

**UNIVERSIDAD CATÓLICA DE SANTA MARÍA**  
**FACULTAD DE CIENCIAS E INGENIERÍAS FÍSICAS Y FORMALES**  
**PROGRAMA PROFESIONAL DE INGENIERÍA MECÁNICA,**  
**MECÁNICA - ELÉCTRICA Y MECATRÓNICA**



**“CONTROL E IMPLEMENTACIÓN DE UN MÓDULO DE  
ALMACENAJE VERTICAL CON PLC Y SCADA PARA LA  
OPTIMIZACIÓN DE UN PROCESO DE ALMACENAJE INDUSTRIAL”**

Tesis presentada por los Bachilleres:

**BARREDA CORDOVA, Eder Carlos**  
**OPORTO FIGUEROA, Davis Cesar**

Para optar el Título Profesional de  
**INGENIERO MECATRÓNICO**

AREQUIPA – PERÚ  
**2014**



## **AGRADECIMIENTOS:**

Agradecemos el apoyo de nuestros padres, amigos, compañeros, colegas y seres queridos que nos dieron la perseverancia y apoyo para culminar este trabajo.



## ÍNDICE

<b>RESUMEN .....</b>	<b>VIII</b>
<b>ABSTRACT.....</b>	<b>IX</b>
<b>INTRODUCCION.....</b>	<b>X</b>
<b>Capítulo1: GENERALIDADES.....</b>	<b>12</b>
1.1. Nombre de la tesis .....	12
1.2. Antecedentes.....	12
1.3. Justificación.....	13
1.4. Objetivos.....	15
1.4.1. Objetivo General.....	15
1.4.2. Objetivos Específicos.....	15
1.5. Alcances y limitaciones .....	16
<b>Capítulo 2: FUNDAMENTO TEÓRICO.....</b>	<b>18</b>
2.1 Almacenes.....	18
2.1.1 Consideraciones para el diseño de un almacen .....	19
2.1.2 Unidad de carga.....	19
2.1.3 Equipo de almacenamiento.....	21
2.1.4 Tipos de almacenes.....	22
2.1.5 Operaciones fundamentales .....	24
2.1.6 Métodos de almacenamiento.....	24
2.2 Automatización.....	26
2.2.1 Concepto de Automatización.....	26
2.2.2 Objetivos de la Automatización.....	27
2.2.3 Teoría de control.....	27
2.2.4 Control Automático.....	29
2.2.5 Manufactura integrada por computadora.....	30
2.2.6 Razones para la automatización .....	31
2.2.7 Argumentos a favor y en contra de la automatización.....	31
2.3 Sistemas para la automatización.....	32
2.3.1 Sistemas de automatizacion mecanica .....	32
2.3.2 Sistemas de automatizacion neumatico.....	32
2.3.3 Sistemas de automatizacion hidraulica .....	32
2.3.4 Sistemas de automatizacion electronica y electrica .....	32
2.4 La Pirámide de Automatización.....	36



2.5	Sistemas Mecatrónicos .....	39
2.6	Conceptos fundamentales sobre las comunicaciones digitales .....	41
2.6.1	Concepto de Red (Net-Working).....	41
2.6.2	Topología de redes.....	41
2.6.3	Interfaces de conectividad en las comunicaciones industriales.....	44
2.6.4	El modelo ISO/OSI.....	45
2.6.5	Ethernet.....	49
2.6.6	TCP/IP.....	50
2.6.7	Puertos TCP/IP.....	52
2.7	Controlador industrial de procesos.....	53
2.7.1	Áreas de aplicación del Controlador Lógico Programable (PLC).....	54
2.7.2	Características destacadas de un PLC.....	56
2.7.3	Ventajas.....	57
2.7.4	Definición básica de un PLC.....	57
2.7.5	Estructura de un PLC.....	58
2.7.6	Modo de funcionamiento de un PLC.....	59
2.7.7	Dispositivo programador de un PLC.....	61
2.7.8	PLC SIEMENS S7-200.....	62
2.7.9	Puesta a punto de un PLC.....	63
2.7.10	Seguridad funcional de un PLC.....	65
2.7.11	Lenguajes de programación para PLC según IEC 61131-3.....	68
2.8	Sistemas SCADA .....	71
2.8.1	Características De Un Sistema SCADA .....	73
2.8.2	Funciones Principales Del Sistema SCADA.....	74
2.8.3	Pantallas HMI .....	75
2.8.4	Características HMI.....	76
2.8.5	Funciones De Un Software HMI.....	77
2.8.6	Estructura General De Un Software HMI.....	78
2.8.7	Tendencias.....	80
<b>Capítulo 3: DESCRIPCIÓN DEL MÓDULO .....</b>		<b>81</b>
3.1.	Descripción funcional del módulo .....	81
3.1.1.	Matriz de almacenamiento:.....	82
3.1.2.	Movimiento vertical:.....	83
3.1.3.	Movimiento horizontal:.....	83
3.1.4.	Modos de operacion:.....	84
3.1.4.1.	Modo manual:.....	84
3.1.4.1.	Modo automático: .....	85
3.2.	Descripción del Sistema Mecatrónico por subsistemas.....	85
3.2.1.	Subsistema Neumático.....	85



3.2.2. Subsistema Eléctrico - Electrónico.....	92
3.2.3. Subsistema Estructural – Transmisión.....	108
<b>Capítulo 4: CONFIGURACIÓN Y PROGRAMACIÓN .....</b>	<b>113</b>
4.1. Configuración del PLC.....	113
4.1.1. Hardware y software necesarios para la programación en el entorno STEP 7.....	113
4.1.2. Elementos de programación.....	114
4.1.3. Lenguaje de programación.....	116
4.1.4. Lista de asignación del programa.....	118
4.1.5. Carga y ejecución de programa.....	121
4.1.6. Diagramas de flujo.....	124
4.2. Diseño del HMI.....	128
4.2.1. Definir parámetros iniciales de MGCS:.....	129
4.2.2. Programación en la Pantalla hmi.....	132
4.3. Instructivo de funcionamiento módulo didáctico.....	136
4.3.1. Modo Automático:.....	136
4.3.2. Modo Manual:.....	137
<b>Capítulo 5: ANALISIS DE COSTOS.....</b>	<b>140</b>
5.1. Materiales para la elaboración del módulo.....	140
5.2. Costos de ingeniería.....	142
<b>CONCLUSIONES Y RECOMENDACIONES .....</b>	<b>143</b>
Conclusiones.....	143
Recomendaciones.....	145
<b>BIBLIOGRAFÍA.....</b>	<b>147</b>

## ÍNDICE DE FIGURAS

<i>Figura 1 : Almacenamiento Vertical de Productos.....</i>	<i>13</i>
<i>Figura 2: Sistema Automatizado de Almacenamiento.....</i>	<i>14</i>
<i>Figura 3 : Diagrama de bloques de un sistema automatizado .....</i>	<i>30</i>
<i>Figura 4 : Pirámide de automatización según la operatividad en la fábrica.....</i>	<i>37</i>
<i>Figura 5 : Sinergia de la Mecatrónica.....</i>	<i>40</i>
<i>Figura 6 : Conexionado de una red configurada en topología estrella .....</i>	<i>42</i>
<i>Figura 7 : Conexionado de una red configurada en topología anillo.....</i>	<i>43</i>
<i>Figura 8 : Conexionado de una red configurada en topología bus.....</i>	<i>44</i>
<i>Figura 9 : Conector DB-9 como interfaz RS-232 .....</i>	<i>45</i>
<i>Figura 10 : Conector RJ-45 .....</i>	<i>45</i>
<i>Figura 11 : Modelo de referencia ISO/OSI.....</i>	<i>46</i>
<i>Figura 12: PLC industrial con terminales electro neumáticos MPA, en una aplicación industrial.....</i>	<i>55</i>
<i>Figura 13 : Estructura interna clásica de un PLC .....</i>	<i>58</i>
<i>Figura 14 : Ciclo de funcionamiento de un PLC .....</i>	<i>59</i>
<i>Figura 15 : Ejemplo de lenguaje de LD.....</i>	<i>69</i>
<i>Figura 16 : Ejemplo de lenguaje de FBD.....</i>	<i>69</i>
<i>Figura 17 : Ejemplo de lenguaje de IL.....</i>	<i>70</i>
<i>Figura 18 : Ejemplo de lenguaje de ST .....</i>	<i>70</i>
<i>Figura 19 : Ejemplo de lenguaje de SF.....</i>	<i>71</i>
<i>Figura 20 : Estructura de un Sistema HMI.....</i>	<i>79</i>
<i>Figura 21 : Vista general del módulo implementado .....</i>	<i>81</i>
<i>Figura 22: Diagrama de flujo Módulo Implementado .....</i>	<i>82</i>
<i>Figura 23: Matriz de Almacenamiento.....</i>	<i>82</i>
<i>Figura 24: Diagrama de Flujo Sistema Neumático .....</i>	<i>86</i>
<i>Figura 25: Alimentador de aire comprimido .....</i>	<i>86</i>
<i>Figura 26 : Diagrama de conexión neumática.....</i>	<i>87</i>
<i>Figura 27 : Filtro regulador de aire .....</i>	<i>88</i>
<i>Figura 28 : Designación por código de Filtro regulador de aire .....</i>	<i>88</i>
<i>Figura 29 : Cilindro Neumático .....</i>	<i>89</i>
<i>Figura 30 : Designación de Cilindro Neumático.....</i>	<i>89</i>
<i>Figura 31 : Monograma Fuerza- Presión.....</i>	<i>90</i>



<i>Figura 32 : Regulación del cilindro .....</i>	<i>91</i>
<i>Figura 33 : Electro-válvulas.....</i>	<i>92</i>
<i>Figura 34 : Diagrama de flujo Sistema Eléctrico.....</i>	<i>93</i>
<i>Figura 35 : Motor DC.....</i>	<i>94</i>
<i>Figura 36 : Relés usados en el módulo y diagrama esquemático de funcionamiento.....</i>	<i>97</i>
<i>Figura 37 : Distribución de componentes del PLC.....</i>	<i>99</i>
<i>Figura 38 : Configuración Tapa Superior.....</i>	<i>99</i>
<i>Figura 39 : Configuración Puente de Alimentación.....</i>	<i>99</i>
<i>Figura 40 : Tensión de las cargas .....</i>	<i>100</i>
<i>Figura 41 : Configuración Tapa Inferior.....</i>	<i>101</i>
<i>Figura 42 : Alimentación de las entradas .....</i>	<i>102</i>
<i>Figura 43: Configuración Tapa Frontal.....</i>	<i>103</i>
<i>Figura 44 : Módulo de Ampliación.....</i>	<i>104</i>
<i>Figura 45 : Conexión PC – CPU 224.....</i>	<i>105</i>
<i>Figura 46 : HMI MCGS modelo TPC 7062.....</i>	<i>106</i>
<i>Figura 47 : Dispositivos de mando.....</i>	<i>107</i>
<i>Figura 48 : Sensor magnético para la detección no invasiva del cilindro neumático .....</i>	<i>108</i>
<i>Figura 49 : Sensor de Final de Carrera.....</i>	<i>108</i>
<i>Figura 50 : Diagrama de flujo Sistema Estructural .....</i>	<i>109</i>
<i>Figura 51 : Unidad de Almacenamiento.....</i>	<i>110</i>
<i>Figura 52 : Unidad de desplazamiento .....</i>	<i>110</i>
<i>Figura 53 : Eje roscado y ejes guía del carrito feeding.....</i>	<i>111</i>
<i>Figura 54 : Eje roscado y ejes guía del carrito feeding.....</i>	<i>112</i>
<i>Figura 55 : Diagrama de conexión de PLC S7-200 y módulo de E/S.....</i>	<i>113</i>
<i>Figura 56 : Instrucciones lógicas.....</i>	<i>115</i>
<i>Figura 57 : Esquema de contactos KOP.....</i>	<i>116</i>
<i>Figura 58 : Diagrama de funciones FUP.....</i>	<i>117</i>
<i>Figura 59 : Lista de instrucciones AWL.....</i>	<i>117</i>
<i>Figura 60 : Lista de opciones.....</i>	<i>121</i>
<i>Figura 61 : Selección de CPU.....</i>	<i>121</i>
<i>Figura 62 : Bloques de programa.....</i>	<i>121</i>
<i>Figura 63 : Apertura de archivo en step 7.....</i>	<i>122</i>
<i>Figura 64 : Carga del CPU- PLC .....</i>	<i>122</i>
<i>Figura 65 : Selección de Puerto de comunicación PPI.....</i>	<i>123</i>



<i>Figura 66 : Se carga y corre el Programa.....</i>	<i>123</i>
<i>Figura 67 : Simbología de diagramas de flujo. ....</i>	<i>124</i>
<i>Figura 68 : Diagrama de flujo programa 1 / secuencia principal. ....</i>	<i>125</i>
<i>Figura 69 : Diagrama de flujo programa 2 / secuencia Automático. ....</i>	<i>126</i>
<i>Figura 70 : Diagrama de flujo programa 3 / secuencia Manual.....</i>	<i>127</i>
<i>Figura 71 : Touch screen - HMI.....</i>	<i>128</i>
<i>Figura 72 : Características técnicas de la Pantalla.....</i>	<i>128</i>
<i>Figura 73 : Entorno MGCS, inicio de programa.....</i>	<i>129</i>
<i>Figura 74 : MGCS-selección de PLC y medio de comunicación .....</i>	<i>130</i>
<i>Figura 75 : MGCS-Creación de Ventanas de Trabajo.....</i>	<i>130</i>
<i>Figura 76 : MGCS-Definición de señales de ingreso.....</i>	<i>131</i>
<i>Figura 77 : MGCS – Selección de estrategia de programa.....</i>	<i>131</i>
<i>Figura 78 : MGCS- Ventana de inicio.....</i>	<i>132</i>
<i>Figura 79 : MGCS – Programación botón de automático.....</i>	<i>133</i>
<i>Figura 80 : MGCS – Programación botón de Manual .....</i>	<i>133</i>
<i>Figura 81: MGCS- Ventana de Secuencia Automática.....</i>	<i>134</i>
<i>Figura 82 : MGCS- Ventana de Manual.....</i>	<i>135</i>
<i>Figura 83 : HMI – Pantalla de inicio – Selecciona Automático.....</i>	<i>136</i>
<i>Figura 84 : HMI – Pantalla secuencia Automático.....</i>	<i>137</i>
<i>Figura 85 : HMI – Pantalla de inicio – Selecciona Manual.....</i>	<i>138</i>
<i>Figura 86 : HMI – Pantalla secuencia Manual.....</i>	<i>138</i>
<i>Figura 87 : HMI – Pantalla secuencia Manual – Indicadores de Llenado de Piezas.....</i>	<i>139</i>



## ÍNDICE DE TABLAS

<i>Tabla 1 : Características principales de red configurada en topología estrella.....</i>	<i>41</i>
<i>Tabla 2 : Características principales de red configurada en topología anillo.....</i>	<i>42</i>
<i>Tabla 3 : Características principales de red configurada en topología bus.....</i>	<i>43</i>
<i>Tabla 4: Designación de puertos TCP/IP reservados para aplicaciones.....</i>	<i>52</i>
<i>Tabla 5 : Consumo eléctrico de fuente de alimentación.....</i>	<i>95</i>
<i>Tabla 6 : Datos técnicos del Disyuntor Termo-magnético.....</i>	<i>96</i>
<i>Tabla 7 : Características de Relé.....</i>	<i>97</i>
<i>Tabla 8 : Datos técnicos de los relés del módulo.....</i>	<i>97</i>
<i>Tabla 9 : Datos técnicos del PLC.....</i>	<i>98</i>
<i>Tabla 10 : Variables de Salida.....</i>	<i>101</i>
<i>Tabla 11 : Variables de entrada.....</i>	<i>102</i>
<i>Tabla 12 : Configuración de Módulos de ampliación.....</i>	<i>104</i>
<i>Tabla 13 : Características HMI.....</i>	<i>106</i>
<i>Tabla 14 : Lista de Asignación de Entradas en el PLC, según planos eléctricos.....</i>	<i>118</i>
<i>Tabla 15 : Lista de Asignación de salida en el PLC, según planos eléctricos.....</i>	<i>118</i>
<i>Tabla 16 : Lista de Asignación de memorias en el PLC.....</i>	<i>120</i>
<i>Tabla 17 : Lista de Asignación de Subrutinas en el PLC.....</i>	<i>120</i>
<i>Tabla 18 : Tabla de Costos de Materiales Estructurales.....</i>	<i>140</i>
<i>Tabla 19 : Tabla de Costos de Materiales Eléctricos – Neumáticos.....</i>	<i>141</i>
<i>Tabla 20 : Tabla de Costos total de materiales.....</i>	<i>141</i>
<i>Tabla 21 : Tabla de Costos de Ingeniería.....</i>	<i>142</i>



## RESUMEN

El presente trabajo de “CONTROL E IMPLEMENTACIÓN DE UN MÓDULO DE ALMACENAJE VERTICAL CON PLC Y SCADA PARA LA OPTIMIZACIÓN DE UN PROCESO DE ALMACENAJE INDUSTRIAL” realiza la función de almacenamiento automatizado de objetos en una ubicación especificada y comandada desde un centro de control, este sistema automatizado de almacenamiento ha sido utilizado durante muchos años en la industria de muchos países desarrollados.

El módulo implementado tiene dos modalidades de función: modo manual para que el usuario designe en cuál de las 9 posiciones ha de almacenar el objeto deseado y modo automático para completar los materiales en la totalidad de lugares de almacenaje de manera automática.

El equipo está compuesto por tres niveles de almacenamiento cada uno cuenta de tres ubicaciones de almacenaje, con dispositivos de control eléctrico de movimiento en tres dimensiones que logran que los despachadores coloquen en la posición deseada los objetos almacenar, todos estos programado y controlado por un PLC Siemens y toda la secuencia de trabajo se realiza por un sistema SCADA para un fácil control y monitoreo de las funciones a ser realizadas por el usuario.

El diseño del módulo permite reforzar los fundamentos:

- Tener una apreciación del funcionamiento de los componentes mecánicos, Con actuadores neumáticos, transmisión de potencia, transportador de faja, etc.
- Desarrollar la programación a través de un PLC industrial y el desarrollo de la supervisión y adquisición de datos (SCADA) utilizados, a través de un HMI, trabajo que mejoro las habilidades en la programación de PLC y SCADA de los participante en la elaboración del módulo.

El módulo cumple con la meta propuesta de optimizar espacio, tiempo y recursos humanos con la implementación del sistema automatizado de almacenaje.



## ABSTRACT

The present work "CONTROL AND IMPLEMENTATION OF A VERTICAL STORAGE MODULE WITH PLC AND SCADA SYSTEM FOR THE OPTIMIZATION OF INDUSTRIAL PROCESS STORAGE" performs the function of automated storage of objects in a specified location which is order from a control center; this automated storage system has been used for many years in the industry in many developed countries.

The implemented module has two operating modes: manual mode for the user to designate which of the 9 position is to store the desired object and automatically mode in order to complete the materials in all storage locations automatically.

The equipment is composed of three tiers of storage each of those ones has three storage locations, is also compose with electrical controls movement in three dimensions that make dispatchers placed in the desired position objects store all these programmed and controlled by a PLC Siemens sequence and all the work is done by a SCADA system for easy control and monitoring functions to be performed by the user.

The module design allows reinforcing the fundamentals:

- Have an appreciation of the operation of mechanical components, with pneumatic actuators, power transmission, conveyor belt, etc.
- Develop programming through an industrial PLC and the development of supervision and data acquisition (SCADA) used via an HMI, I improve job skills in PLC programming and SCADA.

The module complies with the proposed goal of optimizing space, time and human resources to the implementation of the automated storage system.



## INTRODUCCION

El presente trabajo de tesis, se basa en el desarrollo e implementación de un Sistema Mecatrónico cuyo objetivo es optimizar y dar una solución automatizada para el almacenaje (de materiales, equipos y otros) de manera didáctica, usando los componentes más cercanos al ambiente industrial. Este sistema es controlado por un Controlador Industrial (PLC Siemens), dentro de una estructura de transportador, supervisado y controlado mediante un HMI (Interfaz Hombre Maquina) en un entorno con pantalla Táctil, como controlador SCADA,

La aplicación e integración de equipos de supervisión SCADA con la tecnología Touch screen significa un desarrollo de las implementaciones acorde los avances de la tecnología, se da mediante la integración de máquinas a procesos o administración en producción industrial, mejorando sus procesos de Almacenaje la cual es una actividad critica en todo proceso industrial.

El documento está dividido en 5 capítulos, el CAPITULO 1 denominado "Generalidades", detallamos la aplicación y la justificación del desarrollo del módulo. El CAPITULO 2 denominado "Marco teórico" donde se recopila conceptos básicos y precisos que están relacionados a la implementación del módulo, controladores y conceptos de SCADA. El CAPITULO 3 lleva por título "Descripción del módulo", este capítulo hace un análisis del funcionamiento del módulo dividiéndolo en subsistemas, el análisis de los distintos elementos estructurales y mecanismos así como de los periféricos eléctricos y características del controlador.

En el CAPITULO 4 "Configuración del controlador y diseño del HMI", se desarrolla los procedimientos, configuraciones básicas y programación del controlador un PLC de la marca Siemens que tiene como entorno al Step 7. El enlace HMI entre la pantalla Touch y el controlador se establece mediante el software MCGSE (Monitor and Control Generated System for Embedded). En este capítulo también se describe el procedimiento para diseñar una Interface Hombre Maquina (HMI) usando una serie de comandos y opciones de programación Basic que abren y cierran la comunicación. El CAPITULO 5 hace referencia al "Análisis de Costos", donde se hace un balance económico de los costó que el módulo ha generado.



Por último se incluye un apartado de las Conclusiones y de las Recomendaciones, la Bibliografía y los Anexos, en este último están planos estructurales, eléctricos, y hojas de datos (Datasheets) de los distintos dispositivos usados en el módulo y el MOM (Manual de Operación y Mantenimiento).





## CAPÍTULO1: GENERALIDADES

En este capítulo se muestran los principales parámetros que conllevaron a la ejecución de este trabajo de tesis de implementación y desarrollo tecnológico.

### 1.1. NOMBRE DE LA TESIS

El presente trabajo de tesis lleva por título:

**“CONTROL E IMPLEMENTACIÓN DE UN MÓDULO DE ALMACENAJE VERTICAL CON PLC Y SCADA PARA LA OPTIMIZACIÓN DE UN PROCESO DE ALMACENAJE INDUSTRIAL”**

### 1.2. ANTECEDENTES

El trabajo de tesis **“Control e implementación de un módulo de almacenaje vertical con PLC y SCADA para la optimización de un proceso de almacenaje industrial”** es el primer módulo que abarca la automatización de un proceso de almacenaje industrial en la lista de tesis que se han presentado hasta el día de hoy por nuestros compañeros ya graduados del Programa Profesional de Ingeniería Mecánica-Eléctrica y Mecatrónica en nuestra Universidad Católica de Santa María, el cual servirá como guía y base para trabajos de tesis futuros, que deseen profundizar el campo de la logística automatizada.

Las primeras actividades de almacenaje que se realizaban eran de forma artesanal e involucraba mayor cantidad de personal, tiempo y espacio; estas fueron evolucionando con los avances tecnológicos hasta cierto grado de automatización. Estos avances se han desarrollado especialmente en la industria donde la eficiencia de los almacenes repercute económicamente y en el soporte a la producción.

La búsqueda de soluciones automatizadas a estos procesos de almacenaje solo puede ser desarrollada mediante la simulación en el cual intervengan áreas multidisciplinarias que hagan uso de tecnologías híbridas tales como Mecatrónica, electro neumática, eléctrica y sensorica, apoyadas por sistemas de hardware y software de control tipo PLC y el uso de sistemas SCADA aplicados, que desarrollen potentes

sistemas de control programas para brindar soluciones estándar o personalizadas a los diferentes problemas planteados.

Los almacenes automatizados ofrecen la mejor solución en la gestión de materiales en un entorno controlado donde se busca integrar el proceso logístico, minimizar errores, incrementar la productividad y garantizar la seguridad del personal y de los materiales almacenados dando la mayor calidad al servicio. En la **Figura 1** observamos un ejemplo de almacén Vertical a automatizar, donde se podría utilizar el módulo.



**Figura 1: Almacenamiento Vertical de Productos**

Fuente: <http://www.viastore.com/automatic-tray-warehouses/>

### 1.3. JUSTIFICACIÓN

Frente a los nuevos retos y necesidades de la industria de almacenes nuestra tesis **“Control e implementación de un módulo de almacenaje vertical con PLC y SCADA para la optimización de un proceso de almacenaje industrial”** ofrece una solución eficaz para enfrentar las exigencias específicas de espacio y optimizar los flujos operativos por medio de un PLC (Controlador Industrial) configurado en lenguaje de programación industrial el cual podrá establecer un control en tiempo real y supervisado por un sistema SCADA, mediante la instalación de una Interfaz Hombre-Máquina (HMI-Human Machine Interface); pudiendo seleccionar distintos tipos de funcionamiento para programar el almacenaje tal como se presenta en la vida real en los grandes almacenes, se obtiene como resultados como el mostrado en la **Figura 2**.



**Figura 2: Sistema Automatizado de Almacenamiento**

Fuente: <http://almadeherrero.blogspot.com/2012/09/almacenes-automatizados.html>

La automatización de un almacén, en su concepción teórica, no presenta una gran dificultad, pero la exigencia de integrar el conjunto de operaciones puede llegar a resultar un problema complejo, exigiendo una cuidadosa planificación y un radical cambio en la mentalidad y formación de los usuarios y diseñadores, la selección apropiada de los medios de transporte, la selección por capacidad de los actuadores así como la instalación de las fuentes de energía apropiadas.

Para qué sirve un almacenaje automatizado:

- Para optimizar espacio
- Para optimizar tiempo.
- Automatización de las operaciones de entrada y salida de productos.
- Para optimizar los recursos de un sistema de almacenaje.
- Para reducir los errores y accidentes en la gestión de almacenamiento.

En la tesis "*Control e implementación de un módulo de almacenaje vertical con PLC y SCADA para la optimización de un proceso de almacenaje industrial*", se realiza la implementación en forma versátil y se explica paso a paso la selección de los medios de transporte, la selección por capacidad de los actuadores, la instalación de las fuentes de energía apropiadas, la programación del PLC así como el diseño y programación del HMI que sirvieron para el desarrollo del trabajo de tesis.



## 1.4. OBJETIVOS

### 1.4.1. OBJETIVO GENERAL

- Implementar un módulo de almacenaje vertical automatizado controlado por un PLC y supervisado por un sistema SCADA, para optimizar y solucionar las exigencias de espacio, tiempo y recursos humanos en sistemas de almacenaje, de una forma eficaz y eficiente.

### 1.4.2. OBJETIVOS ESPECÍFICOS

- a) Diseñar los diagramas eléctricos y neumáticos del módulo de almacenaje vertical, realizar la selección de actuadores y elementos de control necesarios para la implementación del módulo, respetando las normas aplicables para su implementación.
- b) Implementar la estructura soporte y elementos mecánicos necesarios para el funcionamiento del módulo de almacenaje vertical propuesto, seleccionando el material con características adecuadas.
- c) Diseñar el control secuencial para el módulo de almacenaje vertical e implementar su programación mediante un Controlador Lógico Programable, utilizando las herramientas complementarias del software Step 7 para la programación en el PLC Siemens S7-200.
- d) Establecer el control y supervisión del módulo de almacenaje vertical a través de un sistema SCADA mediante un Interfaz Humano-Maquina (HMI), se realizará la comunicación entre el PLC y la pantalla Táctil, el diseño se realizará a través del Software MCGS.
- e) Realizar la puesta en marcha y pruebas de funcionamiento del módulo.
- f) Elaborar prácticas de laboratorio con fines didácticos, para el aprovechamiento del módulo de Almacenaje Vertical implementado en prácticas y/o laboratorios de la especialidad de Ingeniería Mecatrónica.



## 1.5. ALCANCES Y LIMITACIONES

---

### ALCANCES:

- Implementación un módulo de almacenaje vertical automatizado controlado por un PLC y supervisado por un sistema SCADA.
- Una vez desarrollado el módulo, se dejará al alcance de los alumnos de siguientes promociones para que pueda utilizarse como prototipo y referencia de posteriores trabajos de investigación y desarrollo tecnológico.
- Se estudiará con detenimiento lo concerniente a la programación PLC y al diseño del HMI.
- Se describirá la configuración y programación del PLC, así como el desarrollo paso a paso en el programa Step7.
- Se establecerá la comunicación entre el PLC y el HMI y realizará su configuración mediante el Software MCGS.
- Se realizará prácticas y pruebas de los sensores de posición, actuador neumático, motores eléctricos programados desde el PLC y pantalla táctil, verificando así el correcto funcionamiento de la secuencia inicial propuesta.

### LIMITACIONES

De acuerdo a lo indicado anteriormente esta tesis es delimitada de la siguiente manera:

- El presente módulo a escala se limita a una base de almacenamiento de 3 filas y 3 columnas.
- Como unidad de Almacenamiento a escala, se establece Piezas con dimensiones de 86mm x 86mm x 10mm
- Los límites de carrera son limitados a las siguientes dimensiones: Carrera Horizontal: 485 mm; Carrera Vertical: 400 mm.
- No se desarrolla el diseño de la parte estructural del módulo.
- El HMI solo tendrá la función de supervisar y controlar, más no la de recopilación de datos e historial.



- La programación de los Modos Automático y Manual se desarrollara como programas independientes.
- Una limitación del módulo de almacenaje es la falta de capacidad de conocer si físicamente existe o no un producto dentro de la matriz de almacenamiento de productos, queda abierta la posibilidad de una mejora a este módulo en trabajos futuros.





## CAPÍTULO 2: FUNDAMENTO TEÓRICO

En el presente capítulo revisaremos los conceptos de almacenamiento y paso hacia la automatización así como los diferentes modelos y tipos de almacenaje automatizado que existen hoy en el mercado industrial.

Repasaremos las definiciones de automatización y del manejo de la información entre dispositivos digitales y comprender el principio de funcionamiento de los sistemas de control aplicados, describiremos el dispositivo de control utilizado el PLC Siemens S7-200 y el interfaz Human-Machine del sistema SCADA.

### 2.1 ALMACENES

Un almacén es un espacio destinado a guardar materias primas, productos semielaborados o mercancías acabadas de una empresa industrial o las existencias de reserva de un establecimiento comercial<sup>1</sup>.

La palabra Almacén proviene de Almagacen, vocablo árabe que significa "Tesoro", por tal significación se identifica el almacén y sus mercancías como un tesoro muy valioso, apreciado que se debe guardar, custodiar y cuidar para el futuro<sup>2</sup>.

Las características del material a manejar, como pueden ser el tamaño, peso, durabilidad, duración y tamaño de lote harán que varíe la configuración de las instalaciones, equipo y técnicas a utilizar.

Otros aspectos a tener en cuenta a la hora del diseño son el tamaño del inventario y la ubicación, medidas respecto a la inspección de la calidad, medidas relativas a la selección y empaquetamiento, apilamiento para recibir y embarcar mercancías, número de andenes apropiados para embarcar y recibir mercancías, y mantenimiento de registros.

---

<sup>1</sup> [10]El diario de un logístico, <http://eldiariodeunlogistico.blogspot.com/2011/09/definicion-funcion-y-clases-de-almacen.html>

<sup>2</sup> [11]**Humberto Jasso**, Administración de Operaciones  
<http://es.scribd.com/doc/94485660/ALMACENES>

La gestión de un almacén, como parte esencial de la cadena logística, implica controlar la entrada, ubicación, movimientos y salida hasta su destino final, de las mercancías ubicadas en el recinto de almacenaje, con el consecuente tratamiento de la información generada de los movimientos lo que permitirá obtener fácilmente una información fiable y altamente disponible que nos ayudará a conocer al instante dónde y cómo deben ser almacenados los productos que llegan a la compañía cuyo destino es el almacén de ésta.

---

### 2.1.1 CONSIDERACIONES PARA EL DISEÑO DE UN ALMACEN

---

#### Tipo y número de materiales

Las dimensiones, peso, forma y durabilidad determinan en gran parte los métodos para almacenamiento y manejo. Un primer paso para la planificación sería detectar los diferentes materiales que se almacenarán, la cantidad de cada material y sus características físicas. Estas características son:

- **Ratio volumen/peso:** Una alta densidad conlleva un uso eficiente de los elementos de transporte y almacén y que los sistemas de almacén deban ser resistentes.
- **Ratio valor/peso:** Un alto valor refleja que el producto tendrá una mayor capacidad de absorber los costes de distribución.
- **Posibilidad de sustitución:** Productos como los alimentos básicos de sustitución elevada obligan a tener grandes cantidades de stock (almacén caro) o sistemas de distribución muy eficientes (transporte caro).
- **Características especiales:** Tales como materiales peligrosos, productos frágiles, bienes perecederos, productos refrigerados o productos de muy elevado valor.

---

### 2.1.2 UNIDAD DE CARGA

---

Cuando determinamos el nivel de inventario de cada artículo almacenado, se selecciona una unidad de almacenamiento, la cual es el número mínimo de artículos que se almacenan como unidad. Ésta se suele seleccionar de acuerdo con las características físicas del material, el equipo disponible para manejo y almacenamiento, la cantidad y la forma en la cual se recibe o embarca el material.



La selección de una unidad de carga óptima está muy ligada a los costes logísticos generales de una empresa, los costes asociados a la manipulación, los costes de No-Calidad asociados a roturas y desperfectos y los costes de carga y descarga. Su objetivo será, por tanto, minimizar los costes al mismo tiempo que sea homogénea y única la unidad de carga, para simplificar los sistemas de almacenamiento y utilizar unos equipamientos estandarizados y más baratos.

A la hora de elegir una unidad de carga se deben de tener en cuenta las siguientes dos características:

- **Resistencia:** Capacidad de soportar su peso o el de otras unidades de carga. Si es posible apilar la unidad de carga no necesitaremos equipo de almacén, pero en el caso de que no lo sea se puede hacer uso de elementos tales como paletas, pilares,... que dotan a la unidad de carga de esa característica.
- **Estabilidad:** Esta se lograra realizando una correcta configuración de la unidad de carga, envolviendo mediante películas de plástico que abrazan la mayor parte de la superficie de las cargas de forma regular o utilizando cinchas o bandas de goma, de Nylon, o de otros materiales, con resistencia a la cizalladura pero con un cierto grado de elasticidad.

La unidad de carga puede disponer de elementos auxiliares, los cuales pueden ser:

- **Skids:** Barras de madera o metal que eleva el producto lo suficiente para que el equipo de manutención quepa por debajo del producto y lo pueda coger.
- **Plataformas:** Placas de madera con cuatro pies, que permiten depositar los productos sobre ellos y que el método de transporte pueda llevarse de modo conjunto, conocidas comúnmente como "parihuela".
- **Paleta:** Plataforma horizontal de altura reducida al mínimo compatible con su manejo mediante carretillas elevadoras de horquilla, transpaleta o cualquier otro mecanismo elevador de manutención.
- **Cajas:** Contenedores pequeños para piezas y para rápida preparación manual de pedidos. Pueden remplazar a los elementos de estantería por su mayor adaptabilidad ya que sirven como medio de transporte y como medio de almacenaje en el taller.

- **Cajas-Paletas:** Paletas con al menos tres paredes verticales, llenas o caladas, fijas plegables o desmontables, provistas o no de tapadera. Si permiten el apilado pueden construir simultáneamente el sistema de almacenamiento y la unidad de carga con las múltiples ventajas que ello supone
- **Contenedores ISO:** Utilizados sobre todo para el transporte marítimo.
- **Practicable:** Contenedor que siempre recibe el mismo tipo de piezas y que permite la manutención y almacenamiento de estos con medios convencionales.<sup>3</sup>

---

### 2.1.3 EQUIPO DE ALMACENAMIENTO

---

Después de establecer las unidades y niveles de almacenamiento se selecciona el equipo de almacenamiento el cual debe ser compatible con la capacidad de carga de los pisos, espaciamiento entre columnas, ubicación de andenes para recepción y embarque, etc....

El tipo de equipo a utilizar se determinará por las características de la unidad de carga. Los niveles de inventario que se deben mantener, determinan el número de unidades del equipo de almacenamiento. Las características de los materiales y su volumen de movimiento son factores decisivos en la selección del equipo para manejo de materiales.

El equipo de almacenamiento aparte de los distintos elementos de transporte y carga-descarga, está constituido también por las estanterías utilizadas para almacenar las unidades de carga. Además de las estanterías, también se puede definir el piso como equipo de almacenamiento si se usa en parte o en su totalidad. Las unidades de almacenamiento pesadas o voluminosas y carentes de rigidez y cuyo número es pequeño se almacenan mejor en estanterías. Las unidades de almacenamiento pequeñas, como relojes de pulsera o tornillería se almacenan en estanterías y casilleros.

El tiempo de transporte y almacenaje del producto supone habitualmente la mayor parte del tiempo total que el producto permanece en la planta industrial. De ahí radica la importancia de optimizar el rendimiento de la planta respecto al abanico de

---

<sup>3</sup> [19]Desarrollo de un Sistema HMI para un almacén automatizado,  
[http://e-archivo.uc3m.es/bitstream/handle/10016/16073/Proyecto\\_ESTHER\\_SALICHS.pdf](http://e-archivo.uc3m.es/bitstream/handle/10016/16073/Proyecto_ESTHER_SALICHS.pdf)

posibilidades en sistemas de transporte y almacenaje. El almacén automático es una de las soluciones más eficaces gracias a su optimización del espacio, tiempo y recursos. Debe desarrollarse en un entorno totalmente controlado, diseñado como un todo en el que cada solución parcial debe integrarse por completo en el entorno global.

Toda la instalación debe de cumplir con una velocidad de flujo alineada con la productividad deseada, con flexibilidad a la hora de adaptar las diferentes necesidades así como una alta precisión en todos los movimientos.

---

#### 2.1.4 TIPOS DE ALMACENES<sup>4</sup>

---

Se pueden clasificar de las siguientes formas:

##### **Según su relación con el flujo de producción**

- **Almacenes de materias primas:** Aquellos que contiene materiales, suministros, envases, etc., que serán posteriormente utilizados en el proceso de transformación.
- **Almacenes de productos intermedios:** Aquellos que sirven de colchón entre las distintas fases de obtención de un producto.
- **Almacenes de productos terminados:** Exclusivamente destinados al almacenaje del resultado final del proceso de transformación.
- **Almacenes de materiales auxiliares:** Sirven para almacenar repuestos, productos de limpieza, aceites, pinturas, etc. La demanda de estos productos suele ser estocástica.
- **Almacenes de preparación de pedidos y distribución:** Su objeto es acondicionar el producto terminado y ponerlo a disposición del cliente.

##### **Según el material a almacenar:**

- **Almacén para bultos:** El material se junta en unidades de transporte y de almacén óptimas para el aprovechamiento pleno de la capacidad de carga de un vehículo para conseguir su transporte económico.
- **Almacén de gráneles:** Si es posible, debe estar en las proximidades del lugar de consumo debido a que el transporte es costoso. Hay que hacer transportable y

---

<sup>4</sup> [12] **José P. García –Sabater**, Diseño de Sistemas Productivos y Logísticos <http://personales.upv.es/jpgarcia/LinkedDocuments/7%20Almacenes.pdf>

almacenable el material que se puede verter. Su contenido debe poderse medir automáticamente, su extracción regulable y con conexión a un medio de transporte.

- **Almacén de líquidos:** El material utilizado es específico de granel y se puede transportar por cañerías.
- **Almacén de gases:** Requiere unas medidas de seguridad especiales que han de ser observadas por la alta presión o la particular inflamabilidad.

**Según su localización:** El tipo de carga y estructura de costes de la empresa son los parámetros para la elección de almacenes centrales o almacenes regionales. Así productos de bajo valor, o costes de transporte elevados, conducen al uso de almacenes regionales. Por el contrario, con costes de almacén elevados por el valor del producto, se tiende al uso de almacenes centrales. En cualquier caso, existen métodos para la evaluación de la mejor decisión.

- **Almacenes centralizados:** Aquellos que se localizan lo más cerca posible del centro de fabricación.
- **Almacenes locales y regionales:** Aquellos que se ubican cerca del punto de consumo.

#### Según su función logística

- **Centro de consolidación:** Estos almacenes reciben productos de múltiples proveedores y los agrupan para servirlos al mismo cliente. Se justifica en aquellos casos en los que los pedidos incluyan una gran diversidad de productos
- **Centro de ruptura:** Tienen la función inversa de los centros de consolidación. Recibe la carga de un número reducido de proveedores y sirven a un gran número de clientes, con necesidades dispares. Reducen el número de contactos de los fabricantes con los clientes finales y reducen el movimiento de los clientes que únicamente han de acudir a un centro de ruptura para recoger múltiples productos.
- **Centro de tránsito (Cross-dock):** Son almacenes que mueven productos sin llegar a almacenarlos. Se define como la utilización de instalaciones intermedias (almacenes temporales) para el movimiento de mercancías de una ubicación a otra, especialmente entre los muelles de carga y los de descarga.

• **Almacenes cíclicos o estacionales:** Recogen una producción puntual para hacer frente a una demanda constante o una demanda puntual frente a una producción más constante.

• **Almacenes de custodia a largo plazo:** Su principal objetivo es estar lleno, sin importar los costes de transporte, demandas o ritmos de producción.

---

### 2.1.5 OPERACIONES FUNDAMENTALES

---

1. **Entrada de bienes:** Recepción de las mercancías a través de los muelles de carga, pasando por los controles de calidad, cuarentenas y cambios de embalaje necesarios.

2. **Almacenamiento:** Disposición de las cargas en su ubicación con el objeto de retenerlas hasta su puesta a disposición

3. **Recogida de pedidos:** Conocida también por picking, es la operación por la que se convierten las unidades de carga de compra en unidades de venta.

4. **Agrupación-Ordenación:** Dependiendo del procedimiento de generación de pedidos, y de la configuración del sistema de distribución, será necesario establecer un sistema para agrupar y ordenar los pedidos según las rutas de distribución.

5. **Salida de bienes:** El control de salidas, recuento numérico o control de calidad y el embarque en el medio de transporte correspondiente son las funciones con las que finaliza el proceso.

---

### 2.1.6 METODOS DE ALMACENAMIENTO

---

#### **METODOS TRADICIONALES (Sistemas de almacenamiento no automatizados)**

Son aquellos cuyos sistemas están muy difundidos y accesibles por sus costos de adquisición, los cuales están siendo desplazados por los automatizados, en vista de la mayor rapidez que ofrecen. La creciente necesidad de contar con el espacio físico, que se hace inaccesible en el sistema tradicional, obliga a los empresarios a optar por los sistemas de almacenamiento en donde el espacio físico aéreo es bien aprovechado.

Los almacenes tradicionales utilizados por las empresas son:

- Almacenamiento de bulto (Bula storage)

- Almacenamiento de cajón
- Almacenamiento de estanterías

### **MÉTODOS NO TRADICIONALES (Sistemas de Almacenamiento Automatizados)**

Son aquellos que cuentan con la última tecnología, en ellos predomina la rapidez de su manipulación de almacenar y recuperar y ganar el espacio aéreo. Además reducen o eliminan la cantidad de inversión y participación humana. Son estructuras, generalmente de gran altura, donde los elementos de almacenamiento y los elementos de manutención van integrados y controlados por un sistema informático. Predomina la rapidez de manipulación de almacenaje y recuperación y se gana en espacio aéreo.

Existen dos tipos de almacenes automatizados:

#### **Almacenamiento automatizado /Sistemas de recuperación (AS/RS)**

El sistema de almacenamiento ASRS es un sistema que permite controlar, almacenar y recuperar con bastante precisión y velocidad un elemento definido.

Tiene como componentes la estructura de almacenamiento que viene a ser la estantería donde se almacenaran los componentes, los módulos de almacenamiento, la máquina de RS encargada de almacenar y recuperar los componentes almacenados y la estación de comando. Estos sistemas, sobre todo, revolucionan la forma de almacenaje rápido y permiten ganar espacio aéreo.

El sistema informático ubica los productos en las estanterías mediante el transelevador. Cuando las mercancías son requeridas, el sistema informático lanza la orden de recogida. Los AS/RS pueden sobrepasar los 35 metros de altura mientras que los MiniLoads sobrepasan los 12 metros de altura. Los almacenes tipo AS/RS suelen ser estructuras autoportantes en los que el soporte del edificio coincide con el soporte de las cargas.

Algunos ejemplos de sistemas AS/RS son:

- Almacenes verticales de pallets.
- Almacén automático de recambios pequeños (miniloads)
- Sistemas multi-shuttle de alto rendimiento.



Los almacenes AS/RS se recomiendan para empresas con una alta rotación de artículos, muy alta gama de referencias, de unidades homogéneas de volumen de paleta o superior donde la superficie disponible exija grandes alturas de almacenamiento. Los almacenes MiniLoad se recomiendan para artículos de poco volumen y elevada cantidad de referencias, con un muy alto movimiento de artículos.

### **Sistemas de almacenamiento de carrusel:**

Los almacenes rotativos verticales también llamados carruseles verticales, norias o paternóster, están compuestos por un conjunto de bandejas que giran en sentido vertical para facilitar al operario el acceso a la mercancía que necesita. La bandeja solicitada se detiene en la ventana de acceso a la máquina. Estos almacenes rotativos son manejados por un trabajador que se sitúa en la estación de carga y descarga, cuando la bandeja donde se encuentra el material requerido pasa por la ventana de acceso, el empleado frena la máquina y extrae o repone el material que se encuentra dentro de unas cestas o arcas.

La gestión de suministros de materiales así como los avances en la logística de almacenaje ha hecho que el uso de estos carruseles verticales se generalice en aplicaciones dentro de la industria. La altura de estas máquinas abarca entre los 2.5m hasta los 30m, aunque la medida estándar es de 10m<sup>5</sup>.

## 2.2 AUTOMATIZACIÓN

### 2.2.1 CONCEPTO DE AUTOMATIZACIÓN

La automatización es el uso de sistemas o elementos computarizados y electromecánicos para controlar maquinarias o procesos industriales que antes se manejaban por operadores humanos a un proceso autónomo, es decir, por cuenta propia.

Esta tecnología incluye:

- Máquinas herramientas automáticas para procesar piezas

---

<sup>5</sup> [13] **María Moneo**, Automatización Carruseles verticales, Artículo Carruseles verticales de gran altura. <http://img.mecalux.pe/external/magazine/41899.pdf>



- Máquinas automáticas de ensamble
- Robots industriales
- Sistemas automáticos de manejo y almacenamiento de partes
- Sistemas automáticos de inspección para control de calidad
- Procesos y su retroalimentación controlados por computadora
- Sistemas computacionales para planear, reunir información, y tomar decisiones relacionadas con actividades de manufactura.

---

## 2.2.2 OBJETIVOS DE LA AUTOMATIZACIÓN

---

- Mejorar la productividad de la empresa, reduciendo los costes de la producción y mejorando la calidad de la misma.
- Mejorar las condiciones de trabajo del personal, suprimiendo los trabajos penosos e incrementando la seguridad.
- Realizar las operaciones imposibles de controlar intelectual o manualmente.
- Mejorar la disponibilidad de los productos, pudiendo proveer las cantidades necesarias en el momento preciso.
- Simplificar el mantenimiento de forma que el operario no requiera grandes conocimientos para la manipulación del proceso productivo.
- Integrar la gestión y producción.

---

## 2.2.3 TEORIA DE CONTROL<sup>6</sup>

---

El control automático ha desempeñado una función vital en el avance de la ingeniería y la ciencia. Además de su extrema importancia en los sistemas de vehículos espaciales, de guiado de misiles, robóticos y similares; el control automático se ha vuelto una parte importante e integral de los procesos modernos industriales y de manufactura. Por ejemplo, el control automático es esencial en el control numérico de las máquinas-herramienta de las industrias de manufactura, en el diseño de sistemas de pilotos automáticos en la industria aeroespacial, y en el diseño de automóviles y camiones en la industria automotriz. También es esencial en las operaciones industriales como el control de presión, temperatura, humedad, viscosidad y flujo en las industrias de proceso.

---

<sup>6</sup>[7]K. Ogata. "Ingeniería de Control Moderna", 3ra Ed., Pearson.



Debido a que los avances en la teoría y la práctica del control automático aportan los medios para obtener un desempeño óptimo de los sistemas dinámicos, mejorar la productividad, aligerar la carga de muchas operaciones manuales repetitivas y rutinarias, así como de otras actividades, casi todos los ingenieros y científicos deben tener un buen conocimiento de este campo.

## **CONTROL EN LAZO CERRADO EN COMPARACIÓN CON EL CONTROL EN LAZO ABIERTO**

Sistemas de control realimentados. Un sistema que mantiene una relación prescrita entre la salida y la entrada de referencia, comparándolas y usando la diferencia como medio de control, se denomina sistema de control realimentado. Un ejemplo sería el sistema de control de temperatura de una habitación. Midiendo la temperatura real y comparándola con la temperatura de referencia (la temperatura deseada), el termostato activa o desactiva el equipo de calefacción o de enfriamiento para asegurar que la temperatura de la habitación se conserve en un nivel cómodo sin considerar las condiciones externas.

Los sistemas de control realimentados no se limitan a la ingeniería, sino que también se encuentran en diversos campos ajenos a ella. Por ejemplo, el cuerpo humano es un sistema de control realimentado muy avanzado. Tanto la temperatura corporal como la presión sanguínea se conservan constantes mediante una realimentación fisiológica. De hecho, la realimentación realiza una función vital: vuelve el cuerpo humano relativamente insensible a las perturbaciones externas, por lo cual lo habilita para funcionar en forma adecuada en un ambiente cambiante.

**Sistemas de control en lazo cerrado.** Los sistemas de control realimentados se denominan también sistemas de control en lazo cerrado. En la práctica, los términos control realimentado y control en lazo cerrado se usan indistintamente. En un sistema de control en lazo cerrado, se alimenta al controlador la señal de error de actuación, que es la diferencia entre la señal de entrada y la señal de realimentación (que puede ser la señal de salida misma o una función de la señal de salida y sus derivadas y/o integrales), a fin de reducir el error y llevar la salida del sistema a un valor conveniente. El término control en lazo cerrado siempre implica el uso de una acción de control realimentado para reducir el error del sistema.

**Sistemas de control en lazo abierto.** Los sistemas en los cuales la salida no afecta la acción de control se denominan sistemas de control en lazo abierto. En otras palabras, en un sistema de control en lazo abierto no se mide la salida ni se realimenta para compararla con la entrada. Un ejemplo práctico es una lavadora. El remojo, el lavado y el enjuague en la lavadora operan con una base de tiempo. La máquina no mide la señal de salida, que es la limpieza de la ropa.

**Sistemas de control en lazo cerrado en comparación con los sistemas en lazo abierto.**

Una ventaja del sistema de control en lazo cerrado es que el uso de la realimentación vuelve la respuesta del sistema relativamente insensible a las perturbaciones externas y a las variaciones internas en los parámetros del sistema. Por tanto, es posible usar componentes relativamente precisos y baratos para obtener el control adecuado de una planta determinada, en tanto que hacer eso es imposible en el caso de un sistema en lazo abierto.

---

#### 2.2.4 CONTROL AUTOMÁTICO

---

Un sistema automatizado consta de dos partes principales:

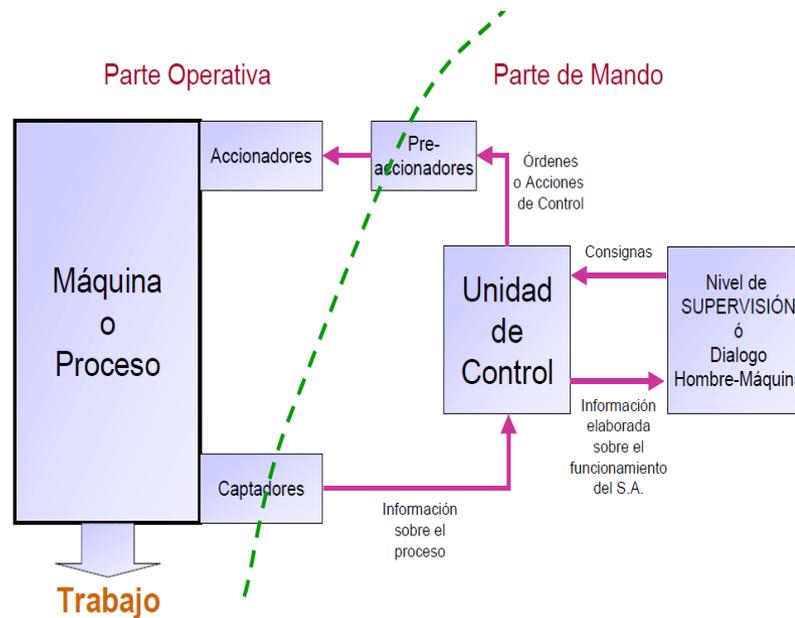
- Parte Operativa
- Parte de Mando

La Parte Operativa parte del sistema que actúa directamente sobre el proceso. Son los elementos que hacen que la máquina se mueva y realice la operación deseada. Los elementos que forman la parte operativa son los accionadores y pre-accionadores de las máquinas como motores, cilindros, compresores y los detectores o captadores como fotodiodos, finales de carrera, etc.

La Parte de Mando comúnmente autómatas programables (tecnología programada), aunque hasta hace bien poco se utilizaban relés electromagnéticos, tarjetas electrónicas o módulos lógicos neumáticos (tecnología cableada). Actualmente también se utiliza, aunque en menor medida, los ordenadores de control de proceso y los reguladores industriales.

En un sistema de fabricación automatizado, el autómatas programables está en el centro del sistema. Este debe ser capaz de comunicarse con todos los constituyentes del sistema automatizado. Para que pueda existir una comunicación entre el operador y el

sistema de control existen los elementos de salida de información y los elementos de entrada de órdenes (ver **Figura 3**).<sup>7</sup>



**Figura 3 : Diagrama de bloques de un sistema automatizado**

**Fuente:** <http://es.scribd.com/doc/214837312/1-0-CE-La-Automatizacion-17p>

### 2.2.5 MANUFACTURA INTEGRADA POR COMPUTADORA

CIM acrónimo de Computer Integrated Manufacturing es una filosofía y estrategia de producción, caracterizada por integrar toda la información de las distintas áreas de una empresa a través de sistemas informáticos y la utilización de equipos electrónicos para el control, supervisión y gestión de los procesos.

Las diferencias entre automatización y manufactura integrada por computadora es que la automatización está relacionada con las actividades físicas en la manufactura; los sistemas de producción automatizada están diseñados para ejecutar el procesamiento, montaje, manejo de material y actividades de inspección con poca o nula participación humana. La manufactura integrada por computadora está más relacionada con las funciones de información de procesamiento que son requeridas

<sup>7</sup>[14]<http://www.sc.edu/es/sbweb/webcentro/automatica/WebCOMH1/PAGINA%20PRINCIPAL/Automatizacion/Automatizacion.htm>

para apoyar las operaciones de producción además involucra el uso de sistemas por computadora para llevar a cabo los cuatro tipos de funciones de información de procesamiento.

---

### 2.2.6 RAZONES PARA LA AUTOMATIZACIÓN

---

- Aumento de productividad
- Alto costo de mano de obra
- Mano de obra escasa
- Tendencia de mano de obra con respecto al sector de servicios
- Seguridad
- Alto costo de materiales en bruto
- Mejora la calidad del producto
- Reduce el tiempo de manufactura
- Reducción del proceso de inventarios
- Alto costo de la no automatización

Todos estos elementos actúan conjuntamente para hacer de la producción automatizada una atractiva alternativa para métodos manuales de manufactura.

---

### 2.2.7 ARGUMENTOS A FAVOR Y EN CONTRA DE LA AUTOMATIZACIÓN

---

#### **En contra:**

1. La automatización genera la dependencia de las máquinas en los procesos industriales, creando que no sea necesario la mano de obra para la producción.
2. La reducción de la fuerza laboral por el reemplazo de la mano de obra por máquinas de producción.
3. La inversión para actualizar todo un proceso en una planta tiene un costo elevado inicial.
4. Genera competencia laboral por tener las habilidades necesarias en un mundo automatizado.

#### **A favor:**

1. La automatización exige el cambio en la mentalidad de las personas de una comunidad, pueblo o país con respecto a la tecnología.
2. La automatización genera la necesidad por estar a la vanguardia de los conocimientos de automatización.



3. Se genera la competencia intelectual por parte de los trabajadores al ser una ciencia viva y de constante cambio, exige avanzar al paso de la tecnología.
4. Promueve la generación de mayores avances tecnológicos creados para suplir nuevas necesidades de automatizar los procesos.
5. Reducción de accidentes de personal en trabajos riesgosos.
6. Reducción de errores de producción y de calidad en los productos.

## 2.3 SISTEMAS PARA LA AUTOMATIZACIÓN

### 2.3.1 SISTEMAS DE AUTOMATIZACION MECANICA

Equipos que transforman la energía eléctrica en mecánica para el desarrollo de trabajos iterativos de producción o mecanizado.

Como mecanismos habituales tenemos los engranajes, las correas de transmisión, palancas, etc.

### 2.3.2 SISTEMAS DE AUTOMATIZACION NEUMATICO

Sistema que utiliza el aire del ambiente para trabajarlo, existen maquinas las cuales generan el aire para trabajar y aquellas que utilizan el aire generado para un proceso.

Se tiene como mecanismos habituales a los: compresores, electroválvulas, pistones, etc.

### 2.3.3 SISTEMAS DE AUTOMATIZACION HIDRAULICA

Sistema similar al neumático con la diferencia que este sistema utiliza en vez de aire un fluido debido a sus características de incomprensión, las cuales generan mayores presiones y precisión de trabajos.

El fluido se utiliza como generador de potencia y al mismo tiempo como lubricante, la única diferencia con el sistema neumático es que la velocidad de respuesta es un poco menor que de un sistema neumático.

### 2.3.4 SISTEMAS DE AUTOMATIZACION ELECTRONICA Y ELECTRICA

Uno de los sistemas de automatización más extendido al día de hoy, los sistemas de actuación eléctrica entre los más comunes tenemos a los motores, actuadores electromagnéticos, etc.

El mando eléctrico suele implantarse mediante relés.

El mando electrónico puede ser implementado mediante componentes electrónicos discretos digitales o mediante sistemas de lógica programable (FPGA).

El método de automatización electrónico más extendido es el microprocesador.

#### **Componentes electrónicos utilizados en la automatización:**

- Tubos de vacío
- Transistores
- Circuitos integrados
- Resistencias
- Bobinas
- Dispositivos de detección y transductores

El desarrollo de los circuitos integrados ha revolucionado los campos de las comunicaciones, la gestión de la información y la informática. Los circuitos integrados han permitido reducir el tamaño de los dispositivos con el consiguiente descenso de los costes de fabricación y de mantenimiento de los sistemas. Al mismo tiempo, ofrecen mayor velocidad y fiabilidad.

Los componentes eléctricos son los más comunes en todo tipo de industria. En los siguientes puntos comentaremos sobre aquellos componentes que son necesarios en cualquier sistema de automatización.

#### **2.3.4.1 MOTORES**

##### **MOTORES C.A.**

Son aquellos motores que utilizan como alimentación la corriente alterna (CA), los principales componentes son el estator el cual es suministrado con la corriente y crea un campo magnético giratorio y el rotor el cual gira dentro de las bobinas y es el que alimenta al eje de salida para la torsión necesaria.

Tenemos dos diferentes tipos de motor c.a. los cuales utilizan diferentes tipos de rotor.

Motor asíncrono el cual genera una corriente inducida debido al campo magnético creado en el rotor.

Motor síncrono debido a que gira a la frecuencia de alimentación, cabe resaltar que el motor síncrono gira a una mayor velocidad que el motor asíncrono debido a que la velocidad se reduce por el deslizamiento del motor asíncrono.

### **MOTORES C.D.**

Los motores eléctricos de corriente directa tienen un voltaje inducido giratorio devanado de armadura, y un marco del campo inducido no giratorio devanado que es un campo estático, o un imán permanente. Los motores eléctricos de CD utilizan diferentes conexiones, y un devanado inducido para producir diferentes velocidades y la regulación de la torsión.

La diferencia que presenta estos motores contra los de a.c. es que se puede variar su velocidad aplicando una tensión a la armadura del motor, o mediante el ajuste de la corriente del marco del campo.

### **SERVOMOTORES**

Motores utilizados preferentemente en trabajos de automatización, debido a que posee un eje de rendimiento controlado, es decir, que podemos posicionarlo a nuestro antojo, siempre dentro de su rango de actuación. Por lo general los servomotores suelen tener un rango de 180° aunque existen de 210° e incluso de rotación continua.

### **OTROS MOTORES**

Existen en el mercado motores llamados de pulso o de paso, con los que se puede controlar posicionamiento sin recurrir a costosos servosistemas. Invariablemente requieren un control especial para su funcionamiento.

#### **2.3.4.2 CONTROL DE MOTORES**

Entre tantos tipos de controles de motores en el mercado, podemos clasificarlos en varios grandes grupos:

### **ARRANCADORES**

Un arrancador consiste en la combinación de un contacto y un relevador de sobrecargas conectadas entre sí y a una estación de botones, ya sea remota o local.



En el mercado existen dos tipos de arrancadores para la misma función. Los que siguen las normas NEMA<sup>8</sup> y los de tipo europeo o IEC<sup>9</sup>.

La diferencia entre ambos es la filosofía de diseño. El NEMA está fabricado para todos los motores que correspondan a una potencia, y en cambio, el IEC, de acuerdo con el número de arranques y de sobrecargas del motor, se selecciona el arrancador.

### **VARIADORES DE C.D.**

Por muchos años ha sido empleado este tipo de control, debido a su construcción sencilla, y aplicación sin problemas.

Muchos controles de éstos han sido fabricados para retroalimentación de velocidad por tacómetro.

### **INVERSORES**

Con los circuitos integrados de muy alta densidad y semiconductores de potencia baratos, ha sido posible la fabricación de sistemas de control de velocidad de corriente alterna a precios competitivos con los de c.d.

Existen dos tipos, los de modulación de voltaje (PWM en Inglés) y los de modulación de corriente. Los primeros causan gran interferencia con otros equipos electrónicos por el gran contenido de armónicas que producen. Los segundos, más caros, son más eficientes y no causan gran interferencia.

### **SERVOCONTROLES**

Los servocontroles son amplificadores de muy alta ganancia que se retroalimentan con la información proveniente de los tacómetros de los servomotores.

Estos amplificadores reciben como entrada una señal analógica de un control manual o automático; esto es, de un potenciómetro o de un PLC por ejemplo.

---

<sup>8</sup>**NEMA:** National Electrical Manufacturers Association

<sup>9</sup>**IEC:** International Electrotechnical Commission

Su uso es muy específico para lugares donde se requiere exactitud en la velocidad y/o en la posición de una máquina. Ejemplo: Los servos de las máquinas herramienta de control numérico.

### **CABLES Y ALAMBRES**

Se tiene una gama completa de cables a la disposición de las industrias para la conexión de los diferentes elementos de las máquinas.

Además, se implementó un sistema de Verificación privado de Instalaciones Eléctricas, para que se cumplan las Normas de Seguridad mínima al manejar la electricidad.

### **BOTONES Y SEÑALIZACION**

Los botones de señalización eléctrica han sufrido un cambio en los últimos años bajo la influencia de las normas europeas y de los nuevos sistemas electrónicos de control.

El cableado de hace unas décadas debía resistir algunos amperes de corriente y, por ende los contactos de todos los interruptores.

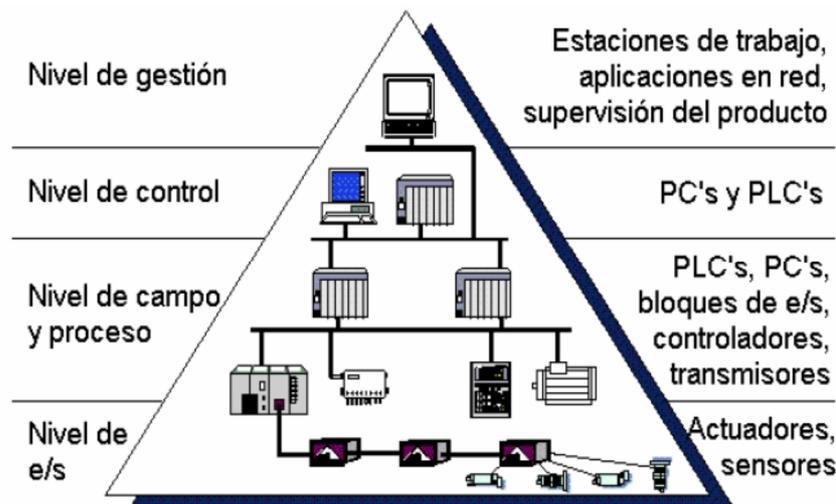
### **CONTROLES ALAMBRADOS**

Estos controles han estado presentes desde el inicio de la Electricidad hace 100 años y, aún siguen vigentes en nuestros días.

En ciertas aplicaciones no hay mejor control ni más barato que el control alambrado. Ej. El control de velocidad sin retroalimentación de un motor de corriente directa mediante un reóstato de campo.

## **2.4 LA PIRÁMIDE DE AUTOMATIZACIÓN**

El empleo de computadores y autómatas programables como herramienta esencial de tratamiento de la información es habitual y la implementación de redes de comunicación internas, necesaria. La **Figura 4** muestra de manera genérica estas comunicaciones que interrelacionadas dan una distribución jerárquica en un entorno de automatización integrada en una división en diferentes niveles



**Figura 4 : Pirámide de automatización según la operatividad en la fábrica**

- a) **Nivel de Bus de Campo – Entradas/Salidas:** Llamado usualmente *fieldbus*, es un nivel de instrumentación; formado por los dispositivos de medida (sensores) y elementos de mando (actuadores) distribuidos en una línea de producción. Son los elementos más directamente relacionados con el proceso productivo ya que los actuadores son los encargados de ejecutar las órdenes de los elementos de control para modificar el proceso productivo (por ejemplo, un motor de mezcla de materias primas) y los sensores miden variables en el proceso productivo (por ejemplo la temperatura de un horno).
- b) **Nivel de Campo y proceso:** En este nivel se sitúan los elementos capaces de gestionar los actuadores y sensores del nivel anterior, estos son autómatas programables o equipos de aplicación específica basados en microprocesador como robots, máquinas herramientas o controladores de motor. Estos dispositivos permiten que los actuadores y sensores funcionen de forma conjunta para ser capaces de realizar el proceso deseado. Son dispositivos programables, de tal modo que es posible ajustar y personalizar su funcionamiento según las necesidades de cada caso. Los dispositivos de este nivel de control junto con los del nivel inferior (*fieldbus*) poseen capacidad suficiente como para realizar procesos productivos por sí mismos, de hecho, gran cantidad de procesos industriales están basadas exclusivamente en estos dos niveles, de tal modo que un proceso productivo completo se desglosa en subprocesos de este tipo sin que exista un intercambio de información entre ellos.

A pesar de tratarse de procesos aislados, esto no implica que no se empleen buses de comunicación, ya que para procesos que requieran de gran número de sensores y actuadores, es recomendable la utilización de buses de campo para leer el estado de los sensores, proporcionar señal de control a los actuadores y conectar diferentes autómatas programables para compartir información acerca de la marcha del proceso completo.

También es importante que estos dispositivos posean unas buenas características de interconexión para ser enlazados con el nivel superior (supervisión), generalmente a través de buses de campo.

**c) Nivel de Control y Supervisión:** A todos los dispositivos de control existentes en planta es posible monitorizarlos si existe un sistema de comunicación adecuado capaz de comunicar estos elementos con otro tipo de dispositivos no dedicados al control sino para la gestión y supervisión y que habitualmente están constituidos por computadores o sistemas de visualización tales como pantallas industriales. En este nivel es posible visualizar como se están llevando a cabo los procesos de planta y a través de entornos SCADA (Supervisión, Control y Adquisición de Datos) poseer una imagen virtual de la planta de modo que esta se puede recorrer de manera detallada o bien mediante pantallas de resumen ser capaces de disponer de un panel virtual donde se muestran las posibles alarmas, fallos o alteraciones en cualquiera de los procesos que se llevan a cabo. Mediante este tipo de acciones resulta inmediato disponer de acceso inmediato a cada uno de los sectores de la planta, para ello, resulta imprescindible la conexión con el nivel de control mediante buses de campo de altas prestaciones, pues a veces es necesaria la transmisión de importantes cantidades de datos y la conexión con un gran número de elementos de control.

**d) Nivel de Gestión:** Dado que el nivel de supervisión ya está constituido principalmente por computadores, el nivel de gestión también lo estará ya que se encuentra más alejado de los procesos productivos. De hecho, en este nivel no es relevante el estado y la supervisión de los procesos de planta, en cambio, si adquiere importancia toda la información relativa a la producción y su gestión asociada es decir a través del nivel de supervisión de los procesos de planta, en cambio, si adquiere importancia toda la información relativa a la producción y su gestión asociada, es decir, a través del nivel de supervisión es posible obtener información de todos los



niveles inferiores de una o varias plantas. Con esta información los gestores de la empresa pueden extraer estadísticas acerca de los costes de fabricación, rendimiento de la planta y en general disponer de los datos que permitan a los niveles directivos la toma de decisiones conducentes a una mejor optimización en el funcionamiento de la planta, todo ello de manera rápida y flexible dada la rapidez del acceso a los datos de fabricación.

## 2.5 SISTEMAS MECATRÓNICOS

La Mecatrónica, o más recientemente Sistemas Mecatrónicos se han convertido en estos últimos años en un área de interés en Universidades y Centros de Investigación en el mundo. El término *Mechatronics* se origina más o menos hace 15 años en Japón, como una palabra compuesta entre unión de las palabras: mecánica y electrónica. Su origen se remonta a la integración entre el Diseño Asistido por Computadora (CAD) y la Manufactura Auxiliada por Computadora (CAM). Más recientemente se han integrado conceptos de Integración computacional e Inteligencia Artificial. Una definición más apropiada es la siguiente:

Mecatrónica es el acto de combinar el diseño y la fabricación de sistemas mecánicos, electrónicos y computacionales (ver **Figura 5**) para crear productos o sistemas de producción inteligentes. En este sentido, Mecatrónica se ha descrito como una aproximación requerida de conocimientos y una nueva cultura para realizar la nueva generación de máquinas, robots y mecanismos inteligentes requeridos en nuevas aplicaciones de producción, así como en ambientes hostiles para los seres humanos.



**Figura 5 : Sinergia de la Mecatrónica**

Haciendo un esfuerzo por dividir los elementos que integran de forma general a los Sistemas Mecatrónicos. A continuación se muestra una clasificación personal de lo que razonablemente podemos considerar como una clasificación de los elementos tecnológicos que integran a los sistemas mecatrónicos:

- Sistemas híbridos diseñados por computadora (electromecánica y computacional).
- Sistemas de manufactura asistidos por computadora.
- Sistemas de calidad asistidos por microcomputadoras.
- Sistemas sensoriales de visión.
- Actuadores de aplicación específica (Máquinas eléctricas, electro-neumática, etc.).
- Sistemas de ensamble y manipulación.
- Mecanismos de precisión.
- Técnicas de integración sensorial.
- Sistemas inteligentes de control.
- Interfaces inteligentes (Máquina/Máquina y Hombre/Máquina).

## 2.6 CONCEPTOS FUNDAMENTALES SOBRE LAS COMUNICACIONES DIGITALES

### 2.6.1 CONCEPTO DE RED (NET-WORKING)

Una red, es la conexión de dos o más dispositivos a través de algún medio de transmisión. Cuando se habla de un conjunto de dispositivos se ha referencia a computadores, controladores u otros dispositivos electrónicos conectados para compartir data (información digital).

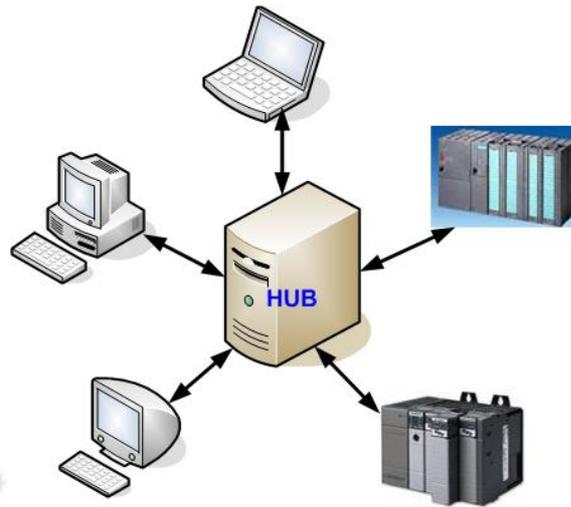
### 2.6.2 TOPOLOGÍA DE REDES

La topología de redes describe la forma en que varios equipos son conectados a la red. A continuación se detallan las topologías básicas, cada una de ellas con características propias.

**a) Topología tipo Estrella.-** En este tipo de conexionado, todos los nodos están conectados a un nodo central a través del cual pasan todos los datos de todos. El nodo central es común para todos y normalmente posee mayor capacidad de manejo de información para un determinado proceso, en la **Tabla 1** observamos las características de un red tipo Estrella, representada en la **Figura 6**.

<b>Tipo de control</b>	Control centralizado. Acceso regulado por un control central
<b>Redundancia</b>	Si falla la inteligencia central (HUB), la red falla
<b>Expansibilidad</b>	Limitado al número de conexiones del controlador central
<b>Requerimiento de cableado</b>	Cable apantallado
<b>Requerimiento de interface</b>	Es suficiente con RS-232C – conexión punto a punto

**Tabla 1 : Características principales de red configurada en topología estrella**

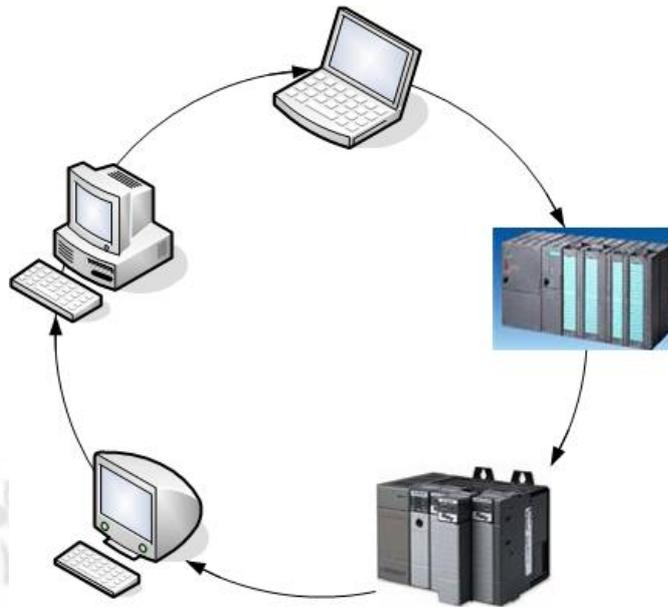


**Figura 6 : Conexión de una red configurada en topología estrella**

**b) Topología tipo Anillo.-** Este tipo de red está constituido por un conjunto de estaciones conectadas en serie y formando un anillo cerrado. Cada estación está conectada a la red a través de una interface que tiene la función de retransmitir datos que no están destinados a ese nodo, leer los datos destinados a ese nodo e insertar los datos enviados por él, sus características está en la **Tabla 2** y en la **Figura 7** observamos su forma característica de conexión.

<b>Tipo de control</b>	Control descentralizado. Acceso pasa de equipo a equipo.
<b>Redundancia</b>	Si falla la línea, la red falla.
<b>Expansibilidad</b>	Ilimitado teóricamente, pero el tiempo de rotación define un límite práctico pues este gobierna al tiempo de respuesta.
<b>Requerimiento de cableado</b>	Cable coaxial.
<b>Requerimiento de interface</b>	Debe de proveer inmunidad a interferencias en la transmisión.

**Tabla 2 : Características principales de red configurada en topología anillo.**

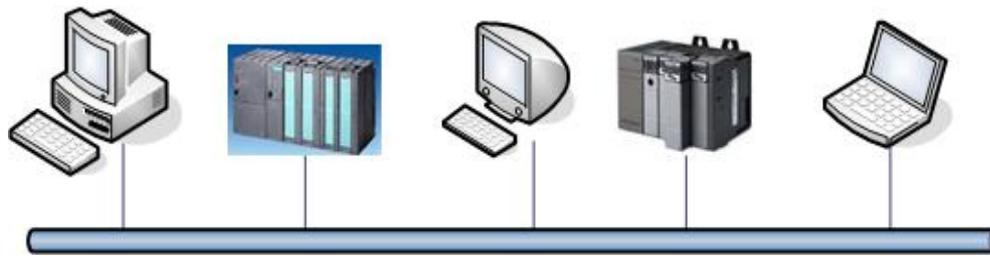


**Figura 7 : Conexión de una red configurada en topología anillo.**

c) **Topología tipo Bus.-** En esta distribución todas las estaciones se conectan a un medio de transmisión común a través de interfaces pasivas. Su alta fiabilidad radica en que las interfaces de enlace son pasivas, no afectando el funcionamiento global de la red en caso de falla. Cuando se desea insertar más estaciones es necesario tener presente las características físicas del medio de transmisión, sus características y representación en la **Tabla 3** y en la **Figura 8** respectivamente.

<b>Tipo de control</b>	Control centralizado y descentralizado. Acceso pasa de equipo a equipo.
<b>Redundancia</b>	Si falla algún equipo (nodo), esto no afecta el funcionamiento de la red.
<b>Expansibilidad</b>	Ilimitado teóricamente, pero el tiempo para acceder secuencialmente a todos los nodos es un límite práctico.
<b>Requerimiento de cableado</b>	Cable par trenzado apantallado.
<b>Requerimiento de interface</b>	Debe de proveer inmunidad a interferencias en la transmisión.

**Tabla 3 : Características principales de red configurada en topología bus**



**Figura 8 : Conexión de una red configurada en topología bus**

### 2.6.3 INTERFACES DE CONECTIVIDAD EN LAS COMUNICACIONES INDUSTRIALES

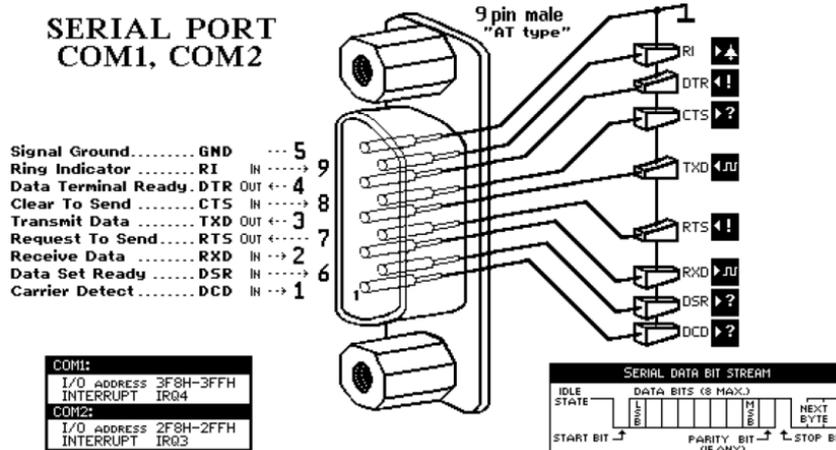
Las interfaces de comunicación serial son elementos que permiten la transmisión de información de un **Equipo Terminal de Datos (DTE)** hacia el medio de transmisión (cable coaxial, apantallado, fibra óptica, etc.) por el cual se va a comunicar con otro DTE.

Es importante que la transmisión de datos vía este medio, estén sujetos a estándares internacionales para asegurar la compatibilidad entre equipos de diferentes fabricantes.

**a) Interface RS-232.-**(ver **Figura 9**) Es una interface común, sobre todo en aplicaciones informáticas pero prácticas cuando se trata de controladores embebidos en máquinas inteligentes las cuales poseen un HMI enlazada a dicho controlador. La interface funciona a tensión de +/- 12 v de transmisión serial.

Estas son sus características más resaltantes:

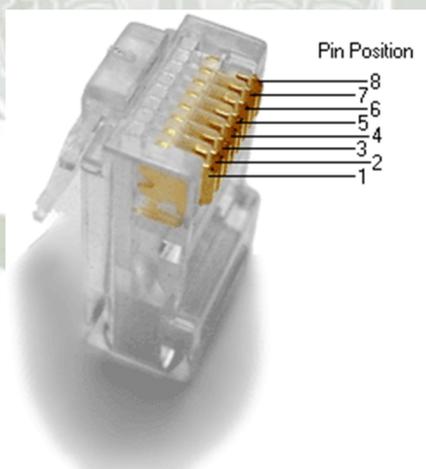
- La distancia entre estaciones a transmitir debe ser menor a 15 metros.
- La velocidad de transmisión puede ser menor o igual a 19200 bps.
- El conector utilizado consta de 9 o 25 pines.
- Sólo permite tener un transmisor y un receptor de manera dúplex.
- Existe dos hilos para la comunicación de datos, pin 2 para la transmisión y pin 3 para la recepción. El retorno, tanto para las señales de datos como para el control, se da a través del hilo conectado al pin 7.



**Figura 9 : Conector DB-9 como interfaz RS-232**

b) **RJ-45.-** (ver **Figura 10**) La RJ-45 es una interfaz física comúnmente usada para conectar redes de cableado estructurado, (categorías 4, 5, 5e y 6). RJ es un acrónimo inglés de Registered Jack que a su vez es parte del Código Federal de Regulaciones de Estados Unidos. Posee ocho "pines" o conexiones eléctricas, que normalmente se usan como extremos de cables de par trenzado.

Es utilizada comúnmente con estándares como TIA/EIA-568-B, que define la disposición de los pines o wiring pinout. Una aplicación común es su uso en cables de red Ethernet, donde suelen usarse 8 pines (4 pares).



**Figura 10 : Conector RJ-45**

#### 2.6.4 EL MODELO ISO/OSI

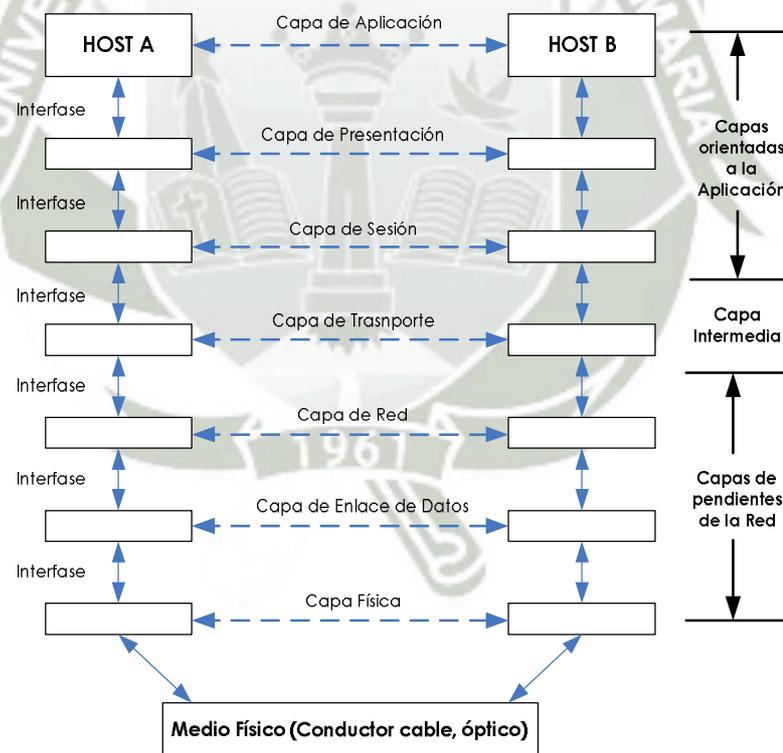
A finales de 1982, la Organización Internacional para la Normalización (ISO de las siglas en inglés International Standard Organization) desarrolló un modelo conceptual

que permite la conexión entre sistemas cuyo nombre es **Open Systems Interconnection Reference Model** o Modelo de Referencia de Interconexión de Sistemas Abiertos. En los entornos de trabajo con redes, se le conoce más comúnmente como el modelo OSI.

En 1984, este modelo pasó a ser el estándar internacional para las comunicaciones en red al ofrecer un marco de trabajo conceptual que permita explicar el modo en que los datos se desplazaban dentro de una red.

Así, el objetivo planteado, fue alcanzado por esta presentación enfocada en capas, específicamente dividida 7 capas representadas en la **Figura 11**.

Las 3 capas más bajas son dependientes de la red (network). Ellos proveen de soporte para la comunicación y el enlace entre dos sistemas (o usuarios). Las 3 capas superiores están orientadas a la aplicación, permitiendo a los usuarios finales procesar la aplicación para interactuar uno del otro. Finalmente la capa intermedia (la capa de transporte), aísla las capas orientadas a la aplicación de las de comunicación.



**Figura 11 : Modelo de referencia ISO/OSI**

Cada capa desarrolla una función bien definida. Esto permite la reducción de los niveles de complejidad en cada capa, definida así mismo por cada una de ellas. Por lo



tanto la información fluye entre las capas, por cada interface, las cuales deberían ser minimizadas en su complejidad. Cada capa intercambia mensajes usando los servicios de la capa anterior, realizando una comunicación de manera remota y proveyendo de servicios a la capa subsiguiente. En cada capa se añade al paquete fuente una señalización denominada cabecera, la cual es leída y reconocida nuevamente por el receptor, es importante notar que la implementación de una capa es por lo tanto independiente de la implementación de otras capas.

El modelo OSI abarca una serie de eventos importantes que se producen durante la comunicación entre sistemas proporcionando las normas básicas empíricas para una serie de procesos distintos de conexión en red, así por ejemplo podemos presentarlas a continuación:

- El modo en que los datos se traducen a un formato apropiado para la arquitectura de red que se esté utilizando. Cuando se envía un mensaje de correo electrónico por ejemplo, o un simple archivo de una computadora a otra, se está trabajando en realidad, con una determinada aplicación, como lo es un cliente de correo electrónico o un cliente FTP. Los datos que se transmiten utilizando dicha aplicación tienen que convertirse a un formato más genérico si van a viajar por la red hasta llegar a su destino.
- El modo en que los PC u otro dispositivo como un PLC en una red se comunican. Cuando se envían datos desde una PC, tiene que existir algún tipo de mecanismo que proporcione un canal de comunicación entre el remitente (sender) y el destinatario (receiver). Lo mismo que cuando se desea hablar por teléfono, para lo cual hay que descolgar el teléfono y marcar el número.
- El modo en que los datos se transmiten entre los distintos dispositivos y la forma en que se resuelve la secuencia y comprobación de errores. Una vez establecida la sesión de comunicación entre computadores y/o controladores, tiene que existir un conjunto de reglas que controlen la forma en que los datos van de uno a otro.
- El modo en que el direccionamiento lógico de los paquetes pasa a convertirse en el direccionamiento físico que proporciona a la red. Las redes informáticas por ejemplo utilizan esquemas de direccionamiento lógico como direcciones IP,

por tanto, dichas direcciones lógicas tiene que convertirse en las direcciones reales de hardware.

- Así el modelo OSI ofrece los mecanismos y reglas que permiten resolver todas las cuestiones que se acaban de mencionar. Comprender las distintas capas del modelo OSI no sólo permite internarse en los conjuntos de protocolos de red que actualmente se utilizan, sino que también proporciona un marco de trabajo conceptual del que puede servirse cualquiera para comprender el funcionamiento de dispositivos de red complejas, como Dispositivos de control industrial, variadores de velocidad, PLC's, computadores industriales, conmutadores, etc.

#### 2.6.4.1 ANALOGÍA DE FUNCIONAMIENTO DE UNA RED EN LA COMUNICACIÓN DE DIGITAL

En conclusión como análisis hasta acá sobre las comunicaciones digitales, hagamos una analogía. Se puede pensar por un momento en el servicio de correos:

Cuando alguien desea mandar una carta a otra persona, la escribe, la mete en un sobre con el formato impuesto por correos, le pone un sello y la introduce en un buzón; la carta es recogida por el cartero, clasificada por el personal de correos, según su destino y enviada a través de medios de transporte hacia la ciudad destino; una vez allá otro cartero irá a llevarla a la dirección indicada en el sobre; si la dirección no existe, al cabo del tiempo la carta devolverá al origen por los mismos cauces que llegó al supuesto destino.

Más o menos, esta es la forma en que funciona una red: "La carta escrita es la información que se quiere transmitir; el sobre y sello es el paquete con el formato impuesto por el protocolo que se utiliza en la transmisión; la dirección del destinatario es la dirección del nodo destino

Más o menos, esta es la forma en que funciona una red: "La carta escrita es la información que se quiere transmitir; el sobre y sello es el paquete con el formato impuesto por el protocolo que se utiliza en la transmisión; la dirección del destinatario es la dirección del nodo destino y la dirección del remitente, será la dirección del nodo origen, los medios de transporte que llevan la carta cerca del destino es el medio de transmisión (cable coaxial, fibra óptica, UTP, etc.); las normas del servicio de correos, carteros y demás personal son los protocolos de comunicaciones establecidos".

Si se supone que se está utilizando el modelo OSI de la ISO. Este modelo tiene 7 niveles, es como decir que la carta escrita pasa por 7 filtros diferentes (trabajadores con diferentes cargos) desde que la ponemos en el buzón hasta que llega al destino.

Cada nivel de se encarga de realizar funciones diferentes en la información a transmitir. Cada nivel por el que pasa la información a transmitir que se ha insertado en un paquete, añade información de control, que el mismo nivel en el nodo destino irá eliminando. Además se encarga de cosas muy distintas: desde el control de errores, hasta la reorganización de la información transmitida cuando esta se ha fragmentado en tramas.

Si la información va dirigida a una red diferente (otra ciudad en el caso de la carta), la trama debe llegar a un dispositivo de interconexión de redes (router, gateway, bridges), que decidirá, dependiendo de su capacidad, el camino que debe seguir la trama. Por eso es imprescindible que el paquete lleve la dirección destino y que esta contenga, además de la dirección que idéntica al nodo y la dirección que identifica la red a la que pertenece el nodo.

---

## 2.6.5 ETHERNET

---

### 2.6.5.1 DEFINICION

---

Ethernet es una tecnología de redes ampliamente aceptada con conexiones disponibles para PCs, estaciones de trabajo científicas y de alta desempeño, mini computadoras y sistemas mainframe.

La arquitectura Ethernet provee detección de errores pero no corrección de los mismos. Tampoco posee una unidad de control central, todos los mensajes son transmitidos a través de la red a cada dispositivo conectado. Cada dispositivo es responsable de reconocer su propia dirección y aceptar los mensajes dirigidos a ella. El acceso al canal de comunicación es controlado individualmente por cada dispositivo utilizando un método de acceso probabilístico conocido como disputa (contention).

### 2.6.5.2 OBJETIVOS DE ETHERNET

---

Los objetivos principales de Ethernet son consistentes con los que se han convertido en los requerimientos básicos para el desarrollo y uso de redes LAN.

Los objetivos originales de Ethernet son:



- a) **Simplicidad.** Las características que puedan complicar el diseño de la red sin hacer una contribución substancial para alcanzar otros objetivos se han excluido.
- b) **Bajo Costo.** Las mejoras tecnológicas van a continuar reduciendo el costo global de los dispositivos. de conexión.
- c) **Compatibilidad.** Todas las implementaciones de Ethernet deberán ser capaces de intercambiar datos a nivel de capa de enlace de datos. Para eliminar la posibilidad de variaciones incompatibles de Ethernet, la especificación evita características opcionales.
- d) **Direccionamiento flexible.** El mecanismo de direccionamiento debe proveer la capacidad de dirigir datos a un único dispositivo, a un grupo de dispositivos, o alternativamente, difundir (broadcast) el mensaje a todos los dispositivos conectados a la red.
- e) **Equidad.** Todos los dispositivos conectados deben tener el mismo acceso a la red.
- f) **Progreso.** Ningún dispositivo conectado a la red, operando de acuerdo al protocolo Ethernet, debe ser capaz de prevenir la operación de otros dispositivos.
- g) **Alta velocidad.** La red debe operar eficientemente a una tasa de datos de 10 Mb/s.
- h) **Bajo retardo.** En cualquier nivel de tráfico de la red, debe presentarse el mínimo tiempo de retardo posible en la transferencia de datos.
- i) **Estabilidad.** La red debe ser estable bajo todas las condiciones de carga. Los mensajes entregados deben mantener un porcentaje constante de la totalidad del tráfico de la red.
- j) **Arquitectura en capas.** El diseño Ethernet debe ser especificado en término de capas de forma de separar las operaciones lógicas de los protocolos de capa de enlace de las especificaciones de comunicaciones físicas del canal de comunicación.

---

### 2.6.6 TCP/IP

---

Bajo las siglas TCP/IP (Transfer Control Protocol/Internet Protocol) se agrupa un paquete de protocolos de comunicación de datos. El paquete toma este nombre por dos de los protocolos que lo integran, el TCP, o Protocolo de Control de Transferencia, y el IP, o Protocolo de Internet, dos de los más importantes protocolos que podemos hallar

en dicho paquete. Teniendo esto en cuenta, desde ahora nos referiremos a dicho paquete como a los protocolos TCP/IP, en plural.

#### 2.6.6.1 MOTIVOS PARA TRABAJAR CON ENLACES DE REDES

---

- La comunicación de datos se ha convertido en parte fundamental de la computación.
- Las redes globales reúnen datos sobre sistemas diversos.
- La mayor parte de las redes son entidades independientes.

Una nueva tecnología hace posible interconectar muchas redes físicas diferentes y hacerlas funcionar como una unidad coordinada, llamada internetworking, unificando diferentes tecnologías de HARDWARE subyacentes al proporcionar un conjunto de normas de comunicación y una forma de interconectar redes heterogéneas. La tecnología de red de redes oculta los detalles del hardware de red y permite que las computadoras se comuniquen en forma independiente de sus conexiones físicas de red.

Los enlaces de redes son Sistemas Abiertos porque las especificaciones están disponibles públicamente, cualquier persona puede desarrollar el software necesario para comunicarse a través de una red de redes.

#### 2.6.6.2 CARACTERÍSTICAS DE TCP/IP

---

Los protocolos TCP/IP presentan las siguientes características:

- Son estándares de protocolos abiertos y gratuitos. Su desarrollo y modificaciones se realizan por consenso, no a voluntad de un determinado fabricante. Cualquiera puede desarrollar productos que cumplan sus especificaciones.
- Independencia a nivel software y hardware Su amplio uso los hace especialmente idóneos para interconectar equipos de diferentes fabricantes, no sólo a Internet sino también formando redes locales. La independencia del hardware nos permite integrar en una sola varios tipos de redes (Ethernet, Token Ring, X.25...)
- Proporcionan un esquema común de direccionamiento que permite a un dispositivo con TCP/IP localizar a cualquier otro en cualquier punto de la red.
- Son protocolos estandarizados de alto nivel que soportan servicios al usuario y son ampliamente disponibles y consistentes.



### 2.6.7 PUERTOS TCP/IP

La dirección IP sirve para identificar de manera única un equipo en la red mientras que el número de puerto especifica la aplicación a la que se dirigen los datos. Así, cuando el equipo recibe información que va dirigida a un puerto, los datos se envían a la aplicación relacionada. Si se trata de una solicitud enviada a la aplicación, la aplicación se denomina aplicación servidor. Si se trata de una respuesta, entonces hablamos de una aplicación cliente. Existen miles de puertos (codificados en 16 bits, es decir que se cuenta con 65536 posibilidades). Es por ello que la IANA (Internet Assigned Numbers Authority - Agencia de Asignación de Números de Internet) desarrolló una aplicación estándar para ayudar con las configuraciones de red.

Los puertos del 0 al 1023 son los "puertos conocidos" o reservados. En términos generales, están reservados para procesos del sistema o programas ejecutados por usuarios privilegiados. Sin embargo, un administrador de red puede conectar servicios con puertos de su elección.

- Los puertos del 1024 al 49151 son los "puertos registrados".
- Los puertos del 49152 al 65535 son los "puertos dinámicos y/o privados".

En la siguiente **Tabla 4** se indican algunos de los puertos conocidos más utilizados:

Puerto	Servicio o aplicación
21	FTP (File Transfer Protocol - Protocolo de transferencia de archivos)
23	Telnet (TELEcommunication NETwork Cliente Terminal)
25	SMTP (Simple Mail Transfer Protocol Protocolo Simple de Transferencia de Correo)
53	DNS (Domain Name System Sistema de nombre de dominio)
80	HTTP (HyperText Transfer Protocol protocolo de transferencia de hipertexto)
110	POP3 (Post Office Protocol Protocolo de Correo versión 3)

**Tabla 4: Designación de puertos TCP/IP reservados para aplicaciones**



## 2.7 CONTROLADOR INDUSTRIAL DE PROCESOS

---

El primer Control Lógico Programable (Programmable Logic Control o PLC) fue desarrollado por un grupo de ingenieros en la General Motors en 1968<sup>10</sup>, cuando la empresa estaba buscando una alternativa para reemplazar los complejos sistemas de control por relés.

El nuevo sistema de control tenía que cumplir con los siguientes requerimientos:

- Programación sencilla
- Cambios de programa sin intervención en el sistema (sin tener que rehacer el cableado interno)
- Más pequeño, más económico y más fiable que los correspondientes sistemas de control por relés
- Sencillo y con bajo coste de mantenimiento

Los sucesivos desarrollos llevaron a un sistema que permitía la conexión sencilla de señales binarias. Los requerimientos de cómo estaban conectadas estas señales se especificaba en el programa de control. Con los nuevos sistemas, fue posible por primera vez mostrar las señales en una pantalla y archivar los programas en memorias electrónicas. Desde entonces han pasado tres décadas, durante las cuales los enormes progresos hechos en el desarrollo de la micro-electrónica han favorecido la proliferación de los controles lógicos programables. Por ejemplo, a pesar de que en sus comienzos, la optimización del programa y con ello la necesidad de reducir la ocupación de memoria representaba una tarea importante para el programador, en la actualidad esto apenas tiene importancia. Además, las funciones disponibles han crecido considerablemente.

Hace 25 años, la visualización de procesos, el procesamiento analógico o incluso la utilización de un PLC como un regulador, eran considerados una utopía. Actualmente, muchos de estos elementos son parte integral de muchos PLC's.

---

<sup>10</sup>[17] Manual Teórico PLC / 1.1 Introducción (<http://es.scribd.com/doc/77252987/Manual-Teorico-PLC#download>)

---

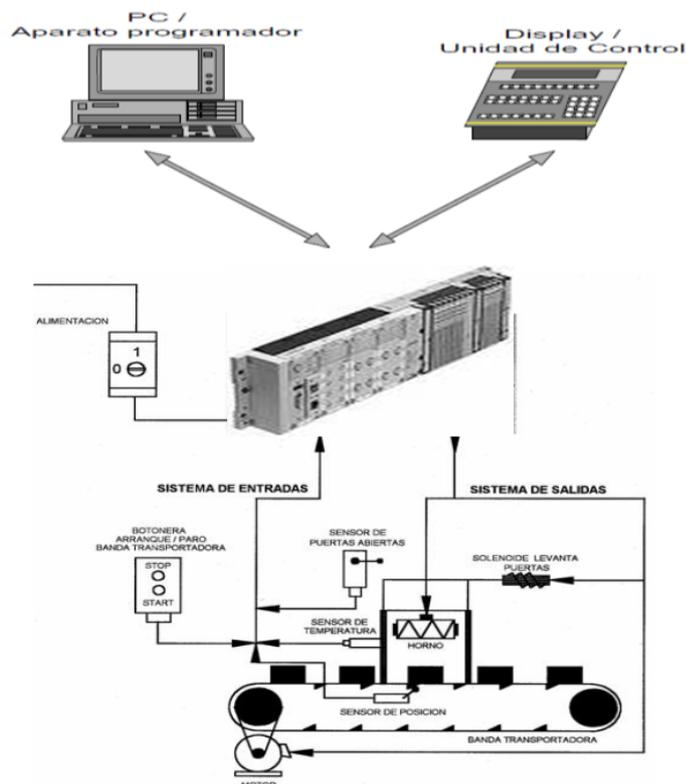
## 2.7.1 ÁREAS DE APLICACIÓN DEL CONTROLADOR LÓGICO PROGRAMABLE (PLC)

---

Todas las máquinas o sistemas automáticos tienen un control. Dependiendo del tipo de tecnología utilizada, los controles pueden dividirse en neumáticos, hidráulicos, eléctricos y electrónicos. Con frecuencia se utiliza una combinación de las diferentes tecnologías (sistema mecatrónico). Además, debe distinguirse entre controles con programa cableado (es decir, conexión física de componentes electromecánicos (relés, etc.) o componentes electrónicos (circuitos integrados)) y controles lógicos programables. Los primeros se utilizan principalmente en casos en los que la reprogramación por el usuario está fuera de toda duda y el alcance de la tarea justifica el desarrollo de un sistema de control especial. Las aplicaciones típicas de tales controles pueden hallarse en los electrodomésticos, vídeo cámaras, vehículos, etc.

Sin embargo, si la tarea de control no justifica el desarrollo de un control especial, o si el usuario debe tener la posibilidad de hacer cambios sencillos, o de modificar tiempos o valores de contadores, entonces el uso de un control universal, en el que el programa se escribe en una memoria electrónica, es la opción preferida. El PLC representa un control universal. Puede utilizarse para diferentes aplicaciones y, dado que el programa se halla escrito en su memoria electrónica, el usuario puede modificar, ampliar y optimizar con cierta sencillez sus procesos de control.

La tarea original de un PLC es la interconexión de señales de entrada, de acuerdo con un determinado programa y, si el resultado de esta interconexión es "cierta", activar la correspondiente salida. El álgebra de Boole forma la base matemática para esta operación, ya que solamente reconoce dos estados definidos de una variable: "0" (falso) y "1" (cierto). Consecuentemente, una salida sólo asume estos dos estados. Por ejemplo, una electroválvula conectada a la salida puede estar activada o desactivada, es decir, controlada; su relación con los demás periféricos de entrada y salida lo observamos en el siguiente **Figura 12**.



**Figura 12: PLC industrial con terminales electro neumáticos MPA, en una aplicación industrial.**

Esta función ha acuñado el nombre de PLC: Programmable Logic Control o Control Lógico Programable. En él, el comportamiento de las entradas/salidas es similar al de los controles realizados con relés electromagnéticos o con elementos lógicos neumáticos o electrónicos; la diferencia reside en que el programa en lugar de estar 'cableado' está almacenado en una memoria electrónica.

Sin embargo las tareas del PLC se ampliaron rápidamente: las funciones de temporización y recuento, operaciones de cálculo matemático, conversión de señales analógicas, etc. representan funciones que pueden ejecutarse en casi todos los PLC's actuales.

Las demandas que se requieren de los PLC's siguen creciendo al mismo ritmo que su amplia utilización y desarrollo en la tecnología de automatización. Por ejemplo: la visualización, es decir, la representación de los estados de las máquinas o la supervisión de la ejecución del programa por medio de una pantalla o monitor. También el control directo, es decir, la facilidad de intervenir en los procesos de control o, alternativamente, impedir tal intervención a las personas no autorizadas. También se

ha visto la necesidad de interconectar y armonizar sistemas individuales controlados por PLC, por medio de redes o buses de campo. Aquí, un ordenador máster permite la generación de órdenes de mayor nivel para el procesamiento de programas en los diversos sistemas PLC interconectados.

La conexión en red de varios PLC's, así como la de un PLC con el ordenador máster se realiza por medio de interfaces de comunicación especiales. Para ello, la mayoría de los más recientes PLC's son compatibles con sistemas de bus abiertos estandarizados, tales como Profibus según DIN 19 245. Gracias al enorme aumento de la potencia y capacidad de los PLC's avanzados, estos pueden incluso asumir directamente la función de un ordenador máster.

Hacia finales de los setenta, las entradas y salidas binarias fueron finalmente ampliadas con la adición de entradas y salidas analógicas, ya que hay muchas aplicaciones técnicas que emiten y requieren señales analógicas (medición de fuerzas, velocidades, sistemas de posicionado servo neumáticos, etc.). Al mismo tiempo la adquisición y emisión de señales analógicas permite la comparación de valores reales con los de consigna y, como consecuencia, la realización de funciones de regulación automática; una tarea que va más allá del ámbito que sugiere el nombre de control lógico programable.

Los PLC's que existen actualmente el mercado han sido adaptados a los requerimientos de los clientes hasta tal punto que ya es posible adquirir un PLC exactamente adaptado para casi cada aplicación. Así, hay disponibles actualmente desde PLC's en miniatura con unas decenas de entradas/salidas hasta grandes PLC's con miles de entradas/salidas. Muchos PLC's pueden ampliarse por medio de módulos adicionales de entradas/salidas, módulos analógicos y de comunicación. Hay PLC's disponibles para sistemas de seguridad, barcos o tareas de minería.

Otros PLC's son capaces de procesar varios programas al mismo tiempo (Multitarea). Finalmente, los PLC's pueden conectarse con otros componentes de automatización, creando así áreas considerablemente amplias de aplicación.

---

### 2.7.2 CARACTERÍSTICAS DESTACADAS DE UN PLC

---

- Tecnología de banda ancha.
- Velocidades de transmisión de hasta 45 Megabits por segundo (Mbps).



- Proceso de instalación sencillo y rápido para el cliente final.
- Enchufe eléctrico; toma única de alimentación, voz y datos.
- Sin necesidad de obras ni cableado adicional.
- Equipo de conexión (Modem PLC).
- Transmisión simultánea de voz y datos.
- Conexión de datos permanente (activa las 24 horas del día).
- Permite seguir prestando el suministro eléctrico sin ningún problema.

---

### 2.7.3 VENTAJAS

---

Dentro de las ventajas que estos equipos poseen se encuentra que, gracias a ellos, es posible ahorrar tiempo en la elaboración de proyectos, pudiendo realizar modificaciones sin costos adicionales. Por otra parte, son de tamaño reducido y mantenimiento de bajo costo, además permiten ahorrar dinero en mano de obra y la posibilidad de controlar más de una máquina con el mismo equipo.

Sin embargo, y como sucede en todos los casos, los controladores lógicos programables, o PLC's, presentan ciertas desventajas como es la necesidad de contar con técnicos calificados y adiestrados específicamente para ocuparse de su buen funcionamiento.

---

### 2.7.4 DEFINICIÓN BÁSICA DE UN PLC

---

El término 'Control Lógico Programable' se define como sigue<sup>11</sup>:

"Un aparato electrónico de operación digital, el cual utiliza una memoria programable que almacena internamente las instrucciones para implementar funciones específicas, tales como: lógicas, secuenciales, de tiempo, de conteo, y aritméticas; para controlar por medio de entradas y salidas digitales o analógicas, varios tipos de máquinas o procesos."

Por lo tanto, un control lógico programable es sencillamente un ordenador, adaptado específicamente para ciertas tareas de control.

---

<sup>11</sup>Definición de la Asociación Nacional (EUA) de Manufactureros Eléctricos (NEMA), 1978

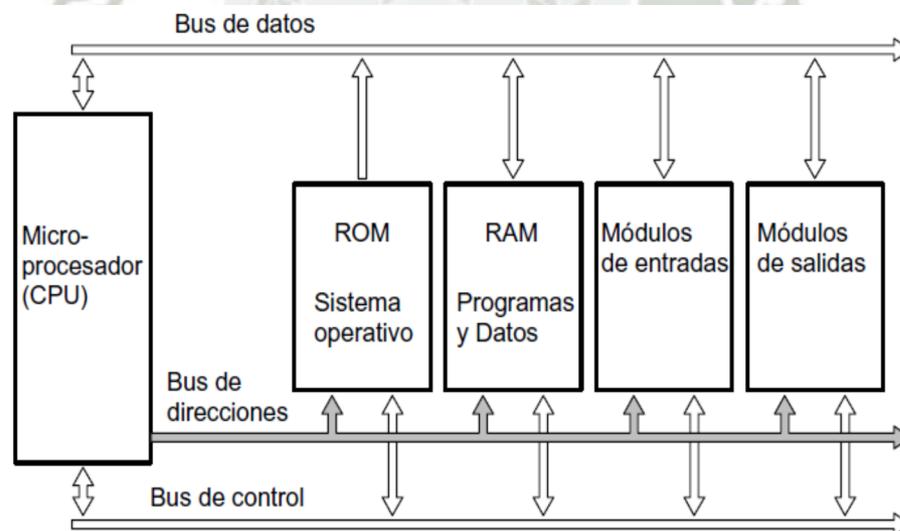
### 2.7.5 ESTRUCTURA DE UN PLC

En los ordenadores, generalmente se distingue entre hardware, firmware y software. Lo mismo se aplica a los PLC's, ya que esencialmente también están basados en un microprocesador.

El Hardware se refiere a las partes físicas del dispositivo, el decir, los circuitos impresos, los circuitos integrados, el cableado, la batería, el chasis, etc.

El firmware los constituyen aquellos programas (software) que se hallan permanentemente instalados en el hardware del ordenador y que son suministrados por el fabricante del PLC. Esto incluye las rutinas fundamentales del sistema, utilizadas para poner en marcha el procesador al aplicar la tensión. Adicionalmente, hay el sistema operativo que, en el caso de los controles lógicos programables, generalmente se halla almacenado en una memoria ROM de sólo lectura o en una EPROM.

Finalmente, hay el software, que es el programa escrito por el usuario del PLC. Los programas de usuario se instalan generalmente en la memoria RAM, una memoria de acceso aleatorio, en donde pueden ser fácilmente modificados.



**Figura 13 : Estructura interna clásica de un PLC**

La **Figura 13** ilustra el diseño fundamental de un microordenador. El hardware del PLC o como es el caso de casi todos los sistemas microordenadores actuales – está basado en un sistema de bus. Un sistema de bus es un determinado número de líneas eléctricas divididas en líneas de direcciones, de datos y de control. La línea de direcciones se

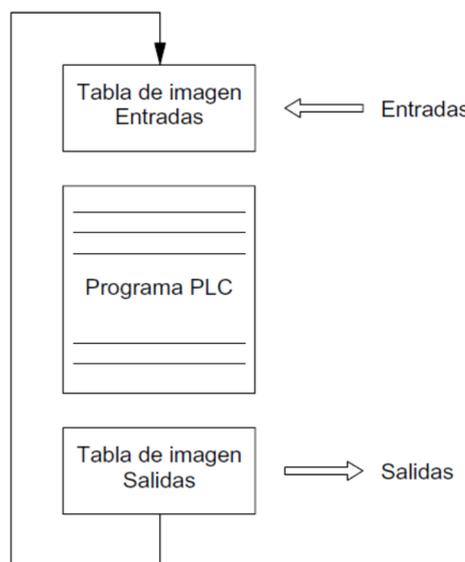
utiliza para seleccionar la dirección de un elemento conectado al bus y la línea de datos para transmitir la información requerida. Las líneas de control son necesarias para habilitar el dispositivo conectado al bus como emisor o como receptor. Los principales elementos conectados al sistema de bus son el microprocesador y la memoria. La memoria puede dividirse en memoria para el firmware y memoria para el programa y los datos del usuario.

Según la estructura del PLC, los módulos de entradas y salidas se conectan a un simple bus común o – con la ayuda de un interface de bus – a un bus externo de E/S. Especialmente en el caso de grandes sistemas modulares de PLC, es más usual un bus externo de E/S.

Finalmente, se necesita una conexión para el aparato programador o un PC, actualmente y en la mayoría de los casos en forma de un interface serie.

#### 2.7.6 MODO DE FUNCIONAMIENTO DE UN PLC

Los programas para el procesamiento convencional de datos, generalmente se procesan una sola vez, de arriba a abajo y terminan. A diferencia de estos, el programa de un PLC se procesa continuamente y cíclicamente, ver **Figura 14**.



**Figura 14 : Ciclo de funcionamiento de un PLC**

Las características del procesamiento cíclico son:

- Así que el programa ha sido ejecutado una vez, salta automáticamente al principio y se va repitiendo el proceso continuamente.
- Antes de que se procese la primera línea del programa, es decir, al inicio del ciclo, el estado de las entradas es almacenado en la tabla de imagen de entradas. La imagen del proceso es una zona de memoria aparte a la que se accede durante un ciclo. Así, el estado lógico de una entrada permanece constante durante un ciclo, incluso aunque en este intervalo haya cambiado físicamente.
- De forma similar a la entradas, las salidas no son inmediatamente activadas o desactivadas durante un ciclo, sino que su estado es almacenado temporalmente en la tabla imagen de salidas. Solamente al final del ciclo se activan o desactivan físicamente las salidas según el estado lógico almacenado en la memoria. El procesamiento de una línea de programa a través de la unidad central de un PLC ocupa un tiempo que, dependiendo del PLC y de la instrucción que contenga puede variar desde unos pocos microsegundos hasta unos pocos milisegundos.

El tiempo requerido por el PLC para una simple ejecución de un programa, incluyendo la actualización de las salidas y la imagen del proceso, se denomina tiempo de ciclo o tiempo de "scan". Cuanto más largo sea el programa y cuanto más tiempo necesite el PLC respectivo para procesar cada línea del programa, tanto más largo será el tiempo de ciclo. Los tiempos reales de ciclo varían aproximadamente entre 1 y 100 milisegundos.

Las consecuencias del procesamiento cíclico de un programa de PLC que utilice una imagen del proceso son las siguientes:

- Las señales de entrada de una duración inferior al tiempo de ciclo, posiblemente no serán reconocidas.
- En algunos casos, puede haber un retardo de dos ciclos entre la presencia de una señal de entrada y la deseada reacción de una salida ante esta señal.
- Dado que las instrucciones se procesan secuencialmente, el comportamiento específico de la secuencia de un programa de PLC puede ser crucial.

En algunas aplicaciones, es esencial que pueda accederse directamente a entradas y salidas durante un ciclo. Por ello, este tipo de procesamiento de programa, saltándose la imagen del proceso, también es posible en algunos sistemas PLC.



### 2.7.7 DISPOSITIVO PROGRAMADOR DE UN PLC

Cada PLC tiene una herramienta de diagnóstico y programación para soportar la aplicación del PLC. Estas herramientas de programación y diagnóstico son o bien dispositivos de programación específicos del fabricante u ordenadores personales con su software correspondiente. Actualmente, estos últimos son casi exclusivamente las variantes preferidas, ya que la enorme capacidad de los modernos PCs, combinada con su comparativamente bajo coste inicial y alta flexibilidad, representan ventajas cruciales.

También se han desarrollado los denominados programadores de mano para pequeños sistemas de control y para tareas de mantenimiento. Con la creciente utilización de ordenadores personales portátiles (LapTop), funcionando con baterías, la importancia de los programadores de mano disminuye paulatinamente.

Las funciones esenciales del sistema de software forman parte de la herramienta de programación y diagnóstico. Cualquier software de programación según IEC 1131-1 deben proporcionar al usuario una serie de funciones. Así, el software de programación comprende módulos de software para:

- **Introducción de programas.-** Creación y modificación de programas en uno de los lenguajes de programación de un PLC
- **Verificación de la sintaxis.-** Comprobación de la sintaxis del programa y los datos, minimizando así la introducción de programas defectuosos.
- **Traductor.-** Traducción del programa introducido en un programa que puede ser leído y procesado por el PLC, es decir, la generación del código máquina del correspondiente PC
- **Conexión entre PLC y PC.-** A través de este enlace se realiza la carga de los programas al PLC y la ejecución de funciones de verificación.
- **Funciones de verificación.-** Ayuda al usuario durante la escritura y en la eliminación de fallos y verificación a través de:
  - Una verificación del estado de las entradas, salidas temporizadores, contadores, etc.
  - Verificación de secuencias de programa por medio de operaciones de paso a paso, órdenes de STOP, etc.

- Simulación por medio de activación manual de entradas/salidas, establecimiento de valores, etc.
- **Indicación del estado de sistemas de control.**- Emisión de información relacionada con la máquina, proceso y estado del sistema PLC:
  - Indicación del estado de señales de entrada y salida.
  - Registro/indicación de cambios de estado en señales externas y datos internos.
  - Supervisión de los tiempos de ejecución.
  - Formato en tiempo real de la ejecución del programa.
- **Documentación.**- Creación de una descripción del sistema PLC y el programa del usuario. Esto consiste en:
  - Descripción de la configuración del hardware.
  - Impresión del programa de usuario con los correspondientes datos e identificadores para las señales y comentarios.
  - Lista de referencias cruzadas para todos los datos procesados tales como entradas, salidas, temporizadores, etc.
  - Descripción de las modificaciones.
  - Archivado del programa de usuario.
  - Protección del programa de usuario en memorias no volátiles tales como EPROM, etc.

---

### 2.7.8 PLC SIEMENS S7-200<sup>12</sup>

---

La CPU S7-200 es un equipo autónomo, compacto y robusto que está compuesto por:

- Una unidad central de proceso CPU,
- Una fuente de alimentación
- Entradas y salidas digitales (y analógicas en CPU 224XP).

La CPU ejecuta el programa y almacena los datos referentes al proceso a automatizar.

El sistema controla mediante entradas o salidas digitales (E/S). Las entradas vigilan las señales de los dispositivos de campo (por ej. Sensores, interruptores, transmisores de señal), mientras que las salidas comandan a los actuadores (bombas, motores u otros

---

<sup>12</sup>[16] Siemens, Manual del sistemas de Automatización S7-200  
<http://isa.uniovi.es/~vsuarez/Download/Simatic%20S7200.pdf>

aparatos del proceso). Las CPUs S7-200 tienen integrada una fuente de alimentación capaz de abastecer la CPU, los módulos de ampliación y otras cargas que precisen DC 24 V.

La CPU S7-200 suministra la corriente continua de 5 V necesaria para los módulos de ampliación del sistema. Los LEDs indican el modo de operación de la CPU (RUN o STOP), el estado de las entradas y las salidas físicas, así como los posibles fallos del sistema que se haya detectado (SF).

Utilizando módulos de ampliación se pueden agregar entradas y salidas (E/S) digitales o analógicas, adicionales a la CPU. (La CPU 224 tiene la capacidad de ampliación de hasta 7 módulos).

Los módulos de comunicaciones mejoran la performance de la comunicación.

Un cartucho enchufable EEPROM (opcional) sirve para almacenar todo el programa de la CPU, así como las áreas de datos de usuario y los datos de configuración y transferirlo de una CPU a otra, o enviarlo por correo a otra parte del mundo.

El S7-200 soporta también un cartucho de pila opcional que prolonga el tiempo durante el que se puede respaldar la memoria RAM después de un corte de alimentación. El cartucho de pila se activa sólo cuando se descarga el condensador de alto rendimiento.

Algunas CPUs tienen un reloj de tiempo real integrado, en tanto que otras pueden disponer de un cartucho (opcional) de reloj de tiempo real. La finalidad del mismo es contar horas de funcionamiento, etiquetar mensajes con fecha y hora (tiene precisión al segundo, aun en años bisiestos), entre otras aplicaciones

---

## 2.7.9 PUESTA A PUNTO DE UN PLC

---

Los programas de PLC nunca se terminan, en el sentido de que siempre es posible hacer correcciones y las consiguientes adaptaciones a los nuevos requerimientos del sistema. Incluso durante la puesta a punto, a menudo son necesarios los cambios en el programa. La puesta a punto de un sistema puede dividirse básicamente en cuatro etapas:

---

### 2.7.9.1 VERIFICACIÓN DEL HARDWARE

---

Cada sensor se conecta a una entrada específica y cada actuador a una salida; las direcciones no deben mezclarse. Esta es la razón por la que el primer paso en la verificación del hardware siempre debe hacerse tras la lista de asignaciones. ¿Están



todos los sensores y actuadores asignados a las direcciones de entrada y salida correctas? ¿Se identifica inequívocamente la función – señal-0 y señal-1? La lista de asignaciones debe corregirse y completarse ya que forma parte de la documentación antes de la puesta a punto de un programa.

Durante la verificación, las salidas se activan para comprobarlas. Los actuadores deben cumplir con las funciones especificadas.

#### 2.7.9.2 TRANSFERENCIA Y PRUEBAS DEL SOFTWARE

Incluso antes de la puesta a punto, todas las facilidades de verificación del programa sin conexión con el control, deberían utilizarse intensivamente.

Una de estas funciones de prueba muy práctica es, por ejemplo, la simulación del programa.

A continuación, el programa es transferido a la unidad central de control el PLC. Un pequeño número de PLC's ofrecen actualmente facilidades para la simulación: Todo el programa se ejecuta sin que haya entradas ni salidas conectadas. De forma similar, sólo la conexión de las salidas puede omitirse. Así, el procesamiento de las salidas del PLC se realiza solamente en la tabla de imagen, cuyos estados no son traspasados físicamente a las salidas disponibles. Esto, por lo tanto, elimina el riesgo de dañar partes de la máquina o del sistema, lo cual es de mucha importancia en el caso de procesos críticos o peligrosos.

Tras esto, las partes individuales del programa y las funciones del sistema se verifican: Funcionamiento manual, ajuste, programas de supervisión individuales, etc., y finalmente la interacción de las partes del programa con la ayuda de todo el conjunto.

Con ello, el programa es puesto a punto paso a paso. Aspectos importantes de la puesta a punto y la detección de errores son las funciones de test de los sistemas de programación, tales como el funcionamiento en paso a paso o el establecimiento de puntos de parada. El modo de paso a paso tiene una particular importancia, ya que con ello el programa se ejecuta en la memoria del PLC línea a línea o pasó a paso. De esta forma, cualquier error que pudiera haber en el programa puede ser localizado inmediatamente.



### 2.7.9.3 OPTIMIZACIÓN DEL SOFTWARE

---

Casi siempre, los programas largos pueden mejorarse tras la primera prueba de funcionamiento. Es importante que cualquier modificación o corrección se haga no tan sólo en el programa del PLC, sino que también se tenga en cuenta en la documentación. Además de la documentación, debe guardarse cada vez el nuevo estado del software

### 2.7.9.4 PUESTA EN MARCHA DEL SISTEMA

---

Esto ya se produce en parte durante la fase de optimización y verificación. Una vez que el estado final del programa PLC y la documentación han sido establecidos, hay que ejecutar de nuevo todas las funciones del control (de acuerdo con la tarea). Entonces el sistema está listo para ser aceptado por el cliente o el correspondiente departamento.

## 2.7.10 SEGURIDAD FUNCIONAL DE UN PLC

---

### 2.7.10.1 ALIMENTACIÓN DE TENSIÓN AL PLC

---

Debe distinguirse entre la **tensión de control** (tensión de las señales entre la máquina controlada y el PLC) y la tensión de la lógica (tensión de alimentación interna de la unidad central de control).

El nivel de tensión de funcionamiento de un PLC está especificado en DIN IEC 1131- /Parte 2. Se halla entre 24 VDC y 48 VDC o 48 VAC y 220 VAC respectivamente.

### 2.7.10.2 TENSIÓN DE CONTROL

---

La tensión de control alimenta a los sensores y actuadores. Para ello el usuario debe conectar una fuente de alimentación al control. La tensión de control de un PLC es generalmente de 24 VDC o 230 VAC.

(Principalmente se utiliza corriente continua (DC)). En algunos países, se utilizan también diferentes tensiones, p. ej., 48 VDC o 120 VAC. La potencia de la fuente de alimentación depende el control utilizado y de las entradas/salidas conectadas. El valor de la tensión de control permite una cierta tolerancia. En general, los módulos del PLC están protegidos contra sobretensiones, dependiendo del módulo a través del cual se realiza la unidad central de control.

### 2.7.10.3 TENSIÓN DE LA LÓGICA

---

Además, un PLC requiere una alimentación para la lógica interna: La tensión de la lógica, que forma las señales en la unidad central. Dado que la tensión de la lógica debe cumplir con exigencias muy estrictas, debe estar estabilizada. Se utilizan o bien 5 V (nivel TTL) o 10 V (nivel CMOS), dependiendo del módulo a través del cual se realiza la unidad central de control.

Hay tres posibilidades de tensión de alimentación:

- I. La tensión de control y la tensión de la lógica se generan completamente separadas de la tensión principal.
- II. Dos fuentes de alimentación combinadas en una caja para la generación de ambas tensiones.
- III. La tensión de la lógica es generada a partir de la tensión de control (no de la tensión principal).

### 2.7.10.4 SUPRESIÓN DE INTERFERENCIAS

---

Todos los PLC's son extremadamente sensibles a las interferencias eléctricas. Debe distinguirse entre dos versiones diferentes:

- **Interferencias que alcanzan la tensión de la lógica desde la alimentación de tensión a través de la fuente de alimentación.-** Un filtro principal de supresión de interferencias y un condensador protegen contra interferencias de este tipo. El filtro principal de supresión de interferencias protege contra sobretensiones y señales de interferencia de la alimentación de tensión. Un condensador almacena energía eléctrica, con lo que la tensión de alimentación del control está protegida en el caso de breves fallos de tensión. Si este tipo de supresión de interferencias no ha sido incorporado por el fabricante del PLC, deberá ser el usuario quien instale consecuentemente un filtro principal de supresión de interferencias y un condensador.
- **Interferencias que afectan a las líneas desde y hacia los sensores y actuadores.-** Los pulsos de interferencia en las líneas eléctricas pueden producir falsas señales 1 ó 0 en las entradas de los PLC's, que no corresponden a las emitidas por los sensores. Estas señales pueden generarse por efecto de otros cables. Este tipo de interferencias es peligrosa: Por ello, como norma, los módulos de entrada de un PLC están protegidos por medio de optoacopladores conectados en serie y por un retardo

de las señales. El optoacoplador protege contra tensiones de hasta 5000 V aproximadamente. El retardo de la señal evita señales espurias, ya que estas generalmente son muy breves. El retardo puede oscilar entre 1 y 20 ms, según el tipo de PLC. Los módulos de entrada de "Alta velocidad" (sin señal de retardo) deben apantallarse, por ejemplo por medio de cables blindados.

Los módulos de salida contienen también un optoacoplador para protección contra sobretensiones. Además, las salidas está protegidas ante cortocircuitos, aunque normalmente no ante cortocircuitos muy prolongados.

#### 2.7.10.5 TENSIÓN INDUCIDA

---

Cuando se corta la tensión que alimenta a una carga inductiva (p. ej. una bobina de una electroválvula), se crea un tensión inducida en la bobina.

Esta tensión inducida, que puede alcanzar valores muy altos, debe eliminarse para proteger el módulo de salida. Para ello se utiliza un diodo supresor. Los módulos de salida de muchos PLC's ya están provistos de diodos supresores de este tipo. Sin embargo, en este caso, la tensión residual se convierte en un factor de interferencia en los cables de interconexión. Por ello deben tomarse medidas protectoras directamente en el punto de origen, es decir, en la bobina: por medio de un diodo supresor (sólo para corriente continua) o por medio de un varistor (resistencia que depende de la tensión). También pueden utilizarse en paralelo con la bobina dos diodos Zener polarizados inversamente. Sin embargo, para tensiones que sobrepasen los 150 V, deben conectarse en serie varios diodos.

#### 2.7.10.6 PARO DE EMERGENCIA

---

Si se acciona el PARO DE EMERGENCIA, es esencial llegar a la situación que sea menos peligrosa, tanto para las personas como para el sistema. Los elementos finales de control y los actuadores que pudieran provocar situaciones peligrosas, deben desconectarse inmediatamente (p. ej. husillos). Por el contrario, los elementos de control finales y los actuadores que podrían provocar situaciones peligrosas a las personas o al sistema cuando se desactivan, deberán seguir activados incluso en un caso de emergencia (p. ej. dispositivos de sujeción). En un sistema automatizado, en cualquier momento debe disponerse de la posibilidad de accionar el PARO DE EMERGENCIA.

Esta es la razón por la que los controles electrónicos no deben asumir la función de PARO DE EMERGENCIA. El circuito de PARO DE EMERGENCIA debe ser establecido independientemente del PLC por medio de la técnica de relés (contactos físicos).

DIN 57 113 también lo especifica, ya que sería imposible activar un PARO DE EMERGENCIA con un control averiado o simplemente detenido. Una vez que el PARO DE EMERGENCIA ha sido desenclavado, no debe ser posible que la máquina se ponga en marcha automáticamente.

El circuito de PARO DE EMERGENCIA por Hardware realiza la verdadera función de seguridad. Adicionalmente, también puede enviarse al PLC una señal de PARO DE EMERGENCIA. Independientemente de la acción que se haya realizado por hardware, los programas del PLC deben reaccionar en consecuencia; en este caso, desactivando todas las salidas. Esto se define en un programa paralelo. Una vez que se ha repuesto el PARO DE EMERGENCIA, el sistema no debería poder arrancar de nuevo por sí solo. Debe accionarse un pulsador independiente para volver a poner en marcha el sistema.

---

### 2.7.11 LENGUAJES DE PROGRAMACIÓN PARA PLC SEGÚN IEC 61131-3

---

La norma IEC 61131-3 define cinco lenguajes de programación. Aunque la funcionalidad y estructura de estos lenguajes es muy diferente, son tratados como una sola familia de lenguajes por IEC 61131-3, con elementos de estructura solapados (declaración de variables, partes de organización tales como funciones y bloques de función, etc.) y elementos de configuración.

Los lenguajes pueden mezclarse de cualquier forma dentro de un proyecto de PLC. La unificación y estandarización de estos cinco lenguajes representa un compromiso de requerimientos históricos, regionales y específicos de cada sector. Se ha previsto la futura expansión (tal como el principio de bloque de función o el lenguaje de Texto Estructurado); además, se ha incorporado la información necesaria sobre detalles tecnológicos (tipo de datos, etc.).

#### 2.7.11.1 LENGUAJE DE DIAGRAMA DE CONTACTOS O ESCALERA (LADDER DIAGRAM) - LD

---

El Diagrama de contactos es un lenguaje de programación gráfico derivado de los esquemas de circuitos de los mandos por relés directamente cableados. El diagrama de

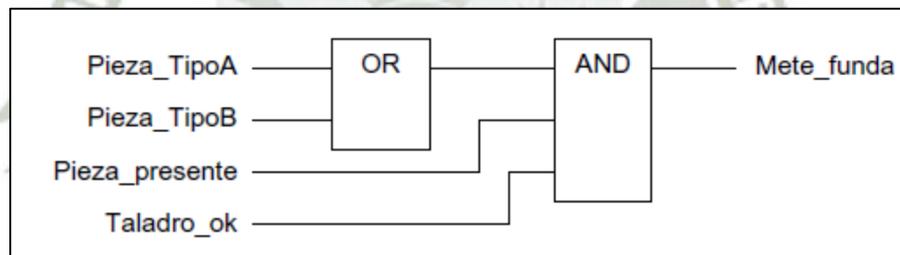
contactos contiene líneas de alimentación a derecha e izquierda del diagrama; a estas líneas están conectados los reglones, que se componen de contactos (normalmente abiertos y normalmente cerrados) y elementos de bobina, ver ejemplo en la **Figura 15**.



**Figura 15 : Ejemplo de lenguaje de LD**

### 2.7.11.2 LENGUAJE DE DIAGRAMA DE BLOQUES DE FUNCIÓN (FUNCTION BLOCK DIAGRAM) – FBD

En el diagrama de bloques de función, las funciones y los bloques de función están representados gráficamente e interconectados en redes. El diagrama de bloques de función tiene su origen en el diagrama lógico que se utiliza en el diseño de circuitos electrónicos, ejemplo en la **Figura 16**.



**Figura 16 : Ejemplo de lenguaje de FBD**

### 2.7.11.3 LENGUAJE DE LISTA DE INSTRUCCIONES (STATEMENT LIST O INSTRUCTION LIST) – IL

La lista de instrucciones es un lenguaje textual tipo Assembler, caracterizado por un modelo de máquina simple (procesador con un sólo registro). La lista de instrucciones se formula a partir de instrucciones de control consistentes en un operador y un operando, como un ejemplo tenemos la **Figura 17**.

LD	Pieza_TipoA
OR	Pieza_TipoB
AND	Pieza_presente
AND	Taladro_ok
ST	Mete_funda

**Figura 17 : Ejemplo de lenguaje de IL**

En lo que se refiere a filosofía de lenguaje, el diagrama de contactos, el diagrama de bloques de función y la lista de instrucciones han sido definidos en la forma en que son utilizados en la actual tecnología de PLC. Sin embargo, están limitados a las funciones básicas en lo que concierne a sus elementos. Esto los aparta esencialmente de los dialectos que las empresas utilizan en la actualidad. La competitividad de estos lenguajes es mantenida debido al uso de bloques y bloques de función.

#### 2.7.11.4 LENGUAJE TEXTO ESTRUCTURADO (STRUCTURE TEXT) – ST

El texto estructurado es un lenguaje de alto nivel basado en Pascal, que consiste en expresiones e instrucciones. Las instrucciones pueden definirse principalmente como: Instrucciones de selección, tales como IF...THEN...ELSE, etc., instrucciones de repetición tales como FOR, WHILE etc. y llamadas a bloques de función.

```
Mete_funda := (Pieza_TipoA OR Pieza_TipoB) AND Pieza_presente AND Taladro_OK;
```

**Figura 18 : Ejemplo de lenguaje de ST**

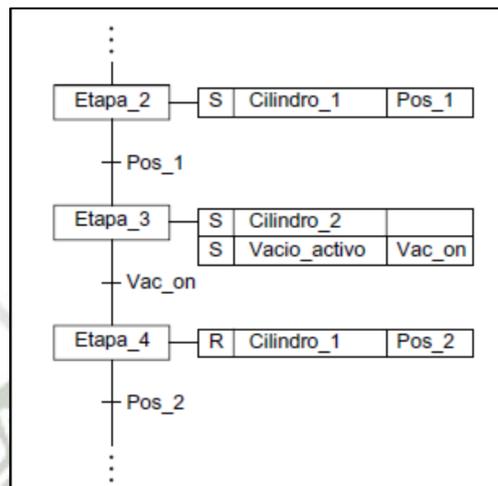
El texto estructurado mostrado en la **Figura 18**, permite la formulación de numerosas aplicaciones, más allá de la pura tecnología de funciones, tales como algoritmos (algoritmos de regulación de nivel superior, etc.) y manejo de datos (análisis de datos, procesamiento de estructuras de datos complejas, etc.)

#### 2.7.11.5 DIAGRAMA DE FUNCIONES SECUENCIALES (SEQUENTIAL FUNCTION CHART) – SFC

El diagrama de funciones secuencial (casi idéntico al GRAFCET francés) (ver **Figura 19**) es un recurso de lenguaje para la estructuración de los programas de control orientados a secuencias.

Los elementos del diagrama de funciones secuencial son las etapas, las transiciones y las derivaciones alternativas y en paralelo. Cada etapa representa un estado del proceso de un programa de control, que se halla activo o inactivo. Una etapa consiste

en acciones que, al igual que las transiciones, están formuladas en los lenguajes IEC 1131-3. Las propias acciones pueden contener de nuevo estructuras secuenciales. Esta característica permite la estructura jerárquica de un programa de control. Por lo tanto, el diagrama de funciones secuencial es una herramienta excelente para el diseño y la estructuración de programas de control.



**Figura 19 : Ejemplo de lenguaje de SF**

## 2.8 SISTEMAS SCADA

Los sistemas SCADA (Supervisory Control And Data Acquisition) se crean para alcanzar este objetivo, estos comprenden una serie de aplicaciones de software, en la que se toman datos provenientes de un determinado proceso y se evalúan para determinar si están dentro de los valores deseados o si es necesario realizar alguna acción correctiva de control.<sup>13</sup>

Se trata de una aplicación de software, especialmente diseñada para funcionar sobre ordenadores en el control de producción, proporcionando comunicación con los dispositivos de campo (controladores autónomos, autómatas programables, etc.) y controlando el proceso de forma automática desde una computadora. Además, envía la información generada en el proceso productivo a diversos usuarios, tanto del mismo nivel como hacia otros supervisores dentro de la empresa, es decir, que permite la

<sup>13</sup> [20] **Aquilino Rodríguez** (2007). "Sistemas SCADA". 2da Ed. Editorial Alfaomega. México. México.



participación de otras áreas como por ejemplo: control de calidad, supervisión, mantenimiento, etc.

Cada uno de los ítems de SCADA (Supervisión, Control y Adquisición de datos) involucran muchos subsistemas, por ejemplo, la adquisición de los datos puede estar a cargo de un PLC (Controlador Lógico Programable) el cual toma las señales y las envía a las estaciones remotas usando un protocolo determinado, otra forma podría ser que una computadora realice la adquisición vía un hardware especializado y luego esa información la transmita hacia un equipo de radio vía su puerto serial, y así existen muchas otras alternativas. Las tareas de Supervisión y Control generalmente están más relacionadas con el software SCADA, en él, el operador puede visualizar en la pantalla del computador de cada una de las estaciones remotas que conforman el sistema, los estados de ésta, las situaciones de alarma y tomar acciones físicas sobre algún equipo lejano, la comunicación se realiza mediante buses especiales o redes LAN. Todo esto se ejecuta normalmente en tiempo real, y están diseñados para dar al operador de planta la posibilidad de supervisar y controlar dichos procesos.

Estos sistemas actúan sobre los dispositivos instalados en la planta, como son los controladores, autómatas, sensores, actuadores, registradores, etc.

Además permiten controlar el proceso desde una estación remota, para ello el software brinda una interfaz gráfica que muestra el comportamiento del proceso en tiempo real.

Generalmente se vincula el software al uso de una computadora o de un PLC, la acción de control es realizada por los controladores de campo, pero la comunicación del sistema con el operador es necesariamente vía computadora. Sin embargo el operador puede gobernar el proceso en un momento dado si es necesario.

Un software SCADA debe ser capaz de ofrecer al sistema:

- Posibilidad de crear paneles de alarma, que exigen la presencia del operador para reconocer una parada o situación de alarma, con registro de incidencias.
- Generación de datos históricos de las señale de planta, que pueden ser volcados para su proceso sobre una hoja de cálculo.
- Ejecución de programas, que modifican la ley de control, o incluso anular o modificar las tareas asociadas al autómata, bajo ciertas condiciones.



- Posibilidad de programación numérica, que permite realizar cálculos aritméticos de elevada resolución sobre la CPU del ordenador.

Existen diversos tipos de sistemas SCADA dependiendo del fabricante y sobre todo de la finalidad con que se va a hacer uso del sistema, por ello antes de decidir cuál es el más adecuado hay que tener presente si cumple o no ciertos requisitos básicos:

- Todo sistema debe tener arquitectura abierta, es decir, debe permitir su crecimiento y expansión, así como deben poder adecuarse a las necesidades futuras del proceso y de la planta.
- La programación e instalación no debe presentar mayor dificultad, debe contar con interfaces gráficas que muestren un esquema básico y real del proceso
- Deben permitir la adquisición de datos de todo equipo, así como la comunicación a nivel interno y externo (redes locales y de gestión)
- Deben ser programas sencillos de instalar, sin excesivas exigencias de hardware, y fáciles de utilizar, con interfaces amigables para el usuario.

---

### 2.8.1 CARACTERÍSTICAS DE UN SISTEMA SCADA

---

Sistemas de control hay muchos y muy variados y todos, bien aplicados, ofrecen soluciones óptimas en entornos industriales. Lo que hace de los sistemas SCADA una herramienta diferencial es la característica de control supervisado.

En consecuencia, se supervisa el control de la planta y no solamente se monitorizan las variables que en un momento determinado están actuando sobre la planta; es decir, se puede actuar y variar las variables de control en tiempo real, algo que pocos sistemas permiten con la facilidad intuitiva que dan los sistemas SCADA.

Se puede definir la palabra supervisar como ejercer la inspección superior en determinados casos, ver con atención o cuidado y someter un objeto a un nuevo examen para corregirlo o repararlo permitiendo una acción sobre el objeto supervisado. La labor del supervisor representa una tarea delicada y esencial desde el punto de vista normativo y operativo; de esta acción depende en gran medida garantizar la calidad y eficiencia del proceso que se desarrolla. En el supervisor descansa la responsabilidad de orientar o corregir las acciones que se desarrollan. Por

lo tanto tenemos una toma de decisiones sobre las acciones de control por parte del supervisor, que en el caso de los sistemas SCADA recaen sobre el operario.

Esto diferencia notablemente los sistemas SCADA de los sistemas clásicos de automatización donde las variables de control están distribuidas sobre los controladores electrónicos de la planta y dificulta mucho una variación en el proceso de control, ya que estos sistemas una vez implementados no permiten un control a tiempo real óptimo. La función de monitorización de estos sistemas se realiza sobre un PC industrial ofreciendo una visión de los parámetros de control sobre la pantalla de ordenador lo que se denomina HMI -Human Machine Interface- (Se explicará más adelante en otro apartado), como en los sistemas SCADA, pero sólo ofrecen una función complementaria de monitorización: Observar mediante aparatos especiales el curso de uno o varios parámetros fisiológicos o de otra naturaleza para detectar posibles anomalías (Definición Real Academia de la Lengua).

En los sistemas SCADA, se utiliza un HMI interactivo el cual permite detectar alarmas y a través de la pantalla solucionar el problema mediante las acciones adecuadas en tiempo real. Esto otorga una gran flexibilidad a los sistemas SCADA. En definitiva, el modo supervisor del HMI de un sistema SCADA no solamente señala los problemas, sino lo más importante, orienta en los procedimientos para solucionarlos.

A menudo, las palabras SCADA y HMI inducen cierta confusión en los profanos (frecuentemente alentada por los mismos fabricantes en su afán de diferenciar el producto o exaltar comercialmente el mismo). Ciertamente es que todos los sistemas SCADA ofrecen una interfaz gráfica PC-Operario tipo HMI, pero no todos los sistemas de automatización que tienen HMI son SCADA. La diferencia radica en la función de supervisión que pueden realizar estos últimos a través del HMI.

---

## 2.8.2 FUNCIONES PRINCIPALES DEL SISTEMA SCADA

---

- Supervisión remota de instalaciones y equipos: Permite al operador conocer el estado de desempeño de las instalaciones y los equipos alojados en la planta, lo que permite dirigir las tareas de mantenimiento y estadística de fallas.
- Control remoto de instalaciones y equipos: Mediante el sistema se puede activar o desactivar los equipos remotamente (por ejemplo abrir Válvulas,

activar interruptores, prender motores, etc.), de manera automática y también manual.

Además es posible ajustar parámetros, valores de referencia, algoritmos de control, etc.

- **Procesamiento de datos:** El conjunto de datos adquiridos conforman la información que alimenta el sistema, esta información es procesada, analizada, y comparada con datos anteriores, y con datos de otros puntos de referencia, dando como resultado una información confiable y veraz.
- **Visualización gráfica dinámica:** El sistema es capaz de brindar imágenes en movimiento que representen el comportamiento del proceso, dándole al operador la impresión de estar presente dentro de una planta real. Estos gráficos también pueden corresponder a curvas de las señales analizadas en el tiempo.
- **Generación de reportes:** El sistema permite generar informes con datos estadísticos del proceso en un tiempo determinado por el operador.
- **Representación de señales de alarma:** A través de las señales de alarma se logra alertar al operador frente a una falla o la presencia de una condición perjudicial o fuera de lo aceptable. Estas señales pueden ser tanto visuales como sonoras.
- **Almacenamiento de información histórica:** Se cuenta con la opción de almacenar los datos adquiridos, esta información puede analizarse posteriormente, el tiempo de almacenamiento dependerá del operador o del autor del programa.
- **Programación de eventos:** Esta referido a la posibilidad de programar subprogramas que brinden automáticamente reportes, estadísticas, gráfica de curvas, activación de tareas automáticas, etc.

---

### 2.8.3 PANTALLAS HMI

---

Un HMI o un interfaz Hombre-Máquina, es un sistema que presenta datos a un operador y a través del cual este controla un proceso determinado.

Las HMI podemos definirlos como una ventana de un proceso. Donde esta ventana puede estar en dispositivos especiales como paneles de operador o en una PC.



Interacción Hombre-Máquina tiene como objeto de estudio el diseño. La evaluación y la implementación de sistemas interactivos de computación para el uso humano, así como los principales fenómenos que los rodean. Dado que este es un campo muy amplio han surgido áreas más especializadas, entre las cuales se encuentran Diseño de interacción o de interfaces de usuario, arquitectura de información y **Usabilidad**<sup>14</sup>.

El diseño de Interacción se refiere a la creación de la interfaz de usuario y de los proceso de interacción. La arquitectura de Información apunta a la organización y estructura de la información brindada mediante software. La Usabilidad se aboca al estudio de las interfaces y aplicaciones con el objeto de hacerlas fáciles de usar, fáciles de recordar, fáciles de aprender y eficientes con bajo coeficiente de error en su uso y que generan satisfacción en el usuario. A su vez, se asemeja a una disciplina ingenieril por que plantea objetivos mesurables y métodos rigurosos para alcanzarlos.

La industrial de HMI nace de la necesidad de estandarizar la manera de monitorear y de controlar múltiples sistemas remotos, PLCs y otros mecanismos de control, con la necesidad de tener un control más preciso y agudo de las variables de producción y de contar con información relevante de los distintos procesos en tiempo real. Aunque un PLC realiza automáticamente un control pre-programado sobre un proceso, normalmente se distribuye a lo largo de toda la planta, haciendo difícil recoger los datos de manera manual, los sistemas SCADA lo hacen de manera automática.

Desde fines de la década de los '90, la mayoría de productores de PLC ofrecen integración con sistemas HMI/SCADA. Y muchos de ellos utilizan protocolos de comunicaciones abiertos y no propietarios, que han permitido masificar este tipo de sistemas y ponerlos al alcance de pequeñas empresas.

---

#### 2.8.4 CARACTERÍSTICAS HMI

---

Hardware estándar para distintas aplicaciones: permite controlar varias aplicaciones según el requerimiento del operador.

---

<sup>14</sup> ISO/IEC 9126: "La *usabilidad* se refiere a la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso"

Posibilidad de modificaciones futuras para el proceso, mediante el software se puede modificar las condiciones de trabajo para la obtención del proceso deseado.

Posibilidades de ampliación; se puede reemplazar y añadir dispositivos de acuerdo al crecimiento del proceso en la industria.

Interconexión y cableado exterior: Es muy baja ya que sustituyen sistemas cableados (elementos físicos como botones, interruptores, equipos de relés, lámparas, led) por sistemas programables compactos.

Tiempo de implantación: es muy corto.

Mantenimiento: es más fácil ya que se lo realiza mediante el programa que fue previamente cargado en el proceso que está siendo objeto de control:

Configuración: permite definir el control de trabajo del SCADA, adaptándolo a la aplicación particular que se desea desarrollar. Interfaz gráfica del operador: proporciona al operador las funciones de control y supervisión de la planta.

Módulo de proceso: ejecuta las acciones de mando pre-programadas a partir de los valores actuales de variables leídas.

Comunicaciones: transferencia de información entre la planta y la arquitectura hardware que soporta el SCADA, y también entre esta y el resto de elementos informáticos de gestión.

---

### 2.8.5 FUNCIONES DE UN SOFTWARE HMI

---

El software permite entre otras funciones las siguientes: Interfaz gráfica de modo que se pueda ver el proceso e interactuar con él, registro en tiempo real e histórico de datos y manejo de alarmas.

Si bien es cierto que sólo la primera función enunciada es el propio HMI, casi todos los proveedores incluyen las otras dos ya sea en el mismo paquete o bien como opcionales. También es normal que dispongan de muchas más herramientas. Al igual que en los terminales de operador, se requiere de una herramienta de diseño o desarrollo, la cual se usa para configurar la aplicación deseada, y luego debe quedar corriendo en el PC un software de ejecución (Run Time). Por otro lado, este software puede comunicarse

directamente con los dispositivos externos (proceso) o bien hacerlo a través de un software especializado en la comunicación. Las funciones más destacables serían:

- **Monitorización.** Es la habilidad de obtener y mostrar datos de la planta en tiempo real. Estos datos se pueden mostrar como números, texto o gráficos que permitan una lectura más fácil de interpretar.
- **Supervisión.** Esta función permite, junto con la monitorización, la posibilidad de ajustar las condiciones de un trabajo del proceso directamente desde el computador.
- **Alarmas.** Es la capacidad de reconocer eventos excepcionales dentro del proceso y reportar estos eventos. Las alarmas son reportadas basadas en límites de control preestablecidos.

---

#### 2.8.6 ESTRUCTURA GENERAL DE UN SOFTWARE HMI

---

Los software HMI están compuestos por un conjunto de programas y archivos<sup>15</sup>. Hay programas para diseño y configuración del sistema y otros que son el motor mismo del sistema. En la **Figura 20** se muestra cómo funcionan algunos de los programas y archivos más importantes. Los rectángulos de la figura representan programas y las elipses representan archivos. Los programas que están con recuadro simple representan programas de diseño o configuración del sistema; los que tienen doble recuadro representan programas que son el motor del HMI.

Con los programas de diseño, como el “editor de pantallas” se crean moldes de pantallas para visualización de datos del proceso. Estos moldes son guardados en “Archivos de pantalla” y almacenan la forma de cómo serán visualizados los datos en las pantallas.

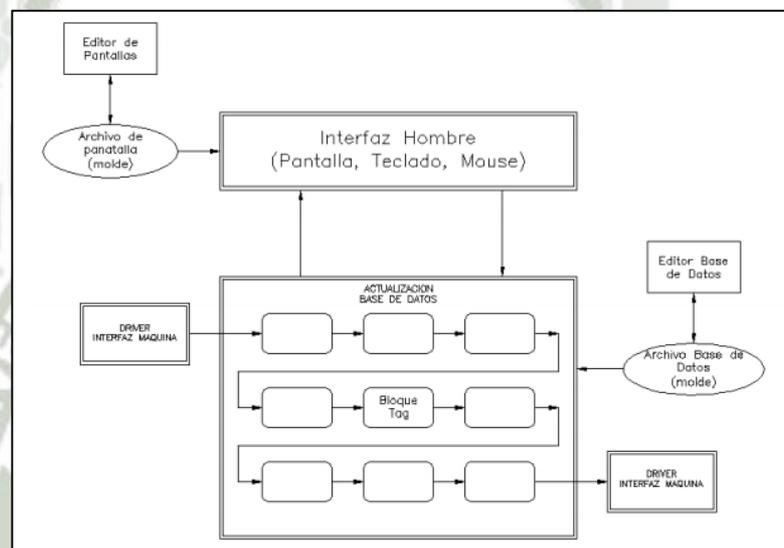
- **Interfaz Hombre-Máquina:** es un programa que se encarga de refrescar las variables de la base de datos en la pantalla, y actualizarla, si corresponde, por entradas del teclado o el mouse. Este programa realiza la interfaz entre la base

---

<sup>15</sup> [18] Introducción a HMI / Estructura General de Software HMI  
(<http://iaci.unq.edu.ar/materias/laboratorio2/HMI%5CIntroduccion%20HMI.pdf>)

de datos y el hombre. El diseño de esta interfaz está establecido en el archivo molde "Archivo de pantalla" que debe estar previamente creado.

- **Base de datos:** Es un lugar de memoria de la computadora donde se almacenan los datos requeridos del proceso. Estos datos varían en el tiempo según cambien los datos del proceso, por esta razón se denomina "base de datos dinámica". La base de datos está formada por bloques que pueden estar interconectados. La creación de la base de datos, sus bloques y la relación entre ellos se realiza a través de "editor de base de datos".



**Figura 20 : Estructura de un Sistema HMI**

- **Driver:** La conexión entre los bloques de la base de datos y las señales del proceso se realiza por medio de drivers. Estos drivers manejan los protocolos de comunicación entre el HMI y los distintos dispositivos de campo. Los drivers son entonces la interfaz hacia la máquina.
- **Bloques (tags):** La base de datos está compuesta por bloques como se ha mencionado. Para agregar o modificar las características de cada bloque se utiliza el editor de la base de datos. Los bloques pueden recibir información de los drivers u otros bloques y enviar información hacia los drivers u otros bloques. Las funciones principales de los bloques son:
  - Recibir datos de otros bloques o del driver.



- Enviar datos a otros bloques o al driver.
- Establecer enlaces (links) a la pantalla (visualización, teclado o mouse).
- Realizar cálculos de acuerdo a instrucciones del bloque.
- Comparar los valores con umbrales de alarmas.
- Escalar los datos del driver a unidades de ingeniería.

---

### 2.8.7 TENDENCIAS

---

Teniendo en cuenta las tendencias que se han ido llevando los últimos años en cuanto a la adquisición de datos en tiempo real y la supervisión y control de procesos, se podría ofrecer una evolución en los siguientes aspectos:

- Su integración completa en entornos para gestionar el negocio de forma que se disponga de información de la planta en tiempo real<sup>16</sup>, control y tratamientos de datos, y supervisión y gestión de la empresa. Un ejemplo sería la existencia de aplicaciones MES (Manufacturing Execution Systems), los servidores de datos y los servidores de web son una prueba de ello.
- Tratamiento de datos adquiridos en planta por sistema expertos para detección y diagnóstico de fallos. De esta forma se ayudaría al personal de planta a detectar los fallos, diagnosticar el problema que los provoca y conocer las acciones correctas a seguir.
- Mejora de las interfaces con el usuario en cuanto a gráficos de alta calidad, elementos multimedia de audio y vídeos, mejora de los sistemas operativos para incrementar los tiempos de respuesta, utilización de software orientado a objetos, diálogos conversacionales con programador y usuario, etc. Todo ello utilizando un hardware que lo soporte y que a su vez sea cada vez más compacto, fiable, potente, más rápido y que permita un mayor ancho de banda.

---

<sup>16</sup> [19] <http://es.scribd.com/doc/202955998/Desarrollo-de-Un-Sistema-HMI-Para-Un-Almacen-Automatizado>

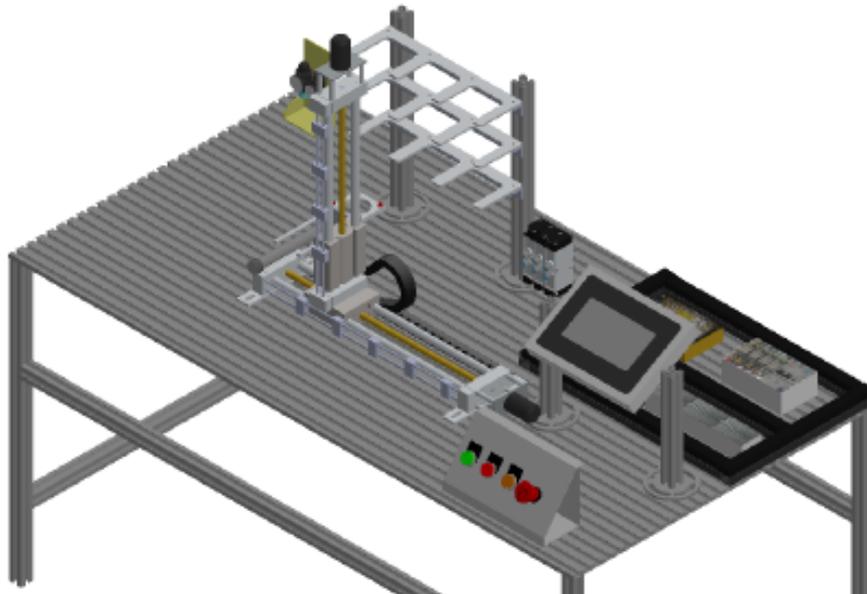


## CAPÍTULO 3: DESCRIPCIÓN DEL MÓDULO

En este capítulo describiremos el funcionamiento del módulo, la matriz de almacenamiento así como los diversos componentes tales como dispositivos eléctricos, neumáticos, de control y supervisión, detallando los criterios para la selección de determinados elementos.

### 3.1. DESCRIPCIÓN FUNCIONAL DEL MÓDULO

La tesis explica el módulo de almacenamiento vertical automatizado que se ha implementado, en la **Figura 21** observamos una vista general del módulo, el módulo controlado y supervisado por un HMI; sistema que se ha aplicado en almacenes automatizados en países industrializados durante muchos años.



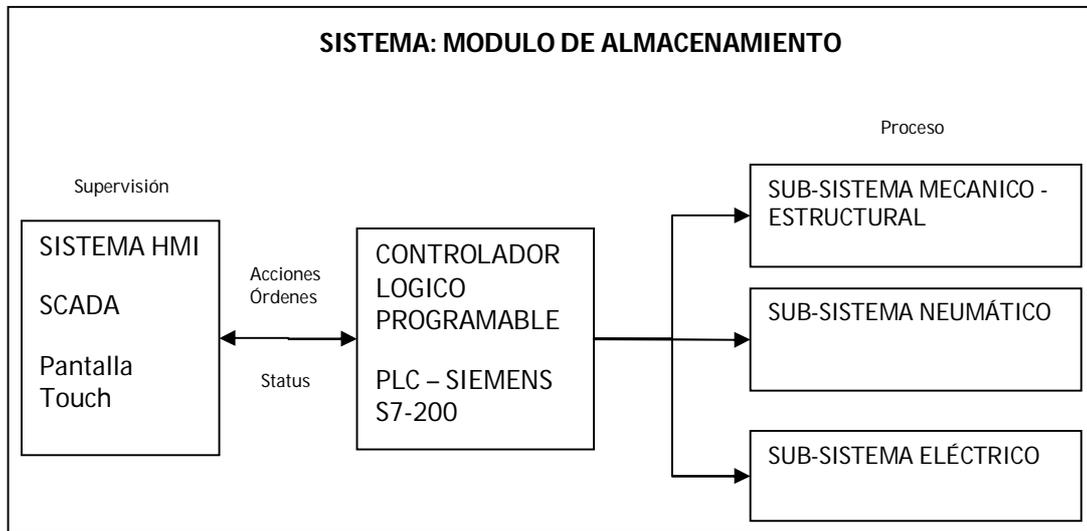
**Figura 21 : Vista general del módulo implementado**

*(Fuente propia)*

Para el módulo de tesis implementado se trabajó con la integración de diferentes disciplinas de la ingeniería como son: neumática, eléctrica y dispositivos de control y supervisión; todas estas combinadas sobre un soporte estructural para cumplir con la



implementación del módulo planteado, en la **Figura 22** se puede apreciar un diagrama de flujos para el desarrollo del módulo.

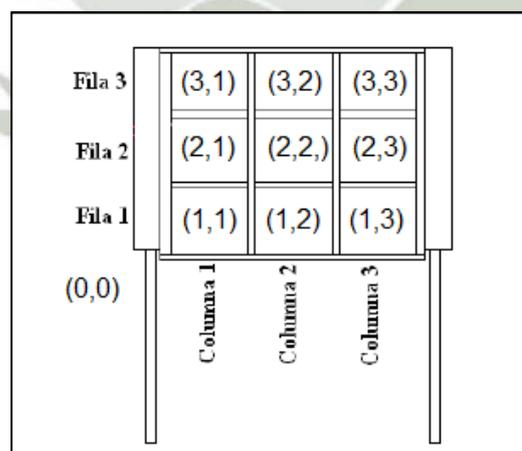


**Figura 22: Diagrama de flujo Módulo Implementado**

(Fuente Propia)

### 3.1.1. MATRIZ DE ALMACENAMIENTO:

La matriz de almacenamiento está compuesta de tres columnas y tres filas, lo que significa 9 posiciones disponibles para almacenar (ver Figura). Podemos suponer que cada posición representa un tipo de producto diferente por lo tanto se tiene la capacidad de almacenar nueve distintos productos.



**Figura 23: Matriz de Almacenamiento**

(Fuente Propia)



Las dimensiones de esta matriz son las que determinan el recorrido de trabajo de los dispositivos de desplazamiento, tanto en el eje vertical como en el horizontal.

---

### 3.1.2. MOVIMIENTO VERTICAL:

---

En el módulo implementado este movimiento representaría el primer grado de libertad, tiene la capacidad de ubicación de tres posiciones que representan las tres filas que componen la matriz de almacenamiento.

Se basó en las siguientes consideraciones para el diseño del movimiento vertical.

- Debe ser capaz de desplazarse de arriba hacia abajo recorriendo la distancia de la matriz de almacenamiento.
- Saber la posición exacta donde se desea llegar, lo cual se realizara con finales de carrera.
- El movimiento debe ser suave y sin vibraciones, por lo cual se utilizó como mecanismo de transmisión un tornillo sin fin y tuerca.
- Para soportar el peso de las partes que se requiere mover y el efecto de gravedad, debe constar con un buen mecanismo de sujeción.
- El punto inicial de este movimiento se encuentra por debajo de la Fila 1, en esta posición se colocó un final de carrera que permite limitar el desplazamiento hacia abajo.

---

### 3.1.3. MOVIMIENTO HORIZONTAL:

---

En el módulo implementado este movimiento representaría el segundo grado de libertad, tiene la capacidad de ubicación de tres posiciones que representan las tres columnas que componen la matriz de almacenamiento.

Se basó en las siguientes consideraciones para el diseño del movimiento horizontal.

- El robot debe desplazarse de izquierda a derecha abarcando el ancho de la matriz de almacenamiento.
- Su movimiento debe ser lineal, suave y sin vibraciones; por lo cual se utilizó como mecanismo de transmisión un tornillo sin fin y tuerca.



- Saber la posición exacta donde se desea llegar, lo cual se realizara con finales de carrera.
- Debe soportar el peso del transportador final y el carrito de desplazamiento vertical.
- El punto inicial de este movimiento se encuentra a la izquierda de la Columna 1, en esta posición se colocó un final de carrera que permite limitar el desplazamiento hacia abajo.

---

#### 3.1.4. MODOS DE OPERACION:

---

El programa del HMI tiene la opción de elegir dos tipos diferentes de procesos de almacenamiento, los cuales se explican a continuación:

---

##### 3.1.4.1. MODO MANUAL:

---

En la pantalla principal del HMI se seleccionara la opción "Modo Manual" previamente realizando un reseteo del sistema con el pulsador de Emergencia.

Para iniciar la función de trabajo se presionara el pulsador Start el cual activara la secuencia del sistema, seguido a eso seleccionamos en la pantalla "Llevar a Posición 0" para que el brazo actuador se ubique en el punto inicial del sistema.

En la pantalla del HMI se observan los indicadores de cada posición y seleccionando la posición donde deseamos alojar el producto, el brazo se dirigirá al punto y dejara el pallet en la posición indicada y en la pantalla se encenderá el led de la ubicación deseada.

Seleccionamos de nuevo "Llevar a Posición 0" para realizar un nuevo movimiento, después de que el brazo se encuentre en la posición inicial se escogerá otra posición donde guardar el pallet y así sucesivamente.



---

#### 3.1.4.1. MODO AUTOMÁTICO:

---

En la pantalla principal del HMI se seleccionara la opción “Modo Automático” previamente realizando un reseteo del sistema con el pulsador de Emergencia.

Para iniciar la función de trabajo se presionara el pulsador Start el cual activara la secuencia del sistema, en la pantalla del Modo Automático se visualizara los botones de LLEVAR A POSICION 0 y EJECUTAR SECUENCIA DE AUTOMATICO, así mismo los indicadores de cada posición.

Después seleccionamos en la pantalla “Llevar a Posición 0” para que el brazo actuador se ubique en el punto inicial del sistema y de ahí se escogerá “Secuencia de Modo Automático” donde el brazo se ubicara en la primera posición, en la pantalla se marcara que la posición ha sido llenada.

Después volverá a la “Posición 0” para seguidamente ubicarse en la segunda posición, en la pantalla se colocara que la posición ha sido llenada y volverá a la posición inicial para recoger el rack y dirigirse a la tercera posición y así sucesivamente hasta llenar todo las posiciones de la matriz de almacenamiento y en la pantalla figurara que todos los espacios han sido ocupados.

---

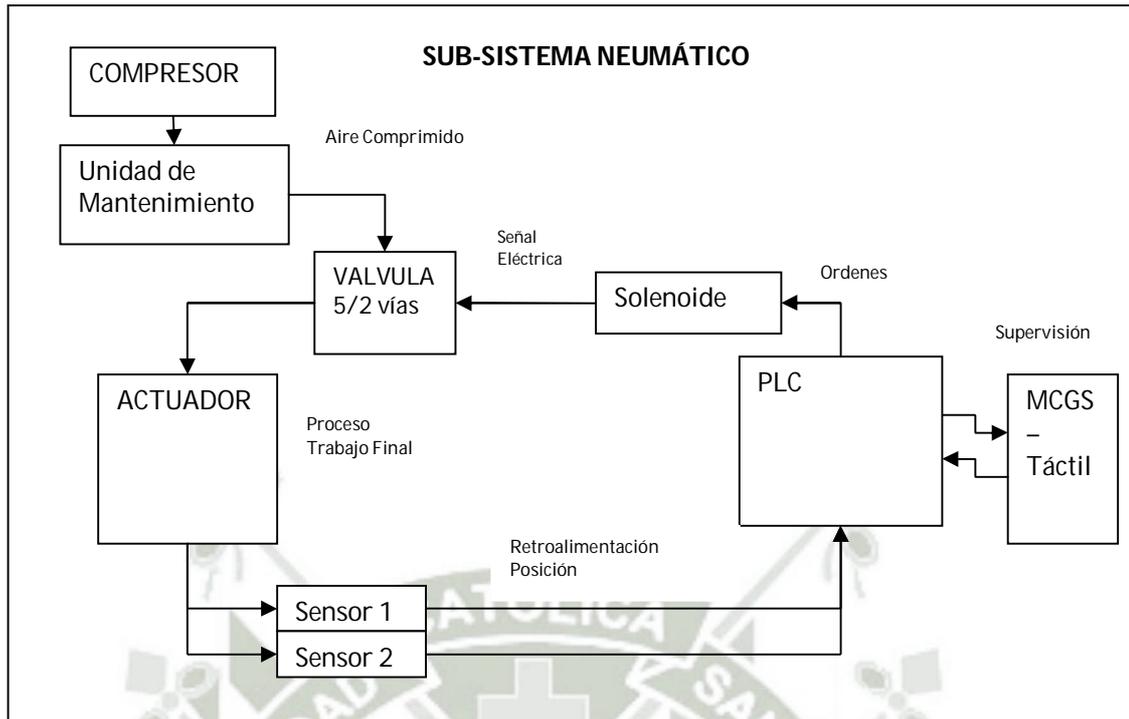
### 3.2. DESCRIPCIÓN DEL SISTEMA MECATRÓNICO POR SUBSISTEMAS

---

#### 3.2.1. SUBSISTEMA NEUMÁTICO

---

La unidad neumática se compone por el alimentador de aire (compresor), unidad de mantenimiento, electroválvulas y el cilindro neumático, todos estos combinados para que el actuador final ubique las piezas en su ubicación especificada cuando el PLC envíe la señal al solenoide de la electroválvula aperturando el pase de aire al actuador final, en la **Figura 24** se puede ver el diagrama de flujo para el sistema Neumático.



**Figura 24: Diagrama de Flujo Sistema Neumático**

(Fuente Propia)

### 3.2.1.1. COMPRESOR DE AIRE.

El compresor es una máquina eléctrica que sirve para elevar la presión del aire a una presión de trabajo determinada, se requirió para este proyecto un compresor que pueda abastecer el caudal y presión necesaria para accionar el pistón neumático.



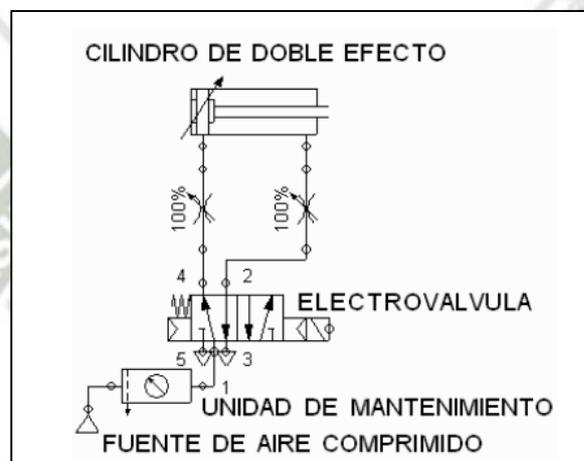
**Figura 25: Alimentador de aire comprimido**

Fuente: [www.aliexpress.com/product/550w-8l.html](http://www.aliexpress.com/product/550w-8l.html)

De acuerdo a los requerimientos seleccionamos un compresor móvil comercial con las siguientes características (ver Figura)

- Voltaje de Entrada: 220V
- Frecuencia: 60Hz
- Potencia: 600W
- Caudal: 102 L/min
- Presión Promedio: 70 Bar

Para la conexión de proveniente del compresor se siguió el siguiente diagrama neumático. (Ver **Figura 26**).



**Figura 26 : Diagrama de conexión neumática**

(Fuente: Propia)

Para la conexión neumática se usó manguera flexible y acoples rápidos facilitándonos su rápida instalación del sistema de alimentación de aire en la estación de almacenamiento. No se requiere herramienta alguna para la inserción o remoción de la manguera en la conexión, lo cual simplifica las tareas de ensamble, reparación o modificación en la instalación del módulo en proceso.

### 3.2.1.2. UNIDAD DE MANTENIMIENTO NEUMÁTICO (FRL):

El aire comprimido antes de ser utilizado en cualquier herramienta o equipo neumático deberá sufrir un acondicionamiento final, el cual se logra utilizando una unidad FRL (Filtro, Regulador, Lubricador) que logra una duración prolongada y funcionamiento regular de los componentes neumáticos.



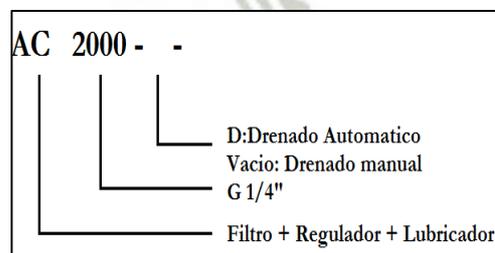
La unidad FRL, ilustrado en la **Figura 27**, tiene la función extraer del aire circulante todas las impurezas y el agua condensada en el último tramos antes de llegar a los puntos de utilización para evitar daños en el sistema; el regulador de presión se utiliza para mantener constante la presión de trabajo y compensar automáticamente el volumen de aire requerido por los equipos neumáticos y servir a la vez como válvula de seguridad, y el lubricador tiene la misión de lubricar los elementos neumáticos en medida suficiente así previene el desgaste prematuro de piezas móviles, reduce el rozamiento y protege los elementos contra la corrosión.



**Figura 27 : Filtro regulador de aire**

Fuente: [spanish.alibaba.com/goods/oil-filter-regulator.html](http://spanish.alibaba.com/goods/oil-filter-regulator.html)

Para el proyecto se utilizó un FRL de modelo AC 2000 de marca AIR TAC, el código se describe en la **Figura 28**:



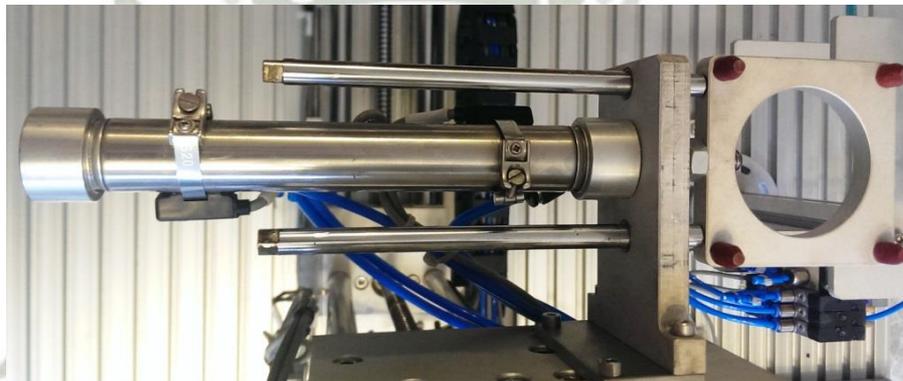
**Figura 28 : Designación por código de Filtro regulador de aire**



### 3.2.1.3. PISTÓN DE ACCIONAMIENTO:

En esta parte se hace uso de un cilindro neumático el cual se desplaza por medio de la unidad de posicionamiento para primero ubicarse en la posición inicial recoger el pallet con el producto a almacenar y después ser transportado a la posición de almacenamiento especificada, extenderse y dejar el pallet.

Este cilindro (ver **Figura 29**), tiene como herramienta un marco cuadrado de 12 x 12 cm en el cual descansaran los pallets de almacenaje, anclado al marco a ambos lados del eje del cilindro neumático lleva dos guías paralelas los cuales sirven para darle estabilidad al marco y a los pallets sostenidos, ambos ejes y el cilindros están empotrados en una placa de Aluminio que pertenece al carrito de desplazamiento vertical el cual posiciona el cilindro al nivel correspondiente.

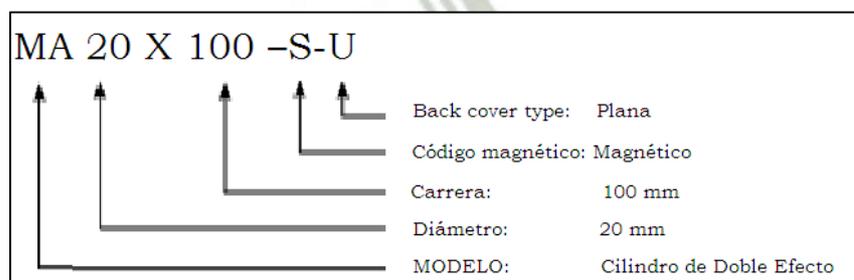


**Figura 29 : Cilindro Neumático**

(Fuente: Propia)

#### Análisis ingenieril del actuador neumático:

Se dispone de un actuador marca AIR TAC cuyo código se describe en la **Figura 30**:



**Figura 30 : Designación de Cilindro Neumático**



Las características son diámetro de embolo de 20 mm, carrera de 100 mm. Con detección de posición. Para nuestro caso se tiene como cargas las siguientes:

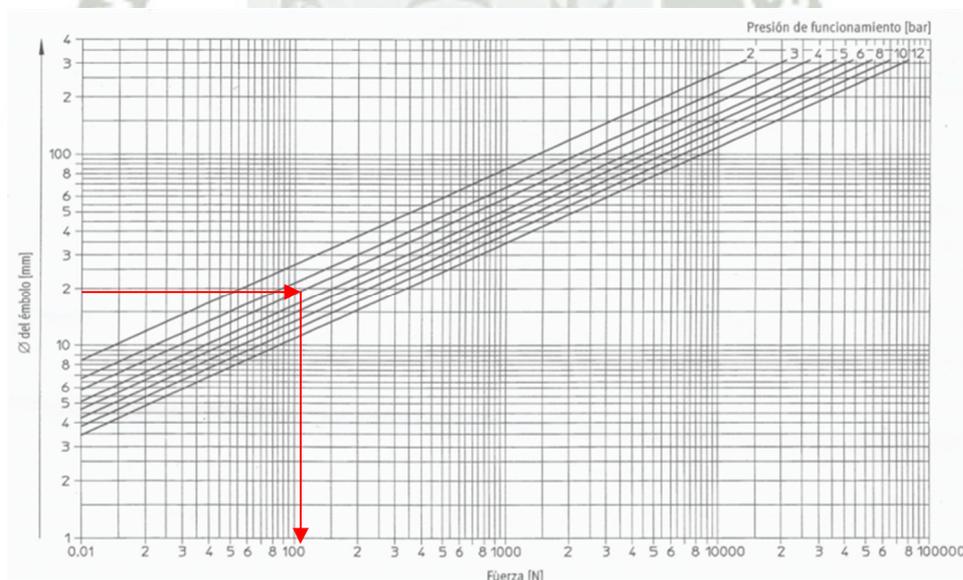
- Carga Horizontal: 01 rack (220 gr. Aprox.) de baquelita.
- Carga Vertical: El peso de cualquier material a almacenar (1200 gr. Aprox.) como un peso aproximado a escala (Caso Crítico)
- Marco de Aluminio (230 gr. Aprox)

Todas estas fuerzas concentradas en el extremo del vástago del cilindro siendo un total de 1650 gr. (llevando a unidades internacionales es 16.17 N).

Otros parámetros para la selección del actuador son las siguientes, relacionadas a la geometría de trabajo y energía de sistema:

- Presión de trabajo: 4 Bar (58 psi)
- Regulación de velocidad secundaria (expulsión y contracción)

Con estos datos primero haremos un análisis de Presión fuerza con el siguiente monograma (ver **Figura 31**):



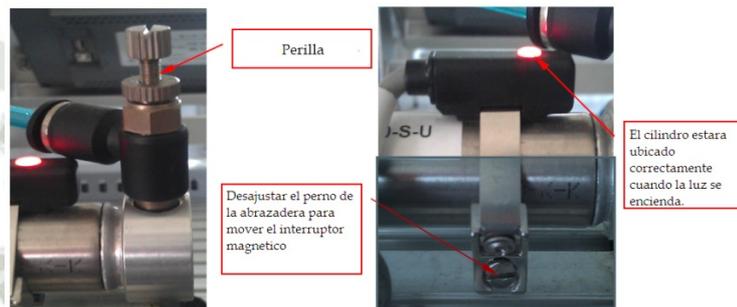
**Figura 31 : Monograma Fuerza- Presión**

Fuente: [http://www.guillesime.galeon.com/index\\_archivos/Page736.htm](http://www.guillesime.galeon.com/index_archivos/Page736.htm)

Trazamos una línea imaginaria con el diámetro del embolo disponible (20 mm.), hasta chocar con la recta de la presión de trabajo, para nuestro caso en 4 bares, en esta

intersección trazamos una línea vertical hacia abajo en el eje horizontal en la que se ubica la fuerza generada por ese embolo a la presión seleccionada. De este análisis vemos que incide en 110 N aproximadamente, este resultado es satisfactorio para nuestros requerimientos.

La velocidad de desplazamiento del cilindro neumático puede ser regulada ajustando el pase de aire del gobernador de velocidad simplemente regulando el caudal de alimentación con la perilla mostrada en la **Figura 32**.



**Figura 32 : Regulación del cilindro**

Si el vástago del cilindro no alcanza su distancia total o no está en su lugar correspondiente se debe al interruptor magnético de los dos finales del cilindro. Regular el interruptor magnético correspondiente a la posición correcta, las luces indicadores del sensor de carrera nos dirán si están en la posición correcta este extendido o retraído totalmente.

#### 3.2.1.4. ELECTROVÁLVULA:

El módulo posee energía neumática como eléctrica con la presencia de actuadores neumáticos y electroválvulas, estos son los elementos de maniobra de los actuadores neumáticos, que son utilizados para el accionamiento del pistón.

Una electroválvula es una válvula electromecánica, diseñada para controlar el flujo de un fluido a través de un conducto como puede ser una tubería. La válvula está controlada por una corriente eléctrica a través de una bobina solenoide.

Las electroválvulas resultan del acoplamiento de un sistema electromecánico (solenoide – electroimán de accionamiento) a una válvula de distribución neumática elemental con vertiéndola a una de accionamiento eléctrico, en **Figura 33** se muestra la forma física de la Electroválvula seleccionada.



El circuito consta de un actuador neumático de doble efecto para el cual existe una electroválvula la cual se encuentra asociadas en un manifold que proporciona mediante la vía de presión (A) y de dos vías de escape (R y S) los flujos de trabajo de las electroválvulas, la presión de trabajo máxima del módulo está en 8 bar (114).

Las electroválvulas usadas son de la marca FESTO, 5/2 monoestable, eléctrico con conector tipo zócalo, reseteo manual y código siguiente:

**CPE10-M1BH-5L-M5**



**Figura 33 : Electro-válvulas**

*Fuente Propia*

---

### 3.2.2. SUBSISTEMA ELÉCTRICO - ELECTRÓNICO

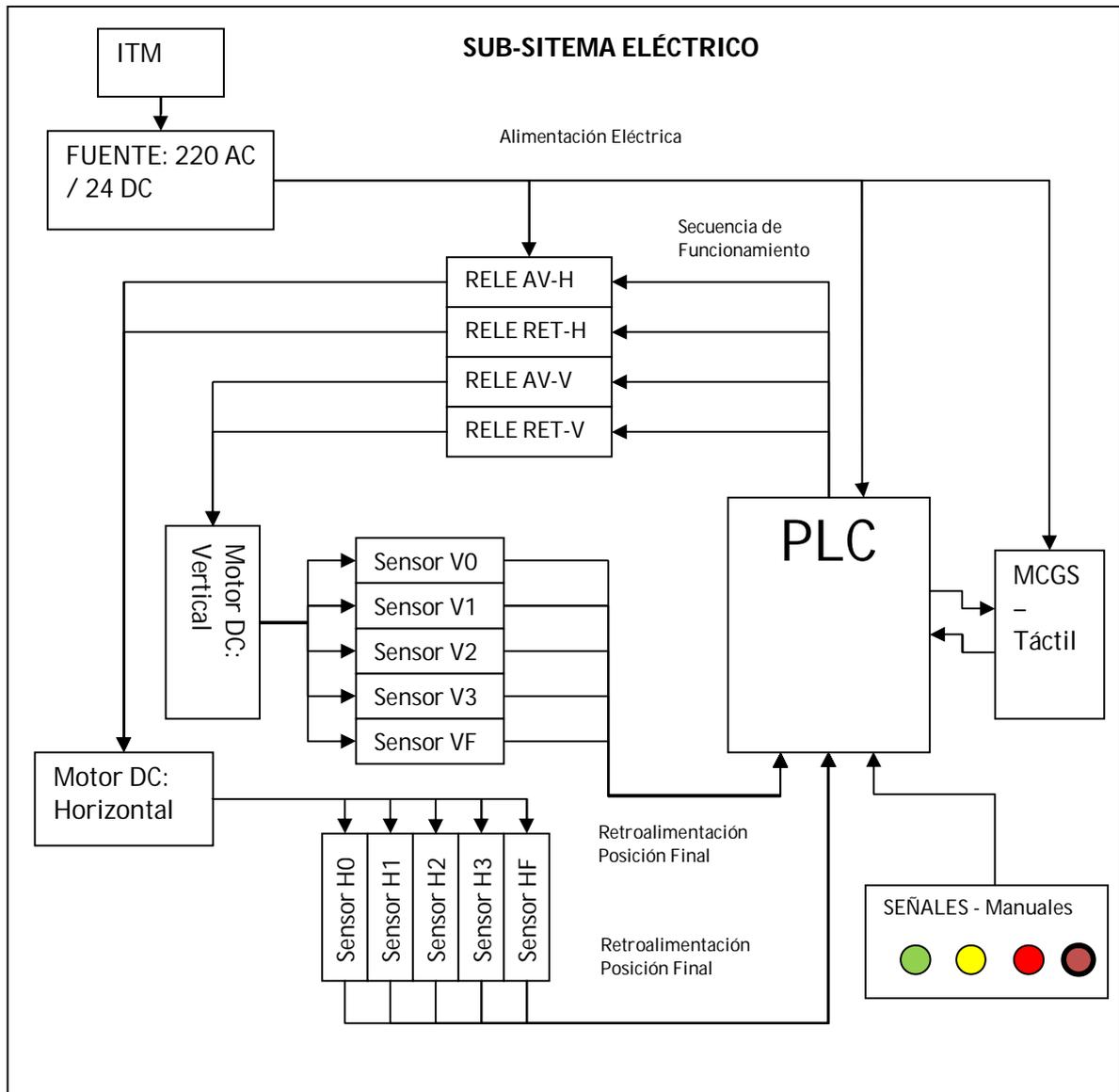
---

El sistema eléctrico tiene como unidad principal el tablero de control eléctrico, es ahí donde encuentra el controlador electrónico y las conexiones hacia sus periféricos, el tablero de control está diseñado de tal forma que pueda haber un correcto funcionamiento.

Todos los periféricos tanto de entrada tales como los elementos de mando y los sensores de posición, así también como los dispositivos de salida ya sean los relés que harán que se accionen los motores DC cuando el PLC envíe la señal de accionamiento, son conectados hacia el tablero de control eléctrico, el cual posee también las fuentes de alimentaciones necesarias para el trabajo de los actuadores y del PLC, , en la **Figura**



34 se puede ver el diagrama de flujo para la interacción de los componentes del sistema Eléctrico para el desarrollo del módulo.



**Figura 34 : Diagrama de flujo Sistema Eléctrico**

*Fuente Propia*

#### 3.2.4.1. MOTOR DC:

El movimiento giratorio para el almacenamiento de las piezas se realizó con dos motores DC los cuales realizan el movimiento en los ejes X-Z que hacen girar el tornillo sin fin de transmisión para el desplazamiento del pistón alimentador, cuyas características del Motor DC (ver **Figura 35**) son las siguientes:



- Tensión: 24V corriente continua
- Revoluciones a la salida del eje: 150 RPM (24V)
- Potencia: 800W
- Amperaje: 130 mA



**Figura 35 : Motor DC**

#### 3.2.4.2. FUENTE DE ALIMENTACIÓN:

El módulo de control necesita de fuentes de corriente continua para que el controlador y los demás componentes eléctricos operen así como los distintos periféricos conectados a este, por lo tanto se utilizó las siguientes fuentes de alimentación para los consumos mostrados en la

**Tabla 5:**

Cantidades activas	Descripción	Voltaje (Vdc)	Intensidad (mA)	Subtotal (mA)
1	Motor DC Eje X	24	130.00	130.00
1	Motor DC Eje Z	24	130.00	130.00
1	Electroválvulas	24	55.00	55.00
4	Relés	24	20.00	80.00
6	Sensor de selección inductivo o capacitivo	24	25.00	150.00
6	Final de carrera	24	5.00	30.00
1	PLC	24	100.00	100.00



1	HMI	24	150.00	150.00
	<b>TOTAL</b>		<b>615.00</b>	<b>825.00</b>

**Tabla 5 : Consumo eléctrico de fuente de alimentación**

### 3.2.4.3. DISPOSITIVOS DE PROTECCIÓN

#### A) DISYUNTOR TERMO-MAGNÉTICO Y DIFERENCIAL.

El módulo tiene un circuito de protección (disyuntor termo magnético y diferencial) para proteger los circuitos de control, auxiliares y a las personas contra las sobrecargas, cortocircuitos y con el fin de proteger a las personas de las derivaciones causadas por faltas de aislamiento entre los conductores activos y tierra o masa de los aparatos.

Los interruptores termo magnéticos (térmicas) se utilizan, en primer término, para proteger contra sobrecargas (cuando las corrientes son superiores a las permitidas por la instalación, pero no llegan al nivel de intervención del dispositivo magnético ya que el consumo va aumentando conforme se van conectando aparatos produciendo la sobrecarga es ahí donde actúa el circuito térmico abriendo el contacto) y a los cortocircuitos ( cuando la intensidad  $I$  que circula por la carga sobrepasa el límite de intervención fijado. cortocircuitos a los cables y conductores eléctricos se abre el contacto cortando el flujo).

El interruptor diferencial exponencial, también llamado disyuntor por corriente diferencial o residual, es un dispositivo electromecánico que se coloca en las instalaciones eléctricas con el fin de proteger a las personas de las derivaciones causadas por faltas de aislamiento entre los conductores activos y tierra o masa de los aparatos, este se activa cuando la intensidad de corriente de entrada es diferente a la de salida abriendo el contacto y cortando el flujo.

Este dispositivo cumple los siguientes requerimientos, para el correcto funcionamiento en el módulo, según su descripción técnica indicada en la **Tabla 6**:

- a) Bipolar, con retorno automático a posición abierta
- b) Capacidad de corriente: 10 Amp – 230 Vac.
- c) Corriente de fuga : 30 mA



d) Tiempo de Respuesta : <0.1 s

e) Capacidad de ruptura: 630 A

**Main Technical Parameter**

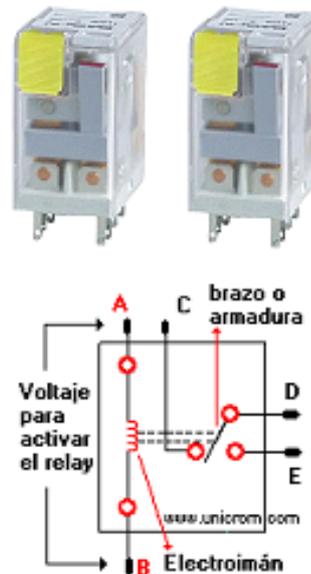
Rated current A	P	With neutral line	In A	V	I $\Delta$ n mA	I $\Delta$ no mA	I $\Delta$ n breaking time S	I $\Delta$ m A	overload instantameons release
32	1	N		230	30	15	<0.1	630	C D
	2								
	2	N	6 10 16						
	3		20 25 32						
	3	N							
	4								
60	1	N		230	300	150			C D
	2								
	2	N	40 50 60						
	3								
	3	N							
	4								

**Tabla 6 : Datos técnicos del Disyuntor Termo-magnético**

**Fuente: Data sheet-Anexo 2**

**B) RELÉS:**

Para la ejecución de acciones comandadas por el PLC, este envía señales pero estas señales no están directamente conectadas a los actuadores ya que en la mayoría de los casos estos consumen potencias o valores de tensión e intensidad muy por encima de la emitida por el módulo de salida del PLC, es por eso que para controlar las corrientes o tensiones de potencia que utilizan los actuadores se utiliza como medio los relés, los cuales reciben señales del PLC y permiten el paso de las corrientes o tensiones mientras el PLC siga emitiendo señales, activándose o desactivándose cuando el electroimán (que forma parte del Relé) es energizado (emisión de voltaje del controlador para que funcione). En la **Figura 36** observamos su forma física y su diagrama eléctrico.



**Figura 36 : Relés usados en el módulo y diagrama esquemático de funcionamiento**

Fuente: <http://todoreparacionelectronica.es.tl/TODO-ELECTR%D3NICA.htm>

Las características de los relés utilizados según la **Tabla 7:**

VOLTAJE NOMINAL VDC	VOLTAJE ACTIVACION	VOLTAJE LIBERACION	RESISTENCIA BOBINA $\Omega$ : $\pm 10\%$
24	19.2	2.4	640

**Tabla 7 : Características de Relé**

Los relés usados en el módulo tienen los datos técnicos mostrados en la **Tabla 8:**

<b>MARCA</b>	DAQCN
<b>TIPO</b>	57.02
<b>CANTIDAD DE CONECTATOS</b>	4 NA/NC COMUN
<b>RESISTENCIA DE CONTACTO</b>	50m $\Omega$ (1A 6VDC)
<b>CAPACIDAD DE CONTACTO</b>	10A/220VAC 30VDC
<b>TIEMPO DE OPERACIÓN</b>	25ms /25ms
<b>POTENCIA NOMINAL BOBINA</b>	0.9W/1.2VA

**Tabla 8 : Datos técnicos de los relés del módulo**



#### 3.2.4.4. TECNOLOGÍA DEL CONTROLADOR PLC SIEMENS S7-200

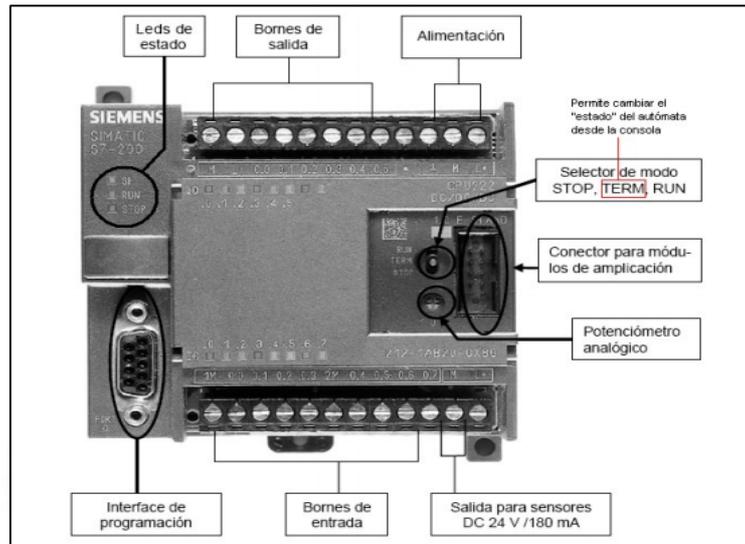
El PLC seleccionado y puesto en funcionamiento es de marca Siemens modelo S7 200, con el modelo de CPU 224.

Las características para su selección se describen en la **Tabla 9** a continuación:

FUNCION	CPU 224
Tamaño físico	120,5 mm x 80 mm x 62 mm
Memoria	
Programa	4096 palabras
Datos de usuario	2560 palabras
Tipo de memoria	EEPROM
Respaldo (condensador alto rendimiento)	190 horas (tip.)
E/S Integradas	
E/S Integradas	14 DI / 10DO
Numero de módulos de ampliación	7 módulos
Operaciones	
Relés internos	256
Contadores/temporizadores	256
Potenciómetros analógicos	2
Comunicación	
Numero de puertos de comunicación	PC/PPI ( RS-232)

**Tabla 9 : Datos técnicos del PLC.**

La configuración del PLC S7-200 está formada por una CPU propiamente dicha, una fuente de alimentación, entradas/salidas digitales y un conector de expansión que permite ampliar la CPU con la adición de módulos de expansión, se señalan sus partes en la **Figura 37**.

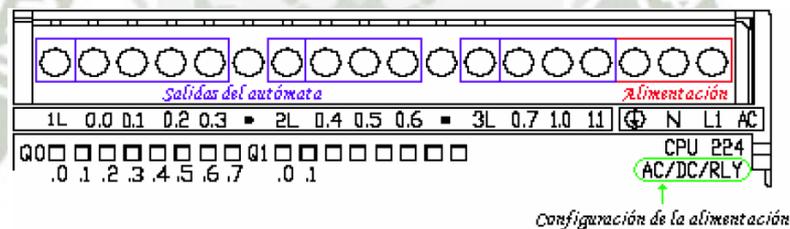


**Figura 37 : Distribución de componentes del PLC**

Fuente: [fa.jonweb.net/Siemens/images/Article/Technical/automatas.pdf](http://fa.jonweb.net/Siemens/images/Article/Technical/automatas.pdf)

La estructura física del PLC es la siguiente:

**A) TAPA SUPERIOR**

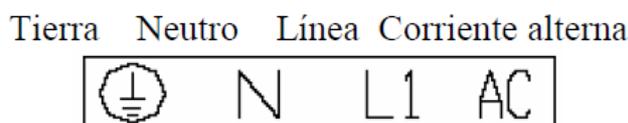


**Figura 38 : Configuración Tapa Superior**

Fuente: [isa.uniovi.es/docencia/ra\\_marina/cuatrim2/Temas/s7200](http://isa.uniovi.es/docencia/ra_marina/cuatrim2/Temas/s7200)

**• Puente de alimentación:**

Después de realizar la alimentación de tensión externa, proporciona los niveles de tensión necesarios para el correcto funcionamiento de los distintos circuitos electrónicos del automatismo (en este caso la CPU únicamente) se representan mediante la **Figura 39**.



**Figura 39 : Configuración Puente de Alimentación**



El CPU alimenta a través del bus interno a la memoria y las interfaces.

- **Características de la alimentación del autómeta:**

Se observa en la tapa superior:

**AC/DC/RLY** Esto significa que el autómeta se alimenta con una tensión alterna AC, posee una salida de continua DC y las salidas tienen conexión de relé o contacto libre de potencial RLY.

- **SALIDAS:**

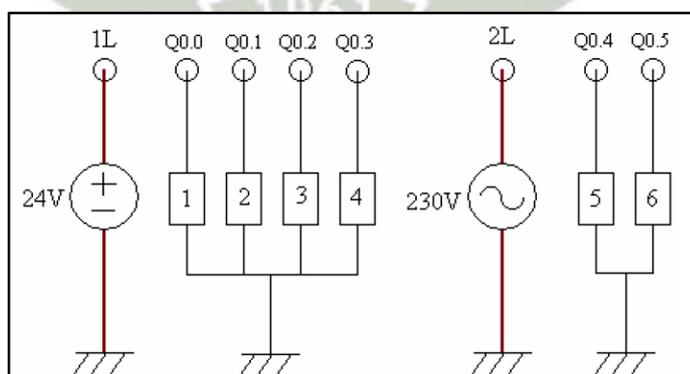
Las salidas son conectadas por relé en el caso del CPU 224 (contacto libre de potencial, ver **Figura 40**). Debido a esto, la tensión con la que debemos alimentar los comunes (1L, 2L, 3L) de las salidas debe coincidir exactamente con la tensión nominal de la carga que se encuentre conectada a la salida.

Esta tensión puede ser:

- 24V de corriente continua
- De 24 V a 230V de corriente alterna

Puesto que normalmente disponemos de varias cargas que requieren distintos niveles de tensión, deberemos conectar todas aquellas cargas que precisen la misma tensión a las salidas pertenecientes a un mismo común, y alimentar dicho común con la tensión nominal que necesiten dichas cargas.

Las cargas 1 - 4 necesitan una tensión de 24V de continua, mientras que las cargas 5 y 6, requieren 230V de alterna.



**Figura 40 : Tensión de las cargas**

Fuente: [isa.uniovi.es/docencia/ra\\_marina/cuatrim2/Temas/s7200](http://isa.uniovi.es/docencia/ra_marina/cuatrim2/Temas/s7200)

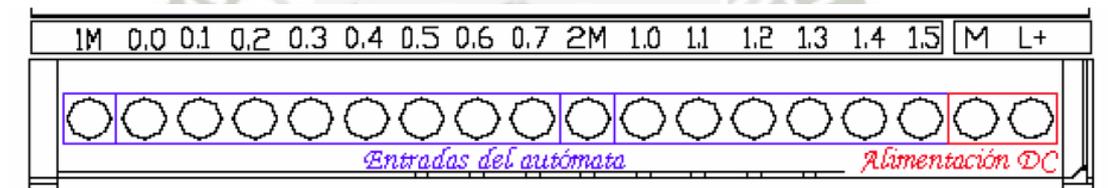


Las salidas configuradas en el PLC para la implementación del módulo son las que describen a en la **Tabla 10** a continuación:

ITEM	SIMBOLO	DIRECCION	COMENTARIO
1	LUZ_PAUSE	Q2.0	LUZ DE PULSADOR DE PAUSE
2		Q1.1	
3	LUZ_RESET	Q1.0	LUZ DE PULSADOR DE REFERENCIA
4	LUZ_PULINI	Q0.7	LUZ DE PULSADOR DE INICIO
5		Q0.6	
6	MOTORIZ_ATARAS	Q0.3	RELE PARA MOVIMIENTO DE EJE Y - ATRÁS
7	MOTORIZ_DELANTE	Q0.2	RELE PARA MOVIMIENTO DE EJE Y - ADELANTE
8	MOTORX_ATRAS	Q0.1	RELE PARA MOVIMIENTO DE EJE X - ATRÁS
9	MOTORX_DELANTE	Q0.0	RELE PARA MOVIMIENTO DE EJE X - ADELANTE

**Tabla 10 : Variables de Salida**

#### B) TAPA INFERIOR



**Figura 41 : Configuración Tapa Inferior**

Fuente: [isa.uniovi.es/docencia/ra\\_marina/cuatrim2/Temas/s7200](http://isa.uniovi.es/docencia/ra_marina/cuatrim2/Temas/s7200)

- **Alimentación DC:**

Para el PLC S7-200 Siemens utilizado, existe una salida de tensión de 24V de continua que se puede utilizar para alimentar las entradas del autómata.

- **Entradas:**

Las características de las entradas son las siguientes:

- Necesitan una tensión de entrada de 0V ó 24V de corriente continua para activarse.

Tienen una separación galvánica vía optoacoplador. De esta forma, si a la entrada llega un pico de tensión, la circuitería interna de la CPU-224 permanece intacta.

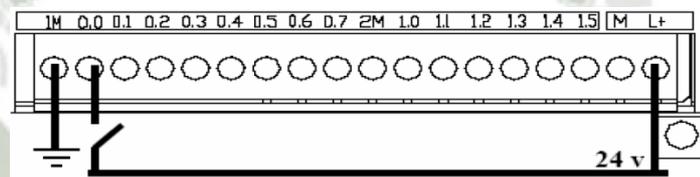


Para activar las entradas se deben hacer dos cosas:

Conectar a las entradas comunes 1M, 2M una tensión de 0V o de 24V de continua.

Dependiendo de qué tensión hayamos aplicado a los comunes, tendremos que introducir a las entradas I0.0, I0.1, etc., 0V ó 24V para provocar una diferencia de tensión y activarlas.

- Si queremos que las entradas se activen al aplicar 24V, debemos introducir 0V al común como se muestra en la **Figura 42**:



**Figura 42 : Alimentación de las entradas**

Fuente: [isa.uniovi.es/docencia/ra\\_marina/cuatrim2/Temas/s7200](http://isa.uniovi.es/docencia/ra_marina/cuatrim2/Temas/s7200)

Las entradas configuradas en el PLC para la implementación del módulo son las que describen en la **Tabla 11** a continuación:

ITEM	SIMBOLO	DIRECCION	COMENTARIO
4	EJEZ_FIN	I1.5	SENSOR EJEY POSICIÓN DE INICIO SECUENCIA
5	EJEZ_POS3	I1.4	SENSOR EJEY POSICION 3
6	EJEZ_POS2	I1.3	SENSOR EJEY POSICION 2
7	EJEZ_POS1	I1.2	SENSOR EJEY POSICION 1
8	EJEZ_REF	I1.1	SENSOR EJEY POSICION DE REFERENCIA
9	EJEX_FIN	I1.0	SENSOR EJEX POSICION DE INICIO SECUENCIA
10	EJEX_POS3	I0.7	SENSOR EJEX POSICION 3
11	EJEX_POS2	I0.6	SENSOR EJEX POSICION 2
12	EJEX_POS1	I0.5	SENSOR EJEX POSICION 1
13	EJEX_REF	I0.4	SENSOR EJEX POSICION DE REFERENCIA
14	PUL_REFERENCIA	I0.3	PULSADOR DE REFERENCIA
15	PAUSE	I0.2	PULSADOR DE PAUSE
16	INICIO	I0.1	PULSADOR DE INICIO
17	EMERGENCIA	I0.0	PULSADOR DE EMERGENCIA

**Tabla 11 : Variables de entrada**



### C) TAPA FRONTAL

En la tapa frontal (ver **Figura 43**) tenemos una pestaña con tres posiciones que nos permite situar al autómata en tres modos de funcionamiento distinto:



**Figura 43: Configuración Tapa Frontal**

Fuente: [isa.uniovi.es/docencia/ra\\_marina/cuatrim2/Temas/s7200](http://isa.uniovi.es/docencia/ra_marina/cuatrim2/Temas/s7200)

- **RUN:** El autómata ejecuta cíclicamente las instrucciones del programa de usuario.
- **TERM:** Este estado permite el control del autómata desde un terminal externo como, por ejemplo, un PC. Desde este terminal se puede poner el autómata en modo RUN o STOP.
- **STOP:** El autómata está encendido, pero el programa de usuario no se ejecuta.

Debajo de esta tapa también se ubican dos potenciómetros analógicos y la conexión a módulos de ampliación.

Los potenciómetros permiten incrementar o decrementar valores almacenados en los bytes de marcas especiales SMB28 y SMB29. Estos valores están comprendidos en el rango [0..255]. El programa puede utilizar estos valores de sólo lectura para diversas funciones, p.ej. para actualizar el valor actual de un temporizador o de un contador.

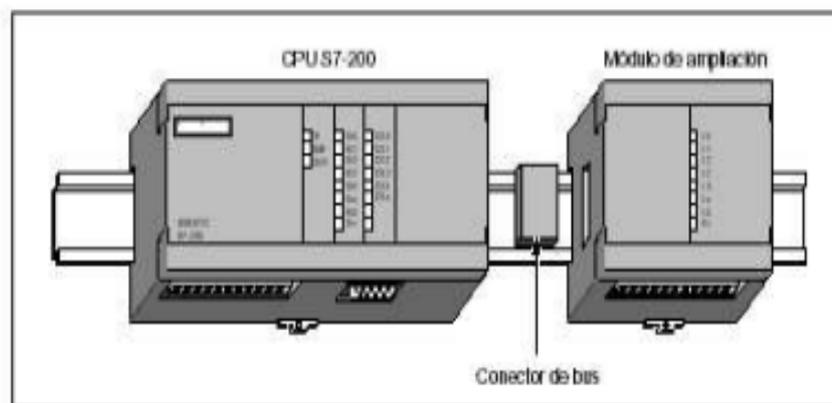
#### LEDs de estado

En la CPU existen una serie de LEDs que proporcionan información acerca el modo de funcionamiento de la CPU (RUN, STOP o TERMINAL), del estado de las entradas y salidas locales y de un fallo en el sistema.



#### D) MÓDULOS DE AMPLIACIÓN

Para la implementación del módulo de tesis las entradas y salidas disponibles por el PLC son suficientes, pero se analizó la probabilidad de implementación del proyecto donde se tendrían que agregar nuevas entradas y salidas es por eso que se decidió utilizar un módulo de ampliación (ver **Figura 44**). Para el proyecto ejecutado se utilizó un módulo de ampliación SIMATIC S7-200 Módulo EM 223, estos módulos vienen en digitales, analógicos e inteligentes que para nuestro proyecto se utilizó un digital con una combinación de 8 entradas DC y 8 salidas de relé, según la **Tabla 12**.



**Figura 44 : Módulo de Ampliación**

Fuente: [www.swe.siemens.com/spain/web/es/industry/automatizacion/simatic/Documents/S7200Manual](http://www.swe.siemens.com/spain/web/es/industry/automatizacion/simatic/Documents/S7200Manual)

Módulos de ampliación	Tipo de datos			
<b>Módulos digitales</b>				
Entrada	Entrada 8 entradas DC	8 entradas AC	16 entradas DC	
Salida	Salida 4 salidas DC	4 salidas de relé	8 salidas de relé	
	8 salidas DC	8 salidas AC		
Combinación	4 entradas DC / 4 salidas DC	8 entradas DC / 8 salidas DC	16 entradas DC / 16 salidas DC	32 entradas DC / 32 salidas DC
	4 entradas DC / 4 salidas de relé	8 entradas DC / 8 salidas de relé	16 entradas DC / 16 salidas de relé	32 entradas DC / 32 salidas de relé

**Tabla 12 : Configuración de Módulos de ampliación**



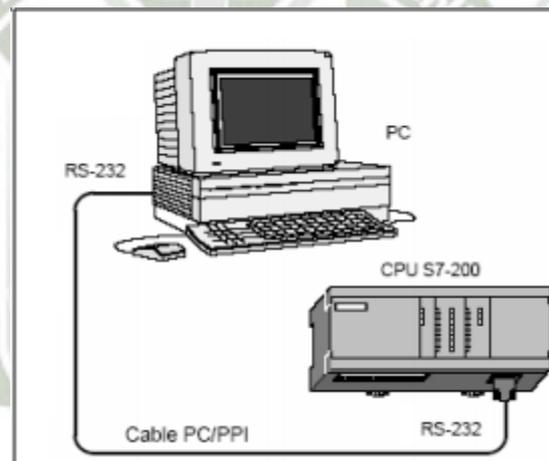
### E) COMUNICACIÓN DEL PC A LA CPU S7-224

En el proyecto se utilizó un cable PC/PPI (Point to Point Interface) para la conexión entre la PC y el PLC, este cable se conecta al puerto serie del PC, y convierte la señal RS-232 en el protocolo IPP (Interfaz Punto a Punto), con el programa o red de la CPU S7-200.

- **Configuración típica para conectar el PC a la CPU mediante el cable PC/PPI**

Para establecer un enlace correcto entre los componentes según la **Figura 45**, se requieren los siguientes pasos:

- 1.- Conectar el extremo RS-232 ("PC") del cable PC/PPI al puerto de comunicación de su PC (COM1 o COM2) y apretar los tornillos de conexión.
- 2.- Conectar el otro extremo (RS-232) del cable PC/PPI al interface de comunicación de la CPU y apretar los tornillos de conexión.



**Figura 45 : Conexión PC – CPU 224**

Fuente: [www.swe.siemens.com/spain/web/es/industry/automatizacion/simatic/Documents/S7200Manual](http://www.swe.siemens.com/spain/web/es/industry/automatizacion/simatic/Documents/S7200Manual)

#### 3.2.4.5. HMI – INTERFAZ HUMAN MACHINE

El HMI o "Human Machine Interface" es el dispositivo o sistema que permite la comunicación entre la persona y la máquina. Tradicionalmente estos sistemas consistían en paneles compuestos por indicadores y comandos, tales como luces pilotos, indicadores digitales y análogos, registradores, pulsadores, selectores y otros que se interconectaban con la máquina o proceso. En la actualidad, dado que las



máquinas y procesos en general están implementadas con controladores y otros dispositivos electrónicos que dejan disponibles puertas de comunicación, es posible contar con sistemas de HMI bastantes más poderosos y eficaces, además de permitir una conexión más sencilla y económica con el proceso o máquinas a través de una interfaz amigable.



**Figura 46 : HMI MCGS modelo TPC 7062**

El HMI utilizado en este proyecto es el MCGS modelo TPC 7062 (ver **Figura 46**) y tiene las características indicadas en la

**Tabla 13:**

Tamaño de pantalla	7 inch
Resolución	800*480
Brillo	200cd/m <sup>2</sup>
Panel de Control	Resistance type touch screen
Memoria	64M/FLASH+64M SDRAM
CPU	ARM CPU,400MHZ
Tensión de trabajo	DC24V (±15%)
Temperatura de trabajo	0-45 degree
Dimensiones	226.5*163(mm)

**Tabla 13 : Características HMI**

### 3.2.4.6. DISPOSITIVOS ELÉCTRICOS (PERIFÉRICOS)

#### A) ELEMENTOS DE MANDO

Como dispositivos de entrada del PLC, existen controles tales como botones y selectores, existe un panel de mando principal con el cual se emite información para que el proceso pueda ejecutarse, el módulo posee los siguientes controles (ver **Figura 47**) con sus características eléctricas correspondientes:

- Botón de **Start** (inicio de proceso) – botón normalmente abierto (NA verde)
- Botón de Reinicio (**Reset**) - botón normalmente abierto (NA verde)
- Botón de paro (Stop) – botón normalmente cerrado (NC rojo)
- Botón de Emergencia - (**Emergency**) – botón normalmente cerrado (NC rojo tipo hongo)



**Figura 47 : Dispositivos de mando**

#### B) SENSORES

- **Sensor Magnético**

El módulo posee también sensores magnéticos, estos sensores son usados para detectar algún campo magnético orientado hacia la proximidad de mecanismos o elementos, estos elementos usualmente poseen un imán, que genera su propio campo magnético adherido mecánicamente de alguna forma.

En el módulo hay dos lugares específicos para el uso de estos sensores:

Pistones del ALMACENADOR, internamente los cilindros usados poseen una cinta magnética adherida a su embolo, para detectar la posición de inicio (pistón contraído) colocados en la parte inferior del carrito, de esta manera se detecta las posiciones adecuadas (derecha e izquierda) límites de su desplazamiento.

- **Finales de carrera :** (ver *Figura 48*)



**Figura 48 :** Sensor magnético para la detección no invasiva del cilindro neumático

- **Sensor de contacto o finales de carrera:**

El sensor de contacto (también conocido como "interruptor de límite") o limit switch, son dispositivos eléctricos, neumáticos o mecánicos situados al final del recorrido de un elemento móvil en nuestro caso sirve para detectar la posición del carrito despachador y ubicarse en el nivel correcto de la unidad de almacenaje, en la *Figura 49* observamos su forma Física.



**Figura 49 :** Sensor de Final de Carrera

Fuente: [sensoresdeproximidad.blogspot.com/p/sensor-fin-de-carrera-el-final-de.html](http://sensoresdeproximidad.blogspot.com/p/sensor-fin-de-carrera-el-final-de.html)

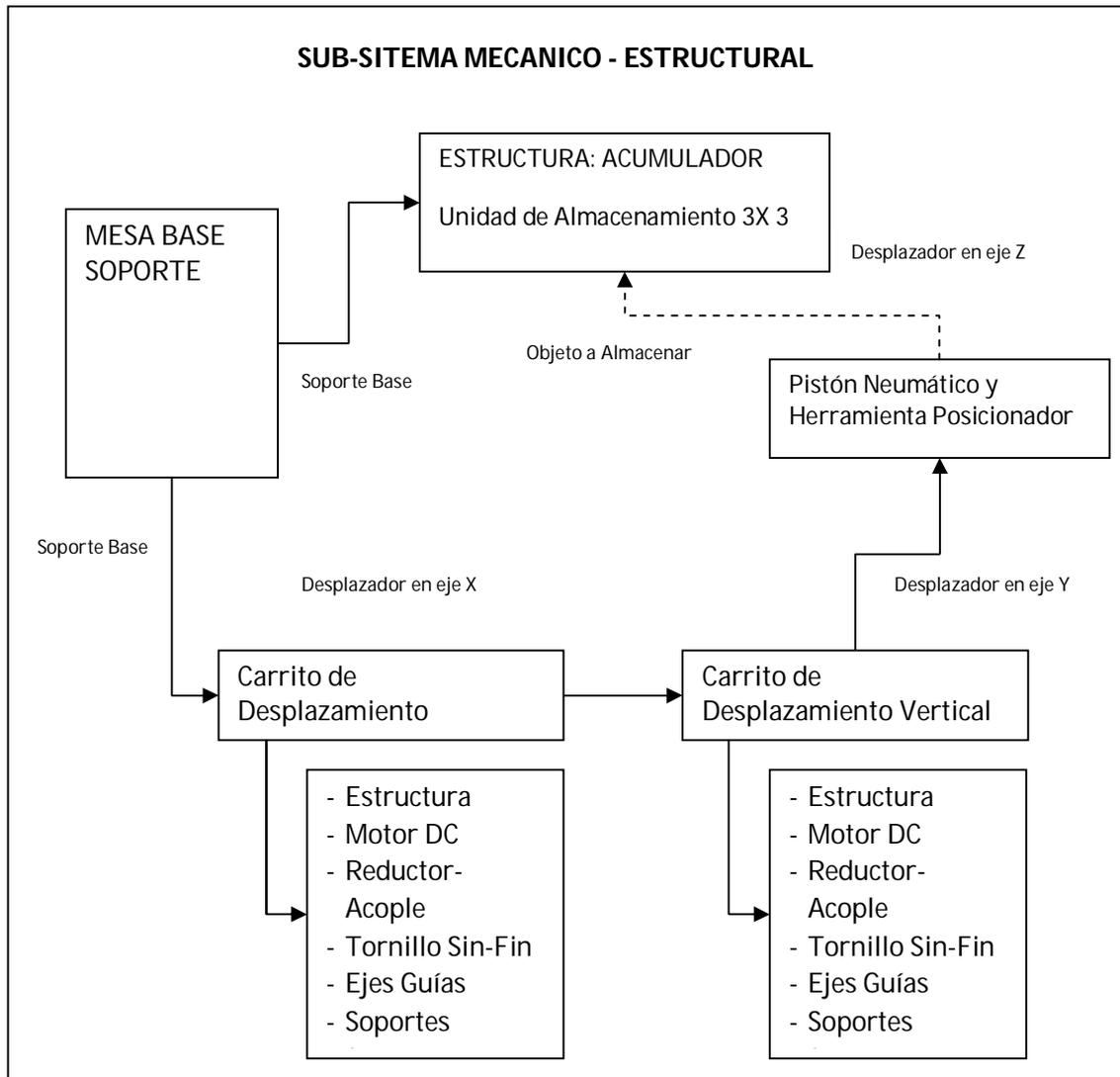
---

### 3.2.3 SUBSISTEMA ESTRUCTURAL – TRANSMISIÓN

---

Para el sistema estructural se implementó una mesa de trabajo de material aluminio para que sirva de soporte para los demás elementos de todos los sistemas utilizados, se implementó dos carritos desplazadores los cuales son los encargados de ubicar el pistón neumático en su posición correspondiente en los ejes "X" y "Y", una vez ubicado la se procede a dejar el objeto a almacenar en la Unidad de Almacenamiento así completando la tarea encomendada.

Se puede apreciar el diagrama de flujo del sistema estructural en la Figura.



**Figura 50 : Diagrama de flujo Sistema Estructural**

*Fuente: Propia*

### 3.3.6.1. UNIDAD DE ALMACENAMIENTO:

El módulo cuenta con una unidad de almacenamiento de aluminio de 3 columnas por 3 filas (ver **Figura 51**), que forman parte de la matriz de almacenamiento donde se almacenaran los pallet con los objetos a almacenar.

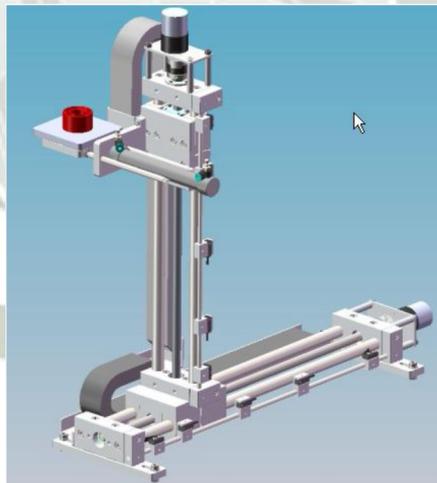


**Figura 51 : Unidad de Almacenamiento**

Ver Plano: UCSM-CIMAV-02-01@03 Acumulador (Anexo 04)

### 3.3.6.2. UNIDAD DE DESPLAZAMIENTO

El módulo consta con una unidad de desplazamiento que sujeta el cilindro neumático, (ver **Figura 52**), este carrito despachador se desplaza en dos ejes Vertical y Horizontal los cuales son accionados por el Motor DC de 24 que genera el movimiento.



**Figura 52 : Unidad de desplazamiento**

Ver planos UCSM-CIMAV-03/04/05/06 (Anexo 04)

### 3.3.6.3. CORREDERA HORIZONTAL

Sobre esta corredera se desplaza el bloque que viene unido al carrito de desplazamiento vertical, mostrado en la **Figura 53**, este bloque se desplaza en un eje



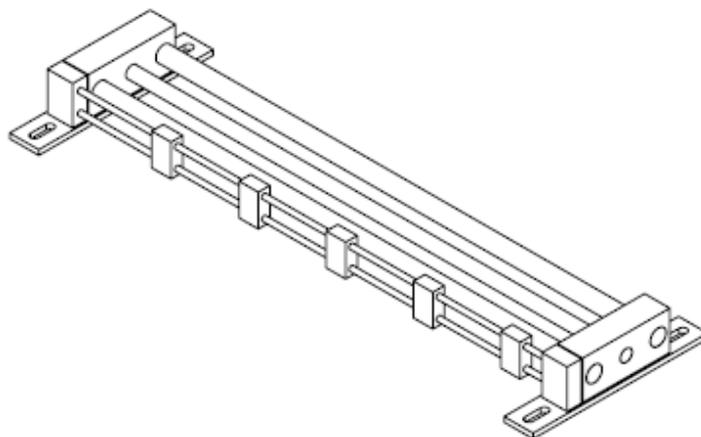
horizontal accionado por un motor de DC 24V, este motor está acoplado a un eje roscado.

- o Material: Acero aleado AISI 1020.
- o Tipo de Rosca: Trapezoidal con 10 Hilos/plg.
- o Diámetro externo: 10 mm.

Para poder obtener un desplazamiento lineal de la estructura de aluminio del carrito (aproximadamente 50 mm/seg), se adicionan dos tuercas de bronce (5) perforadas, las cuales son unidas mecánicamente mediante tornillos a la base de este, cuando gira el eje roscado, proporciona un deslizamiento interno por medio de sus roscas, ya que unida al carrito se da restricción en una dirección con respecto al eje.

Este eje roscado es alineado y soportado mediante rodamientos a los extremos, esto permite un adecuado movimiento rotacional, el eje roscado es un árbol, el cual transmite un torque. Las fuerzas que involucran acción hacia el rodamiento son esencialmente radiales, para ello se utilizara un pequeño rodamiento de bolas.

Así mismo el carrito horizontal posee dos guías paralelas al eje roscado, para poder darle estabilidad y deslizamiento adecuado en la dirección del movimiento, estos ejes son barras de acero pulido, estos ejes se deslizan mediante bujes de bronce al plomo. Tanto el eje roscado como los ejes guías están asegurados para evitar que puedan salirse de su posición de ensamble mediante seguros.



**Figura 53 : Eje roscado y ejes guía del carrito feeding**

**Ver Plano UCSM-CIMAV-04-01@02 Corredora Horizontal (Anexo 04)**

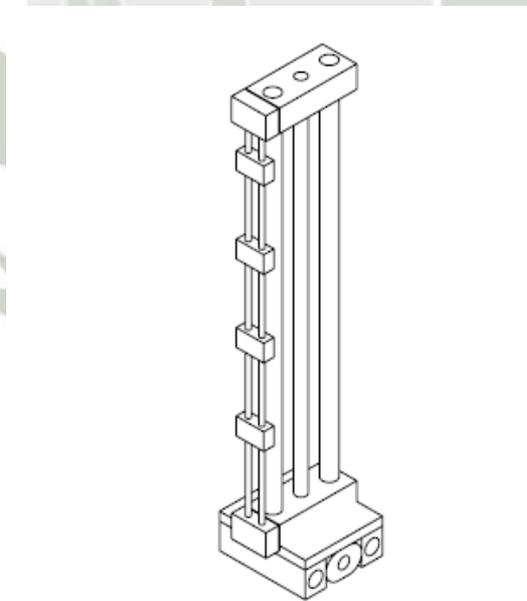


#### 3.3.6.4. CARRITO DE DESPLAZAMIENTO VERTICAL

Este carrito de desplazamiento vertical mostrado en la **Figura 54**, está unido a la corredera de desplazamiento horizontal por medio de un bloque de aluminio manufacturado el cual lleva tuercas que se deslizarán a través del tornillo sin fin. A medida que el carrito de desplazamiento horizontal se traslada para ubicarse en la columna ordenada el carrito vertical se desplaza en el eje para ubicarse a la altura de la fila deseada donde descargará el cilindro neumático su carga, este movimiento es accionado por un motor de 24 V DC que se une mediante un acoplamiento rígido de bronce hacia un eje roscado con las siguientes características:

- o Material: Acero aleado AISI 1020.
- o Tipo de Rosca: Trapezoidal con 10 Hilos/plg. (Ver Tabla 3.1 para más detalles de perfil de rosca trapezoidal).
- o Diámetro externo: 10 mm.

Las consideración del tornillo sin fin y rosca, así como los ejes de alineación utilizados para su movimiento vertical son los mismo utilizados en la descripción para el desplazamiento horizontal, la única diferencia que tiene es que el bloque que se desplazara en forma vertical está unido al cilindro actuador.



**Figura 54 : Eje roscado y ejes guían del carrito feeding**

**Ver Plano UCSM-CIMAV-06-01@02 Corredera Vertical (Anexo 04)**

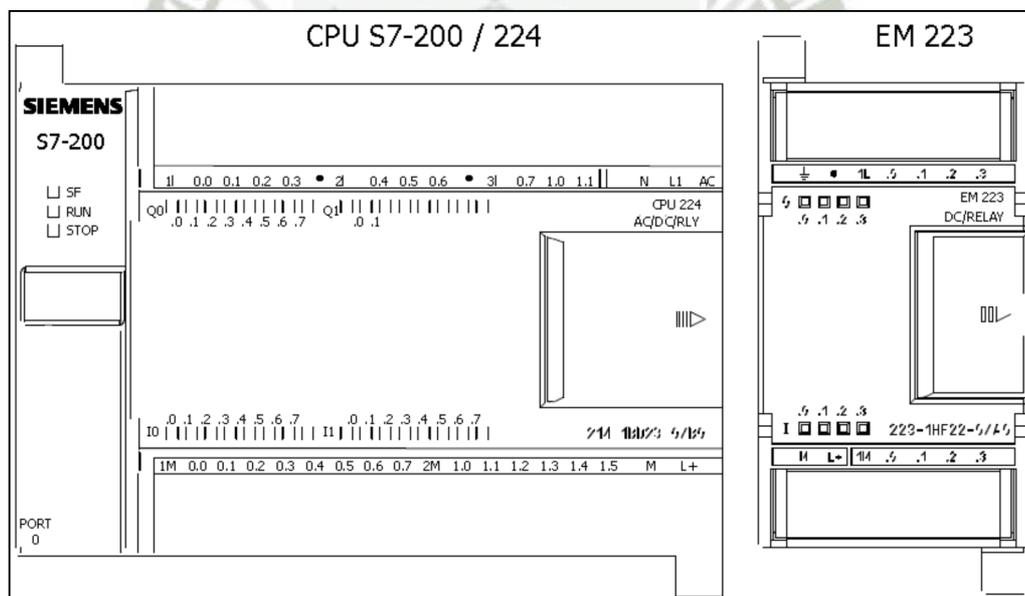
## CAPÍTULO 4: CONFIGURACIÓN Y PROGRAMACIÓN

En este capítulo se muestran las principales configuraciones que deben realizarse a fin de implementar la programación del controlador y el diseño de la interfaz hombre máquina, tanto en el PLC como en la Pantalla HMI utilizando el software respectivo a cada equipo.

### 4.1. CONFIGURACIÓN DEL PLC

Para PLC S7-200 seleccionado para esta aplicación, sus características se mencionan en el punto 2.9.8, en la **Figura 55** se muestra una vista general del conexionado del PLC S7-200, la programación se realiza desde el software del controlador STEP 7, donde se procede a realizar 2 pasos claramente identificados:

- La creación del proyecto y configuración de los controladores.
- La programación, compilación y carga del programa en el controlador.



**Figura 55 : Diagrama de conexión de PLC S7-200 y módulo de E/S**

#### 4.1.1. HARDWARE Y SOFTWARE NECESARIOS PARA LA PROGRAMACIÓN EN EL ENTORNO STEP 7.

Para proceder a la programación en el entorno STEP 7 se requiere cumplir con los requisitos básicos de HARDWARE Y SOFTWARE que continuación se detalla:



- PC, sistema operativo Windows 95/98/2000/me/nt 4.0 o superior
- Mínimo: 133mhz y 64mb ram.
- Aprox. 65 mb de espacio libre en disco duro.
- Óptimo: 500mhz y 128mb ram.
- Software step 7 v 5.x.
- Interfase MPI para PC (p.e. Pc- adapter).
- PLC simatic S7-200 con al menos un módulo de entradas/salidas.

Ejemplo de configuración:

- Fuente de Alimentación: PS 307 2ª.
- CPU: CPU 314.
- Entradas Digitales: DI 08x DC24V.
- Salidas Digitales: DO 08x DC24V / 0.5 A.

---

#### 4.1.2. ELEMENTOS DE PROGRAMACIÓN.

---

Para programar los PLC's se disponen de dos elementos básicos:

##### LOS OPERANDOS.

---

Los operandos identifican los diferentes tipos de variables de un programa, cuyos estados son almacenados en la memoria de datos. Estas variables corresponden a: puntos de entrada y salida, relés lógicos internos, memorias de temporizadores y contadores constantes, etc.

Algunos fabricantes, identifican a los operandos simplemente como archivos de datos, los que a su vez contienen elementos de memoria.

