizar la estructura de bloques de datos con FB asociado (DBs de instancia) .....	10-4
10.4.3	Introducir la estructura de los tipos de datos de usuario (UDT).....	10-6
10.4.4	Introducir / visualizar la estructura de bloques de datos con UDT asociado.....	10-6
10.4.5	Cambiar valores de datos en la vista "Datos" .....	10-7
10.4.6	Inicializar los valores de datos .....	10-8
10.4.7	Guardar bloques de datos.....	10-8
<b>11</b>	<b>Crear fuentes AWL</b>	<b>11-1</b>
11.1	Nociones básicas para programar en fuentes AWL.....	11-1
11.2	Reglas para introducir instrucciones en fuentes AWL.....	11-2
11.2.1	Reglas para introducir instrucciones en fuentes AWL.....	11-2
11.2.2	Reglas para declarar variables en fuentes AWL.....	11-3
11.2.3	Orden de los bloques en fuentes AWL.....	11-4
11.2.4	Reglas para definir los atributos de sistema en fuentes AWL .....	11-4
11.2.5	Reglas para definir las propiedades de bloques en fuentes AWL .....	11-5
11.2.6	Propiedades válidas para los diferentes tipos de bloques.....	11-6
11.3	Estructura de bloques en fuentes AWL .....	11-7
11.3.1	Estructura de bloques en fuentes AWL .....	11-7
11.3.2	Estructura de los bloques lógicos en fuentes AWL .....	11-7
11.3.3	Estructura de bloques de datos en fuentes AWL.....	11-8
11.3.4	Estructura de tipos de datos de usuario en fuentes AWL.....	11-8
11.4	Sintaxis y formatos de bloques en fuentes AWL.....	11-9
11.4.1	Sintaxis y formatos de bloques en fuentes AWL.....	11-9
11.4.2	Formato de los OBs.....	11-9
11.4.3	Formato de los FBs .....	11-10
11.4.4	Formato de las FCs .....	11-11
11.4.5	Formato de los DBs .....	11-12
11.5	Crear fuentes AWL .....	11-13
11.5.1	Crear fuentes AWL .....	11-13
11.5.2	Editar fuentes S7 .....	11-14
11.5.3	Insertar plantillas de bloques en fuentes AWL.....	11-14
11.5.4	Insertar fuentes externas .....	11-14
11.5.5	Generar fuentes AWL de bloques .....	11-15
11.6	Guardar y compilar fuentes AWL y comprobar su coherencia .....	11-16
11.6.1	Guardar fuentes AWL .....	11-16
11.6.2	Comprobar la coherencia en fuentes AWL.....	11-16
11.6.3	Buscar errores en fuentes AWL .....	11-16
11.6.4	Compilar fuentes AWL.....	11-17
11.7	Ejemplos de fuentes AWL.....	11-18
11.7.1	Ejemplos de declaración de variables en fuentes AWL .....	11-18
11.7.2	Ejemplo de OBs en fuentes AWL.....	11-19
11.7.3	Ejemplo de FCs en fuentes AWL .....	11-20
11.7.4	Ejemplo de FBs en fuentes AWL .....	11-22
11.7.5	Ejemplos de DBs en fuentes AWL .....	11-23
11.7.6	Ejemplo de UDTs en fuentes AWL.....	11-24

<b>12</b>	<b>Mostrar datos de referencia</b>	<b>12-1</b>
12.1	Sinopsis de los datos de referencia.....	12-1
12.1.1	Sinopsis de los datos de referencia.....	12-1
12.1.2	Lista de referencias cruzadas .....	12-2
12.1.3	Estructura del programa .....	12-3
12.1.4	Plano de ocupación para entradas, salidas y marcas (E/A/M) .....	12-5
12.1.5	Plano de ocupación para temporizadores y contadores (T/Z).....	12-7
12.1.6	Símbolos no utilizados .....	12-8
12.1.7	Operandos sin símbolo .....	12-8
12.1.8	Visualizar la información del bloque en KOP, FUP, AWL.....	12-9
12.2	Trabajar con datos de referencia .....	12-10
12.2.1	Métodos de visualización de los datos de referencia.....	12-10
12.2.2	Visualizar listas en ventanas de trabajo adicionales .....	12-11
12.2.3	Crear y visualizar datos de referencia .....	12-11
12.2.4	Posicionamiento rápido en los puntos de aplicación del programa .....	12-12
12.2.5	Ejemplo para trabajar con puntos de aplicación .....	12-13
<b>13</b>	<b>Comprobar coherencia del bloque y fecha y hora como propiedad del bloque</b>	<b>13-1</b>
13.1	Comprobar coherencia de bloques .....	13-1
13.2	Fecha y hora como propiedad del bloque y conflictos de fecha y hora .....	13-3
13.3	Indicación de fecha y hora de bloques lógicos .....	13-4
13.4	Indicación de fecha y hora de bloques de datos globales .....	13-5
13.5	Indicación de fecha y hora de bloques de datos de instancia .....	13-5
13.6	Indicación de fecha y hora de UDTs y DBs derivados de UDTs.....	13-6
<b>14</b>	<b>Configurar mensajes</b>	<b>14-1</b>
14.1	El sistema de mensajes .....	14-1
14.1.1	El sistema de notificación .....	14-1
14.1.2	¿Qué procedimientos existen para crear mensajes? .....	14-1
14.1.3	Selección del procedimiento de notificación .....	14-2
14.1.4	Componentes SIMATIC .....	14-4
14.1.5	Partes de un mensaje .....	14-5
14.1.6	Asignar números a los mensajes .....	14-5
14.2	Asignar y editar mensajes de bloque.....	14-6
14.2.1	Asignar y editar mensajes de bloque.....	14-6
14.2.2	¿Qué bloques de notificación existen?.....	14-6
14.2.3	Parámetros formales, atributos del sistema y bloques de notificación.....	14-7
14.2.4	Plantilla y mensajes .....	14-8
14.2.5	Crear mensajes de bloque .....	14-9
14.2.6	Configuración de mensajes PCS7.....	14-12
14.3	Asignar y editar mensajes de símbolos .....	14-14
14.3.1	Asignar y editar mensajes de símbolos .....	14-14
14.4	Crear y editar mensajes de diagnóstico personalizados .....	14-15
14.4.1	Crear y editar mensajes de diagnóstico personalizados .....	14-15
14.5	Traducir y editar textos de usuario .....	14-16
14.5.1	Traducir y editar textos de usuario .....	14-16
14.5.2	Traducir y editar textos de usuario .....	14-16
14.5.3	Traducir librerías de textos.....	14-18
14.6	Transferir los datos de configuración al sistema de destino .....	14-19
14.6.1	Transferir los datos de configuración al sistema de destino .....	14-19
14.7	Visualizar mensajes de CPU y mensajes de diagnóstico personalizados .....	14-20
14.7.1	Visualizar mensajes de CPU y mensajes de diagnóstico personalizados .....	14-20
14.7.2	Configurar mensajes de CPU.....	14-22
14.7.3	Visualizar los mensajes de CPU guardados .....	14-22
14.8	Configurar 'Notificación de errores del sistema' .....	14-23
14.8.1	Configurar la 'notificación de errores del sistema' .....	14-23
14.8.2	Componentes compatibles y funcionalidad .....	14-25
14.8.3	Ajustes para notificar errores del sistema.....	14-27
14.8.4	OBs de error generados.....	14-28

14.8.5	FB, DB generados .....	14-29
<b>15</b>	<b>Configurar variables para manejo y visualización</b>	<b>15-1</b>
15.1	Configurar variables para manejo y visualización .....	15-1
15.2	Configurar atributos M+V con AWL, KOP y FUP .....	15-3
15.2.1	Configurar atributos M+V con AWL, KOP y FUP .....	15-3
15.3	Configurar atributos M+V a través de la tabla de símbolos .....	15-4
15.3.1	Configurar atributos M+V a través de la tabla de símbolos .....	15-4
15.4	Cambiar atributos M+V con CFC .....	15-5
15.4.1	Cambiar atributos M+V con CFC .....	15-5
15.5	Transferir los datos de configuración al sistema de destino M+V .....	15-6
15.5.1	Transferir los datos de configuración al sistema de destino M+V .....	15-6
<b>16</b>	<b>Establecer enlaces online y ajustar la CPU</b>	<b>16-1</b>
16.1	Establecer enlaces online .....	16-1
16.1.1	Establecer enlaces online .....	16-1
16.1.2	Establecer un enlace online desde la ventana "Estaciones accesibles" .....	16-1
16.1.3	Establecer un enlace online desde la ventana online del proyecto .....	16-2
16.1.4	Protección con contraseña para acceder a sistemas de destino .....	16-3
16.1.5	Nota respecto a la actualización del contenido de ventanas .....	16-3
16.2	Visualizar y cambiar el estado operativo .....	16-4
16.2.1	Visualizar y cambiar el estado operativo .....	16-4
16.3	Visualizar y ajustar la fecha y la hora .....	16-5
16.3.1	Visualizar y ajustar la fecha y la hora .....	16-5
<b>17</b>	<b>Cargar</b>	<b>17-1</b>
17.1	Cargar en el sistema de destino desde la PG .....	17-1
17.1.1	Requisitos para cargar .....	17-1
17.1.2	Diferencia entre guardar y cargar bloques .....	17-2
17.1.3	Memorias de carga y de trabajo de la CPU .....	17-2
17.1.4	Posibilidades de cargar en función de la memoria de carga .....	17-4
17.1.5	Recargar bloques en el sistema de destino .....	17-5
17.1.6	Cargar mediante Memory Cards EPROM .....	17-6
17.1.7	Cargar la configuración del hardware y la de los enlaces por separado .....	17-7
17.2	Cargar en la PG desde el sistema de destino .....	17-12
17.2.1	Cargar en la PG desde el sistema de destino .....	17-12
17.2.2	Cargar un equipo en la PG .....	17-14
17.2.3	Cargar bloques desde una CPU S7 .....	17-14
17.2.4	Editar los bloques cargados en la PG/en el PC .....	17-15
17.2.5	Cargar nuevamente la configuración del hardware y la de los equipos .....	17-16
17.2.6	Borrar en el sistema de destino .....	17-18
17.2.7	Comprimir la memoria de usuario (RAM) .....	17-19
<b>18</b>	<b>Test con la tabla de variables</b>	<b>18-1</b>
18.1	Introducción al test con tablas de variables .....	18-1
18.2	Procedimiento básico para observar y forzar con tablas de variables .....	18-2
18.3	Editar y guardar tablas de variables .....	18-2
18.3.1	Crear y abrir una tabla de variables .....	18-2
18.3.2	Copiar/mover tablas de variables .....	18-3
18.3.3	Guardar una tabla de variables .....	18-3
18.4	Introducir variables en tablas de variables .....	18-4
18.4.1	Insertar operandos o símbolos en una tabla de variables .....	18-4
18.4.2	Insertar valores de forzado .....	18-5
18.4.3	Valores máximos para introducir temporizadores .....	18-5
18.4.4	Valores máximos para introducir contadores .....	18-6
18.4.5	Introducir líneas de comentario .....	18-7
18.5	Ejemplos .....	18-8
18.5.1	Ejemplo de introducción de operandos en tablas de variables .....	18-8
18.5.2	Ejemplo de introducción de un área de operandos conexos .....	18-9

18.5.3	Ejemplos de introducción de valores de forzado normal y de forzado permanente	18-9
18.6	Establecer un enlace con la CPU.....	18-12
18.6.1	Establecer un enlace con la CPU.....	18-12
18.7	Observar variables.....	18-13
18.7.1	Introducción a la observación de variables.....	18-13
18.7.2	Ajustar el disparo para observar variables.....	18-13
18.8	Forzar variables.....	18-15
18.8.1	Introducción al forzado de variables.....	18-15
18.8.2	Ajustar las condiciones de disparo para forzar variables .....	18-16
18.9	Forzado permanente de variables.....	18-18
18.9.1	Introducción al forzado permanente de variables .....	18-18
18.9.2	Reglas de seguridad para el forzado permanente de variables .....	18-20
18.9.3	Diferencias entre el forzado normal y el forzado permanente .....	18-21
<b>19</b>	<b>Test con el estado del programa</b>	<b>19-1</b>
19.1	Test con el estado del programa.....	19-1
19.2	Visualización del estado de programas .....	19-3
19.3	Información importante para comprobar en modo Etapa individual / Puntos de parada .....	19-5
19.4	Información importante sobre el estado operativo PARADA.....	19-7
19.5	Observar el estado de bloques de datos .....	19-8
19.6	Definir el entorno de llamada del bloque .....	19-9
<b>20</b>	<b>Test con el programa de simulación S7-PLCSIM (software opcional)</b>	<b>20-1</b>
20.1	Test con el programa de simulación (software opcional) .....	20-1
<b>21</b>	<b>Diagnóstico</b>	<b>21-1</b>
21.1	Diagnóstico del hardware y búsqueda de errores.....	21-1
21.2	Símbolos de diagnóstico en la vista online .....	21-3
21.3	Diagnosticar el hardware: Vista rápida.....	21-5
21.3.1	Acceder a la vista rápida.....	21-5
21.3.2	Funciones de información de la vista rápida.....	21-5
21.4	Diagnosticar el hardware: Vista del diagnóstico .....	21-6
21.4.1	Acceder a la vista de diagnóstico de HW Config .....	21-6
21.4.2	Funciones de información de la vista de diagnóstico.....	21-8
21.5	Información del módulo.....	21-9
21.5.2	Funciones de la información del módulo .....	21-9
21.5.3	Volumen de información del módulo en función del tipo de módulo.....	21-12
21.6	Diagnóstico en el estado operativo STOP .....	21-14
21.6.1	Procedimiento básico para averiguar la causa de un STOP .....	21-14
21.6.2	Contenido de las pilas en estado operativo STOP.....	21-14
21.7	Control de los tiempos de ciclo para evitar errores de tiempo.....	21-16
21.7.1	Control de los tiempos de ciclo para evitar errores de tiempo.....	21-16
21.8	Transmisión de informaciones de diagnóstico .....	21-17
21.8.1	Transmisión de informaciones de diagnóstico .....	21-17
21.8.2	Lista de estado del sistema (SZL).....	21-18
21.8.3	Enviar mensajes de diagnóstico personalizados .....	21-20
21.8.4	Funciones de diagnóstico .....	21-21
21.9	Medidas en el programa para tratar fallos .....	21-23
21.9.1	Medidas en el programa para tratar fallos .....	21-23
21.9.2	Evaluar el parámetro de salida RET_VAL .....	21-24
21.9.3	OBs de error para reaccionar a errores detectados.....	21-25
21.9.4	Insertar valores de sustitución al detectar errores .....	21-28
21.9.5	Error de redundancia en periferia (OB 70).....	21-31
21.9.6	Error de redundancia en CPU (OB 72).....	21-31
21.9.7	Error de redundancia de comunicación OB 73 .....	21-32
21.9.8	Error de tiempo (OB 80).....	21-33
21.9.9	Fallo de alimentación (OB 81).....	21-33
21.9.10	Alarma de diagnóstico (OB 82) .....	21-34

21.9.11	Alarma de presencia de módulo (OB 83).....	21-35
21.9.12	Fallo de CPU (OB 84).....	21-36
21.9.13	Error de ejecución del programa (OB 85).....	21-36
21.9.14	Fallo en el bastidor (OB 86).....	21-37
21.9.15	Error de comunicación (OB 87).....	21-37
21.9.16	Error de programación (OB 121).....	21-38
21.9.17	Error de acceso a la periferia (OB 122).....	21-39
<b>22</b>	<b>Imprimir y archivar</b>	<b>22-1</b>
22.1	Imprimir la documentación de un proyecto.....	22-1
22.1.1	Imprimir la documentación de un proyecto.....	22-1
22.1.2	Procedimiento básico para imprimir.....	22-2
22.1.3	Funciones para imprimir.....	22-2
22.1.4	Particularidades al imprimir el árbol de objetos.....	22-4
22.2	Archivar proyectos y librerías.....	22-5
22.2.1	Archivar proyectos y librerías.....	22-5
22.2.2	Utilización de las funciones 'Guardar / Archivar'.....	22-5
22.2.3	Requisitos para poder archivar.....	22-6
22.2.4	Procedimiento para archivar y desarchivar.....	22-6
<b>23</b>	<b>Configuración multiusuario en una red Windows</b>	<b>23-1</b>
23.1	Configuración multiusuario en una red Windows.....	23-1
<b>24</b>	<b>Trabajar con sistemas de automatización M7</b>	<b>24-1</b>
24.1	Procedimiento para sistemas M7.....	24-1
24.2	Software opcional para la programación M7.....	24-3
24.3	Sistemas operativos para M7-300/400.....	24-6
<b>25</b>	<b>Consejos y trucos</b>	<b>25-1</b>
25.1	Intercambiar módulos en la tabla de configuración.....	25-1
25.2	Proyectos con un gran número de equipos en red.....	25-1
25.3	Reorganización.....	25-2
25.4	Test con la tabla de variables.....	25-2
25.5	Memoria de trabajo virtual.....	25-4

<b>A</b>	<b>Anexos</b>	<b>A-1</b>
A.1	Estados operativos .....	A-1
A.1.1	Estados operativos y cambios de estado .....	A-1
A.1.2	Estado operativo STOP .....	A-4
A.1.3	Estado operativo ARRANQUE .....	A-5
A.1.4	Estado operativo RUN .....	A-11
A.1.5	Estado operativo PARADA .....	A-12
A.2	Áreas de memoria de las CPUs S7 .....	A-13
A.2.1	Subdivisión de la memoria en áreas.....	A-13
A.2.2	Memorias de carga y de trabajo.....	A-14
A.2.3	Memoria de sistema .....	A-16
A.3	Tipos de datos y de parámetros.....	A-28
A.3.1	Introducción a los tipos de datos y de parámetros.....	A-28
A.3.2	Tipos de datos simples .....	A-29
A.3.3	Tipos de datos compuestos .....	A-37
A.3.4	Tipos de parámetros.....	A-47
A.4	Trabajar con proyectos de versiones anteriores .....	A-66
A.4.1	Convertir proyectos de la versión 1 .....	A-66
A.4.2	Convertir proyectos de la versión 2.....	A-67
A.4.3	Observaciones sobre los proyectos de STEP 7 V2.1 con comunicación de datos globales (GD).....	A-68
A.4.4	Ampliar esclavos DP creados con versiones anteriores de STEP 7.....	A-68
A.4.5	Esclavos DP con archivos GSD defectuosos o inexistentes .....	A-69
A.5	Programas de ejemplo.....	A-70
A.5.1	Proyectos y programas de ejemplo.....	A-70
A.5.2	Proyectos y programas de ejemplo.....	A-71
A.5.3	Ejemplo de aplicación de las alarmas horarias.....	A-87
A.5.4	Ejemplo de aplicación de las alarmas de retardo.....	A-94
A.6	Acceso a áreas de datos del proceso y de la periferia.....	A-106
A.6.1	Acceso al área de datos de proceso .....	A-106
A.6.2	Acceso al área de datos de periferia .....	A-107
A.7	Ajustar el comportamiento del sistema.....	A-109
A.7.1	Ajustar el comportamiento del sistema.....	A-109
A.7.2	Modificar el comportamiento y las propiedades de los módulos.....	A-110
A.7.3	Aplicación de las funciones de reloj .....	A-112
A.7.4	Uso de marcas de ciclo y temporizadores .....	A-113
	<b>Índice alfabético</b>	<b>Índice-1</b>



# 1 Presentación del producto e instalación

## 1.1 Guía de orientación de STEP 7

### ¿En qué consiste el software STEP 7?

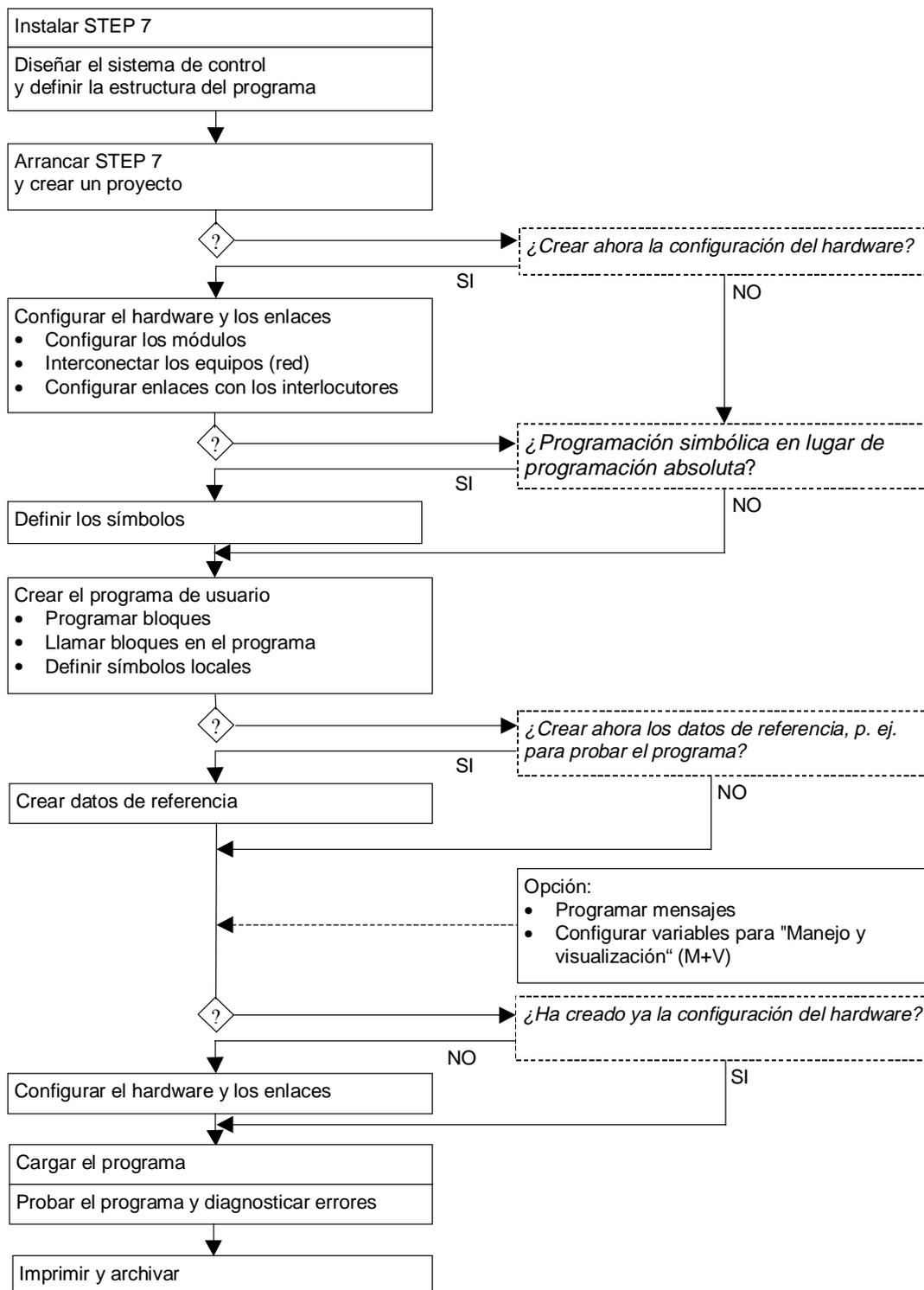
STEP 7 es el software estándar para configurar y programar los sistemas de automatización SIMATIC. STEP 7 forma parte del software industrial SIMATIC. El software estándar STEP 7 presenta las siguientes variantes:

- STEP 7-Micro/DOS y STEP 7-Micro/WIN para aplicaciones stand-alone sencillas en sistemas de automatización SIMATIC S7-200.
- STEP 7 para aplicaciones en sistemas de automatización SIMATIC S7-300/400, SIMATIC M7-300/400 y SIMATIC C7 con funciones ampliadas:
  - ampliable con los productos de software opcionales integrados en el Software Industrial SIMATIC (consulte también Posibilidades de ampliar el software estándar STEP 7)
  - Posibilidad de parametrizar bloques de función y de comunicación
  - Forzado y modo multiprocesador
  - Comunicación de datos globales
  - Transferencia de datos controlada por eventos con bloques de comunicación y de función
  - Configuración de enlaces

La presente documentación contempla el software STEP 7, mientras que STEP 7-Micro se describe en la documentación "STEP 7-Micro/DOS".

## Trabajos básicos

Al crear una solución de automatización con STEP 7 se deben realizar los trabajos que describiremos a continuación. La siguiente figura muestra las tareas básicas que se deben realizar en la mayoría de los proyectos, las cuales aparecen representadas en la figura en forma de organigrama, al cual nos referiremos en adelante con "Guía de orientación de STEP 7". Ésta señala los capítulos en cuestión, permitiéndole orientarse a través del manual en función de las tareas requeridas.



## Procedimientos alternativos

Como muestra la figura anterior, hay dos procedimientos alternativos:

- Puede configurar primero el hardware y programar luego los bloques, o bien
- programar primero los bloques sin tener que configurar antes el hardware. Esto es especialmente recomendable cuando se deban realizar trabajos de mantenimiento, p.ej. al integrar bloques programados en un proyecto ya existente.

## Resumen breve de los pasos a realizar:

- **Instalación y autorización**  
Al utilizar STEP 7 por primera vez, es preciso instalar el software y transferir la autorización residente en el disquete al disco duro (consulte también Instalar STEP 7 y Autorización).
- **Diseñar el control**  
Antes de trabajar con STEP 7, planifique su solución de automatización dividiendo primero el proceso en diversas tareas y creando luego un plano de configuración (consulte también Procedimiento básico para diseñar una solución de automatización).
- **Crear la estructura del programa**  
Las tareas descritas en el diseño del control se tienen que plasmar en un programa estructurado en base a los bloques disponibles en STEP 7 (consulte también Bloques en el programa de usuario).
- **Iniciar STEP 7**  
STEP 7 se arranca desde el interface de Windows (consulte también Arrancar STEP 7).
- **Crear la estructura del proyecto**  
Un proyecto es una carpeta que contiene todos los datos estructurados jerárquicamente, estando disponibles en cualquier momento. Tras crear un proyecto, todos los demás trabajos se realizan en el mismo (consulte también Estructura de los proyectos).
- **Crear el equipo**  
Al crear el equipo se define el sistema de automatización utilizado: p.ej. SIMATIC 300, SIMATIC 400, SIMATIC S5 (consulte también Insertar equipos).
- **Configurar el hardware**  
Al configurar el hardware se define en una tabla de configuración qué módulos se utilizarán para la solución de automatización y a través de qué direcciones se accederá a los módulos desde el programa de usuario. Además, las propiedades de los módulos se pueden ajustar mediante parámetros (consulte también Pasos fundamentales para configurar el hardware).
- **Configurar redes y enlaces de comunicación**  
Para poder establecer comunicaciones con otras estaciones primero hay que configurar una red. Para ello se deben crear las subredes necesarias para la red de autómatas, definir las propiedades de las subredes, parametrizar las propiedades de conexión de los equipos que la integran, así como determinar los enlaces de comunicación requeridos (consulte también Procedimiento para configurar una subred).
- **Definir los símbolos**  
En lugar de utilizar direcciones absolutas es posible definir símbolos locales o globales en una tabla de símbolos, empleando nombres autoexplicativos que se utilizarán luego en el programa (consulte también Crear una tabla de símbolos)

- Crear el programa  
El programa, que puede estar asignado o no a un módulo, se crea utilizando uno de los lenguajes de programación disponibles. Después se deposita en una carpeta en forma de bloque, fuente o esquema (consulte también Procedimiento básico para crear bloques lógicos y Nociones básicas para programar en fuentes AWL).
- Sólo S7: Crear y evaluar los datos de referencia  
Los datos de referencia se pueden utilizar para poder comprobar y modificar más fácilmente el programa de usuario (consulte también Sinopsis de los posibles datos de referencia).
- Configurar mensajes  
Por ejemplo, se pueden crear mensajes de bloques con sus textos y atributos. Utilizando el programa de transferencia, los datos de configuración de mensajes que se hayan creado se transfieren al contingente de datos del sistema de manejo y visualización (p.ej. SIMATIC WinCC, SIMATIC ProTool), consulte también Configurar mensajes.
- Configurar variables M+V  
En STEP 7 se crea una sola vez una variable M+V, asignándole a la misma los atributos deseados. Utilizando el programa de transferencia, las variables M+V que se hayan creado se transfieren al contingente de datos del sistema de manejo y visualización WinCC (consulte también Configurar variables para manejo y visualización).
- Cargar programas en el sistema de destino  
Sólo para S7: Tras concluir la configuración, la parametrización y la creación del programa, es posible cargar el programa de usuario entero o cualquiera de sus bloques en el sistema de destino (módulo programable perteneciente a la configuración de hardware; consulte también Requisitos para cargar). La CPU ya contiene el sistema operativo.  
  
Sólo para M7: Elija el sistema operativo más adecuado para su solución de automatización y cárguelo desde el soporte de datos deseado en el sistema de destino M7, bien sea por separado o junto con el programa de usuario.
- Comprobar los programas  
Sólo para S7: Para probar el programa puede visualizar los valores de las variables de su programa de usuario o de una CPU, asignarles valores a las mismas y crear una tabla de las variables que desea visualizar o forzar (consulte también Introducción al test con tablas de variables).  
Sólo M7: Comprobar el programa de usuario utilizando un depurador con un lenguaje de alto nivel.
- Vigilar el funcionamiento, diagnosticar el hardware  
La causa de un fallo en un módulo se determina visualizando informaciones online acerca del mismo. La causa de un fallo en la ejecución del programa de usuario se determina evaluando el búfer de diagnóstico y el contenido de las pilas. Asimismo es posible comprobar si un programa de usuario se puede ejecutar en una CPU determinada (consulte también Diagnosticar el hardware y buscar errores).
- Documentar la instalación  
Tras crear un proyecto o una instalación es recomendable documentar claramente los datos del proyecto para facilitar las tareas de ampliación y modificación y los trabajos de mantenimiento (consulte también Imprimir la documentación de un proyecto). DOCPRO, la herramienta opcional para crear y gestionar la documentación de la instalación, permite estructurar los datos del proyecto, prepararlos en forma de manuales de circuitos normalizados e imprimirlos con un diseño homogéneo.

## Incorporación de temas especiales en la guía de orientación

A la hora de crear una solución de automatización hay diversos temas especiales que pueden ser de interés:

- Modo multiprocesador - Funcionamiento síncrono de varias CPUs (consulte también Modo multiprocesador - Funcionamiento síncrono de varias CPUs)
- Creación de un proyecto por un grupo de usuarios (consulte también Elaboración de proyectos por un grupo de usuarios)
- Trabajo con sistemas M7 (consulte también Procedimiento para sistemas M7)

## 1.2 El software estándar STEP 7

### Estándares utilizados

Los lenguajes de programación SIMATIC integrados en STEP 7 cumplen con la norma DIN EN 6.1131-3. El software estándar se ejecuta bajo el sistema operativo Windows 95/98/NT/2000, estando adaptado a su funcionamiento gráfico y orientado a los objetos.

### Funciones del software estándar

El software estándar le asiste en todas las fases de creación de soluciones de automatización, tales como

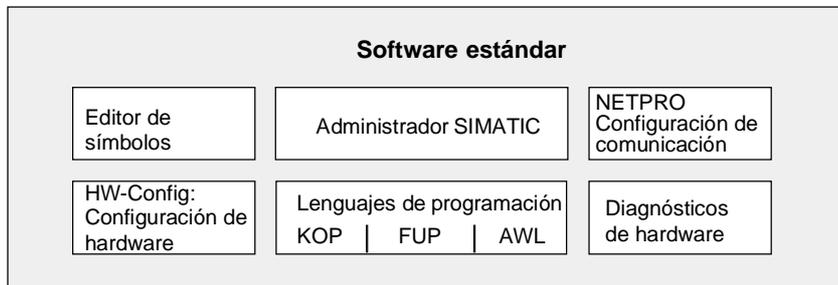
- crear y gestionar proyectos
- configurar y parametrizar el hardware y la comunicación
- gestionar símbolos
- crear programas, p.ej. para sistemas de destino S7
- cargar programas en sistemas de destino
- comprobar el sistema automatizado
- diagnosticar fallos de la instalación

El interface de usuario del software STEP 7 ha sido diseñado siguiendo los criterios ergonómicos más avanzados, lo que permite conocer rápidamente sus funciones.

La documentación del software STEP 7 contiene la información completa en la Ayuda en pantalla y en los manuales electrónicos en formato PDF.

## Herramientas auxiliares

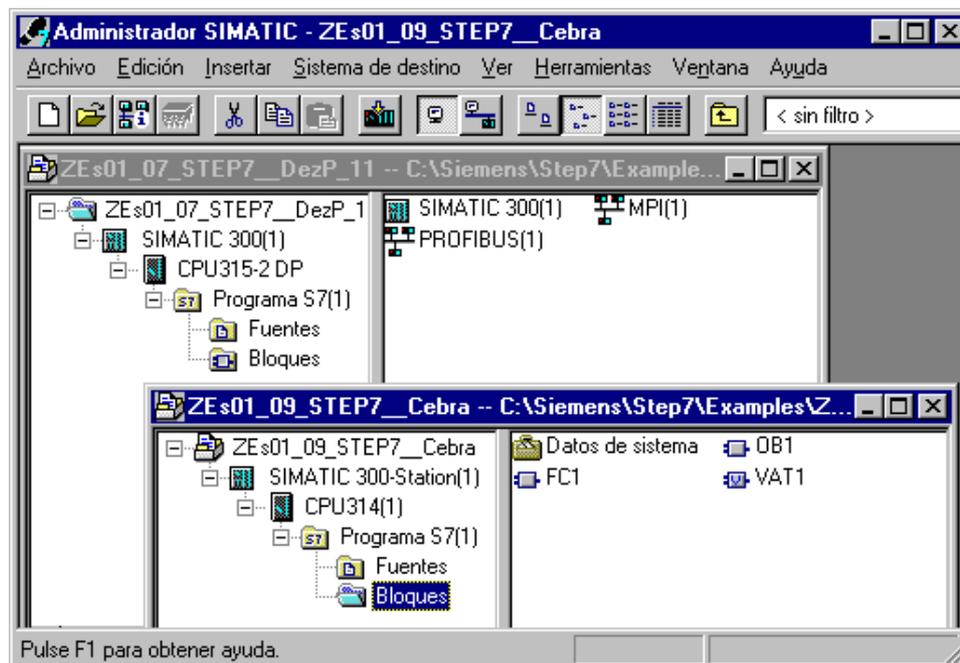
El software estándar STEP 7 ofrece toda una serie de herramientas:



Las herramientas no se deben llamar por separado, puesto que arrancan automáticamente al seleccionarse una determinada función o al abrirse un objeto.

## Administrador SIMATIC

El Administrador SIMATIC gestiona todos los datos pertenecientes al proyecto de automatización, independientemente del sistema de destino (S7/M7/C7) donde se encuentren. El Administrador SIMATIC arranca automáticamente las herramientas necesarias para tratar los datos seleccionados.



## Editor de símbolos

Con el editor de símbolos se gestionan todas las variables globales. Se dispone de las siguientes funciones:

- Definir nombres simbólicos y comentarios para las señales del proceso (entradas y salidas), las marcas y los bloques,
- funciones de ordenación,
- importación/exportación de/hacia otros programas de Windows.

Todas las herramientas pueden acceder a la tabla de símbolos creada. Por consiguiente, detectan automáticamente si se ha modificado un parámetro de un símbolo.

## Diagnóstico del hardware

El diagnóstico del hardware permite visualizar el estado del sistema de automatización, ofreciendo una panorámica que indica mediante un símbolo si alguno de los módulos presenta un fallo o no. Con un doble clic en el módulo averiado se visualizan informaciones detalladas sobre el error. El volumen de información disponible depende del módulo en cuestión:

- visualización de informaciones generales sobre el módulo (p.ej. número de referencia, versión, denominación) y sobre su estado (p.ej. fallo),
- visualización de los fallos del módulo (p.ej. errores de canal) de la periferia centralizada y de los esclavos DP,
- visualización de los mensajes del búfer de diagnóstico.

En el caso de las CPUs se visualizan además las siguientes informaciones:

- causas de una ejecución errónea del programa de usuario,
- duración del ciclo (máximo, mínimo y último),
- características y grado de utilización de la comunicación MPI,
- datos característicos (cantidad de entradas y salidas, marcas, contadores, temporizadores y bloques posibles).

## Lenguajes de programación

Los lenguajes de programación KOP, AWL y FUP para S7-300/400 son parte integrante del software estándar.

- KOP (esquema de contactos) es un lenguaje de programación gráfico. La sintaxis de las instrucciones es similar a la de un esquema de circuitos. KOP permite observar la circulación de la corriente a través de contactos, elementos complejos y bobinas.
- AWL (lista de instrucciones) es un lenguaje de programación textual orientado a la máquina. En un programa creado en AWL, las instrucciones equivalen en gran medida a los pasos con los que la CPU ejecuta el programa. Para facilitar la programación, AWL se ha ampliado con estructuras de lenguajes de alto nivel (tales como accesos estructurados a datos y parámetros de bloques).
- FUP (diagrama de funciones) es un lenguaje de programación gráfico que utiliza los cuadros del álgebra booleana para representar la lógica. Asimismo, permite representar funciones complejas (p.ej. funciones matemáticas) mediante cuadros lógicos.

Además ofrecemos otros lenguajes de programación opcionales.

## HW-Config: Configuración del hardware

Esta herramienta se utiliza para configurar y parametrizar el hardware de un proyecto de automatización. Se dispone de las siguientes funciones:

- Para configurar el sistema de automatización, se eligen primero los bastidores (racks) de un catálogo electrónico y luego se asignan los módulos seleccionados a los slots de los bastidores.
- La configuración de la periferia descentralizada se efectúa del mismo modo. También se asiste la periferia canal a canal (granular).
- Al parametrizar la CPU se pueden ajustar mediante menús propiedades tales como el comportamiento en el arranque y la vigilancia del tiempo de ciclo. Se asiste el modo multiprocesador. Los datos introducidos se depositan en bloques de datos del sistema.
- Al configurar los módulos, todos los datos se pueden ajustar en cuadros de diálogo. No es preciso efectuar ajustes mediante los interruptores DIP. La parametrización de los módulos se efectúa automáticamente durante el arranque de la CPU. Por consiguiente se puede p.ej. sustituir un módulo sin necesidad de repetir la parametrización.
- La parametrización de módulos de función (FMs) y de procesadores de comunicaciones (CPs) se efectúa con la misma herramienta de configuración del hardware de forma idéntica a como se parametrizan los demás módulos. Para los FM y CP se dispone de cuadros de diálogo específicos de los módulos (que forman parte del volumen de suministro del paquete de funciones FM/CP). El sistema impide que se efectúen entradas incorrectas, ya que los cuadros de diálogo sólo ofrecen las entradas admisibles.

## NetPro

Con NetPro, los datos se pueden transferir de forma cíclica y temporizada a través de MPI, permitiendo

- seleccionar las estaciones que intervienen en la comunicación e
- introducir la fuente y el destino de los datos en una tabla. La creación de todos los bloques a cargar (SDB) y su transferencia completa a todas las CPUs se efectúa de forma automática.

Además, existe la posibilidad de transferir los datos de forma controlada por eventos, pudiéndose

- definir los enlaces de comunicación,
- seleccionar los bloques de comunicación o de función de la librería de bloques integrada,
- parametrizar en el lenguaje de programación habitual los bloques de comunicación o de función seleccionados.

## 1.3 Novedades de la versión 5.1 de STEP 7

### Administrador SIMATIC

- Para traducir proyectos a otros idiomas se pueden editar los textos del proyecto (p. ej., títulos de bloques y comentarios) con un editor ASCII externo a STEP 7 o una herramienta de procesamiento de tablas con los comandos de menú **Herramientas > Gestionar textos en varios idiomas > Exportar** y, a continuación, volver a importarlos en STEP 7 con **Herramientas > Gestionar textos en varios idiomas > Importar**. El formato del archivo de exportación está definido como "\*.csv" (comma separated value).
- Los datos completos del proyecto se pueden cargar en una Memory Card de una CPU apta para ello (nuevos comandos de menú **Sistema de destino > Guardar proyecto en Memory Card** y **Sistema de destino > Cargar proyecto de Memory Card**)
- Con los comandos de menú **Herramientas > Datos de referencia > Borrar** se pueden borrar los datos de referencia existentes.
- La información sobre la versión de los productos instalados con sus componentes y DLLs se puede ver con el comando de menú **Ayuda > Acerca de**.
- Con el comando de menú **Edición > Comprobar coherencia del bloque** se puede comprobar la coherencia de todos los bloques S7 contenidos en la carpeta de bloques después de modificar el programa. Así podrá comprobar mejor las repercusiones que haya podido tener una modificación del interface sobre otros bloques y así solucionar más rápido los errores.
- Los atributos de sistema que ya haya definido para los bloques de su programa de usuario, se pueden conservar al importar nuevas versiones de bloques (p.ej., al actualizar una librería de sistema). Existe un cuadro de diálogo que permite sincronizar los atributos de cada bloque.

### Programar bloques KOP/AWL/FUP

- Con el nuevo comando de menú **Archivo > Comprobar y actualizar accesos** se inicia la comprobación de la coherencia de los bloques.
- Existe la posibilidad de observar los bloques llamados con "**Observar**" o con "**Observar con ruta de llamada**" si el modo de funcionamiento elegido es "Test". Para ello deberá abrir el bloque llamante y colocar el cursor sobre la llamada deseada (línea CALL en AWL y Callbox en KOP/FUP). Con la tecla derecha del ratón elija entre los comandos **Bloque llamado > Observar** y **Bloque llamado > Observar con ruta de llamada**.
- Al borrar un bloque también se borra el símbolo del bloque. Como consecuencia, las fuentes de direccionamiento simbólico ya no se pueden compilar si se han borrado del programa los bloques correspondientes. Al copiar o mover un bloque permanece intacto el símbolo.

## Observar/forzar variables

- La tabla para observar y forzar variables ha sido revisada:
  - Selección de columnas
  - Selección de varias columnas
  - Todas las columnas se pueden visualizar y ocultar
  - El formato de visualización se puede editar.
- Cuadro de diálogo "Preferencias" con dos nuevas fichas ("General" y "Online"); en la ficha "Online" se pueden elegir las siguientes opciones:
  - Preselección para el enlace online: a la CPU conectada directamente o bien a la CPU configurada.
  - Desactivación de advertencias
  - Opción "Agrupar variables" para aumentar el número máximo de variables observables
- Cambiar de enlace si haber deshecho el enlace existente
- El disparo para la observación se puede ajustar mientras se están observando variables.
- Las variables seleccionadas se pueden forzar marcando las líneas correspondientes y ejecutando la función "Forzar". Se forzarán solamente las variables que están visibles.
- Numerosos comandos nuevos: entre otros:
  - Presentación preliminar (menú "Tabla")
  - Restablecer organización (menú "Ventana")
  - Establecer enlace con 1, 2, 3, 4 (menú "Sistema de destino"; para cambiar rápidamente de un enlace a otro de los ya utilizados)

## Configurar y diagnosticar el hardware

- Observar/forzar entradas/salidas al configurar el hardware (nuevo comando de menú **Sistema de destino > Observar/Forzar**)
- Nuevos módulos, p. ej., IM 151/CPU como esclavo DP inteligente de la familia ET 200S
- Posibilidades de configuración ampliadas en los esclavos DP inteligentes:  
Asignación de la imagen de proceso parcial para las CPUs S7-400 con intercambio directo de datos y  
asignación de OBs de alarma de proceso para el interlocutor PROFIBUS (para esclavos I que pueden disparar una alarma de proceso en el maestro DP controlados por el programa de usuario)
- Mejoras ergonómicas para la función online "Información del módulo":  
Para la ficha "Búfer de diagnóstico" se pueden mostrar los eventos filtrados (ocultar determinadas clases de evento)  
En la ficha "Datos característicos" ahora se recoge información sobre bloques de organización, funciones del sistema (SFC y SFB) y áreas de operandos. La información sobre la memoria está ahora completa en la ficha "Memoria".  
La representación gráfica del tiempo de ciclo con los correspondientes tiempos de vigilancia se ha mejorado mediante una disposición horizontal de la coordenada de tiempo. En esta representación se pueden detectar más fácilmente los rebases por exceso y por defecto de los tiempos de vigilancia parametrizados.

## Configurar redes y enlaces

- Nuevas columnas en la tabla de enlaces: Interface local y del interlocutor así como dirección local y de interlocutor. Las columnas se pueden mostrar y ocultar independientemente. De esta forma se puede consultar completamente la vía de enlace y p. ej., ordenar por interfaces o subredes.
- Los ajustes en NetPro se almacenan al cerrar el proyecto y vuelven a estar disponibles la siguiente vez que se abre (también en otro PG).
- Las subredes se pueden distinguir más fácilmente porque se representan en la pantalla con diferentes colores.  
En el cuadro de diálogo para las preferencias de impresión se pueden desactivar los colores para la impresión gráfica.  
Además, se puede ajustar el zoom para que la impresión de la representación de la red se adapte de forma que se pueda aprovechar de la mejor forma posible el número de páginas disponibles.
- Además de los parámetros de bus para PROFIBUS, se pueden imprimir también parámetros de bus para otras subredes (MPI).
- Compatibilidad de la configuración de los enlaces (enlaces S7) y el estado de los enlaces para las nuevas WinAC-SlotCPU (CPU 41x-2 DP PCI)

## Datos de referencia

- Con el comando de menú **Edición > Borrar símbolos** se pueden borrar los símbolos no utilizados, en la vista "Símbolos no utilizados".
- Con el comando de menú **Edición > Editar símbolos** se pueden asignar símbolos a los operandos seleccionados, en la vista "Operandos sin símbolo".
- Si está seleccionado el comando de menú **Ventana > Guardar configuración al salir**, la disposición de las ventanas se guarda al cerrar la aplicación y se vuelve a restablecer, independientemente de la vista mostrada (Referencias cruzadas, Estructura del programa, ...)

## Configurar mensajes

- También se pueden crear mensajes de diagnóstico personalizados para programas M7.
- El cuadro de diálogo "Configuración de mensajes PCS7" para editar un bloque de comunicación controlado por eventos contiene dos fichas en las que se pueden introducir hasta 10 textos de mensaje.

## Mensajes de CPU

- Existen distintas opciones para editar los mensajes que llegan a la aplicación Mensajes de CPU:
  - Con el comando de menú **Ver > Desplazar automáticamente**, los mensajes nuevos que llegan siempre se desplazan a la ventana y se seleccionan.
  - Con el comando de menú **Ver > Primer plano**, la ventana pasa a primer plano y se muestra el mensaje.
  - Con el comando de menú **Ver > Segundo plano**, los mensajes se muestran en la ventana, pero ésta permanece en segundo plano.
  - Con el comando de menú **Ver > Ignorar mensaje**, los mensajes ni se muestran en la ventana ni se guardan en el archivador.
- Con el comando de menú **Sistema de destino > Eliminar módulo** puede eliminar de la lista los módulos seleccionados.
- Con el cuadro de diálogo "Preferencias - Mensajes de CPU" se puede ajustar el tamaño del archivador, guardar la lista de los módulos registrados y restablecer el estado de los enlaces al iniciar. También se pueden ver los textos informativos en ALARM S/SQ.

## Notificar errores del sistema

- Con la función "Notificar errores del sistema", STEP 7 ofrece una forma cómoda de ver en forma de mensajes la información de diagnóstico proporcionada por los componentes. Los bloques y textos de mensaje necesarios para ello son generados automáticamente por STEP 7. El usuario sólo tiene que cargar en la CPU los bloques generados y transferir los textos a los aparatos HMI conectados. Encontrará una sinopsis detallada de la información de diagnóstico que ofrecen los diferentes esclavos DP bajo el tema "Componentes compatibles y funcionalidad".

## 1.4 Posibilidades de ampliar el software estándar STEP 7

### 1.4.1 Posibilidades de ampliar el software estándar STEP 7

El software estándar se puede ampliar con productos de software opcionales agrupados en las tres categorías siguientes:

- Herramientas de ingeniería:  
abarcan lenguajes de programación de alto nivel y software orientado a la tecnología.
- Software Runtime:  
contiene software ya listo para el proceso de producción.
- Interfaces hombre-máquina (HMI):  
constituyen el software especial para manejo y visualización.

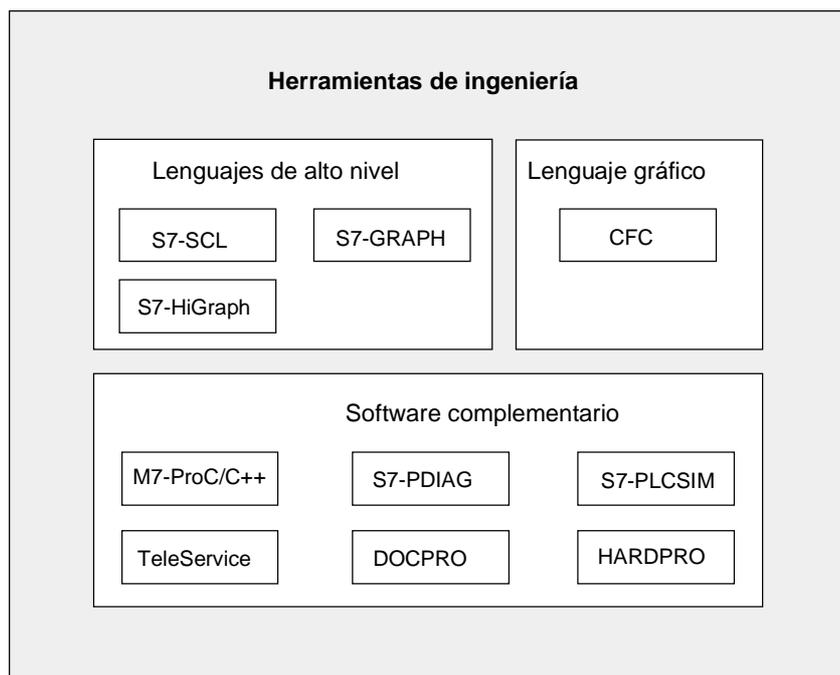
La siguiente tabla muestra el software opcional disponible en función del sistema de automatización utilizado:

	STEP 7		
	S7-300 S7-400	M7-300 M7-400	C7-620
Herramientas de ingeniería			
Borland C/C++		o	
CFC	+ <sup>1)</sup>	+	+ <sup>2)</sup>
DOCPRO	+	+ <sup>3)</sup>	+
HARDPRO	+		
• M7-ProC/C++		o	
• S7-GRAPH	+ <sup>1)</sup>		+ <sup>2)</sup>
• S7-HiGraph	+		+
• S7-PDIAG	+		
• S7-PLCSIM	+		+
S7-SCL	+		+
• TeleService	+	+	+
Software Runtime			
• Fuzzy Control	+		+
• M7-DDE-Server		+	
• M7-SYS RT		o	
• Modular PID Control	+		+
• PC-DDE-Server	+		
• PRODAVE MPI	+		
• Standard PID Control	+		+
Interfaces hombre-máquina			
• ProAgent			
• SIMATIC ProTool			
• SIMATIC ProTool/Lite			o
• SIMATIC WinCC			
o = absolutamente necesario + = opcional <sup>1)</sup> = recomendado a partir de S7-400 <sup>2)</sup> = no recomendado para C7-620 <sup>3)</sup> = no apto para programas C			

## 1.4.2 Herramientas de ingeniería

Las herramientas de ingeniería están orientadas a aquellas tareas que se pueden instalar para ampliar el software estándar. Las herramientas de ingeniería abarcan:

- Lenguajes de alto nivel para los programadores
- Lenguaje gráfico para los tecnólogos
- Software opcional para el diagnóstico, la simulación, la asistencia técnica a distancia, la documentación de la instalación, etc.



### Lenguajes de alto nivel

Para programar los sistemas de automatización SIMATIC S7-300/400 se dispone de los siguientes lenguajes opcionales:

- S7-GRAPH es un lenguaje de programación que permite describir cómodamente controles secuenciales (programación de cadenas secuenciales) dividiendo el proceso en diferentes etapas. Estas últimas contienen sobre todo acciones para controlar las salidas. El paso de una etapa a otra se controla mediante condiciones de transición.
- S7-HiGraph es un lenguaje de programación que permite describir cómodamente los procesos asíncronos y no secuenciales en forma de grafos de estado. Para ello se divide la instalación en unidades funcionales que pueden adoptar diversos estados. Las unidades funcionales se pueden sincronizar mediante el intercambio de mensajes.
- S7-SCL es un lenguaje textual de alto nivel según la norma DIN EN 61131-3. Contiene estructuras similares a las de los lenguajes de programación Pascal y C. Por consiguiente, S7-SCL es especialmente apropiado para los usuarios que ya estén acostumbrados a utilizar lenguajes de nivel superior. S7-SCL se puede utilizar p.ej. para programar funciones repetitivas o muy complejas.

## Lenguaje gráfico

CFC para S7 y M7 es un lenguaje de programación para interconectar gráficamente las funciones existentes. Estas últimas abarcan una amplia gama que incluye desde combinaciones lógicas sencillas hasta regulaciones y controles complejos. Se dispone de un gran número de funciones en forma de bloques contenidos en una librería. La programación se lleva a cabo copiando los bloques en un esquema (plano) y conectándolos entre sí mediante líneas.

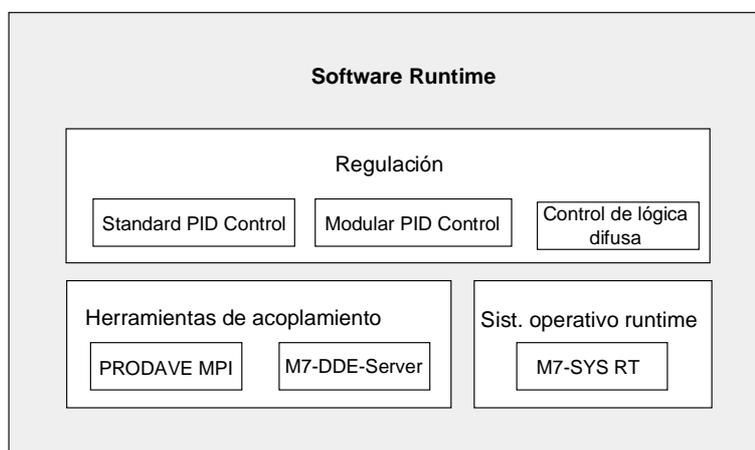
## Software opcional

- Borland C++ (sólo M7) contiene el entorno de desarrollo Borland.
- Con DOCPRO se pueden organizar todos los datos de configuración que se hayan creado con STEP 7 en manuales de circuitos. Dichos manuales permiten gestionar cómodamente los datos de configuración y preparar una impresión normalizada.
- HARDPRO es el sistema de configuración de hardware para S7-300 que le facilita al usuario la configuración de tareas de automatización complejas.
- M7-ProC/C++ (sólo M7) permite incorporar el entorno de desarrollo Borland para los lenguajes de programación C y C++ en el entorno de desarrollo STEP 7.
- Con S7-PLCSIM (sólo S7) es posible simular sistemas de automatización S7 conectados al sistema de origen (PC/PG) para someterlos a un test.
- S7-PDIAG (sólo S7) permite configurar globalmente el diagnóstico de procesos para SIMATIC S7-300/400. Con dicho diagnóstico se pueden detectar estados erróneos fuera del sistema de automatización (p.ej. final de carrera no alcanzado).
- A través de la red telefónica, TeleService permite programar los sistemas de automatización S7 y M7, así como realizar trabajos de mantenimiento, utilizando para ello una PG/un PC .

### 1.4.3 Software Runtime

El Software Runtime abarca programas ya listos que se pueden llamar desde el programa de usuario. El Software Runtime se incorpora directamente en la solución de automatización. Dicho software abarca:

- regulaciones para SIMATIC S7, p.ej. regulaciones estándar, modulares y de lógica difusa
- herramientas para acoplar los sistemas de automatización con aplicaciones de Windows
- un sistema operativo de tiempo real para SIMATIC M7



## Regulaciones para SIMATIC S7

- Standard PID Control permite integrar en el programa de usuario los reguladores continuos, por impulsos y discontinuos . La herramienta de parametrización con ajuste de regulación incorporado permite parametrizar y ajustar rápidamente los reguladores.
- Modular PID Control se utiliza cuando un regulador PID sencillo no es suficiente para solucionar la tarea de automatización. Interconectando el bloque de función estándar incorporado se pueden realizar casi todas las estructuras de regulación.
- Con el Control de lógica difusa se pueden crear sistemas de lógica polivalente. Dichos sistemas son necesarios cuando resulta muy difícil o imposible describir los procesos matemáticamente, cuando éstos se ejecutan de forma impredecible o cuando se presentan no linealidades, y por otro lado se dispone de conocimientos acerca del comportamiento del proceso.

## Herramientas de acoplamiento

- PRODAVE MPI es una caja de herramientas para la comunicación de datos del proceso entre SIMATIC S7, SIMATIC M7 y SIMATIC C7. PRODAVE MPI realiza independientemente la comunicación de datos a través del interface MPI.
- Con el M7-DDE-Server (**D**ynamic **D**ata **E**xchange), las aplicaciones de Windows se pueden conectar a las variables de proceso de SIMATIC M7 sin necesidad de programas adicionales.

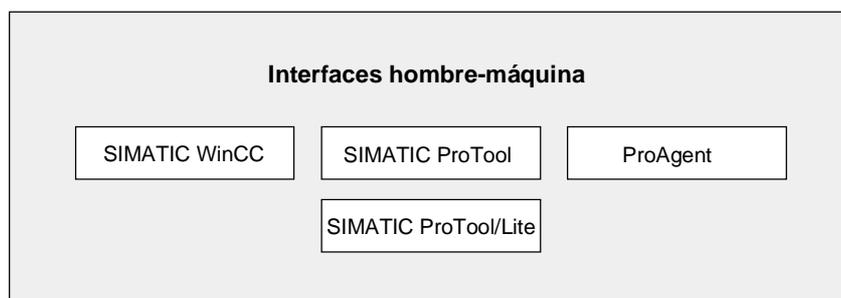
## Sistema operativo runtime

- M7-SYS RT incorpora el sistema operativo M7 RMOS 32 y programas de sistema. Se requiere para poder utilizar los paquetes M7 ProC/C++ y CFC para SIMATIC M7.

### 1.4.4 Interfaces hombre-máquina

Human Machine Interface, el software especial de manejo y visualización para SIMATIC.

- El sistema abierto de visualización de procesos SIMATIC WinCC es un sistema básico independiente del ramo y de la tecnología, que incorpora todas las funciones importantes de manejo y visualización.
- SIMATIC ProTool y SIMATIC ProTool/Lite son herramientas modernas para configurar los paneles de operador SIMATIC y los equipos compactos SIMATIC C7.
- ProAgent facilita el diagnóstico de procesos puntualizado y rápido en instalaciones y máquinas, averiguando informaciones relativas a la ubicación y a la causa del error.



## 2 Instalación y autorización

### 2.1 Autorización

#### 2.1.1 Autorización

Para poder utilizar el software de programación STEP 7 se requiere una autorización específica para el producto (permiso de utilización). Este software protegido sólo se puede utilizar si en el disco duro de la PG o del PC se detecta la autorización necesaria para el programa o para el paquete de software.

Se requieren autorizaciones diferentes p.ej. para STEP 7 y el software opcional.

#### 2.1.2 Instalar y desinstalar la autorización

##### Disquete de autorización

Para instalar la autorización se requiere el correspondiente disquete protegido contra copia que se suministra con el software. Dicho disquete contiene la autorización en sí. El programa "AuthorsW", necesario para visualizar, instalar y desinstalar la autorización, se encuentra en el CD-ROM de STEP 7 V5.1.

El número permitido de autorizaciones está fijado en dicho disquete mediante un contador de instalaciones. Dicho número se reduce en 1 al instalar una autorización. Cuando el contador alcanza el valor "cero", no se pueden instalar más autorizaciones con el disquete.

---

##### Nota

Para el software estándar STEP 7 se suministra un disquete de autorización amarillo con la autorización correspondiente.

Para cada software opcional se suministra un disquete de autorización rojo con una autorización.

---



##### Cuidado

Tenga en cuenta las indicaciones contenidas en el archivo LEAME.WRI del disquete de autorización, así como las "Reglas para el empleo de autorizaciones". En caso contrario se corre el peligro de perder la autorización de forma irreversible.

---

El software estándar se puede utilizar también sin autorización para familiarizarse con el interface de usuario y el volumen de funciones. No obstante, sólo se podrá trabajar de forma correcta y admisible tras haber instalado la autorización. Si aún no ha instalado la autorización, se le recordará en intervalos regulares que debe hacerlo.

### Si se pierde la autorización ...

La autorización se puede perder a causa de un defecto en el disco duro que impida desinstalarla.

En tal caso se puede utilizar la autorización de emergencia. Esta se encuentra también en el disquete de autorización. La autorización de emergencia permite seguir utilizando el software por un tiempo limitado. Al iniciar el software, se indica el tiempo que queda hasta su vencimiento. Dentro de dicho plazo es necesario sustituir la autorización que se ha perdido. Para ello diríjase a la representación de SIEMENS más próxima.

---

#### Nota

El plazo disponible para la autorización de emergencia comienza en el momento de la instalación de la misma, aun cuando no se arranque STEP 7. Aunque vuelva a transferir la autorización al disquete, no se detendrá el transcurso del plazo.

---

### Instalación de AuthorsW

El programa "AuthorsW", necesario para visualizar, instalar y desinstalar la autorización, se suministra en el CD-ROM de STEP 7 V5.1. Dicho programa se instala en el disco duro, pudiéndose utilizar desde allí para las operaciones de autorización.

---

#### Nota

El programa AuthorsW se encuentra en **Inicio > Simatic > AuthorsW > AuthorsW**

---

### Transferir la autorización durante la primera instalación

Es recomendable transferir la autorización durante la primera instalación de STEP 7 cuando aparezca el mensaje correspondiente. Para transferir la autorización durante la primera instalación de STEP 7:

1. Introduzca el disquete de autorización cuando aparezca el correspondiente mensaje.
2. Confirme dicho mensaje.
3. La autorización se transfiere a una unidad de disco física.

### Instalar la autorización posteriormente

Si arranca el software STEP 7 sin disponer de la autorización pertinente, aparecerá un mensaje indicando su ausencia. Para instalar la autorización posteriormente:

1. Introduzca el disquete de autorización en una unidad de disco, p. ej., en la unidad A.
2. Llame el programa "Authorsw.exe" del disco duro.
3. Elija en uno de ambos cuadros de lista aquel<sla unidad de disco en la que reside la autorización y, en el otro, la unidad de destino. Se visualizarán las autorizaciones que residen en ambas unidades.
4. Seleccione la autorización deseada.
5. Haga clic en el botón "←" o "→". La autorización que esté marcada se transferirá a la unidad seleccionada.
6. Cierre el cuadro de diálogo.

### Nota

La autorización sólo funcionará bajo Windows NT si tiene pleno acceso a la unidad de disco duro "C:", así como a la unidad de origen.

---

### Actualizar la autorización

Utilice el comando de menú "Actualizar" para actualizar las autorizaciones. Para esta función se necesitan:

- el disquete de la autorización a actualizar,
- el programa de autorización "AuthorsW", en el disco duro,
- la nueva autorización de STEP 7 en disquete y
- la antigua autorización en disquete o en el disco duro.

En el proceso de actualización se borran todas las autorizaciones antiguas, sustituyéndose por nuevas. Por tanto, el disquete de autorización no puede estar protegido contra escritura en ningún momento.

1. Introduzca el nuevo disquete de autorización.
2. Llame el programa "Authorsw.exe" del disco duro.
3. Elija el comando **Autorización > Actualizar**. Aparecerá un cuadro de diálogo. Elija allí el programa de actualización. A continuación se le solicita que introduzca el disquete de la antigua autorización.
4. Introduzca el disquete de autorización requerido. Entonces se le pregunta si desea actualizar realmente la autorización. Esta es la última oportunidad de cancelar la acción. Tras confirmar el cuadro de diálogo, el proceso no se podrá cancelar por ningún motivo. En caso contrario se perderá la autorización.
5. Haga clic en el botón "Aceptar". A continuación se le solicita que introduzca el disquete de la nueva autorización.

Entonces se comprueban todas las condiciones necesarias. Si finaliza la comprobación sin haberse presentado errores, se finalizará la actualización, activándose la nueva autorización.

La nueva autorización se encontrará en aquella unidad, en la que residía la antigua autorización, por lo que, dado el caso, puede ser necesario instalar la autorización desde el disquete al disco duro.

### Restablecer la autorización

Si su autorización está defectuosa, diríjase por favor al servicio de ayuda permanente (hot line). Este servicio le ayudará a "salvar" la autorización con el comando de menú **Autorización > Restablecer**.

## Desinstalar la autorización

En caso de requerir una nueva autorización, p.ej. si desea reformatear la unidad donde se encuentra la misma, deberá retransferirla previamente a un disquete de autorización Siemens (o sea, desinstalarla). En dicho disquete de autorización también se pueden guardar las autorizaciones del software opcional utilizado.

Para retransferir la autorización al disquete de autorización:

1. Si quiere desinstalar la autorización en un disquete, introduzca el disquete de autorización en una unidad de disco, p. ej., en la unidad A:.
2. Llame al programa "Authorsw.exe" del disco duro.
3. Elija en uno de ambos cuadros de lista aquella unidad de disco en la que reside la autorización y, en el otro, la unidad de destino. Se visualizarán todas las autorizaciones que se encuentran en ambas unidades.
4. Seleccione la autorización deseada.
5. Haga clic en el botón "←" o "→". La autorización seleccionada se transferirá al disquete de autorización o a la unidad seleccionada.
6. Cierre el cuadro de diálogo si no desea desinstalar más autorizaciones. Más adelante puede volver a utilizar el disquete si desea instalar otra autorización.

También existe la posibilidad de mover las autorizaciones entre un disco duro y otro (y entre una unidad de red y otra).

### 2.1.3 Reglas para el empleo de autorizaciones



#### **Cuidado**

Tenga en cuenta las observaciones que figuran en el presente apartado y en el archivo LEAME.TXT del disquete de autorización. En caso contrario se corre el peligro de perder la autorización de forma irreversible.

---

#### **Desinstalación necesaria**

Antes de formatear, comprimir o restaurar su disco duro o de instalar un nuevo sistema operativo, debe desinstalar las autorizaciones existentes.

#### **Backup (copia de seguridad)**

Si un backup de su disco duro contiene copias de autorizaciones, es posible que al volver a copiar los datos de la copia de seguridad en el disco duro se sobrescriban las autorizaciones válidas instaladas allí, por lo que se perderán.

Para evitar que una copia de seguridad sobrescriba las autorizaciones deberá

- desinstalarlas antes de crear una copia de seguridad o
- excluirlas de la copia de seguridad.

#### **Optimización del disco duro**

Si utiliza un programa de optimización que permita desplazar bloques fijos, sólo podrá utilizar esta opción después de haber retransferido las autorizaciones del disco duro al disquete de autorización.

## Sectores defectuosos

Junto con la autorización se crea en la unidad de destino un cluster especial que, en algunas ocasiones, se marca como "defectuoso". No intente reparar dicho cluster. En tal caso se podría deteriorar la autorización.

## Protección contra escritura y contra copia

El disquete de autorización no puede estar protegido contra escritura.

Los archivos contenidos en el disquete de autorización se pueden copiar también a otra unidad (p.ej. al disco duro) y utilizar desde allí. No obstante, con dichos archivos copiados no se puede efectuar la autorización. Para ello se requiere el disquete de autorización original.

## Unidades de disco admisibles

La autorización sólo se puede instalar en el disco duro. En el caso de unidades comprimidas (p.ej. DBLSPACE), se puede instalar en la correspondiente unidad host.

La herramienta de autorización evita que las autorizaciones se instalen en unidades de disco no admisibles.

## Lugar de almacenamiento

Durante la instalación de la autorización, los archivos de autorización se depositan en el directorio de protección "AX NF ZZ" con los atributos "Sistema" y "Oculto".

- Los atributos no se pueden modificar.
- Los archivos no se pueden modificar ni borrar.
- La carpeta no se puede desplazar. Los archivos copiados de la carpeta (autorizaciones) se consideran defectuosos, por lo que no serán autorizaciones admisibles.

En caso contrario, la autorización se pierde irreversiblemente.

En cada unidad de disco local se crea un solo directorio de seguridad 'AX NF ZZ'. Este contiene todas las autorizaciones que se hayan instalado en dicha unidad. Se crea durante la instalación de la primera autorización y se borra al desinstalar la última.

Por cada autorización se crean en el directorio de seguridad dos archivos de igual nombre con extensión diferente. El nombre de dichos archivos coincide con el de la autorización.

## Número de autorizaciones

En una unidad de disco se puede instalar un número cualquiera de autorizaciones, en tanto que se disponga del espacio de memoria necesario. Sin embargo, sólo se podrá instalar una autorización por cada versión (p.ej. sólo una para STEP 7 V4.x y sólo una para STEP 7 V5.x). No hay riesgo de conflictos entre las autorizaciones.

## Autorización defectuosa

Las autorizaciones defectuosas que se encuentren en un disco duro no se pueden desinstalar con la herramienta de autorización AuthorsW. Incluso podrían bloquear la instalación de autorizaciones nuevas y válidas. En este caso, diríjase a la representación de SIEMENS más próxima.

## **Herramienta de autorización**

Utilice la versión actual de la herramienta de autorización AuthorsW y no las versiones anteriores.

---

### **Nota**

Puesto que a partir de la versión V2.0 no se detectan todas las autorizaciones antiguas, en estos casos es necesario utilizar una versión anterior de AUTHORS (versión DOS) < V3.x.

---

## 2.2 Instalar STEP 7

### 2.2.1 Instalar STEP 7

STEP 7 contiene un programa que efectúa la instalación automáticamente. En la pantalla van apareciendo instrucciones que le conducirán paso a paso por todo el proceso de instalación. Dicho programa se inicia de la forma usual en Windows 95/98/NT o Windows 2000 para instalar software.

Las principales fases de la instalación son:

- copiar los datos en el sistema de origen,
- instalar los drivers para la EPROM y la comunicación,
- introducir el nº de identificación e
- instalar la autorización (opcional).

---

#### Nota

Las unidades de programación de SIEMENS (tales como la PG 740) se suministran con software STEP 7 a instalar en el disco duro.

---

### Requisitos de instalación

- Sistema operativo  
Microsoft Windows 95, Windows 98, Windows 2000 o Windows NT.
- Hardware básico  
PC o unidad de programación (PG) con
  - procesador 80486 o superior (para Windows NT/2000 se requiere un procesador Pentium),
  - memoria RAM: 32 MB como mínimo, pero se recomiendan 64 MB y
  - monitor color, teclado y ratón compatibles con Microsoft Windows.

Una unidad de programación (PG) es un ordenador personal compacto e idóneo para fines industriales. Posee el equipamiento completo para programar los sistemas de automatización SIMATIC.

- Capacidad de memoria  
En el archivo "LEAME" se indica el espacio de memoria requerido en el disco duro.
- Interface MPI (opcional)  
El interface MPI entre el sistema de origen (unidad de programación o PC) y el sistema de destino sólo es necesario para comunicarse en STEP 7 vía MPI con el sistema de destino.  
Utilice para ello:
  - un cable PC/MPI conectado al interface de comunicación de su unidad o
  - una tarjeta MPI incorporada en la unidad de programación.El interface MPI ya está incorporado en algunas unidades de programación.
- Prommer externo (opcional)  
Si utiliza un PC, sólo necesita un prommer externo en caso de que desee grabar EPROMs.

---

### Indicaciones

Lea las indicaciones de instalación de STEP 7 contenidas en el archivo LEAME.WRI y la "Lista de compatibilidad de los paquetes de software SIMATIC con las versiones del software estándar STEP 7".

El archivo Léame se encuentra en el menú Inicio bajo **Inicio > Simatic > Indicaciones sobre el producto.**

La lista de compatibilidad se encuentra en el menú Inicio bajo **Inicio > Simatic > Documentación.**

---

## 2.2.2 Procedimiento de instalación

### Preparativos

Antes de poder comenzar con la instalación se deberá haber iniciado Windows 95/98/NT/2000.

- No se requieren soportes de datos externos si el software STEP 7 a instalar ya se encuentra en el disco duro de la PG.
- Para instalar STEP 7 desde disquetes, introduzca el disquete 1 en la correspondiente unidad de la PG/ del PC.
- Para instalar STEP 7 desde un CD-ROM, introdúzcalo en la unidad de CD-ROM de su PC.

## Iniciar el programa de instalación

Para iniciar la instalación:

1. Introduzca el soporte de datos (disquete 1) o el CD-ROM y arranque el programa de instalación haciendo doble clic en el archivo "setup.exe".
2. Siga paso a paso las instrucciones que el programa de instalación le muestra en la pantalla.

El programa le conduce paso a paso por el proceso de instalación, permitiendo avanzar al paso siguiente o retroceder al paso anterior.

Durante el proceso de instalación aparecen cuadros de diálogo con consultas, o bien con opciones a elegir. Lea las observaciones que se indican a continuación para responder a los diálogos de forma más rápida y fácil.

## Si ya está instalada una versión de STEP 7 ...

Si el programa de instalación detecta que ya se ha instalado una versión de STEP 7 en el sistema de origen, aparece un mensaje, pudiéndose optar por una de las siguientes alternativas:

- interrumpir la instalación (para desinstalar luego con Windows la antigua versión de STEP 7 y arrancar posteriormente la instalación) o
- proseguir con la instalación, sobrescribiendo así la antigua versión con la nueva.

Si ya existe una versión antigua, es recomendable desinstalarla antes de instalar la nueva. Si decide sobrescribir la versión antigua, puede suceder que al desinstalarla posteriormente no se borren las secciones que pudieran existir de una versión anterior.

## Volumen de instalación

Para determinar el volumen de instalación, puede elegir una de las tres posibilidades siguientes:

- Normal: todos los idiomas del interface de usuario, todas las aplicaciones y todos los ejemplos. En la "Información sobre el producto" actual se indica el espacio de memoria requerido para ello.
- Mínima: sólo un idioma, sin ejemplos. En la "Información sobre el producto" actual se indica el espacio de memoria requerido.
- Personalizada: se puede elegir el volumen de instalación que necesite en lo que respecta a los programas, a la base de datos, a los ejemplos y a la comunicación.

## Número de identificación

Durante la instalación se le solicita que introduzca un número de identificación. Este número se encuentra en el certificado de producto del software o en el disquete de autorización correspondiente.

## Autorización

Durante la instalación se comprueba si en el disco duro existe una autorización. Si ésta no se detecta, aparece un mensaje indicando que el software sólo se puede utilizar con la debida autorización. Si lo desea, puede transferir la autorización inmediatamente o continuar con la instalación y transferirla posteriormente. En el primer caso, introduzca el disquete de autorización en la correspondiente unidad cuando se le solicite.

## Ajuste del interface PG/PC

Durante la instalación aparece un cuadro de diálogo que permite ajustar el interface de la PG/del PC. Consulte a este respecto el tema de la Ayuda "Ajustar interface PG/PC".

## Parametrizar Memory Cards

Durante la instalación aparece un cuadro de diálogo para parametrizar Memory Cards.

- Si no va a utilizar Memory Cards, no necesita ningún driver EPROM. En este caso, elija la opción "sin driver EPROM".
- En caso contrario, elija el driver correspondiente a su PG.
- Si va a utilizar un PC, puede elegir un driver para un prommer externo. A tal efecto debe indicar adicionalmente el interface al que está conectado el prommer (p. ej., LPT1).

Los parámetros ajustados pueden modificarse después de la instalación, llamando el programa "Parametrizar Memory Card" en el grupo de programas de STEP 7 o en el Panel de control.

## Flash File System

En el cuadro de diálogo para parametrizar Memory Cards puede indicar si desea instalar un Flash File System.

El Flash File System se utiliza, por ejemplo, para escribir o borrar en SIMATIC M7 algunos archivos que se encuentren en una Memory Card EPROM sin tener que modificar el resto de la misma.

Si utiliza una unidad de programación apropiada (PG 720/740/760) o un prommer externo adecuado y desea utilizar esta función, elija la instalación del Flash File System.

## Errores durante la instalación

La instalación se interrumpe cuando se presenta alguno de los siguientes errores:

- Si inmediatamente después de arrancar el programa de instalación se presenta un error de inicialización, es muy probable que el *setup* no se haya iniciado en Windows.
- El espacio de memoria no es suficiente: dependiendo del volumen de la instalación, se requieren para el software estándar unos 100 MB de memoria en el disco duro.
- Disquete defectuoso o CD: si hay un disquete defectuoso, diríjase a su representante SIEMENS más próximo.
- Errores de usuario: comience de nuevo con la instalación y lea las instrucciones atentamente.

### Al concluir la instalación ...

Una vez concluida la instalación, aparece el mensaje correspondiente en la pantalla.

Si durante la instalación se actualizaron archivos de sistema, será necesario reiniciar Windows. Entonces es posible arrancar el Administrador SIMATIC, que es el interface de STEP 7.

Una vez concluida la instalación, se creará un grupo de programas para STEP 7.

### 2.2.3 Ajustar el interface PG/PC

Los ajustes que se efectúen aquí determinan la comunicación entre la PG/el PC y el sistema de automatización. Durante la instalación aparece un cuadro de diálogo que permite ajustar el interface de la PG/del PC. El cuadro de diálogo también se puede visualizar después de la instalación, llamando el programa "Ajustar interface PG/PC". Así es posible cambiar los parámetros del interface a posteriori, independientemente de la instalación.

#### Procedimiento básico

Para poder utilizar un interface es preciso:

- realizar los ajustes necesarios en el sistema operativo y
- parametrizar el interface de forma adecuada

Si utiliza su PG con una conexión MPI, no tendrá que adaptar el sistema operativo.

Si utiliza un PC con una tarjeta MPI o con procesadores de comunicaciones (CP), compruebe a través del "Panel de control" de Windows la asignación de interrupciones y de direcciones para asegurarse de que no se presenten conflictos de interrupciones y de que las áreas de direccionamiento no se solapen.

Para simplificar la parametrización del interface de la PG/del PC, es posible elegir en varios cuadros de diálogo entre diversos juegos predeterminados de parámetros básicos (parametrizaciones de interfaces).

#### Parametrizar el interface PG/PC

Proceda de la forma siguiente (en la Ayuda en pantalla se incluye una descripción detallada al respecto):

1. En el "Panel de control", haga doble clic en el icono "Ajustar interface PG/PC".
2. Ajuste "S7ONLINE" en el cuadro "Punto de acceso de la aplicación".
3. En la lista "Parametrización utilizada", elija la parametrización deseada. Si no se visualiza la parametrización deseada, deberá instalar previamente un módulo o un protocolo mediante el botón "Seleccionar". La parametrización del interface se creará entonces automáticamente.

- Si elige un interface con reconocimiento automático de los parámetros de bus (p.ej.: MPI-ISA-Card (Auto)), puede conectar la PG o el PC a MPI o a PROFIBUS, sin necesidad de ajustar previamente los parámetros de bus. No obstante, si la velocidad de transferencia es inferior a 187,5 kbit/s, se pueden producir tiempos de espera de hasta un minuto.

Requisito para el reconocimiento automático: Hay maestros conectados al bus que distribuyen cíclicamente parámetros de bus. Este es el caso en todos los nuevos componentes MPI. En las subredes PROFIBUS, la distribución cíclica de los parámetros de bus no puede estar desconectada (ajuste estándar de red PROFIBUS).

- Si se elige un interface sin reconocimiento automático de los parámetros de bus, habrá que visualizar las propiedades y adaptarlas a la subred.

Asimismo, es preciso efectuar cambios cuando se presenten conflictos con otros ajustes (p. ej., asignación de interrupciones o de direcciones). En estos casos los cambios necesarios se deberán realizar con la herramienta de detección de hardware y el "Panel de control" de Windows (v. más abajo).



### **Cuidado**

¡Si se visualiza la parametrización "TCP/IP", **no la retire!**

Ello podría afectar a la ejecución de otras aplicaciones.

---

## **Comprobar la asignación de interrupciones y direcciones**

Si utiliza un PC con tarjeta MPI, deberá comprobar si la interrupción y el área de direccionamiento predeterminadas están libres y, en caso necesario, elegir una interrupción o una área de direccionamiento libre(s).

### *Windows 95/98*

Las asignaciones actuales se pueden visualizar en Windows 95/98 de la siguiente manera:

1. En el "Panel de control", abra el "Sistema" y, en el cuadro de diálogo que aparece entonces, elija la ficha "Administrador de dispositivos".
2. En la lista, seleccione la entrada "PC" y haga clic en el botón "Propiedades".
3. En otro cuadro de diálogo puede visualizar la lista de interrupciones asignadas (IRQ) o la lista de áreas de direccionamiento asignadas (Entrada/Salida (E/S)), seleccionando la opción correspondiente.

### *Windows NT*

Bajo Windows NT puede

- visualizar los ajustes de los recursos a través de **Inicio > Programas > Herramientas administrativas > Diagnóstico de Windows NT > Recursos** y
- modificar los recursos mediante **Interface PG/PC > Instalar > Recursos**.

### *Windows 2000*

Bajo Windows 2000 puede

- visualizar los ajustes de los recursos a través de **Panel de control > Administrative Tools > Administración del equipo > Herramientas del sistema > Información del sistema > Recursos de hardware**.

## **Diferencias entre Windows 9x y Windows NT/2000**

En Windows NT/2000, las interrupciones, las áreas de direccionamiento y otros recursos se asignan en un cuadro de diálogo propio (para obtener información detallada al respecto consulte la Ayuda en pantalla).

## **2.3 Desinstalar STEP 7**

### **2.3.1 Desinstalar STEP 7**

Para desinstalar el software utilice el procedimiento usual en Windows:

1. En el "Panel de control" de Windows, haga doble clic en el icono "Agregar o quitar programas" para abrir el cuadro de diálogo que permite instalar programas.
2. Seleccione la entrada STEP 7 en la lista de programas instalados. A continuación, haga clic en el icono de "Agregar o quitar programas".
3. Si aparecen cuadros de diálogo "Agregar o quitar archivo compartido", haga clic en el botón "No" en caso de duda.

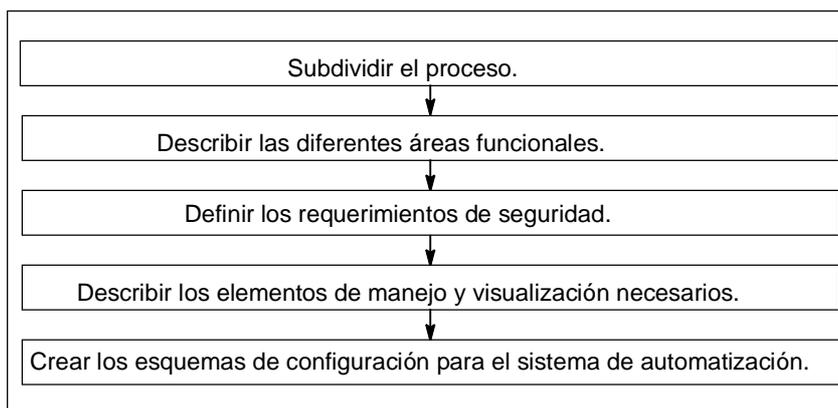


## 3 Planificar una solución de automatización

### 3.1 Procedimiento básico para planificar una solución de automatización

El presente capítulo contiene informaciones sobre las tareas básicas a tener en cuenta al estudiar una solución de automatización para un sistema de automatización (PLC). A la vista de un ejemplo se describe cómo proceder paso a paso para automatizar un proceso de mezcla industrial.

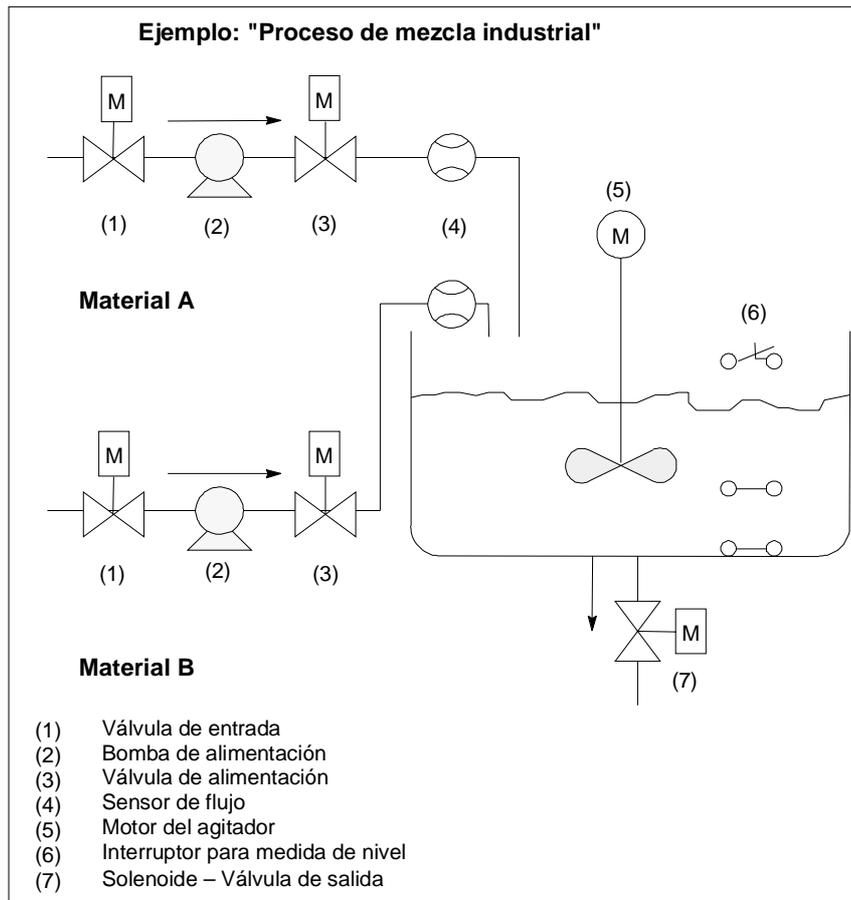
Existen muchos métodos para estudiar una solución de automatización. El procedimiento básico, que se puede utilizar para cualquier proyecto, se representa en la figura siguiente.



### 3.2 Subdividir el proceso en tareas y áreas

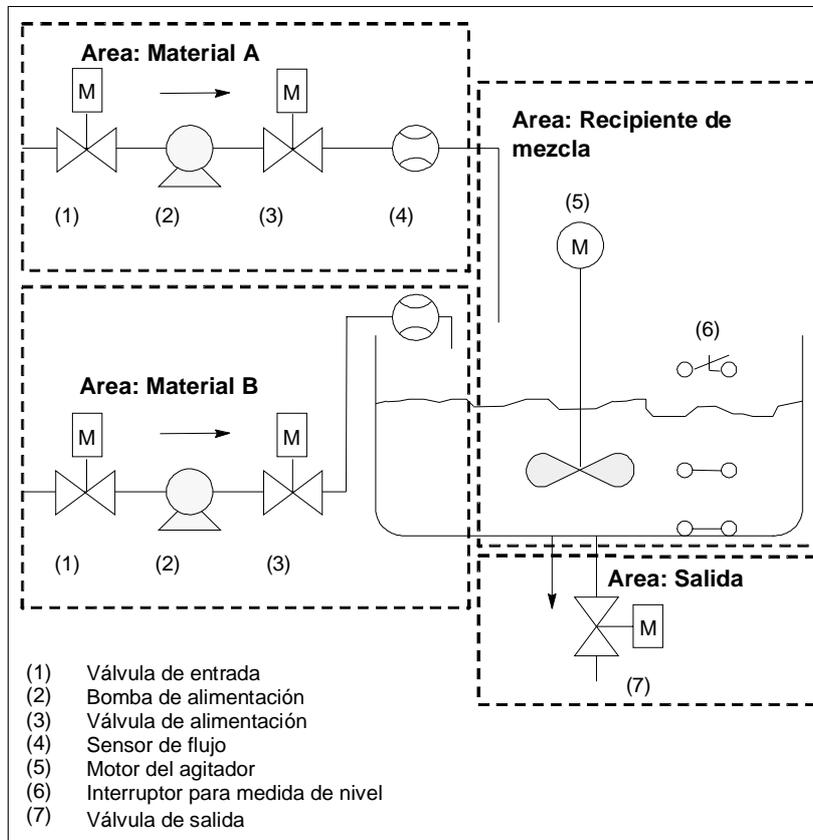
Un proceso de automatización siempre se divide en distintas tareas. Incluso el más complicado de los procesos puede ser definido, siempre y cuando se indique cómo están relacionadas las distintas tareas en las que se divide el proceso y se subdividan éstas en tareas más pequeñas.

El ejemplo siguiente ilustra, a la vista de un proceso de mezcla industrial, cómo estructurar un proceso en áreas funcionales y tareas.



## Definir las áreas de un proceso

Después de haber definido el proceso a controlar, éste se ha de subdividir en grupos o áreas relacionadas entre sí:



Subdividiendo cada área en tareas más pequeñas se simplifican las tareas de control.

En el ejemplo del proceso de mezcla industrial se pueden definir cuatro áreas (v. siguiente tabla). El área para el material A contiene los mismos aparatos que el área para el material B.

Área funcional	Aparatos correspondientes
Material A	Bomba de alimentación para material A Válvula de entrada para material A Válvula de alimentación para material A Sensor de flujo para material A
Material B	Bomba de alimentación para material B Válvula de entrada para material B Válvula de alimentación para material B Sensor de flujo para material B
Recipiente de mezcla	Motor del agitador Interruptor para medida de nivel
Salida	Válvula de salida

### 3.3 Describir el funcionamiento en diversas áreas

Al describir cada área y tarea de un proceso, se define no solamente el funcionamiento de cada área, sino también los diferentes elementos que controlan dicha área. Estos comprenden:

- entradas y salidas eléctricas, mecánicas y lógicas de cada tarea
- enclavamientos y dependencias entre las diferentes tareas

En el ejemplo del proceso de mezcla industrial se utilizan bombas, motores y válvulas. Estos se deben definir exactamente para determinar las características operativas y el tipo de enclavamientos que se requieren durante el funcionamiento. Las tablas siguientes contienen ejemplos de descripción de los aparatos que se utilizan en un proceso de mezcla industrial. Esta descripción puede utilizarse también para adquirir los aparatos necesarios.

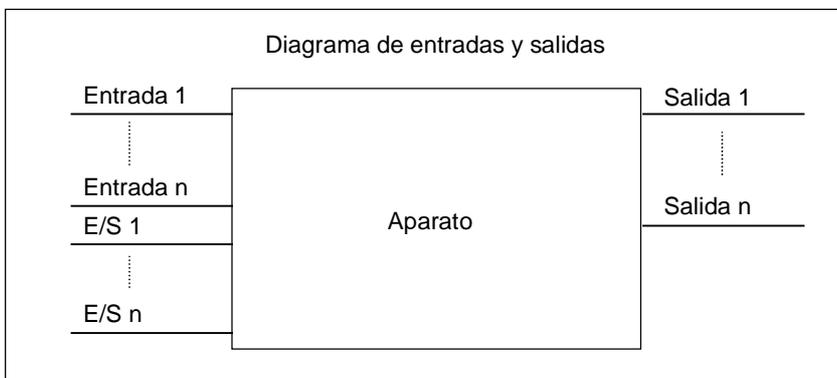
<b>Material A/B: motores para bombas de alimentación</b>
Las bombas de alimentación conducen los materiales A y B al recipiente de mezcla. <ul style="list-style-type: none"> <li>• caudal: 400 l por minuto</li> <li>• potencia: 100 KW a 1200 rpm</li> </ul>
Las bombas se controlan desde un panel de mando (Marcha/Paro), localizado en las cercanías del recipiente de mezcla. El número de puestas en marcha se cuenta para fines de mantenimiento. La puesta a 0 del contador y la cancelación del indicador de mantenimiento se realizan mediante un pulsador común.
Rigen las siguientes condiciones de habilitación: <ul style="list-style-type: none"> <li>• El recipiente de mezcla no está lleno.</li> <li>• La válvula de salida del recipiente de mezcla está cerrada.</li> <li>• El PARO DE EMERGENCIA no está activado.</li> </ul>
Rige la siguiente condición de desconexión: <ul style="list-style-type: none"> <li>• El sensor de flujo señala que no hay caudal 7 segundos tras arrancar el motor de la bomba.</li> <li>• Mientras funciona el motor, el sensor de flujo no señala circulación de material.</li> </ul>

<b>Material A/B: válvulas de entrada y alimentación</b>
Las válvulas de entrada y alimentación para los materiales A y B permiten/impiden la entrada de los materiales al recipiente de mezcla. Disponen de un solenoide con muelle antagonista. <ul style="list-style-type: none"> <li>• Cuando el solenoide está activo, está abierta la válvula.</li> <li>• Cuando el solenoide está desactivado, está cerrada la válvula.</li> </ul>
Las válvulas de entrada y alimentación son controladas por el programa de usuario.
Rige la siguiente condición de habilitación: <ul style="list-style-type: none"> <li>• El motor de la bomba de alimentación marcha por lo menos durante 1 segundo.</li> </ul>
Rige la siguiente condición de desconexión: <ul style="list-style-type: none"> <li>• El sensor de flujo no señala ningún caudal.</li> </ul>

<b>Motor del agitador</b>
Con el motor del agitador se mezclan los materiales A y B en el recipiente de mezcla. <ul style="list-style-type: none"> <li>• potencia: 100 KW a 1200 rpm</li> </ul>
El motor del agitador se controla desde un panel de mando (Marcha/Paro) localizado en las cercanías del recipiente de mezcla. El número de puestas en marcha se cuenta para fines de mantenimiento. La puesta a 0 del contador y la cancelación del indicador de mantenimiento se realizan mediante un pulsador común.
Rigen las siguientes condiciones de habilitación: <ul style="list-style-type: none"> <li>• El interruptor de nivel no señala "recipiente debajo del mínimo".</li> <li>• La válvula de salida del recipiente de mezcla está cerrada.</li> <li>• El PARO DE EMERGENCIA no está activado.</li> </ul>
Rige la siguiente condición de desconexión: <ul style="list-style-type: none"> <li>• El sensor de flujo no señala como muy tarde 10 s tras el arranque del motor que se ha alcanzado la velocidad nominal.</li> </ul>
<b>Válvula de salida</b>
La mezcla pasa por gravedad a la próxima fase del proceso. La válvula dispone de un solenoide con muelle antagonista. <ul style="list-style-type: none"> <li>• Cuando el solenoide está activado, está abierta la válvula de salida.</li> <li>• Cuando el solenoide está desactivado, está cerrada la válvula de salida.</li> </ul>
La válvula de salida se controla desde un panel de mando (abrir/cerrar).
La válvula de salida se puede abrir bajo las siguientes condiciones: <ul style="list-style-type: none"> <li>• El motor del agitador está desconectado.</li> <li>• El interruptor de nivel no señala "recipiente vacío".</li> <li>• El PARO DE EMERGENCIA no está activado.</li> </ul>
Rige la siguiente condición de desconexión: <ul style="list-style-type: none"> <li>• El sistema de medida de nivel señala "Recipiente vacío".</li> </ul>
<b>Interruptor para medida de nivel</b>
Los interruptores del recipiente de mezcla informan sobre el nivel de llenado del recipiente y se utilizan para el enclavamiento de las bombas de alimentación y del motor del agitador.

### 3.4 Listado de entradas y salidas

Después de haber definido físicamente cada uno de los aparatos a controlar, se han de dibujar diagramas de entradas y salidas para cada aparato o cada área de tareas.



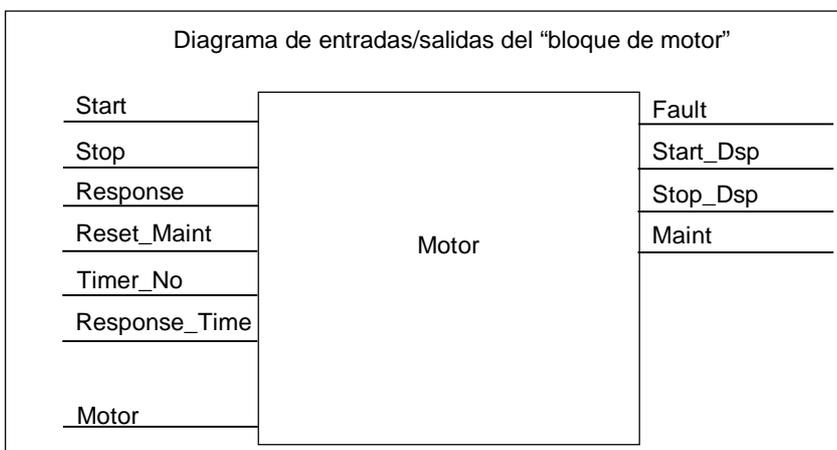
Estos diagramas equivalen a los bloques lógicos o de código que han de ser programados.

### 3.5 Crear un diagrama de E/S para los motores

En el ejemplo del proceso de mezcla industrial se utilizan dos bombas de alimentación y un agitador. El control de cada uno de los motores se efectúa mediante un "bloque de motor" idéntico para los tres aparatos. Este bloque necesita seis entradas: dos para marcha o paro, una para poner a 0 (Reset) el indicador de mantenimiento, una entrada para señalar el estado del motor (motor gira/detenido), una entrada para la temporización dentro de la cual debe llegar la señal de estado (respuesta) y una entrada para el número de temporizador que deberá utilizarse para medir el tiempo.

Además, el bloque lógico precisa cuatro salidas: dos para señalar el estado del motor, una para señalar errores y otra para señalar la necesidad de mantenimiento.

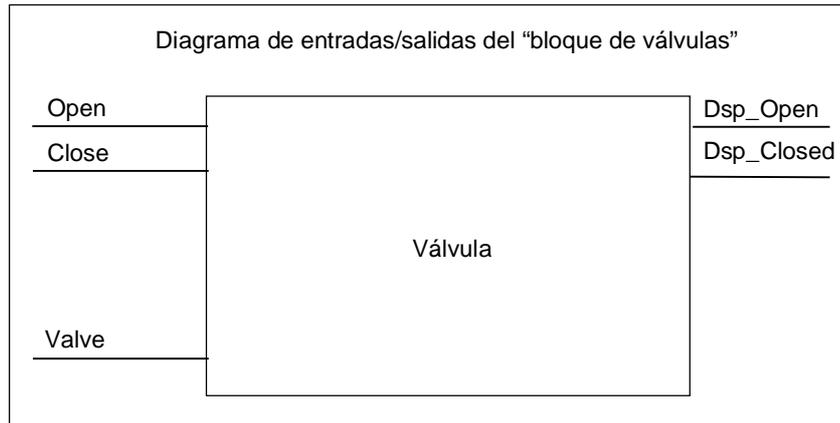
Además, se requiere una entrada/salida que sirve para controlar el motor, pero también simultáneamente para procesar o modificar el "bloque de motor" dentro del programa.



### 3.6 Crear un diagrama de E/S para las válvulas

Cada válvula se controla a través de un "bloque de válvulas" idéntico para todas las válvulas utilizadas. El bloque lógico dispone de dos entradas: una para abrir la válvula y otra para cerrarla. Además existen dos salidas: una para señalar que la válvula está abierta y otra para señalar que la válvula está cerrada.

El bloque dispone de una entrada/salida: sirve para controlar la válvula, pero también simultáneamente para procesar o modificar el "bloque de válvulas" dentro del programa.



### 3.7 Definir los requerimientos de seguridad

Determine - conforme a las prescripciones legales y a las directrices de procedimiento de la empresa - qué elementos se requieren para garantizar la seguridad del proceso. Describa también en qué medida influyen estos elementos de seguridad sobre las áreas del proceso.

#### Definir los requerimientos de seguridad

Definir los equipos que, por razones de seguridad, requieren circuitos fijamente cableados. Por definición, estos circuitos de seguridad trabajan independientemente del sistema de automatización (a pesar de que el circuito de seguridad ofrece normalmente un interface de entrada/salida para la coordinación con el programa de usuario). Usualmente se configura una matriz para conectar cada actuador con su propia área de PARO DE EMERGENCIA. Esta matriz constituye la base para los esquemas de los circuitos de seguridad.

Proceda de la siguiente manera al diseñar los dispositivos de protección:

- Definir los enclavamientos lógicos y mecánicos/eléctricos entre las diferentes tareas de automatización.
- Diseñar circuitos para poder manejar manualmente, en caso de emergencia, los aparatos integrantes del proceso.
- Definir otros requerimientos de seguridad para garantizar un desarrollo seguro del proceso.

#### Diseñar un circuito de seguridad

En el ejemplo del proceso de mezcla industrial se utiliza el siguiente circuito de seguridad:

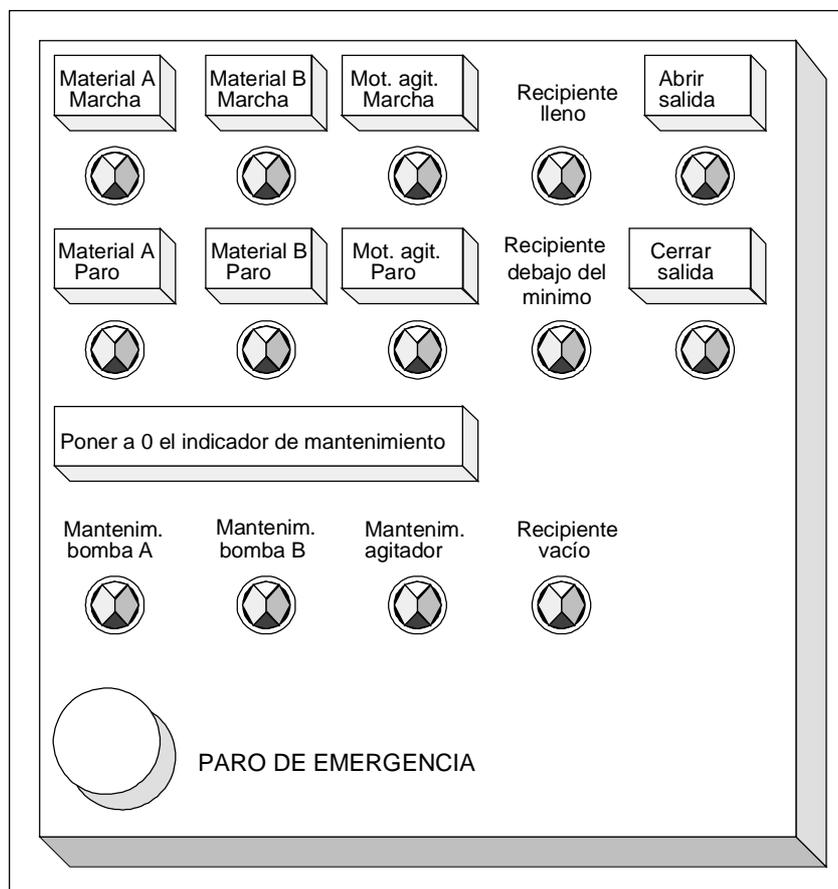
- Un pulsador de PARO DE EMERGENCIA que desconecta, independientemente del sistema de automatización (PLC), los aparatos siguientes:
  - Bomba de alimentación para material A
  - Bomba de alimentación para material B
  - Motor del agitador
  - Válvulas
- El pulsador de PARO DE EMERGENCIA está localizado en el panel de manejo.
- Una entrada del autómatas capta el estado del pulsador de PARO DE EMERGENCIA.

### 3.8 Describir los elementos necesarios para manejo y visualización

Cada proceso requiere un sistema de manejo y visualización que permita que las personas puedan intervenir en dicho proceso. Como parte de la descripción del proyecto se define también la estructura del panel de mando.

#### Definir un panel de mando

En el proceso de mezcla industrial descrito en nuestro ejemplo, cada aparato se pone en marcha o se para a través de un interruptor localizado en el panel de mando. Este panel de mando dispone de elementos de señalización que informan sobre el estado operativo (v. siguiente figura ).



Incluye también las lámparas de señalización para aquellos equipos que requieren mantenimiento tras una determinada cantidad de arranques, así como el pulsador de PARO DE EMERGENCIA para parar de inmediato el proceso en caso necesario. El panel incluye también un pulsador de puesta a "0" o rearme de los indicadores de mantenimiento de los tres motores. Con él se apagan las lámparas de señalización de mantenimiento del motor afectado y se pone a 0 el contador que controla los intervalos de mantenimiento.

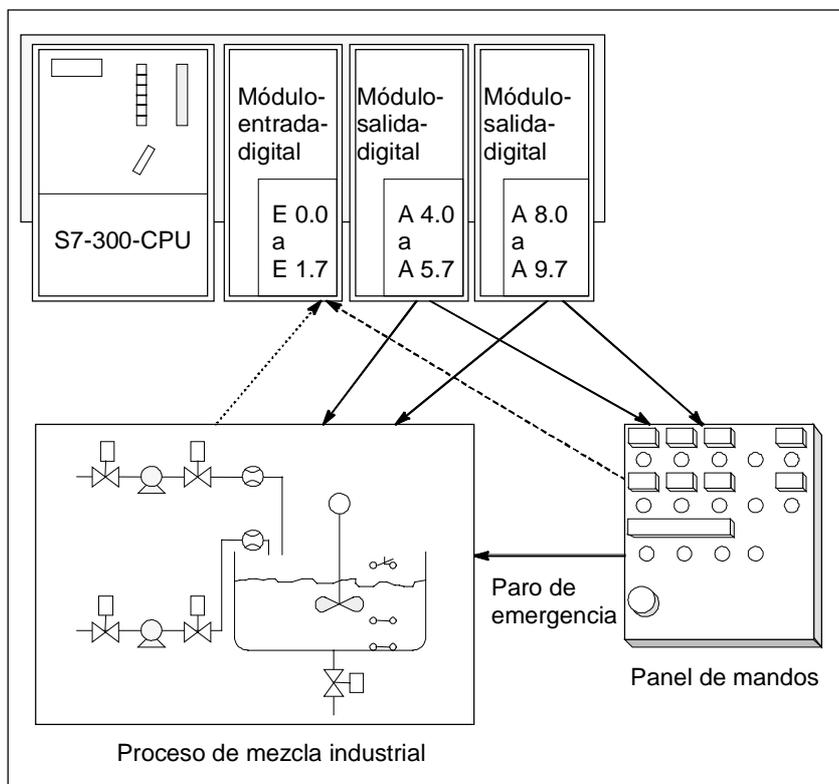
### 3.9 Crear un esquema de configuración

Después de haber documentado los requerimientos de diseño, se han de definir los equipos de control requeridos para este proyecto.

Al decidir qué módulos se han de utilizar, se define prácticamente la estructura del sistema de automatización. Crear un esquema de configuración considerando los puntos siguientes:

- tipo de la CPU
- cantidad y tipo de los módulos de señales
- configuración de las entradas y salidas físicas

La figura siguiente muestra un ejemplo de configuración S7 para el proceso de mezcla industrial.



## 4 Nociones básicas para diseñar la estructura del programa

### 4.1 Programas de una CPU

En una CPU se ejecutan principalmente dos programas diferentes:

- el sistema operativo y
- el programa de usuario.

#### Sistema operativo

El sistema operativo, que está integrado en las CPUs, organiza todas las funciones y procesos de la CPU que no están ligados a una tarea de control específica. Sus funciones son:

- gestionar el re arranque normal y el completo
- actualizar la imagen de proceso de las entradas y emitir la imagen de proceso de las salidas
- llamar el programa de usuario
- detectar las alarmas y llamar los OBs de tratamiento de alarmas
- detectar y tratar los errores
- administrar las áreas de memoria
- comunicar con unidades de programación y otras estaciones de comunicación

Modificando los parámetros del sistema operativo (preajustes) se puede controlar el comportamiento de la CPU en áreas determinadas.

#### Programa de usuario

El programa de usuario primero se ha de crear y luego se ha de cargar en la CPU. Contiene todas las funciones requeridas para procesar la tarea específica de automatización. Las tareas del programa de usuario son:

- definir las condiciones del re arranque completo y del re arranque normal de la CPU (p.ej. preestablecer un valor determinado para las señales)
- tratar datos del proceso (p.ej. efectuar combinaciones lógicas de señales binarias, leer y evaluar valores analógicos, definir señales binarias de salida, emitir valores analógicos)
- reaccionar a alarmas
- tratamiento de perturbaciones en el desarrollo normal del programa.

## 4.2 Bloques del programa de usuario

### 4.2.1 Bloques del programa de usuario

El software de programación STEP 7 permite estructurar el programa de usuario, es decir, subdividirlo en distintas partes. Esto aporta las siguientes ventajas:

- los programas de gran tamaño se pueden programar de forma clara
- se pueden estandarizar determinadas partes del programa
- se simplifica la organización del programa
- las modificaciones del programa pueden ejecutarse más fácilmente
- se simplifica el test del programa, ya que puede ejecutarse por partes
- se simplifica la puesta en servicio.

En el ejemplo del proceso de mezcla industrial ha podido apreciar cómo se puede subdividir un proceso de automatización en tareas individuales. Las partes de un programa de usuario estructurado equivalen a las distintas tareas y se definen como los bloques del programa.

### Tipos de bloques

En un programa de usuario S7 se pueden utilizar diversos tipos de bloques:

Bloque	Descripción breve de la función	Consulte también
Bloques de organización (OB)	Los OBs definen la estructura del programa de usuario.	"Bloques de organización y estructura del programa"
Bloques de función del sistema (SFBs) y funciones de sistema (SFCs)	Los SFBs y SFCs están integrados en la CPU S7, permitiéndole acceder a importantes funciones del sistema.	"Bloques de función de sistema (SFB) y funciones de sistema (SFC)"
Bloques de función (FB)	Los FBs son bloques con "memoria" que puede programar el mismo usuario.	"Bloques de función (FB)"
Funciones (FC)	Las FCs contienen rutinas de programa para funciones frecuentes.	"Funciones (FC)"
Bloques de datos de instancia (DBs de instancia)	Al llamarse a un FB/SFB, los DBs de instancia se asocian al bloque. Los DBs de instancia se generan automáticamente al efectuarse la compilación.	"Bloques de datos de instancia"
Bloques de datos (DB)	Los DBs son áreas de datos para almacenar los datos de usuario. Adicionalmente a los datos asociados a un determinado bloque de función, se pueden definir también datos globales a los que pueden acceder todos los bloques.	"Bloques de datos globales (DB)"

Los OBs, FBs, SFBs, FCs y SFCs contienen partes del programa, por lo que se denominan también bloques lógicos. El número permitido de bloques de cada tipo y su longitud admisible dependen de la CPU.

## 4.2.2 Bloques de organización y estructura del programa

Los bloques de organización (OB) constituyen el interface entre el sistema operativo y el programa de usuario. Son llamados por el sistema operativo y controlan el procesamiento cíclico y controlado por alarmas del programa, el comportamiento de arranque del sistema de automatización y el tratamiento de los errores. Programando los bloques de organización se define el comportamiento de la CPU.

### Prioridad de los bloques de organización

Los bloques de organización determinan la secuencia (eventos de arranque) en la que habrán de ejecutarse las diferentes partes del programa. La ejecución de un OB puede ser interrumpida por la llamada de otro OB. Qué OB puede interrumpir a otro OB depende de su prioridad. Los OBs de mayor prioridad pueden interrumpir a los de menor prioridad. La prioridad más baja la tiene el OB de tarea no prioritaria.

### Tipos de alarma y prioridades

Los eventos de arranque que provocan la llamada de un determinado OB se denominan también alarmas. La tabla siguiente muestra los tipos de alarma en STEP 7 y la prioridad de los bloques de organización asociados. No todos los bloques de organización y sus prioridades existen en todas las CPUs S7 (consulte los manuales "Sistema de automatización S7-300, Configuración, instalación y datos de las CPU" y "Sistemas de automatización S7-400, M7-400, Datos de los módulos").

Tipo de alarma	Bloque de organización	Prioridad (predeterminada)	Consulte también:
Ciclo libre	OB 1	1	"Bloque de organización para la ejecución cíclica del programa (OB 1)"
Alarmas horarias	OB 10 a OB 17	2	"Bloques de organización de alarma horaria (OB 10 a OB 17)"
Alarmas de retardo	OB 20	3	"Bloques de organización de alarma de retardo (OB 20 a OB 23)"
	OB 21	4	
	OB 22	5	
	OB 23	6	
Alarmas cíclicas	OB 30	7	"Bloques de organización de alarma cíclica (OB 30 a OB 38)"
	OB 31	8	
	OB 32	9	
	OB 33	10	
	OB 34	11	
	OB 35	12	
	OB 36	13	
	OB 37	14	
Alarmas de proceso	OB 40	16	"Bloques de organización de alarma de proceso (OB 40 a OB 47)"
	OB 41	17	
	OB 42	18	
	OB 43	19	
	OB 44	20	
	OB 45	21	
	OB 46	22	
	OB 47	23	

Tipo de alarma	Bloque de organización	Prioridad (predeterminada)	Consulte también:
Alarma de multiprocesamiento	OB 60 multiprocesamiento	25	Modo multiprocesador - Funcionamiento síncrono de varias CPUs
Error de redundancia	OB 70 Error de redundancia en periferia (sólo en sistemas H) OB 72 Error de redundancia en CPU (sólo en sistemas H) Error de redundancia de comunicación OB 73	25 28	"Bloques de organización para el tratamiento de errores (OB 70 a OB 87 / OB 121 a OB 122)"
Errores asíncronos	OB 80 Error de tiempo OB OB 81 Fallo de alimentación OB 82 Alarma de diagnóstico OB 83 Alarma de presencia de módulo OB 84 Fallo de CPU OB 85 Error de ejecución del programa OB 86 Fallo en el bastidor OB 87 Error de comunicación	26 (o 28, si el OB de error asíncrono existe en el programa de arranque)	"Bloques de organización para el tratamiento de errores (OB 70 a OB 87 / OB 121 a OB 122)"
Ciclo no prioritario	OB 90	29 <sup>1)</sup>	"Bloque de organización de tarea no prioritaria (OB 90)"
Arranque	OB 100 Rearranque completo OB 101 Rearranque OB 102 Arranque en frío	27 27 27	"Bloques de organización de arranque (OB 100 / OB 101 / OB 102)"
Errores síncronos	OB 121 Error de programación OB 122 Error de acceso a la periferia	Prioridad del OB que ha ocasionado el error	"Bloques de organización para el tratamiento de errores (OB 70 a OB 87 / OB 121 a OB 122)"
1) La prioridad 29 equivale a la prioridad 0.29. Por tanto, el ciclo de tarea no prioritaria tiene menor prioridad que el ciclo libre.			

## Cambiar la prioridad

Las alarmas se pueden parametrizar con STEP 7. Así es posible deseleccionar, por ejemplo, los OBs de alarma o cambiar las prioridades en los bloques de parámetros: alarmas horarias, alarmas de retardo, alarmas cíclicas y alarmas de proceso.

En las CPUs S7-300, la prioridad de los bloques de organización está asignada fijamente.

En las CPUs S7-400 (y en la CPU 318) se puede modificar con STEP 7 la prioridad de los siguientes bloques de organización:

- OB 10 a OB 47
- OB 70 a OB 72 (sólo CPUs H) y OB 81 a OB 87 en estado operativo RUN.

Son admisibles:

- para OB 10 a OB 47 las prioridades 2 a 23,
- para OB 70 a OB 72 las prioridades 2 a 28, y
- para OB 81 a OB 87 las prioridades 24 a 26.

Es posible asignar la misma prioridad a varios OBs. Los OBs de igual prioridad se ejecutan según el orden de aparición de sus eventos de arranque.

Los OBs de errores que arrancan cuando se producen errores síncronos se ejecutan con la misma prioridad que el bloque que se está ejecutando al detectarse el error.

## Datos locales

Al crear los bloques lógicos (OBs, FCs, FBs) se pueden definir datos locales temporales. El área de datos locales disponible en la CPU se divide entre las diferentes prioridades.

Las CPUs S7-400 permiten modificar la cantidad de datos locales por prioridades a través de STEP 7 en el bloque de parámetros "Prioridad".

## Información de arranque de un OB

Cada bloque de organización contiene una información de arranque de 20 bytes de datos locales, suministrada por el sistema operativo durante el arranque de un OB. La información de arranque informa sobre el evento de arranque del OB, la fecha y hora de arranque de OB, así como errores ocurridos y eventos de diagnóstico.

La información de arranque del OB 40 de alarma de proceso indica, por ejemplo, en la información de arranque la dirección del módulo que ha causado la alarma.

## OBs de alarma desactivados

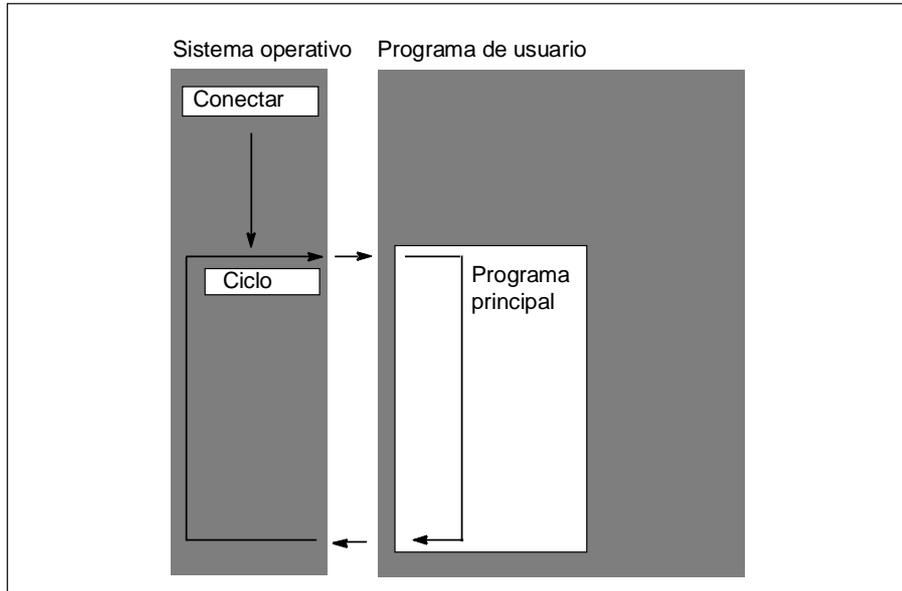
Eligiendo la prioridad 0 o asignando menos de 20 bytes de datos locales a una prioridad se desactiva el correspondiente OB de alarma. Los OBs de alarma desactivados:

- no se pueden copiar en el estado operativo RUN ni integrar en el programa de usuario.
- se pueden copiar en el estado operativo "STOP" e integrar en el programa de usuario, pero en caso de re arranque completo de la CPU no provocan una interrupción del arranque y generan una entrada en el búfer de diagnóstico.

Desactivando los OBs de alarma no requeridos se incrementa el área disponible para datos locales que se puede utilizar para memorizar datos temporales de otras prioridades.

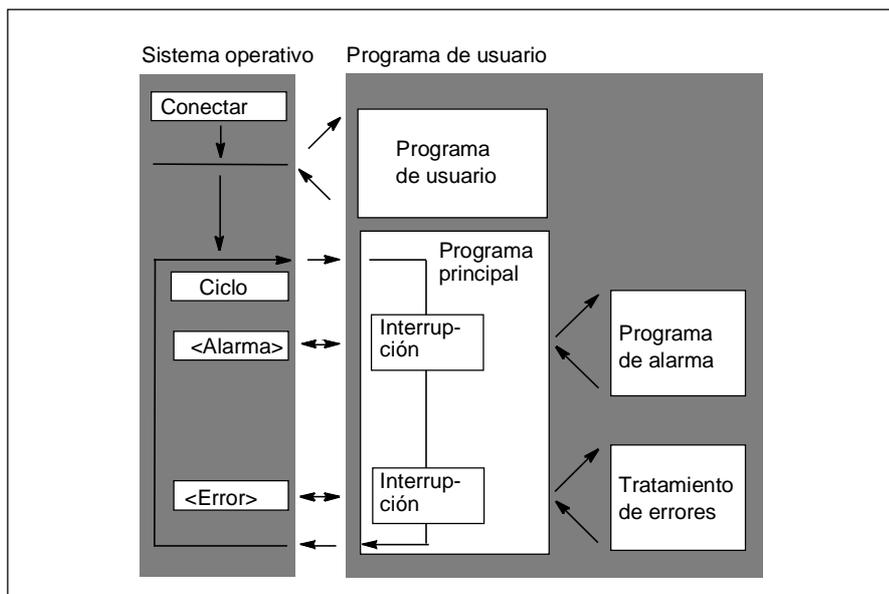
### Ejecución cíclica de programas

La ejecución cíclica de programas es la ejecución "normal" en autómatas programables, es decir, el sistema operativo se ejecuta en un bucle llamado ciclo. Cada vez que se recorre un ciclo, el sistema operativo llama al bloque de organización OB 1 en el programa principal. Por consiguiente, el programa de usuario se trata cíclicamente en el OB 1.



### Ejecución del programa controlada por alarmas

La ejecución cíclica del programa puede ser interrumpida por determinados eventos de arranque (alarmas). Si se presenta un evento de arranque tal, se interrumpe el bloque que está en tratamiento en el límite de una instrucción y se procesa el bloque de organización asignado al evento de arranque. Luego se continúa ejecutando el programa cíclico a partir del punto de interrupción.

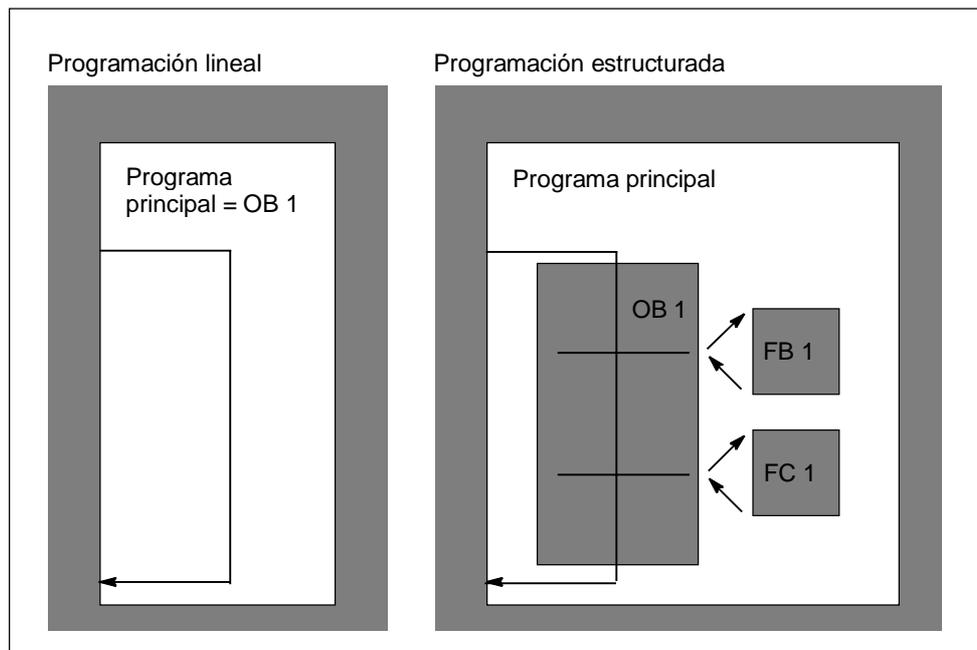


De este modo existe la posibilidad de ejecutar sólo en caso necesario aquellas partes del programa de usuario que no deben procesarse cíclicamente. El programa de usuario se puede dividir en subprogramas y repartir en diferentes bloques de organización. Si el programa de usuario debe reaccionar a una señal importante que se presente con poca frecuencia (p.ej., si el indicador de nivel de un depósito indica que se ha alcanzado el nivel de llenado), el subprograma que se deba ejecutar cuando se emita la señal se puede depositar en un OB que se ejecute de forma controlada por eventos.

### Programación lineal o estructurada

El programa de usuario completo se puede escribir en el OB 1 (programación lineal). Esto se recomienda únicamente cuando los programas son simples y se ejecutan en las CPUs del S7-300 con poco volumen de memoria.

Las funciones complejas de automatización se pueden procesar mejor si se dividen en tareas más pequeñas que correspondan a las funciones tecnológicas del proceso de automatización o si se deben utilizar frecuentemente. Estas tareas parciales están representadas (programación estructurada) en el programa de usuario mediante bloques.



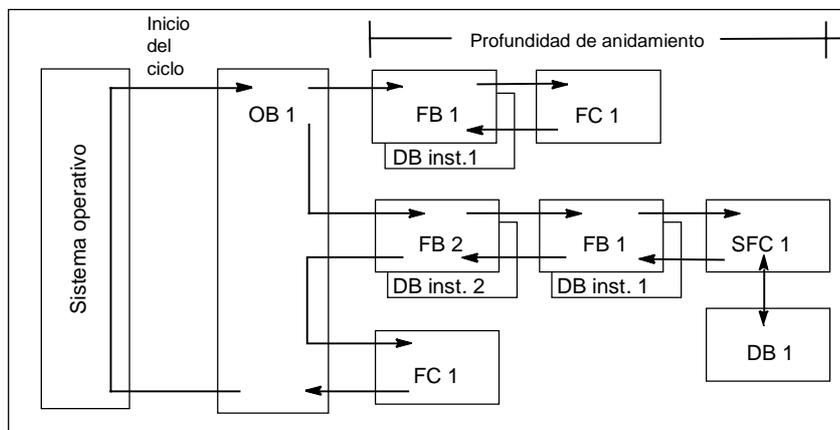
### 4.2.3 Jerarquía de llamada en el programa de usuario

Para que el programa de usuario pueda funcionar, los bloques que lo componen se deben poder llamar. Esto se efectúa mediante las llamadas de bloques, siendo éstas operaciones especiales de STEP 7 que sólo se pueden programar e iniciar en bloques lógicos.

#### Secuencia y profundidad de anidamiento

La secuencia y el anidamiento de las llamadas de bloques se denomina jerarquía de llamadas. La profundidad de anidamiento admisible depende del tipo de CPU.

La figura siguiente muestra, a la vista de un ejemplo, la secuencia y la profundidad de anidamiento de las llamadas de bloques dentro de un ciclo de ejecución.



Orden de creación de bloques:

- Los bloques se crean de arriba hacia abajo, es decir, que se comienza con la fila superior de bloques.
- Cada bloque que se llame ya deberá existir. Por tanto, en una fila de bloques, el orden de creación deberá ser de derecha a izquierda.
- El OB 1 es el último bloque que se crea.

Conforme a dichas reglas, el orden de creación de los bloques de la figura de ejemplo sería el siguiente:

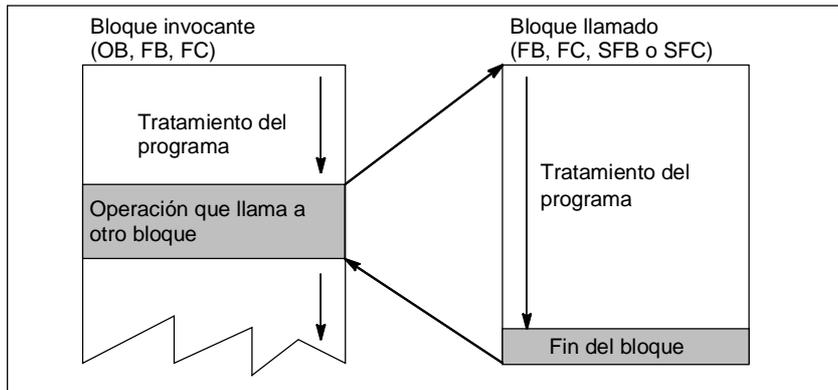
FC 1 > FB 1 + DB 1 de instancia > DB 1 > SFC 1 > FB 2 + DB 2 de instancia > OB 1

#### Nota

Si la profundidad de anidamiento es excesiva, la pila de datos locales puede desbordarse (consulte también Pila de datos locales).

## Llamadas de bloques

La figura siguiente ilustra el desarrollo de la llamada de un bloque dentro del programa de usuario: el programa llama el segundo bloque, cuyas operaciones se ejecutan luego hasta el final. Finalizado el tratamiento del bloque llamado, se continúa el procesamiento del bloque invocante con la operación subsiguiente a la llamada del bloque.



Antes de programar un bloque, se debe definir previamente con qué datos ha de efectuarse la ejecución del programa: se han de declarar las variables del bloque.

---

### Nota

Los parámetros OUT se deben describir en cada llamada de bloques.

---

### Nota

El sistema operativo inicializa las instancias del SFB 3 "TP" al efectuarse un arranque en frío. Si las instancias de ese SFB se deben inicializar tras un re arranque (arranque en caliente), dichas instancias a inicializar se deberán llamar con PT = 0 ms a través del OB 100. Esto se puede llevar a cabo p.ej. mediante una rutina de inicialización en los bloques que contengan instancias de dicho SFB.

---

## 4.2.4 Tipos de bloques y ejecución cíclica del programa

### 4.2.4.1 Bloque de organización para la ejecución cíclica del programa (OB 1)

La ejecución cíclica de programas es la ejecución "normal" en los sistemas de automatización. El sistema operativo llama cíclicamente al OB 1 y arranca la ejecución cíclica del programa de usuario.

#### Ejecución cíclica del programa

La tabla siguiente muestra las fases de la ejecución cíclica del programa:

Paso	Secuencia en las CPUs antiguas	Secuencia en las CPUs nuevas (a partir de 10/98)
1.	El sistema operativo inicia el tiempo de vigilancia del ciclo.	El sistema operativo inicia el tiempo de vigilancia del ciclo.
2.	La CPU lee el estado de las entradas en los módulos de entradas y actualiza la imagen de proceso de las entradas.	La CPU escribe los valores de la imagen de proceso de las salidas en los módulos de salida.
3.	La CPU ejecuta el programa de usuario y las operaciones indicadas en dicho programa.	La CPU lee el estado de las entradas en los módulos de entradas y actualiza la imagen de proceso de las entradas.
4.	La CPU escribe los valores de la imagen de proceso de las salidas en los módulos de salida.	La CPU ejecuta el programa de usuario y las operaciones indicadas en dicho programa.
5.	Al final del ciclo, el sistema operativo realiza las tareas pendientes, p.ej. cargar y borrar bloques, recibir y enviar datos globales.	Al final del ciclo, el sistema operativo realiza las tareas pendientes, p.ej. cargar y borrar bloques, recibir y enviar datos globales.
6.	Finalmente, la CPU regresa al principio del ciclo y arranca nuevamente la vigilancia del tiempo de ciclo.	Finalmente, la CPU regresa al principio del ciclo y arranca nuevamente la vigilancia del tiempo de ciclo.

#### Imágenes del proceso

Para garantizar que la CPU disponga de una imagen coherente de las señales del proceso durante la ejecución cíclica del programa, al activarse las áreas de operandos entradas (E) y salidas (A), la CPU no accede directamente a los módulos de señales, sino a un área de memoria interna de la CPU que contiene una imagen de las entradas/salidas.

#### Programar ejecución cíclica del programa

La ejecución cíclica se programa escribiendo el programa de usuario con STEP 7 en el OB 1 y en los bloques allí llamados.

