

# Habilitação técnica em Elettrônica



6

## Automação Industrial

CENTRO PAULA SOUZA

GOVERNO DE  
SÃO PAULO

**CENTRO PAULA SOUZA DO GOVERNO DE SÃO PAULO**



# Eletrônica

Volume 6

# Eletrônica

## Automação industrial

Ismael Moura Parede  
Luiz Eduardo Lemes Gomes  
(autores)

Edson Horta  
(coautor)



2011



#### Presidência

João Sayad

#### Vice-presidência

Ronaldo Bianchi, Fernando Vieira de Mello

#### DIRETORIA DE PROJETOS EDUCACIONAIS

**Direção:** Fernando José de Almeida

**Gerência:** Monica Gardelli Franco, Júlio Moreno

**Coordenação Técnica:** Maria Luiza Guedes

**Equipe de autoria Centro Paula Souza**

**Coordenação geral:** Ivone Marchi Lainetti Ramos

**Coordenação da série Eletrônica:** Jun Suzuki

**Autores:** Ismael Moura Parede, Luiz Eduardo Lemes Gomes

**Coautor:** Edson Horta

**Revisão técnica:** Luiz Carlos da Cunha e Silva

**Equipe de Edição**

**Coordenação geral:** Carlos Tabosa Seabra, Rogério Eduardo Alves

**Coordenação editorial:** Luiz Marin

**Edição de texto:** Roberto Matajs

**Secretário editorial:** Antonio Mello

**Revisão:** Conexão Editorial

**Direção de arte:** Bbox Design

**Diagramação:** LCT Tecnologia

**Ilustrações:** Adilson Secco

**Pesquisa iconográfica:** Completo Iconografia

**Capa**

**Fotografia:** Eduardo Pozella, Carlos Piratininga

**Tratamento de imagens:** Sidnei Testa

**Abertura capítulos:** © Lize Streeter/Dorling Kindersley/ Getty Images

O Projeto Manual Técnico Centro Paula Souza – Coleção Técnica Interativa oferece aos alunos da instituição conteúdo relevante à formação técnica, à educação e à cultura nacional, sendo também sua finalidade a preservação e a divulgação desse conteúdo, respeitados os direitos de terceiros.

O material apresentado é de autoria de professores do Centro Paula Souza e resulta de experiência na docência e da pesquisa em fontes como livros, artigos, jornais, internet, bancos de dados, entre outras, com a devida autorização dos detentores dos direitos desses materiais ou contando com a permissibilidade legal, apresentando, sempre que possível, a indicação da autoria/crédito e/ou reserva de direitos de cada um deles.

Todas as obras e imagens expostas nesse trabalho são protegidas pela legislação brasileira e não podem ser reproduzidas ou utilizadas por terceiros, por qualquer meio ou processo, sem expressa autorização de seus titulares.

Agradecemos as pessoas retratadas ou que tiveram trechos de obras reproduzidas neste trabalho, bem como a seus herdeiros e representantes legais, pela colaboração e compreensão da finalidade desse projeto, contribuindo para que essa iniciativa se tornasse realidade. Adicionalmente, colocamo-nos à disposição e solicitamos a comunicação, para a devida correção, de quaisquer equívocos nessa área porventura cometidos em livros desse projeto.

O Projeto Manual Técnico Centro Paula Souza – Coleção Técnica Interativa, uma iniciativa do Governo do Estado de São Paulo, resulta de um esforço colaborativo que envolve diversas frentes de trabalho coordenadas pelo Centro Paula Souza e é editado pela Fundação Padre Anchieta. A responsabilidade pelos conteúdos de cada um dos trabalhos/textos inseridos nesse projeto é exclusiva do autor. Respeitam-se assim os diferentes enfoques, pontos de vista e ideologias, bem como o conhecimento técnico de cada colaborador, de forma que o conteúdo exposto pode não refletir as posições do Centro Paula Souza e da Fundação Padre Anchieta.

#### Dados Internacionais de Catalogação na Publicação (CIP) (Bibliotecária Silvia Marques CRB 8/7377)

P227

Parede, Ismael Moura

Eletrônica: automação industrial / Ismael Moura Parede, Luiz Eduardo Lemes Gomes (autores); Edson Horta (coautor), Luiz Carlos da Cunha e Silva (revisor); Jun Suzuki (coordenador). -- São Paulo: Fundação Padre Anchieta, 2011 (Coleção Técnica Interativa. Série Eletrônica, v. 6)

Manual técnico Centro Paula Souza

ISBN 978-85-8028-050-0

1. Eletrônica 2. Automação industrial 3. Sistemas automatizados  
I. Gomes, Luiz Eduardo Lemes II. Horta, Edson III. Silva, Luiz Carlos da Cunha e IV. Suzuki, Jun V. Título

CDD 607



#### GOVERNADOR

Geraldo Alckmin

#### VICE-GOVERNADOR

Guilherme Afif Domingos

#### SECRETÁRIO DE DESENVOLVIMENTO ECONÔMICO, CIÊNCIA E TECNOLOGIA

Paulo Alexandre Barbosa



#### Presidente do Conselho Deliberativo

Yolanda Silvestre

#### Diretora Superintendente

Laura Laganá

#### Vice-Diretor Superintendente

César Silva

#### Chefe de Gabinete da Superintendência

Elenice Belmonte R. de Castro

#### Coordenadora da Pós-Graduação, Extensão e Pesquisa

Helena Gemignani Peterossi

#### Coordenador do Ensino Superior de Graduação

Angelo Luiz Cortelazzo

#### Coordenador de Ensino Médio e Técnico

Almério Melquíades de Araújo

#### Coordenadora de Formação Inicial e Educação Continuada

Clara Maria de Souza Magalhães

#### Coordenador de Desenvolvimento e Planejamento

João Carlos Paschoal Freitas

#### Coordenador de Infraestrutura

Rubens Goldman

#### Coordenador de Gestão Administrativa e Financeira

Armando Natal Maurício

#### Coordenador de Recursos Humanos

Elio Lourenço Bolzani

#### Assessora de Comunicação

Gleise Santa Clara

#### Procurador Jurídico Chefe

Benedito Libério Bergamo

# Sumário

<b>13</b>	<b>Capítulo 1</b>	
	<b>Introdução à automação industrial</b>	
	1.1 Evolução	15
	1.2 Controlador lógico programável	18
	1.2.1 Unidade central de processamento (UCP)	19
	1.2.2 Memória	21
	1.2.3 Dispositivos de entrada e saída (E/S)	22
	1.2.4 Terminal de programação	23
	1.2.5 Periféricos de interface com o usuário	25

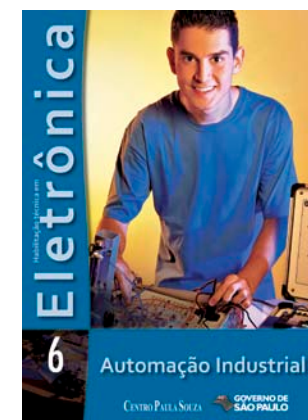
<b>27</b>	<b>Capítulo 2</b>	
	<b>Tipos de automação industrial</b>	
	2.1 Comandos pneumáticos	28
	2.1.1 Válvulas de controle direcional	31
	2.1.2 Atuadores	34
	2.1.3 Outros dispositivos	35
	2.2 Comandos elétricos e eletropneumáticos com CLP	37
	2.3 CLP versus computador	39
	2.4 Sistema digital de controle distribuído (SDCD)	42
	2.5 Ambiente industrial (IP ou NEMA)	43
	2.6 Relação custo-benefício	45

<b>47</b>	<b>Capítulo 3</b>	
	<b>CLP: estrutura e funcionamento</b>	
	3.1 Unidade central de processamento (UCP)	48
	3.2 Bloco de memória	49

3.3 Módulos de entrada e saída (E/S)	50
3.3.1 Módulos de E/S digitais	51
3.3.2 Módulos de E/S analógicas	58
3.3.3 Módulos de E/S inteligentes	60
3.4 Fontes de alimentação	61
3.5 Funcionamento	62

<b>65</b>	<b>Capítulo 4</b>	
	<b>Linguagem de programação</b>	
	4.1 Norma IEC 61131	66
	4.2 IL – lista de instruções	69
	4.3 ST – texto estruturado	71
	4.4 FBD – diagrama de blocos funcionais	71
	4.5 LD – diagrama Ladder	74
	4.6 SFC – sequenciamento gráfico de funções	74

<b>77</b>	<b>Capítulo 5</b>	
	<b>A programação Ladder</b>	
	5.1 Diretivas básicas	78
	5.2 Ligação das entradas e saídas digitais do CLP genérico	80
	5.2.1 Entradas digitais	80
	5.2.2 Entrada rápida	81
	5.2.3 Saídas digitais	81
	5.2.4 Saídas a relé	82
	5.3 Contatos NA/NF	83
	5.3.1 Contato NA	83
	5.3.2 Contato NF	83



**Capa:** Sérgio Martines Jr., aluno do Centro Paula Souza  
**Foto:** Eduardo Pozella e Carlos Piratininga



OKNOART / SHUTTERSTOCK

# Sumário

5.4 Saída simples/saída complementar . . . . .	84
5.4.1 Saída simples . . . . .	84
5.4.2 Saída complementar . . . . .	85
5.5 Contato por borda positiva . . . . .	90
5.6 Contato por borda negativa. . . . .	91
5.7 Saída set e saída reset . . . . .	92
5.8 Memória ou <i>flag</i> . . . . .	93
5.9 Contadores. . . . .	94
5.9.1 Contador UP. . . . .	94
5.9.2 Contador DOWN . . . . .	97
5.9.3 Contador rápido. . . . .	100
5.10 Temporizadores . . . . .	102
5.10.1 Temporizador na energização (TON) .	103
5.10.2 Temporizador na desenergização (TOFF). . . . .	104
5.10.3 Temporizador de pulso (TP). . . . .	105
5.11 Entradas e saídas analógicas: endereçamento .	107
5.12 Instruções de comparação . . . . .	108
5.12.1 Igual a (=) . . . . .	108
5.12.2 Maior que (>). . . . .	109
5.12.3 Menor que (<) . . . . .	109
5.12.4 Maior ou igual a ( $\geq$ ) . . . . .	110
5.12.5 Menor ou igual a ( $\leq$ ) . . . . .	110
5.12.6 Diferente de ( $\neq$ ). . . . .	111
5.13 Operações matemáticas . . . . .	111
5.13.1 Somador (+). . . . .	111
5.13.2 Subtrator (-) . . . . .	112
5.13.3 Multiplicador ( $\cdot$ ) . . . . .	112
5.13.4 Divisor ( $\div$ ) . . . . .	113

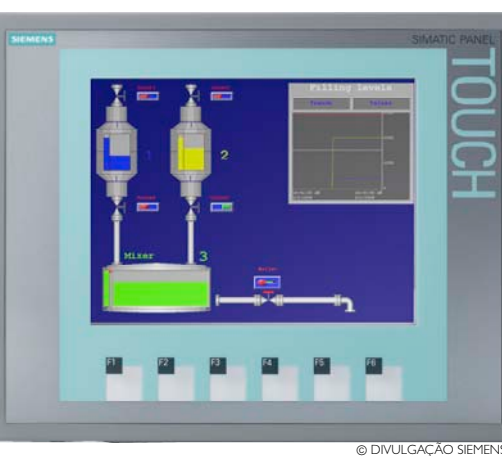
5.14 Funções especiais . . . . .	114
5.14.1 CALL . . . . .	114
5.14.2 JUMP . . . . .	114
5.14.3 MOVE . . . . .	115

## 117 Capítulo 6 Projeto de automação utilizando o CLP

6.1 Definição dos pontos de entrada e saída. . . . .	118
6.2 Descritivo de funcionamento. . . . .	118
6.3 Elaboração de programas. . . . .	119
6.3.1 Ligar, manter ligado e desligar . . . . .	119
6.3.2 Passos e transições. . . . .	126
6.4 Testes, simulações e alterações . . . . .	133
6.5 Liberação de projeto . . . . .	134
6.6 Projetos para estudo . . . . .	134
6.6.1 Motor trifásico com reversão . . . . .	134
6.6.2 Carro transportador . . . . .	136
6.6.3 Semáforos. . . . .	138
6.6.4 Transportadora versão 2 . . . . .	142

## 151 Capítulo 7 Interface Homem-Máquina

7.1 Tipos de IHM . . . . .	153
7.2 Configuração e programação da IHM . . . . .	155
7.2.1 Navegação. . . . .	155
7.2.2 Evento . . . . .	155
7.2.3 Alarme . . . . .	157
7.3 Considerações sobre a aplicação de IHMs . . . . .	158
7.4 Etapas de um projeto com CLP e IHM . . . . .	158



# Sumário

## 161 Capítulo 8

### Comunicação em sistemas automatizados

- 8.1 Redes de comunicação . . . . . 165
- 8.2 Protocolos de comunicação. . . . . 170
  - 8.2.1 Protocolo Modbus . . . . . 171
  - 8.2.2 Protocolo Profibus. . . . . 174
  - 8.2.3 Protocolo CAN . . . . . 176
  - 8.2.4 Protocolo Devicenet . . . . . 178
  - 8.2.5 Protocolo Hart. . . . . 180
- 8.3 Wireless na automação . . . . . 185
- 8.4 Sistemas multimestre . . . . . 187
- 8.5 Topologias utilizadas na indústria. . . . . 190

## 195 Referências bibliográficas

# Capítulo I

## Introdução à automação industrial





A automação industrial é uma área de pesquisa que vem ampliando sua atuação gradativamente nos últimos anos. O uso de dispositivos e a aplicação de soluções desenvolvidos em automação industrial tem grande repercussão sobretudo no setor industrial. As aplicações não se resumem a substituir o trabalho humano em tarefas exaustivas, monótonas e perigosas; elas trazem melhoria na qualidade de processos, otimização dos espaços, redução no tempo de produção e custos.

Existem diversos equipamentos utilizados na automação industrial, porém, o controlador lógico programável (CLP) é um dos mais importantes. O CLP surgiu no final da década de 1960 e revolucionou os comandos e controles industriais. Nessa época, a automação era executada quase totalmente por relés com base em lógica fixa, ou lógica *hardwired*, o que resultava em enormes armários de relés eletromecânicos interligados por circuitos elétricos e extensas fiações.

O primeiro CLP foi criado em 1968 por Dick Morley, funcionário da empresa Bedford Associates. Ele foi desenvolvido com o objetivo de substituir os armários empregados para controlar operações sequenciais e repetitivas na linha de montagem da indústria automobilística General Motors.

Essa primeira geração de CLPs usava componentes discretos e tinha baixa escala de integração. Sua utilização só era viável quando substituía painéis que continham mais de 300 relés. Tal equipamento ficou conhecido pela sigla PLC (*programmable logic controller*) – em português, CLP (controlador lógico programável).

Segundo a Associação Brasileira de Normas Técnicas (ABNT), CLP é um “equipamento eletrônico digital com *hardware* e *software* compatíveis com aplicações industriais”. Já para a National Electrical Manufacturers Association (NEMA), trata-se de um “aparelho eletrônico digital que utiliza uma memória programável para o armazenamento interno de instruções para implementações específicas, tais como lógica, sequenciamento, temporização, contagem e aritmética, para controlar através de módulos de entrada e saída vários tipos de máquinas e processos”.

Com o surgimento dos circuitos integrados, foi possível viabilizar e difundir a utilização do CLP em grande escala, melhorando o poder de processamento e diminuindo o tamanho dos equipamentos. Esse avanço está atrelado, em grande

parte, ao desenvolvimento tecnológico dos computadores, até mesmo em sua arquitetura de *hardware* e *software*. O uso de microprocessadores e microcontroladores de última geração e o de arquitetura híbrida, aliada às novas técnicas de processamento paralelo e às redes de comunicação, contribuíram para o sucesso desse equipamento industrial.

Há pouco tempo o CLP possuía arquitetura proprietária, na qual cada fabricante produzia o próprio modelo e desenvolvia os *softwares* de programação e simulação exclusivos para seus equipamentos, ou seja, não existia portabilidade. Com a adoção da norma IEC 61131-3, ocorreu a padronização da linguagem de programação e a solução para *softwares* e aplicativos foi alcançada.

Atualmente, os CLPs possuem funções específicas de controle e canais de comunicação que permitem interligá-los entre si e a computadores em rede, formando um sistema integrado. Enquanto se estudavam as propostas de padronização do *fieldbus* (barramento de campo), as redes *wireless* suplantaram essa tecnologia e se incorporaram aos CLPs como opção de coleta de sinais de chão de fábrica. Dessa maneira, eliminaram-se os condutores usados para interligá-los, propiciando troca de informações e distribuição de dados por todo o processo.

As vantagens da utilização do CLP em aplicações industriais são inúmeras e cada dia surgem novas, que resultam em maior economia, superando o custo do equipamento. Essa evolução oferece grande número de benefícios, por exemplo:

- Maior produtividade.
- Otimização de espaço nas fábricas.
- Melhoria na qualidade do produto final.
- Alto MTBF (tempo médio entre falhas).
- Baixo MTTR (tempo de máquina parada).
- Maior segurança para os operadores.
- Menor consumo de energia.
- Redução de refugos.
- Reutilização do cabeamento.
- Maior confiabilidade.
- Fácil manutenção.
- Projeto de sistema mais rápido.
- Maior flexibilidade, satisfazendo maior número de aplicações.
- Interface com outros CLPs através de rede de comunicação.

## 1.1 Evolução

No final da década de 1960 e início da de 1970, as fábricas automobilísticas não possuíam flexibilidade para mudanças constantes no processo de produção. Qualquer alteração no processo de fabricação dos automóveis, por menor que fosse, era trabalhosa e demorada. Portanto, para produzir um automóvel de cor ou modelo diferente, havia longa espera, uma vez que a produção era feita em grandes lotes. Nessa época, a tecnologia era a automação com lógica de contatos, utilizando painéis com relés eletromecânicos e muita fiação. Obviamente, essas fábricas não haviam sido projetadas para constantes modificações. Por



Figura 1.1

Típico painel de relés.



causa das limitações tecnológicas, qualquer alteração na configuração da linha de produção tinha custo elevado, pois todos os intertravamentos para controle e segurança eram feitos com painéis de relés e contadores (figura 1.1).

A utilização de lógica de contatos apresentava as seguintes desvantagens:

- **Inflexibilidade** – Efetuar mudanças no processo produtivo significava alterar toda a lógica de contatos e intertravamento. Frequentemente, isso se traduzia em sucatear enormes painéis e produzir um novo, o que demandava tempo, fator responsável por grandes interrupções e perda de produção.
- **Custo operacional elevado** – Os painéis utilizados na automação eram enormes e ocupavam grandes áreas do processo produtivo, com montagem lenta e componentes eletromecânicos suscetíveis a falhas.
- **Alto custo de desenvolvimento e manutenção** – As lógicas com relés tinham de ser minimizadas para reduzir o número de componentes e o custo envolvido na montagem. Esse fato aumentava a complexidade do projeto (implementação e documentação).

O CLP, quando foi desenvolvido, buscava substituir os enormes painéis controladores com relés eletromecânicos e quilômetros de fios que predominavam na indústria automobilística, pois esses painéis não possibilitavam flexibilidade em relação ao modelo de veículo produzido.

Com o objetivo de resolver esses problemas, a General Motors solicitou o desenvolvimento e a produção de um equipamento com as seguintes características:

- Ser facilmente programável e ter sua sequência de operação prontamente mudada, de preferência na própria planta.
- Possuir manutenção e reparo facilitados usando uma montagem de módulos encaixáveis (tipo *plug-in*).
- Funcionar em ambientes industriais com mais confiabilidade que um painel de relés.
- Ser fisicamente menor que um painel de relés para minimizar o custo de ocupação do chão de fábrica.
- Produzir dados para um sistema central de coleta de informações.
- Ser competitivo quanto ao custo em relação a painéis de relés em uso na época.

O primeiro CLP projetado previa duas placas: a placa fonte e a placa processadora controlada por *software*. Um protótipo construído com essa arquitetura mostrou-se muito lento, exigindo o acréscimo de mais uma placa, chamada Logic Solver, que tinha como finalidade implementar em *hardware* as funções mais usadas pelo *software*, acelerando, assim, o processamento.

Em 1969, a Bedford Associates apresentou para a General Motors um equipamento que atendia a suas necessidades, além de funcionar em operações distintas e ser de fácil programação. Esse equipamento era o MODICON (*modular digital controller*).

O MODICON 084 foi o primeiro CLP comercial com essas tecnologias. Tal modelo proporcionou diversos benefícios à indústria, entre eles:

- Grande economia nas mudanças de funções, por ser facilmente programável.
- Aumento na vida útil do controlador, por utilizar componentes eletrônicos.
- Menor custo de manutenção preventiva e corretiva, por ser intercambiável.
- Diminuição dos espaços físicos ocupados nas áreas produtivas, pois suas dimensões eram reduzidas.

Os CLPs foram introduzidos na indústria brasileira na década de 1980, inicialmente nas filiais de empresas multinacionais, que implantavam a tecnologia utilizada na matriz. Em pouco tempo, essa tecnologia proliferou e o CLP adquiriu grande aceitação no mercado. Em 1994, havia mais de 50 fabricantes de CLP, o que demonstrava seu sucesso e aceitação. A tabela 1.1 mostra os principais eventos na evolução tecnológica do CLP.

Década	Evento
1960	Surgimento do CLP em substituição aos painéis de controle com relés eletromecânicos – economia de energia, facilidade de manutenção, redução de espaço e diminuição de custos.

Tabela 1.1

A evolução do CLP.



Década	Evento
1970	O CLP adquiriu instruções de temporização, operações aritméticas, movimentação de dados, operações matriciais, terminais de programação, controle analógico PID. No final da década, foram incorporados recursos de comunicação, propiciando a integração entre controladores distantes e a criação de vários protocolos de comunicação proprietários (incompatíveis entre si).
1980	Redução do tamanho físico em virtude da evolução da eletrônica e adoção de módulos inteligentes de E/S, proporcionando alta velocidade e controle preciso em aplicações de posicionamento. Introdução da programação por <i>software</i> em microcomputadores e primeira tentativa de padronização do protocolo de comunicação.
1990	Padronização das linguagens de programação sob o padrão IEC 61131-3, introdução interface homem-máquina (IHM), <i>softwares</i> supervisores e de gerenciamento, interfaces para barramento de campo e blocos de funções.
Hoje	Preocupação em padronizar os protocolos de comunicação para os CLPs de modo que haja interoperabilidade, possibilitando que o equipamento de um fabricante se comunique com o de outro, o que facilita a automação, o gerenciamento e o desenvolvimento de plantas industriais mais flexíveis e normalizadas.

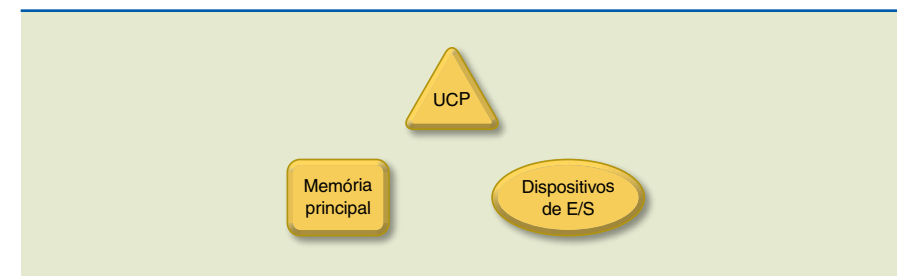
## 1.2 Controlador lógico programável

A estrutura física do CLP é um conjunto de circuitos eletrônicos interligados formados por processadores, memórias, barramentos, dispositivos de entrada e saída, fonte de alimentação e terminal de programação.

Ao analisarmos o CLP quanto a sua arquitetura e forma construtiva, podemos compará-lo com um computador especialmente adaptado para aplicações industriais. Em razão de suas características físicas, ele pode funcionar em ambientes industriais agressivos, pois suporta variações de temperatura, vibrações, ruídos elétricos, pequenas variações na tensão etc. Por tudo isso, é considerado um equipamento robusto.

Na maioria das vezes, os CLPs são disponibilizados em módulos, ou seja, cada configuração pode ser montada pelo usuário de acordo com sua necessidade. Existem também modelos na forma compacta, cuja configuração o usuário não consegue alterar. Independentemente de serem compactos ou modulares, todos os CLPs têm a mesma arquitetura e seus blocos principais são (figura 1.2):

- Unidade central de processamento (UCP).
- Memórias.
- Módulos de entrada e saída.



**Figura 1.2**  
Blocos principais do CLP.

Outros componentes secundários de *hardware* podem ser adicionados à arquitetura do CLP:

- Periféricos de interface com o usuário.
- Terminal de programação.

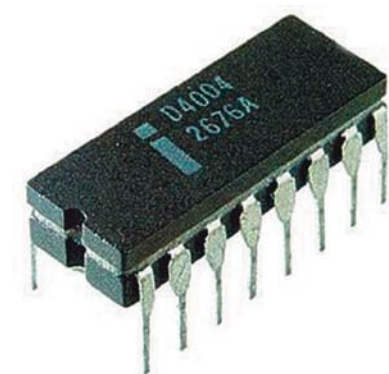
Nas próximas seções, será feita uma breve introdução ao funcionamento desses componentes e, no capítulo 3, serão apresentados mais detalhes.

### 1.2.1 Unidade central de processamento (UCP)

A unidade central de processamento (UCP) gerencia todo o sistema, controlando as operações realizadas pelas diferentes unidades funcionais.

A principal função da UCP é controlar e executar instruções presentes na memória. Ela é responsável pelo gerenciamento total do sistema, controlando os barramentos de endereços, de dados e de controle.

O microprocessador é um dos componentes vitais da UCP. O primeiro microprocessador fabricado foi o Intel 4004, lançado em 1971. Somente depois de esses dispositivos adquirirem confiabilidade é que passaram a ser utilizados na fabricação dos controladores lógicos programáveis.



**Figura 1.3**  
Processador Intel 4004.

Os primeiros CLPs usavam microprocessadores com lógica discreta. Alguns microprocessadores possuíam uma característica conhecida como *microcoded*, que foi muito importante para o desenvolvimento dos CLPs. Esses microprocessadores



tinham seu conjunto de instruções formado pela combinação de um conjunto de operações básicas.

O preço dos microprocessadores no final dos anos 1970 e início dos 1980 caiu muito e eles se tornaram componentes permanentes do CLP.