##### LAS INSTRUCCIONES.

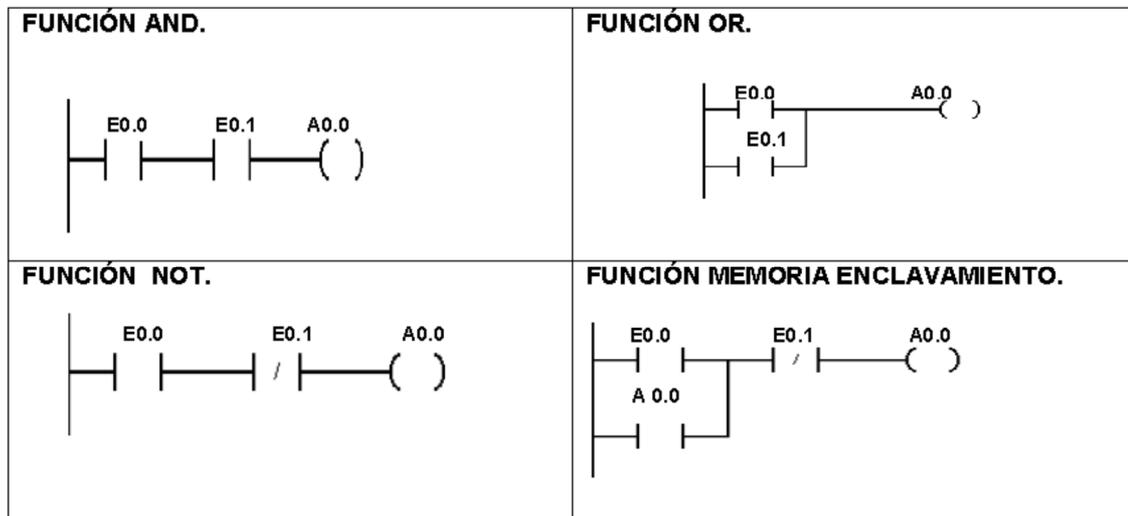
---

Son comandos de programa que permiten acceder, o alterar el estado de los operandos.

En forma general las instrucciones más comunes empleadas para programar a los controladores son:

- *Instrucciones lógicas:*

En el diagrama de escalera estas funciones se representan de la siguiente forma  
(Ver **Figura 56**).



**Figura 56 : Instrucciones lógicas.**

- *Instrucciones de módulos de programación:*
  - Temporizadores.
  - Contadores.
  - Comparadores.
  - Etc.
- *Instrucciones de control:*
  - Control de arranque.
  - Saltos condicionales.
  - Forzado de etapas.
  - Etc.
- *Instrucciones matemáticas:*
  - Suma.
  - Resta.
  - Multiplicación.
  - División.

### 4.1.3. LENGUAJE DE PROGRAMACIÓN.

El STEP 7 permite representar y programar un programa con los bloques de programación en tres clases diferentes:

#### ESQUEMA DE CONTACTOS KOP.

El esquema de contactos es la representación gráfica de las tareas de control con símbolos, tiene muchas similitudes con los esquemas de corriente, pero sin embargo esta representación no se ordena en sentido vertical, si no que se ordena en sentido horizontal.

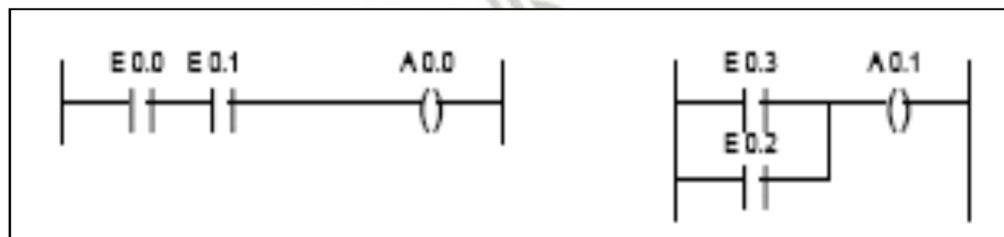
Los elementos básicos de este lenguaje son (Ver **Figura 57**):

**Contactos:** que representan interruptores por los que circula la corriente cuando está cerrado; lo que implica que existen dos tipos de contacto normalmente abiertos (N / A) y normalmente cerrados (N / C).

**Bobinas:** que representan relés que se excitan cuando se aplica voltaje.

**Cuadros:** que representan una función que se ejecuta cuando la corriente circula por él. Un cuadro puede representar, por ejemplo: un contador, un temporizador, etc.

**Segmentos:** que constituyen un circuito completo. La corriente circula desde la barra de alimentación ubicada a la izquierda pasando por los contactos cerrados para excitar las bobinas o cuadros.



**Figura 57 : Esquema de contactos KOP**

### DIAGRAMA DE FUNCIONES FUP.

El diagrama de funciones es la representación gráfica de las tareas de control con símbolos, las funciones individuales se representan a través de un símbolo con una función característica. En la parte izquierda del símbolo se ordenan las entradas y en la parte derecha las salidas (Ver **Figura 58**).



**Figura 58 : Diagrama de funciones FUP**

### LISTA DE INSTRUCCIONES AWL.

En la lista de instrucciones se describe la tarea de control con instrucciones individuales de control, las instrucciones de control (operación y operando) representan la tarea con mnemónicos de las designaciones de las funciones (Ver **Figura 59**).

Parte de la operación:	Parte del operando:	
	Característica	Parámetro
U	E	0.0
U	E	0.1
=	A	4.0
O	E	0.2
O	E	0.3
=	A	4.1

**Figura 59 : Lista de instrucciones AWL.**

Cada clase de representación contiene propiedades específicas y determinadas limitaciones, el control de programas que se programan en KOP o en FUP se pueden traducir normalmente a lista de instrucciones AWL.

En la memoria de programa de los aparatos se guarda siempre el programa en AWL (Realmente en código máquina).

#### 4.1.4. LISTA DE ASIGNACION DEL PROGRAMA

##### LISTA DE ENTRADAS

ITEM	SIMBOLO	DIREC	COMENTARIO
76	ENPUJEPISTONSS2	I2.3	SENSOR EXTENDIDO ENPUJE DE PISTON
77	ENPUJEPISTONSS1	I2.2	SENSOR RETRAIDO ENPUJE DE PISTON
78		I2.1	BLANCO
79	EJEZ_FIN	I1.5	SENSOR EJEY POSICION DE INICIO SECUENCIA
80	EJEZ_POS3	I1.4	SENSOR EJEY POSICION 3
81	EJEZ_POS2	I1.3	SENSOR EJEY POSICION 2
82	EJEZ_POS1	I1.2	SENSOR EJEY POSICION 1
83	EJEZ_REF	I1.1	SENSOR EJEY POSICION DE REFERENCIA
84	EJEX_FIN	I1.0	SENSOR EJEX POSICION DE INICIO SECUENCIA
85	EJEX_POS3	I0.7	SENSOR EJEX POSICION 3
86	EJEX_POS2	I0.6	SENSOR EJEX POSICION 2
87	EJEX_POS1	I0.5	SENSOR EJEX POSICION 1
88	EJEX_REF	I0.4	SENSOR EJEX POSICION DE REFERENCIA
89	PUL_REFERENCIA	I0.3	PULSADOR DE REFERENCIA
90	PAUSE	I0.2	PULSADOR DE PAUSE
91	INICIO	I0.1	PULSADOR DE INICIO
92	EMERGENCIA	I0.0	PULSADOR DE EMERGENCIA

**Tabla 14 : Lista de Asignación de Entradas en el PLC, según planos eléctricos**

##### LISTA DE SALIDAS

ITEM	SIMBOLO	DIREC	COMENTARIO
1	LUZ_PAUSE	Q2.0	LUZ DE PULSADOR DE PAUSE
3	LUZ_RESET	Q1.0	LUZ DE PULSADOR DE REFENCIA
4	LUZ_PULINI	Q0.7	LUZ DE PULSADOR DE INICIO
5	ENPUJE_PISTON	Q0.6	ACTUADOR QUE EMPUJA EL PISTON
6	MOTORIZ_ATARAS	Q0.3	RELE PARA MOVIMIENTO DE EJE Y - ATRÁS
7	MOTORIZ_DELANTE	Q0.2	RELE PARA MOVIMIENTO DE EJE Y - ADELANTE
8	MOTORX_ATRAS	Q0.1	RELE PARA MOVIMIENTO DE EJE X - ATRÁS
9	MOTORX_DELANTE	Q0.0	RELE PARA MOVIMIENTO DE EJE X - ADELANTE

**Tabla 15 : Lista de Asignación de salida en el PLC, según planos eléctricos**



## LISTA DE MEMORIAS

ITEM	SIMBOLO	DIREC	COMENTARIO
10	Z1X1	M31.0	
11	Z1X2	M30.7	
12	Z1X3	M30.6	
13	Z2X1	M30.5	
14	Z2X2	M30.4	
15	Z2X3	M30.3	
16	Z3X1	M30.2	
17	Z3X2	M30.1	
18	Z3X3	M30.0	
28	BOBINA_AUTOMATICO	M27.0	ACTIVACION DE SISTEMA AUTOMATICO
29	BOBINA_EJEZREF	M26.6	
30	BOBINA_EJEXREF	M26.5	
31	BOBINA_MANEJEZATRAS	M26.4	DESACTIVACION MANUAL MOTRO Y ADELANTE
32	BOBINA_MANEJEZDELANTE	M26.3	ACTIVACION MANUAL MOTRO Y ADELANTE
33	BOBINA_MANEJEXATRAS	M26.2	DESACTIVACION MANUAL MOTRO X ADELANTE
34	BOBINA_MANEJEXDELANTE	M26.1	ACTIVACION MANUAL MOTRO X ADELANTE
36	BOBINA_MANENPUPISTON1	M25.5	ACTIVACION MANUAL ENPUJADOR DE PISTON
37	BOBINA_MANUAL	M25.0	ACTIVACION DE SISTEMA MANUAL
38	Z1X1_HABILITACION	M18.5	INGRESO DE PISTON EN POSICION Z1X1
39	Z1X2_HABILITACION	M18.4	INGRESO DE PISTON EN POSICION Z1X2
40	Z1X3_HABILITACION	M18.3	INGRESO DE PISTON EN POSICION Z1X3
41	Z2X1_HABILITACION	M18.2	INGRESO DE PISTON EN POSICION Z2X1
42	Z2X2_HABILITACION	M18.1	INGRESO DE PISTON EN POSICION Z2X2
43	Z2X3_HABILITACION	M18.0	INGRESO DE PISTON EN POSICION Z2X3
44	Z3X1_HABILITACION	M17.7	INGRESO DE PISTON EN POSICION Z3X1
45	Z3X2_HABILITACION	M17.6	INGRESO DE PISTON EN POSICION Z3X2
46	Z3X3_HABILITACION	M17.5	INGRESO DE PISTON EN POSICION Z3X3
47	FILA3_ACCTIVACION	M17.4	PASO DE SUB PROGRAMA CASILLA EN FILA 1
48	FILA2_ACTIVACION	M17.3	PASO DE SUB PROGRAMA CASILLA EN FILA 2
49	FILA1_ACTIVACION	M17.2	PASO DE SUB PROGRAMA CASILLA EN FILA 3
52	BOBINA_NOTINICIO	M15.4	CANCELA EL PRIMER PASO DE SECUENCIA
53	PARADA_GENERAL	M15.3	SE LLENO TODAS LAS CASILLAS
54	PARADA3	M15.2	ACTIVACION 1 PARA PARADA GENERAL
55	PARADA2	M15.1	ACTIVACION 2 PARA PARADA GENERAL
56	PARADA1	M15.0	ACTIVACION 3 PARA PARADA GENERAL
57	HMI_EMERGENCIA	M8.3	
58	HMI_REFERENCIA	M8.2	PULSADOR DE REFERENCIA DE HMI



59	HMI_PAUSE	M8.1	PULSADOR DE PAUSE DE HMI
60	HMI_INICIO	M8.0	PULSADOR DE INICIO DE HMI
61	HMI_POSINICIAL	M6.5	
62	CICLO_CONTINUO	M4.3	
63	BOBINA_FINALEJEZ	M3.1	FLANCO DE POSICION INICIAL EJEX - FINAL
64	BOBINA_FINALEJEX	M2.7	FLANCO DE POSICION INICIAL EJEX - FINAL
65	BOBINA_POS3EJEZ	M2.3	FLANCO DE POSICION 3 EJEY
66	BOBINA_POS2EJEZ	M2.2	FLANCO DE POSICION 2 EJEY
67	BOBINA_POS1EJEZ	M2.1	FLANCO DE POSICION 1 EJEY
68	BOBINA_POS3EJEX	M2.0	FLANCO DE POSICION 3 EJEX
69	BOBINA_POS2EJEX	M1.7	FLANCO DE POSICION 2 EJEX
70	BOBINA_POS1EJEX	M1.6	FLANCO DE POSICION 1 EJEX
71	BOBINA1_PAUSE	M0.5	BOBINA DE PAUSE
72	BOBINA4_REF	M0.4	BOBINA PARA REFERENCIA 4
73	BOBINA3_REF	M0.3	BOBINA PARA REFERENCIA 3
74	BOBINA2_REF	M0.2	BOBINA PARA REFERENCIA 2
75	BOBINA1_REF	M0.1	BOBINA PARA REFERENCIA 1

**Tabla 16 : Lista de Asignación de memorias en el PLC.**

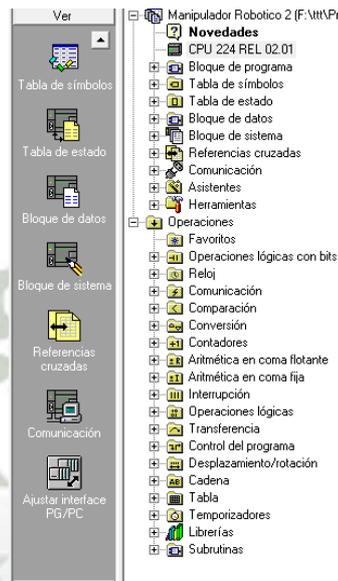
**SÍMBOLOS UOP**

ITEM	SIMBOLO	DIREC	COMENTARIO
1	REFERENCIA	SBR0	COMENTARIOS DE LA SUBROUTINA
2	AUTOMATICO	SBR1	COMENTARIOS DE LA SUBROUTINA
3	AUTOMATICO_FILA1	SBR2	COMENTARIOS DE LA SUBROUTINA
4	AUTOMATICO_FILA2	SBR3	COMENTARIOS DE LA SUBROUTINA
5	AUTOMATICO_FILA3	SBR4	COMENTARIOS DE LA SUBROUTINA
6	Toggle	SBR5	LIBRARY: Toggle (V1.1)
7	MANUAL	SBR6	COMENTARIOS DE LA SUBROUTINA
8	MANUAL_FILA1	SBR7	COMENTARIOS DE LA SUBROUTINA
9	MANUAL_FILA2	SBR8	COMENTARIOS DE LA SUBROUTINA
10	MANUAL_FILA3	SBR9	COMENTARIOS DE LA SUBROUTINA
11	INT_0	INT0	COMENTARIOS DE LA RUTINA DE INTERRUPTCIÓN
12	INT_1	INT1	COMENTARIOS DE LA RUTINA DE INTERRUPTCIÓN
13	PRINCIPAL	OB1	COMENTARIOS DEL PROGRAMA

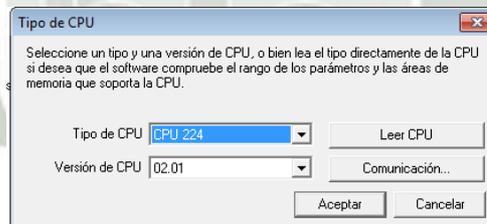
**Tabla 17 : Lista de Asignación de Subrutinas en el PLC**

#### 4.1.5. CARGA Y EJECUCION DE PROGRAMA

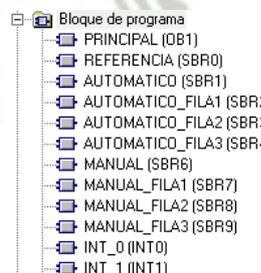
Se carga el archivo “Manipulador Robótico 2.mwp” con el programa STEP7-Micro/WIN, cargamos las opciones de configuración relativas al PLC-CPU S7-200 / 224 y la extensión de Entradas y Salidas EM 223, de acuerdo a los pasos impresos en las siguientes **Figura 60, Figura 61 y Figura 62:**



**Figura 60 : Lista de opciones.**



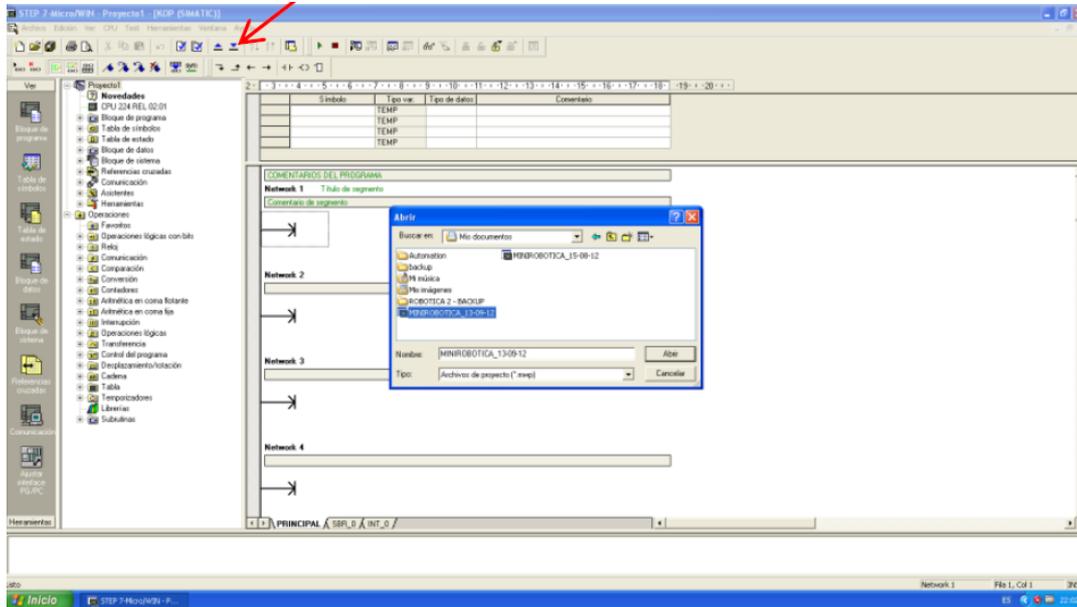
**Figura 61 : Selección de CPU**



**Figura 62 : Bloques de programa**

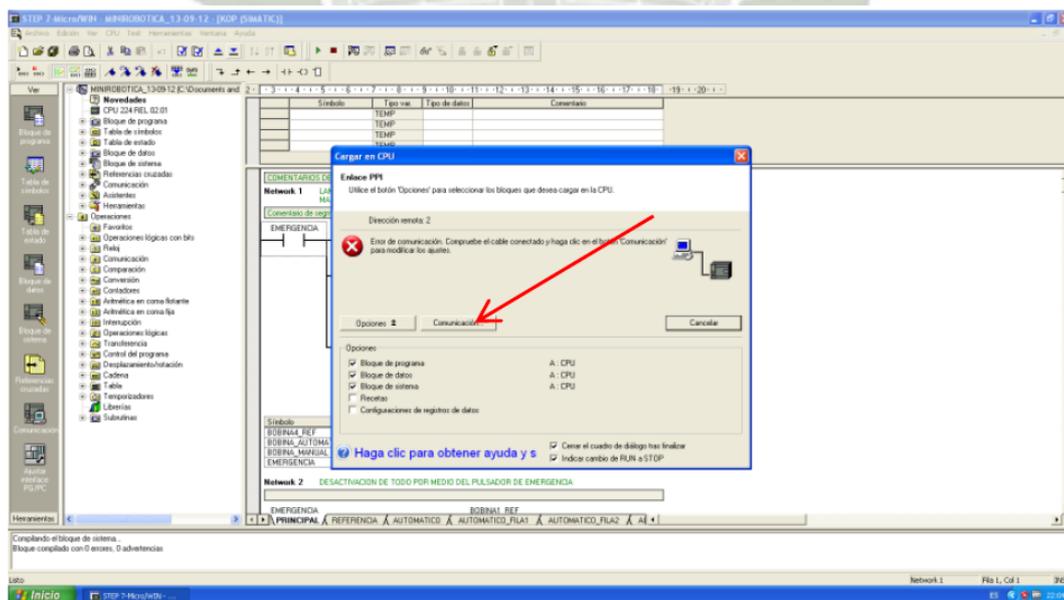
## DESCARGA DE PROGRAMA EN S7-200

Abrir el programa Microwin 4.0, luego abrir el archivo del Programa Inicial (Manipulador Robótico 2) se mostrada la pantalla mostrada en la **Figura 63**.

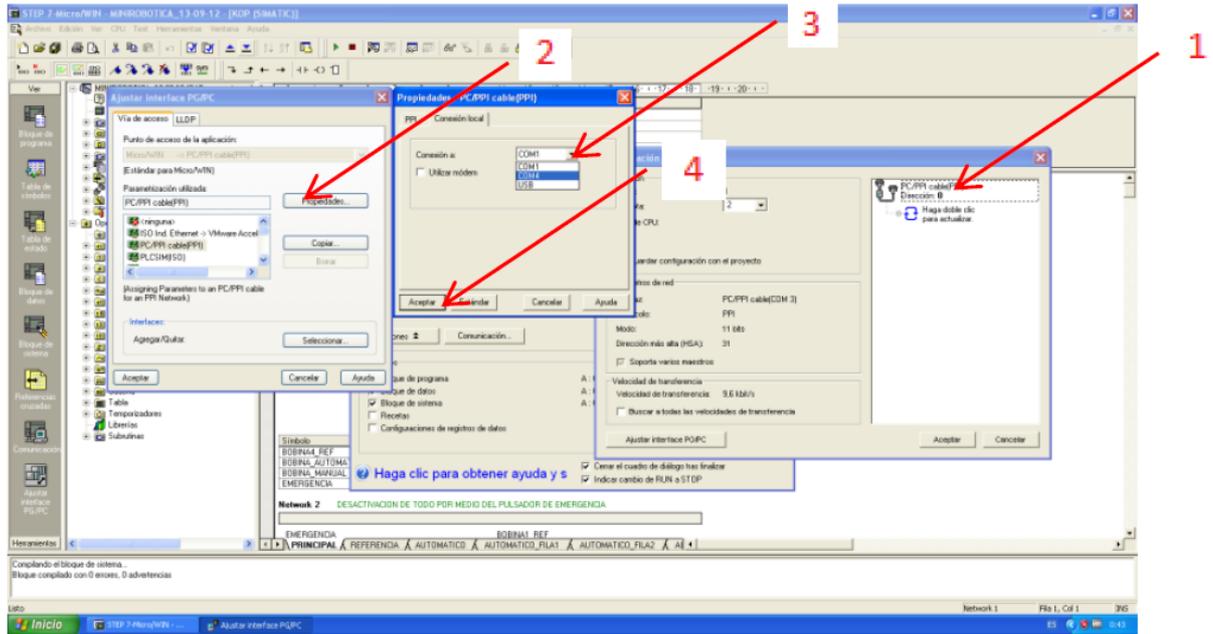


**Figura 63 : Apertura de archivo en STEP 7.**

Se presentará el proyecto seleccionado, luego descargar el programa en el PLC dando click en **Cargar en CPU**, y configurar el puerto para el cable PPI seguir los pasos numerados en las **Figura 64** y **Figura 65**.

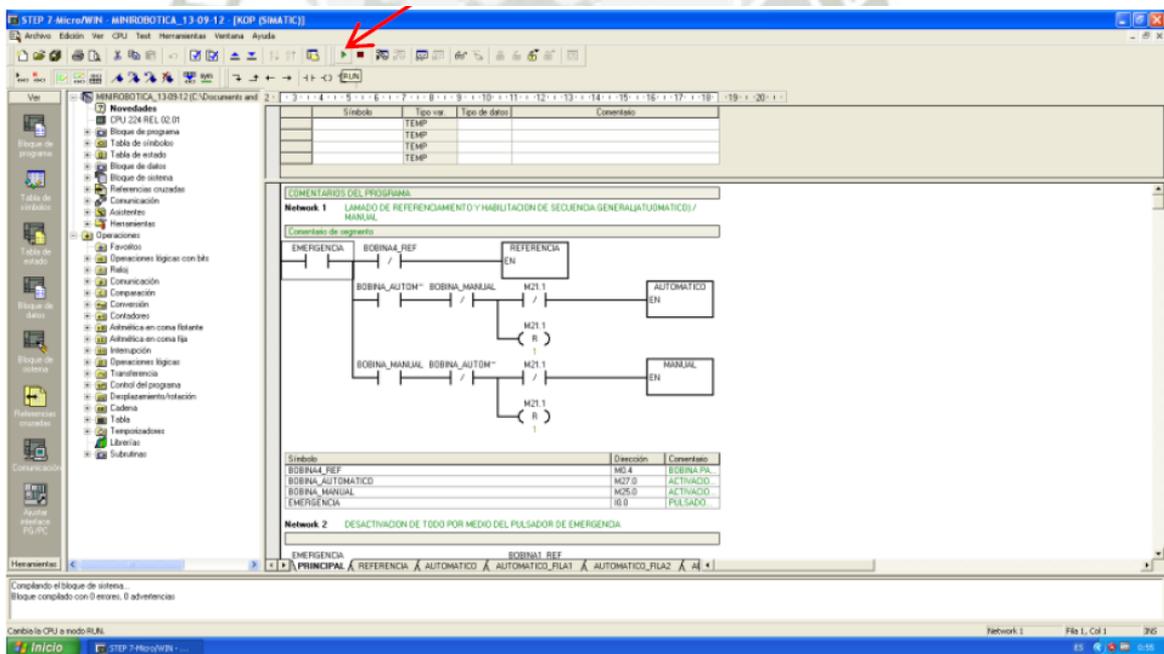


**Figura 64 : Carga del CPU- PLC**



**Figura 65 : Selección de Puerto de comunicación PPI**

Luego de haber configurado el puerto de comunicación para PPI, dar click en **Cargar a PC**. Por último dar click en **RUN** para iniciar el programa (ver **Figura 66**)



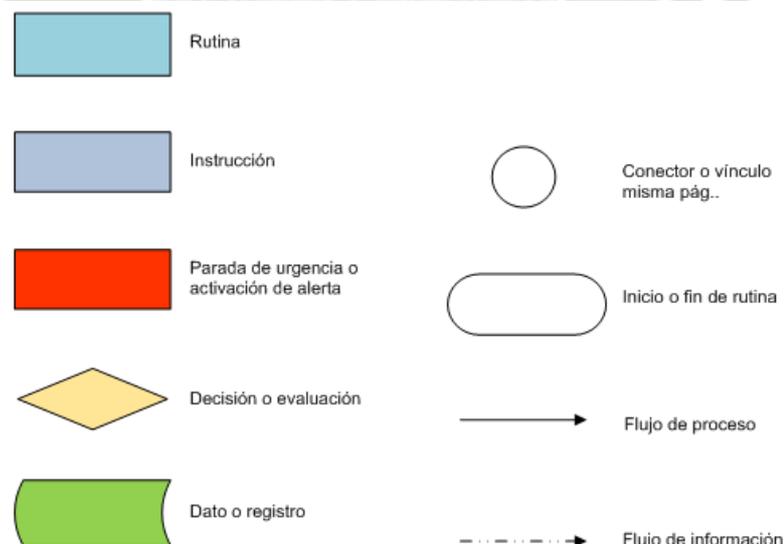
**Figura 66 : Se carga y corre el Programa**

Se anexa en la versión en Digital del presente trabajo un manual de Programación en STEP 7 para un PLC S7-200, en el Anexo 05, donde se amplían los pasos descritos en este trabajo y proponiéndose otros ejercicios de ejemplo.

#### 4.1.6. DIAGRAMAS DE FLUJO

Un diagrama de flujo es una representación gráfica de un proceso. Cada paso del proceso es representado por un símbolo diferente (ver **Figura 67**), que contiene una breve descripción de la etapa de proceso. Los símbolos gráficos del flujo del proceso están unidos entre sí con flechas que indican la dirección de flujo del proceso.

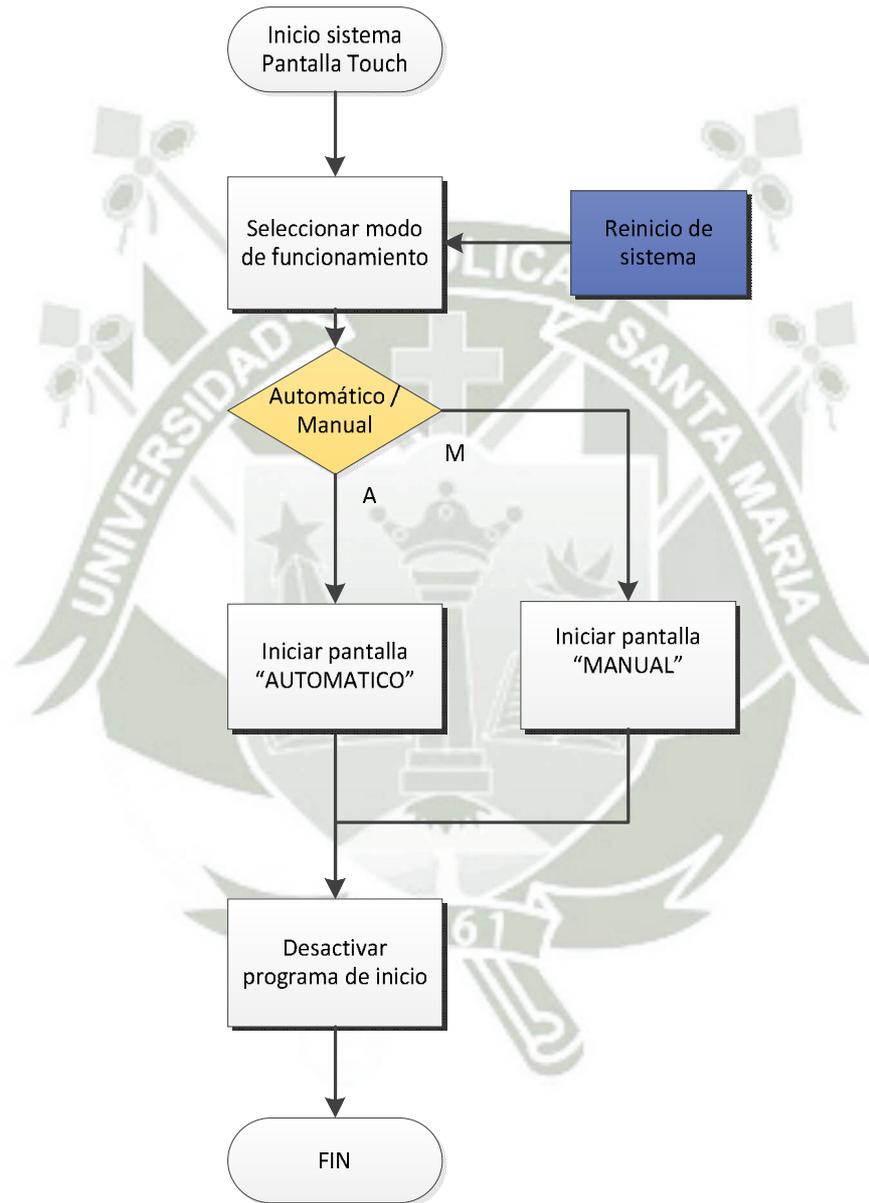
El diagrama de flujo ofrece una descripción visual de las actividades implicadas en un proceso mostrando la relación secuencial entre ellas, facilitando la rápida comprensión de cada actividad y su relación con las demás, el flujo de la información y los materiales, las ramas en el proceso, la existencia de bucles repetitivos, el número de pasos del proceso, las operaciones de interdepartamentales... Facilita también la selección de indicadores de proceso. La lógica de funcionamiento del módulo se encuentra representado en los siguientes diagramas de flujo, organizado de tal manera que se aprecie cómo funciona cada uno de los programas vistos anteriormente.



**Figura 67: Simbología de diagramas de flujo.**

Los diagramas de flujo para 03 procesos que se ejecutan según nuestra secuencia de funcionamiento, se muestran a continuación:

- Figura 68).
- Diagrama de flujo programa 2 / secuencia Automático (ver **Figura 69**).
- Diagrama de flujo programa 3 / secuencia Manual (ver **Figura 70**).



**Figura 68 : Diagrama de flujo programa 1 / secuencia principal.**

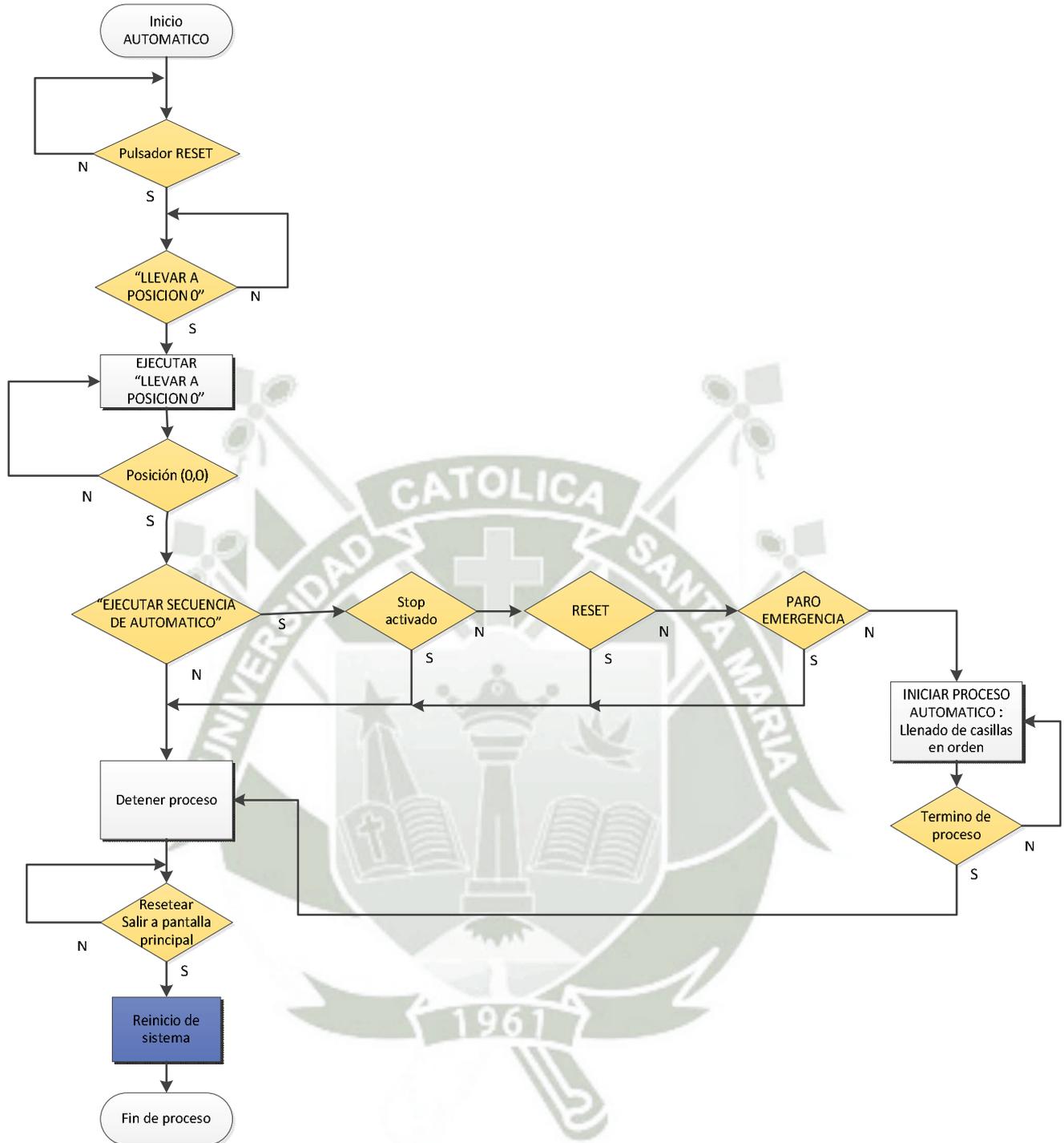


Figura 69 : Diagrama de flujo programa 2 / secuencia Automático.

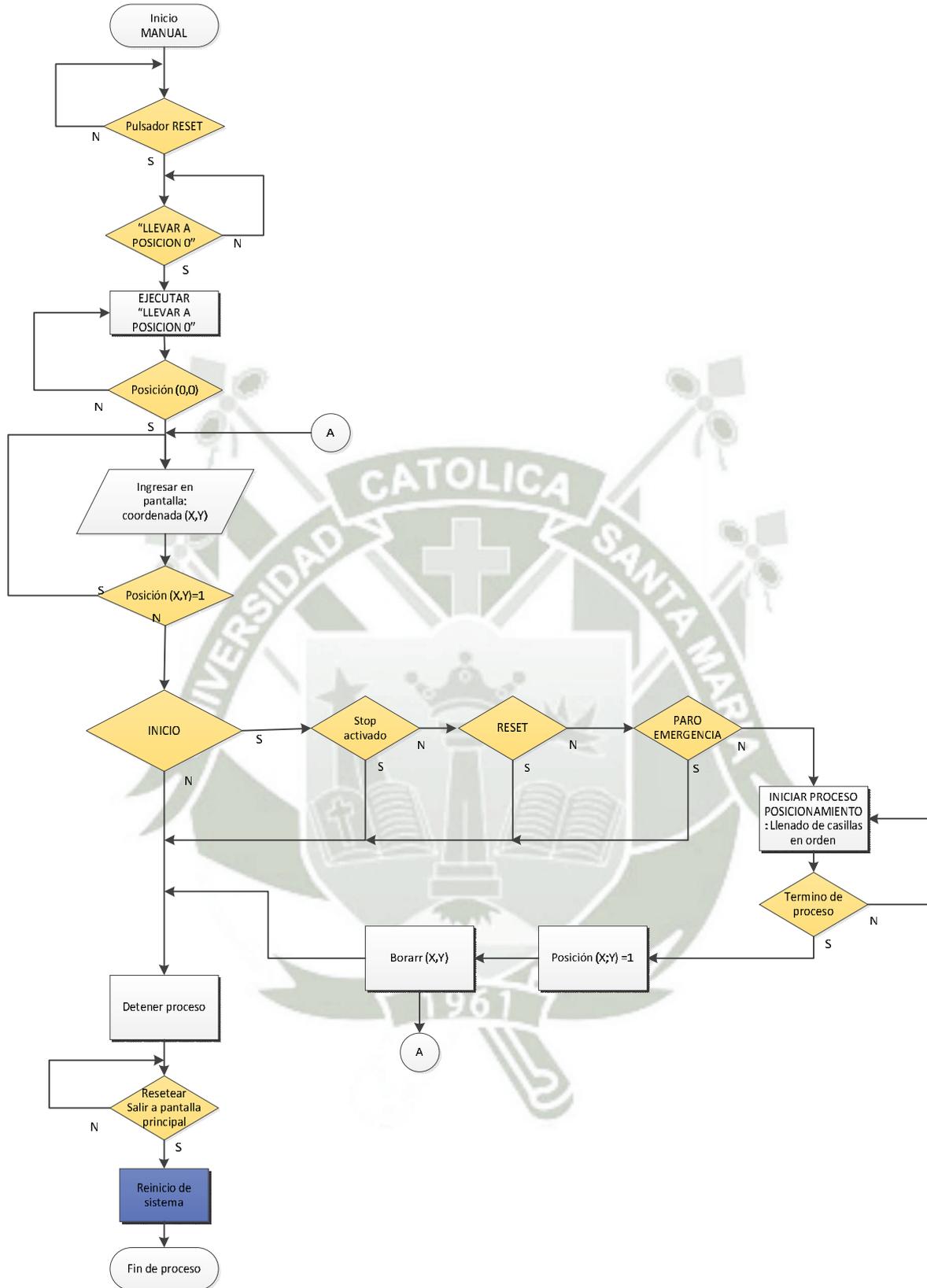


Figura 70 : Diagrama de flujo programa 3 / secuencia Manual.

## 4.2. DISEÑO DEL HMI

El diseño de una interfaz entre la máquina y el ser humano, debe permitir realizar una supervisión, monitoreo, almacenar una data histórica y poder establecer estados fuera de los parámetros de funcionamiento.

La capacidad que tienen muchos programas prediseñados para la elaboración de estas HMI como por ejemplo Wonderware, WinCC, etc. Son muy buenas, pero el tema de costos de licencias y del propio diseño del entorno, hacen que su uso no sea viable en pequeñas empresas cuyo presupuesto en muchas ocasiones no incluye una inversión en ello.

Para nuestro módulo, utilizamos una pantalla con las siguientes características:

Marca: MCGSTCP Modelo: TPC7062KX, es una pantalla de baja potencia con núcleo integrado de CPU (CPU ARM, velocidad de reloj de 400 MHz), de alto rendimiento integrado con pantalla táctil integrada, está diseñado con una pantalla de 7 pulgadas de alto brillo TFT LCD (resolución de 800 × 480), pantalla táctil resistiva de cuatro hilos (resolución de 1024 × 1024). Ver **Figura 71** y **Figura 72**.



**Figura 71 : Touch screen - HMI**



**Figura 72 : Características técnicas de la Pantalla**

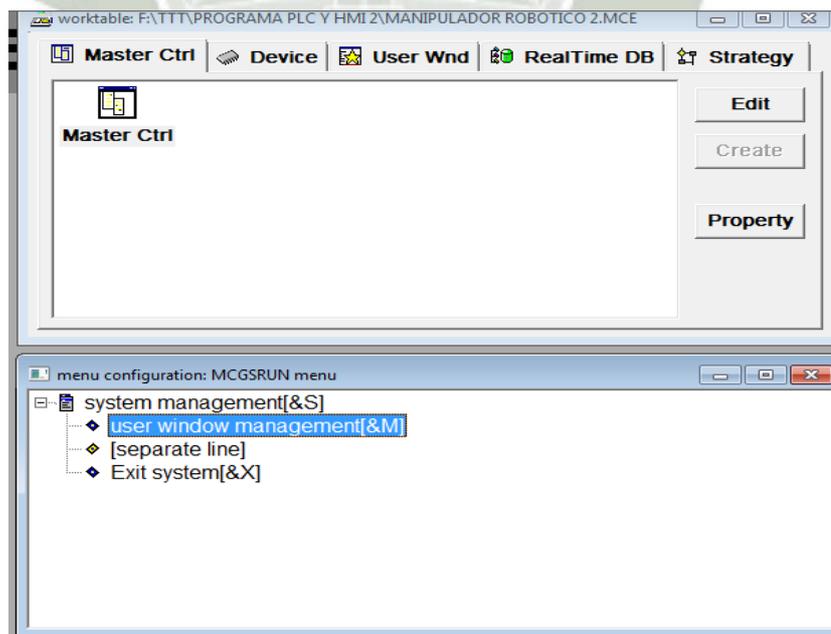
La pantalla es compatible con el Software de programación MCGS, es cual es libre y de fácil programación con un entorno amigable al programador:

MCGSE (Monitor and Control Generated System for Embedded) es un tipo de software de configuración utilizado para la construcción rápida y generación de sistema de seguimiento informático integrado, su entorno de configuración puede operar en diversas plataformas de Microsoft Windows de 32 bits, mientras que su entorno operativo funciona en el Windows CE como sistema operativo integrado multi-tarea en tiempo real. A través de la adquisición y procesamiento de datos sobre el terreno para proporcionar programas para resolver problemas prácticos de ingeniería de diversos medios de pantalla de animación, el procesamiento de alarmas, control de procesos y la salida de informes, etc., que es ampliamente utilizado en el campo de la automatización.

Software de configuración integrada MCGS se utiliza especialmente en sistemas operativos incorporados y es aplicable al sistema de ordenador dedicado para que el sistema de aplicación cuente con mejoras en la funcionalidad, fiabilidad, coste, volumen, consumo de energía, etc.

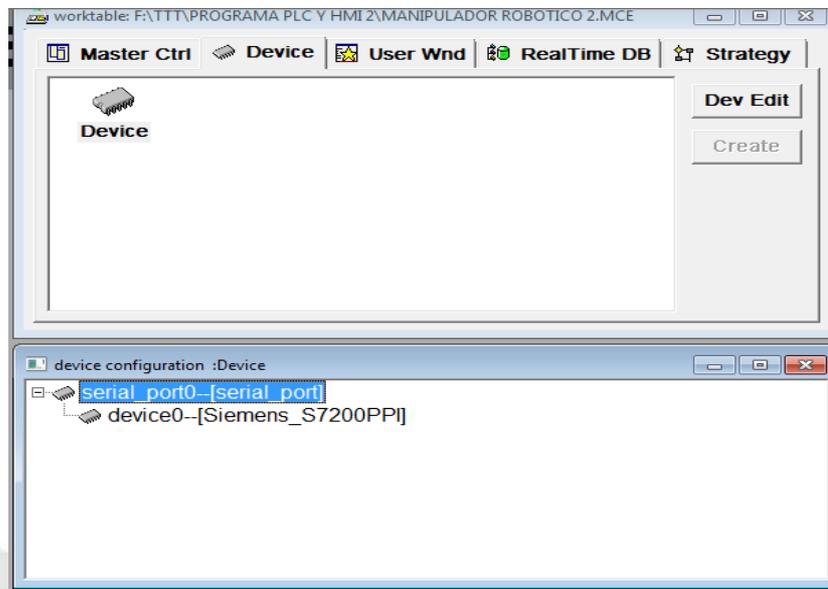
#### 4.2.1. DEFINIR PARÁMETROS INICIALES DE MGCS:

1. Abrir entorno MGCS, y seleccionar “menú de configuración” para Windows (ver **Figura 73**).



**Figura 73 : Entorno MGCS, inicio de programa**

2. Seleccionamos el dispositivo y medio de comunicación vía puerto Serial (ver **Figura 74**).



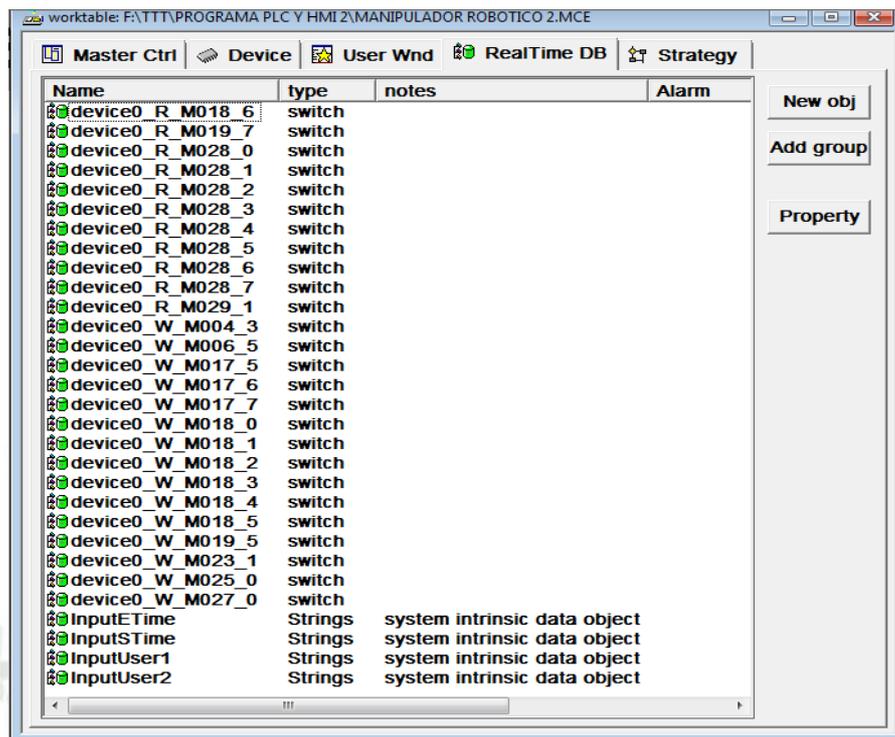
**Figura 74 : MGCS-selección de PLC y medio de comunicación**

3. A continuación en el archivo visualizamos en la mesa de trabajo las 03 ventanas de trabajo, la primera relacionada al programa principal, y las dos siguientes relacionadas a la secuencia en automático y manual, respectivamente (ver **Figura 75**).



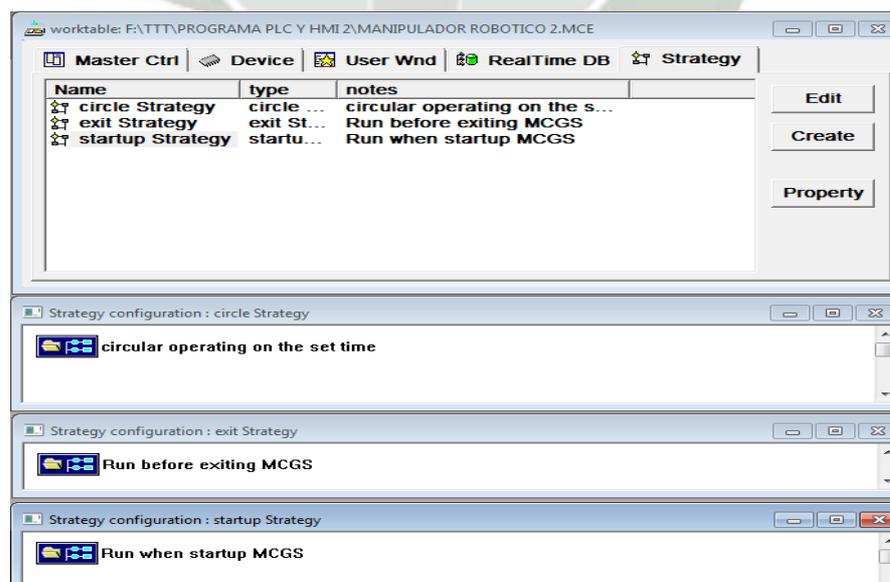
**Figura 75 : MGCS-Creación de Ventanas de Trabajo**

4. En la Pestaña "REAL TIME DB" debemos relacionar los valores: entradas, salidas, memorias, contadores a los creados en el controlador PLC, estos valores vinculados permitirán el control e inspección en tiempo real del proceso (ver **Figura 76**).



**Figura 76 : MGCS-Definición de señales de ingreso**

5. Por último en la pestaña "STRATEGY" creamos tres estrategias de actualización de datos 1) Inicio del sistema: STARUP que se relaciona al encendido de la pantalla; 2) Salida del Sistema: EXIT relacionada al antes del apagado de la pantalla y 3) Estrategia Circular: para la actualización de datos y operación en forma cíclica del sistema (ver **Figura 77**).



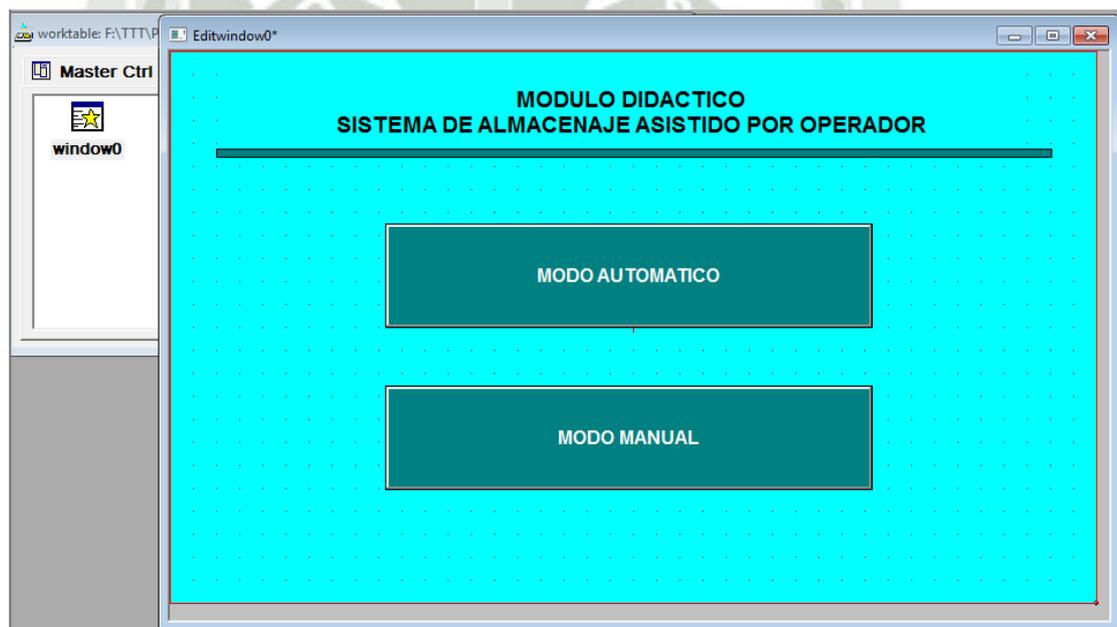
**Figura 77 : MGCS – Selección de estrategia de programa**

#### 4.2.2. PROGRAMACIÓN EN LA PANTALLA HMI

En la opción de la pestaña Windows, definimos las 03 ventanas con las que trabajamos, ahora en la etapa de diseño trabajamos en cada una de las ventanas, de manera secuencial y ordenada seguimos los siguientes pasos de programación:

##### WINDOWS 0: PANTALLA PRINCIPAL

Utilizando las herramientas del Toolbox se editan 02 botones a los cuales nombramos de acuerdo a los modos de funcionamiento Automático y Manual respectivamente; así como el resto de elementos que aparecen en la ventana: Título, color de fondo, estilo de letra y otros elementos de diseño de la ventana (ver **Figura 78**).



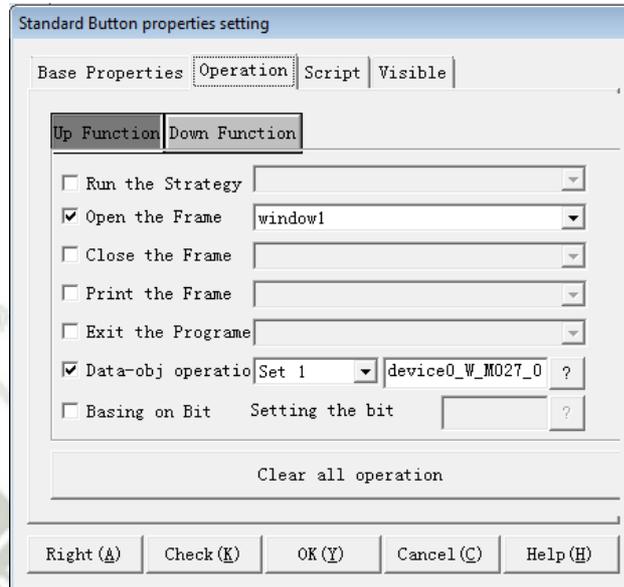
**Figura 78 : MGCS- Ventana de inicio**

Ahora vamos a la programación y relación de elementos, para esta pantalla inicial las opciones las indicamos en el menú de opciones de cada STAR BUTTON, en la pestaña Operation definimos operaciones a realizarse con la selección de cada botón;

**BOTON AUTOMATICO:** Acciones:

Abre la Ventana 1: modo Automático (ver **Figura 79**).

Setea a 1 la variable M027.0

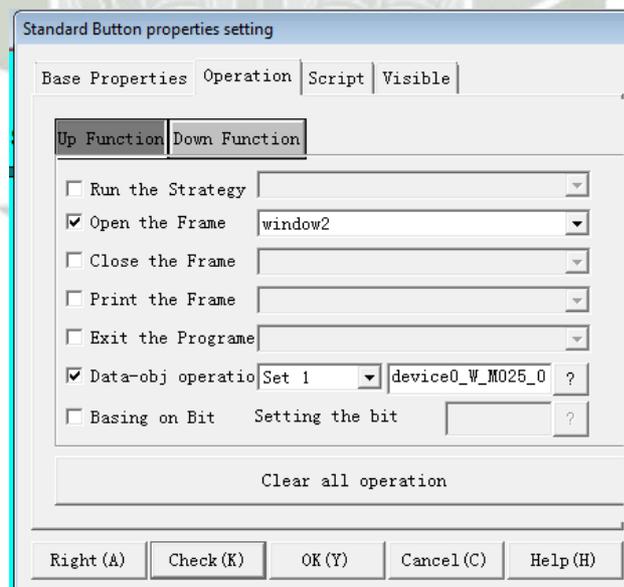


**Figura 79 : MGCS – Programación botón de automático**

**BOTON MANUAL:** Acciones:

Abre la Ventana 2: modo Manual (ver **Figura 80**).

Setea a 1 la variable M025.0



**Figura 80 : MGCS – Programación botón de Manual**

## WINDOWS 1: SECUENCIA AUTOMÁTICO

El programa es más complejo, con las diferentes opciones que posee el TOOLBOX creamos la siguiente ventana (ver **Figura 81**), esta debe contar con las características necesarias para realizar el control en tiempo real del proceso:

- Título del trabajo en la parte superior
- Icono de Salida que permita el regreso a Pantalla Principal
- Botón "LLEVAR A POSICION 0" al cual vinculamos a proceso de posicionar en la posición (0,0) nuestro actuador principal
- Indicador POSICION 0, nos da la confirmación de la ubicación del actuador
- Botón "EJECUTAR SECUENCIA - AUTOMATICO" el cual previa confirmación de la ubicación (0,0) dará inicio a la secuencia de llenado de almacén en forma automática y secuencial.
- 09 Indicadores de Posición que indicaran el estado de cada una de las 09 posiciones de almacenaje, constituido por 03 columnas y 03 filas, estos indicadores se relacionan con las variables correspondientes y constituyen la visualización en tiempo real del proceso de almacenaje.



**Figura 81: MGCS- Ventana de Secuencia Automática**

## WINDOWS 2: SECUENCIA MANUAL

El programa se elabora con las diferentes opciones que posee el TOOLBOX, creamos la siguiente ventana (ver **Figura 82**), esta debe contar con las características necesarias para realizar el control en tiempo real del proceso:

- Título del trabajo en la parte superior
- Icono de Salida que permita el regreso a Pantalla Principal
- Botón "LLEVAR A POSICION 0" al cual vinculamos a proceso de posicionar en la posición (0,0) nuestro actuador principal
- Indicador POSICION 0, nos da la confirmación de la ubicación del actuador
- Indicador "ESCOJA LA DIRECCION DONDE ALMACENAR" solo como un cuadro de texto de cabecera.
- 09 Botones de Ingreso; ingreso de coordenada donde se desea almacenar el producto inicial, en una constitución de por 03 columnas por 03 filas, una vez solicitado el ingreso de la pieza se inicia el proceso de almacenaje en dicha posición.
- 09 Indicadores de Posición que confirmará la ejecución de un proceso de almacenaje en las 09 posiciones de almacenaje, constituido por 03 columnas y 03 filas, estos indicadores se relacionan con las variables correspondientes y constituyen la visualización en tiempo real del proceso de almacenaje.



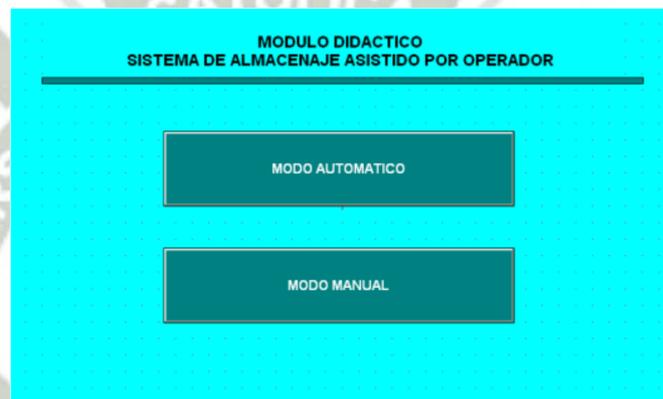
**Figura 82 : MGCS- Ventana de Manual**

Para mayor información anexamos un manual de funcionamiento y operación del programa MGCS, en la versión en Digital (CD-ROM) con ejemplos de programación y funcionamiento del programa tanto para módulos didácticos, como este, como para procesos industriales de mayor complejidad.

#### 4.3. INSTRUCTIVO DE FUNCIONAMIENTO MÓDULO DIDACTICO

##### 4.3.1. MODO AUTOMÁTICO:

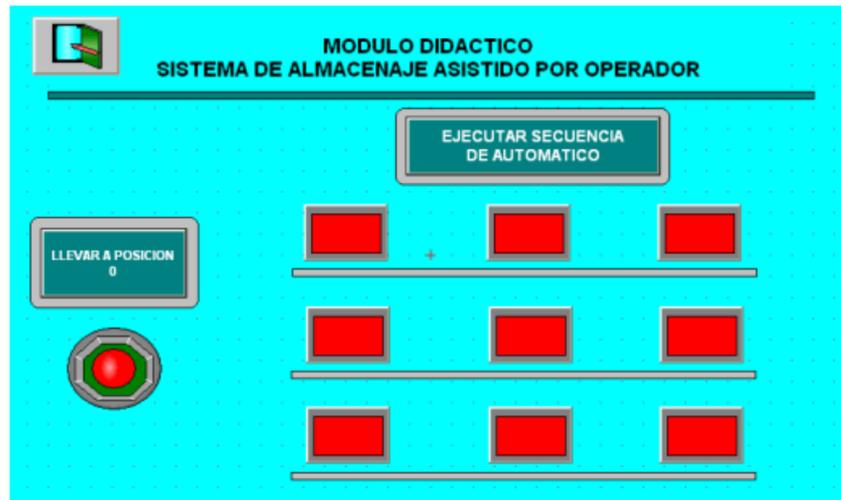
Para iniciar el MODO AUTOMATICO se ha de presenciar la página principal del HMI (ver **Figura 83**).



**Figura 83 : HMI – Pantalla de inicio – Selecciona Automático**

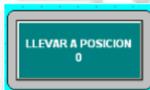
Ingresar al modo Automático presionando el botón que lo identifica, luego en la pantalla de MODO AUTOMATICO (ver **Figura 84**), se visualiza los botones de LLEVAR A POSICION 0 y EJECUTAR SECUENCIA DE AUTOMATICO, así mismo los indicadores de cada posición.

Antes de ingresar al Modo Automático se ha de hacer primero un reseteo del sistema con el pulsador de Emergencia, si desea puede realizar un referenciamiento de los ejes ya que este es independiente del modo automático y mal.



**Figura 84 : HMI – Pantalla secuencia Automático**

Presionar el pulsador físico de color verde de la botonera para activar la secuencia del sistema, este pulsador encenderá su propio led indicador para mostrar la segura activación del sistema Automático.

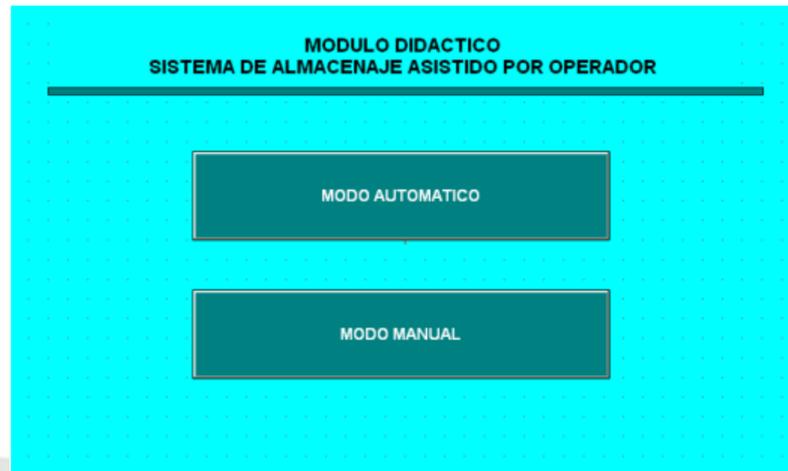
Pulsar el botón de LLEVAR A POSICION 0  del HMI AUTOMATICO para llevar los ejes a la posición de inicio de trabajo, se tiene que realizar este paso para poder realizar el trabajo.

Para salir del Modo Automático pulsar el botón  , así se reseteara todo el sistema y se ingresara a la página principal

Si desea pausar el sistema automático presionar el pulsador físico rojo, este apagará el indicador del pulsador verde y encenderá el indicador del propio pulsador.

#### 4.3.2. MODO MANUAL:

Para la manipulación del MODO MANUAL se ha de presenciar la página principal del HMI (ver **Figura 85**).



**Figura 85 : HMI – Pantalla de inicio – Selecciona Manual**

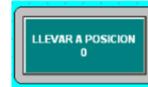
Ingresar al modo manual presionando el botón que lo identifica, luego en la pantalla de MODO MANUAL (ver **Figura 86**), se visualizará el botón de LLEVAR A POSICION 0 y los pulsadores e indicadores de cada posición.



**Figura 86 : HMI – Pantalla secuencia Manual**

Antes de ingresar al Modo Automático se ha de hacer primero un reseteo del sistema con el pulsador de Emergencia, si desea puede realizar un referenciamiento de los ejes ya que este es independiente del modo automático y manual.

Presionar el pulsador físico de color verde de la botonera para activar la secuencia del sistema, este pulsador encenderá su propio led indicador para mostrar la segura activación del sistema Manual.



Pulsar el botón de LLEVAR A POSICION 0 del HMI AUTOMATICO para llevar los ejes a la posición de inicio de trabajo, se tiene que realizar este paso para poder realizar el trabajo. Para realizar el trabajo manual presionar el pulsador e indicador de la posición que desea alojar la tapa (ver **Figura 87**).



**Figura 87 : HMI – Pantalla secuencia Manual – Indicadores de Llenado de Piezas**

Los ejes se desplazaran al lugar seleccionado y al dejar la tapa en su posición se encenderá el led de la posición seleccionada.

Para salir del Modo Manual pulsar el botón , así se reseteara todo el sistema y se ingresara a la página principal



## CAPITULO 5: ANALISIS DE COSTOS

Para el desarrollo del siguiente proyecto se realizó una inversión para desarrollar el trabajo de investigación planteado y recursos necesarios para la implementación. Este análisis se divide en dos los cuales son:

- a) Materiales para la elaboración del módulo
- b) Costos de ingeniería

### 5.1. MATERIALES PARA LA ELABORACIÓN DEL MÓDULO

En el desarrollo del proyecto se tuvieron que desarrollar varios diseños e implementaciones para escoger el más óptimo. En la siguiente Tabla se listan los Materiales Estructurales utilizados en el módulo (todos los precios en nuevos soles peruanos).

Dispositivo	Observaciones	Cant	Precio Unit	Costo (S/.)
Perfil de Aluminio Balaustre	Barra de 6 metros	1	53.00	53.00
Mesa de aluminio	Pl Rexroth 1.4 m x 0.75 m	1	450.00	450.00
Manufactura de soporte de piezas	Estantería.	3	30.00	90.00
Bases circulares	Estantería y HMI	4	15.00	60.00
Plancha 0.4x0.25x0.002 m	Base Botonera	1	8.00	8.00
Manufactura de elementos estructurales deslizador vertical		1	130.00	130.00
Manufactura de elementos estructurales deslizador horizontal		1	130.00	130.00
Manufactura de elementos de Herramienta pistón		1	65.00	65.00
Pernos varios tipo Allen		36	0.80	28.80
Rodamientos para deslizadores		2	15.00	30.00
Accesorios, herramientas varios		1	150.00	150.00
Sumatoria en Soles				<b>S/ 1,194.80</b>

**Tabla 18 : Tabla de Costos de Materiales Estructurales**

Para los Materiales Eléctricos y Neumáticos utilizados en el Módulo, se elabora la siguiente Tabla 19 del costo:



Dispositivo	Observaciones	Cant	Precio Unit	Costo (S/.)
Compresor de aire	220V, 60 Hz, 600W	1	450.00	450.00
Unidad FRL	AIRTAC	3	160.00	480.00
Motor DC para deslizadores	Tensión 24 V	2	50.00	100.00
Electroválvula Festo 5/2	CPE10-M1BH-5L-M5	3	210.00	630.00
Manifold		1	90.00	90.00
Sensores Magnéticos		2	80.00	160.00
Sensores de Posición		8	30.00	240.00
Relés	2 contactos	6	10.00	60.00
Cilindro Neumático AIRTAC	Doble efecto	1	180.00	180.00
PLC Siemens S7-200		1	1355.00	1355.00
HMI MCGS	modelo TPC 7062	1	1700.00	1700.00
Disyuntor Termo-magnético y diferencial.		1	35.00	35.00
Botones		3	45.00	135.00
Manguera neumática de 6 y 4 mm	Metros	15	2.50	37.50
Rollo de cable N° 18	100 metros	1	65.00	65.00
Accesorios eléctricos para tablero	Canaletas, Etiquetas , cintillos y tornillos	1	200.00	200.00
<b>Sumatoria en soles</b>				<b>S/. 5,917.50</b>

**Tabla 19 : Tabla de Costos de Materiales Eléctricos – Neumáticos**

En resumen se muestra en **Tabla 20** para el costo total de Materiales:

Subtotales	Valor (S/.)
Costos Estructurales	1194.8
Costos Eléctricos-Neumáticos	5917.5
<b>Total Costos Materiales</b>	<b>S/. 7,112.30</b>

**Tabla 20 : Tabla de Costos total de materiales**



## 5.2. COSTOS DE INGENIERÍA

Estos están relacionados al aporte profesional y de diseño requeridos para que el módulo pueda tener la funcionalidad necesaria se detalla en la **Tabla 21**:

ETAPA	Observaciones	Costo (S/.)
Ingeniería y Diseño	CAD y Programación: Dos personas dedicadas 8 hora/día -2 meses	4,000.00
Construcción e instalación	Ensamblaje y conexiónado y commisioning - 1 semana	2,500.00
Programas y Software	MS Office e Internet (Instalado en un PC)	0
Gastos Varios	Transporte, asesoría, alimentación , etc.	2,000.00
<b>Total Costos Ingeniería</b>		<b>S/. 8,500.00</b>

**Tabla 21 : Tabla de Costos de Ingeniería**





## CONCLUSIONES Y RECOMENDACIONES

### CONCLUSIONES

A lo largo de este trabajo se han podido encontrar resultados satisfactorios e interesantes, a continuación mencionaremos las conclusiones de cada capítulo que definen el trabajo realizado.

**Primero.-** El proyecto de tesis desarrollado cumple con el diseño e implementación del módulo para la automatización de un almacenaje industrial controlado mediante un PLC Siemens S7-200 y supervisado por un sistema SCADA.

**Segundo.-** En el módulo implementado se obtiene una optimización del 200% en el espacio de almacenaje, donde antes se almacenaban 03 piezas hoy se almacenan 9 piezas.

**Tercero.-** La optimización de tiempo y recurso Humano (HH) es del 77%, debido a la implementación realizada se reduce el tiempo de almacenaje desde un punto de distribución a 9 posiciones de almacenaje diferentes contra solo el apilamiento para la alimentación del módulo de almacenamiento., ahorramos tiempo del personal operativo.

**Cuarto.-** Para el diseño e implementación de los soportes estructurales y mecánicos, el material utilizado fueron perfiles y bloques de aluminio manufacturados, debido a su facilidad de manejo, manufacturación, peso y resistencia que se necesitaba para este sistema ya que se trata de un prototipo con fines educativos, se anexan los planos de construcción ( Ver Anexo 4).

**Quinto.-** En base a los planos Eléctricos y Neumáticos elaborados (Ver anexo 3), se realizó la correcta instalación de los siguientes dispositivos: (sensores, actuadores, interfaces de comunicación), realizándose el cableado eléctrico y el tendido de líneas de aire comprimido a los dispositivos que conforman el módulo de Almacenamiento Vertical.

**Sexto.-** La elaboración del diagrama de bloques del proceso desarrollado en el capítulo 4 del presente trabajo es la base para la programación mediante secuencia Ladder del Controlador Lógicos Programable (PLC).



**Séptimo.-** Las capacidades adicionales que se le puedan otorgar a un controlador PLC siempre deben respetar el hecho que su principal función es la de controlar los procesos eficientemente.

**Octavo.-** La implementación del sistema SCADA como HMI desarrollado en el proyecto es una forma rápida y económica de supervisión usando recursos de bajo coste, al alcance de empresarios de PYMES, orientados a una productividad regular contando con tecnologías de automatización e información industrial de buena calidad.

**Noveno.-** La configuración del HMI y el diseño de la pantalla táctil se realizó de una manera sencilla y clara, permitiendo una mayor interacción entre el operador y el sistema, permitiendo un trabajo rápido y eficiente.

**Décimo.-** Después del diseño e implementación mecánica, eléctrica y programación, se realizaron las pruebas respectivas verificando así el correcto funcionamiento del módulo de almacenaje vertical.

**Undécimo.-** El desarrollo de las prácticas elaboradas enfocadas en el módulo de un proceso industrial (almacenaje) permite al alumno familiarizarse y entender el principio de funcionamiento de determinados procesos industriales, ayudándolo a elegir adecuadamente los distintos elementos que interviene en la automatización con controladores industriales (ver anexo 6).



## RECOMENDACIONES

---

También es importante realizar algunas observaciones, producto de las metodologías usadas durante la realización de este sistema, esto se plasman en recomendaciones que ayudaran a futuros trabajos de investigación y desarrollo tecnológico en nuestra facultad.

- a) El actual diseño del módulo sirve de base para comprender los almacenes automatizados industriales, pudiéndose mejorar y optimizarlo, en el afán de encontrar las mejores opciones en un posterior diseño en una industria u otra aplicación en automatización.
- b) Existen varios diseños de almacenes industriales, el módulo usado en este proyecto solo es un tipo de los tantos que se pueden diseñar e implementar en la industria.
- c) Una limitación del módulo de almacenaje es la incapacidad dentro de la matriz de almacenamiento de productos de conocer si físicamente existe o no un producto colocado dentro de cada celda de la misma, es por esto que en ocasiones se puede engañar al sistema. Esta limitación implicaría una mejora para el módulo, implementando sensores de presencia dentro de cada celda o un sensor en el brazo actuador, para saber de forma precisa si existe o no presencia de algún producto.
- d) El módulo implementado es un módulo amigable en el que se pueden presentar cambios y/o mejoras tanto en la programación como en la adición de componentes, sensores, señales y herramientas.
- e) Los finales de carreras por ser sensores de contacto han de tener un periodo de vida útil corto, se recomienda realizar el cambio por sensores magnéticos o inductivos como una mejora propuesta.
- f) El equilibrio entre la funcionalidad y la cantidad de información que se presenta en una interfaz SCADA o HMI debe ser siempre aquel que permita la toma de decisiones y la capacidad de acción más alta posible, acompañado claro con el criterio de viabilidad de implementación tanto económica como técnica.



- g) En la parte de configuración y programación se enseña cómo se puede programar paso a paso, lo cual puede usarse como referencia para alguna modificación o mejora por parte del futuro usuario.
- h) Se recomienda para el uso del módulo algunos puntos importantes:
- Leer detenidamente el instructivo de funcionamiento y características técnicas.
  - Evitar la manipulación no autorizada de los distintos mecanismos y o dispositivos de alto riesgo, por ejemplo: líneas eléctricas de 220 VAC, líneas neumáticas, etc.
  - Al momento de energizar el módulo asegurarse que las conexiones especialmente de alimentación, estén correctamente conectadas.
  - Cumplir con la ley de salud y seguridad ocupacional en el trabajo.





## BIBLIOGRAFÍA

- [1]. **E. García.** "Automatización de Procesos Industriales". AlfaOmega Grupo Editor S.A. Universidad Politécnica de Valencia- México D.F. 2001.
- [2]. **A. Rosado.** "Sistemas industriales distribuidos: Una filosofía de automatización". Universidad de Valencia Dpto. Ing. Electrónica, 2003.
- [3]. **W. Bolton.** "Mecatrónica: Sistemas de Control Electrónico en Ingeniería Mecánica y Electricidad". Addison Wesley Longman 1999.
- [4]. **R. Biekert.** "CIM Technology Fundamentals and applications". Goodheart-Willcox Company, Inc. Tinley Park Illinois 1998.
- [5]. **J. Balcells, J. L. Romeral.** "Autómatas Programables". AlfaOmega Grupo Editor S.A. - México D.F. 1997.
- [6]. **D. J. Dorantes.** "Automatización y Control", Mc Graw Hill, 2004.
- [7]. **K. Ogata.** "Ingeniería de Control Moderna", 3ra Ed., Pearson.
- [8]. **J. Carranza,** "Tópicos de instrumentación y control", Edición especial de Ingeniería electrónica PUCP, Lima 2002.
- [9]. **R. Piedrafita.** "Ingeniería de la Automatización". AlfaOmega Grupo Editor S.A. - México D.F. 2001.
- [10]. El diario de un logístico, (<http://eldiariodeunlogistico.blogspot.com/2011/09/definicion-funcion-y-clases-de-almacen.html>)
- [11]. **Humberto Jasso,** Administración de Operaciones (<http://es.scribd.com/doc/94485660/ALMACENES>)
- [12]. **José P. García –Sabater,** Diseño de Sistemas Productivos y Logísticos (<http://personales.upv.es/jpgarcia/LinkedDocuments/7%20Almacenes.pdf>)
- [13]. **Maria Moneo,** Automatización Carruseles verticales, Artículo Carruseles verticales de gran altura. (<http://img.mecalux.pe/external/magazine/41899.pdf>)
- [14]. <http://www.sc.ehu.es/sbweb/webcentro/automatica/WebCQMH1/PAGINA%20PRINCIPAL/Automatizacion/Automatizacion.htm>
- [15]. Marco Teórico / 6.1 LOS PLC (<http://sistemamid.com/preview.php?a=78613>)
- [16]. Siemens, Manual del sistema de Automatización S7-200 (<http://isa.uniovi.es/~vsuarez/Download/Simatic%20S7200.pdf>)
- [17]. Manual Teórico PLC / 1.1 Introducción (<http://es.scribd.com/doc/77252987/Manual-Teorico-PLC#download>)
- [18]. Introducción a HMI / Estructura General de Software HMI (<http://iaci.unq.edu.ar/materias/laboratorio2/HMI%5CIntroduccion%20HMI.pdf>)
- [19]. Desarrollo de un Sistema HMI para un almacén automatizado, [http://e-archivo.uc3m.es/bitstream/handle/10016/16073/Proyecto\\_ESTHER\\_SALICHS.pdf](http://e-archivo.uc3m.es/bitstream/handle/10016/16073/Proyecto_ESTHER_SALICHS.pdf)
- [20]. **Aquilino Rodríguez** "Sistemas SCADA". 2da Ed. Editorial Alfaomega. México. 2007.
- [21]. <http://es.scribd.com/doc/202955998/Desarrollo-de-Un-Sistema-HMI-Para-Un-Almacen-Automatizado>



## ANEXOS LISTADO

**ANEXO 1** – TECHNICAL FILE

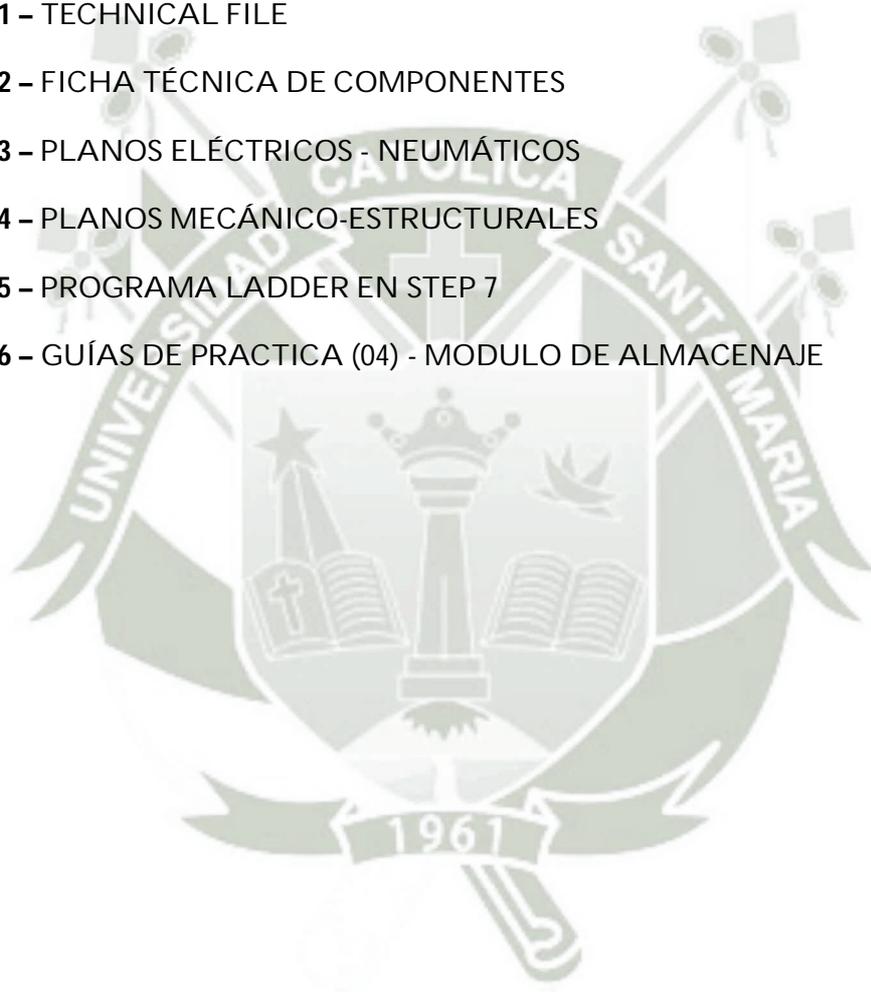
**ANEXO 2** – FICHA TÉCNICA DE COMPONENTES

**ANEXO 3** – PLANOS ELÉCTRICOS - NEUMÁTICOS

**ANEXO 4** – PLANOS MECÁNICO-ESTRUCTURALES

**ANEXO 5** – PROGRAMA LADDER EN STEP 7

**ANEXO 6** – GUÍAS DE PRACTICA (04) - MODULO DE ALMACENAJE





# ANEXO 1

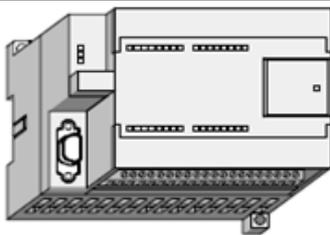
## TECHNICAL FILE



**Technical file for the system**

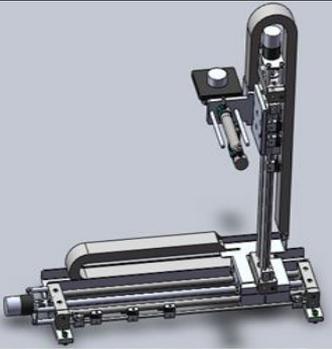
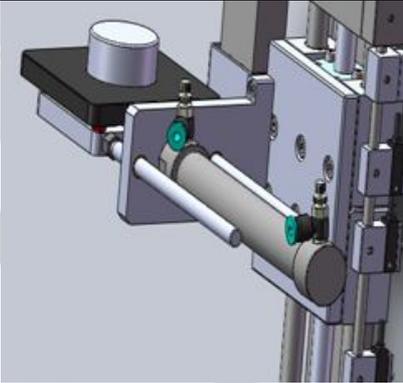
- 1、 Single AC power : AC220V±10% 50Hz;
- 2、 Temperature : -10°C ~ 40°C ; Humidity: ≤90% (25°C)
- 3、 Capacity: ≤1KVA
- 4、 Air source working pressure: Min 0.6 MPa, Max 1.0MPa
- 5、 Drive type: DC 24V motor, ZYT520 ,3500r/min

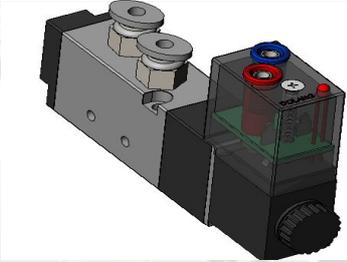
**Part List for electrical part**

No.	Item	Picture	Remark
1	Touch screen		Colored touch screen size 7 MCGS. Use for system operate and animation configuration
2	PLC module		Siemens PLC CPU-224, AC/DC/relay output. 14 digital input 10 relay output
3	Switch power		Input: AC 220V Output: DC 24V/3A Provide DC power for external equipment
4	Button module		Operate the system. Yellow, green and red button are self reset button with indicator. Another one is emergency off

5	Breaker and fuse		<p>Fuse:DZ47-2P 6A Short Circuitor : RT18-32X 3A Protection against over-load and short circuit protection</p>
6	Relay		<p>DC24V Dual open, dual close with manual switch. Control positive and negative of two DC motor</p>

Part list for mechanical part

NO.	Item	Picture	remark
1	Handling unit		<p>The handling unit is controlled by 2 DC motors to drive X and Y to ballscrew. Every axis with 2 micro switch and 3 magnetic switch for positioning</p>
2	Transport unit		<p>This unit is composed of cylinder, pallet and block. Use for the block in and out. There are 2 magnetic switch on both ends of cylinder to test position.</p>
3	Storage unit		<p>There are 3 lines, 3 rows totally 9 storage</p>

4	DC motor		<p>DC 24V motor, ZYT520, 3500r/min. to drive handling unit bypass coupling and screw rod</p>
5	Dyad		<p>0.6MPA-1MPA Use for adjust the pressure and remove the moisture from air</p>
6	Solenoid valve single control		<p>Coil voltage DC24V Use for controlling the cylinder stretch out and retract back</p>



# ANEXO 2

## FICHA TÉCNICA DE COMPONENTES



Product Description



MA Series stainless steel Mini pneumatic cylinder.MA20X100-S.With magnet

MA Series stainless steel Mini pneumatic cylinder  
Model:MA20X100-S.With magnet  
Free-shipping&Fast delivery  
Brand: NUOPAI PNEUMATC

Specification

Bore (mm)	20
Motion Pattern	Double Action or Single Action
Working Medium	Air
Fixed Type	Basic type FA / type LB / type SDB / type
Operating Voltage Range	1.0~9.0 Kg/cm <sup>2</sup>
Ensured Pressure Resistance	13.5 Kg/cm <sup>2</sup>
Operating Temperature Range	0~70C
Operating Speed Range	50~800 mm/s
Buffer Type	Buffer Spacer
Joint Pipe Bore	G1/8"

订购码 Ordering Code

**MA - U 50 × 50 - 25 - S - LB**

<p>型号 MA: 复动型 MSA: 单动型 MAD: 双轴复动型 MAJ: 双轴可调型 <b>Type</b> MA: Double Action MSA: Single Action MAD: Double-shaft Double Action MAJ: Adjustable of Double-shaft</p>	<p>后盖型式 CA: 搭尾型 CM: 圆尾型 U: 平尾型 <b>Back Cover Type</b> CA: Fishtail type CM: Rounded type U: Horizontal type</p>	<p>缸径 <b>Cylinder Bore</b></p>	<p>行程 <b>Stroke</b></p>	<p>MAJ可调行程 MAJ adjustable stroke type</p>	<p>磁石代号 S: 附磁石 空白: 不附磁石 <b>Magnet Code</b> S: attach magnet Blank: not attach magnet</p>	<p>固定型式 空白: 基本型 LB: 前后固定型 FA: 前盖固定型 SDB: 后盖摇摆式 <b>Fixed Type</b> Blank: Basic type LB: Front and back fixed type FA: Front cover fixed type SDB: Back cover swinging type</p>
---	---	------------------------------------	-----------------------------	---	--	---

规格 Specification

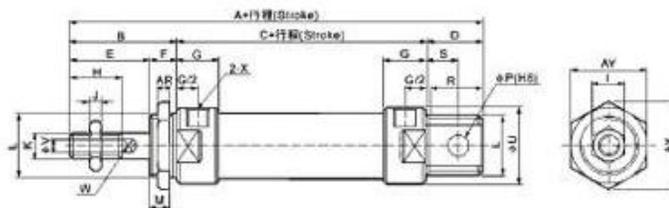
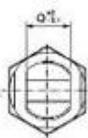
内径/Bore (mm)	16	20	25	32	40
动作形式/Motion Pattern	复动型或单动型 / Double Action or Single Action				
工作介质/Working Medium	空气 / Air				
固定型式/Fixed Type	基本型 / Basic type FA型 / type LB型 / type SDB型 / type				
使用压力范围/Operating Voltage Range	1.0~9.0 Kg/cm <sup>2</sup>				
保证耐压力/Ensured Pressure Resistance	13.5 Kg/cm <sup>2</sup>				
使用温度范围/Operating Temperature Range	0~70℃				
使用速度范围/Operating Speed Range	50~800 mm/s				
缓冲型式/Buffer Type	缓冲垫片 / Buffer Spacer				
接管口径/Joint Pipe Bore	MEX0.8		G1/8"		

气缸理论出力表 Cylinder theory output Sheet

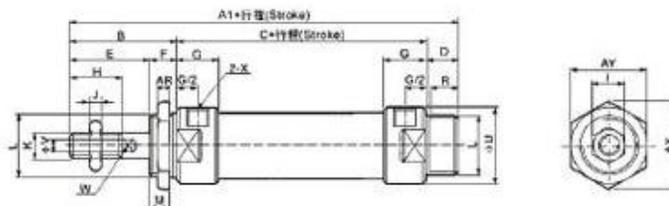
气缸内径 Cylinder Inside Diameter	16			20			25			32			40					
	活塞杆外径 External Diameter of Piston Rod			6			8			10			12			16		
动作型式 Motion Pattern	推出 Single-action	复动型 Double Action		推出 Single-action	复动型 Double Action		推出 Single-action	复动型 Double Action		推出 Single-action	复动型 Double Action		推出 Single-action	复动型 Double Action		推出 Single-action	复动型 Double Action	
		Extrusion Side	Pull Side		Extrusion Side	Pull Side		Extrusion Side	Pull Side		Extrusion Side	Pull Side		Extrusion Side	Pull Side			
受压面积 (cm <sup>2</sup> ) Compression Area	2.01	2.01	1.81	3.14	3.14	2.64	4.90	4.90	4.12	8.04	8.04	6.90	12.56	12.56	10.55			
1	-	2.01	1.81	-	3.14	2.64	-	4.90	4.12	-	8.04	6.90	-	12.56	10.55			
2	-	4.02	3.62	1.57	6.28	5.28	2.45	9.80	8.24	4.02	16.08	13.80	6.28	25.12	21.10			
3	2.01	6.03	5.43	4.71	9.42	7.92	7.35	14.70	12.36	12.06	24.12	20.70	18.84	37.68	31.65			
4	4.02	8.04	7.24	7.85	12.56	10.56	12.25	19.60	16.48	20.10	32.16	27.60	31.40	50.24	42.20			

● 外形尺寸 Overall Dimension

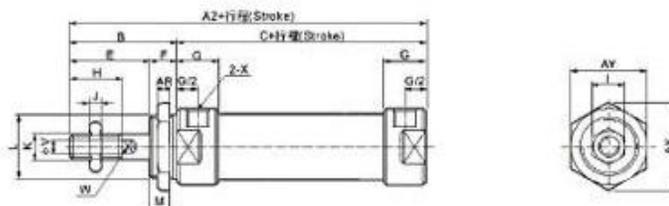
MA-CA



MA-CM



MA-U



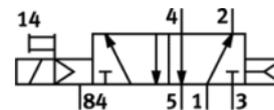
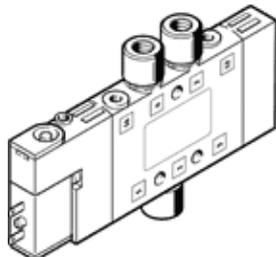
● 尺寸表 Dimension Sheet

内径 / 符号 Inside Diameter / Symbol	A	A1	A2	B	C	D	D1	E	F	G	H	I	J	K	L	M	P	Q	R	R1	S	U	V	W	X	AR	AX	AY
16	114	114	98	38	60	15	15	22	16	10	16	10	5	M6X1	M16X1.5	14	6	12	14	14	9	21	6	5	M5	6	24	27.5
20	137	128	116	40	76	21	12	28	12	16	20	12	6	M8x1.25	M22X1.5	10	8	16	19	12	12	27	8	6	G1/8	7	33	29
25	141	134	120	44	76	21	14	30	14	16	22	17	6	M10x1.25	M22X1.5	12	8	16	19	14	12	30	10	8	G1/8	7	33	29
32	147	134	120	44	76	27	14	30	14	16	22	17	6	M10x1.25	M24X2.0	12	10	16	25	14	15	35	12	10	G1/8	8	37	32
40	149	136	122	46	76	27	14	32	14	16.7	24	17	7	M12x1.25	M30X2.0	12	12	20	25	14	15	41.6	16	14	G1/8	9	47	41

## Electroválvula CPE10-M1BH-5L-M5

Número pieza: 196881  
Disponibile desde almacén  
Producto de serie

FESTO



### Hoja de datos

Caracter.	Propiedades
Función de las válvulas	5/2 monoestable
Tipo de accionamiento	eléctrico
Ancho	10 mm
Caudal nominal normal	180 l/min
Presión de funcionamiento	3 ... 8 bar
Construcción	Corredera
Tipo de reposición	muelle neumático
Tipo de protección	IP65 con conector tipo zócalo según IEC 60529
Homologación	Germanischer Lloyd c UL us - Recognized (OL)
Diámetro nominal	4 mm
Función de escape	Estrangulable
Principio de hermetización	blando
Posición de montaje	indistinto
Accionamiento manual auxiliar	con accesorios enclavables mediante pulsador
Tipo de control	prepiloto
Alimentación del aire de control	interno
Sentido del flujo	reversible
identificación de la posición de válvula	soporte del apantallamiento
Desconexión del tiempo de conmutación	20 ms
Conexión del tiempo de conmutación	16 ms
Duración de la conexión	100% con reducción de la corriente de mantenimiento
Valores característicos de las bobinas	24V DC: 1,28W
Fluctuación de tensión permisible	-15 % / +10 %
Fluido	Aire comprimido según ISO8573-1:2010 [7:4:4]
Indicación sobre los fluidos de funcionamiento y de mando	Opción de funcionamiento con lubricación (necesaria en otro modo de funcionamiento)
Clase de resistencia a la corrosión KBK	2
Temperatura del medio	-5 ... 50 °C
Temperatura ambiente	-5 ... 50 °C
Conexión eléctrica	2 contactos
Tipo de fijación	con taladro pasante
Conexión del aire de escape de pilotaje 82	M3
Conexión del aire de escape de pilotaje 84	M3
Conexión del aire de pilotaje 12	M3
Conexión del aire de pilotaje 14	M3
Conexión neumática 1	M5
Conexión neumática 2	M5
Conexión neumática 3	M7

Caracter.	Propiedades
Conexión neumática 4	M5
Conexión neumática 5	M7
Indicación sobre el material	Conforme con RoHS
Información sobre el material de las juntas	NBR
Información sobre el material del cuerpo	Fundición inyectada de aluminio



## Airtac type AFC2000-BFC4000 air source treatment pressure filter regulator

### Specifications

- 1.SMC type Air filter regulator
  - 2.Airtac air filter regulator
  - 3Media pressure air 0.85Mpa
- High pressure air filter regulator

Air filter regulator

Airtac ,SMC style air filter regulator

**FC 3010 - 03 D**

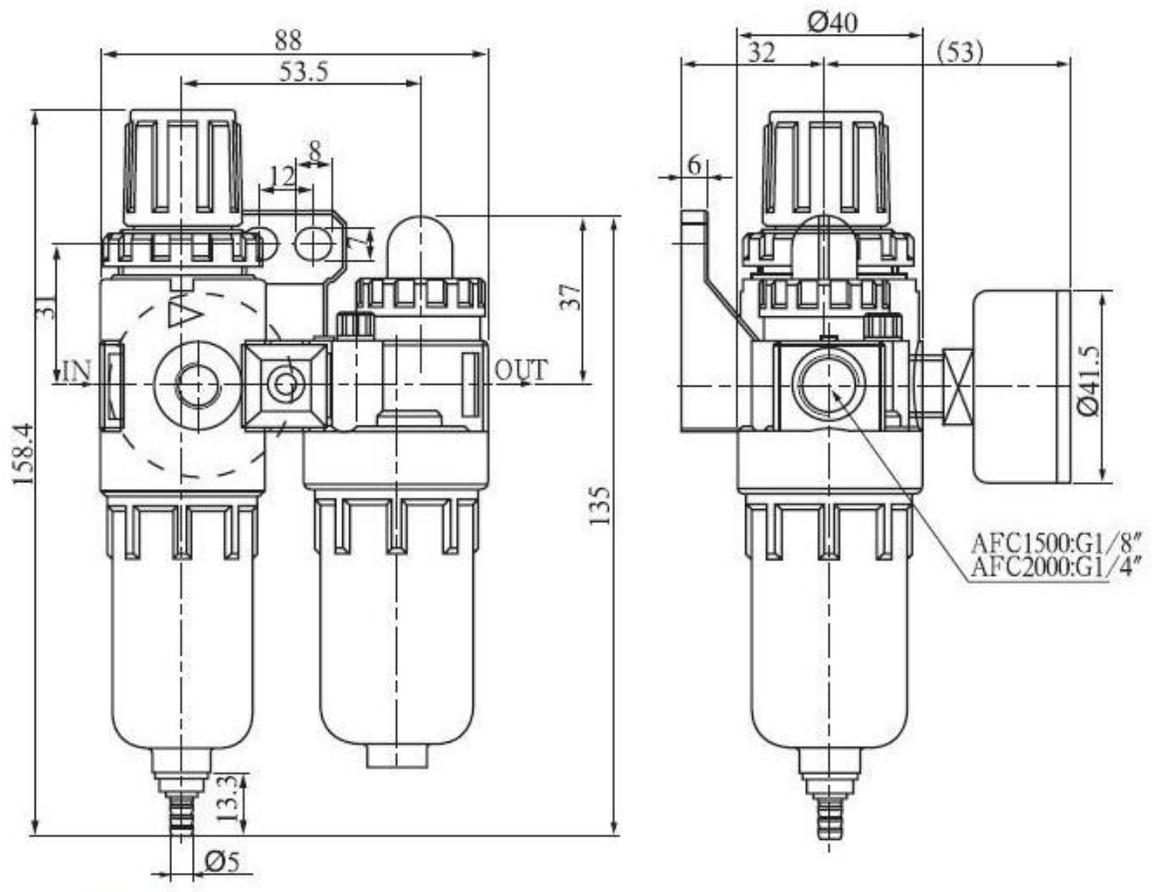
FR.L Unit Body size Port size Auto-drain filter (No mark: Manual drain)

P/N	Combination Parts		Specifications			Accessories	
	Regulator	Lubricator	Flow rate	Port size	Gauge port	Bracket	Gauge
FC2010-01	FW2000	FL2000	500 L/min	G1/8	G1/8	B210T	SAG40
FC2010-02				G1/4			
FC3010-02	FW3000	FL3000	1700 L/min	G1/4		B310T	
FC3010-03				G3/8			
FC4010-03	FW4000	FL4000	3000 L/min	G3/8	G1/4	B410T	SAG50
FC4010-04				G1/2			
FC4010-06	FW4000	FL4000	3000 L/min	G3/4		B510T	
FC5010-06				G3/4			
FC5010-10	FW5000	FL5000	4000 L/min	G1	B610T		

Note: Not only G thread, NPT, BSPT thread, etc can be selected by customers.

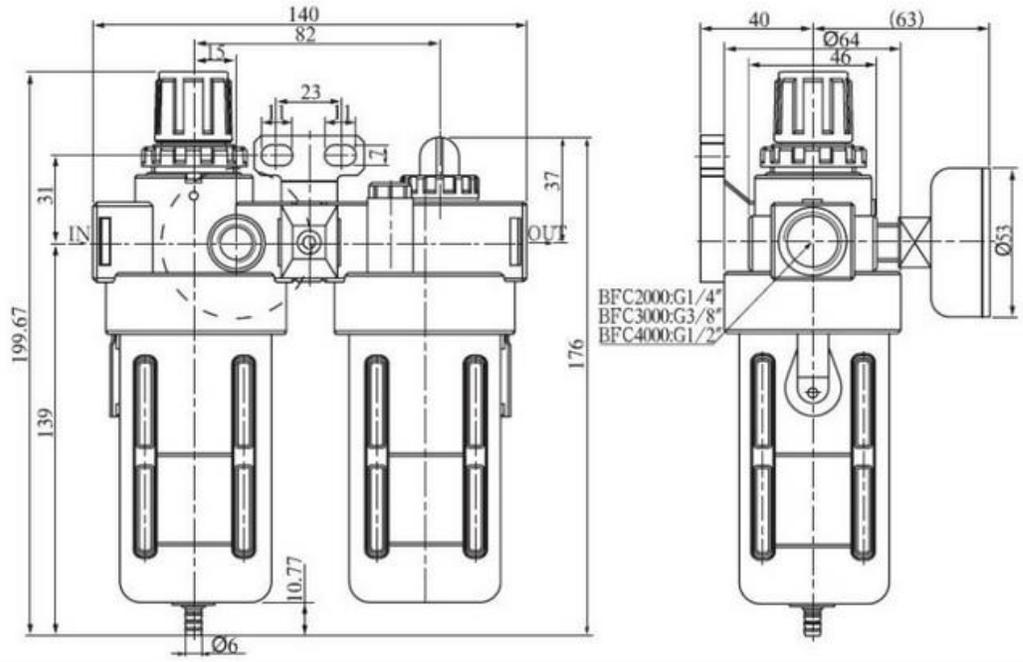
Specifications:

Media	Compressed air
Adjustable pressure range	0.05~0.85Mpa
Max. operating pressure	1.0Mpa
Proof pressure	1.5Mpa
Filtration rate	40µm , 25µm , 5µm
Ambient and fluid temperature	5~60°C
Material of body	Aluminum alloy
Material of bowl	Polycarbonate
Material of guard	FC2010: No Guard; FC3010~FC5010: Iron guard
Material of filter element	Plastic or brass
Recommended lubrication oil	Turbine No.1 oil (ISO VG32)



BFC





**Specifications**

we have been exporting to Middle East, East Asia and SouthAmerican markets in large quantities. Quality is reliable and at fair



<b>Contact Ratings</b>	
Contact Ratings	2Z,3Z 4Z
Contact Resistance	50mΩ (1A 6VDC)
Contact capacity	10A/220VAC 30VDC 5A/220VAC 30VDC
<b>Specification</b>	
Insulation Resistance	500MΩ, 500VDC BCC 1500VAC 1minute BOC 1000VAC 1minute CCC 1500VAC 1minute
Dielectric Strength	

Operate Time	25ms /25ms						
Terminal Type	PCB and Socket						
<b>Coil Ratings</b>							
Nominal Coil Power	0.9W/1.2VA						
<b>Coil Versions</b>							
Nominal Voltage VDC	Pull-in Voltage VDC	Release Voltage VDC	Coil Resistance $\Omega$ : $\pm$ 10%	Nominal Voltage VAC	Pull-in Voltage VDC	Release Voltage VDC	Coil Resistance $\Omega$ : $\pm$ 10%
5	4.0	0.50	30	6	4.80	1.80	12
6	4.8	0.60	40	12	9.60	3.60	42
12	9.6	1.20	160	24	19.2	7.20	168
24	19.2	2.40	640	48	38.4	14.4	675
48	38.4	4.80	2560	120	96.0	36.0	3500
110	88.0	11.0	12100	220/240	176.0	66.0	14000/16500



## DZ47LE Series Current Circuit Breakers

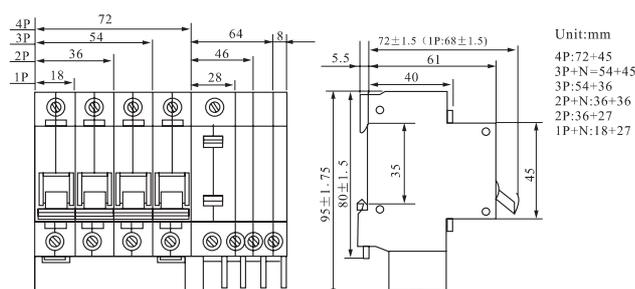
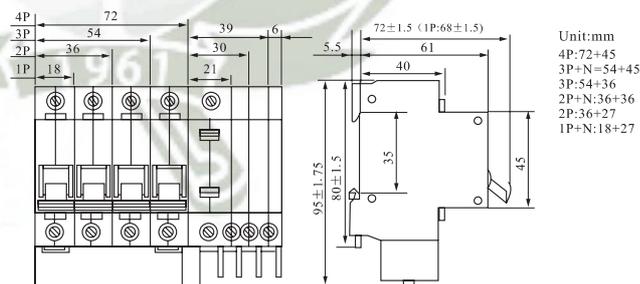
### Scope of Application

DZ47LE series earth leakage circuit breakers are mainly applied to the AC 50Hz/60Hz circuit rated 230V/400V and 63A or below. They have the electric leakage (electric shock), overload, short circuit and so on protection functions. The overvoltage protection function can also be provided upon requirements. They are mainly used to protect the building illumination and power distribution system.

### Main Technical Parameter

Rated current A	P	With neutral line	In A	V	$I\Delta n$ mA	$I\Delta no$ mA	$I\Delta n$ breaking time S	$I\Delta m$ A	overload instantaneous release
32	1	N		230					C D
	2								
	2	N	6 10 16						
	3		20 25 32	400	30	15			
60	3	N							C D
	4								
	3		40 50 60	400	50	25			
	4				100	50	<0.1	630	
60	1	N		230					C D
	2								
	2	N	40 50 60	400	300	150			
	3								

### Overall and Installation Dimensions





75W Single Output with PFC Function

SP-75



■ Features :

- Universal AC input / Full range
- Built-in active PFC function, PF>0.93
- Protections: Short circuit / Overload / Over voltage
- Built-in constant current limiting circuit
- Low profile: 33mm thickness
- LED indicator for power on
- 100% full load burn-in test
- Fixed switching frequency at PFC:67KHz PWM:134KHz
- 3 years warranty



SPECIFICATION

MODEL	SP-75-3.3	SP-75-5	SP-75-7.5	SP-75-12	SP-75-13.5	SP-75-15	SP-75-24	SP-75-27	SP-75-48	
OUTPUT	DC VOLTAGE	3.3V	5V	7.5V	12V	13.5V	15V	24V	27V	48V
	RATED CURRENT	15A	15A	10A	6.3A	5.6A	5A	3.2A	2.8A	1.6A
	CURRENT RANGE	0 ~ 15A	0 ~ 15A	0 ~ 10A	0 ~ 6.3A	0 ~ 5.6A	0 ~ 5A	0 ~ 3.2A	0 ~ 2.8A	0 ~ 1.6A
	RATED POWER	49.5W	75W	75W	75.6W	75.6W	75W	76.8W	75.6W	76.8W
	RIPPLE & NOISE (max.) Note.2	80mVp-p	80mVp-p	80mVp-p	80mVp-p	80mVp-p	80mVp-p	100mVp-p	100mVp-p	100mVp-p
	VOLTAGE ADJ. RANGE	3.14 ~ 3.63V	4.75 ~ 5.5V	7.13 ~ 8.25V	11.4 ~ 13.2V	12.8 ~ 14.9V	14.3 ~ 16.5V	22.8 ~ 26.4V	25.7 ~ 29.7V	45.6 ~ 52.8V
	VOLTAGE TOLERANCE Note.3	±2.0%	±2.0%	±2.0%	±2.0%	±2.0%	±2.0%	±1.0%	±1.0%	±1.0%
	LINE REGULATION	±0.5%	±0.5%	±0.5%	±0.5%	±0.5%	±0.5%	±0.5%	±0.5%	±0.5%
	LOAD REGULATION	±1.0%	±1.0%	±1.0%	±0.5%	±0.5%	±0.5%	±0.5%	±0.5%	±0.5%
	SETUP, RISE TIME	800ms, 60ms at full load								
HOLD UP TIME (Typ.)	36ms at full load									
INPUT	VOLTAGE RANGE Note.5	85 ~ 264VAC		120 ~ 370VDC						
	FREQUENCY RANGE	47 ~ 63Hz								
	POWER FACTOR (Typ.)	PF>0.93/230VAC		0.96/115VAC at full load						
	EFFICIENCY (Typ.)	70%	76%	79%	81%	82%	82%	83%	83%	83%
	AC CURRENT (Typ.)	1.3A/115VAC		0.7A/230VAC						
	INRUSH CURRENT (Typ.)	COLD START 45A/230VAC								
	LEAKAGE CURRENT	<2mA / 240VAC								
PROTECTION	OVERLOAD	105 ~ 150% rated output power Protection type : Constant current limiting, recovers automatically after fault condition is removed								
	OVER VOLTAGE	3.8 ~ 4.46V	5.75 ~ 6.75V	8.63 ~ 10.13V	13.8 ~ 16.2V	15.53 ~ 18.23V	17.25 ~ 20.25V	27.6 ~ 32.4V	31.05 ~ 36.45V	55.2 ~ 64.8V
FUNCTION	REMOTE CONTROL(OPTION)	CN3:4 ~ 10VDC POWER OFF, <0 ~ 0.8VDC POWER ON								
ENVIRONMENT	WORKING TEMP.	-10 ~ +60°C (Refer to output load derating curve)								
	WORKING HUMIDITY	20 ~ 90% RH non-condensing								
	STORAGE TEMP., HUMIDITY	-20 ~ +85°C, 10 ~ 95% RH								
	TEMP. COEFFICIENT	±0.05%/°C (0 ~ 50°C)								
	VIBRATION	10 ~ 500Hz, 2G 10min./1cycle, 60min. each along X, Y, Z axes								
SAFETY & EMC (Note 4)	SAFETY STANDARDS	UL60950-1, TUV EN60950-1 approved								
	WITHSTAND VOLTAGE	I/P-O/P:3KVAC		I/P-FG:2KVAC		O/P-FG:0.5KVAC				
	ISOLATION RESISTANCE	I/P-O/P, I/P-FG, O/P-FG:100M Ohms / 500VDC / 25°C / 70% RH								
	EMC EMISSION	Compliance to EN55022 (CISPR22) Class B, EN61000-3-2,-3								
	EMC IMMUNITY	Compliance to EN61000-4-2,3,4,5,6,8,11; ENV50204, EN55024, light industry level, criteria A								
OTHERS	MTBF	208.8K hrs min. MIL-HDBK-217F (25°C)								
	DIMENSION	179*97*33mm (L*W*H)								
	PACKING	0.58Kg; 20pcs/12Kg/0.64CUFT								
NOTE	<p>1. All parameters NOT specially mentioned are measured at 230VAC input, rated load and 25°C of ambient temperature.</p> <p>2. Ripple &amp; noise are measured at 20MHz of bandwidth by using a 12" twisted pair-wire terminated with a 0.1uf &amp; 47uf parallel capacitor.</p> <p>3. Tolerance : includes set up tolerance, line regulation and load regulation.</p> <p>4. The power supply is considered a component which will be installed into a final equipment. The final equipment must be re-confirmed that it still meets EMC directives. For guidance on how to perform these EMC tests, please refer to "EMI testing of component power supplies." (as available on <a href="http://www.meanwell.com">http://www.meanwell.com</a>)</p> <p>5. Derating may be needed under low input voltages. Please check the derating curve for more details.</p>									

hoja de datos del producto

6ES7223-1PH22-0XA0



SIMATIC S7-200, MODULO E/S DIG. EM 223,  
SOLO PARA CPU S7-22X, 8 ED 24 V DC,  
TIPO P/M, 8 SD RELES, 2A/CANAL

<b>Tensión de alimentación</b>	
<b>Tensión de carga L+</b>	
Valor nominal (DC)	24 V
Rango admisible, límite inferior (DC)	5 V
Rango admisible, límite superior (DC)	30 V
<b>Tensión de carga L1</b>	
Valor nominal (AC)	230 V ; 24 a 230 V AC
Rango admisible, límite inferior (AC)	5 V
Rango admisible, límite superior (AC)	250 V
<b>Intensidad de entrada</b>	
De bus de fondo 5 V DC, máx.	80 mA
De corriente de bobinas, máx.	9 mA ; por salida con señal "1"
De alimentación de sensores o de alimentación externa (24 V DC), máx.	72 mA
<b>Pérdidas</b>	
Pérdidas, típ.	3 W
<b>Entradas digitales</b>	
Cantidad/entradas binarias	8
<b>Tensión de entrada</b>	

<b>Tipo de tensión de entrada</b>	AC/DC
<b>Valor nominal, DC</b>	24 V
<b>para señal "0"</b>	0 a 5 V
<b>para señal "1"</b>	15 a 30 V DC
<b>Intensidad de entrada</b>	
<b>para señal "1", típ.</b>	4 mA
<b>Retardo de entrada (a tensión nominal de entrada)</b>	
<b>para entradas estándar</b>	
<b>en transición "0" a "1", máx.</b>	4,5 ms
<b>Salidas digitales</b>	
<b>Número/salidas binarias</b>	8 ; Relé
<b>Funcionalidad/resistencia a cortocircuitos</b>	No ; a prever externamente
<b>Tensión de salida</b>	
<b>para señal "0", máx.</b>	0,1 V ; con carga de 10 kOhm
<b>para señal "1", mín.</b>	L+/L1
<b>Intensidad de salida</b>	
<b>para señal "1" valor nominal</b>	2000 mA
<b>Intensidad suma de las salidas (por grupo)</b>	
<b>Todas las posiciones de montaje</b>	
<b>Intensidad máxima por conductor/grupo</b>	8 A
<b>Salidas de relé</b>	
<b>Número de ciclos de maniobra, máx.</b>	10000000 ; mecánicos: 10 millones, con tensión nominal de carga: 100000
<b>Poder de corte de los contactos</b>	
<b>con carga inductiva, máx.</b>	0,75 A ; por salida
<b>con carga tipo lámpara, máx.</b>	200 W ; 30 W con DC, 200 W con AC
<b>Poder de corte/contactos/con carga resistiva/máximo</b>	0,75 A ; por salida
<b>Longitud del cable</b>	
<b>Longitud del cable apantallado, máx.</b>	500 m
<b>Longitud de cable no apantallado, máx.</b>	150 m
<b>Sensor</b>	
<b>Sensores compatibles</b>	
<b>Sensor a 2 hilos</b>	Sí
<b>Intensidad permitida en reposo (sensor a 2 hilos), máx.</b>	1 mA
<b>Aislamiento galvánico</b>	
<b>Aislamiento galvánico módulos de E digitales</b>	
<b>Aislamiento galvánico módulos de E digitales</b>	Sí ; Optoacoplador
<b>entre los canales, en grupos de</b>	4

<b>Aislamiento galvánico módulos de S digitales</b>	
<b>Aislamiento galvánico módulos de S digitales</b>	Sí ; Relé
<b>entre los canales, en grupos de</b>	4
<b>Aislamiento</b>	
<b>Aislamiento ensayado con</b>	500 V AC
<b>Sistema de conexión</b>	
<b>Bornes de E/S enchufables</b>	Sí
<b>Dimensiones</b>	
<b>Anchura</b>	71,2 mm
<b>Altura</b>	80 mm
<b>Profundidad</b>	62 mm
<b>Peso</b>	
<b>Peso, aprox.</b>	300 g
Última actualización	15-jun-2013



## A.4 Datos técnicos de la CPU 224

Tabla A-4 Datos técnicos de la CPU 224 DC/DC/DC y de la CPU 224 AC/DC/relé

Descripción N° de referencia	CPU 224 DC/DC/DC 6ES7 214-1AD20-0XB0	CPU 224 AC/DC/relé 6ES7 214-1BD20-0XB0
<b>Tamaño físico</b>		
Dimensiones (l x a x p)	120.5 mm x 80 mm x 62 mm	120.5 mm x 80 mm x 62 mm
Peso	360 g	410 g
Pérdida de corriente (disipación)	8 W	9 W
<b>Características de la CPU</b>		
Entradas digitales integradas	14 entradas	14 entradas
Salidas digitales integradas	10 salidas	10 salidas
Contadores rápidos (valor de 32 bits)		
Total	6 contadores rápidos	6 contadores rápidos
Contadores de fase simple	6, con una frecuencia de reloj de 20 kHz c/u	6, con una frecuencia de reloj de 20 kHz c/u
Contadores de dos fases		
Salidas de impulsos	4, con una frecuencia de reloj de 20 kHz c/u	4, con una frecuencia de reloj de 20 kHz c/u
Potenciómetros analógicos		
Interrupciones temporizadas	2 a una frecuencia de impulsos de 20 kHz	2 a una frecuencia de impulsos de 20 kHz
Interrupciones de flanco	2 con resolución de 8 bits	2 con resolución de 8 bits
	2 con resolución de 1 ms	2 con resolución de 1 ms
Tiempos de filtración de entradas		
Captura de impulsos	4 flancos positivos y/o 4 flancos negativos	4 flancos positivos y/o 4 flancos negativos
Reloj de tiempo real (precisión del reloj)	7 márgenes de 0,2 ms a 12,8 ms	7 márgenes de 0,2 ms a 12,8 ms
	14 entradas de captura de impulsos	14 entradas de captura de impulsos
	2 minutos por mes a 25° C	2 minutos per month at 25° C
	7 minutos por mes 0° C a 55° C	7 minutes per month at 0° C a 55° C
Tamaño del programa (almacenado permanentemente)	4096 palabras	4096 palabras
Tamaño del bloque de datos (almacenado permanentemente):	2560 palabras	2560 palabras
Almacenamiento permanente	2560 palabras	2560 palabras
Respaldo por condensador de alto rendimiento o pila	2560 palabras	2560 palabras
	7 módulos	7 módulos
N° de módulos de ampliación	256 E/S	256 E/S
E/S de ampliación digitales (máx.)	16 entradas y 16 salidas	16 entradas y 16 salidas
E/S analógicas (máx.)		
Marcas internas	256 bits	256 bits
Almacenamiento permanente al apagar	112 bits	112 bits
Respaldo por condensador de alto rendimiento o pila	256 bits	256 bits
Temporizadores (total)	256 temporizadores	256 temporizadores
Respaldo por condensador de alto rendimiento o pila	64 temporizadores	64 temporizadores
1 ms	4 temporizadores	4 temporizadores
10 ms	16 temporizadores	16 temporizadores
100 ms	236 temporizadores	236 temporizadores

Tabla A-4 Datos técnicos de la CPU 224 DC/DC/DC y de la CPU 224 AC/DC/relé

Descripción Nº de referencia	CPU 224 DC/DC/DC 6ES7 214-1AD20-0XB0	CPU 224 AC/DC/relé 6ES7 214-1BD20-0XB0
Contadores (total)	256 contadores	256 contadores
Respaldo por condensador de alto rendimiento o pila	256 contadores 0,37 µs por operación	256 contadores 0,37 µs por operación
Velocidad de ejecución booleana	34 µs por operación	34 µs por operación
Velocidad de ejecución de Transferir palabra	50 µs a 64 µs por operación	50 µs a 64 µs por operación
Velocidad de ejecución de temporizadores/contadores	46 µs por operación	46 µs por operación
Velocidad de ejecución de aritmética de precisión simple	100 µs a 400 µs por operación	100 µs a 400 µs por operación
Velocidad de ejecución de aritmética en coma flotante	típ. 190 h, mín. 120 h a 40° C	típ. 190 h, mín. 120 h a 40° C
Tiempo de respaldo por el condensador de alto rendimiento		
<b>Comunicación integrada</b>		
Nº de puertos	1 puerto	1 puerto
Interface eléctrico	RS-485	RS-485
Aislamiento (señal externa a circuito lógico)	Sin aislamiento	Sin aislamiento
Velocidades de transferencia PPI/MPI	9,6, 19,2 y 187,5 kbit/s	9,6, 19,2 y 187,5 kbit/s
Velocidades de transferencia Freeport	0,3, 0,6, 1,2, 2,4, 4,8, 9,6, 19,2 y 38,4 kbit/s	0,3, 0,6, 1,2, 2,4, 4,8, 9,6, 19,2 y 38,4 kbit/s
Longitud máx. del cable por segmento hasta 38,4 kbit/s	1200 m	1200 m
187,5 kbit/s	1000 m	1000 m
Nº máximo de estaciones		
Por segmento	32 estaciones	32 estaciones
Por red	126 estaciones	126 estaciones
Nº máximo de maestros	32 maestros	32 maestros
Modo maestro PPI (NETR/NETW)	Sí	Sí
Enlaces MPI	4 en total; 2 reservados: 1 para PG y 1 OP	1 OP
<b>Cartuchos opcionales</b>		
Cartucho de memoria (almacenamiento permanente)	Programa, datos y configuración típ. 200 días	Programa, datos y configuración típ. 200 días
Cartucho de pila (tiempo de respaldo de datos)		
<b>Alimentación</b>		
Tensión de línea (margen admisible)	DC 20,4 a 28,8 V	AC 85 a 264 V 47 a 63 Hz
Corriente de entrada (sólo CPU) / carga máx.	120/900 mA a DC 24 V	35/100 mA a AC 240 V 35/220 mA a AC 120 V
Extra-corriente de serie (máx.)	10 A a DC 28,8 V	20 A a AC 264 V
Aislamiento (corriente de entrada a lógica)	Sin aislamiento	AC 1500 V
Tiempo de retardo (desde la pérdida de corriente de entrada)	mín. 10 ms de DC 24 V	80 ms de AC 240 V, 20 ms de 120 VAC
Fusible interno (no reemplazable por el usuario)	2 A, 250 V, de acción lenta	2 A, 250 V, de acción lenta
<b>+5 alimentación para módulos de ampliación (máx.)</b>	660 mA	660 mA
<b>Alimentación para sensores DC 24 V</b>		
Margen de tensión	DC 15,4 a 28,8 V	DC 20,4 a 28,8 V
Corriente máxima	280 mA	280 mA
Rizado corriente parásita	Igual que línea de entrada	Menos de 1 V pico a pico (máx.)
Corriente límite	600 mA	600 mA
Aislamiento (alimentación de sensores a circuito lógico)	Sin aislamiento	Sin aislamiento

Tabla A-4 Datos técnicos de la CPU 224 DC/DC/DC y de la CPU 224 AC/DC/relé

Descripción Nº de referencia	CPU 224 DC/DC/DC 6ES7 214-1AD20-0XB0	CPU 224 AC/DC/relé 6ES7 214-1BD20-0XB0
<b>Características de las entradas</b>		
Nº de entradas integradas	14 entradas	14 entradas
Tipo de entrada	Sumidero de corriente/fuente (tipo 1 IEC)	Sumidero de corriente/fuente (tipo 1 IEC)
<b>Tensión de entrada</b>		
Tensión máx. continua admisible	DC 30 V	DC 30 V
Sobretensión transitoria	DC 35 V, 0,5 s	DC 35 V, 0,5 s
Valor nominal	DC 24 V a 4 mA, nominal	DC 24 V a 4 mA, nominal
Señal 1 lógica (mín.)	mín. DC 15 V a 2,5 mA	mín. DC 15 V a 2,5 mA
Señal 0 lógica (máx.)	máx. DC 5 V a 1 mA	máx. DC 5 V a 1 mA
<b>Aislamiento (campo a circuito lógico)</b>		
Separación galvánica	AC 500 V, 1 minuto	AC 500 V, 1 minuto
Grupos de aislamiento de	8 y 6 entradas	8 y 6 entradas
<b>Tiempos de retardo de las entradas</b>		
Entradas filtradas y entradas de interrupción	0,2 a 12,8 ms, seleccionable por el usuario	0,2 a 12,8 ms, seleccionable por el usuario
<b>Entradas de reloj de los contadores rápidos</b>		
Fase simple		
DC 30 V Nivel 1 lógico = DC 15 V a	20 kHz	20 kHz
DC 26 V Nivel 1 lógico = DC 15 V a	30 kHz	30 kHz
Contadores A/B Nivel 1 lógico = DC 15 V a	10 kHz	10 kHz
DC 30 V Nivel 1 lógico = DC 15 V a	20 kHz	20 kHz
DC 26 V Nivel 1 lógico = DC 15 V a		
<b>Conexión de sensor de proximidad de 2 hilos (Bero)</b>		
Corriente de fuga admisible	máx. 1 mA	máx. 1 mA
<b>Longitud del cable</b>		
No apantallado (no HSC)	300 m	300 m
Apantallado	500 m	50 m
Entradas HSC, apantalladas	50 m	50 m
<b>Nº de entradas ON simultáneamente</b>		
40 ° C	14	14
55 ° C	14	14
<b>Características de las salidas</b>		
Nº de salidas integradas	10 salidas	10 salidas
Tipo de salida	Estado sólido-MOSFET	Relé, contacto de baja potencia
<b>Tensión de salida</b>		
Margen admisible	DC 20,4 a 28,8 V	DC 5 a 30 V ó AC 5 a 250 V
Valor nominal	DC 24 V	–
Señal 1 lógica a corriente máxima	mín. DC 20 V	–
Señal 0 lógica a 10 K Ω de carga	máx. DC 0,1 V	–

Tabla A-4 Datos técnicos de la CPU 224 DC/DC/DC y de la CPU 224 AC/DC/relé

Descripción Nº de referencia	CPU 224 DC/DC/DC 6ES7 214-1AD20-0XB0	CPU 224 AC/DC/relé 6ES7 214-1BD20-0XB0
<b>Corriente de salida</b>		
Señal 1 lógica	0,75 A	2.00 A
Nº de grupos de salidas	2	3
Nº de salidas ON (máx.)	10	10
Por grupo – montaje horizontal (máx.)	5	4/3/3
Por grupo – montaje vertical (máx.)	5	4/3/3
Corriente máx. por común/grupo	3,75 A	8 A
Carga LEDs	5 W	30 W DC/200 W AC
Resistencia estado ON (resistencia contactos)	0,3Ω	0,002 Ω, máx. si son nuevas
Corriente de derivación por salida	máx. 10 µA	–
Sobrecorriente momentánea	máx. 8 A, 100 ms	7A al estar cerrados los contactos
Protección contra sobrecargas	no	no
<b>Aislamiento (campo a circuito lógico)</b>		
Separación galvánica	AC 500 V, 1 minuto	–
Resistencia de aislamiento	–	100 M Ω, mín. si son nuevas
Aislamiento bobina a contacto	–	AC 500 V, 1 minuto
Aislamiento entre contactos abiertos	–	AC 750 V, 1 minuto
En grupos de	5 salidas	4 salidas/3 salidas/3 salidas
<b>Carga inductiva, apriete</b>		
Repetición	1 W, en todos los canales	–
disipación de energía	L+ menos 48V	–
< 0.5 L <sup>2</sup> x frecuencia de conmutación		
Límites tensión de bloqueo		
<b>Retardo de las salidas</b>		
OFF a ON (Q0.0 y Q0.1)	máx. 2 µs	–
ON a OFF (Q0.0 y Q0.1)	máx. 10 µs	–
OFF a ON (Q0.2 hasta Q1.1)	máx. 15 µs	–
ON a OFF (Q0.2 hasta Q1.1)	máx. 100 µs	–
<b>Frecuencia de conmutación (salida de impulsos)</b>		
Q0.0 y I0.1	máx. 20 kHz	máx. 1 Hz
<b>Relé</b>		
Retardo de conmutación	–	máx. 10 ms
Vida útil mecánica (sin carga)	–	10.000.000 ciclos abiertos/cerrados
Vida útil contactos a carga nominal	–	100.000 ciclos abiertos/cerrados
<b>Longitud del cable</b>		
No apantallado	150 m	150 m
Apantallado	500 m	500 m

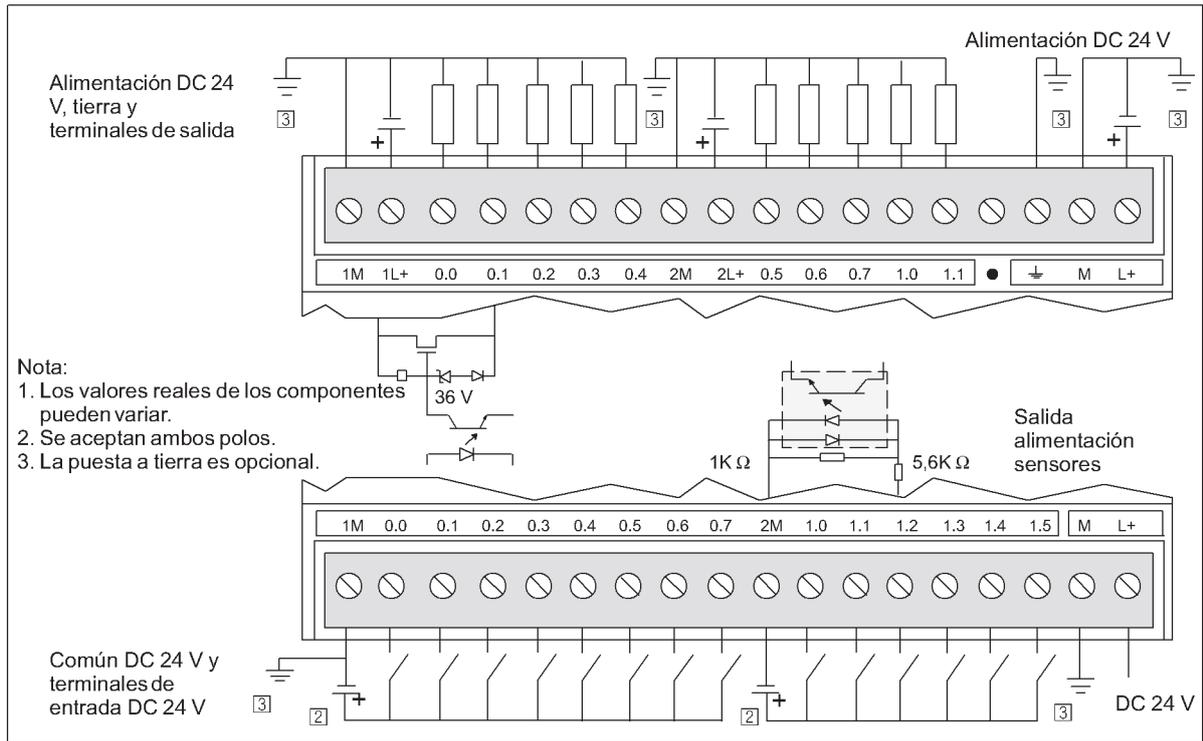


Figura A-6 Identificación de terminales de conexión para la CPU 224 DC/DC/DC

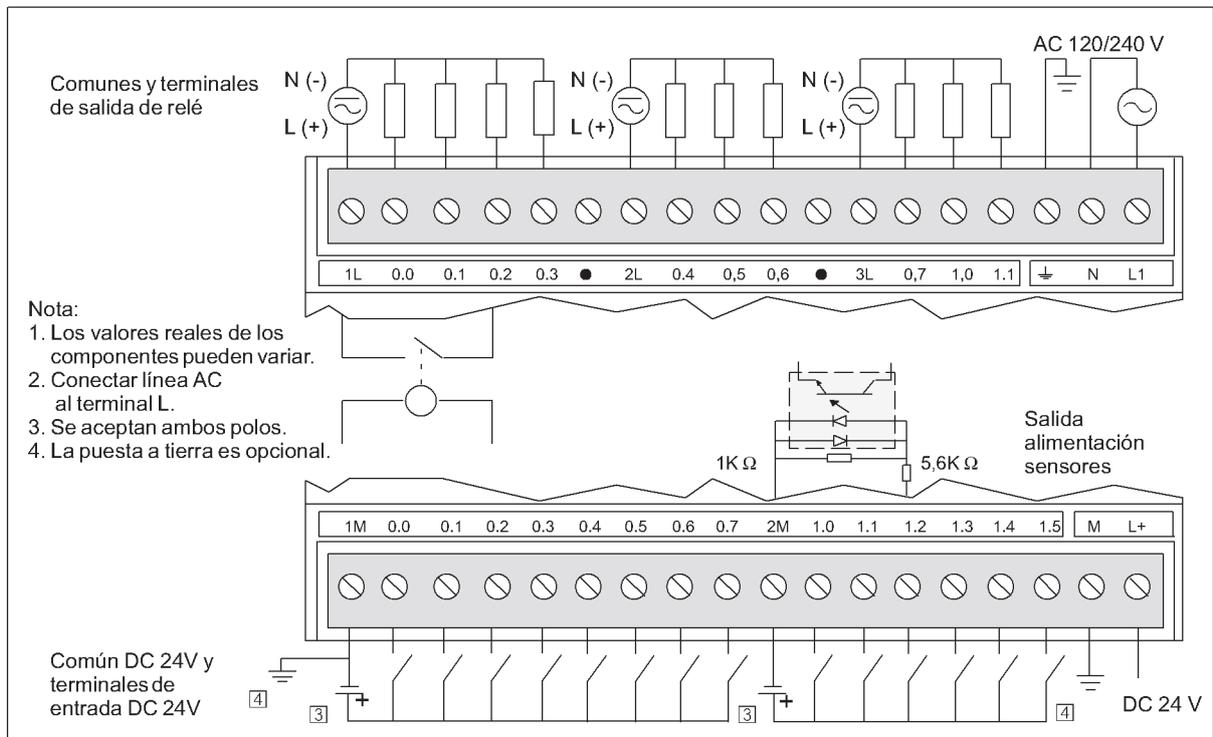


Figura A-7 Identificación de terminales de conexión para la CPU 224 AC/DC/relé

## A.1 Datos técnicos generales

### Homologaciones nacionales e internacionales

Las características de funcionamiento y las pruebas realizadas con los productos de la gama S7-200 se basan en las homologaciones nacionales e internacionales que se indican a continuación. En la tabla A-1 se indica la conformidad específica con dichas homologaciones.

- Underwriters Laboratories, Inc.: UL 508 registrado (Industrial Control Equipment)
- Canadian Standards Association: CSA C22.2 n° 142, certificado (Process Control Equipment)
- Factory Mutual Research: FM clase I, categoría 2, grupo A, B, C, y D Hazardous Locations, T4A
- VDE 0160: equipos electrónicos de uso en instalaciones de corriente eléctrica
- Directiva de Baja Tensión de la Comunidad Europea 73/23/CEE (EN 61131-2): Autómatas programables – requisitos del equipo
- Directiva EMC de la Comunidad Europea (CE) 89/336/CEE

Normas de emisión electromagnética:

EN 50081-1: entornos residenciales, comerciales y semi-industriales

EN 50081-2: entornos industriales

Normas de inmunidad electromagnética:

EN 50082-2: entornos industriales

## Datos técnicos

La tabla A-1 muestra los datos técnicos de las CPUs S7-200 y sus módulos de ampliación.

Tabla A-1 Datos técnicos de la gama S7-200

<b>Condiciones ambientales – Transporte y almacenamiento</b>	
IEC 68-2-2, ensayo Bb, calor seco y IEC 68-2-1, ensayo Ab, Frío	-40° C a +70° C
IEC 68-2-30, ensayo Dd, calor húmedo	25° C a 55° C, 95% humedad
IEC 68-2-31, vuelco	100 mm, 4 gotas, desembalado
IEC 68-2-32, caída libre	1m, 5 veces, embalado para embarque
<b>Condiciones ambientales – Funcionamiento</b>	
Condiciones ambientales (aire de entrada 25 mm debajo de la unidad)	0° C a 55° C montaje horizontal 0° C a 45° C montaje vertical 95% humedad no condensante
IEC 68-2-14 Ensayo Nb	5° C a 55° C, 3° C/minuto
IEC 68-2-27 Choque mecánico	15 G, 11 ms impulso, 6 choques en c/u de 3 ejes
IEC 68-2-6 Vibración sinusoidal	0,30 mm pico a pico 10 a 57 Hz; 2G montaje en armario eléctrico, 1G montaje en perfil soporte, 57 a 150 Hz; 10 barridos por eje, 1 octava/minuto
EN 60529, IP20 Protección mecánica	Protege los dedos contra el contacto con alto voltaje, según pruebas realizadas con sondas estándar. Se requiere protección externa contra polvo, impurezas, agua y objetos extraños de menos de 12,5 mm de diámetro.
<b>Compatibilidad electromagnética — Inmunidad<sup>1</sup> según EN50082-2<sup>1</sup></b>	
EN 61000-4-2 (IEC 801-2) Descargas electrostáticas	8 kV descarga en el aire a todas las superficies y al interface de comunicación
EN 50140 (IEC 801-3) Campos electromagnéticos radiados	80 MHz a 1 GHz 10 V/m, 80% modulación con señal de 1 kHz
EN 50141 Perturbaciones conducidas	0,15 MHz a 80 GHz 10 V/m, 80% modulación con señal de 1 kHz
EN 50204 Inmunidad a radioteléfonos digitales	900 MHz ± 5 MHz, 10 V/m, 50% ciclo de trabajo, frecuencia de repetición 200 Hz
EN 61000-4-4 (IEC 801-4) Transitorios eléctricos rápidos	2 kV, 5 kHz con red de unión a la alimentación AC y DC 2 kV, 5 kHz con abrazadera de unión a las E/S digitales y a la comunicación
EN 61000-4-5 (IEC 801-5) Inmunidad a ondas de choque	2 kV asimétrico, 1 kV simétrico 5 impulsos positivos/5 impulsos negativos 0°, +90°, -90° decalaje de fase (para los circuitos de DC 24 V se necesita una protección externa contra sobrecargas)
VDE 0160 Sobrevoltaje no periódico	a AC 85 V línea, 90° decalaje de fase, aplicar cresta de 390 V, impulso de 1,3 ms a AC 180 V línea, 90° decalaje de fase, aplicar cresta de 750 V, impulso de 1,3 ms

Tabla A-1 Datos técnicos de la gama S7-200

<b>Compatibilidad electromagnética — Emisiones conducidas y radiadas según EN50081 -1 2 y -2</b>	
EN 55011, clase A, grupo 1, conducida <sup>1</sup> 0,15 a 0,5 MHz 0,15 a 5 MHz 5 MHz a 30 MHz	< 79 dB (µV) casi cresta; < 66 dB (µV) promedio < 73 dB (µV) casi cresta; < 60 dB (µV) promedio < 73 dB (µV) casi cresta; < 60 dB (µV) promedio
EN 55011, clase A, grupo 1, radiada <sup>1</sup> 30 MHz a 230 kHz 230 MHz a 1 GHz	30 dB (µV/m) casi cresta; medido a 30 m 37 dB (µV/m) casi cresta; medido a 30 m
EN 55011, clase B, grupo 1, conducida <sup>2</sup> 0,15 a 0,5 MHz 0,5 MHz a 5 MHz 5 MHz a 30 MHz	< 66 dB (µV) decremento casi cresta con frecuencia logarítmica a 56 dB (µV) < 56 dB (µV) decremento promedio con frecuencia logarítmica a 46 dB (µV) < 56 dB (µV) casi cresta; < 46 dB (µV) promedio < 60 dB (µV) casi cresta; < 50 dB (µV) promedio
EN 55011, clase B, grupo 1, radiada <sup>2</sup> 30 MHz a 230 kHz 230 MHz a 1 GHz	30 dB (µV/m) casi cresta; medido a 10 m 37 dB (µV/m) casi cresta; medido a 10 m
<b>Prueba de aislamiento a hipervoltajes</b>	
24 V/5 V circuitos nominales 115/230 V circuitos a tierra 115/230 V circuitos hasta 115/230 V circuitos 230 V circuitos hasta 24 V/5V circuitos 115 V circuitos hasta 24 V/5V circuitos	AC 500 V (límites de aislamiento óptico) AC 1,500 V AC 1,500 V AC 1,500 V AC 1,500 V

- 1 La unidad deberá montarse en un soporte metálico puesto a tierra. El S7-200 deberá ponerse a tierra directamente a través del soporte metálico. Los cables se deberán conducir a lo largo de los soportes metálicos.
- 2 La unidad deberá montarse en una caja metálica puesta a tierra. La línea de alimentación de corriente alterna se deberá equipar con un filtro SIEMENS B84115-E-A30 o similar, teniendo el cable una longitud máxima de 25 cm entre los filtros y el S7-200. El cableado de la alimentación DC 24 V y de la alimentación de sensores se deberá apantallar.

### Vida útil de los relés

La figura A-1 muestra los datos típicos de rendimiento de los relés proporcionados por el comercio especializado. El rendimiento real puede variar dependiendo de la aplicación.

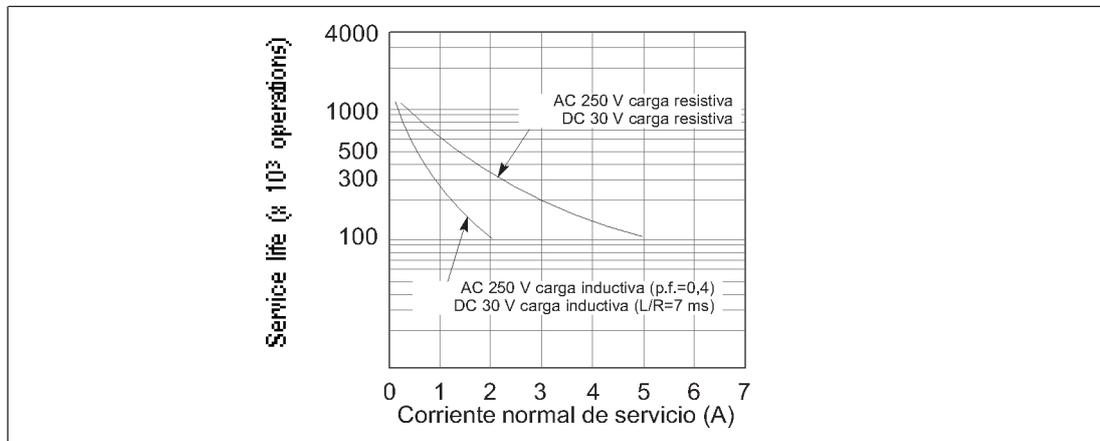


Figura A-1 Vida útil de los relés



# ANEXO 3

## PLANOS ELÉCTRICO - NEUMÁTICO



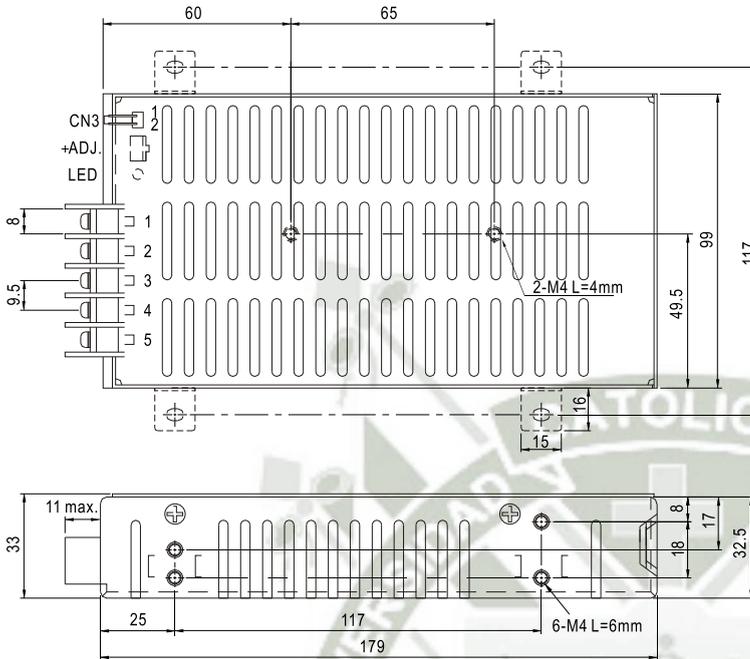


75W Single Output with PFC Function

SP-75

**Mechanical Specification**

Case No. 920A Unit:mm



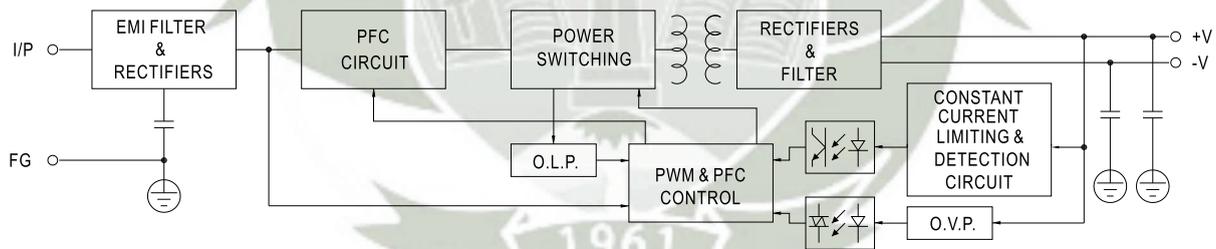
Terminal Pin No. Assignment

Pin No.	Assignment	Pin No.	Assignment
1	DC OUTPUT +V	4	AC/N
2	DC OUTPUT -V	5	AC/L
3	FG $\oplus$		

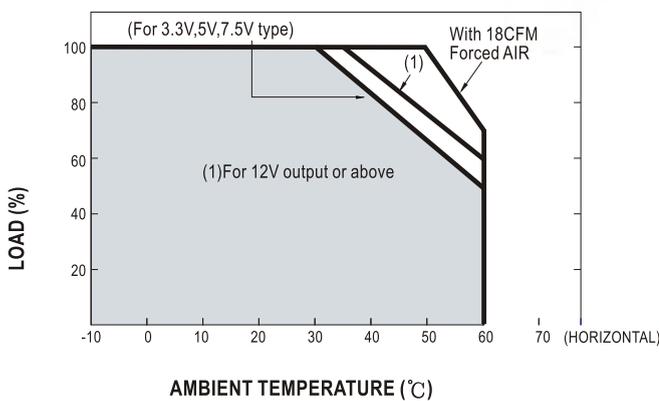
Remote ON/OFF(CN3): Molex 5046-02 or equivalent(optional)

Pin No.	Assignment	Mating Housing	Terminal
1	RC-	Molex 5051 or equivalent	Molex 2759 or equivalent
2	RC+		

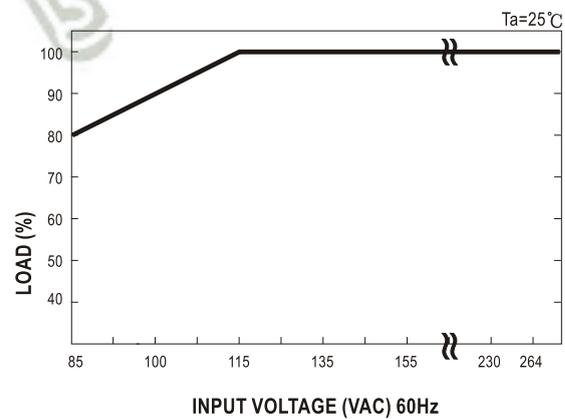
**Block Diagram**



**Derating Curve**



**Output Derating VS Input Voltage**

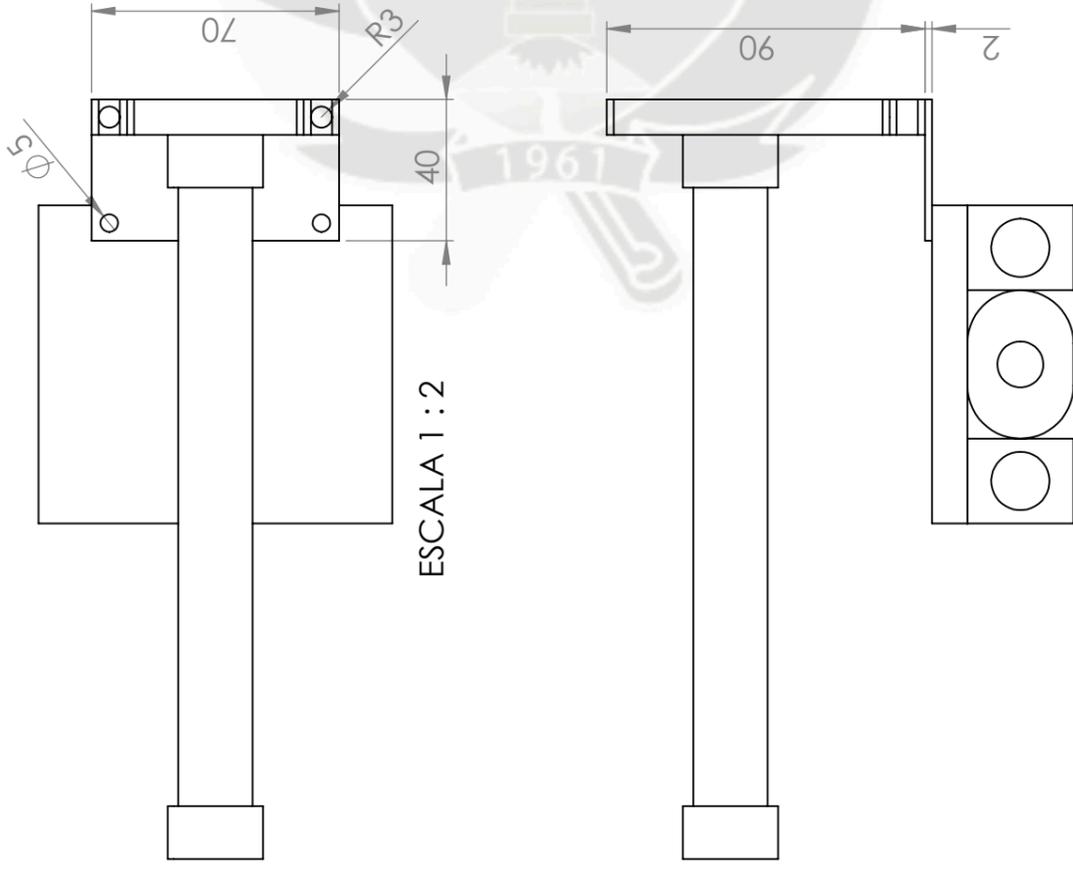




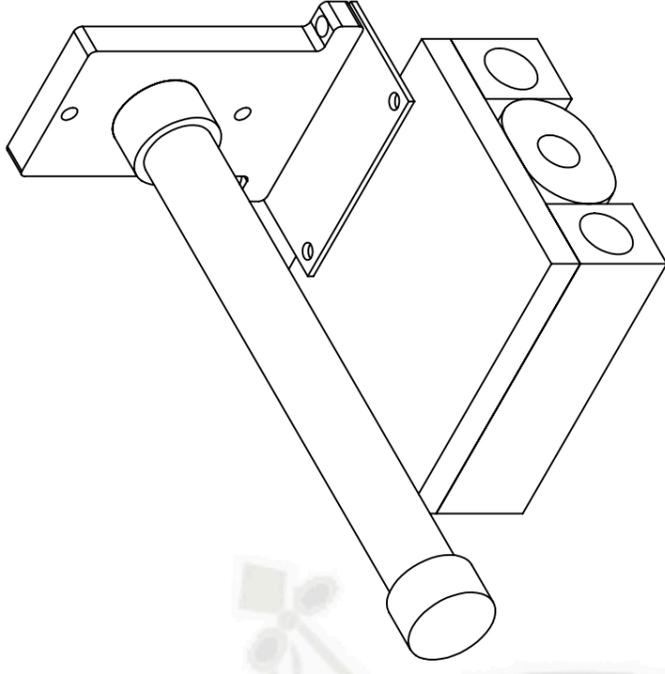
# ANEXO 4

## PLANOS MECÁNICO -ESTRUCTURALES





ESCALA 1 : 2

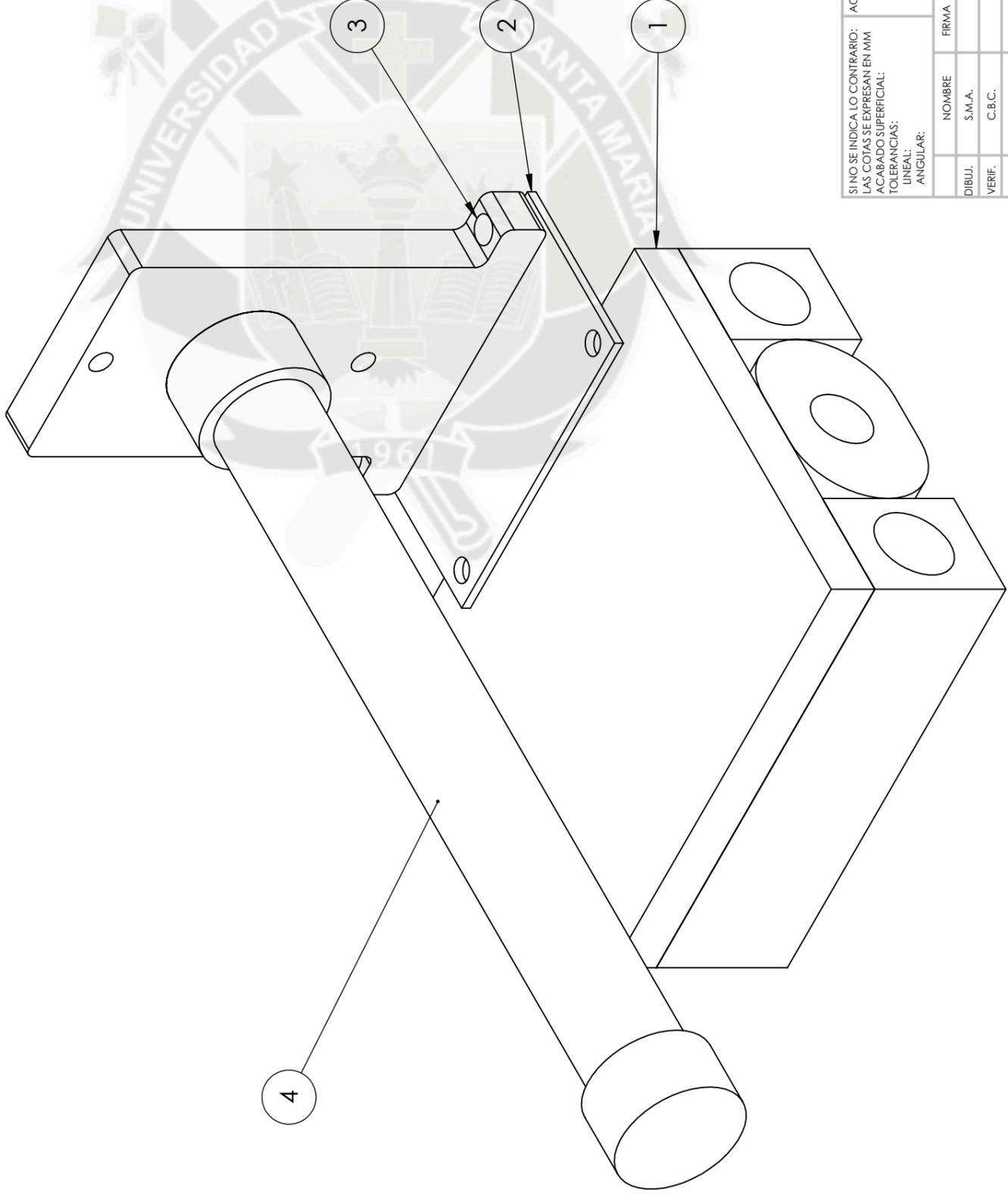


ESCALA 1 : 2

ESCALA 1 : 2

SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: ANGULAR:		ACABADO: REBARBAR Y ROMPER ARISTAS VIVAS		TÍTULO: "CONTROL E IMPLEMENTACION DE UN MODULO DE ALMACENAJE VERTICAL A TRAVEZ DE UN PIC Y UN SISTEMA SCADA PARA LA OPTIMIZACION DE UN PROCESO DE ALMACENAJE INDUSTRIAL."	
NOMBRE S.M.A. C.B.C. D.O.F. FABR. CALID.		FIRMA REVISION : 1 MATERIAL:		UNIVERSIDAD CATOLICA DE SANTA MARIA PROGRAMA PROFESIONAL DE ING. MECANICA MECANICA ELECTRICA Y MECATRONICA	
DIBUJ. VERIF. APROB. FABR. CALID.		FECHA REVISION : 1 MATERIAL:		DIBUJO Base mobile-Corredera Vertical	
		COD: UCSM-CIMAV-11-01		ESCALA:1:5 HOJA 1 DE 2	
				A3	

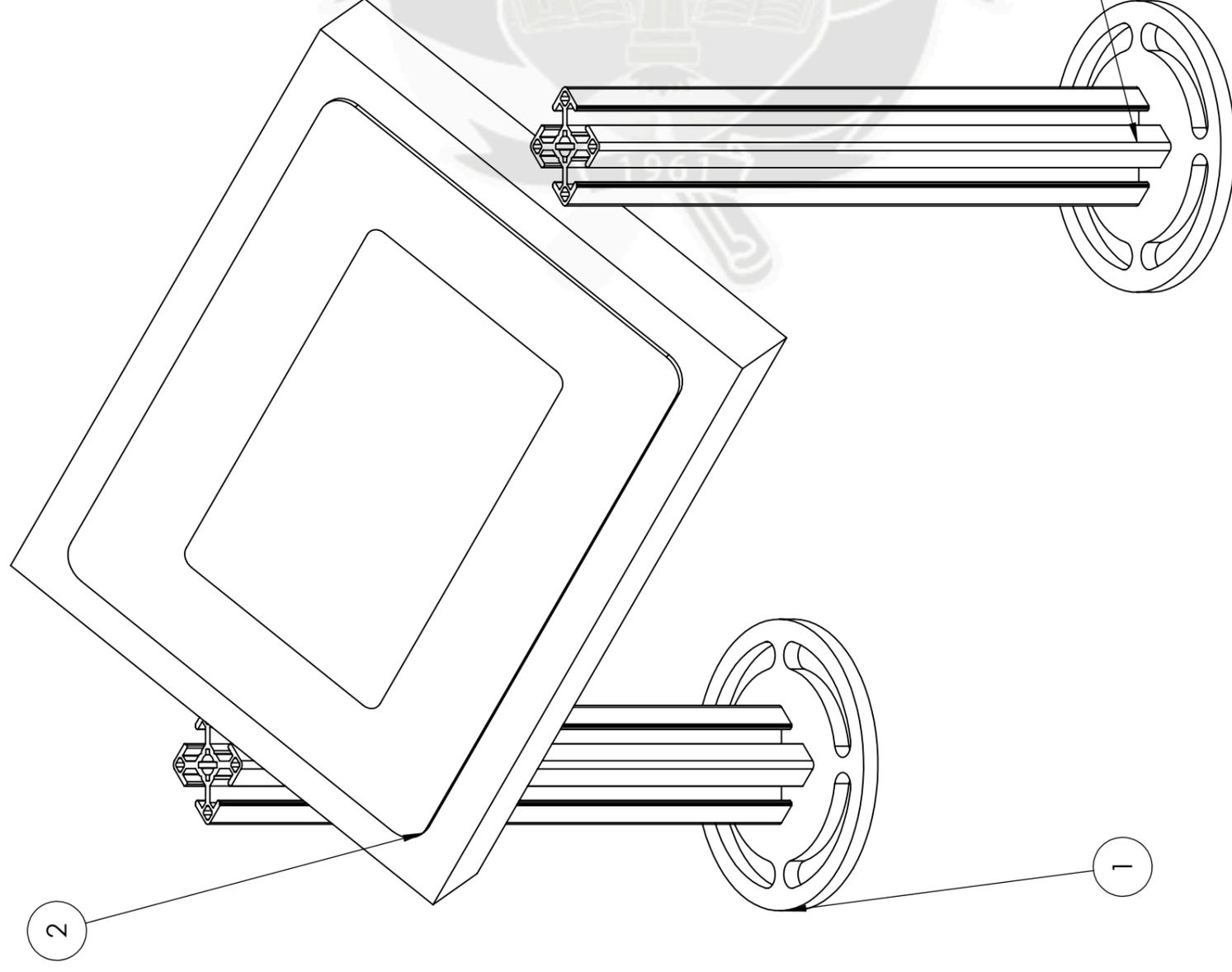
N.º DE ELEMENTO	N.º DE PIEZA	DESCRIPCIÓN	CANT
1	Base móvil - corredera vertical		1
2	Platina	R <sub>L</sub> 40mmx70mmx2mm	1
3	SopORTE piston		1
4	Piston		1



ESCALA 1 : 1

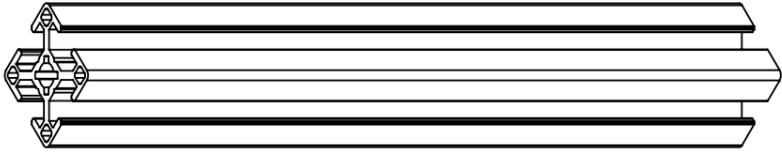
SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: ANGULAR:		ACABADO: REBARBAR Y ROMPER ARISTAS VIVAS		TÍTULO: "CONTROL E IMPLEMENTACION DE UN MODULO DE ALMACENAJE VERTICAL A TRAVES DE UN PLC Y UN SISTEMA SCADA PARA LA OPTIMIZACION DE UN PROCESO DE ALMACENAJE INDUSTRIAL."	
NOMBRE S.M.A. C.B.C. D.O.F.		FIRMA REVISION : 1		"UNIVERSIDAD CATOLICA DE SANTA MARIA" PROGRAMA PROFESIONAL DE ING. MECANICA MECANICA ELECTRICA Y MECATRONICA	
DIBUJ. VERIF. APROB. FABR. CALID.		FECHA MATERIAL:		DIBUJO Base móvil-Corredera vertical	
NO CAMBIE LA ESCALA		COD: UCSM-CIMAV-11-02		ESCALA:1:5	
		ESCALA:1:5		HOJA 2 DE 2	

N.º DE ELEMENTO	N.º DE PIEZA	DESCRIPCIÓN	CANT
1	Base		2
2	Pantalla touch 7"		1
3	Soporte de pantalla	Rexroth 30mmx30mmx210mm	1

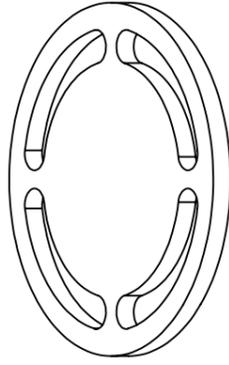


ESCALA 1 : 2

SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: LINEAL: ANGULAR:		ACABADO: REBARBAR Y ROMPER ARISTAS VIVAS		TÍTULO: "CONTROL E IMPLEMENTACION DE UN MODULO DE ALMACENAJE VERTICAL A TRAVEZ DE UN PLC Y UN SISTEMA SCADA PARA LA OPTIMIZACION DE UN PROCESO DE ALMACENAJE INDUSTRIAL."	
NOMBRE S.M.A. C.B.C. D.O.F. FABR. CALID.		FIRMA FECHA REVISION : 1 MATERIAL:		"UNIVERSIDAD CATOLICA DE SANTA MARIA" PROGRAMA PROFESIONAL DE ING. MECANICA Y MECATRONICA	
DIBUJ. VERIF. APROB. FABR. CALID.		NO CAMBIE LA ESCALA		DIBUJO Touch Screen	
		COD: UCSM-CIMAV-10-01		ESCALA:1:5 HOJA 1 DE 2	



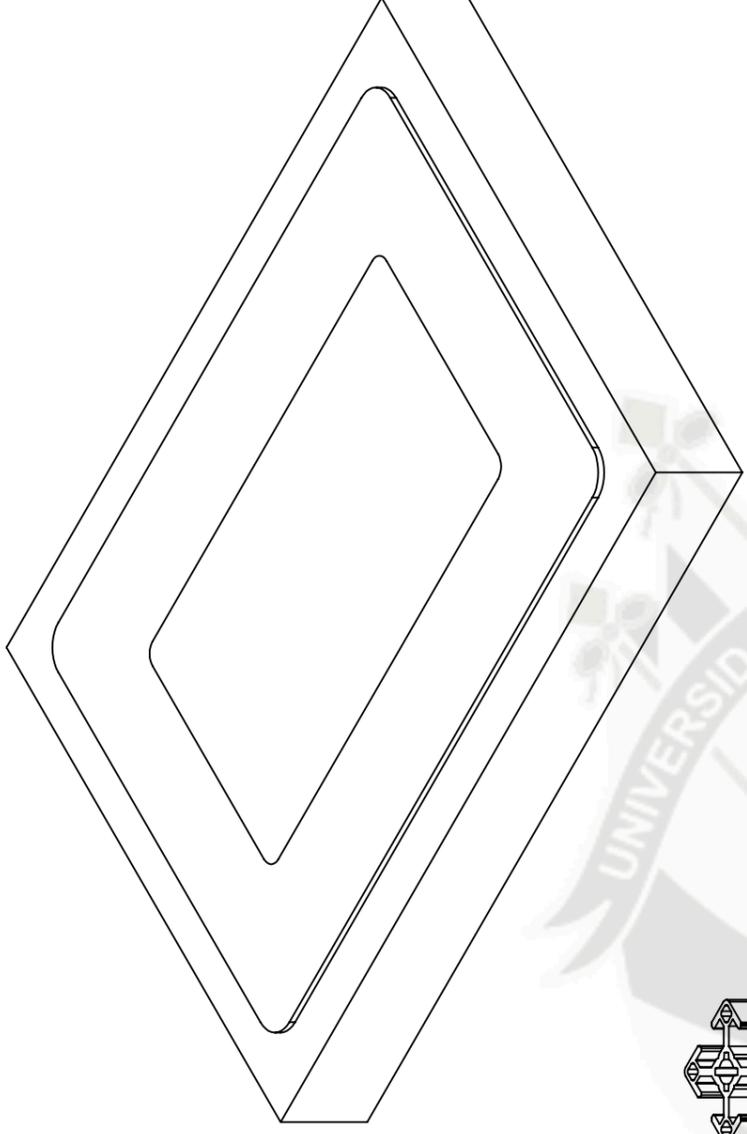
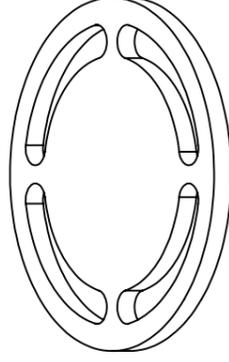
ESCALA 1 : 2



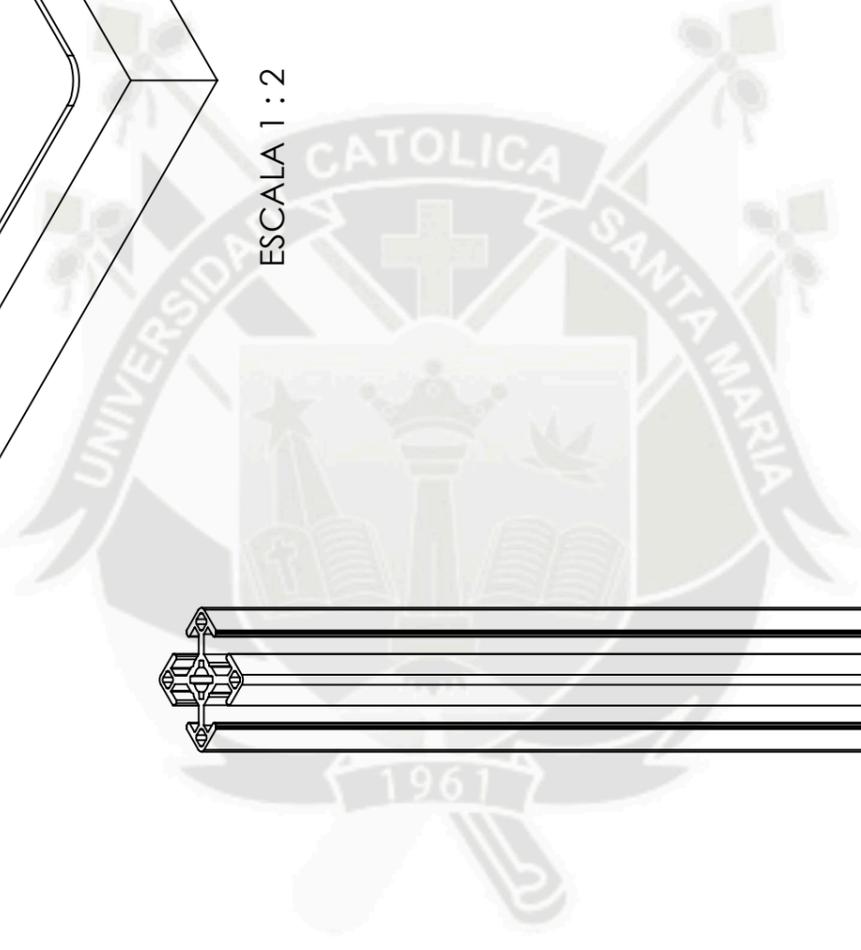
ESCALA 1 : 2



ESCALA 1 : 2



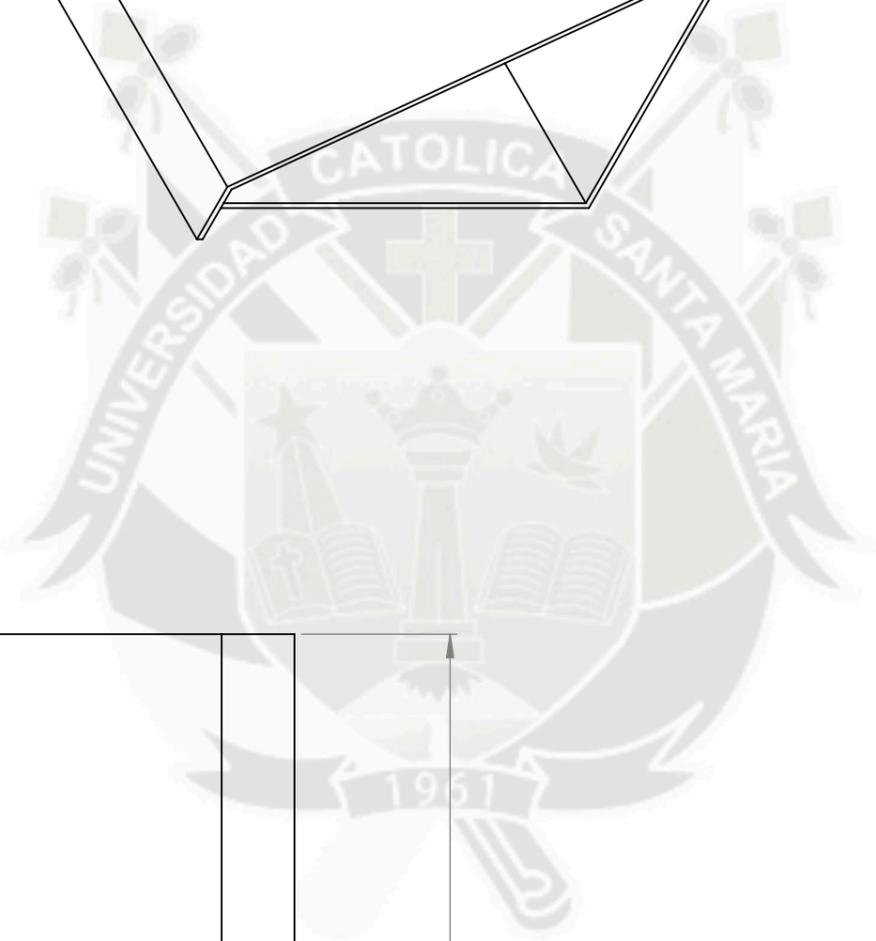
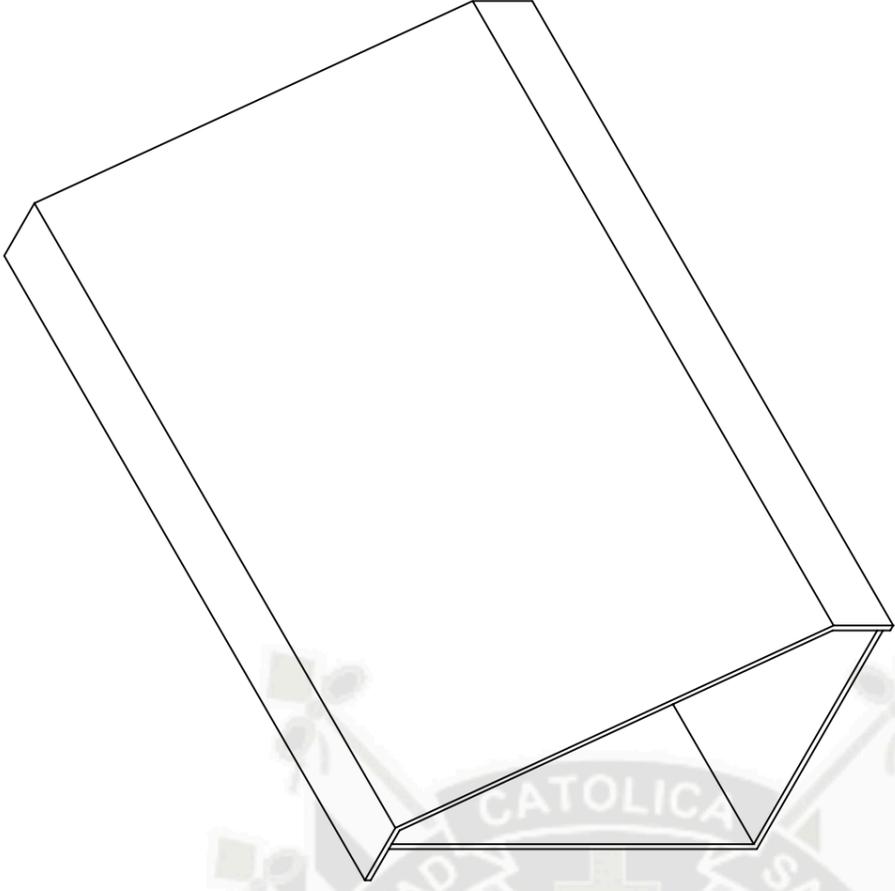
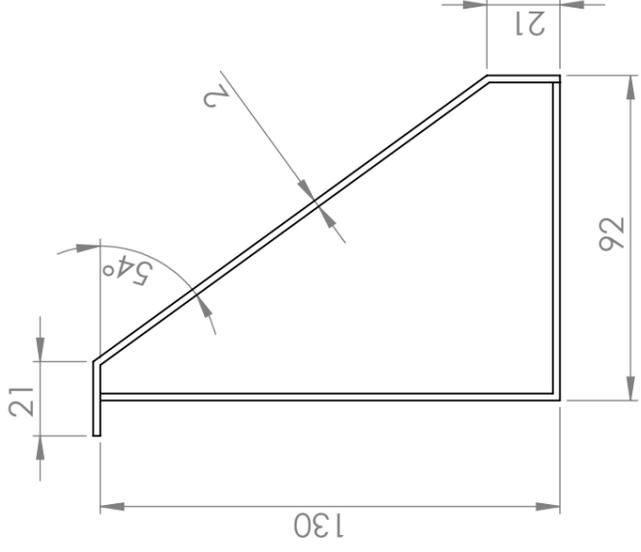
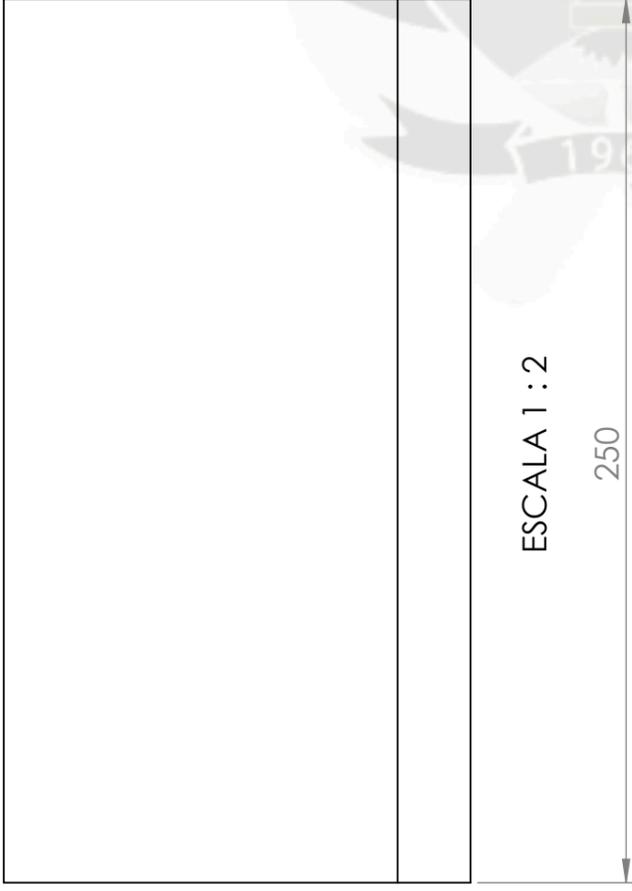
ESCALA 1 : 2

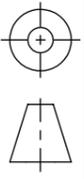


SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: ANGULAR:		ACABADO: REBARBAR Y ROMPER ARISTAS VIVAS		TÍTULO: "CONTROL E IMPLEMENTACION DE UN MODULO DE ALMACENAJE VERTICAL A TRAVEZ DE UN PLC Y UN SISTEMA SCADA PARA LA OPTIMIZACION DE UN PROCESO DE ALMACENAJE INDUSTRIAL."	
NOMBRE S.M.A. C.B.C. D.O.F.		FIRMA REVISION : 1		"UNIVERSIDAD CATOLICA DE SANTA MARIA" PROGRAMA PROFESIONAL DE ING. MECANICA MECANICA ELECTRICA Y MECATRONICA	
DIBUJ. VERIF. APROB. FABR. CALID.		FECHA MATERIAL:		DIBUJO Touch Screen	
NO CAMBIE LA ESCALA		COD: UCSM-CIMAV-10-02		ESCALA:1:5	
		ESCALA:1:5		HOJA 2 DE 2	

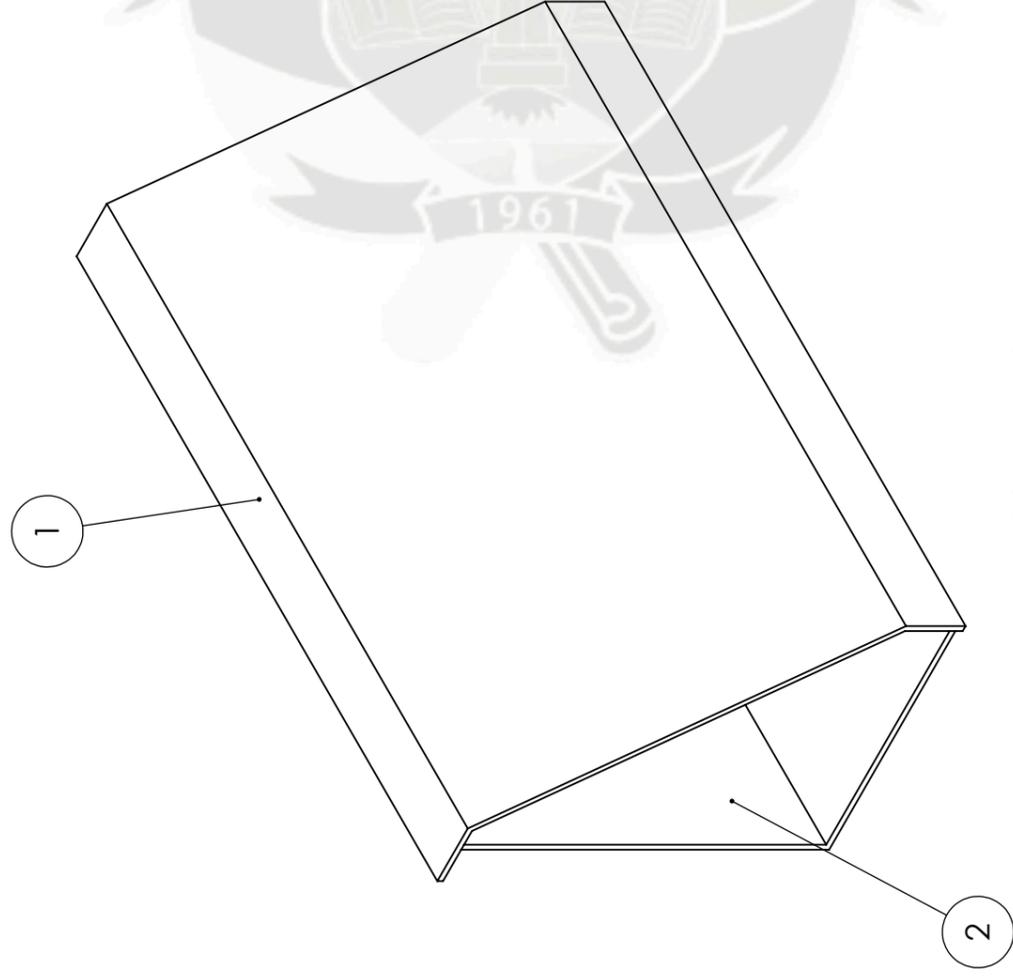


A3

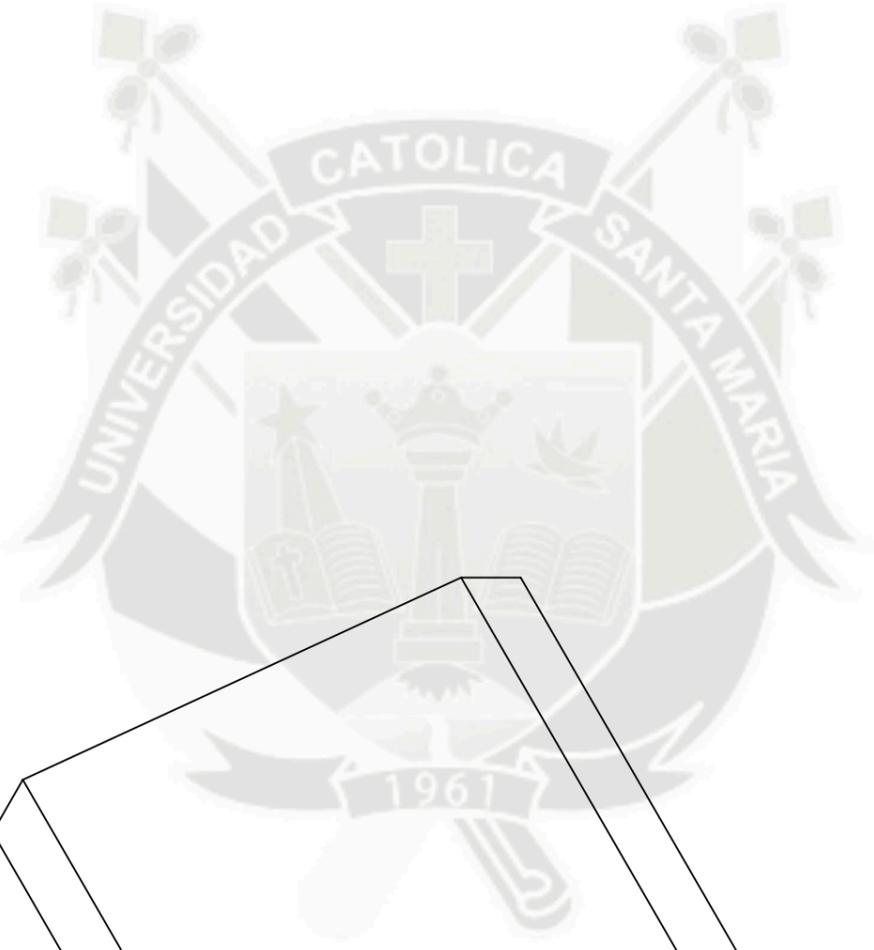


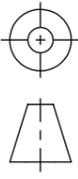
SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: ANGULAR:		ACABADO: REBARBAR Y ROMPER ARISTAS VIVAS		TÍTULO: "CONTROL E IMPLEMENTACION DE UN MODULO DE ALMACENAJE VERTICAL A TRAVEZ DE UN PLC Y UN SISTEMA SCADA PARA LA OPTIMIZACION DE UN PROCESO DE ALMACENAJE INDUSTRIAL."	
DIBUJ.	NOMBRE	FECHA	NO CAMBIE LA ESCALA	"UNIVERSIDAD CATOLICA DE SANTA MARIA" PROGRAMA PROFESIONAL DE ING. MECANICA MECANICA ELECTRICA Y MECATRONICA	
VERIF.	S.M.A.	REVISION : 1			
APROB.	C.B.C.			DIBUJO	
FABR.	D.O.F.			Tablero	
CALID.	MATERIAL:			A3	
		COD: UCSM-CIMAV-09-01		ESCALA:1:5	
				HOJA 1 DE 2	

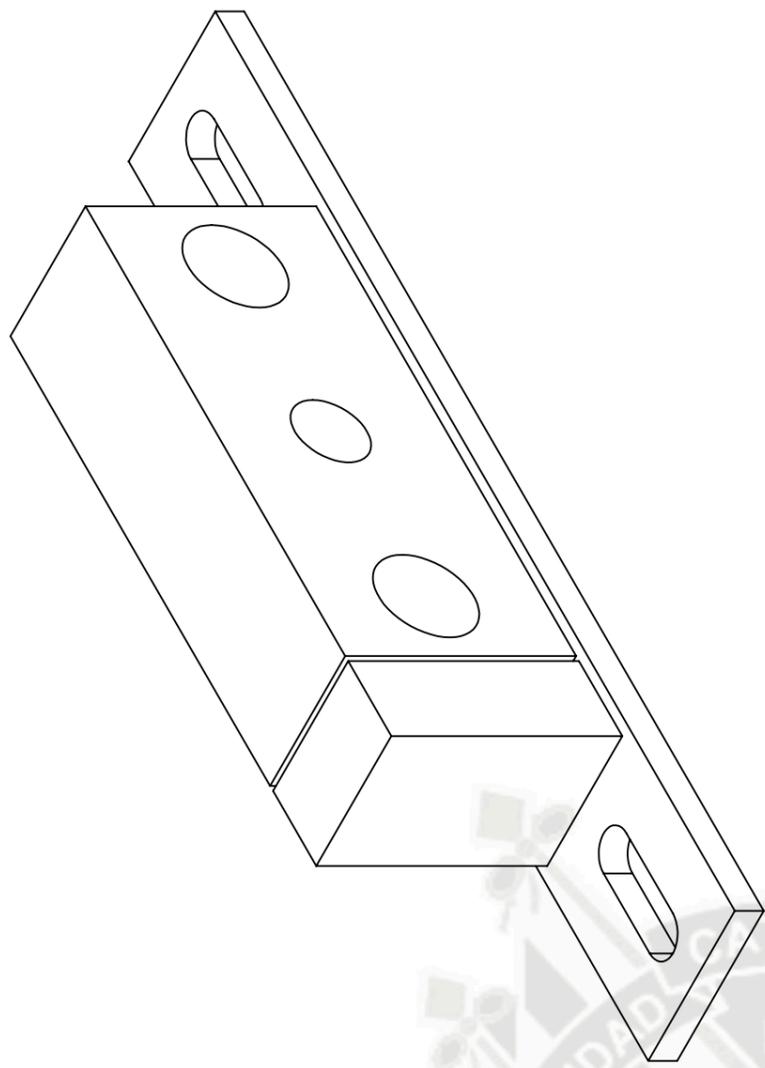
N.º DE ELEMENTO	N.º DE PIEZA	DESCRIPCIÓN	CANT
1	Platina frontal delgada	Platina para tablero de control	1
2	SopORTE posterior		1



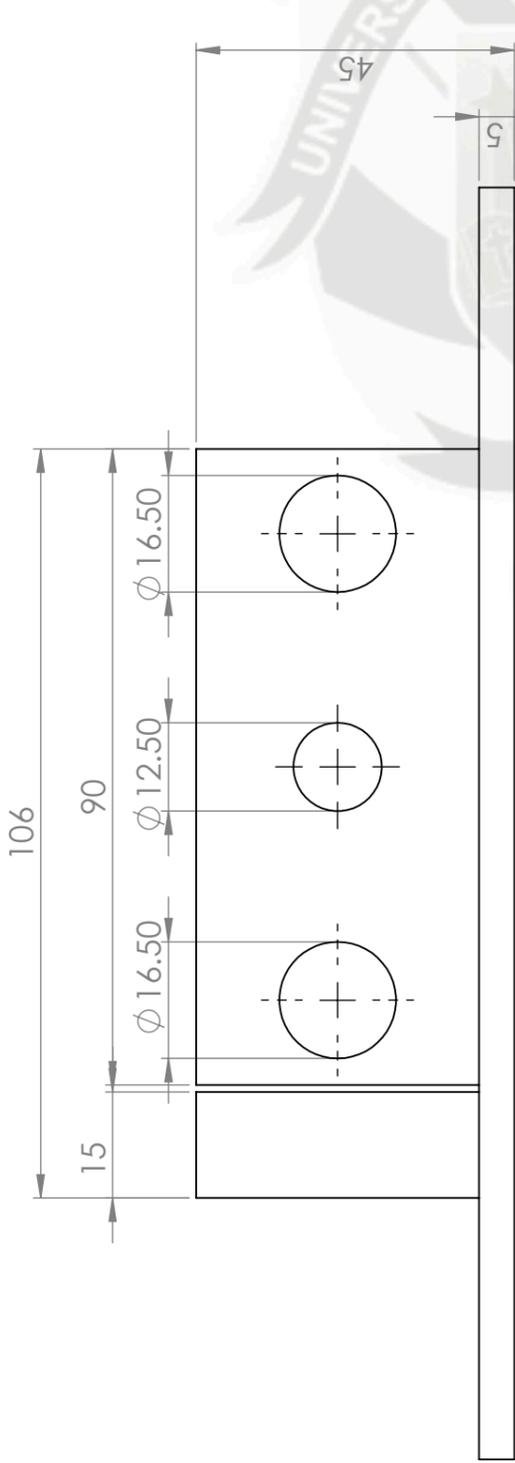
ESCALA 1 : 2



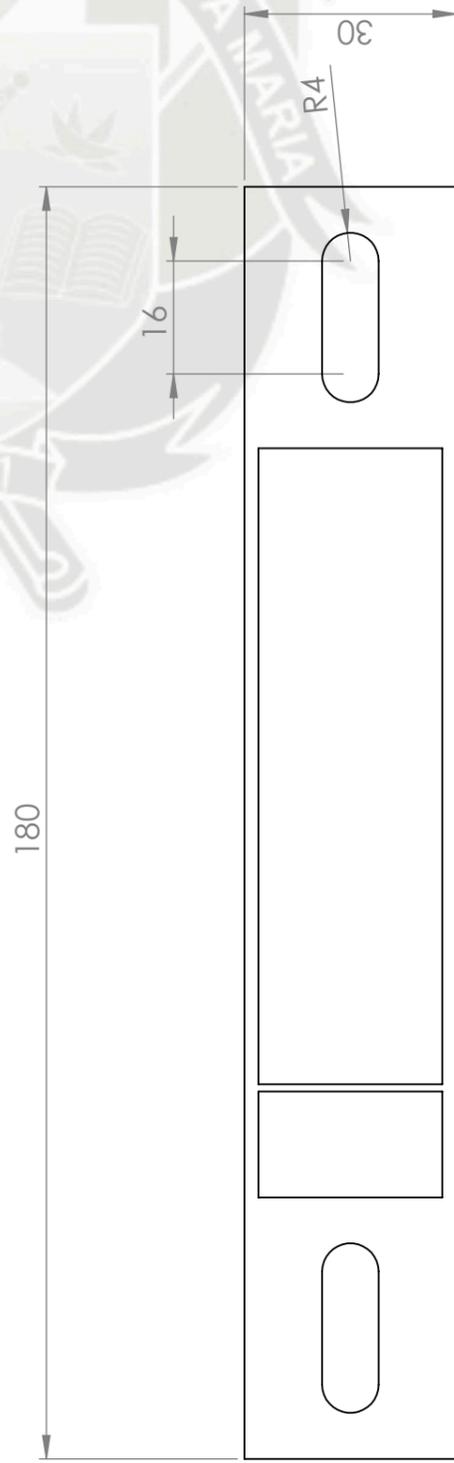
SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: LINEAL: ANGULAR:		ACABADO: REBARBAR Y ROMPER ARISTAS VIVAS		TÍTULO: "CONTROL E IMPLEMENTACION DE UN MODULO DE ALMACENAJE VERTICAL A TRAVEZ DE UN PLC Y UN SISTEMA SCADA PARA LA OPTIMIZACION DE UN PROCESO DE ALMACENAJE INDUSTRIAL."	
DIBUJ.	NOMBRE	FECHA	REVISION :	"UNIVERSIDAD CATOLICA DE SANTA MARIA" PROGRAMA PROFESIONAL DE ING. MECANICA MECANICA ELECTRICA Y MECATRONICA	
VERIF.	S.M.A.		1		
APROB.	C.B.C.			DIBUJO	
FABR.	D.O.F.			Tablero	
CALID.	MATERIAL:			A3	
		COD: UCSM-CIMAY-09-02		ESCALA:1:5	
				HOJA 2 DE 2	



ESCALA 1 : 1



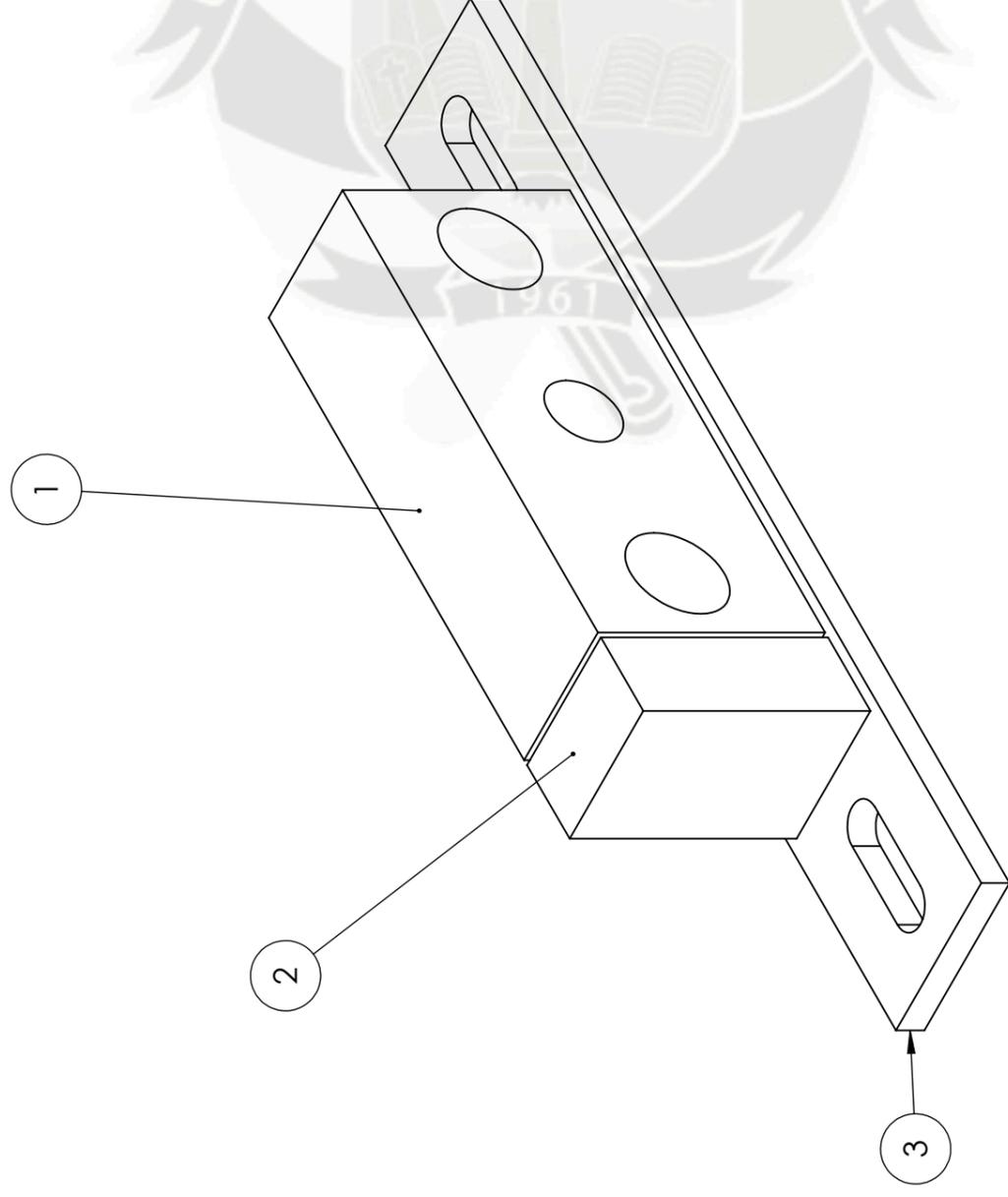
ESCALA 1 : 1



ESCALA 1 : 1

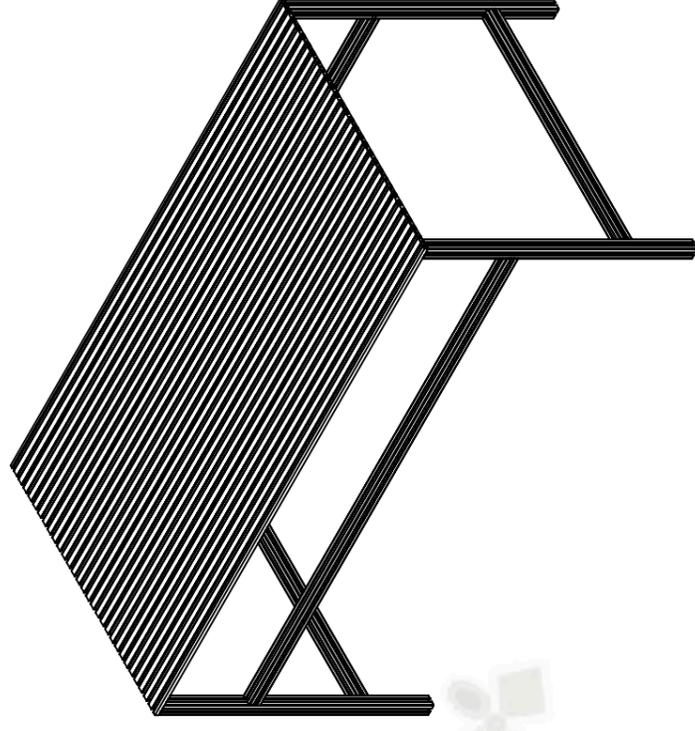
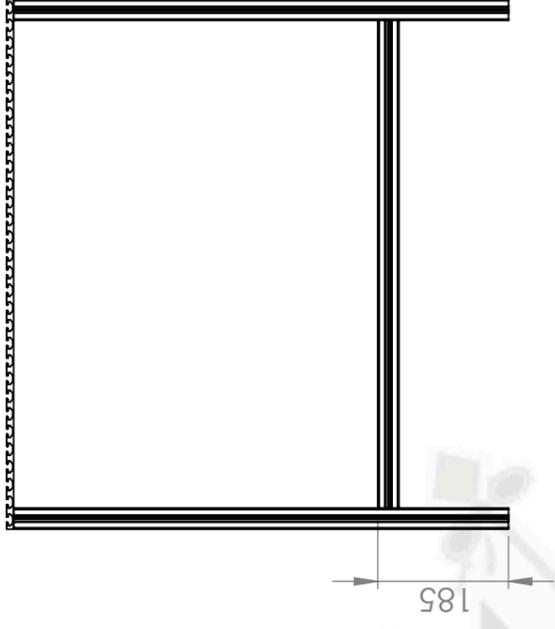
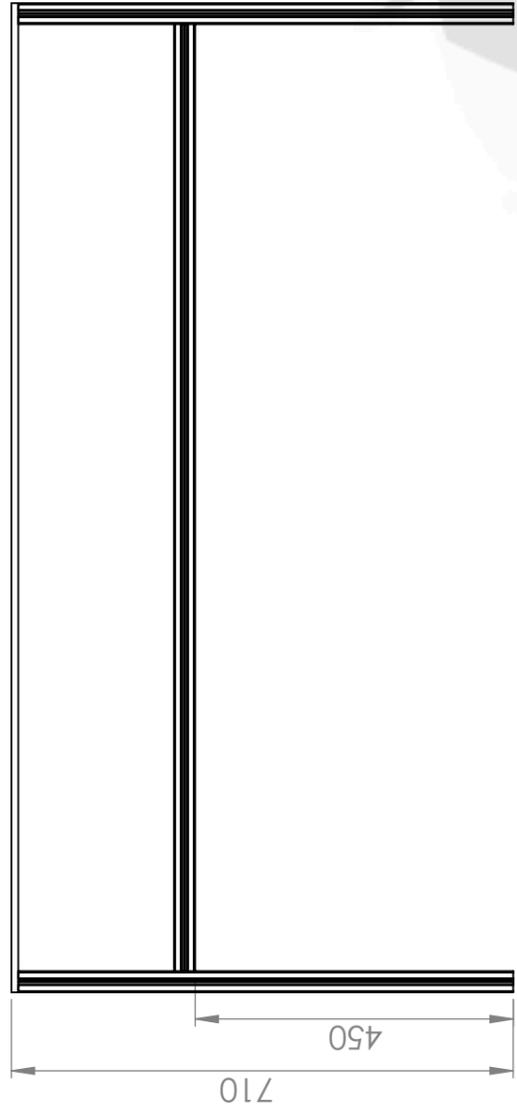
SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: ANGULAR:		ACABADO: REBARBAR Y ROMPER ARISTAS VIVAS		TÍTULO: "CONTROL E IMPLEMENTACION DE UN MODULO DE ALMACENAJE VERTICAL A TRAVEZ DE UN PIC Y UN SISTEMA SCADA PARA LA OPTIMIZACION DE UN PROCESO DE ALMACENAJE INDUSTRIAL."	
DIBUJ.	NOMBRE	FRMA	FECHA	"UNIVERSIDAD CATOLICA DE SANTA MARIA" PROGRAMA PROFESIONAL DE ING. MECANICA MECANICA ELECTRICA Y MECATRONICA	
VERIF.	S.M.A.			DIBUJO Tope de correderas	
APROB.	C.B.C.		REVISION : 1	A3	
FABR.	D.O.F.			ESCALA:1:2	
CALID.			MATERIAL:	COD: UCSM-CIMAV-08-01	

Nº	PIEZA	DESCRIPCION	CANT
1	Soporte guias y tornillos		1
2	Soporte guias sensores		1
3	Platina base		1



ESCALA 1 : 1

SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: ANGULAR:		ACABADO:		REBARBAR Y ROMPER ARISTAS VIVAS		TÍTULO:	
DIBUJ.	NOMBRE	FRMA	FECHA	NO CAMBIE LA ESCALA		"CONTROL E IMPLEMENTACION DE UN MODULO DE ALMACENAJE VERTICAL A TRAVEZ DE UN PIC Y UN SISTEMA SCADA PARA LA OPTIMIZACION DE UN PROCESO DE ALMACENAJE INDUSTRIAL."	
VERIF.	S.M.A.					"UNIVERSIDAD CATOLICA DE SANTA MARIA"	
APROB.	C.B.C.		REVISION :			PROGRAMA PROFESIONAL DE ING. MECANICA MECANICA ELECTRICA Y MECATRONICA	
FABR.	D.O.F.						
CALID.			MATERIAL:			DIBUJO	
						Tope de Correderas	
						A3	
						ESCALA:1:2	
				COD: UCSM-CIMAV-08-02		HOJA 2 DE 2	

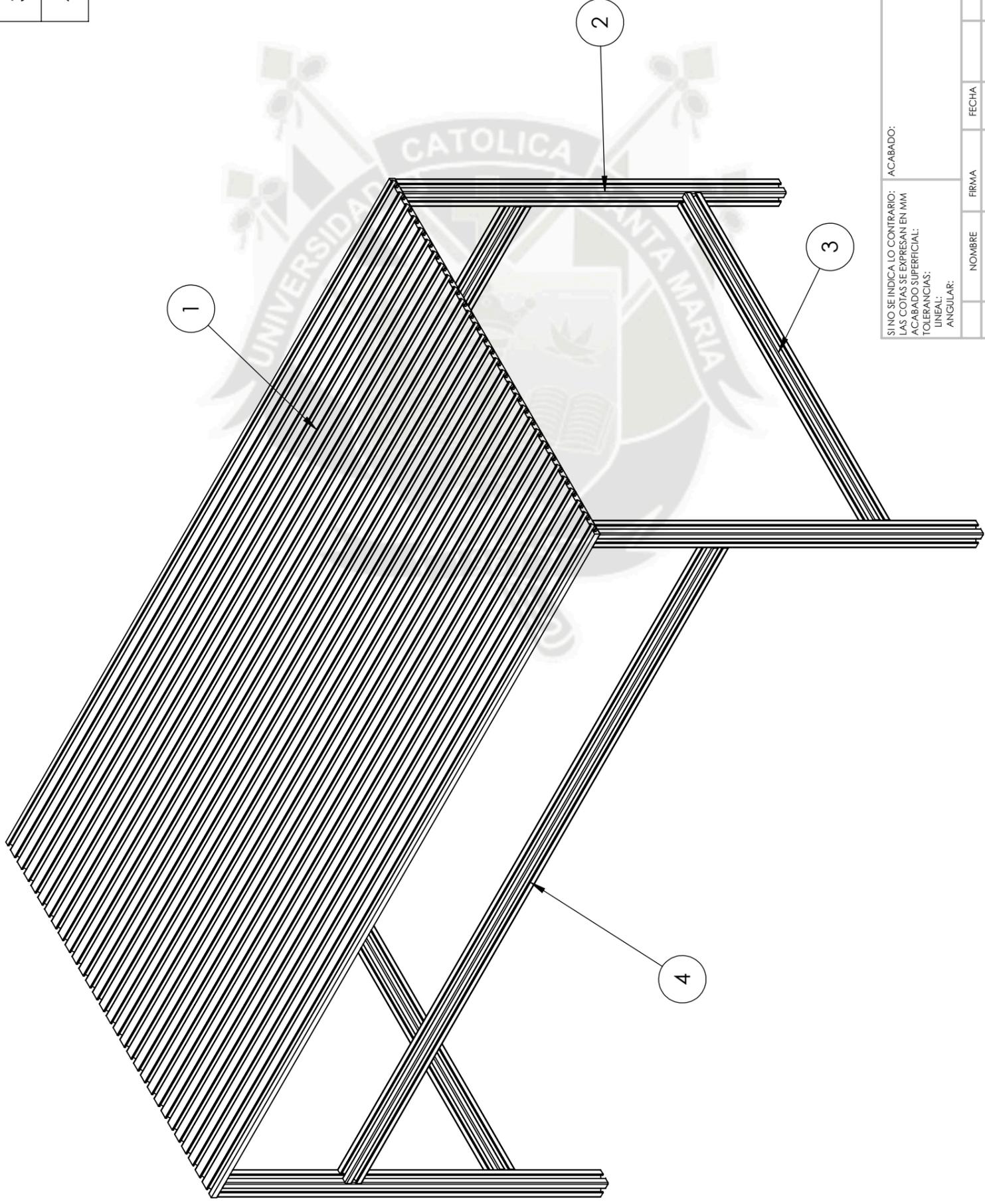


750

ESCALA 1 : 15

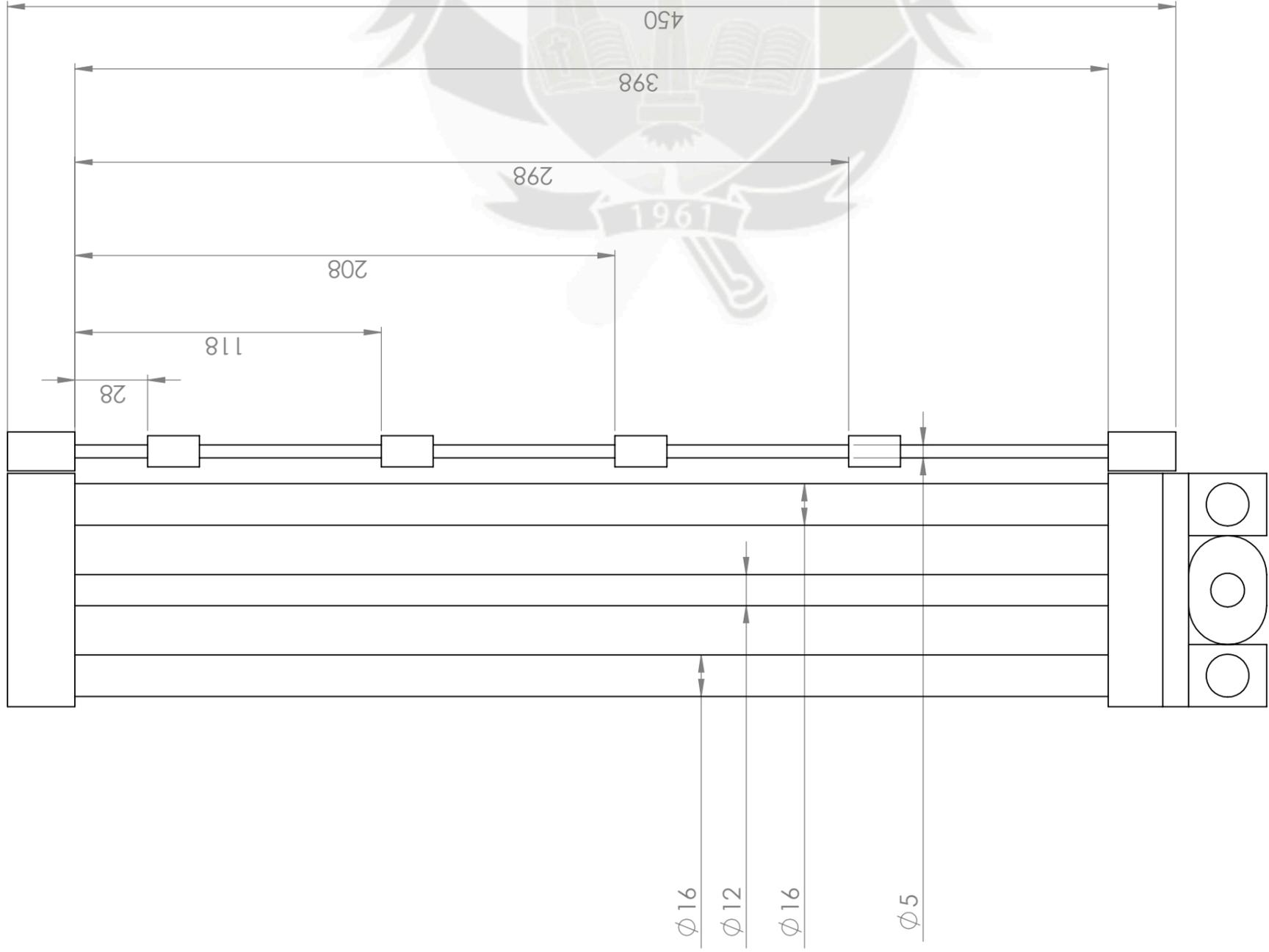
SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: LINEAL: ANGULAR:		ACABADO: REBARBAR Y ROMPER ARISTAS VIVAS	TÍTULO: "CONTROL E IMPLEMENTACION DE UN MODULO DE ALMACENAJE VERTICAL A TRAVEZ DE UN PLC Y UN SISTEMA SCADA PARA LA OPTIMIZACION DE UN PROCESO DE ALMACENAJE INDUSTRIAL."		 "UNIVERSIDAD CATOLICA DE SANTA MARIA" PROGRAMA PROFESIONAL DE ING. MECANICA MECANICA ELECTRICA Y MECATRONICA		A3
DIBUJ. VERIF. APROB. FABR. CALID.	NOMBRE S.M.A. C.B.C. D.OF.	FIRMA REVISION : 1 MATERIAL:	FECHA DIBUJO Mesa de Trabajo	ESCALA: 1:10 HOJA 1 DE 2			

Nº	PIEZA	DESCRIPCION	CANT.
1	Mesa	Rexroth 1.4mx0.75m	1
2	Patas	□ Rexroth 30mmx30mmx0.71m	4
3	Travesaño 1	□ Rexroth 30mmx30mmx0.69m	2
4	Travesaño 2	□ Rexroth 30mmx30mmx1.34m	2

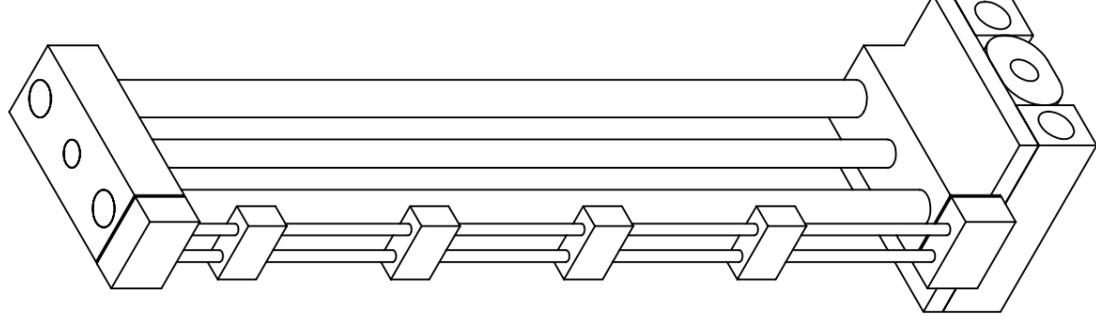


ESCALA 1 : 7

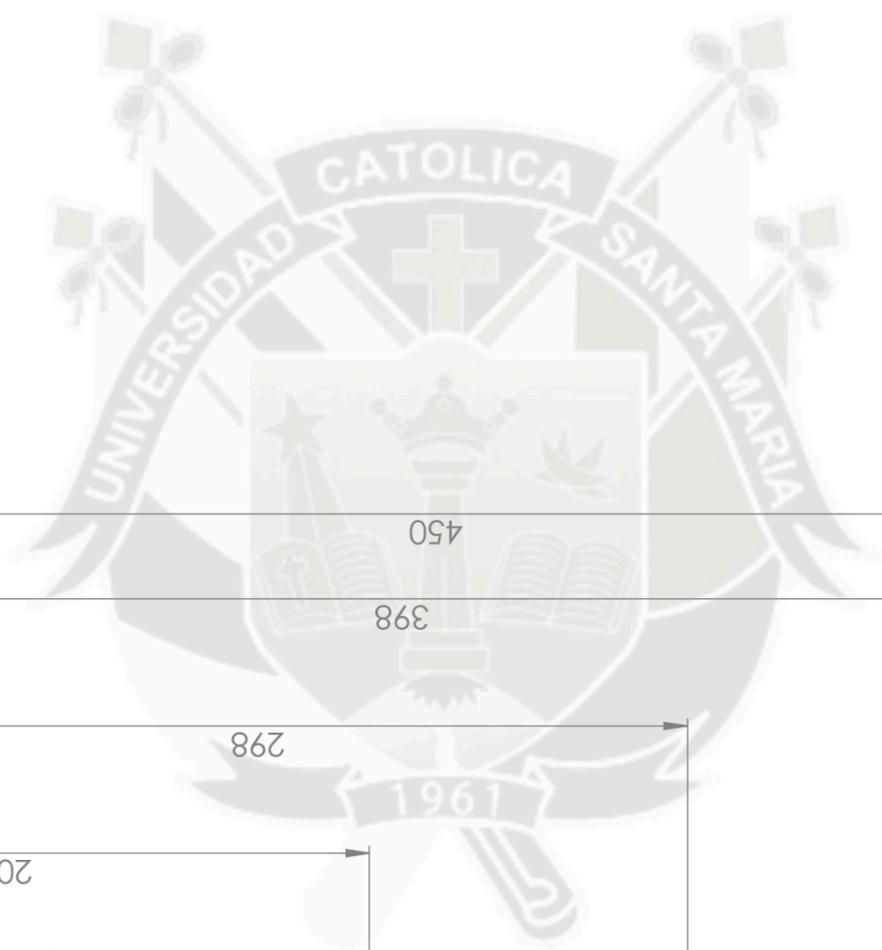
SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: LINEAL: ANGULAR:		ACABADO: REBARBAR Y ROMPER ARISTAS VIVAS		TÍTULO: "CONTROL E IMPLEMENTACION DE UN MODULO DE ALMACENAJE VERTICAL A TRAVEZ DE UN PIC Y UN SISTEMA SCADA PARA LA OPTIMIZACION DE UN PROCESO DE ALMACENAJE INDUSTRIAL."	
DIBUJ. VERIF. APROB. FABR. CALID.		NOMBRE S.M.A. C.B.C. D.O.F.		UNIVERSIDAD CATOLICA DE SANTA MARIA PROGRAMA PROFESIONAL DE ING. MECANICA MECANICA ELECTRICA Y MECATRONICA	
FIRMA REVISION : 1		NO CAMBIE LA ESCALA		DIBUJO Mesa de Trabajo	
FECHA		MATERIAL:		A3	
COD: UCSM-CIMAY-07-02		ESCALA:1:10		HOJA 2 DE 2	



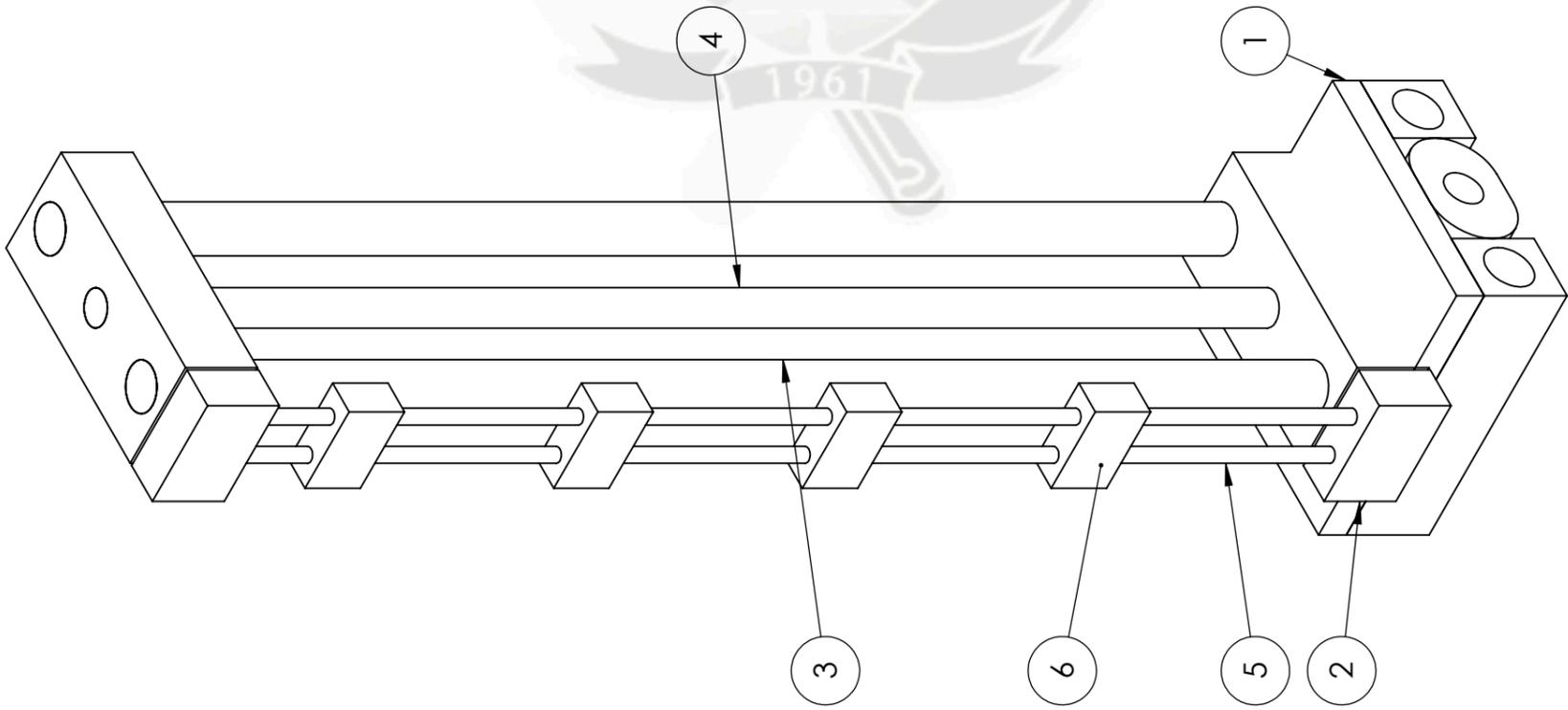
ESCALA 1 : 2



ESCALA 1 : 3



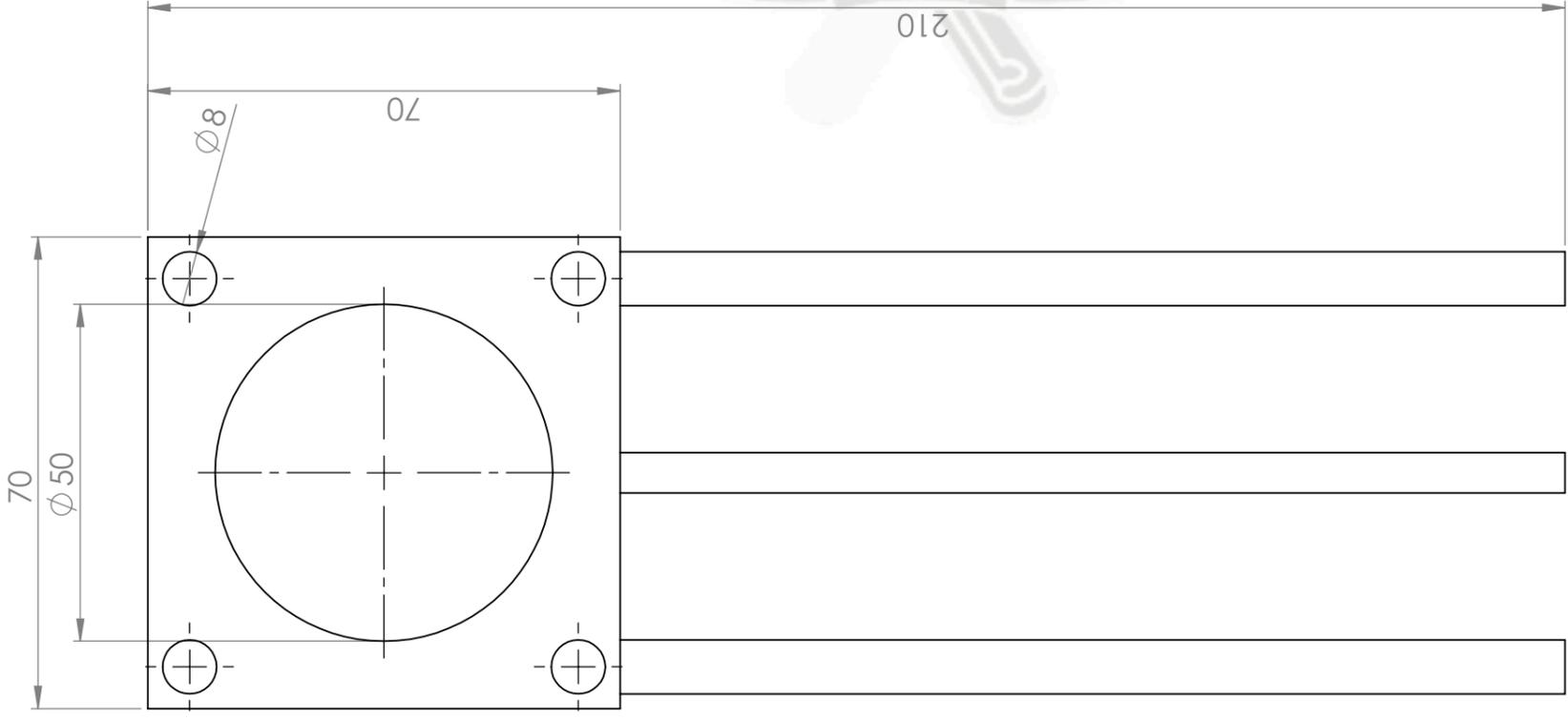
SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: ANGULAR:		ACABADO: REBARBAR Y ROMPER ARISTAS VIVAS		TÍTULO: "CONTROL E IMPLEMENTACION DE UN MODULO DE ALMACENAJE VERTICAL A TRAVEZ DE UN PLC Y UN SISTEMA SCADA PARA LA OPTIMIZACION DE UN PROCESO DE ALMACENAJE INDUSTRIAL."	
NOMBRE S.M.A. C.B.C. D.O.F. FABR. CALID.		FIRMA REVISION : 1 MATERIAL:		DIBUJO Corredera Vertical	
DIBUJ. VERIF. APROB. FABR. CALID.		FECHA REVISION : 1 MATERIAL:		ESCALA: 1:5 COD: UCSM-CIMAY-06-01	
UNIVERSIDAD CATOLICA SANTA MARIA PROGRAMA PROFESIONAL DE ING. MECANICA Y MECATRONICA		PPIMEEM		A3	



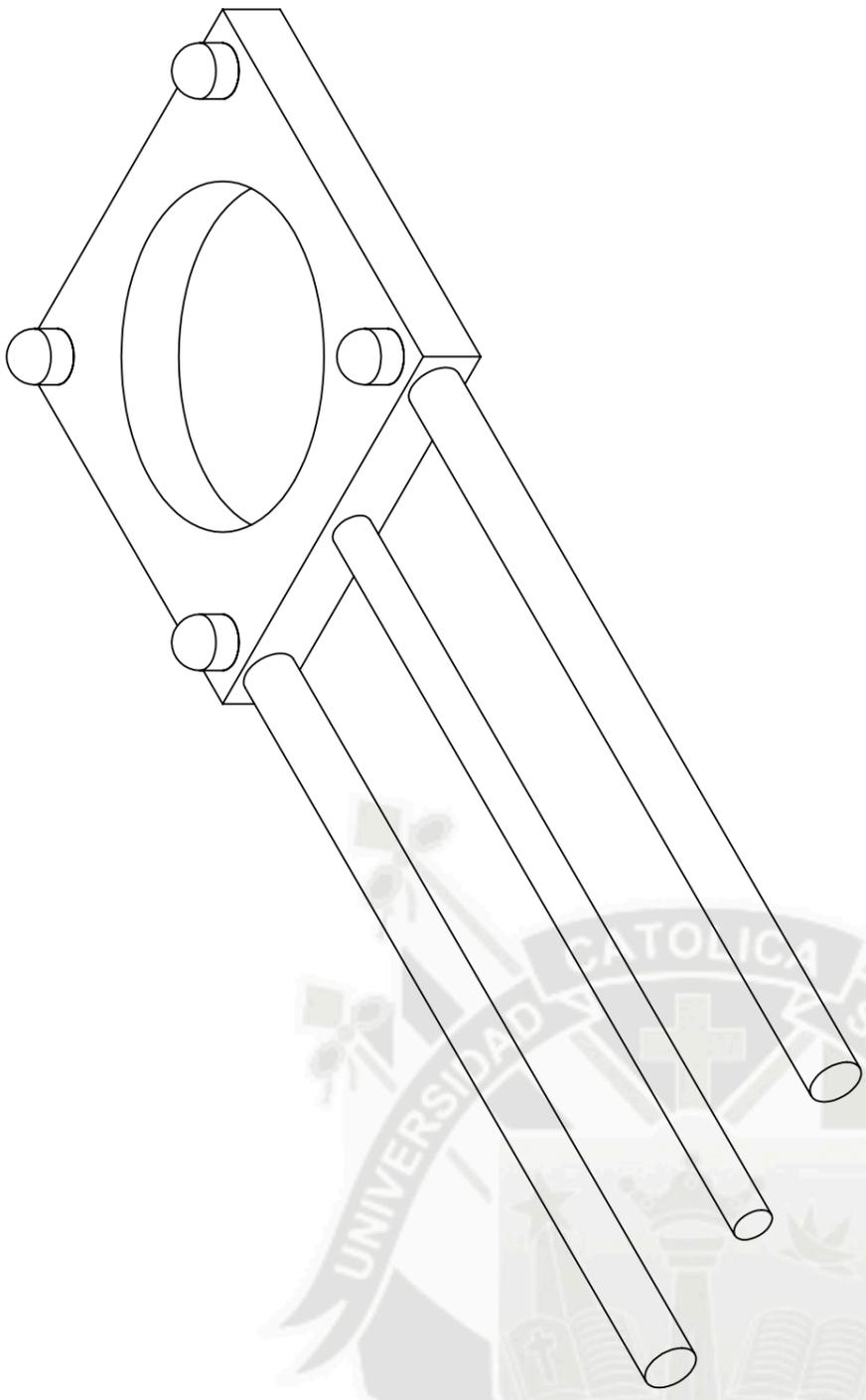
ESCALA 1 : 2

N.º DE ELEMENTO	N.º DE PIEZA	DESCRIPCIÓN	CANT
1	Base movable		1
2	Soportes laterales		2
3	Guía vertical		2
4	Tornillo vertical		1
5	Guía sensores		2
6	Sensores		4

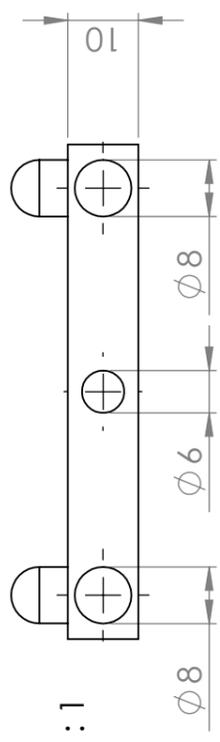
SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: ANGULAR:		ACABADO: REBARBAR Y ROMPER ARISTAS VIVAS	TÍTULO: "CONTROL E IMPLEMENTACION DE UN MODULO DE ALMACENAJE VERTICAL A TRAVEZ DE UN PLC Y UN SISTEMA SCADA PARA LA OPTIMIZACION DE UN PROCESO DE ALMACENAJE INDUSTRIAL."
NOMBRE S.M.A. C.B.C. D.O.F. CALID.	FIRMA REVISION : 1 MATERIAL:	NO CAMBIE LA ESCALA DIBUJO	"UNIVERSIDAD CATOLICA DE SANTA MARIA" PROGRAMA PROFESIONAL DE ING. MECANICA MECANICA ELECTRICA Y MECATRONICA
ESCALA: 1:5 COD: UCSM-CIMAV-06-02			A3 Corredera Vertical



ESCALA 1 : 1



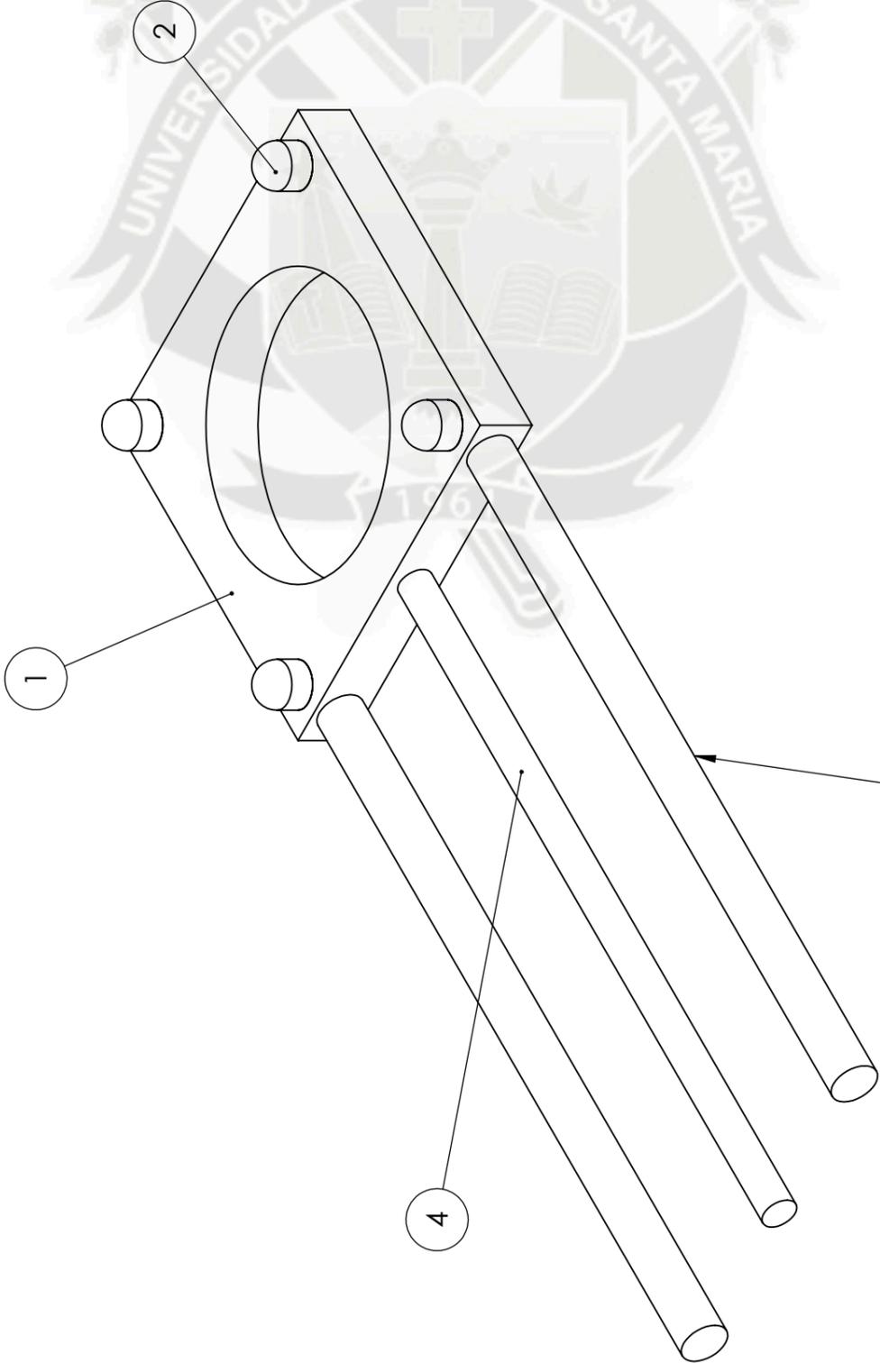
ESCALA 1 : 1



ESCALA 1 : 1

SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: LINEAL: ANGULAR:		ACABADO: RERABAR Y ROMPER ARISTAS VIVAS		TÍTULO: "CONTROL E IMPLEMENTACION DE UN MODULO DE ALMACENAJE VERTICAL A TRAVEZ DE UN PIC Y UN SISTEMA SCADA PARA LA OPTIMIZACION DE UN PROCESO DE ALMACENAJE INDUSTRIAL."	
NOMBRE S.M.A. C.B.C. D.O.F. FABR. CALID.		FIRMA REVISION : 1 MATERIAL:		UNIVERSIDAD CATOLICA DE SANTA MARIA PROGRAMA PROFESIONAL DE ING. MECANICA Y MECATRONICA	
DIBUJ. VERIF. APROB. FABR. CALID.		FECHA NO CAMBIE LA ESCALA		DIBUJO Herramienta Piston	
		COD: UCSM-CIMAV-05-01		ESCALA:1:2 HOJA 1 DE 2	

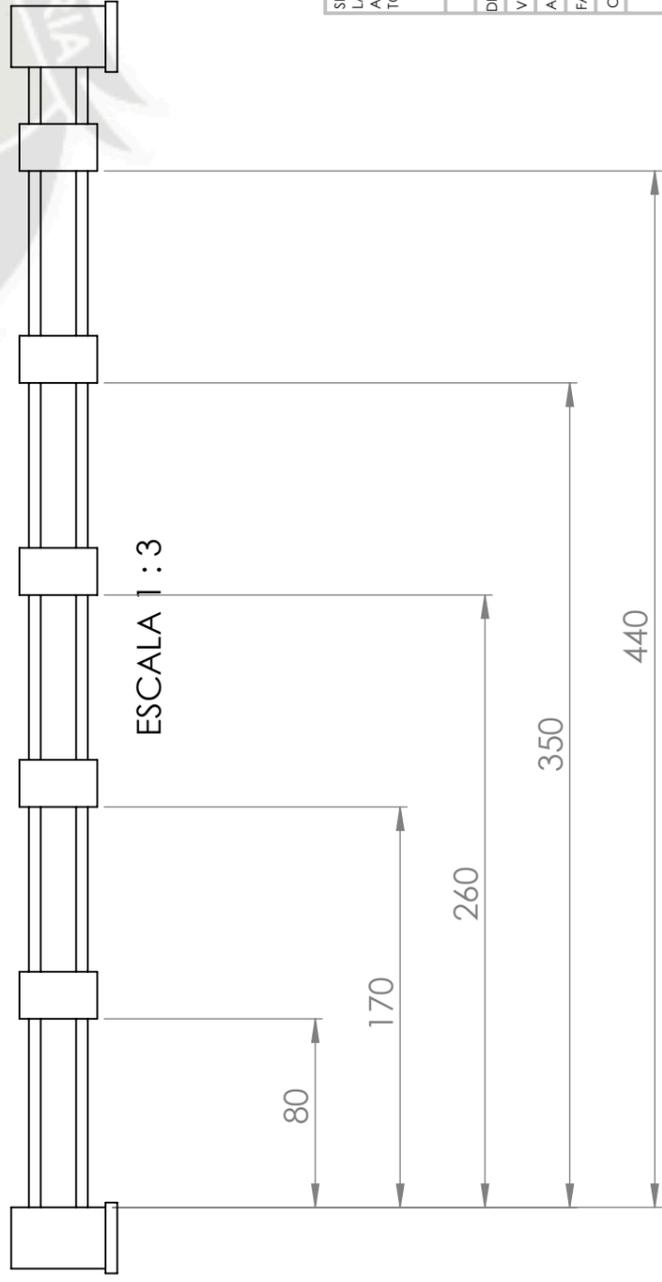
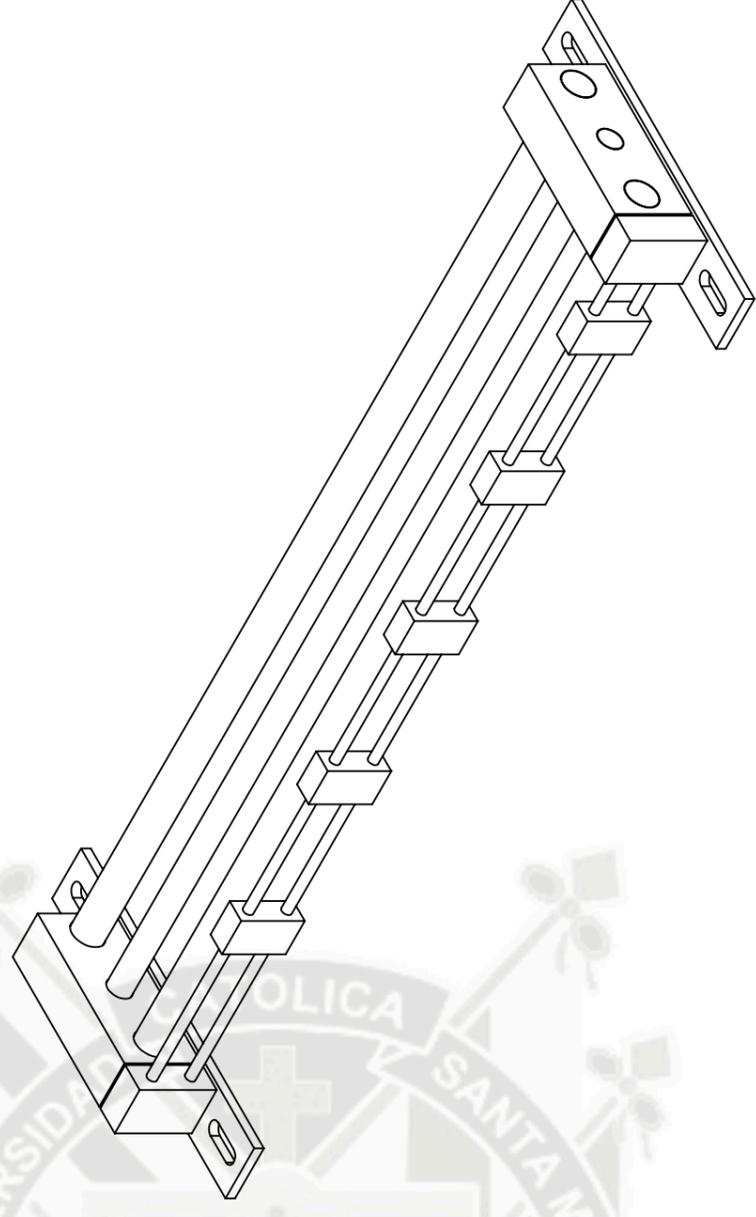
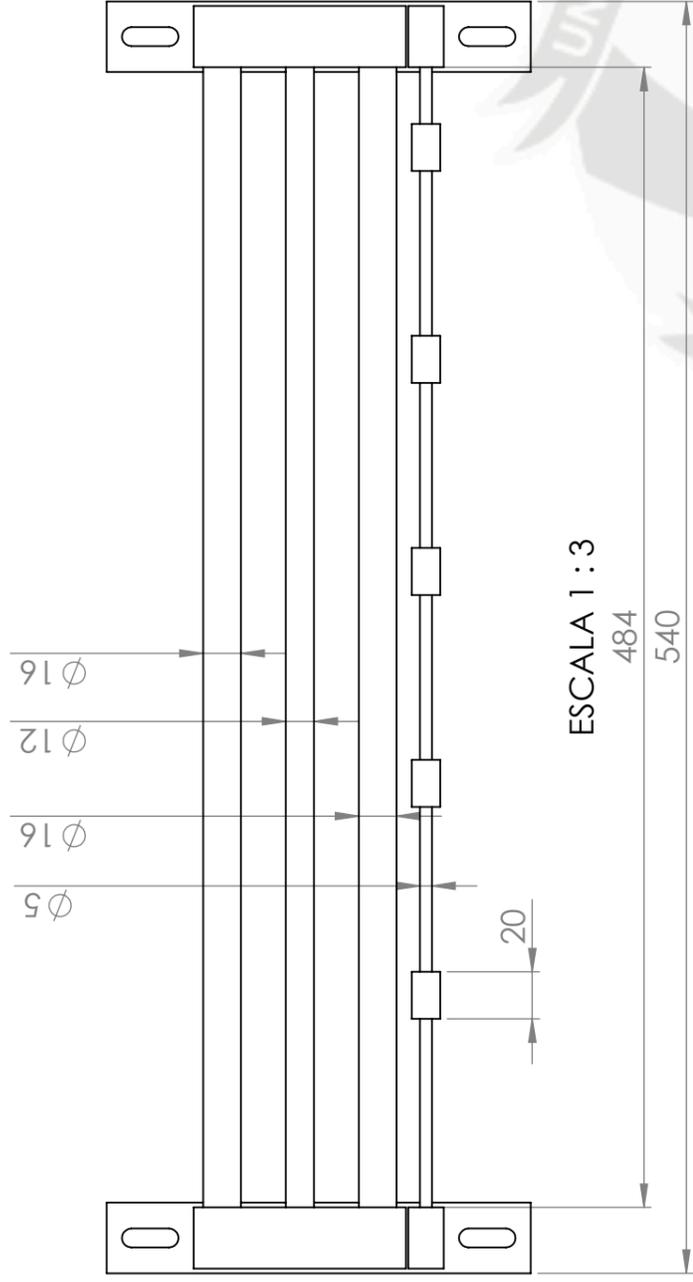
N.º DE ELEMENTO	N.º DE PIEZA	DESCRIPCIÓN	CANT
1	Herramienta piston	Platina cuadrada para transportar la pieza	1
2	Soportes	Soportes para colocar la pieza	4
3	Guías	Guías laterales de la herramienta	2
4	Eje piston	Actuador que va conectado al piston neumático	1



ESCALA 1 : 1

3

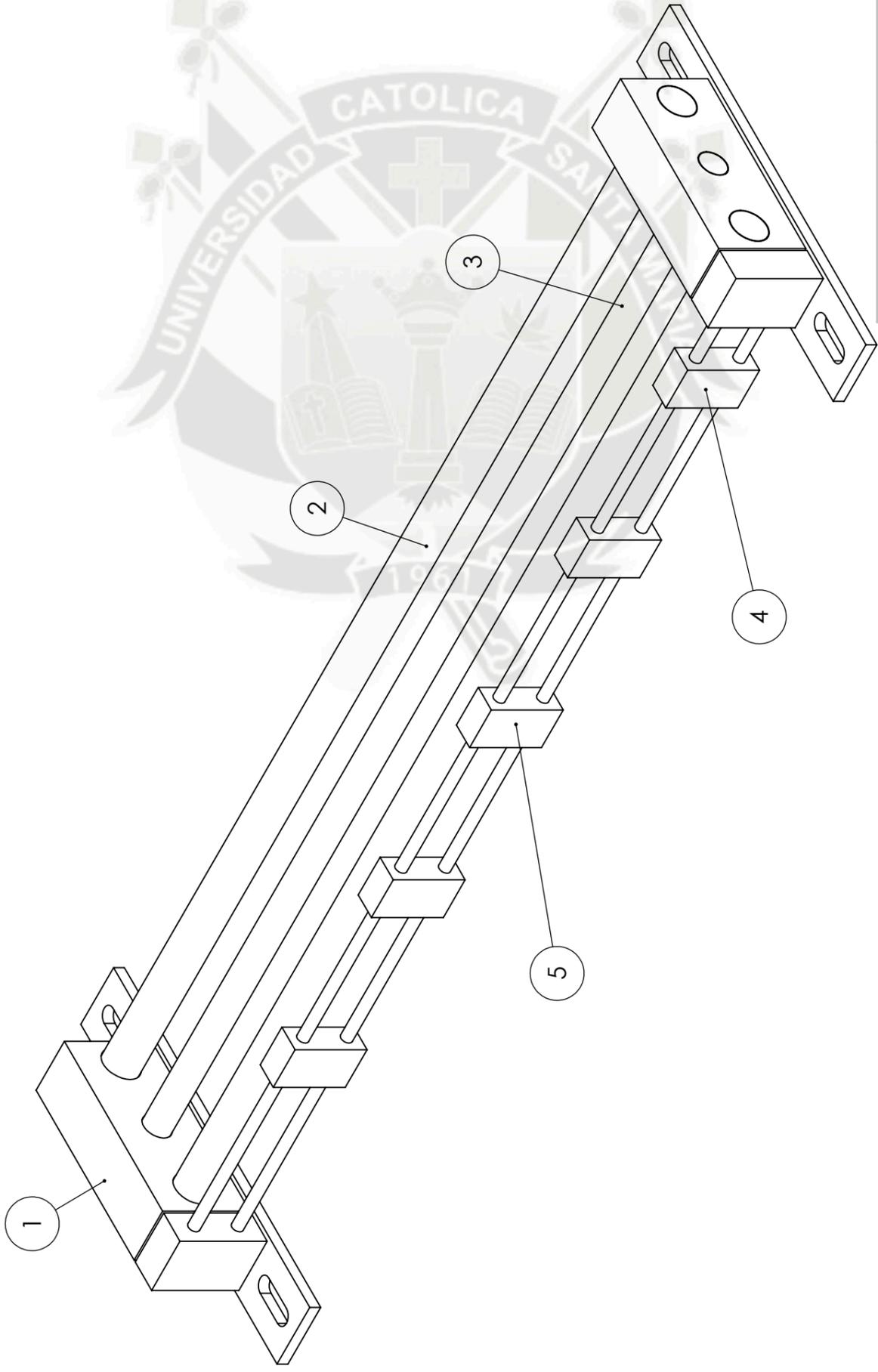
SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: LINEAL: ANGULAR:		ACABADO: ROMPER ARISTAS VIVAS		TÍTULO: "CONTROL E IMPLEMENTACION DE UN MODULO DE ALMACENAJE VERTICAL A TRAVEZ DE UN PLC Y UN SISTEMA SCADA PARA LA OPTIMIZACION DE UN PROCESO DE ALMACENAJE INDUSTRIAL."	
NOMBRE S.M.A. C.B.C. D.O.F. FABR. CALID.		FIRMA REVISION : 1 MATERIAL:		"UNIVERSIDAD CATOLICA DE SANTA MARIA" PROGRAMA PROFESIONAL DE ING. MECANICA MECANICA ELECTRICA Y MECATRONICA	
DIBUJ. VERIF. APROB. FABR. CALID.		FECHA NO CAMBIE LA ESCALA		DIBUJO Herramienta Piston	
		COD: UCSM-CIMAV-05-02		ESCALA:1:2 HOJA 2 DE 2	



ESCALA 1 : 3

SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: ANGULAR:		ACABADO: ROMPER ARISTAS VIVAS		TÍTULO: <b>"IMPLEMENTACION DE UN MODULO AUTOMATIZADO DE ALMACENAJE VERTICAL INDUSTRIAL A ESCALA Y SU IMPACTO EN LA OPTIMIZACION DEL PROCESO DE ALMACENAJE INDUSTRIAL"</b>	
NO CAMBIE LA ESCALA		REVISION : 1		"UNIVERSIDAD CATOLICA DE SANTA MARIA" PROGRAMA PROFESIONAL DE ING. MECANICA Y MECATRONICA	
DIBUJ. NOMBRE FRMA		DIBUJO		A3	
VERIF. S.M.A.		MATERIAL:		Corredera Horizontal	
APROB. C.B.C.		COD: UCSM-CIMAV-04-01		ESCALA:1:5	
FABR. D.O.F.		HOJA 1 DE 2			
CALID.					