La ejecución cíclica del programa comienza tan pronto el programa de arranque sea finalizado sin errores.

## Posibilidades de interrupción

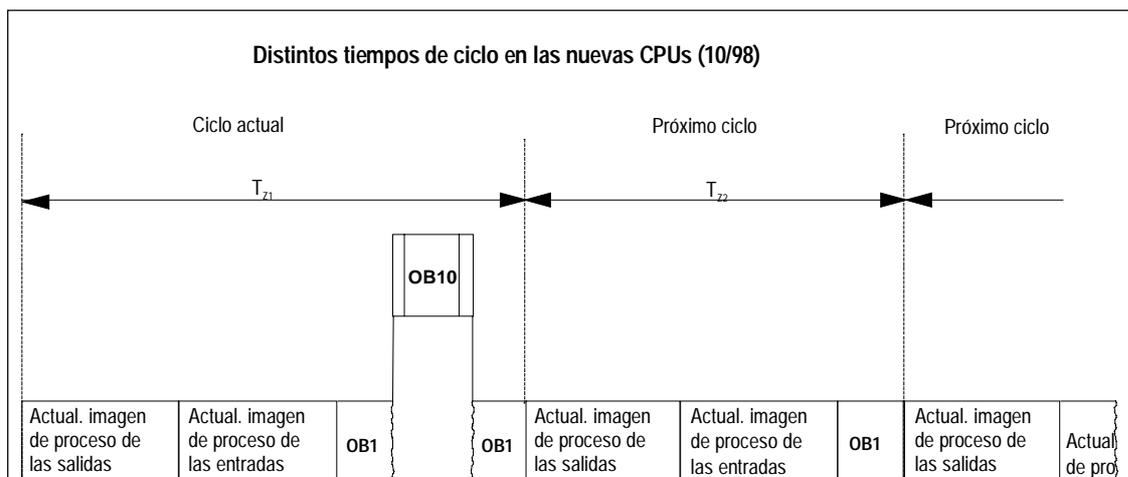
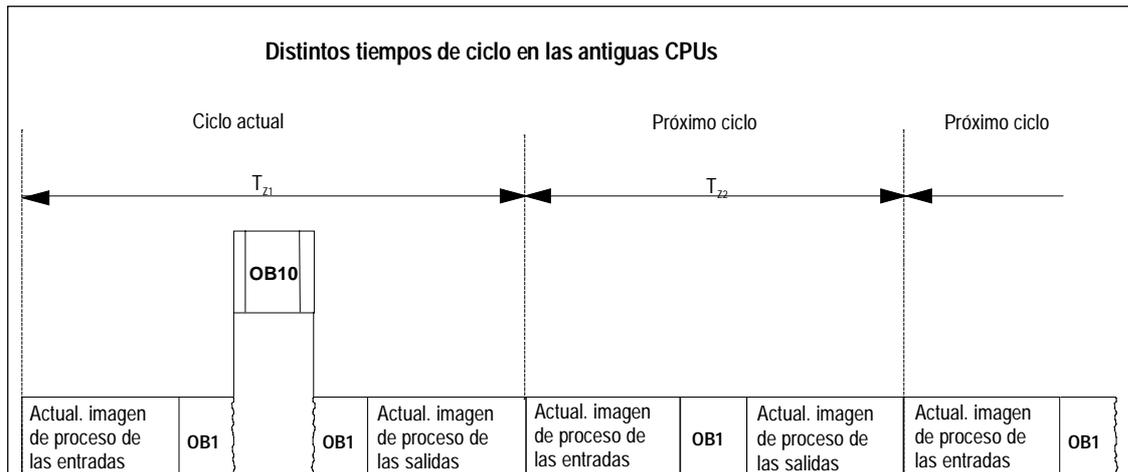
La ejecución cíclica del programa puede ser interrumpida por:

- una alarma
- una orden STOP (selector de modo de operación, comando de menú desde la PG, SFC 46 STP, SFB 20 STOP)
- un corte de tensión de red (alimentación)
- el fallo de un aparato o por un error del programa

## Tiempo de ciclo

El tiempo de ciclo es el tiempo que el sistema operativo necesita para ejecutar el programa cíclico, así como todas las partes del programa que interrumpen dicho ciclo (p.ej. la ejecución de otros bloques de organización) y las actividades del sistema (p.ej. la actualización de las imágenes del proceso). Este tiempo es vigilado por el sistema.

El tiempo de ciclo (TZ) no es igual para cada ciclo. Las siguientes figuras muestran los diversos tiempos de ciclo ( $TZ1 \neq TZ2$ ) para las CPUs antiguas y nuevas.



En el ciclo actual, el OB 1 es interrumpido por una alarma horaria.

### Tiempo de ciclo máximo

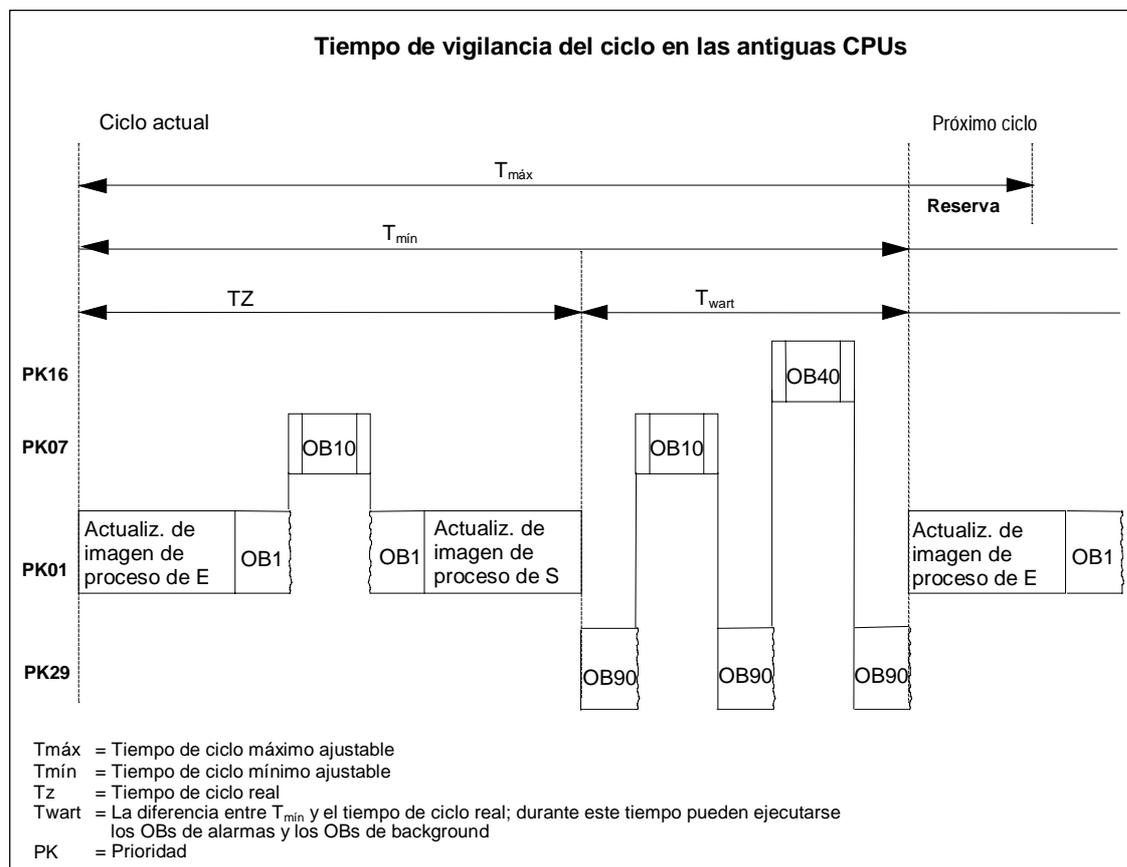
Con STEP 7 se puede modificar el tiempo de ciclo máximo preajustado. Transcurrido este tiempo, la CPU pasa a STOP o se llama el OB 80, en el cual puede definirse cómo debe reaccionar la CPU al error de tiempo.

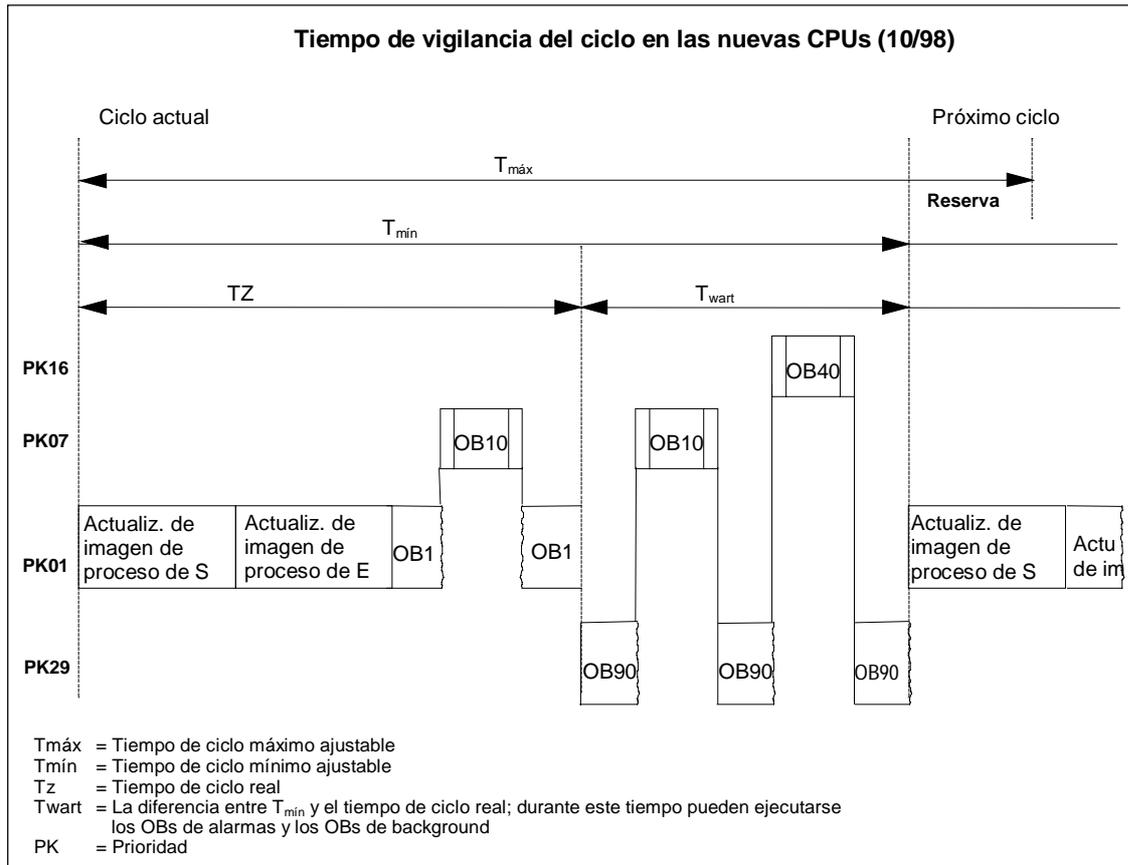
### Tiempo de ciclo mínimo

Con STEP 7 se puede ajustar un tiempo de ciclo mínimo para las CPUs S7-400 y para la CPU 318. Esto se recomienda, cuando

- los intervalos entre los arranques de la ejecución del programa del OB 1 (ciclo libre) deban ser iguales o
- el tiempo de ciclo es muy corto, para evitar que las imágenes del proceso se actualicen innecesariamente con demasiada frecuencia.

Las siguientes figuras muestran la función del tiempo de vigilancia del ciclo en la ejecución del programa de las CPUs antiguas y nuevas.





## Actualización de la imagen del proceso

En la ejecución cíclica del programa de la CPU la imagen del proceso se actualiza automáticamente. En las CPUs S7-400 y en la CPU 318 se puede deseleccionar la actualización de la imagen del proceso si se desea

- acceder directamente a la periferia o
- actualizar en otro momento una o varias imágenes de proceso de las entradas y las salidas a través de las funciones del sistema SFC 26 UPDAT\_PI y SFC 27 UPDAT\_PO.

## Carga por comunicaciones

Cion el parámetro de CPU "Carga del ciclo por comunicaciones" puede controlar dentro de cierto margen la duración de los procesos de comunicación, que a su vez siempre prolongan el tiempo de ciclo. Pueden ser procesos de comunicación, p.ej: la transferencia de datos a otra CPU vía MPI o la carga de bloques activada mediante PG.

Este parámetro apenas influye en las funciones de test con la PG, pero pueden prolongar considerablemente el tiempo de ciclo. El tiempo disponible para funciones de test puede limitarse en el proceso (sólo S7-300).

## Efecto del parámetro

El sistema operativo de la CPU pone permanentemente a disposición de la comunicación el porcentaje configurado de la capacidad total de procesamiento de la CPU (técnica de segmentación de tiempos). Si esta capacidad de procesamiento no se necesita para la comunicación, queda disponible para otras labores de procesamiento.

## Efecto sobre el tiempo de ciclo real

Si no hay eventos asíncronos adicionales, el tiempo de ciclo del OB 1 se prolonga en un factor que puede calcularse con la siguiente fórmula:

$$\frac{100}{100 - \text{"Carga del ciclo por comunicaciones (\%)\"}}$$

Ejemplo 1 (sin eventos asíncronos adicionales):

Si la carga del ciclo por comunicaciones se ajusta al 50%, el tiempo de ciclo del OB 1 puede duplicarse.

Simultáneamente, los eventos asíncronos (p.ej. alarmas de proceso o alarmas cíclicas) influyen en el tiempo de ciclo del OB1. Estadísticamente hablando, la prolongación del tiempo de ciclo debida a la parte de comunicaciones hace que se produzcan más eventos asíncronos dentro de un ciclo de OB1, lo que prolonga adicionalmente el tiempo de ciclo del OB1. Esta prolongación depende del número de eventos que se produzcan por cada ciclo del OB1 y de la duración del procesamiento del evento.

Ejemplo 2 (considerando eventos asíncronos adicionales):

Si el tiempo de ejecución del OB1 es 500 ms, una carga por comunicaciones del 50 % puede producir un tiempo de ciclo real hasta de 1000 ms (presuponiendo que la CPU tenga siempre suficientes peticiones de comunicación que procesar). Si paralelamente cada 100 ms se activa una alarma cíclica de 20 ms de tiempo de procesamiento, la repercusión sobre el ciclo sin carga por comunicaciones sería en total  $5 \times 20 \text{ ms} = 100 \text{ ms}$ ; es decir, el tiempo real de ciclo sería 600 ms. Dado que una alarma cíclica también interrumpe la comunicación, con una carga por comunicaciones del 50 % la repercusión sobre el tiempo de ciclo es  $10 \times 20 \text{ ms}$ ; es decir, en este caso el tiempo real de ciclo no es 1000 ms, sino 1200 ms.

## Notas

- Compruebe las repercusiones de un cambio de valor del parámetro "Carga del ciclo por comunicaciones" con la instalación en funcionamiento.
- Al ajustarse el tiempo de ciclo mínimo es preciso tener en cuenta la carga por comunicaciones, puesto que de lo contrario se producirán errores de tiempo.

## Recomendaciones

- Si es posible, adopte el valor predeterminado
- Aumente el valor predeterminado sólo si la finalidad principal de la CPU es la comunicación y el programa de usuario no es crítico en el tiempo.
- En todos los demás casos, a lo sumo reduzca el valor predeterminado.
- Ajuste el proceso (sólo con S7-300) y restrinja el tiempo necesario en el mismo para las funciones de test.

#### 4.2.4.2 Funciones (FC)

Las funciones son bloques programables. Una función es un bloque lógico "sin memoria". Las variables temporales de las FCs se memorizan en la pila de datos locales. Estos datos se pierden tras el tratamiento de las FCs. Para fines de memorización de datos, las funciones pueden utilizar bloques de datos globales.

Como una FC no tiene asignada ninguna memoria, se han de indicar siempre parámetros actuales. A los datos locales de una FC no se pueden asignar valores iniciales.

#### Campo de aplicación

La FC contiene un programa que se ejecuta cada vez que la FC es llamada por otro bloque lógico. Las funciones se pueden utilizar para

- devolver un valor de función al bloque invocante (ejemplo: funciones matemáticas)
- ejecutar una función tecnológica (ejemplo: control individual con combinación binaria).

#### Asignación de parámetros actuales a parámetros formales

El parámetro formal es un comodín para el parámetro real, es decir, el parámetro actual. Los parámetros actuales sustituyen a los parámetros formales al efectuar la llamada a una FC. A los parámetros formales de una FC se han de asignar siempre parámetros actuales (p.ej. al parámetro formal "Start" un parámetro actual "E3.6"). Los parámetros de entrada, de salida y de entrada/salida utilizados por la FC se depositan en forma de punteros en los parámetros actuales del bloque lógico que ha llamado a la FC.

#### 4.2.4.3 Bloques de función (FB)

Los bloques de función son bloques programables. Un FB es un bloque "con memoria". Dispone de un bloque de datos asignado como memoria (bloque de datos de instancia). Los parámetros que se transfieren al FB, así como las variables estáticas, se memorizan en el DB de instancia. Las variables temporales se memorizan en la pila de datos locales.

Los datos memorizados en el DB de instancia no se pierden al concluir el tratamiento del FB. Los datos memorizados en la pila de datos locales se pierden al concluir el tratamiento del FB.

---

#### Nota

Con objeto de evitar errores al trabajar con FBs, lea el tema de la Ayuda Tipos de datos admisibles al transferir parámetros.

---

#### Campo de aplicación

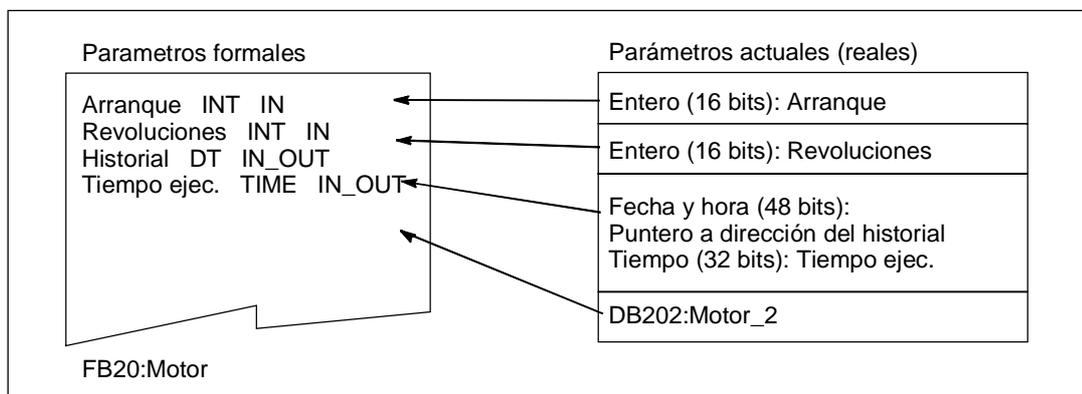
Un FB contiene un programa que se ejecuta siempre cuando el FB es llamado por otro bloque lógico. Los bloques de función simplifican la programación de funciones complejas de uso frecuente.

## FBs y DBs de instancia

A cada llamada de un bloque de función que transfiere parámetros está asignado un bloque de datos de instancia.

Mediante la llamada de varias instancias de un FB es posible controlar varios equipos con un FB. Un FB para un tipo de motor puede controlar, por ejemplo, diferentes motores, utilizando datos de instancia diferentes para los diferentes motores. Los datos para cada motor (tales como número de revoluciones, rampas, tiempo de funcionamiento acumulado, etc.) se pueden memorizar en uno o varios DBs de instancia.

La figura siguiente muestra los parámetros formales de un FB que utiliza los parámetros actuales. Los parámetros formales están memorizados en el DB de instancia.



## Variable del tipo de datos FB

Si el programa de usuario está estructurado de tal manera que en un FB se puedan llamar bloques de función ya existentes, los FBs a llamar se pueden incluir como variables estáticas del tipo de datos FB en la tabla de declaración de variables del FB invocante. Esto permite anidar las variables y concentrar los datos en un bloque de datos de instancia (multiinstancia).

## Asignación de parámetros actuales a parámetros formales

En general, en STEP 7 no es necesario asignar parámetros actuales al parámetro formal de un FB. No obstante, hay ciertas excepciones. Los parámetros actuales se han de asignar:

- a los parámetros de entrada/salida de un tipo de datos compuestos (p.ej. STRING, ARRAY o DATE\_AND\_TIME)
- a todos los tipos de parámetros (p.ej. TIMER, COUNTER o POINTER)

STEP 7 asigna los parámetros actuales a los parámetros formales de un FB de la siguiente manera:

- *Si se indican parámetros actuales en la instrucción de llamada:* Las operaciones del FB utilizan los parámetros actuales suministrados.
- *Si en la instrucción de llamada no se indica ningún parámetro actual:* Las operaciones del FB utilizan los valores memorizados en el DB de instancia.

La tabla siguiente muestra qué variables del parámetro actual FB se deben asignar.

Variables	Tipo de datos		
	Tipo de datos simple	Tipo de datos compuesto	Tipo de parámetro
Entrada	Parámetro no necesario	Parámetro no necesario	Parámetro actual necesario
Salida	Parámetro no necesario	Parámetro no necesario	Parámetro actual necesario
Entrada/salida	Parámetro no necesario	Parámetro actual necesario	–

### Asignación de valores iniciales a parámetros formales

En el área de declaración del FB se pueden asignar valores iniciales a los parámetros formales. Estos datos se incluyen en el DB de instancia asignado al FB.

Si en la instrucción de llamada no se asignan parámetros actuales a los parámetros formales, entonces STEP 7 utiliza los valores memorizados en el DB de instancia. Estos datos pueden ser valores iniciales, que han sido indicados en la tabla de declaración de variables de un FB.

La tabla siguiente muestra qué variables se pueden asignar a un valor inicial. Como los datos temporales no se memorizan tras el tratamiento del bloque, entonces no se les puede asignar valores.

Variables	Tipo de datos		
	Tipo de datos simple	Tipo de datos compuesto	Tipo de parámetro
Entrada	Valor inicial admisible	Valor inicial admisible	–
Salida	Valor inicial admisible	Valor inicial admisible	–
Entrada/salida	Valor inicial admisible	–	–
Estáticas	Valor inicial admisible	Valor inicial admisible	–
Temporales	–	–	–

#### 4.2.4.4 Bloques de datos de instancia

A cada llamada de un bloque de función que transfiere parámetros está asignado un bloque de datos de instancia. En el DB de instancia están depositados los parámetros actuales y los datos estáticos del FB. Las variables declaradas en el FB definen la estructura del bloque de datos de instancia. La instancia define la llamada de un bloque de función. Si, por ejemplo, un bloque de función se llama cinco veces en el programa de usuario S7, existen cinco instancias de dicho bloque.

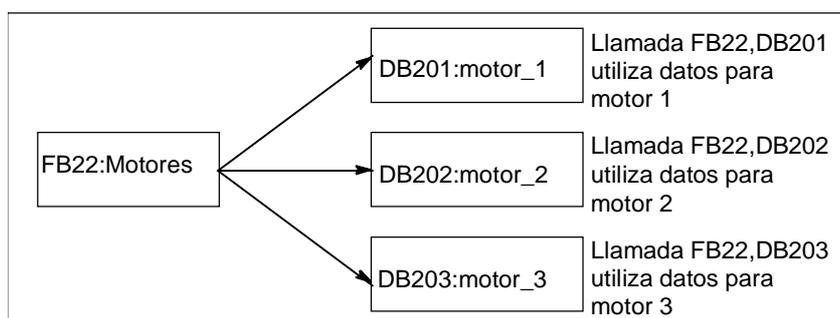
#### Crear un DB de instancia

Antes de crear un bloque de datos de instancia debe existir el FB asociado. El número de dicho FB se debe indicar al crear el bloque de datos de instancia.

#### Un DB de instancia para cada instancia

Si se asignan varios bloques de datos de instancia a un bloque de función (FB) que controla un motor, se puede utilizar este FB para controlar varios motores.

Los diversos datos de cada uno de los motores (p.ej. número de revoluciones, tiempo de aceleración, tiempo total de servicio) se memorizan en los diversos bloques de datos. Dependiendo de qué DB se asigne al FB al efectuar la llamada, se puede controlar un motor diferente. De esta manera se utiliza un solo bloque de función para varios motores (v. siguiente figura).

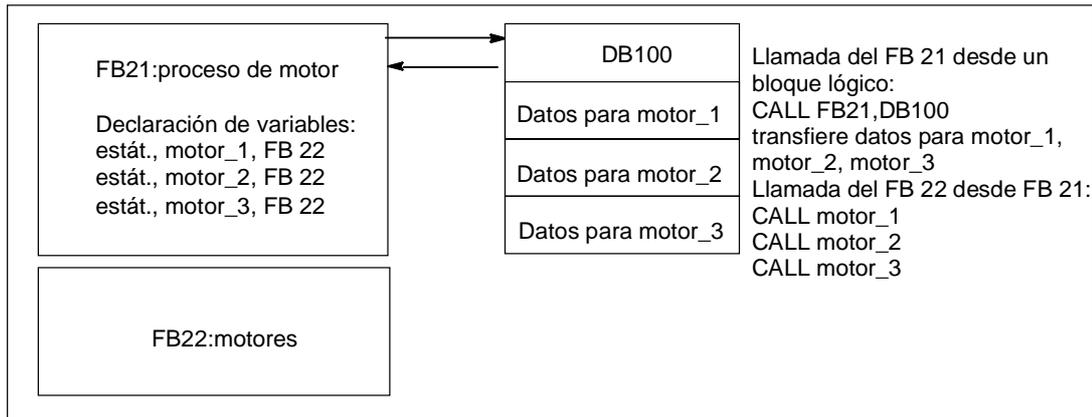


#### Un DB de instancia para varias instancias de un FB (multiinstancias)

A un FB se pueden transferir conjuntamente en un DB de instancia los datos de instancia para diferentes motores. A tal efecto, la llamada de los controles de motores se ha de efectuar en otro FB y en el área de declaración del FB invocante se deben declarar las variables estáticas con el tipo de datos de un FB para las diferentes instancias.

Utilizando un DB de instancia para varias instancias de un FB se ahorra capacidad de memoria y optimiza el uso de los bloques de datos.

En el caso ejemplificado por la figura siguiente, el bloque invocante es el FB 21 "Proceso de motor", las variables son del tipo de datos FB 22 y las instancias son designadas con motor\_1, motor\_2 y motor\_3.



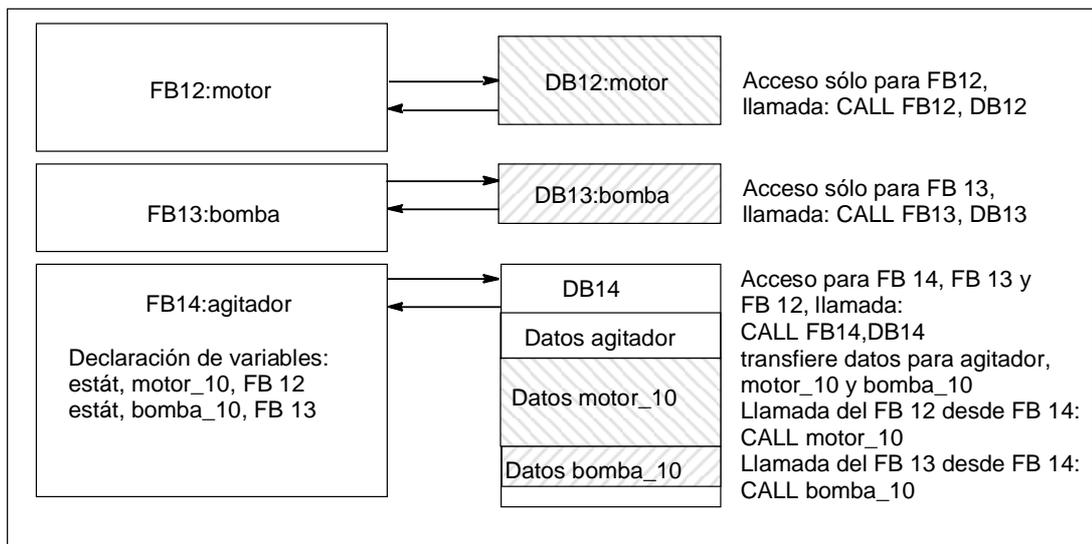
En este ejemplo, el FB 22 no necesita bloque de datos de instancia propio, ya que sus datos de instancia están memorizados en el bloque de datos de instancia del FB invocante.

### Un DB de instancia para varias instancias de FB diferentes (multiinstancias)

En un bloque de función se pueden llamar a instancias de otros FBs ya existentes. Los datos de instancia necesarios al respecto se pueden asignar al bloque de datos de instancia del FB invocante, es decir que en este caso no se necesitan bloques de datos adicionales para los FBs que se han llamado.

Para dichas multiinstancias de un DB de instancia deberá declarar, en la tabla del FB invocante, variables estáticas del mismo tipo de datos del FB llamado, haciéndolo para cada una de las instancias. La llamada en el FB se efectúa entonces sólo con el nombre de la variable, es decir, sin indicar un DB de instancia.

En el ejemplo de la figura, los datos de instancia asignados se memorizan conjuntamente en un DB de instancia.



#### 4.2.4.5 Bloques de datos globales (DB)

Al contrario de los bloques lógicos, los bloques de datos no contienen instrucciones STEP 7. En cambio, sirven para depositar datos de usuario, es decir que los bloques de datos contienen datos variables con los que trabaja el programa de usuario. Los bloques de datos globales contienen datos de usuario utilizables desde otros bloques.

El tamaño de los DBs puede variar. El tamaño máximo admisible se indica en las descripciones de la CPUs /70/ y /101/.

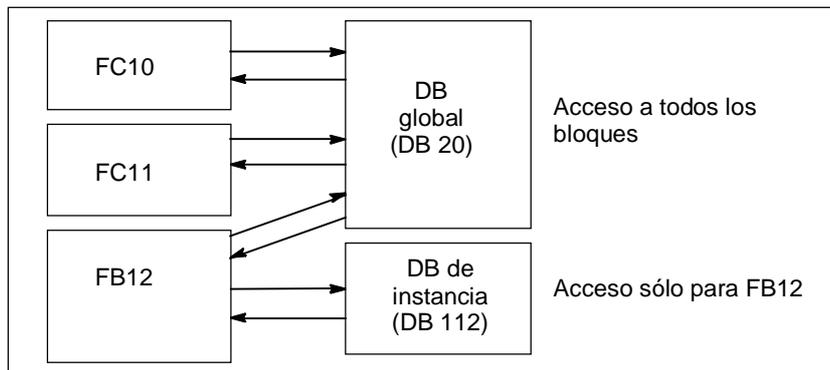
La estructura de bloques de datos globales se puede definir discrecionalmente.

#### Bloques de datos globales en el programa de usuario

Si se llama un bloque lógico (FC, FB o OB), éste puede ocupar determinada capacidad de memoria en el área de datos locales (pila L). Además de este área de datos locales, un bloque lógico puede abrir un área de memoria en forma de un DB. Al contrario de los datos en el área de datos locales, los datos contenidos en un DB no son borrados al cerrar el DB o al concluir el tratamiento del correspondiente bloque lógico.

Cada FB, FC o OB puede leer los datos de un DB global o escribir datos en un DB global. Estos datos se conservan en el DB incluso al abandonar dicho DB.

Un DB global y un DB de instancia pueden estar abiertos al mismo tiempo. La figura siguiente ilustra diferentes accesos a bloques de datos.



#### 4.2.4.6 Bloques de función de sistema (SFB) y funciones del sistema (SFC)

##### Bloques preprogramados

No es necesario programar cada función. Las CPUs S7 ofrecen bloques preprogramados que se pueden llamar desde el programa de usuario.

Para obtener informaciones más detalladas, consulte los temas de Ayuda de los bloques del sistema y las funciones del sistema (saltos a descripciones de lenguajes, ayudas acerca de bloques, atributos del sistema).

##### Bloques de funciones del sistema

Un SFB es un bloque de funciones integrado en la CPU S7. Como los SFBs forman parte del sistema operativo, no se cargan como parte integrante del programa. Al igual que los FBs, los SFBs son bloques "con memoria". Para los SFBs se han de crear también bloques de datos de instancia y cargar en la CPU como parte integrante del programa.

Las CPUs S7 ofrecen SFBs

- para la comunicación vía enlaces configurados
- para las funciones especiales integradas (p.ej. SFB 29 "HS\_COUNT" en la CPU 312 IFM y en la CPU 314 IFM)

##### Funciones del sistema

Una función del sistema es una función preprogramada integrada en la CPU S7. La SFC se puede llamar desde el programa. Como las SFCs forman parte del sistema operativo, no se cargan como parte integrante del programa. Al igual que las FCs, las SFCs son bloques "sin memoria".

Las CPUs S7 ofrecen SFCs para:

- funciones de copia y de bloque
- control del programa
- manipulación del reloj y del contador de horas de funcionamiento
- transferencia de registros
- transferencia de eventos en el modo Multiprocesamiento desde una CPU a todas las CPUs insertadas
- manipulación de alarmas horarias y de retardo
- manipulación de eventos de errores síncronos, eventos de errores de alarma y asíncronos
- información sobre datos de sistema estáticos y dinámicos, p. ej. diagnóstico
- actualización de imágenes del proceso y tratamiento de campos de bits
- direccionamiento de módulos
- periferia descentralizada
- comunicación por datos globales
- la comunicación vía enlaces no configurados
- generar mensajes de bloque

## Informaciones adicionales

Para obtener informaciones más detalladas sobre los SFBs y las SFCs, consulte el manual de referencia "Software de sistema para S7-300/400, funciones estándar y funciones de sistema". En los manuales "Sistema de automatización S7-300, Configuración, instalación y datos de las CPU" o "Sistemas de automatización S7-400, M7-400, Datos de los módulos" se indican los SFBs y las SFCs que se encuentran disponibles.

## 4.2.5 Bloques de organización para la ejecución controlada por alarmas

### 4.2.5.1 Bloques de organización para la ejecución controlada por alarmas

Gracias a los OBs de alarma las CPUs ofrecen las siguientes prestaciones:

- Posibilidad de ejecutar partes del programa por control de tiempo
- Posibilidad de reaccionar eficazmente a señales externas del proceso.

El programa de usuario cíclico no necesita consultar una y otra vez si han aparecido eventos de alarma, sino que cuando aparece una alarma el sistema operativo se encarga de que se ejecute aquella parte del programa que reside en el OB de alarma y que define cómo ha de reaccionar sistema de automatización a esta alarma.

### Tipos de alarmas y su aplicación

La tabla siguiente muestra la aplicación que se les da a los distintos tipos de alarmas.

Tipo de alarma	OBs de alarma	Ejemplos de aplicación
Alarma horaria	OB 10 bis OB 17	Calcular el flujo de un proceso de mezcla al final de un turno
Alarma de retardo	OB 20 bis OB 23	Controlar un ventilador para que se pare 20 segundos después de pararse un motor.
Alarma cíclica	OB 30 bis OB 38	Muestrear el nivel de una señal para una planta de regulación
Alarma de proceso	OB 40 bis OB 47	Notificar que se ha alcanzado el nivel máximo de un depósito.

#### 4.2.5.2 Bloques de organización de alarma horaria (OB 10 a OB 17)

Las CPUs S7 ofrecen OBs de alarmas horarias que pueden ejecutarse a una fecha determinada o en intervalos específicos.

Las alarmas horarias se pueden activar:

- una vez, en una fecha determinada (indicación de hora absoluta con fecha)
- periódicamente, indicando la fecha de arranque y la frecuencia de repetición (p. ej., cada minuto, cada hora, cada día).

#### Reglas para las alarmas horarias

Las alarmas horarias sólo se pueden tratar si se ha parametrizado la alarma horaria y el correspondiente bloque de organización está contenido en el programa de usuario. En caso contrario se escribe un mensaje de error en el búfer de diagnóstico y se efectúa un tratamiento asíncrono del error (OB 80, véase "Bloques de organización de tratamiento de errores (OB 70 a OB 87 / OB 121 a OB 122)").

Las alarmas horarias periódicas deben corresponder a una fecha real. No será posible la repetición mensual de un OB 10 cuyo tiempo de arranque sea 31.1. En este caso, el OB sólo sería arrancado en los meses que tengan 31 días.

Una alarma horaria activada durante el arranque (rearranque completo o rearranque) se ejecuta sólo tras finalizar el arranque.

Los OBs de alarmas horarias desactivados por parametrización no se pueden arrancar. La CPU reconoce un error de programación y pasa a STOP.

Tras un rearranque completo será preciso reactivar todas las alarmas horarias ajustadas (p.ej. mediante la SFC 30 ACT\_TINT en el programa de arranque).

#### Arrancar la alarma horaria

Para que la CPU pueda arrancar una alarma horaria, ésta ha de ser ajustada previamente y activada luego. Se dispone de tres posibilidades de arranque:

- arranque automático de la alarma horaria por parametrización con STEP 7 (bloque de parámetros "alarmas horarias")
- ajustar y activar desde el programa de usuario la alarma horaria a través de la SFC 28 SET\_TINT y SFC 30 ACT\_TINT
- ajustar la alarma horaria por parametrización con STEP 7 y activarla desde el programa de usuario a través de la SFC 30 ACT\_TINT.

#### Consultar la alarma horaria

Para consultar si hay alarmas horarias ajustadas, y a qué hora, se puede

- llamar la SFC 31 QRY\_TINT o
- solicitar la lista parcial "Estado de alarma" de la lista de estado del sistema.

### Desactivar la alarma horaria

Las alarmas horarias aún no ejecutadas se pueden desactivar con la SFC 29 CAN\_TINT. Las alarmas horarias desactivadas se pueden ajustar nuevamente con la SFC 28 SET\_TINT y activar con la SFC 30 ACT\_TINT.

### Prioridad de los OBs de alarma horaria

Los ocho OBs de alarmas horarias están preajustados con la misma prioridad (2) y, por consiguiente, se ejecutan en la secuencia de sus eventos de arranque. La prioridad se puede modificar por parametrización.

### Cambiar la hora ajustada

Para cambiar la hora ajustada se dispone de las siguientes posibilidades:

- un reloj maestro sincroniza la hora para maestros y esclavos
- la hora se ajusta nuevamente en el programa de usuario con la SFC 0 SET\_CLK.

### Comportamiento en caso de cambiar la hora

La tabla siguiente ilustra cómo se comportan las alarmas horarias tras cambiar la hora.

Si...	entonces...
por adelantar la hora se saltan una o varias alarmas horarias,	se arranca el OB 80 y se registra en la información de arranque del OB 80 qué alarmas horarias han sido saltadas.
no se han desactivado en el OB 80 las alarmas horarias saltadas,	no se reactivan dichas alarmas horarias.
no se han desactivado en el OB 80 las alarmas horarias saltadas,	se reactiva la primera alarma horaria saltada y se ignoran las demás alarmas horarias saltadas.
por retrasar la hora, quedan pendientes nuevamente alarmas horarias ya ejecutadas,	no se repite la ejecución de dichas alarmas horarias.

#### 4.2.5.3 Bloques de organización de alarma de retardo (OB 20 a OB 23)

Las CPUs S7 ofrecen OBs de alarmas de retardo que permite programar la ejecución retardada de partes del programa de usuario.

#### Reglas para las alarmas de retardo

Las alarmas de retardo sólo se pueden ejecutar si el correspondiente bloque de organización está contenido en el programa de la CPU. En caso contrario se escribe un mensaje de error en el búfer de diagnóstico y se efectúa un tratamiento asíncrono del error (OB 80, véase "Bloques de organización de tratamiento de errores (OB 70 a OB 87 / OB 121 a OB 122)").

Los OBs de alarmas de retardo desactivados por parametrización no se pueden arrancar. La CPU reconoce un error de programación y pasa a STOP.

Las alarmas de retardo se activan al transcurrir el tiempo de retardo ajustado en la SFC 32 SRT\_DINT.

### Arrancar la alarma de retardo

Para arrancar una alarma de retardo se debe ajustar en la SFC 32 el tiempo de retardo, tras cuya expiración se ha de llamar al correspondiente OB de alarma de retardo. En los manuales "Sistema de automatización S7-300, Configuración, instalación y datos de las CPU" y "Sistemas de automatización S7-400, M7-400, Datos de los módulos" se indica la duración máxima admisible del tiempo de retardo.

### Prioridad de los OBs de alarma de retardo

Para los OBs de alarma de retardo se han preajustado las prioridades 3 a 6. Estas últimas se pueden modificar por parametrización.

#### 4.2.5.4 Bloques de organización de alarma cíclica (OB 30 a OB 38)

Las CPUs S7 ofrecen OBs de alarmas cíclicas que interrumpen la ejecución cíclica del programa en intervalos determinados.

Las alarmas cíclicas se activan en intervalos de tiempo determinados. El tiempo de arranque del período corresponde al cambio de estado operativo de STOP a RUN.

### Reglas para las alarmas cíclicas

Al ajustar los períodos se debe tener en cuenta que entre los eventos de arranque de las diferentes alarmas cíclicas haya tiempo suficiente para la ejecución de dichas alarmas.

Los OBs de alarma cíclica desactivados por parametrización no se pueden arrancar. La CPU reconoce un error de programación y pasa a STOP.

### Arrancar la alarma cíclica

Para arrancar una alarma cíclica es necesario indicar el período (base de tiempo) correspondiente con STEP 7 en el bloque de parámetros de alarmas cíclicas. El período es siempre un múltiplo entero del período básico de 1 ms.

Período =  $n \times$  período básico 1 ms

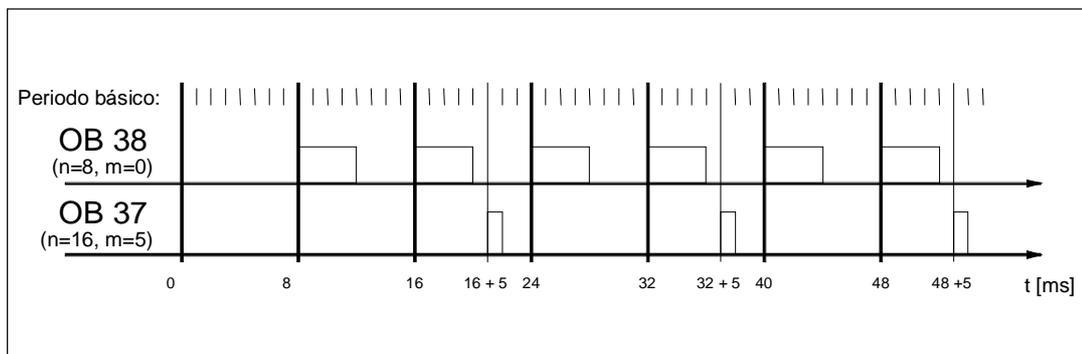
Los nueve OBs de alarmas cíclicas disponibles contienen períodos preajustados (v. tabla siguiente). El período predeterminado entra en vigor cuando está cargado el OB de alarmas cíclicas asignado. Sin embargo, los valores preajustados se pueden modificar por parametrización. En los manuales "Sistema de automatización S7-300, Configuración, instalación y datos de las CPU" y "Sistemas de automatización S7-400, M7-400, Datos de los módulos" se indica el límite superior.

### Alarmas cíclicas con desfase

Para impedir que las alarmas de diferentes OBs de alarmas cíclicas reciban al mismo tiempo una petición de arranque y se produzca eventualmente un error de tiempo (rebase del tiempo de ciclo), existe la posibilidad de prescribir un desfase. El desfase permite que, al transcurrir el período, la ejecución de una alarma cíclica se desplace en un intervalo de tiempo determinado.

Desfase =  $m \times$  período básico (con  $0 \leq m < n$ )

La figura siguiente muestra la ejecución de un OB de alarma cíclica con desfase (OB 37) en contraposición a una alarma cíclica sin desfase (OB 38).



### Prioridad del OB de alarma cíclica

La tabla siguiente muestra los períodos preajustados y las prioridades de los OBs de alarma cíclica. Tanto el período como la prioridad se pueden modificar por parametrización.

OBs de alarma cíclica	Período en ms	Prioridad
OB 30	5000	7
OB 31	2000	8
OB 32	1000	9
OB 33	500	10
OB 34	200	11
OB 35	100	12
OB 36	50	13
OB 37	20	14
OB 38	10	15

### 4.2.5.5 Bloques de organización de alarma de proceso (OB 40 a OB 47)

Las CPUs S7 ofrecen OBs de alarma de proceso que reaccionan a las señales de los módulos (p.ej. módulos de señales SMs, procesadores de comunicaciones CPs, módulos de función FMs). Para los módulos digitales y analógicos parametrizables se puede ajustar con STEP 7 qué señal debe arrancar al OB. Para los CPs y FMs se deben utilizar al respecto las pantallas de parametrización correspondientes.

Las alarmas de proceso son activadas cuando un módulo de señales con habilitación de alarma de proceso parametrizada transfiere a la CPU una señal de proceso recibida o cuando un módulo de función señala a la CPU una alarma.

## **Reglas para las alarmas de proceso**

Las alarmas de proceso sólo se pueden ejecutar si el correspondiente bloque de organización está contenido en el programa de la CPU. En caso contrario se escribe un mensaje de error en el búfer de diagnóstico y se efectúa un tratamiento asíncrono del error (véase "Bloques de organización de tratamiento de errores (OB 70 a OB 87 / OB 121 a OB 122)").

Los OBs de alarmas de proceso desactivados por parametrización no se pueden arrancar. La CPU reconoce un error de programación y pasa a STOP.

## **Parametrizar módulos de señales aptos para alarmas de proceso**

Cada canal de un módulo de señales apto para alarmas de proceso puede disparar una alarma de proceso. Por consiguiente, es necesario definir con STEP 7 en los registros de parámetros de tales módulos de señales:

- con qué se ha de disparar una alarma de proceso
- qué OB de alarma de proceso se debe ejecutar (el preajuste prevé el OB 40 para la ejecución de todas las alarmas de proceso).

Con STEP 7 se activa la generación de alarmas de proceso de los módulos de función. Otros parámetros se pueden asignar en las pantallas de parametrización de dichos módulos de función.

## **Prioridad de los OBs de alarma de proceso**

Para los OBs de alarma de proceso se han preajustado las prioridades 16 a 23. Estas últimas se pueden modificar por parametrización.

#### 4.2.5.6 Bloques de organización de arranque (OB 100 / OB 101 / OB 102)

##### Tipos de arranque

Se diferencia entre los siguientes tipos de arranque:

- Rearranque (no en S7-300 y S7-400H)
- Rearranque completo
- Arranque en frío

En la tabla siguiente se indica qué OB llama al sistema operativo durante el arranque.

Tipo de arranque	OB correspondiente
Rearranque	OB 101
Rearranque completo	OB 100
Arranque en frío	OB 102

##### Eventos de arranque de los OBs de arranque

La CPU realiza un arranque por

- tras CONEXIÓN
- al cambiar el selector de modos de operación de STOP a "RUN"/"RUN-P"
- tras petición por una función de comunicación
- tras sincronización en modo multiprocesador
- en un sistema H tras acoplar (sólo en CPU de reserva)

El OB de arranque (OB 100, OB 101 u OB 102) se invoca dependiendo del evento de arranque, de la CPU existente y de los parámetros ajustados en la misma.

##### Programa de arranque

Las condiciones para el comportamiento durante el arranque (valores de inicialización para RUN, valores de arranque para módulos de periferia) de la CPU se pueden definir depositando el programa para el arranque en el bloque de organización OB 100 para el rearranque completo, el OB 101 para el rearranque u OB 102 para el arranque en frío.

El programa de arranque puede tener una longitud discrecional, su ejecución no tiene límite de tiempo y la vigilancia del tiempo de ciclo no está activa. La ejecución controlada por tiempo o por alarma no es posible en el programa de arranque. En el arranque, todas las salidas digitales tienen el estado de señal 0.

### **Modo de arranque tras arranque manual**

En las CPUs S7-300 sólo se puede efectuar un re arranque completo manual o un arranque en frío (sólo CPU 318-2).

Las CPUs S7-400 permiten efectuar un re arranque manual a través del selector de modos de operación y del selector de modos de arranque (CRST/WRST), siempre y cuando se haya parametrizado correspondientemente con STEP 7. El re arranque completo manual también es posible sin parametrización previa.

### **Modo de arranque tras arranque automático**

En las CPUs S7-300 sólo es posible efectuar un re arranque completo tras CONEXION.

En las CPUs S7-400 se puede definir si, tras CONEXION, el arranque automático debe provocar un re arranque completo o un re arranque normal.

### **Borrar la imagen de proceso**

Durante el re arranque de una CPU S7-400 se borra, por defecto, la imagen de proceso de las salidas tras la ejecución del ciclo residual. El borrado de la imagen de proceso se puede inhibir si se desea que el programa de usuario continúe utilizando los valores que eran actuales antes de dicho re arranque.

### **Vigilancia de la configuración teórica/real de los módulos**

A través de la parametrización se puede definir si, antes de efectuar el arranque, se debe comprobar si todos los módulos listados en la tabla de configuración están insertados realmente y si el tipo de módulo es correcto.

Si la vigilancia de módulos está activa, el arranque no se ejecuta en caso que se detecte una diferencia entre las configuraciones teórica y real.

### **Tiempos de vigilancia**

Para garantizar un arranque sin errores del sistema de automatización, se han de parametrizar los siguientes tiempos de vigilancia:

- el tiempo máximo admisible para la transmisión de parámetros a los módulos
- el tiempo máximo admisible para el mensaje "ready" de los módulos tras CONEXION
- en las CPUs S7-400, el tiempo máximo de interrupción en el cual puede ejecutarse aún un re arranque.

Transcurridos los tiempos de vigilancia, la CPU pasa a STOP o sólo es posible efectuar un re arranque completo.

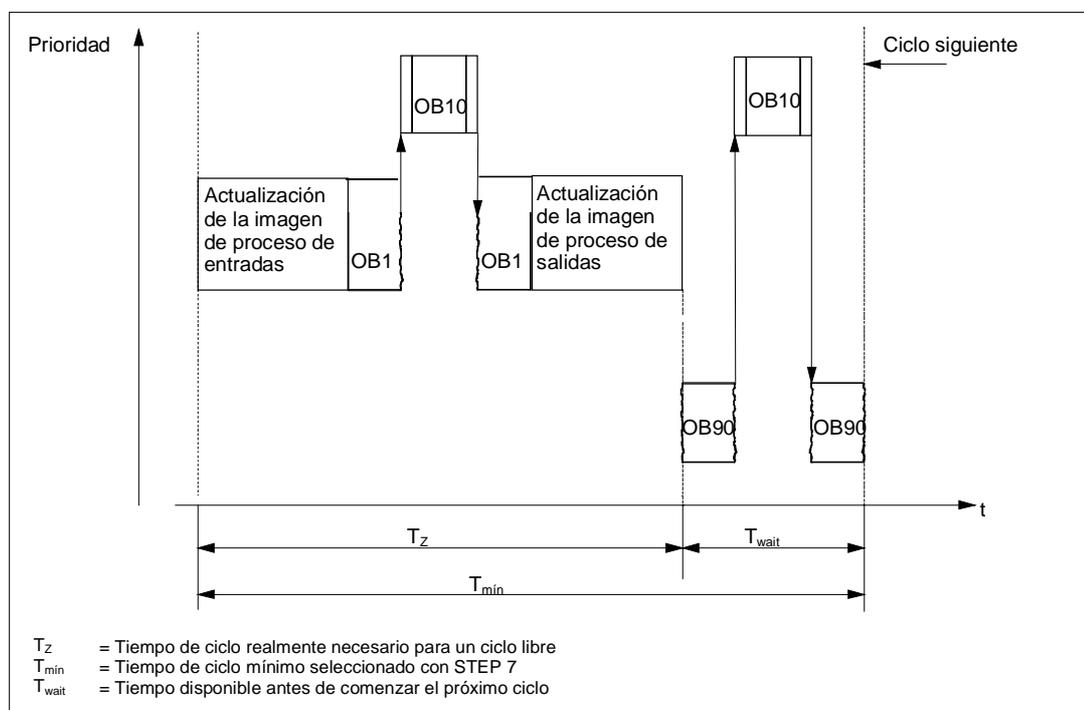
#### 4.2.5.7 Bloque de organización de tarea no prioritaria (OB 90)

Si con STEP 7 se ha definido un tiempo de ciclo mínimo y éste es mayor que el tiempo de ciclo real, al finalizar el programa cíclico la CPU dispone aún de tiempo de ejecución. Dicho tiempo se utiliza para procesar el OB de tarea no prioritaria. Si la CPU utilizada carece de OB 90, entonces espera hasta que transcurre el tiempo de ciclo mínimo seleccionado. Por tanto, a través del OB 90 se pueden ejecutar procesos de tiempo no crítico, evitando así tiempos de espera.

#### Prioridad del OB de tarea no prioritaria

El OB de tarea no prioritaria tiene la prioridad 29, que equivale a la prioridad 0.29. Por tanto, se trata del OB con la prioridad más baja. La prioridad no puede modificarse por parametrización.

La figura siguiente muestra un ejemplo de ejecución del ciclo de tarea no prioritaria, del ciclo libre y del OB 10 (en las CPUs antiguas).



#### Programar el OB 90

El sistema operativo de la CPU no supervisa el tiempo de ejecución del OB 90, por lo que en éste es posible programar bucles de cualquier longitud. Asegúrese de que los datos utilizados en el programa de baja prioridad sean coherentes, observando lo siguiente al programar:

- los eventos de inicialización del OB 90 (consulte a este respecto el manual de referencia "Software de sistema para S7-300/400, funciones estándar y funciones de sistema"),
- la actualización de la imagen del proceso asíncrona al OB 90.

#### 4.2.5.8 Bloques de organización de tratamiento de errores (OB 70 a OB 87 / OB 121 a OB 122)

##### Tipos de errores

Los errores reconocibles por las CPUs S7 y a los cuales se puede reaccionar con los bloques de organización se clasifican en dos categorías:

- Errores síncronos: estos errores se pueden asignar a una parte determinada del programa de usuario. El error es provocado por una determinada operación durante la ejecución. Si el OB de error síncrono correspondiente no está cargado, la CPU pasa a STOP al presentarse el error.
- Errores asíncronos: estos errores no se pueden asignar directamente al programa de usuario en ejecución. Se trata de errores de prioridad, errores en el sistema de automatización (p. ej., fallos de módulos) o de errores de redundancia. Si el OB de error asíncrono correspondiente no está cargado, la CPU pasa a STOP al presentarse el error. (Excepciones: OB 70, OB 72, OB 81).

La tabla siguiente muestra los tipos de errores que se pueden presentar, subdivididos según la categoría de los OBs de error.

Errores asíncronos / errores de redundancia	Errores síncronos
OB 70 Errores de redundancia en la periferia (sólo en CPUs H)	OB 121 Error de programación (p.ej. DB sin cargar)
OB 72 Errores de redundancia en la CPU (sólo en CPUs H, p.ej. fallo de una CPU)	OB 122 Error de acceso a periferia (p.ej. acceso a un módulo de señales que no existe)
OB 73 Error de redundancia de comunicación (sólo en CPUs H, p.ej. pérdida de redundancia de un enlace S7 de alta disponibilidad)	
OB 80 Error de tiempo (p.ej. tiempo de ciclo sobrepasado)	
OB 81 Error de alimentación (p.ej. fallo de pila)	
OB 82 Alarma de diagnóstico (p.ej. cortocircuito en un módulo de entradas)	
OB 83 Alarma de presencia de módulo (p.ej. extraer un módulo de entradas)	
OB 84 Avería de hardware CPU (error en interface a red MPI)	
OB 85 Error de ejecución del programa (p.ej. OB sin cargar)	
OB 86 Error en el bastidor	
OB 87 Error de comunicación (p.ej. identificación de telegrama errónea en comunicación por datos globales)	

### Usar los OBs para errores síncronos

Los errores síncronos son causados durante la ejecución de una operación determinada. Cuando ocurren estos errores, el sistema operativo registra una entrada en la pila U (USTACK) y arranca el OB para errores síncronos.

Los OBs de errores que son llamados por errores síncronos se ejecutan como parte del programa con la misma prioridad que el bloque en tratamiento al detectarse el error. Así, el OB 121 y el OB 122 pueden acceder a los valores memorizados en los acumuladores y otros registros en el momento de la interrupción. Los valores se pueden utilizar para reaccionar a la condición de error y regresar luego a la ejecución del programa (p.ej. en caso de errores de acceso a módulo de entradas analógicas, predeterminar un valor de sustitución en el OB 122 con la SFC 44 RPL\_VAL). No obstante, los datos locales de los OBs de error representan una carga adicional para la LSTACK (pila de datos locales) de esta prioridad. En las CPUs S7-400 es posible arrancar desde un OB de error síncrono a otro OB de error síncrono. Esto no es posible en las CPUs S7-300.

### Usar los OBs para errores asíncronos

Si el sistema operativo de la CPU detecta un error asíncrono, arranca el correspondiente OB de error (OB 70 a OB 73 y OB 80 a OB 87). Los OBs para errores asíncronos tienen la máxima prioridad: No pueden ser interrumpidos por otros OBs, si todos los OBs para errores asíncronos tienen la misma prioridad. Al presentarse simultáneamente varios OBs para errores asíncronos de igual prioridad, serán ejecutados en el orden de aparición.

### Enmascarar eventos de arranque

Las funciones del sistema (SFC) permiten enmascarar, retardar o inhibir eventos de arranque para algunos OBs de tratamiento de errores. Para obtener informaciones más detalladas a este respecto, así como en relación con los diversos bloques de organización, consulte el manual de referencia "Software de sistema para S7-300/400, funciones estándar y funciones de sistema".

Tipo de OB de error	SFC	Función de la SFC
OB de errores síncronos	SFC 36 MSK_FLT	Enmascarar eventos de error síncronos individuales. Los eventos de error enmascarados no arrancan ningún OB de error y no conducen a la reacción de sustitución programada.
	SFC 37 DMSK_FLT	Desenmascarar eventos de error síncronos.
OB de errores asíncronos	SFC 39 DIS_IRT	Inhibir globalmente los eventos de alarma y los de errores asíncronos. Los eventos de error inhibidos no arrancan OBs de error en ninguno de los ciclos posteriores de la CPU y no conducen a la reacción de sustitución programada.
	SFC 40 EN_IRT	Habilitar eventos de alarma y de errores asíncronos
	SFC 41 DIS_AIRT	Retardar los eventos de alarma y los de errores asíncronos de mayor prioridad hasta el final del OB.
	SFC 42 EN_AIRT	Habilitar los eventos de alarma y los de errores asíncronos de mayor prioridad

---

#### Nota

Para ignorar alarmas resulta más efectivo inhibirlas mediante una SFC durante el arranque, en vez de cargar un OB vacío (con resultado binario RB).

---

## 5 Cómo arrancar y utilizar STEP 7

### 5.1 Cómo arrancar y utilizar STEP 7

#### 5.1.1 Cómo iniciar STEP 7



Una vez arrancado Windows, en la pantalla aparece el icono del Administrador SIMATIC que permite acceder al software STEP 7.

La manera más rápida de arrancar STEP 7 consiste en hacer un doble clic en el icono "Administrador SIMATIC", con lo que se abrirá la ventana del Administrador SIMATIC. Desde allí es posible acceder a todas las funciones instaladas, tanto del paquete estándar como de los paquetes opcionales.

Asimismo, puede iniciar el Administrador SIMATIC a través del botón "Inicio" de la barra de tareas de Windows 95/98/NT (entrada en "Simatic").

---

#### Nota

Para obtener más información sobre las funciones y opciones estándar de Windows, consulte el manual del usuario o la Ayuda en pantalla de su sistema operativo Windows.

---

### Administrador SIMATIC

El Administrador SIMATIC es el interface de acceso a la configuración y programación. Éste permite:

- crear proyectos,
- configurar y parametrizar el hardware,
- configurar redes de hardware,
- programar bloques,
- probar y hacer funcionar los programas.

El acceso a las funciones es orientado a objetos, con lo cual resulta fácil de aprender.

Puede trabajar con el Administrador SIMATIC

- offline, es decir, sin conectar el sistema de automatización, o bien
- online, es decir, estando conectado el sistema de automatización.

(Tenga en cuenta las correspondientes observaciones de seguridad.)

## Procedimiento ulterior

Las soluciones de automatización se crean en forma de "proyectos". Si se familiariza previamente con los siguientes temas generales, el trabajo le resultará más sencillo:

- el interface de usuario,
- algunos manejos básicos,
- la Ayuda en pantalla.

### 5.1.2 Arrancar STEP 7 con los parámetros de arranque estándar

A partir de la versión 5 de STEP 7 se pueden crear varios iconos del Administrador SIMATIC e indicar parámetros de arranque en la línea de llamada con objeto de que el Administrador SIMATIC se posicione en el objeto descrito por esos parámetros. Con un doble clic puede acceder directamente a los puntos correspondientes en un determinado proyecto.

Llamando a **s7tgotopx.exe** puede indicar los siguientes parámetros de arranque:

**/e** <ruta física completa del proyecto>

**/o** < ruta lógica del objeto donde se debe posicionar >

**/h** < ID del objeto > /on u /off

A continuación se indica el método más sencillo para averiguar los parámetros apropiados.

#### Averiguar los parámetros utilizando las funciones "Copiar" y "Pegar"

Proceda como se describe a continuación:

1. Cree en su escritorio un nuevo acceso directo al archivo s7tgotopx.exe.
2. Abra el cuadro de diálogo de propiedades.
3. Elija la ficha "Acceso directo". La entrada que aparece en "Destino" se complementa de la siguiente forma:
4. Seleccione el objeto deseado en el Administrador SIMATIC.
5. Copie el objeto al portapapeles, utilizando la combinación de teclas CTRL+C.
6. Sitúe el cursor al final de la entrada "Destino" en la ficha "Acceso directo".
7. Pegue el contenido del portapapeles utilizando la combinación de teclas CTRL+V.
8. Cierre el cuadro de diálogo haciendo clic en "Aceptar".

Ejemplos de parámetros:

**/e** F:\SIEMENS\STEP7\S7proj\MyConfig\MyConfig.s7p

**/o** "1,8:MyConfig\SIMATIC 400(1)\CPU416-1\ProgramaS7(1)\Bloques\FB1"

**/h** T00112001;129;T00116001;1;T00116101;16e

Nota respecto a la estructura de la ruta del proyecto

La ruta del proyecto es la ruta física del sistema de archivos. Aquí no se asiste la notación UNC, como p.ej. F:\SIEMENS\STEP7\S7proj\MyConfig\MyConfig.s7p.

La ruta lógica completa está estructurada de la siguiente forma:

[Identificador de visualización,identificador online]:nombre del proyecto{\nombre del objeto}\* \ nombre del objeto

Ejemplo: /o 1,8:MyConfig\SIMATIC 400(1)\CPU416-1\ProgramaS7(1)\Bloques\FB1

Nota respecto a la estructura de la ruta lógica

La ruta lógica completa y el ID del objeto sólo se pueden crear utilizando las funciones Copiar y Pegar. No obstante, también existe la posibilidad de indicar la ruta que el usuario puede leer, es decir, conforme al ejemplo indicado arriba:

/o "MyConfig\SIMATIC 400(1)\CPU416-1\ProgramaS7(1)\Bloques\FB1".

Con /onl u /off, el usuario puede indicar si la ruta es válida en la ventana online u offline. Esta indicación no es necesaria al utilizarse las funciones Copiar y Pegar.

Nota importante: Si la ruta contiene espacios en blanco, se deberá colocar entre comillas.

### 5.1.3 Cómo acceder a las funciones de ayuda

#### Ayuda en pantalla

La Ayuda en pantalla ofrece informaciones directas. Así puede consultar de forma puntualizada la información que necesite, sin tener que buscarla en un manual. En la Ayuda en pantalla encontrará:

- **Temas de Ayuda:** ofrece diversos accesos para visualizar los temas de Ayuda.
- **Ayuda contextual** (tecla F1): muestra informaciones sobre el objeto seleccionado, o bien, acerca del cuadro de diálogo o de la ventana actual.
- **Introducción:** presenta una panorámica de la aplicación, de sus principales características y del volumen de sus funciones.
- **Primeros pasos:** resume los primeros pasos que se deben dar para poder comenzar a trabajar con éxito.
- **Uso de la Ayuda:** describe las posibilidades disponibles para encontrar determinadas informaciones en la Ayuda en pantalla.
- **Acerca de:** proporciona información sobre la versión actual de la aplicación.

A través del menú "Ayuda" es posible acceder también desde cualquier ventana a los temas que guarden relación con el cuadro de diálogo actual.

### Llamar la Ayuda en pantalla

Para llamar la Ayuda en pantalla puede utilizar alguna de las siguientes alternativas:

- Elegir un comando del menú "Ayuda" en la barra de menús.
- Hacer clic en el botón "Ayuda" de un cuadro de diálogo. A continuación, obtendrá ayuda sobre dicho cuadro.
- Situar el puntero del ratón en una ventana o en un cuadro de diálogo, en el tema sobre el que requiera ayuda y pulsando la tecla F1 o elegir el comando **Ayuda > Ayuda contextual**.
- Hacer clic con el cursor de interrogación de Windows (¿Qué es esto?).

Las tres últimas alternativas constituyen la Ayuda contextual.

### Llamar la Ayuda rápida

Si sitúa el puntero del ratón en un botón de la barra de herramientas y lo deja allí por corto tiempo, obtendrá ayuda rápida sobre ese botón.

### Cambiar el tamaño de letra

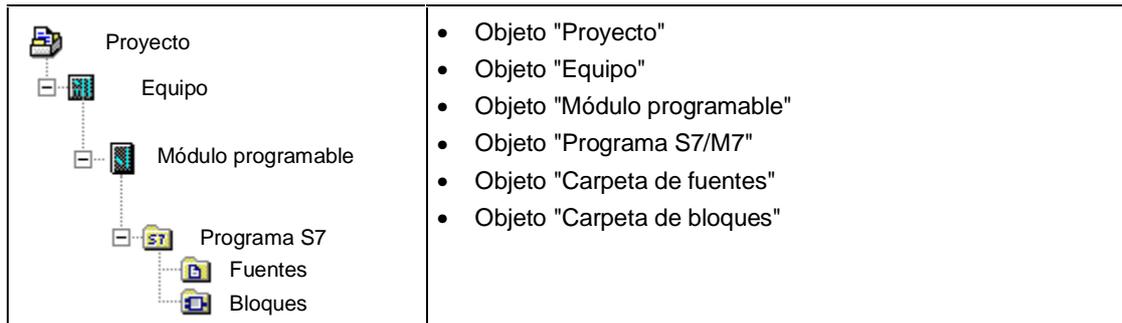
Con el comando **Opciones > Fuente** de la ventana de la Ayuda contextual es posible cambiar el tamaño de letra, eligiendo "Pequeña", "Normal" o "Grande".

## 5.2 Objetos y su jerarquía

### 5.2.1 Objetos y su jerarquía

La jerarquía de objetos para proyectos y librerías del Administrador SIMATIC está estructurada de la misma forma que los directorios con carpetas y archivos del Explorador de Windows.

La figura siguiente muestra un ejemplo de jerarquía de objetos.



Los objetos sirven de:

- portadores de propiedades,
- carpetas,
- portadores de funciones (p.ej., para arrancar una aplicación determinada).

#### Objetos portadores de propiedades

Los objetos pueden ser tanto portadores de funciones como de propiedades (p. ej., de ajustes). Una vez seleccionado un objeto es posible

- editar el objeto eligiendo el comando **Edición > Abrir objeto**.
- abrir un cuadro de diálogo con el comando **Edición > Propiedades del objeto** y determinar allí los ajustes específicos del objeto.

Una carpeta puede ser también portador de propiedades.

#### Objetos del tipo "Carpeta"

Una carpeta puede contener otras carpetas u objetos. Estos se visualizan al abrir la carpeta.

#### Objetos portadores de funciones

Al abrir un objeto aparece una ventana para poder editarlo.

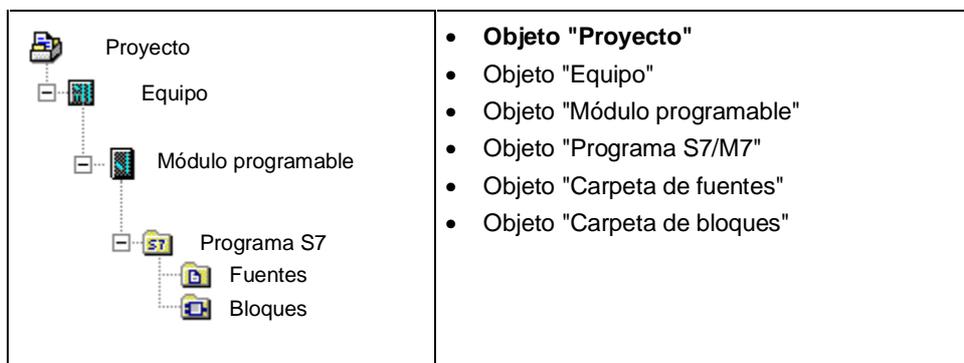
Un objeto puede ser una carpeta o un portador de funciones. Los equipos constituyen la excepción a la regla, puesto que son tanto carpetas (de módulos programables) como portadores de funciones (para configurar el hardware).

- Si hace doble clic en un equipo, se mostrarán los objetos que contiene: los módulos programables y la configuración del equipo (equipo como carpeta).
- Si abre un equipo con el comando **Edición > Abrir objeto**, puede configurar y parametrizar dicho equipo (equipo portador de una función). Este comando de menú tiene el mismo efecto que un doble clic en el objeto "Hardware".

## 5.2.2 Objeto "Proyecto"

Un proyecto representa la totalidad de los datos y programas de una tarea de automatización, constituyendo el nivel superior de la jerarquía de objetos.

### Posicionamiento en la ventana del proyecto

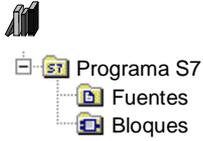


Símbolo	Carpeta de objetos	Selección de funciones importantes
	Proyecto	<ul style="list-style-type: none"> <li>• Crear un proyecto</li> <li>• Archivar proyectos y librerías</li> <li>• Imprimir la documentación de un proyecto</li> <li>• Reorganizar</li> <li>• Traducir y editar textos de usuario</li> <li>• Insertar estaciones de operador (OS)</li> <li>• Compartir la elaboración de proyectos</li> <li>• Convertir proyectos de la versión 1</li> <li>• Convertir proyectos de la versión 2</li> <li>• Ajustar interface PG/PC</li> </ul>

Símbolo	Objetos en el nivel de proyecto	Selección de funciones importantes
	Equipo: Equipo SIMATIC 300 Equipo SIMATIC 400	<ul style="list-style-type: none"> <li>• Insertar equipos</li> <li>• Los equipos son al mismo tiempo objetos (del nivel de proyectos) y carpetas de objetos (del nivel de equipo). En el tema Objeto "Equipo" se describen las demás funciones.</li> </ul>
	Programa S7  Programa M7	<ul style="list-style-type: none"> <li>• Programa S7/M7 sin equipo ni CPU</li> <li>• Los programas S7/M7 son al mismo tiempo objetos (del nivel de proyectos) y carpetas de objetos (del nivel de programas). En el tema Objeto "Programa S7/M7" se describen las demás funciones.</li> </ul>
	Red para arrancar la aplicación para configurar la red y ajustar sus propiedades	<ul style="list-style-type: none"> <li>• Propiedades de subredes y estaciones de la red</li> <li>• Comunicación de datos globales. Sinopsis</li> <li>• Procedimiento para configurar la comunicación de datos globales</li> </ul>

### 5.2.3 Carpeta de objetos "Librería"

Una librería puede contener programas S7/M7 y sirve para depositar bloques. Se encuentra en el nivel superior de una jerarquía de objetos.

	<ul style="list-style-type: none"> <li>• Objeto "Librería"</li> <li>• Objeto "Programa S7/M7"</li> <li>• Objeto "Carpeta de fuentes"</li> <li>• Objeto "Carpeta de bloques"</li> </ul>
---	--

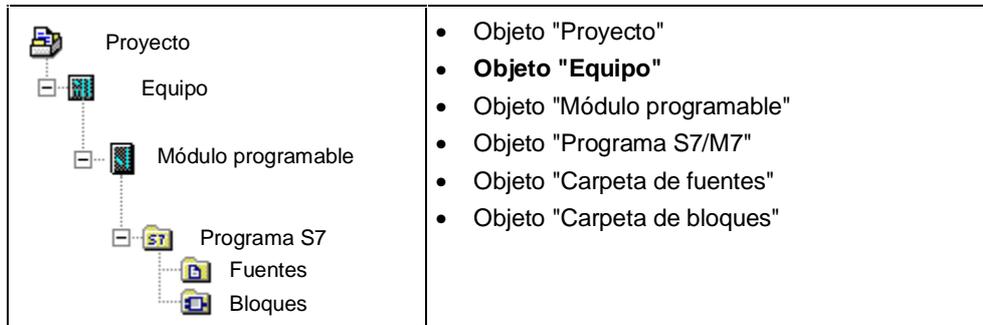
Símbolo	Carpeta de objetos	Selección de funciones importantes
	Librería	<ul style="list-style-type: none"> <li>• Panorámica de las librerías estándar</li> <li>• Trabajar con librerías</li> <li>• Archivar proyectos y librerías</li> </ul>

Símbolo	Objetos en el nivel de librerías	Selección de funciones importantes
	Programa S7	<ul style="list-style-type: none"> <li>• Insertar un programa S7 o M7</li> <li>• Los programas S7/M7 son tanto objetos (en el nivel de proyectos) como carpetas de objetos (en el nivel de programas). En el tema Objeto "Programa S7/M7" se describen las demás funciones.</li> </ul>
	Programa M7	

### 5.2.4 Objeto "Equipo"

Un equipo SIMATIC 300/400 representa una configuración de hardware S7 compuesta por uno o varios módulos programables.

#### Posicionamiento en la ventana del proyecto



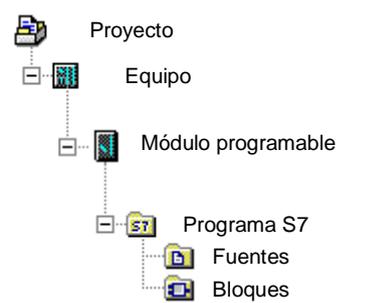
Símbolo	Carpeta de objetos	Selección de funciones importantes
	Equipo	<ul style="list-style-type: none"> <li>• Insertar equipos</li> <li>• Cargar equipo en PG</li> <li>• Cargar una configuración en un sistema de destino</li> <li>• Cargar la configuración desde otro equipo</li> <li>• Visualizar los mensajes de la CPU y los mensajes de diagnóstico personalizados</li> <li>• Configurar la 'notificación de errores del sistema'</li> <li>• Diagnóstico del hardware y visualizar el estado del módulo</li> <li>• Visualizar y cambiar el estado operativo de la CPU</li> <li>• Visualizar y ajustar la hora y la fecha de la CPU</li> <li>• Borrar las memorias de carga/de trabajo y borrado total de la CPU</li> </ul>
	Equipo PC Simatic -	<ul style="list-style-type: none"> <li>• Crear y parametrizar equipos PC SIMATIC</li> <li>• Configurar enlaces para un equipo PC SIMATIC</li> </ul>

Símbolo	Objetos en el nivel de equipos	Selección de funciones importantes
	Hardware	<ul style="list-style-type: none"> <li>• Manejo básico al configurar el hardware</li> <li>• Pasos fundamentales para configurar un equipo</li> <li>• Visión de conjunto: Procedimiento para configurar y parametrizar un sistema centralizado</li> <li>• Procedimiento básico para configurar un sistema maestro DP</li> <li>• Configurar el modo multiprocesador</li> </ul>
	Módulo programable	<ul style="list-style-type: none"> <li>• Los módulos programables son al mismo tiempo objetos (del nivel de equipo) y carpetas de objetos (del nivel módulos programables). En el tema Objeto "Módulos programables" se describen las demás funciones.</li> </ul>

### 5.2.5 Objeto "Módulo programable"

Un módulo programable representa los datos de parametrización de un módulo programable (CPUxxx, FMxxx, CPxxx). Los datos de sistema de los módulos que no disponen de memoria remanente (p.ej. CP441), se cargan a través de la CPU del equipo. Por tanto, a dichos módulos no se ha asociado ningún objeto "Datos de sistema", por lo que no se visualizan en la jerarquía de proyectos.

#### Posicionamiento en la ventana del proyecto

 <p>Proyecto</p> <p>Equipo</p> <p>Módulo programable</p> <p>Programa S7</p> <p>Fuentes</p> <p>Bloques</p>	<ul style="list-style-type: none"> <li>• Objeto "Proyecto"</li> <li>• Objeto "Equipo"</li> <li>• <b>Objeto "Módulo programable"</b></li> <li>• Objeto "Programa S7/M7"</li> <li>• Objeto "Carpeta de fuentes"</li> <li>• Objeto "Carpeta de bloques"</li> </ul>
--	---

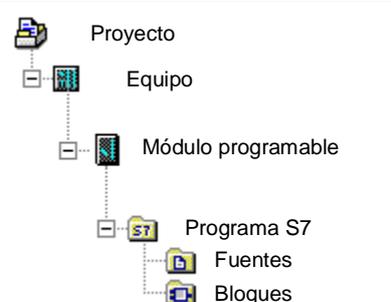
Símbolo	Carpeta de objetos	Selección de funciones importantes
	<p>Módulo programable</p>	<ul style="list-style-type: none"> <li>• Visión de conjunto: Procedimiento para configurar y parametrizar un sistema centralizado</li> <li>• Visualizar los mensajes de la CPU y los mensajes de diagnóstico personalizados</li> <li>• Configurar 'Notificar errores del sistema'</li> <li>• Diagnóstico del hardware y visualizar el estado del módulo</li> <li>• Cargar mediante Memory Cards EPROM</li> <li>• Protección con contraseña para acceder a sistemas de destino</li> <li>• Visualizar ventanas de forzado permanente</li> <li>• Visualizar y cambiar el estado operativo</li> <li>• Visualizar y ajustar la fecha y hora</li> <li>• Ajustar el comportamiento del sistema</li> <li>• Borrar las memorias de carga/de trabajo y borrado total de la CPU</li> <li>• Símbolos de diagnóstico en la ventana online</li> <li>• Subdivisión de la memoria en áreas</li> <li>• Guardar en la EPROM integrada los bloques cargados</li> <li>• Actualizar el sistema operativo en el sistema de destino</li> </ul>

Símbolo	Objetos en el nivel de "Módulos programables"	Selección de funciones importantes
	Programas: Programa S7 Programa M7 Programa	<ul style="list-style-type: none"> <li>• Insertar un programa S7 o M7</li> <li>• Los programas S7/M7 son tanto objetos (en el nivel de proyectos) como carpetas de objetos (en el nivel de programas). En el tema Objeto "Programa S7/M7" se describen las demás funciones.</li> </ul>
	Enlaces para definir enlaces en la red	<ul style="list-style-type: none"> <li>• Conectar equipos de un mismo proyecto a la red</li> <li>• Tipos de enlaces e interlocutores</li> <li>• Informaciones importantes sobre los distintos tipos de enlaces</li> <li>• Introducir un nuevo enlace</li> <li>• Configurar enlaces para módulos en un equipo SIMATIC</li> </ul>

### 5.2.6 Objeto "Programa S7/M7"

Un programa (S7/M7) es una carpeta de software que contiene módulos CPU S7/M7 o módulos que no sean CPUs (p.ej. módulos CP o FM programables).

#### Emplazamiento en la vista del proyecto

	<ul style="list-style-type: none"> <li>• Objeto "Proyecto"</li> <li>• Objeto "Equipo"</li> <li>• Objeto "Módulo programable"</li> <li>• <b>Objeto "Programa S7/M7"</b></li> <li>• Objeto "Carpeta de fuentes"</li> <li>• Objeto "Carpeta de bloques"</li> </ul>
---	---

Símbolo	Carpeta de objetos	Selección de funciones importantes
	Programa S7	<ul style="list-style-type: none"> <li>• Insertar un programa S7 o M7</li> <li>• Ajustar la preferencia de operandos</li> <li>• Procedimiento básico para crear bloques lógicos</li> <li>• Asignar números de mensaje</li> <li>• Generar y editar mensajes de diagnóstico personalizados</li> <li>• Traducir y editar textos de usuario</li> <li>• Visualizar mensajes de CPU y mensajes de diagnóstico personalizados</li> <li>• Medidas en el programa para tratar fallos</li> </ul>
	Programa M7	<ul style="list-style-type: none"> <li>• Procedimiento para sistemas M7</li> </ul>
	Programa	<ul style="list-style-type: none"> <li>• Crear el software en el proyecto</li> </ul>

Símbolo	Objetos en el nivel de programas	Selección de funciones importantes
	Tabla de símbolos para asignar símbolos a señales y otras variables	<ul style="list-style-type: none"> <li>• Direccionamiento absoluto y simbólico</li> <li>• Estructura y componentes de la tabla de símbolos</li> <li>• Distintas posibilidades para introducir símbolos globales</li> <li>• Observaciones generales para introducir símbolos</li> <li>• Asignar y editar mensajes de símbolos</li> <li>• Traducir y editar textos de usuario</li> <li>• Configurar atributos M+V mediante la tabla de símbolos</li> <li>• Editar el atributo de comunicación</li> <li>• Exportar e importar tablas de símbolos</li> </ul>
	Carpeta de fuentes	<ul style="list-style-type: none"> <li>• En el tema Objeto "Carpeta de fuentes" se describen las demás funciones.</li> </ul>
	Carpeta de bloques	<ul style="list-style-type: none"> <li>• En el tema Objeto "Carpeta de bloques" se describen las demás funciones.</li> </ul>

### 5.2.7 Objeto "Carpeta de bloques"

Una carpeta de bloques de una vista offline puede contener bloques lógicos (OB, FB, FC, SFB, SFC), bloques de datos (DB), tipos de datos de usuario (UDT) y tablas de variables. El objeto "Datos de sistema" representa bloques de datos de sistema.

La carpeta de bloques de una vista online contiene las partes ejecutables del programa residentes en el sistema de destino.

#### Posicionamiento en la ventana del proyecto

	<ul style="list-style-type: none"> <li>Objeto "Proyecto"</li> <li>Objeto "Equipo"</li> <li>Objeto "Módulo programable"</li> <li>Objeto "Programa S7/M7"</li> <li>Objeto "Carpeta de fuentes"</li> <li><b>Objeto "Carpeta de bloques"</b></li> </ul>
--	---

Símbolo	Carpeta de objetos	Selección de funciones importantes
	Bloques	<ul style="list-style-type: none"> <li>Cargar con gestión de proyectos</li> <li>Cargar sin gestión de proyectos</li> <li>Visión de conjunto de los datos de referencia posibles</li> <li>Recablear</li> <li>Comparar bloques</li> <li>Traducir y editar textos de usuario</li> <li>Salto a descripciones de lenguajes, ayudas sobre bloques y atributos de sistema</li> </ul>

Símbolo	Objetos en la carpeta de bloques	Selección de funciones importantes
	Bloques en general	<ul style="list-style-type: none"> <li>Procedimiento básico para crear bloques lógicos</li> <li>Crear bloques</li> <li>Reglas para programar en fuentes AWL</li> <li>Comparar bloques</li> </ul>
	OB (Bloques de organización)	Funciones adicionales: <ul style="list-style-type: none"> <li>Introducción a los tipos de datos y a los tipos de parámetros</li> <li>Requisitos para cargar</li> <li>Test con el estado del programa</li> <li>Información importante para el test en modo Etapa individual / Puntos de parada</li> <li>Recablear</li> <li>Ayuda de bloques</li> </ul>

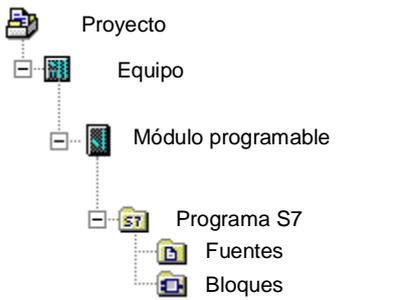
Símbolo	Objetos en la carpeta de bloques	Selección de funciones importantes
	FC (Funciones)	Funciones adicionales: <ul style="list-style-type: none"> <li>• Introducción a los tipos de datos y a los tipos de parámetros</li> <li>• Requisitos para cargar</li> <li>• Test con el estado del programa</li> <li>• Información importante para el test en modo Etapa individual / Puntos de parada</li> <li>• Recablear</li> <li>• Atributos para bloques y parámetros</li> </ul>
	FB (Bloques de función)	Funciones adicionales: <ul style="list-style-type: none"> <li>• Introducción a los tipos de datos y a los tipos de parámetros</li> <li>• Uso de multiinstancias</li> <li>• Requisitos para cargar</li> <li>• Test con el estado del programa</li> <li>• Información importante para el test en modo Etapa individual / Puntos de parada</li> <li>• Recablear</li> <li>• Atributos para bloques y parámetros</li> <li>• Asignar y editar mensajes de bloque</li> <li>• Configuración de mensajes PCS7</li> <li>• Traducir y editar textos de usuario</li> <li>• Asignar atributos de M+V a parámetros FB</li> </ul>
	UDT (Tipos de datos de usuario)	<ul style="list-style-type: none"> <li>• Crear bloques</li> <li>• Reglas para programar en fuentes AWL</li> <li>• Introducción a los tipos de datos y a los tipos de parámetros</li> <li>• Uso de tipos de datos de usuario para acceder a los datos</li> <li>• Atributos para bloques y parámetros</li> </ul>
	DB (Bloques de datos globales)	<ul style="list-style-type: none"> <li>• Ver datos de los bloques de datos</li> <li>• Ver declaración de bloques de datos</li> <li>• Requisitos para cargar</li> <li>• Observar el estado de bloques de datos</li> <li>• Introducción a los tipos de datos y a los tipos de parámetros</li> <li>• Utilización de multiinstancias</li> <li>• Atributos para bloques y parámetros</li> <li>• Asignar y editar mensajes de bloque (sólo DBs de instancia)</li> <li>• Configuración de mensajes PCS7(sólo DBs de instancia)</li> <li>• Traducir y editar textos de usuario(sólo DBs de instancia)</li> </ul>

Símbolo	Objetos en la carpeta de bloques	Selección de funciones importantes
	SFC (Funciones de sistema)	<ul style="list-style-type: none"> <li>Requisitos para cargar</li> <li>Atributos para bloques y parámetros</li> <li>Ayuda de bloques</li> </ul>
	SFB (Bloques de función de sistema)	<ul style="list-style-type: none"> <li>Requisitos para cargar</li> <li>Atributos para bloques y parámetros</li> <li>Asignar y editar mensajes de bloque</li> <li>Configuración de mensajes PCS7</li> <li>Traducir y editar textos de usuario</li> <li>Ayuda sobre bloques</li> </ul>
	VAT (Tabla de variables)	<ul style="list-style-type: none"> <li>Procedimiento básico para observar y forzar con tablas de variables</li> <li>Introducción a la comprobación de programas utilizando una tabla de variables</li> <li>Introducción a la observación de variables</li> <li>Introducción al forzado normal de variables</li> <li>Introducción al forzado permanente de variables</li> </ul>
	Datos del sistema (SDB)	<ul style="list-style-type: none"> <li>Los SDBs sólo se procesan indirectamente a través de funciones de:</li> <li>Introducción a la configuración del hardware</li> <li>Propiedades de subredes y estaciones de la red</li> <li>Resumen: Comunicación mediante datos globales. Sinopsis</li> <li>Asignar y editar mensajes de símbolos</li> <li>Requisitos para cargar</li> </ul>

### 5.2.8 Objeto "Carpeta de fuentes"

Una carpeta de fuentes contiene programas fuente en forma de texto.

#### Posicionamiento en la ventana del proyecto

	<ul style="list-style-type: none"> <li>Objeto "Proyecto"</li> <li>Objeto "Equipo"</li> <li>Objeto "Módulo programable"</li> <li>Objeto "Programa S7/M7"</li> <li><b>Objeto "Carpeta de fuentes"</b></li> <li>Objeto "Carpeta de bloques"</li> </ul>
---	---

Símbolo	Carpeta de objetos	Selección de funciones importantes
	Carpeta de fuentes	<ul style="list-style-type: none"> <li>• Reglas para programar en fuentes AWL</li> <li>• Exportar fuentes</li> <li>• Importar fuentes</li> </ul>

Símbolo	Objetos en la carpeta de fuentes	Selección de funciones importantes
	Fuente (p. ej., fuente AWL)	<ul style="list-style-type: none"> <li>• Reglas para programar en fuentes AWL</li> <li>• Crear fuentes AWL</li> <li>• Insertar plantillas de bloques en fuentes AWL</li> <li>• Insertar en fuentes AWL el código de bloques existentes</li> <li>• Comprobar la coherencia en fuentes AWL</li> <li>• Compilar fuentes AWL</li> <li>• Generar fuentes AWL de bloques</li> <li>• Exportar fuentes</li> <li>• Importar fuentes</li> </ul>
	Plantilla de segmento	<ul style="list-style-type: none"> <li>• Crear una plantilla de segmento</li> <li>• Insertar una plantilla de segmento en un programa</li> </ul>

### 5.2.9 Programa S7/M7 sin equipo ni CPU

Es posible crear programas sin haber configurado previamente un equipo SIMATIC. Con ello, el usuario puede trabajar sin considerar el módulo a programar ni los ajustes del mismo.

#### Crear el programa S7/M7

1. Abra el proyecto que corresponda con el comando de menú **Archivo > Abrir** o active la ventana del proyecto.
2. Seleccione el proyecto en la ventana offline del proyecto.
3. Dependiendo del sistema de destino para el que esté creando el programa, elija el comando de menú:  
**Insertar > Programa > Programa S7**, si desea ejecutar su programa en un SIMATIC S7  
o  
**Insertar > Programa > Programa M7**, si desea ejecutar su programa en un SIMATIC M7.

El programa S7/M7 se inserta directamente bajo el proyecto. Contiene una carpeta para los bloques y una tabla de símbolos vacía. Ahora puede proceder a crear bloques y programarlos.

### **Asignación a un módulo programable**

Los programas insertados que no estén asignados a ningún módulo pueden ser posteriormente asignados a un módulo en la ventana del proyecto mediante Drag&Drop (Arrastrar y soltar), al copiarlos o arrastrarlos al símbolo del módulo.

### **Integrar en librerías**

Si el programa está destinado al sistema de destino SIMATIC S7 y debe ser reutilizado igual que un "Software-Pool", entonces se puede insertar bajo una librería. Para efectuar un test, los programas se tienen que encontrar bajo un proyecto, ya que sólo entonces es posible establecer un enlace con el sistema de destino.

### **Acceso a un sistema de destino**

Elija la vista online del proyecto. En el diálogo de propiedades del programa puede ajustar las direcciones.

---

#### **Nota**

Al borrar equipos o módulos programables se le preguntará si también desea borrar el programa que contienen. Si responde "No", el programa se insertará directamente bajo el proyecto en calidad de "programa sin equipo".

---

## 5.3 Interface de usuario

### 5.3.1 Manejo orientado a objetos

#### Objetivo: fácil manejo

El interface gráfico permite que el manejo sea lo más intuitivo posible. Por lo tanto, encontrará objetos que conoce de su trabajo cotidiano, tales como equipos, módulos, programas y bloques.

Con STEP 7 es posible crear, seleccionar y manipular dichos objetos.

#### Comparación con el manejo orientado a las herramientas

En el manejo orientado a las herramientas, hasta ahora el método habitual, era necesario determinar primero qué herramienta se requería para solucionar una tarea específica y activar luego dicha herramienta.

En el manejo orientado a los objetos se determina qué objeto se desea editar, para después abrirlo y editarlo.

Por lo tanto, en el manejo orientado a los objetos no es necesario conocer una sintaxis especial de comandos. Los objetos se representan en la pantalla mediante iconos que se pueden abrir con comandos de menú o haciendo clic con el ratón.

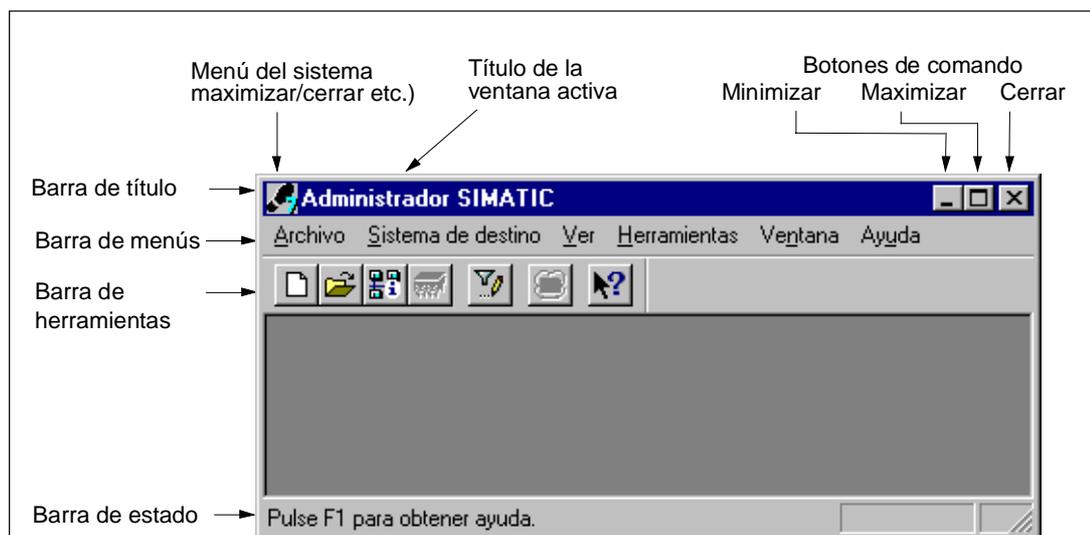
Al abrir un objeto se activa automáticamente la aplicación apropiada para visualizar o editar su contenido.

#### Siga leyendo ...

A continuación se describen las acciones básicas para editar objetos. Familiarícese con estas acciones puesto que se tratan de conceptos fundamentales que se repiten en el resto del manual.

### 5.3.2 Estructura de la ventana

La figura siguiente muestra los componentes estándar de una ventana:



## Barra de título y barra de menús

La barra de título y la barra de menús se encuentran siempre en el borde superior de la ventana. La barra de título contiene el título de la ventana y los botones para modificar el tamaño de la misma y para cerrarla. La barra de menús contiene todos los menús disponibles en la ventana.

## Barra de herramientas

La barra de herramientas contiene botones con los que es posible ejecutar rápidamente con un clic del ratón los comandos de menú de uso frecuente que estén disponibles en ese momento. Situando el puntero del ratón unos instantes en un botón, se obtiene breve información sobre su función. Además, en la barra de estado se visualiza una explicación adicional.

Si no es posible acceder a la configuración actual, los botones aparecen atenuados.

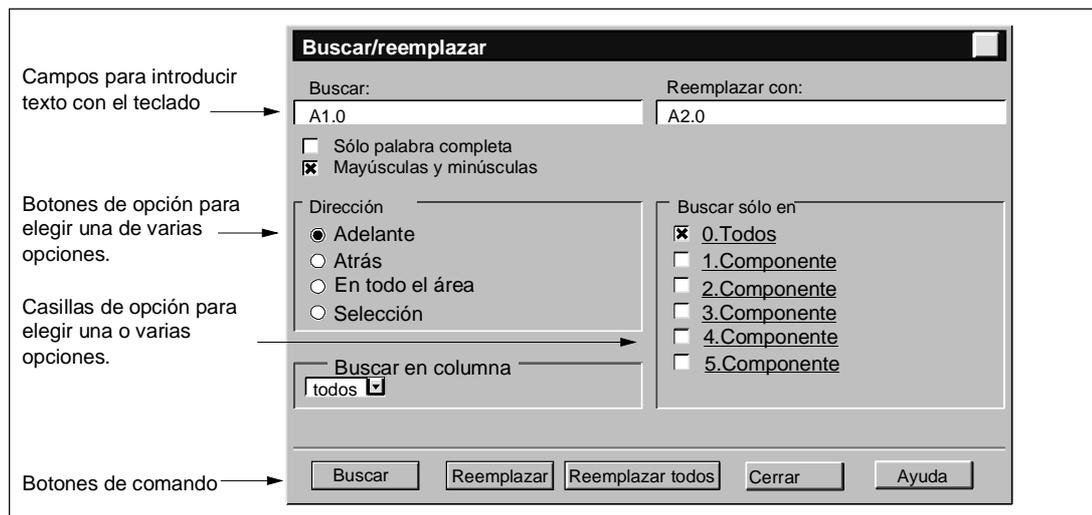
## Barra de estado

En la barra de estado se muestran informaciones contextuales.

### 5.3.3 Elementos de los cuadros de diálogo

#### Uso de los cuadros de diálogo

En los cuadros de diálogo es posible introducir informaciones necesarias para ejecutar una tarea determinada. Los componentes más frecuentes de los cuadros de diálogo se muestran en forma de ejemplo en la figura siguiente:



#### Cuadros de lista / cuadros combinados

Algunos cuadros de texto tienen una flecha que señala hacia abajo. Dicha flecha indica que para ese cuadro se dispone de más posibilidades de selección. Haga clic en la flecha para abrir un cuadro de lista o un cuadro combinado. Si hace clic en una de las entradas, ésta será adoptada automáticamente en el cuadro de texto.

## Cuadros de diálogo con fichas

Algunos cuadros de diálogo comprenden diversas fichas para facilitar la orientación (v. la figura siguiente).



El título de cada una de las fichas se indica en la pestaña correspondiente. Para que la ficha deseada se visualice en primer plano, haga clic en su título.

### 5.3.4 Crear y manejar objetos

Algunas funciones básicas son aplicables a todos los objetos. A continuación se ofrece una sinopsis de las mismas. Al describirse procedimientos en los siguientes apartados del manual se da por supuesto que el usuario conoce dichas funciones básicas.

El modo habitual de proceder al manipular objetos es el siguiente:

- crear objetos,
- seleccionar objetos,
- realizar acciones con el objeto (p. ej., copiar, borrar, etc.).

### Ajustar la ruta para guardar nuevos proyectos o librerías

Antes de crear nuevos proyectos o librerías por primera vez, es necesario ajustar la ruta donde se guardarán dichos objetos. Elija para ello el comando **Herramientas > Preferencias**. En la ficha "General" del cuadro de diálogo que aparece entonces es posible determinar la ruta donde se han de guardar los nuevos proyectos o librerías.

### Crear objetos

El Asistente STEP 7 "Nuevo proyecto" ayuda a crear nuevos proyectos y a insertar objetos. Para llamarlo, elija el comando **Archivo > Asistente 'Nuevo proyecto'**. En los cuadros de diálogo que aparecen entonces puede determinar la estructura de su proyecto y hacerlo crear por el Asistente.

Si no desea utilizar el Asistente, puede crear proyectos y librerías con el comando **Archivo > Nuevo**. Dichos objetos constituyen el nivel superior de una jerarquía. Todos los demás objetos de la jerarquía - a no ser que se creen automáticamente - pueden ser creados utilizando los comandos del menú "Insertar". No obstante, los módulos de un equipo SIMATIC constituyen una excepción, ya que sólo pueden ser generados por el Asistente 'Nuevo Proyecto' al configurarse el hardware.

## Abrir objetos

Existen varias alternativas para abrir un objeto en modo de visualización "Detalles":

- hacer doble clic en el icono del objeto o
- seleccionar el objeto y elegir el comando Edición > Abrir objeto. Esta alternativa sólo puede ser aplicada a objetos que no sean carpetas.

Una vez abierto el objeto, su contenido se puede editar o modificar.

Una vez abierto un objeto de este último tipo, su contenido se muestra en una nueva ventana para que pueda ser editado. No es posible modificar objetos cuyo contenido esté siendo utilizado.

---

### Nota

Excepción: Los equipos aparecen en forma de carpetas para módulos programables (al hacer doble clic) y para la configuración de equipos. La aplicación para configurar el hardware se arranca bien haciendo doble clic en el objeto "Hardware", o bien seleccionando el equipo y eligiendo el comando de menú **Edición > Abrir objeto**.

---

## Estructurar la jerarquía de objetos

Utilice el Asistente 'Nuevo proyecto' para estructurar el árbol de objetos. Tras abrir una carpeta, se representan en la pantalla los objetos que contiene. Con el menú "Insertar" es posible crear entonces otros subobjetos, como p.ej. otros equipos de un proyecto. En dicho menú se pueden activar sólo aquellos comandos que permiten insertar aquellos objetos que son admisibles en la carpeta actual.

## Ajustar las propiedades de objetos

Las propiedades de objetos son atributos que determinan el comportamiento de los mismos. El cuadro de diálogo para ajustar dichas propiedades aparece automáticamente al crear un objeto. No obstante, éstas se pueden también modificar posteriormente.

Con el comando **Edición > Propiedades del objeto** se llama un cuadro de diálogo donde se pueden visualizar o ajustar las propiedades del objeto seleccionado.

Con el comando **Edición > Propiedades especiales del objeto** es posible llamar diversos cuadros de diálogo e introducir los datos necesarios para manejo y visualización (M+V) y para configurar mensajes.

Por ejemplo, para llamar las propiedades especiales de un bloque para manejo y visualización, dicho bloque se deberá haber preparado para M+V. Es decir, en la ficha "Atributos" de las propiedades del bloque es preciso ajustar el atributo "S7\_m\_c" con el valor "true".

---

### Nota

- Las propiedades de la carpeta "Datos de sistema" y del objeto "Hardware" no se pueden visualizar ni modificar.
  - Los cuadros de diálogo de propiedades de los objetos de un proyecto protegido contra escritura no se pueden editar. En este caso, los campos de entrada aparecen atenuados.
  - Si desea visualizar las propiedades de los módulos programables, los parámetros visualizados no se podrán editar por motivos de coherencia. Para poder editar los parámetros deberá llamar la aplicación "Configurar hardware".
  - Si se modifican en el sistema de origen los ajustes realizados en objetos (p.ej. los datos de parametrización de un módulo), estos cambios no tendrán efecto en el sistema de destino. Para que tengan efecto, los bloques de datos de sistema en los que se guardan estos ajustes tienen que encontrarse en el sistema de destino.
  - Si se carga un programa de usuario entero, los bloques de datos de sistema se transfieren automáticamente. Si, tras haber cargado el programa, se modifican los ajustes, puede volver a cargar el objeto "Datos de sistema" para transferir los ajustes al sistema de destino.
  - Se recomienda editar la carpeta únicamente con STEP 7, ya que ésta puede tener una estructura física diferente de la que se ve en el Administrador SIMATIC.
- 

### Cortar, pegar y copiar

La mayoría de los objetos se pueden cortar, pegar o copiar siguiendo el procedimiento habitual en Windows. Los comandos correspondientes se encuentran en el menú "Edición".

Asimismo, con "arrastrar y soltar" (drag&drop) es posible copiar objetos. Si se señala un destino no admisible, aparecerá un cursor de prohibición.

Al copiar un objeto se copian también todos los niveles subordinados que contiene, lo que permite reutilizar componentes que se hayan creado en otra solución de automatización.

---

### Nota

La tabla de enlaces en la carpeta "Enlaces" no se puede copiar. Tenga en cuenta que al copiar listas de textos importantes para el usuario sólo se adoptarán los idiomas instalados en el objeto de destino.

---

El tema Copiar objetos contiene instrucciones paso a paso al respecto.

### Cambiar el nombre de objetos

El Administrador SIMATIC asigna nombres estandarizados a los nuevos objetos que se hayan insertado. Esos nombres resultan por lo general del tipo de objeto y, si se pueden crear varios objetos del mismo tipo en una misma carpeta, de un número correlativo.

Por ejemplo, el primer programa S7 se llamará "Programa S7(1)" y el segundo, "Programa S7(2)". En cambio, la tabla de símbolos se llama sólo "Símbolos", puesto que existe sólo una vez en cada carpeta de orden superior.

Si lo desea, puede cambiar los nombres de la mayoría de los objetos (incluyendo el nombre de un proyecto) por denominaciones que se autoexplicativas.

En los proyectos, los nombres de los directorios indicados en la ruta pueden comprender 8 caracteres como máximo. De lo contrario pueden surgir problemas al archivar y al emplear "C para M7" (compilador Borland).

Para cambiar el nombre de un objeto (en caso de que se pueda modificar) puede elegir una de las siguientes posibilidades:

- Edición directa:  
Haciendo dos veces clic lentamente en el nombre de un objeto seleccionado, aparecerá un marco alrededor del texto. Entonces podrá introducir el nombre utilizando el teclado.
- Con el comando "Propiedades del objeto":  
Seleccione el objeto deseado y luego el comando **Edición > Propiedades del objeto**. Cambie el nombre en el cuadro de diálogo. Cuando cierre el cuadro se habrá cambiado la denominación del objeto, el cual se visualizará entonces con su nuevo nombre.

Si no es posible cambiar el nombre de un objeto, aparecerá atenuado el campo de entrada en el cuadro de diálogo. Entonces se visualizará el nombre actual, siendo imposible introducir texto.

---

#### **Nota**

Si, durante la edición, el puntero del ratón se desplaza fuera del campo del nombre para realizar otra acción (p.ej. para elegir un comando de menú), el proceso de edición finalizará. El nuevo nombre será adoptado en caso de que sea admisible.

---

En el tema Cambiar el nombre de objetos encontrará instrucciones detalladas al respecto.

### **Mover objetos**

Con el Administrador SIMATIC se puede desplazar los objetos de una carpeta a otra, aunque la carpeta de destino forme parte de un proyecto diferente. Al desplazar una carpeta se traslada también todo su contenido.

---

#### **Nota**

Los siguientes objetos no se pueden desplazar:

- Enlaces
  - Bloques de datos de sistema (SDB) en la vista online
  - Funciones de sistema (SFC) y bloques de función de sistema (SFB) en la vista online
- 

El tema Desplazar objetos contiene instrucciones detalladas al respecto.

### **Ordenar objetos**

En el modo de visualización "Detalles" (comando de menú **Ver > Detalles**), los objetos se pueden ordenar según sus atributos. A tal efecto, haga clic en el título del atributo deseado. Haciendo nuevamente clic se invierte la secuencia de ordenamiento. Los bloques se ordenan conforme a su complemento numérico, p.ej. FB 1, FB 2, FB 11, FB 12, FB 21, FC 1.

### Secuencia de ordenamiento preajustada:

Cuando se abre un proyecto, los objetos se abren en el modo de visualización "Detalles" conforme a la secuencia de ordenamiento preajustada. Ejemplo:

- Los bloques se representan en el siguiente orden: "Datos de sistema, OB, FB, FC, DB, UDT, VAT, SFB, SFC"..
- En el proyecto se visualizan primero todos los equipos y luego los programas S7.

Por tanto, el ajuste estándar del modo de visualización "Detalles" no es una secuencia alfanumérica en orden ascendente o descendente.

### Restablecer la secuencia de ordenamiento preajustada:

Tras haberse cambiado el orden, p.ej. haciendo clic en el título de la columna "Nombre del objeto", se puede restablecer la secuencia de ordenamiento preajustada. Proceda a tal efecto de la siguiente forma:

- En el modo de visualización "Detalles", haga clic en el título de la columna "Tipo".
- Cierre el proyecto y ábralo de nuevo.

### Borrar objetos

Tanto las carpetas como los objetos se pueden borrar. Si borra una carpeta se borrarán también todos los objetos contenidos en ella.

El proceso de borrado es irreversible. Si no está completamente seguro de que ya no necesita un objeto, es preferible que archive previamente el proyecto entero.

---

#### Nota

Los siguientes objetos no se pueden borrar:

- Enlaces
  - Bloques de datos de sistema (SDB) en la vista online
  - Funciones de sistema (SFC) y bloques de función de sistema (SFB) en la vista online
- 

El tema Borrar objetos contiene instrucciones paso a paso al respecto.

### 5.3.5 Elegir objetos en cuadros de diálogo

En diversas situaciones es necesario elegir objetos en un cuadro de diálogo.

#### Llamar el cuadro de diálogo

El cuadro de diálogo se llama, como p.ej. en el caso de la configuración del hardware, mediante comandos de menú tales como **Equipo > Nuevo.../Abrir...** (a excepción de la ventana de acceso "Administrador SIMATIC").

#### Estructura del cuadro de diálogo

La figura siguiente muestra las opciones del cuadro de diálogo.

The image shows a screenshot of the 'Abrir' (Open) dialog box in STEP 7 software. The dialog box has a title bar 'Abrir' and several sections:

- Acceso:** A dropdown menu set to 'Proyecto'.
- Nombre Proyecto:** A dropdown menu set to 'muestra'.
- Vista:** A dropdown menu set to 'Vista de componentes'.
- Ruta:** A dropdown menu set to 'C:\SIEMENS\STEP7\E'.
- Online/Offline:** Radio buttons for 'Online' and 'Offline', with 'Offline' selected.
- Examinar...:** A button with a magnifying glass icon.
- Tree View:** A tree structure showing folders like 'Muestra', 'Red MPI 1', 'Subred SINEC L2 1', 'Subred SINEC H1 1', 'Equipo SIMATIC 300 1', and 'Programa S7'.
- Nombre del objeto:** An empty text input field.
- Tipo de objeto:** A dropdown menu set to 'Todos los ejecutables'.
- Buttons:** 'Aceptar', 'Cancelar', and 'Ayuda' at the bottom.

Annotations with arrows point to various parts of the dialog box:

- Punto de acceso:** elija el tipo de objeto desde el que desea iniciar la búsqueda (p. ej. "Proyecto", "Librería", pero también entradas que permitan el acceso a unidades de disco o autómatas conectados).
- Vista:** permite elegir la vista de componentes o la vista tecnológica.
- Online/Offline:** si el punto de acceso es "Proyecto" puede elegir la vista offline (selección de datos de proyecto en la PG/PC) u online (selección de datos del proyecto en el sistema de automatización).
- Examinar:** haga clic en este botón de comando para buscar objetos que no figuran en la lista.
- Nombre:** aquí aparece una lista de los objetos del tipo indicado bajo Acceso. Elija un nombre de la lista o tecléelo.
- Tipo de objeto:** aquí puede indicar un filtro para filtrar sólo un tipo determinado de objetos de los que indica la lista. De este modo se limita la cantidad de objetos representados.
- Nombre del objeto:** una vez elegido el objeto se registra aquí su nombre. También es posible introducirlo directamente.
- Vista del proyecto:** aquí aparece el árbol de objetos, los cuales a su vez pueden contener otros objetos.
- Vista Detalle:** aquí aparece el contenido del objeto seleccionado a la izquierda.

### 5.3.6 Memoria de sesión

El Administrador SIMATIC memoriza el contenido de la ventana, es decir, los proyectos y librerías que están abiertos así como la disposición de las ventanas.

- Con el comando de menú **Herramientas > Preferencias** puede hacer que se memorice el contenido y la disposición de las ventanas al finalizar la sesión. Cuando empiece la siguiente sesión, la ventana se abrirá con el mismo contenido y la misma disposición. En los proyectos abiertos, el cursor se posiciona en la última carpeta que se abrió.
- Con el comando de menú **Ventana > Guardar organización** se guarda el contenido actual de la ventana así como su disposición.
- Con el comando de menú **Ventana > Restablecer organización** se restablece el contenido y la disposición de la ventana que se guardó con el comando **Ventana > Guardar organización** . En los proyectos abiertos el cursor se posiciona en la última carpeta que se seleccionó.

---

#### Nota

El contenido de las ventanas de proyectos on-line, así como de la ventana "Estaciones accesibles" y el de la ventana "Memory Card S7" no se memoriza.

Las posibles contraseñas que haya introducido para limitar el acceso a los sistemas de destino (S7-300/S7-400) no se memorizan para otras sesiones .

---

### 5.3.7 Cambiar la organización de las ventanas de la tabla de símbolos

Para representar en cascada las ventanas correspondientes a todas las tablas de símbolos abiertas dispone de los siguientes procedimientos:

- Seleccionar el comando de menú **Ventana > Organizar > Cascada**.
- Pulsar simultáneamente las teclas MAYÚS + F5.

Para representar una debajo de otra y con el mismo tamaño todas las ventanas correspondientes a las tablas de símbolos abiertas seleccione el comando de menú **Ventana > Organizar > Mosaico horizontal**.

Para representar una al lado de otra y con el mismo tamaño todas las ventanas mostradas correspondientes a las tablas de símbolos abiertas seleccione el comando de menú **Ventana > Organizar > Mosaico vertical**.

Para representar con el mismo tamaño los iconos de las ventanas minimizadas junto al borde inferior de la ventana principal seleccione el comando de menú **Ventana > Organizar iconos**.

### 5.3.8 Guardar y restablecer la organización de las ventanas

Las aplicaciones de STEP 7 permiten guardar y restablecer posteriormente la organización actual de las ventanas. El ajuste se puede efectuar con los comandos de menú **Herramientas > Preferencias > General**.

#### ¿Qué se guarda?

Al guardar la organización de las ventanas se almacenan las siguientes informaciones:

- Posición de la ventana principal
- Proyectos y librerías abiertos y las correspondientes posiciones de las ventanas
- Orden de las ventanas que pudieran estar superpuestas

---

#### Nota

No se guarda el contenido de las ventanas de proyectos online, ni tampoco el de las ventanas "Estaciones accesibles" y "Memory Card S7".

---

#### Guardar la organización

Para guardar la organización actual de las ventanas, elija el comando **Ventana > Guardar organización**.

#### Restablecer la organización

Para restablecer las ventanas como estaban dispuestas previamente, elija el comando **Ventana > Restablecer organización**.

---

#### Nota

Al restablecerse una ventana, se representa detalladamente sólo la parte de la jerarquía donde se encuentra el objeto que estaba seleccionado al guardar la organización.

---

## 5.4 Manejo de las teclas

### 5.4.1 Manejo de las teclas

Denominación internacional de las teclas	Denominación española de las teclas
Tecla HOME	Tecla INICIO
Tecla END	Tecla FIN
Tecla PAGE-UP	Tecla AvPág
Tecla PAGE-DOWN	Tecla RePág
Tecla CTRL	Tecla CTRL
Tecla ENTER	Tecla INTRO
Tecla DEL	Tecla SUPR
Tecla INSERT	Tecla INSERT

### 5.4.2 Combinaciones de teclas para comandos de menú

Se puede activar cualquier comando de menú pulsando la siguiente combinación: ALT y la(s) tecla(s) correspondiente(s).

Pulse las teclas indicadas en el orden siguiente:

- Tecla ALT
- La letra que aparece subrayada en el menú deseado (p.ej. ALT+A para el menú "Archivo" - en el caso de que el menú "Archivo" aparezca en el menú). El menú se desplegará.
- La letra que aparece subrayada en el comando de menú deseado (p.ej. N para el comando "Nuevo"). Si se trata de un comando que comprenda submenús, éstos se desplegarán entonces. Proceda de igual forma hasta que haya seleccionado el comando de menú completo introduciendo las letras correspondientes.

Una vez introducida la última letra de la combinación de teclas se ejecutará el comando de menú.

Ejemplo:

#### Comandos de menú Teclas

Archivo > Archivar ALT, A, A

Archivo > Abrir ALT, A, A

### Teclas de método abreviado para comandos de menú

Función	Teclas de método abreviado
Nuevo (menú "Archivo")	CTRL+N
Abrir (menú "Archivo")	CTRL+O
Cerrar (menú "Archivo")	-
Compilar (menú "Archivo")	CTRL+B
Imprimir (objeto) (menú "Archivo")	CTRL+P
Salir (menú "Archivo")	ALT+F4
Copiar (menú "Edición")	CTRL+C
Cortar (menú "Edición")	CTRL+X
Pegar (menú "Edición")	CTRL+V
Borrar (menú "Edición")	SUPR
Seleccionar todo (menú "Edición")	CTRL+A
Propiedades del objeto (menú "Edición")	ALT+INTRO
Abrir objeto (menú "Edición")	CTRL+ALT+O
Cargar (menú "Sistema de destino")	CTRL+L
Estado operativo (menú "Sistema de destino")	CTRL+I
Actualizar (menú "Ver")	F5
actualiza la visualización del estado de las CPUs visibles en la vista online	CTRL+F5
Preferencias (menú "Herramientas")	CTRL+ALT+E
Datos de referencia, mostrar (menú "Herramientas")	CTRL+ALT+R
Cascada (menú "Ventana")	MAYÚS + F5
Mosaico horizontal (menú "Ventana")	MAYÚS + F2
Mosaico vertical (menú "Ventana")	MAYÚS + F3
Ayuda contextual (menú "Ayuda")	F1 (Si existe un contexto actual, p.ej., un comando de menú seleccionado, se llama al tema de Ayuda correspondiente. En caso contrario, al índice de la Ayuda.)

### 5.4.3 Combinaciones de teclas para desplazar el cursor

Desplazar el cursor en la barra de menús / en el menú emergente

<b>Función</b>	<b>Teclas</b>
A la barra de menús.	F10
Sobre el menú emergente.	MAYUS + F10
Al menú que contiene la letra X subrayada.	ALT+X
Comando de menú asociado	Letra subrayada en el comando de menú
Un comando de menú a la izquierda.	FLECHA IZQUIERDA
Un comando de menú a la derecha.	FLECHA DERECHA
Un comando de menú hacia arriba.	FLECHA ARRIBA
Un comando de menú hacia abajo.	FLECHA ABAJO
Activar el comando de menú seleccionado.	TECLA INTRO
Abandonar el menú y volver al texto.	ESC

### Desplazar el cursor al editar textos

<b>Función</b>	<b>Teclas</b>
Una línea hacia arriba o un carácter a la izquierda en un texto que se componga de una sola línea.	FLECHA ARRIBA
Una línea hacia abajo o un carácter a la derecha en un texto que se componga de una sola línea.	FLECHA ABAJO
Un carácter a la derecha.	FLECHA DERECHA
Un carácter a la izquierda.	FLECHA IZQUIERDA
Una palabra a la derecha.	CTRL+FLECHA DERECHA
Una palabra a la izquierda.	CTRL+FLECHA IZQUIERDA
Al comienzo de la línea.	INICIO
Al final de la línea.	FIN
Una pantalla hacia arriba.	RE PAG
Una pantalla hacia abajo.	AV PAG
Al comienzo del texto.	CTRL+INICIO
Al final del texto.	CTRL+FIN

## Desplazar el cursor en cuadros de diálogo

<b>Función</b>	<b>Teclas</b>
Al siguiente cuadro de entrada (de izquierda a derecha y de arriba hacia abajo).	TAB
Retrocede un cuadro de entrada.	Mayusc+TAB
Al cuadro de entrada que contiene la letra X subrayada.	ALT+X
Selecciona en la lista de selección	TECLAS DE CURSOR
Abre una lista de selección	ALT+FLECHA ABAJO
Activa y desactiva la selección de un objeto	BARRA ESPACIADORA
Confirma las entradas efectuadas y cierra el cuadro de diálogo (botón "Aceptar").	TECLA INTRO
Cierra el cuadro de diálogo sin guardar la selección (botón "Cancelar").	ESC

### 5.4.4 Combinaciones de teclas para seleccionar textos

<b>Función</b>	<b>Teclas</b>
Un carácter a la derecha.	MAYÚS+FLECHA DERECHA
Un carácter a la izquierda.	MAYÚS+FLECHA IZQUIERDA
Al comienzo de la línea de comentario	MAYUS + INICIO
Al final de la línea de comentario	MAYUS + FIN
Una línea arriba.	MAYÚS+FLECHA ARRIBA
Una línea abajo.	MAYÚS+FLECHA ABAJO
Una pantalla hacia arriba.	MAYUS+REPAG
Una pantalla hacia abajo.	MAYUS+AVPAG
Principio de archivo.	CTRL + MAYUS + INICIO
Fin de archivo.	CTRL+MAYUS+FIN

### 5.4.5 Combinaciones de teclas para acceder a la Ayuda en pantalla

<b>Función</b>	<b>Teclas</b>
Abre la Ayuda.	F1 (Si existe un contexto actual, p.ej., un comando de menú seleccionado, se llama al tema de Ayuda correspondiente. En caso contrario, al índice de la Ayuda.)
Activa el icono de interrogación para la ayuda contextual	MAYUS+F1
Cierra la ventana de ayuda, vuelve a la ventana de trabajo	ALT + F4

### 5.4.6 Combinaciones de teclas para cambiar de una ventana a otra

<b>Función</b>	<b>Teclas</b>
Cambiar de una sección de ventana a otra	F6
Retornar a la sección de ventana anterior si no existe ninguna ventana acoplable	MAYÚS + F6
Cambiar de la ventana de documento a la ventana acoplable del documento (p.ej. a la ventana de declaración de variables) Si no existe una ventana acoplable se retorna a la sección de ventana anterior.	MAYÚS + F6
Cambiar de una ventana de documento a otra	CTRL + F6
Retornar a la ventana de documento anterior	MAYÚS + CTRL + F6
Cambiar entre ventanas que no sean de documento (marco de la aplicación y ventanas acoplables del mismo; al retornar al marco se cambia a la última ventana activa del documento )	ALT+F6
Retornar a la ventana precedente que no sea de documento	MAYÚS + ALT + F6
Cerrar la ventana activa	CTRL + F4



## 6 Elaboración de proyectos

### 6.1 Estructura de los proyectos

Los proyectos sirven para almacenar de forma ordenada los datos y programas necesarios para crear una tarea de automatización. Los datos que conforman un proyecto comprenden principalmente:

- los datos para configurar el hardware y parametrizar los módulos,
- los datos para configurar la comunicación por redes y
- los programas para los módulos programables.

Al crear un proyecto el objetivo principal consiste en programar y en poner a disposición dichos datos.

En un proyecto, los datos se depositan en forma de objetos. Dentro de un proyecto, los objetos están estructurados en forma de árbol (jerarquía del proyecto). La representación de la jerarquía en la ventana del proyecto es similar a la del Explorador de Windows 95/98/NT. Sólo los iconos de los objetos tienen un aspecto diferente.

El nivel superior de la jerarquía de los proyectos está estructurado de la siguiente forma:

1. Nivel : Proyecto
2. Nivel : Subredes, equipos o programas S7/M7
3. Nivel: Dependiendo del objeto que se encuentre en el segundo nivel.

#### Ventana del proyecto

La ventana del proyecto se divide en dos partes. En la mitad izquierda se representa la estructura en árbol del proyecto. En la mitad derecha aparece el contenido del objeto seleccionado a la izquierda, conforme a la visualización elegida (iconos grandes, iconos pequeños, lista o detalles) .

Para visualizar la estructura completa del proyecto en la mitad izquierda de la ventana, haga clic en la casilla "+". Aparecerá entonces una representación similar a la que muestra la figura siguiente.



A la cabeza de la jerarquía de objetos se encuentra el objeto "S7\_Pro1", que es el icono del proyecto entero. Se puede utilizar para visualizar las propiedades del objeto y sirve de carpeta de redes (para configurar redes), de equipos (para configurar el hardware) y de programas S7 o M7 (para crear el software). Si se selecciona el icono del proyecto, los objetos contenidos allí se visualizan en la mitad derecha de la ventana. Los objetos que encabezan la jerarquía (que pueden ser tanto proyectos como librerías) permiten acceder a los cuadros de diálogo para seleccionar objetos.

### Vista del proyecto

Puede visualizar en ventanas la estructura de proyectos para el contingente de datos del sistema de origen con el modo "offline", y la del correspondiente contingente de datos del sistema de destino con el modo "online".

Si está cargado el correspondiente paquete opcional, podrá disponer también de la vista del Administrador de instalaciones.

---

### Nota

La configuración del hardware y de las redes sólo se puede llevar a cabo en modo de visualización "offline".

---

## 6.2 Crear proyectos

### 6.2.1 Crear un proyecto

Para poder solucionar su tarea de automatización a nivel de gestión de un proyecto deberá crear primero un nuevo proyecto. El nuevo proyecto se crea en el directorio que haya indicado en la ficha "General ", una vez elegido el comando **Herramientas > Preferencias**.

#### Nota

El Administrador SIMATIC permite utilizar nombres de más de 8 letras. Sin embargo, el nombre del directorio del proyecto queda limitado a 8 caracteres, por lo que resulta imprescindible que los nombres de los proyectos se distingan ya en los primeros 8 caracteres. Entre mayúsculas y minúsculas no se hace distinción alguna.

Los temas Crear un proyecto manualmente o Crear un proyecto utilizando el Asistente contienen instrucciones detalladas para crear un proyecto.

### Crear un proyecto utilizando el Asistente

La manera más fácil de crear un nuevo proyecto es utilizando el Asistente 'Nuevo proyecto'. Para llamarlo, elija el comando **Archivo > Asistente 'Nuevo proyecto'**. El Asistente consulta los datos necesarios en diversos cuadros de diálogo y crea luego el proyecto. Además del equipo, la CPU, las carpetas de programas, fuentes y bloques, así como el OB1, se pueden seleccionar allí también los OBs para el tratamiento de errores y de alarmas.

La figura siguiente muestra un ejemplo de un proyecto creado utilizando el Asistente.



### Crear un proyecto manualmente

No obstante, un proyecto también se puede crear utilizando los comandos de menú **Archivo > Nuevo** en el Administrador SIMATIC. El proyecto contendrá ya el objeto "Redes MPI".

## Alternativas para el trabajo ulterior

Para seguir elaborando un proyecto dispone de máxima flexibilidad. Una vez creado un proyecto puede, p.ej.,

- configurar primero el hardware y crear luego el software, o bien
- crear primero el software sin haber configurado antes el hardware.

### 1a. alternativa: configurar primero el hardware

Si desea configurar primero el hardware, proceda de la forma descrita en el segundo tomo del manual "Configuración del hardware con STEP 7". Al configurar el hardware, se insertan automáticamente las carpetas necesarias para crear el software, es decir, la de "Programa S7" o la de "Programa M7". Proceda entonces a insertar los objetos necesarios para crear el programa. Por último, cree el software para los módulos programables.

### 2a. alternativa: crear primero el software

También es posible crear el software sin haber configurado antes el hardware. La configuración se puede realizar posteriormente. Para crear programas no es necesario haber configurado antes el hardware de un equipo.

Si desea crear primero el software:

1. Inserte en su proyecto las carpetas de software necesarias (Programa S7/M7 sin equipo ni CPU).  
Para ello basta con que decida si la carpeta "Programas" debe comprender programas para equipos S7 o M7.
2. Por último, cree el software para los módulos programables.
3. Configure el hardware.
4. Una vez configurado el hardware, asigne el programa S7 o M7 a una CPU.

## 6.2.2 Insertar equipos

En un proyecto, un equipo representa la configuración física del sistema de automatización, conteniendo los datos para configurar y parametrizar los diversos módulos.

Los nuevos proyectos que se hayan creado con el Asistente 'Nuevo proyecto' ya contienen un equipo. En caso contrario, el equipo se podrá crear mediante el comando de menú **Insertar > Equipo**.

Se puede elegir entre los siguientes equipos:

- Equipo SIMATIC 300
- Equipo SIMATIC 400
- Equipo H SIMATIC
- Equipo PC SIMATIC
- PC/PG
- SIMATIC S5
- Otros equipos, es decir, que no sean de SIMATIC S7/M7 o de SIMATIC S5

El equipo se inserta con un nombre estándar (p.ej. Equipo SIMATIC 300 (1), Equipo SIMATIC 300 (2) etc.). Los nombres de los equipos se pueden cambiar por denominaciones más explícitas.

El tema Insertar equipos contiene instrucciones paso a paso al respecto.

## Configurar el hardware

En la configuración del hardware se determinan la CPU y todos los módulos contenidos en su sistema de automatización, utilizando para ello un catálogo de hardware. La configuración del hardware se inicia haciendo doble clic en el equipo.

Una vez guardada y concluida la configuración del hardware, para cada módulo programable que se haya configurado se crean automáticamente un programa S7 o M7 como carpeta de software y una tabla de enlaces (objeto "Enlaces"). Los nuevos proyectos que se hayan creado con el Asistente 'Nuevo proyecto' ya contienen estos objetos desde el principio.

El tema Configurar el hardware contiene instrucciones detalladas al respecto. En el tema Procedimiento básico para configurar un equipo encontrará los pasos fundamentales.

## Crear una tabla de enlaces

Para cada módulo programable se crea automáticamente una tabla (vacía) de enlaces (objeto "Enlaces"). Dicha tabla se utiliza para definir los enlaces entre los módulos programables de una red. Tras abrirla se visualiza una ventana con una tabla para definir los enlaces entre los módulos programables.

Para obtener información detallada, consulte el tema Conectar equipos de un mismo proyecto a la red.

## Pasos siguientes

Tras elaborar la configuración del hardware se puede crear el software para los módulos programables (consulte también Insertar un programa S7 o M7).

### 6.2.3 Insertar un programa S7 o M7

El software para los módulos programables se deposita en carpetas de objetos. En el caso de los módulos SIMATIC S7, dicha carpeta se denomina "Programa S7" y, en el caso de los módulos SIMATIC M7, "Programa M7".

A modo de ejemplo, la figura siguiente muestra un programa S7 contenido en un módulo programable de un equipo SIMATIC 300.



#### Componentes ya existentes

Para cada módulo programable se crea automáticamente un programa S7 o M7 como carpeta de software.

El programa S7 contiene ya los siguientes componentes:

- una tabla de símbolos (objeto "Símbolos"),
- una carpeta "Bloques" con el primer bloque y
- una carpeta "Fuentes" para programas fuente.

El programa M7 contiene ya los siguientes componentes:

- una tabla de símbolos (objeto "Símbolos"),
- una carpeta "Bloques".

#### Crear bloques S7

Si desea crear programas AWL, KOP o FUP, seleccione el objeto "Bloques" ya creado y elija el comando de menú **Insertar > Bloque S7**. En el menú siguiente podrá elegir el tipo de bloque (p.ej. bloque de datos, tipo de datos de usuario (UDT), función, bloque de función, bloque de organización, tabla de variables (VAT)).

Una vez abierto el bloque (vacío), podrá introducir el programa en AWL, KOP o FUP, respectivamente. Para obtener más informaciones al respecto, consulte el tema Procedimiento básico para crear bloques lógicos y los manuales de AWL, KOP y FUP, respectivamente.

---

#### Nota

El objeto "Datos del sistema" (SDB), que puede encontrarse en el programa de usuario, lo crea el sistema. Aunque se puede abrir, no es posible modificar su contenido por motivos de coherencia. Sirve para modificar la configuración a posteriori, es decir, una vez cargado un programa, y para cargar los cambios en el sistema de destino.

---

### Utilizar bloques de librerías estándar

Para crear programas de usuario puede utilizar también bloques de las librerías estándar incluidas en el volumen de suministro. Para acceder a las librerías, elija el comando **Archivo > Abrir**. Para obtener más información sobre cómo utilizar las librerías estándar y para crear librerías propias, consulte el tema de la Ayuda en pantalla Trabajar con librerías.