Na década de 1980, os CLPs já eram utilizados normalmente em sistemas de automação que envolviam lógica e sequenciamento. Surgia, então, a necessidade de desenvolver novas aplicações para esses equipamentos. O crescimento do mercado fez com que eles começassem a migrar para aplicações como:

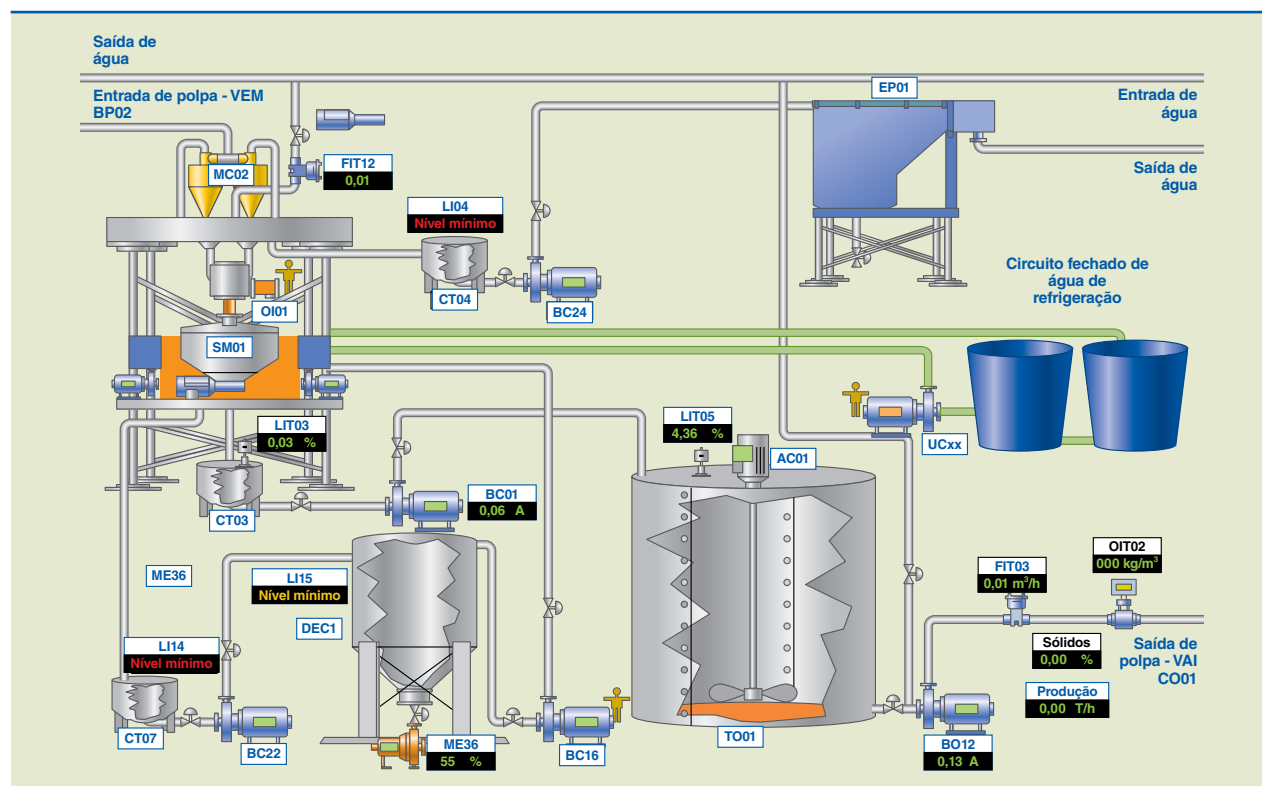
- controle de processo;
- comunicações entre homens e máquinas;
- processamento numérico.

O surgimento de novas aplicações fez com que os CLPs sofressem evoluções, de início em razão da insuficiência de memória, das limitações do barramento e do aumento da complexidade do sistema. Para melhorar seu desempenho, associado à necessidade de maior processamento, os CLPs passaram a realizar multiprocessamento.

A introdução significativa de novos produtos e das redes de comunicação de CLPs e a disponibilidade de processadores cada vez mais poderosos fizeram com que muitas das tarefas executadas por processadores, antes localizados em módulos de entrada e saída inteligentes, fossem executadas em *software* pela unidade central de processamento.

**Figura 1.4**

Esquema de controle de processo em indústria.

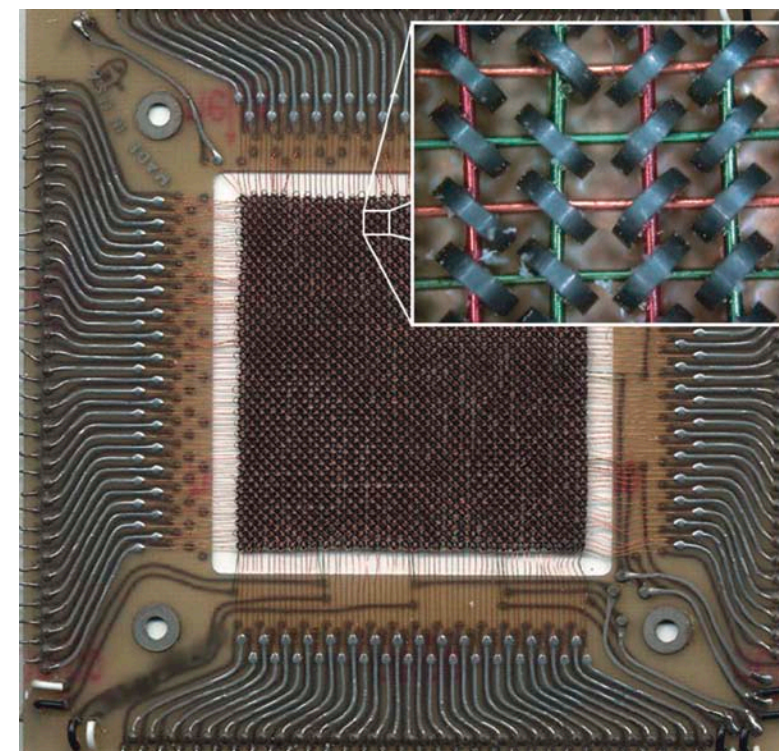


## 1.2.2 Memória

Memória é o local em que são armazenados instruções e dados, ou seja, as informações necessárias para o funcionamento do CLP. As memórias podem ser classificadas em:

- Memória executiva ou do programa monitor.
- Memória do sistema.
- Memória imagem das entradas e saídas (E/S).
- Memória de dados.
- Memória do usuário ou de aplicação.

As primeiras memórias foram desenvolvidas usando pequenos anéis de ferro magnético, cada um com 1,5 mm de diâmetro. Quando o anel era magnetizado em certo sentido, a direção de magnetização indicava o estado “0” ou “1” do bit associado a esse núcleo. Os núcleos de ferrite eram bem maiores que os *chips* hoje utilizados para armazenar dados. A memória com esse tipo de núcleo era bastante rápida: um bit nela armazenado era lido em um milionésimo de segundo. No entanto, essa memória era cara, volumosa e de leitura destrutiva, pois apagava os dados armazenados no núcleo, tornando necessário armazená-los novamente cada vez que fossem lidos.



**Figura 1.5**

Memória de núcleo de ferrite.

A primeira memória de semicondutores foi produzida em 1970 pela Fairchild. Tratava-se de uma pastilha que tinha quase o tamanho de um núcleo de ferrite, podia conter até 256 bits de memória, era de leitura não destrutiva e muito mais

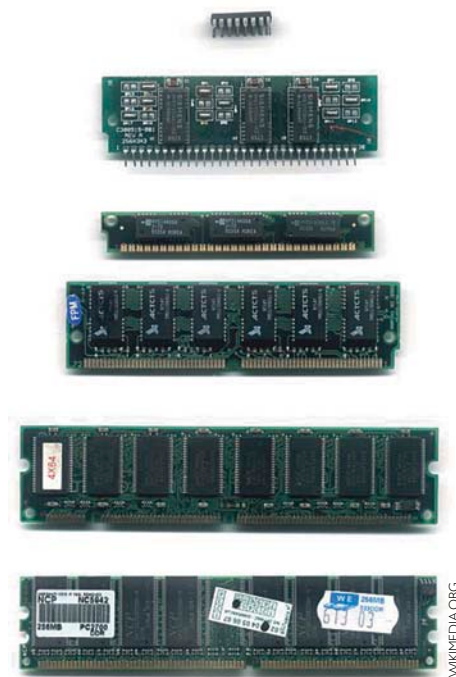


rápida do que o núcleo de ferrite, mas volumosa quando comparada com as memórias existentes hoje em dia. Apenas 70 bilionésimos de segundo eram necessários para ler um bit. O custo do bit da memória de semicondutor, entretanto, era mais alto do que o do núcleo de ferrite.

Já em 1974, o preço por bit da memória de semicondutores tornou-se menor do que o da memória de núcleo de ferrite. Em seguida, houve rápido declínio do custo de memória e aumento da densidade da memória física.

A evolução da microeletrônica viabilizou o uso de memórias semicondutoras. No início, foram utilizadas as memórias RAM (*random access memory*), que, por serem intrinsecamente voláteis, eram acompanhadas de baterias que as mantinham permanentemente alimentadas. Depois, as memórias PROM (*programmable read only memory*) passaram a ser empregadas, porém, não eram reprogramáveis. O próximo passo foi adotar as memórias não voláteis EPROM (*erasable programmable read only memory*), que eram apagadas pela exposição à luz ultravioleta. Surgiram, então, as memórias EEPROM (*electrically erasable read only memory*), que podiam ser apagadas eletricamente.

**Figura 1.6**  
Evolução das memórias RAM.



O desenvolvimento tecnológico da memória e do microprocessador melhorou significativamente o desempenho dos CLPs.

### 1.2.3 Dispositivos de entrada e saída (E/S)

Os dispositivos de entrada e saída (E/S) são utilizados para permitir a comunicação entre o CLP e os dispositivos externos. Por ser destinado ao controle de

processos, esse equipamento é obrigado a adquirir dados relativos ao ambiente no qual está inserido e àquele que deve controlar fornecendo sinais de comando. No capítulo 3, veremos detalhes dos diversos dispositivos de entrada e saída apresentados nesta seção.

Esses dispositivos podem ser divididos em dois grandes grupos:

- **Dispositivos de entrada** — classificados como entradas digitais e entradas analógicas.
- **Dispositivos de saída** — classificados como saídas digitais e saídas analógicas.

### 1.2.4 Terminal de programação

A forma mais comum de interação entre o CLP e o usuário ocorre pela combinação teclado/monitor. Antes da popularização dos computadores pessoais (PCs, *personal computers*), a programação dos CLPs era feita usando terminais de programação dedicados, com os dados fornecidos ao CLP por meio de programação via teclado. Esses terminais eram equipamentos robustos que dispunham de teclado e monitor.

Há informações indicando que os terminais de programação mais primitivos e antigos compartilhavam a memória e o processador do próprio CLP a ser programado. Eles estavam muito próximos dos terminais “burros” usados para acesso aos computadores *mainframes*. A adoção desse método deve-se ao fato de que processadores e memórias eram muito caros na fase inicial da história do CLP.

**Figura 1.7**  
Terminal de programação antigo.



Ficava economicamente inviável inserir mais componentes (memória e microprocessador) ao terminal, pois ele era utilizado apenas durante o desenvolvimento do programa do usuário e em eventuais manutenções. Depois, foram criados terminais inteligentes, dotados de memória e processador próprios. Esses terminais introduziram a chamada programação *off-line*, que tinha a vantagem de escrever o programa do usuário sem conexão com o CLP. Isso permitia o desenvolvimento do programa em local distante do CLP e, depois de pronto, era necessário testá-lo no chão de fábrica e, se necessário, retornar ao local em que havia sido desenvolvido para efetuar as mudanças necessárias.

Os protocolos usados na comunicação entre os terminais de programação e os CLPs eram proprietários, ou seja, desenvolvidos pelos fabricantes de CLP, o que impossibilitava a utilização de terminais de programação em CLPs de fabricantes diferentes. Portanto, se em uma indústria existissem quatro CLPs, um de cada fabricante, ela era obrigada a ter quatro terminais de programação. Esses terminais possuíam terminologias diferenciadas de fabricante para fabricante:

- Os da Allen-Bradley chamavam-se “terminal industrial” (*industrial terminal*).
- Os da General Electric, “terminal de desenvolvimento de programa” (PDT, *program development terminal*).
- Os da Texas Instruments, “unidade de programação em vídeo” (VPU, *video programming unit*).
- Os da Square D, “programador com tubo de raios catódicos” (CRT *programmer* ou *catode ray tube programmer*).

As linguagens de programação também eram proprietárias e não havia padronização. As teclas de atalhos desses terminais, para acelerar as chamadas de funções usadas nas linguagens que cada fabricante havia definido, estavam em posições diferentes. Dessa maneira, nem mesmo o formato e as funções dos teclados eram padronizados.

Os terminais de programação portáteis possuíam poucas teclas e *display* de LED ou cristal líquido. Eles tinham funcionalidade reduzida, sendo utilizados apenas para alterar dados ou pequenos trechos de código do programa do usuário.

Com esses terminais de programação havia outros dispositivos com funções específicas:

- Unidades de fita perfurada e fita magnética, para armazenar os programas desenvolvidos. As fitas perfuradas eram mais baratas e resistentes às agressões do ambiente industrial do que as fitas magnéticas.
- Impressora, que, na época dos terminais “burros”, era diretamente conectada ao CLP, imprimindo os programas desenvolvidos pelo usuário sem muitos refinamentos. A razão do estilo lacônico das impressões era a economia de memória.

Os computadores pessoais começaram a se popularizar em meados dos anos 1980. Nesse período, foram desenvolvidos programas para que esses computadores substituíssem os terminais de programação e desempenhassem suas funções. Tal processo não foi aceito de imediato, pois os PCs foram recebidos com certo ceticismo, devido aos altos custos e à fragilidade em ambientes industriais. O disco rígido, frágil diante de agentes agressivos, como temperatura, poeira e umidade, era o principal foco de preocupação, porque uma falha desse componente envolveria diversas horas de trabalho de programação de um sistema complexo. Para contornar o problema, os fabricantes de CLP passaram a produzir computadores pessoais mais resistentes.

## 1.2.5 Periféricos de interface com o usuário

A maioria das aplicações com controladores lógicos programáveis necessita de uma interface homem-máquina (IHM). Em um processo produtivo, o operador precisa interagir com o processo ou a máquina, comandando operações, alterando parâmetros, visualizando situações e recebendo diagnósticos.

Quando se iniciou o desenvolvimento dos CLPs, a interface com o usuário era praticamente igual à existente nos painéis de relés. Funções booleanas podiam ser introduzidas por meio de botoeiras e visualizadas com o uso de lâmpadas de sinalização. Os CLPs com grande número de bits de entrada e capacidade de processamento aritmético permitiam a leitura de dados numéricos utilizando chaves *thumbwheel*. Tratava-se de chaves com dez posições, cada uma delas representando um número. Quando conectada às entradas digitais do CLP, a *thumbwheel* indicava qual número o usuário escolheu. A indicação podia ser em código hexadecimal ou BCD. Várias chaves podiam ser combinadas para a entrada de números com vários dígitos.

Com a redução do custo dos *displays* de cristal líquido, surgiram as interfaces homem-máquina (IHMs) conforme mostra a figura 1.8. Esses dispositivos eram constituídos de teclado, *display* e processador, conectados por meio de uma rede de comunicação de dados a um ou mais CLPs.



**Figura 1.8**  
Interface homem-máquina (IHM).



Com a adoção do CLP, as fiações dos painéis de relés tornaram-se bem mais simples. A fiação relacionada à interface com o usuário permaneceu a mesma, ou seja, as conexões de botoeiras, lâmpadas de sinalização e chaves *thumbweel* ao CLP continuaram existindo. Essa fiação foi eliminada com a utilização de IHMs e houve aumento nos dados de entrada exibidos ao usuário. Tudo isso pôde ser feito a um custo muito baixo, bastando fazer a programação de novas telas. No tempo das botoeiras, inserir novos dados significava a compra de mais botoeiras ou lâmpadas de sinalização e mais módulos de entrada e saída do CLP. Com isso, os sistemas se tornaram muito mais flexíveis e amigáveis, permitindo melhor interação com o usuário.

# Capítulo 2

## Tipos de automação industrial



Diversos são os meios de controle de máquinas, processos e plantas industriais. O objetivo deste capítulo é comentar alguns desses meios, destacando características básicas, seus componentes e para qual tipo de automação eles são mais indicados.

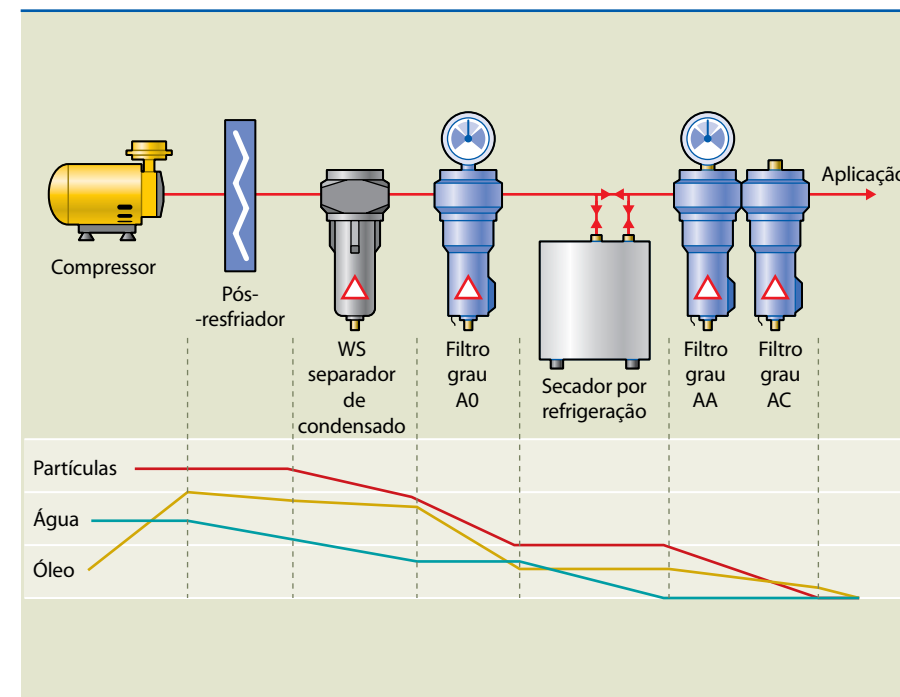
A avaliação para novos projetos deve levar em conta a eficiência necessária ao processo, o ambiente em que será realizado, bem como a relação entre custo e benefício que proporcionará à unidade produtiva. Com esses pontos resolvidos, é iniciada a apresentação dos tópicos. Levando em conta que os leitores já possuem requisitos mínimos em comandos elétricos, não vamos nos aprofundar nesse conteúdo.

## 2.1 Comandos pneumáticos

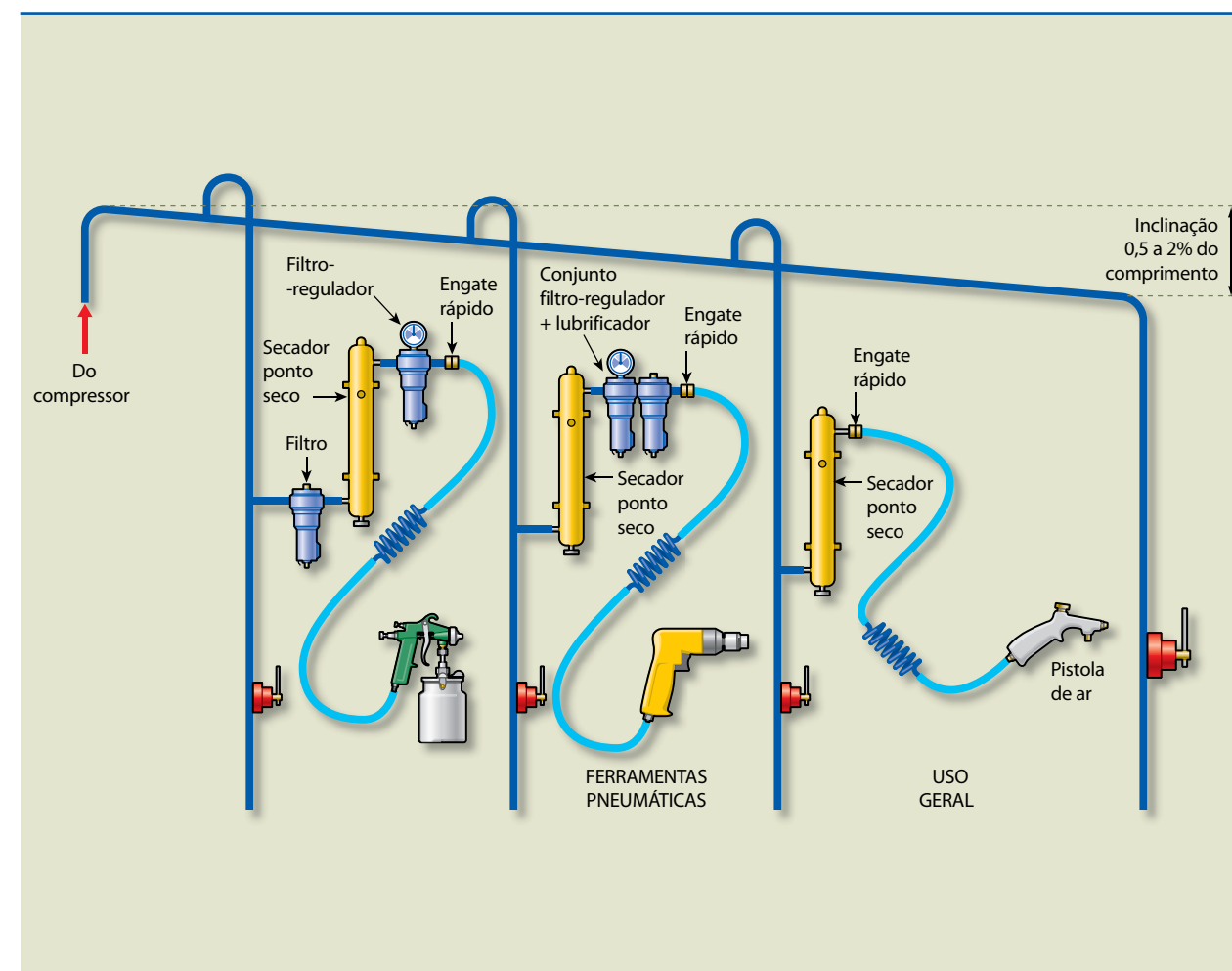
Outra maneira de automatizar máquinas e dispositivos é com a utilização da pneumática. Esse tipo de automação tem maior foco em aplicações de movimentação, seleção e transporte de produtos. O ar possui características positivas, como compressibilidade e baixo custo, além de não gerar faíscas, podendo ser usado em grande escala. Um sistema pneumático é formado basicamente pelos blocos: gerador, qualidade e distribuição, manobra e atuadores.

O **bloco gerador** é composto por um ou vários compressores que armazenam o ar atmosférico em reservatórios, nos quais a pressão interna é superior à pressão atmosférica do local. O ar comprimido é convertido em energia cinética quando transferido aos atuadores pneumáticos na ponta dos dispositivos. O bloco gerador pode ter a função de filtragem de partículas e desumidificação, ou seja, ele retira toda a água possível do ar que foi comprimido. A figura 2.1 mostra o esquema de funcionamento de um sistema gerador.

O **bloco de qualidade e distribuição** é um conjunto de filtros secundários que retiram impurezas e umidade do ar distribuído. Existem ainda um regulador de pressão, que mantém a pressão constante na linha de distribuição, e um dispositivo de lubrificação, quando necessário, que vaporiza óleo e o mistura no ar que será distribuído, garantindo a lubrificação de todos os elementos do sistema. Além desses componentes, há também dutos que levam o ar comprimido até o local onde será utilizado. Exemplos desses elementos podem ser verificados na figura 2.2.



**Figura 2.1**  
Sistema gerador de ar comprimido.



**Figura 2.2**  
Sistema de qualidade e distribuição de ar.





O **bloco de manobra** é constituído de válvulas essenciais ao controle da manobra que se pretende executar com os atuadores pneumáticos. Alguns modelos de válvulas e respectivas funcionalidades serão apresentados na próxima seção. A figura 2.3 mostra um exemplo.

**Figura 2.3**

Válvula de controle de vazão.



O **bloco dos atuadores** é composto, em geral, por atuadores pneumáticos que possuem um eixo que se movimenta linearmente de modo bidirecional, ou seja, avançam e recuam de acordo com a entrada e saída do ar. A figura 2.4 apresenta alguns modelos de atuadores pneumáticos.

**Figura 2.4**

Atuadores pneumáticos.



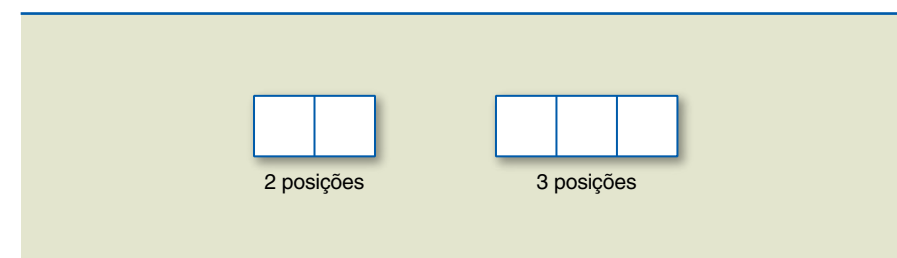
Para compreender melhor a automação pneumática, é necessário conhecer seus componentes. A seguir, são apresentados modelos genéricos que podem ser compatibilizados com os mais diversos tipos de válvulas e atuadores comercialmente encontrados, além de outros componentes menos complexos.

## 2.1.1 Válvulas de controle direcional

No momento em que o sistema exige, as válvulas de controle direcional distribuem o ar aos dispositivos atuadores. Elas permitem ou não a passagem de ar em um único sentido, abrindo ou fechando dutos internos que a compõem. São utilizadas normalmente no avanço ou no recuo dos atuadores.

As válvulas pneumáticas são classificadas conforme o número de vias que possuem, as posições em que podem ser atuadas e a forma de seu acionamento. O dispositivo de acionamento de uma válvula pneumática também é comumente chamado de piloto.

Para a diagramação de circuitos pneumáticos, as válvulas são representadas de modo a simular seu funcionamento interno. São sempre simbolizadas por um retângulo, dividido em quadrados, que indicam o número de posições da válvula, correspondendo ao número de movimentações que essa válvula executa por meio de acionamentos externos. A figura 2.5 exemplifica os retângulos e números de posições de válvulas.



**Figura 2.5**

Representação de posições de válvulas.

O controle de fluxo do ar feito pelas válvulas é possível graças à disposição mecânica interna de seus cilindros em relação às vias de acesso a esse atuador. As vias de uma válvula são os acessos pelos quais o sistema de distribuição de ar comprimido chega aos atuadores. A representação das vias é feita com números em volta do diagrama esquemático da válvula.

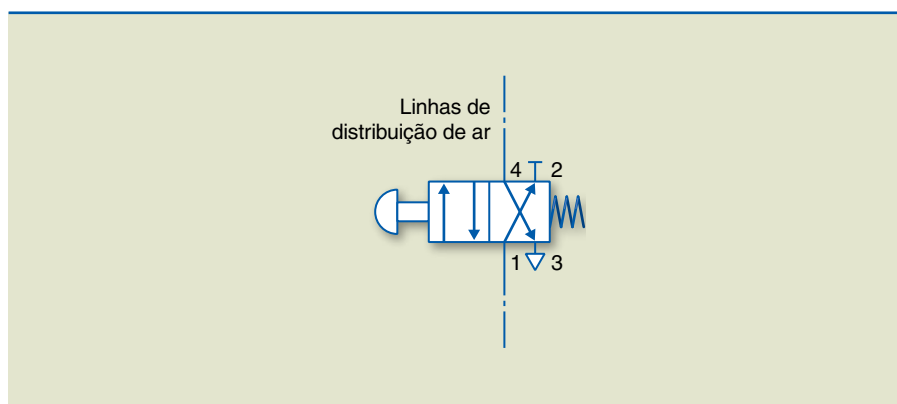
No interior de cada quadrado desenhado no retângulo que simboliza uma válvula, podem existir setas (figura 2.6), representando a livre passagem de ar entre duas vias, e bloqueios (T), representando a interrupção da passagem de ar de uma das vias. Esses quadrados recebem os sinais da linha de distribuição do ar (linhas que interligam válvulas e acionadores do diagrama esquemático).

O quadrado interligado no diagrama de distribuição indica a situação de funcionamento da válvula no estado normal. O quadrado ao lado que não possui as linhas de distribuição de ar ligadas a ele representa a situação em que as vias estarão intercomunicadas a partir do acionamento do piloto. Na figura 2.6, o estado normal de funcionamento da válvula garante que a via 4 está ligada à via 3 e que a via 1 está ligada à via 2. Quando o piloto da esquerda é acionado, a via 1 passa a ter comunicação direta com a via 4, e a via 3, comunicação direta com a via 2. Essa situação permanece enquanto o piloto estiver pressionado. Ao ser liberado, a mola à direita garante o retorno automático à posição original da válvula.



**Figura 2.6**

Esquema de funcionamento de uma válvula.



O acionamento do piloto dessas válvulas nada mais é que a movimentação de seus cilindros internos. Essa movimentação pode ser executada por ação muscular (provocada pelo homem), por ação mecânica (provocada por um dispositivo mecânico ou outro atuador pneumático) ou por ação eletromecânica (por meio de solenoides que movimentam um núcleo magnético e provocam o deslocamento). A figura 2.7 mostra a simbologia utilizada para cada modelo de piloto usado em pneumática.