N.º DE ELEMENTO	N.º DE PIEZA	DESCRIPCIÓN	CANT
1	SopORTE Lateral		2
2	Guía horizontal		2
3	Tornillo Horizontal		1
4	Guía Sensores		1
5	Sensores		5



ESCALA 1 : 2

ACABADO:

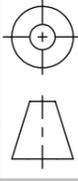
SI NO SE INDICA LO CONTRARIO:  
LAS COTAS SE EXPRESAN EN MM  
ACABADO SUPERFICIAL:  
TOLERANCIAS:  
LINEAL:  
ANGULAR:

REBARBAR Y  
ROMPER ARISTAS  
VIVAS

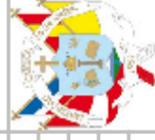
NO CAMBIE LA ESCALA

REVISION : 1

MATERIAL:



TÍTULO:  
"CONTROL E IMPLEMENTACION DE UN  
MODULO DE ALMACENAJE VERTICAL A TRAVEZ  
DE UN PLC Y UN SISTEMA SCADA PARA LA  
OPTIMIZACION DE UN PROCESO DE  
ALMACENAJE INDUSTRIAL."



"UNIVERSIDAD CATOLICA DE  
SANTA MARIA"  
PROGRAMA PROFESIONAL  
DE ING. MECANICA  
Y MECATRONICA

DIBUJO

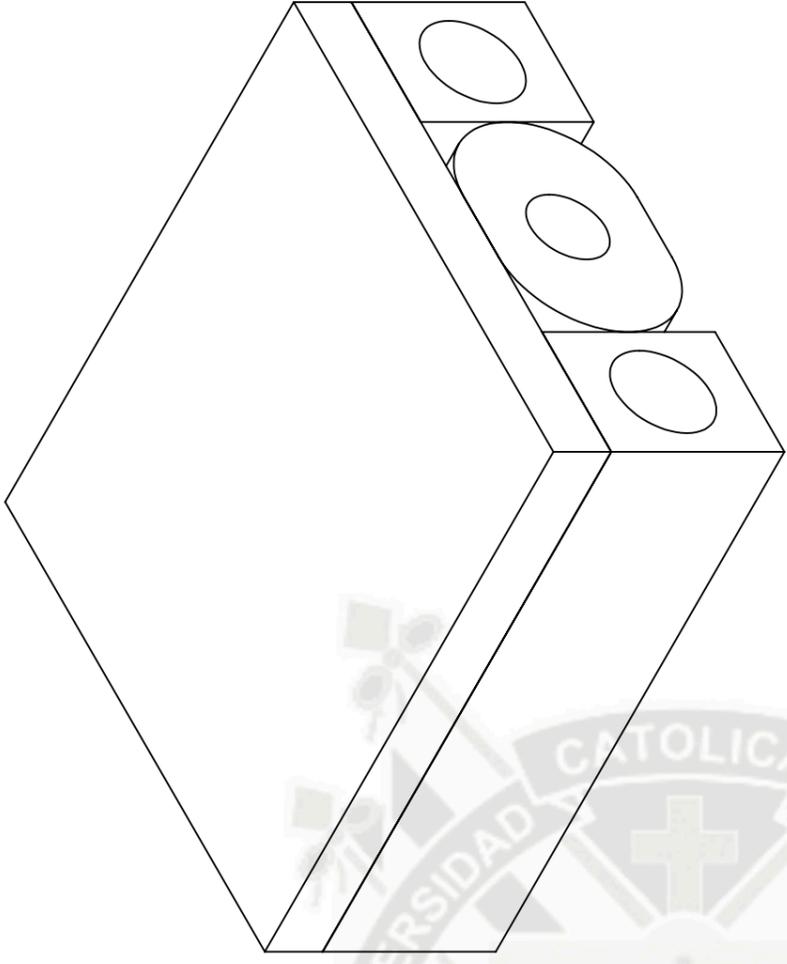
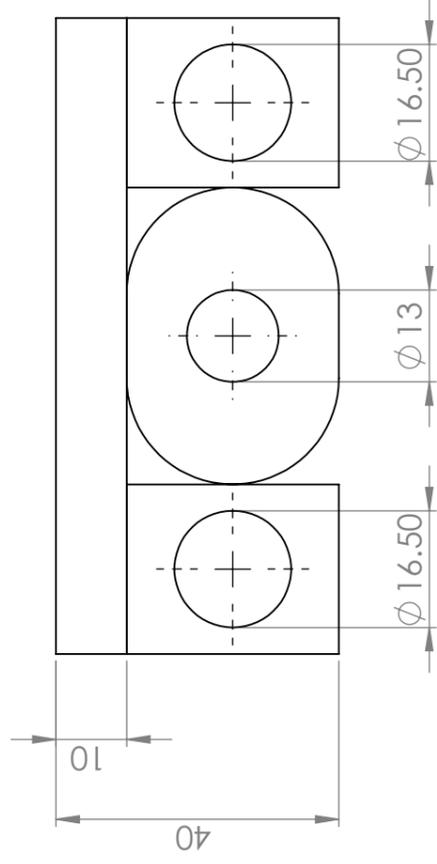
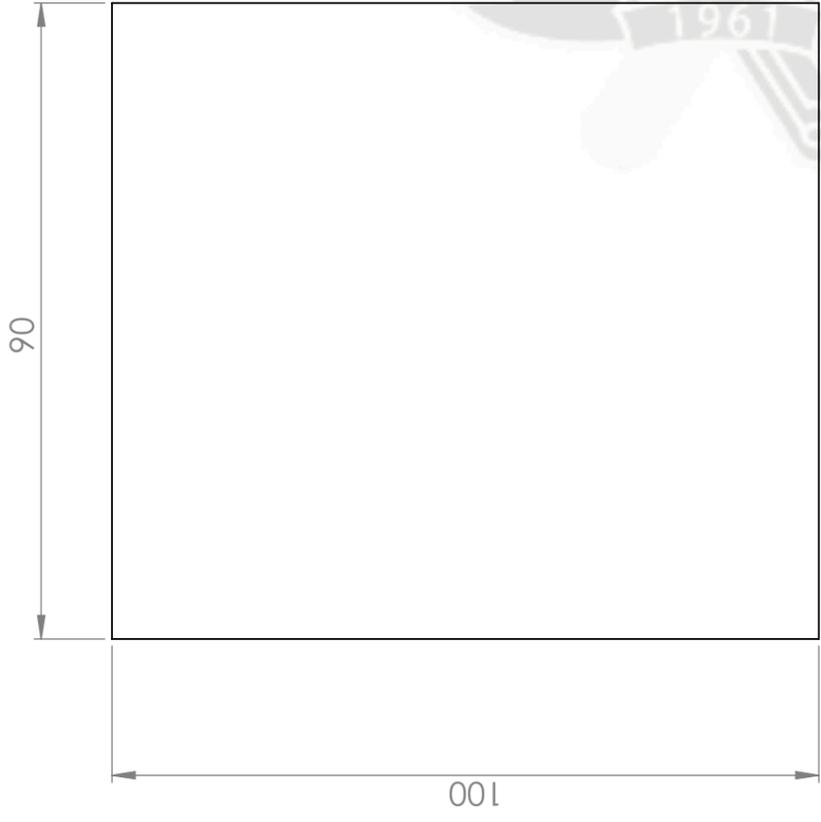
Corredera Horizontal

A3

ESCALA:1:5

COD: UCSM-CIMAY-04-02

HOJA 2 DE 2



SI NO SE INDICA LO CONTRARIO:  
LAS COTAS SE EXPRESAN EN MM  
ACABADO SUPERFICIAL:  
TOLERANCIAS:  
LINEAL:  
ANGULAR:

ACABADO:

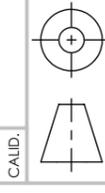
REBARBAR Y  
ROMPER ARISTAS  
VIVAS

TÍTULO:  
"CONTROL E IMPLEMENTACION DE UN  
MODULO DE ALMACENAJE VERTICAL A TRAVEZ  
DE UN PIC Y UN SISTEMA SCADA PARA LA  
OPTIMIZACION DE UN PROCESO DE  
ALMACENAJE INDUSTRIAL."

NOMBRE	FRMA	FECHA
S.M.A.		
C.B.C.		REVISION : 1
D.O.F.		
FABR.		
CALID.		MATERIAL:

NO CAMBIE LA ESCALA

"UNIVERSIDAD CATOLICA DE  
SANTA MARIA"  
PROGRAMA PROFESIONAL  
DE ING. MECANICA  
MECANICA ELECTRICA  
Y MECATRONICA



DIBUJO

Base móvil -Corredera Horizontal

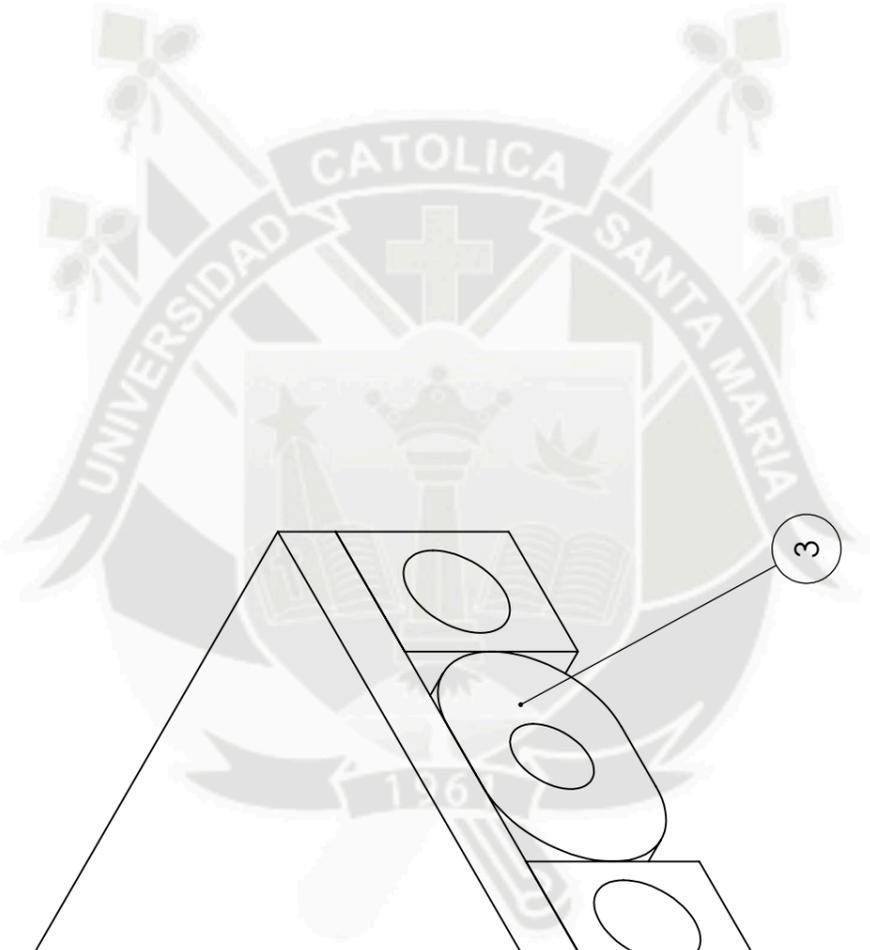
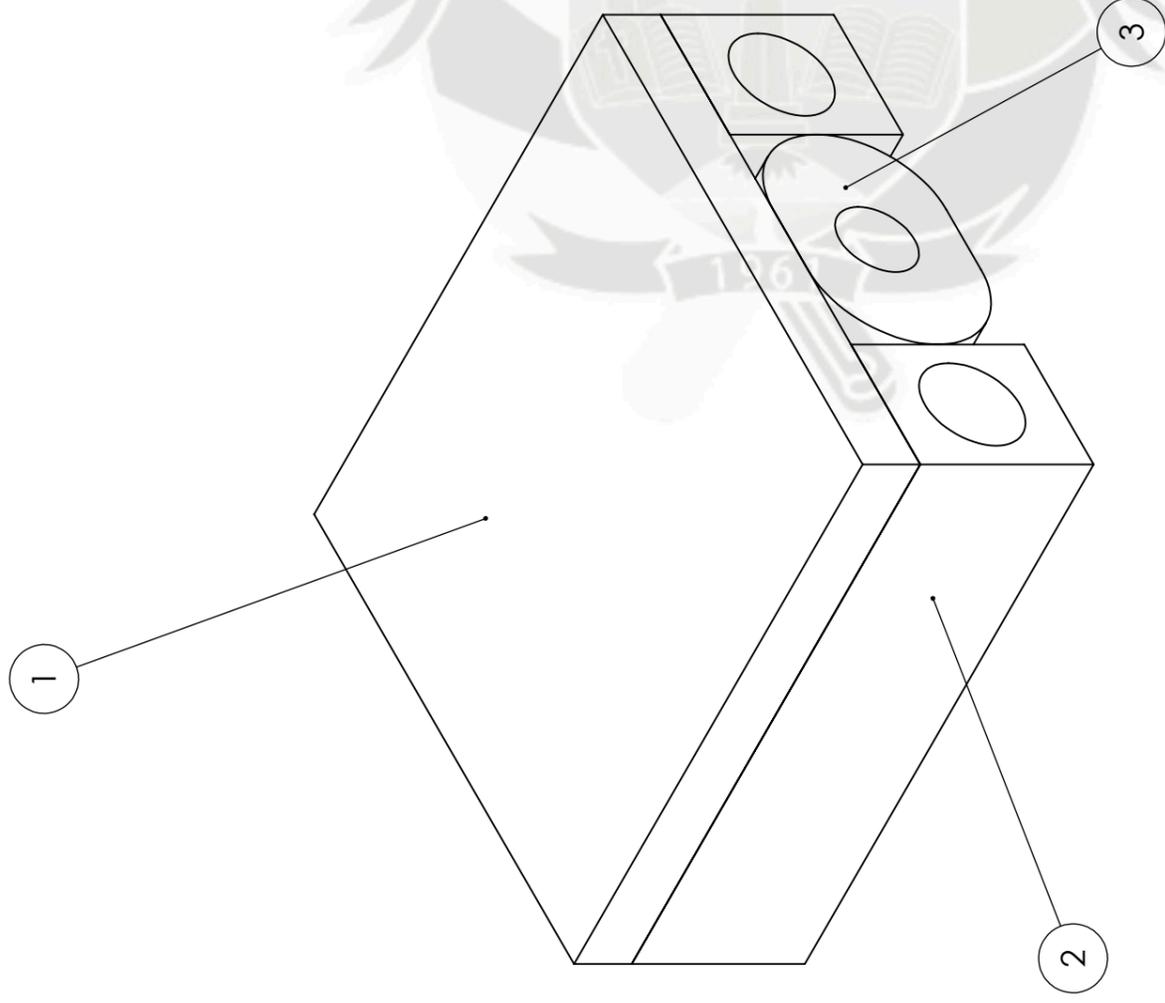
A3

ESCALA:1:1

COD: UC5M-CIMAV-03-01

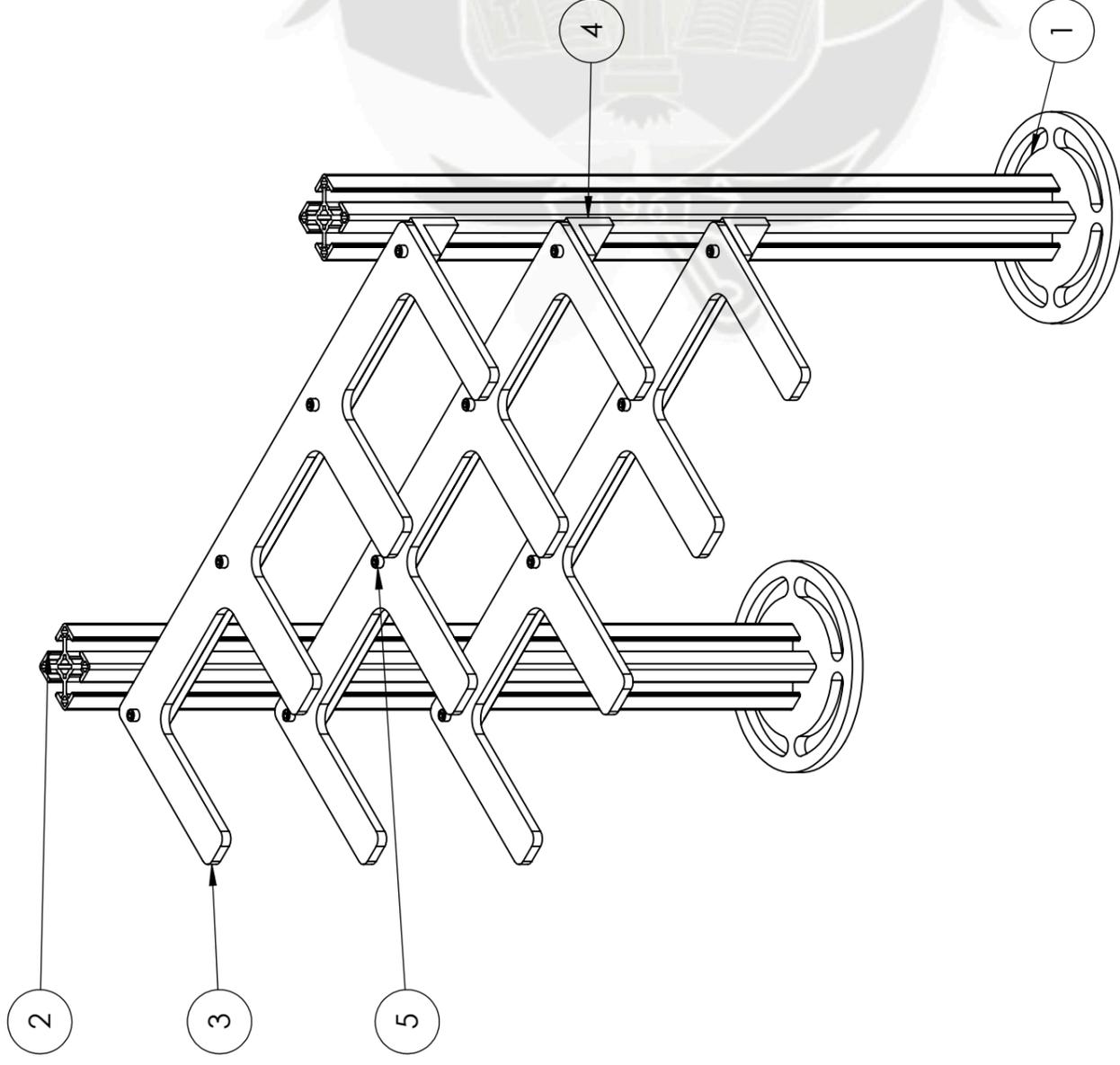
HOJA 1 DE 2

Nº	ELEMENTO	DESCRIPCION	CANT
1	BLOQUE SUPERIOR		1
2	BUJE	PARA LAS GUIAS	2
3	SISTEMA DE AVANCE	TRANSMISION POR MEDIO DEL ESPARRAGO	1



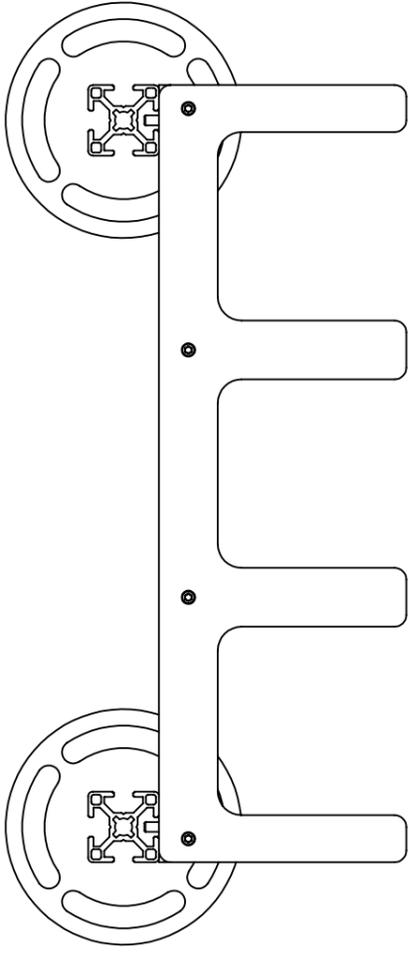
SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: ANGULAR:		ACABADO: ROMPER ARISTAS VIVAS		TÍTULO: <b>"IMPLEMENTACION DE UN MODULO AUTOMATIZADO DE ALMACENAJE VERTICAL INDUSTRIAL A ESCALA Y SU IMPACTO EN LA OPTIMIZACION DEL PROCESO DE ALMACENAJE INDUSTRIAL"</b>	
NOMBRE S.M.A. C.B.C. D.O.F. FABR. CALID.		FIRMA FECHA REVISION : 1 MATERIAL:		"UNIVERSIDAD CATOLICA DE SANTA MARIA" PROGRAMA PROFESIONAL DE ING. MECANICA MECANICA ELECTRICA Y MECATRONICA	
DIBUJ. VERIF. APROB. FABR. CALID.		NO CAMBIE LA ESCALA		DIBUJO <b>Base mobile-Corredera Horizontal</b>	
		COD: UCSM-CIMAY-03-02		ESCALA:1:1	
				A3	
				HOJA 2 DE 2	

N.º DE ELEMENTO	N.º DE PIEZA	DESCRIPCIÓN	CANT
1	Base		2
2	Perfiles soporte rexroth 30 x 30x420		2
3	SopORTE piezas		3
4	Perfil L		3
5	HX-SHCS 0.138-40x0.5625x0.5625-N		12
6	HX-SHCS 0.164-36x0.4375x0.4375-N		6

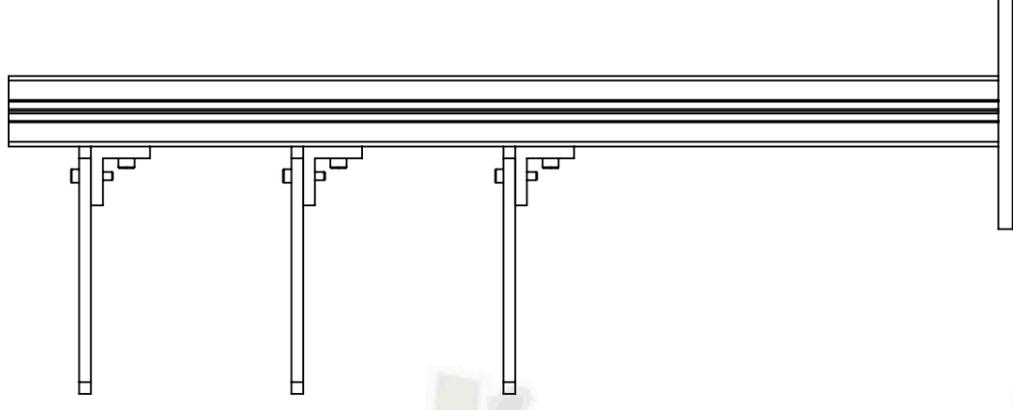


ESCALA 1 : 3

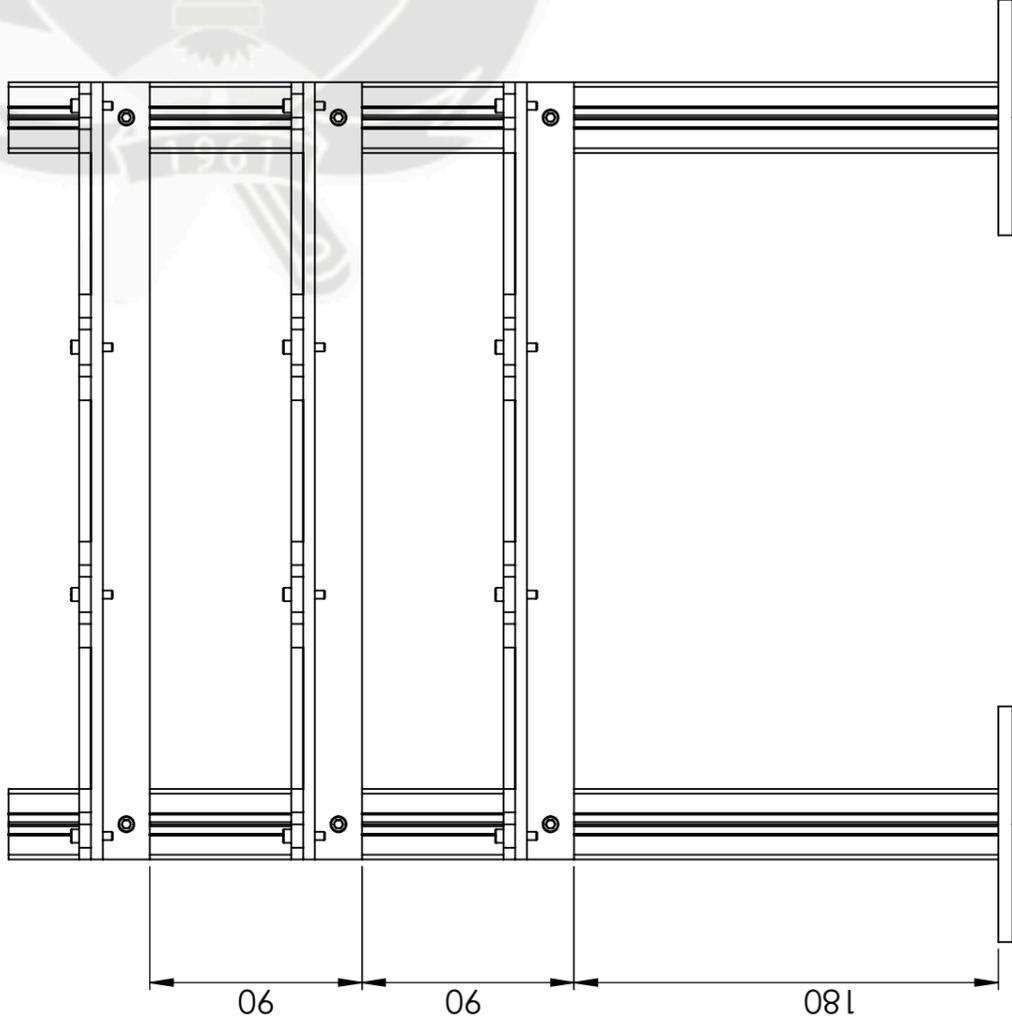
SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: LINEAL: ANGULAR:		ACABADO: REBARBAR Y ROMPER ARISTAS VIVAS	TÍTULO: "CONTROL E IMPLEMENTACION DE UN MODULO DE ALMACENAJE VERTICAL A TRAVEZ DE UN PLC Y UN SISTEMA SCADA PARA LA OPTIMIZACION DE UN PROCESO DE ALMACENAJE INDUSTRIAL."
NOMBRE: S.M.A.: C.B.C.: D.O.F.: FABR.: CALID.:	FIRMA: FECHA: REVISION: 1 MATERIAL:	NO CAMBIE LA ESCALA	"UNIVERSIDAD CATOLICA DE SANTA MARIA" PROGRAMA PROFESIONAL DE ING. MECANICA MECANICA ELECTRICA Y MECATRONICA
			DIBUJO acumulador A3
			ESCALA:1:5 COD: UCSM-CIMAY-02-01 HOJA 1 DE 3



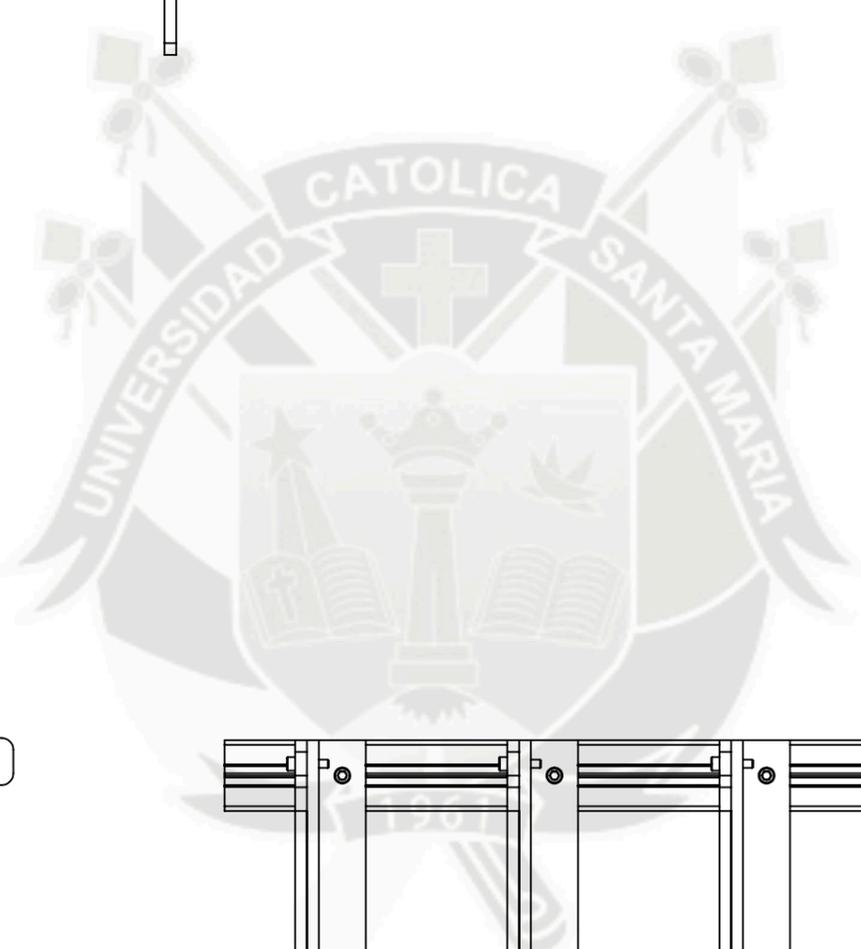
ESCALA 1 : 3



ESCALA 1 : 3



ESCALA 1 : 3

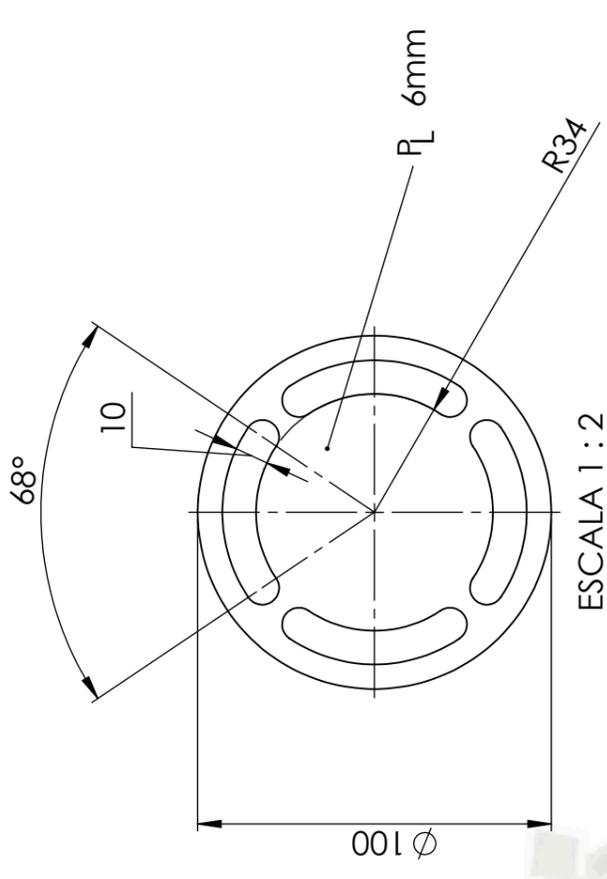
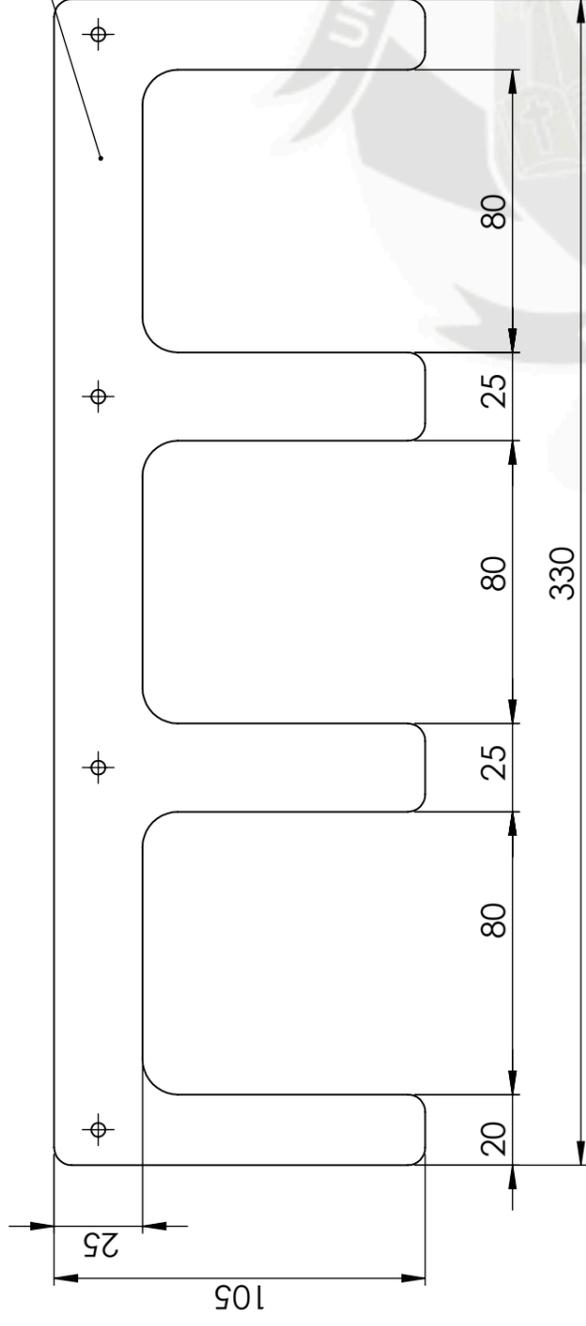


SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: LINEAL: ANGULAR:		ACABADO: REBARBAR Y ROMPER ARISTAS VIVAS		TÍTULO: "CONTROL E IMPLEMENTACION DE UN MODULO DE ALMACENAJE VERTICAL A TRAVEZ DE UN PIC Y UN SISTEMA SCADA PARA LA OPTIMIZACION DE UN PROCESO DE ALMACENAJE INDUSTRIAL."	
DIBUJ. VERIF. APROB. FABR. CALID.		NOMBRE S.M.A. C.B.C. D.O.F.		UNIVERSIDAD CATOLICA DE SANTA MARIA PROGRAMA PROFESIONAL DE ING. MECANICA MECANICA ELECTRICA Y MECATRONICA	
FRMA REVISION : MATERIAL:		FECHA NO CAMBIE LA ESCALA		DIBUJO A3	
		COD: UCSM-CIMAV-02-02		ESCALA:1:5 HOJA 2 DE 3	

acumulador

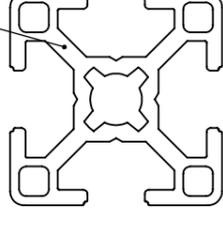
ESCALA 1 : 2

L 5mm



ESCALA 1 : 2

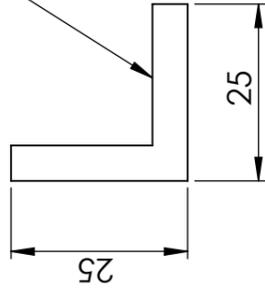
Rexroth 30 x 30 x 420mm



ESCALA 1 : 2



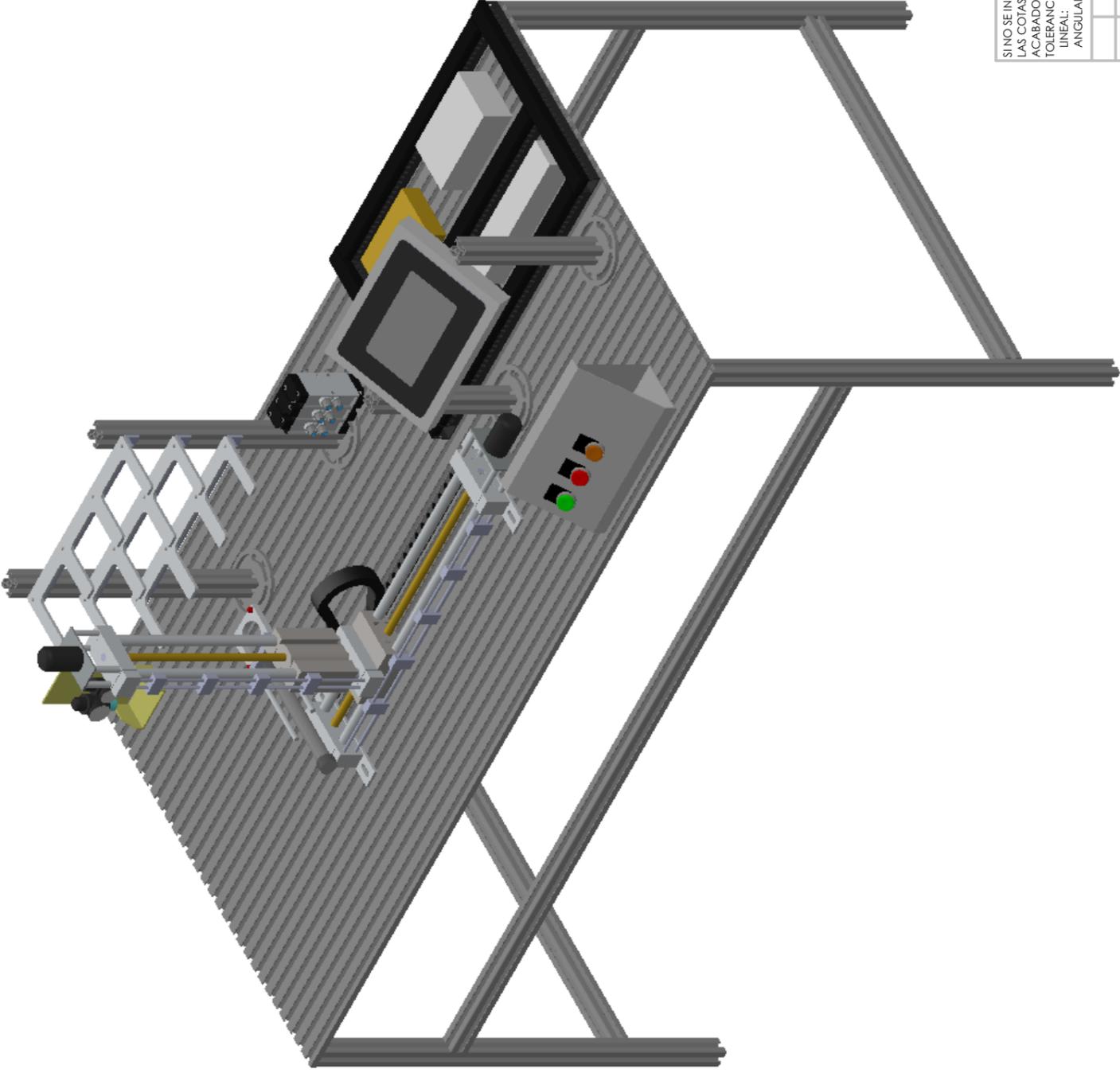
perfil L 25x 25 x 5mm



ESCALA 1 : 1

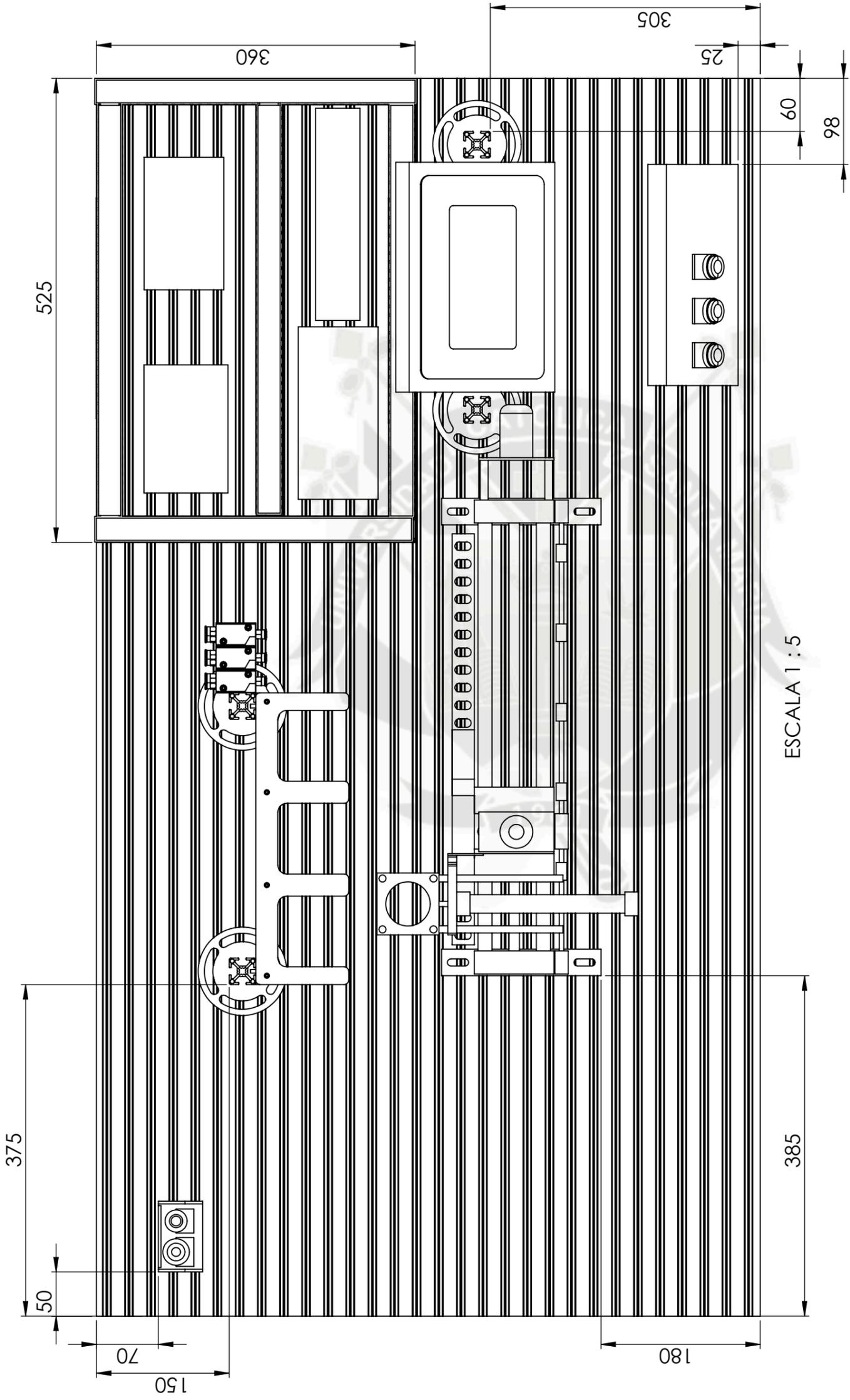
SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: LINEAL: ANGULAR:		ACABADO: REBARBAR Y ROMPER ARISTAS VIVAS		TÍTULO: "CONTROL E IMPLEMENTACION DE UN MODULO DE ALMACENAJE VERTICAL A TRAVEZ DE UN PLC Y UN SISTEMA SCADA PARA LA OPTIMIZACION DE UN PROCESO DE ALMACENAJE INDUSTRIAL."	
NOMBRE: S.M.A.: C.B.C.: D.O.F.: FABR.: CALID.:		FIRMA: REVISION: 1 MATERIAL:		UNIVERSIDAD CATOLICA DE SANTA MARIA PROGRAMA PROFESIONAL DE ING. MECANICA MECANICA ELECTRICA Y MECATRONICA	
DIBUJ.: VERIF.: APROB.: FABR.: CALID.:		NO CAMBIE LA ESCALA		DIBUJO acumulador	
COD: UCSM-CIMAV-02-03		ESCALA: 1:5		HOJA 3 DE 3	

# "CONTROL E IMPLEMENTACION DE UN MODULO DE ALMACENAJE VERTICAL A TRAVEZ DE UN PLC Y UN SISTEMA SCADA PARA LA OPTIMIZACION DE UN PROCESO DE ALMACENAJE INDUSTRIAL"

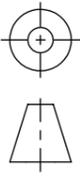


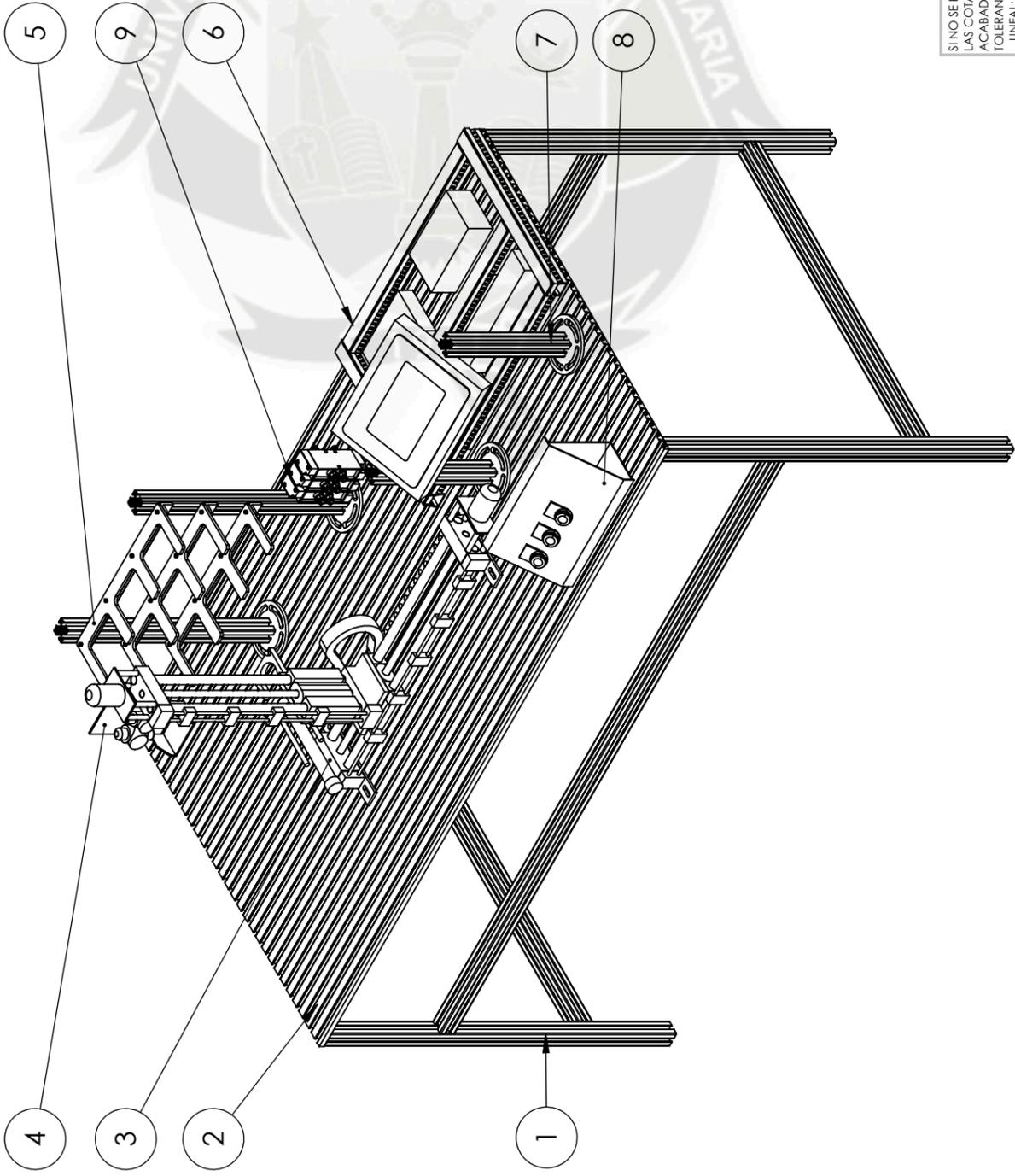
ESCALA 1 : 8

SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: LINEAL: ANGULAR:		ACABADO: ROMPER ARISTAS VIVAS		TÍTULO: "CONTROL E IMPLEMENTACION DE UN MODULO DE ALMACENAJE VERTICAL A TRAVEZ DE UN PLC Y UN SISTEMA SCADA PARA LA OPTIMIZACION DE UN PROCESO DE ALMACENAJE INDUSTRIAL"	
DIBUJ.	NOMBRE	FRMA	FECHA	"UNIVERSIDAD CATOLICA DE SANTA MARIA" PROGRAMA PROFESIONAL DE ING. MECANICA MECANICA ELECTRICA Y MECATRONICA	
VERIF.	S.M.A.				
APROB.	C.B.C.		REVISION : 1		
FABR.	D.O.F.				
CALID.			MATERIAL:		
				DIBUJO MODULO DE ALMACENAJE	
				ESCALA:1:20 COD: UCSM-CIMAY-01-01 HOJA 1 DE 3	



ESCALA 1 : 5

SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: LINEAL: ANGULAR:		ACABADO: REBARBAR Y ROMPER ARISTAS VIVAS		TÍTULO: "CONTROL E IMPLEMENTACION DE UN MODULO DE ALMACENAJE VERTICAL A TRAVEZ DE UN PIC Y UN SISTEMA SCADA PARA LA OPTIMIZACION DE UN PROCESO DE ALMACENAJE INDUSTRIAL."	
DIBUJ.	NOMBRE	FECHA	NO CAMBIE LA ESCALA	"UNIVERSIDAD CATOLICA DE SANTA MARIA" PROGRAMA PROFESIONAL DE ING. MECANICA MECANICA ELECTRICA Y MECATRONICA	
VERIF.	S.M.A.	REVISION : 1			
APROB.	C.B.C.			DIBUJO MODULO DE ALMACENAJE	
FABR.	D.O.F.			A3	
CALID.	MATERIAL:			ESCALA:1:20 HOJA 2 DE 3	
		COD: UCSM-CIMAY-01-02		ESCALA:1:20	



Nº	NOMBRE DE PIEZA	DESCRIPCIÓN	CANT
1	Estructura soporte	Rexroth 30x30	1
2	Base de trabajo	tipo Rexroth	1
3	Sistema X-Y-Z	Accionamiento lineal	1
4	LFR	Alimentador de aire	1
5	Rack	Almacenamiento de piezas	1
6	Tablero de control	Fuentes, Relees, Plc	1
7	Touch Pad	SIEMENS	1
8	Botonera de Control	Botones Start, Stop, Pausa y Parada de Emergencia	1
9	Válvulas neumáticas	JMFH-FESTO	3

ESCALA 1 : 10

SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: LINEAL: ANGULAR:		ACABADO: REBARBAR Y ROMPER ARISTAS VIVAS	TÍTULO: "CONTROL E IMPLEMENTACION DE UN MODULO DE ALMACENAJE VERTICAL A TRAVEZ DE UN PLC Y UN SISTEMA SCADA PARA LA OPTIMIZACION DE UN PROCESO DE ALMACENAJE INDUSTRIAL."
DIBUJ. VERIF. APROB. FABR. CALID.	NOMBRE S.M.A. C.B.C. D.O.F.	REVISION : 1 MATERIAL:	"UNIVERSIDAD CATOLICA DE SANTA MARIA" PROGRAMA PROFESIONAL DE ING. MECANICA MECANICA ELECTRONICA Y MECATRONICA
ESCALA: 1:20 COD: UCSM-CIMAV-01-03 DIBUJO: MODULO DE ALMACENAJE			A3



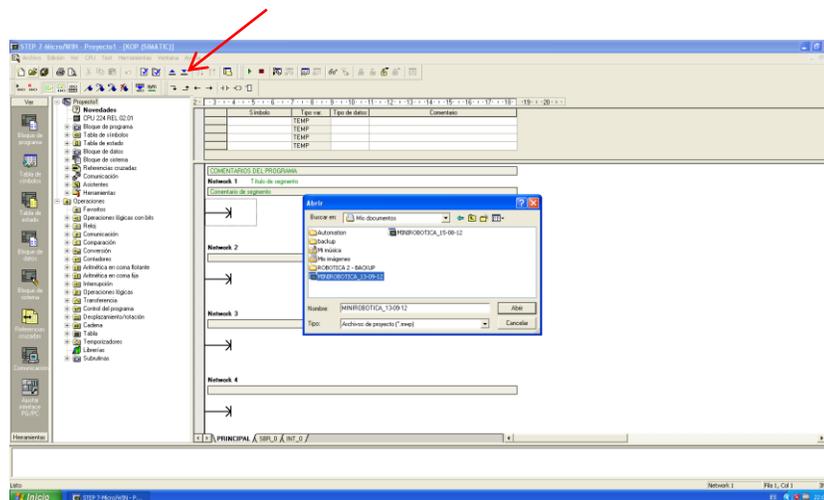
# ANEXO 5

## PROGRAMA LADDER EN STEP 7

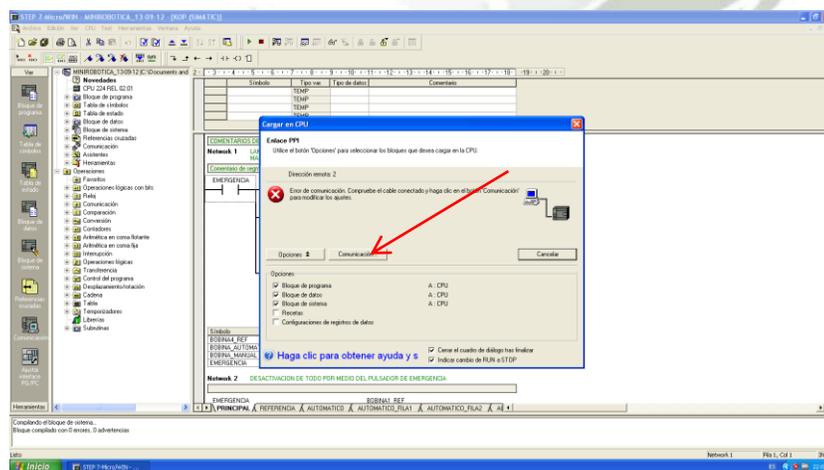


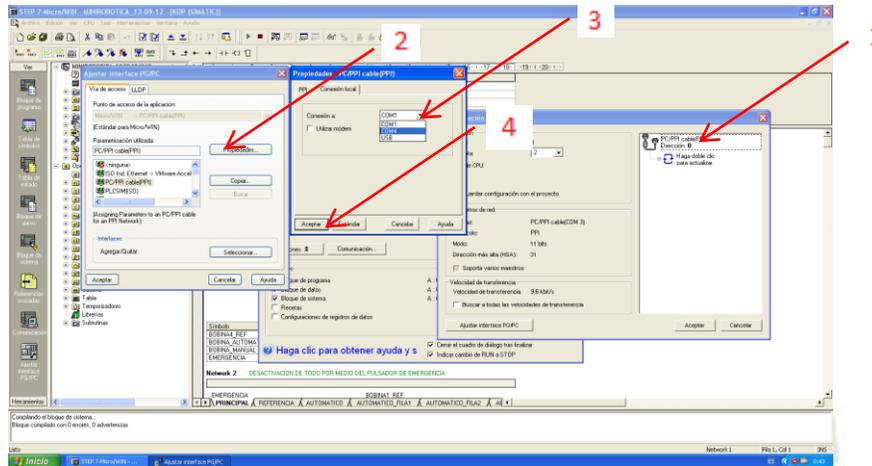
## DESCARGA DE PROGRAMA EN S7-200

Abrir el programa Microwin 4.0, luego abrir el archivo del **Programa Inicial (Manipulador Robótico 2)** o **Programa mejorado (Manipulador Robótico 1)**.

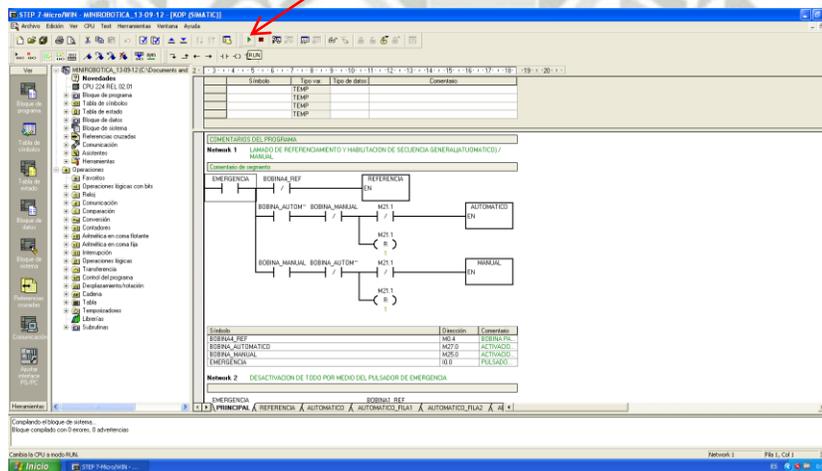


Se presentara el proyecto seleccionado, luego descargar el programa en el PLC dando click en **Cargar en CPU**, y configurar el puerto para el cable PPI.





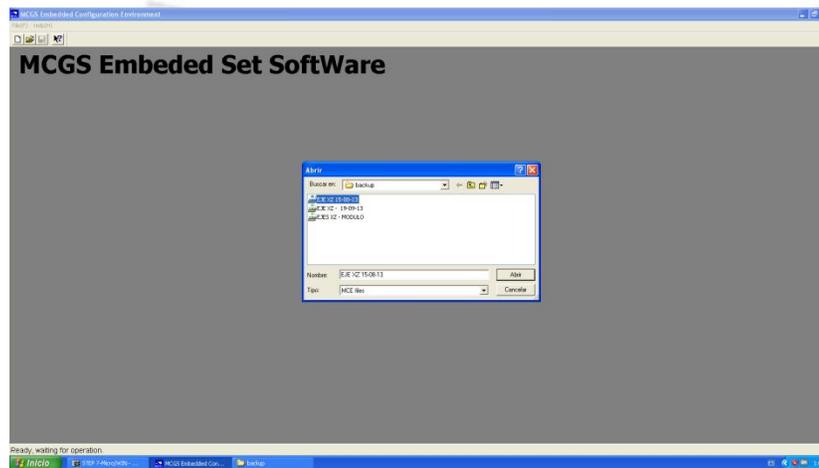
Luego de haber configurado el puerto de comunicación para PPI, dar click en **Cargar a PC**. Por ultimo dar click en **RUN** para iniciar el programa.



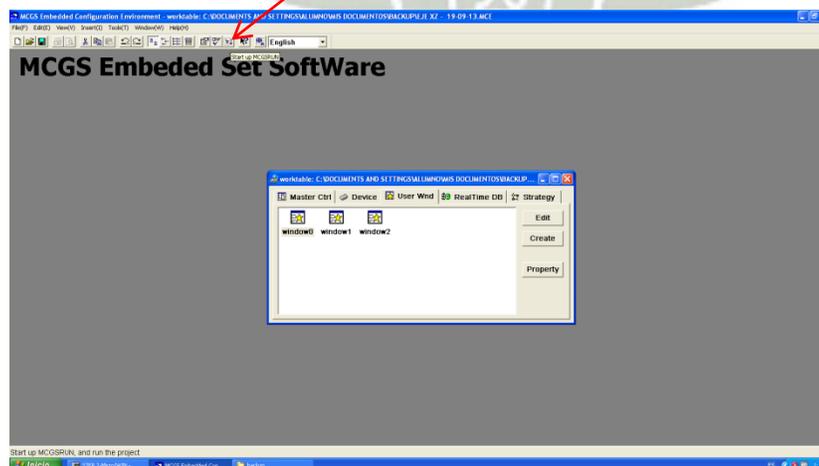
## DESCARGA DE PROGRAMA EN HMI MCGS

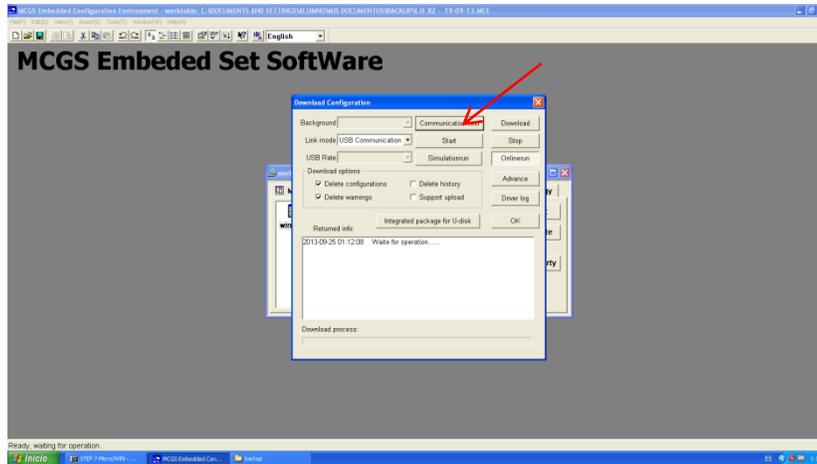
Paso 1:

Abrir el programa MCGS, luego abrir el archivo del **Programa Inicial (Manipulador Robótico 2)** o **Programa mejorado (Manipulador Robótico 1)**.

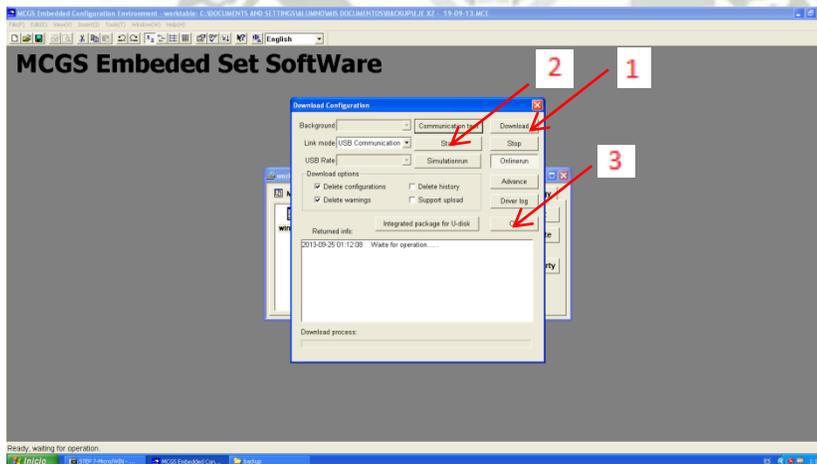


Se presentara el proyecto seleccionado, luego descargar el programa en el HMI dando click en **StarUp MCGSRUN**, y dar click en **Communication test** para verificar la comunicación desde el puerto USB y el HMI.





Luego de haber confirmado la comunicación entre PC y HMI, dar click en **Donwload** para descargar el programa y por último en **Start** para dar inicio al programa.





## ANEXO 6

### GUÍAS DE PRÁCTICA (04) - MÓDULO DE ALMACENAJE



## SIMATIC

### Sistemas de automatización S7-200

#### Manual de sistema

Referencia del manual:  
**6ES7298-8FA20-8DH0**

03/99

Edición 01

Prólogo, Índice

Introducción a los Micro-PLCs S7-200	<b>1</b>
Instalar un Micro-PLC S7-200	<b>2</b>
Instalación y configuración del sistema de automatización S7-200	<b>3</b>
Nociones básicas para programar una CPU S7-200	<b>4</b>
Memoria de la CPU: tipos de datos y direccionamiento	<b>5</b>
Configurar las entradas y salidas	<b>6</b>
Configurar el hardware para la comunicación en redes	<b>7</b>
Convenciones para las operaciones S7-200	<b>8</b>
Operaciones SIMATIC	<b>9</b>
Operaciones IEC 1131-3	<b>10</b>
Datos técnicos S7-200	<b>A</b>
Códigos de error	<b>B</b>
Marcas especiales (SM)	<b>C</b>
Eliminar errores	<b>D</b>
Números de referencia	<b>E</b>
Tiempos de ejecución de las operaciones AWL	<b>F</b>
Breviario del S7-200	<b>G</b>
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, o bien 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 a lo descrito en 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 sólo se podrán utilizar para los casos de aplicación previstos en el catálogo y en la descripción técnica, y sólo con los equipos y componentes de proveniencia tercera recomendados y homologados por Siemens.

El funcionamiento correcto y seguro del producto presupone un transporte, un almacenamiento, una instalación y un montaje conforme a las prácticas de la buena ingeniería, así como un manejo y un mantenimiento rigurosos.

## Marcas registradas

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

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

## Copyright © Siemens AG 1999 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  
Geschäftsgebiet Industrie-Automatisierungssysteme  
Postfach 4848, D-90327 Nuernberg

Siemens Aktiengesellschaft

## 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 1999  
Sujeto a cambios sin previo aviso.

6ES7298-8FA20-8DH0

## Prólogo

### Finalidad del manual

La gama S7-200 comprende diversos sistemas de automatización pequeños (Micro-PLCs) que se pueden utilizar para numerosas tareas. Gracias a su diseño compacto, su bajo costo y su amplio juego de operaciones, los sistemas de automatización S7-200 son idóneos para controlar tareas sencillas. La gran variedad de modelos de CPUs y el software de programación basado en Windows ofrecen la flexibilidad necesaria para solucionar las tareas de automatización.

La gama S7-200 se ha rediseñado, ofreciendo ahora productos más pequeños, más rápidos y con más funciones. Los nuevos productos S7-200 han de sustituir a los anteriores.

El *Manual del sistema de automatización S7-200* contiene informaciones acerca de cómo instalar y programar los nuevos Micro-PLCs S7-200, incluyendo los siguientes temas:

- Montaje y cableado.
- Descripción de las funciones de la CPU, tipos de datos y modos de direccionamiento, ciclo de la CPU, protección con contraseña y comunicación en redes.
- Datos técnicos.
- Descripciones y ejemplos de las operaciones de programación SIMATIC e IEC 1131-3.
- Tiempos de ejecución típicos de las operaciones SIMATIC AWL.

### Destinatarios

Este manual va dirigido a ingenieros, programadores, técnicos de instalación y electricistas que dispongan de conocimientos básicos sobre los sistemas de automatización.

## Ámbito de validez

Las informaciones contenidas en este manual son aplicables en particular a los siguientes productos:

- CPUs S7-200: CPU 221, CPU 222 y CPU 224
- STEP 7-Micro/WIN 32 (versión 3.0), un paquete de software de programación (32 bits) para los entornos Windows 95, Windows 98 y Windows NT.

## Homologaciones

La gama S7-200 SIMATIC cumple las siguientes normas:

- Comunidad Europea (CE), Directiva de Baja Tensión 73/23/EEC
- Directiva EMC de la Comunidad Europea (CE) 89/336/CEE
- Underwriters Laboratories, Inc.: UL 508, homologado (Industrial Control Equipment)
- Canadian Standards Association: CSA C22.2 n° 142, certificado (Process Control Equipment)
- Factory Mutual Research: FM clase I, categoría 2, grupo A, B, C, y D Hazardous Locations, T4A

Consulte el Anexo A para obtener más información sobre el cumplimiento de normas.

## Informaciones adicionales

Los siguientes productos y manuales proporcionan informaciones más detalladas acerca de temas especiales:

- STEP 7-Micro/WIN 32 CD/disquetes: incorpora la Ayuda en pantalla y la *Guía de iniciación a STEP 7-Micro/WIN* (un manual electrónico imprimible).
- Estándar PROFIBUS (Process Field Bus) (EN 50170): describe el protocolo estándar para la comunicación DP de las CPUs S7-200.
- *Manual del usuario del visualizador de textos TD 200*: describe cómo instalar y utilizar el TD 200 con un sistema de automatización S7-200.

## Cómo utilizar el manual

Si es la primera vez que trabaja con los Micro-PLCs S7-200, es recomendable que lea todo el *Manual del sistema de automatización S7-200*. Si ya dispone de los conocimientos necesarios, consulte el índice para encontrar la información que precise.

El *Manual del sistema de automatización S7-200* se divide en los siguientes capítulos:

- El capítulo 1 (“Introducción a los Micro-PLCs S7-200”) ofrece una panorámica de algunas de las funciones de los sistemas de automatización.
- El capítulo 2 (“Instalar un Micro-PLC S7-200”) informa acerca de los procedimientos, dimensiones y reglas básicas para instalar las CPUs y los módulos de ampliación.
- El capítulo 3 (“Instalación y configuración del sistema de programación S7-200”) describe cómo instalar y configurar un sistema de programación S7-200.
- El capítulo LEERER MERKER (“Nociones básicas para programar una CPU S7-200”), el capítulo 5 (“Memoria de la CPU: tipos de datos y direccionamiento”) y el capítulo 6 (“Configurar las entradas y salidas”) proporcionan información acerca de cómo la CPU S7-200 procesa los datos y ejecuta el programa de usuario.
- El capítulo 7 (“Configurar el hardware para la comunicación en redes”) informa acerca de cómo instalar y desinstalar equipos de hardware de comunicación y sobre cómo conectar la CPU S7-200 a diferentes tipos de redes.
- El capítulo 8 (“Convenciones para las operaciones S7-200”) ofrece una panorámica de los diferentes lenguajes de programación y de la correspondiente terminología.
- El capítulo 9 contiene descripciones y ejemplos de las operaciones KOP, FUP y AWL SIMATIC.
- El capítulo 10 contiene descripciones y ejemplos de las operaciones LD y FBD IEC-1131.

En los anexos se proporcionan informaciones adicionales (datos técnicos de los equipos, descripciones de los códigos de error, eliminación de errores y tiempos de ejecución de las operaciones AWL).

## Asistencia complementaria

Si tiene preguntas técnicas, si necesita información sobre los cursos de entrenamiento o para efectuar pedidos, diríjase por favor a su representante de Siemens más próximo.

En las siguientes direcciones de Internet puede obtener informaciones acerca de los productos y servicios de Siemens, asistencia técnica y respuestas a las preguntas más frecuentes:

- |   |  |
|---|--|
| <a href="http://www.ad.siemens.de">http://www.ad.siemens.de</a>           | (informaciones generales sobre Siemens)    |
| <a href="http://www.siemens.com/s7-200">http://www.siemens.com/s7-200</a> | (informaciones sobre los productos S7-200) |

## Indice

<b>Prólogo</b> .....	<b>iii</b>
<b>Indice</b> .....	<b>vii</b>
<b>1 Introducción a los Micro-PLCs S7-200</b> .....	<b>1-1</b>
1.1 Comparativa de los diversos Micro-PLCs S7-200 .....	1-2
1.2 Principales componentes de un Micro-PLC S7-200 .....	1-4
<b>2 Instalar un Micro-PLC S7-200</b> .....	<b>2-1</b>
2.1 Preparar el montaje .....	2-2
2.2 Montar y desmontar un Micro-PLC S7-200 o un módulo de ampliación .....	2-6
2.3 Instalar el cableado de campo .....	2-9
2.4 Utilizar circuitos de supresión .....	2-16
2.5 Alimentación de corriente .....	2-18
<b>3 Instalación y configuración del sistema de automatización S7-200</b> .....	<b>3-1</b>
3.1 Resumen breve .....	3-2
3.2 Cómo instalar STEP 7-Micro/WIN 32 .....	3-3
3.3 Cómo configurar la comunicación utilizando el cable PC/PPI .....	3-5
3.4 Cómo establecer la comunicación con la CPU S7-200 .....	3-9
3.5 Cómo cambiar los parámetros de comunicación de la CPU .....	3-10
<b>4 Nociones básicas para programar una CPU S7-200</b> .....	<b>4-1</b>
4.1 Crear una solución de automatización con un Micro-PLC .....	4-2
4.2 Programas S7-200 .....	4-5
4.3 Lenguajes y editores de programación S7-200 .....	4-6
4.4 Diferencias entre las operaciones SIMATIC e IEC 1131-3 .....	4-10
4.5 Elementos básicos para estructurar un programa .....	4-18
4.6 El ciclo de la CPU .....	4-22
4.7 Ajustar el modo de operación de la CPU .....	4-25
4.8 Definir una contraseña para la CPU .....	4-27
4.9 Comprobar y observar el programa .....	4-30
4.10 Eliminar errores de las CPUs S7-200 .....	4-36

<b>5</b>	<b>Memoria de la CPU: tipos de datos y direccionamiento</b> .....	<b>5-1</b>
5.1	Direccionamiento directo de las áreas de memoria de la CPU .....	5-2
5.2	Direccionamiento indirecto (SIMATIC) de las áreas de memoria de la CPU ....	5-13
5.3	Respaldo de datos en la CPU S7-200 .....	5-15
5.4	Utilizar el programa para guardar datos en la memoria no volátil .....	5-20
5.5	Guardar el programa en un cartucho de memoria .....	5-22
<b>6</b>	<b>Configurar las entradas y salidas</b> .....	<b>6-1</b>
6.1	Entradas y salidas integradas y adicionales .....	6-2
6.2	Utilizar filtros de entrada para suprimir interferencias .....	6-4
6.3	Capturar impulsos .....	6-5
6.4	Configurar los estados de señal de las salidas .....	6-8
6.5	Filtrar entradas analógicas .....	6-9
6.6	Entradas y salidas rápidas .....	6-10
6.7	Potenciómetros analógicos .....	6-13
<b>7</b>	<b>Configurar el hardware para la comunicación en redes</b> .....	<b>7-1</b>
7.1	Opciones de comunicación .....	7-2
7.2	Instalar y desinstalar interfaces de comunicación .....	7-7
7.3	Seleccionar y cambiar parámetros .....	7-9
7.4	Comunicación con módems .....	7-16
7.5	Redes y protocolos .....	7-27
7.6	Componentes de redes .....	7-31
7.7	Utilizar el cable PC/PPI con otros dispositivos y en modo Freeport .....	7-35
7.8	Rendimiento de la red .....	7-41
<b>8</b>	<b>Convenciones para las operaciones S7-200</b> .....	<b>8-1</b>
8.1	Conceptos y convenciones para programar con STEP 7-Micro/WIN 32 .....	8-2
8.2	Márgenes válidos para las CPUs S7-200 .....	8-7

<b>9</b>	<b>Operaciones SIMATIC</b> .....	<b>9-1</b>
9.1	Operaciones lógicas con bits (SIMATIC) .....	9-2
9.2	Operaciones de comparación (SIMATIC) .....	9-10
9.3	Operaciones de temporización (SIMATIC) .....	9-15
9.4	Operaciones con contadores (SIMATIC) .....	9-23
9.5	Operaciones con contadores rápidos (SIMATIC) .....	9-27
9.6	Operaciones de salida de impulsos (SIMATIC) .....	9-49
9.7	Operaciones de reloj (SIMATIC) .....	9-70
9.8	Operaciones aritméticas con enteros (SIMATIC) .....	9-72
9.9	Operaciones aritméticas con números reales (SIMATIC) .....	9-81
9.10	Operaciones de transferencia (SIMATIC) .....	9-99
9.11	Operaciones de tabla (SIMATIC) .....	9-104
9.12	Operaciones lógicas (SIMATIC) .....	9-110
9.13	Operaciones de desplazamiento y rotación (SIMATIC) .....	9-116
9.14	Operaciones de conversión (SIMATIC) .....	9-126
9.15	Operaciones de control del programa (SIMATIC) .....	9-141
9.16	Operaciones de interrupción y comunicación (SIMATIC) .....	9-165
9.17	Operaciones lógicas de pilas (SIMATIC) .....	9-192
<b>10</b>	<b>Operaciones IEC 1131-3</b> .....	<b>10-1</b>
10.1	Operaciones lógicas con bits (IEC) .....	10-2
10.2	Operaciones de comparación (IEC) .....	10-7
10.3	Operaciones de temporización (IEC) .....	10-11
10.4	Operaciones con contadores (IEC) .....	10-15
10.5	Operaciones aritméticas (IEC) .....	10-19
10.6	Operaciones de transferencia (IEC) .....	10-24
10.7	Operaciones lógicas (IEC) .....	10-26
10.8	Operaciones de desplazamiento y rotación (IEC) .....	10-29
10.9	Operaciones de conversión (IEC) .....	10-32

<b>A</b>	<b>Datos técnicos S7-200</b> .....	<b>A-1</b>
A.1	Datos técnicos generales .....	A-2
A.2	Datos técnicos de la CPU 221 .....	A-6
A.3	Datos técnicos de la CPU 222 .....	A-11
A.4	Datos técnicos de la CPU 224 .....	A-16
A.5	Datos técnicos del módulo de ampliación EM221 de entradas digitales .....	A-21
A.6	Datos técnicos de los módulos de ampliación EM222 de salidas digitales .....	A-23
A.7	Datos técnicos de los módulos de ampliación EM223, 8 entradas digitales/8 salidas digitales .....	A-25
A.8	Cartuchos opcionales .....	A-28
A.9	Cable de módulo de ampliación .....	A-29
A.10	Cable PC/PPI .....	A-30
<b>B</b>	<b>Códigos de error</b> .....	<b>B-1</b>
B.1	Códigos de errores fatales y mensajes .....	B-2
B.2	Errores de programación del tiempo de ejecución .....	B-3
B.3	Violación de reglas de compilación .....	B-4
<b>C</b>	<b>Marcas especiales (SM)</b> .....	<b>C-1</b>
<b>D</b>	<b>Eliminar errores</b> .....	<b>D-1</b>
<b>E</b>	<b>Números de referencia</b> .....	<b>E-1</b>
<b>F</b>	<b>Tiempos de ejecución de las operaciones AWL</b> .....	<b>F-1</b>
<b>G</b>	<b>Breviario del S7-200</b> .....	<b>G-1</b>
	<b>Índice alfabético</b> .....	<b>Índice alfabético-1</b>

# 1

## Introducción a los Micro-PLCs S7-200

La gama S7-200 comprende diversos sistemas de automatización pequeños (Micro-PLCs) que se pueden utilizar para numerosas tareas. La figura 1-1 muestra un Micro-PLC S7-200. Gracias a su diseño compacto, su capacidad de ampliación, su bajo costo y su amplio juego de operaciones, los Micro-PLCs S7-200 son especialmente apropiados para solucionar tareas de automatización sencillas. Además, los diversos tamaños y fuentes de alimentación de las CPUs ofrecen la flexibilidad necesaria para solucionar las tareas de automatización.

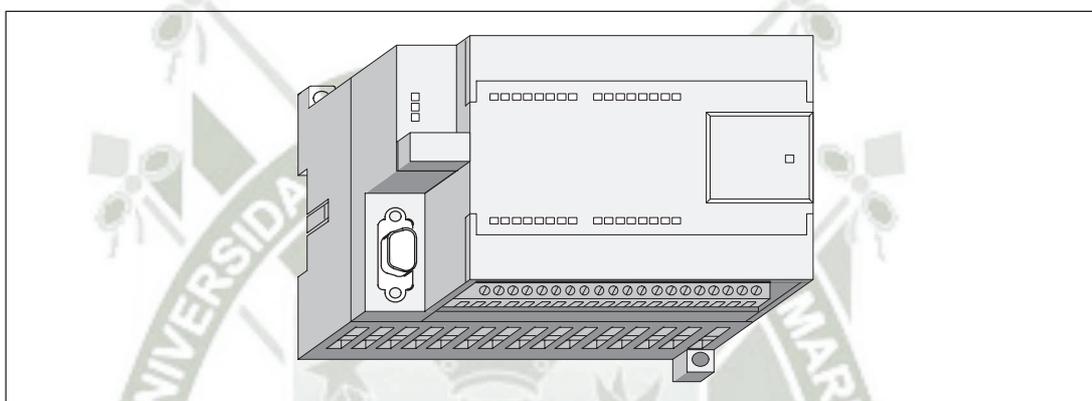


Figura 1-1 Micro-PLC S7-200

### Índice del capítulo

Apartado	Descripción	Página
1.1	Comparativa de los diversos Micro-PLCs S7-200	1-2
1.2	Principales componentes de un Micro-PLC S7-200	1-4

## 1.1 Comparativa de los diversos Micro-PLCs S7-200

### Equipos necesarios

La figura 1-2 muestra los componentes básicos de un sistema Micro-PLC S7-200, incluyendo una CPU S7-200, un PC, el software de programación STEP 7-Micro/WIN 32 (versión 3.0) y un cable de comunicación.

Si desea utilizar un PC, debe disponer de uno de los siguientes equipos adicionales:

- Un cable PC/PPI.
- Un procesador de comunicaciones (CP) y un cable de interface multipunto (MPI)
- Una tarjeta de interface multipunto (MPI). El cable de comunicación se suministra junto con la tarjeta MPI.

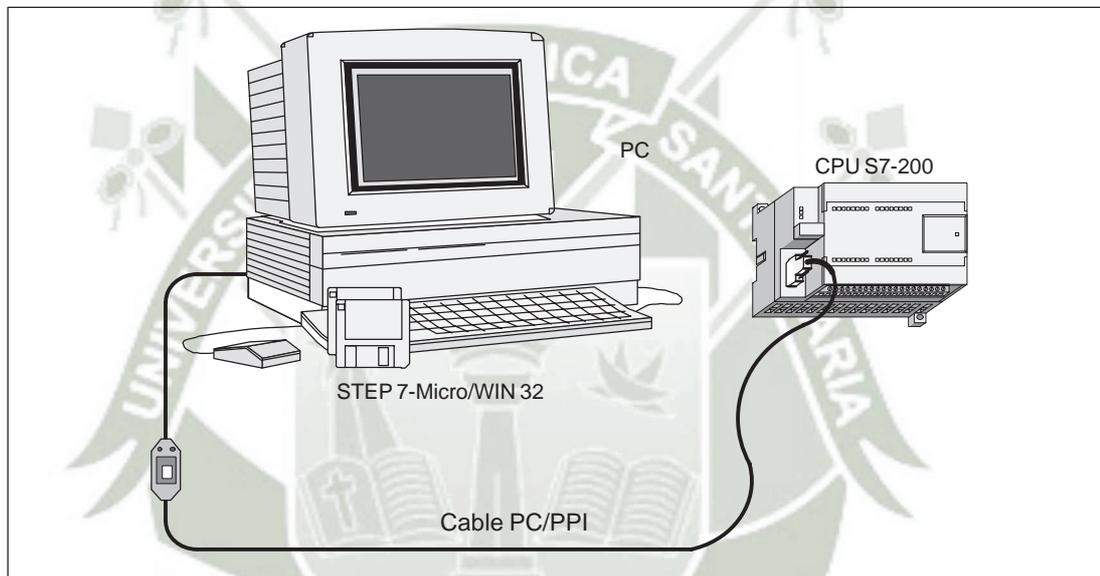


Figura 1-2 Componentes de un Micro-PLC S7-200

### Capacidad de las CPUs S7-200

La gama S7-200 comprende diversas CPUs. Por lo tanto, se dispone de una gran variedad de funciones que permiten diseñar soluciones de automatización a un precio razonable. La tabla 1-1 resume las principales funciones de las CPUs S7-200.

Tabla 1-1 Resumen de las CPUs S7-200

Función	CPU 221	CPU 222	CPU 224
<b>Tamaño físico</b>	90 mm x 80 mm x 62 mm	90 mm x 80 mm x 62 mm	120,5 mm x 80 mm x 62 mm
<b>Memoria</b>			
Programa	2048 palabras	2048 palabras	4096 palabras
Datos de usuario	1024 palabras	1024 palabras	2560 palabras
Tipo de memoria	EEPROM	EEPROM	EEPROM
Cartucho de memoria	EEPROM	EEPROM	EEPROM
Respaldo (condensador de alto rendimiento)	50 horas (típ.)	50 horas (típ.)	190 horas (típ.)
<b>E/S integradas</b>			
E/S integradas	6 DI/ 4 DQ	8 DI / 6 DQ	14 DI / 10 DQ
Número de módulos de ampliación	ninguno	2 módulos	7 módulos
<b>E/S (total)</b>			
Tamaño de la imagen de E/S digitales	256 (128 entradas/128 salidas)	256 (128 entradas/128 salidas)	256 (128 entradas/128 salidas)
Tamaño físico de E/S digitales	10	62	128
Tamaño de la imagen de E/S analógicas	ninguno	16 DI/16 DQ	16 DI/16 DQ
Tamaño físico de E/S analógicas	ninguno	12 DI/10 DQ	12 DI/10 DQ
<b>Operaciones</b>			
Velocidad de ejecución booleana	0,37 $\mu$ s/operación	0,37 $\mu$ s/operación	0,37 $\mu$ s/operación
Relés internos	256	256	256
Contadores/temporizadores	256/256	256/256	256/256
Relés de control secuencial	256	256	256
Bucles FOR/NEXT	Sí	Sí	Sí
Aritmética en coma fija (+ - * /)	Sí	Sí	Sí
Aritmética en coma flotante (+ - * /)	Sí	Sí	Sí
<b>Funciones adicionales</b>			
Contadores rápidos	4 (20 KHz)	4 (20 KHz)	6 (20 KHz)
Potenciómetros analógicos	1	1	2
Salidas de impulsos	2 (20 KHz, sólo DC)	2 (20 KHz, sólo DC)	2 (20 KHz, sólo DC)
Interrupciones de comunicación	1 transmisión/ 2 recepción	1 transmisión/ 2 recepción	1 transmisión/ 2 recepción
Interrupciones temporizadas	2 (1 ms a 255 ms)	2 (1 ms a 255 ms)	2 (1 ms a 255 ms)
Entradas de interrupción de hardware	4	4	4
Reloj de tiempo real	Sí (cartucho)	Sí (cartucho)	Sí (incorporado)
Protección con contraseña	Sí	Sí	Sí
<b>Comunicación</b>			
Número de puertos de comunicación:	1 (RS-485)	1 (RS-485)	1 (RS-485)
Protocolos asistidos Puerto 0:	PPI, MPI esclavo, Freeport	PPI, MPI esclavo, Freeport	PPI, MPI esclavo, Freeport
PROFIBUS punto a punto	(NETR/NETW)	(NETR/NETW)	(NETR/NETW)

## 1.2 Principales componentes de un Micro-PLC S7-200

Un Micro-PLC S7-200 puede comprender una CPU S7-200 sola o conectada a diversos módulos de ampliación opcionales.

### CPU S7-200

La CPU S7-200 es un equipo autónomo compacto que incorpora una unidad central de procesamiento (CPU), una fuente de alimentación, así como entradas y salidas digitales.

- La CPU ejecuta el programa y almacena los datos para la tarea de automatización o el proceso.
- Utilizando módulos de ampliación se pueden agregar entradas y salidas (E/S) adicionales a la CPU hasta el tamaño físico máximo indicado en la tabla 1-1.
- La fuente de alimentación suministra corriente a la unidad central y a los módulos de ampliación conectados.
- El sistema se controla mediante entradas y salidas (E/S). Las entradas vigilan las señales de los dispositivos de campo (p.ej. sensores e interruptores), mientras que las salidas supervisan las bombas, motores u otros aparatos del proceso.
- El puerto de comunicación permite conectar la CPU a una unidad de programación o a otros dispositivos que intervengan en el proceso.
- Los diodos luminosos indican el modo de operación de la CPU (RUN o STOP), el estado de las entradas y salidas integradas, así como los posibles fallos del sistema que se hayan detectado.
- Algunas CPUs tienen un reloj de tiempo real incorporado, en tanto que otras necesitan un cartucho de reloj de tiempo real.
- Un cartucho enchufable EEPROM en serie permite almacenar programas de la CPU y transferir programas de una CPU a otra.
- Un cartucho enchufable de pila permite prolongar el respaldo de los datos en la RAM.

La figura 1-3 muestra una CPU S7-200.

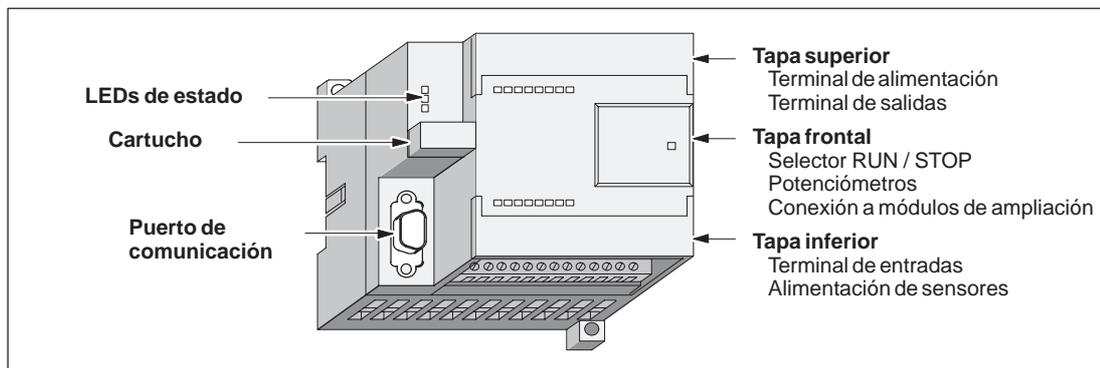


Figura 1-3 CPU S7-200

### Módulos de ampliación

La CPU S7-200 dispone de un número determinado de entradas y salidas integradas. Conectando un módulo de ampliación se dispondrá de más entradas y salidas (v. fig. 1-4).

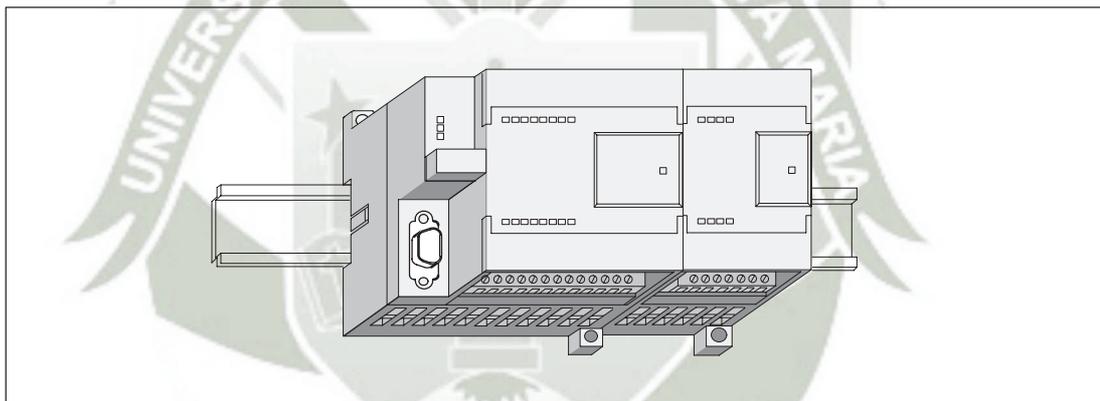


Figura 1-4 CPU con un módulo de ampliación

# 2

## Instalar un Micro-PLC S7-200

Los sistemas de automatización S7-200 (Micro-PLCs) son fáciles de instalar. Se pueden montar bien sea en un armario eléctrico, utilizando los orificios de sujeción previstos a tal efecto, o bien en un raíl normalizado (DIN) usando ganchos de retención. Sus pequeñas dimensiones permiten ahorrar espacio.

Este capítulo explica cómo instalar y cablear un sistema de automatización S7-200.

### Índice del capítulo

Apartado	Descripción	Página
2.1	Preparar el montaje	2-2
2.2	Montar y desmontar un Micro-PLC S7-200 o un módulo de ampliación	2-6
2.3	Instalar el cableado de campo	2-9
2.4	Utilizar circuitos de supresión	2-16
2.5	Alimentación de corriente	2-18

## 2.1 Preparar el montaje

### Disposición

Los sistemas de automatización S7-200 se pueden disponer en un armario eléctrico o en un raíl DIN. Es posible montarlos de forma horizontal o vertical. Un sistema de automatización S7-200 se puede conectar a un módulo de ampliación utilizando uno de los métodos siguientes:

- Un cable plano con el correspondiente conector está incorporado en el módulo de ampliación para poder conectarlo fácilmente a la CPU o a otro módulo de ampliación.
- Con objeto de flexibilizar aún más el montaje, se ofrecen también cables de conexión para los módulos de ampliación.

La figura 2-1 muestra dos ejemplos típicos de disposición.

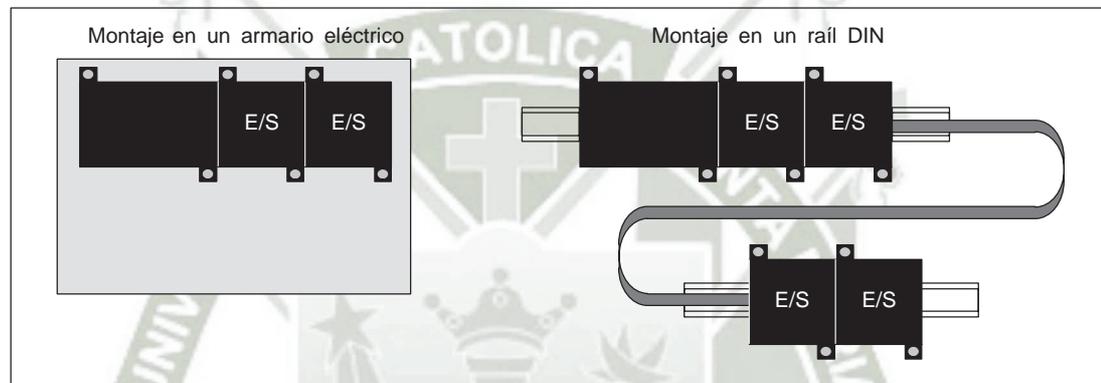


Figura 2-1 Disposición para el montaje

### Espacio necesario para montar una CPU S7-200

Al planificar la disposición de los módulos en el bastidor se deberán respetar las siguientes reglas:

- Para las CPUs S7-200 y los módulos de ampliación se ha previsto la ventilación por convección natural. Por lo tanto, se deberá dejar un margen mínimo de 25 mm por encima y por debajo de las unidades para garantizar su ventilación (v. fig. 2-2). El funcionamiento continuo a una temperatura ambiente máxima y con una carga muy elevada reduce la vida útil de cualquier dispositivo electrónico.
- Para el montaje vertical, la temperatura ambiente máxima se reduce en 10° C. La CPU se debe montar debajo de los módulos de ampliación. Si el montaje se efectúa en un raíl DIN vertical es preciso utilizar un tope.
- Para el montaje horizontal o vertical en un armario eléctrico, se deben prever 75 mm (v. fig. 2-2).
- Al planificar la disposición de los módulos, prevea suficiente espacio para el cableado de las entradas y salidas, así como para las conexiones de los cables de comunicación.

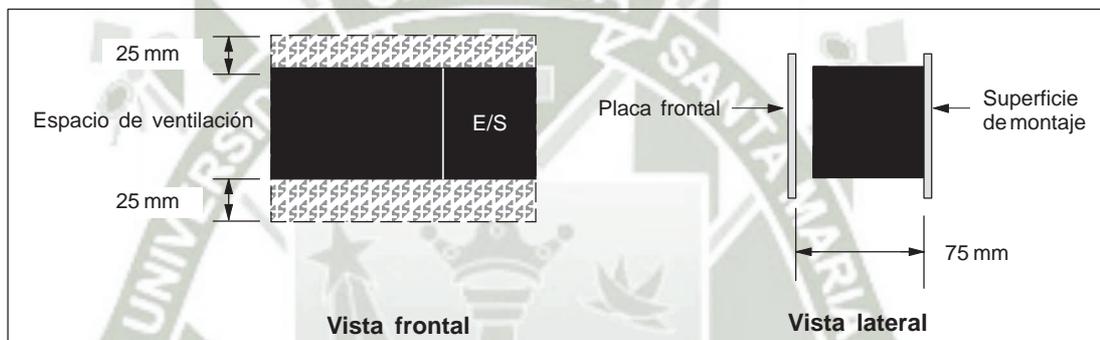


Figura 2-2 Espacio necesario para montar una CPU S7-200 de forma horizontal o vertical

### Montaje en un raíl DIN

Las CPUs S7-200 y los módulos de ampliación se pueden montar en un raíl DIN estándar (DIN EN 50 022). La figura 2-3 muestra las dimensiones de dicho raíl.

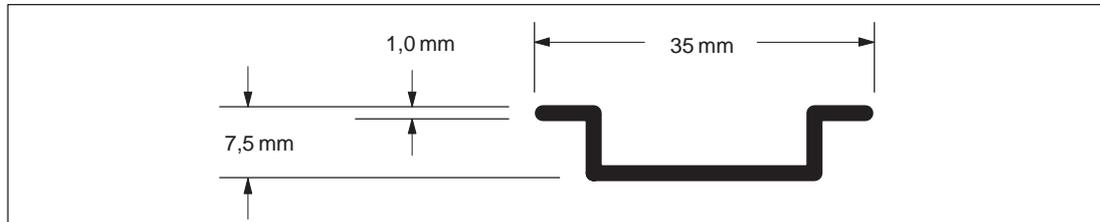


Figura 2-3 Dimensiones del raíl DIN

### Dimensiones del armario eléctrico

Las CPUs S7-200 y los módulos de ampliación disponen de orificios para facilitar su montaje en armarios eléctricos. En las figuras 2-4 a 2-6 se indican las dimensiones necesarias para montar las diversas CPUs S7-200 y los módulos de ampliación.

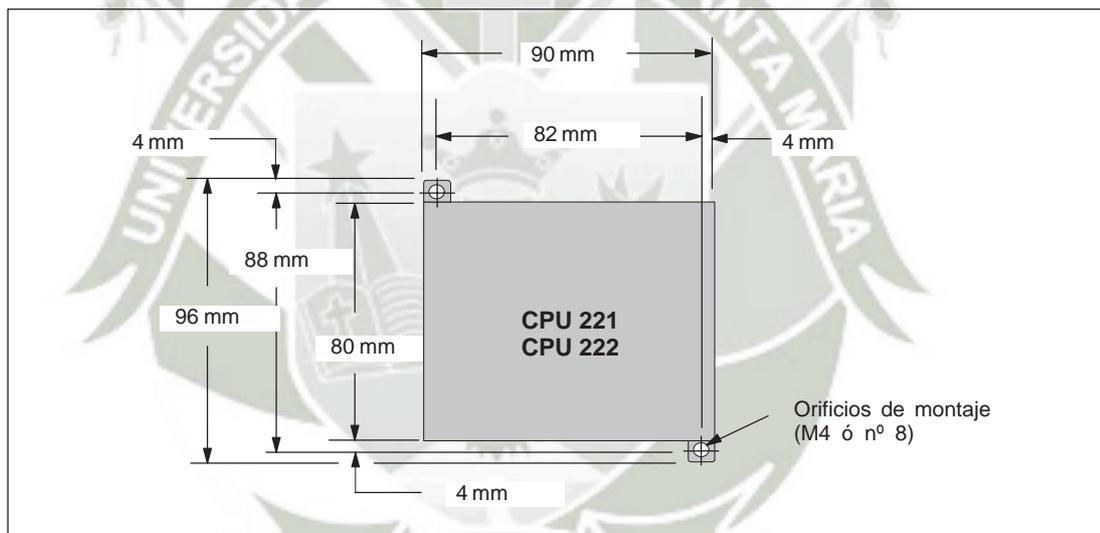


Figura 2-4 Dimensiones de montaje para una CPU 221 y una CPU 222

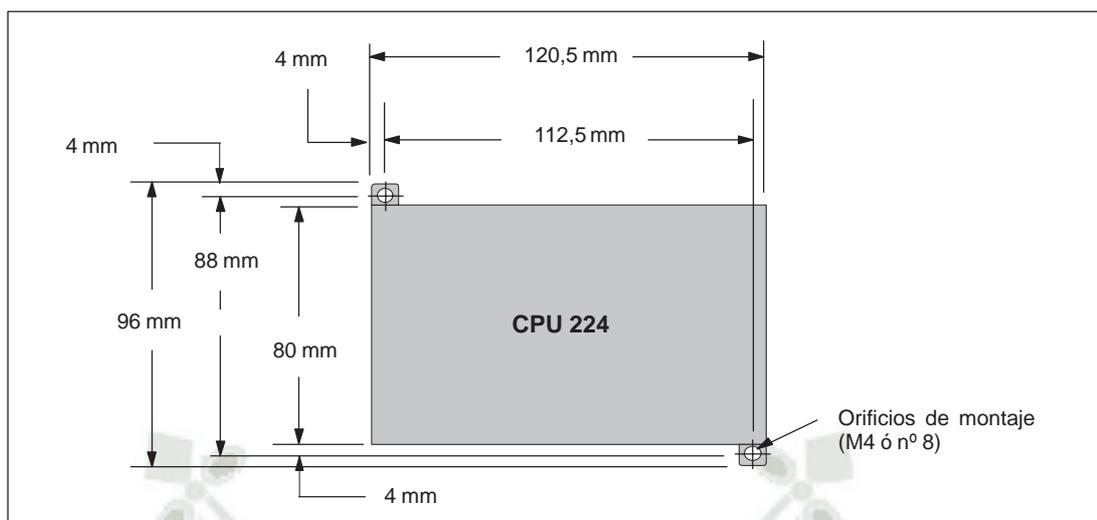


Figura 2-5 Dimensiones de montaje para una CPU 224

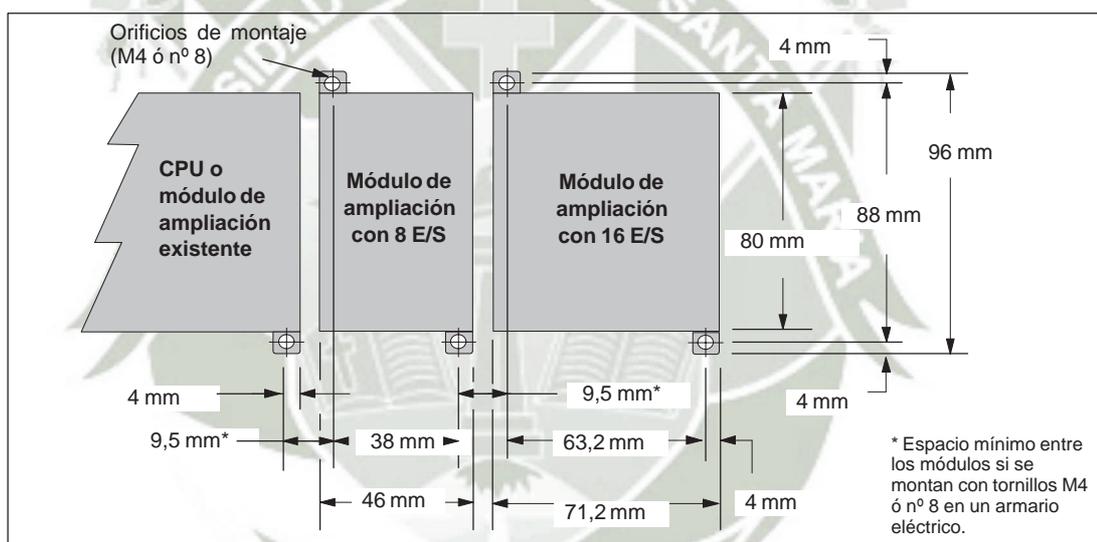


Figura 2-6 Dimensiones de montaje para módulos de ampliación

## 2.2 Montar y desmontar un Micro-PLC S7-200 o un módulo de ampliación

### Montar un Micro-PLC S7-200 o un módulo de ampliación en un armario eléctrico



#### Precaución

Si se intenta montar o desmontar los sistemas de automatización y/o los equipos conectados a los mismos durante el encendido, puede producirse un choque eléctrico o fallos en los equipos.

Si antes del montaje o desmontaje no se ha desconectado por completo la alimentación eléctrica de los módulos S7-200 y de los equipos conectados, ello podría causar la muerte o heridas graves al personal, y/o daños materiales.

Respete siempre las medidas de seguridad necesarias y verifique que la alimentación eléctrica de los módulos S7-200 se haya desconectado antes del montaje.

Para montar una CPU S7-200 en un armario eléctrico, siga los siguientes pasos:

1. Posicione y taladre los orificios de sujeción para los tornillos de tamaño DIN M4 (estándar americano nº 8). En el apartado 2.1 se indican las dimensiones de montaje y otros datos importantes al respecto.
2. Atornille la CPU S7-200 al armario eléctrico, utilizando tornillos de tamaño DIN M4 (estándar americano nº 8).

Para montar un módulo de ampliación en un armario eléctrico, siga los siguientes pasos:

1. Posicione y taladre los orificios de sujeción para los tornillos de tamaño DIN M4 (estándar americano nº 8). En el apartado 2.1 se indican las dimensiones de montaje y otros datos importantes al respecto.
2. Coloque el módulo de ampliación cerca de la CPU o del módulo de ampliación y fíjelo correctamente.
3. Enchufe el cable de cinta flexible del módulo de ampliación en el conector de la CPU ubicado debajo de la tapa frontal. El cable muestra la orientación correcta.
4. Así se finaliza el montaje.

## Montar un Micro-PLC S7-200 o un módulo de ampliación en un perfil soporte



### Precaución

Si se intenta montar o desmontar los sistemas de automatización y/o los equipos conectados a los mismos durante el encendido, puede producirse un choque eléctrico o fallos en los equipos.

Si antes del montaje o desmontaje no se ha desconectado por completo la alimentación eléctrica de las CPUs S7-200 y de los equipos conectados a las mismas, ello podría causar la muerte o heridas graves al personal, y/o daños materiales.

Respete siempre las medidas de seguridad necesarias y verifique que la alimentación eléctrica de los módulos S7-200 se haya desconectado antes del montaje.

Para montar una CPU S7-200 en un raíl DIN, siga los siguientes pasos:

1. Atornille el raíl DIN al armario eléctrico dejando un espacio de 75 mm entre tornillo y tornillo.
2. Abra el gancho de retención (ubicado en el lado inferior de la CPU S7-200) y enganche la parte posterior de la CPU al raíl DIN.
3. Cierre el gancho de retención y verifique que la CPU S7-200 haya enganchado correctamente en el raíl.

Para montar un módulo de ampliación en un raíl DIN, siga los siguientes pasos:

1. Abra el gancho de retención y enganche la parte posterior del módulo en el raíl próximo a la CPU o al módulo de ampliación.
2. Cierre el gancho de retención para fijar el módulo de ampliación al raíl. Asegúrese de que el módulo se haya enganchado correctamente en el raíl.
3. Enchufe el cable de cinta flexible del módulo de ampliación en el conector de la CPU ubicado debajo de la tapa frontal. El cable muestra la orientación correcta.
4. Así se finaliza el montaje.

### Nota

Si los módulos se montan en entornos donde se presenten vibraciones fuertes, o bien en posición vertical, puede ser necesario asegurarlos con topes.

## Desmontar un Micro-PLC o un módulo de ampliación S7-200



### Precaución

Si se intenta montar o desmontar los sistemas de automatización y/o los equipos conectados a los mismos durante el encendido, puede producirse un choque eléctrico o fallos en los equipos.

Si antes del montaje o desmontaje no se ha desconectado por completo la alimentación eléctrica de las CPUs S7-200 y de los equipos conectados a las mismas, ello podría causar la muerte o heridas graves al personal, y/o daños materiales.

Respete siempre las medidas de seguridad necesarias y verifique que la alimentación eléctrica de los módulos S7-200 se haya desconectado antes de proceder a su montaje.

Para desmontar una CPU o un módulo de ampliación S7-200, siga los siguientes pasos:

1. Desconecte todos los cables enchufados al módulo que desee desmontar (v. fig. 2-7). Algunas CPUs y algunos módulos de ampliación disponen de conectores extraíbles.
2. Abra la tapa de acceso frontal y desconecte el cable de cinta flexible de los módulos adyacentes.
3. Desatornille los tornillos de montaje o abra el gancho de retención y desmonte el módulo.



### Precaución

Si instala un módulo incorrecto, es posible que el programa instalado en la CPU funcione de forma impredecible.

Si un módulo y un cable de ampliación se sustituyen con otro modelo o si no se instalan con la orientación correcta, ello podría causar la muerte o heridas graves al personal, y/o daños materiales.

Sustituya un módulo de ampliación con el mismo modelo y oriéntelo correctamente.

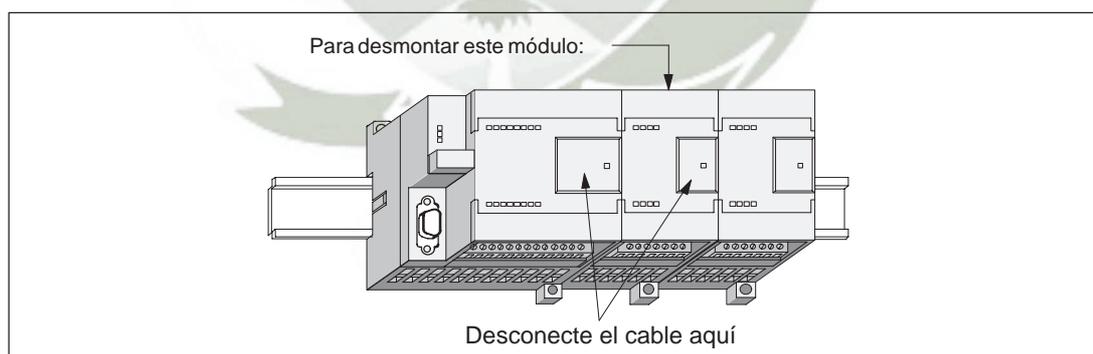


Figura 2-7 Desmontar un módulo de ampliación

## 2.3 Instalar el cableado de campo



### Precaución

Si se intenta montar o desmontar los sistemas de automatización y/o los equipos conectados a los mismos durante el encendido, puede producirse un choque eléctrico o fallos en los equipos.

Si antes del montaje o desmontaje no se ha desconectado por completo la alimentación eléctrica de las CPUs S7-200 y de los equipos conectados a las mismas, ello podría causar la muerte o heridas graves al personal, y/o daños materiales.

Antes de instalar el cableado de campo es necesario tomar las precauciones de seguridad adecuadas y verificar que estén desconectadas las fuentes de alimentación de los sistemas de automatización S7-200.

### Reglas de carácter general

Los puntos siguientes constituyen reglas de carácter general para la instalación y el cableado de los Micro-PLCs S7-200:

- Al cablear un Micro-PLC S7-200 es necesario respetar todos los reglamentos, códigos y normas eléctricas vinculantes. Instale y utilice el equipo conforme a todas las normas nacionales y locales vigentes. Dirijase a las autoridades locales para informarse acerca de qué reglamentos, códigos o normas rigen en el lugar de instalación.
- Utilice siempre cables con una sección adecuada para la intensidad. Los S7-200s aceptan cables con sección de 1,50 mm<sup>2</sup> a 0,50 mm<sup>2</sup> (14 AWG a 22 AWG).
- No apriete excesivamente los bornes de tornillo, pues podrían pasarse de rosca. El par máximo de apriete es de 0.56 N-m.
- Utilice siempre un cable lo más corto posible (apantallado o blindado, como máximo 500 metros, sin pantalla o blindaje, 300 metros). El cableado deberá efectuarse por pares; con el cable de neutro o común combinado con un cable de fase o uno de señal.
- Separe el cableado de corriente alterna y el cableado de corriente continua de alta tensión y rápida conmutación de los cables de señal de baja tensión.
- Identifique y disponga adecuadamente el cableado hacia los S7-200s. De ser necesario, prevea un alivio de tracción. Para obtener más información sobre la identificación de terminales o bornes, consulte las hojas de datos técnicos en el Anexo A.
- Instale dispositivos de supresión de sobretensiones apropiados en el cableado susceptible de recibir sobretensiones causadas por rayos.
- Ninguna alimentación externa deberá aplicarse a una carga de salida en paralelo con una salida de corriente continua (DC). En caso contrario puede circular corriente inversa a través de la salida a menos que se instale un diodo u otra barrera.

**Precaución**

Un funcionamiento anormal de los equipos de control puede causar un funcionamiento inesperado del equipo controlado.

Dicho funcionamiento inesperado puede causar la muerte o lesiones personales graves y/o daños al equipo.

Prevea dispositivos de parada de emergencia, dispositivos electromecánicos de mayor jerarquía y otras medidas redundantes de seguridad que sean independientes del sistema de automatización.

**Reglas de puesta a tierra de referencia de potencial para circuitos aislados**

Seguidamente se indican las reglas de puesta a tierra para circuitos aislados:

- Deberá identificarse el punto de referencia (referencia de tensión 0) para cada circuito de la instalación así como los puntos donde puedan interconectarse circuitos con referencias de potencial diferentes. Tal tipo de conexiones puede causar circulaciones parásitas de corriente con consecuencias indeseadas, tales como errores lógicos o circuitos deteriorados. Una causa muy común de diferentes potenciales de referencia son tomas de tierra que están separadas físicamente por una gran distancia. Cuando se interconectan dispositivos con tierras muy separadas a través de un cable de comunicación o de sensor, por el circuito creado por el cable y tierra pueden circular corrientes inesperadas. Las corrientes de carga de maquinaria pesada pueden causar, incluso con distancias reducidas, diferencias de potencial de tierra o generar corrientes indeseadas por fenómenos de inducción electromagnética. Las fuentes de alimentación que no tengan coordinada su referencia de potencial 0 pueden causar corrientes dañinas al circular entre sus circuitos asociados.
- Si una CPU con potencial de tierra diferente se conecta a una misma red PPI, es preciso utilizar un repetidor RS 485 aislado.
- Los productos S7-200 incluyen aislamientos en ciertos puntos para prevenir la circulación de corrientes indeseadas en la instalación. Al planear la instalación, se deberá considerar dónde existen tales elementos de aislamiento y dónde no. También se deberán considerar los puntos de aislamiento en fuentes de alimentación asociadas y otros equipos, así como los puntos que utilizan como referencia las fuentes de alimentación asociadas.
- Los puntos de referencia de tierra y los aislamientos que ofrece el equipo deberán elegirse de forma que se interrumpan bucles de circuito innecesarios que pueden causar la circulación de corrientes indeseadas. No olvide considerar aquellas conexiones temporales que pueden introducir cambios en el potencial de referencia de los circuitos, p.ej. la conexión de una unidad de programación a la CPU.
- Al definir físicamente las tierras es necesario considerar los requisitos de puesta a tierra de protección y el correcto funcionamiento de los aparatos de protección por corte.
- Para obtener el mayor nivel posible de inmunidad a interferencias en la mayoría de las instalaciones, la conexión M de la fuente de alimentación de sensores se deberá conectar a tierra.

Las descripciones siguientes constituyen una introducción a las características de aislamiento generales de la gama S7-200. Sin embargo, algunas prestaciones pueden diferir en determinados productos. Las hojas de datos técnicos en el Anexo A incluyen información sobre los puntos de aislamiento y los valores nominales de los mismos para determinados productos. Los aislamientos con valores nominales inferiores a AC 1.500 V han sido diseñados únicamente como aislamiento funcional y no deberán tomarse para definir barreras de seguridad.

- El potencial de referencia de la lógica de circuito es el mismo que el de la conexión M de la fuente de alimentación DC de sensores.
- El potencial de referencia de la lógica de circuito es el mismo que el punto M de la alimentación de entrada en el caso de una CPU con alimentación en corriente continua.
- Los puertos de comunicación de la CPU tienen el mismo potencial de referencia que la lógica de circuito.
- Las entradas y salidas (E/S) analógicas no están aisladas con respecto a la lógica de circuito. Las entradas analógicas son de tipo diferencial, es decir tienen una baja razón de rechazo en modo común.
- La lógica de circuito está aislada de la tierra hasta AC 500 V.
- Las E/S digitales en DC están aisladas de la lógica de la CPU hasta AC 500 V.
- Los grupos de E/S digitales en DC están aislados unos de otros hasta AC 500 V.
- Las salidas de relé están aisladas de la lógica de la CPU hasta AC 1.500 V.
- Los grupos de salida están aislados unos de otros hasta AC 1.500 V.
- La fase y el neutro de la alimentación en alterna están aislados de tierra, la lógica de la CPU y todas las E/S hasta AC 1.500 V.

### Utilizar el bloque de bornes opcional en CPUs sin conector extraíble

El bloque de bornes opcional para el cableado de campo (v. fig. 2-8) tiene la ventaja de que las conexiones permanecen fijas aun al desmontar o montar la CPU S7-200. En el Anexo E se indica el número de referencia de dicho bloque.

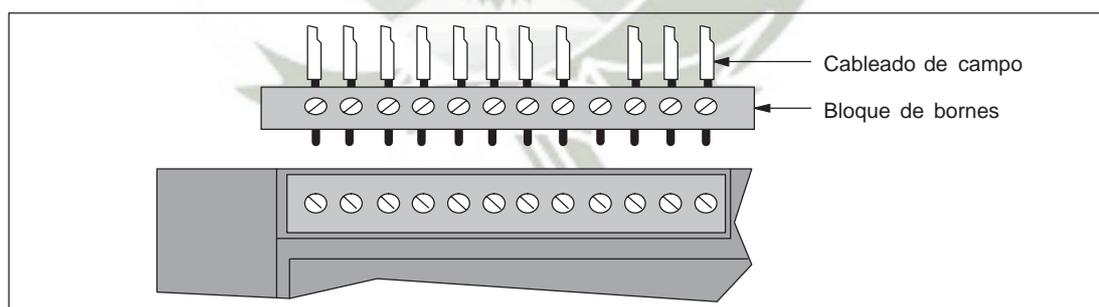


Figura 2-8 Bloque de bornes opcional

### Bloque de terminales extraíble

El bloque de terminales extraíble para el cableado de campo (v. fig. 2-9) tiene la ventaja de que las conexiones permanecen fijas aun al desmontar o montar la CPU S7-200 y los módulos de ampliación.

Para soltar el bloque de terminales de la CPU o del módulo de ampliación, siga los siguientes pasos:

1. Levante la tapa de acceso superior de la CPU o del módulo de ampliación.
2. En la mitad del bloque de terminales, inserte un destornillador en la ranura como muestra la figura 2-9.
3. Empuje hacia abajo y haga palanca para soltar el bloque de terminales como se muestra abajo.

Para enganchar nuevamente un bloque de terminales en una CPU o en un módulo de ampliación, siga los siguientes pasos:

1. Levante la tapa de acceso superior de la CPU o del módulo de ampliación.
2. Verifique que el nuevo bloque de terminales esté alineado correctamente con los pines de la CPU o del módulo de ampliación.
3. Empuje el bloque de terminales hacia abajo hasta que enganche correctamente en la CPU o en el módulo de ampliación.

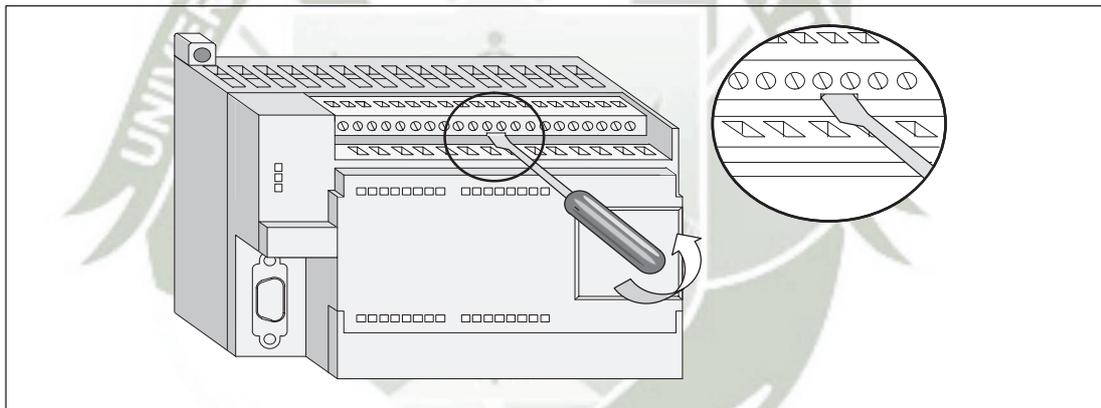


Figura 2-9 Bloque de terminales extraíble para la CPU 224 y los módulos de ampliación

## Reglas para la instalación con corriente alterna

Seguidamente se indican reglas de carácter general para instalaciones con corriente alterna (v. fig. 2-10).

[a] Instale un interruptor unipolar para cortar la alimentación de la CPU, todos los circuitos de entrada y todos los circuitos de salida (de carga).

[b] Prevea dispositivos de sobrecorriente para proteger la alimentación de la CPU, las salidas y las entradas. Para mayor protección es posible instalar un fusible en cada salida.

[c] No se precisa protección de sobrecorriente externa para las entradas si se utiliza la fuente de alimentación de DC 24 V para sensores integrada en el Micro-PLC. Esta fuente para sensores está protegida contra cortocircuitos.

[d] Conecte todos los terminales de tierra del S7-200 por el camino más corto a tierra para obtener el mayor nivel posible de inmunidad a interferencias. Es recomendable conectar todos los terminales de masa a un solo punto eléctrico. Para establecer esta conexión, utilice un cable con una sección de 14 AWG ó 1,5 mm<sup>2</sup>.

[e] La fuente de alimentación DC para sensores integrada en el módulo base puede usarse también para alimentar las entradas de dicho componente, [f] las entradas DC de ampliación y [g] las bobinas de los relés del módulo de ampliación. Esta fuente para sensores está protegida contra cortocircuitos.

[h] Para obtener el mayor nivel posible de inmunidad a interferencias en la mayoría de las instalaciones, la conexión M de la fuente de alimentación de sensores se deberá conectar a tierra.

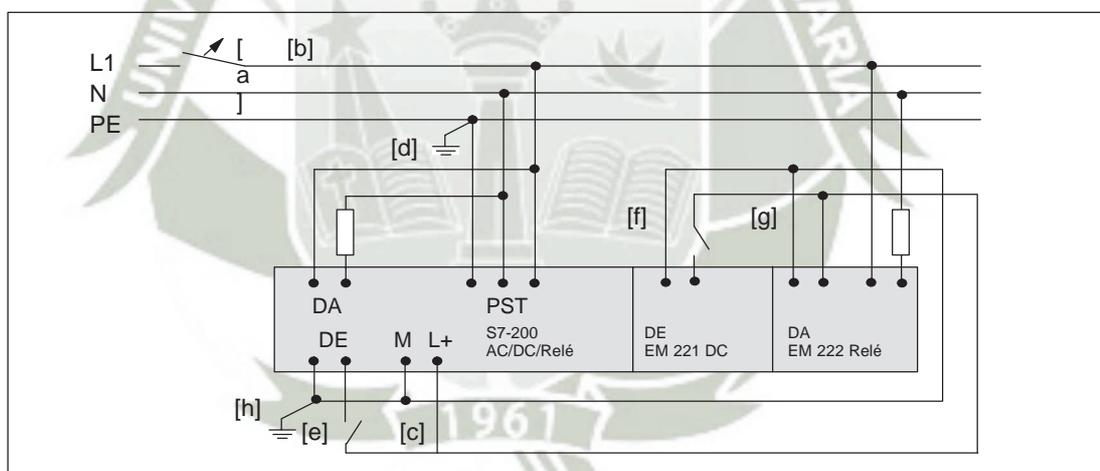


Figura 2-10 AC 120/230 V Uso de un interruptor de sobrecorriente único para proteger la CPU y la carga

## Reglas para la instalación con corriente continua

Seguidamente se indican reglas de carácter general para instalaciones con corriente continua aisladas (v. fig. 2-11).

[a] Instale un interruptor unipolar para cortar la alimentación de la CPU, todos los circuitos de entrada y todos los circuitos de salida (de carga).

[b] Prevea dispositivos de sobrecorriente para proteger la alimentación de la CPU, [c] las salidas y [d] las entradas. Para mayor protección es posible instalar un fusible en cada salida. No se precisa protección de sobrecorriente externa para las entradas si se utiliza la fuente de alimentación de DC 24 V para sensores integrada en el Micro-PLC. Esta última fuente de alimentación dispone de una función de limitación interna de corriente.

[e] Verifique que la fuente de alimentación DC tenga suficiente capacidad para mantener la tensión en caso de que se produzcan cambios bruscos de carga. De no ser así, prevea condensadores externos adecuados.

[f] Para obtener el mayor nivel posible de inmunidad a interferencias en la mayoría de las instalaciones, es recomendable conectar a tierra todas las fuentes de alimentación DC. Equipe las fuentes de alimentación DC no puestas a tierra con una resistencia y un condensador en paralelo [g] conectado entre el común de la alimentación y el conductor de protección. Dicha resistencia ofrece una vía de fuga para prevenir acumulaciones de carga estática; el condensador permite derivar las interferencias de alta frecuencia. Los valores típicos son 1 M $\Omega$  y 4.700 pf.

[h] Conecte todos los terminales de tierra del S7-200 por el camino más corto a tierra para obtener el mayor nivel posible de inmunidad a interferencias. Es recomendable conectar todos los terminales de masa a un solo punto eléctrico. Para establecer esta conexión, utilice un conductor con un sección de 14 AWG ó 1,5 mm<sup>2</sup>.

Para alimentar circuitos de DC 24 V, utilice siempre una fuente que ofrezca separación eléctrica segura de la red de AC 120/230 V y fuentes de peligro similares.

Los documentos siguientes incluyen definiciones de separación segura de circuitos:

- Protected extra low voltage: conforme a EN60204-1
- Clase 2 ó Limited Voltage/Current Circuit conforme a UL 508

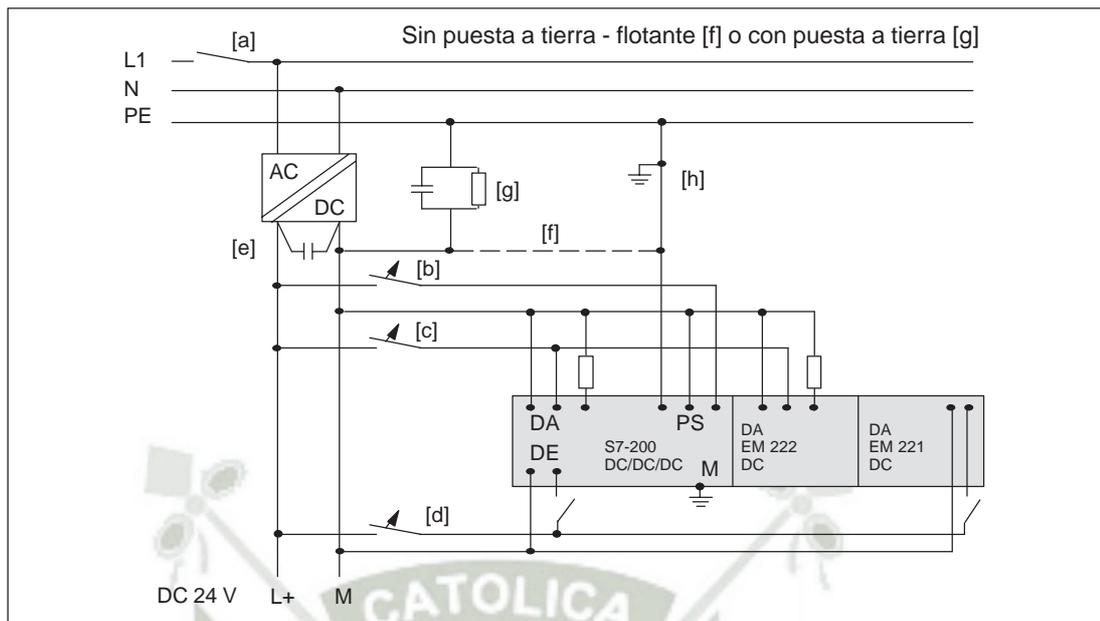


Figura 2-11 Instalación en una red de corriente continua aislada

## 2.4 Utilizar circuitos de supresión

### Reglas de carácter general

Las cargas inductivas deberán equiparse con circuitos supresores destinados a limitar el incremento de tensión producido al cortarse la alimentación. Tenga en cuenta las reglas siguientes al diseñar la supresión adecuada. La eficacia de un determinado diseño depende de la aplicación. Por tanto, deberá verificarse para cada caso particular. Verifique que las características nominales de todos los componentes sean adecuadas para la aplicación en cuestión.

### Proteger transistores en DC

Las salidas en DC a transistores del S7-200 contienen diodos zener adecuados para múltiples aplicaciones. Para prevenir la sobrecarga de los diodos internos, utilice diodos supresores externos en caso de cargas inductivas elevadas o que se conmuten con frecuencia. Las figuras 2-12 y 2-13 muestran aplicaciones típicas para salidas DC a transistor.

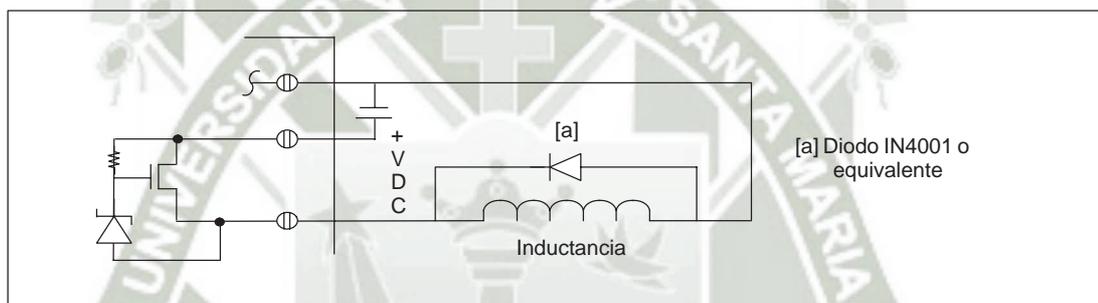


Figura 2-12 Protección por diodo para salidas en DC a transistores

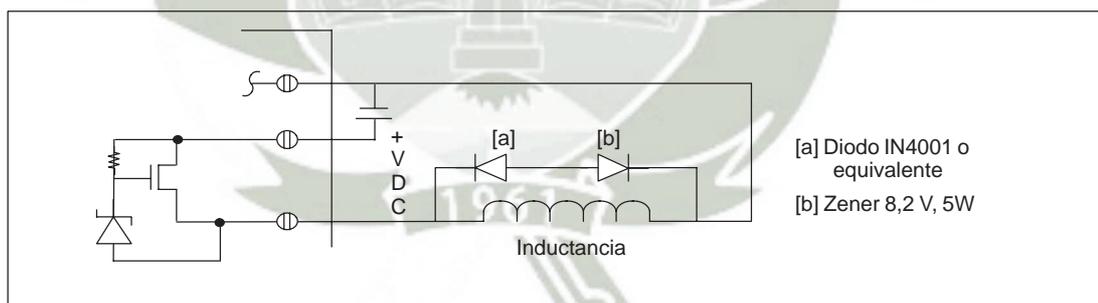


Figura 2-13 Protección por diodo zener para salidas en DC a transistores

### Proteger relés que controlan corriente continua

La figura 2-14 muestra las redes de resistencia/condensador que se pueden utilizar para aplicaciones de relé en baja tensión (30 V) DC. Conecte la red en los terminales de la carga.

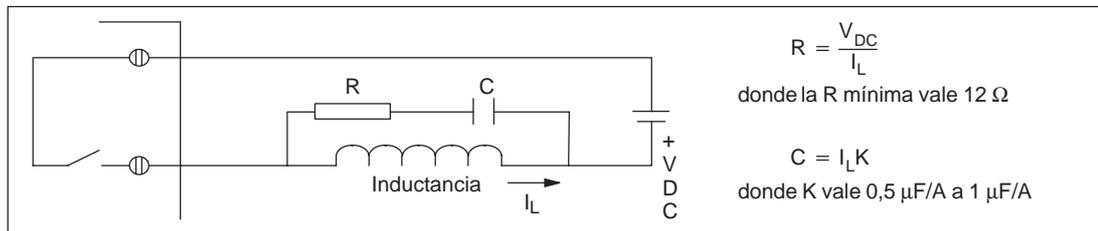


Figura 2-14 Red resistencia/condensador conectada a una carga DC controlada por relés

Para las aplicaciones con relé en corriente continua se puede utilizar también la protección por diodo supresor como muestran las figuras 2-12 y 2-13. Se permite una tensión de umbral de hasta 36 V si se utiliza un diodo zener conectado de forma inversa.

### Proteger relés que controlan corriente alterna

Si se utiliza un relé para conmutar cargas inductivas con AC 115 V/230 V, se deben conectar redes de resistencia/condensador entre los contactos del relé como muestra la figura 2-15. También pueden utilizarse varistores de óxido metálico (MOV) para limitar la tensión de pico. Verifique que la tensión de trabajo del varistor MOV sea como mínimo un 20% superior a la tensión nominal de fase.

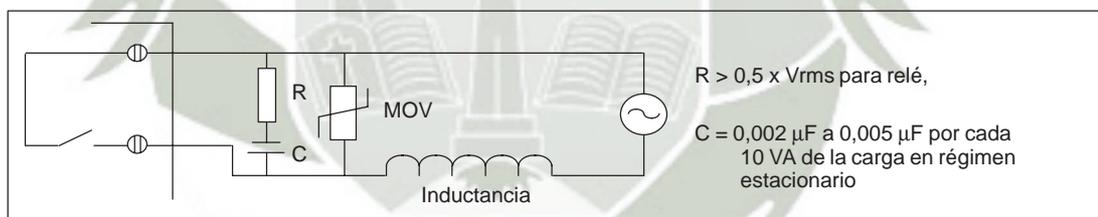


Figura 2-15 Carga AC con red conectada entre contactos del relé

El condensador permite la circulación de la corriente de fugas aunque esté abierto el contacto. Verifique que la corriente de fugas,  $I(\text{fuga}) = 2 \times 3,14 \times f \times C \times V_{ef}$ , sea aceptable para la aplicación.

Ejemplo: Las especificaciones de un contactor NEMA de tamaño 2 muestran un consumo transitorio de la bobina de 183 VA y un consumo de 17 VA en régimen estacionario. Con AC 115 V, la corriente transitoria vale  $183 \text{ VA}/115 \text{ V} = 1,59 \text{ A}$ , es decir, es inferior a la capacidad de cierre, que vale 2A, de los contactos del relé.

Dimensionamiento de la resistencia =  $0,5 \times 115 = 57,5 \Omega$ ; elegir 68  $\Omega$  como valor estándar.  
Dimensionamiento del condensador =  $(17 \text{ VA}/10) \times 0,005 = 0,0085 \mu F$ ; elegir 0,01  $\mu F$  como valor estándar.

La corriente de fugas vale =  $2 \times 3,14 \times 60 \times 0,01 \times 10^{-6} \times 115 = 0,43 \text{ mA}$  eficaces (rms).

## 2.5 Alimentación de corriente

Los módulos base del S7-200 tienen integrada una fuente de alimentación capaz de abastecer el módulo base, los módulos de ampliación y otras cargas que precisen DC 24 V. Utilice la información siguiente como guía para determinar cuánta energía (o corriente) puede suministrar el módulo central a la configuración deseada.

### Requisitos de alimentación

Cada CPU S7-200 ofrece alimentación tanto en DC 5 V como DC 24 V:

- Cada CPU dispone de una fuente de alimentación para sensores de DC 24 V que puede suministrar esta tensión para las entradas locales o para las bobinas de relés en los módulos de ampliación. Si el consumo de DC 24 V supera la corriente que es capaz de aportar el módulo CPU, entonces puede añadirse una fuente de alimentación externa de DC 24 V para abastecer con DC 24 V los módulos de ampliación. La alimentación de DC 24 V se debe conectar manualmente a dichas entradas o bobinas de relé.
- La CPU alimenta también con DC 5 V los módulos de ampliación cuando se conectan al módulo base. Si el consumo de DC 5 V de los módulos de ampliación supera la corriente aportable por la CPU, entonces es necesario desconectar tantos módulos de ampliación como sean necesarios para no superar la corriente aportable por la CPU.

Las hojas de datos técnicos que se incluyen en el Anexo A informan sobre las corrientes suministrables por las CPUs y sobre el consumo de los módulos de ampliación.



---

### Precaución

Si se conecta una fuente de alimentación externa de DC 24 V en paralelo con la fuente de alimentación para sensores DC del S7-200, esto puede causar un conflicto entre ambas fuentes ya que cada una intenta establecer su propio nivel de tensión de salida.

Este conflicto puede tener como consecuencia una reducción de la vida útil o la avería inmediata de una o ambas fuentes de alimentación y, en consecuencia, el funcionamiento imprevisible del sistema de automatización. Un funcionamiento imprevisible puede ocasionar la muerte o lesiones graves al personal, y/o daños al equipo o a bienes materiales.

La fuente de alimentación para sensores del S7-200 y la fuente de alimentación externa deben alimentar diferentes puntos. Se permite una conexión común de los cables neutros.

---

### Ejemplo de cálculo de los requisitos de alimentación

La tabla 2-1 muestra un ejemplo de cálculo de los requisitos de alimentación de un Micro-PLC S7-200 compuesto de los módulos siguientes:

- CPU 224 AC/DC/relé
- 3 módulos de ampliación EM 223, 8 entradas DC / 8 salidas de relé
- 1 módulo de ampliación EM 221, 8 entradas DC

Esta instalación tiene un total de 46 entradas y 34 salidas.

La CPU de este ejemplo suministra suficiente corriente (DC 5 V) para los módulos de ampliación, pero la alimentación de sensores no suministra suficiente corriente DC 24 V para todas las entradas y salidas de relé. Las E/S requieren 400 mA, pero la CPU sólo puede suministrar 280 mA. Para esta instalación se necesita una fuente adicional de alimentación de 120 mA (como mínimo) y con corriente DC 24 V para que todas las entradas y salidas puedan funcionar correctamente.

Tabla 2-1 Cálculo de requisitos de alimentación en una configuración de ejemplo

Corriente máx. CPU	DC 5 V	DC 24 V
CPU 224 AC/DC/relé	660 mA	280mA
<i>menos</i>		
Consumo del sistema	DC 5 V	DC 24 V
CPU 224, 14 entradas		14 * 4 mA = 56 mA
3 EM 223, alimentación necesaria de 5 V	3 * 80 mA = 240 mA	
1 EM 221, alimentación necesaria de 5V	1 * 30 mA = 30 mA	
3 EM 223, 8 entradas c/u		3 * 8 * 4 mA = 96 mA
3 EM 223, 8 salidas de relé c/u		3 * 8 * 9 mA = 216 mA
1 EM 221, 8 entradas		8 * 4 mA = 32 mA
Consumo total	270 mA	400 mA
<i>igual a</i>		
Balance de corriente	DC 5 V	DC 24 V
Balance total de corriente	390 mA	[120 mA]

### Calcular la corriente necesaria

Utilice la tabla siguiente para determinar cuánta corriente (o energía) puede suministrar la CPU a la configuración en cuestión. Consulte el anexo A para obtener información sobre las corrientes suministrables por la CPU y el consumo de los módulos de ampliación.

Alimentación	DC 5 V	DC 24 V

*menos*

Consumo del sistema	DC 5 V	DC 24 V
	Unidad central	
Consumo total		

*igual a*

Balance de corriente	DC 5 V	DC 24 V
Balance total de corriente		

## Instalación y configuración del sistema de automatización S7-200

# 3

El presente capítulo describe cómo instalar y configurar un sistema de automatización S7-200. El sistema de automatización S7-200 descrito aquí comprende:

- Una CPU S7-200.
- Un PC o una unidad de programación con STEP 7-Micro/WIN 32 instalado.
- Un cable de interconexión.

### Índice del capítulo

Apartado	Descripción	Página
3.1	Resumen breve	3-2
3.2	Cómo instalar STEP 7-Micro/WIN 32	3-3
3.3	Cómo configurar la comunicación utilizando el cable PC/PPI	3-5
3.4	Cómo establecer la comunicación con la CPU S7-200	3-9
3.5	Cómo cambiar los parámetros de comunicación de la CPU	3-10

### 3.1 Resumen breve

#### Información general

Para la instalación se deben tener en cuenta los siguientes criterios:

- El sistema operativo utilizado (Windows 95, Windows 98, o Windows NT 4.0).
- El tipo de hardware utilizado, p.ej.:
  - PC con cable PC/PPI
  - PC o unidad de programación SIMATIC con procesador de comunicaciones (CP)
  - CPU 221, CPU 222, CPU 224
  - Módem
- La velocidad de transferencia utilizada.

#### Hardware y software recomendados

STEP 7-Micro/WIN 32, versión 3.0 es un software de programación que asiste los entornos Windows 95 (de 32 bits), Windows 98 y Windows NT. Para poder utilizar STEP 7-Micro/WIN 32 se recomiendan los siguientes componentes:

- Un ordenador personal (PC) con un procesador 80586 o superior y 16 MB de RAM, o bien una unidad de programación Siemens con STEP 7-Micro/WIN 32 instalado (p.ej. una PG 740); requisito mínimo de PC: procesador 80486 con 8 MB de RAM
- Uno de los componentes siguientes:
  - Un cable PC/PPI conectado al puerto de comunicación.
  - Una tarjeta de procesador de comunicaciones (CP).
- Una pantalla VGA, o una pantalla asistida por Microsoft Windows.
- 50 MB libres en el disco duro (como mínimo).
- Windows 95, Windows 98 o Windows NT 4.0.
- Opcional pero recomendable: un ratón asistido por Microsoft Windows.

STEP 7-Micro/WIN 32 incorpora una extensa Ayuda en pantalla y una *Guía de iniciación* también en pantalla. Elija el comando de menú **Ayuda** o pulse la tecla F1 para obtener las informaciones más recientes.

## 3.2 Cómo instalar STEP 7-Micro/WIN 32

### Instrucciones previas a la instalación

Siga los siguientes pasos antes de instalar el programa:

- Si ya está instalada una versión anterior de STEP 7-Micro/WIN 32, cree una copia de seguridad de todos los proyectos de STEP 7-Micro/WIN en un disquete.
- Verifique que todas las demás aplicaciones estén cerradas, incluyendo la barra de herramientas de Microsoft Office.
- Verifique que esté conectado el cable entre el PC y la CPU. Consulte el apartado 3.3 para obtener más información al respecto.

### Instalación de STEP 7-Micro/WIN 32

Para instalar STEP 7-Micro/WIN 32, siga los siguientes pasos:

1. Inserte el CD o el disquete en la correspondiente unidad del PC.
2. Haga clic en el botón "Inicio" para abrir el menú de Windows.
3. Haga clic en **Ejecutar...**
4. Si la instalación se efectúa desde un  
Disquete: En el cuadro de diálogo "Ejecutar", teclee **a:\setup** y haga clic en el botón "Aceptar" o pulse la tecla INTRO. Así se inicia la instalación.  
CD: En el cuadro de diálogo "Ejecutar", teclee **e:\setup** y haga clic en Aceptar o pulse la tecla INTRO. Así se inicia la instalación.
5. Siga las instrucciones que van apareciendo en pantalla hasta finalizar la instalación.
6. Al final de la instalación aparece automáticamente el cuadro de diálogo "Ajustar interface PG/PC". Haga clic en "Cancelar" para acceder a la ventana principal de STEP 7-Micro/WIN 32 (v. fig. 3-1).

Lea el archivo READMEX.TXT incluido en el CD o en los disquetes de instalación para obtener las informaciones más recientes acerca de STEP 7-Micro/WIN 32. (En lugar de x, aparecerá la letra A = alemán, B = inglés, C = francés, D = español, E = italiano).

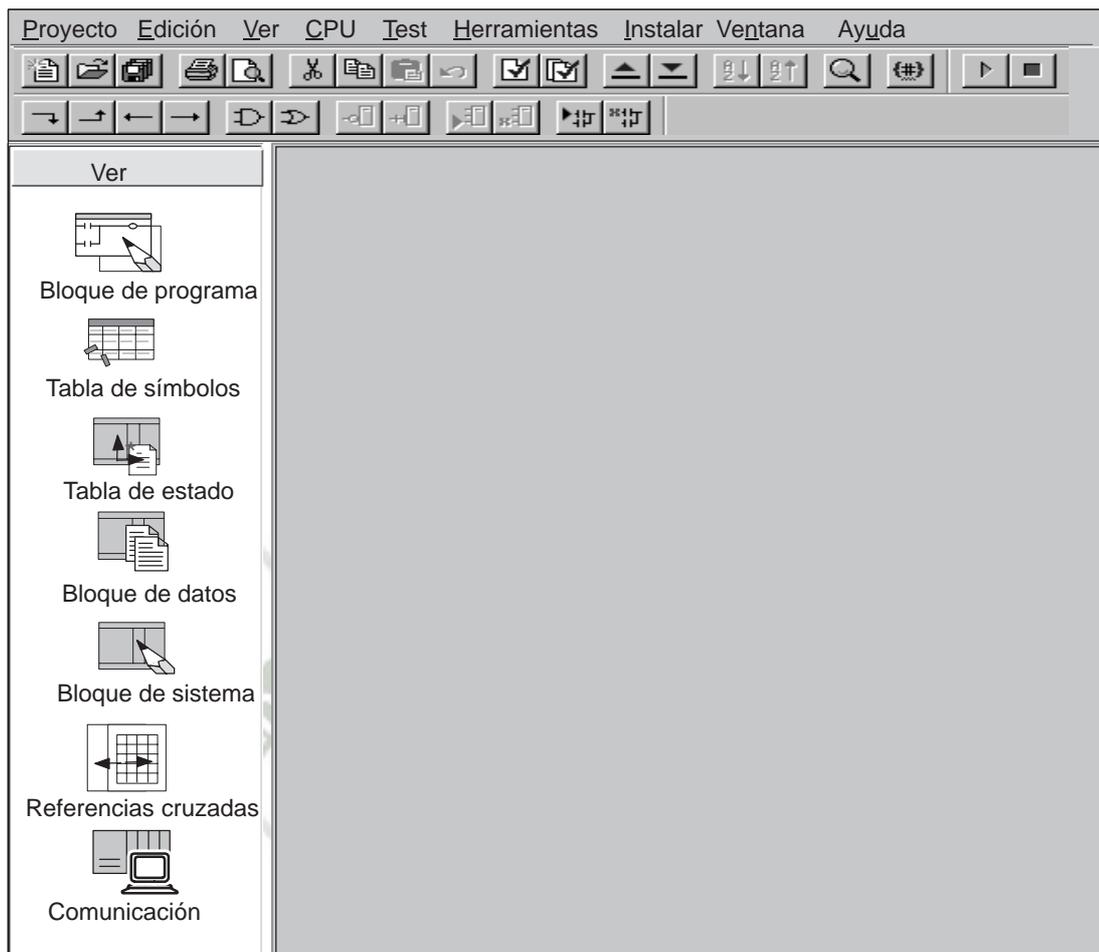


Figura 3-1 Menú Ver de STEP 7-Micro/WIN 32

### Fallos de comunicación

Las siguientes situaciones pueden causar fallos de comunicación:

- Velocidad de transferencia incorrecta: corrija la velocidad de transferencia.
- Dirección de estación incorrecta: corrija la dirección de estación.
- Cable PC/PPI ajustado incorrectamente: verifique los ajustes de los interruptores DIP del cable PC/PPI.
- Puerto de comunicaciones incorrecto en el PC: verifique el puerto COM.
- CPU en modo Freeport (puerto de comunicaciones bajo control del programa de usuario): cambie la CPU a modo STOP.
- Conflicto con otros maestros: desconecte la CPU de la red.

### 3.3 Cómo configurar la comunicación utilizando el cable PC/PPI

Aquí se explica cómo configurar la comunicación entre la CPU S7-200 y el PC utilizando el cable PC/PPI. Esta es una configuración con un solo maestro y sin ningún otro equipo de hardware instalado (como p.ej. un módem o una unidad de programación).

#### Cómo conectar el PC a la CPU

La figura 3-2 muestra una configuración típica para conectar el PC a la CPU utilizando el cable PC/PPI. Para establecer un enlace correcto entre los componentes:

1. Ajuste los interruptores DIP del cable PC/PPI a la velocidad de transferencia asistida por su PC. Seleccione también las opciones "11 bits" y "DCE" si su cable PC/PPI las asiste.
2. Conecte el extremo RS-232 ("PC") del cable PC/PPI al puerto de comunicaciones de su PC (COM1 ó COM2) y apriete los tornillos.
3. Conecte el extremo RS-485 ("PPI") del cable PC/PPI al puerto de comunicaciones de la CPU y apriete los tornillos.

Los datos técnicos del cable PC/PPI se indican en el Anexo A. Su número de referencia figura en el Anexo E.

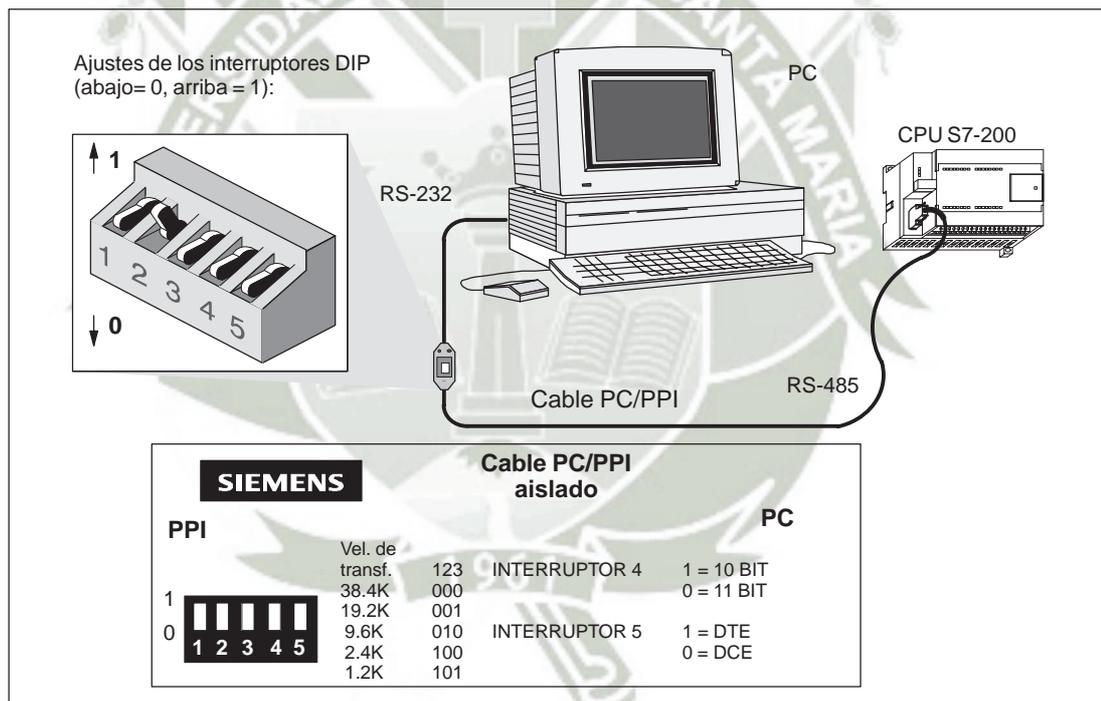


Figura 3-2 Comunicación con una CPU en modo PPI

## Cómo verificar los parámetros estándar del interface

Para verificar los parámetros estándar de su interface, siga los siguientes pasos:

1. En la ventana de STEP 7-Micro/WIN 32, haga clic en el icono "Comunicación" o elija el comando de menú **Ver > Comunicación**. Aparecerá el cuadro de diálogo "Configurar comunicación".
2. En el cuadro de diálogo "Configurar comunicación", haga doble clic en el icono del cable PC/PPI. Aparecerá el cuadro de diálogo "Ajustar interface PG/PC" (v. fig. 3-3).
3. Haga clic en el botón "Propiedades" para acceder al cuadro de diálogo donde se visualizan las propiedades del interface (v. fig. 3-4). Verifique las propiedades. La velocidad de transferencia debe estar ajustada a 9.600 bit/s.
4. Para obtener información sobre cómo cambiar los parámetros estándar, consulte el apartado 7.3 del capítulo 7.

---

### Nota

Si el hardware que está utilizando no aparece en la lista visualizada en el cuadro de diálogo "Ajustar interface PG/PC", deberá instalar el hardware correcto (v. "Instalar y desinstalar tarjetas" en el apartado 7.2).

---



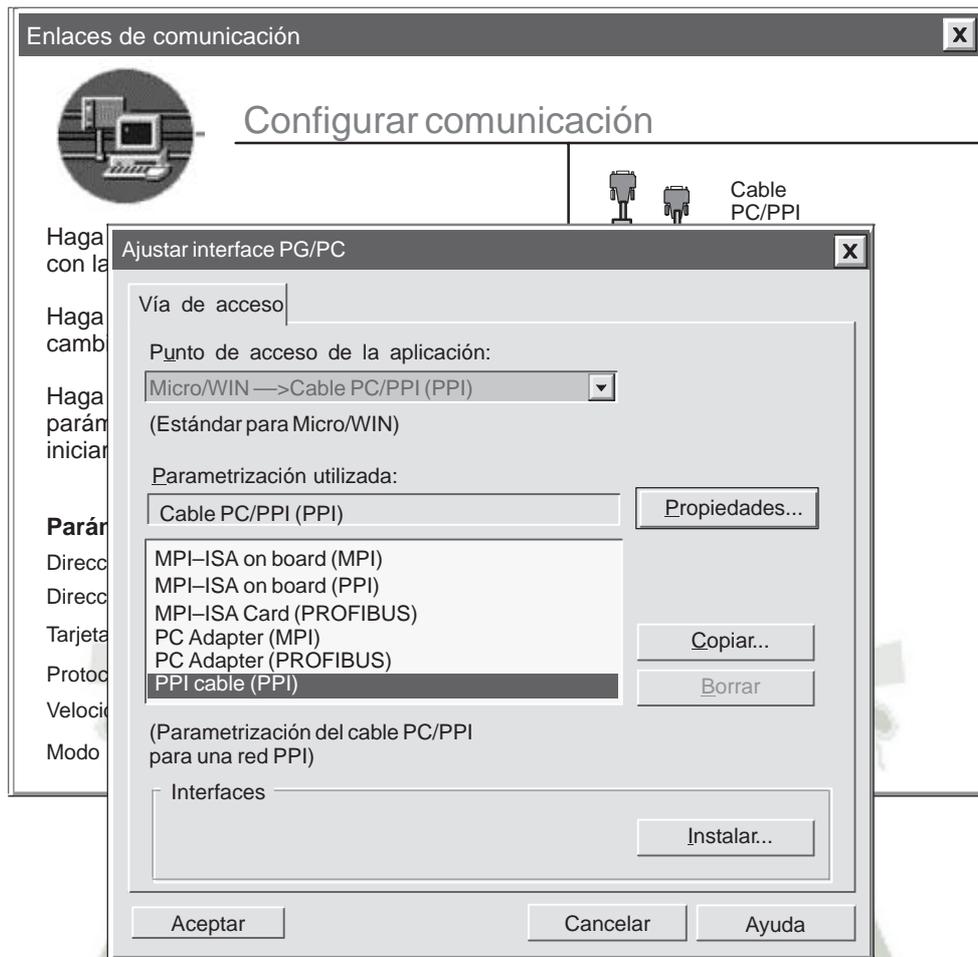


Figura 3-3 Ajustes en el cuadro de diálogo "Interface PG/PC"

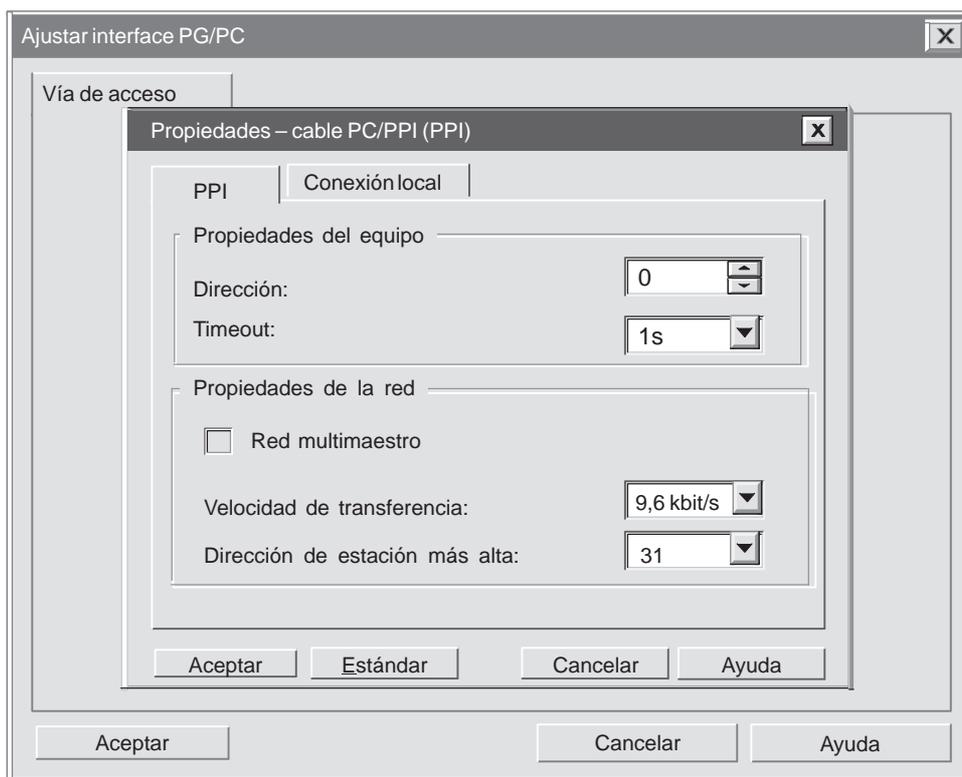


Figura 3-4 Cuadro de diálogo "Propiedades - Interface PG/PC"



### 3.4 Cómo establecer la comunicación con la CPU S7-200

Tras haber instalado el software STEP 7-Micro/WIN 32 en el PC y determinado los ajustes de comunicación del cable PC/PPI, podrá establecer un enlace con la CPU S7-200. (Si está utilizando una unidad de programación, STEP 7-Micro/WIN 32 ya estará instalado).

Para establecer la comunicación con la CPU S7-200, siga los siguientes pasos:

1. En la pantalla de STEP 7-Micro/WIN 32, haga clic en el icono "Comunicación" o elija el comando de menú **Ver > Comunicación**. Aparecerá el cuadro de diálogo "Configurar comunicación" donde se indica que no hay ninguna CPU conectada.
2. En el cuadro de diálogo "Configurar comunicación", haga doble clic en el icono "Actualizar". STEP 7-Micro/WIN 32 comprueba si hay CPUs S7-200 (estaciones) conectadas. Por cada estación conectada aparecerá un icono de CPU en el cuadro de diálogo "Configurar comunicación" (v. fig. 3-5).
3. Haga doble clic en la estación con la que desea establecer la comunicación. Como podrá apreciar, los parámetros de comunicación visualizados en el cuadro de diálogo corresponden a la estación seleccionada.
4. Así queda establecido el enlace con la CPU S7-200.

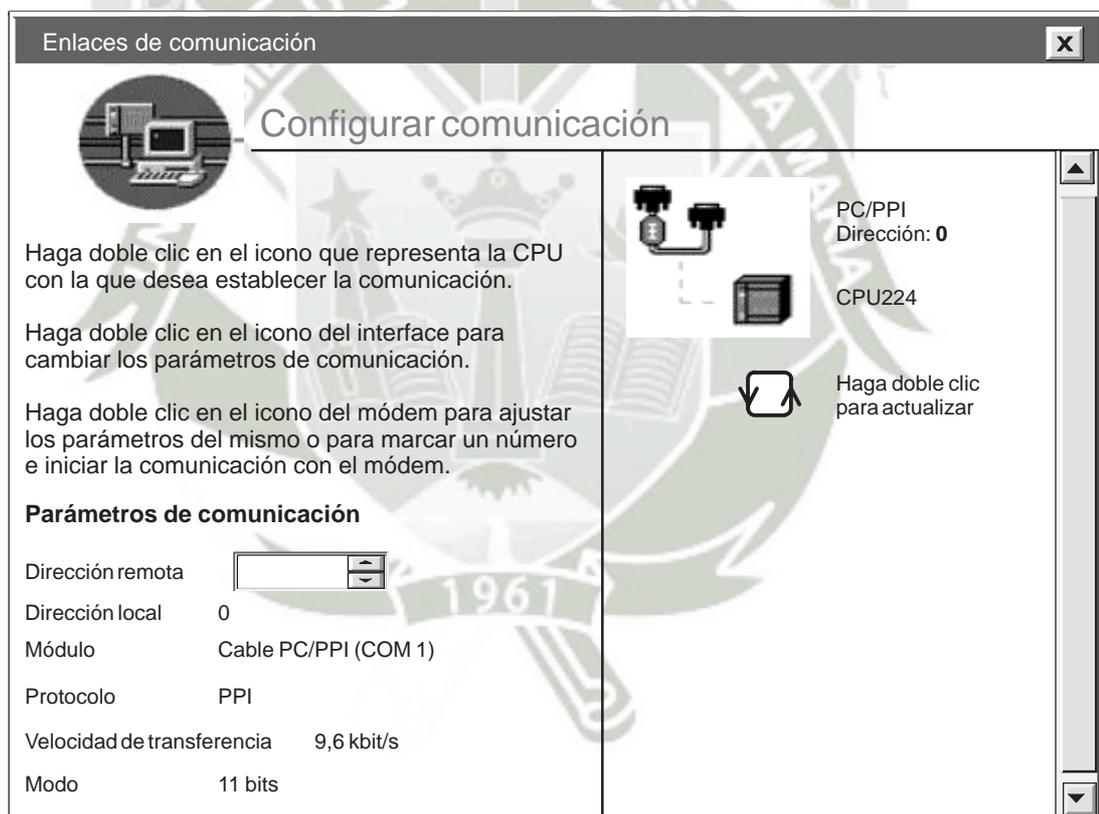


Figura 3-5 Cuadro de diálogo "Configurar comunicación"

### 3.5 Cómo cambiar los parámetros de comunicación de la CPU

Tras haber establecido un enlace con la CPU S7-200 puede verificar o cambiar los parámetros de comunicación de la CPU.

Para cambiar los parámetros de comunicación, siga los siguientes pasos:

1. En la barra de navegación, haga clic en el icono "Bloque de sistema" o elija el comando de menú **Ver > Bloque de sistema**.
2. Aparecerá el cuadro de diálogo "Bloque de sistema". Haga clic en la ficha "Puerto(s)" (v. fig. 3-6). El ajuste estándar de la dirección de estación es 2 y el de la velocidad de transferencia es de 9,6 kbit/s.
3. Haga clic en "Aceptar" para conservar dichos parámetros. Si desea modificar la parametrización, efectúe los cambios deseados, haga clic en el botón "Aplicar" y, por último, en el botón "Aceptar".
4. En la barra de herramientas, haga clic en el botón "Cargar en CPU" para cargar los cambios en la CPU.
5. Así se adopta la parametrización deseada para la comunicación.

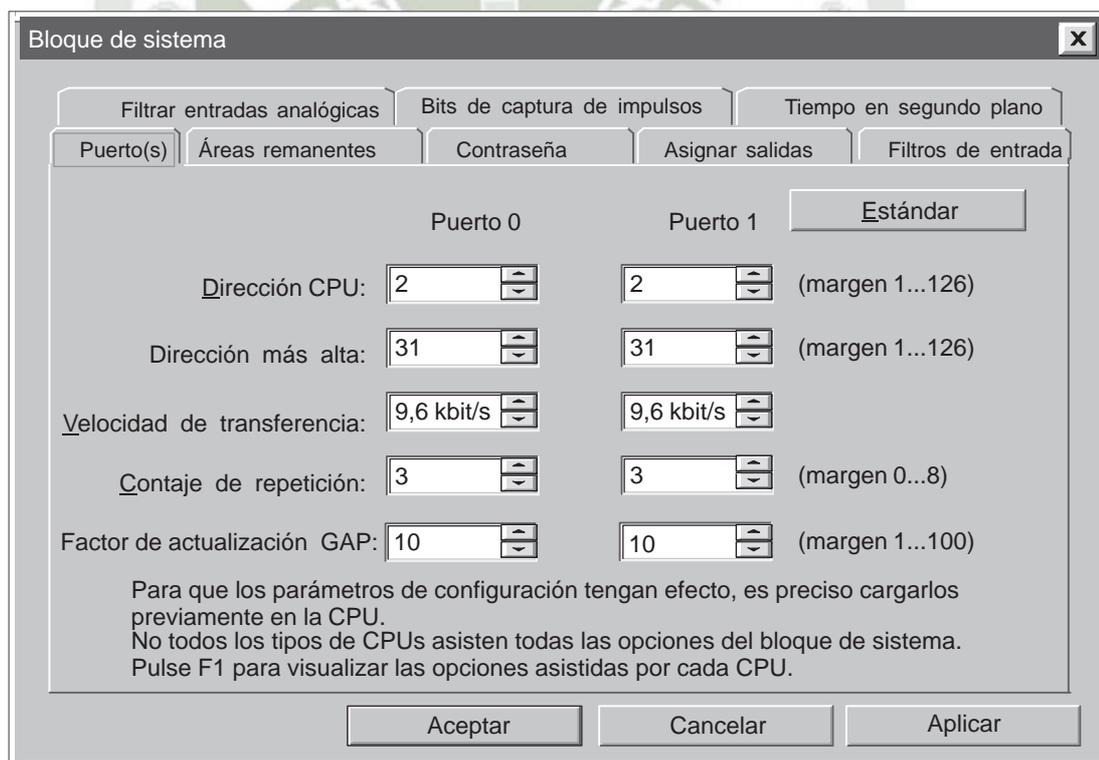


Figura 3-6 Cambiar los parámetros de comunicación

# Nociones básicas para programar una CPU S7-200

# 4

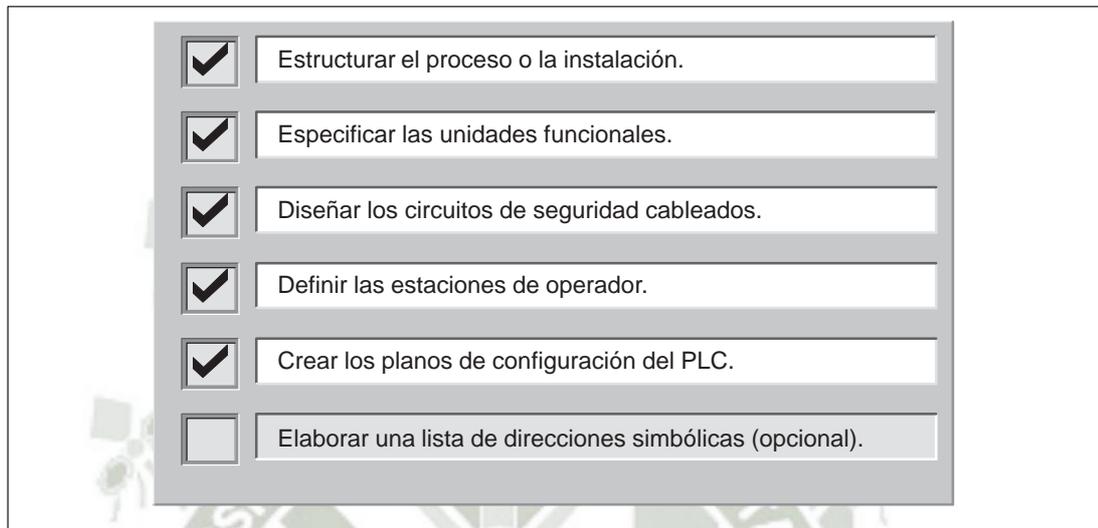
Antes de comenzar a programar aplicaciones para la CPU S7-200, es recomendable que se familiarice con algunas funciones básicas de la misma.

## Índice del capítulo

Apartado	Descripción	Página
4.1	Crear una solución de automatización con un Micro-PLC	4-2
4.2	Programas S7-200	4-5
4.3	Lenguajes y editores de programación S7-200	4-6
4.4	Diferencias entre las operaciones SIMATIC e IEC 1131-3	4-10
4.5	Elementos básicos para estructurar un programa	4-18
4.6	El ciclo de la CPU	4-22
4.7	Ajustar el modo de operación de la CPU	4-25
4.8	Definir una contraseña para la CPU	4-27
4.9	Comprobar y observar el programa	4-30
4.10	Eliminar errores de las CPUs S7-200	4-36

## 4.1 Crear una solución de automatización con un Micro-PLC

Existen diversos métodos para crear una solución de automatización con un Micro-PLC. En el presente apartado se indican algunas reglas generales aplicables a numerosos proyectos. No obstante, también deberá tener en cuenta las reglas de su empresa y su propia experiencia. La figura 4-1 muestra los pasos básicos al respecto.



<input checked="" type="checkbox"/>	Estructurar el proceso o la instalación.
<input checked="" type="checkbox"/>	Especificar las unidades funcionales.
<input checked="" type="checkbox"/>	Diseñar los circuitos de seguridad cableados.
<input checked="" type="checkbox"/>	Definir las estaciones de operador.
<input checked="" type="checkbox"/>	Crear los planos de configuración del PLC.
<input type="checkbox"/>	Elaborar una lista de direcciones simbólicas (opcional).

Figura 4-1 Pasos básicos para crear una solución de automatización

### Estructurar el proceso o la instalación

Divida el proceso o la instalación en secciones independientes. Dichas secciones determinan los límites entre los diversos sistemas de automatización e influyen en las descripciones de las áreas de funciones y en la asignación de recursos.

### Especificar las unidades funcionales

Describa las funciones de cada sección del proceso o de la instalación. Incorpore los siguientes aspectos:

- Entradas y salidas (E/S)
- Descripción del funcionamiento de la operación
- Condiciones de habilitación (es decir, los estados que se deben alcanzar antes de ejecutar una función) de cada actuador (electroválvulas, motores, accionamientos, etc).
- Descripción del interface de operador
- Interfaces con otras secciones del proceso o de la instalación

### Diseñar los circuitos de seguridad cableados

Determine qué aparatos requieren un cableado permanente por motivos de seguridad. Si fallan los sistemas de automatización, puede ocurrir un arranque inesperado o un cambio en el funcionamiento de las máquinas. En tal caso, pueden producirse heridas graves o daños materiales. Por lo tanto, es preciso utilizar dispositivos de protección contra sobrecargas electromecánicas que funcionen independientemente de la CPU, evitando así las condiciones inseguras.

Para diseñar los circuitos de seguridad cableados:

- Defina el funcionamiento erróneo o inesperado de los actuadores que pudieran causar peligros.
- Defina las condiciones que garanticen un funcionamiento seguro y determine cómo detectar dichas condiciones, independientemente de la CPU.
- Defina cómo la CPU y los módulos de ampliación deberán influir en el proceso cuando se conecte y desconecte la alimentación, así como al detectarse errores. Estas informaciones se deberán utilizar únicamente para diseñar el funcionamiento normal y el funcionamiento anormal esperado, sin poderse aplicar para fines de seguridad.
- Prevea dispositivos de parada de emergencia manual o de protección contra sobrecargas electromagnéticas que impidan un funcionamiento peligroso, independientemente de la CPU.
- Desde los circuitos independientes, transmita informaciones de estado apropiadas a la CPU para que el programa y los interfaces de operador dispongan de los datos necesarios.
- Defina otros requisitos adicionales de seguridad para que el proceso se lleve a cabo de forma segura y fiable.

### Definir las estaciones de operador

Conforme a las funciones exigidas, cree planos de las estaciones de operador incorporando los siguientes puntos:

- Panorámica de la ubicación de todas las estaciones de operador con respecto al proceso o a la instalación.
- Disposición mecánica de los aparatos (pantalla, interruptores, lámparas, etc) de la estación de operador.
- Esquemas eléctricos con las correspondientes entradas y salidas de la CPU o de los módulos de ampliación.

### Crear los planos de configuración del PLC

Conforme a las funciones exigidas, cree planos de configuración del sistema de automatización incorporando los siguientes puntos:

- Ubicación de todas las CPUs y de todos los módulos de ampliación con respecto al proceso o a la instalación.
- Disposición mecánica de las CPUs y de los módulos de ampliación (incluyendo armarios, etc.)
- Esquemas eléctricos de todas las CPUs y de los módulos de ampliación (incluyendo los números de referencia, las direcciones de comunicación y las direcciones de las entradas y salidas).

### Elaborar una lista de nombres simbólicos

Si desea utilizar nombres simbólicos para el direccionamiento, elabore una lista de nombres simbólicos para las direcciones absolutas. Incluya no sólo las entradas y salidas físicas, sino también todos los demás elementos que utilizará en su programa.

## 4.2 Programas S7-200

### Referencias a las entradas y salidas en el programa

El funcionamiento básico de la CPU S7-200 es muy sencillo:

- La CPU lee el estado de las entradas.
- El programa almacenado en la CPU utiliza dichas entradas para evaluar la lógica. Durante la ejecución del programa, la CPU actualiza los datos.
- La CPU escribe los datos en las salidas.

La figura 4-2 muestra cómo se procesa un esquema de circuitos simple en una CPU S7-200. En este ejemplo, el estado del interruptor de la estación de operador para abrir la electroválvula de vaciado se suma a los estados de otras entradas. El resultado obtenido establece entonces el estado de la salida que corresponde a dicha electroválvula.

La CPU procesa el programa cíclicamente, leyendo y escribiendo los datos.

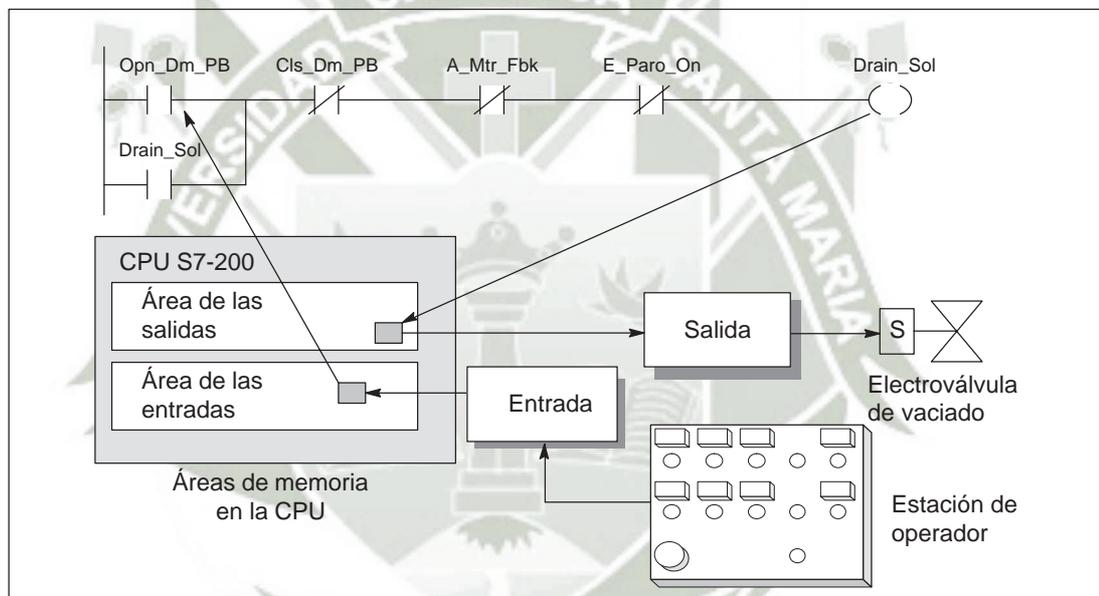


Figura 4-2 Referencias a las entradas y salidas en el programa

### 4.3 Lenguajes y editores de programación S7-200

Las CPUs S7-200 ofrecen numerosos tipos de operaciones que permiten solucionar una gran variedad de tareas de automatización. Disponen de dos juegos básicos de operaciones, a saber: SIMATIC e IEC 1131-3. El software de programación STEP 7-Micro/WIN 32 permite elegir entre diferentes editores para crear programas de control utilizando dichas operaciones. Por ejemplo, puede ser que Ud. prefiera crear programas en un entorno de programación gráfico, en tanto que otra persona que trabaje en su misma empresa opta por utilizar un editor textual, similar al lenguaje ensamblador.

Para crear sus programas debe hacer dos selecciones básicas:

- El tipo de juego de operaciones a utilizar (SIMATIC o IEC 1131-3).
- El tipo de editor a utilizar (Lista de instrucciones, Esquema de contactos o Diagrama de funciones)

La figura 4-1 muestra las combinaciones posibles de juegos de operaciones y de editores S7-200.

Tabla 4-1 Juegos de operaciones y editores SIMATIC e IEC 1131-3

Juego de operaciones SIMATIC	Juegos de operaciones IEC 1131-3
Editor AWL (Lista de instrucciones)	no disponible
Editor KOP (Esquema de contactos)	Editor LD (Diagrama de escalera)
Editor FUP (Diagrama de funciones)	Editor FBD (Diagrama de bloques funcionales)

#### Editor AWL (Lista de instrucciones)

El editor AWL (Lista de instrucciones) de STEP 7-Micro/WIN 32 permite crear programas de control introduciendo la nemotécnica de las operaciones. Por lo general, el editor AWL se adecúa especialmente para los programadores expertos ya familiarizados con los sistemas de automatización (PLCs) y con la programación lógica. El editor AWL también permite crear ciertos programas que, de otra forma, no se podrían programar con los editores KOP ni FUP. Ello se debe a que AWL es el lenguaje nativo de la CPU, a diferencia de los editores gráficos en los que son aplicables ciertas restricciones para poder dibujar los diagramas correctamente. La figura 4-3 muestra un ejemplo de un programa AWL.

AWL	
NETWORK	
LD	I0.0
LD	I0.1
LD	I2.0
A	I2.1
OLD	
ALD	
=	Q5.0

Figura 4-3 Ejemplo de un programa AWL

Como muestra la figura 4-3, esta forma textual es muy similar a la programación en lenguaje ensamblador. La CPU ejecuta cada operación en el orden determinado por el programa, de arriba a abajo, reiniciando luego arriba nuevamente. AWL y el lenguaje ensamblador también son similares en otro sentido. Las CPUs S7-200 utilizan una pila lógica para resolver la lógica de control (v. figura 4-4). Los editores KOP y FUP insertan automáticamente las operaciones necesarias para procesar la pila. En AWL, es el usuario quien debe insertar dichas operaciones.

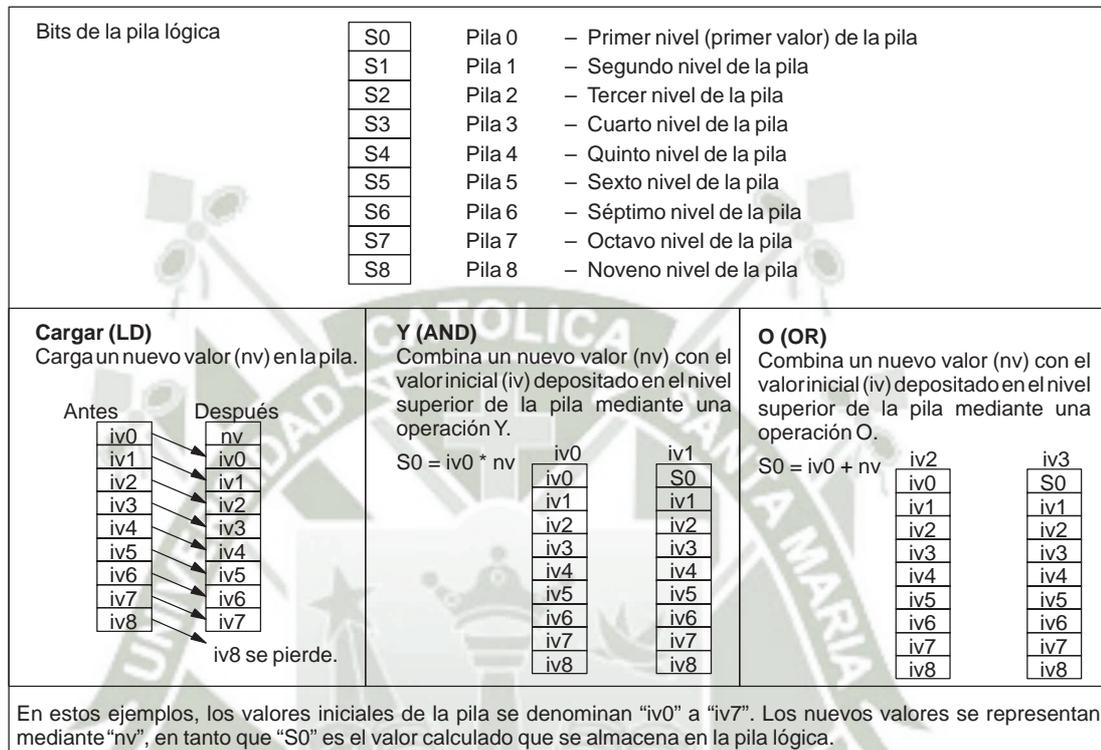


Figura 4-4 Pila lógica de la CPU S7-200

A continuación se indican los aspectos principales a considerar cuando se desee utilizar el editor AWL:

- El lenguaje AWL es más apropiado para los programadores expertos.
- En algunos casos, AWL permite solucionar problemas que no se podrían resolver muy fácilmente con los editores KOP o FUP.
- El editor AWL sólo se puede utilizar con el juego de operaciones SIMATIC.
- En tanto que el editor AWL se puede utilizar siempre para ver o editar un programa creado con los editores KOP o FUP SIMATIC, lo contrario no es posible en todos los casos. Los editores KOP o FUP SIMATIC no siempre se pueden utilizar para visualizar un programa que se haya creado en AWL.

## Editor KOP (Esquema de contactos)

El editor KOP (Esquema de contactos) de STEP 7-Micro/WIN 32 permite crear programas con componentes similares a los elementos de un esquema de circuitos. KOP es probablemente el lenguaje predilecto de numerosos programadores y encargados del mantenimiento de sistemas de automatización. Básicamente, los programas KOP hacen que la CPU emule la circulación de corriente eléctrica desde una fuente de alimentación, a través de una serie de condiciones lógicas de entrada que, a su vez, habilitan condiciones lógicas de salida. Por lo general, la lógica se divide en unidades pequeñas y de fácil comprensión llamadas “segmentos” o “networks”. El programa se ejecuta segmento por segmento, de izquierda a derecha y luego de arriba a abajo. Tras alcanzar la CPU el final del programa, comienza nuevamente en la primera operación del mismo.

La figura 4-5 muestra un ejemplo de un programa KOP.

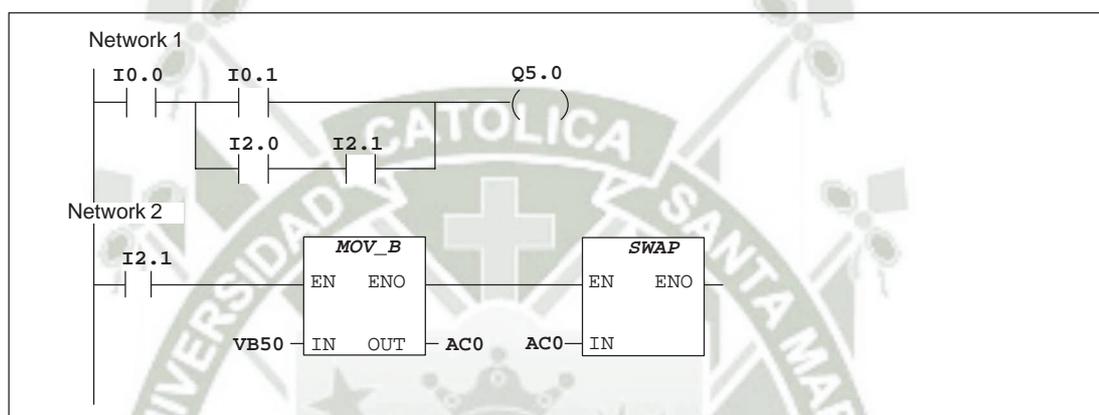


Figura 4-5 Ejemplo de un programa KOP

Las operaciones se representan mediante símbolos gráficos que incluyen tres formas básicas. Como muestra la figura 4-5, se pueden conectar en serie incluso varias operaciones de cuadros.

- Contactos - representan condiciones lógicas de “entrada” tales como interruptores, botones, condiciones internas, etc.
- Bobinas - representan condiciones lógicas de “salida” tales como lámparas, arrancadores de motor, relés interpuestos, condiciones internas de salida, etc.
- Cuadros - representan operaciones adicionales tales como temporizadores, contadores u operaciones aritméticas.

A continuación se indican los aspectos principales a considerar cuando se desee utilizar el editor KOP:

- El lenguaje KOP les facilita el trabajo a los programadores principiantes.
- La representación gráfica es a menudo fácil de comprender, siendo popular en el mundo entero.
- El editor KOP se puede utilizar con los juegos de operaciones SIMATIC e IEC 1131-3.
- El editor AWL siempre se puede utilizar para visualizar un programa creado en SIMATIC KOP.

### Editor FUP (Diagrama de funciones)

El editor FUP (Diagrama de funciones) de STEP 7-Micro/WIN 32 permite visualizar las operaciones en forma de cuadros lógicos similares a los circuitos de puertas lógicas. En FUP no existen contactos ni bobinas como en el editor KOP, pero sí hay operaciones equivalentes que se representan en forma de cuadros. La lógica del programa se deriva de las conexiones entre dichas operaciones de cuadro. Ello significa que la salida de una operación (p.ej. un cuadro AND) se puede utilizar para habilitar otra operación (p.ej. un temporizador) para crear la lógica de control necesaria. Dichas conexiones permiten solucionar numerosos problemas lógicos.

La figura 4-6 muestra un ejemplo de un programa creado con el editor FUP.

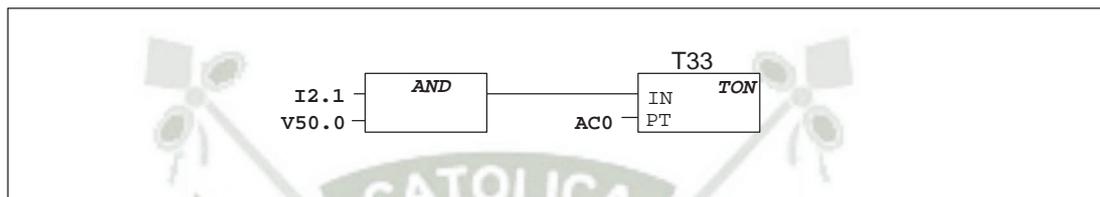


Figura 4-6 Ejemplo de un programa FUP

A continuación se indican los aspectos principales a considerar cuando se desee utilizar el editor FUP:

- El estilo de representación en forma de puertas gráficas se adecúa especialmente para observar el flujo del programa.
- El editor FUP se puede utilizar con los juegos de operaciones SIMATIC e IEC 1131-3.
- El editor AWL siempre se puede utilizar para visualizar un programa creado en SIMATIC FUP.

## 4.4 Diferencias entre las operaciones SIMATIC e IEC 1131-3

### Juego de operaciones SIMATIC

La mayoría de los sistemas de automatización ofrecen los mismos tipos básicos de operaciones, pero por lo general existen pequeñas diferencias en cuanto al aspecto, al funcionamiento, etc. de los productos de los distintos fabricantes. El juego de operaciones SIMATIC se ha diseñado para los sistemas de automatización S7-200. Un gran número de estas operaciones tienen un aspecto y un funcionamiento diferentes si se comparan con otras marcas de sistemas de automatización (autómatas programables). A continuación se indican los aspectos principales a considerar cuando se desee utilizar el juego de operaciones SIMATIC:

- Por lo general, el tiempo de ejecución de las operaciones SIMATIC es más breve.
- El juego de operaciones SIMATIC se puede utilizar con los tres editores (KOP, AWL y FUP).

### Juego de operaciones IEC 1131-3

La Comisión Electrotécnica Internacional (CEI) o International Electrotechnical Commission (IEC) es un organismo mundial que desarrolla normas globales para todos los campos de la electrotécnica. (Nota: En el presente manual se utilizan las siglas inglesas de dicho organismo). Durante los últimos años, dicha comisión ha desarrollado una norma que se dedica a numerosos aspectos de la programación de autómatas programables (denominados "sistemas de automatización" en la terminología SIMATIC). El objetivo de dicha norma es que los diferentes fabricantes de autómatas programables ofrezcan operaciones similares tanto en su aspecto como en su funcionamiento. Existen algunas diferencias básicas entre los juegos de operaciones SIMATIC e IEC 1131-3.

- El juego de operaciones IEC 1131-3 se limita a las operaciones estándar comunes entre los fabricantes de autómatas programables. Algunas operaciones incluidas en el juego SIMATIC no están normalizadas en la norma IEC 1131-3. (Éstas se pueden utilizar en calidad de operaciones no normalizadas. No obstante, entonces el programa ya no será absolutamente compatible con la norma IEC 1131-3).
- Algunos cuadros aceptan varios formatos de datos. A menudo, ésto se denomina sobrecarga. Por ejemplo, en lugar de tener cuadros aritméticos por separado, tales como ADD\_I (Sumar enteros), ADD\_R (Sumar reales) etc., la operación ADD definida en la norma IEC 1131-3 examina el formato de los datos a sumar y selecciona automáticamente la operación correcta en la CPU. Así se puede ahorrar tiempo al diseñar los programas.
- Si se utilizan las operaciones IEC-1131, se comprueba automáticamente si los parámetros de la operación corresponden al formato de datos correcto. Dicha comprobación no es obvia para el usuario. Por ejemplo, si se ha intentado introducir un valor de entero en una operación para la que se deba utilizar un valor binario (on/off), se indica un error. Esta función permite reducir los errores de sintaxis de programación.

A continuación se indican los aspectos principales a considerar cuando se desee utilizar las operaciones IEC 1131-3:

- Por lo general, es más fácil aprender a crear programas para autómatas programables de diferentes fabricantes.
- Aunque se dispone de menos operaciones (conforme a lo especificado en la norma), también se pueden utilizar numerosas operaciones SIMATIC.
- El funcionamiento de algunas operaciones IEC-1131 es diferente al de sus equivalentes en SIMATIC (temporizadores, contadores, multiplicación, división, etc.)
- Es posible que el tiempo de ejecución de las operaciones IEC-1131 sea más largo.
- Las operaciones sólo se pueden utilizar en los editores KOP y FUP.
- La norma IEC 1131-3 especifica que las variables se deben declarar tipificadas, soportando que el sistema verifique el tipo de datos.

### Tipificación de variables en SIMATIC e IEC 1131-3

Cada operación o subrutina parametrizada se identifica en SIMATIC e IEC 1131-3 mediante una definición precisa denominada *signatura*. En todas las operaciones estándar, los tipos de datos permitidos para cada operando de la operación se indican en la *signatura*. En el caso de las subrutinas parametrizadas, el usuario crea la *signatura* de la subrutina en la tabla de variables locales.

STEP 7-Micro/WIN 32 implementa una verificación simple de los tipos de datos en el modo SIMATIC y una verificación precisa en el modo IEC 1131-3. Cuando un tipo de datos se especifica para una variable (bien sea local o global), STEP 7-Micro/WIN 32 verifica que el tipo de datos del operando concuerde con la *signatura* de la operación en el nivel indicado. En la tabla 4-2 se definen los tipos de datos simples y la tabla 4-3 muestra los tipos de datos complejos disponibles en STEP 7-Micro/WIN 32.

Tabla 4-2 Tipos de datos simples IEC 1131-3

Tipos de datos simples	Descripción	Margen de datos
BOOL	Valor booleano	0 a 1
BYTE	Byte sin signo	0 a 255
WORD	Entero sin signo	0 a 65.535
INT	Entero con signo	-32768 a +32767
DWORD	Entero doble sin signo	0 a $2^{32} - 1$
DINT	Entero doble con signo	$-2^{31}$ a $+2^{31} - 1$
REAL	Valor de 32 bits en coma flotante (IEEE)	$-10^{38}$ a $+10^{38}$

Tabla 4-3 Tipos de datos complejos IEC 1131-3

Tipos de datos complejos	Descripción	Margen de direcciones
TON <sup>1</sup>	Temporizador con retardo al conectar	1 ms T32, T96 10 ms T33 a T36, T97 a T100 100 ms T37 a T63, T101 a T255
TOF	Temporizador con retardo al desconectar	1 ms T32, T96 10 ms T33 a T36, T97 a T100 100 ms T37 a T63, T101 a T255
TP	Temporizador por impulsos (v. nota 1)	1 ms T32, T96 10 ms T33 a T36, T97 a T100 100 ms T37 a T63, T101 a T255
CTU	Contador ascendente	0 a 255
CTD	Contador descendente	0 a 255
CTUD	Contador ascendente–descendente	0 a 255
SR	Bloque funcional biestable (posicionar dominante)	—
RS	Bloque funcional biestable (rear-mar dominante)	—
<sup>1</sup> El bloque funcional del temporizador por impulsos utiliza temporizadores TON para ejecutar la función de salida de impulsos. Ello reduce el total disponible de temporizadores TON.		

**Verificación de los tipos de datos** Hay tres niveles de verificación de los tipos de datos: verificación precisa, verificación simple y sin verificación.

**Verificación precisa de los tipos de datos** En este modo, el tipo de datos del parámetro debe concordar con el tipo de datos del símbolo o de la variable. Cada parámetro formal tiene sólo un tipo de datos (con excepción de las operaciones sobrecargadas). Por ejemplo, el parámetro IN de una operación SRW (Desplazar palabra a la derecha) tiene el tipo de datos WORD. Sólo las variables asignadas al tipo de datos WORD se compilarán correctamente. Las variables tipificadas como INT no serán válidas para los parámetros de operaciones de tipo WORD si se exige la verificación precisa de los tipos de datos.

La verificación precisa de los tipos de datos se efectúa sólo en el modo IEC 1131-3 (v. tabla 4-4).

Tabla 4-4 Verificación precisa de los tipos de datos seleccionados por el usuario y sus equivalentes

Tipo de datos seleccionado por el usuario	Tipo de datos equivalente
BOOL	BOOL
BYTE	BYTE
WORD	WORD
INT	INT
DWORD	DWORD
DINT	DINT
REAL	REAL

**Verificación simple de los tipos de datos** En este modo, cuando se le asigna un tipo de datos a un símbolo o a una variable, se asocia automáticamente también a todos los tipos cuyo tamaño binario concuerde con el tipo de datos seleccionado. Por ejemplo, si se selecciona el tipo de datos DINT, la variable local asignará automáticamente el tipo de datos DWORD, puesto que ambos son tipos de datos de 32 bits. El tipo de datos REAL no se asigna automáticamente, aunque se trata también de un tipo de datos de 32 bits. En el caso del tipo de datos REAL, se considera que no tiene ningún otro tipo de datos equivalente, siendo siempre único. La verificación simple de los tipos de datos se efectúa sólo en el modo SIMATIC cuando se utilizan variables locales (v. tabla 4-5).

Tabla 4-5 Verificación simple de los tipos de datos seleccionados por el usuario y sus equivalentes

Tipo de datos seleccionado por el usuario	Tipo de datos equivalente
BOOL	BOOL
BYTE	BYTE
WORD	WORD, INT
INT	WORD, INT
DWORD	DWORD, DINT
DINT	DWORD, DINT
REAL	REAL

**Sin verificación de los tipos de datos** Este modo se encuentra disponible únicamente para las variables globales SIMATIC que no permitan seleccionar los tipos de datos. Todos los tipos de datos de tamaño equivalente se asignan automáticamente al símbolo. Por ejemplo, a un símbolo que se le haya asignado la dirección VD100, STEP 7-Micro/WIN 32 le asignará automáticamente los tipos de datos que muestra la tabla 4-6.

Tabla 4-6 Tipo de datos de tamaño determinado para los símbolos globales SIMATIC

Dirección seleccionada por el usuario	Tipo de datos equivalente
V0.0	BOOL
VB0	BYTE
VW0	WORD, INT
VD0	DWORD, DINT, REAL

### Ventajas de la verificación de los tipos de datos

La verificación de los tipos de datos contribuye a evitar errores de programación. Si una operación soporta números con signo y se utiliza un número sin signo para el operando de esa operación, STEP 7-Micro/WIN 32 marcará el número sin signo. Por ejemplo, la comparación  $< I$  es una operación con signo.  $-1$  es menor que 0 en el caso de los operandos con signo. No obstante, si la operación  $< I$  puede soportar tipos de datos sin signo, el programador debe tener en cuenta que durante la ejecución del programa, un valor sin signo de 40.000 es menor que 0 en la operación  $< I$ .



#### Precaución

Verifique que la utilización de números sin signo en las operaciones con signo no cruce el límite entre los números positivos y negativos.

En caso contrario se pueden producir resultados impredecibles en el programa y en el funcionamiento del sistema de automatización.

Tenga siempre en cuenta que el número sin signo de una operación con signo no cruce el límite entre los números positivos y negativos.

En resumen, la verificación precisa de los tipos de datos en el modo de edición IEC 1131-3 contribuye a identificar dichos errores durante la compilación, indicando si se utilizan tipos de datos no válidos para la operación. Esta función no se encuentra disponible en los editores SIMATIC.

### Seleccionar entre los modos de programación SIMATIC e IEC 1131-3

Puesto que, en contraposición a SIMATIC, el modo IEC 1131-3 exige una tipificación precisa de los datos, STEP 7-Micro/WIN 32 no permite transferir programas entre los dos modos de edición. Por tanto, el usuario debe elegir uno de dichos modos.

## Operaciones sobrecargadas

Las operaciones sobrecargadas soportan diversos tipos de datos. No obstante, se efectúa aún una verificación precisa de los tipos de datos, puesto que todos los tipos de datos de los operandos deben concordar para que la operación se pueda compilar correctamente. La tabla 4-7 muestra un ejemplo de la operación ADD sobrecargada (IEC).

Tabla 4-7 Ejemplo de la operación ADD sobrecargada (IEC)

Operación	Tipos de datos admisibles (verificación precisa)	Tipos de datos admisibles (verificación simple)	Operación compilada
ADD	INT	WORD, INT	ADD_I (Sumar enteros)
ADD	DINT	DWORD, DINT	ADD_D (Sumar enteros dobles)
ADD	REAL	REAL	ADD_R (Sumar reales)

Si todos los operandos tienen el tipo de datos DINT, el compilador generará una operación Sumar enteros dobles. Si se mezclan los tipos de datos de la operación sobrecargada, ocurrirá un error de compilación. El nivel de verificación de los tipos de datos determina lo que es válido o no. El ejemplo siguiente generará un error de compilación sólo si se efectúa una verificación precisa de los tipos de datos, mas no si se realiza una verificación simple.

ADD IN1 = INT, IN2 = WORD, IN3 = INT

Verificación precisa: error de compilación

Verificación simple: se compila a ADD\_I (Sumar enteros)

Al igual que en el ejemplo del contacto de comparación, la verificación simple de los tipos de datos no evitará que ocurran errores de programación en el tiempo de ejecución. Si se efectúa una verificación simple de los tipos de datos, el compilador no detectará el siguiente error de programación: ADD 40000, 1 será un número negativo, mas no un valor 40.001 sin signo.

## Utilizar el direccionamiento directo en IEC para operaciones sobrecargadas

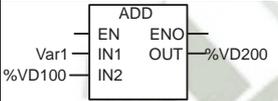
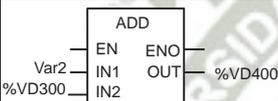
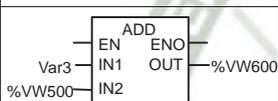
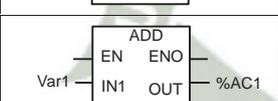
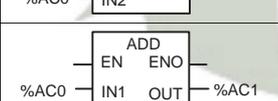
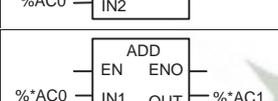
El modo de programación IEC 1131-3 permite utilizar direcciones directas como parte de la configuración de los parámetros de la operación. En los parámetros se pueden utilizar tanto variables como direcciones de la memoria. Recuerde que las direcciones directas no contienen información explícita sobre el tipo de datos. Además, la información de tipo no se puede deducir de ninguna de las operaciones IEC sobrecargadas, toda vez que éstas aceptan diversos tipos de datos.

Los tipos de datos de los parámetros representados directamente se determinan examinando otros parámetros tipificados incluidos en la operación. Si se configura que un parámetro utilice una variable de un tipo específico, se supone que todos los parámetros representados directamente sean de ese mismo tipo. Las tablas 4-8 y 4-9 muestran ejemplos de tipos de datos de parámetros representados directamente.

Tabla 4-8 Ejemplo de tipos de datos para el direccionamiento directo

Nombre	Dirección	Tipo de datos	Comentario
Var1		REAL	Esta es una variable en coma flotante.
Var2		DINT	Esta es una variable de entero doble.
Var3		INT	Esta es una variable de entero.

Tabla 4-9 Ejemplos de direccionamiento directo en operaciones sobrecargadas

Ejemplo	Descripción
	Se supone que VW100 y VW200 son de tipo REAL, puesto que el tipo de Var1 es REAL.
	Se supone que VW100 y VW200 son de tipo REAL, puesto que el tipo de Var1 es REAL.
	Se supone que VW500 y VW600 son de tipo INT, puesto que el tipo de Var3 es INT.
	Se supone que AC0 y AC1 son de tipo REAL, puesto que el tipo de Var1 es REAL.
	Esta configuración no es válida, puesto que el tipo no se puede determinar. El tipo de datos de los acumuladores podría ser uno cualquiera.
	Esta configuración no es válida, puesto que el tipo no se puede determinar. El tipo de datos de los punteros de los acumuladores podría ser uno cualquiera.

### Utilizar operaciones de conversión

Las operaciones de conversión permiten transferir valores de un tipo de datos a otro. STEP 7-Micro/WIN 32 soporta las operaciones de conversión que figuran en la tabla 4-10 para transferir valores entre los tipos de datos simples.

Tabla 4-10 Operaciones de conversión

Operación de conversión	Operandos admisibles en la verificación precisa de los tipos de datos	Operandos admisibles en la verificación simple de los tipos de datos
BYTE a INT	IN: BYTE OUT: INT	IN: BYTE OUT: WORD, INT
INT a BYTE	IN: INT OUT: BYTE	IN: WORD, INT OUT: BYTE
INT a DINT	IN: DINT OUT: DINT	IN: WORD, INT OUT: DWORD, DINT
DINT a INT	IN: DINT OUT: INT	IN: DWORD, DINT OUT: WORD, INT
DINT a REAL	IN: DINT OUT: REAL	IN: DWORD, DINT OUT: REAL
REAL a DINT (ROUND)	IN: REAL OUT: DINT	IN: REAL OUT: DWORD, DINT

En el modo de edición IEC 1131-3, la operación MOVE sobrecargada se puede utilizar para convertir entre INT y WORD, DINT y DWORD. La operación MOVE permite transferir tipos de datos de un mismo tamaño, sin que el compilador genere errores (v. tabla 4-11).

Tabla 4-11 Utilizar la operación sobrecargada MOVE

MOVE sobrecargada (IEC 1131-3)	IN	OUT
MOVE (INT a WORD)	INT	WORD
MOVE (WORD a INT)	WORD	INT
MOVE (DINT a DWORD)	DINT	DWORD
MOVE (DWORD a DINT)	DWORD	DINT

## 4.5 Elementos básicos para estructurar un programa

La CPU S7-200 ejecuta continuamente el programa para controlar una tarea o un proceso. El programa se crea con STEP 7-Micro/WIN 32 y se carga en la CPU. Desde el programa principal se pueden invocar diversas subrutinas o rutinas de interrupción.

### Estructurar el programa

Los programas para la CPU S7-200 comprenden tres partes básicas: el programa principal, las subrutinas (opcional) y las rutinas de interrupción (opcional). Un programa S7-200 se divide en los siguientes elementos:

- Programa principal: En esta parte del programa se disponen las operaciones que controlan la aplicación. Las operaciones del programa principal se ejecutan de forma secuencial en cada ciclo de la CPU.
- Subrutinas: Estos elementos opcionales del programa se ejecutan sólo cuando se llaman desde el programa principal.
- Rutinas de interrupción: Estos elementos opcionales del programa se ejecutan cada vez que ocurra el correspondiente evento de interrupción.

### Programa de ejemplo con subrutinas y rutinas de interrupción

A continuación se muestran programas de ejemplo para una interrupción temporizada que se puede utilizar en aplicaciones tales como leer el valor de una entrada analógica. En este ejemplo, el intervalo de muestreo de la entrada analógica es de 100 ms.

Las figuras 4-7 a 4-11 muestran programas que utilizan una subrutina y una rutina de interrupción en los diversos lenguajes de programación S7-200.