### Crear fuentes/esquemas CFC

Si desea crear una fuente en un lenguaje de programación determinado, o bien crear un plano CFC, seleccione en el programa S7 el objeto "Fuentes" o "Planos" (esquemas), respectivamente, y elija luego el comando de menú **Insertar > Software S7**. En el menú siguiente, elija la fuente correspondiente al lenguaje de programación. Una vez abierta la fuente vacía, podrá introducir el programa. Para obtener más información al respecto, consulte el tema Reglas para programar en fuentes AWL.

### Crear programas para M7

Si desea crear programas para el sistema operativo RMOS de un módulo programable de la gama M7, seleccione en el programa M7 y elija luego el comando de menú **Insertar > Software M7**. En el menú siguiente, elija el objeto correspondiente al lenguaje de programación o al sistema operativo, respectivamente. Una vez abierto el objeto, podrá acceder al entorno de creación de programas.

### Crear tablas de símbolos

Al crear un programa S7 o M7, se genera automáticamente una tabla de (vacía) símbolos (objeto "Símbolos"). Una vez abierta, aparece la ventana "Editor de símbolos", visualizándose allí la tabla de símbolos. Para obtener más información, consulte el tema Introducir varios símbolos globales en la tabla de símbolos.

### Insertar fuentes externas

Es posible crear y editar archivos fuente con un editor ASCII cualquiera. Dichos archivos se pueden importar luego en un proyecto y compilar en bloques.

Los bloques creados al compilar una fuente importada se depositan en la carpeta "Bloques".

Para obtener más información, consulte el tema Insertar fuentes externas.

## 6.3 Elaborar un proyecto

### 6.3.1 Elaborar un proyecto

#### Abrir un proyecto

Para abrir un proyecto, elija primero el comando **Archivo > Abrir**. Seleccione luego un proyecto en los cuadros de diálogo siguientes. Entonces se abrirá la ventana del proyecto.

---

#### Nota

Si el proyecto no se visualiza en la lista de proyectos, haga clic en el botón "Examinar". Así aparecerá el cuadro de diálogo "Examinar", con el cual podrá buscar otros proyectos e integrar los que se encuentren en la lista de proyectos. Las entradas en la lista de proyectos se pueden modificar seleccionando el comando de menú **Archivo > Gestionar**.

---

#### Copiar un proyecto

Para copiar un proyecto, elija el comando de menú **Archivo > Guardar como** y guarde el proyecto con otro nombre.

Los componentes del proyecto, tales como equipos, programas, bloques etc. se copian utilizando el comando de menú **Edición > Copiar**.

Los temas Copiar un proyecto y Copiar un componente de un proyecto contienen instrucciones detalladas al respecto.

#### Borrar un proyecto

Un proyecto se borra utilizando el comando de menú **Archivo > Borrar**.

Los componentes del proyecto, tales como los equipos, los programas, los bloques etc. se borran con el comando de menú **Edición > Borrar**.

Los temas Borrar un proyecto y Borrar un componente de un proyecto contienen instrucciones detalladas al respecto.

### 6.3.2 Gestionar textos en varios idiomas

STEP 7 ofrece la posibilidad de exportar, traducir, volver a importar y ver en el idioma traducido los textos monolingües guardados en un proyecto.

Los siguientes tipos de textos se pueden gestionar en varios idiomas.

- Comentarios y títulos
  - Títulos y comentarios de bloques
  - Títulos y comentarios de segmentos
  - Comentarios de líneas y programas AWL
  - Comentarios de tablas de símbolos, tablas de declaración de variables, tipos de datos de usuario y bloques de datos
  - Comentarios, nombres de estado y de transición en programas HiGraph
  - Ampliaciones a nombres y comentarios de etapas en programas S7-GRAPH
- Textos visualizados
  - Textos de mensajes generados por STEP 7, S7-GRAPH, S7-HiGraph o S7-PDIAG
  - Librerías de textos del sistema.

#### Exportar

La exportación se realiza para todos los bloques y tablas de símbolos que hay por debajo del objeto seleccionado. Para cada tipo de texto se genera un archivo de exportación. Éste contiene una columna para el idioma de origen y otra para el de destino. Los textos del idioma de origen no deben modificarse.

#### Importar

Al importar se traspa al proyecto seleccionado el contenido de las columnas para el idioma de destino (columna derecha). Al hacerlo sólo se toman los textos para los que en la columna del idioma de origen se detecta una concordancia con un texto existente.

#### Cambiar idioma

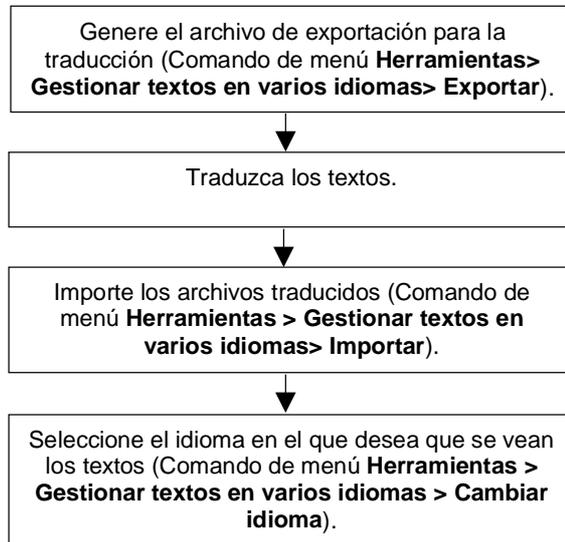
Al cambiar de idioma se pueden seleccionar todos los idiomas que se indicaron al importar en el proyecto seleccionado. A continuación se cambia el idioma de todos los objetos seleccionados.

#### Borrar idioma

Al borrar un idioma se borran todos los textos de este idioma de la gestión de datos interna.

En su proyecto siempre debería existir un idioma de referencia. Este podría ser, por ejemplo, el idioma de su país. Este idioma no se debería borrar. Al exportar e importar deberá ajustar siempre este idioma de referencia como idioma de origen. El idioma de destino podrá ajustarlo a su elección.

## Procedimiento básico



## 7 Definir símbolos

### 7.1 Direccionamiento absoluto y simbólico

En un programa de STEP 7 se utilizan operandos tales como señales de E/S, marcas, contadores, temporizadores, bloques de datos y bloques de función. Si lo desea, puede direccionar dichos operandos en su programa de forma absoluta. No obstante, la legibilidad del programa aumentará considerablemente si utiliza nombres simbólicos (p.ej. Motor\_A\_On, o bien, denominaciones usuales en su ramo). En el programa de usuario será posible entonces direccionar un operando mediante dicho símbolo.

#### Dirección absoluta

Una dirección absoluta se compone de un identificador de operando y de una dirección (p. ej., A 4.0, E 1.1, M 2.0, FB 21).

#### Direccionamiento simbólico

Asignándoles nombres simbólicos a las direcciones absolutas, será posible diseñar el programa de forma más clara y facilitar la corrección de errores.

STEP 7 puede traducir automáticamente los nombres simbólicos a las direcciones absolutas necesarias. Si prefiere acceder a los ARRAYS, STRUCTs, bloques de datos, datos locales, bloques lógicos y tipos de datos de usuario a través de los nombres simbólicos, deberá asignarles previamente éstos últimos a las direcciones absolutas antes de poder direccionar los datos de forma simbólica.

Por ejemplo, puede asignar al operando A 4.0 el nombre simbólico MOTOR\_ON y utilizar luego MOTOR\_ON como dirección en una instrucción del programa. A través de las direcciones simbólicas podrá reconocer fácilmente si los elementos del programa concuerdan con los componentes de su solución de automatización.

---

#### Nota

En un nombre simbólico (denominación de variable) no se pueden utilizar dos caracteres de subrayado seguidos(p.ej.: MOTOR\_\_ON).

---

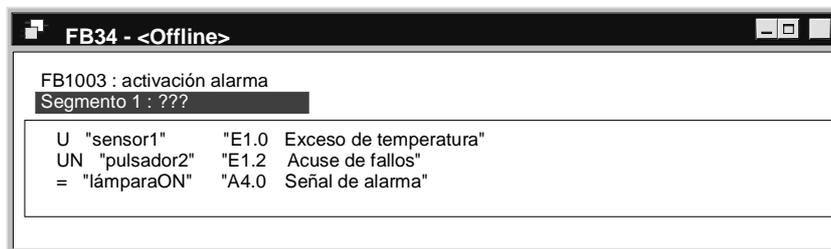
## Ayudas al introducir el programa

Las direcciones, los parámetros y los nombres de los bloques se pueden indicar de forma absoluta o bien en forma de símbolos en los lenguajes de programación KOP, FUP y AWL.

Con el comando de menú **Ver > Representación simbólica** puede conmutar entre la representación absoluta y la representación simbólica de las direcciones.

Para facilitar la programación con direcciones simbólicas, es recomendable visualizar las direcciones absolutas y los comentarios de los símbolos utilizados. Utilice para ello el comando de menú **Ver > Información del símbolo**. Procediendo así, tras cada instrucción AWL la línea de comentario se sustituye por dicha información. El texto visualizado no se puede editar. Los cambios se deben efectuar en la tabla de símbolos o en la tabla de declaración de variables.

La figura siguiente muestra una información del símbolo en AWL.



Al imprimir el bloque, se reproduce la representación actual en pantalla con el comentario de la instrucción o con el del símbolo.

## 7.2 Símbolos globales y locales

Los símbolos permiten utilizar denominaciones claras en vez de direcciones absolutas. Mediante la combinación de símbolos breves con comentarios más detallados puede crear programas eficientemente y buenas documentaciones.

Se distingue entre símbolos locales y símbolos globales.

	<b>Símbolos globales</b>	<b>Símbolos locales</b>
Ámbito de validez	<ul style="list-style-type: none"> <li>• Se pueden aplicar en todo el programa de usuario</li> <li>• Pueden ser utilizados por todos los bloques,</li> <li>• Tienen la misma importancia en todos los bloques,</li> <li>• Su denominación debe ser unívoca en todo el programa de usuario.</li> </ul>	<ul style="list-style-type: none"> <li>• Sólo se conocen en el bloque donde fueron definidos.</li> <li>• Es posible utilizar la misma denominación en diversos bloques para fines diferentes.</li> </ul>
Caracteres admisibles	<ul style="list-style-type: none"> <li>• Letras, cifras, caracteres especiales</li> <li>• Diéresis, excepto 0x00, 0xFF y comillas;</li> <li>• Al utilizarse caracteres especiales, el símbolo se deberá colocar entre comillas.</li> </ul>	<ul style="list-style-type: none"> <li>• Letras,</li> <li>• cifras,</li> <li>• carácter de subrayado (_),</li> </ul>
Aplicación	<p>Los símbolos globales se pueden definir para:</p> <ul style="list-style-type: none"> <li>• señales de E/S (E, EB, EW, ED, A, AB, AW, AD)</li> <li>• entradas y salidas periféricas (PE, PA)</li> <li>• marcas (M, MB, MW, MD)</li> <li>• temporizadores (T)/ contadores (Z)</li> <li>• bloques lógicos (OB, FB, FC, SFB, SFC)</li> <li>• bloques de datos (DB)</li> <li>• tipos de datos de usuario</li> <li>• tabla de variables (VAT)</li> </ul>	<p>Los símbolos locales se pueden definir para:</p> <ul style="list-style-type: none"> <li>• parámetros de bloques (parámetros de entrada, de salida y de entrada/salida),</li> <li>• datos estáticos de un bloque,</li> <li>• datos temporales de un bloque.</li> </ul>
Lugar de definición	Tabla de símbolos	Tabla de declaración de variables del bloque

### 7.3 Representación de símbolos globales y locales

En el área de instrucciones de un programa, los símbolos globales se pueden diferenciar de los locales de la siguiente forma:

- Los símbolos (globales) de la tabla de símbolos se representan entre comillas "..".
- Los símbolos (locales) de la tabla de declaración de variables se representan precedidos de un "#".

No es necesario introducir las comillas o el signo "#". Tras introducir el programa en KOP, FUP o AWL, la identificación se agrega una vez concluida la comprobación de sintaxis.

No obstante, si hubiera confusión, debido a que un mismo símbolo se utiliza tanto en la tabla de símbolos como en la declaración de variables, es preciso identificar explícitamente el símbolo global. De lo contrario, los símbolos no identificados se interpretarán como variables locales de un bloque.

Además, será preciso identificar los símbolos globales que contengan espacios en blanco.

En caso de programar en una fuente AWL se deberán respetar los mismos signos y reglas de utilización. En la entrada orientada a la fuente, las identificaciones no se complementan automáticamente. No obstante, sólo serán necesarias si pudiesen presentarse confusiones.

---

#### Nota

Con el comando de menú **Ver > Representación simbólica** puede optar entre visualizar los símbolos globales declarados o la correspondiente dirección absoluta.

---

## 7.4 Ajustar la preferencia de operandos

En el diálogo de propiedades del programa S7 se puede ajustar si, tras efectuar cambios en la tabla de símbolos, debe ser el símbolo o el valor absoluto el que predomine cuando se abran los bloques. En las versiones de STEP 7 anteriores a la 5 predomina siempre el valor absoluto. Al ejecutar el bloque CALL predomina siempre el valor absoluto.

### Ejemplo

Un bloque guardado contiene la instrucción "U Símbolo\_A" y Símbolo\_A se ha definido en la tabla de símbolos para el valor absoluto E0.1. La tabla de símbolos se modifica y el bloque se abre nuevamente. El ajuste de la preferencia del operando tendrá el siguiente efecto en la instrucción:

Preferencia del operando	Cambio de la asignación "Símbolo_A = E0.1"	Instrucción tras abrir el bloque	Explicación
Valor absoluto	Símbolo_A = E0.2	U E0.1	En la instrucción se visualiza el valor absoluto E0.1, puesto que ya no tiene asignado ningún símbolo.
Valor absoluto	Símbolo_B = E0.1	U Símbolo_B	En la instrucción se visualiza el nuevo símbolo del valor absoluto E0.1 que sigue siendo válido.
Símbolo	Símbolo_A = E0.2	U Símbolo_A	La instrucción se mantiene. Se visualiza un mensaje que indica que la asignación del símbolo ha cambiado.
Símbolo	Símbolo_B = E0.1	U Símbolo_A	La instrucción se marca como errónea (letra roja), puesto que Símbolo_A ya no está definido.

## 7.5 Tabla de símbolos para los símbolos globales

### 7.5.1 Tabla de símbolos para los símbolos globales

Los símbolos globales se definen en la tabla de símbolos.

Al crear un programa S7 o M7 se genera automáticamente una tabla de símbolos vacía (objeto "Símbolos").

#### Ámbito de validez

La tabla de símbolos será válida sólo para el módulo al que se asigne el programa. Si desea utilizar los mismos símbolos en diversas CPUs, deberá asegurarse de que coincidan las entradas correspondientes en las tablas de símbolos en cuestión (p. ej., copiándolas).

### 7.5.2 Estructura y componentes de la tabla de símbolos

#### Estructura de la tabla de símbolos



#### Columnas S/V/M/C

En las columnas S/V/M/C se indica si se han asignado propiedades especiales a un símbolo:

- S significa que para este símbolo se han creado definiciones de error para el diagnóstico de procesos con el paquete opcional S7-PDIAG (V5).
- V significa que el símbolo se puede manejar y visualizar con WinCC.
- M significa que al símbolo se le asignó un mensaje (SCAN).
- C significa que al símbolo se le han asignado propiedades de comunicación (sólo seleccionable con NCM).

## Símbolo

El nombre de un símbolo puede comprender 24 caracteres como máximo. Las tablas de símbolos pueden contener un máximo de 16380 símbolos.

En la tabla de símbolos no se pueden asignar nombres a los operandos de bloques de datos (DBD, DBW, DBB, DBX). Sus nombres están predefinidos por haberlos declarado en los bloques de datos.

Para bloques de organización (OBs), así como para algunos bloques de funciones de sistema (SFBs) y funciones de sistema (SFCs), la tabla de símbolos dispone de registros estándar que pueden ser importados a la tabla de símbolos de su programa S7 cuando ésta es editada. El archivo importado se encuentra en el directorio de STEP 7 bajo ...\\S7data\\Symbol\\Symbol.sdf.

## Dirección

Una dirección sirve para identificar un operando determinado.

Ejemplo: entrada E 12.1

Al introducir una dirección, se comprueba automáticamente si su sintaxis es correcta y si es posible asignar la dirección al tipo de datos indicado.

## Tipo de datos

Puede elegir entre los tipos de datos disponibles en STEP 7. En el campo aparece un tipo de datos predeterminado que se puede cambiar si es necesario. Si el cambio no concuerda con la dirección o si la sintaxis es errónea, aparecerá un mensaje de error al salir del campo.

## Comentario

Es posible asignar comentarios a cada uno de los símbolos. Combinando símbolos breves con comentarios detallados se crea tanto un programa efectivo como una buena documentación del mismo. Los comentarios comprenden un máximo de 80 caracteres.

## Conversión a variables C

Puede elegir símbolos de la tabla de símbolos de un programa M7 y, mediante el software opcional ProC/C++, convertirlos a variables C.

### 7.5.3 Direcciones y tipos de datos admisibles en la tabla de símbolos

Sólo hay una notación posible para toda la tabla de símbolos. La conmutación entre alemán (anteriormente SIMATIC) e inglés (anteriormente IEC) debe efectuarse en el Administrador SIMATIC eligiendo el comando de menú **Herramientas > Preferencias** y la ficha "Idioma".

Inglés	Alemán	Explicación:	Tipo de datos:	Direcciones:
I	E	Bit de entrada	BOOL	0.0..65535.7
IB	EB	Byte de entrada	BYTE, CHAR	0..65535
IW	EW	Palabra de entrada	WORD, INT, S5TIME, DATE	0..65534
ID	ED	Palabra doble de entrada	DWORD, DINT, REAL, TOD, TIME	0..65532
Q	A	Bit de salida	BOOL	0.0..65535.7
QB	AB	Byte de salida	BYTE, CHAR	0..65535
QW	AW	Palabra de salida	WORD, INT, S5TIME, DATE	0..65534
QD	AD	Palabra doble de salida	DWORD, DINT, REAL, TOD, TIME	0..65532
M	M	Bit de marcas	BOOL	0.0..65535.7
MB	MB	Byte de marcas	BYTE, CHAR	0..65535
MW	MW	Palabra de marcas	WORD, INT, S5TIME, DATE	0..65534
MD	MD	Palabra doble de marcas	DWORD, DINT, REAL, TOD, TIME	0..65532
PIB	PEB	Byte de entrada de periferia	BYTE, CHAR	0..65535
PQB	PAB	Byte de salida de periferia	BYTE, CHAR	0..65535
PIW	PEW	Palabra de entrada de periferia	WORD, INT, S5TIME, DATE	0..65534
PQW	PAW	Palabra de salida de periferia	WORD, INT, S5TIME, DATE	0..65534
PID	PED	Palabra doble de entrada de periferia	DWORD, DINT, REAL, TOD, TIME	0..65532
PQD	PAD	Palabra doble de salida de periferia	DWORD, DINT, REAL, TOD, TIME	0..65532
T	T	Temporizador	TIMER	0..65535
C	Z	Contadores	COUNTER	0..65535
FB	FB	Bloque de función	FB	0..65535
OB	OB	Bloque de organización	OB	0..65535
DB	DB	Bloque de datos	DB, FB, SFB, UDT	0..65535
FC	FC	Función	FC	0..65535
SFB	SFB	Bloque de función de sistema	SFB	0..65535
SFC	SFC	Función de sistema	SFC	0..65535
VAT	VAT	Tabla de variables		0..65535
UDT	UDT	Tipo de datos de usuario	UDT	0..65535

## 7.5.4 Símbolos incompletos y ambiguos en la tabla de símbolos

### Símbolos incompletos

Existe la posibilidad de guardar símbolos incompletos. Así puede indicar en primer lugar, por ejemplo el nombre y fijar más tarde la dirección. La ventaja es que puede interrumpir en cualquier momento el trabajo en la tabla de símbolos y guardar el estado intermedio. No obstante, para poder utilizar el símbolo mientras se va creando el programa (sin que aparezca un mensaje de error), deberá haber introducido el nombre del mismo, la dirección y el tipo de datos.

### Crear símbolos ambiguos

Un símbolo será ambiguo si se introduce en la tabla con un nombre (símbolo) o dirección que ya existan en otro símbolo. En tal caso, tanto el símbolo que existía ya como el que se ha insertado se declararán ambiguos.

Dicho caso se presenta, por ejemplo, cuando se copia y se inserta un símbolo, modificando luego ligeramente la entrada en la copia.

### Identificar símbolos ambiguos

Los símbolos ambiguos se destacan gráficamente (color, tipo de letra) en la tabla de símbolos. Ello indica que es necesario corregirlos. Es posible hacerse mostrar todos los símbolos o filtrar la visualización de manera que se vean sólo los símbolos unívocos o sólo los ambiguos.

### Eliminar la ambigüedad

Un símbolo ambiguo se convertirá en un símbolo unívoco cuando se modifique el nombre o la dirección que haya causado la ambigüedad. Si hay dos símbolos ambiguos y uno de ellos se corrige, el otro volverá a ser unívoco.

## 7.6 Métodos para introducir símbolos globales

### 7.6.1 Métodos para introducir símbolos globales

Existen tres métodos distintos para introducir los símbolos que se utilizarán en la programación:

- Entrada en un cuadro de diálogo  
Al editar el programa, puede abrir un cuadro de diálogo en una ventana y definir allí un nuevo símbolo. Este procedimiento es especialmente apropiado para definir símbolos de forma individual, como p.ej. si, mientras está programando, se da cuenta que hace falta un símbolo o que es necesario corregir un símbolo ya existente. De esta forma no es necesario visualizar la tabla de símbolos.
- Entrada en la tabla de símbolos  
En la "tabla de símbolos" se pueden introducir directamente los símbolos y las direcciones absolutas correspondientes. Este procedimiento es recomendable para introducir varios símbolos a la vez y crear por primera vez la tabla de símbolos. Los símbolos ya definidos se visualizan entonces en la pantalla, ofreciendo así una visión de conjunto.
- Importar tablas de símbolos de otros editores de tablas  
Los datos de la tabla de símbolos también se pueden crear con un editor de tablas diferente (p.ej. Microsoft Excel), importando después a la tabla de símbolos el archivo creado con dicho editor.

### 7.6.2 Observaciones generales para introducir símbolos

Para poder introducir símbolos en la tabla, vaya a la primera línea vacía de la misma y rellene los campos. Para insertar nuevas líneas delante de la línea actual en la tabla de símbolos, ejecute el comando **Insertar > Símbolo**. Si la línea que se encuentra encima de la posición del cursor ya contiene un operando, al insertar nuevos símbolos se rellenarán automáticamente las columnas "Dirección" y "Tipos de datos": se adoptará la dirección que contenga la línea anterior y se introducirá el tipo de datos predeterminado.

Si desea modificar entradas ya existentes, utilice los comandos del menú "Edición". A continuación, guarde y cierre la tabla de símbolos. También es posible guardar símbolos que no se hayan terminado de definir.

Al introducir símbolos en la tabla hay que prestar atención a las siguientes particularidades:

Columna	Nota
Símbolo	El nombre debe ser unívoco en toda la tabla de símbolos. Al aceptar el nombre en este campo y al salir del mismo, se le indicará si el símbolo es ambiguo. El símbolo puede tener como máximo 24 caracteres. No se admiten las comillas ".
Dirección	Si acepta el contenido de este campo o sale del mismo, se comprobará si la dirección dada es válida.
Tipo de datos	Después de introducir la dirección, a este campo se le asigna un valor predeterminado. Si lo modifica, entonces se comprueba si el nuevo tipo de datos corresponde a la dirección.
Comentario	Aquí puede introducir notas para describir brevemente (máx. 80 caracteres) las funciones del símbolo. Este campo de comentario es opcional.

### 7.6.3 Introducir símbolos globales en un cuadro de diálogo

Aquí se describe cómo modificar símbolos existentes o definir nuevos símbolos en un cuadro de diálogo al programar bloques y sin tener que abrir la tabla de símbolos.

Este procedimiento resulta muy útil si desea editar un solo símbolo. Para modificar varios símbolos es recomendable abrir la tabla de símbolos y trabajar allí directamente.

#### Activar la visualización de símbolos en el bloque

Estando abierto un bloque, visualice los símbolos eligiendo el comando **Ver > Representación simbólica**. Una marca de verificación delante del comando de menú indica que está activada la representación simbólica.

#### Definir símbolos al introducir el programa

1. Asegúrese de que en la ventana del bloque esté activada la representación simbólica (comando **Ver > Representación simbólica**).
2. En el área de instrucciones de su programa, seleccione la dirección absoluta que desee asignar a un símbolo.
3. Seleccione el comando de menú **Edición > Símbolo**.
4. Rellene el cuadro de diálogo que se visualiza entonces. Introduzca un símbolo y cierre el cuadro.

El símbolo definido se introduce en la tabla de símbolos. Los datos que pudieran causar símbolos ambiguos se rechazarán con un mensaje de error.

#### Editar en la tabla de símbolos

Con el comando de menú **Herramientas > Tabla de símbolos** puede abrir la tabla de símbolos para su edición.

## 7.6.4 Introducir símbolos globales en la tabla de símbolos

### Abrir una tabla de símbolos

Existen dos posibilidades para abrir una tabla de símbolos:

- Haciendo doble clic en la tabla de símbolos en la ventana del proyecto,
- Seleccionando la tabla de símbolos en la ventana del proyecto y activando luego el comando **Edición > Abrir objeto**.

La tabla de símbolos del programa actual se muestra en una ventana por separado. Ahora se puede proceder a crear o modificar símbolos. La tabla de símbolos está vacía cuando se abre por primera vez.

### Introducir símbolos

Para poder introducir nuevos símbolos en la tabla, vaya a la primera línea vacía de la misma y rellene los campos. Para insertar nuevas líneas vacías delante de la línea actual en la tabla de símbolos, active el comando **Insertar > Símbolo**. Las entradas ya existentes se pueden copiar con los comandos del menú "Edición" y modificar luego. A continuación, guarde y cierre la tabla de símbolos. También es posible guardar símbolos que no se hayan terminado de definir.

### Ordenar símbolos

Los registros de la tabla de símbolos se pueden ordenar alfabéticamente por símbolos, por direcciones, por tipos de datos o por comentarios.

El criterio de ordenación, se puede modificar p.ej. en el cuadro de diálogo que se abre al elegir el comando de menú **Ver > Ordenar....**

### Filtrar símbolos

Activando filtros puede seleccionar aquella parte de los registros de la tabla de símbolos que desee visualizar.

Con el comando **Ver > Filtrar** se abre el cuadro de diálogo "Filtrar".

Allí puede definir los criterios que deben cumplir los registros para que se puedan filtrar. Se puede filtrar por:

- nombres, direcciones, tipos de datos, comentarios
- símbolos con atributo de M+V, símbolos con propiedades de comunicación, símbolos para variables binarias en mensajes (marca o entrada del proceso)
- símbolos con estado "válido", "no válido (ambiguo, incompleto)"

Los criterios están combinados entre sí mediante Y (AND). Los registros filtrados comienzan con las secuencias de caracteres indicadas.

Si desea obtener más información acerca del cuadro de diálogo "Filtrar", abra la Ayuda en pantalla pulsando la tecla de función F1.

## 7.6.5 Mayúsculas y minúsculas en los símbolos

### Entre mayúsculas y minúsculas ya no se hace distinción.

En las versiones anteriores de STEP 7 se podían definir símbolos que se diferenciaban sólo por la escritura en mayúsculas o minúsculas de algunos caracteres. Esta característica ha sido modificada a partir de la versión 4.02 de STEP 7. A partir de ésta última, ya no es posible diferenciar los símbolos por su escritura en mayúsculas o minúsculas. Esta modificación responde al deseo de nuestros clientes, puesto que así se reducen considerablemente las posibles causas de errores en los programas. Limitando así la definición de símbolos se apoyan también los objetivos de PLCopen relativos a la definición de un estándar para los programas transferibles.

De ahora en adelante no se asistirá la diferenciación de los símbolos basada sólo en el uso de mayúsculas o minúsculas. Hasta ahora se permitía, por ejemplo, la siguiente diferenciación en la tabla de símbolos:

Motor1 = E 0.0

motor1 = E 1.0

Dichos símbolos se diferenciaban por la escritura (mayúsculas/minúsculas) de la primera letra. Este tipo de diferenciación da lugar a numerosas confusiones. Gracias a la definición actual se ha eliminado esta posible causa de errores.

### Posibles repercusiones en programas existentes

Si ha utilizado hasta ahora este criterio de diferenciación al definir los símbolos, es posible que se presenten conflictos debido a la nueva definición en caso de que:

- los símbolos **sólo** se diferencien entre sí por su escritura en mayúsculas/minúsculas
- los parámetros **sólo** se diferencien entre sí por su escritura en mayúsculas/minúsculas
- los símbolos **sólo** se diferencien de los parámetros por su escritura en mayúsculas/minúsculas

No obstante, dichos tres casos se pueden analizar y solucionar como se describe a continuación.

### **Símbolos que sólo se diferencien por su escritura en mayúsculas/minúsculas**

#### Conflicto:

Si la tabla de símbolos no se ha editado todavía con la versión actual del software, al compilarse los archivos fuente se utilizará el símbolo ambiguo que se encuentre más arriba en la tabla de símbolos.

Si la tabla de símbolos ya se ha editado, dichos símbolos no serán válidos, es decir, que al abrir los bloques no se visualizarán los símbolos, en tanto que los archivos fuente que los utilicen ya no se podrán compilar sin errores.

#### Remedio:

Compruebe si su tabla de símbolos contiene errores. Para ello, ábrala y guárdela nuevamente. Ello permite reconocer los símbolos ambiguos. Estos sólo se podrán visualizar y corregir a través del filtro "Símbolos ambiguos". Corrija luego los archivos fuente que presenten conflictos. En los bloques no será necesario efectuar cambios, puesto que al abrirlos utilizan automáticamente la tabla de símbolos actual (que ya no contiene conflictos).

### **Parámetros que sólo se diferencien por su escritura en mayúsculas/minúsculas**

#### Conflicto:

Los archivos fuente que contengan dichos interfaces ya no se podrán compilar sin errores. Aunque los bloques que contengan dichos interfaces todavía se pueden abrir, no es posible acceder al segundo de dichos parámetros. Al guardar, los accesos al segundo parámetro se aplican automáticamente al primer parámetro.

#### Remedio:

Para comprobar qué bloques contienen dichos conflictos, es recomendable generar un archivo fuente para todos los bloques de un programa utilizando la función "Generar fuente". Si se presentan errores al intentar compilar de nuevo el archivo fuente creado, significa que existe un conflicto.

Corrija los archivos fuente haciendo unívocos los parámetros, por ejemplo, mediante la función "Buscar/reemplazar" y compile dichos archivos nuevamente.

### **Símbolos que sólo se diferencien de los parámetros por su escritura en mayúsculas/minúsculas**

#### Conflicto:

Si los símbolos globales y locales de un archivo fuente sólo se diferencian por su escritura en mayúsculas/minúsculas y si no se han utilizado caracteres de identificación para los símbolos globales ("nombre simbólico") o locales (#nombre simbólico), al compilar se utilizará siempre el símbolo local. Ello cambiará el código máquina.

#### Remedio:

En este caso es recomendable generar nuevamente una fuente para cada uno de los bloques. Así se les adjudican automáticamente los correspondientes caracteres de identificación a los accesos locales y globales, procesándose correctamente en los siguientes procesos de compilación.

## 7.6.6 Exportar e importar tablas de símbolos

La tabla de símbolos visualizada se puede exportar a un archivo de texto, p.ej., para modificarla con un editor cualquiera.

Asimismo, es posible importar a la tabla de símbolos otras tablas que se hayan creado con una herramienta diferente para seguir editándolas en la primera. La función "Importar" permite, p.ej., incorporar en la tabla de símbolos las listas de asignación que se hayan creado con STEP5/ST (después de su conversión).

Es posible elegir entre los formatos de archivos \*.SDF, \*.ASC, \*.DIF y \*.SEQ.

### Reglas para exportar

De la tabla de símbolos se puede exportar bien su totalidad, bien una parte, o bien sólo las líneas que se hayan seleccionado.

No es posible exportar las propiedades de los símbolos que se ajusten con el comando Edición Propiedades especiales del objeto.

### Reglas para importar

- Para bloques de función de sistema (SFB), funciones de sistema (SFC) y bloques de organización (OB) de uso frecuente, encontrará símbolos predefinidos en el archivo ...\\S7DATA\SYMBOL\SYMBOL.SDF que podrá utilizar en caso necesario.
- Las propiedades de los símbolos que se ajusten con el comando **Edición > Propiedades especiales del objeto** no se tienen en cuenta ni al exportar ni al importar la tabla.

## 7.6.7 Formatos de archivos para importar/exportar una tabla de símbolos

Se pueden importar/exportar los siguientes formatos de archivo desde/hacia la tabla de símbolos:

- Formato ASCII (ASC)
- Data Interchange Format (DIF)  
Los archivos DIF se pueden abrir, editar y guardar de nuevo en Microsoft Excel.
- System Data Format (SDF)  
Los archivos SDF se pueden abrir, editar y guardar de nuevo en Microsoft Excel.
  - Para la importación y exportación de datos de la aplicación Access de Microsoft utilice el formato SDF.
  - En ACCESS, elija el formato de datos "Texto (con carácter separador)".
  - Utilice las comillas (") como carácter separador de textos.
  - Utilice la coma (,) como carácter separador de campos.
- Lista de asignación (SEQ)  
**Cuidado:** ¡Al exportar la tabla de símbolos a un archivo del tipo SEQ, los comentarios que excedan los 40 caracteres quedarán cortados tras el 40º carácter!

### Formato ASCII (ASC)

<b>Tipo de archivo:</b>	*.ASC
<b>Estructura:</b>	Longitud del registro, Coma separadora, Registro
<b>Ejemplo:</b>	126,verde_fase_peat T 2 TIMER Duración de la fase verde para peatones 126,rojo_peat A 0.0 BOOL Rojo para peatones

### Data Interchange Format (DIF)

<b>Tipo de archivo:</b>	*.DIF
<b>Estructura:</b>	Un archivo DIF se compone del encabezado (header) y los datos:

Encabezado	TABLE	Inicio de un archivo DIF
	0,1	
	"<Título>"	Cadena de caracteres de comentario
	VECTORS	Número de registros en el archivo
	0,<Número de registro>	
	""	
	TUPLES	Número de campos de datos de un registro
	0,<Cantidad de columnas>	
	""	
	DATA	Identificador de final del encabezado e inicio de datos
	0,0	
	""	
Datos (para cada registro)	<Tipo>,<valor numérico>	Identificador del tipo de datos, valor numérico.
	<Zeichenkette>	Sección alfanumérica, o
	V	Si no se utiliza la sección alfanumérica

**Encabezado:** El encabezado del archivo debe contener los tipos de registro TABLE, VECTORS, TUPLES y DATA en el orden indicado. En los archivos DIF, antes de DATA se pueden incluir opcionalmente otros tipos de registro. No obstante, éstos serán ignorados por el editor de símbolos.

**Datos:** En la sección de datos, cada registro consta de 3 partes: el identificador de tipo, un valor numérico y una sección alfanumérica.

Los archivos DIF se pueden abrir, editar y volver a guardar con Microsoft Excel. No obstante, hay que renunciar a la utilización de la diéresis y de otros caracteres especiales de diversos idiomas.

**System Data Format (SDF)**

<b>Tipo de archivo:</b>	*.SDF
<b>Estructura:</b>	Cadena de caracteres entre comillas, secciones separadas por comas
<b>Ejemplo:</b>	"verde_fase_peat","T 2","TIMER","Duración de la fase verde para peatones" "rojo_peat","A 0.0","BOOL","Rojo para peatones"

Para abrir un archivo SDF desde Microsoft Access, seleccione el formato de archivo "Texto (con caracteres de separación)". Introduzca las comillas como carácter separador de textos ("), y la coma (,) como carácter separador de campos.

**Lista de asignación (SEQ)**

<b>Tipo de archivo:</b>	*.SEQ
<b>Estructura:</b>	TAB Dirección TAB Símbolo TAB Comentario CR
<b>Ejemplo:</b>	T 2 verde_fase_peat Duración de la fase verde para peatones A 0.0 rojo_peat Rojo para peatones

TAB representa el carácter de tabulación (09H),  
CR representa un salto de línea con la tecla INTRO (0DH).



## 8 Crear bloques y librerías

### 8.1 Seleccionar el método de creación

Dependiendo del lenguaje de programación utilizado para representar el programa, se puede utilizar la edición incremental y/o la edición orientada a la fuente.

#### Editores incrementales para los lenguajes de programación KOP, FUP, AWL o S7-GRAPH

Con los editores incrementales para KOP, FUP, AWL o S7-GRAPH es posible crear bloques que se depositan en el programa de usuario. Es recomendable que elija la edición incremental si desea que sus entradas se comprueben inmediatamente. Este tipo de edición es adecuado también para los programadores principiantes. En la edición incremental se comprueba inmediatamente la sintaxis de cada línea o elemento. En caso de detectarse errores, éstos se visualizan inmediatamente, debiendo ser corregidos antes de finalizar la edición. Las entradas con sintaxis correcta se compilan automáticamente y se depositan en el programa de usuario.

Los símbolos utilizados se deben definir antes de editar las instrucciones. En caso de que no existan determinados símbolos, el bloque no se puede compilar por completo. No obstante, este "estado intermedio incoherente" se podrá guardar.

#### Editores (de texto) orientados a la fuente para los lenguajes de programación AWL, S7-SCL o S7-HiGraph

Con estos editores se crean **fuentes** que se compilan luego en bloques.

Elija la edición orientada a la fuente si desea introducir un programa rápidamente.

En este tipo de edición, el programa o un bloque se editan en un archivo de texto que se compila luego.

Los archivos de texto (fuentes) se depositan en la carpeta de fuentes de su programa S7, por ejemplo en forma de **fuelle AWL** o de **fuelle SCL**. Un archivo fuente puede contener la lógica de uno o varios bloques. Con los editores de texto de AWL o de SCL se puede crear el código para **OBs, FBs, FCs, DBs y UDTs** (tipos de datos de usuario), es decir, también para un programa de usuario entero. El programa entero de una CPU, es decir, todos los bloques, puede estar depositado en un solo archivo de texto.

Al compilar el archivo fuente se crearán los bloques que luego se depositarán en el programa de usuario. Antes de la compilación es preciso definir los símbolos utilizados. El compilador anunciará los posibles errores una vez que se empiece a compilar.

Por lo que respecta a la compilación, es importante tener en cuenta la sintaxis prescrita para el respectivo lenguaje de programación. La sintaxis sólo se verifica al comprobar la coherencia o al compilar la fuente en bloques.

## 8.2 Elegir el lenguaje de programación

### 8.2.1 Elegir el lenguaje de programación

#### Definir el lenguaje de programación del editor

Al generar un bloque o un archivo fuente hay que indicar en las propiedades del objeto con qué lenguaje de programación y con qué editor se desea crear el bloque o la fuente. Conforme a su elección, se arrancará el editor correspondiente al abrir el bloque o el archivo fuente.

#### Abrir el editor de lenguaje

El editor de lenguaje se abre en el Administrador SIMATIC haciendo doble clic en el objeto correspondiente (bloque, archivo fuente, etc.), o bien activando el comando **Edición " Abrir objeto** o el botón correspondiente de la barra de herramientas.

Para crear el programa S7 se dispone de los lenguajes de programación indicados en la tabla. El software básico STEP 7 contiene los lenguajes de programación KOP, FUP y AWL. Los demás lenguajes de programación pueden adquirirse opcionalmente.

Gracias a ello tiene la posibilidad de elegir entre diversos lenguajes de programación (esquema de contactos, diagrama de funciones, lista de instrucciones, lenguaje de alto nivel, control secuencial o grafo de estado), pudiendo elegir también si desea programar de forma textual o gráfica.

Mediante la elección del lenguaje de programación se definen también los métodos de introducción admisibles (●).

Lenguaje de programación	Destinatarios	Caso de aplicación	Entrada incremental	Entrada orientada a fuente	Bloque autodocumentable de la CPU
Lista de instrucciones AWL	Usuarios que desean programar a nivel de máquina	Programas optimizados en tiempo de ejecución y uso de memoria	●	●	●
Esquema de contactos KOP	Usuarios familiarizados con esquemas eléctricos	Programación de controles combinacionales	●	—	●
Diagrama de funciones FUP	Usuarios familiarizados con los cuadros lógicos del álgebra de Bool	Programación de controles combinacionales	●	—	●

Lenguaje de programación	Destinatarios	Caso de aplicación	Entrada incremental	Entrada orientada a fuente	Bloque autodocumentable de la CPU
SCL (Structured Control Language) Paquete opcional	Usuarios que han programado con lenguajes de alto nivel tales como PASCAL o C	Programar tareas de procesamiento de datos	–	•	–
GRAPH Paquete opcional	Usuarios que desean programar conforme a la tecnología sin tener conocimientos profundos de programación/ autómatas programables	Descripción confortable de procesos secuenciales	•	–	•
HiGraph Paquete opcional	Usuarios que desean programar conforme a la tecnología sin tener conocimientos profundos de programación/ autómatas programables	Descripción confortable de procesos asíncronos, no secuenciales	–	•	–
CFC Paquete opcional	Usuarios que desean programar conforme a la tecnología sin tener conocimientos profundos de programación/ autómatas programables	Descripción de procesos continuos	–	•	–

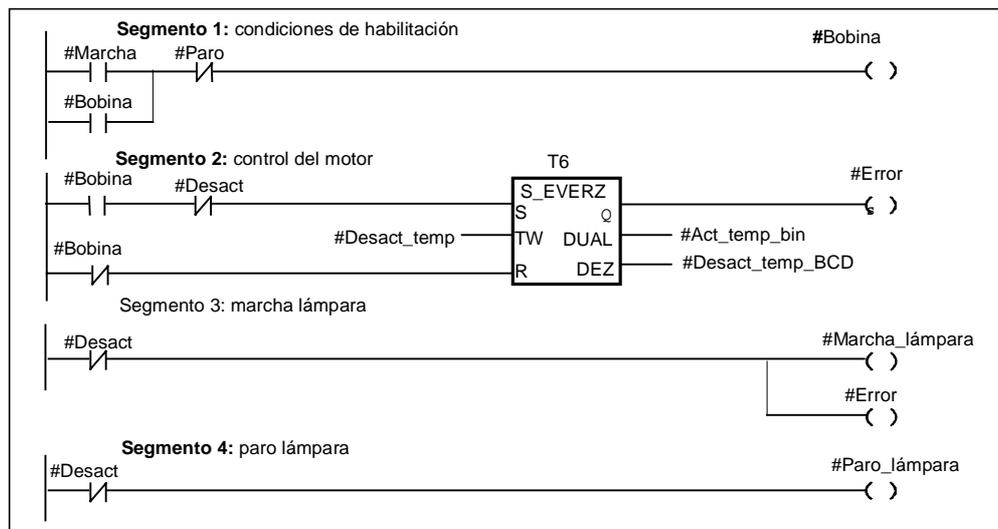
Es posible conmutar entre KOP, FUP y AWL si el bloque en cuestión no contiene errores. Las partes del programa no representables en el lenguaje de destino se representan en AWL.

Los bloques se pueden crear en AWL partiendo de archivos fuente y también se pueden volver a compilar en archivos fuente.

## 8.2.2 Lenguaje de programación KOP (esquema de contactos)

La representación del lenguaje de programación gráfico KOP (esquema de contactos) es similar a la de los esquemas de circuitos. Los elementos de un esquema de circuitos, tales como los contactos normalmente cerrados y normalmente abiertos, se agrupan en segmentos. Uno o varios segmentos constituyen el área de instrucciones de un bloque lógico.

### Ejemplo de segmentos en KOP



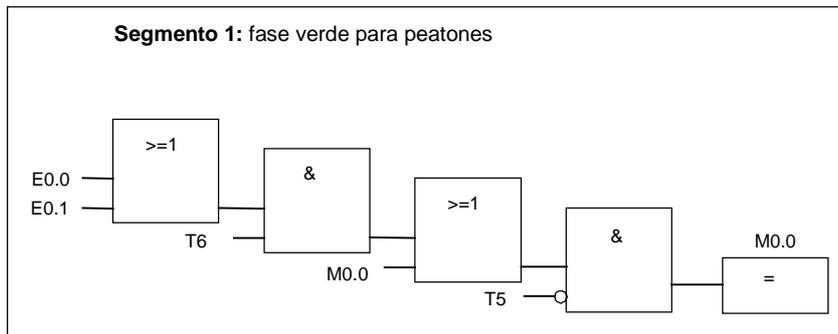
El lenguaje de programación KOP está incluido en el software básico STEP 7. Los programas se crean en KOP con un editor incremental.

## 8.2.3 Lenguaje de programación FUP (diagrama de funciones)

El lenguaje de programación FUP (diagrama de funciones) utiliza los símbolos gráficos del álgebra booleana para representar la lógica. También es posible representar en conexión directa con los cuadros lógicos funciones complejas, como por ejemplo funciones matemáticas.

El lenguaje de programación FUP está incluido en el software básico STEP 7.

### Ejemplo de un segmento en FUP:

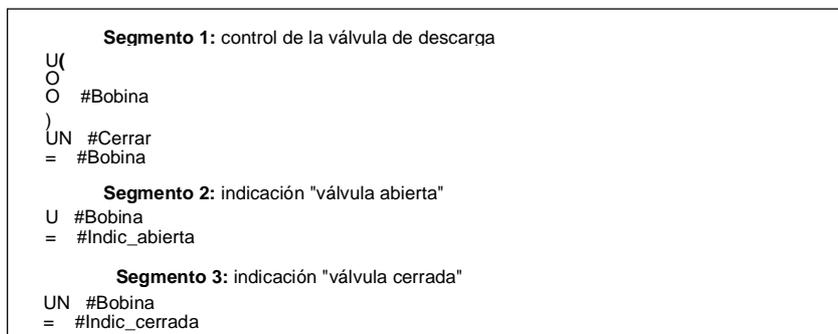


Los programas se crean en FUP con un editor incremental.

### 8.2.4 Lenguaje de programación AWL (lista de instrucciones)

El lenguaje de programación AWL (lista de instrucciones) es un lenguaje textual orientado a la máquina. Las diversas instrucciones equivalen a los pasos de trabajo con los que la CPU ejecuta el programa. Las instrucciones pueden ser agrupadas en segmentos.

#### Ejemplo de segmentos en AWL



El lenguaje de programación AWL está incluido en el software básico STEP 7. Con él puede editar bloques S7 de forma incremental o crear su programa en una fuente AWL con un editor orientado a la fuente para ser luego compilado en bloques.

## 8.2.5 Lenguaje de programación S7-SCL

El lenguaje de programación SCL (Structured Control Language), suministrado como software opcional, es un lenguaje textual de alto nivel cuya definición equivale básicamente a la norma IEC 1131-3. Este lenguaje, similar al PASCAL, ayuda a simplificar - en contraposición a AWL - gracias a sus instrucciones de alto nivel, como p.ej. la programación de bucles y de ramificaciones condicionadas. Por esta razón, SCL es especialmente apropiado para el cálculo de fórmulas, para complejos algoritmos de optimización o para gestionar grandes cantidades de datos.

Los programas se crean en S7-SCL con un editor orientado a la fuente (en una fuente SCL).

### Ejemplo:

```
FUNCTION_BLOCK FB 20
VAR_INPUT
ENDWERT:          INT;
END_VAR
VAR_IN_OUT
IQ1   :          REAL;
END_VAR
VAR
INDEX:          INT;
END_VAR

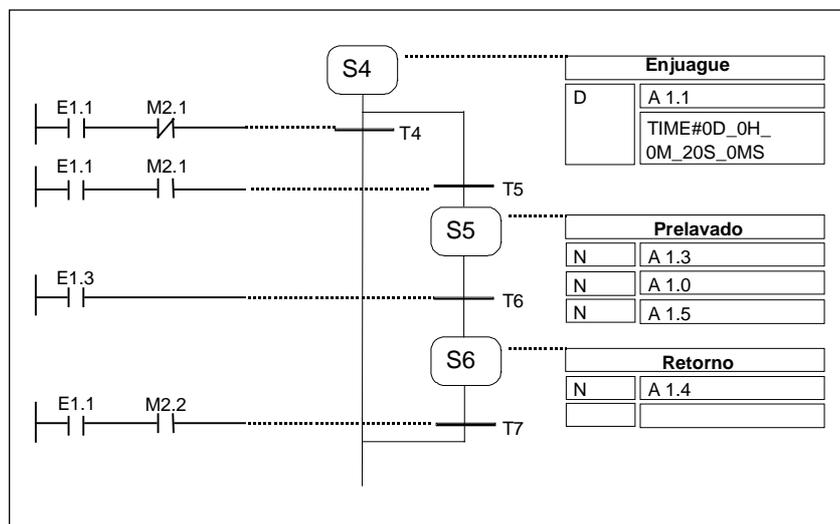
BEGIN
CONTROL:=FALSE;
FOR INDEX:= 1 TO ENDWERT DO
    IQ1:= IQ1 * 2;
    IF IQ1 >10000 THEN
        CONTROL = TRUE
    END_IF;
END_FOR;
END_FUNCTION_BLOCK
```

## 8.2.6 Lenguaje de programación S7-GRAPH (control secuencial)

El software opcional S7-GRAPH es un lenguaje gráfico de programación que permite programar controles secuenciales. Comprende la creación de una cadena de etapas, la definición de los contenidos de las mismas y las condiciones de transición. El contenido de las etapas se define con un lenguaje de programación especial (similar a AWL), en tanto que las condiciones de transición se introducen en una representación del esquema de contactos (parte del lenguaje de programación KOP).

S7-GRAPH permite representar también procesos complejos de forma muy clara, permitiendo así una programación y una búsqueda de errores efectivas.

### Ejemplo de un control secuencial en S7 GRAPH



### Crear bloques

Con el editor de S7-GRAPH se programa el bloque de función que contiene la cadena de etapas. Los datos de la cadena de etapas, tales como los parámetros del FB, así como las condiciones de etapas y de transición, están contenidos en un bloque de datos de instancia asignado. Con el editor de S7-GRAPH es posible crear automáticamente el DB de instancia mencionado.

### Archivo fuente

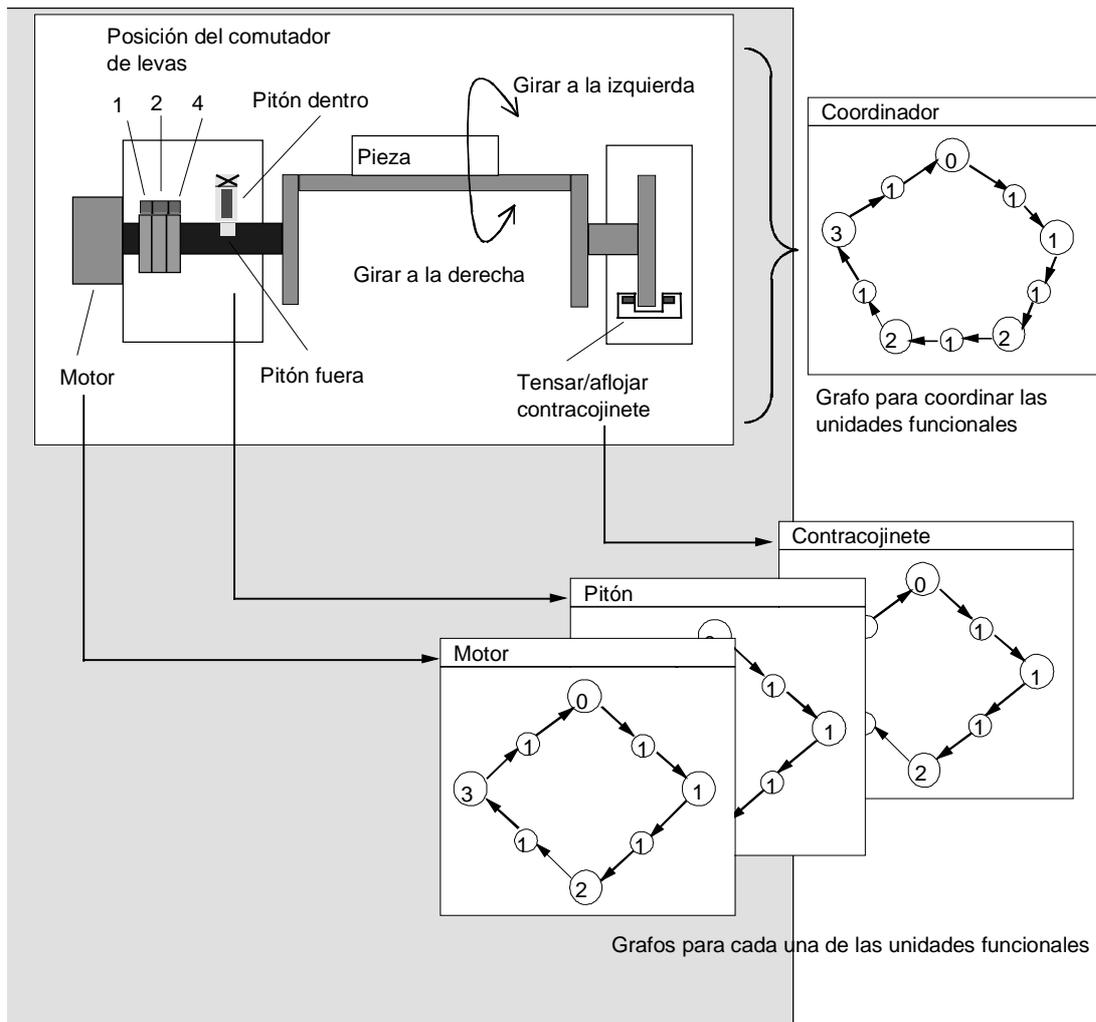
A partir de un bloque de función creado en S7-GRAPH es posible generar un archivo fuente textual (fuente GRAPH) que puede ser interpretado con paneles de operador o visualizadores de textos para visualizar el control secuencial.

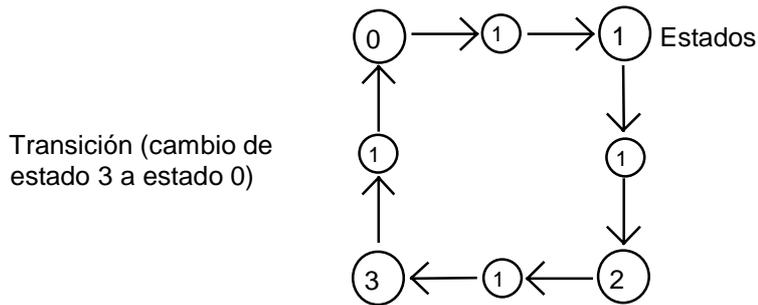
## 8.2.7 Lenguaje de programación S7-HiGraph (grafo de estado)

El lenguaje gráfico de programación S7-HiGraph (opcional) permite programar algunos bloques de su programa en forma de grafo de estado. Para ello debe subdividir su instalación en unidades funcionales independientes que puedan adoptar diversos estados. Para cambiar de un estado a otro debe definir transiciones. Las acciones asociadas a los estados y las condiciones de transición entre los mismos se describen en un lenguaje detallado similar a AWL.

Para cada unidad funcional se deberá crear un grafo que describa el comportamiento de la misma. Los grafos de una instalación se reúnen en grupos de grafos. Entre los diversos grafos es posible intercambiar mensajes para la sincronización de las unidades funcionales.

La representación esquemática de las transiciones de estado de una unidad funcional permite programar sistemáticamente y facilita la búsqueda de errores. Al contrario de S7-GRAPH, en S7-HiGraph sólo hay un estado ("etapa" en S7-GRAPH) activado en un determinado momento. La figura siguiente muestra la creación de grafos para unidades funcionales (ejemplo).





Un grupo de grafos se guarda en una fuente HiGraph en la carpeta "Fuentes" bajo el programa S7. Dicha fuente se compila luego en los bloques S7 del programa de usuario.

Una vez efectuada la última entrada de un grafo (al cerrar la ventana de trabajo) se comprueban la sintaxis y los parámetros formales. Las direcciones y los símbolos se comprueban al compilarse la fuente.

### 8.2.8 Lenguaje de programación S7-CFC

El software opcional CFC (*C*ontinuous *F*unction *C*hart) es un lenguaje de programación que permite interconectar gráficamente las funciones complejas.

En el lenguaje de programación S7-CFC se programa interconectando gráficamente las funciones existentes. No es necesario programar numerosas funciones estándar, puesto que se dispone de librerías que contienen bloques predefinidos (p.ej. para funciones lógicas, aritméticas, de regulación y de procesamiento de datos). Para poder utilizar el lenguaje CFC no se requieren conocimientos especiales de programación o sobre sistemas de automatización, lo que permite concentrarse en la técnica.

El programa creado se guarda en forma de esquemas CFC. Estos se depositan en la carpeta "Planos" bajo el programa S7. De dichos planos o esquemas se compilan luego los bloques S7 para el programa de usuario.

Los bloques a interconectar se programan para el SIMATIC S7 con uno de los lenguajes de S7, y para el SIMATIC M7, con C/C++.

## 8.3 Crear bloques

### 8.3.1 Carpeta de bloques

Puede crear programas para una CPU S7 en forma de

- Bloques
- Fuentes

Para depositar los bloques se dispone de la carpeta "Bloques" bajo Programa S7.

La carpeta "Bloques" contiene los bloques que debe cargar en la CPU S7 para llevar a cabo la solución de automatización. Los bloques lógicos (OBs, FBs, FCs) y los bloques de datos (DB) forman parte de dichos bloques cargables. El bloque lógico OB1 se crea automáticamente con la carpeta "Bloques", puesto que se necesita en todo caso para editar el programa en la CPU S7.

Además, la carpeta "Bloques" comprende los siguientes objetos:

- Los tipos de datos de usuario que usted haya definido. Estos simplifican la programación, pero no se cargan en la CPU.
- Las tablas de variables (VATs) que puede crear para observar y forzar variables, a fin de comprobar su programa. Las tablas de variables no se cargan en la CPU.
- El objeto datos de sistema" (bloques de datos de sistema), que contiene información sobre el sistema (configuración y parámetros del sistema). Estos bloques se crean al configurarse el hardware, abasteciéndose con datos.
- Las funciones de sistema (SFCs) y los bloques de función del sistema (SFBs) que desee llamar en su programa de usuario. Las SFCs y los SFBs no se pueden editar.

Con excepción de los bloques de datos del sistema (que se crean y se editan sólo a través de la configuración del sistema de automatización), los bloques del programa de usuario se editan siempre con el editor asignado. Este se arranca automáticamente haciendo doble clic en el bloque en cuestión.

---

#### Nota

En la carpeta "Bloques" también se guardan los bloques que se hayan programado en forma de fuentes y compilado posteriormente.

---

### 8.3.2 Tipos de datos de usuario (UDT)

Los tipos de datos de usuario son estructuras de datos creadas por usted mismo y que, una vez definidas, pueden utilizarse en todo el programa de usuario S7:

- como tipos de datos simples o como tipos de datos compuestos en la declaración de variables de bloques lógicos (FC, FB, OB) o como tipo de datos para variables en un bloque de datos (DB). La ventaja es que basta con definir una sola vez una estructura especial de datos que vaya a utilizar varias veces, pudiéndola asignar luego a cualquier cantidad de variables.
- como plantilla para crear bloques de datos de estructura idéntica, es decir que basta con elaborar una sola vez la estructura, pudiendo crear luego los bloques de datos necesarios mediante la simple asignación del UDT (por ejemplo, en el caso de las recetas: la estructura del DB es siempre igual, variando sólo las cantidades).

Al igual que los demás bloques los tipos de datos de usuario (UDTs) se crean en el Administrador SIMATIC o con el editor incremental.

#### Estructura de un UDT

Después de abrir un UDT se visualiza en una nueva ventana de trabajo la tabla de este tipo de datos de usuario en modo "Declaración".

- La primera y la última línea contienen ya las declaraciones STRUCT y END\_STRUCT, respectivamente, que indican el comienzo y el fin del tipo de datos de usuario. Estas dos líneas no se pueden editar.
- Puede editar el tipo de datos de usuario comenzando a partir de la segunda línea de la tabla de declaración en las columnas correspondientes.
- Los tipos de datos de usuario pueden estar formados por:
  - tipos de datos simples,
  - tipos de datos compuestos o
  - tipos de datos de usuario ya existentes.

Los tipos de datos de usuario del programa de usuario de S7 no pueden cargarse en la CPU S7. Se crean y se editan bien sea directamente con editores incrementales, o bien resultan de la compilación de fuentes.

### 8.3.3 Propiedades del bloque

Mediante las propiedades de bloques podrá identificar mejor los bloques que ha creado (p.ej., en las diferentes versiones) o proteger los bloques contra cambios no autorizados.

Para poder editar las propiedades debe estar abierto el bloque. Además de las propiedades editables, en el cuadro de diálogo correspondiente se visualizan también datos a título de información. Dichos datos no se pueden editar.

Aunque las propiedades de los bloques y los atributos de sistema se indican también en el Administrador SIMATIC, en las propiedades de objeto de un bloque, allí sólo se pueden editar las propiedades NAME, FAMILY, AUTHOR y VERSION.

Una vez insertado el bloque, las propiedades del objeto se editan desde el Administrador SIMATIC. En caso de que un bloque no haya sido creado con el Administrador SIMATIC, sino con uno de los editores, dichos ajustes (p.ej., el lenguaje de programación) se guardan automáticamente en las propiedades del objeto.

**Nota**

La nemotécnica para programar sus bloques S7 se elige en el Administrador SIMATIC con el comando **Herramientas > Preferencias** en la ficha "Idioma".

**Tabla de las propiedades de bloques**

Si desea introducir propiedades de bloques, deberá respetar el orden indicado en la tabla que aparece a continuación.

Palabra clave / propiedad	Significado	Ejemplo
[KNOW_HOW_PROTECT]	Protección de bloques: los bloques compilados con esta opción no permiten ver el área de instrucciones.	KNOW_HOW_PROTECT
[AUTHOR:]	Nombre del autor, nombre de la empresa, del departamento u otros nombres (máx. 8 caracteres, sin blancos)	AUTHOR : Siemens, pero ninguna palabra clave
[FAMILY:]	Nombre de la familia de bloques: p.ej. regulador. (máx. 8 caracteres, sin espacios en blanco).	FAMILY : Reguladores, pero ninguna palabra clave
[NAME:]	Nombre del bloque (máx. 8 caracteres)	NAME : PID, pero ninguna palabra clave
[VERSION: int1 . int2]	Número de versión del bloque (ambos números entre 0 y 15, es decir, 0.0 - 15.15)	VERSION : 3.10
[CODE_VERSION1]	Indica si un FB es multiinstancia. Si desea declarar multiinstancias, el FB no puede llevar dicha propiedad.	CODE_VERSION1
[UNLINKED] sólo para DBs	Un bloque de datos con la propiedad UNLINKED no se incluye en el programa.	
[READ_ONLY] sólo para DBs	Protección contra escritura para bloques de datos; sus datos sólo pueden leerse, pero no modificarse.	READ_ONLY

La protección de bloques KNOW\_HOW\_PROTECT tiene las siguientes consecuencias:

- Si desea visualizar posteriormente el bloque compilado en el editor AWL, FUP o KOP incremental, no podrá ver el área de instrucciones del bloque.
- En la tabla de declaración de variables del bloque se visualizan sólo las variables de tipo var\_in, var\_out y var\_in\_out. Las variables de tipo var\_stat y var\_temp permanecerán ocultas.

### Asignar propiedades a tipos de bloques

La siguiente tabla muestra qué propiedades pueden asignarse a qué tipos de bloques.

Propiedad	OB	FB	FC	DB	UDT
KNOW_HOW_PROTECT	•	•	•	•	–
AUTHOR	•	•	•	•	–
FAMILY	•	•	•	•	–
NAME	•	•	•	•	–
VERSION	•	•	•	•	–
UNLINKED	–	–	–	•	–
READ_ONLY	–	–	–	•	–

La propiedad KNOW\_HOW\_PROTECT se puede ajustar en una fuente al programar el bloque. Dicha propiedad se visualiza en el cuadro de diálogo de propiedades del bloque, pero no se puede modificar.

#### 8.3.4 Visualizar longitudes de bloques

Las longitudes de bloques se visualizan en la unidad "Byte".

#### Visualizar características de una carpeta de bloques

Las siguientes longitudes se muestran dentro de las características de una carpeta de bloques (vista offline):

- Tamaño (suma de todos los bloques sin datos del sistema) en la memoria de carga del sistema de destino
- Tamaño (suma de todos los bloques sin datos del sistema) en la memoria de trabajo del sistema de destino

No se muestran las longitudes de bloques en el sistema de origen (PG/PC) dentro de las propiedades de la carpeta de bloques.

#### Visualizar propiedades de un bloque

En las propiedades de un bloque se visualizan:

- Cantidad necesaria de datos locales: Tamaño de los datos locales (en bytes)
- MC7: tamaño del código MC7 (en bytes) o tamaño de los datos útiles DB
- Tamaño en la memoria de carga del sistema de destino
- Tamaño en la memoria de trabajo del sistema de destino: sólo se visualiza al conocerse la asignación del hardware.

Los datos se visualizarán independientemente de si el bloque se encuentra en una ventana de vista online u offline.

### Visualizar en el Administrador SIMATIC (Ver > Detalles)

Si la carpeta de bloques está abierta y "Ver > Detalles" activado, en la ventana del proyecto se visualizará la memoria de trabajo requerida, independientemente de si la carpeta de bloques se encuentra en una ventana de vista online u offline.

Puede sumar las longitudes de todos los bloques seleccionando los relevantes. En este caso, la suma se mostrará en la barra de estado del Administrador SIMATIC.

En el caso de los bloques no cargables en el sistema de destino (p.ej. VATs) no se muestra la longitud.

En Ver > Detalles no se muestran las longitudes de los bloques del sistema de origen (PG/PC).

### 8.3.5 Recablear

Se pueden recablear los siguientes bloques y operandos :

Entradas, salidas

Marcas, temporizadores, contadores

Funciones, bloques de función

Proceda como sigue:

1. En el Administrador SIMATIC, seleccione la carpeta "Bloques" en la que se encuentran los bloques que desea recablear.
2. Elija el comando de menú **Herramientas > Recablear**.
3. En el cuadro de diálogo "Recablear" elija los cambios deseados (Antiguo operando / Nuevo operando) en la tabla.
4. Elija la opción "Considerar todos los bits a los que accede el operando" si desea recablear todas las áreas de operandos (BYTE, WORD, DWORD).  
Ejemplo: Si se indican las áreas de operandos EW0 y EW4, los operandos E0.0 – E1.7 se recablearán en los operandos E4.0 – E5.7. Los operandos del área recableada (p.ej. E0.1) no se podrán introducir luego de forma individual en la tabla.
5. Haga clic en el botón "Aceptar".

Así se inicia el proceso de recableado. Tras el recableado se puede decidir en un cuadro de diálogo si se desea visualizar el archivo de información al respecto. El archivo de información contiene la lista de operandos, "Antiguo operando" y "Nuevo operando". Además, se indican los diversos bloques con la cantidad de recableados que se hayan efectuado en ellos.

Al recablear debe tenerse en cuenta que:

- al recablear un bloque (es decir, al cambiar el nombre), el nuevo bloque no debe existir previamente. Si el bloque ya existe, el proceso se interrumpirá.
- Al recablear un bloque de función (FB), su DB de instancia se asignará automáticamente al FB recableado. El DB de instancia no cambiará (es decir, se mantendrá el número de DB).

### 8.3.6 Atributos de bloques y parámetros

En los temas de Ayuda de los atributos de sistema se incluye una descripción de los atributos.

## 8.4 Trabajar con librerías

### 8.4.1 Trabajar con librerías

Las librerías sirven para depositar componentes reutilizables de programas para SIMATIC S7/M7. Los componentes de programas pueden copiarse de proyectos existentes a una librería, o bien, crearse directamente en la librería, independientemente de los proyectos.

La programación se puede simplificar en gran medida depositando en un programa S7 o en una librería los bloques que se deseen utilizar una y otra vez. De allí se podrán copiar siempre al programa de usuario en cuestión.

Para crear programas S7/M7 en una librería se dispone – a excepción de las funciones de test – de las mismas funciones que en los proyectos.

#### Crear librerías

Al igual que los proyectos, las librerías se pueden crear utilizando el comando de menú **Archivo > Nuevo**. La nueva librería se crea en el directorio que haya indicado en la ficha "General", una vez elegido el comando de menú **Herramientas > Preferencias**.

---

#### Nota

El Administrador SIMATIC permite utilizar nombres de más de 8 letras. Sin embargo, el nombre del directorio de la librería queda cortado a 8 caracteres. Por consiguiente, es imprescindible que los nombres de las librerías se distingan en los primeros 8 caracteres. Entre mayúsculas y minúsculas no se hace distinción. Al abrir se volverá a visualizar el nombre completo en el examinador; al examinar aparecerá el corto.

Tenga en cuenta que en los proyectos de una versión anterior de STEP 7 no se pueden utilizar bloques de librerías de una versión posterior de STEP 7.

---

#### Abrir librerías

Para abrir una librería, elija primero el comando **Archivo > Abrir**. Seleccione luego una librería en los cuadros de diálogo siguientes. Entonces se abrirá la ventana de la librería.

---

#### Nota

Si no aparece en la lista de librerías la librería deseada, haga clic en el botón "Examinar" del cuadro de diálogo "Abrir". Entonces se visualizará el cuadro de diálogo estándar de Windows con la estructura de directorios en el que podrá buscar la librería.

Tenga en cuenta que el nombre del archivo corresponde siempre al nombre original de la librería que se haya creado, es decir que los cambios de nombre efectuados en el Administrador SIMATIC no tienen efecto en el nivel de archivos.

Una vez seleccionada la librería, ésta se adoptará en la lista de librerías. Las entradas en la lista de librerías se pueden modificar seleccionando el comando de menú **Archivo > Gestionar**.

---

## Copiar librerías

Para copiar una librería, elija el comando de menú **Archivo > Guardar como** y guarde la librería con otro nombre.

Los componentes de la librería, tales como programas, bloques, fuentes, etc. se copian utilizando el comando de menú **Edición > Copiar**.

## Borrar librerías

Las librerías se borran utilizando el comando de menú **Archivo > Borrar**.

Los componentes de la librería, tales como programas, bloques, fuentes, etc. se borran utilizando el comando de menú **Edición > Borrar**.

### 8.4.2 Estructura jerárquica de las librerías

Al igual que los proyectos, las librerías están estructuradas jerárquicamente:

- Las librerías pueden contener programas S7/M7.
- Un programa S7 puede comprender un sola carpeta "Bloques" (programa de usuario), una carpeta "Fuentes" y una carpeta "Esquemas", así como un objeto "Símbolos" (tabla de símbolos).
- Un programa M7 puede contener planos y programas C para módulos M7 programables, así como un objeto "Símbolos" (tabla de símbolos) y una carpeta de bloques para DBs y VATs (tablas de variables).
- La carpeta "Bloques" contiene los bloques que se cargan en la CPU S7. Las tablas de variables (VAT) y los tipos de datos de usuario no se cargan en la CPU.
- La carpeta "Fuentes" comprende las fuentes de los programas creados en los diversos lenguajes de programación.
- La carpeta "Esquemas" (planos) contiene los esquemas CFC (sólo en el caso del software opcional S7-CFC).

Al insertar un nuevo programa S7/M7, se crean automáticamente la carpeta "Bloques", la carpeta "Fuentes" (sólo para S7) y el objeto "Símbolos".

### 8.4.3 Librerías estándar disponibles

El software estándar STEP 7 ofrece las siguientes librerías estándar (versión 2/versión 3):

- stlibs (V2): librería estándar para la versión 2
- Standard Library: librería estándar a partir de la versión 3

Las librerías estándar contienen los componentes siguientes:

- **Sytem Function Blocks:** funciones de sistema (SFC) y bloques de función de sistema (SFB)
- **S5-S7 Converting Blocks:** bloques para la conversión de programas STEP 5
- **TI-S7 Converting Blocks:** funciones estándar utilizables en general
- **IEC Function Blocks:** bloques para funciones IEC, tales como para editar indicaciones de fecha y hora, para operaciones de comparación, para el tratamiento de cadenas y para seleccionar el máximo y el mínimo
- **Organization Blocks:** bloques de organización estándar (OB)

La librería estándar para la versión 3 contiene además los componentes siguientes:

- **PID Control Blocks:** bloques de función (FBs) para el PID Control
- **Communication Blocks:** Funciones (FCs) y bloques de función (FBs) para CPs SIMATIC NET.

Si se instala software opcional se pueden añadir más librerías.

### Borrar e instalar las librerías adjuntas

Las librerías adjuntas se pueden borrar e instalar luego nuevamente en el Administrador SIMATIC. Para la instalación es preciso volver a ejecutar el programa de instalación Setup de STEP 7 V5.0.

---

#### Nota

Al instalarse STEP 7 las librerías adjuntas se copian automáticamente. Si ya las ha modificado, se sobrescribirán con las originales al instalarse STEP 7 de nuevo.

Por tanto, antes de efectuar cambios es recomendable copiar las librerías adjuntas y editar sólo las copias.

---



# 9 Crear bloques lógicos

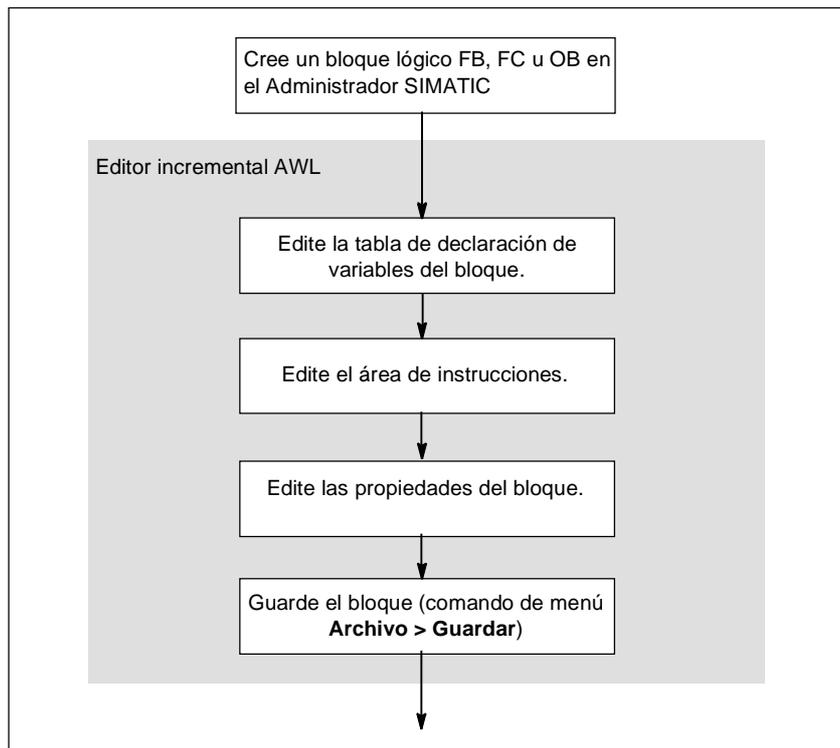
## 9.1 Nociones básicas para crear bloques lógicos

### 9.1.1 Procedimiento básico para crear bloques lógicos

Los bloques lógicos (OBs, FBs, FCs) se componen de una área de declaración de variables, una área de instrucciones y sus propiedades. En otras palabras, al programar se tienen que editar las tres partes siguientes:

- **Tabla de declaración de variables.** En la tabla de declaración de variables se definen los parámetros, los atributos de sistema de los parámetros y las variables locales del bloque.
- **Área de instrucciones:** En el área de instrucciones se programa el código del bloque que tiene que ser editado por el sistema de automatización. Este código se compone de uno o varios segmentos. Para crear los segmentos puede utilizar, p.ej., los lenguajes de programación lista de instrucciones (AWL), esquema de contactos (KOP) y diagrama de funciones (FUP).
- **Propiedades de bloques:** Las propiedades de los bloques contienen información adicional, como por ejemplo la indicación de fecha y hora o la ruta, que el mismo sistema registra. También puede indicar el nombre, la familia, la versión y el autor del bloque y asignarle atributos de sistema.

En principio es indistinto en qué orden se editan las distintas partes de un bloque lógico. También se pueden corregir/completar.



---

#### Nota

Si desea recurrir a símbolos de la tabla de símbolos, primero debería verificarlos y, si es necesario, completarlos.

---

### 9.1.2 Preajustes para el editor de programas KOP/FUP/AWL

Antes de comenzar a programar es recomendable que conozca las posibilidades de ajuste para poder trabajar de la forma más cómoda posible y conforme a sus preferencias.

Elija el comando de menú **Herramientas > Preferencias** y se abrirá un cuadro de diálogo compuesto por varias fichas. En las fichas puede efectuar preajustes para programar los bloques, por ejemplo en la ficha "Editor"

- la fuente y el tamaño de letra de los textos y tablas;
- si al crear un nuevo bloque desea visualizar primero los símbolos y comentarios.

Los ajustes relativos al lenguaje, a los comentarios y a los símbolos se pueden modificar durante la edición mediante los comandos del menú **Ver > ...**

El color en el que se representan las selecciones (p.ej. de segmentos o de líneas de instrucciones) se puede cambiar en la ficha "KOP".

### 9.1.3 Derechos de acceso a bloques y fuentes

En la ejecución de un proyecto se utiliza a menudo una gestión de datos común, con lo cual puede suceder que varios usuarios quieran acceder al mismo bloque o a la misma fuente de datos.

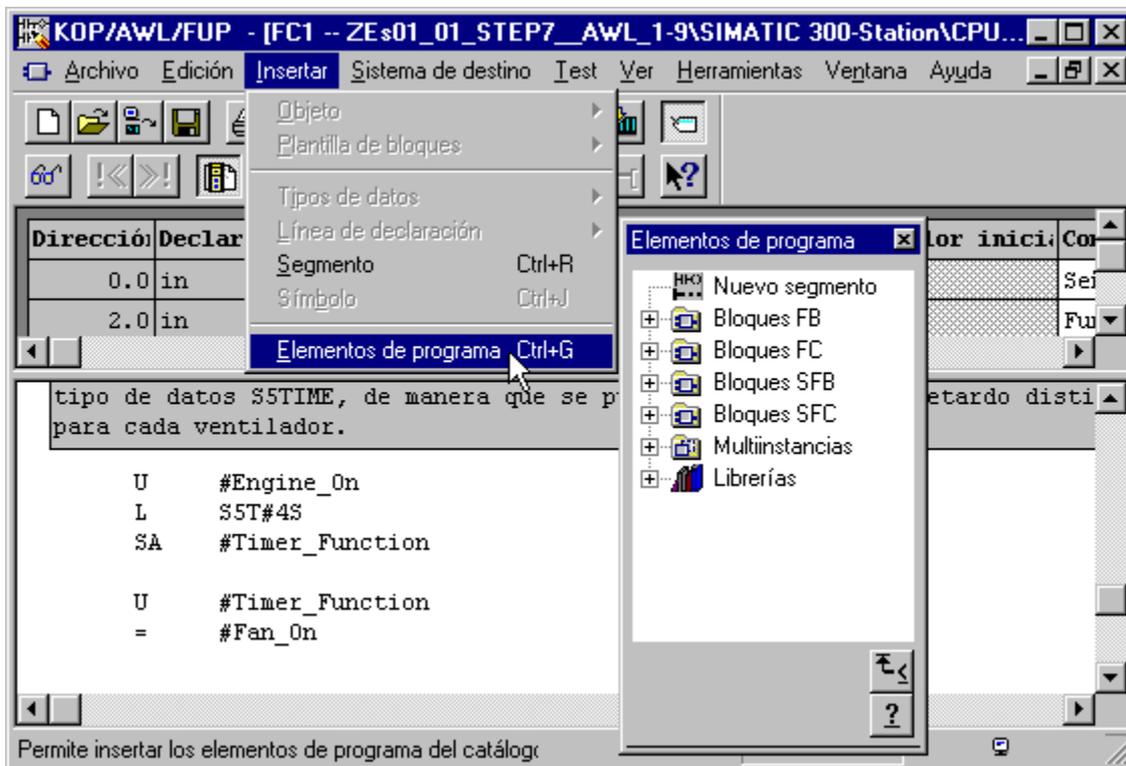
Los derechos de lectura y escritura se adjudican como sigue:

- **Ejecución offline:**  
Al abrir un bloque o una fuente se comprueba si éstos pueden ser abiertos con el derecho 'de escritura'. Si el bloque o fuente ya están abiertos, sólo se puede trabajar con una copia. En el caso de que quiera guardar la copia, se le preguntará si desea sobrescribir el original o guardar la copia con otro nombre.
- **Ejecución online:**  
Si el usuario abre un bloque online a través de un enlace configurado, el bloque offline correspondiente queda bloqueado y con ello se impiden las modificaciones simultáneas en el mismo.

### 9.1.4 Instrucciones del catálogo de elementos de programa

El catálogo "Elementos de programa" comprende elementos de los lenguajes KOP y FUP, así como multiinstancias ya declaradas, bloques ya listos y bloques de librerías. Dicho catálogo se puede llamar con el comando de menú **Ver > Catálogo**. Los elementos de programa se pueden adoptar en el área de instrucciones eligiendo el comando de menú **Insertar > Elementos de programa**.

#### Ejemplo del catálogo de elementos de programa en AWL



## 9.2 Editar la tabla de declaración de variables

### 9.2.1 Utilización de la declaración de variables en los bloques lógicos

Tras abrir un bloque lógico aparece una ventana con la tabla de declaración de variables de dicho bloque en la parte superior, así como el área de instrucciones en la parte inferior, donde se edita la lógica del bloque en sí.

#### Ejemplo: Tabla de declaración de variables y área de instrucciones en AWL

The screenshot shows the STEP 7 software interface. The main window title is 'KOP\AWL\FUP:-FB6-<Offline>'. The menu bar includes 'Archivo', 'Edición', 'Insertar', 'Sistema de destino', 'Test', 'Ver', 'Herramientas', 'Ventana', and 'Ayuda'. Below the menu bar is a toolbar with various icons. The main area is divided into two sections:

The top section is titled 'TRAFICO\...\FB6-<Offline>' and contains a table with the following data:

Dirección	Decl.	Nombre	Tipo	Valor inicial	Comentario
0.0	in	dur_g_p	S5TIME	S5T#0MS	
2.0	in	del_r_p	S5TIME	S5T#0MS	
4.0	in	starter	BOOL	FALSE	
6.0	in	t_dur_y_car	TIMER		
8.0	in	t_dur_y_car	TIMER		
10.0	in	t_delay_y_car	TIMER		

The bottom section is titled 'FB6: Semáforo' and shows the instruction area for 'Segmento 1' with the following code:

```

U(
U #starter
U
O #condition
)
UN #t_dur_r_car
= #condition
Segmento 2 : ???
UN #condition
= #g_car
    
```

En la tabla de declaración de variables se definen las variables locales, incluyendo los parámetros formales del bloque y los atributos de sistema de los parámetros. Las consecuencias son, entre otras, las siguientes:

- Al declarar las variables se reserva suficiente espacio de memoria en la pila de datos locales para las variables temporales y, en el caso de los bloques de función, para las variables estáticas del DB de instancia que se asociará posteriormente.
- Al definir los parámetros de entrada, salida y de entrada/salida se fija también el "interface" que llamará al bloque en el programa.
- Al declarar variables en un bloque de función, dichas variables determinan (con excepción de las variables temporales) también la estructura de todos los DBs de instancia que se asocien al FB.
- Al fijar los atributos de sistema se les asignan a los parámetros determinadas propiedades, por ejemplo para la configuración de mensajes y de enlaces, para funciones de manejo y visualización o para la configuración del control de procesos.

### 9.2.2 Interacción entre la tabla de declaración de variables y el área de instrucciones

La tabla de declaración de variables y el área de instrucciones de los bloques lógicos están estrechamente relacionadas, puesto que en esta última se utilizan los nombres que aparecen en la tabla de declaración de variables. Por tanto, los cambios que se hagan en la declaración de variables tienen efecto en toda el área de instrucciones.

Acción en la declaración de variables	Reacción en el área de instrucciones
Nueva entrada correcta	Si existe un código no válido, una variable que no se haya declarado antes será válida ahora.
Cambio correcto de nombre sin modificación de tipo	El símbolo se representará inmediatamente con su nuevo nombre en todas partes.
Cambio de un nombre correcto por un nombre no válido	El código no se modificará.
Cambio de un nombre no válido por un nombre correcto	Si existe un código no válido, éste se convertirá en uno válido.
Cambio de tipo	Si existe un código no válido, éste se convertirá en uno válido. Si existe un código válido, es posible que se convierta en un código no válido.
Borrado de una variable (símbolo) utilizada en el código.	El código válido se convierte en un código no válido.

Los cambios de los comentarios, la entrada incorrecta de una nueva variable, el cambio de un valor inicial o el borrado de una variable no utilizada no afectan al área de instrucciones.

### 9.2.3 Estructura de la tabla de declaración de variables

La tabla de declaración de variables comprende entradas para la dirección, el tipo de declaración, el nombre, el tipo de datos, el valor inicial y el comentario de la variable. Cada línea de la tabla representa una declaración de una variable. Las variables del tipo ARRAY o STRUCT requieren varias líneas.

Los tipos de datos admisibles para los datos locales de los diversos tipos de bloques se indican en el tema "Asignar tipos de datos a los datos locales de bloques lógicos".

Columna	Significado	Observaciones	Ejecución
Dirección	Dirección en formato BYTE.BIT.	Si se trata de tipos de datos que requieran más de un byte, la dirección muestra la asignación con un salto a la siguiente dirección de byte. Leyenda: * : tamaño de un elemento de campo en bytes. + : dirección inicial en relación con el comienzo de la estructura = : memoria total requerida por una estructura	Entrada de sistema: El sistema adjudica y visualiza la dirección al terminar de declarar una entrada.
Variable	Nombre simbólico de la variable	El nombre debe comenzar con una letra. No está permitido utilizar palabras clave reservadas.	Necesaria

Columna	Significado	Observaciones	Ejecución
Declaración	Tipo de declaración, "Finalidad" de la variable	Dependiendo del tipo de bloque, se permiten: Parámetros de entrada "in" Parámetros de salida "out" Parámetros de entrada/salida "in_out" Variables estáticas "stat" Variables temporales "temp"	Estándar conforme al tipo de bloque
Tipo de datos	Tipo de datos de la variable (BOOL, INT, WORD, ARRAY etc.).	Los tipos de datos simples se pueden elegir del menú emergente al oprimir la tecla derecha del ratón.	Necesaria
Valor inic.	Valor inicial si el software no debe adoptar el valor estándar.	Debe ser compatible con el tipo de datos. Al guardar por primera vez un bloque de datos, el valor inicial se adoptará como valor actual de la variable, a menos que defina expresamente un valor actual.	Opcional
Comentario	Comentario explicativo		Opcional

### Valor estándar

Tras abrir un bloque lógico recién creado se visualiza una tabla de declaración de variables estándar. En ella se indican, en el orden prescrito, únicamente los tipos de declaración admisibles para el bloque seleccionado (in, out, in\_out, stat, temp). Al crear un OB se visualiza una declaración de variables estándar cuyos valores se pueden modificar.

### Columnas no editables en la tabla de declaración

Columna	Introducción
Dirección	El sistema adjudica y visualiza la entrada al terminar de declarar la entrada.
Tipo de declaración	La posición de la declaración en la tabla define el tipo de declaración. Así se garantiza que las variables se introduzcan sólo en el orden correcto de los tipos de declaración. Si desea cambiar un tipo de declaración, corte primero la declaración y péguela luego nuevamente debajo del nuevo tipo de declaración.

## 9.2.4 Observaciones generales sobre las tablas de declaración de variables

Para editar la tabla se dispone de las funciones conocidas del menú **Edición**. Para facilitar la edición se pueden utilizar los comandos del menú contextual que aparece al pulsar la tecla derecha del ratón. Dicho menú le asiste también al introducir el tipo de datos.

### Seleccionar en tablas de declaración de variables

Para seleccionar una línea individual, haga clic en el correspondiente campo de dirección protegido contra escritura. Las líneas adicionales de ese mismo tipo de declaración se seleccionan manteniendo pulsada la tecla SHIFT. Las líneas seleccionadas se visualizan con un fondo negro.

Para seleccionar un ARRAY, haga clic en el campo de dirección de la línea en cuestión.

Para **seleccionar una estructura**, haga clic en el campo de dirección de la primera o de la última línea (en las que aparece la palabra clave STRUCT o END\_STRUCT, respectivamente). Para seleccionar una declaración individual en una estructura, haga clic en el correspondiente campo de dirección de la línea.

Si desea introducir una estructura dentro de otra, la jerarquía se visualiza mediante el correspondiente sangrado del nombre de la variable.

### Deshacer acciones

Con el comando de menú **Edición > Deshacer** se puede deshacer en la tabla de variables la última operación de cortar o borrar que se haya efectuado.

## 9.3 Multiinstancias en la tabla de declaración de variables

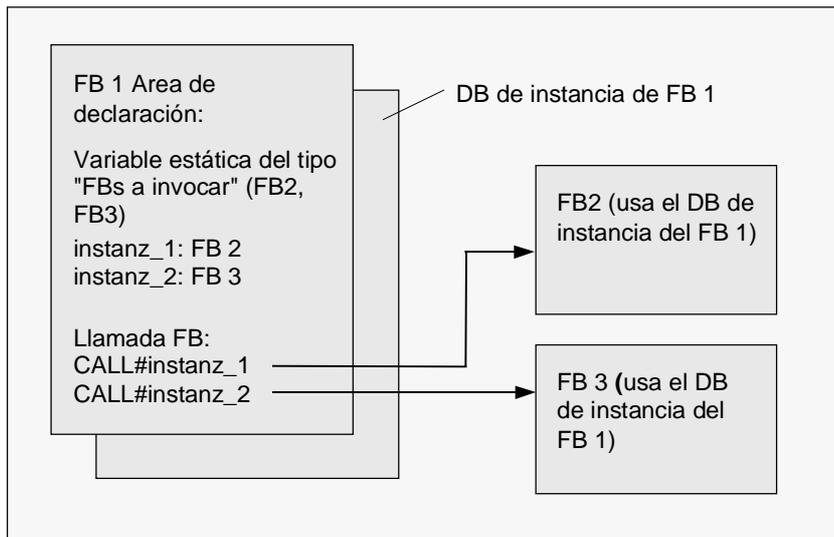
### 9.3.1 Utilización de multiinstancias

Es posible que, debido a los datos característicos (p.ej. capacidad de memoria) de las CPUS S7 utilizadas, sólo pueda o desee utilizar una cantidad limitada de DBs para los datos de instancia. Si en un FB de su programa de usuario se invocan otros bloques de función ya existentes (jerarquía de llamada de FBs), éstos últimos se pueden llamar sin DBs de instancia propios (es decir, adicionales).

Para solucionarlo se puede proceder de la manera siguiente:

- incorpore los FBs a llamar en forma de variables estáticas en la declaración de variables del FB invocante;
- en este último FB se llaman otros bloques de función sin DBs de instancia propios (es decir, adicionales),
- lo que permite concentrar los datos de instancia en un solo bloque de datos de instancia, pudiéndose así aprovechar mejor el número disponible de DBs.

El siguiente ejemplo muestra la solución descrita: El FB 2 y el FB 3 aprovechan el DB de instancia del FB 1 que es el bloque de función invocante.



Única condición: Es preciso "comunicarle" al bloque de función invocante qué instancias se han de llamar, así como el tipo (de FB) de dichas instancias. Las entradas se deben efectuar en el área de declaración del FB invocante. El FB a utilizar debe tener por lo menos una variable o un parámetro del área de datos (es decir, que no sea VAR\_TEMP).

No utilice bloques de datos multiinstancia si espera realizar cambios online con la CPU en marcha. Para evitar que el proceso automatizado se vea perturbado por la carga del programa modificado es necesario utilizar bloques de datos de instancia.

### 9.3.2 Reglas para declarar multiinstancias

Para declarar multiinstancias se aplican las siguientes reglas: se pueden

- Las multiinstancias sólo se pueden declarar en bloques de función creados a partir de la versión 2 de STEP 7 (v. el atributo de bloque en las propiedades del FB).
- Para poder declarar las multiinstancias, el bloque de función se deberá haber creado como FB apto para multiinstancias (preajuste a partir de la versión x.x de STEP 7; deseleccionable en el editor mediante el comando de menú **Herramientas > Preferencias**).
- Al bloque de función en el que se declara una multiinstancia debe asociarse un DB de instancia.
- Una multiinstancia sólo se puede declarar como variable estática (tipo de declaración "stat").

---

#### Nota

- También se pueden crear multiinstancias para bloques de función de sistema.
  - Si el FB creado no es apto para multiinstancias y, sin embargo, Ud. precisa esta característica, puede generar una fuente del FB, borrar de ella la propiedad CODE\_VERSION1 y volver a compilar el FB.
- 

### 9.3.3 Introducir la multiinstancia en la tabla de declaración de variables

1. Abra el FB desde el que se deben invocar los FBs subordinados.
2. Al declarar las variables del FB invocante, defina una variable estática para cada llamada de un bloque de función, si no desea indicar el bloque de datos de la instancia correspondiente.
  - Situe el cursor en una línea en blanco con la declaración "stat" en la segunda columna.
  - Detrás del tipo de declaración "stat", introduzca en la columna "Nombre" una denominación para la llamada del FB.
  - En la columna "Tipo", introduzca la dirección absoluta del FB a llamar, o bien su nombre simbólico.
  - En la columna "Comentario" se pueden introducir explicaciones opcionales.

#### *Llamadas en el área de instrucciones*

En caso de declarar multiinstancias, las llamadas de FBs se pueden utilizar sin indicar el DB de instancia.

Ejemplo: Si se ha definido la variable estática "Nombre: Motor\_1 , tipo de datos: FB20", la instancia se puede llamar de la siguiente manera:

Call Motor\_1 // Llamada del FB 20 sin DB de instancia

## 9.4 Indicaciones generales para introducir instrucciones y comentarios

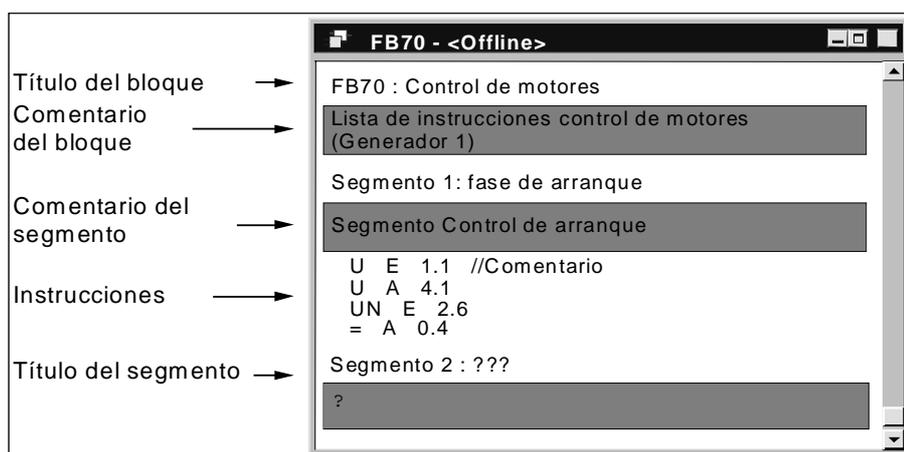
### 9.4.1 Estructura del área de instrucciones

En el área de instrucciones se programa la lógica del bloque. Dependiendo del lenguaje de programación elegido, deberá introducir instrucciones en segmentos. Tras introducirse una instrucción, el editor realiza inmediatamente una comprobación de sintaxis, indicando los posibles errores en letra cursiva roja.

En la mayoría de los casos, el área de instrucciones de un bloque lógico se compone de varios segmentos que, a su vez, comprenden una serie de instrucciones.

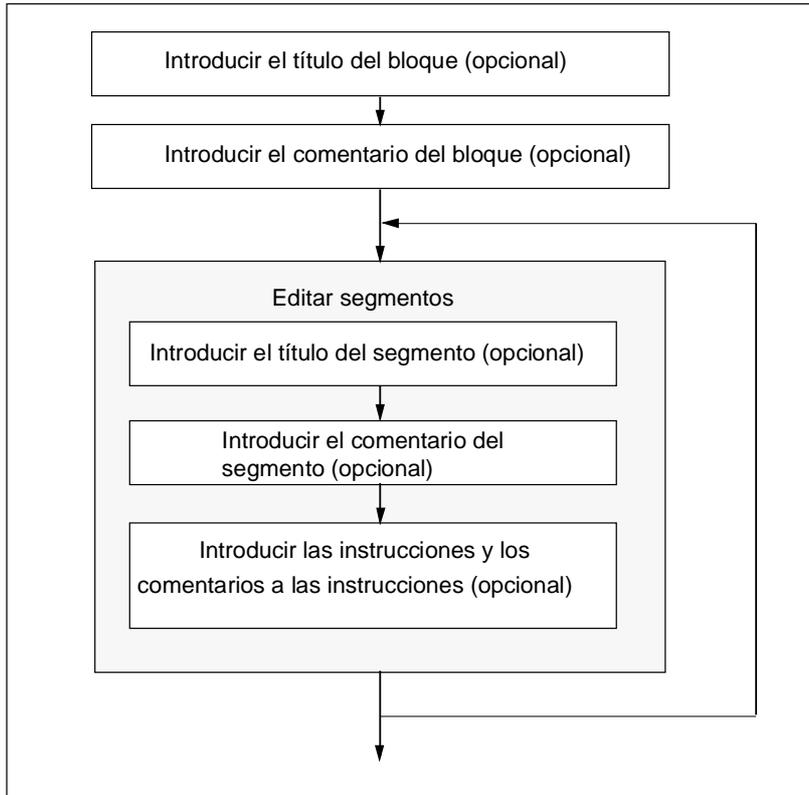
En el área de instrucciones se pueden editar los títulos y los comentarios de los bloques y de los segmentos, así como las líneas de instrucciones en los segmentos.

### Estructura del área de instrucciones tomando como ejemplo el lenguaje de programación AWL



### 9.4.2 Procedimiento para introducir instrucciones

Por regla general, los componentes del área de instrucciones se pueden editar en un orden cualquiera. Al programar un bloque por primera vez se recomienda el siguiente procedimiento:



Además del modo de inserción, los cambios se pueden efectuar también en el modo de sobreescritura. Utilice la tecla **INS** para conmutar entre ambos modos.

### 9.4.3 Introducir símbolos globales en un programa

Utilice el comando de menú **Insertar > Símbolo** para insertar símbolos en el área de instrucciones de su programa. Si el cursor se encuentra al comienzo, al final o dentro de una cadena de caracteres, ya estará seleccionado el símbolo con el que comienza la cadena (si el símbolo ya existe). Si modifica la cadena de caracteres, el símbolo seleccionado en la lista se corregirá automáticamente.

Los caracteres separadores que indican el comienzo y el final de una cadena son p.ej. un carácter en blanco, un punto o dos puntos. Los caracteres separadores no se interpretan dentro de los símbolos globales.

Para insertar símbolos puede proceder de la siguiente forma:

1. Introduzca en el programa la letra inicial del símbolo deseado.
2. Pulse simultáneamente las teclas CTRL y J para visualizar la lista de símbolos. Aparecerá seleccionado el primer símbolo que tenga la letra inicial indicada.
3. Adopte el símbolo pulsando la tecla INTRO o elija un símbolo diferente.

El símbolo entre comillas quedará insertado, sustituyendo la letra inicial

Regla general: Si el cursor se encuentra al comienzo, al final o dentro de una cadena de caracteres, dicha cadena se reemplazará con el símbolo entre comillas cuando éste sea insertado.

### 9.4.4 Títulos y comentarios de bloques y segmentos

Los comentarios facilitan la lectura del programa de usuario, aumentando así la eficacia del funcionamiento y la búsqueda de errores. Los comentarios son una parte importante de la documentación del programa y deberían utilizarse siempre.

#### Comentarios para programas KOP, FUP y AWL:

Se dispone de los siguientes comentarios:

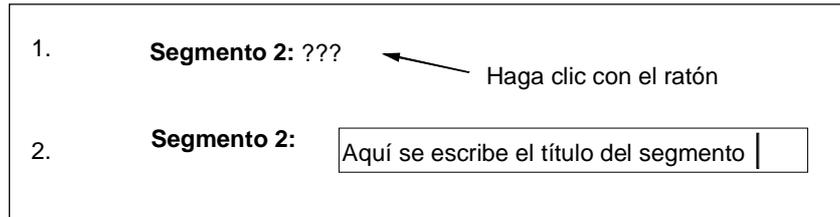
- Título del bloque: título de un bloque (máx. 64 caracteres).
- Comentario del bloque: documenta todo el bloque lógico, como p.ej., la función del bloque.
- Título del segmento: título de un segmento (máx. 64 caracteres).
- Comentario de segmento: documenta la función de los diferentes segmentos.
- Columna de comentario de la tabla de declaración de variables: comentarios acerca de los datos locales declarados.
- Comentario del símbolo: comentarios que se introdujeron en la tabla de símbolos para un operando en la definición del nombre.  
Puede visualizar dichos comentarios con el comando de menú **Ver > Mostrar > Información del símbolo**.

En el área de instrucciones del bloque lógico se pueden introducir los títulos y los comentarios de los bloques y de los segmentos.

## Títulos de bloques o de segmentos

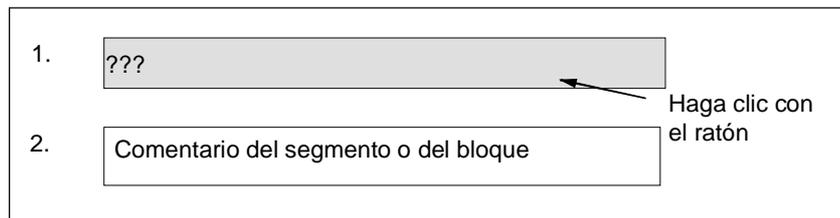
Para introducir títulos de bloques o de segmentos, sitúe el cursor en los tres signos de interrogación que aparecen a la derecha del nombre del bloque o del segmento, p.ej. Segmento 1 : ???). Se abrirá un campo de texto para introducir el título. Éste puede comprender 64 caracteres como máximo.

Los comentarios de bloques se refieren al bloque lógico entero. En ellos se puede describir la función de éste último. Los comentarios de segmento se refieren a los diferentes segmentos y describen sus particularidades.



## Comentarios de bloques y de segmentos

Los campos de comentario grises se pueden mostrar u ocultar eligiendo el comando de menú **Ver > Mostrar > Comentario**. Haciendo doble clic en uno de dichos campos se abre el cuadro de texto donde se puede introducir el comentario. Por cada bloque dispone de 64 KB para comentarios de bloques y de segmentos.



### 9.4.5 Función de búsqueda de errores en el área de instrucciones

Los errores se reconocen fácilmente en el área de instrucciones por estar destacados en rojo. Para desplazarse más fácilmente a los errores que se encuentren fuera del área visible, el editor incorpora las dos funciones de búsqueda **Edición > Ir a > Error precedente/error siguiente**.

La búsqueda de los errores se efectúa en todos los segmentos. Por tanto, se busca en toda el área de instrucciones y no sólo en un solo segmento o en el área visible en ese momento.

Activando la barra de estado con el comando de menú **Ver > Barra de estado** se visualizan allí observaciones relativas a los errores.

Las correcciones de errores y los cambios también se pueden efectuar en el modo de sobreescritura. Utilice la tecla INS para conmutar del modo de inserción al modo de sobreescritura y viceversa.

## 9.5 Editar instrucciones KOP en el área de instrucciones

### 9.5.1 Ajustes para el lenguaje de programación KOP

#### Ajustar el diseño de página de KOP

Para crear el programa en lenguaje KOP, puede definir el diseño de página. El diseño seleccionado (formato vertical DIN A4/formato horizontal/tamaño máximo) determinará el número de elementos KOP representables en la rama.

1. Elija el comando de menú **Herramientas > Preferencias**.
2. En el cuadro de diálogo que aparece a continuación, elija la ficha "KOP/FUP".
3. Elija el formato deseado en el cuadro de lista "Diseño de página". Introduzca el formato deseado.

#### Ajustes para imprimir:

Si desea imprimir el área de instrucciones KOP, deberá ajustar el diseño de página adecuado antes de crear dicha área.

#### Ajustes en la ficha "KOP/FUP"

En la ficha "KOP/FUP", a la que se accede mediante el comando de menú **Herramientas > Preferencias**, se pueden efectuar ajustes básicos, p.ej. el diseño de página y el ancho del campo de operandos.

### 9.5.2 Reglas para introducir operaciones KOP

El lenguaje de programación "KOP" se describe en el manual "KOP para S7-300/400 – Programación de bloques" o en la Ayuda en pantalla de KOP.

Un segmento KOP puede comprender varios elementos en diversas ramas. Todos los elementos y ramas deben estar unidos entre sí, sin que la barra de alimentación izquierda se considere una unión (IEC 1131–3).

Al programar en KOP es preciso tener en cuenta ciertas reglas. En caso de detectarse un error se visualizará el correspondiente mensaje.

## Finalizar un segmento KOP

Cada segmento KOP debe terminar con una bobina o con un cuadro. No obstante, los siguientes elementos KOP no se pueden utilizar para finalizar un segmento:

- Cuadros de comparación
- Bobinas para conectores  $\_/(#)\_ /$
- Bobinas para la evaluación positiva  $\_/(P)\_ /$  o negativa  $\_/(N)\_ /$  de flancos

## Emplazamiento de los cuadros

El punto inicial de la rama para la conexión de un cuadro debe ser siempre la barra de alimentación izquierda. No obstante, en la rama que precede al cuadro pueden encontrarse operaciones lógicas u otros cuadros.

## Emplazamiento de las bobinas

Las bobinas se emplazan automáticamente en el borde derecho del segmento, constituyendo allí el final de una rama.

Excepciones: Las bobinas para conectores  $\_/(#)\_ /$ , así como la evaluación positiva  $\_/(P)\_ /$  o negativa  $\_/(N)\_ /$  de flancos no se pueden emplazar ni en el extremo izquierdo ni en el derecho. Tampoco son admisibles en ramas paralelas.

Algunas de las bobinas exigen una operación booleana, en tanto que otras no permiten dicha operación.

- Bobinas que exigen una operación booleana:
  - Salida  $\_/( )$ , Activar  $\_/(S)$ , Desactivar  $\_/(R)$
  - Conector  $\_/(#)\_ /$ , Flanco positivo  $\_/(P)\_ /$ , Flanco negativo  $\_/(N)\_ /$
  - Todas las bobinas de contadores y temporizadores
  - Saltar si es 0  $\_/(JMPN)$
  - Conectar Master Control Relay  $\_/(MCR<)$
  - Cargar RLO en registro RB  $\_/(SAVE)$
  - Retorno  $\_/(RET)$
- Bobinas que no permiten una operación booleana:
  - Inicio Master Control Relay  $\_/(MCRA)$
  - Fin Master Control Relay  $\_/(MCRD)$
  - Abrir bloque de datos  $\_/(OPN)$
  - Desconectar Master Control Relay  $\_/(MCR>)$

Todas las demás bobinas pueden tener o no una operación booleana.

Las siguientes bobinas no se pueden utilizar **en calidad de salida paralela**:

- Saltar si es 0  $\_/(JMPN)$
- Saltar si es 1  $\_/(JMP)$
- Llamada de bloque  $\_/(CALL)$
- Retorno  $\_/(RET)$

### Entrada/salida de habilitación

La entrada de habilitación "EN" y/o la salida de habilitación "ENO" de los cuadros se puede conectar, sin que ello sea absolutamente necesario.

### Retirar y cambiar

Si una rama se compone de un solo elemento, al retirarse éste se retirará toda la rama.

Al retirarse un cuadro, se retirarán también todas las ramas unidas al mismo con entradas booleanas, con excepción de la rama principal.

Para la sustitución simple de elementos de un mismo tipo se puede utilizar el modo de sobreescritura.

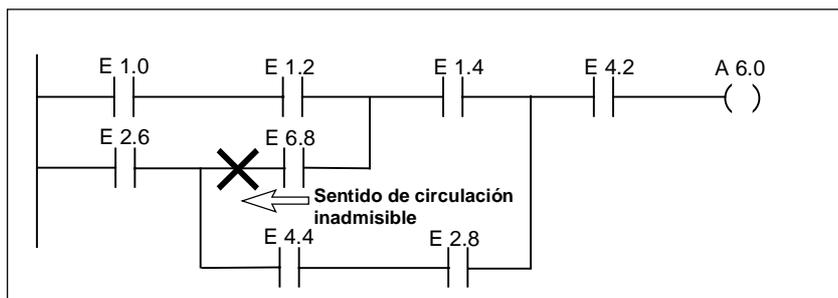
### Ramas paralelas

- Las ramas O se añaden de izquierda a derecha.
- Las ramas paralelas se abren hacia abajo y se cierran hacia arriba.
- Las ramas paralelas se abren siempre detrás del elemento KOP seleccionado.
- Las ramas paralelas se cierran siempre detrás del elemento KOP seleccionado.
- Para borrar una rama paralela deberá borrar primero todos los elementos KOP de la rama. Al borrar el último elemento KOP, se borrará también el resto de la misma.

### 9.5.3 Interconexiones no admisibles en KOP

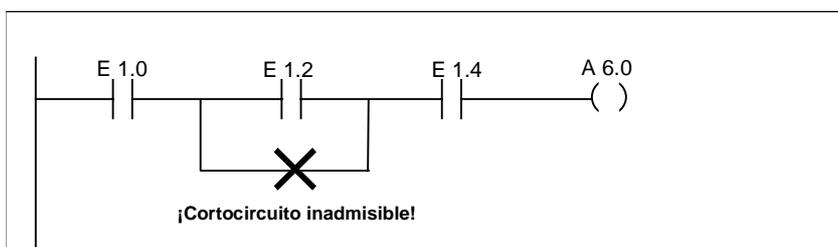
#### Circulación de corriente de derecha a izquierda

No se pueden editar ramas que pudieran hacer que la corriente fluya en sentido contrario. La figura siguiente muestra un ejemplo al respecto. Un estado de señal "0" en E 1.4 causaría un flujo de corriente de derecha a izquierda en E 6.8, lo cual no es admisible.



#### Cortocircuito

No se pueden editar ramas que causen un cortocircuito. La figura siguiente muestra un ejemplo al respecto:



## 9.6 Editar instrucciones FUP en el área de instrucciones

### 9.6.1 Ajustes para el lenguaje de programación FUP

#### Ajustar el diseño de página de FUP

Para crear el programa en lenguaje FUP se puede definir el diseño de página. El diseño seleccionado (formato vertical DIN A4/formato horizontal/tamaño máximo) repercute en el número de elementos FUP representables en una rama.

1. Elija el comando de menú **Herramientas > Preferencias**.
2. En el cuadro de diálogo que aparece a continuación, elija la ficha "KOP/FUP".
3. Elija el formato deseado en el cuadro de lista "Diseño de página". Introduzca el formato deseado.

#### Notas respecto a la impresión

Si desea imprimir el área de instrucciones FUP, deberá ajustar el diseño de página adecuado antes de crear dicha área.

#### Ficha KOP/FUP en Herramientas > Preferencias

En la ficha "KOP/FUP", a la que se accede mediante el comando de menú **Herramientas > Preferencias**, se pueden efectuar ajustes básicos, por ejemplo el diseño de página y el ancho del campo de operandos.

### 9.6.2 Reglas para introducir operaciones FUP

El lenguaje de programación "FUP" se describe en el manual "FUP para S7-300/400 – Programación de bloques" o en la Ayuda en pantalla de FUP.

Un segmento FUP puede comprender varios elementos. Todos los elementos deben estar interconectados (IEC 1131–3).

Al programar en FUP es preciso tener en cuenta ciertas reglas. En caso de detectarse un error se visualizará el correspondiente mensaje.

#### Insertar y editar direcciones y parámetros

Al insertar un elemento FUP se introducen las secuencias de caracteres "???" y "..." como comodines para direcciones o parámetros.

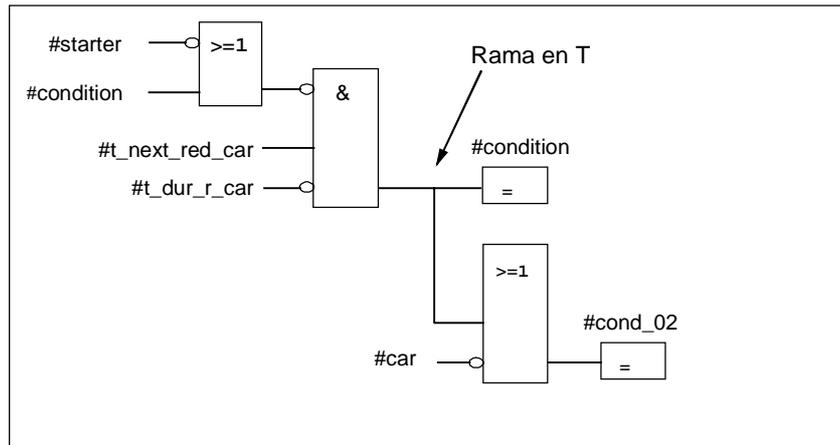
- La cadena de caracteres representada en rojo "???" indica direcciones y parámetros que se deben asignar.
- La cadena de caracteres representada en negro "..." indica direcciones y parámetros que se pueden asignar.

Si mueve el puntero del ratón en el comodín, se mostrará el tipo de datos deseado.

## Emplazamiento de los cuadros

A los cuadros con operaciones binarias (&, >=1, XOR) se pueden añadir cuadros estándar (flipflops, contadores, temporizadores, operaciones aritméticas, etc.). Esta regla excluye los cuadros de comparación.

En un segmento no se pueden programar diferentes operaciones lógicas con salidas por separado. No obstante, mediante una rama T de una cadena de operaciones lógicas se pueden asociar varias asignaciones. La figura siguiente muestra un segmento con dos asignaciones.



Los cuadros siguientes sólo se pueden emplazar en el borde derecho de la cadena de operaciones lógicas, constituyendo allí el final de la misma:

- Poner contador al valor inicial
- Incrementar contador, decrementar contador
- Arrancar temporizador como impulso, arrancar temporizador como impulso prolongado
- Arrancar temporizador como retardo a la conexión/desconexión.

Algunos de los cuadros exigen una operación booleana, en tanto que otros no permiten dicha operación.

### Cuadros que exigen una operación booleana:

- Salida, activar, desactivar \_/[R]
- Conector \_/[#]\_, flanco positivo \_/[P]\_, flanco negativo \_/[N]\_
- Todos los cuadros de contadores y temporizadores
- Saltar si es 0 \_/[JMPN]
- Conectar Master Control Relay \_/[MCR<]
- Cargar RLO en registro RB \_/[SAVE]
- Retorno \_/[RET]

### **Cuadros que no permiten una operación booleana:**

- Inicio Master Control Relay [MCRA]
- Fin Master Control Relay [MCRD]
- Abrir bloque de datos [OPN]
- Desconectar Master Control Relay [MCR>]

Todos los demás cuadros pueden tener o no una operación booleana.

### **Entrada/salida de habilitación**

La entrada de habilitación "EN" y/o la salida de habilitación "ENO" de los cuadros se puede conectar, sin que ello sea absolutamente necesario.

### **Retirar y cambiar**

Al retirarse un cuadro, se retirarán también todas las ramas unidas al mismo con entradas booleanas, con excepción de la rama principal.

Para la sustitución simple de elementos de un mismo tipo se puede utilizar el modo de sobreescritura.

## 9.7 Editar instrucciones AWL en el área de instrucciones

### 9.7.1 Ajustes para el lenguaje de programación AWL

#### Ajustar la nemotécnica

Se puede elegir entre dos ajustes:

- Alemán o
- Inglés

La nemotécnica no se puede cambiar mientras se está editando el bloque, por lo que antes de abrir un bloque en el Administrador SIMATIC, puede ajustarla con el comando de menú **Herramientas > Preferencias** en la ficha "Lenguaje".

Las **propiedades de bloques** se editan en un cuadro de diálogo por separado.

En el editor es posible tener abiertos simultáneamente varios bloques que se pueden editar uno tras otro a voluntad.

### 9.7.2 Reglas para introducir instrucciones AWL

El lenguaje de programación "AWL" se describe en el manual "AWL para S7-300/400 – Programación de bloques" o en la Ayuda en pantalla de AWL (Descripción de lenguajes ).

Para utilizar el método de entrada incremental de instrucciones AWL deberá respetar las siguientes reglas básicas:

- Aténgase al orden de programación de los bloques. Los bloques llamados deben programarse antes de los bloques invocantes.
- Toda instrucción se compone de una marca de salto (opcional), una operación, un operando y un comentario (opcional).  
Ejemplo: M001: U E1.0 //Comentario
- Cada instrucción ocupa una línea propia.
- Por cada bloque se puede introducir un máximo de 999 segmentos.
- Por cada segmento se pueden introducir unas 2000 líneas. Si utiliza el zoom (ampliar, reducir), aumentará o disminuirá el número posible de líneas.
- Al introducir operaciones o direcciones absolutas no se distingue entre mayúsculas y minúsculas.

## 9.8 Actualizar llamadas a bloques

### 9.8.1 Actualizar llamadas a bloques

Con el comando de menú **Edición > Llamada > Actualizar** en "KOP/AWL/FUP: Programar bloques" se pueden actualizar automáticamente las llamadas de bloques o los UDTs que ya no sean válidos, después de cambiar los siguientes interfaces:

- Insertar nuevos parámetros
- Borrar parámetros
- Cambiar el nombre de parámetros
- Cambiar el tipo de parámetros
- Cambiar el orden de parámetros (copiándolos a otro lugar)

Al asignar el lado formal y el actual se procede conforme a las siguientes reglas en el orden indicado:

1. **Nombres idénticos de parámetros:**  
Los parámetros actuales se asignan automáticamente si el nombre del parámetro formal no ha cambiado.  
Excepción: La precombinación lógica de parámetros de entrada binarios sólo se puede asignar automáticamente en KOP y FUP si no cambia el tipo de datos (BOOL). Si el tipo de datos ha cambiado, la precombinación lógica previa se conservará en calidad de rama abierta.
2. **Tipos de parámetros idénticos:**  
Después de asignar los parámetros de igual nombre, los parámetros actuales aún no asignados se asignarán a los parámetros formales que tengan el mismo tipo de datos que los "antiguos" parámetros formales.
3. **Posición idéntica de los parámetros:**  
Los parámetros actuales aún no asignados conforme a las reglas 1 y 2, se asignarán a los nuevos parámetros formales conforme a su posición en el "antiguo" interface.
4. Si los parámetros actuales no se pueden asignar conforme a las tres reglas mencionadas, se borrarán o se conservarán en calidad de ramas abiertas al tratarse de precombinaciones lógicas binarias en KOP o FUP.

Tras ejecutar esta función, los cambios efectuados se deberán comprobar en la tabla de declaración de variables y en el área de instrucciones del programa.

## 9.9 Guardar bloques lógicos

### 9.9.1 Guardar bloques lógicos

Para adoptar bloques nuevos o bien los cambios realizados en el área de instrucciones de bloques lógicos o en tablas de declaración de la gestión de datos de la PG, deberá guardar el bloque correspondiente, escribiendo los datos en el disco duro de la PG.

#### Para guardar bloques en el disco duro de la PG:

1. Active la ventana de trabajo del bloque que desea guardar.
2. Elija:
  - el comando de menú **Archivo > Guardar** si desea guardar el bloque con el mismo nombre
  - el comando de menú **Archivo > Guardar** como si desea guardar el bloque en otro programa de usuario S7, o bien con otro nombre. Introduzca la nueva ruta o el nuevo nombre del bloque en el cuadro de diálogo que aparece a continuación.

En ambos casos, el bloque se guarda solamente si la sintaxis es correcta. Los errores sintácticos se detectan inmediatamente al crear el bloque y se destacan en rojo. Dichos errores deben corregirse antes de poder guardar el bloque.

---

#### Nota

- También puede guardar los bloques o las fuentes en otros proyectos o librerías utilizando también el Administrador SIMATIC (p.ej., con arrastrar y soltar).
  - Los bloques o los programas de usuario completos sólo se pueden guardar en una Memory Card con ayuda del Administrador SIMATIC.
  - Si surgen problemas al guardar o compilar bloques, tendrá que reorganizar el proyecto. Para ello elija en el Administrador SIMATIC el comando de menú **Archivo > Reorganizar**. Después vuelva a guardar o a compilar el bloque.
- 

### 9.9.2 Corregir los interfaces en una FC, un FB o un UDT

Cuando sea necesario corregir el interface de un FB, una FC o un UDT, proceda de la siguiente forma para evitar conflictos de fecha y hora:

1. Genere una fuente AWL con el bloque a modificar y con todos los bloques referenciados directa o indirectamente.
2. Guarde los cambios en la fuente creada.
3. Compile nuevamente en los bloques la fuente adaptada.

Ahora puede guardar o cambiar el cambio del interface.

### 9.9.3 Evitar errores al llamar bloques

#### STEP 7 sobrescribe los datos en el registro DB.

STEP 7 modifica los registros de las CPUs S7-300/S7-400 en diferentes operaciones. Por ejemplo, el contenido de los registros DB y DI se intercambia al efectuar la llamada de un FB. De esta manera es posible abrir el DB de instancia del FB llamado sin perder la dirección del DB de instancia anterior.

Si se trabaja con direccionamiento absoluto, pueden producirse errores al acceder a los datos memorizados en los registros: en algunos casos se podrían sobrescribir las direcciones contenidas en los registros AR1 (registro de direcciones 1) y en el registro DB, lo que podría provocar el que se leyera o escribiera en direcciones erróneas.



#### Peligro

Peligro de daños materiales o personales al utilizar:

1. CALL FC, CALL FB, CALL multinstancia
2. Accesos a DBs con la dirección completa (p.ej., DB20.DBW10)
3. Acceso a variables de un tipo de datos compuestos

Podría cambiar el contenido del registro DB (DB y DI), del registro de direcciones (AR1, AR2) y de los acumuladores (ACU1, AC2).

Al efectuar las llamadas CALL FB/FC tampoco se puede utilizar el resultado lógico (RLO) como parámetro adicional (implícito).

Si se utilizan las posibilidades de programación anteriores, el usuario deberá tomar las precauciones necesarias para restablecer los contenidos y así evitar fallos.

#### Memorizar datos correctos

Para la memorización puede resultar problemático utilizar el contenido del registro DB si se accede a los datos utilizando el formato abreviado de las direcciones absolutas. Por ejemplo, si se presupone que el DB20 está abierto (y su número está memorizado en el registro del DB), entonces se indicará DBX0.2 para acceder a los datos memorizados en el bit 2 del byte 0 del DB, cuya dirección esté depositada en el registro DB (o sea, DB20). Si el registro DB contiene, no obstante, otro número de DB, entonces se accederá erróneamente a otros datos.

Para evitar errores que se producen al acceder a datos del registro DB direcciona los datos:

- utilizando direccionamiento simbólico, o bien
- utilizando la dirección absoluta completa (p. ej., DB20.DBX0.2)

Con cualquiera de estos métodos de direccionamiento, STEP 7 abrirá automáticamente el DB correcto. Si el registro AR1 se utiliza para direccionamiento indirecto, se ha de cargar siempre la dirección directa en AR1.

## Situaciones en las que se modifican registros

La manipulación de los registros de direcciones para fines de direccionamiento indirecto sólo es importante en AWL. En efecto, los otros lenguajes de programación no ofrecen prestaciones de acceso indirecto a los registros de direcciones.

En todos los lenguajes de programación es necesario que el compilador adapte el registro DB a fin de garantizar una transferencia correcta de parámetros al efectuar cualquier llamada a un bloque.

El contenido del registro de direcciones AR1 y del registro DB del bloque invocante se sobrescribe cuando se presentan las situaciones siguientes:

Situación	Explicación
En parámetros actuales procedentes de un DB	<ul style="list-style-type: none"> <li>Una vez que se ha asignado a un bloque un parámetro actual almacenado en un DB (p. ej., DB20.DBX0.2), STEP 7 abre dicho DB (DB20) y adapta el contenido del registro DB. Tras la llamada al bloque, el programa trabaja con el DB adaptado.</li> </ul>
Al llamar a bloques si se utilizan tipos de datos compuestos	<ul style="list-style-type: none"> <li>Tras la llamada a un bloque desde una FC que transfiere a dicho bloque invocante un componente de un parámetro formal con formato de datos compuesto (String, Array, Struct o UDT) se modifica el contenido de AR1 y del registro DB del bloque invocante.</li> <li>Esto mismo es aplicable a la llamada desde un FB siempre que el parámetro esté situado dentro del área VAR_IN_OUT del bloque invocante.</li> </ul>
Al acceder a componentes de tipo de datos compuesto	<ul style="list-style-type: none"> <li>Cuando un FB accede a un componente de un parámetro formal que tiene como formato un tipo de datos compuesto en el área VAR_IN_OUT (String, Array, Struct o UDT), STEP7 utiliza el registro de direcciones AR1 y el registro DB. Como consecuencia se modifican los contenidos de ambos registros.</li> <li>Cuando un FC accede a un componente de un parámetro formal que tiene como formato un tipo de datos compuesto (String, Array, Struct o UDT), STEP7 utiliza el registro de direcciones AR1 y el registro DB. Como consecuencia se modifican los contenidos de ambos registros.</li> </ul>

### Nota

- Cuando se llama un FB desde un bloque de la versión 1 no se transfiere correctamente el parámetro actual del primer parámetro booleano IN o IN\_OUT cuando la instrucción previa a la llamada no limita el RLO. En este caso se combina con el RLO pendiente.
- Al llamar a un FB (normal o multiinstancia) se sobrescribe el registro de direcciones AR2.
- Si dentro de un FB se modifica el registro de direcciones AR2, no se garantiza una ejecución correcta del FB.
- Si a un parámetro ANY no se le transfiere la dirección absoluta completa del DB, el puntero ANY no contendrá el número del DB abierto sino un 0.

# 10 Crear bloques de datos

## 10.1 Nociones básicas para crear bloques de datos

En el bloque de datos se depositan valores a los que deba acceder su máquina o instalación. Contrariamente al bloque lógico, programado con uno de los lenguajes de programación (KOP, FUP o AWL), el bloque de datos sólo contiene la parte de declaración de variables. No requiere un área de instrucciones y, por tanto, tampoco la programación de segmentos.

Tras abrir un bloque de datos, éste se puede visualizar en la vista "Declaración" o "Datos". Para conmutar de una vista a otra, elija los comandos de menú **Ver > Declaración** y **Ver > Datos**.

### Vista "Declaración"

Elija la vista "Declaración" para

- leer o fijar la estructura de los DBs globales,
- leer la estructura de los DBs con tipo de datos de usuario (UDT) asociado,
- leer la estructura de los DBs con bloque de función (FB) asociado.

La estructura de los bloques de datos asociados a un FB o a un UDT no se puede cambiar. A tal efecto sería preciso modificar antes el FB o el UDT en cuestión y crear luego nuevamente el DB.

### Vista "Datos"

Elija la vista "Datos" para modificar los datos. Solamente en la vista "Datos" es posible visualizar, introducir o cambiar el valor actual de cada uno de los elementos. Si se trata de bloques de datos cuyas variables son tipos de datos compuestos, los elementos se listarán individualmente con su nombre completo.

### Diferencia entre un bloque de datos de instancia y un bloque de datos global

El bloque de datos global no se encuentra asociado a ningún bloque lógico. Contiene valores requeridos por la instalación o por la máquina y se puede llamar directamente desde cualquier lugar del programa.

El bloque de datos de instancia se encuentra asociado directamente a un bloque lógico, p.ej. a un bloque de función. El bloque de datos de instancia contiene los datos depositados en la tabla de declaración de variables de un bloque de función.

## 10.2 Ver declaración de bloques de datos

En los bloques de datos no globales no se puede modificar la vista "Declaración".

Columna	Explicación
Dirección	Indica la dirección que STEP 7 asigna automáticamente a la variable al terminar de introducir una declaración.
Declaración	Esta columna se representa sólo para los bloques de datos de instancia. Ella le informa de cómo se han definido las variables en la declaración de variables del FB: <ul style="list-style-type: none"> <li>• parámetros de entrada (IN)</li> <li>• parámetros de salida (OUT)</li> <li>• parámetros de entrada/salida (IN_OUT)</li> <li>• datos estáticos (STAT)</li> </ul>
Designación	Introduzca el nombre que debe asignar a cada variable.
Tipo	Introduzca el tipo de datos de la variable (BOOL, INT, WORD, ARRAY, etc.). Las variables pueden pertenecer a los tipos de datos simples, a los tipos de datos compuestos, o bien, a los tipos de datos de usuario.
Valor inic.	Indique el valor inicial, en caso de que el software no deba tomar el valor predeterminado del tipo de datos introducido. Todos los valores introducidos deben ser compatibles con los tipos de datos.  Cuando guarde por primera vez el bloque de datos, el valor inicial será adoptado como valor actual de la variable, a menos que defina expresamente su valor actual.
Comentario	En este campo puede introducir un comentario para documentar la variable. El comentario no debe tener más de 80 caracteres.

### 10.3 Ver datos de los bloques de datos

En la vista "Datos" se representan los valores actuales de todas las variables del bloque de datos. Estos valores sólo pueden ser modificados en la vista "Datos". La forma de representar tablas con dicha vista es igual para todos los bloques de datos globales. En el caso de los bloques de datos de instancia, se visualiza adicionalmente la columna "Declaración".

En el caso de las variables con tipos de datos compuestos o con tipos de datos de usuario, cada uno de los elementos se representa en la vista "Datos" en una línea propia con nombre completo. Si los elementos se encuentran en el área IN\_OUT de un bloque de datos de instancia, en la columna "Valor actual" aparecerá el puntero en el tipo de datos compuesto o de usuario.