**Figura 2.7**

Tipos de acionamento de válvulas pneumáticas.

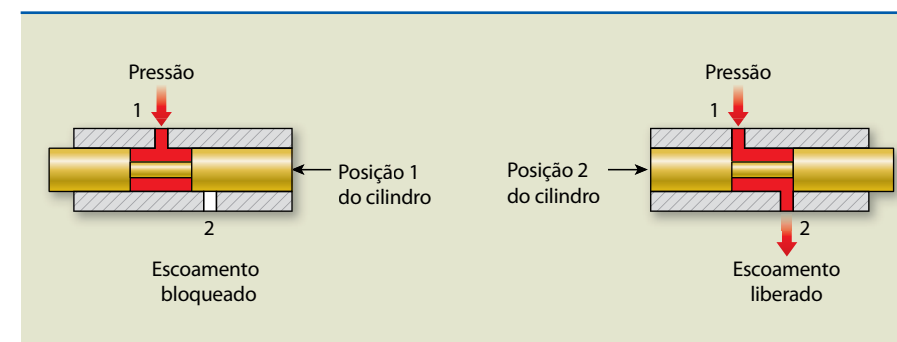
Ação muscular		Ação mecânica	
	Uso geral		Pino
	Botão		Mola
	Alavanca		Rolete
	Pedal		Rolete em sentido único
Acionamento elétrico			
	Bobina solenoide		

Vamos analisar três modelos de válvulas comuns encontrados na automação industrial: duas vias e duas posições; três vias e duas posições; cinco vias e duas posições.

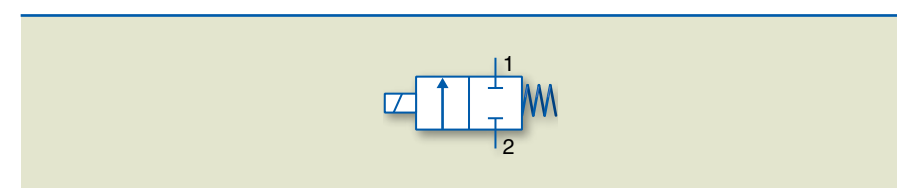
Na válvula de duas vias e duas posições (figura 2.8), o ar que entra pela via 1 é interrompido pela posição 1 do cilindro da válvula, porém, tem possibilidade de escoamento pela via 2, uma vez que o piloto esteja acionado para movimentar o cilindro da válvula para a posição 2.

**Figura 2.8**

Esquema de funcionamento de uma válvula de duas vias e duas posições.



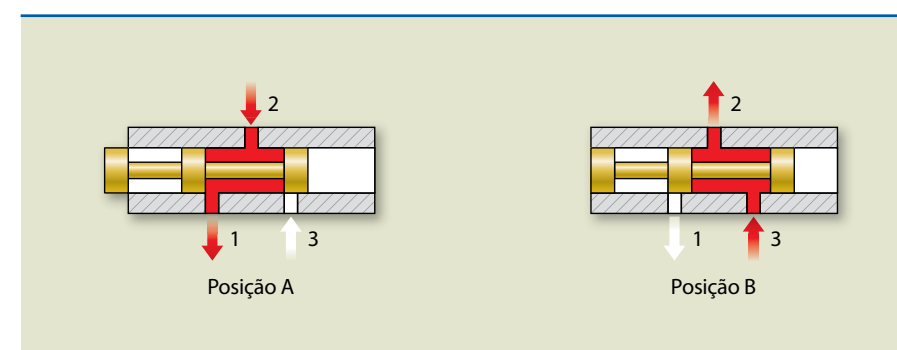
A figura 2.9 exemplifica o diagrama de uma válvula de duas vias e duas posições com simples solenoide e retorno por mola.



**Figura 2.9**

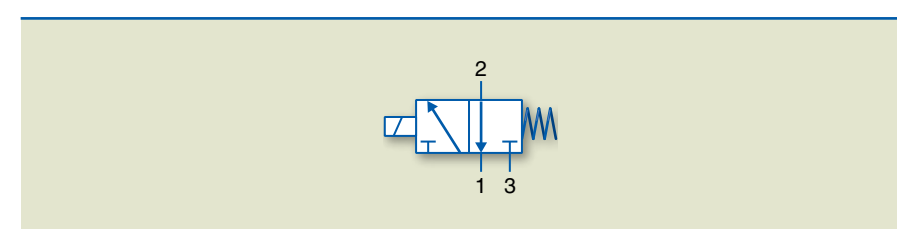
Válvula de duas vias e duas posições com retorno por mola.

Na posição de descanso (posição A da figura 2.10), a válvula de três vias e duas posições permite que o ar tenha livre passagem entre as vias 2 e 3 da válvula, mantendo a via 1 bloqueada. Quando seu piloto é acionado (posição B da figura 2.10), deslocando o cilindro interno à direita, a válvula passa a permitir a passagem de ar da via 1 para a via 2, o que interrompe o fluxo de ar através da via 3. Em tais condições, a válvula permanece nessa posição até que deixe de ser atuada, voltando à posição A. A figura 2.10 exemplifica o esquema de funcionamento dessa válvula, e a figura 2.11, a simbologia de uma válvula de três vias e duas posições com piloto acionado por solenoide e retorno por mola.



**Figura 2.10**

Esquema de funcionamento da válvula de três vias e duas posições.



**Figura 2.11**

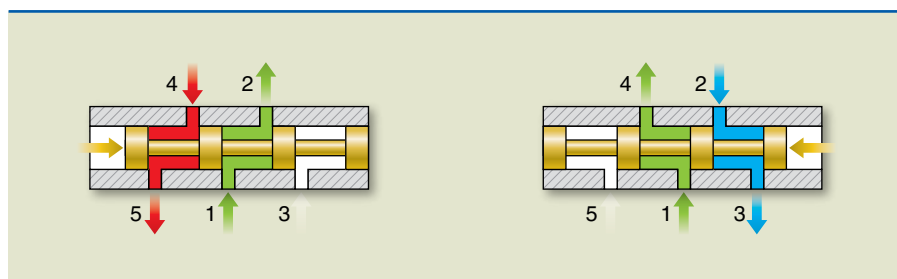
Válvula de três vias e duas posições com piloto acionado por solenoide e retorno por mola.



Na posição de descanso (cilindro à direita), a válvula de cinco vias e duas posições permite a passagem de ar entre as vias 1 e 2 e também, de maneira independente, entre as vias 4 e 5, mantendo a via 3 com fluxo de ar interrompido. Quando seu piloto é acionado (cilindro à esquerda), passa a permitir a passagem de ar da via 2 para a via 3 e da via 1 para a via 4, mantendo o fluxo de ar interrompido através da via 5. A figura 2.12 exemplifica o esquema de funcionamento dessa válvula, e a figura 2.13, a representação de uma válvula de cinco vias e duas posições com duplo acionamento por solenoide.

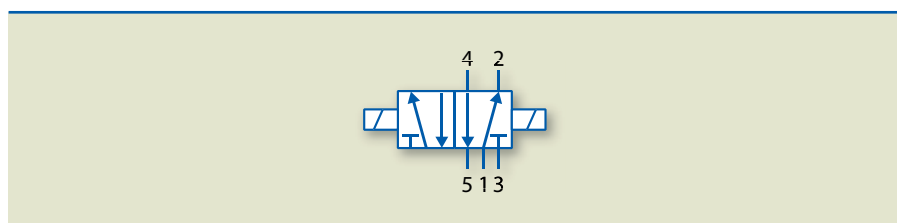
**Figura 2.12**

Funcionamento de uma válvula de cinco vias e duas posições.



**Figura 2.13**

Simbologia de uma válvula de cinco vias e duas posições com duplo acionamento por solenoide.

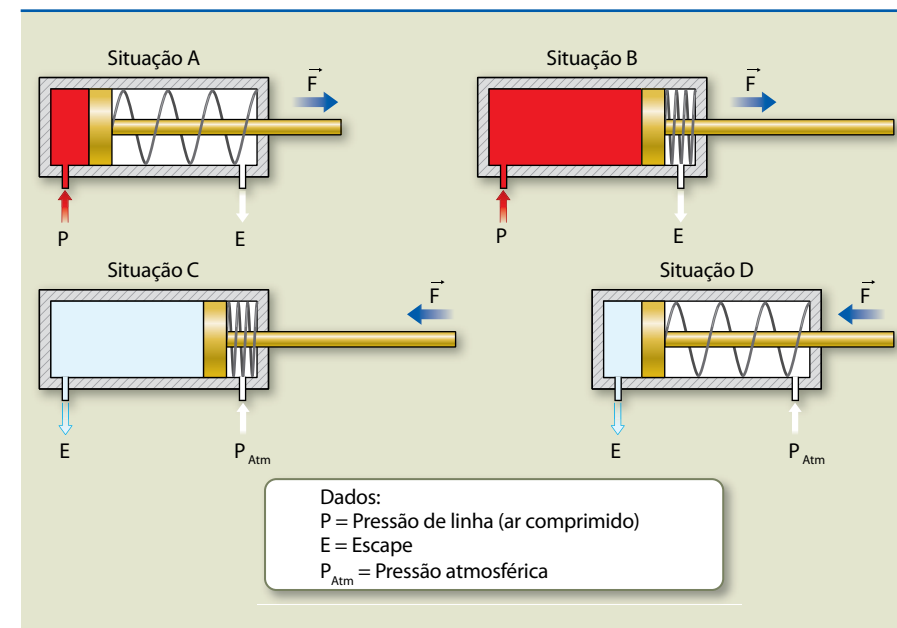


### 2.1.2 Atuadores

Os atuadores pneumáticos são os dispositivos que realizam o trabalho quando recebem a energia do ar comprimido por meio das válvulas de distribuição e controle. Em sua grande maioria, são cilindros ou atuadores pneumáticos que propiciam o deslocamento de um eixo de acordo com suas características de construção interna. São basicamente classificados como atuadores de simples ação e atuadores de dupla ação.

A figura 2.14 ilustra o atuador de simples ação. Apesar de ser compressível, quando o ar comprimido enche uma das câmaras do cilindro, passa a exercer força na parede do êmbolo que isola uma câmara da outra. Dessa maneira, a força contrária que a mola exerce é vencida e ocorre o deslocamento axial de acordo com o modelo do cilindro. Esse avanço é representado pelas situações A e B da figura 2.14.

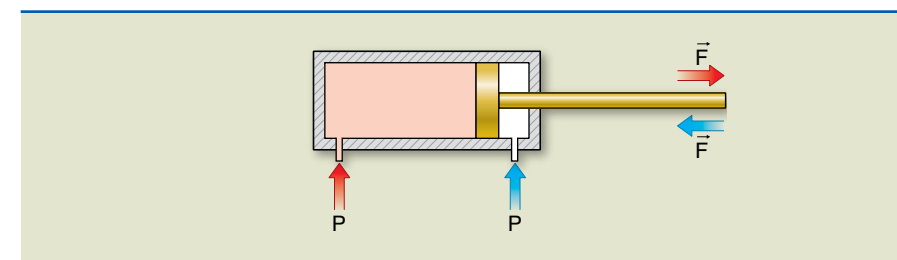
Ao ser acionado, o eixo permanecerá avançado enquanto a pressão interna da câmara for superior à pressão atmosférica, mais a força que a mola exerce no sentido contrário. Assim, uma vez interrompido o fluxo de ar, a pressão em ambos os lados passa a ser a mesma. A mola devidamente instalada faz sua função, ou seja, garantir que o eixo do cilindro recue, até que seja novamente requisitado seu avanço com uma nova “injeção” de ar comprimido. Esse recuo é representado pelas situações C e D da figura 2.14.



**Figura 2.14**

Atuadores de simples ação.

O atuador de dupla ação possui avanço igual ao atuador de simples ação, porém, não utiliza mola interna. Depois de se movimentar, o eixo desse atuador permanecerá na mesma posição e imóvel, até que receba uma força mecânica externa ou que a câmara oposta receba ar comprimido. O esquema de funcionamento é apresentado na figura 2.15.

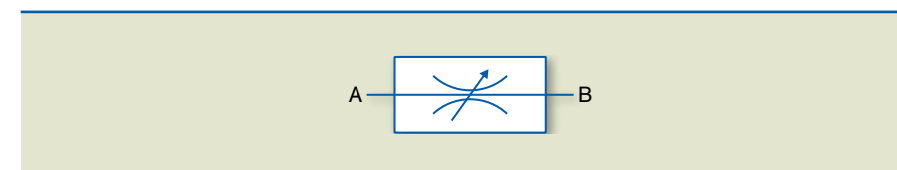


**Figura 2.15**

Atuador de dupla ação.

### 2.1.3 Outros dispositivos

A **válvula reguladora de fluxo bidirecional** (figura 2.16) possibilita a redução da área de seção em determinado ponto da passagem do ar entre as vias da válvula. Esse recurso possibilita que o volume de ar que passa instantaneamente por ela seja reduzido e, por consequência, controla a velocidade de avanço ou recuo do eixo do atuador. Essa válvula é comumente utilizada em processos mais delicados ou que exigem cuidados especiais com os produtos ou objetos que serão manipulados.



**Figura 2.16**

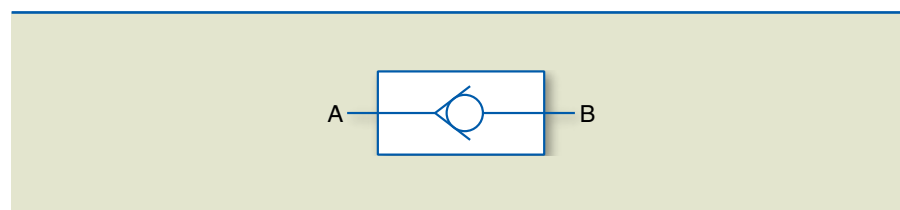
Válvula reguladora de fluxo bidirecional.



A **válvula de retenção** (figura 2.17) garante que o ar das vias que a compõem siga somente em um sentido, ou seja, permite a passagem de ar de A para B, mas não de B para A. É muito usada em construções de lógicas específicas de acionamentos e processos controlados somente por ar comprimido.

**Figura 2.17**

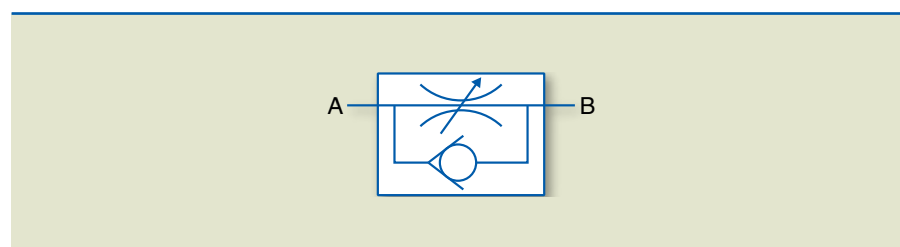
Válvula de retenção.



A **válvula reguladora de fluxo unidirecional** (figura 2.18) agrega as funções das válvulas anteriores, ou seja, regula a passagem de ar em um sentido e o deixa passar livremente no outro.

**Figura 2.18**

Válvula reguladora de fluxo unidirecional.

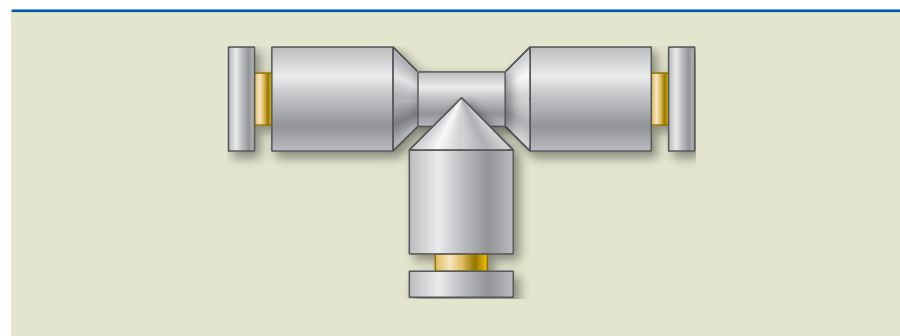


Os **silenciadores** são terminais utilizados nas ligações pneumáticas de maneira estratégica para reduzir o ruído causado pela liberação do ar comprimido em atuadores que estavam acionados.

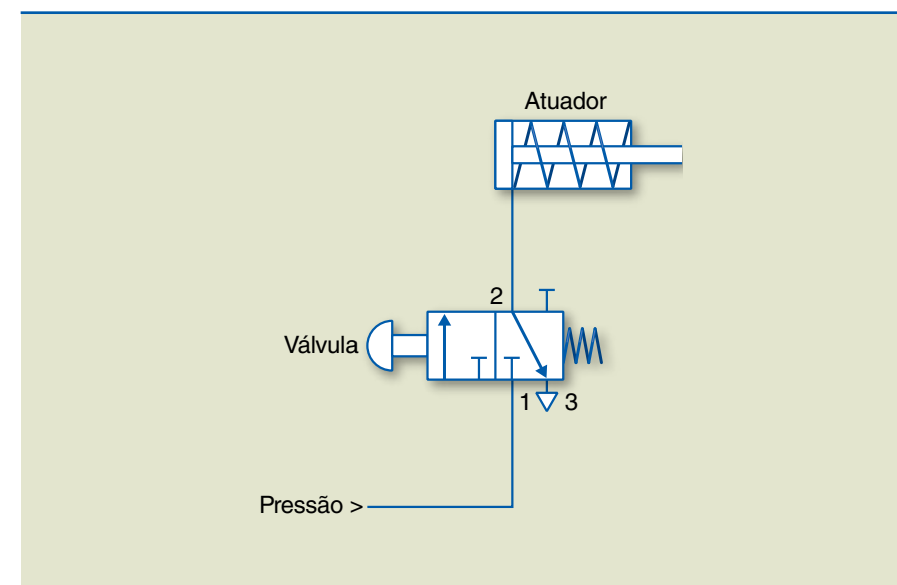
A **união em T** é um bifurcador que compartilha o ar entre seus terminais. Não tem por função o controle do fluxo de escoamento entre suas vias, porém, é de grande importância na implementação de projetos de pneumática. A figura 2.19 apresenta um modelo de elemento de união em T usado nesses processos.

**Figura 2.19**

Bifurcador T.



Um exemplo básico de esquema de acionamento de um atuador de simples ação é apresentado na figura 2.20. Uma vez acionado o botão, o atuador avança e permanece avançado até que o botão seja solto.



**Figura 2.20**

Avanço de atuador por ação muscular.

## 2.2 Comandos elétricos e eletropneumáticos com CLP

A eletropneumática engloba o controle e a detecção dos movimentos realizados por válvulas de comando e atuadores pneumáticos por meio de dispositivos e sensores elétricos. As lógicas de acionamento utilizadas em comandos elétricos também são empregadas para o acionamento de solenoides, que, por sua vez, servem como pilotos em válvulas eletropneumáticas, assim como os sensores de fim de curso detectam o movimento dos atuadores. Entre os dispositivos e sensores elétricos, pode-se destacar o uso de controladores eletrônicos “inteligentes” (CLPs e computadores).

Para que o CLP possa interagir com sistemas de comandos eletropneumáticos, é necessário realizar uma simples adequação das grandezas manipuladas no painel elétrico com os padrões utilizados em suas interfaces. As botoeiras e chaves usadas em painéis de comandos elétricos convencionais continuam servindo a necessidade das entradas digitais do CLP, assim como os contadores continuam servindo como atuadores no acionamento de motores e carga de maior potência, uma vez que o CLP não tem por função o “chaveamento” de cargas de potência diretamente em suas interfaces. Mais detalhes sobre as características técnicas e o funcionamento do CLP serão apresentados no capítulo 3 deste livro.

Já para a substituição da lógica em sistemas pneumáticos, é necessária a adequação dos elementos utilizados nesse sistema. Para o acionamento dos atuadores, o CLP assume o controle das válvulas eletropneumáticas, fazendo com que os solenoides executem o acionamento mecânico dos pilotos dos mais diversos tipos de válvulas, independentemente do número de vias e mecanismos de retorno. Desse modo, a corrente elétrica que o CLP pode chavear é utilizada no acionamento das válvulas eletropneumáticas, que, por sua vez, podem controlar o



avanço e o retorno de atuadores pneumáticos. Já a detecção dos movimentos dos atuadores, se realmente ocorreram ou não, é possibilitada por chaves chamadas sensores de fim de curso. O acionamento das chaves fim de curso é mecânico e possibilita a passagem de corrente elétrica por seus terminais quando o movimento esperado é executado.

Outra forma muito comum de detectar se o acionamento dos atuadores foi executado é por meio de sensores magnéticos instalados em posições estratégicas no corpo do cilindro. O êmbolo do cilindro possui características magnéticas que acionam pequenas microchaves magnéticas instaladas em suas extremidades. Na figura 2.22, podem-se observar detalhes de um atuador pneumático.

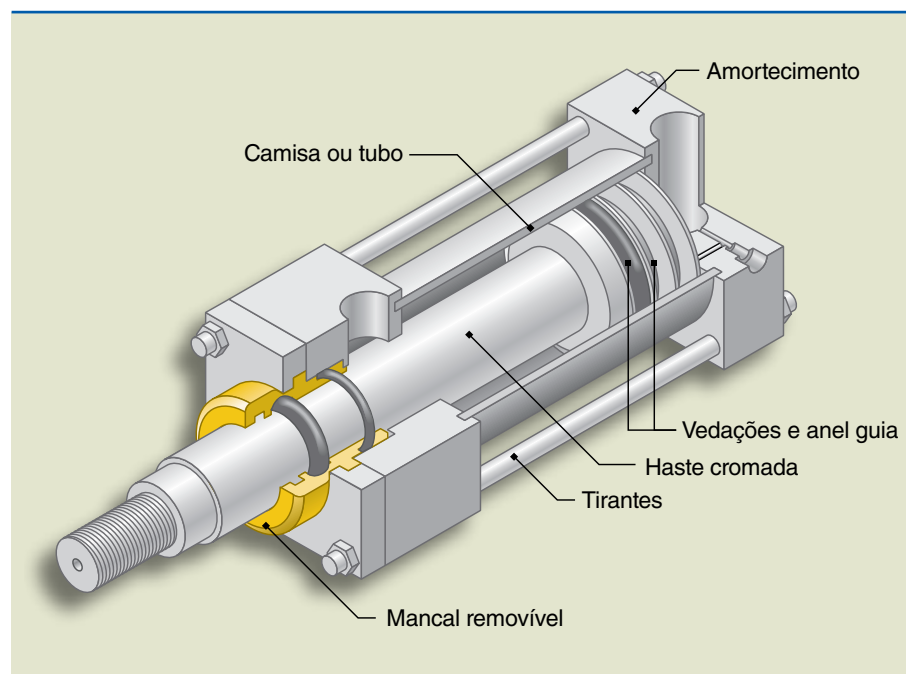
**Figura 2.21**

Válvula pneumática com acionamento por solenoide.

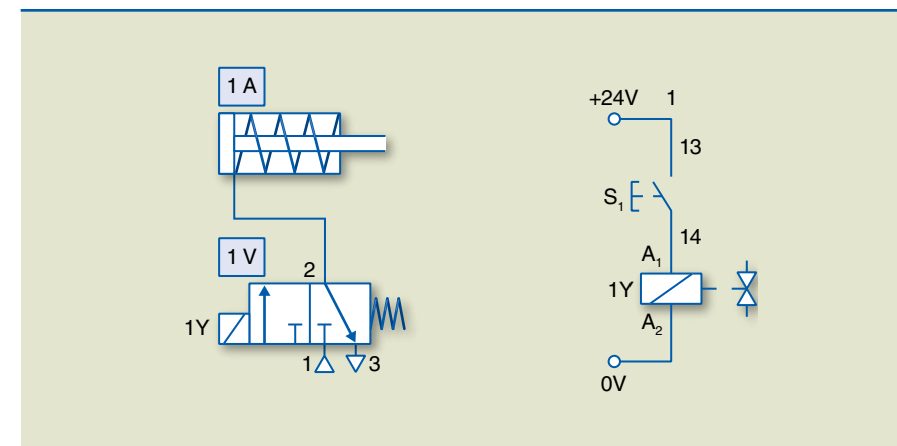


**Figura 2.22**

Atuador pneumático em detalhes.



Comandos eletropneumáticos são dispositivos que utilizam energia elétrica para acionar dispositivos pneumáticos, assim como para detectar eletricamente o posicionamento de seus elementos controlados. A figura 2.23 exemplifica uma simples automação em comando eletropneumático.



**Figura 2.23**

Acionamento de um atuador pneumático com comando elétrico.

## 2.3 CLP versus computador

Uma das principais dúvidas no tema automação industrial é se um computador pode ser utilizado para controlar tais tarefas. Nesta seção, são apresentados os prós e os contras da utilização de computadores em aplicações industriais.

Os computadores atuais possuem grande capacidade de processamento e armazenamento de dados e vasta possibilidade de interfaceamento com dispositivos externos. No entanto, existem pontos fracos que devem ser levados em consideração. A princípio, os sistemas operacionais projetados para trabalharem com várias funções, muitas delas não testadas previamente, podem gerar estragos em programas dedicados e específicos, concebidos para controles. Outro problema é o fato de serem plataformas abertas, pois, caso não sejam tomados os devidos cuidados, pode ocorrer invasão de *softwares* mal-intencionados. A instabilidade de alguns sistemas operacionais também pode ser um item decisivo à não utilização de computadores coordenando processos de alta segurança.

Um fato importante é que alguns fabricantes têm se especializado na produção de computadores para o trabalho na indústria, os chamados computadores industriais. Aos olhos de um operador, por trás do monitor, essas máquinas são similares aos computadores (possuem teclado e mouse), mas tecnicamente existem grandes diferenças construtivas entre ambos.

Para compatibilizar seus computadores aos ambientes industriais, alguns fabricantes utilizam recursos mecânicos, o que torna os equipamentos mais robustos e resistentes a vibrações e choques, evitando que as placas internas saiam dos *slots* em que foram encaixadas, bem como coxins de borracha que absorvem parte dessas interferências mecânicas, protegendo o conjunto disco rígido e unidades de CD e outras mídias.



Além desses cuidados, aplicam o chassi metálico com pintura condutiva e antiestática, que garante níveis de aterramento em todos os pontos do gabinete e forma também uma gaiola de Faraday, proporcionando grande redução de interferências eletromagnéticas nos componentes internos do gabinete. Dessa maneira, completam as características físicas e técnicas desse equipamento os componentes eletrônicos industriais, que toleram altas temperaturas e o posicionamento das placas, que facilita o fluxo de ar filtrado e forçado para dentro do gabinete, criando uma bolsa de ar que impede a entrada de pequenas partículas prejudiciais ao funcionamento do computador. As figuras 2.24 a 2.26 exemplificam alguns dos detalhes comentados.

**Figura 2.24**

Modelo gabinete de micro industrial.



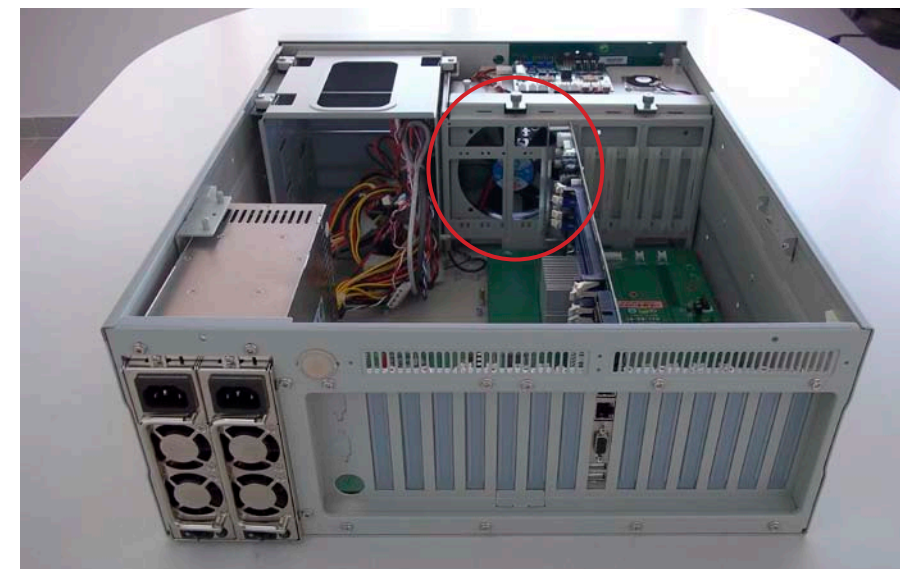
© CORTEX INDUSTRIAL SYSTEM LTDA.