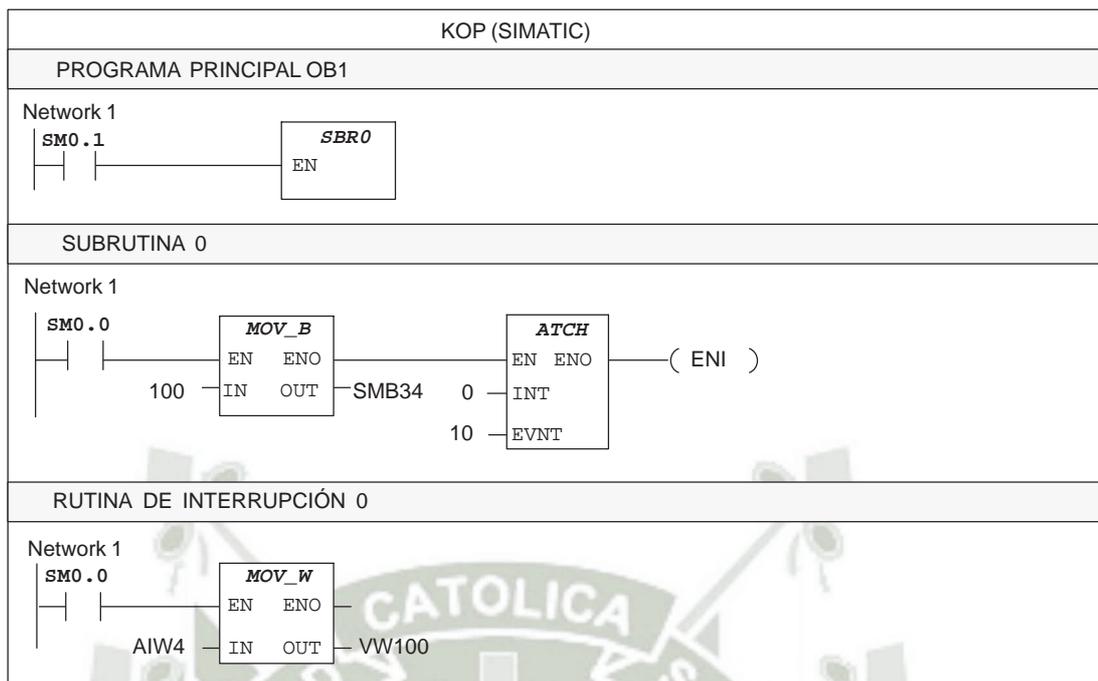


Figura 4-7 Programa KOP (SIMATIC) con una subrutina y una rutina de interrupción

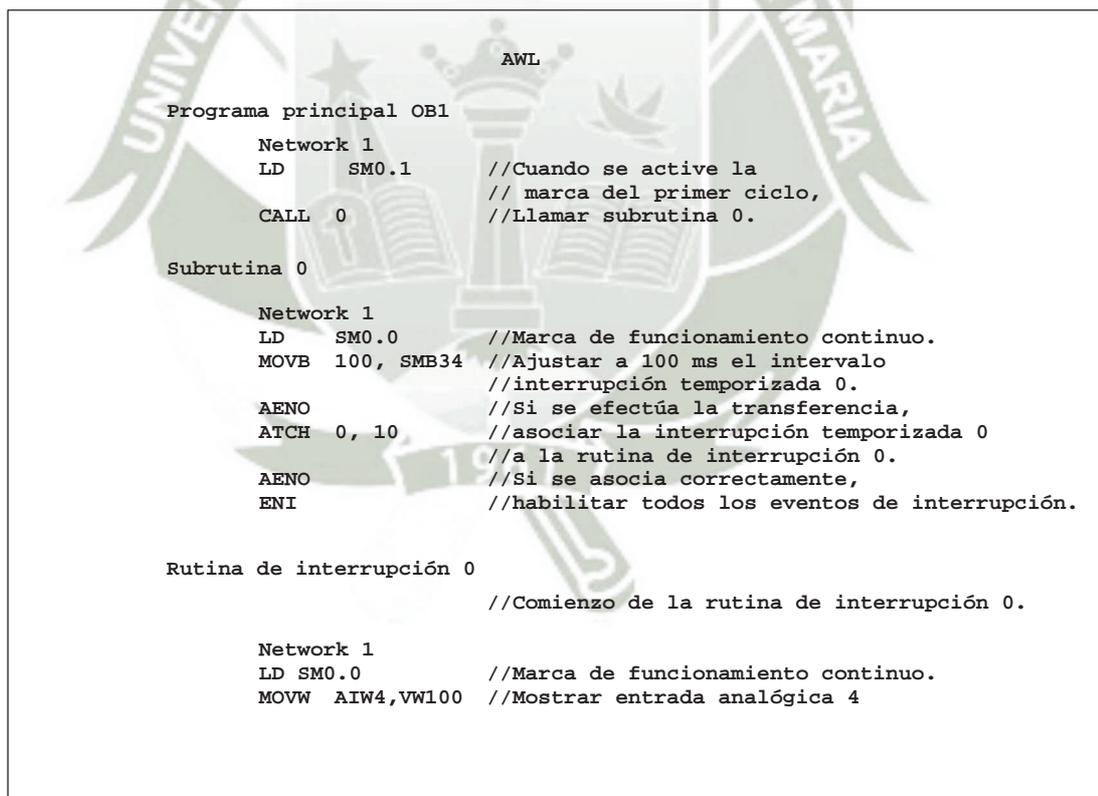


Figura 4-8 Programa AWL (SIMATIC) con una subrutina y una rutina de interrupción

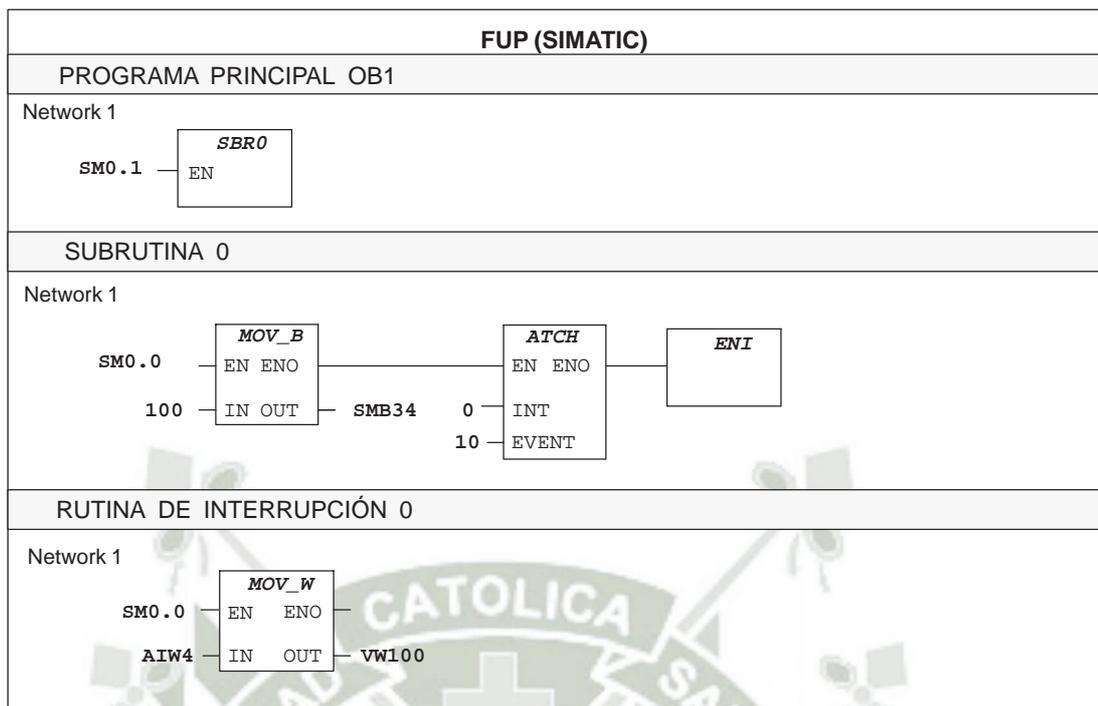


Figura 4-9 Programa FUP (SIMATIC) con una subrutina y una rutina de interrupción

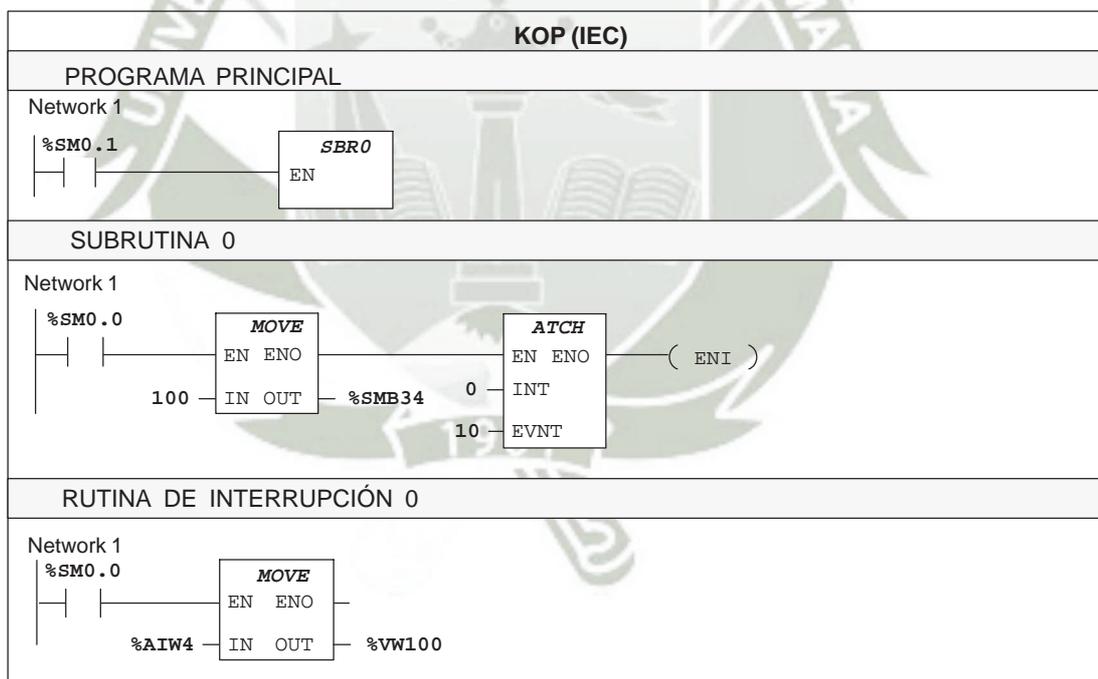


Figura 4-10 Programa LD (IEC) con una subrutina y una rutina de interrupción

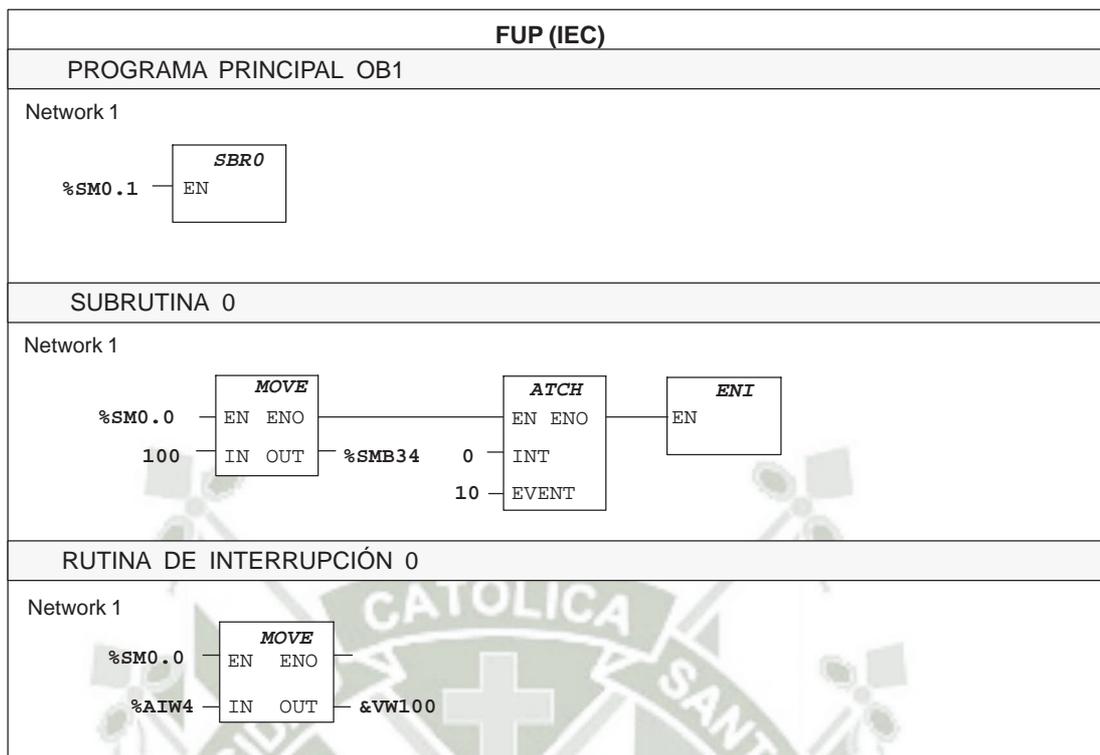


Figura 4-11 Programa FBD (IEC) con una subrutina y una rutina de interrupción

## 4.6 El ciclo de la CPU

La CPU S7-200 se ha previsto para que ejecute cíclicamente una serie de tareas, incluyendo el programa de usuario. Dicha ejecución se denomina ciclo. Durante el ciclo que se muestra en la figura 4-12, la CPU ejecuta la mayoría de las tareas siguientes (o todas ellas):

- Lee las entradas.
- Ejecuta el programa de usuario.
- Procesa las peticiones de comunicación.
- Efectúa un autodiagnóstico.
- Escribe en las salidas.



Figura 4-12 Ciclo de la CPU S7-200

La serie de tareas que se ejecutan durante el ciclo depende del modo de operación de la CPU. La CPU S7-200 tiene dos modos de operación: STOP y RUN. Con respecto al ciclo, la principal diferencia entre STOP y RUN es que el programa se ejecuta al estar la CPU en modo RUN, mas no en STOP.

### Leer las entradas digitales

Al principio de cada ciclo se leen los valores actuales de las entradas digitales y se escriben luego en la imagen del proceso de las entradas.

La CPU reserva un espacio de la imagen del proceso de las entradas en incrementos de ocho bits (un byte). Si la CPU o el módulo de ampliación no proveen una entrada física para cada bit del byte reservado, no será posible asignar dichos bits a los módulos siguientes en la cadena de E/S o utilizarlos en el programa de usuario. Al comienzo de cada ciclo, la CPU pone a 0 estos bits no utilizados en la imagen del proceso. No obstante, si la CPU asiste varios módulos de ampliación y no se está utilizando su capacidad de E/S (porque no se han instalado los módulos de ampliación), los bits de entradas de ampliación no utilizados se pueden usar como marcas internas adicionales.

La CPU no actualiza las entradas analógicas como parte del ciclo normal, a menos que se haya habilitado la filtración digital de las mismas. Dicha filtración es una opción seleccionable por el usuario, pudiéndose habilitar individualmente para cada una de las entradas analógicas.

La filtración digital se ha previsto para su utilización en módulos analógicos de bajo costo que no disponen de una filtración interna al módulo. Es recomendable utilizar la filtración digital en aplicaciones donde la señal de entrada cambia lentamente. Si la señal es rápida, no es recomendable habilitar la filtración digital.

Si se habilita la filtración de una entrada analógica, la CPU actualiza dicha entrada una vez por ciclo, efectúa la filtración y almacena internamente el valor filtrado. El valor filtrado se suministra entonces cada vez que el programa accede a la entrada analógica.

Si no se habilita la filtración de una entrada analógica, la CPU lee su valor del módulo físico cada vez que el programa de usuario accede a la entrada analógica.

### **Ejecutar el programa**

Durante esta fase del ciclo, la CPU ejecuta el programa desde la primera operación hasta la última (= Finalizar programa). El control directo de las entradas y salidas permite acceder directamente a las mismas mientras se ejecuta el programa o una rutina de interrupción.

Si se utilizan interrupciones, las rutinas asociadas a los eventos de interrupción se almacenan como parte del programa (v. apt. 4.5). Las rutinas de interrupción no se ejecutan como parte del ciclo, sino sólo cuando ocurre el evento (en cualquier punto del ciclo).

### **Procesar las peticiones de comunicación**

Durante esta fase del ciclo, la CPU procesa los mensajes que haya recibido por el puerto de comunicación.

### **Efectuar el autodiagnóstico de la CPU**

Durante el autodiagnóstico se comprueba el firmware de la CPU y la memoria del programa (sólo en modo RUN), así como el estado de los módulos de ampliación.

### **Escribir en las salidas digitales**

Al final de cada ciclo, la CPU escribe los valores de la imagen del proceso de las salidas en las salidas digitales.

La CPU reserva un espacio de la imagen del proceso de las salidas en incrementos de ocho bits (un byte). Si la CPU o el módulo de ampliación no proveen una salida física para cada bit del byte reservado, no será posible asignar dichos bits a los módulos siguientes en la cadena de E/S.

Cuando el modo de operación de la CPU se cambia de RUN a STOP, las salidas digitales adoptan los valores definidos en la tabla de salidas o conservan su estado actual (v. apt. 6.4). Las salidas analógicas conservan su último valor. Por defecto, las salidas digitales están desactivadas.

### Interrumpir el ciclo

Si se utilizan interrupciones, las rutinas asociadas a los eventos de interrupción se almacenan como parte del programa. Las rutinas de interrupción no se ejecutan como parte del ciclo, sino sólo cuando ocurre el evento (en cualquier punto del ciclo). La CPU procesa las interrupciones según su prioridad y después en el orden que aparecen.

### Imagen del proceso de las entradas y salidas

Por lo general, es recomendable utilizar la imagen del proceso, en vez de acceder directamente a las entradas o salidas durante la ejecución del programa. Las imágenes del proceso existen por tres razones:

- El sistema verifica todas las entradas al comenzar el ciclo. De este modo se sincronizan y "congelan" los valores de estas entradas durante la ejecución del programa. La imagen del proceso actualiza las salidas cuando termina de ejecutarse el programa. Ello tiene un efecto estabilizador en el sistema.
- El programa de usuario puede acceder a la imagen del proceso mucho más rápido de lo que podría acceder directamente a las entradas y salidas físicas, con lo cual se acelera su tiempo de ejecución.
- Las entradas y salidas son unidades de bit a las que se debe acceder en formato de bit. No obstante, la imagen del proceso permite acceder a ellas en formato de bits, bytes, palabras y palabras dobles, lo que ofrece flexibilidad adicional.

### Control directo de las entradas y salidas

Las operaciones de control directo de las entradas y salidas (E/S) permiten acceder a la entrada o salida física, aunque el acceso a las E/S se efectúa por lo general a través de las imágenes del proceso. El acceso directo a una entrada no modifica la dirección correspondiente en la imagen del proceso de las entradas. En cambio, el acceso directo a una salida actualiza simultáneamente la dirección correspondiente en la imagen del proceso de las salidas.

La CPU gestiona las E/S analógicas como datos directos, a menos que se haya habilitado la filtración digital de las entradas analógicas (v. apt. 6.5). Cuando se escribe un valor en una salida analógica, la salida se actualiza inmediatamente.

#### 4.7 Ajustar el modo de operación de la CPU

La CPU S7-200 tiene dos modos de operación:

- STOP: La CPU no ejecuta el programa. Cuando está en modo STOP, es posible cargar programas o configurar la CPU.
- RUN: La CPU ejecuta el programa.

El diodo luminoso (LED) en la parte frontal de la CPU indica el modo de operación actual.

El modo de operación se puede cambiar como se indica a continuación:

- Accionando manualmente el selector de modos de operación de la CPU.
- Utilizando el software de programación STEP 7-Micro/WIN 32 y colocando el selector de la CPU en posición TERM o RUN.
- Insertando una operación STOP en el programa.

#### Cambiar el modo de operación con el selector

El modo de operación de la CPU se puede cambiar manualmente accionando el selector (ubicado debajo de la tapa de acceso frontal de la CPU):

- Si el selector se pone en STOP, se detendrá la ejecución del programa.
- Si el selector se pone en RUN, se iniciará la ejecución del programa.
- Si el selector se pone en TERM (terminal), no cambiará el modo de operación de la CPU. No obstante, será posible cambiarlo utilizando el software de programación (STEP 7-Micro/WIN 32).

Si se interrumpe la alimentación estando el selector en posición STOP o TERM, la CPU pasará a modo STOP cuando se le aplique tensión. Si se interrumpe la alimentación estando el selector en posición RUN, la CPU pasará a modo RUN cuando se le aplique tensión.

### Cambiar el modo de operación con STEP 7-Micro/WIN 32

Como muestra la figura 4-13, el modo de operación de la CPU se puede cambiar también con STEP 7-Micro/WIN 32. Para que ello sea posible, el selector de la CPU deberá estar en posición TERM o RUN.

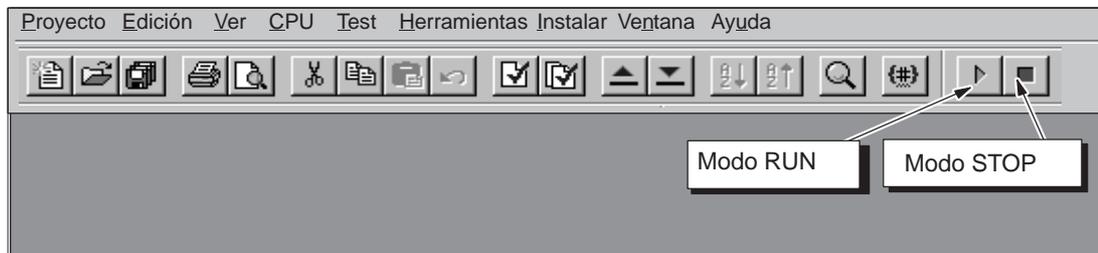


Figura 4-13 Utilizar STEP 7-Micro/WIN 32 para cambiar el modo de operación de la CPU

### Cambiar el modo de operación desde el programa

Para cambiar la CPU a modo STOP es posible introducir la correspondiente operación (STOP) en el programa. Ello permite detener la ejecución del programa en función de la lógica. Para obtener más información acerca de la operación STOP, consulte el capítulo 9 (operaciones SIMATIC) o el capítulo 10 (operaciones IEC 1131-3).



## 4.8 Definir una contraseña para la CPU

Todas las CPUs S7-200 ofrecen una protección con contraseña para restringir el acceso a determinadas funciones. Con una contraseña se puede acceder a las funciones y a la memoria de la CPU. Si no se utiliza la opción de contraseña, la CPU permite un acceso ilimitado. Si está protegida con una contraseña, la CPU prohíbe todas las operaciones restringidas conforme a la configuración definida al definir la contraseña.

### Restringir el acceso a la CPU

Como muestra la tabla 4-12, las CPUs S7-200 ofrecen tres niveles de protección para acceder a sus funciones. Cada uno de dichos niveles permite ejecutar determinadas funciones sin la contraseña. Si se introduce la contraseña correcta, es posible acceder a todas las funciones de la CPU. El ajuste estándar para la CPU S7-200 es el nivel 1 (privilegios totales).

Si se introduce la contraseña a través de una red, no se afecta la protección con contraseña de la CPU. Si un usuario tiene acceso a las funciones restringidas de la CPU, ello no autoriza a los demás usuarios a acceder a dichas funciones. El acceso ilimitado a las funciones de la CPU sólo se permite a un usuario a la vez.

#### Nota

Una vez introducida la contraseña, el nivel de protección se conservará aproximadamente durante un minuto después de haber desconectado la unidad de programación de la CPU. Si otro usuario se conecta inmediatamente a la CPU durante ese tiempo es posible que pueda acceder a la unidad de programación.

Tabla 4-12 Restringir el acceso a la CPU S7-200

Tarea	Nivel 1	Nivel 2	Nivel 3
Leer y escribir datos de usuario	No restringido	No restringido	No restringido
Arrancar, detener y rearrancar la CPU			
Leer y escribir el reloj de tiempo real			
Cargar en la PG el programa de usuario, los datos y la configuración		Restringido	Restringido
Cargar en la CPU			
Borrar el programa de usuario, los datos y la configuración			
Forzar datos o ejecutar uno/varios ciclo(s)			
Copiar en el cartucho de memoria			
Escribir en las salidas en modo STOP			

## Configurar la contraseña para la CPU

STEP 7-Micro/WIN 32 permite definir una contraseña para acceder a las funciones de la CPU. Elija el comando de menú **Ver > Bloque de sistema** y haga clic en la ficha "Contraseña" (v. fig. 4-14). Indique el nivel de protección deseado. Introduzca y verifique luego la contraseña.

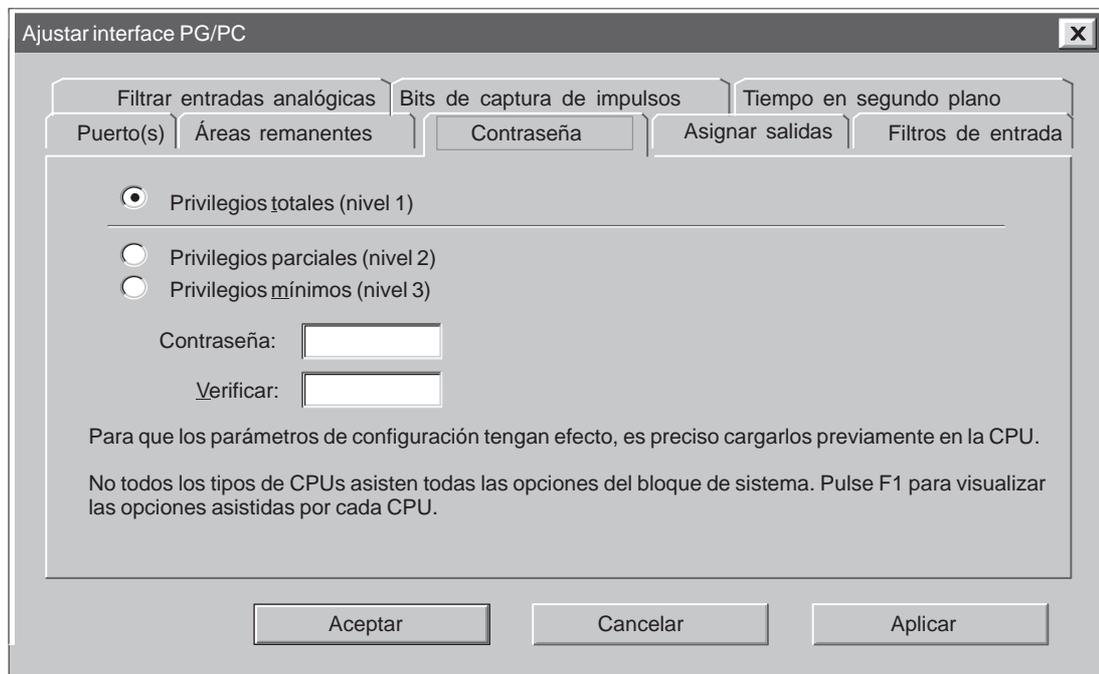


Figura 4-14 Configurar una contraseña para la CPU

## Remedio si se olvida la contraseña de la CPU

Si se olvida la contraseña, es preciso efectuar un borrado total de la memoria de la CPU y volver a cargar el programa. Al borrar la memoria de la CPU, ésta pasa a modo STOP y recupera los ajustes estándar, con excepción de la dirección de estación, la velocidad de transferencia y el reloj de tiempo real.

Si desea borrar el programa de la CPU, elija el comando de menú **CPU > Borrar...** con objeto de visualizar el cuadro de diálogo "Borrar CPU". Seleccione los tres bloques y confirme haciendo clic en el botón "Aceptar". Si ha configurado una contraseña, aparecerá el cuadro de diálogo "Contraseña". Introduciendo la contraseña "clearplc" podrá iniciar el borrado total.

La función de borrado total no borra el programa contenido en el cartucho de memoria. Puesto que en éste último se encuentra almacenado no sólo el programa, sino también la contraseña, es preciso volver a programar también dicho cartucho para borrar la contraseña olvidada.



---

### Precaución

Al efectuarse un borrado total de la CPU, se desactivan las salidas (si son salidas analógicas, éstas se congelan en un valor determinado).

Si la CPU S7-200 está conectada a otros equipos durante el borrado total, es posible que los cambios de las salidas se transfieran también a dichos equipos. Si ha determinado que el "estado seguro" de las salidas sea diferente al ajustado de fábrica, es posible que los cambios de las salidas provoquen reacciones inesperadas en los equipos, lo que podría causar la muerte o heridas graves personales y/o daños materiales.

Adopte siempre las medidas de seguridad apropiadas y asegúrese de que su proceso se encuentra en un estado seguro antes de efectuar un borrado total de la CPU.

---

## 4.9 Comprobar y observar el programa

STEP 7-Micro/WIN 32 ofrece diversas posibilidades para comprobar y observar el programa de usuario.

### Ejecutar uno o varios ciclos para observar el programa

Es posible indicar que la CPU ejecute el programa durante un número limitado de ciclos (entre 1 y 65.535 ciclos). Seleccionando el número de ciclos que la CPU debe ejecutar, se puede observar el programa a medida que van cambiando las variables del proceso. Elija el comando de menú **Test > Varios ciclos** para especificar el número de ciclos a ejecutar. La figura 4-15 muestra el cuadro de diálogo para introducir el número de ciclos a ejecutar por la CPU.

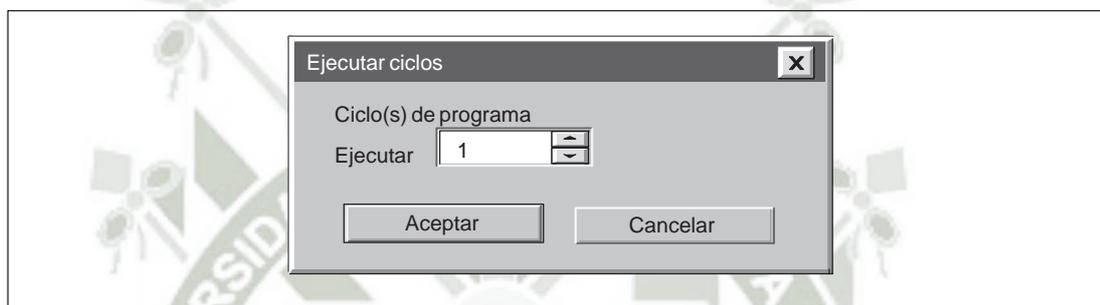


Figura 4-15 Ejecutar el programa un número determinado de ciclos

### Utilizar una tabla de estado para observar y modificar el programa

Como muestra la figura 4-16, la tabla de estado se puede utilizar para leer, escribir, forzar y observar las variables mientras se ejecuta el programa. Elija el comando de menú **Ver > Tabla de estado**.

- Los botones de la barra de herramientas de la tabla de estado se visualizan en el área de la barra de herramientas de STEP 7-Micro/WIN 32. Dichos botones (Orden ascendente, Orden descendente, Lectura sencilla, Escribir todo, Forzar, Desforzar, Desforzar todo y Leer todo) se pueden activar al seleccionar la tabla de estado.
- Se pueden crear varias tablas de estado.
- Para seleccionar el formato de una celda, seleccione la celda y pulse el botón derecho del ratón a fin de abrir la lista desplegable (v. fig. 4-16).

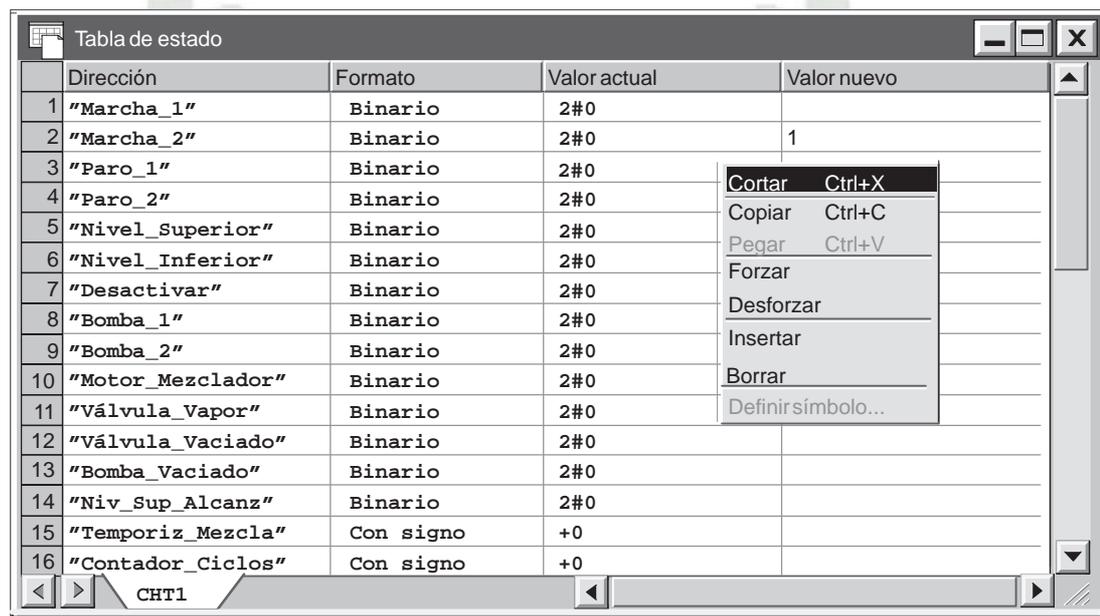


Figura 4-16 Observar y modificar variables con una tabla de estado

### Visualizar el estado del programa en KOP

El estado del programa KOP se puede ver en STEP 7-Micro/WIN 32. STEP 7-Micro/WIN 32 debe estar visualizando el programa KOP. El estado KOP muestra el estado de todos los valores de los operandos de las operaciones (v. fig. 4-17). Todas las informaciones de estado se basan en los valores leídos al final de un ciclo de la CPU. STEP 7-Micro/WIN 32 adquiere los valores para visualizar el estado durante varios ciclos de la CPU, actualizando luego la ventana de estado KOP. Por consiguiente, el estado KOP visualizado no refleja el estado real de ejecución de cada elemento KOP.

Para abrir la ventana del estado KOP, seleccione el icono de estado en la barra de herramientas (v. fig. 4-17).

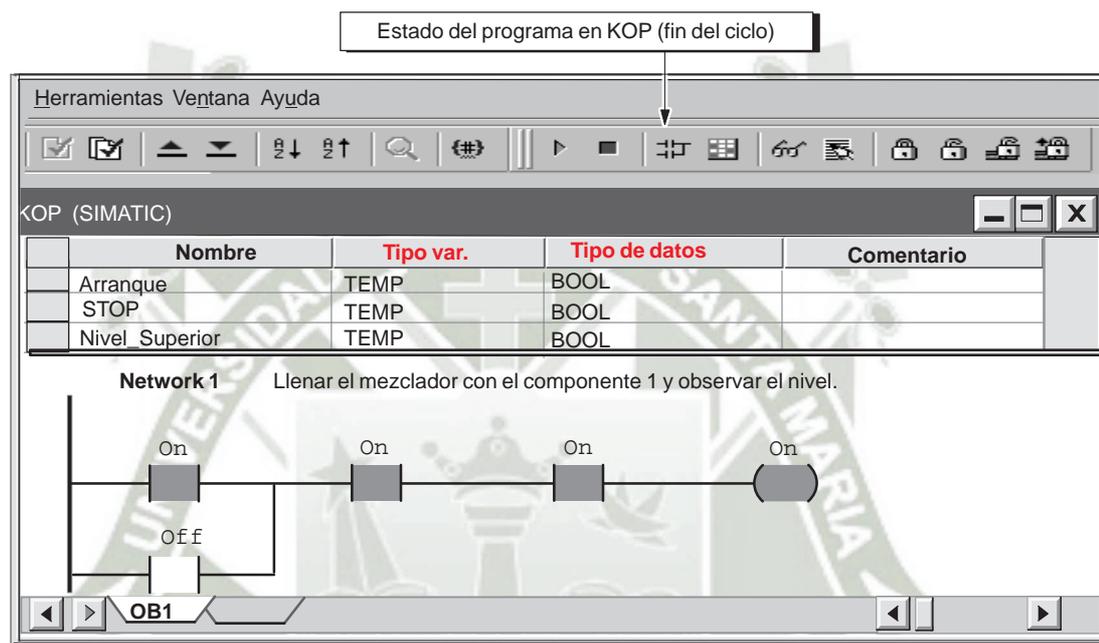


Figura 4-17 Visualizar el estado de un programa en KOP

### Visualizar el estado del programa en FUP

El estado del programa FUP se puede ver en STEP 7-Micro/WIN 32. STEP 7-Micro/WIN 32 debe estar visualizando el programa FUP. El estado FUP muestra el estado de todos los valores de los operandos de las operaciones. Todas las informaciones de estado se basan en los valores leídos al final de un ciclo de la CPU. STEP 7-Micro/WIN 32 adquiere los valores para visualizar el estado durante varios ciclos de la CPU, actualizando luego la ventana de estado FUP. Por consiguiente, el estado FUP visualizado no refleja el estado real de ejecución de cada elemento FUP.

Para abrir la ventana de estado FUP, seleccione el icono de estado en la barra de herramientas (v. fig. 4-18).

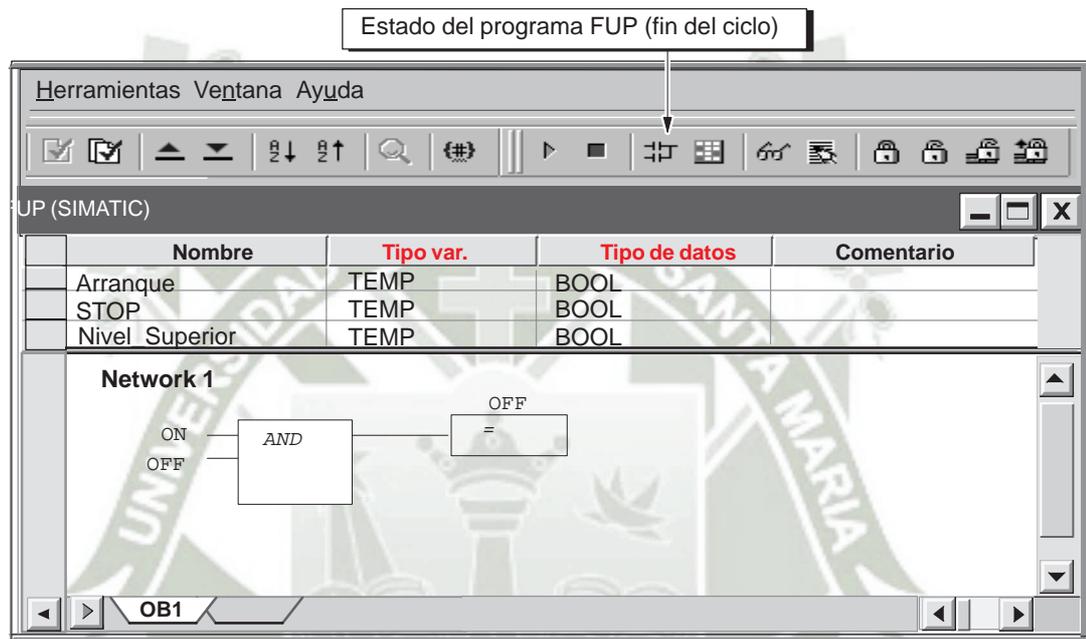


Figura 4-18 Visualizar el estado de un programa en FUP

## Forzar valores específicos

La CPU S7-200 permite forzar algunas o todas las entradas y salidas (I y Q). Además es posible forzar hasta 16 marcas internas (V o M) o bien los valores de las entradas y salidas analógicas (AI o AQ). Los valores de la memoria V o de las marcas se pueden forzar en formato de bytes, palabras o palabras dobles. Los valores analógicos se fuerzan únicamente como palabras, en bytes de número par (p.ej. AIW6 ó AQW14). Todos los valores forzados se almacenan en la memoria EEPROM no volátil de la CPU.

Puesto que los valores forzados se pueden modificar durante el ciclo (por el programa, al actualizarse las entradas y salidas o al procesarse las comunicaciones), la CPU los vuelve a forzar en diversos puntos del ciclo. La figura 4-19 muestra el ciclo, indicando dónde la CPU actualiza las variables forzadas.

La función Forzar se impone a las operaciones de lectura y de escritura directas. Asimismo, se impone a una salida que se haya configurado para que adopte un valor determinado cuando la CPU cambie a STOP. En este último caso, la salida conservará el valor forzado y no el valor configurado.

Como muestra la figura 4-20, la tabla de estado se puede utilizar para forzar valores. Para forzar un nuevo valor, introduzca el valor en la columna "Nuevo valor" de la tabla de estado y haga clic en el botón "Forzar" en la barra de herramientas. Para forzar un valor existente, destaque el valor en la columna "Valor actual" y pulse luego el botón "Forzar".



Figura 4-19 Ciclo de la CPU S7-200

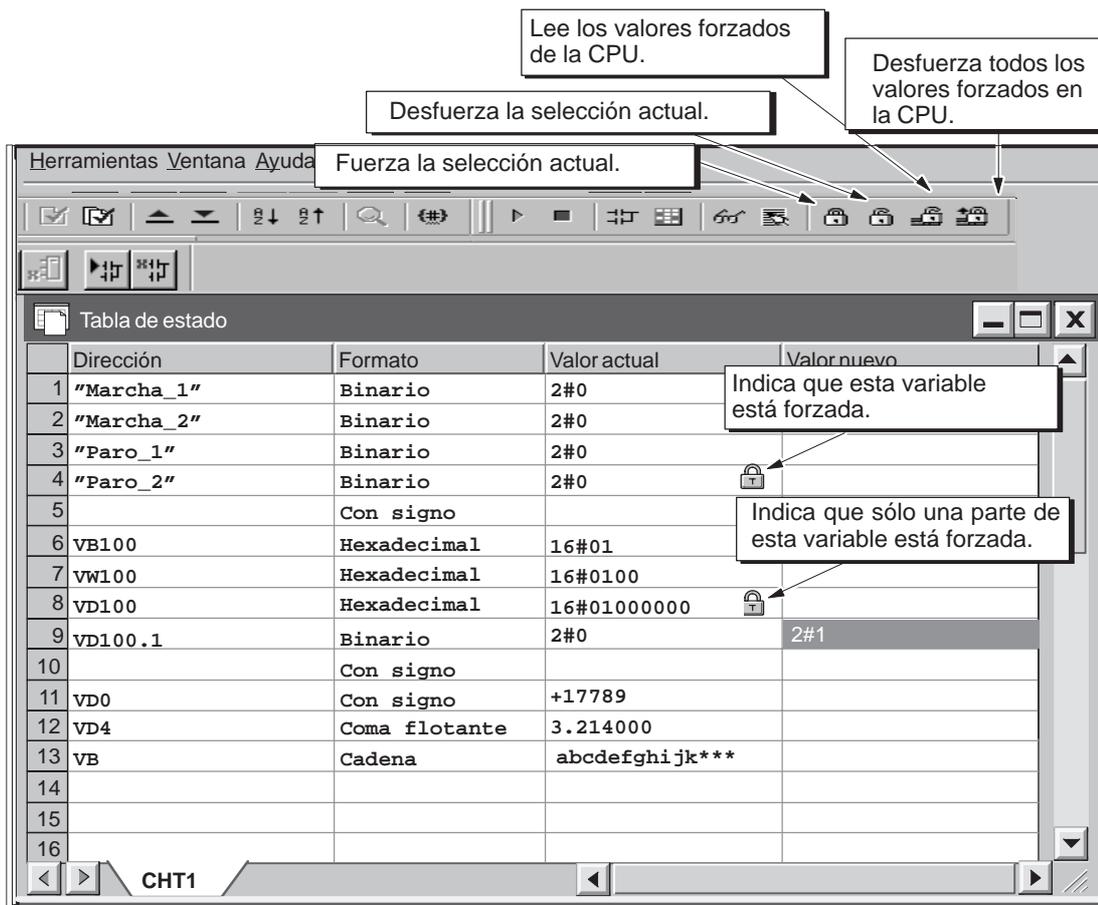


Figura 4-20 Forzar variables mediante la tabla de estado

#### 4.10 Eliminar errores de las CPUs S7-200

Las CPUs S7-200 clasifican los errores en errores fatales y no fatales.

STEP 7-Micro/WIN 32 permite visualizar los códigos generados por los errores. Elija el comando de menú **CPU > Información** para visualizar dichos errores. La figura 4-21 muestra un cuadro de diálogo donde se visualizan el código y la descripción del error. El Anexo B incluye una lista completa de los códigos de error.

En la figura 4-21, el campo "Último fatal" muestra el último código de error fatal generado por la CPU. Al desconectarse la alimentación, este valor se conserva si se respalda la RAM. El valor se pone a 0 si se efectúa un borrado total de la CPU o si la RAM no se respalda tras un corte prolongado de la alimentación.

El campo "Total fatales" muestra el contaje de los errores fatales generados por la CPU desde la última vez que se efectuó un borrado total de la misma. Al desconectarse la alimentación, este valor se conserva si se respalda la RAM. Este valor se pone a 0 si se efectúa un borrado total de la CPU o si la RAM no se respalda tras un corte prolongado de la alimentación.

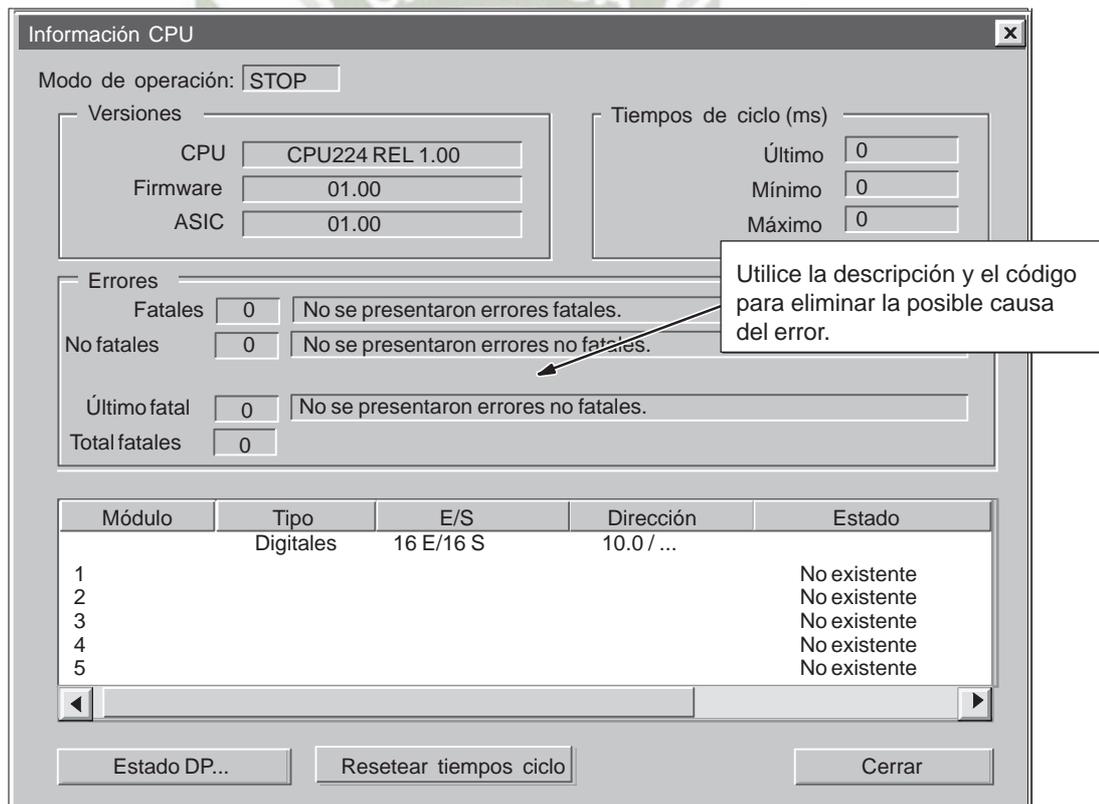


Figura 4-21 Cuadro de diálogo "Información CPU": ficha "Estado de error"

## Eliminar errores fatales

Cuando ocurre un error fatal, la CPU detiene la ejecución del programa. Según la gravedad del error, es posible que la CPU no pueda ejecutar todas las funciones, o incluso ninguna de ellas. El objetivo del tratamiento de errores fatales es conducir a la CPU a un estado seguro, en el que se puedan analizar y eliminar las condiciones que hayan causado el error. Cuando la CPU detecta un error fatal, cambia a modo STOP, enciende los indicadores "SF" y "STOP" y desactiva las salidas. La CPU permanece en dicho estado hasta que haya eliminado la causa del error fatal.

Una vez efectuados los cambios para eliminar el error fatal, es preciso rearrancar la CPU. La CPU se puede rearrancar utilizando uno de los métodos siguientes:

- Desconectando la alimentación y conectándola luego nuevamente.
- Cambiando el selector de modos de RUN o TERM a STOP.
- Utilizando STEP 7-Micro/WIN. STEP 7-Micro/WIN 32 incorpora el comando de menú **CPU > Reset arranque** que obliga a la CPU a efectuar un re arranque y a borrar todos los errores fatales.

Al re arrancar la CPU se borra la condición de error fatal y se ejecuta un diagnóstico de arranque para verificar si se ha corregido el error. En caso de detectarse otro error fatal, se encenderá de nuevo el indicador "SF". De lo contrario, la CPU comenzará a funcionar con normalidad.

Existen diversas condiciones posibles de error que incapacitan a la CPU para la comunicación. En esos casos no es posible visualizar el código de error de la CPU. Dichos errores indican fallos de hardware, por lo que es necesario reparar la CPU. No se pueden solucionar modificando el programa ni borrando la memoria de la CPU.

## Eliminar errores no fatales

Los errores no fatales pueden mermar parcialmente el funcionamiento de la CPU, pero no le impiden ejecutar el programa o actualizar las entradas y salidas. Como muestra la figura 4-21, STEP 7-Micro/WIN 32 permite visualizar los códigos generados por los errores no fatales. Hay tres categorías básicas de errores no fatales:

- Errores durante el tiempo de ejecución. Todos los errores no fatales que se detectan en modo RUN se depositan en marcas especiales (SM). El programa puede observar y evaluar dichas marcas. Consulte el Anexo C para obtener más información acerca de las marcas especiales utilizadas para indicar los errores no fatales durante el tiempo de ejecución.

Cuando se enciende la CPU, ésta lee la configuración de las entradas y salidas, almacenando dicha información en la memoria de datos del sistema y en las marcas especiales. Durante el funcionamiento normal de la CPU, el estado de las entradas y salidas se actualiza periódicamente y se almacena en las marcas especiales. Si la CPU detecta una configuración de E/S diferente, activa el correspondiente bit del byte de error en el módulo. El módulo de ampliación no se actualizará hasta que dicho bit se desactive de nuevo. Para que la CPU pueda desactivar ese bit, las entradas y salidas del módulo deberán coincidir nuevamente con la configuración almacenada en la memoria de datos del sistema.

- Errores de compilación del programa. Al cargar un programa en la CPU, ésta lo compila. Si durante la compilación se detecta una violación de las reglas, el proceso de carga se suspenderá, generándose entonces un código de error. (Si ya se ha cargado un programa en la CPU, seguirá existiendo en la EEPROM, por lo que no se perderá). Una vez corregido el programa, se podrá cargar de nuevo.
- Errores de programación durante el tiempo de ejecución. El programa puede crear condiciones de error mientras se ejecuta el programa. Por ejemplo, un puntero de direccionamiento indirecto que era válido cuando se compiló el programa puede haber cambiado durante la ejecución del programa, señalando entonces a una dirección fuera de área. Esto se considera un error de programación durante el tiempo de ejecución. Utilice el cuadro de diálogo que muestra la figura 4-21 en la página 4-36 para determinar el tipo de error que ha ocurrido.

La CPU no cambia a modo STOP cuando detecta un error no fatal. Tan sólo deposita el evento en la marca especial en cuestión y continúa ejecutando el programa. No obstante, es posible programar que la CPU cambie a modo STOP cuando se detecte un error no fatal. La figura 4-22 muestra un segmento de un programa que controla una marca especial. La operación prevé que la CPU cambie a modo STOP si se detecta un error de E/S.

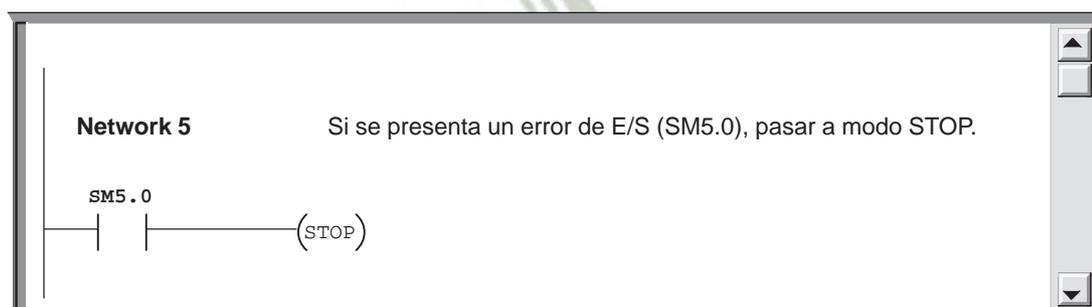


Figura 4-22 Detectar errores no fatales mediante el programa de usuario

## Memoria de la CPU: tipos de datos y direccionamiento

# 5

La CPU S7-200 dispone de áreas de memoria especiales para que los datos se puedan procesar de forma más rápida y eficiente.

### Índice del capítulo

Apartado	Descripción	Página
5.1	Direccionamiento directo de las áreas de memoria de la CPU	5-2
5.2	Direccionamiento indirecto (SIMATIC) de las áreas de memoria de la CPU	5-13
5.3	Respaldar datos en la CPU S7-200	5-15
5.4	Utilizar el programa para guardar datos en la memoria no volátil	5-20
5.5	Guardar el programa en el cartucho de memoria	5-22

## 5.1 Direccionamiento directo de las áreas de memoria de la CPU

La CPU S7-200 almacena información en diferentes áreas de la memoria que tienen direcciones unívocas. Es posible indicar explícitamente la dirección a la que se desea acceder. El programa puede acceder entonces directamente a la información.

### Acceder a los datos a través de direcciones

Para acceder a un bit en un área de memoria es preciso indicar la dirección del mismo, la cual está formada por un identificador de área, la dirección del byte y el número del bit. La figura 5-1 muestra un ejemplo de direccionamiento de un bit (denominado también direccionamiento "byte.bit"). En el ejemplo, el área de memoria y la dirección del byte (I=entrada y 3=byte 3) van seguidas de un punto decimal (".") que separa la dirección del bit (bit 4).

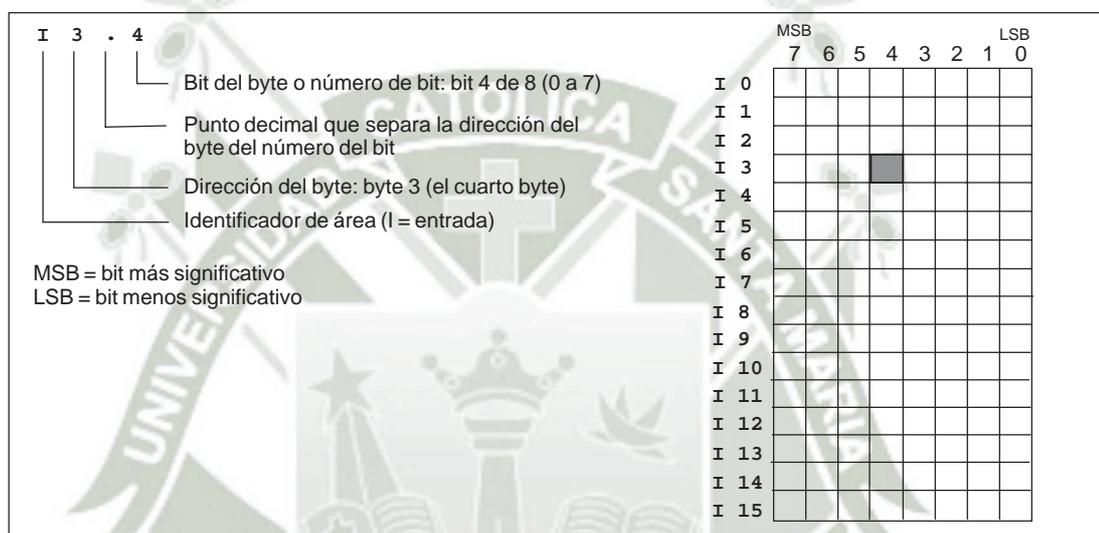


Figura 5-1 Acceder a un bit de datos en la memoria de la CPU (direccionamiento "byte.bit")

Utilizando el formato de dirección de byte se puede acceder a los datos de numerosas áreas de la memoria de la CPU (V, I, Q, M, S y SM) en formato de bytes, palabras o palabras dobles. La dirección de un byte, de una palabra o de una palabra doble de datos en la memoria de la CPU se indica de forma similar a la dirección de un bit. Esta última está formada por un identificador de área, el tamaño de los datos y la dirección inicial del valor del byte, de la palabra o de la palabra doble, como muestra la figura 5-2. Para acceder a los datos comprendidos en otras áreas de la memoria de la CPU (p.ej. T, C, HC y acumuladores) es preciso utilizar una dirección compuesta por un identificador de área y un número de elemento.

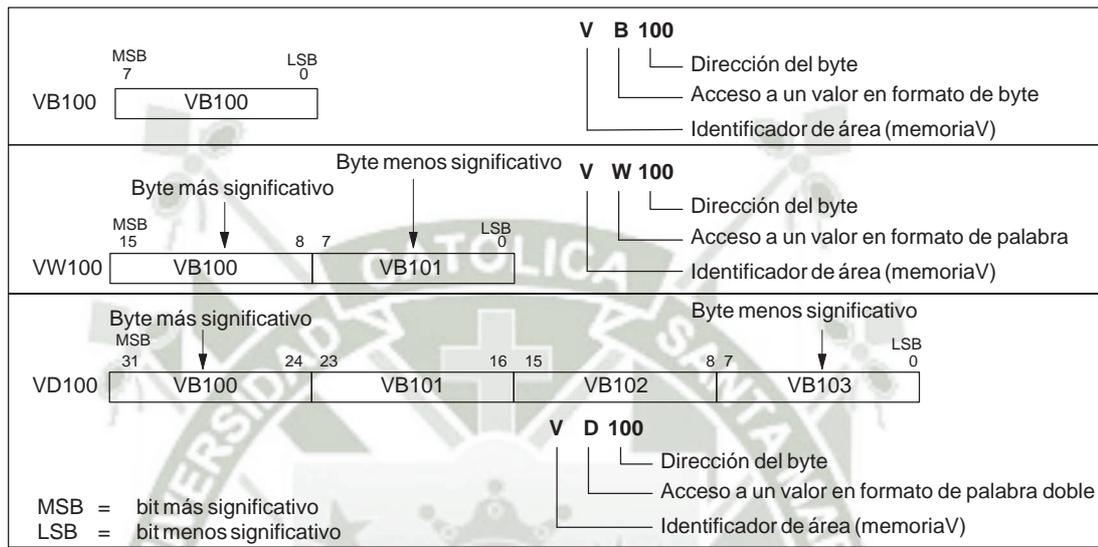


Figura 5-2 Acceso a la misma dirección en formato de byte, palabra y palabra doble

## Representación numérica

La tabla 5-1 muestra el margen de números enteros representables en diversos tamaños de datos.

Los números reales (en coma flotante) se representan como números de precisión simple de 32 bits, siendo su formato: +1,175495E-38 a +3,402823E+38 (positivo), y -1,175495E-38 a -3,402823E+38 (negativo). A los valores de números reales se accede en formato de palabra doble. Para obtener más información sobre los números reales (o en coma flotante), consulte la norma ANSI/IEEE 754-1985.

Tabla 5-1 Indicadores de tamaño (y sus respectivos márgenes de números enteros)

Tamaño de los datos	Margen de enteros sin signo		Margen de enteros con signo	
	Decimal	Hexadecimal	Decimal	Hexadecimal
B (byte): valor de 8 bits	0 a 255	0 a FF	-128 a 127	80 a 7F
W (palabra): valor de 16 bits	0 a 65.535	0 a FFFF	-32.768 a 32.767	8000 a 7FFF
D (palabra doble): valor de 32 bits	0 a 4.294.967.295)	0 a FFFF FFFF	-2.147.483.648 a 2.147.483.647	8000 0000 a 7FFF FFFF

## Direccionamiento de la imagen del proceso de las entradas (I)

Como se describe en el apartado 4.6, la CPU lee las entradas físicas al comienzo de cada ciclo y escribe los correspondientes valores en la imagen del proceso de las entradas. A ésta última se puede acceder en formato de bit, byte, palabra o palabra doble.

Formato:

Bit *I[direcc. del byte].[direcc. del bit] I0.1*  
 Byte, palabra, palabra doble *I[tamaño][direcc. del byte inicial] IB4*

## Direccionamiento de la imagen del proceso de las salidas (Q)

Al final de cada ciclo, la CPU copia en las salidas físicas el valor almacenado en la imagen del proceso de las salidas. A ésta última se puede acceder en formato de bit, byte, palabra o palabra doble.

Formato:

Bit *Q[direcc. del byte].[direcc. del bit] Q1.1*  
 Byte, palabra, palabra doble *Q[tamaño][direcc. del byte inicial] QB5*

### Direccionamiento de la memoria de variables (V)

La memoria de variables (memoria V) se puede utilizar para depositar los resultados intermedios calculados por las operaciones en el programa. La memoria V también permite almacenar otros datos que pertenezcan al proceso o a la tarea actuales. A la memoria de variables se puede acceder en formato de bit, byte, palabra o palabra doble.

Formato:

Bit *V[direcc. del byte].[direcc. del bit]* V10.2  
Byte, palabra, palabra doble *V[tamaño][direcc. del byte inicial]* VW100

### Direccionamiento del área de marcas (M)

El área de marcas (memoria M) se puede utilizar en calidad de relés de control para almacenar el estado inmediato de una operación u otra información de control. Al área de marcas se puede acceder en formato de bit, byte, palabra o palabra doble.

Formato:

Bit *M[direcc. del byte].[direcc. del bit]* M26.7  
Byte, palabra, palabra doble *M[tamaño][direcc. del byte inicial]* MD20

### Direccionamiento de los relés de control secuencial (S)

Los relés de control secuencial (S) permiten organizar los pasos del funcionamiento de una máquina en segmentos equivalentes en el programa. Los SCRs permiten segmentar lógicamente el programa de usuario. A los relés de control secuencial (SCR) se puede acceder en formato de bit, byte, palabra o palabra doble.

Formato:

Bit *S[direcc. del byte].[direcc. del bit]* S3.1  
Byte, palabra, palabra doble *S[tamaño][direcc. del byte inicial]* SB4

### Direccionamiento de las marcas especiales (SM)

Las marcas especiales permiten intercambiar datos entre la CPU y el programa. Dichas marcas se puede utilizar para seleccionar y controlar algunas funciones especiales de la CPU S7-200, tales como:

- Una marca que se activa sólo en el primer ciclo.
- Marcas que se activan y se desactivan en determinados intervalos.
- Marcas que muestran el estado de operaciones matemáticas y de otras operaciones.

Para obtener más información acerca de las marcas especiales, consulte el Anexo C. Aunque el área de las marcas especiales se basa en bits, es posible acceder a los datos en formato de bit, byte, palabra o palabra doble.

Formato:

Bit *SM[direcc. del byte].[direcc. del bit]* SM0.1  
Byte, palabra, palabra doble *SM[tamaño][direcc. del byte inicial]* SMB86

## Direccionamiento del área de memoria local (L)

Las CPUs S7-200 disponen de 64 bytes de memoria local (L), de los cuales 60 se pueden utilizar como memoria "borrador" para transferir parámetros formales a las subrutinas. Si se programa en KOP o FUP, STEP 7-Micro/WIN 32 reserva los últimos cuatro bytes de la memoria local para su propio uso. Si se programa en AWL, se podrá acceder a todos los 64 bytes de la memoria L, pero se recomienda no utilizar los últimos cuatro bytes de la misma.

La memoria local es similar a la memoria V (memoria de variables), con una excepción: la memoria V tiene un alcance global, en tanto que la memoria L tiene un alcance local. El término "alcance global" significa que a una misma dirección de la memoria se puede acceder desde cualquier parte del programa (programa principal, subrutinas o rutinas de interrupción). El término "alcance local" significa que la dirección de la memoria está asociada a una determinada parte del programa. Las CPUs S7-200 asignan 64 bytes de la memoria L al programa principal, 64 bytes a cada nivel de anidado de las subrutinas y 64 bytes a las rutinas de interrupción.

A los bytes de la memoria L asignados al programa principal no se puede acceder ni desde las subrutinas ni desde las rutinas de interrupción. Una subrutina no puede acceder a la asignación de la memoria L del programa principal, ni a la de una rutina de interrupción, ni tampoco a la de otra subrutina. De igual manera, una rutina de interrupción no puede acceder a la asignación de la memoria L del programa principal ni tampoco a la de una subrutina.

La CPU S7-200 asigna la memoria L según sea necesario en ese momento. Ello significa que mientras se está ejecutando la parte principal del programa, no existen las asignaciones de la memoria L para las subrutinas y las rutinas de interrupción. Cuando ocurre una interrupción o cuando se llama a una subrutina, la memoria local se asigna según sea necesario. La nueva asignación de la memoria L puede reutilizar las mismas direcciones de la memoria L de una subrutina o de una rutina de interrupción diferentes.

La CPU no inicializa la memoria L durante la asignación de direcciones, pudiendo contener cualquier valor. Al transferir parámetros formales a una llamada de subrutina, los valores de los parámetros que se transfieren se depositarán en las direcciones de la memoria L que se hayan asignado a dicha subrutina. Las direcciones de la memoria L que no reciban un valor como resultado de la transferencia de parámetros formales no se inicializarán, pudiendo contener cualquier valor en el momento de la asignación.

Se puede acceder a la memoria L en formato de bits, bytes, palabras o palabras dobles. La memoria L se puede utilizar como puntero de direccionamiento indirecto, pero a las direcciones de la memoria L no se puede acceder indirectamente.

Formato:

Bit	<b>L [direcc. del byte].[direcc. del bit]</b>	<b>L0.0</b>
Byte, palabra, palabra doble	<b>L [tamaño][direcc. del byte inicial]</b>	<b>LB33</b>

### Direccionamiento del área de temporizadores (T)

En las CPUs S7-200, los temporizadores son elementos que cuentan intervalos de tiempo. Los temporizadores de las CPUs S7-200 tienen resoluciones (intervalos) de 1 ms, 10 ms y 100 ms. Hay dos variables asociadas a los temporizadores:

- Valor actual: en este número entero de 16 bits con signo se deposita el valor de tiempo contado por el temporizador.
- Bit del temporizador (bit T): este bit se activa o se desactiva como resultado de la comparación del valor actual con el valor de preselección. Este último se introduce como parte de la operación del temporizador.

A estas dos variables se accede utilizando la dirección del temporizador (T + número del temporizador). Dependiendo de la operación utilizada, se accede al bit del temporizador o al valor actual. Las operaciones con operandos en formato de bit acceden al bit del temporizador, en tanto que las operaciones con operandos en formato de palabra acceden al valor actual. Como muestra la figura 5-3, la operación Contacto normalmente abierto accede al bit del temporizador, en tanto que la operación Transferir palabra (MOV\_W) accede al valor actual del temporizador. Para obtener más información acerca de las operaciones S7-200, consulte el capítulo 9 (operaciones SIMATIC) o el capítulo 10 (operaciones IEC 1131-3).

Formato:  $T[\text{número del temporizador}]$  T24

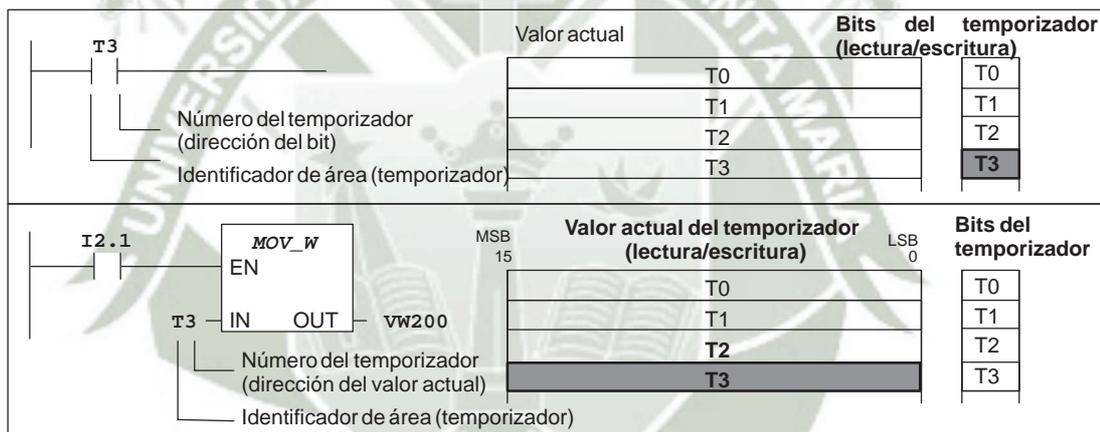


Figura 5-3 Acceso a los datos del temporizador SIMATIC

### Direccionamiento de los contadores (C)

Los contadores de las CPUs S7-200 son elementos que cuentan los cambios de negativo a positivo en la(s) entrada(s) de contaje. Hay tres tipos de contadores: uno que cuenta sólo adelante, uno que cuenta atrás y uno que cuenta tanto adelante como atrás. Hay dos variables asociadas a los contadores:

- Valor actual: en este número entero de 16 bits con signo se deposita el valor de contaje acumulado.
- Bit del contador (bit C): este bit se activa o se desactiva como resultado de la comparación del valor actual con el valor de preselección. El valor de preselección se introduce como parte de la operación del contador.

A estas dos variables se accede utilizando la dirección del contador (C + número del contador). Dependiendo de la operación utilizada, se accede al bit del contador o al valor actual. Las operaciones con operandos en formato de bit acceden al bit del contador, en tanto que las operaciones con operandos en formato de palabra acceden al valor actual. Como muestra la figura 5-4, la operación Contacto normalmente abierto accede al bit del contador, en tanto que la operación Transferir palabra (MOV\_W) accede al valor actual del contador. Para obtener más información acerca de las operaciones S7-200, consulte el capítulo 9 (operaciones SIMATIC) o el capítulo 10 (operaciones IEC 1131-3).

Formato:  $C[\text{número del contador}]$  C20

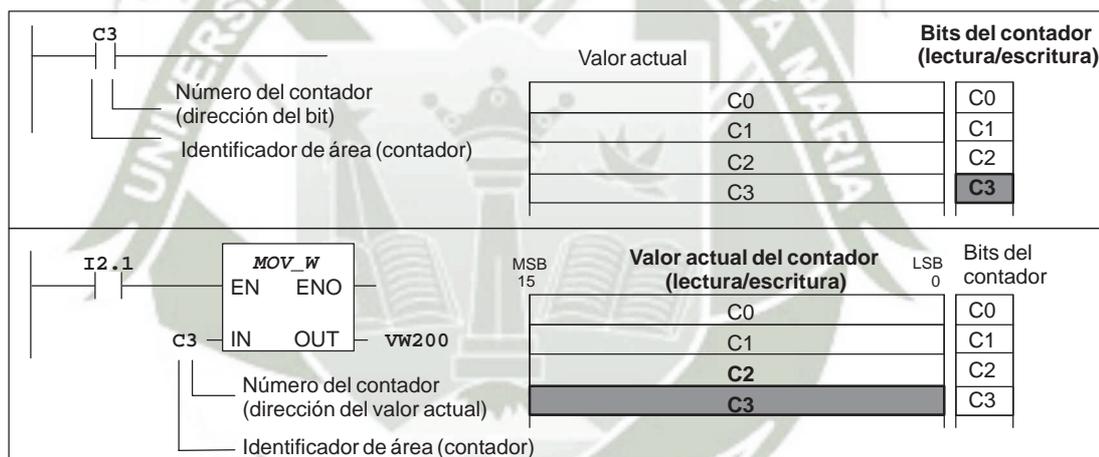


Figura 5-4 Acceso a los datos del contador SIMATIC

### Direccionamiento de las entradas analógicas (AI)

La CPU S7-200 convierte valores reales analógicos (p.ej. temperatura, tensión, etc). en valores digitales en formato de palabra (de 16 bits). A estos valores se accede con un identificador de área (AI), seguido del tamaño de los datos (W) y de la dirección del byte inicial. Puesto que las entradas analógicas son palabras que comienzan siempre en bytes pares (p.ej. 0, 2, 4, etc)., es preciso utilizar direcciones con bytes pares (p.ej. AIW0, AIW2, AIW4, etc). para acceder a las mismas, como muestra la figura 5-5. Las entradas analógicas son valores de sólo lectura.

Formato:  $AIW[dirección\ del\ byte\ inicial]$  **AIW4**

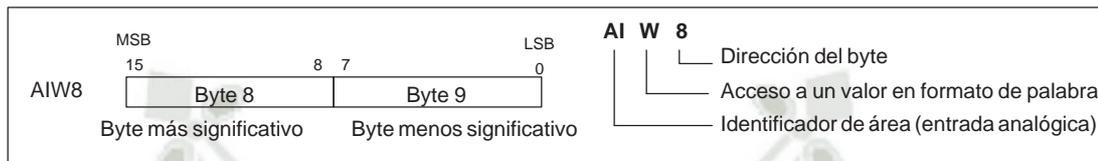


Figura 5-5 Acceso a una entrada analógica

### Direccionamiento de las salidas analógicas (AQ)

La CPU S7-200 convierte valores digitales en formato de palabra (de 16 bits) en valores reales analógicos (p.ej. intensidad o tensión), proporcionales al valor digital. A estos valores se accede con un identificador de área (AQ), seguido del tamaño de los datos (W) y de la dirección del byte inicial. Puesto que las salidas analógicas son palabras que comienzan siempre en bytes pares (p.ej. 0, 2, 4, etc)., es preciso utilizar direcciones con bytes pares (p.ej. AQW0, AQW2, AQW4, etc). para acceder a las mismas, como muestra la figura 5-6. Las salidas analógicas son valores de sólo escritura.

Formato:  $AQW[dirección\ del\ byte\ inicial]$  **AQW4**

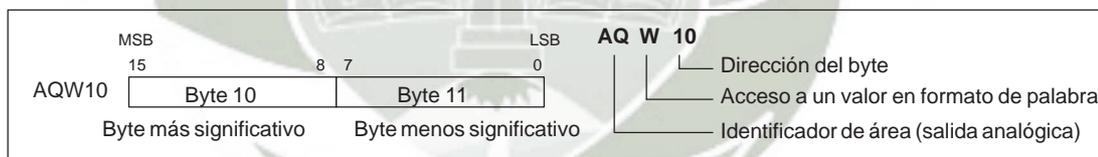


Figura 5-6 Acceso a una salida analógica

### Direccionamiento de los acumuladores (AC)

Los acumuladores son elementos de lectura/escritura que se utilizan igual que una memoria. Los acumuladores se pueden usar p.ej. para transferir parámetros de y a subrutinas, así como para almacenar valores intermedios utilizados en cálculos. La CPU dispone de cuatro acumuladores de 32 bits (AC0, AC1, AC2 y AC3). A los acumuladores se puede acceder en formato de byte, palabra o palabra doble. Como muestra la figura 5-7, cuando se accede a un acumulador en formato de byte o de palabra se utilizan los 8 ó 16 bits menos significativos del valor almacenado en el acumulador. Cuando se accede a un acumulador en formato de palabra doble, se usan todos los 32 bits. La operación utilizada para el acceso al acumulador determina el tamaño de los datos a los que se accede.

Formato:  $AC[\text{número del acumulador}]$  AC0

#### Nota

Consulte el apartado 9.16 (Operaciones de comunicación SIMATIC) en el capítulo 9 para obtener más información acerca de cómo utilizar los acumuladores en las rutinas de interrupción.

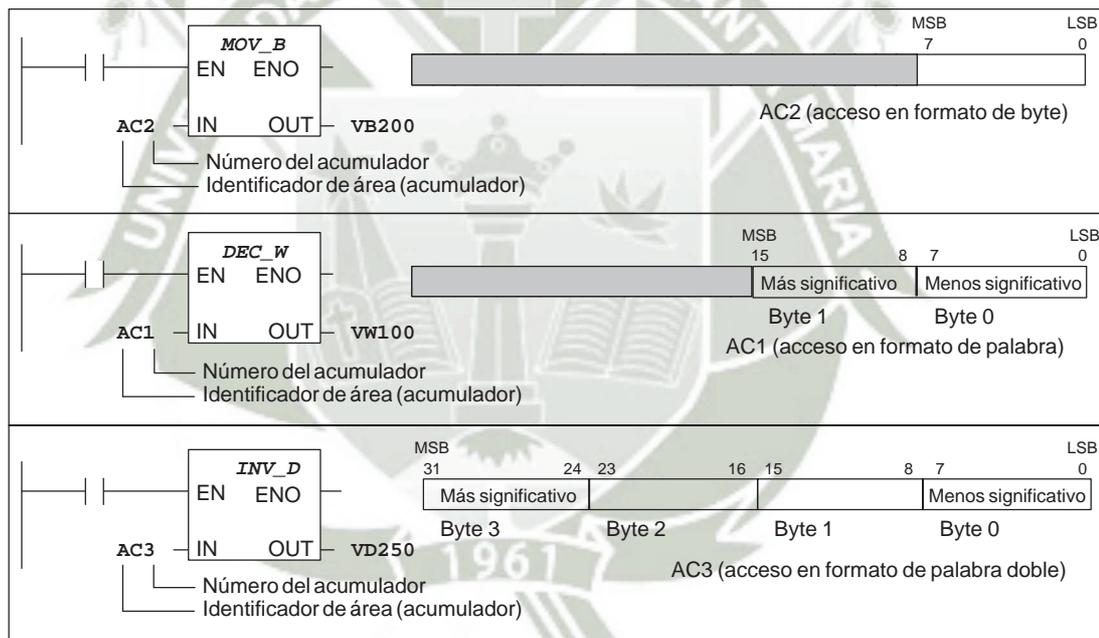


Figura 5-7 Acceso a los acumuladores

### Direccionamiento de los contadores rápidos (HC)

Los contadores rápidos se han diseñado para contar eventos muy rápidos, independientemente del ciclo de la CPU. Tienen un valor de contaje de entero de 32 bits con signo (denominado también valor actual). Para acceder al valor de contaje del contador rápido, se indica la dirección del mismo (utilizando el identificador HC) y el número del contador (p.ej. HC0). El valor actual del contador rápido es de sólo lectura, pudiéndose acceder al mismo sólo en formato de palabra doble de 32 bits, como muestra la figura 5-8.

Formato:  $HC[\text{número del contador rápido}] \quad HC1$

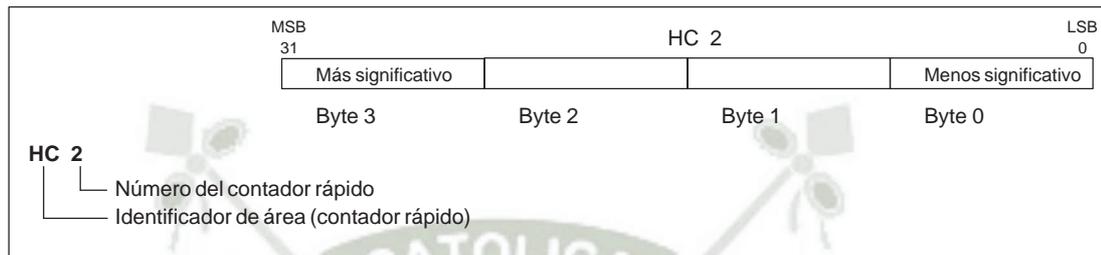
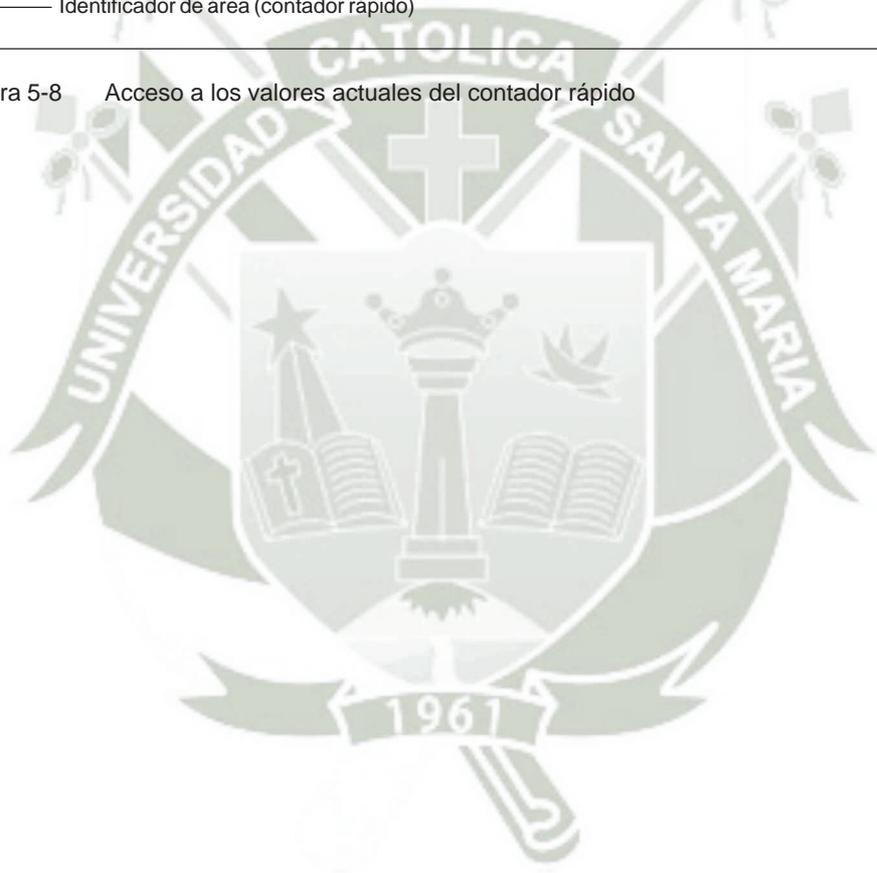


Figura 5-8 Acceso a los valores actuales del contador rápido



## Utilizar valores de constantes

Las constantes se pueden utilizar en numerosas operaciones de la CPU S7-200. Pueden ser valores de bytes, palabras o palabras dobles. La CPU almacena todas las constantes como números binarios que se pueden representar en formato decimal, hexadecimal, ASCII o en coma flotante.

Formato decimal:                    [valor decimal]  
Formato hexadecimal:            16#[valor hexadecimal]  
Formato ASCII:                    .'[texto ASCII]'  
Formato real o en coma flotante:   ANSI/IEEE 754-1985

El formato binario es el siguiente:                    2#1010\_0101\_1010\_0101

La CPU S7-200 no permite indicar tipos de datos específicos ni comprobar datos (p.ej. indicar si la constante es un entero de 16 bits, un entero con signo o un entero de 32 bits). Por ejemplo, la operación Sumar puede utilizar el valor depositado en VW100 como entero con signo, en tanto que una operación de combinación con O-exclusiva puede emplear ese mismo valor de VW100 como valor binario sin signo.

A continuación se indican ejemplos de constantes en formato decimal, hexadecimal, ASCII y en coma flotante:

- Constante decimal:                    20047
- Constante hexadecimal:                16#4E4F
- Constante ASCII:                    'El texto aparece entre comillas sencillas.'
- Formato real o en coma flotante:   +1.175495E-38 (positivo)  
  -1.175495E-38 (negativo)
- Formato binario                        2#1010\_0101\_1010\_0101

## 5.2 Direccionamiento indirecto (SIMATIC) de las áreas de memoria de la CPU

El direccionamiento indirecto utiliza un puntero para acceder a los datos de la memoria. La CPU S7-200 permite utilizar punteros para direccionar indirectamente las siguientes áreas de memoria: I, Q, V, M, S, T (sólo el valor actual) y C (sólo el valor actual). Los valores analógicos o de bits individuales no se pueden direccionar de forma indirecta.

### Crear un puntero

Para acceder indirectamente a una dirección en la memoria es preciso crear primero un puntero que señale a esa dirección. Los punteros son valores de palabra doble que señalan a otra dirección en la memoria. Como punteros sólo se pueden utilizar direcciones de las memorias V y L, o bien los acumuladores (AC1, AC2 y AC3). Para crear un puntero se debe utilizar la operación Transferir palabra doble (MOVD) con objeto de transferir la dirección indirecta a la del puntero. El operando de entrada de la operación debe ir precedido de un carácter "&" para determinar que a la dirección indicada por el operando de salida (es decir, el puntero) se debe transferir la dirección y no su contenido.

*Ejemplo:*

```
MOVD    &VB100, VD204
MOVD    &MB4, AC2
MOVD    &C4, L6
```

### Utilizar un puntero para acceder a los datos

Introduciendo un asterisco (\*) delante de un operando de una operación, se indica que el operando es un puntero. En el ejemplo que muestra la figura 5-9, \*AC1 significa que AC1 es el puntero del valor de palabra indicado por la operación Transferir palabra (MOVW). En este ejemplo, los valores almacenados en V200 y V201 se transfieren al acumulador AC0.

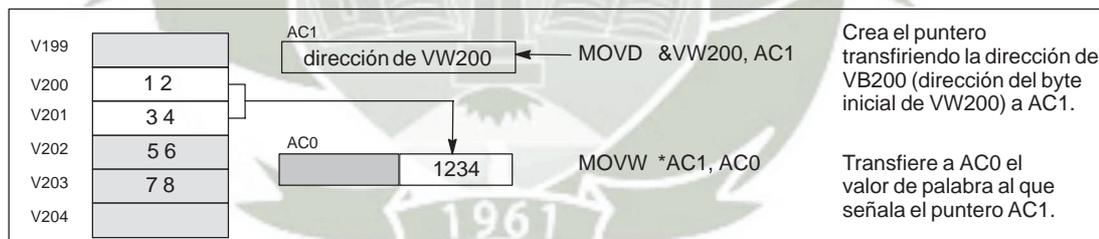


Figura 5-9 Utilizar un puntero para el direccionamiento indirecto

## Modificar punteros

Los valores de los punteros se puede modificar. Puesto que los punteros son valores de 32 bits, para cambiarlos es preciso utilizar operaciones de palabra doble. Las operaciones aritméticas simples, tales como sumar o incrementar, se pueden utilizar para modificar los valores de los punteros. Recuerde que debe indicar el tamaño de los datos a los que desee acceder:

- Para acceder a un byte, sume o incremente el valor del puntero en 1.
- Para acceder a una palabra, o bien al valor actual de un temporizador o de un contador, sume o incremente el valor del puntero en 2.
- Para acceder a una palabra doble, sume o incremente el valor del puntero en 4.

La figura 5-10 muestra un ejemplo de cómo crear un puntero de direccionamiento indirecto y de cómo acceder indirectamente a los datos e incrementar el puntero.

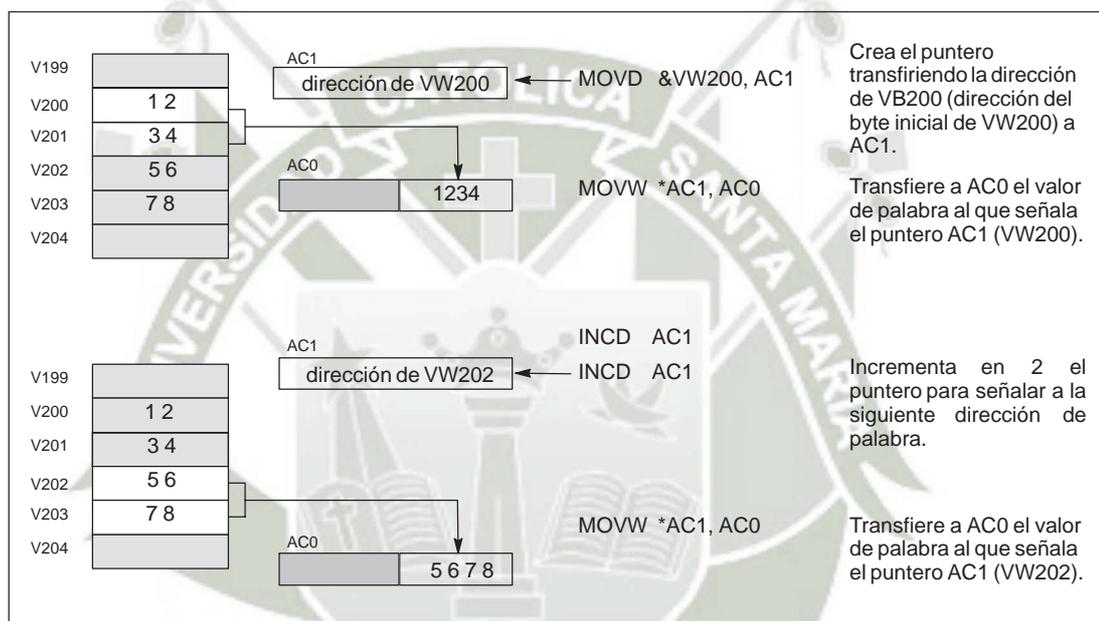


Figura 5-10 Modificar un puntero al acceder a un valor de palabra

### 5.3 Respaldo datos en la CPU S7-200

La CPU S7-200 ofrece diversos métodos para garantizar que el programa, los datos del mismo y los datos de configuración de la CPU se almacenen de forma segura (v. fig. 5-11).

- La CPU dispone de una EEPROM no volátil para almacenar todo el programa, así como las áreas de datos de usuario y la configuración de la CPU.
- La CPU dispone de un condensador de alto rendimiento que conserva todo el contenido de la memoria RAM después de un corte de alimentación. Según el tipo de CPU, el condensador puede respaldar la memoria durante varios días.
- La CPU soporta un cartucho de pila opcional que prolonga el tiempo durante el que se puede respaldar la memoria RAM después de un corte de alimentación. El cartucho de pila se activa sólo cuando se descarga el condensador de alto rendimiento.

En el presente apartado se describe el almacenamiento no volátil y el respaldo de los datos en la RAM en determinadas circunstancias.

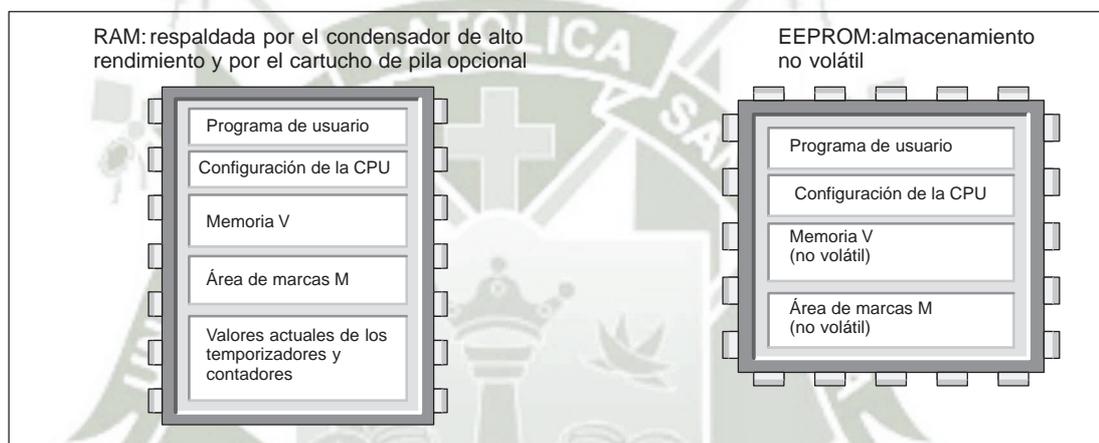


Figura 5-11 Áreas de memoria de la CPU S7-200

#### Cargar el proyecto en la CPU y en la PG

El proyecto comprende tres elementos: el programa de usuario, el bloque de datos (opcional) y la configuración de la CPU (opcional). Como muestra la figura 5-12, cargando el proyecto en la CPU se almacenan dichos elementos en la memoria RAM (de la CPU). La CPU también copia automáticamente el programa de usuario, el bloque de datos (DB1) y la configuración de la CPU en la EEPROM no volátil para que se almacenen allí.

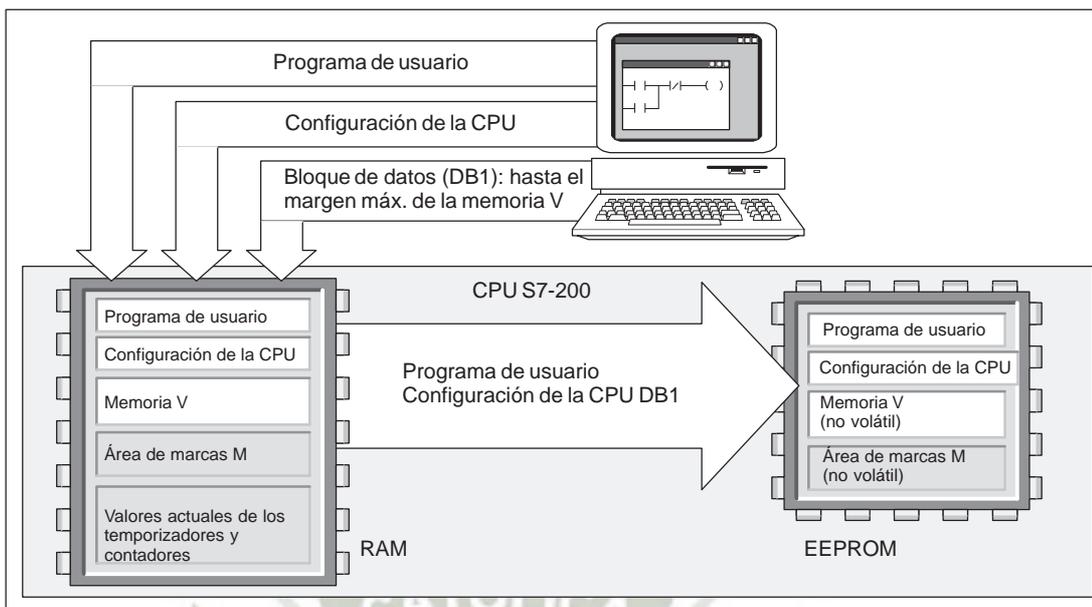


Figura 5-12 Cargar los elementos del proyecto en la CPU

Como muestra la figura 5-13, cuando un proyecto se carga en el PC (o la PG) desde la CPU, la configuración de la CPU se carga en el PC (o la PG) desde la RAM. El programa de usuario y el área no volátil de la memoria V se cargan en el PC desde la EEPROM, en tanto que la configuración de la CPU se carga en el PC desde la RAM.

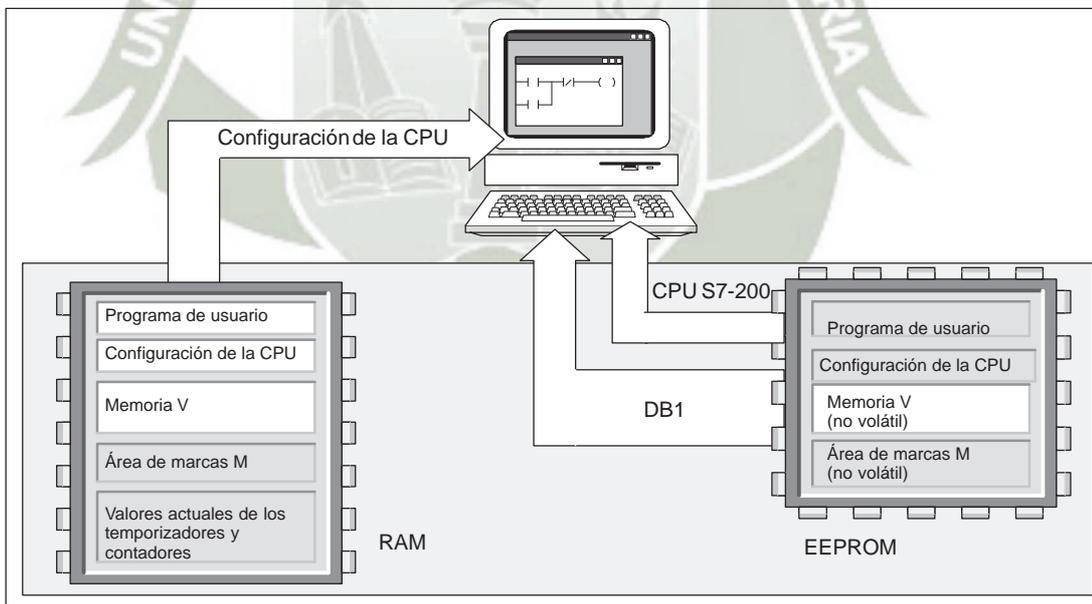


Figura 5-13 Cargar los elementos del proyecto en el PC (o la PG)

### Almacenar automáticamente los datos del área de marcas (M) en caso de un corte de alimentación

Si se define que los primeros 14 bytes del área de marcas (MB0 a MB13) sean remanentes, se copiarán automáticamente en la EEPROM cuando se produzca un corte de alimentación de la CPU. Como muestra la figura 5-14, la CPU transfiere dichas áreas remanentes del área de marcas a la EEPROM. En STEP 7-Micro/WIN 32, el ajuste estándar es "off".

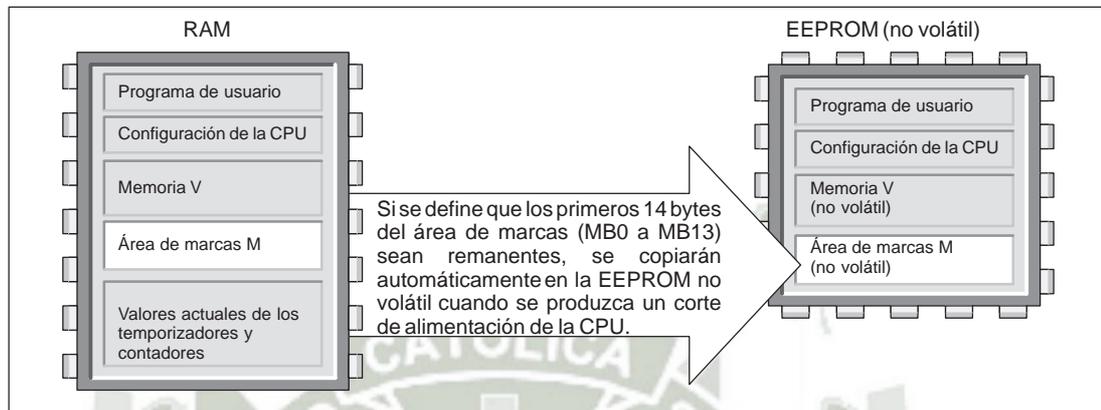


Figura 5-14 Almacenar partes del área de marcas (M) en la EEPROM en caso de un corte de alimentación

### Respaldar la memoria al conectar la alimentación

Cuando se conecta la alimentación, la CPU restablece el programa de usuario y la configuración (de la CPU) que se han depositado en la memoria EEPROM (v. fig. 5-15).

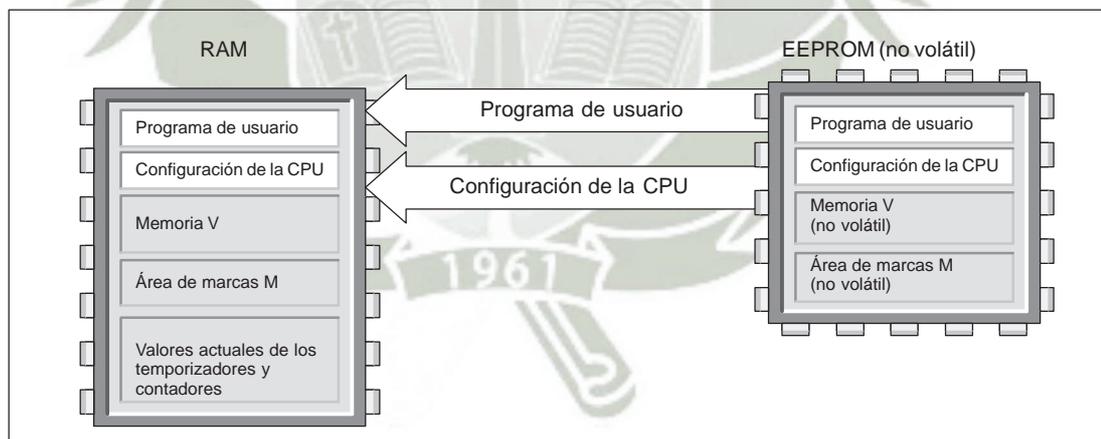


Figura 5-15 Restablecer el programa de usuario y la configuración de la CPU al conectar la alimentación

Cuando se conecta la alimentación, la CPU comprueba la memoria RAM para verificar si el condensador de alto rendimiento ha respaldado los datos almacenados en la RAM. En caso afirmativo, no se modificarán las áreas remanentes de la misma. Como muestra la figura 5-16, las áreas no remanentes de la memoria V se restablecen conforme a la correspondiente área no volátil de la memoria V contenida en la EEPROM.

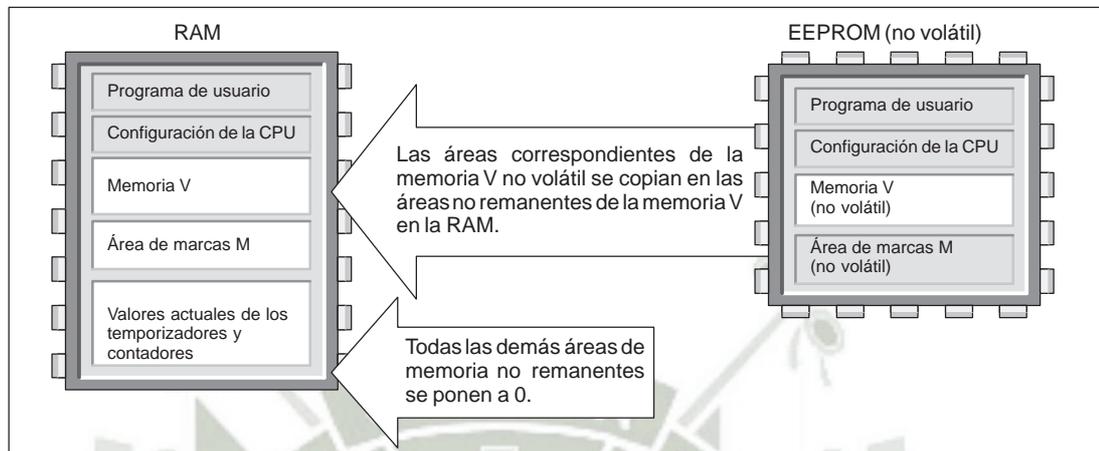


Figura 5-16 Restablecer los datos del programa al conectarse la alimentación (habiéndose respaldado los datos en la RAM)

Si el contenido de la RAM no se ha respaldado (p.ej. en el caso de un corte de alimentación prolongado), la CPU borra la memoria RAM (tanto las áreas remanentes como las no remanentes) y activa la marca Datos remanentes perdidos (SM0.2) en el primer ciclo que le sigue a la puesta en marcha. Como muestra la figura 5-17, los datos almacenados en la EEPROM no volátil se copian entonces en la memoria RAM.

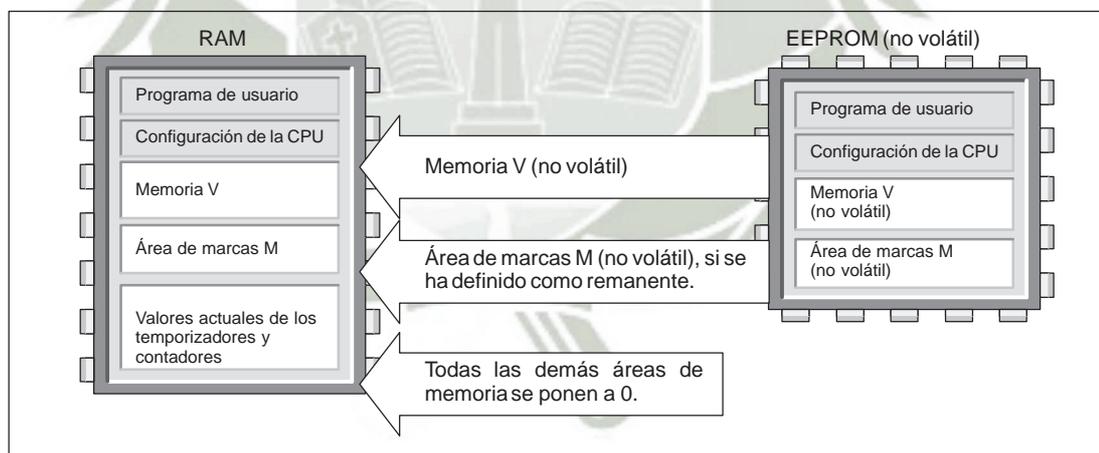


Figura 5-17 Restablecer los datos del programa al conectarse la alimentación (sin haberse respaldado los datos en la RAM)

### Definir las áreas remanentes

El usuario puede definir hasta seis áreas remanentes para elegir las áreas de memoria que se deberán respaldar cuando se interrumpa la alimentación (v. figura 5-18). Se puede determinar que sean remanentes los márgenes de direcciones en las áreas de memoria V, M, C y T. En el caso de los temporizadores, sólo es posible respaldar los de retardo a la conexión memorizado (TONR). En STEP 7-Micro/WIN 32, el ajuste estándar de la memoria M es no remanente. Dicho ajuste inhibe la función "power down" de la CPU.

#### Nota

Sólo se pueden respaldar los valores actuales de los temporizadores y contadores. Los bits de los temporizadores y de los contadores no son remanentes.

Para definir las áreas remanentes, elija el comando de menú **Ver > Bloque de sistema** y haga clic en la ficha "Áreas remanentes". La figura 5-18 muestra el cuadro de diálogo donde se definen dichas áreas. Para visualizar las áreas remanentes predeterminadas de la CPU, haga clic en el botón **Estándar**.

Área	Área de datos	Offset	Nº de elementos	
Área 0:	VB	0	5120	Borrar
Área 1:	VB	0	0	Borrar
Área 2:	T	0	32	Borrar
Área 3:	T	64	32	Borrar
Área 4:	C	0	256	Borrar
Área 5:	MB	14	18	Borrar

Para que los parámetros de configuración tengan efecto, es preciso cargarlos previamente en la CPU.

No todos los tipos de CPUs asisten todas las opciones del bloque de sistema. Pulse F1 para visualizar las opciones asistidas por cada CPU.

Figura 5-18 Configurar las áreas remanentes de la memoria de la CPU

## 5.4 Utilizar el programa para guardar datos en la memoria no volátil

Cualquier valor (byte, palabra o palabra doble) almacenado en la memoria de variables (memoria V) se puede guardar en la memoria EEPROM. Esta función se puede utilizar para almacenar un valor en cualquier dirección de la memoria V no volátil.

Por lo general, la operación de guardar en EEPROM prolonga 5 ms el tiempo de ciclo. Si en esta operación se escribe un valor en el área no volátil de la memoria V de la EEPROM no volátil, se sobrescribirá el valor anterior de dicha dirección.

### Nota

La operación de guardar en EEPROM no actualiza los datos contenidos en el cartucho de memoria.

### Copiar la memoria de variables en la EEPROM

El byte de marcas 31 (SMB31) y la palabra de marcas 32 (SMW32) indican a la CPU que copie un valor de la memoria V en el área no volátil de la memoria V de la EEPROM. La figura 5-19 muestra el formato de SMB31 y SMW32. Para programar la CPU para guardar o escribir un valor determinado en la memoria V, siga los siguientes pasos:

1. Cargue la dirección de la memoria V del valor a almacenar en SMW32.
2. Cargue el tamaño de los datos en SM31.0 y SM31.1 (v. fig. 5-19).
3. Active la marca SM31.7.

Al final de cada ciclo, la CPU comprueba el estado de SM31.7. Si SM31.7 está activada (puesta a 1), el valor indicado se guardará en la EEPROM. La operación se finalizará cuando la CPU desactive SM31.7. No cambie el valor en la memoria V antes de finalizar la operación de guardar.

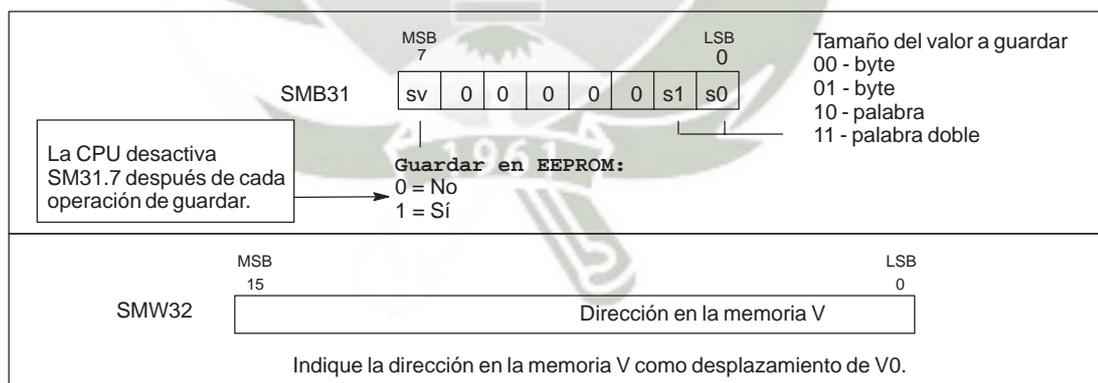


Figura 5-19 Formato de SMB31 y SMW32

### Limitar el número de las operaciones de guardar en EEPROM

Puesto que el número de operaciones de guardar en EEPROM es limitado (mín. 100.000, típ. 1.000.000), sólo se deberán almacenar los valores realmente necesarios. De lo contrario, es posible que se sobrecargue la EEPROM y que falle la CPU. Generalmente, las operaciones de guardar se ejecutan sólo cuando se presentan determinados eventos, lo cual no suele ocurrir con frecuencia.

Por ejemplo, si el tiempo de ciclo del S7-200 es de 50 ms y un valor se almacena una vez por ciclo, la EEPROM se llenaría al cabo de 5.000 segundos, es decir, en menos de una hora y media. En cambio, si dicho valor se almacena cada hora, la EEPROM podría utilizarse 11 años como mínimo.



## 5.5 Guardar el programa en un cartucho de memoria

Las CPUs asisten un cartucho de memoria opcional que permite almacenar el programa en una EEPROM portátil. La CPU guarda los siguientes datos en el cartucho de memoria:

- Programa de usuario
- Datos almacenados en la memoria de variables no volátil de la EEPROM
- Configuración de la CPU

Para obtener más información sobre el cartucho de memoria, consulte el Anexo A.

### Copiar en el cartucho de memoria

El programa se puede copiar en el cartucho de memoria desde la RAM sólo si se ha arrancado la CPU, si ésta se encuentra en modo STOP y si dicho cartucho está insertado.



---

#### Cuidado

Las descargas electrostáticas pueden deteriorar el cartucho de memoria o su receptáculo en la CPU.

Cuando utilice el cartucho de memoria, deberá estar en contacto con una superficie conductiva puesta a tierra y/o llevar puesta una pulsera puesta a tierra. Guarde el cartucho en una caja conductiva.

---

El cartucho de memoria se puede enchufar o extraer estando conectada la alimentación de la CPU. Para enchufarlo, retire la tapa de plástico de la CPU e inserte el cartucho en ésta última. (El cartucho de memoria se ha diseñado de forma que sólo se pueda insertar en un sólo sentido en el receptáculo). Una vez insertado el cartucho, copie el programa como se indica a continuación.

1. Cargue el programa en la CPU si no lo ha hecho todavía.
2. Elija el comando de menú **CPU > Cartucho de memoria** para copiar el programa en el cartucho de memoria. La figura 5-20 muestra los componentes de la memoria de la CPU que se almacenan en dicho cartucho.
3. Extraiga el cartucho de memoria del receptáculo (opcional).

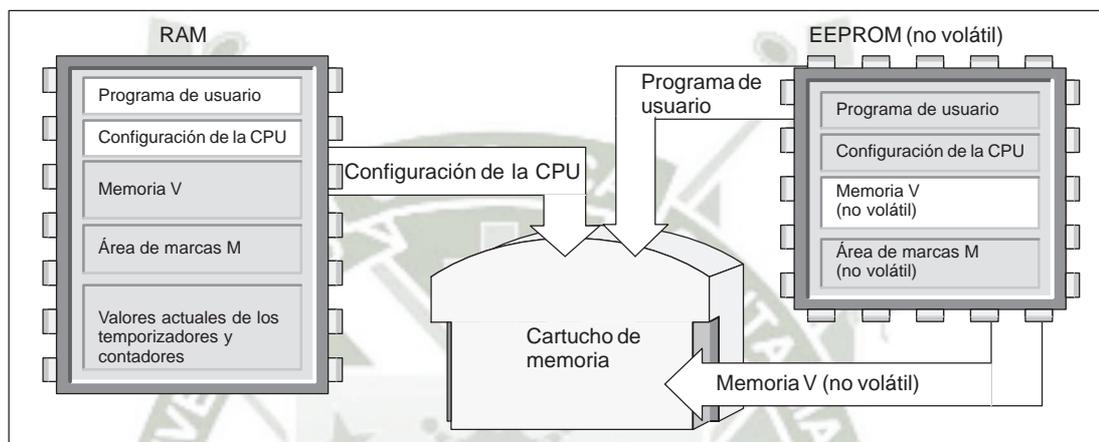


Figura 5-20 Copiar la memoria de la CPU en el cartucho de memoria

## Restablecer el programa y la memoria mediante un cartucho de memoria

Para transferir el programa de un cartucho de memoria en la CPU, es preciso desconectar la alimentación de la CPU y conectarla de nuevo con el cartucho insertado. Como muestra la figura 5-21, la CPU ejecuta las siguientes tareas después del arranque (si está insertado el cartucho de memoria):

- Se borra la memoria RAM.
- El contenido del cartucho de memoria se copia en la memoria RAM.
- El programa de usuario, la configuración de la CPU y la memoria V se copian en la EEPROM no volátil.

### Nota

Si se conecta la alimentación de la CPU estando insertado un cartucho de memoria vacío o programado con un modelo diferente de CPU, se puede producir un error. Los cartuchos de memoria programados en una CPU 221 ó 222 se pueden leer en una CPU 224. En cambio, los que se hayan programado en una CPU 224 serán rechazados por las CPUs 221 y 222.

Retire el cartucho de memoria y arranque la CPU de nuevo. Así podrá insertar y programar el cartucho.

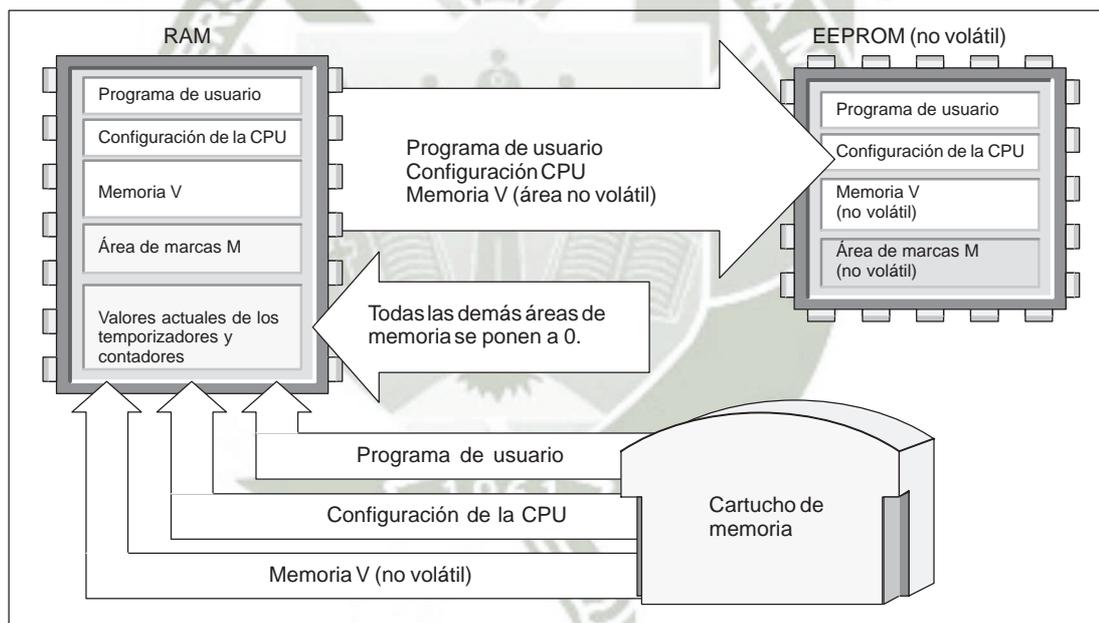


Figura 5-21 Restablecer la memoria durante el arranque (estando insertado un cartucho de memoria)

# 6

## Configurar las entradas y salidas

El sistema se controla mediante entradas y salidas (E/S). Las entradas vigilan las señales de los dispositivos de campo (p.ej. sensores e interruptores), mientras que las salidas supervisan las bombas, motores u otros aparatos del proceso. Se dispone de entradas y salidas integradas (en la CPU), así como de E/S adicionales (en los módulos de ampliación). Las CPUs S7-200 incorporan además entradas y salidas rápidas.

### Índice del capítulo

Apartado	Descripción	Página
6.1	Entradas y salidas integradas y adicionales	6-2
6.2	Utilizar filtros de entrada para suprimir interferencias	6-4
6.3	Capturar impulsos	6-5
6.4	Configurar los estados de señal de las salidas	6-8
6.5	Filtrar entradas analógicas	6-9
6.6	Entradas y salidas rápidas	6-10
6.7	Potenciómetros analógicos	6-13

## 6.1 Entradas y salidas integradas y adicionales

El sistema se controla mediante entradas y salidas (E/S). Las entradas vigilan las señales de los dispositivos de campo (p.ej. sensores e interruptores), mientras que las salidas supervisan las bombas, motores u otros aparatos del proceso. Se dispone de entradas y salidas integradas (en la CPU), así como de E/S adicionales (en los módulos de ampliación).

- Las CPUs S7-200 disponen de un número determinado de entradas y salidas digitales. Para obtener más información acerca de las E/S integradas en su CPU, consulte las hojas de datos técnicos que se incluyen en el Anexo A.
- Las CPUs 222 y 224 asisten módulos de ampliación con entradas y salidas tanto digitales como analógicas. Para obtener más información acerca de los diversos módulos de ampliación, consulte las hojas de datos técnicos que se incluyen en el Anexo A.

### Direccionar las E/S integradas y adicionales

Las entradas y salidas integradas en la unidad central de procesamiento (CPU) tienen direcciones fijas. Para añadir a la CPU entradas y salidas adicionales, se pueden conectar módulos de ampliación a la derecha de la CPU, formando una cadena de E/S. Las direcciones de las E/S de cada módulo vienen determinadas por el tipo de E/S y por la posición del módulo en la cadena, con respecto al anterior módulo de entradas o de salidas del mismo tipo. Por ejemplo, un módulo de salidas no afecta las direcciones de un módulo de entradas y viceversa. Igualmente, los módulos analógicos no afectan al direccionamiento de los módulos digitales y viceversa.

Los módulos de ampliación digitales reservan siempre un espacio de la imagen del proceso en incrementos de ocho bits (un byte). Si un módulo no dispone de un punto físico para cada bit de cada byte reservado, se pierden estos bits no utilizados y no se pueden asignar a los módulos siguientes en la cadena de E/S. En cuanto a los módulos de entradas, los bits no utilizados en los bytes reservados se ponen a cero cada vez que se actualizan las entradas.

Las direcciones de los módulos de ampliación analógicos se asignan siempre en incrementos de dos puntos. Si un módulo no ofrece E/S físicas para cada uno de dichos puntos, se pierden los mismos y no se pueden asignar a los módulos siguientes en la cadena de E/S.

### Ejemplos de E/S integradas y adicionales

Las figuras 6-1 y 6-2 muestran ejemplos de cómo las diferentes configuraciones del hardware afectan la numeración de las entradas y salidas. Tenga en cuenta que algunas configuraciones tienen espacios entre las direcciones que no se pueden utilizar en el programa.

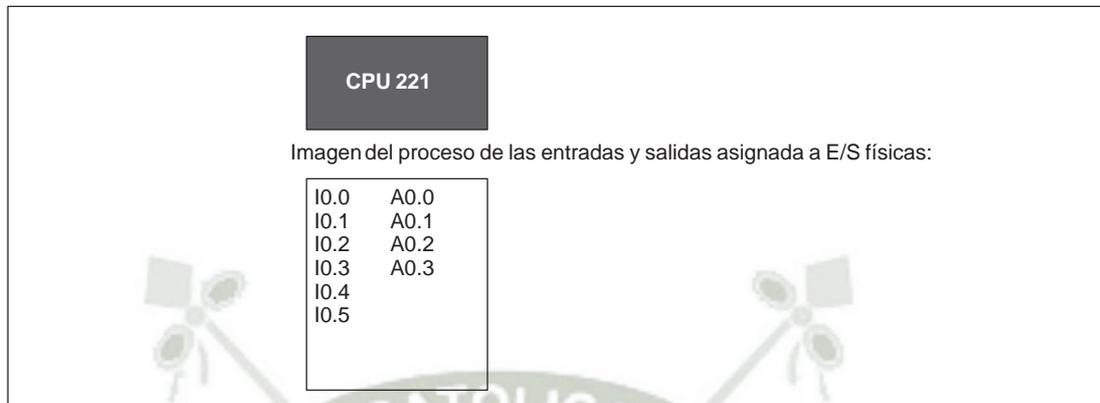


Figura 6-1 Ejemplos de numeración de E/S para una CPU 221

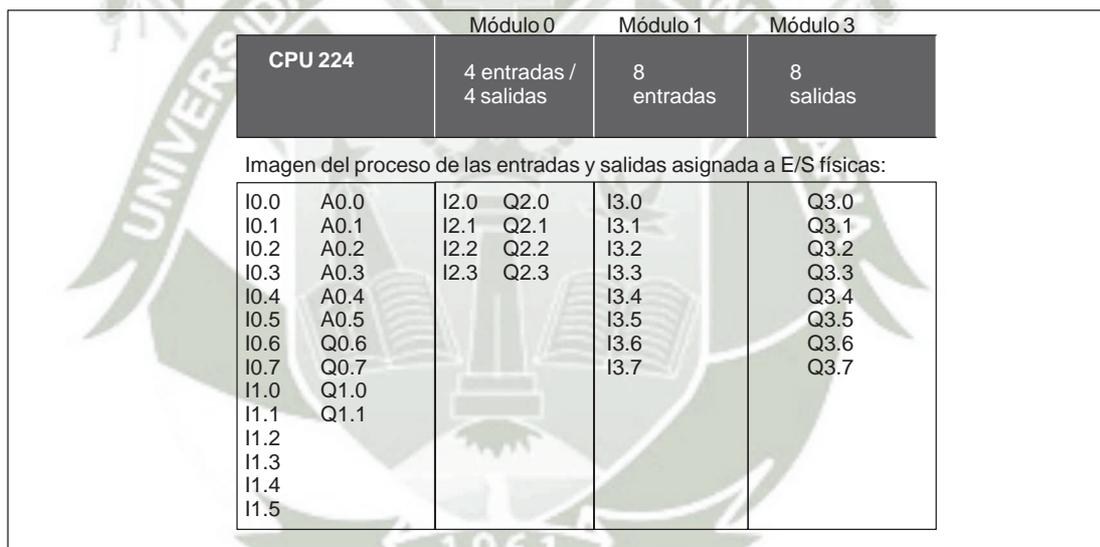


Figura 6-2 Ejemplos de numeración de E/S para una CPU 224

## 6.2 Utilizar filtros de entrada para suprimir interferencias

Las CPU S7-200 permiten seleccionar un filtro de entrada que define un tiempo de retardo (comprendido entre 0,2 ms y 12,8 ms) para algunas o bien para todas las entradas digitales integradas. (Para obtener más información acerca de su CPU, consulte el Anexo A). Como muestra la figura 6-3, cada indicación del tiempo de retardo se aplica a grupos de cuatro entradas. Dicho retardo permite filtrar en el cableado de entrada las interferencias que pudieran causar cambios accidentales de los estados de las salidas.

El filtro de entrada forma parte de los datos de configuración de la CPU que se cargan y se almacenan en la memoria de la misma. Para configurar los tiempos de retardo del filtro de entrada, elija el comando de menú **Ver > Bloque de sistema** y haga clic en la ficha "Filtros de entrada".

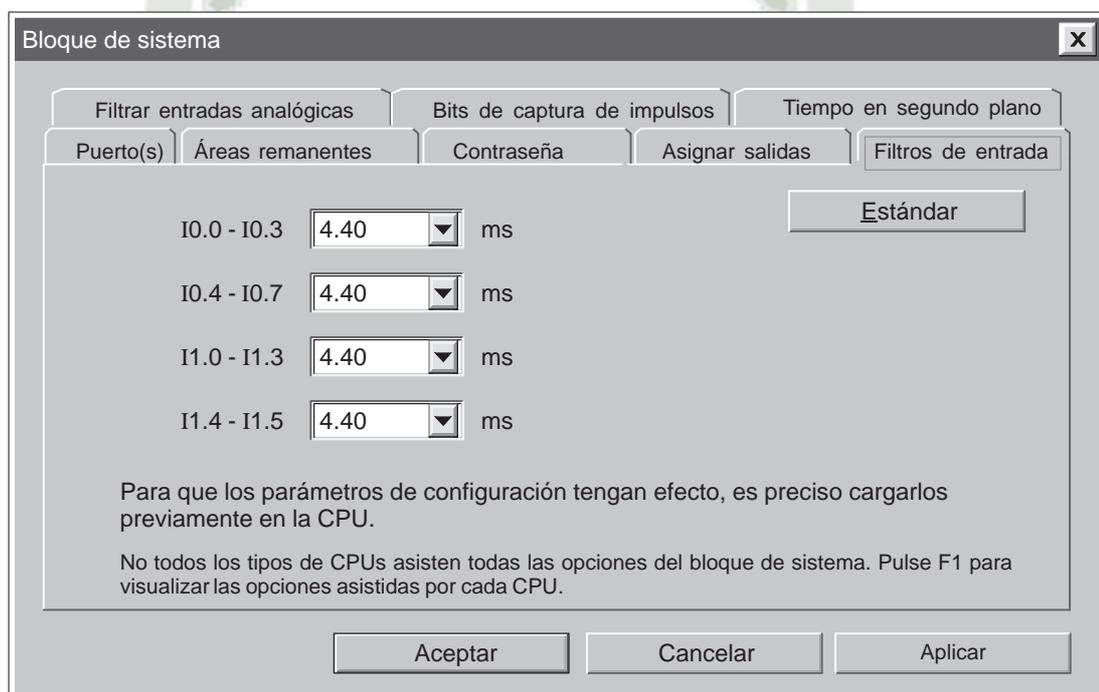


Figura 6-3 Configurar el filtro de entrada para suprimir interferencias

### 6.3 Capturar impulsos

Las CPUs S7-200 ofrecen una función de captura de impulsos para cada una de las entradas digitales integradas. Dicha función permite capturar impulsos altos o bajos de tan corta duración que no se registrarían en todos los casos, cuando la CPU lee las entradas digitales al comienzo del ciclo.

La función de captura de impulsos se puede habilitar individualmente para cada una de las entradas digitales integradas. Si se ha habilitado la captura de impulsos en una entrada y se produce un cambio de estado de la misma, dicho cambio se marcará, conservándose hasta que la entrada se actualice al comienzo del siguiente ciclo. De esta forma, un impulso de breve duración se capturará y se conservará hasta que la CPU lea las entradas, garantizando así que el impulso no pase desapercibido. La figura 6-4 muestra el funcionamiento básico de la CPU con y sin captura de impulsos.

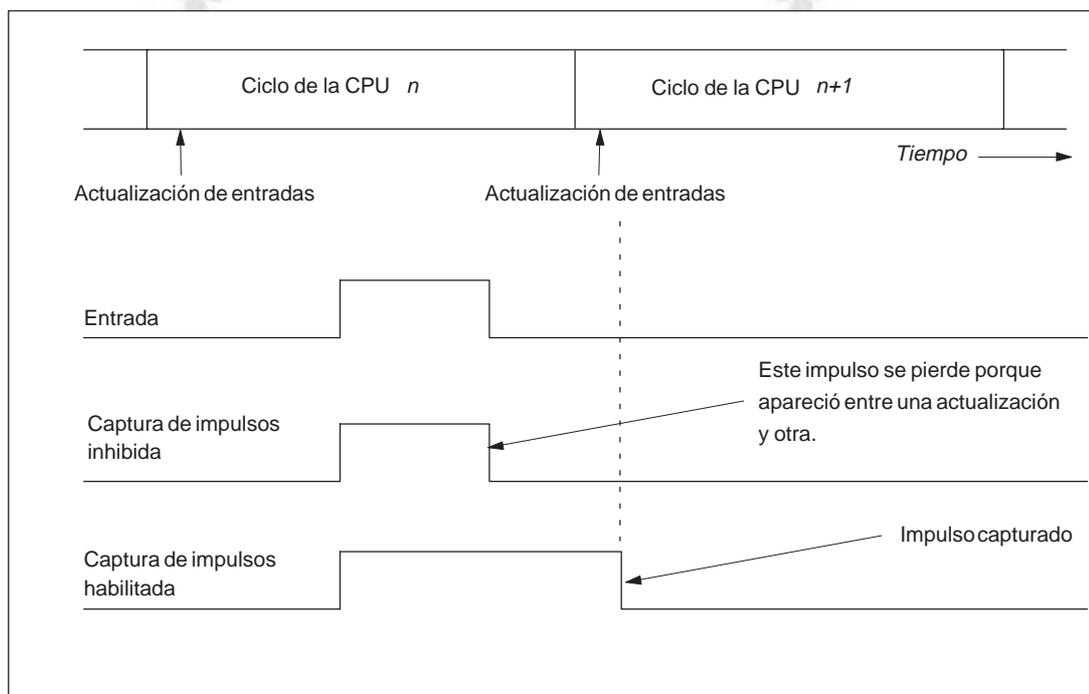


Figura 6-4 Funcionamiento de la CPU con y sin captura de impulsos

Cuando se utilice la función de captura de impulsos, el tiempo del filtro de entrada se deberá ajustar de manera que dicho filtro no elimine el impulso. (La función de captura de impulsos se ejecuta en la entrada tras haber pasado ésta por el filtro).

La figura 6-5 muestra un esquema funcional de una entrada digital.

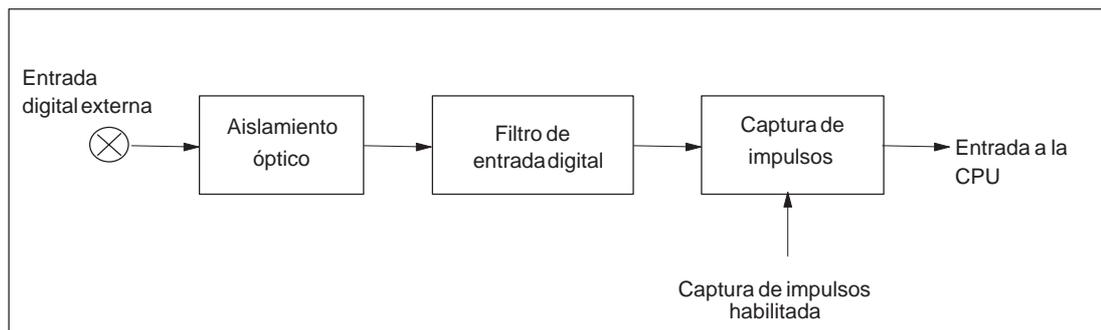


Figura 6-5 Esquema funcional de una entrada digital

La figura 6-6 muestra la reacción de un circuito de captura de impulsos a diversas condiciones de entrada.

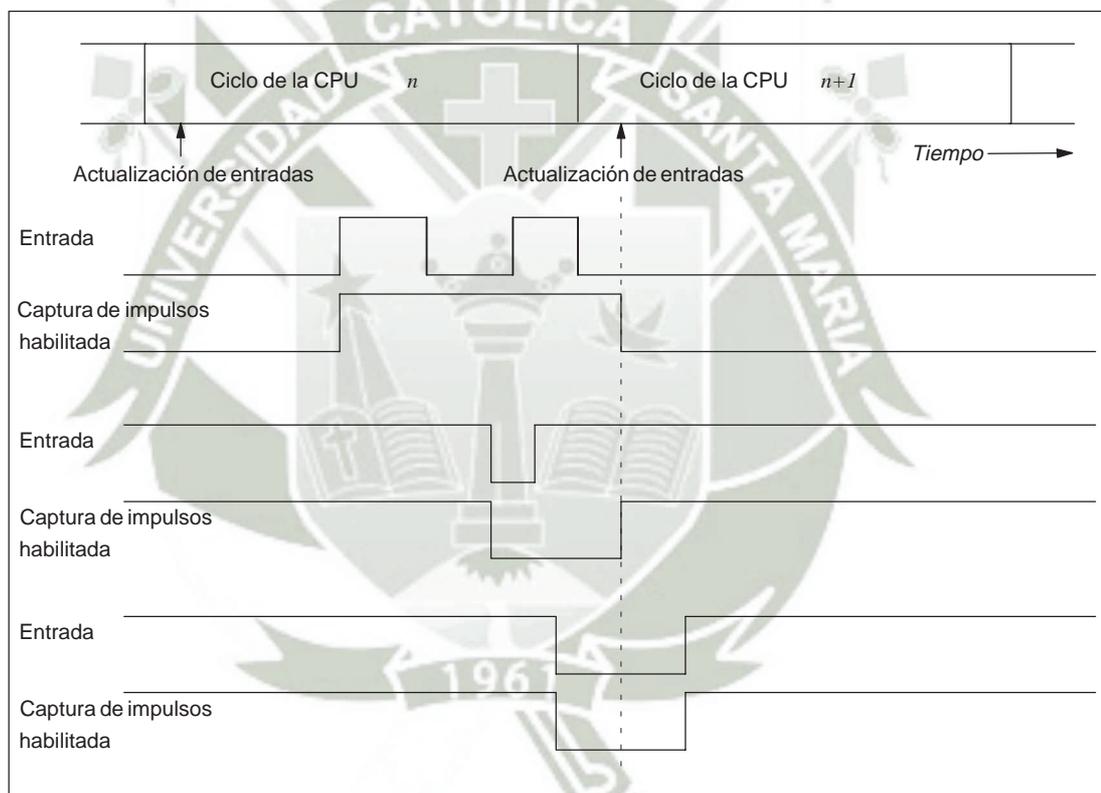


Figura 6-6 Ejemplo de captura de impulsos

Para acceder a la ventana donde se configura la captura de impulsos, elija el comando de menú **Ver > Bloque de sistema** y haga clic en la ficha "Bits de captura de impulsos". La figura 6-8 muestra la ventana para configurar la captura de impulsos. Las configuraciones estándar de la CPU y de STEP 7-Micro/WIN 32 se inhiben para todas las entradas.

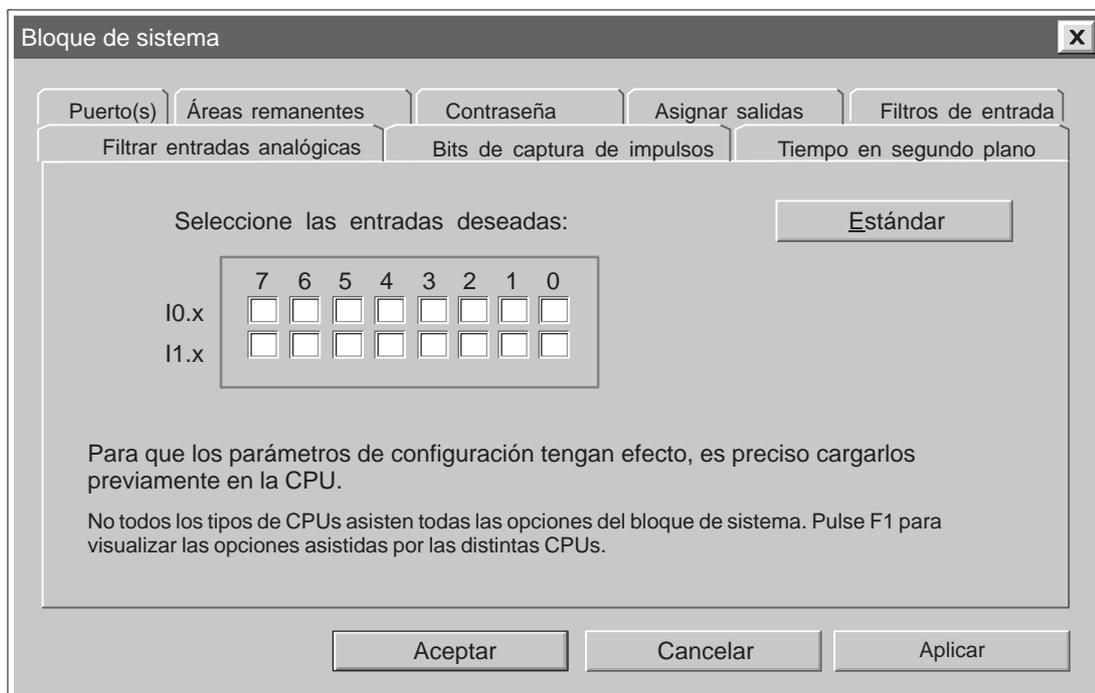


Figura 6-7 Ventana para configurar la captura de impulsos

## 6.4 Configurar los estados de señal de las salidas

La CPU S7-200 permite elegir si las salidas digitales deben adoptar valores conocidos cuando cambie a modo STOP, o bien congelar las salidas en su último estado antes de dicho cambio.

La tabla de salidas forma parte de los datos de configuración de la CPU que se cargan y se almacenan en la memoria de la misma.

La configuración de estos valores es aplicable sólo a las salidas digitales. Los valores de las salidas analógicas se congelan cuando se produce un cambio a modo STOP. La CPU no actualiza las entradas o salidas analógicas como una función del sistema, ni dispone tampoco de una imagen del proceso para las mismas.

Para acceder al cuadro de diálogo donde se configuran los estados de señal de las salidas, elija el comando de menú **Ver > Bloque de sistema** y haga clic en la ficha "Asignar salidas" (v. fig. 6-8). Hay dos opciones para configurar las salidas:

- Si desea que las salidas se congelen en su último estado, elija la opción "Congelar salidas" y haga clic en "Aceptar".
- Si desea copiar los valores de la tabla en las salidas, introduzca los correspondientes valores. Haga clic en la casilla de verificación correspondiente a cada salida que desee activar (poner a 1) cuando se produzca un cambio de RUN a STOP. Haga clic en "Aceptar" para guardar sus ajustes.

En la tabla, todas las salidas están desactivadas (puestas a 0) por defecto. Las configuraciones estándar de STEP 7-Micro/WIN 32 y de la CPU se inhiben para todas las salidas.

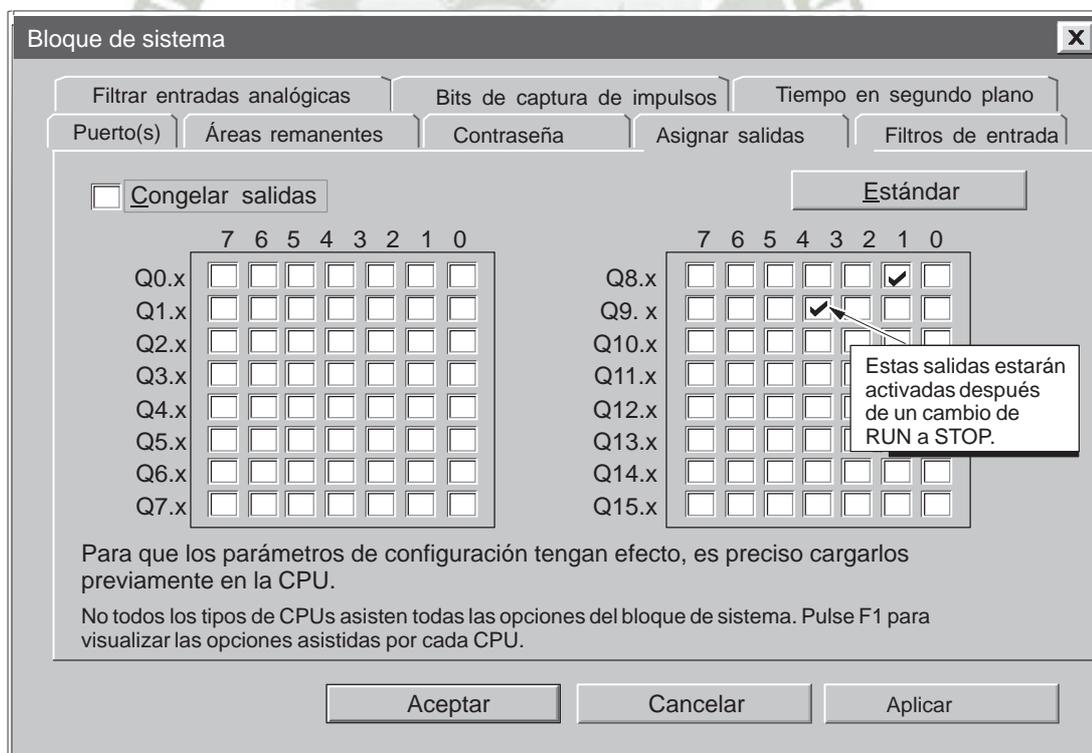


Figura 6-8 Configurar el estado de las salidas

## 6.5 Filtrar entradas analógicas

En las CPUs 222 y 224, cada una de las entradas analógicas se puede filtrar utilizando el software. El valor filtrado es el valor promedio de la suma de un número predeterminado de muestreos de la entrada analógica. Los datos de filtración indicados (número de impulsos y tiempo muerto) se aplica a todas las entradas analógicas para las que se habilite dicha función.

El filtro dispone de una función de respuesta rápida para que los cambios considerables se puedan reflejar rápidamente en el valor de filtración. El filtro cambia al último valor de la entrada analógica cuando ésta exceda una determinada diferencia del valor promedio. Dicha diferencia se denomina tiempo muerto, indicándose en contajes del valor digital de la entrada analógica.

### Nota

Verifique que en su aplicación se puedan filtrar las entradas analógicas. En caso contrario, en la ventana de configuración de STEP 7-Micro/WIN 32, inhiba el filtro de entradas analógicas, como muestra la figura 6-9.

Para acceder al filtro de entradas analógicas, elija el comando de menú **Ver > Bloque de sistema** y haga clic en la ficha "Filtrar entradas analógicas". Elija las entradas analógicas que desee filtrar y haga clic en "Aceptar" (v. figura 6-9). La configuración estándar de STEP 7-Micro/WIN 32 está habilitada para todas las entradas.

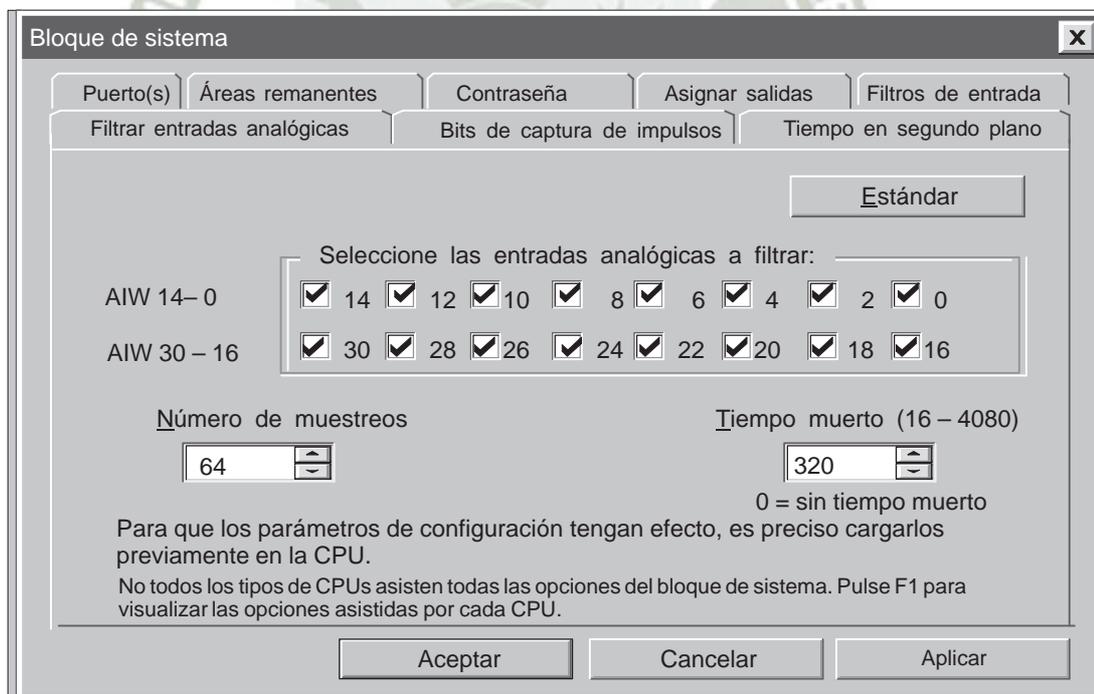


Figura 6-9 Filtrar entradas analógicas

## 6.6 Entradas y salidas rápidas

Las CPUs S7-200 disponen de entradas y salidas para controlar los eventos rápidos. Para obtener más información acerca de las entradas y salidas rápidas de cada modelo de CPU, consulte las hojas de datos técnicos que se incluyen en el Anexo A.

### Contadores rápidos

Las CPUs S7-200 disponen de contadores rápidos integrados que cuentan eventos externos a velocidades de hasta 20 KHz sin influir en el funcionamiento de la CPU. A continuación se describen dichos contadores rápidos:

- HSC0 y HSC4 son contadores versátiles que se pueden configurar para uno de ocho modos de operación diferentes, incluyendo entradas de reloj de fase simple y de dos fases.
- HSC1 y HSC2 son contadores versátiles que se pueden configurar para uno de doce modos de operación diferentes, incluyendo entradas de reloj de fase simple y de dos fases.
- HSC3 y HSC5 son contadores sencillos que tienen sólo un modo de operación (sólo entradas de reloj de fase simple).

La tabla 6-1 muestra los modos de operación asistidos por los contadores rápidos HSC0, HSC3, HSC4 y HSC5. Todas las CPUs S7-200 asisten dichos contadores rápidos.

Tabla 6-1 Contadores rápidos HSC0, HSC3, HSC4, HSC5

Modo	HSC0			HSC3	HSC4			HSC5
	I0.0	I0.1	I0.2	I0.1	I0.3	I0.4	I0.5	I0.4
0	Reloj	–	–	Reloj	Reloj	–	–	Reloj
1	Reloj	–	Puesta a 0	–	Reloj	–	Puesta a 0	–
2	–	–	–	–	–	–	–	–
3	Reloj	Sentido	–	–	Reloj	Sentido	–	–
4	Reloj	Sentido	Puesta a 0	–	Reloj	Sentido	Puesta a 0	–
5	–	–	–	–	–	–	–	–
6	Reloj adelante	Reloj atrás	–	–	Reloj adelante	Reloj atrás	–	–
7	Reloj adelante	Reloj atrás	Puesta a 0	–	Reloj adelante	Reloj atrás	Puesta a 0	–
8	–	–	–	–	–	–	–	–
9	Fase A	Fase B	–	–	Fase A	Fase B	–	–
10	Fase A	Fase B	Puesta a 0	–	Fase A	Fase B	Puesta a 0	–
11	–	–	–	–	–	–	–	–

Como muestra la tabla, si HSC0 se está utilizando en los modos 3 a 10 (reloj y sentido o alguna de las dos fases de reloj), HSC3 no se podrá utilizar, puesto que tanto HSC0 como HSC3 utilizan la entrada I0.1. Esto mismo es aplicable a HSC4 y HSC5 (ambos utilizan I0.4).

Las entradas I0.0 a I0.3 se pueden utilizar no sólo para los contadores rápidos, sino también para ocho eventos de interrupción de flanco. Dichas entradas no se pueden utilizar simultáneamente para interrupciones de flanco y para los contadores rápidos.

Una misma entrada no se puede utilizar para dos funciones diferentes. No obstante, cualquier entrada que no se esté utilizando en el modo actual del contador rápido se puede utilizar para otro fin. Por ejemplo, si HSC0 se está utilizando en modo 2 (que utiliza las entradas I0.0 e I0.2), I0.1 se podrá utilizar para interrupciones de flanco o para HSC3.

La tabla 6-2 muestra los modos de operación asistidos por los contadores rápidos HSC1 y HSC2. La CPU 224 es la única que soporta dichos contadores rápidos.

Tabla 6-2 Contadores rápidos HSC1 y HSC2

Modo	HSC1				HSC2			
	I0.6	I0.7	I1.0	I1.1	I1.2	I1.3	I1.4	I1.5
0	Reloj	–	–	–	Reloj	–	–	–
1	Reloj	–	Puesta a 0	–	Reloj	–	Puesta a 0	–
2	Reloj	–	Puesta a 0	Arranque	Reloj	–	Puesta a 0	Arranque
3	Reloj	Sentido	–	–	Reloj	Sentido	–	–
4	Reloj	Sentido	Puesta a 0	–	Reloj	Sentido	Puesta a 0	–
5	Reloj	Sentido	Puesta a 0	Arranque	Reloj	Sentido	Puesta a 0	Arranque
6	Reloj adelante	Reloj atrás	–	–	Reloj adelante	Reloj atrás	–	–
7	Reloj adelante	Reloj atrás	Puesta a 0	–	Reloj adelante	Reloj atrás	Puesta a 0	–
8	Reloj adelante	Reloj atrás	Puesta a 0	Arranque	Reloj adelante	Reloj atrás	Puesta a 0	Arranque
9	Fase A	Fase B	–	–	Fase A	Fase B	–	–
10	Fase A	Fase B	Puesta a 0	–	Fase A	Fase B	Puesta a 0	–
11	Fase A	Fase B	Puesta a 0	Arranque	Fase A	Fase B	Puesta a 0	Arranque

Cada contador dispone de entradas que asisten funciones tales como relojes, control del sentido, puesta a 0 y arranque. Los contadores A/B permiten elegir una velocidad simple o cuádruple para el conteo. HSC1 y HSC2 trabajan de forma completamente independiente y sin afectar a otras funciones rápidas. Ambos contadores funcionan a velocidades máximas sin interferirse mutuamente.

Para obtener más información acerca de cómo utilizar los contadores rápidos, consulte el apartado 9.5 (Operaciones con contadores rápidos SIMATIC) en el capítulo 9.

## Salidas de impulsos rápidos

Las CPUs S7-200 asisten salidas de impulsos rápidos. Q0.0 y Q0.1 pueden generar trenes de impulsos rápidos (PTO) o controlar la modulación del ancho de impulsos (PWM).

- La función PTO ofrece una salida en cuadratura (con un ancho de impulsos de 50%) para un número determinado de impulsos y un tiempo de ciclo determinado. El número de impulsos puede estar comprendido entre 1 y 4.294.967.295. El tiempo de ciclo se puede indicar en microsegundos o milisegundos, bien sea de 50  $\mu$ s a 65.535  $\mu$ s, o bien de 2 ms a 65.535 ms. Un número impar de microsegundos o milisegundos (p.ej. 75 ms) causa una distorsión del factor de trabajo relativo. La función Tren de impulsos (PTO) se puede programar para producir un tren de impulsos, o bien un perfil de impulsos compuesto por varios trenes de impulsos. En éste último caso, la función PTO se puede programar para controlar un motor paso a paso utilizando una secuencia simple de aceleración, funcionamiento y desaceleración, o bien secuencias más complicadas. El perfil de impulsos puede comprender hasta 255 segmentos, correspondiendo un segmento a la función de aceleración, funcionamiento o desaceleración.
- La función PWM ofrece un tiempo de ciclo fijo con una salida de ancho de impulsos variable. El tiempo de ciclo y el ancho de impulsos pueden indicarse en incrementos de microsegundos o milisegundos. El tiempo de ciclo puede estar comprendido entre 50  $\mu$ s y 65.535  $\mu$ s, o bien entre 2 ms y 65.535 ms. El tiempo del ancho de impulsos puede estar comprendido entre 0  $\mu$ s y 65.535  $\mu$ s, o bien entre 0 ms y 65.535 ms. Si el ancho de impulsos y el tiempo de ciclo son iguales, entonces el factor de trabajo relativo (relación impulso-pausa) será de 100% y la salida se activará continuamente. Si el ancho de impulsos es cero, el factor de trabajo relativo (relación impulso-pausa) será de 0% y se desactivará la salida.

Para obtener más información acerca de la salida de impulsos, consulte el apartado 9.5 (Operaciones con contadores rápidos SIMATIC) en el capítulo 9.



## 6.7 Potenciómetros analógicos

Los potenciómetros analógicos están ubicados debajo de la tapa de acceso frontal de la CPU. Dichos potenciómetros permiten incrementar o decrementar valores almacenados en los bytes de marcas especiales SMB28 y SMB29. El programa puede utilizar estos valores de sólo lectura para diversas funciones, p.ej. para actualizar el valor actual de un temporizador o de un contador, para introducir o modificar los valores estándar, o bien, para ajustar límites.

SMB28 almacena el valor digital que representa la posición del potenciómetro analógico 0. SMB29 almacena el valor digital que representa la posición del potenciómetro analógico 1. El potenciómetro analógico tiene un margen nominal comprendido entre 0 y 255, así como una capacidad de repetición de  $\pm 2$  contajes.

Utilice un destornillador pequeño para ajustar los potenciómetros. Gire el potenciómetro hacia la derecha para incrementar el valor, o bien hacia la izquierda para decrementarlo. La figura 6-10 muestra un programa de ejemplo donde se utiliza el potenciómetro analógico.

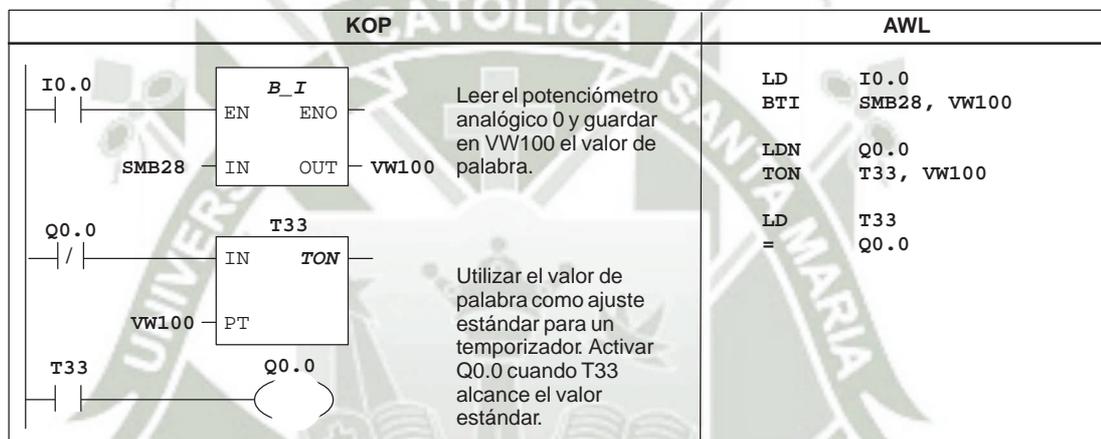


Figura 6-10 Ejemplo de un potenciómetro analógico



## Configurar el hardware para la comunicación en redes

# 7

Este capítulo describe las comunicaciones utilizando la versión 3.0 de STEP 7-Micro/WIN 32. Las versiones anteriores del software ofrecían otras funciones. También se explica cómo configurar el hardware y cómo instalar una red de comunicaciones S7-200.

### Índice del capítulo

Apartado	Descripción	Página
7.1	Opciones de comunicación	7-2
7.2	Instalar y desinstalar interfaces de comunicación	7-7
7.3	Seleccionar y cambiar parámetros	7-9
7.4	Comunicación con módems	7-16
7.5	Redes y protocolos	7-27
7.6	Componentes de redes	7-31
7.7	Utilizar el cable PC/PPI con otros dispositivos y en modo Freeport	7-35
7.8	Rendimiento de la red	7-41

## 7.1 Opciones de comunicación

Las CPUs S7-200 se pueden disponer en diversas configuraciones para asistir la comunicación en redes. El software STEP 7-Micro/WIN 32 se puede instalar en un PC dotado con el sistema operativo Windows 95, Windows 98 o Windows NT, o bien, en una unidad de programación SIMATIC (p.ej. la PG 740). El PC o la PG se pueden utilizar como unidades maestras en cualquiera de las siguientes configuraciones:

- Monomaestro: Un sólo maestro se conecta a uno o varios esclavos (v. fig. 7-1).
- Multimaestro: Un sólo maestro se conecta a uno o varios esclavos y a uno o más maestros (v. fig. 7-2).
- Para usuarios de módems de 11 bits: Un sólo maestro se conecta a uno o varios esclavos. El maestro se conecta a través de módems de 11 bits a una CPU S7-200 esclava o a una red de CPUs S7-200 esclavas.
- Para usuarios de módems de 10 bits: Un sólo maestro se conecta a través de un módem de 10 bits a una CPU S7-200 esclava.

Las figuras 7-1 y 7-2 muestran una configuración con un PC conectado a varias CPUs S7-200. STEP 7-Micro/WIN 32 se ha diseñado para comunicarse con una sola CPU S7-200. No obstante, se puede acceder a cualquier CPU que intervenga en la red. Las CPUs pueden ser tanto maestras como esclavas. El TD 200 es una unidad maestra. Para obtener más información sobre la comunicación en redes, consulte el apartado 7.5.

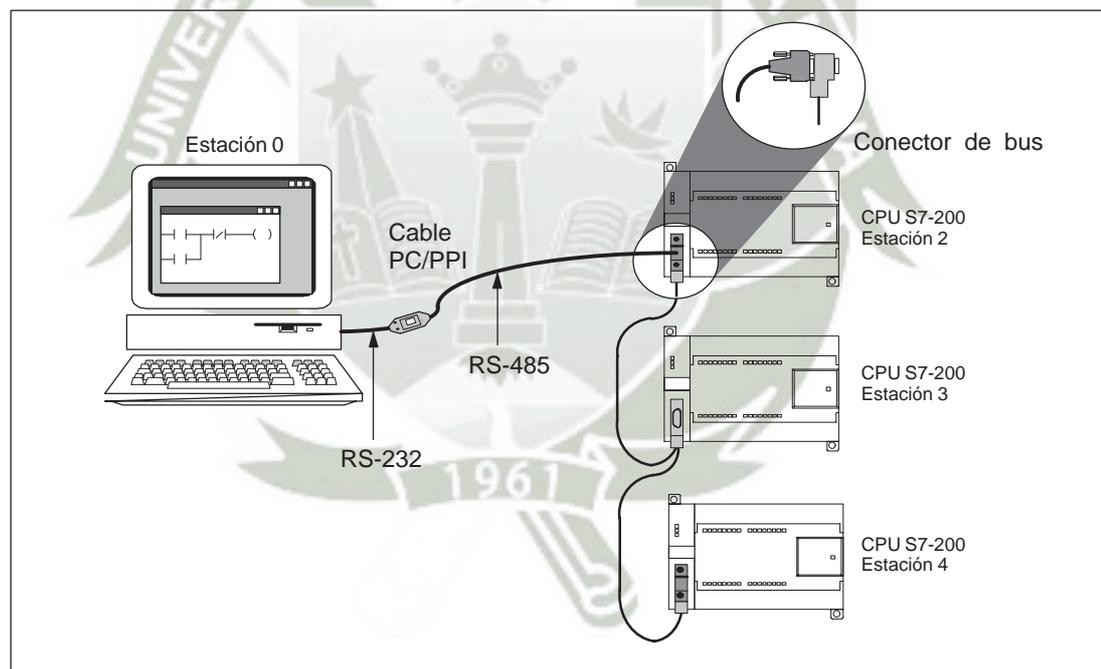


Figura 7-1 Utilizar un cable PC/PPI para la comunicación con varias CPUs S7-200

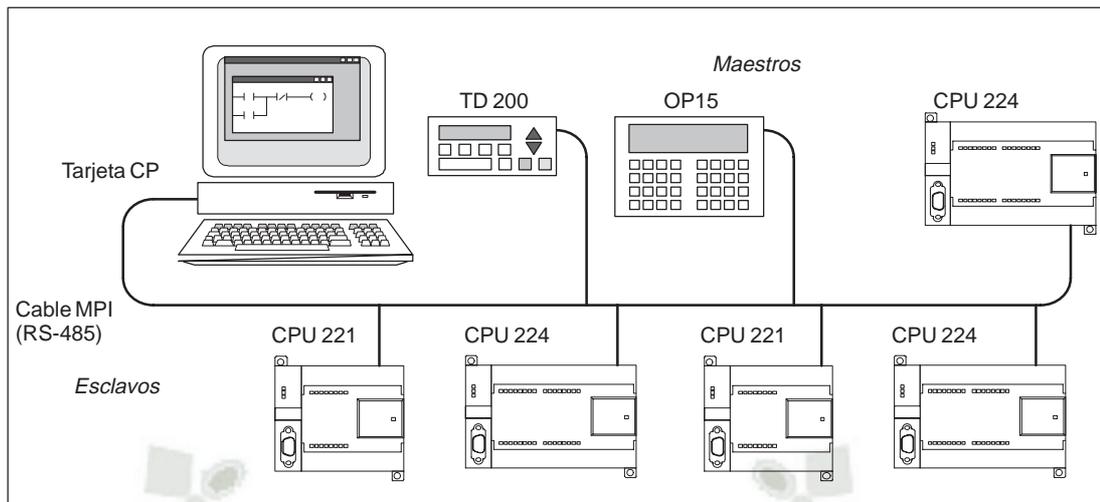


Figura 7-2 Ejemplo de una tarjeta CP con maestros y esclavos

### Cómo elegir la configuración de comunicación

La tabla 7-1 muestra las configuraciones de hardware posibles y las velocidades de transferencia que asiste STEP 7-Micro/WIN 32.

Tabla 7-1 Configuraciones de hardware asistidas por STEP 7-Micro/WIN 32

Hardware asistido	Tipo de entrada	Velocidad de transferencia asistida	Comentario
Cable PC/PPI	Conector de cable al puerto COM del PC	9,6 kbit/s 19,2 kbit/s	Asiste el protocolo PPI.
CP 5511	Tipo II, tarjeta PCMCIA		Asiste los protocolos PPI, MPI y PROFIBUS para PCs portátiles formato notebook.
CP 5611	Tarjeta PCI (versión 3 o superior)	9,6 kbit/s 19,2 kbit/s 187,5 kbit/s	Asiste los protocolos PPI, MPI y PROFIBUS para PCs.
MPI	Tarjeta PC ISA integrada en la PG		

## Comunicación utilizando tarjetas CP o MPI

Siemens ofrece diversas tarjetas de interface que se pueden incorporar en un PC o en una PG (unidad de programación) SIMATIC. Las tarjetas permiten que el PC o la PG actúen de estaciones maestras en la red. Dichas tarjetas contienen componentes de hardware especiales para asistir al PC o a la PG en la gestión de la red multimaestro, soportando diferentes protocolos y diversas velocidades de transferencia (v. tabla 7-1).

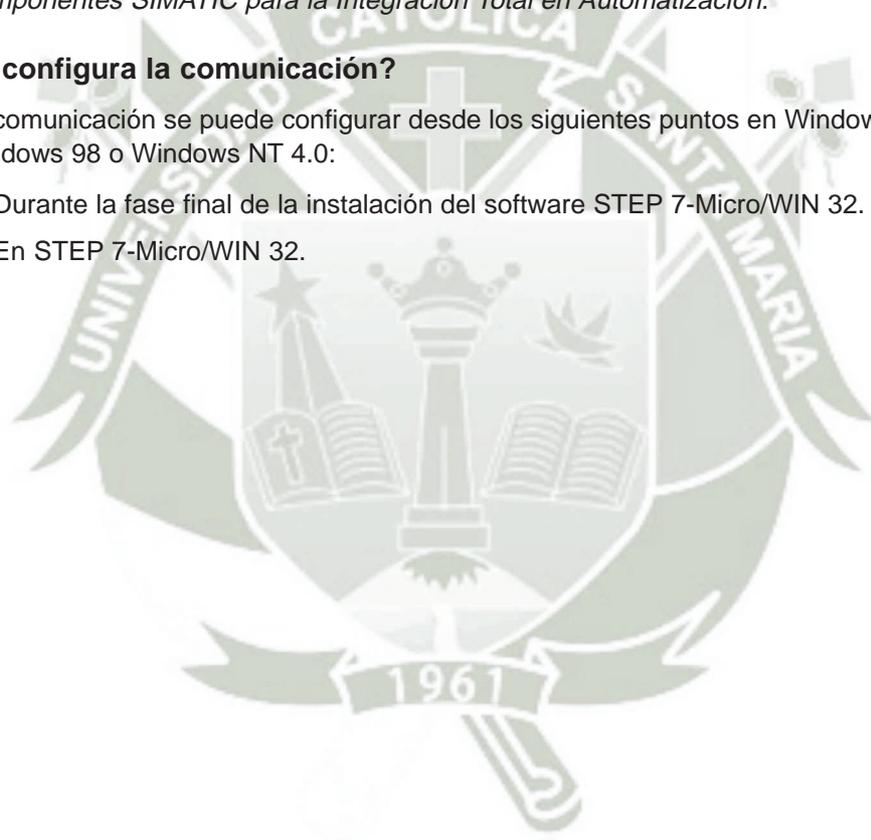
La tarjeta y el protocolo específicos se ajustan en el cuadro de diálogo "Ajustar interface PG/PC" accesible desde STEP 7-Micro/WIN 32 (v. apt. 7.3). Utilizando Windows 95, Windows 98 o Windows NT se puede seleccionar cualquier protocolo (PPI, MPI o PROFIBUS) a utilizar con cualquiera de las tarjetas de red.

Cada tarjeta incorpora un puerto RS-485 sencillo para la conexión a la red PROFIBUS. La tarjeta CP 5511 PCMCIA dispone de un adaptador que incorpora el conector D subminiatura de 9 pines. Uno de los extremos del cable MPI se conecta al puerto RS-485 de la tarjeta y el otro, al conector del puerto de programación (v. fig. 7-2). Para obtener informaciones más detalladas acerca de los procesadores de comunicación, consulte el Catálogo ST 70 1997: *Componentes SIMATIC para la Integración Total en Automatización*.

## ¿Dónde se configura la comunicación?

La comunicación se puede configurar desde los siguientes puntos en Windows 95, Windows 98 o Windows NT 4.0:

- Durante la fase final de la instalación del software STEP 7-Micro/WIN 32.
- En STEP 7-Micro/WIN 32.



## Cómo configurar la comunicación en STEP 7-Micro/WIN 32

En el cuadro de diálogo "Configurar comunicación" de STEP 7-Micro/WIN 32 se pueden configurar los parámetros de comunicación. Para acceder a dicho cuadro puede optar por uno de los siguientes métodos:

- Elija el comando de menú **Ver > Comunicación**.
- Haga clic en el icono "Comunicación" en la pantalla de STEP 7-Micro/WIN 32 (v. fig. 7-3).

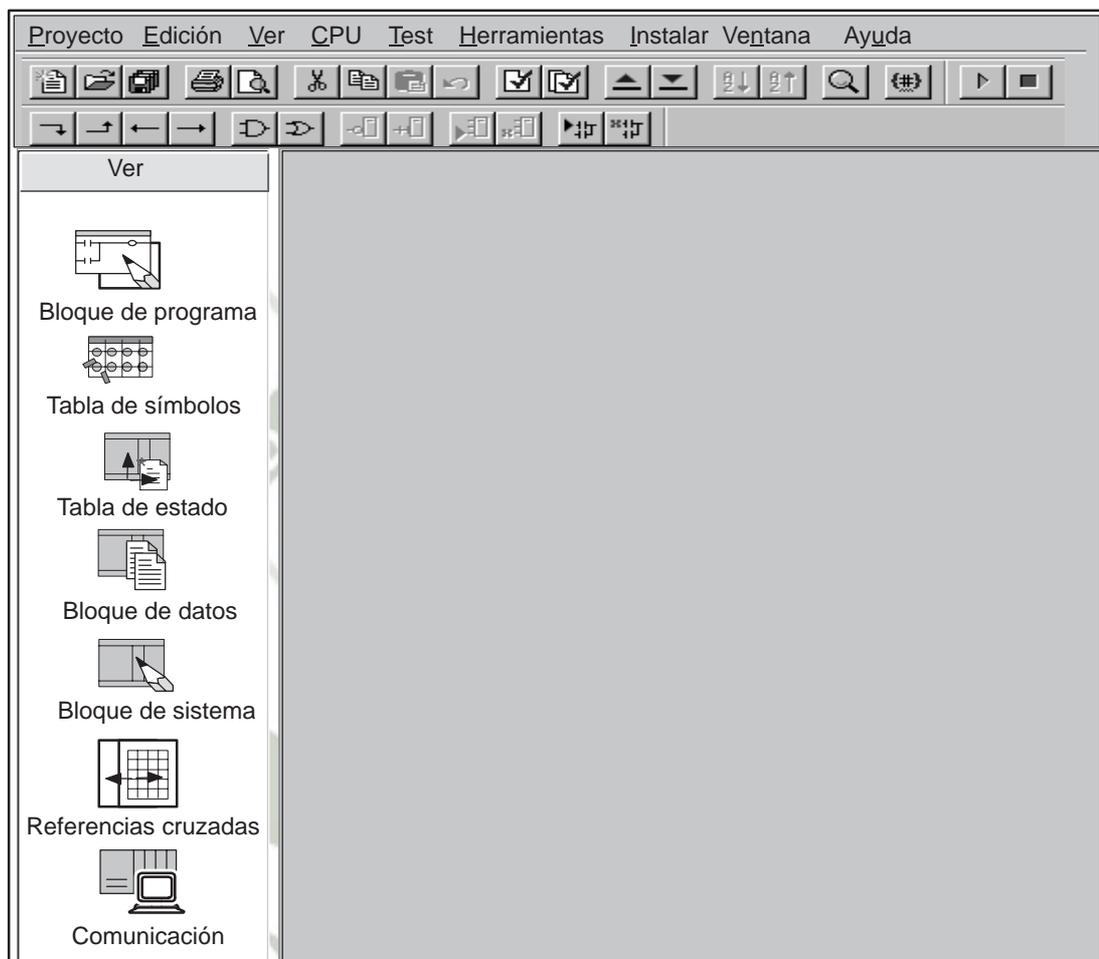


Figura 7-3 Menú Ver de STEP 7-Micro/WIN 32

En el cuadro de diálogo "Configurar comunicación", haga doble clic en el icono superior que aparece en el lado derecho. Aparecerá el cuadro de diálogo "Ajustar interface PG/PC" (v. fig. 7-4).

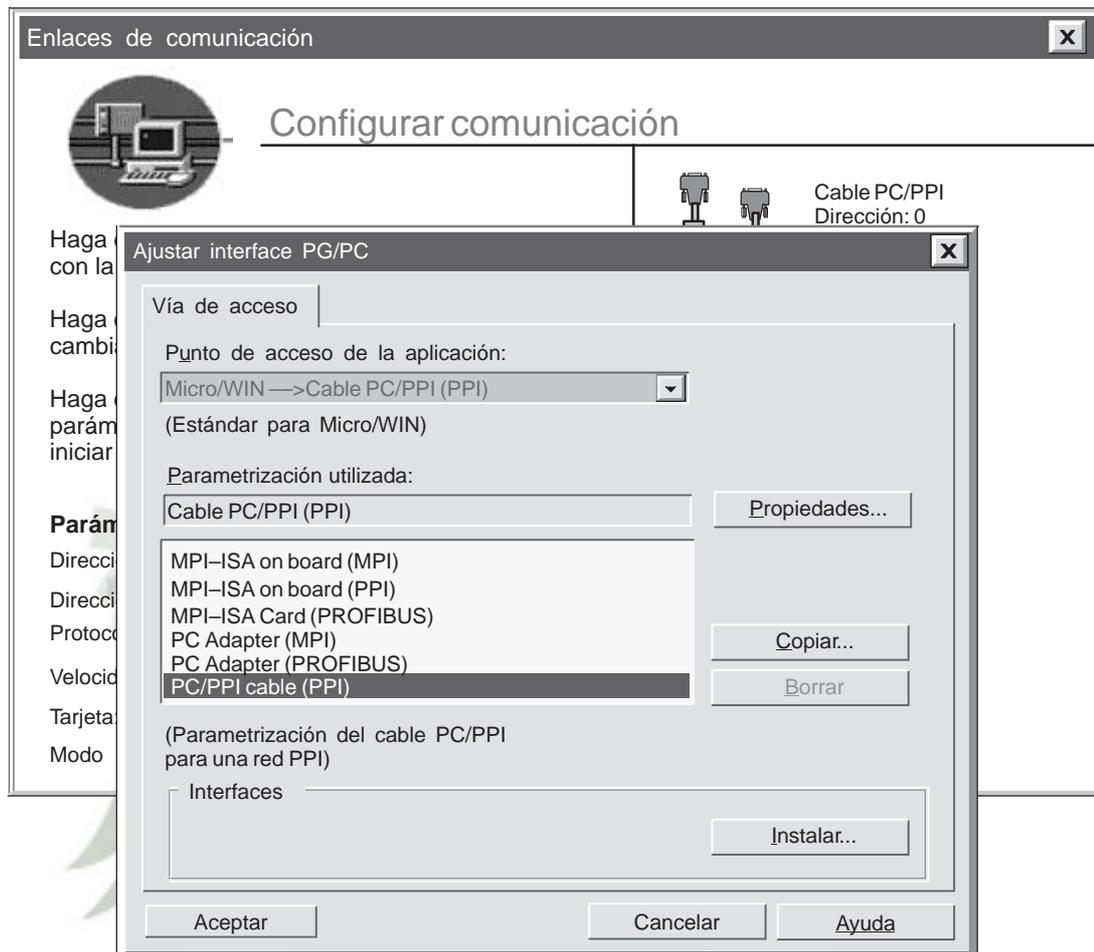


Figura 7-4 Ajustes en el cuadro de diálogo "Interface PG/PC"

## 7.2 Instalar y desinstalar interfaces de comunicación

Los componentes de hardware de comunicación se pueden instalar o desinstalar en el cuadro de diálogo "Instalar/desinstalar interfaces" que muestra la figura 7-5. En el lado izquierdo de dicho cuadro de diálogo figura una lista de los componentes que no se han instalado todavía. En el lado derecho aparece una lista de los componentes instalados actualmente. En caso de utilizar el sistema operativo Windows NT 4.0, el botón "Recursos" se visualiza debajo de ésta última lista.

### Instalar componentes de hardware:

Para instalar componentes de hardware, siga los siguientes pasos:

1. En el cuadro de diálogo "Ajustar interface PG/PC" (v. fig. 7-4), haga clic en el botón "Instalar" para acceder al cuadro de diálogo "Instalar/desinstalar interfaces" que muestra la figura 7-5.
2. En el cuadro de lista "Selección", elija el componente de hardware que desea instalar. En la ventana inferior se visualiza una descripción del componente seleccionado.
3. Haga clic en el botón "Instalar -->".
4. Cuando termine de instalar los componentes de hardware, haga clic en el botón "Cerrar". Aparecerá el cuadro de diálogo "Ajustar interface PG/PC" y los componentes que haya seleccionado se visualizarán en el cuadro de lista "Parametrización utilizada" (v. fig. 7-4).

### Desinstalar componentes de hardware:

Para desinstalar componentes de hardware, siga los siguientes pasos:

1. En el cuadro de lista "Instalados" que aparece en el lado derecho, elija los componentes que desea desinstalar.
2. Haga clic en el botón "<-- Desinstalar".
3. Cuando termine de desinstalar los componentes de hardware, haga clic en el botón "Cerrar". Aparecerá el cuadro de diálogo "Ajustar interface PG/PC" y los componentes que haya seleccionado se visualizarán en el cuadro de lista "Parametrización utilizada" (v. fig. 7-4).



Figura 7-5 Cuadro de diálogo "Instalar/desinstalar interfaces"

## Instalación de componentes de hardware en Windows NT

La instalación de componentes de hardware en el sistema operativo Windows NT difiere un poco de la instalación en Windows 95. Aunque para ambos sistemas operativos se utilizan los mismos componentes, la instalación bajo Windows NT exige conocimientos más detallados del hardware a instalar. Windows 95 intenta instalar automáticamente los recursos del sistema, contrariamente a Windows NT que sólo proporciona los valores estándar. Dichos valores pueden o no corresponder a la configuración del hardware. No obstante, los parámetros se pueden modificar fácilmente para que concuerden con los ajustes que exige el sistema.

Tras haber instalado un componente de hardware, selecciónelo en el cuadro de lista "Instalados" y haga clic en el botón "Recursos" (v. fig. 7-5). Aparecerá el cuadro de diálogo "Recursos" (v. fig. 7-6). Allí se pueden modificar los ajustes del componente de hardware que se ha instalado. Si dicho botón aparece atenuado (gris), no es necesario tomar más medidas al respecto.

En caso necesario, consulte el manual del componente de hardware en cuestión para determinar los valores de los parámetros que aparecen en el cuadro de diálogo, dependiendo de los ajustes del hardware. Es posible que deba realizar varios intentos a la hora de elegir la interrupción adecuada para establecer la comunicación.

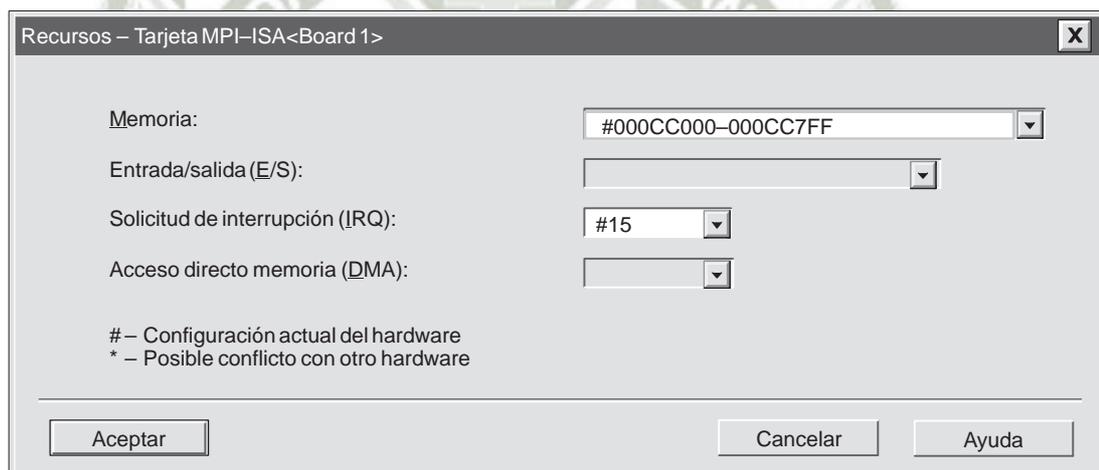


Figura 7-6 Cuadro de diálogo "Recursos" para Windows NT

### Nota

Si utiliza Windows NT y un cable PC/PPI, en la red no podrá intervenir ningún otro maestro.

## 7.3 Seleccionar y cambiar parámetros

### Seleccionar y configurar la parametrización correcta

Tras acceder al cuadro de diálogo "Ajustar interface PG/PC", verifique si "Micro/WIN" aparece en el cuadro de lista "Punto de acceso de la aplicación" (v. fig. 7-4). El cuadro de diálogo "Ajustar interface PG/PC" se utiliza en diferentes aplicaciones, como p.ej. STEP 7 y WinCC. Por tanto, puede que haya que indicar al programa para qué aplicación desea configurar los parámetros.

Tras seleccionar "Micro/WIN" e instalar el hardware, se deben ajustar las propiedades actuales para la comunicación con éste último. Primero que todo es necesario determinar el protocolo a utilizar en la red. Es recomendable utilizar el protocolo PPI para todas las CPUs.

Tras seleccionar el protocolo que desea utilizar, puede elegir la parametrización correcta en el cuadro de lista "Parametrización utilizada" del cuadro de diálogo "Ajustar interface PG/PC". Allí se indican los componentes de hardware instalados, junto con el tipo de protocolo (entre paréntesis). Por ejemplo, una configuración sencilla puede exigir que se utilice un cable PC/PPI para la comunicación con una CPU 222. En este caso se debe seleccionar "Cable PC/PPI(PPI)".

Tras haber elegido la parametrización correcta, debe ajustar los distintos parámetros para la configuración actual. Haga clic en el botón "Propiedades..." del cuadro de diálogo "Ajustar interface PG/PC". Esta acción le conduce a uno de varios cuadros de diálogo posibles, dependiendo de la parametrización que haya seleccionado (v. fig. 7-7). En los apartados siguientes se describe detalladamente cada uno de ellos.

En resumen, para seleccionar la parametrización de un interface, siga los siguientes pasos:

1. En la ficha "Vía de acceso" del cuadro de diálogo "Ajustar interface PG/PC" (v. fig. 7-4), seleccione "Micro/WIN" en el cuadro de lista "Punto de acceso de la aplicación".
2. Verifique que los componentes de hardware estén instalados (v. apt. 7.2).
3. Determine el protocolo que desea utilizar. Es recomendable utilizar el protocolo PPI para todas las CPUs.
4. En el cuadro de lista "Parametrización utilizada" del cuadro de diálogo "Ajustar interface PG/PC", elija la configuración correcta.
5. Haga clic en el botón "Propiedades..." en el cuadro de diálogo "Ajustar interface PG/PC".

Allí puede efectuar los ajustes conforme a la parametrización elegida.

### Ajustar los parámetros del cable PC/PPI (PPI)

Aquí se explica cómo ajustar los parámetros PPI en los sistemas operativos Windows 95, Windows 98 o Windows NT 4.0 para el cable PC/PPI.

Si en el cuadro de diálogo "Ajustar interface PG/PC" está seleccionado el cable PC/PPI (PPI) y se hace clic en el botón "Propiedades...", aparecerá la ficha de propiedades del mismo (v. fig. 7-7).

STEP 7-Micro/WIN 32 utiliza por defecto un protocolo PPI multimaestro para comunicarse con las CPUs S7-200. Dicho protocolo le permite a STEP 7-Micro/WIN 32 coexistir con otros maestros (TDs 200 y paneles de operador) en una red. Este modo se habilita marcando la casilla de verificación "Red multimaestro" en el cuadro de propiedades del cable PC/PPI accesible desde el cuadro de diálogo "Ajustar interface PG/PC". Windows NT 4.0 no soporta la opción multimaestro.

STEP 7-Micro/WIN 32 también asiste el protocolo PPI con un maestro único. Si se utiliza dicho protocolo, STEP 7-Micro/WIN 32 supone que es el único maestro en la red, por lo que no coopera para compartir la red con otros maestros. Este protocolo sólo se deberá utilizar al transmitir datos vía módems o en redes con muchas interferencias. El protocolo con un maestro único se selecciona borrando la marca de verificación de la casilla "Red multimaestro" en el cuadro de propiedades del cable PC/PPI accesible desde el cuadro de diálogo "Ajustar interface PG/PC".



Para ajustar los parámetros PPI, siga los siguientes pasos:

1. En el área "Propiedades del equipo" de la ficha "PPI", elija un número en el cuadro "Dirección". Dicho número indica qué dirección debe tener STEP 7-Micro/WIN 32 en la red de sistemas de automatización. El ajuste estándar para el PC en el que se está ejecutando STEP 7-Micro/WIN 32 es la dirección 0. El ajuste estándar para la primera CPU de la red es la dirección 2. Todo dispositivo (PC, CPU, etc.) que intervenga en la red debe tener una dirección unívoca. No asigne una misma dirección a varios dispositivos.
2. Elija un valor en el cuadro "Timeout". Dicho valor representa el tiempo durante el que los drivers de comunicación deben intentar establecer enlaces. El valor estándar debería ser suficiente.
3. Determine si desea que STEP 7-Micro/WIN 32 intervenga en una red multimaestro. Puede dejar marcada la casilla "Red multimaestro", a menos que esté utilizando un módem o Windows NT 4.0. En ese caso, la casilla no se puede marcar, puesto que STEP 7-Micro/WIN 32 no asiste dicha funcionalidad.
4. Ajuste la velocidad de transferencia que desea utilizar para la comunicación de STEP 7-Micro/WIN 32 en la red. El cable PPI asiste velocidades de 9,6 kbit/s y 19,2 kbit/s.
5. Elija la dirección de estación más alta. STEP 7-Micro/WIN 32 busca otros maestros en la red hasta esta dirección como máximo.

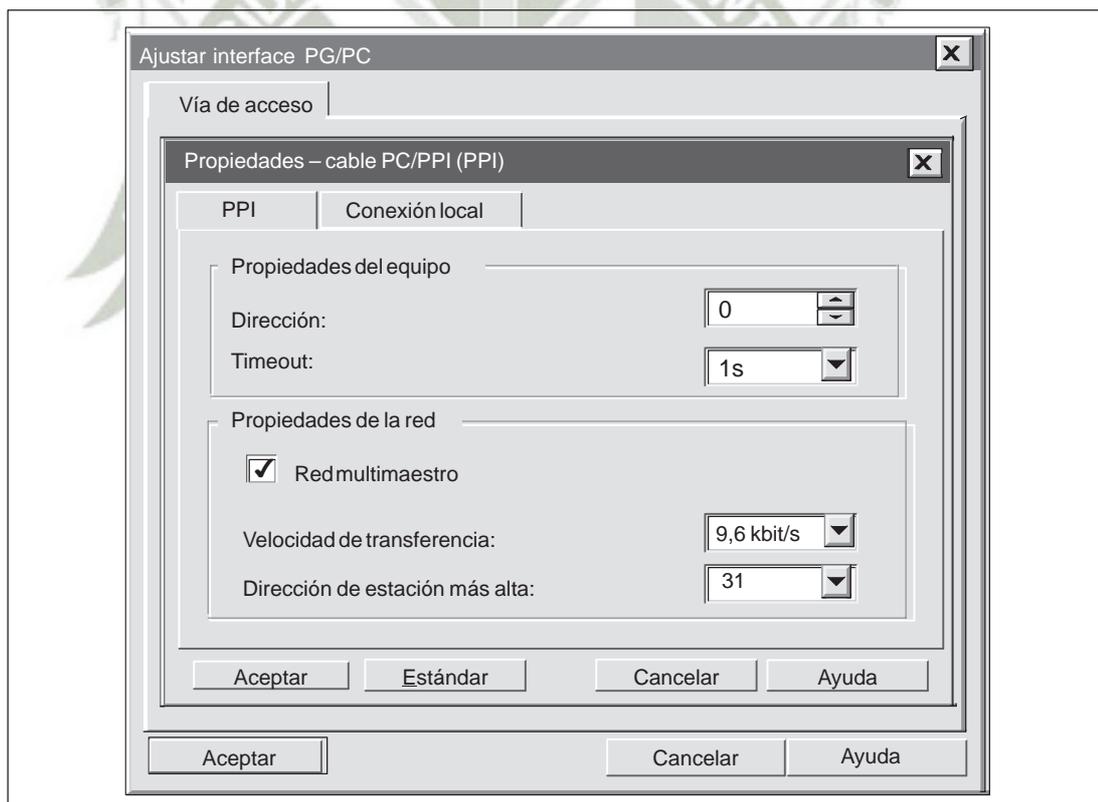


Figura 7-7 Cuadro de diálogo "Propiedades - Cable PC/PPI (PPI)", ficha "PPI"

6. Haga clic en la ficha "Conexión local" (v. fig. 7-8).
7. En la ficha "Conexión local", seleccione el puerto COM al que está conectado el cable PC/PPI. Si utiliza un módem, seleccione el puerto COM al que esté conectado el módem y marque la casilla de verificación "Utilizar módem".
8. Haga clic en el botón "Aceptar" para salir del cuadro de diálogo "Ajustar interface PG/PC".

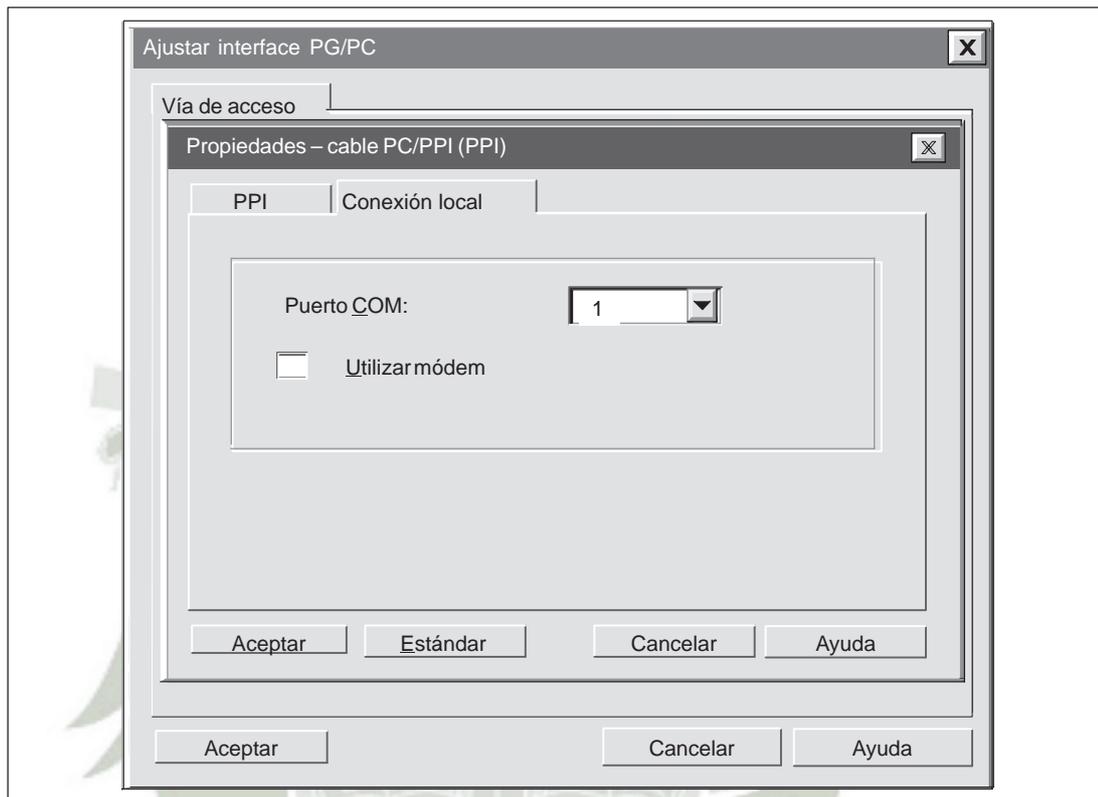


Figura 7-8 Cuadro de diálogo "Propiedades - cable PC/PPI (PPI)", ficha "Conexión local"

### Configuraciones posibles al utilizar un PC con una tarjeta MPI o un CP en una red multimaestro

Una tarjeta interface multipunto (tarjeta MPI) o un procesador de comunicaciones (CP) permiten crear numerosas configuraciones. Ambos componentes disponen de un puerto RS-485 sencillo para la conexión a la red mediante un cable MPI. Una estación en la que se ejecute el software de programación STEP 7-Micro/WIN 32 (PC con tarjeta MPI o CP, o bien una unidad de programación SIMATIC) se puede conectar a una red en la que intervengan varios maestros. (Ello es aplicable también al cable PC/PPI si se han habilitado varios maestros). Los maestros pueden ser también paneles de operador y visualizadores de textos (TDs 200). La figura 7-9 muestra una configuración con dos TDs 200 que se han integrado en la red.

**Nota**

Si se utiliza la parametrización PPI, STEP 7-Micro/WIN 32 no soportará la ejecución simultánea de dos aplicaciones diferentes en una misma tarjeta MPI o CP. Cierre la otra aplicación antes de conectar STEP 7-Micro/WIN 32 a la red a través de la tarjeta MPI o CP.

Esta configuración ofrece las siguientes posibilidades de comunicación:

- STEP 7-Micro/WIN 32 (en la estación 0) puede vigilar el estado de la estación de programación 2, mientras que los visualizadores de textos TD 200 (estaciones 5 y 1) se comunican con las CPUs 224 (estaciones 3 y 4, respectivamente).
- Ambas CPUs 224 se pueden habilitar para que envíen mensajes utilizando operaciones de red (NETR y NETW).
- La estación 3 puede leer datos de y escribir datos en las estaciones 2 (CPU 222) y 4 (CPU 224).
- La estación 4 puede leer datos de y escribir datos en las estaciones 2 (CPU 222) y 3 (CPU 224).

A una sola red se pueden conectar numerosos maestros y esclavos. No obstante, el rendimiento de la misma puede disminuir cuantas más estaciones se incorporen.

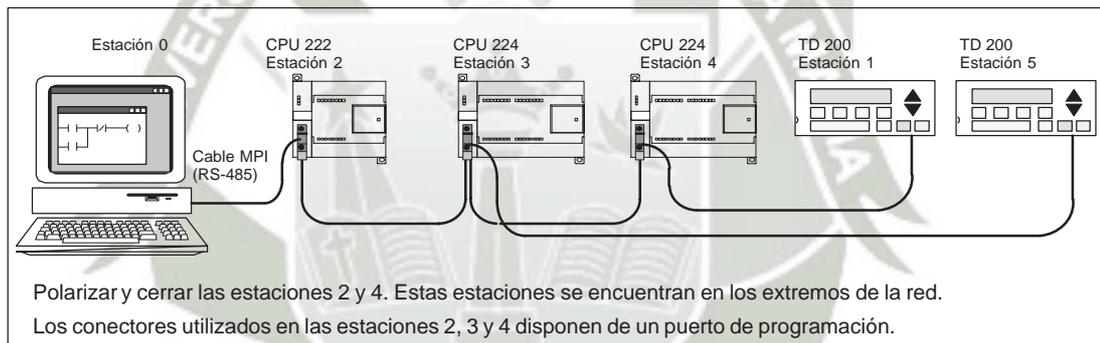


Figura 7-9 Utilizar una tarjeta MPI o un CP para la comunicación con CPUs S7-200

## Ajustar los parámetros de las tarjetas CP o MPI (PPI)

Aquí se explica cómo ajustar los parámetros PPI en los sistemas operativos Windows 95, Windows 98 o Windows NT 4.0 para los siguientes componentes de hardware:

- CP 5511
- CP 5611
- MPI

Partiendo del cuadro de diálogo "Ajustar interface PG/PC", si alguna de las tarjetas MPI o CP mencionadas se utiliza junto con el protocolo PPI y se hace clic en el botón "Propiedades...", aparecerá la ficha de propiedades de la tarjeta XXX (PPI), donde "XXX" es el tipo de tarjeta que se ha instalado (p.ej. MPI-ISA) (v. fig. 7-10).

---

### Nota

Utilice el protocolo MPI al comunicarse con una CPU S7-200 215 (por el puerto 1). Para obtener más información sobre la CPU 215 y el protocolo MPI, consulte la versión anterior del *Manual del sistema de automatización S7-200* (referencia: 6ES7-298-8FA01-8BH0).

---

Para ajustar los parámetros PPI, siga los siguientes pasos:

1. En la ficha "PPI", elija un número en el cuadro "Dirección". Dicho número indica qué dirección debe tener STEP 7-Micro/WIN 32 en la red de sistemas de automatización.
2. Elija un valor en el cuadro "Timeout". Dicho valor representa el tiempo durante el que los drivers de comunicación deben intentar establecer enlaces. El valor estándar debería ser suficiente.
3. Ajuste la velocidad de transferencia que desea utilizar para la comunicación de STEP 7-Micro/WIN 32 en la red.
4. Elija la dirección de estación más alta. STEP 7-Micro/WIN 32 busca otros maestros en la red hasta esta dirección como máximo.
5. Haga clic en el botón "Aceptar" para salir del cuadro de diálogo "Ajustar interface PG/PC".

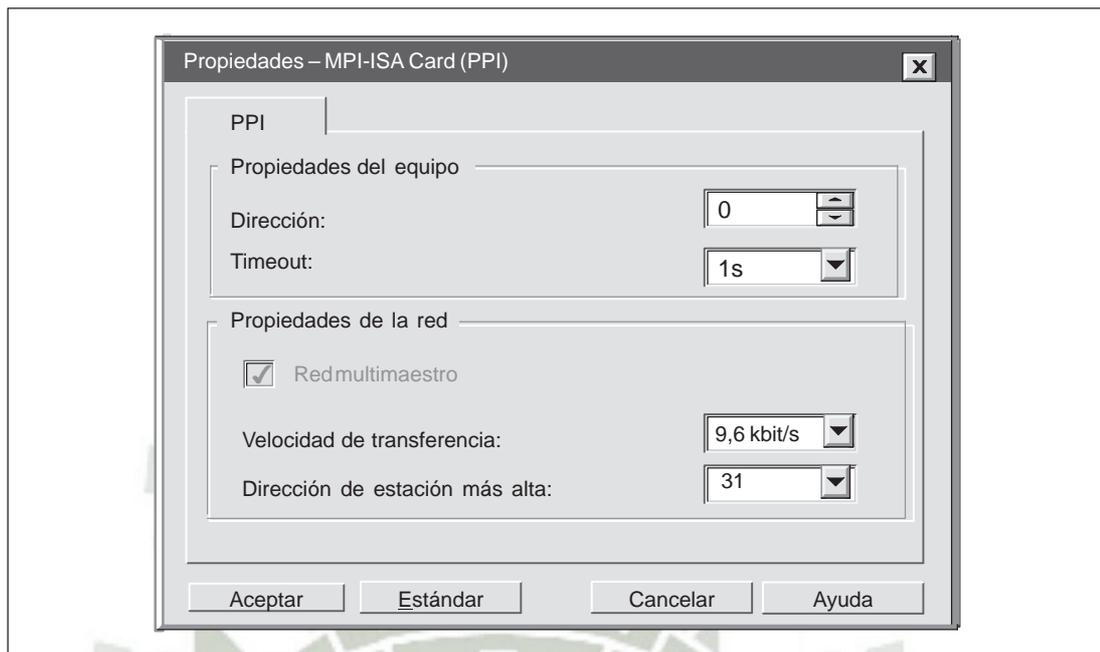


Figura 7-10 Propiedades de la tarjeta MPI-ISA (PPI)



## 7.4 Comunicación con módems

### Ajustar los parámetros de comunicación al utilizar módems

Para ajustar los parámetros de comunicación entre la unidad de programación (PG) o el PC y la CPU al utilizar módems, es preciso utilizar la parametrización del cable PC/PPI. En caso contrario no se dispondrá de la función "Configurar módem". Verifique que dicha función esté habilitada y ajuste los parámetros de configuración como se indica a continuación:

---

#### Nota

STEP 7-Micro/WIN 32 visualiza módems estándar en el cuadro de diálogo "Configurar módem". Dichos módems se han comprobado, verificándose que trabajen con STEP 7-Micro/WIN 32 conforme a los ajustes visualizados.

---

#### Configurar el módem local:

1. Elija el comando de menú **Ver > Comunicación** (o haga clic en el icono "Comunicación").  
  
En el cuadro de diálogo "Configurar comunicación", haga doble clic en el icono del cable PC/PPI. Aparecerá el cuadro de diálogo "Ajustar interface PG/PC". Salte al paso 3.  
  
Si en el cuadro de diálogo "Configurar comunicación" no se visualiza el icono del cable PC/PPI, haga doble clic en el icono del PC o en el icono superior del área derecha.
2. En el cuadro de diálogo "Ajustar interface PG/PC", elija "Cable PC/PPI(PPI)". Si dicha selección no figura en el cuadro de lista, será preciso instalarla (v. apt. 7.2).
3. Haga clic en el botón "Propiedades". Se visualizarán las propiedades del cable PC/PPI(PPI) para la CPU y el módem (v. fig. 7-8).
4. En la ventana "Propiedades – Cable PC/PPI(PPI)", haga clic en la ficha "Conexión local".
5. En el área "Puerto COM", verifique que esté marcada la casilla "Utilizar módem". Si la casilla está vacía, haga clic allí para insertar una marca de verificación (v. fig. 7-8).
6. Haga clic en el botón "Aceptar". Aparecerá el cuadro de diálogo "Ajustar interface PG/PC".
7. Haga clic en el botón "Aceptar". Aparecerá el cuadro de diálogo "Configurar comunicación". Ahora se visualizan dos iconos de módem y un icono "Conectar módem" (v. fig. 7-11).

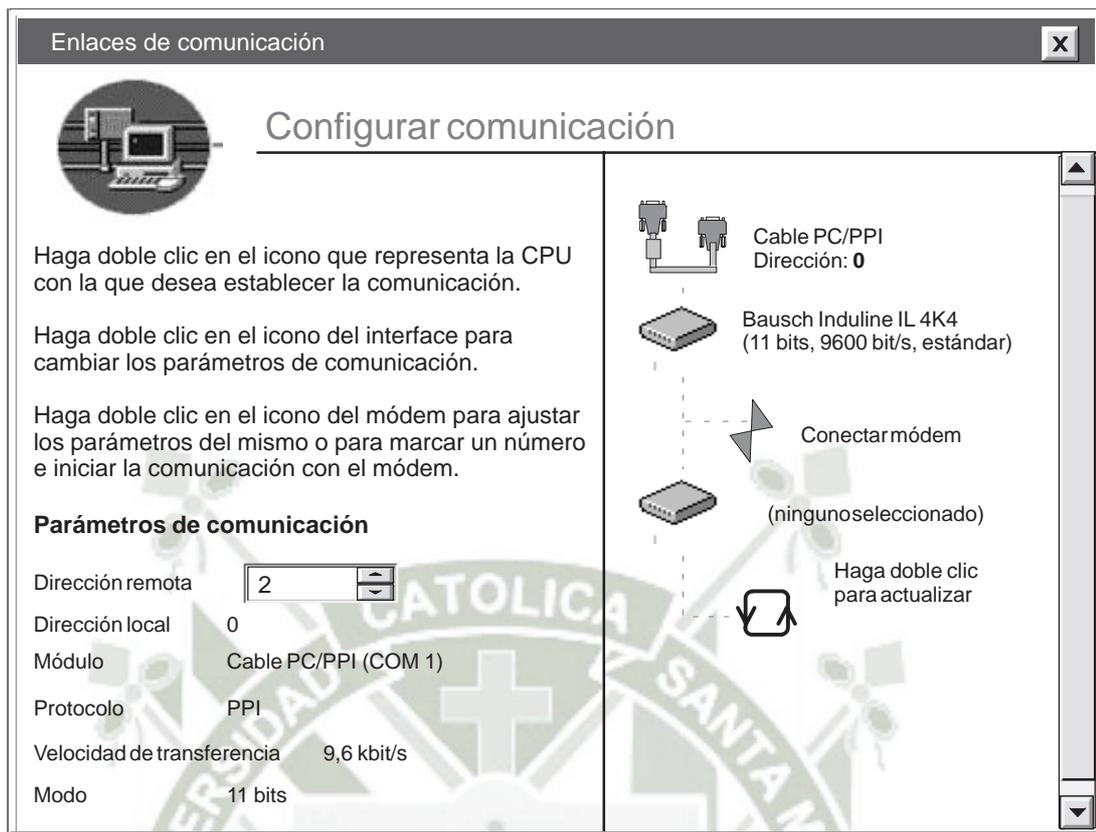


Figura 7-11 Cuadro de diálogo "Configurar comunicación"

8. En el cuadro de diálogo "Configurar comunicación", haga doble clic en el primer icono de módem. Aparecerá el cuadro de diálogo "Configurar módem" correspondiente al módem local (v. fig. 7-12).
9. En el área "Módem local", elija su tipo de módem. Si su módem no figura en la lista, haga clic en el botón "Agregar" para configurarlo. Para ello debe conocer los comandos AT del módem. Consulte a este respecto la documentación del módem.
10. En el área "Modo de comunicación", elija el modo deseado (10 u 11 bits). Éste depende de la capacidad del módem. (Los modos de comunicación de 10 bits y de 11 bits se describen más abajo). Los módems local y remoto deben tener el mismo modo de comunicación. Haga clic en el botón "Configurar".

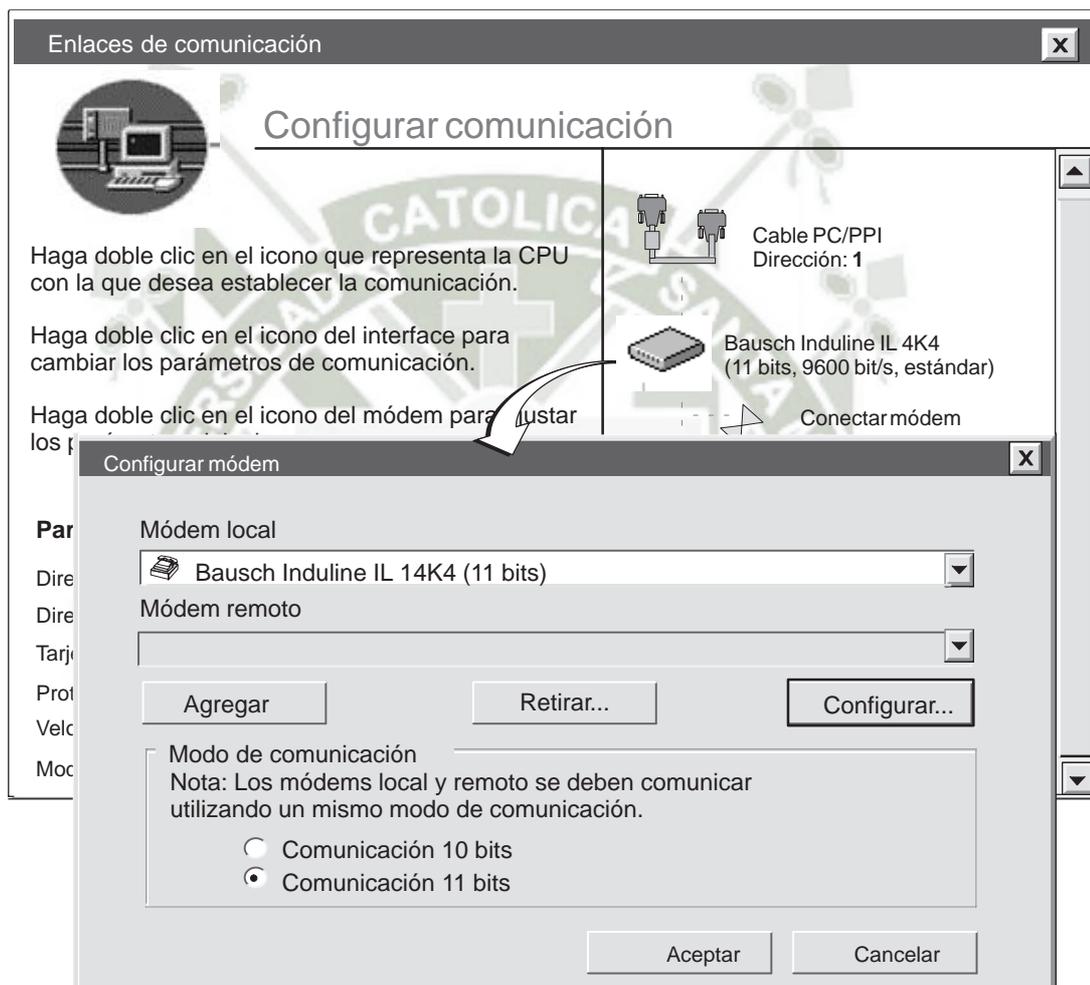
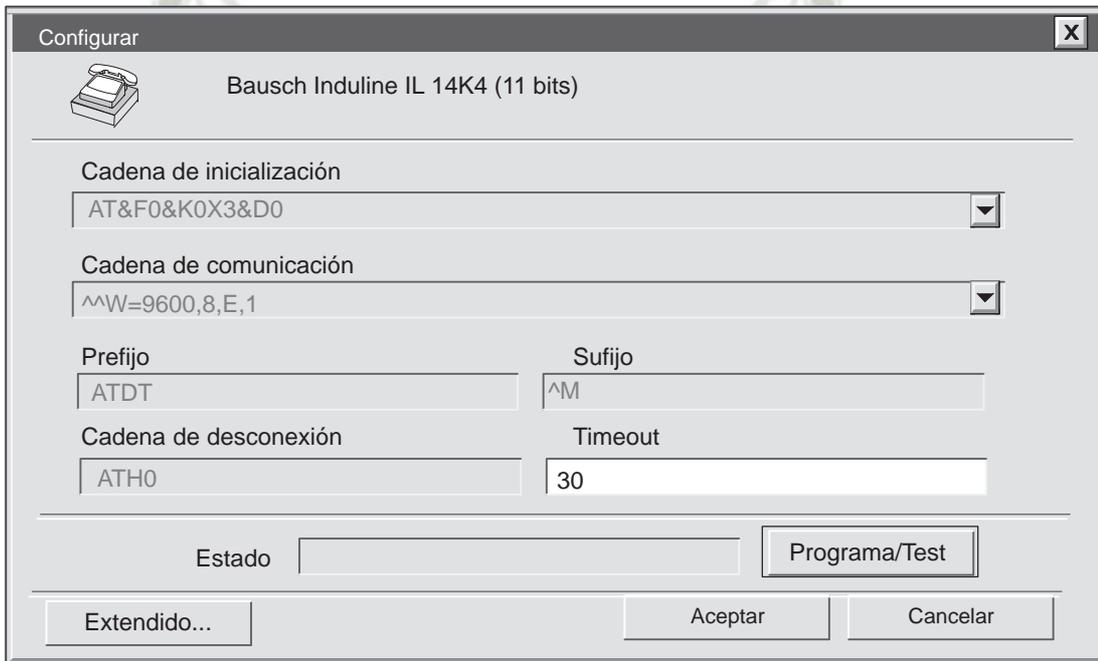


Figura 7-12 Cuadro de diálogo "Configurar módem" para el módem local

11. Aparecerá el cuadro de diálogo "Configurar" (v. fig. 7-13). Si utiliza un módem estándar, el único campo que podrá editar en este cuadro de diálogo es el cuadro "Timeout". El timeout representa el tiempo durante el cual el módem local intenta establecer la comunicación con el módem remoto. Si el tiempo indicado (en segundos) en el cuadro "Timeout" transcurre antes de establecerse la comunicación, fallará el intento de conexión. Si no está utilizando un módem estándar, deberá introducir la cadena de comando AT del módem. Consulte a este respecto la documentación del módem.
12. Si desea comprobar la configuración del módem local, haga clic en el botón "Programa/Test" mientras el módem está conectado a su equipo local (unidad de programación o PC). Así se configura el módem conforme al protocolo y a los ajustes actuales, verificando que el módem acepte los ajustes de configuración. Haga clic en "Aceptar" para regresar al cuadro de diálogo "Configurar comunicación".
13. Desconecte el módem local y conecte el módem remoto a su equipo local (unidad de programación o PC).



Configurar

Bausch Induline IL 14K4 (11 bits)

Cadena de inicialización  
AT&F0&K0X3&D0

Cadena de comunicación  
^W=9600,8,E,1

Prefijo  
ATDT

Sufijo  
^M

Cadena de desconexión  
ATH0

Timeout  
30

Estado

Programa/Test

Extendido... Aceptar Cancelar

Figura 7-13 Configuración del módem local

**Configurar el módem remoto:**

1. En el cuadro de diálogo "Configurar comunicación", haga doble clic en el segundo icono de módem (v. fig. 7-11). Aparecerá el cuadro de diálogo "Configurar módem" correspondiente al módem remoto (v. fig. 7-14).
2. En el área "Módem remoto", elija su tipo de módem. Si su módem no figura en la lista, haga clic en el botón "Agregar" para configurarlo. Para ello debe conocer los comandos AT del módem. Consulte a este respecto la documentación del módem.
3. En el área "Modo de comunicación", elija el modo deseado (10 u 11 bits). Éste depende de la capacidad del módem. (Los modos de comunicación de 10 bits y de 11 bits se describen más abajo). Los módems local y remoto deben tener el mismo modo de comunicación. Haga clic en el botón "Configurar".
4. Aparecerá el cuadro de diálogo "Configurar" (v. fig. 7-15). Si está utilizando un módem estándar no podrá editar ningún campo. En caso contrario, deberá introducir la cadena de comando AT del módem. Consulte a este respecto la documentación del módem.
5. Para comprobar la configuración del módem remoto, haga clic en el botón "Programa/Test" mientras el módem está conectado a su equipo local (unidad de programación o PC) para transferir los parámetros a un chip de memoria del módem remoto.
6. Haga clic en el botón "Aceptar". Aparecerá el cuadro de diálogo "Configurar comunicación".

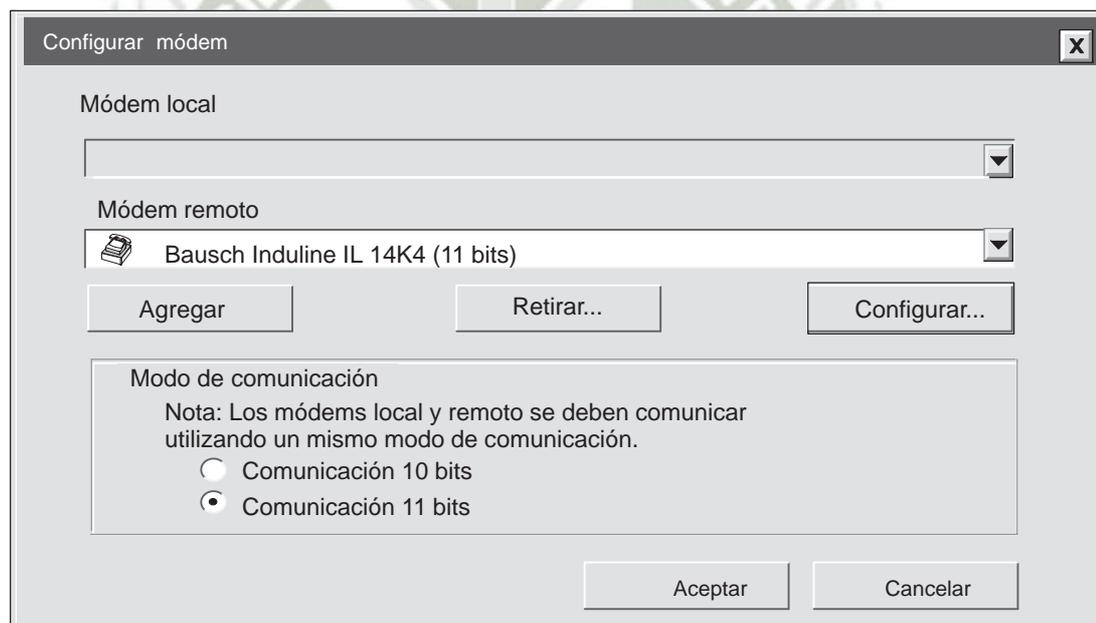


Figura 7-14 Cuadro de diálogo "Configurar módem" para el módem remoto

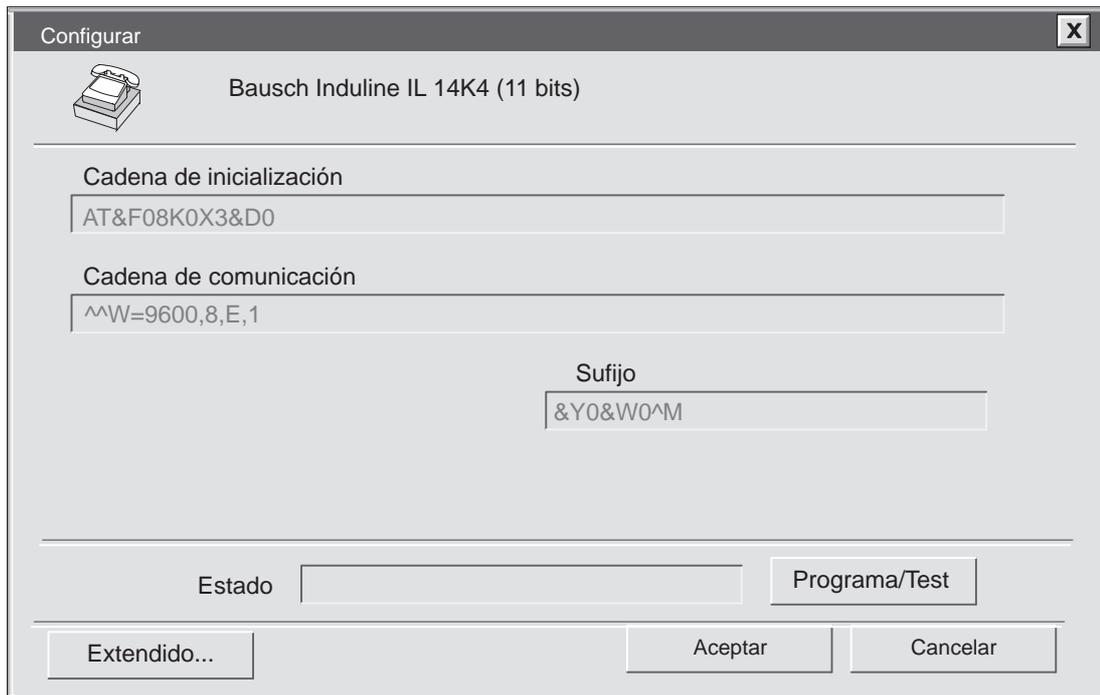


Figura 7-15 Configuración del módem remoto



7. Desconecte el módem remoto de su equipo local (unidad de programación o PC).
8. Conecte el módem remoto a su sistema de automatización S7-200.
9. Conecte el módem local a su unidad de programación o PC.

**Conectar los módems:**

1. Para conectar el módem, haga doble clic en el icono "Conectar módem" en el cuadro de diálogo "Configurar comunicación". Aparecerá el cuadro de diálogo "Marcar" (v. fig. 7-16).
2. En el cuadro de diálogo "Marcar", introduzca el número de teléfono en el cuadro correspondiente.
3. Para conectar el módem local al módem remoto, haga clic en el botón "Conectar".
4. Así se finaliza la configuración de los módems.



Figura 7-16 Conectar los módems

### Utilizar un módem de 10 bits para conectar una CPU S7-200 a un maestro STEP 7-Micro/WIN 32

Utilizando STEP 7-Micro/WIN 32 en un PC con Windows 95, Windows 98 o Windows NT, o usando una unidad de programación SIMATIC (p.ej. la PG 740) como maestro único es posible conectar sólo una CPU S7-200. Se puede utilizar un módem de 10 bits compatible con Hayes para comunicarse con una sola CPU S7-200 remota.

Para ello se necesitan los siguientes equipos:

- Una sola CPU S7-200 esclava. Las CPUs 221, 222 y 224 asisten el formato de 10 bits. Los modelos anteriores de CPUs S7-200 no asisten dicho formato.
- Un cable RS-232 para conectar el PC o la unidad de programación SIMATIC a un módem local full-dúplex de 10 bits.
- Un cable PC/PPI de 5 interruptores DIP (ajustado a la velocidad de transferencia correcta, modo de comunicación de 10 bits y modo DTE) para conectar el módem remoto a la CPU.
- Un adaptador opcional de 9 a 25 pines (si fuera necesario).

**Nota**

El cable PC/PPI de 4 interruptores DIP no asiste el formato de 10 bits.

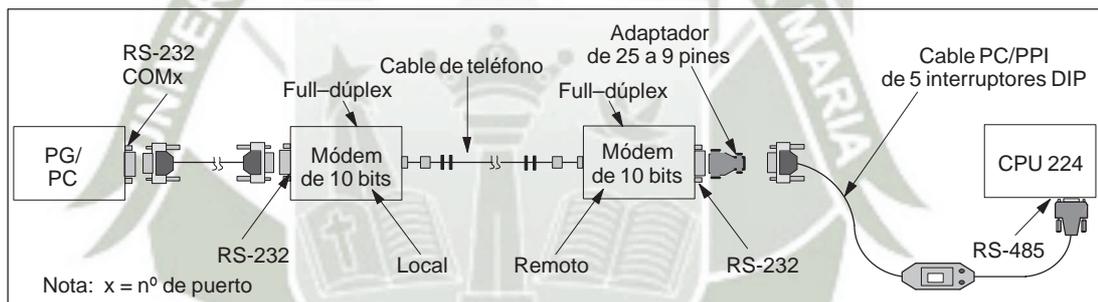


Figura 7-17 Comunicación de datos S7-200 utilizando un módem de 10 bits con un cable PC/PPI de 5 interruptores DIP

Esta configuración sólo permite utilizar un maestro y un esclavo. El sistema de automatización S7-200 exige aquí un bit de inicio, ocho bits de datos, ningún bit de paridad y un bit de parada, comunicación asíncrona y una velocidad de transferencia de 9600 bit/s. Para el módem se necesitan los ajustes que figuran en la tabla 7-2. La figura 7-18 muestra la asignación de pines para un adaptador de 25 a 9 pines.

Tabla 7-2 Ajustes necesarios para un módem de 10 bits

Módem	Formato de datos en bits	Velocidad de transferencia entre módem y PC	Velocidad de transferencia en el cable	Demás propiedades
10 bits	8 bits de datos	9600 bit/s	9600 bit/s	Ignorar señal DTR
	1 bit de inicio			Sin flujo de control de hardware
	1 bit de parada			Sin flujo de control de software
	sin paridad			

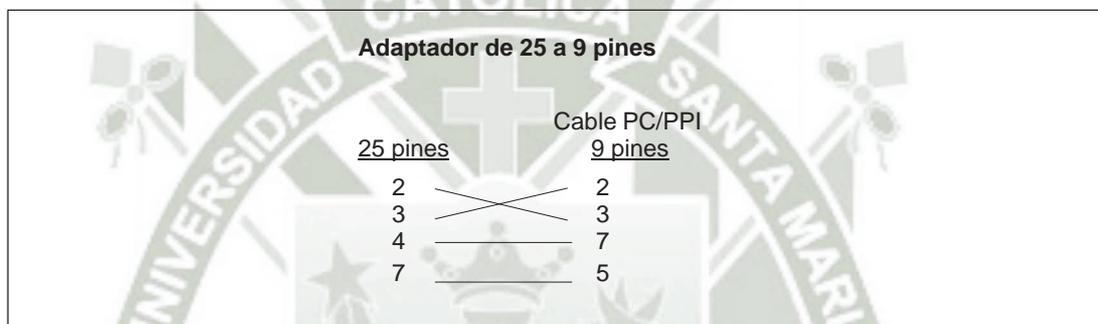


Figura 7-18 Asignación de pines para un adaptador de 25 a 9 pines

### Utilizar un módem de 11 bits para conectar una CPU S7-200 a un maestro STEP 7-Micro/WIN 32

Utilizando STEP 7-Micro/WIN 32 en un PC con Windows 95, Windows 98 o Windows NT, o usando una unidad de programación SIMATIC (p.ej. la PG 740) como maestro único es posible conectar una o más CPUs S7-200. La mayoría de los módems no asisten el protocolo de 11 bits.

Dependiendo de si desea conectar sólo una CPU S7-200 o una red de CPUs (v. fig. 7-19), necesitará los siguientes componentes:

- Un cable RS-232 estándar para conectar el PC o la unidad de programación SIMATIC a un módem local full-dúplex de 11 bits.
- Uno de los siguientes cables PC/PPI:
  - Un cable PC/PPI de 5 interruptores DIP (ajustado a la velocidad de transferencia correcta, modo de comunicación de 11 bits y modo DTE) para conectar el módem remoto a la CPU.
  - Un cable PC/PPI de 4 interruptores DIP (ajustado a la velocidad de transferencia correcta) y un adaptador de módem nulo para conectar el módem remoto a la CPU.
- Si hay varias CPUs conectadas al módem remoto se necesitará un conector de puerto de programación Siemens en una red PROFIBUS (la figura 7-23 muestra cómo polarizar y cerrar los cables de interconexión).

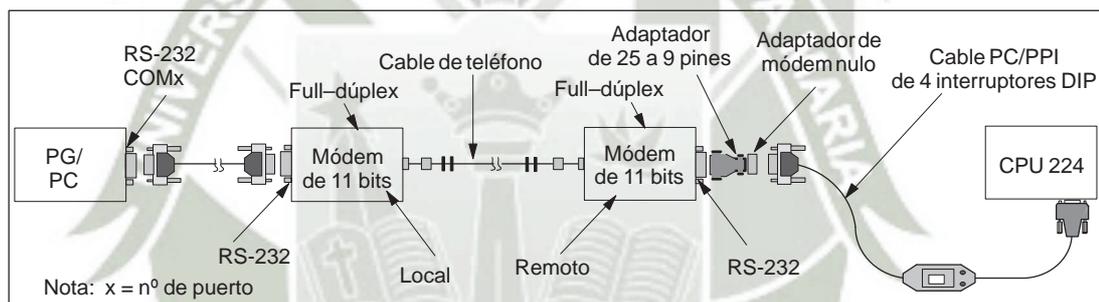


Figura 7-19 Comunicación de datos S7-200 utilizando un módem de 11 bits con un cable PC/PPI de 4 interruptores DIP

Esta configuración sólo permite utilizar un maestro y asiste únicamente el protocolo PPI. Para poder comunicarse por el interface PPI, la CPU S7-200 exige que el módem utilice una cadena de datos de 11 bits. El sistema de automatización S7-200 exige aquí un bit de inicio, ocho bits de datos, un bit de paridad par y un bit de parada, comunicación asincrónica y una velocidad de transferencia de 9600 bit/s. Numerosos módems no asisten este formato de datos. El módem exige los ajustes que figuran en la tabla 7-3.

La figura 7-20 muestra la asignación de pines para un adaptador de módem nulo y para un adaptador de 25 a 9 pines.

Tabla 7-3 Ajustes necesarios para un módem de 11 bits

Módem	Formato de datos en bits	Velocidad de transferencia entre módem y PC	Velocidad de transferencia en el cable	Demás propiedades
11 bits	8 bits de datos	9600 bit/s	9600 bit/s	Ignorar señal DTR
	1 bit de inicio			Sin flujo de control de hardware
	1 bit de parada			Sin flujo de control de software
	1 bit de paridad (par)			

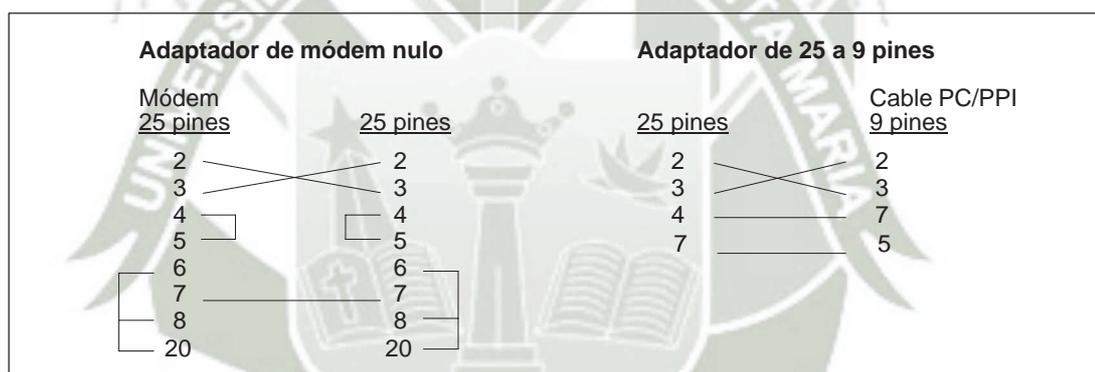


Figura 7-20 Asignación de pines para un adaptador de módem nulo y para un adaptador de 25 a 9 pines

## 7.5 Redes y protocolos

### Maestros

La figura 7-21 muestra una configuración con un PC conectado a varias CPUs S7-200. STEP 7-Micro/WIN 32 se ha diseñado para comunicarse con una sola CPU S7-200. No obstante, se puede acceder a cualquier CPU que intervenga en la red. Las CPUs de la figura 7-21 podrían actuar de esclavas o de maestras. El TD 200 es una unidad maestra.