La vista "Datos" muestra las siguientes columnas:

Columna	Explicación
Dirección	Indica la dirección que STEP 7 asigna automáticamente a la variable.
Declaración	Esta columna se representa sólo para los DBs de instancia. Ella le informa de cómo se han definido las variables en la declaración de variables del FB: <ul style="list-style-type: none"> <li>• parámetros de entrada (IN)</li> <li>• parámetros de salida (OUT)</li> <li>• parámetros de entrada/salida (IN_OUT)</li> <li>• datos estáticos (STAT)</li> </ul>
Designación	Se trata del nombre que se ha definido para la variable. Este campo no se puede editar estando en la vista "Datos".
Tipo	Se trata del tipo de datos definido para la variable. Puesto que, en la vista "Datos" de las variables con tipos de datos compuestos o de usuario, los elementos se listan individualmente, se indican en el caso de un bloque de datos global sólo los tipos de datos simples. En el caso de un bloque de datos de instancia también se indicarán los tipos de datos Parámetro; en el caso de parámetros de entrada/salida (INOUT) con tipo de datos compuesto o de usuario el puntero aparecerá - en la columna "Valor actual" - el puntero en el tipo de datos.
Valor inic.	Se trata del valor inicial que usted ha definido para la variable, en caso de que el software no deba tomar el valor predefinido para el tipo de datos indicado. Cuando guarde por primera vez el bloque de datos, el valor inicial se adoptará como valor actual de la variable, a menos que defina expresamente su valor actual.
Valor actual	Offline: Se trata del valor que tenía la variable al abrir el bloque de datos o después del último cambio que haya sido almacenado (aun cuando haya abierto el DB online, este valor no se actualiza). Online: El valor actual al abrir el bloque de datos se muestra, pero no se actualiza. Para actualizar la visualización debe pulsar la tecla F5. Este campo se puede editar, si no pertenece a un parámetro de entrada/salida (INOUT) con tipo de datos compuesto o de usuario. Todos los valores introducidos deben ser compatibles con los tipos de datos.
Comentario	Se trata del comentario introducido al documentar la variable. Este campo no se puede editar estando en la vista "Datos".

## 10.4 Editar bloques de datos y guardar

### 10.4.1 Introducir la estructura de los bloques de datos globales

Si ha abierto un bloque de datos que no esté asociado a ningún UDT o FB, puede definir su estructura en la vista Declaración del bloque de datos. En los bloques de datos no globales no se puede modificar la vista "Declaración".

1. Abra un bloque de datos global, es decir, un bloque no asociado a ningún UDT o FB.
2. Active la vista "Declaración" del bloque de datos en caso de que ésta no se visualice todavía.
3. Defina la estructura rellenando la tabla visualizada conforme a los datos que se indican a continuación.

En los bloques de datos no globales no se puede modificar la vista "Declaración".

Columna	Explicación
Dirección	Indica la dirección que STEP 7 asigna automáticamente a la variable al terminar de introducir una declaración.
Designación	Introduzca el nombre que debe asignar a cada variable.
Tipo	Introduzca el tipo de datos de la variable (BOOL, INT, WORD, ARRAY, etc.). Las variables pueden pertenecer a los tipos de datos simples, a los tipos de datos compuestos, o bien, a los tipos de datos de usuario.
Valor inic.	Indique el valor inicial, en caso de que el software no deba tomar el valor predeterminado del tipo de datos introducido. Todos los valores introducidos deben ser compatibles con los tipos de datos. Cuando guarde por primera vez el bloque de datos, el valor inicial será adoptado como valor actual de la variable, a menos que defina expresamente su valor actual.
Comentario	En este campo puede introducir un comentario para documentar la variable. El comentario no debe tener más de 80 caracteres.

### 10.4.2 Introducir / visualizar la estructura de bloques de datos con FB asociado (DBs de instancia)

#### *Introducción*

Si asocia un bloque de datos a un FB (FB de instancia), la declaración de variables del FB define la estructura del DB. Si desea realizar cambios, sólo podrá hacerlo en el FB asociado.

1. Abra el bloque de función asociado.
2. Edite la tabla de declaración de variables del bloque de función.
3. Cree nuevamente el bloque de datos.

### Visualización

En la vista "Declaración" del DB de instancia puede apreciar cómo se declararon las variables en el FB.

1. Abra el bloque de datos.
2. Active la vista "Declaración" del bloque de datos en caso de que ésta no se visualice todavía.
3. Más abajo se proporcionan explicaciones respecto a la tabla visualizada.

En los bloques de datos no globales no se puede modificar la vista "Declaración".

Columna	Explicación
Dirección	Indica la dirección que STEP 7 asigna automáticamente a la variable.
Declaración	Esta columna muestra cómo están declaradas las variables en la declaración del FB: <ul style="list-style-type: none"> <li>• parámetros de entrada (IN)</li> <li>• parámetros de salida (OUT)</li> <li>• parámetros de entrada/salida (IN_OUT)</li> <li>• datos estáticos (STAT)</li> </ul> Los datos temporales declarados del FB no están en el bloque de datos de instancia.
Nombre:	Indica el nombre definido para la variable en la declaración del FB.
Tipo	Indica los tipos de datos definidos para la variable en la declaración del FB. Las variables pueden pertenecer a los tipos de datos simples, a los tipos de datos compuestos, o bien, a los tipos de datos de usuario. Si dentro del FB se llama a más bloques de función para cuya llamada se hayan declarado variables estáticas, se puede indicar allí también un FB o bloque de función del sistema (SFB) como tipo de datos.
Valor inicial:	Se trata del valor predeterminado que usted ha definido en la declaración del FB para la variable, en caso de que el software no deba tomar el valor predefinido para el tipo de datos. Cuando guarde por primera vez el bloque de datos, el valor inicial se adoptará como valor actual de la variable, a menos que defina expresamente su valor actual.
Comentario	Se trata del comentario escrito en la declaración de variables del FB para documentar el elemento de datos. Este campo no se puede editar.

### Nota

En el caso de los bloques de datos asociados a un FB sólo podrá editar los valores actuales de las variables. La entrada de valores actuales de las variables se efectúa en la vista "Datos" de los bloques de datos.

### 10.4.3 Introducir la estructura de los tipos de datos de usuario (UDT)

1. Abra el tipo de datos de usuario asociado (UDT).
2. Active la vista "Declaración" del bloque de datos en caso de que ésta no se visualice todavía.
3. La estructura del tipo de datos de usuario se define fijando el orden de las variables, su tipo de datos y, en determinados casos, un valor inicial conforme a los datos siguientes.
4. Finalice la entrada de una variable pulsando la tecla TAB o la TECLA INTRO.

Columna	Explicación
Dirección:	Indica la dirección que STEP 7 asigna automáticamente a la variable al terminar de introducir una declaración.
Designación	Introduzca el nombre que debe asignar a cada variable.
Tipo	Introduzca el tipo de datos de la variable (BOOL, INT, WORD, ARRAY, etc.). Las variables pueden pertenecer a los tipos de datos simples, a los tipos de datos compuestos, o bien, a los tipos de datos de usuario.
Valor inic.	Indique el valor inicial cuando el software no deba adoptar el valor predefinido para el tipo de datos introducido. Todos los valores introducidos deben ser compatibles con los tipos de datos.  Cuando guarde por primera vez una instancia del UDT (de una variable o de un bloque de datos), el valor inicial se adoptará como valor actual de la variable, a menos que defina expresamente su valor actual.
Comentario	En este campo puede introducir un comentario para documentar la variable. El comentario no debe tener más de 80 caracteres.

### 10.4.4 Introducir / visualizar la estructura de bloques de datos con UDT asociado

#### Introducción

Si asocia un bloque de datos a un UDT, la estructura de datos del UDT definirá la estructura del DB. Los cambios sólo se pueden efectuar en el UDT asociado:

1. Abra el tipo de datos de usuario asociado (UDT).
2. Edite la estructura del tipo de datos de usuario.
3. Cree nuevamente el bloque de datos.

#### Visualización

En la vista "Declaración" del DB sólo se puede apreciar cómo se declararon las variables en el UDT:

1. Abra el bloque de datos.
2. Active la vista "Declaración" del bloque de datos en caso de que ésta no haya sido todavía visualizada.
3. Encontrará al final más información respecto a la tabla visualizada.

La vista "Declaración" no se puede modificar. Los cambios sólo se pueden efectuar en el UDT asociado.

Columna	Explicación
Dirección	Indica la dirección que STEP 7 asigna automáticamente a la variable.
Designación	Indica el nombre definido por el UDT para la variable.
Tipo:	Indica los tipos de datos definidos en el UDT. Las variables pueden pertenecer a los tipos de datos simples, a los tipos de datos compuestos, o bien, a los tipos de datos de usuario.
Valor inic.	Indica el valor predeterminado que se ha definido para la variable en el UDT, en caso de que el software no deba tomar el valor predefinido para el tipo de datos. Cuando guarde por primera vez el bloque de datos, el valor inicial se adoptará como valor actual de la variable, a menos que defina expresamente su valor actual.
Comentario	Se trata del comentario definido por el UDT para documentar el elemento de datos.

#### Nota

En el caso de los bloques de datos asociados a un UDT sólo podrá editar los valores actuales de las variables. La entrada de valores actuales de las variables se efectúa en la vista "Datos" de los bloques de datos.

### 10.4.5 Cambiar valores de datos en la vista "Datos"

Los valores actuales sólo se pueden editar en la vista "Datos" de bloques de datos.

1. En caso necesario, cambie la representación de la tabla a la vista "Datos" con el comando de menú **Ver > Datos**.
2. Introduzca los valores actuales deseados para los elementos de datos en los campos de la columna "Valor actual". Los valores actuales deberán ser compatibles con el tipo de datos de los elementos.

Las entradas erróneas se detectan inmediatamente y se resaltan en rojo (p.ej., cuando el valor actual introducido no sea compatible con el tipo de datos). Dichos errores deben eliminarse antes de guardar el bloque de datos.

#### Nota

Los cambios realizados se almacenan al guardar los bloques de datos.

### 10.4.6 Inicializar los valores de datos

La inicialización sólo es posible en la vista "Datos" de los bloques de datos.

1. En caso necesario, cambie la representación de la tabla a la vista "Datos" con el comando de menú **Ver > Datos**.
2. Elija para ello el comando de menú **Edición > Inicializar bloque de datos**.

Todas las variables vuelven a recibir el valor inicial previsto, es decir, los valores actuales de todas las variables son reemplazados por el valor inicial correspondiente.

---

#### Nota

Los cambios realizados se almacenan al guardar los bloques de datos.

---

### 10.4.7 Guardar bloques de datos

Para adoptar en la gestión de datos de la PG bloques de datos nuevos o valores modificados de bloques de datos, deberá guardar el bloque correspondiente, escribiendo los datos en el disco duro de la PG.

#### Para guardar bloques en el disco duro de la PG:

1. Active la ventana de trabajo del bloque que desea guardar.
2. Elija:
  - el comando de menú Archivo > Guardar si desea guardar el bloque con el mismo nombre, o
  - el comando de menú Archivo > Guardar como si desea guardar el bloque en otro programa de usuario S7, o bien con otro nombre. Introduzca la nueva ruta o el nuevo nombre del bloque en el cuadro de diálogo que aparece a continuación. Tenga en cuenta que el DB0 no se puede usar como número de DB, puesto que dicho número está reservado para el sistema.

En ambos casos, el bloque sólo se guardará si la sintaxis es correcta. Los errores sintácticos se detectan inmediatamente al crear el bloque y se destacan en rojo. Dichos errores deben corregirse antes de guardar el bloque.

---

#### Nota

Los bloques o las fuentes también pueden guardarse en otros proyectos o librerías desde el Administrador SIMATIC (p.ej., con arrastrar y soltar).

Los bloques o los programas de usuario completos sólo se pueden guardar en una Memory Card desde el Administrador SIMATIC.

Si surgen problemas al guardar o compilar bloques, tendrá que reorganizar el proyecto. Para ello elija en el Administrador SIMATIC el comando de menú **Archivo > Reorganizar**. Después vuelva a guardar o a compilar el bloque.

---

# 11 Crear fuentes AWL

## 11.1 Nociones básicas para programar en fuentes AWL

Existe la posibilidad de introducir el programa o partes del mismo en forma de fuente AWL y compilarlos a continuación en bloques. La fuente puede contener el código para varios bloques, los cuales se compilarán seguidamente de una sola pasada como bloques.

La creación de un programa mediante una fuente se tiene las siguientes ventajas:

- Permite crear y editar la fuente con cualquier editor ASCII, importarla a continuación y compilarla con esta aplicación en bloques individuales. Al compilar el archivo fuente se crearán los bloques que luego se depositarán en el programa de usuario S7.
- Se pueden programar varios bloques en una sola fuente.
- Las fuentes se pueden guardar aunque contengan errores de sintaxis. Ello no es posible al crear bloques lógicos con comprobación incremental de sintaxis. No obstante, hasta que no compile la fuente no se dará cuenta de los errores que contiene.

La fuente se crea respetando la sintaxis del lenguaje de programación "lista de instrucciones (AWL)". Se estructura en bloques, declaración de variables o segmentos mediante las palabras clave.

Al crear bloques en fuentes AWL se debe considerar lo siguiente:

- reglas de programación para fuentes AWL
- sintaxis y formatos permitidos por las fuentes AWL
- estructura de bloque permitida por las fuentes AWL

## 11.2 Reglas para introducir instrucciones en fuentes AWL

### 11.2.1 Reglas para introducir instrucciones en fuentes AWL

En principio, una fuente AWL se compone de texto continuo. Para poder compilarla en bloques se deben respetar ciertas estructuras y reglas de sintaxis.

Para crear programas de usuario en forma de fuentes AWL se deben respetar las siguientes reglas generales:

Tema	Regla
Sintaxis	La sintaxis de las instrucciones AWL es la misma que en el editor AWL incremental, a excepción de la instrucción de llamada CALL.
CALL	<p>En las fuentes, los parámetros se introducen entre paréntesis. Los parámetros se separan unos de otros utilizando comas.</p> <p>Ejemplo de llamada FC (una línea):            CALL FC 10 (param1 :=E0.0,param2 :=E0.1);</p> <p>Ejemplo de llamada FB (una línea):            CALL FB10, DB100 (para1 :=E0.0,para2 :=E0.1);</p> <p>Ejemplo de llamada FB (varias líneas):            CALL FB10, DB100 (                              para1 :=E0.0,                              para2 :=E0.1);</p> <p>Nota:            En el editor ASCII, en las llamadas de bloques hay que transferir los parámetros en el orden definido. De lo contrario no coincidirán los comentarios asignados en estas líneas al visualizar el programa en AWL o en forma de fuente.</p>
Mayúsculas / minúsculas	Por lo general, el editor de esta aplicación no distingue las mayúsculas de las minúsculas, excepto cuando se trata de atributos del sistema y metas de salto. Al introducir cadenas de caracteres (tipo de datos STRING) se deben tener en cuenta también las mayúsculas y las minúsculas. Las palabras clave se representan en mayúsculas. No obstante, al compilar no se tienen en cuenta las mayúsculas y las minúsculas. Por tanto, las palabras clave se pueden indicar tanto en mayúsculas como en minúsculas, o bien, mezclando ambas.
Punto y coma	Finalice todas las instrucciones AWL y todas las declaraciones de variables con un punto y coma. Se puede introducir más de una instrucción por línea.
Dos barras inclinadas (//)	Comience todos los comentarios con dos barras inclinadas (//) y finalícelos pulsando la tecla INTRO.

## 11.2.2 Reglas para declarar variables en fuentes AWL

Para cada bloque contenido en la fuente se deben declarar las correspondientes variables.

La declaración de variables se encuentra antes del área de instrucciones del bloque.

Las variables existentes deben declararse respetando el orden de los tipos de declaración. Así figurarán juntas todas las variables de un mismo tipo.

Mientras que con KOP, FUP o AWL es necesario rellenar una tabla de declaración de variables, aquí debe utilizar palabras clave.

### Palabras clave en la tabla de declaración de variables

Tipo de declaración	Palabras clave	rige para ...
Parámetros de entrada	"VAR_INPUT" Lista de declaración "END_VAR"	FBs, FCs
Parámetros de salida	"VAR_OUTPUT" Lista de declaración "END_VAR"	FBs, FCs
Parámetros de entrada/salida	"VAR_IN_OUT" Lista de declaración "END_VAR"	FBs, FCs
Variables estáticas	"VAR" Lista de declaración "END_VAR"	FBs
Variables temporales	"VAR_TEMP" Lista de declaración "END_VAR"	OBs, FBs, FCs

Con la palabra clave END\_VAR se termina la lista de declaración.

La lista de declaración es la lista de variables de un tipo de declaración. La lista permite preasignar valores a las variables (excepción: VAR\_TEMP). El ejemplo siguiente muestra la estructura de una entrada en la lista de declaración:

```

Duración_Motor1      :      S5TIME      :=      S5T#1H_30M  ;
Variable              Tipo de datos      Valor preasignado
    
```

#### Nota

- El nombre de la variable debe comenzar con una letra o con un carácter de subrayado. No se pueden dar nombres que correspondan a una de las palabras clave reservadas.
- Si los nombres de las variables de las declaraciones locales son idénticos a los de la tabla de símbolos, añade el signo # antes del nombre de las variables locales y ponga las variables de la tabla de símbolos entre comillas. En caso contrario, el bloque interpretará la variable como variable local.

### 11.2.3 Orden de los bloques en fuentes AWL

Los bloques llamados deben figurar delante de los bloques invocantes, es decir:

- El bloque OB1, que es el que más se usa para llamar otros bloques, debe figurar al final. Los bloques llamados por los bloques que invoca el OB1 tienen que figurar antes de éstos, etc.
- Los tipos de datos de usuario (UDTs) deben figurar delante de los bloques donde van a ser utilizados.
- Los bloques de datos con un tipo de datos de usuario asociado (UDT) deben figurar detrás del UDT.
- Los bloques de datos globales deben figurar delante de todos los bloques que los invocan.
- Un bloque de datos de instancia figura detrás del bloque de función asociado.
- El DB 0 se encuentra preasignado. Por tanto, no se puede crear ningún otro DB con dicho nombre.

### 11.2.4 Reglas para definir los atributos de sistema en fuentes AWL

Los atributos de sistema pueden ser asociados a bloques y parámetros. Se controla la configuración de mensajes y enlaces, las funciones de manejo y visualización y la configuración para control de procesos.

Para la entrada en la fuente hay que tener en cuenta:

- Las palabras clave de los atributos de sistema empiezan siempre con S7\_.
- Los atributos de sistema se especifican entre llaves.
- Sintaxis: {S7\_identifier := 'string'}  
Los diversos identificadores están separados por ";".
- Los atributos de sistema para bloques se encuentran delante de las propiedades de bloque y tras las palabras clave ORGANIZATION\_ o TITLE.
- Los atributos de sistema para parámetros se encuentran en la declaración de parámetros, es decir, delante de los dos puntos de la declaración de datos.
- Para introducir los atributos de sistema se diferencia entre las mayúsculas y las minúsculas.

Los atributos de sistema para bloques pueden ser controlados y/o modificados cuando se esté trabajando con el método de entrada incremental con el comando de menú **Archivo > Propiedades** en la ficha "Atributos".

Los atributos de sistema para parámetros pueden ser controlados y/o modificados cuando se esté trabajando con el método de entrada incremental con el comando de menú **Edición > Propiedades del objeto**. Para ello, el cursor debe encontrarse en el campo "Nombre" de la declaración de parámetros.

### 11.2.5 Reglas para definir las propiedades de bloques en fuentes AWL

Mediante las propiedades de bloques podrá identificar mejor los bloques que ha creado (p.ej., en las diferentes versiones) o proteger los bloques contra cambios no autorizados.

Las propiedades de bloques pueden ser controladas y/o modificadas cuando se está trabajando con el método de entrada incremental con el comando de menú **Archivo > Propiedades** en las fichas "General 1a. parte" y "General 2a. parte".

Todas las demás propiedades sólo se pueden introducir en la fuente.

Indicaciones para la fuente:

- Introduzca las propiedades de bloques antes de la tabla de declaración de variables.
- A cada propiedad le corresponde una línea propia.
- Al final de la línea no puede haber un punto y coma.
- Las propiedades de bloques se indican mediante palabras clave.
- Si desea introducir propiedades de bloques, es necesario respetar el orden de los mismos conforme a la tabla de propiedades de bloques.
- Las propiedades posibles para cada tipo de bloque se indican en la asignación de propiedades de bloques a tipos de bloques.

#### Nota

Las propiedades de los bloques se indican también en el Administrador SIMATIC, en las propiedades de objeto de un bloque. Allí pueden editarse también las propiedades AUTHOR, FAMILY, NAME y VERSION.

### Propiedades de bloques y orden

Si desea introducir propiedades de bloques, deberá respetar el orden indicado en la tabla que aparece a continuación.

Orden	Palabra clave / propiedad	Significado	Ejemplo
1.	[KNOW_HOW_PROTECT]	Protección de bloques: los bloques compilados con esta opción no permiten ver el área de instrucciones.	KNOW_HOW_PROTECT
2.	[AUTHOR:]	Nombre del autor: nombre de la empresa, del departamento u otros nombres (máx. 8 caracteres, sin espacios)	AUTHOR : Siemens, pero ninguna palabra clave
3.	[FAMILY:]	Nombre de la familia de bloques: p.ej. regulador. (máx. 8 caracteres, sin espacios).	FAMILY : Reguladores, pero ninguna palabra clave
4.	[NAME:]	Nombre del bloque (máx. 8 caracteres)	NAME : PID, pero ninguna palabra clave
5.	[VERSION: int1 . int2]	Número de versión del bloque (ambos números entre 0 y 15, es decir, 0.0 - 15.15)	VERSION : 3.10

Orden	Palabra clave / propiedad	Significado	Ejemplo
6.	[CODE_VERSION1]	Indica si un FB es multiinstancia. Si desea declarar multiinstancias, el FB no puede llevar dicha propiedad.	CODE_VERSION1
7.	[UNLINKED] sólo para DBs	Un bloque de datos con la propiedad UNLINKED no se incluye en el programa.	
8.	[READ_ONLY] sólo para DBs	Protección contra escritura para bloques de datos; sus datos sólo pueden leerse, pero no modificarse.	FAMILY= ejemplos VERSION= 3.10 READ_ONLY

### 11.2.6 Propiedades válidas para los diferentes tipos de bloques

La siguiente tabla muestra qué propiedades pueden asignarse a qué tipos de bloques.

Propiedad	OB	FB	FC	DB	UDT
KNOW_HOW_PROTECT	•	•	•	•	–
AUTHOR	•	•	•	•	–
FAMILY	•	•	•	•	–
DESIGNACION	•	•	•	•	–
VERSION	•	•	•	•	–
UNLINKED	–	–	–	•	–
READ_ONLY	–	–	–	•	–

#### Protección de bloques (KNOW\_HOW\_PROTECT)

Puede proteger sus bloques indicando la palabra clave KNOW\_HOW\_PROTECT al programar el bloque en la fuente AWL.

La protección de bloques tiene las consecuencias siguientes:

- Si desea visualizar posteriormente el bloque compilado en el editor AWL, FUP o KOP incremental, no podrá ver el área de instrucciones del bloque.
- En la tabla de declaración de variables del bloque se visualizan sólo las variables de tipo var\_in, var\_out y var\_in\_out. Las variables de tipo var\_stat y var\_temp permanecerán ocultas.
- Hay que introducir la palabra clave KNOW\_HOW\_PROTECT antes que las demás propiedades.

#### Protección contra escritura de bloques de datos (READ\_ONLY)

Para los bloques de datos puede activar una protección para evitar que éstos se sobrescriban al ejecutarse el programa. A tal efecto, el bloque de datos deberá existir como fuente AWL.

Introduzca en la fuente la palabra clave READ\_ONLY. Esta palabra deberá figurar en una línea propia inmediatamente antes de la declaración de variables.

## 11.3 Estructura de bloques en fuentes AWL

### 11.3.1 Estructura de bloques en fuentes AWL

Para estructurar los bloques en fuentes AWL se utilizan palabras clave. Dependiendo del tipo de bloque se diferencia la estructura de

- Bloques lógicos
- Bloques de datos
- Tipos de datos de usuario

### 11.3.2 Estructura de los bloques lógicos en fuentes AWL

Un bloque lógico se compone de las siguientes áreas, caracterizadas por las correspondientes palabras clave:

- Comienzo del bloque,
- Caracterizado por la palabra clave y el número o el nombre de bloque, como p.ej.
  - "ORGANIZATION\_BLOCK OB 1" para un bloque de organización,
  - "FUNCTION\_BLOCK FB6" para un bloque de función,
  - "FUNCTION FC 1 : INT" para una función. En el caso de las funciones se indica adicionalmente el tipo de función. Su tipo de datos puede ser simple o compuesto (excepto ARRAY y STRUCT), determinando el tipo de datos del valor de respuesta (RET\_VAL). Si no se ha de obtener nuevamente ningún valor, hay que indicar la palabra clave VOID.
- Título del bloque (opcional), comenzando con la palabra clave "TITLE (longitud máxima del título: 64 caracteres).
- Comentario adicional, comenzando con dos barras inclinadas //.
- Indicación de las propiedades (opcional).
- Tabla de declaración de variables.
- Área de instrucciones, comenzando con BEGIN. El área de instrucciones se compone a su vez de uno o varios segmentos caracterizados con NETWORK. No se permite introducir números de segmento.
- Título opcional para cada segmento implementado, comenzando con la palabra clave "TITLE =" (longitud máxima del título: 64 caracteres).
- Comentarios adicionales para cada segmento, comenzando con dos barras inclinadas //.
- Fin del bloque, caracterizado por END\_ORGANIZATION\_BLOCK, END\_FUNCTION\_BLOCK o END\_FUNCTION.
- Entre el tipo y el número de bloque debe haber un espacio en blanco. El nombre simbólico del bloque puede caracterizarse con comillas para no confundir los nombres de las variables locales con los nombres de la tabla de símbolos.

### 11.3.3 Estructura de bloques de datos en fuentes AWL

Un bloque de datos se compone de las siguientes áreas que comienzan con la palabra clave correspondiente:

- Comienzo del bloque, caracterizado por la palabra clave y el número o nombre de bloque, p.ej. DATA\_BLOCK DB 26
- Indicación del UDT o del FB asociado al DB (opcional).
- Título opcional del bloque, comenzando con la palabra clave "TITLE =" (los títulos de más de 64 caracteres se cortan).
- Comentario opcional del bloque, comenzando con "//".
- Indicación de las propiedades (opcional).
- Tabla de declaración de variables (opcional)
- Area de asignación con valores preasignados, comenzando con BEGIN (opcional)
- Fin del bloque, comenzando con END\_DATA\_BLOCK.

Hay tres tipos de bloques de datos:

- Bloques de datos de usuario
- Bloques de datos con tipo de datos de usuario asociado (UDT)
- Bloques de datos asociados a un FB (DB de instancia)

### 11.3.4 Estructura de tipos de datos de usuario en fuentes AWL

Un tipo de datos de usuario se compone de las siguientes áreas, que comienzan con las correspondientes palabras clave:

- Comienzo del bloque, caracterizado por la palabra clave TYPE y el número o nombre, p.ej. TYPE UDT 20
- Indicación de un tipo de datos estructurado
- Fin del bloque, caracterizado por END\_TYPE.

Los tipos de datos de usuario tienen que introducirse antes que los bloques en los que se van a utilizar.

## 11.4 Sintaxis y formatos de bloques en fuentes AWL

### 11.4.1 Sintaxis y formatos de bloques en fuentes AWL

Las tablas indican la sintaxis y los formatos que se debe tener en cuenta al programar fuentes AWL. La sintaxis se representa de la manera siguiente:

- Cada elemento se describe en la columna derecha.
- Los elementos imprescindibles se representan entre comillas simples.
- Los corchetes [...] significan que no es necesario indicar el contenido de los mismos.
- Las palabras clave se indican en mayúsculas.

### 11.4.2 Formato de los OBs

La siguiente tabla muestra un resumen del formato de los bloques de organización en una fuente AWL:

Estructura	Descripción
"ORGANIZATION_BLOCK" ob_nr u ob_name	ob_nr es el número del bloque, p.ej., OB 1; ob_name es el nombre simbólico del bloque tal como figura en la tabla de símbolos
[TITLE= ]	Comentario hasta <RETURN>; los títulos de más de 64 caracteres quedan cortados.
[Comentario de bloque]	Los comentarios se introducen comenzando con "/"
[Atributos de sistema para bloques]	Atributos de sistema para bloques
[Propiedades de bloques]	Propiedades de bloques
Tabla de declaración de variables	Declaración de variables temporales
"BEGIN"	Palabra clave para separar las instrucciones AWL de la tabla de declaración de variables
NETWORK	Comienzo de un segmento
[TITLE= ]	Título de segmento (máx. 64 caracteres).
[Comentario de segmento]	Los comentarios se introducen comenzando con "/"
Lista de instrucciones AWL	Instrucciones del bloque
"END_ORGANIZATION_BLOCK"	Palabra clave para finalizar el bloque de organización

### 11.4.3 Formato de los FBs

La siguiente tabla muestra un resumen del formato de los bloques de función en una fuente AWL:

<b>Estructura</b>	<b>Descripción</b>
"FUNCTION_BLOCK" fb_nr ó fb_name	fb_nr es el número del bloque, p.ej., FB 6; fb_name es el nombre simbólico del bloque tal como figura en la tabla de símbolos
[TITLE= ]	Comentario hasta <RETURN>; los títulos de más de 64 caracteres quedan cortados.
[Comentario de bloque]	Los comentarios se introducen comenzando con "///"
[Atributos de sistema para bloques]	Atributos de sistema para bloques
[Propiedades de bloques]	Propiedades de bloques
Tabla de declaración de variables	Declaración de parámetros de entrada, de salida y de entrada/salida, así como de variables temporales o estáticas La declaración de parámetros puede contener las declaraciones de los atributos de sistema para los parámetros
"BEGIN"	Palabra clave para separar las instrucciones AWL de la tabla de declaración de variables
NETWORK	Comienzo de un segmento
[TITLE= ]	Título de segmento (máx. 64 caracteres).
[Comentario de segmento]	Los comentarios se introducen comenzando con "///".
Lista de instrucciones AWL	Instrucciones del bloque
"END_FUNCTION_BLOCK"	Palabra clave para terminar el bloque de función

### 11.4.4 Formato de las FCs

La siguiente tabla muestra un resumen del formato de las funciones en una fuente AWL:

Estructura	Descripción
"FUNCTION"    fc_nr : fc_type ó fc_name : fc_type	fc_nr es el número del bloque, p.ej., FC 5; fc_name es el nombre simbólico del bloque tal como figura en la tabla de símbolos fc_type indica el tipo de datos al que pertenece el valor de respuesta (RET_VAL) de la función. Puede ser un tipo de datos simple o compuesto, o bien, de tipo VOID. Si desea utilizar atributos de sistema para el valor de respuesta (RET_VAL), deberá introducir los atributos de sistema de los parámetros antes de los dos puntos de la declaración de datos.
[TITLE= ]	Comentario hasta <RETURN>; los títulos de más de 64 caracteres quedan cortados.
[Comentario de bloque]	Los comentarios se introducen comenzando con "///"
[Atributos de sistema para bloques]	Atributos de sistema para bloques
[Propiedades de bloques]	Propiedades de bloques
Tabla de declaración de variables	Declaración de parámetros de entrada, de salida y de entrada/salida, así como de variables temporales
"BEGIN"	Palabra clave para separar las instrucciones AWL de la tabla de declaración de variables
NETWORK	Comienzo de un segmento
[TITLE= ]	Título de segmento (máx. 64 caracteres).
[Comentario de segmento]	Los comentarios se introducen comenzando con "///".
Lista de instrucciones AWL	Instrucciones del bloque
"END_FUNCTION"	Palabra clave para finalizar la función

### 11.4.5 Formato de los DBs

La siguiente tabla muestra un resumen del formato de los bloques de datos en una fuente AWL:

<b>Estructura</b>	<b>Descripción</b>
"DATA_BLOCK" db_nr ó db_name	db_nr es el número del bloque, p.ej., DB 5; db_name es el nombre simbólico del bloque tal como figura en la tabla de símbolos
[TITLE= ]	Comentario hasta <RETURN>; los títulos de más de 64 caracteres quedan cortados.
[Comentario de bloque]	Los comentarios se introducen comenzando con "///"
[Atributos de sistema para bloques]	Atributos de sistema para bloques
[Propiedades de bloques]	Propiedades de bloques
Area de declaración	Indicación del UDT o FB al que se refiere el bloque. Puede ser un número de bloque, un nombre simbólico o la indicación del tipo de datos compuesto.
"BEGIN"	Palabra clave para separar la tabla de declaración de la lista de asignación de valores.
[Asignación de valores iniciales]	A las variables se les pueden asignar determinados valores iniciales. Se pueden asignar constantes a variables o referenciar otros bloques.
"END_DATA_BLOCK"	Palabra clave para finalizar el bloque de datos

## 11.5 Crear fuentes AWL

### 11.5.1 Crear fuentes AWL

La fuente se debe crear debajo del programa S7 en una carpeta de fuentes. Las fuentes se pueden crear en el Administrador SIMATIC o en la ventana del editor.

#### *Crear fuentes en el Administrador SIMATIC*

1. Abra la carpeta "Fuentes" en cuestión, haciendo doble clic en la misma.
2. Para insertar una fuente AWL, elija el comando de menú **Insertar > Software S7 > Fuente AWL**.

#### *Crear fuentes en la ventana del editor*

1. Elija el comando **Archivo > Nuevo**.
2. En el cuadro de diálogo que aparece a continuación, elija la carpeta de fuentes del mismo programa S7, en el que se encuentra también el programa de usuario con los bloques.
3. Asígnele un nombre a la fuente que desea crear.
4. Confirme pulsando el botón "Aceptar".

La fuente se creará con el nombre que usted le haya dado y se visualizará en una ventana de trabajo para poderla editar posteriormente.

### 11.5.2 Editar fuentes S7

Las propiedades de las fuentes determinan en qué lenguaje de programación y con qué editor se ha de editar la fuente en cuestión. Al abrirse una fuente se arrancará siempre el editor correcto con el correspondiente lenguaje de programación. El software básico STEP 7 asiste la programación en fuentes AWL.

En el software opcional se ofrecen otros lenguajes de programación adicionales. Si ha cargado el software opcional en cuestión en el ordenador, podrá arrancar el editor correspondiente haciendo doble clic en la fuente.

Proceda de la siguiente forma:

1. Abra la carpeta "Fuentes" en cuestión, haciendo doble clic en la misma.
2. Arranque el editor necesario
  - haciendo doble clic en la fuente deseada en la mitad derecha de la ventana o
  - seleccionando la fuente en la mitad derecha de la ventana y activando el comando Edición > Abrir objeto.

### 11.5.3 Insertar plantillas de bloques en fuentes AWL

Para programar en fuentes AWL se le ofrecen plantillas de bloques para OBs, FBs, FCs, DBs, DBs de instancia, DBs de UDTs y UDTs. Las plantillas de bloques le ayudan a introducir y respetar la sintaxis y la estructura.

Proceda de la siguiente forma:

1. Active la ventana de trabajo de la fuente en la que desea insertar la plantilla.
2. Sitúe el cursor en la posición detrás de la cual desea insertar la plantilla de bloque.
3. Elija la plantilla deseada con el correspondiente comando de menú **Insertar > Plantilla de bloques > OB/FB/FC/DB/IDB/DB de UDT/UDT**.

La plantilla de bloques se insertará detrás de la posición del cursor.

### 11.5.4 Insertar fuentes externas

Puede crear y editar su fuente con cualquier editor ASCII, importarla a continuación en un proyecto y compilarla en bloques individuales. A tal efecto es necesario importar las fuentes a la carpeta "Fuentes" del programa S7, en cuyo programa de usuario S7 se hayan de depositar los bloques creados en la compilación.

Para compilar y cargar una tabla GD:

1. Seleccione la carpeta "Fuentes" del programa S7 a la que desea importar las fuentes externas.
2. Elija el comando **Insertar > Fuente externa**.
3. En el cuadro de diálogo que aparece a continuación, indique la fuente que desea importar.

El nombre de archivo de la fuente a importar deberá tener una extensión válida. El tipo de fuente se define en STEP 7 conforme a la extensión del archivo. Por ejemplo, al importar un archivo con la extensión **.awl** se creará una fuente AWL. Las extensiones válidas se indican en el cuadro de diálogo bajo "Tipo de archivo".

**Nota**

Con el comando **Insertar > Fuente externa** puede insertar también las fuentes que haya creado con la versión 1 de STEP 7.

---

### 11.5.5 Generar fuentes AWL de bloques

Puede crear una fuente AWL con bloques existentes, editándola con cualquier editor de texto. La fuente generada se encuentra en la carpeta de fuentes del mismo programa S7 de cuyo programa de usuario se seleccionan los bloques.

Para compilar y cargar una tabla GD:

1. Elija el comando de menú **Archivo > Generar fuente**.
2. Elija en el cuadro de diálogo la carpeta de fuentes en la que desea crear la nueva fuente.
3. Introduzca en el cuadro de texto el nombre que desea darle a la fuente.
4. Elija en el cuadro de diálogo "Seleccionar bloques de STEP7" el o los bloques que desee generar en la fuente indicada anteriormente. Los bloques seleccionados aparecerán en el cuadro de lista a la derecha.
5. Confirme pulsando el botón "Aceptar".

Con los bloques seleccionados se genera una fuente AWL coherente que se muestra en una ventana de trabajo para su posterior edición.

## 11.6 Guardar y compilar fuentes AWL y comprobar su coherencia

### 11.6.1 Guardar fuentes AWL

Las fuentes AWL se pueden guardar en todo momento en su estado actual. En dicha operación no se compila el programa ni tampoco se comprueba la sintaxis, por lo que también se guardan los posibles errores.

Los errores de sintaxis se detectan y se indican al compilar la fuente o al comprobarse la coherencia.

*Para guardar una fuente con el mismo nombre:*

1. Active la ventana de trabajo de la fuente que desea guardar.
2. Elija el comando de menú **Archivo > Guardar**.

*Para guardar una fuente con otro nombre/en otro proyecto:*

1. Active la ventana de trabajo de la fuente que desea guardar.
2. Elija el comando de menú **Archivo > Guardar como**.
3. Elija en el cuadro de diálogo que aparece a continuación la carpeta de fuentes a la que se ha de asignar la fuente e introduzca su nuevo nombre.

### 11.6.2 Comprobar la coherencia en fuentes AWL

Con el comando de menú **Archivo > Comprobar coherencia** se pueden visualizar los posibles errores de sintaxis en la fuente AWL. A diferencia de la compilación, al comprobarse la coherencia no se generan bloques.

Una vez comprobada la coherencia aparece un cuadro de diálogo que le comunica el número total de errores localizados.

Si existen errores, éstos se listan individualmente en la parte inferior de la ventana de trabajo, indicándose la línea correspondiente. Los errores deben corregirse antes de compilar la fuente para que puedan crearse todos los bloques.

### 11.6.3 Buscar errores en fuentes AWL

La ventana de trabajo de las fuentes está dividida en dos. En la mitad inferior se listan los siguientes errores:

- los errores que se detectan después de iniciar un proceso de compilación con el **comando de menú Archivo > Compilar** o
- los errores que se detectan después de iniciar una comprobación de coherencia, con el comando de menú **Archivo > Comprobar coherencia**.

Para poder encontrar un error en la fuente, sitúe el cursor en el correspondiente mensaje de error que aparece en la mitad inferior de la ventana. Así se selecciona automáticamente la correspondiente línea de texto en la mitad superior. Además, el mensaje de error aparece en la barra de estado.

## 11.6.4 Compilar fuentes AWL

### *Requisitos*

Para poder compilar en bloques un programa creado en una fuente, se deberán cumplir las siguientes condiciones:

- Sólo se pueden compilar fuentes que estén guardadas en la carpeta "Fuentes" bajo un programa S7.
- Además de la carpeta "Fuentes" también deberá existir bajo el programa S7 una carpeta "Bloques" en la que se guardarán los bloques que se creen en la compilación. Los bloques programados en la fuente sólo se crearán si la fuente ha sido compilada sin errores. Si una fuente comprende varios bloques se crearán sólo los que no contengan errores. Luego puede abrir, editar, cargar en la CPU y comprobar cada bloque.

### *Procedimiento en el editor*

1. Abra la fuente que desea compilar. La fuente debe encontrarse en la carpeta de fuentes del programa S7 en cuyo programa de usuario S7 desea depositar los bloques compilados.
2. Elija el comando de menú **Ver > Mostrar > Representación simbólica** para que los símbolos puedan visualizarse luego en los bloques compilados.
3. Elija el comando de menú **Archivo > Compilar**.
4. A continuación aparece el cuadro de diálogo "Informe de la compilación" que indica el número de líneas compiladas y de errores de sintaxis encontrados.

Los bloques indicados en el archivo sólo se crean si la fuente ha sido compilada sin errores. Si una fuente comprende varios bloques se crearán sólo los que no contengan errores. Las advertencias no impiden que se creen bloques.

Los errores de sintaxis que se detectan durante la compilación se representarán en la mitad inferior de la ventana de trabajo y deben corregirse para que se puedan crear también dichos bloques.

### *Procedimiento en el Administrador SIMATIC*

1. Abra la carpeta "Fuentes" en cuestión, haciendo doble clic en la misma.
2. Elija una o varias fuentes a compilar. No es posible arrancar la compilación de una carpeta "Fuentes" cerrada para compilar todas las fuentes contenidas en la misma.
3. Elija el comando **Archivo > Compilar**, para arrancar la compilación. Dependiendo de la fuente elegida se arrancará el correspondiente compilador. Los bloques que se hayan compilado sin errores se depositarán a continuación en la carpeta "Bloques" bajo el programa S7.

Los errores de sintaxis detectados durante la compilación se indican en un cuadro de diálogo, siendo necesario corregirlos para que se puedan crear también esos bloques.

## 11.7 Ejemplos de fuentes AWL

### 11.7.1 Ejemplos de declaración de variables en fuentes AWL

#### Variables del tipo de datos simples

```

// Los comentarios se separan de la tabla de declaración con dos barras
// inclinadas.
VAR_INPUT // Palabra clave variable de entrada
in1 : INT; // Nombre y tipo de la variable se separan con un ":"
in3 : DWORD; // Cada declaración de variable se termina con un punto y coma.
in2 : INT := 10; // Definición opcional del valor inicial en la declaración
END_VAR // Fin de la declaración de variables de un mismo tipo de declaración
VAR_OUTPUT // Palabra clave variable de salida
out1 : WORD;
END_VAR // Palabra clave variable temporal
VAR_TEMP
temp1 : INT;
END_VAR

```

#### Variables de tipo de datos "Array"

```

VAR_INPUT // Variable de entrada
campo1 : ARRAY [1..20] of INT; // campo1 es un campo unidimensional
bidimensional
campo2 : ARRAY [1..20, 1..40] of DWORD; // campo2 es un campo
END_VAR

```

#### Variables del tipo de datos "estructura"

```

VAR_OUT // Variable de salida
SALIDA1: STRUCT // SALIDA1 es del tipo de datos STRUCT
var1 : BOOL; // Elemento 1 de la estructura
var2 : DWORD; // Elemento 2 de la estructura
END_STRUCT; // Fin de la estructura
END_VAR

```

## 11.7.2 Ejemplo de OBs en fuentes AWL

```

ORGANIZATION_BLOCK OB 1
TITLE = Ejemplo de OB 1 con diferentes llamadas de bloques
// Los 3 segmentos representados muestran llamadas de bloques
//con y sin parámetros

{S7_pdiag := 'true'} //Atributo de sistema para bloques
AUTHOR:      :      Siemens
FAMILY      :      Ejemplo
Nombre      :      Test_OB
VERSION     :      1.1
VAR_TEMP
Valor intermedio : INT;           // Memoria intermedia
END_VAR

BEGIN

NETWORK
TITLE = Llamada de una función con transferencia de parámetros
// Transferencia de parámetros en una línea
CALL FC1 (param1 :=E0.0,param2 :=E0.1);

NETWORK
TITLE = Llamada de un bloque de función con
// transferencia de parámetros
// transferencia de parámetros en varias líneas
CALL control de semáforo , DB 6 ( // Nombre del FB, bloque de datos de instancia
dur_g_p      := S5T#10S, // Asignación de valores actuales a parámetros

del_r_p      := S5T#30S,
starter      := TRUE,
t_dur_y_car  := T 2,
t_dur_g_ped  := T 3,
t_delay_y_car := T 4,
t_dur_r_car  := T 5,
t_next_red_car := T 6,
r_car       := "re_main", // Agregar comillas
y_car       := "ye_main", // Nombres de la tabla de símbolos
g_car       := "gr_main",
r_ped       := "re_int",
g_ped       := "gr_int");

NETWORK
TITLE = Llamada de un bloque de función con
// transferencia de parámetros
// indicación de parámetro en una línea
CALL FB10, DB100 (para1 :=E0.0,para2 :=E0.1);

END_ORGANIZATION_BLOCK

```

### 11.7.3 Ejemplo de FCs en fuentes AWL

```

FUNCTION FC 1: VOID
// ¡Sólo a causa del Call!!
VAR_INPUT
  param1 : bool;
  param2 : bool;
END_VAR
begin
end_function

FUNCTION FC2 : INT
TITLE = Aumento del número de piezas
// Mientras que el valor transferido sea < 1000, esta función aumenta el
//valor transferido. Si el número de piezas es superior a 1000, se obtiene
// nuevamente la función (RET_VAL) "-1" a través del valor de respuesta.

AUTHOR      :      Siemens
FAMILY      :      Por
Nombre      :      STUECK
VERSION     :      1.0

VAR_IN_OUT
STUECKZAHL : INT;           // Número de piezas fabricado hasta ahora
END_VAR

BEGIN

NETWORK
TITLE = Aumento del número de piezas en 1
// En tanto que el número de piezas actual sea inferior a 1000,
// puede incrementarse (+1)
L STUECKZAHL; L 1000;           // Ejemplo de varias
> I; SPB ERR;                   // instrucciones en una línea.
L 0; T RET_VAL;
L STUECKZAHL; INC 1; T STUECKZAHL; BEA;
ERR: L -1;
T RET_VAL;
END_FUNCTION

```

```

FUNCTION FC3 {S7_pdiag := 'true'} : INT
TITLE = Aumento del número de piezas
// Mientras que el valor transferido sea < 1000, esta función aumenta el
//valor transferido. Si el número de piezas es
//superior a 1000, se obtiene
// nuevamente la función (RET_VAL) "-1" a través del valor de respuesta.
//
//RET_VAL tiene aquí un atributo de sistema para parámetros
AUTHOR      :      SIEMENS
FAMILY      :      POR
NOMBRE      :      STUECK
VERSION     :      1.0

VAR_IN_OUT
STUECKZAHL {S7_visible := 'true'}: INT;    // Número de piezas fabricado actualmente
//Atributos de sistema para parámetros
END_VAR

BEGIN

NETWORK
TITLE = Aumento del número de piezas en 1
// En tanto que el número de piezas actual sea inferior a 1000,
// puede incrementarse (+1)
L STUECKZAHL; L 1000;                      // Ejemplo de varias
> I; SPB ERR;                               // instrucciones en una línea.
L 0; T RET_VAL;
L STUECKZAHL; INC 1; T STUECKZAHL; BEA;
ERR: L -1;
T RET_VAL;
END_FUNCTION

```

### 11.7.4 Ejemplo de FBs en fuentes AWL

```

FUNCTION_BLOCK FB 6
TITLE = Control de semáforo sencillo
// Control de un semáforo para un paso de peatones
// en la calle principal

{S7_m_c := 'true'}           //Atributo de sistema para bloques
AUTHOR      :      SIEMENS
FAMILY      :      Semáforo
Nombre     :      Semáforo01
VERSION    :      1.3

VAR_INPUT

starter      :      BOOL    :=      FALSE; // Petición de un peatón de cruzar la calle
t_dur_y_car  :      TIMER;   // Duración fase verde peatón
t_next_r_car :      TIMER;   // Duración entre fases rojas para vehículos
t_dur_r_car  :      TIMER;

anz          {S7_server := 'alarm_archiv'; S7_a_type := 'alarm_8'} :DWORD;
// Número de vehículos
// anz tiene un atributo de sistema para parámetros

END_VAR
VAR_OUTPUT

g_car        :      BOOL          :=      FALSE; // VERDE para vehículos

END_VAR
VAR
condition    :      BOOL    :=      FALSE; // Petición de cambiar a rojo para
//vehículos
END_VAR

BEGIN
NETWORK
TITLE = Petición de cambiar a rojo para los vehículos
// Una vez transcurrido un tiempo mínimo,
//la señal del paso de peatones
//indicará la petición de cambiar a rojo para los vehículos.
      U(
      U      #starter;           // Señal en el paso de peatones y vencimiento
      U      #t_next_r_car;     // del tiempo entre dos fases rojas
      O      #condition;        // o petición de cambiar a rojo ("autoparada")
      );
      UN      #t_dur_y_car;     // y actualmente sin semáforo rojo
      =      #condition;        // petición de cambiar a rojo
    
```

```

NETWORK
TITLE = Semáforo verde para los vehículos
        UN    #condition;    // sin petición de cambiar a rojo para los vehículos
        =    #g_car;        // VERDE para los vehículos

NETWORK
TITLE = Duración de la fase amarilla para los vehículos
        // programa adicional para realizar el
        // control del semáforo

END_FUNCTION_BLOCK

FUNCTION_BLOCK FB 10
VAR_INPUT
    para1 : bool;
    para2: bool;
end_var
begin
end_function_block

data_block db 10
FB 10
begin
end_data_block

data_block db 6
FB 6
begin
end_data_block

```

### 11.7.5 Ejemplos de DBs en fuentes AWL

#### Bloque de datos:

```

DATA_BLOCK DB 10
TITLE = DB Ejemplo 10
STRUCT
        aa : BOOL;    // variable aa del tipo BOOL
        bb : INT;    // variable bb del tipo INT
        cc : WORD;

END_STRUCT;
BEGIN    // Asignación de valores iniciales
        aa := TRUE;
        bb := 1500;
END_DATA_BLOCK

```

### DB con tipo de datos de usuario asociado:

```
DATA_BLOCK DB 20
TITLE = DB (UDT) Ejemplo
UDT 20           // Indicación de los UDTs asociados
BEGIN
    start := TRUE;           // Asignación de valores iniciales
    soll := 10;
END_DATA_BLOCK
```

---

#### Nota

El UDT utilizado debe figurar en la fuente antes del bloque de datos.

---

### DB con bloque de función asociado:

```
DATA_BLOCK DB 30
TITLE = DB (FB) Ejemplo
FB 30           // Indicación de los FBs asociados
BEGIN
    start := TRUE;           // Asignación de valores iniciales
    soll := 10;
END_DATA_BLOCK
```

---

#### Nota

El FB debe figurar en la fuente antes del bloque de datos.

---

## 11.7.6 Ejemplo de UDTs en fuentes AWL

```
TYPE UDT 20
STRUCT
    arranque : BOOL;           //variable del tipo BOOL
    teórico : INT;             //variable del tipo INT
    valor : WORD;              //variable del tipo WORD
END_STRUCT;
END_TYPE
```

## 12 Mostrar datos de referencia

### 12.1 Sinopsis de los datos de referencia

#### 12.1.1 Sinopsis de los datos de referencia

Es posible crear y evaluar datos de referencia para probar y modificar cómodamente el programa de usuario. Los datos de referencia se utilizan, p.ej., como

- sinopsis de todo el programa de usuario,
- base para efectuar cambios y tests y
- complemento de la documentación del programa.

En la tabla siguiente se indica la información que figura en las diversas vistas:

Ver	Aplicación
Lista de referencias cruzadas	Panorámica de los operandos de las áreas de memoria E, A, M, P, T, Z y de los DB, FB, FC, SFB, SFC utilizados en el programa de usuario.
Plano de ocupación Plano de ocupación para temporizadores y contadores (T/Z)	Panorámica de los bits de los operandos de las áreas de memoria E, A y M, así como de los temporizadores y contadores que ya estén ocupados en el programa de usuario. Es una base importante para buscar errores o para efectuar cambios en el programa de usuario.
Estructura del programa	Jerarquía de llamada de los bloques en un programa de usuario y panorámica de los bloques utilizados y de su interdependencia.
Símbolos no utilizados	Panorámica de los símbolos definidos en la tabla de símbolos, pero que no se utilizan en las partes del programa de usuario para las que existen datos de referencia.
Operandos sin símbolo	Panorámica de las direcciones absolutas (operandos absolutos y bloques) que se utilizan en las partes del programa de usuario para las que existen datos de referencia. No obstante, para dichas direcciones no se ha definido ningún símbolo en la tabla de símbolos.

Los datos de referencia del programa de usuario seleccionado comprenden las listas mencionadas en la tabla. Es posible crear y visualizar varias listas de un programa de usuario o de diversos programas.

## Visualizar varias vistas a la vez

Visualizando listas en distintas ventanas se puede, por ejemplo:

- comparar una determinada lista para distintos programas de usuario S7.
- tener en la pantalla, una junto a otra, distintas vistas de una lista optimizadas de distinta manera, p.ej., una lista de referencias cruzadas. Así puede visualizar, p.ej., en una lista de referencias cruzadas sólo las entradas y en la otra, sólo las salidas de un programa de usuario.
- abrir varias listas de un programa de usuario S7 a la vez, p.ej., la estructura del programa y la lista de referencias cruzadas.

### 12.1.2 Lista de referencias cruzadas

La lista de referencias cruzadas ofrece una panorámica de los operandos utilizados en el programa de usuario S7.

En la lista de referencias cruzadas se visualizan los operandos de las áreas de memoria de entradas (E), salidas (A), marcas (M), temporizadores (T), contadores (Z), bloques de función (FB), funciones (FC), bloques de función de sistema (SFB), funciones de sistema (SFC), periferia (P) y bloque de datos (DB), así como sus direcciones (dirección absoluta, nombre) y su utilización. La lista se visualiza en una ventana de trabajo. En la barra del título de esta ventana se muestra el nombre del programa de usuario al que pertenece la lista de referencias cruzadas.

Cada línea de la ventana corresponde a una entrada de la lista de referencias cruzadas. La función de búsqueda permite encontrar puntualizadamente los operandos y los símbolos.

La lista de referencias cruzadas es la representación estándar de los datos de referencia. Este ajuste predeterminado puede ser modificado.

### Estructura

Cada entrada de la lista de referencias cruzadas tiene las siguientes columnas:

Columna	Contenido / significado
Operando	Dirección absoluta del operando
Símbolo	Nombre del operando
Bloque	Indicación del bloque donde se utiliza el operando
Acceso	Tipo de acceso al operando: de lectura (R) o de escritura (W)
Lenguaje/Detalles	Información sobre el bloque, depende del lenguaje de programación del bloque

Las columnas símbolo, bloque, tipo y lenguaje/detalles serán mostradas sólo si se han seleccionado las correspondientes propiedades para la lista de referencias cruzadas. La información sobre el lenguaje y los detalles se representan en una misma columna, por lo que sólo se podrán seleccionar o rechazar como columna entera. Esta información sobre el bloque varía según el lenguaje en el que se creó el bloque.

Con el ratón puede modificar el ancho de las columnas de la lista de referencias cruzadas.

## Ordenación

La lista de referencias cruzadas está ordenada por áreas de memoria, siendo éste el ajuste predeterminado. Si hace clic en un título de una columna, puede ordenar las entradas de la misma.

## Ejemplo de la estructura de la lista de referencias cruzadas

Operando	Símbolo	SFC	Acceso	Lenguaje	Detalles
E 1.0	Motor on	OB 2	R	AWL	NW 2 Anw 33 /O
M1.2	Bit de marcas	FC 2	R	KOP	Seg33
Z2	Contadores2	FB 2		FUP	Seg2

### 12.1.3 Estructura del programa

La estructura del programa describe la jerarquía de llamada de los bloques dentro de un programa de usuario S7. Además ofrece una visión general de los bloques utilizados, su interdependencia y su requerimiento de datos locales.

Con el comando de menú **Ver > Filtrar** en la ventana "Mostrar datos de referencia S7" se abre un cuadro de diálogo compuesto por varias fichas. En la ficha "Estructura del programa" puede definir la representación de la estructura del programa.

Puede elegir entre

- estructura en árbol y
- estructura por padres e hijos.

Puede indicar si desea que se representen todos los bloques o sólo la jerarquía a partir de un bloque determinado.

## Símbolos de la estructura del programa

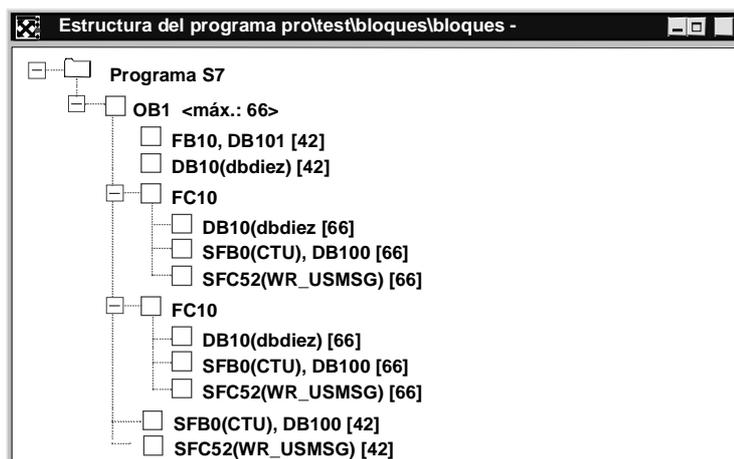
Símbolo	Significado
<input type="checkbox"/>	Llamada regular de bloque (CALL FB10)
<input type="checkbox"/>	Llamada incondicional de bloque (UC FB10)
<input type="checkbox"/>	Llamada condicional de bloque (CC FB10)
<input type="checkbox"/>	Bloque de datos
<input type="checkbox"/>	Recursión
<input type="checkbox"/>	Recursión y llamada condicional
<input type="checkbox"/>	Recursión y llamada incondicional
<input type="checkbox"/>	bloque no llamado

- Los bloques que se llamen repetidamente se destacan gráficamente en un árbol.
- Las recursiones dentro de la jerarquía de llamada se representan con distintos símbolos.
- Los bloques llamados regularmente (CALL), los llamados condicionalmente (CC) o los llamados incondicionalmente (UC) se caracterizan con distintos símbolos.
- Los bloques que no han sido llamados se visualizan al final de la estructura del programa, están marcados con una cruz negra y su estructura no será desglosada.

## Estructura en árbol

Se representa toda la jerarquía de llamada a partir de un bloque determinado.

Todas las estructuras del programa tienen un determinado bloque como raíz. Este bloque puede ser el OB1 o cualquier otro que el usuario defina previamente como bloque de arranque.



Si quiere crear la estructura del programa para la totalidad de los bloques de organización (OB), pero el OB 1 no está en el programa de usuario S7, o se ha indicado un bloque de arranque que no existe en el programa, entonces se le pedirá automáticamente que especifique otro bloque para que sirva como raíz de la estructura.

Ajustando convenientemente las opciones se puede desactivar la visualización de llamadas múltiples de bloques, tanto en lo que se refiere a la estructura en árbol como a la "estructura por padres e hijos".

## Visualización del requerimiento máximo de datos locales en la estructura en árbol

Para facilitarle una panorámica rápida del requerimiento de datos locales de los OBs en el programa de usuario visualizado, puede apreciar lo siguiente en la estructura en árbol:

- el requerimiento máximo de datos locales por cada OB
- el requerimiento de datos locales por ruta

Dicha visualización se puede activar y desactivar en la ficha "Estructura del programa".

Para definir los datos locales requeridos por un bloque seleccionado, haga clic en la tecla derecha del ratón. Del menú contextual, elija el comando **Información del bloque**.

En caso de que haya OBs de errores síncronos (OB 121, OB 122), detrás del valor numérico del requerimiento máximo de datos locales aparecerá el signo "+" y el requerimiento adicional para los OBs de errores síncronos.

## Estructura por padres e hijos

En la estructura por padres e hijos se representan todos los bloques invocantes que contiene el programa de usuario S7 junto con los bloques invocados.

## Visualización de los bloques borrados

Las líneas de los bloques borrados se destacan en rojo y detrás de los bloques se visualiza la cadena de caracteres "?????".

### 12.1.4 Plano de ocupación para entradas, salidas y marcas (E/A/M)

Los planos de ocupación muestran los operandos ya ocupados en el programa de usuario. Constituye una base importante para buscar errores o para efectuar cambios en el programa de usuario.

Al visualizar el plano de ocupación se obtiene una relación de los distintos bits y sus respectivas áreas de memoria Entradas (E), Salidas (A) o Marcas (M). El plano de ocupación se visualiza en una ventana de trabajo. La barra del título de la ventana de trabajo muestra el nombre del programa de usuario S7 al que pertenece el plano de ocupación.

Cada línea contiene un byte del área de memoria en el cual están identificados los ocho bits según el acceso. Igualmente se indica si el acceso se realiza por un acceso a byte, a palabra o a palabra doble.

#### Signos utilizados en el plano de ocupación E/A/M:

- . no se ha accedido al operando y, por tanto, no está ocupado
- o el operando se utiliza directamente
- x el operando es ejecutado directamente (acceso a byte, palabra o palabra doble)

### Columnas en el plano de ocupación E/A/M

Columna	Contenido / significado
7	Número de bit del byte correspondiente
6	
5	
4	
3	
2	
1	
0	
B	El byte está ocupado por un acceso a byte
W	El byte está ocupado por un acceso a palabra
D	El byte está ocupado por un acceso a palabra doble

### Ejemplo de un plano de ocupación (E/A/M)

El ejemplo siguiente muestra la estructura característica del plano de ocupación para entradas, salidas y marcas (E/A/M).

Operando	7	6	5	4	3	2	1	0		B	W	D
AB 0	O	X	X	O	X	X	X	X		O	.	.
AB 1	.	O	.	.	O	.	O	.		.	.	.
EB 0	O	O	O	.	O	.	O	.		.	.	.
EB 1	.	.	.	.	.	.	.	.		.	.	.
MB 0	X	X	X	X	X	X	X	X		.	O	.
MB 1	X	X	X	X	X	X	O	X		.	.	.

En la primera línea se enumera la ocupación del byte de salida AB 0. A este operando se accede mediante un acceso a byte. No obstante, se efectúa simultáneamente un acceso a los bits de salida A 0.4 y A 0.7. Por tanto, en las columnas "4" y "7" aparece una "O". En cada una de las columnas "0", "1", "2", "3", "5" y "6" aparece una "X" como identificador del acceso a byte. En la columna "B" aparece una "O", puesto que al operando AB 0 se accede mediante un acceso a byte.

### 12.1.5 Plano de ocupación para temporizadores y contadores (T/Z)

Los planos de ocupación muestran los operandos ya ocupados en el programa de usuario. Constituye una base importante para buscar errores o para efectuar cambios en el programa de usuario.

Visualizando el plano de ocupación T/Z se obtiene una panorámica de los temporizadores (T) y contadores (Z) que se han utilizado.

El plano de ocupación se visualiza en una ventana de trabajo. La barra del título de la ventana de trabajo muestra el nombre del programa de usuario S7 al que pertenece el plano de ocupación. En cada línea se representan 10 temporizadores y/o contadores.

#### Signos utilizados en el plano de ocupación T/Z:

- . no utilizado
- x utilizado

#### Ejemplo de un plano de ocupación (T/Z)

	0	1	2	3	4	5	6	7	8	9
T 00-09	.	X	.	.	.	.	X	.	.	.
T 10-19	.	.	X	.	.	.	.	X	.	X
T 20-29	.	.	.	.	X	.	.	.	.	.
Z 00-09	.	.	X	.	.	.	.	X	.	.
Z 10-19	.	.	.	.	.	.	.	.	.	X
Z 20-29	.	.	.	.	.	.	.	.	.	.
Z 30-39	.	.	.	.	X	.	.	.	.	.

En este ejemplo están ocupados los temporizadores T1, T6, T12, T17, T19, T24 y los contadores Z2, Z7, Z19, Z34.

Las listas están ordenadas alfabéticamente. Las entradas se pueden ordenar haciendo clic en el encabezado de la columna en cuestión.

### 12.1.6 Símbolos no utilizados

Esta función ofrece una panorámica de todos los símbolos que tengan las siguientes propiedades:

- Los símbolos están definidos en la tabla de símbolos.
- No obstante, no se utilizan en las partes del programa de usuario para las que existen datos de referencia.

La lista se visualiza en una ventana de trabajo. En la barra del título de esta ventana se muestra el nombre del programa de usuario al que pertenece la lista.

Cada línea visualizada en la ventana corresponde a una entrada de la lista. En una línea se incluyen el operando, el símbolo, el tipo de datos y el comentario.

Columna	Contenido / Significado
Símbolo	Designación
Operando	Dirección absoluta del operando
Tipo de datos	Tipo de datos del operando
Comentario	Comentario del operando que se indica en la tabla de símbolos

### Ejemplo de una lista de símbolos no utilizados

Símbolo	Operando	Tipo de datos	Comentario
MS1	E103.6	BOOL	guardamotor 1
MS2	E120.5	BOOL	guardamotor 2
MS3	E121.3	BOOL	guardamotor 3

Las entradas se pueden ordenar haciendo clic en el encabezado de la columna en cuestión.

### 12.1.7 Operandos sin símbolo

Al visualizar la lista de operandos sin símbolo obtiene una lista de aquellos operandos que, si bien se están utilizando en el programa de usuario S7, aún no se han definido en la tabla de símbolos, es decir, no se les ha dado un nombre simbólico. La lista se visualiza en una ventana de trabajo. En la barra del título de esta ventana se muestra el nombre del programa de usuario al que pertenece la lista.

Una línea incluye el operando y el número de veces que ha sido utilizado dicho operando.

#### Ejemplo:

Operando	Cantidad
A 2.5	4
E 23.6	3
M 34.1	20

La lista está ordenada por operandos.

### 12.1.8 Visualizar la información del bloque en KOP, FUP, AWL

La información sobre el bloque en KOP, FUP y AWL - que comprende el lenguaje del bloque y los detalles.- se visualiza en la lista de referencias cruzadas y en la estructura del programa.

En la vista "Estructura del programa" se puede llamar la información sobre el bloque eligiendo el comando de menú **Ver > Información sobre el bloque** o bien pulsando la tecla derecha del ratón. La elección dependerá de si en los ajustes del filtro de la ficha "Estructura del programa" se escogió la visualización "Estructura por padres e hijos" o "Estructura en árbol".

En la vista "Referencias cruzadas" se mostrará la información sobre el bloque al elegir el comando **Ver > Filtrar**.

- Active la casilla de verificación "Información sobre el bloque" en la ficha "Referencias cruzadas" del cuadro de diálogo "Filtrar" para visualizar la información sobre el bloque.

Dicha información, en la cual aparecen abreviaturas, varía según el lenguaje en el que se haya programado el bloque.

Lenguaje	Segmento	Instrucción	Operación
AWL	Seg	Ins	/
KOP	Seg		
FUP	Seg		

**Seg e Ins** indican en qué segmento y en qué instrucción se utiliza el operando (referencias cruzadas) o se llama el bloque (estructura del programa).

### Mostrar la información sobre el bloque en lenguajes de programación opcionales

La Ayuda en pantalla sobre la información sobre el bloque se puede llamar al estar instalado el correspondiente software opcional.

## 12.2 Trabajar con datos de referencia

### 12.2.1 Métodos de visualización de los datos de referencia

Para visualizar los datos de referencia puede elegir una de las siguientes alternativas:

#### Visualización en el Administrador SIMATIC:

1. Seleccione en la ventana de proyecto en modo de visualización "offline" la carpeta "Bloques".
2. Elija el comando **Herramientas > Datos de referencia > Mostrar**.

#### Visualización en la ventana del editor:

1. En la carpeta "Bloques", abra un bloque.
2. En la ventana del correspondiente editor, elija el comando de menú **Herramientas > Datos de referencia**.

Aparece el diálogo "Preferencias". Aquí puede elegir la vista que deba aparecer siempre. La vista predeterminada es la última vista que se utilizó en la aplicación para visualizar los datos de referencia. Este cuadro de diálogo se puede desactivar para próximas sesiones.

Entonces se arranca la aplicación Mostrar datos de referencia y aparece la lista de referencias cruzadas del programa de usuario seleccionado (ajuste estándar al visualizar los datos de referencia por primera vez). Si los datos de referencia están incompletos aparecerá un cuadro de diálogo desde donde podrá actualizarlos.

#### Visualización directa desde el bloque compilado

Los datos de referencia de un bloque compilado se pueden visualizar directamente con el editor, obteniendo así una panorámica actual del programa de usuario.

## 12.2.2 Visualizar listas en ventanas de trabajo adicionales

Con el comando de menú **Ventana > Nueva ventana** se pueden abrir ventanas adicionales y visualizar allí otras listas (p.ej. lista de los símbolos no utilizados) de los datos de referencia.

Con el comando de menú **Datos de referencia > Abrir** se abre una ventana de trabajo que contiene los datos de referencia aún no visualizados.

Mediante el menú **Ver** o con el correspondiente botón de la barra de herramientas se puede cambiar a una representación diferente de los datos de referencia:

Representación de los datos de referencia	Comando de menú para visualizar dicha representación
Operandos sin símbolo	Ver > Operandos sin símbolo
Símbolos no utilizados	Ver > Símbolos no utilizados
Ocupación E/A/M	Ver > Ocupación > Entradas, salidas y marcas
Ocupación T/Z	Ver > Ocupación > Temporizadores y contadores
Estructura del programa.	Ver > Estructura del programa
Referencias cruzadas	Ver > Referencias cruzadas

## 12.2.3 Crear y visualizar datos de referencia

### Crear datos de referencia:

1. En el Administrador SIMATIC, seleccione la carpeta de bloques para la que se deben generar los datos de referencia.
2. En el Administrador SIMATIC, elija el comando **Herramientas > Datos de referencia > Generar**.

Antes de generar los datos de referencia se comprueba si existen ya algunos y si éstos son actuales.

- Los datos de referencia se generarán si no existen ya.
- Si no existen datos de referencia actuales, podrá decidir en un cuadro de diálogo si dichos datos se deben actualizar, o bien generar nuevamente.

### Mostrar datos de referencia:

Con el comando **Herramientas > Datos de referencia > Mostrar** se pueden visualizar los datos de referencia.

Antes de visualizar los datos de referencia se comprueba si existen ya y si son actuales.

- Los datos de referencia se crearán si no existen ya.
- Si los datos de referencia están incompletos se visualizará un cuadro de diálogo donde se indicará que son incoherentes. Podrá decidir entonces si se deben actualizar y en qué medida desea hacerlo. A tal efecto puede elegir una de las siguientes opciones:

Elección	Significado
sólo para los bloques modificados	Se actualizan los datos de referencia de los bloques modificados y de los bloques nuevos, eliminándose las informaciones sobre los bloques que se hayan borrado.
para todos los bloques	Los datos de referencia se crean nuevamente para todos los bloques.
no actualizar	Los datos de referencia no se actualizan.

Para poder actualizar los datos de referencia se compilan de nuevo los bloques utilizando un compilador adecuado para cada bloque. Con el comando **Ver > Actualizar** se pueden actualizar en la ventana activa los datos de referencia ya visualizados.

#### 12.2.4 Posicionamiento rápido en los puntos de aplicación del programa

Al programar utilice los datos de referencia para posicionarse en los puntos de aplicación de un operando. Para ello tiene que disponer de datos de referencia actuales. La llamada de la aplicación para mostrar los datos de referencia no es necesaria.

##### Procedimiento básico

1. Elija en el Administrador SIMATIC el comando de menú **Herramientas > Datos de referencia > Generar** para generar los datos de referencia actuales. Este paso solamente es necesario si no dispone de datos de referencia o si no son actuales.
2. Seleccione el operando en un bloque abierto.
3. Elija el comando de menú **Edición > Ir a > Punto de aplicación**. Aparece un cuadro de diálogo que contiene una lista con los puntos de aplicación del operando en el programa.
4. Elija la opción "Acceso solapado a las áreas de memoria", si además desea hacerse mostrar los puntos de aplicación de los operando cuyas direcciones o áreas de memoria se solapen con la del operando llamado. La tabla contendrá asimismo la columna "Operando".
5. Seleccione un punto de aplicación de la lista y haga clic en el botón de comando "Ir a".

Si al llamar el cuadro de diálogo los datos de referencia no son actuales, aparecerá un mensaje indicándoselo. En este caso puede hacer actualizar los datos de referencia.

## Lista de los puntos de aplicación

La lista de los puntos de aplicación en el cuadro de diálogo contiene los siguientes datos:

- Bloque en el que se utiliza el operando
- Símbolo del bloque, si es que existe
- Detalles, es decir, informaciones sobre el punto de aplicación y la operación en lo referente al lenguaje del bloque/de la fuente (SCL)
- Informaciones sobre el bloque y lenguaje
- Tipo de acceso al operando: lectura (R), escritura (W), lectura y escritura (RW), no determinable (?).
- Lenguaje del bloque

Si lo desea puede filtrar la visualización de los puntos de aplicación, y p. ej. visualizar así sólo los accesos de escritura de un operando. Para más información sobre las opciones de entrada y visualización consulte la Ayuda en pantalla de este cuadro de diálogo.

---

### Nota

Los datos de referencia solamente están disponibles online. Esta función opera siempre con las referencias cruzadas de los bloques offline, incluso si llama la función en un bloque online.

---

## 12.2.5 Ejemplo para trabajar con puntos de aplicación

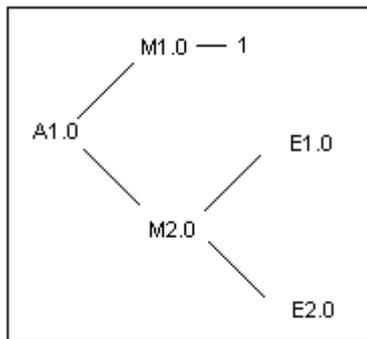
Supongamos que desea averiguar en qué puntos del programa se activa la salida A1.0 (directo/indirecto). Como ejemplo se utiliza el siguiente código AWL en el bloque OB1:

```
Segmento1: .....  
U A 1.0 // en este ejemplo  
= A 1.1 // no relevante  
  
Segmento 2:  
U M1.0  
U M2.0  
= A 1.0 // Asignación  
  
Segmento3:  
//sólo línea de comentario  
SET  
= M1.0 // Asignación
```

```

Segmento 4:
U E 1.0
U E 2.0
= M2.0 // Asignación
    
```

Para A1.0 resulta el árbol de asignaciones siguiente:



Proceda ahora de la siguiente manera:

1. Posiciónese en el editor KOP/AWL/FUP OB 1 en A1.0 (Seg 1, Ins 1).
2. Con Edición > Ir a > Punto de aplicación o con la tecla derecha del ratón "Ir a punto de aplicación".  
En el cuadro de diálogo se visualizan todas las asignaciones a A1.0:

```

OB1    Cycle Execution    Seg 2  Ins 3  /=    W    AWL
OB1    Cycle Execution    Seg 1  Ins 1  /U    R    AWL
    
```

3. Saltar mediante "Ir a" del cuadro de diálogo de "Seg 2 Ins 3" al editor:

```

Segmento 2:
U M1.0
U M2.0
= A 1.0
    
```

4. Ahora hay que controlar tanto las asignaciones a M1.0 como a M2.0. Posiciónese entonces en el editor KOP/AWL/FUP primero en M1.0.
5. Con Edición > Ir a > Punto de aplicación o con la tecla derecha del ratón abra el cuadro de diálogo "Ir a punto de aplicación". Aquí se visualizan entre otras cosas todas las asignaciones a M1.0 :

```

OB1    Cycle Execution    Seg 3  Ins 2  /=    W    AWL
OB1    Cycle Execution    Seg 2  Ins 1  /U    R    AWL
    
```

6. Salte con el comando "Ir a" a "Seg 3 Ins 2" en el editor KOP/AWL/FUP.

7. En el segmento 3 del editor KOP/AWL/FUP se detecta que la ocupación de M1.0 no es interesante (por estar siempre a TRUE) y que en su lugar habría que comprobar la ocupación de M2.0.

**En las versiones de STEP 7 anteriores a la versión V5 había que volver a recorrer toda la cadena de asignaciones desde el principio. Los botones de comando ">>" y "<<" simplifican el procedimiento siguiente:**

8. O bien pase a primer plano el cuadro de diálogo que aún está abierto "Ir a punto de aplicación" o bien llame este cuadro de diálogo desde la posición actual del editor KOP/AWL/FUP.
9. Activar el botón de comando "<<" una o dos veces hasta que se visualicen todos los puntos de aplicación de A 1.0. El último punto "Seg 2 Ins 3" está seleccionado.
10. Salte con el comando "Ir a" (como bajo el punto 3) desde el diálogo de los puntos de aplicación "Seg 2 Ins 3" al editor:

```
Segmento 2:
U M1.0
U M2.0
= A 1.0
```

11. En el punto 4 y siguientes se ha controlado la asignación a M1.0. Ahora hay que controlar todas las asignaciones (directas/indirectas) a M2.0. Es decir, posicione en el editor sobre M2.0 y active la función "Ir a punto de aplicación": Se visualizarán entre otras cosas todas las asignaciones a M2.0:

```
OB1    Cycle Execution    Seg 4  Ins 3  /=    W    AWL
OB1    Cycle Execution    Seg 2  Ins 2  /U    R    AWL
```

12. Salte con el comando "Ir a" al "Seg 4 Ins 3" del editor KOP/AWL/FUP:

```
Segmento 4:
U E 1.0
U E 2.0
= M2.0
```

13. Ahora tiene que comprobar las asignaciones a E1.0 y E2.0. Esto ya no se explica en el presente ejemplo, pues el procedimiento a seguir es el mismo (punto 4 y siguientes)

Saltando del editor KOP/AWL/FUP al cuadro de diálogo "Puntos de aplicación" y vice versa puede determinar y analizar los puntos relevantes de su programa.



# 13 Comprobar coherencia del bloque y fecha y hora como propiedad del bloque

## 13.1 Comprobar coherencia de bloques

### Introducción

Si durante o después de crear el programa hay que adaptar o completar los interfaces o el código de bloques aislados, es posible que se originen conflictos de indicaciones de fecha y hora. Los conflictos de indicaciones de fecha y hora pueden causar a su vez incoherencias de bloques entre los bloques llamantes y llamados, o en los bloques de referencia y generar de esta forma un gran esfuerzo de corrección.

La función "Comprobar coherencia del bloque" le ahorrará gran parte de este trabajo de corrección. La función "Comprobar coherencia del bloque" se ocupa automáticamente de gran parte de todos los conflictos de indicaciones de fecha y hora y de las incoherencias de bloques. En aquellos bloques, en los que no se pudieron eliminar automáticamente las incoherencias de bloques, la función le guiará a las posiciones que se desean modificar en el editor correspondiente. Allí podrá realizar las modificaciones necesarias. Paso a paso se eliminarán todas las incoherencias y se compilarán los bloques.

### Requisitos

La comprobación de la coherencia de los bloques sólo es posible en los proyectos creados con la versión de STEP 7 V5.0, Servicepack 3, o posteriores. En los proyectos más antiguos primero habrá que compilar todo antes de poder iniciar la comprobación de la coherencia de los bloques (comando de menú **Programa > Compilar todo**).

En el caso de los bloques creados con un paquete opcional, éste deberá estar instalado para poder realizar la comprobación de coherencia.

## **Iniciar la comprobación de la coherencia de los bloques**

Al iniciar la comprobación de la coherencia de los bloques se comprueba la indicación de fecha y hora de los interfaces de los bloques, y los bloques en los que es posible que se produzcan incoherencias se marcan en la vista en árbol (árbol de interdependencias/árbol de referencias).

1. Seleccione el comando de menú **Programa > Compilar**  
STEP 7 detecta automáticamente el lenguaje en que se creó el bloque afectado y llama al editor correspondiente. Los conflictos de indicaciones de fecha y hora y las incoherencias de bloques se eliminan en la medida de lo posible y los bloques se compilan. Si no se ha podido eliminar automáticamente el conflicto de indicaciones de fecha y hora o la incoherencia en un bloque, se genera un mensaje de error en la ventana de resultados (para el procedimiento a seguir, ver el paso 2). Este proceso se ejecuta automáticamente en todos los bloques de la vista en árbol.
2. Si en el proceso de compilación no se han podido eliminar automáticamente todas las incoherencias de los bloques, los bloques correspondientes se marcarán en la ventana de resultados como mensaje de error. Posicione la entrada de error correspondiente con el ratón y seleccione con la tecla derecha del ratón el comando del menú contextual **Mostrar errores**. El editor correspondiente se abre y se salta a las posiciones que hay que modificar. Elimine todas las incoherencias del bloque y cierre y guarde el bloque. Repita el proceso para todos los bloques marcados como errores.
3. Inicie de nuevo los pasos 1 y 2. Repita este proceso hasta que en la ventana de mensajes ya no queden errores.

## 13.2 Fecha y hora como propiedad del bloque y conflictos de fecha y hora

Los bloques contienen dos indicaciones de fecha y hora: una de la lógica o código y otra del interface. Dichas indicaciones de fecha y hora se visualizan en el diálogo de las propiedades del bloque. Mediante las indicaciones de fecha y hora se vigila que los programas STEP 7 sean coherentes.

STEP 7 indica un conflicto de fecha y hora cuando al comparar las indicaciones de fecha y hora, se detecta una violación de las reglas.

Si no se respetan las reglas, pueden ocurrir los siguientes errores:

- El bloque llamado es más reciente que el bloque que lo llama (CALL).
- El bloque direccionado es más reciente que el bloque que lo utiliza.
- Ejemplos del último caso (segundo punto) son los siguientes:
- Un UDT es más reciente que el bloque que lo utiliza, p. ej. un DB u otro UDT o un FC, FB, OB que utiliza el UDT en la tabla de declaración de variables.
- Un FB es más reciente que el DB de instancia correspondiente.
- En el FB1 hay un FB2 definido como multiinstancia y el FB2 es más reciente que el FB1.

---

### Nota

Aunque la relación de las indicaciones de fecha y hora de los interfaces sea correcta pueden surgir conflictos:

- La definición del interface del bloque direccionado no se corresponde con el interface utilizado en su punto de aplicación.

Estos problemas se denominan conflictos de interfaces y pueden aparecer al copiar bloques de distintos programas o cuando al compilar una fuente ASCII sólo se genere una parte de los bloques del programa.

---

## **13.3 Indicación de fecha y hora de bloques lógicos**

### **Indicación de fecha y hora del código del programa**

Aquí se introduce la fecha y hora de creación del bloque. Dicha indicación se actualiza

- al cambiar la lógica del programa
- al cambiar la descripción del interface
- al cambiar el comentario
- al crearse por primera vez y al compilarse una fuente ASCII
- al cambiar las propiedades del bloque (cuadro de diálogo: Propiedades)

### **Indicación de fecha y hora del interface**

Esta indicación de fecha y hora se actualiza

- al cambiar la descripción del interface (cambio de tipos de datos o de valores iniciales, nuevos parámetros)
- al crearse por primera vez y al compilarse una fuente ASCII, en caso de que cambie la estructura del interface.

La indicación de fecha y hora no se actualiza

- al cambiar los símbolos
- al cambiar los comentarios en la declaración de variables
- al efectuarse cambios en el área TEMP

### **Reglas para las llamadas de bloques**

- La indicación de fecha y hora del interface del bloque llamado debe ser anterior a la indicación de fecha y hora de la lógica del bloque invocante.
- Cambie el interface de un bloque sólo cuando no esté abierto ningún otro bloque que lo llame. Si los bloques invocantes se guardan posteriormente al bloque modificado, no reconocerán la incoherencia en la indicación de fecha y hora.

### **Procedimiento al presentarse conflictos de fecha y hora**

Al abrir el bloque invocante se visualiza un conflicto de fecha y hora . Después de modificarse un interface de una FC o de un FB, todas las llamadas a dichos bloques se representarán de forma expandida en los bloques invocantes.

Si se cambia el interface de un bloque, será preciso adaptar todos los bloques que lo llamen.

Después de cambiar un interface FB, se deberán actualizar todas las definiciones existentes de multiinstancias, así como los bloques de datos de instancia.

## 13.4 Indicación de fecha y hora de bloques de datos globales

### Indicación de fecha y hora de la lógica

Esta indicación de fecha y hora se actualiza

- al crear por primera vez,
- al compilar una fuente ASCII,
- al efectuarse cambios en la vista "Declaración" o en la vista "Datos" del bloque.

### Indicación de fecha y hora del interface

Esta indicación se actualiza

- al efectuarse cambios de la descripción del interface en la vista "Declaración" (cambio de tipos de datos o de valores iniciales, nuevos parámetros)

## 13.5 Indicación de fecha y hora de bloques de datos de instancia

Los bloques de datos de instancia permiten almacenar los parámetros formales y los datos estáticos de los bloques de función.

### Indicación de fecha y hora de la lógica

Aquí se introduce la fecha y hora de creación del bloque de datos de instancia. Dicha indicación se actualiza al introducirse valores actuales en la vista "Datos" del bloque de datos de instancia. El usuario no puede cambiar la estructura de un bloque de datos de instancia, puesto que dicha estructura se deriva del bloque de función (FB) asignado o del bloque de función de sistema (SFB).

### Indicación de fecha y hora del interface

Al crearse un bloque de datos de instancia se introduce la indicación de fecha y hora del FB o SFB asignado.

### Reglas para abrir bloques sin que se presenten conflictos

Deben concordar las indicaciones de fecha y hora del interface del FB o SFB y del bloque de datos de instancia asignado.

### Procedimiento al presentarse conflictos de fecha y hora

Al cambiarse el interface de un FB se actualiza la indicación de fecha y hora del interface del FB. Al abrirse un bloque de datos de instancia asignado se visualizará un conflicto de fecha y hora, puesto que la indicación de fecha y hora del bloque de datos de instancia ya no concuerda con la del FB. En el área de declaración del DB se representará el interface con los símbolos generados por el compilador (pseudosímbolos). En este caso, el bloque de datos de instancia sólo se podrá visualizar.

Para solucionar este conflicto es preciso volver a crear un DB de instancia asignado al FB modificado.

## **13.6 Indicación de fecha y hora de UDTs y DBs derivados de UDTs**

Por ejemplo, un tipo de datos definido por el usuario (UDT) se puede utilizar para generar varios bloques de datos de la misma estructura.

### **Indicación de fecha y hora de la lógica**

Esta indicación se actualiza cada vez que se realice un cambio.

### **Indicación de fecha y hora del interface**

Esta indicación se actualiza al cambiar la descripción del interface (cambio de tipos de datos o de valores iniciales, nuevos parámetros).

La indicación de fecha y hora de un UDT se actualiza también al compilarse la fuente ASCII.

### **Reglas para abrir bloques sin que se presenten conflictos**

- La indicación de fecha y hora del interface del tipo de datos de usuario (UDT) debe ser anterior a la de los bloques lógicos donde se utiliza dicho UDT.
- La indicación de fecha y hora del interface del tipo de datos de usuario (UDT) debe ser idéntica a la de un DB derivado de un UDT.
- La indicación de fecha y hora del interface del tipo de datos de usuario (UDT) debe ser anterior a la de un UDT subordinado.

### **Procedimiento al presentarse conflictos de fecha y hora**

Si se modifica una definición de un UDT utilizada en un DB, una FC, un FB o en una definición de UDT diferente, STEP 7 visualizará un conflicto de fecha y hora al abrirse dicho bloque.

El componente UDT se representa en forma de estructura. Todos los nombres de variables se sobrescriben con valores predeterminados por el sistema.

# 14 Configurar mensajes

## 14.1 El sistema de mensajes

### 14.1.1 El sistema de notificación

Los mensajes permiten detectar, localizar y eliminar rápidamente los fallos que se presenten en los sistemas de automatización durante el proceso. Así se pueden reducir considerablemente los tiempos de parada de una instalación.

Antes de poder visualizar los mensajes es preciso configurarlos.

STEP 7 permite crear, editar y compilar mensajes de eventos con sus textos y atributos en visualizadores.

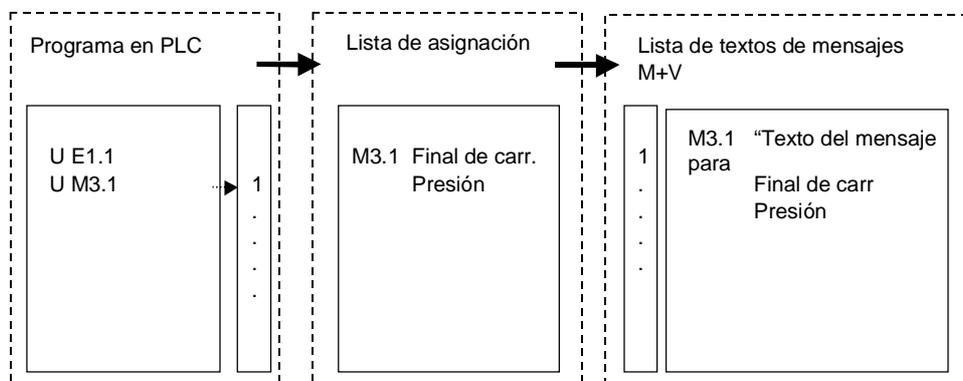
### 14.1.2 ¿Qué procedimientos existen para crear mensajes?

Para crear mensajes se dispone de distintos procedimientos.

#### Procedimiento de notificación por bits

El procedimiento de notificación por bits requiere 3 pasos por parte del programador:

- Crear el programa de usuario en la PG y activar el bit deseado.
- Crear con un editor de texto cualquiera una lista de asignación, en la que se asigna un texto a cada bit de notificación (p.ej., M 3.1 = Final de carrera Presión).
- Crear la lista de los textos de los mensajes en el sistema de manejo a partir de la lista de asignación.

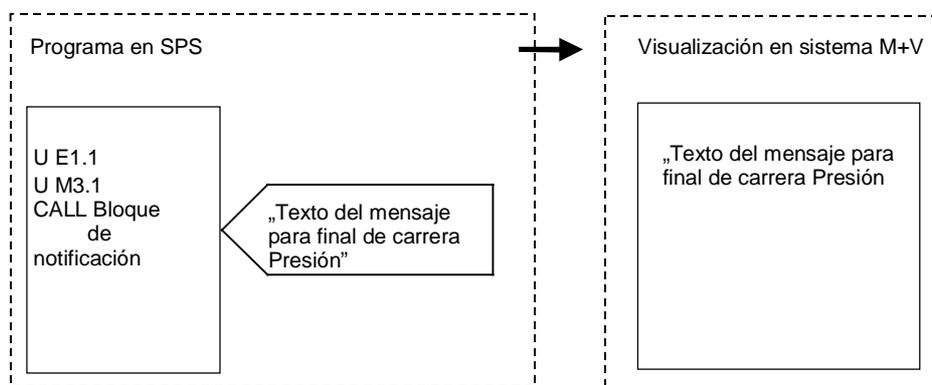


El sistema de manejo y visualización consulta cíclicamente el sistema de automatización para saber si se ha modificado o no el bit de notificación. Cuando el PLC notifica una modificación, se visualiza el mensaje correspondiente. Entonces, se asigna al mensaje la indicación de hora y fecha del sistema de manejo y visualización.

### Procedimiento por números de mensajes

El procedimiento por números de mensajes sólo requiere un paso por parte del programador:

- Crear el programa de usuario en la PG, activar el bit deseado y asignar directamente al programar el texto deseado al bit.



El sistema de automatización no se consulta cíclicamente. Tan pronto como el PLC notifique una modificación, se notificará el número del mensaje correspondiente al sistema de manejo y visualización y se visualizará el texto correspondiente. La indicación de hora y fecha del mensaje es la misma que la del PLC, por lo que se puede determinar de forma mucho más exacta que en el caso del procedimiento de notificación por bits.

### 14.1.3 Selección del procedimiento de notificación

#### Vista general

La tabla siguiente muestra las propiedades y requisitos de los distintos procedimientos de notificación.

Procedimiento por números de mensajes	Procedimiento de notificación por bits
<ul style="list-style-type: none"> <li>• Los mensajes se gestionan en una base de datos común para la PG y el sistema de manejo</li> <li>• La carga del bus es mínima (el PLC está activo)</li> <li>• Los mensajes reciben la indicación de fecha y hora del sistema de automatización</li> </ul>	<ul style="list-style-type: none"> <li>• No hay ninguna base de datos común para la PG y el sistema de manejo</li> <li>• La carga del bus es alta (el sistema de manejo sondea)</li> <li>• Los mensajes reciben la indicación de fecha y hora del sistema de manejo</li> </ul>

El procedimiento por números de mensajes distingue los tres tipos siguientes de mensajes:

Mensajes de bloques	Mensajes de símbolos	Mensajes de diagnóstico personalizados
<ul style="list-style-type: none"> <li>• Síncronos con el programa</li> <li>• Visualización mediante WinCC y ProTool (sólo ALARM_S)</li> <li>• Posible en S7-300/S7-400</li> <li>• Programación con bloques de notificación:</li> <li>• ALARM</li> <li>• ALARM_8</li> <li>• ALARM_8P</li> <li>• NOTIFY</li> <li>• ALARM_S(Q)</li> <li>• AR_SEND</li> <li>• Transferencia al sistema de manejo</li> <li>• Para WinCC a través de la configuración de enlaces PLC-OS</li> <li>• Para ProTool a través de las funciones ProTool</li> </ul>	<ul style="list-style-type: none"> <li>• Asíncrono con el programa</li> <li>• Visualización mediante WinCC</li> <li>• Posible sólo en S7-400</li> <li>• Configuración mediante la tabla de símbolos</li> <li>• Transferencia al PLC a través de los bloques de datos del sistema (SDBs)</li> <li>• Transferencia al sistema de manejo vía configuración de enlaces PLC-OS</li> </ul>	<ul style="list-style-type: none"> <li>• Asíncrono con el programa</li> <li>• Visualización en el búfer de diagnóstico de la PG</li> <li>• Posible en S7-300/S7-400</li> <li>• Programación con bloques de notificación (función del sistema)</li> <li>• WR_USMSG</li> <li>• Sin transferencia al sistema de manejo</li> </ul>

STEP 7 soporta el procedimiento más cómodo por números de mensajes, el cual se describe detalladamente a continuación.

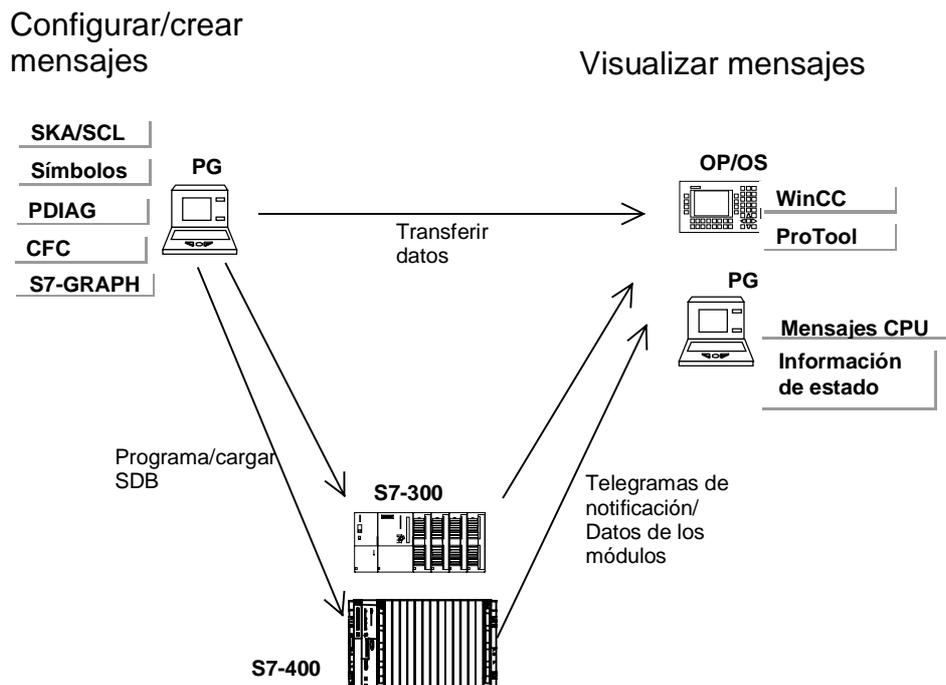
### Ejemplos del procedimiento por números de mensajes

Procedimiento de notificación	Campo de aplicación
Mensajes de bloque	Para notificar eventos síncronos con el programa; p.ej., se debe indicar que un regulador ha alcanzado un valor límite
Mensajes de símbolos	Para notificar eventos que no dependen del programa; p.ej., se debe vigilar una posición del interruptor
Mensajes personalizados	Para notificar eventos de diagnóstico en el búfer de diagnóstico, en cada llamada de la SFC

## 14.1.4 Componentes SIMATIC

### Vista general

La figura siguiente muestra, en una vista general, los componentes SIMATIC que participan en la configuración y visualización de mensajes.



### 14.1.5 Partes de un mensaje

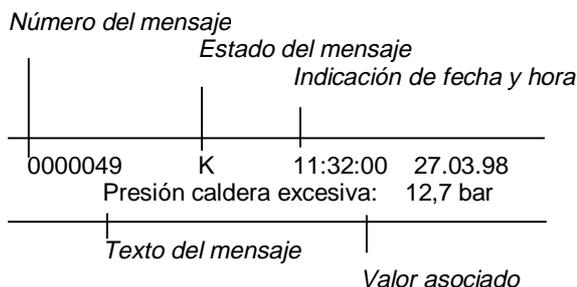
El modo de visualización de un mensaje depende del procedimiento de notificación, del bloque de notificación utilizado y del visualizador.

En la tabla siguiente se listan las partes que pueden integrar un mensaje:

Parte	Descripción
Indicación de fecha y hora	Se genera cuando aparece el evento de notificación en el sistema de automatización
Estado del mensaje	Estados posibles: entrante, saliente, saliente sin acuse y saliente con acuse
Valor asociado	A algunos mensajes se les puede asociar un valor del proceso, el cual puede ser evaluado por el bloque de notificación
Imagen	En caso de una caída del sistema, los mensajes que habían aparecido pueden visualizarse posteriormente en el OS.
Número del mensaje	Número asignado por el sistema, unívoco para todo el proyecto, que identifica un mensaje.
Texto del mensaje	Configurado por el usuario

### Ejemplo

El ejemplo siguiente muestra un mensaje de fallo en un panel de operador.



### 14.1.6 Asignar números a los mensajes

Los mensajes se identifican con un número unívoco para todo el proyecto. Para ello a cada programa S7 se le asigna un margen numérico dentro del margen numérico general disponible (1 – 2097151). Si aparece un conflicto al copiar un programa, es decir, si los números de mensaje de este programa ya han sido asignados en el área de destino, deberá asignarse otro margen numérico al nuevo programa. En este caso, STEP 7 abre automáticamente el cuadro de diálogo en el que se puede indicar el nuevo margen numérico.

El margen numérico de un programa S7 también se puede definir o modificar eligiendo el comando de menú **Edición > Propiedades especiales del objeto > Números de mensaje**. Sin embargo esto sólo es posible si aún no se han configurado mensajes para este programa.

Los márgenes numéricos se asignan de forma estándar en intervalos de 20.000.

## 14.2 Asignar y editar mensajes de bloque

### 14.2.1 Asignar y editar mensajes de bloque

Los mensajes de bloque se asignan a un bloque (FB). Para crear un mensaje de bloque se pueden utilizar bloques de funciones del sistema (SFB) y funciones del sistema (SFC).

### 14.2.2 ¿Qué bloques de notificación existen?

Es posible elegir entre los siguientes bloques de notificación en los que ya está programada la correspondiente función:

- SFB 33: "ALARM"
- SFB 34: "ALARM\_8",
- SFB 35 "ALARM\_8P",
- SFB 36 "NOTIFY",
- SFC 18: "ALARM\_S" y SFC 17: "ALARM\_SQ", así como
- SFB 37: "AR\_SEND" (para enviar archivadores).

Para obtener información más detallada consulte la ayuda de referencia sobre los Bloques .

### ¿Cuándo utilizar qué bloque de notificación?

La tabla siguiente facilita la elección del bloque de notificación a utilizar. Dicha elección depende de:

- el número de canales disponibles en el bloque y, por lo tanto, del número de señales que se vigilan en cada llamada del bloque
- la posibilidad de confirmar mensajes
- la posibilidad de adjuntar valores asociados
- los visualizadores que se desean utilizar
- el alcance de la CPU.

Bloque de notificación	Canales	Acuse	Valores asociados	Visualización WinCC	Visualización PROTOOL	Visualización mensajes de CPU/S7 Status	PLC	Particularidades
ALARM SFB 33	1	posible	máx. 10	sí	no	no	S7-400	Envía un mensaje por flanco entrante o saliente
ALARM_8 SFB 34	8	posible	no	sí	no	no	S7-400	Envía un mensaje por flanco entrante o saliente de una o varias señales
ALARM_8P SFB 35	8	posible	máx. 10	sí	no	no	S7-400	Como en ALARM_8
NOTIFY SFB 36	1	no	máx. 10	sí	no	no	S7-400	Como en ALARM
AR_SEND SFB 37	1	-	-	sí	no	no	S7-400	sirve para enviar un archivador
ALARM_SQ SFC 17	1	posible	1	sí	sí*	sí	S7-300/400	No es cada cambio de flanco el que genera un mensaje, sino cada llamada de SFC
ALARM_S SFC 18	1	no	1	sí	sí*	sí	S7-300/400	Como en ALARM_SQ

\* depende del tipo de OP

### 14.2.3 Parámetros formales, atributos del sistema y bloques de notificación

#### Parámetro formal como entrada del número de mensaje

Para cada mensaje o grupo de mensajes, en su programa necesita un parámetro formal que pueda utilizar como variable de entrada en la tabla de declaración de variables de su programa. El parámetro formal se utiliza como entrada de número de mensaje y constituye la base de un mensaje.

#### Asignar atributos del sistema a los parámetros formales

Para poder acceder a la configuración de mensajes ha tenido que haber asignado atributos del sistema a los parámetros formales:

1. Ha asignado los siguientes atributos del sistema para parámetros: "S7\_server" y "S7\_a\_type"
2. Ha asignado a los atributos del sistema valores adecuados a los bloques de mensajes que usted ha llamado en su código de programa. El valor para "S7\_server" es siempre "alarm\_archiv", y el valor para "S7\_a\_type" corresponde al bloque de mensajes llamado.

## Atributos del sistema y bloques de mensajes correspondientes

Los objetos del Administrador de mensajes que se visualizan no son los propios bloques de mensajes, sino los valores correspondientes a los atributos del sistema "S7\_a\_type". Estos valores tienen los mismos identificadores que los bloques de mensajes existentes, que pueden ser SFB o SFC (excepción: "alarm\_s").

S7_a_type	Bloque de mens...	Identificador	Propiedades
alarm_8	ALARM_8	SFB 34	8 canales, posibilidad de acuse, sin valores asociados
alarm_8p	ALARM_8P	SFB 35	8 canales, posibilidad de acuse, hasta 10 valores asociados por canal
notify	NOTIFY	SFB 36	1 canal, sin acuse, hasta 10 valores asociados
alarm	ALARM	SFB 33	1 canal, posibilidad de acuse, hasta 10 valores asociados
alarm_s	ALARM_S	SFC 18	1 canal, sin acuse, hasta 1 valor asociado
alarm_s	ALARM_SQ	SFC 17	1 canal, posibilidad de acuse, hasta 1 valor asociado
ar_send	AR_SEND	SFB 37	sirve para enviar un archivador

Para obtener información más detallada consulte la ayuda de referencia acerca de los atributos del sistema

Los atributos del sistema se asignan automáticamente cuando los bloques de notificación utilizados en el programa de usuario son SFB o FB con los atributos de sistema correspondientes y se llaman en calidad de multiinstancias.

### 14.2.4 Plantilla y mensajes

Con la configuración de mensajes puede crear una plantilla o mensajes en distintas fases. El procedimiento depende del bloque apto para notificación a través del cual se accede a la configuración de mensajes.

#### El bloque apto para notificación puede ser un FB o un DB de instancia

- Si es un FB, puede crear una plantilla para otros mensajes. Todas las indicaciones que haga para la plantilla son aceptadas automáticamente por los mensajes. Si asigna un DB de instancia al FB, después de generarse la plantilla se generarán automáticamente mensajes para el DB de instancia y se les asignarán números de mensaje.
- Si se trata de un DB de instancia puede modificar para cada instancia los mensajes generados a partir de la plantilla.

La diferencia visible consiste en que en los mensajes ya se asignan números de mensaje, mientras que en una plantilla no.

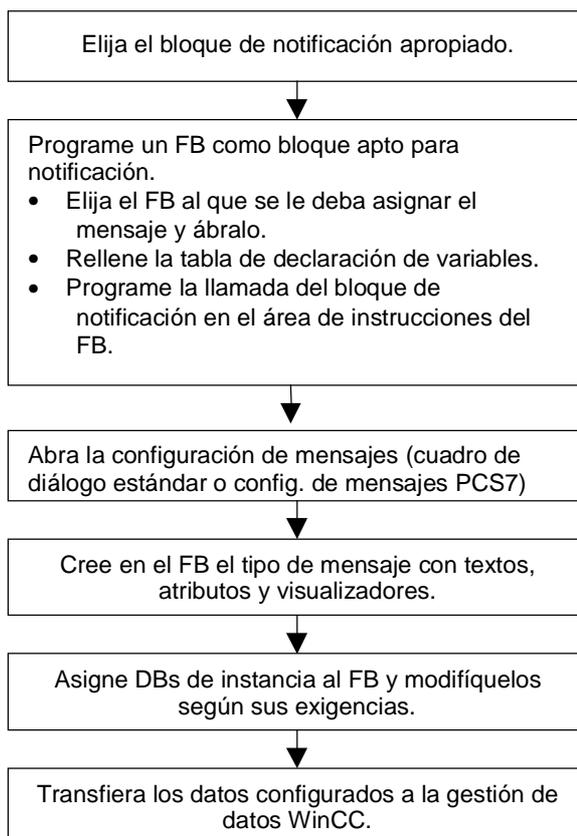
## Inhibir datos en la plantilla

Al configurar mensajes se indican textos y atributos de mensajes controlados por eventos. Además, se puede definir cómo se visualizarán los mensajes en determinados visualizadores. Para simplificar la creación de los mensajes se puede operar con plantillas.

- Al introducir los datos (atributos y textos) de la plantilla se puede determinar si deben inhibirse o no. Junto al cuadro de entrada de los atributos inhibidos aparece un símbolo con una llave. Los textos inhibidos reciben una marca de verificación en la columna "Inhibido".
- Los datos inhibidos en la plantilla no se podrán modificar en los mensajes específicos de las instancias. Únicamente se visualizarán.
- No obstante, si necesita efectuar modificaciones, debe retornar a la plantilla, cancelar la inhibición y efectuar allí la modificación. Sin embargo, las modificaciones así efectuadas no serán válidas para las instancias que se hayan generado anteriormente.

## 14.2.5 Crear mensajes de bloque

### Procedimiento general



## Programar un bloque apto para notificación (FB)

1. En el Administrador SIMATIC, elija el bloque de función (FB) para el que desea crear un mensaje y ábralo haciendo doble clic en él.

**Resultado:** El bloque seleccionado se abre y se visualiza en la ventana "KOP/AWL/FUP".

2. Rellene la tabla de declaración de variables. Para cada bloque de notificación llamado en el FB es necesario declarar variables en el FB invocante.

Para ello introduzca las siguientes variables en la columna "Declaración" de la tabla de declaración de variables:

- en el tipo de declaración "in" un nombre simbólico para la entrada del bloque de notificación, p.ej., "Not01" (para la entrada de notificación 01), y el tipo (tiene que ser "DWORD" sin valor inicial).
  - en el tipo de declaración "stat" un nombre simbólico para el bloque de notificación invocante, p.ej., "alarm", y el tipo correspondiente, aquí: "SFB33".
3. En el área de instrucciones del FB, introduzca la llamada del bloque de notificación seleccionado (en este caso "CALL alarma") y confirme pulsando la tecla INTRO.

**Resultado:** En el área de instrucciones del FB se visualizarán las variables de entrada del bloque llamado, es decir, del SFB 33 en este caso.

4. Asigne a la variable "EV\_ID" el nombre simbólico que ha definido en el paso 2 para la entrada del bloque de notificación, aquí: "Not01", y confirme la aceptación de los atributos de sistema para la configuración de mensajes.

**Resultado:** en la columna "Nombre", si no está seleccionada, aparecerá una "banderilla". El bloque seleccionado se preparará para notificar mensajes. Los atributos de sistema requeridos (p.ej., "S7\_server" y "S7\_a\_type") y sus valores correspondientes se asignan automáticamente.

5. Repita los pasos 2, 3 y 4 para todas las llamadas de bloques de notificación en este FB.
6. Guarde el bloque con el comando **Archivo > Guardar**.
7. Cierre la ventana "KOP/AWL/FUP".

## Abrir la configuración de mensajes

- Elija en el Administrador SIMATIC el comando de menú **Edición > Propiedades especiales del objeto > Mensaje**.

**Resultado:** se abrirá el cuadro de diálogo (estándar) para la configuración de mensajes en STEP 7. Si desea saber qué tiene que hacer para llamar la configuración de mensajes PCS7, consulte Configuración de mensajes PCS7.

## Crear una plantilla

1. Seleccione el bloque de notificación deseado e introduzca en las fichas "Atributos" y "Texto" los atributos y el texto deseado.  
Caso que haya seleccionado un bloque de notificación multicanal (p.ej., "ALARM\_8"), podrá asignar a cada subnúmero un texto de mensaje propio. Los atributos son válidos para todos los subnúmeros.
2. Asigne los visualizadores deseados a la plantilla. A tal efecto, haga clic en el botón "Nuevo visualizador". Aparecerá el cuadro de diálogo "Insertar visualizador", en el que podrá elegir los dispositivos deseados.

En las fichas posteriores, introduzca los textos y atributos deseados para los visualizadores. Salga del cuadro de diálogo haciendo clic en "Aceptar".

---

### Nota

Para editar los textos y atributos específicos de los visualizadores, tenga en cuenta la documentación suministrada con el dispositivo en cuestión.

---

## Crear DBs de instancia

1. Una vez creada una plantilla, puede asignarle bloques de datos de instancia (DBs) y editar los mensajes para dichos DBs por instancias.  
Abra en el Administrador SIMATIC el bloque que debe llamar el FB configurado previamente (p.ej.: el "OB1"), haciendo doble clic en él. En el área de instrucciones abierta en el OB, introduzca la llamada ("CALL"), así como el nombre y el número del FB que desea llamar y del DB que se ha de asignar al FB como instancia. Confirme sus entradas pulsando la tecla INTRO.

**Ejemplo:** Introduzca "CALL FB1, DB1". Si el DB1 no existe todavía, responda "Sí" cuando se le pregunte si desea generar el DB de instancia.

**Resultado:** se creará el DB de instancia. En el área de instrucciones del OB se visualizan las variables de entrada del FB correspondiente, aquí, p.ej., "Not01", y el número de mensaje asignado por el sistema, aquí: el "1".

2. Con el comando **Archivo > Guardar** guarde el OB y cierre la ventana "KOP/AWL/FUP".

## Editar mensajes

1. Seleccione en el Administrador SIMATIC el DB de instancia generado, p.ej., el "DB1", y llame la configuración de mensajes eligiendo el comando de menú **Edición > Propiedades especiales del objeto > Mensaje**.  
**Resultado:** Aparecerá el cuadro de diálogo "Configuración de mensajes", en el que se visualizan el DB de instancia seleccionado y el número de mensaje adjudicado por el sistema.
2. Introduzca en las fichas del cuadro de diálogo los cambios para el correspondiente DB de instancia y, en caso necesario, inserte más visualizadores. Abandone la función con "Aceptar".  
**Resultado:** la configuración de mensajes para el DB de instancia seleccionado ha concluido.

## Transferir los datos de configuración

Transfiera los datos configurados a la gestión de datos de WinCC (mediante la configuración de enlaces PLC-OS) o a la gestión de datos de ProTool.

### 14.2.6 Configuración de mensajes PCS7

Para editar las plantillas y mensajes que se deseen visualizar en WinCC, STEP 7 ofrece - con la configuración de mensajes PCS7- las siguientes posibilidades:

- configuración simplificada de visualizadores (creación automática)
- introducción simplificada de atributos y textos para mensajes
- homogeneización de mensajes garantizada

## Llamar la configuración de mensajes PCS7

1. Seleccione en el Administrador SIMATIC el bloque (FB o DB) cuyos textos de mensaje desea editar, y llame el cuadro de diálogo para la introducción de los atributos de sistema eligiendo el comando de menú **Edición > Propiedades del objeto**.
2. En la tabla que se visualiza entonces, introduzca uno de los siguientes atributos del sistema:
  - Atributo: "S7\_alarm\_ui" y valor: "1" para mensajes PCS 7 no controlados por eventos (el valor "0" desactiva la herramienta de configuración de mensajes PCS 7). Los parámetros de propiedades se pueden asignar en KOP/FUP/AWL. Los DBs que se generen y se asignen posteriormente al FB adoptan automáticamente el ajuste del atributo. Este ajuste se puede modificar en los ajustes del DB independientemente de la plantilla del mensaje (FB).
  - Atributo: "S7\_alarm" y valor "alarm\_16" (si el bloque seleccionado es un bloque de comunicación controlado por eventos del tipo EDC). Los parámetros de propiedades no se pueden asignar en la comunicación controlada por eventos. Los textos de los mensajes de los DBs que están asignados a este FB no se pueden editar en el Administrador SIMATIC.

---

### Nota

Al introducir los atributos del sistema se comprueba su sintaxis. Las entradas erróneas se destacan en color rojo.

---

3. Salga del cuadro de diálogo haciendo clic en "Aceptar".

4. Seleccione el comando de menú **Edición > Propiedades especiales del objeto > Mensaje**.  
**Resultado:** se abrirá el cuadro de diálogo para la configuración de mensajes PCS7.

### Editar plantillas

1. Seleccione en el Administrador SIMATIC el FB cuyos textos de mensaje desea editar y llame la configuración de mensajes PCS7.  
**Resultado:** en el cuadro de diálogo se visualizará una ficha para cada uno de los bloques de notificación para los que haya declarado una variable en el FB, y dos fichas para el bloque de comunicaciones controlado por eventos.
2. Rellene los cuadros de texto para las partes del mensaje "Proveniencia", "Area OS" e "Ident. batch".
3. Indique la categoría y el texto de todos los eventos de los bloques de notificación utilizados y decida si se debe acusar cada evento por separado.
4. Haga clic en la marca de verificación "Inhibido" para aquellas partes del mensaje que son válidas para todas las instancias y no se deben modificar.

### Editar mensajes

1. Seleccione en el Administrador SIMATIC el DB de instancia cuyos textos de mensaje desea editar y llame la configuración de mensajes PCS7.
2. Modifique las partes específicas de cada instancia que no han sido inhibidas.

---

#### Nota

Los textos de mensajes de DBs de instancia asignados a bloques de comunicaciones controlados por eventos sólo puede editarse en CFC.

---

## 14.3 Asignar y editar mensajes de símbolos

### 14.3.1 Asignar y editar mensajes de símbolos

Los mensajes de símbolos (SCAN) se asignan directamente a una señal en la tabla de símbolos. Se pueden utilizar todos los operandos booleanos: es decir, entradas (E), salidas (A) y marcas (M). Al configurar los mensajes es posible asignar diversos atributos, textos y hasta 10 valores asociados a dichas señales. Para facilitar la elección de señales de la tabla de símbolos se pueden ajustar filtros.

Con un mensaje de símbolo se puede escrutar una señal en un período predeterminado para averiguar si se ha presentado un cambio.

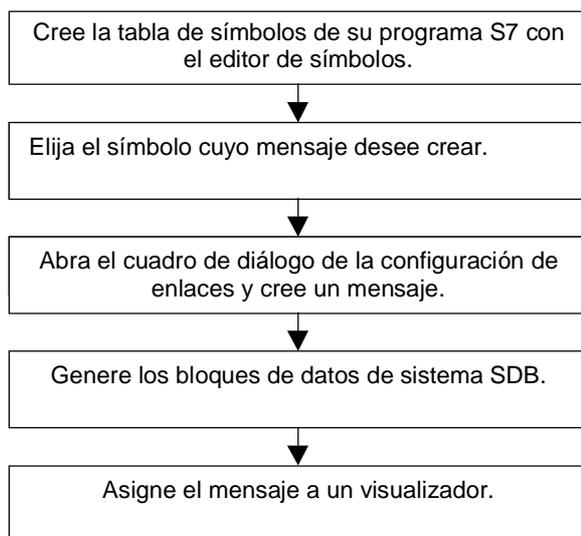
---

#### Nota

La duración del período depende de la CPU utilizada.

---

#### Procedimiento general



Durante la ejecución del programa se comprueban asincrónicamente las señales para las cuales se hayan configurado los mensajes. Las comprobaciones se llevan a cabo en las bases de tiempo configuradas. Los mensajes se visualizarán en los visualizadores asignados.

## 14.4 Crear y editar mensajes de diagnóstico personalizados

### 14.4.1 Crear y editar mensajes de diagnóstico personalizados

Esta función permite escribir una entrada en el búfer de diagnóstico y emitir el mensaje correspondiente. Los mensajes de diagnóstico personalizados se realizan con la función del sistema SFC 52 (WR\_USMSG) que se utiliza como bloque de notificación. La llamada de la SFC 52 se debe insertar en el programa de usuario, adjudicándole además el ID del evento.

A diferencia de los mensajes de bloques o de símbolos, los mensajes de diagnóstico personalizados sólo se pueden visualizar en una unidad de programación (PG). Por lo tanto, al configurar los mensajes no es posible asignar ningún otro visualizador.

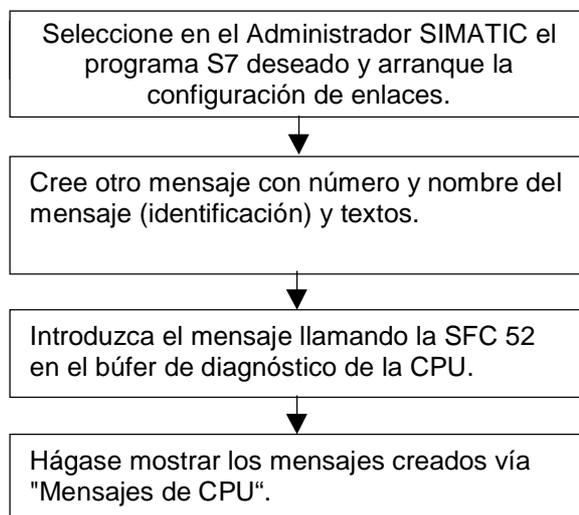
#### Requisitos

Para poder generar un mensaje de diagnóstico personalizado es necesario haber creado previamente:

- un proyecto en el Administrador SIMATIC y
- en el proyecto, el programa S7/M7 al que se desea asignar el mensaje.

#### Procedimiento general

Para crear y visualizar mensajes de diagnóstico personalizados proceda de la siguiente forma:



## 14.5 Traducir y editar textos de usuario

### 14.5.1 Traducir y editar textos de usuario

Los textos que aparecen en los visualizadores al ejecutar el proceso se suelen mostrar en el idioma en el que se programó la solución de automatización.

Sin embargo, a menudo ocurre que el operador que debe reaccionar a estos mensajes no entiende este idioma, por lo que necesita los textos en su lengua. Sólo de este modo se garantiza una rápida reacción y una respuesta adecuada a los mensajes que aparezcan.

STEP 7 ofrece la posibilidad de traducir todos los textos de usuario en el idioma deseado. Para ello tiene que haberse instalado el idioma en cuestión en el proyecto (comando de menú del Administrador SIMATIC: **Herramientas > Idioma para visualizador**). El número de idiomas disponibles se define al instalar Windows (propiedad del sistema).

De esta forma se garantiza que los operadores puedan visualizar los mensajes en el idioma que deseen, optimizando así la seguridad del proceso.

Son textos de usuario los textos de usuario y las librerías de textos.

### 14.5.2 Traducir y editar textos de usuario

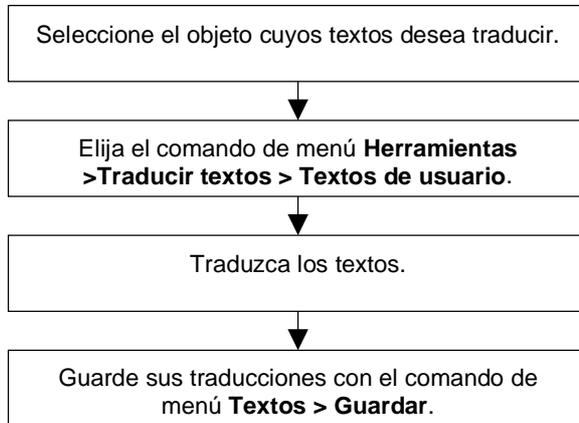
Esta función permite crear textos de usuario para un proyecto entero, para programas S7, para la carpeta de bloques o para determinados bloques así como para la tabla de símbolos, siempre que en dichos objetos puedan configurarse mensajes. Contienen todos los textos y mensajes que pueden visualizarse p.ej. en visualizadores. Para un proyecto puede haber varias listas de textos de usuario que se pueden traducir a los idiomas necesarios.

Puede elegir los idiomas disponibles en un proyecto (comando de menú: **Herramientas > Idioma para visualizador**). También puede añadir o borrar idiomas posteriormente.

Si abre los textos de usuario de un objeto STEP 7 (comando de menú: **Herramientas > Traducir textos > Textos de usuario**), en la pantalla se visualiza una tabla con una columna para cada idioma. En la primera columna se visualiza siempre el idioma ajustado como idioma estándar.

## Procedimiento básico

Asegúrese de que en el Administrador SIMATIC ha ajustado con el comando de menú **Herramientas > Idioma para visualizador** los idiomas a los que van a traducirse los textos de usuario.



## Exportar e importar textos de usuario

Los textos de usuario creados en STEP 7 pueden traducirse o editarse fuera de STEP 7. Para ello, exporte la lista visualizada de textos de usuario a archivos de texto en formato CSV que pueda editar con un editor ASCII o con una herramienta de procesamiento de tablas, p.ej. la aplicación EXCEL de Microsoft. A continuación importe de nuevo los textos a STEP 7.

Los textos de usuario sólo pueden importarse a la parte del proyecto desde la que se exportaron.

### 14.5.3 Traducir librerías de textos

Las librerías de textos proporcionan una lista de textos que pueden integrarse en mensajes, actualizarse dinámicamente durante el tiempo de ejecución y visualizarse en la PG o en otros visualizadores. Estas librerías están asignadas a CPUs. Los textos de librerías de textos del sistema son proporcionados por STEP 7 o por paquetes opcionales de STEP 7. Para una CPU pueden existir varias librerías de textos que se pueden traducir a los idiomas necesarios.

En el Administrador SIMATIC puede elegir los idiomas disponibles en un proyecto (comando de menú: **Herramientas > Idioma para visualizador**). También puede añadir o borrar idiomas posteriormente.

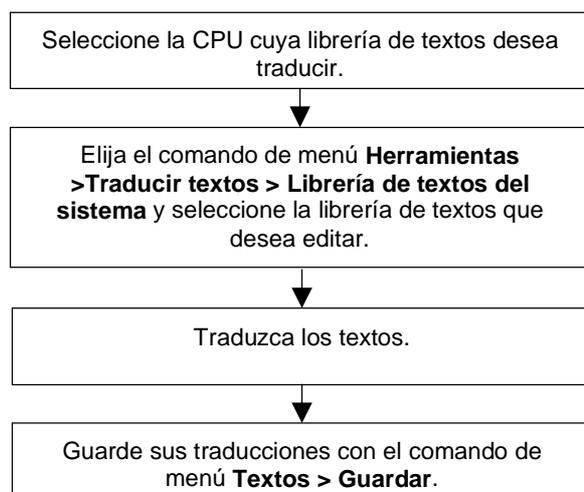
Si abre una librería de textos (comando de menú: **Herramientas > Traducir textos > Librería de textos del sistema**), en la pantalla se visualiza una tabla con una columna para cada idioma. En la primera columna se visualiza siempre el índice, que le permite referenciar los diferentes textos.

#### Ejemplo

índice	alemán	español
1732	ausgefallen	no responde
1733	gestört	averiado
1734	Parametrierfehler	error de parametrización

#### Procedimiento básico

Asegúrese de que en el Administrador SIMATIC ha ajustado con el comando de menú **Herramientas > Idioma para visualizador** los idiomas a los que va a traducirse la librería de textos.



## 14.6 Transferir los datos de configuración al sistema de destino

### 14.6.1 Transferir los datos de configuración al sistema de destino

#### Introducción

Los datos de configuración para manejo y visualización se transfieren con el programa "Ingeniería para PLC-OS" a la gestión de datos de WinCC.

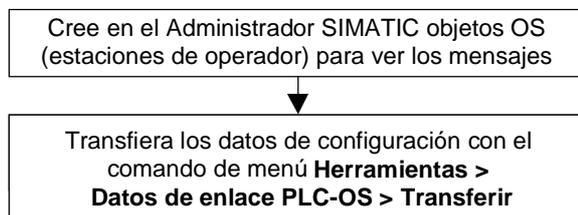
Hay varias opciones de transferencia. Por ejemplo, es posible actualizar las direcciones y el texto para garantizar que se transfieran los datos correctos.

#### Requisitos

Antes de transferir los datos se deberán cumplir los siguientes requisitos:

- Se deberá haber ejecutado el programa de instalación "Configuración de enlaces PLC-OS".
- Se deberán haber generado los datos de configuración para crear mensajes.

#### Procedimiento general



## 14.7 Visualizar mensajes de CPU y mensajes de diagnóstico personalizados

### 14.7.1 Visualizar mensajes de CPU y mensajes de diagnóstico personalizados

Con la función "Mensajes de CPU" es posible visualizar mensajes asíncronos de eventos de diagnósticos y mensajes personalizados o mensajes ALARM\_S/SQ.

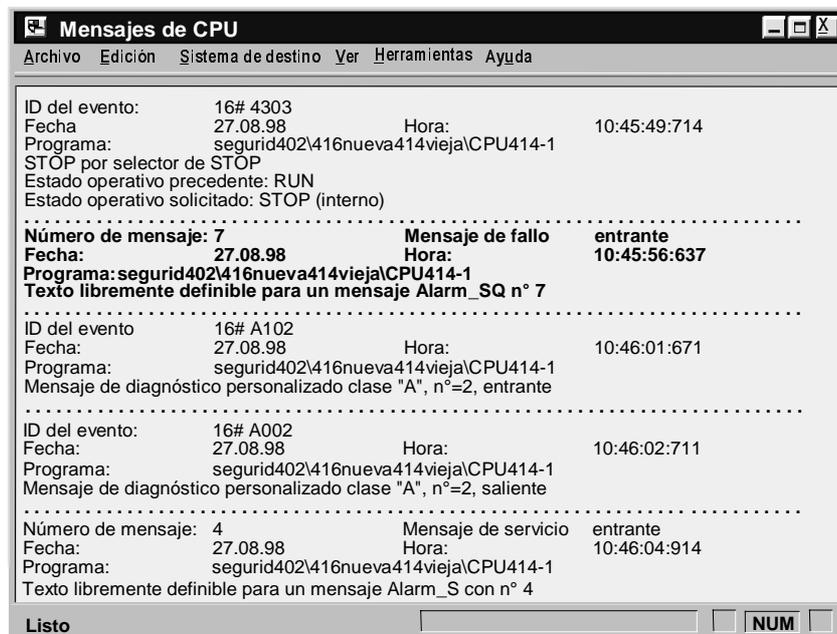
Además, si en la ventana "Mensajes de CPU" se elige el comando **Herramientas > Configurar mensajes**, es posible iniciar la configuración y crear mensajes de diagnóstico personalizados. Para ello inicie la función "Mensajes de CPU" en un proyecto online.

#### Posibilidades de visualización

La función "Mensajes de CPU" permite ajustar la visualización de los mensajes de las CPUs seleccionadas y qué mensajes se deben visualizar:

- **"Primer plano"**: la ventana con los mensajes de CPU aparecerá en primer plano. Esta ventana permanece en primer plano aunque se reciban nuevos mensajes.
- **"Segundo plano"**: los mensajes de CPU se reciben en segundo plano. La ventana permanece en segundo plano aunque se reciban nuevos mensajes, pudiéndose visualizar en caso necesario.
- **"Ignorar mensaje"**: los mensajes de CPU **no** se visualizan y, contrariamente a los dos primeros casos, **no** se archivan.

En la ventana "Mensajes de CPU" se pueden ver los mensajes que contiene el archivador. La siguiente figura muestra algunos ejemplos:



Los mensajes acusables (ALARM\_SQ) aparecen en negrita y se pueden acusar con el comando de menú **Edición > Acusar mensaje de CPU**.

## Función del archivador

Para archivar los mensajes se dispone de un archivador que puede contener entre 40 y 2000 mensajes de CPU. Si se excede el límite máximo ajustado para el archivador, se borra el mensaje más antiguo.

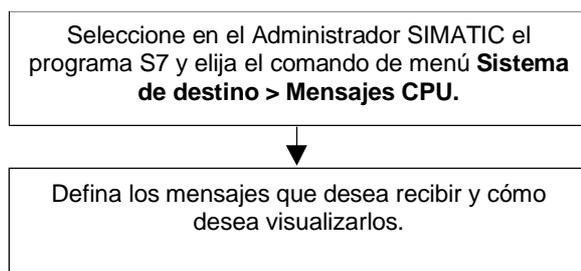
## Actualizar mensajes ALARM\_S/SQ

Al actualizar mensajes ALARM\_S/SQ se vuelven a registrar en el archivador los mensajes que no han salido o los que todavía no se han acusado. Los mensajes se actualizan en los siguientes casos:

- al rearrancar el módulo al que se refieren los mensajes (pero no al realizar un re arranque completo)
- al configurar los mensajes de CPU si hace clic en el campo "A" para "ALARM\_S/SQ" del cuadro de diálogo "Preferencias - Visualizar mensajes de CPU".

## Procedimiento general

Para configurar mensajes de CPU para los módulos seleccionados:



## 14.7.2 Configurar mensajes de CPU

Para configurar mensajes de CPU para determinados módulos proceda como sigue:

1. En el Administrador SIMATIC, inicie la aplicación "Mensajes de CPU" desde un proyecto online. Para ello seleccione online un programa S7 y, con el comando **Sistema de destino > Mensajes de CPU**, llame la aplicación "Mensajes de CPU" para la CPU seleccionada.

**Resultado:** Aparecerá la ventana de la aplicación "Mensajes de CPU", donde figura la CPU dada de alta.

2. Si es necesario, seleccione otras CPUs; para ello repita el primer paso para otros programas o interfaces.
3. Haga clic en las casillas de verificación que aparecen delante de las entradas de la lista y determine qué mensajes habrá de recibir el módulo:

A: activa ALARM\_SQ (SFC17) y ALARM\_S (SFC 18) p. ej., mensajes de diagnóstico de proceso de S7-PDIAG; S7-GRAPH o 'Notificar errores de sistema'.

W: activa eventos de diagnóstico  
Determine el tamaño del archivador.

4. **Resultado:** Cuando se presenten los mensajes mencionados, éstos se escribirán en el archivador y se visualizarán conforme al modo elegido.

---

### Nota

En la lista de módulos dados de alta que aparece en la ventana de la aplicación "Mensajes de CPU" están registradas las CPUs para las que eligió el comando de menú **Sistema de destino > Mensajes de CPU**. Las entradas de esta lista no se borran hasta que no se borren del cuadro de diálogo.

---

## 14.7.3 Visualizar los mensajes de CPU guardados

Los mensajes de la CPU se guardan siempre en el archivador, a menos que se haya elegido el comando de menú **Ver > Ignorar mensaje**. Todos estos mensajes archivados se visualizarán siempre.