**Figura 2.25**

Detalhe da placa de interconexão (Backplane).



© CORTEX INDUSTRIAL SYSTEM LTDA.



© CORTEX INDUSTRIAL SYSTEM LTDA.

**Figura 2.26**

Detalhe da ventilação forçada e fonte redundante.

Os CLPs possuem um sistema operacional dedicado ao gerenciamento do equipamento, prevendo quais os tipos de periféricos que serão utilizados. Esses periféricos, como os cartões de entrada e saída, digitais e analógicos, são fabricados pela mesma empresa que desenvolveu a unidade de processamento e o respectivo sistema operacional. Normalmente, não aceitam cartões de outros fabricantes, reduzindo, assim, a probabilidade de falhas por incompatibilidades de CLPs.

Os computadores possuem barramentos universais que permitem que outros fabricantes de outras marcas compatibilizem seus produtos. Isso requer processos de homologação que nem sempre são rápidos e completos, pois tais homologações são executadas com softwares e versões de sistemas operacionais específicos. Qualquer variação nessa configuração pode exigir nova homologação.

Os computadores possuem capacidade de processamento para equacionamentos matemáticos superiores à boa parte dos CLPs de mercado. Esse detalhe, em alguns processos, é um diferencial importante, como no caso de equipamentos de medição em massa e aferição de produtos acabados. A possibilidade de geração de relatórios locais e a totalização desses resultados já formatados para análises posteriores acabam flexibilizando a solução de automação em células de aferição de produtos.

Já nas aplicações lógicas e sequenciais, o CLP se destaca pela velocidade de resposta, pela facilidade de implementação de projetos e por sua robustez. Mesmo com todas as características apresentadas sobre computadores industriais, ainda assim os CLPs superam os computadores industriais no que se refere à velocidade de programação e à tolerância a ruídos externos nos canais de comunicação e na entrada de dados, além de suas interfaces já estarem dimensionadas para os diversos padrões elétricos industriais utilizados.



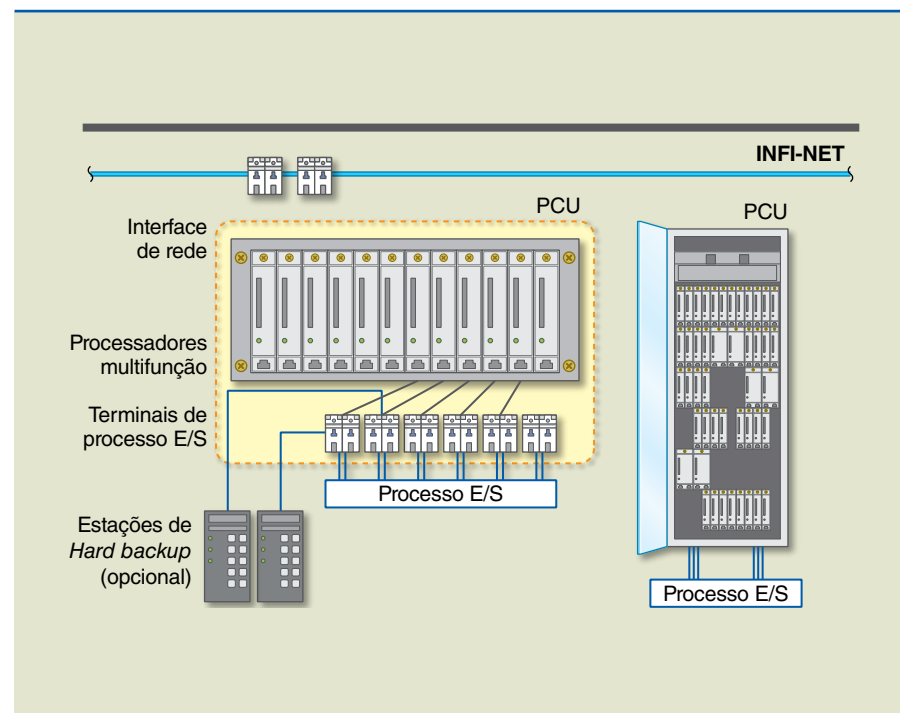
O tempo reduzido de manutenção e substituição de interfaces dos CLPs também é um ponto muito forte na utilização dessa tecnologia em processos e controles industriais, o que motiva os projetistas a adotá-los. No entanto, ainda resta um papel muito importante do computador na indústria: o monitoramento do processo. Exercendo a função de interface homem-máquina no monitoramento e interação com os processos industriais, o computador tem ganhado espaço importante nessa aplicação. Dada a relevância desse fato, o tema é apresentado em um capítulo exclusivo.

## 2.4 Sistema digital de controle distribuído (SDCD)

O sistema digital de controle distribuído (SDCD) não é um único produto, mas um conjunto de dispositivos e elementos interligados em redes de comunicação de alta velocidade e confiabilidade, utilizados em sistemas e plantas industriais mais complexas. Esse conjunto de dispositivos possui certa independência da parte do processo para a qual foi destinado, porém, obedece a uma lógica única em um nível superior hierárquico no controle do sistema. O SDCD é um sistema complexo e redundante, no qual eventuais falhas são corrigidas de maneira automática por outros elementos previamente programados que assumem o controle em caso de falhas.

A programação desse tipo de topologia é avançada, exigindo conhecimento muito claro do processo a ser controlado. A programação de cada elemento do processo deve ser sincronizada com a programação digital do sistema. Sistemas desse porte são utilizados em indústrias petroquímicas e químicas, que exigem grande capacidade de processamento, distribuição da inteligência no processo e velocidade no tratamento das informações. Um esquema de interligação de um SDCD pode ser visualizado na figura 2.27.

**Figura 2.27**  
Diagrama em blocos de um SDCD.



## 2.5 Ambiente industrial (IP ou NEMA)

Os ambientes industriais são classificados de acordo com determinados parâmetros de agressividade do meio. Esses parâmetros são muito importantes, pois ajudam os fabricantes a testar seus produtos e os projetistas a informar qual a necessidade para cada equipamento em cada ambiente em que será instalado.

Poeira, umidade, gases ácidos, vibração e jatos de líquidos muitas vezes fazem parte do dia a dia de máquinas, equipamentos e plantas industriais. A classificação e índices de agressividade desses fatores são normalizados por duas grandes instituições: National Electrical Manufacturers Association (NEMA) e IP Institute. Os índices apresentados por essas instituições servem de referência para a classificação de produtos fabricados para as indústrias e alguns deles são apresentados nas tabelas 2.1 e 2.2.

**Tabela 2.1**  
Classificação IP

IP – Classificação		
Dígito	Primeiro dígito característico	Segundo dígito característico
0	Não protegido	Não protegido
1	Protegido contra objetos sólidos maiores que 50 mm	Protegido contra quedas verticais de gotas d'água
2	Protegido contra objetos sólidos maiores que 12,5 mm	Protegido contra queda de gotas d'água para uma inclinação máxima de 15°
3	Protegido contra objetos sólidos maiores que 2,5 mm	Protegido contra água aspergida
4	Protegido contra objetos sólidos maiores que 1 mm	Protegido contra projeções d'água
Dígito	Primeiro dígito característico	Segundo dígito característico
5	Protegido contra poeira	Protegido contra jatos d'água
6	Totalmente protegido contra poeira	Protegido contra ondas do mar
7		Protegido contra imersão
8		Protegido contra submersão

Exemplo: IP65 – Equipamento totalmente protegido contra poeira e protegido contra jatos de água.



**Tabela 2.2**  
Classificação NEMA

<b>NEMA 1</b> – Uso geral. Para uso interno, onde as circunstâncias raramente são severas.
<b>NEMA 2</b> – À prova de respingos. Proteção designada a resistir a penetração de umidade e poeira. Particularmente aplicável em ambientes com ar-condicionado, lavanderias etc., onde a condensação é prevalente. Uso em ambientes internos.
<b>NEMA 3</b> – À prova de tempo. Para uso externo. Designado a suportar as condições de exposição normais aos elementos do tempo. Os controles são montados em sistema de racks para fácil acesso. Com proteção contra chuva e vedação à prova d'água.
<b>NEMA 4</b> – À prova d'água e penetração de poeira. Projetado para uso interno, externo e ao ar livre. Deve ser resistente a condensação externa, queda d'água livre e jato d'água dirigido de uma mangueira com um bocal de 1" com vazão de 245 L/min por um período de 5 minutos a uma distância não inferior a 3,05 m.
<b>NEMA 4 X</b> – Proteção à prova d'água, penetração de poeira e resistência à corrosão. Possui as mesmas especificações de NEMA 4, porém com resistência adicional à corrosão.
<b>NEMA 5</b> – Proteção a penetração de poeira. Invólucro equipado com vedação à prova de poeiras. Apropriado para indústrias têxteis, moinhos e outras atmosferas de elevado teor de poeiras.
<b>NEMA 6</b> – Submergível. Para a operação submersa sob pressões e com tempos especificados.
<b>NEMA 7</b> – Proteção para ambientes internos potencialmente perigosos da divisão I, classe I com atmosfera de gás ou vapor; NEMA código de classe I (circuito que abre quando exposto ao ar).
<b>NEMA 8</b> – Zonas perigosas. NEMA código de classe I (circuito emergido em óleo, abre quando exposto ao ar).
<b>NEMA 9</b> – Para zonas potencialmente com atmosferas explosivas da divisão I classe II que contém atmosfera com poeira de combustíveis. NEMA código de classe II.
<b>NEMA 10</b> – À prova de explosão. Departamento de regulamentação de minas dos Estados Unidos para atmosferas explosivas (unidades são projetadas para conter explosões de gás ou de vapor e impedir a ignição na atmosfera dentro das proteções).
<b>NEMA 11</b> – Resistente a vapor de ácidos. Equipamento enclausurado fornecido em banho de óleo.
<b>NEMA 12</b> – Uso industrial. Exclui óleos, poeira e umidade. É produzido para satisfazer as exigências individuais.
<b>NEMA 13</b> – Protegido contra a penetração de óleo e poeira. Uso interno.

Tais informações e muitas outras sobre a classificação de produtos e normas podem ser observadas em <<http://www.nema.org>> e <<http://www.ip-institute.org.uk/index.html>>.

## 2.6 Relação custo-benefício

Tão importante quanto conhecer as características de cada uma das tecnologias apresentadas é tratar da questão custo-benefício de uma automatização.

O fator custo só perde em importância para o quesito segurança aos operadores e à planta. Nenhuma máquina deve ser construída sem premissas básicas de segurança aos que a utilizarão como ferramenta de trabalho. O custo aliado à eficiência desejada e necessária é o segundo fator mais importante na automação do processo. A simplicidade e a criatividade muitas vezes viabilizam projetos e geram eficiência a uma planta produtiva. Projetos sólidos e bem planejados costumam reduzir possíveis desperdícios. Cabos e elementos de controle adquiridos em excesso podem pesar no custo de um projeto, bem como a compra de, por exemplo, um CLP com 128 entradas e saídas digitais em um projeto que poderia ser atendido com oito entradas e oito saídas.

O bom-senso e o planejamento em projetos são essenciais para uma implementação bem-sucedida. Também é dispensável a utilização de um CLP, mesmo que de pequeno porte, para o simples acionamento de um motor sem necessidade alguma de interligação com outros pontos do processo. Nesse caso, um pequeno painel com duas botoeiras e um contator é suficiente tanto para a segurança do operador como para o funcionamento do motor.

Qualquer informação deve ser pesada nesse tipo de análise, como implementações futuras e complementos de projetos programados a serem realizados. Um gasto maior em um primeiro momento pode significar a redução de custo com o não descarte de equipamentos seminovos na implementação do projeto como um todo.





# Capítulo 3

## CLP: estrutura e funcionamento



Conforme definição da norma IEC 61131-1, o CLP é um “sistema eletrônico digital, desenvolvido para uso em ambiente industrial, que usa uma memória programável para armazenamento interno de instruções do usuário, que realiza funções específicas, tais como lógica, sequenciamento, temporização, contagem e aritmética, para controlar, através de entradas e saídas, vários tipos de máquinas e processos”.

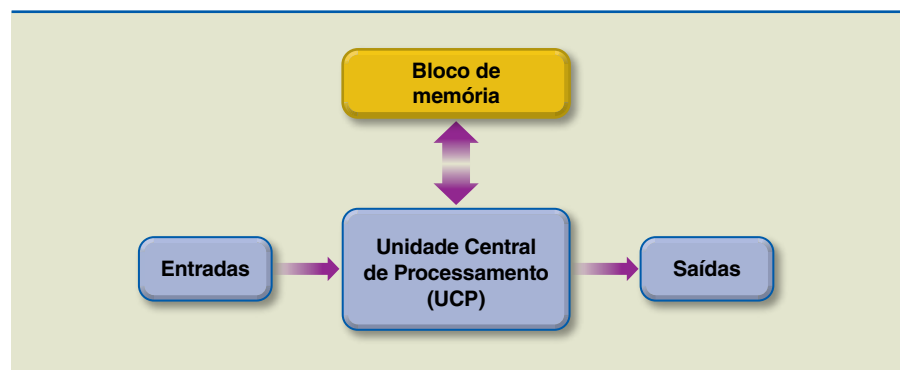
O CLP é basicamente um computador para aplicações industriais com três módulos básicos, mostrados na figura 3.1:

- Unidade central de processamento (UCP).
- Bloco de memória.
- Módulos de entrada e saída.

Adicionalmente, o CLP possui os blocos:

- Fonte de alimentação.
- Terminal de programação.
- Bloco de comunicações.
- Interface homem-máquina.

**Figura 3.1**  
Diagrama dos blocos básicos do CLP.



### 3.1 Unidade central de processamento (UCP)

A unidade central de processamento (UCP) controla e supervisiona todas as operações realizadas nos circuitos eletrônicos do CLP, por meio das instruções que estão armazenadas na memória. A comunicação interna entre a UCP, o

bloco de memória e os módulos de entrada e saída (E/S) é feita por um barramento interno sob o controle da UCP. A frequência de *clock* é responsável pela velocidade de operação do CLP e também pelo sincronismo entre os elementos do sistema da UCP. O *clock* é fornecido por um cristal externo ou oscilador RC.

Existe relação entre o tamanho da memória necessária para rodar o programa e a capacidade do sistema. Controlar um sistema pequeno, de baixa complexidade, requer um CLP simples e memória limitada; da mesma forma que controlar um sistema grande, de alta complexidade, exige um CLP maior, com mais funções e memória. Alguns modelos de CLP têm memória expansível, e outros, memória fixa.

Atualmente, todos os CLPs funcionam por meio de microprocessadores. Alguns CLPs maiores utilizam microprocessadores adicionais para controlar funções complexas e demoradas, como processamento matemático, controle PID e módulos de E/S inteligentes.

O CLP contém diversos conectores para interligá-lo a outros equipamentos (redes, terminal de programação, fonte, interface homem-máquina etc.) via cabos.

### 3.2 Bloco de memória

A memória armazena os programas desenvolvidos pelo usuário e pelo fabricante. Esse componente desempenha as funções de armazenamento:

- Do programa desenvolvido pelo fabricante (*firmware*).
- Do código do programa desenvolvido pelo usuário.
- Dos dados do programa desenvolvido pelo usuário.

A memória do CLP é organizada de maneira hierárquica: o nível superior, que fica junto ao processador, é constituído dos registradores do processador; depois, vem a memória cache; em seguida, a memória principal, que normalmente utiliza memória de acesso aleatório (RAM). Essa hierarquia continua com as memórias ROM, EPROM, EEPROM e Flash, que armazenam os programas desenvolvidos pelo fabricante.

A memória é um componente com grande variedade de tipos, tecnologias, organizações, desempenho e custos, em comparação com outros componentes do CLP. Apesar dessas diversidades técnicas, nenhuma das tecnologias de memórias satisfaz totalmente todos os requisitos de armazenamento de dados dos CLPs. Assim, um sistema de CLP típico é equipado com uma hierarquia de subsistemas de memórias, algumas internas, diretamente acessíveis pelo processador, e outras externas, acessíveis pelo processador por meio de um módulo de E/S.

Em geral, a memória RAM é usada para o desenvolvimento inicial e teste do programa, pois permite que este seja alterado com facilidade. Existe uma tendência em implementar memória RAM com CMOS (alimentada por bateria reserva recarregável ou de longa vida) que armazena as informações por mais de



um ano, em razão do baixíssimo consumo de potência, ou memórias retentivas especiais que dispensam o uso de baterias.

Em CLPs antigos, ainda com custo de memória RAM muito elevado, o programa era totalmente desenvolvido e testado, carregado em *chips* de memória PROM ou EPROM (via programador externo ou interno), que posteriormente se inverte em um soquete dentro do sistema.

Além de armazenar o programa, o CLP utiliza a memória para outras funções, como:

- Armazenar temporariamente o *status* dos canais E/S – RAM E/S.
- Armazenar temporariamente o *status* de funções internas, como temporizadores e contadores.

Como essas funções utilizam dados variáveis, é necessário o uso de memória RAM.

Um CLP pequeno ou simples geralmente possui tamanho fixo de memória devido às dimensões físicas da unidade. Para esse modelo, a capacidade de memória varia entre 300 e 1 000 instruções, dependendo do fabricante. Essa memória pode parecer insuficiente, mas estima-se que 90% de todas as tarefas de controle binário podem ser resolvidas usando menos de 1 000 instruções, de modo que sobra bastante memória para atender às necessidades da maioria dos usuários.

Um CLP maior usa módulos de memória entre 1 e 64 K, permitindo que o sistema seja expandido pela colocação de cartões de memória RAM ou PROM no bastidor. Como os custos das memórias de circuito integrado (CI) são acessíveis, os fabricantes de CLPs estão aumentando a oferta de memória de programas em seus produtos.

### 3.3 Módulos de entrada e saída (E/S)

Os módulos de entrada e saída (E/S) de um CLP conectam a interface com o sistema externo. Existem diversos tipos de módulos (analógicos, digitais e inteligentes), com número variável de entradas e saídas. Os módulos de entrada do CLP recebem sinais dos sensores e das botoeiras de campo. Os módulos de saída comunicam aos atuadores qual será a ação de controle ou sinalização.

Em CLPs de médio e grande portes, os módulos de E/S são encaixados em *slots* situados na caixa da UCP ou em *rack* de expansão separado, permitindo flexibilidade na configuração do CLP. A utilização de *slots* também possibilita que o módulo danificado seja substituído rapidamente.

Nos CLPs de pequeno porte (micro-CLPs), a quantidade de entradas e saídas é pequena e fixa (menor que oito) e geralmente se resume a sinais digitais.

Os módulos de entrada e saída têm como funções adicionais:

- Isolar eletricamente os sinais do processo dos sinais da UCP, além de prover proteção contra surtos de tensão nas entradas e saídas. Essa isolamento é necessária porque o processo automatizado trabalha com tensões elevadas e com a possibilidade de ocorrência de surtos de tensão, provenientes de descargas atmosféricas, sobretudo se os módulos operarem ao ar livre. Sem essa isolamento elétrica e a proteção contra surtos de tensão, todo o sistema de controle seria avariado.
- Os módulos de entrada fazem a aquisição, a filtragem e o armazenamento dos sinais de entrada quando solicitado pela UCP.
- Os módulos de entrada enviam os valores dos sinais de entrada, armazenados na memória do módulo de E/S, para a UCP.
- Os módulos de saída recebem da UCP os dados de saída e os armazenam em sua memória, aplicando-os às respectivas saídas quando solicitado pela UCP.
- Os módulos de E/S verificam a ocorrência de erros de comunicação módulo-UCP, falha nos sensores, falha na fonte de alimentação dos módulos, falha nos cabos de sinal etc. e os comunica à UCP.

Os principais tipos de módulos de entrada e saída são classificados em digitais, analógicos e inteligentes, descritos detalhadamente a seguir.

#### 3.3.1 Módulos de E/S digitais

Os módulos de E/S digitais operam com sinais de dois estados: ligado ou desligado. Os módulos de entrada digital são capazes de detectar e converter sinais de entrada em níveis lógicos de tensão usados no CLP; os de saída digital convertem os sinais lógicos de saída usados no CLP em sinais próprios capazes de energizar os atuadores.

Alguns módulos de E/S digitais trabalham com sinais contínuos; outros operam com sinais alternados. Para uso em CC (corrente contínua), o valor de tensão padrão adotado é de 24 V, pois permite uma relação sinal/ruído adequada para ambientes industriais. Já para os módulos de CA (corrente alternada), o padrão é de 110 ou 220 V.

Dentre os diversos tipos de dispositivos transdutores e atuadores que podem ser ligados às entradas digitais estão:

- Microchaves.
- Chaves *push-button*.
- Chaves fim de curso.
- Sensores de proximidade.
- Chaves automáticas.
- Portas lógicas.
- Células fotovoltaicas.
- Contatos de *starters* de motores.
- Contatos de relés.
- Pressostatos.
- Termostatos.
- Sensor de proximidade.
- Sensor de presença.



Como dispositivos de saídas digitais, podem-se citar:

- Anunciadores.
- Relés de controle.
- Lâmpadas.
- Portas lógicas.
- Buzinas.
- *Starters* de motores.
- Válvulas elétricas.
- Solenoides.
- Bobinas de contadores que acionarão cargas de elevadas correntes e tensões.
- Solenoides de válvulas pneumáticas.
- Válvulas solenoides.
- Disjuntores.
- Pequenos motores.

Dependendo do modelo utilizado, toda entrada e saída é alimentada por alguma fonte de tensão, na maioria das vezes, o valor de tensão é  $24 V_{CC}$  ou  $120 V_{CA}$ .

Quando em operação, aplicação de tensão à entrada resulta em nível lógico “1” (verdadeiro) no bit de memória correspondente àquela entrada física; aplicação de tensão nula à entrada, em nível lógico “0” (falso).

Se a UCP impõe o nível lógico “1” no bit de memória correspondente a uma saída digital, uma chave se fechará no módulo de saída, energizando o elemento conectado à saída.

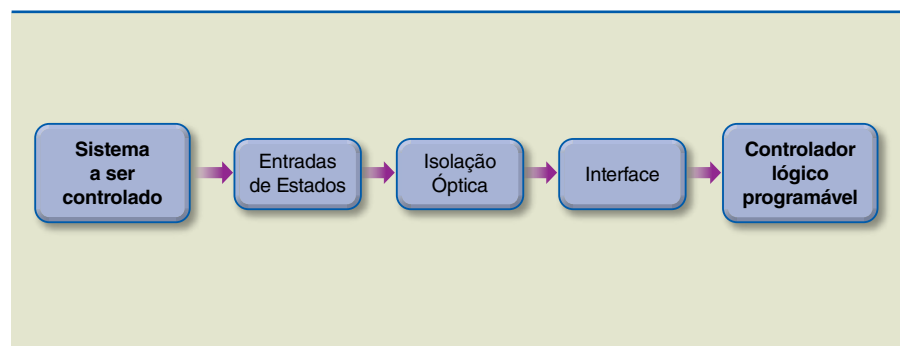
### Módulos de entrada digital em CC

A comutação executada por um transdutor digital de corrente contínua pode ser do tipo P ou N, ou seja, o acionamento pode ser lógico positivo (comum negativo) ou lógico negativo (comum positivo). Não existe vantagem alguma de um tipo sobre o outro, mas deve-se adotar apenas um deles, visando à padronização dos itens de estoque. Além disso, essa padronização evita incompatibilidades, por exemplo: termos comprado um cartão P e contarmos somente com sensores N no estoque.

A figura 3.2 ilustra o diagrama de blocos de uma entrada digital.

**Figura 3.2**

Diagrama em blocos do sinal de entrada digital.



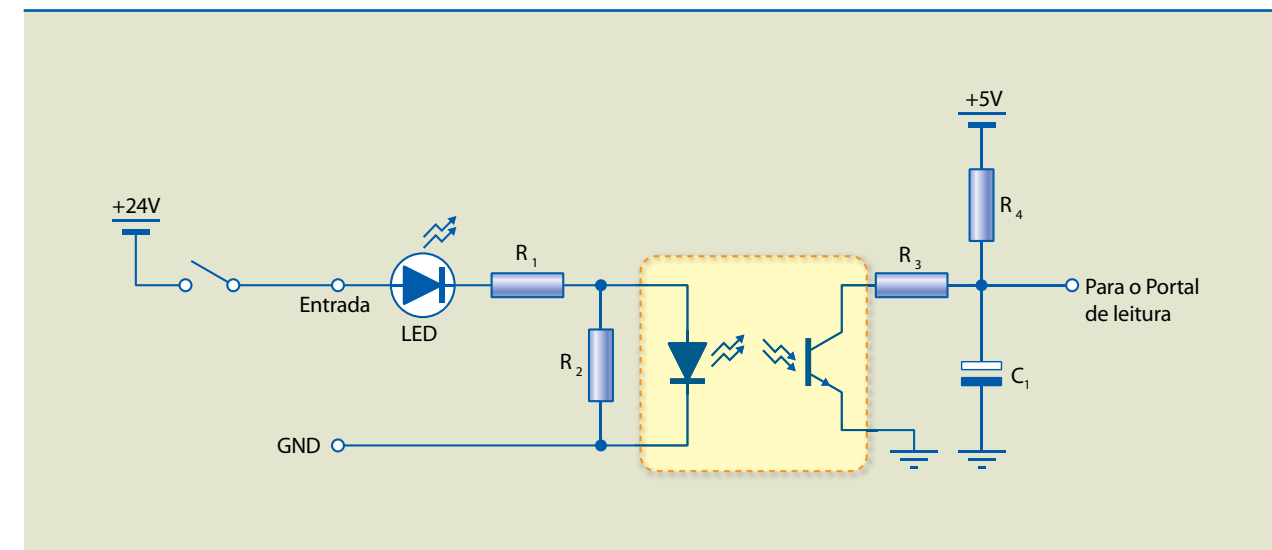
Os módulos de entrada CC convertem sinais discretos de corrente contínua ( $12 V_{CC}$ ,  $24 V_{CC}$ ,  $48 V_{CC}$ ,  $125 V_{CC}$ ) para os níveis lógicos aceitáveis pela UCP.

A figura 3.3 mostra como conectar uma chave entre o sinal +24 V e a entrada digital do tipo P. Com a chave na posição aberta, a tensão de entrada é nula. Com a chave fechada, a tensão passa a 24 V. Observe também, no circuito da figura 3.3, que existe um filtro RC formado por  $C_1$ ,  $R_3$  e  $R_4$ . A função desse filtro é amenizar a interferência de ruídos existentes na alimentação, próprios de redes elétricas industriais, de maneira que não causem acionamento indevido no CLP. No entanto, frequências maiores que 1 kHz não serão atendidas pela entrada por causa da existência desse filtro (tal limitação não existe nas entradas especiais de contadores rápidos).

Em uma entrada do tipo N, o ânodo do LED fica ligado permanentemente ao +24 V, e o terminal de entrada, ao cátodo do LED. Nessa configuração, para acionar uma entrada do tipo N, é necessário fornecer o potencial do borne negativo da fonte auxiliar ao borne da entrada.

**Figura 3.3**

Circuito de entrada digital do tipo P.



### Módulos de entrada digital em CA

De modo semelhante às entradas de corrente contínua, as entradas digitais de corrente alternada obtêm, como sinais do processo, tensões de 110 ou 220 V. Esse modelo apresenta como vantagem a possibilidade de posicionar os sensores ou chaves a maiores distâncias do CLP. Isso se deve à melhor relação sinal/ruído ao operar com tensões de 110 ou 220 V. Em geral, se os sensores estão a uma distância superior a 50 m do controlador, é necessário trabalhar com entradas CA.