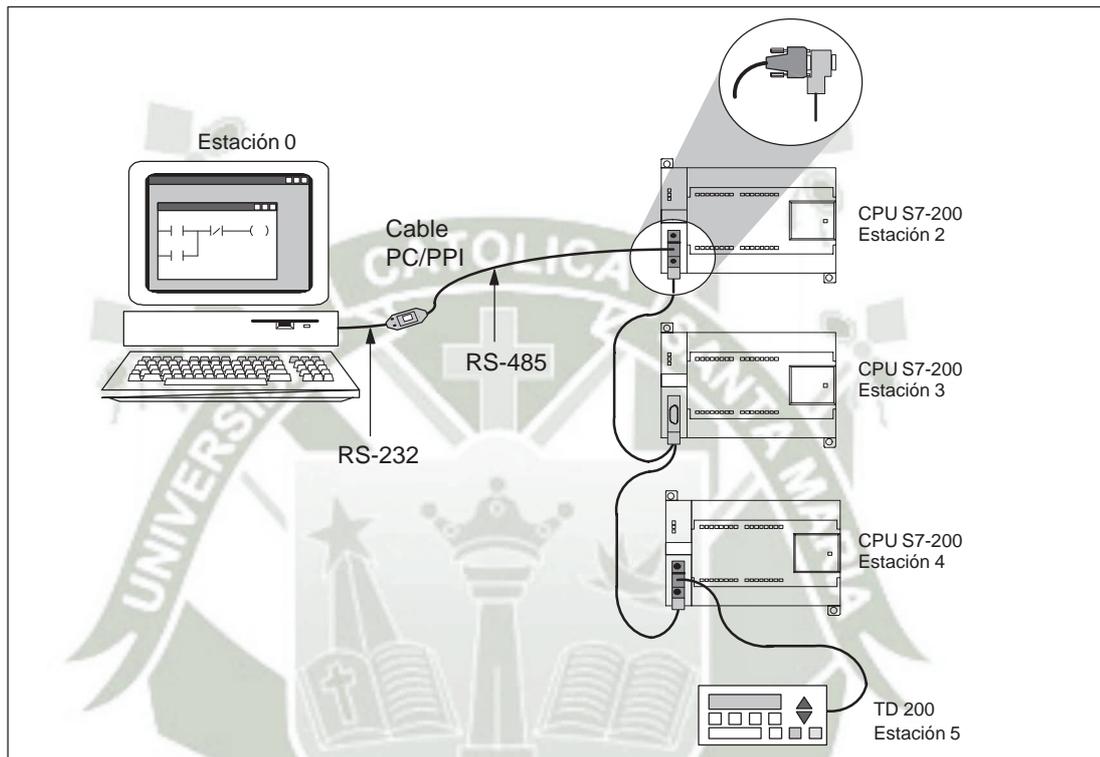


Figura 7-21 Utilizar un cable PC/PPI para la comunicación con varias CPUs S7-200 estando habilitada la opción multimaestro

## Protocolos de comunicación

Las CPUs S7-200 asisten diversos métodos de comunicación. Dependiendo de la CPU S7-200 utilizada, la red puede asistir uno o varios de los siguientes protocolos de comunicación:

- Interface punto a punto (PPI)
- Interface multipunto (MPI)
- PROFIBUS

Estos protocolos se basan en la intercomunicación de sistemas abiertos (OSI) de la arquitectura de siete capas. Los protocolos PPI y MPI se implementan en una red "token ring" (red de anillo con testigo) conforme al estándar Process Field Bus (PROFIBUS) que se describe en la norma europea EN 50170.

Se trata de protocolos asíncronos de caracteres que utilizan un bit de inicio, ocho bits de datos, un bit de paridad par y un bit de parada. Los bloques de comunicación dependen de los caracteres especiales de inicio y de parada, de las direcciones de estación de fuente y de destino, de la longitud de dichos bloques y de la suma de verificación para garantizar la integridad de los datos. Los tres protocolos se pueden utilizar simultáneamente en una red sin que interfieran entre sí, con la condición de que usen una misma velocidad de transferencia.

La red PROFIBUS utiliza el estándar RS-485 con cables de par trenzado. Ello permite interconectar hasta 32 dispositivos en un segmento de la red. Los segmentos pueden tener una longitud máxima de 1.200 m, dependiendo de la velocidad de transferencia. Es posible conectar repetidores para poder incorporar más dispositivos en la red o con objeto de utilizar cables más largos. Si se usan repetidores, las redes pueden tener una longitud de hasta 9.600 m, dependiendo de la velocidad de transferencia (v. tabla 7-6).

Los protocolos prevén dos tipos de dispositivos de red: los maestros y los esclavos. Los maestros pueden enviar una petición a otros dispositivos. En cambio, los esclavos sólo pueden responder a las peticiones de los maestros, sin poder lanzar nunca una petición por su propia cuenta.

Los protocolos asisten 127 direcciones (0 a 126) en una red. Una red puede comprender 32 maestros como máximo. Todos los dispositivos que formen parte de una red deberán tener direcciones unívocas para poder comunicarse entre sí. El ajuste estándar para las unidades de programación SIMATIC y los PCs con STEP 7-Micro/WIN 32 es la dirección "0". Los visualizadores de textos TD 200 y los paneles de operador OP3 y OP7 tienen la dirección predeterminada "1". La dirección estándar de los sistemas de automatización es "2".

## Protocolo PPI

PPI es un protocolo maestro/esclavo. Los maestros (otras CPUs, unidades de programación SIMATIC o visualizadores de textos TD 200) envían peticiones a los esclavos y éstos últimos responden. Los esclavos no inician mensajes, sino que esperan a que un maestro les envíe una petición o solicite una respuesta. Todas las CPUs S7-200 actúan de estaciones esclavas en la red.

Estando en modo RUN, algunas CPUs S7-200 pueden actuar de estaciones maestras en la red si se encuentra habilitado el modo maestro PPI en el programa de usuario. (Consulte la descripción de SMB30 en el Anexo C). Una vez habilitado el modo maestro PPI, se podrán enviar mensajes a otras CPUs, usando las operaciones Leer de la red (NETR) y Escribir en la red (NETW). En el apartado 9.16 (Operaciones de comunicación SIMATIC) encontrará una descripción de dichas operaciones. Mientras actúa de estación maestra PPI, la CPU S7-200 sigue respondiendo en calidad de esclava a las peticiones de otros maestros.

El protocolo PPI no limita la cantidad de maestros que pueden comunicarse con una CPU cualquiera que actúe de esclava, pero la red no puede comprender más de 32 maestros.

## Protocolo MPI

MPI puede ser un protocolo maestro/maestro, o bien maestro/esclavo. El funcionamiento de dicho protocolo depende de los equipos utilizados. Si el dispositivo de destino es una CPU S7-300, se establece un enlace maestro/maestro, puesto que todas las CPUs S7-300 son estaciones maestras en la red. Si es una CPU S7-200, se establece un enlace maestro/esclavo, ya que las CPUs S7-200 son unidades esclavas.

El protocolo MPI crea siempre un enlace entre los dos dispositivos intercomunicados. Un enlace es una conexión privada entre los dos dispositivos. Ningún otro maestro puede interferir en un enlace establecido entre dos dispositivos. Un maestro puede establecer un enlace para utilizarlo durante un tiempo breve o indefinido.

Puesto que los enlaces son conexiones privadas entre los dispositivos y utilizan recursos de la CPU, cada CPU puede asistir sólo una cantidad limitada de enlaces. Cada CPU asiste cuatro enlaces. Cada CPU reserva dos de sus enlaces; uno para una unidad de programación SIMATIC o un PC y el otro para paneles de operador. El enlace reservado para una unidad de programación SIMATIC o un PC garantiza que el usuario pueda conectar siempre por lo menos una unidad de programación SIMATIC o un PC a la CPU. Las CPUs también reservan un enlace para un panel de operador. Los enlaces reservados no pueden ser utilizados por otros maestros (p.ej. CPUs).

Las CPUs S7-300 y S7-400 se pueden comunicar con las CPUs S7-200 estableciendo una conexión a través de los enlaces no reservados de éstas últimas. Las CPUs S7-300 y S7-400 pueden leer y escribir datos en las CPUs S7-200, utilizando las operaciones XGET y XPUT (consulte el manual de programación de la CPU S7-300 ó S7-400, respectivamente).

## Protocolo PROFIBUS

El protocolo PROFIBUS se ha diseñado para la comunicación rápida con unidades periféricas descentralizadas (E/S remotas). Hay numerosos dispositivos PROFIBUS ofrecidos por diversos fabricantes. Dichos dispositivos abarcan desde módulos sencillos de entradas o salidas hasta controladores de motores y sistemas de automatización (autómatas programables).

Por lo general, las redes PROFIBUS tienen un maestro y varios esclavos. La configuración del maestro permite detectar los tipos de esclavos que están conectados, así como sus respectivas direcciones. El maestro inicializa la red y verifica si los esclavos coinciden con la configuración. Continuamente, el maestro escribe los datos de salida en los esclavos y lee de allí los datos de entrada. Una vez que un maestro DP haya configurado correctamente a un esclavo, éste último le pertenecerá. Si hay otro maestro en la red, tendrá apenas un acceso muy limitado a los esclavos del primer maestro.

## Protocolos definidos por el usuario (Freeport)

La comunicación Freeport es un modo de operación que permite al programa de usuario controlar el puerto de comunicación de la CPU S7-200. Con el modo Freeport se pueden implementar protocolos de comunicación definidos por el usuario para crear enlaces con numerosos dispositivos inteligentes.

El programa de usuario controla el funcionamiento del puerto de comunicación utilizando interrupciones de recepción y de transmisión, así como las operaciones Transmitir mensaje (XMT) y Recibir mensaje (RCV). En modo Freeport, el programa de usuario controla por completo el protocolo de comunicación. El modo Freeport se habilita con la marca SMB30 (puerto 0), estando activo únicamente cuando la CPU está en modo RUN. Cuando la CPU retorna a modo STOP, la comunicación Freeport se detiene y el puerto de comunicación vuelve a utilizar el protocolo PPI normal. En el apartado 9.16 (Operaciones de comunicación SIMATIC) encontrará una descripción de las operaciones Transmitir mensaje y Recibir mensaje.

## 7.6 Componentes de redes

Un sistema de automatización S7-200 se puede conectar a través del puerto de comunicación a un bus de red. A continuación se describen dicho puerto, los conectores para el bus, el cable de conexión y los repetidores utilizados para ampliar la red.

### Puerto de comunicación

Los puertos de comunicación de las CPUs S7-200 son compatibles con el estándar RS-485 mediante un conector D subminiatura de 9 pines conforme al estándar PROFIBUS definido en la norma europea EN 50170. La figura 7-22 muestra el conector que ofrece el enlace físico para el puerto de comunicación y en la tabla 7-4 figuran las asignaciones de pines para los puertos de comunicación.

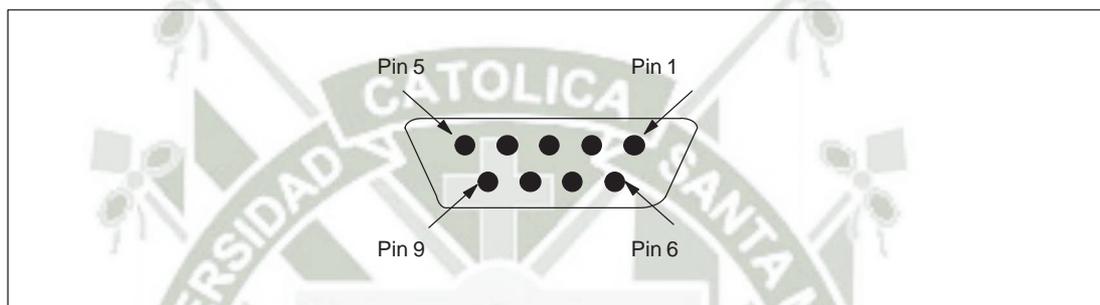


Figura 7-22 Pines del puerto de comunicación de la CPU S7-200

Tabla 7-4 Asignación de pines del puerto de comunicación de la CPU S7-200

Pin	Denominación PROFIBUS	Puerto 0
1	Blindaje	Hilo lógico
2	Hilo de retorno 24 V	Hilo lógico
3	Señal B RS-485	Señal B RS-485
4	Petición de transmitir	RTS (TTL)
5	Hilo de retorno 5 V	Hilo lógico
6	+5 V	+5 V, 100 $\Omega$ resistor en serie
7	+24 V	+24 V
8	Señal A RS-485	Señal A RS-485
9	No aplicable	Selección protocolo de 10 bits (entrada)
Carcasa del enchufe	Blindaje	Tierra

## Conectores de bus

Siemens ofrece dos tipos de conectores de bus que permiten conectar fácilmente varios dispositivos a una red. Ambos conectores poseen dos juegos de tornillos para fijar los cables de entrada y salida. Asimismo, disponen de interruptores para polarizar y cerrar la red de forma selectiva. Uno de ellos ofrece sólo un enlace a la CPU, en tanto que el otro añade un puerto de programación (v. fig. 7-23). En el Anexo E se indican los números de referencia.

El conector que provee un puerto de programación permite añadir a la red una unidad de programación SIMATIC o un panel de operador, sin perturbar ningún enlace existente. Dicho conector transmite todas las señales de la CPU a través del puerto de programación, adecuándose para conectar dispositivos alimentados por la CPU (p.ej. un TD 200 o un OP3). Los pines de alimentación del conector del puerto de comunicación se pasan por el puerto de programación.



---

### Cuidado

En caso de interconectar equipos con potenciales de referencia diferentes pueden circular corrientes indeseadas por el cable de conexión.

Dichas corrientes indeseadas pueden causar errores de comunicación o deteriorar los equipos.

Para evitar corrientes indeseadas, asegúrese de que todos los equipos que se deban conectar con un cable de comunicación compartan un circuito de referencia, o bien estén aislados unos de otros. Para obtener más información al respecto, consulte el tema "Reglas de puesta a tierra de referencia de potencial para circuitos aislados" en el apartado 2.3.

---

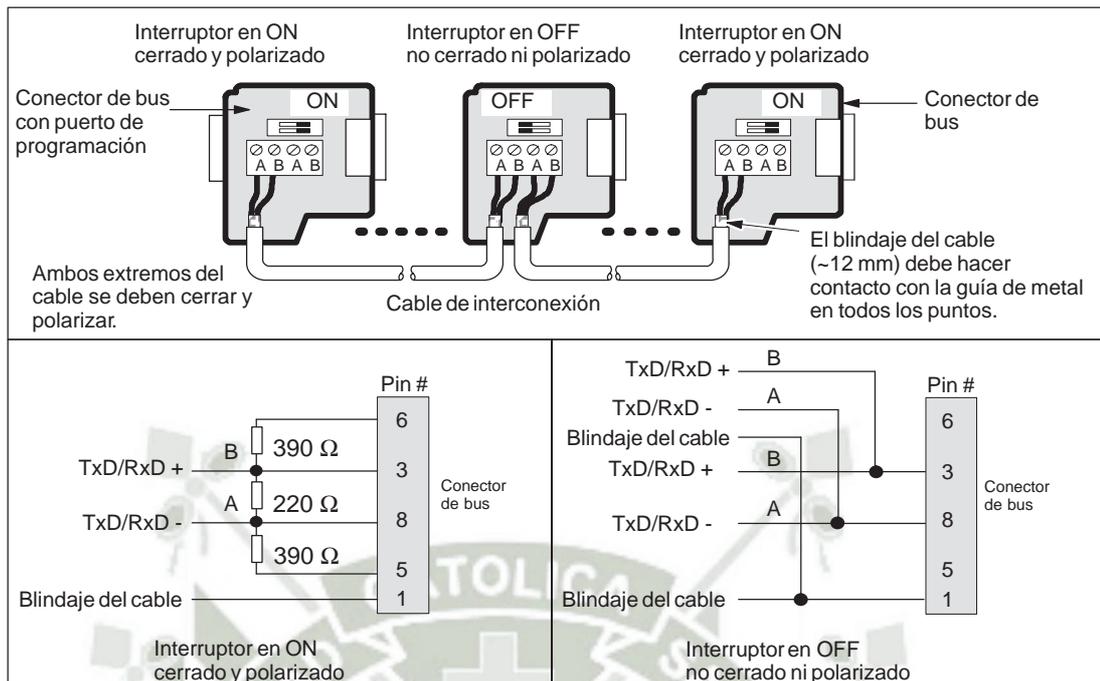


Figura 7-23 Polarizar y cerrar el cable de interconexión

### Cable para una red PROFIBUS

En la tabla 7-5 figuran los datos técnicos generales de un cable para una red PROFIBUS. En el Anexo E se indica la referencia de Siemens de cables PROFIBUS que cumplan los requisitos indicados.

Tabla 7-5 Datos técnicos generales de un cable para una red PROFIBUS

Características generales	Datos técnicos
Tipo de entrada	Apantallado, con par trenzado
Sección transversal del cable	24 AWG (0,22 mm <sup>2</sup> ) o superior
Capacidad del cable	< 60 pF/m
Impedancia nominal	100 Ω a 120 Ω

La longitud máxima de un segmento de red PROFIBUS depende de la velocidad de transferencia y del tipo de cable utilizados. En la tabla 7-6 figuran las longitudes máximas de los segmentos para el cable indicado en la tabla 7-5.

Tabla 7-6 Longitud máxima del cable en un segmento de una red PROFIBUS

Velocidad de transferencia	Longitud máxima del cable en un segmento
9,6 kbit/s a 19,2 kbit/s	1.200 m
187,5 kbit/s	1.000 m

## Repetidores

Siemens ofrece repetidores para interconectar segmentos de redes PROFIBUS (v. fig. 7-24). Utilizando repetidores es posible ampliar la longitud total de la red, añadir dispositivos a la misma y/o aislar diferentes segmentos de la red. El protocolo PROFIBUS asiste máximo 32 dispositivos en un segmento de red de hasta 1.200 m a una velocidad de transferencia de 9.600 bit/s. Cada repetidor permite añadir 32 dispositivos adicionales a la red y así ampliarla 1.200 m con una velocidad de transferencia de 9.600 bit/s. En una red se pueden utilizar 9 repetidores como máximo. Cada repetidor permite polarizar y cerrar el segmento de red en cuestión. En el Anexo E se indican los números de referencia.

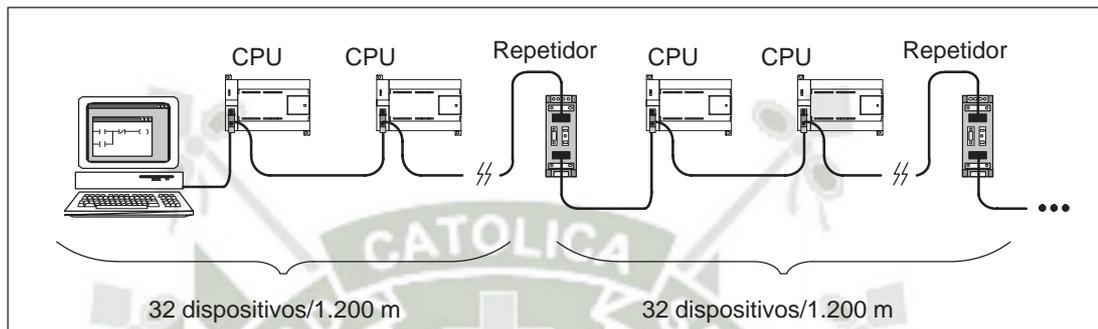


Figura 7-24 Red con repetidores

## 7.7 Utilizar el cable PC/PPI con otros dispositivos y en modo Freepport

El cable PC/PPI y el modo Freepport se pueden utilizar para conectar las CPUs S7-200 a numerosos dispositivos compatibles con el estándar RS-232.

Hay dos tipos de cables PC/PPI:

- Un cable PC/PPI aislado con un puerto RS-232 que tiene 5 interruptores DIP para ajustar la velocidad de transferencia y configurar otros parámetros más (v. fig. 7-26). Los datos técnicos de dicho cable PC/PPI se indican en el Anexo A.
- Un cable PC/PPI no aislado con un puerto RS-232 que tiene 4 interruptores DIP para ajustar la velocidad de transferencia. Los datos técnicos del cable PC/PPI no aislado se indican en la versión anterior del *Manual del sistema de automatización S7-200* (referencia: 6ES7298-8FA01-8BH0).

Ambos cables PC/PPI asisten velocidades de transferencia comprendidas entre 600 bit/s y 38.400 bit/s. Utilice los interruptores DIP dispuestos en la carcasa del cable PC/PPI para configurar la velocidad de transferencia correcta. La tabla 7-7 muestra las velocidades de transferencia y las posiciones de los interruptores DIP.

Tabla 7-7 Posición de los interruptores DIP en el cable PC/PPI para seleccionar la velocidad de transferencia

Velocidad de transferencia	Interruptor DIP (1 = arriba)
38400	000
19200	001
9600	010
4800	011
2400	100
1200	101
600	110

El cable PC/PPI se encuentra en modo de transmisión cuando los datos se envían del puerto RS-232 al RS-485. En cambio, se encuentra en modo de recepción al estar inactivo, o bien cuando los datos se transmiten del puerto RS-485 al RS-232. El cable cambia inmediatamente de modo de recepción a transmisión cuando detecta caracteres en el canal de transmisión del RS-232. El cable cambia nuevamente a modo de recepción cuando el canal de transmisión del RS-232 está inactivo durante el tiempo de inversión del cable. Dicho tiempo depende de la velocidad de transferencia seleccionada con los interruptores DIP del cable (v. tabla 7-8).

Si el cable PC/PPI se utiliza en un sistema que use también el modo Freeport, el tiempo de inversión se deberá tener en cuenta en el programa de usuario de la CPU S7-200 en las siguiente situaciones:

- La CPU S7-200 responde a los mensajes que envía el dispositivo RS-232.  
Tras recibir una petición del dispositivo RS-232, la transmisión de una respuesta de la CPU S7-200 se deberá retardar por un período mayor o igual al tiempo de inversión del cable.
- El dispositivo RS-232 responde a los mensajes que envía la CPU S7-200.  
Tras recibir una respuesta del dispositivo RS-232, la transmisión de la siguiente petición de la CPU S7-200 se deberá retardar por un período mayor o igual al tiempo de inversión del cable.

En ambos casos, el tiempo de retardo es suficiente para que el cable PC/PPI pueda cambiar de modo de transmisión a modo de recepción, enviando entonces los datos del puerto RS-485 al RS-232.

Tabla 7-8 Tiempo de inversión del cable PC/PPI (cambio de transmisión a recepción)

Velocidad de transferencia	Tiempo de inversión (en milisegundos)
38400	0,5
19200	1
9600	2
4800	4
2400	7
1200	14
600	28

### Utilizar un módem con un cable PC/PPI de 5 interruptores

El cable PC/PPI de 5 interruptores DIP se puede utilizar para conectar el puerto de comunicación RS-232 de un módem a una CPU S7-200. Por lo general, los módems utilizan las señales de control RS-232 (tales como RTS, CTS y DTR) para que un PC pueda controlar el módem. El cable PC/PPI no vigila ninguna de estas señales pero ofrece RTS en modo DTE. Si se utiliza un módem con un cable PC/PPI, el módem se deberá configurar para que funcione sin estas señales. Como mínimo, el módem se debe configurar de manera que ignore la señal DTR. Consulte el manual del módem para determinar los comandos necesarios para configurarlo.

Para el puerto RS-232 del cable PC/PPI de 5 interruptores se puede ajustar el modo DCE (equipo de comunicación de datos), o bien el modo DTE (equipo terminal de datos). Las únicas señales presentes en dicho puerto son las de transmitir datos, petición de transmitir, recibir datos y tierra. El cable PC/PPI de 5 interruptores no usa ni emite la señal CTS (preparado para transmitir). En las tablas 7-9 y 7-10 se indica la asignación de los pines del cable PC/PPI.

Un módem es considerado un equipo de comunicación de datos (DCE). Al conectar un cable PC/PPI a un módem, el puerto RS-232 del cable PC/PPI se deberá configurar en modo DTE (equipo terminal de datos), conforme a la selección efectuada con el interruptor DIP 5 del cable. Así se evita la necesidad de utilizar un adaptador de módem nulo entre el cable PC/PPI y el módem. Sin embargo, puede precisarse un adaptador de 9 a 25 pines (dependiendo del conector del módem). La figura 7-25 muestra una configuración típica y la asignación de pines de un adaptador de 25 a 9 pines.

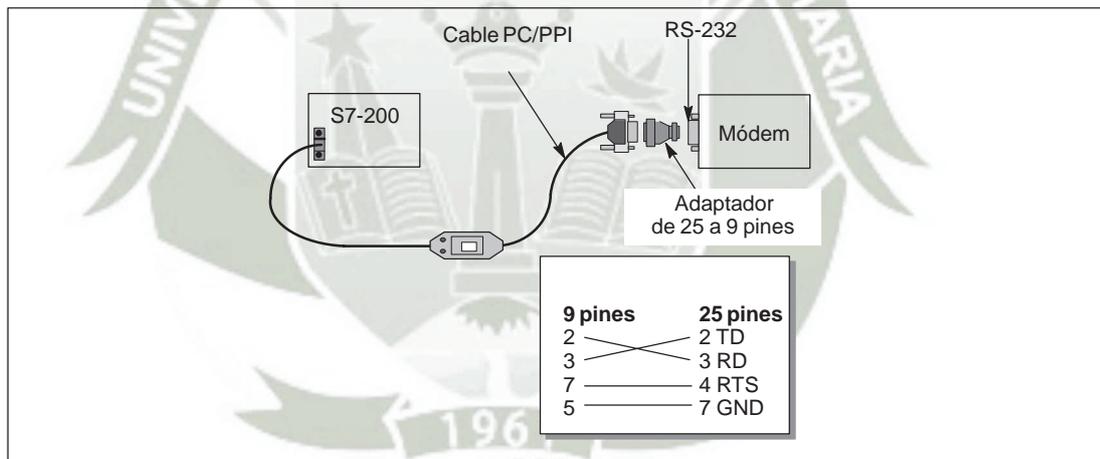


Figura 7-25 Asignación de pines para un cable PC/PPI de 5 interruptores DIP con un módem

Para ajustar el modo DCE (equipo de comunicación de datos), es preciso colocar el 5º interruptor en la posición "0" (= abajo). Para ajustar el modo DTE (equipo terminal de datos), es preciso colocar el 5º interruptor en la posición "1" (= arriba). La tabla 7-9 muestra los números de los pines y las funciones del puerto RS-485 a RS-232 del cable PC/PPI en modo DTE. La tabla 7-10 muestra los números de los pines y las funciones del puerto RS-485 a RS-232 del cable PC/PPI en modo DCE. Hay que tener en cuenta que el cable PC/PPI sólo envía RTS (peticiones de transmitir) si está en modo DTE.

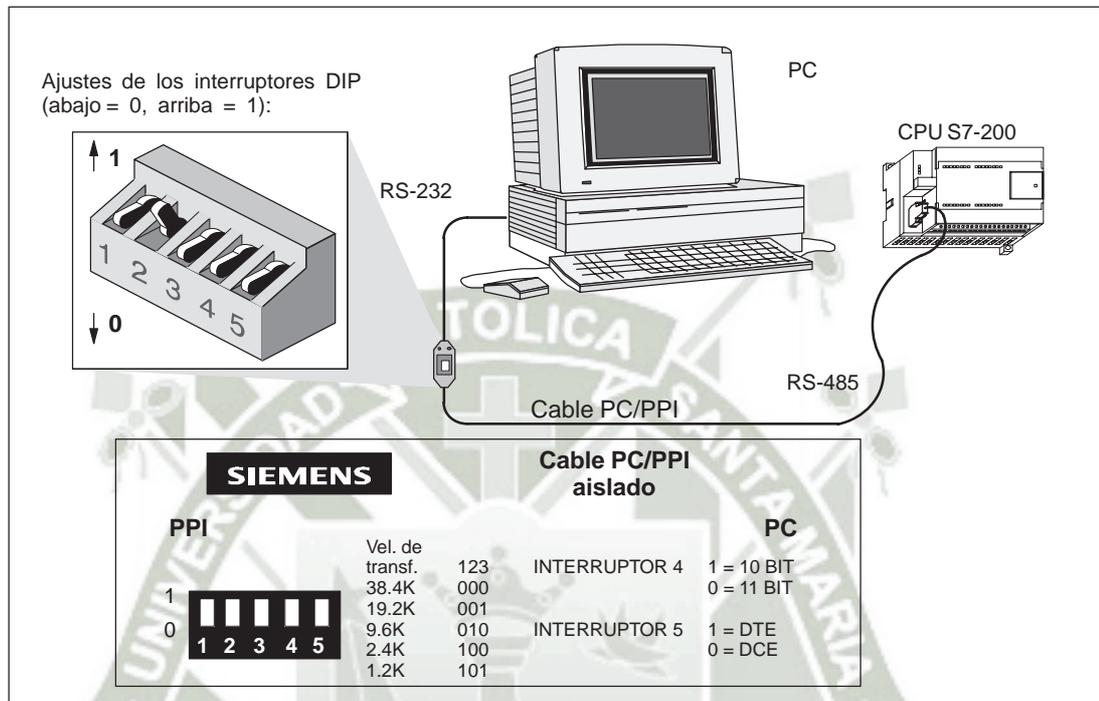


Figura 7-26 Comunicación con una CPU en modo PPI

El 4º interruptor DIP del cable PC/PPI le indica a la CPU S7-200 si debe utilizar el protocolo de 10 bits o el protocolo PPI normal de 11 bits. Si la CPU no está conectada a STEP 7-Micro/WIN 32, no se deberá cambiar el ajuste del interruptor (11 bits) para que pueda funcionar correctamente con otros dispositivos.

Tabla 7-9 Asignación de pines para un conector de RS-485 a RS-232 DTE<sup>1</sup>

Asignación de pines del conector RS-485		Asignación de pines del conector RS-232 DTE <sup>1</sup>	
Nº de pin	Descripción de la señal	Nº de pin	Descripción de la señal
1	Tierra (RS-485)	1	Data Carrier Detect (DCD) (no utilizado)
2	Hilo de retorno 24 V (tierra RS-485)	2	Receive Data (RD) (entrada al cable PC/PPI)
3	Señal B (RxD/TxD+)	3	Transmit Data (TD) (salida del cable PC/PPI)
4	RTS (nivel TTL)	4	Data Terminal Ready (DTR) (no utilizado)
5	Tierra (RS-485)	5	Tierra (RS-232)
6	+5 V (con resistor en serie de 100 $\Omega$ )	6	Data Set Ready (DSR) (no utilizado)
7	Alimentación 24 V	7	Request To Send (RTS) (salida del cable PC/PPI)
8	Señal A (RxD/TxD-)	8	Clear To Send (CTS) (no utilizado)
9	Selección de protocolo	9	Ring Indicator (RI) (no utilizado)

<sup>1</sup> Para los módems se necesita un adaptador de hembra a macho y un adaptador de 9 a 25 pines.

Tabla 7-10 Asignación de pines para un conector de RS-485 a RS-232 DCE

Asignación de pines del conector RS-485		Asignación de pines del conector RS-232 DCE	
Nº de pin	Descripción de la señal	Nº de pin	Descripción de la señal
1	Tierra (RS-485)	1	Data Carrier Detect (DCD) (no utilizado)
2	Hilo de retorno 24 V (tierra RS-485)	2	Receive Data (RD) (salida del cable PC/PPI)
3	Señal B (RxD/TxD+)	3	Transmit Data (TD) (entrada al cable PC/PPI)
4	RTS (nivel TTL)	4	Data Terminal Ready (DTR) (no utilizado)
5	Tierra (RS-485)	5	Tierra (RS-232)
6	+5 V (con resistor en serie de 100 $\Omega$ )	6	Data Set Ready (DSR) (no utilizado)
7	Alimentación 24 V	7	Request To Send (RTS) (no utilizado)
8	Señal A (RxD/TxD-)	8	Clear To Send (CTS) (no utilizado)
9	Selección de protocolo	9	Ring Indicator (RI) (no utilizado)

### Utilizar un módem con un cable PC/PPI de 4 interruptores

El cable PC/PPI de 4 interruptores DIP se puede utilizar para conectar el puerto de comunicación RS-232 de un módem a una CPU S7-200. Por lo general, los módems utilizan las señales de control RS-232 (tales como RTS, CTS y DTR) para que un PC pueda controlar el módem. Este cable PC/PPI no utiliza ninguna de estas señales. Por tanto, si un módem se utiliza con un cable PC/PPI de 4 interruptores DIP, el módem se deberá configurar para que no utilice ninguna de estas señales. Como mínimo, se deberán ignorar las señales RTS y DTR. Consulte el manual del módem para determinar los comandos necesarios para configurarlo.

Un módem es un equipo de comunicación de datos (DCE). El puerto RS-232 del cable PC/PPI de 4 interruptores también es un DCE. Al conectarse dos dispositivos de una misma clase (ambos DCE), los pines para transmitir y recibir datos se deberán invertir utilizando para ello un adaptador de módem nulo. La figura 7-27 muestra una configuración típica y la asignación de pines de un adaptador de módem nulo.

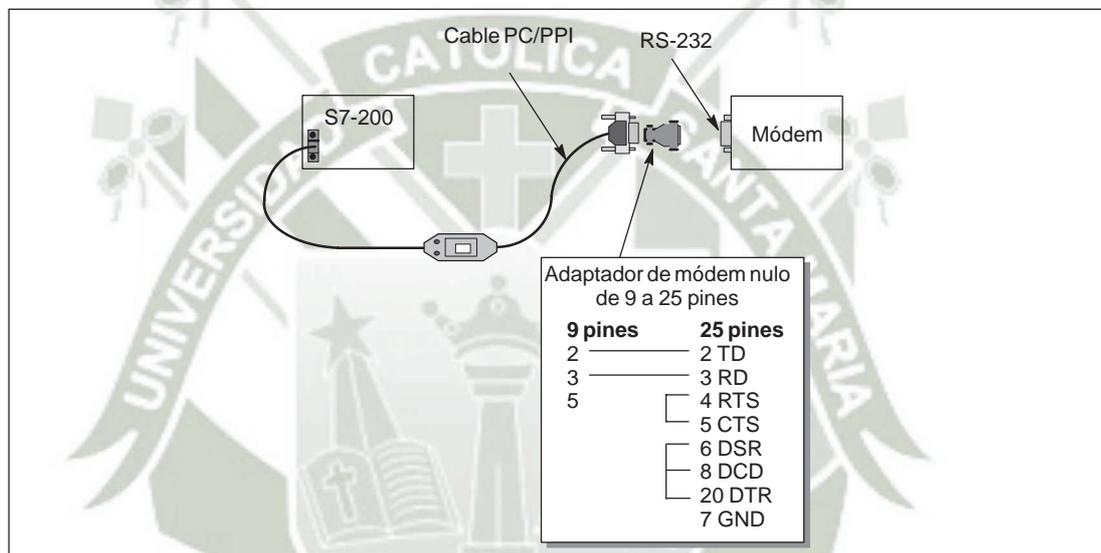


Figura 7-27 Módem de 11 bits con unadaptador de módem nulo combinado con un adaptador de 9 a 25 pines

## 7.8 Rendimiento de la red

### Optimizar el rendimiento de la red

Los dos factores con mayor efecto en el rendimiento de la red son la velocidad de transferencia y el número de maestros. El rendimiento óptimo de la red se logra utilizando la velocidad de transferencia máxima asistida por todos los dispositivos. Si el número de maestros se reduce a un mínimo, aumenta también el rendimiento de la red. Cada maestro de la red prolonga el tiempo de procesamiento en la red. Por tanto, dicho tiempo se acortará cuanto menor sea el número de maestros.

Los siguientes factores influyen también en el rendimiento de la red:

- Las direcciones elegidas para los maestros y esclavos.
- El factor de actualización GAP.
- La dirección de estación más alta.

Las direcciones de los maestros se deberán elegir de forma secuencial, evitando huecos entre las mismas. Si hay un hueco (GAP) entre las direcciones de los maestros, éstos comprueban continuamente las direcciones del GAP para averiguar si hay otro maestro que desee conectarse online. Dicha comprobación aumenta el tiempo de procesamiento de la red. Si no hay ningún hueco entre las direcciones de los maestros, la comprobación no se efectúa, por lo que se minimiza el tiempo de procesamiento.

Las direcciones de los esclavos se pueden ajustar a cualquier valor sin que ello influya en el rendimiento de la red, a menos que los esclavos se encuentren entre los maestros. En este último caso aumentaría también el tiempo de procesamiento de la red como si existieran huecos entre las direcciones de los maestros.

Las CPUs S7-200 se pueden configurar para que comprueben sólo periódicamente si hay huecos entre las direcciones. Para ello, en STEP 7-Micro/WIN 32 se ajusta el factor de actualización GAP cuando se configura el correspondiente puerto de la CPU. El factor de actualización GAP le indica a la CPU la frecuencia con la que debe comprobar el hueco de direcciones para determinar si hay otros maestros. Si se elige "1" como factor de actualización GAP, la CPU comprobará el hueco de direcciones cada vez que tenga el testigo en su poder. Si se elige "2", la CPU comprobará el hueco cada 2 veces que tenga el testigo en su poder. Ajustándose un factor de actualización GAP más elevado se reduce el tiempo de procesamiento en la red si hay huecos entre las direcciones de los maestros. Si no existen huecos, el factor de actualización GAP no tendrá efecto alguno en el rendimiento. Si se ajusta un factor de actualización GAP elevado pueden producirse grandes demoras cuando se desee incorporar nuevos maestros a la red, puesto que las direcciones se comprueban con menor frecuencia. El factor de actualización GAP se utiliza únicamente cuando una CPU actúa de maestro PPI.

La dirección de estación más alta es el valor donde un maestro debe buscar a otro. Ajustándose dicho valor se limita el hueco de direcciones que el último maestro (la dirección más alta) debe comprobar en la red. Limitando el tamaño del hueco de direcciones se reduce el tiempo necesario para buscar e incorporar en la red a un nuevo maestro. La dirección de estación más alta no tiene efecto sobre las direcciones de los esclavos. Los maestros pueden comunicarse con esclavos cuyas direcciones sean superiores a la dirección de estación más alta. Ésta última se utiliza sólo cuando una CPU actúa de maestro PPI. La dirección de estación más alta se puede ajustar en STEP 7-Micro/WIN 32 al configurar el puerto de la CPU.

Por regla general, se deberá ajustar en todos los maestros un mismo valor para la dirección de estación más alta. Dicha dirección debería ser mayor o igual a la dirección más alta de los maestros. El ajuste estándar de la dirección de estación más alta en las CPUs S7-200 es "31".

### Rotación del testigo

En una red con token passing (paso de testigo), la estación que tiene el testigo en su poder es la única que puede iniciar la comunicación. Por tanto, un importante factor en una red con token passing es el tiempo de rotación del testigo. Éste es el tiempo que el testigo necesita para recorrer el anillo lógico, o sea, para circular por todos los maestros (token holders) que lo constituyen. El ejemplo de la figura 7-28 muestra el funcionamiento de una red multimaestro.

La red de la figura 7-28 comprende cuatro CPUs S7-200, teniendo cada una de ellas su propio TD 200. Dos CPUs 224 recopilan datos de las demás CPUs.

#### Nota

El ejemplo indicado se basa en la configuración que muestra la figura 7-28. Dicha configuración incluye visualizadores de textos TD 200. Las CPUs 224 utilizan operaciones NETR y NETW. Las fórmulas para calcular el tiempo de posesión y de rotación del testigo que muestra la figura 7-29 se basan también en dicha configuración.

El software COM PROFIBUS permite analizar el rendimiento de la red.

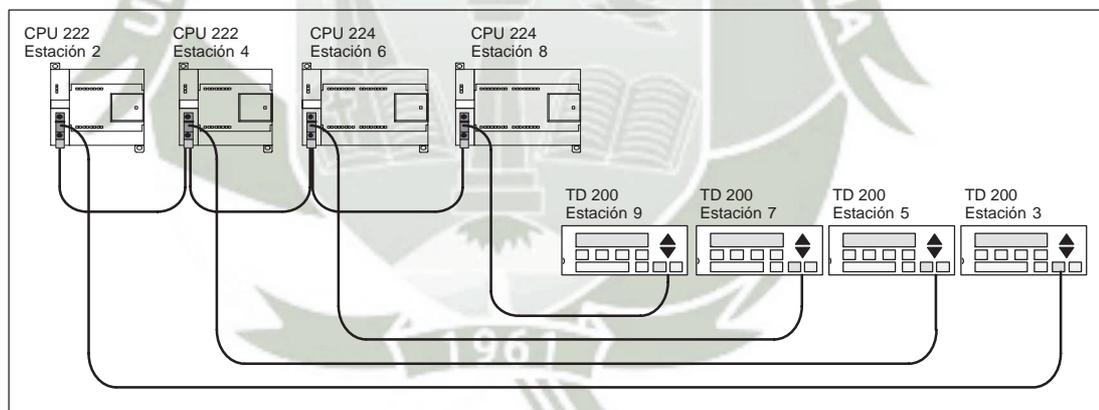


Figura 7-28 Ejemplo de una red con token passing

En esta configuración, un TD 200 (estación 3) se comunica con una CPU 222 (estación 2), otro TD 200 (estación 5) se comunica con la otra CPU 222 (estación 4), etc. Además, una CPU 224 (estación 6) envía mensajes a las estaciones 2, 4 y 8, y la otra CPU 224 (estación 8) envía mensajes a las estaciones 2, 4 y 6. Esta red comprende seis estaciones maestras (los cuatro TDs 200 y las dos CPUs 224), así como dos estaciones esclavas (las dos CPUs 222).

### Enviar mensajes

Para que un maestro pueda enviar un mensaje deberá tener el testigo en su poder. Ejemplo: cuando la estación 3 tiene el testigo en su poder, envía una petición a la estación 2 y pasa el testigo a la estación 5. La estación 5 envía una petición a la estación 4 y pasa el testigo a la estación 6. La estación 6 envía un mensaje a las estaciones 2, 4 u 8 y pasa el testigo a la estación 7. Este proceso de enviar un mensaje y pasar el testigo continúa por el anillo lógico de la estación 3 a la estación 5, a la estación 6, a la estación 7, a la estación 8, a la estación 9 y de allí retorna finalmente a la estación 3. El testigo debe recorrer todo el anillo lógico para que un maestro pueda enviar una petición de información. En un anillo lógico compuesto por seis estaciones que envían una petición para leer o escribir un valor de doble palabra (cuatro bytes de datos) cada vez que tienen el testigo en su poder, el tiempo de rotación del mismo será de unos 900 milisegundos a una velocidad de transferencia de 9.600 bit/s. Si aumenta el número de bytes de datos a los que se debe acceder por mensaje o si se incorporan más estaciones, se prolongará el tiempo de rotación del testigo.



## Tiempo de rotación del testigo

Éste depende del tiempo que cada estación tiene el testigo en su poder. El tiempo de rotación del testigo en redes S7-200 multimaestro se puede determinar sumando los tiempos de posesión del testigo por parte de cada maestro. Si se ha habilitado el modo maestro PPI (en el protocolo PPI de la red en cuestión), es posible enviar mensajes a otras CPUs utilizando las operaciones Leer de la red (NETR) y Escribir en la red (NETW) con la CPU. En el apartado 9.16 (Operaciones de comunicación SIMATIC) del capítulo 9 se describen dichas operaciones. Si envía mensajes utilizando las operaciones NETR y NETW, puede utilizar la fórmula que muestra la figura 7-29 para calcular el tiempo aproximado de rotación del testigo, dando por supuesto que:

- Cada estación envía una petición cuando tiene el testigo en su poder.
- La petición es una operación de lectura o de escritura a direcciones consecutivas de datos.
- No hay conflictos de acceso al único búfer de comunicación de la CPU.
- Ninguna CPU tiene un tiempo de ciclo superior a aprox. 10 ms.

Tiempo de posesión del testigo ( $T_{\text{pos}}$ ) = (tiempo necesario 128 +  $n$  caráct. datos) \* 11 bits/caráct. \* 1/vel. transf.

Tiempo de rotación del testigo ( $T_{\text{rot}}$ ) =  $T_{\text{pos}}$  del maestro 1 +  $T_{\text{pos}}$  del maestro 2 + ... +  $T_{\text{pos}}$  del maestro  $m$

siendo  $n$  el número de caracteres de datos (bytes)  
y  $m$  el número de maestros

Conforme al ejemplo indicado arriba, el tiempo de rotación se calcula de la siguiente forma si el tiempo de posesión del testigo es igual en los seis maestros:

$T$  (tiempo de posesión del testigo) =  $(128 + 4 \text{ caracteres}) * 11 \text{ bits/carácter} * 1/9.600 \text{ "bit times"/s}$   
= 151,25ms/maestro

$T$  (tiempo de rotación del testigo) =  $151,25 \text{ ms/maestro} * 6 \text{ maestros}$   
= 907.5 ms

(Un "bit time" equivale a la duración de un período de señal).

Figura 7-29 Fórmulas para determinar los tiempos de posesión y de rotación del testigo utilizando las operaciones NETR y NETW

### Tiempo de rotación del testigo en función del número de estaciones

Las tablas 7-11, 7-12 y 7-13 muestran el tiempo de rotación del testigo en función del número de estaciones y del volumen de datos a transferir a 9,6 kbit/s, 19,2 kbit/s y 187,5 kbit/s, respectivamente. Dichos tiempos son válidos utilizando las operaciones Leer de la red (NETR) y Escribir en la red (NETW) con la CPU u otros maestros.

Tabla 7-11 Tiempo de rotación del testigo en función del número de estaciones y del volumen de datos a 9,6 kbit/s

Bytes transferidos por estación a 9,6 kbit/s	Número de estaciones (indicaciones de tiempo en segundos)								
	2 estaciones	3 estaciones	4 estaciones	5 estaciones	6 estaciones	7 estaciones	8 estaciones	9 estaciones	10 estaciones
1	0,30	0,44	0,59	0,74	0,89	1,03	1,18	1,33	1,48
2	0,30	0,45	0,60	0,74	0,89	1,04	1,19	1,34	1,49
3	0,30	0,45	0,60	0,75	0,90	1,05	1,20	1,35	1,50
4	0,30	0,45	0,61	0,76	0,91	1,06	1,21	1,36	1,51
5	0,30	0,46	0,61	0,76	0,91	1,07	1,22	1,37	1,52
6	0,31	0,46	0,61	0,77	0,92	1,07	1,23	1,38	1,54
7	0,31	0,46	0,62	0,77	0,93	1,08	1,24	1,39	1,55
8	0,31	0,47	0,62	0,78	0,94	1,09	1,25	1,40	1,56
9	0,31	0,47	0,63	0,78	0,94	1,10	1,26	1,41	1,57
10	0,32	0,47	0,63	0,79	0,95	1,11	1,27	1,42	1,58
11	0,32	0,48	0,64	0,80	0,96	1,11	1,27	1,43	1,59
12	0,32	0,48	0,64	0,80	0,96	1,12	1,28	1,44	1,60
13	0,32	0,48	0,65	0,81	0,97	1,13	1,29	1,45	1,62
14	0,33	0,49	0,65	0,81	0,98	1,14	1,30	1,46	1,63
15	0,33	0,49	0,66	0,82	0,98	1,15	1,31	1,47	1,64
16	0,33	0,50	0,66	0,83	0,99	1,16	1,32	1,49	1,65

Tabla 7-12 Tiempo de rotación del testigo en función del número de estaciones y del volumen de datos a 19,2 kbit/s

Bytes transferidos por estación a 19,2 kbit/s	Número de estaciones (indicaciones de tiempo en segundos)								
	2 estaciones	3 estaciones	4 estaciones	5 estaciones	6 estaciones	7 estaciones	8 estaciones	9 estaciones	10 estaciones
1	0,15	0,22	0,30	0,37	0,44	0,52	0,59	0,67	0,74
2	0,15	0,22	0,30	0,37	0,45	0,52	0,60	0,67	0,74
3	0,15	0,23	0,30	0,38	0,45	0,53	0,60	0,68	0,75
4	0,15	0,23	0,30	0,38	0,45	0,53	0,61	0,68	0,76
5	0,15	0,23	0,30	0,38	0,46	0,53	0,61	0,69	0,76
6	0,15	0,23	0,31	0,38	0,46	0,54	0,61	0,69	0,77
7	0,15	0,23	0,31	0,39	0,46	0,54	0,62	0,70	0,77
8	0,16	0,23	0,31	0,39	0,47	0,55	0,62	0,70	0,78
9	0,16	0,24	0,31	0,39	0,47	0,55	0,63	0,71	0,78
10	0,16	0,24	0,32	0,40	0,47	0,55	0,63	0,71	0,79
11	0,16	0,24	0,32	0,40	0,48	0,56	0,64	0,72	0,80
12	0,16	0,24	0,32	0,40	0,48	0,56	0,64	0,72	0,80
13	0,16	0,24	0,32	0,40	0,48	0,57	0,65	0,73	0,81
14	0,16	0,24	0,33	0,41	0,49	0,57	0,65	0,73	0,81
15	0,16	0,25	0,33	0,41	0,49	0,57	0,66	0,74	0,82
16	0,17	0,25	0,33	0,41	0,50	0,58	0,66	0,74	0,83

Tabla 7-13 Tiempo de rotación del testigo en función del número de estaciones y del volumen de datos a 187,5 kbit/s

Bytes transferidos por estación a 187,5 kbit/s	Número de estaciones (indicaciones de tiempo en milisegundos)								
	2 estaciones	3 estaciones	4 estaciones	5 estaciones	6 estaciones	7 estaciones	8 estaciones	9 estaciones	10 estaciones
1	8,68	13,02	17,37	21,71	26,05	30,39	34,73	39,07	43,41
2	8,80	13,20	17,60	22,00	26,40	30,80	35,20	39,60	44,00
3	8,92	13,38	17,83	22,29	26,75	31,21	35,67	40,13	44,59
4	9,03	13,55	18,07	22,59	27,10	31,62	36,14	40,66	45,17
5	9,15	13,73	18,30	22,88	27,46	32,03	36,61	41,18	45,76
6	9,27	13,90	18,54	23,17	27,81	32,44	37,08	41,71	46,35
7	9,39	14,08	18,77	23,47	28,16	32,85	37,55	42,24	46,93
8	9,50	14,26	19,01	23,76	28,51	33,26	38,02	42,77	47,52
9	9,62	14,43	19,24	24,05	28,86	33,67	38,49	43,30	48,11
10	9,74	14,61	19,48	24,35	29,22	34,09	38,95	43,82	48,69
11	9,86	14,78	19,71	24,64	29,57	34,50	39,42	44,35	49,28
12	9,97	14,96	19,95	24,93	29,92	34,91	39,89	44,88	49,87
13	10,09	15,14	20,18	25,23	30,27	35,32	40,36	45,41	50,45
14	10,21	15,31	20,42	25,52	30,62	35,73	40,83	45,84	51,04
15	10,33	15,49	20,65	25,81	30,98	36,14	41,30	46,46	51,63

## Convenciones para las operaciones S7-200

# 8

En el presente capítulo se utilizan las siguientes convenciones para representar las operaciones en los lenguajes de programación KOP (esquema de contactos), FUP (diagrama de funciones) y AWL (lista de instrucciones), indicándose también las CPUs que asisten la correspondiente operación.

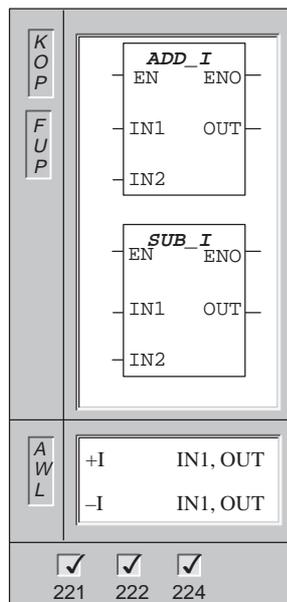
### Índice del capítulo

Apartado	Descripción	Página
8.1	Conceptos y convenciones para programar con STEP 7-Micro/WIN 32	8-2
8.2	Márgenes válidos para las CPUs S7-200	8-7

## 8.1 Conceptos y convenciones para programar con STEP 7-Micro/WIN 32

El siguiente diagrama muestra el formato de las operaciones Micro/WIN 32 utilizado en este capítulo. Los componentes del formato de la operación se describen a continuación del diagrama.

### Sumar y restar enteros de 16 bits



Las operaciones **Sumar enteros de 16 bits** y **Restar enteros de 16 bits** suman/restan dos enteros de 16 bits, arrojando un resultado de 16 bits (OUT).

En KOP y FUP:  $IN1 + IN2 = OUT$   
 $IN1 - IN2 = OUT$

En AWL:  $IN1 + OUT = OUT$   
 $OUT - IN1 = OUT$

Condiciones de error que ponen ENO a 0: SM1.1 (desbordamiento), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero); SM1.1 (desbordamiento); SM1.2 (negativo)

Entradas/salidas	Operandos	Tipos de datos
IN1, IN2	VW, IW, QW, MW, SW, SMW, LW, AIW, T, C, AC, constante, *VD, *AC, *LD	INT
OUT	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, *VD, *AC, *LD	INT

**Título de la operación o del grupo de operaciones:** En el presente ejemplo, el título es **Sumar enteros de 16 bits**.

**Figura correspondiente a la operación STEP 7-Micro/WIN 32:** La figura que aparece debajo del título de la operación muestra los elementos KOP y FUP de la operación, así como (en las operaciones SIMATIC) la nemotécnica y los operandos AWL. En algunos casos, la figura de las operaciones KOP y FUP es idéntica, mostrando un solo cuadro que contiene tanto la figura correspondiente a KOP como a FUP (como en el presente ejemplo). La nemotécnica y los operandos AWL aparecen siempre en un cuadro por separado.

En el ejemplo, los cuadros KOP/FUP tienen tres entradas (que se muestran siempre en el lado izquierdo del cuadro) y dos salidas (siempre en el lado derecho). En KOP hay dos tipos básicos de entradas y salidas (E/S). El primer tipo de E/S es una entrada o salida de circulación de corriente.

En KOP, de forma similar a los escalones de un diagrama lógico de escalera de relés, hay una barra de alimentación a la izquierda que está energizada. Los contactos cerrados permiten que la corriente circule por ellos hasta el siguiente elemento, en tanto que los contactos abiertos bloquean el flujo de energía. Cualquier elemento KOP que se puede conectar a la barra de alimentación izquierda o derecha, o bien a un contacto, tiene una entrada y/o una salida de circulación de corriente.

En el lenguaje de programación FUP (SIMATIC), que no utiliza las barras de alimentación izquierda ni derecha, el término "circulación de corriente" se utiliza para expresar el concepto análogo del flujo de señales por los bloques lógicos FUP. El recorrido "1" lógico por los elementos FUP se denomina circulación de corriente.

En KOP, una entrada o salida de circulación de corriente no se utiliza más que para el flujo de señales, no pudiendo asignarse a ningún operando. En FUP, el origen de una entrada de circulación de corriente y el destino de una salida de circulación de corriente se pueden asignar directamente a un operando.

Adicionalmente a la circulación de corriente, numerosas operaciones KOP/FUP/AWL (si bien no todas) tienen uno o más operandos de entrada y de salida. Los parámetros permitidos para dichos operandos se indican en la tabla de entradas/salidas que aparece debajo de la figura KOP/FUP/AWL.

**Tipo de CPU:** La figura muestra los tipos de CPU que asisten la operación. En el presente ejemplo, la operación se puede utilizar con las CPUs 221, 222 y 224.

**Descripción de la operación:** El texto que aparece a la derecha de la figura de la operación en la página 8-2 describe su funcionamiento. En algunos casos, la operación se describe por separado para cada uno de los lenguajes de programación. En otros, hay una sola descripción aplicable a los tres lenguajes. Tenga en cuenta que la terminología IEC difiere considerablemente de la terminología SIMATIC (tanto en cuanto a los nombres de las operaciones como al de los lenguajes de programación). Por ejemplo, en SIMATIC existe la operación Contar adelante (CTU), en tanto que en IEC se hace referencia al bloque funcional Contador ascendente (CTU). Además, en SIMATIC se habla del lenguaje KOP (Esquema de contactos) que equivale en IEC al lenguaje LD (Diagrama de escalera). Igualmente, el lenguaje FUP (Diagrama de funciones) de SIMATIC se denomina FBD (Diagrama de bloques funcionales) en IEC.

**Condiciones de error que ponen ENO a 0:** Si las operaciones KOP/FUP tienen una salida ENO (salida de habilitación), se indican aquí las condiciones de error que ponen ENO a cero.

**Marcas especiales afectadas:** Si la operación afecta a ciertas marcas especiales como parte normal de la ejecución de la misma, se indican aquí las marcas en cuestión y la forma en que se ven afectadas.

**Tabla de operandos:** Debajo de la figura KOP/FUP/AWL aparece una tabla en la que se indican los operandos permitidos para cada una de las entradas y salidas, así como los tipos de datos de cada uno de los operandos. Los márgenes de memoria de los operandos correspondientes a cada CPU figuran en la tabla 8-3.

Los operandos y los tipos de datos EN/ENO no figuran en la tabla de operandos de la operación, puesto que son idénticos para todas las operaciones KOP y FUP. La tabla 8-1 muestra los operandos y tipos de datos EN/ENO para KOP y FUP, siendo aplicables a todas las operaciones KOP y FUP descritas en el presente manual.

Tabla 8-1 Operandos y tipos de datos EN/ENO para KOP y FUP

Editor	Entradas/salidas	Operandos	Tipos de datos
KOP	EN	Circulación de corriente	BOOL
	ENO	Circulación de corriente	BOOL
FUP	EN	I, Q, M, S, SM, T, C, V, L, circulación de corriente	BOOL
	ENO	I, Q, M, S, SM, T, C, V, L, circulación de corriente	BOOL

## Convenciones generales para programar en SIMATIC

**Segmento ("Network"):** En KOP, el programa se divide en segmentos denominados "networks". Un segmento es una red organizada, compuesta por contactos, bobinas y cuadros que se interconectan para conformar un circuito completo entre las barras de alimentación izquierda y derecha (no se permiten los cortocircuitos, ni los circuitos abiertos, ni la circulación de corriente inversa). STEP 7-Micro/WIN 32 ofrece la posibilidad de crear comentarios para cada uno de los segmentos del programa KOP.

El lenguaje FUP utiliza el concepto de segmentos para subdividir y comentar el programa. Los programas AWL no utilizan segmentos. Sin embargo, la palabra clave NETWORK se puede utilizar para estructurar el programa. En este caso, el programa se puede convertir a KOP o a FUP.

**Unidades de organización del programa:** En KOP, FUP o AWL, un programa comprende una parte obligatoria (como mínimo), pudiendo incluir también otras secciones opcionales. La parte obligatoria es el programa principal. Las secciones opcionales pueden incluir una o más subrutinas y/o rutinas de interrupción. Para desplazarse por las unidades de organización del programa, basta con seleccionar o hacer clic en la correspondiente ficha en STEP 7-Micro/WIN 32.

**Definición de EN/ENO:** EN (entrada de habilitación) es una entrada booleana para los cuadros KOP y FUP. Para que la operación se pueda ejecutar, el estado de señal de dicha entrada deberá ser "1" (ON). En AWL, las operaciones no tienen una entrada EN, pero el valor del nivel superior de la pila deberá ser un "1" lógico para poder ejecutar la correspondiente operación AWL.

ENO (salida de habilitación) es una salida booleana para los cuadros KOP y FUP. Si el estado de señal de la entrada EN es "1" y el cuadro ejecuta la función sin que se presenten errores, la salida ENO conducirá corriente al siguiente elemento. Si se detecta un error en la ejecución del cuadro, la circulación de corriente se detendrá en el cuadro que ha generado el error.

En AWL (SIMATIC) no existe la salida ENO, pero las operaciones AWL correspondientes a las funciones KOP y FUP con salidas ENO activan un bit ENO especial. A dicho bit se accede mediante la operación AWL AENO (Y-ENO), pudiendo utilizarse para generar el mismo efecto que el bit ENO de un cuadro.

**Entradas condicionadas e incondicionadas:** En KOP y FUP, un cuadro o una bobina que dependa de la circulación de corriente aparecerá sin conexión a ningún elemento a la izquierda. Una bobina o un cuadro que no dependa de la circulación de corriente se mostrará con una conexión directa a la barra de alimentación izquierda. La figura 8-1 muestra las entradas condicionadas e incondicionadas.

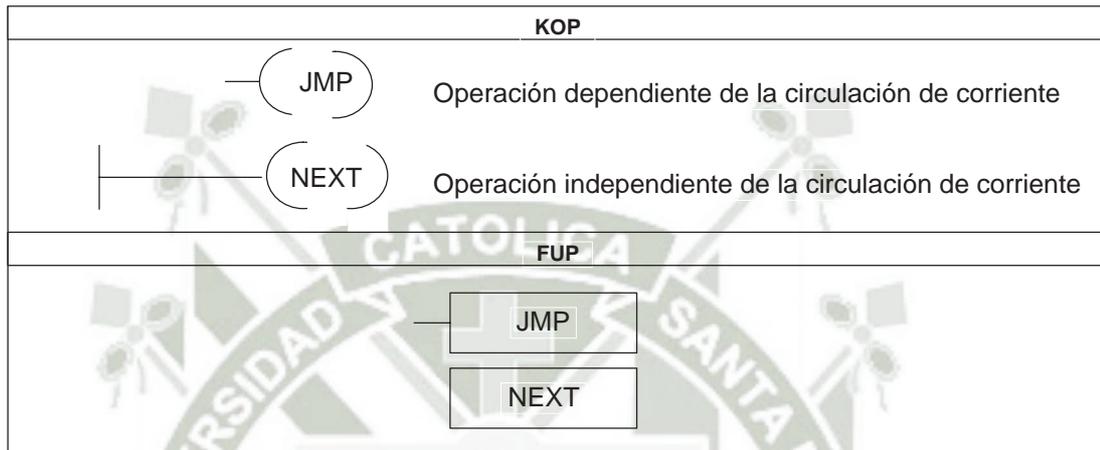


Figura 8-1 Diagrama KOP de entradas condicionadas e incondicionadas

**Operaciones sin salidas:** Los cuadros que no se puedan conectar en cascada se representan sin salidas booleanas. Dichos cuadros incluyen las llamadas a subrutinas, JMP, CRET, etc. También hay bobinas KOP que sólo se pueden disponer en la barra de alimentación izquierda, incluyendo LBL, NEXT, SCR, SCRE, etc. Éstas se representan en FUP en forma de cuadros con entradas sin meta y sin salidas.

**Operaciones de comparación:** Las operaciones de comparación FUP (SIMATIC), LD (IEC) y FBD (IEC) se representan mediante cuadros, aunque la operación se ejecute en forma de contacto.

La operación de comparación se ejecutará sin tener en cuenta el estado de señal. Si dicho estado es "0" (FALSO), el estado de señal de la salida también será "0" (FALSO). Si el estado de señal es "1" (VERDADERO), la salida se activará dependiendo del resultado de la comparación.

**Convenciones de STEP 7-Micro/WIN 32:** En STEP 7-Micro/WIN 32 rigen las siguientes convenciones:

- El símbolo “—>” del editor KOP es una conexión opcional para la circulación de corriente.
- El símbolo “—>>” es una conexión necesaria para la circulación de corriente.
- Si un nombre simbólico (p.ej. “var1”) aparece entre comillas dobles, significa que se trata de un símbolo global.
- Si un nombre simbólico (p.ej. #var1) va antecedido de un signo de número (#), significa que se trata de un símbolo local.
- El símbolo de operando “?” o “????” indica que se requiere un valor.
- Los símbolos “<<” o “>>” indican que se puede utilizar bien sea un valor, o bien la circulación de corriente.
- La >| indica que se trata de una salida de habilitación ENO.
- El símbolo % identifica una dirección directa en IEC.

**Símbolo de negación en FUP:** La condición lógica NOT del estado del operando o la corriente que circula por la entrada se representa mediante un pequeño círculo en la entrada de una operación FUP. En la figura 8-2, Q0.0 es igual al NOT de I0.0 AND I0.1.

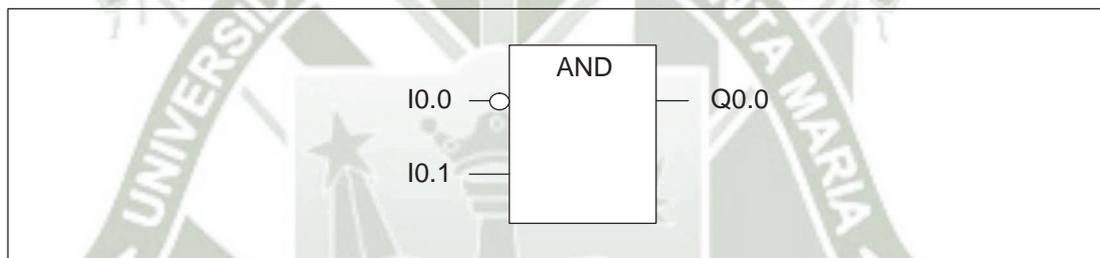


Figura 8-2 Diagrama FUP de la condición lógica NOT

**Indicadores directos en FUP:** La condición directa de un operando booleano se muestra mediante la línea vertical en la entrada de una operación FUP.

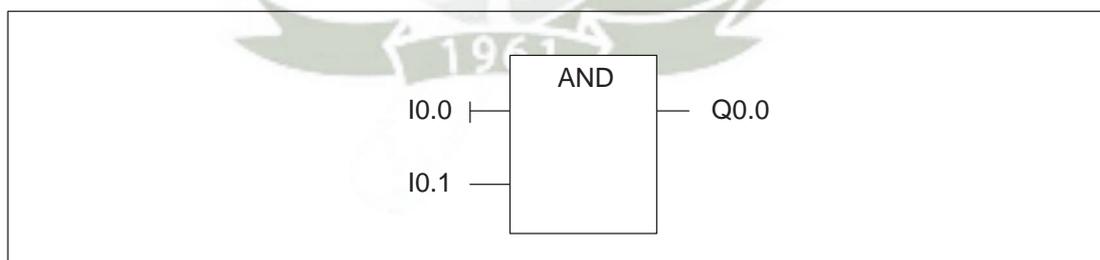


Figura 8-3 Diagrama FUP de la condición directa

**Tecla TAB en FUP:** La tecla TAB desplaza el cursor de una entrada a otra. La entrada seleccionada actualmente se destaca en rojo.

## 8.2 Márgenes válidos para las CPUs S7-200

Tabla 8-2 Resumen de los márgenes de memoria y funciones de las CPUs S7-200

Descripción	CPU 221	CPU 222	CPU 224
Tamaño del programa de usuario	2K palabras	2K palabras	4K palabras
Tamaño de los datos de usuario	1K palabras	1K palabras	2,5K palabras
Imagen del proceso de las entradas	I0.0 a I15.7	I0.0 a I15.7	I0.0 a I15.7
Imagen del proceso de las salidas	Q0.0 a Q15.7	Q0.0 a Q15.7	Q0.0 a Q15.7
Entradas analógicas (sólo lectura)	—	AIW0 a AIW30	AIW0 a AIW30
Salidas analógicas (sólo escritura)	—	AQW0 a AQW30	AQW0 a AQW30
Memoria de variables (V) <sup>1</sup>	VB0.0 a VB2047.7	VB0.0 a VB2047.7	VB0.0 a VB5119.7
Memoria local (L) <sup>2</sup>	LB0.0 a LB63.7	LB0.0 a LB63.7	LB0.0 a LB63.7
Área de marcas (M)	M0.0 a M31.7	M0.0 a M31.7	M0.0 a M31.7
Marcas especiales (SM) Sólo lectura	SM0.0 a SM179.7 SM0.0 a SM29.7	SM0.0 a SM179.7 SM0.0 a SM29.7	SM0.0 a SM179.7 SM0.0 a SM29.7
Temporizadores	256 (T0 a T255)	256 (T0 a T255)	256 (T0 a T255)
Retardo a la conexión memorizado 1 ms	T0, T64	T0, T64	T0, T64
Retardo a la conexión memorizado 10 ms	T1 a T4, T65 a T68	T1 a T4, T65 a T68	T1 a T4, T65 a T68
Retardo a la conexión memorizado 100 ms	T5 a T31, T69 a T95	T5 a T31, T69 a T95	T5 a T31, T69 a T95
Retardo a la con./descon. 1 ms	T32, T96	T32, T96	T32, T96
Retardo a la con./descon. 10 ms	T33 a T36, T97 a T100	T33 a T36, T97 a T100	T33 a T36, T97 a T100
Retardo a la con./descon. 100 ms	T37 a T63, T101 a T255	T37 a T63, T101 a T255	T37 a T63, T101 a T255
Contadores	C0 a C255	C0 a C255	C0 a C255
Contadores rápidos	HC0, HC3, HC4, HC5	HC0, HC3, HC4, HC5	HC0 a HC5
Relés de control secuencial (S)	S0.0 a S31.7	S0.0 a S31.7	S0.0 a S31.7
Acumuladores	AC0 a AC3	AC0 a AC3	AC0 a AC3
Saltos a metas	0 a 255	0 a 255	0 a 255
Llamadas a subrutinas	0 a 63	0 a 63	0 a 63
Rutinas de interrupción	0 a 127	0 a 127	0 a 127
Lazos PID	0 a 7	0 a 7	0 a 7
Puerto	Puerto 0	Puerto 0	Puerto 0
<sup>1</sup> Todo el contenido de la memoria V se puede guardar en la memoria no volátil.			
<sup>2</sup> STEP 7-Micro/WIN 32 (versión 3.0 o posterior) reserva LB60 a LB63.			

Tabla 8-3 Áreas de operandos de las CPUs S7-200

Tipo de acceso	CPU 221	CPU 222	CPU 224
Bit (byte.bit)	V 0.0 a 2047.7 I 0.0 a 15.7 Q 0.0 a 15.7 M 0.0 a 31.7 SM 0.0 a 179.7 S 0.0 a 31.7 T 0 a 255 C 0 a 255 L 0.0 a 63.7	V 0.0 a 2047.7 I 0.0 a 15.7 Q 0.0 a 15.7 M 0.0 a 31.7 SM 0.0 a 179.7 S 0.0 a 31.7 T 0 a 255 C 0 a 255 L 0.0 a 63.7	V 0.0 a 5119.7 I 0.0 a 15.7 Q 0.0 a 15.7 M 0.0 a 31.7 SM 0.0 a 179.7 S 0.0 a 31.7 T 0 a 255 C 0 a 255 L 0.0 a 63.7
Byte	VB 0 a 2047 IB 0 a 15 QB 0 a 15 MB 0 a 31 SMB 0 a 179 SB 0 a 31 LB 0 a 63 AC 0 a 3 constante	VB 0 a 2047 IB 0 a 15 QB 0 a 15 MB 0 a 31 SMB 0 a 179 SB 0 a 31 LB 0 a 63 AC 0 a 3 constante	VB 0 a 5119 IB 0 a 15 QB 0 a 15 MB 0 a 31 SMB 0 a 179 SB 0 a 31 LB 0 a 63 AC 0 a 3 constante
Palabra	VW 0 a 2046 IW 0 a 14 QW 0 a 14 MW 0 a 30 SMW 0 a 178 SW 0 a 30 T 0 a 255 C 0 a 255 LW 0 a 62 AC 0 a 3 AIW 0 a 30 AQW 0 a 30 constante	VW 0 a 2046 IW 0 a 14 QW 0 a 14 MW 0 a 30 SMW 0 a 178 SW 0 a 30 T 0 a 255 C 0 a 255 LW 0 a 62 AC 0 a 3 AIW 0 a 30 AQW 0 a 30 constante	VW 0 a 5118 IW 0 a 14 QW 0 a 14 MW 0 a 30 SMW 0 a 178 SW 0 a 30 T 0 a 255 C 0 a 255 LW 0 a 62 AC 0 a 3 AIW 0 a 30 AQW 0 a 30 constante
Palabra doble	VD 0 a 2044 ID 0 a 12 QD 0 a 12 MD 0 a 28 SMD 0 a 176 SD 0 a 28 LD 0 a 60 AC 0 a 3 HC 0, 3, 4, 5 constante	VD 0 a 2044 ID 0 a 12 QD 0 a 12 MD 0 a 28 SMD 0 a 176 SD 0 a 28 LD 0 a 60 AC 0 a 3 HC 0, 3, 4, 5 constante	VD 0 a 5116 ID 0 a 12 QD 0 a 12 MD 0 a 28 SMD 0 a 176 SD 0 a 28 LD 0 a 60 AC 0 a 3 HC 0, 3, 4, 5 constante

# 9

## Operaciones SIMATIC

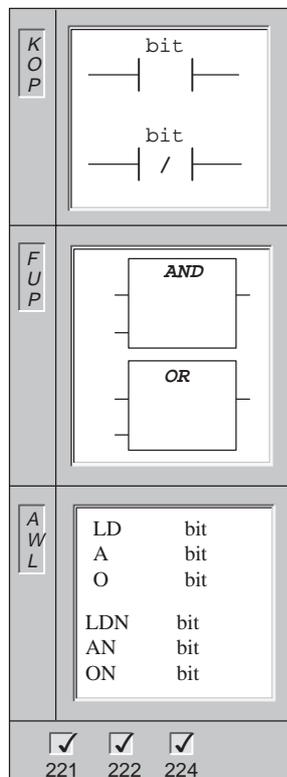
El presente capítulo describe el juego de operaciones SIMATIC para los sistemas de automatización S7-200.

### Índice del capítulo

Apartado	Descripción	Página
9.1	Operaciones lógicas con bits (SIMATIC)	9-2
9.2	Operaciones de comparación (SIMATIC)	9-10
9.3	Operaciones de temporización (SIMATIC)	9-15
9.4	Operaciones con contadores (SIMATIC)	9-23
9.5	Operaciones con contadores rápidos (SIMATIC)	9-27
9.6	Operaciones de salida de impulsos (SIMATIC)	9-49
9.7	Operaciones de reloj (SIMATIC)	9-70
9.8	Operaciones aritméticas con enteros (SIMATIC)	9-72
9.9	Operaciones aritméticas con números reales (SIMATIC)	9-81
9.10	Operaciones de transferencia (SIMATIC)	9-99
9.11	Operaciones de tabla (SIMATIC)	9-104
9.12	Operaciones lógicas (SIMATIC)	9-110
9.13	Operaciones de desplazamiento y rotación (SIMATIC)	9-116
9.14	Operaciones de conversión (SIMATIC)	9-126
9.15	Operaciones de control del programa (SIMATIC)	9-141
9.16	Operaciones de interrupción y comunicación (SIMATIC)	9-165
9.17	Operaciones lógicas de pilas (SIMATIC)	9-192

## 9.1 Operaciones lógicas con bits (SIMATIC)

### Contactos estándar



Estas operaciones leen el valor direccionado de la memoria o de la imagen del proceso si el tipo de datos es I o Q. Para los cuadros AND y OR se pueden utilizar siete entradas como máximo.

El **Contacto normalmente abierto** se cierra (ON) si el bit es igual a 1.

El **Contacto normalmente cerrado** se cierra (ON) si el bit es igual a 0.

En KOP, las operaciones Contacto normalmente abierto y Contacto normalmente cerrado se representan mediante contactos.

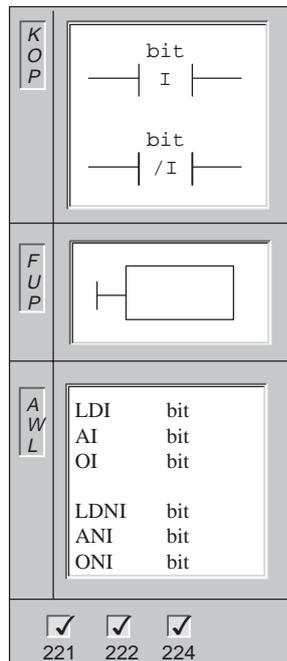
En FUP, los contactos normalmente abiertos se representan mediante cuadros AND/OR. Estas operaciones sirven para manipular señales booleanas de la misma forma que los contactos KOP. Los contactos cerrados también se representan mediante cuadros. Una operación Contacto normalmente cerrado se realiza situando el símbolo de negación en la raíz de la señal de entrada.

En AWL, el contacto normalmente abierto se representa con las operaciones **Cargar, Y** y **O**. Dichas operaciones cargan el valor binario del bit de dirección en el nivel superior de la pila y lo combinan mediante Y u O.

En AWL, el contacto normalmente cerrado se representa con las operaciones **Cargar valor negado, Y-NO** y **O-NO**. Dichas operaciones cargan el valor binario invertido del bit de la dirección en el nivel superior de la pila y lo combinan mediante Y u O.

Entradas/salidas	Operandos	Tipos de datos
Bit (KOP, AWL)	I, Q, M, SM, T, C, V, S, L	BOOL
Entrada (FUP)	I, Q, M, SM, T, C, V, S, L, circulación de corriente	BOOL
Salida (FUP)	I, Q, M, SM, T, C, V, S, L, circulación de corriente	BOOL

## Contactos directos



Estas operaciones leen el valor de la entrada física al ejecutarse la operación, pero la imagen del proceso no se actualiza.

El **Contacto abierto directo** se cierra (se activa) si la entrada física (bit) es 1.

El **Contacto cerrado directo** se cierra (se activa) si la entrada física (bit) es 0.

En KOP, las operaciones Contacto abierto directo y Contacto cerrado directo se representan mediante contactos.

En FUP, la operación Contacto abierto directo se representa mediante un corchete delante del operando. El corchete puede faltar si se usa la circulación de corriente. Esta operación sirve para manipular señales físicas de la misma forma que los contactos KOP.

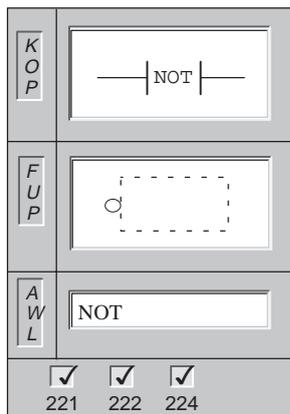
En FUP, la operación Contacto cerrado directo se representa también mediante un corchete y el símbolo de negación delante del operando. El corchete puede faltar si se usa la circulación de corriente. La operación Contacto normalmente cerrado se realiza situando el símbolo de negación en la raíz de la señal de entrada.

En AWL, el contacto abierto directo se representa con las operaciones **Cargar directamente, Y directa** y **O directa**. Estas operaciones cargan directamente el valor de la entrada física en el nivel superior de la pila y lo combinan mediante Y u O.

En AWL, el contacto cerrado directo se representa con las operaciones **Cargar valor negado directamente, Y-NO directa** y **O-NO directa** (ONI). Estas operaciones cargan directamente el valor binario negado de la entrada física en el nivel superior de la pila y lo combinan mediante Y u O.

Entradas/salidas	Operandos	Tipos de datos
Bit (KOP, AWL)	I	BOOL
Entrada (FUP)	I	BOOL

## NOT



El contacto **NOT** invierte el sentido de circulación de la corriente. La corriente se detiene al alcanzar el contacto NOT. Si no logra alcanzar el contacto, entonces hace circular la corriente.

En KOP, la operación **NOT** se representa en forma de contacto.

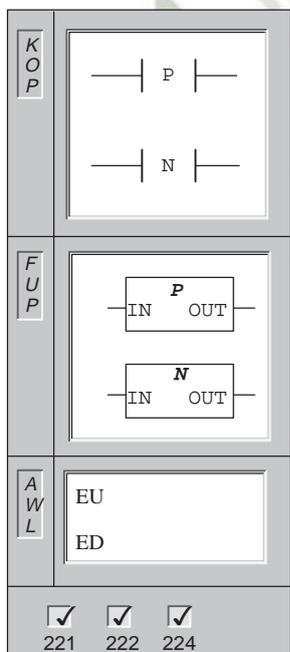
En FUP, la operación **NOT** utiliza el símbolo gráfico de negación con entradas booleanas de cuadro.

En AWL, la operación **Invertir primer valor** (NOT) invierte el primer valor de la pila de 0 a 1, o bien de 1 a 0.

Operandos: ninguno

Tipos de datos: ninguno

## Detectar flanco positivo y negativo



El contacto **Detectar flanco positivo** permite que la corriente circule durante un ciclo cada vez que se produce un cambio de 0 a 1 (de "off" a "on").

El contacto **Detectar flanco negativo** permite que la corriente circule durante un ciclo cada vez que se produce un cambio de señal de 1 a 0 (de "on" a "off").

En KOP, las operaciones Detectar flanco positivo y negativo se representan mediante contactos.

En FUP, dichas operaciones se representan mediante los cuadros P y N.

En AWL, la transición positiva se representa con la operación **Detectar flanco positivo**. Cuando se detecta un cambio de señal de 0 a 1 en el primer valor de la pila, éste se pone a 1. En caso contrario, se pone a 0.

En AWL, la transición negativa se representa con la operación **Detectar flanco negativo**. Cuando se detecta un cambio de señal de 1 a 0 en el primer valor de la pila, éste se pone a 1. En caso contrario, se pone a 0.

Entradas/salidas	Operandos	Tipos de datos
IN (FUP)	I, Q, M, SM, T, C, V, S, L, circulación de corriente	BOOL
OUT (FUP)	I, Q, M, SM, T, C, V, S, L, circulación de corriente	BOOL

Ejemplos de operaciones con contactos

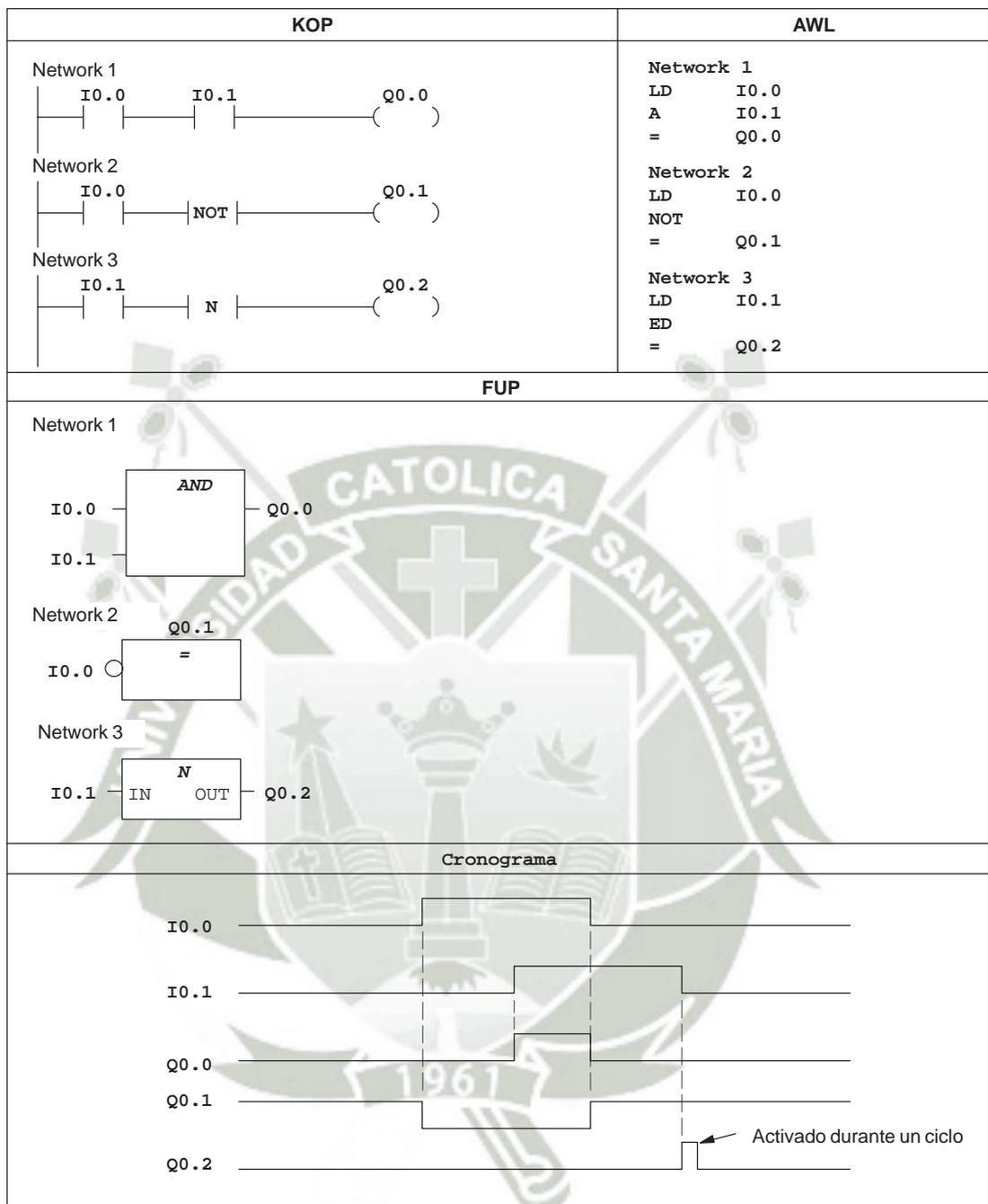
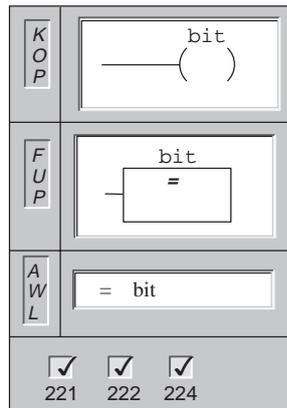


Figura 9-1 Ejemplos de operaciones con contactos booleanos para KOP, AWL y FUP (SIMATIC)

## Asignar



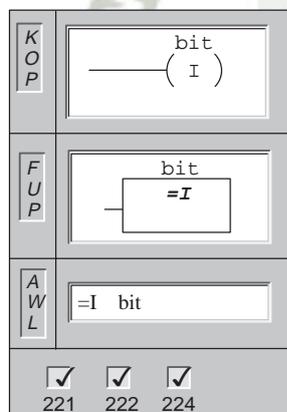
Cuando se ejecuta la operación **Asignar**, el bit de salida se activa en la imagen del proceso.

Cuando la operación Asignar se ejecuta en KOP y FUP, el bit indicado se ajusta de forma equivalente a la circulación de la corriente.

En AWL, la operación Asignar copia el primer valor de la pila en el bit indicado.

Entradas/salidas	Operandos	Tipos de datos
Bit	I, Q, M, SM, T, C, V, S, L	BOOL
Entrada (KOP)	Circulación de corriente	BOOL
Entrada (FUP)	I, Q, M, SM, T, C, V, S, L, circulación de corriente	BOOL

## Asignar directamente



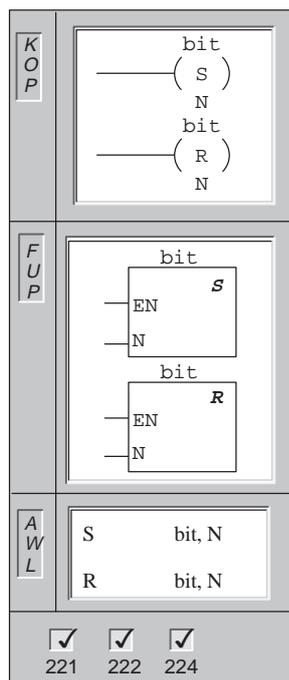
Cuando se ejecuta la operación **Asignar directamente**, la entrada física (bit u OUT) se ajusta de forma equivalente a la circulación de la corriente.

La "I" indica que la operación se ejecuta directamente. El nuevo valor se escribe entonces tanto en la salida física como en la correspondiente dirección de la imagen del proceso. En cambio, en las operaciones no directas, el nuevo valor se escribe sólo en la imagen del proceso.

En AWL, la operación Asignar directamente copia el primer valor de la pila directamente en la salida física indicada (bit).

Entradas/salidas	Operandos	Tipos de datos
Bit	Q	BOOL
Entrada (KOP)	Circulación de corriente	BOOL
Entrada (FUP)	I, Q, M, SM, T, C, V, S, L, circulación de corriente	BOOL

### Poner a 1, Poner a 0 (N bits)



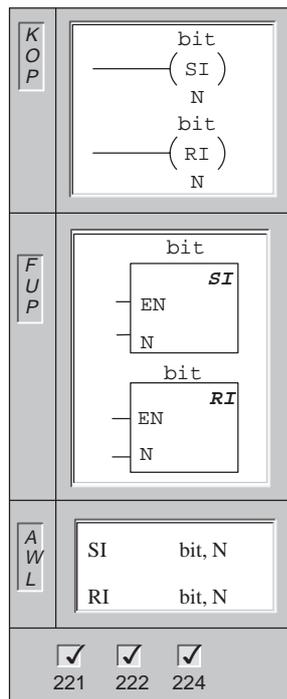
Cuando se ejecutan las operaciones **Poner a 1** y **Poner a 0**, se activa (se pone a 1) o se desactiva (se pone a 0) el número indicado de salidas (N) a partir del valor indicado por el bit o por el parámetro OUT.

El margen de entradas y/o salidas que se pueden activar o desactivar está comprendido entre 1 y 255. Con la operación **Poner a 0**, si el bit indicado es un bit T (bit de temporización) o un bit C (bit de contaje), se desactivará el bit de temporización/contaje y se borrará el valor actual del temporizador/contador.

Condiciones de error que ponen ENO a 0: SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto), 0091 (operando fuera de área)

Entradas/salidas	Operandos	Tipos de datos
Bit	I, Q, M, SM, T, C, V, S, L	BOOL
N	VB, IB, QB, MB, SMB, SB, LB, AC, constante, *VD, *AC, *LD	BYTE

### Poner a 1 directamente, Poner a 0 directamente (N bits)



Cuando se ejecutan las operaciones **Poner a 1 directamente** y **Poner a 0 directamente** se activa (se pone a 1) o se desactiva (se pone a 0) directamente el número indicado de salidas físicas (N) a partir del bit o de OUT.

El margen de entradas y/o salidas que se pueden activar o desactivar está comprendido entre 1 y 128.

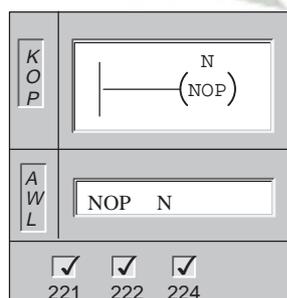
La "I" indica que la operación se ejecuta directamente. Al ejecutarse ésta, el nuevo valor se escribe tanto en la salida física como en la correspondiente dirección de la imagen del proceso. En cambio, en las operaciones no directas, el nuevo valor se escribe sólo en la imagen del proceso.

Condiciones de error que ponen ENO a 0:

SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto), 0091 (operando fuera de área)

Entradas/salidas	Operandos	Tipos de datos
Bit	Q	BOOL
N	VB, IB, QB, MB, SMB, SB, LB, AC, constante, *VD, *AC, *LD	BYTE

### Operación nula



La **Operación nula** no tiene efecto alguno sobre la ejecución del programa. El operando N es un número comprendido entre 0 y 255.

Operandos: N: Constante (0 a 255)

Tipos de datos: BYTE

Ejemplos de operaciones con salidas

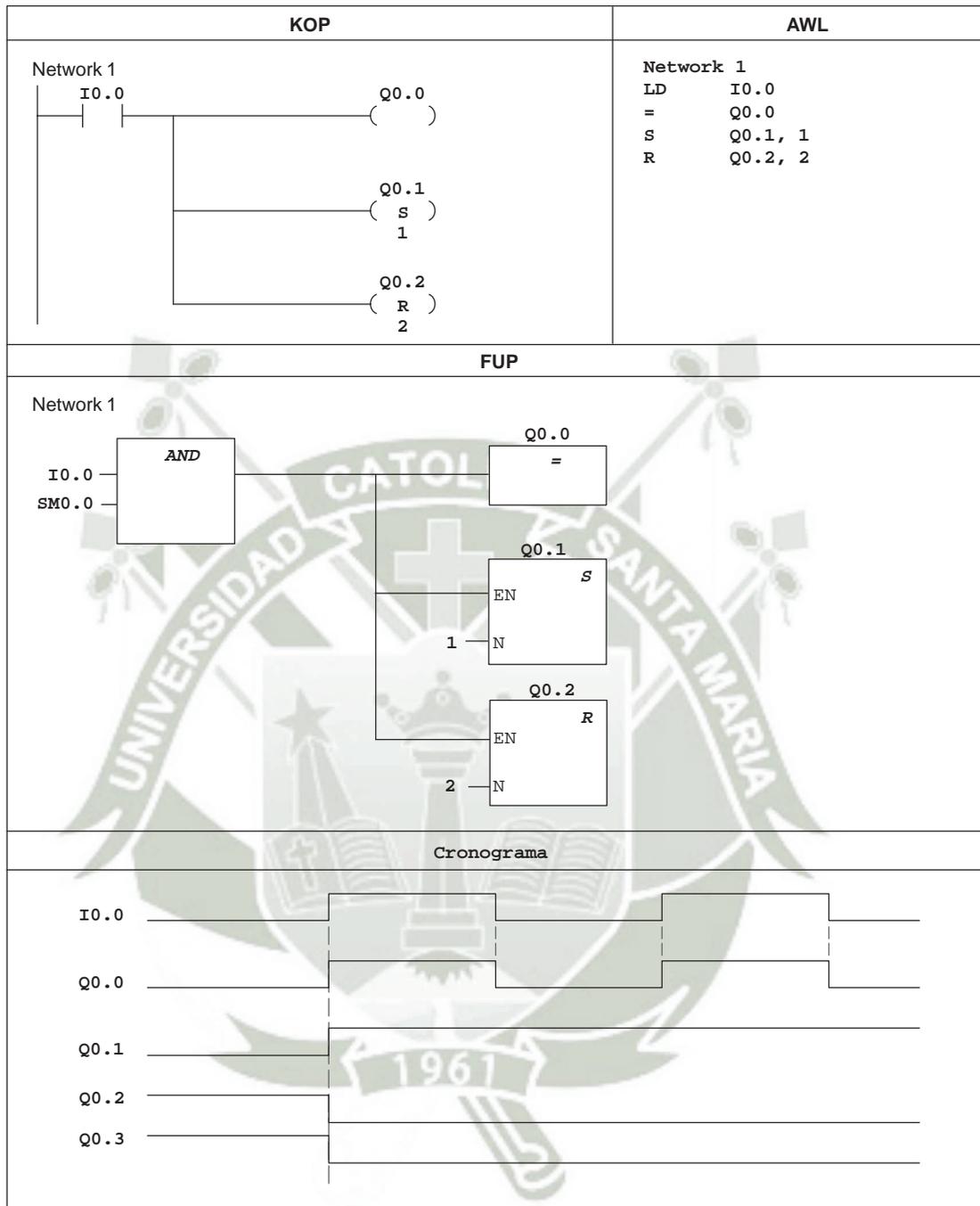


Figura 9-2 Ejemplos de operaciones con salidas en KOP, AWL y FUP (SIMATIC)

## 9.2 Operaciones de comparación (SIMATIC)

### Comparar byte

K O P																																					
F U P																																					
A W L	<table border="0"> <tr><td>LDB=</td><td>IN1, IN2</td></tr> <tr><td>AB=</td><td>IN1, IN2</td></tr> <tr><td>OB=</td><td>IN1, IN2</td></tr> <tr><td>LDB&lt;&gt;</td><td>IN1, IN2</td></tr> <tr><td>AB&lt;&gt;</td><td>IN1, IN2</td></tr> <tr><td>OB&lt;&gt;</td><td>IN1, IN2</td></tr> <tr><td>LDB&lt;</td><td>IN1, IN2</td></tr> <tr><td>AB&lt;</td><td>IN1, IN2</td></tr> <tr><td>OB&lt;</td><td>IN1, IN2</td></tr> <tr><td>LDB&lt;=</td><td>IN1, IN2</td></tr> <tr><td>AB&lt;=</td><td>IN1, IN2</td></tr> <tr><td>OB&lt;=</td><td>IN1, IN2</td></tr> <tr><td>LDB&gt;</td><td>IN1, IN2</td></tr> <tr><td>AB&gt;</td><td>IN1, IN2</td></tr> <tr><td>OB&gt;</td><td>IN1, IN2</td></tr> <tr><td>LDB&gt;=</td><td>IN1, IN2</td></tr> <tr><td>AB&gt;=</td><td>IN1, IN2</td></tr> <tr><td>OB&gt;=</td><td>IN1, IN2</td></tr> </table>	LDB=	IN1, IN2	AB=	IN1, IN2	OB=	IN1, IN2	LDB<>	IN1, IN2	AB<>	IN1, IN2	OB<>	IN1, IN2	LDB<	IN1, IN2	AB<	IN1, IN2	OB<	IN1, IN2	LDB<=	IN1, IN2	AB<=	IN1, IN2	OB<=	IN1, IN2	LDB>	IN1, IN2	AB>	IN1, IN2	OB>	IN1, IN2	LDB>=	IN1, IN2	AB>=	IN1, IN2	OB>=	IN1, IN2
LDB=	IN1, IN2																																				
AB=	IN1, IN2																																				
OB=	IN1, IN2																																				
LDB<>	IN1, IN2																																				
AB<>	IN1, IN2																																				
OB<>	IN1, IN2																																				
LDB<	IN1, IN2																																				
AB<	IN1, IN2																																				
OB<	IN1, IN2																																				
LDB<=	IN1, IN2																																				
AB<=	IN1, IN2																																				
OB<=	IN1, IN2																																				
LDB>	IN1, IN2																																				
AB>	IN1, IN2																																				
OB>	IN1, IN2																																				
LDB>=	IN1, IN2																																				
AB>=	IN1, IN2																																				
OB>=	IN1, IN2																																				
	<input checked="" type="checkbox"/> 221 <input checked="" type="checkbox"/> 222 <input checked="" type="checkbox"/> 224																																				

La operación **Comparar byte** se utiliza para comparar dos valores: IN1 e IN2. Las comparaciones incluyen: IN1 = IN2, IN1 >= IN2, IN1 <= IN2, IN1 > IN2, IN1 < IN2, o IN1 <> IN2.

Las comparaciones de bytes no llevan signo.

En KOP, el contacto se activa si la comparación es verdadera.

En FUP, la salida se activa si la comparación es verdadera.

En AWL, las operaciones cargan un "1" en el nivel superior de la pila si la comparación es verdadera.

Entradas/salidas	Operandos	Tipos de datos
Entradas	IB, QB, MB, SMB, VB, SB, LB, AC, constante, *VD, *AC, *LD	BYTE
Salidas (FUP)	I, Q, M, SM, T, C, V, S, L, circulación de corriente	BOOL

### Comparar entero

K O P	
F U P	
A W L	<p>LDW= IN1, IN2                  AW= IN1, IN2                  OB= IN1, IN2                  LDW&lt;&gt; IN1, IN2                  AW&lt;&gt; IN1, IN2                  OW&lt;&gt; IN1, IN2                  LDW&lt; IN1, IN2                  AW&lt; IN1, IN2                  OW&lt; IN1, IN2                  LDW&lt;= IN1, IN2                  AW&lt;= IN1, IN2                  OW&lt;= IN1, IN2                  LDW&gt; IN1, IN2                  AW&gt; IN1, IN2                  OW&gt; IN1, IN2                  LDW&gt;= IN1, IN2                  AW&gt;= IN1, IN2                  OW&gt;= IN1, IN2</p>
	<input checked="" type="checkbox"/> 221 <input checked="" type="checkbox"/> 222 <input checked="" type="checkbox"/> 224

La operación **Comparar entero** se utiliza para comparar dos valores: IN1 e IN2. Las comparaciones incluyen: IN1 = IN2, IN1 >= IN2, IN1 <= IN2, IN1 > IN2, IN1 < IN2, o IN1 <> IN2.

Las comparaciones de enteros llevan signo (16#7FFF > 16#8000).

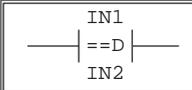
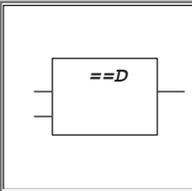
En KOP, el contacto se activa si la comparación es verdadera.

En FUP, la salida se activa si la comparación es verdadera.

En AWL, las operaciones cargan un "1" en el nivel superior de la pila y combinan el valor "1" con el primer valor de la pila mediante Y u O si la comparación es verdadera.

Entradas/salidas	Operandos	Tipos de datos
Entradas	IW, QW, MW, SW, SMW, T, C, VW, LW, AIW, AC, constante, *VD, *AC, *LD	INT
Salidas (FUP)	I, Q, M, SM, T, C, V, S, L, circulación de corriente	BOOL

### Comparar palabra doble

K O P																																					
F U P																																					
A W L	<table border="0"> <tr><td>LDD=</td><td>IN1, IN2</td></tr> <tr><td>AD=</td><td>IN1, IN2</td></tr> <tr><td>OD=</td><td>IN1, IN2</td></tr> <tr><td>LDD&lt;</td><td>IN1, IN2</td></tr> <tr><td>AD&lt;</td><td>IN1, IN2</td></tr> <tr><td>OD&lt;</td><td>IN1, IN2</td></tr> <tr><td>LDD&lt;</td><td>IN1, IN2</td></tr> <tr><td>AD&lt;</td><td>IN1, IN2</td></tr> <tr><td>OD&lt;</td><td>IN1, IN2</td></tr> <tr><td>LDD&lt;=</td><td>IN1, IN2</td></tr> <tr><td>AD&lt;=</td><td>IN1, IN2</td></tr> <tr><td>OD&lt;=</td><td>IN1, IN2</td></tr> <tr><td>LDD&gt;</td><td>IN1, IN2</td></tr> <tr><td>AD&gt;</td><td>IN1, IN2</td></tr> <tr><td>OD&gt;</td><td>IN1, IN2</td></tr> <tr><td>LDD&gt;=</td><td>IN1, IN2</td></tr> <tr><td>AD&gt;=</td><td>IN1, IN2</td></tr> <tr><td>OD&gt;=</td><td>IN1, IN2</td></tr> </table>	LDD=	IN1, IN2	AD=	IN1, IN2	OD=	IN1, IN2	LDD<	IN1, IN2	AD<	IN1, IN2	OD<	IN1, IN2	LDD<	IN1, IN2	AD<	IN1, IN2	OD<	IN1, IN2	LDD<=	IN1, IN2	AD<=	IN1, IN2	OD<=	IN1, IN2	LDD>	IN1, IN2	AD>	IN1, IN2	OD>	IN1, IN2	LDD>=	IN1, IN2	AD>=	IN1, IN2	OD>=	IN1, IN2
LDD=	IN1, IN2																																				
AD=	IN1, IN2																																				
OD=	IN1, IN2																																				
LDD<	IN1, IN2																																				
AD<	IN1, IN2																																				
OD<	IN1, IN2																																				
LDD<	IN1, IN2																																				
AD<	IN1, IN2																																				
OD<	IN1, IN2																																				
LDD<=	IN1, IN2																																				
AD<=	IN1, IN2																																				
OD<=	IN1, IN2																																				
LDD>	IN1, IN2																																				
AD>	IN1, IN2																																				
OD>	IN1, IN2																																				
LDD>=	IN1, IN2																																				
AD>=	IN1, IN2																																				
OD>=	IN1, IN2																																				
	<input checked="" type="checkbox"/> 221 <input checked="" type="checkbox"/> 222 <input checked="" type="checkbox"/> 224																																				

La operación **Comparar palabra doble** se utiliza para comparar dos valores: IN1 e IN2. Las comparaciones incluyen: IN1 = IN2, IN1 >= IN2, IN1 <= IN2, IN1 > IN2, IN1 < IN2, o IN1 <> IN2.

Las comparaciones de palabras dobles llevan signo (16#7FFFFFFF > 16#80000000).

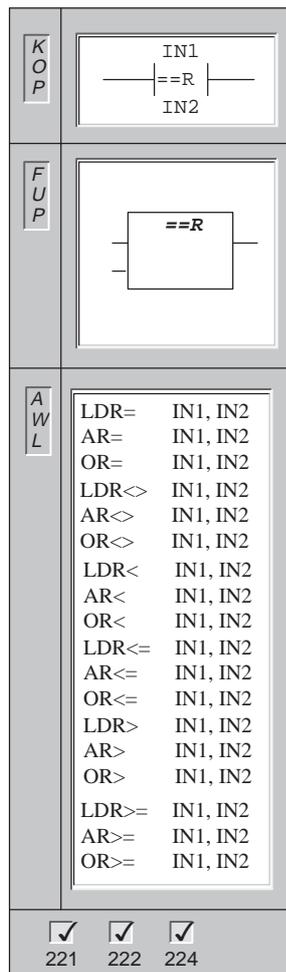
En KOP, el contacto se activa si la comparación es verdadera.

En FUP, la salida se activa si la comparación es verdadera.

En AWL, las operaciones cargan un "1" en el nivel superior de la pila y combinan el valor "1" con el primer valor de la pila mediante Y u O si la comparación es verdadera.

Entradas/salidas	Operandos	Tipos de datos
Entradas	ID, QD, MD, SD, SMD, VD, LD, HC, AC, constante, *VD, *AC, *LD	DINT
Salidas (FUP)	I, Q, M, SM, T, C, V, S, L, circulación de corriente	BOOL

### Comparar real



La operación **Comparar real** se utiliza para comparar dos valores: IN1 e IN2. Las comparaciones incluyen: IN1 = IN2, IN1 >= IN2, IN1 <= IN2, IN1 > IN2, IN1 < IN2, o IN1 <> IN2.

Las comparaciones de números reales llevan signo.

En KOP, el contacto se activa si la comparación es verdadera.

En FUP, la salida se activa si la comparación es verdadera.

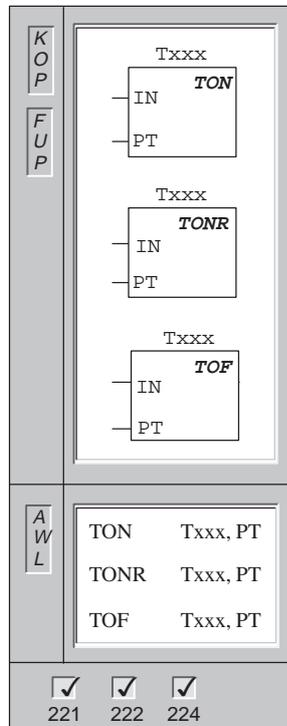
En AWL, las operaciones cargan un "1" en el nivel superior de la pila y combinan el valor "1" con el primer valor de la pila mediante Y u O si la comparación es verdadera.

Entradas/salidas	Operandos	Tipos de datos
Entradas	ID, QD, MD, SD, SMD, VD, LD, AC, constante, *VD, *AC, *LD	REAL
Salidas (FUP)	I, Q, M, SM, T, C, V, S, L, circulación de corriente	BOOL



### 9.3 Operaciones de temporización (SIMATIC)

#### Temporizador de retardo a la conexión, Temporizador de retardo a la conexión memorizado, Temporizador de retardo a la desconexión



Las operaciones **Temporizador de retardo a la conexión** y **Temporizador de retardo a la conexión memorizado** cuentan el tiempo al estar activada (ON) la entrada de habilitación. Si el valor actual (Txxx) es mayor o igual al valor de preselección (PT), se activa el bit de temporización (bit T).

Cuando la entrada de habilitación está desconectada (OFF), el valor actual se borra en el caso del temporizador de retardo a la conexión. En cambio, se conserva en el temporizador de retardo a la conexión memorizado. Éste último sirve para acumular varios períodos de tiempo de la entrada en ON. Para borrar el valor actual del temporizador de retardo a la conexión memorizado se utiliza la operación Poner a 0 (R).

Tanto el temporizador de retardo a la conexión como el temporizador de retardo a la conexión memorizado continúan contando tras haberse alcanzado el valor de preselección y paran de contar al alcanzar el valor máximo de 32767.

El **Temporizador de retardo a la desconexión** se utiliza para retardar la puesta a 0 (OFF) de una salida durante un período determinado tras haberse desactivado (OFF) una entrada. Cuando la entrada de habilitación se activa (ON), el bit de temporización se activa (ON) inmediatamente y el valor actual se pone a 0. Cuando la entrada se desactiva (OFF), el temporizador cuenta hasta que el tiempo transcurrido alcance el valor de preselección. Una vez alcanzado éste, el bit de temporización se desactiva (OFF) y el valor actual detiene el conteo. Si la entrada está desactivada (OFF) durante un tiempo inferior al valor de preselección, el bit de temporización permanece activado (ON). Para que la operación TOF comience a contar se debe producir un cambio de ON a OFF.

Si un temporizador TOF está dentro de una sección SCR y ésta se encuentra desactivada, el valor actual se pone a 0, el bit de temporización se desactiva (OFF) y el valor actual no cuenta.

Entradas/salidas	Operandos	Tipos de datos
IN (KOP)	Circulación de corriente	BOOL
IN (FUP)	I, Q, M, SM, T, C, V, S, L, circulación de corriente	BOOL
PT	VW, IW, QW, MW, SW, SMW, LW, AIW, T, C, AC, constante, *VD, *AC, *LD	INT

Se dispone de temporizadores TON, TONR y TOF con tres resoluciones. La resolución viene determinada por el número del temporizador (v. tabla 9-1). El valor actual resulta del valor de contaje multiplicado por la base de tiempo. Por ejemplo, el valor de contaje 50 en un temporizador de 10 ms equivale a 500 ms.

Tabla 9-1 Temporizadores y sus resoluciones

Tipo de temporizador	Resolución en milisegundos (ms)	Valor máximo en segundos (s)	Nº de temporizador
TONR	1 ms	32,767 s	T0, T64
	10 ms	327,67 s	T1 a T4, T65 a T68
	100 ms	3276,7 s	T5 a T31, T69 a T95
TON, TOF	1 ms	32,767 s	T32, T96
	10 ms	327,67 s	T33 a T36, T97 a T100
	100 ms	3276,7 s	T37 a T63, T101 a T255

**Nota**

No se pueden compartir números iguales para los temporizadores TOF y TON. Por ejemplo, no puede haber tanto un TON T32 como un TOF T32.



## Operaciones de temporización del S7-200

Es posible utilizar temporizadores para implementar funciones controladas por tiempo. El juego de operaciones S7-200 ofrece tres tipos de temporizadores como se indica a continuación. La tabla 9-2 muestra las acciones de los diferentes temporizadores.

- Temporizador de retardo a la conexión (TON) para temporizar un solo intervalo.
- Temporizador de retardo a la conexión memorizado (TONR) para acumular varios intervalos temporizados.
- Temporizador de retardo a la desconexión (TOF) para ampliar el tiempo después de un cambio a "falso" (por ejemplo, para enfriar un motor tras haber sido desconectado).

Tabla 9-2 Acciones de los temporizadores

Tipo de temporizador	Actual $\geq$ Preselección	Entrada de habilitación ON	Entrada de habilitación OFF	Alimentación/ primer ciclo
TON	Bit de temporización ON. El valor actual continúa contando hasta 32.767.	El valor actual cuenta el tiempo.	Bit de temporización OFF. Valor actual = 0	Bit de temporización OFF. Valor actual = 0
TONR	Bit de temporización ON. El valor actual continúa contando hasta 32.767.	El valor actual cuenta el tiempo.	El bit de temporización y el valor actual conservan el último estado.	Bit de temporización OFF. El valor actual se puede conservar. <sup>1</sup>
TOF	Bit de temporización OFF. Valor actual = valor de preselección, se detiene el conteo.	Bit de temporización ON. Valor actual = 0	El temporizador cuenta tras un cambio de ON a OFF.	Bit de temporización OFF. Valor actual = 0

<sup>1</sup> El valor actual del temporizador de retardo a la conexión memorizado se selecciona para que quede memorizado cuando se interrumpe la alimentación. Consulte el apartado 5.3 para obtener información sobre el respaldo de la memoria de la CPU S7-200.

### Nota

La operación Poner a 0 (R) sirve para inicializar cualquier temporizador. El temporizador TONR sólo se puede inicializar mediante la operación Poner a 0. Dicha operación arroja los siguientes resultados:

Bit de temporización = OFF.  
Valor actual = 0

Tras inicializarse un temporizador TOF, la entrada de habilitación debe cambiar de ON a OFF para poder rearmar el temporizador.

A continuación se explican las actividades de los temporizadores con diferentes resoluciones.

### **Resolución de 1 milisegundo**

Los temporizadores con resolución de 1 ms cuentan el número de intervalos de 1 ms que han transcurrido desde que se habilitó el temporizador activo de 1 ms. La temporización arranca al comenzarse a ejecutar la operación. No obstante, los temporizadores de 1 ms se actualizan (tanto el bit de temporización como el valor actual) cada milisegundo de forma asíncrona al ciclo. En otras palabras, el bit de temporización y el valor actual se actualizan varias veces en un ciclo que dure más de 1 ms.

La operación de temporización se utiliza para activar e inicializar el temporizador o, en el caso del temporizador de retardo a la conexión memorizado (TONR), para desactivarlo.

Puesto que el temporizador puede arrancar en cualquier momento durante un milisegundo, el valor de preselección se debe ajustar a un intervalo de tiempo que exceda el intervalo mínimo deseado. Por ejemplo, para garantizar un intervalo mínimo de 56 ms utilizando un temporizador de 1 ms, es preciso ajustar el valor de preselección a 57.

### **Resolución de 10 milisegundos**

Los temporizadores con resolución de 10 ms cuentan el número de intervalos de 10 ms que han transcurrido desde que se habilitó el temporizador activo de 10 ms. La temporización arranca al comenzarse a ejecutar la operación. No obstante, los temporizadores de 10 ms se actualizan al comienzo de cada ciclo (en otras palabras, el valor actual y el bit de temporización permanecen constantes durante el ciclo), sumando el número acumulado de intervalos de 10 ms (desde el comienzo del ciclo anterior) al valor actual del temporizador activo.

Puesto que el temporizador puede arrancar en cualquier momento durante un intervalo de 10 ms, el valor de preselección se debe ajustar a un intervalo de tiempo que exceda el intervalo mínimo deseado. Por ejemplo, para garantizar un intervalo mínimo de 140 ms utilizando un temporizador de 10 ms, es preciso ajustar el valor de preselección a 15.

### **Resolución de 100 milisegundos**

Los temporizadores con resolución de 100 ms cuentan el número de intervalos de 100 ms que han transcurrido desde la última vez que se actualizó el temporizador activo de 100 ms. Estos temporizadores se actualizan sumando el valor acumulado de intervalos de 100 ms (desde el ciclo anterior) al valor actual del temporizador cuando se ejecuta la operación del mismo.

El valor actual de un temporizador de 100 ms se actualiza únicamente si se ha ejecutado la operación correspondiente. Por consiguiente, si un temporizador de 100 ms está habilitado, pero la correspondiente operación no se ejecuta en cada ciclo, no se actualizará el valor actual de ese temporizador y disminuirá el tiempo. Por otra parte, si se ejecuta una misma operación con un temporizador de 100 ms varias veces en un ciclo, el valor de 100 ms acumulado se sumará también varias veces al valor actual del temporizador, con lo cual se prolonga el tiempo. Debido a ello, es recomendable utilizar los temporizadores con una resolución de 100 ms sólo cuando se ejecute exactamente una operación de temporización en cada ciclo.

Puesto que el temporizador puede arrancar en cualquier momento durante un intervalo de 100 ms, el valor de preselección se debe ajustar a un intervalo de tiempo que exceda el intervalo mínimo deseado. Por ejemplo, para garantizar un intervalo mínimo de 2.100 ms utilizando un temporizador de 100 ms, es preciso ajustar el valor de preselección a 22.

### Actualizar el valor actual de un temporizador

El efecto de las diferentes maneras de actualizar el valor actual de los temporizadores depende de cómo se utilicen los mismos. Consideremos p.ej. la operación de temporización que muestra la figura 9-4.

- Si se utiliza un temporizador con una resolución de 1 ms, Q0.0 se activará durante un ciclo, siempre que el valor actual del temporizador se actualice tras ejecutarse el contacto cerrado T32 y antes de ejecutarse el contacto normalmente abierto T32.
- Si se utiliza un temporizador con una resolución de 10 ms, Q0.0 no se activará nunca, porque el bit de temporización T33 permanece activado desde el principio del ciclo hasta que se ejecute el cuadro del temporizador. Una vez ejecutado éste, se ponen a 0 el valor actual del temporizador y su bit T. Tras ejecutarse el contacto normalmente abierto T33, se desactivarán T33 y Q0.0.
- Si se utiliza un temporizador con una resolución de 100 ms, Q0.0 se activará durante un ciclo, siempre que el valor actual del temporizador alcance el valor de preselección.

Si en vez del bit de temporización se utiliza el contacto normalmente cerrado Q0.0 como entrada de habilitación para el cuadro del temporizador, la salida Q0.0 quedará activada durante un ciclo cada vez que el valor del temporizador alcance el valor de preselección.

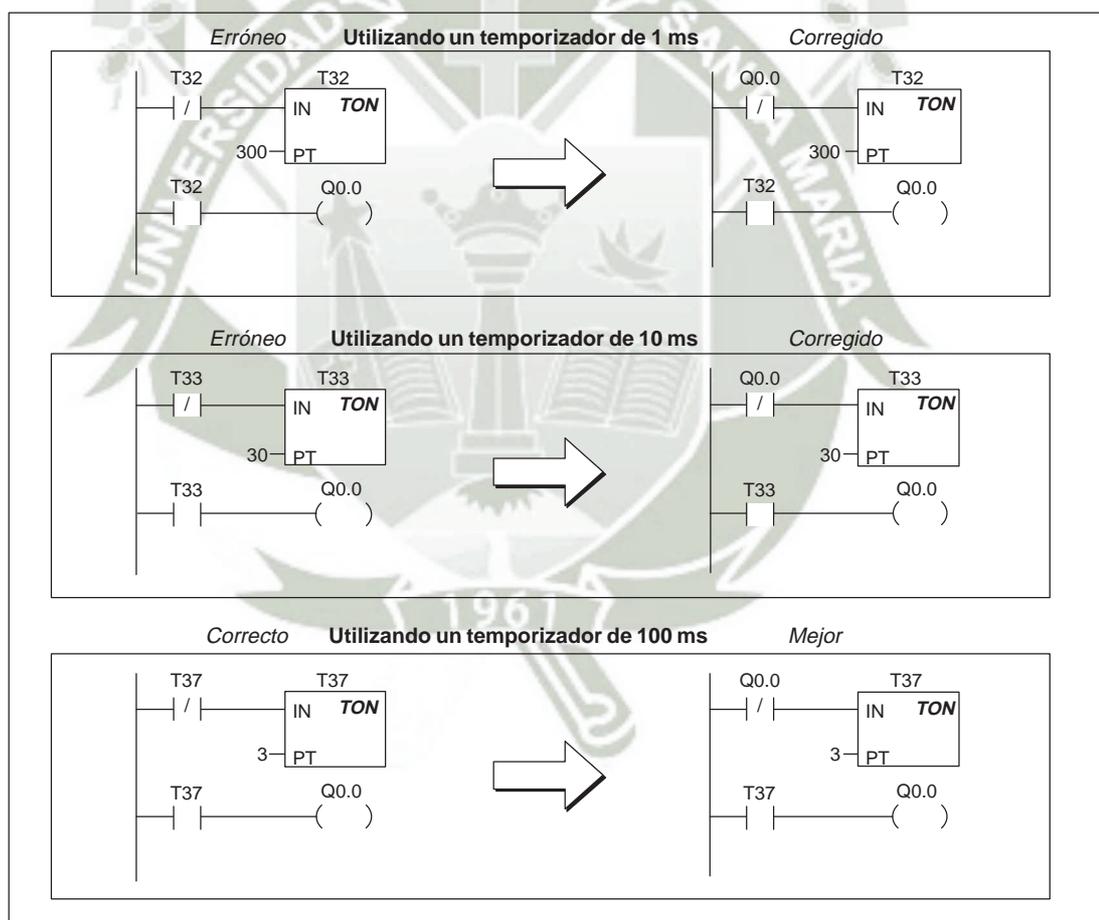


Figura 9-4 Ejemplo del redisparo automático de un temporizador

### Ejemplo de un temporizador de retardo a la conexión

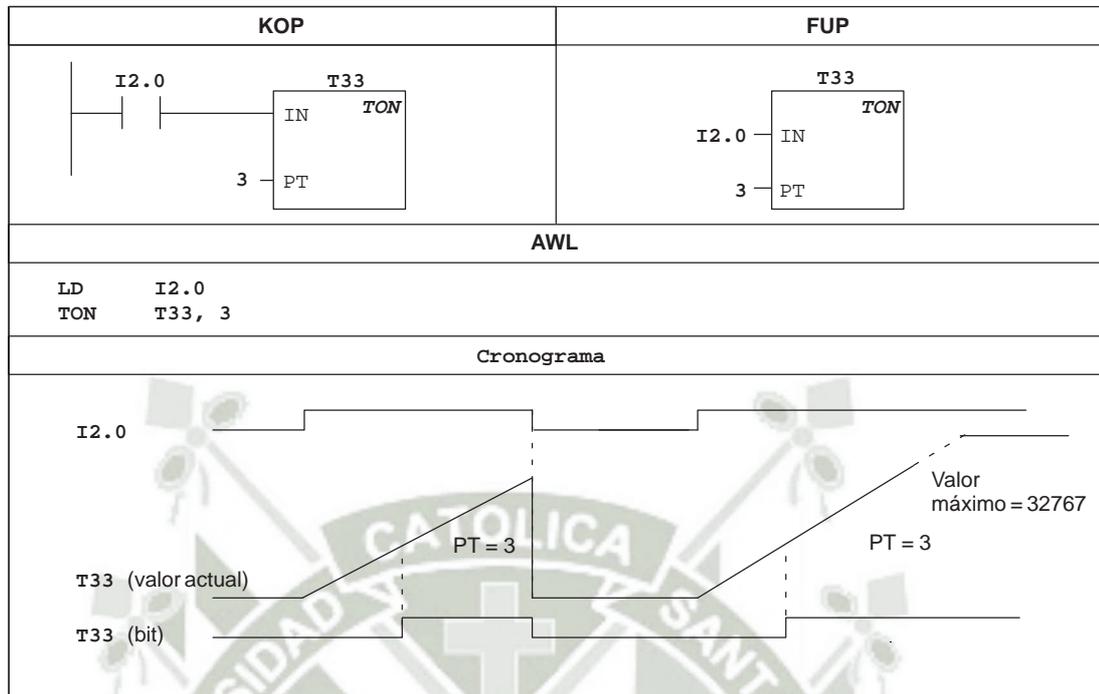


Figura 9-5 Ejemplo de la operación Temporizador de retardo a la conexión en KOP, FUP y AWL



Ejemplo de un temporizador de retardo a la conexión memorizado

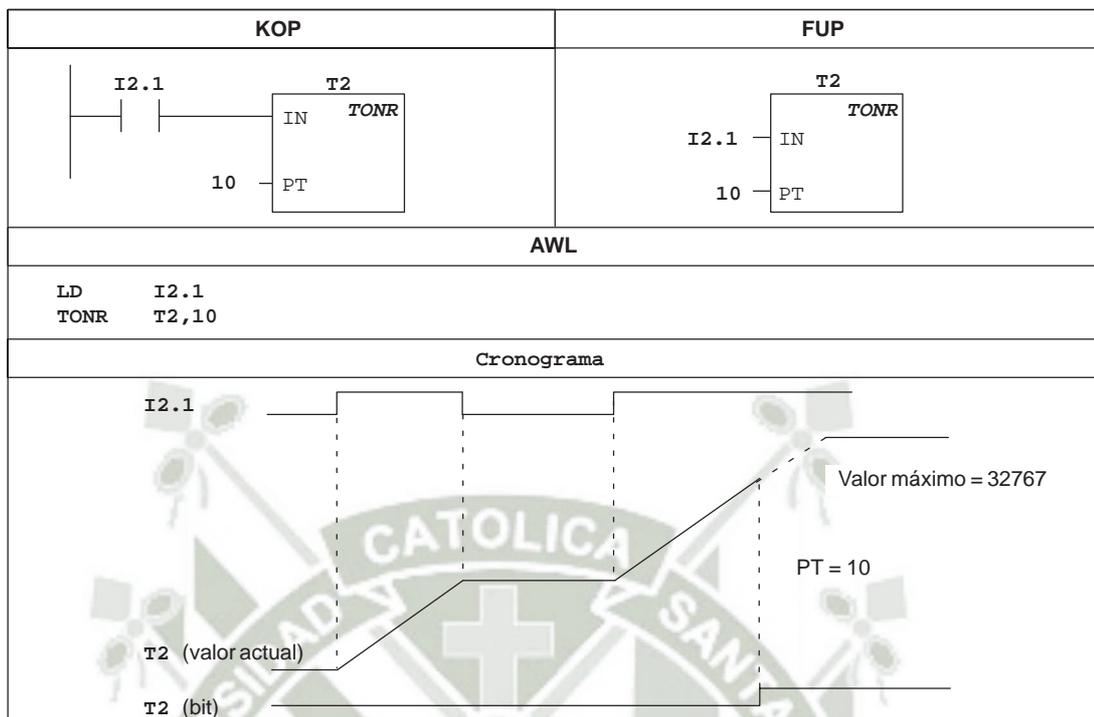


Figura 9-6 Ejemplo de un temporizador de retardo a la conexión memorizado en KOP, FUP y AWL

### Ejemplo de un temporizador de retardo a la desconexión

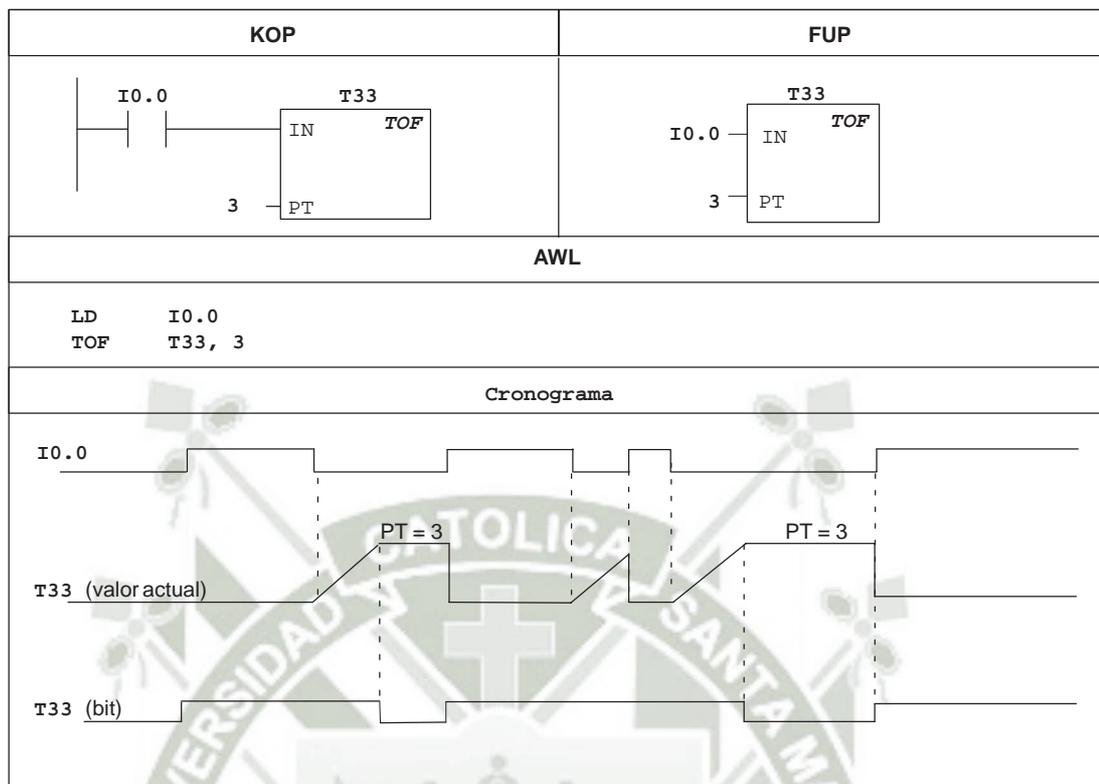
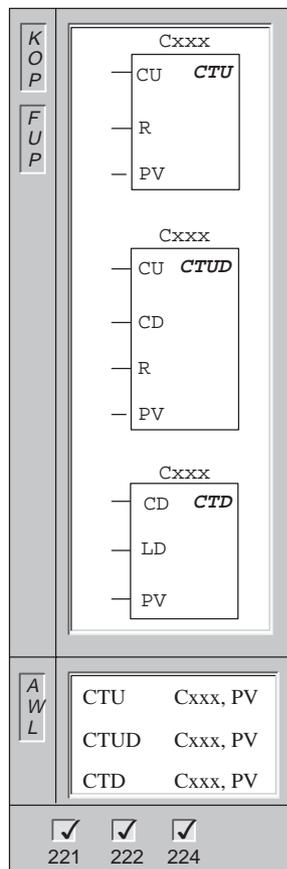


Figura 9-7 Ejemplo de la operación Temporizador de retardo a la desconexión en KOP, FUP y AWL

## 9.4 Operaciones con contadores (SIMATIC)

### Contar adelante, Contar adelante/atrás, Contar atrás



La operación **Contar adelante** empieza a contar hasta el valor máximo cuando se produce un flanco positivo en la entrada de contaje adelante (CU). Si el valor actual (Cxxx) es mayor o igual al valor de preselección (PV), se activa el bit de contaje (Cxxx). El contador se inicializa al activarse la entrada de desactivación (R).

La operación **Contar adelante/atrás** empieza a contar adelante cuando se produce un flanco positivo en la entrada de contaje adelante (CU). Por el contrario, empieza a contar atrás cuando se produce un flanco positivo en la entrada de contaje atrás (CD). Si el valor actual (Cxxx) es mayor o igual al valor de preselección (PV), se activa el bit de contaje (Cxxx). El contador se inicializa al activarse la entrada de desactivación (R).

La operación **Contar atrás** empieza a contar atrás desde el valor de preselección cuando se produce un flanco positivo en la entrada de contaje atrás (CD). Si el valor actual es igual a cero, se activa el bit de contaje (Cxxx). El contador desactiva el bit de contaje (Cxxx) y carga el valor actual con el valor de preselección (PV) cuando se activa la entrada de carga (LD). El contador atrás se detiene al alcanzar el valor cero.

Márgenes de contaje: Cxxx=C0 hasta C255

En AWL, la entrada de desactivación CTU es el primer valor de la pila y la entrada de contaje adelante se carga en el segundo nivel de la pila.

En AWL, la entrada de desactivación CTUD es el primer valor de la pila, la entrada de contaje atrás se carga en el segundo nivel de la pila y la entrada de contaje adelante, en el tercer nivel.

En AWL, la entrada de carga CTD es el primer nivel de la pila y la entrada de contaje atrás es el valor cargado en el segundo nivel de la pila.

Entradas/salidas	Operandos	Tipos de datos
CU, CD (FUP)	I, Q, M, SM, T, C, V, S, L, circulación de corriente	BOOL
R, LD (FUP)	I, Q, M, SM, T, C, V, S, L, circulación de corriente	BOOL
PV	VW, IW, QW, MW, SMW, LW, AIW, AC, T, C, constante, *VD, *AC, *LD, SW	INT

## Descripción de las operaciones de contaje del S7-200

La operación Contar adelante (CTU) empieza a contar adelante a partir del valor actual cuando se produce un flanco positivo en la entrada de contaje adelante. El contador se inicializa cuando se activa la entrada de desactivación o al ejecutarse la operación Poner a 0. El contador para de contar cuando se alcanza el valor máximo (32.767).

La operación Contar adelante/atrás (CTUD) empieza a contar adelante cuando se produce un flanco positivo en la entrada de contaje adelante, y empieza a contar atrás cuando se produce un flanco positivo en la entrada de contaje atrás. El contador se inicializa cuando se activa la entrada de desactivación o al ejecutarse la operación Poner a 0. Cuando se alcanza el valor máximo (32.767), el siguiente flanco positivo en la entrada de contaje adelante invertirá el contaje hasta alcanzar el valor mínimo (-32.768). Igualmente, cuando se alcanza el valor mínimo (-32.768), el siguiente flanco positivo en la entrada de contaje atrás invertirá el contaje hasta alcanzar el valor máximo (32.767).

Los contadores Contar adelante y Contar adelante/atrás tienen un valor actual que almacena el valor de contaje actual. También disponen de un valor de preselección (PV) que se compara con el valor actual cuando se ejecuta la operación de contaje. Si el valor actual es mayor o igual al valor de preselección, se activa el bit de contaje (bit C). En caso contrario, dicho bit se desactiva.

La operación Contar atrás empieza a contar atrás a partir del valor actual cuando se produce un flanco negativo en la entrada de contaje adelante. El contador desactiva el bit de contaje y carga el valor actual con el valor de preselección cuando se activa la entrada de carga. El contador se detiene al alcanzar el valor cero y el bit de contaje (bit C) se activa.

Cuando se inicializa un contador con la operación Poner a 0, se desactivan tanto el bit de contaje como el valor actual del contador. El número del contador se debe utilizar para direccionar tanto el valor actual como el bit C de dicho contador.

---

### Nota

Puesto que cada contador dispone sólo de un valor actual, no se podrá asignar un mismo número a varios contadores. (Los contadores Contar adelante, Contar adelante/atrás y Contar atrás acceden a un mismo valor actual).

---

Ejemplos de una operación de contaje

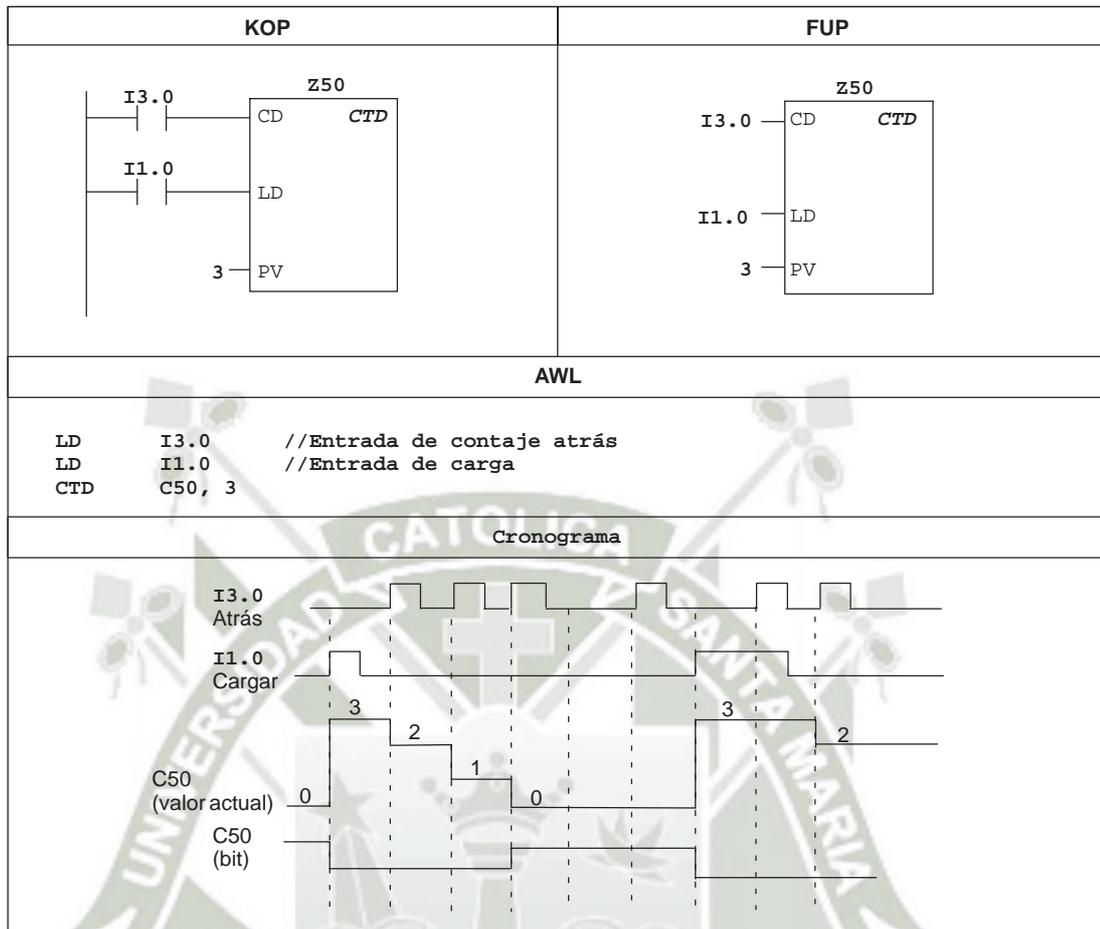


Figura 9-8 Ejemplo de una operación de contaje atrás en KOP, FUP y AWL

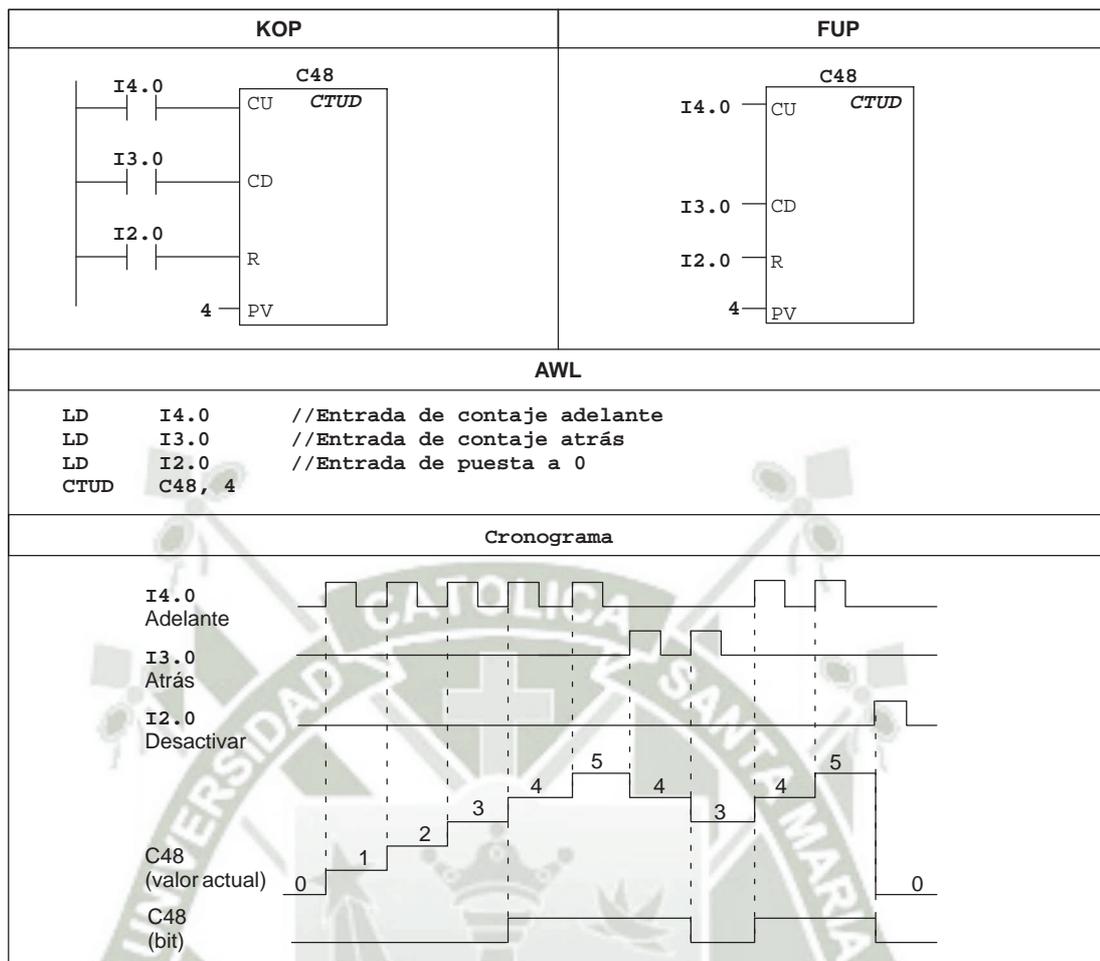
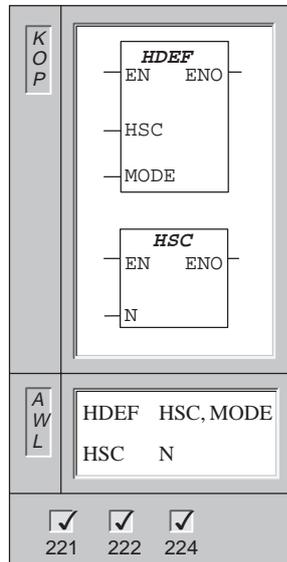


Figura 9-9 Ejemplo de una operación de contaje adelante/atrás en KOP, FUP y AWL

## 9.5 Operaciones con contadores rápidos (SIMATIC)

### Definir modo para contador rápido, Activar contador rápido



La operación **Definir modo para contador rápido** asigna un modo (MODE) al contador rápido direccionado (HSC). Consulte la tabla 9-5.

La operación **Activar contador rápido** configura y controla el funcionamiento del contador rápido direccionado, basándose en el estado de las marcas especiales del mismo. El parámetro N indica el número del contador rápido.

Las CPUs 221 y 222 no asisten los contadores HSC1 y HSC2.

Por cada contador rápido puede utilizarse un solo cuadro HDEF.

HDEF: Condiciones de error que ponen ENO a 0:

SM4.3 (tiempo de ejecución), 0003 (conflicto de entradas), 0004 (operación no válida en interrupción), 000A (redefinición de HSC)

HSC: Condiciones de error que ponen ENO a 0:

SM4.3 (tiempo de ejecución), 0001 (HSC antes de HDEF), 0005 (operaciones HSC/PLS simultáneas)

Entradas/salidas	Operandos	Tipos de datos
HSC	constante	BYTE
MODE	constante	BYTE
N	constante	WORD

### Descripción de las operaciones con contadores rápidos

Los contadores rápidos cuentan eventos que se ejecutan más deprisa de lo que es posible controlarlos en los ciclos de la CPU. Dichos contadores se pueden configurar como máximo para 12 modos de operación diferentes. La tabla 9-5 muestra los modos de los contadores. La frecuencia máxima de un contador rápido depende del tipo de CPU. Para obtener más información acerca de la CPU utilizada, consulte el Anexo A.

Cada contador dispone de entradas que asisten funciones tales como relojes, control del sentido, puesta a 0 y arranque. Para los contadores de dos fases, ambos relojes pueden funcionar a máxima frecuencia. En el caso de los contadores A/B, se puede elegir entre una velocidad máxima de contaje simple (1x) o cuádruple (4x). Todos los contadores funcionan a velocidades máximas sin interferirse mutuamente.

### Utilización de los contadores rápidos

Los contadores rápidos se utilizan habitualmente como accionamiento para temporizadores que funcionan impulsados por un árbol que gira a un régimen constante y provisto de un encoder incremental. Éste último provee un número determinado de valores de conteo por giro, así como un impulso de puesta a 0 una vez por giro. El reloj (o relojes) y el impulso de puesta a 0 del encoder suministran las entradas para el contador rápido. El primero de los valores predeterminados se carga en el contador y las salidas deseadas se activan para el intervalo de tiempo en que el valor actual del contador es menor que el valor predeterminado. El contador se ajusta para que una interrupción se active cuando el conteo actual sea igual al predeterminado o cuando el contador se ponga a 0.

Cuando el valor actual es igual al predeterminado y se presenta un evento de interrupción, entonces se carga un nuevo valor predeterminado y se activa el siguiente estado de señal para las salidas. Si se produce un evento de interrupción porque el contador se ha inicializado, entonces se ajusta el primer valor predeterminado y los primeros estados de las salidas, repitiéndose el ciclo.

Puesto que las interrupciones se producen a una velocidad muy inferior a la de los contadores rápidos, es posible implementar un control preciso de las operaciones rápidas con un impacto relativamente bajo en el ciclo total del sistema de automatización. La posibilidad de asociar interrupciones a rutinas de interrupción permite cargar nuevos valores predeterminados en una rutina de interrupción separada, lo cual simplifica el control del estado, obteniéndose además un programa muy rectilíneo y fácil de leer. Obviamente, todos los eventos de interrupción se pueden ejecutar también en una sola rutina de interrupción. Para obtener más información a este respecto, consulte el apartado 9.16.

### Descripción de los cronogramas de los contadores rápidos

Los siguientes cronogramas (figura 9-10 hasta figura 9-16) muestran cómo cada contador funciona conforme a su categoría. El funcionamiento de las entradas de puesta a 0 y de arranque se representa en dos diagramas por separado, siendo aplicable a todos los modos que utilizan dichas entradas. En los diagramas de las entradas de puesta a 0 y de arranque se ha programado la actividad alta para ambas entradas.

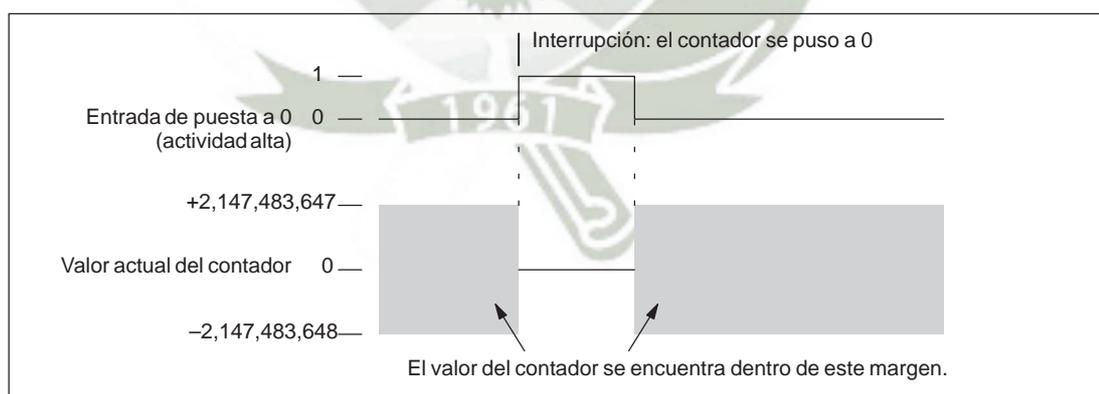


Figura 9-10 Ejemplo del funcionamiento con puesta a 0 y sin arranque

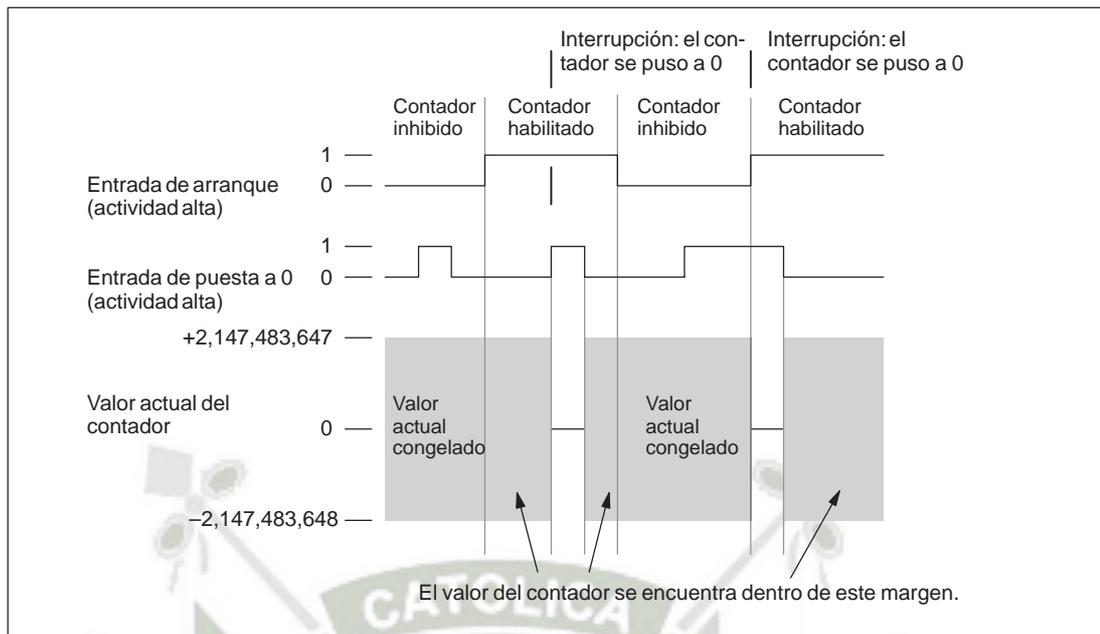


Figura 9-11 Ejemplo del funcionamiento con puesta a 0 y arranque

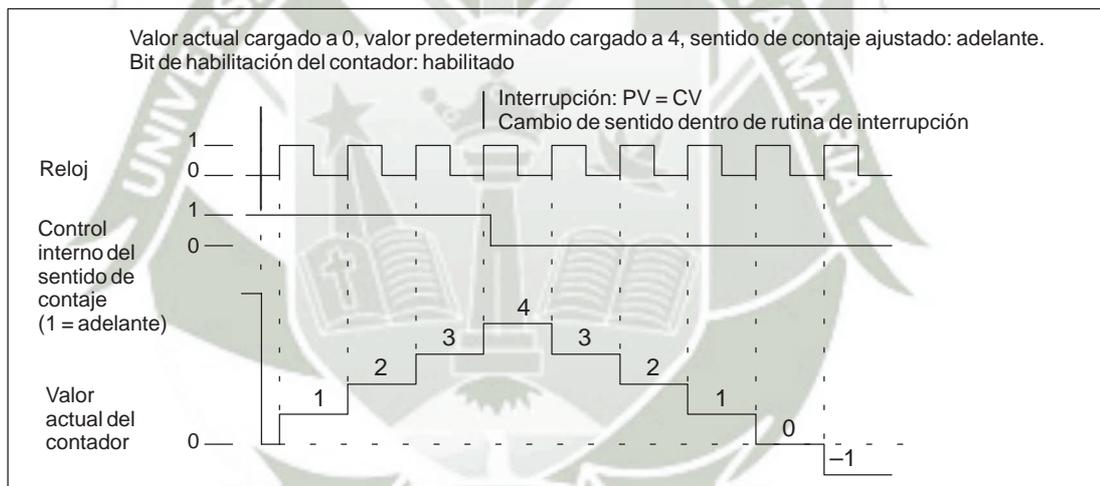


Figura 9-12 Ejemplo del funcionamiento de los modos 0, 1 ó 2

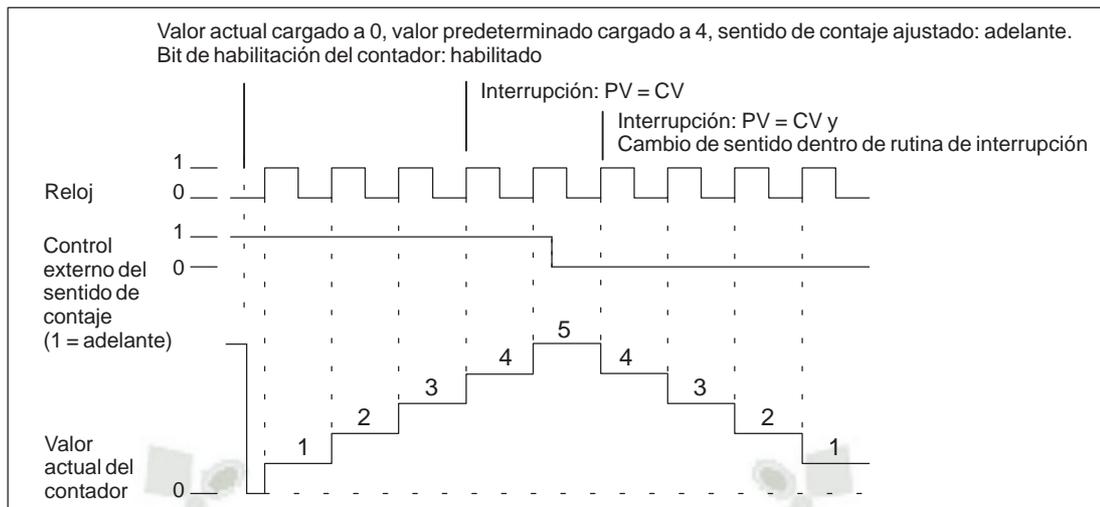


Figura 9-13 Ejemplo del funcionamiento de modos 3, 4 ó 5

Si se utilizan los modos de conteo 6, 7 u 8 y ocurre un flanco positivo tanto en la entrada de reloj de conteo adelante como en la de conteo atrás en menos de 0,3 microsegundos de diferencia, puede ser que el contador rápido considere simultáneos ambos eventos. En este caso, el valor actual permanecerá inalterado y tampoco cambiará el sentido de conteo. Si entre el flanco positivo de la entrada de conteo adelante y el flanco positivo de la entrada de conteo atrás transcurren más de 0,3 microsegundos, el contador rápido recibirá ambos eventos por separado. En ninguno de ambos casos se produce un error (v. figuras 9-14, 9-15 y 9-16).

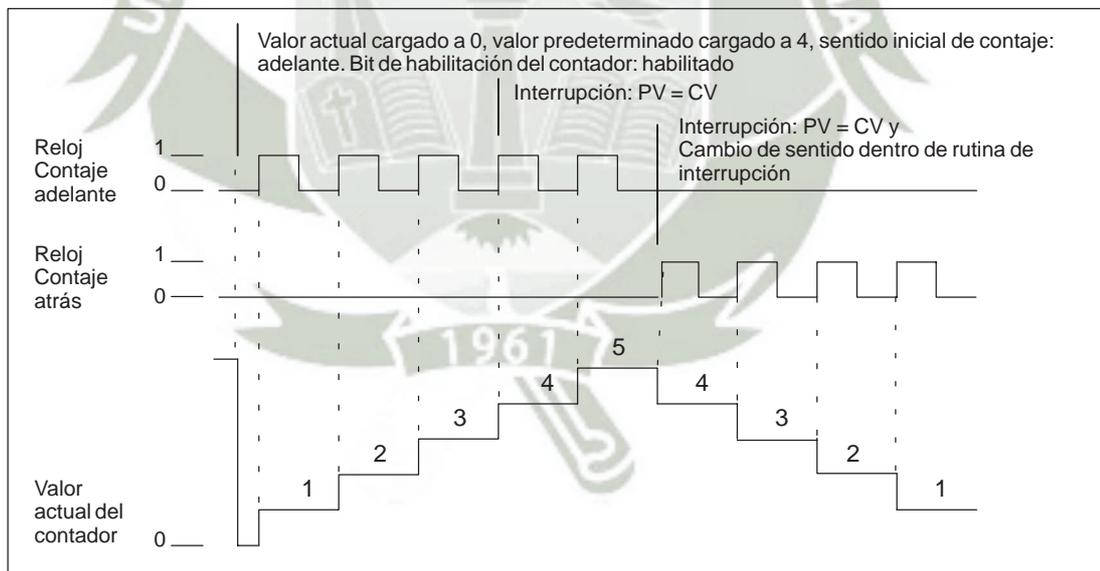


Figura 9-14 Ejemplo del funcionamiento de los modos 6, 7 u 8

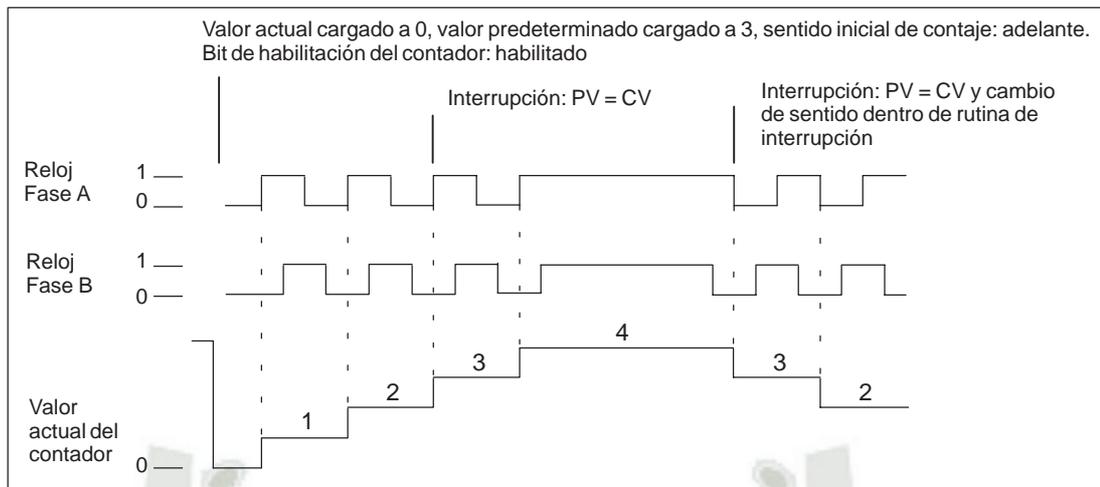


Figura 9-15 Ejemplo del funcionamiento de los modos 9, 10 u 11 (contador A/B, velocidad simple)

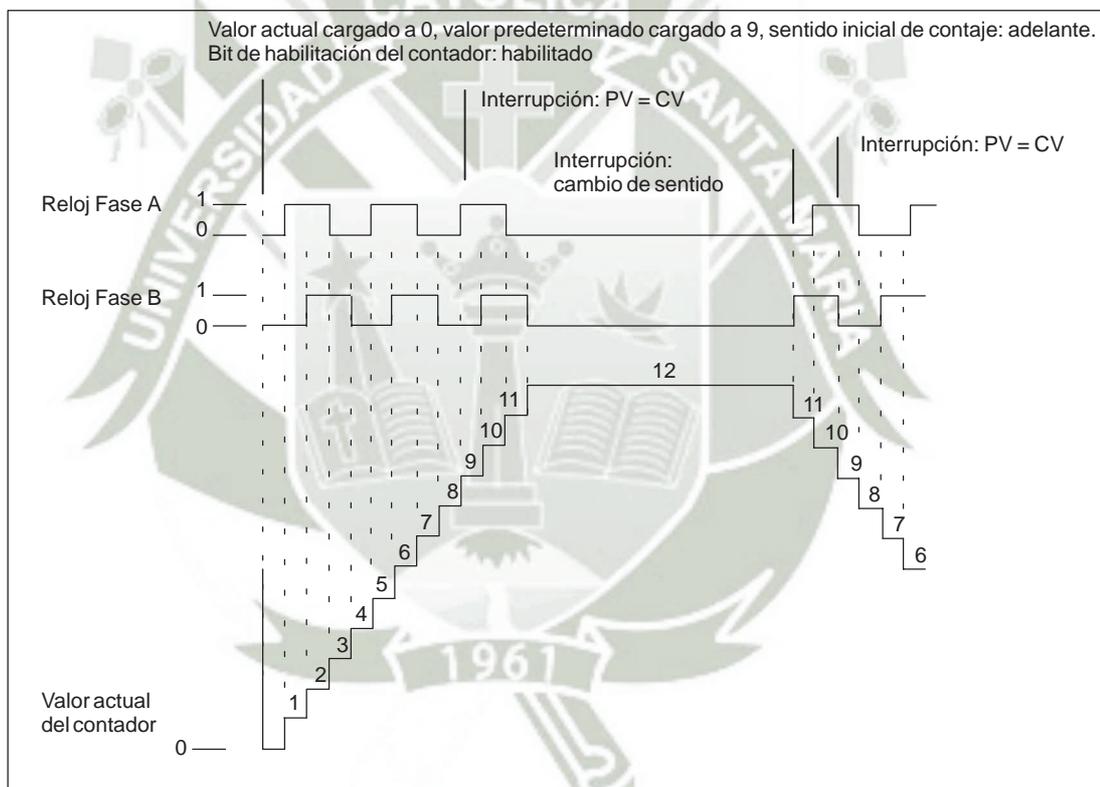


Figura 9-16 Ejemplo del funcionamiento de los modos 9, 10 u 11 (contador A/B, velocidad cuádruple)

### Conectar el cableado de las entradas de los contadores rápidos

La tabla 9-3 muestra las entradas correspondientes al reloj, el control del sentido, la puesta a 0 y las funciones de arranque de los contadores rápidos. Estas funciones de entrada y los modos de operación de los contadores rápidos se describen en las tablas 9-5 a 9-10.

Tabla 9-3 Entradas para los contadores rápidos

Contador rápido	Entradas utilizadas
HSC0	I0.0, I0.1, I0.2
HSC1	I0.6, I0.7, I1.0, I1.1
HSC2	I1.2, I1.3, I1.4, I1.5
HSC3	I0.1
HSC4	I0.3, I0.4, I0.5
HSC5	I0.4

Como muestra el área sombreada de la tabla 9-4, la asignación de entradas de algunos contadores rápidos se solapa con las interrupciones de flanco. Una misma entrada no se puede utilizar para dos funciones diferentes. No obstante, cualquier entrada que no se esté utilizando en el modo actual del contador rápido se puede utilizar para otro fin. Por ejemplo, si HSC0 se está utilizando en modo 2 (que utiliza las entradas I0.0 e I0.2), I0.1 se podrá utilizar para interrupciones de flanco o para HSC3.

Si se utiliza un modo de HSC0 que no use la entrada I0.1, ésta se podrá emplear para HSC3 o para interrupciones de flanco. De forma similar, si I0.2 no se utiliza en el modo de HSC0 seleccionado, dicha entrada estará disponible para interrupciones de flanco. Asimismo, si I0.4 no se usa en el modo de HSC4 seleccionado, dicha entrada se podrá utilizar para HSC5. Es preciso tener en cuenta que todos los modos de HSC0 utilizan siempre I0.0 y que todos los de HSC4 usan siempre I0.3. Por tanto, dichas entradas nunca estarán disponibles para otros fines cuando se estén utilizando dichos contadores.

Tabla 9-4 Asignación de entradas para los contadores rápidos y las interrupciones de flanco

Entrada (I)														
Elemento	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	1.0	1.1	1.2	1.3	1.4	1.5
HSC0	x	x	x											
HSC1							x	x	x	x				
HSC2											x	x	x	x
HSC3		x												
HSC4				x	x	x								
HSC5					x									
Interrupciones de flanco	x	x	x	x										

Tabla 9-5 Modos de operación de HSC0

HSC0					
Modo	Descripción	I0.0	I0.1	I0.2	
0	Contador adelante/atrás de fase simple con control interno del sentido de contaje.	Reloj			
1	SM37.3 = 0, contaje atrás SM37.3 = 1, contaje adelante			Puesta a 0	
3	Contador adelante/atrás de fase simple con control externo del sentido de contaje.	Reloj	Sentido		
4	I0.1 = 0, contaje atrás I0.1 = 1, contaje adelante			Puesta a 0	
6	Contador de dos fases con entradas de reloj para contaje adelante/atrás	Reloj (adelante)	Reloj (atrás)	Puesta a 0	
7					
9	Contador A/B, la fase A está desfasada 90 grados respecto a B en sentido horario,	Reloj (fase A)	Reloj (fase B)		
10	la fase B está desfasada 90 grados respecto a A en sentido antihorario			Puesta a 0	

Tabla 9-6 Modos de operación de HSC1

HSC1					
Modo	Descripción	I0.6	I0.7	I1.0	I1.1
0	Contador adelante/atrás de fase simple con control interno del sentido de contaje. SM47.3 = 0, contaje atrás SM47.3 = 1, contaje adelante	Reloj		Puesta a 0	Arranque
1					
2					
3	Contador adelante/atrás de fase simple con control externo del sentido de contaje. I0.7 = 0, contaje atrás I0.7 = 1, contaje adelante	Reloj	Sentido	Puesta a 0	Arranque
4					
5					
6	Contador de dos fases con entradas de reloj para contaje adelante/atrás	Reloj (adelante)	Reloj (atrás)	Puesta a 0	Arranque
7					
8					
9	Contador A/B, la fase A está desfasada 90 grados respecto a B en sentido horario, la fase B está desfasada 90 grados respecto a A en sentido antihorario	Reloj (fase A)	Reloj (fase B)	Puesta a 0	Arranque
10					
11					

Tabla 9-7 Modos de operación de HSC2

HSC2					
Modo	Descripción	I1.2	I1.3	I1.4	I1.5
0	Contador adelante/atrás de fase simple con control interno del sentido de contaje. SM 57.3 = 0, contaje atrás SM 57.3 = 1, contaje adelante	Reloj		Puesta a 0	Arranque
1					
2					
3	Contador adelante/atrás de fase simple con control externo del sentido de contaje. I1.3 = 0, contaje atrás I1.3 = 1, contaje adelante	Reloj	Sentido	Puesta a 0	Arranque
4					
5					
6	Contador de dos fases con entradas de reloj para contaje adelante/atrás	Reloj (adelante)	Reloj (atrás)	Puesta a 0	Arranque
7					
8					
9	Contador A/B, la fase A está desfasada 90 grados respecto a B en sentido horario, la fase B está desfasada 90 grados respecto a A en sentido antihorario	Reloj (fase A)	Reloj (fase B)	Puesta a 0	Arranque
10					
11					

Tabla 9-8 Modos de operación de HSC3

HSC3					
Modo	Descripción	I0.1			
0	Contador adelante/atrás de fase simple con control interno del sentido de contaje. SM137.3 = 0, contaje atrás SM137.3 = 1, contaje adelante	Reloj			

Tabla 9-9 Modos de operación de HSC4

HSC4					
Modo	Descripción	I0.3	I0.4	I0.5	
0	Contador adelante/atrás de fase simple con control interno del sentido de contaje. SM147.3 = 0, contaje atrás SM147.3 = 1, contaje adelante	Reloj			
1				Puesta a 0	
3	Contador adelante/atrás de fase simple con control externo del sentido de contaje. I0.4 = 0, contaje atrás I0.4 = 1, contaje adelante	Reloj	Sentido		
4				Puesta a 0	
6	Contador de dos fases con entradas de reloj para contaje adelante/atrás	Reloj (adelante)	Reloj (atrás)		
7				Puesta a 0	
9	Contador A/B, la fase A está desfasada 90 grados respecto a B en sentido horario, la fase B está desfasada 90 grados respecto a A en sentido antihorario	Reloj (fase A)	Reloj (fase B)		
10				Puesta a 0	

Tabla 9-10 Modos de operación de HSC5

HSC5					
Modo	Descripción	I0.4			
0	Contador adelante/atrás de fase simple con control interno del sentido de contaje. SM157.3 = 0, contaje atrás SM157.3 = 1, contaje adelante	Reloj			

### Direccionamiento de los contadores rápidos (HC)

Para acceder al valor de contaje del contador rápido, se indica la dirección del mismo (utilizando el identificador HC) y el número del contador (p.ej. HC0). El valor actual del contador rápido es de sólo lectura, pudiéndose acceder al mismo sólo en formato de palabra doble (32 bits), como muestra la figura 9-17.

Formato: **HC[número del contador rápido]** **HC1**

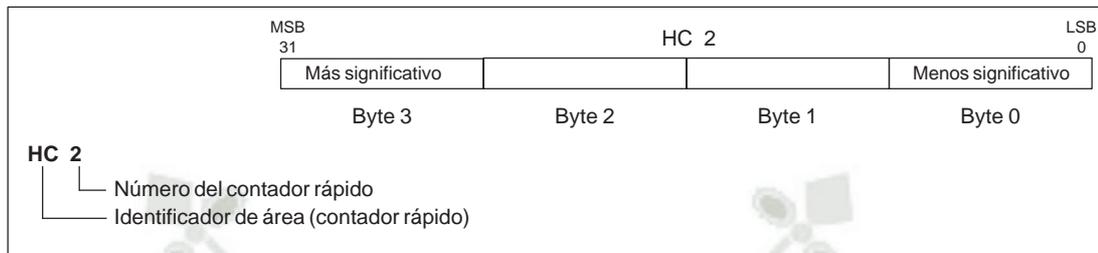


Figura 9-17 Acceso a los valores actuales del contador rápido

### Descripción de los diferentes contadores rápidos

Todos los contadores funcionan de la misma manera en el mismo modo de operación. Como muestra la tabla 9-5, hay cuatro tipos básicos de contadores. Es preciso tener en cuenta que no todos los contadores asisten todos los modos. Todos los contadores se pueden utilizar sin entrada de puesta a 0 ni de arranque, con entrada de puesta a 0 pero sin entrada de arranque, o bien, con entrada de puesta a 0 y de arranque.

Activando la entrada de puesta a 0 se borra el valor actual del contador hasta que vuelve a ser desactivada. Al activarse la entrada de arranque se habilita el contador. Si se desactiva dicha entrada se mantiene el valor actual del contador, ignorándose los eventos de reloj. Si se activa la entrada de puesta a 0 mientras está desactivada la entrada del arranque, se ignorará la activación de la entrada de puesta a 0, con lo que no se modificará el valor actual. Si la entrada de arranque se activa mientras está activada la entrada de puesta a 0, el valor actual se borrará.

Antes de poder utilizar un contador rápido es preciso elegir su modo de operación. Para ello se utiliza la operación HDEF (Definir modo para contador rápido). HDEF establece la conexión entre un contador rápido (HSCx) y el modo de contaje. Por cada contador sólo se puede ejecutar una operación HDEF. Un contador rápido se define utilizando la marca del primer ciclo SM0.1 (este bit se activa sólo en el primer ciclo y se desactiva posteriormente) para llamar a la subrutina que contiene la operación HDEF.

### Elegir el nivel de actividad y el modo de contaje simple o cuádruple

Cuatro contadores tienen tres marcas de control que se utilizan para configurar el estado activo de las entradas de puesta a 0 y de arranque, así como para seleccionar la velocidad simple o cuádruple (ésto sólo en los contadores A/B). Dichas marcas están depositadas en el byte de control del respectivo contador y se emplean solamente cuando se ejecuta la operación HDEF. La tabla 9-11 muestra las marcas.

Antes de poder ejecutar la operación HDEF es preciso ajustar las marcas de control de HSC1 y HSC2 al estado deseado. De lo contrario, el contador adoptará la configuración pre-determinada del modo de contaje elegido. El ajuste estándar de las entradas de puesta a 0 y de arranque es de actividad alta, y la velocidad de contaje es la cuádruple (es decir, la frecuencia del reloj de entrada multiplicada por cuatro). Una vez ejecutada la operación HDEF, ya no se podrá modificar el ajuste de los contadores, a menos que la CPU se cambie a modo STOP.

Tabla 9-11 Nivel de actividad de las entradas de puesta a 0 y de arranque, marcas para elegir la velocidad simple o cuádruple

HSC0	HSC1	HSC2	HSC4	Descripción (sólo cuando se ejecuta HDEF)
SM37.0	SM47.0	SM57.0	SM147.0	Bit de control para nivel de actividad de la entrada de puesta a 0: 0 = actividad alta; 1 = actividad baja
—	SM47.1	SM57.1	—	Bit de control para nivel de actividad de la entrada de arranque: 0 = actividad alta; 1 = actividad baja
SM37.2	SM47.2	SM57.2	SM147.2	Velocidad de contaje de los contadores A/B: 0 = velocidad cuádruple; 1 = velocidad simple

## Byte de control

Una vez definido el contador y el modo de conteje se deben programar los parámetros dinámicos del mismo. Cada contador rápido dispone de un byte que lo habilita o inhibe, fijando el sentido de control (sólo en los modos 0, 1 y 2). El byte de control determina asimismo el sentido de conteje inicial para todos los modos restantes, así como el valor actual y el valor predeterminado que se cargarán. El byte de control, los valores actuales asignados y los valores predeterminados se comprueban al ejecutarse la operación HSC. La tabla 9-12 describe cada una de las marcas del byte de control.

Tabla 9-12 Marcas de control de HSC0, HSC1 y HSC2

HSC0	HSC1	HSC2	HSC3	HSC4	HSC5	Descripción
SM37.0	SM47.0	SM57.0	SM137.0	SM147.0	SM157.0	No se utilizan tras ejecutar HDEF (los contadores que no tengan una entrada externa de puesta a 0 no las usan nunca).
SM37.1	SM47.1	SM57.1	SM137.1	SM147.1	SM157.1	No se utilizan tras ejecutar HDEF (los contadores que no tengan una entrada de arranque no las usan nunca).
SM37.2	SM47.2	SM57.2	SM137.2	SM147.2	SM157.2	No se utilizan tras ejecutar HDEF (los contadores que no asistan la velocidad cuádruple no las utilizan nunca).
SM37.3	SM47.3	SM57.3	SM137.3	SM147.3	SM157.3	Bit de control para el sentido de conteje: 0 = conteje atrás; 1 = conteje adelante
SM37.4	SM47.4	SM57.4	SM137.4	SM147.4	SM157.4	Escribir el sentido de conteje en el contador rápido: 0 = no actualizar; 1 = actualizar el sentido de conteje
SM37.5	SM47.5	SM57.5	SM137.5	SM147.5	SM157.5	Escribir el nuevo valor predeterminado en el contador rápido: 0 = no actualizar; 1 = actualizar el valor predeterminado
SM37.6	SM47.6	SM57.6	SM137.6	SM147.6	SM157.6	Escribir el nuevo valor actual en el contador rápido: 0 = no actualizar; 1 = actualizar el valor actual
SM37.7	SM47.7	SM57.7	SM137.7	SM147.7	SM157.7	Habilitar el contador rápido: 0 = inhibir el contador rápido; 1 = habilitar el contador rápido

## Ajustar los valores actuales y predeterminados

Todo contador rápido dispone de un valor actual y de un valor predeterminado de 32 bits. Ambos son valores enteros con signo. Para cargar un nuevo valor actual o predeterminado en el contador rápido es preciso activar el byte de control y los bytes de las marcas especiales que contienen los valores actuales y/o predeterminados. Después se ejecuta la operación HSC para transferir los nuevos valores al contador rápido. La tabla 9-13 describe los bytes de marcas especiales que contienen los nuevos valores y los valores predeterminados.

Además de los bytes de control y de los bytes que contienen los nuevos valores predeterminados, también se puede leer el valor actual de cada uno de los contadores rápidos, utilizando el tipo de datos HC (valor actual del contador rápido) seguido del número de contador (0, 1, 2, 3, 4 ó 5). Ello permite acceder directamente al valor actual para operaciones de lectura. Por el contrario, dicho valor sólo se puede escribir utilizando la operación HSC que se describe más arriba.

Tabla 9-13 Valores actuales y predeterminado de los contadores HSC0, HSC1, HSC2, HSC3, HSC4 y HSC5

Valor a cargar	HSC0	HSC1	HSC2	HSC3	HSC4	HSC5
Nuevo actual	SMD38	SMD48	SMD58	SMD138	SMD148	SMD158
Nuevo predeterminado	SMD42	SMD52	SMD62	SMD142	SMD152	SMD162

### Byte de estado

Todos los contadores rápidos disponen de un byte para marcas de estado. Éstas indican el sentido de contaje actual y si el valor actual es igual o mayor que el valor predeterminado. La tabla 9-14 muestra las marcas de estado de los contadores rápidos.

Tabla 9-14 Marcas de estado de los contadores HSC0, HSC1, HSC2, HSC3, HSC4 y HSC5

HSC0	HSC1	HSC2	HSC3	HSC4	HSC5	Descripción
SM36.0	SM46.0	SM56.0	SM136.0	SM146.0	SM156.0	No utilizado
SM36.1	SM46.1	SM56.1	SM136.1	SM146.1	SM156.1	No utilizado
SM36.2	SM46.2	SM56.2	SM136.2	SM146.2	SM156.2	No utilizado
SM36.3	SM46.3	SM56.3	SM136.3	SM146.3	SM156.3	No utilizado
SM36.4	SM46.4	SM56.4	SM136.4	SM146.4	SM156.4	No utilizado
SM36.5	SM46.5	SM56.5	SM136.5	SM146.5	SM156.5	Bit de estado para sentido de contaje actual: 0 = contaje atrás; 1 = contaje adelante
SM36.6	SM46.6	SM56.6	SM136.6	SM146.6	SM156.6	Bit de estado para valor actual igual a valor predeterminado: 0 = diferente, 1 = igual
SM36.7	SM46.7	SM56.7	SM136.7	SM146.7	SM156.7	Bit de estado para valor actual mayor que valor predeterminado: 0 = menor o igual, 1 = mayor que

### Nota

Las marcas de estado son válidas únicamente mientras se está ejecutando la rutina de interrupción para el contador rápido. El estado del contador rápido se supervisa con objeto de habilitar las interrupciones para los eventos que puedan afectar a la operación que se está ejecutando.

## Interrupciones de los contadores rápidos

Todos los modos de los contadores asisten una interrupción si el valor actual es igual al valor predeterminado. Los modos de los contadores que utilizan una entrada de puesta a 0 externa asisten una interrupción que se ejecuta cuando se activa dicha entrada. Todos los modos de conteo (con excepción de los modos 0, 1 y 2) asisten una interrupción que se ejecuta cuando se produce un cambio del sentido de conteo. Cada una de estas condiciones puede habilitarse o inhibirse por separado. Para obtener más información acerca de la utilización de interrupciones, consulte el apartado 9.16.

---

### Nota

Cuando esté utilizando la entrada de puesta a 0 externa, no intente cargar un nuevo valor actual ni tampoco inhibir y habilitar luego el contador rápido desde la rutina de interrupción asociada a ese evento. En tal caso podría producirse un error fatal.

---

A continuación se describen las secuencias de inicialización y de operación de los contadores rápidos para facilitar la comprensión de su funcionamiento. En las siguientes descripciones se ha utilizado el contador HSC1 a título de ejemplo. En las explicaciones relativas a la inicialización se supone que el sistema de automatización S7-200 está en modo RUN y que, por consiguiente, la marca del primer ciclo es verdadera. En otro caso se deberá tener en cuenta que la operación HDEF sólo puede ejecutarse una vez por cada contador rápido, después de haber cambiado a modo RUN. Si la operación HDEF se ejecuta por segunda vez para un contador rápido se producirá un error de tiempo de ejecución. Los ajustes del contador permanecerán entonces tal y como se configuraron con la primera operación HDEF que se ejecutó para el contador en cuestión.

## Modos de inicialización 0, 1 ó 2

Para inicializar HSC1 como contador adelante/atrás de fase simple con control interno del sentido de contaje (modos 0, 1 ó 2):

1. Con la marca del primer ciclo, llame a una subrutina para ejecutar la inicialización. Puesto que se utiliza una llamada a subrutina, los siguientes ciclos ya no llaman a la misma, con lo cual se acorta el tiempo de ciclo y el programa queda mejor estructurado.
2. Cargue la marca SMB47 en la subrutina de inicialización conforme a la operación de control deseada. Ejemplo:  
SMB47 = 16#F8 Resultados:  
Se habilita el contador.  
Se escribe un nuevo valor actual.  
Se escribe un nuevo valor predeterminado.  
Se ajusta el sentido de contaje adelante.  
Se ajusta la actividad alta de las entradas de arranque y de puesta a 0.
3. Ejecute la operación HDEF con la entrada HSC puesta a 1 y la entrada MODE a 0 para puesta a 0 o arranque no externos, o bien a 1 para puesta a 0 externa y sin arranque, o bien a 2 para puesta a 0 y arranque externos.
4. Cargue el valor actual deseado en SMD48 (valor de palabra doble) (cargue 0 para borrar la marca).
5. Cargue el valor predeterminado deseado en SMD52 (valor de palabra doble).
6. Para averiguar si el valor actual es igual al predeterminado, programe una interrupción asociando el evento de interrupción CV = PV (evento 13) a una rutina de interrupción. Para obtener más información acerca de la utilización de interrupciones, consulte el apartado 9.16.
7. Para poder detectar una puesta a 0 externa, programe una interrupción asociando el evento de interrupción Puesta a 0 externa (evento 15) a una rutina de interrupción.
8. Ejecute la operación Habilitar todos los eventos de interrupción (ENI) para habilitar las interrupciones.
9. Ejecute la operación HSC para que el sistema de automatización S7-200 programe el contador HSC1.
10. Finalice la subrutina.

### Modos de inicialización 3, 4 ó 5

Para inicializar HSC1 como contador adelante/atrás de fase simple con control externo del sentido de contaje (modos 3, 4 ó 5):

1. Con la marca del primer ciclo, llame a una subrutina para ejecutar la inicialización. Puesto que se utiliza una llamada a subrutina, los siguientes ciclos ya no llaman a la misma, con lo cual se acorta el tiempo de ciclo y el programa queda mejor estructurado.
2. Cargue la marca SMB47 en la subrutina de inicialización conforme a la operación de control deseada. Ejemplo:  
SMB47 = 16#F8 Resultados:  
Se habilita el contador.  
Se escribe un nuevo valor actual.  
Se escribe un nuevo valor predeterminado.  
Se ajusta el sentido inicial de contaje adelante.  
Se ajusta la actividad alta de las entradas de arranque y de puesta a 0.
3. Ejecute la operación HDEF con la entrada HSC puesta a 1 y la entrada MODE a 3 para puesta a 0 o arranque no externos, o bien a 4 para puesta a 0 externa y sin arranque, o bien a 5 para puesta a 0 y arranque externos.
4. Cargue el valor actual deseado en SMD48 (valor de palabra doble) (cargue 0 para borrar la marca).
5. Cargue el valor predeterminado deseado en SMD52 (valor de palabra doble).
6. Para averiguar si el valor actual es igual al predeterminado, programe una interrupción asociando el evento de interrupción CV = PV (evento 13) a una rutina de interrupción. Para obtener más información acerca de la utilización de interrupciones, consulte el apartado 9.16.
7. Para poder detectar un cambio del sentido de contaje, programe una interrupción asociando el evento de interrupción de cambio de sentido (evento 14) a una rutina de interrupción.
8. Para poder detectar una puesta a 0 externa, programe una interrupción asociando el evento de interrupción Puesta a 0 externa (evento 15) a una rutina de interrupción.
9. Ejecute la operación Habilitar todos los eventos de interrupción (ENI) para habilitar las interrupciones.
10. Ejecute la operación HSC para que el sistema de automatización S7-200 programe el contador HSC1.
11. Finalice la subrutina.

### Modos de inicialización 6, 7 u 8

Para inicializar HSC1 como contador adelante/atrás de dos fases con relojes adelante/atrás (modos 6, 7 u 8):

1. Con la marca del primer ciclo, llame a una subrutina para ejecutar la inicialización. Puesto que se utiliza una llamada a subrutina, los siguientes ciclos ya no llaman a la misma, con lo cual se acorta el tiempo de ciclo y el programa queda mejor estructurado.
2. Cargue la marca SMB47 en la subrutina de inicialización conforme a la operación de control deseada. Ejemplo:  
SMB47 = 16#F8 Resultados:  
Se habilita el contador.  
Se escribe un nuevo valor actual.  
Se escribe un nuevo valor predeterminado.  
Se ajusta el sentido inicial de contaje adelante.  
Se ajusta la actividad alta de las entradas de arranque y de puesta a 0.
3. Ejecute la operación HDEF con la entrada HSC puesta a 1 y la entrada MODE a 6 para puesta a 0 o arranque no externos, o bien a 7 para puesta a 0 externa y sin arranque, o bien a 8 para puesta a 0 y arranque externos.
4. Cargue el valor actual deseado en SMD48 (valor de palabra doble) (cargue 0 para borrar la marca).
5. Cargue el valor predeterminado deseado en SMD52 (valor de palabra doble).
6. Para averiguar si el valor actual es igual al predeterminado, programe una interrupción asociando el evento de interrupción CV = PV (evento 13) a una rutina de interrupción. Para obtener más información acerca de la utilización de interrupciones, consulte el apartado 9.16.
7. Para poder detectar un cambio del sentido de contaje, programe una interrupción asociando el evento de interrupción de cambio de sentido (evento 14) a una rutina de interrupción.
8. Para poder detectar una puesta a 0 externa, programe una interrupción asociando el evento de interrupción Puesta a 0 externa (evento 15) a una rutina de interrupción.
9. Ejecute la operación Habilitar todos los eventos de interrupción (ENI) para habilitar las interrupciones.
10. Ejecute la operación HSC para que el sistema de automatización S7-200 programe el contador HSC1.
11. Finalice la subrutina.

## Modos de inicialización 9, 10 u 11

Para inicializar HSC1 como contador A/B (modos 9, 10 u 11):

1. Con la marca del primer ciclo, llame a una subrutina para ejecutar la inicialización. Puesto que se utiliza una llamada a subrutina, los siguientes ciclos ya no llaman a la misma, con lo cual se acorta el tiempo de ciclo y el programa queda mejor estructurado.
2. Cargue la marca SMB47 en la subrutina de inicialización conforme a la operación de control deseada.

Ejemplo (frecuencia simple):

SMB47 = 16#FC Resultados:

- Se habilita el contador.
- Se escribe un nuevo valor actual.
- Se escribe un nuevo valor predeterminado.
- Se ajusta el sentido inicial de contaje adelante.
- Se ajusta la actividad alta de las entradas de arranque y de puesta a 0.

Ejemplo (frecuencia cuádruple):

SMB47 = 16#F8 Resultados:

- Se habilita el contador.
- Se escribe un nuevo valor actual.
- Se escribe un nuevo valor predeterminado.
- Se ajusta el sentido inicial de contaje adelante.
- Se ajusta la actividad alta de las entradas de arranque y de puesta a 0.

3. Ejecute la operación HDEF con la entrada HSC puesta a 1 y la entrada MODE a 9 para puesta a 0 o arranque no externos, o bien a 10 para puesta a 0 externa y sin arranque, o bien a 11 para puesta a 0 y arranque externos.
4. Cargue el valor actual deseado en SMD48 (valor de palabra doble) (cargue 0 para borrar la marca).
5. Cargue el valor predeterminado deseado en SMD52 (valor de palabra doble).
6. Para averiguar si el valor actual es igual al predeterminado, programe una interrupción asociando el evento de interrupción CV = PV (evento 13) a una rutina de interrupción. Para obtener más información acerca de la utilización de interrupciones, consulte el apartado 9.16.
7. Para poder detectar un cambio del sentido de contaje, programe una interrupción asociando el evento de interrupción de cambio de sentido (evento 14) a una rutina de interrupción.
8. Para poder detectar una puesta a 0 externa, programe una interrupción asociando el evento de interrupción Puesta a 0 externa (evento 15) a una rutina de interrupción.
9. Ejecute la operación Habilitar todos los eventos de interrupción (ENI) para habilitar las interrupciones.
10. Ejecute la operación HSC para que el sistema de automatización S7-200 programe el contador HSC1.
11. Finalice la subrutina.

### Cambio de sentido en los modos 0, 1 ó 2

Para configurar el cambio de sentido de HSC1 como contador de fase simple con control interno del sentido de contaje (modos 0,1 ó 2):

1. Cargue SMB47 para escribir la dirección deseada:
  - SMB47 = 16#90 Habilita el contador.  
Ajusta el sentido de contaje atrás.
  - SMB47 = 16#98 Habilita el contador.  
Ajusta el sentido de contaje adelante.
2. Ejecute la operación HSC para que el sistema de automatización S7-200 programe el contador HSC1.

### Cargar nuevo valor actual (cualquier modo)

Para modificar el valor actual del contador HSC1 (cualquier modo):

Si se modifica el valor actual, el contador se inhibirá automáticamente. Mientras está inhibido el contador, no cuenta ni tampoco se generan interrupciones.

1. Cargue SMB47 para escribir el valor actual deseado:
  - SMB47 = 16#C0 Habilita el contador.  
Escribe el nuevo valor actual.
2. Cargue el valor actual deseado en SMD48 (valor de palabra doble) (cargue 0 para borrar la marca).
3. Ejecute la operación HSC para que el sistema de automatización S7-200 programe el contador HSC1.

### **Cargar nuevo valor predeterminado (cualquier modo)**

Para modificar el valor predeterminado de HSC1 (cualquier modo):

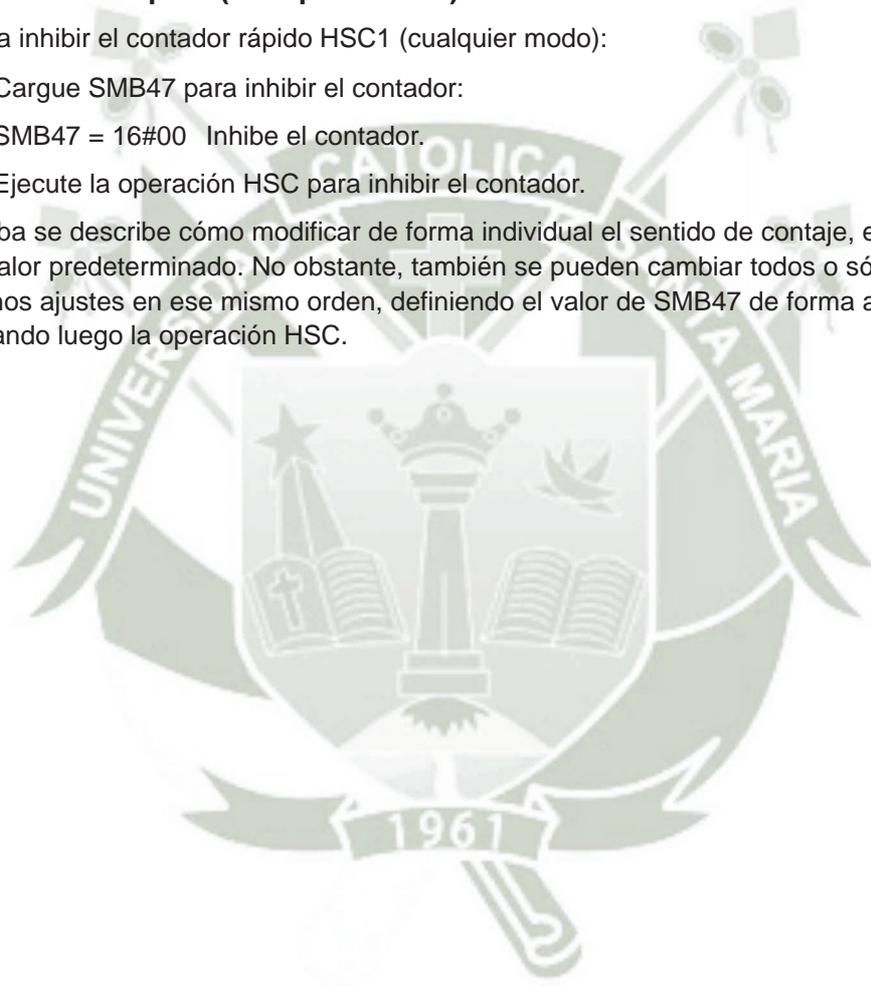
1. Cargue SMB47 para escribir el valor predeterminado deseado:  
SMB47 = 16#A0 Habilita el contador.  
Escribe el nuevo valor predeterminado.
2. Cargue el valor predeterminado deseado en SMD52 (valor de palabra doble).
3. Ejecute la operación HSC para que el sistema de automatización S7-200 programe el contador HSC1.

### **Inhibir un contador rápido (cualquier modo)**

Para inhibir el contador rápido HSC1 (cualquier modo):

1. Cargue SMB47 para inhibir el contador:  
SMB47 = 16#00 Inhibe el contador.
2. Ejecute la operación HSC para inhibir el contador.

Arriba se describe cómo modificar de forma individual el sentido de contaje, el valor actual o el valor predeterminado. No obstante, también se pueden cambiar todos o sólo algunos de dichos ajustes en ese mismo orden, definiendo el valor de SMB47 de forma apropiada y ejecutando luego la operación HSC.



Ejemplo contador rápido

KOP		AWL
PROGRAMA PRINCIPAL OB1		
<p>Network 1</p>	<p>Llamar subrutina 0 en el primer ciclo.</p> <p>Fin programa principal.</p>	<p>Network 1</p> <p>LD SM0.1</p> <p>CALL 0</p>
SUBROUTINA 0		
<p>Network 1</p>	<p>Habilitar el contador. Escribir un nuevo valor actual. Escribir un nuevo valor predeterminado. Ajustar el sentido inicial de conteo adelante. Ajustar la actividad alta de las entradas de arranque y de puesta a 0. Ajustar la velocidad cuádruple.</p> <p>HSC1 configurado para frecuencia cuádruple con entradas de puesta a 0 y de arranque.</p> <p>Poner a 0 el valor actual de HSC1.</p> <p>Ajustar a 50 el valor predeterminado de HSC1.</p> <p>HSC 1 valor actual = valor predeterminado (EVENTO 13) asociado a rutina de interrupción 0.</p> <p>Habilitar todos los eventos de interrupción.</p> <p>Programar HSC1.</p>	<p>Network 1</p> <p>LD SM0.0</p> <p>MOVB 16#F8, SMB47</p> <p>HDEF 1, 11</p> <p>MOVD 0, SMD48</p> <p>MOVD 50, SMD52</p> <p>ATCH 0, 13</p> <p>ENI</p> <p>HSC 1</p>
RUTINA DE INTERRUPCIÓN 0		
<p>Network 1</p>	<p>Poner a 0 el valor actual de HSC1.</p> <p>Escribir un nuevo valor actual y habilitar el contador.</p> <p>Programar HSC1.</p>	<p>Network 1</p> <p>LD SM 0.0</p> <p>MOVD 0, SMD48</p> <p>MOVB 16#C0, SMB47</p> <p>HSC 1</p>

Figura 9-18 Ejemplo de inicialización de HSC1 (KOP y AWL)

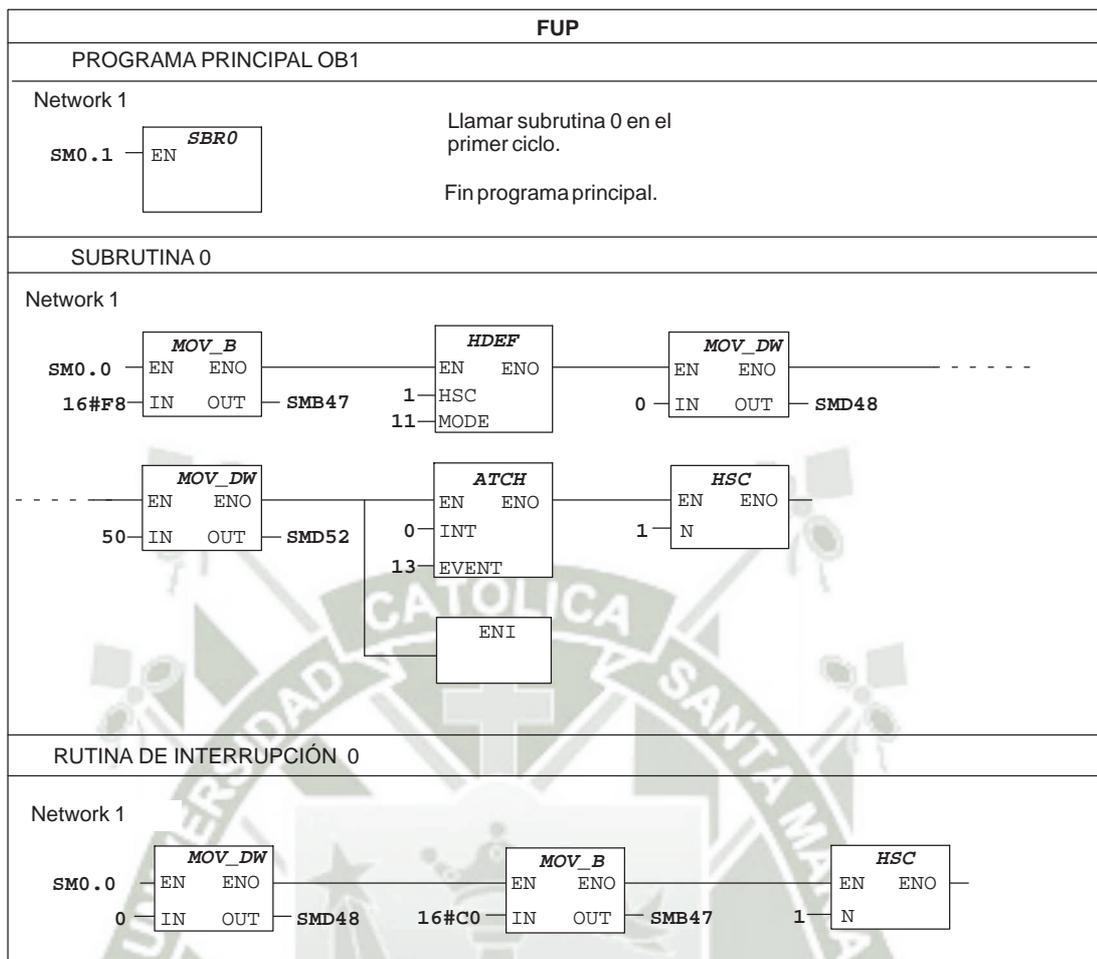
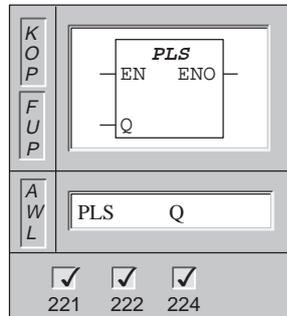


Figura 9-19 Ejemplo de inicialización de HSC1 (FUP)

## 9.6 Operaciones de salida de impulsos (SIMATIC)

### Salida de impulsos



La operación **Salida de impulsos** examina las marcas especiales de la salida de impulsos (Q0.0 ó Q0.1). A continuación se invoca la operación de salida de impulsos definida por las marcas especiales.

Operandos: Q Constante (0 ó 1)

Tipos de datos: WORD

Márgenes de salida de impulsos Q0.0 hasta Q0.1

### Descripción de las operaciones rápidas de salida del S7-200

Todas las CPUs disponen de dos generadores PTO/PWM para producir trenes de impulsos rápidos y formas de onda moduladas por ancho de impulsos. Uno de dichos generadores está asignado a la salida digital Q0.0 y, el otro, a la salida digital Q0.1.

Los generadores PTO/PWM y la imagen del proceso comparten el uso de las salidas Q0.0 y Q0.1. Cuando está activa una función PTO o PWM en Q0.0 ó Q0.1, el generador PTO/PWM controla la salida, inhibiéndose el uso normal de la misma. La forma de onda de la salida no se ve afectada por el estado de la imagen del proceso, ni por el valor forzado de la salida o la ejecución de las operaciones directas de salida. Si el generador PTO/PWM está desactivado, el control de la salida retorna a la imagen del proceso. La imagen del proceso determina los estados inicial y final de la forma de onda de la salida, para que la forma de onda comience y termine en un nivel alto o bajo.

#### Nota

Se recomienda ajustar a 0 la imagen del proceso de Q0.0 y Q0.1 antes de habilitar las operaciones PTO o PWM.

La función Tren de impulsos (PTO) ofrece una onda cuadrada (con un factor de trabajo relativo de 50%), pudiendo el usuario controlar el tiempo de ciclo y el número de impulsos. La función Modulación del ancho de impulsos (PWM) ofrece una salida continua con un factor de trabajo relativo variable, pudiendo el usuario controlar el tiempo de ciclo y el ancho de impulsos.

Cada generador PTO/PWM tiene un byte de control (8 bits), un valor de tiempo de ciclo y un valor de ancho de impulsos (ambos son valores de 16 bits sin signo), así como un valor de conteo de impulsos (valor de 32 bits sin signo). Estos valores están almacenados en determinadas direcciones del área de marcas especiales (SM). Una vez que se disponga de las direcciones de marcas especiales para seleccionar la función deseada, ésta se invoca ejecutando la operación Salida de impulsos (PLS). Con esta operación, la CPU S7-200 lee las direcciones de las marcas especiales (SM) y programa el generador PTO/PWM.

Para cambiar las características de una forma de onda PTO o PWM es preciso modificar las direcciones deseadas en el área SM (incluyendo el byte de control), ejecutando luego la operación PLS.

La generación de una forma de onda PTO o PWM se puede inhibir en cualquier momento, poniendo a cero el bit de habilitación PTO/PWM del byte de control (SM67.7 ó SM77.7) y ejecutando luego la operación PLS.

---

**Nota**

El ajuste estándar de los bits de control, del tiempo de ciclo, del ancho de impulsos y de los valores de contaje es 0.

---

---

**Nota**

Las salidas PTO/PWM requieren una carga mínima de 10% de la carga nominal para poder garantizar transiciones idóneas de "off" a "on", y viceversa.

---

### **Modulación del ancho de impulsos (PWM)**

La función PWM ofrece una salida con un factor de trabajo relativo variable. El tiempo de ciclo y el ancho de impulsos pueden indicarse con una base de tiempo en microsegundos o milisegundos. El tiempo de ciclo se puede especificar en microsegundos (de 50 a 65.535 microsegundos) o en milisegundos (de 2 a 65.535 milisegundos). El tiempo de ancho de impulsos puede estar comprendido entre 0 y 65.535 microsegundos ó 0 y 65.535 milisegundos. Si se indica un valor del ancho de impulsos mayor o igual al valor del tiempo de ciclo, el factor de trabajo relativo de la forma de onda será de 100% y la salida estará activada continuamente. Si el ancho de impulsos es 0, el factor de trabajo relativo de la forma de onda será de 0% y se desactivará la salida. Si se indica un tiempo de ciclo inferior a dos unidades de tiempo, el tiempo de ciclo se predeterminará en dos unidades de tiempo.

Hay dos maneras diferentes de cambiar las características de una forma de onda PWM: con una actualización síncrona o asíncrona.

- Actualización síncrona: Si no es necesario cambiar la base de tiempo se puede ejecutar una actualización síncrona. En este caso, el cambio de la forma de onda se efectúa en el límite de un ciclo, ofreciendo una transición suave.
- Actualización asíncrona: En una función PWM típica, el ancho de impulsos se modifica, permaneciendo constante el tiempo de ciclo. Por tanto, no es necesario cambiar la base de tiempo. No obstante, si es preciso modificar la base de tiempo del generador PTO/PWM se utiliza una actualización asíncrona. Ésta inhibe el generador PTO/PWM momentáneamente, de forma asíncrona a la forma de onda PWM. Ello puede causar fluctuaciones no deseadas en el dispositivo controlado. Por tanto, se recomienda la actualización síncrona de las formas de ondas PWM. Elija una base de tiempo que se adecúe para todos los valores previstos para el tiempo de ciclo.

La marca del método de actualización PWM (SM67.4 ó SM77.4) en el byte de control se utiliza para especificar el tipo de actualización. La operación PLS se debe ejecutar para invocar los cambios. Es preciso tener en cuenta que si se modifica la base de tiempo se efectuará en todo caso una actualización asíncrona, sin importar el estado de dicha marca especial.

### Tren de impulsos (PTO)

La función PTO genera un tren de impulsos de onda cuadrada (con un factor de trabajo relativo de 50% ) con un número determinado de impulsos. El tiempo de ciclo puede indicarse en incrementos de microsegundos o milisegundos. El tiempo de ciclo se puede especificar en microsegundos (de 50 a 65.535 microsegundos) o en milisegundos (de 2 a 65.535 milisegundos). Si el tiempo de ciclo es un número impar, se distorsionará levemente el factor de trabajo relativo. El número de impulsos puede estar comprendido entre 1 y 4.294.967.295.

Si se indica un tiempo de ciclo inferior a dos unidades de tiempo, el tiempo de ciclo se predeterminará en dos unidades de tiempo. Si se indica un conteo de impulsos de 0, se adoptará un valor predeterminado de un impulso.

El bit de inactividad PTO (SM66.7 or SM76.7) en el byte de estado indica si se ha completado el tren de impulsos programado. Además, tras finalizarse un tren de impulsos se puede llamar a una rutina de interrupción (consulte el apartado 9.16 para obtener más información acerca de las operaciones de interrupción y comunicación). Si se utiliza el pipelining multisegmento, la rutina de interrupción se invocará tras completarse la tabla de perfiles (v. "Pipelining multisegmento").

La operación PTO permite efectuar un "pipelining", es decir, concatenar o canalizar los trenes de impulsos. Tras haberse finalizado el tren de impulsos activo, comienza inmediatamente la salida de un nuevo tren de impulsos. Así se garantiza la continuidad de los trenes de impulsos subsiguientes.

El pipelining se puede efectuar de dos maneras: en un solo segmento o en varios segmentos.

**Pipelining monosegmento** En este modo, el usuario debe actualizar las direcciones de las marcas especiales para el siguiente tren de impulsos. Tras haber arrancado el segmento PTO inicial, el usuario debe modificar inmediatamente las direcciones de las marcas especiales conforme a lo requerido para la segunda forma de onda, y ejecutar nuevamente la operación PLS. Los atributos del segundo tren de impulsos se conservarán en una pipeline hasta finalizar el primer tren de impulsos. En la pipeline sólo se puede almacenar un registro. Tras haber finalizado el primer tren de impulsos, comenzará la salida de la segunda forma de onda y la pipeline quedará libre para acoger un nuevo tren de impulsos. Este proceso se puede repetir entonces para ajustar las características del siguiente tren de impulsos.

Las transiciones entre los trenes de impulsos serán suaves, con excepción de las siguientes situaciones:

- Si se cambia la base de tiempo.
- Si el tren de impulsos activo se termina antes de que la ejecución de la operación PLS capture el ajuste de un nuevo tren de impulsos.

Si se intenta cargar la pipeline mientras está llena, se activará la marca de desbordamiento PTO en el registro de estado (SM66.6 ó SM76.6). Al pasar a RUN, dicha marca se vuelve a poner a 0. Para poder detectar los desbordamientos siguientes, la marca se deberá poner a 0 manualmente tras haberse detectado un desbordamiento.

**Pipelining multisegmento** En este modo, la CPU lee automáticamente las características de cada tren de impulsos en una tabla de perfiles almacenada en la memoria de variables (memoria V). Las únicas marcas especiales utilizadas en este modo son el byte de control y el byte de estado. Para seleccionar el pipelining multisegmento es preciso cargar el offset inicial de la memoria V de la tabla de perfiles (SMW168 ó SMW178). La base de tiempo se puede indicar en microsegundos o milisegundos, pero la selección será aplicable a todos los valores de tiempo de ciclo en la tabla de perfiles, no pudiendo modificarse durante la ejecución del perfil. El pipelining multisegmento se puede arrancar ejecutando la operación PLS.

Cada registro de segmento tiene una longitud de 8 bytes, comprendiendo un valor de tiempo de ciclo de 16 bits, un valor delta de tiempo de ciclo de 16 bits y un valor de contaje de impulsos de 32 bits.

La tabla 9-15 muestra el formato de la tabla de perfiles. Una función adicional disponible en el pipelining multisegmento PTO es la posibilidad de prolongar o acortar automáticamente el tiempo de ciclo en cada impulso. El tiempo de ciclo se prolonga o se acorta programando en su campo delta un valor positivo o negativo, respectivamente. Este tiempo permanecerá inalterado si se programa el valor 0.

Si se indica un valor delta que produzca un tiempo de ciclo no válido al cabo de un número de impulsos, ocurrirá una condición de desbordamiento aritmética. La función PTO terminará y la salida retornará al control de la imagen del proceso. Además, se activará la marca especial de error de cálculo delta en el byte de estado (SM66.4 ó SM76.4).

Si el usuario interrumpe un perfil PTO que se esté ejecutando, se activará la marca especial de interrupción manual en el byte de estado (SM66.5 ó SM76.5).

Mientras se está ejecutando el perfil PTO, el número del segmento activo actualmente se indica en SMB166 (ó SMB176).

Tabla 9-15 Formato de la tabla de perfiles para la función PTO multisegmento

Offset (en bytes) desde el comienzo de la tabla	Número de segmento	Descripción de los registros de la tabla
0		Número de segmentos (1 a 255); el valor 0 genera un error no fatal. No se genera una salida PTO.
1	#1	Tiempo de ciclo inicial (2 a 65535 unidades de la base de tiempo)
3		Valor delta del tiempo de ciclo por impulso (valor con signo) (-32768 a 32767 unidades de la base de tiempo)
5		Contaje de impulsos (1 a 4294967295)
9	#2	Tiempo de ciclo inicial (2 a 65535 unidades de la base de tiempo)
11		Valor delta del tiempo de ciclo por impulso (valor con signo) (-32768 a 32767 unidades de la base de tiempo)
13		Contaje de impulsos (1 a 4294967295)
:	:	:
:	:	:

### Calcular los valores de la tabla de perfiles

El pipelining multisegmento que ofrecen los generadores PTO/PWM se puede utilizar para numerosas aplicaciones, en particular para el control de motores paso a paso.

El ejemplo de la figura 9-20 muestra cómo determinar los valores de la tabla de perfiles necesarios para generar una forma de onda de salida con objeto de acelerar un motor paso a paso, permitir que funcione a una velocidad constante y desacelerarlo luego.

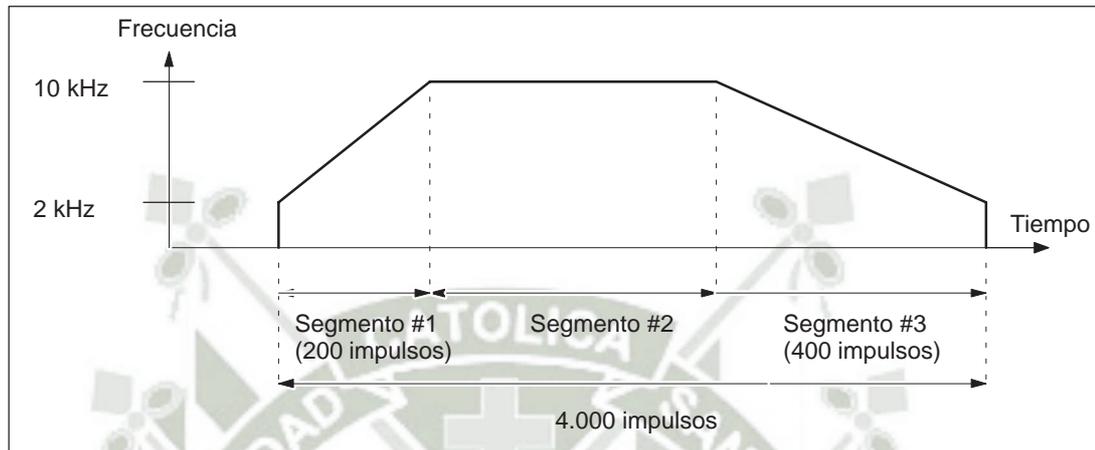


Figura 9-20 Diagrama frecuencia/tiempo de una aplicación de ejemplo para un motor paso a paso

En el presente ejemplo se supone que se necesitan 4.000 impulsos para alcanzar el número deseado de revoluciones del motor. La frecuencia de impulsos inicial y final es de 2 kHz, en tanto que la frecuencia máxima de impulsos es de 10 kHz. Puesto que los valores de la tabla de perfiles se expresan en términos de período (tiempo de ciclo) y no de frecuencia, es preciso convertir los valores de frecuencia dados en valores de tiempo de ciclo. Por tanto, el tiempo de ciclo inicial y final es de 500  $\mu$ s, en tanto que el tiempo de ciclo correspondiente a la frecuencia máxima es de 100  $\mu$ s.

Durante la etapa de aceleración del perfil de salida, la frecuencia máxima de impulsos se desea alcanzar en aproximadamente 200 impulsos. Asimismo, se supone que la etapa de desaceleración del perfil se debe finalizar en unos 400 impulsos.

En el presente ejemplo se puede utilizar una fórmula sencilla para calcular el valor delta del tiempo de ciclo utilizado por el generador PTO/PWM para ajustar el tiempo de ciclo de cada impulso:

$$\text{tiempo de ciclo delta} = |\text{tiempo de ciclo final} - \text{tiempo de ciclo inicial}| / \text{cantidad de impulsos}$$

Utilizando esta fórmula, se calcula que el tiempo de ciclo delta de la etapa de aceleración (o segmento #1) es  $-2$ . De forma similar, el tiempo de ciclo delta de la etapa de desaceleración (o segmento #3) es  $1$ . Puesto que el segmento #2 representa la etapa de velocidad constante de la forma de onda de salida, el valor delta del tiempo de ciclo correspondiente a dicho segmento es  $0$ .

Suponiendo que la tabla de perfiles se encuentra almacenada en la memoria V (comenzando en V500), los valores utilizados para generar la forma de onda deseada figuran en la tabla 9-16.

Tabla 9-16 Valores de la tabla de perfiles

Dirección en la memoria V	Valor
VB500	3 (número total de segmentos)
VW501	500 (tiempo de ciclo inicial – segmento #1)
VW503	-2 (tiempo de ciclo inicial – segmento #1)
VW505	200 (número de impulsos – segmento #1)
VW509	100 (tiempo de ciclo inicial – segmento #2)
VW511	0 (tiempo de ciclo delta – segmento #2)
VW513	3400 (número de impulsos – segmento #2)
VW517	100 (tiempo de ciclo inicial – segmento #3)
VW519	1 (tiempo de ciclo delta – segmento #3)
VD521	400 (número de impulsos – segmento #3)

Los valores de esta tabla se pueden depositar en la memoria V utilizando operaciones en el programa. Una alternativa consiste en definir los valores del perfil en el bloque de datos. La figura 9-23 muestra un ejemplo que contiene las operaciones para utilizar la función multi-segmento PTO.

El tiempo de ciclo del último impulso de un segmento no se indica directamente en el perfil, sino que se debe calcular (a menos que el tiempo de ciclo delta sea 0). Para determinar si son aceptables las transiciones entre los segmentos de formas de ondas, es aconsejable conocer el tiempo de ciclo del último impulso de un segmento. La fórmula para calcular dicho tiempo es la siguiente:

$$\text{tiempo de ciclo del último impulso} = \text{tiempo de ciclo inicial} + (\text{tiempo de ciclo delta} * (\text{número de impulsos} - 1))$$

Aunque el ejemplo simplificado descrito arriba sirve a título de introducción, es posible que en una aplicación real se necesiten perfiles de formas de ondas más complejos. Recuerde que:

- El tiempo de ciclo delta sólo se puede indicar en forma de número entero en microsegundos o milisegundos.
- En cada impulso se modifica el tiempo de ciclo.

El efecto de estos dos puntos es que el cálculo del valor delta del tiempo de ciclo para un determinado segmento puede exigir una aproximación iterativa. Puede ser necesaria cierta flexibilidad en el valor del tiempo de ciclo final o en el número de impulsos para un determinado segmento.

La duración de un segmento de perfil puede ser útil a la hora de determinar los valores correctos para la tabla de perfiles. Dicha duración se puede calcular utilizando la fórmula siguiente:

$$\text{Duración} = \text{nº de impulsos} * (\text{tiempo de ciclo inicial} + ((\text{tiempo de ciclo delta} / 2) * (\text{nº de impulsos} - 1)))$$

### Registros de control PTO/PWM

En la tabla 9-17 se describen los registros utilizados para controlar las funciones PTO/PWM. Mediante la tabla 9-18 es posible determinar rápidamente el valor que debe depositarse en el registro de control PTO/PWM para solicitar la operación deseada. Es preciso utilizar SMB67 para PTO/PWM 0 y SMB77 para PTO/PWM 1. Si se desea cargar un nuevo valor de contaje (SMD72 o SMD82), ancho de impulsos (SMW70 o SMW80) o tiempo de ciclo (SMW68 o SMW78), es necesario cargar tanto estos valores como el registro de control antes de ejecutar la operación PLS. Si se desea utilizar la función PTO multisegmento, antes de ejecutar la operación PLS también será preciso cargar el offset inicial (SMW168 o SMW178) de la tabla de perfiles y los valores de ésta.



Tabla 9-17 Registros de control PTO /PWM

Q0.0	Q0.1	Byte de estado
SM66.4	SM76.4	Interrupción anormal del perfil PTO debido a error de cálculo delta 0 = sin error; 1 = interrupción
SM66.5	SM76.5	Interrupción anormal del perfil PTO causada por el usuario 0 = sin interrupción; 1 = interrupción
SM66.6	SM76.6	Desbordamiento positivo/negativo PTO 0 = sin desbordamiento; 1 = desbordamiento positivo/negativo
SM66.7	SM76.7	PTO en vacío 0 = ejecución; 1 = PTO en vacío
Q0.0	Q0.1	Byte de control
SM67.0	SM77.0	Actualizar tiempo de ciclo PTO/PWM 0 = no actualizar; 1 = actualizar tiempo de ciclo
SM67.1	SM77.1	Actualizar tiempo de ancho de impulsos PWM 0 = no actualizar; 1 = actualizar ancho de impulsos
SM67.2	SM77.2	Actualizar valor de conteo de impulsos PTO 0 = no actualizar; 1 = actualizar valor de conteo de impulsos
SM67.3	SM77.3	Elegir base de tiempo PTO/PWM 0 = 1 $\mu$ s/ciclo; 1 = 1 ms/ciclo
SM67.4	SM77.4	Método de actualización PWM: 0 = actualización asíncrona, 1 = actualización síncrona
SM67.5	SM77.5	Función PTO 0 = función monosegmento; 1 = función multisegmento
SM67.6	SM77.6	Elegir modo PTO/PWM 0 = elige PTO; 1 = elige PWM
SM67.7	SM77.7	Habilitar PTO/PWM 0 = inhibe PTO/PWM; 1 = habilita PTO/PWM
Q0.0	Q0.1	Otros registros PTO/PWM
SMW68	SMW78	Valor del tiempo de ciclo PTO/PWM (margen: 2 a 65535)
SMW70	SMW80	Valor del ancho de impulsos PWM (margen: 0 a 65535)
SMD72	SMD82	Valor del conteo de impulsos PTO (margen: 1 a 4294967295)
SMB166	SMB176	Número del segmento que se está ejecutando (se utiliza sólo en la función multisegmento PTO)
SMW168	SMW178	Dirección inicial de la tabla de perfiles, expresada en forma de offset (en bytes) a partir de V0 (se utiliza sólo en la función multisegmento PTO)

Tabla 9-18 Referencias del byte de control PTO/PWM

Registro de control (valor hexadecimal)	Resultado de la operación PLS							
	Habilitar	Modo	Función PTO	Método de actualización PWM	Base de tiempo	Valor de contaje	Ancho de impulso	Tiempo de ciclo
16#81	Sí	PTO	Monosegmento		1 $\mu$ s/ciclo			Cargar
16#84	Sí	PTO	Monosegmento		1 $\mu$ s/ciclo	Cargar		
16#85	Sí	PTO	Monosegmento		1 $\mu$ s/ciclo	Cargar		Cargar
16#89	Sí	PTO	Monosegmento		1 ms/ciclo			Cargar
16#8C	Sí	PTO	Monosegmento		1 ms/ciclo	Cargar		
16#8D	Sí	PTO	Monosegmento		1 ms/ciclo	Cargar		Cargar
16#A0	Sí	PTO	Multisegmento		1 $\mu$ s/ciclo			
16#A8	Sí	PTO	Multisegmento		1 ms/ciclo			
16#D1	Sí	PWM		Síncrona	1 $\mu$ s/ciclo			Cargar
16#D2	Sí	PWM		Síncrona	1 $\mu$ s/ciclo		Cargar	
16#D3	Sí	PWM		Síncrona	1 $\mu$ s/ciclo		Cargar	Cargar
16#D9	Sí	PWM		Síncrona	1 ms/ciclo			Cargar
16#DA	Sí	PWM		Síncrona	1 ms/ciclo		Cargar	
16#DB	Sí	PWM		Síncrona	1 ms/ciclo		Cargar	Cargar

### Inicialización y secuencias de las funciones PTO/PWM

Para comprender mejor el funcionamiento de PTO y PWM, se describe seguidamente su inicialización paso a paso, así como las correspondientes operaciones. En las descripciones se utiliza la salida de impulsos Q0.0 a título de ejemplo. En las explicaciones acerca de la inicialización se supone que el sistema de automatización S7-200 se encuentra en modo RUN y que, por consiguiente, la marca del primer ciclo es verdadera. En otro caso o si se debe inicializar nuevamente la función PTO/PWM, es preciso llamar a la rutina de inicialización usando una condición diferente a la marca del primer ciclo.

### Inicializar la función PWM

Siga los pasos siguientes para inicializar la función PWM para Q0.0:

1. Con la marca del primer ciclo (SM0.1), inicialice la salida y llame a la subrutina necesaria para ejecutar la inicialización. Puesto que se utiliza una llamada a subrutina, los siguientes ciclos ya no llaman a la misma. Así se acorta el tiempo de ciclo y el programa queda mejor estructurado.
2. En la subrutina de inicialización, cargue 16#D3 para PWM en SMB67 para incrementar en microsegundos (o cargue 16#DB para PWM si desea incrementar en milisegundos). Estos valores ajustan el byte de control para habilitar la operación PTO/PWM, seleccionan la operación PWM, deciden si se incrementa en micro o milisegundos y ajustan los valores para actualizar el ancho de impulsos y el tiempo de ciclo.
3. Cargue el tiempo de ciclo deseado en SMW68 (valor de palabra).
4. Cargue el ancho de impulsos deseado en SMW70 (valor de palabra).
5. Ejecute la operación PLS para que el S7-200 programe el generador PTO/PWM.
6. Cargue el valor 16#D2 en SM67 para incrementar en microsegundos (o 16#DA si desea incrementar en milisegundos). Así se precarga un nuevo valor del byte de control para los posteriores cambios del ancho de impulsos.
7. Finalice la subrutina.

### Cambiar el ancho de impulsos para las salidas PWM

Siga los pasos siguientes para cambiar el ancho de impulsos para salidas PWM en una subrutina: (Se supone que SMB67 se ha precargado con un valor de 16#D2 ó 16#DB).

1. Llame a una subrutina para cargar el ancho de impulsos deseado en SMW70 (valor de palabra).
2. Ejecute la operación PLS para que el S7-200 programe el generador PTO/PWM.
3. Finalice la subrutina.

### Inicializar la función PTO monosegmento

Siga los pasos siguientes para inicializar la función PTO:

1. Con la marca del primer ciclo (SM0.1), inicialice la salida y llame a la subrutina necesaria para ejecutar la inicialización. Así se acorta el tiempo de ciclo y el programa queda mejor estructurado.
2. En la subrutina de inicialización, cargue 16#85 para PTO en SMB67 para incrementar en microsegundos (o cargue 16#8D para PTO si desea incrementar en milisegundos). Estos valores ajustan el byte de control para habilitar la operación PTO/PWM, seleccionan la operación PTO, deciden si se incrementa en micro o milisegundos y ajustan los valores para actualizar el ancho de impulsos y el tiempo de ciclo.
3. Cargue el tiempo de ciclo deseado en SMW68 (valor de palabra).
4. Cargue el tiempo de ciclo deseado en SMD72 (valor de palabra doble).
5. Este paso es opcional. Si desea ejecutar una operación asociada en cuanto termine la operación Tren de impulsos, puede programar una interrupción asociando el evento Fin del tren de impulsos (clase de interrupción 19) a una rutina de interrupción (mediante la operación ATCH) y ejecutando la operación Habilitar todos los eventos de interrupción (ENI). Para obtener más información acerca de la utilización de interrupciones, consulte el apartado 9.16.
6. Ejecute la operación PLS para que el S7-200 programe el generador PTO/PWM.
7. Finalice la subrutina.

### Cambiar el tiempo de ciclo PTO en la función monosegmento

Siga los pasos siguientes para cambiar el tiempo de ciclo PTO en una subrutina o en una rutina de interrupción al utilizar la función PTO monosegmento:

1. Cargue 16#81 para PTO en SMB67 para incrementar en microsegundos (o cargue 16#89 para PTO si desea incrementar en milisegundos). Estos valores ajustan el byte de control para habilitar la operación PTO/PWM, seleccionan la operación PTO, deciden si se incrementa en micro o milisegundos y ajustan los valores para actualizar el tiempo de ciclo.
2. Cargue el tiempo de ciclo deseado en SMW68 (valor de palabra).
3. Ejecute la operación PLS para que el S7-200 programe el generador PTO/PWM. Si se está ejecutando un PTO, la CPU debe finalizarlo antes de que pueda comenzar la salida de la forma de onda PTO con el tiempo de ciclo actualizado.
4. Finalice la subrutina o la rutina de interrupción.

### **Cambiar el valor de contaje de impulsos PTO en la función monosegmento**

Siga los pasos siguientes para cambiar el valor de contaje de impulsos PTO en una subrutina o en una rutina de interrupción al utilizar la función PTO monosegmento:

1. Cargue 16#84 para PTO en SMB67 para incrementar en microsegundos (o cargue 16#8C para PTO si desea incrementar en milisegundos). Estos valores ajustan el byte de control para habilitar la operación PTO/PWM, seleccionan la operación PTO, deciden si se incrementa en micro o milisegundos y ajustan los valores para actualizar el valor de contaje de impulsos.
2. Cargue el tiempo de ciclo deseado en SMD72 (valor de palabra doble).
3. Ejecute la operación PLS para que el S7-200 programe el generador PTO/PWM. Si se está ejecutando un PTO, la CPU debe finalizarlo antes de que pueda comenzar la salida de la forma de onda PTO con el tiempo de ciclo actualizado.
4. Finalice la subrutina o la rutina de interrupción.

### **Cambiar el tiempo de ciclo PTO y el contaje de impulsos en la función monosegmento**

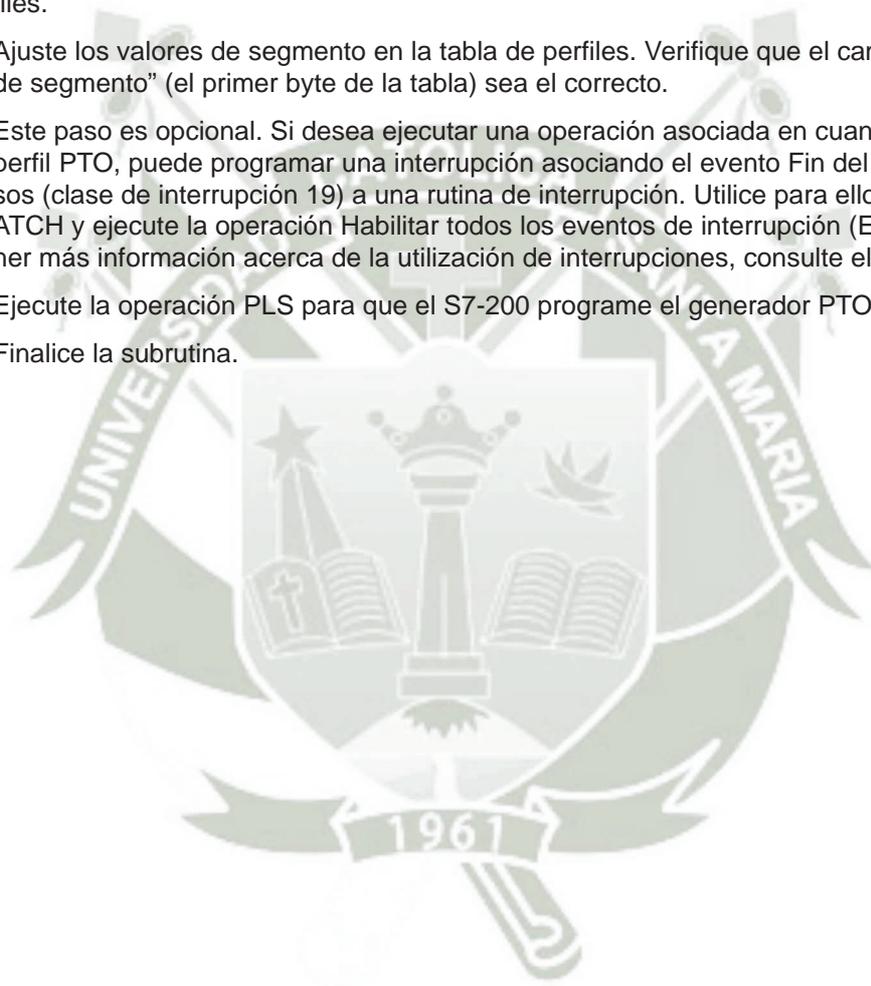
Siga los pasos siguientes para cambiar el tiempo de ciclo PTO y el contaje de impulsos en una subrutina o en una rutina de interrupción al utilizar la función PTO monosegmento:

1. Cargue 16#85 para PTO en SMB67 para incrementar en microsegundos (o cargue 16#8D para PTO si desea incrementar en milisegundos). Estos valores ajustan el byte de control para habilitar la operación PTO/PWM, seleccionan la operación PTO, deciden si se incrementa en micro o milisegundos y ajustan los valores para actualizar el tiempo de ciclo y el valor de contaje de impulsos.
2. Cargue el tiempo de ciclo deseado en SMW68 (valor de palabra).
3. Cargue el tiempo de ciclo deseado en SMD72 (valor de palabra doble).
4. Ejecute la operación PLS para que el S7-200 programe el generador PTO/PWM. Si se está ejecutando un PTO, la CPU debe finalizarlo antes de que pueda comenzar la salida de la forma de onda PTO con el tiempo de ciclo actualizado.
5. Finalice la subrutina o la rutina de interrupción.