## 14.8 Configurar 'Notificación de errores del sistema'

### 14.8.1 Configurar la 'notificación de errores del sistema'

#### Introducción

En caso de error del sistema, los componentes S7 y los esclavos DP normalizados pueden provocar llamadas a un bloque de organización.

Ejemplo: en caso de rotura de hilo, un módulo apto para diagnóstico puede disparar la alarma de diagnóstico (OB 82).

Los componentes S7 proporcionan información sobre los errores del sistema que se han producido. La información del evento de arranque, es decir, los datos locales del OB asignado (contienen, entre otros, el registro 0), proporcionan información general sobre el lugar (p. ej., dirección lógica del módulo) y el tipo del error (p. ej., error de canal o fallo del respaldo de la pila).

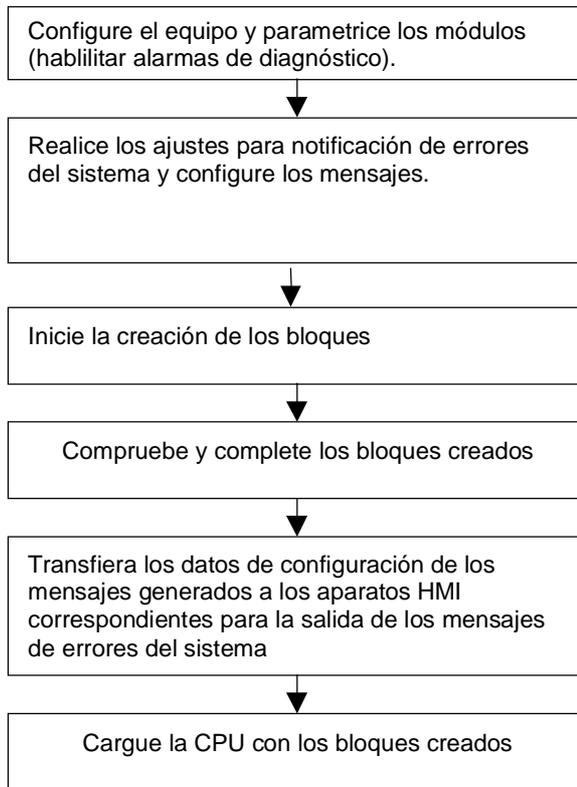
Además, informaciones de diagnóstico adicionales (lectura del registro 1 con SFC51 o lectura del telegrama de diagnóstico del esclavo normalizado DP con SFC13) permiten especificar con mayor precisión el error: p. ej., canal 0 o 1, rotura de hilo o rebase por exceso del margen de medida.

Con la función "Notificar errores del sistema", STEP 7 ofrece una forma cómoda de ver en forma de mensajes la información de diagnóstico proporcionada por los componentes.

Los bloques y textos de mensaje necesarios para ello son generados automáticamente por STEP 7. El usuario sólo tiene que cargar en la CPU los bloques generados y transferir los textos a los aparatos HMI conectados.

Encontrará una visión general de la información de diagnóstico que ofrecen los diferentes esclavos en el apartado Componentes compatibles y funcionalidad.

### Procedimiento básico



Los mensajes se envían a través de la ruta de notificación estándar ALARM\_S/SQ a "Mensajes de CPU" en la PG o a los aparatos HMI conectados.

## 14.8.2 Componentes compatibles y funcionalidad

Los componentes de los equipos S7-300, S7-400 y esclavos DP son compatibles con la herramienta "Notificar errores de sistema", siempre que soporten funciones tales como alarma de diagnóstico, alarma de presencia de módulo y diagnóstico específico de un canal.

Los siguientes componentes **no** son compatibles con "Notificar errores del sistema":

- Configuraciones M7, C7 y PROFIBUS-DP a través de interfaces maestra DP (CP 342-5 DP) en equipos S7-300
- sistemas H
- equipos PC SIMATIC (Soft-PLC, Slot-PLC)

En caso de un rearranque normal hay que tener en cuenta, además, que pueden faltar mensajes de alarma. Esto se debe a que la memoria de acuse de mensajes de la CPU no se borra en el rearranque normal, pero la herramienta "Notificar errores del sistema" restaura los datos internos.

En las dos tablas siguientes se encuentran todos los bloques de diagnóstico de los distintos esclavos compatibles con "Notificar errores del sistema".

Bloque de diagnóstico	Denom. ID (Slot defectuoso)	Denom. canal (error de canal) 1)	Estado del módulo (error de módulo, mód. erróneo/sin mód.)	Espec. fabricante	Denom. aparato (sólo en ET 200 B)
<b>ID encabez.2)</b>	<b>0x01</b>	<b>0x10</b>	<b>0x00 Tipo 0x82</b>	<b>0x00 Tipo &gt; 0x9F</b>	<b>0x00 + 1 Byte info diag.</b>
ET 200 S	Mensaje: 'Consultar el diagnóstico'	Mensaje en texto claro	Mensaje en texto claro	Mensaje en texto claro del archivo GSD 3)	- 4)
ET 200 M como esclavo S7	no se evalúa	no se evalúa	no se evalúa	-	-
ET 200 M como esclavo normalizado	Mensaje: 'Consultar el diagnóstico'	Mensaje en texto claro	Mensaje en texto claro	-	-
ET 200 X	Mensaje: 'Consultar el diagnóstico'	-	-	-	-
ET 200 X DESINA	Mensaje: 'Consultar el diagnóstico'	Mensaje en texto claro	Mensaje en texto claro	-	-
ET 200 L	Mensaje: 'Consultar el diagnóstico'	-		-	-

Bloque de diagnóstico	Denom. ID (Slot defectuoso)	Denom. canal (error de canal) 1)	Estado del módulo (error de módulo, mód. erróneo/sin mód.)	Espec. fabricante	Denom. aparato (sólo en ET 200 B)
ET 200 B Digital	Mensaje: 'Consultar el diagnóstico'	-	-	-	Mensaje: 'Módulo defectuoso'
ET 200 B Analógico	Mensaje: 'Consultar el diagnóstico'	-	-	-	

1) Mensajes estándar según DPV0, en esclavos Norm también los del archivo GSD  
 2) ID encabez.: ID en el telegrama de diagnóstico que identifica distintas partes del diagnóstico.  
 3) Archivo GSD: Descripción de un esclavo Norm DPV0 o DPV1 en forma de archivo.  
 4) "-" significa que el componente no aporta la información.

Bloque de diagnóstico	DS0/DS1 1)	Estado H (Los sistemas H aún no son compatibles)	otras expresiones
<b>ID encabezado 2)</b>	<b>0x00</b> <b>Tipo 0x01</b>	<b>0x00</b> <b>Tipo = 0x9E o</b> <b>Tipo = 0x9F</b>	<b>0x00</b> <b>otros tipos</b>
ET 200 S	Mensaje en texto claro	- 3)	no se evalúa
ET 200 M como esclavo S7	Mensaje en texto claro	no se evalúa	no se evalúa
ET 200 M como esclavo normalizado	Mensaje en texto claro	aún no se evalúa	no se evalúa
ET 200 X	Mensaje: 'Módulo defectuoso'	-	no se evalúa
ET 200 X DESINA	Mensaje en texto claro	-	no se evalúa
ET 200 L	Mensaje: 'Módulo defectuoso'	-	-
ET 200 B Digital	-	-	-
ET 200 B Analógico	Mensaje en texto claro	-	no se evalúa

1) DS0: Diagnóstico estándar, p. ej., fallo del módulo, falta tensión auxiliar externa o el conector frontal, tamaño 4 bytes, reside en los datos locales de OB 82.  
 DS1: Error de canal, definido de forma distinta para cada tipo de canal, legible en el programa de usuario mediante SFC 51.  
 Los textos provienen del diagnóstico HW S7.  
 2) ID encabez.: ID contenido en el telegrama de diagnóstico que identifica distintas partes del diagnóstico.  
 3) "-" significa que el componente no aporta la información.

El telegrama de diagnóstico (también denominado telegrama de esclavo normalizado) está formado por los bloques de diagnóstico mencionados anteriormente y se puede leer en el programa de usuario a través de SFC 13.

En STEP 7 se visualiza el telegrama de diagnóstico a través de la llamada del estado del módulo en la ventana online "HW-Config" (Diagnosticar hardware) en la ficha "Diagnóstico de esclavo DP" bajo "Representación hexadecimal".

### 14.8.3 Ajustes para notificar errores del sistema

Dispone de varias posibilidades de abrir el diálogo para definir los ajustes:

- Seleccione en HW Config la CPU para la que desea configurar la notificación de errores del sistema. Elija a continuación el comando de menú **Herramientas > Notificar errores del sistema**.
- Si ya ha creado bloques para notificar errores del sistema, puede abrir el diálogo haciendo doble clic sobre uno de los bloques creados (FB, DB).
- En el diálogo de propiedades del equipo, seleccione la opción para abrir automáticamente al guardar y compilar la configuración

Puede acceder a la opción para abrir automáticamente al guardar y compilar:

1. Seleccionando el equipo correspondiente en el Administrador SIMATIC.
2. Seleccionando el comando de menú **Edición > Propiedades del objeto**.
3. Seleccionando la ficha "Preferencias".

En el diálogo, introduzca, entre otros datos:

- el FB y el DB de instancia asignado que debe crearse
- puede seleccionar si la CPU deberá ponerse en STOP, o no, después de notificar un error de sistema
- si en el programa S7 se deben o no generar OBs de errores no existentes
- si los mensajes deben ser acusables
- si deben generarse datos de referencia
- si se deben visualizar advertencias durante la generación de Notificar errores del sistema
- si debe mostrarse el diálogo si se llama automáticamente Notificar errores del sistema después de guardar y compilar la configuración (ver ajuste más arriba)
- la forma de visualización de los mensajes (estructura y orden de las partes de texto posibles)

Puede consultar informaciones detalladas en la Ayuda del diálogo abierto.

#### 14.8.4 OBs de error generados

Si ha activado la casilla de verificación "Generar OBs de error" en la ficha "General" del cuadro de diálogo "Notificar errores de sistema", Al generar se crearán los siguientes OBs de error:

- OB 81 (fallo de alimentación) con llamada del FB de diagnóstico generado
- OB 82 (OB de alarma de diagnóstico; sólo si se han configurado módulos o esclavos DP) con llamada del FB de diagnóstico generado
- OB 83 (alarma de presencia de módulo) con llamada del FB de diagnóstico generado.
- OB 84 (fallo de CPU)  
Este OB se genera sin contenido, para que la CPU no pase a estado operativo STOP al producirse errores de comunicación (por ejemplo, si surgieran problemas con la resistencia terminadora del MPI al enchufar o desenchufar el cable MPI). Los errores no se evalúan y no se genera ningún mensaje.
- OB 85 (error de ejecución del programa)  
Se impide que la CPU pase a STOP solamente cuando se producen errores de actualización de la imagen del proceso (p. ej., por extraer el módulo) para que así se pueda procesar el FB de diagnóstico en OB 83. Si se ha ajustado CPU-STOP, será efectiva en OB 83 después del mensaje de "Notificar errores del sistema".  
En los demás eventos de error de OB 85, la CPU pasará a STOP.
- OB 86 (fallo de un bastidor de ampliación, de un sistema maestro DP o de un aparato de la periferia descentralizada) con llamada del FB de diagnóstico generado.  
Este OB sólo se genera si se ha configurado uno de los componentes mencionados.

#### Si ya existen OBs de error...

Los OBs de error existentes no se sobrescriben. Debe insertar la llamada del FB generado en el OB correspondiente.

#### Si la configuración incluye aparatos de la periferia descentralizada...

Para evaluar errores en la periferia descentralizada, el FB generado llama automáticamente al SFC 13 (lectura de los datos de diagnóstico de los esclavos DP). Para garantizar que esto funcione, debe llamar el FB generado o sólo en el OB 1 o en un OB de alarma cíclica con una frecuencia de reloj breve en los OB de arranque.

#### ATENCIÓN:

Tenga en cuenta que:

- El OB 85 generado por "Notificar errores del sistema" en el evento de error "Error al actualizar la imagen del proceso", hace que no se produzca NINGÚN STOP de la CPU.
- El OB 85 de la CPU también será llamado en los casos de error
  - "Evento de error para un OB no cargado"
  - "Error al llamar o acceder a un bloque no cargado"

En estos casos de error se produce con el OB 85 generado por "Notificar errores del sistema" también un STOP de la CPU como antes de utilizar "Notificar errores del sistema"

- El ajuste "La CPU pasa a STOP tras ejecutar el FB de diagnóstico" no tiene efecto para OB 84 y OB 85 porque en estos OB no se llama el FB de "Notificar errores del sistema". En el caso de OB 85, este ajuste se tiene en cuenta indirectamente por la llamada de FB en OB 83.

### 14.8.5 FB, DB generados

El FB generado evalúa los datos locales del OB de error, lee, dado el caso, las informaciones de diagnóstico adicionales de los componentes S7 que originan los errores y genera automáticamente el mensaje correspondiente.

Sus propiedades son:

- Lenguaje SFM (= "Notificar errores del sistema", también válido para el DB de instancia generado)
- Protección Know-How activada (también válido para el DB de instancia generado)
- Retarda las alarmas entrantes durante el tiempo de ejecución
- Mediante doble clic abre el diálogo para ajustar la función "Notificar errores del sistema" (también válido para el DB de instancia generado).

#### Bloque de usuario

Dado que el FB de diagnóstico tiene activada la protección Know-how, no se puede editar. El FB proporciona un interface para el programa de usuario, de manera que se tiene acceso, por ejemplo, al estado del error o al número de mensaje.

El bloque para la evaluación en el programa de usuario (que puede ajustarse en la ficha "Bloque de usuario" del diálogo) se llama desde el FB generado con los siguientes parámetros:

```
EV_C : BOOL ;           //mensaje entrante (TRUE) o saliente (FALSE)
EV_ID : DWORD ;        //número de mensaje generado
IO_Flag: BYTE ;        //módulo de entrada: B#16#54 módulo de salida: B#16#55
logAdr : WORD ;        //dirección lógica
TextlistId : WORD ;    //ID de la librería de textos (librería estándar = 1)
ErrorNo : WORD ;      //número de error generado
Channel_Error : BOOL ; //error de canal (TRUE)
ChannelNo : WORD ;     //número de canal
```

Si aún no existe ningún FB de usuario lo generará SFM (=Notificar errores de sistema) con los parámetros que se mencionan en la lista anterior.



# 15 Configurar variables para manejo y visualización

## 15.1 Configurar variables para manejo y visualización

### Sinopsis

STEP 7 ofrece un método cómodo para manejar y visualizar magnitudes variables del proceso o del sistema de automatización con WinCC.

La ventaja frente a los métodos habituales es que los datos no se deben configurar individualmente para cada estación de operador (OS), sino una sola vez con STEP7. Los datos creados con STEP 7 durante la configuración se pueden transferir mediante el programa 'Ingeniería para PLC-OS' (que forma parte del paquete de software "Sistema de control de procesos PCS7") a la gestión de datos de WinCC. En dicho proceso se comprueba la coherencia y la compatibilidad de los datos con el sistema de visualización. WinCC utiliza los datos en bloques de imágenes y en gráficos.

STEP 7 permite configurar y/o cambiar los atributos de manejo y visualización (M+V) de las siguientes variables:

- Parámetros de entrada, de salida y de entrada/salida de bloques de función
- Marcas y señales de E/S
- Parámetros de bloques CFC en esquemas CFC

### Procedimiento general

El procedimiento para configurar variables que se puedan manejar y visualizar depende del lenguaje de programación o de configuración elegido, así como del tipo de variables que se deseen manejar y visualizar. No obstante, el procedimiento general comprende siempre los siguientes pasos:

1. Asigne atributos de manejo y visualización (M+V) a los parámetros de un bloque de función o a los símbolos de una tabla de símbolos.

Este paso no es necesario en CFC, puesto que los bloques se pueden extraer ya listos de una librería.

2. En un cuadro de edición, asigne a las variables que desea manejar y visualizar los atributos M+V necesarios (S7\_m\_c). En el cuadro de diálogo "Manejo y visualización" (comando de menú **Edición > Propiedades especiales del objeto > Manejo y visualización**) podrá modificar los atributos de WinCC tales como los límites, los valores de sustitución, las propiedades de los informes, etc.
3. Con el programa 'Ingeniería para PLC-OS', transfiera los datos de configuración creados con STEP 7 al sistema de visualización (WinCC).

## Convenciones relativas a los nombres

Para que los datos de configuración se puedan guardar y transferir a WinCC, se almacenan con un nombre unívoco adjudicado automáticamente por STEP 7. Los nombres de las variables que se pueden manejar y visualizar, así como de los esquemas CFC y de los programas S7, forman parte de dicho nombre unívoco, por lo que se deben respetar determinadas convenciones:

- Los nombres de los programas S7 contenidos en un proyecto S7 deben ser unívocos (en diferentes equipos no puede haber programas S7 con un mismo nombre).
- Los nombres de las variables, de los programas S7 y de los esquemas CFC no pueden contener caracteres de subrayado, blancos o los siguientes caracteres especiales: [ ' ] [ . ] [ % ] [ - ] [ / ] [ \* ] [ + ].

## 15.2 Configurar atributos M+V con AWL, KOP y FUP

### 15.2.1 Configurar atributos M+V con AWL, KOP y FUP

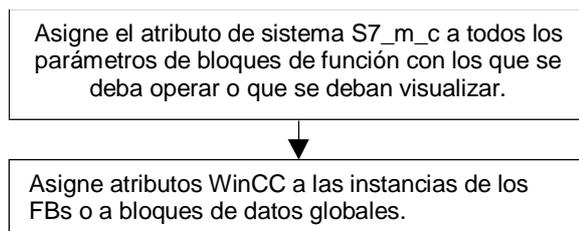
#### Introducción

Con el procedimiento aquí descrito es posible manejar y visualizar los parámetros de bloques de función y asignar los atributos M+V (para manejo y visualización) necesarios a los bloques de datos (DBs) de instancia o globales asociados en el programa de usuario.

#### Requisito

Se deberán haber creado previamente un proyecto STEP7, un programa S7 y un FB.

#### Procedimiento general



## 15.3 Configurar atributos M+V a través de la tabla de símbolos

### 15.3.1 Configurar atributos M+V a través de la tabla de símbolos

#### Introducción

Independientemente del lenguaje de programación utilizado, con el procedimiento descrito abajo es posible configurar las siguientes variables:

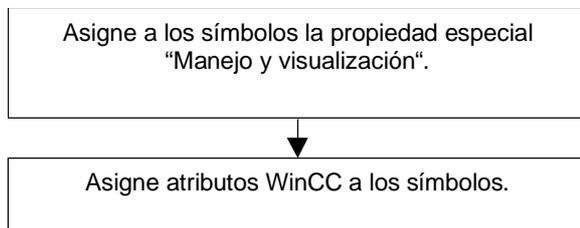
- Marcas
- Señales de E/S

#### Requisito

Antes de comenzar con la configuración se deberán cumplir los siguientes requisitos:

- Se deberá haber creado un proyecto en el Administrador SIMATIC.
- En dicho proyecto deberá existir un programa S7 con una tabla de símbolos.
- La tabla de símbolos deberá estar abierta.

#### Procedimiento general



## 15.4 Cambiar atributos M+V con CFC

### 15.4.1 Cambiar atributos M+V con CFC

#### Introducción

En CFC los programas de usuario se crean seleccionando de una librería los bloques destinados a manejo y visualización y posicionándolos e interconectándolos en un esquema.

#### Requisito

Antes será preciso insertar un programa S7 en un proyecto de STEP7, generar un esquema CFC y colocar allí los bloques.

#### Procedimiento general

Edite las propiedades de los bloques  
(Propiedades del objeto)

---

#### Nota

Si utiliza bloques creados por usted y a los que ha asignado el atributo de sistema S7\_m\_c en el lenguaje de programación, puede asignar a dichos bloques los atributos M+V (para manejo y visualización) necesarios, activando la opción "Manejo y visualización" en el cuadro de diálogo "Manejo y visualización" (comando de menú **Edición > Propiedades especiales del objeto > Manejo y visualización**).

---

## 15.5 Transferir los datos de configuración al sistema de destino M+V

### 15.5.1 Transferir los datos de configuración al sistema de destino M+V

#### Introducción

Los datos de configuración para manejo y visualización se transfieren con el programa 'Configuración de enlaces PLC-OS' a la gestión de datos de WinCC.

Hay varias opciones de transferencia. Por ejemplo, es posible actualizar las direcciones y el texto para garantizar que se transfieran los atributos WinCC correctos.

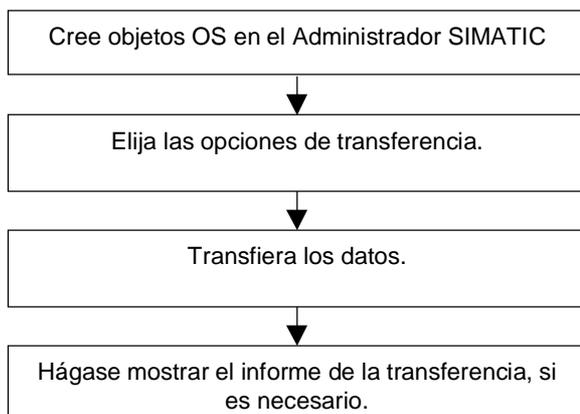
#### Requisito

Antes de transferir los datos se deberán cumplir los siguientes requisitos:

- Se deberá haber ejecutado el programa de instalación "Configuración de enlaces PLC-OS" (Engineering).
- Se deberán haber generado los datos de configuración para manejo y visualización.

#### Procedimiento general

Para transferir los datos de configuración para manejo y visualización a la gestión de datos de WinCC proceda como sigue:



# 16 Establecer enlaces online y ajustar la CPU

## 16.1 Establecer enlaces online

### 16.1.1 Establecer enlaces online

Un enlace online entre el sistema de origen y el sistema de destino se requiere para cargar programas de usuario S7 y bloques del sistema de origen en el sistema de destino y para cargar bloques del sistema de destino S7 en el sistema de origen, así como para las siguientes actividades:

- Comprobar programas de usuario
- Visualizar y cambiar el estado operativo de la CPU
- Visualizar y ajustar la fecha y la hora de la CPU
- Visualizar el estado de los módulos
- Comparar bloques online/offline
- Diagnosticar el hardware

Para poder establecer un enlace online, el sistema de origen y el sistema de destino deberán estar comunicados mediante un interface apropiado (p.ej. interface multipunto (MPI)). Desde la ventana online del proyecto o desde la ventana "Estaciones accesibles" se podrá acceder entonces al sistema de destino.

### 16.1.2 Establecer un enlace online desde la ventana "Estaciones accesibles"

Esta alternativa le permite acceder rápidamente al sistema de destino, p.ej., para efectos de mantenimiento. Puede acceder a todos los módulos programables accesibles en la red. Elija este método si no existen datos de proyecto para los sistemas de destino en su PG.

A la ventana "Estaciones accesibles" se accede con el comando de menú **Sistema de destino > Estaciones accesibles**. El objeto "Estaciones accesibles" permite visualizar todos los módulos programables accesibles por red con su respectiva dirección.

También pueden visualizarse estaciones no programables con STEP 7 (p.ej.: unidades de programación o paneles de operador).

### 16.1.3 Establecer un enlace online desde la ventana online del proyecto

Elija este método si ha configurado el sistema de destino en un proyecto en su PG/PC. Para acceder a la ventana online desde el Administrador SIMATIC, elija el comando de menú **Ver > Online**. Allí se indican los datos del proyecto contenidos en el sistema de destino (contrariamente a la ventana offline, donde aparecen los datos del proyecto contenidos en la PG o en el PC). En la ventana online se visualizan los datos tanto del programa S7 como del programa M7 que se encuentran depositados en el sistema de destino.

Esta vista se utiliza para acceder al sistema de destino. Algunas funciones del menú "Sistema de destino" del Administrador SIMATIC pueden llamarse en el modo de visualización "online", pero no en el modo "offline".

Se distingue entre:

- Acceso con hardware configurado  
Así puede acceder sólo a los módulos configurados offline. De la dirección MPI que se haya ajustado al configurar el módulo programable depende a qué módulos es posible acceder online.
- Acceso sin hardware configurado  
Para este tipo de acceso se requiere un programa S7 o un programa M7 que se haya creado independientemente del hardware (es decir, directamente bajo el proyecto). De la dirección MPI indicada en las propiedades del programa S7 o M7, respectivamente, depende a qué módulos es posible acceder online.

Este tipo de acceso combina los datos del sistema de destino con los del sistema de origen. Si, por ejemplo, abre online un bloque S7 bajo un proyecto, se muestra lo siguiente:

- área de código del bloque de la CPU del sistema de destino S7 y
- comentarios y símbolos de la gestión de datos del sistema de origen (siempre que existan offline). Si abre bloques directamente en la CPU conectada sin que exista una estructura del proyecto, se visualizan tal y como se encuentran en la CPU, es decir, sin símbolos ni comentarios.

### 16.1.4 Protección con contraseña para acceder a sistemas de destino

Con la contraseña es posible

- proteger el programa de usuario en la CPU y sus datos contra cambios no deseados (protección contra escritura)
- preservar el know how contenido en su programa de usuario (protección contra lectura)
- impedir funciones online que pudieran perturbar el proceso

Un módulo sólo se podrá proteger con contraseña si aquel asiste dicha función.

Si desea proteger un módulo con una contraseña deberá definir el nivel de protección y la contraseña en la parametrización del módulo y cargar luego ésta última en el mismo.

Si la contraseña se requiere para poder ejecutar una función online, se visualizará el cuadro de diálogo "Introducir contraseña". Introduciendo la contraseña debida se obtiene un permiso de acceso a los módulos para los cuales se ha definido un nivel de protección en la parametrización. Ello le permitirá establecer enlaces online con el módulo protegido y ejecutar las funciones online que corresponden al nivel de protección.

Con el comando de menú **Sistema de destino > Permiso de acceso > Crear** puede llamar directamente el cuadro de diálogo para introducir la contraseña. Así, p.ej., es posible introducir una vez la contraseña al principio de la sesión, con lo cual ya no se le pedirá que la introduzca cada vez que acceda online. La contraseña introducida permanecerá activa hasta que se vuelva a salir del Administrador SIMATIC o bien hasta ser anulada con el comando de menú **Sistema de destino > Permiso de acceso > Cancelar**.

Parámetros de la CPU	Notas
Test/Proceso (no para S7-400 ni CPU 318-2)	Ajustable en la ficha "Protección". En el modo de proceso, las funciones de test tales como el estado del programa o la observación y el forzado de variables se limitan de manera tal que no se exceda el tiempo de ciclo admisible ajustado. Por ejemplo, en el estado de programa no se permiten las condiciones de llamada y la visualización del estado de un bucle programado se interrumpe en el punto de retorno. La comprobación con puntos de parada y ejecución del programa paso a paso no se puede efectuar en el modo de proceso. En el modo de test se pueden utilizar sin limitación alguna todas las funciones de comprobación a través de la PG o del PC, incluyendo las que pudieran prolongar en gran medida el tiempo de ciclo.
nivel de protección	Ajustable en la ficha "Protección". Los accesos de escritura y/o de lectura a la CPU pueden someterse a la previa introducción de una contraseña. Dicha contraseña se parametriza en esa ficha.

### 16.1.5 Nota respecto a la actualización del contenido de ventanas

Tenga en cuenta lo siguiente:

- Los cambios en la ventana online de un proyecto debidos a acciones efectuadas por el usuario (p.ej. cargar o borrar bloques) no se adoptan automáticamente en la ventana "Estaciones accesibles" que pudiese estar abierta.
- Los cambios que se hagan en la ventana "Estaciones accesibles" tampoco se adoptan automáticamente en una ventana online de un proyecto que pudiese estar abierta.

Con objeto de adoptar la visualización actual en otra ventana abierta simultáneamente, ésta se deberá actualizar también (mediante el correspondiente comando de menú o la tecla de función F5).

## 16.2 Visualizar y cambiar el estado operativo

### 16.2.1 Visualizar y cambiar el estado operativo

Utilizando esta función puede cambiar la CPU nuevamente al estado operativo "RUN" tras haber corregido un error.

#### Visualizar el estado operativo

1. Abra su proyecto y seleccione un programa S7 o M7, o bien elija el comando de menú **Sistema de destino > Mostrar estaciones accesibles** para abrir la ventana "Estaciones accesibles" y seleccione una estación ("MPI=...").
2. Elija el comando **Sistema de destino > Estado operativo**.

Este cuadro de diálogo muestra el estado operativo actual y el último, así como la posición actual del selector del módulo. Para los módulos cuya posición actual de selector no pueda ser leída aparece el texto "sin definir".

#### Cambiar el estado operativo

Mediante los botones de comando puede cambiar el estado operativo del módulo CPU. Sólo estarán activos los botones de comando que se puedan seleccionar en el estado operativo actual.

## 16.3 Visualizar y ajustar la fecha y la hora

### 16.3.1 Visualizar y ajustar la fecha y la hora

Proceda como sigue:

1. Abra su proyecto y seleccione un programa S7 o M7, o bien elija el comando de menú **Sistema de destino > Estaciones accesibles** para abrir la ventana "Estaciones accesibles" y seleccione una estación ("MPI=...").
2. Elija el comando **Sistema de destino > Ajustar la hora**.  
El comando se puede activar si en la ventana del proyecto (modo online) hay seleccionado un programa S7/M7, o en la ventana "Estaciones accesibles" una estación ("MPI=...").
3. En el cuadro de diálogo visualizado podrá leer la hora y la fecha actuales del módulo seleccionado.
4. Si fuera necesario, podrá introducir nuevos valores en los cuadros de entrada "Fecha" y "Hora" o adoptar la hora y la fecha de su PG/PC mediante la opción predeterminada.

---

#### Nota

En un módulo que no tenga integrado un reloj de tiempo real aparece como fecha 00.00.00 y como hora 00:00:00.

---



# 17 Cargar

## 17.1 Cargar en el sistema de destino desde la PG

### 17.1.1 Requisitos para cargar

#### Requisitos para cargar en el sistema de destino

- Haber establecido un enlace entre la PG y la CPU del sistema de destino (p.ej., vía el interface MPI).
- Poder acceder al sistema de destino.
- El programa que se desea cargar se deberá haber compilado sin errores.
- La CPU se tiene que encontrar en un estado operativo en el que se pueda cargar (STOP o RUN-P).  
No obstante, si se cargan en modo RUN-P, el programa se transferirá de bloque en bloque. Si sobrescribe un antiguo programa de CPU, pueden aparecer conflictos, p.ej., si se han modificado parámetros de bloques. Entonces, al procesarse el ciclo, la CPU pasa al estado operativo STOP. Es por ello que se recomienda cambiar la CPU a "STOP" antes de cargar.
- Caso que haya abierto el bloque offline y lo desee cargar, el módulo CPU tendrá que estar asignado a un programa de usuario online en el Administrador SIMATIC.
- Antes de cargar el programa de usuario en la CPU es recomendable efectuar un borrado total de la misma para evitar que contenga bloques "antiguos".

#### Estado operativo STOP

Cambie el estado operativo de RUN a STOP antes de

- cargar todo el programa de usuario o partes del mismo en la CPU,
- activar el borrado total de la CPU,
- comprimir la memoria de usuario.

#### Rearranque completo (cambio al estado operativo RUN)

Si realiza un reارئانque completo partiendo del modo "STOP", se reinicia el programa arrancando primero en modo "ARRANQUE" (en el bloque OB100). Si el arranque ha tenido éxito, la CPU cambia a modo RUN. Un reارئانque completo es necesario después de:

- efectuar el borrado total de la CPU,
- cargar el programa de usuario en el estado STOP,

### 17.1.2 Diferencia entre guardar y cargar bloques

Por principio, se diferencia entre guardar y cargar bloques.

	<b>Guardar</b>	<b>Cargar</b>
Comandos de menú	<b>Archivo &gt; Guardar</b> <b>Archivo &gt; Guardar como</b>	<b>Sistema de destino &gt; Cargar</b>
Función	El estado actual del bloque en el editor se guarda en el disco duro de la PG.	El estado actual del bloque en el editor se carga sólo en la CPU.
Comprobación de sintaxis	Se comprueba la sintaxis. En caso de detectarse errores, éstos se visualizarán en cuadros de diálogo donde se indican la causa y el lugar de los errores. Dichos errores deberán corregirse antes de guardar o cargar el bloque. Si la sintaxis es correcta, el bloque se compila a continuación en código máquina y se guarda o se carga.	Se comprueba la sintaxis. En caso de detectarse errores, éstos se visualizarán en cuadros de diálogo donde se indican la causa y el lugar de los errores. Dichos errores deberán corregirse antes de guardar o cargar el bloque. Si la sintaxis es correcta, el bloque se compila a continuación en código máquina y se guarda o se carga.

La tabla será válida independiente de si el bloque se ha abierto online u offline.

#### Nota respecto a los cambios en bloques – guardar primero y cargar luego

Para adoptar bloques nuevos o bien los cambios realizados en el área de instrucciones de bloques lógicos, en tablas de declaración, o bien, en valores de bloques de datos, deberá guardar el bloque correspondiente. Los cambios que se efectúen en el editor y que se transfieran a la CPU con el comando de menú **Sistema de destino > Cargar** – p.ej., para comprobar cambios pequeños – se deberán guardar en todo caso también en el disco duro de la PG antes de salir del editor. En caso contrario, el estado del programa de usuario en la CPU será diferente de la versión contenida en la PG. En general, es aconsejable guardar primero los cambios y cargar a continuación.

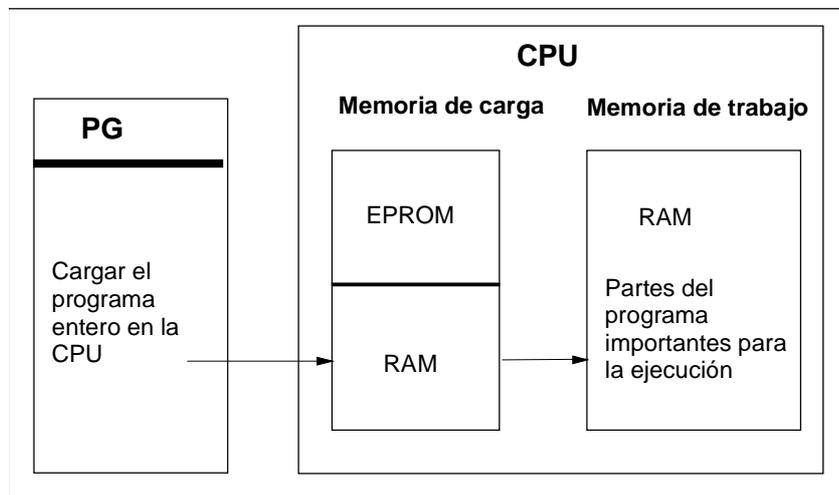
### 17.1.3 Memorias de carga y de trabajo de la CPU

Una vez concluida la configuración, la parametrización y la creación de programas, así como el establecimiento del enlace online, es posible transferir programas de usuario enteros o bloques individuales a un sistema de destino. Para comprobar bloques individuales es preciso cargar por lo menos un OB, así como los FBs y las FCs llamados por dicho OB, así como los DBs utilizados. Para poder transferir al sistema de destino los datos de sistema que se han generado después de configurar el hardware y las redes, o bien, después de haber creado una tabla de enlaces, hay que cargar el objeto "Bloques de datos del sistema".

Utilizando el Administrador SIMATIC, los programas de usuario se cargan en un sistema de destino, p.ej. en la fase final de la comprobación del programa o para ejecutar el programa de usuario ya listo.

## Interacción de las memorias de carga y de trabajo en la CPU

El programa de usuario entero se transfiere a la memoria de carga, en tanto que las partes ejecutables del mismo se cargan también en la memoria de trabajo.



### Memoria de carga de la CPU

- En la memoria de carga se guarda el programa de usuario sin la tabla de símbolos y sin comentarios (éstos permanecen en el área de memoria de la PG).
- Los bloques caracterizados como no relevantes para la ejecución se memorizan exclusivamente en la memoria de carga.
- La memoria de carga puede ser una RAM, una ROM o una EPROM.
- En el caso de los S7-300 la memoria de carga puede integrar una parte de memoria RAM y una parte de memoria EEPROM (p. ej., CPU312 IFM y CPU314 IFM).
- En el caso de los S7-400 para poder ampliar la memoria de usuario es imprescindible utilizar una Memory Card (RAM o EEPROM).

### Memoria de trabajo de la CPU

En la memoria de trabajo (RAM integrada) se guardan las partes ejecutables del programa de usuario.

## Procedimientos posibles al cargar

El programa de usuario o los objetos cargables (p.ej. los bloques) se pueden cargar en el sistema de destino mediante la función de carga. Si un bloque ya existe en la RAM de la CPU, se le consulta, antes del proceso de carga, si desea sobrescribirlo o no.

- Es posible seleccionar los objetos mencionados en la ventana del proyecto y cargarlos desde el Administrador SIMATIC (comando: **Sistema de destino > Cargar**).
- Al configurar el hardware y las redes, así como al programar bloques, el objeto recién editado se puede cargar directamente mediante el menú de la ventana principal correspondiente a la actividad en cuestión (con el comando **Sistema de destino > Cargar**).
- Otra posibilidad consiste en abrir una ventana online con vista al sistema de destino (p.ej. utilizando los comandos de menú **Ver > Online** o **Sistema de destino > Estaciones accesibles**) y copiar luego en la ventana online el objeto a cargar.

Con esta función también se puede cargar en la PG el contenido actual de los bloques de la memoria de carga RAM de la CPU.

### 17.1.4 Posibilidades de cargar en función de la memoria de carga

La división de la memoria de carga en un área RAM y un área EEPROM tiene consecuencias al cargar el programa de usuario o bloques del mismo. Para cargar los datos en la CPU dispone de las siguientes posibilidades:

Memoria de carga	Posibilidades de carga	Tipo de comunicación entre los sistemas de origen y de destino
RAM	Cargar y borrar bloques individuales	Enlace online PG – sistema de destino
	Cargar y borrar un programa de usuario entero	Enlace online PG – sistema de destino
	Recargar bloques individuales	Enlace online PG – sistema de destino
EPROM integrada (sólo en el S7-300) o enchufable	Cargar programas de usuario enteros	Enlace online PG – sistema de destino
EPROM enchufable	Cargar programas de usuario enteros	Cargar la EPROM externamente e insertar la Memory Card o bien acceder online a la EPROM que está insertada en el sistema de destino.

### Cargar en la RAM mediante un enlace online

Al producirse un corte de alimentación, los datos no se guardan en el sistema de destino si la RAM no tiene respaldo. En este caso, se pierden los datos que residen en la RAM.

## Guardar en una Memory Card EPROM

Los bloques o el programa de usuario se guardan en una Memory Card EPROM que se enchufa luego en una ranura de la CPU.

Las Memory Cards son soportes de datos portátiles. Se escriben en el sistema de origen, siendo enchufadas luego en el correspondiente receptáculo de la CPU.

Los datos que se guardan allí se conservan aún en caso de un corte de alimentación o de un borrado total de la CPU. Tras un borrado total de la CPU o a un fallo de la alimentación, si la RAM no tiene respaldo, el contenido de la EPROM se copia nuevamente en el área RAM de la memoria de la CPU cuando retorna la alimentación.

## Guardar en la EPROM integrada

En el caso de la CPU 312 existe también la posibilidad de copiar el contenido de la RAM en la EPROM integrada. Los datos contenidos en dicha EPROM permanecen remanentes aunque se produzca un corte de la alimentación. Tras un borrado total de la CPU o un corte de alimentación, si la RAM no tiene respaldo, el contenido de la EPROM integrada se copia nuevamente en el área RAM de la memoria de la CPU al retornar la alimentación.

### 17.1.5 Recargar bloques en el sistema de destino

Puede sobrescribir los bloques existentes en la memoria de carga (RAM) o en la memoria de trabajo de la CPU del sistema de destino S7. La nueva versión del bloque reemplazará el contenido anterior.

El procedimiento de sobrescritura es similar al de cargar bloques S7. Sólo se le pregunta si desea existir sobrescribir el bloque existente.

No es posible borrar un bloque que esté guardado en la EPROM, pero éste se declara no válido durante la función de sobrescritura. El bloque que lo sustituye se carga en la RAM. Así se crean vacíos en la memoria de carga o en la de trabajo. Si, a causa de dichos vacíos, no es posible cargar más bloques nuevos, se deberá comprimir la memoria.

---

#### Nota

Cuando retorne la tensión después de un corte de alimentación, si la RAM no tiene respaldo, o bien, después de un borrado total de la CPU, los "antiguos" bloques de la EPROM serán los que se admitan como válidos y se cargarán.

---

## 17.1.6 Cargar mediante Memory Cards EPROM

### Requisitos

Para poder acceder en el sistema de origen a Memory Cards EPROM previstas para un sistema de destino S7, se necesitan los correspondientes drivers EPROM. Para poder acceder a Memory Cards EPROM previstas para un sistema de destino M7 se deberá haber instalado el Flash File System (ésto sólo es posible en las PGs 720, 740 y 760). Los drivers EPROM y el Flash File System se ofrecen opcionalmente al instalar el paquete básico de STEP 7. Si utiliza un PC, para guardar en la Memory Cards EPROM necesita además un Prommer externo.

Los drivers también se pueden instalar posteriormente. A tal efecto, abra el cuadro de diálogo correspondiente, partiendo de la barra de inicio (**Inicio > Simatic > STEP 7 > Parametrizar Memory Card**) o del Panel de control (haciendo doble clic en el icono "Parameterizar Memory Card").

### Guardar en una Memory Card

Para guardar bloques o programas de usuario en una Memory Card, proceda de la forma siguiente:

1. Enchufe la Memory Card (tarjeta de memoria en formato de tarjeta de crédito) en la correspondiente ranura del sistema de origen.
2. Abra la ventana "Memory Card S7"
  - haciendo clic en el botón de las "Memory Cards S7" de la barra de herramientas. En caso necesario, active la visualización de la misma con el comando Ver > Barra de herramientas o
  - Eligiendo el comando de menú **Archivo > Memory Card S7 > Abrir**.
3. Abra o active una ventana visualizando los bloques que desea guardar. Puede escoger entre una de las ventanas siguientes:
  - ventana de proyecto, vista "ONLINE"
  - ventana de proyecto, vista "OFFLINE"
  - ventana de librería
  - ventana "Estaciones accesibles"
4. Seleccione la carpeta "Bloques" o los bloques que desea guardar y cópielo(s) en la ventana "Memory Card S7".
5. Si en la Memory Card ya existe un bloque, aparecerá un mensaje de error. En este caso, borre el contenido de la misma y repita el proceso a partir del 2º paso.

## 17.1.7 Cargar la configuración del hardware y la de los enlaces por separado

### 17.1.7.1 Cargar una configuración en un sistema de destino

#### Consejo

Antes de cargar debería comprobar que la configuración de su equipo no contenga errores. Para ello utilice el comando de menú **Equipo > Comprobar coherencia**. STEP 7 comprobará entonces si se pueden crear datos de sistema cargables a partir de la configuración actual. Al comprobar la coherencia se le mostrarán los errores encontrados en una ventana aparte.

#### Requisitos para la carga

- La unidad de programación está conectada al interface MPI de la CPU a través de un cable MPI
- Si la instalación está conectada a una red (unidad de programación conectada a una subred:  
Cada uno de los módulos de una subred tienen que tener una dirección de estación diferente y la configuración real tiene que coincidir con la configuración de red creada.
- La configuración creada corresponde a la configuración real del equipo.  
Únicamente si la configuración es coherente y no contiene errores, se podrá cargar en el equipo. Sólo entonces será posible crear bloques de datos del sistema (SDBs) que, a su vez, se cargarán en los módulos.
- Si la configuración del equipo contiene módulos configurados y parametrizados con un paquete de software opcional: el software opcional tiene que estar instalado con una autorización.

#### Procedimiento

- elija el comando de menú **Sistema de destino > Cargar en módulo**  
STEP 7 le conducirá al resultado a través de diversos cuadros de diálogo.

La configuración del sistema de automatización se carga entera en la CPU. Los parámetros de la CPU tendrán efecto inmediato, en tanto que los parámetros de los demás módulos se transferirán a los mismos durante el arranque.

---

#### Nota

No es posible cargar en el equipo partes de la configuración, como p. ej. la configuración de distintos bastidores. Por razones de coherencia, STEP 7 carga siempre la configuración entera en el equipo.

---

## Cambiar el estado operativo de la CPU durante la carga

Con el comando **Sistema de destino > Cargar en módulo** podrá ejecutar - guiado por menús - las siguientes acciones desde la PG:

- Conmutar la CPU a STOP  
(si el selector de modo no está en RUN-P o si el enlace con la CPU ha sido legitimado mediante contraseña)
- Comprimir la memoria  
(si no se dispone de suficiente memoria libre junta)
- Volver a conmutar la CPU a RUN

### 17.1.7.2 Cargar por primera vez la configuración de la red

Antes de cargar la configuración por primera vez, los módulos conectados a la subred no poseen todavía la dirección de estación configurada para ellos, sino la dirección predeterminada. Para que su red pueda funcionar correctamente es preciso que todas las estaciones de la subred tengan direcciones diferentes.

- Subred MPI con conexión a través de la CPU  
Las CPUs se suministran con la dirección 2 que es la preajustada. Puesto que dicha dirección de estación se puede utilizar una sola vez, es necesario modificar la dirección preajustada de las demás CPUs.
- Subredes PROFIBUS e Industrial Ethernet  
Los CPs de los equipos que funcionan en estas subredes tienen que ser configurados y además es necesario asignarles direcciones de estación. La dirección se ha de asignar siempre a través del interface MPI del equipo para que se puedan realizar procesos de carga y de comunicación a través de la subred (para más información al respecto consulte los manuales SIMATIC NET, NCM S7 para PROFIBUS y NCM para Industrial Ethernet).

### Si la estación de la red no es un equipo S7...

Si la estación de la red no es un equipo S7, tiene que ajustar las propiedades de la red y de la estación con la herramienta prevista al efecto o con los interruptores previstos. Este es el caso p. ej. de los esclavos DP cuya dirección PROFIBUS se ha de ajustar mediante interruptores.

Asegúrese de que estos ajustes coincidan con los ajustes de los objetos de la representación gráfica de la red (PG/PC, Otro equipo, Equipo S5).

### Cambie la dirección PROFIBUS de los esclavos DP

Los esclavos conectados a la subred PROFIBUS tienen que poseer a su vez una dirección PROFIBUS unívoca. Si el esclavo DP a conectar soporta la función "Set\_Slave\_Add" (p. ej. ET 200C), puede asignar la dirección con STEP 7:

Tanto en el Administrador SIMATIC y como en la configuración del hardware las direcciones PROFIBUS se asignan con el comando de menú **Sistema de destino > Asignar dirección PROFIBUS**.

**Consejo:** Si no está seguro de que las direcciones actualmente asignadas sean del todo correctas, debería conectar los esclavos DP uno por uno a la PG/PC y cambiar su dirección.

## Cambiar la dirección de estación de equipos S7

Para cambiar la dirección de estación preajustada para los equipos S7, proceda del siguiente modo:

1. Configure el equipo; ajuste en la ficha "General" la dirección de estación (botón de comando "Propiedades" bajo "Interface") del módulo conectado (p. ej. de una CPU).
2. Cambie el módulo a STOP y conecte la unidad de programación al interface del módulo con un cable.
3. Determine la dirección de estación predeterminada del módulo conectado (p. ej. con el comando de menú **Sistema de destino > Mostrar estaciones accesibles** en el Administrador SIMATIC).
4. Cargue la configuración con la nueva dirección de estación en el sistema de destino (es decir en el módulo conectado):
  - En la ventana de equipos (configurar el hardware) con el comando de menú **Sistema de destino > Cargar en módulo**
  - Seleccione en la representación de la red (NetPro) el equipo que desea cargar y elija el comando de menú **Sistema de destino > Cargar > Equipos seleccionados**. Introduzca la "antigua" dirección (que aún sigue siendo válida).

### 17.1.7.3 Cargar la configuración de la red en un sistema de destino

#### Requisito

En adelante suponemos que el proyecto ya está completamente configurado, lo que significa que:

- todos los equipos están configurados
- todas las subredes están creadas y sus propiedades están definidas
- los enlaces están configurados (si es necesario)
- el interface PG/PC está ajustado de manera que es posible comunicar la PG/PC con el sistema de automatización a través de la subred conectada.
- la configuración es coherente (tras comprobarla)

Sólo cuando una configuración no contenga errores, es decir, cuando todos los módulos conectados a una misma red tengan direcciones de estación diferentes y su configuración real corresponda a la configuración teórica (creada mediante el software), será posible cargar la configuración a través de la subred (PROFIBUS o MPI) en los sistemas de destino.

### 17.1.7.4 Cargar modificaciones de la configuración de red

#### Requisitos

Todos los módulos que integran una subred tienen direcciones de estación de red diferente y su configuración real coincide con la configuración de red que ha creado.

Si desea conectar un nuevo equipo a una subred y la dirección de estación preajustada ya figura en la red, debe proceder como se describe en el apartado "Primera carga".

#### ¿Qué se carga y dónde?

Tras compilar la configuración de la red (comando de menú **Red > Guardar y compilar**) o **Sistema de destino > Cargar > ...** NetPro crea bloques de datos de sistema (SDBs) para módulos que pueden interpretar los datos en los SDBs. Los SDBs pueden contener tablas de enlaces, direcciones de estación, propiedades de subredes, direcciones de entrada/salida y parametrizaciones de tarjetas.

Dependiendo del comando de menú que elija para cargar la configuración, se cargarán distintos contenidos o en distintos sistemas de destino.

#### Nota

Sólo la opción **Cargar > Enlaces y transiciones de red** permite cargar con las CPUs en estado operativo RUN-P. Con las restantes opciones es preciso conmutar la CPU a STOP.

Comando de menú Sistema de destino > Cargar >	¿Qué se carga y dónde?	¿Dónde?
... Equipos seleccionados	Tablas de enlaces, direcciones de estaciones, propiedades de subredes, direcciones de entrada/salida y parametrización de tarjetas de los equipos seleccionados	En los equipos seleccionados
... Equipos seleccionados y equipos interlocutores	Tablas de enlaces, direcciones de estaciones, propiedades de la red, direcciones de entrada/salida y parametrizaciones de tarjetas del equipo seleccionado y el interlocutor del equipo seleccionado	En el equipo seleccionado y en todos aquellos equipos que sean interlocutores de este equipo
... Equipos conectados a la subred	Tablas de enlaces, direcciones de estaciones, propiedades de subredes, direcciones de entrada/salida y parametrización de tarjetas	Sucesivamente en todos los equipos de la subred seleccionada
... Enlaces seleccionados	Enlaces seleccionados (posibilidad de elegir varios a la vez)	En el equipo local y (en el caso de enlaces bilaterales) en sus interlocutores
... Enlaces y transiciones de la red	Enlaces (también puede ser una tabla de enlaces vacía) e información de transiciones de la red	En el módulo seleccionado (posible en el estado operativo RUN-P)

## Procedimiento

1. Conecte la PG a la subred a la que también está conectada la estación que desea cargar
2. Abra NetPro.
3. Seleccione el equipo que desea cargar o la subred (con..**Cargar > Subred seleccionada**) en la representación gráfica de la red.
4. Elija una de las opciones descritas del comando de menú **Sistema de destino > Cargar**.

## Más información

Encontrará más información sobre los comandos de menú respecto a Cargar en la ayuda contextual (seleccionar el comando de menú y pulsar la tecla F1).

### 17.1.7.5 Cargar la configuración de datos globales

Al compilar, los datos de la tabla de datos globales se transforman en datos de sistema. Si después de la compilación no se ha visualizado ningún error, los datos de sistema pueden transferirse a las CPUs.

- Seleccione el comando de menú **Sistema de destino > Cargar**.

## 17.2 Cargar en la PG desde el sistema de destino

### 17.2.1 Cargar en la PG desde el sistema de destino

Esta función le asiste en las siguientes acciones:

- Guardar informaciones del sistema de destino (p.ej. para fines de mantenimiento)
- Configuración y edición rápida de un equipo cuando los componentes del hardware ya están disponibles al empezar a configurar.

### Salvaguardar informaciones del sistema de destino

Esta medida puede ser necesaria, p. ej. cuando los datos del proyecto offline de la versión que se esté ejecutando en la CPU no existan o existan sólo en parte. En este caso puede acceder por lo menos a los aquella parte de los datos del proyecto a la que se puede acceder online en su PG.

### Configuración rápida

Para editar con mayor facilidad la configuración de su equipo puede cargar los datos de configuración del sistema de destino en su PG tras haber montado el hardware y haber realizado un rearranque completo del equipo. Entonces obtendrá la configuración del equipo con las indicaciones de tipo de los distintos módulos. Después basta que especifique (referencia) y parametrice los distintos módulos.

En la PG se cargan las siguientes informaciones:

- S7-300: configuración para el bastidor central y posibles bastidores de ampliación
- S7-400: configuración para el bastidor central con una CPU y módulos de seal sin bastidores de ampliación
- Los datos de configuración para la periferia descentralizada no se puede cargar en la PG.

Este volumen de información se carga cuando el sistema de destino no contiene datos de configuración, p.ej. tras realizar un borrado total del sistema. Por lo demás, la función "Cargar en la PG" da mejores resultados.

En los sistemas S7-300 sin periferia descentralizada sólo tiene que especificar (referencia) y parametrizar los módulos.

---

#### Nota

Al cargar en la PG (sin que exista una configuración offline), STEP 7 no puede determinar todas las referencias de los componentes.

Las referencias que no aparecen se pueden introducir al configurar el hardware con el comando de menú **Herramientas >Especificar módulo**. De esta manera se pueden parametrizar módulos desconocidos por STEP 7 (es decir, que no se parametrizan en la ventana "Catálogo de hardware"); sin embargo no se verifica si se respetan las reglas de parametrización.

---

## Limitaciones al cargar desde el sistema de destino

Para los datos que se cargan en la PG desde el sistema de destino rigen las siguientes limitaciones:

- Los bloques no contienen nombres simbólicos para parámetros, variables y marcas.
- Los bloques no contienen comentarios.
- En la PG se carga todo el programa con todos los datos del sistema. Sin embargo, sólo se pueden seguir editando de forma habitual aquellos datos del sistema que son parte de "HW-Config: Configurar el hardware".
- Los datos relacionados con los temas "Comunicación mediante datos globales (GD)", "Configuración de mensajes de símbolos" ya no se pueden modificar.
- Las peticiones de forzado permanente no se cargan en la PG. Estas peticiones se tienen que guardar por separado en forma de tabla de variables (VAT) en el diálogo de la petición.
- Los comentarios que se hagan en los diálogos de los módulos no se cargan.
- Los nombres de los módulos sólo se visualizan si ha seleccionado la opción correspondiente en la configuración (HW-Config: Herramientas > Preferencias: "Guardar nombre del objeto en sistema de destino" )

## 17.2.2 Cargar un equipo en la PG

Utilice el comando de menú **Sistema de destino > Cargar equipo en PG** para cargar en la PG la configuración actual y todos los bloques del sistema de automatización a elegir.

A tal efecto, STEP 7 crea un nuevo equipo en el proyecto actual, bajo el cual se guarda la configuración. El nombre estándar del equipo insertado se puede cambiar (p.ej.. "Equipo SIMATIC 300(1)"). El equipo insertado se visualiza tanto en la vista "online" como en la vista "offline".

Para elegir este comando de menú tiene que estar abierto un proyecto. No es necesario seleccionar un objeto en la ventana de proyecto o en la vista (online u offline) para ejecutar el comando de menú.

Utilice esta función para que la configuración resulte más fácil.

- En los sistemas de destino S7-300 se carga la configuración del sistema de automatización utilizado incluidos los bastidores de ampliación pero sin la periferia descentralizada (DP).
- En los sistemas de destino S7-400 se carga la configuración del bastidor sin bastidores de ampliación ni periferia descentralizada.

En los sistemas S7-300 sin periferia descentralizada basta con especificar los módulos (indicando su referencia) y parametrizarlos.

## Limitaciones en cuanto a la carga de un equipo en la PG

Para los datos cargados en la PG rigen las siguientes limitaciones:

- Los bloques no contienen nombres simbólicos para parámetros formales, variables temporales y marcas.
- Los bloques no contienen comentarios.
- En la PG se carga todo el programa con todos los datos del sistema. Sin embargo, no todos los datos pueden ser procesados por la misma.
- Los datos relacionados con los temas "Comunicación mediante datos globales (GD)", "Configurar mensajes de símbolos" y "Configurar redes" ya no se pueden modificar.
- Las peticiones de forzado permanente no se cargan en la PG.

## 17.2.3 Cargar bloques desde una CPU S7

Con el Administrador SIMATIC puede cargar bloques S7 de la CPU en el disco duro del sistema de origen. La posibilidad de cargar bloques en la PG desde la CPU puede servirle, p.ej., para:

- hacer una copia de seguridad del programa de usuario que está cargado actualmente en la CPU. Por ejemplo, después de trabajos de mantenimiento, como p.ej. después de un borrado total de la CPU, el personal encargado puede transferir nuevamente dicha copia de seguridad.
- Además, puede cargar el programa de usuario de la CPU en el sistema de origen y editarlo allí, como p.ej. para buscar errores a efectos de mantenimiento. En este caso, no dispone de símbolos ni de comentarios para documentar el programa. Por lo tanto, este procedimiento se recomienda sólo a efectos de mantenimiento.

## 17.2.4 Editar los bloques cargados en la PG/en el PC

La posibilidad de editar bloques en la PG ofrece las siguientes ventajas:

- Corregir el bloque directamente en la CPU durante la fase de test y documentar el resultado.
- Cargar en la PG el contenido actual de los bloques de la memoria de carga RAM de la CPU.

---

### Nota

#### *Conflicto de la indicación de hora y fecha al trabajar online y offline*

Las acciones descritas a continuación pueden causar conflictos de la indicación de hora y fecha, por lo que deberán evitarse.

Dichos conflictos se presentan al abrir un bloque online si

- los cambios efectuados online no se han guardado offline en el programa de usuario S7
- los cambios efectuados offline no se han cargado en la CPU.

Dichos conflictos se presentan al abrir un bloque offline si

- un bloque online con conflicto de indicación de hora y fecha se copia offline en el programa de usuario S7, abriéndose luego el bloque offline.
- 

## Diferenciación entre dos casos

Al cargar bloques de la CPU en la unidad de programación es preciso diferenciar entre dos casos:

1. el programa de usuario al que pertenecen los bloques se encuentra en la unidad de programación.
2. el programa de usuario al que pertenecen los bloques no se encuentra en la unidad de programación.

Por consiguiente, no se dispone de las partes del programa que no se pueden cargar en la CPU. Estas son:

- la tabla de símbolos con los nombres simbólicos de los operandos y los comentarios,
- los comentarios de los segmento de un programa FUP o KOP,
- los comentarios de las líneas de un programa AWL y
- tipos de datos de usuario,

## 17.2.5 Cargar nuevamente la configuración del hardware y la de los equipos

### 17.2.5.1 Cargar la configuración desde otro equipo

#### Requisito

Se tiene que haber conectado la unidad de programación al interface MPI de la CPU mediante un cable MPI.

#### Consejos

Cargue equipos en un proyecto vacío recién creado.

Los equipos que dependen de alguna manera de otros equipos (esclavo I de un equipo maestro DP, receptor y emisor de una configuración con comunicación directa/cruzada) deberían cargarse siempre juntos en un proyecto. Motivo: sin el "partner" de este equipo, el proyecto será incoherente.

#### Procedimiento

1. Elija el comando de menú **Sistema de destino > Cargar en la PG**  
Aparece el cuadro de diálogo desde el cual se abre la configuración.
2. Elija el proyecto en el que desee guardar la configuración posteriormente y confirme haciendo clic en "Aceptar".
3. En el cuadro de diálogo que aparece entonces, ajuste la dirección de estación, el número de bastidor y el slot del módulo cuya configuración se vaya a leer (por lo general, la CPU). Confirme haciendo clic en "Aceptar".

Para adjudicar un nombre a dicha configuración, elija el comando **Equipo > Propiedades** y guárdela luego en el proyecto predeterminado (con el comando **Equipo > Guardar**).

### 17.2.5.2 Cargar una configuración de red en la PG

#### Introducción

Existe la posibilidad de cargar en la PG equipo por equipo la configuración real de red de un proyecto.

Por una parte puede cargar desde el Administrador SIMATIC equipo por equipo toda la configuración de un proyecto en la PG (comando de menú **Sistema de destino > Cargar en PG**). En este caso, STEP 7 creará en el proyecto actual un objeto "Equipo" para cada equipo que se desee cargar.

Por otra parte, al configurar el hardware es posible cargar la configuración de un equipo (comando de menú **Sistema de destino > Cargar en PG**).

A continuación explicaremos cómo se carga con NetPro toda la configuración de la red equipo por equipo.

## Requisitos

La PG (o el PC) tiene que estar conectada a la misma subred que los equipos que desea cargar, o bien los equipos son accesibles a través de transiciones de la red. Se conocen las direcciones de estación y los bastidores/slots de los módulos conectados.

## Procedimiento

1. Conecte la PG a la subred a la que también está conectada la estación que desea cargar.
2. Si es necesario, cree un proyecto para la configuración de red cargada.
3. Abra NetPro a través de un proyecto en el que guardará más adelante la configuración de red cargada (p. ej. desde un nuevo proyecto).
4. Elija el comando de menú **Sistema de destino > Cargar equipo en PG**  
Para poder ejecutar este comando de menú tiene que estar abierto un proyecto.
5. Introduzca en el siguiente cuadro de diálogo el equipo que desea cargar indicando su dirección de estación así como el bastidor y el slot.  
El objeto "Equipo" aparece en la representación gráfica de la red con todos los módulos que tienen una conexión de red. También se indican las subredes a las que está conectado el equipo. El nombre (del equipo) dado por el sistema se puede cambiar con el comando de menú **Edición > Propiedades del objeto**.  
También se cargan los enlaces configurados. Para verlos seleccione un módulo que sea punto final de un enlace.
6. También es posible cambiar la configuración del equipo o los enlaces y cargar después los cambios en el equipo. Si los enlaces han sido creados con paquetes opcionales, tiene que estar instalado el paquete opcional para que se puedan editar estos enlaces y volver a cargarlos en el equipo.
7. Proceda como arriba descrito hasta haber cargado todos los equipos deseados.
8. Si lo desea puede guardar la configuración de la red en el proyecto actual (comando de menú **Red > Guardar** o **..> Guardar y compilar**).

## Particularidades de enlaces que se cargaron en la PG

En la tabla de enlaces falta el interlocutor que se ha configurado offline, por lo que éste está "sin especificar". Si necesita información sobre su direccionamiento vaya al cuadro de diálogo que sigue al diálogo de propiedades.

STEP 7 no puede determinar siempre el sentido de la comunicación de enlaces PtP; sin embargo le indica los sentidos posibles.

## 17.2.6 Borrar en el sistema de destino

### 17.2.6.1 Borrar las memorias de carga/de trabajo y borrado total de la CPU

Antes de cargar su programa de usuario en el sistema de destino S7 es recomendable que lleve a cabo un borrado total de la CPU para garantizar que en la CPU no se encuentren más bloques "antiguos".

#### Requisito para el borrado total

Para poder llevar a cabo un borrado total, la CPU deberá estar en estado operativo STOP (selector en STOP o en RUN-P y cambio del estado operativo a STOP mediante el comando **Sistema de destino > Estado operativo**).

#### Borrado total de CPUs S7

En un borrado total de una CPU S7 sucede lo siguiente:

- La CPU se pone a cero.
- Se borran todos los datos de usuario (bloques y SDBs, exceptuando los parámetros del interface MPI)
- La CPU deshace todos los enlaces existentes.
- Si existen datos en una EPROM (Memory Card o EPROM integrada), la CPU vuelve a copiar su contenido en la memoria RAM después del borrado total.

Se conservan el contenido del búfer de diagnóstico y los parámetros del interface MPI.

#### Borrado total de CPUs/FMs M7

En un borrado total de una CPU M7 o de un FM M7 sucede lo siguiente:

- Se restablece el estado inicial.
- Se borran los SDBs, exceptuando los parámetros del interface MPI.
- La CPU/FM deshace todos los enlaces existentes. Los programas de usuario no se pierden y siguen corriendo tras cambiar de STOP a RUN.

Si se producen errores muy graves, con la función Borrado total puede restablecer el estado que tenía la CPU/el FM M7 en su origen. Para ello borre de la memoria de trabajo los bloques de datos de sistema (SDB) actuales y vuelva a cargar los SDBs que se encuentran en la memoria ROM. En algunos casos es necesario además un rearranque (arranque en caliente) del sistema operativo. Para ello debe borrar por completo el M7 con el selector de modos de operación (posición MRES). Con el selector de modos de operación de las CPUs/de los FMs SIMATIC M7 sólo se podrá efectuar un reset si se utiliza el sistema operativo RMOS32.

### 17.2.6.2 Borrar bloques S7 en el sistema de destino

Durante la fase de comprobación del programa de CPU puede ser necesario borrar bloques en la CPU. Los bloques están guardados en la memoria de usuario de la CPU bien sea en la EPROM, o bien, en la RAM (dependiendo de la CPU y del proceso de carga).

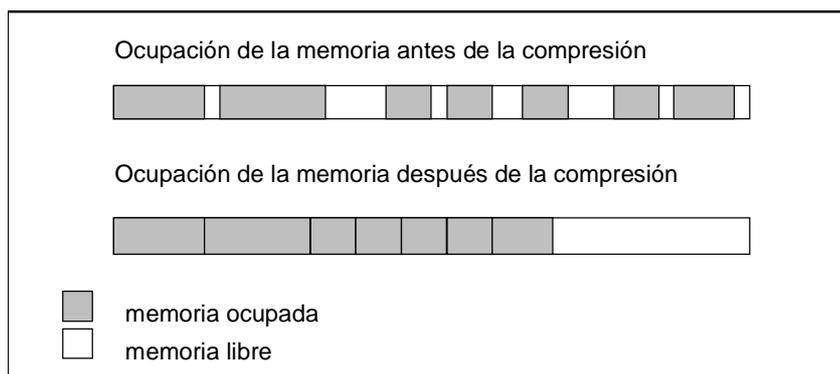
- Los bloques que estén cargados en la RAM se pueden borrar directamente. El espacio de memoria ocupado se libera en las memorias de carga y de trabajo.
- Los bloques que estén guardados en la EPROM integrada se copian en la memoria RAM siempre después de haberse borrado totalmente la CPU. Las copias pueden borrarse directamente en la memoria RAM. En la EPROM, los bloques borrados son declarados no válidos hasta el próximo borrado total o hasta el próximo corte de tensión con memoria RAM no respaldada. Después de un corte de tensión, si la memoria RAM no tiene respaldo, o bien, después de un borrado total de la CPU, los bloques "borrados" se copian nuevamente de la EPROM en la RAM, siendo entonces los bloques activos. Los bloques contenidos en la EPROM integrada (p.ej. en el caso de la CPU 312) se borran al sobrescribirse con el nuevo contenido de la RAM.
- Las Memory Cards EPROM se deben borrar en el sistema de origen.

## 17.2.7 Comprimir la memoria de usuario (RAM)

### 17.2.7.1 Creación de huecos en la memoria de usuario (RAM)

Al borrar y sobrescribir bloques con frecuencia, se forman huecos en la memoria de usuario (memorias de carga y de trabajo) que reducen el espacio de memoria aprovechable. Comprimiendo la memoria de usuario es posible repositionar los bloques existentes uno junto a otro, creándose un espacio contiguo de memoria libre.

En la figura siguiente se muestra de forma esquemática cómo se desplazan los bloques de memoria ocupados al activar la función "Comprimir memoria".



### En lo posible, se deberá comprimir en estado operativo "STOP".

Para poder rellenar todos los huecos de la memoria es preciso comprimirla en el estado operativo "STOP". Si se comprime en modo RUN-P (posición del selector de modos de operación), no se desplazan los bloques que se están ejecutando actualmente, puesto que están abiertos. En modo RUN (posición del selector de modos de operación) no es posible llevar a cabo la compresión (debido a la protección contra escritura).

## 17.2.7.2 Comprimir el contenido de la memoria de una CPU S7

### Alternativas para comprimir

Hay dos alternativas para comprimir la memoria de usuario:

- Si al cargar falta espacio de memoria en el sistema de destino, aparece un cuadro de diálogo donde se indica el error. Haciendo clic en el correspondiente botón es posible comprimir la memoria.
- Como medida preventiva puede visualizar el grado de utilización de la memoria (elija a tal efecto el comando de menú **Sistema de destino > Información del módulo.../ficha "Memoria"**) y, en caso necesario, iniciar la compresión.

### Procedimiento

1. Seleccione el programa S7 en la vista online o en la ventana "Estaciones accesibles".
2. Elija el comando **Sistema de destino > Información del módulo**.
3. En el cuadro de diálogo siguiente elija la ficha "Memoria", donde encontrará el botón de comando "Comprimir" para comprimir la memoria en el caso de que la CPU soporte esta función.

# 18 Test con la tabla de variables

## 18.1 Introducción al test con tablas de variables

Las tablas de variables ofrecen la ventaja de que pueden guardar distintos entornos de test. Así se pueden reproducir sin esfuerzo tests y observaciones durante la puesta en servicio o para operaciones de servicio técnico o de mantenimiento. El número de tablas de variables que se pueden almacenar es ilimitado.

Para comprobar el programa con tablas de variables se dispone de las siguientes funciones:

- **Observar variables**  
Esta función permite visualizar en la PG/el PC los valores actuales de determinadas variables de un programa de usuario o de una CPU.
- **Forzar variables**  
Esta función permite asignar valores fijos a determinadas variables de un programa de usuario o de una CPU. El test con el estado del programa posibilita asimismo un forzado único e inmediato.
- **Desbloquear salidas y Activar valores forzados**  
Ambas funciones permiten asignar valores fijos a determinadas salidas de la periferia de una CPU en el estado operativo STOP.
- **Forzar variables de forma permanente**  
Utilice esta función para asignar valores permanentes (fijos) a las variables de un programa de usuario o de una CPU de manera que el programa de usuario no los pueda sobrescribir.

Puede hacerse mostrar los valores de las siguientes variables:

- Entradas, salidas, marcas, temporizadores y contadores
- Contenido de bloques de datos
- Periferia

Elabore una tabla con aquellas variables cuyos valores desee visualizar o forzar.

Definiendo el punto de disparo y la condición de disparo se define cuándo se deben observar o sobrescribir las variables y con qué frecuencia.

## 18.2 Procedimiento básico para observar y forzar con tablas de variables

Para poder utilizar las funciones **Observar** y **Forzar**:

1. Cree una tabla de variables nueva o abra una ya existente.
2. Edite o compruebe la tabla de variables.
3. Con el comando **Sistema de destino > Establecer enlace con**, establezca un enlace online entre la tabla de variables actual y la CPU deseada.
4. Con el comando **Variable > Condiciones de disparo**, elija un punto de disparo apropiado y ajuste la condición de disparo.
5. Con los comandos **Variable > Observar** y **Variable > Forzar** es posible activar o desactivar la función en cuestión.
6. Guarde la tabla de variables creada utilizando los comandos **Tabla > Guardar** o **Tabla > Guardar como** para poder llamarla de nuevo en cualquier momento.

## 18.3 Editar y guardar tablas de variables

### 18.3.1 Crear y abrir una tabla de variables

Antes de poder observar o forzar variables, tiene que crear una tabla de variables (VAT) e introducir las variables deseadas. Para crear una tabla de variables puede elegir una de las alternativas siguientes:

#### En el Administrador SIMATIC:

- Seleccione la carpeta "Bloques" y elija el comando de menú **Insertar > Bloque S7 > Tabla de variables**. En el cuadro de diálogo que aparece a continuación puede asignar un nombre a la tabla (cuadro de texto "Nombre simbólico"). Este nombre aparecerá en la ventana de proyecto. Para abrir la tabla de variables, haga doble clic en el objeto.
- Seleccione un enlace de la lista de las estaciones accesibles o un programa S7 o M7 de la vista online. Con el comando **Sistema de destino > Observar/forzar variable** se puede crear una tabla de variables sin nombre.

#### En "Observar/forzar variables":

- Eligiendo el comando **Tabla > Nueva** puede crear una tabla que no esté asociada todavía a ningún programa S7/M7. Para acceder a las tablas ya existentes, elija el comando **Tabla > Abrir**.
- Puede usar los botones de la barra de herramientas para crear o abrir tablas de variables.

La tabla de variables se puede guardar, imprimir y utilizar en repetidas ocasiones para observar y forzar.

### 18.3.2 Copiar/mover tablas de variables

Las tablas de variables se pueden copiar o mover en las carpetas de bloques de programas S7/M7.

Al copiar y mover tablas de variables debe tener en cuenta los siguientes puntos:

- En la tabla de símbolos del programa de destino se actualizan los símbolos que ya figuran en el programa.
- Al mover una tabla de variables, también se mueven los símbolos de la tabla del programa de origen a la tabla de símbolos del programa de destino.
- Al borrar la tabla de variables de una carpeta de bloques, también se borran los símbolos correspondientes de la tabla de símbolos del programa S7/M7.
- Si el programa de destino ya contiene una tabla de variables con el mismo nombre, al copiar se dispone de la posibilidad de cambiarle el nombre a la tabla de variables (de forma predeterminada se le añade un número al nombre existente).

### 18.3.3 Guardar una tabla de variables

Puede guardar esta tabla de variables para poderla utilizar en un nuevo test de su programa para observar y forzar variables.

1. Guarde la tabla de variables con el comando de menú **Tabla > Guardar**.
2. Si se ha generado una nueva tabla de variables debe indicar un nombre para la tabla, p.ej., "Test\_1".

Al guardar la tabla de variables se almacenan todos los ajustes actuales, así como el formato de la tabla. Por tanto, se guardarán todos los ajustes efectuados mediante el comando de menú "Condiciones de disparo".

## 18.4 Introducir variables en tablas de variables

### 18.4.1 Insertar operandos o símbolos en una tabla de variables

Defina las variables cuyos valores desea predeterminedar u observar, e introdúzcalas en la tabla de variables. Al hacerlo proceda de "fuera" a "dentro", es decir, seleccione primero las entradas y después las variables sobre las que influyen las entradas, o que influyen en las salidas, y finalmente seleccione las salidas.

Por ejemplo, si desea observar el bit de entrada 1.0, la palabra de marcas 5 y el byte de salidas 0, en la columna de operando introduzca lo siguiente:

**Ejemplo:**

E 1.0  
MW 5  
AB 0

#### Ejemplo de una tabla de variables rellena

La figura siguiente muestra una tabla de variables con las siguientes columnas: operando, símbolo, formato de visualización, valor de estado y valor de forzado

	Operando	Símbolo	Formato de vi	V. de estad	V. de fuerza
1	//OB1 Segmento 1				
2	E 0.1	"Pulsador 1"	BOOL	true	
3	E 0.2	"Pulsador 2"	BOOL	true	
4	A 4.0	"Lampara verde"	BOOL	false	
5	//OB1 Segmento 3				
6	E 0.5	"Modo Autom. ON"	BOOL	true	
7	E 0.6	"Modo Manual ON"	BOOL	true	
8	A 4.2	"Modo automático"	BOOL	true	true
9	//OB1 Llamada FB1 para encender motor de gasolina				
10	E 1.0	"Encender_MG"	BOOL	false	
11	E 1.1	"Apagar_MG"	BOOL	false	
12	E 1.2	"Fallo_MG"	BOOL	false	
13	A 5.1	"Vel_teor_alcanzada"	BOOL	false	
14	A 5.0	"MG_ON"	BOOL	X	X true
15	//OB1 Llamada FB1 para encender motor diese				
16	E 1.4	"Encender_MD"	BOOL	false	
17	E 1.5	"Apagar_MD"	BOOL		

MPI = 3 (directo) Run

## Notas para insertar símbolos

- La variable a forzar se indica con su operando o con su nombre simbólico. Los símbolos y los operandos se pueden registrar en las columnas denominadas "Símbolo" y "Operando". El registro se escribe automáticamente en la columna adecuada. Si se ha definido el símbolo correspondiente en la tabla de símbolos, la entrada en la columna "Símbolo" u "Operando" se actualiza automáticamente.
- Solamente se pueden registrar aquellos símbolos que ya hayan sido definidos en la tabla de símbolos.
- Cuando introduzca el símbolo regístrelo exactamente igual que como está definido en la tabla de símbolos.
- Los nombres simbólicos que contengan caracteres especiales tienen que escribirse entre comillas (p.ej.: "Motor.on", "Motor+on", "Motor-on").
- Si quiere definir nuevos símbolos en la tabla de símbolos, seleccione el comando **Herramientas > Tabla de símbolos**. También puede copiar los símbolos en la tabla de símbolos y pegarlos en una tabla de variables.

## Comprobación de sintaxis

Al introducir las variables en la tabla, la sintaxis se comprueba antes de abandonar cada línea. Las entradas erróneas se destacan en rojo. Si se coloca el cursor en una línea marcada en rojo, en la barra de estado se puede leer la causa del error. Pulsando la tecla F1 se obtendrá información acerca de cómo subsanar el mismo.

## Tamaño máximo

El tamaño máximo de una tabla es de 1024 líneas y cada línea puede incluir como máximo 255 caracteres. Si se sobrepasa el límite de caracteres el programa le impedirá el cambio a la línea siguiente.

### 18.4.2 Insertar valores de forzado

#### Valor de forzado como comentario

Si desea dejar sin efecto el "Valor de forzado" de una variable utilice el comando de menú **Variable > Valor de forzado como comentario**. Un carácter iniciador de comentario "/" antes del valor que va a forzarse de una variable indica que dicho valor pierde su validez. En lugar de la llamada del comando de menú también se puede iniciar el comentario con el carácter "/". También puede cambiarse la invalidez del "valor de forzado" activando de nuevo el comando de menú **Variable > Valor de forzado como comentario** o eliminando el carácter de comentario.

### 18.4.3 Valores máximos para introducir temporizadores

Al introducir los valores de los temporizadores, tenga en cuenta los siguientes valores máximos:

W#16#3999 (valor máximo en el formato BCD).

**Ejemplos:**

Operando	Formato de visualización	Introducción	Mostrar valor de forzado	Explicación
T 1	TIEMPO_SIMATIC	137	S5TIME#130MS	Conversión a milisegundos
MW 4	TIEMPO_SIMATIC	137	W#16#0089	Posibilidad de visualizar en formato BCD
MW 4	HEX	137	W#16#0089	Posibilidad de visualizar en formato BCD
MW 6	HEX	157	W#16#009D	Visualización en formato BCD imposible, por lo que no se puede seleccionar el formato de visualización TIEMPO_SIMATIC

**Nota**

- Si bien los valores de tiempo se pueden introducir tomando como unidad el milisegundo, el valor que se haya especificado será adaptado conforme a un determinado periodo. La magnitud de este periodo dependerá de la magnitud del valor de tiempo especificado (137 se redondea, resultando así 130 ms).
- Los valores de forzado de los operandos del tipo de datos WORD (p.ej.EW 1), se convierten al formato BCD. ¡No todas las configuraciones binarias son números BCD válidos! Si en el caso de un operando del tipo de datos WORD no es posible representar la entrada en forma de TIEMPO\_SIMATIC, se cambiará automáticamente al formato predeterminado (en este caso: HEX, véase "Elegir formato de estado", preajuste en el menú Ver), de manera que el valor introducido se pueda visualizar.

**Formato BCD para variables en formato TIEMPO\_SIMATIC**

Los valores de variables en formato TIEMPO\_SIMATIC se introducen en formato BCD. Los 16 bits tienen el siguiente significado:

| 0 0 x x | h h h h | z z z z | e e e e |

Bits 15 y 14 son siempre cero.

Bits 13 y 12 (señalados con xx) definen el multiplicador para los bits 0 a 11:

- 00 => Multiplicador 10 milisegundos
- 01 => Multiplicador 100 milisegundos
- 10 => Multiplicador 1 segundo
- 11 => Multiplicador 10 segundos

Bits 11 a 8 Centenas (hhhh)

Bits 7 a 4 Decenas (zzzz)

Bits 3 a 0 Unidades (eeee)

**18.4.4 Valores máximos para introducir contadores**

Al introducir los valores de los contadores, tenga en cuenta los siguientes límites:

Límites superiores para contadores: C#999  
W#16#0999 (valor máximo en el formato BCD)

**Ejemplos:**

Operando		Formato de visualización	Introducción	Mostrar valor de forzado	Explicación
C	1	COUNTER	137	C#137	Conversión
MW	4	COUNTER	137	C#89	Posibilidad de visualizar en formato BCD
MW	4	HEX	137	W#16#0089	Posibilidad de visualizar en formato BCD
MW	6	HEX	157	W#16#009D	Visualización en formato BCD imposible, por lo que no se puede seleccionar el formato de visualización CONTADOR

**Nota**

- Si introduce en un contador un número decimal y no identifica el valor con C#, dicho valor se convertirá automáticamente al formato BCD (137 se convierte en C#137).
- Los valores de forzado de los operandos del tipo de datos WORD (p.ej. EW 1), se convierten al formato BCD. ¡No todas las configuraciones binarias son números BCD válidos! Si en el caso de un operando del tipo de datos WORD no es posible representar la entrada en forma de COUNTER, se cambiará automáticamente al formato predeterminado (en este caso: HEX, véase "Elegir formato de estado", preajuste en el menú Ver), de manera que el valor introducido se pueda visualizar.

**18.4.5 Introducir líneas de comentario**

Las líneas de comentario se introducen con dos barras inclinadas "//".

Si desea dejar sin efecto una o varias líneas de la tabla de variables utilice el comando

**Edición > Desactivar línea** o el correspondiente botón  de la barra de herramientas.

## 18.5 Ejemplos

### 18.5.1 Ejemplo de introducción de operandos en tablas de variables

Operando permitido:	Tipo de datos:	Ejemplo (nematécnica SIMATIC):
Entrada   Salida   Marca	BOOL	E 1.0   A 1.7   M 10.1
Entrada   Salida   Marca	BYTE	EB 1   AB 10   MB 100
Entrada   Salida   Marca	WORD	EW 1   AW 10   MW 100
Entrada   Salida   Marca	DWORD	ED 1   AD 10   MD 100
Periferia (Entrada   Salida)	BYTE	PEB 0   PAB 1
Periferia (Entrada   Salida)	WORD	PEW 0   PAW 1
Periferia (Entrada   Salida)	DWORD	PED 0   PAD 1
Temporizadores	TIMER	T 1
Contadores	COUNTER	Z 1
Bloque de datos	BOOL	DB1.DBX 1.0
Bloque de datos	BYTE	DB1.DBB 1
Bloque de datos	WORD	DB1.DBW 1
Bloque de datos	DWORD	DB1.DBD 1

**Nota:**

Por razones de utilización interna no está permitida la entrada "DB0...".

**Ventana "Valores de forzado":**

Los módulos S7-300 solamente permiten un forzado permanente de entradas, salidas y periferia (salidas).

Los módulos S7-400 solamente permiten un forzado permanente de entradas, salidas, marcas y periferia (entradas/salidas).

## 18.5.2 Ejemplo de introducción de un área de operandos conexos

Abra una tabla de variables y visualice el cuadro de diálogo "Insertar área" con el comando de menú **Insertar > Área**.

Rellene el cuadro de diálogo indicando el número de líneas, a partir de qué operando y el formato en el que se introducirán en la tabla de variables, como por ejemplo:

- Desde operando: M 3.0
- Cantidad: 10
- Formato de visualización: BIN

Operando	Formato de estado
M 3.0	BIN
M 3.1	BIN
M 3.2	BIN
M 3.3	BIN
M 3.4	BIN
M 3.5	BIN
M 3.6	BIN
M 3.7	BIN
M 4.0	BIN
M 4.1	BIN

En este ejemplo puede observar cómo varía la denominación en la columna "Operando" después de la octava entrada.

## 18.5.3 Ejemplos de introducción de valores de forzado normal y de forzado permanente

### Operandos de bit

Operandos de bit posibles	Valores permitidos de forzado normal y forzado permanente
E1.0	True
M1.7	False
A10.7	0
DB1.DBX1.1	1
E1.1	2#0
M1.6	2#1

### Operandos de byte

Operandos de byte posibles	Valores permitidos de forzado / de forzado permanente
EB 1	2#00110011
MB 12	b#16#1F
MB 14	1F
AB 10	'a'
DB1.DBB 1	10
PAB 2	-12

### Operandos de palabra

Operandos de palabra posibles	Valores permitidos de forzado normal y forzado permanente
EW 1	2#0011001100110011
MW 12	w#16#ABCD
MW 14	ABCD
AW 10	b#(12,34)
DB1.DBW 1	'ab'
PAW 2	-12345
MW 3	12345
MW 5	s5t#12s340ms
MW 7	0.3s ó 0,3s
MW 9	c#123
MW 11	d#1990-12-31

### Operandos de palabra doble

Operandos de palabra doble posibles	Valores permitidos de forzado / de forzado permanente
ED 1	2#00110011001100110011001100110011
MD 0	1.23e4
MD 4	1.2
AD 10	dw#16#abcdef10
AD 12	ABCDEF10
DB1.DBD 1	b#(12,34,56,78)
PAD 2	'abcd'
MD 8	L# -12
MD 12	L#12
MD 16	123456789
MD 20	123456789
MD 24	T#12s345ms
MD 28	Tod#1:2:34.567
MD 32	p#e0.0

## Temporizador

Operandos posibles del tipo temporizador	Valores permitidos de forzado normal y forzado permanente	Explicación
T 1	0	Valor temporizado en milisegundos (ms)
T 12	20	Valor temporizado en milisegundos
T 14	12345	Valor temporizado en milisegundos
T 16	s5t#12s340ms	Valor temporizado 12s 340ms
T 18	1.3	Valor temporizado 1s 300 ms
T 20	1.3s	Valor temporizado 1s 300 ms

El forzado de un temporizador sólo afecta su valor, mas no su estado. Por tanto, el temporizador T1 se puede forzar al valor 0, pero el resultado lógico en U T1 no se modifica.

Las secuencias de caracteres "s5t" y "s5time" se pueden escribir tanto en mayúsculas como en minúsculas.

## Contadores

Operandos posibles del tipo contador	Valores permitidos de forzado / de forzado permanente
Z 1	0
Z 14	20
Z 16	c#123

El forzado de un contador sólo afecta a su valor, pero no a su estado. Por tanto, el contador T1 se puede forzar al valor 0, pero el resultado lógico en U Z1 no se modifica.

## 18.6 Establecer un enlace con la CPU

### 18.6.1 Establecer un enlace con la CPU

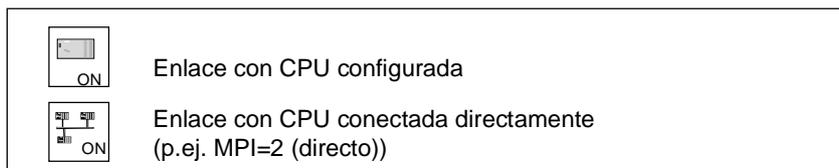
Para poder observar y forzar las variables que ha reunido en su tabla de variables (VAT) actual, debe establecer un enlace con la CPU correspondiente. Cualquier tabla de variables se puede enlazar con otra CPU.

#### Visualizar el enlace online

Si existe un enlace online, en la barra de título de la ventana de la tabla de variables aparecerá la palabra "ONLINE". En la barra de estado se muestran, dependiendo de la CPU utilizada, los estados operativos "RUN", "STOP", "DESCONECTADO" o "CONECTADO".

#### Establecer un enlace online con la CPU

Si no existe un enlace online con la CPU deseada, éste se deberá definir con el comando **Sistema de destino > Establecer enlace con > ...**, para poder observar o forzar las variables. Alternativamente es posible hacer clic en los correspondientes botones de la barra de herramientas:



#### Deshacer el enlace online con la CPU

Para deshacer un enlace online entre la tabla de variables y la CPU, elija el comando **Sistema de destino > Deshacer enlace**.

---

#### Nota

Si ha creado una tabla de variables sin nombre con el comando **Tabla > Nueva**, sólo podrá establecer un enlace con la última CPU configurada si está definida.

---

## 18.7 Observar variables

### 18.7.1 Introducción a la observación de variables

Para observar variables puede elegir una de las alternativas siguientes:

- Active la función "Observar" con el comando **Variable > Observar**. Dependiendo del punto y de la condición de disparo que se hayan ajustado, los valores de las variables seleccionadas se visualizan en la tabla de variables. Si ha ajustado la condición de disparo "Cíclico", desactive nuevamente la función "Observar", con el comando **Variable > Observar**.
- Con el comando **Variable > Actualizar valores de estado** se actualizan inmediatamente los valores de las variables seleccionadas una única vez. Los valores actuales de las variables seleccionadas se visualizan en la tabla de variables.

### Cancelar la observación con la tecla ESC

Si pulsa la tecla ESC con la función "Observar" en curso, saldrá de "Observar" sin consulta previa.

### 18.7.2 Ajustar el disparo para observar variables

Para observar variables puede visualizar en la PG los valores actuales de variables de un programa de usuario en un punto predeterminado de la ejecución del programa (el punto de disparo).

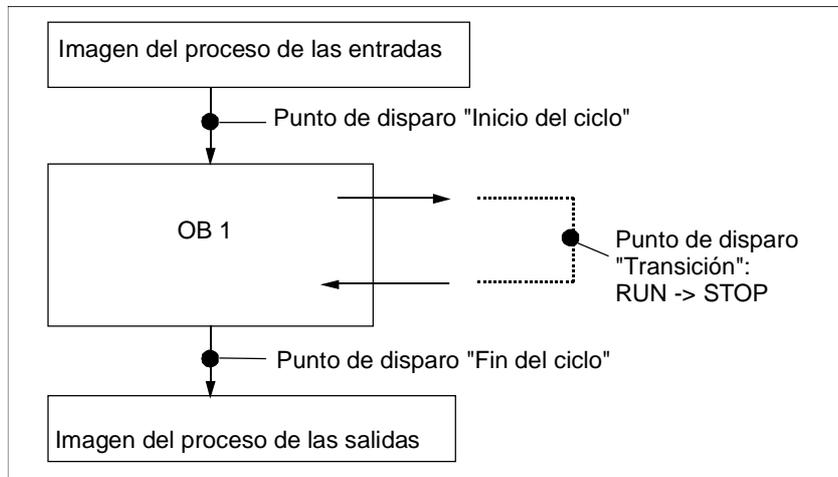
Mediante la elección del punto de disparo se determina el momento en el que se visualizan los valores de estado de las variables.

Con el comando **Variable > Condiciones de disparo** es posible ajustar los puntos y las condiciones de disparo

Condiciones de disparo	Posibilidades de ajuste
Punto de disparo	Inicio del ciclo Fin del ciclo Transición de RUN a STOP.
Condición de disparo	Unico Permanente

## Punto de disparo

La figura siguiente ilustra la posición de los puntos de disparo.



De la posición de los puntos de disparo se deduce lo siguiente:

- para forzar las entradas conviene elegir el punto de disparo "Inicio del ciclo" (corresponde al principio del programa de usuario OB1), ya que de lo contrario, la imagen del proceso de las entradas se volvería a actualizar después del forzado y, por lo tanto, se sobrescribiría.
- para forzar las salidas conviene elegir el punto de disparo "Fin del ciclo" (corresponde al final del programa de usuario OB 1), ya que de lo contrario, la imagen del proceso de las salidas podría ser sobrescrito por el programa de usuario.

Para poder visualizar el valor forzado, el punto de disparo para "Observar" se deberá ajustar en "Inicio del ciclo", en tanto que el punto de disparo para "Forzar" se deberá definir en "Fin del ciclo".

## Disparo inmediato

Los valores de variables seleccionadas se pueden actualizar con el comando de menú **Variable > Actualizar valores de estado**. Esta petición se ejecuta una sola vez tan pronto como sea posible, sin tener relación con una posición determinada en el programa de usuario. Estas funciones se utilizan principalmente en modo STOP para observar y forzar variables.

## Condición de disparo

La siguiente tabla muestra el efecto que tiene la condición de disparo ajustada en la observación de variables:

	Condición de disparo única	Condición de disparo cíclica
Observar variables	Actualización única dependiendo del punto de disparo	Observar con condición de disparo definida Al comprobar un bloque se puede observar exactamente el procesamiento ulterior.

## 18.8 Forzar variables

### 18.8.1 Introducción al forzado de variables

Para forzar variables puede elegir una de las alternativas siguientes:

- Active la función "Forzar" con el comando **Variable > Forzar**. Dependiendo del punto y de la condición de disparo que se hayan ajustado, el programa de usuario adopta los valores de forzado de las variables seleccionadas en la tabla de variables. Si ha ajustado la condición de disparo "Cíclico", desactive nuevamente la función "Forzar" con el comando **Variable > Forzar**.
- Con el comando **Variable > Activar valores de forzado** se actualizan inmediatamente los valores de las variables seleccionadas una única vez.

Las funciones Forzado permanente y Desbloquear salidas ofrecen más posibilidades al respecto.

#### Tenga en cuenta lo siguiente al forzar:

- Sólo se forzarán los operandos visibles en la tabla de variables al comienzo del forzado. Si después de iniciar el forzado se reduce el área visible de la tabla de variables, podrían forzarse operandos que han dejado de ser visibles. Si se amplía el área visible de la tabla de variables, puede haber operandos que ya no se puedan forzar.
- El proceso de forzado es irreversible (p. ej., con **Edición > Deshacer**).
- Durante el forzado cíclico será imposible desplazar la pantalla.



#### **Peligro**

Si los valores de las variables se modifican estando la instalación en servicio y se presentan fallos de funcionamiento o errores del programa, pueden producirse serios daños materiales o personales.

Antes de ejecutar la función "Forzar", asegúrese de que no puedan presentarse situaciones peligrosas.

---

## Cancelar el forzado con la tecla ESC

Si, con la función "Forzar" en curso, pulsa la tecla ESC, saldrá de "Forzar" sin consulta previa.

## 18.8.2 Ajustar las condiciones de disparo para forzar variables

En un punto predeterminado en la ejecución del programa de usuario (punto de disparo) se pueden asignar valores únicos o cíclicos a variables individuales de dicho programa.

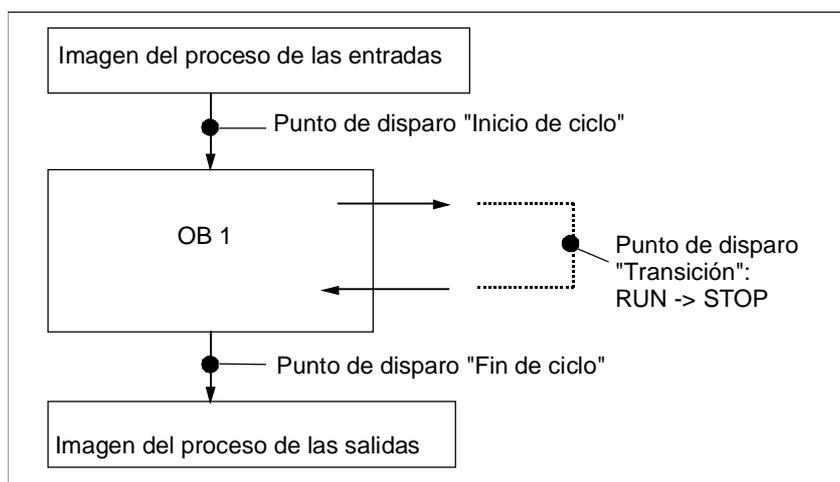
Mediante la elección del punto de disparo se determina el momento en el que los valores de forzado se asignan a las variables.

Con el comando **Variable > Condiciones de disparo** es posible ajustar los puntos y las condiciones de disparo

Condiciones de disparo	Posibilidades de ajuste
Punto de disparo	Inicio del ciclo Fin del ciclo Transición a STOP.
Condición de disparo	Unico Cíclico

## Punto de disparo

La figura siguiente muestra la posición de los puntos de disparo.



Si se ajusta un mismo punto de disparo para observar y forzar variables, el valor de estado se visualiza **antes** de forzar, puesto que la función "Observar" se ejecuta antes de "Forzar". Para poder visualizar el valor forzado, el punto de disparo para "Observar" se deberá ajustar en "Inicio del ciclo", en tanto que el punto de disparo para "Forzar" se deberá definir en "Fin del ciclo".

Al forzar variables se deberán tener en cuenta las siguientes observaciones respecto a los puntos de disparo:

- Si se ha ajustado la condición de disparo "Único", aparecerá un mensaje de error en caso de que no sea posible forzar las variables seleccionadas.
- Si se ha ajustado la condición de disparo "Cíclico", no aparecerá ningún mensaje.

### Disparo inmediato

Con el comando de menú **Variable > Activar valores de estado** puede forzar los valores de las variables. Esta petición se ejecuta una sola vez tan pronto como sea posible, sin tener relación con una posición determinada en el programa de usuario. Esta función se utiliza principalmente en modo STOP para forzar variables.

### Condición de disparo

La siguiente tabla muestra el efecto que tiene la condición de disparo ajustada en el forzado de variables:

	<b>Condición de disparo única</b>	<b>Condición de disparo cíclica</b>
Forzar variables	<i>Activación única (forzar variables)</i> Dependiendo del punto de disparo, a las variables se les pueden asignar valores una sola vez.	<i>Forzar con condición de disparo definida</i> Asignando valores fijos puede simular determinadas situaciones para su programa de usuario, comprobando así las funciones programadas.

## 18.9 Forzado permanente de variables

### 18.9.1 Introducción al forzado permanente de variables

Es posible asignar valores permanentes (fijos) a las variables de un programa de usuario, de manera que el programa que se ejecute en la CPU no los pueda cambiar ni sobrescribir. Para ello es necesario que la CPU ofrezca esta prestación (p. ej., las CPUs S7-400). Puede ajustar diversas situaciones para predefinir valores fijos en variables para su programa de usuario, y hacer así un test de las funciones programadas.

#### Ventana "Valores de forzado":

Sólo pueden seleccionarse las opciones de menú para forzado permanente si está activa la ventana "Valores de forzado permanente".

Para visualizar dicha ventana, elija el comando de menú **Variable > Mostrar valores de forzado permanente**.

Para una CPU se deberá abrir una sola ventana "Valores de forzado permanente". En dicha ventana se visualizan todas las variables con los valores de la petición activa.

#### Ejemplo de una ventana de forzado permanente

	Operand	Símbol	Formato de vis	Valor de forz
1	<b>EB 0</b>		HEX	<b>B#16#10</b>
2	A 0.1		BOOL	true
3	A 1.2		BOOL	true
4				

En la **barra de título** figura el nombre del enlace online actual.

En la **barra de estado** figura el momento (fecha y hora) en el que se ha leído en la CPU la petición de forzado permanente.

Cuando no hay ninguna petición de forzado activa, la ventana está vacía.

Las distintas formas de **mostrar variables** en la ventana Valores de forzado permanente tienen los siguientes significados:

Visualización	Significado
Negrita	Variables predefinidas con carácter fijo con un valor en la CPU.
Normal:	Variables recién editadas.
Atenuado	Variables de un módulo no existente / no enchufado. o Variables con error de direccionamiento; se muestra un mensaje de error.

## Aceptar los operandos de la tabla de variables que se puedan forzar permanentemente

Al visualizarse la ventana "Valores de forzado permanente" aparecerá un mensaje que le preguntará si quiere adoptar o no los operandos que se puedan forzar permanentemente desde la tabla de variables a la ventana "Valores de forzado permanente". Los operandos susceptibles de ser forzados permanentemente son operandos a los que se les puede asignar valores fijos, es decir, que no varíen. Como máximo se pueden transferir 512 operandos de este tipo.

## Aceptar la petición de forzado permanente de la CPU o crear una nueva petición de forzado permanente

Cuando la ventana "Valores de forzado permanente" está abierta y activa se muestra un segundo mensaje:

- Si confirma dicho mensaje, las modificaciones efectuadas en la ventana "Valores de forzado permanente" son sobrescritas con la petición de forzado permanente existente en la CPU. Podrá restablecer el contenido anterior de la ventana usando el comando **Edición > Deshacer**.
- Si cancela el mensaje, se mantiene el contenido actual de la ventana "Valores de forzado permanente". Después puede guardar el contenido en la ventana "Valores de forzado permanente" como tabla de variables (usando el comando de menú **Tabla > Guardar como**), o seleccionar el comando de menú **Variable > Forzado permanente**: al hacerlo, el contenido actual de la ventana "Valores de forzado permanente" se escribe como nueva petición de forzado permanente en la CPU.

En la ventana "Valores de forzado permanente" no se pueden observar ni forzar variables. Tan sólo es posible en la tabla de variables.

## Guardar el contenido de la ventana "Valores de forzado permanente"

El contenido de la ventana "Valores de forzado permanente" se puede guardar en una tabla de variables. Con el comando de menú **Insertar > Tabla de variables** se puede insertar el contenido guardado en la ventana Valores de forzado permanente.

## Notas respecto a los símbolos en la ventana "Valores de forzado permanente"

Los símbolos de la última ventana activa se importan siempre, a no ser que se haya iniciado la función "Observar y forzar variable", desde una aplicación que no tenga símbolos.

Si no se puede introducir ningún nombre simbólico, la columna "Símbolo" está oculta. Entonces tampoco puede seleccionarse el comando de menú **Herramientas > Tabla de símbolos**.

## 18.9.2 Reglas de seguridad para el forzado permanente de variables



### Prevenir daños personales y materiales

Al ejecutar la función "Forzado permanente" tenga en cuenta que una acción errónea puede:

- poner en peligro la vida o la salud de personas, o
- provocar daños en la máquina o la instalación completa.



---

#### Cuidado



- Antes de arrancar la función "Forzado permanente", debe cerciorarse de que nadie ejecute simultáneamente la misma función en la misma CPU.
- Sólo puede borrar o salir de una petición de forzado permanente con el comando de menú **Variable > Anular forzado permanente**. Con sólo cerrar la ventana "Valores de forzado permanente" o salir de la aplicación "Observar y forzar variable" no se anula la petición de forzado permanente.
- El "Forzado permanente" no puede deshacerse con el comando de menú **Edición > Deshacer**.
- Infórmese sobre las diferencias entre forzado permanente y forzado normal de variables.
- Si una CPU no soporta la función de forzado permanente, no pueden seleccionarse ninguno de los comandos de menú relativos al forzado permanente en el menú Variable.

Cuando se ha anulado el bloqueo de salidas usando el comando de menú **Variable > Desbloquear salidas**, todos los módulos de salidas forzados devuelven su valor de forzado permanente.

---

### 18.9.3 Diferencias entre el forzado normal y el forzado permanente

En la panorámica siguiente se resumen las diferencias entre el forzado normal y el forzado permanente:

Característica / función	Forzado permanente con S7-400	Forzado permanente con S7-300	Forzado normal
Marcas (M)	•	–	•
Temporizadores y contadores (T, Z)	–	–	•
Bloques de datos (DB)	–	–	•
Entradas periféricas (PEB, PEW, PED)	•	–	–
Salidas de la periferia (PAB, PAW, PAD)	•	–	•
Entradas y salidas (E, A)	•	•	•
Ajustar las condiciones de disparo	siempre disparo inmediato	siempre disparo inmediato	único o cíclico
La función tiene efecto sólo en la variable que está en el área visible de la ventana activa.	tiene efecto en todos los valores de forzado perman.	tiene efecto en todos los valores de forzado perman.	•
El programa de usuario puede sobrescribir los valores de forzado normal o permanente.	–	•	•
El valor de forzado permanente se mantiene activo sin interrupciones.	•	•	–
Al salir de la aplicación, las variables conservan sus valores.	•	•	–
Al deshacer el enlace con la CPU, las variables conservan sus valores.	•	•	–
Se permiten errores de direccionamiento: p ej.,           EW1    Valor de forzado normal/permanente: 1 EW1    Valor de forzado normal/permanente: 0	–	–	el último tiene efecto

#### Nota

- Al "Desbloquear salidas", los valores de forzado permanente de las salidas periféricas forzadas tendrán efecto en los módulos de salidas. Por el contrario, los valores de forzado de las salidas periféricas forzadas permanentemente no tendrán efecto en ellos.
- En el forzado permanente, la variable conserva siempre el valor forzado. Dicho valor se lee en cada acceso de lectura en el programa de usuario. Ningún acceso de escritura será válido.
- Los accesos de escritura del programa tienen efecto durante el forzado permanente del programa, permaneciendo efectivos hasta el siguiente punto de disparo.



## 19 Test con el estado del programa

### 19.1 Test con el estado del programa

El programa se puede comprobar visualizando para cada instrucción el estado del programa (RLO, bit de estado) o el contenido de la ficha en cuestión. El volumen de la información visualizada se define en la ficha "AWL" del cuadro de diálogo "Preferencias". Dicho cuadro de diálogo se abre con el comando de menú **Herramientas > Preferencias** en la ventana "KOP/AWL/FUP: Programar bloques".



#### **Precaución**

Si el test se realiza con la instalación en marcha y se presentan fallos en el funcionamiento del sistema o errores del programa, pueden producirse serios daños materiales o personales.

Antes de ejecutar esta función asegúrese de que no pueden presentarse situaciones peligrosas.

---

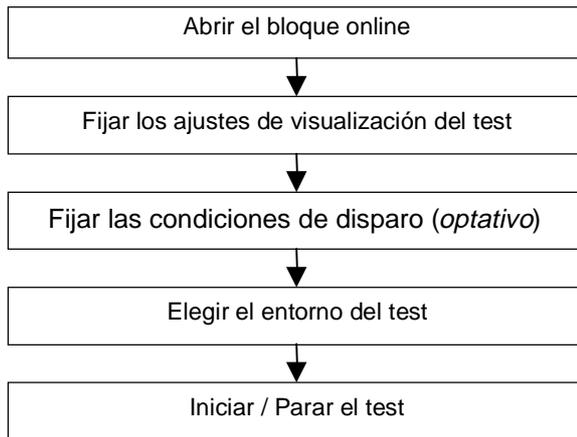
#### **Requisitos**

Para poder visualizar el estado del programa se deberán cumplir los siguientes requisitos:

- El bloque se deberá haber guardado sin errores, cargándose posteriormente en la CPU.
- La CPU deberá estar en servicio y el programa de usuario se deberá estar ejecutando.
- El bloque está abierto online.

#### **Procedimiento básico para observar el estado del programa:**

No es aconsejable llamar y comprobar inmediatamente todo el programa, sino llamar y comprobar los bloques uno por uno. Comience con los bloques lógicos subordinados, es decir, compruebe los bloques que se encuentran en el último nivel de la jerarquía de llamada, p.ej. llamándolos en el OB1 y preparando el entorno del test para el bloque con la función Observar y forzar variables.



Para posicionar puntos de parada y ejecutar el programa por etapas tiene que estar ajustado el modo de funcionamiento "Test" (véase el comando de menú **Test > Modo de funcionamiento**). Dichas funciones de test no se pueden llevar a cabo en el modo "Proceso".

## 19.2 Visualización del estado de programas

La visualización del **estado de programa** se actualiza cíclicamente.

### Identificadores preajustados en KOP Y FUP

- El estado se cumple: líneas verdes continuas
- El estado no se cumple: líneas azules punteadas
- El estado es desconocido: líneas negras continuas

Este ajuste (tipo de línea y color) se puede modificar seleccionando el comando de menú **Herramientas > Preferencias** / Ficha KOP/FUP.

### Estado de los elementos

- El estado de un contacto
  - se cumple si el valor del operando es "1",
  - no se cumple si el valor del operando es "0",
  - es desconocido si el valor del operando es desconocido.
- El estado de elementos con una salida de habilitación (ENO) corresponde al estado de un contacto con el valor de la salida ENO como operando.
- El estado de elementos con salida Q corresponde al estado de un contacto con el valor del operando.
- El estado en CALLs se cumple si tras la llamada se ha puesto a 1 el bit RB.
- El estado de una operación de salto se cumple si se realiza el salto; es decir, si la condición del salto se cumple.
- Los elementos con salida de habilitación (ENO) se representan en negro si la salida de habilitación no está asignada a una conexión.

### Estado de las líneas

- Las líneas son negras si no ha circulado corriente por ellas o si el estado es desconocido.
- El estado de aquellas líneas que comiencen en la barra de alimentación se cumple siempre ("1").
- El estado de aquellas líneas que se encuentren al comienzo de ramas paralelas se cumple siempre ("1").
- El estado de la línea situada tras un elemento se cumple si se cumplen tanto el estado de la línea situada antes del elemento como el estado del elemento.
- El estado de la línea situada tras NOT se cumple si no se cumple el estado de la línea situada antes de NOT (y viceversa).
- El estado de la línea situada **tras** la confluencia de varias líneas se cumple si
  - se cumplen tanto el estado de como mínimo una línea situada antes de la confluencia de líneas
  - como el estado de la línea situada antes de la rama.

### Estado de los parámetros

- Los valores de los parámetros **en negrita** son actuales.
- Los valores de los parámetros en letra normal provienen de un ciclo anterior; este punto del programa no se ha ejecutado en el ciclo actual.

### 19.3 Información importante para comprobar en modo Etapa individual / Puntos de parada

Para realizar un test en modo Etapa individual puede

- Editar programas instrucción por instrucción (= etapa individual)
- Posicionar puntos de parada

La función "Test en modo Etapa individual" no se puede llevar a cabo en todos los sistemas de automatización (v. la documentación del sistema de automatización correspondiente)

Palabra de estado				
/ER	STA	OS	A0	RB
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
RLO	OR	OV	A1	
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
ACU1	<input type="text" value="3039"/>	ACU2	<input type="text" value="58"/>	
AR1	<input type="text" value="0"/>	AR2	<input type="text" value="84000000"/>	
DBglob	<input type="text"/>	DBinst	<input type="text"/>	

#### Requisitos

- Deberá estar ajustado el modo de funcionamiento Test. En el modo de funcionamiento Proceso no se puede realizar el test en modo Etapa individual (v. comando de menú **Test > Modo de funcionamiento**).
- El test en modo Etapa individual sólo se puede realizar en AWL. Para realizar el test de bloques en KOP o FUP tendrá que cambiar de vista (modo de visualización) mediante el comando de menú **Ver > AWL**.
- El bloque no puede estar protegido.
- El bloque debe estar abierto online.
- El bloque abierto no puede ser modificado con el Editor

### Número de puntos de parada:

El número de puntos de parada es variable y depende del:

- número de puntos de parada ya posicionados
- número de estados de las variables en curso
- número de estados de los programas en curso

Lea, por favor, en la documentación de su sistema de automatización si éste admite el test en modo Etapa individual.

Los comandos de menú para posicionar, activar o borrar los puntos de parada se encuentran en el menú "Test". Además, dichos comandos de menú se pueden elegir mediante los botones de la barra de puntos de parada. Para visualizar ésta última, elija el comando de menú **Ver > Barra de puntos de parada**.

### Funciones de test permitidas:

- Observar/forzar variables
- Información del módulo
- Estado operativo



#### **Peligro**

Atención: evite situaciones peligrosas en la instalación durante el estado operativo "PARADA".

---

## 19.4 Información importante sobre el estado operativo PARADA

Si el programa llega a un punto de parada, el sistema de automatización cambia al estado operativo "PARADA".

### Situación de los diodos luminosos (LEDs) en el estado operativo Parada:

- LED RUN intermitente
- LED STOP encendido

### Ejecución de programas en el estado operativo "PARADA":

- En el estado operativo "PARADA" no se puede ejecutar lógica S7, es decir, todos los niveles de ejecución quedan sin reprocesar.
- Se congelan todos los temporizadores:
  - no se ejecutan las celdas de los temporizadores
  - se detienen todos los tiempos de vigilancia
  - se detienen los períodos básicos de los niveles controlados por tiempo
- Se pone en marcha el reloj de tiempo real
- Por razones de seguridad, en el estado operativo "PARADA" se desconectan siempre las salidas (comp. Módulos de salidas "output disable")

### Comportamiento durante un corte de alimentación en el estado operativo "PARADA":

- Si se produce un corte de alimentación, al retornar la alimentación de la red eléctrica, los sistemas de automatización respaldados que se encuentran en el estado operativo "PARADA", entran en el estado operativo "STOP" y se mantienen así. La CPU no vuelve a arrancar automáticamente. Partiendo de STOP podrá proseguir con los demás procedimientos (p.ej., activar/desactivar puntos de parada, arrancar manualmente).
- Los sistemas de automatización que no están respaldados no tienen "memoria" y, por eso, cuando retorna la alimentación arrancan automáticamente, independientemente del estado operativo previo.

## 19.5 Observar el estado de bloques de datos

A partir de la versión 5 de STEP 7 es posible observar un bloque de datos en la vista "Datos" online. La visualización se puede activar tanto desde un bloque de datos online como desde uno offline. En ambos casos se representa el contenido del bloque de datos online del sistema de destino.

El bloque de datos no puede ser modificado antes de iniciar el estado del programa (status). Si hay diferencias estructurales (declaración) entre el bloque de datos online y el bloque de datos offline, es posible cargar el bloque de datos offline en el sistema de destino. El programa visualiza en este caso un mensaje.

El bloque de datos se tiene que encontrar en la vista "Datos" para que se puedan representar los valores online en la columna "Valor actual". Solamente se actualiza aquella parte del bloque de datos que esté visible en la pantalla. Mientras está activo el estado (status) no se puede cambiar a la vista "Declaración".

Durante la actualización se ve una barra verde en la barra de estado del bloque de datos así como el estado operativo del sistema.

Los valores se indican en el formato del tipo de datos que corresponda, no siendo posible cambiar el formato.

Al salir del estado del programa, se vuelve a visualizar en la columna de valores actuales el valor que era válido antes de iniciar el estado del programa. No es posible adoptar los valores online actualizados en el bloque de datos offline.

### Actualización de tipos de datos:

Todos los tipos de datos simples se actualizan tanto en un DB global como en todas las declaraciones (in/out/inout/stat ) de un bloque de datos de instancia.

Algunos tipos de datos no se pueden actualizar. Estando activado el estado del programa, estos campos se visualizan sobre fondo gris en la columna "Valor actual" e indican valores no actualizados.

- Los tipos de datos compuestos DATE\_AND\_TIME y STRING no se actualizan.
- En los tipos de datos compuestos ARRAY, STRUCT, UDT, FB, SFB sólo se actualizan aquellos elementos que son tipos de datos simples.
- En la declaración INOUT de un bloque de datos de instancia solamente se representa el puntero que señala al tipo de datos compuesto, pero no sus elementos. El puntero no se actualiza.
- Los tipos de parámetros no se actualizan.

## 19.6 Definir el entorno de llamada del bloque

Para analizar el estado del programa puede especificar condiciones de llamada definiendo el entorno de llamada. Así, el estado del programa sólo se analizará cuando se cumpla la condición de disparo indicada.

Para compilar y cargar una tabla GD:

1. Elija el comando de menú **Test > Entorno de llamada**.
2. En el cuadro de diálogo visualizado, defina las condiciones de disparo y confírmelas haciendo clic en "Aceptar".

Selección posible	Significado
Ruta de llamada	Aquí se puede indicar la ruta de llamada a través de la cual se debe llamar el bloque a comprobar, con objeto de iniciar la visualización de estado. Se pueden indicar los tres últimos niveles de llamada antes de alcanzar el bloque a comprobar.
Con dirección	Active esta opción si desea anular la condición de la ruta de llamada.
Bloques de datos de abiertos	El entorno de llamada se define aquí indicando uno o dos bloques de datos. El análisis de estado se efectúa cuando el bloque a comprobar se haya llamado con los bloques de datos indicados.

### Definir el entorno de llamada para instancias de bloques

Para mostrar el estado de programa de un bloque en una instancia determinada proceda de la siguiente forma:

1. Seleccione el comando de menú **Test > Funcionamiento** y ajuste el modo de funcionamiento "Modo Test".
2. Abra el bloque de llamada y coloque el cursor sobre el comando de llamada deseado (línea CALL en AWL o caja del bloque en KOP/FUP).
3. Seleccione por medio del botón derecho del ratón el comando de menú **Bloque llamado > Observar con ruta de llamada**.

**Resultado:** Se abre el bloque llamado, se introduce la llamada como criterio en las condiciones de disparo del bloque y se activa el estado para esta instancia del bloque.

Las condiciones de disparo existentes para bloques de datos permanecen intactas.



## **20 Test con el programa de simulación S7-PLCSIM (software opcional)**

### **20.1 Test con el programa de simulación (software opcional)**

Con el paquete opcional Simulación puede ejecutar y comprobar su programa en un sistema de automatización simulado existente en su ordenador o unidad de programación (p. ej., PG 740). Dado que la simulación tiene lugar por completo en el software STEP 7, no necesitará hardware S7 (CPU o módulos de señales). Con la CPU S7 simulada puede comprobar programas para CPUs S7-300 y S7-400 y eliminar errores.

La aplicación dispone de un sencillo interface de usuario para supervisar y modificar los distintos parámetros que se usan en su programa (p.ej. para activar y desactivar las entradas). Además podrá utilizar otras aplicaciones del software de STEP 7 mientras su programa se ejecuta desde la CPU simulada. Así podrá, por ejemplo, observar y modificar variables con la tabla de variables.

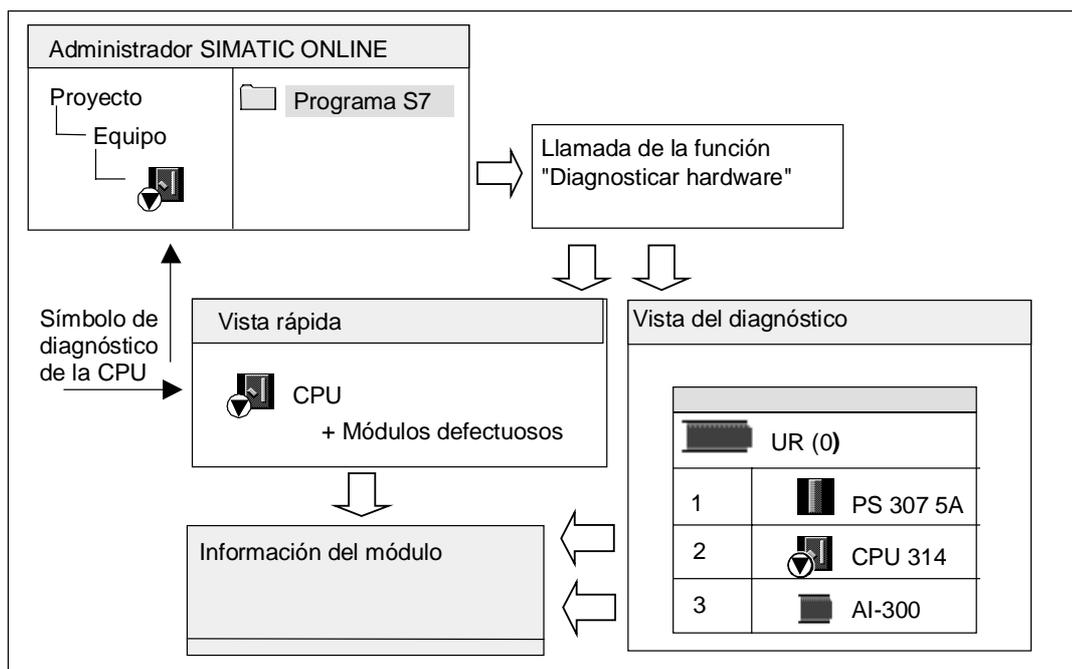


# 21 Diagnóstico

## 21.1 Diagnóstico del hardware y búsqueda de errores

Los símbolos de diagnóstico indican si existe información de diagnóstico para un módulo determinado. Estos símbolos muestran el estado del módulo correspondiente y, tratándose de CPUs, también indican su estado operativo.

Los símbolos de diagnóstico se visualizan en la ventana del proyecto en la vista "Online" así como tras llamar la función "Diagnosticar hardware" en la vista rápida (ajuste por defecto) o en la vista del diagnóstico. Si requiere información de diagnóstico más detallada vaya a la "Información del módulo" a la que se accede haciendo doble clic en un símbolo de diagnóstico de la vista rápida o de la vista del diagnóstico.



## Procedimiento para localizar fallos

1. Abra la ventana online del proyecto con el comando de menú **Ver > Online**.
2. Abra todos los equipos, de manera que se vean los módulos programables que tienen configurados.
3. Determine las CPUs en las que aparece un símbolo de diagnóstico que indique un error o fallo. Acceda con la tecla F1 a la página de la Ayuda que explica el significado de los distintos símbolos.
4. Seleccione el equipo que desea analizar.
5. Elija el comando de menú **Sistema de destino > Información del módulo** para hacerse mostrar informaciones sobre el estado de la CPU de este equipo.
6. Elija el comando de menú **Sistema de destino > Diagnosticar hardware** para hacerse mostrar la "Vista rápida" con la CPU y los módulos defectuosos de este equipo. La vista rápida aparece por defecto (comando de menú **Herramientas > Preferencias**, ficha "Vista").
7. Seleccione un módulo defectuoso en la vista rápida.
8. Haga clic en el botón de comando "Información del módulo", para obtener información sobre este módulo.
9. Estando en la vista rápida, haga clic en el botón de comando "Abrir equipo online" para hacerse mostrar la vista del diagnóstico. Esta contiene todos los módulos del equipo en el orden en que están enchufados en los slots.
10. Estando en la vista del diagnóstico, haga doble clic en un módulo para hacerse mostrar informaciones sobre el estado del mismo. También obtendrá información sobre aquellos módulos que no están defectuosos y que por consiguiente no aparecen en la vista rápida.

No es necesario realizar todos los pasos, sino que basta realizar los suficientes para acceder a la información que necesita.

## 21.2 Símbolos de diagnóstico en la vista online

Los símbolos de diagnóstico se visualizan en la vista online del proyecto y en la ventana "Configurar hardware" con la vista online de las tablas de configuración.

Los símbolos de diagnóstico facilitan la detección de fallos. Basta con echar un vistazo al símbolo del módulo para detectar si existen informaciones de diagnóstico al respecto. Si no han ocurrido fallos, los símbolos de los módulos se representarán sin símbolos adicionales de diagnóstico.

Si existe información de diagnóstico para un módulo en su símbolo, se visualizará también un símbolo de diagnóstico. En caso contrario, el icono del módulo aparecerá atenuado.

### Símbolos de diagnóstico de los módulos (ejemplo: FM / CPU)

Símbolo	Significado
	La configuración real difiere de la teórica: El módulo configurado no existe o se ha insertado un módulo de otro tipo.
	Error: fallo en el módulo. Posibles causas: detección de una alarma de diagnóstico, de un error de acceso a periferia o de un LED de error.
	El diagnóstico no es posible porque no hay ningún enlace online o porque la CPU no proporciona informaciones de diagnóstico del módulo (p. ej., fuente de alimentación, submódulos).

### Símbolos de diagnóstico de los estados operativos (ejemplo: CPU)

Símbolo	Estado operativo
	ARRANQUE
	STOP
	STOP debido al estado operativo "STOP" de otra CPU en modo multiprocesador
	RUN
	PARADA

### Símbolo de diagnóstico para el forzado permanente

Símbolo	Estado operativo
	<p>En este módulo se está efectuando un forzado permanente de las variables, es decir que a las variables del respectivo programa de usuario se les han asignado valores fijos que el programa no puede modificar.</p> <p>El símbolo de forzado permanente puede aparecer también junto con otros (aquí junto con el símbolo del estado operativo "RUN").</p>

### Actualizar los símbolos de diagnóstico

Tiene que estar activada la ventana en cuestión.

- Pulse la tecla de función F5 o
- Elija en la ventana el comando de menú **Ver > Actualizar**.

## 21.3 Diagnosticar el hardware: Vista rápida

### 21.3.1 Acceder a la vista rápida

La vista rápida ofrece un acceso rápido a la función "Diagnosticar hardware" con información menos detallada que la que se indica en la vista del diagnóstico de HWConfig. La vista rápida aparece de forma estándar al llamar la función "Diagnosticar hardware".

#### Mostrar la vista rápida

Esta función se activa desde el Administrador SIMATIC con el comando de menú **Sistema de destino > Diagnosticar hardware**.

El comando de menú se utiliza de la manera siguiente:

- en la ventana online del proyecto, cuando hay un módulo o un programa S7/M7 seleccionado.
- en la ventana "Estaciones accesibles" cuando hay una estación seleccionada ("MPI=...") y esta entrada pertenece a un módulo CPU.

Así podrá seleccionar módulos de las tablas de configuración visualizadas y hacerse mostrar información sobre su estado.

### 21.3.2 Funciones de información de la vista rápida

En la vista rápida se visualizan las siguientes informaciones:

- Datos sobre el enlace online con la CPU
- Símbolo de diagnóstico de la CPU
- Símbolo de diagnóstico de los módulos en los que la CPU ha detectado un fallo (p. ej. alarma de diagnóstico, error de acceso a la periferia).
- El tipo y la dirección del módulo (bastidor, slot, sistema maestro DP con número de equipo)

#### Otras posibilidades de diagnóstico en la vista rápida

- **Información del módulo**

Con el botón de comando "Información del módulo" se abre este cuadro de diálogo. Dependiendo de las funciones de diagnóstico que ofrezca el módulo se visualizarán informaciones detalladas de diagnóstico del módulo que esté seleccionado. Especialmente puede hacerse mostrar las entradas del búfer de diagnóstico.

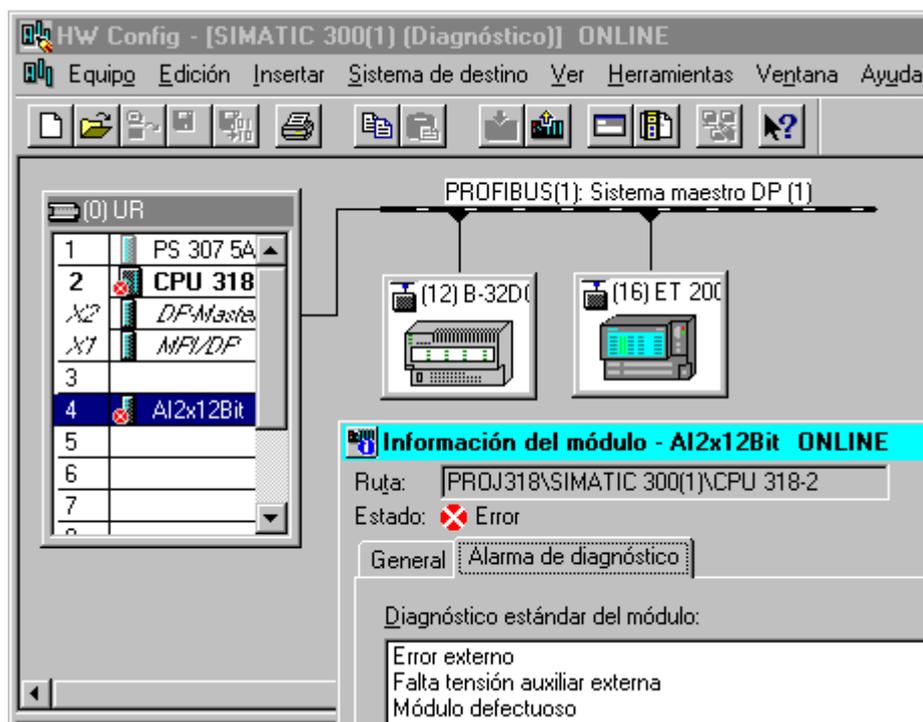
- **Vista del diagnóstico**

Con el botón de comando "Abrir equipo online" abra el cuadro de diálogo que contiene – a diferencia de la vista rápida – un gráfico con todo el equipo así como con los datos de configuración. Se posiciona en el módulo que esté seleccionado en la lista "CPU / Módulos defectuosos".

## 21.4 Diagnosticar el hardware: Vista del diagnóstico

### 21.4.1 Acceder a la vista de diagnóstico de HW Config

Utilizando esta alternativa se puede visualizar el diálogo "Información del módulo" para todos los módulos del bastidor. La vista de diagnóstico (tabla de configuración) muestra la configuración real de un equipo a nivel de bastidor, así como las estaciones DP con sus módulos.



#### Nota

- Si la tabla de configuración ya está abierta offline, puede acceder a la vista online de la tabla de configuración con el comando de menú **Equipo > Abrir online**.
- Las fichas visualizadas en el cuadro de diálogo "Información del módulo" pueden variar dependiendo de si el módulo es apto para el diagnóstico o no.
- En la ventana "Estaciones accesibles" se visualizan solamente los módulos con dirección de estación propia (dirección MPI o PROFIBUS).

### Llamada en el Administrador SIMATIC desde la vista ONLINE de un proyecto

1. En la vista del proyecto en el Administrador SIMATIC, establezca un enlace online con el sistema de destino, eligiendo el comando de menú **Ver > Online**.
2. Seleccione un equipo y ábralo haciendo doble clic en el mismo.
3. Abra seguidamente el objeto "Hardware" que contiene. Se abrirá la vista de diagnóstico.

Allí podrá seleccionar un módulo y acceder a la información del mismo, eligiendo el comando de menú **Sistema de destino > Información del módulo**.

### Llamada en el Administrador SIMATIC desde la vista OFFLINE de un proyecto

Proceda como sigue:

1. En la vista del proyecto en el Administrador SIMATIC, seleccione un equipo y ábralo haciendo doble clic en el mismo.
2. Abra seguidamente el objeto "Hardware" que contiene. Se abrirá la tabla de configuración.
3. Elija el comando **Equipo > Abrir online**.
4. Se abrirá la vista de diagnóstico de HW Config con la configuración del equipo tal y como ha sido determinada en los módulos (p.ej. CPU). El estado de los módulos se representa mediante símbolos. El significado de dichos símbolos se explica en la Ayuda en pantalla. Si hay módulos defectuosos o si faltan módulos configurados, se indican en un cuadro de diálogo aparte. Desde dicho cuadro de diálogo se puede navegar directamente a uno de los módulos visualizados (botón "Ir a").
5. Haga doble clic en el símbolo del módulo cuyo estado le interesa saber. En un cuadro de diálogo compuesto por varias fichas (dependiendo del tipo de módulo) es posible analizar detalladamente su estado.

### Llamada en el Administrador SIMATIC desde la ventana "Estaciones accesibles"

Proceda como sigue:

1. Vaya al Administrador SIMATIC y abra la ventana "Estaciones accesibles" con el comando **Sistema de destino > Mostrar estaciones accesibles**.
2. En la ventana "Estaciones accesibles", seleccione una estación.
3. Seleccione a continuación el comando de menú **Sistema de destino > Diagnosticar hardware**.

---

#### Nota

En la ventana "Estaciones accesibles" se visualizan solamente los módulos con dirección de estación propia (dirección MPI o PROFIBUS).