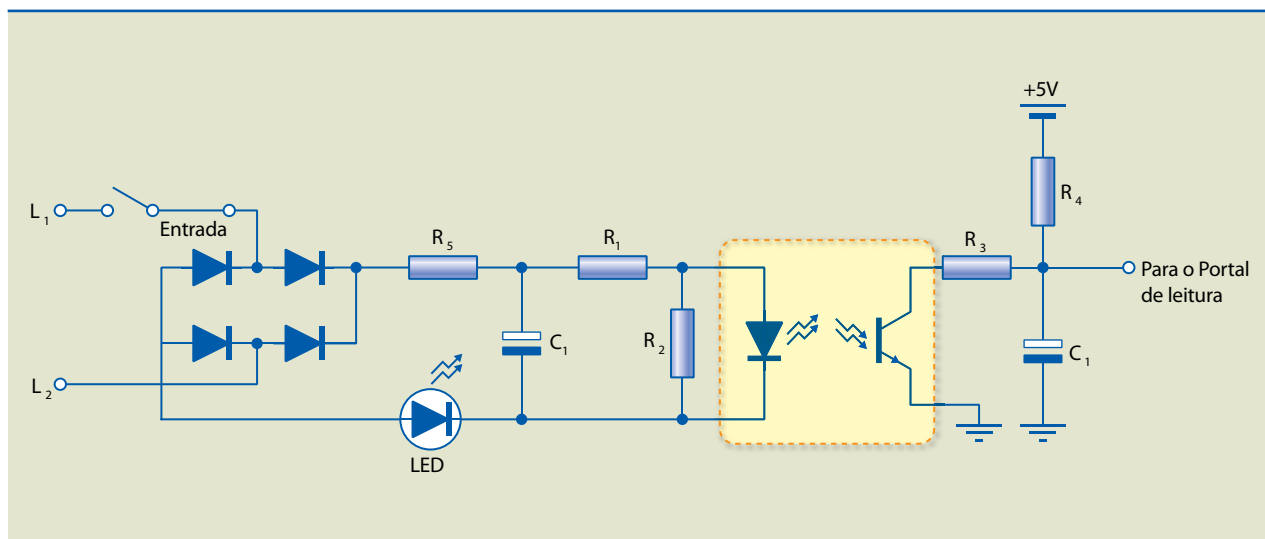
É importante lembrar que, trabalhando com níveis CA, deve-se tomar mais cuidado no que se refere à isolação geral da instalação. Um bom exemplo de aplicação de módulos de entrada CA está em usinas hidroelétricas, onde os sensores podem estar situados a distâncias de até 500 m.



Os modelos de módulos de entrada variam conforme o fabricante, porém, a maioria é constituída de blocos básicos. O circuito de entrada é composto de duas partes principais: a de alta tensão e a lógica (figura 3.4).

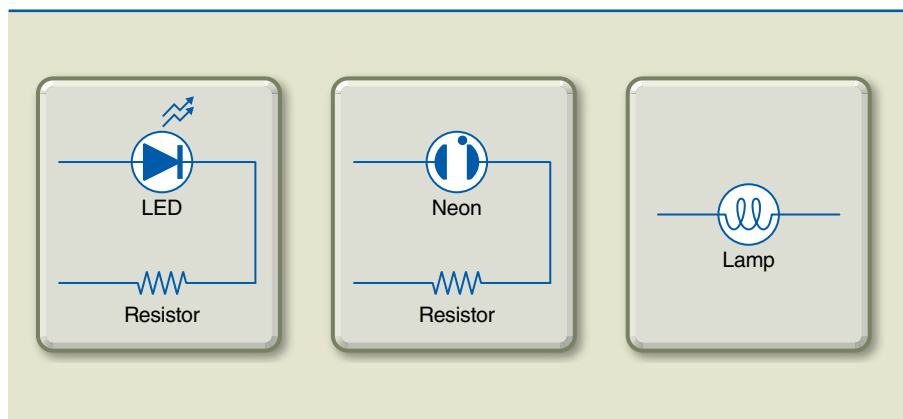
Essas duas partes são acopladas por um circuito (em geral, um acoplador óptico) que isola eletricamente a entrada da saída. A seção de entrada converte a tensão de entrada (115 ou 230  $V_{CA}$ ) de um dispositivo em um sinal de nível lógico para ser usado pelo processador. O sinal senoidal é retificado e aplicado a um filtro RC, que o torna constante. O sinal retificado passa pelo acoplador óptico e finalmente é aplicado à UCP. O filtro RC causa um atraso típico na transmissão de sinal de 10 a 25 ms, o que limita sua utilização para sinais lentos, com frequências menores que 10 Hz. Contudo, esse atraso não interfere na detecção de sinais de botoeiras, termostatos, pressostatos etc.

**Figura 3.4**  
Circuito de entrada digital em CA.



A maioria dos módulos de entrada contém indicações visuais do estado lógico de suas entradas em LED ou lâmpada (figura 3.5). Essa sinalização visual é muito útil no período de testes para detectar falhas no programa. Se o programa não está funcionando, é fácil saber se o sinal realmente está sendo aplicado à entrada digital, sem a necessidade de utilizar um voltímetro.

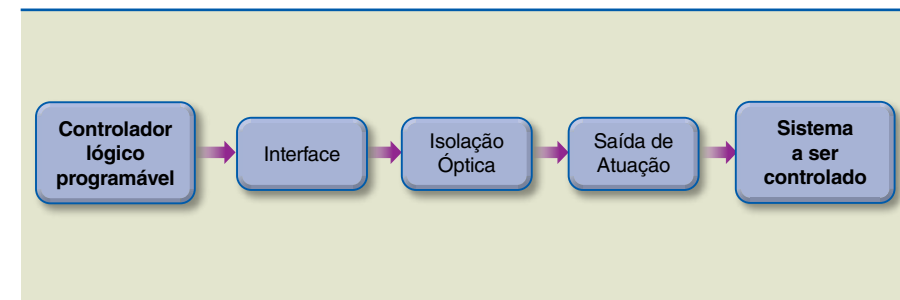
**Figura 3.5**  
Indicações visuais no módulo de entrada.



### Módulos de saída digital em CA

Esses módulos são geralmente usados para acionar cargas (bobinas de contatores, relés etc.) ligadas em corrente alternada com tensões de alimentação entre 90 e 240  $V_{CA}$ . Essas saídas convertem os sinais lógicos usados no CLP em sinais característicos capazes de energizar os atuadores. Em geral, possuem capacidade de corrente de até 2 A.

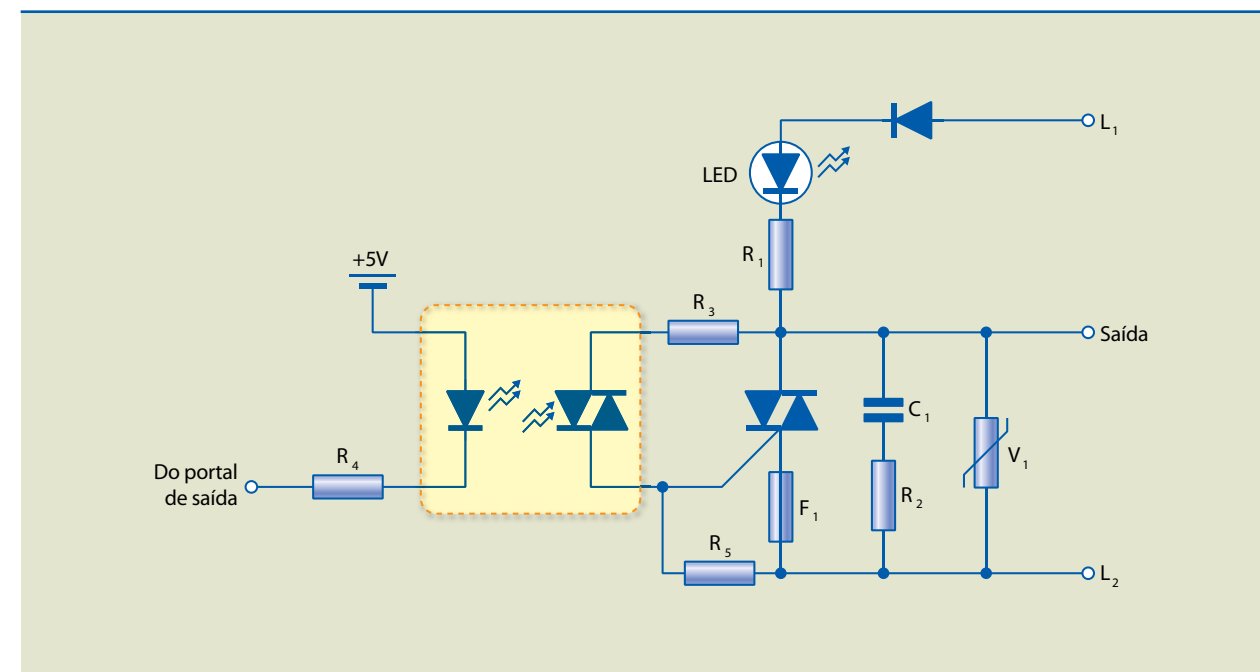
O diagrama em blocos da figura 3.6 descreve a operação de grande parte dos módulos de saída em corrente alternada disponíveis no mercado.



**Figura 3.6**  
Diagrama em blocos do sinal de saída digital.

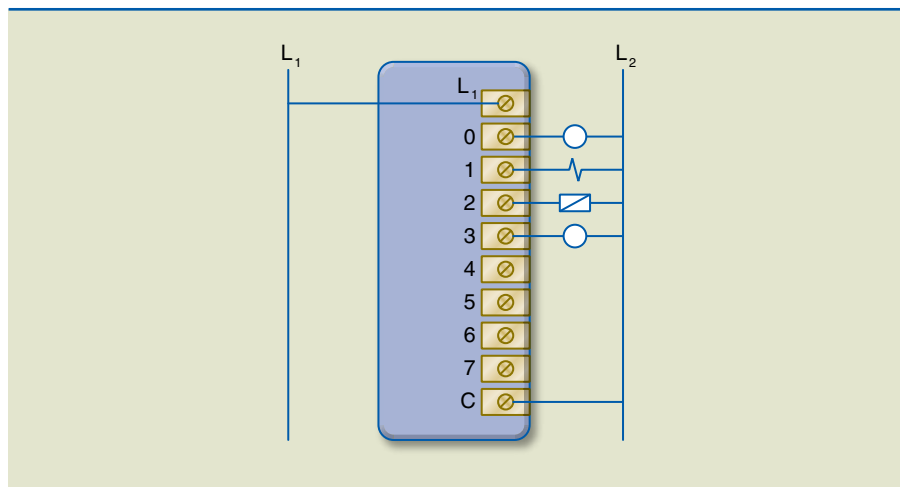
Se a saída da UCP fica energizada, o estado passa para a etapa de potência e instantaneamente dispara uma chave, nesse caso um relé ou um TRIAC. No TRIAC, apresentado no circuito da figura 3.7, a carga será ligada entre o terminal  $L_1$  e o terminal "Saída". Os terminais  $L_1$  e  $L_2$  correspondem à rede de 110 ou 220  $V_{CA}$  destinada ao acionamento dos dispositivos de saída. Nesse circuito, podemos notar a existência de um varistor ( $V_1$ ) que protege a saída contra surtos de tensão, um circuito RC que protege contra disparos indevidos e um TRIAC isolado por acoplador óptico.

**Figura 3.7**  
Circuito de saída digital em corrente alternada com TRIAC.



A figura 3.8 mostra um esquema de ligação em um módulo de saída CA.

**Figura 3.8**  
Diagrama de saída para módulo digital CA.



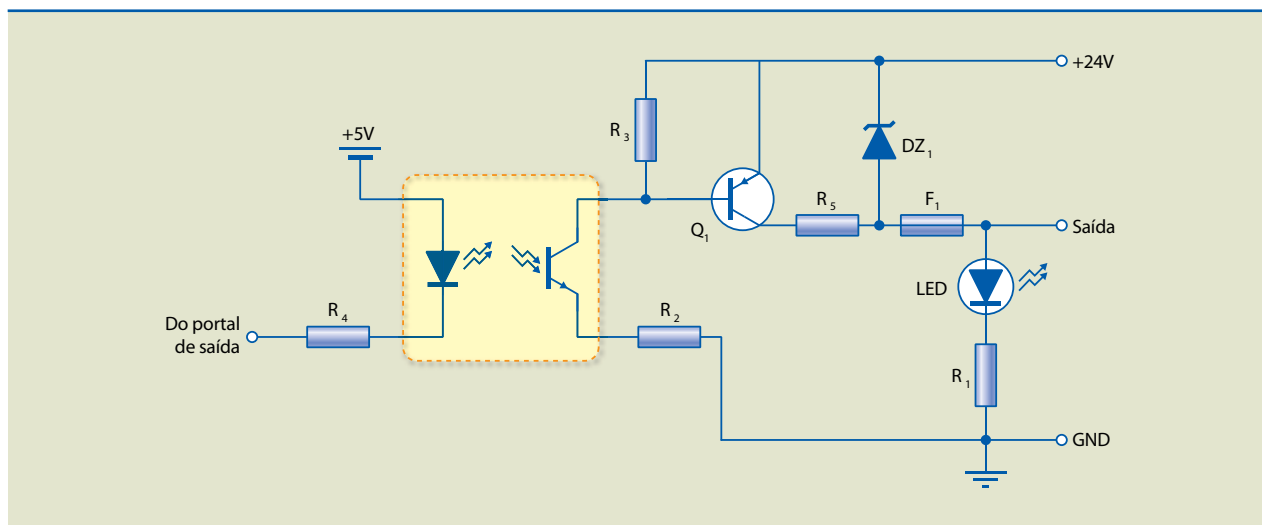
Os módulos de saída também possuem LED para indicar o estado da lógica. Alguns módulos têm um fusível na parte de potência para proteger o elemento de chaveamento (TRIAC ou relé) contra sobrecorrentes ou curtos na carga.

**Módulos de saída digital em CC**

Os módulos de saída em CC têm funcionamento similar aos módulos de saída em CA, porém, foram projetados para atuar em dispositivos de corrente contínua. Seus circuitos de chaveamento da carga são diferentes; eles utilizam FETs (transistores de efeito de campo) ou transistores bipolares para o chaveamento. Essas saídas digitais em CC podem ser de dois tipos:

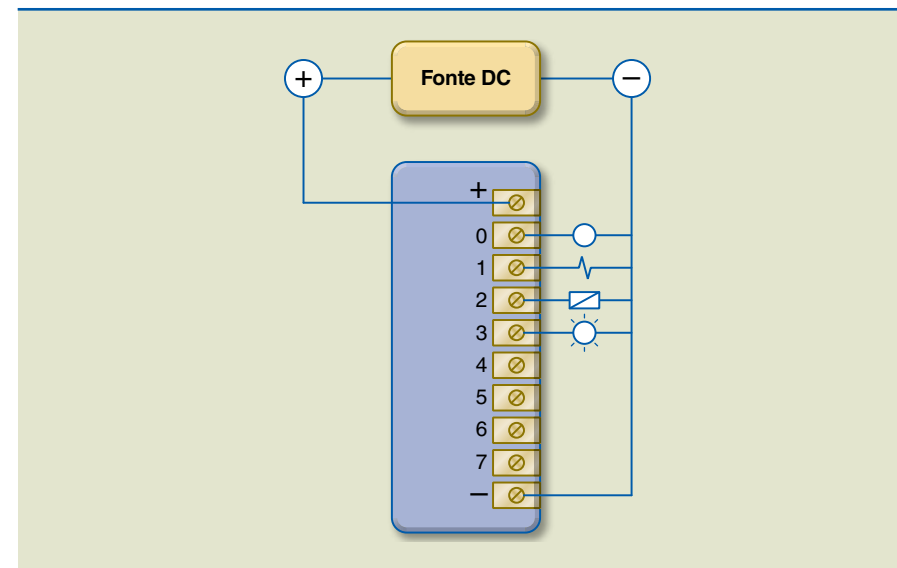
- **Tipo P** – Fornece potencial positivo e a carga deve ser ligada entre o borne positivo da saída e o negativo da fonte de alimentação (figura 3.9).
- **Tipo N** – A carga deve ser ligada entre o potencial positivo da fonte de alimentação e o borne de saída.

**Figura 3.9**  
Circuito de saída digital do tipo P.



A figura 3.10 mostra um esquema de ligação em um módulo de saída CC.

**Figura 3.10**  
Diagrama de saída para módulo digital CC.

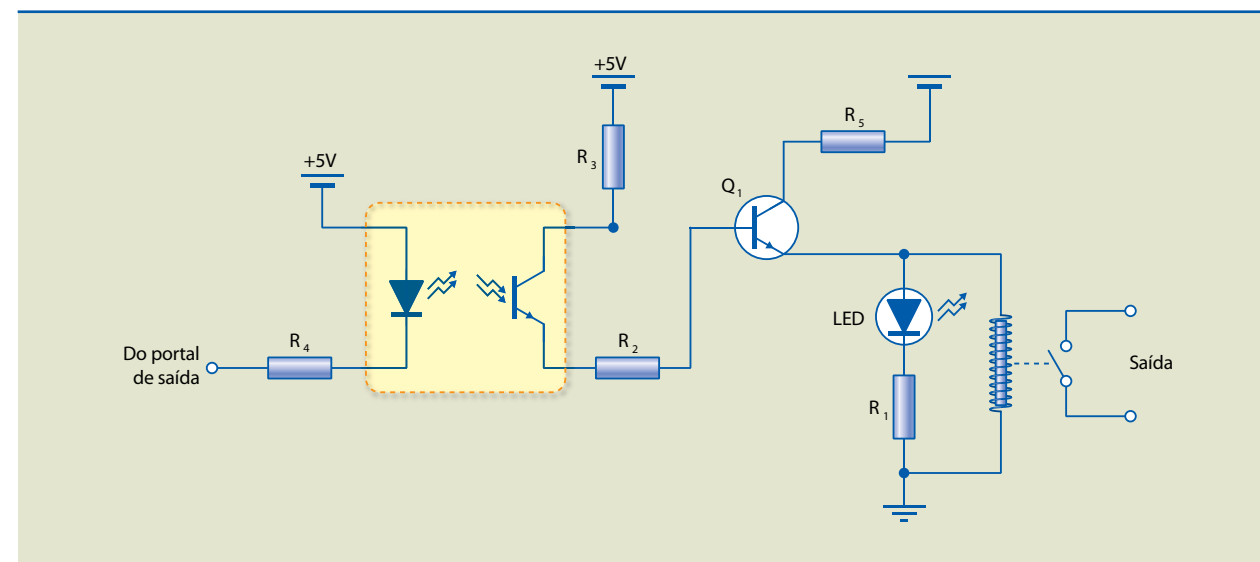


**Módulos de saída a relé**

Os módulos de saída a relé são utilizados por sua versatilidade em comutar sinais, podendo ser aplicados a cargas de corrente alternada e de corrente contínua (figura 3.11).

Esses módulos apresentam desgaste em seus contatos elétricos em razão da circulação de corrente elétrica em seus terminais (arco voltaico) e desgaste mecânico proporcional ao número de chaveamentos realizados. Para aumentar sua vida útil, deve-se utilizá-los como contatos auxiliares, ou seja, intercalar entre a saída do CLP e a carga outro relé de maior potência ou uma chave estática, poupando os contatos do relé interno do módulo do CLP. Um inconveniente desse modelo é que as saídas a relé são mais lentas que as saídas a transistor.

**Figura 3.11**  
Circuito do módulo de saída a relé.



### Módulos de saída TTL

Os módulos de saída TTL permitem interfacear dispositivos externos compatíveis com esses níveis e dispositivos alimentados por  $5 V_{CC}$ , como LED de sete segmentos e circuitos integrados digitais. Como os demais módulos de saída, requerem alimentação externa para atuar na carga.

### 3.3.2 Módulos de E/S analógicas

A alta escala de fabricação, o baixo custo e a disponibilidade de circuitos integrados possibilitaram que na estrutura do CLP fosse incluída grande capacidade de processamento de dados. Isso também propiciou a expansão da capacidade de E/S e a implementação de módulos analógicos e digitais no CLP. Os módulos de E/S analógicas permitem controlar e monitorar correntes e tensões compatíveis com muitos sensores, válvulas, acionadores de motores e instrumentos de processo.

Apresentam-se a seguir os dispositivos típicos que são interfaceados com os módulos de E/S analógicas.

a) Dispositivos com módulos de entrada analógica:

- transdutores diversos;
- transmissores eletrônicos;
- instrumentos analíticos;
- potenciômetros.

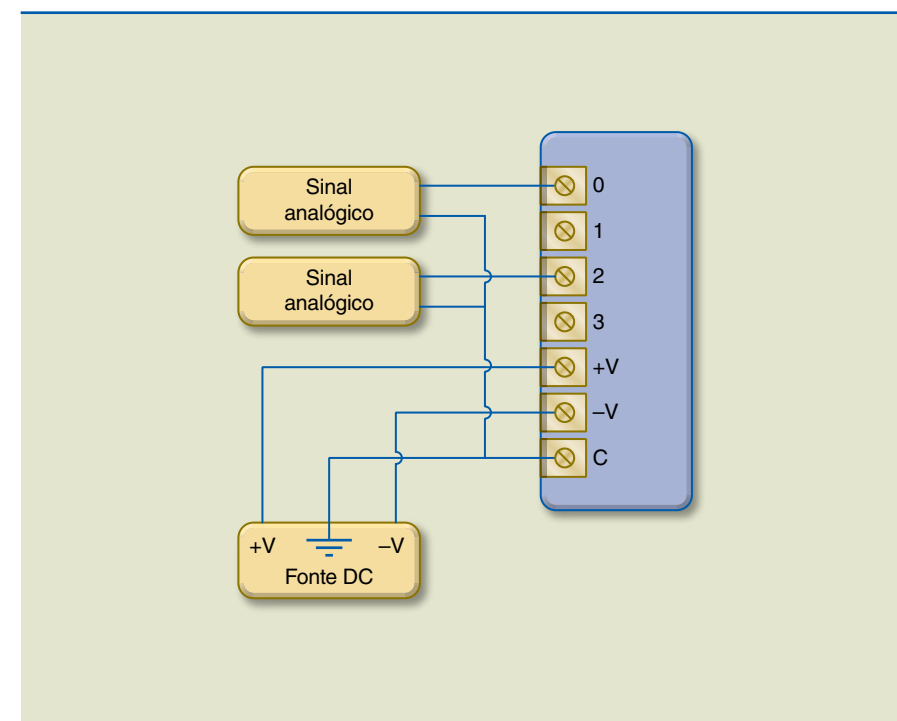
b) Dispositivos com módulos de saída analógica:

- indicadores analógicos;
- *drives* de motores;
- registradores;
- transdutores I/P;
- válvulas elétricas.

As interfaces analógicas normalmente estão disponíveis para diversos tipos de padrão, como operação unipolar e diferencial de corrente ou de tensão.

### Módulos de entrada analógica

Os dispositivos de campo fornecem sinais (de tensão ou corrente elétrica) para os módulos de entrada analógica, que, por meio de circuito (conversor analógico-digital – A/D), converte esses sinais analógicos em uma informação digital. Esses sinais analógicos de entrada são inseridos no CLP na forma binária e armazenados em uma posição de memória para uso posterior. Os módulos de entrada possuem alta impedância, possibilitando a interface com os dispositivos de campo sem causar sobrecarga ou afetar sua operação. A interligação com os dispositivos de campo é feita por cabos blindados, que reduzem os ruídos elétricos de fontes externas (figura 3.12), além de possuírem filtro e isolamento elétrica para proteger os módulos.



**Figura 3.12**

Diagrama de ligação para módulo analógico.

Os padrões usuais para os módulos de entrada analógica são apresentados na tabela 3.1.

Tensão (V)	Corrente (mA)
1 a 5	4 a 20
0 a 5	0 a 20
-10 a +10	-20 a +20
0 a 10	

**Tabela 3.1**

Padrões de módulos de entrada analógica

Os módulos são projetados para receber quatro, oito ou dezesseis entradas analógicas. A conversão analógico-digital é feita em 12 ou 14 bits codificados em binário. A configuração de entrada e a operação bipolar ou unipolar podem ser selecionadas por *jumpers* ou por *software*.

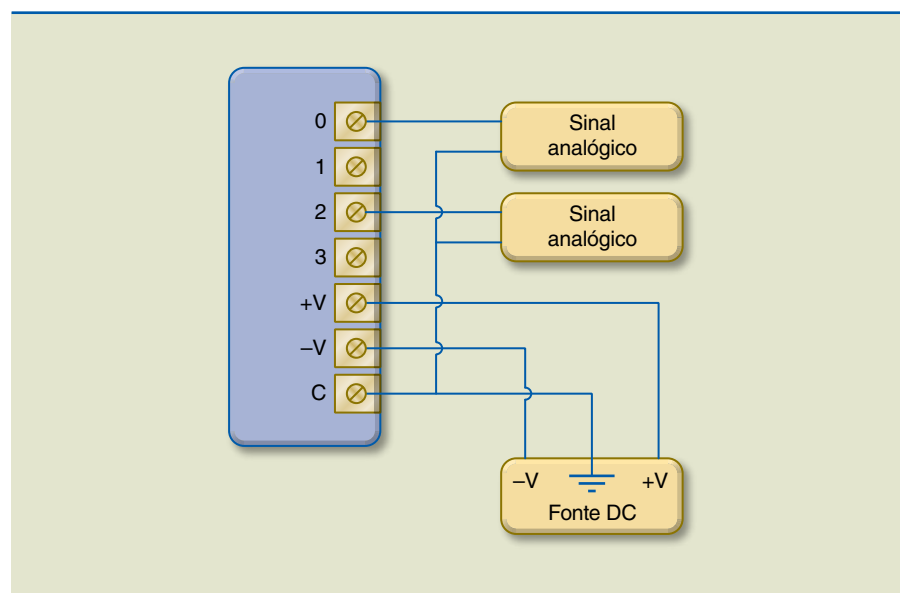
### Módulos de saída analógica

Os módulos de saída analógica possuem um conversor digital-analógico (D/A) que recebe os valores numéricos na forma binária alocados nos registros do CLP e os converte em grandezas analógicas de tensão ou corrente proporcionais à informação digital e as envia para a saída. A resolução do conversor é de 12 ou 14 bits. Uma saída analógica tem como função fornecer sinal analógico para comandar atuadores analógicos, conforme ilustrado na figura 3.13. Todas as outras características são semelhantes às do módulo de entrada analógica.



**Figura 3.13**

Diagrama de ligação para módulo analógico.



### 3.3.3 Módulos de E/S inteligentes

As necessidades de E/S digitais de um CLP são atendidas em 90% dos casos pelos módulos convencionais. Em algumas situações, como na contagem de pulsos de elevada frequência, as entradas convencionais são extremamente lentas e acionadas pela UCP, que, além de ler as entradas digitais, precisa gerenciar todos os módulos de E/S e executar o programa. Como a UCP não pode ficar permanentemente contando pulsos, acaba perdendo a contagem de alguns. Essa deficiência pode produzir erros inaceitáveis – por exemplo, se estivermos controlando a posição de uma máquina pela contagem de pulsos gerados por um *encoder*. Nesse caso, recomendam-se os módulos contadores rápidos, que possuem um microprocessador dedicado em tempo integral à contagem de pulsos. Quando a UCP precisar saber o resultado da contagem, basta solicitá-lo ao módulo contador.

Outro exemplo inclui os condicionadores de sinal, como uma entrada de termopar, em que, além da amplificação do sinal e de sua leitura, deve-se efetuar a linearização, para que o dado represente a grandeza física medida. Os módulos contadores rápidos podem realizar todas as tarefas necessárias para esses procedimentos, aliviando a carga computacional da UCP principal.