## Inicializar la función PTO multisegmento

Siga los pasos siguientes para inicializar la función PTO:

1. Con la marca del primer ciclo (SM0.1), inicialice la salida y llame a la subrutina necesaria para ejecutar la inicialización. Así se acorta el tiempo de ciclo y el programa queda mejor estructurado.
2. En la subrutina de inicialización, cargue 16#A0 para PTO en SMB67 para incrementar en microsegundos (o cargue 16#A8 para PTO si desea incrementar en milisegundos). Estos valores ajustan el byte de control para habilitar la operación PTO/PWM, seleccionan la función PTO multisegmento y determinan si se debe incrementar en microsegundos o en milisegundos.
3. Cargue en SMW168 (valor de palabra) el offset inicial en la memoria V de la tabla de perfiles.
4. Ajuste los valores de segmento en la tabla de perfiles. Verifique que el campo "Número de segmento" (el primer byte de la tabla) sea el correcto.
5. Este paso es opcional. Si desea ejecutar una operación asociada en cuanto termine el perfil PTO, puede programar una interrupción asociando el evento Fin del tren de impulsos (clase de interrupción 19) a una rutina de interrupción. Utilice para ello la operación ATCH y ejecute la operación Habilitar todos los eventos de interrupción (ENI). Para obtener más información acerca de la utilización de interrupciones, consulte el apartado 9.16.
6. Ejecute la operación PLS para que el S7-200 programe el generador PTO/PWM.
7. Finalice la subrutina.



### Ejemplo de la función Modulación del ancho de impulsos (PWM)

La figura 9-21 muestra un ejemplo de la operación Modulación del ancho de impulsos (PWM).

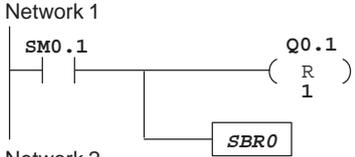
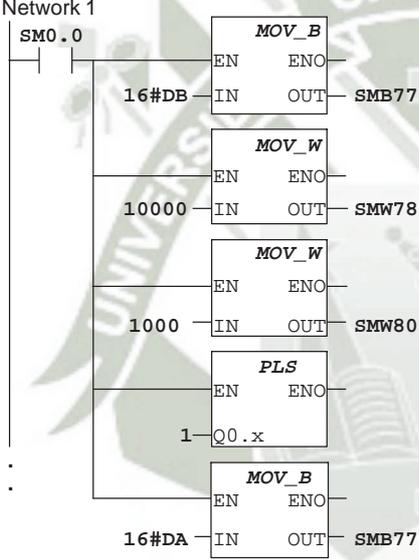
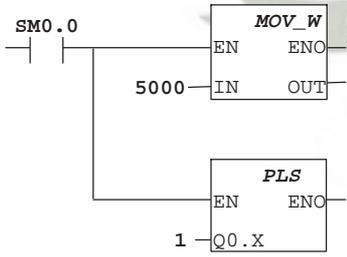
KOP	AWL
PROGRAMA PRINCIPAL OB1	
<p>Network 1</p>  <p>Network 2</p>  <p>·</p> <p>·</p>	<p>Network 1</p> <pre>LD SM0.1 R Q0.1, 1 CALL 0</pre> <p>Network 2</p> <pre>LD M0.0 EU CALL 1</pre> <p>·</p> <p>·</p>
SUBROUTINA 0	
<p>Network 1</p>  <p>·</p> <p>·</p>	<p>Comienzo subrutina 0.</p> <p>Ajustar byte de control:          – Elegir función PWM          – Seleccionar incrementos en ms para la actualización síncrona          – Ajustar valores para ancho de impulsos y tiempo de ciclo          – Habilitar función PWM</p> <p>Ajustar el tiempo de ciclo a 10.000 ms.</p> <p>Ajustar el ancho de impulsos a 1.000 ms.</p> <p>Llamar a la operación PWM.          PLS 1 =&gt; Q 0.1</p> <p>Precargar el byte de control para los cambios posteriores del ancho de impulsos.</p>
SUBROUTINA 1	
<p>Network 1</p> 	<p>Comienzo subrutina 1.</p> <p>Ajustar el ancho de impulsos a 5000 ms.</p> <p>Confirmar el cambio del ancho de impulsos.</p>

Figura 9-21 Ejemplo de operaciones rápidas de salida con modulación del ancho de impulsos

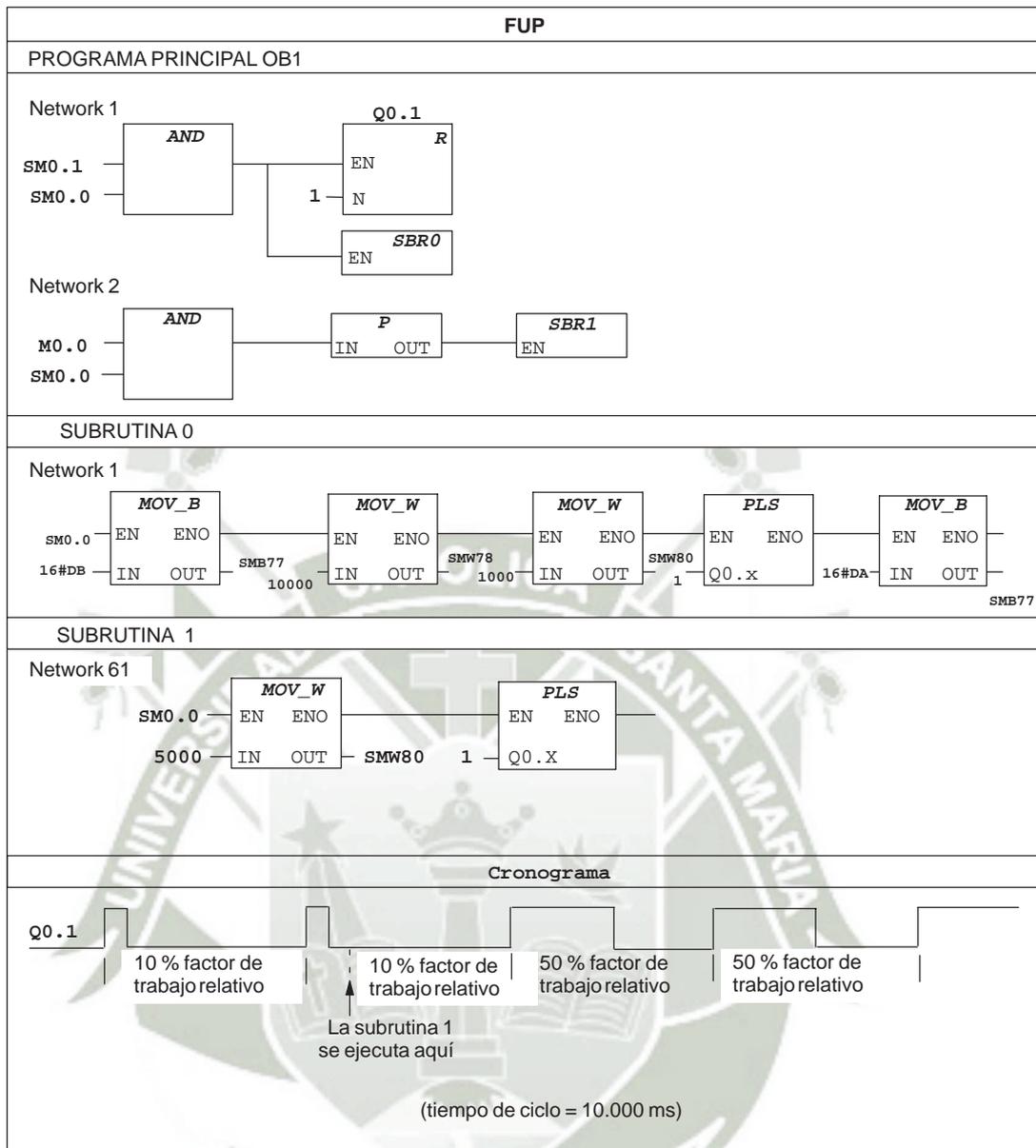


Figura 9-21 Ejemplo de operaciones rápidas de salida con modulación del ancho de impulsos (continuación)

Ejemplo de la función Tren de impulsos en modo monosegmento

KOP		AWL
PROGRAMA PRINCIPAL OB1		
<p>Network 1</p>	<p>Reducir en el primer ciclo el bit de la imagen del proceso y llamar a la subrutina 0.</p>	<p>Network 1</p> <pre>LD SM0.1 R Q0.0, 1 CALL 0</pre>
SUBROUTINA 0		
<p>Network 1</p>	<p>Ajustar byte de control:                      – Elegir función PTO                      – Elegir incremento en milisegundos                      – Ajustar valores para conteo de impulsos y tiempo de ciclo                      – Habilitar función PTO</p> <p>Ajustar tiempo de ciclo a 500 ms.</p> <p>Ajustar valor de conteo a 4 impulsos.</p> <p>Definir que la rutina de interrupción 3 sea la que procese las interrupciones completas PTO.</p> <p>Habilitar todos los eventos de interrupción.</p> <p>Llamar operación PTO.                      PLS 0 =&gt; Q0.0</p> <p>Precargar byte de control para los cambios de tiempo de ciclo subsiguientes.</p>	<p>Network 1</p> <pre>LD SM0.0 MOVB 16#8D, SMB67 MOVW 500, SMW68 MOVD 4, SMD72 ATCH 3, 19 ENI PLS 0, MOVB 16#89, SMB67</pre>

Figura 9-22 Ejemplo de un tren de impulsos utilizando la función monosegmento en el área de marcas especiales

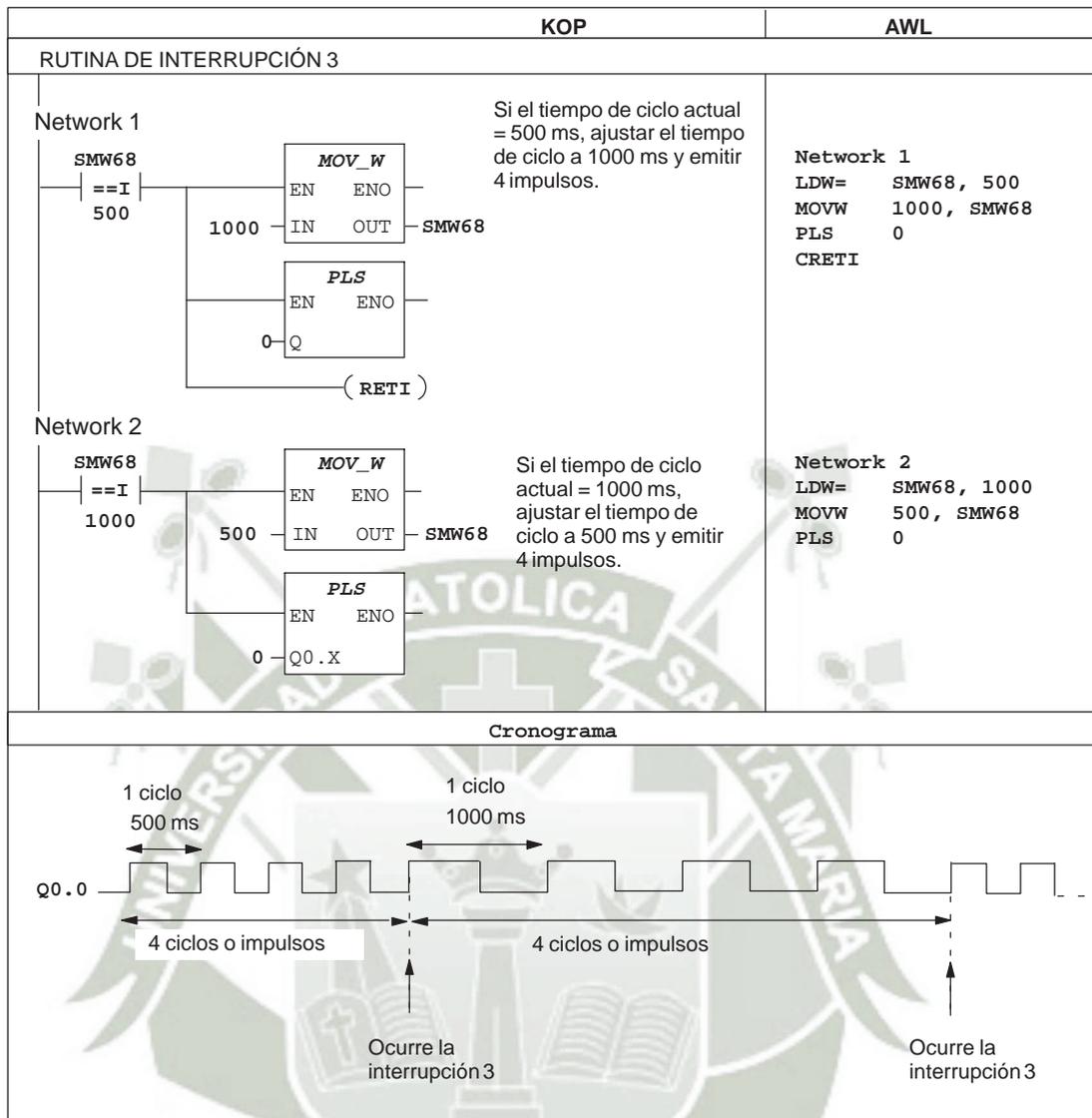


Figura 9-22 Ejemplo de un tren de impulsos utilizando la función monosegmento (continuación)

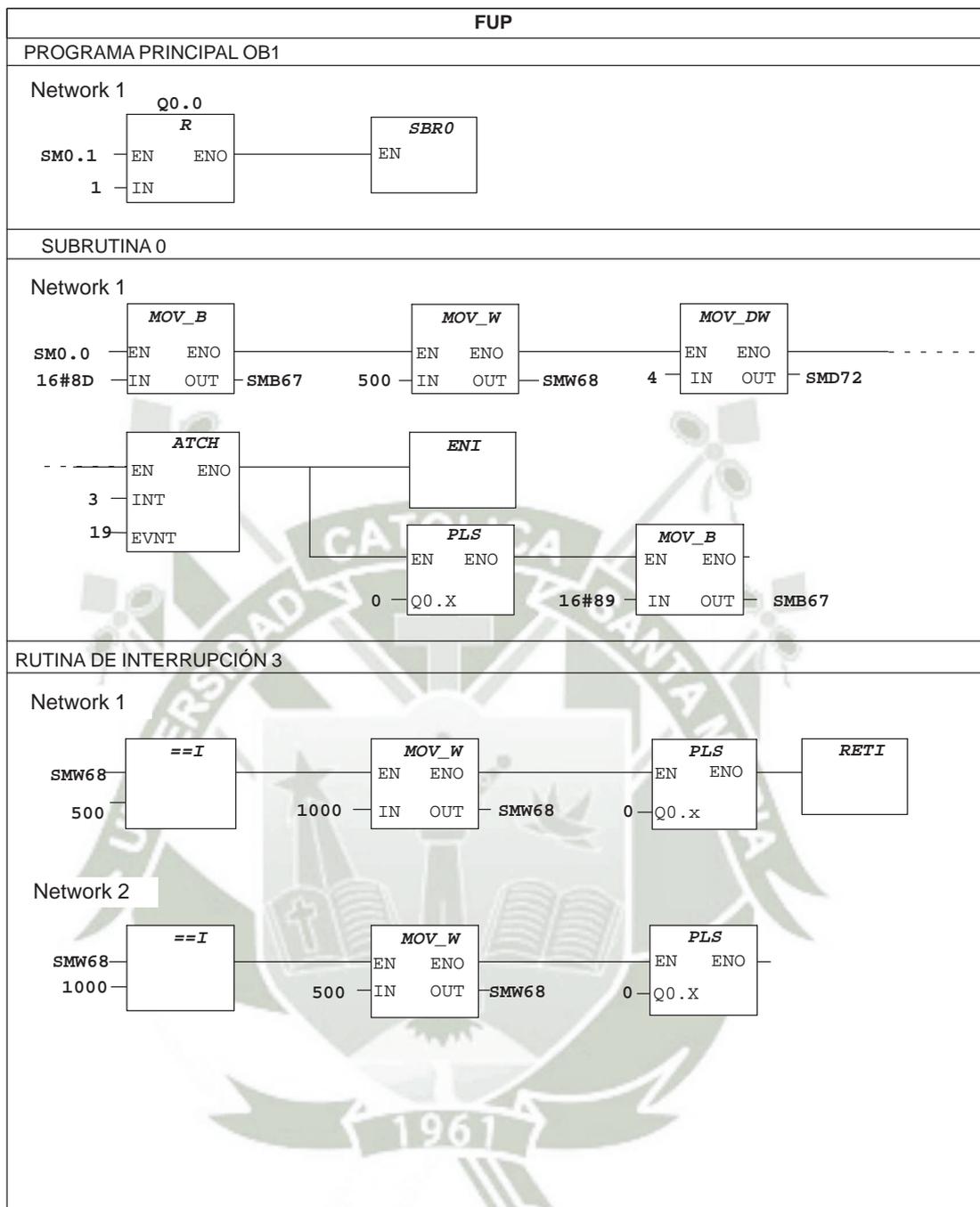


Figura 9-22 Ejemplo de un tren de impulsos utilizando la función monosegmento (continuación)

Ejemplo de la función Tren de impulsos en modo multisegmento

KOP		AWL
PROGRAMA PRINCIPAL OB1		
<p>Network 1</p>	<p>Reducir en el primer ciclo el bit de la imagen del proceso y llamar a la subrutina 0.</p>	<p>Network 1</p> <pre>LD SM0.1 R Q0.0, 1 CALL 0</pre>
SUBROUTINA 0		
<p>Network 1</p>	<p>Ajustar byte de control:                      – Elegir función PTO                      – Seleccionar el modo multisegmento</p> <p>– Seleccionar incrementos en <math>\mu</math>s                      – Habilitar función PTO</p> <p>Indicar que la dirección inicial de la tabla de perfiles sea V500.</p> <p>Ajustar a 3 el número de segmentos de la tabla.</p> <p>Ajustar a 500 <math>\mu</math>s el tiempo de ciclo inicial del segmento #1.</p> <p>Ajustar a <math>-2 \mu</math>s el tiempo de ciclo delta del segmento #1.</p> <p>Ajustar a 200 el número de impulsos del segmento #1.</p>	<p>Network 1</p> <pre>LD SM0.0 MOVB 16#A0, SMB67 MOVW 500, SMW168 MOVB 3, VB500 MOVW 500, VW501 MOVW -2, VW503 MOVD 200, VD505</pre>

Figura 9-23 Ejemplo de un tren de impulsos utilizando la función multisegmento

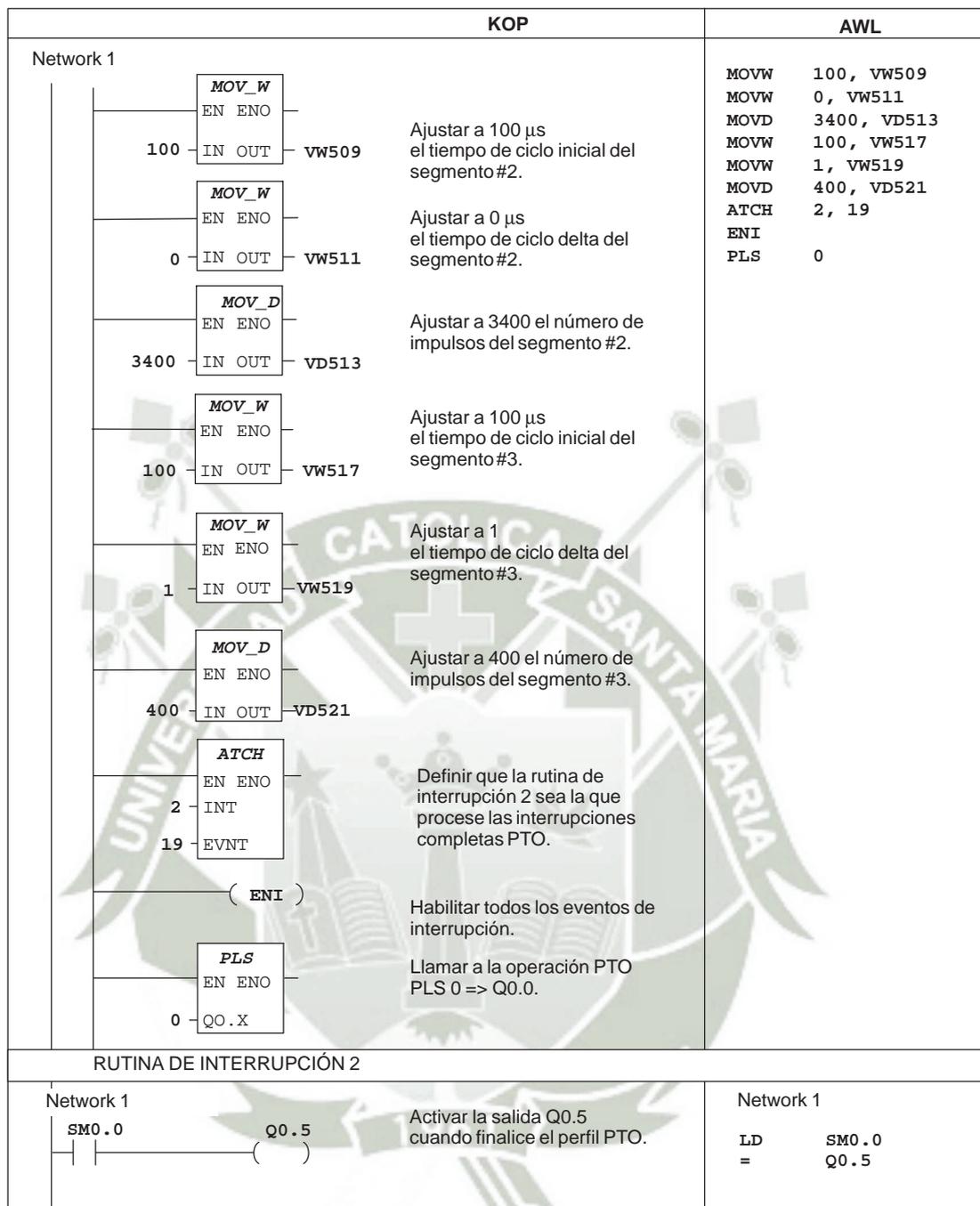
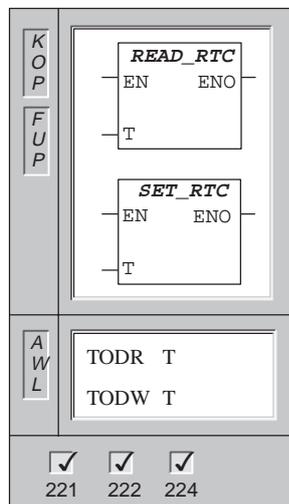


Figura 9-23 Ejemplo de un tren de impulsos utilizando la función multisegmento (continuación)

## 9.7 Operaciones de reloj (SIMATIC)

### Leer reloj de tiempo real, Ajustar reloj de tiempo real



La operación **Leer reloj de tiempo real** lee la hora y fecha actuales del reloj y carga ambas en un búfer de 8 bytes (que comienza en la dirección T).

La operación **Ajustar reloj de tiempo real** escribe en el reloj la hora y fecha actuales que están cargadas en un búfer de 8 bytes (que comienza en la dirección T).

En AWL, dichas operaciones se representan mediante las instrucciones TODR (Leer reloj de tiempo real) y TODW (Escribir reloj de tiempo real).

TODR: Condiciones de error que ponen ENO a 0: SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto), 000C (falta cartucho de reloj)

TODW: Condiciones de error que ponen ENO a 0: SM 4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto), 0007 (error de datos TOD), 000C (falta cartucho de reloj)

Entradas/salidas	Operandos	Tipos de datos
T	VB, IB, QB, MB, SMB, SB, LB, *VD, *AC, *LD	BYTE

La figura 9-24 muestra el formato del búfer de tiempo (T).

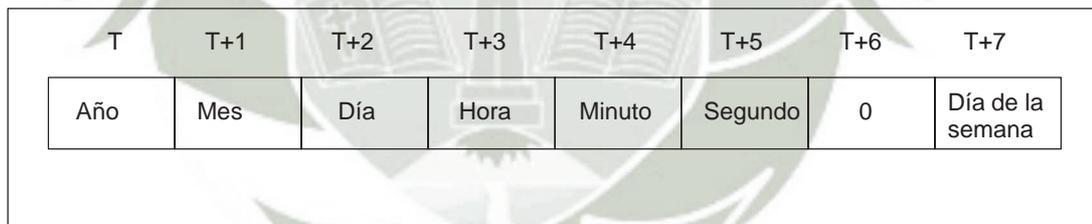


Figura 9-24 Formato del búfer de tiempo

El reloj de tiempo real se inicializa con la siguiente fecha y hora tras un corte de alimentación prolongado o una pérdida de memoria:

Fecha: 01-Ene-90  
Hora: 00:00:00  
Día de la semana Domingo

El reloj de tiempo real de la CPU S7-200 utiliza sólo los dos dígitos menos significativos para representar el año. Por tanto, el año 2000 se representará como "00" (el reloj pasará de 99 a 00).

Todos los valores de la fecha y la hora se deben codificar en BCD (p.ej. 16#97 para el año 1997). Utilice los siguientes formatos de datos:

Año/Mes	aamm	aa – 0 a 99	mm – 1 a 12
Día/Hora	ddhh	dd – 1 a 31	hh – 0 a 23
Minutos/Segundos	mmss	mm – 0 a 59	ss – 0 a 59
Día de la semana	d	d – 0 a 7	1 = Domingo 0 = desactiva el día de la semana (permanece 0)

---

#### Nota

La CPU S7-200 no comprueba si el día de la semana coincide con la fecha. Así puede ocurrir que se acepten fechas no válidas, p.ej. el 30 de febrero. Asegúrese de que los datos introducidos sean correctos.

No utilice nunca las operaciones TODR y TODW en el programa principal y en una rutina de interrupción a la vez. Si se está procesando una operación TODR/TODW y se intenta ejecutar simultáneamente otra operación TODR/TODW en una rutina de interrupción, ésta no se procesará. SM4.3 se activa indicando que se intentaron dos accesos simultáneos al reloj (error no fatal 0007).

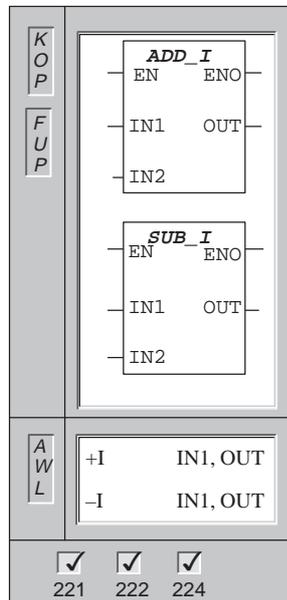
El sistema de automatización S7-200 no utiliza la información relativa al año de ninguna forma y no se verá afectado por el cambio de siglo (en el año 2000). No obstante, si en los programas de usuario se utilizan operaciones aritméticas o de comparación con el valor del año, se deberá tener en cuenta la representación de dos dígitos y el cambio de siglo.

Los años bisiestos se tratan correctamente hasta el año 2096.

---

## 9.8 Operaciones aritméticas con enteros (SIMATIC)

### Sumar y restar enteros de 16 bits



Las operaciones **Sumar enteros de 16 bits** y **Restar enteros de 16 bits** suman/restan dos enteros de 16 bits, arrojando un resultado de 16 bits (OUT).

En KOP y FUP:  $IN1 + IN2 = OUT$   
 $IN1 - IN2 = OUT$

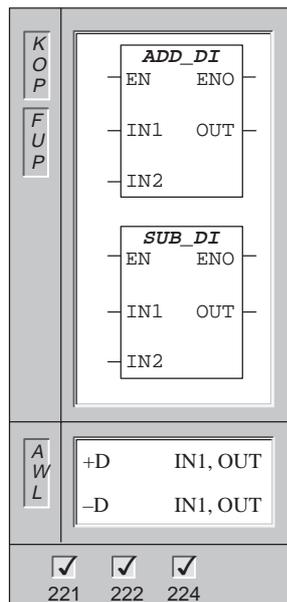
En AWL:  $IN1 + OUT = OUT$   
 $OUT - IN1 = OUT$

Condiciones de error que ponen ENO a 0: SM1.1 (desbordamiento), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero); SM1.1 (desbordamiento); SM1.2 (negativo)

Entradas/salidas	Operandos	Tipos de datos
IN1, IN2	VW, IW, QW, MW, SW, SMW, LW, AIW, T, C, AC, constante, *VD, *AC, *LD	INT
OUT	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, *VD, *AC, *LD	INT

### Sumar y restar enteros de 32 bits



Las operaciones **Sumar enteros de 32 bits** y **Restar enteros de 32 bits** suman/restan dos enteros de 32 bits, arrojando un resultado de 32 bits (OUT).

En KOP y FUP:  $IN1 + IN2 = OUT$   
 $IN1 - IN2 = OUT$

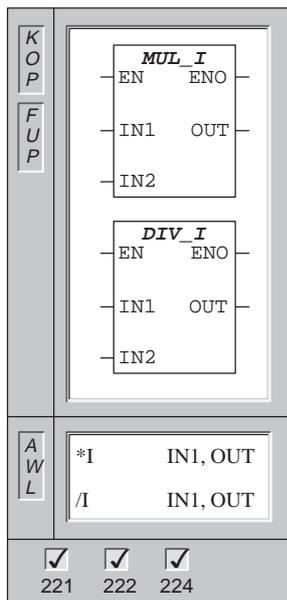
En AWL:  $IN1 + OUT = OUT$   
 $OUT - IN1 = OUT$

Condiciones de error que ponen ENO a 0: SM1.1 (desbordamiento), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero); SM1.1 (desbordamiento); SM1.2 (negativo)

Entradas/salidas	Operandos	Tipos de datos
IN1, IN2	VD, ID, QD, MD, SMD, SD, LD, AC, HC, constante, *VD, *AC, *LD	DINT
OUT	VD, ID, QD, MD, SM, SD, LD, AC, *VD, *AC, *LD	DINT

### Multiplicar y dividir enteros de 16 bits



La operación **Multiplicar enteros de 16 bits** multiplica dos números enteros de 16 bits, arrojando un producto de 16 bits.

La operación **Dividir enteros de 16 bits** divide dos números enteros de 16 bits, arrojando un cociente de 16 bits. No se guarda ningún resto.

La marca de desbordamiento se activa si el resultado es mayor que una salida de palabra.

En KOP y FUP:  $IN1 * IN2 = OUT$   
 $IN1 / IN2 = OUT$

En AWL:  $IN1 * OUT = OUT$   
 $OUT / IN1 = OUT$

Condiciones de error que ponen ENO a 0: SM1.1 (desbordamiento), SM1.3 (división por cero), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

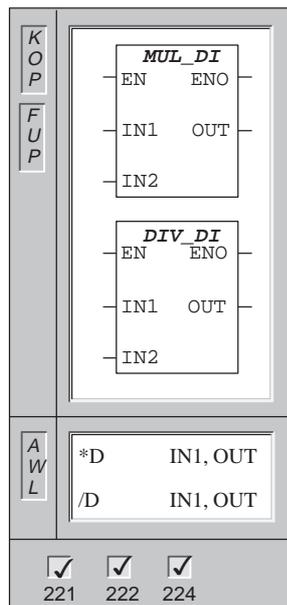
Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero); SM1.1 (desbordamiento); SM1.2 (negativo); SM1.3 (división por cero)

Si SM1.1 (marca de desbordamiento) se activa durante una operación de multiplicación o de división, no se escribe en la salida y todos los demás bits de estado aritméticos se ponen a 0.

Si SM1.3 (división por cero) se activa durante una operación de división, permanecerán inalterados los demás bits aritméticos de estado, así como los operandos de entrada originales. En otro caso, todos los bits aritméticos de estado asistidos contendrán el estado válido al finalizar la operación aritmética.

Entradas/salidas	Operandos	Tipos de datos
IN1, IN2	VW, IW, QW, MW, SW, SMW, LW, AIW, T, C, AC, constante, *VD, *AC, *LD	INT
OUT	VW, QW, IW, MW, SW, SMW, LW, T, C, AC, *VD, *LD, *AC	INT

### Multiplicar y dividir enteros de 32 bits



La operación **Multiplicar enteros de 32 bits** multiplica dos enteros de 32 bits, arrojando un producto de 32 bits.

La operación **Dividir enteros de 32 bits** divide dos enteros de 32 bits, arrojando un cociente de 32 bits. No se guarda ningún resto.

En KOP y FUP:  $IN1 * IN2 = OUT$   
 $IN1 / IN2 = OUT$

En AWL:  $IN1 * OUT = OUT$   
 $OUT / IN1 = OUT$

Condiciones de error que ponen ENO a 0: SM1.1 (desbordamiento), SM1.3 (división por cero), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

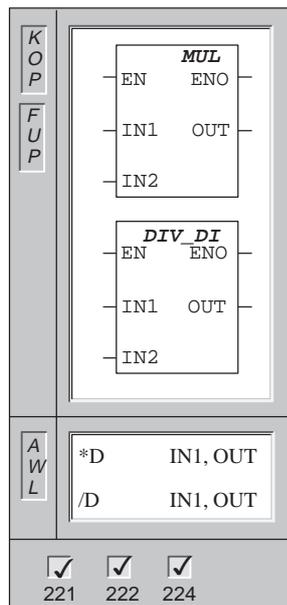
Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero); SM1.1 (desbordamiento); SM1.2 (negativo); SM1.3 (división por cero)

Si SM1.1 (marca de desbordamiento) se activa durante una operación de multiplicación o de división, no se escribe en la salida y todos los demás bits de estado aritméticos se ponen a 0.

Si SM1.3 (división por cero) se activa durante una operación de división, permanecerán inalterados los demás bits aritméticos de estado, así como los operandos de entrada originales. En otro caso, todos los bits aritméticos de estado asistidos contendrán el estado válido al finalizar la operación aritmética.

Entradas/salidas	Operandos	Tipos de datos
IN1, IN2	VD, ID, QD, MD, SMD, SD, LD, HC, AC, constante, *VD, *AC, *LD	DINT
OUT	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD, *AC	DINT

## Multiplicar y dividir enteros de 32 bits



La operación **Multiplicar enteros de 16 bits a enteros de 32 bits** multiplica dos números enteros de 16 bits, arrojando un producto de 32 bits.

La operación **Dividir enteros de 16 bits a enteros de 32 bits** divide dos números enteros de 16 bits, arrojando un resultado de 32 bits compuesto de un cociente de 16 bits (los menos significativos) y un resto de 16 bits (los más significativos).

En la operación AWL de multiplicación, la palabra menos significativa (16 bits) del OUT de 32 bits se utiliza como uno de los factores.

En la operación AWL de división, la palabra menos significativa (16 bits) del OUT de 32 bits se utiliza como dividendo.

En KOP y FUP:  $IN1 * IN2 = OUT$   
 $IN1 / IN2 = OUT$

En AWL:  $IN1 * OUT = OUT$   
 $OUT / IN1 = OUT$

Condiciones de error que ponen ENO a 0: SM1.1 (desbordamiento), SM1.3 (división por cero), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero); SM1.1 (desbordamiento); SM1.2 (negativo); SM1.3 (división por cero)

Si SM1.3 (división por cero) se activa durante una operación de división, permanecerán inalterados los demás bits aritméticos de estado, así como los operandos de entrada originales. En otro caso, todos los bits aritméticos de estado asistidos contendrán el estado válido al finalizar la operación aritmética.

Entradas/salidas	Operandos	Tipos de datos
IN1, IN2	VW, IW, QW, MW, SW, SMW, LW, AC, AIW, T, C, constante, *VD, *AC, *LD	INT
OUT	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD, *AC	DINT

Ejemplos de operaciones aritméticas

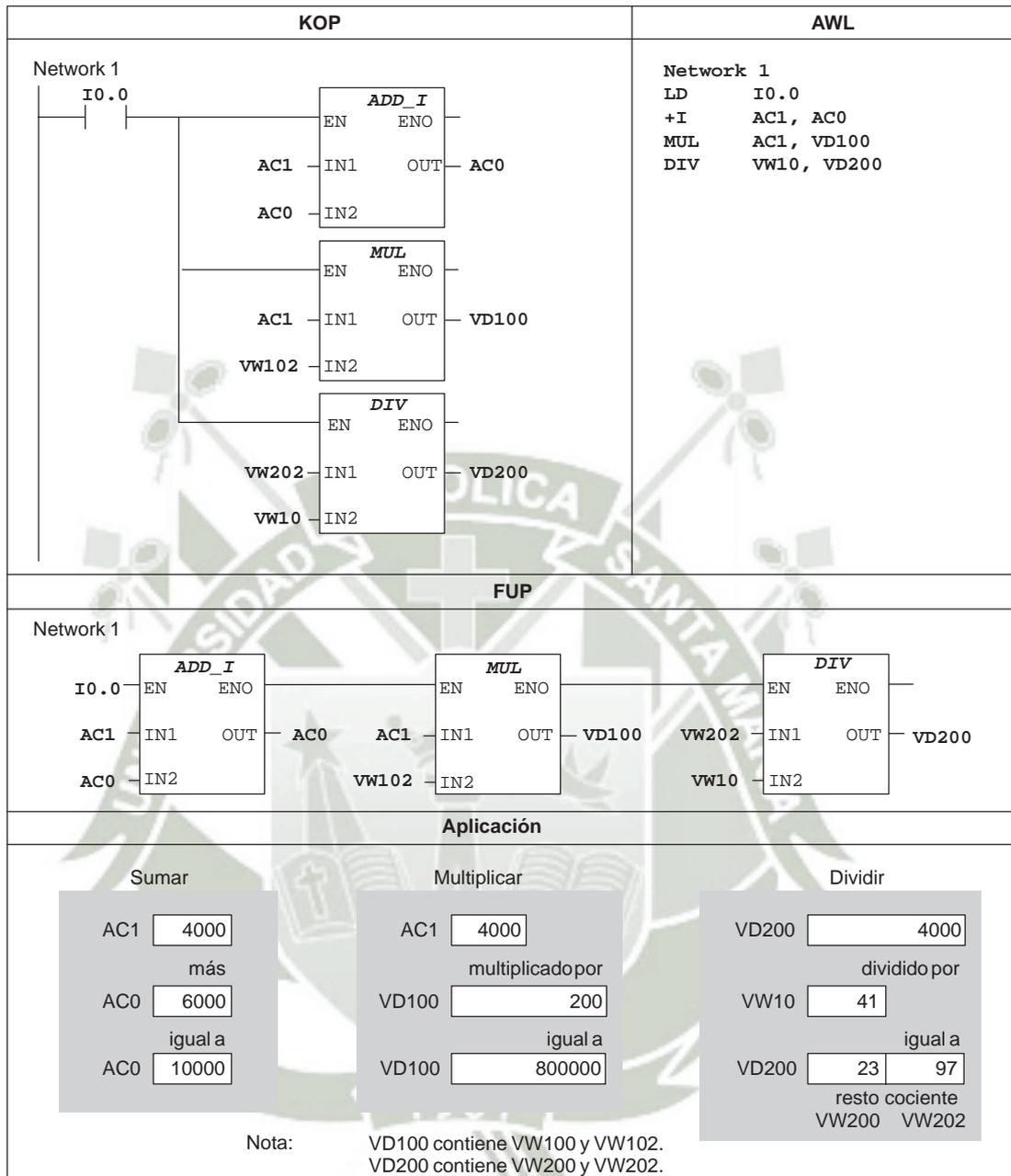
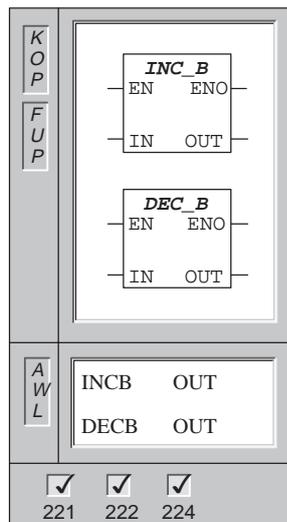


Figura 9-25 Ejemplos de operaciones aritméticas con enteros en KOP, AWL y FUP

### Incrementar y Decrementar byte



Las operaciones **Incrementar byte** y **Decrementar byte** suman/restan 1 al byte de entrada (IN) y depositan el resultado en la variable indicada por OUT.

Las operaciones Incrementar byte y Decrementar byte no llevan signo.

En KOP y FUP:  $IN + 1 = OUT$   
 $IN - 1 = OUT$

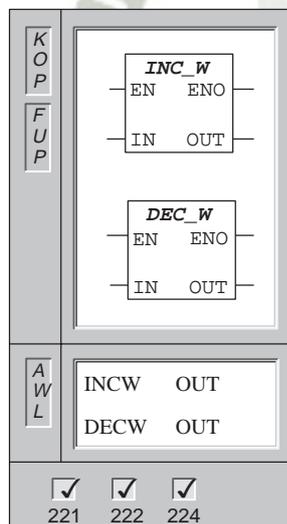
En AWL:  $OUT + 1 = OUT$   
 $OUT - 1 = OUT$

Condiciones de error que ponen ENO a 0: SM1.1 (desbordamiento), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero); SM1.1 (desbordamiento)

Entradas/salidas	Operandos	Tipos de datos
IN	VB, IB, QB, MB, SB, SMB, LB, AC, constante, *VD, *AC, *LD	BYTE
OUT	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD	BYTE

### Incrementar y decrementar palabra



Las operaciones **Incrementar palabra** y **Decrementar palabra** suman/restan 1 al valor de la palabra de entrada (IN) y depositan el resultado en OUT.

Las operaciones Incrementar palabra y Decrementar palabra llevan signo ( $16\#7FFF > 16\#8000$ ).

En KOP y FUP:  $IN + 1 = OUT$   
 $IN - 1 = OUT$

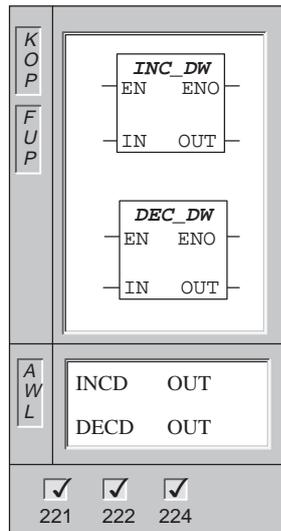
En AWL:  $OUT + 1 = OUT$   
 $OUT - 1 = OUT$

Condiciones de error que ponen ENO a 0: SM1.1 (desbordamiento), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero); SM1.1 (desbordamiento); SM1.2 (negativo)

Entradas/salidas	Operandos	Tipos de datos
IN	VW, IW, QW, MW, SW, SMW, AC, AIW, LW, T, C, constante, *VD, *AC, *LD	INT
OUT	VW, IW, QW, MW, SW, SMW, LW, AC, T, C, *VD, *AC, *LD	INT

### Incrementar y decrementar palabra doble



Las operaciones **Incrementar palabra doble** y **Decrementar palabra doble** suman/restan 1 al valor de la palabra doble de entrada (IN) y depositan el resultado en OUT.

En KOP y FUP:  $IN + 1 = OUT$   
 $IN - 1 = OUT$

Las operaciones Incrementar palabra doble y Decrementar palabra doble llevan signo (16#7FFFFFFF > 16#80000000).

En AWL:  $OUT + 1 = OUT$   
 $OUT - 1 = OUT$

Condiciones de error que ponen ENO a 0: SM1.1 (desbordamiento), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero); SM1.1 (desbordamiento); SM1.2 (negativo)

Entradas/salidas	Operandos	Tipos de datos
IN	VD, ID, QD, MD, SD, SMD, LD, AC, HC, constante, *VD, *AC, *LD	DINT
OUT	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *AC, *LD	DINT

### Ejemplos de las operaciones Incrementar y Decrementar

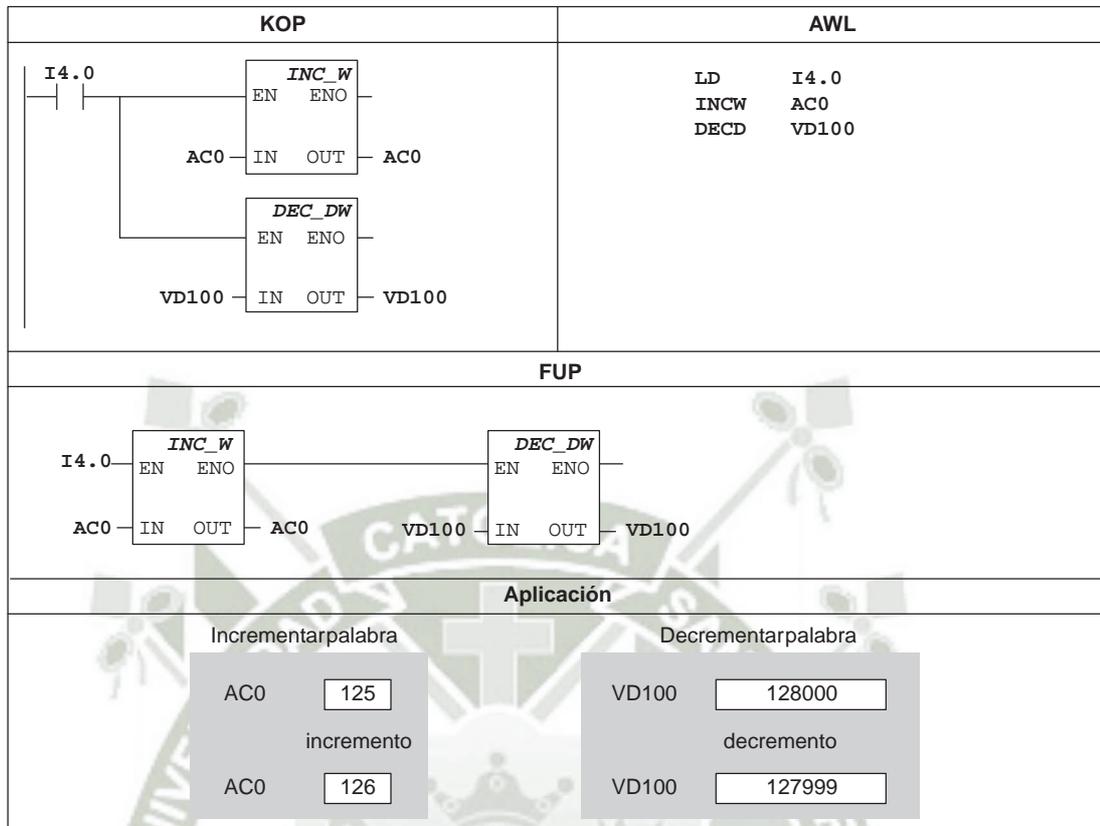
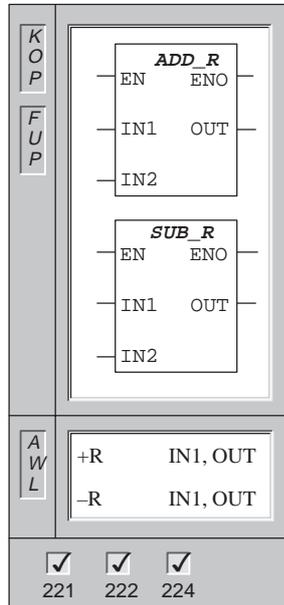


Figura 9-26 Ejemplos de las operaciones Incrementar y Decrementar en KOP, AWL y FUP

## 9.9 Operaciones aritméticas con números reales (SIMATIC)

### Sumar y restar reales



Las operaciones **Sumar reales** y **Restar reales** suman/restan dos números reales de 32 bits, dando como resultado un número real de 32 bits (OUT).

En KOP y FUP:  $IN1 + IN2 = OUT$   
 $IN1 - IN2 = OUT$

En AWL:  $IN1 + OUT = OUT$   
 $OUT - IN1 = OUT$

Condiciones de error que ponen ENO a 0: SM1.1 (desbordamiento), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero); SM1.1 (desbordamiento); SM1.2 (negativo)

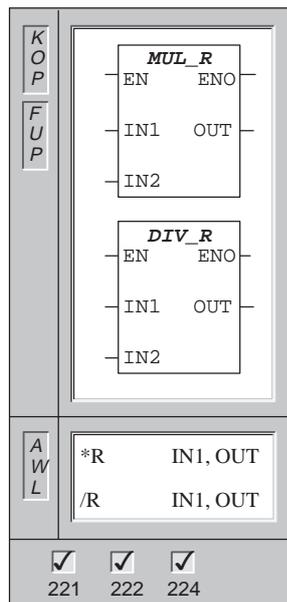
SM1.1 se utiliza para indicar errores de desbordamiento y valores no válidos. Si SM1.1 se activa, el estado de SM1.0 y de SM1.2 no será válido y no se alterarán los operandos de entrada originales. Si SM1.1 no se activa, la operación aritmética habrá finalizado con un resultado válido, y tanto SM1.0 como SM1.2 contendrán un estado válido.

Entradas/salidas	Operandos	Tipos de datos
IN1, IN2	VD, ID, QD, MD, SD, SMD, AC, LD, constante, *VD, *AC, *LD	REAL
OUT	VD, ID, QD, MD, SD, SMD, AC, LD, *VD, *AC, *LD	REAL

#### Nota

Los números reales (o números en coma flotante) se representan en el formato descrito en la norma ANSI/IEEE 754-1985 (precisión sencilla). Para obtener más información al respecto, consulte dicha norma.

## Multiplicar y dividir reales



La operación **Multiplicar reales** multiplica dos números reales de 32 bits, dando como resultado un número real de 32 bits (OUT).

La operación **Dividir reales** divide dos números reales de 32 bits, dando como resultado un cociente de número real de 32 bits.

En KOP y FUP:  $IN1 * IN2 = OUT$   
 $IN1 / IN2 = OUT$

En AWL:  $IN1 * OUT = OUT$   
 $OUT / IN1 = OUT$

Condiciones de error que ponen ENO a 0: SM1.1 (desbordamiento), SM1.3 (división por cero), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero); SM1.1 (desbordamiento o valor no válido generado durante la operación o parámetro de entrada no válido); SM1.2 (negativo); SM1.3 (división por cero)

Si SM1.3 se activa durante una operación de división, permanecerán inalterados los demás bits aritméticos de estado, así como los operandos de entrada originales. SM1.1 se utiliza para indicar errores de desbordamiento y valores no válidos. Si SM1.1 se activa, el estado de SM1.0 y de SM1.2 no será válido y no se alterarán los operandos de entrada originales. Si SM1.1 y SM1.3 no se activan (durante una operación de división), la operación aritmética habrá finalizado con un resultado válido, y tanto SM1.0 como SM1.2 contendrán un estado válido.

Entradas/salidas	Operandos	Tipos de datos
IN1, IN2	VD, ID, QD, MD, SMD, SD, LD, AC, constante, *VD, *AC, *LD	REAL
OUT	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *AC, *LD	REAL

### Nota

Los números reales (o números en coma flotante) se representan en el formato descrito en la norma ANSI/IEEE 754-1985 (precisión sencilla). Para obtener más información al respecto, consulte dicha norma.

Ejemplos de operaciones aritméticas

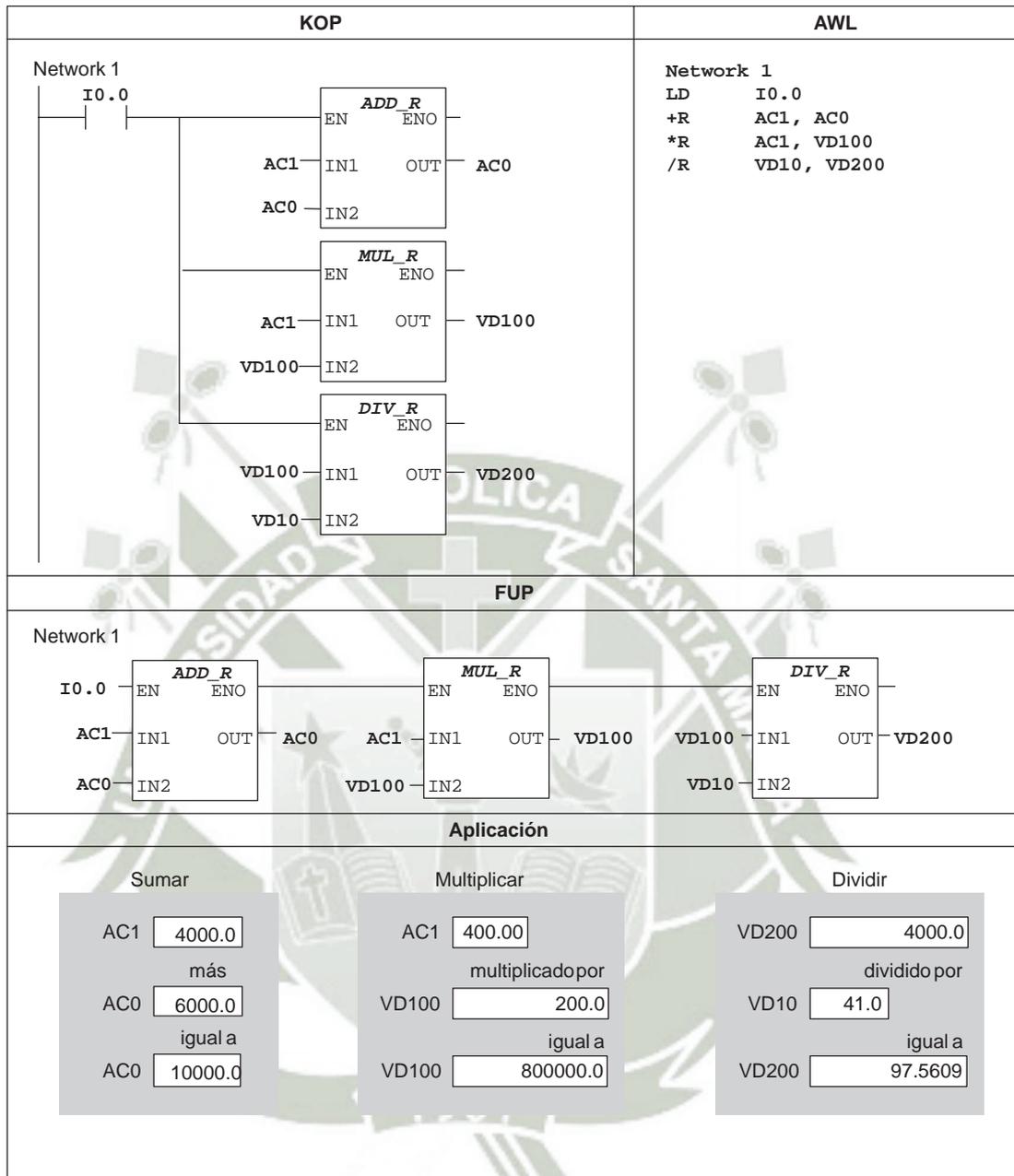
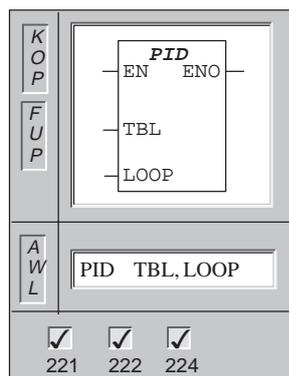


Figura 9-27 Ejemplos de operaciones aritméticas con reales en KOP, AWL y FUP

## Regulación PID



La operación **Regulación PID** ejecuta el cálculo de un lazo de regulación PID en el LOOP referenciado en base a las informaciones de entrada y configuración definidas en Table (TBL).

Condiciones de error que ponen ENO a 0: SM1.1 (desbordamiento), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Esta operación afecta a la siguiente marca especial: SM1.1 (desbordamiento)

Entradas/salidas	Operandos	Tipos de datos
TBL	VB	BYTE
LOOP	Constante (0 a 7)	BYTE

La operación PID (lazo de regulación con acción proporcional, integral, derivada) sirve para ejecutar el cálculo PID. Para habilitar el cálculo PID, el primer nivel de la pila lógica (TOS) deberá estar a ON (circulación de corriente). Esta operación tiene dos operandos: una dirección TABLE que constituye la dirección inicial de la tabla del lazo y un número LOOP que es una constante comprendida entre 0 y 7. Un programa sólo admite ocho operaciones PID. Si se utilizan dos o más operaciones PID con el mismo número de lazo (aunque tengan diferentes direcciones de tabla), los dos cálculos PID se interferirán mutuamente siendo impredecible la salida resultante.

La tabla del lazo almacena nueve parámetros que sirven para controlar y supervisar la operación del mismo. Incluye el valor actual y previo de la variable del proceso (valor real), la consigna, la salida o magnitud manipulada, la ganancia, el tiempo de muestreo, el tiempo de acción integral, el tiempo de acción derivada y la suma integral (bias).

Para poder realizar el cálculo PID con el intervalo de muestreo deseado, la operación PID deberá ejecutarse bien dentro de una rutina de interrupción temporizada o desde el programa principal, a intervalos controlados por un temporizador. El tiempo de muestreo debe definirse en calidad de entrada para la operación PID a través de la tabla del lazo.

### Utilizar el Asistente PID en STEP 7-Micro/WIN 32

STEP 7-Micro/WIN 32 incorpora el Asistente PID que ayuda a definir un algoritmo PID para un proceso de control de bucle cerrado. Seleccione el comando de menú **Herramientas > Asistente de operaciones** y elija "PID" en la ventana del Asistente.

## Algoritmo PID

En estado estacionario, un regulador PID varía el valor de su salida para llevar a cero el error de regulación ( $e$ ). El error es la diferencia entre el valor de consigna (SP) (el punto de trabajo deseado) y la variable del proceso (PV) (el punto de trabajo real). El principio de una regulación PID se basa en la ecuación que se indica a continuación y que expresa la salida  $M(t)$  como una función de un término proporcional, uno integral y uno diferencial:

<b>M(t)</b>	=	<b>K<sub>C</sub> * e</b>	+	<b>K<sub>C</sub> ∫<sub>0</sub><sup>t</sup> e dt + M<sub>inicial</sub></b>	+	<b>K<sub>C</sub> * de/dt</b>
Salida	=	término proporcional	+	término integral	+	término diferencial

donde:

$M(t)$  es la salida del lazo en función del tiempo  
 $K_C$  es la ganancia del lazo  
 $e$  es el error de regulación (diferencia entre consigna y variable de proceso)  
 $M_{inicial}$  es el valor inicial de la salida del lazo

Para poder implementar esta función de regulación en un sistema digital, la función continua deberá cuantificarse mediante muestreos periódicos del valor del error, calculándose seguidamente el valor de la salida. La ecuación que constituye la base de la solución en un sistema digital es:

<b>M<sub>n</sub></b>	=	<b>K<sub>C</sub> * e<sub>n</sub></b>	+	<b>K<sub>I</sub> * ∑<sub>1</sub><sup>n</sup> + M<sub>inicial</sub></b>	+	<b>K<sub>D</sub> * (e<sub>n</sub>-e<sub>n-1</sub>)</b>
Salida	=	término proporcional	+	término integral	+	término diferencial

donde:

$M_n$  es el valor de salida del lazo calculado en el muestreo n-ésimo  
 $K_C$  es la ganancia del lazo  
 $e_n$  es el valor del error de regulación en el muestreo n-ésimo  
 $e_{n-1}$  es el valor previo del error de regulación (en el muestreo (n-1)-ésimo)  
 $K_I$  es la constante proporcional del término integral  
 $M_{inicial}$  es el valor inicial de la salida del lazo  
 $K_D$  es la constante proporcional del término diferencial

Para esta ecuación, el término integral se muestra en función de todos los términos del error, desde el primer muestreo hasta el muestreo actual. El término diferencial es una función del muestreo actual y del muestreo previo; mientras que el término proporcional sólo es función del muestreo actual. En un sistema digital no es práctico almacenar todos los muestreos del término del error, además de no ser necesario.

Como un sistema digital debe calcular el valor de salida cada vez que se muestre el error, comenzando por el primer muestreo, sólo es necesario almacenar el valor previo del error y el valor previo del término integral. Debido a la naturaleza repetitiva de la solución basada en un sistema digital es posible simplificar la ecuación a resolver en cada muestreo. La ecuación simplificada es:

<b><math>M_n</math></b>	<b>=</b>	<b><math>K_C * e_n</math></b>	<b>+</b>	<b><math>K_I * e_n + MX</math></b>	<b>+</b>	<b><math>K_D * (e_n - e_{n-1})</math></b>
Salida	=	término proporcional	+	término integral	+	término diferencial

donde:

- $M_n$  es el valor de salida del lazo calculado en el muestreo n-ésimo
- $K_C$  es la ganancia del lazo
- $e_n$  es el valor del error de regulación en el muestreo n-ésimo
- $e_{n-1}$  es el valor previo del error de regulación (en el muestreo (n-1)-ésimo)
- $K_I$  es la constante proporcional del término integral
- $MX$  es el valor previo del término integral (en el muestreo (n-1)-ésimo)
- $K_D$  es la constante proporcional del término diferencial

Para calcular el valor de salida del lazo, la CPU utiliza una forma modificada de la ecuación simplificada anterior. Esta ecuación modificada equivale a la siguiente:

<b><math>M_n</math></b>	<b>=</b>	<b><math>MP_n</math></b>	<b>+</b>	<b><math>MI_n</math></b>	<b>+</b>	<b><math>MD_n</math></b>
Salida	=	término proporcional	+	término integral	+	término diferencial

donde:

- $M_n$  es el valor de salida del lazo calculado en el muestreo n-ésimo
- $MP_n$  es el valor del término proporcional de salida del lazo en el muestreo n-ésimo
- $MI_n$  es el valor del término integral de salida del lazo en el muestreo n-ésimo
- $MD_n$  es el valor del término diferencial de salida del lazo en el muestreo n-ésimo

### Término proporcional

El término proporcional MP es el producto de la ganancia ( $K_C$ ), la cual controla la sensibilidad del cálculo de la salida, y del error (e), que es la diferencia entre el valor de consigna (SP) y el valor real o de la variable del proceso (PV) para un instante de muestreo determinado. La ecuación que representa el término proporcional según la resuelve la CPU es la siguiente:

$$MP_n = K_C * (SP_n - PV_n)$$

donde:

$MP_n$	es el valor del término proporcional de salida del lazo en el muestreo n-ésimo
$K_C$	es la ganancia del lazo
$SP_n$	es el valor de la consigna en el muestreo n-ésimo
$PV_n$	es el valor de la variable del proceso en el muestreo n-ésimo

### Término integral

El término integral MI es proporcional a la suma del error a lo largo del tiempo. La ecuación que representa el término integral tal y como la resuelve la CPU es:

$$MI_n = K_C * T_S / T_I * (SP_n - PV_n) + MX$$

donde:

$MI_n$	es el valor del término integral de salida del lazo en el muestreo n-ésimo
$K_C$	es la ganancia del lazo
$T_S$	es el tiempo de muestreo del lazo
$T_I$	es el período de integración del lazo (también llamado tiempo de acción integral)
$SP_n$	es el valor de la consigna en el muestreo n-ésimo
$PV_n$	es el valor de la variable del proceso en el muestreo n-ésimo
$MX$	es el valor previo del término integral (en el muestreo (n-1)-ésimo) (también llamado suma integral o "bias")

La suma integral o bias (MX) es la suma acumulada de todos los valores previos del término integral. Después de cada cálculo de  $MI_n$  se actualiza la suma integral con el valor de  $MI_n$  que puede ajustarse o limitarse (para más detalles, v. la sección "Variables y márgenes"). Por regla general, el valor inicial de la suma integral se ajusta al valor de salida ( $M_{inicial}$ ) justo antes de calcular la primera salida del lazo. El término integral incluye también varias constantes tales como la ganancia ( $K_C$ ), el tiempo de muestreo ( $T_S$ ), que define el intervalo con que se recalcula periódicamente el valor de salida del lazo PID, y el tiempo de acción integral ( $T_I$ ), que es un tiempo que se utiliza para controlar la influencia del término integral en el cálculo de la salida.

### Término diferencial

El término diferencial MD es proporcional a la tasa de cambio del error. La ecuación del término diferencial equivale a la siguiente:

$$MD_n = K_C * T_D / T_S * ((SP_n - PV_n) - (SP_{n-1} - PV_{n-1}))$$

Para evitar cambios o saltos bruscos de la salida debidos a cambios de la acción derivada o de la consigna se ha modificado esta ecuación bajo la hipótesis de que la consigna es constante ( $SP_n = SP_{n-1}$ ). En consecuencia, se calcula el cambio en la variable del proceso en lugar del cambio en el error, como puede verse a continuación:

$$MD_n = K_C * T_D / T_S * (SP_n - PV_n - SP_n + PV_{n-1})$$

o simplificando:

$$MD_n = K_C * T_D / T_S * (PV_{n-1} - PV_n)$$

donde:

$MD_n$	es el valor del término diferencial de la salida del lazo en el muestreo n-ésimo
$K_C$	es la ganancia del lazo
$T_S$	es el tiempo de muestreo del lazo
$T_D$	es el período de diferenciación de lazo (también llamado tiempo de acción derivada)
$SP_n$	es el valor de la consigna en el muestreo n-ésimo
$SP_{n-1}$	es el valor de la consigna en el muestreo (n-1)-ésimo
$PV_n$	es el valor de la variable del proceso en el muestreo n-ésimo
$PV_{n-1}$	es el valor de la variable del proceso en el muestreo (n-1)-ésimo

En lugar del error es necesario guardar la variable del proceso para usarla en el próximo cálculo del término diferencial. En el instante del primer muestreo, el valor de  $PV_{n-1}$  se inicializa a un valor igual a  $PV_n$ .

### Elegir el tipo de regulación

En muchos sistemas de regulación basta emplear una o dos acciones de regulación. Así, por ejemplo, puede requerirse únicamente regulación proporcional o regulación proporcional e integral. El tipo de regulación se selecciona ajustando correspondientemente los valores de los parámetros constantes.

Así, si no se desea acción integral (sin "I" en el cálculo PID), entonces el tiempo de acción integral deberá ajustarse a infinito. Incluso sin acción integral el valor del término integral puede no ser cero debido a que la suma integral MX puede tener un valor inicial.

Si no se desea acción derivada (sin "D" en el cálculo PID), entonces el tiempo de acción derivada deberá ajustarse a 0.0.

Si no se desea acción proporcional (sin "P" en el cálculo PID) y se desea regulación I o ID, entonces la ganancia deberá ajustarse a 0.0. Como la ganancia interviene en las ecuaciones para calcular los términos integral y diferencial, si se ajusta a 0.0 resulta un valor de 1.0, que es el utilizado para calcular los términos integral y diferencial.

## Convertir y normalizar las entradas del lazo

El lazo tiene dos variables o magnitudes de entrada: la consigna y la variable del proceso. La consigna es generalmente un valor fijo como el ajuste de velocidad en el computador de abordaje de su automóvil. La variable del proceso es una magnitud relacionada con la salida del lazo y que mide por ello el efecto que tiene la misma sobre el sistema regulado. En el ejemplo del computador de abordaje, la variable del proceso sería la entrada al tacómetro que es una señal proporcional a la velocidad de giro de las ruedas.

Tanto la consigna como la variable del proceso son valores físicos que pueden tener diferente magnitud, margen y unidades de ingeniería. Para que la operación PID pueda utilizar esos valores físicos, éstos deberán convertirse a representaciones normalizadas en coma flotante.

El primer paso es convertir el valor físico de un valor entero de 16 bits a un valor en coma flotante o real. La siguiente secuencia de instrucciones muestra la forma de convertir un valor entero en un número real.

```
XORD AC0, AC0 // Borrar el acumulador.
MOVW AIW0, AC0 // Guardar en el acumulador el valor analógico.
LDW>= AC0, 0 // Si el valor analógico es positivo,
JMP 0 // entonces convertir a número real.
NOT // Si no,
ORD 16#FFFF0000, AC0 // el signo amplía el valor en AC0.
LBL 0
DTR AC0, AC0 // Convertir entero de 32 bits a un número real.
```

El próximo paso consiste en convertir el número real representativo del valor físico en un valor normalizado entre 0.0 y 1.0. La ecuación siguiente se utiliza para normalizar tanto la consigna como el valor de la variable del proceso.

$$R_{\text{Norm}} = (R_{\text{No norm}} / \text{Alcance}) + \text{Offset}$$

donde:

$R_{\text{Norm}}$  es la representación como número real normalizado del valor físico  
 $R_{\text{No norm}}$  es la representación como número real no normalizado del valor físico  
 Offset vale 0.0 para valores unipolares  
 vale 0.5 para valores bipolares  
 Alcance es la diferencia entre el máximo valor posible menos el mínimo valor posible  
 = 32.000 para valores unipolares (típico)  
 = 64.000 para valores bipolares (típico)

La siguiente secuencia de instrucciones muestra la forma de normalizar el valor bipolar contenido en AC0 (cuyo alcance vale 64.000), continuando la secuencia previa:

```
/R 64000.0, AC0 // Normaliza el valor en el acumulador
+R 0.5, AC0 // Desplaza el valor al margen entre 0.0 y 1.0
MOVR AC0, VD100 // Almacena el valor normalizado en la tabla del lazo
```

### Convertir la salida del lazo en un valor entero escalado

La salida del lazo constituye la variable manipulada; en el caso del automóvil, la posición de la mariposa en el carburador. La salida del lazo es un valor real normalizado comprendido entre 0.0 y 1.0. Antes de que la salida del lazo pueda utilizarse para excitar una salida analógica, deberá convertirse a un valor escalado de 16 bits. Esta operación constituye el proceso inverso de convertir PV y SP en un valor normalizado. El primer paso es convertir la salida del lazo en un valor real escalado usando la fórmula siguiente:

$$R_{Scal} = (M_n - Offset) * Alcance$$

donde:

- $R_{Scal}$  es el valor real escalado de la salida del lazo
- $M_n$  es el valor real normalizado de la salida del lazo
- Offset vale 0.0 para valores unipolares  
 vale 0.5 para valores bipolares
- Alcance es la diferencia entre el máximo valor posible menos el mínimo valor posible  
 = 32.000 para valores unipolares (típico)  
 = 64.000 para valores bipolares (típico)

La siguiente secuencia de instrucciones muestra la forma de escalar la salida del lazo:

```
MOVR VD108, AC0 // Mover la salida del lazo al acumulador.
-R 0.5, AC0 // Incluir esta operación sólo si el valor es
// bipolar.
*R 64000.0, AC0 // Escalar el valor en el acumulador.
```

Seguidamente es necesario convertir en un entero de 16 bits el valor real escalado representativo de la salida del lazo. La siguiente secuencia muestra la forma de realizar esta conversión:

```
ROUND AC0 AC0 // Convertir entero de 32 bits a un número real.
MOVW AC0, AQW0 // Escribir el entero de 16 bits en la salida
// analógica.
```

### Lazos con acción positiva o negativa

El lazo tiene acción positiva si la ganancia es positiva y acción negativa si la ganancia es negativa. (En regulación I o ID, donde la ganancia vale 0.0, si se especifica un valor positivo para el tiempo de acción integral y derivada resulta un lazo de acción positiva y de acción negativa al especificarse valores negativos).

## Variables y márgenes

La variable del proceso y la consigna son magnitudes de entrada para el cálculo PID. Por ello, la operación PID lee los campos definidos para estas variables en la tabla del lazo, pero no los modifica.

El valor de salida se genera al realizar el cálculo PID; como consecuencia, el campo en la tabla del lazo que contiene el valor de salida se actualiza cada vez que se termina un cálculo PID. El valor de salida está limitado entre 0.0 y 1.0. El usuario puede utilizar el campo de valor de salida en calidad de campo de entrada para especificar un valor de salida inicial cuando se conmute de control manual a automático (consulte también la sección "Modos").

Si se utiliza regulación integral, la suma integral es actualizada por el cálculo PID y el valor actualizado se utiliza como entrada para el siguiente cálculo PID. Si el valor de salida calculado se sale de margen (salida inferior a 0.0 o superior a 1.0), la suma integral se ajusta de acuerdo con las fórmulas siguientes:

$$MX = 1.0 - (MP_n + MD_n) \quad \text{si la salida calculada, } M_n > 1.0$$

o

$$MX = - (MP_n + MD_n) \quad \text{si la salida calculada, } M_n < 0.0$$

donde:

MX	es el valor de la suma integral ajustada
MP <sub>n</sub>	es el valor del término proporcional de salida del lazo en el muestreo n-ésimo
MD <sub>n</sub>	es el valor del término diferencial de la salida del lazo en el muestreo n-ésimo
M <sub>n</sub>	es el valor de la salida del lazo en el muestreo n-ésimo

Si la suma integral se calcula de la forma descrita, se mejora la respuesta del sistema cuando la salida calculada retorna al margen adecuado. Es decir, la suma integral calculada se limita entre 0.0 y 1.0 y luego se escribe en el campo reservado para ella en la tabla del lazo cada vez que se finaliza un cálculo PID. El valor almacenado en la tabla del lazo se utiliza para el próximo cálculo PID.

A fin de evitar problemas con valores de la suma integral en determinadas situaciones de aplicación, el usuario puede modificar, antes de ejecutar la operación PID, el valor de la suma integral en la tabla del lazo. Cualquier modificación manual de la suma integral deberá realizarse con mucho cuidado. En cualquier caso, el valor de la suma integral escrito en la tabla del lazo deberá ser un número real comprendido entre 0.0 y 1.0.

En la tabla del lazo se mantiene un valor de comparación de la variable del proceso para su uso en la parte de acción derivada del cálculo PID. El usuario no deberá modificar dicho valor.

## Modos

Los lazos PID del S7-200 no incorporan control de modo de operación. El cálculo PID sólo se ejecuta si circula corriente hacia el cuadro PID. Por ello resulta el modo "automático" o "auto" cuando se ejecuta cíclicamente el cálculo PID. Resulta el modo "manual" cuando no se ejecuta el cálculo PID.

La operación PID tiene un bit de historial de circulación de corriente similar a una operación de contador. La operación utiliza dicho bit de historial para detectar una transición de la circulación de corriente de 0 a 1. Cuando se detecta dicha transición, la operación ejecuta una serie de acciones destinadas a lograr un cambio sin choques de modo manual a automático. Para evitar choques en la transición al modo automático, el valor de la salida ajustado por control manual deberá entregarse en calidad de entrada a la operación PID (escrita en la entrada para  $M_n$  en la tabla del lazo) antes de conmutar a modo automático. La operación PID ejecuta las siguientes acciones con los valores de la tabla del lazo a fin de asegurar un cambio sin choques entre control manual y automático cuando se detecta una transición de la circulación de corriente de 0 a 1:

- Ajusta consigna ( $SP_n$ ) = variable de proceso ( $PV_n$ )
- Ajusta variable del proceso antigua ( $PV_{n-1}$ ) = variable del proceso ( $PV_n$ )
- Ajusta suma integral (MX) = valor de salida ( $M_n$ )

El estado por defecto de los bits de historial PID es "activado"; dicho estado se establece en el arranque de la CPU o cada vez que hay una transición de modo STOP a RUN en el sistema de automatización. Si circula corriente hacia el cuadro PID la primera vez que se ejecuta tras entrar en el modo RUN, entonces no se detecta ninguna transición de circulación de corriente y, por consecuencia, no se ejecutan las acciones destinadas a evitar choques en el cambio de modo.

## Alarmas y operaciones especiales

La operación PID es simple, pero ofrece grandes prestaciones para ejecutar cálculos PID. Si se precisan funciones de postprocesamiento tales como funciones de alarma o cálculos especiales en base a las variables de lazo, ésto deberá implementarse utilizando las instrucciones básicas admitidas por la CPU en cuestión.

### Condiciones de error

A la hora de compilar, la CPU generará un error de compilación (error de margen) y la compilación fallará si los operandos correspondientes a la dirección inicial o al número de lazo PID en la tabla del lazo están fuera de margen.

La operación PID no comprueba si todos los valores de entrada en la tabla del lazo respetan los límites de margen. Es decir, el usuario debe asegurarse de que la variable del proceso y la consigna (al igual que la suma integral y la variable del proceso previa, si se utilizan como entradas) sean números reales comprendidos entre 0.0 y 1.0.

Si se detecta algún error al ejecutar las operaciones aritméticas del cálculo PID se activa la marca SM1.1 (desbordamiento o valor no válido) y se finaliza la ejecución de la operación PID. (La actualización de los valores de salida en la tabla del lazo puede ser incompleta por lo que deberán descartarse dichos valores y corregir el valor de entrada que ha causado el error matemático antes de volver a efectuar la operación de regulación PID).

### Tabla del lazo

La tabla de lazo tiene 36 bytes de longitud y el formato que muestra la tabla 9-19:

Tabla 9-19 Formato de la tabla del lazo

Offset	Campo	Formato	Tipo	Descripción
0	Variable del proceso ( $PV_n$ )	Palabra doble – real	IN	Contiene la variable del proceso que debe estar escalada entre 0.0 y 1.0.
4	Consigna ( $SP_n$ )	Palabra doble – real	IN	Contiene la consigna que debe estar escalada entre 0.0 y 1.0.
8	Salida ( $M_n$ )	Palabra doble – real	IN/OUT	Contiene la salida calculada, escalada entre 0.0 y 1.0.
12	Ganancia ( $K_C$ )	Palabra doble – real	IN	Contiene la ganancia, que es una constante proporcional. Puede ser un número positivo o negativo.
16	Tiempo de muestreo ( $T_S$ )	Palabra doble – real	IN	Contiene, en segundos, el tiempo de muestreo. Tiene que ser un número positivo.
20	Tiempo de acción integral ( $T_I$ )	Palabra doble – real	IN	Contiene, en minutos, el tiempo de acción integral. Tiene que ser un número positivo.
24	Tiempo de acción derivada ( $T_D$ )	Palabra doble – real	IN	Contiene, en minutos, el tiempo de acción derivada. Tiene que ser un número positivo.
28	Suma integral (MX)	Palabra doble – real	IN/OUT	Contiene el valor de la suma integral entre 0.0 y 1.0.
32	Variable del proceso previa ( $PV_{n-1}$ )	Palabra doble – real	IN/OUT	Contiene el valor previo de la variable del proceso almacenada desde la última ejecución de la operación PID.

## Ejemplo de programa PID

En este ejemplo se utiliza un depósito para mantener una presión de agua constante. Para ello se toma continuamente agua del depósito en una cantidad variable. Una bomba de velocidad variable se utiliza para añadir agua al depósito con un caudal apto para mantener una presión adecuada del agua y evitar así que se vacíe.

La consigna de este sistema es el nivel de agua; en este caso, un valor equivalente al 75% de llenado del depósito. La variable del proceso la suministra un sensor flotador que señala el nivel de llenado del depósito; equivale a 0 % cuando está vacío y a 100 % cuando está completamente lleno. La salida es una señal que permite controlar la velocidad de la bomba, del 0 al 100 % de su velocidad máxima.

La consigna está predeterminada y se introduce directamente en la tabla del lazo. El sensor flotador suministra la variable del proceso que es un valor analógico unipolar. La salida del lazo se escribe en una salida analógica unipolar que se utiliza para controlar la velocidad de la bomba. El alcance tanto de la entrada como de la salida analógica es de 32.000.

En este ejemplo sólo se utiliza acción proporcional e integral. La ganancia del lazo y las constantes de tiempo se han determinado durante cálculos de ingeniería y se ajustan para obtener una regulación óptima. Los valores calculados de las constantes de tiempo se indican a continuación:

$K_C$  es 0.,5

$T_S$  es 0,1 segundos

$T_I$  es 30 minutos

La velocidad de la bomba se controlará de forma manual hasta que el depósito esté lleno al 75 %, seguidamente se abre la válvula para sacar agua del mismo. Simultáneamente se conmuta la bomba de modo manual a automático. La entrada digital se utiliza para conmutar de manual a automático. Esta entrada se describe seguidamente:

I0.0 es control manual/automático; 0 = manual, 1 = automático

En modo manual, el operador ajusta la velocidad de la bomba en VD108 mediante un valor real de 0.0 a 1.0.

La figura 9-28 muestra el programa de control (regulación) para esta aplicación.

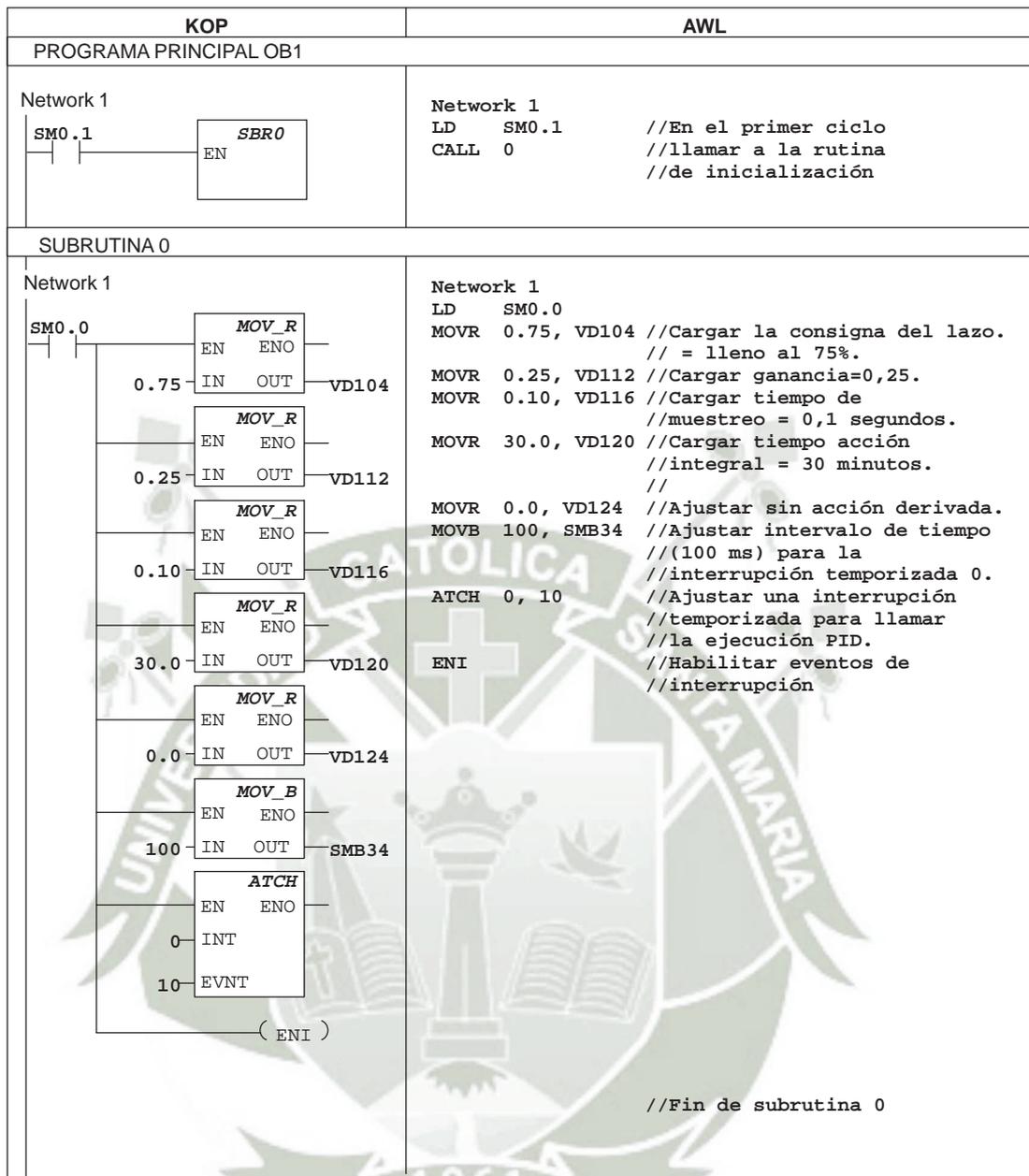


Figura 9-28 Ejemplo de regulación PID

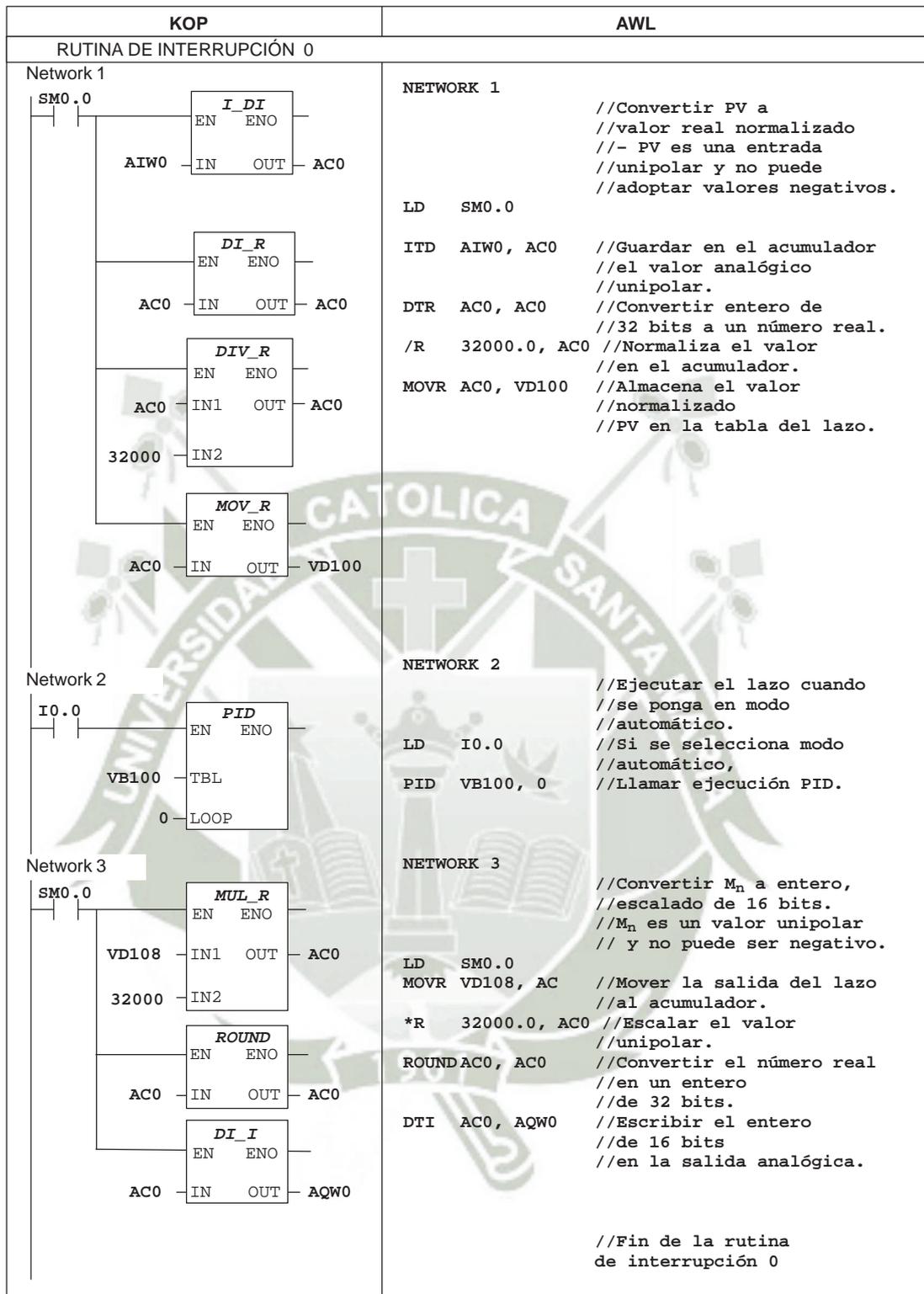


Figura 9-28 Ejemplo de regulación PID (continuación)

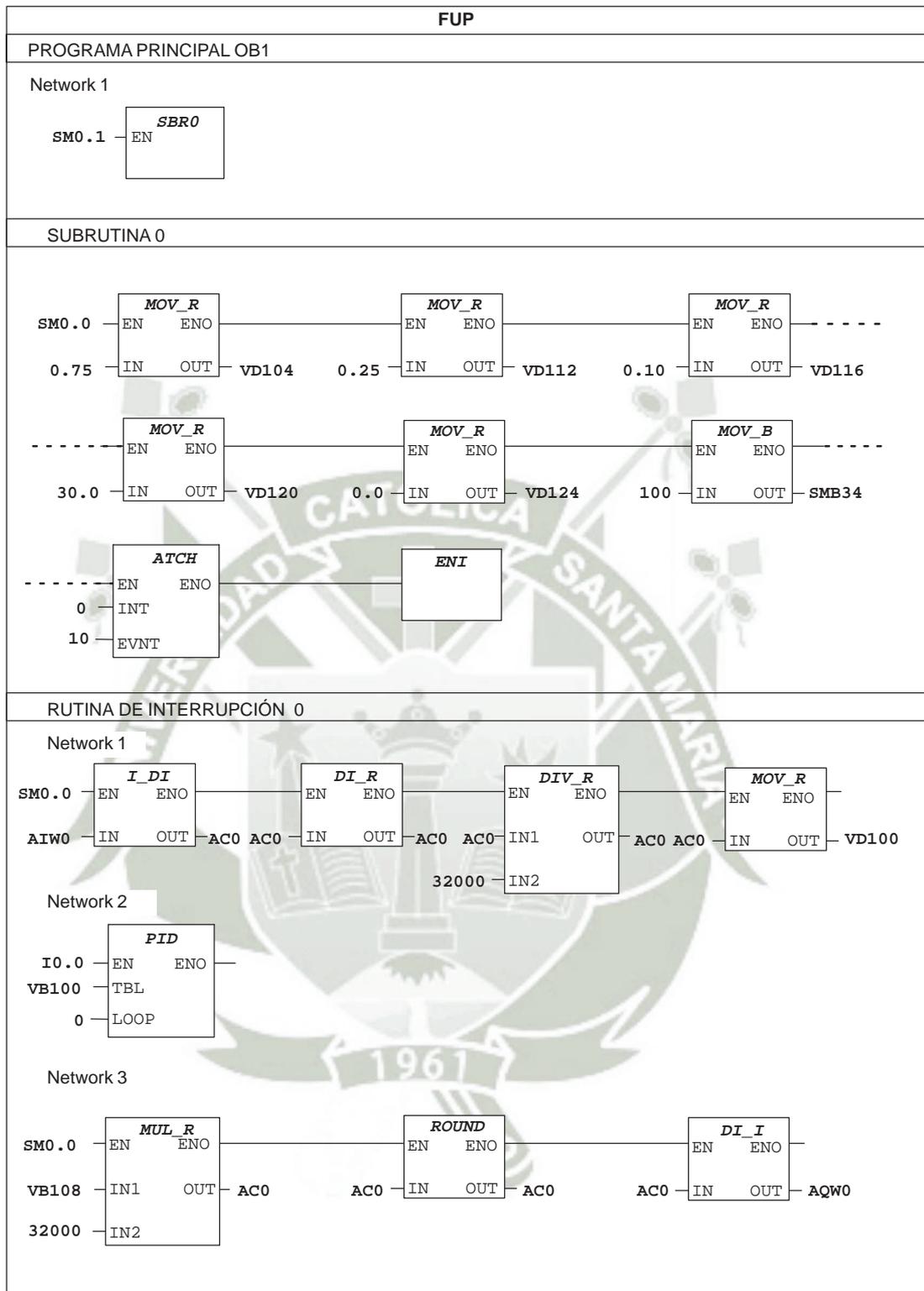
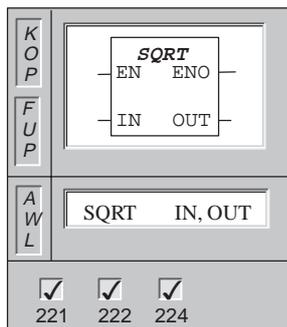


Figura 9-28 Ejemplo de regulación PID (continuación)

### Raíz cuadrada



La operación **Raíz cuadrada** extrae la raíz cuadrada de un número real de 32 bits (IN), dando como resultado un número real de 32 bits (OUT), como muestra la ecuación:

$$\sqrt{\text{IN}} = \text{OUT}$$

Condiciones de error que ponen ENO a 0: SM1.1 (desbordamiento), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

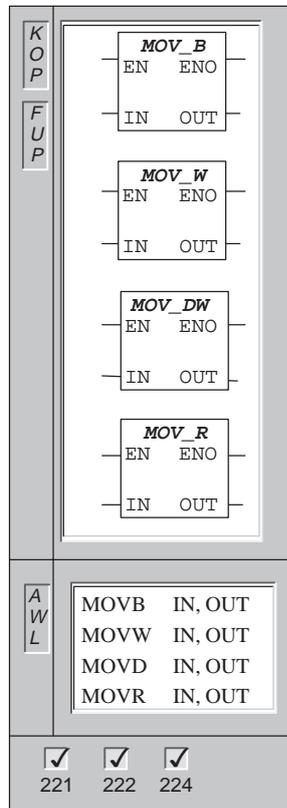
Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero); SM1.1 (desbordamiento); SM1.2 (negativo)

SM1.1 se utiliza para indicar errores de desbordamiento y valores no válidos. Si SM1.1 se activa, el estado de SM1.0 y de SM1.2 no será válido y no se alterarán los operandos de entrada originales. Si SM1.1 no se activa, la operación aritmética se habrá finalizado con un resultado válido, y tanto SM1.0 como SM1.2 contendrán un estado válido.

Entradas/salidas	Operandos	Tipos de datos
IN	VD, ID, QD, MD, SMD, SD, LD, AC, constante, *VD, *AC, *LD	REAL
OUT	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *AC, *LD	REAL

## 9.10 Operaciones de transferencia (SIMATIC)

### Transferir byte, Transferir palabra, Transferir palabra doble y Transferir real



La operación **Transferir byte** transfiere el byte de entrada (IN) al byte de salida (OUT). El byte de entrada permanece inalterado.

La operación **Transferir palabra** transfiere la palabra de entrada (IN) a la palabra de salida (OUT). La palabra de entrada permanece inalterada.

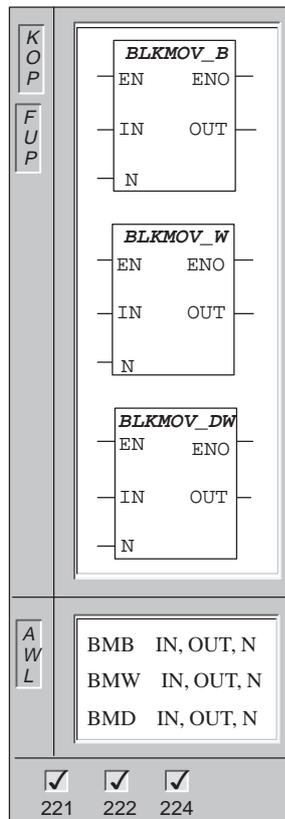
La operación **Transferir palabra doble** transfiere la palabra doble de entrada (IN) a la palabra doble de salida (OUT). La palabra doble de entrada permanece inalterada.

La operación **Transferir real** transfiere un número real de 32 bits de la palabra doble de entrada (IN) a la palabra doble de salida (OUT). La palabra doble de entrada permanece inalterada.

Condiciones de error que ponen ENO a 0: SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Transferir...	Entradas/salidas	Operandos	Tipos de datos
BYTE	IN	VB, IB, QB, MB, SB, SMB, LB, AC, constante, *VD, *AC, *LD	BYTE
	OUT	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD	BYTE
WORD	IN	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, constante, AC *VD, *AC, *LD	WORD, INT
	OUT	VW, T, C, IW, QW, SW, MW, SMW, LW, AC, AQW, *VD, *AC, *LD	WORD, INT
Palabra doble	IN	VD, ID, QD, MD, SD, SMD, LD, HC, &VB, &IB, &QB, &MB, &SB, &T, &C, AC, constante, *VD, *AC, *LD	DWORD, DINT
	OUT	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *AC, *LD	DWORD, DINT
Real	IN	VD, ID, QD, MD, SD, SMD, LD, AC, constante, *VD, *AC, *LD	REAL
	OUT	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *AC, *LD	REAL

### Transferir bytes en bloque, Transferir palabras en bloque, Transferir palabras dobles en bloque



La operación **Transferir bytes en bloque** transfiere un número determinado de bytes (N) de la dirección de entrada IN a la dirección de salida OUT. N puede estar comprendido entre 1 y 255.

La operación **Transferir palabras en bloque** transfiere un número determinado de palabras (N) de la dirección de entrada IN a la dirección de salida OUT. N puede estar comprendido entre 1 y 255.

La operación **Transferir palabras dobles en bloque** transfiere un número determinado de palabras dobles (N) de la dirección de entrada IN a la dirección de salida OUT. N puede estar comprendido entre 1 y 255.

Condiciones de error que ponen ENO a 0: SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto), 0091 (operando fuera de área)

Transferir ... en bloque	Entradas/salidas	Operandos	Tipos de datos
BYTE	IN, OUT	VB, IB, QB, MB, SB, SMB, LB, *VD, *AC, *LD	BYTE
	N	VB, IB, QB, MB, SB, SMB, LB, AC, constante, *VD, *AC, *LD	BYTE
WORD	IN	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, *VD, *AC, *LD	WORD
	N	VB, IB, QB, MB, SB, SMB, LB, AC, constante, *VD, *AC, *LD	BYTE
	OUT	VW, IW, QW, MW, SW, SMW, LW, T, C, AQW, *VD, *LD, *AC	WORD
Palabra doble	IN, OUT	VD, ID, QD, MD, SD, SMD, LD, *VD, *AC, *LD	DWORD
	N	VB, IB, QB, MB, SB, SMB, LB, AC, constante, *VD, *AC, *LD	BYTE

Ejemplo de una operación de transferencia de bloques

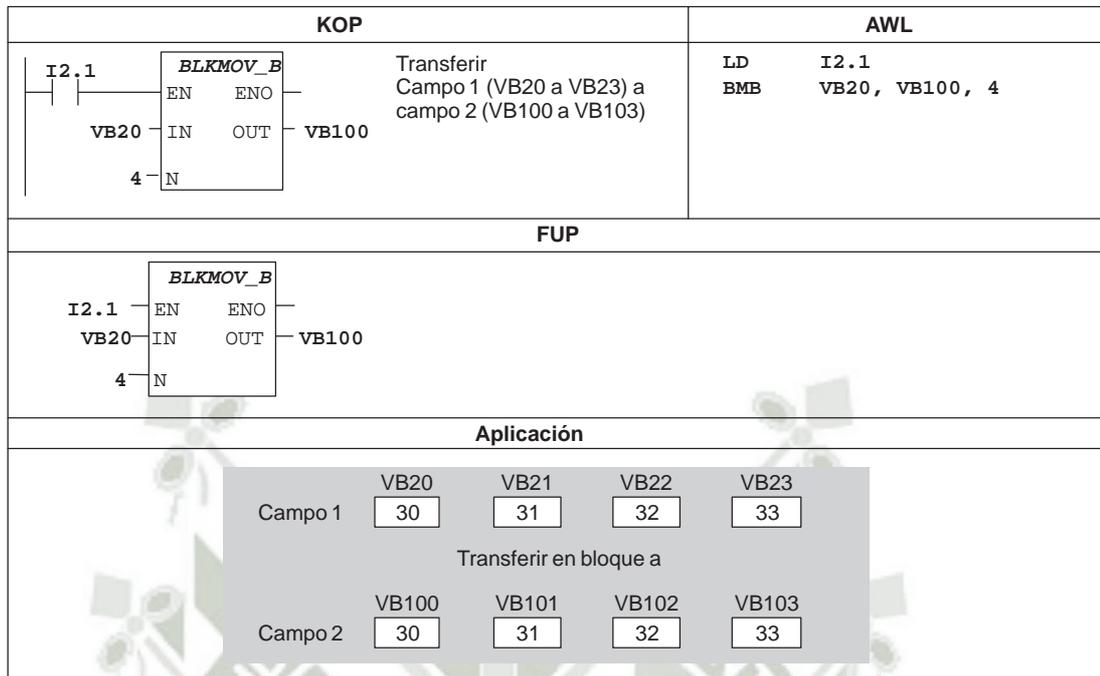
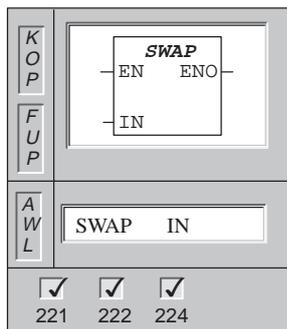


Figura 9-29 Ejemplo de operaciones de transferencia en bloque en KOP, AWL y FUP

### Invertir bytes de una palabra



La operación **Invertir bytes de una palabra** intercambia el byte más significativo y el byte menos significativo de una palabra (IN).

Condiciones de error que ponen ENO a 0: SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Entradas/salidas	Operandos	Tipos de datos
IN	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, *VD, *AC, *LD	WORD

### Ejemplos de operaciones de transferir e invertir

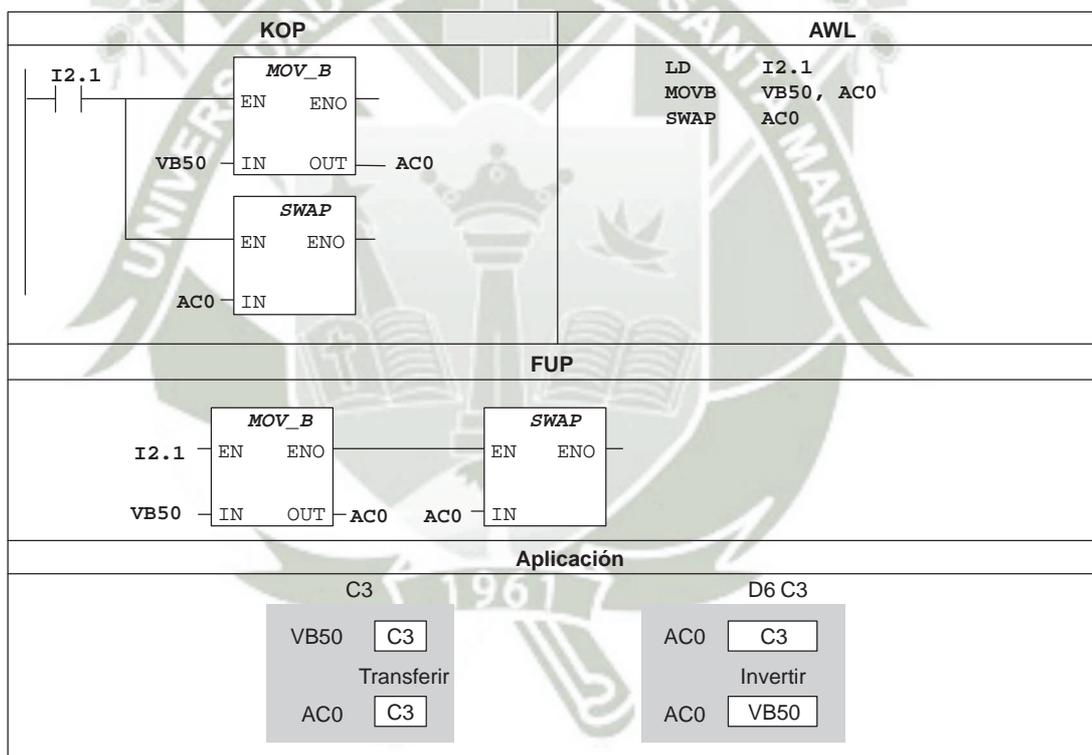
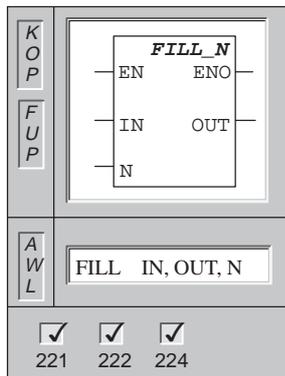


Figura 9-30 Ejemplo de las operaciones de transferencia y de inicializar memoria en KOP, AWL y FUP

### Inicializar memoria



La operación **Inicializar memoria** rellena la memoria que comienza en la palabra de salida (OUT) con la configuración de la palabra de entrada (IN) para el número de palabras indicado por N. N tiene un margen comprendido entre 1 y 255.

Condiciones de error que ponen ENO a 0: SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto), 0091 (operando fuera de área)

Entradas/salidas	Operandos	Tipos de datos
IN	VW, IW, QW, MW, SW, SMW, LW, AIW, T, C, AC, constante, *VD, *AC, *LD	WORD
N	VB, IB, QB, MB, SB, SMB, LB, AC, constante, *VD, *AC, *LD	BYTE
OUT	VW, IW, QW, MW, SW, SMW, LW, T, C, AQW, *VD, *AC, *LD	WORD

### Ejemplo de la operación Inicializar memoria

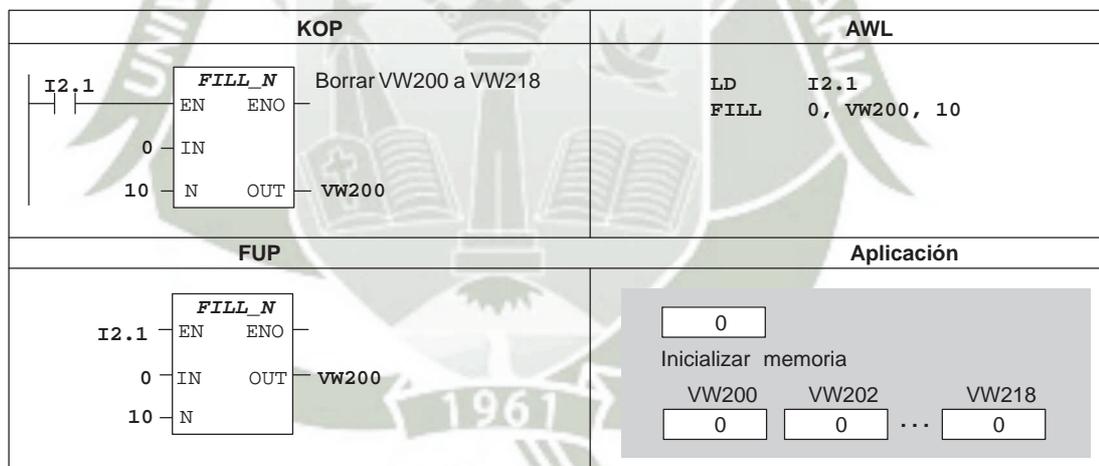
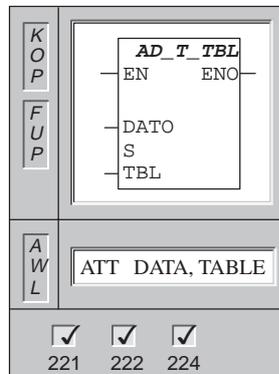


Figura 9-31 Ejemplos de la operación Inicializar memoria en KOP, AWL y FUP

## 9.11 Operaciones de tabla (SIMATIC)

### Registrar valor en tabla



La operación **Registrar valor en tabla** registra valores de palabra (DATA) en la tabla (TBL).

El primer valor de la tabla indica la longitud máxima de la misma (TL). El segundo valor (EC) indica el número de registros que contiene la tabla (v. fig. 9-32). Los nuevos datos se añaden al final de la tabla, debajo del último registro. Cada vez que se añade un registro a la tabla, se incrementa el número efectivo de registros. Una tabla puede tener como máximo 100 registros.

Condiciones de error que ponen ENO a 0: SM1.4 (desbordamiento de tabla), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto), 0091 (operando fuera de área)

Estas operaciones afectan a las siguientes marcas especiales: SM1.4 se activa si se intenta introducir demasiados registros en la tabla.

Entradas/salidas	Operandos	Tipos de datos
DATA	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, AC, constante, *VD, *AC, *LD	WORD
TBL	VW, IW, QW, MW, SW, SMW, LW, T, C, *VD, *AC, *LD	WORD

Ejemplo de la operación Registrar valor en tabla

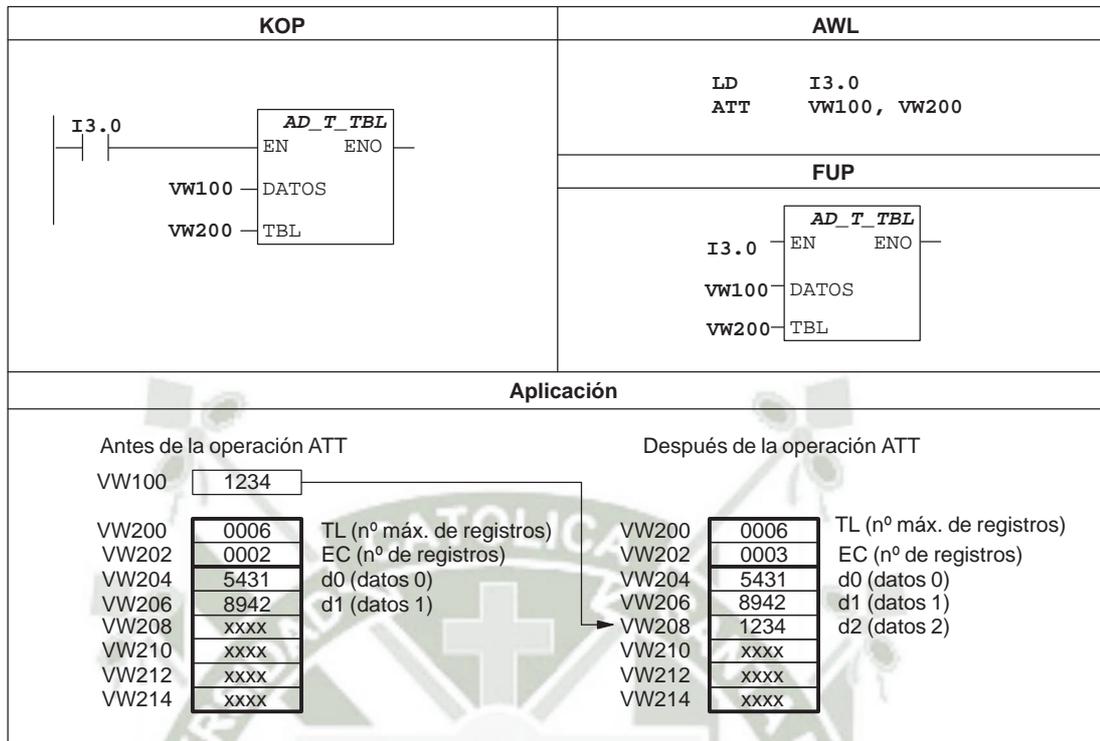


Figura 9-32 Ejemplo de la operación Registrar valor en tabla (ATT)

### Buscar valor en tabla

K O P	<b>TBL_FIND</b>	
	— EN	ENO
	— SRC	
	— PTN	
	— INDX	
	— CMD	
A W L	FND=	SRC, PATRN INDX
	FND<>	SRC, PATRN, INDX
	FND<	SRC, PATRN, INDX
	FND>	SRC, PATRN, INDX
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	221	222 224

La operación **Buscar valor en tabla** rastrea la tabla (SRC), comenzando con el registro indicado por INDX, y busca el valor (PTN) que corresponda a los criterios de búsqueda definidos por CMD. El parámetro de comando (CMD) indica un valor numérico comprendido entre 1 y 4 que corresponde a la relación =, <>, <, y >, respectivamente.

Si se cumple un criterio, INDX señalará el registro en cuestión. Para buscar el siguiente registro se habrá de incrementar INDX antes de volver a llamar nuevamente a la operación Buscar valor en tabla. Si no se encuentra ningún registro que corresponda al criterio, el valor INDX será igual al número de registros que contiene la tabla.

Una tabla puede tener como máximo 100 registros. Los registros de la tabla (el área donde se desea buscar) están numerados de 0 hasta el valor máximo (99).

Condiciones de error que ponen ENO a 0: SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto), 0091 (operando fuera de área)

Entradas/salidas	Operandos	Tipos de datos
SRC	VW, IW, QW, MW, SMW, LW, T, C, *VD, *AC, *LD	WORD
PTN	VW, IW, QW, MW, SW, SMW, AIW, LW, T, C, AC, constante, *VD, *AC, *LD	INT
INDX	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, *VD, *AC, *LD	WORD
CMD	constante	BYTE

#### Nota

Si las operaciones de búsqueda se utilizan en tablas creadas con las operaciones ATT, LIFO y FIFO, el valor de contaje corresponderá al número de registros. Contrariamente a las operaciones ATT, LIFO y FIFO, donde una palabra indica el número máximo de registros, las operaciones de búsqueda no requieren dicha palabra. Por consiguiente, la dirección del operando SRC de una operación de búsqueda supera en una palabra (dos bytes) al operando TBL correspondiente a la operación ATT, LIFO o FIFO, como muestra la figura 9-33.

Formato de tabla para ATT, LIFO y FIFO			Formato de tabla para TBL_FIND		
VW200	0006	TL (nº máx. de registros)	VW202	0006	EC (nº de registros)
VW202	0006	EC (nº de registros)	VW204	xxxx	d0 (datos 0)
VW204	xxxx	d0 (datos 0)	VW206	xxxx	d1 (datos 1)
VW206	xxxx	d1 (datos 1)	VW208	xxxx	d2 (datos 2)
VW208	xxxx	d2 (datos 2)	VW210	xxxx	d3 (datos 3)
VW210	xxxx	d3 (datos 3)	VW212	xxxx	d4 (datos 4)
VW212	xxxx	d4 (datos 4)	VW214	xxxx	d5 (datos 5)
VW214	xxxx	d5 (datos 5)			

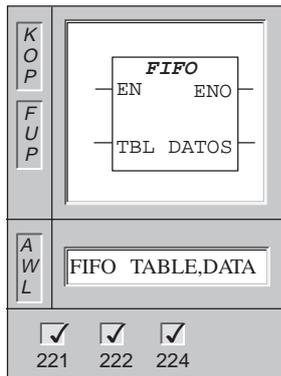
Figura 9-33 Diferencia de los formatos de tabla entre las operaciones de búsqueda y las operaciones ATT, LIFO, FIFO

Ejemplo de la operación Buscar valor en tabla

KOP	AWL																					
<p style="margin-left: 40px;">Si I2.1 está activada, buscar un valor en la tabla que sea igual a 3130 HEX.</p>	<p>LD I2.1 FND= VW202, 16#3130, AC1</p> <p style="text-align: center;"><b>FUP</b></p>																					
Aplicación																						
<p>Esta es la tabla que se va a rastrear. Si la tabla se creó utilizando las operaciones ATT, LIFO y FIFO, VW200 contendrá el número máximo de registros posibles y no será requerido por las operaciones de búsqueda.</p> <table style="margin-left: auto; margin-right: auto;"> <tr><td>VW202</td><td style="border: 1px solid black;">0006</td><td>EC (nº de registros)</td></tr> <tr><td>VW204</td><td style="border: 1px solid black;">3133</td><td>d0 (datos 0)</td></tr> <tr><td>VW206</td><td style="border: 1px solid black;">4142</td><td>d1 (datos 1)</td></tr> <tr><td>VW208</td><td style="border: 1px solid black;">3130</td><td>d2 (datos 2)</td></tr> <tr><td>VW210</td><td style="border: 1px solid black;">3030</td><td>d3 (datos 3)</td></tr> <tr><td>VW212</td><td style="border: 1px solid black;">3130</td><td>d4 (datos 4)</td></tr> <tr><td>VW214</td><td style="border: 1px solid black;">4541</td><td>d5 (datos 5)</td></tr> </table> <p>AC1 <input style="width: 50px;" type="text" value="0"/> AC1 se debe poner a 0 para poder iniciar la búsqueda desde el primer registro de la tabla.</p> <p>Buscar</p> <p>AC1 <input style="width: 50px;" type="text" value="2"/> AC1 contiene el número del primer registro que corresponde al criterio de búsqueda (d2).</p> <p>AC1 <input style="width: 50px;" type="text" value="3"/> Incrementar INDX en pasos de 1 antes de buscar los demás registros de la tabla.</p> <p>Buscar</p> <p>AC1 <input style="width: 50px;" type="text" value="4"/> AC1 contiene el número del segundo registro que corresponde al criterio de búsqueda (d4).</p> <p>AC1 <input style="width: 50px;" type="text" value="5"/> Incrementar INDX en pasos de 1 antes de buscar los demás registros de la tabla.</p> <p>Buscar</p> <p>AC1 <input style="width: 50px;" type="text" value="6"/> AC1 contiene un valor igual al número de registros. Se ha rastreado toda la tabla sin encontrar otro registro que corresponda al criterio de búsqueda.</p> <p>AC1 <input style="width: 50px;" type="text" value="0"/> Antes de que la tabla se pueda rastrear de nuevo, es preciso poner a 0 el valor de INDX.</p>		VW202	0006	EC (nº de registros)	VW204	3133	d0 (datos 0)	VW206	4142	d1 (datos 1)	VW208	3130	d2 (datos 2)	VW210	3030	d3 (datos 3)	VW212	3130	d4 (datos 4)	VW214	4541	d5 (datos 5)
VW202	0006	EC (nº de registros)																				
VW204	3133	d0 (datos 0)																				
VW206	4142	d1 (datos 1)																				
VW208	3130	d2 (datos 2)																				
VW210	3030	d3 (datos 3)																				
VW212	3130	d4 (datos 4)																				
VW214	4541	d5 (datos 5)																				

Figura 9-34 Ejemplos de una operación de búsqueda en KOP, AWL y FUP

### Borrar primer registro de la tabla



La operación **Borrar primer registro de la tabla** borra el primer registro de la tabla (TBL) y transfiere el valor a la dirección indicada (DATA). Todos los demás registros se desplazan una posición hacia arriba. El número de registros (EC) de la tabla decrementa cada vez que se ejecuta esta operación.

Condiciones de error que ponen ENO a 0: SM1.5 (tabla vacía), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto), 0091 (operando fuera de área)

Estas operaciones afectan a las siguientes marcas especiales: SM1.5 se activa si se intenta borrar un registro de una tabla vacía.

Entradas/salidas	Operandos	Tipos de datos
TABLE	VW, IW, QW, MW, SW, SMW, LW, T, C, *VD, *AC, *LD	WORD
DATA	VW, IW, QW, MW, SW, SMW, LW, AC, AQW, T, C, *VD, *AC, *LD	WORD

### Ejemplo de la operación Borrar primer registro de la tabla

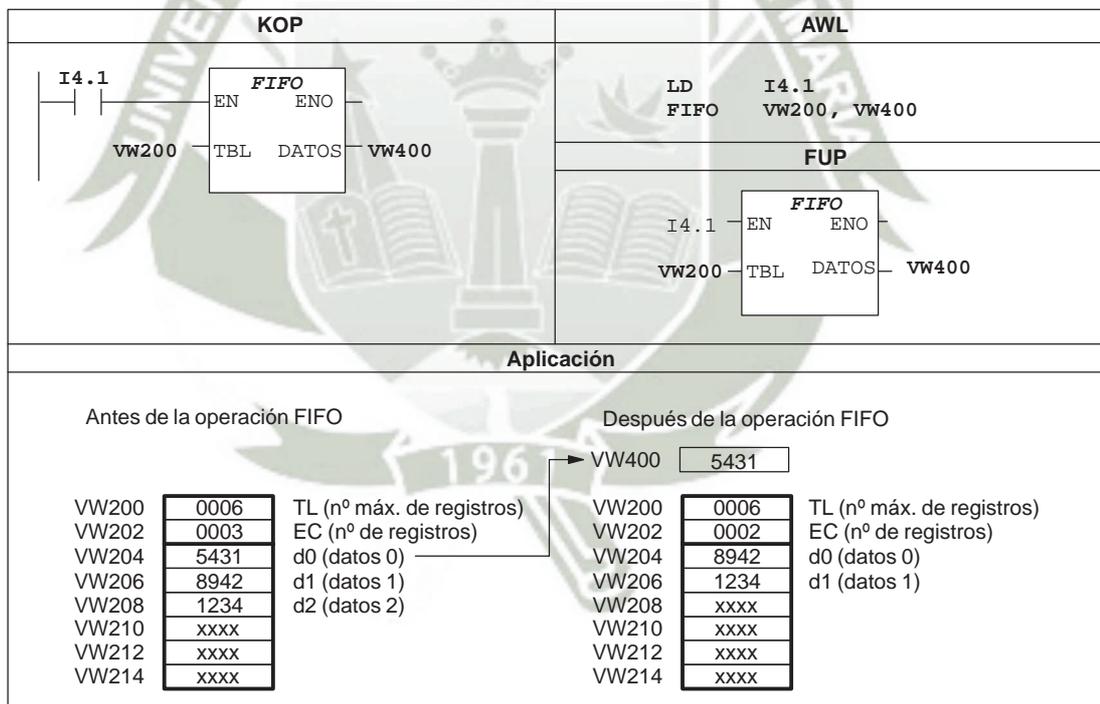
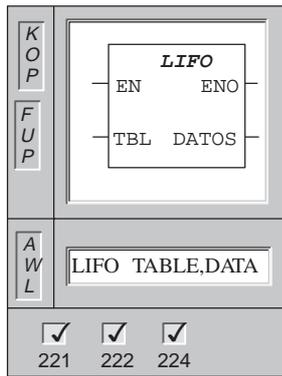


Figura 9-35 Ejemplo de la operación Borrar primer registro de la tabla (FIFO)

**Borrar último registro de la tabla**



La operación **Borrar último registro de la tabla** borra el último registro de la tabla (TBL) y transfiere el valor a la dirección indicada por DATA. El número de registros (EC) de la tabla decrementa cada vez que se ejecuta esta operación.

Condiciones de error que ponen ENO a 0: SM1.5 (tabla vacía), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto), 0091 (operando fuera de área)

Estas operaciones afectan a las siguientes marcas especiales: SM1.5 se activa si se intenta borrar un registro de una tabla vacía.

Entradas/salidas	Operandos	Tipos de datos
TABLE	VW, IW, QW, MW, SW, SMW, LW, T, C, *VD, *AC, *LD	WORD
DATA	VW, IW, QW, MW, SW, SMW, LW, AQW, T, C, AC, *VD, *AC, *LD	WORD

**Ejemplo de la operación Borrar último registro de la tabla**

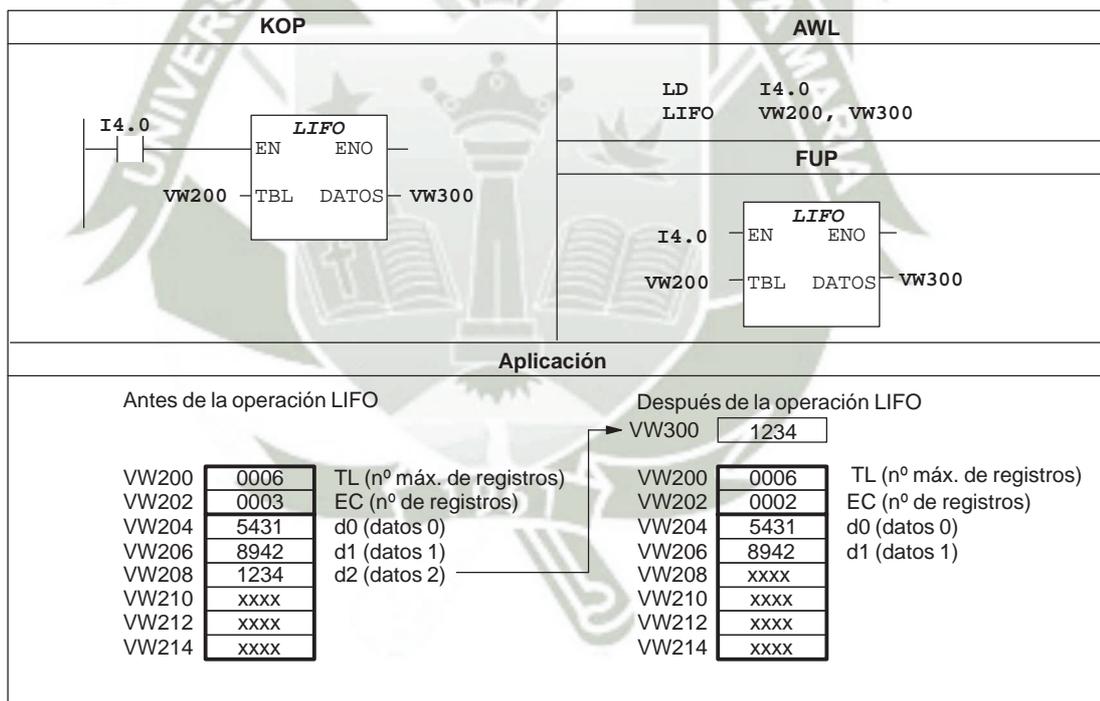
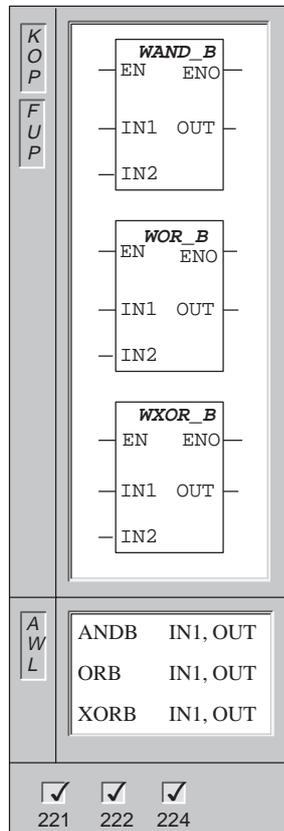


Figura 9-36 Ejemplo de la operación Borrar último registro de la tabla (LIFO)

## 9.12 Operaciones lógicas (SIMATIC)

### Combinación Y con bytes, Combinación O con bytes y Combinación O-exclusiva con bytes



La operación **Combinación Y con bytes** combina los bits correspondientes de los dos bytes de entrada mediante Y, y carga el resultado (OUT) en un byte.

La operación **Combinación O con bytes** combina los bits correspondientes de los dos bytes de entrada mediante O, y carga el resultado (OUT) en un byte.

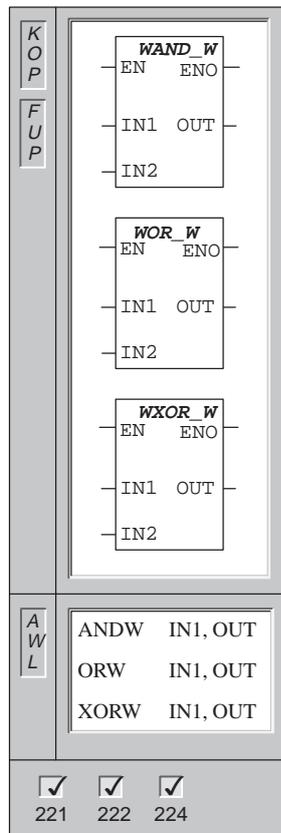
La operación **Combinación O-exclusiva con bytes** combina los bits correspondientes de los dos bytes de entrada mediante O-exclusiva, y carga el resultado (OUT) en un byte.

Condiciones de error que ponen ENO a 0: SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero)

Entradas/salidas	Operandos	Tipos de datos
IN1, IN2	VB, IB, QB, MB, SB, SMB, LB, AC, constante, *VD, *AC, *LD	BYTE
OUT	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD	BYTE

### Combinación Y con palabras, Combinación O con palabras y Combinación O-exclusiva con palabras



La operación **Combinación Y con palabras** combina los bits correspondientes de las dos palabras de entrada mediante Y, y carga el resultado (OUT) en una palabra.

La operación **Combinación O con palabras** combina los bits correspondientes de las dos palabras de entrada mediante O, y carga el resultado (OUT) en una palabra.

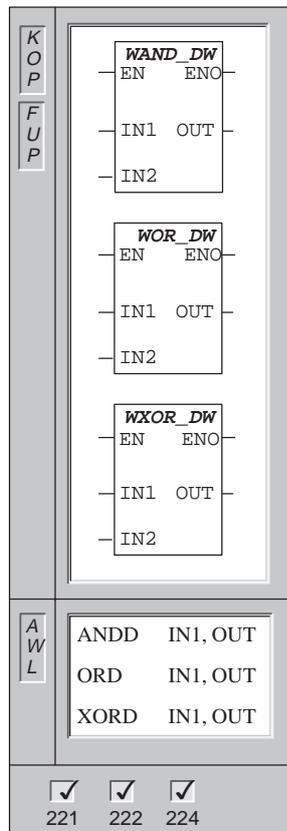
La operación **Combinación O-exclusiva con palabras** combina los bits correspondientes de las dos palabras de entrada mediante O-exclusiva, y carga el resultado (OUT) en una palabra.

Condiciones de error que ponen ENO a 0: SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero)

Entradas/salidas	Operandos	Tipos de datos
IN1, IN2	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, AC, constante, *VD, *AC, *LD	WORD
OUT	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, *VD, *AC, *LD	WORD

### Combinación Y con palabras dobles, Combinación O con palabras dobles y Combinación O-exclusiva con palabras dobles



La operación **Combinación Y con palabras dobles** combina los bits correspondientes de las dos palabras dobles de entrada mediante Y, y carga el resultado (OUT) en una palabra doble.

La operación **Combinación O con palabras dobles** combina los bits correspondientes de las dos palabras dobles de entrada mediante O, y carga el resultado (OUT) en una palabra doble.

La operación **Combinación O-exclusiva con palabras dobles** combina los bits correspondientes de las dos palabras dobles de entrada mediante O-exclusiva, y carga el resultado (OUT) en una palabra doble.

Condiciones de error que ponen ENO a 0: SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero)

Entradas/salidas	Operandos	Tipos de datos
IN1, IN2	VD, ID, QD, MD, SMD, AC, LD, HC, constante, *VD, *AC, SD, *LD	DWORD
OUT	VD, ID, QD, MD, SMD, LD, AC, *VD, *AC, SD, *LD	DWORD

Ejemplos de las operaciones de combinación con Y, O y O-exclusiva

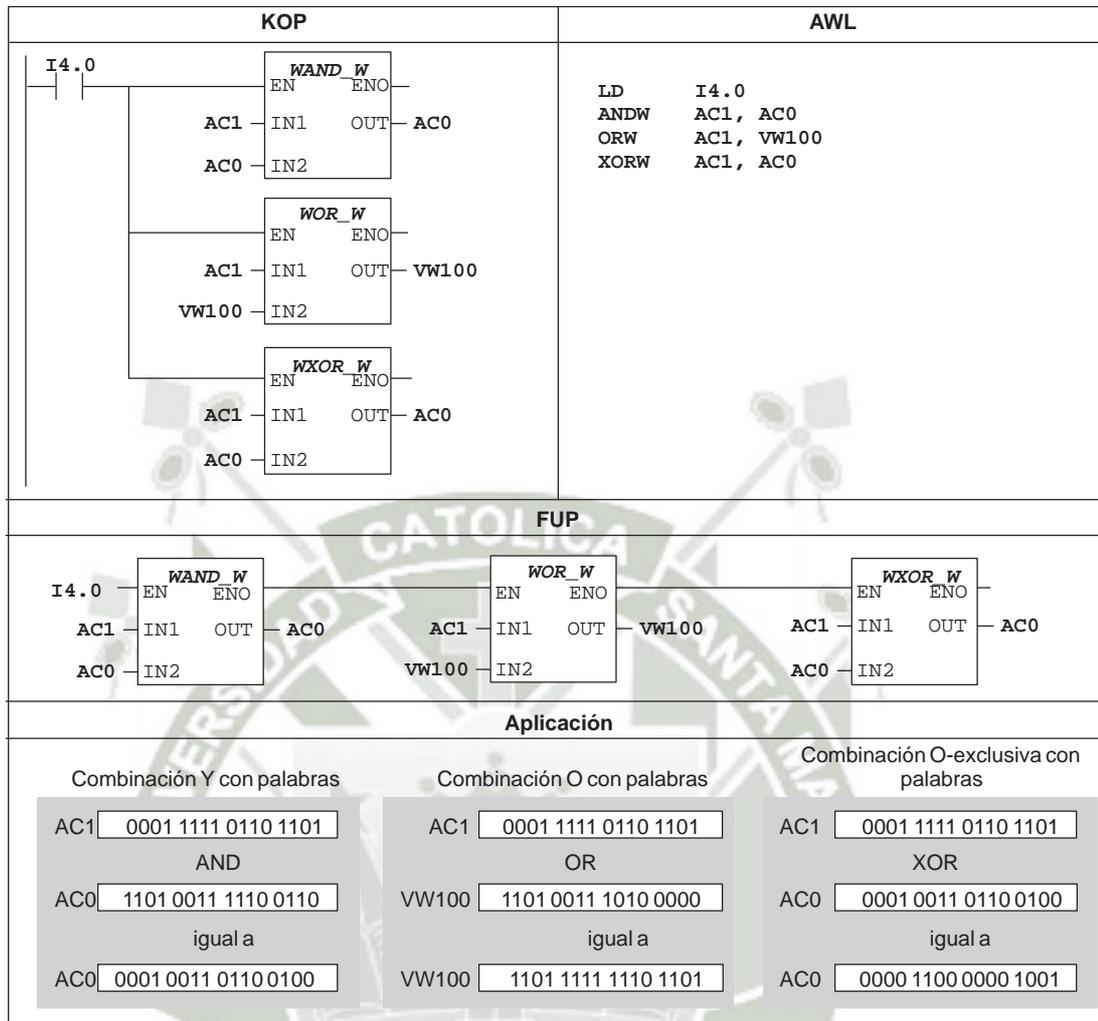
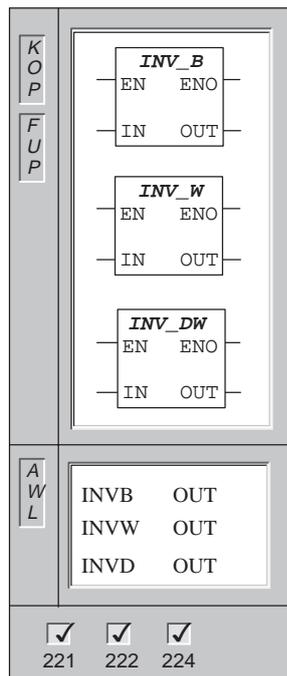


Figura 9-37 Ejemplo de las operaciones lógicas en KOP, AWL y FUP

### Invertir byte, Invertir palabra, Invertir palabra doble



La operación **Invertir byte** forma el complemento a 1 del valor del byte de entrada IN y carga el resultado en el valor de byte OUT.

La operación **Invertir palabra** forma el complemento a 1 del valor de la palabra de entrada IN y carga el resultado en el valor de palabra OUT.

La operación **Invertir palabra doble** forma el complemento a 1 del valor de la palabra doble de entrada IN y carga el resultado en el valor de palabra doble OUT.

Condiciones de error que ponen ENO a 0: SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero)

Invertir...	Entradas/salidas	Operandos	Tipos de datos
BYTE	IN	VB, IB, QB, MB, SB, SMB, LB, AC, constante, *VD, *AC, *LD	BYTE
	OUT	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD	BYTE
WORD	IN	VW, IW, QW, MW, SW, SMW, T, C, AIW, LW, AC, constante, *VD, *AC, *LD	WORD
	OUT	VW, IW, QW, MW, SW, SMW, T, C, LW, AC, *VD, *AC, *LD	WORD
Palabra doble	IN	VD, ID, QD, MD, SD, SMD, LD, HC, AC, constante, *VD, *AC, *LD	DWORD
	OUT	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *AC, *LD	DWORD

Ejemplo de la operación Invertir

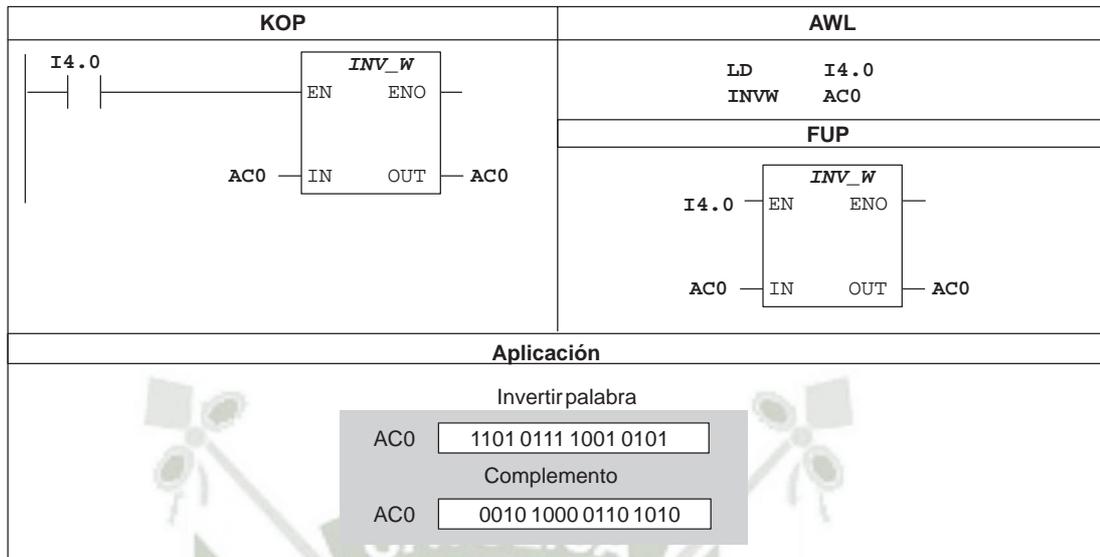
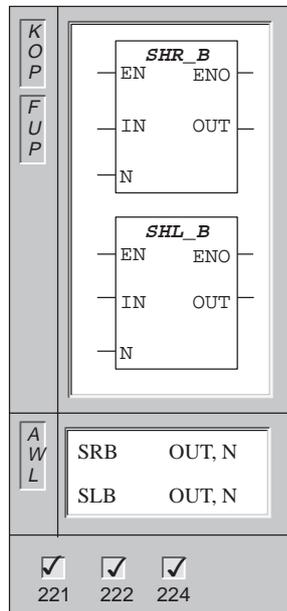


Figura 9-38 Ejemplo de una operación Invertir en KOP, AWL y FUP

### 9.13 Operaciones de desplazamiento y rotación (SIMATIC)

#### Desplazar byte a la derecha, Desplazar byte a la izquierda



Las operaciones **Desplazar byte a la derecha** y **Desplazar byte a la izquierda** desplazan el valor del byte de entrada (IN) a la derecha y a la izquierda respectivamente, tantas posiciones como indique el valor de desplazamiento (N), y cargan el resultado en el byte de salida (OUT).

Las **operaciones de desplazamiento** se rellenan con ceros cada vez que se desplaza un bit. Si el valor de desplazamiento (N) es mayor o igual a 8, el valor se desplazará como máximo 8 veces.

Si el valor de desplazamiento es mayor que 0, la marca de desbordamiento (SM1.1) adoptará el valor del último bit desplazado hacia afuera. La marca cero (SM1.0) se activará si el resultado de la operación de desplazamiento es cero.

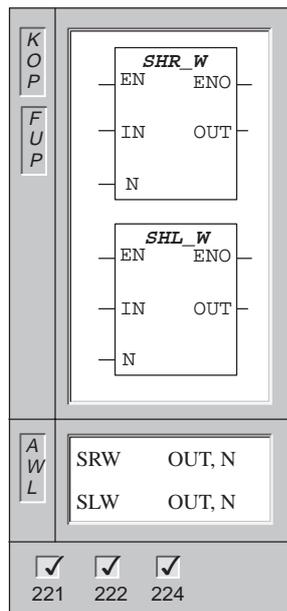
Las operaciones de desplazamiento de bytes no llevan signo.

Condiciones de error que ponen ENO a 0: SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero); SM1.1 (desbordamiento)

Entradas/salidas	Operandos	Tipos de datos
IN, OUT	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD	BYTE
N	VB, IB, QB, MB, SB, SMB, LB, AC, constante, *VD, *AC, *LD	BYTE

### Desplazar palabra a la derecha, Desplazar palabra a la izquierda



Las operaciones **Desplazar palabra a la derecha** y **Desplazar palabra a la izquierda** desplazan el valor de la palabra de entrada (IN) a la derecha y a la izquierda respectivamente, tantas posiciones como indique el valor de desplazamiento (N) y cargan el resultado en la palabra de salida (OUT).

Las **operaciones de desplazamiento** se rellenan con ceros cada vez que se desplaza un bit. Si el valor de desplazamiento (N) es mayor o igual a 16, el valor se desplazará como máximo 16 veces. Si el valor de desplazamiento es mayor que 0, la marca de desbordamiento (SM1.1) adoptará el valor del último bit desplazado hacia afuera. La marca cero (SM1.0) se activará si el resultado de la operación de desplazamiento es cero.

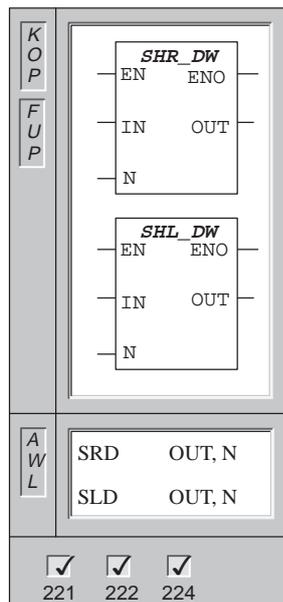
Las operaciones de desplazamiento de palabras no llevan signo.

Condiciones de error que ponen ENO a 0: SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero); SM1.1 (desbordamiento)

Entradas/salidas	Operandos	Tipos de datos
IN	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, AC, constante, *VD, *AC, *LD	WORD
N	VB, IB, QB, MB, SB, SMB, LB, AC, constante, *VD, *AC, *LD	BYTE
OUT	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, *VD, *AC, *LD	WORD

### Desplazar palabra doble a la derecha, Desplazar palabra doble a la izquierda



Las operaciones **Desplazar palabra doble a la derecha** y **Desplazar palabra doble a la izquierda** desplazan el valor de la palabra doble de entrada (IN) a la derecha y a la izquierda respectivamente, tantas posiciones como indique el valor de desplazamiento (N) y cargan el resultado en la palabra doble de salida (OUT).

Las **operaciones de desplazamiento** se rellenan con ceros cada vez que se desplaza un bit. Si el valor de desplazamiento (N) es mayor o igual a 32, el valor se desplazará como máximo 32 veces. Si el valor de desplazamiento es mayor que 0, la marca de desbordamiento (SM1.1) adoptará el valor del último bit desplazado hacia afuera. La marca cero (SM1.0) se activará si el resultado de la operación de desplazamiento es cero.

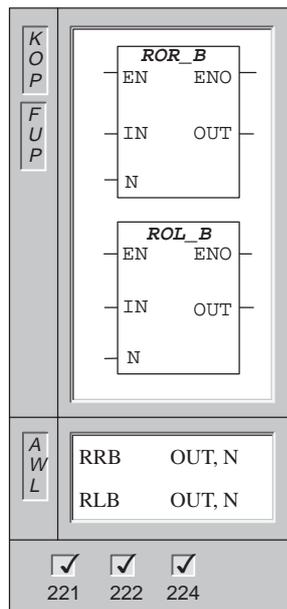
Las operaciones de desplazamiento de palabras dobles no llevan signo.

Condiciones de error que ponen ENO a 0: SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero); SM1.1 (desbordamiento)

Entradas/salidas	Operandos	Tipos de datos
IN	VD, ID, QD, MD, SD, SMD, LD, AC, HC, constante, *VD, *AC, *LD	DWORD
N	VB, IB, QB, MB, SB, SMB, LB, AC, constante, *VD, *AC, *LD	BYTE
OUT	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *AC, *LD	DWORD

### Rotar byte a la derecha, Rotar byte a la izquierda



Las operaciones **Rotar byte a la derecha** y **Rotar byte a la izquierda** rotan el valor del byte de entrada (IN) a la derecha y a la izquierda respectivamente, tantas posiciones como indique el valor de desplazamiento (N) y cargan el resultado en el byte de salida (OUT).

Si el valor de desplazamiento (N) es mayor o igual a 8, antes de la operación de rotación se ejecutará una operación módulo 8 en el valor de desplazamiento (N). De ello resulta un valor de rotación de 0 a 7. Si el valor de desplazamiento es igual a 0, no se rotará el valor. Si se ejecuta la rotación, el valor del último bit rotado se copiará en la marca de desbordamiento (SM1.1).

Si el valor de desplazamiento no es un entero múltiplo de 8, el último bit rotado se copiará en la marca de desbordamiento (SM1.1). La marca cero (SM1.0) se activará si el valor a rotar es igual a cero.

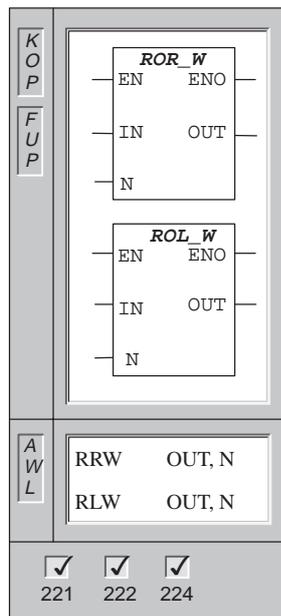
Las operaciones de rotación de bytes no llevan signo.

Condiciones de error que ponen ENO a 0: SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero); SM1.1 (desbordamiento)

Entradas/salidas	Operandos	Tipos de datos
IN	VB, IB, QB, MB, SMB, SB, LB, AC, *VD, *AC, *LD	BYTE
N	VB, IB, QB, MB, SMB, SB, LB, AC, constante, *VD, *AC, *LD	BYTE
OUT	VB, IB, QB, MB, SMB, SB, LB, AC, *VD, *AC, *LD	BYTE

### Rotar palabra a la derecha, Rotar palabra a la izquierda



Las operaciones **Rotar palabra a la derecha** y **Rotar palabra a la izquierda** rotan el valor de la palabra de entrada (IN) a la derecha y a la izquierda respectivamente, tantas posiciones como indique el valor de desplazamiento (N) y cargan el resultado en la palabra de salida (OUT).

Si el valor de desplazamiento (N) es mayor o igual a 16, antes de la operación de rotación se ejecutará una operación módulo 16 en el valor de desplazamiento (N). De ello resulta un valor de rotación de 0 a 15. Si el valor de desplazamiento es igual a 0, no se rotará el valor. Si se ejecuta la rotación, el valor del último bit rotado se copiará en la marca de desbordamiento (SM1.1).

Si el valor de desplazamiento no es un entero múltiplo de 16, el último bit rotado se copiará en la marca de desbordamiento (SM1.1). La marca cero (SM1.0) se activará si el valor a rotar es igual a cero.

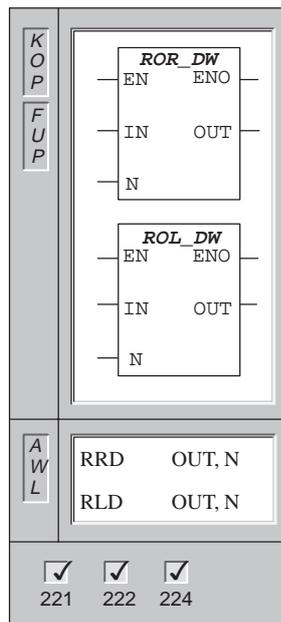
Las operaciones de rotación de palabras dobles no llevan signo.

Condiciones de error que ponen ENO a 0: SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero); SM1.1 (desbordamiento)

Entradas/salidas	Operandos	Tipos de datos
IN	VW, T, C, IW, MW, SMW, AC, QW, LW, AIW, constante, *VD, *AC, SW, *LD	WORD
N	VB, IB, QB, MB, SMB, LB, AC, constante, *VD, *AC, SB, *LD	BYTE
OUT	VW, T, C, IW, QW, MW, SMW, LW, AC, *VD, *AC, SW, *LD	WORD

### Rotar palabra doble a la derecha, Rotar palabra doble a la izquierda



Las operaciones **Rotar palabra doble a la derecha** y **Rotar palabra doble a la izquierda** rotan el valor de la palabra doble de entrada (IN) a la derecha y a la izquierda respectivamente, tantas posiciones como indique el valor de desplazamiento (N) y cargan el resultado en la palabra doble de salida (OUT).

Si el valor de desplazamiento (N) es mayor o igual a 32, antes de la operación de rotación se ejecutará una operación módulo 32 en el valor de desplazamiento (N). De ello resulta un valor de rotación de 0 a 31. Si el valor de desplazamiento es igual a 0, no se rotará el valor. Si se ejecuta la rotación, el valor del último bit rotado se copiará en la marca de desbordamiento (SM1.1).

Si el valor de desplazamiento no es un entero múltiplo de 32, el último bit rotado se copiará en la marca de desbordamiento (SM1.1). La marca cero (SM1.0) se activará si el valor a rotar es igual a cero.

Las operaciones de rotación de palabras dobles no llevan signo.

Condiciones de error que ponen ENO a 0: SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero); SM1.1 (desbordamiento)

Entradas/salidas	Operandos	Tipos de datos
IN	VD, ID, QD, MD, SMD, LD, AC, HC, constante, *VD, *AC, SD, *LD	DWORD
N	VB, IB, QB, MB, SMB, LB, AC, constante, *VD, *AC, SB, *LD	BYTE
OUT	VD, ID, QD, MD, SMD, LD, AC, *VD, *AC, SD, *LD	DWORD

Ejemplos de operaciones de rotación y desplazamiento

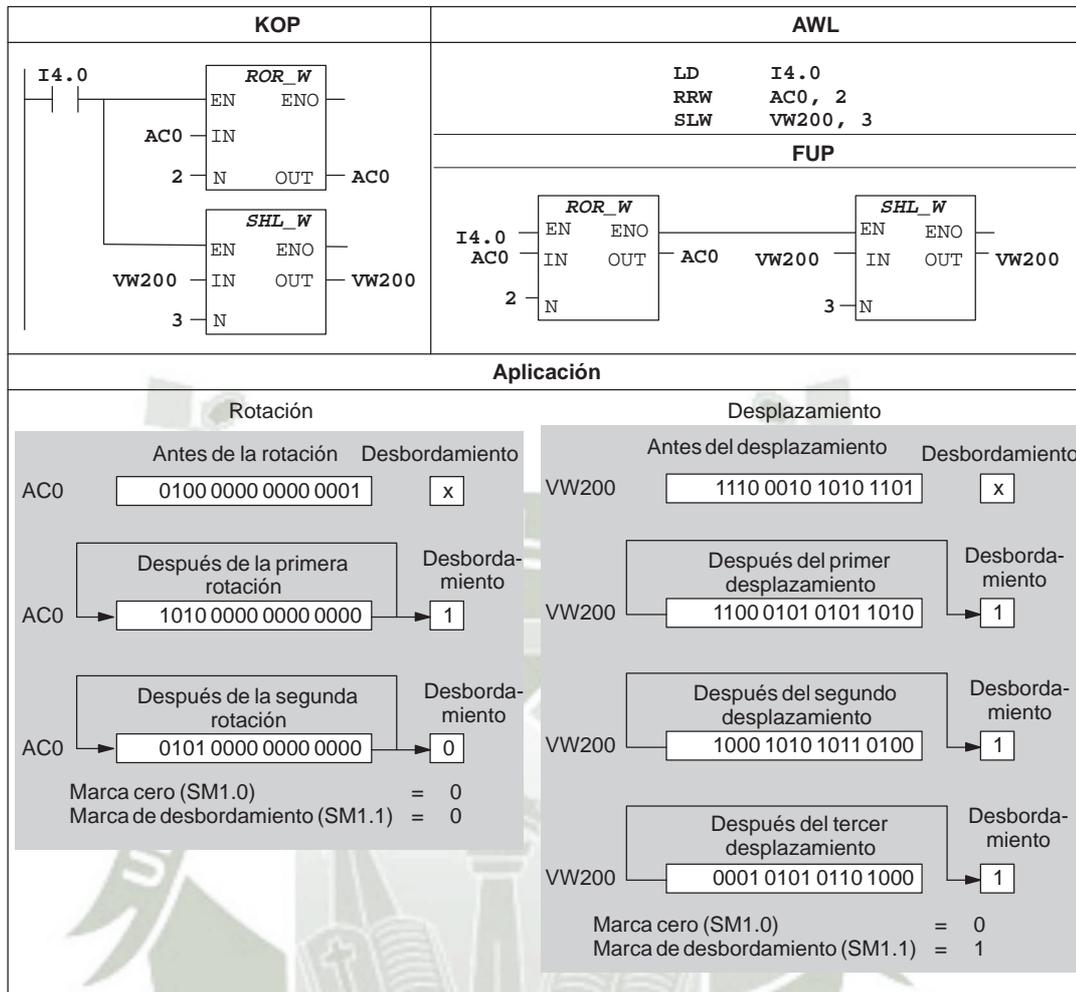
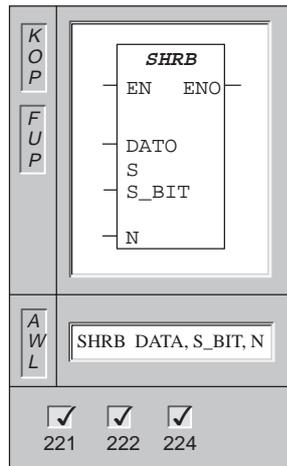


Figura 9-39 Ejemplo de operaciones de desplazamiento y rotación en KOP, AWL y FUP

### Registro de desplazamiento



La operación **Registro de desplazamiento** (SHRB) desplaza el valor de DATA al registro de desplazamiento. S\_BIT indica el bit menos significativo de dicho registro. N indica la longitud del registro y el sentido de desplazamiento (valor positivo = N, valor negativo = -N).

Los bits desplazados por la operación Registro de desplazamiento se depositan en la marca de desbordamiento (SM1.1).

Condiciones de error que ponen ENO a 0: SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto), 0091 (operando fuera de área), 0092 (error en campo de contaje)

Estas operaciones afectan a las siguientes marcas especiales: SM1.1 (desbordamiento)

Entradas/salidas	Operandos	Tipos de datos
DATA, S_BIT	I, Q, M, SM, T, C, V, S, L	BOOL
N	VB, IB, QB, MB, SMB, LB, AC, constante, *VD, *AC, SB, *LD	BYTE

### Descripción de la operación Registro de desplazamiento

La operación Registro de desplazamiento permite secuenciar y controlar fácilmente el flujo de productos o de datos. Esta operación se debe utilizar para desplazar todo el registro un bit en cada ciclo. El registro de desplazamiento está definido por el bit menos significativo (S\_BIT) y por el número de bits indicados por la longitud (N). La figura 9-41 muestra un ejemplo de la operación Registro de desplazamiento.

La dirección del bit más significativo del registro de desplazamiento (MSB.b) se calcula con la siguiente ecuación:

$$MSB.b = [(\text{byte de S\_BIT}) + ([N] - 1 + (\text{bit de S\_BIT})) / 8] \cdot [\text{resto de la división por 8}]$$

Se debe restar 1 bit, porque S\_BIT es uno de los bits del registro de desplazamiento.

Por ejemplo, si S\_BIT es V33.4 y N es 14, el bit MSB.b será V35.1 ó:

$$\begin{aligned} MSB.b &= V33 + ([14] - 1 + 4) / 8 \\ &= V33 + 17 / 8 \\ &= V33 + 2 \text{ con el resto de } 1 \\ &= V35.1 \end{aligned}$$

Si el valor de desplazamiento es negativo, es decir, si la longitud (N) indicada es negativa, los datos de entrada se desplazarán desde el bit menos significativo (S\_BIT) al bit más significativo del registro de desplazamiento.

Si el valor de desplazamiento es positivo, es decir, si la longitud (N) indicada es positiva, los datos de entrada (DATA) se desplazarán desde el bit más significativo al bit menos significativo (indicado por S\_BIT) del registro de desplazamiento.

Los datos desplazados se depositan en la marca de desbordamiento (SM1.1). El registro de desplazamiento puede tener una longitud máxima de 64 bits (positiva o negativa). La figura 9-40 muestra el desplazamiento de bits de un valor N positivo y de un valor N negativo.

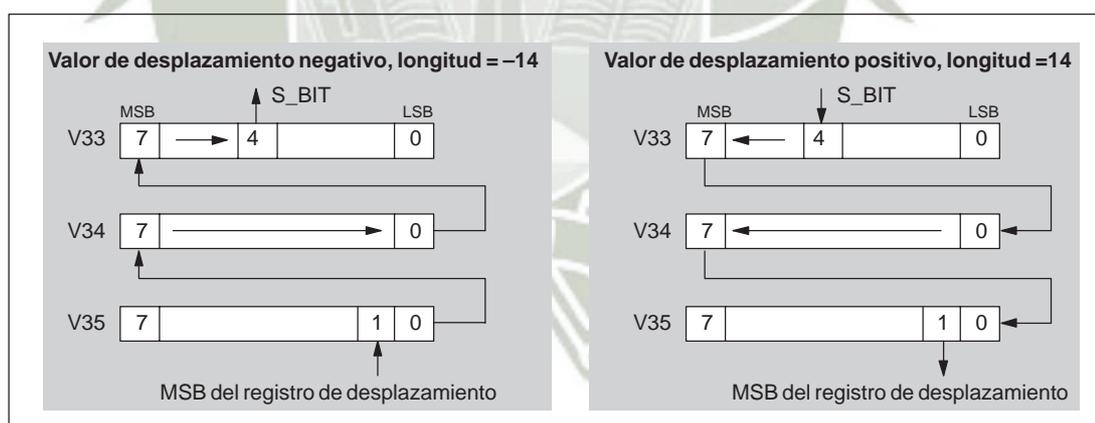


Figura 9-40 Entrada y salida de valores positivos y negativos en el registro de desplazamiento

Ejemplo de la operación Registro de desplazamiento

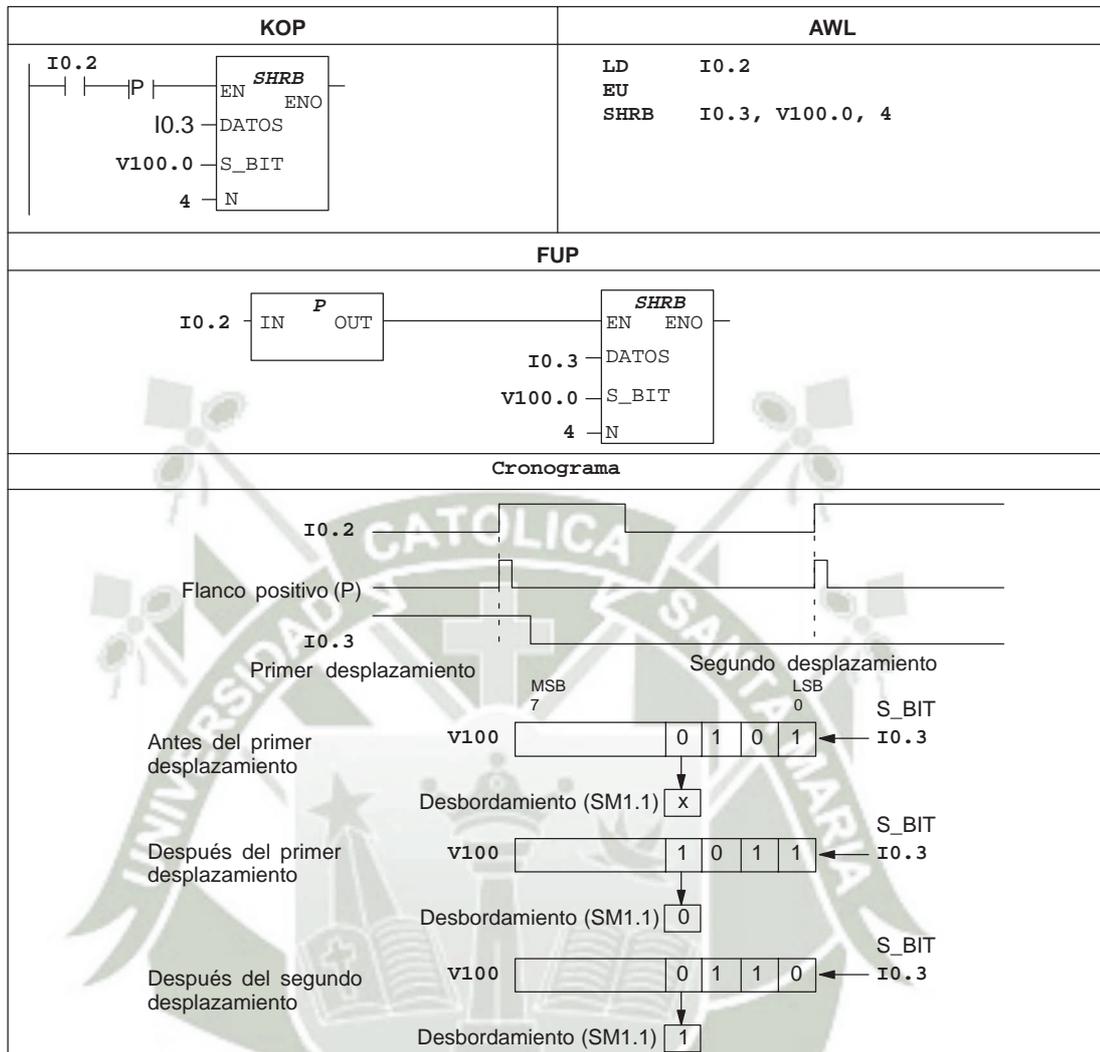
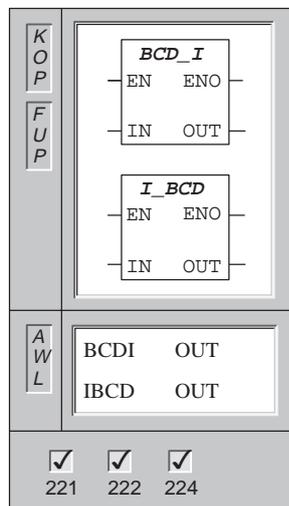


Figura 9-41 Ejemplo de la operación Registro de desplazamiento en KOP, AWL y FUP

## 9.14 Operaciones de conversión (SIMATIC)

### Convertir de BCD a entero, Convertir de entero a BCD



La operación **Convertir de BCD a entero** convierte el valor BCD de entrada (IN) en un valor de entero y carga el resultado en la variable indicada por OUT. El margen válido de IN está comprendido entre 0 y 9999 BCD.

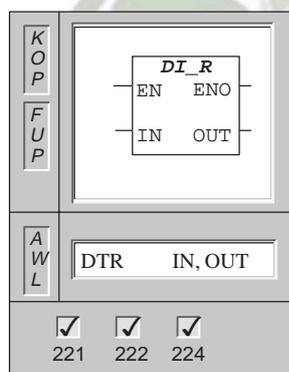
La operación **Convertir de entero a BCD** convierte el valor entero de entrada (IN) en un valor BCD y carga el resultado en la variable indicada por OUT. El margen válido de IN está comprendido entre 0 y 9999 entero.

Condiciones de error que ponen ENO a 0: SM1.6 (error BCD), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Estas operaciones afectan a las siguientes marcas especiales: SM1.6 (BCD no válido)

Entradas/salidas	Operandos	Tipos de datos
IN	VW, T, C, IW, QW, MW, SMW, LW, AC, AIW, constante, *VD, *AC, SW, *LD	WORD
OUT	VW, T, C, IW, QW, MW, SMW, LW, AC, *VD, *AC, SW, *LD	WORD

### Convertir de entero doble a real

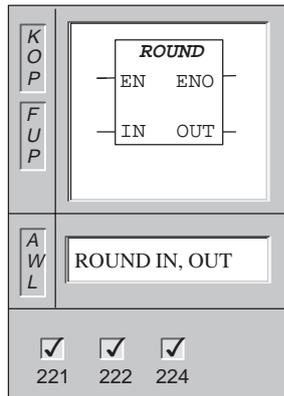


La operación **Convertir de entero doble a real** convierte un entero de 32 bits con signo (IN) en un número real de 32 bits y deposita el resultado en la variable indicada por OUT.

Condiciones de error que ponen ENO a 0: SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Entradas/salidas	Operandos	Tipos de datos
IN	VD, ID, QD, MD, SMD, AC, LD, HC, constante, *VD, *AC, SD, *LD	DINT
OUT	VD, ID, QD, MD, SMD, LD, AC, *VD, *AC, SD, *LD	REAL

### Redondear a entero doble



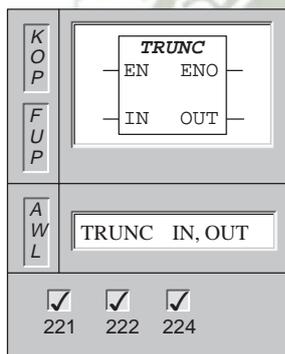
La operación **Redondear a entero doble** convierte el valor real (IN) en un valor de entero doble y deposita el resultado en la variable indicada por OUT. Si la fracción es 0,5 o superior, el número se redondeará al próximo entero superior.

Condiciones de error que ponen ENO a 0: SM1.1 (desbordamiento), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Estas operaciones afectan a las siguientes marcas especiales: SM1.1 (desbordamiento)

Entradas/salidas	Operandos	Tipos de datos
IN	VD, ID, QD, MD, SMD, AC, LD, HC, constante, *VD, *AC, SD, *LD	REAL
OUT	VD, ID, QD, MD, SMD, LD, AC, *VD, *AC, SD, *LD	DINT

### Truncar



La operación **Truncar** convierte un número real de 32 bits (IN) en un entero de 32 bits con signo y carga el resultado en la variable indicada por OUT. Sólo se convierte la parte entera del número real y la fracción se pierde.

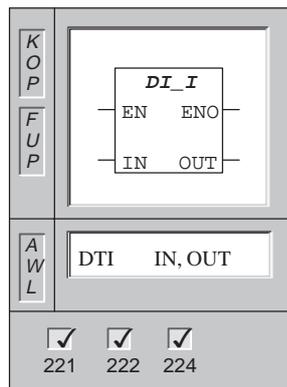
Si el valor a convertir no es un número real válido o si es demasiado grande para ser representado en la salida, la marca de desbordamiento se activará y la salida no se verá afectada.

Condiciones de error que ponen ENO a 0: SM1.1 (desbordamiento), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Estas operaciones afectan a las siguientes marcas especiales: SM1.1 (desbordamiento)

Entradas/salidas	Operandos	Tipos de datos
IN	VD, ID, QD, MD, SMD, LD, AC, constante, *VD, *AC, SD, *LD	REAL
OUT	VD, ID, QD, MD, SMD, LD, AC, *VD, *AC, SD, *LD	DINT

### Convertir de entero doble a entero



La operación **Convertir de entero doble a entero** convierte el valor de entero doble (IN) en un valor de entero y deposita el resultado en la variable indicada por OUT.

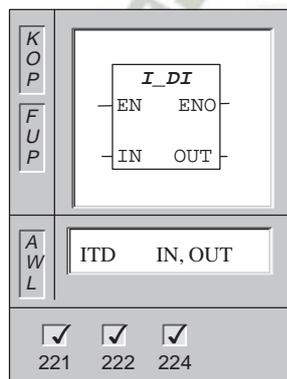
Si el valor a convertir es demasiado grande para ser representado en la salida, la marca de desbordamiento se activará y la salida no se verá afectada.

Condiciones de error que ponen ENO a 0: SM1.1 (desbordamiento), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Estas operaciones afectan a las siguientes marcas especiales: SM1.1 (desbordamiento)

Entradas/salidas	Operandos	Tipos de datos
IN	VD, ID, QD, MD, SMD, AC, LD, HC, constante, *VD, *AC, SD, *LD	DINT
OUT	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, *VD, *LD, *AC	INT

### Convertir de entero a entero doble

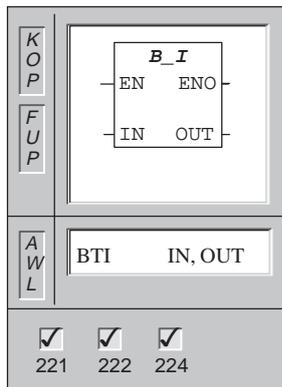


La operación **Convertir de entero a entero doble** convierte el valor de entero (IN) en un valor de entero doble y deposita el resultado en la variable indicada por OUT. El signo se amplía.

Condiciones de error que ponen ENO a 0: SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Entradas/salidas	Operandos	Tipos de datos
IN	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, AC, constante, *AC, *VD, *LD	INT
OUT	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *LD, *AC	DINT

### Convertir de byte a entero

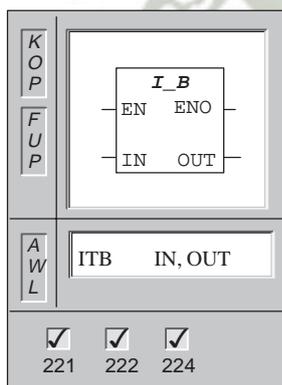


La operación **Convertir de byte a entero** convierte el valor de byte (IN) en un valor de entero y deposita el resultado en la variable indicada por OUT. El byte no tiene signo. Por tanto, no hay ampliación de signo.

Condiciones de error que ponen ENO a 0: SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Entradas/salidas	Operandos	Tipos de datos
IN	VB, IB, QB, MB, SB, SMB, LB, AC, constante, *AC, *VD, *LD	BYTE
OUT	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, *VD, *LD, *AC	INT

### Convertir de entero a byte



La operación **Convertir de entero a byte** convierte el valor de entero (IN) en un valor de byte y deposita el resultado en la variable indicada por OUT.

Se convierten los valores comprendidos entre 0 y 255. Todos los demás valores producen un desbordamiento y la salida no se ve afectada.

Condiciones de error que ponen ENO a 0: SM1.1 (desbordamiento), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Estas operaciones afectan a las siguientes marcas especiales: SM1.1 (desbordamiento)

Entradas/salidas	Operandos	Tipos de datos
IN	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, AC, constante, *VD, *LD, *AC	INT
OUT	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD	BYTE

### Ejemplos de conversión

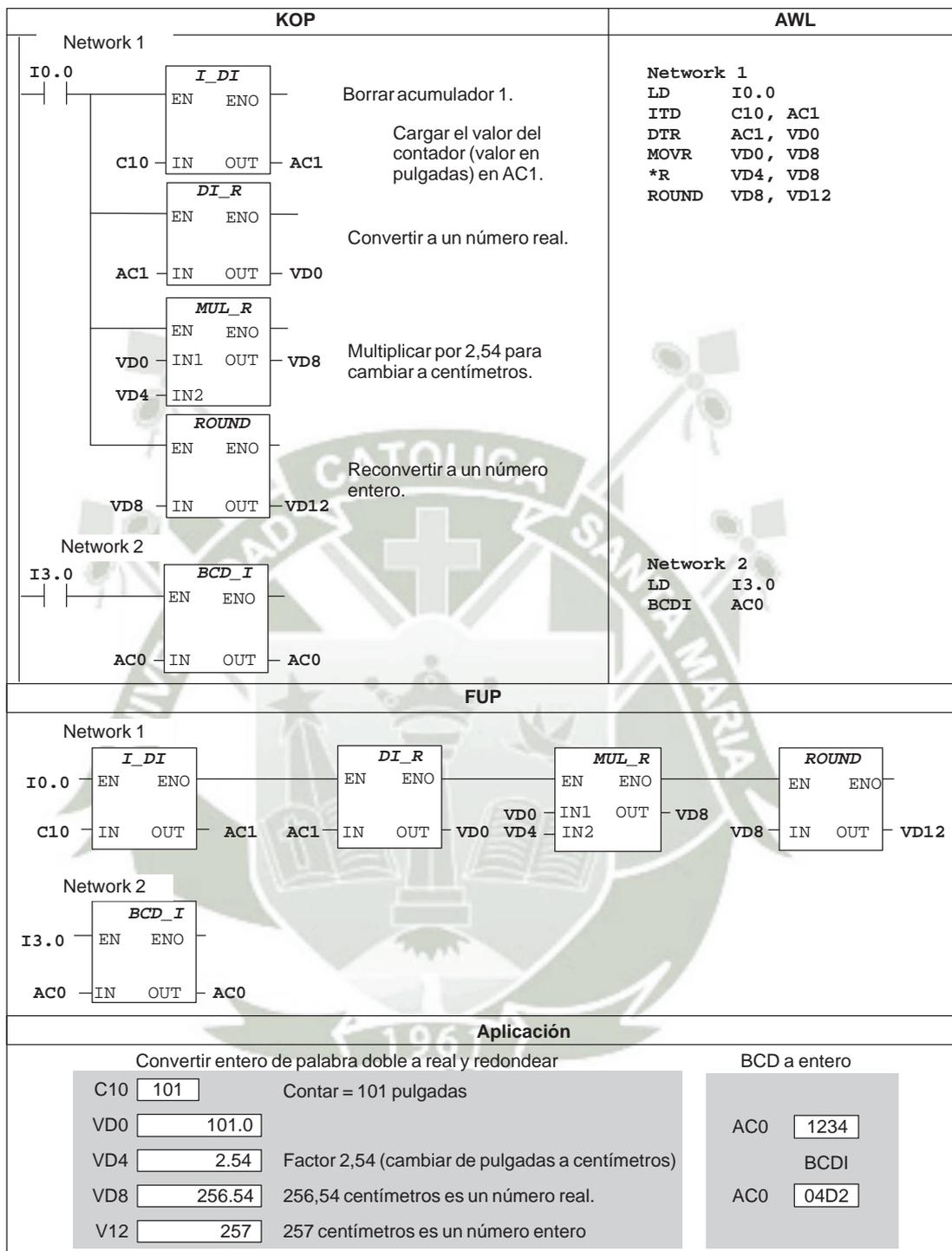
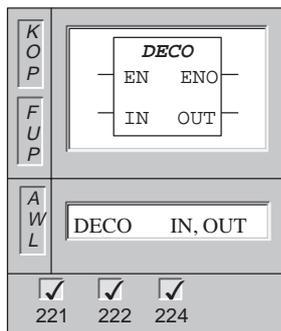


Figura 9-42 Ejemplo de las operaciones de conversión

### Decodificar

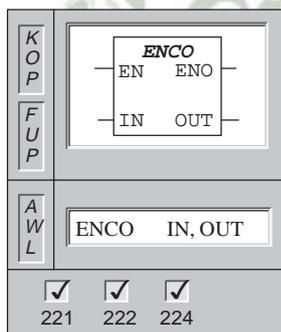


La operación **Decodificar** activa el bit de la palabra de salida (OUT). Dicho bit corresponde al número de bit representado por el medio byte menos significativo (4 bits) del byte de entrada (IN). Todos los demás bits de la palabra de salida se ponen a 0.

Condiciones de error que ponen ENO a 0: SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Entradas/salidas	Operandos	Tipos de datos
IN	VB, IB, QB, MB, SMB, LB, SB, AC, constante, *VD, *AC, *LD	BYTE
OUT	VW, IW, QW, MW, SMW, LW, SW, AQW, T, C, AC, *VD, *AC, *LD	WORD

### Codificar



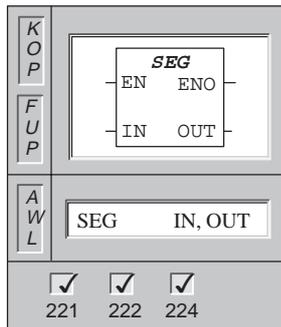
La operación **Codificar** escribe el número del bit menos significativo de la palabra de entrada (IN) en el medio byte menos significativo (4 bits) del byte de salida (OUT).

Condiciones de error que ponen ENO a 0: SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Entradas/salidas	Operandos	Tipos de datos
IN	VW, T, C, IW, QW, MW, SMW, AC, LW, AIW, constante, *VD, *AC, SW, *LD	WORD
OUT	VB, IB, QB, MB, SMB, LB, AC, *VD, *AC, SB, *LD	BYTE



**Segmento**



La operación **Segmento** utiliza el carácter indicado por IN para generar una configuración binaria (OUT) que ilumina los segmentos de un indicador de siete segmentos. Los segmentos iluminados representan el carácter depositado en el dígito menos significativo del byte de entrada (IN).

Condiciones de error que ponen ENO a 0: SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

La figura 9-45 muestra la codificación del indicador de siete segmentos utilizado por la operación Segmento.

Entradas/salidas	Operandos	Tipos de datos
IN	VB, IB, QB, MB, SMB, LB, AC, constante, *VD, *AC, SB, *LD	BYTE
OUT	VB, IB, QB, MB, SMB, LB, AC, *VD, *AC, SB, *LD	BYTE

(IN) LSD	Indicador Segmentos	(OUT) -gfe dcba	(IN) LSD	Indicador Segmentos	(OUT) -gfe dcba
0		0011 1111	8		0111 1111
1		0000 0110	9		0110 0111
2		0101 1011	A		0111 0111
3		0100 1111	B		0111 1100
4		0110 0110	C		0011 1001
5		0110 1101	D		0101 1110
6		0111 1101	E		0111 1001
7		0000 0111	F		0111 0001

Figura 9-45 Codificación del indicador de siete segmentos

### Ejemplo de la operación Segmento

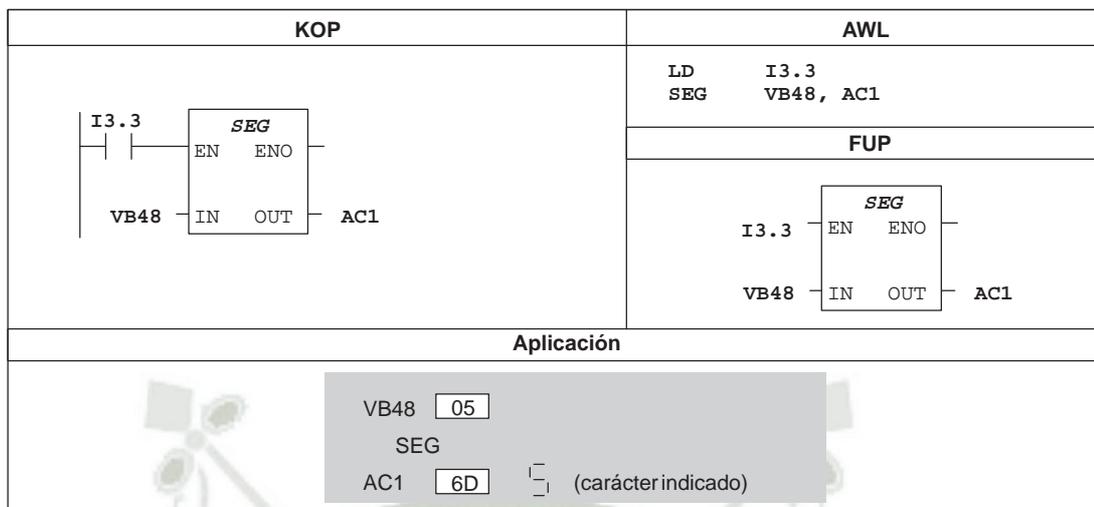
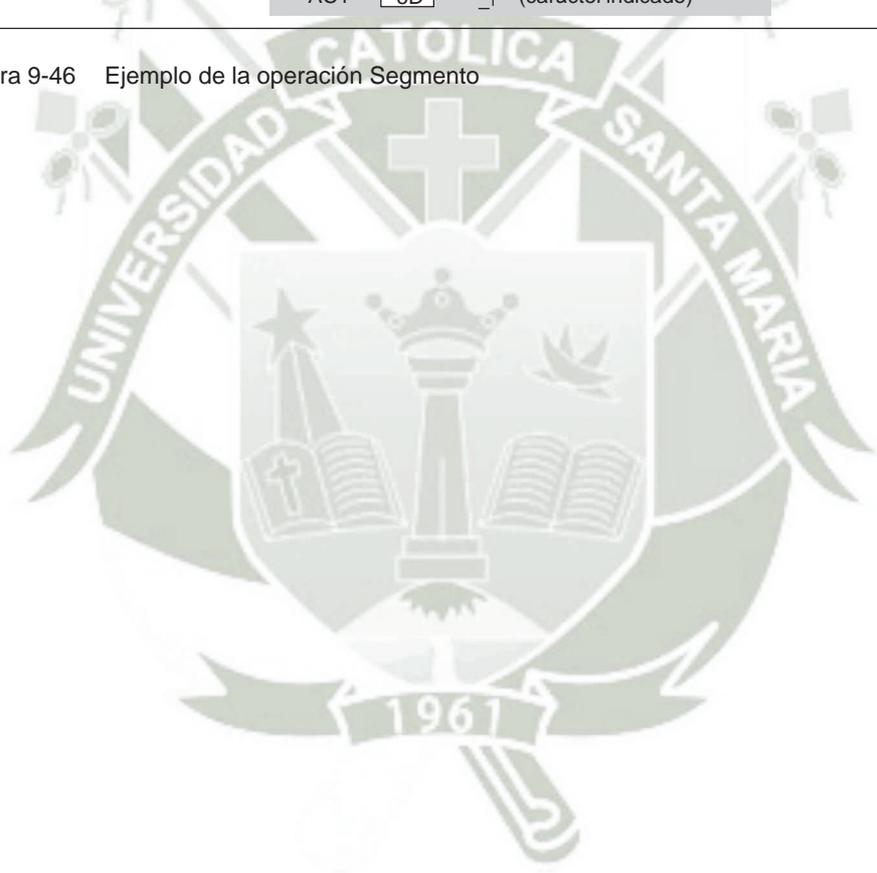
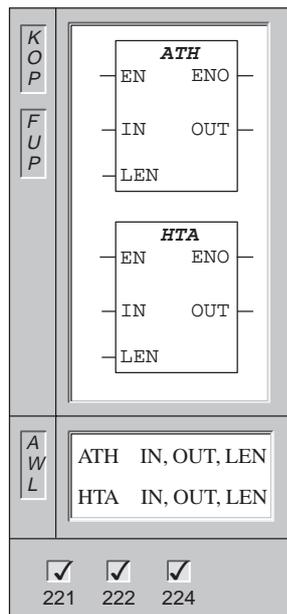


Figura 9-46 Ejemplo de la operación Segmento



### Convertir de ASCII a hexadecimal, Convertir de hexadecimal a ASCII



La operación **Convertir de ASCII a hexadecimal** convierte la cadena ASCII de longitud (LEN), a partir del carácter IN, en dígitos hexadecimales, comenzando en OUT. La cadena ASCII puede tener una longitud máxima de 255 caracteres.

La operación **Convertir de hexadecimal a ASCII** convierte los dígitos hexadecimales a partir del byte de entrada (IN) en una cadena ASCII, comenzando en OUT. El número de dígitos hexadecimales a convertir viene indicado por la longitud (LEN). Es posible convertir 255 dígitos hexadecimales como máximo.

Los caracteres ASCII admisibles son los valores hexadecimales 30 a 39 y 41 a 46.

Convertir de ASCII a hexadecimal: Condiciones de error que ponen ENO a 0: SM1.7 (ASCII no válido), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto), 0091 (operando fuera de área)

Convertir de hexadecimal a ASCII: Condiciones de error que ponen ENO a 0: SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto), 0091 (operando fuera de área)

Estas operaciones afectan a las siguientes marcas especiales: SM1.7 (ASCII no válido)

Entradas/salidas	Operandos	Tipos de datos
IN, OUT	VB, IB, QB, MB, SMB, LB, *VD, *AC, SB, *LD	BYTE
LEN	VB, IB, QB, MB, SMB, LB, AC, constante, *VD, *AC, SB, *LD	BYTE

### Ejemplo de la operación Convertir de ASCII a hexadecimal

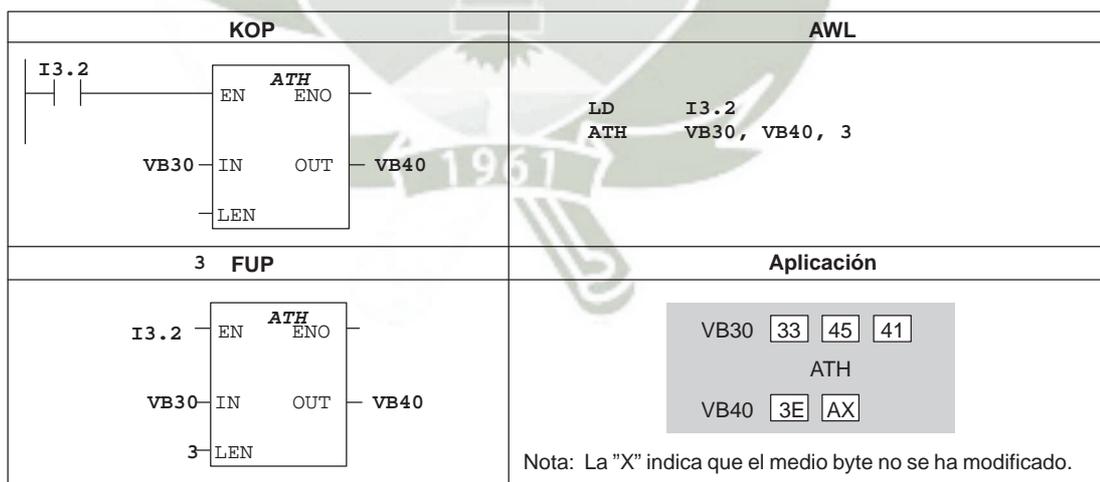
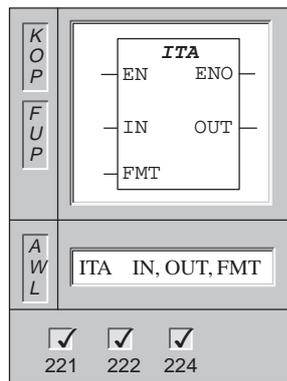


Figura 9-47 Ejemplo de la operación Convertir de ASCII a hexadecimal

### Convertir de entero a ASCII



La operación **Convertir de entero a ASCII** convierte un entero (IN) en una cadena ASCII. El formato (FMT) indica la precisión de la conversión a la derecha del decimal, así como si el punto decimal debe aparecer en forma de coma o de punto. La conversión resultante se deposita en 8 bytes consecutivos comenzando en OUT. La cadena ASCII comprende siempre 8 caracteres.

Condiciones de error que ponen ENO a 0: SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto), sin salida (formato no válido)

Entradas/salidas	Operandos	Tipos de datos
IN	VW, IW, QW, MW, SW, SMW, LW, AIW, T, C, AC, constante, *VD, *AC, *LD	INT
FMT	VB, IB, QB, MB, SMB, LB, AC, constante, *VD, *AC, SB, *LD	BYTE
OUT	VB, IB, QB, MB, SMB, LB, *VD, *AC, SB, *LD	BYTE

El operando de formato (FMT) para la operación ITA (Convertir de entero a ASCII) se define en la figura 9-48. El tamaño del búfer de salida es siempre de 8 bytes. El campo nnn indica el número de dígitos a la derecha del punto decimal en el búfer de salida. El margen válido para el campo nnn está comprendido entre 0 y 5. Si se indican 0 dígitos a la derecha del punto decimal, el valor se visualizará sin punto decimal. Si el valor nnn es mayor que 5, el búfer de salida se llenará con espacios ASCII. El bit c indica si se utiliza una coma (c=1) o un punto decimal (c=0) como separador entre el número entero y la fracción. Los 4 bits superiores deben ser cero.

El búfer de salida se formatea conforme a las siguientes reglas:

1. Los valores positivos se escriben sin signo en el búfer de salida.
2. Los valores negativos se escriben precedidos de un signo menos (-) en el búfer de salida.
3. Los ceros a la izquierda del punto decimal (con excepción del dígito adyacente a dicho punto) se suprimen.
4. Los valores se justifican a la derecha en el búfer de salida.

La figura 9-48 muestra ejemplos de valores que se formatean utilizando un punto decimal (c=0) con tres dígitos a la derecha del punto decimal (nnn=011).

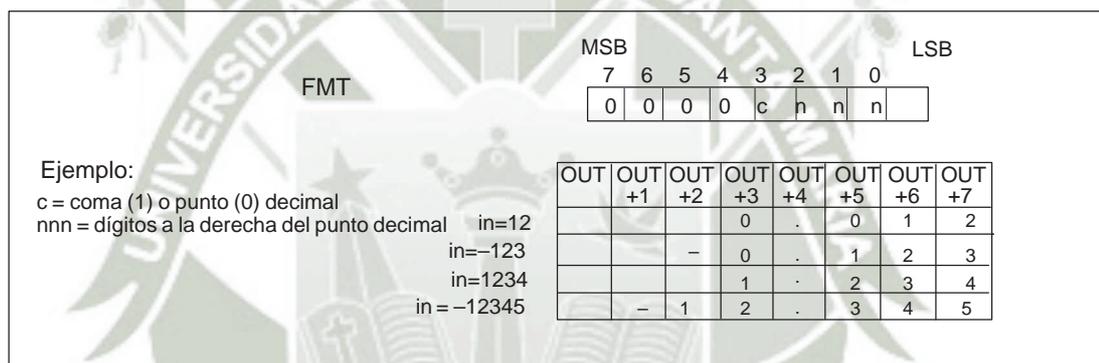
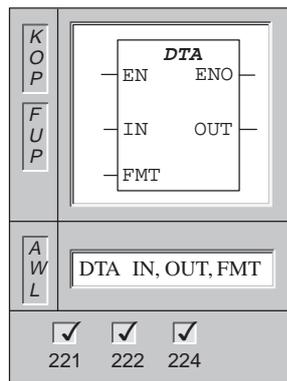


Figura 9-48 Operando FMT para la operación Convertir de entero a ASCII (ITA)

### Convertir de entero doble a ASCII



La operación **Convertir de entero doble a ASCII** convierte una palabra doble (IN) en una cadena ASCII. El formato (FMT) indica la precisión de conversión a la derecha del decimal. La conversión resultante se deposita en 12 bytes consecutivos comenzando en OUT.

Condiciones de error que ponen ENO a 0: SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto), sin salida (formato no válido)

Entradas/salidas	Operandos	Tipos de datos
IN	VD, ID, QD, MD, SD, SMD, LD, HC, constante, AC, *VD, *AC, *LD	DINT
FMT	VB, IB, QB, MB, SMB, LB, AC, constante, *VD, *AC, SB, *LD	BYTE
OUT	VB, IB, QB, MB, SMB, LB, *VD, *AC, SB, *LD	BYTE

El operando de formato (FMT) para la operación DTA se define en la figura 9-49. El tamaño del búfer de salida es siempre de 12 bytes. El campo nnn indica el número de dígitos a la derecha del punto decimal en el búfer de salida. El margen válido para el campo nnn está comprendido entre 0 y 5. Si se indican 0 dígitos a la derecha del punto decimal, el valor se visualizará sin punto decimal. Si el valor nnn es mayor que 5, el búfer de salida se llenará con espacios ASCII. El bit c indica si se utiliza una coma (c=1) o un punto decimal (c=0) como separador entre el número entero y la fracción. Los 4 bits superiores deben ser cero. El búfer de salida se formatea conforme a las siguientes reglas:

1. Los valores positivos se escriben sin signo en el búfer de salida.
2. Los valores negativos se escriben precedidos de un signo menos (–) en el búfer de salida.
3. Los ceros a la izquierda del punto decimal (con excepción del dígito adyacente a dicho punto) se suprimen.
4. Los valores se justifican a la derecha en el búfer de salida.

La figura 9-49 muestra ejemplos de valores que se formatean utilizando un punto decimal (c=0) con cuatro dígitos a la derecha del punto decimal (nnn=100).

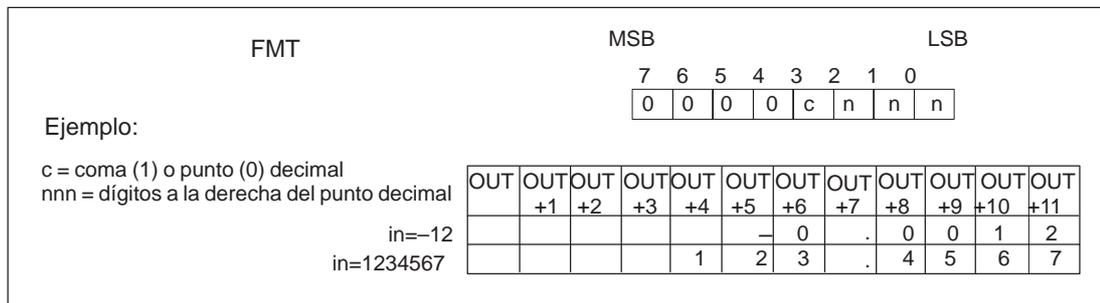
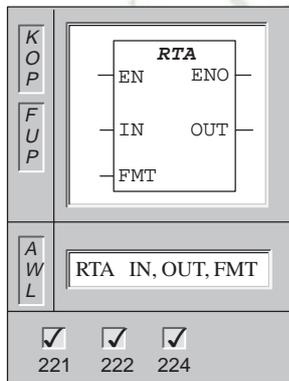


Figura 9-49 Operando FMT para la operación Convertir de entero doble a ASCII

### Convertir de real a ASCII



La operación **Convertir de real a ASCII** convierte el valor en coma flotante (IN) en una cadena ASCII. El formato (FMT) indica la precisión de la conversión a la derecha del decimal, así como si el punto decimal debe aparecer en forma de coma o de punto, y también el tamaño del búfer de salida. La conversión resultante se deposita en un búfer de salida que comienza en OUT. La longitud de la cadena ASCII resultante corresponde al tamaño del búfer de salida, pudiendo indicarse en un margen comprendido entre 3 y 15.

Condiciones de error que ponen ENO a 0: SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto), sin salida (formato no válido o búfer demasiado pequeño)

Entradas/salidas	Operandos	Tipos de datos
IN	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *AC, *LD	REAL
FMT	VB, IB, QB, MB, SMB, LB, AC, constante, *VD, *AC, SB, *LD	BYTE
OUT	VB, IB, QB, MB, SMB, LB, *VD, *AC, SB, *LD	BYTE

El operando de formato (FMT) para la operación RTA se define en la figura 9-50. El campo ssss indica el tamaño del búfer de salida. No es válido un tamaño de 0, 1 ó 2 bytes. El campo nnn indica el número de dígitos a la derecha del punto decimal en el búfer de salida. El margen válido para el campo nnn está comprendido entre 0 y 5. Si se indican 0 dígitos a la derecha del punto decimal, el valor se visualizará sin punto decimal. El búfer de salida se rellena con espacios ASCII si los valores nnn son mayores que 5 o si dicho búfer es demasiado pequeño para almacenar el valor convertido. El bit c indica si se utiliza una coma (c=1) o un punto decimal (c=0) como separador entre el número entero y la fracción. El búfer de salida se formatea conforme a las siguientes reglas:

1. Los valores positivos se escriben sin signo en el búfer de salida.
2. Los valores negativos se escriben precedidos de un signo menos (-) en el búfer de salida.
3. Los ceros a la izquierda del punto decimal (con excepción del dígito adyacente a dicho punto) se suprimen.
4. Los valores a la derecha del punto decimal se redondean para que correspondan al número de dígitos indicado.
5. El búfer de salida deberá ser por lo menos tres bytes más grande que el número de dígitos a la derecha del punto decimal.
6. Los valores se justifican a la derecha en el búfer de salida.

La figura 9-50 muestra ejemplos de valores que se formatean utilizando un punto decimal (c=0) con un dígito a la derecha del punto decimal (nnn=001) y un tamaño de búfer de seis bytes (ssss=0110).

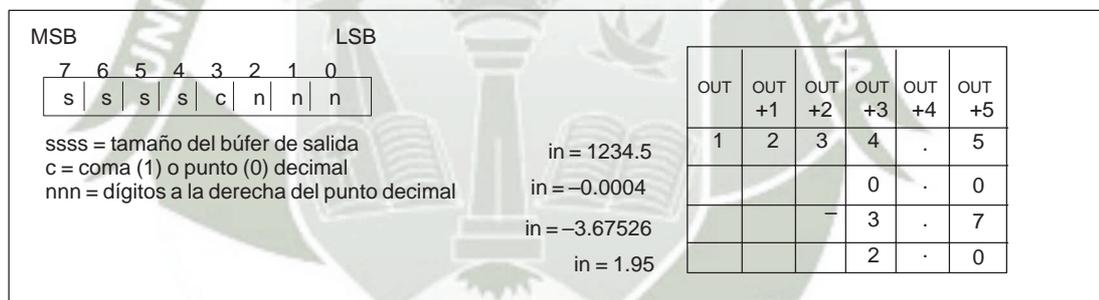


Figura 9-50 Operando FMT para la operación Convertir de real a ASCII

**Nota**

El formato en coma flotante utilizado por la CPU S7-200 asiste 7 dígitos significativos como máximo. Si se intenta visualizar más de dichos 7 dígitos significativos se producirá un error de redondeo.

## 9.15 Operaciones de control del programa (SIMATIC)

### END

K O P	—(END)
F U P	—[END]
A W L	END
<input checked="" type="checkbox"/> 221	<input checked="" type="checkbox"/> 222
<input checked="" type="checkbox"/> 224	

La operación condicional **Finalizar programa principal** finaliza el programa en función de la combinación lógica precedente.

Operandos: ninguno

Tipos de datos: ninguno

#### Nota

La operación END condicional se puede utilizar en el programa principal, pero no en subrutinas ni en rutinas de interrupción.

#### Nota

Micro/WIN 32 añade automáticamente un fin absoluto al programa principal de usuario.

### STOP

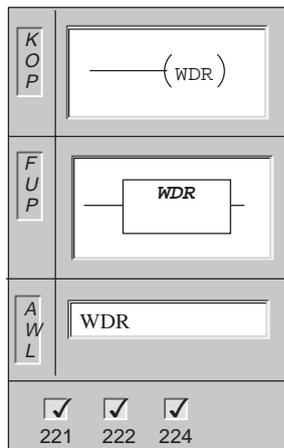
K O P	—(STOP)
F U P	—[STOP]
A W L	STOP
<input checked="" type="checkbox"/> 221	<input checked="" type="checkbox"/> 222
<input checked="" type="checkbox"/> 224	

La operación **STOP** finaliza inmediatamente la ejecución del programa haciendo que la CPU cambie de RUN a STOP.

Operandos: ninguno

Si la operación STOP se ejecuta en una rutina de interrupción, ésta se finalizará inmediatamente ignorando las interrupciones pendientes. Las demás acciones en el ciclo actual se completan, incluyendo la ejecución del programa principal. El cambio de RUN a STOP se produce al final del ciclo actual.

### Borrar temporizador de vigilancia



La operación **Borrar temporizador de vigilancia** permite que la CPU redispere el temporizador de vigilancia. Así se prolonga el tiempo de ciclo sin que se indique un error de vigilancia.

Operandos: ninguno

### Utilizar la operación WDR para inicializar el temporizador de vigilancia

Esta operación se debe utilizar con mucha cautela. En caso de utilizar bucles para que no finalice el ciclo o para prolongarlo excesivamente, es posible que no se ejecuten los procesos siguientes hasta completar el ciclo:

- Comunicación (excepto modo Freeport)
- Actualización de las entradas y salidas (excepto control directo de las E/S)
- Actualización de los valores forzados
- Actualización de las marcas especiales (no se actualizan las marcas SM0 y SM5 a SM29)
- Tareas de diagnóstico en el tiempo de ejecución
- Los temporizadores con resolución de 10 ms y 100 ms no contarán correctamente los ciclos que excedan los 25 segundos.
- Operación STOP si se utiliza en una rutina de interrupción

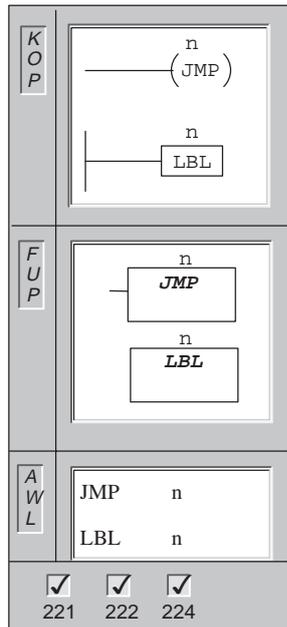
#### Nota

Si se prevé que el tiempo de ciclo durará más de 300 ms o que la actividad de interrupción aumentará de modo que el ciclo principal quede interrumpido más de 300 ms, es preciso utilizar la operación WDR para redispere el temporizador de vigilancia.

Cambiando el selector a la posición STOP, la CPU pasará a modo STOP en 1,4 segundos.



### Saltar a meta, Definir meta



La operación **Saltar a meta** deriva la ejecución del programa a la meta indicada (n). Al saltar, el primer valor de la pila es siempre un "1" lógico.

La operación **Definir meta** indica la dirección de la meta de salto (n).

Operandos: n: 0 a 255

Tipos de datos: WORD

Tanto la operación de salto como la correspondiente meta deben encontrarse en el programa principal, en una subrutina o en una rutina de interrupción. Desde el programa principal no se puede saltar a una meta que se encuentre en una subrutina o en una rutina de interrupción. Tampoco es posible saltar desde una subrutina o una rutina de interrupción a una meta que se encuentre fuera de ella.

### Ejemplo de la operación Saltar a meta

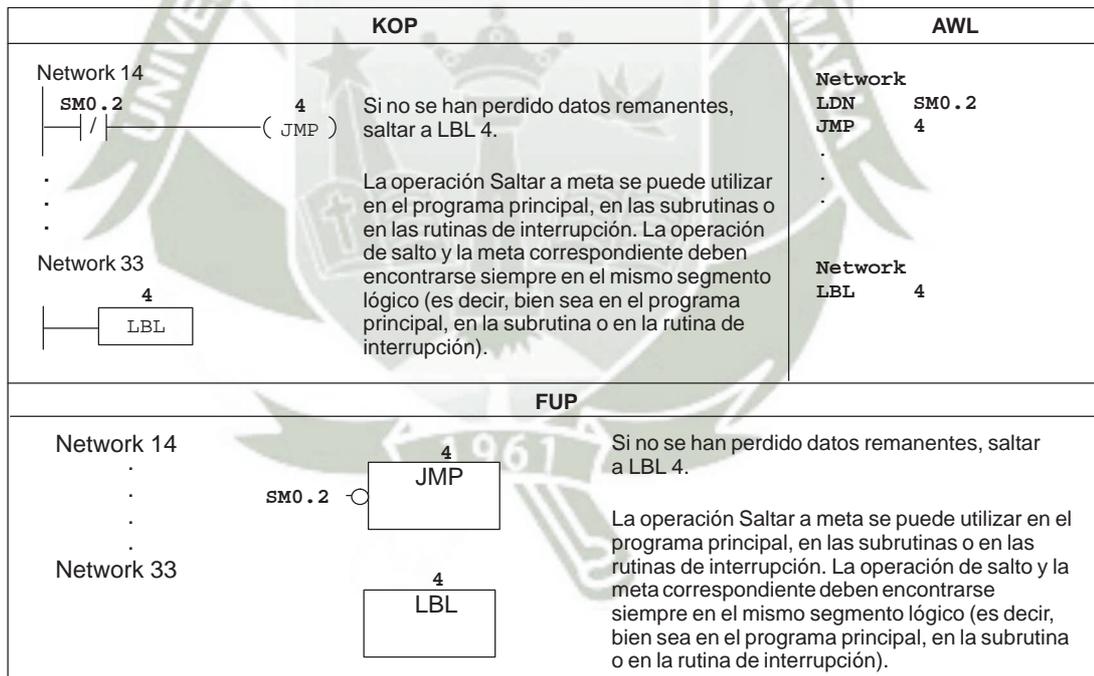


Figura 9-52 Ejemplo de las operaciones Saltar a meta y Definir meta en KOP, AWL y FUP

## Llamar subrutina, Retorno de subrutina

K O P	
F U P	
A W L	
	<input checked="" type="checkbox"/> 221 <input checked="" type="checkbox"/> 222 <input checked="" type="checkbox"/> 224

La operación **Llamar subrutina** transfiere el control a la subrutina (n). Dicha operación se puede utilizar con o sin parámetros. Para añadir una subrutina, elija los comandos de menú **Edición > Insertar > Subrutina**.

La operación **Retorno condicional de subrutina** se utiliza para finalizar una subrutina en función de la combinación lógica precedente.

Operandos:            n:            constante

Tipos de datos:      BYTE

Una vez ejecutada la subrutina, el control vuelve a la operación que sigue a la llamada de la subrutina (CALL).

La figura 9-55 muestra ejemplos de las operaciones Llamar subrutina y Retorno de subrutina.

Condiciones de error que ponen ENO a 0 para la llamada de subrutina con parámetros:

SM4.3 (tiempo de ejecución), 0008 (excedida la profundidad máxima de anidamiento)

### Nota

Micro/WIN 32 añade automáticamente un retorno desde cada subrutina.

Se pueden anidar (situar una llamada a subrutina en otra) hasta ocho subrutinas (profundidad de anidamiento = 8 niveles). Si bien la recursión (la subrutina se llama a sí misma) está permitida, hay que utilizarla con cautela.

Cuando se llama a una subrutina, se almacena toda la pila lógica, poniéndose a "1" el nivel superior de la pila. Sus demás niveles se ponen a "0" y la ejecución se transfiere a la subrutina que se ha llamado. Cuando ésta se termina de ejecutar, se restablece la pila con los valores almacenados al llamar a la subrutina y se retorna a la rutina que ha efectuado la llamada.

Los acumuladores son comunes a las subrutinas y a la rutina de llamada. Los acumuladores no se almacenan ni se restablecen si se utilizan con subrutinas.

## Llamar a una subrutina con parámetros

Las subrutinas pueden contener parámetros que hayan sido transferidos. Los parámetros se definen en la tabla de variables locales de la subrutina (v. figura 9-53). Dichos parámetros deben tener un nombre simbólico (de 8 caracteres como máximo), un tipo de variable y un tipo de datos. Se pueden transferir 16 parámetros a o desde una subrutina.

El campo "Tipo de variable" en la tabla de variables locales define si la variable se transfiere a la subrutina (IN), a y desde la subrutina (IN\_OUT) o desde la subrutina (OUT). A continuación se indican las características de los tipos de parámetros:

- IN: los parámetros se transfieren a la subrutina. Si el parámetro es una dirección directa (p.ej. VB10), el valor de la dirección indicada se transferirá a la subrutina. Si el parámetro es una dirección indirecta (p.ej. \*AC1), el valor de la dirección a la que señala el puntero se transferirá a la subrutina. Si el parámetro es una constante de datos (16#1234) o una dirección (VB100), la constante o el valor de dirección se transferirán a la subrutina.
- IN\_OUT: el valor de la dirección del parámetro indicado se transfiere a la subrutina y el valor resultante de la subrutina se retorna luego a la misma dirección. Para los parámetros de entrada/salida no se pueden utilizar ni constantes (p.ej. 16#1234) ni direcciones (p.ej. &VB100).
- OUT: El valor resultante de la subrutina se retorna a la dirección del parámetro indicado. Para los parámetros de salida no se pueden utilizar ni constantes (p.ej. 16#1234) ni direcciones (p.ej. &VB100).
- TEMP:  
 Cualquier memoria local que no se utilice para el paso de parámetros se puede emplear para el almacenamiento temporal dentro de la subrutina.

Para añadir un registro de parámetro, en el campo "Tipo de variable" sitúe el cursor en el tipo (IN, IN\_OUT<OUT) que desea añadir. Pulse el botón derecho del ratón para visualizar un menú que ofrece diversas opciones. Seleccione la opción "Insertar" y luego la opción "Fila inferior". Debajo del registro actual aparecerá un nuevo registro de parámetro.

	Nombre	Tipo var.	Tipo de datos	Comentario
	EN	IN	BOOL	
L0.0	IN1	IN	BOOL	
LB1	IN2	IN	BYTE	
LB2.0	IN3	IN	BOOL	
LD3	IN4	IN	DWORD	
LW7	IN/OUT1	IN/OUT	WORD	
LD9	OUT1	OUT	DWORD	
		TEMP		

Figura 9-53 Tabla de variables locales de STEP 7-Micro/WIN 32

El campo "Tipo de datos" de la tabla de variables locales define el tamaño y el formato del parámetro. Los tipos de parámetros son:

- **POWER FLOW:** La circulación de corriente booleana sólo se permite en las entradas binarias (booleanas). Dicha declaración le indica a STEP 7-Micro/WIN 32 que este tipo de parámetro de entrada es el resultado de la circulación de la corriente conforme a una combinación de operaciones lógicas con bits. Las entradas booleanas de circulación de corriente deben aparecer en la tabla de variables locales antes de cualquier otro tipo de entrada. Aquí se permite utilizar sólo parámetros de entrada. La entrada de habilitación (EN) y las entradas IN1 en la figura 9-54 usan la lógica booleana.
- **BOOL:** Este tipo de datos se utiliza para entradas y salidas binarias sencillas. IN2 en la figura 9-54 es una entrada booleana.
- **BYTE, WORD, DWORD:** Estos tipos de datos identifican parámetros de entrada o de salida sin signo compuestos por 1, 2 ó 4 bytes, respectivamente.
- **INT, DINT:** Estos tipos de datos identifican parámetros de entrada o de salida con signo compuestos por 2 ó 4 bytes, respectivamente.
- **REAL:** Este tipo de datos identifica un valor en coma flotante IEEE de precisión simple (4 bytes).

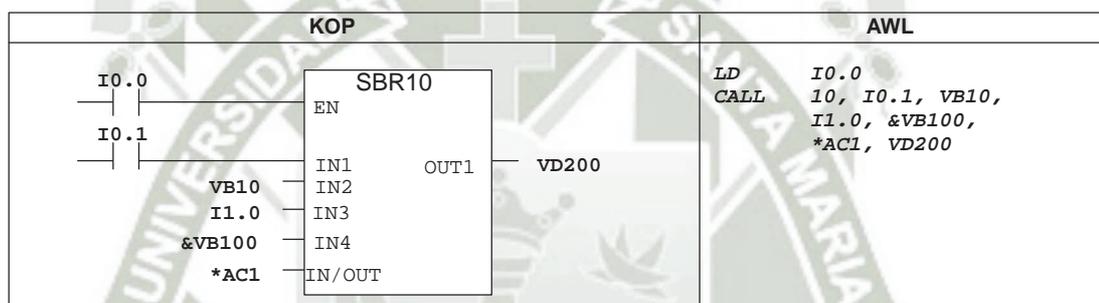


Figura 9-54 Llamada a subrutina en KOP y AWL

Los parámetros de dirección tales como IN4 en la figura 9-54 (&VB100) se transfieren a una subrutina como valor DWORD (palabra doble sin signo). El tipo de parámetro de una constante se debe indicar en la rutina de llamada mediante un descriptor delante del valor de la constante. Por ejemplo, para transferir como parámetro una constante de palabra doble sin signo cuyo valor sea 12.345, el parámetro de dicha constante se deberá indicar de la siguiente forma: DW#12345. Si se omite el descriptor de la constante, se podría deducir que la constante es de un tipo diferente.

En el caso de los parámetros de entrada o de salida no se realiza una conversión automática de datos. Por ejemplo, si en la tabla de variables locales se indica que un parámetro es del tipo de datos REAL y en la rutina de llamada se indica una palabra doble (DWORD) para dicho parámetro, el valor en la subrutina será una palabra doble.

Los valores que se transfieren a una subrutina se depositan en la memoria local de la misma. La columna del extremo izquierdo de la tabla de variables locales (v. figura 9-53) muestra la dirección local de cada parámetro que se ha transferido. Cuando se llama a la subrutina, los valores de los parámetros de entrada se copian a la memoria local de la subrutina. Cuando se finaliza la ejecución de la subrutina, los valores de los parámetros de salida se copian de la memoria local de la subrutina a las direcciones indicadas de los parámetros de salida.

ntamaño y el tipo de los elementos de datos está codificado en los parámetros. Los valores de los parámetros se asignan de la siguiente forma a la memoria local de la subrutina:

- Los valores de parámetros se asignan a la memoria local en el orden indicado por la operación Llamar subrutina, comenzando dichos parámetros en L.0.
- Uno a ocho valores binarios de parámetros consecutivos se asignan a un sólo byte comenzando en Lx.0 hasta Lx.7.
- Los valores de byte, palabra y palabra doble se asignan a la memoria local en bytes (LBx, LWx ó LDx).

En la operación Llamar subrutina con parámetros, éstos se deben organizar de la siguiente forma: primero los parámetros de entrada, luego los de entrada/salida y, por último, los de salida.

En AWL, el formato de la operación Llamar subrutina (CALL) es el siguiente:

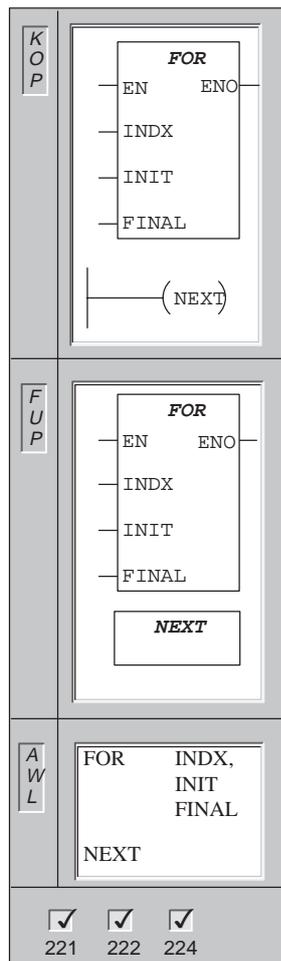
CALL    número de subrutina, parámetro 1, parámetro 2, ... , parámetro

Condiciones de error que ponen ENO a 0 para la llamada de una subrutina con parámetros: SM4.3 (tiempo de ejecución), 0008 (excedida la profundidad máxima de anidamiento)





## FOR, NEXT



La operación **FOR** ejecuta las operaciones que se encuentren entre FOR y NEXT. Se deben indicar el valor del índice o el conteo actual del bucle (INDX), el valor inicial (INIT) y el valor final (FINAL).

La operación **NEXT** marca el final del bucle FOR y pone a "1" el primer valor de la pila.

Ejemplo: si el valor de INIT es 1 y si el de FINAL es 10, las operaciones que se encuentren entre FOR y NEXT se ejecutarán 10 veces, incrementando el valor de conteo INDX en 1, 2, 3, ...10.

Si el valor inicial es mayor que el valor final, no se ejecuta el bucle. Después de ejecutarse las operaciones que se encuentran entre FOR y NEXT, se incrementa el valor de INDX y el resultado se compara con el valor final. Si INDX es mayor que el valor final, finaliza el bucle.

**FOR:** Condiciones de error que ponen ENO a 0: SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Entradas/salidas	Operandos	Tipos de datos
INDX	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, *VD, *AC, *LD	INT
INIT	VW, IW, QW, MW, SW, SMW, T, C, AC, LW, AIW, constante, *VD, *AC, *LD	INT
FINAL	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, AIW, constante, *VD, *AC, *LD	INT

Reglas para utilizar el bucle FOR/NEXT:

- Al habilitar el bucle FOR/NEXT, éste se ejecuta hasta finalizar las iteraciones, a menos que Ud. cambie el valor final dentro del bucle. Los valores se pueden cambiar mientras se ejecute FOR/NEXT.
- Si se vuelve a habilitar el bucle, éste copia el valor inicial (INIT) en el valor actual de conteo del bucle (IDX). La operación FOR/NEXT se desactiva automáticamente la próxima vez que se habilite.

Las operaciones FOR/NEXT repiten un bucle del programa un número determinado de veces. Toda operación FOR exige una operación NEXT. Los bucles FOR/NEXT pueden anidarse (insertar un bucle FOR/NEXT dentro de otro) hasta una profundidad de ocho niveles.



### Ejemplo de las operaciones FOR/NEXT

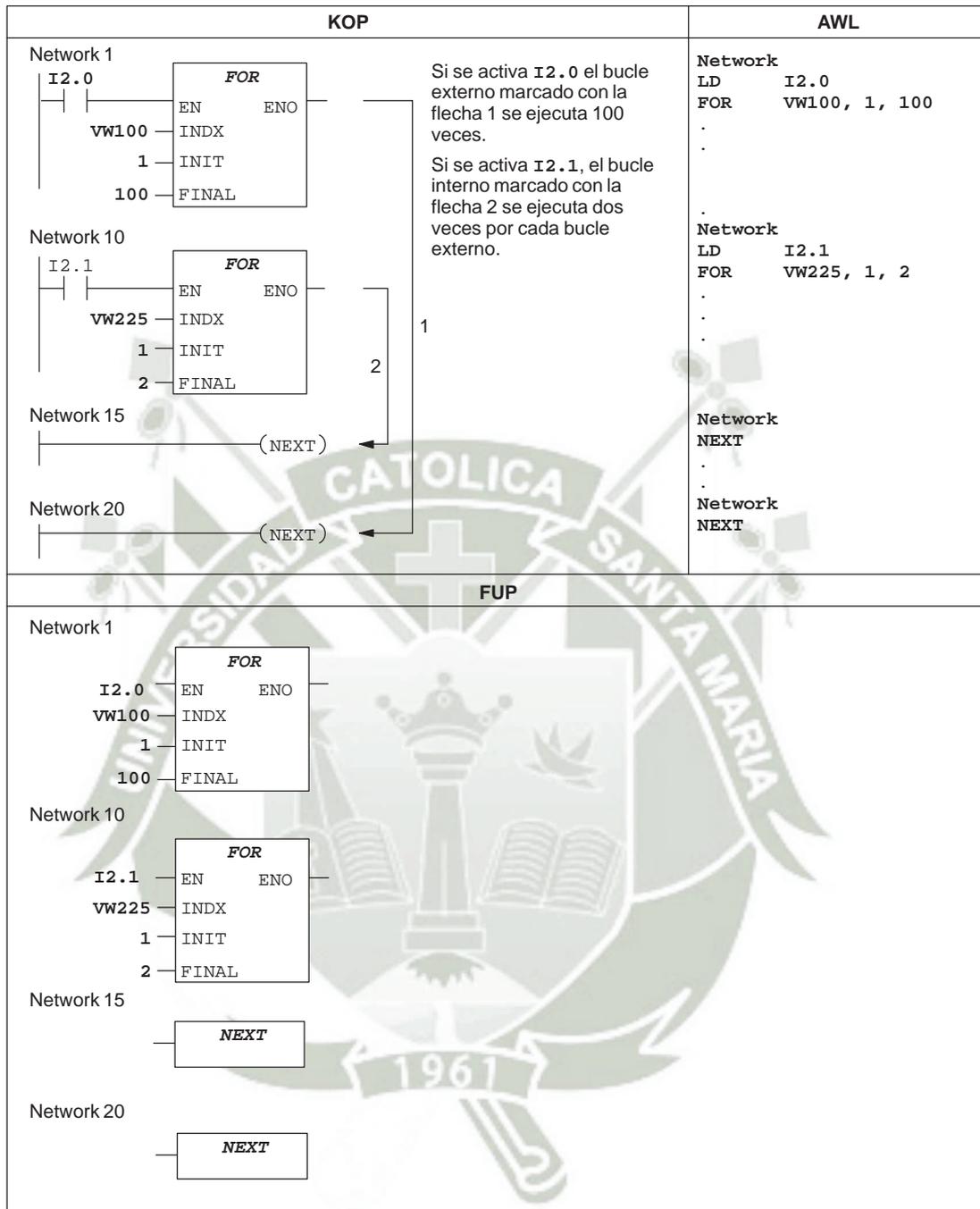
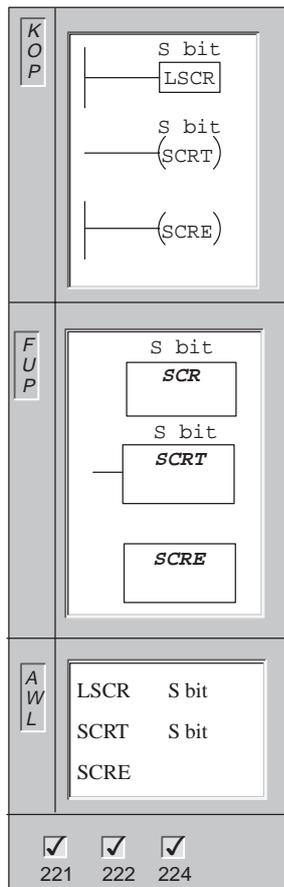


Figura 9-56 Ejemplo de las operaciones FOR/NEXT en KOP, AWL y FUP

### Relé de control secuencial



La operación **Cargar relé de control secuencial** indica el comienzo de un segmento SCR. Si el bit S está activado se habilita la circulación de la corriente hasta el segmento SCR. La operación LSCR se debe finalizar con una operación SCRE.

La operación **Transición del relé de control secuencial** identifica el bit SCR que se debe habilitar (el siguiente bit S a activar). Cuando la corriente fluye hasta la bobina o hasta el cuadro FUP, el bit S direccionado se activa y el bit S de la operación LSCR (que habilitó este segmento SCR) se desactiva.

La operación **Fin del relé de control secuencial** indica el fin de un segmento SCR.

Entradas/salidas	Operandos	Tipos de datos
n	S	BOOL

### Descripción de las operaciones del relé de control secuencial

En KOP y AWL, los relés de control secuencial (SCRs) se utilizan para estructurar instalaciones o etapas en segmentos equivalentes del programa. Los SCR permiten segmentar lógicamente el programa de usuario.

La operación LSCR carga el valor del bit S que indica la operación en la pila del relé de control secuencial (pila SCR) así como en la pila lógica. El segmento SCR se activa o se desactiva en función del resultado de la pila SCR. El valor superior de la pila se carga en el bit S indicado, pudiéndose conectar directamente los cuadros y las bobinas a la barra de alimentación izquierda sin necesidad de interconectar un contacto. La figura 9-57 muestra la pila SCR y la pila lógica, así como los efectos de la operación LSCR.

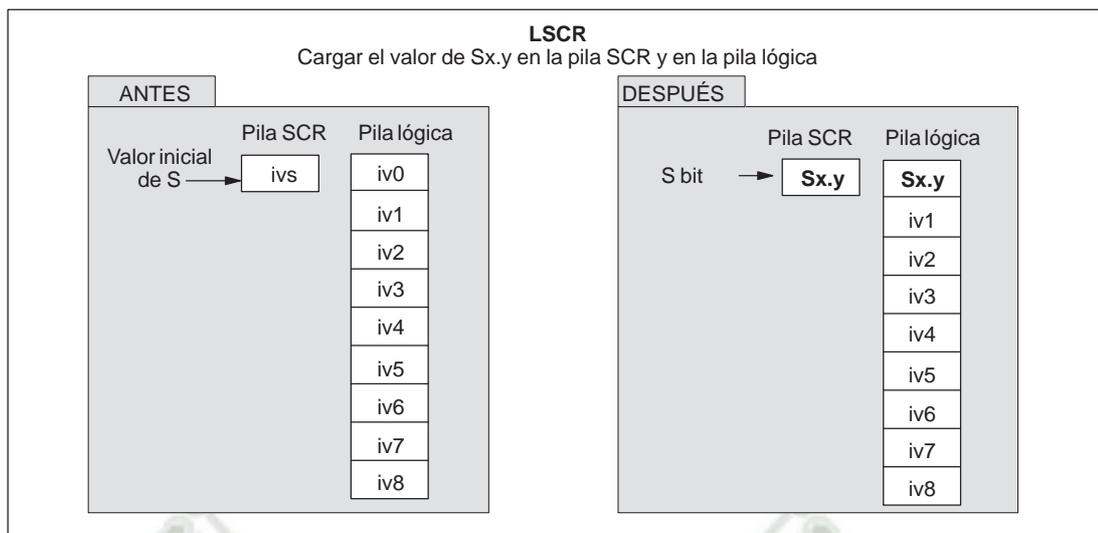


Figura 9-57 Efectos de la operación LSCR en la pila lógica

Observaciones en relación con las operaciones del relé de control secuencial:

- Todas las operaciones que se encuentren entre la operación LSCR y la operación SCRE constituyen el segmento SCR, dependiendo su ejecución del valor de la pila SCR. La lógica que se encuentra entre la operación SCRE y la siguiente operación LSCR no depende del valor de la pila SCR.
- La operación SCRT activa un bit S que habilita el siguiente relé de control secuencial. Asimismo, desactiva el bit S que se cargó para habilitar esta parte del segmento SCR.

### Uso restringido

Al utilizar los relés de control secuencial es preciso tener en cuenta los siguientes puntos:

- Un mismo bit S no se puede utilizar en más de una rutina. Por ejemplo, si S0.1 se usa en el programa principal, no se podrá emplear además en la subrutina.
- En un segmento SCR no se pueden usar las operaciones Saltar a meta (JMP) ni Definir meta (LBL). Por tanto, no se pueden utilizar para saltar dentro ni fuera del segmento SCR, ni tampoco en el mismo. No obstante, las operaciones de salto y de meta se pueden emplear para saltar segmentos SCR.
- En un segmento SCR no se pueden utilizar las operaciones FOR, NEXT ni END.

### Ejemplo de una operación SCR

La figura 9-58 muestra cómo funciona un relé de control secuencial.

- En el ejemplo, S0.1 se activa con la marca especial SM0.1 (marca del primer ciclo). S0.1 será entonces la etapa 1 activa en el primer ciclo.
- Una vez transcurrido un retardo de 2 segundos, T37 provoca una transición a la etapa 2. Esta transición desactiva el segmento SCR (S0.1) de la primera etapa y activa el segmento SCR (S0.2) de la segunda etapa.