---

### 21.4.2 Funciones de información de la vista de diagnóstico

En la vista online se visualiza toda la configuración del equipo accesible online. Esta abarca:

- Configuración de los bastidores.
- Símbolos de diagnóstico de **todos** los módulos configurados.  
De los símbolos se deduce el estado del módulo en cuestión, y si el módulo es una CPU se indica además su estado operativo.
- Tipo y referencia del módulo así como direcciones y comentarios sobre la configuración.

### Otras posibilidades de diagnóstico en la vista del diagnóstico

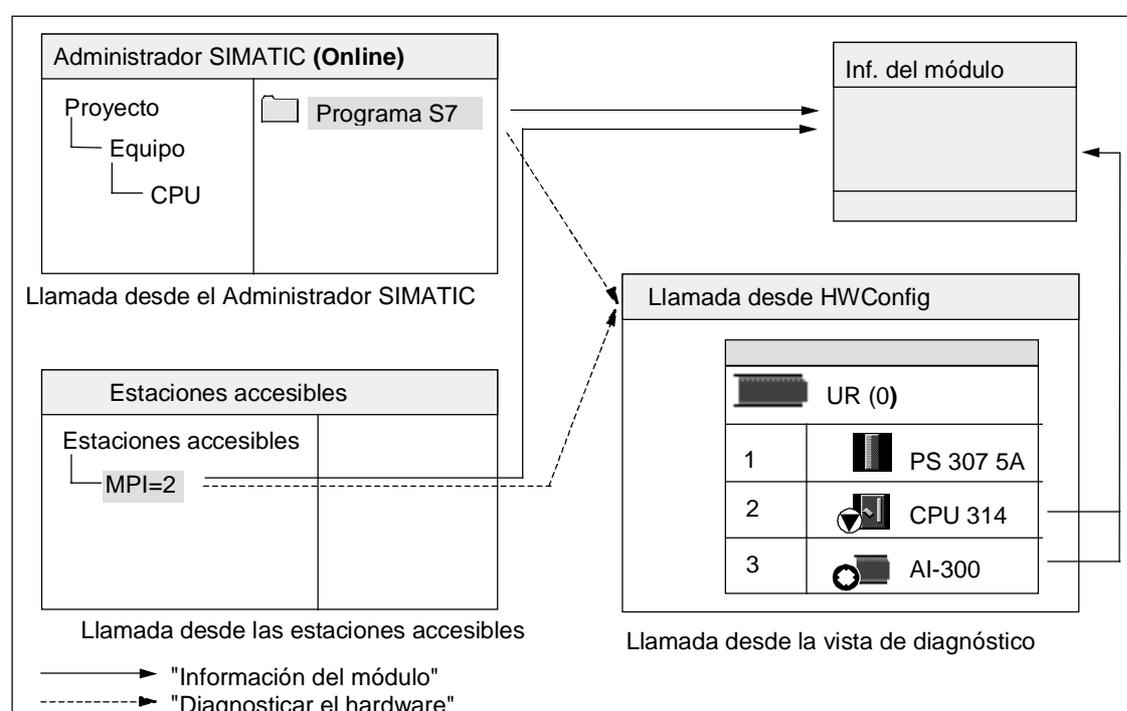
Haciendo doble clic en un módulo puede hacerse mostrar información sobre su estado.

## 21.5 Información del módulo

### 21.5.1.1 Métodos para acceder a la información del módulo

Es posible acceder al cuadro de diálogo "Información del módulo" desde diversos puntos. A continuación se proponen algunos procedimientos como ejemplo para distintos tipos de aplicación de frecuente uso:

- Llamada en el Administrador SIMATIC desde una ventana del proyecto "online" u "offline"
- Llamada en el Administrador SIMATIC desde una ventana "Estaciones accesibles"
- Llamada desde la vista de diagnóstico de HW Config



Para averiguar el estado de un **módulo con dirección de estación propia** necesita un enlace online con el sistema de destino. Lo podrá establecer con el modo de visualización online del proyecto o en la ventana "Estaciones accesibles".

### 21.5.2 Funciones de la información del módulo

Las funciones de información se encuentran a la disposición en la ficha de ese mismo nombre en el cuadro de diálogo "Información del módulo". En un caso concreto se visualizan únicamente las fichas apropiadas para el módulo seleccionado.

Función	Información	Aplicación
General	Datos de identificación del módulo seleccionado, p.ej. tipo, referencia, versión, estado, slot en el bastidor.	La información online del módulo físico se puede comparar con los datos del módulo configurado.

Función	Información	Aplicación
Búfer de diagnóstico	Panorámica de los eventos residentes en el búfer de diagnóstico e informaciones detalladas sobre el evento seleccionado..	Para evaluar la causa del STOP de una CPU y los eventos que se han presentado en el módulo seleccionado.  Gracias al búfer de diagnóstico, los errores de sistema se pueden evaluar incluso al cabo de un tiempo prolongado para averiguar la causa de un STOP o para determinar e interpretar la aparición de determinados eventos de diagnóstico.
Alarma de diagnóstico	Datos de diagnóstico del módulo seleccionado	Para averiguar la causa del fallo de un módulo.
Diagnóstico del esclavo DP	Datos de diagnóstico del esclavo DP seleccionado (según la norma EN50170).	Para averiguar la causa del fallo de un esclavo DP.
Memoria	Ampliación de la memoria, Utilización actual de la memoria de trabajo y de la memoria de carga de la CPU seleccionada o del módulo de función (FM) M7.	Antes de transferir bloques nuevos o ampliados a una CPU, con objeto de comprobar si en dicha CPU/FM se dispone de suficiente memoria de carga, así como para comprimir el contenido de la memoria.
Tiempo de ciclo	Duración del ciclo máximo, mínimo y del último ciclo de la CPU seleccionada o del módulo de función (FM) M7.	Para comprobar el tiempo de ciclo mínimo parametrizado, así como los tiempos de ciclo máximo y actual.
Sistema de reloj	Hora actual, horas de funcionamiento e informaciones respecto a la sincronización del reloj (intervalos de sincronización).	Para visualizar la hora y la fecha de un módulo y para comprobar la sincronización del reloj.
Datos característicos	Áreas de operandos y bloques disponibles del módulo (CPU o FM) seleccionado.	Antes y durante la creación de un programa de usuario y para comprobar si la CPU cumple los requisitos para poder ejecutar dicho programa, p.ej. respecto al tamaño de la imagen del proceso.
	Visualización de todos los tipos de bloques que ofrece el módulo seleccionado. Lista de los OBs, SFBs y SFCs que se pueden utilizar en el módulo en cuestión.	Para comprobar qué bloques estándar puede contener o llamar su programa de usuario para poder ejecutarse en la CPU seleccionada.
Comunicación	Las velocidades de transferencia, los enlaces, la carga de la comunicación, así como el tamaño máximo de los telegramas en el bus K del módulo seleccionado.	Para determinar cuántos y qué enlaces de la CPU o del FM M7 son posibles o están ocupados.
Pilas	Ficha <b>Pilas</b> : sólo se puede activar en estado operativo STOP o PARADA. Se muestra la pila BSTACK del módulo seleccionado. Puede leer además las pilas USTACK y LSTACK y saltar a la posición donde se ha producido el error en el bloque interrumpido.	Para averiguar la causa de un cambio a modo STOP y para corregir un bloque.

### **Informaciones adicionales**

En cada ficha se visualizan las siguientes informaciones adicionales:

- Ruta online del módulo seleccionado
- Estado operativo de la CPU correspondiente (p. ej., RUN, STOP)
- Estado del módulo seleccionado (p. ej., error, OK)
- Estado operativo del módulo seleccionado (p.ej. RUN, STOP), en tanto que dicho módulo disponga de un estado operativo propio (p.ej. CP342-5).

El estado operativo de la CPU y el estado del módulo seleccionado no se pueden visualizar si el cuadro de diálogo "Información del módulo" de un módulo que no sea una CPU se abre desde la ventana "Estaciones accesibles".

### **Visualizar varios módulos a la vez**

Si lo desea, puede hacerse mostrar información sobre distintos módulos a la vez. Para ello deberá cambiar de módulo, seleccionar un módulo diferente y llamar la correspondiente información del mismo. Entonces aparecerá otro grupo de fichas. No obstante, sólo es posible abrir un grupo de fichas por módulo.

### **Actualizar los indicadores de la información del módulo**

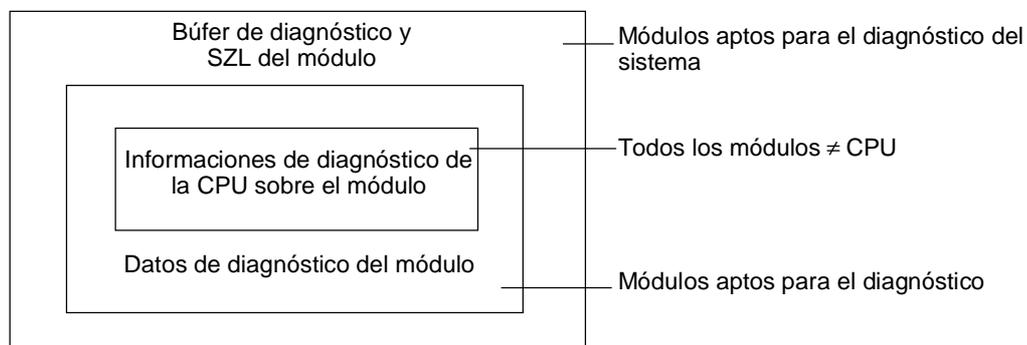
Cada vez que se cambia a otra ficha del cuadro de diálogo "Información del módulo" se vuelven a leer los datos actuales del módulo. Sin embargo, mientras se está leyendo una de las fichas no se actualizan automáticamente los valores. Haciendo clic en el botón de comando "Actualizar" se vuelven a leer los datos del módulo sin cambiar de ficha.

### 21.5.3 Volumen de información del módulo en función del tipo de módulo

El volumen de información que se puede evaluar y visualizar depende:

- del módulo seleccionado y
- del punto de vista desde el que se consulta la información del módulo.  
El volumen total de información se obtiene haciendo la llamada desde la vista online de la tabla de configuración o desde la ventana del proyecto.  
Se puede alcanzar un ámbito limitado de información con la ventana "Estaciones accesibles".

Según el volumen de información disponible, los módulos se dividen en módulos aptos para el diagnóstico del sistema, aptos para el diagnóstico y no aptos para el diagnóstico. El gráfico siguiente ilustra esta relación:



- Módulos aptos para el diagnóstico del sistema: FM351 y FM354 (por ejemplo).
- Módulos aptos para el diagnóstico: la mayoría de los módulos de señales (SM) analógicas.
- Módulos no aptos para el diagnóstico: la mayoría de los módulos de señales (SM) digitales.

## Fichas disponibles

La tabla indica qué fichas del cuadro de diálogo "Información del módulo" están disponibles para cada tipo.

Ficha	CPU o FM M7	Módulo apto para el diagnóstico del sistema	Módulo apto para el diagnóstico	Módulo no apto para el diagnóstico	Esclavo DP
General	sí	sí	sí	sí	sí
Búfer de diagnóstico	sí	sí	–	–	–
Alarma de diagnóstico	–	sí	sí	–	–
Memoria	sí	–	–	–	–
Tiempo de ciclo	sí	–	–	–	–
Sistema de reloj	sí	–	–	–	–
Datos característicos	sí	–	–	–	–
Pilas	sí	–	–	–	–
Comunicación	sí	–	–	–	–
Diagnóstico del esclavo DP	–	–	–	–	sí
Estado H <sup>1)</sup>	sí	–	–	–	–

<sup>1)</sup> sólo en CPUs de sistemas H

Además de las informaciones que figuran en las fichas, se visualiza también el estado operativo de los módulos. Si el cuadro de diálogo se llama online desde las tablas de configuración, el estado del módulo se indicará desde el punto de vista de la CPU (p. ej., OK, Error, Módulo no existente).

## 21.6 Diagnóstico en el estado operativo STOP

### 21.6.1 Procedimiento básico para averiguar la causa de un STOP

Para averiguar por qué la CPU ha pasado a STOP proceda como sigue:

1. Seleccione la CPU que ha pasado al estado operativo STOP.
2. Elija el comando **Sistema de destino > Información del módulo**.
3. Seleccione la ficha "Búfer de diagnóstico".
4. A través de las últimas entradas podrá averiguar la causa del STOP.

En caso de errores de programación:

1. La entrada "STOP por OB de error de programación no cargado" significa p.ej. que la CPU ha detectado un error de programación, habiendo intentado arrancar el OB (no existente) para remediar dicho error. La entrada precedente indica el error de programación en sí.
2. Seleccione el mensaje relativo al error de programación.
3. Haga clic en el botón "Abrir bloque".
4. Seleccione la ficha "Pilas".

### 21.6.2 Contenido de las pilas en estado operativo STOP

Sirviéndose del búfer de diagnóstico y el contenido de las pilas podrá averiguar la causa de los distintos errores de ejecución de un programa de usuario.

Si la CPU ha pasado al estado operativo "STOP" p.ej. debido a un error de programación o al comando de parada, en la ficha "Pilas" de la información del módulo se visualizará la pila BSTACK (pila de bloques). El contenido de las demás pilas se puede visualizar mediante los botones "USTACK", "LSTACK" y "Pila de paréntesis". Los contenidos de las pilas indican qué instrucción en qué bloque ha causado el STOP de la CPU.

#### Contenido de la pila BSTACK

En la pila BSTACK se indican los bloques que se llamaron antes de que la CPU cambiara al modo STOP y que todavía no se han terminado de ejecutar.

## Contenido de la pila USTACK

Si hace clic en el botón "USTACK", se visualizan los datos del punto de interrupción. La pila de interrupción (USTACK) contiene los datos o estados que eran válidos cuando se produjo la interrupción, como p.ej.

- contenido de los acumuladores y de los registros
- DBs abiertos y su tamaño
- contenido de la palabra de estado
- prioridad
- bloque interrumpido
- bloque donde continuaría la ejecución del programa después de la interrupción.

## Contenido de la pila LSTACK

Es posible seleccionar un bloque cualquiera de la pila BSTACK y, haciendo clic en el botón "LSTACK", visualizar los datos locales correspondientes.

La pila de datos locales (LSTACK) contiene los valores de los datos locales de los bloques que ha utilizado el programa de usuario hasta su interrupción.

Para poder interpretar y evaluar los datos locales visualizados es necesario conocer muy bien el sistema. La primera parte de los datos visualizados corresponde a las variables temporales del bloque.

## Contenido de la pila de paréntesis

Activando el botón "Pila de paréntesis" se visualiza el contenido de la pila de paréntesis en el punto de interrupción.

La pila de paréntesis es un área de memoria que utilizan las operaciones lógicas **U(**, **UN(**, **O(**, **ON(**, **X(** y **XN(**.

El botón de comando sólo se activa cuando aún quedan paréntesis sin cerrar al producirse la interrupción.

## **21.7 Control de los tiempos de ciclo para evitar errores de tiempo**

### **21.7.1 Control de los tiempos de ciclo para evitar errores de tiempo**

La ficha "Tiempo de ciclo" de la información del módulo indica los tiempos de ciclo del programa de usuario.

Si la duración del ciclo más largo se aproxima al tiempo de vigilancia, es posible que las fluctuaciones del tiempo de ciclo causen un error de tiempo. Para evitarlo, prolongue el tiempo de ciclo máximo del programa de usuario.

Si el ciclo dura menos que el tiempo mínimo parametrizado, la CPU o el FM prolongará automáticamente el ciclo al tiempo mínimo parametrizado. Si se trata de una CPU, durante el tiempo que se prolongue el ciclo se ejecutará el OB de tarea no prioritaria (OB 90), si es que está cargado.

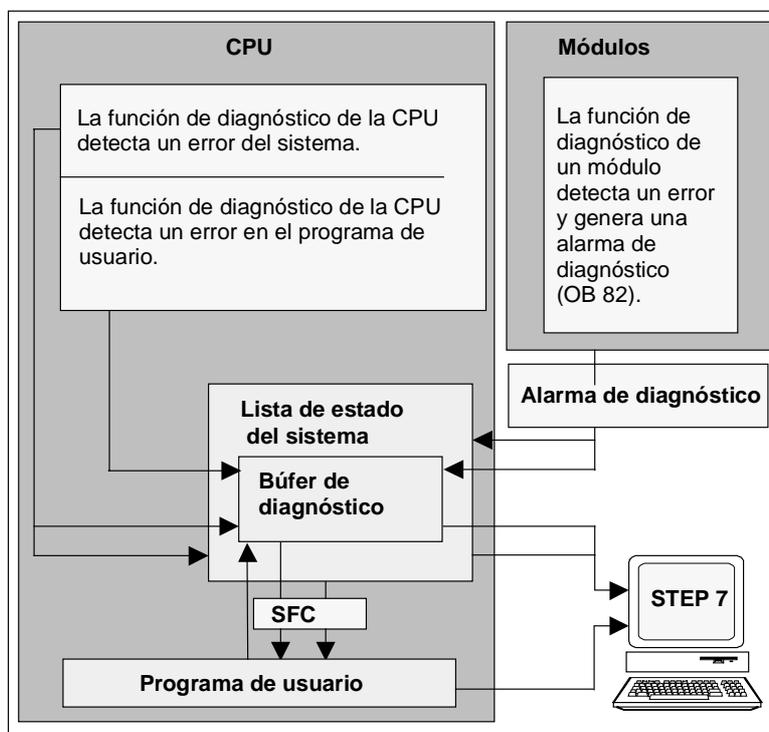
#### **Ajustar el tiempo de ciclo**

Los tiempos de ciclo máximo y mínimo se pueden ajustar al configurar el hardware. A tal efecto, en la vista offline de la tabla de configuración, haga doble clic en la CPU (o en el FM) para definir sus propiedades. Los valores deseados se pueden ajustar en la ficha "Ciclo/marca de ciclo".

## 21.8 Transmisión de informaciones de diagnóstico

### 21.8.1 Transmisión de informaciones de diagnóstico

La figura siguiente muestra la transmisión de informaciones de diagnóstico en SIMATIC S7.



#### Leer las informaciones de diagnóstico

Desde el programa de usuario es posible leer las entradas de diagnóstico utilizando la SFC 51 RDSYSST o visualizar en texto explícito los mensajes de diagnóstico con STEP 7.

Estos ofrecen informaciones sobre:

- dónde y cuándo ocurrió el error
- a qué tipo de eventos de diagnóstico pertenece la entrada (evento de diagnóstico de usuario, error síncrono/asíncrono, cambio de estado operativo).

#### Generar mensajes colectivos del sistema de control

La CPU inscribe los eventos de diagnóstico estándar y de diagnóstico ampliado en el búfer de diagnóstico. Además, en el caso de eventos de diagnóstico estándar genera un mensaje del sistema de control siempre que se cumplan las condiciones siguientes:

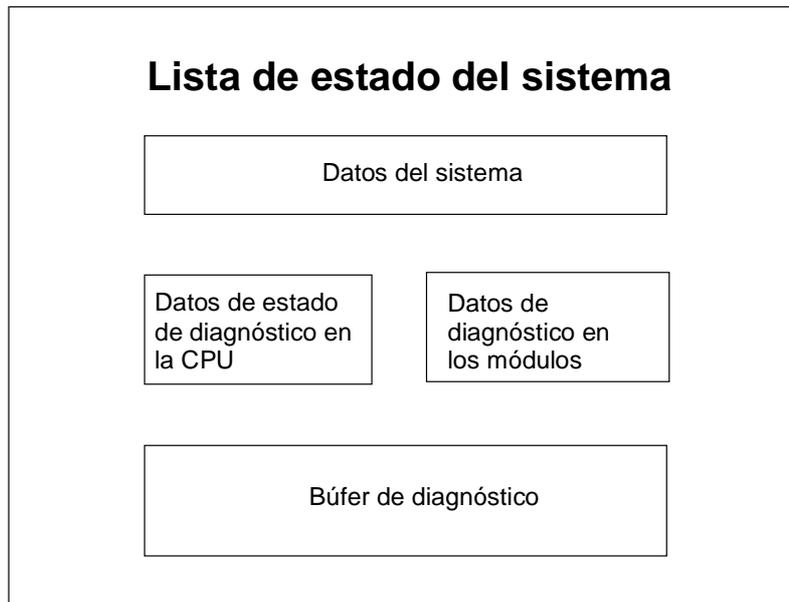
- Con STEP 7 se ha definido la generación de mensajes colectivos del sistema de control.
- Para la CPU está registrado como mínimo un visualizador para la presentación en el mismo de los mensajes colectivos del sistema de control.
- Los mensajes colectivos del sistema de control sólo se forman cuando aún no existe ningún mensaje colectivo del sistema de control de la clase asociada (hay 7 clases).
- Por cada clase es posible generar un mensaje colectivo del sistema de control.

## 21.8.2 Lista de estado del sistema (SZL)

La lista de estado del sistema SZL describe el estado actual del sistema de automatización: ofrece una visión general sobre la configuración, la parametrización actual, los estados y procesos actuales de la CPU y los módulos asociados.

Los datos de la SZL sólo se pueden leer y no se pueden modificar. Es una lista virtual que solamente se genera cuando se solicita.

Las informaciones que se pueden emitir a través de la SZL se subdividen en cuatro grupos:



## Leer la lista de estado del sistema

Se dispone de dos posibilidades para leer las informaciones de la lista de estado del sistema SZL:

- implícitamente a través de los comandos de menú de STEP 7 desde la unidad de programación (p.ej. capacidad de memoria, datos estáticos de la CPU, búfer de diagnóstico, visualizaciones de estado),
- explícitamente mediante la función de sistema SFC 51 RDSYSST desde el programa de usuario, indicando el número de referencia deseado (v. Ayuda sobre bloques ).

## Datos del sistema de la SZL

Los datos del sistema son datos identificadores de una CPU fijos o parametrizados. La tabla siguiente muestra qué informaciones se pueden emitir (listas parciales de la SZL) para los diferentes grupos de temas:

Grupo	Información
Identificación de módulos	Referencia, identificación de tipo y versión del módulo
Características de la CPU	Sistema de reloj, comportamiento del sistema (p.ej. modo multiprocesador) y descripción del lenguaje de la CPU
Areas de memoria	Capacidad de memoria del módulo (p.ej. capacidad de la memoria de trabajo)
Areas del sistema	Memoria de sistema del módulo (p.ej. cantidad de marcas, temporizadores, contadores, tipo de memoria)
Tipos de bloques	Qué tipos de bloques (OB, DB, SDB, FC, FB) están contenidos en el módulo, cantidad máxima contenida de bloques de un tipo y tamaño máximo de un tipo de bloque
Asignación de errores de alarma	Asignación de alarmas/errores a los OBs
Estado de alarmas	Generación/tratamiento de alarmas actuales
Estado de prioridades	Qué OB está en ejecución, qué prioridad ha sido inhibida por parametrización
Estado operativo y cambio de estados operativos	Qué estados operativos son posibles, último cambio, estado operativo actual

### Datos de estado de diagnóstico en la CPU

Los datos de estado de diagnóstico describen el estado actual de los componentes vigilados por el diagnóstico del sistema. La tabla siguiente muestra qué informaciones se pueden emitir (listas parciales de la SZL) para los diferentes grupos de temas:

Grupo	Información
Datos de estado de comunicación	Funciones de comunicación actualmente ajustadas en el sistema
Estación de diagnóstico	Módulos diagnosticables registrados en la CPU
Lista de información de arranque del OB	Informaciones de arranque a los OBs de la CPU
Lista de eventos de arranque	Eventos de arranque y prioridades de los OBs
Información de estado de los módulos	Informaciones de estado de todos los módulos insertados, perturbados, generadores de alarmas de proceso y asociados

### Datos de diagnóstico en los módulos

Además de la CPU, existen otros módulos diagnosticables (SM, CP, FM), cuyos datos de diagnóstico se registran en la lista de estado del sistema. La tabla siguiente muestra qué informaciones se pueden emitir (listas parciales de la SZL) para los diferentes grupos de temas:

Grupo	Información
Información de diagnóstico de módulos	Dirección inicial de módulos, errores internos/externos, error de canal, error de parametrización (4 bytes)
Datos de diagnóstico de módulos	Todos los datos de diagnóstico de un módulo determinado

#### 21.8.3 Enviar mensajes de diagnóstico personalizados

El diagnóstico del sistema estándar de SIMATIC S7 se puede ampliar adicionalmente a través de la función del sistema SFC 52 WR\_USMSG, que permite

- registrar informaciones de diagnóstico personalizadas (p. ej., informaciones sobre la ejecución del programa de usuario) en el búfer de diagnóstico
- enviar mensajes de diagnóstico personalizados a las estaciones registradas (visualizadores como PG, OP, TD).

## Eventos de diagnóstico personalizados

Los eventos de diagnóstico están subdivididos en las clases de evento 1 a F. Los eventos de diagnóstico personalizados pertenecen a las clases de evento 8 a B y se subdividen en dos grupos:

- las clases de evento 8 y 9 comprenden los mensajes con número predeterminado y texto predefinido que se puede consultar mediante su número asociado.
- las clases de evento A y B comprenden mensajes con número (A000 a A0FF, B000 a B0FF) y texto que puede elegir a voluntad.

## Enviar mensajes de diagnóstico a estaciones

Además de registrar una entrada para un evento de diagnóstico personalizado en el búfer de diagnóstico, la SFC 52 WR\_USMSG permite también enviar los eventos de diagnóstico personalizado a visualizadores registrados. Al llamar la SFC 52 con SEND= 1, el mensaje de diagnóstico se registra en el búfer de emisión y se envía automáticamente a la estación o estaciones registradas en la CPU.

Si no se puede enviar (p.ej. porque no se ha registrado ninguna estación o porque el búfer de emisión está lleno), el evento de diagnóstico personalizado se registrará sin embargo en el búfer de diagnóstico.

## Generar mensaje con indicación de acuse

Si acusa un evento de diagnóstico personalizado y desea detectar dicho acuse mediante el programa, proceda de la forma siguiente:

- ponga a 1 una variable del tipo BOOL al entrar el evento y póngala a 0 cuando el evento salga.
- vigile dicha variable con ayuda del SFB 33 ALARM.

## 21.8.4 Funciones de diagnóstico

El diagnóstico del sistema consiste en el reconocimiento, la valoración y la notificación de errores aparecidos en el sistema de automatización. A tal efecto, cada CPU y cada módulo apto para el diagnóstico (p.ej. FM354) dispone de un búfer de diagnóstico en el que se depositan informaciones detalladas sobre todos los eventos de diagnóstico en su orden de aparición.

## Eventos de diagnóstico

Los eventos de diagnóstico pueden ser, p. ej.:

- errores internos y externos de un módulo
- errores de sistema en la CPU
- cambios de estado operativo (p. ej., de RUN a STOP)
- error en el programa de usuario
- extraer/insertar módulos
- mensajes personalizados introducidos con la función de sistema SFC52

El contenido del búfer de diagnóstico se mantiene incluso después del borrado total de la CPU. Gracias al búfer de diagnóstico, los errores de sistema se pueden evaluar incluso al cabo de un tiempo prolongado para averiguar la causa de un STOP o para determinar e interpretar la aparición de determinados eventos de diagnóstico.

## Registro de los datos de diagnóstico

El registro de los datos de diagnóstico a través del diagnóstico del sistema no tiene que ser programado, está disponible de modo estándar y funciona automáticamente. SIMATIC S7 ofrece diferentes funciones de diagnóstico. Algunas de ellas están integradas en la CPU, otras son ofrecidas por los módulos (SM, CP y FM).

## Visualización de errores

Los errores internos y externos de los módulos se visualizan a través de LED en los paneles frontales del módulo correspondiente. Las visualizaciones por LED y su evaluación se describen en los manuales sobre el hardware S7. En los sistemas de automatización S7-300, los errores internos y externos se resumen en un error colectivo.

La CPU detecta errores del sistema, así como errores en el programa de usuario, y registra los eventos de diagnóstico en la lista de estado del sistema en el búfer de diagnóstico. Estos mensajes de diagnóstico se pueden leer en la PG.

Los módulos de función y de señales diagnosticables detectan errores internos y externos de módulo y generan una alarma de diagnóstico, ante la cual se puede reaccionar con un OB de alarmas.

## 21.9 Medidas en el programa para tratar fallos

### 21.9.1 Medidas en el programa para tratar fallos

Al detectar errores en la ejecución del programa (errores síncronos) y errores en el sistema de automatización (errores asíncronos), la CPU llama el OB de error correspondiente:

Error detectado	OB de error
Error de redundancia en periferia	OB 70
Error de redundancia en CPU	OB 72
Error de tiempo	OB 80
Fallo de alimentación	OB 81
Alarma de diagnóstico	OB 82
Alarma de extraer/insertar	OB 83
Error de hardware CPU	OB 84
Error de ejecución del programa	OB 85
Fallo en el bastidor o fallo de un equipo de la periferia descentralizada	OB 86
Error de comunicación	OB 87
Error de programación	OB 121
Errores de acceso a periferia	OB 122

Si el OB en cuestión no existe, la CPU pasará al estado operativo "STOP". En caso contrario, es posible programar instrucciones en el OB, indicando cómo se debe reaccionar al error. Así se pueden reducir los efectos del error o remediarlo.

### Procedimiento general

#### *Crear y abrir el OB*

1. Llame la información del módulo de su CPU.
2. Elija la ficha "Datos característicos".
3. En la lista visualizada, consulte si el OB a programar es admisible para su CPU.
4. Inserte el OB en la carpeta "Bloques" de su programa y abra este último.
5. Introduzca el programa para corregir el error.
6. Cargue el OB en el sistema de destino.

#### *Programar las medidas para el tratamiento de errores*

1. Evalúe los datos locales del OB para averiguar la causa exacta del error.  
Las variables OB8x\_FLT\_ID u OB12x\_SW\_FLT de los datos locales contienen el código de error. Su significado se explica en el manual de referencia "Funciones estándar y funciones de sistema".
2. Bifurque a la parte del programa donde se debe reaccionar a dicho error.

En el tema "Ejemplo del diagnóstico de módulos con la SFC 51 (RDSYSST)" contenido en los temas de Ayuda de las funciones estándar y de las funciones del sistema se indica un ejemplo de cómo tratar las alarmas de diagnóstico.

Para más información sobre los OBs, SFBs y SFCs consulte la Ayuda de bloques.

## 21.9.2 Evaluar el parámetro de salida RET\_VAL

Cualquier función del sistema (SFC) muestra, a través del parámetro de salida RET\_VAL (valor de respuesta), si la CPU la ha ejecutado correctamente.

### Informaciones de error en el valor de respuesta

El valor de respuesta es de tipo entero (INT). El signo de un entero indica si se trata de un número entero positivo o negativo. La relación del valor de respuesta con respecto a "0" indica si ha ocurrido un error al ejecutarse la función (v. también tabla 11-5):

- Si al ejecutarse la función ocurre un error, el valor de respuesta es inferior a 0. El bit de signo del número entero es "1".
- Si la función se ejecuta sin error, el valor de respuesta es mayor o igual a 0. El bit de signo del entero es "0".

Ejecución de la SFC por la CPU	Valor de respuesta	Signo del número entero
errónea	inferior a "0"	negativo (el bit de signo es "1")
sin error	mayor o igual a "0"	positivo (el bit de signo es "0")

### Reaccionar a informaciones de error

Si ocurre un error al ejecutar una SFC, ésta emite un código de error a través del valor de respuesta RET\_VAL.

Se distingue entre:

- un código de error general, que puede ser emitido por todas las SFCs y
- un código de error específico, que puede ser emitido por una SFC conforme a sus funciones específicas.

### Emisión del valor de la función

Algunas SFCs utilizan el parámetro de salida RET\_VAL para emitir el valor de la función. Por ejemplo, la SFC 64 TIME\_TCK emite el tiempo (hora) del sistema leído a través de RET\_VAL.

Para más información sobre el parámetro RET\_VAL consulte la Ayuda de los SFBs/SFCs.

### 21.9.3 OBs de error para reaccionar a errores detectados

#### Errores detectables

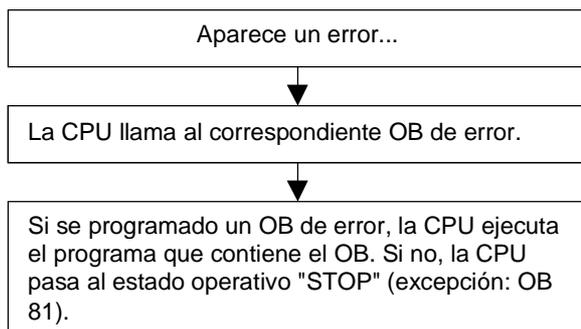
El programa del sistema puede detectar los errores siguientes:

- funcionamiento anómalo de la CPU
- error en la ejecución del programa de sistema
- error en el programa de usuario
- error en la periferia

Según el tipo de error, se conmuta la CPU al estado STOP o se llama un OB de error.

#### Programar las reacciones

Es posible crear programas para reaccionar a los distintos tipos de errores y definir así el comportamiento de la CPU. El programa para un error determinado se puede memorizar entonces en un OB (de tratamiento) de errores. Al llamar éste, se ejecuta entonces el programa.



#### OBs de error

Se distingue entre errores síncronos y asíncronos:

- Los errores síncronos se pueden asignar a una instrucción MC7 (p. ej., instrucción de carga para un módulo de señales extraído).
- Los errores asíncronos se pueden asignar a una prioridad o a todo el sistema de automatización (p. ej., desbordamiento de ciclo).

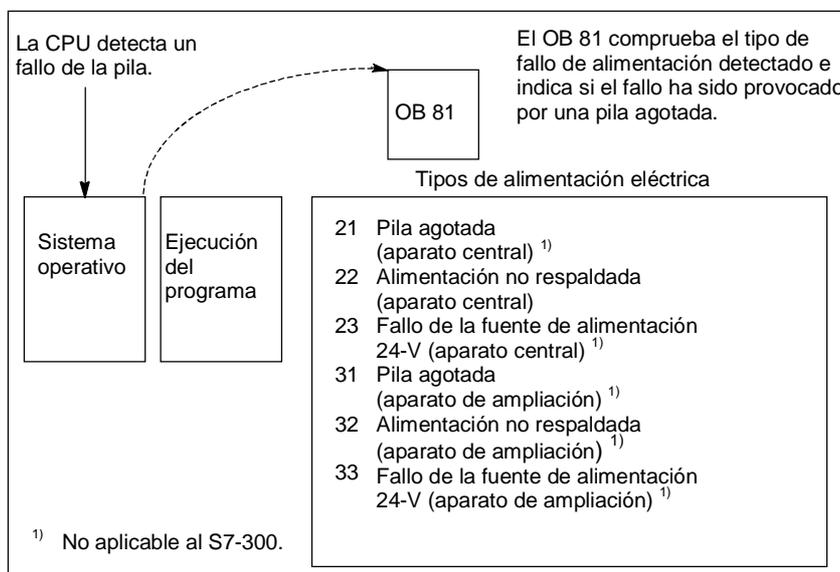
La tabla siguiente muestra qué tipos de errores ocurren generalmente. En el manual "Sistema de automatización S7-300, Configuración, instalación y datos de la CPU" o en el manual de referencia "Sistemas de automatización S7-400/M7-400, Configuración e instalación" puede consultar si su CPU asiste los OBs indicados.

Tipo de error	Clase de error	OB	Prioridad
Redundancia	Error de redundancia en periferia (sólo en CPUs H)	OB 70	25
	Error de redundancia en CPU (sólo en CPUs H)	OB 72	28
Asíncrono	Error de tiempo	OB 80	26
	Error de alimentación	OB 81	(ó 28, si el OB de error está en el programa de arranque)
	Alarma de diagnóstico	OB 82	
	Alarma de extraer/insertar	OB 83	
	Error de hardware CPU	OB 84	
	Error de ejecución del programa	OB 85	
	Error de bastidor	OB 86	
	Error de comunicación	OB 87	
Síncrono	Error de programación	OB 121	Prioridad del OB que ha ocasionado el error
	Error de acceso	OB 122	

### Ejemplo de aplicación del OB 81

Los datos locales (información de arranque) del OB de error permiten evaluar la clase del error ocurrido.

Si, p. ej., la CPU detecta un error de pila tampón, entonces el sistema operativo llama el OB 81 (v. fig. ).



Es posible escribir un programa que evalúe el código del evento que ha provocado la llamada del OB 81. También es posible escribir un programa que provoque una reacción determinada, p. ej., activar una salida conectada a una lámpara de la estación de operador.

### Datos locales del OB 81 de error

La tabla siguiente describe las variables temporales (TEMP) definidas en la tabla de declaración de variables del OB 81.

El símbolo *Fallo de la pila* (BOOL) también se deberá identificar como salida en la tabla de símbolos (p.ej. A 4.0), de manera que las demás partes del programa puedan acceder también a dichos datos.

Declaración	Designación	Tipo	Descripción
TEMP	OB81_EV_CLASS	BYTE	Clase de error/identificador de error 39xx
TEMP	OB81_FLT_ID	BYTE	Código de error: b#16#21 = Por lo menos una pila tampón del bastidor central vacía <sup>1</sup> b#16#22 = Falta tensión de respaldo en el bastidor central b#16#23 = Fallo de la alimentación de 24 V en el bastidor central <sup>1</sup> b#16#31 = Por lo menos una pila tampón vacía en un bastidor de ampliación <sup>1</sup> b#16#32 = Falta tensión de respaldo en un bastidor de ampliación <sup>1</sup> b#16#33 = Fallo de la alimentación de 24 V en un bastidor de alimentación <sup>1</sup>
TEMP	OB81_PRIORITY	BYTE	Prioridad = 26/28
TEMP	OB81_OB_NUMBR	BYTE	81 = OB 81
TEMP	OB81_RESERVED_1	BYTE	Reservado
TEMP	OB81_RESERVED_2	BYTE	Reservado
TEMP	OB81_MDL_ADDR	INT	Reservado
TEMP	OB81_RESERVED_3	BYTE	Sólo relevante para los códigos de error B#16#31, B#16#32, B#16#33
TEMP	OB81_RESERVED_4	BYTE	
TEMP	OB81_RESERVED_5	BYTE	
TEMP	OB81_RESERVED_6	BYTE	
TEMP	OB81_DATE_TIME	DATE_AND_TIME	Fecha y hora del arranque del OB
<sup>1)</sup> = No en el S7-300.			

## Programa de ejemplo para el OB de tratamiento de errores

Con un programa AWL se ejemplificamos cómo leer el código de error en el OB 81.

El programa está estructurado de la siguiente forma:

- El código de error en el OB 81 (OB81\_FLT\_ID) se lee y compara con el valor del evento "pila vacía" (B#16#3921).
- Si el código de error corresponde al código para "pila vacía", entonces el programa salta a la meta FPil y activa la salida Error de pila.
- Si el código de error no corresponde al código para "pila vacía", entonces el programa compara dicho código con el código para "fallo de pila".
- Si el código de error corresponde al código para "fallo de pila", entonces el programa salta a la meta FPil y activa la salida Fallo de pila. En caso contrario se finaliza el bloque.

AWL	Descripción
L B#16#3921	Comparar código de evento "pila vacía" (B#16#3921) con el código de error para el OB 81. Si es igual (pila está vacía), entonces saltar a BPil
L #OB81_FLT_ID	
== I	
SPB FPil	Comparar código de evento "fallo de pila" (b#16#3922) con el código de error para el OB 81.
L b#16#3922	
<> I	Si es diferente (no hay fallo de pila en el bastidor central), entonces concluir el bloque.
BEB	
BPil S #Fallo de la pila	FPil activa la salida fallo de pila cuando se produce un fallo de pila o hay una pila vacía

Para más información sobre los OBs, SFBs y SFCs y explicaciones sobre las ID de eventos consulte la Ayuda de bloques.

### 21.9.4 Insertar valores de sustitución al detectar errores

Para determinados tipos de error (p.ej. rotura de hilo en señal de entrada) se pueden asignar valores de sustitución para aquellos valores que quedan indisponibles a causa del error. Existen dos posibilidades para asignar valores de sustitución:

- Los valores de sustitución se pueden parametrizar con STEP 7 para módulos de salidas parametrizables. Los módulos de salidas no parametrizables tienen preajustado el valor de sustitución 0.
- La SFC 44 RPL\_VAL permite programar valores de sustitución en OBs de errores (sólo para módulos de entradas).

Para todas las instrucciones de carga que producen errores síncronos se puede asignar en el OB de error un valor de sustitución para el contenido del ACU.



Declaración	Designación	Tipo	Descripción
TEMP	OB122_EV_CLASS	BYTE	Clase de error/identificador de error 29xx
TEMP	OB122_SW_FLT	BYTE	Código de error: 16#42, 16#43, 16#44 <sup>1)</sup> , 16#45 <sup>1)</sup>
TEMP	OB122_PRIORITY	BYTE	Clase de prioridad = Prioridad del OB en el cual ha ocurrido el error
TEMP	OB122_OB_NUMBR	BYTE	122 = OB 122
TEMP	OB122_BLK_TYPE	BYTE	Tipo de bloque en el cual ha ocurrido el error
TEMP	OB122_MEM_AREA	BYTE	Area de memoria y tipo de acceso
TEMP	OB122_MEM_ADDR	WORD	Dirección de memoria en la cual ha ocurrido el error
TEMP	OB122_BLK_NUM	WORD	Número del bloque en el cual ha ocurrido el error
TEMP	OB122_PRG_ADDR	WORD	Dirección relativa de la instrucción que ha provocado el error
TEMP	OB122_DATE_TIME	DATE_AND_TIME	Fecha y hora del arranque del OB
TEMP	Error	INT	Memoriza el código de error de la SFC44

<sup>1)</sup> = No en el S7-300.

AWL	Descripción
<pre> L      B#16#2942 L      #OB122_SW_FLT ==I SPB   QFeh L      B#16#2943 &lt;&gt; I SPB   Stop  QFeh: CALL "REPL_VAL" VAL := DW#16#2912 RET_VAL := #Error L      #Error L      0 ==I BEB  Stop: CALL "STP" </pre>	<p>Comparar el código de evento del OB 122 con el código de evento (B#16#2942) para el acuse de un error de tiempo al leer la periferia. Si es igual, saltar a "QFeh".</p> <p>Comparar el código de evento del OB 122 con el código de evento (B#16#2943) para un error de direccionamiento (escribir en un módulo que no existe). Si es diferente, saltar a "Stop".</p> <p>Meta "QFeh": transfiere DW#16#2912 (binario 10010) a la SFC44 (REPL_VAL). La SFC44 carga este valor en el ACU 1 (y sustituye el valor que ha provocado la llamada del OB 122). Memoriza el código de error de la SFC en #Error.</p> <p>Compara #Error con 0 (si es igual, entonces no ha ocurrido ningún error al tratar el OB 122). Finalizar el bloque si no ha ocurrido ningún error.</p> <p>Meta "Stop": llama la SFC46 "STP" y lleva la CPU al estado operativo STOP.</p>

## 21.9.5 Error de redundancia en periferia (OB 70)

### Descripción

El sistema operativo de la CPU H llama el OB 70 al presentarse una pérdida de redundancia en el DP PROFIBUS (p. ej. en caso de un fallo de bus del maestro DP activo o de un error en el módulo de interconexión (interfase) del esclavo DP) o al cambiar el maestro DP activo de esclavos DP con periferia conmutada.

### Programar el OB 70

El OB 70 se debe generar como objeto en el programa STEP 7. Escribir el programa, que se debe ejecutar en el OB 70, en el bloque generado y cargarlo en la CPU como parte del programa de usuario.

El OB 70 se puede utilizar, p. ej., para:

- evaluar la información de arranque del OB 70 y averiguar qué evento ha causado la pérdida de redundancia en periferia;
- mediante la SFC 51 RDSYSST, averiguar el estado de su sistema (SZL\_ID=B#16#71).

La CPU no cambia a estado operativo "STOP" si se presenta un error de redundancia en periferia y si el OB 70 no está programado.

Si el OB 70 está cargado y el sistema H se encuentra en modo redundante, el OB 70 se procesará en ambas CPUs. El sistema H permanece en modo redundante.

Para más información sobre los OBs, SFBs y SFCs consulte la Ayuda de bloques

## 21.9.6 Error de redundancia en CPU (OB 72)

### Descripción

El sistema operativo de la CPU H llama el OB 72 al presentarse uno de los eventos siguientes:

- Pérdida de redundancia de las CPUs
- Error de comparación (p.ej., RAM, PAA)
- Conmutación reserva/maestro
- Error de sincronización
- Error en un módulo SYNC
- Interrupción del proceso de igualación de datos
- El OB 72 es ejecutado por todas las CPUs que se encuentren en estado operativo RUN o ARRANQUE tras el correspondiente evento de arranque.

## Programar el OB 72

El OB 72 se debe generar como objeto en el programa STEP 7. Escriba el programa, que se debe ejecutar en el OB 72, en el bloque generado y cárguelo en la CPU como parte del programa de usuario.

El OB 72 se puede utilizar, p. ej., para:

- evaluar la información de arranque del OB 72 y averiguar qué evento ha causado la pérdida de redundancia en la CPU;
- averiguar el estado de su sistema (SZL\_ID=B#16#71) con ayuda de la SFC 51 RDSYSST;
- reaccionar a la pérdida de redundancia en la CPU conforme a la instalación.

La CPU no cambia a estado operativo "STOP" si se presenta un error de redundancia en la CPU y si el OB 72 no está programado.

Encontrará información más detallada acerca de los OBs, SFBs y SFCs en la Ayuda de bloques.

## 21.9.7 Error de redundancia de comunicación OB 73

### Descripción

El sistema operativo de la CPU H llama el OB 73 la primera vez que se pierde la redundancia de un enlace S7 de alta disponibilidad (únicamente hay enlaces S7 de alta disponibilidad en la comunicación S7, véase "Sistema de automatización S7-400 Sistemas H de alta disponibilidad"). Si se pierde la redundancia de otros enlaces S7 de alta disponibilidad no se produce un nuevo arranque del OB73.

Sólo se producirá un nuevo arranque del OB73 cuando se haya reestablecido la redundancia de todos los enlaces S7 que tenían alta disponibilidad.

La CPU no cambia a estado operativo STOP si se produce el evento de arranque correspondiente y si el OB73 no está programado.

### Programar el OB 73

El OB 73 se debe generar como objeto en el programa STEP 7. Escriba el programa, que se debe ejecutar en el OB 73, en el bloque generado y cárguelo en la CPU como parte del programa de usuario.

El OB 73 se puede utilizar, p.ej., para:

- evaluar la información de arranque del OB 73 y averiguar qué evento ha causado la pérdida de redundancia de la periferia.
- Averiguar el estado de su sistema (SZL\_ID=B#16#71) con ayuda de la SFC 51 RDSYSST.

La CPU no cambia a estado operativo "STOP" si se presenta un error de redundancia de comunicación y si el OB 73 no está programado.

Si el OB 73 está cargado y el sistema H se encuentra en modo de operación redundante, el OB 73 se ejecuta en ambas CPUs. El sistema H permanece en modo de operación redundante.

Encontrará información más detallada acerca de los OBs, SFBs y SFCs en la Ayuda de bloques.

## 21.9.8 Error de tiempo (OB 80)

### Descripción

El sistema operativo de la CPU llama el OB 80 cuando ha ocurrido un error de tiempo. Errores de tiempo pueden ser, p. ej.:

- exceder el tiempo de ciclo máximo
- saltar alarmas horarias por adelanto de la hora
- retardo demasiado grande al tratar una prioridad

### Programar el OB 80

El OB 80 se debe generar como objeto en el programa S 7 con ayuda de STEP 7. Escribir el programa, que se debe ejecutar en el OB 80, en el bloque generado y cargarlo en la CPU como parte del programa de usuario.

El OB 80 se puede utilizar, p. ej., para:

- evaluar la información de arranque del OB 80 y averiguar qué alarmas horarias se han saltado
- desactivar con la SFC 29 CAN\_TINT la alarma horaria saltada, para que ésta no se ejecutada y el tratamiento de las alarmas horarias se pueda continuar correctamente a la hora nuevamente ajustada.

Si las alarmas horarias saltadas en el OB 80 no se desactivan, entonces se ejecuta la primera alarma horaria saltada y se ignoran todas las demás.

Si no se programa el OB 80, entonces la CPU pasa al estado operativo "STOP" al detectarse un error de tiempo.

Para más información sobre los OBs, SFBs y SFCs consulte la Ayuda de bloques

## 21.9.9 Fallo de alimentación (OB 81)

### Descripción

El sistema operativo de la CPU llama el OB 81 cuando en el bastidor central o en un bastidor de ampliación ha fallado

- la alimentación de 24 V,
- una pila,
- todo el respaldo por pila

o se ha eliminado una perturbación (llamada en caso de evento entrante y saliente).

## Programar el OB 81

El OB 81 se debe generar como objeto en el programa S 7 con ayuda de STEP 7. Escriba el programa, que se debe ejecutar en el OB 81, en el bloque generado y cárguelo en la CPU como parte del programa de usuario.

El OB 81 se puede utilizar, p. ej., para

- evaluar la información de arranque del OB 81 y averiguar qué fallo de alimentación existe
- determinar el número del bastidor con fallo de alimentación
- activar una lámpara en una estación de operador para visualizar al personal de mantenimiento que se ha de sustituir una pila.

Al contrario que otros OBs de errores asíncronos, si no se programa el OB 81 la CPU no pasará, al estado operativo STOP al detectarse un fallo de alimentación. No obstante, el fallo se registra en el búfer de diagnóstico y el LED correspondiente visualiza dicho fallo en el panel frontal.

Para más información sobre los OBs, SFBs y SFCs consulte la Saltos a descripciones de lenguajes, ayuda de bloques y atributos del sistema

## 21.9.10 Alarma de diagnóstico (OB 82)

### Descripción

El sistema operativo de la CPU llama el OB 82 cuando en un módulo diagnosticable, para el cual se ha habilitado la alarma de diagnóstico, detecta un error y al eliminarse dicho error (llamada en caso de evento entrante y saliente).

### Programar el OB 82

El OB 82 se debe generar como objeto en el programa S7 con ayuda de STEP 7. Escribir el programa, que se debe ejecutar en el OB 82, en el bloque generado y cargarlo en la CPU como parte del programa de usuario.

El OB 82 se puede utilizar, p. ej., para

- evaluar la información de arranque del OB 82
- efectuar un diagnóstico exacto del error ocurrido.

Cuando se dispara una alarma de diagnóstico, el módulo averiado registra automáticamente 4 bytes de datos de diagnóstico, así como su dirección inicial, en la información de arranque del OB de tratamiento de alarmas de diagnóstico y en el búfer de diagnóstico. Esto permite saber cuándo y en qué módulo ocurrió el error.

Otros datos de diagnóstico del módulo averiado (en qué canal ha ocurrido el error, de qué tipo de error se trata) se pueden evaluar con un programa correspondiente en el OB82. La SFC 51 RDSYSST permite leer los datos de diagnóstico del módulo y con la SFC 52 WR\_USRMSG se pueden registrar estas informaciones en el búfer de diagnóstico. Además, el mensaje de diagnóstico autodefinido adicionalmente se puede enviar a un visualizador registrado.

Si no se programa el OB 82, entonces la CPU cambia al estado de operación "STOP" al activarse una alarma de diagnóstico.

Para más información sobre los OBs, SFBs y SFCs consulte la Ayuda de bloques

## 21.9.11 Alarma de presencia de módulo (OB 83)

### Descripción

Las CPUs S7-400 vigilan cíclicamente en intervalos de aprox. 1 segundo si se han extraído e insertado módulos en el bastidor central y en los bastidores de ampliación.

Tras conectar la tensión de alimentación, la CPU comprueba si todos los módulos listados en la tabla de configuración creada usando STEP 7 están realmente insertados. Si es así, esta configuración real se memoriza y sirve como valor de referencia para la vigilancia cíclica de los módulos. En cada ciclo de consulta, se compara la nueva configuración real determinada con la configuración real anterior. En caso de diferencias, se dispara una alarma de extraer/insertar y se registra una entrada tanto en el búfer de diagnóstico como en la lista de estados del sistema. En el estado operativo RUN se arranca el OB de presencia de módulo.

---

### Nota

En el estado de operación RUN no se pueden extraer los módulos de alimentación ni las CPUs e IMs.

Entre la extracción e inserción de un módulo deben transcurrir por lo menos 2 s, para que dichas operaciones pueden ser detectadas correctamente por la CPU.

---

### Parametrizar un módulo una vez insertado

Si se inserta un módulo durante el estado operativo RUN, la CPU comprueba si el tipo del módulo que ha sido insertado coincide con el tipo del módulo originalmente insertado. Si coinciden, se efectúa la parametrización. Se transfieren bien los parámetros predeterminados o bien los parámetros asignados con STEP 7.

### Programar el OB 83

Cree primero OB 83 en el programa S7. Escriba el programa, que se debe ejecutar en el OB 83, en el bloque generado y cárguelo en la CPU como parte del programa de usuario.

El OB 83 se puede utilizar, p. ej., para

- evaluar la información de arranque del OB 83
- reparametrizar el módulo insertado con las funciones del sistema SFC 55 a 59.

Si no se programa el OB 83, la CPU cambiará de RUN al estado operativo STOP al dispararse una alarma de extraer/insertar.

Encontrará información detallada sobre OBs, SFBs y SFCs en Saltos a descripciones de lenguajes, ayudas de bloques y atributos de sistema

### 21.9.12 Fallo de CPU (OB 84)

#### Descripción

El sistema operativo de la CPU llama el OB 84 cuando se detecta un fallo en el interface a la red MPI, al bus de comunicación o al módulo de interconexión (interfase) para la periferia descentralizada, p.ej. nivel de señal erróneo en el cable o cuando se elimina el error (llamada en caso de evento entrante y saliente).

#### Programar el OB 84

El OB 84 se debe generar como objeto en el programa S7 con ayuda de STEP 7. Escriba el programa que se debe ejecutar en el OB 84, en el bloque generado y cárguelo en la CPU como parte del programa de usuario.

El OB 84 se puede utilizar, p. ej., para

- evaluar la información de arranque del OB 84
- enviar un mensaje al búfer de diagnóstico a través de la función del sistema SFC 52 WR\_USMSG.

Si no se programa el OB 84, la CPU cambiará al estado operativo STOP al detectarse un fallo de hardware de la CPU.

Para más información sobre los OBs, SFBs y SFCs consulte la Ayuda de bloques

### 21.9.13 Error de ejecución del programa (OB 85)

#### Descripción

El sistema operativo de la CPU llama el OB 85 cuando

- existe un evento de arranque para un OB de alarma, pero el OB no se puede ejecutar porque no ha sido cargado en la CPU
- ha ocurrido un error al acceder al bloque de datos de instancia de un bloque de funciones del sistema
- ha ocurrido un error al actualizar la imagen de proceso (módulo no existente o defectuoso).

#### Programar el OB 85

El OB 85 se debe generar como objeto en el programa S7 con ayuda de STEP 7. Escriba en el bloque generado el programa que se debe ejecutar en el OB 85 y cárguelo en la CPU como parte del programa de usuario.

El OB 85 se puede utilizar, p. ej., para

- evaluar la información de arranque del OB 85 y determinar qué módulo está defectuoso o falta (indicación de la dirección inicial del módulo)
- determinar el puesto/slot del módulo correspondiente con la SFC 49 LGC\_GADR.

Si no se programa el OB 85, entonces la CPU pasa al estado operativo "STOP" al detectarse un error de prioridad.

## 21.9.14 Fallo en el bastidor (OB 86)

### Descripción

El sistema operativo de la CPU llama el OB 86 al detectarse un fallo de bastidor p. ej., en caso de

- fallo de bastidor (módulo IM defectuoso o faltante o cable de conexión interrumpido)
- fallo de tensión a nivel de bastidor
- fallo de un esclavo de periferia descentralizada en un sistema maestro del sistema de bus PROFIBUS-DP

o cuando se ha eliminado el fallo (llamada en caso de evento entrante y saliente).

### Programar el OB 86

Cree en primer lugar el objeto OB 86 en el programa S7. Escriba en el bloque generado el programa que se debe ejecutar en el OB 86 y cárguelo en la CPU como parte del programa de usuario.

El OB 86 se puede utilizar, p. ej., para

- evaluar la información de arranque del OB 86 y determinar qué bastidor está defectuoso o falta.
- enviar un mensaje al búfer de diagnóstico y a un visualizador a través de la función del sistema SFC 52 WR\_USMSG.

Si no se programa el OB 86, la CPU cambia al estado operativo STOP al detectarse un fallo de bastidor.

Para más información sobre los OBs, SFBs y SFCs consulte la Ayuda de bloques

## 21.9.15 Error de comunicación (OB 87)

### Descripción

El sistema operativo de la CPU llama el OB 87 al ocurrir un error de comunicación durante el intercambio de datos a través de bloques de función para comunicaciones o de comunicación de datos globales, p. ej.,

- se detectó un identificador de telegrama erróneo durante la recepción de datos globales
- el bloque de datos para la información de estado de los datos globales no existe o es demasiado corto.

## Programar el OB 87

El OB 87 se debe generar como objeto en el programa S7 con ayuda de STEP 7. Escribir el programa, que se debe ejecutar en el OB 87, en el bloque generado y cargarlo en la CPU como parte del programa de usuario.

El OB 87 se puede utilizar, p. ej., para

- evaluar la información de arranque del OB 87 y
- crear un bloque de datos cuando falta el bloque de datos para la información de estado de la comunicación de datos globales.

Si no se programa el OB 87, la CPU cambia al estado operativo "STOP" al detectarse un error de comunicación.

Para más información sobre los OBs, SFBs y SFCs consulte la Ayuda de bloques

## 21.9.16 Error de programación (OB 121)

### Descripción

El sistema operativo de la CPU llama el OB 121 cuando ocurre un error de programación, como p. ej., cuando

- no existen temporizadores direccionados
- el bloque llamado no está cargado

### Programar el OB 121

El OB 121 se debe generar como objeto en el programa S7 con ayuda de STEP 7. Escriba el programa, que se debe ejecutar en el OB 121, en el bloque generado y cárguelo en la CPU como parte del programa de usuario.

El OB 121 se puede utilizar, p. ej., para

- evaluar la información de arranque del OB 121
- registrar la causa del error en un bloque de datos de mensajes.

Si no se programa el OB 121, la CPU cambia al estado operativo "STOP" al detectarse un error de programación.

Para más información sobre los OBs, SFBs y SFCs consulte la Ayuda de bloques

## 21.9.17 Error de acceso a la periferia (OB 122)

### Descripción

El sistema operativo de la CPU llama el OB 122 cuando se accede con una operación STEP 7 a una entrada o salida de un módulo de señales que no estaba asignada a ningún módulo en el momento de efectuar el último re arranque completo, p. ej.,

- error al acceder directamente a la periferia (módulo defectuoso o no existente)
- acceso a una dirección de periferia desconocida por la CPU.

### Programar el OB 122

El OB 122 se debe generar como objeto en el programa S7 con ayuda de STEP 7. Escriba en el bloque generado el programa que se debe ejecutar en el OB 122 y cárguelo en la CPU como parte del programa de usuario.

El OB 122 puede utilizar, p. ej., para

- evaluar la información de arranque del OB 122
- llamar la función del sistema SFC 44 y asignar un valor sustitutivo para un módulo de entradas, para que el programa pueda seguir corriendo con un valor razonable, adecuado al proceso.

Si no se programa el OB 122, la CPU cambiará al estado operativo "STOP" al detectarse un error de acceso a periferia.

Para más información sobre los OBs, SFBs y SFCs consulte la Ayuda de bloques.



## 22 Imprimir y archivar

### 22.1 Imprimir la documentación de un proyecto

#### 22.1.1 Imprimir la documentación de un proyecto

Una vez creado el programa para la solución de automatización, es posible imprimir todos los datos importantes para documentar el proyecto utilizando la función de impresión integrada en STEP7.

#### Componentes imprimibles del proyecto

El contenido de los objetos se puede imprimir bien directamente desde el Administrador SIMATIC o bien abriendo el objeto deseado y activando el comando de impresión.

Desde el Administrador SIMATIC se pueden imprimir automáticamente los siguientes componentes de un proyecto:

- árbol de objetos (estructura del proyecto/de la librería),
- listas de objetos (contenido de una carpeta de objetos),
- contenido de un objeto y
- mensajes

Abriendo el objeto en cuestión se pueden imprimir p.ej. los siguientes componentes del proyecto:

- los bloques en lenguaje KOP, FUP, AWL u otro lenguaje (software opcional),
- la tabla de símbolos con los nombres simbólicos de las direcciones absolutas,
- la tabla de configuración con la disposición de los módulos en el PLC y los parámetros de los mismos,
- el contenido del búfer de diagnóstico,
- la tabla de variables con los formatos de estado, así como los valores de estado y de forzado,
- los datos de referencia, es decir, las listas de referencias cruzadas, los planos de ocupación, las estructuras del programa, los operandos no utilizados y los operandos sin símbolo,
- la tabla de datos globales,
- los datos de los módulos e información sobre su estado,
- textos de usuario (textos de usuario y librerías de textos),
- los documentos del software opcional, p.ej. de lenguajes de programación

## Paquete opcional DOCPRO

Para crear, editar e imprimir manuales de circuitos normalizados está disponible el paquete opcional DOCPRO. Dicho paquete permite crear una documentación conforme a las normas DIN y ANSI.

### 22.1.2 Procedimiento básico para imprimir

Para imprimir un componente de un proyecto:

1. Abra el objeto en cuestión para visualizar los datos que desea imprimir.
2. Active el cuadro de diálogo "Imprimir" con el comando **Archivo > Imprimir...** en la respectiva ventana. Dependiendo de la ventana, el primer menú de la barra de menús puede tener un nombre diferente, p.ej. "Tabla de símbolos", en vez de "Archivo".
3. En caso necesario, cambie en el cuadro de diálogo los ajustes de impresión (p.ej. impresora, zona de impresión, número de copias) y cierre luego dicho cuadro.

Algunos cuadros de diálogo contienen el botón de comando "Imprimir", p. ej., el cuadro de diálogo 'Información del módulo'. Para imprimir el contenido del cuadro de diálogo, haga clic en este botón.

No es necesario abrir los bloques. Estos se pueden imprimir directamente desde el Administrador SIMATIC con el comando **Archivo > Imprimir...** .

### 22.1.3 Funciones para imprimir

Para imprimir objetos se dispone de las siguientes funciones complementarias:

Objetos de impresión	Comando de menú	Función	Función	Función	Función
		Presentación preliminar	Preparar página	Encabezado/ Pie de página	Instalar impresora
Bloques, fuentes AWL	Archivo > *	•	•	–	•
Información del módulo		–	•	–	–
Tabla de datos globales	Tabla GD > *	•	•	–	•
Tabla de configuración	Equipo > *	•	•	–	•
Objeto, carpeta de objetos	Archivo > *	–	•	•	•
Datos de referencia	Datos de referencia > *	•	•	–	•
Tabla de símbolos	Tabla > *	•	•	–	•

Objetos de impresión	Comando de menú	Función	Función	Función	Función
Tabla de variables	Tabla > *	–	•	–	•
Tabla de enlaces	Red > *	•	•	–	•
Listas de textos de usuario (textos de usuario y librerías de textos)	Textos > *	•	•	–	•

\* : El caracter \* es un comodín en el comando de menú para la función correspondiente (p. ej., Presentación preliminar o Prepara página)

Las instrucciones paso a paso para imprimir los objetos las encontrará bajo:

Pasos para imprimir .

### Presentación preliminar

La función "Presentación preliminar" permite visualizar el documento tal y como será impreso.

Si la impresión se compone de varias páginas, en el margen inferior derecho de la hoja, tras el número de página, aparece un identificador de sucesión. La última página no tiene dicho identificador.

---

#### Nota

La imagen de la impresión optimizada no se muestra en la presentación preliminar.

---

### Ajustar el formato de página

Con la función "Preparar página" del Administrador SIMATIC se ajusta el formato de página para todas las aplicaciones y paquetes opcionales de STEP 7. No obstante, en las distintas aplicaciones, p. ej. en el editor de símbolos, es posible ajustar temporalmente otros formatos, es decir, mientras dure la sesión actual, no pudiéndose memorizar dicho ajuste para otras sesiones.

Ajuste el formulario a imprimir al tamaño de papel deseado. Si el formulario es demasiado ancho, el margen derecho se imprimirá en la página siguiente.

Si elige el formato de página con borde (p.ej., "borde A4"), el documento se imprimirá dejando un borde en la página izquierda que se puede utilizar para el taladrado.

---

#### Nota

Si necesita ayuda para el cuadro de diálogo "Preparar página", haga clic en el botón "Ayuda" o pulse la tecla F1 mientras esté abierto el cuadro de diálogo.

---

## Ajustar encabezados y pies de página

Con la función "**Archivo > Encabezado/Pie de página**" del Administrador SIMATIC se pueden ajustar los encabezados y pies de página para los documentos a imprimir de un proyecto determinado. En las aplicaciones sólo se define el formato de página. Si la impresión se compone de varias páginas, en el margen inferior derecho de la hoja aparece un identificador de sucesión. La última página no tiene dicho identificador. De este modo se ve rápidamente si falta algo por imprimir. Este identificador de sucesión también aparece en la presentación preliminar.

## Instalar impresora

Con la función "Instalar impresora" se puede elegir una impresora y ajustar el formato del papel (vertical u horizontal). Las posibilidades de ajuste dependen del driver de impresora que se utilice.

### 22.1.4 Particularidades al imprimir el árbol de objetos

En el cuadro de diálogo "Imprimir lista de objetos" puede seleccionar el cuadro de opción "Árbol", con objeto de imprimir no sólo dicha lista, sino también el árbol de objetos.

Si en "Imprimir" selecciona la opción "Todo", se imprimirá toda la estructura del árbol. Si se elige "Selección", se imprimirá la estructura del árbol a partir del objeto seleccionado hacia abajo.

---

#### Nota

Los ajustes efectuados en el cuadro de diálogo son válidos únicamente para la impresión de la lista o del árbol, pero no para los contenidos. Para ello se utilizan los ajustes de las aplicaciones en cuestión.

---

## 22.2 Archivar proyectos y librerías

### 22.2.1 Archivar proyectos y librerías

Los proyectos y las librerías se pueden guardar de forma comprimida en un archivo comprimido. El archivero puede copiarse en disco duro o en soportes de datos portátiles (p.ej. disquetes).

#### Programas archiveros

La función de compresión (archivado) ofrece un interface para ejecutar el programa de compresión que usted prefiera. Los programas archiveros ARJ y PKZIP 2.50 forman parte del volumen de suministro de STEP 7. Para poder utilizar un determinado programa archivero, se requiere la siguiente versión (o una posterior):

- ARJ a partir de la versión 2.4.1a
- PKZIP a partir de la versión 2.04g
- LHARC a partir de la versión 2.13
- WinZip a partir de la versión 6.0
- JAR a partir de la versión 1.02

#### Recomendación para archivar

Los proyectos que contengan "nombres de archivo largos" (que excedan lo establecido por la convención del DOS, versión 8.3) o que contengan árboles de directorios muy ramificados (directorios cuyo nombre de ruta supere los 64 caracteres) se deberán comprimir con los programas archiveros PKZIP 2.50, WinZip o JAR. Los demás programas archiveros no garantizan una descompresión correcta o completa. Esto rige especialmente para proyectos que contienen objetos del paquete opcional.

### 22.2.2 Utilización de las funciones 'Guardar / Archivar'

#### Guardar como

Con esta función se puede crear una **copia** del proyecto con otro nombre.

Esta función se puede utilizar

- al crear copias de seguridad
- al duplicar un proyecto ya existente para adaptarlo a otras necesidades.

Para crear una copia de la forma más rápida posible, seleccione en el cuadro de diálogo que aparece a continuación 'Guardar sin reorganizar'. Así se copiará el árbol completo del archivo desde el directorio del proyecto sin comprobar si es coherente y se guardará con otro nombre.

En el soporte de datos debe haber memoria suficiente para depositar la copia de seguridad. No intente guardar proyectos en un disquete, puesto que normalmente la memoria no es suficiente. Para transportar datos del proyecto a disquetes utilice la función "Archivar".

La función 'Guardar reorganizando' dura más, pero si no se puede copiar o guardar un objeto aparecerá un mensaje indicándolo. Las causas se pueden deber a que falta el paquete opcional o a que los datos de un objeto son defectuosos.

## Archivar

Es posible guardar proyectos o librerías en archivos comprimidos en archivadores. Dicho archivo comprimido puede copiarse en disco duro o en soportes de datos portátiles (p.ej. disquetes).

Si desea transportar proyectos a disquetes, hágalo sólo en forma de archivos comprimidos. Si el proyecto es demasiado grande, elija un programa archivador con el que pueda crear archivadores de múltiples disquetes.

Los proyectos o librerías que se hayan comprimido en un archivador no se pueden utilizar. Si desea procesar nuevamente dichos proyectos o librerías, deberá descomprimir los datos, es decir desarchivar el proyecto o la librería en cuestión.

### 22.2.3 Requisitos para poder archivar

Para archivar un proyecto o una librería se deberán cumplir los siguientes requisitos:

- El programa archivador deberá estar instalado en su sistema. La incorporación en STEP 7 se indica en el tema de la Ayuda "Pasos para archivar y desarchivar".
- Sin excepción alguna, todos los datos del proyecto deberán estar depositados directamente en el directorio del proyecto o en uno de sus subdirectorios. Aunque, al trabajar en el entorno de desarrollo C, se podrían depositar los datos en otro lugar, en tal caso éstos no se incorporarían en el archivo comprimido.
- Los nombres de los archivos deberán corresponder a las convenciones del DOS respecto a los nombres (8 caracteres más 3 caracteres para la extensión del archivo), si desea trabajar con uno de los programas archivadores ARJ, PKZIP (versión 2.04g) o LHArc, puesto que éstos últimos son programas del DOS. Dicha limitación no afecta a PKZip (versión 2.50), Jar y WinZip.

### 22.2.4 Procedimiento para archivar y desarchivar

Para archivar y desarchivar un proyecto o una librería, utilice los comandos de menú **Archivo > Archivar** y **Archivo > Desarchivar**, respectivamente.

---

#### Nota

Los proyectos o librerías que se hayan comprimido en un archivador no pueden ser utilizadas. Si desea procesar nuevamente dichos proyectos o librerías, deberá descomprimir los datos, es decir desarchivar el proyecto o la librería en cuestión.

---

Al desarchivar los proyectos y librerías se integran automáticamente en la lista de proyectos/librerías.

### **Ajustar el directorio de destino**

Para ajustar el directorio de destino, elija en el Administrador SIMATIC el comando de menú **Herramientas > Preferencias** para visualizar el cuadro de diálogo "Preferencias".

En la ficha "Archivar" de este cuadro de diálogo puede activar y desactivar la opción "Consultar el directorio de destino al desarchivar".

Si dicha opción está desactivada, al desarchivar se utilizará como ruta de destino la ruta indicada en la ficha "General" de este cuadro de diálogo (Ruta para proyectos y Ruta para librerías).

### **Copiar un archivo comprimido en un disquete**

Asimismo, es posible comprimir un proyecto o una librería y copiar luego el archivo comprimido en un disquete. Además, también se puede elegir directamente una unidad de disquete en el cuadro de diálogo "Archivar".

### **Indicaciones para desarchivar con PKZIP 2.04g**

Si al crear un archivo en disquete con el programa de compresión PKZIP se activó la opción "Archivador de múltiples disquetes", al desarchivar se le pedirá que introduzca el último disquete del archivador. PKUNZIP visualiza siempre el siguiente mensaje en la ventana del DOS:

`Insert the LAST disk of the backup set - Press a key when ready.`

Este mensaje se visualizará también si el archivo comprimido creado con la "Archivador de múltiples disquetes" se ha logrado copiar en un solo disquete.

En tal caso ignore el mensaje y confirme el mensaje pulsando una tecla cualquiera.



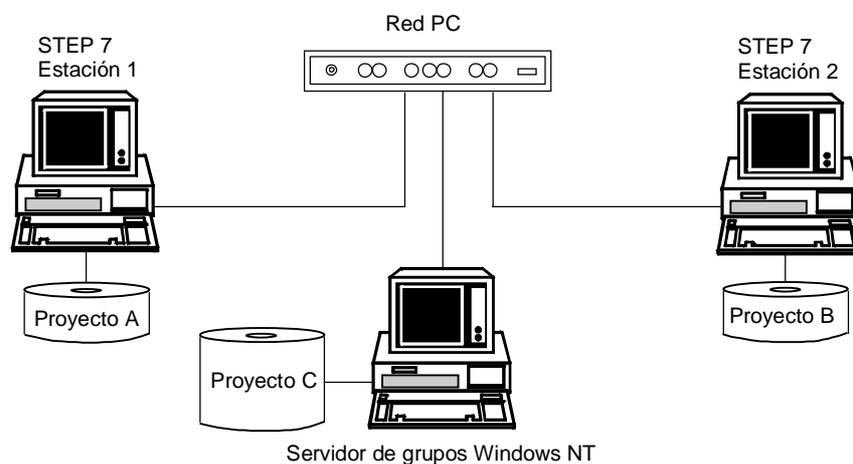
## 23 Configuración multiusuario en una red Windows

### 23.1 Configuración multiusuario en una red Windows

#### Introducción

STEP 7 se puede utilizar en una configuración multiusuario en grupos de trabajo Windows 95/98/NT/2000 y en redes NT o Novell. Se puede trabajar con tres métodos fundamentales:

- El proyecto se encuentra en una unidad de disco local y es utilizado también por otra estación de trabajo.  
**Ejemplo:** las estaciones de trabajo 1 y 2 acceden al proyecto A de la estación de trabajo 1.
- El proyecto se encuentra en un servidor de proyectos o de red.  
**Ejemplo:** las estaciones de trabajo 1 y 2 acceden al proyecto C del servidor de red.
- Los proyectos están repartidos en diversas unidades de disco locales y en uno o varios servidores de proyectos o de red.  
**Ejemplo:** las estaciones de trabajo 1 y 2 acceden a los proyectos A, B y C.



### Reglas para depositar proyectos en servidores de red

- Si desea depositar sus proyectos en servidores de red, la ruta de acceso deberá estar asociada siempre a una letra de unidad de disco.
- Si deposita sus proyectos en servidores de red o en unidades de disco liberadas de otros usuarios de la red, Windows 95/98/NT/2000 sólo se podrá finalizar en dichos servidores o unidades tras haberse cerrado todas las aplicaciones de STEP 7 que accedan a dichos proyectos.

### Reglas para compartir la edición de un programa S7

Tenga en cuenta lo siguiente:

- Antes de que varias personas puedan compartir la edición de un programa S7, será preciso ajustar configuración de las estaciones de trabajo (**Inicio > Simatic > STEP 7 > Configurar estación de trabajo**).
- Bloques y fuentes AWL:  
Cada persona debería programar un bloque o una fuente AWL diferente. Si dos personas intentan editar simultáneamente un bloque o una fuente, se visualizará un mensaje y se impedirá el acceso a la segunda persona.
- Tabla de símbolos:  
Varias personas pueden abrir simultáneamente la tabla de símbolos, pero sólo una de ellas la puede editar. Si dos personas intentan editar simultáneamente la tabla de símbolos, se visualizará un mensaje y se impedirá el acceso a la segunda persona.
- Tabla de variables:  
Varias personas pueden abrir simultáneamente la tabla de variables, pero sólo una de ellas la puede editar. Si dos personas intentan editar simultáneamente la tabla de variables, se visualizará un mensaje y se impedirá el acceso a la segunda persona. Un programa S7 puede contener varias tablas de variables. Claro está que éstas se pueden editar de forma independiente.

### Reglas para compartir la edición de un equipo

Tenga en cuenta lo siguiente:

- La configuración del hardware y la configuración de red de un equipo deberán ser editadas por una sola persona.

## 24 Trabajar con sistemas de automatización M7

### 24.1 Procedimiento para sistemas M7

Gracias a su arquitectura PC estandarizada, los microcomputadores industriales M7-300/400 constituyen una extensión programable de los sistemas de automatización SIMATIC. Los programas de usuario para SIMATIC M7 se pueden crear en un lenguaje de alto nivel, tal como C, o bien gráficamente con CFC (Continuous Function Chart).

Para crear los programas se necesita, además de STEP 7, el software de sistema M7-SYS RT para M7-300/400, así como un entorno de desarrollo para programas M7 (ProC/C++ o CFC).

#### Procedimiento básico

Al crear una solución de automatización con SIMATIC M7 se deben realizar los siguientes trabajos. La siguiente tabla muestra los trabajos que se deben realizar en la mayoría de los proyectos y los asigna a un procedimiento general en forma de guía de orientación. En la tabla se hace referencia a los correspondientes capítulos del presente manual o a otros manuales.

Procedimiento	Descripción
1. Planificar una solución de automatización	Específico de M7; Ver: PHB M7-SYS RT
2. Arrancar STEP 7	Igual que en STEP 7
3. Crear la estructura del proyecto	Igual que en STEP 7
4. Parametrizar el equipo	
5. Configurar el hardware	
6. Configurar los enlaces de comunicación	Igual que en STEP 7
7. Definir la tabla de símbolos	Igual que en STEP 7
8. Crear programa de usuario en C o CFC	Específico de M7; Ver: ProC/C++
9. Configurar el sistema operativo	Específico de M7; Ver: BHB M7-SYS Rt
10. Instalar el sistema operativo en M7-300/400	
11. Cargar la configuración del hardware y el programa de usuario en M7	
12. Comprobar y depurar el programa de usuario	ProC/C++
13. Vigilar el funcionamiento y diagnosticar M7	Igual que en STEP 7, pero sin diagnóstico definido por el usuario
14. Imprimir y archivar	Igual que en STEP 7

### **Diferencias con respecto al S7**

STEP 7 no asiste las siguientes funciones para los sistemas M7-300/400:

- Modo multiprocesador – funcionamiento síncrono de varias CPUs
- Forzado permanente
- Comunicación de datos globales
- Diagnóstico personalizado

### **Gestionar los sistemas de destino M7**

STEP 7 le asiste especialmente en las siguientes tareas con sistemas de automatización M7:

- Instalar el sistema operativo en el M7-300/400
- Configurar el sistema operativo editando archivos de sistema
- Transferir programas de usuario al M7-300/400
- Actualizar el firmware

Para acceder a la gestión de sistemas de destino M7, elija el siguiente comando de menú desde un proyecto que contenga equipos con CPUs o FMs M7, estando seleccionada la carpeta de programas M7:

#### **Sistema de destino > Gestionar sistema de destino M7**

La Ayuda en pantalla y el manual de usuario M7-SYS RT contienen instrucciones más detalladas.

## 24.2 Software opcional para la programación M7

### Software opcional M7

STEP 7 ofrece las funciones básicas para:

- crear y gestionar proyectos,
- configurar y parametrizar el hardware,
- configurar redes y enlaces, así como
- gestionar datos simbólicos.

Estas funciones son aplicables a los sistemas de automatización SIMATIC S7 y SIMATIC M7.

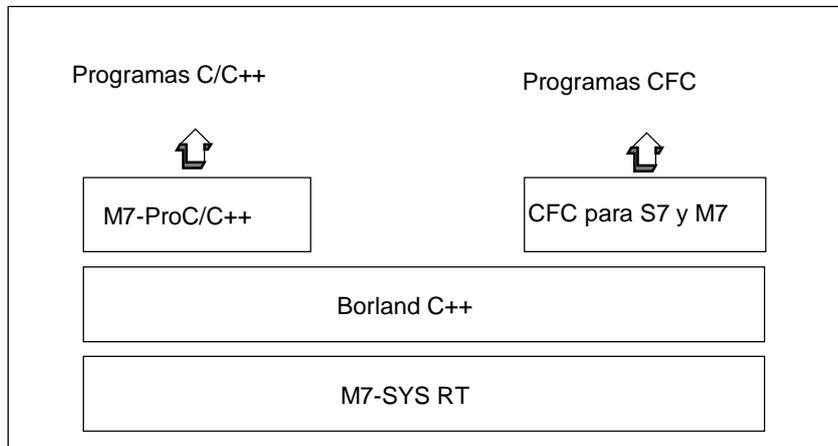
Para crear programas de usuario M7 se requiere, además de STEP 7, el software opcional M7;

Software	Indice del capítulo
M7-SYS RT	<ul style="list-style-type: none"> <li>• Sistema operativo M7 RMOS32</li> <li>• Librería de sistema M7-API</li> <li>• Soporte MPI</li> </ul>
CFC para S7 y M7	Software de programación para aplicaciones CFC (CFC = Continuous Function Chart)
M7-ProC/C++	<ul style="list-style-type: none"> <li>• Incorporación del entorno de desarrollo Borland en STEP 7</li> <li>• Editor y generador de importación de símbolos</li> <li>• Depurador con lenguaje de alto nivel Organon xdb386</li> </ul>
Borland C++	Entorno de desarrollo Borland C++

Junto con el software opcional M7, STEP 7 le ayuda a realizar las siguientes actividades:

- transferir datos al M7 vía MPI
- consultar informaciones sobre el sistema de automatización M7
- efectuar determinados ajustes en el sistema de automatización M7 y realizar un borrado total del M7

La figura siguiente muestra las dependencias del software opcional M7 para la programación M7.



## Resumen

Para crear ...	necesita el software opcional M7 ...
Programas C/C++	<ol style="list-style-type: none"> <li>1. M7-SYS RT</li> <li>2. M7-ProC/C++</li> <li>3. Borland C++</li> </ol>
Programas CFC	<ol style="list-style-type: none"> <li>1. M7-SYS RT</li> <li>2. CFC para S7 y M7</li> <li>3. Borland C++</li> </ol>

## Para obtener ayuda

Las herramientas específicas para crear aplicaciones M7 están integradas en parte en STEP 7 y en parte en el software opcional M7.

En la tabla siguiente se indica qué ayuda ofrecen los diversos paquetes de software:

El software ...	permite ...
STEP 7	<ul style="list-style-type: none"> <li>• instalar el sistema operativo M7,</li> <li>• gestionar el sistema de automatización M7,</li> <li>• transferir, iniciar y borrar programas de usuario M7,</li> <li>• llamar informaciones de estado y de diagnóstico, así como</li> <li>• efectuar el borrado total de la CPU</li> </ul>
M7-SYS RT	gracias al sistema operativo y al software de sistema M7: <ul style="list-style-type: none"> <li>• controlar la secuencia del programa,</li> <li>• gestionar la memoria y los recursos,</li> <li>• acceder al ordenador y al hardware SIMATIC,</li> <li>• gestionar alarmas,</li> <li>• realizar diagnósticos,</li> <li>• controlar el estado y</li> <li>• establecer enlaces de comunicación</li> </ul>
M7-ProC/C++	<ul style="list-style-type: none"> <li>• gracias a la creación de lógica integrada (integración del interface de desarrollo Borland en STEP 7)</li> <li>• gracias a la integración de los símbolos del proyecto en la lógica fuente e</li> <li>• gracias a la integración de funciones de depuración</li> </ul>
Borland C++	<ul style="list-style-type: none"> <li>• crear programas C y C++</li> </ul>
CFC para S7 y M7	<ul style="list-style-type: none"> <li>• crear, comprobar y depurar programas CFC e</li> <li>• iniciar y ejecutar programas CFC</li> </ul>

## 24.3 Sistemas operativos para M7-300/400

Los servicios que ofrece el sistema operativo tienen importancia decisiva para las aplicaciones creadas en los lenguajes de alto nivel C y C++. El sistema operativo se encarga de las siguientes tareas:

- Acceso al hardware
- Gestión de recursos
- Integración en el sistema
- Comunicación con otros componentes del sistema

Para solucionar tareas de automatización, en los microcomputadores industriales SIMATIC M7 se utiliza el sistema operativo de tiempo real M7 RMOS32 (RMOS = **R**ealtime-**M**ultitasking-**O**perating-**S**ystem). Para poder incorporar M7 RMOS32 en el sistema SIMATIC, se le ha agregado un interface de llamada M7-API (API = **A**pplication **P**rogramming **I**nterface).

El sistema operativo de tiempo real M7 RMOS32 se utiliza para aplicaciones de 32 bits destinadas a solucionar tareas críticas de tiempo real y de multitarea. Se encuentra disponible en las siguientes configuraciones para los módulos M7:

- M7 RMOS32
- M7 RMOS32 con MS-DOS

La configuración del sistema operativo a seleccionar para el sistema de automatización M7 depende de los módulos M7 utilizados:

Configuración del sistema operativo	Módulo / memoria principal	PROFIBUS-DP y TCP/IP sí/no	Instalación en memorias de masa
M7 RMOS32	FM 356-4 / 4MB	no	Memory Card $\geq$ 4 MB o disco duro
	FM 356-4 / 8MB	sí	
	CPU 388-4 / 8MB	sí	
	FM 456-4 / 16MB	sí	
	CPU 488-3 / 16MB	sí	
	CPU 486-3 / 16MB	sí	
M7 RMOS32 con MS-DOS	FM 356-4 / 8MB	no	Memory Card $\geq$ 4 MB o disco duro
	CPU 388-4 / 8MB	no	
	FM 456-4 / 16MB	sí	
	CPU 488-3 / 16MB	sí	
	CPU 486-3 / 16MB	sí	

## 25 Consejos y trucos

### 25.1 Intercambiar módulos en la tabla de configuración

Si está corrigiendo una configuración de equipos con HW-Config. y desea intercambiar un módulo, p.ej. por una nueva referencia, proceda de la siguiente forma:

1. Mediante "Arrastrar y soltar" (Drag&Drop) saque el módulo de la ventana "Catálogo de hardware" sobre el módulo ("antiguo") ya ubicado.
2. Deje "caer" el nuevo módulo, que en cuanto sea posible adoptará los parámetros del previamente insertado.

Este procedimiento es más rápido que intercambiar los módulos borrando el antiguo e insertando y parametrizando a continuación el nuevo.

Esta función puede activarse o desactivarse a voluntad en HW-Config. con el comando de menú **Herramientas > Preferencias** ("Permitir sustitución de módulos").

### 25.2 Proyectos con un gran número de equipos en red

Si configura equipos sucesivamente y después llama a NetPro con **Herramientas > Configurar red** para configurar enlaces, los equipos se ubicarán automáticamente en la representación de la red. Este procedimiento tiene la desventaja de que posteriormente deberá ordenar por criterios topológicos los equipos y rubredes.

Si su proyecto abarca un gran número de equipos en red y dese configurar enlaces entre dichos equipos, debe configurar desde el principio en la representación de la red la estructura de la instalación, para conservar una visión panorámica:

1. Cree el nuevo proyecto en el Administrador SIMATIC (comando de menú **Archivo > Nuevo**)
2. Arranque NetPro (comando de menú **Herramientas > Configurar red**)
3. En NetPro, cree un equipo tras otro:
  - Mediante "Arrastrar y soltar" (Drag&Drop) mueva el equipo de la ventana "Catálogo".
  - Haga doble clic en el equipo para iniciar HW-Config.
  - Mediante "Arrastrar y soltar" (Drag&Drop) ubique en HW-Config. los módulos aptos para comunicaciones (CPUs, CPs, FMs, módulos IF).
  - Si dese conectar dichos módulos en red: haga doble clic en las líneas correspondientes de la tabla de configuración para crear de nuevo subredes y para conectar en red los interfaces.
  - Guarde la configuración y cambie a NetPro.
  - Posicione en NetPro los equipos y subredes (mover objeto con el cursor del ratón, hasta que se alcance la posición deseada).
4. Configura los enlaces en NetPro y corrija (si fuese necesario) las conexiones a la red.

## 25.3 Reorganización

Cuando surgen problemas inexplicables al trabajar con STEP 7, suele ser útil reorganizar la gestión de datos del proyecto o de la librería en cuestión.

Para ello elija el comando de menú **Archivo > Reorganizar**. Al reorganizar los datos se eliminan los huecos que se fueron creando al borrar datos, es decir, se reduce el espacio de memoria requerido por los datos del proyecto o librería.

La función optimiza el almacenamiento de los datos del proyecto o de la librería, de la misma manera que un programa para desfragmentar un disco duro optimiza el almacenamiento de los datos en el mismo.

El proceso de reorganización puede prolongarse, todo depende de los datos que se tengan que desplazar en la memoria. Por esta razón, la función no se ejecuta automáticamente sino que ha de ser iniciada por el usuario cuando éste lo considere necesario.

### Requisitos

Para poder reorganizar proyectos o librerías es necesario que los objetos afectados no estén ocupados, p. ej. por estar abiertos en otras aplicaciones.

## 25.4 Test con la tabla de variables

Para observar y forzar variables en la tabla de variables hay una serie de consejos prácticos de edición:

- Los símbolos y operandos se pueden introducir tanto en la columna "Símbolo" como en la columna "Operando". La entrada se escribirá automáticamente en la columna correspondiente.
- Para poder ver el valor forzado es necesario ajustar el punto de disparo para "Observar" al "Inicio del ciclo" y el punto de disparo para "Forzar" al "Fin del ciclo".
- Si coloca el cursor en una línea marcada en rojo, en la información breve (tooltip) se puede leer la causa del error. Con la tecla F1 obtendrá indicaciones para eliminar el error.
- Sólo se pueden introducir aquellos símbolos que ya estén definidos en la tabla de símbolos.  
Un símbolo debe introducirse exactamente como está definido en la tabla de símbolos. Los nombres simbólicos que contienen caracteres especiales deben ir entre comillas (p.ej. "Motor.OFF", "Motor+OFF", "Motor-OFF").
- Desactivar las advertencias en la ficha "Online" (cuadro de diálogo "Preferencias")
- Cambiar de enlace sin haber deshecho otro enlace que estuviera establecido
- Ajustar el disparo para "Observar" mientras se observan variables.
- Forzar las variables que seleccione marcando las líneas correspondientes y ejecutando la función "Forzar". En este caso se forzarán solamente las variables seleccionadas que estén visibles.
- Con la función "Agrupar variables" puede aumentar el número de variables observables (ficha "Online" del cuadro de diálogo "Preferencias").

Si pulsa la tecla ESC cuando estén activadas "Observar", "Forzar" o "Desbloquear salidas", se saldrá de las funciones "Observar" y "Forzar" sin consultar.

- Introducir un área de operandos conexos:

Utilice el comando de menú **Insertar > Área**.

- Mostrar y ocultar columnas:

Para mostrar u ocultar columnas concretas, utilice los siguientes comandos de menú:

Símbolo: comando de menú **Ver > Símbolo**

Comentario del símbolo: comando de menú **Ver > Comentario del símbolo**

Formato de representación del valor de estado: comando de menú **Ver > Formato de visualización**

Valor de estado de la variable: comando de menú **Ver > Valor de estado**

Valor de forzado de la variable: comando de menú **Ver > Valor de forzado**

- Cambiar simultáneamente el formato de visualización de varias líneas de la tabla:

1. Seleccione el área de la tabla en la que desee cambiar el formato de visualización arrastrando el ratón (con la tecla izquierda pulsada) por el área que desee de la tabla.
2. Elija la representación con el comando de menú **Ver > Elegir formato de visualización**. Sólo cambia el formato de las líneas de la tabla seleccionada para las que está permitido el cambio de formato.

- Ejemplos de entrada con la tecla F1:

- Si coloca el cursor en la columna de operandos y pulsa F1, obtendrá ejemplos de introducción de operandos.
- Si coloca el cursor en la columna de valores de forzado y pulsa F1 obtendrá ejemplos de introducción de valores de forzado / de forzado permanente.

## 25.5 Memoria de trabajo virtual

La memoria de trabajo virtual también puede afectar al correcto funcionamiento de STEP 7.

Para trabajar con STEP 7 bajo Windows 95/98/NT/2000 es necesario adaptar la memoria de trabajo virtual. Proceda como sigue:

1. Abra el panel de control, p. ej. desde el menú **Inicio > Configuración > Panel de control**.
2. Haga doble clic en el símbolo "Sistema".
3. En Windows 95/98/NT, elija la ficha "Rendimiento" en el cuadro de diálogo que aparecerá.  
En Windows 2000, elija la ficha "Avanzadas" en el cuadro de diálogo que aparecerá y pulse el botón "Opciones de rendimiento".
4. Haga clic en el botón de comando "Memoria virtual" (en Windows 9x) o "Cambiar" (en Windows NT/2000).
5. Sólo para 9x: elija en el cuadro de diálogo "Memoria virtual" la opción "Para la memoria virtual rigen los ajustes definidos por el usuario".
6. Introduzca bajo "Mínimo" o "Tamaño inicial" como mínimo 40 MBytes y bajo "Máximo" o "Tamaño máximo" como mínimo 150 MBytes.
7. Sólo para 9x: asegúrese de que la opción "Desactivar memoria virtual" esté desactivada. Sólo NT: Haga clic en el botón de comando "Establecer".

---

### Nota

Como la memoria virtual reside en el disco duro (por defecto C: y dinámico) hay que prever suficiente espacio de memoria para el directorio TMP o TEMP (aprox. 20 - 30 MB):

- Si el proyecto S7 se encuentra en la misma partición en la que reside la memoria virtual, se debería prever el doble de memoria de la que ocupa el proyecto S7.
  - No obstante, si el proyecto se gestiona en otra partición, no es necesario prever el doble de memoria.
-

# A Anexos

## A.1 Estados operativos

### A.1.1 Estados operativos y cambios de estado

#### Estados operativos

El estado operativo describe el comportamiento de la CPU en cualquier momento. El conocimiento de los diferentes estados operativos de las CPUs sirve de ayuda para la programación del arranque, la prueba del autómatas y el diagnóstico de errores.

Las CPUs S7-300 y S7-400 pueden adoptar los siguientes estados operativos:

- STOP
- ARRANQUE
- RUN
- PARADA

En el estado operativo "STOP", la CPU comprueba la existencia de los módulos configurados o de los que utilizan direcciones predeterminadas conduciendo además la periferia a un estado básico predefinido. El programa de usuario no se ejecuta en el estado operativo "STOP".

En el estado operativo "ARRANQUE" se distingue entre "Rearranque completo" (en caliente), "Arranque en frío" y "Rearranque":

- En el re arranque completo (en caliente) se vuelve a ejecutar el programa desde el principio con un "ajuste básico" de los datos del sistema y de las áreas de operandos de usuario (se inicializan los temporizadores, contadores y marcas no remanentes).
- En el arranque en frío se lee la imagen de proceso de las entradas y el programa de usuario STEP 7 se ejecuta comenzando por la primera instrucción del OB 1 (rige también para el re arranque completo (en caliente)).
  - Se borran los bloques de datos creados mediante SFCs (funciones del sistema) en la memoria de trabajo, en tanto que los demás bloques de datos adoptan el valor estándar de la memoria de carga.
  - La imagen del proceso, así como todos los temporizadores, contadores y marcas se ponen a cero, independientemente de que se hayan parametrizado como remanentes o no.
- En el re arranque, la ejecución del programa se inicia en el punto interrumpido (no se reposicionan los temporizadores, contadores y marcas). El re arranque sólo es posible en las CPUs S7-400.

En el estado operativo "RUN", la CPU ejecuta el programa de usuario, actualiza las entradas y salidas y procesa las alarmas y los mensajes de error.

En el estado "PARADA" se detiene la ejecución del programa de usuario y se puede comprobar dicho programa paso a paso. Este estado sólo puede activarse en la prueba con la PG.

La CPU puede comunicarse en todos estos estados operativos a través del puerto MPI.

### Otros estados operativos

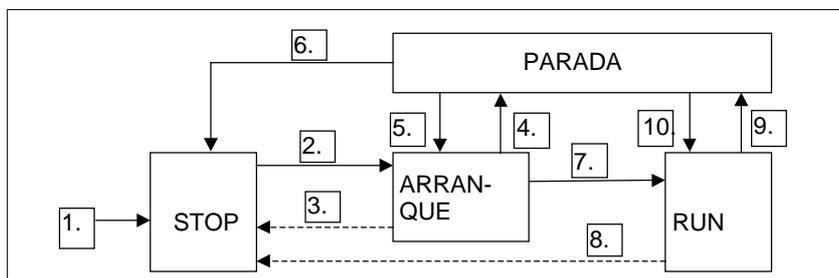
Si la CPU no está operacional, entonces se encuentra en uno de los dos siguientes estados operativos:

- sin tensión, es decir, la tensión de red está desconectada.
- defectuosa, es decir, existe un error no eliminable.

Comprobar si la CPU está realmente defectuosa: poner la CPU en STOP, desconectar y volver a conectar con el interruptor de red. Si la CPU arranca, leer el búfer de diagnóstico para analizar el error. Si la CPU no arranca, es necesario sustituirla.

### Cambios de estados operativos

La figura siguiente muestra los estados operativos y los cambios de estado operativo de las CPUs S7-300 y S7-400:



La tabla siguiente muestra bajo qué condiciones cambian los estados operativos.

Cambio	Descripción
1.	Tras conectar la tensión de alimentación, la CPU se encuentra en el estado "STOP".
2.	La CPU pasa al estado "ARRANQUE" <ul style="list-style-type: none"> <li>• tras llevar la CPU a RUN o RUN-P con el selector de modo o la PG o</li> <li>• tras activarse automáticamente un tipo de arranque por CONEXION.</li> <li>• Al ejecutarse las funciones de comunicación "RESUME" o "START".</li> </ul> El selector debe estar en ambos casos en RUN o RUN-P.
3.	La CPU pasa nuevamente a STOP cuando <ul style="list-style-type: none"> <li>• se detecta un error durante el arranque</li> <li>• la CPU se lleva a STOP con el selector de modo o la PG</li> <li>• se procesa una orden de parada en el OB de arranque o</li> <li>• se ejecuta la función de comunicación "STOP".</li> </ul>
4.	La CPU pasa al estado "PARADA" cuando se alcanza un punto de parada en el programa de arranque.
5.	La CPU pasa a "ARRANQUE" cuando un punto de parada estaba ajustado en el programa de arranque y se ejecuta la orden "ABANDONAR PARADA" (función de prueba).
6.	La CPU pasa nuevamente a STOP cuando <ul style="list-style-type: none"> <li>• la CPU se lleva a STOP con el selector de modo o la PG o</li> <li>• se ejecuta la orden de comunicación "STOP".</li> </ul>
7.	Si el arranque es correcto, la CPU cambia a RUN.
8.	La CPU pasa nuevamente a STOP cuando <ul style="list-style-type: none"> <li>• se detecta un error en el estado RUN y el OB correspondiente no está cargado,</li> <li>• la CPU se lleva a STOP con el selector de modo o la PG,</li> <li>• se procesa una orden de parada en el programa de usuario o</li> <li>• se ejecuta la función de comunicación "STOP".</li> </ul>
9.	La CPU pasa al estado "PARADA" cuando se alcanza un punto de parada en el programa de usuario.
10.	La CPU pasa a "RUN" cuando estaba ajustado un punto de parada y se ejecuta la orden "ABANDONAR PARADA".

### Prioridad de los estados operativos

Si se solicitan simultáneamente varios cambios de estado operativo, se cambia al estado que tenga la prioridad más alta. Por ejemplo, si el selector de modos de operación de la CPU se encuentra en RUN y se intenta cambiar la CPU a STOP desde la PG, ésta pasará a STOP, puesto que éste es el estado operativo de máxima prioridad.

Prioridad	Estado operativo
más alta	STOP
	PARADA
	ARRANQUE
más baja	RUN

### A.1.2 Estado operativo STOP

El programa de usuario no se ejecuta en el estado operativo "STOP". Todas las salidas se ajustan a valores sustitutivos, llevando así el proceso de forma controlada a un estado operativo seguro. La CPU comprueba si

- existen problemas de hardware (p. ej., módulos no disponibles)
- debe regir el ajuste por defecto para la CPU o existen registros de parámetros
- son correctas las condiciones requeridas para el comportamiento en arranque programado
- existen problemas de software del sistema.

En el estado STOP se pueden recibir también datos globales, se puede efectuar comunicación unilateral pasiva a través de SFBs para comunicaciones para enlaces configurados y a través de SFCs para comunicaciones para enlaces no configurados.

### Borrado total

En STOP se puede borrar totalmente la CPU. El borrado total puede ejecutarse manualmente a través del selector de modo (MRES) o desde la PG (p.ej., antes de cargar un programa de usuario).

Con el borrado total se conduce la CPU al "estado original", es decir

- se borra el programa de usuario completo en la memoria de trabajo y en la memoria de carga RAM, así como todas las áreas de operandos,
- se reposicionan los parámetros del sistema y los parámetros de la CPU y de módulos al ajuste por defecto, se conservan los parámetros MPI ajustados antes del borrado total.
- si hay una Memory Card (Flash-EEPROM) insertada, la CPU copia el programa de usuario de dicha Memory Card en la memoria de trabajo (incluidos los parámetros de la CPU y de módulos, si los correspondientes datos de configuración se encuentran también en la Memory Card).

El búfer de diagnóstico, los parámetros MPI, la hora y el contador de horas de funcionamiento no se reposicionan.

### A.1.3 Estado operativo ARRANQUE

Antes de que la CPU comience, tras la conexión, a ejecutar el programa de usuario, se ejecuta un programa de arranque. En el programa de arranque se pueden definir determinados preajustes para el programa cíclico a través de la programación de OB de arranque.

Se dispone de tres tipos de arranque: Rearranque completo (en caliente), arranque en frío y re arranque. Por principio, un re arranque completo sólo se puede efectuar en las CPUs S7-400, debiéndose haber definido mediante STEP 7 en el juego de parámetros de la CPU.

En el estado operativo "ARRANQUE":

- se ejecuta el programa en el OB de arranque (OB 100 para re arranque completo (en caliente), OB 101 para re arranque, OB 102 para arranque en frío)
- no es posible la ejecución de programa controlada por tiempo o por alarmas
- se actualizan los temporizadores
- corre el contador de horas de funcionamiento
- las salidas digitales están bloqueadas en los módulos de señales, pero se pueden posicionar por acceso directo.

#### Rearranque completo (en caliente)

El re arranque completo (en caliente) es siempre posible, a no ser que el sistema haya solicitado el borrado total. En los casos siguientes, el re arranque completo (en caliente) es posible sólo tras:

- borrado total
- carga del programa de usuario en el estado STOP de la CPU
- desbordamiento de la PILA DE INTERRUPCION/PILA DE BLOQUES (USTACK/BSTACK)
- interrupción de re arranque completo (en caliente) (por DESCONEXION o a través del selector de modo)
- rebase del límite de tiempo de interrupción parametrizado para el re arranque.

#### Rearranque completo (en caliente) manual

El re arranque completo (en caliente) manual se puede activar:

- con el selector de modo

(el selector CRST/WRST – si existe - debe estar en CRST)

- con el correspondiente comando de menú desde la PG o a través de las funciones de comunicación

(cuando el selector de modo esté en RUN o RUN-P)

### **Rearranque completo (en caliente) automático**

El re arranque completo (en caliente) automático se puede activar en CONEXION cuando:

- la CPU no estaba en STOP cuando se desconectó la alimentación
- el selector de modo operativo está en RUN o RUN-P
- no hay re arranque automático parametrizado tras CONEXION
- la CPU ha sido interrumpida durante el re arranque completo (en caliente) por corte de alimentación (independientemente de la parametrización del tipo de arranque).

En el re arranque completo (en caliente) automático no tiene efecto el selector CRST/WRST.

### **Rearranque completo (en caliente) automático sin respaldo por pila**

Si la CPU opera sin pila tampón (si se requiere funcionamiento libre de mantenimiento), tras la conexión o al regresar la tensión después de OFF, se efectúa automáticamente el borrado total de la CPU y se realiza luego un re arranque completo (en caliente). A tal efecto, el programa de usuario debe estar memorizado en la Flash-EEPROM (Memory Card).

### **Rearranque**

Después de un corte de tensión en RUN y al regresar nuevamente la tensión, las CPUs S7-400 corren una rutina de inicialización y ejecutan luego automáticamente un re arranque. En caso de re arranque, el programa de usuario se continúa en el punto donde fue interrumpida la ejecución. La parte del programa de usuario que no llegó a ejecutarse antes del corte de tensión se denomina ciclo residual. El ciclo residual puede contener también partes del programa controladas por tiempo y por alarmas.

Básicamente, el re arranque sólo es admisible si el programa de usuario no ha sido modificado en el estado STOP (p. ej. recargando un bloque modificado) y si no se requiere por otras razones un re arranque completo (en caliente). Se distingue entre re arranque manual y automático.

### **Rearranque manual**

El re arranque manual sólo es posible si se ha parametrizado correspondientemente el registro de parámetros de la CPU y tras las siguientes causas de STOP:

- el selector de modo se ha cambiado de RUN a STOP
- STOPs programados por el usuario, STOPs tras llamar OBs no cargados
- el estado STOP ha sido causado por la PG o por una función de comunicación.

El re arranque manual se puede activar:

- con el selector de modo

El selector CRST/WRST debe estar en WRST.

- por comando de menú desde la PG o por funciones de comunicación (si el selector de modo está en RUN o RUN-P)
- se ha parametrizado el registro de parámetros de la CPU para re arranque manual.

## Rearranque automático

El re arranque automático se puede activar en CONEXION cuando:

- la CPU no estaba en estado STOP o PARADA en OFF
- el selector de modo operativo está en RUN o RUN-P
- en el registro de parámetros de la CPU se ha parametrizado re arranque automático tras CONEXION.

El selector CRST/WRST no tiene efecto en el re arranque automático.

## Areas de datos remanentes tras fallar la red

Las CPUs S7-300 y S7-400 reaccionan de forma distinta al regresar la tensión tras un fallo de red.

En las CPUs S7-300 (con excepción de la CPU 318) sólo se dispone del "Rearranque completo". No obstante, con STEP 7 se puede ajustar la remanencia de marcas, temporizadores, contadores y áreas en los bloques de datos para impedir la pérdida de datos al producirse un corte de tensión. Al regresar la tensión, se efectúa un "re arranque completo automático con memoria".

Al regresar la tensión, las CPUs S7-400 reaccionan con un re arranque completo (en caliente) (tras CONEXION con o sin respaldo por pila) o con un re arranque (sólo tras CONEXION con respaldo por pila).

La tabla siguiente muestra el comportamiento de remanencia de las CPUs S7-300 y S7-400 en caso de un re arranque completo (en caliente), un arranque en frío y un re arranque:

X	significa que	se conservan los datos
VC	significa que	se conserva el bloque lógico en la EPROM, un bloque lógico sobrecargado se pierde
VX	significa que	el bloque de datos sólo se conserva si existe en la EPROM, los datos remanentes se adoptan de la RAM NV (los bloques de datos cargados o creados en la RAM se pierden)
0	significa que	los datos se inicializan o se borran (contenido de DBs)
V	significa que	datos se ajustan al valor de inicialización, a partir de la memoria EPROM
---	significa que	no es posible, porque no existe una RAM NV

### Comportamiento de remanencia en la memoria de trabajo (en la memoria de carga EPROM y RAM):

EPROM (Memory Card o integrada)									
CPU con respaldo					CPU sin respaldo por pila				
Datos	Bloques en memoria de carga	DB en memoria de trabajo	Marcas, temp., contadores	Marcas, temp., contadores	Bloques en memoria de carga	DB en memoria de trabajo	DB en memoria de trabajo	Marcas, temp., contadores	Marcas, temp., contadores
			param. como remanentes	param. como no remanentes		param. como remanentes	param. como no remanentes	param. como remanentes	param. como no remanentes
Rearranque completo en S7-300	X	X	X	0	VC	VX	V	X	0
Rearranque completo en S7-400	X	X	X	0	VC	---	V	0	0
Arranque en frío en S7-300	X	X	0	0	VC	V	V	0	0
Arranque en frío en S7-400	X	X	0	0	VC	---	V	0	0
Rearranque en S7-400	X	X	X	X	Sólo se permite reearranque completo				
temp. = temporizadores param. = parametrizados remanen. = remanentes									

### Actividades en el arranque

La tabla siguiente muestra qué actividades efectúa la CPU durante el arranque:

Actividades durante la secuencia de ejecución	en el reearranque completo (en caliente)	en el arranque en frío	en el reearranque
Borrar pila de interrupción/pila de bloques	X	X	0
Borrar marcas, temporizadores, contadores no remanentes	X	0	0
Borrar todas las marcas, temporizadores y contadores	0	X	0
Borrar imagen de proceso de las salidas	X	X	parametrizable
Reponer salidas de los módulos de señales	X	X	parametrizable
Rechazar alarmas de proceso	X	X	0

Actividades durante la secuencia de ejecución	en el re arran- que completo (en caliente)	en el arranque en frío	en el re arranque
Rechazar alarmas de retardo	X	X	0
Rechazar alarmas de diagnóstico	X	X	X
Actualizar lista de estado del sistema (SZL)	X	X	X
Evaluar parámetros de módulos y transferirlos a los módulos o entregar valores por defecto	X	X	X
Ejecutar el OB de arranque correspondiente	X	X	X
Procesar ciclo residual (parte del programa de usuario cuya ejecución no pudo continuar en virtud de una desconexión de alimentación)	0	0	X
Actualizar imagen de proceso de las entradas	X	X	X
Habilitar salidas digitales (desactivar señal OD) tras cambio de estado operativo a RUN	X	X	X
X significa que se realiza			
0 significa que no se realiza			

## Interrumpir un arranque

Si ocurren errores durante el arranque, éste se interrumpe y la CPU pasa o permanece en STOP.

Un re arranque completo (en caliente) interrumpido debe ser repetido. Tras un re arranque interrumpido, es posible tanto un re arranque completo (en caliente) como un re arranque normal.

Un arranque (re arranque completo o re arranque) no se ejecuta, o se interrumpe, si

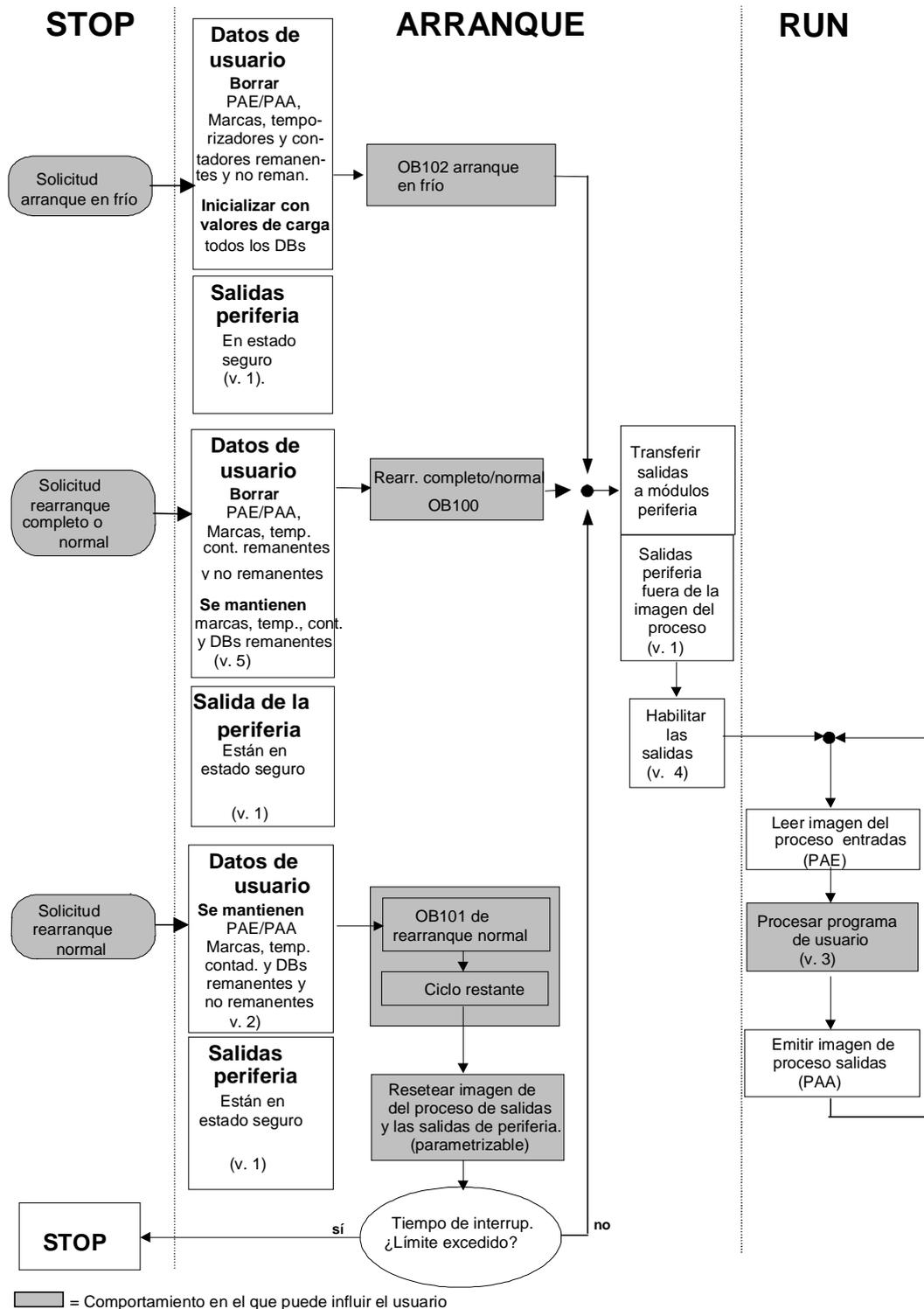
- el selector de modo de la CPU está en STOP
- se solicita borrado total
- hay insertada una Memory Card cuya identificación de aplicación no es admisible para STEP 7 (p. ej. STEP 5)
- en modo monoprocesador hay más de una CPU insertada
- el programa de usuario contiene un OB desconocido por la CPU o que ha sido inhibido
- la CPU detecta tras la conexión de la tensión de red que no todos los módulos listados en la tabla de configuración de STEP 7 están realmente insertados (diferencia no admisible entre la parametrización teórica y la real)
- ocurren errores al evaluar los parámetros de los módulos.

Un arranque no se ejecuta, o se interrumpe, si

- la CPU ha sido borrada totalmente (tras el borrado total sólo es posible el re arranque completo (en caliente))
- se ha rebasado el límite de tiempo de interrupción (el tiempo de interrupción es aquél que transcurre desde que se sale de RUN hasta que se ejecuta el OB de arranque, incluido el ciclo residual)
- se ha modificado la configuración de módulos (p. ej. cambio de un módulo)
- por parametrización sólo es admisible re arranque completo (en caliente)
- se han cargado, borrado o modificado módulos en STOP.

### Transcurso

La figura siguiente muestra las actividades de la CPU en los estados operativos ARRANQUE y RUN.



Leyenda de la figura "Actividades de la CPU en los estados operativos ARRANQUE y RUN":

1. Los módulos de periferia conmutan a estado seguro (valor por defecto = 0) todas las salidas de periferia. Este hecho es independiente de que en el programa de usuario se utilicen dentro o fuera del área de imagen del proceso.  
  
Si se utilizan módulos de señales aptos para valor de sustitución, el comportamiento de las salidas puede parametrizarse, p. ej. mantener el último valor.
2. Es necesario para ejecutar el ciclo residual.
3. Al llamar por primera vez una imagen actual del proceso de las entradas, se dispone también de los Obs de alarma.
4. El estado de las salidas de periferia centralizadas y descentralizadas en el primer ciclo del programa de usuario puede definirse con las siguientes medidas:
  - Utilizando módulos de salida parametrizables, para poder emitir valores de sustitución o para mantener el valor.
  - En el re arranque: activar el parámetro de arranque de CPU "Inicializar salidas al efectuar un arranque normal" para dar salida a un "0" (equivale al ajuste predeterminado).
  - Inicializar salidas en el OB de arranque (OB 100, OB 101, OB 102).
5. En sistemas S7-300 sin respaldo de memoria sólo se mantienen las áreas de DB configuradas como remanentes.

#### A.1.4 Estado operativo RUN

En el estado operativo "RUN" tiene lugar la ejecución del programa cíclica, la controlada por tiempo y la controlada por alarmas:

- se lee la imagen de proceso de las entradas
- se ejecuta el programa de usuario
- se emite la imagen de proceso de las salidas.

El intercambio activo de datos entre las CPUs a través de la comunicación de datos globales (tabla de datos globales) y de SFBs para comunicaciones para enlaces configurados y a través de SFCs para comunicaciones para enlaces no configurados sólo es posible en el estado RUN.

La tabla siguiente ejemplifica cuándo es posible el intercambio de datos en diferentes estados operativos:

Tipo de comunicación	Estado operativo de la CPU 1	Sentido del intercambio de datos	Estado operativo de la CPU 2
Comunicación por datos globales	RUN	↔	RUN
	RUN	→	STOP/PARADA
	STOP	←	RUN
	STOP	X	STOP
	PARADA	X	STOP/PARADA
Comunicación unilateral	RUN	→	RUN
a través de SFBs para comunicaciones	RUN	→	STOP/PARADA
Comunicación bilateral a través de SFBs para comunicaciones	RUN	↔	RUN
Comunicación unilateral	RUN	→	RUN
a través de SFCs para comunicaciones	RUN	→	STOP/PARADA
Comunicación bilateral a través de SFCs para comunicaciones	RUN	↔	RUN
↔ significa que el intercambio de datos es posible en ambos sentidos → significa que el intercambio de datos es posible en un solo sentido X significa que el intercambio de datos no es posible			

### A.1.5 Estado operativo PARADA

El estado operativo PARADA (HALT) representa un estado especial. Se utiliza sólo para fines de prueba durante el arranque o RUN. En el estado PARADA:

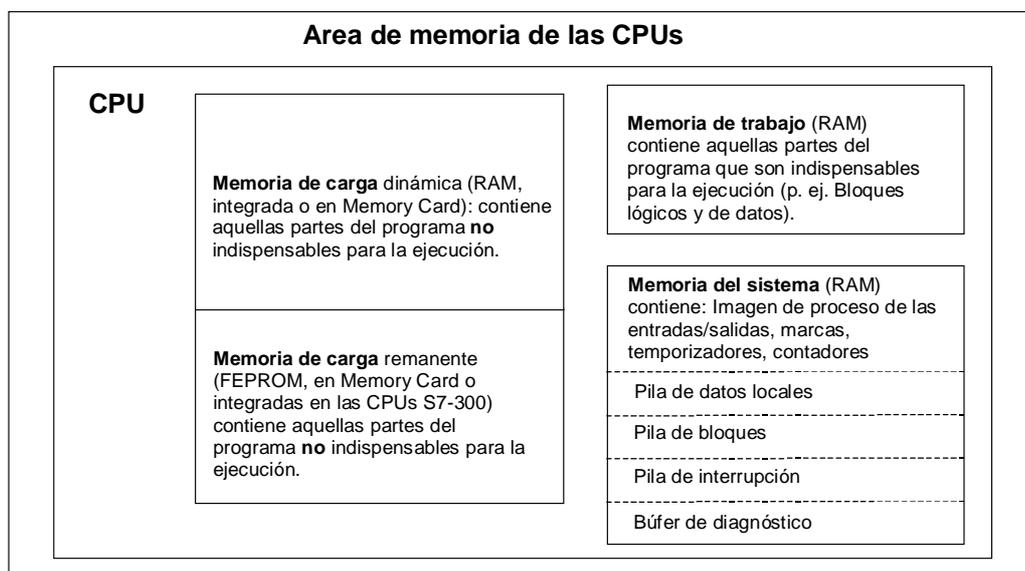
- se congelan todos los temporizadores: no se procesan los temporizadores ni los contadores de horas de funcionamiento, se detienen los tiempos de vigilancia y los ciclos básicos de los niveles con ejecución controlada por tiempo.
- corre el reloj de tiempo real
- no se habilitan las salidas, pero pueden habilitarse para fines de prueba
- se pueden forzar entradas y salidas
- en caso de corte y retorno de la tensión en el estado PARADA, las CPUs respaldadas por pila pasan a STOP y no provocan ningún rearmado o rearmado completo automático. Las CPUs no respaldadas provocan, al retornar la tensión, un rearmado automático no respaldado.
- se pueden recibir también datos globales, se puede efectuar comunicación unilateral pasiva a través de SFBs para comunicaciones para enlaces configurados y a través de SFCs para comunicaciones para enlaces no configurados (v. también tabla en Estado operativo RUN).

## A.2 Areas de memoria de las CPUs S7

### A.2.1 Subdivisión de la memoria en áreas

La memoria de las CPUs S7 se subdivide en tres áreas (v. siguiente figura):

- La memoria de carga permite almacenar el programa de usuario sin asignación simbólica de operandos o comentarios (éstos permanecen en la memoria de la PG). La memoria de carga puede ser RAM o EPROM.
- Los bloques caracterizados como no relevantes para la ejecución se memorizan exclusivamente en la memoria de carga.
- La memoria de trabajo (RAM integrada) contiene la partes del programa S7 relevantes para la ejecución del programa. La ejecución del programa tiene lugar exclusivamente en el área correspondiente a las memorias de trabajo y del sistema.
- La memoria del sistema (RAM) contiene los elementos de memoria que cada CPU pone a disposición del programa de usuario, tales como: la imagen de proceso de las entradas y salidas, marcas, temporizadores y contadores. Contiene además las pilas de bloques y de interrupción.
- La memoria del sistema de la CPU ofrece además una memoria temporal (pila de datos locales) asignada al programa para los datos locales del bloque llamado. Estos datos sólo tienen vigencia mientras esté activo el bloque correspondiente.



## A.2.2 Memorias de carga y de trabajo

Si el programa de usuario se carga en la CPU desde la unidad de programación, se cargan solamente los bloques lógicos y de datos en las memorias de carga y de trabajo de la CPU.

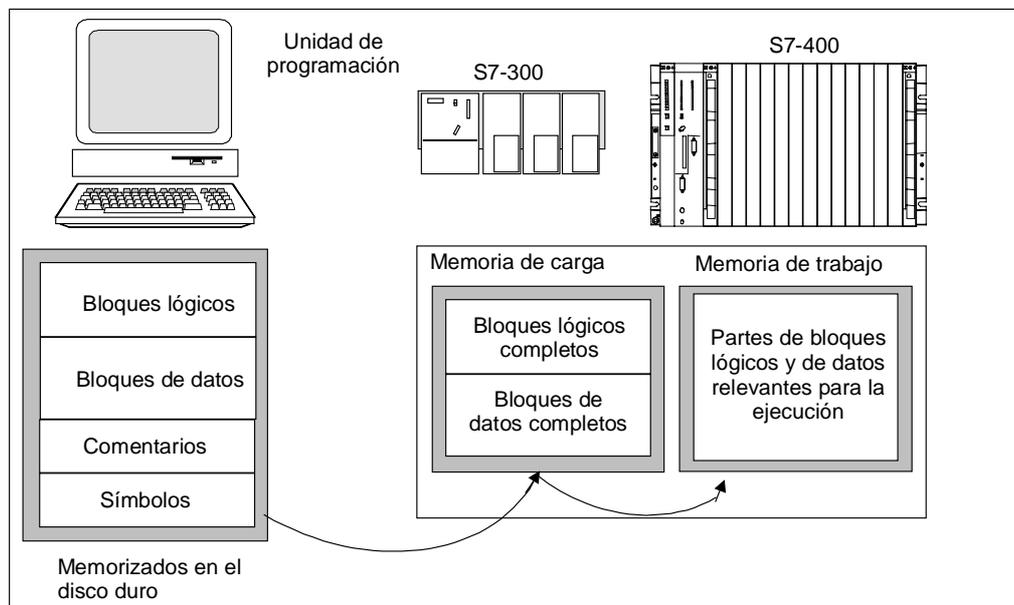
La asignación simbólica de operandos (tabla de símbolos) y los comentarios de los bloques permanecen en la memoria de la PG.

### Estructura del programa de usuario

Para garantizar una rápida ejecución del programa de usuario y no sobrecargar innecesariamente la memoria de trabajo ampliable, en dicha memoria se cargan únicamente aquellas partes de los bloques que son relevantes para la ejecución del programa.

Las partes de los bloques que no son necesarias para ejecutar el programa (p. ej., encabezamientos de bloques) permanecen en la memoria de carga.

La figura siguiente muestra cómo se carga el programa en la memoria de la CPU.



#### Nota

La CPU guarda completamente en la memoria de trabajo los bloques de datos que se hayan creado mediante funciones del sistema (p. ej. SFC 22 CREAT\_DB) en el programa de usuario.

Algunas CPUs disponen en la memoria de trabajo de áreas gestionadas por separado para el código y los datos. En dichas CPUs, el tamaño y la ocupación de las áreas se visualizan en la ficha "Memoria" de la información del módulo.

## Caracterizar los bloques de datos como "no relevantes para la ejecución"

Los bloques de datos programados en un archivo fuente como parte de un programa AWL pueden caracterizarse como "no relevantes para la ejecución" (clave UNLINKED). Es decir, al efectuar la carga en la CPU estos DBs se despositan solamente en la memoria de carga. Su contenido se puede copiar, en caso necesario, en la memoria de trabajo a través de la SFC 20 BLKMOV.

Así se ahorra espacio en la memoria de trabajo. La memoria de carga ampliable sirve de memoria intermedia (p. ej. para recetas: en la memoria de trabajo se carga solamente la próxima receta a ejecutar).

## Estructura de la memoria de carga

La memoria de carga se puede ampliar utilizando Memory Cards. En los manuales "Sistema de automatización S7-300, Configuración, instalación y datos de las CPU" o "Sistemas de automatización S7-400, M7-400, Datos de los módulos" se indica el tamaño máximo de la memoria de carga.

En el caso de las CPUs S7-300 la memoria de carga puede integrar una parte de memoria RAM y una parte de memoria EPROM. Mediante la parametrización con STEP 7 se pueden declarar como remanentes determinadas áreas en los bloques de datos (consulte también Areas remanentes de la memoria en CPUs S7-300 ).

En el caso de las CPUs S7-400 para poder ampliar la memoria de usuario es imprescindible utilizar una Memory Card (RAM o EPROM). La memoria de carga integrada es una memoria RAM, utilizada esencialmente para recargar y corregir bloques. En el caso de las CPUs S7-400 nuevas se puede enchufar también una memoria de trabajo adicional.

## Comportamiento de la memoria de carga en las áreas RAM y EPROM

El comportamiento de la memoria de carga durante las operaciones de carga, recarga o borrado total puede ser diferente dependiendo de la forma de ampliación elegida: Memory Card RAM o EPROM.

La tabla siguiente muestra las posibilidades de carga:

Tipo de memoria	Posibilidades de carga	Tipo de carga
RAM	Cargar y borrar bloques individuales	Enlace PG-CPU
	Cargar y borrar un programa S7 completo	Enlace PG-CPU
	Recargar bloques individuales	Enlace PG-CPU
EPROM integrada (sólo en el S7-300) o enchufable	Cargar programas S7 completos	Enlace PG-CPU
EPROM enchufable	Cargar programas S7 completos	Cargar la EPROM en la PG e insertar la Memory Card en la CPU Cargar la EPROM en la CPU

Los programas memorizados en RAM se pierden si se borra totalmente la CPU (MRES) o si se extrae la CPU o la Memory Card RAM.

Los programas memorizados en Memory Cards EPROM no se pierden al borrar totalmente y permanecen incluso sin respaldo por pila (transporte, copias de seguridad).

## A.2.3 Memoria de sistema

### A.2.3.1 Uso de las áreas de memoria del sistema

La memoria de sistema de las CPUs S7 está subdividida en áreas de operandos (v. siguiente tabla). El uso de las operaciones correspondientes permite direccionar los datos en el programa directamente en las diferentes áreas de operandos.

Area de operandos	Acceso a través de unidades del siguiente tamaño:	Notación S7	Descripción
Imagen del proceso de las entradas	Entrada (bit)	E	Al comienzo de cada ciclo, la CPU lee las entradas de los módulos de entradas y memoriza los valores en la imagen de proceso de las entradas.
	Byte de entrada	EB	
	Palabra de entrada	EW	
	Palabra doble de entrada	ED	
Imagen de proceso de las salidas	Salida (bit)	A	Durante el ciclo, el programa calcula los valores para las salidas y los deposita en la imagen de proceso de las salidas. Al final del ciclo, la CPU escribe los valores de salida calculados en los módulos de salidas.
	Byte de salida	AB	
	Palabra de salida	AW	
	Palabra doble de salida	AD	
Marcas	Marca (bit)	M	Esta área ofrece capacidad de memoria para los resultados intermedios calculados en el programa.
	Byte de marcas	MB	
	Palabra de marcas	MW	
	Palabra doble de marcas	MD	
Temporizadores	Temporizador (T)	T	Esta área contiene los temporizadores disponibles.
Contadores	Contador (Z)	Z	Esta área contiene los contadores disponibles.
Bloque de datos	Bloque de datos, abierto con "AUF DB":	DB	Los bloques de datos memorizan informaciones para el programa. Pueden estar definidos de tal manera que todos los bloques de datos puedan acceder a ellos (DBs globales) o pueden estar asignados a un determinado FB o SFB (DB de instancia).

Area de operandos	Acceso a través de unidades del siguiente tamaño:	Notación S7	Descripción
	Bit de datos	DBX	
	Byte de datos	DBB	
	Palabra de datos	DBD	
	Palabra doble de datos	DBW	
	Bloque de datos, abierto con "AUF DI":	DI	
	Bit de datos	DIX	
	Byte de datos	DIB	
	Palabra de datos	DIW	
	Palabra doble de datos	DID	
Datos locales	Bit de datos locales	L	Esta área de memoria contiene los datos temporales de un bloque durante la ejecución de dicho bloque. La pila L ofrece también memoria para la transferencia de parámetros de bloques y para memorizar los resultados intermedios de segmentos KOP.
	Byte de datos locales	LB	
	Palabra de datos locales	LW	
	Palabra doble de datos locales	LD	
Area de periferia: entradas	Byte de entrada de periferia	PEB	Las áreas de periferia de las entradas y salidas permiten el acceso directo a módulos de entrada y salida centralizados y descentralizados.
	Palabra de entrada de periferia	PEW	
	Palabra doble de entrada de periferia	PED	
Area de periferia: salidas	Byte de salida de periferia	PAB	
	Palabra de salida de periferia	PAW	
	Palabra doble de salida de periferia	PAD	

Las áreas de direcciones disponibles para las diferentes CPUs se indican en las descripciones de CPUs o en las listas de operaciones :

- Manual "Sistema de automatización S7-300, Configuración, instalación y datos de las CPU".
- Manual de referencia "Sistemas de automatización S7-400, M7-400, Datos de los módulos"
- Lista de operaciones "Sistema de automatización S7-300".
- Lista de operaciones "Sistema de automatización S7-400".

### A.2.3.2 Imagen del proceso de las entradas y salidas (E/S)

Si en el programa de usuario se accede a las áreas de operandos: entradas (E) y salidas (A), no se consultan los estados de señal en los módulos de señales digitales, sino los presentes en un área de la memoria del sistema de la CPU y de la periferia descentralizada. Esta área de memoria se designa como imagen del proceso.