Os fabricantes de CLPs oferecem grande variedade de módulos especiais. Os mais comuns são:

- **Módulo de entrada termopar** – Foi projetado para aceitar diretamente as entradas de um termopar e fazer a compensação de junta fria, linearização da medição e detecção de erros. Seu funcionamento é semelhante aos módulos de entrada analógica discutido anteriormente, com exceção do baixo nível de sinal (mV). Os dados obtidos por esse módulo podem ser utilizados pelo CLP para fazer controle, indicação, registro, alarme ou intertravamento.

- **Módulo de entrada PT100** – É utilizado em medição de temperatura a partir de um PT100, que é um resistor cuja resistência varia linearmente com a temperatura.
- **Módulo de motor de passo** – Gera o trem de pulsos a ser aplicado às bobinas de um motor de passo, de modo a controlar sua posição, sua velocidade ou sua aceleração, de acordo com o estabelecido pelo programa do usuário.
- **Módulo de controle de malha** – Realiza o controle de malha fechada com algoritmos PID. O processamento do controle PID é feito pelo próprio módulo, aliviando o trabalho da UCP. Os algoritmos PID encontrados no CLP (rodado na UCP) às vezes deixam a desejar quanto às necessidades de tempo e precisão exigidas para controle de sistemas rápidos.
- **Módulo de contagem rápida.**
- **Módulo para medição de parâmetros elétricos** – Fator de potência, tensão, corrente, energia etc.
- **Módulo de comunicação com redes.**

## 3.4 Fontes de alimentação

Nos CLPs, as fontes de alimentação podem ser internas ou externas. As internas podem ser encaixadas em *slot* ou montadas na caixa do CLP. Algumas fornecem saída de  $24 V_{CC}$ , de baixa potência ( $\sim 2 A$ ), destinada à alimentação dos módulos de E/S, sensores e atuadores. Se a exigência de potência dos módulos for maior, o projetista deverá necessariamente incluir uma fonte externa.

A International Electrotechnical Commission (IEC – Comissão Internacional de Eletrotécnica) tem especificado tensões padronizadas para fontes de alimentação, entradas e saídas digitais dos CLPs. A tabela 3.2 mostra tais tensões.

Tensão	Recomendações	
	Fonte	Sinais E/S
$24 V_{CC}$	Sim	Sim
$48 V_{CC}$	Sim	Sim
$24 V_{ca}$	Não	Não
$48 V_{ca}$	Não	Não
$120 V_{ca}$	Sim	Sim
$230 V_{ca}$	Sim	Sim
$400 V_{ca}$	Sim	Não

**Tabela 3.2**

Valores e faixas de operação

Notas:

- a) A tolerância para as tensões CC é de  $-15$  a  $+20\%$ .





b) A tolerância para as tensões CA é de  $-15$  a  $+10\%$ . As tensões CA são dadas em valor eficaz (RMS).

c) Consultar a norma para CLPs para notas e valores especificados adicionais.

Normalmente utilizam-se fontes chaveadas, alimentadas com tensão AC (90 a 240 V) ou DC (18 a 36 V). Outra preocupação é quanto às tensões fornecidas para o CLP, pois, além do padrão  $5 V_{CC}$ , alguns modelos também alimentam os módulos analógicos com  $+12 V_{CC}$  e  $-12 V_{CC}$ .

O consumo é, por fim, outro fator que tem de ser analisado, pois, em função da quantidade de pontos a serem utilizados, há condições de dimensionar a fonte. Portanto, a fonte deve ser escolhida por último, pesquisando com o fabricante qual o modelo de fonte mais adequado ao processo que se deseja automatizar. Para os CLPs de baixo custo, a fonte é incorporada ao módulo da UCP, visando a diminuir encaixes e, em consequência, barateando o produto. Nesse caso, a preocupação se resume à alimentação do conjunto, uma vez que tanto as tensões como as potências envolvidas serão logicamente compatíveis com o produto.

### 3.5 Funcionamento

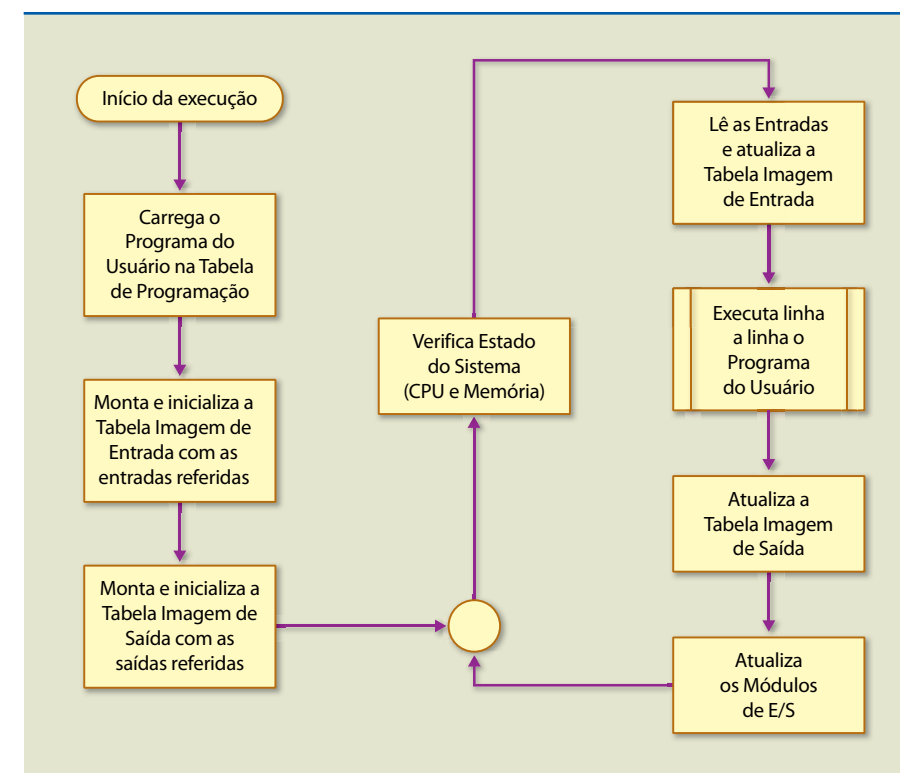
O funcionamento do CLP baseia-se em um conjunto de instruções conhecido como programa, desenvolvido para atender às rotinas do usuário. Os sinais dos sensores ou transdutores acoplados à máquina ou equipamento são aplicados às entradas do controlador. Em cada ciclo, denominado *scan* ou varredura, há três etapas:

- **Etapas de entrada** – Todos os sinais vindos do meio externo e aplicados às entradas são lidos e transferidos para uma área da unidade de memória interna do CLP, chamada tabela imagem da entrada.
- **Etapas de programa** – O programa é executado, usando os sinais adquiridos durante a etapa de entrada e decidindo quais serão os valores dos sinais de saída digitais e analógicos. Esses valores são guardados em uma região de memória denominada imagem de saída. Durante a execução do programa, os sinais de saída não são alterados, a menos que sejam ativadas entradas de elevada prioridade. Tais entradas são utilizadas para emergências, que não podem esperar o término do *scan*. Nesse caso, interrompemos o trabalho do CLP, rodando um programa que atende à interrupção. Terminado o atendimento à interrupção, retornamos ao programa principal.
- **Etapas de saída** – Todos os valores de saída estabelecidos pelo programa e armazenados na imagem de saída são aplicados aos módulos de saída.

Terminadas as três etapas, reinicia-se o ciclo de *scan*, com a etapa de entrada (operação cíclica). Em algumas situações, por exemplo, ao utilizar controladores PID, torna-se necessário fazer as aquisições de sinais (etapa de entrada) em intervalos fixos. Nesse caso, ao terminar a etapa de saída, o CLP fica esperando o instante predefinido para reiniciar o *scan* (operação periódica).

Para realizar a tarefa de controle, o processador do CLP simplesmente executa passo a passo o roteiro de instruções contido na tabela de programação que o

usuário elaborou. O processo de execução segue aproximadamente a rotina mostrada no fluxograma da figura 3.14.



**Figura 3.14**

Fluxograma de funcionamento do CLP.



# Capítulo 4

## Linguagem de programação

No capítulo 1, foram apresentadas as linguagens de programação de um enfoque histórico. Agora serão discutidos alguns detalhes adicionais sobre as linguagens mais empregadas e será ampliada a discussão sobre a tentativa de padronização das linguagens pela norma IEC. Por sua importância e popularidade, a linguagem Ladder terá o capítulo 5 inteiramente dedicado a ela.

## 4.1 Norma IEC 61131

Em 1992, a International Electrotechnical Commission (IEC – Comissão Internacional de Eletrotécnica) publicou a primeira edição da norma IEC 61131, com o objetivo de estabelecer padrões para os controladores programáveis. Essa norma foi dividida em vários capítulos, possibilitando a definição de critérios para cada um dos tópicos relacionados aos CLPs.

Os capítulos são os seguintes:

- 61131-1 – Informações gerais sobre os CLPs.
- 61131-2 – Requisitos de *hardware*.
- 61131-3 – Linguagens de programação.
- 61131-4 – Guia de orientação ao usuário.
- 61131-5 – Comunicação.

São abordados aqui os padrões de linguagens de programação de CLPs apresentados na IEC 61131-3. Vale ressaltar que, mesmo que se trate de um capítulo à parte, a IEC 61131-3 especifica as semânticas e sintaxes para as linguagens de programação dos controladores definidos na IEC 61131-1 no que se refere aos conceitos gerais. A IEC 61131-5 trata da comunicação de dados internos, uma vez que esses tópicos estão indiretamente interligados.

A primeira edição da IEC 61131-3 foi publicada em dezembro de 1993, e a segunda, em 2003.

Essa norma define que, para um CLP estar de acordo com os padrões por ela estabelecidos, deve possuir ao menos duas linguagens gráficas e duas linguagens de texto para sua programação. Assim, é consenso entre os atuais fabricantes trabalhar com as seguintes linguagens:

- Textuais:
  - IL – *instruction list* ou lista de instruções.
  - ST – *structured text* ou texto estruturado.
- Gráficas:
  - LD – *Ladder diagram* ou diagrama Ladder.
  - FBD – *function block diagram* ou diagrama de blocos de funções.

Um quinto modo de programar alguns equipamentos de mercado é o SFC – *sequential function chart* ou sequenciamento gráfico de funções, que possui elementos para organizar programas de maneira sequencial e permite também o controle paralelo de processos.

Entre essas linguagens de programação, alguns parâmetros são definidos para que realmente haja certa compatibilidade entre os equipamentos. Tais parâmetros podem ser definidos como:

a) **Dados** – A norma prevê os seguintes tipos de dados:

- Grupo de bits: grupo de valores binários (*on/off*).
  - BOOL: 1 bit.
  - BYTE: 8 bits.
  - WORD: 16 bits.
  - DWORD: 32 bits.
  - LWORD: 64 bits.
- Inteiros: números inteiros e reais.
  - SINT: curto (1 byte).
  - INT: inteiro (2 bytes).
  - DINT: duplo inteiro (4 bytes).
  - LINT: longo inteiro (8 bytes).
  - U: não definido (adicionar um caractere U para o tipo de inteiro a ser utilizado).
- Real: ponto flutuante, ou seja, considera fracionários na utilização do número conforme a IEC 559 (1982).
  - REAL: 4 bytes.
  - LREAL: 8 bytes.
- Tempo: duração de *timers* e processos.
- Data e hora do dia.
  - DATE: data do calendário.
  - TIME\_OF\_DAY: hora local.
  - DATE\_AND\_TIME: data e hora local.



- *String*: caracteres que podem ser expostos em cotas únicas – normalmente para a transmissão de caracteres ASCII (*american standard code for information interchange*) para outros dispositivos.
  - WSTRING: permitir o envio de vários *strings*.
  - ARRAYS: múltiplos valores armazenados na mesma variável.
  - SUB RANGES: definir limites de valores para a entrada ou para a saída de dados – por exemplo, sinais de 4 a 20 mA.

b) **Variáveis** – Podem ser do tipo:

- Global: serve a todo o programa e não necessariamente só a uma parte dele.
- Local: serve somente a uma parte do programa.
- *I/O mapping*: mapeamento de todas as entradas e saídas em relação a posições de memória predefinidas.
- *External*: mapeamento definido exclusivamente como pontos de entrada e saída de dados.
- *Temporary*: usados momentaneamente durante a execução de parte do programa.

c) **Configuração** – Recursos de *hardware* e vínculos específicos para o processamento dos dados e dos programas.

- Recursos: reserva de memória ou índices de processamento para determinada parte do programa.
- Tarefas: podem seguir paralelas, sendo executadas simultaneamente pela UCP.
- Programas: podem ser executados ciclicamente, a cada determinado período ou quando ocorrer certo evento.

d) **Organização das unidades de programas** – Definidos pelas funções básicas, blocos básicos e possibilidade de criação de funções e blocos de acordo com a necessidade da programação.

- Funções-padrão como: ADD, SQRT, SIN, COS, GT, MIN, MAX, AND, OR, entre outras.
- Funções customizadas: campo no qual o programador pode criar funções ou utilizar mais de uma função preexistente para a criação de outra função em sua programação.
- Blocos de funções: padrões iguais aos apresentados nas funções, só que em linguagem de bloco.
- Blocos customizados: campo para a elaboração ou utilização de mais de uma função preexistente na criação de blocos. Podem ser compostos também por blocos comercializados por outros fabricantes ou empresas especializadas.
- Programas: programas e sub-rotinas específicos. Podem ser armazenados em funções ou blocos criados pelo programador e utilizados mais de uma vez na atual aplicação ou posteriormente em outros programas.

e) **Links externos** – Abrem espaço para o capítulo da IEC 61131-5, que trata especificamente dos formatos de comunicação e das facilidades que o padrão determina como necessários para a compatibilidade com a norma.

Vale ressaltar que essas normas possibilitaram que fabricantes de outros tipos de *hardware* compatibilizassem seus produtos. Hoje existem fabricantes de SoftPLCs, que nada mais são do que computadores que podem ser programados para atender a controles lógicos, obedecendo às normas definidas pela IEC. Trata-se de tecnologia muito recente, que ainda está sendo avaliada e certificada, porém, já é utilizada em alguns processos de automação de máquinas de pequeno porte.

Uma vez apresentadas as informações sobre os parâmetros que a IEC 61131-3 determina que um fabricante de *hardware* e *software* deve disponibilizar aos usuários, vamos ver agora as linguagens de programação definidas pela norma.

## 4.2 IL – lista de instruções

É basicamente a transcrição do diagrama de relés (Ladder), ou seja, a passagem de uma linguagem gráfica para uma linguagem escrita. Essa etapa foi importante nos primórdios do CLP, pois não existiam terminais gráficos como conhecemos atualmente, que permitem desenhar o diagrama Ladder na tela, usando o *mouse*. Antigamente os terminais de vídeo e os *displays* dos terminais de programação eram alfanuméricos; por isso, o programador precisava projetar o diagrama Ladder no papel e depois convertê-lo para a linguagem IL. Um compilador se encarregava de traduzir o IL para a linguagem de máquina (Assembler) do processador utilizado no CLP.

A tabela 4.1 apresenta as instruções mais comuns empregadas nessa linguagem.

**Tabela 4.1**

Lista de comandos na linguagem IL contidas na IEC 61131-3

Operador	Modificador	Tipo de dados	Descrição
LD	N	Diversos	Carrega valor do resultado
ST	N	Diversos	Armazena o resultado no local definido
S, R		BOOL	Seta ou resseta um valor ( <i>latch</i> ou <i>flip-flop</i> )
AND, &	N, (	BOOL	Lógica booleana AND
OR	N, (	BOOL	Lógica booleana OR
XOR	N, (	BOOL	Lógica booleana OR EXCLUSIVE
ADD	(	Diversos	Soma matemática
SUB	(	Diversos	Subtração matemática
MUL	(	Diversos	Multiplicação matemática
DIV	(	Diversos	Divisão matemática
GT	(	Diversos	Compara maior que (>)

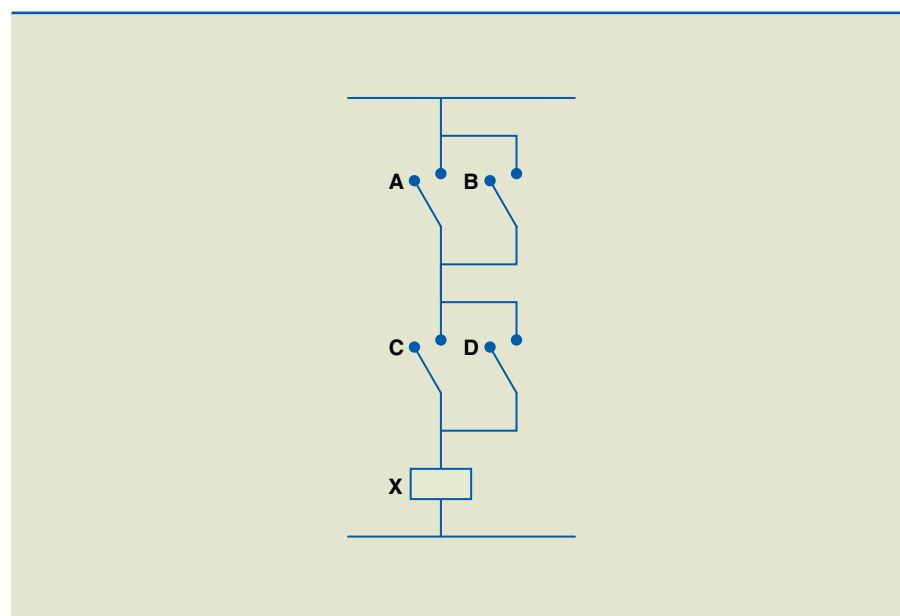


Operador	Modificador	Tipo de dados	Descrição
GE	(	Diversos	Compara maior ou igual a ( $\geq$ )
EQ	(	Diversos	Compara igual a (=)
NE	(	Diversos	Compara diferente ( $\neq$ )
LE	(	Diversos	Compara menor ou igual a ( $\leq$ )
LT	(	Diversos	Compara menor que (<)
JMP	C, N	LABEL	Salta para a linha de programa
CAL	C, N	NAME	Chama sub-rotina de acordo com nome
RET	C, N		Retorna da sub-rotina chamada
)			Conclui valor chamado

Dado o diagrama de comandos elétricos da figura 4.1, que já está na linguagem Ladder (exceto pela simbologia), vamos transcrevê-lo para a linguagem IL.

**Figura 4.1**

Exemplo de diagrama de comandos elétricos.



Analisando a figura, fazemos a seguinte leitura: havendo continuidade elétrica de A ou B e também de C ou D, X deve ser acionado.

Podemos traduzir essa lógica em lista de instruções segmentando o problema:

LD A – Carrega o valor de A em um acumulador.

LD B – Carrega o valor de B em um acumulador.

OR B – Executa a lógica booleana OR entre A e B e armazena o resultado em B.

LD C – Carrega o valor de C em um acumulador.

LD D – Carrega o valor de D em um acumulador.

OR D – Executa a lógica booleana OR entre C e D e armazena o resultado em D.

AND B – Executa a lógica booleana AND entre B e D (últimos acumuladores gravados) e armazena o resultado em B.

ST X – Armazena o valor de B em X.

Percebe-se que tal lógica utiliza uma única instrução por linha de programação, o que dificulta a elaboração de grandes programas. Desse modo, faz-se necessário um controle muito eficiente na utilização dos registradores e respectivas interfaces de entrada e saída para não haver falhas durante a confecção do programa.

### 4.3 ST – texto estruturado

É uma linguagem mais elaborada, considerada de alto nível, que usa o princípio de criação de sentenças para definir e informar ao CLP qual a lógica necessária em determinado ponto. Como possibilita a utilização de mais de uma instrução por linha, agiliza e facilita a tarefa dos programadores em projetos mais complexos.

Com estrutura similar à de linguagens de programação, como o C++ e o Pascal, permite o uso de comandos específicos para a definição de laços de controle, ou seja, funções ou operações lógicas que devem ser executadas até que determinado evento ocorra ou que determinada contagem seja atingida (funções REPEAT-UNTIL, DO-WHILE, entre outras).

Possibilita a utilização de instruções condicionais, referindo-se a reações preestabelecidas do programa para o caso de certos eventos ocorrerem, desde que previamente considerados (funções IF-THEM-ELSE, CASE), e também, por ser uma linguagem mais rica, o emprego de equações trigonométricas (SIN – função seno) e matemáticas (SQRT – raiz quadrada). Mesmo sendo uma linguagem mais fácil de ser compreendida e escrita, ainda demanda mão de obra especializada para a confecção e manutenção de programas.

Levando em conta o exemplo apresentado na figura 4.1, podemos definir a lógica de programação em linguagem estruturada da seguinte forma:

$X = (A \text{ OR } B) \text{ AND } (C \text{ OR } D)$

Ou seja, X é o resultado da operação booleana AND de dois resultados distintos: lógica OR entre A e B e lógica OR entre C e D.

### 4.4 FBD – diagrama de blocos funcionais

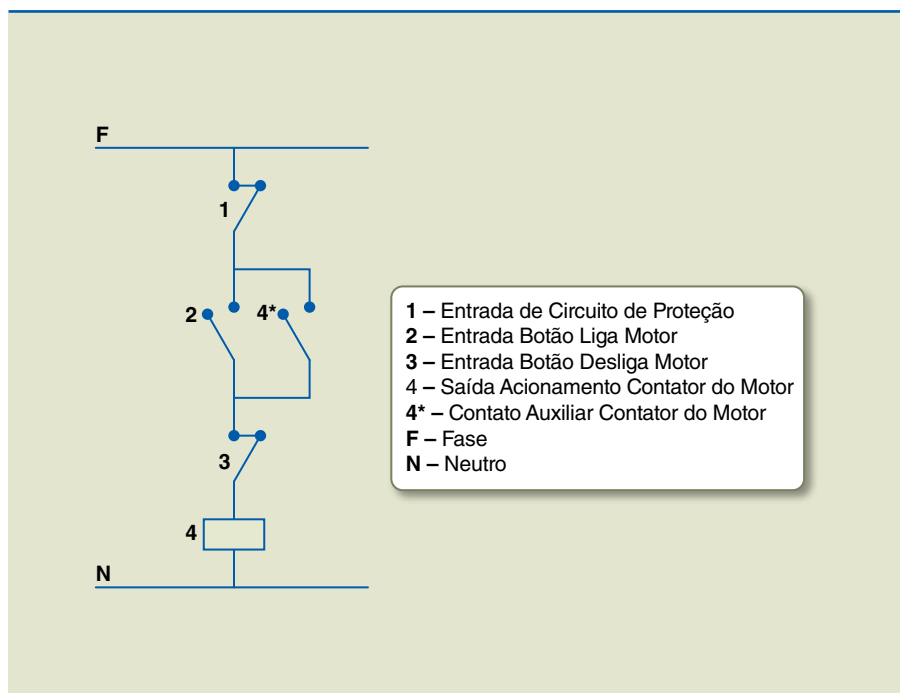
Utilizada na programação de CLPs, é uma linguagem gráfica baseada na interligação de blocos funcionais previamente disponibilizados pelos fabricantes ou que permite ao próprio programador construir os blocos a serem utilizados. As entradas e saídas são conectadas a esses blocos criando malha de interco-



nexões que possibilita a obtenção dos mesmos resultados de outras lógicas de programação.

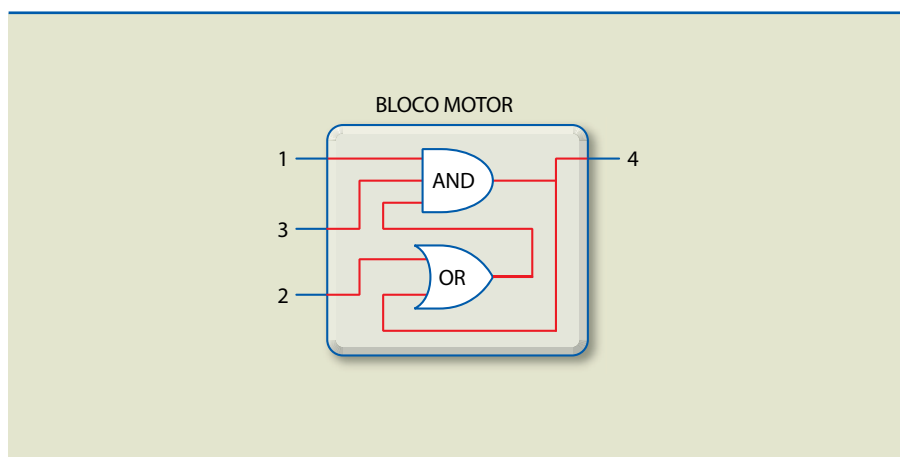
Uma das grandes vantagens dos blocos funcionais é a reutilização de blocos dentro de um programa. Suponha que um projeto use vários motores, todos com o mesmo princípio de funcionamento, conforme lógica predefinida de acionamento de um motor em partida direta (figura 4.2). Uma vez construído o bloco funcional de partida do motor, ele poderá ser utilizado várias vezes no programa, adotando entradas e saídas distintas, que, por sua vez, controlarão motores distintos.

**Figura 4.2**  
Diagrama de comandos elétricos da ligação de um motor:



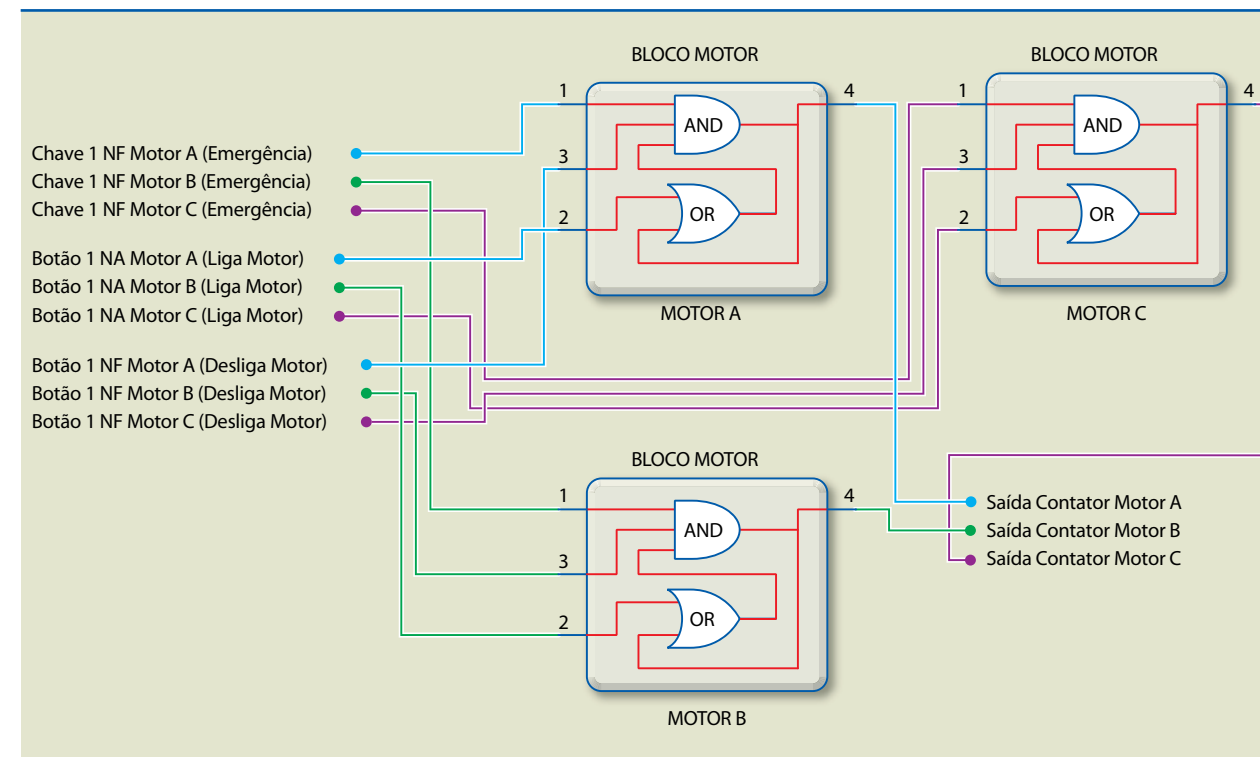
A figura 4.3 mostra como configurar esse bloco de acionamento de motor e a figura 4.4 exemplifica a utilização de um mesmo bloco em mais de um motor, considerando essa distinção de variáveis.