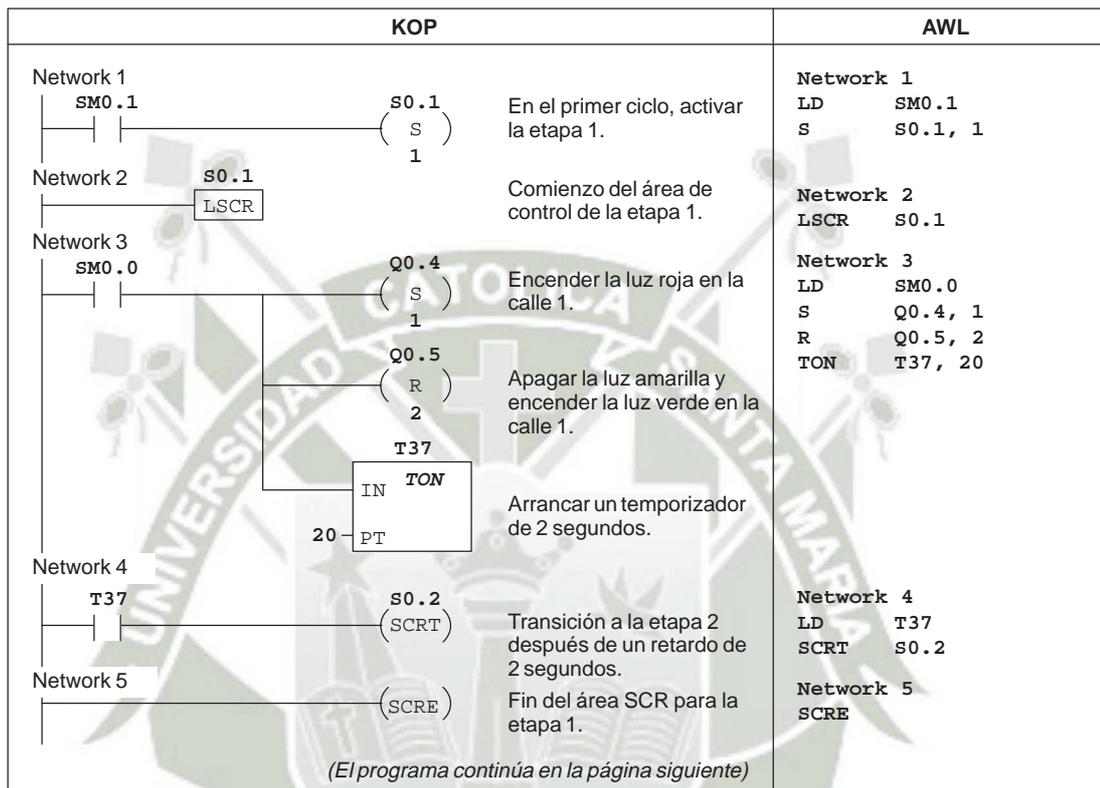


Figura 9-58 Ejemplo de un relé de control secuencial (SCR)

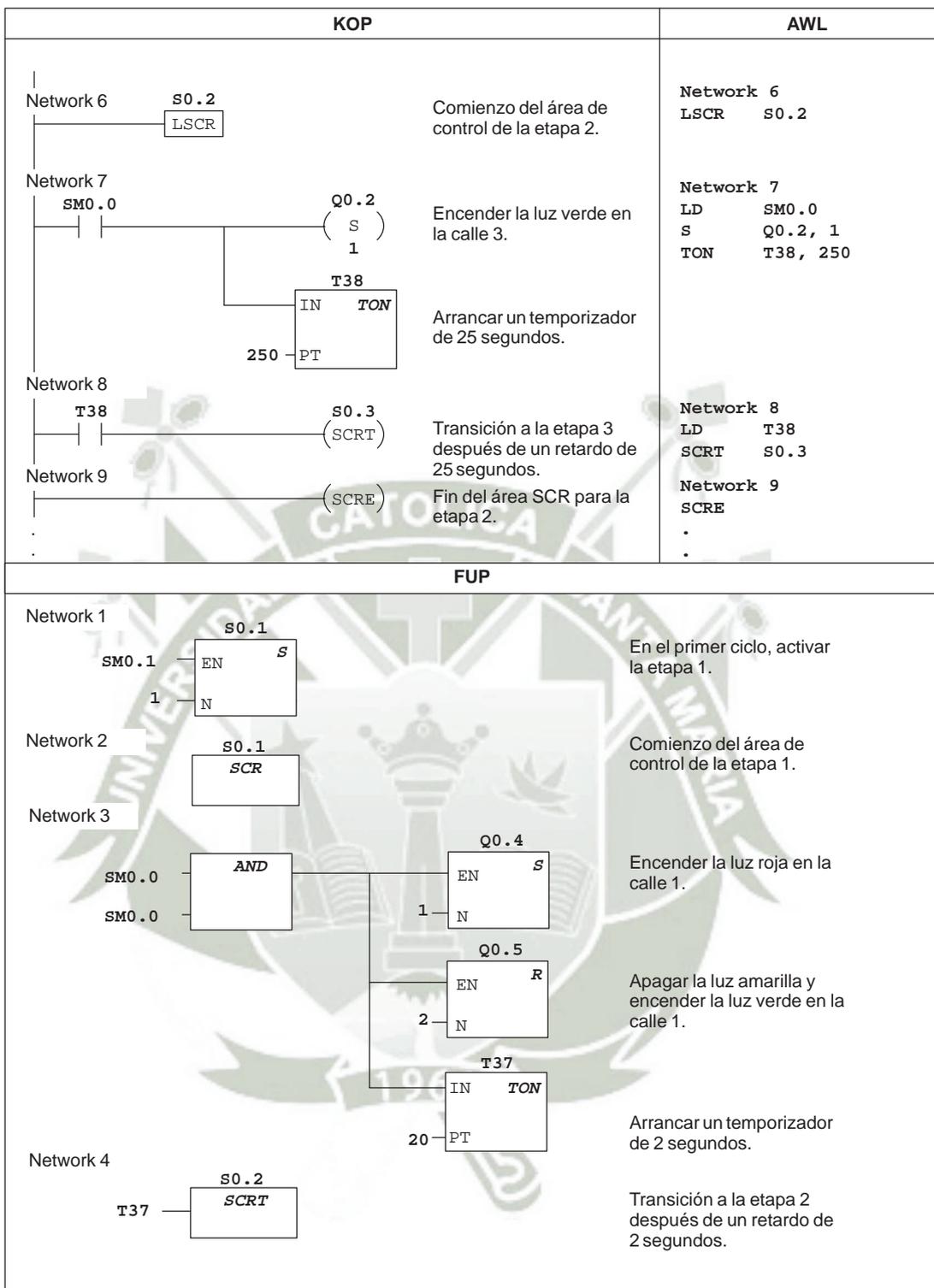


Figura 9-58 Ejemplo de un relé de control secuencial (SCR), (continuación)

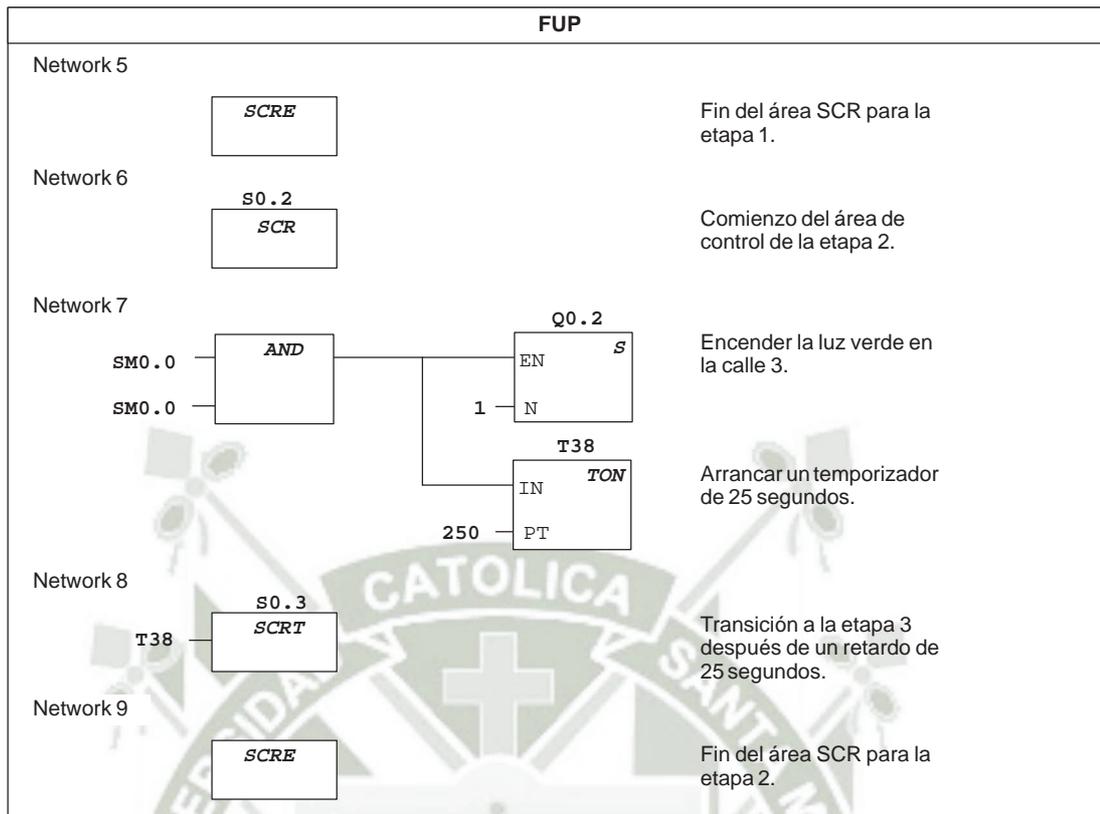


Figura 9-58 Ejemplo de un relé de control secuencial (SCR), (continuación)

### Dividir cadenas secuenciales

En numerosas aplicaciones es necesario dividir una cadena secuencial en dos o más cadenas. Si una cadena secuencial se divide en varias cadenas, es preciso activar simultáneamente todas las nuevas cadenas secuenciales, como muestra la figura 9-59.

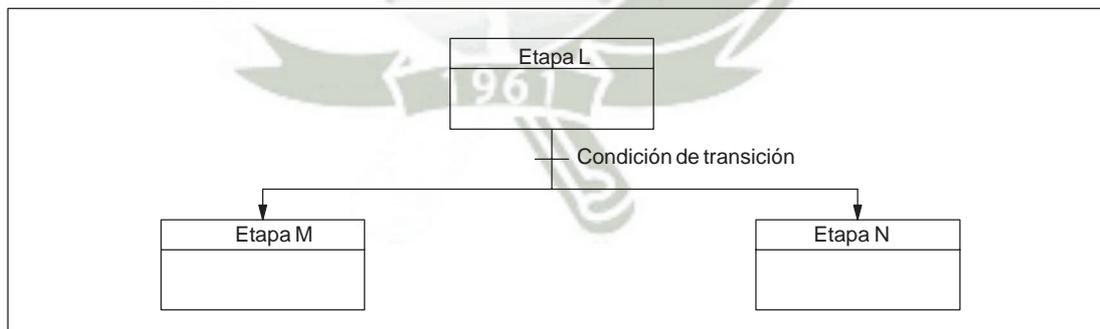


Figura 9-59 División de una cadena secuencial

La división de cadenas secuenciales se puede implementar en un programa SCR, activando varias operaciones SCRT con una misma condición de transición como muestra la figura 9-60.

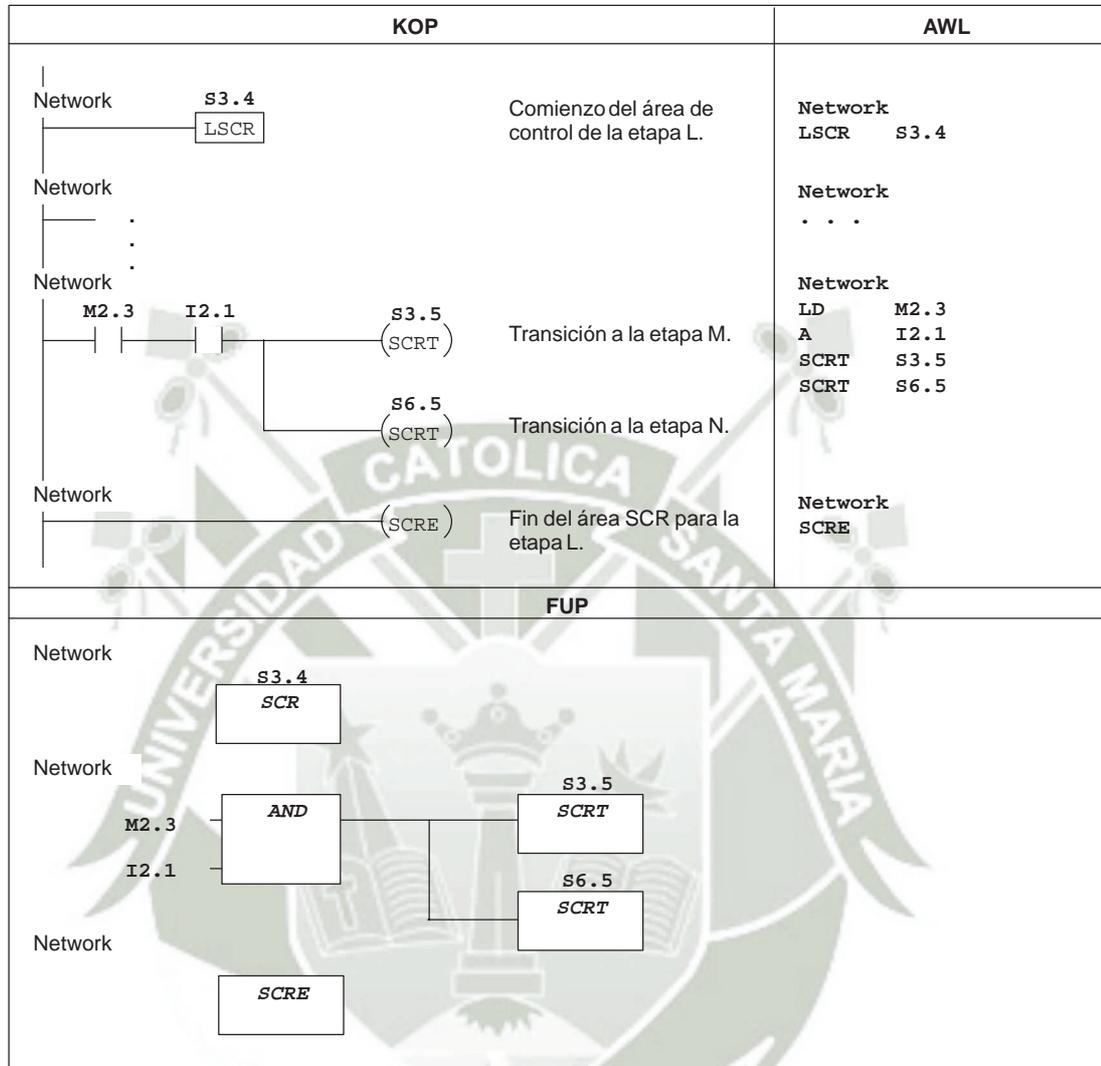


Figura 9-60 Ejemplo de la división de una cadena secuencial

### Convergir cadenas secuenciales

Algo similar ocurre cuando dos o más cadenas secuenciales deban convergir en una sola. Todas las cadenas secuenciales se deben terminar antes de poder ejecutar la siguiente etapa. La figura 9-61 muestra la convergencia de dos cadenas secuenciales.

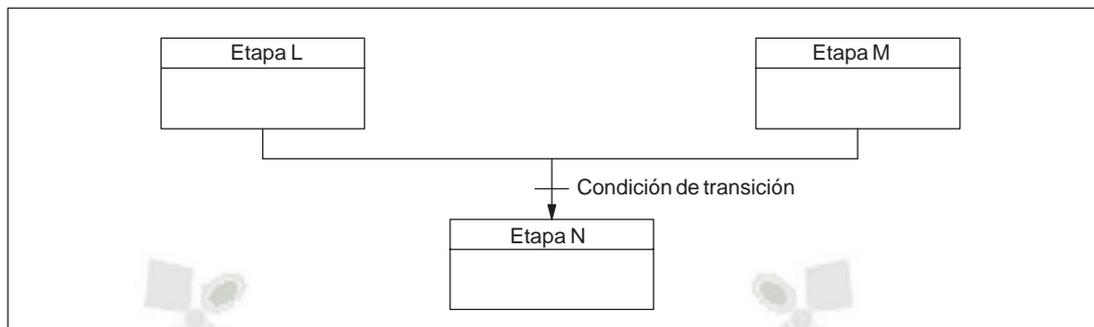


Figura 9-61 Convergencia de cadenas secuenciales



La convergencia de cadenas secuenciales se puede implementar en un programa SCR creando una transición de la etapa L a la etapa L', y de la etapa M a la etapa M'. Si los bits SCR que representan L' y M' son verdaderos, se podrá habilitar la etapa N como muestra la figura 9-62.

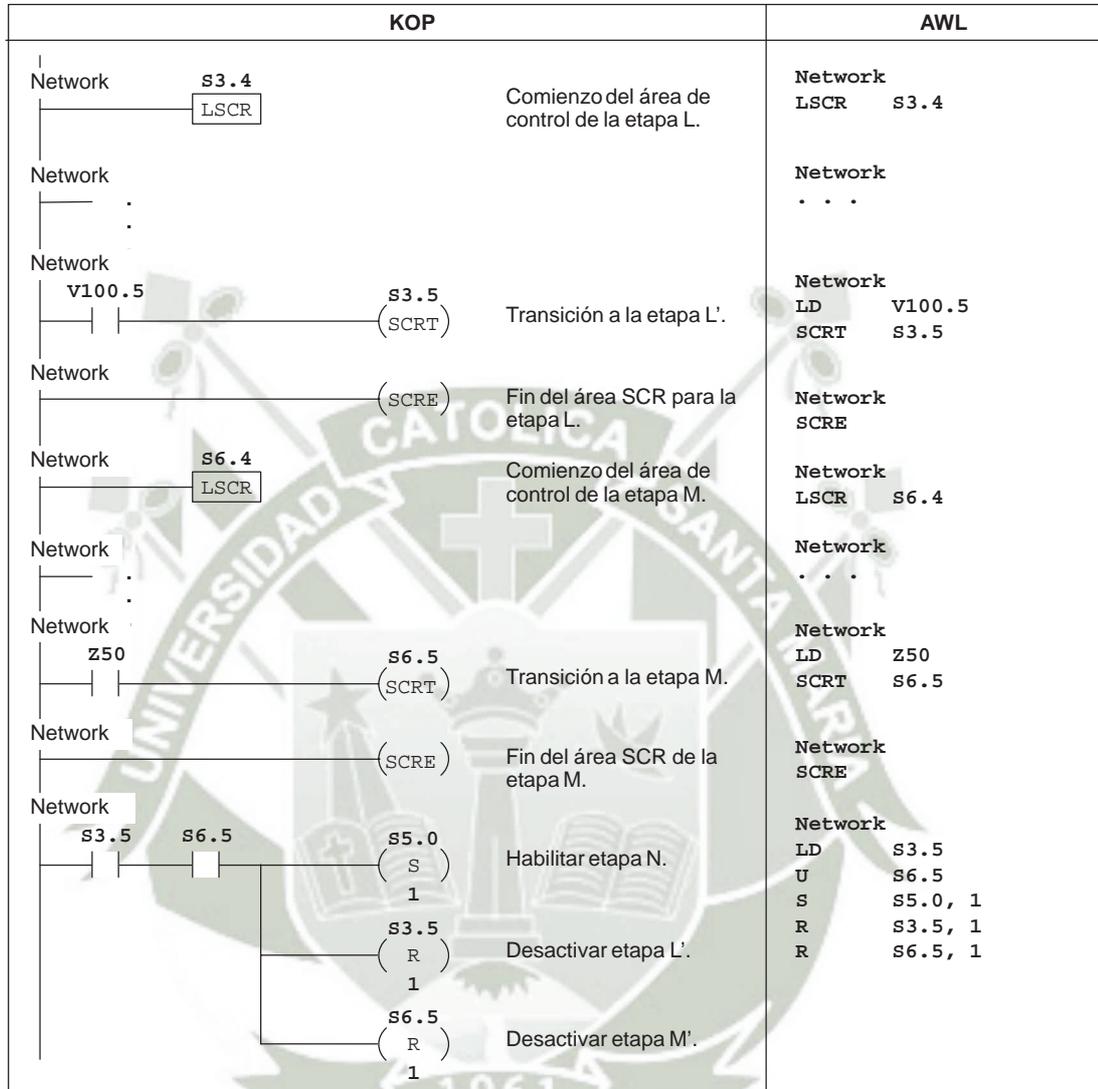


Figura 9-62 Ejemplo de convergencia de cadenas secuenciales

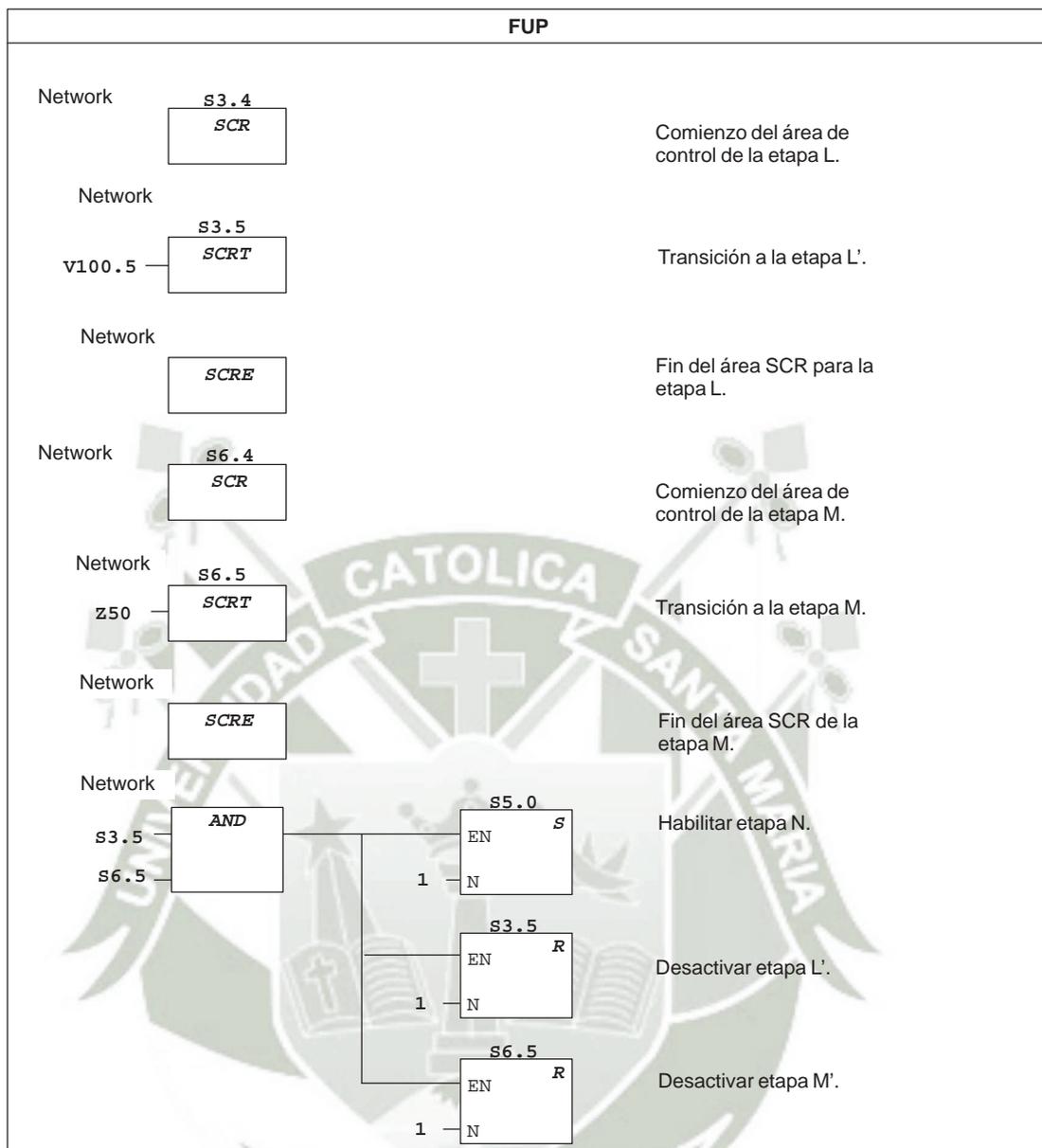


Figura 9-62 Ejemplo de convergencia de dos cadenas secuenciales, continuación

En otras situaciones, una cadena secuencial se puede dirigir a una de varias cadenas secuenciales posibles, dependiendo de la primera condición de transición que sea verdadera. La figura 9-63 muestra dicha situación.

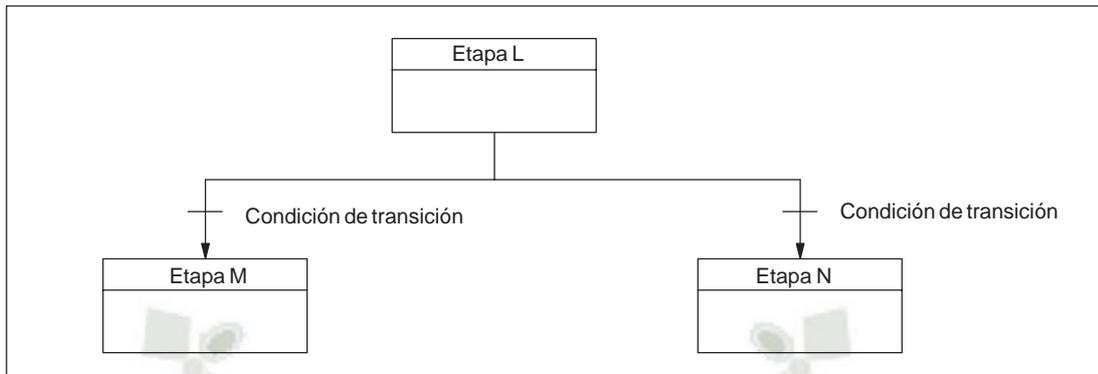


Figura 9-63 Dirigir una cadena secuencial a otra, dependiendo de la condición de transición

La figura 9-64 muestra el correspondiente programa SCR.

KOP		AWL
Network	S3.4 LSCR	Network LSCR S3.4
Network		Network ...
Network	M2.3	Network LD M2.3 SCRT S3.5
Network	S3.5 (SCRT)	
Network	I3.3	Network LD I3.3 SCRT S6.5
Network	S6.5 (SCRT)	
Network	(SCRE)	Network SCRE

Figura 9-64 Ejemplo de transiciones condicionales

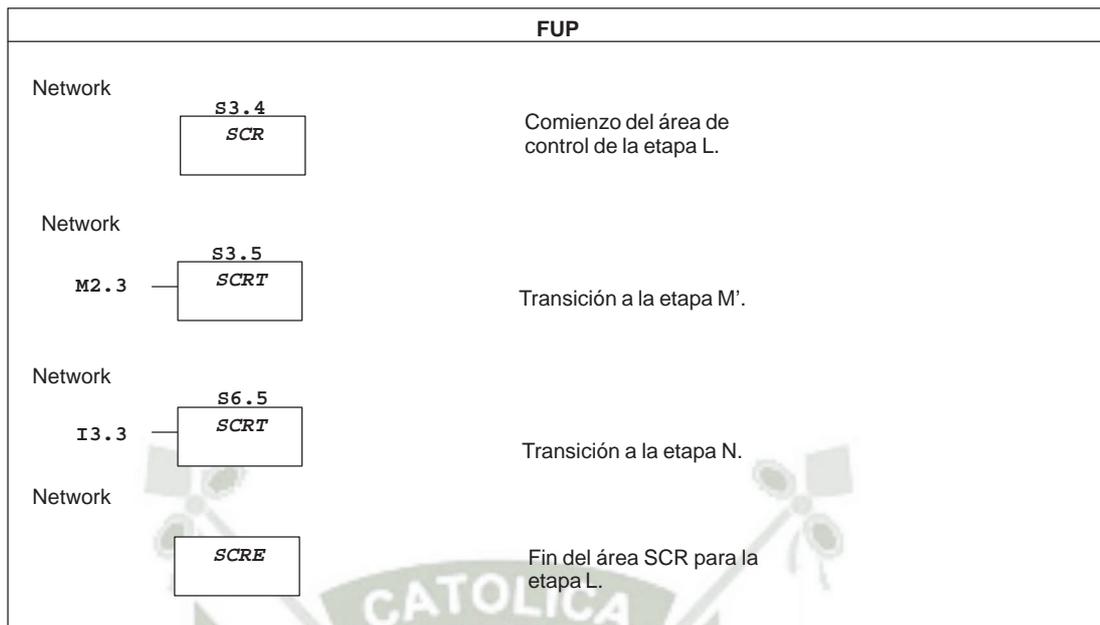
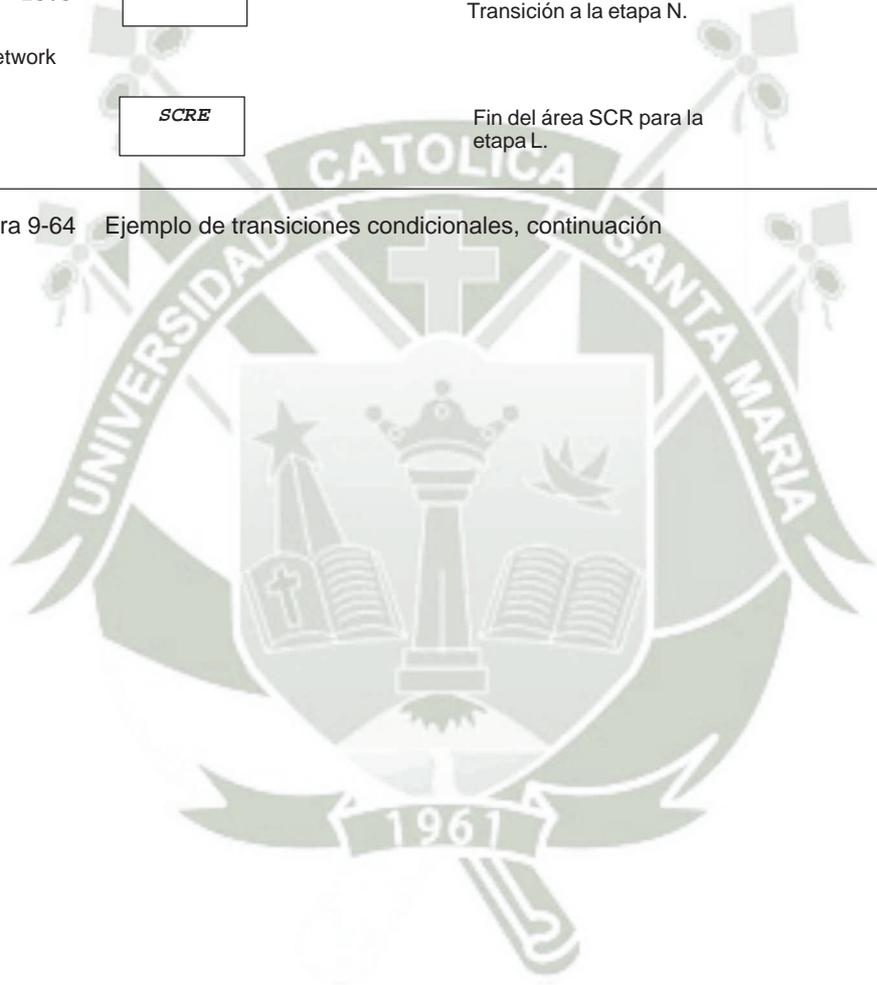
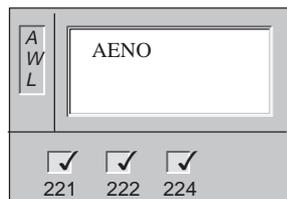


Figura 9-64 Ejemplo de transiciones condicionales, continuación



## ENO



**ENO** es una salida booleana para los cuadros KOP y FUP. Si la corriente fluye en un cuadro por la entrada EN y el cuadro se ejecuta sin error, la salida ENO conduce la corriente al siguiente elemento. ENO se puede utilizar como bit de habilitación para indicar que una operación se ha ejecutado correctamente.

El bit ENO se utiliza en el primer nivel de la pila para influir en la circulación de la corriente al ejecutar las operaciones posteriores.

Las operaciones AWL no tienen una entrada de habilitación (EN). El primer nivel de la pila debe ser un 1 lógico para que la operación se pueda ejecutar.

En AWL no existe la salida de habilitación (ENO), pero las instrucciones AWL correspondientes a las operaciones KOP y FUP con salidas ENO activan un bit ENO especial. A dicho bit se puede acceder con la operación **Y-ENO** (AENO). AENO se puede utilizar para generar el mismo efecto que el bit ENO de un cuadro. La operación AENO sólo está disponible en AWL.

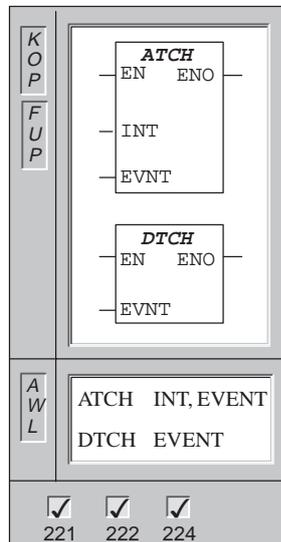
AENO combina el bit ENO y el nivel superior de la pila mediante Y. El resultado de la operación de combinación mediante Y es el nuevo valor en el nivel superior de la pila.

Operandos: ninguno

Tipos de datos: ninguno

## 9.16 Operaciones de interrupción y comunicación (SIMATIC)

### Asociar interrupción, Desasociar interrupción



La operación **Asociar interrupción** asocia el número de una rutina de interrupción (INT) a un evento de interrupción (EVNT), habilitando así éste último.

La operación **Desasociar interrupción** desasocia un evento de interrupción (EVNT) de todas las rutinas de interrupción, deshabilitando así el evento.

**Asociar interrupción:** Condiciones de error que ponen ENO a 0: SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Entradas/salidas	Operandos	Tipos de datos
INT	Constante (CPU 222: 0-12, 19-23, 27-33; CPU 224: 0-23, 27-33)	BYTE
EVNT	Constante (CPU 222: 0-12, 19-23, 27-33; CPU 224: 0-23, 27-33)	BYTE

### Descripción de las operaciones Asociar interrupción y Desasociar interrupción

Antes de poder llamar a una rutina de interrupción es preciso establecer un enlace entre el evento de interrupción y la parte del programa que se desee ejecutar cuando se presente el evento. La operación Asociar interrupción (ATCH) sirve para asignar el evento de interrupción (indicado por el número de evento) a una parte del programa (indicada por el número de la rutina de interrupción). También es posible asociar varios eventos de interrupción a una única rutina de interrupción. Por el contrario, no se puede asociar un sólo evento a distintas rutinas. Cuando se produce un evento estando habilitadas las interrupciones, se ejecuta únicamente la última rutina de interrupción asociada a dicho evento.

Cuando se asocia un evento a una rutina de interrupción, se habilita automáticamente el evento. Si se inhiben todos los eventos de interrupción, entonces cada vez que se presente la interrupción, se pondrá en cola de espera hasta que las interrupciones se habiliten de nuevo, utilizando para ello la operación Habilitar todos los eventos de interrupción.

También es posible inhibir ciertos eventos de interrupción, eliminando la asociación entre el evento y la correspondiente rutina mediante la operación DTCH (Desasociar interrupción). Esta operación retorna la interrupción a un estado inactivo o ignorado.

La tabla 9-20 muestra los diferentes tipos de eventos de interrupción.

Tabla 9-20 Descripción de los eventos de interrupción

Nº de evento	Descripción de la interrupción	CPU 221	CPU 222	CPU 224
0	Flanco positivo, I0.0	Sí	Sí	Sí
1	Flanco negativo, I0.0	Sí	Sí	Sí
2	Flanco positivo, I0.1	Sí	Sí	Sí
3	Flanco negativo, I0.1	Sí	Sí	Sí
4	Flanco positivo, I0.2	Sí	Sí	Sí
5	Flanco negativo, I0.2	Sí	Sí	Sí
6	Flanco positivo, I0.3	Sí	Sí	Sí
7	Flanco negativo, I0.3	Sí	Sí	Sí
8	Puerto 0: Recibir carácter	Sí	Sí	Sí
9	Puerto 0: Transmisión finalizada	Sí	Sí	Sí
10	Interrupción temporizada 0, SMB34	Sí	Sí	Sí
11	Interrupción temporizada 1, SMB35	Sí	Sí	Sí
12	HSC0 CV=PV (valor actual = valor predeterminado)	Sí	Sí	Sí
13	HSC1 CV=PV (valor actual = valor predeterminado)			Sí
14	HSC1 cambio de sentido			Sí
15	HSC1, puesto a 0 externamente			Sí
16	HSC2 CV=PV (valor actual = valor predeterminado)			Sí
17	HSC2 cambio de sentido			Sí
18	HSC2, puesto a 0 externamente			Sí
19	Interrupción Valor de contaje de impulsos PLS0	Sí	Sí	Sí
20	Interrupción Valor de contaje de impulsos PLS1	Sí	Sí	Sí
21	Interrupción temporizador T32 CT = PT	Sí	Sí	Sí
22	Interrupción temporizador T96 CT = PT	Sí	Sí	Sí
23	Puerto 0: Recepción de mensajes finalizada	Sí	Sí	Sí
24	Puerto 1: Recepción de mensajes finalizada			
25	Puerto 1: Recibir carácter			
26	Puerto 1: Transmisión finalizada			
27	HSC0 cambio de sentido	Sí	Sí	Sí
28	HSC0, puesto a 0 externamente	Sí	Sí	Sí
29	HSC4 CV=PV (valor actual = valor predeterminado)	Sí	Sí	Sí
30	HSC4 cambio de sentido	Sí	Sí	Sí
31	HSC4, puesto a 0 externamente	Sí	Sí	Sí
32	HSC3 CV=PV (valor actual = valor predeterminado)	Sí	Sí	Sí
33	HSC5 CV=PV (valor actual = valor predeterminado)	Sí	Sí	Sí

## Retorno desde rutina de interrupción

K O P	—(RETI)
F U P	RETI
A W L	CRETI
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
221	222
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
224	224

La operación **Retorno condicional desde rutina de interrupción** finaliza una rutina en función de la combinación lógica precedente. Para añadir una interrupción, elija los comandos de menú **Edición > Insertar > Interrupción**.

Operandos: ninguno

Tipos de datos: ninguno

En la pantalla de STEP 7-Micro/WIN 32, los retornos desde rutinas de interrupción se identifican con fichas por separado.

## Rutinas de interrupción

La rutina de interrupción se ejecuta como respuesta a un evento interno o externo asociado. Tras haberse ejecutado la última operación de la rutina de interrupción, el control retorna al programa principal. Para salir de la rutina se puede ejecutar una operación Retorno condicional desde rutina de interrupción (CRETI).

## Reglas para utilizar interrupciones

El procesamiento de interrupciones permite reaccionar rápidamente ante determinados eventos internos o externos. Las rutinas de interrupción se deben estructurar de forma que, una vez ejecutadas determinadas tareas, devuelvan el control al programa principal. Para ello es conveniente crear rutinas de interrupción cortas con indicaciones precisas, de manera que se puedan ejecutar rápidamente sin interrumpir otros procesos durante períodos demasiado largos. Si no se observan estas medidas, es posible que se produzcan estados imprevistos que pueden afectar a la instalación controlada por el programa principal. Al utilizar interrupciones, conviene atenerse al lema de "cuanto más breve, mejor".

## Restricciones

No utilice las operaciones DISI, ENI, HDEF, LSCR y END en las rutinas de interrupción.

## Soporte del sistema durante las interrupciones

Como las interrupciones pueden afectar a la lógica de contactos, bobinas y acumuladores, el sistema almacena la pila lógica, los acumuladores y las marcas especiales (SM) que indican el estado de los acumuladores y las operaciones, volviéndolos a cargar posteriormente. De este modo se previenen perturbaciones en el programa principal debidas a derivaciones a rutinas de interrupción o desde ellas.

## Llamar a subrutinas desde rutinas de interrupción

Desde una rutina de interrupción se puede llamar a un nivel de anidamiento de subrutinas. Los acumuladores y la pila lógica son compartidos por la rutina de interrupción y por la subrutina invocada.

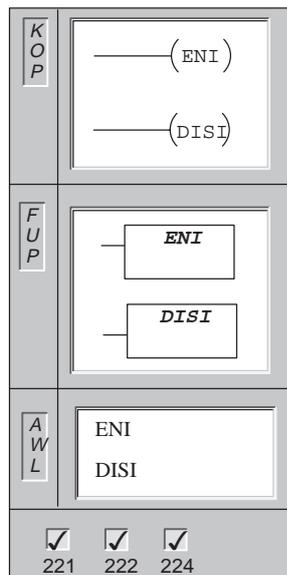
## Datos compartidos por el programa principal y las rutinas de interrupción

El programa principal y una o varias rutinas de interrupción pueden compartir datos. Por ejemplo, una parte del programa principal puede suministrar datos a una rutina de interrupción o viceversa. En el caso de que el programa esté compartiendo datos, habrá que considerar también el hecho de que las rutinas de interrupción se ejecutan de forma asíncrona al programa principal. Por lo tanto, se pueden presentar en cualquier momento durante la ejecución de éste último. Los problemas de coherencia de los datos compartidos pueden ser ocasionados por las acciones de las rutinas de interrupción, al interrumpir éstas la ejecución de las operaciones del programa principal.

Hay diversas técnicas de programación que se pueden utilizar para garantizar que el programa principal y las rutinas de interrupción compartan los datos correctamente. Dichas técnicas restringen la forma de acceder a las direcciones compartidas en la memoria o evitan que se interrumpan las secuencias de operaciones que utilicen direcciones compartidas.

- En un programa AWL que comparta sólo una variable: Si los datos compartidos son una sola variable en formato de byte, palabra o palabra doble, y el programa se ha escrito en AWL, los resultados intermedios de operaciones con datos compartidos sólo se podrán almacenar en direcciones o en acumuladores que no se compartan.
- En un programa KOP que comparta sólo una variable: Si los datos compartidos son una sola variable en formato de byte, palabra o palabra doble, y el programa se ha escrito en KOP, es preciso acceder a las direcciones compartidas utilizando las operaciones de transferencia (MOVB, MOVW, MOVDW, MOVR). En tanto que numerosas operaciones KOP comprenden secuencias de instrucciones AWL que se pueden interrumpir, estas operaciones de transferencia equivalen a una sola operación AWL, cuya ejecución no se ve afectada por los eventos de interrupción.
- En un programa AWL o KOP que comparta varias variables: Si los datos compartidos son varios bytes, palabras o palabras dobles contiguas, la ejecución de la rutina de interrupción se puede controlar con las operaciones Habilitar todos los eventos de interrupción (ENI) e Inhibir todos los eventos de interrupción (DISI). Las interrupciones se deben inhibir en aquel punto del programa principal donde figuran las operaciones que acceden a las direcciones compartidas. Una vez ejecutadas todas las operaciones que utilicen las direcciones compartidas, se deberán habilitar de nuevo las interrupciones. Mientras esté inhibida la interrupción no se podrá ejecutar la rutina correspondiente. Por lo tanto, no será posible acceder entonces a las direcciones compartidas. Sin embargo, esta técnica de programación puede causar que se ignoren los eventos de interrupción.

### Habilitar todos los eventos de interrupción, Inhibir todos los eventos de interrupción



La operación **Habilitar todos los eventos de interrupción** habilita la ejecución de todos los eventos asociados.

La operación **Inhibir todos los eventos de interrupción** inhibe la ejecución de todos los eventos asociados.

Operandos:            ninguno

Tipos de datos:      ninguno

Cuando la CPU pasa a modo RUN, las interrupciones se inhiben. Estando en modo RUN, se pueden habilitar todos los eventos de interrupción con la operación global ENI. La operación DISI permite poner las interrupciones en cola de espera, pero no llamar a ninguna rutina de interrupción.

### Interrupciones de comunicación

El puerto serie de comunicación del sistema de automatización se puede controlar mediante un programa KOP o AWL. La comunicación a través de dicho puerto se denomina modo Freeport (comunicación programable por el usuario). En modo Freeport, el programa define la velocidad de transferencia, los bits por carácter, la paridad y el protocolo. Las interrupciones de transmisión y recepción permiten controlar la comunicación mediante el programa. Para obtener más información al respecto, consulte la descripción de las operaciones Transmitir mensaje y Recibir mensaje.

### Interrupciones E/S

Las interrupciones E/S abarcan interrupciones en flancos positivos y negativos, interrupciones de los contadores rápidos, así como interrupciones de la salida de impulsos. La CPU puede generar una interrupción en flancos positivos y/o negativos en una entrada. En la tabla 9-21 figuran las entradas disponibles para las interrupciones. Los eventos Flanco positivo y Flanco negativo se pueden capturar para cada una de dichas entradas. Estos eventos también se pueden utilizar para indicar una condición que requiera atención inmediata en cuanto se produzca el evento.

Tabla 9-21 Interrupciones asistidas en los flancos positivos y/o negativos

Interrupciones E/S	CPU S7-200
Entradas y salidas	I0.0 a I0.3

Las interrupciones de los contadores rápidos permiten responder rápidamente a condiciones tales como: a) el valor actual alcanza el valor predeterminado, b) el sentido de contaje cambia de forma inversa al sentido de giro del árbol de accionamiento y c) el contador se pone a 0 externamente. Cada uno de estos eventos de los contadores rápidos permite reaccionar ante eventos que no se puedan controlar durante el tiempo de ciclo del sistema de automatización.

Las interrupciones de salida de impulsos dan un aviso inmediato cuando finaliza la salida del número indicado de impulsos. Por lo general, las salidas de impulsos se utilizan para controlar motores paso a paso.

Todas estas interrupciones se habilitan asociando una rutina de interrupción al evento E/S en cuestión.



## Interrupciones temporizadas

Las interrupciones temporizadas incluyen también las de los temporizadores T32/T96. La CPU puede asistir interrupciones temporizadas. Las interrupciones temporizadas se utilizan para indicar tareas que deban ejecutarse cíclicamente. El tiempo de ciclo se incrementa en intervalos de 1 ms, abarcando desde 1 ms hasta 255 ms. El tiempo de ciclo de la interrupción temporizada 0 se debe escribir en SMB34, y el de la interrupción temporizada 1, en SMB35.

Cada vez que termina la temporización, el evento de interrupción temporizado transfiere el control a la rutina de interrupción correspondiente. Típicamente, las interrupciones temporizadas se utilizan para controlar el muestreo de las entradas analógicas en intervalos regulares o para ejecutar un bucle PID.

Al asociar un evento de interrupción temporizado a una rutina de interrupción, se habilita el evento e inmediatamente se empieza a temporizar. Durante ese proceso, el sistema captura el valor del tiempo de ciclo de forma que los cambios siguientes no lo pueden alterar. Para poder modificar el tiempo de ciclo se deberá cambiar el valor del mismo y reasociar luego la rutina de interrupción al evento de la interrupción temporizada. Al reasociarse la rutina de interrupción, la función borra los tiempos acumulados de la asociación anterior, con lo cual se vuelve a temporizar a partir del nuevo valor.

Una vez habilitada, la interrupción funciona de forma continua ejecutando la rutina asociada cada vez que transcurre el intervalo de tiempo indicado. La interrupción temporizada se inhibe saliendo del modo RUN o desasociándola de la rutina correspondiente (mediante la operación DTCH). Si se ejecuta la operación Inhibir todos los eventos de interrupción, se siguen generando interrupciones temporizadas, pero se ponen en cola de espera (hasta que se habiliten nuevamente o hasta llenarse dicha cola). La figura 9-66 muestra un ejemplo de utilización de una interrupción temporizada.

Las interrupciones de los temporizadores T32 y T96 permiten reaccionar de forma temporizada una vez transcurrido un determinado intervalo de tiempo. Dichas interrupciones se asisten únicamente en T32 y T96, siendo éstos temporizadores de retardo a la conexión (TON) con resolución de 1 ms. Por lo demás, T32 y T96 disponen de las funciones habituales. Una vez habilitada la interrupción, la rutina asociada se ejecutará cuando el valor actual del temporizador activo sea igual a su valor de preselección al actualizar la CPU el temporizador de 1 ms. Estas interrupciones se habilitan asociando una rutina de interrupción a los eventos de interrupción T32/T96.

## Prioridades de las interrupciones y colas de espera

La prioridad de las interrupciones es la siguiente:

- Interrupciones de comunicación (prioridad más alta)
- Interrupciones E/S
- Interrupciones temporizadas (prioridad más baja)

La CPU procesa las interrupciones según su prioridad y después en el orden en que aparecen. Sólo se ejecuta una rutina de interrupción en cada caso. Cuando se comienza a ejecutar una rutina de interrupción, se procesa hasta el final. No puede ser interrumpida por otra, ni siquiera por una rutina de mayor prioridad. Las interrupciones que aparezcan mientras se esté ejecutando otra interrupción se ponen en cola de espera para ser procesadas posteriormente.

La tabla 9-22 muestra las tres colas de espera y el número máximo de interrupciones que pueden acoger.

Tabla 9-22 Colas de espera y número máximo de interrupciones que pueden acoger

Cola de espera	CPU 221	CPU 222	CPU 224
Interrupciones de comunicación	4	4	4
Interrupciones E/S	16	16	16
Interrupciones temporizadas	8	8	8

Pueden presentarse más interrupciones de las que puede acoger la cola de espera. Por esta razón, el sistema dispone de marcas de desbordamiento que indican qué eventos de interrupción no se han podido acoger en la cola de espera. La tabla 9-23 muestra dichas marcas de desbordamiento. Estas sólo se pueden utilizar en una rutina de interrupción, porque se desactivan tras vaciarse la cola de espera y reanudarse la ejecución del programa principal.

Tabla 9-23 Definiciones de las marcas especiales para el desbordamiento de las colas de espera

Descripción (0 = sin desbordamiento, 1 = desbordamiento)	Marca especial
Desbordamiento de la cola de espera de interrupciones de comunicación	SM4.0
Desbordamiento de la cola de espera de interrupciones E/S	SM4.1
Desbordamiento de la cola de espera de interrupciones temporizadas	SM4.2

La tabla 9-24 muestra las interrupciones, sus prioridades y los números de los eventos asociados.

Tabla 9-24 Descripción de los eventos de interrupción

Nº de evento	Descripción de la interrupción	Prioridad	Prioridad de grupo
8	Puerto 0: Recibir carácter	Comunicación (más alta)	0
9	Puerto 0: Transmisión finalizada		0
23	Puerto 0: Recepción de mensajes finalizada		0
24	Puerto 1: Recepción de mensajes finalizada		1
25	Puerto 1: Recibir carácter		1
26	Puerto 1: Transmisión finalizada		1
19	PTO 0 interrupción completa	Digital (media)	0
20	PTO 1 interrupción completa		1
0	Flanco positivo, I0.0		2
2	Flanco positivo, I0.1		3
4	Flanco positivo, I0.2		4
6	Flanco positivo, I0.3		5
1	Flanco negativo, I0.0		6
3	Flanco negativo, I0.1		7
5	Flanco negativo, I0.2		8
7	Flanco negativo, I0.3		9
12	HSC0 CV=PV (valor actual = valor predeterminado)		10
27	HSC0 cambio de sentido		11
28	HSC0, puesto a 0 externamente		12
13	HSC1 CV=PV (valor actual = valor predeterminado)		13
14	HSC1, cambio de sentido		14
15	HSC1, puesto a 0 externamente		15
16	HSC2 CV=PV		16
17	HSC2 cambio de sentido		17
18	HSC2, puesto a 0 externamente		18
32	HSC3 CV=PV (valor actual = valor predeterminado)		19
29	HSC4 CV=PV (valor actual = valor predeterminado)		20
30	HSC4 cambio de sentido		21
31	HSC4, puesto a 0 externamente		22
33	HSC5 CV=PV (valor actual = valor predeterminado)	23	
10	Interrupción temporizada 0	Temporizada (más baja)	0
11	Interrupción temporizada 1		1
21	Interrupción temporizador T32 CT = PT		2
22	Interrupción temporizador T96 CT = PT		3

### Ejemplo de interrupciones

La figura 9-65 muestra un ejemplo de operaciones con rutinas de interrupción.

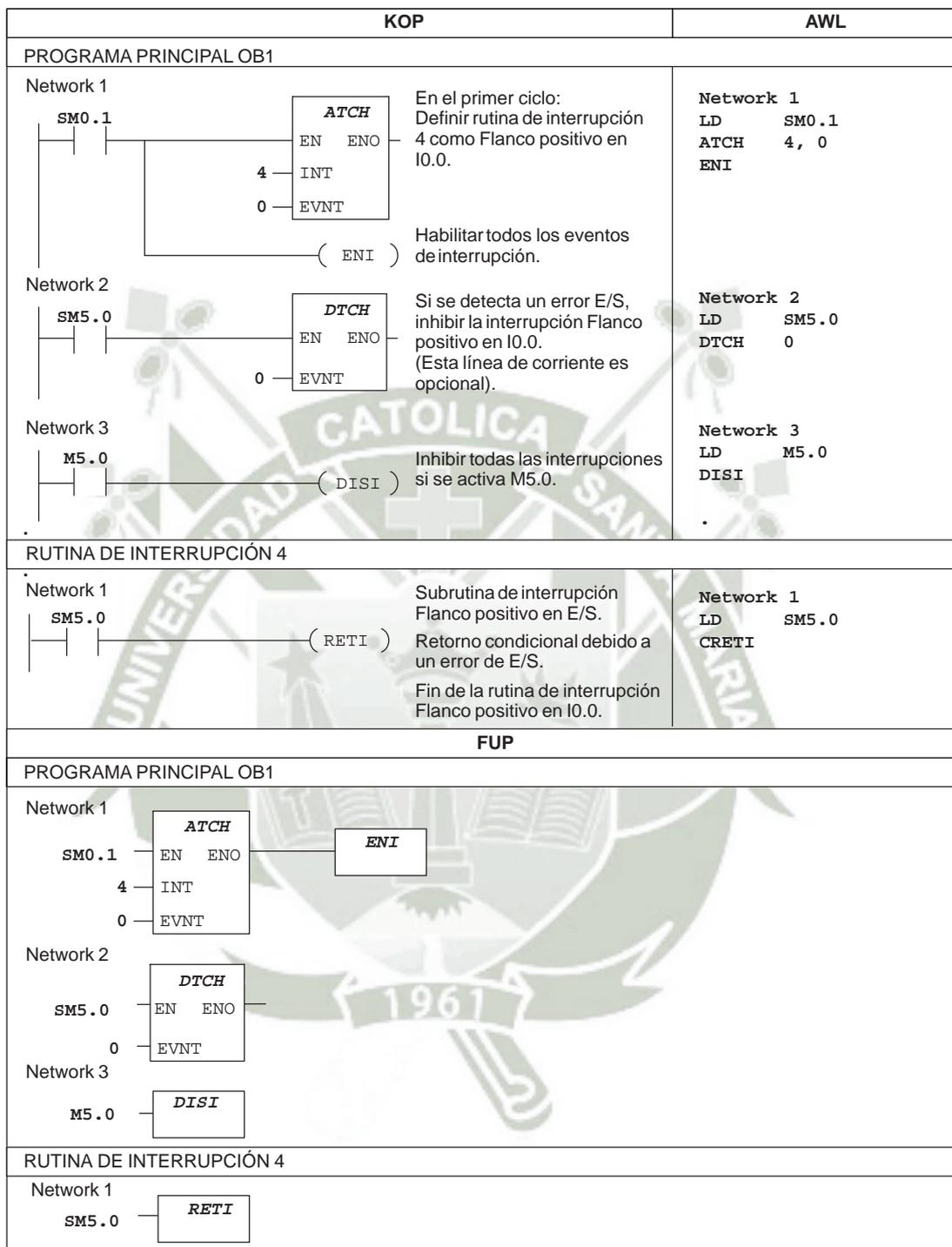


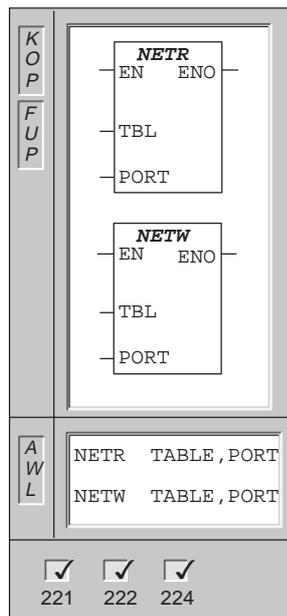
Figura 9-65 Ejemplo de las operaciones de interrupción

La figura 9-66 muestra cómo leer el valor de una entrada analógica mediante una interrupción temporizada.

KOP		AWL	
PROGRAMA PRINCIPAL			
<p>Network 1</p>	<p>Marca del primer ciclo: Llamar subrutina 0.</p>	<p>Network 1</p> <pre>LD SM0.1 CALL 0</pre>	
SUBROUTINA 0			
<p>Network 1</p>	<p>Comenzar subrutina 0. Marca Funcionamiento continuo: ajustar el intervalo de la interrupción temporizada 0 a 100 ms. Habilitar todos los eventos de interrupción. Asociar la interrupción temporizada 0 a la rutina de interrupción 0.</p>	<p>Network 1</p> <pre>LD SM0.0 MOVB 100, SMB34  ATCH 0, 10 ENI</pre>	
RUTINA DE INTERRUPCIÓN 0			
<p>Network 1</p>	<p>Comenzar rutina de interrupción 0. Consultar el estado de AIW4. Finalizar rutina de interrupción.</p>	<p>Network 1</p> <pre>LD SM0.0 MOVW AIW4, VW100</pre>	
<b>FUP</b>			
PROGRAMA PRINCIPAL			
<p>Network 1</p>			
SUBROUTINA 0			
<p>Network 1</p>			
RUTINA DE INTERRUPCIÓN 0			
<p>Network 1</p>			

Figura 9-66 Ejemplo de lectura de una entrada analógica mediante una interrupción temporizada

### Leer de la red, Escribir en la red



La operación **Leer de la red** inicia una comunicación para registrar datos de una estación remota a través del puerto indicado (PORT), según se define en la tabla (TBL).

La operación **Escribir en la red** inicia una comunicación para escribir datos en una estación remota a través del puerto indicado (PORT), según se define en la tabla (TBL).

Con la operación NETR se pueden leer hasta 16 bytes de información de una estación remota, en tanto que con la operación NETW se pueden escribir hasta 16 bytes de información en una estación remota. El programa puede contener un número cualquiera de operaciones NETR/NETW, pero sólo ocho de ellas (en total) pueden estar activadas simultáneamente. Por ejemplo, pueden estar activadas cuatro operaciones NETR y cuatro NETW, o bien, dos operaciones NETR y seis NETW en un sistema de automatización S7-200.

La figura 9-67 muestra la tabla a la que hace referencia el parámetro TBL en las operaciones NETR y NETW.

**NETR:** Condiciones de error que ponen ENO a 0:  
SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

**NETW:** Condiciones de error que ponen ENO a 0:  
SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Entradas/salidas	Operandos	Tipos de datos
TBL	I, Q, M, S, V, VB, MB, *VD, *AC, *LD	BYTE
PORT	constante	BYTE

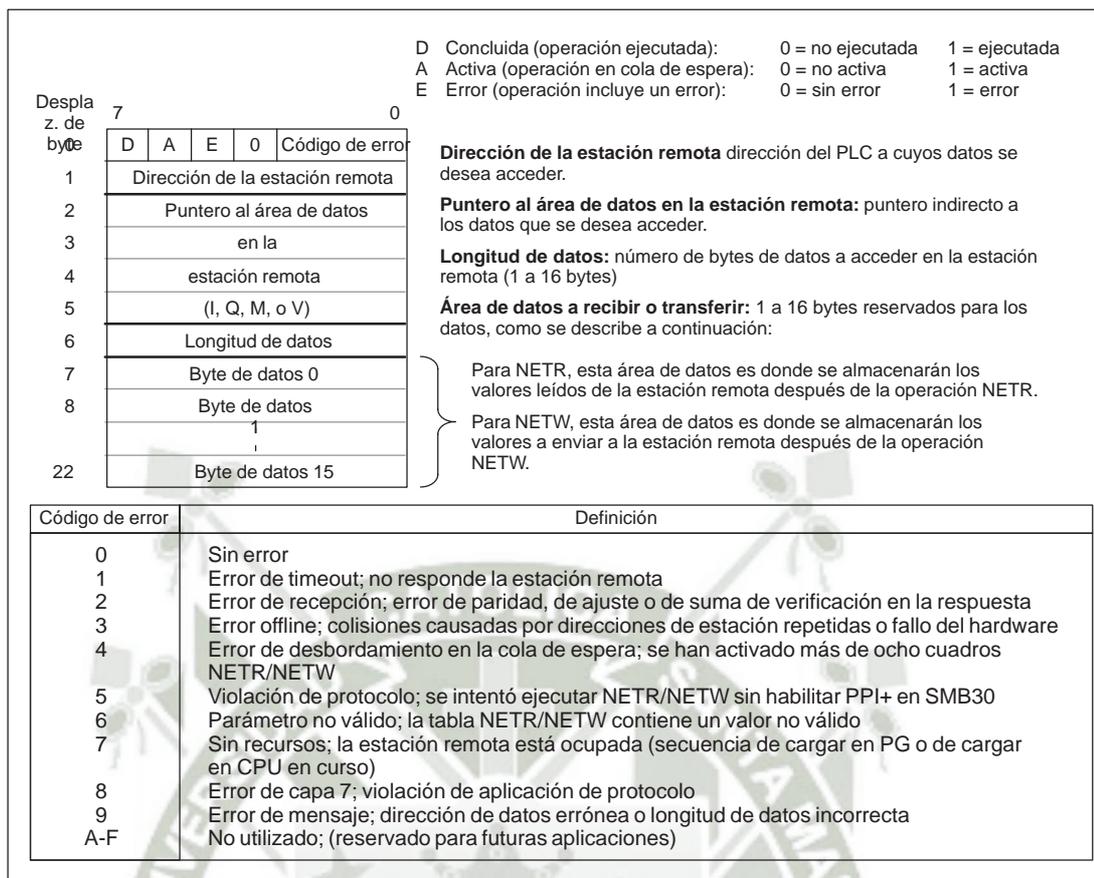


Figura 9-67 Definición de TABLE para NETR y NETW

### Ejemplo de las operaciones Leer de la red y Escribir en la red

La figura 9-68 muestra un ejemplo para ilustrar la utilidad de las operaciones NETR (Leer de la red) y NETW (Escribir en la red). Veamos un ejemplo de una línea de producción donde se están llenando paquetes de mantequilla que se envían a una de las cuatro máquinas empaquetadoras. La empaquetadora embala ocho paquetes de mantequilla en cada caja. Una máquina distribuidora controla el flujo de los paquetes de mantequilla hacia cada una de las empaquetadoras. Se utilizan cuatro CPUs 221 para controlar las empaquetadoras y una CPU 222 equipada con un visualizador de textos TD 200 para controlar a la distribuidora. La figura 9-68 muestra la configuración de la red.

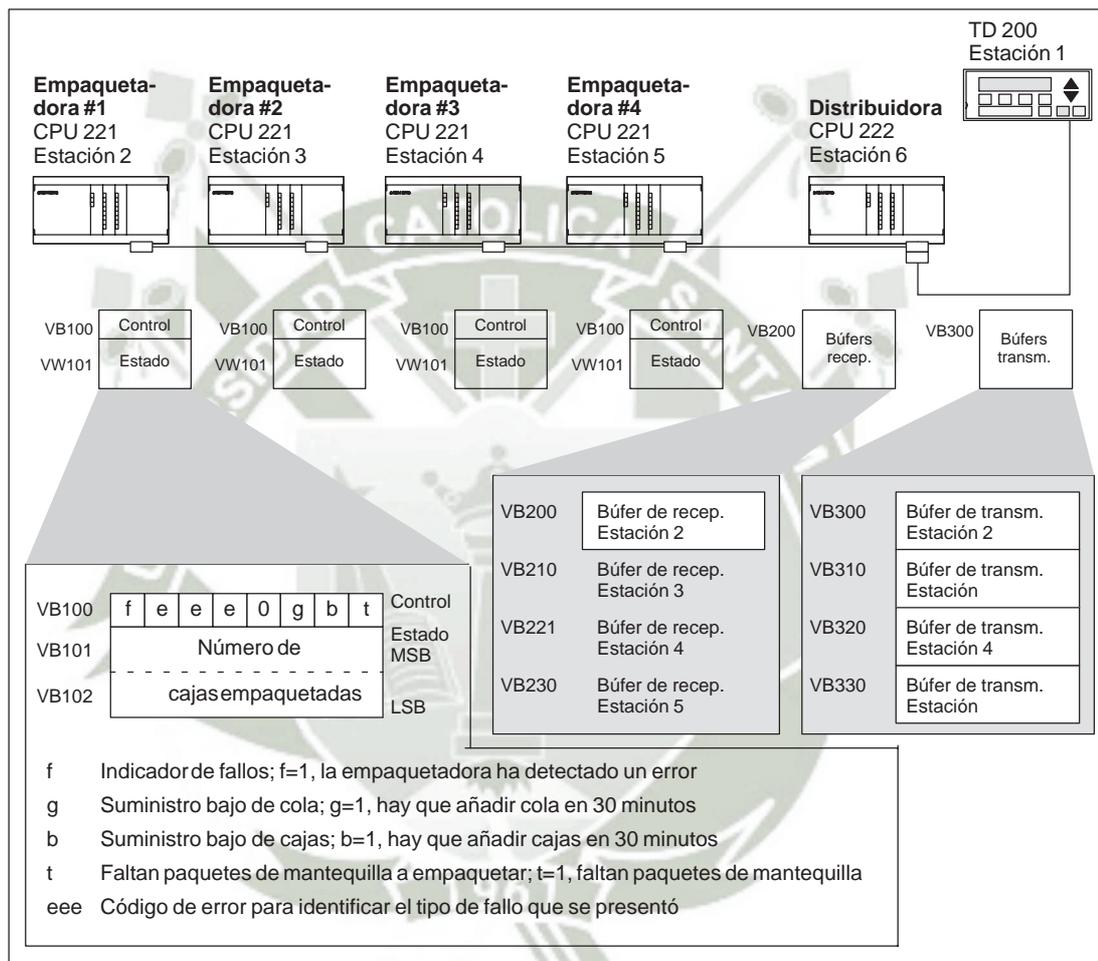


Figura 9-68 Ejemplo de las operaciones NETR y NETW

La figura 9-69 muestra detalladamente los búfers de recepción y transmisión para acceder a los datos de la estación 2 (en VB200 y VB300, respectivamente).

La CPU 224 utiliza la operación NETR para leer continuamente el control y el estado de cada una de las empaquetadoras. Cada vez que una empaquetadora ha embalado 100 cajas, la máquina distribuidora lo registra y envía un mensaje para borrar la palabra de estado utilizando una operación NETW.

La figura 9-70 muestra el programa para leer el byte de control, el número de cajas embaladas y para poner a 0 el número de cajas embalado por una sola empaquetadora (empaquetadora nº 1).

Búfer de recepción de la distribuidora para leer la empaquetadora nº 1						Búfer de transmisión de la distribuidora para borrar el conteo de la empaquetadora nº 1					
7			0			7			0		
VB200	D	A	E	0	Código de error	VB300	D	A	E	0	Código de error
VB201	Dirección de la estación remota					VB301	Dirección de la estación remota				
VB202	Puntero al					VB302	Puntero al				
VB203	área de datos					VB303	área de datos				
VB204	en la					VB304	en la				
VB205	estación remota = (&VB100)					VB305	estación remota = (&VB101)				
VB206	Longitud de datos = 3 bytes					VB306	Longitud de datos = 2 bytes				
VB207	Control					VB307	0				
VB208	Estado (MSB)					VB308	0				
VB209	Estado (LSB)										

Figura 9-69 Ejemplo de TABLE para NETR y NETW

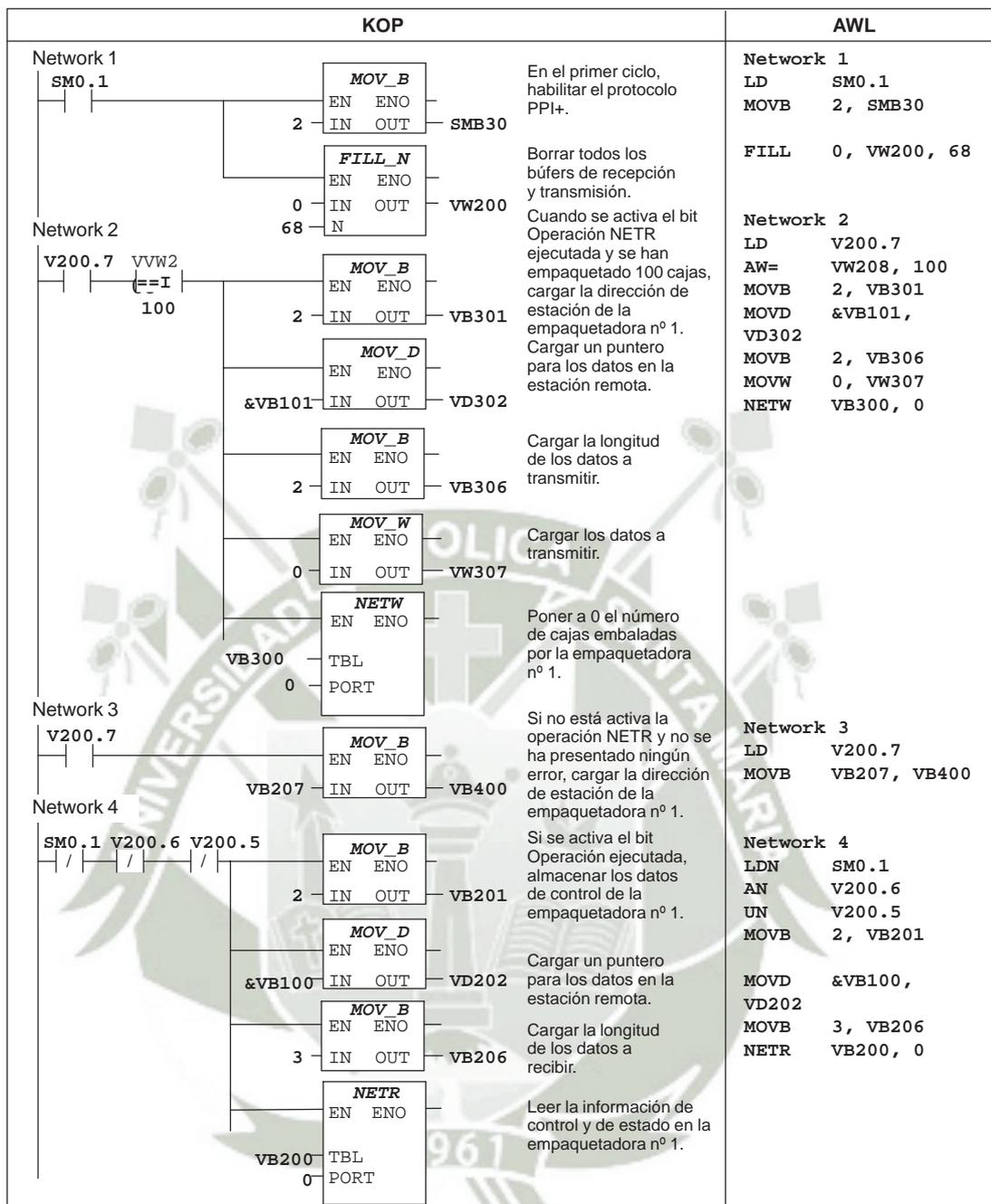


Figura 9-70 Ejemplo de las operaciones NETR y NETW en KOP y AWL

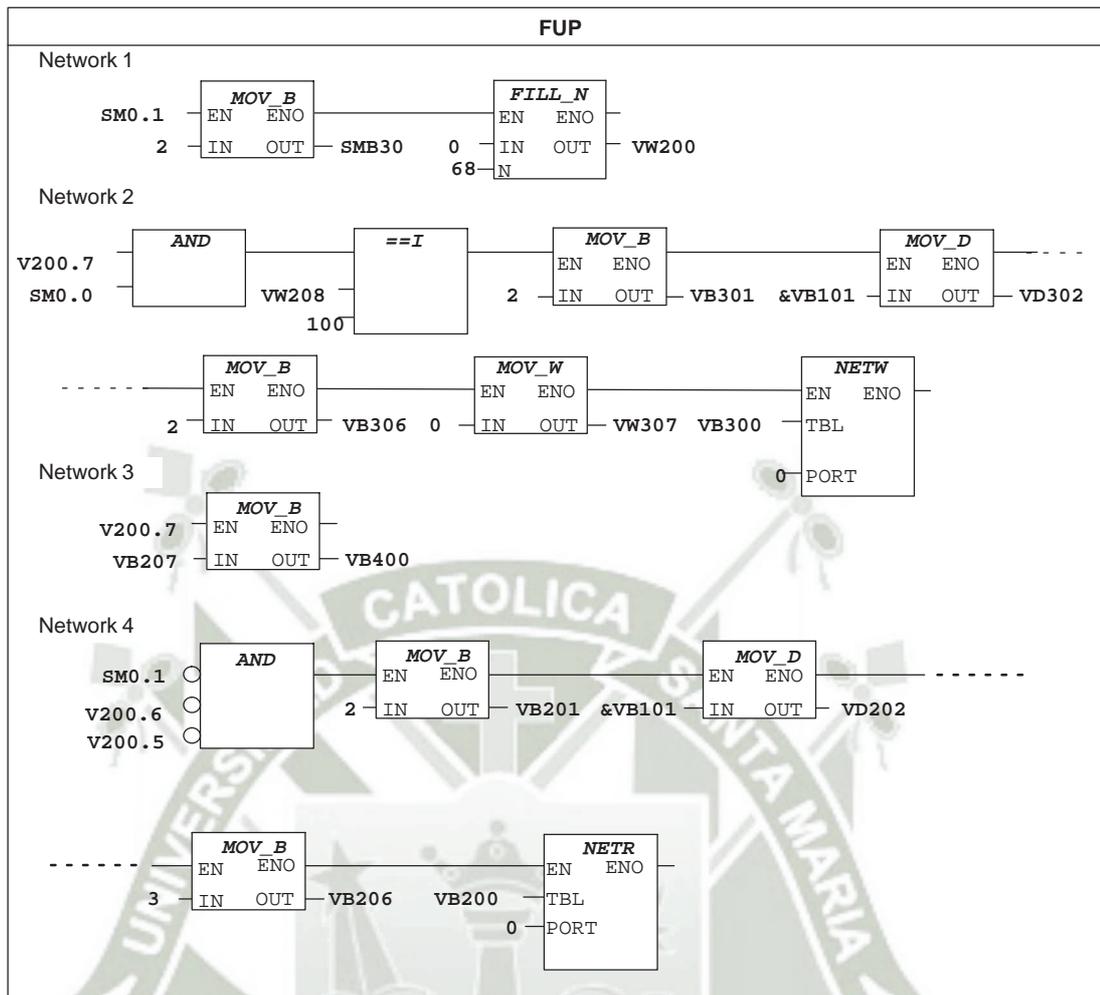
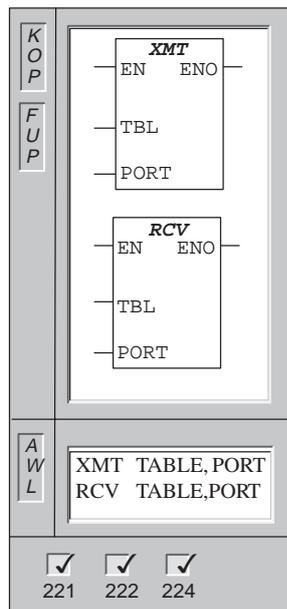


Figura 9-71 Ejemplo de las operaciones NETR y NETW en FUP

## Transmitir mensaje, Recibir mensaje



La operación **Transmitir mensaje** activa la transmisión del búfer de datos (TBL). La primera entrada del búfer indica cuántos bytes se han de transmitir. PORT indica el puerto de programación por donde se va a transmitir.

La operación XMT se utiliza en modo Freeport para transmitir datos por el (los) puerto(s) de comunicación.

El formato del búfer XMT buffer es el siguiente:

La operación **Recibir mensaje** inicia o finaliza la función Recibir mensaje. Para el cuadro Recibir mensaje es preciso indicar una condición inicial y final. Los mensajes que se hayan recibido a través del puerto indicado (PORT) se almacenan en el búfer de datos (TBL). La primera entrada del búfer indica el número de bytes que se han recibido.

**Transmitir mensaje:** Condiciones de error que ponen ENO a 0: SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto), 0009 (XMT/RCV simultáneos en el puerto 0), 000B (XMT/RCV simultáneos en el puerto 1)

**Recibir mensaje:** Condiciones de error que ponen ENO a 0: SM86.6 y SM186.6 (error de parámetro RCV), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto), 0009 (XMT/RCV simultáneos en el puerto 0), 000B (XMT/RCV simultáneos en el puerto 1)

Entradas/salidas	Operandos	Tipos de datos
TABLE	VB, IB, QB, MB, SB, SMB, *VD, *AC, *LD	BYTE
PORT	Constante (0)	BYTE

## Modo Freeport

El programa de usuario puede controlar el puerto serie de la CPU. La comunicación a través de dicho puerto se denomina modo Freeport (comunicación programable por el usuario). Eligiendo el modo Freeport, el programa KOP controla el puerto de comunicación utilizando interrupciones de recepción y de transmisión, así como las operaciones Transmitir mensaje (XMT) y Recibir mensaje (RCV). En modo Freeport, el programa KOP controla todo el protocolo de comunicación. Las marcas especiales SMB30 (para el puerto 0) y SMB130 (para el puerto 1, si la CPU dispone de dos puertos) se utilizan para elegir la velocidad de transferencia y la paridad.

Cuando la CPU pasa a modo STOP se inhibe el modo Freeport y se restablece la comunicación (p.ej. acceso a través de la unidad de programación).

En el caso más simple se puede enviar un mensaje a la impresora o a la pantalla con sólo utilizar la operación Transmitir mensaje (XMT). Otros ejemplos incluyen la conexión a un lector de código de barras, una báscula o una soldadora. En todo caso, el programa deberá asistir el protocolo con el que la CPU se comunica en modo Freeport.

Para poder utilizar el modo Freeport, es preciso que la CPU esté en modo RUN. El modo Freeport se habilita ajustando el valor 01 en el campo de selección del protocolo de SMB30 (puerto 0) o de SMB130 (puerto 1). Estando en modo Freeport, la CPU no se puede comunicar con la unidad de programación.

---

### Nota

La conmutación a modo Freeport se puede controlar con la marca especial SM0.7 que indica la posición actual del selector de modos de operación. Si SM0.7 = 0, el selector está en posición TERM; si SM0.7 = 1, el selector está en posición RUN. Si el modo Freeport se habilita sólo cuando el selector esté en RUN, la unidad de programación se podrá utilizar para vigilar o controlar el funcionamiento de la CPU, cambiando el selector a una posición diferente.

---

## Inicializar el modo Freeport

SMB30 y SMB130 se utilizan para inicializar el modo Freeport en los puertos de comunicación 0 y 1, respectivamente, permitiendo elegir la velocidad de transferencia, la paridad y el número de bits por carácter. La tabla 9-25 muestra los bytes de control del modo Freeport.

Tabla 9-25 Bytes de marcas especiales SMB30 y SMB130

Puerto 0	Puerto 1	Descripción
Formato de SMB30	Formato de SMB130	<div style="display: flex; justify-content: space-between; align-items: center;"> <span>MSB 7</span> <span>LSB 0</span> </div> <div style="display: flex; align-items: center; margin-top: 5px;"> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">p</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">p</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">d</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">b</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">b</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">b</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">m</div> <div style="border: 1px solid black; padding: 2px;">m</div> </div> <p>Byte de control del modo Freeport</p>
SM30.6 y SM30.7	SM130.6 y SM130.7	<p>pp Selección de paridad</p> <p>00 = sin paridad</p> <p>01 = paridad par</p> <p>10 = sin paridad</p> <p>11 = paridad impar</p>
SM30.5	SM130.5	<p>d Bits por carácter</p> <p>0 = 8 bits por carácter</p> <p>1 = 7 bits por carácter</p>
SM30.2 a SM30.4	SM130.2 a SM130.4	<p>bbb Velocidad de transferencia</p> <p>000 = 38.400 bit/s</p> <p>001 = 19.200 bit/s</p> <p>010 = 9.600 bit/s</p> <p>011 = 4.800 bit/s</p> <p>100 = 2.400 bit/s</p> <p>101 = 1.200 bit/s</p> <p>110 = 600 bit/s</p> <p>111 = 300 bit/s</p>
SM30.0 y SM30.1	SM130.0 y SM130.1	<p>mm Selección de protocolo</p> <p>00 = Protocolo de interface punto a punto (PPI/modo esclavo)</p> <p>01 = Protocolo Freeport</p> <p>10 = PPI/modo maestro</p> <p>11 = Reservado (estándar: PPI/modo esclavo)</p>

Nota: Se genera un bit de parada para todas las configuraciones.

## Utilizar la operación XMT para transmitir datos

Con la operación XMT se puede enviar un búfer de uno o más caracteres (hasta un máximo de 255). Una vez transmitido el último carácter del búfer, se genera una interrupción (evento de interrupción 9 para el puerto 0 y evento de interrupción 26 para el puerto 1), si una rutina de interrupción se ha asociado al evento Transmisión finalizada. También es posible transmitir datos sin utilizar interrupciones (p.ej. enviar un mensaje a una impresora), vigilando SM4.5 ó SM4.6 hasta que finalice la transmisión.

La operación XMT se puede utilizar para generar una condición BREAK, poniendo el número de caracteres a cero y ejecutando luego la operación XMT. Así se genera una condición BREAK en la línea de temporizadores de 16 bits a la velocidad de transferencia actual. La transmisión de una condición BREAK se gestiona de la misma forma que la de cualquier otro mensaje. Una interrupción de transmisión se genera cuando se termina de transmitir la condición BREAK, indicando SM4.5 ó SM4.6 el estado actual de la transmisión.

La figura 9-72 muestra el formato del búfer XMT.

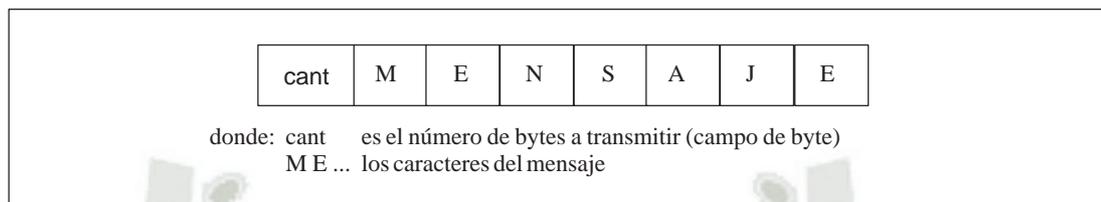


Figura 9-72 Formato del búfer XMT

### Utilizar la operación RCV para recibir datos

Con la operación RCV se puede recibir un búfer de uno o más caracteres (hasta un máximo de 255). Una vez recibido el último carácter del búfer, se genera una interrupción (evento de interrupción 23 para el puerto 0 y evento de interrupción 24 para el puerto 1), si una rutina de interrupción se ha asociado al evento Recepción de mensajes finalizada.

También es posible recibir mensajes sin utilizar interrupciones, vigilando para ello la marca especial SMB86. SMB86 (o SMB186) no serán iguales a cero cuando el cuadro RCV esté inactivo o cuando se haya finalizado. En cambio, será igual a cero cuando se estén recibiendo datos.

La operación RCV permite seleccionar las condiciones para el comienzo y el final de un mensaje. En la tabla 9-26 (SM86 a SM94 para el puerto 0, y SM186 a SM194 para el puerto 1) se describen las condiciones de comienzo y de final de mensajes. La figura 9-73 muestra el formato del búfer RCV.

#### Nota

La recepción de mensajes se finalizará automáticamente si se produce un desbordamiento o un error de paridad. Para la operación Recibir mensaje es preciso definir una condición inicial (x ó z) y una condición final (y, t ó el número máximo de caracteres).

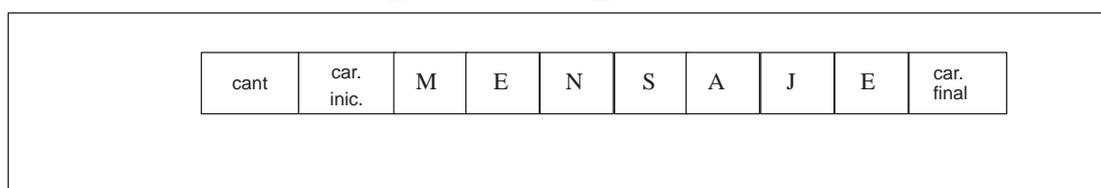


Figura 9-73 Formato del búfer RCV

Tabla 9-26 Marcas especiales SMB86 a SMB94 y SMB186 a SMB194

Puerto 0	Puerto 1	Descripción
SMB86	SMB186	<div style="display: flex; justify-content: space-between; align-items: center;"> <div style="text-align: center;"> <small>MSB</small> 7                             </div> <div style="text-align: center;"> <small>LSB</small> 0                             </div> </div> <div style="display: flex; justify-content: center; align-items: center; margin-top: 5px;"> <div style="border: 1px solid black; padding: 2px 5px; margin: 0 2px;">n</div> <div style="border: 1px solid black; padding: 2px 5px; margin: 0 2px;">r</div> <div style="border: 1px solid black; padding: 2px 5px; margin: 0 2px;">e</div> <div style="border: 1px solid black; padding: 2px 5px; margin: 0 2px;">0</div> <div style="border: 1px solid black; padding: 2px 5px; margin: 0 2px;">0</div> <div style="border: 1px solid black; padding: 2px 5px; margin: 0 2px;">t</div> <div style="border: 1px solid black; padding: 2px 5px; margin: 0 2px;">c</div> <div style="border: 1px solid black; padding: 2px 5px; margin: 0 2px;">p</div> </div> <p style="margin-left: 100px;">Byte de estado de recepción de mensajes</p> <p>n: 1 = El usuario ha inhibido la recepción de mensajes</p> <p>r: 1 = Recepción de mensajes terminada: error en parámetros de entrada o falta condición inicial o final</p> <p>e: 1 = Carácter final recibido</p> <p>t: 1 = Recepción de mensajes terminada: ha transcurrido la temporización</p> <p>c: 1 = Recepción de mensajes terminada: se ha excedido el número máximo de caracteres</p> <p>p: 1 = Recepción de mensajes terminada debido a un error de paridad</p>



Tabla 9-26 Marcas especiales SMB86 a SMB94 y SMB186 a SMB194

Puerto 0	Puerto 1	Descripción
SMB87	SMB187	<div style="display: flex; justify-content: space-between; align-items: center;"> <div style="text-align: center;"> <small>MSB</small> 7                 </div> <div style="text-align: center;"> <small>LSB</small> 0                 </div> </div> <div style="border: 1px solid black; padding: 2px; display: flex; justify-content: center; align-items: center; margin: 5px 0;"> <span style="border: 1px solid black; padding: 2px 5px;">n</span> <span style="border: 1px solid black; padding: 2px 5px;">x</span> <span style="border: 1px solid black; padding: 2px 5px;">y</span> <span style="border: 1px solid black; padding: 2px 5px;">z</span> <span style="border: 1px solid black; padding: 2px 5px;">m</span> <span style="border: 1px solid black; padding: 2px 5px;">t</span> <span style="border: 1px solid black; padding: 2px 5px;">0</span> <span style="border: 1px solid black; padding: 2px 5px;">0</span> </div> <p>Byte de control de recepción de mensajes</p> <p>n: 0 = Inhibida la función de recibir mensajes. 1 = Habilitada la función de recibir mensajes. El bit para habilitar/inhibir la recepción de mensajes se comprueba cada vez que se ejecuta la operación RCV.</p> <p>x: 0 = Ignorar SMB88 o SMB188. 1 = Utilizar el valor de SMB88 o de SMB188 para detectar el comienzo del mensaje.</p> <p>y: 0 = Ignorar SMB89 o SMB189. 1 = Utilizar el valor de SMB89 o de SMB189 para detectar el fin del mensaje.</p> <p>z: 0 = Ignorar SMW90 o SMB190. 1 = Utilizar el valor de SMW90 para detectar una condición de inactividad.</p> <p>m: 0 = Utilizar el temporizador como temporizador entre caracteres. 1 = Utilizar el temporizador como temporizador de mensajes.</p> <p>t: 0 = Ignorar SMW92 o SMW192. 1 = Finalizar la recepción si se excede el período de tiempo indicado en SMW92 o SMW192.</p> <p>Las marcas del byte de control de interrupción de mensajes se utilizan para definir los criterios con los que se identifica el mensaje. Se definen los criterios tanto de comienzo como de final del mensaje. Para determinar el comienzo de un mensaje, uno de los dos juegos de criterios de comienzo de mensaje combinados lógicamente mediante Y deberán ser verdaderos y deberán ocurrir en secuencia (inactividad seguida de un comienzo de carácter o condición BREAK seguida de un comienzo de carácter). Para determinar el fin de un mensaje, los criterios habilitados se combinan mediante O. A continuación se indican las ecuaciones de comienzo y de final:</p> <p style="padding-left: 40px;">Comienzo del mensaje = <math>il * sc + bk * sc</math> Final del mensaje = <math>ec + tmr + \text{contaje máximo de caracteres alcanzado}</math></p> <p>Programar los criterios de comienzo de mensaje para:</p> <ol style="list-style-type: none"> <li>1. Detección de inactividad: <span style="float: right;">il=1, sc=0, bk=0, SMW90&gt;0</span></li> <li>2. Detección de carácter inicial: <span style="float: right;">il=0, sc=1, bk=0, SMW90 no es relevante</span></li> <li>3. Detección BREAK: <span style="float: right;">il=0, sc=1, bk=1, SMW90 no es relevante</span></li> <li>4. Cualquier respuesta a una petición: <span style="float: right;">il=1, sc=0, bk=0, SMW90=0</span> (El temporizador de mensajes se puede utilizar para terminar la recepción si no hay respuesta).</li> <li>5. BREAK y carácter inicial: <span style="float: right;">il=0, sc=1, bk=1, SMW90 no es relevante</span></li> <li>6. Inactividad y carácter inicial: <span style="float: right;">il=1, sc=1, bk=0, SMW90 &gt;0</span></li> <li>7. Inactividad y carácter inicial (no válido): <span style="float: right;">il=1, sc=1, bk=0, SMW90=0</span></li> </ol> <p>Nota: La recepción se finalizará automáticamente si ocurre un error de desbordamiento o de paridad (si se han habilitado).</p>
SMB88	SMB188	Carácter de comienzo del mensaje.
SMB89	SMB189	Carácter de fin del mensaje.

Tabla 9-26 Marcas especiales SMB86 a SMB94 y SMB186 a SMB194

Puerto 0	Puerto 1	Descripción
SMB90 SMB91	SMB190 SMB191	Tiempo de línea de inactividad en milisegundos. El primer carácter recibido una vez transcurrido el tiempo de línea de inactividad es el comienzo del nuevo mensaje. SM90 (o SM190) es el byte más significativo y SM91 (o SM191) es el byte menos significativo.
SMB92 SMB93	SMB192 SMB193	Vigilancia de tiempo del temporizador entre caracteres/mensajes en milisegundos. Si se excede el tiempo, se finaliza la recepción de mensajes. SM92 (o SM192) es el byte más significativo y SM93 (o SM193) es el byte menos significativo.
SMB94	SMB194	Número máximo de caracteres a recibir (1 a 255 bytes). Nota: Este margen debe ajustarse al tamaño máximo esperado para el búfer, incluso si no se utiliza la finalización de mensajes por el conteo de caracteres.

### Recibir datos mediante interrupciones de caracteres

Para disponer de una mayor flexibilidad en los protocolos asistidos, los datos se pueden recibir también de forma controlada por interrupciones de caracteres. Cada carácter recibido genera una interrupción. El carácter recibido se deposita en SMB2 y el estado de la paridad (si se ha habilitado) se deposita en SM3.0. Ello sucede inmediatamente antes de ejecutarse la rutina de interrupción asociada al evento Recibir carácter.

- SMB2 es el búfer de recepción de caracteres en modo Freeport. Cada carácter recibido en modo Freeport se deposita en esta dirección para que el programa de usuario pueda acceder rápidamente a los valores.
- SMB3 se utiliza para el modo Freeport y contiene un bit de error de paridad que se activa si se detecta un error de ese tipo en un carácter recibido. Todos los demás bits del byte se reservan. Utilice este bit para rechazar el mensaje o para generar un acuse negativo del mensaje.

#### Nota

SMB2 y SMB3 son compartidos por los puertos 0 y 1. Si debido a la recepción de un carácter por el puerto 0 se ejecuta la rutina de interrupción asociada a ese evento (evento de interrupción 8), SMB2 contendrá el carácter recibido por el puerto 0, en tanto que SMB3 contendrá la paridad de dicho carácter. Si debido a la recepción de un carácter por el puerto 1 se ejecuta la rutina de interrupción asociada a ese evento (evento de interrupción 25), SMB2 contendrá el carácter recibido por el puerto 1, en tanto que SMB3 contendrá la paridad de dicho carácter.

### Ejemplo de las operaciones Recibir mensaje y Transmitir mensaje

Este programa de ejemplo muestra la utilización de las operaciones Recibir mensaje y Transmitir mensaje. El programa recibirá una cadena de caracteres hasta que se reciba un carácter que indique un cambio de línea. El mensaje se retornará entonces al emisor.

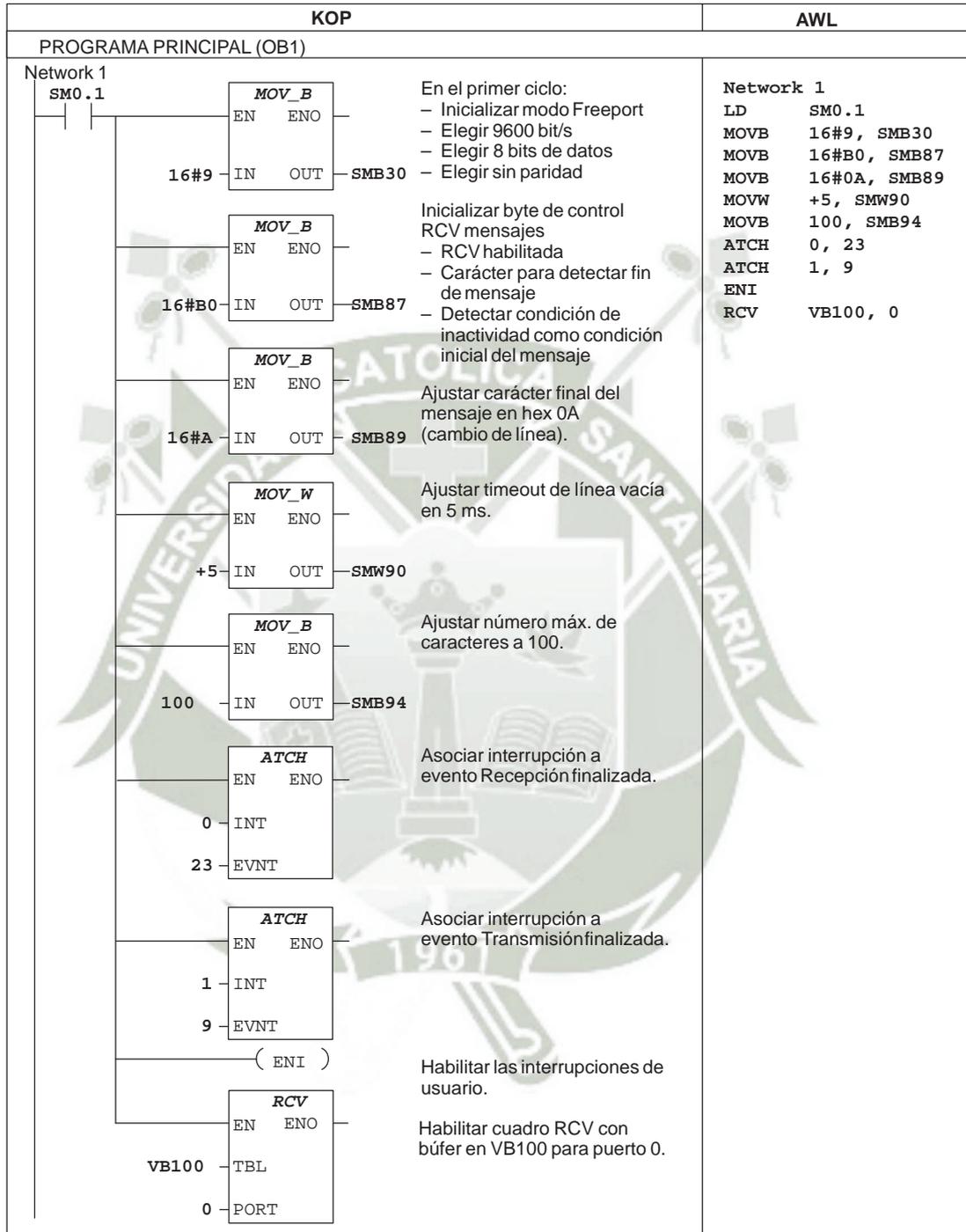


Figura 9-74 Ejemplo de la operación Transmitir mensaje



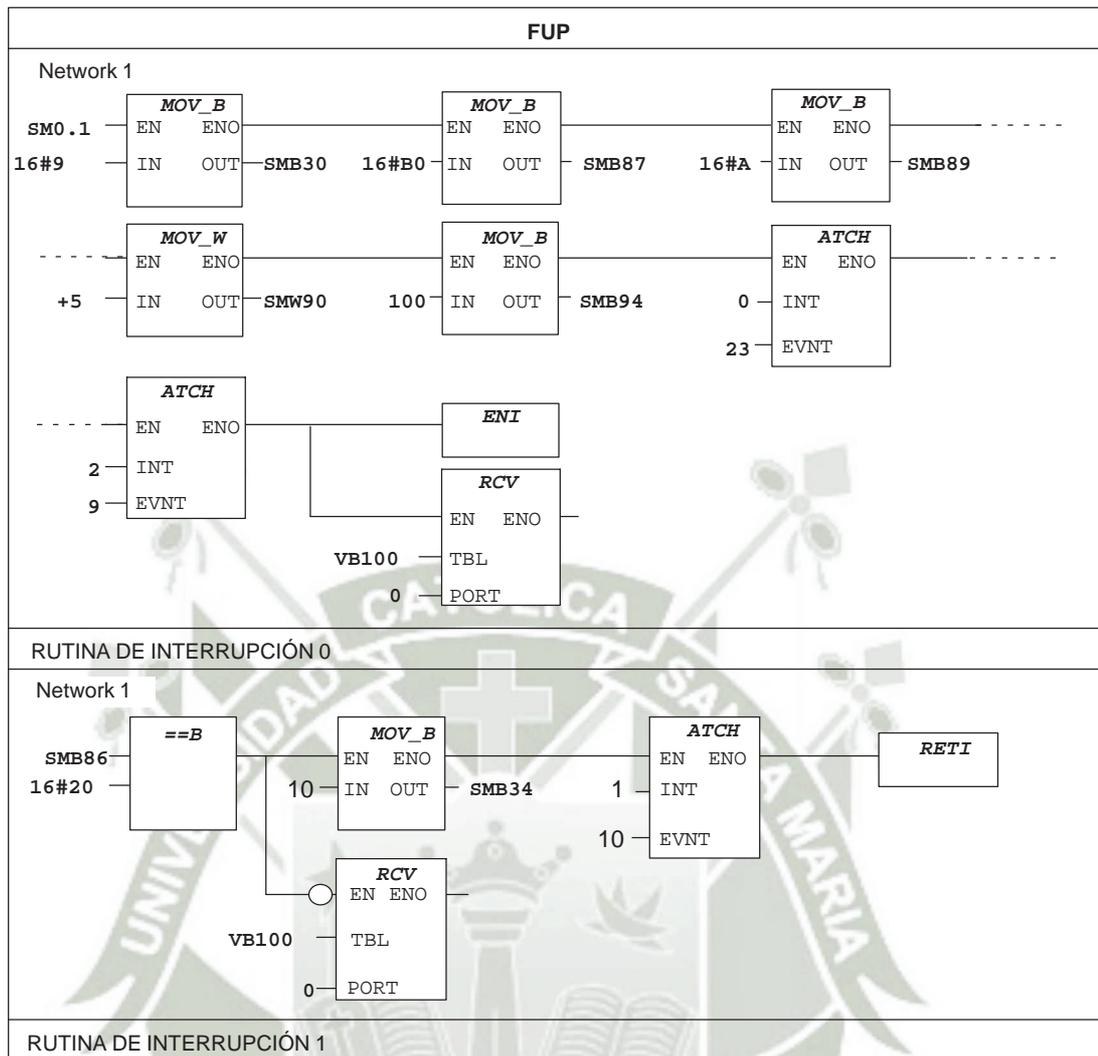


Figura 9-74 Ejemplo de la operación Transmitir mensaje (continuación)

## 9.17 Operaciones lógicas de pilas (SIMATIC)

### Combinar primer y segundo valor mediante Y

A	ALD	
W		
L		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
221	222	224

La operación **Combinar primer y segundo valor mediante Y** (ALD) combina los valores del primer y segundo nivel de la pila mediante una operación lógica Y. El resultado se carga en el nivel superior de la pila. Una vez ejecutada la operación ALD, la profundidad de la pila tiene un nivel menos.

Operandos: ninguno

### Combinar primer y segundo valor mediante O

A	OLD	
W		
L		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
221	222	224

La operación **Combinar primer y segundo valor mediante O** combina los valores del primer y segundo nivel de la pila mediante una operación lógica O. El resultado se carga en el nivel superior de la pila. Una vez ejecutada la operación OLD, la profundidad de la pila tiene un nivel menos.

Operandos: ninguno

### Duplicar primer valor

A	LPS	
W		
L		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
221	222	224

La operación **Duplicar primer valor** duplica el primer valor de la pila y lo desplaza dentro de la misma. El último valor de la pila se expulsa y se pierde.

Operandos: ninguno

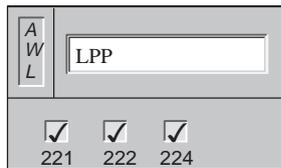
### Copiar segundo valor

A	LRD	
W		
L		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
221	222	224

La operación **Copiar segundo valor** copia el segundo valor de la pila en el nivel superior de la misma. En la pila no se carga ni se expulsa ningún valor. No obstante, el valor que se encontraba en el nivel superior se sobrescribe con el nuevo valor.

Operandos: ninguno

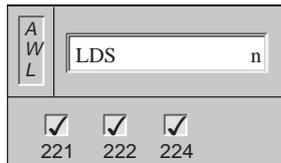
### Sacar primer valor



La operación **Sacar primer valor** desplaza el primer valor fuera de la pila. El segundo valor se convierte entonces en el primer nivel de la pila.

Operandos: ninguno

### Cargar pila



La operación **Cargar pila** duplica el bit n de la pila y lo deposita en el nivel superior de la misma. El último valor de la pila se expulsa y se pierde.

Operandos: n (1 a 8)

### Operaciones lógicas de pilas

La figura 9-75 muestra cómo funcionan las operaciones Combinar primer y segundo valor mediante Y y mediante O.

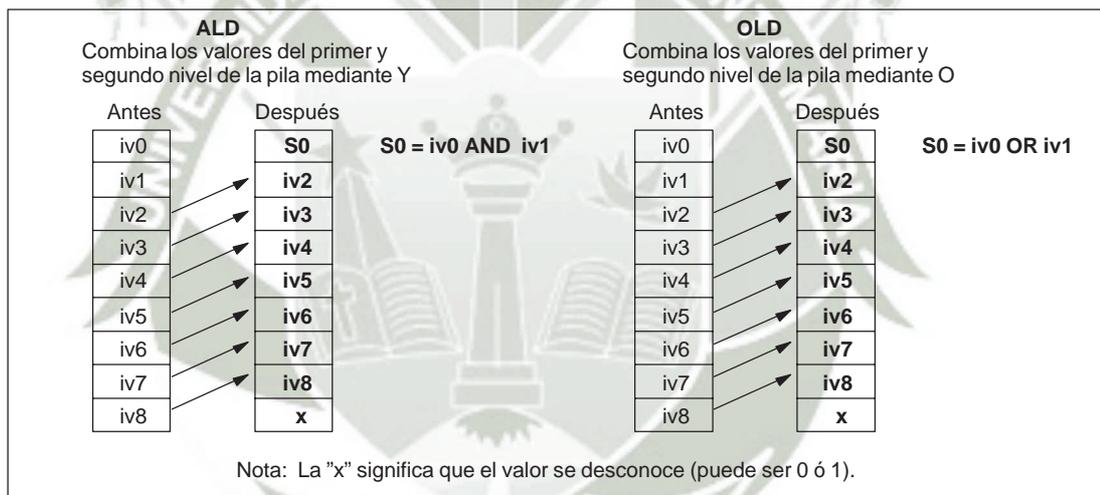


Figura 9-75 Operaciones Combinar primer y segundo valor mediante Y y mediante O

La figura 9-76 muestra cómo funcionan las operaciones Duplicar primer valor, Copiar segundo valor y Sacar primer valor.

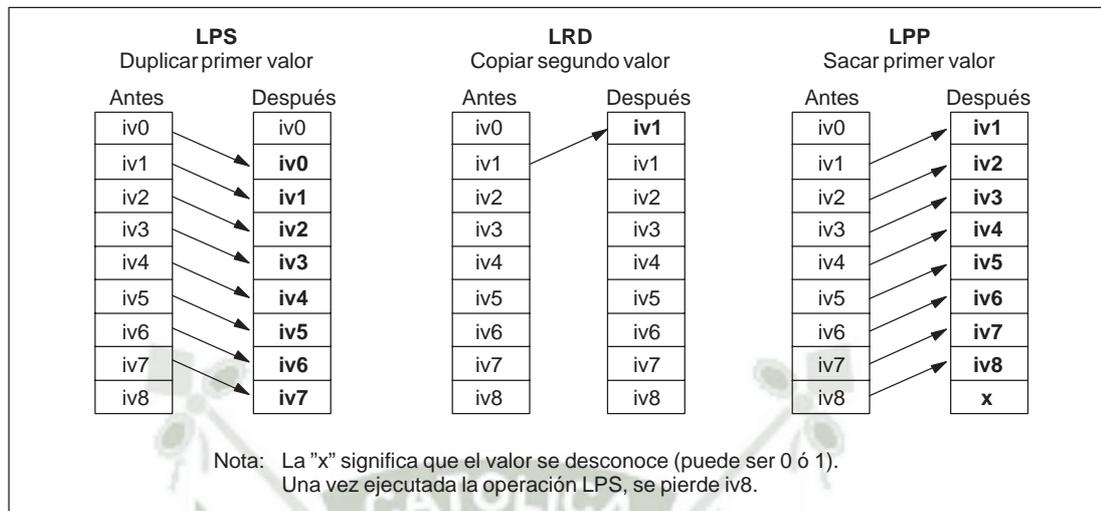


Figura 9-76 Operaciones Duplicar primer valor, Copiar segundo valor y Sacar primer valor

La figura muestra 9-77 cómo funciona la operación Cargar pila.

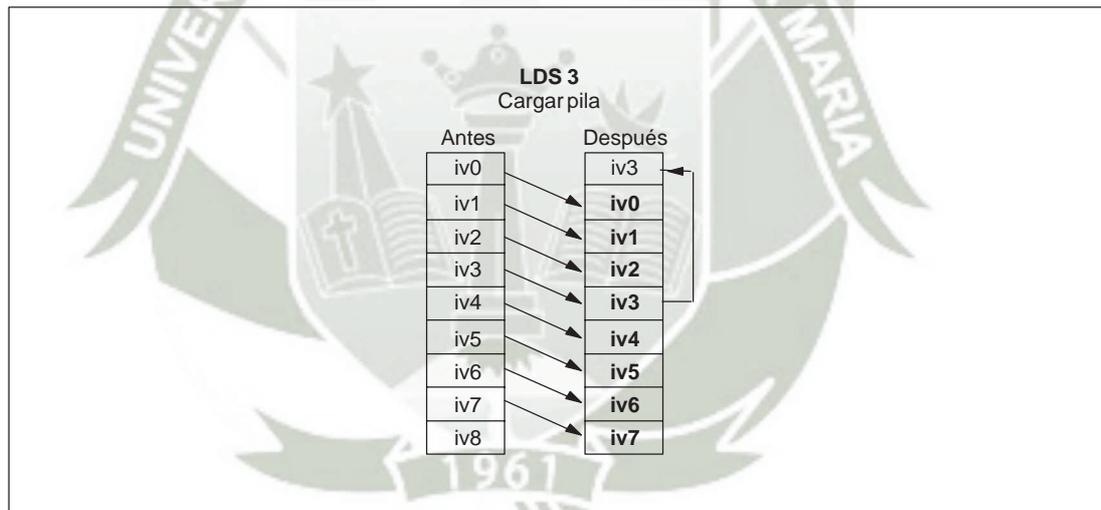


Figura 9-77 Operación Cargar pila

Ejemplo de una operación lógica de pila

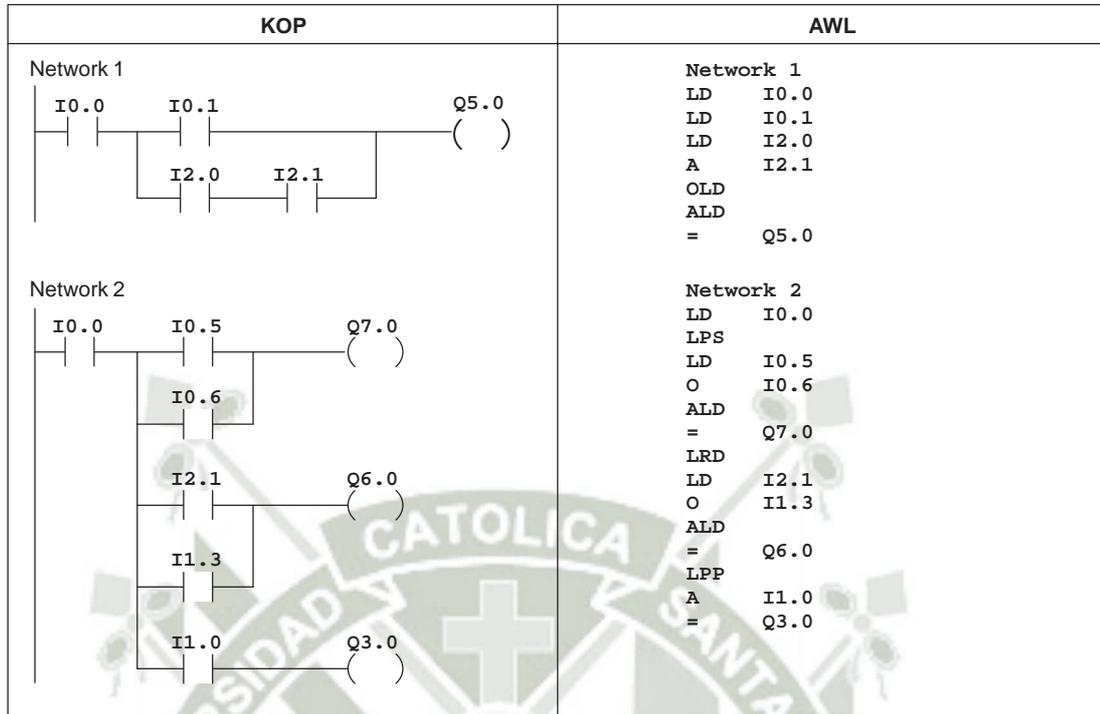


Figura 9-78 Ejemplo de una operación lógica de pila en KOP y AWL

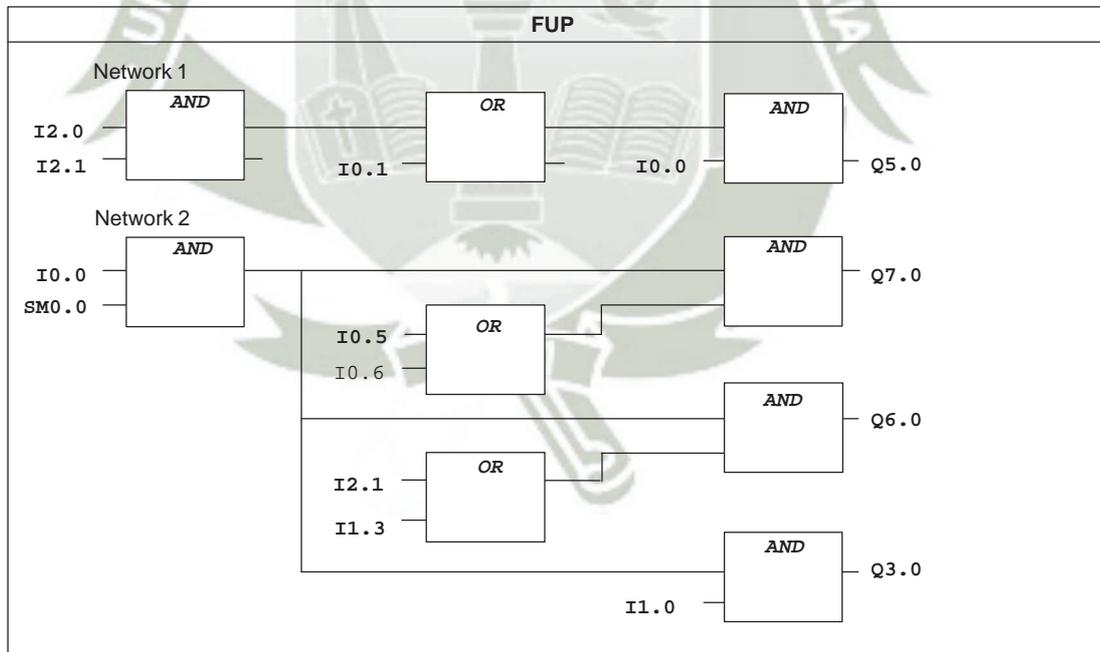


Figura 9-79 Ejemplo de una operación lógica de pila en FUP

# 10

## Operaciones IEC 1131-3

El presente capítulo describe las operaciones IEC 1131-3 estándar. Hay algunas operaciones SIMATIC que se pueden utilizar en programas IEC. Éstas se denominan operaciones IEC no normalizadas, indicándose al comienzo de cada apartado.

### Índice del capítulo

Apartado	Descripción	Página
10.1	Operaciones lógicas con bits (IEC)	10-2
10.2	Operaciones de comparación (IEC)	10-7
10.3	Operaciones de temporización (IEC)	10-11
10.4	Operaciones con contadores (IEC)	10-15
10.5	Operaciones aritméticas (IEC)	10-19
10.6	Operaciones de transferencia (IEC)	10-24
10.7	Operaciones lógicas (IEC)	10-26
10.8	Operaciones de desplazamiento y rotación (IEC)	10-29
10.9	Operaciones de conversión (IEC)	10-32

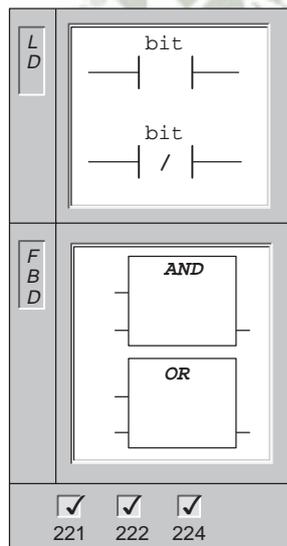
## 10.1 Operaciones lógicas con bits (IEC)

En la tabla 10-1 se indican las páginas donde se describen las operaciones lógicas con bits IEC no normalizadas a las que hace referencia el presente apartado.

Tabla 10-1 Operaciones lógicas con bits IEC no normalizadas

Descripción	Página
Contactos estándar	9-2
Contactos directos	9-3
Contacto NOT	9-4
Detectar flanco positivo y negativo	9-4
Asignar	9-6
Asignar directamente	9-6
Poner a 1 y Poner a 0 (bits N)	9-7

### Contactos estándar (IEC 1131-3 no normalizados)



El **Contacto normalmente abierto** se cierra (ON) si el valor binario de la dirección (bit) es igual a 1.

El **Contacto normalmente cerrado** se cierra (ON) si el valor binario de la dirección (bit) es igual a 0.

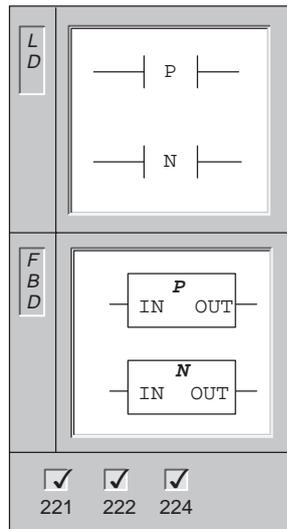
Estas operaciones leen el valor direccionado de la memoria o de la imagen del proceso si el tipo de memoria es I ó Q.

En LD, las operaciones Contacto normalmente abierto y Contacto normalmente cerrado se representan mediante contactos.

En FBD, los contactos normalmente abiertos se representan mediante cuadros AND/OR. Estas operaciones se pueden utilizar para manipular señales booleanas de la misma forma que los contactos LD. Los contactos normalmente cerrados también se representan mediante cuadros. Una operación Contacto normalmente cerrado se construye situando el símbolo de negación en la raíz de la señal de entrada.

Entradas/salidas	Operandos	Tipos de datos
Bit	I, Q, M, SM, T, C, V, S, L	BOOL
Entrada (FBD)	I, Q, M, SM, T, C, V, S, L, circulación de corriente	BOOL
Salida (FBD)	I, Q, M, SM, T, C, V, S, L, circulación de corriente	BOOL

### Transición positiva, Transición negativa



El contacto detector de **Transición positiva** permite que la corriente circule durante un ciclo cada vez que se produce un cambio de 0 a 1 (de "off" a "on").

El contacto detector de **Transición negativa** permite que la corriente circule durante un ciclo cada vez que se produce un cambio de 1 a 0 (de "on" a "off").

En LD, las operaciones Transición positiva y Transición negativa se representan mediante contactos.

En FBD, dichas operaciones se representan mediante los cuadros POS y NEG.

Entradas/salidas	Operandos	Tipos de datos
IN (FBD)	I, Q, M, SM, T, C, V, S, L, circulación de corriente	BOOL
OUT (FBD)	I, Q, M, SM, T, C, V, S, L, circulación de corriente	BOOL

### Ejemplos de operaciones con contactos

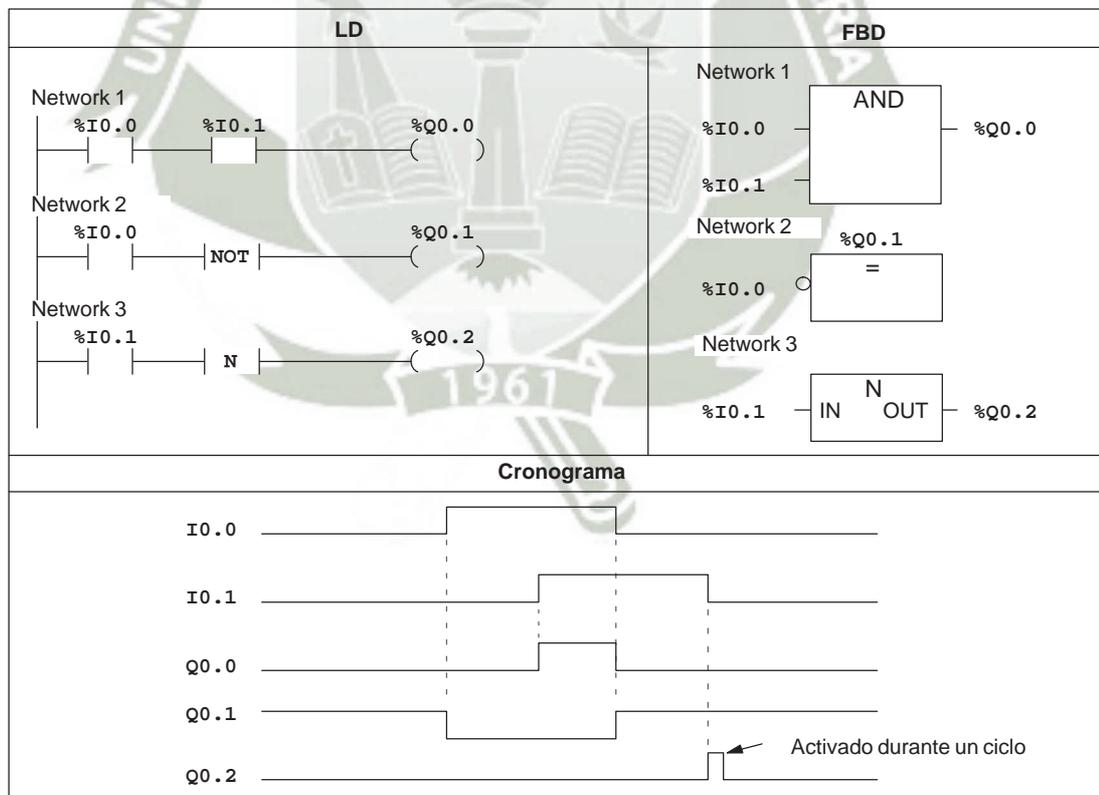
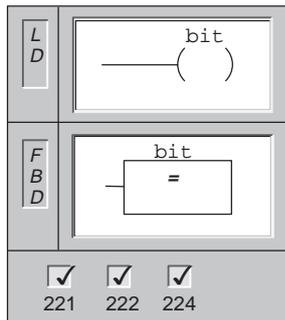


Figura 10-1 Ejemplos de operaciones lógicas con contactos en LD y FBD

### Bobina



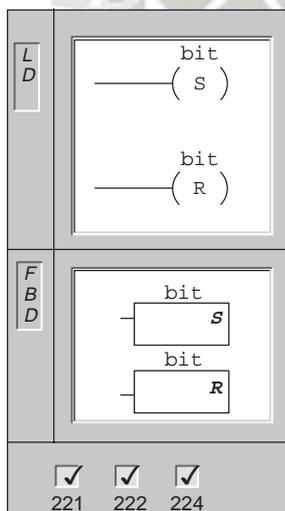
Al ejecutar la **Bobina** se activa la salida.

En LD, la operación Bobina se representa mediante una bobina.

En FBD, dicha operación se representa mediante el cuadro =.

Entradas/salidas	Operandos	Tipos de datos
Bit	I, Q, M, SM, T, C, V, S, L	BOOL

### SET, RESET



Cuando se ejecutan las operaciones **SET** y **RESET**, el valor indicado por OUT se activa o se desactiva, respectivamente.

Entradas/salidas	Operandos	Tipos de datos
Bit (LD, FBD)	I, Q, M, SM, T, C, V, S, L	BOOL

### Ejemplos de operaciones con salidas

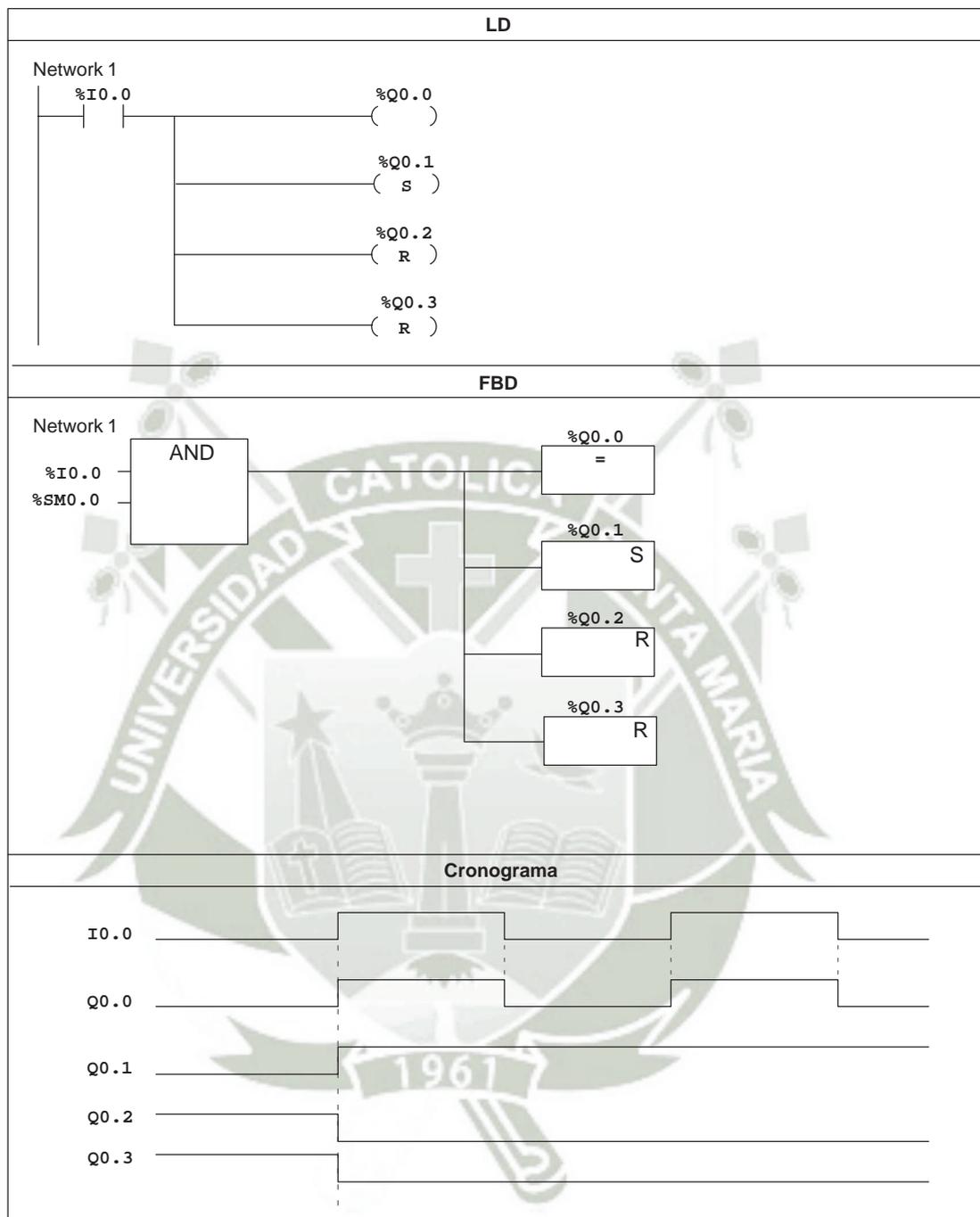
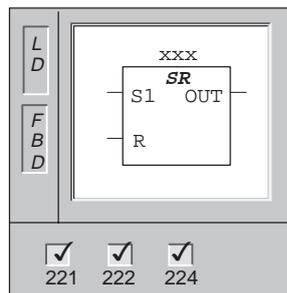


Figura 10-2 Ejemplos de operaciones con salidas en LD y FBD

### Bloque funcional biestable (posicionar dominante)

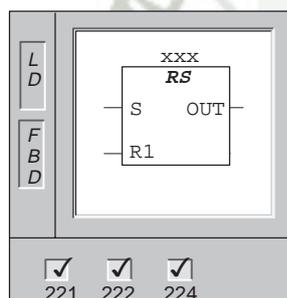


El **Bloque funcional biestable (posicionar dominante)** es un flip-flop en el que domina la señal "posicionar". Si tanto la señal "posicionar" (S1) como la señal "rearmar" (R) son verdaderas, la salida (OUT) será verdadera.

El parámetro xxx del bloque funcional especifica el parámetro booleano que está activado ("posicionado") o desactivado ("rearmado"). La salida opcional refleja el estado de señal del parámetro xxx.

Entradas/salidas	Operandos	Tipos de datos
S1, R (LD)	Circulación de corriente	BOOL
S1, R (FBD)	I, Q, M, SM, T, C,V, S, circulación de corriente	BOOL
OUT (LD)	Circulación de corriente	BOOL
OUT (FBD)	I, Q, M, SM, T, C, V, S, L, circulación de corriente	BOOL
xxx	I, Q, M, V, S	BOOL

### Bloque funcional biestable (rearmar dominante)



El **Bloque funcional biestable (rearmar dominante)** es un flip-flop en el que domina la señal "rearmar". Si tanto la señal "posicionar" (S) como la señal "rearmar" (R1) son verdaderas, la salida (OUT) será falsa.

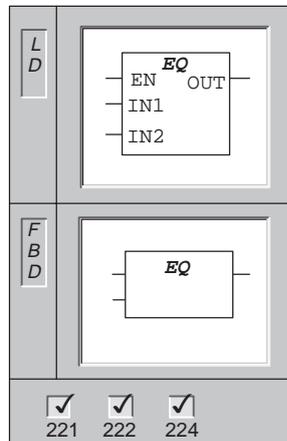
El parámetro xxx del bloque funcional especifica el parámetro booleano que está activado ("posicionado") o desactivado ("rearmado"). La salida opcional refleja el estado de señal del parámetro xxx.

Entradas/salidas	Operandos	Tipos de datos
S, R1 (LD)	Circulación de corriente	BOOL
S, R1 (FBD)	I, Q, M, SM, T, C, V, S, L, circulación de corriente	BOOL
OUT (LD)	Circulación de corriente	BOOL
OUT (FBD)	I, Q, M, SM, T, C, V, S, L, circulación de corriente	BOOL
xxx	I, Q, M, V, S	BOOL

## 10.2 Operaciones de comparación (IEC)

No existen operaciones de comparación IEC no normalizadas.

### Igualdad (EQ)

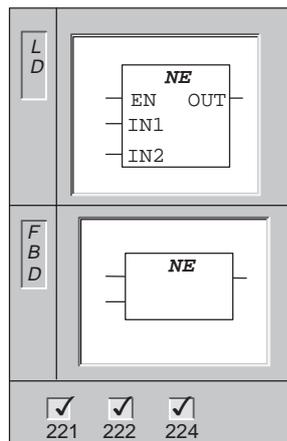


La función **Igualdad** (EQ) compara IN1 e IN2 con el resultado booleano depositado en OUT. Los tipos de datos de entrada y salida pueden variar pero deben ser del mismo tipo.

Las comparaciones de bytes no llevan signo. Las comparaciones de enteros, de enteros dobles y de reales llevan signo.

Entradas/salidas	Operandos	Tipos de datos
Entradas (LD y FBD)	IB, QB, MB, SB, SMB, VB, LB, IW, QW, MW, SW, SMW, VW, LW, T, C, AIW, ID, QD, MD, SD, SMD, VD, LD, HC, AC, constante, *VD, *AC, *LD	BYTE, INT, DINT, REAL
OUT (sólo LD)	Circulación de corriente	BOOL
OUT (sólo FBD)	I, Q, M, SM, T, C, V, S, L, circulación de corriente	BOOL

### Desigualdad (NE)

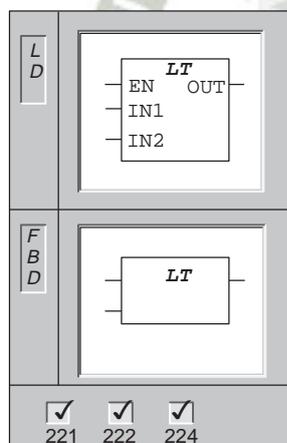


La función **Desigualdad** (NE) compara IN1 e IN2 con el resultado booleano depositado en OUT. Los tipos de datos de entrada y salida pueden variar pero deben ser del mismo tipo.

Las comparaciones de bytes no llevan signo. Las comparaciones de enteros, de enteros dobles y de reales llevan signo.

Entradas/salidas	Operandos	Tipos de datos
Entradas (LD y FBD)	IB, QB, MB, SB, SMB, VB, LB, IW, QW, MW, SW, SMW, VW, LW, T, C, AIW, ID, QD, MD, SD, SMD, VD, LD, HC, AC, constante, *VD, *AC, *LD	BYTE, INT, DINT, REAL
OUT (sólo LD)	Circulación de corriente	BOOL
OUT (sólo FBD)	I, Q, M, SM, T, C, V, S, L, circulación de corriente	BOOL

### Secuencia ascendente (LT)

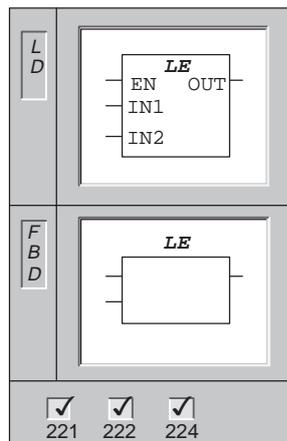


La función **Secuencia ascendente** (LT) compara  $IN1 < IN2$  con el resultado booleano depositado en OUT. Los tipos de datos de entrada y salida pueden variar pero deben ser del mismo tipo.

Las comparaciones de bytes no llevan signo. Las comparaciones de enteros, de enteros dobles y de reales llevan signo.

Entradas/salidas	Operandos	Tipos de datos
Entradas (LD y FBD)	IB, QB, MB, SB, SMB, VB, LB, IW, QW, MW, SW, SMW, VW, LW, T, C, AIW, ID, QD, MD, SD, SMD, VD, LD, HC, AC, constante, *VD, *AC, *LD	BYTE, INT, DINT, REAL
OUT (sólo LD)	Circulación de corriente	BOOL
OUT (sólo FBD)	I, Q, M, SM, V, S, L, circulación de corriente	BOOL

### Secuencia monótona (LE)

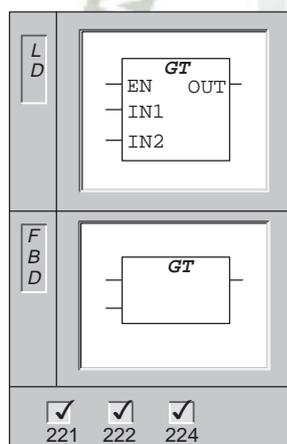


La función **Secuencia monótona (LE)** compara  $IN1 \leq IN2$  con el resultado booleano depositado en OUT. Los tipos de datos de entrada y salida pueden variar pero deben ser del mismo tipo.

Las comparaciones de bytes no llevan signo. Las comparaciones de enteros, de enteros dobles y de reales llevan signo.

Entradas/salidas	Operandos	Tipos de datos
Entradas (LD y FBD)	IB, QB, MB, SB, SMB, VB, LB, IW, QW, MW, SW, SMW, VW, LW, T, C, AIW, ID, QD, MD, SD, SMD, VD, LD, HC, AC, constante, *VD, *AC, *LD	BYTE, INT, DINT, REAL
OUT (sólo LD)	Circulación de corriente	BOOL
OUT (sólo FBD)	I, Q, M, SM, V, S, L, circulación de corriente	BOOL

### Secuencia decreciente (GT)

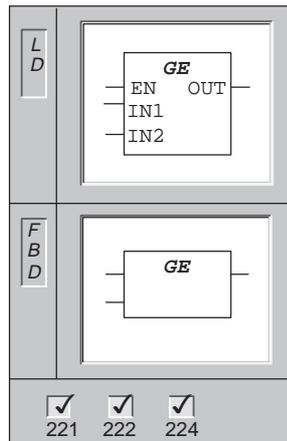


La función **Secuencia decreciente (GT)** compara  $IN1 > IN2$  con el resultado booleano depositado en OUT. Los tipos de datos de entrada y salida pueden variar pero deben ser del mismo tipo.

Las comparaciones de bytes no llevan signo. Las comparaciones de enteros, de enteros dobles y de reales llevan signo.

Entradas/salidas	Operandos	Tipos de datos
Entradas (LD y FBD)	IB, QB, MB, SB, SMB, VB, LB, IW, QW, MW, SW, SMW, VW, LW, T, C, AIW, ID, QD, MD, SD, SMD, VD, LD, HC, AC, constante, *VD, *AC, *LD	BYTE, INT, DINT, REAL
OUT (sólo LD)	Circulación de corriente	BOOL
OUT (sólo FBD)	I, Q, M, SM, V, S, L, circulación de corriente	BOOL

### Secuencia monótona (GE)



La función **Secuencia monótona (GE)** compara  $IN1 \geq IN2$  con el resultado booleano depositado en OUT. Los tipos de datos de entrada y salida pueden variar pero deben ser del mismo tipo.

Las comparaciones de bytes no llevan signo. Las comparaciones de enteros, de enteros dobles y de reales llevan signo.

Entradas/salidas	Operandos	Tipos de datos
Entradas (LD y FBD)	IB, QB, MB, SB, SMB, VB, LB, IW, QW, MW, SW, SMW, VW, LW, T, C, AIW, ID, QD, MD, SD, SMD, VD, LD, HC, AC, constante, *VD, *AC, *LD	BYTE, INT, DINT, REAL
OUT (sólo LD)	Circulación de corriente	BOOL
OUT (sólo FBD)	I, Q, M, SM, V, S, L, circulación de corriente	BOOL

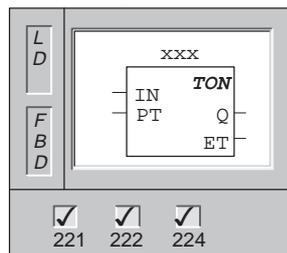
### 10.3 Operaciones de temporización (IEC)

En la tabla 10-2 se indican las páginas donde se describen las operaciones de temporización (IEC) no normalizadas a las que hace referencia el presente apartado.

Tabla 10-2 Operaciones de temporización (IEC) no normalizadas

Descripción	Página
Temporizador de retardo a la conexión	9-15

#### Temporizador con retardo al conectar

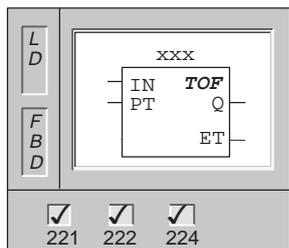


El bloque funcional **Temporizador con retardo al conectar** temporiza hasta el valor prefijado cuando la entrada de habilitación (IN) cambia a "verdadero". Si el tiempo transcurrido (ET) es mayor que o igual al tiempo prefijado (PT), se activará el bit de salida del temporizador (Q).

El bit de salida se desactivará cuando la entrada de habilitación cambie a "falso". Cuando se alcanza el tiempo prefijado (PT), la temporización se detiene y el temporizador se inhibe.

Entradas/salidas	Operandos	Tipos de datos
IN (LD)	Circulación de corriente	BOOL
IN (FBD)	I, Q, M, SM, T, C, V, S, L, circulación de corriente	BOOL
PT (LD y FBD)	VW, IW, QW, MW, SMW, LW, SW, AIW, AC, constante, *VD, *AC, *LD	INT
Q (LD y FBD)	I, Q, M, SM, V, S, L	BOOL
ET (LD y FBD)	VW, IW, QW, MW, SMW, LW, SW, AQW, AC, *VD, *AC, *LD	INT
xxx	consulte la tabla 10-3	TON

### Temporizador con retardo al desconectar



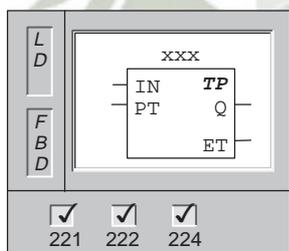
El bloque funcional **Temporizador con retardo al desconectar** se utiliza para retardar el estado "falso" de una salida durante un período determinado tras haber adoptado la entrada el estado "falso". Temporiza hasta el valor predefinido cuando la entrada de habilitación (IN) cambio a "falso". Si el tiempo transcurrido (ET) es mayor que o igual al tiempo prefijado (PT), se activará el bit de salida del temporizador (Q).

Una vez alcanzado el valor prefijado, el bit de salida del temporizador cambia a "falso" y el tiempo transcurrido se mantiene hasta que la entrada de habilitación (IN) cambia a "verdadero". Si la entrada de habilitación (IN) cambia a "falso" durante un período inferior al tiempo prefijado (PT), el bit de salida seguirá siendo "verdadero".

Para obtener más información sobre los números y las resoluciones de los temporizadores, consulte la tabla 10-3.

Entradas/salidas	Operandos	Tipos de datos
IN (LD)	Circulación de corriente	BOOL
IN (FBD)	I, Q, M, SM, T, C, V, S, L, circulación de corriente	BOOL
PT (LD y FBD)	VW, IW, QW, MW, SMW, LW, SW, AIW, AC, constante, *VD, *AC, *LD	INT
Q (LD y FBD)	I, Q, M, SM, V, S, L	BOOL
ET (LD y FBD)	VW, IW, QW, MW, SMW, LW, SW, AQW, AC, *VD, *AC, *LD	INT
xxx	consulte la tabla 10-3	TOF

### Temporizador por impulsos



El bloque funcional **Temporizador por impulsos** se utiliza para generar impulsos de una duración determinada. Cuando el estado de señal de la entrada de habilitación (IN) cambia a "verdadero", se activa el bit de salida (Q). Éste último sigue siendo "verdadero" durante el impulso especificado en el tiempo prefijado (PT). Cuando el tiempo transcurrido (ET) alcanza el valor del tiempo prefijado (PT), el estado de señal del bit de salida (Q) cambia a "falso".

Para obtener más información sobre los números y las resoluciones de los temporizadores, consulte la tabla 10-3.

Entradas/salidas	Operandos	Tipos de datos
IN (LD)	Circulación de corriente	BOOL
IN (FBD)	I, Q, M, SM, T, C, V, S, L, circulación de corriente	BOOL
PT (LD y FBD)	VW, IW, QW, MW, SMW, LW, SW, AIW, AC, constante, *VD, *AC, *LD	INT
Q (LD y FBD)	I, Q, M, SM, S, V, L	BOOL
ET (LD y FBD)	VW, IW, QW, MW, SW, LW, AQW, AC, *VD, *AC, *LD	INT
xxx	consulte la tabla 10-3	TP

### Operaciones de temporización IEC 1131-3

Se dispone de temporizadores TON, TOF y TP con tres resoluciones. La resolución viene determinada por el número de temporizador (v. tabla 10-3). El valor actual resulta del valor de conteaje multiplicado por la base de tiempo. Por ejemplo, el valor de conteaje 50 en un temporizador de 10 ms equivale a 500 ms.

Tabla 10-3 Temporizadores y sus resoluciones

Tipo de temporizador	Resolución en milisegundos (ms)	Valor máximo en segundos (s)	Nº de temporizador
TON, TOF, TP	1 ms	32.767 s	T32, T96
	10 ms	327.67 s	T33 a T36, T97 a T100
	100 ms	3276.7 s	T37 a T63, T101 a T255

#### Nota

No se pueden compartir números iguales para los temporizadores TOF, TP y TON. Por ejemplo, no puede haber tanto un TON T32 como un TOF T32.

### Ejemplo de un temporizador de retardo al conectar

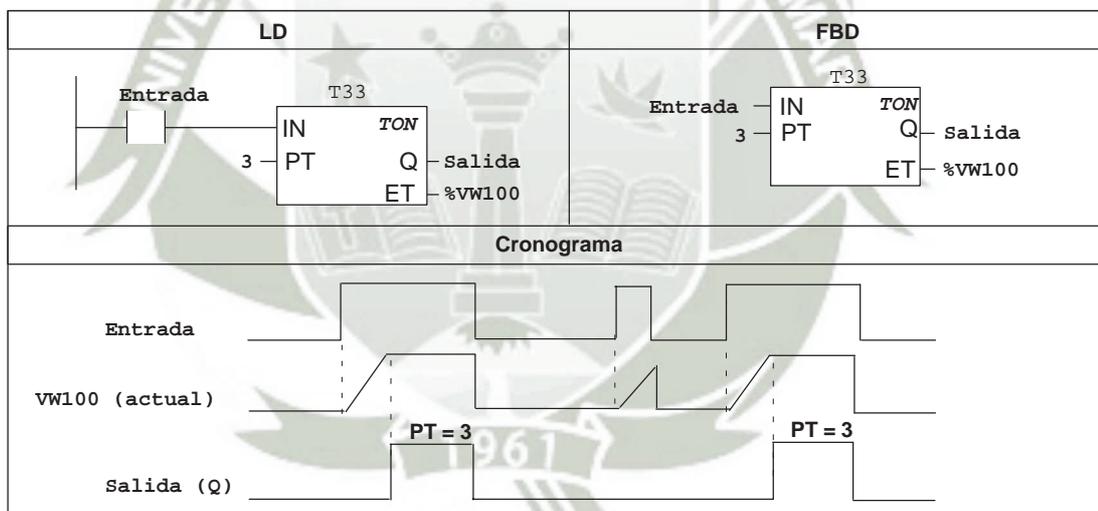


Figura 10-3 Ejemplo de un temporizador de retardo al conectar en LD y FBD

**Ejemplo de un temporizador de retardo al desconectar**

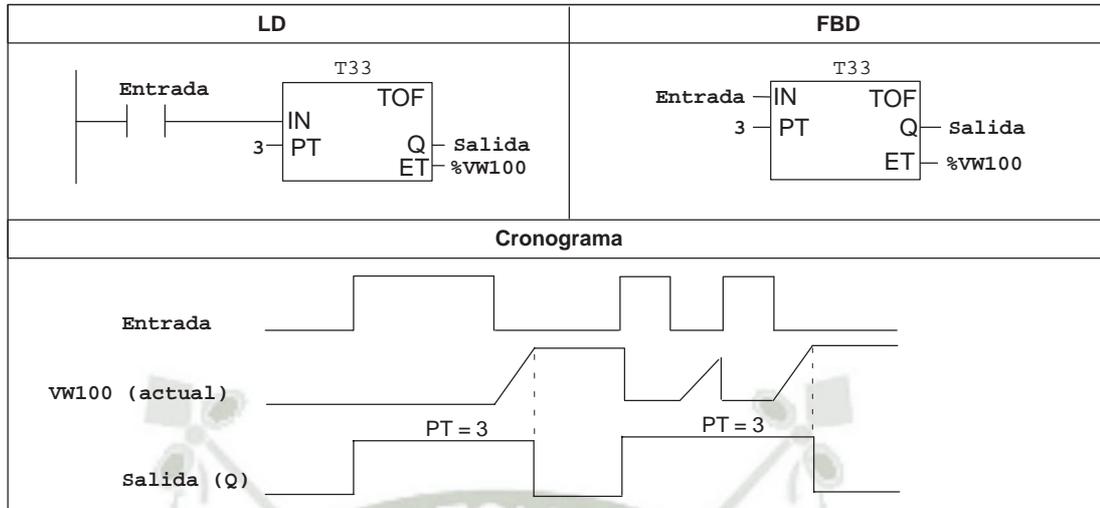


Figura 10-4 Ejemplo de un temporizador de retardo al desconectar en LD y FBD

**Ejemplo de un temporizador por impulsos**

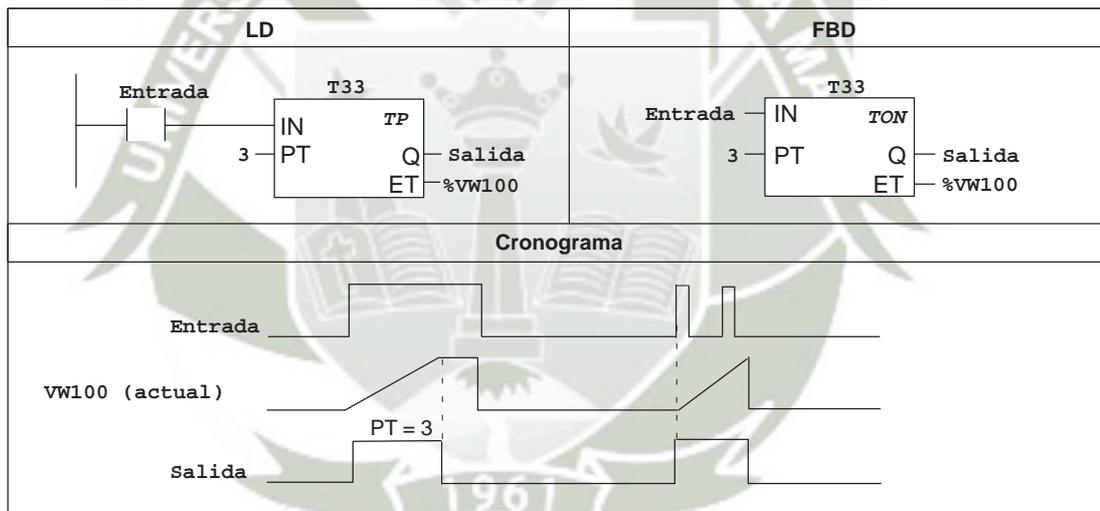


Figura 10-5 Ejemplo de un temporizador por impulsos en LD y FBD

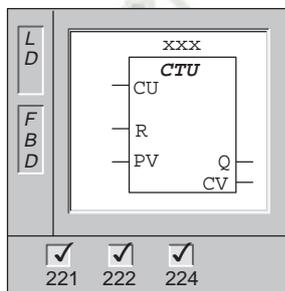
## 10.4 Operaciones con contadores (IEC)

En la tabla 10-4 se indican las páginas donde se describen las operaciones con contadores (IEC) no normalizadas a las que hace referencia el presente apartado.

Tabla 10-4 Operaciones con contadores (IEC) no normalizadas

Descripción	Página
Activar contador rápido	9-27
Definir modo para contador rápido	9-27
Salida de impulsos	9-49

### Contador ascendente



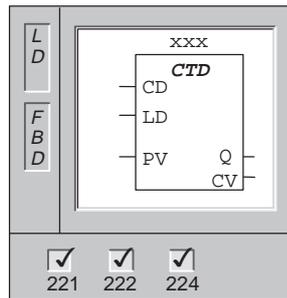
El bloque funcional **Contador ascendente** cuenta adelante desde el valor actual hasta el valor prefijado al producirse un flanco positivo en la entrada de conteo adelante (CU). Si el valor actual (CV) es mayor o igual al valor prefijado (PV), se activa el bit de conteo (Q). El contador se inicializa al activarse la entrada de desactivación (R). El contador ascendente detiene el conteo al alcanzar el valor prefijado (PV).

#### Nota

Puesto que cada contador dispone sólo de un valor actual, no se podrá asignar un mismo número a varios contadores. (Los contadores ascendentes, descendentes y ascendentes-descendentes acceden a un mismo valor actual).

Entradas/salidas	Operandos	Tipos de datos
CU (sólo FBD)	I, Q, M, SM, V, S, L, T, C, circulación de corriente	BOOL
R (sólo FBD)	I, Q, M, SM, V, S, L, T, C, circulación de corriente	BOOL
PV (LD y FBD)	VW, IW, QW, MW, SMW, LW, SW, AIW, AC, constante, *VD, *AC, *LD	INT
Q (LD y FBD)	I, Q, M, SM, V, S, L	BOOL
CV (LD y FBD)	VW, IW, QW, MW, SW, SMW, LW, AC, *VD, *AC, *LD	INT
xxx	C0 hasta C255	CTU

### Contador descendente



El bloque funcional **Contador descendente** cuenta hacia atrás desde el valor prefijado al producirse un flanco positivo en la entrada de conteo atrás (CD). Si el valor actual (CV) es igual a cero, se activa el bit de salida del contador (Q). El contador se inicializa y carga el valor actual (CV) en el valor prefijado (PV) cuando se habilita la entrada de carga (LD). El contador descendente detiene el conteo cuando alcanza el valor cero.

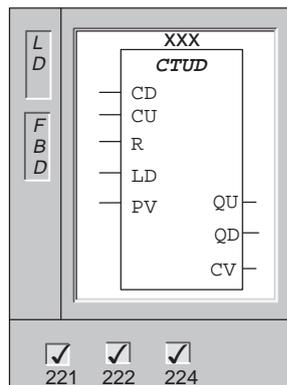
#### Nota

Puesto que cada contador dispone sólo de un valor actual, no se podrá asignar un mismo número a varios contadores. (Los contadores ascendentes, descendentes y ascendentes-descendentes acceden a un mismo valor actual).

Tabla 10-5 Operandos y tipos de datos del contador descendente

Entradas/salidas	Operandos	Tipos de datos
CD (FBD)	I, Q, M, SM, V, S, L, T, C, circulación de corriente	BOOL
LD (FBD)	I, Q, M, SM, V, S, L, T, C, circulación de corriente	BOOL
PV (LD, FBD)	VW, IW, QW, MW, SMW, LW, SW, AIW, AC, constante, *VD, *AC, *LD	INT
Q (LD y FBD)	I, Q, M, SM, V, S, L	BOOL
CV (LD y FBD)	VW, IW, QW, MW, SW, LW, AC, *VD, *AC, *LD	INT
xxx	C0 hasta C255	CTD

### Contador ascendente-descendente



El bloque funcional **Contador ascendente-descendente** cuenta adelante o atrás desde el valor prefijado al producirse un flanco positivo en la entrada de conteo adelante (CU) o de conteo atrás (CD), respectivamente. La salida (QU) se activa cuando el valor actual (CV) es igual al valor prefijado. La salida (QD) se activa cuando el valor actual (CV) es igual a cero. El contador carga el valor actual (CV) en el valor prefijado (PV) cuando se habilita la entrada de carga (LD). De forma similar, el contador se inicializa y carga el valor actual (CV) con cero cuando se habilita la desactivación (R). El contador detiene el conteo cuando alcanza el valor prefijado, o bien cero.

#### Nota

Puesto que cada contador dispone sólo de un valor actual, no se podrá asignar un mismo número a varios contadores. (Los contadores ascendentes, descendentes y ascendentes-descendentes acceden a un mismo valor actual).

Entradas/salidas	Operandos	Tipos de datos
CD (sólo FBD)	I, Q, M, SM, V, S, L, T, C, circulación de corriente	BOOL
CU (sólo FBD)	I, Q, M, SM, V, S, L, T, C, circulación de corriente	BOOL
R (sólo FBD)	I, Q, M, SM, V, S, L, T, C, circulación de corriente	BOOL
LD (sólo FBD)	I, Q, M, SM, V, S, L, T, C, circulación de corriente	BOOL
PV (LD y FBD)	VW, IW, QW, MW, SMW, LW, SW, AIW, AC, constante, *VD, *AC, *LD	INT
QU (LD y FBD)	I, Q, M, SM, V, S, L	BOOL
QD (LD y FBD)	I, Q, M, SM, V, S, L	BOOL
CV (LD y FBD)	VW, T, C, IW, QW, MW, SW, LW, AC, *VD, *AC, *LD	INT
xxx	C0 hasta C255	CTUD

Ejemplo de una operación de contaje

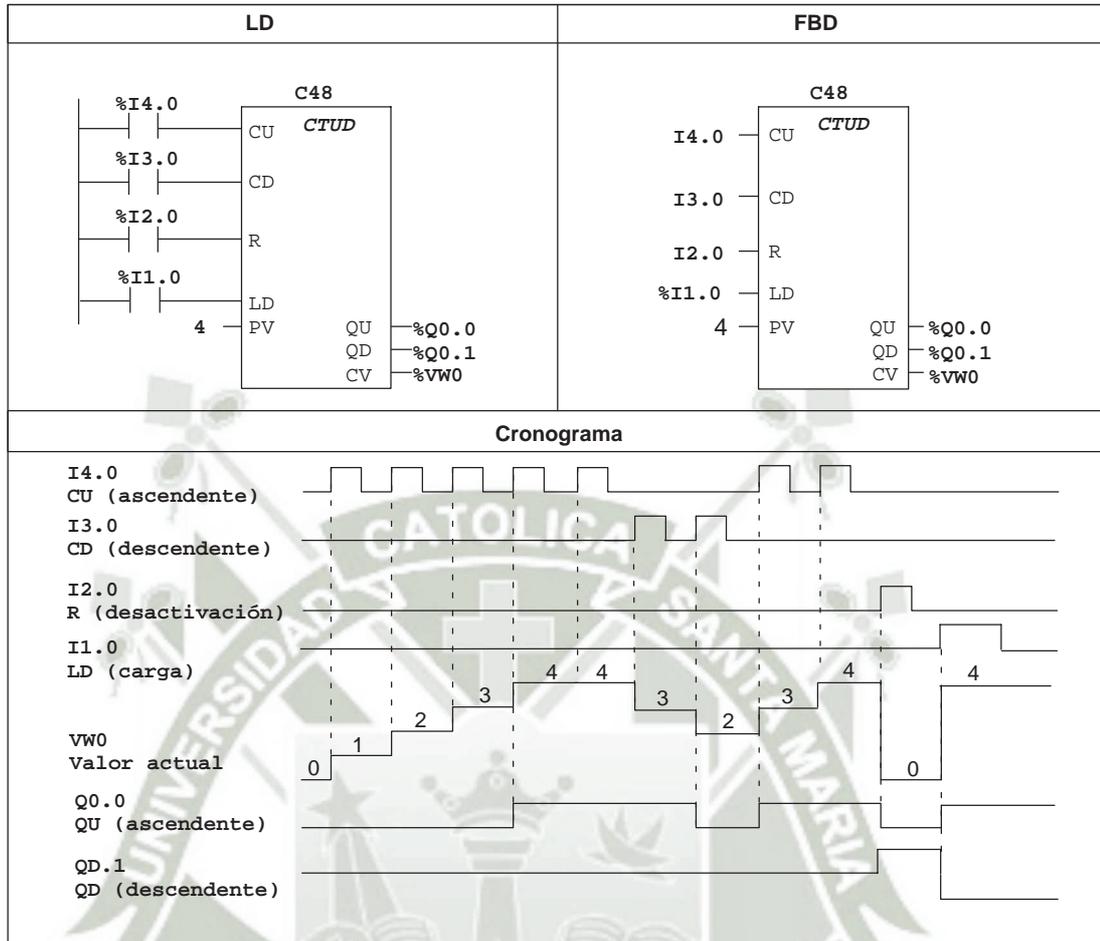


Figura 10-6 Ejemplo de una operación de contaje en LD y FBD

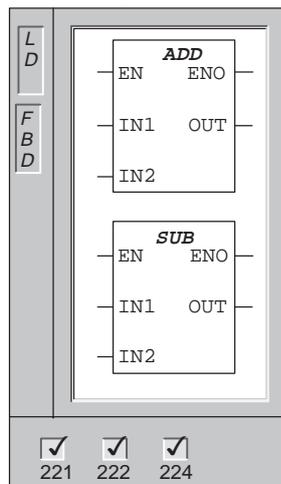
## 10.5 Operaciones aritméticas (IEC)

En la tabla 10-6 se indican las páginas donde se describen las operaciones aritméticas (IEC) no normalizadas a las que hace referencia el presente apartado.

Tabla 10-6 Operaciones aritméticas (IEC) no normalizadas

Descripción	Página
Operación PID	9-84

### Sumar y restar



Las funciones **Sumar** y **Restar** suman o restan IN1 e IN2 y depositan el resultado en OUT. Los tipos de datos de entrada y salida pueden variar pero deben ser del mismo tipo. Por ejemplo, dos variables de 16 bits se pueden sumar o restar, pero el resultado se debe depositar en una variable de 16 bits. El resultado de una suma o de una resta de dos variables de 32 bits se debe depositar en una variable de 32 bits.

En LD:  $IN1 + IN2 = OUT$   
 $IN1 - IN2 = OUT$

Condiciones de error que ponen ENO a 0: SM1.1 (desbordamiento), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

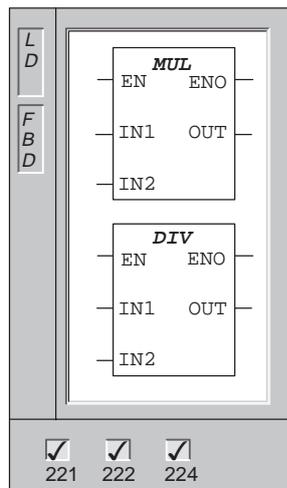
Estas funciones afectan a las siguientes marcas especiales: SM1.0 (cero); SM1.1 (desbordamiento); SM1.2 (negativo)

Entradas/salidas	Operandos	Tipos de datos
IN1, IN2	VW, IW, QW, MW, SW, SMW, LW, AIW, T, C, VD, ID, QD, MD, SMD, SD, LD, HC, AC, constante, *VD, *AC, *LD	INT, DINT, REAL
OUT	VW, IW, QW, MW, SW, SMW, LW, T, C, VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *AC, *LD	INT, DINT, REAL

#### Nota

Los números reales (o números en coma flotante) se representan en el formato descrito en la norma ANSI/IEEE 754-1985 (precisión sencilla). Para obtener más información al respecto, consulte dicha norma.

## Multiplicar y dividir



La función **Multiplicar** (MUL) multiplica IN1 por IN2 y deposita el resultado en la variable indicada por OUT.

La función **Dividir** (DIV) divide IN1 por IN2 y deposita el resultado en la variable indicada por OUT.

Los tipos de datos de entrada y salida pueden variar pero deben ser del mismo tipo. Por ejemplo, el producto de dos variables de 16 bits se debe depositar en una variable de 16 bits. El producto de dos variables de 32 bits se debe depositar en una variable de 32 bits.

En LD:  $IN1 * IN2 = OUT$   
 $IN1 / IN2 = OUT$

Condiciones de error que ponen ENO a 0: SM1.1 (desbordamiento), SM1.3 (división por cero), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Estas funciones afectan a las siguientes marcas especiales: SM1.0 (cero); SM1.1 (desbordamiento); SM1.2 (negativo); SM1.3 (división por cero)

Si SM1.1 (marca de desbordamiento) está activada, los demás bits de estado aritméticos se borrarán y el operando de salida no se alterará. En el caso de operaciones con enteros, si SM1.3 se activa durante una operación de división, permanecerán inalterados los demás bits aritméticos de estado, así como los operandos de entrada originales. En otro caso, todos los bits aritméticos de estado asistirán el estado válido al finalizar la operación aritmética.

Entradas/salidas	Operandos	Tipos de datos
IN1, IN2	VW, IW, QW, MW, SW, SMW, LW, AIW, T, C, VD, ID, QD, MD, SMD, SD, LD, HC, AC, constante, *VD, *AC, *LD	INT, DINT, REAL
OUT	VW, IW, QW, MW, SW, SMW, T, C, LW, VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *AC, *LD	INT, DINT, REAL

### Nota

Los números reales (o números en coma flotante) se representan en el formato descrito en la norma ANSI/IEEE 754-1985 (precisión sencilla). Para obtener más información al respecto, consulte dicha norma.

Ejemplos de operaciones aritméticas

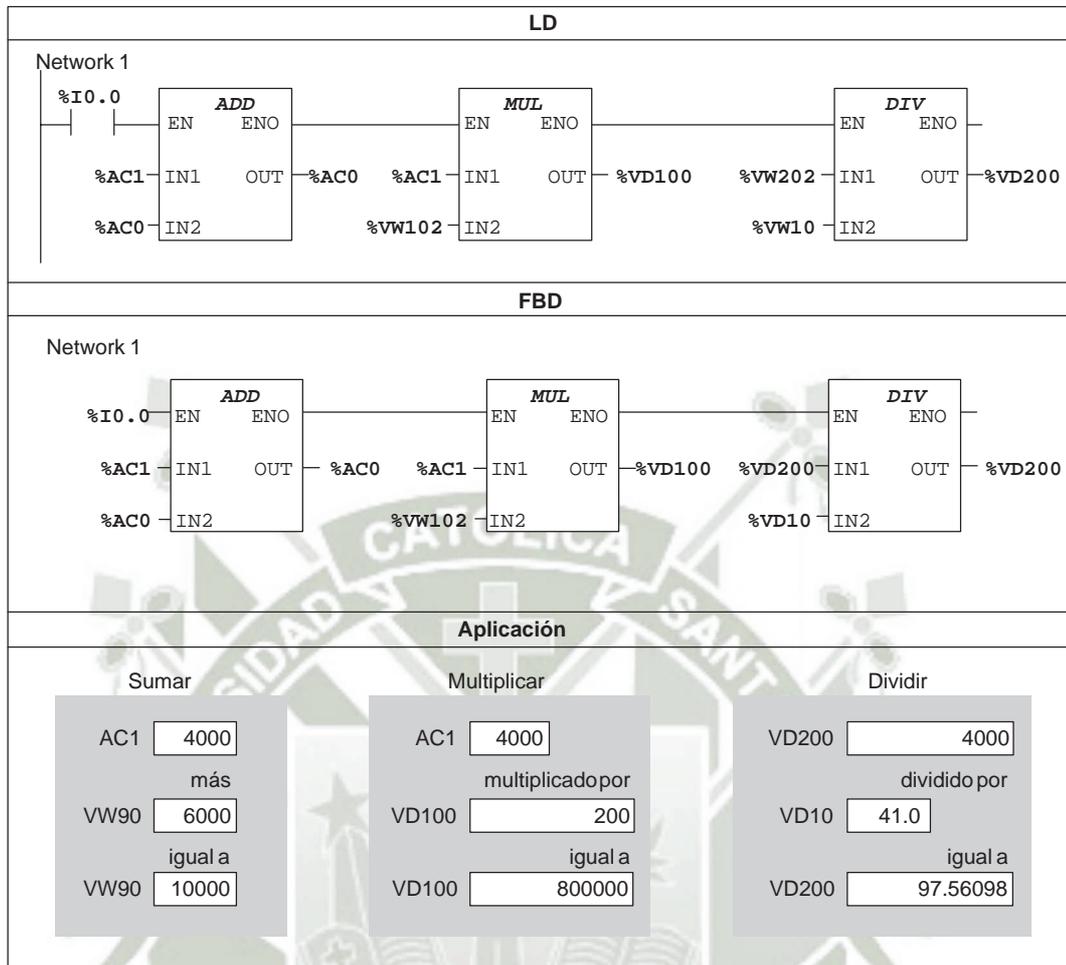
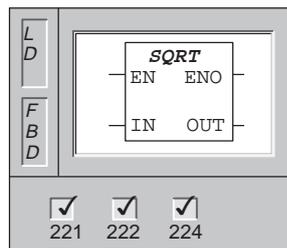


Figura 10-7 Ejemplos de operaciones aritméticas en LD y FBD

### Raíz cuadrada



La función **Raíz cuadrada** saca la raíz cuadrada de un valor especificado por IN y deposita el resultado en OUT.

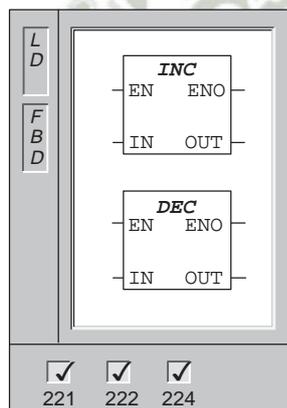
Condiciones de error que ponen ENO a 0: SM1.1 (desbordamiento), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Esta función afecta a las siguientes marcas especiales: SM1.0 (cero); SM1.1 (desbordamiento); SM1.2 (negativo)

Si SM1.1 (marca de desbordamiento) está activada, los demás bits de estado aritméticos se borrarán y el operando de salida no se alterará.

Entradas/salidas	Operandos	Tipos de datos
IN	VD, ID, QD, MD, SMD, SD, LD, AC, constante, *VD, *AC, *LD	REAL
OUT	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *AC, *LD	REAL

### Incrementar, Decrementar



Las funciones **Incrementar** y **Decrementar** suman/restan 1 a IN y depositan el resultado en OUT.

Las funciones Incrementar byte y Decrementar byte no llevan signo.

Condiciones de error que ponen ENO a 0: SM1.1 (desbordamiento), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Estas funciones afectan a las siguientes marcas especiales: SM1.0 (cero); SM1.1 (desbordamiento), SM1.2 (negativo)

Entradas/salidas	Operandos	Tipos de datos
IN	VB, IB, QB, MB, SB, SMB, LB, VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, VD, ID, QD, MD, SD, SMD, LD, HC, AC, constante, *VD, *AC, *LD	BYTE, INT, DINT
OUT	VB, IB, QB, MB, SB, SMB, LB, VW, IW, QW, MW, SW, SMW, T, C, LW, VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *AC, *LD	BYTE, INT, DINT

Ejemplo de las operaciones Incrementar y Decrementar

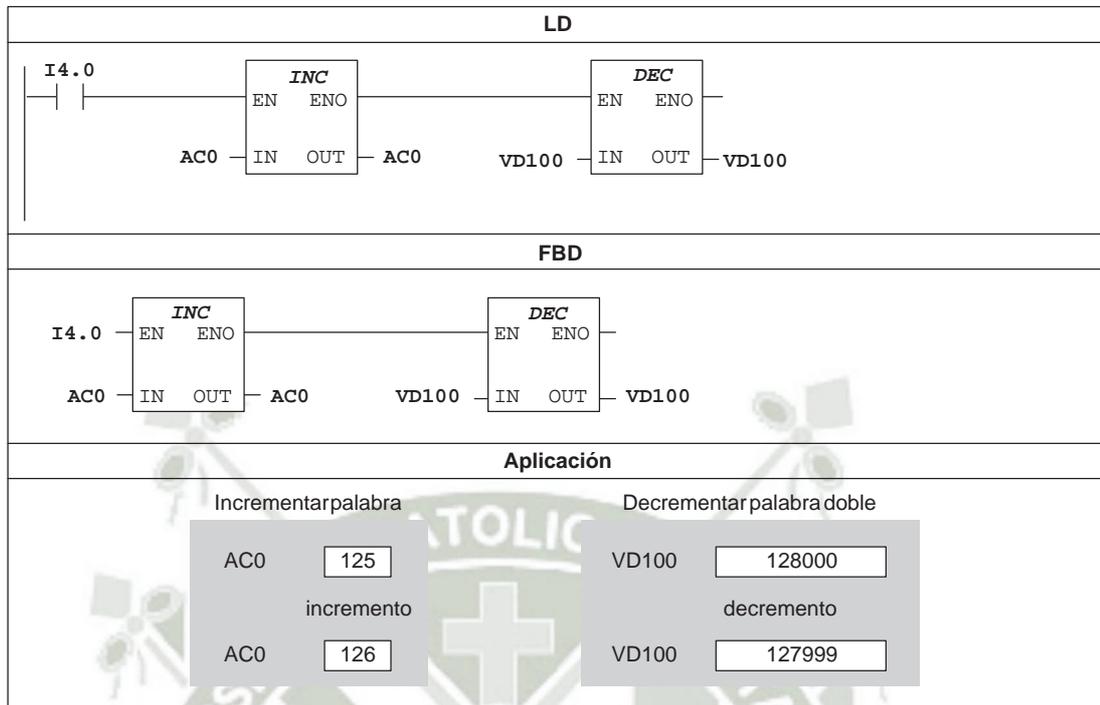


Figura 10-8 Ejemplos de las funciones Incrementar y Decrementar en LD y FBD

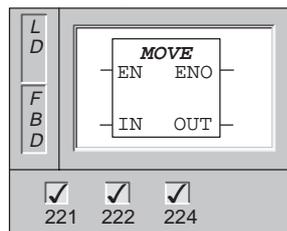
## 10.6 Operaciones de transferencia (IEC)

En la tabla 10-7 se indican las páginas donde se describen las operaciones de transferencia (IEC) no normalizadas a las que hace referencia el presente apartado.

Tabla 10-7 Operaciones de transferencia (IEC) no normalizadas

Descripción	Página
Operaciones de invertir	9-102

### Transferir (MOVE)



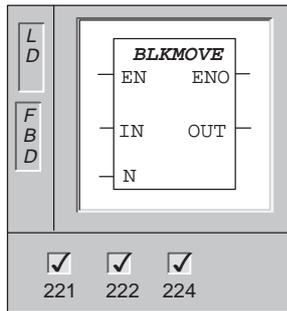
La función **Transferir y asignar valores** transfiere el valor IN a la dirección OUT. Esta función ejecuta una operación de asignación. El parámetro de entrada no se modifica durante la ejecución.

Los tipos de datos de entrada y salida pueden variar pero deben ser del mismo tipo.

Condiciones de error que ponen ENO a 0: SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Entradas/salidas	Operandos	Tipos de datos
IN	VB, IB, QB, MB, SB, SMB, LB, VW, IW, QW, MW, SM, SMW, LW, T, C, AIW, VD, ID, QD, MD, SMD, SD, LD, HC, &VB, &IB, &QB, &MB, &SB, AC, constante, *VD, *AC, *LD	BYTE, WORD, INT, DWORD, DINT, REAL
OUT	VB, IB, QB, MB, SB, SMB, LB, VW, IW, QW, MW, SW, SMW, LW, T, C, AQW, VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *AC, *LD	BYTE, WORD, INT, DWORD, DINT, REAL

### Transferir en bloque



La operación **Transferir en bloque** transfiere un número determinado de palabras (N) indicado por la dirección IN a la dirección OUT. N puede estar comprendido entre 1 y 255.

Los tipos de datos de entrada y salida pueden variar pero deben ser del mismo tipo.

La función Transferir en bloque es una función IEC no normalizada.

Condiciones de error que ponen ENO a 0: SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto), 0091 (operando fuera de área)

Entradas/salidas	Operandos	Tipos de datos
IN	VB, IB, QB, MB, SB, SMB, LB, VW, IW, QW, MW, SM, SMW, LW, T, C, AIW, VD, ID, QD, MD, SMD, SD, LD, HC, &VB, &IB, &QB, &MB, &SB, AC, constante, *VD, *AC, *LD	BYTE, WORD, DWORD
OUT	VB, IB, QB, MB, SB, SMB, LB, VW, IW, QW, MW, SW, SMW, LW, T, C, AQW, VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *AC, *LD	BYTE, WORD, DWORD
N	VB, IB, QB, MB, SB, SMB, LB, AC, constante, *VD, *AC, *LD	Byte

### Ejemplos de operaciones de transferir

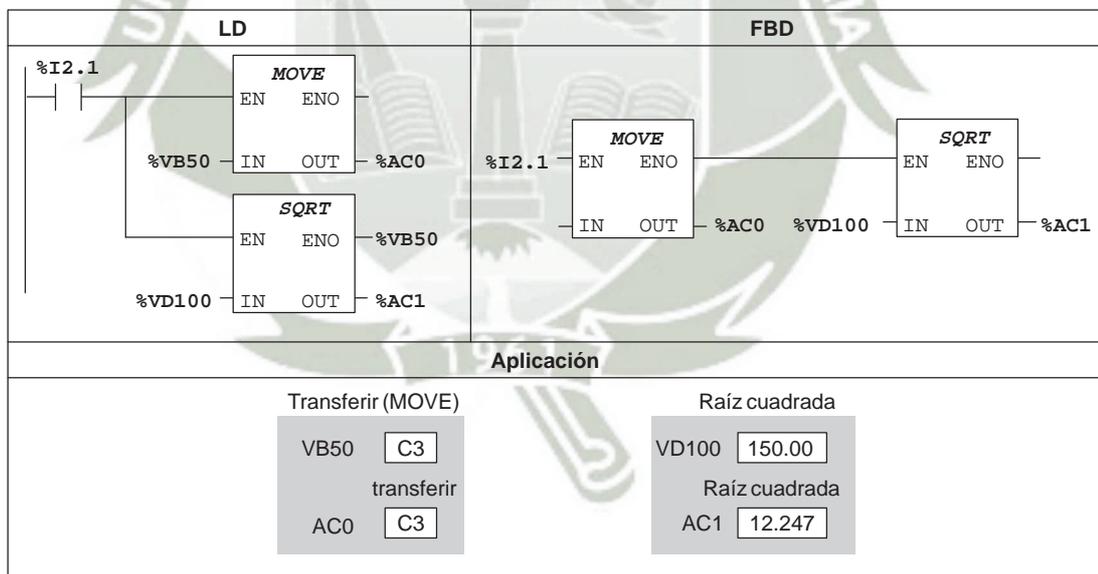
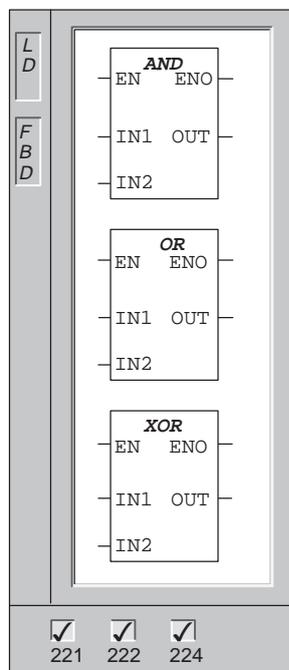


Figura 10-9 Ejemplo de una operación de transferencia en LD y FBD

## 10.7 Operaciones lógicas (IEC)

No existen operaciones lógicas IEC no normalizadas.

### AND, OR, XOR



La función **AND** combina mediante Y los bits correspondientes de IN1 e IN2 y carga el resultado en OUT.

La función **OR** combina mediante O los bits correspondientes de IN1 e IN2 y carga el resultado en OUT.

La función **XOR** combina mediante O-exclusiva los bits correspondientes de IN1 e IN2 y carga el resultado en OUT.

Los tipos de datos de entrada y salida pueden variar pero deben ser del mismo tipo.

Condiciones de error que ponen ENO a 0: SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero)

Entradas/salidas	Operandos	Tipos de datos
IN1, IN2	VB, IB, QB, MB, SB, SMB, LB, VW, IW, QW, MW, SW, SMW, AIW, T, C, LW, VD, ID, QD, MD, SD, SMD, LD, HC, AC, constante, *VD, *AC, *LD	BYTE, WORD DWORD
OUT	VB, IB, QB, MB, SB, SMB, LB, VW, IW, QW, MW, SW, SMW, T, C, LW, VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *AC, *LD	BYTE, WORD DWORD

Ejemplo de las operaciones AND, OR y XOR

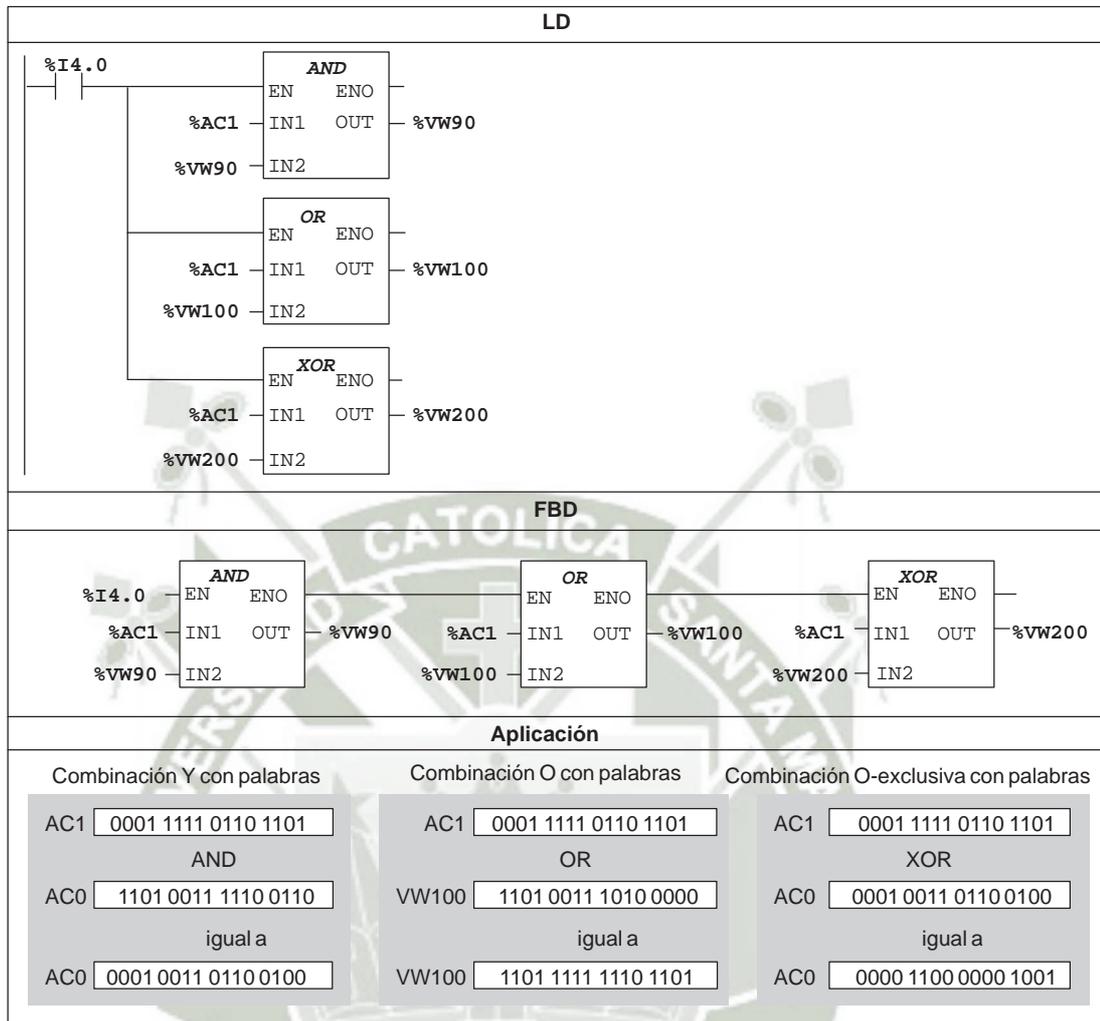
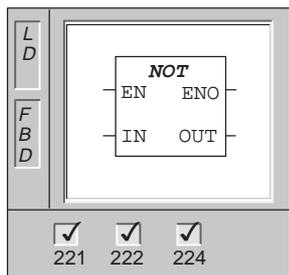


Figura 10-10 Ejemplo de las operaciones AND, OR y XOR

**NOT**



La función **NOT** invierte los bits correspondientes de IN y carga el resultado en OUT.

Los tipos de datos de entrada y salida pueden variar pero deben ser del mismo tipo.

Condiciones de error que ponen ENO a 0: SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero)

Entradas/salidas	Operandos	Tipos de datos
IN	VB, IB, QB, MB, SB, SMB, LB, VW, IW, QW, MW, SW, SMW, AIW, T, C, LW, VD, ID, QD, MD, SD, SMD, LD, HC, AC, constante, *VD, *AC, *LD	BYTE, WORD DWORD
OUT	VB, IB, QB, MB, SB, SMB, LB, VW, IW, QW, MW, SW, SMW, T, C, LW, VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *AC, *LD	BYTE, WORD DWORD



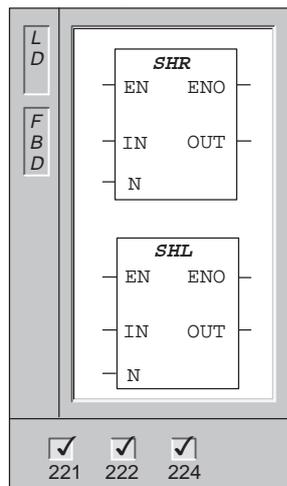
## 10.8 Operaciones de desplazamiento y rotación (IEC)

En la tabla 10-8 se indican las páginas donde se describen las operaciones de desplazamiento (IEC) no normalizadas a las que hace referencia el presente apartado.

Tabla 10-8 Operaciones IEC no normalizadas

Descripción	Página
Registro de desplazamiento	9-123

### Desplazar a la derecha, Desplazar a la izquierda



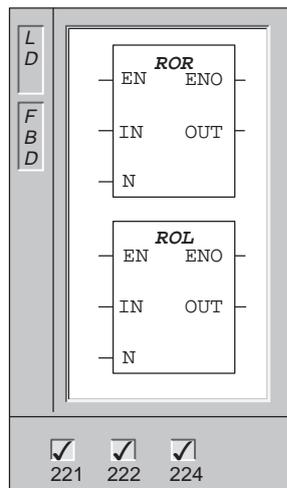
La función **Desplazar a la derecha** desplaza el valor indicado por la variable IN hacia la derecha tantas posiciones como indique N. El resultado se deposita en la variable indicada por OUT. Cada bit se rellena con un cero cuando es desplazado hacia la derecha.

La función **Desplazar a la izquierda** desplaza el valor indicado por la variable IN hacia la izquierda tantas posiciones como indique N. El resultado se deposita en la variable indicada por OUT. Cada bit se rellena con un cero cuando es desplazado hacia la izquierda.

Condiciones de error que ponen ENO a 0: SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Entradas/salidas	Operandos	Tipos de datos
IN	VB, IB, QB, MB, SB, SMB, LB, VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, VD, ID, QD, MD, SD, SMD, LD, HC, AC, constante, *VD, *LD, *AC	BYTE, WORD DWORD
N	VB, IB, QB, MB, SB, SMB, LB, AC, constante, *VD, *LD, *AC	Byte
OUT	VB, IB, QB, MB, SB, SMB, LB, VW, IW, QW, MW, SW, SMW, LW, T, C, VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *LD, *AC	BYTE, WORD DWORD

### Rotar a la derecha, Rotar a la izquierda



Las operaciones **Rotar a la derecha** y **Rotar a la izquierda** rotan el valor de la entrada (IN) a la derecha y a la izquierda respectivamente, tantas posiciones como indique el valor de desplazamiento (N) y cargan el resultado en la salida (OUT).

La rotación es circular. En ROR, el bit cero se rota al bit más significativo. En ROL, el bit más significativo se rota al bit cero.

Condiciones de error que ponen ENO a 0: SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Entradas/salidas	Operandos	Tipos de datos
IN	VB, IB, QB, MB, SB, SMB, LB, VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, VD, ID, QD, MD, SD, SMD, LD, HC, AC, constante, *VD, *LD, *AC	BYTE, WORD DWORD
N	VB, IB, QB, MB, SB, SMB, LB, AC, constante, *VD, *LD, *AC	Byte
OUT	VB, IB, QB, MB, SB, SMB, LB, VW, IW, QW, MW, SW, SMW, LW, T, C, VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *LD, *AC	BYTE, WORD DWORD

Ejemplos de operaciones de rotación y desplazamiento

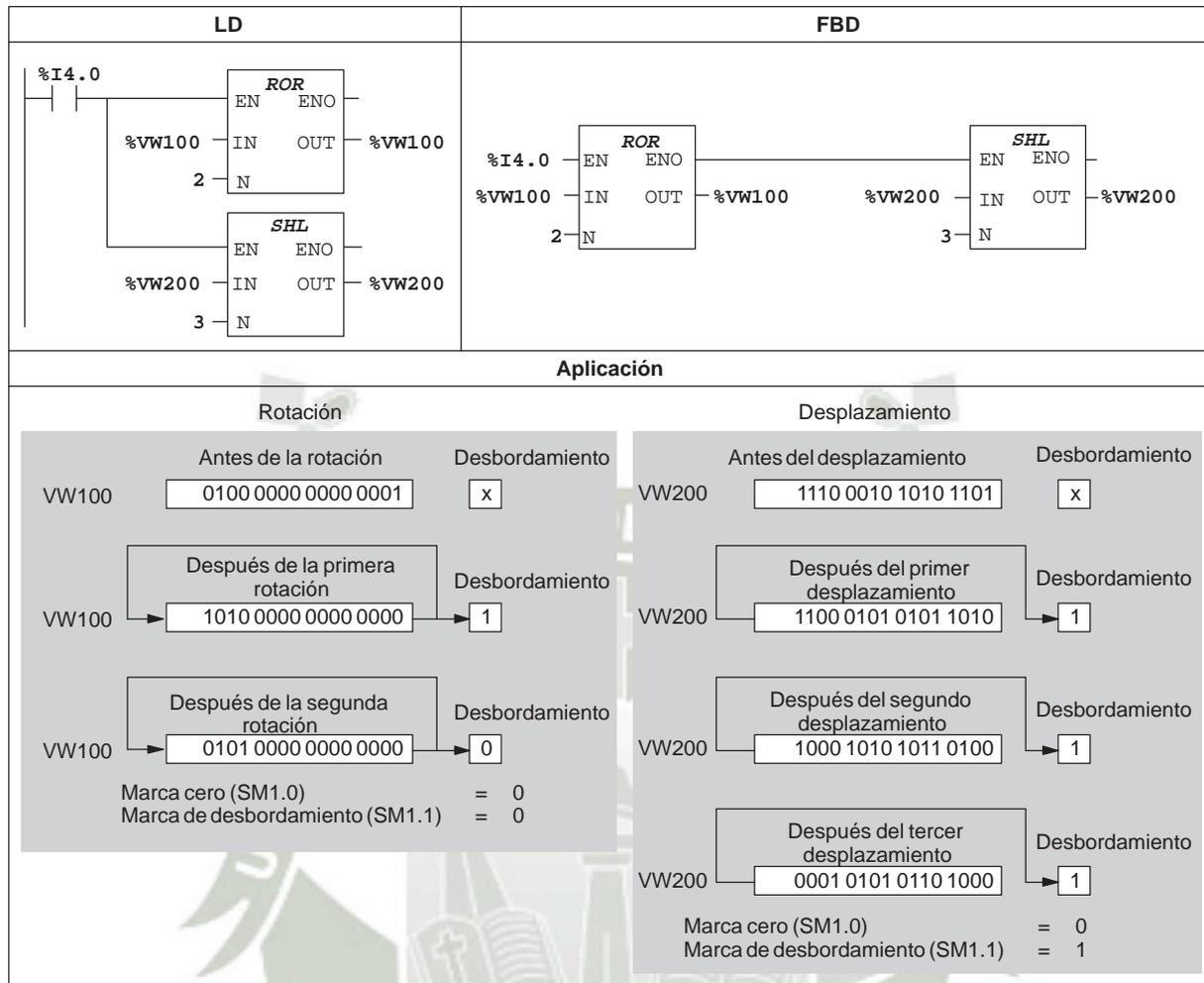


Figura 10-11 Ejemplo de las funciones de desplazamiento y rotación en LD y FBD

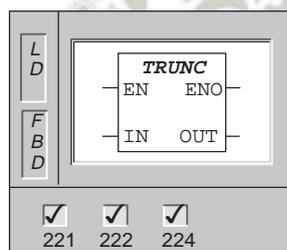
## 10.9 Operaciones de conversión (IEC)

En la tabla 10-9 se indican las páginas donde se describen las operaciones de conversión (IEC) no normalizadas a las que hace referencia el presente apartado.

Tabla 10-9 Operaciones de conversión (IEC) no normalizadas

Descripción	Página
Decodificar	9-131
Codificar	9-131
Segmento	9-133
Convertir de ASCII a hexadecimal, Convertir de hexadecimal a ASCII	9-135
Convertir de entero a ASCII	9-136
Convertir de entero doble a ASCII	9-138
Convertir de real a ASCII	9-139

### Truncar



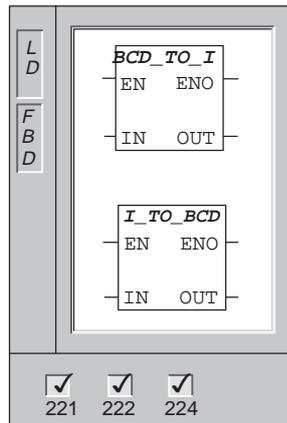
La función **Truncar** convierte un número real (IN) en un valor de entero doble y deposita el resultado en OUT. El resultado no se redondea.

Condiciones de error que ponen ENO a 0: SM1.1 (desbordamiento), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Esta función afecta a las siguientes marcas especiales: SM1.1 (desbordamiento)

Entradas/salidas	Operandos	Tipos de datos
IN	VD, ID, QD, MD, SD, SMD, LD, AC, constante, *VD, *AC, *LD	REAL
OUT	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *AC, *LD	DINT

### Convertir de BCD a entero, Convertir de entero a BCD



La función **Convertir de BCD a entero** convierte el valor BCD (decimal codificado en binario) de entrada (IN) en un valor de entero y carga el resultado en la variable indicada por OUT.

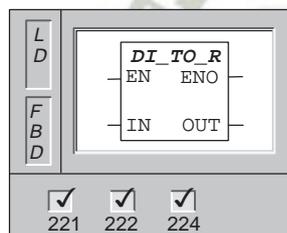
La función **Convertir de entero a BCD** convierte el valor de entero de entrada en un valor BCD (decimal codificado en binario) y carga el resultado en OUT.

Condiciones de error que ponen ENO a 0: SM1.6 (BCD), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Estas funciones afectan a las siguientes marcas especiales: SM1.6 (BCD no válido)

Entradas/salidas	Operandos	Tipos de datos
IN	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, AC, constante, *VD, *LD, *AC	WORD
OUT	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, *VD, *AC, *LD	WORD

### Convertir de entero doble a real

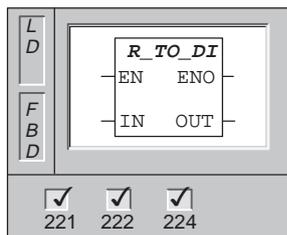


La función **Convertir de entero doble a real** convierte un entero de 32 bits con signo (IN) en un número real de 32 bits y carga el resultado en la variable indicada por OUT.

Condiciones de error que ponen ENO a 0: SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Entradas/salidas	Operandos	Tipos de datos
IN	VD, ID, QD, MD, SD, SMD, LD, HC, AC, constante, *VD, *LD, *AC	DINT
OUT	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *LD, *AC	REAL

### Convertir de real a entero doble

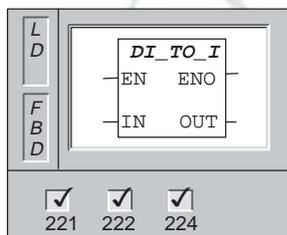


La función **Convertir de real a entero doble** convierte un valor de número real (N) en un valor de entero doble y carga el resultado en la variable indicada por OUT.

Condiciones de error que ponen ENO a 0: SM1.1 (desbordamiento), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Entradas/salidas	Operandos	Tipos de datos
IN	VD, ID, QD, MD, SD, SMD, LD, AC, constante,*VD,*LD, *AC	REAL
OUT	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *LD, *AC	DINT

### Convertir de entero doble a entero



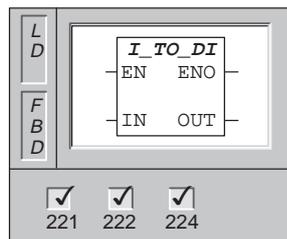
La función **Convertir de entero doble a entero** convierte un entero doble (IN) en un valor de entero y carga el resultado en la variable indicada por OUT.

Condiciones de error que ponen ENO a 0: SM1.1 (desbordamiento), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Esta función afecta a las siguientes marcas especiales: SM1.1 (desbordamiento)

Entradas/salidas	Operandos	Tipos de datos
IN	VD, ID, QD, MD,SD, SMD, LD, HC, AC, constante,*VD,*LD, *AC	DINT
OUT	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, *VD, *LD, *AC	INT

### Convertir de entero a entero doble

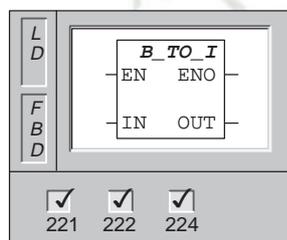


La función **Convertir de entero a entero doble** convierte un valor de entero (IN) en un valor de entero doble y carga el resultado en la variable indicada por OUT.

Condiciones de error que ponen ENO a 0: SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Entradas/salidas	Operandos	Tipos de datos
IN	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, constante, AC, *VD, *LD, *AC	INT
OUT	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *LD, *AC	DINT

### Convertir de byte a entero

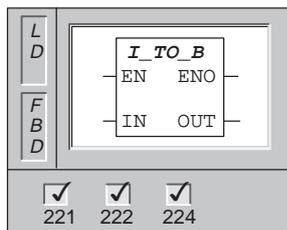


La función **Convertir de byte a entero** convierte el valor de byte (IN) en un valor de entero y carga el resultado en la variable indicada por OUT.

Condiciones de error que ponen ENO a 0: SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Entradas/salidas	Operandos	Tipos de datos
IN	VB, IB, QB, MB, SB, SMB, LB, AC, constante, *VD, *LD, *AC	Byte
OUT	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, *VD, *LD, *AC	INT

### Convertir de entero a byte



La función **Convertir de entero a byte** convierte un valor de entero (IN) en un valor de byte y carga el resultado en la variable indicada por OUT.

Condiciones de error que ponen ENO a 0: SM1.1 (desbordamiento), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Esta función afecta a las siguientes marcas especiales: SM1.1 (desbordamiento)

Entradas/salidas	Operandos	Tipos de datos
IN	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, AC, constante, *VD, *LD, *AC	INT
OUT	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *LD, *AC	Byte

### Ejemplos de conversión

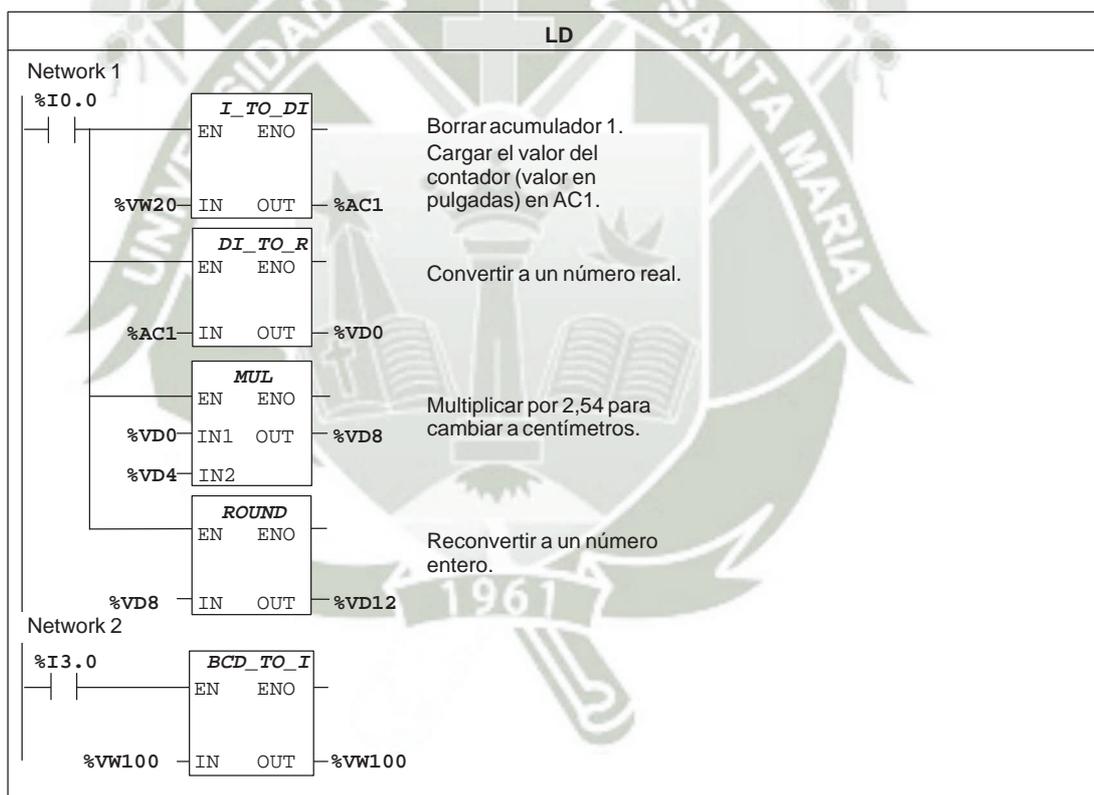


Figura 10-12 Ejemplo de una operación de conversión de un número real en LD

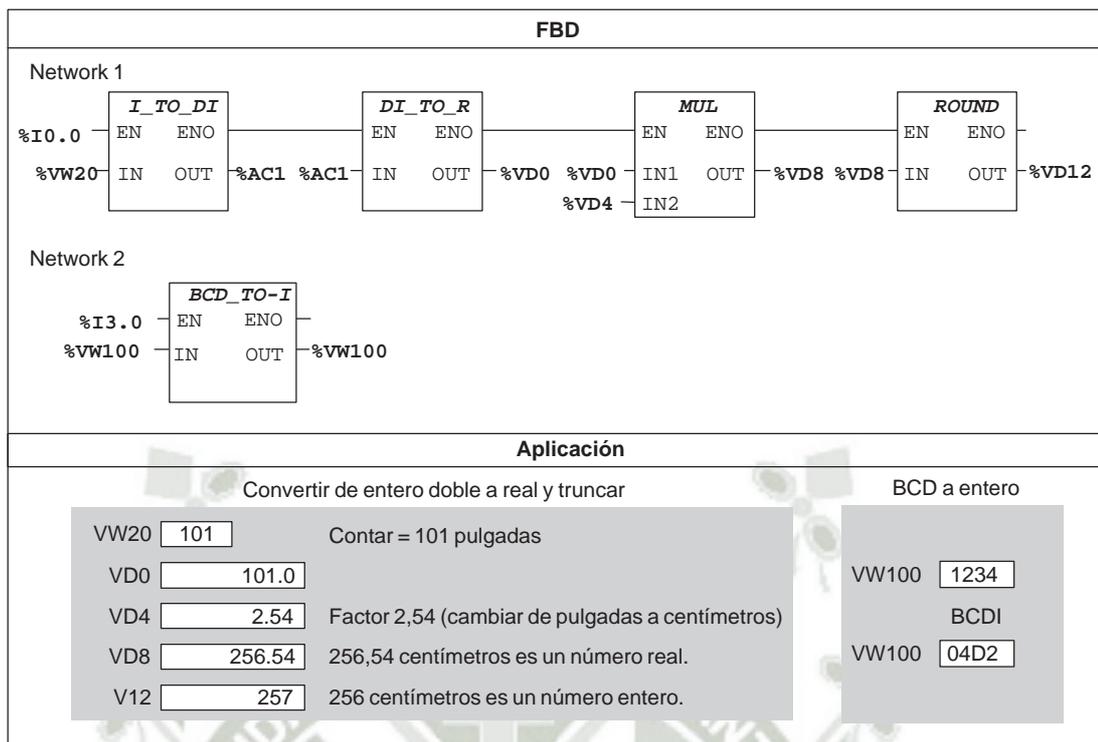


Figura 10-13 Ejemplo de una operación de conversión de un número real en FBD

# A

## Datos técnicos S7-200

### Índice del capítulo

Apartado	Descripción	Página
A.1	Datos técnicos generales	A-2
A.2	Datos técnicos de la CPU 221	A-6
A.3	Datos técnicos de la CPU 222	A-11
A.4	Datos técnicos de la CPU 224	A-16
A.5	Datos técnicos del módulo de ampliación EM221 de entradas digitales	A-21
A.6	Datos técnicos de los módulos de ampliación EM222 de salidas digitales	A-23
A.7	Datos técnicos de los módulos de ampliación EM223, 8 entradas digitales/8 salidas digitales	A-25
A.8	Cartuchos opcionales	A-28
A.9	Cable de módulo de ampliación	A-29
A.10	Cable PC/PPI	A-30

## A.1 Datos técnicos generales

### Homologaciones nacionales e internacionales

Las características de funcionamiento y las pruebas realizadas con los productos de la gama S7-200 se basan en las homologaciones nacionales e internacionales que se indican a continuación. En la tabla A-1 se indica la conformidad específica con dichas homologaciones.

- Underwriters Laboratories, Inc.: UL 508 registrado (Industrial Control Equipment)
- Canadian Standards Association: CSA C22.2 nº 142, certificado (Process Control Equipment)
- Factory Mutual Research: FM clase I, categoría 2, grupo A, B, C, y D Hazardous Locations, T4A
- VDE 0160: equipos electrónicos de uso en instalaciones de corriente eléctrica
- Directiva de Baja Tensión de la Comunidad Europea 73/23/CEE (EN 61131-2): Automatas programables – requisitos del equipo
- Directiva EMC de la Comunidad Europea (CE) 89/336/CEE

Normas de emisión electromagnética:

EN 50081-1: entornos residenciales, comerciales y semi-industriales

EN 50081-2: entornos industriales

Normas de inmunidad electromagnética:

EN 50082-2: entornos industriales

## Datos técnicos

La tabla A-1 muestra los datos técnicos de las CPUs S7-200 y sus módulos de ampliación.

Tabla A-1 Datos técnicos de la gama S7-200

<b>Condiciones ambientales – Transporte y almacenamiento</b>	
IEC 68-2-2, ensayo Bb, calor seco y IEC 68-2-1, ensayo Ab, Frío	-40° C a +70° C
IEC 68-2-30, ensayo Dd, calor húmedo	25° C a 55° C, 95% humedad
IEC 68-2-31, vuelco	100 mm, 4 gotas, desembalado
IEC 68-2-32, caída libre	1m, 5 veces, embalado para embarque
<b>Condiciones ambientales – Funcionamiento</b>	
Condiciones ambientales (aire de entrada 25 mm debajo de la unidad)	0° C a 55° C montaje horizontal 0° C a 45° C montaje vertical 95% humedad no condensante
IEC 68-2-14 Ensayo Nb	5° C a 55° C, 3° C/minuto
IEC 68-2-27 Choque mecánico	15 G, 11 ms impulso, 6 choques en c/u de 3 ejes
IEC 68-2-6 Vibración sinusoidal	0,30 mm pico a pico 10 a 57 Hz; 2G montaje en armario eléctrico, 1G montaje en perfil soporte, 57 a 150 Hz; 10 barridos por eje, 1 octava/minuto
EN 60529, IP20 Protección mecánica	Protege los dedos contra el contacto con alto voltaje, según pruebas realizadas con sondas estándar. Se requiere protección externa contra polvo, impurezas, agua y objetos extraños de menos de 12,5 mm de diámetro.
<b>Compatibilidad electromagnética — Inmunidad<sup>1</sup> según EN50082-2<sup>1</sup></b>	
EN 61000-4-2 (IEC 801-2) Descargas electrostáticas	8 kV descarga en el aire a todas las superficies y al interface de comunicación
EN 50140 (IEC 801-3) Campos electromagnéticos radiados	80 MHz a 1 GHz 10 V/m, 80% modulación con señal de 1 kHz
EN 50141 Perturbaciones conducidas	0,15 MHz a 80 GHz 10 V/m, 80% modulación con señal de 1 kHz
EN 50204 Inmunidad a radioteléfonos digitales	900 MHz ± 5 MHz, 10 V/m, 50% ciclo de trabajo, frecuencia de repetición 200 Hz
EN 61000-4-4 (IEC 801-4) Transitorios eléctricos rápidos	2 kV, 5 kHz con red de unión a la alimentación AC y DC 2 kV, 5 kHz con abrazadera de unión a las E/S digitales y a la comunicación
EN 61000-4-5 (IEC 801-5) Inmunidad a ondas de choque	2 kV asimétrico, 1 kV simétrico 5 impulsos positivos/5 impulsos negativos 0°, +90°, -90° decalaje de fase (para los circuitos de DC 24 V se necesita una protección externa contra sobrecargas)
VDE 0160 Sobrevoltaje no periódico	a AC 85 V línea, 90° decalaje de fase, aplicar cresta de 390 V, impulso de 1,3 ms a AC 180 V línea, 90° decalaje de fase, aplicar cresta de 750 V, impulso de 1,3 ms

Tabla A-1 Datos técnicos de la gama S7-200

<b>Compatibilidad electromagnética — Emisiones conducidas y radiadas según EN50081 -1 2 y -2</b>	
EN 55011, clase A, grupo 1, conducida <sup>1</sup> 0,15 a 0,5 MHz 0,15 a 5 MHz 5 MHz a 30 MHz	< 79 dB (μV) casi cresta; < 66 dB (μV) promedio < 73 dB (μV) casi cresta; < 60 dB (μV) promedio < 73 dB (μV) casi cresta; < 60 dB (μV) promedio
EN 55011, clase A, grupo 1, radiada <sup>1</sup> 30 MHz a 230 kHz 230 MHz a 1 GHz	30 dB (μV/m) casi cresta; medido a 30 m 37 dB (μV/m) casi cresta; medido a 30 m
EN 55011, clase B, grupo 1, conducida <sup>2</sup> 0,15 a 0,5 MHz 0,5 MHz a 5 MHz 5 MHz a 30 MHz	< 66 dB (μV) decremento casi cresta con frecuencia logarítmica a 56 dB (μV) < 56 dB (μV) decremento promedio con frecuencia logarítmica a 46 dB (μV) < 56 dB (μV) casi cresta; < 46 dB (μV) promedio < 60 dB (μV) casi cresta; < 50 dB (μV) promedio
EN 55011, clase B, grupo 1, radiada <sup>2</sup> 30 MHz a 230 kHz 230 MHz a 1 GHz	30 dB (μV/m) casi cresta; medido a 10 m 37 dB (μV/m) casi cresta; medido a 10 m
<b>Prueba de aislamiento a hipervoltajes</b>	
24 V/5 V circuitos nominales 115/230 V circuitos a tierra 115/230 V circuitos hasta 115/230 V circuitos 230 V circuitos hasta 24 V/5V circuitos 115 V circuitos hasta 24 V/5V circuitos	AC 500 V (límites de aislamiento óptico) AC 1,500 V AC 1,500 V AC 1,500 V AC 1,500 V

- 1 La unidad deberá montarse en un soporte metálico puesto a tierra. El S7-200 deberá ponerse a tierra directamente a través del soporte metálico. Los cables se deberán conducir a lo largo de los soportes metálicos.
- 2 La unidad deberá montarse en una caja metálica puesta a tierra. La línea de alimentación de corriente alterna se deberá equipar con un filtro SIEMENS B84115-E-A30 o similar, teniendo el cable una longitud máxima de 25 cm entre los filtros y el S7-200. El cableado de la alimentación DC 24 V y de la alimentación de sensores se deberá apantallar.

### Vida útil de los relés

La figura A-1 muestra los datos típicos de rendimiento de los relés proporcionados por el comercio especializado. El rendimiento real puede variar dependiendo de la aplicación.