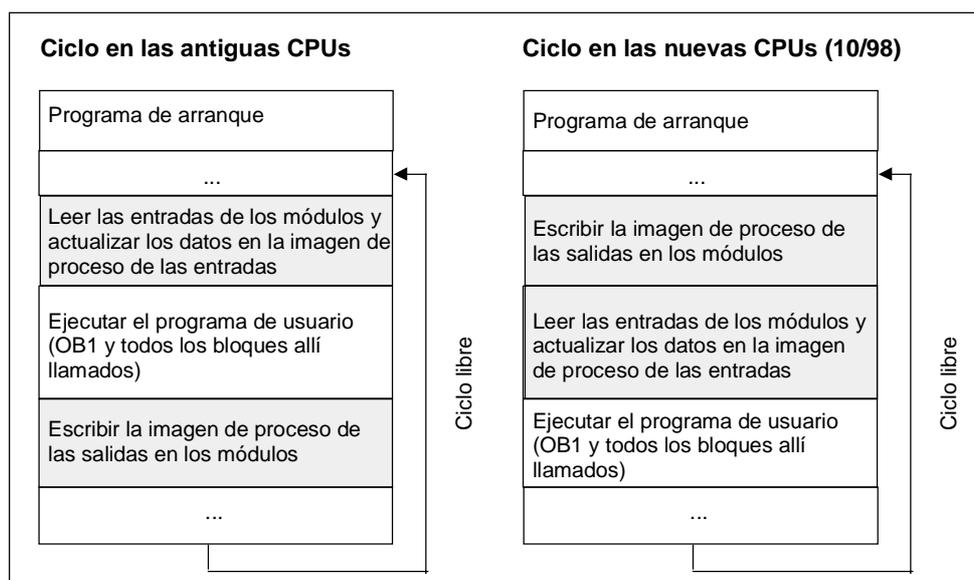
La imagen de proceso está subdividida en dos partes: imagen de proceso de las entradas e imagen de proceso de las salidas.

#### Requisito de acceso a la imagen del proceso

La CPU sólo puede acceder a la imagen de proceso de aquellos módulos que han sido configurados con STEP 7.

#### Actualizar la imagen de proceso

La imagen de proceso es actualizada cíclicamente por el sistema operativo. La figura siguiente muestra los pasos de procesamiento dentro de un ciclo con las diferencias entre las CPUs antiguas y las nuevas (suministrables desde octubre de 1998).



#### Ventajas de la imagen del proceso

En comparación con el acceso directo a los módulos de entrada/salida, el acceso a la imagen de proceso ofrece la ventaja de que la CPU dispone de una imagen coherente de las señales del proceso durante la ejecución cíclica del programa. Si durante la ejecución del programa varía un estado de señal en un módulo de entrada, dicho estado de señal se conserva en la imagen de proceso hasta que ésta sea actualizada en el próximo ciclo. Además, el acceso a la imagen de proceso requiere mucho menos tiempo que el acceso directo a los módulos de señales, ya que la imagen de proceso se encuentra en la memoria interna de la CPU.

## Imágenes parciales del proceso

Además de la imagen (global) del proceso (PAE y PAA), en S7-400 puede parametrizar hasta 15 imágenes parciales del proceso para una CPU (específica de CPU, N° 1 hasta máx. N° 15, consulte "Sistema de automatización S7-300, Configuración, instalación y datos de las CPU" y "Sistemas de automatización S7-400, M7-400, Datos de los módulos"). Esto permite actualizar, en caso necesario, áreas parciales de la imagen de proceso, independientemente de la actualización cíclica de dicha imagen.

Cada dirección de entrada/salida que ha asignado con STEP 7 a una imagen parcial del proceso ya no pertenece a la imagen del proceso de las entradas/salidas en el OB 1.

La imagen parcial del proceso se define con STEP 7 al asignar direcciones (qué direcciones de entrada/salida de los módulos se llevan a qué imagen parcial del proceso). La actualización de la imagen parcial del proceso se realiza a través de las SFCs o por parte del sistema mediante acoplamiento a un OB.

---

### Nota

En las CPUs S7-300, las entradas y salidas de imágenes de proceso no asignadas se pueden utilizar como áreas de marcas adicionales. Los programas que utilizan dicha posibilidad sólo pueden procesarse en CPUs S7-400 más antiguas (es decir, anteriores a 4/99) bajo la siguiente condición:

En dichas CPUs S7-400

- las imágenes del proceso utilizadas como marcas deben estar fuera del "Tamaño de la imagen del proceso" parametrizado, o
  - deben estar en una imagen parcial del proceso que no sea actualizada ni por el sistema ni mediante la SFC 26/SFC 27.
- 

## Actualizar con SFCs las imágenes parciales del proceso

Con ayuda de SFCs puede actualizar la imagen del proceso completa o imágenes parciales del proceso desde el programa de usuario.

Requisito: la correspondiente imagen parcial del proceso no es actualizada por el sistema

- SFC 26 UPDAT\_PI: Actualizar la imagen del proceso de las entradas
- SFC 27 UPDAT\_PO: Actualizar la imagen del proceso de las salidas.

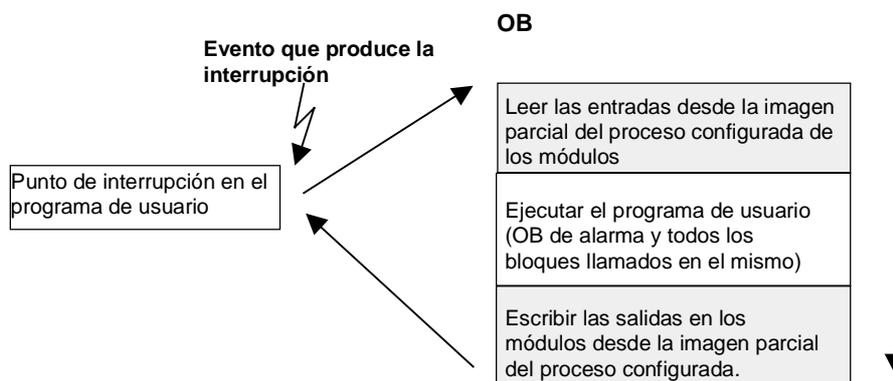
## Actualizar las imágenes parciales del proceso mediante el sistema

También puede hacer que el sistema operativo actualice automáticamente las imágenes parciales del proceso (de forma similar a la imagen (global) del proceso) llamando a un OB que se actualice antes o después de la ejecución del OB 1. Esta función sólo puede parametrizarse para determinadas CPUs.

Durante el funcionamiento, la imagen parcial del proceso asignada se actualizará automáticamente:

- antes de la ejecución del OB, la imagen parcial de las entradas del proceso
- después de la ejecución del OB, la imagen parcial de las salidas del proceso

Junto con la prioridad del OB, para la CPU se parametriza qué imagen parcial del proceso se asigna a cada OB.



### Error de acceso a la periferia (PZF) al actualizar la imagen del proceso

Cuando se produce un error durante la actualización de la imagen del proceso, la respuesta preajustada de las familias de CPUs (S7-300 y S7-400) es diferente:

- S7-300: No hay registro en el búfer de diagnóstico, no hay llamada a OB, los bytes de entrada/salida correspondientes se ponen a 0.
- S7-400: Registro en el búfer de diagnóstico y arranque del OB 85 con cada acceso a la periferia en cada actualización de la imagen correspondiente del proceso. Los bytes de entrada/salida defectuosos se ponen a 0.

En CPUs nuevas (a partir de 4/99) puede reparametrizar la respuesta en caso de que se produzcan errores de acceso a la periferia, de tal manera que la CPU

- genere un registro en el búfer de diagnóstico y arranque el OB 85 sólo con PZF entrante y saliente, o
- muestre el comportamiento preajustado para la S7-300 (sin llamada a OB-85), o
- muestre el comportamiento preajustado para la S7-400 (llamada a OB-85 con cada acceso concreto)

### ¿Con qué frecuencia arranca el OB 85?

Además de la respuesta a PZF parametrizada (PZF entrante/saliente o con cada acceso a la periferia), el espacio de direcciones de un módulo también influye en la frecuencia con la que arranca el OB-85:

En un módulo con un espacio de direcciones hasta de palabra doble, el OB 85 arranca una vez, p. ej. en un módulo digital con un máximo de 32 entradas o salidas, o en un módulo analógico con 2 canales.

En módulos con espacio de direcciones mayor, el BO85 arranca tantas veces como deba accederse a él con comandos de palabra doble, p. ej. dos veces en un módulo analógico de 4 canales.

### A.2.3.3 Pila de datos locales

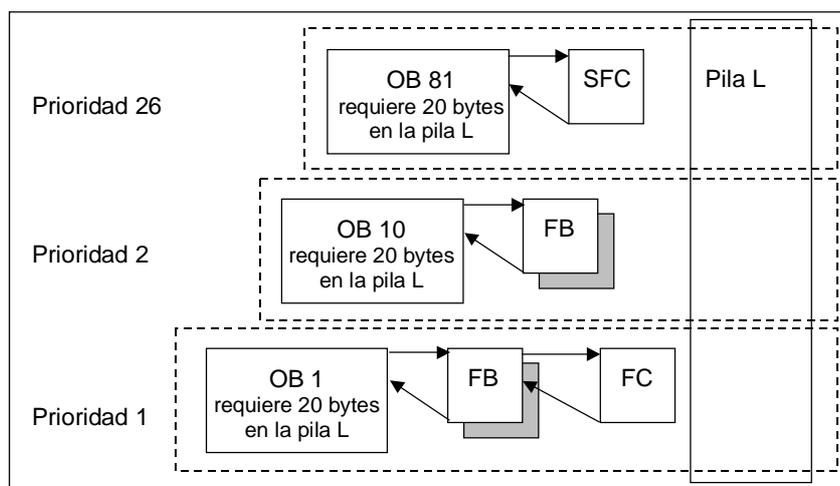
En la pila LSTACK se almacenan:

- las variables temporales de los datos locales de bloques
- la información de arranque de los bloques de organización
- informaciones para la transferencia de parámetros
- resultados intermedios de la lógica en programas escritos en Esquema de contactos.

Para crear bloques de organización se pueden declarar variables temporales (TEMP) que sólo están disponibles durante la ejecución del bloque y se pueden sobrescribir luego. Antes de efectuarse el primer acceso es preciso inicializar los datos locales. Además, cada bloque de organización requiere para su información de arranque 20 bytes de datos locales.

La CPU dispone de una memoria limitada para las variables temporales (datos locales) de los bloques en ejecución. El tamaño de dicha área de la memoria (es decir, de la pila de datos locales) depende del tipo de CPU. La pila de datos locales se divide en partes iguales entre las prioridades (ajuste estándar). Ello significa que cada prioridad dispone de su propia área de datos locales. Así se garantiza que también las prioridades altas, así como sus OBs asignados, dispongan de espacio para sus datos locales.

La figura siguiente muestra la asignación de datos locales a las prioridades. En este ejemplo, la pila LSTACK del OB 1 es interrumpida por el OB 10 que, a su vez, es interrumpido por el OB 81.



#### Cuidado

Todas las variables temporales (TEMP) de un OB, así como sus bloques subordinados, se memorizan en la pila L. Si se utilizan muchos niveles de anidado para la ejecución de bloques, se podría desbordar la pila L.

Las CPUs S7 pasan al estado operativo STOP cuando se sobrepasa el tamaño admisible de la pila L para un programa.

Compruebe en el programa instrucciones el estado de la pila L (las variables temporales).

Tenga en cuenta el requerimiento de datos locales de los OBs de errores síncronos.

### **Asignación de datos locales a las prioridades**

No todas las prioridades requieren la misma cantidad de memoria en la pila de datos locales. Las CPUs S7-400 y la permiten parametrizar con STEP 7 distintos tamaños del área de datos locales para las diferentes prioridades. Las prioridades no utilizadas se pueden deseleccionar, permitiendo así, en las CPUs S7-400 y en la CPU 318 ampliar el área de memoria para las otras prioridades. Los OBs deseleccionados no se toman en cuenta en la ejecución del programa, lo que ahorra tiempo de cálculo.

En las otras CPUs S7-300, cada prioridad tiene asignada una cantidad fija de datos locales (256 bytes), que no se puede modificar.

#### **A.2.3.4 Pila de interrupción (USTACK)**

Si la ejecución del programa es interrumpida por un OB de mayor prioridad, el sistema operativo memoriza los contenidos actuales de los acumuladores y los registros de direcciones, así como el número y tamaño de los bloques de datos abiertos en la pila de interrupción (pila U o USTACK).

Finalizada la ejecución del nuevo OB, el sistema operativo carga las informaciones desde la pila U y continúa la ejecución del bloque interrumpido en el punto donde ocurrió dicha interrupción.

En el estado operativo STOP se puede leer con STEP 7 en la PG el contenido de la pila U. Así se puede detectar con mayor facilidad por qué la CPU ha pasado a "STOP".

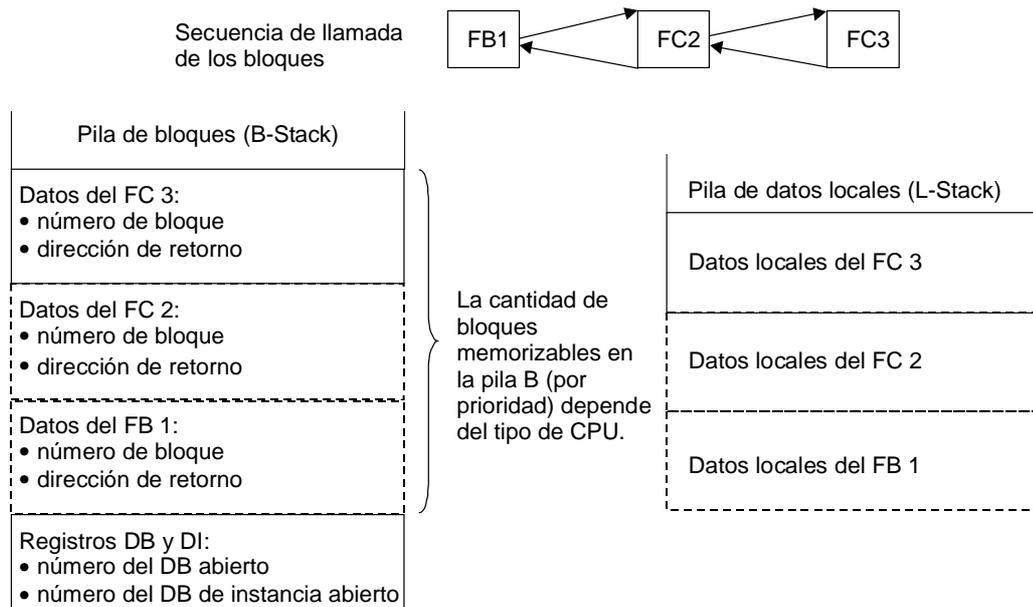
#### **A.2.3.5 Pila de bloques (BSTACK)**

Si el tratamiento de un bloque es interrumpido por la llamada de otro bloque o por una prioridad superior (tratamiento de una alarma o de un error), la pila BSTACK memoriza los datos siguientes:

- número, tipo (OB, FB, FC, SFB, SFC) y dirección de retorno del bloque que ha sido interrumpido.
- número de los bloques de datos (tomados de los registros DB y DI) que estaban abiertos en el momento de la interrupción.

Tras la interrupción, el programa de usuario puede continuar con los datos memorizados.

Si la CPU se encuentra en el estado operativo "STOP", es posible visualizar la pila B con STEP 7 en la PG. La pila B lista todos los bloques cuyo tratamiento no había sido concluido al momento en que la CPU fue llevada al estado operativo "STOP". Los bloques se visualizan en la secuencia en que se inició su tratamiento (v. fig. siguiente).



### Registros de bloques de datos

Se dispone de dos registros de bloques de datos que contienen los números de los bloques de datos abiertos, a saber:

- el registro DB contiene el número del bloque de datos globales abierto
- el registro DI contiene el número del bloque de datos de instancia abierto.

#### A.2.3.6 Búfer de diagnóstico

En el búfer de diagnóstico de la CPU se visualizan los eventos de diagnóstico en la secuencia de su aparición. La primera entrada contiene el evento más reciente. El número de entradas del búfer de diagnóstico depende del módulo en cuestión y de su estado operativo actual.

Los eventos de diagnóstico pueden ser:

- errores en un módulo
- errores en el cableado del proceso,
- errores de sistema en la CPU
- cambios de estado operativo de la CPU,
- error en el programa de usuario
- Eventos de diagnóstico de usuario (a través de la función del sistema SFC 52).

### A.2.3.7 Evaluación del búfer de diagnóstico

El búfer de diagnóstico es una parte de la lista de estado del sistema. En él se registran informaciones más detalladas acerca de los eventos de diagnóstico del sistema y los eventos definidos por el usuario en la secuencia de su aparición. La información que se inscribe en el búfer de diagnóstico es idéntica a la información de arranque que se transfiere al correspondiente bloque de organización.

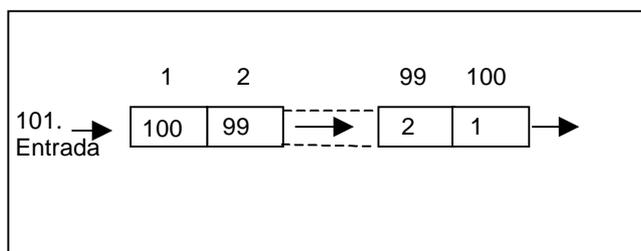
Las entradas en el búfer de diagnóstico no se pueden borrar. El contenido del búfer de diagnóstico permanece aún tras el borrado total.

El búfer de diagnóstico ofrece la posibilidad de:

- en caso de parada de la instalación: evaluar los últimos eventos antes del STOP y localizar la causa del STOP
- detectar las causas de los errores con más rapidez y aumentar así la disponibilidad de la instalación
- evaluar y optimizar el comportamiento dinámico de la instalación.

#### Organización del búfer de diagnóstico

El búfer de diagnóstico está concebido como búfer anular para un número máximo de entradas dependiente del módulo. Si se ha alcanzado dicho número y se presenta un nuevo evento de diagnóstico, se borrará la entrada más antigua. Entonces se desplazan todas las demás entradas. Por lo tanto, el evento de diagnóstico más reciente aparece siempre en primer lugar. En el caso de la CPU 314 S7-300, se trata p.ej. de 100 entradas:



El número de entradas visualizadas en el búfer de diagnóstico depende del módulo y de su estado operativo actual. En determinadas CPUs se puede parametrizar la longitud del búfer de diagnóstico.

#### Contenido del búfer de diagnóstico

El cuadro **superior** contiene una lista de todos los eventos de diagnóstico que se hayan presentado, incluyendo las siguientes informaciones:

- Número correlativo de la entrada (el evento más reciente es el número 1)
- Hora y fecha del evento de diagnóstico Se visualizan la hora y la fecha del módulo, si éste dispone de un reloj. Para poder aprovechar estos datos de forma apropiada es importante ajustar la hora y la fecha del módulo, así como comprobar los ajustes de vez en cuando;
- Texto del evento (descripción breve).

En el cuadro de texto **inferior** se muestran informaciones adicionales del evento seleccionado en la ventana superior. Dichas informaciones comprenden p.ej.:

- Número del evento
- Denominación del evento
- Cambio de estado operativo debido al evento de diagnóstico
- Indicación de la posición del error en un bloque (tipo y número de bloque, dirección relativa) que ocasionó la entrada del evento
- Evento entrante o saliente
- Informaciones adicionales específicas del evento.

Haciendo clic en el botón "Ayuda" se obtienen informaciones adicionales acerca del evento seleccionado en el cuadro de lista.

Encontrará explicaciones sobre las ID de eventos en la ayuda de referencia para los bloques y funciones de sistema (Saltos a descripciones de lenguajes, ayuda de bloques y atributos de sistema).

### **Guardar el contenido en un archivo de texto**

Para guardar el contenido del búfer de diagnóstico en un archivo de texto ASCII, pulse el botón "Guardar como" en la ficha "Búfer de diagnóstico" del cuadro de diálogo "Información del módulo".

### **Leer el búfer de diagnóstico**

El contenido del búfer de diagnóstico se puede visualizar mediante en la ficha "Búfer de diagnóstico" del cuadro de diálogo "Información del módulo", o bien en un programa a través de la función de sistema SFC 51 RDSYSST.

### **Ultima entrada antes de STOP**

Se puede definir que la última entrada del búfer de diagnóstico antes del cambio de RUN a STOP se emita automáticamente a un visualizador registrado (p.ej. PG, OP, TD) para garantizar que la causa del cambio de estado operativo a STOP se pueda detectar y corregir más rápidamente.

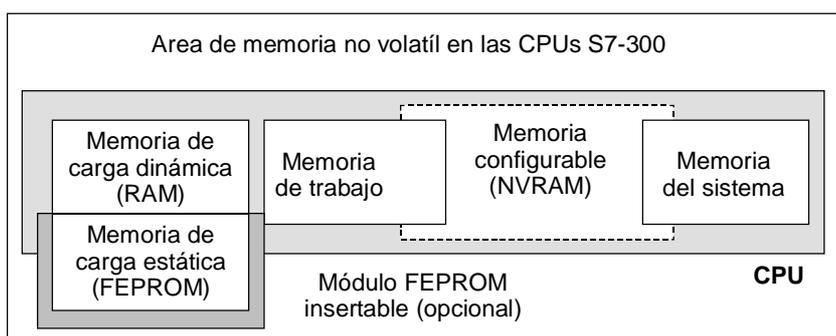
### A.2.3.8 Areas remanentes de la memoria en CPUs S7-300

En caso de corte de la alimentación o de borrado total (MRES), la memoria de la CPU S7-300 (memoria de carga dinámica (RAM), memoria de trabajo y memoria del sistema) es borrada, perdiéndose todos los datos memorizados en dichas áreas. Las CPUs S7-300 ofrecen las siguientes posibilidades para salvaguardar el programa y los datos:

- Los datos memorizados en las memorias de carga, de trabajo y en secciones de la memoria del sistema se pueden respaldar con una pila tampón.
- El programa se puede guardar en la EPROM (en una Memory Card o integrada en la CPU, v. el manual "Sistema de automatización S7-300, Configuración, instalación y datos de las CPU").
- Una determinada cantidad de datos dependientes de la CPU se puede salvaguardar en una área de la NVRAM no volátil.

#### Uso de la RAM no volátil

La CPU S7-300 ofrece un área en NVRAM (RAM no volátil) (v. fig. siguiente). Si el programa ha sido depositado en la EPROM de la memoria de carga, algunos de los datos (en caso de corte de alimentación o al pasar la CPU de STOP a RUN) se pueden memorizar también si se prevé la configuración correspondiente.



A tal efecto, la CPU se ha de ajustar de tal manera que los datos siguientes sean memorizados en la RAM no volátil:

- datos memorizados en un DB (esto sólo resulta ventajoso si el programa ha sido depositado en una EPROM de la memoria de carga)
- valores de temporizadores y contadores
- datos memorizados en marcas

En cada CPU se puede salvaguardar una cantidad determinada de temporizadores, contadores y marcas. Además, se ofrece una cantidad específica de bytes para memorizar los datos depositados en DB.

La dirección MPI de la CPU está depositada en la NVRAM. Esto garantiza la comunicabilidad de la CPU incluso tras un corte de la alimentación o tras el borrado total.

#### Utilizar el respaldo por pila para almacenar los datos

El respaldo por pila permite conservar de forma remanente el contenido de las memorias de carga y de trabajo al producirse un corte de alimentación. Independientemente de dicho respaldo, si en la configuración se memorizaron los temporizadores, contadores y marcas en la RAM no volátil (NVRAM), entonces tampoco se perderán estas informaciones.

## Configurar los datos de la NVRAM

Al configurar la CPU con STEP 7 se puede definir qué áreas de memoria deben ser remanentes, es decir, no volátiles.

La cantidad de memoria configurable en la NVRAM depende del tipo de CPU. No es posible salvaguardar más datos a los indicados para la correspondiente CPU.

### A.2.3.9 Areas remanentes de la memoria en CPUs S7-400

#### Modo sin respaldo

En caso de falta de corriente o de borrado total (MRES) en modo sin respaldo se borra la memoria de la CPU S7-400 (memoria de carga dinámica (RAM), memoria de trabajo y memoria del sistema), perdiéndose todos los datos memorizados en dichas áreas.

En modo sin respaldo sólo es posible el arranque completo y no existen áreas de memoria remanentes. Tras un corte de tensión sólo se conservan los parámetros MPI (p. ej. la dirección MPI de la CPU). Esto garantiza la comunicabilidad de la CPU tras una interrupción de corriente o un borrado total.

#### Modo respaldado

En modo respaldado

- se conservan completamente todas las áreas RAM al reanunciar tras un corte de tensión.
- se borran en el arranque completo las áreas de operandos correspondientes a las marcas, temporizadores y contadores. Se conservan los contenidos de los bloques de datos.
- se conserva el contenido de la memoria de trabajo RAM, exceptuando las marcas, temporizadores y contadores parametrizados como no remanentes.

#### Configurar áreas de datos no remanentes

Conforme al tipo de CPU, se pueden configurar como remanentes una determinada cantidad de marcas, temporizadores y contadores. Así, estos datos se conservan en caso de arranque completo en modo respaldado.

Al parametrizar con STEP 7 se define qué marcas, temporizadores y contadores deben ser remanentes en caso de arranque completo. No es posible respaldar más datos que la cantidad admisible para el tipo de CPU correspondiente.

Para obtener informaciones más detalladas acerca de la parametrización de las áreas de memoria remanentes, consulte el manual de referencia "Sistemas de automatización S7-400, M7-400, Datos de los módulos".

### A.2.3.10 Objetos configurables en la memoria de trabajo

En el caso de algunas CPUs se puede ajustar mediante HW Config el tamaño de objetos tales como los datos locales o el búfer de diagnóstico. Si, por ejemplo, reduce los valores predeterminados, se dispondrá de una mayor parte de la memoria de trabajo para otros fines. Los ajustes de dichas CPUs se pueden visualizar en la ficha "Memoria" de la información del módulo (botón "Detalles").

Tras cambiar la configuración de la memoria y cargar los datos en el sistema de destino, es preciso efectuar un arranque en frío de éste último para que los cambios tengan efecto.

## A.3 Tipos de datos y de parámetros

### A.3.1 Introducción a los tipos de datos y de parámetros

Todos los datos utilizados en el programa de usuario se deben identificar con un tipo de datos. Se distingue entre:

- tipos de datos simples ofrecidos por STEP 7,
- tipos de datos compuestos generados combinando tipos de datos simples y
- tipos para definir los parámetros a transferir a los FBs o las FCs.

#### Informaciones generales

Las operaciones AWL, FUP o KOP utilizan objetos de datos de un tamaño determinado. Como su nombre indica, las operaciones lógicas de combinación de bits utilizan bits. Las operaciones de carga y transferencia (AWL), así como las operaciones de transferencia (FUP y KOP) utilizan bytes, palabras y palabras dobles.

Un bit es una cifra binaria ("0" o "1"). Un byte comprende 8 bits, una palabra se compone de 16 bits y una palabra doble de 32 bits.

Las operaciones aritméticas utilizan también bytes, palabras o palabras dobles. En estos operandos de bytes, palabras o palabras dobles se pueden codificar números de diversos formatos tales como enteros y números en coma flotante.

Si utiliza el direccionamiento simbólico, deberá definir los símbolos e introducir un tipo de datos para los mismos (v. la tabla siguiente). Los diversos tipos de datos tienen diferentes opciones de formato y representaciones numéricas.

En el presente capítulo sólo se describen posibles notaciones para números y constantes. En la tabla siguiente figuran formatos de números y de constantes que no se explican detalladamente.

Formato	Tamaño en bits	Representación numérica
Hexadecimal	8, 16 y 32	B#16#, W#16# y DW#16#
Binario	8, 16 y 32	2#
Fecha IEC	16	D#
Tiempo IEC	32	T#
Hora	32	TOD#
BYTE	8	'A'

### A.3.2 Tipos de datos simples

Cada tipo de dato simple tiene una longitud definida. La tabla siguiente muestra los tipos de datos simples.

Tipo y descripción	Tam. en bits	Opciones de formato	Márgenes y representación numérica (del valor inferior hasta el valor superior)	Ejemplo
BOOL (bit)	1	Texto booleano	TRUE/FALSE	TRUE
BYTE (byte)	8	Número hexadecimal	B16#0 hasta B16#FF	L B#16#10 L byte#16#10
WORD (palabra)	16	Número binario Número hexadecimal BCD Número decimal sin signo	2#0 hasta 2#1111_1111_1111_1111 W#16#0 hasta W#16#FFFF C#0 hasta C#999 B#(0,0) hasta B#(255,255)	L 2#0001_0000_0000_0000 L W#16#1000 L word16#1000 L C#998 L B#(10,20) L byte#(10,20)
DWORD (palabra doble)	32	Número binario Número hexadecimal Número decimal sin signo	2#0 hasta 2#1111_1111_1111_1111_1111_1111_1111_1111 DW#16#0000_0000 hasta DW#16#FFFF_FFFF B#(0,0,0,0) hasta B#(255,255,255,255)	2#1000_0001_0001_1000_1011_1011_0111_1111 L DW#16#00A2_1234 L dword#16#00A2_1234 L B#(1, 14, 100, 120) L byte#(1,14,100,120)
INT (número entero)	16	Número decimal con signo	-32768 hasta 32767	L 1
DINT (entero de 32 bits)	32	Número decimal con signo	L#-2147483648 hasta L#2147483647	L L#1
REAL (número en coma flotante)	32	IEEE Número en coma flotante	Límite superior: $\pm 3.402823e+38$ Límite inferior: $\pm 1.175495e-38$	L 1.234567e+13
S5TIME (tiempo SIMATIC)	16	Tiempo S7 en pasos de 10 ms (valor estándar)	S5T#0H_0M_0S_10MS hasta S5T#2H_46M_30S_0MS y S5T#0H_0M_0S_0MS	L S5T#0H_1M_0S_0MS L S5TIME#0H_1H_1M_0S_0MS
TIME (tiempo IEC)	32	Tiempo IEC en intervalos de 1 ms, número entero con signo	- T#24D_20H_31M_23S_648MS hasta T#24D_20H_31M_23S_647MS	L T#0D_1H_1M_0S_0MS L TIME#0D_1H_1M_0S_0MS
DATE (fecha IEC)	16	Fecha IEC en pasos de 1 día	D#1990-1-1 hasta D#2168-12-31	L D#1994-3-15 L DATE#1994-3-15
TIME_OF_DAY (hora)	32	Hora en intervalos de 1 ms	TOD#0:0:0.0 hasta TOD#23:59:59.999	L TOD#1:10:3.3 L TIME_OF_DAY#1:10:3.3
Número entero (carácter)	8	Caracteres ASCII	'A','B' etc.	L 'E'



### A.3.2.3 Formato del tipo de datos REAL (números en coma flotante)

Los números en coma flotante se representan de la manera habitual "Número =  $m * b$  elevado a  $E$ ". La base " $b$ " y el exponente " $E$ " son números enteros, la mantisa " $m$ " un número racional.

Esta representación ofrece la ventaja de poder representar valores muy grandes y muy pequeños en un espacio muy limitado. Con el número limitado de bits para mantisa y exponente se cubre otro margen numérico.

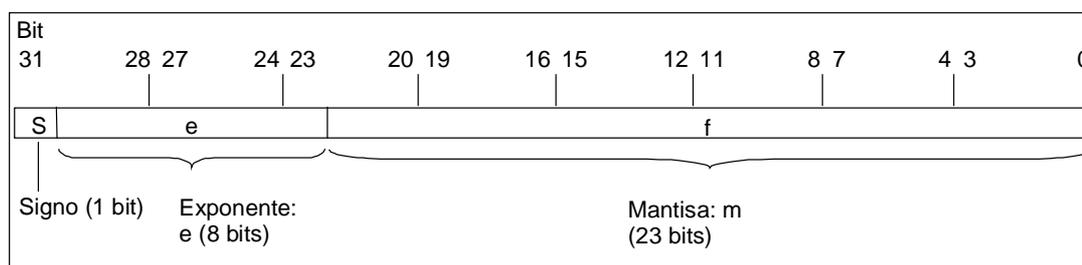
La desventaja radica en la limitada exactitud del cálculo: por ejemplo al formar la suma de dos números hay que igualar los exponentes (suma de las mantisas de dos números de igual exponente) desplazando la mantisa (coma flotante).

#### Coma flotante en STEP 7

Los números en coma flotante corresponden en STEP 7 al formato básico de ancho simple como se describe en la norma ANSI/IEEE Standard 754-1985, *IEEE Standard for Binary Floating-Point Arithmetic*. Están formados por los siguientes componentes:

- el signo  $S$
- el exponente  $e = E + \text{Bias}$  elevado a una constante ( $\text{Bias} = +127$ )
- la fracción de la mantisa  $m$ .  
La parte entera de la mantisa no se guarda, ya que siempre es 1 dentro del margen permitido

Los tres componentes ocupan en total una palabra doble (32 bits):



La tabla siguiente muestra el valor de los distintos bits en formato de coma flotante.

Componente del número en coma flotante	Bit	Valor
Signo $S$	31	
Exponente $e$	30	$2$ elevado a $7$
...	...	...
Exponente $e$	24	$2$ elevado a $1$
Exponente $e$	23	$2$ elevado a $0$
Mantisa $m$	22	$2$ elevado a $-1$
...	...	...
Mantisa $m$	1	$2$ elevado a $-22$
Mantisa $m$	0	$2$ elevado a $-23$

Los tres componentes **S**, **e** y **m** permiten definir el valor de un número representado en este formato aplicando la fórmula:

$$\text{Número} = 1.m * 2^{\text{e}-\text{Bias}}$$

Considerando que:

- e:  $1 \leq e \leq 254$
- Bias: Bias = 127. Así no hace falta un signo adicional para el exponente.
- S: si el número es positivo, S = 0 y si es negativo, S = 1.

### Margen numérico de los números en coma flotante

Del formato de coma flotante que acabamos de describir resulta:

- Número en coma flotante más bajo =  $1.0 * 2^{\text{elevado a } (1-127)} = 1.0 * 2^{\text{elevado a } (-126)}$   
= 1.175 495E-38
- Número en coma flotante más alto =  $2-2^{\text{elevado a } (-23)} * 2^{\text{elevado a } (254-127)} = 2-2^{\text{elevado a } (-23)} * 2^{\text{elevado a } (+127)}$   
= 3.402 823E+38

El número 0 se representa con e = m = 0; e = 255 y m = 0 significan "infinito".

Formato	Margen <sup>1)</sup>
Números en coma flotante según ANSI/IEEE Standard	-3.402 823E+38 hasta -1.175 495E-38 y 0 y +1.175 495E-38 hasta +3.402 823E+38

La tabla siguiente muestra el estado de señal de los bits de la palabra de estado cuando los resultados de las operaciones con números en coma flotante no se encuentran dentro del margen permitido.

Margen no permitido para un resultado	A1	A0	OV	OS
-1.175494E-38 < resultado < -1.401298E-45 (número negativo) rebase por defecto	0	0	1	1
+1.401298E-45 < resultado < +1.175494E-38 (número positivo) rebase por defecto	0	0	1	1
Resultado < -3.402823E+38 (número negativo) rebase por exceso	0	1	1	1
Resultado > 3.402823E+38 (número positivo) rebase por exceso	1	0	1	1
Número u operación no válida (valor de entrada fuera del margen permitido)	1	1	1	1

**Atención con las operaciones matemáticas:**

El resultado "Número en coma flotante no válido" se obtiene, por ejemplo, cuando se intenta extraer la raíz cuadrada de  $-2$ . Por consiguiente, cuando trabaje con operaciones matemáticas evalúe primero los bits de estado antes de seguir calculando con el resultado.

**Atención al "Forzar variables":**

Si deposita los valores de las operaciones en coma flotante p.ej. en una palabra doble de marcas, estos valores se pueden modificar con cualquier configuración binaria. Sin embargo, no todas representan un número válido.

**Precisión en cálculos con números en coma flotante****Cuidado**

En las operaciones de cálculo con números que abarquen p. ej. varias potencias de 10, el resultado puede no ser exacto.

Los números en coma flotante se representan en STEP 7 con una precisión de 6 dígitos detrás de la coma, por lo que al indicar constantes en coma flotante no podrá indicar más de 6.

**Nota**

La precisión de 6 dígitos detrás de la coma significa p. ej., que la suma de número1 + número2 = número1, si el número1 es mayor que el número2 \* 10 elevado a y, siendo  $y > 6$ :

$$100\,000\,000 + 1 = 100\,000\,000.$$

**Ejemplos de formatos de números en coma flotante**

La figura siguiente muestra el formato de números en coma flotante para los siguientes valores decimales:

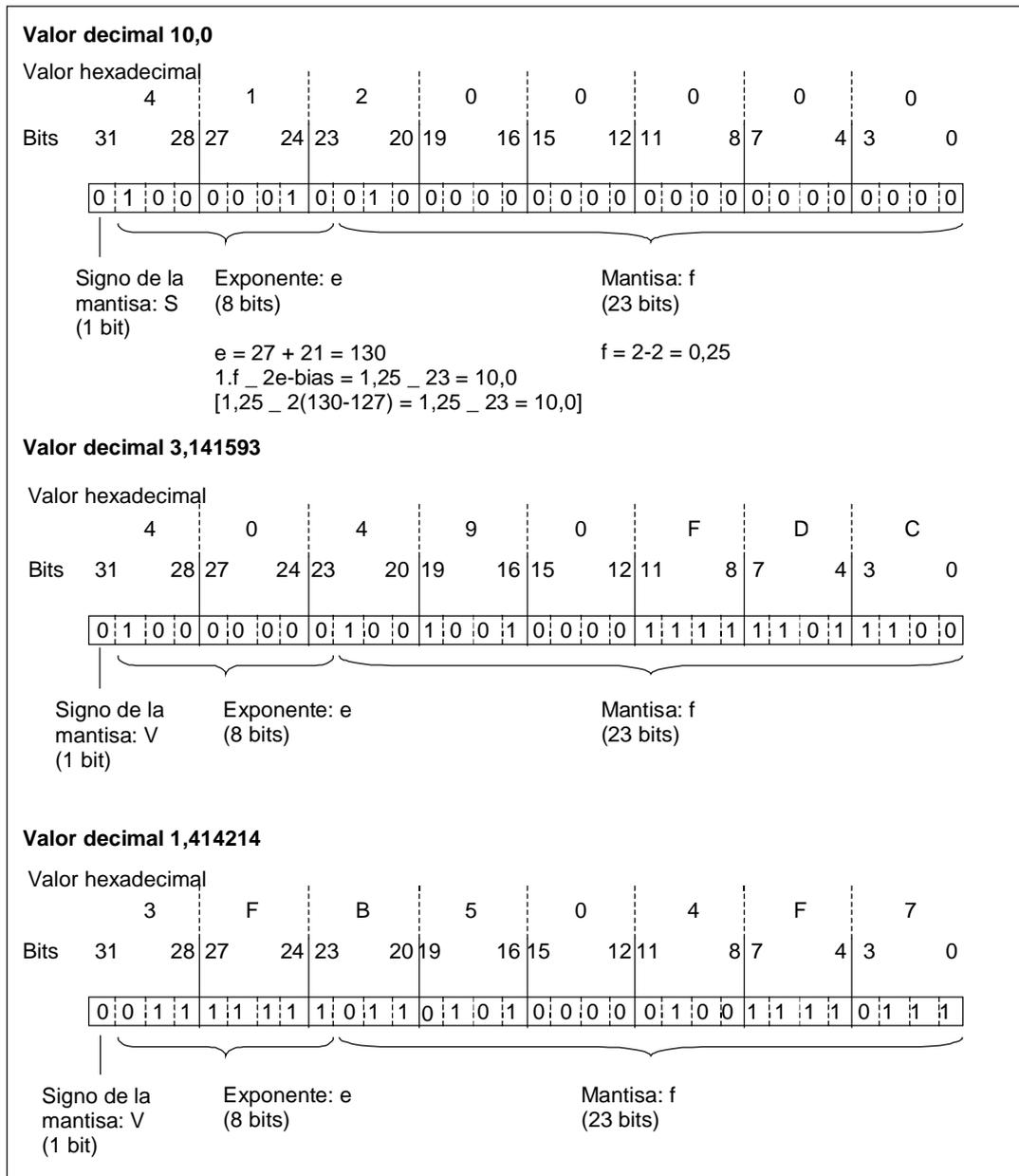
- 10,0
- p (3,141593)
- Raíz cuadrada de 2 ( $p^2 = 1,414214$ )

En el primer ejemplo, el número **10.0** resulta del formato en coma flotante (representación hexadecimal: 4120 0000) según la siguiente fórmula:

$$e = 2 \text{ elevado a } 1 + 2 \text{ elevado a } 7 = 2 + 128 = 130$$

$$m = 2 \text{ elevado a } (-2) = 0,25$$

De lo que resulta:  $1.m * 2 \text{ elevado a } (e - \text{Bias}) = 1.25 * 2 \text{ elevado a } (130 - 127) = 1.25 * 2 \text{ elevado a } 3 = 10.0.$



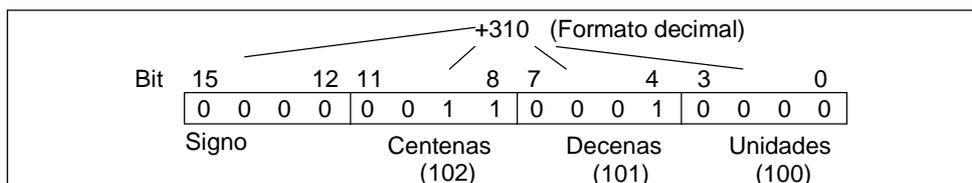
### A.3.2.4 Formato de los tipos de datos WORD y DWORD en los números decimales codificados en binario

La notación codificada en binario (BCD) representa un número decimal en grupos de cifras binarias (bits). Un grupo de 4 bits representa una cifra de un número decimal o bien el signo de dicho número. Los grupos de 4 bits constituyen una palabra (16 bits) o una palabra doble (32 bits). Los cuatro bits más significativos indican el signo del número ("1111" significa "negativo" y "0000" significa "positivo"). Las instrucciones con operandos BCD sólo evalúan el bit más significativo (15 en el caso del formato de palabra o 31 en el de palabra doble). La tabla siguiente muestra el formato y el margen de los dos tipos de números BCD.

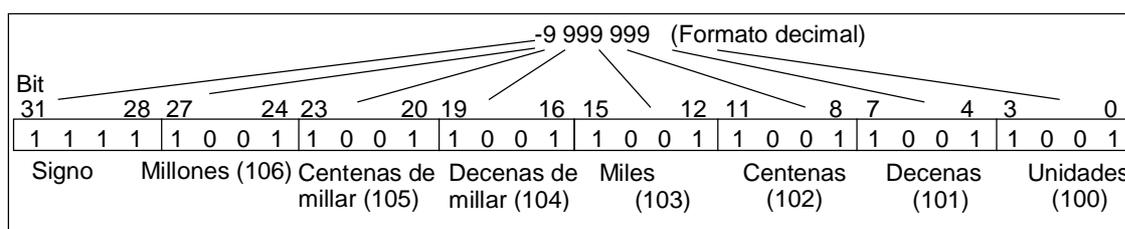
Formato	Grupo
Palabra (16 bits, número BCD de 3 cifras con signo)	-999 hasta +999
Palabra doble (32 bits, número BCD de 7 cifras con signo)	-9 999 999 hasta +9 999 999

Las figuras siguientes muestran ejemplos de un número decimal codificado en binario en los siguientes formatos:

- Formato de palabra

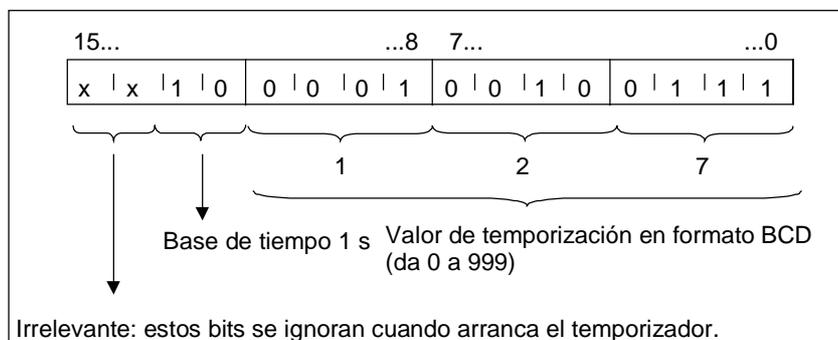


- Formato de palabra doble



### A.3.2.5 Formato del tipo de datos S5TIME (intervalo)

Si introduce el intervalo con el tipo de datos S5TIME, sus entradas se guardarán en formato BCD. La figura siguiente muestra el contenido del operando de tiempo, donde el valor de tiempo es 127 y la base de tiempo es 1 s.



Al trabajar con S5TIME deberá introducir un valor de tiempo comprendido entre 0 y 999, así como determinar una base de tiempo (v. la tabla siguiente). La base de tiempo indica el intervalo en el que un temporizador reduce el valor de tiempo en una unidad hasta alcanzar "0".

Base de tiempo para S5TIME

Base de tiempo	Código binario para la base de tiempo
10 ms	00
100 ms	01
1 s	10
10 s	11

Con la siguiente sintaxis se puede cargar un valor de tiempo predefinido:

- L<sup>1)</sup> W#16#wxyz
  - con: w = base de tiempo (es decir, intervalo de tiempo o resolución)
  - xyz = valor de tiempo en formato BCD
- L<sup>1)</sup> S5T#aH\_bbM\_ccS\_dddMS
  - con: a = horas, bb = minutos, cc = segundos y ddd = milisegundos.
  - La base de tiempo se selecciona automáticamente y el valor se redondea al número inferior siguiente que tenga dicha base de tiempo.

Se puede introducir un valor de tiempo de máx. 9 990 segundos (ó 2H\_46M\_30S).

<sup>1)</sup> = indicar L sólo en la programación AWL

### A.3.3 Tipos de datos compuestos

Los tipos de datos compuestos se definen como grupos de datos superiores a 32 bits o grupos de datos compuestos de varios tipos de datos. STEP 7 admite los siguientes tipos de datos compuestos:

- DATE\_AND\_TIME
- STRING
- ARRAY (campo o matriz)
- STRUCT (estructura)
- UDT (tipos de datos de usuario)
- FB y SFB

La tabla siguiente muestra los tipos de datos compuestos. Las estructuras y los campos se definen bien en la declaración de variables del bloque lógico o bien en un bloque de datos.

Tipo de datos	Descripción
DATE_AND_TIME DT	Define un área de 64 bits (8 bytes). Este tipo de datos memoriza en formato decimal codificado en binario:
STRING	Define un grupo de un máximo de 254 caracteres (tipo de datos CHAR). El área estándar reservada para una cadena de caracteres consta de 256 bytes. Este es el espacio requerido para memorizar 254 caracteres y un encabezamiento de 2 bytes. La capacidad de memoria requerida para una cadena de caracteres se puede reducir definiendo también la cantidad de caracteres a memorizar en dicha cadena (p.ej.: string[9] 'Siemens').
ARRAY	Define un agrupamiento multidimensional, similar a una matriz, de un tipo de datos (simple o compuesto). Por ejemplo: "ARRAY [1..2,1..3] OF INT" define un campo en formato de 2 x 3 números enteros. A los datos memorizados en un campo se accede a través del índice ("[2,2]"). En un campo se pueden definir hasta un máximo de 6 dimensiones. El índice puede ser un número entero discrecional (-32768 a 32767).
STRUCT	Define un agrupamiento de tipos de datos combinados discrecionalmente. Por ejemplo, se puede definir un campo compuesto de estructuras o una estructura compuesta de estructuras y campos.
UDT	Permite estructurar grandes cantidades de datos, simplificando así la entrada de tipos de datos al crear bloques de datos o al declarar las variables en la declaración correspondiente. STEP 7 permite combinar tipos de datos compuestos y simples, creando así un tipo de datos propio "de usuario" (UDT). UDTs tienen un nombre propio y, por consiguiente, pueden utilizarse varias veces.
FB, SFB	Determinan la estructura del bloque de datos de instancia asignado y permiten la transferencia de datos de instancia para varias llamadas de FB en un DB de instancia.

Los tipos de datos estructurados se depositan alineados palabra por palabra (WORD aligned).

### A.3.3.1 Formato del tipo de datos DATE\_AND\_TIME (fecha y hora)

Si se introduce la fecha y la hora utilizando el tipo de datos DATE\_AND\_TIME (DT), sus entradas se almacenarán en formato BCD en 8 bytes. El tipo de datos DATE\_AND\_TIME abarca el siguiente margen:

- DT#1990-1-1-0:0:0.0 a DT#2089-12-31-23:59:59.999

Los siguientes ejemplos muestran la sintaxis para introducir la fecha y la hora del jueves 25 de diciembre de 1993, 8:01 horas y 1,23 segundos. Son posibles los siguientes formatos:

- DATE\_AND\_TIME#1993-12-25-8:01:1.23
- DT#1993-12-25-8:01:1.23

Se dispone de las siguientes funciones estándar IEC (International Electrotechnical Commission) para trabajar con el tipo de datos DATE\_AND\_TIME:

- Convertir la fecha y la hora al formato DATE\_AND\_TIME  
FC3: D\_TOD\_DT
- Extraer la fecha del formato DATE\_AND\_TIME  
FC6: DT\_DATE
- Extraer el día de la semana del formato DATE\_AND\_TIME  
FC7: DT\_DAY
- Extraer la hora del formato DATE\_AND\_TIME  
FC8: DT\_TOD

La tabla siguiente muestra el contenido de los bytes que comprenden las informaciones sobre la fecha y la hora. El ejemplo muestra la fecha y la hora del jueves 25 de diciembre de 1993, 8:01 horas y 1,23 segundos.

Byte	Contenido	Ejemplo
0	Año	B#16#93
1	Mes	B#16#12
2	Día	B#16#25
3	Hora	B#16#08
4	Minuto	B#16#01
5	Segundo	B#16#01
6	Las dos cifras más significativas de MSEC	B#16#23
7 (4MSB)	Las dos cifras menos significativas de MSEC	B#16#0
7 (4LSB)	Día de la semana 1 = Domingo 2 = Lunes ... 7 = Sábado	B#16#5

El margen admisible para el tipo de datos "DATE\_AND\_TIME" es:

- mín.: DT#1990-1-1-0:0:0.0
- máx.: DT#2089-12-31-23:59:59.999

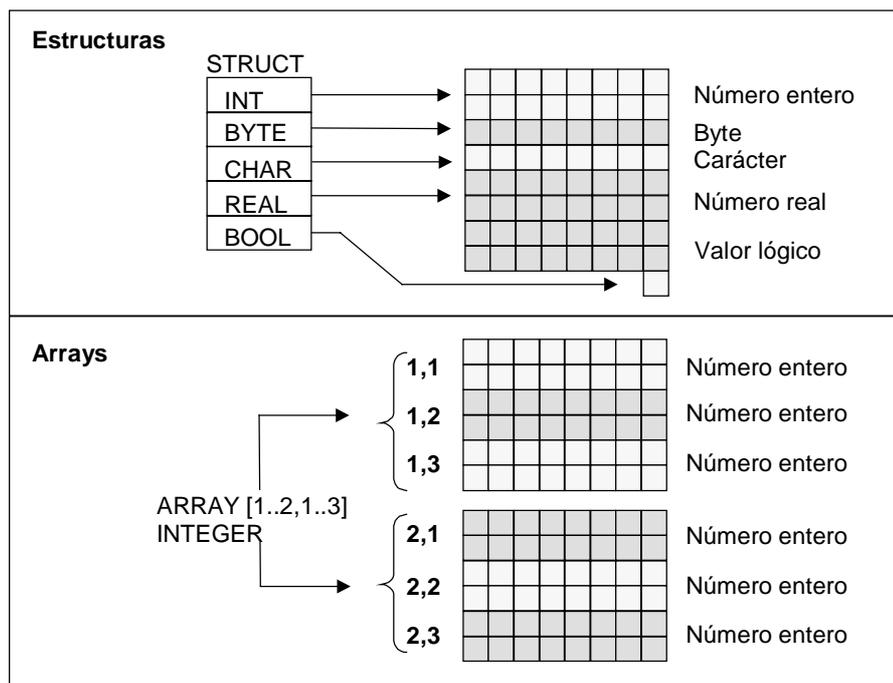
	Margen posible de valores	Código BCD
Año	1990 – 1999 2000 – 2089	90h – 99h 90h – 99h
Mes	1 – 12	01h – 12h
Día	1 – 31	01h – 31h
Hora	00 – 23	00h – 23h
Minuto	00 – 59	00h – 59h
Segundo	00 – 59	00h – 59h
Milisegundo	0 – 999	000h – 999h
Día de la semana	Domingo – Sábado	1h – 7h

### A.3.3.2 Uso de tipos de datos compuestos

Es posible crear nuevos tipos de datos combinando los datos simples y compuestos con los siguientes tipos de datos compuestos:

- Array (tipo de datos ARRAY): Un array o matriz combina un grupo de datos de un tipo formando una unidad.
- Estructura (tipo de datos STRUCT): Una estructura combina datos de diferente tipo formando una unidad.
- Cadena de caracteres (tipo de datos STRING): Una cadena de caracteres define un array unidimensional con un máximo de 254 caracteres (tipo de datos CHAR). Una cadena de caracteres se puede transferir únicamente como unidad. La longitud de la cadena de caracteres debe coincidir en los parámetros formal y actual del bloque.
- Fecha y hora (tipo de datos DATE\_AND\_TIME): La fecha y la hora memorizan año, mes, día, horas, minutos, segundos, milisegundos y día de la semana.

La figura siguiente muestra cómo estructurar los arrays y las estructuras de tipos de datos en un área para memorizar informaciones. Un campo o una estructura se puede definir en un DB o en una declaración de variables de un FB, OB o de una FC.



### A.3.3.3 Uso de arrays para acceder a los datos

#### Arrays

Un array combina un grupo de datos de un tipo (simple o compuesto) formando una unidad. Un array no se puede formar a partir de otros arrays. Los arrays se definen de la siguiente manera:

- Indicar el nombre del array.
- Declarar un array con la clave ARRAY.
- Indicar el tamaño del array a través de un índice. Se ha de indicar el primer y el último número de las diferentes dimensiones (máximo 6) en el array. El índice se indica en corchetes, separando cada dimensión por coma mientras que el primer y el último número de cada dimensión se separan por dos puntos suspensivos. El índice siguiente define, p. ej., un array tridimensional:

[1..5,-2..3,30..32]

- Indicar el tipo de los datos que se han de memorizar en el array.

#### Ejemplo 1

La figura siguiente muestra un array compuesto de tres números enteros. A través del índice se accede a los datos memorizados en un array. El índice consiste en el número escrito entre corchetes. El índice del segundo entero es p. ej. Temperatura\_servicio[2].

El índice puede ser un entero cualquiera (-32768 a 32767), incluidos valores negativos. El array de la figura siguiente podría ser definido también como ARRAY [-1..1]. El índice del primer entero sería entonces Temp\_servicio[-1], el segundo Temp\_servicio[0] y el tercer entero Temp\_servicio[1].

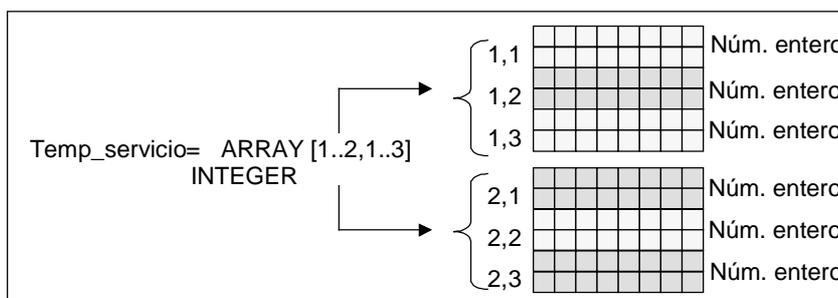
Dirección	Nombre	Tipo	Val. inicial	Comentario
0.0		STRUCT		
+0.0	Temp_servic	ARRAY[1..3]		
*2.0		INT		
=3.0		END_STRUCT		

Temp\_servicio =  
ARRAY [1..3] INTEGER

## Ejemplo 2

Un array puede definir también un grupo multidimensional de tipos de datos. La figura siguiente muestra un array bidimensional compuesto de números enteros.



A los datos de un array multidimensional se accede a través del índice. En el ejemplo, el primer entero es Temp\_servicio[1,1], el tercero el Temp\_servicio[1,3], el cuarto el Temp\_servicio[2,1] y el sexto es Temp\_servicio[2,3].

Para un array se puede definir hasta un máximo de seis dimensiones (seis índices). Por ejemplo, la variable Temp\_servicio se puede definir como array hexadimensional:

ARRAY [1..3,1..2,1..3,1..4,1..3,1..4]

El índice del primer elemento en este array es Temp\_servicio[1,1,1,1,1,1]. El índice del último elemento es Temp\_servicio[3,2,3,4,3,4].

## Crear arrays

Los arrays se definen declarando los datos en un DB o en la declaración de variables. Al declarar un array se ha de indicar la palabra clave (ARRAY) y luego el tamaño entre corchetes:

[límite inferior..límite superior]

En un array multidimensional se indican los límites superior e inferior adicionales y se separan las diferentes dimensiones por coma. La figura siguiente muestra la declaración para crear un array en formato 2 x 3.

Dirección	Nombre	Tipo	Val. inicial	Comentario
0.0		STRUCT		
+0.0	Calor_2x3	ARRAY[1..2,1..3]		
*2.0		INT		
=6.0		END_STRUCT		

### Introducir valores iniciales para un array

Al crear un array se puede asignar un valor inicial a cada elemento del array. STEP 7 ofrece dos posibilidades para entrar los valores iniciales:

- Entrada de valores individuales: para cada elemento del array se indica un valor válido (para el tipo de datos del campo). Indicar los valores en la secuencia de los elementos: [1,1]. Tenga en cuenta que los diferentes elementos se han de separar por coma.
- Indicar un factor de repetición: para elementos secuenciales que disponen del mismo valor inicial, se puede indicar la cantidad de elementos (el factor de repetición) y el valor inicial para dichos elementos. El formato para indicar el factor de repetición es  $x(y)$ , siendo  $x$  el factor de repetición e  $y$  el valor que ha de ser repetido.

Si se utiliza el array declarado en la figura anterior, el valor inicial para los seis elementos se puede indicar de la siguiente manera: 17, 23, -45, 556, 3342, 0. El valor inicial de todos los seis elementos también se puede poner a 10, indicando para ello 6(10). Para los primeros dos elementos pueden indicarse valores determinados y luego los cuatro restantes pueden ajustarse a 0, indicando 17, 23, 4(0).

### Acceder a datos en un array

A los datos memorizados en un array se accede a través del índice. El índice se utiliza con el nombre simbólico.

Ejemplo: Si el array declarado en la figura anterior empieza en el primer byte del DB20 (motor), se accede al segundo elemento en el array a través de la dirección siguiente:

Motor.Calor\_2x3[1,2].

### Utilizar arrays como parámetros

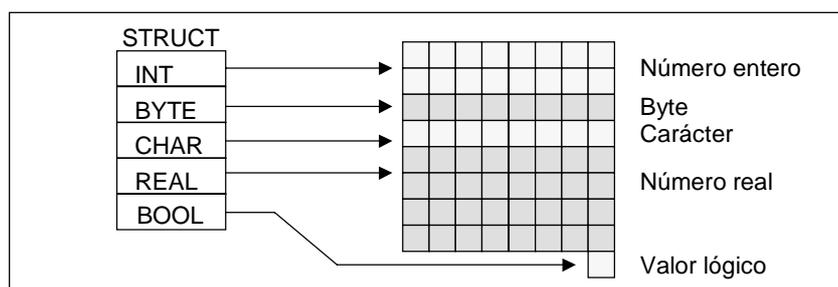
Los arrays se pueden transferir en calidad de parámetros. Si un parámetro ha sido declarado como ARRAY en la declaración de variables, entonces se ha de transferir el array completo (y no solamente elementos individuales). No obstante, es posible asignar un parámetro a un elemento del campo, llamando un bloque y siempre y cuando el elemento del array corresponda al tipo de datos del parámetro.

Si los arrays se utilizan como parámetros, los arrays no deben tener el mismo nombre (en realidad no necesitan ninguno). No obstante, ambos arrays (el parámetro formal y el actual) deben tener la misma estructura. Por ejemplo, un array en formato 2 x 3 enteros se puede transferir como parámetro solamente si el parámetro formal del bloque es un array en formato 2 x 3 enteros y también el parámetro actual, que se suministra a través de la operación de llamada, es un array en formato 2 x 3 enteros.

### A.3.3.4 Uso de estructuras para acceder a los datos

#### Estructuras

Una estructura combina diferentes tipos de datos (datos simples y compuestos, incluidos arrays y estructuras) formando una unidad. Así, los datos se pueden agrupar conforme al control para el proceso en cuestión. Esto permite también transferir parámetros como una unidad de datos y no como elementos individuales. La figura siguiente muestra una estructura compuesta de un entero, un byte, un carácter, un número en coma flotante y un valor booleano o lógico.



Una estructura se puede anidar en 8 niveles como máximo (p. ej. una estructura de estructuras que contenga arrays).

#### Crear una estructura

Las estructuras se definen durante la declaración de datos dentro de un DB o en la declaración de variables de un bloque lógico.

La figura siguiente muestra la declaración de una estructura (*Lote\_1*), que consta de los siguientes elementos: un entero (para memorizar la cantidad), un byte (para memorizar los datos originales), un carácter (para memorizar el código de control), un número en coma flotante (para memorizar la temperatura) y una marca booleana (para finalizar la señal).

Dirección	Nombre	Tipo	Val. inicial	Comentario
0.0	Pila_1	STRUCT		
+0.0	Cantidad	INT	100	
+2.0	Datos_originales	BYTE		
+4.0	Código_de_control	CHAR		
+6.0	Temperatura	REAL	120	
+8.1	Fin	BOOL	FALSE	
=10.0		END_STRUCT		

### Asignar valores iniciales para una estructura

Si se desea asignar un valor inicial a cada elemento de una estructura, entonces se ha de indicar un valor válido tanto para el tipo de datos como para el nombre del elemento. Por ejemplo, es posible asignar (a la estructura declarada en la figura anterior) los siguientes valores iniciales:

Cantidad	=	100	
Datos originales	=		B#(0)
Código de control	=	'Z'	
Temperatura	=	120	
Fin	=	False	

### Guardar datos y acceder a los mismos en estructuras

Debe acceder a cada uno de los elementos de una estructura. Puede utilizar direcciones simbólicas (p. ej. *Lote\_1.Temperatura*). Pero también se puede indicar la dirección absoluta bajo la cual se ha de memorizar el elemento (ejemplo: si *Lote\_1* está memorizado en el DB20 con inicio en el byte 0, la dirección absoluta para la *cantidad* es *DB20.DBW0* y la dirección para la *temperatura* es *DB20.DBD6*).

### Utilizar estructuras como parámetros

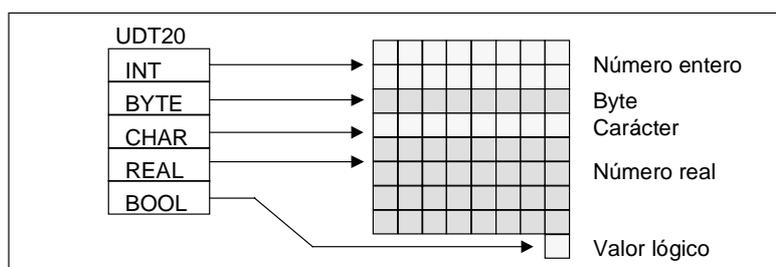
Las estructuras se pueden transferir en calidad de parámetros. Si un parámetro está declarado como STRUCT en la declaración de variables, entonces se ha de transferir una estructura con la misma configuración. No obstante, se puede asignar un elemento de una estructura a un parámetro al llamar el bloque, siempre que el elemento de la estructura corresponda al tipo de datos del parámetro.

Si se utilizan estructuras como parámetros, ambas estructuras (para el parámetro formal y para el parámetro actual) deben tener la misma configuración; es decir, los tipos de datos que sean iguales deben estar ordenados en el mismo orden.

### A.3.3.5 Uso de tipos de datos de usuario para acceder a los datos

#### Tipos de datos de usuario

Los tipos de datos de usuario (UDT) pueden combinar tipos de datos simples y compuestos. A los UDTs se les puede dar un nombre y utilizar varias veces. La figura siguiente muestra la estructura de un UDT compuesta de un entero, un byte, un carácter, un número en coma flotante y un valor booleano o lógico.



En vez de indicar todos los tipos de datos individualmente o como estructura, sólo es necesario indicar "UDT20" como tipo de datos y STEP 7 ordena automáticamente el espacio de memoria correspondiente.

#### Crear un tipo de datos de usuario

Los UDTs se definen con STEP 7. La figura siguiente muestra un UDT que consta de los siguientes elementos: un entero (para memorizar la cantidad), un byte (para memorizar los datos originales), un carácter (para memorizar el código de control), un número en coma flotante (para memorizar la temperatura) y una marca booleana (para finalizar la señal). Al UDT se le puede asignar un nombre simbólico en la tabla de símbolos (p. ej., *Datos proceso*).

Dirección	Nombre	Tipo	Val. inicial	Comentario
0.0	Pila_1	STRUCT		
+0.0	Cantidad	INT	100	
+2.0	Datos_originales	BYTE		
+4.0	Código_de_contr	CHAR		
+6.0	Temperatura	REAL	120	
+8.1	Fin	BOOL	FALSE	
=10.0		END_STRUCT		

Tras haber creado un UDT puede utilizarlo al igual que un tipo de datos, p.ej. al declarar para una variable el tipo de datos *UDT200* en un DB (o en la declaración de variables de un FB).

La figura siguiente muestra un DB con la variable *Datos\_proceso\_1* con el tipo de datos *UDT200*. Debe indicar tan sólo *UDT200* y *Datos\_proceso\_1*. Los campos representados en cursiva se crean al compilar el DB.

Dirección	Nombre	Tipo	Val. inicial	Comentario
0.0		STRUCT		
+6.0	<i>Datos_proceso_1</i>	UDT200		
=6.0		END_STRUCT		

### Asignar valores iniciales a un tipo de datos de usuario (UDT)

Si se desea asignar un valor inicial a cada elemento de un UDT, se ha de indicar un valor válido tanto para el tipo de datos como para el nombre del elemento. Por ejemplo se puede asignar (al UDT declarado en la figura anterior) los siguientes valores iniciales:

```
Cantidad           =      100
Datos originales   =      B#(0)
Código de control  =      'Z'
Temperatura        =      120
Fin                =      False
```

Si se declara una variable como UDT, entonces los valores iniciales de la variable son aquello que fueron indicados al crear el UDT.

### Guardar datos y acceder a los mismos en un tipo de datos de usuario (UDT)

Para acceder a cada uno de los elementos de un UDT puede utilizar direcciones simbólicas (p. ej. *Lote\_1.Temperatura*) o puede indicar la dirección absoluta en la cual se memorizará el elemento (ejemplo: si *Lote\_1* está memorizado en DB20 con inicio en el byte 0, entonces la dirección absoluta para la *cantidad* es *DB20.DBW0* y la dirección para la *temperatura* es *DB20.DBD6*).

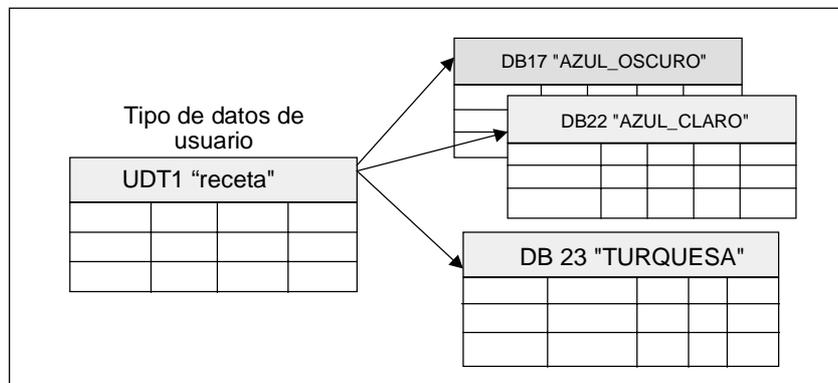
### Utilizar tipos de datos de usuario en calidad de parámetros

Las variables del tipo de datos UDT se pueden transferir en calidad de parámetros. Si el parámetro está declarado como UDT en la declaración de variables, entonces se ha de transferir un UDT que tenga la misma estructura que los elementos de datos. Sin embargo, un elemento de un UDT se puede asignar a un parámetro al llamar un bloque, siempre que el elemento del UDT coincida con el tipo de datos del parámetro.

### Ventaja de los DBs con UDT asociado

Con los UDTs ya creados se puede generar una gran cantidad de bloques de datos que tengan la misma estructura. Estos bloques de datos se pueden adaptar exactamente a la tarea correspondiente; para ello basta indicar valores actuales diferentes.

Si, p. ej., se estructura un UDT como receta (p. ej., para mezclar colores), entonces se puede asignar este UDT a varios DBs que contengan diferentes cantidades.



La estructura del bloque de datos queda definida por el UDT asociado.

### A.3.4 Tipos de parámetros

Además de los tipos de datos simples y compuestos, es posible definir tipos de parámetros para los parámetros formales que se transfieren entre los bloques. STEP 7 conoce los siguientes tipos de parámetros:

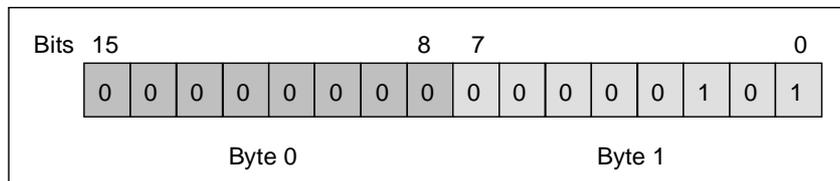
- **TIMER** o **COUNTER**: definen un determinado temporizador o contador que va a ser utilizado en la ejecución. Si se utiliza un parámetro formal del tipo **TIMER** o **COUNTER**, el parámetro actual correspondiente debe ser un temporizador o un contador, es decir, se debe indicar bien una "T" o una "Z" seguida por un número entero positivo.
- **BLOCK**: define un determinado bloque que ha de ser utilizado como entrada o como salida. La declaración del parámetro determina el tipo de bloque (FB, FC, DB etc.) que debe ser utilizado. Si se utiliza un parámetro formal del tipo **BLOCK**, se debe indicar la dirección del bloque como parámetro actual. Ejemplo: "FC101" (para direccionamiento absoluto) o "Válvula" (para direccionamiento simbólico).
- **POINTER**: accede a la dirección de una variable. Un puntero contiene una dirección en lugar de un valor. Si se utiliza un parámetro formal del tipo **POINTER**, se debe indicar la dirección como parámetro actual. STEP 7 permite indicar un puntero en formato Pointer o simplemente como dirección (p. ej., M 50.0). Ejemplo de un formato Pointer para direccionamiento de los datos que comienzan en M 50.0: P#M50.0
- **ANY**: se utiliza cuando el tipo de datos del parámetro actual no se conoce o cuando se puede utilizar cualquier tipo de datos. Para obtener informaciones más detalladas acerca del parámetro **ANY**, consulte los apartados "Formato del tipo de parámetro ANY o Uso del tipo de parámetro ANY".

Un tipo de parámetro puede ser también un tipo de datos de usuario (UDT). Para obtener informaciones más detalladas sobre los UDTs, consulte el apartado "Uso de tipos de datos de usuario para acceder a los datos".

Parámetro	Tamaño	Descripción
TIMER	2 bytes	Designa un temporizador determinado que ha de ser utilizado por el programa en el bloque lógico llamado. Formato: T1
COUNTER	2 bytes	Designa un contador determinado que ha de ser utilizado por el programa en el bloque lógico llamado. Formato: Z10
BLOCK_FB BLOCK_FC BLOCK_DB BLOCK_SDB	2 bytes	Designa un bloque determinado que ha de ser utilizado por el programa en el bloque lógico llamado. Formato: FC101 DB42
POINTER	6 bytes	Designa la dirección. Formato: P#M50.0
ANY	10 bytes	Se utiliza cuando el tipo de datos del parámetro actual no se conoce. Formato: P#M50.0 BYTE 10      Formato ANY en tipos de datos P#M100.0 WORD 5 L#1COUNTER 10      Formato ANY en tipos de parámetros

### A.3.4.1 Formato de los tipos de parámetros BLOCK, COUNTER, TIMER

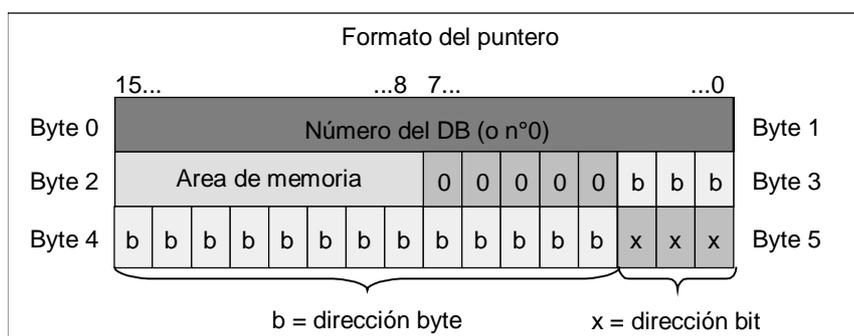
STEP 7 guarda los tipos de parámetros BLOCK, COUNTER y TIMER en forma de números binarios en una palabra (32 bits). La figura siguiente muestra el formato de dichos tipos de parámetros.



El número admisible de bloques, temporizadores y contadores depende del modelo de CPU S7. Para obtener informaciones más detalladas acerca del número admisible de temporizadores y contadores y el número máximo de bloques disponibles, consulte las hojas de datos de su CPU en los manuales "Sistema de automatización S7-300, Configuración, instalación y datos de las CPU" o "Sistemas de automatización S7-400, M7-400, Configuración e instalación".

### A.3.4.2 Formato del tipo de parámetro POINTER

STEP 7 guarda el tipo de parámetro POINTER en 6 bytes (48 bits). La figura siguiente muestra el tipo de datos que se guardan en cada byte.



El tipo de parámetro POINTER almacena las informaciones siguientes:

- Número de DB (0 en caso de que los datos no se guarden en un DB)
- Área de la memoria en la CPU (la tabla siguiente muestra los códigos hexadecimales de las áreas de memoria para el tipo de parámetro POINTER)

Código hexadecimal	Area de memoria	Descripción
b#16#81	E	Area de entradas
b#16#82	A	Area de salidas
b#16#83	M	Area de marcas
b#16#84	DB	Bloque de datos
b#16#85	DI	Bloque de datos de instancia
b#16#86	L	Pila de datos locales (LSTACK)
b#16#87	V	Datos locales precedentes

- Dirección de los datos (en formato byte.bit)

STEP 7 ofrece el formato de puntero: p#área de memoria byte.bit\_dirección. (Si el parámetro formal ha sido declarado como tipo POINTER, sólo se deberá indicar el área de la memoria y la dirección. STEP 7 convertirá automáticamente la entrada al formato de puntero.) Los ejemplos siguientes muestran cómo introducir el tipo de parámetro POINTER para los datos que comiencen en M50.0:

- P#M50.0
- M50.0 (si el tipo de parámetro se ha declarado como POINTER)

### A.3.4.3 Uso del tipo de parámetro POINTER

Los punteros se utilizan para señalar operandos. La ventaja de este tipo de direccionamiento es que el operando de la instrucción se puede modificar dinámicamente durante la ejecución del programa.

#### Puntero para el direccionamiento indirecto por memoria

Las instrucciones del programa que utilizan el direccionamiento indirecto por memoria se componen de una operación, un identificador del operando y un desplazamiento (el desplazamiento se debe indicar entre corchetes).

Ejemplo de un puntero en formato de palabra doble:

L	P#8.7	Cargar el valor del puntero en ACU1.
T	MD2	Transferir el puntero a MD2.
U	E [MD2]	Consultar el estado de señal en la entrada E 8.7,
=	A [MD2]	y asignar el estado de señal a la salida A 8.7.

#### Puntero para los direccionamientos intraárea e interárea

Las instrucciones de programación que utilizan estos tipos de direccionamiento se componen de una operación y las siguientes partes: identificador del operando, identificador del registro de direcciones, desplazamiento.

El registro de direcciones (AR1/2) y el desplazamiento se deben indicar juntos entre corchetes.

### Ejemplo del direccionamiento intraárea

El puntero no contiene una indicación del área de memoria:

L	P#8.7	Cargar el valor del puntero en ACU1.
LAR1		Cargar el puntero de ACU1 en AR1.
U	E [AR1, P#0.0]	Consultar el estado de señal en la entrada E 8.7 y
=	A [AR1, P#1.1]	asignar el estado de señal a la salida A 10.0.

El desplazamiento 0.0 no tiene efecto alguno. La salida 10.0 resulta haber sumado 8.7 (AR1) más el desplazamiento 1.1. El resultado es 10.0 y no 9.8 (véase el formato de puntero).

### Ejemplo del direccionamiento interárea

En el caso del direccionamiento interárea, el área de memoria (en el ejemplo: E ó A) se indica en el puntero.

L	P# E8.7	Cargar el valor del puntero y el identificador de área en ACU1.
LAR1		Cargar el área de memoria E y la dirección 8.7 en AR1.
L	P# A8.7	Cargar el valor del puntero y el identificador de área en ACU1.
LAR2		Cargar el área de memoria A y la dirección 8.7 en AR2.
U	[AR1, P#0.0]	Consultar el estado de señal en la entrada E 8.7 y
=	[AR2, P#1.1]	asignar el estado de señal a la salida A 10.0.

El desplazamiento 0.0 no tiene efecto alguno. La salida 10.0 resulta de sumar 8.7 (AR2) y 1.1 (desplazamiento). El resultado es 10.0 y no 9.8 (véase el formato de puntero).

#### A.3.4.4 Bloque para modificar el puntero

Con el bloque de ejemplo FC3 "Cambio de punteros" se puede modificar la dirección de bit o de byte de un puntero. Al llamar la FC, el puntero a cambiar se transfiere a la variable "Puntero" (se pueden utilizar punteros de direccionamiento intraárea o interárea en formato de palabra doble).

Con el parámetro "Bit-Byte" se puede modificar la dirección de bit o de byte del puntero (0: dirección de bit, 1: dirección de byte). En la variable "Inc\_valor" (en formato de entero) se indica el valor que se ha de sumar o restar al contenido de la dirección. También se pueden indicar números negativos para decrementar la dirección.

Al modificar la dirección de bit se efectúa un acarreo a la dirección de byte (también al decrementar); p.ej.:

- P#M 5.3, Bit\_Byte = 0, Inc\_valor= 6 => P#M 6.1 ó
- P#M 5.3, Bit\_Byte = 0, Inc\_valor= -6 => P#M 4.5.

La función no influye en la información de área del puntero.

La FC detecta si se exceden los límites superior o inferior del puntero. Entonces no se modifica el puntero y la variable de salida "RET\_VAL" (los errores se pueden corregir) se pone a "1" (hasta la siguiente ejecución correcta de la FC 3). Este es el caso cuando:

- 1. se ha elegido la dirección de bit, siendo Inc\_valor >7 ó <-7,
- 2. se ha elegido la dirección de bit o de byte y el cambio tendría por resultado una dirección de byte "negativa",
- 3. se ha elegido la dirección de bit o de byte y el cambio tendría por resultado una dirección de byte demasiado grande.

### Bloque de ejemplo en AWL para modificar el puntero:

```

FUNCTION FC 3: BOOL
TITLE =Cambio de punteros
//La FC 3 se puede utilizar para cambiar punteros.
AUTHOR : AUT1CS1
FAMILY : INDADR
NAME : ADRPOINT
VERSION : 0.0

VAR_INPUT
    Bit_Byte : BOOL ; //0: dirección de bit, 1: dirección de byte
    Inc_valor : INT ; //Incrementar (si valor neg.=> decrementar/si valor pos. =>
                    //incrementar)
END_VAR

VAR_IN_OUT
    Puntero : DWORD ; //el puntero a modificar
END_VAR

VAR_TEMP
    Inc_valor1 : INT ; //Valor intermedio incremento
    Puntero1 : DWORD ; //Valor intermedio puntero
    Val_int : DWORD ; //Variable auxiliar
END_VAR

```

```

BEGIN
NETWORK
TITLE =
//El bloque rechaza automáticamente los cambios que modifiquen las
//informaciones de área del puntero o que conduzcan a punteros "negativos".
    SET    ; //Poner RLO a 1 y
    R      #RET_VAL; //Desactivar rebose del límite superior
    L      #Puntero; //Asignar el valor intermedio temporal
    T      #Puntero1; //del puntero
    L      #Inc_valor; //Asignar el valor intermedio temporal
    T      #Inc_Wert1; //del incremento
    U      #Bit_Byte; //si =1, entonces operación de dirección de byte
    SPB    Byte; //Salto al cálculo de dirección de byte
    L      7; //Si el valor del incremento > 7,
    L      #Inc_valor1;
    <I     ;
    S      #RET_VAL; //entonces activar RET_VAL y
    SPB    Final; //saltar al final
    L      -7; //Si el valor del incremento < -7,
    <I     ;
    S      #RET_VAL; //entonces activar RET_VAL y
    SPB    Final; //saltar al final
    U      L      1.3; //si el bit 4 del valor = 1 (Inc_valor neg)
    SPB    neg; //entonces saltar a sustracción de dirección de bit
    L      #Puntero1; //Cargar información de dirección del puntero
    L      #Inc_Wert1; //y sumar el incremento
    +D     ;
    SPA    test; //Saltar al test de resultado negativo
neg: L      #Puntero1; //Cargar información de dirección del puntero
    L      #Inc_valor1; //Cargar el incremento
    NEGI   ; //negar el valor negativo,
    -D     ; //restar el valor
    SPA    test; //y saltar al test

```

```

Byte: L      0; //Comienzo del cambio de dirección de byte
      L      #Inc_valor1; //Si el incremento >=0, entonces
      <I    ;
      SPB   pos; //saltar a la suma; en caso contrario
      L      #Puntero1; //Cargar información de dirección del puntero
      L      #Inc_valor1; //Cargar el incremento
      NEGI  ; //negar el valor negativo,
      SLD   3; //desplazar el incremento 3 posiciones hacia la izquierda,
      -D    ; //restar el valor
      SPA   test; //y saltar al test
pos:  SLD   3; //desplazar el incremento 3 posiciones hacia la izquierda,
      L      #Puntero1; //Cargar información de dirección del puntero
      +D    ; //sumar el incremento
test: T      #Val_int; //Transferir resultado cálculos a Val_int,
      U      L      7.3; //Si la dirección de byte no es válida (demasiado grande
      S      #RET_VAL; //o negativa), activar RET_VAL
      SPB   Final; //y saltar al final
      L      #Val_int; //en caso contrario, transferir el resultado
      T      #Puntero; //al puntero
Fin:  NOP   0;
END_FUNCTION

```

### A.3.4.5 Formato del tipo de parámetro ANY

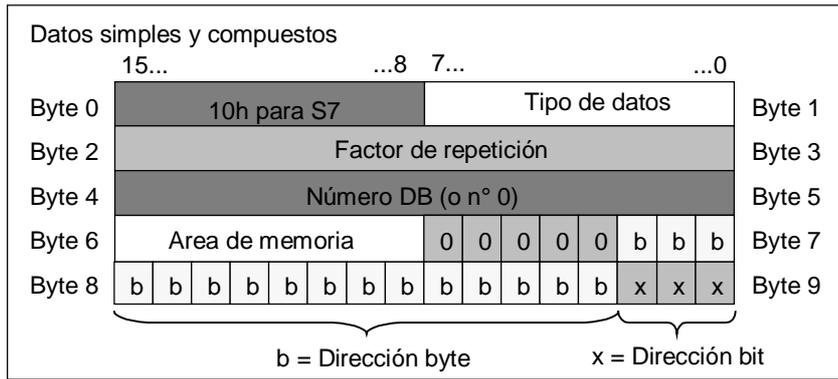
STEP 7 almacena los datos del tipo de parámetro ANY en 10 bytes. Al estructurar un parámetro del tipo ANY, los 10 bytes deben quedar ocupados, ya que el bloque llamado evalúa todo el contenido del parámetro. Si, por ejemplo, especifica un número de DB en el byte 4, deberá indicar también explícitamente el área de la memoria en el byte 6.

STEP 7 gestiona los datos de los tipos de datos simples y compuestos de forma diferente a la de los tipos de parámetros.

### Formato ANY en los tipos de datos

En el caso de los tipos de datos simples y compuestos, STEP 7 almacena los siguientes datos:

- Tipos de datos
- Factor de repetición
- Número de DB
- Área de la memoria donde se almacenan las informaciones
- Dirección inicial de los datos



El factor de repetición indica una cantidad del tipo de datos identificado que el tipo de parámetro ANY debe transferir. Así se puede indicar un área de datos, así como utilizar arrays y estructuras en combinación con el tipo de parámetro ANY. STEP 7 identifica, con ayuda del factor de repetición, los arrays y las estructuras como cantidad de tipos de datos. Si, por ejemplo, se deben transferir 10 palabras, en el factor de repetición se debe indicar el valor 10, y en el tipo de datos el valor 04.

La dirección se almacena en formato byte.bit. La dirección de byte se deposita en los bits 0 a 2 del byte 7, en los bits 0 a 7 del byte 8 y en los bits 3 a 7 del byte 9, mientras que la dirección de bit se deposita en los bits 0 a 2 del byte 9.

En el puntero nulo del tipo de datos NIL, todos los bytes estarán ocupados con 0 a partir del byte 1.

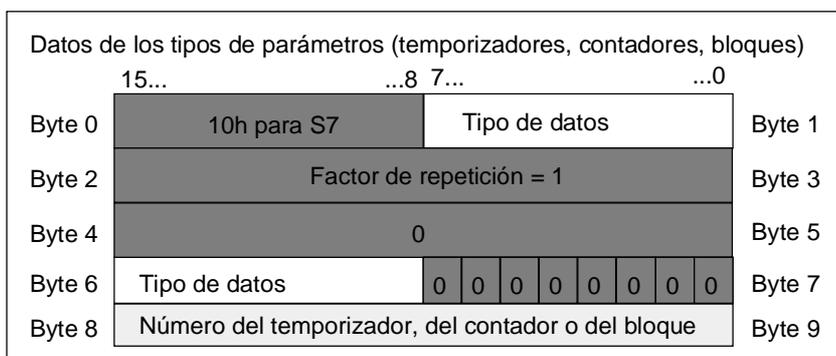
Las tablas siguientes muestran la codificación de los tipos de datos o las áreas de memoria para el tipo de parámetro ANY.

Codificación de los tipos de datos		
Código hexadecimal	Tipo de datos	Descripción
b#16#00	NIL	Puntero nulo
b#16#01	BOOL	Bits
b#16#02	BYTE	Bytes (8 bits)
b#16#03	Número entero	Carácter (8 bits)
b#16#04	WORD	Palabras (16 bits)
b#16#05	INT	Números enteros (16 bits)
B#16#06	DWORD	Palabras (32 bits)
b#16#07	DINT	Números enteros (32 bits)
b#16#08	REAL	Números reales (32 bits)
b#16#09	DATE	Fecha
b#16#0A	TIME_OF_DAY (TOD)	Hora
b#16#0B	TIME	Temporizador
b#16#0C	S5TIME	Tipo de datos S5TIME
b#16#0E	DATE_AND_TIME (DT)	Fecha y hora (64 bits)
b#16#13	STRING	Cadena de caracteres

Codificación de las áreas de memoria		
Código hexadecimal	Área	Descripción
b#16#81	E	Área de entradas
b#16#82	A	Área de salidas
b#16#83	M	Área de marcas
b#16#84	DB	Bloque de datos
b#16#85	DI	Bloque de datos de instancia
b#16#86	L	Pila de datos locales (L-Stack)
b#16#87	V	Datos locales precedentes

### Formato ANY en los tipos de parámetros

En el caso de los tipos de parámetros, STEP 7 almacena el tipo de datos y la dirección de los parámetros. El factor de repetición es siempre 1. Los bytes 4, 5 y 7 son siempre 0. Los bytes 8 y 9 indican el número del temporizador, del contador o del bloque.



La tabla siguiente muestra la codificación de los tipos de datos para el formato ANY en el caso de los tipos de parámetros.

Código hexadecimal	Tipo de datos	Descripción
b#16#17	BLOCK_FB	Número de FB
b#16#18	BLOCK_FC	Número de FC
b#16#19	BLOCK_DB	Número de DB
b#16#1A	BLOCK_SDB	Número de SDB
b#16#1C	COUNTER	Número de contador
b#16#1D	TIMER	Número de temporizador

### A.3.4.6 Uso del tipo de parámetro ANY

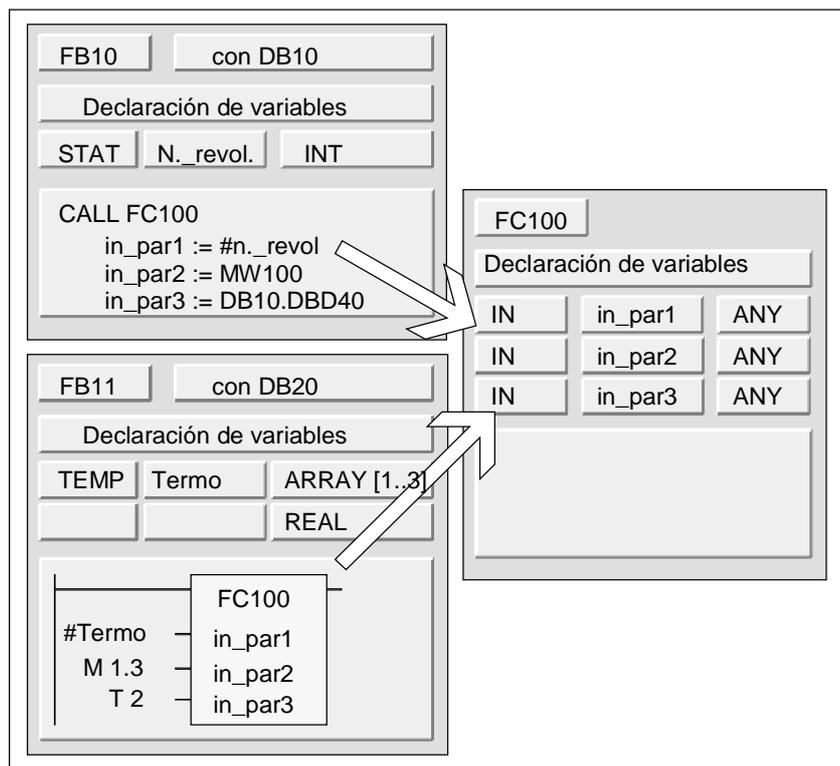
Para un bloque se pueden definir parámetros formales que sean apropiados para parámetros actuales con cualquier tipo de datos. Esto resulta sobre todo de gran utilidad cuando el tipo de datos del parámetro actual que se suministra al llamar el bloque es desconocido o puede variar (y cuando es admisible cualquier tipo de datos). En la declaración de variables del bloque se ha de declarar el parámetro con el tipo de datos ANY. En STEP 7 se puede asignar entonces un parámetro actual de un tipo de datos cualquiera.

STEP 7 asigna 80 bits de memoria a una variable con el tipo de datos ANY. Al asignar un parámetro actual a este parámetro formal, STEP 7 codifica la dirección de arranque, el tipo de datos y la longitud del parámetro actual en los 80 bits. El bloque llamado analiza los 80 bits en cuanto a datos que fueron memorizados para el parámetro ANY, obteniendo así informaciones que pueden utilizarse para el tratamiento ulterior.

#### Asignar un parámetro actual a un parámetro ANY

Si se declara un parámetro con el tipo de datos ANY, se podrá asignar a este parámetro formal un parámetro actual con un tipo de datos cualquiera. En STEP 7 se pueden asignar los siguientes tipos de datos como parámetros actuales:

- Tipos de datos simples: Se indica la dirección absoluta o el nombre simbólico del parámetro actual.
- Tipos de datos compuestos: Se indica el nombre simbólico de los datos con tipo de datos compuestos (p. ej., arrays y estructuras).
- Temporizadores, contadores y bloques: Introduzca el número (p. ej., T1, Z20 ó FB6).
- La figura siguiente muestra cómo se pueden transferir datos a una FC con parámetros del tipo de datos ANY.



En este ejemplo, FC100 tiene tres parámetros (*in\_par1*, *in\_par2* y *in\_par3*) que fueron declarados con el tipo de datos ANY.

- Cuando el FB10 llama la FC100, el FB10 entrega un número entero (la variable estática 'número de revoluciones'), una palabra (MW100) y una palabra doble en DB10 (DB10.DBD40).
- Cuando el FB11 llama la FC100, el FB11 entrega un campo de números reales (la variable temporal "termo"), un valor booleano (M 1.3) y un temporizador (T2).

### Indicar un área de datos para un parámetro ANY

A un parámetro ANY no sólo se le pueden asignar operandos individuales (p. ej. MW100), sino que también es posible indicar un área de datos. Si se desea asignar un área de datos como parámetro actual, entonces se debe utilizar el siguiente formato de una constante para indicar la cantidad de datos a transferir:

*p#      identif. de área Byte.Bit    tipo de datos    factor de repetición*

Para el elemento *tipo de datos* en el formato para constantes se pueden indicar todos los tipos de datos simples y el tipo de datos DATE\_AND\_TIME. Si el tipo de datos no es BOOL, habrá que indicar la dirección de bit 0 (x.0). La tabla siguiente muestra ejemplos de formato para indicar las áreas de memoria que se desean transferir a un parámetro ANY.

Parámetro actual	Descripción
p# M 50.0 BYTE 10	Indica 10 bytes en el área de memoria Marcas: MB50 a MB59.
p# DB10.DBX5.0 S5TIME 3	Indica 3 unidades de datos del tipo S5TIME, memorizadas en el DB10: DB byte 5 a DB byte 10.
p# A 10.0 BOOL 4	Indica 4 bits en el área de memoria Salidas: A 10.0 a A 10.3.

## Ejemplo de utilización del tipo de parámetro ANY

El siguiente ejemplo muestra cómo copiar un área de memoria de 10 bytes utilizando el tipo de parámetro ANY y la función de sistema SFC 20 BLKMOV.

AWL	Explicación
FUNCTION FC 10:VOID	
VAR_TEMP	
Fuente : ANY;	
Objetivo:	
END_VAR	
BEGIN	
LAR1 P#Quelle;	Cargar la dirección inicial del puntero ANY en AR1.
L B#16#10;	Cargar el ID de sintaxis y
T LB[AR1,P#0.0];	transferirlo al puntero ANY.
L B#16#02;	Cargar el tipo de datos byte y
T LB[AR1,P#1.0];	transferirlo al puntero ANY.
L 10;	Cargar 10 bytes y
T LW[AR1,P#2.0];	transferirlos al puntero ANY.
L 22;	La fuente es el DB22, DBB11
T LW[AR1,P#4.0];	
L P#DBX11.0;	
T LD[AR1,P#6.0];	
LAR1 P#Ziel;	Cargar la dirección inicial del puntero ANY en AR1.
L B#16#10;	Cargar el ID de sintaxis y
T LB[AR1,P#0.0];	transferirlo al puntero ANY.
L B#16#02;	Cargar el tipo de datos byte y
T LB[AR1,P#1.0];	transferirlo al puntero ANY.
L 10;	Cargar 10 bytes y
T LW[AR1,P#2.0];	transferirlos al puntero ANY.
L 33;	La meta es el DB33, DBB202
T LW[AR1,P#4.0];	
L P#DBX202.0;	
T LD[AR1,P#6.0];	
CALL SFC 20 (	Llamada de la función de sistema Blockmove
SRC_BLK := Quelle,	
RET_VAL := MW 12,	Evaluación del bit RB y de la MW 12
DSTBLK := Ziel	
);	
END FUNCTION	

### A.3.4.7 Asignar tipos de datos a los datos locales de bloques lógicos

STEP 7 limita los tipos de datos (tipos simples, compuestos y tipos de parámetros) que se pueden asignar en la declaración de variables a los datos locales de un bloque.

#### Tipos de datos válidos para los datos locales de un OB

La tabla siguiente muestra las restricciones existentes (–) al declarar los datos locales para un OB. Como un OB no puede ser llamado, entonces tampoco puede disponer de parámetros (entrada, salida o entrada/salida). Como que el OB no tiene DB de instancia, no se pueden declarar variables estáticas para un OB. Como tipos de datos de las variables temporales de un OB se pueden utilizar datos simples o compuestos, así como ANY.

Las asignaciones válidas se indican mediante el símbolo ●.

Tipo de declaración	Tipos de datos simples	Tipos de datos compuestos	Tipo de parámetro				
			TIMER	COUNTER	BLOCK	POINTER	ANY
Entrada	–	–	–	–	–	–	–
Salida	–	–	–	–	–	–	–
Entrada/salida	–	–	–	–	–	–	–
Estáticas	–	–	–	–	–	–	–
Temporales	● <sup>(1)</sup>	● <sup>(1)</sup>	–	–	–	–	● <sup>(1)</sup>

<sup>(1)</sup> Almacenado en la pila LStack del OB.

#### Tipos de datos válidos para los datos locales de un FB

La tabla siguiente muestra las restricciones existentes (–) al declarar los datos locales para un FB. Como que existe el DB de instancia, habrá menos restricciones al declarar los datos locales para un FB. Para la declaración de parámetros de entrada no existen restricciones; para un parámetro de salida no se pueden indicar ningún tipo de parámetro y para parámetros de entrada/salida sólo se admiten los tipos POINTER y ANY. Las variables temporales se pueden declarar con el tipo de datos ANY. Todos los demás tipos de parámetros son ilegales.

Las asignaciones válidas se indican mediante el símbolo ●.

Tipo de declaración	Tipos de datos simples	Tipos de datos compuestos	Tipo de parámetro				
			TIMER	COUNTER	BLOCK	POINTER	ANY
Entrada	●	●	●	●	●	●	●
Salida	●	●	–	–	–	–	–
Entrada/salida	●	● <sup>(1)(3)</sup>	–	–	–	●	●
Estáticas	●	●	–	–	–	–	–
Temporales	● <sup>(2)</sup>	● <sup>(2)</sup>	–	–	–	–	● <sup>(2)</sup>

<sup>(1)</sup> Como referencia (puntero de 48 bits) en el bloque de datos de instancia.  
<sup>(2)</sup> Almacenado en la pila LSTACK del FB.  
<sup>(3)</sup> Sólo pueden definirse STRINGS con la longitud estándar.

### Tipos de datos válidos para los datos locales de una FC

La tabla siguiente muestra las restricciones existentes (–) al declarar los datos locales para una FC. Como la FC no tiene DB de instancia, entonces tampoco dispone de variables estáticas. Para parámetros de entrada, salida y de entrada/salida de una FC, se admiten los tipos de parámetros POINTER y ANY. También es posible declarar variables temporales con el tipo de parámetro ANY.

Las asignaciones válidas se indican mediante el símbolo ●.

Tipo de declaración	Tipos de datos simples	Tipos de datos compuestos (2)	Tipo de parámetro				
			TIMER	COUNTER	BLOCK	POINTER	ANY
Entrada	●	●	●	●	●	●	●
Salida	●	●	—	—	—	●	●
Entrada/salida	●	●	—	—	—	●	●
Temporales	●(1)	●(1)	—	—	—	—	●(1)

<sup>1</sup> Almacenado en la pila LSTACK de la FC.  
<sup>(2)</sup> Sólo pueden definirse STRINGS con la longitud estándar.

### A.3.4.8 Tipos de datos admisibles al transferir parámetros

#### Reglas para la transferencia de parámetros entre bloques

Si asigna parámetros actuales a parámetros formales, puede indicar una dirección absoluta, un nombre simbólico o una constante STEP 7 limita las asignaciones válidas correspondientes para los diferentes parámetros. Por ejemplo, a los parámetros de salida y de entrada/salida no se les puede asignar un valor constante (ya que la finalidad de una salida o de una entrada/salida consiste en modificar el valor). Estas restricciones rigen sobre todo para parámetros con tipos de datos compuestos, a los cuales no se les puede asignar una dirección absoluta ni tampoco una constante.

Las tablas siguientes contienen las restricciones (–) de los tipos de datos de parámetros actuales que se asignan a los parámetros formales.

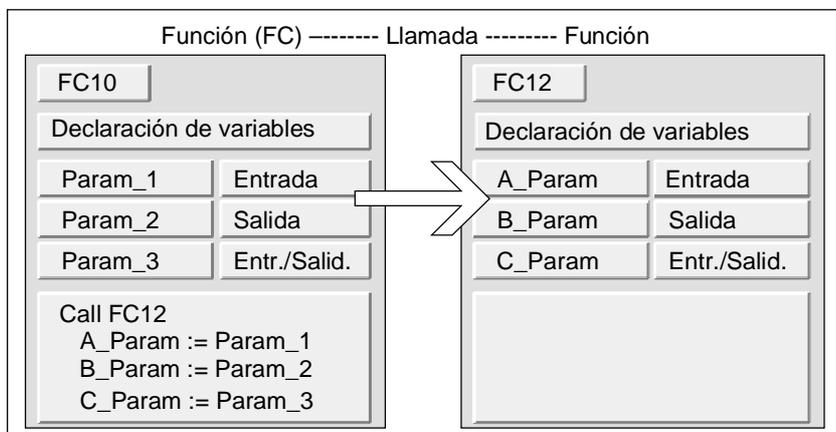
Las asignaciones válidas se indican mediante el símbolo ●.

Tipos de datos simples				
Tipo de declaración	Dirección absoluta	Nombre simbólico (en la tabla de símbolos)	Símbolo local del bloque	Constante
Entrada	●	●	●	●
Salida	●	●	●	—
Entrada/salida	●	●	●	—

Tipos de datos compuestos				
Tipo de declaración	Dirección absoluta	Nombre simbólico del elemento del DB (en la tabla de símbolos)	Símbolo local del bloque	Constante
Entrada	—	●	●	—
Salida	—	●	●	—
Entrada/salida	—	●	●	—

### Tipos de datos permitidos al llamar una FC desde otra FC

A los parámetros formales de una FC llamada se pueden asignar los parámetros formales de una FC invocante. La figura siguiente muestra los parámetros formales de FC10 que se asignan en calidad de parámetros actuales a los parámetros formales de FC12.



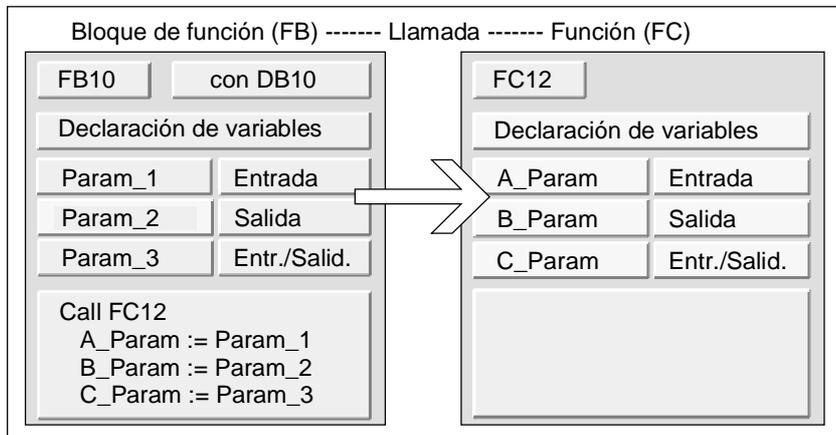
STEP 7 limita la asignación de parámetros formales - en calidad de parámetros actuales - de una FC a los parámetros formales de otra FC. Por ejemplo, no se pueden asignar como parámetros actuales aquellos parámetros con tipo de datos compuestos o tipo de parámetros.

La tabla siguiente muestra los tipos de datos admisibles (●) cuando una FC llama a otra.

Tipo de declaración	Tipos de datos simples	Tipos de datos compuestos	Tipo de parámetro				
			TIMER	COUNTER	BLOCK	POINTER	ANY
Entrada → Entrada	●	—	—	—	—	—	—
Entrada → Salida	—	—	—	—	—	—	—
Entrada → Entrada/salida	—	—	—	—	—	—	—
Salida → Entrada	—	—	—	—	—	—	—
Salida → Salida	●	—	—	—	—	—	—
Salida → Entrada/salida	—	—	—	—	—	—	—
Entrada/salida → Entrada	●	—	—	—	—	—	—
Entrada/salida → Salida	●	—	—	—	—	—	—
Entrada/salida → Entrada/salida	●	—	—	—	—	—	—

### Tipos de datos permitidos al llamar una FC desde un FB

A los parámetros formales de una FC invocante se pueden asignar los parámetros formales de un FB invocante. La figura siguiente muestra los parámetros formales de FB10 que se asignan en calidad de parámetros actuales a los parámetros formales de FC12.

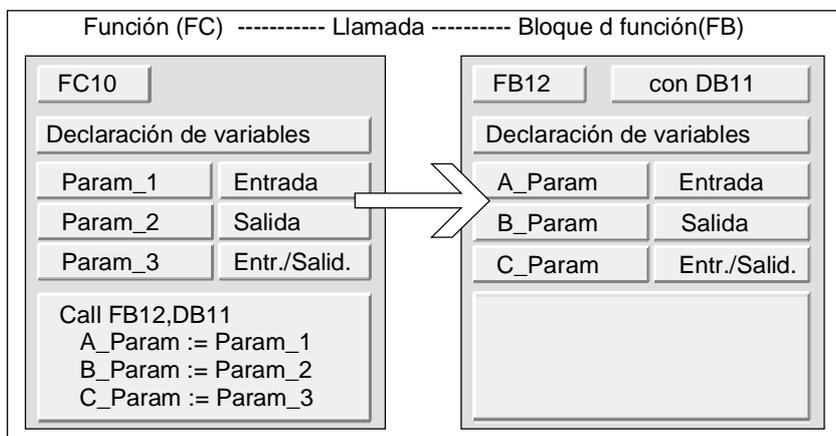


STEP 7 limita la asignación de parámetros formales de un FB a los parámetros formales de una FC. Por ejemplo, los parámetros con tipo de parámetro no se pueden asignar como parámetros actuales. La tabla siguiente muestra los tipos de datos admisibles (●) cuando un FB llama a una FC.

Tipo de declaración	Tipos de datos simples	Tipos de datos compuestos	Tipo de parámetro				
			TIMER	COUNTER	BLOCK	POINTER	ANY
Entrada → Entrada	●	●	—	—	—	—	—
Entrada → Salida	—	—	—	—	—	—	—
Entrada → Entrada/salida	—	—	—	—	—	—	—
Salida → Entrada	—	—	—	—	—	—	—
Salida → Salida	●	●	—	—	—	—	—
Salida → Entrada/salida	—	—	—	—	—	—	—
Entrada/salida → Entrada	●	—	—	—	—	—	—
Entrada/salida → Salida	●	—	—	—	—	—	—
Entrada/salida → Entrada/salida	●	—	—	—	—	—	—

### Tipos de datos permitidos al llamar un FB desde una FC

A los parámetros formales de un FB llamado se pueden asignar los parámetros formales de una FC invocante. La figura siguiente muestra los parámetros formales de FC10 que se asignan en calidad de parámetros actuales a los parámetros formales de FB12.



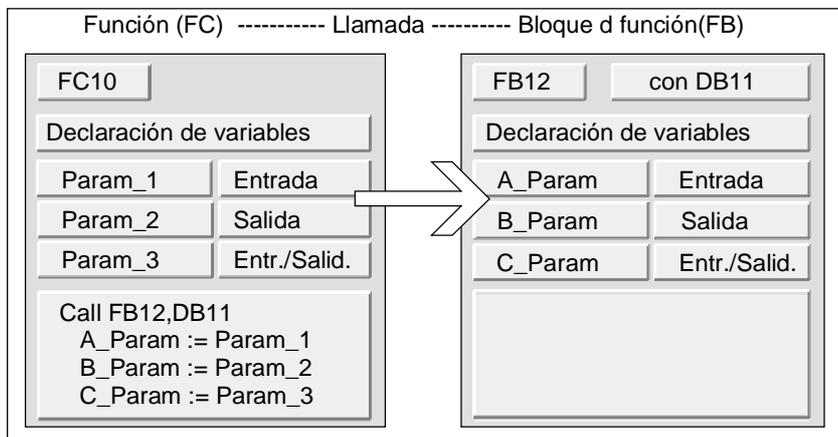
STEP 7 limita la asignación de parámetros formales de una FC a los parámetros formales de un FB. Por ejemplo, los parámetros con tipo de datos compuestos no se pueden asignar como parámetros actuales. Sin embargo, los parámetros de entrada con el tipo de parámetro TIMER, COUNTER o BLOCK se pueden asignar a los parámetros de entrada del FB invocado.

La tabla siguiente muestra los tipos de datos admisibles (●) cuando una FC llama a un FB.

Tipo de declaración	Tipos de datos simples	Tipos de datos compuestos	Tipo de parámetro				
			TIMER	COUNTER	BLOCK	POINTER	ANY
Entrada → Entrada	●	—	●	●	●	—	—
Entrada → Salida	—	—	—	—	—	—	—
Entrada → Entrada/salida	—	—	—	—	—	—	—
Salida → Entrada	—	—	—	—	—	—	—
Salida → Salida	●	—	—	—	—	—	—
Salida → Entrada/salida	—	—	—	—	—	—	—
Entrada/salida → Entrada	●	—	—	—	—	—	—
Entrada/salida → Salida	●	—	—	—	—	—	—
Entrada/salida → Entrada/salida	●	—	—	—	—	—	—

### Tipos de datos permitidos al llamar un FB desde otro FB

A los parámetros formales de un FB llamado se pueden asignar los parámetros formales de un FB invocante. La figura siguiente muestra los parámetros formales de FB10 que se asignan en calidad de parámetros actuales a los parámetros formales de FB12.



STEP 7 limita la asignación de parámetros formales de un FB a los parámetros formales de otro FB. Por ejemplo, los parámetros de entrada y de salida con tipos de datos compuestos no se pueden asignar como parámetros actuales a los parámetros de entrada y de salida de un FB llamado. Sin embargo, los parámetros de entrada con el tipo de parámetro TIMER, COUNTER o BLOCK se pueden asignar a los parámetros de entrada del FB llamado.

La tabla siguiente muestra los tipos de datos admisibles (●) cuando un FB llama a otro.

Tipo de declaración	Tipos de datos simples	Tipos de datos compuestos	Tipo de parámetro TIMER	Tipo de parámetro COUNTER	Tipo de parámetro BLOCK	Tipo de parámetro POINTER	Tipo de parámetro ANY
Entrada → Entrada	●	●	●	●	●	—	—
Entrada → Salida	—	—	—	—	—	—	—
Entrada → Entrada/salida	—	—	—	—	—	—	—
Salida → Entrada	—	—	—	—	—	—	—
Salida → Salida	●	●	—	—	—	—	—
Salida → Entrada/salida	—	—	—	—	—	—	—
Entrada/salida → Entrada	●	—	—	—	—	—	—
Entrada/salida → Salida	●	—	—	—	—	—	—
Entrada/salida → Entrada/salida	●	—	—	—	—	—	—

### **A.3.4.9 Transferencia a parámetros IN\_OUT de un FB**

El transferir tipos de datos compuestos a parámetros IN\_OUT de un bloque de función (FB) se transfiere solamente la dirección del operando de la variable (call by reference).

Al transferir tipos de datos simples a parámetros IN\_OUT de un FB se copian los valores en el bloque de datos de instancia antes de arrancar el FB, y cuando se termina el FB se vuelven a copiar del bloque de datos de instancia.

Así se pueden inicializar con un valor las variables IN\_OUT de un tipo de datos simple.

No obstante, en una llamada no se puede indicar una constante como parámetro actual en la posición de una variable IN\_OUT, ya que no se puede escribir en una constante.

Las variables del tipo de datos STRUCT o ARRAY no se pueden inicializar, ya que en este caso hay una sola dirección en el bloque de datos de instancia.

## A.4 Trabajar con proyectos de versiones anteriores

### A.4.1 Convertir proyectos de la versión 1

Convertiendo los proyectos de la versión 1 en proyectos de la versión 2 podrá reutilizar proyectos que se hayan creado con la versión 1 de STEP 7.

Tras la conversión se conservan los siguientes componentes del proyecto de la versión 1:

- Estructura del proyecto y sus programas
- Bloques
- Archivos fuente AWL
- Tabla de símbolos

La configuración del hardware no se convierte. Los componentes de programa pueden copiarse a otros proyectos. Además, es posible añadir al nuevo proyecto un equipo que deberá ser configurado y parametrizado debidamente.

Tras realizar la conversión a la versión 2 podrá decidir en un cuadro de diálogo si el proyecto de la versión 2 se ha de convertir en un proyecto de la versión que Ud. dispone de STEP 7.

---

#### Nota

Las propiedades de los bloques siguen siendo las correspondientes a la versión 1. El código generado en la versión 1 no cambia, por lo que los bloques no pueden utilizarse junto con multiinstancias.

Si desea declarar multiinstancias en los bloques convertidos, utilizando la aplicación "KOP/AWL: programar bloques" genere fuentes AWL con los bloques convertidos y compílelas de nuevo en bloques.

Una nueva opción de la versión 2 de STEP 7 es la programación de multiinstancias para crear bloques de función (FBs). Si desea volver a utilizar los FBs creados con la versión 1 para las mismas funciones en proyectos de la versión 2, no necesita convertirlos.

---

### Procedimiento

Para convertir proyectos de la versión 1:

1. Elija el comando Archivo > Abrir proyecto versión 1.
2. En el cuadro de diálogo que aparece a continuación, elija el proyecto de la versión 1 que desea reutilizar. Los proyectos de la versión 1 tienen la extensión \*.s7a.
3. En el cuadro de diálogo siguiente, introduzca el nombre del nuevo proyecto en el que se ha de convertir el proyecto de la versión 1.

## A.4.2 Convertir proyectos de la versión 2

Con el comando **Archivo > Abrir** es posible abrir proyectos creados con la versión 2 de STEP 7.

Los proyectos y las librerías de la versión 2 se pueden convertir a la versión actual de STEP 7 (migración) utilizando el comando de menú **Archivo > Guardar como** y el ajuste "reorganizando". Así, el proyecto se guarda como proyecto de la actual versión STEP 7.

Los proyectos y las librerías que se hayan creado con versiones antiguas de STEP 7 se pueden editar y guardar conservando el formato. Para ello se debe seleccionar en el cuadro de diálogo "Guardar proyecto como" el tipo de datos de la correspondiente versión antigua de STEP 7. Para elaborar objetos p. ej. con la versión 2.1 de STEP 7, elija aquí Proyecto 2.x o Librería 2.x (guardar como versión 2 ya no es posible a partir de la versión 5.1, consulte también Editar proyectos y librerías de la versión 2).

### Tipos de archivo

	STEP 7 V3	a partir de STEP 7 V4
Tipo de archivo de la versión actual	Proyecto3.x Librería3.x	Proyecto Librería
Tipo de archivo de la versión antigua	Proyecto2.x Librería2.x	Proyecto2.x Librería2.x

No obstante, en ese caso dispondrá sólo de las funciones de la versión antigua de STEP 7. Sin embargo, los proyectos y las librerías se podrán seguir manteniendo con la versión antigua de STEP 7.

### Nota

Al pasar de la versión 3 a la versión 4 u otra posterior se cambia solamente el nombre del tipo de archivo, el formato no se modifica. Por esta razón en STEP 7 V4 no existe ningún tipo de archivo del tipo Proyecto3.x.

### Procedimiento

Para convertir los proyectos de la versión 2 al formato de la versión actual de STEP 7:

1. Ejecute la función "Guardar como" reorganizando (Menú Archivo).
2. Elija en el cuadro de diálogo "Guardar proyecto como" el tipo de archivo "Proyecto" y active el botón de comando "Guardar".

Para convertir proyectos de la versión 2 a la versión actual de STEP 7 (conservando el formato original):

1. En caso necesario efectúe el paso 1 anteriormente descrito.
2. Elija en el cuadro de diálogo "Guardar proyecto como" el tipo de archivo de la antigua versión de STEP 7 y pulse el botón de comando "Guardar".

### A.4.3 Observaciones sobre los proyectos de STEP 7 V2.1 con comunicación de datos globales (GD)

- Si desea convertir un proyecto con datos globales de STEP 7 V2.1 a STEP 7 V5 deberá abrir previamente la tabla GD con STEP 7 V5.0 en el proyecto de STEP 7 V2.1. Todos los datos de comunicación configurados hasta entonces se convertirán automáticamente a la nueva estructura por comunicación GD.
- Al archivar proyectos de STEP 7 V2.1 puede suceder que los antiguos programas de compresión (ARJ, PKZIP...) generen un mensaje de error si el proyecto contiene archivos cuyos nombres excedan los 8 caracteres. Este mensaje se visualizará también si la red MPI se ha editado en el proyecto de STEP 7 V2.1 con una denominación que comprenda más de 8 caracteres. En los proyectos de STEP 7 V2.1 con datos globales, utilice un nombre para la red MPI que no exceda los 8 caracteres antes de arrancar por primera vez la configuración de la comunicación de datos globales.
- Si desea cambiar el nombre de un proyecto de STEP 7 V2.1 deberá asignar nuevamente los títulos de las columnas (CPUs) en la tabla GD, seleccionando de nuevo la CPU en cuestión. Cuando restablezca el nombre antiguo del proyecto, las asignaciones existirán nuevamente.

### A.4.4 Ampliar esclavos DP creados con versiones anteriores de STEP 7

#### Configuraciones que pueden crearse instalando nuevos archivos GSD

Instalando nuevos archivos GSD en el catálogo de hardware de HW-Config. pueden incorporarse nuevos esclavos DP. Después de la instalación están disponibles en la carpeta "Otros aparatos de campo".

Un esclavo DP modular no puede ampliarse (ni modificarse su configuración) de la forma habitual con STEP 7, Servicepack 3 en los casos siguientes:

- si ha sido configurado con una versión anterior de STEP 7 o
- si en el catálogo de hardware no estaba representado por un archivo GSD, sino por un archivo de tipo, o bien
- si se ha instalado un archivo GSD nuevo

#### Remedio

Si desea utilizar el esclavo DP con **nuevos módulos** descritos en el archivo GSD:

- borre el esclavo DP y reconfigúrelo; el esclavo DP no se describirá mediante el archivo de tipo, sino que quedará completamente definido por el archivo GSD.

Si no desea utilizar **módulos nuevos**, descritos únicamente en el archivo GSD:

- seleccione la carpeta "Otros aparatos de campo/Esclavos compatibles Profibus-DP", dentro de PROFIBUS-DP en la ventana "Catálogo de hardware". STEP 7 desplaza a dicha carpeta los archivos de tipo "antiguos" cuando son reemplazados por archivos GSD nuevos. En dicha carpeta encontrará los módulos con los que podrá ampliar el esclavo DP configurado.

#### A.4.5 Esclavos DP con archivos GSD defectuosos o inexistentes

Si edita configuraciones de equipos antiguos con la versión 5.1 de STEP 7, puede ocurrir en algunos casos que falte o no se pueda compilar el archivo GSD de un esclavo DP (p. ej., debido a errores de sintaxis en el archivo GSD).

En este caso STEP 7 genera un esclavo "Dummy" que representa al esclavo configurado, p. ej., después de cargar un equipo en PG o de abrir y continuar editando un proyecto antiguo. Este esclavo "Dummy" sólo se puede editar de forma muy limitada; no podrá modificar la estructura del esclavo (identificación DP) ni los parámetros. Pero se puede cargar de nuevo en el equipo; la configuración original del esclavo permanecerá intacta. También se puede borrar completamente el esclavo DP.

#### Modificar la configuración y parametrización del esclavo DP

Si desea modificar la configuración o parametrización del esclavo DP deberá solicitar del fabricante un archivo GSD actualizado para este esclavo DP y ponerlo a disposición a través del comando de menú **Herramientas > Instalar nuevo GSD**.

Después de instalar el archivo GSD correcto se utilizará para la representación del esclavo DP. El esclavo DP conserva sus datos y se puede editar de nuevo completamente.

## A.5 Programas de ejemplo

### A.5.1 Proyectos y programas de ejemplo

El CD de instalación contiene diversos proyectos de ejemplo. Los proyectos no descritos en el presente capítulo se describen en el OB 1 en cuestión.

Programas y proyectos de ejemplo	Contenidos en el CD	Descripción en este capítulo	Descripción IEC/internacional el OB 1
Proyecto "ZEs01_08_STEP7_Mezcla" (proceso de mezcla industrial)	•	•	
Proyectos "ZEs01_01_STEP7_*" a "ZEs01_06_STEP7_*" (Introducción y ejercicios prácticos)	•	Manual	•
Proyecto "ZEs01_09_STEP7_CEBRA" (control de semáforo en un paso de peatones)	•		•
Proyecto "ZEs01_10_STEP7_COM_SFB" (intercambio de datos entre dos CPUs S7-400)	•		•
Proyectos "ZEs01_11_STEP7_COM_SFC1" y "ZEs01_12_STEP7_COM_SFC2" (intercambio de datos a través de SFCs de comunicación para enlaces no configurados)	•		•
Ejemplo de manipulación de alarmas horarias		•	
Ejemplo de manipulación de alarmas de retardo		•	
Ejemplo de enmascarar y desenmascarar eventos de errores síncronos		•	
Ejemplo de bloquear y habilitar eventos de alarma y de error asíncrono		•	
Ejemplo de procesamiento retardado de eventos de alarma y asíncronos		•	

El objetivo principal de los ejemplos no reside en proporcionar un determinado estilo de programación ni tampoco los conocimientos técnicos requeridos para controlar un determinado proceso. Estos ejemplos persiguen, más bien, reproducir los pasos requeridos para diseñar un programa.

### Borrar e instalar los proyectos de ejemplo adjuntos

Los proyectos de ejemplo adjuntos se pueden borrar e instalar de nuevo en el Administrador SIMATIC. Para instalarlos se deberá arrancar el programa Setup de STEP 7 V5.0. Los proyectos de ejemplo se pueden instalar posteriormente de forma selectiva.

#### Nota

A no ser que se ordene lo contrario, los proyectos de ejemplo adjuntos se copiarán automáticamente durante la instalación de STEP 7. Si ha modificado ya dichos proyectos, éstos se sobrescribirán con los originales al instalarse nuevamente STEP 7.

Por tanto, antes de efectuar cambios es recomendable copiar los proyectos de ejemplo adjuntos y editar sólo las copias.

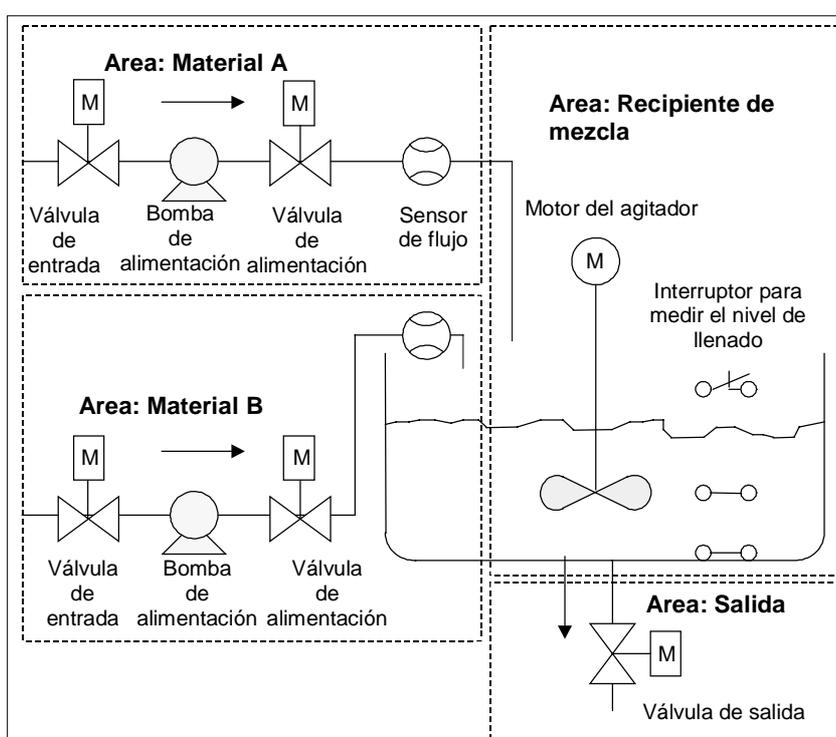
## A.5.2 Proyectos y programas de ejemplo

### A.5.2.1 Programa de ejemplo para un proceso de mezcla industrial

El programa de ejemplo se basa en informaciones suministradas en la 1ª parte del manual, en la que se describía el control de un proceso de mezcla industrial.

#### Tarea

Dos materiales (A y B) se han de mezclar en un recipiente con ayuda de un agitador. La masa se descarga del recipiente de mezcla mediante una válvula de salida. La figura siguiente muestra un diagrama del proceso de ejemplo.



## Descripción de las partes del proceso

En la 1ª parte del manual se describió cómo subdividir el proceso de ejemplo en áreas funcionales y tareas individuales. A continuación se describen las diferentes áreas.

### *Area Materiales A y B:*

- Las tuberías de entrada de material deben tener una válvula de entrada, una válvula de alimentación y una bomba de alimentación.
- Las tuberías de entrada tienen instalado un sensor de flujo.
- La conexión de las bombas de alimentación debe estar bloqueada cuando el medidor de nivel de flujo indica "recipiente lleno".
- La conexión de las bombas de alimentación debe estar bloqueada cuando la válvula de salida está abierta.
- Las válvulas de entrada y de alimentación no se pueden abrir antes de que haya transcurrido 1 segundo tras la activación de la bomba de alimentación.
- Las válvulas se deben cerrar inmediatamente tras parar las bombas de alimentación (señal del sensor de flujo) para impedir que el material salga de la bomba.
- La activación de las bombas de alimentación es temporizada, es decir, el sensor debe señalar la existencia del flujo dentro de los 7 segundos subsiguientes a la activación.
- Las bombas de alimentación se han de desconectar lo más rápido posible en el caso de que los sensores de flujo no señalicen ningún flujo durante la marcha de las bombas.
- La cantidad de arranques de las bombas de alimentación debe ser contada (define el intervalo de mantenimiento).

### *Area Recipiente de mezcla:*

- La conexión del motor del agitador debe estar enclavada cuando el medidor de nivel de llenado señala "recipiente bajo el mínimo" o está abierta la válvula de salida.
- El motor del agitador emite una señal de respuesta (señalización en respuesta) tras alcanzar el número de revoluciones nominal. Si esta señal no se emite dentro de los 10 segundos subsiguientes al arranque del motor, entonces se ha de desconectar dicho motor.
- La cantidad de arranques del motor del agitador debe contarse (intervalo de mantenimiento).
- En el recipiente de mezcla se han de instalar tres sensores:
  - recipiente lleno: Contacto normalmente cerrado. Si se alcanza el nivel máximo de llenado, se abre el contacto.
  - nivel de llenado en el recipiente sobre el mínimo: Contacto normalmente abierto. Si se alcanza el nivel mínimo de llenado, se cierra el contacto..
  - recipiente no vacío: Contacto normalmente abierto. Si el recipiente no está vacío, el contacto está cerrado.

### *Area Salida:*

- La salida debe controlarse a través de una válvula de solenoide.
- La válvula de solenoide es controlada por el operador, pero, en cualquier caso siempre se cierra, lo más tardar, al aparecer la señal "recipiente vacío".
- La apertura de la válvula de salida está enclavada cuando
  - el motor del agitador está en marcha
  - el recipiente está vacío

## Panel de manejo

Para que el operador pueda arrancar, parar y observar el proceso, se ha de prever un panel de manejo. El panel de manejo contiene

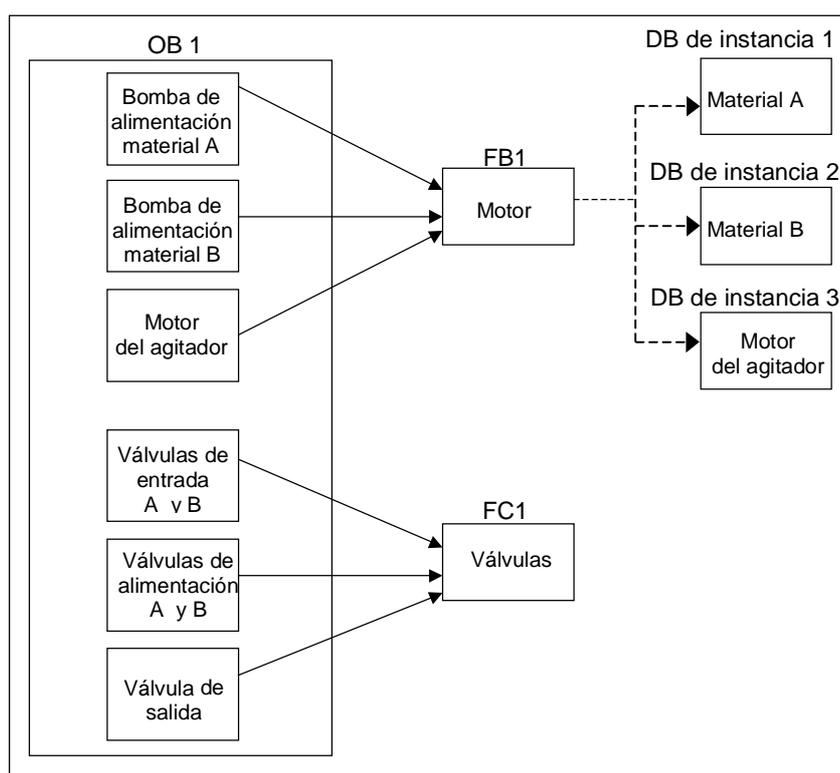
- pulsadores para controlar las operaciones principales Con el pulsador "Desactivar lámpara de señalización" se apagan las lámparas de señalización de mantenimiento del motor afectado y se pone a 0 el contador que controla los intervalos de mantenimiento.
- lámparas de señalización para leer el estado operativo
- el pulsador de PARO DE EMERGENCIA

### A.5.2.2 Definir bloques lógicos

La estructura del programa se define subdividiendo el programa de usuario en diferentes bloques y jerarquizando las llamadas de bloques.

#### Jerarquía de las llamadas de bloques

La figura siguiente representa la jerarquía de los bloques que se pueden llamar en el programa estructurado.



- OB1: Representa el interface con el sistema operativo de la CPU y contiene el programa principal. En el OB 1 se llaman los bloques FB 1 y FC 1 y se transfieren los parámetros específicos necesarios para controlar el proceso.
- FB 1: Un sólo bloque de función puede controlar la bomba de alimentación para el material A, la bomba de alimentación para el material B y el motor del agitado, puesto que las exigencias (conectar, desconectar, contar las utilizaciones, etc.) son idénticas.
- DBs de instancia 1-3: Los parámetros actuales y los datos estáticos para el control del material A, del material B y del motor del agitador son diferentes, por lo que se depositan en tres DBs de instancia asociados al FB 1.
- FC 1: las válvulas de entrada y de alimentación para los materiales A y B y para la válvula de salida utilizan también un bloque lógico común. Como sólo se deben programar las funciones "Abrir" y "Cerrar", basta con utilizar una sola FC.