**Figura 4.3**  
Montagem de um bloco de nome "BLOCO MOTOR" baseado em blocos primários AND e OR.



**Figura 4.4**

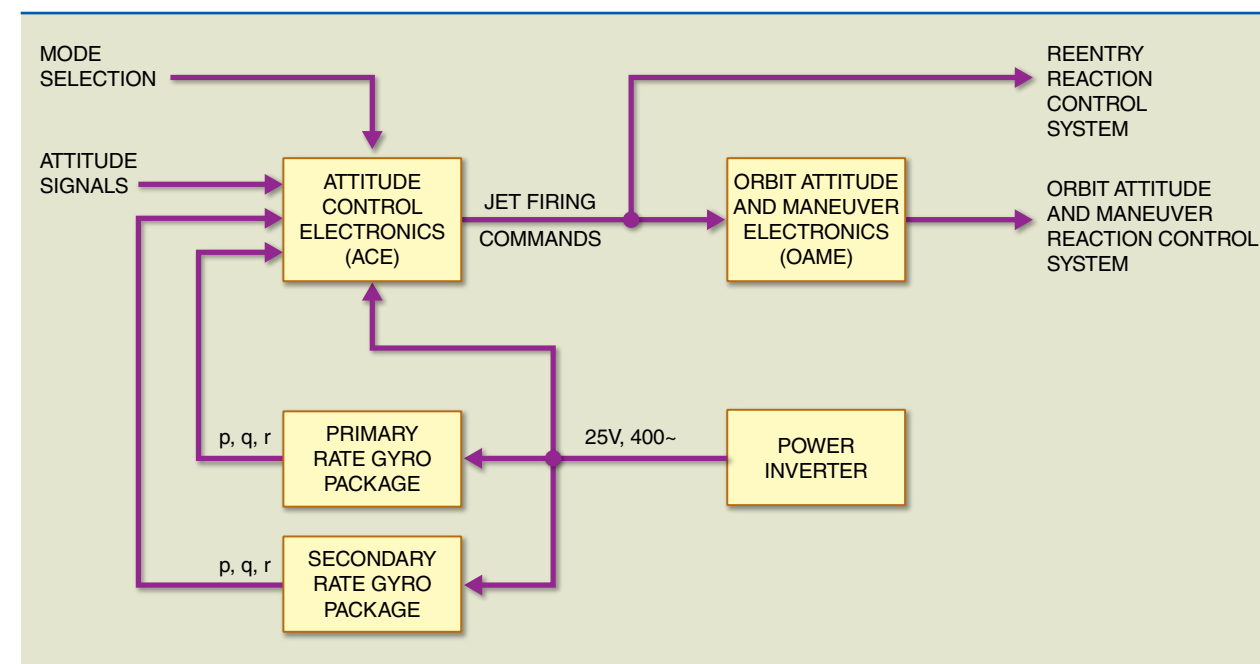
Utilização de vários "BLOCO MOTOR" dentro de um programa em diagrama de blocos funcionais.



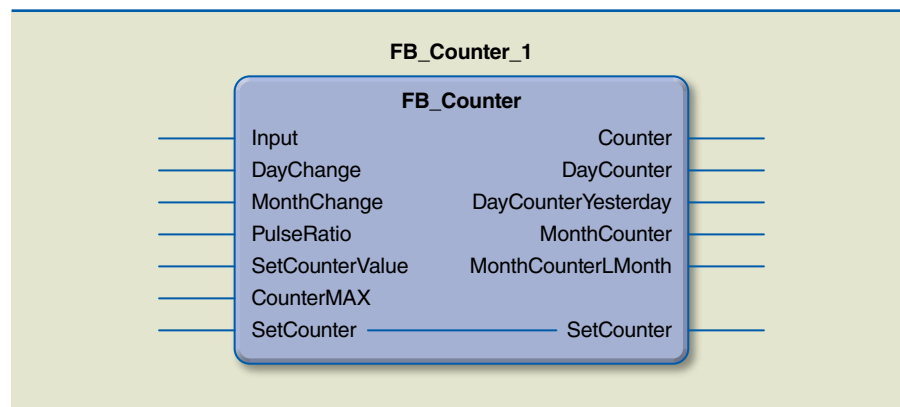
Para fixar o conceito de programação em diagrama de blocos funcionais, são apresentados nas figuras 4.5 e 4.6 outros exemplos de programas de fabricantes distintos.

**Figura 4.5**

Exemplo de programa em diagrama de blocos funcionais.



**Figura 4.6**  
Exemplo de bloco de função criado por programador:



## 4.5 LD – diagrama Ladder

A grande maioria das aplicações atuais em CLPs adota a programação Ladder, assunto que será tratado mais detalhadamente no capítulo 5 deste livro.

## 4.6 SFC – sequenciamento gráfico de funções

Também é uma linguagem gráfica de programação muito poderosa. Proporciona uma representação das sequências do processo controlado na forma de um diagrama. O SFC é utilizado para dividir um problema de controle, permitindo uma visão geral do processo e facilitando o diagnóstico. Outra grande vantagem é o suporte para sequências alternativas e paralelas, tornando possível que sub-rotinas que servem ao interesse do processo controlado sejam executadas de maneira paralela, sem a necessidade de parada da lógica principal de controle. Em resumo, o SFC vai além de uma programação gráfica usada em CLPs: é uma forma de estruturar a lógica e a sequência de eventos desejadas em um processo a ser automatizado.

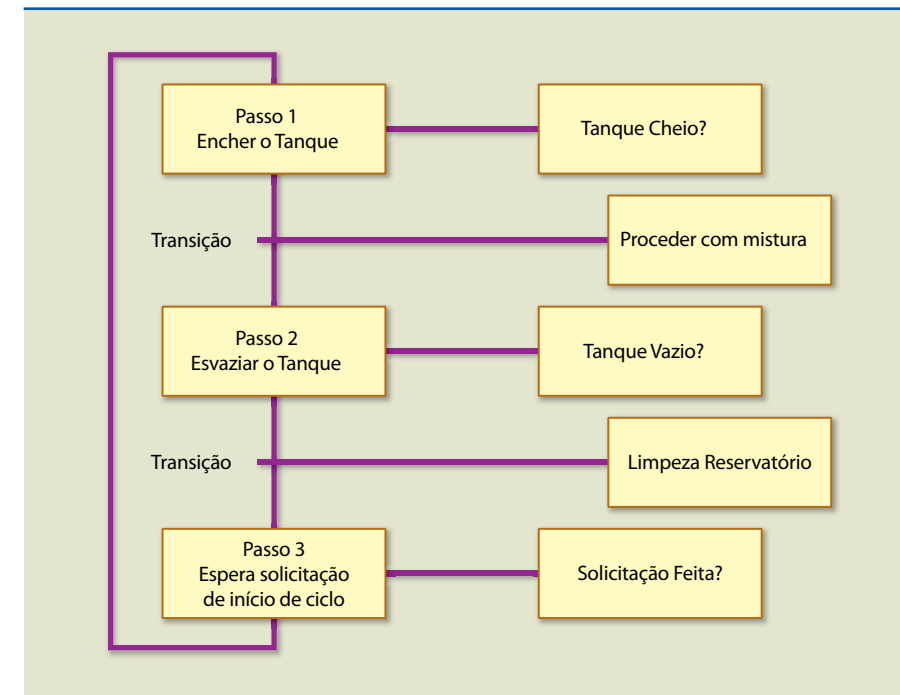
O SFC é elaborado com blocos funcionais dispostos como um fluxograma, possibilitando a confecção e o estudo dos processos por meio de ações e transições que devem ocorrer. Isso permite que um processo seja aberto ao menor nível de análise até que se tenha o modelo desejado mapeado em detalhes.

Também conhecido como GRAFCET, o SFC é baseado no conceito de análise binária das redes de Petri, levando em consideração, para ações futuras, os atuais estados de variáveis monitoradas. De forma prática, podemos analisar o funcionamento de uma lógica produzida em SFC observando a figura 4.7.

Na lógica apresentada na figura 4.7, os retângulos representam os passos a serem executados no controle do processo e, entre alguns retângulos, a condição necessária para que se chegue ao novo passo. Desse modo, podemos garantir que determinado passo nunca ocorra sem que uma transição esteja concluída.

Analisando o exemplo da figura 4.7, para que o passo 1 seja concluído, é necessário que sua resposta seja positiva, ou seja, o tanque está cheio. Enquanto o tanque estiver vazio, ele permanecerá monitorando essa etapa do processo.

**Figura 4.7**  
Exemplo de lógica em SFC.



Cada um dos blocos poderá ser programado na linguagem que for mais conveniente ao programador, pensando nas seguintes facilidades:

- Gerar o código do programa.
- Garantir que outras pessoas compreendam o programa.
- Fazer manutenção e alterações no *software*.

No entanto, vale ressaltar que, se a tarefa envolver lógica simples, poderá ser conveniente o uso do diagrama Ladder; e, se contiver muitas fórmulas matemáticas, será mais conveniente uma linguagem do tipo texto estruturado.



# Capítulo 5

**A programação**

---

**Ladder**

---



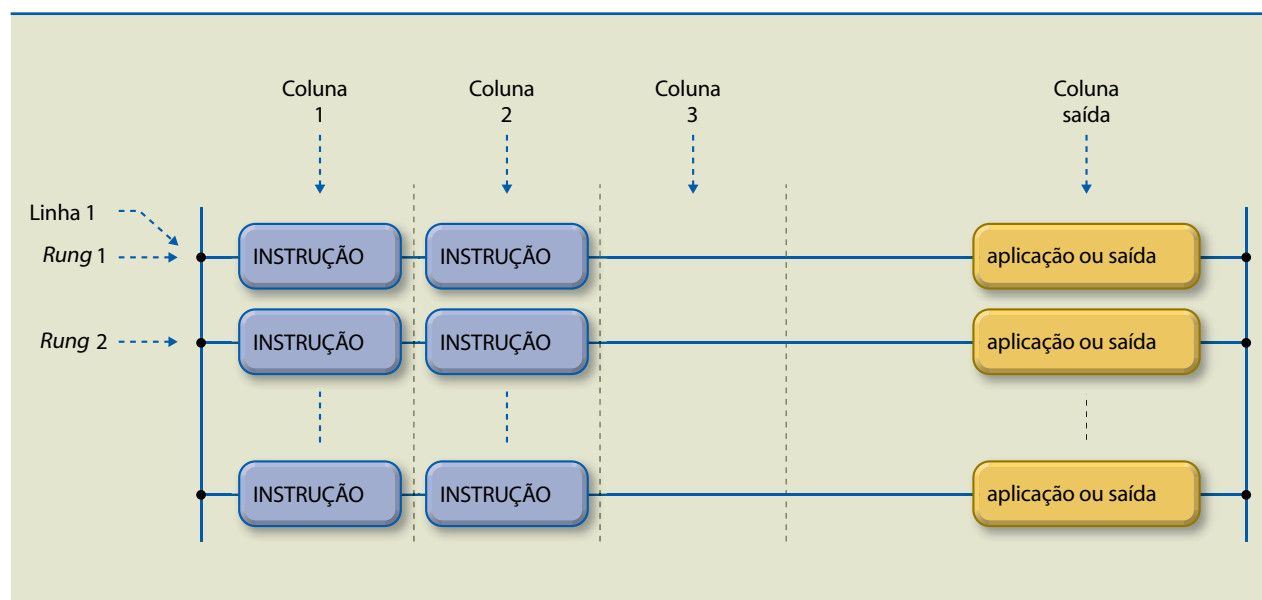
**A**Ladder foi uma das primeiras linguagens destinadas à programação de CLPs, criada para permitir que técnicos e engenheiros da área de automação com conhecimentos de lógica de relés e nenhum de programação conseguissem programar o CLP. Por esse motivo, ela se tornou a linguagem mais popular entre os programadores.

## 5.1 Diretivas básicas

As variáveis associadas aos elementos de entrada, saída, memória, temporizadores e contadores são denominadas operandos. O programa executa operações lógicas e aritméticas com esses operandos.

Na linguagem Ladder, as linhas de contatos (instruções) possuem a aparência de degraus (*rungs*) de uma escada (*ladder*), que podem ser associados a uma estrutura de colunas e linhas, conforme ilustra a figura 5.1. Em cada linha, as instruções correspondem ao programa, ou seja, ao processamento dos operandos, e o resultado é atribuído a outro operando no bloco “Saída”, à direita.

**Figura 5.1**  
Diagrama Ladder:



O número de linhas e colunas ou elementos e associações que cada *rung* admite varia conforme o fabricante do CLP e pode variar também de acordo com a UCP utilizada. Em geral, esses limites não representam preocupação ao usuário no desenvolvimento do programa de aplicação, pois, se o limite for ultrapassado, o *software* de programação apresentará uma mensagem de erro durante a compilação do programa.

Os operandos podem ser divididos em três classes:

- **Memória (M)** – Servem para o armazenamento dos resultados parciais, valores de constantes, dados de transmissão, valores de referência, receitas etc. Esses operandos podem ser livremente lidos e escritos pelo programa.
- **Entradas (I)** – Estão associados aos módulos de entrada. Podem ser lidos pelo programa, mas escritos apenas pelos módulos de entrada.
- **Saídas (Q)** – Estão associados aos módulos de saída. Podem ser livremente lidos e escritos pelo programa.

Os operandos, por sua vez, são divididos, inicialmente, em cinco tipos, segundo sua utilização e número de bits:

- **Bits (X)** – Utilizados para a implementação de lógica, ocupam 1 bit de memória.
- **Bytes (B)** – Utilizados para o armazenamento de caracteres ASCII, ocupam 8 bits.
- **Words (W)** – Utilizados para o armazenamento de valores numéricos inteiros, ocupam 16 bits.
- **Double word (D)** – Semelhante ao tipo W, ocupa 32 bits.
- **Word long (L) de 64 bits** – Semelhante ao tipo W, ocupa 64 bits.

Originalmente, na linguagem Ladder cada instrução correspondia aos contatos NA ou NF dos relés, cujo estado era definido pelo valor do operando (do tipo B) a ele associado. Na mesma época, as saídas eram as bobinas (operando tipo B). Com o tempo, os blocos de instruções passaram a contemplar contadores, temporizadores, operações aritméticas etc., que exigiram que fossem criados os tipos de operando citados anteriormente.

O objeto de estudo das instruções de programação Ladder será um CLP genérico, com os seguintes elementos:

- 8 entradas digitais.
- 8 saídas digitais.
- 4 entradas analógicas.
- 2 saídas analógicas.
- 6 saídas a relé.
- 1 entrada de contagem rápida de pulsos a 4 kHz.
- 32 temporizadores.
- 32 contadores.



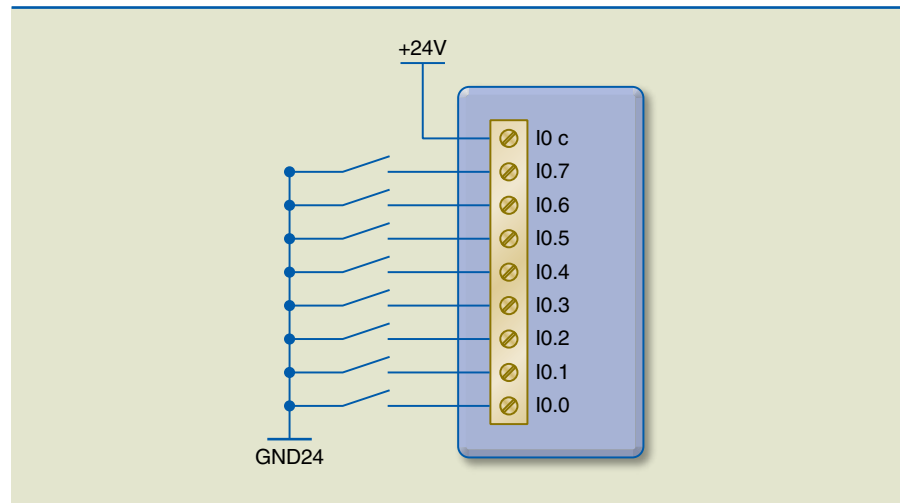
## 5.2 Ligação das entradas e saídas digitais do CLP genérico

Essas instruções possuem apenas dois estados, que são representados pelos números binários “0” ou “1” e podem ser interpretados como desligado e ligado. Geralmente, estão associados a dispositivos elétricos como botoeiras, chaves liga e desliga, válvulas eletropneumáticas, pressostatos, termostatos etc.

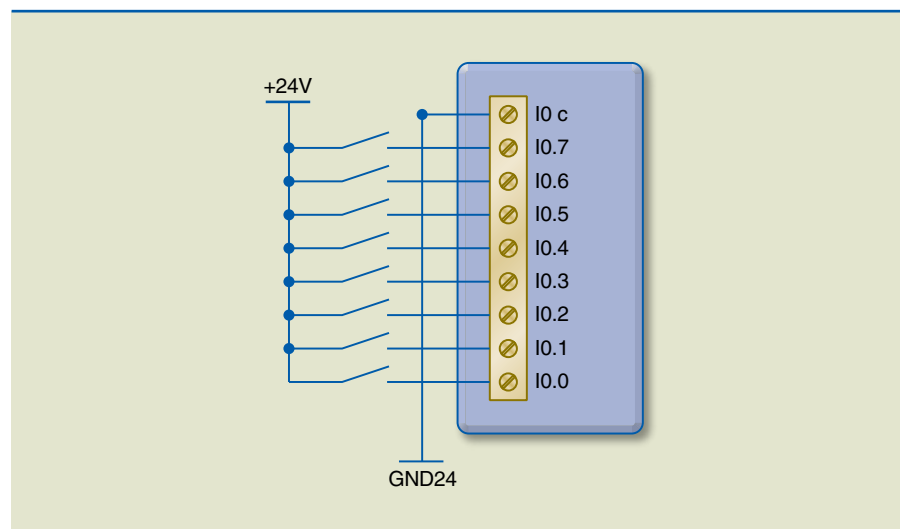
### 5.2.1 Entradas digitais

As entradas digitais do CLP genérico são acionadas por botões, chaves, interfaces, *encoders* e outros dispositivos e podem ser do tipo NPN ou PNP. A entrada NPN é acionada com 0  $V_{cc}$  (figura 5.2). Nessa configuração, o polo negativo é chaveado e o positivo permanece conectado. A entrada PNP é acionada com 24  $V_{cc}$  (figura 5.3). Nessa configuração, o polo positivo é chaveado e o negativo permanece conectado.

**Figura 5.2**  
Entrada digital NPN.



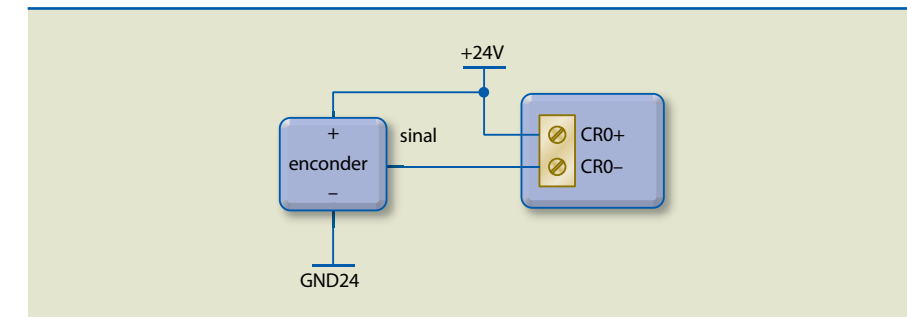
**Figura 5.3**  
Entrada digital PNP.



Em nosso CLP genérico, utilizaremos entradas digitais PNP, que serão identificadas de I0.0 a I0.7. A identificação das entradas pode variar de acordo com o fabricante do CLP, usando, por exemplo, as letras “E” ou “X”. Da mesma forma, a identificação de qualquer componente também pode variar (saídas, temporizadores, contadores, memórias etc.).

### 5.2.2 Entrada rápida

A entrada rápida possui uma frequência máxima de 4 kHz e efetua contagem unidirecional. Em geral, a contagem é efetuada por borda de descida ou subida, dependendo exclusivamente do fabricante do CLP. Na figura 5.4 utiliza-se um *encoder* para demonstrar esse tipo de contagem. Essa entrada é acionada com 24  $V_{cc}$  no borne de CR0+ e com GND no borne CR0-.

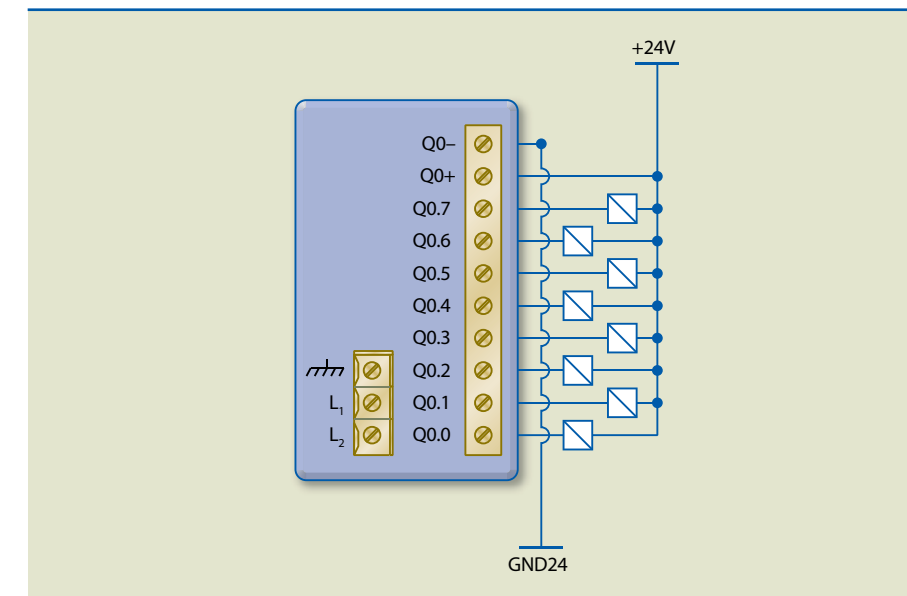


**Figura 5.4**  
Entrada rápida.

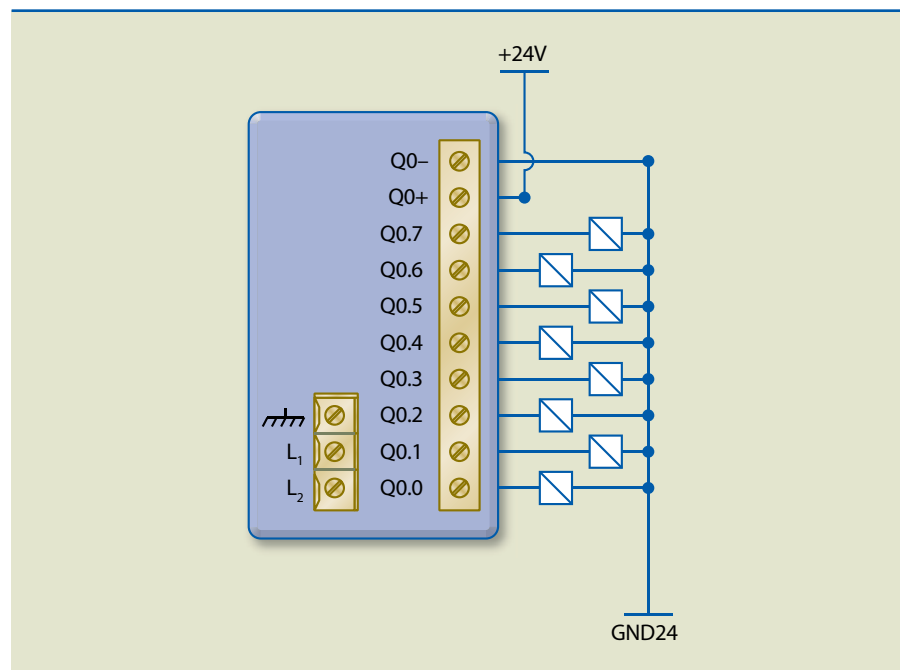
### 5.2.3 Saídas digitais

De maneira análoga às entradas digitais, as saídas do CLP genérico também podem ser do tipo NPN ou PNP. Elas são acionadas de acordo com a programação feita pelo usuário. As saídas NPN, quando acionadas, fornecem para a carga o potencial GND (figura 5.5). As saídas PNP, quando acionadas, fornecem para a carga o potencial 24  $V_{cc}$  (figura 5.6).

**Figura 5.5**  
Saída digital NPN.



**Figura 5.6**  
Saída digital PNP.

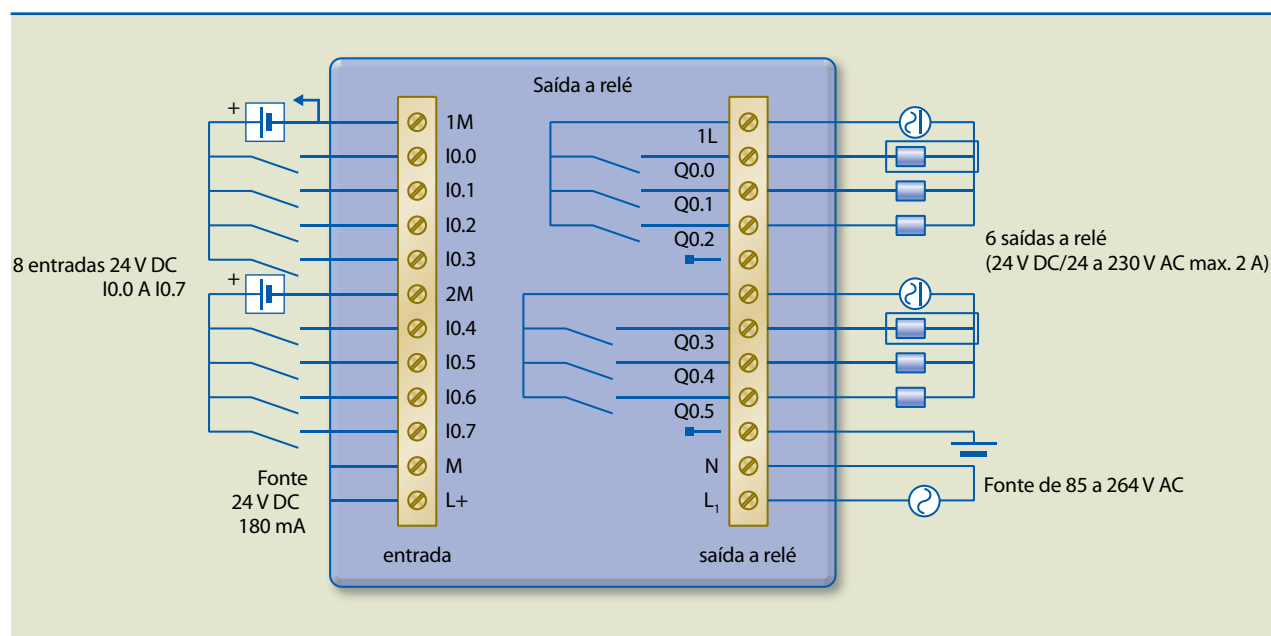


### 5.2.4 Saídas a relé

A saída a relé é muito utilizada por sua versatilidade em comandar cargas que trabalham com tensão alternada ou contínua. Quando ativada, a saída chaveia um contato de relé.

A figura 5.7 mostra saídas a relé para cargas em corrente alternada ou contínua. Nela, pode-se observar que a polaridade na ligação é indiferente, porém, deve-se ficar atento aos limites de corrente e de tensão do contato. Consulte essas informações no manual do CLP.