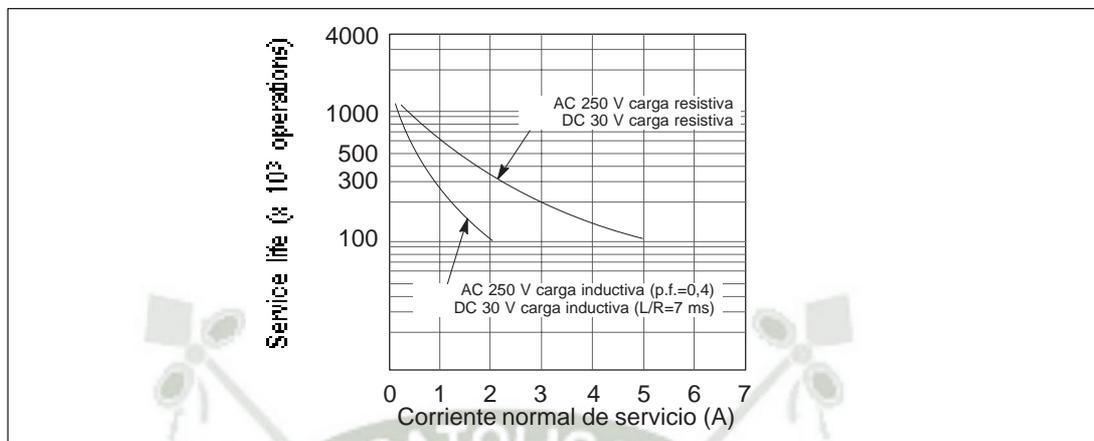


Figura A-1 Vida útil de los relés

## A.2 Datos técnicos de la CPU 221

Tabla A-2 Datos técnicos de la CPU 221 DC/DC/DC y de la CPU 221 AC/DC/relé

Descripción Nº de referencia	CPU 221 DC/DC/DC 6ES7 211-0AA20-0XBO	CPU 221 AC/DC/relé 6ES7 211-0BA20-0XB0
<b>Tamaño físico</b>		
Dimensiones (l x a x p)	90 mm x 80 mm x 62 mm	90 mm x 80 mm x 62 mm
Peso	270 g	310 g
Pérdida de corriente (disipación)	4 W	6 W
<b>Características de la CPU</b>		
Entradas digitales integradas	6 entradas	6 entradas
Salidas digitales integradas	4 salidas	4 salidas
Contadores rápidos (valor de 32 bits)		
Total	4 contadores rápidos	4 contadores rápidos
Nº de contadores de fase simple	4, con una frecuencia de reloj de 20 kHz c/u	4, con una frecuencia de reloj de 20 kHz c/u
Nº de contadores de dos fases	2, con una frecuencia de reloj de 20 kHz c/u	2, con una frecuencia de reloj de 20 kHz c/u
Salidas de impulsos	2 a una frecuencia de impulsos de 20 kHz	2 a una frecuencia de impulsos de 20 kHz
Potenciómetros analógicos	1 con resolución de 8 bits	1 con resolución de 8 bits
Interrupciones temporizadas	2 con resolución de 1 ms	2 con resolución de 1 ms
Interrupciones de flanco	4 flancos positivos y/o 4 flancos negativos	4 flancos positivos y/o 4 flancos negativos
Tiempos de filtración de entradas	7 márgenes de 0,2 ms a 12,8 ms	7 márgenes de 0,2 ms a 12,8 ms
Captura de impulsos	6 entradas de captura de impulsos	6 entradas de captura de impulsos
Tamaño del programa (almacenado permanentemente)	2048 palabras	2048 palabras
Tamaño del bloque de datos:		
Almacenamiento permanente	1024 palabras	1024 palabras
Respaldo por condensador de alto rendimiento o pila	1024 palabras	1024 palabras
E/S de ampliación digitales (máx.)	10 E/S	10 E/S
Marcas internas	256 bits	256 bits
Almacenamiento permanente al apagar	112 bits	112 bits
Respaldo por condensador de alto rendimiento o pila	256 bits	256 bits
Temporizadores (total)	256 temporizadores	256 temporizadores
Respaldo por condensador de alto rendimiento o pila	64 temporizadores	64 temporizadores
1 ms	4 temporizadores	4 temporizadores
10 ms	16 temporizadores	16 temporizadores
100 ms	236 temporizadores	236 temporizadores
Contadores (total)	256 contadores	256 contadores
Respaldo por condensador de alto rendimiento o pila	256 contadores	256 contadores
Velocidad de ejecución booleana	0,37 µs por operación	0,37 µs por operación
Velocidad de ejecución de Transferir palabra	34 µs por operación	34 µs por operación
Velocidad de ejecución de temporizadores/ contadores	50 µs a 64 µs por operación	50 µs a 64 µs por operación
Velocidad de ejecución de aritmética de precisión simple	46 µs por operación	46 µs por operación
Velocidad de ejecución de aritmética en coma flotante	100 µs a 400 µs por operación	100 µs a 400 µs por operación
Tiempo de respaldo por el condensador de alto rendimiento	típ. 50 h, mín. 8 h a 40° C	típ. 50 h, mín. 8 h a 40° C

Tabla A-2 Datos técnicos de la CPU 221 DC/DC/DC y de la CPU 221 AC/DC/relé

Descripción Nº de referencia	CPU 221 DC/DC/DC 6ES7 211-0AA20-0XB0	CPU 221 AC/DC/relé 6ES7 211-0BA20-0XB0
<b>Comunicación integrada</b>		
Nº de puertos	1 puerto	1 puerto
Interface eléctrico	RS-485	RS-485
Aislamiento (señal externa a circuito lógico)	Sin aislamiento	Sin aislamiento
Velocidades de transferencia PPI/MPI	9,6, 19,2 y 187,5 kbit/s	9,6, 19,2 y 187,5 kbit/s
Velocidades de transferencia Freeport	0,3, 0,6, 1,2, 2,4, 4,8, 9,6, 19,2 y 38,4 kbit/s	0,3, 0,6, 1,2, 2,4, 4,8, 9,6, 19,2 y 38,4 kbit/s
Longitud máx. del cable por segmento hasta 38,4 kbit/s 187,5 kbit/s	1200 m 1000 m	1200 m 1000 m
Nº máximo de estaciones Por segmento Por red	32 estaciones 126 estaciones	32 estaciones 126 estaciones
Nº máximo de maestros	32 maestros	32 maestros
Modo maestro PPI (NETR/NETW)	Sí	Sí
Enlaces MPI	4 en total; 2 reservados: 1 para PG y 1 OP	4 en total; 2 reservados: 1 para PG y 1 OP
<b>Cartuchos opcionales</b>		
Cartucho de memoria (almacenamiento permanente)	Programa, datos y configuración típ. 200 días	Programa, datos y configuración típ. 200 días
Cartucho de pila (tiempo de respaldo de datos)	2 minutos por mes a 25° C	2 minutos por mes a 25° C
Cartucho de reloj (precisión del reloj)	7 minutos por mes 0° C a 55° C	7 minutos por mes 0° C a 55° C
<b>Alimentación</b>		
Tensión de línea (margen admisible)	DC 20,4 a 28,8 V	AC 85 a 264 V 47 a 63 Hz
Corriente de entrada (sólo CPU) / carga máx.	70/600 mA a DC 24 V	25/80 mA a AC 240 V 25/180 mA a AC 120 V
Extra-corriente de serie (máx.)	10 A a DC 28,8 V	20 A a AC 264 V
Aislamiento (corriente de entrada a lógica)	Sin aislamiento	AC 1500 V
Tiempo de retardo (desde la pérdida de corriente de entrada)	mín. 10 ms de DC 24 V	80 ms de AC 240 V, 20 ms de 120 VAC
Fusible interno (no reemplazable por el usuario)	2 A, 250 V, de acción lenta	2 A, 250 V, de acción lenta
<b>Alimentación para sensores DC 24 V</b>		
Margen de tensión	DC 15,4 a 28,8 V	DC 20,4 a 28,8 V
Corriente máxima	180 mA	180 mA
Rizado corriente parásita	Igual que línea de entrada	Menos de 1 V pico a pico (máx.)
Corriente límite	600 mA	600 mA
Aislamiento (alimentación de sensores a circuito lógico)	Sin aislamiento	Sin aislamiento

Tabla A-2 Datos técnicos de la CPU 221 DC/DC/DC y de la CPU 221 AC/DC/relé

Descripción Nº de referencia	CPU 221 DC/DC/DC 6ES7 211-0AA20-0XB0	CPU 221 AC/DC/relé 6ES7 211-0BA20-0XB0
<b>Características de las entradas</b>		
Nº de entradas integradas	6 entradas	6 entradas
Tipo de entrada	Sumidero de corriente/fuente (tipo 1 IEC con sumidero de corriente)	Sumidero de corriente/fuente (tipo 1 IEC con sumidero de corriente)
<b>Tensión de entrada</b>		
Tensión máx. continua admisible	DC 30 V	DC 30 V
Sobretensión transitoria	DC 35 V, 0,5 s	DC 35 V, 0,5 s
Valor nominal	DC 24 V a 4 mA, nominal	DC 24 V a 4 mA, nominal
Señal 1 lógica (mín.)	mín. DC 15 V a 2,5 mA	mín. DC 15 V a 2,5 mA
Señal 0 lógica (máx.)	máx. DC 5 V a 1 mA	máx. DC 5 V a 1 mA
<b>Aislamiento (campo a circuito lógico)</b>		
Separación galvánica	AC 500 V, 1 minuto	AC 500 V, 1 minuto
Grupos de aislamiento de	4 entradas/2 entradas	4 entradas/2 entradas
<b>Tiempos de retardo de las entradas</b>		
Entradas filtradas y entradas de interrupción	0,2 a 12,8 ms, seleccionable por el usuario	0,2 a 12,8 ms, seleccionable por el usuario
<b>Entradas de reloj de los contadores rápidos</b>		
Fase simple		
DC 30 V	Nivel 1 lógico = DC 15 V a 20 kHz 30 kHz	20 kHz 30 kHz
DC 26 V	Nivel 1 lógico = DC 15 V a 10 kHz	10 kHz
Contadores A/B	20 kHz	20 kHz
DC 30 V	Nivel 1 lógico = DC 15 V a	
DC 26 V	Nivel 1 lógico = DC 15 V a	
<b>Conexión de sensor de proximidad de 2 hilos (Bero)</b>		
Corriente de fuga admisible	máx. 1 mA	máx. 1 mA
<b>Longitud del cable</b>		
No apantallado (no HSC)	300 m	300 m
Apantallado	500 m	500 m
Entradas HSC, apantalladas	50 m	50 m
<b>Nº de entradas ON simultáneamente</b>		
40 ° C	6	6
55 ° C	6	6
<b>Características de las salidas</b>		
Nº de salidas integradas	4 salidas	4 salidas
Tipo de salida	Estado sólido-MOSFET	Relé, contacto de baja potencia
<b>Tensión de salida</b>		
Margen admisible	DC 20,4 a 28,8 V	DC 5 a 30 V ó AC 5 a 250 V
Valor nominal	DC 24 V	–
Señal 1 lógica a corriente máxima	mín. DC 20 V	–
Señal 0 lógica a 10 K Ω de carga	máx. DC 0,1 V	–

Tabla A-2 Datos técnicos de la CPU 221 DC/DC/DC y de la CPU 221 AC/DC/relé

Descripción Nº de referencia	CPU 221 DC/DC/DC 6ES7 211-0AA20-0XB0	CPU 221 AC/DC/relé 6ES7 211-0BA20-0XB0
<b>Corriente de salida</b>		
Señal 1 lógica	0,75 A	2.00 A
Nº de grupos de salidas	1	2
Nº de salidas ON (máx.)	4	4
Por grupo – montaje horizontal (máx.)	4	3 y 1
Por grupo – montaje vertical (máx.)	4	3 y 1
Corriente máx. por común/grupo	3.0 A	6.0 A
Carga LEDs	5.0 W	30 W DC/200 W AC
Resistencia estado ON (resistencia contactos)	0,3Ω	0,002 Ω, máx. si son nuevas
Corriente de derivación por salida	máx. 10 μA	–
Sobrecorriente momentánea	máx. 8 A, 100 ms	7A al estar cerrados los contactos
Protección contra sobrecargas	no	no
<b>Aislamiento</b>		
Separación galvánica	AC 500 V, 1 minuto	–
Resistencia de aislamiento	–	100 M Ω, mín. si son nuevas
Aislamiento bobina a contacto	–	AC 500 V, 1 minuto
Aislamiento entre contactos abiertos	–	AC 750 V, 1 minuto
En grupos de	4 salidas	3 salidas y 1 salida
<b>Carga inductiva, apriete</b>		
Repetición disipación de energía <	1 W, en todos los canales	–
0.5 LI <sup>2</sup> x frecuencia de conmutación	L+ menos 48 V	–
<b>Límites tensión de bloqueo</b>		
<b>Retardo de las salidas</b>		
OFF a ON (Q0.0 y Q0.1)	máx. 2 μs	–
ON a OFF (Q0.0 y Q0.1)	máx. 10 μs	–
OFF a ON (Q0.2 y Q0.3)	máx. 15 μs	–
ON a OFF (Q0.2 y Q0.3)	máx. 100 μs	–
<b>Frecuencia de conmutación (salida de impulsos)</b>		
Q0.0 y Q0.1	máx. 20 kHz	máx. 1 Hz
<b>Relé</b>		
Retardo de conmutación	–	máx. 10 ms
Vida útil mecánica (sin carga)	–	10.000.000 ciclos abiertos/cerrados
Vida útil contactos a carga nominal	–	100.000 ciclos abiertos/cerrados
<b>Longitud del cable</b>		
No apantallado	150 m	150 m
Apantallado	500 m	500 m

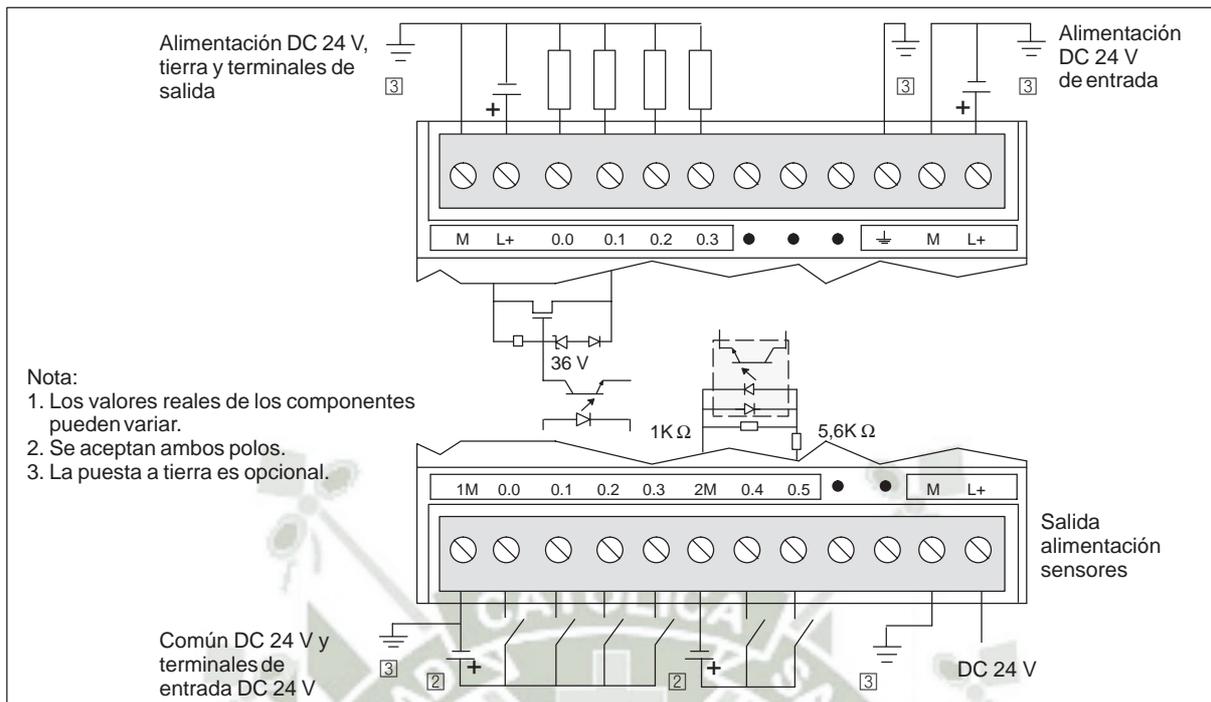


Figura A-2 Identificación de terminales de conexión para la CPU 221 DC/DC/DC

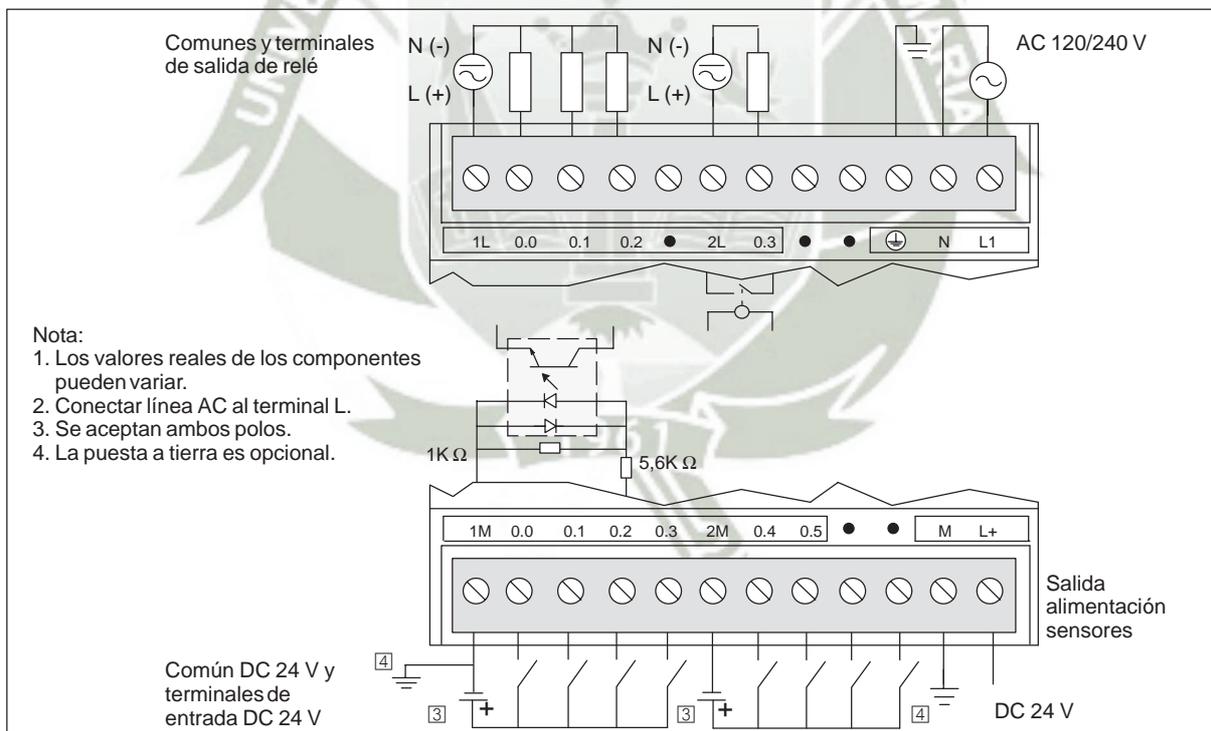


Figura A-3 Identificación de terminales de conexión para la CPU 221 AC/DC/relé

### A.3 Datos técnicos de la CPU 222

Tabla A-3 Datos técnicos de la CPU 222 DC/DC/DC y de la CPU 222 AC/DC/relé

Descripción Nº de referencia	CPU 222 DC/DC/DC 6ES7 212-1AB20-0XB0	CPU 222 AC/DC/relé 6ES7 212-1BB20-0XB0
<b>Tamaño físico</b>		
Dimensiones (l x a x p)	90 mm x 80 mm x 62 mm	90 mm x 80 mm x 62 mm
Peso	270 g	310 g
Pérdida de corriente (disipación)	4 W	6 W
<b>Características de la CPU</b>		
Entradas digitales integradas	8 entradas	8 entradas
Salidas digitales integradas	6 salidas	6 salidas
Contadores rápidos (valor de 32 bits)		
Total	4 contadores rápidos	4 contadores rápidos
Contadores de fase simple	4, con una frecuencia de reloj de 20 kHz c/u	4, con una frecuencia de reloj de 20 kHz c/u
Contadores de dos fases	2, con una frecuencia de reloj de 20 kHz c/u	2, con una frecuencia de reloj de 20 kHz c/u
Salidas de impulsos	2 a una frecuencia de impulsos de 20 kHz	2 a una frecuencia de impulsos de 20 kHz
Potenciómetros analógicos	1 con resolución de 8 bits	1 con resolución de 8 bits
Interrupciones temporizadas	2 con resolución de 1 ms	2 con resolución de 1 ms
Interrupciones de flanco	4 flancos positivos y/o 4 flancos negativos	4 flancos positivos y/o 4 flancos negativos
Tiempos de filtración de entradas	7 márgenes de 0,2 ms a 12,8 ms	7 márgenes de 0,2 ms a 12,8 ms
Captura de impulsos	8 entradas de captura de impulsos	8 entradas de captura de impulsos
Tamaño del programa (almacenado permanentemente)	2048 palabras	2048 palabras
Tamaño del bloque de datos	1024 palabras	1024 palabras
Almacenamiento permanente	1024 palabras	1024 palabras
Respaldo por condensador de alto rendimiento o pila	2 módulos	2 módulos
Nº de módulos de ampliación	256 E/S	256 E/S
E/S de ampliación digitales (máx.)	16 entradas y 16 salidas	16 entradas y 16 salidas
E/S analógicas (máx.)		
Marcas internas	256 bits	256 bits
Almacenamiento permanente al apagar	112 bits	112 bits
Respaldo por condensador de alto rendimiento o pila	256 bits	256 bits
Temporizadores (total)	256 temporizadores	256 temporizadores
Respaldo por condensador de alto rendimiento o pila	64 temporizadores	64 temporizadores
1 ms	4 temporizadores	4 temporizadores
10 ms	16 temporizadores	16 temporizadores
100 ms	236 temporizadores	236 temporizadores
Contadores (total)	256 contadores	256 contadores
Respaldo por condensador de alto rendimiento o pila	256 contadores	256 contadores
Velocidad de ejecución booleana	0,37 $\mu$ s por operación	0,37 $\mu$ s por operación
Velocidad de ejecución de Transferir palabra	34 $\mu$ s por operación	34 $\mu$ s por operación
Velocidad de ejecución de temporizadores/contadores	50 $\mu$ s a 64 $\mu$ s por operación	50 $\mu$ s a 64 $\mu$ s por operación
Velocidad de ejecución de aritmética de precisión simple	46 $\mu$ s por operación	46 $\mu$ s por operación
Velocidad de ejecución de aritmética en coma flotante	100 $\mu$ s a 400 $\mu$ s por operación	100 $\mu$ s a 400 $\mu$ s por operación
Tiempo de respaldo por el condensador de alto rendimiento	típ. 50 h, mín. mín. 8 h a 40° C	típ. 50 h, mín. mín. 8 h a 40° C

Tabla A-3 Datos técnicos de la CPU 222 DC/DC/DC y de la CPU 222 AC/DC/relé

Descripción Nº de referencia	CPU 222 DC/DC/DC 6ES7 212-1AB20-0XB0	CPU 222 AC/DC/relé 6ES7 212-1BB20-0XB0
<b>Comunicación integrada</b>		
Nº de puertos	1 puerto	1 puerto
Interface eléctrico	RS-485	RS-485
Aislamiento (señal externa a circuito lógico)	Sin aislamiento	Sin aislamiento
Velocidades de transferencia PPI/MPI	9,6, 19,2 y 187,5 kbit/s	9,6, 19,2 y 187,5 kbit/s
Velocidades de transferencia Freeport	0,3, 0,6, 1,2, 2,4, 4,8, 9,6, 19,2 y 38,4 kbit/s	0,3, 0,6, 1,2, 2,4, 4,8, 9,6, 19,2 y 38,4 kbit/s
Longitud máx. del cable por segmento hasta 38,4 kbit/s	1200 m	1200 m
187,5 kbit/s	1000 m	1000 m
Nº máximo de estaciones		
Por segmento	32 estaciones	32 estaciones
Por red	126 estaciones	126 estaciones
Nº máximo de maestros	32 maestros	32 maestros
Modo maestro PPI (NETR/NETW)	Sí	Sí
Enlaces MPI	4 en total; 2 reservados: 1 para PG y 1 OP	4 en total; 2 reservados: 1 para PG y 1 OP
<b>Cartuchos opcionales</b>		
Cartucho de memoria (almacenamiento permanente)	Programa, datos y configuración típ. 200 días	Programa, datos y configuración típ. 200 días
Cartucho de pila (tiempo de respaldo de datos)	2 minutes per month at 25° C	2 minutes per month at 25° C
Cartucho de reloj (precisión del reloj)	7 minutes per month at 0° C a 55° C	7 minutes per month at 0° C a 55° C
<b>Alimentación</b>		
Tensión de línea (margen admisible)	DC 20,4 a 28,8 V	AC 85 V a 264 V, 47 a 63 Hz
Corriente de entrada (sólo CPU) / carga máx.	70/600 mA a DC 24 V	25/80 mA a AC 240 V 25/180 mA a AC 120 V
Extra-corriente de serie (máx.)	10 A a DC 28,8 V	20 A a AC 264 V
Aislamiento (corriente de entrada a lógica)	Sin aislamiento	AC 1500 V
Tiempo de retardo (desde la pérdida de corriente de entrada)	mín. 10 ms de DC 24 V	80 ms de AC 240 V, 20 ms de 120 VAC
Fusible interno (no reemplazable por el usuario)	2 A, 250 V, de acción lenta	2 A, 250 V, de acción lenta
<b>+5 alimentación para módulos de ampliación (máx.)</b>	340 mA	340 mA
<b>Alimentación para sensores DC 24 V</b>		
Margen de tensión	DC 15,4 a 28,8 V	DC 20,4 a 28,8 V
Corriente máxima	180 mA	180 mA
Rizado corriente parásita	Igual que línea de entrada	Menos de 1 V pico a pico (máx.)
Corriente límite	600 mA	600 mA
Aislamiento (alimentación de sensores a circuito lógico)	Sin aislamiento	Sin aislamiento

Tabla A-3 Datos técnicos de la CPU 222 DC/DC/DC y de la CPU 222 AC/DC/relé

Descripción Nº de referencia	CPU 222 DC/DC/DC 6ES7 212-1AB20-0XB0	CPU 222 AC/DC/relé 6ES7 212-1BB20-0XB0
<b>Características de las entradas</b>		
Nº de entradas integradas	8 entradas	8 entradas
Tipo de entrada	Sumidero de corriente/fuente (tipo 1 IEC con sumidero de corriente)	Sumidero de corriente/fuente (tipo 1 IEC con sumidero de corriente)
<b>Tensión de entrada</b>		
Tensión máx. continua admisible	DC 30 V	DC 30 V
Sobretensión transitoria	DC 35 V, 0,5 s	DC 35 V, 0,5 s
Valor nominal	DC 24 V a 4 mA, nominal	DC 24 V a 4 mA, nominal
Señal 1 lógica (mín.)	mín. DC 15 V a 2.5 mA	mín. DC 15 V a 2.5 mA
Señal 0 lógica (máx.)	máx. DC 5 V a 1 mA	máx. DC 5 V a 1 mA
<b>Aislamiento (campo a circuito lógico)</b>		
Separación galvánica	AC 500 V, 1 minuto	AC 500 V, 1 minuto
Grupos de aislamiento de	4 entradas	4 entradas
<b>Tiempos de retardo de las entradas</b>		
Entradas filtradas y entradas de interrupción	0,2 a 12,8 ms, seleccionable por el usuario	0,2 a 12,8 ms, seleccionable por el usuario
<b>Entradas de reloj de los contadores rápidos</b>		
Fase simple		
DC 30 V	Nivel 1 lógico = DC 15 V a máx. 20 kHz máx. 30 kHz	máx. 20 kHz máx. 30 kHz
DC 26 V	Nivel 1 lógico = DC 15 V a máx. 10 kHz máx. 20 kHz	máx. 10 kHz máx. 20 kHz
Contadores A/B	Nivel 1 lógico = DC 15 V a máx. 10 kHz máx. 20 kHz	máx. 10 kHz máx. 20 kHz
DC 30 V	Nivel 1 lógico = DC 15 V a máx. 10 kHz máx. 20 kHz	máx. 10 kHz máx. 20 kHz
DC 26 V	Nivel 1 lógico = DC 15 V a máx. 10 kHz máx. 20 kHz	máx. 10 kHz máx. 20 kHz
<b>Conexión de sensor de proximidad de 2 hilos (Bero)</b>		
Corriente de fuga admisible	máx. 1 mA	máx. 1 mA
<b>Longitud del cable</b>		
No apantallado (no HSC)	300 m	300 m
Apantallado	500 m	500 m
Entradas HSC, apantalladas	50 m	50 m
<b>Nº de entradas ON simultáneamente</b>		
40 ° C	8	8
55 ° C	8	8
<b>Características de las salidas</b>		
Nº de salidas integradas	6 salidas	6 salidas
Tipo de salida	Estado sólido-MOSFET	Relé, contacto de baja potencia
<b>Tensión de salida</b>		
Margen admisible	DC 20,4 a 28,8 V	DC 5 a 30 V ó AC 5 a 250 V
Valor nominal	DC 24 V	–
Señal 1 lógica a corriente máxima	mín. DC 20 V	–
Señal 0 lógica a 10 K Ω de carga	máx. DC 0,1 V	–

Tabla A-3 Datos técnicos de la CPU 222 DC/DC/DC y de la CPU 222 AC/DC/relé

Descripción Nº de referencia	CPU 222 DC/DC/DC 6ES7 212-1AB20-0XB0	CPU 222 AC/DC/relé 6ES7 212-1BB20-0XB0
<b>Corriente de salida</b>		
Señal 1 lógica	0,75 A	2.00 A
Nº de grupos de salidas	1	2
Nº de salidas ON (máx.)	6	6
Por grupo – montaje horizontal (máx.)	6	3
Por grupo – montaje vertical (máx.)	6	3
Corriente máx. por común/grupo	4,5 A	6 A
Carga LEDs	5 W	30 W DC/ 200 W AC
Resistencia estado ON (resistencia contactos)	0,3Ω	0,002 Ω, máx. si son nuevas
Corriente de derivación por salida	máx. 10 μA	–
Sobrecorriente momentánea	máx. 8 A, 100 ms	7A al estar cerrados los contactos
Protección contra sobrecargas	no	no
<b>Aislamiento</b>		
Separación galvánica	AC 500 V, 1 minuto	–
Resistencia de aislamiento	–	100 M Ω, mín. si son nuevas
Aislamiento bobina a contacto	–	AC 500 V, 1 minuto
Aislamiento entre contactos abiertos	–	AC 750 V, 1 minuto
En grupos de	6 salidas	3 salidas
<b>Carga inductiva, apriete</b>		
Repetición disipación de energía < 0.5 LI <sup>2</sup> x	1 W, en todos los canales	–
frecuencia de conmutación	L+ menos 48V	–
Límites tensión de bloqueo		
<b>Retardo de las salidas</b>		
OFF a ON (Q0.0 y Q0.1)	máx. 2 μs	–
ON a OFF (Q0.0 y Q0.1)	máx. 10 μs	–
OFF a ON (Q0.2 hasta Q0.5)	máx. 15 μs	–
ON a OFF (Q0.2 hasta Q0.5)	máx. 100 μs	–
<b>Frecuencia de conmutación (salida de impulsos)</b>		
Q0.0 y I0.1	máx. 20 kHz	máx. 1 Hz
<b>Relé</b>		
Retardo de conmutación	–	máx. 10 ms
Vida útil mecánica (sin carga)	–	10.000.000 ciclos abiertos/cerrados
Vida útil contactos a carga nominal	–	100.000 ciclos abiertos/cerrados
<b>Longitud del cable</b>		
Apantallado	150 m	150 m
No apantallado	500 m	500 m

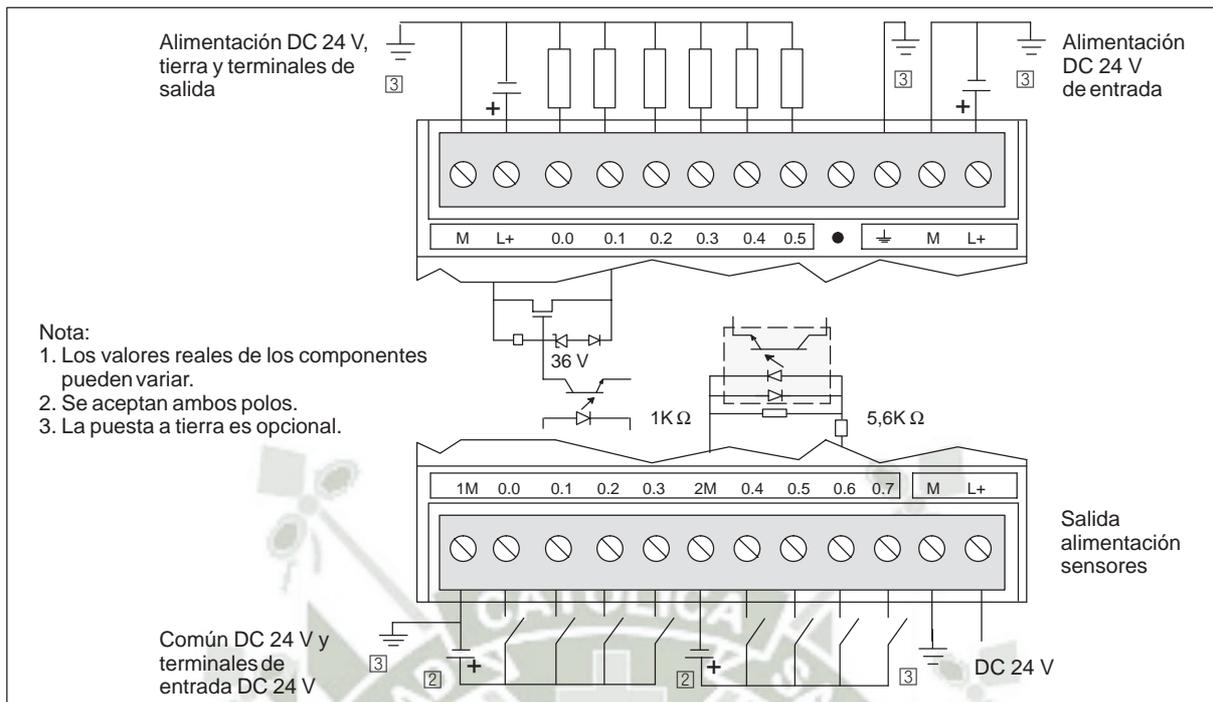


Figura A-4 Identificación de terminales de conexión para la CPU 222 DC/DC/DC

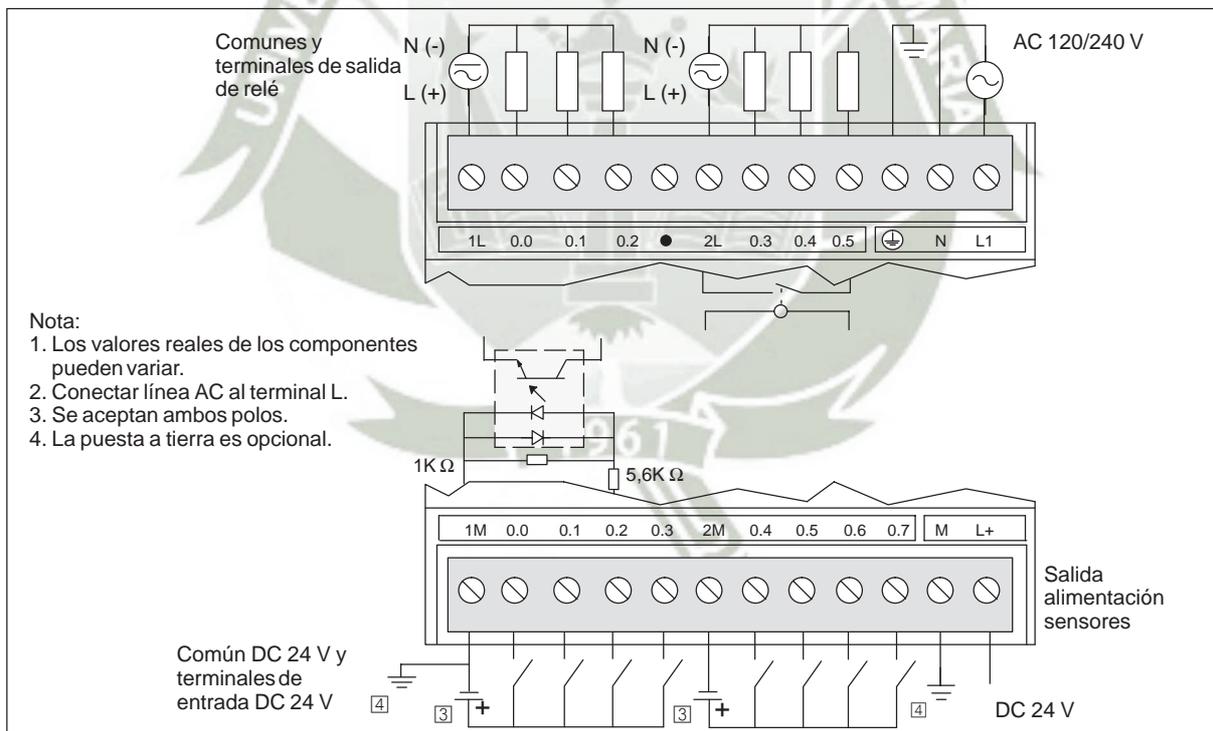


Figura A-5 Identificación de terminales de conexión para la CPU 222 AC/DC/relé

## A.4 Datos técnicos de la CPU 224

Tabla A-4 Datos técnicos de la CPU 224 DC/DC/DC y de la CPU 224 AC/DC/relé

Descripción Nº de referencia	CPU 224 DC/DC/DC 6ES7 214-1AD20-0XB0	CPU 224 AC/DC/relé 6ES7 214-1BD20-0XB0
<b>Tamaño físico</b>		
Dimensiones (l x a x p)	120.5 mm x 80 mm x 62 mm	120.5 mm x 80 mm x 62 mm
Peso	360 g	410 g
Pérdida de corriente (disipación)	8 W	9 W
<b>Características de la CPU</b>		
Entradas digitales integradas	14 entradas	14 entradas
Salidas digitales integradas	10 salidas	10 salidas
Contadores rápidos (valor de 32 bits)		
Total	6 contadores rápidos	6 contadores rápidos
Contadores de fase simple	6, con una frecuencia de reloj de 20 kHz c/u	6, con una frecuencia de reloj de 20 kHz c/u
Contadores de dos fases		
Salidas de impulsos	4, con una frecuencia de reloj de 20 kHz c/u	4, con una frecuencia de reloj de 20 kHz c/u
Potenciómetros analógicos		
Interrupciones temporizadas	2 a una frecuencia de impulsos de 20 kHz	2 a una frecuencia de impulsos de 20 kHz
Interrupciones de flanco	2 con resolución de 8 bits	2 con resolución de 8 bits
	2 con resolución de 1 ms	2 con resolución de 1 ms
Tiempos de filtración de entradas		
Captura de impulsos	4 flancos positivos y/o 4 flancos negativos	4 flancos positivos y/o 4 flancos negativos
Reloj de tiempo real (precisión del reloj)	7 márgenes de 0,2 ms a 12,8 ms	7 márgenes de 0,2 ms a 12,8 ms
	14 entradas de captura de impulsos	14 entradas de captura de impulsos
	2 minutos por mes a 25° C	2 minutos per month at 25° C
	7 minutos por mes 0° C a 55° C	7 minutes per month at 0° C a 55° C
Tamaño del programa (almacenado permanentemente)	4096 palabras	4096 palabras
Tamaño del bloque de datos (almacenado permanentemente):	2560 palabras	2560 palabras
Almacenamiento permanente	2560 palabras	2560 palabras
Respaldo por condensador de alto rendimiento o pila	7 módulos	7 módulos
Nº de módulos de ampliación	256 E/S	256 E/S
E/S de ampliación digitales (máx.)	16 entradas y 16 salidas	16 entradas y 16 salidas
E/S analógicas (máx.)		
Marcas internas	256 bits	256 bits
Almacenamiento permanente al apagar	112 bits	112 bits
Respaldo por condensador de alto rendimiento o pila	256 bits	256 bits
Temporizadores (total)	256 temporizadores	256 temporizadores
Respaldo por condensador de alto rendimiento o pila	64 temporizadores	64 temporizadores
1 ms	4 temporizadores	4 temporizadores
10 ms	16 temporizadores	16 temporizadores
100 ms	236 temporizadores	236 temporizadores

Tabla A-4 Datos técnicos de la CPU 224 DC/DC/DC y de la CPU 224 AC/DC/relé

Descripción Nº de referencia	CPU 224 DC/DC/DC 6ES7 214-1AD20-0XB0	CPU 224 AC/DC/relé 6ES7 214-1BD20-0XB0
Contadores (total)	256 contadores	256 contadores
Respaldo por condensador de alto rendimiento o pila	256 contadores	256 contadores
Velocidad de ejecución booleana	0,37 $\mu$ s por operación	0,37 $\mu$ s por operación
Velocidad de ejecución de Transferir palabra	34 $\mu$ s por operación	34 $\mu$ s por operación
Velocidad de ejecución de temporizadores/contadores	50 $\mu$ s a 64 $\mu$ s por operación	50 $\mu$ s a 64 $\mu$ s por operación
Velocidad de ejecución de aritmética de precisión simple	46 $\mu$ s por operación	46 $\mu$ s por operación
Velocidad de ejecución de aritmética en coma flotante	100 $\mu$ s a 400 $\mu$ s por operación	100 $\mu$ s a 400 $\mu$ s por operación
Tiempo de respaldo por el condensador de alto rendimiento	típ. 190 h, mín. 120 h a 40° C	típ. 190 h, mín. 120 h a 40° C
<b>Comunicación integrada</b>		
Nº de puertos	1 puerto	1 puerto
Interface eléctrico	RS-485	RS-485
Aislamiento (señal externa a circuito lógico)	Sin aislamiento	Sin aislamiento
Velocidades de transferencia PPI/MPI	9,6, 19,2 y 187,5 kbit/s	9,6, 19,2 y 187,5 kbit/s
Velocidades de transferencia Freeport	0,3, 0,6, 1,2, 2,4, 4,8, 9,6, 19,2 y 38,4 kbit/s	0,3, 0,6, 1,2, 2,4, 4,8, 9,6, 19,2 y 38,4 kbit/s
Longitud máx. del cable por segmento hasta 38,4 kbit/s	1200 m	1200 m
187,5 kbit/s	1000 m	1000 m
Nº máximo de estaciones	32 estaciones	32 estaciones
Por segmento	126 estaciones	126 estaciones
Por red	32 maestros	32 maestros
Nº máximo de maestros	Sí	Sí
Modo maestro PPI (NETR/NETW)	4 en total; 2 reservados: 1 para PG y 1 OP	1 OP
Enlaces MPI		
<b>Cartuchos opcionales</b>		
Cartucho de memoria (almacenamiento permanente)	Programa, datos y configuración	Programa, datos y configuración
Cartucho de pila (tiempo de respaldo de datos)	típ. 200 días	típ. 200 días
<b>Alimentación</b>		
Tensión de línea (margen admisible)	DC 20,4 a 28,8 V	AC 85 a 264 V
Corriente de entrada (sólo CPU) / carga máx.	120/900 mA a DC 24 V	47 a 63 Hz 35/100 mA a AC 240 V 35/220 mA a AC 120 V
Extra-corriente de serie (máx.)	10 A a DC 28,8 V	20 A a AC 264 V
Aislamiento (corriente de entrada a lógica)	Sin aislamiento	AC 1500 V
Tiempo de retardo (desde la pérdida de corriente de entrada)	mín. 10 ms de DC 24 V	80 ms de AC 240 V, 20 ms de 120 VAC
Fusible interno (no reemplazable por el usuario)	2 A, 250 V, de acción lenta	2 A, 250 V, de acción lenta
<b>+5 alimentación para módulos de ampliación (máx.)</b>	660 mA	660 mA
<b>Alimentación para sensores DC 24 V</b>		
Margen de tensión	DC 15,4 a 28,8 V	DC 20,4 a 28,8 V
Corriente máxima	280mA	280mA
Rizado corriente parásita	Igual que línea de entrada	Menos de 1 V pico a pico (máx.)
Corriente límite	600 mA	600 mA
Aislamiento (alimentación de sensores a circuito lógico)	Sin aislamiento	Sin aislamiento

Tabla A-4 Datos técnicos de la CPU 224 DC/DC/DC y de la CPU 224 AC/DC/relé

Descripción Nº de referencia	CPU 224 DC/DC/DC 6ES7 214-1AD20-0XB0	CPU 224 AC/DC/relé 6ES7 214-1BD20-0XB0
<b>Características de las entradas</b>		
Nº de entradas integradas	14 entradas	14 entradas
Tipo de entrada	Sumidero de corriente/fuente (tipo 1 IEC)	Sumidero de corriente/fuente (tipo 1 IEC)
<b>Tensión de entrada</b>		
Tensión máx. continua admisible	DC 30 V	DC 30 V
Sobretensión transitoria	DC 35 V, 0,5 s	DC 35 V, 0,5 s
Valor nominal	DC 24 V a 4 mA, nominal	DC 24 V a 4 mA, nominal
Señal 1 lógica (mín.)	mín. DC 15 V a 2.5 mA	mín. DC 15 V a 2.5 mA
Señal 0 lógica (máx.)	máx. DC 5 V a 1 mA	máx. DC 5 V a 1 mA
<b>Aislamiento (campo a circuito lógico)</b>		
Separación galvánica	AC 500 V, 1 minuto	AC 500 V, 1 minuto
Grupos de aislamiento de	8 y 6 entradas	8 y 6 entradas
<b>Tiempos de retardo de las entradas</b>		
Entradas filtradas y entradas de interrupción	0,2 a 12,8 ms, seleccionable por el usuario	0,2 a 12,8 ms, seleccionable por el usuario
Entradas de reloj de los contadores rápidos		
Fase simple		
DC 30 V	Nivel 1 lógico = DC 15 V a 20 kHz 30 kHz	20 kHz 30 kHz
DC 26 V	Nivel 1 lógico = DC 15 V a	
Contadores A/B	10 kHz 20 kHz	10 kHz 20 kHz
DC 30 V	Nivel 1 lógico = DC 15 V a	
DC 26 V	Nivel 1 lógico = DC 15 V a	
<b>Conexión de sensor de proximidad de 2 hilos (Bero)</b>		
Corriente de fuga admisible	máx. 1 mA	máx. 1 mA
<b>Longitud del cable</b>		
No apantallado (no HSC)	300 m	300 m
Apantallado	500 m	50 m
Entradas HSC, apantalladas	50 m	50 m
<b>Nº de entradas ON simultáneamente</b>		
40 ° C	14	14
55 ° C	14	14
<b>Características de las salidas</b>		
Nº de salidas integradas	10 salidas	10 salidas
Tipo de salida	Estado sólido-MOSFET	Relé, contacto de baja potencia
<b>Tensión de salida</b>		
Margen admisible	DC 20,4 a 28,8 V	DC 5 a 30 V ó AC 5 a 250 V
Valor nominal	DC 24 V	–
Señal 1 lógica a corriente máxima	mín. DC 20 V	–
Señal 0 lógica a 10 K Ω de carga	máx. DC 0,1 V	–

Tabla A-4 Datos técnicos de la CPU 224 DC/DC/DC y de la CPU 224 AC/DC/relé

Descripción Nº de referencia	CPU 224 DC/DC/DC 6ES7 214-1AD20-0XB0	CPU 224 AC/DC/relé 6ES7 214-1BD20-0XB0
<b>Corriente de salida</b>		
Señal 1 lógica	0,75 A	2.00 A
Nº de grupos de salidas	2	3
Nº de salidas ON (máx.)	10	10
Por grupo – montaje horizontal (máx.)	5	4/3/3
Por grupo – montaje vertical (máx.)	5	4/3/3
Corriente máx. por común/grupo	3,75 A	8 A
Carga LEDs	5 W	30 W DC/200 W AC
Resistencia estado ON (resistencia contactos)	0,3Ω	0,002 Ω, máx. si son nuevas
Corriente de derivación por salida	máx. 10 μA	–
Sobrecorrientemomentánea	máx. 8 A, 100 ms	7A al estar cerrados los contactos
Protección contra sobrecargas	no	no
<b>Aislamiento (campo a circuito lógico)</b>		
Separación galvánica	AC 500 V, 1 minuto	–
Resistencia de aislamiento	–	100 M Ω, mín. si son nuevas
Aislamiento bobina a contacto	–	AC 500 V, 1 minuto
Aislamiento entre contactos abiertos	–	AC 750 V, 1 minuto
En grupos de	5 salidas	4 salidas/3 salidas/3 salidas
<b>Carga inductiva, apriete</b>		
Repetición	1 W, en todos los canales	–
disipación de energía	L+ menos 48V	–
< 0.5 LI <sup>2</sup> x frecuencia de conmutación	–	–
Límites tensión de bloqueo	–	–
<b>Retardo de las salidas</b>		
OFF a ON (Q0.0 y Q0.1)	máx. 2 μs	–
ON a OFF (Q0.0 y Q0.1)	máx. 10 μs	–
OFF a ON (Q0.2 hasta Q1.1)	máx. 15 μs	–
ON a OFF (Q0.2 hasta Q1.1)	máx. 100 μs	–
<b>Frecuencia de conmutación (salida de impulsos)</b>		
Q0.0 y I0.1	máx. 20 kHz	máx. 1 Hz
<b>Relé</b>		
Retardo de conmutación	–	máx. 10 ms
Vida útil mecánica (sin carga)	–	10.000.000 ciclos abiertos/cerrados
Vida útil contactos a carga nominal	–	100.000 ciclos abiertos/cerrados
<b>Longitud del cable</b>		
No apantallado	150 m	150 m
Apantallado	500 m	500 m

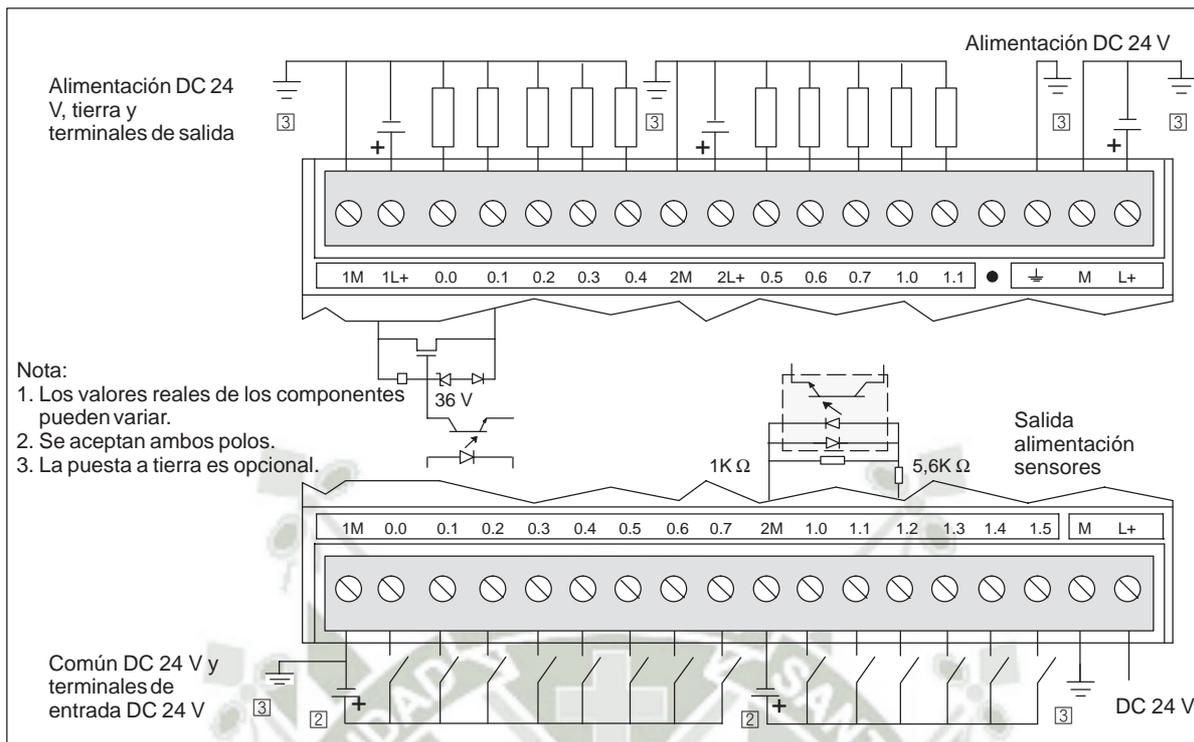


Figura A-6 Identificación de terminales de conexión para la CPU 224 DC/DC/DC

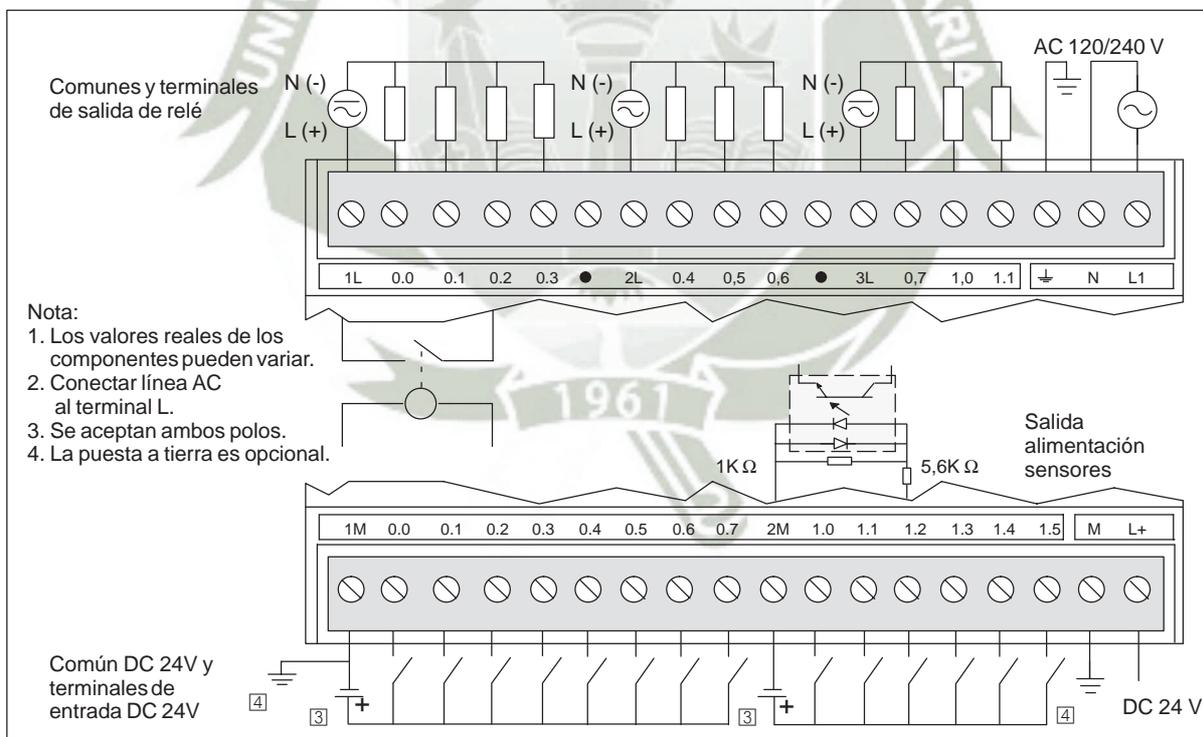


Figura A-7 Identificación de terminales de conexión para la CPU 224 AC/DC/relé

## A.5 Datos técnicos del módulo de ampliación EM221 de entradas digitales

Tabla A-5 Datos técnicos del módulo de ampliación EM221, 8 entradas digitales DC 24 V

Descripción Nº de referencia	Módulo de ampliación EM 221, 8 entradas digitales DC 24 V 6ES7 221-1BF20-0XA0
<b>Tamaño físico</b>	
Dimensiones (l x a x p)	46 x 80 x 62 mm
Peso	150 g
Pérdida de corriente (disipación)	2 W
<b>Características de las entradas</b>	
Nº de entradas integradas	8 entradas
Tipo de entrada	Sumidero de corriente/fuente (tipo 1 IEC con sumidero de corriente)
<b>Tensión de entrada</b>	
Tensión máx. continua admisible	DC 30 V
Sobretensión transitoria	DC 35 V, 0,5 s
Valor nominal	DC 24 V a 4 mA, nominal
Señal 1 lógica (mín.)	mín. DC 15 V a 2,5 mA
Señal 0 lógica (máx.)	máx. DC 5 V a 1 mA
<b>Aislamiento</b>	
Separación galvánica	AC 500 V, 1 minuto
Grupos de aislamiento de	4 entradas
<b>Tiempos de retardo de las entradas</b>	
Máximo	4.5 ms
<b>Conexión de sensor de proximidad de 2 hilos (Bero)</b>	
Corriente de fuga admisible	máx. 1 mA
<b>Longitud del cable</b>	
No apantallado	300 m
Apantallado	500 m
<b>Nº de entradas ON simultáneamente</b>	
40 ° C	8
55 ° C	8
<b>Consumo de corriente</b>	
De +DC 5 V (del bus de ampliación)	30 mA

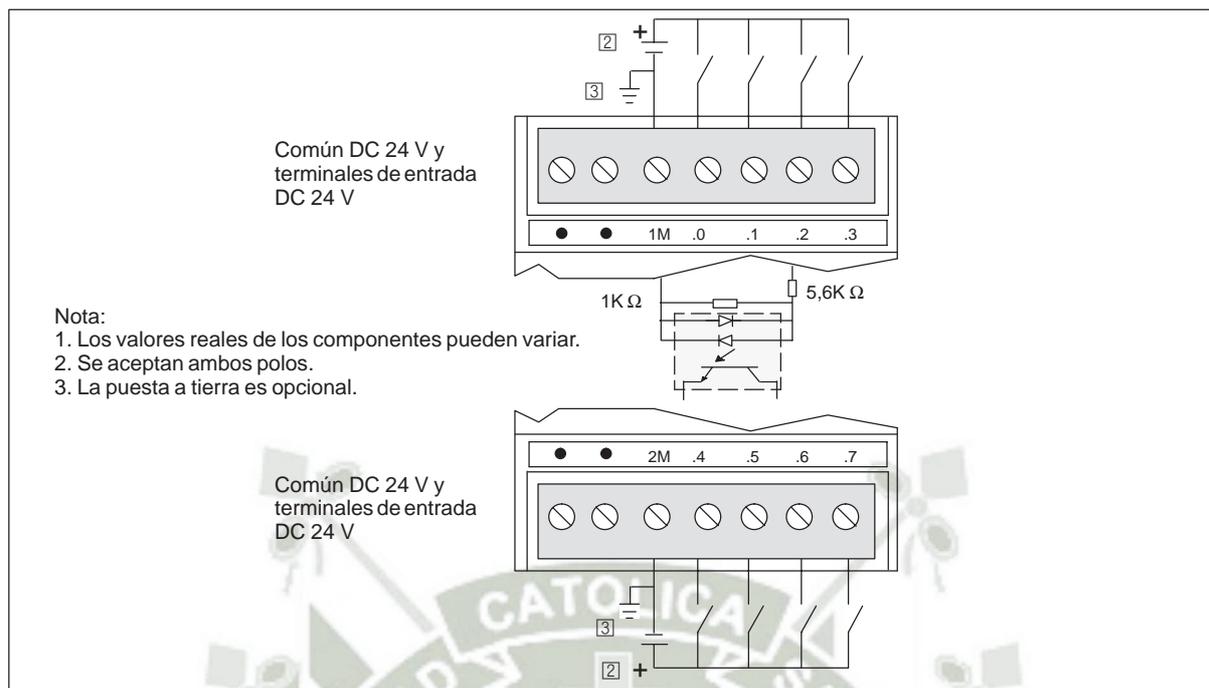


Figura A-8 Identificación de terminales de conexión para el EM221, 8 entradas digitales x DC 24 V

## A.6 Datos técnicos de los módulos de ampliación EM222 de salidas digitales

Tabla A-6 Datos técnicos de los módulos EM222, salidas DC 24 V y salidas de relé

Descripción Nº de referencia	EM222, salidas DC 24 V 6ES7 222-1BF20-0XA0	EM222, salidas de relé 6ES7 222-1HF20-0XA0
<b>Tamaño físico</b>		
Dimensiones (l x a x p)	46 x 80 x 62 mm	46 x 80 x 62 mm
Peso	150 g	170 g
Pérdida de corriente (disipación)	2 W	2 W
<b>Características de las salidas</b>		
Nº de salidas	8 salidas	8 salidas
Tipo de salida	Estadosólido-MOSFET	Relé, contacto de baja potencia
<b>Tensión de salida</b>		
Margenadmisible	DC 20,4 a 28,8 V	DC 5 a 30 V ó AC 5 a 250 V
Valor nominal	DC 24 V	—
Señal 1 lógica a corriente máxima	mín. DC 20 V	—
Señal lógica 0 con 10 K Ω de carga	máx. DC 0,1 V	—
<b>Corriente de salida</b>		
Señal 1 lógica	0,75 A	2.00 A
Nº de grupos de salidas	2	2
Nº de salidas ON (máx.)	8	8
Por grupo – montaje horizontal (máx.)	4	4
Por grupo – montaje vertical (máx.)	4	4
Corriente máx. por común/grupo	3 A	8 A
Carga LEDs	5 W	30 W DC/200 W AC
Resistencia estado ON (resistencia conta.)	0,3Ω	0,002 Ω, máx. si son nuevas
Corriente de derivación por salida	máx. 10 μA	—
Sobrecorrientemomentánea	máx. 8 A, 100 ms	7A al estar cerrados los contactos
Protección contra sobrecargas	no	no
<b>Aislamiento</b>		
Separación galvánica	AC 500 V, 1 minuto	—
Resistencia de aislamiento	—	100 M Ω, mín. si son nuevas
Aislamiento bobina a contacto	—	AC 500 V, 1 minuto
Aislamiento entre contactos abiertos	—	AC 750 V, 1 minuto
En grupos de	4 salidas	4 salidas
<b>Carga inductiva, apriete</b>		
Repetición disipación de energía < 0.5 LI <sup>2</sup> x frecuencia de conmutación	1 W, en todos los canales	—
Límites tensión de bloqueo	L+ menos 48 V	—
<b>Retardo de las salidas</b>		
OFF a ON	máx. 50 μs	—
ON a OFF	máx. 200 μs	—
<b>Relé</b>		
Retardo de conmutación	—	máx. 10 ms
Vida útil mecánica (sin carga)	—	10.000.000 ciclos abiertos/cerrados
Vida útil contactos a carga nominal	—	100.000 ciclos abiertos/cerrados
<b>Longitud del cable</b>		
Noapantallado	150 m	150 m
Apantallado	500 m	500 m
<b>Consumo de corriente</b>		
De +DC 5 V (del bus de ampliación)	50 mA	40 mA
De L+	—	9 mA por salida en ON

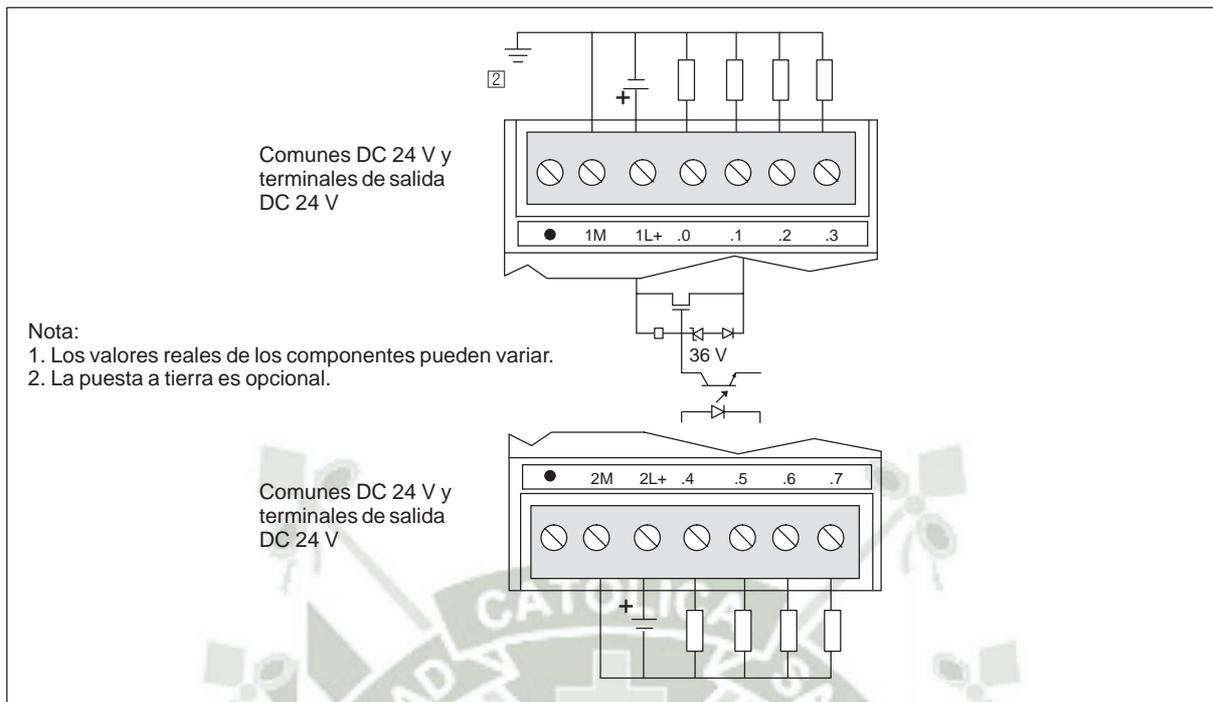


Figura A-9 Identificación de terminales de conexión para el EM222, 8 salidas digitales x DC 24 V

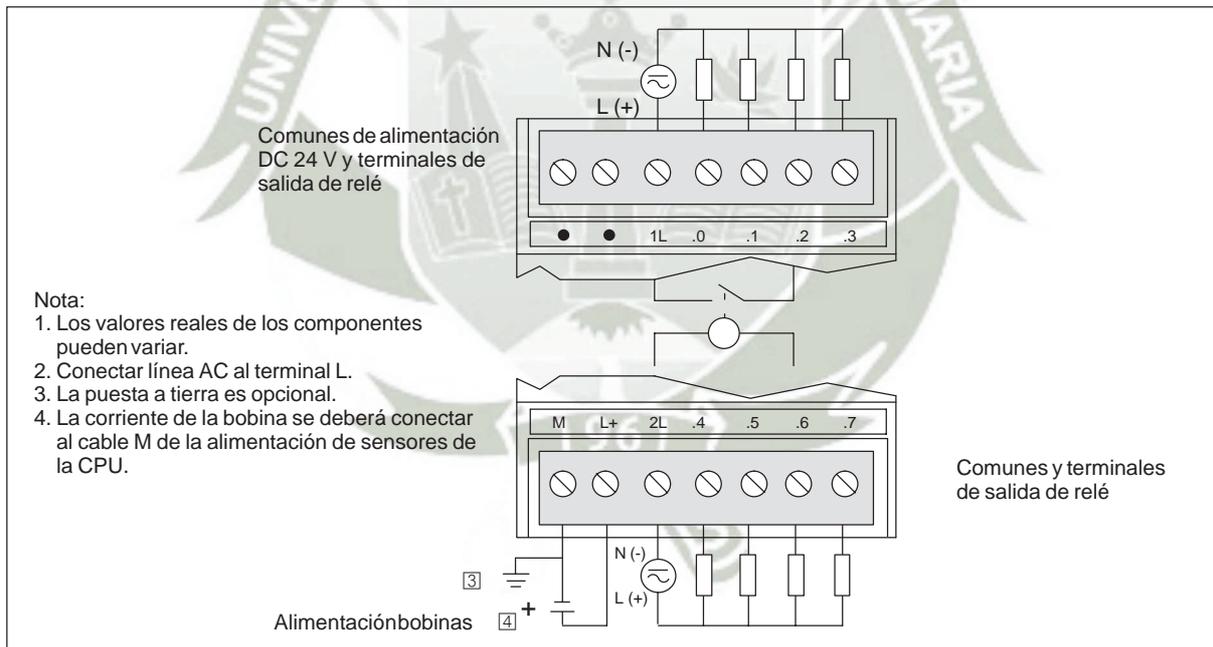


Figura A-10 Identificación de terminales de conexión para el EM222, 8 salidas digitales x relé

## A.7 Datos técnicos de los módulos de ampliación EM223, 8 entradas digitales/8 salidas digitales

Tabla A-7 Datos técnicos del EM223, 8 entradas/8 salidas x DC 24 V y del EM223, 8 entradas DC 24 V/ 8 salidas de relé

Descripción Nº de referencia	EM223 entradas/salidas DC 24 V 6ES7223-1BH20-0XA0	EM223 entradas DC 24V/salidas de relé 6ES7223-1PH20-0XA0
<b>Tamaño físico</b>		
Dimensiones (l x a x p)	71,2 mm x 80 mm x 62 mm	71,2 mm x 80 mm x 62 mm
Peso	200 g	300 g
Pérdida de corriente (disipación)	3 W	3 W
<b>Características de las entradas</b>		
Nº de entradas	8 entradas	8 entradas
Tipo de entrada	Sumidero de corriente/fuente (tipo 1 IEC con sumidero de corriente)	Sumidero de corriente/fuente (tipo 1 IEC con sumidero de corriente)
<b>Tensión de entrada</b>		
Tensión máx. continua admisible	DC 30 V	DC 30 V
Sobretensión transitoria	DC 35 V, 0,5 s	DC 35 V, 0,5 s
Valor nominal	DC 24 V a 4 mA, nominal	DC 24 V a 4 mA, nominal
Señal 1 lógica (mín.)	mín. DC 15 V a 2.5 mA	mín. DC 15 V a 2.5 mA
Señal 0 lógica (máx.)	máx. DC 5 V a 1 mA	máx. DC 5 V a 1 mA
<b>Aislamiento</b>		
Separación galvánica	AC 500 V, 1 minuto	AC 500 V, 1 minuto
Grupos de aislamiento de	4 entradas	4 entradas
<b>Tiempos de retardo de las entradas</b>		
Máximo	4.5 ms	4.5 ms
<b>Conexión de sensor de proximidad de dos hilos (Bero)</b>		
Máximo	máx. 1 mA	máx. 1 mA
<b>Longitud del cable</b>		
No apantallado	300 m	300 m
Apantallado	500 m	500 m
<b>Nº de entradas ON simultáneamente</b>		
40 ° C	8	8
55 ° C	8	8
<b>Características de las salidas</b>		
Nº de salidas integradas	8 salidas	8 salidas
Tipo de salida	Estado sólido-MOSFET	Relé, contacto de baja potencia
<b>Tensión de salida</b>		
Margen admisible	DC 20,4 a 28,8 V	DC 5 a 30 V ó AC 5 a 250 V
Valor nominal	DC 24 V	—
Señal 1 lógica a corriente máxima	mín. DC 20 V	—
Señal 0 lógica con 10K $\Omega$ de carga	máx. DC 0,1 V	—

Tabla A-7 Datos técnicos del EM223, 8 entradas/8 salidas x DC 24 V y del EM223, 8 entradas DC 24 V/  
8 salidas de relé

Descripción Nº de referencia	EM223 entradas/salidas DC 24 V 6ES7223-1BH20-0XA0	EM223 entradas DC 24V/salidas de relé 6ES7223-1PH20-0XA0
<b>Corriente de salida</b>		
Señal 1 lógica	0.5 A	2.00 A
Nº de grupos de salidas	2	2
Nº de salidas ON (máx.)	8	8
Por grupo – montaje horizontal (máx.)	4	4
Por grupo – montaje vertical (máx.)	4	4
Corriente máx. por común/grupo	2 A	8 A
Carga LEDs	5 W	30 W DC/200 W AC
Resistencia estado ON (resistencia contactos)	0.3 Ω máx. 10 µA	0,002 Ω, máx. si son nuevas –
Corriente de derivación por salida	máx. 8 A, 100 ms	7A al estar cerrados los contactos
Sobrecorriente momentánea	no	no
Protección contra sobrecargas		
<b>Aislamiento</b>		
Separación galvánica	AC 500 V, 1 minuto	–
Resistencia de aislamiento	–	100 M Ω, mín. si son nuevas
Aislamiento bobina a contacto	–	AC 500 V, 1 minuto
Aislamiento entre contactos abiertos	–	AC 750 V, 1 minuto
En grupos de	4 salidas	4 salidas
<b>Carga inductiva, apriete</b>		
Repetición		
disipación de energía	1 W, en todos los canales	–
< 0.5 LI <sup>2</sup> x frecuencia de conmutación	L+ menos 48V	–
Límites tensión de bloqueo		
<b>Retardo de las salidas</b>		
OFF a ON	máx. 50 µs	–
ON a OFF	máx. 200 µs	–
<b>Relé</b>		
Retardo de conmutación	–	máx. 10 ms
Vida útil mecánica (sin carga)	–	100.000.000 ciclos abiertos/cerrados
Vida útil contactos a carga nominal	–	100.000 ciclos abiertos/cerrados
<b>Longitud del cable</b>		
No apantallado	150 m	150 m
Apantallado	500 m	500 m
<b>Consumo de corriente</b>		
De +DC 5 V (del bus de ampliación)	100 mA	80 mA
De L+	–	9 mA por salida en ON

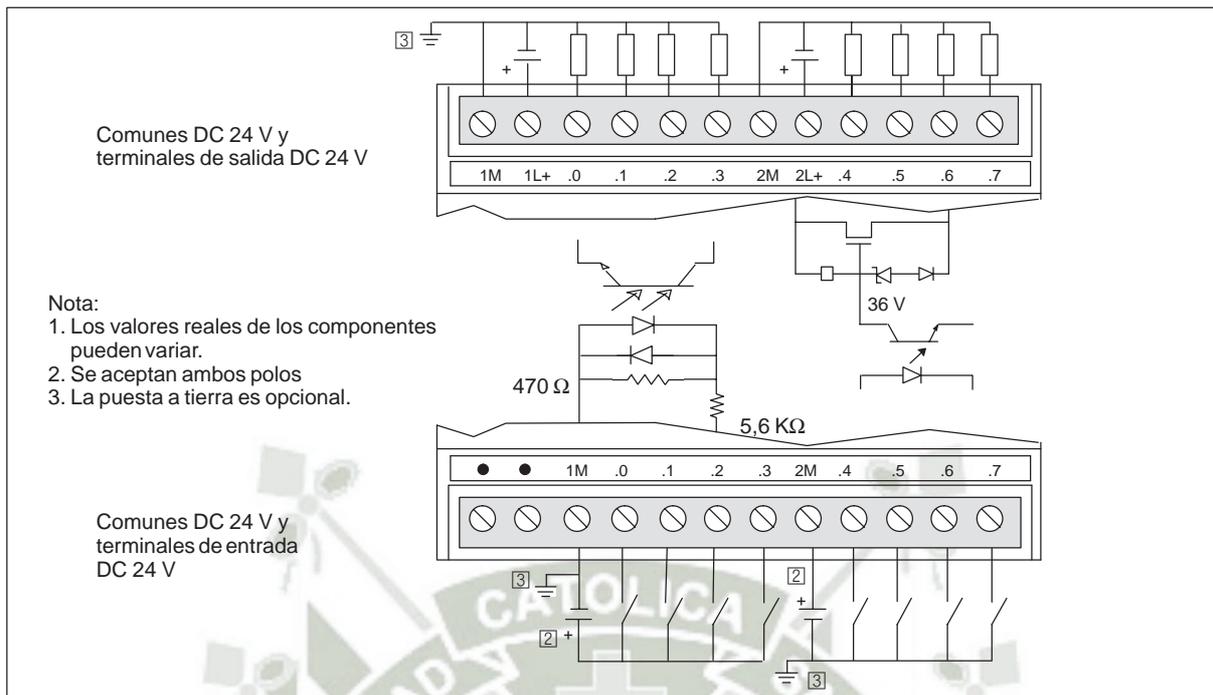


Figura A-11 Identificación de terminales de conexión para el EM223 8 entradas digitales x DC 24 V/8 salidas digitales x DC 24 V

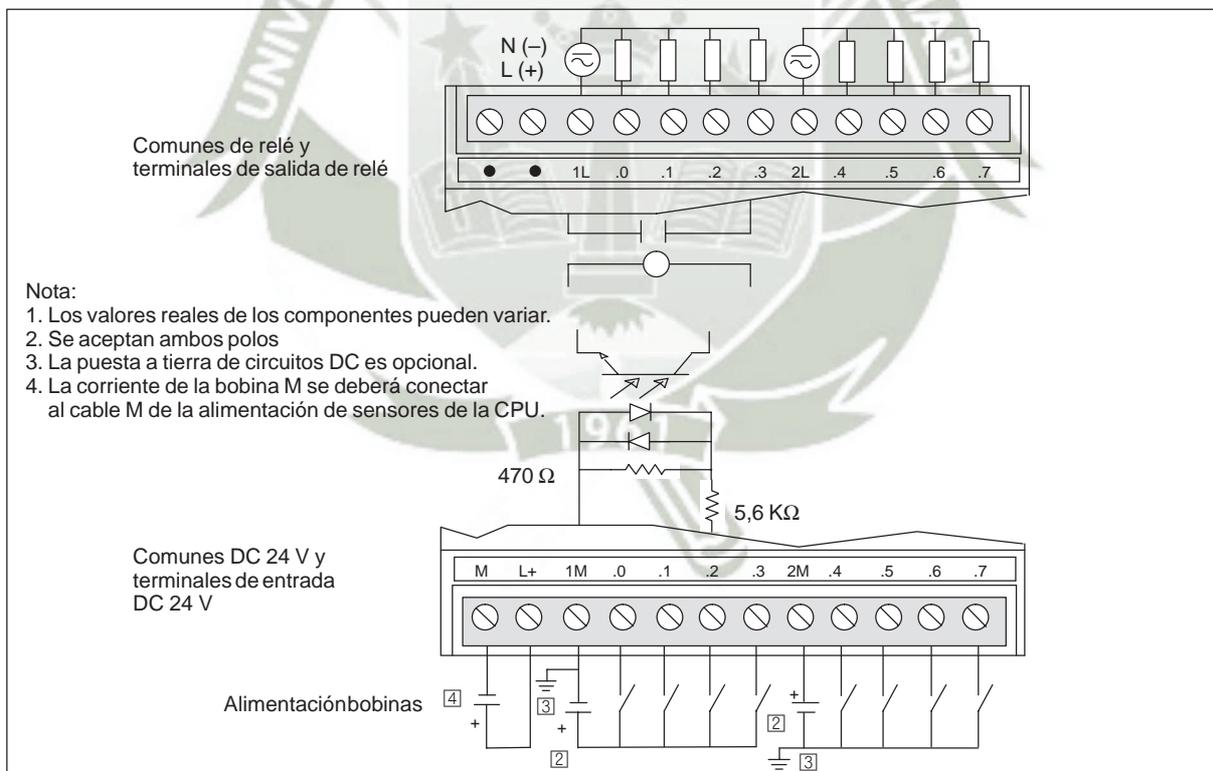
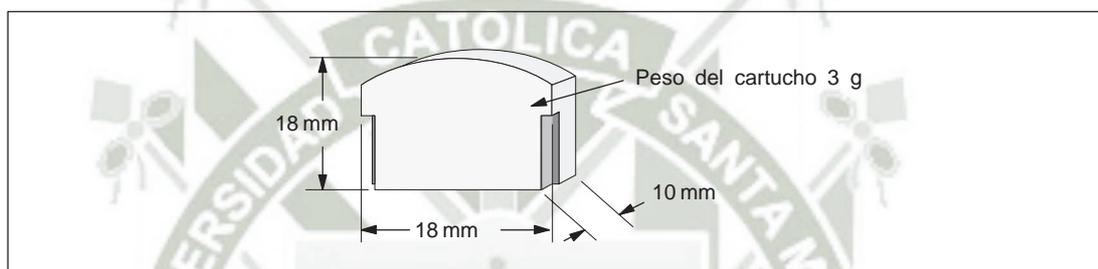


Figura A-12 Identificación de terminales de conexión para el EM223, 8 entradas digitales DC 24 V / 8 salidas digitales de relé

## A.8 Cartuchos opcionales

Nº de referencia	Color	Función del cartucho
6ES7 291 8GE20 0XA0	Gris	Programa de usuario
6ES7 297 1AA20 0XA0	Azul	Reloj de tiempo real con pila
6ES7 291 8BA20 0XA0	Naranja	Cartucho de pila

Cartuchos opcionales	
Almacenamiento en el cartucho de memoria	Programa, datos y configuración
Cartucho de pila (tiempo de respaldo de datos)	típ. 200 días
Precisión del cartucho de reloj	2 minutos por mes @ 25°C 7 minutos por mes @ 0°C a 55°C



Características generales	
Pila	3 V, 30 mA hora, Renata CR 1025
Tamaño	9.9 x 2.5 mm
Tipo de entrada	Litio < 0.6 g
Vida útil de almacenaje	10 años

## A.9 Cable de módulo de ampliación

Nº de referencia 6ES7 290-6AA20-0XA0

Características generales	
Longitud del cable	0,8 m
Peso	25 g
Tipo de conector	Cinta de 10 pines

### Instalación típica del cable de conexión de E/S de ampliación

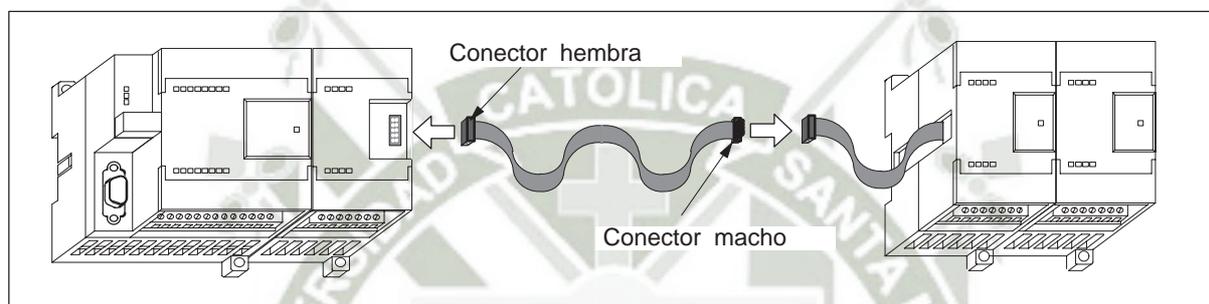


Figura A-13 Instalación típica del cable de conexión de E/S de ampliación

#### Nota

En una cadena de CPU/módulo de ampliación es aconsejable utilizar un solo cable de ampliación.

## A.10 Cable PC/PPI

Nº de referencia 6ES7 901-3BF20-0XA0

### Dimensiones del cable PC/PPI

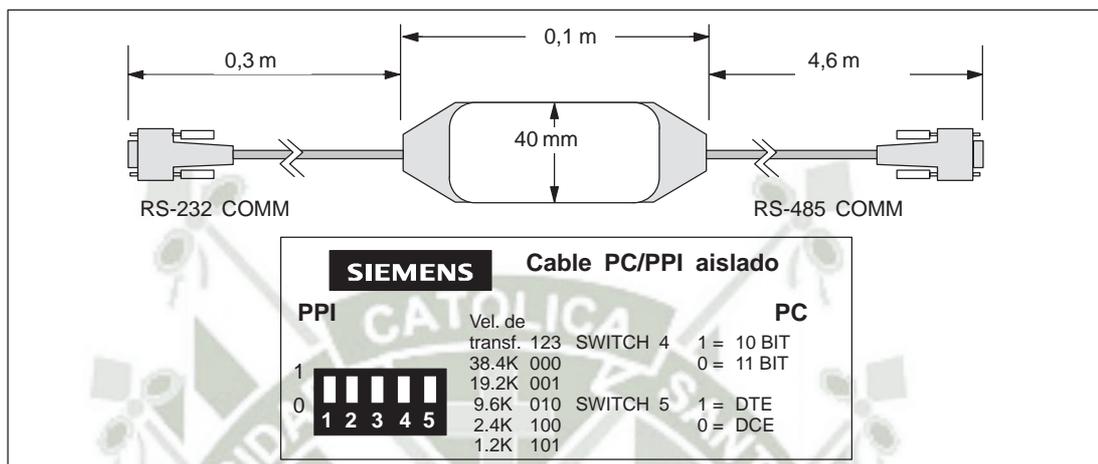


Figura A-14 Dimensiones del cable PC/PPI

Tabla A-8 Posición de los interruptores DIP en el cable PC/PPI para seleccionar la velocidad de transferencia

Velocidad de transferencia	Interruptor DIP (1 = arriba)
38400	000
19200	001
9600	010
4800	011
2400	100
1200	101
600	110

Tabla A-9 Utilización de módems con el cable PC/PPI

Tipo de módem	Interruptor DIP (1 = arriba)
Módem de 11 bits	0
Módem de 10 bits	1

Tabla A-10 Asignación de pines del cable PC/PPI

Asignación de pines	Interruptor DIP (1 = arriba)
DCE	0
DTE	1

Tabla A-11 Asignación de pines para un conector de RS-485 a RS-232 DCE

Asignación de pines del conector RS-485		Asignación de pines del conector RS-232 DCE	
Nº de pin	Descripción de la señal	Nº de pin	Descripción de la señal
1	Tierra (RS-485)	1	Data Carrier Detect (DCD) (no utilizado)
2	Hilo de retorno 24 V (tierra RS-485)	2	Receive Data (RD) (salida del cable PC/PPI)
3	Señal B (RxD/TxD+)	3	Transmit Data (TD) (entrada al cable PC/PPI)
4	RTS (nivel TTL)	4	Data Terminal Ready (DTR) (no utilizado)
5	Tierra (RS-485)	5	Tierra (RS-232)
6	+5 V (con resistor en serie de 100 $\Omega$ )	6	Data Set Ready (DSR) (no utilizado)
7	Alimentación 24 V	7	Request To Send (RTS) (no utilizado)
8	Señal A (RxD/TxD-)	8	Clear To Send (CTS) (no utilizado)
9	Selección de protocolo	9	Ring Indicator (RI) (no utilizado)

Tabla A-12 Asignación de pines para un conector de RS-485 a RS-232 DTE

Asignación de pines del conector RS-485		Asignación de pines del conector RS-232 DTE <sup>1</sup>	
Nº de pin	Descripción de la señal	Nº de pin	Descripción de la señal
1	Tierra (RS-485)	1	Data Carrier Detect (DCD) (no utilizado)
2	Hilo de retorno 24 V (tierra RS-485)	2	Receive Data (RD) (entrada al cable PC/PPI)
3	Señal B (RxD/TxD+)	3	Transmit Data (TD) (salida del cable PC/PPI)
4	RTS (nivel TTL)	4	Data Terminal Ready (DTR) (no utilizado)
5	Tierra (RS-485)	5	Tierra (RS-232)
6	+5 V (con resistor en serie de 100 $\Omega$ )	6	Data Set Ready (DSR) (no utilizado)
7	Alimentación 24 V	7	Request To Send (RTS) (salida del cable PC/PPI)
8	Señal A (RxD/TxD-)	8	Clear To Send (CTS) (no utilizado)
9	Selección de protocolo	9	Ring Indicator (RI) (no utilizado)

<sup>1</sup> Para los módems se debe efectuar una conversión de conector hembra a conectar macho y de 9 pines a 25 pines.

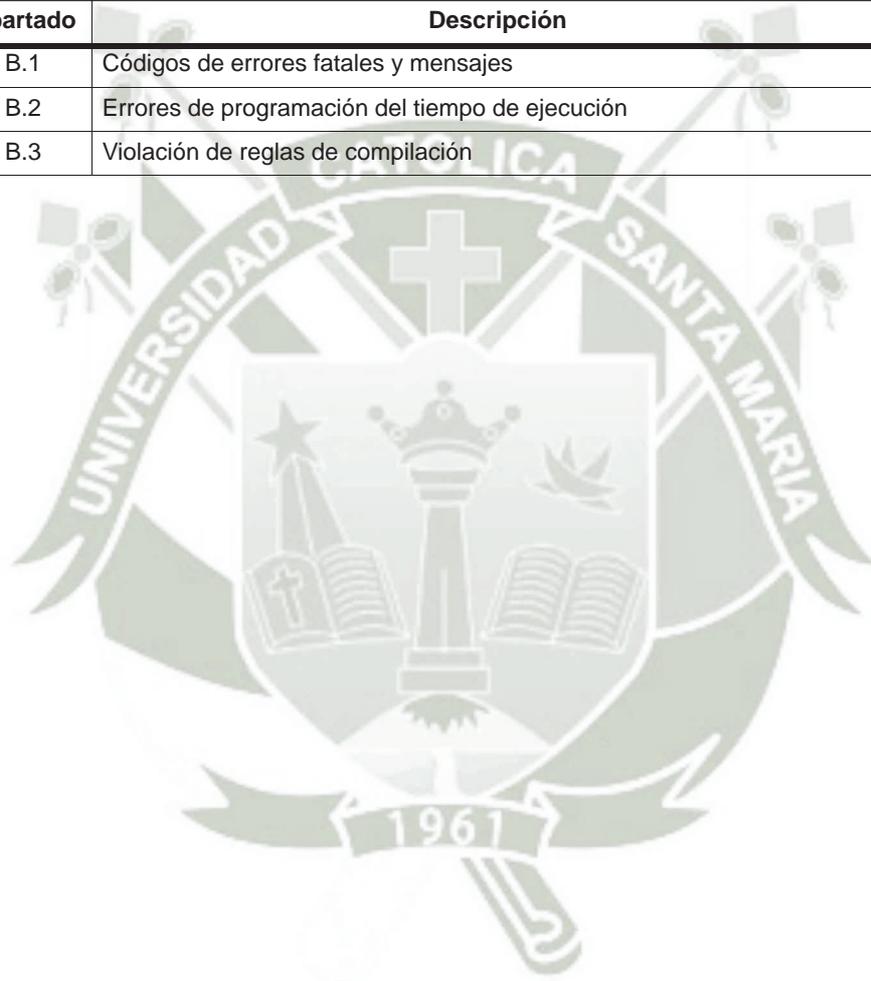
## Códigos de error

# B

La información relativa a los códigos de error permite identificar rápidamente los problemas que se hayan presentado en la CPU S7-200.

### Índice del capítulo

Apartado	Descripción	Página
B.1	Códigos de errores fatales y mensajes	B-2
B.2	Errores de programación del tiempo de ejecución	B-3
B.3	Violación de reglas de compilación	B-4



## B.1 Códigos de errores fatales y mensajes

Cuando ocurre un error fatal, la CPU detiene la ejecución del programa. Dependiendo de la gravedad del error, es posible que la CPU no pueda ejecutar todas las funciones, o incluso ninguna de ellas. El objetivo del tratamiento de errores fatales es conducir a la CPU a un estado seguro, en el que se puedan analizar y eliminar las condiciones que hayan causado el error.

Cuando la CPU detecta un error fatal:

- Cambia a modo STOP.
- Se encienden los indicadores "SF" (fallo del sistema) y "STOP".
- Se desactivan las salidas.

La CPU permanece en dicho estado hasta que se elimine la causa del error fatal. La tabla B-1 muestra una lista con las descripciones de los códigos de errores fatales que se pueden leer de la CPU.

Tabla B-1 Códigos de errores fatales y mensajes

Código de error	Descripción
0000	No hay errores fatales
0001	Error de suma de verificación en el programa de usuario
0002	Error de suma de verificación en el programa KOP compilado
0003	Error de tiempo en la vigilancia del tiempo de ciclo (watchdog)
0004	Error EEPROM interno
0005	Error EEPROM interno de suma de verificación en el programa de usuario
0006	Error EEPROM interno de suma de verificación en los parámetros de configuración
0007	Error EEPROM interno de suma de verificación en los datos forzados
0008	Error EEPROM interno de suma de verificación en los valores predeterminados de la imagen de proceso de las salidas
0009	Error EEPROM interno de suma de verificación en los datos de usuario, DB1
000A	Error en el cartucho de memoria
000B	Error de suma de verificación del cartucho de memoria en el programa de usuario
000C	Error de suma de verificación del cartucho de memoria en los parámetros de configuración
000D	Error de suma de verificación del cartucho de memoria en los datos forzados
000E	Error de suma de verificación del cartucho de memoria en los valores predeterminados de la imagen de proceso de las salidas
000F	Error de suma de verificación del cartucho de memoria en los datos de usuario, DB1
0010	Error interno de software
0011	Error en el direccionamiento indirecto del contacto de comparación
0012	Valor en coma flotante no válido en el contacto de comparación
0013	Cartucho de memoria vacío o programa no apto para esta CPU

## B.2 Errores de programación del tiempo de ejecución

Durante la ejecución normal del programa se pueden presentar errores no fatales (p.ej. errores de direccionamiento). La CPU genera entonces un código de error no fatal de tiempo de ejecución. La tabla B-2 muestra una lista con las descripciones de los errores no fatales.

Tabla B-2 Errores de programación del tiempo de ejecución

Código de error	Error de programación del tiempo de ejecución (no fatal)
0000	No se presentó ningún error.
0001	Cuadro HSC habilitado antes de ejecutar el cuadro HDEF.
0002	Interrupción de entrada asignada a una entrada que ya está asociada a un contador rápido (conflicto).
0003	Entrada asignada a un contador rápido que ya está asociado a una interrupción de entrada u otro contador rápido (conflicto).
0004	Se ha intentado ejecutar una operación ENI, DISI o HDEF en una rutina de interrupción.
0005	Antes de finalizar el primer HSC/PLC se ha intentado ejecutar un segundo HSC con el mismo número (HSC/PLS de la rutina de interrupción en conflicto con HSC/PLC del programa principal).
0006	Error de direccionamiento indirecto.
0007	Error en datos para operación TODW (Escribir en reloj de tiempo real) o TODR (Leer del reloj de tiempo real).
0008	Excedida la profundidad máxima de anidado para subrutina de usuario.
0009	Ejecución de una operación XMT ó RCV simultáneamente con otra operación XMT o RCV en el puerto 0.
000A	Se ha intentado redefinir un HSC ejecutando otra operación HDEF para el mismo HSC.
000B	Ejecución simultánea de las operaciones XMT/RCV en el puerto 1.
000C	Falta cartucho de reloj.
000D	Intento de redefinir la salida de impulsos mientras está activada.
000E	El número de segmento del perfil PTO se ha puesto a 0.
0091	Error de margen (con información sobre direcciones): verificar las áreas de operandos.
0092	Error en el campo de contaje de una operación (con información sobre el contaje): verificar el valor máximo de contaje.
0094	Error de margen al escribir en la memoria no volátil (con información sobre direcciones).
009A	Intento de cambiar a modo Freeport en una interrupción de usuario.

### B.3 Violación de reglas de compilación

Al cargar un programa en la CPU, ésta lo compila. Si durante la compilación se detecta una violación de las reglas (p.ej. una operación no válida), la CPU detendrá el proceso de carga, generando entonces un código de error no fatal (de violación de las reglas de compilación). En la tabla B-3 se describen los códigos de error generados al violarse las reglas de compilación.

Tabla B-3 Violación de reglas de compilación

Código de error	Error de compilación (no fatal)
0080	Programa demasiado extenso para la compilación: reducir el tamaño del programa.
0081	Rebase negativo de la pila: dividir el segmento en varios segmentos.
0082	Operación no válida: comprobar la nemotécnica de la operación.
0083	Falta MEND u operación no admisible en el programa principal: agregar la operación MEND o borrar la operación incorrecta.
0084	Reservados
0085	Falta FOR: agregar la operación FOR o borrar la operación NEXT.
0086	Falta NEXT: agregar la operación NEXT o borrar la operación FOR.
0087	Falta meta (LBL, INT, SBR): agregar la meta apropiada.
0088	Falta RET u operación no admisible en una subrutina: agregar RET al final de la subrutina o borrar la operación incorrecta.
0089	Falta RETI u operación no admisible en una rutina de interrupción: agregar RETI al final de la rutina de interrupción o borrar la operación incorrecta.
008A	Reservados
008B	Reservados
008C	Meta doble (LBL, INT, SBR): cambiar el nombre de una de las metas.
008D	Meta no válida (LBL, INT, SBR): asegurarse de que el número admisible de metas no se haya excedido.
0090	Parámetro no válido: comprobar los parámetros admisibles para la operación.
0091	Error de margen (con información sobre direcciones): verificar las áreas de operandos.
0092	Error en el campo de contaje de una operación (con información sobre el contaje): verificar el valor máximo de contaje.
0093	Excedida la profundidad de anidado FOR/NEXT.
0095	Falta la operación LSCR (cargar SCR).
0096	Falta la operación SCRE (fin de SCR) u operación no admisible antes de la operación SCRE
0097	El programa de usuario contiene operaciones EU/ED con y sin número.
0098	Intento de editar durante el tiempo de ejecución un programa con operaciones EU/ED sin número.
0099	Demasiados segmentos ocultos.

# C

## Marcas especiales (SM)

Las marcas especiales (SM) ofrecen una serie de funciones de estado y control. Sirven para intercambiar informaciones entre la CPU y el programa, pudiéndose utilizar en formato de bits, bytes, palabras o palabras dobles.

### SMB0: Bits de estado

Como muestra la tabla C-1, SMB0 contiene ocho bits de estado que la CPU S7-200 actualiza al final de cada ciclo.

Tabla C-1 Byte de marcas SMB0 (SM0.0 a SM0.7)

Bits de marcas	Descripción
SM0.0	Este bit siempre está activado.
SM0.1	Este bit se activa en el primer ciclo. Se utiliza p.ej. para llamar una subrutina de inicialización.
SM0.2	Este bit se activa durante un ciclo si se pierden los datos remanentes. Se puede utilizar como marca de error o como mecanismo para llamar a una secuencia especial de arranque.
SM0.3	Este bit se activa durante un ciclo cuando se pasa a modo RUN tras conectarse la alimentación. Se puede utilizar durante el tiempo de calentamiento de la instalación antes del funcionamiento normal.
SM0.4	Este bit ofrece un reloj que está activado durante 30 segundos y desactivado durante 30 segundos, siendo el tiempo de ciclo de 1 minuto. Ofrece un retardo fácil de utilizar o un tiempo de reloj de 1 minuto.
SM0.5	Este bit ofrece un reloj que está activado durante 0.5 segundos y desactivado durante 0.5 segundos, siendo el tiempo de ciclo de 1 segundo. Ofrece un reloj que está activado durante 0,5 segundos y desactivado durante 0,5 segundos, siendo el tiempo de ciclo de 1 minuto.
SM0.6	Este bit es un reloj de ciclo que está activado en un ciclo y desactivado en el ciclo siguiente. Se puede utilizar como entrada de conteo de ciclos.
SM0.7	Este bit indica la posición del selector de modos de operación (OFF = TERM; ON = RUN). Si el bit se utiliza para habilitar el modo Freeport cuando el selector esté en posición RUN, se podrá habilitar la comunicación normal con la unidad de programación cambiando el selector a TERM.

### SMB1: Bits de estado

Como muestra la tabla C-2, SMB1 contiene varios indicadores de los posibles errores. Estos bits son activados y desactivados por instrucciones durante el tiempo de ejecución.

Tabla C-2 Byte de marcas SMB1 (SM1.0 a SM1.7)

Bits de marcas	Descripción
SM1.0	Este bit se activa al ejecutarse ciertas operaciones si el resultado lógico es cero.
SM1.1	Este bit se activa al ejecutarse ciertas operaciones si se produce un desbordamiento o si se detecta un valor numérico no válido.
SM1.2	Este bit se activa si el resultado de una operación aritmética es negativo.
SM1.3	Este bit se activa si se intenta dividir por cero.
SM1.4	Este bit se activa si la operación Registrar valor en tabla intenta sobrepasar el límite de llenado de la tabla.
SM1.5	Este bit se activa si las operaciones FIFO o LIFO intentan leer de una tabla vacía.
SM1.6	Este bit se activa si se intenta convertir un valor no BCD en un valor binario.
SM1.7	Este bit se activa si un valor ASCII no se puede convertir en un valor hexadecimal válido.

### SMB2: Búfer de recepción de caracteres en modo Freeport

SMB2 es el búfer de recepción de caracteres en modo Freeport. Como muestra la tabla C-3, cada carácter recibido en dicho modo se deposita en este búfer, fácilmente accesible desde el programa KOP.

Tabla C-3 Byte de marcas SMB2

Byte de marcas	Descripción
SMB2	Este byte contiene todos los caracteres recibidos de los puertos 0 ó 1 en modo Freeport.

### SMB3: Error de paridad en modo Freeport

SMB3 se utiliza para el modo Freeport y contiene un bit de error de paridad que se activa si se detecta un error de este tipo en un carácter recibido. Como muestra la tabla C-4, SM3.0 se activa si se detecta un error de paridad. Utilice esta marca para rechazar el mensaje.

Tabla C-4 Byte de marcas SMB3 (SM3.0 a SM3.7)

Bits de marcas	Descripción
SM3.0	Error de paridad del puerto 0 ó 1 (0 = sin error; 1 = error)
SM3.1 a SM3.7	Reservados

### SMB4: Desbordamiento de la cola de espera

Como muestra la tabla C-5, SMB4 contiene los bits de desbordamiento de la cola de espera, un indicador de estado que muestra las interrupciones habilitadas o inhibidas y una marca de transmisor en vacío. Los bits de desbordamiento de la cola de espera indican que las interrupciones se están presentando más rápidamente de lo que se pueden procesar, o bien que se inhibieron mediante la operación Inhibir todos los eventos de interrupción (DISI).

Tabla C-5 Byte de marcas SMB4 (SM4.0 a SM4.7)

Bits de marcas	Descripción
SM4.0 <sup>1</sup>	Este bit se activa si se desborda la cola de espera para las interrupciones de comunicación.
SM4.1 <sup>1</sup>	Este bit se activa si se desborda la cola de espera para las interrupciones de E/S.
SM4.2 <sup>1</sup>	Este bit se activa si se desborda la cola de espera para las interrupciones temporizadas.
SM4.3	Este bit se activa si se detecta un error de programación del tiempo de ejecución.
SM4.4	Este bit refleja el estado de habilitación de las interrupciones. Se activa cuando se habilitan las interrupciones.
SM4.5	Este bit se activa si el transmisor está en vacío (puerto 0).
SM4.6	Este bit se activa si el transmisor está en vacío (puerto 1).
SM4.7	Este bit se activa al forzarse un valor.

<sup>1</sup> Utilice los bits de estado 4.0, 4.1 y 4.2 sólo en rutinas de interrupción. Dichos bits se desactivan cuando se vacía la cola de espera, retornando entonces el control al programa principal.

### SMB5: Estado de las entradas y salidas

Como muestra la tabla C-6, SMB5 contiene los bits de estado acerca de las condiciones de error detectadas en las entradas y salidas (E/S). Estos bits muestran los errores de E/S detectados.

Tabla C-6 Byte de marcas SMB5 (SM5.0 a SM5.7)

Bits de marcas	Descripción
SM5.0	Este bit se activa si se presenta algún error de E/S.
SM5.1	Este bit se activa si se han conectado demasiadas entradas y salidas digitales al bus E/S.
SM5.2	Este bit se activa si se han conectado demasiadas entradas y salidas analógicas al bus E/S.
SM5.3 a SM5.6	Reservados
SM5.7	Este bit se activa si se presenta un fallo de bus estándar DP.

### SMB6: Identificador de la CPU

Como muestra la tabla C-7, SMB6 es el identificador de la CPU. SM6.4 a SM6.7 indican el tipo de CPU. SM6.0 a SM6.3 están reservados para un uso futuro.

Tabla C-7 Byte de marcas SMB6

Bits de marcas	Descripción								
Formato	<div style="display: flex; justify-content: space-between; align-items: center;"> <span>MSB</span> <span>LSB</span> </div> <div style="display: flex; justify-content: space-between; align-items: center; margin-top: 5px;"> <span>7</span> <span>0</span> </div> <div style="display: flex; justify-content: center; align-items: center; margin-top: 5px;"> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr> <td>X</td><td>X</td><td>X</td><td>X</td><td>r</td><td>r</td><td>r</td><td>r</td> </tr> </table> <span>Identificador de la CPU</span> </div>	X	X	X	X	r	r	r	r
X	X	X	X	r	r	r	r		
SM6.4 a SM6.7	xxxx = 0000 = CPU 212/CPU 222 0010 = CPU 214/CPU 224 0110 = CPU 221 1000 = CPU 215 1001 = CPU 216								
SM6.0 a SM6.3	Reservados								

### SMB7: Reservado

SMB7 está reservado para un uso futuro.





### SMW22 a SMW26: Tiempos de ciclo

Como muestra la tabla C-9, las marcas especiales SMW22, SMW24, y SMW26 informan sobre el tiempo de ciclo. Permiten leer el último tiempo de ciclo, así como los tiempos de ciclo mínimo y máximo.

Tabla C-9 Palabras de marcas SMW22 a SMW26

Palabra de marcas	Descripción
SMW22	Esta palabra indica el tiempo del último ciclo.
SMW24	Esta palabra indica el tiempo de ciclo mínimo.
SMW26	Esta palabra indica el tiempo de ciclo máximo.

### SMB28 y SMB29: Potenciómetros analógicos

Como muestra la tabla C-10, SMB28 almacena el valor digital que representa la posición del potenciómetro analógico 0. SMB29 almacena el valor digital que representa la posición del potenciómetro analógico 1.

Tabla C-10 Bytes de marcas SMB28 y SMB29

Byte de marcas	Descripción
SMB28	Este byte almacena el valor leído del potenciómetro analógico 0. Dicho valor se actualiza una vez por ciclo en STOP/RUN.
SMB29	Este byte almacena el valor leído del potenciómetro analógico 1. Dicho valor se actualiza una vez por ciclo en STOP/RUN.

### SMB30 y SMB130: Registros de control del modo Freeport

SMB30 y SMB130 controlan la comunicación Freeport en los puertos 0 y 1, respectivamente. SMB30 y SMB130 son marcas de lectura y escritura. Como muestra la tabla C-11, dichos bytes configuran la comunicación Freeport en los respectivos puertos y permiten seleccionar si se debe asistir el modo Freeport o el protocolo de sistema.

Tabla C-11 Bytes de marcas SMB30

Puerto 0	Puerto 1	Descripción								
Formato de SMB30	Formato de SMB130	<div style="display: flex; justify-content: space-between;"> <span>MSB 7</span> <span>LSB 0</span> </div> <div style="text-align: center; margin-top: 5px;"> <table border="1" style="border-collapse: collapse; width: 100px;"> <tr> <td style="padding: 2px 5px;">p</td> <td style="padding: 2px 5px;">p</td> <td style="padding: 2px 5px;">d</td> <td style="padding: 2px 5px;">b</td> <td style="padding: 2px 5px;">b</td> <td style="padding: 2px 5px;">b</td> <td style="padding: 2px 5px;">m</td> <td style="padding: 2px 5px;">m</td> </tr> </table> </div> <p style="text-align: right;">Byte de control del modo Freeport</p>	p	p	d	b	b	b	m	m
p	p	d	b	b	b	m	m			
SM30.6 y SM30.7	SM130.6 y SM130.7	<p>pp Selección de paridad</p> <p>00 = sin paridad</p> <p>01 = paridad par</p> <p>10 = sin paridad</p> <p>11 = paridad impar</p>								
SM30.5	SM130.5	<p>d Bits por carácter</p> <p>0 = 8 bits por carácter</p> <p>1 = 7 bits por carácter</p>								
SM30.2 a SM30.4	SM130.2 a SM130.4	<p>bbb Velocidad de transferencia</p> <p>000 = 38,400 bits/s</p> <p>001 = 19.200 bits/s</p> <p>010 = 9.600 bits/s</p> <p>011 = 4.800 bits/s</p> <p>100 = 2.400 bits/s</p> <p>101 = 1.200 bits/s</p> <p>110 = 600 bits/s</p> <p>111 = 300 bits/s</p>								
SM30.0 y SM30.1	SM130.0 y SM130.1	<p>mm Selección de protocolo</p> <p>00 = Protocolo de interface punto a punto (PPI/modo esclavo)</p> <p>01 = Protocolo Freeport</p> <p>10 = PPI/modo maestro</p> <p>11 = Reservado (estándar: PPI/modo esclavo)</p> <p>Nota: Si se selecciona el código mm = 10 (maestro PPI), la CPU pasará a ser una estación maestra en la red, permitiendo que se ejecuten las operaciones NETR y NETW. Los bits 2 a 7 se ignoran en el modo PPI.</p>								

### SMB31 y SMW32: Control de escritura en la memoria no volátil (EEPROM)

Un valor almacenado en la memoria de variables (memoria V) se puede guardar en la memoria no volátil (EEPROM) mediante el programa. A tal efecto, cargue en SMW32 la dirección que desee guardar. Cargue después SMB31 con el comando de guardar el valor. Una vez cargado dicho comando, no modifique el valor en la memoria V hasta que la CPU haya desactivado SM31.7, indicando que ha finalizado la memorización.

La CPU comprueba al final de cada ciclo si se debe guardar algún valor en la memoria no volátil. En caso afirmativo, el valor indicado se almacenará allí.



## SMB36 a SMB65: Bytes de programación de los contadores rápidos HSC0, HSC1 y HSC2

Como muestra la tabla C-14, los bytes de marcas SMB36 a SMB65 se utilizan para supervisar y controlar el funcionamiento de los contadores rápidos HSC0, HSC1 y HSC2.

Tabla C-14 Bytes de marcas SMB36 a SMB65

Byte de marcas	Descripción
SM36.0 a SM36.4	Reservados
SM36.5	Bit de estado del sentido de conteo actual de HSC0: 1 = contar adelante
SM36.6	El valor actual de HSC0 es igual al bit de estado del valor predeterminado: 1 = igual
SM36.7	El valor actual de HSC0 es mayor que el bit de estado del valor predeterminado: 1 = mayor que
SM37.0	Bit de control para nivel de actividad de la entrada de puesta a 0: 0 = actividad alta, 1 = actividad baja
SM37.1	Reservados
SM37.2	Velocidad de conteo de los contadores A/B: 0 = velocidad cuádruple; 1 = velocidad simple
SM37.3	Bit de control del sentido de conteo de HSC0: 1 = contar adelante
SM37.4	Actualizar el sentido de conteo de HSC0: 1 = sentido de actualización
SM37.5	Actualizar el valor predeterminado de HSC0: 1 = escribir nuevo valor predeterminado en HSC0
SM37.6	Actualizar el valor actual de HSC0: 1 = escribir nuevo valor actual en HSC0
SM37.7	Bit de habilitación de HSC0: 1 = habilitar
SMB38 SMB39 SMB40 SMB41	Nuevo valor actual de HSC0. SMB38 es el byte más significativo y SMB41 el byte menos significativo.
SMB42 SMB43 SMB44 SMB45	Nuevo valor predeterminado de HSC0 SMB42 es el byte más significativo y SMB45 el byte menos significativo.
SM46.0 a SM46.4	Reservados
SM46.5	Bit de estado del sentido de conteo actual de HSC1: 1 = contar adelante
SM46.6	El valor actual de HSC1 es igual al bit de estado del valor predeterminado: 1 = igual
SM46.7	El valor actual de HSC1 es mayor que el bit de estado del valor predeterminado: 1 = mayor que
SM47.0	Bit de control de nivel de actividad para puesta a 0 de HSC1: 0 = actividad alta, 1 = actividad baja
SM47.1	Bit de control de nivel de actividad para arranque de HSC1: 0 = actividad alta, 1 = actividad baja
SM47.2	Selección de velocidad de conteo de HSC1: 0 = cuádruple, 1 = simple
SM47.3	Bit de control del sentido de conteo de HSC1: 1 = contar adelante
SM47.4	Actualizar el sentido de conteo de HSC1: 1 = sentido de actualización

Tabla C-14 Bytes de marcas SMB36 a SMB65

Byte de marcas	Descripción
SM47.5	Actualizar el valor predeterminado de HSC1: 1 = escribir nuevo valor predeterminado en HSC1
SM47.6	Actualizar el valor actual de HSC1: 1 = escribir nuevo valor actual en HSC1
SM47.7	Bit de habilitación de HSC1: 1 = habilitar
SMB48 SMB49 SMB50 SMB51	Nuevo valor actual de HSC1 SMB48 es el byte más significativo y SMB51 el byte menos significativo.
SMB52 a SMB55	Nuevo valor predeterminado de HSC1 SMB52 es el byte más significativo y SMB55 el byte menos significativo.
SM56.0 a SM56.4	Reservados
SM56.5	Bit de estado del sentido de contaje actual de HSC2: 1 = contar adelante
SM56.6	El valor actual de HSC2 es igual al bit de estado del valor predeterminado: 1 = igual
SM56.7	El valor actual de HSC2 es mayor que el bit de estado del valor predeterminado: 1 = mayor que
SM57.0	Bit de control de nivel de actividad para puesta a 0 de HSC2: 0 = actividad alta, 1 = actividad baja
SM57.1	Bit de control de nivel de actividad para arranque de HSC2: 0 = actividad alta, 1 = actividad baja
SM57.2	Selección de velocidad de contaje de HSC2: 0 = cuádruple, 1 = simple
SM57.3	Bit de control del sentido de contaje de HSC2: 1 = contar adelante
SM57.4	Actualizar el sentido de contaje de HSC2: 1 = sentido de actualización
SM57.5	Actualizar el valor predeterminado de HSC2: 1 = escribir nuevo valor predeterminado en HSC2
SM57.6	Actualizar el valor actual de HSC2: 1 = escribir nuevo valor actual en HSC2
SM57.7	Bit de habilitación de HSC2: 1 = habilitar
SMB58 SMB59 SMB60 SMB61	Nuevo valor actual de HSC2 SMB58 es el byte más significativo y SMB61 es el byte menos significativo.
SMB62 SMB63 SMB64 SMB65	Nuevo valor predeterminado de HSC2 SMB62 es el byte más significativo y SMB65 es el byte menos significativo.

### SMB66 a SMB85: Funciones PTO/PWM

Como muestra la tabla C-15, SMB66 a SMB85 se utilizan para supervisar y controlar las funciones de modulación de salida de impulsos y de ancho de impulsos de las operaciones PTO/PWM. Consulte las informaciones sobre la operación de salida de impulsos en el apartado 9.5 del capítulo 9 para obtener una descripción detallada de dichas marcas.

Tabla C-15 Bytes de marcas SMB66 a SMB85

Byte de marcas	Descripción
SM66.0 a SM66.3	Reservados
SM66.4	Interrupción anormal del perfil PTO0; 0 = sin error, 1 = interrupción debida a un error de cálculo delta
SM66.5	Interrupción anormal del perfil PTO0; 0 = no causada por el usuario, 1 = causada por el usuario
SM66.6	Desbordamiento de pipeline PTO0 (el sistema lo pone a 0 al utilizarse perfiles externos; en caso contrario, el usuario lo deberá poner a 0); 0 = sin desbordamiento, 1 = desbordamiento
SM66.7	Bit de PTO0 en vacío: 0 = PTO en ejecución, 1 = PTO en vacío
SM67.0	Actualizar el tiempo de ciclo PTO0/PWM0: 1 = escribir nuevo valor del tiempo de ciclo
SM67.1	Actualizar el ancho de impulsos de PWM0: 1 = escribir nuevo ancho de impulsos
SM67.2	Actualizar el valor de conteo de impulsos de PTO0: 1 = escribir nuevo valor de conteo de impulsos
SM67.3	Base de tiempo PTO0/PWM0: 0 = 1 $\mu$ s/ciclo, 1 = 1 ms/ciclo
SM67.4	Actualización de PWM0: 0 = actualización asíncrona, 1 = actualización síncrona
SM67.5	Función PTO0: 0 = función monosegmento (tiempo de ciclo y conteo de impulsos almacenados en la memoria SM), 1 = función multisegmento (tabla de perfiles almacenada en la memoria V)
SM67.6	Elegir función de PTO0/PWM0 0 = PTO, 1 = PWM
SM67.7	Bit de habilitación de PTO0/PWM0: 1 = habilitar
SMB68 SMB69	Valor del tiempo de ciclo PTO0/PWM0 (2 a 65.535 unidades de la base de tiempo); SMB68 es el byte más significativo y SMB69 el byte menos significativo.
SMB70 SMB71	Valor del ancho de impulsos PWM0 (0 a 65.535 unidades de la base de tiempo); SMB70 es el byte más significativo y SMB71 el byte menos significativo.
SMB72 SMB73 SMB74 SMB75	Valor del conteo de impulsos PTO0 (1 a $2^{32}-1$ ); SMB72 es el byte más significativo y SMB75 el byte menos significativo.
SM76.0 a SM76.3	Reservados
SM76.4	Interrupción anormal del perfil PTO1; 0 = sin error, 1 = interrupción debida a un error de cálculo delta
SM76.5	Interrupción anormal de PTO1; 0 = no causada por el usuario, 1 = causada por el usuario

Tabla C-15 Bytes de marcas SMB66 a SMB85

Byte de marcas	Descripción
SM76.6	Desbordamiento de pipeline PTO1 (el sistema lo pone a 0 al utilizarse perfiles externos; en caso contrario, el usuario lo deberá poner a 0); 0 = sin desbordamiento, 1 = desbordamiento
SM76.7	Bit de PTO1 en vacío: 0 = PTO en ejecución, 1 = PTO en vacío
SM77.0	Actualizar el tiempo de ciclo PTO1/PWM1: 1 = escribir nuevo valor del tiempo de ciclo
SM77.1	Actualizar el ancho de impulsos de PWM1: 1 = escribir nuevo ancho de impulsos
SM77.2	Actualizar el valor de conteo de impulsos de PTO1: 1 = escribir nuevo valor de conteo de impulsos
SM77.3	Base de tiempo PTO1/PWM1: 0 = 1 $\mu$ s/ciclo, 1 = 1 ms/ciclo
SM77.4	Actualización de PWM1: 0 = actualización asíncrona, 1 = actualización síncrona
SM77.5	Función PTO1: 0 = función monosegmento (tiempo de ciclo y conteo de impulsos almacenados en la memoria SM), 1 = función multisegmento (tabla de perfiles almacenada en la memoria V)
SM77.6	Elegir función de PTO1/PWM1: 0 = PTO, 1 = PWM
SM77.7	Bit de habilitación de PTO1/PWM1: 1 = habilitar
SMB78 SMB79	Valor de tiempo de ciclo PTO1/PWM1 (2 a 65.535 unidades de la base de tiempo); SMB78 es el byte más significativo y SMB79 el byte menos significativo.
SMB80 SMB81	Valor del ancho de impulsos PWM1 (0 a 65.535 unidades de la base de tiempo); SMB80 es el byte más significativo y SMB81 el byte menos significativo.
SMB82 SMB83 SMB84 SMB85	Valor del conteo de impulsos PTO1 (1 a $2^{32}-1$ ); SMB82 es el byte más significativo y SMB85 el byte menos significativo.

### SMB86 a SMB94 y SMB186 a SMB194: Control de recepción de mensajes

Como muestra la tabla C-16, los bytes de marcas SMB86 a SMB94 y SMB186 a SMB194 se utilizan para controlar y leer el estado de la operación Recibir mensaje.

Tabla C-16 Marcas especiales SMB86 a SMB94 y SMB186 a SMB194

Puerto 0	Puerto 1	Descripción								
SMB86	SMB186	<div style="display: flex; align-items: center; justify-content: center;"> <div style="text-align: center; margin-right: 10px;"> <small>MSB</small> 7         </div> <div style="text-align: center; margin-right: 10px;"> <table border="1" style="border-collapse: collapse;"> <tr> <td style="padding: 2px 5px;">n</td> <td style="padding: 2px 5px;">r</td> <td style="padding: 2px 5px;">e</td> <td style="padding: 2px 5px;">0</td> <td style="padding: 2px 5px;">0</td> <td style="padding: 2px 5px;">t</td> <td style="padding: 2px 5px;">c</td> <td style="padding: 2px 5px;">p</td> </tr> </table> </div> <div style="text-align: center; margin-left: 10px;"> <small>LSB</small> 0         </div> </div> <p style="text-align: center;">Byte de estado de recepción de mensajes</p> <p>n: 1 = El usuario ha inhibido la recepción de mensajes</p> <p>r: 1 = Se finalizó la recepción de mensajes: error en parámetros de entrada o falta condición inicial o final</p> <p>e: 1 = Carácter final recibido</p> <p>t: 1 = Se finalizó la recepción de mensajes: ha transcurrido la temporización</p> <p>c: 1 = Se finalizó la recepción de mensajes: se ha excedido el número máximo de caracteres</p> <p>p: 1 = Se finalizó la recepción de mensajes debido a un error de paridad</p>	n	r	e	0	0	t	c	p
n	r	e	0	0	t	c	p			

Tabla C-16 Marcas especiales SMB86 a SMB94 y SMB186 a SMB194

Puerto 0	Puerto 1	Descripción								
SMB87	SMB187	<div style="display: flex; justify-content: space-between; align-items: center;"> <div style="text-align: center;"> <small>MSB</small> 7                 </div> <div style="text-align: center;"> <small>LSB</small> 0                 </div> </div> <div style="display: flex; justify-content: center; align-items: center; margin-top: 5px;"> <table border="1" style="border-collapse: collapse;"> <tr> <td style="padding: 2px 5px;">n</td> <td style="padding: 2px 5px;">x</td> <td style="padding: 2px 5px;">y</td> <td style="padding: 2px 5px;">z</td> <td style="padding: 2px 5px;">m</td> <td style="padding: 2px 5px;">t</td> <td style="padding: 2px 5px;">bk</td> <td style="padding: 2px 5px;">0</td> </tr> </table> <span style="margin-left: 10px;">Byte de control de recepción de mensajes</span> </div> <p>n: 0 = Inhibida la función de recibir mensajes. 1 = Habilitada la función de recibir mensajes. El bit para habilitar/inhibir la recepción de mensajes se comprueba cada vez que se ejecuta la operación RCV.</p> <p>x: 0 = Ignorar SMB88 o SMB188. 1 = Utilizar el valor de SMB88 o de SMB188 para detectar el comienzo del mensaje.</p> <p>y: 0 = Ignorar SMB89 o SMB189. 1 = Utilizar el valor de SMB89 o de SMB189 para detectar el fin del mensaje.</p> <p>z: 0 = Ignorar SMW90 o SMB190. 1 = Utilizar el valor de SMW90 para detectar una condición de inactividad.</p> <p>m: 0 = Utilizar el temporizador como temporizador entre caracteres. 1 = Utilizar el temporizador como temporizador de mensajes.</p> <p>t: 0 = Ignorar SMW92 o SMW192. 1 = Finalizar la recepción si se excede el período de tiempo indicado en SMW92 o SMW192.</p> <p>bk: 0 = Ignorar condiciones BREAK 1 = Utilizar condición BREAK como comienzo de la detección de mensajes</p> <p>Los bits del byte de control de interrupción de mensajes se utilizan para definir los criterios con los cuales se identifica el mensaje. Se definen los criterios tanto de comienzo como de final del mensaje. Para determinar el comienzo de un mensaje, uno de los dos juegos de criterios de comienzo de mensaje combinados lógicamente mediante Y deberán ser verdaderos y deberán ocurrir en secuencia (inactividad seguida de carácter inicial o condición BREAK seguida de de carácter inicial). Para determinar el fin de un mensaje, los criterios habilitados a tal efecto se combinan mediante O. A continuación se indican las ecuaciones de comienzo y de final:</p> <p style="padding-left: 40px;">Comienzo del mensaje = <math>il * sc + bk * sc</math></p> <p style="padding-left: 40px;">Final del mensaje = <math>ec + tnr + \text{contaje máximo de caracteres alcanzado}</math></p> <p>Programar los criterios de comienzo de mensaje para:</p> <ol style="list-style-type: none"> <li>1. Detección de inactividad: <span style="float: right;">il=1, sc=0, bk=0, SMW90&gt;0</span></li> <li>2. Detección de carácter inicial: <span style="float: right;">il=0, sc=1, bk=0, SMW90 no es relevante</span></li> <li>3. Detección BREAK: <span style="float: right;">il=0, sc=0, bk=1, SMW90 no es relevante</span></li> <li>4. Cualquier respuesta a una petición: <span style="float: right;">il=1, sc=0, bk=0, SMW90=0</span> (El temporizador de mensajes se puede utilizar para terminar la recepción si no hay respuesta).</li> <li>5. Carácter de break e inicial: <span style="float: right;">il=0, sc=1, bk=1, SMW90 no es relevante</span></li> <li>6. Inactividad y carácter inicial: <span style="float: right;">il=1, sc=1, bk=0, SMW90 &gt;0</span></li> <li>7. Inactividad y carácter inicial (no válido): <span style="float: right;">il=1, sc=1, bk=0, SMW90=0</span></li> </ol> <p>Nota: La recepción se finalizará automáticamente si ocurre un error de desbordamiento o de paridad (si se han habilitado).</p>	n	x	y	z	m	t	bk	0
n	x	y	z	m	t	bk	0			
SMB88	SMB188	Carácter de comienzo del mensaje.								

Tabla C-16 Marcas especiales SMB86 a SMB94 y SMB186 a SMB194

Puerto 0	Puerto 1	Descripción
SMB89	SMB189	Carácter de fin del mensaje.
SMB90 SMB91	SMB190 SMB191	Tiempo de línea de inactividad en milisegundos. El primer carácter recibido una vez transcurrido el tiempo de línea de inactividad es el comienzo del nuevo mensaje. SM90 (o SM190) es el byte más significativo y SM91 (ó SM191) es el byte menos significativo.
SMB92 SMB93	SMB192 SMB193	Vigilancia del temporizador entre caracteres/de mensajes en milisegundos. Si se excede el tiempo, se finaliza la recepción de mensajes. SM92 (o SM192) es el byte más significativo y SM93 (ó SM193) es el byte menos significativo.
SMB94	SMB194	Número máximo de caracteres a recibir (1 a 255 bytes). Nota: Este margen debe ajustarse al tamaño máximo esperado para el búfer, incluso si no se utiliza la finalización de mensajes por el conteo de caracteres.

### SMB98 y SMB99

Como muestra la tabla C-17, SMB98 y SMB99 indican el número de errores en el bus de ampliación.

Tabla C-17 Bytes de marcas SMB98 y SMB99

Byte de marcas	Descripción
SMB98 SMB99	Esta dirección se incrementa cada vez que se detecta un error de paridad en el bus de ampliación. Se borra durante el encendido y cuando el usuario la pone a 0. SMB98 es el byte más significativo.

## SMB130 a SMB165: Bytes de programación de los contadores rápidos HSC3, HSC4 y HSC5

Como muestra la tabla C-18, los bytes de marcas SMB130 a SMB165 se utilizan para supervisar y controlar el funcionamiento de los contadores rápidos HSC3, HSC4 y HSC5.

Tabla C-18 Bytes de marcas SMB130 a SMB165

Byte de marcas	Descripción
SMB131 a SMB135	Reservados
SM136.0 a SM136.4	Reservados
SM136.5	Bit de estado del sentido de contaje actual de HSC3: 1 = contar adelante
SM136.6	El valor actual de HSC3 es igual al bit de estado del valor predeterminado: 1 = igual
SM136.7	El valor actual de HSC3 es mayor que el bit de estado del valor predeterminado: 1 = mayor que
SM137.0 a SM137.2	Reservados
SM137.3	Bit de control del sentido de contaje de HSC3: 1 = contar adelante
SM137.4	Actualizar sentido de contaje de HSC3: 1 = sentido de actualización
SM137.5	Actualizar valor predeterminado de HSC3: 1 = escribir nuevo valor predeterminado en HSC3
SM137.6	Bit de habilitación de HSC3: 1 = habilitar
SM138 a SM141	Nuevo valor actual de HSC3. SMB138 es el bit más significativo y SMB141 el bit menos significativo.
SM142 a SM145	Nuevo valor predeterminado de HSC3 SMB142 es el bit más significativo y SMB145 el bit menos significativo.
SM146.0 a SM146.4	Reservados
SM146.5	Bit de estado del sentido de contaje actual de HSC4: 1 = contar adelante
SM146.7	El valor actual de HSC4 es mayor que el bit de estado del valor predeterminado: 1 = mayor que
SM147.0	Bit de control para nivel de actividad de la entrada de puesta a 0: 0 = actividad alta, 1 = actividad baja
SM147.1	Reservados
SM147.2	Velocidad de contaje de los contadores A/B: 0 = velocidad cuádruple; 1 = velocidad simple
SM147.3	Bit de control del sentido de contaje de HSC4: 1 = contar adelante
SM147.4	Actualizar sentido de contaje de HSC4: 1 = sentido de actualización
SM147.5	Actualizar valor predeterminado de HSC4: 1 = escribir nuevo valor predeterminado en HSC4
SM147.6	Actualizar valor actual de HSC4: 1 = escribir nuevo valor actual en HSC4
SM147.7	Bit de habilitación de HSC4: 1 = habilitar
SMB148 a SMB151	Nuevo valor actual de HSC4. SMB148 es el bit más significativo y SMB151 el bit menos significativo.
SMB152 a SMB155	Nuevo valor predeterminado de HSC4. SMB152 es el bit más significativo y SMB155 el bit menos significativo.

Tabla C-18 Bytes de marcas SMB130 a SMB165

Byte de marcas	Descripción
SM156.0 a SM156.4	Reservados
SM156.5	Bit de estado del sentido de contaje actual de HSC5: 1 = contar adelante
SM156.6	El valor actual de HSC5 es igual al bit de estado del valor predeterminado: 1 = igual
SM156.7	El valor actual de HSC5 es mayor que el bit de estado del valor predeterminado: 1 = mayor que
SM157.0 a SM157.2	Reservados
SM157.3	Bit de control del sentido de contaje de HSC5: 1 = contar adelante
SM157.4	Actualizar sentido de contaje de HSC5: 1 = sentido de actualización
SM157.5	Actualizar valor predeterminado de HSC5: 1 = escribir nuevo valor predeterminado en HSC5
SM157.6	Actualizar valor actual de HSC5: 1 = escribir nuevo valor actual en HSC5
SM157.7	Bit de habilitación de HSC5: 1 = habilitar
SMB158 a SMB161	Nuevo valor actual de HSC5. SMB158 es el bit más significativo y SMB161 el bit menos significativo.
SMB162 a SMB165	Nuevo valor predeterminado de HSC5. SMB162 es el bit más significativo y SMB165 el bit menos significativo.

### SMB166 a SMB194: Tabla de definición de perfiles PTO, PT1

Como muestra la tabla C-19, los bytes de marcas SMB166 a SMB194 se utilizan para mostrar el número de pasos del perfil activo y la dirección de la tabla de perfiles en la memoria V.

Tabla C-19 Bytes de marcas SMB166 a SMB194

Byte de marcas	Descripción
SMB166	Número actual del paso del perfil activo de PTO0.
SMB167	Reservados
SMB168 SMB169	Dirección en la memoria V de la tabla del perfil PTO0 indicada como offset desde V0. SM168 es el byte más significativo del offset de dirección.
SMB170 a SMB175	Reservados
SMB176	Número actual del paso del perfil activo de PTO1.
SMB177	Reservados
SMB178 a SMB179	Dirección en la memoria V de la tabla del perfil PTO1 indicada como offset desde V0. SM178 es el byte más significativo del offset de dirección.
SMB180 a SMB194:	Reservados

# D

## Eliminar errores

Tabla D-1 Eliminar errores

Problema	Causas posibles	Solución
Las salidas han dejado de funcionar.	<ul style="list-style-type: none"> <li>• El dispositivo controlado ha causado una sobretensión que ha deteriorado la salida.</li> <li>• Error en el programa de usuario.</li> <li>• Cableado suelto o incorrecto.</li> <li>• Carga excesiva.</li> <li>• Salidas forzadas.</li> </ul>	<ul style="list-style-type: none"> <li>• Al conectar la CPU a una carga inductiva (p.ej. un motor o un relé) es preciso utilizar un circuito de supresión adecuado (consulte el apartado 2.4).</li> <li>• Corregir el programa de usuario.</li> <li>• Comprobar y corregir el cableado.</li> <li>• Comprobar la carga en la E/S.</li> <li>• Comprobar las E/S forzadas.</li> </ul>
El diodo "SF" (System Fault) de la CPU se enciende.	<p>La lista siguiente describe las causas más frecuentes:</p> <ul style="list-style-type: none"> <li>• Error en el programa de usuario.                             <ul style="list-style-type: none"> <li>– 0003 Error de tiempo de vigilancia (watchdog).</li> <li>– 0011 Direccionamiento indirecto.</li> <li>– 0012 Valor en coma flotante no válido.</li> </ul> </li> <li>• Interferencia eléctrica.                             <ul style="list-style-type: none"> <li>– 0001 a 0009</li> </ul> </li> <li>• Componente deteriorado.                             <ul style="list-style-type: none"> <li>– 0001 a 0010</li> </ul> </li> </ul>	<p>Lea el código de error fatal y consulte el apartado B.1:</p> <ul style="list-style-type: none"> <li>• En caso de un error de programación, consulte la descripción de las operaciones FOR, NEXT, JMP, LBL, así como de las operaciones de comparación.</li> <li>• En caso de una interferencia eléctrica:                             <ul style="list-style-type: none"> <li>– Consulte las reglas de cableado en el apartado 2.3. Es muy importante que el armario eléctrico esté conectado correctamente a tierra y que el cableado de alta y baja tensión no se conduzcan en paralelo.</li> <li>– Conecte a tierra el terminal M de la alimentación de sensores de DC 24 V.</li> </ul> </li> </ul>
Fuente de alimentación averiada.	Sobretensión en los cables conectados al equipo.	<p>Conecte un dispositivo para medir la magnitud y la duración de las puntas de sobretensión. Conforme a dichas informaciones, incorpore un dispositivo apropiado de supresión de sobretensiones.</p> <p>Para obtener informaciones más detalladas acerca de la instalación del cableado de campo, consulte el apartado 2.3.</p>
Interferencias eléctricas	<ul style="list-style-type: none"> <li>• Puesta a tierra incorrecta.</li> <li>• Conducción del cableado en el armario eléctrico.</li> <li>• Filtros de entrada configurados para una velocidad demasiado rápida.</li> </ul>	<p>Consulte las reglas de cableado en el apartado 2.3. Es muy importante que el armario eléctrico esté conectado correctamente a tierra y que el cableado de alta y baja tensión no se conduzcan en paralelo.</p> <p>Conecte a tierra el terminal M de la alimentación de sensores de DC 24 V.</p> <p>Incrementar en el bloque de datos el retardo del filtro de entrada (v. apt. 5.2.)</p>

Tabla D-1 Eliminar errores

Problema	Causas posibles	Solución
Red de comunicación deteriorada al conectar un dispositivo externo. (Están deteriorados el puerto del PC, el puerto de la CPU o el cable PC/PPI).	El cable de comunicación puede convertirse en una ruta de corrientes indeseadas si los dispositivos que no tengan separación galvánica (tales como las CPUs, los PCs u otros dispositivos) y que estén conectados al cable, no comparten un mismo hilo de referencia en el circuito. Las corrientes indeseadas pueden causar errores de comunicación o deteriorar los circuitos.	<ul style="list-style-type: none"> <li>• Consulte el apartado 2.3 donde se indican las reglas de cableado y el capítulo 7 para obtener más información acerca de la comunicación en redes.</li> <li>• Sustituya el cable PC/PPI.</li> <li>• Utilice un repetidor de RS-485 a RS-485 con separación galvánica al conectar máquinas que no tengan una referencia eléctrica común.</li> </ul>
Problemas de comunicación en STEP 7-Micro/WIN 32		Para obtener informaciones más detalladas acerca de la comunicación en redes, consulte el capítulo 7 .
Tratamiento de errores		Para obtener informaciones más detalladas acerca de los códigos de error, consulte el Anexo B.



# E

## Números de referencia

CPUs	Nº de referencia
CPU 221 DC/DC/DC 6 entradas/4 salidas	6ES7 211-0AA20-0XB0
CPU 221 AC/DC/relé 6 entradas/4 salidas	6ES7 211-0BA20-0XB0
CPU 222 DC/DC/DC 8 entradas/6 salidas	6ES7 212-1AB20-0XB0
CPU 222 AC/DC/Relay 8 entradas/6 salidas	6ES7 212-1BB20-0XB0
CPU 224 DC/DC/DC 14 entradas/10 salidas	6ES7 214-1AD20-0XB0
CPU 224 AC/DC/relé 14 entradas/10 salidas	6ES7 214-1BD20-0XB0

Módulos de ampliación	Nº de referencia
EM221 8 entradas digitales DC 24 V	6ES7 221-1BF20-0XA0
EM222 8 salidas digitales DC 24 V	6ES7 222-1BF20-0XA0
EM222 8 salidas de relé	6ES7 222-1HF20-0XA0
EM223 8 entradas digitales/8 salidas digitales DC 24 V	6ES7 223-1BH20-0XA0
EM223 8 entradas digitales/8 salidas de relé DC 24 V	6ES7 223-1PH20-0XA0

Cartuchos y cables	Nº de referencia
Cartucho de memoria MC 291, CPU 22x	6ES7 291-8GE20-0XA0
Cartucho, reloj/calendario con pila CC 292, CPU 22x	6ES7 297-1AA20-0XA0
Cartucho de pila BC 293, CPU 22x	6ES7 291-8BA20-0XA0
Cable para módulos de ampliación, 0,8 metros, CPU 22x/EM	6ES7 290-6AA20-0XA0
Cable, PC/PPI, aislado, 5 interruptores DIP	6ES7 901-3BF20-0XA0

Software de programación	Nº de referencia
STEP 7-Micro/WIN 32 (V3.0) licencia única (disquete)	6ES7 810-2BA00-0YX0
STEP 7-Micro/WIN 32 (V3.0) licencia de actualización (disquete)	6ES7 810-2BA00-0YX3
STEP 7-Micro/WIN 32 (V3.0) licencia única (CD-ROM)	6ES7 810-2BC00-0YX0
STEP 7-Micro/WIN 32 (V3.0) licencia de actualización (CD-ROM)	6ES7 810-2BC00-0YX3

Tarjetas de comunicación	Nº de referencia
Tarjeta MPI: Short AT ISA	6ES7 793-2AA01-0AA0
CP 5411: Short AT ISA	6GK1 41-1AA00
CP 5511: PCMCIA, tipo II	6GK1 551-1AA00
CP 5611: Tarjeta PCI (versión 3.0 o superior)	6GK1 561-1AA00

Manuales	Nº de referencia
Manual del usuario del visualizador de textos TD 200	6ES7 272-0AA00-8BA0
S7-200 Point-to-Point Interface Communication Manual (inglés/alemán)	6ES7 298-8GA00-8XH0
Sistema de automatización S7-200, Manual del sistema (alemán)	6ES7 298-8FA20-8AH0
Sistema de automatización S7-200, Manual del sistema (inglés)	6ES7 298-8FA20-8BH0
Sistema de automatización S7-200, Manual del sistema (francés)	6ES7 298-8FA20-8CH0
Sistema de automatización S7-200, Manual del sistema (español)	6ES7 298-8FA20-8DH0
Sistema de automatización S7-200, Manual del sistema (italiano)	6ES7 298-8FA20-8EH0

Cables, conectores de bus y repetidores	Nº de referencia
Cable MPI	6ES7 901-0BF00-0AA0
Cable para redes PROFIBUS	6XVI 830-0AH10
Conector de bus con conector de interface de programación, salida vertical del cable	6ES7 972-0BB11-0XA0
Conector de bus (sin conector de interface de programación), salida vertical del cable	6ES7 972-0BA11-0XA0
Bloque de conectores CPU 22x/EM, 7 terminales, extraíble	6ES7 292-1AD20-0AA0
Bloque de conectores CPU 22x/EM, 12 terminales, extraíble	6ES7 292-1AE20-0AA0
Bloque de conectores CPU 22x/EM 18 terminales, extraíble	6ES7 292-1AG20-0AA0
Conector de bus RS-485 con salida de cable a 35°	6ES7 972-0BA40-0XA0
Repetidor RS-485 IP 20, aislado	6ES7 972-0AA00-0XA0

Visualizadores de textos y paneles de operador	Nº de referencia
Visualizador de textos TD 200	6ES7 272-0AA00-0YA0
Panel de operador OP3	6AV3 503-1DB10
Panel de operador OP7	6AV3 607-1JC20-0AX1
Panel de operador OP17	6AV3 617-1JC20-0AX1

Varios	Nº de referencia
Topes para perfiles soporte	6ES5 728-8MA11
Bloque de 12 bornes de conexión para cableado de campo (CPU 221, CPU 222) paquete de 10	6ES7 290-2AA00-0XA0
Kit de tapas de repuesto (contiene 4 de las siguientes tapas): tapa de bloque de 12 terminales para la CPU 221/222 EM22x, tapa de bloque de 18 terminales para la CPU 224, tapa de bloque de 7 terminales para el EM 22x, tapa de acceso a la CPU, tapa de acceso al módulo de ampliación	6ES7 291-3AX20-0XA0

## Tiempos de ejecución de las operaciones AWL

# F

### Impacto de la circulación de la corriente sobre el tiempo de ejecución

El cálculo del tiempo de ejecución básico de una operación AWL (v. tabla F-4) indica el tiempo necesario para ejecutar la lógica o la función de la operación cuando se aplica corriente (estando activado (puesto a "1") ese valor). En algunas operaciones, la ejecución de la función depende de si se aplica corriente o no. La CPU sólo ejecuta la función si se aplica corriente a la operación, estando entonces activado (puesto a "1") el primer valor de la pila. Si no se aplica corriente a la operación (estando desactivado (puesto a "0") el primer valor de la pila), se deberá utilizar un tiempo de ejecución sin circulación de corriente para calcular dicho tiempo. La tabla F-1 muestra el tiempo de ejecución de una operación AWL sin circulación de corriente (estando desactivado (puesto a "0") el primer valor de la pila) para cada una de las CPUs S7-200.

Tabla F-1 Tiempo de ejecución de operaciones sin circulación de corriente

Operación sin circulación de corriente	CPU S7-200
Todas las operaciones AWL	3 $\mu$ s

### Impacto del direccionamiento indirecto sobre el tiempo de ejecución

El cálculo del tiempo de ejecución básico de una operación AWL (v. tabla F-4) indica el tiempo necesario para ejecutar la operación si los operandos o las constantes se direccionan directamente. Si en el programa se usan direcciones indirectas, el tiempo de ejecución de cada operando direccionado indirectamente se incrementa como muestra la tabla F-2.

Tabla F-2 Tiempo adicional para el direccionamiento indirecto

Operación de direccionamiento indirecto	CPU S7-200
Cada operando direccionado indirectamente	22 $\mu$ s

## Tiempos de ejecución

El acceso a ciertas áreas de memoria, tales como AI (entradas analógicas), AQ (salidas analógicas), L (memoria local) y AC (acumuladores), prolonga también el tiempo de ejecución. La tabla F-3 muestra un factor que se debe añadir al tiempo de ejecución básico de cada operando que acceda a dichas áreas.

Tabla F-3 Factor a añadir al tiempo de ejecución para acceder a algunas áreas de memoria

Área de memoria	CPU S7-200
Entradas analógicas (AI)	149 $\mu$ s
Salidas analógicas (AQ)	73 $\mu$ s
Memoria local (L)	5,4 $\mu$ s
Acumuladores (AC)	4,4 $\mu$ s

## Tiempos de ejecución básicos de las operaciones AWL

La tabla F-4 indica los tiempos de ejecución básicos de las operaciones AWL para cada una de las CPUs S7-200.

Tabla F-4 Tiempos de ejecución de las operaciones AWL (en  $\mu$ s)

Operación	Descripción	S7-200 CPU (en $\mu$ s)
=	Tiempo de ejecución básico: I L SM, T, C, V, S, Q, M	0,37 19,2 1,8
+D	Tiempo de ejecución básico	55
-D	Tiempo de ejecución básico	55
*D	Tiempo de ejecución básico	92
/D	Tiempo de ejecución básico	376
+I	Tiempo de ejecución básico	46
-I	Tiempo de ejecución básico	47
*I	Tiempo de ejecución básico	71
/I	Tiempo de ejecución básico	115
=I	Tiempo de ejecución básico: salida integrada salida en un módulo de ampliación	29 39
+R	Tiempo de ejecución básico Tiempo de ejecución máximo	110 163
-R	Tiempo de ejecución básico Tiempo de ejecución máximo	113 166
*R	Tiempo de ejecución básico Tiempo de ejecución máximo	100 130
/R	Tiempo de ejecución básico Tiempo de ejecución máximo	300 360

Tabla F-4 Tiempos de ejecución de las operaciones AWL (en  $\mu$ s)

Operación	Descripción	S7-200 CPU (en $\mu$ s)
A	Tiempo de ejecución básico: I L SM, T, C, V, S, Q, M	0,37 10,8 1,1
AB <=, =, >=, >, <, <>	Tiempo de ejecución básico	35
AD <=, =, >=, >, <, <>	Tiempo de ejecución básico	53
AI	Tiempo de ejecución básico: Entrada integrada Entrada en un módulo de ampliación	27 35
ALD	Tiempo de ejecución básico	0,37
AN	Tiempo de ejecución básico: I L SM, T, C, V, S, Q, M	0,37 10,8 1,1
ANDB	Tiempo de ejecución básico	37
ANDD	Tiempo de ejecución básico	55
ANDW	Tiempo de ejecución básico	48
ANI	Tiempo de ejecución básico: Entrada integrada Entrada en un módulo de ampliación	27 35
AR <=, =, >=, >, <, <>	Tiempo de ejecución básico	54
ATCH	Tiempo de ejecución básico	20
ATH	Total = tiempo básico + (longitud) * (ML) Tiempo de ejecución básico (longitud constante) Tiempo de ejecución básico (longitud variable) Multiplicador de longitud (ML)	177 186 23
ATT	Tiempo de ejecución básico	125
AW <=, =, >=, >, <, <>	Tiempo de ejecución básico	45
BCDI	Tiempo de ejecución básico	66
BMB	Total = tiempo básico + (longitud) * (ML) Tiempo de ejecución básico (longitud constante) Tiempo de ejecución básico (longitud variable) Multiplicador de longitud (ML)	172 181 11
BMD	Total = tiempo básico + (longitud) * (ML) Tiempo de ejecución básico (longitud constante) Tiempo de ejecución básico (longitud variable) Multiplicador de longitud (ML)	173 183 20
BMW	Total = tiempo básico + (longitud) * (ML) Tiempo de ejecución básico (longitud constante) Tiempo de ejecución básico (longitud variable) Multiplicador de longitud (ML)	172 181 16

Tabla F-4 Tiempos de ejecución de las operaciones AWL (en  $\mu$ s)

Operación	Descripción	S7-200 CPU (en $\mu$ s)
CALL	Sin parámetros: Tiempo de ejecución	15
	Con parámetros: Tiempo total de ejecución = Tiempo básico + $\Sigma$ (tiempo de gestión de los operandos de entrada)	
	Tiempo de ejecución básico	32
	Tiempo de gestión de los operandos de entrada (bit)	23
	Tiempo de gestión de los operandos de entrada (byte)	21
	Tiempo de gestión de los operandos de entrada (palabra)	24
	Tiempo de gestión de los operandos de entrada (palabra doble)	27
CRET	Tiempo total de ejecución = Tiempo básico + $\Sigma$ (tiempo de gestión de los operandos de salida)	
	Tiempo de ejecución básico	13
	Tiempo de gestión de los operandos de salida (bit)	21
	Tiempo de gestión de los operandos de salida (byte)	14
	Tiempo de gestión de los operandos de salida (palabra)	18
	Tiempo de gestión de los operandos de salida (palabra doble)	20
CRETI	Tiempo de ejecución básico	23
CTD	Tiempo de ejecución básico al producirse un flanco en la entrada de contaje	48
	Tiempo de ejecución básico en otro caso	36
CTU	Tiempo de ejecución básico al producirse un flanco en la entrada de contaje	53
	Tiempo de ejecución básico en otro caso	35
CTUD	Tiempo de ejecución básico al producirse un flanco en la entrada de contaje	64
	Tiempo de ejecución básico en otro caso	45
DECB	Tiempo de ejecución básico	30
DECD	Tiempo de ejecución básico	42
DECO	Tiempo de ejecución básico	36
DECW	Tiempo de ejecución básico	37
DISI	Tiempo de ejecución básico	18
DIV	Tiempo de ejecución básico	119
DTCH	Tiempo de ejecución básico	18
DTR	Tiempo de ejecución básico	60
	Tiempo de ejecución máximo	70
ED	Tiempo de ejecución básico	15
ENCO	Tiempo de ejecución mínimo	39
	Tiempo de ejecución máximo	43
END	Tiempo de ejecución básico	0,9
ENI	Tiempo de ejecución básico	53
EU	Tiempo de ejecución básico	15
FIFO	Total = tiempo básico + (ML) < * (longitud)	
	Tiempo de ejecución básico	109
	Multiplicador de longitud (ML)	14

Tabla F-4 Tiempos de ejecución de las operaciones AWL (en  $\mu$ s)

Operación	Descripción	S7-200 CPU (en $\mu$ s)
FILL	Total = tiempo básico + (LM) * $\times$ (longitud) Tiempo de ejecución básico (longitud constante) Tiempo de ejecución básico (longitud variable) Multiplicador de longitud (ML)	156 165 7
FND <, =, >, <>	Total = tiempo básico + (ML) * $\times$ (longitud) Tiempo de ejecución básico Multiplicador de longitud (ML)	224 12
FOR	Total = tiempo básico + (ML) * $\times$ (número de repeticiones) Tiempo de ejecución básico Multiplicador de lazos (ML)	73 72
HDEF	Tiempo de ejecución básico	35
HSC	Tiempo de ejecución básico	37
HTA	Total = tiempo básico + (LM) * $\times$ (longitud) Tiempo de ejecución básico (longitud constante) Tiempo de ejecución básico (longitud variable) Multiplicador de longitud (ML)	175 184 11
IBCD	Tiempo de ejecución básico	114
INCB	Tiempo de ejecución básico	29
INCD	Tiempo de ejecución básico	42
INCW	Tiempo de ejecución básico	37
INT	Tiempo de ejecución típico con 1 interrupción	47
INVB	Tiempo de ejecución básico	31
INVD	Tiempo de ejecución básico	42
IN VW	Tiempo de ejecución básico	38
JMP	Tiempo de ejecución básico	0,9
LBL	Tiempo de ejecución básico	0,37
LD	Tiempo de ejecución básico: I L SM, T, C, V, S, Q, M SM0.0	0,37 10,9 1,1 0,37
LDB <=, =, >=, >, <, <>	Tiempo de ejecución básico	35
LDD <=, =, >=, >, <, <>	Tiempo de ejecución básico	52
LDI	Tiempo de ejecución básico: Entrada integrada Entrada en un módulo de ampliación	26 34
LDN	Tiempo de ejecución básico: I L SM, T, C, V, S, Q, M	0,37 10,9 1,1
LDNI	Tiempo de ejecución básico: Entrada integrada Entrada en un módulo de ampliación	26 34
LDR <=, =, >=, >, <, <>	Tiempo de ejecución básico	55
LDS	Tiempo de ejecución básico	0,37

Tabla F-4 Tiempos de ejecución de las operaciones AWL (en  $\mu$ s)

Operación	Descripción	S7-200 CPU (en $\mu$ s)
LDW <=, =, >=, >, <, <>	Tiempo de ejecución básico	42
LIFO	Tiempo de ejecución básico	121
LPP	Tiempo de ejecución básico	0,37
LPS	Tiempo de ejecución básico	0,37
LRD	Tiempo de ejecución básico	0,37
LSCR	Tiempo de ejecución básico	12
MEND	Tiempo de ejecución básico	0,5
MOVB	Tiempo de ejecución básico	29
MOVD	Tiempo de ejecución básico	38
MOVR	Tiempo de ejecución básico	38
MOVW	Tiempo de ejecución básico	34
MUL	Tiempo de ejecución básico	70
NEXT	Tiempo de ejecución básico	0
NETR	Tiempo de ejecución básico	286
NETW	Total = tiempo básico + (ML) * (longitud) Tiempo de ejecución básico Multiplicador de longitud (ML)	274 8
NOP	Tiempo de ejecución básico	0,37
NOT	Tiempo de ejecución básico	0,37
O	Tiempo de ejecución básico: I L SM, T, C, V, S, Q, M	0,37 10,8 1,1
OB <=, =, >=, >, <, <>	Tiempo de ejecución básico	35
OD <=, =, >=, >, <, <>	Tiempo de ejecución básico	53
OI	Tiempo de ejecución básico: Entrada integrada Entrada en un módulo de ampliación	27 35
OLD	Tiempo de ejecución básico	0,37
ON	Tiempo de ejecución básico: I L SM, T, C, V, S, Q, M	0,37 10,8 1,1
ONI	Tiempo de ejecución básico: Entrada integrada Entrada en un módulo de ampliación	27 35
OR<=, =, >=, >, <, <>	Tiempo de ejecución básico	55
ORB	Tiempo de ejecución básico	37
ORD	Tiempo de ejecución básico	55
ORW	Tiempo de ejecución básico	48
OW <=, =, >=, >, <, <>	Tiempo de ejecución básico	45

Tabla F-4 Tiempos de ejecución de las operaciones AWL (en  $\mu$ s)

Operación	Descripción	S7-200 CPU (en $\mu$ s)
PID	Tiempo de ejecución básico	750
	Sumador para recalcar ( $K_C * T_s/T_i$ ) y ( $K_C * T_d/T_s$ ) antes del cálculo PID. Se recalcula si $K_C$ , $T_s$ , $T_i$ , o $T_s$ se han modificado desde la anterior ejecución de esta operación o si se ha cambiado a modo automático.	1000
PLS	Tiempo de ejecución básico: PWM	57
	PTO monosegmento	67
	PTO multisegmento	92
R	Para longitud=1 y especificado como constante (p.ej. R V0.2,1)	
	Tiempo de ejecución del operando = C	17
	Tiempo de ejecución del operando = T	24
	Tiempo de ejecución de todos los demás operandos	5
	En otro caso, Total=tiempo de ejecución básico +(LM)*(longitud)	
	Tiempo de ejecución básico del operando = C, T	19
	Tiempo de ejecución básico de todos los demás operandos	28
	Multiplicador de longitud (ML) del operando = C	8,6
Multiplicador de longitud (ML) del operando = T	16,5	
Multiplicador de longitud (ML) de todos los demás operandos	0,9	
Si la longitud se almacena en una variable en lugar de especificarse en calidad de constante, incrementar el tiempo de ejecución básico sumando:	29	
RCV	Tiempo de ejecución básico	104
RET	Tiempo de ejecución total = tiempo básico + $\Sigma$ (tiempo de gestión de los operandos de salida)	
	Tiempo de ejecución básico	13
	Tiempo de gestión de los operandos de salida (bit)	21
	Tiempo de gestión de los operandos de salida (byte)	14
	Tiempo de gestión de los operandos de salida (palabra)	18
Tiempo de gestión de los operandos de salida (palabra doble)	20	
RETI	Tiempo de ejecución básico	23
RI	Total = tiempo básico + (ML) * (longitud)	
	Tiempo de ejecución básico	18
	Multiplicador de longitud (salida integrada)	22
	Multiplicador de longitud (salida en módulo de ampliación)	32
Si la longitud se almacena en una variable, en vez de ser una constante, incrementar el tiempo de ejecución básico sumando:	30	
RLB	Total = tiempo básico + (ML) * (longitud)	
	Tiempo de ejecución básico	42
	Multiplicador de longitud (ML)	0,6
RLD	Total = tiempo básico + (ML) * (longitud)	
	Tiempo de ejecución básico	52
	Multiplicador de longitud (ML)	2,5
RLW	Total = tiempo básico + (ML) * (longitud)	
	Tiempo de ejecución básico	49
	Multiplicador de longitud (ML)	1,7

Tabla F-4 Tiempos de ejecución de las operaciones AWL (en  $\mu$ s)

Operación	Descripción	S7-200 CPU (en $\mu$ s)
RRB	Total = tiempo básico + (ML) * (longitud) Tiempo de ejecución básico Multiplicador de longitud (ML)	42 0,6
RRD	Total = tiempo básico + (ML) * (longitud) Tiempo de ejecución básico Multiplicador de longitud (ML)	52 2,5
RRW	Total = tiempo básico + (ML) * (longitud) Tiempo de ejecución básico Multiplicador de longitud (ML)	49 1,7
S	Para longitud = 1 y especificada como constante (p.ej. S V0.2, 1) Tiempo de ejecución En otro caso, Total=tiempo de ejecución básico+(ML)*(longitud) Tiempo de ejecución básico de todos los demás operandos Multiplicador de longitud (ML) para todos los demás operandos Si la longitud se almacena en una variable, en vez de ser una constante, incrementar el tiempo de ejecución básico sumando:	5  27 0,9 29
SBR	Tiempo de ejecución básico	0
SCRE	Tiempo de ejecución básico	0,37
SCRT	Tiempo de ejecución básico	17
SEG	Tiempo de ejecución básico	30
SHRB	Total = tiempo básico + (ML) * (longitud) Tiempo de ejecución básico Multiplicador de longitud (ML)	140 1,6
SI	Total = tiempo básico + (ML) * (longitud) Tiempo de ejecución básico Multiplicador de longitud (ML) (salida integrada) Multiplicador de longitud (ML) (salida en módulo de ampliación) Si la longitud se almacena en una variable, en vez de ser una constante, incrementar el tiempo de ejecución básico sumando:	18 22 32 30
SLB	Total = tiempo básico + (ML) * (longitud) Tiempo de ejecución básico Multiplicador de longitud (ML)	43 0,7
SLD	Total = tiempo básico + (ML) * (longitud) Tiempo de ejecución básico Multiplicador de longitud (ML)	53 2,6
SLW	Total = tiempo básico + (ML) * (longitud) Tiempo de ejecución básico Multiplicador de longitud (ML)	51 1,3
SQRT	Tiempo de ejecución básico Tiempo de ejecución máximo	725 830
SRB	Total = tiempo básico + (ML) * (longitud) Tiempo de ejecución básico Multiplicador de longitud (ML)	43 0,7

Tabla F-4 Tiempos de ejecución de las operaciones AWL (en  $\mu$ s)

Operación	Descripción	S7-200 CPU (en $\mu$ s)
SRD	Total = tiempo básico + (ML) * (longitud) Tiempo de ejecución básico Multiplicador de longitud (ML)	53 2,6
SRW	Total = tiempo básico + (ML) * (longitud) Tiempo de ejecución básico Multiplicador de longitud (ML)	51 1,3
STOP	Tiempo de ejecución básico	16
SWAP	Tiempo de ejecución básico	32
TODR	Tiempo de ejecución básico	2400
TODW	Tiempo de ejecución básico	1600
TOF	Tiempo de ejecución básico	64
TON	Tiempo de ejecución básico	64
TONR	Tiempo de ejecución básico	56
TRUNC	Tiempo de ejecución básico Tiempo de ejecución máximo	103 178
WDR	Tiempo de ejecución básico	16
XMT	Tiempo de ejecución básico	113
XORB	Tiempo de ejecución básico	37
XORD	Tiempo de ejecución básico	55
XORW	Tiempo de ejecución básico	48

# G

## Breviario del S7-200

Este anexo contiene informaciones sobre los siguientes temas:

- Marcas especiales
- Descripción de los eventos de interrupción
- Resumen de las áreas de memoria y funciones de las CPUs S7-200
- Contadores rápidos HSC0, HSC1, HSC2, HSC3, HSC4, HSC5
- Operaciones S7-200

Tabla G-1 Marcas especiales

Marcas especiales			
SM0.0	Siempre ON	SM1.0	Resultado de la operación = 0
SM0.1	Primer ciclo	SM1.1	Desbordamiento o valor no válido
SM0.2	Datos remanentes perdidos	SM1.2	Resultado negativo
SM0.3	Alimentación	SM1.3	División por 0
SM0.4	30 s OFF / 30 s ON	SM1.4	Desbordamiento tabla
SM0.5	0,5 s OFF / 0,5 s ON	SM1.5	Tabla vacía
SM0.6	OFF 1 ciclo / ON 1 ciclo	SM1.6	Error de conversión de BCD a binario
SM0.7	Selector en posición RUN	SM1.7	Error de conversión ASCII a hexadecimal

Tabla G-2 Descripción de los eventos de interrupción

Nº de evento	Descripción de la interrupción	Prioridad	Prioridad de grupo
8	Puerto 0 Recibir carácter	Comunicación (más alta)	0
9	Puerto 0 Transmisión finalizada		0
23	Puerto 0 Recepción de mensajes finalizada		0
24	Puerto 1: Recepción de mensajes finalizada		1
25	Puerto 1: Recibir carácter		1
26	Puerto 1: Transmisión finalizada		1
0	Flanco positivo, I0.0	Digital (media)	0
2	Flanco positivo, I0.1		1
4	Flanco positivo, I0.2		2
6	Flanco positivo, I0.3		3
1	Flanco negativo, I0.0		4
3	Flanco negativo, I0.1		5
5	Flanco negativo, I0.2		6
7	Flanco negativo, I0.3		7
12	HSC0 CV=PV (valor actual = valor predeterminado)		0
27	HSC0 cambio de sentido		16
28	HSC0, puesto a 0 externamente/fase Z		2
13	HSC1 CV=PV (valor actual = valor predeterminado)		8
14	HSC1, cambio de sentido		9
15	HSC1, puesto a 0 externamente		10
16	HSC2 CV=PV (valor actual = valor predeterminado)		11
17	HSC2 cambio de sentido		12
18	HSC2, puesto a 0 externamente		13
32	HSC3 CV=PV		1
29	HSC4 CV=PV		3
30	HSC4 cambio de sentido		17
31	HSC4, puesto a 0 externamente/fase Z	18	
33	HSC5 CV=PV	19	
19	PTO 0 interrupción completa	14	
20	PTO 1 interrupción completa	15	
10	Interrupción temporizada 0	Temporizada (más baja)	0
11	Interrupción temporizada 1		1
21	Interrupción temporizador T32 CT = PT		2
22	Interrupción temporizador T96 CT = PT		3

Tabla G-3 Resumen de las áreas de memoria y funciones de las CPUs S7-200

Descripción	Margen			Accesible como ...			
	CPU 221	CPU 222	CPU 224	Binario	BYTE	WORD	Palabra doble
Tamaño del programa de usuario	2K palabras	2K palabras	4K palabras				
Tamaño de los datos de usuario	1K palabras	1K palabras	2,5K palabras				
Imagen del proceso de las entradas	I0.0 a I15.7	I0.0 a I15.7	I0.0 a I15.7	Ix.y	IBx	IWx	IDx
Imagen del proceso de las salidas	Q0.0 a Q15.7	Q0.0 a Q15.7	Q0.0 a Q15.7	Qx.y	QBx	QWx	QDx
Entradas analógicas (sólo lectura)	—	AIW0 a AIW30	AIW0 a AIW30			AIWx	
Salidas analógicas (sólo escritura)	—	AQW0 a AQW30	AQW0 a AQW30			AQWx	
Memoria de variables (V) <sup>1</sup>	VB0.0 a VB2047.7	VB0.0 a VB2047.7	VB0.0 a VB5119.7	Vx.y	VBx	VWx	VDx
Memoria local (L) <sup>2</sup>	LB0.0 a LB63.7	LB0.0 a LB63.7	LB0.0 a LB63.7	Lx.y	LBx	LWx	LDx
Área de marcas (M)	M0.0 a M31.7	M0.0 a M31.7	M0.0 a M31.7	Mx.y	MBx	MWx	MDx
Marcas especiales (SM)	SM0.0 a SM179.7	SM0.0 a SM179.7	SM0.0 a SM179.7	SMx.y	SMBx	SMWx	SMDx
Sólo lectura	SM0.0 a SM29.7	SM0.0 a SM29.7	SM0.0 a SM29.7				
Temporizadores	256 (T0 a T255)	256 (T0 a T255)	256 (T0 a T255)	Tx		Tx	
Ret. a la conexión mem. 1 ms	T0, T64	T0, T64	T0, T64				
Ret. a la conexión mem. 10 ms	T0, T64	T0, T64	T0, T64				
Ret. a la conexión mem. 100 ms	T1 a T4, T65 a T68	T1 a T4, T65 a T68	T1 a T4, T65 a T68				
Retardo a la con./descon. 1 ms	T1 a T4, T65 a T68	T1 a T4, T65 a T68	T1 a T4, T65 a T68				
Retardo a la con./descon. 10 ms	T5 a T31, T69 a T95	T5 a T31, T69 a T95	T5 a T31, T69 a T95				
Retardo a la con./descon. 100 ms	T5 a T31, T69 a T95	T5 a T31, T69 a T95	T5 a T31, T69 a T95				
	T32, T96	T32, T96	T32, T96				
	T33 a T36, T97 a T100	T33 a T36, T97 a T100	T33 a T36, T97 a T100				
	T37 a T63, T101 a T255	T37 a T63, T101 a T255	T37 a T63, T101 a T255				
Contadores	C0 a C255	C0 a C255	C0 a C255	Cx		Cx	
Contadores rápidos	HC0, HC3, HC4, HC5	HC0, HC3, HC4, HC5	HC0 a HC5				HCx
Relés de control secuencial (S)	S0.0 a S31.7	S0.0 a S31.7	S0.0 a S31.7	Sx.y	SBx	SWx	SDx
Acumuladores	AC0 a AC3	AC0 a AC3	AC0 a AC3		ACx	ACx	ACx
Salto a metas	0 a 255	0 a 255	0 a 255				
Llamadas a subrutinas	0 a 63	0 a 63	0 a 63				
Rutinas de interrupción:	0 a 127	0 a 127	0 a 127				
Lazos PID	0 a 7	0 a 7	0 a 7				
Puerto	Puerto 0	Puerto 0	Puerto 0				

<sup>1</sup> Todo el contenido de la memoria V se puede guardar en la memoria no volátil.

<sup>2</sup> STEP 7-Micro/WIN 32 (versión 3.0 o posterior) reserva LB60 a LB63.

Tabla G-4 Contadores rápidos HSC0, HSC3, HSC4 y HSC5

Modo	HSC0			HSC3	HSC4			HSC5
	I0.0	I0.1	I0.2	I0.1	I0.3	I0.4	I0.5	I0.4
0	Reloj	–	–	Reloj	Reloj	–	–	Reloj
1	Reloj	–	Puesta a 0	–	Reloj	–	Puesta a 0	–
2	–	–	–	–	–	–	–	–
3	Reloj	Sentido	–	–	Reloj	Sentido	–	–
4	Reloj	Sentido	Puesta a 0	–	Reloj	Sentido	Puesta a 0	–
5	–	–	–	–	–	–	–	–
6	Reloj adelante	Reloj atrás	–	–	Reloj adelante	Reloj atrás	–	–
7	Reloj adelante	Reloj atrás	Puesta a 0	–	Reloj adelante	Reloj atrás	Puesta a 0	–
8	–	–	–	–	–	–	–	–
9	Fase A	Fase B	–	–	Fase A	Fase B	–	–
10	Fase A	Fase B	Puesta a 0	–	Fase A	Fase B	Puesta a 0	–
11	–	–	–	–	–	–	–	–

Tabla G-5 Contadores rápidos HSC1 y HSC2

Modo	HSC1				HSC2			
	I0.6	I0.7	I1.0	I1.1	I1.2	I1.3	I1.4	I1.5
0	Reloj	–	–	–	Reloj	–	–	–
1	Reloj	–	Puesta a 0	–	Reloj	–	Puesta a 0	–
2	Reloj	–	Puesta a 0	Arranque	Reloj	–	Puesta a 0	Arranque
3	Reloj	Sentido	–	–	Reloj	Sentido	–	–
4	Reloj	Sentido	Puesta a 0	–	Reloj	Sentido	Puesta a 0	–
5	Reloj	Sentido	Puesta a 0	Arranque	Reloj	Sentido	Puesta a 0	Arranque
6	Reloj adelante	Reloj atrás	–	–	Reloj adelante	Reloj atrás	–	–
7	Reloj adelante	Reloj atrás	Puesta a 0	–	Reloj adelante	Reloj atrás	Puesta a 0	–
8	Reloj adelante	Reloj atrás	Puesta a 0	Arranque	Reloj adelante	Reloj atrás	Puesta a 0	Arranque
9	Fase A	Fase B	–	–	Fase A	Fase B	–	–
10	Fase A	Fase B	Puesta a 0	–	Fase A	Fase B	Puesta a 0	–
11	Fase A	Fase B	Puesta a 0	Arranque	Fase A	Fase B	Puesta a 0	Arranque

Operaciones booleanas			Operaciones aritméticas, incrementar y decrementar		
LD	N	Cargar	+I	IN1, OUT	Sumar enteros, palabras dobles o números reales
LDI	N	Cargar directamente	+D	IN1, OUT	IN1+OUT=OUT
LDN	N	Cargar valor negado	+R	IN1, OUT	OUT-IN1=OUT
LDNI	N	Cargar valor negado directamente	-I	IN1, OUT	Restar enteros, palabras dobles o números reales
A	N	Combinación lógica Y	-D	IN1, OUT	OUT-IN1=OUT
AI	N	Y directa	-R	IN1, OUT	Multiplicar enteros o números reales
AN	N	Y-NO	*R	IN1, OUT	IN1 * OUT = OUT
ANI	N	Y-NO directa	*D, *I	IN1, OUT	Multiplicar enteros o enteros dobles
O	N	Combinación lógica O	DIV	IN1, OUT	Dividir enteros o números reales
OI	N	O directa	/R	IN1, OUT	IN1 / OUT = OUT
ON	N	O-NO	/D, /I	IN1, OUT	Dividir enteros o enteros dobles
ONI	N	O-NO directa	SQRT	IN, OUT	Raíz cuadrada
LDBx	N1, N2	Cargar resultado de la comparación de bytes N1 (x:<, <=, =, >=, >, <>) N2	INCB	OUT	Incrementar byte, palabra o palabra doble
ABx	N1, N2	Combinar mediante Y el resultado de la comparación de bytes N1 (x:<, <=, =, >=, >, <>) N2	INCW	OUT	
OBx	N1, N2	Combinar mediante O el resultado de la comparación de bytes N1 (x:<, <=, =, >=, >, <>) N2	INCD	OUT	
LDWx	N1, N2	Cargar resultado de la comparación de palabras N1 (x:<, <=, =, >=, >, <>) N2	DECB	OUT	Decrementar byte, palabra o palabra doble
AWx	N1, N2	Combinar mediante Y el resultado de la comparación de palabras N1 (x:<, <=, =, >=, >, <>) N2	DECW	OUT	
OWx	N1, N2	Combinar mediante O el resultado de la comparación de palabras N1 (x:<, <=, =, >=, >, <>) N2	DECD	OUT	
LDDx	N1, N2	Cargar resultado de la comparación de palabras dobles N1 (x:<, <=, =, >=, >, <>) N2	PID	Table, Loop	Regulación PID
ADx	N1, N2	Combinar mediante Y el resultado de la comparación de palabras dobles N1 (x:<, <=, =, >=, >, <>) N2	<b>Operaciones de temporización y conteo</b>		
ODx	N1, N2	Combinar mediante O el resultado de la comparación de palabras dobles N1 (x:<, <=, =, >=, >, <>) N2	TON	Txxx, PT	Temporizador con retardo al conectar
LDRx	N1, N2	Cargar resultado de la comparación de números reales N1 (x:<, <=, =, >=, >, <>) N2	TOF	Txxx, PT	Temporizador con retardo al desconectar
ARx	N1, N2	Combinar mediante Y el resultado de la comparación de números reales N1 (x:<, <=, =, >=, >, <>) N2	TONR	Txxx, PT	Temporizador de retardo a la conexión memorizado
ORx	N1, N2	Combinar mediante O el resultado de la comparación de números reales N1 (x:<, <=, =, >=, >, <>) N2	CTU	Cxxx, PV	Contar adelante
NOT		Negar primer valor de pila	CTD	Cxxx, PV	Contar atrás
EU		Detectar flanco positivo	CTUD	Cxxx, PV	Contar adelante/atrás
ED		Detectar flanco negativo	<b>Operaciones del reloj de tiempo real</b>		
=	N	Asignar	TODR	T	Leer reloj de tiempo real
=I	N	Asignar directamente	TODW	T	Escribir reloj de tiempo real
S	S_BIT, N	Poner a 1 (activar)	<b>Operaciones de control del programa</b>		
R	S_BIT, N	Poner a 0 (desactivar)	END		Fin condicional del programa
SI	S_BIT, N	Poner a 1 directamente	STOP		Pasar a STOP
RI	S_BIT, N	Poner a 0 directamente	WDR		Borrar temporizador de vigilancia (300 ms)
			JMP	N	Saltar a meta
			LBL	N	Definir meta
			CALL	N [N1,...]	Llamar a subrutina [N1, ... hasta 16 parámetros opcionales]
			CRET		Retorno condicional de subrutina
			FOR	INDX,INI T, FINAL	Bucle FOR/NEXT
			NEXT		
			LSCR	N	Cargar, transición y fin del relé de control secuencial
			SCRT	N	Segmento
			SCRE		
			<b>Operaciones de transferencia, desplazamiento, rotación e inicialización</b>		
			MOVB	IN, OUT	Transferir bytes, palabras, palabras dobles o números reales
			MOVW	IN, OUT	
			MOVD	IN, OUT	
			MOVR	IN, OUT	

BMB IN, OUT, N			FND= SRC,PAT RN, INDX	
BMW IN, OUT, N		Transferir bytes, palabras o palabras dobles en bloque	FND<>SRC,PAT RN, INDX	Buscar valor en tabla que concuerde con la comparación
BMD IN, OUT, N			FND< SRC,PATRN, INDX	
SWAP IN		Invertir bytes de una palabra	FND> SRC,PATRN, INDX	
SHRB DATA, S_BIT, N		Registro de desplazamiento	BCDI OUT	Convertir de BCD a entero
SRB OUT, N		Desplazar byte, palabra o palabra doble a la derecha	IBCD OUT	Convertir de entero a BCD
SRW OUT, N			BTI IN, OUT	Convertir de byte a entero
SRD OUT, N			ITB IN, OUT	Convertir de entero a byte
SLB OUT, N		Desplazar byte, palabra o palabra doble a la izquierda	ITD IN, OUT	Convertir de entero a entero doble
SLW OUT, N			DTI IN, OUT	Convertir de entero doble a entero
SLD OUT, N			DTR IN, OUT	Convertir de palabra doble a real
RRB OUT, N		Rotar byte, palabra o palabra doble a la derecha	TRUNC IN, OUT	Convertir de real a palabra doble
RRW OUT, N			ROUND IN, OUT	Convertir de real a entero doble
RRD OUT, N			ATH IN, OUT, LEN	
RLB OUT, N		Rotar byte, palabra o palabra doble a la izquierda	HTA IN, OUT, LEN	Convertir de ASCII a hexadecimal
RLW OUT, N			ITA IN, OUT, FMT	Convertir de hexadecimal a ASCII
RLD OUT, N			DTA IN, OUT, FM	Convertir de entero a ASCII
FILL IN, OUT, N		Inicializar memoria	RTA IN, OUT, FM	Convertir de entero doble a ASCII
<b>Operaciones lógicas</b>			DECO IN, OUT	Decodificar
ALD		Combinar primer y segundo valor mediante Y	ENCO IN, OUT	Codificar
OLD		Combinar primer y segundo valor mediante O	SEG IN, OUT	Generar configuración binaria de 7 segmentos
LPS		Duplicar primer valor de la pila	<b>Operaciones de interrupción</b>	
LRD		Copiar segundo valor de la pila	CRETI	Retorno condicional desde rutina de interrupción
LPP		Sacar primer valor de la pila	ENI	Habilitar todos los eventos de interrupción
LDS		Cargar pila	DISI	Inhibir todos los eventos de interrupción
AENO		Y-ENO	ATCH INT, EVENT	Asociar interrupción
ANDB IN1, OUT		Combinación Y con bytes, palabras o palabras dobles	DTCH EVENT	Desasociar interrupción
ANDW IN1, OUT			<b>Operaciones de comunicación</b>	
ANDD IN1, OUT			XMT TABLE,P ORT	Transmitir mensaje (Freeport)
ORB IN1, OUT		Combinación O con bytes, palabras o palabras dobles	RCV TABLE,P ORT	Recibir mensaje (Freeport)
ORW IN1, OUT			NETR TABLE,P ORT	Leer de la red
ORD IN1, OUT			NETW TABLE,P ORT	Escribir en la red
XORB IN1, OUT		Combinación O-exclusiva con bytes, palabras o palabras dobles	<b>Operaciones con contadores rápidos</b>	
XORW IN1, OUT			HDEF HSC, Mode	Definir modo para contador rápido
XORD IN1, OUT			HSC N	Activar contador rápido
INVB OUT		Invertir byte, palabra o palabra doble (complemento a 1)	PLS X	Salida de impulsos
INWV OUT			<b>Operaciones de tabla, de búsqueda y de conversión</b>	
INVD OUT			ATT TABLE,D ATA	Registrar valor en tabla
LIFO TABLE,D ATA		Obtener datos de la tabla	<b>Operaciones con contadores rápidos</b>	
FIFO TABLE,D ATA			HDEF HSC, Mode	Definir modo para contador rápido
			HSC N	Activar contador rápido
			PLS X	Salida de impulsos

## Índice alfabético

- Acceder, áreas de operandos, 8-8
- Acceso
- a áreas de memoria, modificar un puntero, 5-14
  - áreas de memoria
    - direccionamiento indirecto, 5-13–5-15
    - direccionamiento directo, 5-2
- Acceso a bits, 5-2
- Acceso a bytes, 5-2
- utilizando punteros, 5-14
- Acceso a palabras, 5-2
- utilizando punteros, 5-14
- Acceso mediante bits, CPU 221/222/224, 8-8
- Acceso mediante bytes, CPU 221/222/224, 8-8
- Acceso mediante palabras, CPU 221/222/224, 8-8
- Acceso mediante palabras dobles, CPU 221/222/224, 8-8
- Activar contador rápido (HSC), 9-27
- Actualización síncrona, función PWM, 9-59
- Acumuladores, direccionamiento, 5-10
- Adaptador
- de módem nulo, 7-25–7-26
  - módem nulo, 7-40
- Adaptador de módem nulo, 7-25–7-26, 7-37, 7-40
- Ajustar reloj de tiempo real, 9-70
- Ajustes en el cuadro de diálogo Interface PG/PC, 7-6
- Algoritmo para regulación PID, 9-85–9-89
- Algoritmo PID, 9-85–9-89
- Almacenar el programa en la memoria no volátil, 5-20
- Área de marcas, direccionamiento, 5-5
- Área de memoria de variables, direccionamiento, 5-5
- Áreas de funciones, crear, 4-3
- Áreas de memoria, G-3
- acceder a los datos, 5-2
  - áreas de operandos, 8-8
  - bits, 5-2
  - bytes, 5-2
  - CPU, 5-2
- Áreas de operandos, CPU 221/222/224, 8-8
- Áreas remanentes, definir, 5-19
- Armario eléctrico
- desmontaje del, 2-8
  - dimensiones
    - CPU 221, 2-4
    - CPU 222, 2-4
  - módulos de ampliación, 2-5
  - montaje, 2-6
- Asignación de pines, puerto de comunicación, 7-31
- Asignar, 9-6
- Asignar directamente, 9-6
  - Asignar salidas, configurar el estado de las salidas, 6-8
- Asociar interrupción, 9-165
- Ayuda en pantalla
- STEP 7-Micro/WIN 32, 3-2
- B**
- Bias (suma integral), algoritmo PID, 9-87
- Bits, 5-2
- Bits de estado (SMB0), C-1
- Bits, marcas especiales, C-1–C-13
- Bloque de terminales
- CPU 224, 2-12
  - módulo de ampliación, 2-12
  - soltar, 2-12
- Borrar primer registro de la tabla, 9-108
- Borrar temporizador de vigilancia, 9-142–9-144
- consideraciones, 9-142
- Borrar último registro de la tabla, 9-109
- Buscar valor en tabla, 9-106
- Byte de estado de los contadores rápidos, 9-39
- Byte, formato de direcciones, 5-2
- Bytes, margen de enteros, 5-4
- Bytes de programación de los contadores rápidos, C-9
- C**
- Cable de ampliación, datos técnicos e instalación, A-29

- Cable de cinta flexible, módulos de ampliación, 1-5
- Cable de E/S de ampliación, instalación, A-29
- Cable PC/PPI
  - ajustar los interruptores DIP, 3-5, 7-38
  - ajustar los parámetros, 7-10
  - asignación de pines, A-31
  - conexión, 3-5, 7-38
  - datos técnicos, A-30
  - posición de los interruptores DIP para seleccionar la velocidad de transferencia, 7-35, A-30
  - utilización con un módem, 7-37, 7-40
  - utilización en modo Freeport, 7-35-7-36
  - utilizar con un módem, 7-25-7-26
- Cableado
  - bloque de bornes opcional, 2-11
  - circuitos de supresión, 2-16-2-17
  - desmontar módulos, 2-8
  - entradas de los contadores rápidos, 9-32
  - reglas, 2-9-2-14
    - instalación con corriente alterna, 2-13
    - instalación con corriente continua, 2-14
- Cableado (diagrama)
  - CPU 214 AC/DC/relé, A-20
  - CPU 221 AC/DC/relé, A-10
  - CPU 221 DC/DC/DC, A-10
  - CPU 222 AC/DC/relé, A-15
  - CPU 222 DC/DC/DC, A-15
  - CPU 224 DC/DC/DC, A-20
  - EM221, 8 entradas digitales x DC 24 V, A-22
  - EM222, 8 salidas digitales x DC 24 V, A-24
  - EM222, 8 salidas digitales x relé, A-24
  - EM223, 8 entradas digitales DC 24 V / 8 salidas digitales de relé, A-27
  - EM223, 8 entradas digitales/8 salidas digitales, A-27
- Cableado de campo
  - conector opcional, 2-11
  - diámetro, 2-9
  - instalación, 2-9
- Cables
  - desmontar módulos, 2-8
  - nº de referencia, E-2
  - PC/PPI, ajustar los parámetros, 7-10
  - red PROFIBUS, 7-33
- Calcular la corriente necesaria, 2-20
- Calcular requisitos de alimentación, 2-18
- Cambiar un puntero, 5-14
- Canadian Standards Association (CSA), A-2
- Captura de impulsos, 6-5
- Cargar en la CPU
  - modo de operación necesario, 4-25
  - programa, 5-15
- Cargar en la PG, programa, 5-15
- Cargar pila, 9-193-9-195
- Cartucho de memoria
  - códigos de error, B-2
  - copiar en, 5-22
  - datos técnicos, A-28
  - dimensiones, A-28
  - instalar, 5-22
  - nº de referencia, E-1
  - restablecer el programa, 5-24
  - retirar, 5-22
  - utilización, 5-22
- Cartucho de pila, 5-15
  - datos técnicos, A-28
  - nº de referencia, E-1
- Cartucho de reloj, datos técnicos, A-28
- Casilla de verificación Red multimaestro, 7-11
- Cerrar, redes, 7-32
- Certificado CE, A-2
- Ciclo
  - bits de estado, C-1
  - interrumpir, 4-24
  - tareas, 4-22
  - y la función de forzado, 4-34
  - y tabla de estado/forzado, 4-34
- Circuitos de seguridad cableados, diseñar, 4-3
- Circuitos de supresión, reglas
  - salidas AC, 2-17
  - transistores en DC, 2-16, 2-17
- Codificar, 9-131
- Combinación O con bytes, 9-110
- Combinación O con palabras, 9-111
- Combinación O con palabras dobles, 9-112
- Combinación O exclusiva con bytes, 9-110
- Combinación O exclusiva con palabras, 9-111
- Combinación O-exclusiva con palabras dobles, 9-112
- Combinación Y con bytes, 9-110
- Combinación Y con palabras, 9-111
- Combinación Y con palabras dobles, 9-112
- Combinar primer y segundo valor mediante O, 9-192-9-194
- Combinar primer y segundo valor mediante Y, 9-192-9-194
- Comenzar rutina de interrupción, 9-167
- Comparación, CPUs S7-200, 1-3
- Comparar byte, 9-10
- Comparar entero, 9-11
- Comparar palabra doble, 9-12
- Comparar real, 9-13
- Compatibilidad electromagnética, S7-200, A-4
- Compilación, errores
  - reacción del sistema, 4-38
  - violación de reglas, B-4
- Comprobación de datos, 5-12
- Comprobar, programa, 4-30-4-32

- Comunicación  
 cambiar los parámetros de la CPU, 3-10  
 componentes de redes, 7-31  
 conectar el PC, 7-2  
 conectar un cable PC/PPI, 3-5  
 configurar, 7-2-7-19  
 establecer con la CPU, 3-9  
 hardware  
     instalar en Windows NT, 7-8  
     instalar/desinstalar, 3-2-3-4  
 módem, 7-25-7-30  
 modo Freeport, 9-183, C-6  
 MPI, 7-29  
 PPI, 7-2, 7-29  
 procesar peticiones, 4-23  
 protocolo PROFIBUS, 7-30  
 protocolos asistidos, 7-28  
 seleccionar la parametrización utilizada,  
     7-9-7-10  
 utilizando una tarjeta CP, 7-4-7-5  
 utilizando una tarjeta MPI, 7-4-7-5  
 utilizar módems, 7-16  
 velocidades de transferencia, 7-26  
 verificar la configuración, 7-4
- Comunicación MPI, tarjetas CP, 7-4
- Comunicación punto a punto, 1-3
- Comunidad Europea (CE), certificado, A-2
- Conceptos de programación, 4-5
- Condensador de alto rendimiento, 5-15
- Condiciones ambientales, A-3
- Conectar la alimentación, respaldar datos,  
 5-17-5-21
- Conector de bus, desmontar módulos de amplia-  
 ción, 2-8
- Conectores  
 de bus, 7-32  
 nº de referencia, E-2  
 puerto de ampliación de bus, retirar la tapa,  
 2-8
- Configuración  
 áreas remanentes, 5-19  
 comunicación, 7-2-7-19  
 crear planos, 4-4  
 de un PC con tarjeta CP y unidad de progra-  
 mación, 7-12  
 de un PC con tarjeta MPI y unidad de progra-  
 mación, 7-12  
 estado de las salidas, 6-8  
 hardware de comunicación, 3-2, 7-3
- Configurar, parámetros de comunicación, 7-4
- Congelar salidas, 6-8
- Consideraciones  
 al utilizar la operación Borrar temporizador de  
 vigilancia, 9-142  
 entornos con vibraciones fuertes, 2-7  
 montaje del hardware, 2-2-2-4  
 montaje vertical, 2-7  
 utilización de topes, 2-7
- Constantes, 5-12
- Constantes ASCII, 5-4
- Constantes decimales, 5-4
- Constantes hexadecimales, 5-4
- Contactos de comparación, ejemplo, 9-14
- Contactos directos, operaciones, 9-3
- Contactos estándar, operaciones, 9-2
- Contador rápido, 6-10  
 cambio de sentido, 9-45  
 ejemplos, 9-47
- Contadores  
 CPU 221/222/224, 8-7  
 direccionamiento, 5-8  
 tipos, 5-8  
 variables, 5-8
- Contadores rápidos, 9-27-9-46  
 ajustar los valores actuales y predeterminados,  
 9-38  
 área de memoria, direccionamiento, 5-11  
 byte de control, 9-38  
 byte de estado, 9-39  
 bytes de programación (SMB36 – SMB65),  
 C-9  
 cableado de las entradas, 9-32  
 cargar nuevo valor actual/predeterminado,  
 9-45  
 cronogramas, 9-28-9-31  
 direccionamiento, 9-36  
 ejemplos, 9-28-9-31  
 elegir el nivel de actividad, 9-37  
 inhibir, 9-46  
 interrupciones, 9-39  
 modos, G-4  
 modos de inicialización, 9-41-9-44  
 modos de operación, 9-33  
 utilización, 9-28
- Contraseña  
 borrar, 4-29  
 CPU, 4-27  
     configurar, 4-28  
 nivel de protección, 4-27  
 olvidada, 4-29  
 restringir el acceso, 4-27
- Control de escritura, C-7

- Control de interrupción de caracteres, 9-188
- Control de lazos PID, modos, 9-92
- Control directo de las entradas y salidas, 4-24
- Convenciones, programación con Micro/WIN 32, 8-2
- Conversión, operaciones de, 4-16
- Convertir
  - enteros de 16 bits a números reales, 9-89
  - entradas del lazo, 9-89
  - números reales a valores normalizados, 9-89
- Convertir de ASCII a hexadecimal, 9-135
- Convertir de BCD a entero, 9-126
- Convertir de byte a entero, 9-129
- Convertir de entero a ASCII, 9-136
- Convertir de entero a BCD, 9-126
- Convertir de entero a byte, 9-129
- Convertir de entero a entero doble, 9-128
- Convertir de entero doble a ASCII, 9-138
- Convertir de entero doble a entero, 9-128
- Convertir de entero doble a real, 9-126
- Convertir de hexadecimal a ASCII, 9-135
- Convertir de real a ASCII, 9-139
- Copiar segundo valor, 9-192–9-194
- Corriente necesaria, calcular, 2-20
- CP (procesador de comunicaciones), 7-4
  - configuración con PC, 7-12
- CP 5511
  - ajustar los parámetros de la tarjeta MPI (PPI), 7-14
  - nº de referencia, E-1
- CP 5611
  - ajustar los parámetros de la tarjeta MPI (PPI), 7-14
  - nº de referencia, E-1
- CPU
  - ajustar el modo de operación, 4-25
  - áreas de memoria, 5-2, G-3
  - áreas de operandos, 8-8
  - borrar memoria, 4-29
  - cambiar los parámetros de comunicación, 3-10
  - ciclo, 4-22
  - conexión de módem, 7-25–7-30
  - contraseña, 4-27
  - datos técnicos generales, A-3
  - dimensiones
    - CPU 221, 2-4
    - CPU 222, 2-4
    - CPU 224, 2-5
    - módulos de ampliación, 2-5
    - tornillos para el montaje, 2-4–2-6
  - eliminación de errores, 4-36
  - errores fatales, B-2
  - establecer la comunicación con, 3-9
  - funcionamiento básico, 4-5
  - hardware asistido para la comunicación en redes, 7-3
  - identificador (SMB6), C-4
  - módulos, 1-5
  - montaje, armario eléctrico, 2-6
  - procedimiento, desmontaje, 2-8
  - requisitos de alimentación, 2-18
  - tornillos para el montaje, 2-4–2-6
- CPU 212, ejemplo de numeración de E/S, 6-3
- CPU 221
  - áreas de operandos, 8-8
  - backup, 1-3
  - E/S, 1-3
  - filtros de entrada, 1-3
  - funciones, 8-7
  - interrupciones, número máximo, 9-172
  - interrupciones asistidas, 1-3
  - memoria, 1-3
    - márgenes, 8-7
  - módulos de ampliación, 1-3
  - operaciones asistidas, 1-3
  - protocolos asistidos, 1-3
  - puertos de comunicación, 1-3
  - resumen, 1-3
- CPU 221 AC/DC/Relé, nº de referencia, E-1
- CPU 221 AC/DC/relé
  - datos técnicos, A-6
  - identificación de terminales de conexión, A-10
- CPU 221 DC/DC/DC
  - datos técnicos, A-6
  - identificación de terminales de conexión, A-10
  - nº de referencia, E-1

- CPU 222  
 áreas de operandos, 8-8  
 copias de seguridad, 1-3  
 E/S, 1-3  
 filtros de entrada, 1-3  
 funciones, 8-7  
 interrupciones, número máximo, 9-172  
 interrupciones asistidas, 1-3  
 memoria, 1-3  
     márgenes, 8-7  
 módulos de ampliación, 1-3  
 operaciones asistidas, 1-3  
 protocolos asistidos, 1-3  
 puertos de comunicación, 1-3  
 resumen, 1-3
- CPU 222 AC/DC/Relé, nº de referencia, E-1
- CPU 222 AC/DC/relé  
 datos técnicos, A-11  
 identificación de terminales de conexión, A-15
- CPU 222 DC/DC/DC  
 datos técnicos, A-11  
 identificación de terminales de conexión, A-15
- CPU 222 DC/DC/DC , nº de referencia, E-1
- CPU 224  
 áreas de operandos, 8-8  
 bloque de terminales, 2-12  
 copias de seguridad, 1-3  
 E/S, 1-3  
 ejemplo de numeración de E/S, 6-3  
 ejemplos de numeración de E/S, 6-3  
 filtros de entrada, 1-3  
 funciones, 8-7  
 interrupciones, número máximo, 9-172  
 interrupciones asistidas, 1-3  
 márgenes de memoria , 8-7  
 memoria, 1-3  
     márgenes, 8-7  
 módulos de ampliación, 1-3  
 operaciones asistidas, 1-3  
 protocolos asistidos, 1-3  
 puertos de comunicación, 1-3  
 resumen, 1-3
- CPU 224 AC/DC/Relé, nº de referencia, E-1
- CPU 224 AC/DC/relé  
 datos técnicos, A-16  
 identificación de terminales de conexión, A-20
- CPU 224 DC/DC/DC  
 datos técnicos, A-16  
 identificación de terminales de conexión, A-20  
 nº de referencia, E-1
- Crear un programa, ejemplo: ajustar una interrupción temporizada, 4-18
- Cronogramas, contadores rápidos, 9-28
- Cuadro de diálogo , 7-6
- Cuadro de diálogo Instalar/desinstalar interfaces, 7-7
- Cuadro de diálogo Recursos" para Windows NT, 7-8
- Cuadro Registro de desplazamiento (SHRB), 9-124
- D**
- Datos técnicos, gama S7-200, A-3
- Decodificar, 9-131
- Decrementar, ejemplo, 9-80
- Decrementar byte, 9-78
- Decrementar palabra, 9-78
- Decrementar palabra doble, 9-79
- Definir meta, 9-144
- Definir modo para contador rápido, 9-27  
     modos de los contadores, 9-36
- Definir modo para contador rápido (HDEF), 9-27
- Desasociar interrupción, 9-165
- Desbordamiento de la cola de espera (SMB4), C-3
- Desmontaje  
 CPU, 2-8  
 dimensiones  
     CPU 221, 2-4  
     CPU 222, 2-4  
     CPU 224, 2-5  
     módulos de ampliación, 2-5  
     tornillos para el montaje, 2-4–2-6  
 espacio necesario para el montaje, 2-3  
 módulo de ampliación, 2-8  
 orientación correcta del módulo, 2-8  
 procedimiento, 2-8  
 tornillos para el montaje, 2-4–2-6
- Desplazar byte a la derecha, 9-116
- Desplazar byte a la izquierda, 9-116
- Desplazar palabra a la derecha, 9-117
- Desplazar palabra a la izquierda, 9-117
- Desplazar palabra doble a la derecha, 9-118
- Desplazar palabra doble a la izquierda, 9-118
- Detectar flanco negativo, 9-4
- Detectar flanco positivo, 9-4
- Determinación del tipo de datos, 5-12
- Diagrama de funciones, elementos básicos, 4-6
- Dimensiones  
     cartucho de memoria, A-28  
     CPU 221, 2-4  
     CPU 222, 2-4  
     CPU 224, 2-5  
     módulos de ampliación, 2-5  
     tornillos para el montaje, 2-4–2-6
- Diodos supresores, 2-16

Dirección de estación más alta, 7-41

Direccionamiento

acumuladores, 5-10

área de marcas, 5-5

áreas de memoria, 5-2

byte.bit, 5-2

contadores, 5-8

contadores rápidos, 5-11, 9-36

E/S de los módulos de ampliación, 6-2

E/S integradas, 6-2

entradas analógicas, 5-9

equipos en una red, 7-28

imagen del proceso de las entradas, 5-4

imagen del proceso de las salidas, 5-4

indirecto (punteros), 5-13–5-15

    modificar un puntero, 5-14

marcas especiales, 5-5

memoria de variables, 5-5

relés de control secuencial, 5-5

salidas analógicas, 5-9

temporizadores, 5-7

Direccionamiento directo, 5-2

    operaciones sobrecargadas, 4-15

Direccionamiento indirecto, 5-13–5-15

    modificar un puntero, 5-14

Diseñar una solución de automatización con un  
Micro-PLC, 4-2

Diseño de sistemas, Micro-PLC, 4-2

Dividir enteros de 16 bits, 9-74

Dividir enteros de 16 bits a enteros de 32 bits,  
9-76

Dividir enteros de 32 bits, 9-75

Dividir reales, 9-82

Documentación adicional, iv

Duplicar primer valor, 9-192–9-194

## E

E/S integradas, direccionamiento, 6-2

Editor AWL (Lista de instrucciones), 4-6

Editor FUP (Diagrama de funciones), 4-9

Editor KOP (Esquema de contactos), 4-8

Editores

    AWL (Lista de instrucciones), 4-6

    FUP (Diagrama de funciones), 4-9

    KOP (Esquema de contactos), 4-8

EEPROM, 5-15, 5-17

    códigos de error, B-2

    copiar memoria de variables en, 5-20

    guardar de la memoria V, C-7

Ejemplos

    ajustar una interrupción temporizada, 4-18

    aritmética, 9-77, 9-83

    Borrar primer registro de la tabla, 9-108

    Borrar último registro de la tabla, 9-109

    Buscar valor en tabla, 9-107

    calcular requisitos de alimentación, 2-18

    contador rápido, 9-47

    funcionamiento con puesta a 0 y arranque,  
    9-29

    funcionamiento con puesta a 0 y sin arran-  
    que, 9-28

    funcionamiento de HSC0 en modo 0 y de  
    HSC1 en modos 0, 1 ó 2, 9-29

    funcionamiento de HSC1 o HSC2 en modo  
    3, 4 ó 5, 9-30

    funcionamiento de HSC1 o HSC2 en modo  
    6, 7 u 8, 9-30

    funcionamiento de HSC1 o HSC2 en modo  
    9, 10 u 11, 9-31

    funcionamiento de HSC1 ó HSC2 en modo  
    9, 10 u 11, 9-31

    Convertir de ASCII a hexadecimal, 9-135

    Convertir números reales, 9-130

    Convertir y truncar, 9-130

    de operación lógica de pila, 9-195–9-197

    de una operación de conteo, 9-25

    Decodificar/Codificar, 9-132

    decrementar, 9-80

    desplazamiento y rotación, 9-122–9-124

    FOR/NEXT, 9-152–9-154

    incrementar, 9-80

    Inicializar memoria, 9-103–9-105

    Invertir, 9-115–9-117

    Leer de la red/Escribir en la red, 9-178–9-180

    Llamar subrutina, 9-149–9-151

    modulación del ancho de impulsos, 9-63

    numeración de E/S, 6-2, 6-3

    operación Registro de desplazamiento,  
    9-125–9-127

    operaciones con contactos, 9-5

    operaciones con rutinas de interrupción, 9-174

    operaciones con salidas, 9-9

- operaciones de comparación, 9-14  
operaciones de transmisión, 9-189  
potenciómetros analógicos, 6-13  
Registrar valor en tabla, 9-105  
regulación PID, 9-94–9-96  
relé de control secuencial, 9-155–9-160  
    convergir cadenas secuenciales,  
    9-159–9-162  
    dividir cadenas secuenciales, 9-157  
    transiciones condicionales, 9-162  
Saltar a meta, 9-144–9-146  
Segmento, 9-134  
STOP, END y Borrar temporizador de vigilancia, 9-143–9-145  
tarjeta MPI con esclavo/maestro, 7-4  
TDs 200 integrados en una red, 7-12  
temporizador de retardo a la conexión, 9-20,  
9-21  
temporizador de retardo a la desconexión,  
9-22  
transferencia de bloques, 9-101–9-103  
transferir e invertir, 9-102–9-104  
Tren de impulsos, 9-65, 9-68  
Truncar, 9-130  
Y, O y O-exclusiva, 9-113–9-115
- Eliminación  
errores de compilación, B-4  
errores de programación del tiempo de ejecución, B-3  
errores fatales, B-2
- Eliminación de errores, 4-36  
contraseña olvidada, 4-29  
fatales, 4-36, 4-37  
lazo PID, 9-93  
leer de la red/escribir en la red, 9-176  
no fatales, 4-38  
rearrancar la CPU después de un error fatal,  
4-37  
responder a errores, 4-36  
S7-200, D-1
- Eliminar fallos, instalación de Micro/WIN 32, 3-4  
EM221 DC 24 V, 8 entradas digitales, nº de referencia, E-1  
EM221, 8 entradas digitales DC 24 V, datos técnicos, A-21  
EM221, 8 entradas digitales x DC 24 V, identificación de terminales de conexión, A-22  
EM222 DC 24 V, 8 entradas digitales, nº de referencia, E-1  
EM222, 8 salidas de relé, nº de referencia, E-1  
EM222, 8 salidas digitales DC 24 V x relé, identificación de terminales de conexión, A-24  
EM222, 8 salidas digitales x DC 24 V, identificación de terminales de conexión, A-24  
EM222, salidas DC 24 V y salidas de relé, datos técnicos, A-23
- EM223 DC 24 V, 8 entradas digitales/8 salidas de relé, nº de referencia, E-1  
EM223 DC 24 V, 8 entradas digitales/8 salidas digitales, nº de referencia, E-1  
EM223, 8 entradas DC 24 V/8 salidas de relé, datos técnicos, A-25  
EM223, 8 entradas digitales DC 24 V / 8 salidas digitales de relé, identificación de terminales de conexión, A-27  
EM223, 8 entradas/8 salidas x DC 24 V  
datos técnicos, A-25  
identificación de terminales de conexión, A-27  
EN/ENO, reglas, 4-18  
END, 9-141  
Energía necesaria, tabla de cálculo, 2-20  
Enlaces, MPI lógicos, 7-29  
Enlaces lógicos, MPI, 7-29  
ENO instructions, 9-164  
Enteros de 16 bits, convertir a números reales,  
9-89  
Entornos con vibraciones fuertes, utilización de topes, 2-7  
Entradas, funcionamiento básico, 4-5  
Entradas analógicas  
    acceder a, 4-22  
    direccionamiento, 5-9  
    leer el valor con una rutina de interrupción,  
    9-175  
Entradas digitales  
    leer, 4-22  
    y captura de impulsos, 6-5  
Entradas y salidas rápidas, 6-10  
Equipos necesarios, S7-200, 1-2  
Errores  
    de programación del tiempo de ejecución, B-3  
    fatales, B-2  
    lazo PID, 9-93  
    Leer de la red/Escribir en la red, 9-176  
    no fatales, B-3, B-4  
    SMB1, errores de ejecución, C-2  
    violación de reglas de compilación, B-4  
Errores de programación del tiempo de ejecución,  
B-3  
Errores durante el tiempo de ejecución, reacción del sistema, 4-38  
Errores fatales, B-2  
    y el funcionamiento de la CPU, 4-37  
Errores no fatales  
    eliminación de, 4-38  
    reacción del sistema, 4-38  
    y el funcionamiento de la CPU, 4-38  
Escalar la salida del lazo, 9-90  
Escribir en la red, 9-176  
    ejemplo, 9-178–9-180  
    errores, 9-176  
Espacio necesario para el montaje, 2-3

Esquema de contactos, elementos básicos, 4-6  
Estaciones de operador, definir, 4-4  
Estado de las entradas y salidas (SMB5), C-3  
Eventos de interrupción, descripción, G-2

## F

Factor de actualización GAP, 7-41  
Factory Mutual Research, A-2  
Fecha, ajustar, 9-70  
Filtrar entradas analógicas, 6-9  
Filtro de entrada, supresión de ruidos, 6-4  
Filtros de entrada, y captura de impulsos, 6-5  
FOR, 9-150  
Forzar valores, 4-34  
Freeport  
  definición del modo, 9-169  
  protocolos definidos por el usuario, 7-30  
Freeport, modo, utilización del cable PC/PPI,  
  7-35–7-36  
Funciones PTO/PWM  
  inicialización, 9-58  
  registro de control, 9-56  
  registros de control, SMB66 – SMB85, C-11  
  tabla de referencia hexadecimal, 9-56  
Funciones PTO/PWM (SMB66–SMB85), C-11  
FUP, estado del programa, 4-33

## G

Ganancia del lazo, convertir, 9-89  
GAP. Véase Gap update factor  
Guardar  
  el programa en la memoria no volátil, 5-20  
  valor en EEPROM, C-7

## H

Habilitar todos los eventos de interrupción, 9-169  
Hardware  
  desinstalar en Micro/WIN 32, 7-7  
  instalar en Micro/WIN 32, 7-7  
Hardware y software recomendados,  
  STEP 7-Micro/WIN 32, 3-2  
Homologaciones, iv, A-2  
HSC3, HSC4, HSC5, SMB130 – SMB165, C-15

## I

Identificación de terminales de conexión  
  CPU 214 AC/DC/relé, A-20  
  CPU 221 AC/DC/relé, A-10  
  CPU 221 DC/DC/DC, A-10  
  CPU 222 AC/DC/relé, A-15  
  CPU 222 DC/DC/DC, A-15  
  CPU 224 DC/DC/DC, A-20  
  EM221, 8 entradas digitales x DC 24 V, A-22  
  EM222, 8 salidas digitales x DC 24 V, A-24  
  EM222, 8 salidas digitales x relé, A-24  
  EM223, 8 entradas digitales DC 24 V / 8 salidas  
  digitales de relé, A-27  
  EM223, 8 entradas digitales/8 salidas digitales,  
  A-27  
IEC 1131–3, tipificación de variables, 4-11  
IEC-1131, juego de operaciones, 4-10  
Imagen del proceso de las entradas, 4-24  
  direccionamiento, 5-4  
  funcionamiento, 4-22  
Imagen del proceso de las salidas, 4-23, 4-24  
  direccionamiento, 5-4  
Incrementar byte, 9-78  
Incrementar palabra, 9-78  
Incrementar palabra doble, 9-79  
Incrementar un puntero, 5-14  
Inhibir todos los eventos de interrupción, 9-169  
Inicialización  
  contadores rápidos, 9-41–9-44  
  función PWM, 9-59  
  modo Freeport, 9-184  
  PTO/PWM, 9-58  
Inicializar, Tren de impulsos (PTO), 9-60  
Inicializar memoria, 9-103

- Instalación  
 cable de E/S de ampliación, A-29  
 cartucho de memoria, 5-22  
 dimensiones  
   CPU 221, 2-4  
   CPU 222, 2-4  
   CPU 224, 2-5  
 módulos de ampliación, 2-5  
 raíl estándar, 2-4  
 tornillos para el montaje, 2-4–2-6  
 en entornos con vibraciones fuertes, utilizando  
 topes, 2-7  
 espacio necesario para el montaje, 2-3  
 hardware de comunicación, 3-2–3-4  
 Micro/WIN 32, 3-3  
 procedimiento  
   armario eléctrico, 2-6  
   módulo de ampliación, 2-6–2-8  
   raíl DIN, 2-7  
 tornillos para el montaje, 2-4–2-6  
 Instalación con corriente alterna, reglas, 2-13  
 Instalación con corriente continua, reglas, 2-14  
 Instalar, hardware de comunicación, instrucciones  
 especiales para usuarios de Windows NT, 7-8  
 Interface, verificar los parámetros estándar, 3-6  
 Interrupción temporizada, ejemplo, 4-18, 9-175  
 Interrupciones  
   asistencia del sistema, 9-167  
   ciclo de la CPU, 4-24  
   colas de espera, 9-172  
   configurar, 9-165  
   contadores rápidos, 9-39, 9-40  
   CPU 221/222/224, 8-7  
   datos compartidos con el programa principal ,  
   9-168  
   E/S, 9-169  
   flancos positivo y negativo, 9-169  
   habilitar e inhibir, 9-169  
   marcas para el desbordamiento de colas de  
   espera, 9-172  
   prioridades, 9-172  
   restricciones, 9-167  
   rutinas, 9-167  
   temporizadas, 9-171, C-8  
     configurar para leer entradas analógicas,  
     9-175  
   tipos y números de eventos  
     CPUs 221/222/224, 9-165  
     prioridades, 9-173  
 Interrupciones temporizadas, 9-171  
   SMB34, SMB35, C-8  
 Interruptores DIP, cable PC/PPI, 3-5, 7-38  
 Invertir byte, 9-114  
 Invertir bytes de una palabra, 9-102  
 Invertir palabra, 9-114  
 Invertir palabra doble, 9-114
- J**  
 Juegos de operaciones  
   IEC-1131, 4-10  
   SIMATIC, 4-10
- K**  
 KOP, estado del programa, 4-32
- L**  
 Lazos PID  
   acción positiva/negativa, 9-90  
   ajustar la suma integral (bias), 9-91  
   convertir salidas, 9-90  
   elegir el tipo de regulación, 9-88  
   errores, 9-93  
   márgenes, variables, 9-90  
   modos, 9-92  
   programa de ejemplo, 9-94–9-96  
   tabla del lazo, 9-93  
 Leer de la red, 9-176  
   ejemplo, 9-178–9-180  
   errores, 9-176  
 Leer reloj de tiempo real, 9-70  
 Lenguajes de programación, descripción, 4-6  
 Lista de instrucciones, 4-6  
 Llamar subrutina, con parámetros, 9-146  
 Loop control, forward/reverse, 9-90
- M**  
 Maestros  
   módem, 7-25  
   protocolo MPI, 7-4, 7-29  
   protocolo PPI, 7-29  
   protocolo PROFIBUS, 7-30  
 Manuales, nº de referencia, E-2

- Marcas de control, contadores rápidos, 9-37
- Marcas especiales, C-1–C-13
- direccionamiento, 5-5
  - SMB0: bits de estado, C-1
  - SMB1: bits de estado, C-2
  - SMB166 – 194: Tabla de definición de perfiles PTO, PT1, C-16
  - SMB186 – SMB194: control de recepción de mensajes, C-12
  - SMB2: búfer de recepción de caracteres en modo Freeport, C-2
  - SMB28 y SMB29: potenciómetros analógicos, C-6
  - SMB3: error de paridad en modo Freeport, C-2
  - SMB30 – SMB165: Registro HSC, C-15
  - SMB30, SMB130 (control del modo Freeport), 9-184
  - SMB30, SMB130: registros de control del modo Freeport, C-6
  - SMB31: control de escritura en la memoria no volátil (EEPROM), C-7
  - SMB34/SMB35: intervalos de interrupciones temporizadas, C-8
  - SMB36–SMB65: bytes de programación de los contadores rápidos, C-9
  - SMB4: desbordamiento de la cola de espera, C-3
  - SMB5: estado de las entradas y salidas, C-3
  - SMB6: identificador de la CPU, C-4
  - SMB66 – SMB85: funciones PTO/PWM, C-11
  - SMB7: reservado, C-4
  - SMB8 – SMB21: registro de errores e identificadores de los módulos de ampliación, C-5
  - SMB86 – SMB94: control de recepción de mensajes, C-12
  - SMB98 y SMB99, C-14
  - SMW222 – SMW26: tiempos de ciclo, C-6
  - SMW32: Control de escritura en la memoria no volátil (EEPROM), C-7
- Márgenes de memoria, CPU 221/222/224, 8-7
- Márgenes válidos para las CPUs, 8-7
- Memoria, borrar, 4-29
- Memoria de byte, 5-2
- Memoria V, copiar en la EEPROM, 5-20
- Mensajes, red con token passing, 7-43
- Micro/WIN 32
- convenciones de programación, 8-2
  - hardware y software recomendados, 3-2
  - instalación, 3-3
  - eliminar fallos, 3-4
- Módem
- 10 bits, 7-23
  - 11 bits, 7-25
  - adaptador de módem nulo, 7-37, 7-40
  - cables necesarios, 7-25
  - comunicación en la red, 7-25–7-30
  - conexión PC/PG a CPU, 7-25–7-26
  - utilización con el cable PC/PPI, 7-37, 7-40
- Módem nulo, adaptador de, 7-37
- Módems, configurar la comunicación, 7-16
- Modificar un puntero (direccionamiento indirecto), 5-14
- Modo Freeport
- control de interrupción de caracteres, 9-188
  - funcionamiento, 9-183
  - habilitar, 9-183
  - inicializar, 9-184
  - SMB2, búfer de recepción de caracteres, C-2
  - SMB3, error de paridad, C-2
  - SMB30, SMB130 (control del modo Freeport), 9-184
  - SMB30, SMB130: registros de control del modo Freeport, C-6
  - y modos de operación, 9-183
- Modos de operación
- bits de estado, C-1
  - cambiar, 4-25, 4-26
  - comunicación Freeport, 9-183
  - contadores rápidos, 9-33
  - y la función de forzado, 4-34
- Modos, control en lazos PID, 9-92
- Modulación del ancho de impulsos (PWM), 6-12, 9-49
- cambiar el ancho de impulsos, 9-59
  - ejemplo, 9-63
  - función, 9-50
  - inicializar, 9-59
- Módulo de ampliación, bloque de terminales, 2-12

- Módulo de ampliación analógico, direccionamiento, 6-2
- Módulo de ampliación digital, direccionamiento, 6-2
- Módulos de ampliación, 1-4, 1-5
- cable de cinta flexible, 1-5
- desmontaje, 2-8
- dimensiones
- 8 y 16 E/S, 2-5
  - CPU 221, 2-4
  - CPU 222, 2-4
  - CPU 224, 2-5
  - tornillos para el montaje, 2-4–2-6
- direccionamiento de E/S, 6-2
- identificadores y registro de errores (SMB8 a SMB21), C-5
- montaje
- armario eléctrico, 2-6
  - raíl DIN, 2-7
  - retirar la tapa del puerto de ampliación de bus, 2-6–2-8
- nº de referencia, E-1
- requisitos de alimentación, 2-18
- tornillos para el montaje, 2-4–2-6
- Montaje
- configuración, 2-2
- dimensiones
- CPU 221, 2-4
  - CPU 222, 2-4
  - CPU 224, 2-5
  - módulos de ampliación, 2-5
  - raíl estándar, 2-4
  - tornillos para el montaje, 2-4–2-6
- en entornos con vibraciones fuertes, utilización de topes, 2-7
- en sentido vertical, utilizar topes, 2-7
- espacio necesario para el montaje, 2-3
- procedimiento
- armario eléctrico, 2-6
  - módulo de ampliación, 2-6–2-8
  - raíl DIN, 2-7
  - tornillos para el montaje, 2-4–2-6
- MPI (interface multipunto), protocolo, 7-29
- velocidad de transferencia, 7-4
- MPI comunicación, 7-29
- MPI, enlaces lógicos, 7-29
- MPI, tarjeta
- ajustar los parámetros de la tarjeta MPI (PPI), 7-14
  - configuración con PC, 7-12
  - parámetros PPI, 7-14
- Multimaestro, red, 7-4
- Multiplicar enteros de 16 bits, 9-74
- Multiplicar enteros de 16 bits a enteros de 32 bits, 9-76
- Multiplicar enteros de 32 bits, 9-75
- Multiplicar reales, 9-82
- N**
- NEXT, 9-150
- Nombres simbólicos, elaborar, 4-4
- Normas, nacionales e internacionales, A-2
- Normas de emisión electromagnética, A-2
- Normas de inmunidad electromagnética, A-2
- O**
- Observar
- estado del programa, 4-32, 4-33
  - programa, 4-30–4-32
- Operación NOT, 9-4
- Operación nula, 9-8
- Operaciones
- Activar contador rápido, 9-27
  - Activar contador rápido (HSC), 9-27
  - Ajustar reloj de tiempo real, 9-70
  - Asignar, 9-6
  - Asignar directamente, 9-6
  - Asociar interrupción, 9-165
  - Borrar primer registro de la tabla, 9-108
  - Borrar temporizador de vigilancia, 9-142–9-144
  - Borrar último registro de la tabla, 9-109
  - Buscar valor en tabla, 9-106
  - Cargar pila, 9-193–9-195
  - Codificar, 9-131
  - Combinación O con bytes, 9-110
  - Combinación O con palabras, 9-111
  - Combinación O con palabras dobles, 9-112
  - Combinación O exclusiva con bytes, 9-110
  - Combinación O exclusiva con palabras, 9-111
  - Combinación O-exclusiva con palabras dobles, 9-112
  - Combinación Y con bytes, 9-110
  - Combinación Y con palabras, 9-111
  - Combinación Y con palabras dobles, 9-112

- Combinar primer y segundo valor mediante O, 9-192–9-194
- Combinar primer y segundo valor mediante Y, 9-192–9-194
- Comenzar rutina de interrupción, 9-167
- Comparar byte, 9-10
- Comparar entero, 9-11
- Comparar palabra doble, 9-12
- Comparar real, 9-13
- con contadores, 9-24
- contactos directos, 9-3
- contactos estándar, 9-2
- contadores rápidos, 9-27–9-48
- conversión, 4-16–4-18
- Convertir de ASCII a hexadecimal, 9-135
- Convertir de BCD a entero, 9-126
- Convertir de byte a entero, 9-129
- Convertir de entero a ASCII, 9-136
- Convertir de entero a BCD, 9-126
- Convertir de entero a byte, 9-129
- Convertir de entero a entero doble, 9-128
- Convertir de entero doble a ASCII, 9-138
- Convertir de entero doble a entero, 9-128
- Convertir de entero doble a real, 9-126
- Convertir de hexadecimal a ASCII, 9-135
- Convertir de real a ASCII, 9-139
- Copiar segundo valor, 9-192–9-194
- de búsqueda, 9-104–9-109
- de tabla, 9-104–9-109
- Decodificar, 9-131
- Decrementar byte, 9-78
- Decrementar palabra, 9-78
- Decrementar palabra doble, 9-79
- Definir modo para contador rápido, 9-27
- Definir modo para contador rápido (HDEF), 9-27
- Desasociar interrupción, 9-165
- Desplazar byte a la derecha, 9-116
- Desplazar byte a la izquierda, 9-116
- Desplazar palabra a la derecha, 9-117
- Desplazar palabra a la izquierda, 9-117
- Desplazar palabra doble a la derecha, 9-118
- Desplazar palabra doble a la izquierda, 9-118
- Detectar flanco negativo, 9-4
- Detectar flanco positivos, 9-4
- Dividir enteros de 16 bits, 9-74
- Dividir enteros de 16 bits a enteros de 32 bits, 9-76
- Dividir enteros de 32 bits, 9-75
- Dividir reales, 9-82
- Duplicar primer valor, 9-192–9-194
- END, 9-141
- ENO, 9-164
- Escribir en la red, 9-176
- FOR, 9-150
- Habilitar todos los eventos de interrupción, 9-169
- Incrementar byte, 9-78
- Incrementar palabra, 9-78
- Incrementar palabra doble, 9-79
- incrementar un puntero, 5-14
- Inhibir todos los eventos de interrupción, 9-169
- Inicializar memoria, 9-103
- Invertir byte, 9-114
- Invertir bytes de una palabra, 9-102
- Invertir palabra, 9-114
- Invertir palabra doble, 9-114
- Leer de la red, 9-176
- Leer reloj de tiempo real, 9-70
- modificar un puntero, 5-14
- Multiplicar enteros de 16 bits, 9-74
- Multiplicar enteros de 16 bits a enteros de 32 bits, 9-76
- Multiplicar enteros de 32 bits, 9-75
- Multiplicar reales, 9-82
- NEXT, 9-150
- NOT, 9-4
- Operación nula, 9-8
- PID, 9-84–9-98
- Poner a 0, 9-7
- Poner a 0 directamente , 9-8
- Raíz cuadrada, 9-98
- Recibir mensaje, 9-182
- Redondear a entero doble, 9-127
- Registrar valor en tabla, 9-104
- Registro de desplazamiento, 9-123
- Registro de desplazamiento (SHRB), 9-124
- Regulación PID, 9-84–9-98
- Relé de control secuencial, 9-153
- reloj de tiempo real, 9-70
- Restar enteros de 32 bits, 9-73
- Restar reales, 9-81
- Retorno de subrutina, 9-145
- Retorno desde rutina de interrupción, 9-167
- Rotar byte a la derecha, 9-119
- Rotar byte a la izquierda, 9-119
- Rotar palabra a la derecha, 9-120
- Rotar palabra a la izquierda, 9-120
- Rotar palabra doble a la derecha, 9-121
- Rotar palabra doble a la izquierda, 9-121
- Sacar primer valor, 9-193–9-195
- Salida de impulsos, 6-12, 9-49–9-69
- Salida de impulsos (PLS), 6-12
- Saltar a meta, 9-144
- Segmento, 9-133
- sobrecargadas, 4-15
- STOP, 9-141
- Sumar enteros de 16 bits, 9-72
- Sumar enteros de 32 bits, 9-73
- Sumar reales, 9-81
- Temporizador de retardo a la conexión, 9-15
- Temporizador de retardo a la conexión memo-

- rizado, 9-15  
 Temporizador de retardo a la desconexión, 9-15  
 tiempo de ejecución, F-1  
 Transferir byte, 9-99  
 Transferir bytes en bloque, 9-100  
 Transferir palabra, 9-99  
 Transferir palabra doble, 9-99  
 Transferir palabras dobles en bloque, 9-100  
 Transferir palabras en bloque, 9-100  
 Transferir real, 9-99  
 Transmitir mensaje, 9-182  
 Truncar, 9-127
- Operaciones aritméticas  
 Dividir enteros de 16 bits, 9-74  
 Dividir enteros de 16 bits a enteros de 32 bits, 9-76  
 Dividir enteros de 32 bits, 9-75  
 Dividir reales, 9-82  
 ejemplo, 9-77, 9-83  
 Multiplicar enteros de 16 bits, 9-74  
 Multiplicar enteros de 16 bits a enteros de 32 bits, 9-76  
 Multiplicar enteros de 32 bits, 9-75  
 Multiplicar reales, 9-82  
 Raíz cuadrada, 9-98  
 Restar enteros de 16 bits, 9-72  
 Restar enteros de 32 bits, 9-73  
 Restar reales, 9-81  
 Sumar enteros de 16 bits, 9-72  
 Sumar enteros de 32 bits, 9-73  
 Sumar reales, 9-81
- Operaciones AWL  
 breviarío, G-5  
 tiempo de ejecución, F-1
- Operaciones con contactos  
 ejemplo, 9-5  
 NOT, 9-4
- Operaciones con contadores rápidos, 9-27–9-48  
 Activar contador rápido, 9-27  
 Definir modo para contador rápido, 9-27
- Operaciones con salidas  
 Asignar, 9-6  
 Asignar directamente, 9-6  
 ejemplo, 9-9  
 Operación nula, 9-8  
 Poner a 0, 9-7  
 Poner a 0 directamente, 9-8
- Operaciones de búsqueda, 9-104–9-109  
 Borrar primer registro de la tabla, 9-108  
 Borrar último registro de la tabla, 9-109  
 Buscar valor en tabla, 9-106  
 Registrar valor en tabla, 9-104
- Operaciones de comparación  
 Comparar byte, 9-10  
 Comparar entero, 9-11  
 Comparar palabra doble, 9-12  
 Comparar real, 9-13  
 ejemplo, 9-14
- Operaciones de comunicación  
 Escribir en la red, 9-176  
 Leer de la red, 9-176  
 Recibir mensaje, 9-182  
 Transmitir mensaje, 9-182
- Operaciones de contaje, 9-24  
 ejemplo, 9-25
- Operaciones de control del programa  
 Borrar temporizador de vigilancia, 9-142–9-144  
 ejemplo, 9-143–9-145  
 END, 9-141  
 ejemplo, 9-143–9-145  
 ENO, 9-164  
 FOR, 9-150  
 FOR/NEXT, ejemplo, 9-152–9-154  
 Llamar subrutina, ejemplo, 9-149–9-151  
 NEXT, 9-150  
 Relé de control secuencial, 9-153  
 Retorno de subrutina, 9-145  
 Saltar a meta, 9-144  
 ejemplo, 9-144–9-146  
 STOP, 9-141  
 ejemplo, 9-143–9-145
- Operaciones de conversión  
 Codificar, 9-131  
 Convertir de ASCII a hexadecimal, 9-135  
 Convertir de BCD a entero, 9-126  
 Convertir de byte a entero, 9-129  
 Convertir de entero a ASCII, 9-136  
 Convertir de entero a BCD, 9-126  
 Convertir de entero a byte, 9-129  
 Convertir de entero a entero doble, 9-128  
 Convertir de entero doble a ASCII, 9-138  
 Convertir de entero doble a entero, 9-128  
 Convertir de entero doble a real, 9-126  
 Convertir de hexadecimal a ASCII, 9-135  
 Convertir de real a ASCII, 9-139  
 Decodificar, 9-131  
 Redondear a entero doble, 9-127  
 Segmento, 9-133  
 Truncar, 9-127
- Operaciones de decrementar  
 Decrementar byte, 9-78  
 Decrementar palabra, 9-78  
 Decrementar palabra doble, 9-79  
 Restar enteros de 16 bits, 9-72  
 Restar enteros de 32 bits, 9-73

- Operaciones de desplazamiento
    - Desplazar byte a la derecha, 9-116
    - Desplazar byte a la izquierda, 9-116
    - Desplazar palabra a la derecha, 9-117
    - Desplazar palabra a la izquierda, 9-117
    - Desplazar palabra doble a la derecha, 9-118
    - Desplazar palabra doble a la izquierda, 9-118
    - ejemplo de desplazamiento y rotación, 9-122–9-124
    - ejemplo de registro de desplazamiento, 9-125–9-127
    - Registro de desplazamiento, 9-123
  - Operaciones de incrementar
    - Incrementar byte, 9-78
    - Incrementar palabra, 9-78
    - Incrementar palabra doble, 9-79
    - Sumar enteros de 16 bits, 9-72
    - Sumar enteros de 32 bits, 9-73
  - Operaciones de inicialización, ejemplo, 9-103–9-105
  - Operaciones de interrupción
    - Asociar interrupción, 9-165
    - Comenzar rutina de interrupción, 9-167
    - Desasociar interrupción, 9-165
    - descripción, 9-165
    - ejemplo, 9-174
    - Habilitar todos los eventos de interrupción, 9-169
    - Inhibir todos los eventos de interrupción, 9-169
    - Retorno desde rutina de interrupción, 9-167
  - Operaciones de rotación
    - ejemplo de desplazamiento y rotación, 9-122–9-124
    - Rotar byte a la derecha, 9-119
    - Rotar byte a la izquierda, 9-119
    - Rotar palabra a la derecha, 9-120
    - Rotar palabra a la izquierda, 9-120
    - Rotar palabra doble a la izquierda, 9-121
  - Operaciones de segmentación (operaciones SCR), 9-154
  - Operaciones de tabla, 9-104–9-109
    - Borrar primer registro de la tabla, 9-108
    - Borrar último registro de la tabla, 9-109
    - Buscar valor en tabla, 9-106
    - Inicializar memoria, 9-103
    - Registrar valor en tabla, 9-104
  - Operaciones de temporización
    - ejemplo de un temporizador de retardo a la conexión, 9-20, 9-21
    - ejemplo de un temporizador de retardo a la desconexión, 9-22
    - Temporizador de retardo a la conexión, 9-15
    - Temporizador de retardo a la conexión memorizado, 9-15
    - Temporizador de retardo a la desconexión, 9-15
  - Operaciones de transferencia
    - ejemplo de transferir e invertir, 9-102–9-104
    - ejemplo de una operación de transferencia de bloques, 9-101–9-103
    - Invertir bytes de una palabra, 9-102
    - Transferir byte, 9-99
    - Transferir bytes en bloque, 9-100
    - Transferir palabra, 9-99
    - Transferir palabra doble, 9-99
    - Transferir palabras dobles en bloque, 9-100
    - Transferir palabras en bloque, 9-100
    - Transferir real, 9-99
  - Operaciones del relé de control secuencial, ejemplos, 9-155–9-159
  - Operaciones del reloj de tiempo real, 9-70
    - Ajustar reloj de tiempo real, 9-70
    - Leer reloj de tiempo real, 9-70
  - Operaciones incrementar, ejemplo, 9-80
  - Operaciones lógicas
    - Combinación O con bytes, 9-110
    - Combinación O con palabras, 9-111
    - Combinación O con palabras dobles, 9-112
    - Combinación O exclusiva con bytes, 9-110
    - Combinación O exclusiva con palabras, 9-111
    - Combinación O-exclusiva con palabras dobles, 9-112
    - Combinación Y con bytes, 9-110
    - Combinación Y con palabras, 9-111
    - Combinación Y con palabras dobles, 9-112
    - ejemplo
      - Invertir, 9-115–9-117
      - Y, O y O-exclusiva, 9-113–9-115
    - Invertir byte, 9-114
    - Invertir palabra, 9-114
    - Invertir palabra doble, 9-114
  - Operaciones lógicas con contactos, ejemplo, 9-5
  - Operaciones lógicas de pilas
    - Cargar pila, 9-193–9-195
    - Combinar primer y segundo valor mediante O, 9-192–9-194
    - Combinar primer y segundo valor mediante Y, 9-192–9-194
    - Copiar segundo valor, 9-192–9-194
    - Duplicar primer valor, 9-192–9-194
    - ejemplo, 9-195–9-197
    - funcionamiento, 9-193
    - Sacar primer valor, 9-193–9-195
  - Operaciones PID, ejemplo, 9-94–9-96
- P**
- Palabras, y margen de enteros, 5-4
  - Palabras dobles, y margen de enteros, 5-4
  - Parametrización, cable PC/PPI (PPI), 7-10–7-11

- Parametrización utilizada  
seleccionar, 7-9–7-10  
tarjeta MPI (PPI), 7-14
- Parámetros  
cable PC/PPI (PPI), 7-10–7-11  
tarjeta MPI (PPI), 7-14
- PID, lazos, CPU 221/222/224, 8-7
- PID, operaciones, 9-84–9-98
- Pila lógica, relés de control secuencial (SCRs), 9-153
- Polarizar, redes, 7-32
- Poner a 0, 9-7
- Poner a 0 directamente, 9-8
- Potenciómetros, SMB28 y SMB29, 6-13
- Potenciómetros analógicos, 6-13  
SMB28, SMB29, C-6
- Potenciómetros analógicos SMB28 y SMB29, 6-13
- Potenciómetros analógicos: SMB28 y SMB29, C-6
- PPI (interface punto a punto)  
comunicación, 7-2, 7-29  
protocolo, 7-29
- Procesador de comunicaciones (CP), n° de referencia, E-1
- Process Field Bus, iv
- PROFIBUS  
comunicación, 7-30  
datos técnicos del cable, 7-33  
protocolo, 7-30  
repetidores, 7-34
- PROFIBUS, asignación de pines, 7-31
- Programa  
almacenamiento, 5-15–5-18, 5-22  
cargar en la CPU, 5-15  
cargar en la PG, 5-15  
comprobar, 4-30–4-32  
ejecutar, 4-23  
elementos básicos, 4-18  
entradas analógicas, 4-22  
entradas/salidas (E/S), 4-5  
estructura, 4-18  
guardar datos en la memoria no volátil, 5-20  
observar, 4-30–4-32  
observar el estado, 4-32, 4-33  
restablecer con el cartucho de memoria, 5-24  
utilizar subrutinas, 9-145  
utilizar una tabla de estado/forzado, 4-31
- Protocolos definidos por el usuario, Freeport, 7-30
- PTO/PWM  
ancho y conteo de impulsos, 9-57  
marca especial de estado, 9-57  
marcas especiales de control, 9-57  
tiempo de ciclo, 9-57
- PTO/PWM, funciones  
calcular los valores de la tabla de perfiles, 9-54  
registros de control, 9-56
- PTO/PWM, tabla de referencia hexadecimal, 9-56
- Puerto de comunicación  
asignación de pines, 7-31  
interrupciones, 9-169
- Puesta a tierra para circuitos, reglas de cableado, 2-10
- Punteros, 5-13–5-15  
modificar un puntero, 5-14
- ## R
- Raíl  
dimensiones, 2-4  
entornos con vibraciones fuertes, 2-7  
espacio necesario para el montaje, 2-3–2-5  
instalación, 2-7  
montaje vertical, 2-7  
utilización de topes, 2-7
- Raíl DIN  
desmontaje, 2-8  
dimensiones, 2-4  
entornos con vibraciones fuertes, 2-7  
espacio necesario para el montaje, 2-3–2-5  
instalación, 2-7  
montaje vertical, 2-7  
n° de referencia, E-2  
utilización de topes, 2-7
- Raíl estándar  
dimensiones, 2-4  
entornos con vibraciones fuertes, 2-7  
espacio necesario para el montaje, 2-3–2-5  
instalación, 2-7  
montaje vertical, 2-7  
utilización de topes, 2-7
- Raíz cuadrada, 9-98
- Rearrancar la CPU, después de un error fatal, 4-37
- Recibir mensaje, 9-182, 9-185  
SMB86 – SMB94, SMB186 – SMB194, C-12

- Red
  - componentes, 7-31
  - multimaestro, 7-4
  - puerto de comunicación, 7-31
- Redes
  - cerrar, 7-32
  - conectores, 7-32
  - configurar la comunicación, 7-2–7-19
  - datos técnicos del cable, 7-33
  - dirección de estación más alta, 7-41
  - enviar mensajes, 7-43
  - factor de actualización GAP, 7-41
  - instalar el hardware de comunicación, 3-2–3-4
  - optimizar el rendimiento, 7-41
  - polarizar, 7-32
  - repetidores, 7-34
  - segmentos, 7-28
  - seleccionar la parametrización utilizada, 7-9
  - tiempo de rotación del testigo (token), 7-44–7-47
- Redes de resistencia/condensador, aplicaciones de relé, 2-17
- Redondear a entero doble, 9-127
- Registrar valor en tabla, 9-104
- Registro de desplazamiento, 9-123, 9-124
- Registro de desplazamiento (SHRB), 9-124
- Registro de errores e identificadores de los módulos de ampliación (SMB8–SMB21), C-5
- Reglas
  - cableado, 2-9
    - aislamiento, 2-10
  - circuitos de supresión, 2-16
    - salidas AC, 2-17
    - transistores en DC, 2-17
  - diseñar una solución de automatización, 4-2–4-4
  - entornos con vibraciones fuertes, 2-7
  - instalación con corriente alterna, 2-13
  - instalación con corriente continua, 2-14
  - modificar un puntero para el direccionamiento indirecto, 5-14
  - montaje vertical, 2-7
  - puesta a tierra para circuitos, 2-10
  - utilización de topes, 2-7
- Reglas de cableado para fases unipolares, 2-13
- Reglas para instalaciones aisladas con corriente continua, 2-14
- Regulación del lazo, convertir entradas, 9-89
- Regulación PID, 9-84–9-98
  - acción positiva/negativa, 9-90
  - ajustar suma integral (bias), 9-91
  - bits de historial, 9-92
  - convertir entradas, 9-89
  - convertir salidas, 9-90
  - ejemplo, 9-94–9-96
  - elegir el tipo, 9-88
  - errores, 9-93
  - márgenes/variables, 9-90
  - programa de ejemplo, 9-94–9-96
  - tabla del lazo, 9-93
- Regulación PID Loop, modos, 9-92
- Relé corriente continua, 2-17
- Relé de control secuencial, 9-153
- Relés, redes de resistencia/condensador, 2-17
- Relés de control secuencial
  - CPU 221/222/224, 8-7
  - direccionamiento, 5-5
- Reloj, bits de estado, C-1
- Reloj, tiempo real, 9-70
- Repetidores
  - nº de referencia, E-2
  - red PROFIBUS, 7-34
- Requisitos de alimentación
  - calcular, 2-18
  - CPU, 2-18
  - ejemplo, 2-19
  - módulos de ampliación, 2-18
- Respaldar datos, 5-15–5-20
  - áreas remanentes, 5-19
  - cartucho de pila (opcional), 5-15
  - condensador de alto rendimiento, 5-15
  - conectar la alimentación, 5-17–5-21
  - EEPROM, 5-15, 5-17
  - en la EEPROM, 5-20
- Respaldar la memoria, 5-15–5-20
- Restar enteros de 16 bits, 9-72
- Restar enteros de 32 bits, 9-73
- Restar reales, 9-81
- Restringir el acceso a la CPU. Véase Password
- Resumen de las CPUs S7-200
  - áreas de operandos, 8-8
  - márgenes de memoria, 8-7
- Resumen de las CPUs S7-200, funciones, 1-3
- Retirar, cartucho de memoria, 5-22
- Retorno de subrutina, 9-145
- Retorno desde rutina de interrupción, 9-167

- Rotación del testigo, rendimiento de la red, 7-42  
 Rotar byte a la izquierda, 9-119  
 Rotar palabra a la derecha, 9-120  
 Rotar palabra a la izquierda, 9-120  
 Rotar palabra doble a la derecha, 9-121  
 Rotar palabra doble a la izquierda, 9-121  
 RUN, 4-25  
 Rutinas de interrupción, reglas, 4-18
- S**
- S7-200  
 compatibilidad electromagnética, A-4  
 componentes, 1-4  
 Componentes de sistema, 1-2  
 condiciones ambientales, A-3  
 CPU, desmontaje, 2-8  
 datos técnicos, A-3  
 dimensiones  
     CPU 221, 2-4  
     CPU 222, 2-4  
     CPU 224, 2-5  
     módulos de ampliación, 2-5  
     tornillos para el montaje, 2-4–2-6  
 módulos de ampliación, 1-4  
     desmontaje, 2-8  
 montaje, armario eléctrico, 2-6  
 resumen de las CPUs, 1-3  
 tornillos para el montaje, 2-4–2-6  
 Sacar primer valor, 9-193–9-195  
 Salida de impulsos, 6-10, 9-49  
     cambiar el ancho de impulsos, 6-12  
     funciones PTO/PWM, SMB66 – SMB85: marcas especiales, C-11  
     operación, 9-49  
 Salida de impulsos (PLS), 6-12  
 Salidas  
     congelar, 6-8  
     de impulsos rápidos, 6-12  
     funcionamiento básico, 4-5  
 Salidas AC, 2-17  
 Salidas analógicas  
     acceder a, 4-23  
     direccionamiento, 5-9  
 Salidas de impulsos, 6-12  
 Salidas digitales, escribir, 4-23  
 Saltar a meta, 9-144  
 Segmento  
     direcciones de los equipos, 7-28  
     unidades esclavas, 7-28  
     unidades maestras, 7-28  
 Segmento (operación de conversión), 9-133  
 Segmentos, red, 7-28  
 Selector, de modos de operación, 4-25  
 SIMATIC, juego de operaciones, 4-10  
 SM0.2 (marca Datos remanentes perdidos), 5-18  
 SMB0: bits de estado, C-1  
 SMB1: bits de estado, C-2  
 SMB166 – SMB194: Tabla de definición de perfiles PTO, PT1, C-16  
 SMB186 – SMB194: control de recepción de mensajes, C-12  
 SMB2 (recepción de caracteres en modo Freeport), control de interrupción de caracteres, 9-188  
 SMB2: búfer de recepción de caracteres en modo Freeport, C-2  
 SMB3 (error de paridad en modo Freeport), control de interrupción de caracteres, 9-188  
 SMB3: error de paridad en modo Freeport, C-2  
 SMB30 – SMB165: Registro HSC, C-15  
 SMB30, SMB130 (control del modo Freeport) , 9-184  
 SMB30, SMB130: registros de control del modo Freeport, C-6  
 SMB34/SMB35: intervalos de interrupciones temporizadas, C-8  
 SMB36–SMB65: bytes de programación de los contadores rápidos, C-9  
 SMB5: estado de las entradas y salidas, C-3  
 SMB6: identificador de la CPU, C-4  
 SMB7 (reservado), C-4  
 SMB86 – SMB94: control de recepción de mensajes, C-12  
 SMB98 y SMB99, C-14  
 SMW22 – SMW26: tiempos de ciclo, C-6  
 Sobrecarga, 4-15  
 Software de programación, nº de referencia, E-1  
 Soltar, bloque de terminales, 2-12  
 STEP 7-Micro/WIN 32  
     ayuda en pantalla, 3-2  
     comunicación con módems, 7-25–7-30  
     configurar la comunicación, 7-5  
     hardware para la comunicación en redes, 3-2, 7-3  
     hardware y software recomendados, 3-2  
     instalar el hardware de comunicación, 3-2–3-4  
     nº de referencia, E-1  
     nº de referencia, actualización, E-1  
 STOP, 4-25, 9-141  
 Subrutina  
     con parámetros, 9-146  
     ejemplo, 4-18  
 Subrutinas  
     agregar al programa, 9-145  
     reglas, 4-18  
 Suma integral (bias), ajustar, regulación PID, 9-91  
 Sumar enteros de 16 bits, 9-72  
 Sumar enteros de 32 bits, 9-73  
 Sumar reales, 9-81

Supresión de ruidos, filtro de entrada, 6-4

## T

Tabla de definición de perfiles PTO, PT1, SMB166 – SMB194, C-16  
 Tabla de estado/forzado ciclo de la CPU, 4-34  
 modificar el programa, 4-31  
 Tabla de referencia hexadecimal PTO/PWM, 9-56  
 Tabla del lazo, 9-93  
 Tabla del lazo PID, 9-93  
 Tamaño, módulos de ampliación, 2-5  
 Tamaño de los módulos  
 CPU 221, 2-4  
 CPU 222, 2-4  
 CPU 224, 2-5  
 tornillos para el montaje, 2-4–2-6  
 Tamaño físico  
 CPU 221, 2-4  
 CPU 222, 2-4  
 CPU 224, 2-5  
 módulos de ampliación, 2-5  
 tornillos para el montaje, 2-4–2-6  
 Tarjeta MPI, 7-4  
 TD200, n° de referencia, E-2  
 Temporizador de retardo a la conexión, 9-15  
 Temporizador de retardo a la conexión memorizado, 9-15  
 Temporizador de retardo a la desconexión, 9-15  
 Temporizadores  
 CPU 221/222/224, 8-7  
 direccionamiento, 5-7  
 Temporizadores T32/T96, interrupciones, 9-171  
 TERM, 4-25  
 Término diferencial, algoritmo PID, 9-88  
 Término integral, algoritmo PID, 9-87  
 Término proporcional, algoritmo PID, 9-87  
 Test de aislamiento de alto potencial, A-4  
 Tiempo de ciclo, función Tren de impulsos (PTO), 9-60  
 Tiempo de ciclo, SMW22 – SMW26), C-6  
 Tiempo de ejecución, operaciones AWL, F-1  
 Tiempo de rotación del testigo (token), 7-44–7-47  
 Tiempo de rotación del testigo, número de estaciones, 7-45  
 Tiempo, ajustar, 9-70  
 Tipos de datos  
 complejos, 4-12  
 elementales, 4-11  
 verificación, 4-12–4-16  
 ventajas, 4-14  
 Tornillos (para el montaje), 2-4–2-6  
 Transferir byte, 9-99

Transferir bytes en bloque, 9-100  
 Transferir palabra doble, 9-99  
 Transferir palabras dobles en bloque, 9-100  
 Transferir palabras en bloque, 9-100  
 Transferir real, 9-99  
 Transistores en DC, proteger, 2-16  
 Transmitir mensaje, 9-182, 9-184  
 ejemplo, 9-189  
 Tratamiento de errores, fatales, 4-37  
 Tren de impulsos (PTO), 6-12  
 cambiar el conteo de impulsos, 9-61  
 cambiar el tiempo de ciclo, 9-60  
 cambiar el tiempo de ciclo y el conteo de impulsos, 9-61  
 ejemplo, 9-65, 9-68  
 función, 9-51  
 inicializar, 9-60  
 Trenes de impulsos rápidos (PTO), 9-49  
 Truncar, 9-127

## U

Utilizar punteros, 5-13  
 & y \*, 5-13  
 modificar un puntero, 5-14  
 Utilizar subrutinas, 9-145

## V

Valores actuales de los temporizadores, actualizar, 9-19  
 Valores en coma flotante  
 regulación PID, 9-89  
 representación de, 5-4  
 Variable del proceso, convertir, 9-89  
 Variables, forzar, 4-34  
 VDE 0160, A-2  
 Velocidad de transferencia  
 ajustar los interruptores del cable PC/PPI, 3-5, 7-38  
 posición de los interruptores DIP en el cable PC/PPI, 7-35, A-30  
 Vibraciones fuertes, utilizar topes, 2-7  
 Vida útil de los relé, A-5  
 Visualizador de textos TD 200, manual del usuario, iv  
 Visualizadores de textos y paneles de operador, n° de referencia, E-2

## W

Windows NT, instalar hardware, 7-8

Siemens AG  
A&D AS E 81

Oestliche Rheinbrueckenstr. 50  
D-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    |   |