### A.5.2.3 Asignar nombres simbólicos

#### Definir nombres simbólicos

En el programa ejemplificado se utilizan símbolos que se deben definir con STEP 7 en la tabla de símbolos. Las tablas siguientes muestran los nombres simbólicos y las direcciones absolutas de los elementos utilizados en el programa.

#### Direcciones simbólicas de la bomba de alimentación, del motor del agitador y de las válvulas de entrada:

Nombre simbólico	Dirección	Tipo de datos	Descripción
Feed_pump_A_start	E 0.0	BOOL	Pulsador de arranque de la bomba de alimentación del material A
Feed_pump_A_stop	E 0.1	BOOL	Pulsador de parada de la bomba de alimentación del material A
Flow_A	E 0.2	BOOL	Material A fluye
Inlet_valve_A	A 4.0	BOOL	Activación de la válvula de entrada para material A
Feed_valve_A	A 4.1	BOOL	Activación de la bomba de alimentación para material A
Feed_pump_A_on	A 4.2	BOOL	Lámpara de señalización "Bomba alimentación material A en marcha"
Feed_pump_A_off	A 4.3	BOOL	Lámpara de señalización "Bomba alimentación material A no en marcha"
Feed_pump_A	A 4.4	BOOL	Activación de la bomba de alimentación para material A
Feed_pump_A_fault	A 4.5	BOOL	Lámpara de señalización "Error de la bomba de alimentación A"
Feed_pump_A_maint	A 4.6	BOOL	Lámpara de señalización "Mantenimiento bomba aliment. A"
Feed_pump_B_start	E 0.3	BOOL	Pulsador de arranque de la bomba de alimentación del material B

Nombre simbólico	Dirección	Tipo de datos	Descripción
Feed_pump_B_stop	E 0.4	BOOL	Pulsador de parada de la bomba de alimentación del material B
Flow_B	E 0.5	BOOL	Material B fluye
Inlet_valve_B	A 5.0	BOOL	Activación de la válvula de entrada para material B
Feed_valve_B	A 5.1	BOOL	Activación de la bomba de alimentación para material B
Feed_pump_B_on	A 5.2	BOOL	Lámpara de señalización "Bomba alimentación material B en marcha"
Feed_pump_B_off	A 5.3	BOOL	Lámpara de señalización "Bomba alimentación material B no en marcha"
Feed_pump_B	A 5.4	BOOL	Activación de la bomba de alimentación para material B
Feed_pump_B_fault	A 5.5	BOOL	Lámpara de señalización "Error de la bomba de alimentación B"
Feed_pump_B_maint	A 5.6	BOOL	Lámpara de señalización "Mantenimiento bomba aliment. B"
Agitator_running	E 1.0	BOOL	Señal de respuesta del motor del agitador
Agitator_start	E 1.1	BOOL	Pulsador de arranque del agitador
Agitator_stop	E 1.2	BOOL	Pulsador de parada del agitador
Agitator	A 8.0	BOOL	Activación del agitador
Agitator_on	A 8.1	BOOL	Lámpara de señalización "Agitador ON"
Agitator_off	A 8.2	BOOL	Lámpara de señalización "Agitador OFF"
Agitator_fault	A 8.3	BOOL	Lámpara de señalización "Fallo del motor del agitador"
Agitator_maint	A 8.4	BOOL	Lámpara de señalización "Mantenimiento del agitador"

#### **Direcciones simbólicas de sensores e indicador de nivel del recipiente:**

Nombre simbólico	Dirección	Tipo de datos	Descripción
Tank_below_max	E 1.3	BOOL	Sensor "Recipiente de mezcla no está lleno"
Tank_above_min	E 1.4	BOOL	Sensor "Recipiente de mezcla sobre el mínimo"
Tank_not_empty	E 1.5	BOOL	Sensor "Recipiente de mezcla no está vacío"
Tank_max_disp	A 9.0	BOOL	Lámpara de señalización "Recipiente de mezcla lleno"
Tank_min_disp	A 9.1	BOOL	Lámpara de señalización "Recipiente de mezcla bajo mínimo"
Tank_empty_disp	A 9.2	BOOL	Lámpara de señalización "Recipiente de mezcla vacío"

**Direcciones simbólicas de la válvula de salida**

Nombre simbólico	Dirección	Tipo de datos	Descripción
Drain_open	E 0.6	BOOL	Pulsador para abrir la válvula de salida
Drain_closed	E 0.7	BOOL	Pulsador para cerrar la válvula de salida
Drain	A 9.5	BOOL	Activación de la válvula de salida
Drain_open_disp	A 9.6	BOOL	Lámpara de visualización "Válvula de salida abierta"
Drain_closed_disp	A 9.7	BOOL	Lámpara de visualización "Válvula de salida cerrada"

**Direcciones simbólicas de los demás elementos del programa**

Nombre simbólico	Dirección	Tipo de datos	Descripción
EMER_STOP_off	E 1.6	BOOL	Pulsador de PARO DE EMERGENCIA
Reset_maint	E 1.7	BOOL	Tecla de reset de las lámparas de señalización de mantenimiento de todos los motores
Motor_block	FB 1	FB 1	FB para controlar las bombas y el motor
Valve_block	FC 1	FC 1	FC para controlar las válvulas
DB_feed_pump_A	DB 1	FB 1	DB de instancia para controlar la bomba de alimentación A
DB_feed_pump_B	DB 2	FB 1	DB de instancia para controlar la bomba de alimentación B
DB_agitator	DB 3	FB 1	DB de instancia para controlar el motor del agitador

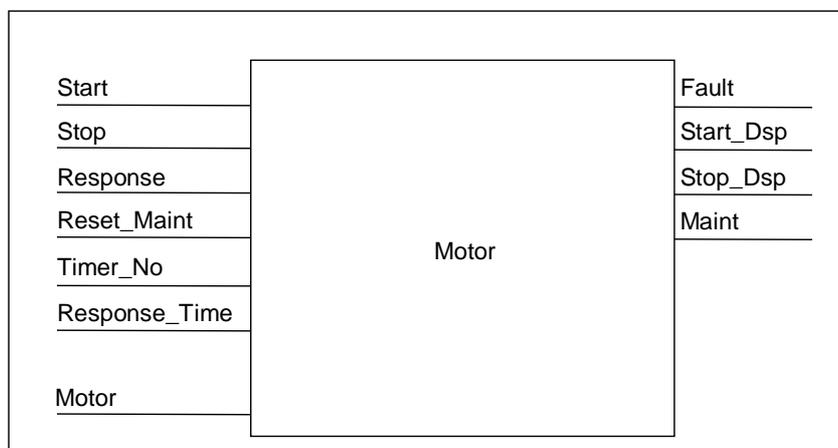
**A.5.2.4 Crear el FB para el motor****Lógica requerida en el FB**

El FB para el motor contiene las siguientes funciones lógicas:

- Existe una entrada para arranque y una para parada.
- Una serie de enclavamientos permite el servicio de los equipos (bombas y motor de agitador). El estado de los enclavamientos se memoriza en los datos locales temporales (pila L) del OB 1 (p.ej., "Habilitación\_motor") y se combinan lógicamente con las entradas de arranque y de parada al ejecutarse el FB para el motor.
- La respuesta de los equipos debe recibirse dentro de un determinado tiempo. En caso contrario se supone que se ha presentado un error. En tal caso, el FB detiene el motor.
- El temporizador y el intervalo para el ciclo de respuesta o de fallo deben ser definidos.
- Tras oprimir el pulsador de arranque (Marcha) y concederse la habilitación, se activa el equipo y continúa operando hasta que se oprima el pulsador de parada (Paro).
- Conectado el equipo, el temporizador comienza a correr. Si la señal de respuesta del aparato no se recibe antes de transcurrir el temporizador, entonces se detiene la marcha del equipo.

## Determinar las entradas y las salidas

La figura siguiente muestra las entradas y las salidas del FB general para el motor.



## Definir los parámetros para el FB

Si se desea crear un FB apto para todos los motores (para controlar ambas bombas y el motor), se han de definir nombres de parámetros generales para las entradas y las salidas.

El FB para el motor del proceso ejemplificado tiene los siguientes requerimientos:

- Necesita señales del panel de operador para parar o arrancar el motor y las bombas.
- Necesita una señal de respuesta del motor y de las bombas de que el motor está en marcha.
- Tiene que calcular el tiempo transcurrido entre la transmisión de la señal de marcha del motor y la recepción de la señal de respuesta. Si la señal de respuesta no se recibe durante un tiempo determinado, se debe desconectar el motor.
- Debe conectar o desconectar las lámparas de señalización correspondientes en el panel de operador.
- Transmite una señal para controlar el motor.

Estos requerimientos se pueden definir como entradas y salidas del FB. La tabla siguiente muestra los parámetros del FB para el motor del proceso de ejemplo.

Nombre del parámetro	Entrada	Salida	Entrada/salida
Start	n		
Stop	n		
Response	n		
Reset_Maint	n		
Timer_No	n		
Response_Time	n		
Fault		n	
Start_Dsp		n	
Stop_Dsp		n	
Maint		n	
Motor			n

## Declaración de variables del FB para el motor

Es preciso declarar los parámetros de entrada, de salida y de entrada/salida del FB para el motor.

Dirección	Declaración	Designación	Tipo	Valor inicial
0.0	IN	Arranque	BOOL	FALSE
0.1	IN	Stop	BOOL	FALSE
0.2	IN	Response	BOOL	FALSE
0.3	IN	Reset_Maint	BOOL	FALSE
2.0	IN	Time_No	TIMER	
4.0	IN	Response_Time	S5TIME	S5T#0MS
6.0	OUT	Fault	BOOL	FALSE
6.1	OUT	Start_Dsp	BOOL	FALSE
6.2	OUT	Stop_Dsp	BOOL	FALSE
6.3	OUT	Maint	BOOL	FALSE
8.0	IN_OUT	Motor	BOOL	FALSE
10.0	STAT	Time_bin	WORD	W#16#0
12.0	STAT	Time_BCD	WORD	W#16#0
14.0	STAT	Starts	INT	0
16.0	STAT	Start_Edge	BOOL	FALSE

En los FBs, las variables de entradas, salidas, entrada/salida y variables estáticas se memorizan en el DB de instancia indicado en la instrucción de llamada. Las variables temporales se memorizan en la pila L.

## Programar el FB para el motor

En STEP 7, cada bloque que es llamado por otro bloque debe ser creado antes del bloque que contiene la llamada. Por lo tanto, en el programa de ejemplo deberá crear el FB para el motor antes que el OB 1.

El área de instrucciones del FB 1 tiene la siguiente estructura en el lenguaje de programación AWL:

### Segmento 1 Marcha/parada y autorretención

```

U(
  O    #Start
  O    #Motor
)
UN    #Stop
=     #Motor

```

**Segmento 2 Vigilancia del arranque**

```

U #Motor
L   #Response_Time
SE  #Timer_No
UN  #Motor
R   #Timer_No
L   #Timer_No
T   #Timer_bin
LC  #Timer_No
T   #Timer_BCD
U   #Timer_No
UN  #Response
S   #Fault
R   #Motor

```

**Segmento 3 Lámpara Marcha y Reset Error**

```

U #Response
=  #Start_Dsp
R  #Fault

```

**Segmento 4 Lámpara Parada**

```

UN#Response
=  #Stop_Dsp

```

**Segmento 5 Contaje de los arranques**

```

U #Motor
FP  #Start_Edge
SPBN lab1
L   #Starts
+   1
T   #Starts

```

```
lab1: NOP 0
```

**Segmento 6 Lámpara de señalización de mantenimiento**

```

L #Starts
L   50
>=I
=  #Maint

```

**Segmento 7 Reset del contador del número de arranques**

```

U #Reset_Maint
U  #Maint
SPBN END
L   0
T   #Starts

```

```
END: NOP 0
```

**Crear los bloques de datos de instancia**

Cree tres bloques de datos y ábralos uno tras otro. En el cuadro de diálogo "Nuevo bloque de datos", elija la opción "DB asociado a un FB". En el cuadro de lista "Asignación", elija "FB 1". Así, los bloques de datos quedarán asociados fijamente al FB 1 en calidad de DBs de instancia.

### A.5.2.5 Crear la FC para las válvulas

#### Lógica requerida en la FC

La FC para las válvulas de entrada, alimentación y de salida contiene las siguientes funciones lógicas:

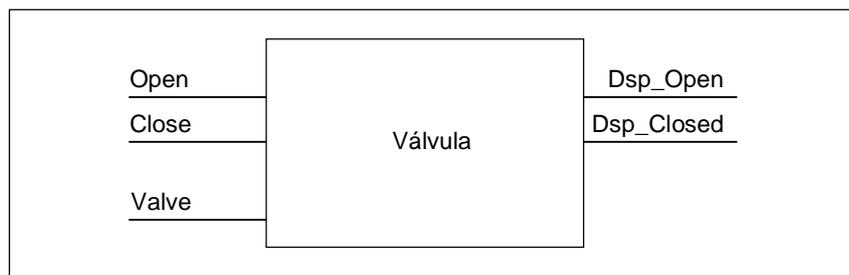
- Existe una entrada para abrir y una para cerrar las válvulas.
- Una serie de enclavamientos permiten la apertura de las válvulas. El estado de los enclavamientos se memoriza en los datos locales temporales (pila L) del OB 1 (p.ej., "Habilitación\_válvula") y se combinan luego con las entradas para abrir y cerrar al ejecutarse la FC para las válvulas.

La tabla siguiente muestra los parámetros que deben transferirse a la FC.

Parámetros para las válvulas	Entrada	Salida	Entrada/salida
Open	✓		
Close	✓		
Dsp_Open		✓	
Dsp_Closed		✓	
Valve			✓

#### Determinar las entradas y las salidas

La figura siguiente muestra las entradas y las salidas de la FC general para las válvulas. Los aparatos que llaman el FB para el motor transfieren los parámetros de entrada. La FC para las válvulas devuelve los parámetros de salida.



## Declaración de variables de la FC para las válvulas

Al igual que en el FB para el motor, en la FC para las válvulas se deben declarar los parámetros de entrada, salida y entrada/salida (v. siguiente tabla de declaración de variables).

Dirección	Declaración	Designación	Tipo	Valor inicial
0.0	IN	Open	BOOL	FALSE
0.1	IN	Close	BOOL	FALSE
2.0	OUT	Dsp_Open	BOOL	FALSE
2.1	OUT	Dsp_Closed	BOOL	FALSE
4.0	IN_OUT	Valve	BOOL	FALSE

En las FC, las variables temporales se memorizan en la pila L. Las variables de entrada, salida y de entrada/salida se memorizan como punteros en el bloque lógico que ha llamado la FC. Para estas variables se utiliza un área de memoria adicional en la pila L (tras las variables temporales).

## Programar la FC para las válvulas

También la función FC 1 para las válvulas debe ser creada antes del OB 1, ya que los bloques llamados deben ser creados antes de los invocantes.

El área de instrucciones de la FC 1 en el lenguaje de programación AWL tiene la estructura siguiente:

### Segmento 1 Abrir/cerrar y autorretención

```

U(
  O   #Open
  O   #Valve
)
UN   #Close
=   #Valve

```

### Segmento 2 Señalización "válvula abierta"

```

U #Valve
= #Dsp_Open

```

### Segmento 3 Señalización "válvula cerrada"

```

UN#Valve
= #Dsp_Closed

```

### A.5.2.6 Crear el OB 1

El OB 1 determina la estructura del programa de ejemplo. Además, el OB 1 contiene los parámetros que se transfieren a las diferentes funciones, tales como:

- Los segmentos AWL para las bombas de alimentación y el motor suministran al FB para el motor los parámetros de entrada para arrancar ("Marcha"), parar ("Paro") y para la señalización en respuesta ("Respuesta"), así como para el reset del indicador de mantenimiento (Reset\_mant). El FB para el motor se ejecuta en cada ciclo del PLC.
- Cuando se ejecuta el FB para el motor, las entradas "Timer\_No" y "Response\_Time" determinan qué temporizador se utilizará y el intervalo de tiempo en el que se debe recibir una respuesta.
- Las FC para las válvulas son ejecutadas automáticamente en cada ciclo de programa del PLC.

El programa utiliza el FB para el motor con diferentes DBs de instancia para cumplir las tareas de control de las bombas de alimentación y del motor del agitador.

### Declarar variables para el OB 1

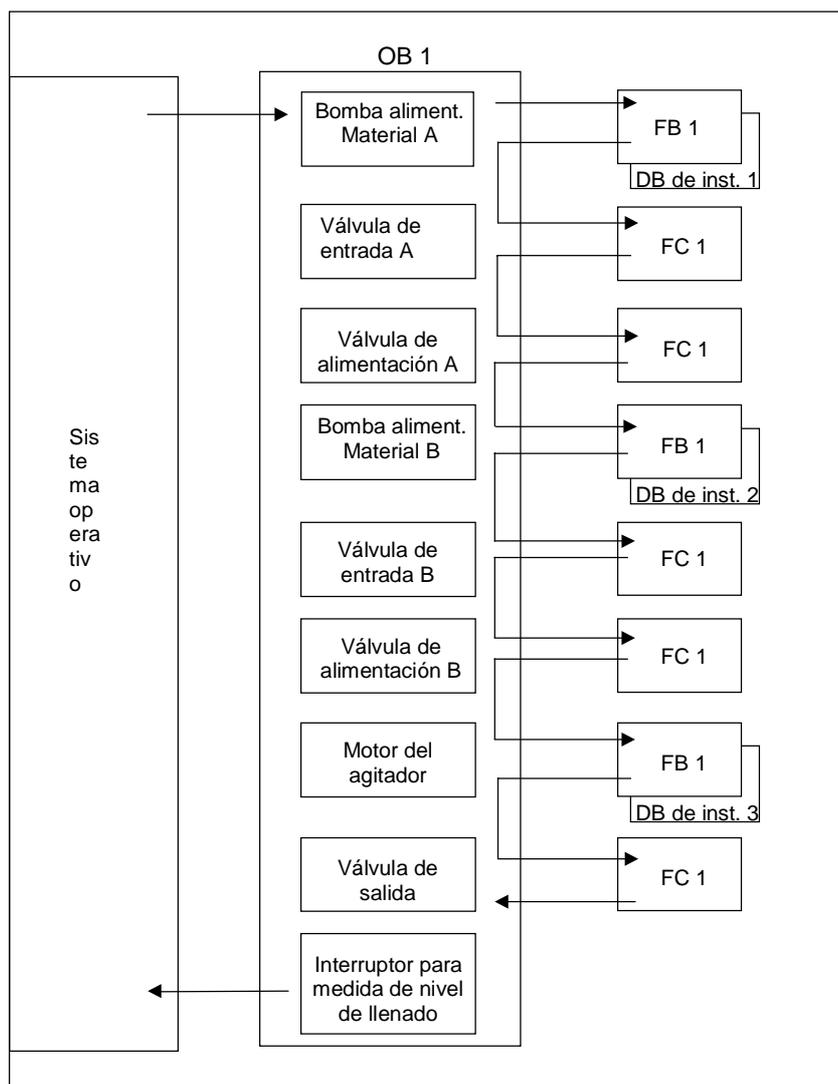
A continuación se muestra la tabla de declaración de variables del OB 1. Los primeros 20 bytes contienen la información de arranque del OB 1 y no se pueden modificar.

Dirección	Declaración	Designación	Tipo
0.0	TEMP	OB1_EV_CLASS	BYTE
1.0	TEMP	OB1_SCAN1	BYTE
2.0	TEMP	OB1_PRIORITY	BYTE
3.0	TEMP	OB1_OB_NUMBR	BYTE
4.0	TEMP	OB1_RESERVED_1	BYTE
5.0	TEMP	OB1_RESERVED_2	BYTE
6.0	TEMP	OB1_PREV_CYCLE	INT
8.0	TEMP	OB1_MIN_CYCLE	INT
10.0	TEMP	OB1_MAX_CYCLE	INT
12.0	TEMP	OB1_DATE_TIME	DATE_AND_TIME
20.0	TEMP	Enable_Motor	BOOL
20.1	TEMP	Enable_Valve	BOOL
20.2	TEMP	Start_Fulfilled	BOOL
20.3	TEMP	Stop_Fulfilled	BOOL
20.4	TEMP	Inlet_Valve_A_Open	BOOL
20.5	TEMP	Inlet_Valve_A_Closed	BOOL
20.6	TEMP	Feed_Valve_A_Open	BOOL
20.7	TEMP	Feed_Valve_A_Closed	BOOL
21.0	TEMP	Inlet_Valve_B_Open	BOOL
21.1	TEMP	Inlet_Valve_B_Closed	BOOL
21.2	TEMP	Feed_Valve_B_Open	BOOL
21.3	TEMP	Feed_Valve_B_Closed	BOOL
21.4	TEMP	Open_Drain	BOOL
21.5	TEMP	Close_Drain	BOOL
21.6	TEMP	Close_Valve_Fulfilled	BOOL

## Crear el programa para el OB 1

En STEP 7, cada bloque que es llamado por otro bloque debe ser creado antes que el bloque que contiene la llamada. Por lo tanto, en el programa de ejemplo, tanto el FB para el motor como la FC para las válvulas se deben crear antes que el programa contenido en el OB 1.

Los bloques FB 1 y FC 1 se llaman varias veces en el OB 1 (el FB 1 con diversos DBs de instancia):



El área de instrucciones del OB 1 en el lenguaje de programación AWL tiene la estructura siguiente:

### Segmento 1 Enclavamientos para bomba de alimentación A

```
U "EMER_STOP_off"
U   "Tank_below_max"
UN  "Drain"
=   #Enable_Motor
```

### Segmento 2 Llamada FB de motor para material A

```
U "Feed_pump_A_start"
U   #Enable_Motor
=   #Start_Fulfilled
U(
O   "Feed_pump_A_stop"
ON  #Enable_Motor
)
=   #Stop_Fulfilled
CALL "Motor_block", "DB_feed_pump_A"
Start :=#Start_Fulfilled
Stop  :=#Stop_Fulfilled
Response :="Flow_A"
Reset_Maint :="Reset_maint"
Timer_No :=T12
Reponse_Time:=S5T#7S
Fault :="Feed_pump_A_fault"
Start_Dsp :="Feed_pump_A_on"
Stop_Dsp :="Feed_pump_A_off"
Maint :="Feed_pump_A_maint"
Motor :="Feed_pump_A"
```

### Segmento 3 Retardo de la habilitación de válvula material A

```
U "Feed_pump_A"
L   S5T#1S
SE  T   13
UN  "Feed_pump_A"
R   T   13
U   T   13
=   #Enable_Valve
```

### Segmento 4 Control de válvula de entrada para material A

```
UN"Flow_A"
UN  "Feed_pump_A"
=   #Close_Valve_Fulfilled
CALL "Valve_block"
Open :=#Enable_Valve
Close :=#Close_Valve_Fulfilled
Dsp_Open :=#Inlet_Valve_A_Open
Dsp_Closed:=#Inlet_Valve_A_Closed
Valve :="Inlet_Valve_A"
```

**Segmento 5 Control de válvula de alimentación para material A**

```

UN "Flow_A"
  UN   "Feed_pump_A"
  =    #Close_Valve_Fulfilled
  CALL "Valve_block"
  Open :=#Enable_Valve
  Close :=#Close_Valve_Fulfilled
  Dsp_Open :=#Feed_Valve_A_Open
  Dsp_Closed:=#Feed_Valve_A_Closed
  Valve :="Feed_Valve_A"

```

**Segmento 6 Enclavamientos para bomba de alimentación B**

```

U "EMER_STOP_off"
  U   "Tank_below_max"
  UN  "Drain"
  =   "Enable_Motor"

```

**Segmento 7 Llamada FB de motor para material B**

```

U "Feed_pump_B_start"
  U   #Enable_Motor
  =   #Start_Fulfilled
  U(
  O   "Feed_pump_B_stop"
  ON  #Enable_Motor
  )
  =   #Stop_Fulfilled
  CALL "Motor_block", "DB_feed_pump_B"
  Start :=#Start_Fulfilled
  Stop :=#Stop_Fulfilled
  Response :="Flow_B"
  Reset_Maint :="Reset_maint"
  Timer_No :=T14
  Reponse_Time:=S5T#7S
  Fault :="Feed_pump_B_fault"
  Start_Dsp :="Feed_pump_B_on"
  Stop_Dsp :="Feed_pump_B_off"
  Maint :="Feed_pump_B_maint"
  Motor :="Feed_pump_B"

```

**Segmento 8 Retardo de la habilitación de válvula material B**

```

U "Feed_pump_B"
  L   S5T#1S
  SE  T   15
  UN  "Feed_pump_B"
  R   T   15
  U   T   15
  =   #Enable_Valve

```

**Segmento 9 Control de válvula de entrada para material B**

```

UN "Flow_B"
  UN   "Feed_pump_B"
  =    #Close_Valve_Fulfilled
  CALL "Valve_block"
  Open :=#Enable_Valve
  Close :=#Close_Valve_Fulfilled
  Dsp_Open :=#Inlet_Valve_B_Open
  Dsp_Closed:=#Inlet_Valve_B_Closed
  Valve :="Inlet_Valve_B"

```

**Segmento 10 Control de válvula de alimentación para material B**

```

UN "Flow_B"
  UN   "Feed_pump_B"
  =    #Close_Valve_Fulfilled
  CALL "Valve_block"
  Open :=#Enable_Valve
  Close :=#Close_Valve_Fulfilled
  Dsp_Open :=#Feed_Valve_B_Open
  Dsp_Closed:=#Feed_Valve_B_Closed
  Valve :="Feed_Valve_B"

```

**Segmento 11 Enclavamientos para agitador**

```

U "EMER_STOP_off"
  U   "Tank_above_min"
  UN  "Drain"
  =   #Enable_Motor

```

**Segmento 12 Llamada FB de motor para agitador**

```

U "Agitator_start"
  U           #Enable_Motor
  =          #Start_Fulfilled
  U(
  O           "Agitator_stop"
  ON         #Enable_Motor
  )
  =          #Stop_Fulfilled
  CALL "Motor_block", "DB_Agitator"
  Start :=#Start_Fulfilled
  Stop :=#Stop_Fulfilled
  Response :="Agitator_running"
  Reset_Maint :="Reset_maint"
  Timer_No :=T16
  Reponse_Time:=S5T#10S
  Fault :="Agitator_fault"
  Start_Dsp :="Agitator_on"
  Stop_Dsp :="Agitator_off"
  Maint :="Agitator_maint"
  Motor :="Agitator"

```

**Segmento 13 Enclavamientos para válvula de salida**

```

U "EMER_STOP_off"
  U   "Tank_not_empty"
  UN  "Agitator"
  =   "Enable_Valve"

```

**Segmento 14 Control de válvula de salida**

```

U "Drain_open"
  U           #Enable_Valve
  =           #Open_Drain
  U(
  O           "Drain_closed"
  ON  #Enable_Valve
  )
  =           #Close_Drain
  CALL "Valve_block"
  Open :=#Open_Drain
  Close :=#Close_Drain
  Dsp_Open :="Drain_open_disp"
  Dsp_Closed :="Drain_closed_disp"
  Valve :="Drain"

```

**Segmento 15 Indicación del nivel de llenado de recipiente**

```

UN"Tank_below_max"
  =   "Tank_max_disp"
UN  "Tank_above_min"
  =   "Tank_min_disp"
UN  "Tank_not_empty"
  =   "Tank_empty_disp"

```

**A.5.3 Ejemplo de aplicación de las alarmas horarias****A.5.3.1 Estructura del programa de usuario "Alarmas horarias"****Tarea**

La salida A 4.0 debe estar activada en el período comprendido entre el lunes, 5.00 horas y el viernes, 20.00 horas. En el período comprendido entre el viernes, 20.00 horas y el lunes, 5.00 horas, la salida A 4.0 debe estar desactivada.

## Aplicación en el programa de usuario

La tabla siguiente muestra las tareas parciales de los bloques utilizados.

Bloque	Tarea parcial
OB 1	Llamada de la función FC 12
FC 12	Dependiendo del estado de la salida A 4.0, del estado de la alarma horaria y de las entradas E 0.0 y E 0.1 <ul style="list-style-type: none"> <li>• Predeterminar hora de arranque</li> <li>• Activar la alarma horaria</li> <li>• Disparar la alarma horaria</li> <li>• CAN_TINT</li> </ul>
OB 10	Dependiendo del día actual de la semana <ul style="list-style-type: none"> <li>• Predeterminar la hora de arranque</li> <li>• Activar o desactivar la salida A 4.0</li> <li>• Activar la siguiente alarma horaria</li> <li>• Disparar la alarma horaria</li> </ul>
OB 80	Activar la salida A 4.1 Almacenar en el área de marcas la información de evento de arranque del OB 80

## Operandos utilizados

La tabla siguiente muestra los operandos globales utilizados. Las variables temporales de los bloques se declaran en el área de declaración del bloque en cuestión.

Operando	Significado
E0.0	Entrada de habilitación de "Activar alarma horaria" y "Disparar alarma horaria"
E0.1	Entrada para cancelar una alarma horaria
A 4.0	Salida activada/desactivada por el OB de alarma horaria (OB 10)
A 4.1	Salida activada al ocurrir un error de tiempo (OB 80)
MW 16	ESTADO (STATUS) de la alarma horaria (SFC 31 "QRY_TINT")
MB 100 hasta MB 107	Memoria para la información de evento de arranque del OB 10 (sólo hora)
MB 110 hasta MB 129	Memoria para la información de evento de arranque del OB 80 (error de tiempo)
MW 200	RET_VAL de la SFC 28 "SET_TINT"
MB 202	Memoria intermedia del resultado binario (bit de estado RB) para SFCs
MW 204	RET_VAL de la SFC 30 "ACT_TINT"
MW 208	RET_VAL de la SFC 31 "QRY_TINT"

## SFCs y FCs utilizadas

En el programa de ejemplo se utilizan las siguientes funciones de sistema:

- SFC 28 "SET\_TINT" : Ajustar la alarma horaria
- SFC 29 "CAN\_TINT" : Cancelar la alarma horaria
- SFC 30 "ACT\_TINT" : Disparar la alarma horaria
- SFC 31 "QRY\_TINT" : Consultar la alarma horaria
- FC 3 "D\_TOD\_DT" : Reunir DATE y TIME\_OF\_DAY para formar DT

### A.5.3.2 FC 12

#### Area de declaración

En el área de declaración de la FC 12 se declaran las siguientes variables temporales del bloque:

Nombre de la variable	Tipo de datos	Declaración	Comentario
IN_HORA	TIME_OF_DAY	TEMP	Tiempo de arranque predeterminado
IN_FECHA	DATE	TEMP	Fecha de arranque predeterminada
OUT_HORA_FECHA	DATE_AND_TIME	TEMP	Cambio fecha/hora de arranque
OK_MARCA	BOOL	TEMP	Activar habilitación de alarma horaria

#### Area de instrucciones en AWL

Introduzca el siguiente programa de ejemplo AWL en el área de instrucciones de la FC 12:

AWL (FC 12)	Explicación
<pre> Segmento 1: CALL SFC 31 NUM OB := 10 RET_VAL:= MW 208 STATUS := MW 16 </pre>	<pre> SFC QRY_TINT Consultar ESTADO (STATUS) de alarmas horarias. </pre>
<pre> Segmento 2: UN      A      4.0 SPB     lun L       D#1995-1-27 T       #IN_FECHA L       TOD#20:0:0.0 T       #IN_HORA SPA     wnd1 mont:   L       D#1995-1-23 T       #IN_FECHA L       TOD#5:0:0.0 T       #IN_HORA wnd1:   NOP     0 </pre>	<pre> Predeterminar el momento de arranque dependiendo de A 4.0 (en la variable #IN_DATUM y #IN_UHRZEIT) La fecha de arranque es un viernes.  La fecha de arranque es un lunes. </pre>

AWL (FC 12)	Explicación
Segmento 3: CALL FC 3 IN1 := #IN_FECHA IN2 := #IN_HORA RET_VAL := #OUT_HORA FECHA	Cambiar formato de DATE y TIME_OF_DAY a DATE_AND_TIME (ajustar para alarma horaria)
Segmento 4: U E 0.0 UN M 17.2 U M 17.4 = #OK_MARCA	¿Cumplidos todos los requisitos para ajustar la alarma horaria? (entrada activada para habilitación y alarma horaria no activa y OB de alarma horaria cargado)
Segmento 5: U #OK_MARCA SPBNB m001 CALL SFC 28 NUM OB := 10 SDT := #OUT_HORA FECHA PERIOD := W#16#1201 RET_VAL := MW 200	En caso afirmativo, ajustar la alarma horaria ...
m001 U RB = M 202.3	
Segmento 6: U #OK_MARCA SPBNB m002 CALL SFC 30 NUM OB := 10 RET_VAL := MW 204	...y activar la alarma horaria.
m002 U RB = M 202.4	
Segmento 7: U E 0.1 SPBNB m003 CALL SFC 29 NUM OB := 10 RET_VAL := MW 210	Cancelar la alarma horaria si está activada la entrada para cancelar alarmas horarias.
m003 U RB = M 202.5	

### A.5.3.3 OB 10

#### Area de declaración

A diferencia del área de declaración predeterminada para el OB 10, se declaran las siguientes variables temporales del bloque:

- Estructura para toda la información de evento de arranque (STARTINFO)
- Dentro de la estructura STARTINFO, una estructura para la hora (T\_STMP)
- Otras variables temporales del bloque WTAG, IN\_DATUM, IN\_UHRZEIT y OUT\_UHRZEIT\_DATUM

Nombre de la variable	Tipo de datos	Declaración	Comentario
STARTINFO	STRUCT	TEMP	Toda la información de evento de arranque del OB 10 declarada como estructura
E_ID	WORD	TEMP	ID del evento:
PRIORIDAD	BYTE	TEMP	Prioridad
NUM_OB	BYTE	TEMP	Nº de OB
RESERVED_1	BYTE	TEMP	Reservado
RESERVED_2	BYTE	TEMP	Reservado
PERIOD	WORD	TEMP	Periodicidad de la alarma horaria
RESERVED_3	DWORD	TEMP	Reservado
T_STMP	STRUCT	TEMP	Estructura para las indicaciones de hora
AÑO	BYTE	TEMP	
MES	BYTE	TEMP	
DIA	BYTE	TEMP	
HORA	BYTE	TEMP	
MINUTOS	BYTE	TEMP	
SEGUNDOS	BYTE	TEMP	
MSEC_DIA	WORD	TEMP	
	END_STRUCT	TEMP	
	END_STRUCT	TEMP	
DIA	INT	TEMP	Día de la semana
IN_FECHA	DATE	TEMP	Variable de entrada para FC 3 (conversión del formato de tiempo)
IN_HORA	TIME_OF_DAY	TEMP	Variable de entrada para FC 3 (conversión del formato de tiempo)
OUT_HORA_FECHA	DATE_AND_TIME	TEMP	Variable de salida para FC 3 y variable de entrada para SFC 28

### Area de instrucciones en AWL

Introduzca el siguiente programa de ejemplo AWL en el área de instrucciones del OB 10:

AWL (OB 10)	Explicación
Segmento 1:	
L     #STARTINFO.T_STMP.MSEG_DIA	Seleccionar día de la semana
L     W#16#F	
UW	
T     #DIA	y memorizar.
Segmento 2:	
L     #DIA	Si el día de la semana no es un lunes,
L     2	predeterminar el lunes, 5.00 horas como
<>I	siguiente fecha y hora de arranque y
SPB   lun	desactivar A 4.0.
Segmento 3:	
L     D#1995-1-27	
T     #IN_FECHA	De otro modo, es decir, si el día de la
L     TOD#20:0:0.0	semana es un lunes, predeterminar el
T     #IN_HORA	viernes, 20.00 horas como siguiente
SET	fecha y hora de arranque y activar la
=     A 4.0	salida A 4.0.
SPA   wndl	
mont: L     D#1995-1-23	
T     #IN_FECHA	
L     TOD#5:0:0.0	
T     #IN_HORA	
CLR	
=     A 4.0	
wndl: NOP    0	
Segmento 4:	
CALL   FC 3	Concluida la predeterminación de la
IN1     := #IN_FECHA	fecha y hora de arranque.
IN2     := #IN_HORA	Convertir la fecha y hora de arranque
RET_VAL := #OUT_HORA_FECHA	predeterminada al formato DATE_AND_TIME
	(para SFC 28).
Segmento 5:	
CALL SFC 28	
NUM OB  := 10	Ajustar la alarma horaria.
SDT     := #OUT_HORA_FECHA	
PERIOD  := W#16#1201	
RET_VAL := MW 200	
U     RB	
=     M 202.1	
Segmento 6:	
CALL SFC 30	
NUM OB  := 10	Activar la alarma horaria.
RET_VAL := MW 204	
U     RB	
=     M 202.2	
Segmento 7:	
CALL SFC 20	
SRCBLK := #STARTINFO.T_STMP	Transferencia de bloque: Salvar la
RET_VAL := MW 206	indicación de hora de la información de
DSTBLK := P#M 100.0 BYTE 8	evento de arranque del OB 10 en el área
	de marcas comprendida entre MB 100 y MB
	107.

### A.5.3.4 OB 1 y OB 80

Puesto que la información de evento de arranque del OB 1 (programa cíclico) no se evalúa en el presente ejemplo, aquí sólo se representa la información de evento de arranque del OB 80.

#### Area de instrucciones del OB 1

Introduzca el siguiente programa de ejemplo AWL en el área de instrucciones del OB 1:

AWL (OB 1)	Explicación
CALL FC 12	Llamada de la función FC 12

#### Area de declaración del OB 80

A diferencia del área de declaración predeterminada para el OB 80, se declaran las siguientes variables temporales del bloque:

- Estructura para toda la información de evento de arranque (STARTINFO)
- Dentro de la estructura STARTINFO, una estructura para la hora (T\_STMP)

Nombre de la variable	Tipo de datos	Declaración	Comentario
STARTINFO	STRUCT	TEMP	Toda la información de evento de arranque del OB 80 declarada como estructura
E	WORD	TEMP	ID del evento
P	BYTE	TEMP	Prioridad
O	BYTE	TEMP	Nº de OB
R	BYTE	TEMP	Reservado
R	BYTE	TEMP	Reservado
Z1_INFO	WORD	TEMP	Información adicional sobre el evento que ha causado el error.
Z2_INFO	DWORD	TEMP	Informaciones adicionales sobre el ID del evento, la prioridad y el número de OB del evento de error
T	STRUCT	TEMP	Estructura para las indicaciones de hora
AÑO	BYTE	TEMP	
MES	BYTE	TEMP	
DIA	BYTE	TEMP	
HORA	BYTE	TEMP	
MINUTOS	BYTE	TEMP	
SEGUNDOS	BYTE	TEMP	
MSEG_DIA	WORD	TEMP	
	END_STRUCT	TEMP	
	END_STRUCT	TEMP	

## Area de instrucciones del OB 80

Introduzca el siguiente programa de usuario AWL en el área de instrucciones del OB 80 que es invocado por el sistema operativo al presentarse un error de tiempo:

AWL (OB 80)	Explicación
Segmento 1:	
UN A 4.1	Activar salida A 4.1 al ocurrir un error de tiempo.
S A 4.1	
CALL SFC 20	Transferencia de bloque: Salvar toda la información de evento de arranque en el área de marcas comprendida entre MB 110 y MB 129.
SRCBLK := #STARTINFO	
RET_VAL := MW 210	
DSTBLK := P#M 110.0 Byte 20	

### A.5.4 Ejemplo de aplicación de las alarmas de retardo

#### A.5.4.1 Estructura del programa de usuario "Alarmas de retardo"

##### Tarea

Cuando se active la entrada E 0.0, la salida A 4.0 se deberá activar 10 segundos después. Cada vez que se active la entrada E 0.0 se deberá iniciar de nuevo el tiempo de retardo.

En calidad de identificador específico de la aplicación, el momento (segundos y milisegundos) en el que comience la alarma de retardo deberá aparecer en la información de evento de arranque del OB de alarma de retardo (OB 20).

Si E 0.1 se activa en esos 10 segundos, el bloque de organización OB 20 no se deberá llamar, es decir que la salida A 4.0 no se deberá activar.

Cuando se active la entrada E 0.2, la salida A 4.0 se deberá desactivar nuevamente.

##### Aplicación en el programa de usuario

La tabla siguiente muestra las tareas parciales de los bloques utilizados.

Bloque	Tarea parcial
OB 1	Leer la hora actual y preparar para el inicio de la alarma de retardo Dependiendo del cambio de flanco en la entrada E 0.0, iniciar la alarma de retardo Dependiendo del estado de la alarma de retardo y del cambio de flanco en la entrada E 0.1, cancelar la alarma de retardo Dependiendo del estado de la entrada E 0.2, desactivar la salida A 4.0
OB 20	Activar la salida A 4.0 Leer la hora actual y preparar Salvar la información de evento de arranque en el área de marcas

## Operandos utilizados

La tabla siguiente muestra los datos globales utilizados. Las variables temporales de los bloques se declaran en el área de declaración del bloque en cuestión.

Operando	Significado
E0.0	Entrada para la habilitación de "Arrancar alarma de retardo"
E0.1	Entrada para cancelar una alarma de retardo
E0.2	Entrada para desactivar la salida A 4.0
A 4.0	Salida activada por el OB de alarma de retardo (OB 20)
MB 1	Utilizada para la marca de flanco y la memoria intermedia del resultado binario (bit de estado RB) para SFCs
MW 4	STATUS (ESTADO) de la alarma de retardo (SFC 34 "QRY_TINT")
MD 10	Segundos y milisegundos en formato decimal codificado en binario de la información de evento de arranque del OB 1
MW 100	RET_VAL de la SFC 32 "SRT_DINT"
MW 102	RET_VAL de la SFC 34 "QRY_DINT"
MW 104	RET_VAL de la SFC 33 "CAN_DINT"
MW 106	RET_VAL de la SFC 20 "BLKMOV"
MB 120 hasta MB 139	Memoria para la información de evento de arranque del OB 20
MD 140	Segundos y milisegundos en formato decimal codificado en binario de la información de evento de arranque del OB 20
MW 144	Segundos y milisegundos en formato decimal codificado en binario de la información de evento de arranque del OB 1; conforme a la información de evento de arranque del OB 20 (identificador específico de usuario SIGN)

## SFCs utilizadas

En el programa de usuario "Alarmas de retardo" se utilizan las siguientes funciones de sistema:

- SFC 32 "SRT\_DINT" : Arrancar la alarma de retardo
- SFC 33 "CAN\_DINT" : Cancelar la alarma de retardo
- SFC 34 "QRY\_DINT" : Consultar el estado de una alarma de retardo

### A.5.4.2 OB 20

#### Area de declaración

A diferencia del área de declaración predeterminada para el OB 20, se declaran las siguientes variables temporales del bloque:

- Estructura para toda la información de evento de arranque (STARTINFO)
- Dentro de la estructura STARTINFO, una estructura para la hora (T\_STMP)

Nombre de la variable	Tipo de datos	Declaración	Comentario
STARTINFO	STRUCT	TEMP	Información de arranque del OB 20
E_ID	WORD	TEMP	ID del evento
AE_NR	BYTE	TEMP	Nivel de ejecución
OB_NR	BYTE	TEMP	Nº de OB
DK1	BYTE	TEMP	Identificador de datos 1
DK2	BYTE	TEMP	Identificador de datos 2
SIGN	WORD	TEMP	Identificador específico de usuario
DTIME	TIME	TEMP	Tiempo con el que se arranca la alarma de retardo
T_STMP	STRUCT	TEMP	Estructura para las indicaciones de hora (indicación de hora y fecha)
AÑO	BYTE	TEMP	
MES	BYTE	TEMP	
DIA	BYTE	TEMP	
HORA	BYTE	TEMP	
MINUTOS	BYTE	TEMP	
SEGUNDOS	BYTE	TEMP	
MSEC_DIA	WORD	TEMP	
	END_STRUCT	TEMP	
	END_STRUCT	TEMP	

## Area de instrucciones

Introduzca el siguiente programa de ejemplo AWL en el área de instrucciones del OB 20:

AWL (OB 20)	Explicación
Segmento 1: SET =           A 4.0	Activar incondicionalmente la salida A 4.0
Segmento 2: L           AW 4 T           PAW 4	Actualizar inmediatamente la palabra de salida
Segmento 3: L #STARTINFO.T_STMP.SEGUNDOS T           MW 140 L #STARTINFO.T_STMP.MSEC_DIA T           MW 142 L           MD 140 SRD        4 T           MD 140	Leer los segundos de la información de evento de arranque Leer los milisegundos y el día de la semana de la información de evento de arranque
Segmento 4: L           #STARTINFO.SIGN T           MW 144	Eliminar el día de la semana y regrabar los milisegundos (están ahora en formato decimal codificado en binario en MW 142). Leer el momento de arranque de la alarma de retardo (=llamada de SFC 32) de la información de evento de arranque
Segmento 5: CALL SFC 20 SRCBLK    := STARTINFO RET_VAL   := MW 106 DSTBLK    := P#M 120.0 BYTE 20	Copiar la información de evento de arranque en el área de marcas (MB 120 hasta MB 139)

### A.5.4.3 OB 1

#### Area de declaración

A diferencia del área de declaración predeterminada para el OB 1, se declaran las siguientes variables temporales del bloque:

- Estructura para toda la información de evento de arranque (STARTINFO)
- Dentro de la estructura STARTINFO, una estructura para la hora (T\_STMP)

Nombre de la variable	Tipo de datos	Declaración	Comentario
STARTINFO	STRUCT	TEMP	Información de arranque del OB 1
E_ID	WORD	TEMP	ID del evento
AE_NR	BYTE	TEMP	Nivel de ejecución
OB_NR	BYTE	TEMP	Nº de OB
DK 1	BYTE	TEMP	Identificador de datos 1
DK 2	BYTE	TEMP	Identificador de datos 2
CIC_ACT	INT	TEMP	Tiempo de ciclo actual
CIC_MIN	INT	TEMP	Tiempo de ciclo mínimo
CIC_MAX	INT	TEMP	Tiempo de ciclo máximo
T_STMP	STRUCT	TEMP	Estructura para las indicaciones de hora (indicación de hora y fecha)
AÑO	BYTE	TEMP	
MES	BYTE	TEMP	
DIA	BYTE	TEMP	
HORA	BYTE	TEMP	
MINUTOS	BYTE	TEMP	
SEGUNDOS	BYTE	TEMP	
MSEC_DIA	WORD	TEMP	
	END_STRUCT	TEMP	
	END_STRUCT	TEMP	

## Area de instrucciones

Introduzca el siguiente programa de ejemplo AWL en el área de instrucciones del OB 1:

AWL (OB 1)	Explicación
<pre> Segmento 1:   L      #STARTINFO.T_STMP.SEGUNDOS   T      MW 10   L      #STARTINFO.T_STMP.MSEC_DIA   T      MW 12   L      MD 10   SRD    4   T      MD 10 Segmento 2:   U      E 0.0   FP     M 1.0   =      M 1.1 Segmento 3:   U      M 1.1   SPBNB  m001   CALL SFC 32          OB_NR := 20          DTME  := T#10S          SIGN  := MW 12          RET_VAL:= MW 100 m001:  NOP    0 Segmento 4:   CALL SFC 34          OB_NR := 20          RET_VAL:= MW 102          STATUS := MW 4 Segmento 5:   U      E 0.1   FP     M 1.3   =      M 1.4 Segmento 6:   U      M 1.4   U      M 5.2   SPBNB  m002   CALL SFC 33          OB_NR := 20          RET_VAL:= MW 104 m002:  NOP    0          U      E 0.2          R      A 4.0 </pre>	<p>Leer los segundos de la información del evento de arranque</p> <p>Leer los milisegundos y el día de la semana de la información del evento de arranque</p> <p>Eliminar el día de la semana y regrabar los milisegundos (están ahora en formato decimal codificado en binario en MW 12).</p> <p>¿Flanco positivo en la entrada E 0.0?</p> <p>En caso afirmativo, arrancar la alarma de retardo (momento de arranque de la alarma de retardo asignado al parámetro SIGN)</p> <p>Consultar el estado de la alarma de retardo (SFC QRY_DINT)</p> <p>¿Flanco positivo en la entrada E 0.1?</p> <p>... ¿y alarma de retardo activada (bit 2 del estado (STATUS) de la alarma de retardo)?</p> <p>Cancelar entonces la alarma de retardo</p> <p>Desactivar la salida A 4.0 junto con la entrada E 0.2</p>

#### A.5.4.4 Ejemplo de enmascarar y desenmascarar eventos de errores síncronos

El ejemplo siguiente de un programa de usuario muestra cómo enmascarar y desenmascarar eventos de errores síncronos. Con la SFC 36 "MSK\_FLT" se enmascara los siguientes errores en la máscara de errores de programación:

- Errores de infracción de área al leer
- Errores de infracción de área al escribir

Al llamarse la SFC 36 "MSK\_FLT" por segunda vez se enmascara además un error de acceso:

- Error de acceso a la periferia al escribir.

Con la SFC 38 "READ\_ERR" se consultan los eventos de errores síncronos enmascarados. El "error de acceso a la periferia al escribir" se desenmascara nuevamente con la SFC 37 "DMSK\_FLT".

#### Instrucción

A continuación se muestra el OB 1 programado en el ejemplo para el programa de usuario en AWL.

AWL (Segmento 1)	Explicación
UN M 255.0	Marca no remanente M 255.0 (sólo en la primera ejecución=0)
SPBNB m001	
CALL SFC 36	SFC 36 MSK_FLT (enmascarar eventos de errores síncronos)
PRGFLT_SET_MASK :=DW#16#C	Bit2=Bit3=1 (BLFL y BLFS se enmascaran)
ACCFLT_SET_MASK :=DW#16#0	todos los bits=0 (no se enmascara ningún error de acceso)
RET_VAL :=MW 100	Valor de retorno
PRGFLT_MASKED :=MD 10	Salida de la máscara de error de programación actual en MD 10
ACCFLT_MASKED :=MD 14	Salida de la máscara de error de acceso actual en MD 14
m001: U RB	Activar M255.0 cuando se haya enmascarado correctamente
S M 255.0	

AWL (Segmento 2)	Explicación
CALL SFC 36	SFC 36 MSK_FLT (enmascarar eventos de errores síncronos)
PRGFLT_SET_MASK :=DW#16#0	Todos los bits=0 (no se enmascara ningún otro error de programación)
ACCFLT_SET_MASK :=DW#16#8	Bit3=1 (se enmascaran los errores de acceso de escritura)
RET_VAL :=MW 102	Valor de retorno
PRGFLT_MASKED :=MD 20	Salida de la máscara de error de programación actual en MD 20
ACCFLT_MASKED :=MD 24	Salida de la máscara de error de acceso actual en MD 24

AWL (Segmento 3)	Explicación
UN M 27.3 BEB	Fin del bloque cuando el error de acceso de escritura (bit3 en ACCFLT_MASKED) no esté enmascarado

AWL (Segmento 4)	Explicación
L B#16#0 T PAB 16	Acceso de escritura (con valor 0) a PAB 16

AWL (Segmento 5)	Explicación
CALL SFC 38	SFC 38 READ_ERR (consultar eventos de errores síncronos)
PRGFLT_QUERY :=DW#16#0	Todos los bits=0 (no se enmascara ningún error de programación)
ACCFLT_QUERY :=DW#16#8	Bit3=1 (se consulta el error de acceso de escritura)
RET_VAL :=MW 104	Valor de retorno
PRGFLT_CLR :=MD 30	Salida de la máscara de error de programación actual en MD 30
ACCFLT_CLR :=MD 34	Salida de la máscara de error de acceso actual en MD 34
U RB	No se ha presentado ningún error y se ha detectado un error de acceso de escritura
U M 37.3	
NOT	
= M 0.0	Invertir RLO M 0.0=1, si existe PAB 16

AWL (Segmento 6)	Explicación
L B#16#0 T PAB 17	Acceso de escritura (con valor 0) a PAB 17

AWL (Segmento 7)	Explicación
CALL SFC 38	SFC 38 READ_ERR (consultar eventos de errores síncronos)
PRGFLT_QUERY :=DW#16#0	Todos los bits=0 (no se enmascara ningún error de programación)
ACCFLT_QUERY :=DW#16#8	Bit3=1 (se consulta el error de acceso de escritura)
RET_VAL :=MW 104	Valor de retorno
PRGFLT_CLR :=MD 30	Salida de la máscara de error de programación actual en MD 30
ACCFLT_CLR :=MD 34	Salida de la máscara de error de acceso actual en MD 34
U RB	No se ha presentado ningún error y se ha detectado un error de acceso de escritura
U M 37.3	
NOT	
= M 0.1	Invertir RLO M 0.1=1, si existe PAB 17

AWL (Segmento 8)	Explicación
L B#16#0 T PAB 18	Acceso de escritura (con valor 0) a PAB 18

AWL (Segmento 9)	Explicación
CALL SFC 38	SFC 38 READ_ERR (consultar eventos de errores síncronos)
PRGFLT_QUERY :=DW#16#0	Todos los bits=0 (no se enmascara ningún error de programación)
ACCFLT_QUERY :=DW#16#8	Bit3=1 (se consulta el error de acceso de escritura)
RET_VAL :=MW 104	Valor de retorno
PRGFLT_CLR :=MD 30	Salida de la máscara de error de programación actual en MD 30
ACCFLT_CLR :=MD 34	Salida de la máscara de error de acceso actual en MD 34
U RB	No se ha presentado ningún error y se ha detectado un error de acceso de escritura
U M 37.3	
NOT	
= M 0.2	Invertir RLO M 0.2=1, si existe PAB 18

AWL (Segmento 10)	Explicación
L B#16#0	
T PAB 19	Acceso de escritura (con valor 0) a PAB 19

AWL (Segmento 11)	Explicación
CALL SFC 38	SFC 38 READ_ERR (consultar eventos de errores síncronos)
PRGFLT_QUERY :=DW#16#0	Todos los bits=0 (no se enmascara ningún error de programación)
ACCFLT_QUERY :=DW#16#8	Bit3=1 (se consulta el error de acceso de escritura)
RET_VAL :=MW 104	Valor de retorno
PRGFLT_CLR :=MD 30	Salida de la máscara de error de programación actual en MD 30
ACCFLT_CLR :=MD 34	Salida de la máscara de error de acceso actual en MD 34
U RB	No se ha presentado ningún error y se ha detectado un error de acceso de escritura
U M 37.3	
NOT	
= M 0.3	Invertir RLO M 0.3=1, si existe PAB 19

AWL (Segmento 12)	Explicación
CALL SFC 37	SFC 37 DMSK_FLT (desenmascarar eventos de errores síncronos)
PRGFLT_RESET_MASK :=DW#16#0	todos los bits=0 (no se desenmascara ningún error de programación)
ACCFLT_RESET_MASK :=DW#16#8	Bit3=1 (se desenmascara el error de acceso de escritura)
RET_VAL :=MW 102	Valor de retorno
PRGFLT_MASKED :=MD 20	Salida de la máscara de error de programación actual en MD 20
ACCFLT_MASKED :=MD 24	Salida de la máscara de error de acceso actual en MD 24

AWL (Segmento 13)	Explicación
U M 27.3 BEB	Fin del bloque cuando el error de acceso de escritura (bit3 en ACCFLT_MASKED) no esté desenmascarado

AWL (Segmento 14)	Explicación
U M 0.0 SPBNB m002 L EB 0 T PAB 16 m002: NOP 0	Transferir EB 0 a PAB 16, si existe

AWL (Segmento 15)	Explicación
U M 0.1 SPBNB m003 L EB 1 T PAB 17 m003: NOP 0	Transferir EB 1 a PAB 17, si existe

AWL (Segmento 16)	Explicación
U M 0.2 SPBNB m004 L EB 2 T PAB 18 m004: NOP 0	Transferir EB 2 a PAB 18, si existe

AWL (Segmento 17)	Explicación
U M 0.3 SPBNB m005 L EB 3 T PAB 19 m005: NOP 0	Transferir EB 3 a PAB 19, si existe

### A.5.4.5 Ejemplo de bloquear y habilitar eventos de alarma y de error asíncrono (SFC 39 y 40)

En el siguiente ejemplo de programa de usuario se utiliza una parte del programa que no puede ser interrumpida por alarmas. Para esta parte del programa se inhiben con la SFC 39 "DIS\_IRT" las llamadas del OB 35 (alarma horaria) que se habilitan nuevamente con la SFC 40 "EN\_IRT".

En el OB 1 se llaman la SFC 39 y la SFC 40:

AWL (OB 1)	Explicación
U M 0.0	Parte del programa que se puede interrumpir sin problema alguno:
S M 90.1	
U M 0.1	Parte del programa que no se puede interrumpir:
S M 90.0	
:	
:	
CALL SFC 39	Bloquear y rechazar alarmas
MODE :=B#16#2	Modo 2: bloquear OBs de alarma
OB_NR :=35	individuales
RET_VAL :=MW 100	Bloquear el OB35
:	
:	
L PEW 100	
T MW 200	
L MW 90	
T MW 92	
:	
:	
CALL SFC 40	Habilitar alarmas
MODE :=B#16#2	Modo 2: habilitar OBs de alarma
OB_NR :=35	individuales
RET_VAL :=MW 102	Habilitar el OB35
	Parte del programa que se puede
U M 10.0	interrumpir sin problema alguno:
S M 190.1	
U M 10.1	
S M 190.0	
:	
:	

### A.5.4.6 Ejemplo del tratamiento retardado de eventos de alarma y de error asíncrono (SFC 41 y 42)

En el siguiente ejemplo de programa de usuario se utiliza una parte del programa que no puede ser interrumpida por alarmas. En esta parte del programa, las alarmas se retardan con la SFC 41 "DIS\_AIRT", habilitándose luego nuevamente con la SFC 42 "EN\_AIRT".

En el OB 1 se llaman la SFC 41 y la SFC 42:

AWL (OB 1)	Explicación
U M 0.0	Parte del programa que se puede interrumpir sin problema alguno:
S M 90.1	
U M 0.1	Parte del programa que no se puede interrumpir:
S M 90.0	
:	Bloquear y retardar alarmas
:	
CALL SFC 41	Habilitar alarma
RET_VAL :=MW 100	
L PEW 100	En el valor de retorno se indica el número de bloqueos de alarmas introducidos
T MW 200	
L MW 90	En el valor de retorno se indica el número de bloqueos de alarmas introducidos
T MW 92	
:	Tras habilitarse la alarmas, el número deberá tener el mismo valor que antes de bloquearse las mismas (aquí "0")
:	
CALL SFC 42	Parte del programa que se puede interrumpir sin problema alguno:
RET_VAL :=MW 102	
L MW 100	Se indica el número de bloqueos de alarmas introducidos
DEC 1	
L MW 102	
<>I	
SPB err	
U M 10.0	
S M 190.1	
U M 10.1	
S M 190.0	
:	
:	
BEA	
err: L MW 102	
T AW 12	

## A.6 Acceso a áreas de datos del proceso y de la periferia

### A.6.1 Acceso al área de datos de proceso

La CPU puede acceder, bien indirectamente a través de la imagen del proceso o bien directamente a través del bus posterior/de periferia, a las entradas y salidas de los módulos de entradas/salidas digitales centralizados y descentralizados.

La CPU accede directamente a las entradas y salidas de módulos de entradas/salidas analógicas centralizados y descentralizados a través del bus posterior/de periferia.

### Direccionamiento de módulos

Para asignar a los módulos direcciones que son utilizadas en el programa de usuario, hay que configurar los módulos con STEP 7

- para periferia centralizada: la disposición del bastidor y la asignación de los módulos a slots tiene lugar en la tabla de configuración
- en el caso de utilizar periferia descentralizada (PROFIBUS-DP): disposición de los esclavos DP en la tabla de configuración "sistema maestro" otorgando la dirección PROFIBUS y asignando los módulos a slots.

La configuración de los módulos sustituye el ajuste de direcciones de los diferentes módulos a través de interruptores. Como resultado de la configuración, la CPU recibe de la PG los datos que identifican a los módulos asignados.

### Direccionamiento de periferia

Las entradas y las salidas disponen de una área propia de direcciones. Por consiguiente, la dirección de una área de periferia debe contener, además de la indicación de byte o palabra, el identificador E para las entradas y A para las salidas.

La tabla siguiente muestra las áreas de direcciones de periferia disponibles.

Area de operandos	Acceso a través de unidades del tamaño siguiente:	Notación S7
Area de periferia: entradas	Byte de entrada de periferia Palabra de entrada de periferia Palabra doble de entrada de periferia	PEB PEW PED
Area de periferia: salidas	Byte de salida de periferia Palabra de salida de periferia Palabra doble de salida de periferia	PAB PAW PAD

Los manuales siguientes indican qué áreas de direcciones son posibles en cada módulo:

- Manual "Sistema de automatización S7-300, Configuración, instalación y datos de las CPU"
- Manual de referencia "Sistemas de automatización S7-300, M7-300, Datos de los módulos"
- Manual de referencia "Sistemas de automatización S7-400, M7-400, Datos de los módulos"

## Dirección inicial de módulos

La dirección inicial de un módulo es la menor dirección de byte de dicho módulo. Representa la dirección inicial del área de datos útiles del módulo y se utiliza en muchos casos como representante de todo el módulo.

Se introduce en la información de arranque del correspondiente bloque de organización, p. ej., en el caso de alarmas de proceso, alarmas de diagnóstico, alarmas de presencia de módulo y errores de alimentación, identificando así al módulo que ha indicado la alarma.

### A.6.2 Acceso al área de datos de periferia

El área de datos de periferia se subdivide en:

- datos útiles y
- datos de diagnóstico y de parámetros.

Ambas disponen de una área de entrada (sólo acceso en lectura) y una área de salida (sólo acceso en escritura).

#### Datos útiles

A los datos útiles se accede a través de la dirección de byte (para módulos de señales digitales) o de la dirección de palabra (para módulos de señales analógicas) del área de entrada o salida. A los datos útiles se puede acceder a través de instrucciones de carga y transferencia, funciones de comunicaciones (accesos M+V) o a través de transferencia de imágenes de proceso. Datos útiles pueden ser:

- señales de entrada/salida digitales y analógicas de módulos de señales,
- informaciones de control y estado de módulos de función e
- informaciones para conexiones punto a punto y por bus vía módulos de comunicaciones (sólo S7-300).

Al transferirse datos útiles, se puede alcanzar una coherencia de datos de 4 bytes como máximo (con excepción de los esclavos estándar DP, consulte también "Ajustar el comportamiento de servicio"). Si se utiliza la instrucción "transferir palabra doble", se transmiten los 4 bytes correspondientes y sin modificar (coherentes). Si se utilizan cuatro instrucciones individuales "transferir byte de entrada", en un límite de instrucción podría arrancarse un OB de alarma de proceso, el cual transmite los datos a la misma dirección y, por consiguiente, modifica el contenido de los cuatro bytes originales.

## Datos de diagnóstico y de parámetros

Los datos de diagnóstico y de parámetros de un módulo no se pueden direccionar individualmente, sino reunidos en registros completos. Se transfieren normalmente de forma coherente.

Los datos de diagnóstico y de parámetros se direccionan a través de la dirección inicial del módulo correspondiente y del número de registro. Los registros se subdividen en registros de entrada y de salida. Los registros de entrada sólo se pueden leer; en los de salida sólo se puede escribir. A los registros se puede acceder a través de las funciones del sistema o de funciones de comunicaciones (manejo y visualización, M+V). La tabla siguiente muestra la asignación de registros a los datos de diagnóstico y de parámetros.

Datos	Descripción
Datos de diagnóstico	En módulos diagnosticables, al leer los registros 0 y 1 se obtienen los datos de diagnóstico del módulo.
Datos de parámetros	En módulos parametrizables, el escribir en los registros 0 y 1 se transfieren los parámetros del módulo.

## Acceso a los registros

Las informaciones contenidas en los registros de un módulo se pueden utilizar para reparametrizar módulos parametrizables y leer las informaciones de diagnóstico de módulos diagnosticables.

La tabla siguiente muestra con qué funciones del sistema se puede acceder a los registros.

SFC	Aplicación
Parametrizar módulos	
SFC 55 WR_PARM	Transmitir los parámetros modificables (registro 1) al módulo de señales direccionado.
SFC 56 WR_DPARM	Transmitir los parámetros desde los SDB 100 a 129 al módulo de señales direccionado.
SFC 57 PARM_MOD	Transmitir los parámetros desde los SDB 100 a 129 al módulo de señales direccionado.
SFC 58 WR_REC	Transmitir un registro cualquiera al módulo de señales direccionado.
Leer informaciones de diagnóstico	
SFC 59 RD_REC	Leer los datos de diagnóstico.

## Direccionamiento de módulos S5

Tiene la posibilidad de:

- acoplar aparatos de ampliación de SIMATIC S5 a un S7-400 a través del módulo de interconexión (interfase) IM 463-2 e
- insertar algunos módulos S5, metidos en cápsulas de adaptación, en los bastidores centrales del S7-400.

Para obtener informaciones acerca de cómo direccionar los módulos S5 en SIMATIC S7, consulte el manual de instalación "Sistemas de automatización S7-400, M7-400, Configuración e instalación" o la descripción adjunta de la cápsula de adaptación.

## A.7 Ajustar el comportamiento del sistema

### A.7.1 Ajustar el comportamiento del sistema

Este capítulo explica cómo se puede actuar sobre aquellas propiedades de los sistemas de automatización S7-300 y S7-400 que no están predeterminadas ajustando los parámetros del sistema o utilizando las funciones del sistema SFC.

Para información más detallada sobre los parámetros de los módulos, consultar la Ayuda en pantalla STEP 7, así como los siguientes manuales:

- Manual "Sistema de automatización S7-300, Configuración, instalación y datos de las CPU"
- Manual de referencia "Sistemas de automatización S7-300, M7-300, Datos de los módulos"
- Manual de referencia "Sistemas de automatización S7-400, M7-400, Datos de los módulos"

El manual de referencia "Software de sistema para S7-300/400, funciones estándar y funciones de sistema" contiene descripciones detalladas de las SFCs.

#### Direccionamiento de esclavos DP estándar

Si en esclavos DP estándar se requiere transferir o recibir datos mayores de 4 bytes, es necesario utilizar SFCs especiales para la transferencia de dichos datos.

SFC	Aplicación
Parametrizar módulos	
SFC 15 DPWR_DAT	Transmitir un dato cualquiera al módulo de señales direccionado.
Leer informaciones de diagnóstico	
SFC 13 DPNRM_DG	Leer los datos de diagnóstico (operación de lectura asíncrona).
SFC 14 DPRD_DAT	Leer datos coherentes (longitud 3 o mayor de 4 bytes).

Cuando llega un telegrama de diagnóstico DP se envía a la CPU una alarma de diagnóstico con 4 bytes de datos de diagnóstico. Estos 4 bytes se pueden leer con la SFC 13 DPNRM\_DG.

## A.7.2 Modificar el comportamiento y las propiedades de los módulos

### Ajustes por defecto

- Todos los módulos parametrizables del sistema de automatización S7 se suministran con ajustes por defecto apropiados para aplicaciones estándar. Estos valores por defecto permiten utilizar los módulos sin necesidad de ajustes adicionales. Los valores estándar se indican en las descripciones de los módulos contenidas en los siguientes manuales:
- Manual "Sistema de automatización S7-300, Configuración, instalación y datos de las CPU"
- Manual de referencia "Sistemas de automatización S7-300, M7-300, Datos de los módulos"
- Manual de referencia "Sistemas de automatización S7-400, M7-400, Datos de los módulos"

### ¿Qué módulos pueden parametrizarse?

También es posible parametrizar el comportamiento y las propiedades de los módulos para ajustarlos a los requerimientos y las particularidades de la instalación. Módulos parametrizables son CPUs, FMs y CPs, así como módulos de entradas/salidas analógicas y módulos de entradas digitales.

Se dispone de módulos parametrizables con y sin respaldo de memoria.

Los módulos sin respaldo se deben abastecer con los datos correspondientes tras cada corte de tensión. Los parámetros de dichos módulos están memorizados en el área de memoria remanente de la CPU (parametrización indirecta a través de CPU).

### Ajustar y cargar los parámetros

Los parámetros de los módulos se ajustan con STEP 7. Al guardar los parámetros, STEP 7 crea el objeto "Bloques de datos del sistema" que se carga con el programa de usuario en la CPU, transfiriéndose desde allí durante el arranque a los módulos correspondientes.

## ¿Qué se puede parametrizar?

Los parámetros de módulos se subdividen en bloques de parámetros. En los manuales "Sistema de automatización S7-300, Configuración, instalación y datos de las CPU" y "Sistemas de automatización S7-400, M7-400, Datos de los módulos" se indican los bloques de parámetros disponibles en cada una de las CPUs.

Ejemplos de bloques de parámetros:

- comportamiento en arranque
- ciclo
- MPI
- diagnóstico
- remanencia
- marcas de ciclo
- tratamiento de alarmas
- periferia integrada (sólo para S7-300)
- nivel de protección
- datos locales
- reloj de tiempo real
- error asíncrono

## Parametrizar con SFCs

Además de la parametrización con STEP 7, también es posible modificar desde el programa S7 los parámetros de los módulos a través de las funciones del sistema. La tabla siguiente muestra con qué SFCs se pueden modificar qué parámetros de los módulos.

SFC	Aplicación
SFC 55 WR_PARM	Transmitir los parámetros modificables (registro 1) al módulo de señales direccionado.
SFC 56 WR_DPARM	Transmitir los parámetros desde los SDB correspondientes al módulo de señales direccionado.
SFC 57 PARM_MOD	Transmitir los parámetros desde los SDB correspondientes al módulo de señales direccionado.
SFC 58 WR_REC	Transmitir un registro cualquiera al módulo de señales direccionado.

Las funciones de sistema se describen detalladamente en el manual de referencia "Software de sistema para S7-300/400, funciones estándar y funciones de sistema".

Los manuales siguientes indican qué parámetros de módulos se pueden modificar dinámicamente.

- Manual "Sistema de automatización S7-300, Configuración, instalación y datos de las CPU"
- Manual de referencia "Sistemas de automatización S7-300, M7-300, Datos de los módulos"
- Manual de referencia "Sistemas de automatización S7-400, M7-400, Datos de los módulos"

### A.7.3 Aplicación de las funciones de reloj

Todas las CPUs S7-300 y S7-400 están dotadas con un reloj (reloj de tiempo real o reloj software). En el sistema de automatización el reloj puede operar tanto como maestro como de esclavo con sincronización externa. Permite utilizar alarmas horarias y contadores de horas de funcionamiento.

#### Formato de hora

El reloj muestra siempre la hora (resolución mínima 1 s) y la fecha con el día de la semana. En algunas CPUs también se pueden indicar los milisegundos (consulte los manuales "Sistema de automatización S7-300, Configuración, instalación y datos de las CPU" y "Sistemas de automatización S7-400, M7-400, Datos de los módulos").

#### Ajustar y leer la hora

La hora y la fecha del reloj de la CPU puede ser ajustada bien llamando la SFC 0 SET\_CLK desde el programa de usuario o bien desde la PG mediante el comando de menú, arrancándose así el reloj. Con la SFC 1 READ\_CLK o por comando de menú desde la PG se puede leer la fecha y hora actual de la CPU.

Nota: Para evitar diferencias en la indicación de la hora en sistemas HMI, ajuste el **horario de verano** en la CPU

#### Parametrizar el reloj

Si una red dispone de más de un módulo con reloj, es necesario parametrizar con STEP 7 qué CPU debe funcionar como maestra para la sincronización de la hora y cuál como esclavo. Con la parametrización se define también si la sincronización debe tener lugar a través del bus de comunicación o a través del puerto MPI y con qué intervalos debe efectuarse la sincronización automática.

#### Sincronizar la hora

Para garantizar que coincida la hora de todos los módulos existentes en la red, los relojes esclavos son sincronizados por el programa del sistema en intervalos regulares (parametrizables). Con la función del sistema SFC 48 SFC\_RTCD se puede transferir la fecha y la hora del reloj maestro a los relojes esclavos.

## Utilizar un contador de horas de funcionamiento

Un contador de horas de funcionamiento totaliza los períodos de conexión de un equipo o la duración de servicio de la CPU como suma de las horas de funcionamiento.

El contador de horas de funcionamiento se para durante el estado operativo STOP. Su valor se conserva incluso tras el borrado total. En caso de re arranque completo, el contador de horas de funcionamiento debe ser arrancado nuevamente por el programa de usuario y, en caso de re arranque, continúa operando si ha sido arrancado previamente.

La SFC 2 SET\_RTM permite ajustar el contador de horas de funcionamiento a un valor inicial. Con la SFC 3 CTRL\_RTM se puede arrancar o parar el contador de horas de funcionamiento. La SFC 4 READ\_RTM permite leer la cantidad actual de horas de funcionamiento y el estado del contador ("parado" o "contando").

Una CPU puede tener hasta 8 contadores de horas de funcionamiento. La numeración comienza a partir de 0.

### A.7.4 Uso de marcas de ciclo y temporizadores

#### Marcas de ciclo

Una marca de ciclo es una marca que modifica su estado binario periódicamente con un ciclo de trabajo de 1:1. Parametrizando la marca de ciclo con STEP 7 se puede definir qué byte de marcas de la CPU se utiliza como byte de marcas de ciclo.

#### Utilidad

Las marcas de ciclo se pueden utilizar en el programa de usuario, por ejemplo, para controlar avisadores luminosos con luz intermitente o para iniciar procesos que se repitan periódicamente (como la captación de un valor real).

#### Frecuencias posibles

Cada bit del byte de marcas de ciclo tiene asignada una frecuencia. La tabla siguiente muestra la asignación:

Bit del byte de la marca de ciclo	7	6	5	4	3	2	1	0
Duración del período (s)	2,0	1,6	1,0	0,8	0,5	0,4	0,2	0,1
Frecuencia (Hz)	0,5	0,625	1	1,25	2	2,5	5	10

#### Nota

Las marcas de ciclo corren de forma asíncrona al ciclo de la CPU, es decir, en los ciclos largos puede cambiar varias veces el estado de la marca de ciclo.

## Temporizadores

Los temporizadores representan una área de memoria de la memoria del sistema. La función de un temporizador es determinada por el programa de usuario (p. ej., retardo a la conexión), mientras que la cantidad de temporizadores disponibles depende del tipo de CPU.

---

### Nota

- Si en el programa de usuario se utilizan más temporizadores de los que admite la CPU, se señala un error síncrono y se arranca el OB 121.
  - En el S7-300 (a excepción de la CPU 318), los temporizadores sólo se pueden arrancar y actualizar simultáneamente en el OB 1 y en el OB 100; en todos los demás OBs solamente se pueden arrancar los temporizadores.
-

# Índice alfabético

¿Qué bloques de notificación existen? .....	14-6
¿Qué procedimientos existen para crear mensajes? .....	14-1
Abrir	
tabla de variables .....	18-2
tablas de símbolos .....	7-12
Acceder a la vista rápida .....	21-5
Acceso a la información del módulo desde la vista online del proyecto .....	21-6
ACT_TINT .....	4-23, 4-24
Activar	
visualización de símbolos en el bloque .....	7-11
Actualizar .....	2-3
autorización .....	2-1
Imagen del proceso .....	4-13
llamadas a bloques .....	9-21
Administrador SIMATIC .....	5-1
bloques	
visualizar longitudes de bloques .....	8-13
Ajustar	
condiciones de disparo para forzar variables .....	18-16
el disparo para observar variables .....	18-13
fecha y hora .....	16-5
interface PG/PC .....	2-11
memoria de trabajo virtual .....	25-4
preferencia de operandos .....	7-5
Ajustar las direcciones de las estaciones de la red .....	17-8
Ajustes	
editor AWL .....	9-2
para el lenguaje de programación AWL .....	9-20
Ajustes para el lenguaje de programación FUP	9-17
Ajustes para el lenguaje de programación KOP	9-14
Ajustes para notificar errores del sistema .....	14-27
Alarma cíclica .....	4-25, 4-26
arrancar .....	4-25
Regulación .....	4-25
Alarma de diagnóstico (OB 82) .....	21-34
Alarma de presencia de módulo (OB 83) .....	21-35
Alarma de proceso	
arrancar .....	4-26, 4-27
prioridad .....	4-27
reglas .....	4-27
Alarma de retardo	
arrancar .....	4-24, 4-25
prioridad .....	4-25
reglas .....	4-24
Alarma horaria .....	4-23, 4-24
arrancar .....	4-23
cambiar la hora .....	4-24
consultar .....	4-23
desactivar .....	4-24
prioridad .....	4-24
regular .....	4-23
Alarmas cíclicas .....	4-25
Alarmas de proceso .....	4-26, 4-27
Alarmas horarias .....	4-23, 4-24
Ancho del campo de operando .....	9-14, 9-17
Archivador mensajes de CPU .....	14-20, 14-21
Archivar	
ejemplos de aplicación .....	22-5
procedimiento .....	22-6
proyectos y librerías .....	22-5
requisitos .....	22-6
Archivos fuente en S7-GRAPH .....	8-7
Area de instrucciones .....	9-1, 9-5
editar .....	9-10
en KOP .....	9-4
estructura .....	9-10
función de búsqueda de errores .....	9-13
Areas y tareas	
subdividir el proceso en .....	3-2
Arquitectura del sistema	
ciclo .....	4-10
Arrancar .....	4-23
alarma cíclica .....	4-25
alarma de proceso .....	4-26
alarma de retardo .....	4-24
alarma horaria .....	4-23
Arrancar STEP 7 con los parámetros de arranque estándar .....	5-2
Asignar números a los mensajes .....	14-5
Asignar y editar mensajes de símbolos .....	14-14
Asistente para crear un proyecto .....	6-3
Atributos de bloques y parámetros .....	8-14
Atributos de sistema .....	14-8
en la tabla de símbolos .....	7-7
para la configuración de mensajes .....	14-7
para la configuración de mensajes PCS7 ...	14-13
Atributos de sistema para parámetros .....	9-4
Atributos M+V .....	15-1
cambiar con CFC .....	15-5
configurar a través de la tabla de símbolos ...	15-4

configurar con AWL		introducir en AWL.....	9-10
FUP		recablear.....	8-14
KOP.....	15-3	Bloques (cargados)	
AuthorsW .....	2-2	editar en la PG/en el PC .....	17-15
AuthorsW.exe .....	2-1	Bloques de datos .....	10-1
Autorización.....	2-1	cambiar valores en la vista "Datos" .....	10-7
a posteriori .....	2-1	guardar .....	10-8
actualización .....	2-3	inicializar los valores de datos.....	10-8
cantidad.....	2-4	introducir/visualizar la estructura de datos con FB	
de emergencia .....	2-2	asociado (DBs de instancia).....	10-4
desinstalar .....	2-1, 2-2, 2-4	nociones básicas.....	10-1
disquete original .....	2-1	ver datos .....	10-3
pérdida .....	2-1	ver declaración.....	10-2
primera instalación .....	2-2	Bloques de datos de instancia .....	4-18
reglas.....	2-4	indicación de fecha y hora .....	13-5
transferir .....	2-2, 2-4	Bloques de datos globales.....	13-5
AWL.....	8-2, 8-3, 8-5	indicación de fecha y hora .....	13-5
ajustes .....	9-20	introducir la estructura .....	10-4
introducir bloques.....	9-10	Bloques de datos globales (DB).....	4-20
visualizar información sobre el bloque.....	12-9	Bloques de función (FB) .....	4-15
Ayuda (en pantalla)		Bloques de función (FBs).....	4-2, 4-15
llamar.....	5-4	campo de aplicación .....	4-15
Temas .....	5-3	parámetros actuales .....	4-16, 4-17
Ayuda contextual .....	5-4	Bloques de función del sistema.....	4-2
Ayuda en pantalla		Bloques de funciones del sistema .....	4-21
cambiar el tamaño de letra .....	5-4	tipos.....	4-21
llamar.....	5-3	Bloques de funciones del sistema (SFB)	
temas.....	5-3	y funciones del sistema (SFC).....	4-21
Barra de estado		Bloques de notificación, Sinopsis.....	14-6
ejemplo.....	5-17	Bloques de organización .....	4-2
Barra de herramientas		definición.....	4-3
botones.....	5-18	valores de sustitución .....	21-28, 21-29
Bautizar las estaciones de la red .....	17-8	información de arranque .....	4-5
Bloque		prioridades .....	4-3, 4-5
asignar y editar mensajes de bloque .....	14-6	reaccionar a errores.....	4-31
definir el entorno de llamada.....	19-9	Bloques de organización de alarma cíclica	
Bloque apto para notificación.....	14-10	(OB 30 a OB 38).....	4-25
Bloque de datos (DB) .....	4-2	Bloques de organización de alarma de proceso	
bloques de datos de instancia.....	4-15, 4-18	(OB 40 a OB 47).....	4-26
estructura.....	4-20	Bloques de organización de alarma de retardo	
global.....	4-20	(OB 20 a OB 23).....	4-24
Bloque de datos de instancia.....	4-18, 4-19	Bloques de organización de alarma horaria	
crear varias instancias para un FB .....	4-15	(OB 10 a OB 17).....	4-23
Bloque de organización (OB)		Bloques de organización de arranque	
OB de tarea no prioritaria (OB 90).....	4-3, 4-30	(OB100/OB101/OB102).....	4-28
Bloque de organización de tarea no prioritaria		Bloques de organización de tratamiento de errores	
(OB 90).....	4-30	(OB70aOB87 / OB 121 a OB 122).....	4-31
Bloque de organización para la ejecución		Bloques de organización para la alarma	
cíclica del programa (OB1) .....	4-10	horaria (OB 10 a OB 17) .....	4-23
Bloques .....	4-2, 13-1, 13-2	Bloques de organización para la ejecución	
atributos.....	8-14	controlada por alarmas .....	4-22
borrar en el sistema de destino .....	17-19	Bloques de organización y estructura del	
cargar desde una CPU S7 .....	17-14	programa.....	4-3
crear con S7-GRAPH .....	8-7	Bloques lógicos .....	13-4
del programa de usuario.....	4-2	crear .....	9-1
derechos de acceso .....	9-3	en el editor incremental.....	9-1
guardar .....	9-22	estructura.....	9-1

guardar .....	9-22	Cargar la configuración de un equipo en la PG .....	17-16
indicación de fecha y hora .....	13-4	Cargar la configuración de un equipo en un sistema de destino .....	17-7
Bloques modificados .....	17-5	Cargar la configuración desde otro equipo .....	17-16
cargar en el sistema de destino .....	17-5	Cargar modificaciones de la configuración de red .....	17-10
Bloques preprogramados .....	4-21	Cargar por primera vez la configuración de la red .....	17-8
Bobinas		Cargar una configuración de red en la PG .....	17-16
emplazar .....	9-15	Cargar una configuración en la PG .....	17-16
Borrado total		Cargar una configuración en un sistema de destino .....	17-7
de la CPU .....	17-18	Carpeta de bloques .....	5-12, 8-10
Borrar .....	5-19, 5-23	Carpeta de fuentes .....	5-14, 5-15
bloques S7 en el sistema de destino .....	17-19	Catálogo de hardware .....	1-9
memorias de carga y de trabajo .....	17-18	Catálogo de elementos de programa .....	9-3
objetos STEP 7 .....	5-19	CFC .....	8-3
Botones		Ciclo .....	4-3, 4-4, 4-6, 4-10, 4-11, 4-12, 4-13, 4-14
barra de herramientas .....	5-18	Coherencia en fuentes AWL, comprobar .....	11-16
Búfer de diagnóstico		Combinaciones de teclas para acceder a la Ayuda en pantalla .....	5-30
contenido .....	21-22	Combinaciones de teclas para cambiar de una ventana a otra .....	5-31
leer .....	21-17	Combinaciones de teclas para comandos de menú .....	5-27
Buscar errores en fuentes AWL .....	11-16	Combinaciones de teclas para desplazar el cursor .....	5-29
Búsqueda de errores		Combinaciones de teclas para seleccionar textos .....	5-30
en bloques .....	9-13	Comentarios	
Búsqueda de errores y fallos .....	21-1	de bloques .....	9-12
Cambiar .....	4-23	de segmentos .....	9-13
estado operativo .....	16-4	Comentarios de bloques .....	9-13
la hora de una alarma horaria .....	4-23	Comentarios de segmentos .....	9-12
valores en la vista "Datos" de bloques de datos .....	10-7	Cómo iniciar STEP 7 .....	5-1
Cambiar atributos M+V con CFC .....	15-5	Compilar	
Cambiar de una ventana a otra .....	5-31	fuentes AWL .....	11-17
Cambiar el estado operativo de la CPU durante la carga .....	17-8	Componentes compatibles y funcionalidad .....	14-25
Cambiar el nombre .....	5-21, 5-22	Componentes de proyectos .....	22-1
proyectos .....	5-19, 5-22	imprimir .....	22-1, 22-2
Cambiar la dirección de estación .....	17-9	Componentes SIMATIC para la configuración de mensajes .....	14-4
Cambiar la dirección de estación de equipos S7 .....	17-9	Comprimir	
Cambiar la dirección PROFIBUS de esclavos DP .....	17-8	el contenido de la memoria de una CPU S7 .....	17-20
Cambiar la organización de ventanas .....	5-25	la memoria de usuario .....	17-19
CAN_TINT .....	4-24	Comprobar	
Carga del ciclo por comunicaciones .....	4-14	coherencia en fuentes AWL .....	11-16
Carga por comunicaciones .....	4-13, 4-14	datos de referencia .....	12-11
Cargar		Comprobar coherencia de bloques .....	13-1
bloques desde una CPU S7 .....	17-14	Comprobar la asignación de direcciones .....	2-11
en la PG desde el sistema de destino .....	17-13	Comprobar la asignación de interrupciones .....	2-12
la configuración actual y todos los bloques en la PG .....	17-14	Comprobar la coherencia de la configuración de un equipo .....	17-7
mediante Memory Cards EPROM .....	17-6	Comunicación .....	17-8, 17-9, 17-10
programas de usuario en el sistema de destino .....	17-2		
requisitos .....	17-1		
Cargar enlaces en la PG .....	17-16		
Cargar la configuración de datos globales .....	17-11		
Cargar la configuración de la red .....	17-8, 17-10		
Cargar la configuración de la red en un sistema de destino .....	17-9		

Condición de disparo.....	18-13	Crear .....	5-19
Condiciones de disparo para analizar el estado del programa.....	19-9	datos de referencia .....	12-11
Configuración de datos globales configurar.....	17-11	diagrama de E/S para las válvulas .....	3-7
Configuración de mensajes componentes SIMATIC.....	14-4	diagrama de E/S para los motores .....	3-6
transferir datos a WinCC.....	14-19	esquema de configuración .....	3-10
Configuración de mensajes PCS7 .....	14-12, 14-13	fuentes AWL.....	11-13
Configuración multiusuario .....	23-1	objetos .....	5-19
Configuración multiusuario en una red Windows .....	23-1	tabla de variables .....	18-2
Configurar.....	25-1	Crear el programa	
Configurar atributos M+V en AWL KOP y FUP .....	15-3	procedimiento general .....	1-1
Configurar atributos M+V a través de la tabla de símbolos.....	15-4	Crear objeto .....	5-19
Configurar la notificación de errores del sistema.....	14-23	Crear programas de usuario .....	9-1
Configurar mensajes con errores del sistema..	14-23	Crear un diagrama de entradas para las válvulas .....	3-7
Configurar mensajes de CPU .....	14-22	Crear un diagrama de entradas para los motores.....	3-6
Configurar variables para manejo y visualización .....	15-1	Crear un diagrama de salidas para las válvulas .....	3-7
Conflictos de fecha y hora .....	13-3	Crear un diagrama de salidas para los motores.....	3-6
Consejos y.....	25-2	Crear un proyecto.....	6-3
Consejos y trucos .....	1-9, 25-1, 25-2, 25-4	Crear y manejar objetos .....	5-19
Consultar la alarma horaria .....	4-23	Cuadro combinado, definición.....	5-18
Contadores		Cuadro de diálogo de selección .....	5-24
plano de ocupación .....	12-7	Cuadro de lista .....	5-18
valores máximos de introducción .....	18-6	Cuadros	
Contenido de las pilas en estado operativo STOP.....	21-14	emplazar .....	9-14, 9-18
Contraseña .....	16-3	retirar	
Contraseña para acceder a los sistemas de destino .....	16-3	cambiar .....	9-19
Control de los tiempos de ciclo para evitar errores de tiempo.....	21-16	Cuadros de diálogo .....	5-18, 5-19
Control de variables .....	18-15	Cuadros de diálogo con fichas .....	5-18
introducción .....	18-15	Datos, tipos de datos de usuario .....	8-11
Control por alarma ejecución del programa .....	4-6	Datos de configuración.....	15-1, 15-2
Control secuencial.....	8-7	requisitos para la transferencia .....	15-6
Controlar el tiempo de ciclo para evitar errores de tiempo .....	21-16	requisitos para transferir .....	14-19
Convenciones para nombres para datos de configuración.....	15-1	transferir.....	14-19, 15-6
Copiar/mover tablas de variables.....	18-3	Datos de diagnóstico en los módulos .....	21-18
Corregir interfaces en una FC un FB o un UDT .....	9-22	Datos de estado de diagnóstico .....	21-20
Cortocircuito .....	9-16	Datos de referencia .....	12-1
CPU		aplicación.....	12-1
borrado total.....	17-18	crear .....	12-11
simular.....	20-1	sinopsis.....	12-1
Creación de huecos en la memoria de usuario (RAM) .....	17-19	visualizar .....	12-10, 12-11
		Datos de sistema .....	21-18
		DB .....	4-20
		formato.....	11-12
		DBs en fuentes AWL, ejemplos.....	11-23
		Declaración de variables en fuentes AWL ejemplos .....	11-18
		Definir, requerimientos de seguridad.....	3-8
		símbolos al introducir el programa.....	7-11
		Definir el entorno de llamada del bloque.....	19-9
		Derechos de acceso a bloques y fuentes .....	9-3
		Desactivar .....	4-23
		alarma horaria .....	4-23
		Desarchivar, procedimiento .....	22-6

Describir	
elementos necesarios para manejo y visualización.....	3-9
requerimientos de seguridad para el ejemplo de un proceso de mezcla industrial	3-8
Describir el funcionamiento en diversas áreas....	3-4
Describir el panel de manejo a la vista del ejemplo de un proceso de mezcla industrial ...	3-9
Describir las tareas y áreas individuales ejemplo de un proceso de mezcla industrial ...	3-4
Describir los elementos de manejo a la vista del ejemplo de un proceso de mezcla industrial ...	3-9
Descripción del panel de manejo .....	3-9
Desenmascarar	
eventos de arranque.....	4-31
Desfase .....	4-25, 4-26
Desinstalar	
autorización .....	2-1, 2-2, 2-3, 2-4
STEP 7 .....	2-13
Detección de errores	
Programas de ejemplo	
valores de sustitución .....	21-28
utilizar OBs de error para reaccionar a errores .....	4-31
Detección e errores, tipos de OBs, OB81 .....	21-25
Diagnóstico de sistema, ampliar .....	21-20
Diagnóstico del hardware .....	21-1
Diagrama de funciones.....	8-4
Diferencia entre guardar y cargar bloques.....	17-2
Diferencias entre el forzado normal y el forzado permanente.....	18-21
Direccionamiento	
absoluto.....	7-1
simbólico.....	7-1
Direccionamiento absoluto y simbólico.....	7-1
Direccionamiento simbólico .....	7-4
Direcciones y tipos de datos admisibles en la tabla de símbolos.....	7-8
DIS_AIRT .....	4-32
DIS_IRT .....	4-32
Diseño de página FUP .....	9-17
Diseño de página KOP .....	9-14
Disparo	
ajustar para observar variables .....	18-13
Disponer	
cuadros.....	9-18, 9-19
Disquete de autorización .....	2-1
DMSK_FLT.....	4-32
DOCPRO.....	22-2
Documentación.....	1-1, 1-4
de componentes de un proyecto	
imprimir.....	22-1
de un proyecto entero	
imprimir.....	22-1
Documentación del proyecto, imprimir .....	22-1
Download (configuración de la red).....	17-10
Editar	
en la tabla de símbolos.....	7-11
fuentes S7.....	11-14
los bloques cargados en la PG/en el PC.....	17-15
Editor, ajustes para AWL .....	9-2
Editor de lenguaje, arrancar.....	8-2
Ejecución cíclica del programa.....	4-3, 4-6
Ejecución del programa .....	4-22
cíclica.....	4-3, 4-6
controlada por alarma.....	4-3
controlada por alarmas .....	4-22
Ejemplo	
FBs en fuentes AWL.....	11-22
FCs en fuentes AWL.....	11-20
introducción de operandos en tablas de variables.....	18-8
UDTs en fuentes AWL .....	11-24
Ejemplo de introducción de un área de operandos conexos .....	18-9
Ejemplo para trabajar con puntos de aplicación.....	12-13
Ejemplos	
DBs en fuentes AWL .....	11-23
declaración de variables en fuentes AWL ...	11-18
introducción de valores de forzado normal y de forzado permanente.....	18-9
Ejemplos de programas	
ejemplo de un proceso de mezcla industrial crear un esquema de configuración.....	3-10
Describir los requerimientos de seguridad .....	3-8
descripción de las distintas tareas y áreas.....	3-4
Descripción de las tareas y áreas individuales	
Crear un diagrama de E/S .....	3-6
Panel de manejo	
describir .....	3-9
Subdividir un proceso en áreas de tareas .....	3-2
El software estándar STEP 7 .....	1-5
Elaborar .....	6-8
proyecto.....	6-8
Elegir el lenguaje de programación .....	8-2
Elegir objetos en cuadros de diálogo.....	5-24
Elementos de los cuadros de diálogo.....	5-18
Elementos de programa	
insertar.....	9-3
Elementos de visualización	
describir .....	3-9
Eliminación de errores	
programas de ejemplo .....	21-25
Emplazar	
cuadros .....	9-17
EN / ENO	
conexión.....	9-17
EN_AIRT .....	4-32
EN_IRT.....	4-32
Encabezados y pies de página .....	22-4
Enlace, establecer con la CPU.....	18-12
Enlace online .....	16-2

establecer desde la ventana "Estaciones accesibles" .....	16-1	STOP, contenido de las pilas .....	21-14
establecerlo en la ventana online .....	16-2	visualizar y cambiar .....	16-4
Enlaces online, establecer .....	16-1	Estructura	
Enmascarar, eventos de arranque .....	4-31	área de instrucciones .....	9-10
Entorno de llamada del bloque		de bloques en fuentes AWL .....	11-7
definir .....	19-9	de bloques lógicos en fuentes AWL .....	11-7
Entradas		lista de referencias cruzadas .....	12-2
listar .....	3-6	tabla de declaración de variables .....	9-5, 9-6
plano de ocupación .....	12-5	UDT .....	8-11
Entradas en cuadros de diálogo .....	5-18	ventana .....	5-18
Entradas/salidas		Estructura de bloques de datos en fuentes	
listar .....	3-6	AWL .....	11-8
Enviar .....	21-20, 21-21	Estructura de los proyectos .....	6-1
mensajes de diagnóstico personalizados .....	21-20	Estructura de proyectos .....	6-2
Enviar mensajes de diagnóstico		Estructura de tipos de datos de usuario en	
personalizados .....	21-20	fuentes AWL .....	11-8
Equipo .....	5-8	Estructura del programa .....	12-3, 12-4
cargar en PG .....	17-14	visualizar .....	12-11
Equipo PC .....	1-9	Estructura en árbol .....	12-3, 12-4
Equipos, insertar .....	6-5	Estructura jerárquica de las librerías .....	8-16
Error de comunicación (OB 87) .....	21-37	Estructura y componentes de la tabla	
Error de ejecución del programa (OB 85) .....	21-36	de símbolos .....	7-6
Error de programación (OB 121) .....	21-38	Evaluar, parámetro de salida RET_VAL .....	21-24
Error de redundancia de comunicación OB 73 .....	21-32	Evento de diagnóstico .....	21-21
Error de redundancia en CPU (OB 72) .....	21-31	Eventos .....	4-14
Error de redundancia en periferia (OB 70) .....	21-31	asíncronos .....	4-10
Error de tiempo (OB 80) .....	21-33	Eventos asíncronos .....	4-14
Errores		Eventos de arranque	
durante la instalación .....	2-9, 2-10, 2-11	enmascarar .....	4-32
evitarlos al llamar bloques .....	9-23	OBs de arranque .....	4-28
Errores asíncronos		retardar .....	4-32
OB81 .....	21-25	Evitar errores al llamar bloques .....	9-23
utilizar OBs para reaccionar a errores .....	4-31	Exportar	
Errores de sistema .....	21-22	tabla de símbolos .....	7-15
notificar .....	14-25	Fallo de alimentación (OB 81) .....	21-33
Errores del sistema .....	14-29	Fallo de CPU (OB 84) .....	21-36
Ajustes para la notificación .....	14-27	Fallo en el bastidor (OB 86) .....	21-37
Errores detectables .....	21-25	FB .....	4-15, 4-16, 4-17
Errores síncronos		corregir el interface .....	9-22
utilizar OBs para reaccionar a errores .....	4-31	DB generados .....	14-29
Esquema de configuración		formato .....	11-10
crear .....	3-10	FBs en fuentes AWL	
Esquema de contactos .....	8-4	ejemplo .....	11-22
Establecer .....	16-1	FC .....	4-15
un enlace online .....	16-1	corregir el interface .....	9-22
un enlace online desde la ventana		formato .....	11-11
"Estaciones accesibles" .....	16-1	FCs en fuentes AWL, ejemplo .....	11-20, 11-21
Establecer un enlace con la CPU .....	18-12	Fecha y hora .....	13-3
Establecer un enlace online desde la ventana		propiedad del bloque .....	13-3
online del proyecto .....	16-2	Fecha y hora como propiedad del bloque y	
Estaciones accesibles .....	16-1	conflictos de fecha y hora .....	13-3
Estado de programas		Filtrar, símbolos .....	7-12
visualización .....	19-3	Flash File System .....	2-8
Estado del programa .....	19-1	Flujo de corriente .....	9-16
Estado operativo		Formato de las FCs .....	11-11
PARADA, información importante .....	19-7	Formato de los DBs .....	11-12
		Formato de los FBs .....	11-10

Formato de los OBs .....	11-9	FUP .....	8-4, 8-5
Formato de página, ajustar .....	22-3, 22-4	visualizar información sobre el bloque .....	12-9
Formatos de archivos para importar/exportar		Generar, fuentes AWL de bloques .....	11-15
una tabla de símbolos .....	7-15	Gestionar textos en varios idiomas.....	6-9
Formatos de bloques en fuentes AWL .....	11-9	Grafo de estado .....	8-8
Forzado permanente de variables		Grandes proyectos .....	25-1
reglas de seguridad .....	18-20	GRAPH.....	8-3
Forzar variables .....	18-19	Guardar .....	22-5, 22-6
ajustar las condiciones de disparo .....	18-16	bloques .....	9-22
introducción .....	18-18	bloques de datos .....	10-8
procedimiento básico.....	18-2	bloques lógicos.....	9-22
Fuente AWL, formatos de bloques .....	11-9	ejemplos de aplicación.....	22-5
Fuentes		fuentes AWL.....	11-16
derechos de acceso .....	9-3	organización de las ventanas.....	5-26
externas.....	6-7	tabla de variables .....	18-3
Fuentes AWL		Guía de orientación .....	1-1
buscar errores.....	11-16	Guía de orientación de STEP 7.....	1-1
compilar .....	11-17	Herramientas de ingeniería.....	1-14
comprobar coherencia .....	11-16	HiGraph .....	8-3
crear .....	11-13	Hora .....	4-23, 4-24
ejemplo de FBs .....	11-22	cambiar .....	4-24
ejemplo de FCs.....	11-20	Iconos para objetos en el Administrador	
ejemplo de UDTs .....	11-24	SIMATIC .....	5-5
ejemplos de DBs .....	11-23	Identificación, de símbolos.....	7-4
ejemplos de declaración de variables.....	11-18	Idioma.....	14-16
estructura de bloques .....	11-7	Imagen del proceso.....	4-10
estructura de bloques de datos.....	11-8	actualizar.....	4-13
estructura de los bloques lógicos .....	11-7	borrar .....	4-29
estructura de tipos de datos de usuario .....	11-8	Importar, fuente externa.....	6-6
generar de bloques .....	11-15	Importar la tabla de símbolos.....	7-15
guardar .....	11-16	Impresora, instalar.....	22-2, 22-4
insertar fuentes externas .....	11-14	Imprimir	
insertar plantillas de bloques.....	11-14	bloques .....	22-1
orden de los bloques .....	11-4	componentes del proyecto .....	22-1
reglas para declarar variables.....	11-3	contenido del búfer de diagnóstico .....	22-1
reglas para definir las propiedades de		datos de referencia .....	22-1
bloques .....	11-5	documentación del proyecto .....	22-1
reglas para definir los atributos de sistema...	11-4	tabla de configuración.....	22-1
sintaxis de bloques.....	11-9	tabla de datos globales .....	22-1
Fuentes externas, insertar.....	11-14, 11-15	tabla de símbolos .....	22-1
Fuentes S7, editar.....	11-14	tabla de variables .....	22-1
Función (FC).....	4-15	Indicación de fecha y hora .....	13-3, 13-4
campo de aplicación.....	4-15	de bloques de datos de instancia .....	13-5
Función de búsqueda de errores en el		de bloques de datos globales.....	13-5
área de instrucciones .....	9-13	de bloques lógicos.....	13-4
Funcionalidad de "Notificar errores de		de UDTs y DBs derivados de UDTs .....	13-6
sistema" .....	14-25	Información del módulo .....	1-9, 21-12
Funciones.....	22-2	acceso desde la vista online del proyecto .....	21-6
Funciones (FC) .....	4-2, 4-15	funciones.....	21-9
Funciones de diagnóstico .....	21-22	llamar .....	21-9
Funciones de información.....	1-9, 21-12	visualizar .....	21-1
Funciones de información de la vista		Información del símbolo.....	7-2
de diagnóstico.....	21-8	Información importante sobre el	
Funciones de información de la vista rápida.....	21-5	estado operativo "PARADA".....	19-7
Funciones de la información del módulo .....	21-9	Inicializar	
Funciones del sistema.....	4-2, 4-21	valores de datos .....	10-8
tipos.....	4-21	Iniciar, instalación de STEP 7 .....	2-8

Insertar operandos o símbolos en una	
tabla de variables .....	18-4
plantillas de bloques en fuentes AWL.....	11-14
valores de forzado .....	18-5
valores de sustitución al detectar errores ....	21-28
Insertar equipos .....	6-5
Insertar fuentes externas .....	11-14
Instalación	
Flash File System.....	2-10
introducir nº de identificación .....	2-8
parametrización de Memory Card .....	2-8
Instalar	
STEP 7 .....	2-7, 2-8
Instalar la autorización .....	2-1
después de la instalación.....	2-1
durante la instalación.....	2-1
Instalar STEP 7.....	2-7
Instalar y desinstalar la autorización .....	2-1
Instancia.....	4-18, 4-19
Instrucciones	
introducir	
Procedimiento .....	9-10
Instrucciones AWL	
reglas de introducción .....	9-20, 11-2
Instrucciones del catálogo de elementos de	
programa .....	9-3
Interacción entre la tabla de declaración de	
variables y el área de instrucciones.....	9-5
Interconexiones no admisibles en KOP.....	9-16
Interface de usuario .....	5-17
Interface MPI .....	2-8
Interface PG/PC.....	2-12
parametrizar.....	2-11
Interfaces hombre-máquina.....	1-16
Introducción al forzado de variables.....	18-15
Introducción al forzado permanente	
de variables .....	18-18
Introducción al test con tablas de variables .....	18-1
Introducir	
estructura de bloques de datos con	
FB asociado (DBs de instancia) .....	10-4
estructura de bloques de datos con UDT	
asociado .....	10-6
estructura de bloques de datos globales .....	10-4
estructura de tipos de datos de usuario	
(UDT).....	10-6
símbolos .....	7-12
símbolos globales en un cuadro de diálogo ..	7-11
símbolos globales en un programa.....	9-12
una multiinstancia en la tabla de	
declaración de variables .....	9-9
Introducir líneas de comentario.....	18-7
Introducir nº de identificación.....	2-8
Introducir símbolos globales en la tabla	
de símbolos .....	7-12
Jerarquía .....	5-5
Jerarquía de llamada en el programa de usuario	
Jerarquía de objetos.....	5-20
crear .....	5-19, 5-20, 5-21
KOP.....	8-2, 8-3, 8-4
interconexiones no admisibles .....	9-16
Visualizar información sobre el bloque.....	12-9
Lenguaje de programación	
definir.....	8-2
elegir.....	8-2
FUP (diagrama de funciones).....	8-4
KOP (esquema de contactos) .....	8-4
S7-Graph (control secuencial).....	8-7
S7-SCL .....	8-6
Lenguaje de programación AWL	
(lista de instrucciones) .....	8-5
Lenguaje de programación S7-CFC.....	8-9
Lenguaje de programación S7-HiGraph	
(grafo de estado) .....	8-8
Lenguajes de programación.....	1-5
Librería .....	5-7
Librería estándar .....	6-6
Librerías.....	6-7
archivar .....	22-5
estructura jerárquica .....	8-16
trabajar con .....	8-15
Librerías de textos.....	14-18
Librerías estándar	
disponibles .....	8-17
licencia de uso .....	2-1
Línea de comentario.....	18-4
Líneas de comentario	
introducir .....	18-7
Lista de estado del sistema.....	21-18, 21-19, 21-20
contenido .....	21-18
leer.....	21-18, 21-19
Lista de estado del sistema (SZL) .....	21-18
Lista de instrucciones .....	8-5
Lista de referencias cruzadas .....	12-2
Listado de entradas y salidas.....	3-6
Listas de textos	
v. Listas de textos de usuario.....	14-16
Listas de textos de usuario .....	14-16
Llamadas a bloques	
actualizar.....	9-21
Llamadas de bloques .....	4-8
Llamar las funciones de ayuda.....	5-3
Localizar los fallos .....	21-1
Longitudes de bloques, visualizar .....	8-13, 8-14
M7-300/400.....	24-1, 24-2
Make (v. Comprobar coherencia del bloque) ....	13-1
Manejar.....	5-19
objetos .....	5-19, 5-20, 5-21, 5-22, 5-23
Manejo de las teclas.....	5-27
Manejo orientado a objetos.....	5-17
Manejo y visualización de variables .....	15-1
Manipulación de grandes proyectos .....	25-1
Manuales de circuitos.....	22-2
imprimir .....	22-1

Marcas, plano de ocupación.....	12-5, 12-6	Nociones básicas para programar en fuentes AWL .....	11-1
Mayúsculas y minúsculas en los símbolos .....	7-13	Nota respecto a la actualización del contenido de ventanas .....	16-3
Medidas en el programa para tratar fallos .....	21-23	Notificación .....	14-1, 14-24
Memoria de carga .....	17-3, 17-4	el sistema y su definición .....	14-1
Memoria de sesión.....	5-25	Notificar errores de sistema	
Memoria de trabajo .....	17-3	componentes compatibles.....	14-25
Memoria de trabajo virtual .....	25-4	Novedades de la versión 5.1 de STEP 7 .....	1-9
Memoria de usuario, comprimir.....	17-19	Números de mensajes, asignar.....	14-5
Memorias de carga y de trabajo, borrar.....	17-18	OB .....	4-3, 4-4, 4-5, 4-6, 4-7
Memorias de carga y de trabajo de la CPU .....	17-2	Formato .....	11-9
Memory Card, parametrizar .....	2-10	OB 72 .....	21-31, 21-32
Mensaje		OB 73 .....	21-32
de bloque.....	14-6	OB 80 .....	21-33
de símbolo .....	14-14	OB 81 .....	21-33, 21-34
ejemplo.....	14-5	OB 82 .....	21-34
mensaje de diagnóstico personalizado.....	14-15	OB 83 .....	21-35
partes .....	14-5	OB 84 .....	21-36
Mensaje de diagnóstico.....	21-21	OB 85 .....	21-36
enviar a estaciones.....	21-20	OB 86, fallo en el bastidor.....	21-37
escribir uno personalizado .....	21-20	OB 87 .....	21-37, 21-38
Mensaje SCAN, v. Mensaje de símbolo .....	14-14	OB de alarma de diagnóstico.....	21-34, 21-36
Mensajes de bloque, crear .....	14-10, 14-11	OB de alarma de presencia de módulo .....	21-35
Mensajes de CPU		OB de error .....	21-25, 21-26
configurar.....	14-22	tipos de OBs	
tamaño del archivador .....	14-20	OB121 y OB122 .....	4-31
visualizar.....	14-20, 14-21	OB70 y OB72 .....	4-31
Mensajes de diagnóstico personalizados		OB80 a OB87 .....	4-31
crear y editar .....	14-15	utilizar OBs de error para reaccionar	
visualizar.....	14-20	a eventos.....	4-31
Mensajes de símbolos		OB de error de acceso a la periferia.....	21-39
asignar a la tabla de símbolos.....	14-14	OB de error de comunicación .....	21-37
señales permitidas .....	14-14	OB de error de ejecución del programa .....	21-36
Método de creación de programas		OB de error de programación .....	21-38
seleccionar.....	8-1	OB de error de redundancia en periferia.....	21-31
Métodos para acceder a la información		OB de error de tiempo .....	21-33
del módulo .....	21-9	OB de fallo de alimentación .....	21-33
Métodos para introducir símbolos globales.....	7-10	OB de fallo de CPU .....	21-36
Modo "sobreescribir" .....	9-13	OB de tarea no prioritaria	
Módulo, simular .....	20-1	prioridad.....	4-30
Módulo de señales, simular .....	20-1	programar .....	4-30
Módulos actuales .....	1-9	Objeto	
Módulos de señales aptos para alarmas de proceso parametrizar.....	4-27	abrir .....	5-20
Mostrar, operandos sin símbolo.....	12-11	borrar .....	5-19
Motores, crear un diagrama de E/S .....	3-6	cambiar el nombre.....	5-19
Mover .....	5-22	cortar, copiar, pegar.....	5-21
objeto.....	5-19, 5-20, 5-21, 5-22, 5-23	elegir.....	5-24
MSK_FLT .....	4-32	manejar .....	5-19
Multiinstancia .....	4-18	propiedades .....	5-20, 5-21, 5-22
introducir en la tabla de declaración		Objeto "Carpeta de bloques" .....	5-12
de variables.....	9-9	Objeto "Carpeta de fuentes".....	5-14
Multiinstancias .....	4-15	Objeto "Equipo".....	5-8
reglas.....	9-9	Objeto "Librería" .....	5-7
utilización.....	9-8	Objeto "Módulo programable" .....	5-9
Nemotécnica, ajustar.....	9-20	Objeto "Programa S7/M7".....	5-11
Nociones básicas para crear bloques de datos	10-1	Objeto "Proyecto" .....	5-6

Objetos.....	5-5	atributos de sistema y bloques de	
como carpetas .....	5-5	notificación .....	14-7
como portadores de funciones.....	5-5	Partes de un mensaje.....	14-5
como portadores de propiedades.....	5-5	Particularidades al imprimir el árbol de objetos. 22-4	
Objetos y su jerarquía.....	5-5	Pasos	
OBs de alarma.....	4-22	para comprimir el contenido de la memoria de	
aplicación.....	4-22	una CPU S7 .....	17-20
deseleccionar.....	4-5	para visualizar y ajustar la fecha y la hora de la	
parametrizar.....	4-5, 4-23	CPU .....	16-5
OBs de arranque.....	4-28	para visualizar y cambiar el estado operativo 16-4	
eventos de arranque.....	4-28	Pasos de programación	
vigilancia de la configuración teórica/real		S7 .....	1-1
de los módulos .....	4-29	Pérdida de la autorización .....	2-1
OBs de error.....	14-28	Perfiles.....	1-9
OBs de error generados (Notificar errores		Permiso de acceso.....	16-3
del sistema) .....	14-28	PG/PC	
OBs de error para reaccionar a la detección		editar los bloques cargados .....	17-15
de un error .....	21-25	Pila LSTACK	
OBs en fuentes AWL		almacenamiento de variables temporales .....	4-15
ejemplo.....	11-19	Planificar una solución de automatización .....	3-1
Observaciones generales		crear un diagrama de E/S para las válvulas ....	3-7
introducir símbolos .....	7-10	crear un diagrama de E/S para los motores ....	3-6
sobre las tablas de declaración de variables... 9-7		crear un esquema de configuración.....	3-10
Observar el estado de bloques de datos .....	19-8	definir los requerimientos de seguridad .....	3-8
Observar variables .....	18-2	describir los elementos necesarios de	
ajustar el disparo .....	18-13	manejo y visualización .....	3-9
introducción .....	18-13	listado de entradas y salidas .....	3-6
procedimiento básico.....	18-2	Procedimiento básico.....	3-1
Operaciones FUP, reglas de introducción .....	9-17	Tareas y áreas, subdividir el proceso en.....	3-2
Operaciones KOP, reglas de introducción.....	9-14	Plano de ocupación, entradas	
Operandos		salidas y marcas (E/A/M) .....	12-6
insertar en una tabla de variables .....	18-4	temporizadores y contadores (T/Z).....	12-7
recablear.....	8-14	Plantilla.....	14-8, 14-9
sin símbolo.....	12-11	Plantilla y mensajes.....	14-8
Operandos no utilizados.....	12-11	Plantillas de bloques	
visualizar.....	12-11	insertar en fuentes AWL.....	11-14
Operandos sin símbolo .....	12-8	Posibilidades de ampliar el software	
visualizar.....	12-11	estándar STEP .....	1-13
Ordenar		Posibilidades de cargar en función de la	
la lista de referencias cruzadas.....	12-2, 12-3	memoria de carga.....	17-4
símbolos .....	7-12	Posicionamiento rápido en los puntos de	
Organización de las ventanas		aplicación del programa .....	12-12
guardar .....	5-26	Preajustes para el editor de programas	
restablecer .....	5-26	KOP/FUP/AWL.....	9-2
Organización de ventanas		Preferencia de operandos	
cambiar.....	5-25	ajustar .....	7-5
Outdoor .....	1-9	Prevenir daños a personas .....	18-18
Parametrizar		Prevenir daños materiales .....	18-18
módulos de señales aptos para alarmas de		Prioridad .....	4-23
proceso.....	4-27	alarma de proceso .....	4-26, 4-27
Parametrizar el interface PG/PC.....	2-11	alarma de retardo .....	4-25
Parámetro de CPU "Carga del ciclo por		alarma horaria .....	4-23
comunicaciones" .....	4-13	cambiar .....	4-5
Parámetro de salida, evaluar RET_VAL.....	21-24	OB de tarea no prioritaria.....	4-30
Parámetros, atributos.....	8-14	Procedimiento .....	24-1
Parámetros actuales .....	4-15	para archivar y desarchivar .....	22-6
Parámetros formales.....	14-7	para introducir instrucciones.....	9-10

para los sistemas M7.....	24-1	Protección con contraseña para acceder a sistemas de destino .....	16-3
Procedimiento básico		protección contra copias no autorizadas.....	2-1
para averiguar la causa de un STOP .....	21-14	Proyecto .....	5-6
para crear bloques lógicos.....	9-1	abrir .....	6-8
para imprimir.....	22-2	borrar .....	6-8
para observar y forzar variables.....	18-2	copiar.....	6-8
Procedimiento básico para planificar una solución de automatización.....	3-1	crear con el Asistente .....	6-3
Procedimiento de instalación .....	2-8	crear manualmente.....	6-3
Procedimiento de notificación		imprimir la documentación .....	22-1
selección.....	14-2	Proyectos.....	5-19
Procedimiento de notificación por bits .....	14-1, 14-2	archivar .....	22-5
Procedimiento para sistemas M7 .....	24-1	cambiar el nombre.....	5-19
Procedimiento por números de mensajes .....	14-2	orden de elaboración .....	6-3
Proceso .....	4-10, 4-12, 4-13, 4-14	reorganizar.....	25-2
subdividir .....	3-3	Proyectos con un gran número de equipos	
subdividir en áreas de tareas a la vista de un proceso de mezcla industrial.....	3-2	en red.....	25-1
Procesos de comunicaciones .....	4-10	QRY_TINT .....	4-23
PROFIBUS-DP .....	1-9	Rama T.....	9-18
Profundidad de anidamiento.....	4-8	RDSYSST .....	21-17, 21-19
Programa CFC.....	24-1	Recablear	
Programa de arranque .....	4-28	bloques .....	8-14
Programa de autorización.....	2-1	operandos .....	8-14
Programa de simulación.....	20-1	Recargar bloques en el sistema de destino .....	17-5
Programa de usuario		Recursos del enlace .....	1-9
elementos .....	4-2	Reglas	
tareas .....	4-1	alarma de proceso .....	4-26
Programa estructurado		alarma de retardo .....	4-24
Ventajas.....	4-2	orden de los bloques en fuentes AWL .....	11-4
Programa M7		para declarar multiinstancias.....	9-9
insertar .....	6-6, 6-7	para definir las propiedades de bloques en fuentes AWL.....	11-5
Programa S7, insertar .....	6-6	para definir los atributos de sistema en fuentes AWL.....	11-4
Programa S7/M7 sin equipo ni CPU.....	5-15	para el empleo de autorizaciones .....	2-4
Programación		para exportar tablas de símbolos .....	7-15
transferencia de parámetros .....	4-15	para FUP.....	9-17
utilización de bloques de datos .....	4-15	para importar tablas de símbolos .....	7-15
Programación estructurada .....	4-3	para introducir instrucciones AWL .....	9-20
Programación lineal .....	4-7	para introducir instrucciones en fuentes AWL.....	11-2
Programar		para introducir operaciones FUP .....	9-17
OB de tarea no prioritaria.....	4-30	para introducir operaciones KOP.....	9-14
Programas de ejemplo		para KOP .....	9-14
insertar valores de sustitución.....	21-28	Reglas de seguridad para el forzado permanente de variables.....	18-20
reaccionar a fallos de la pila.....	21-25	Reglas para declarar variables en fuentes AWL.....	11-3
valores de sustitución .....	21-28	Reglas para la autorización.....	2-4
Programas de una CPU .....	4-1	Regulación.....	4-23
Programas de usuario		alarma cíclica .....	4-25
cargar en el sistema de destino .....	17-4	alarma horaria .....	4-23
Propiedad del bloque .....	13-3	Reorganizar proyectos y librerías.....	25-2
fecha y hora .....	13-3	Representación	
Propiedades de bloques		elementos FUP.....	9-17
visualizar longitudes de bloques.....	8-13	elementos KOP .....	9-14
Propiedades de la carpeta de bloques		símbolos globales y locales.....	7-4
visualizar longitudes de bloques.....	8-13		
Propiedades del bloque.....	8-13, 9-1		
Propiedades válidas para los diferentes tipos de bloques .....	11-6		

requerimiento de datos locales .....	12-3, 12-5	definir al introducir el programa .....	7-11
Requerimientos de seguridad		en la estructura del programa.....	12-3
descripción para el ejemplo de un proceso de		filtrar.....	7-12
mezcla industrial.....	3-8	globales .....	7-3
Requisitos		insertar en una tabla de variables.....	18-4
para cargar .....	17-1	introducir .....	7-12
para poder archivar .....	22-6	locales.....	7-3
Requisitos de instalación.....	2-7	mayúsculas y minúsculas .....	7-13
Restablecer		no utilizados .....	12-8
autorización .....	2-1	ordenar .....	7-12
organización de las ventanas.....	5-26	Símbolos de diagnóstico	
Retardar, eventos de arranque .....	4-32	en la vista online.....	21-3
RPL_VAL.....	21-28, 21-29	Símbolos de objetos en el Administrador	
S7-CFC .....	8-9	SIMATIC .....	5-5
S7-GRAPH .....	8-7	Símbolos globales	
S7-HiGraph.....	8-8	introducir en la tabla de símbolos .....	7-12
S7-SCL.....	8-6	introducir en un cuadro de diálogo.....	7-11
Salidas, listar .....	3-6	introducir en un programa .....	9-12
plano de ocupación .....	12-5	Símbolos globales y locales.....	7-3
SCL.....	8-3	Símbolos incompletos y ambiguos en la	
Segmentos .....	8-4	tabla de símbolos .....	7-9
finalización en KOP .....	9-14	Símbolos no utilizados.....	12-8
Selección del procedimiento de notificación .....	14-2	Simulación de una CPU o de un módulo	
Seleccionar		de señales.....	20-1
método de creación de programas .....	8-1	Sinopsis	
Servidor de red .....	23-1	Bloques de notificación .....	14-6
SET_CLK .....	4-24	de los datos de referencia .....	12-1
SET_TINT .....	4-23, 4-24	Sintaxis de bloques en fuentes AWL .....	11-9
SFB .....	4-21	Sistema de destino.....	17-5
SFB 20 STOP.....	4-10	cargar bloques modificados .....	17-5
SFB 33 .....	14-7	Sistema de notificación.....	14-1
SFB 34 .....	14-7	Sistema operativo, tareas .....	4-1
SFB 35 .....	14-6, 14-7	Sistema operativo de la CPU.....	4-14
SFB 36 .....	14-6, 14-7	Sistemas DP maestros .....	1-9
SFB 3767 .....	14-6	Sistemas operativos para M7-300/400 .....	24-6
SFC.....	4-21	Software estándar .....	1-5
SFC 0 SET_CLK.....	4-23	software opcional .....	20-1
SFC 17/18 .....	14-6	Software opcional.....	24-3, 24-4, 24-5
SFC 26 UPDAT_PI .....	4-10	Software opcional para la programación M7 .....	24-3
SFC 27 UPDAT_PO.....	4-13	Software Runtime.....	1-15
SFC 28 SET_TINT .....	4-23	Solución de automatización	
SFC 29 CAN_TINT .....	4-23	planificar	
SFC 30 ACT_TINT.....	4-23	crear un diagrama de E/S para las válvulas ....	3-7
SFC 31 QRY_TINT .....	4-23	crear un diagrama de E/S para los motores ....	3-6
SFC 36 MSK_FLT.....	4-31	crear un esquema de configuración.....	3-10
SFC 37 DMSK_FLT .....	4-31	definir los requerimientos de seguridad .....	3-8
SFC 39 DIS_IRT .....	4-31	planificar	
SFC 40 EN_IRT .....	4-31	describir las áreas de funcionamiento.....	3-4
SFC 41 DIS_AIRT.....	4-31	elementos de manejo y visualización	
SFC 42 EN_AIRT.....	4-31	describir .....	3-9
SFC 44 RPL_VAL .....	21-28	listar entradas y salidas .....	3-6
SFC 46 STP .....	4-11	proceso, subdividir en áreas y tareas .....	3-2
SFC 51 RDSYSST.....	21-17, 21-18	SRT_DINT .....	4-24
SFC 52 WR_USMSG .....	21-20	STEP 7 .....	1-5
Signo de comentario .....	18-4	desinstalar.....	2-13
SIMATIC PC .....	1-9	errores durante la instalación .....	2-8
Símbolos .....	7-4, 7-13, 7-14	iniciar el software.....	5-1

Instalación.....	2-7, 2-8	Tiempo de ciclo .....	4-10
interface de usuario .....	5-17	Tiempo de ciclo de datos .....	1-9
lenguajes de programación.....	1-5, 1-7	Tiempo de ciclo de OB 1 .....	4-10
OBs de error		Tiempo de ciclo máximo .....	4-12
reaccionar a errores .....	4-31	Tiempo de ciclo mínimo.....	4-12, 4-14
software estándar.....	1-5, 1-6, 1-7	Tiempo de vigilancia de ciclo .....	4-10
STOP, averiguar la causa.....	21-14	Tiempos de vigilancia .....	4-29
Subdividir el proceso en tareas y áreas.....	3-2	Tipo de datos, UDT .....	8-11
Subdividir un proceso en áreas de tareas a		Tipo de declaración, cambiar .....	9-6
la vista de un proceso de mezcla industrial.....	3-2	Tipos de alarmas.....	4-3
Sustituir módulos .....	25-1	Tipos de datos, FB, SFB.....	4-15
Sustituir módulos en la tabla de configuración..	25-1	Tipos de datos de usuario	
SZL .....	21-18	introducir la estructura .....	10-6
Tabla de declaración		Tipos de datos de usuario (UDT) .....	8-11
introducir una multiinstancia.....	9-9	Títulos de bloques .....	9-12
Tabla de declaración de variables.....	9-1, 9-4, 9-5	de segmentos.....	9-12
atributos de sistema para parámetros .....	9-4	Títulos de bloques .....	9-13
estructura.....	9-5	Títulos de segmentos .....	9-12
introducir una multiinstancia.....	9-9	Trabajar	
tarea .....	9-4	con librerías.....	8-15
Tabla de declaración de variables para		Traducir librerías de textos.....	14-18
el OB81.....	21-25	Traducir y editar	
Tabla de enlaces.....	1-9	textos de usuario .....	14-16
Tabla de símbolos.....	7-4	Traducir y editar textos de usuario .....	14-16
abrir .....	7-12	Transferencia de parámetros	
configurar atributos M+V.....	15-4	almacenamiento de los valores transferidos .	4-15
direcciones admisibles.....	7-8	Transferir los datos de configuración al	
estructura y componentes.....	7-6	sistema de destino.....	14-19
formatos de archivos para importar/exportar.	7-15	Transferir los datos de configuración al	
para símbolos globales.....	7-6	sistema de destino M+V.....	15-6
tipos de datos admisibles.....	7-8	Transmisión de informaciones de	
Tabla de variables.....	18-3	diagnóstico.....	21-17
comprobación de sintaxis .....	18-5	Tratamiento de fallos .....	21-23
copiar/mover .....	18-3	TTR, tipo.....	1-9
editar .....	18-4	UDT .....	8-11
ejemplo.....	18-4	corregir el interface .....	9-22
ejemplo de introducción de operandos .....	18-8	UDTs, introducir la estructura .....	10-6
guardar .....	18-1, 18-3	UDTs en fuentes AWL, ejemplo .....	11-24
insertar operandos o símbolos .....	18-4	UPDAT_PI .....	4-13
tamaño máximo.....	18-5	UPDAT_PO .....	4-10
uso .....	18-1	Upload (cargar una configuración de red	
Tarjeta MPI en la PG/el PC .....	2-11	en la PG).....	17-16
Tarjeta MPI-ISA (Auto) .....	2-11	Utilización de la declaración de variables	
Temporizadores		en los bloques lógicos.....	9-4
plano de ocupación .....	12-7	Utilización de las funciones	
valores máximos de introducción .....	18-5	'Guardar / Archivar' .....	22-5
Test.....	19-1, 19-2	Utilización de multiinstancias .....	9-8
con el estado del programa.....	19-1	Valores de datos	
con el programa de simulación		cambiar en la vista "Datos" de los bloques de	
(software opcional) .....	20-1	datos .....	10-7
Test con el estado del programa.....	19-1	inicializar .....	10-8
Test con la tabla de variables .....	25-2	Valores de forzado, insertar .....	18-5
Test de programas con la tabla de variables ....	18-1	Valores de forzado normal	
Textos de usuario		ejemplos de introducción .....	18-9
exportar/importar .....	14-16	Valores de forzado permanente	
requisitos .....	14-16	ejemplos de introducción .....	18-9
v. Textos de usuario .....	14-16	Valores de sustitución	

utilización de SFC44 (RPL_VAL) .....	21-28	Visualización del estado de programas .....	19-3
Valores máximos para introducir contadores....	18-6	Visualizar	
Valores máximos para introducir		datos de referencia .....	12-10, 12-11, 12-12
temporizadores .....	18-5	estado operativo .....	16-4
Válvulas, crear un diagrama de E/S .....	3-7	estructura de bloques de datos con	
Variables .....	18-15	FB asociado (DBs de instancia).....	10-4
forzar .....	18-15	estructura de bloques de datos con UDT	
manejo y visualización.....	15-1	asociado.....	10-6
observar.....	18-13	estructura del programa .....	12-11
Ventana del proyecto .....	6-1	información del bloque en KOP, FUP, AWL..	12-9
Ventanas, cambiar de una a otra .....	5-31	listas en ventanas de trabajo adicionales....	12-11
Ver datos de los bloques de datos .....	10-3	longitudes de bloques .....	8-13
Ver declaración de bloques de datos .....	10-2	operandos no utilizados .....	12-11
Vigilancia de la configuración teórica/real		operandos sin símbolo.....	12-11
de los módulos, OBs de arranque .....	4-28	Visualizar la información del módulo .....	21-1
Vigilancia del proceso .....	18-2	Visualizar las estaciones accesibles.....	16-1
Vista del proyecto .....	6-1	Visualizar longitudes de bloques .....	8-13
Vista online, símbolos de diagnóstico .....	21-3, 21-4	Visualizar los mensajes de CPU guardados ...	14-22
Visualización		Visualizar mensajes de CPU y mensajes de	
activar los símbolos en el bloque .....	7-11	diagnóstico personalizados .....	14-20
bloque borrado .....	12-3	Volumen de información del módulo en	
estructura en árbol .....	12-3	función del tipo de módulo .....	21-12
estructura por padres e hijos.....	12-3, 12-5	WR_USMSG.....	21-20, 21-21
requerimiento máximo de datos locales			
en la estructura en árbol .....	12-3		

Siemens AG  
A&D AS E 81  
D-Oestliche Rheinbrueckenstr. 50  
76181 Karlsruhe  
R.F.A.

Remitente:

Nombre: .....

Cargo:.....

Empresa:.....

Calle: .....

Código postal: .....

Población:.....

País:

Teléfono: .....

Indique el ramo de la industria al que pertenece:

- |  |   |
|--|---|
| <input type="checkbox"/> Industria del automóvil   | <input type="checkbox"/> Industria farmacéutica |
| <input type="checkbox"/> Industria química         | <input type="checkbox"/> Industria del plástico |
| <input type="checkbox"/> Industria eléctrica       | <input type="checkbox"/> Industria papelera     |
| <input type="checkbox"/> Industria alimentaria     | <input type="checkbox"/> Industria textil       |
| <input type="checkbox"/> Control e instrumentación | <input type="checkbox"/> Transportes            |
| <input type="checkbox"/> Industria mecánica        | <input type="checkbox"/> Otros.....             |
| <input type="checkbox"/> Industria petroquímica    |   |

Observaciones/sugerencias

Sus observaciones y sugerencias nos permiten mejorar la calidad y utilidad de nuestra documentación. Por ello le rogamos que rellene el presente formulario y lo envíe a Siemens.

Responda por favor a las siguientes preguntas dando una puntuación comprendida entre 1 = muy bien y 5 = muy mal

- 1. ¿ Corresponde el contenido del manual a sus exigencias ?
- 2. ¿ Resulta fácil localizar las informaciones requeridas ?
- 3. ¿ Es comprensible el texto ?
- 4. ¿ Corresponde el nivel de los detalles técnicos a sus exigencias ?
- 5. ¿ Qué opina de la calidad de las ilustraciones y tablas ?

En las líneas siguientes puede exponer los problemas concretos que se le hayan planteado al manejar el manual:

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....