**Figura 5.7**  
Saídas a relé para cargas em CC ou CA.



## 5.3 Contatos NA/NF

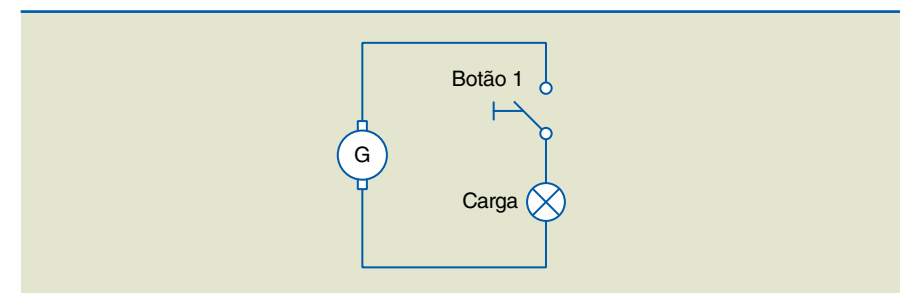
O diagrama de contatos Ladder funciona como um esquema elétrico cujos principais elementos são o contato normalmente aberto, o contato normalmente fechado e a bobina do relé.

### 5.3.1 Contato NA

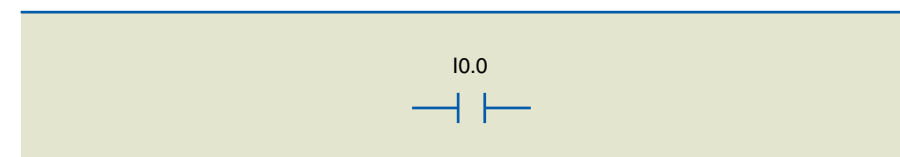
Essa instrução funciona do seguinte modo: quando o bit associado a um contato normalmente aberto for acionado, o contato fechará; caso contrário, ele permanecerá aberto. Outra maneira de entender é imaginando um botão com o contato normalmente aberto: enquanto esse botão estiver solto, o contato ficará aberto, porém, ao ser pressionado, o contato do botão fechará.

Caso o botão NA esteja em um circuito elétrico, ocorrerá a passagem de corrente elétrica nos componentes do circuito. Se houver uma carga em série com esse botão e uma tensão de alimentação, a carga será acionada.

A figura 5.8 mostra o circuito elétrico, e a figura 5.9, a representação gráfica de um contato NA em diagrama Ladder. Note que, na figura 5.8, em cima da instrução NA, aparece o endereço do operando relacionado a ela.



**Figura 5.8**  
Circuito elétrico: contato NA (botão I).



**Figura 5.9**  
Representação gráfica do contato NA em diagrama Ladder.

Na figura 5.9, o contato NA relacionado ao operando I0.0 (entrada) estará aberto se a entrada estiver desacionada (nível lógico "0") e fechado se a entrada estiver acionada (nível lógico "1").

### 5.3.2 Contato NF

Essa instrução funciona do seguinte modo: quando o bit associado a um contato normalmente fechado for acionado, o contato abrirá; caso contrário, ele permanecerá fechado. Outra maneira de entender é imaginando um botão com o contato normalmente fechado: enquanto esse botão estiver solto, o contato ficará fechado, porém, ao ser pressionado, o contato do botão abrirá.

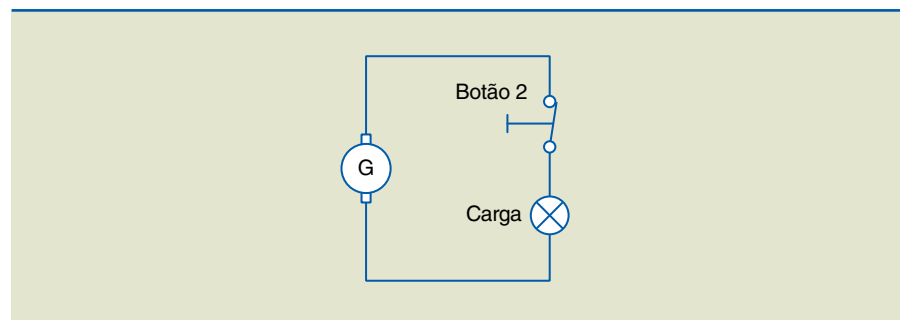


Caso o botão NF esteja em um circuito elétrico, não ocorrerá passagem de corrente elétrica. Se houver uma carga em série com esse botão e uma tensão de alimentação, a carga será desligada.

A figura 5.10 mostra o circuito elétrico, e a figura 5.11, a representação gráfica de um contato NF em diagrama Ladder. Note que, na figura 5.10, em cima da instrução NF, aparece o endereço do operando relacionado a ela.

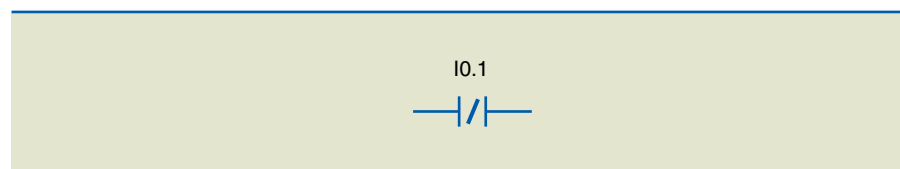
**Figura 5.10**

Circuito elétrico: contato NF (botão 2).



**Figura 5.11**

Representação gráfica do contato NF em diagrama Ladder.



Na figura 5.11, o contato NF relacionado ao operando I0.1 (entrada) estará fechado se a entrada estiver desligada (nível lógico “0”) e aberto se a entrada estiver acionada (nível lógico “1”).

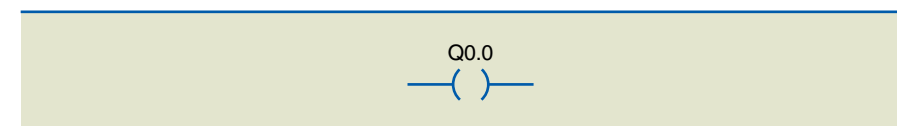
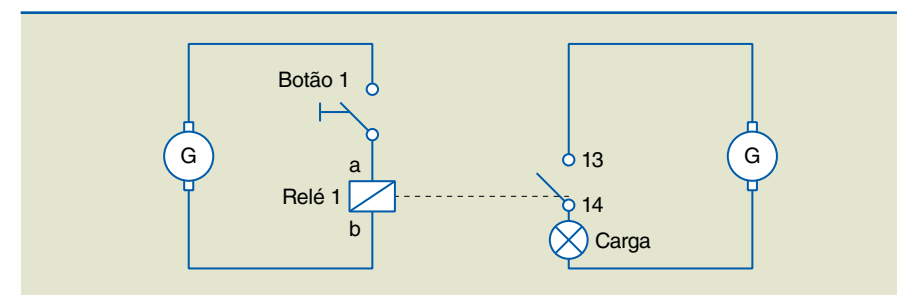
## 5.4 Saída simples/saída complementar

A saída tem por base a ideia de continuidade lógica a ser garantida entre os extremos das linhas de programação. Uma saída será verdadeira se todas as instruções declaradas na linha lógica forem verdadeiras.

### 5.4.1 Saída simples

Essa instrução, ao ser acionada, transfere para o endereço associado a ela o valor da tensão que estiver em sua entrada. Por exemplo, em circuitos elétricos, utilizam-se diretamente relés ou contatores para acionar cargas como motores, resistências etc. Na figura 5.12, quando acionam-se o botão 1, energizam-se a bobina do relé 1, o que, conseqüentemente, fecha os contatos 13 e 14, acionando a carga.

Em nosso CLP genérico, o relé 1 representa uma saída simples que tem como operando o endereço de saída Q0.0. A figura 5.13 mostra a representação gráfica de uma saída simples. Note que, em cima da instrução bobina, aparece o endereço do operando relacionado a ela.



**Figura 5.12**

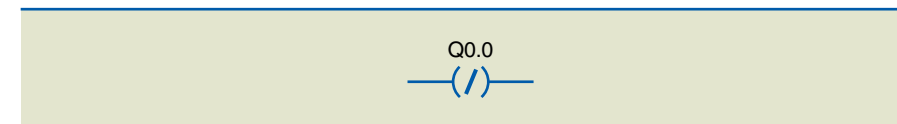
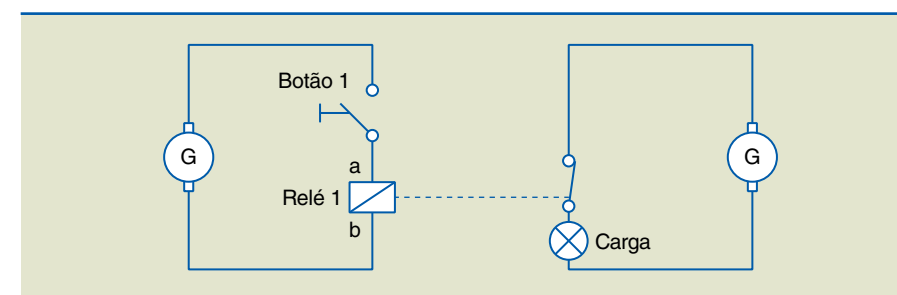
Circuito elétrico utilizando relé para acionamento de uma carga.

**Figura 5.13**

Representação gráfica de uma saída simples em diagrama Ladder.

### 5.4.2 Saída complementar

Essa instrução, ao ser acionada, transfere para o endereço associado a ela o valor de tensão oposto ao de sua entrada. Na figura 5.14, quando acionamos o botão 1, energizamos a bobina do relé 1, o que, conseqüentemente, abre os contatos 21 e 22, desligando a carga. Nesse caso, o relé 1 representa uma saída complementar que tem como operando o endereço de saída Q0.1. A figura 5.15 mostra a representação gráfica de uma saída complementar. Note que, em cima da instrução bobina, aparece o endereço do operando relacionado a ela.



**Figura 5.14**

Circuito elétrico utilizando relé para desacionar uma carga.

**Figura 5.15**

Representação gráfica de uma saída complementar em diagrama Ladder.

Agora que já conhecemos algumas instruções e suas representações gráficas, vamos ver alguns exemplos de programas, inicialmente utilizando circuitos simples, para facilitar a compreensão.

### Exemplos

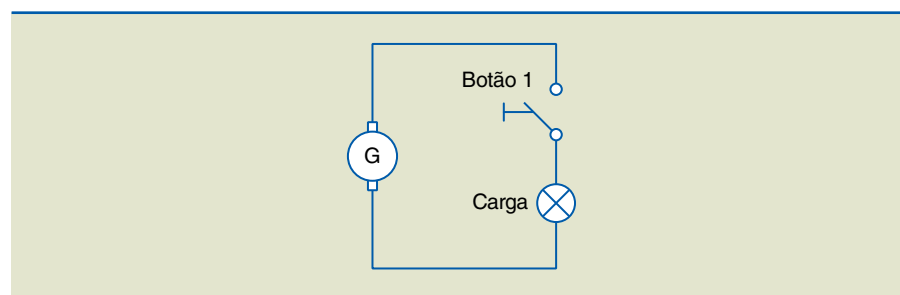
1. Funcionamento da instrução NA e da saída simples.

Faça o diagrama Ladder para o circuito da figura 5.16.



**Figura 5.16**

Circuito elétrico – contato NA representado pelo botão 1.



*Solução:*

O programa começa com a identificação das entradas e das saídas. Faça uma tabela mostrando cada um desses endereços e relacione-os a uma simbologia que identifique a função das instruções (tabela 5.1). Em programas complexos, isso é essencial na resolução de problemas e em modificações técnicas. Se possível, adicione um comentário.

**Tabela 5.1**

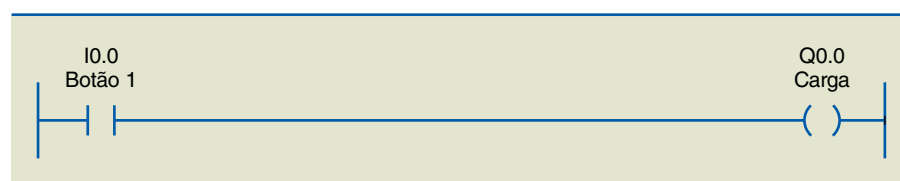
Endereços, símbolos e comentários

Endereço	Símbolo	Comentário
I0.0	Botão 1	Aciona a carga
Q0.0	Carga	Pode ser motor, lâmpada, relé, resistência etc.

O programa em Ladder para o circuito, apresentado na figura 5.17, mostra que a saída simples (Q0.0) será acionada somente quando a entrada (I0.0) for acionada, ou seja, quando estiver em nível lógico “1”.

**Figura 5.17**

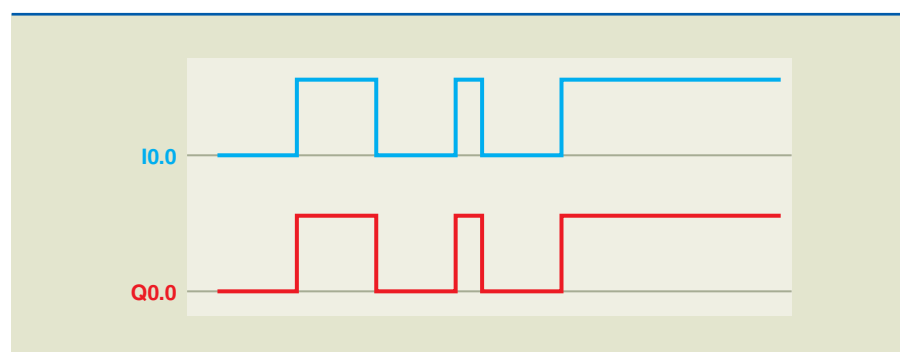
Programa em Ladder para o circuito da figura 5.16.



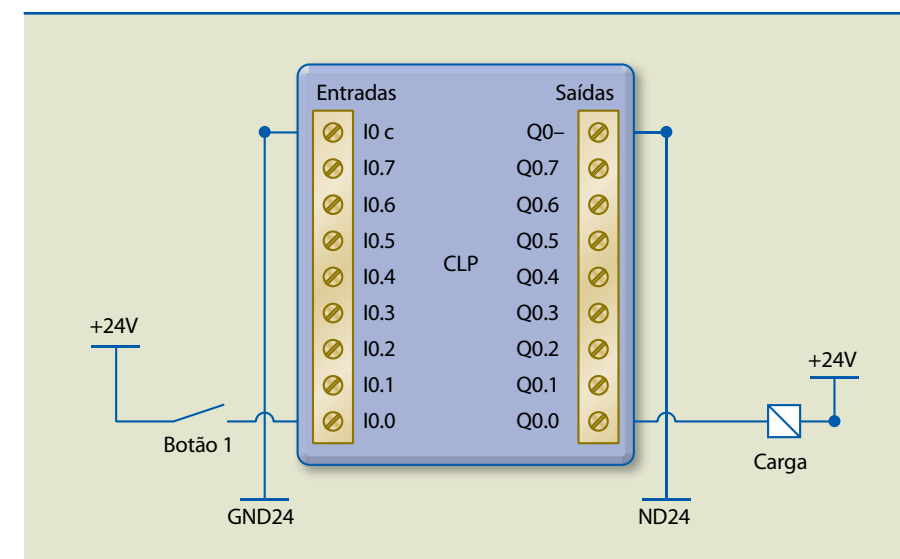
A figura 5.18 ilustra o diagrama de estado da entrada (I0.0) e da saída (Q0.0) em função do tempo. Note que o estado da saída acompanha o estado da entrada.

**Figura 5.18**

Diagrama de estado da entrada NA (I0.0) e da saída simples (Q0.0).



O esquema de ligação no CLP genérico está representado na figura 5.19, que mostra somente as entradas e saídas digitais.

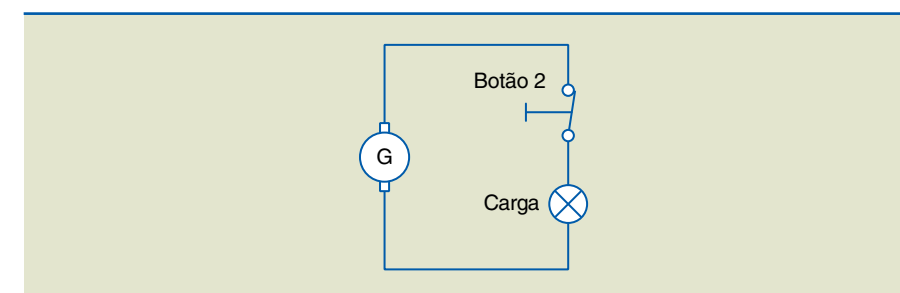


**Figura 5.19**

Esquema de ligação no CLP genérico.

**2. Funcionamento da instrução NF e da saída simples.**

Faça o diagrama Ladder para o circuito da figura 5.20.



**Figura 5.20**

Circuito elétrico – contato NF representado pelo botão 2.

*Solução:*

Inicie a programação pela tabela de endereços, símbolos e comentários (tabela 5.2). Quanto mais informações forem incluídas no diagrama, mais fácil será modificá-lo caso necessário.

Endereço	Símbolo	Comentário
I0.0	Botão 2	Desacionar a carga quando pressionado
Q0.0	Carga	Pode ser motor, lâmpada, relé, resistência etc.

**Tabela 5.2**

Endereços, símbolos e comentários

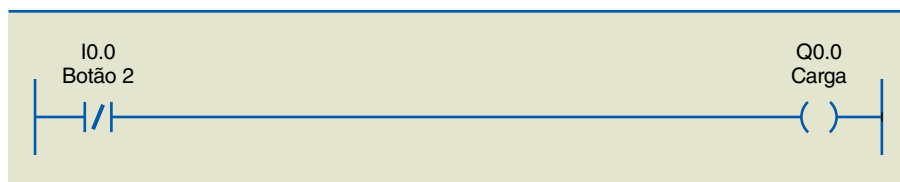
O programa em Ladder para o circuito, apresentado na figura 5.21, mostra que a saída simples (Q0.0) será desacionada (nível lógico “0”) somente quando a entra-



da (I0.0) for acionada, ou seja, quando estiver em nível lógico “1”. Portanto, a saída é o oposto da entrada.

**Figura 5.21**

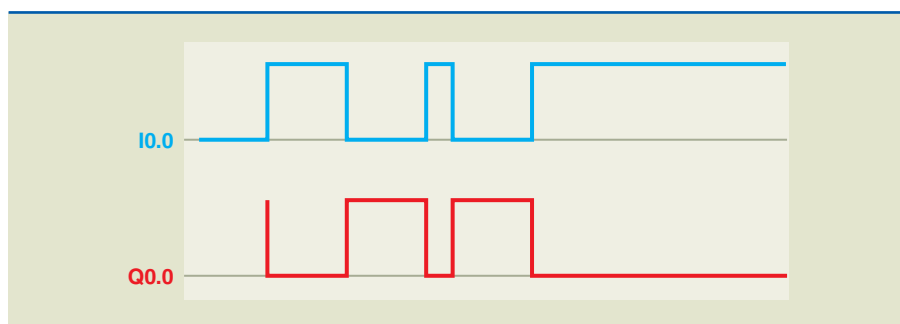
Programa em Ladder para o circuito da figura 5.20.



A figura 5.22 ilustra o diagrama de estado da entrada (I0.0) e da saída (Q0.0) em função do tempo. Note que o estado da saída é o contrário do estado da entrada.

**Figura 5.22**

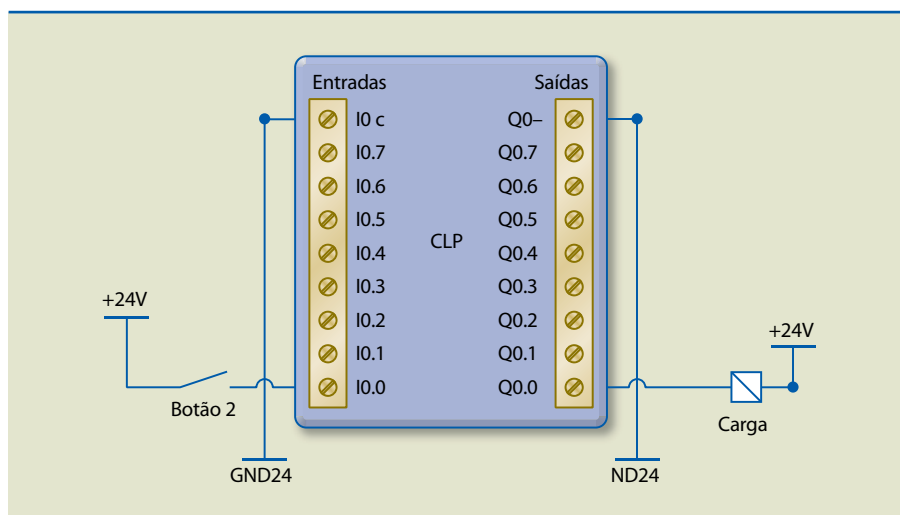
Diagrama de estado da entrada NF (I0.0) e da saída simples (Q0.0).



O esquema de ligação no CLP está representado na figura 5.23. Observe que a ligação não foi alterada: tanto a entrada como a saída permanecem da mesma forma; porém, o funcionamento é diferente. Essa é uma característica dos CLPs. Sem alteração na ligação elétrica, mas com mudança na programação, altera-se o funcionamento do sistema.

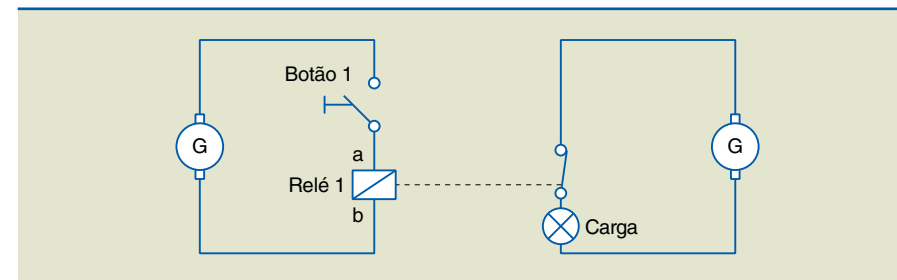
**Figura 5.23**

Esquema de ligação no CLP genérico.



**3. Funcionamento da instrução NA e da saída complementar.**

Faça o diagrama Ladder para o circuito da figura 5.24.



**Figura 5.24**

Circuito elétrico utilizando relé para desacionar uma carga.

*Solução:*

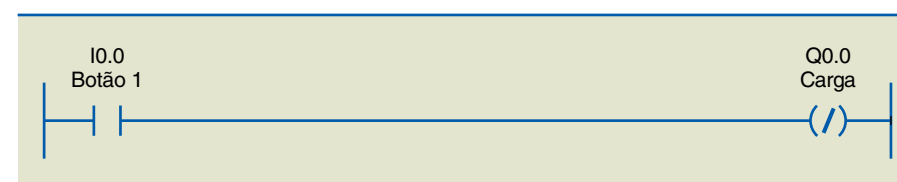
Inicie a programação pela tabela de endereços, símbolos e comentários (tabela 5.3). Quanto mais informações forem incluídas no diagrama, mais fácil será modificá-lo caso necessário.

Endereço	Símbolo	Comentário
I0.0	Botão 1	Desacionar a carga quando pressionado
Q0.0	Carga	Pode ser motor, lâmpada, relé, resistência etc.

**Tabela 5.3**

Endereços, símbolos e comentários

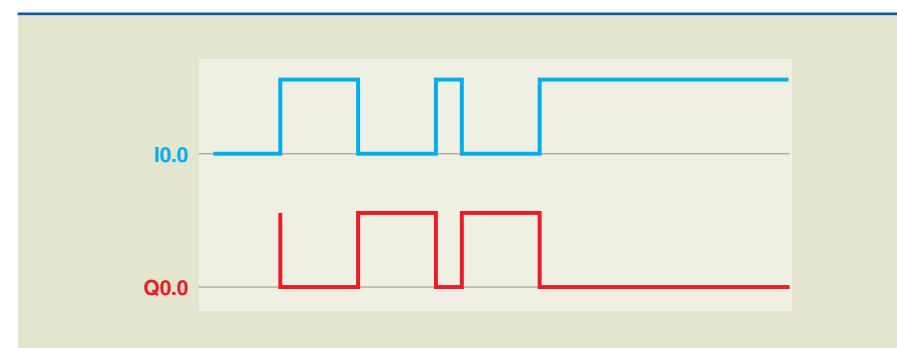
O programa em Ladder para o circuito, apresentado na figura 5.25, mostra que, quando acionada, a entrada (I0.0) transferirá para a saída complementar (Q0.0) o inverso do sinal da entrada associada a ela (nível lógico “0”). Isto significa que somente quando a entrada (I0.0) for acionada, ou seja, quando estiver em nível lógico “1”, a saída será o oposto da entrada (nível lógico “0”).



**Figura 5.25**

Programa em Ladder para o circuito da figura 5.24.

A figura 5.26 ilustra o diagrama de estado da entrada (I0.0) e da saída complementar (Q0.0) em função do tempo. Note que o estado da saída é o contrário do estado da entrada.



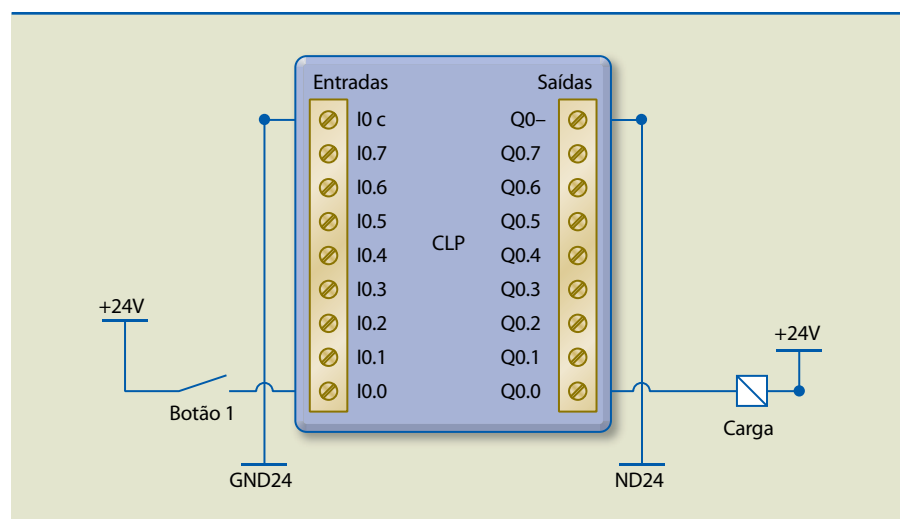
**Figura 5.26**

Diagrama de estado da entrada NA (I0.0) e da saída complementar (Q0.0).



O esquema de ligação no CLP está representado na figura 5.27, mostrando de novo que, com o funcionamento diferente, pode-se utilizar a mesma ligação externa.

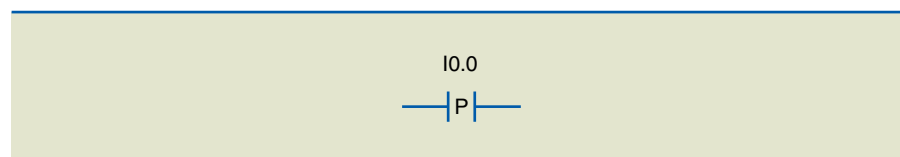
**Figura 5.27**  
Esquema de ligação no CLP genérico.



### 5.5 Contato por borda positiva

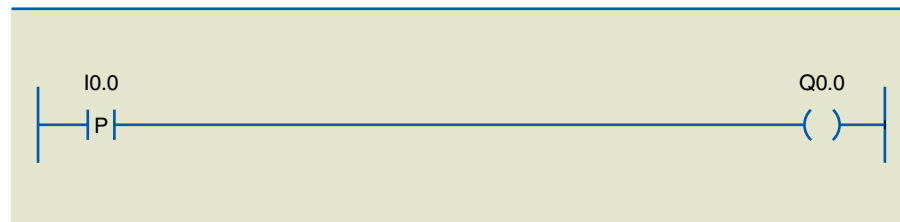
Outra das instruções consideradas especiais, por possuírem características e aplicações peculiares, é o contato por borda positiva (figura 5.28). Essa instrução gera um pulso na saída associada a ela. O pulso tem o período de 1 scan e inicia-se quando a entrada faz a passagem do nível lógico “0” para o “1”.

**Figura 5.28**  
Representação gráfica do contato por borda positiva.

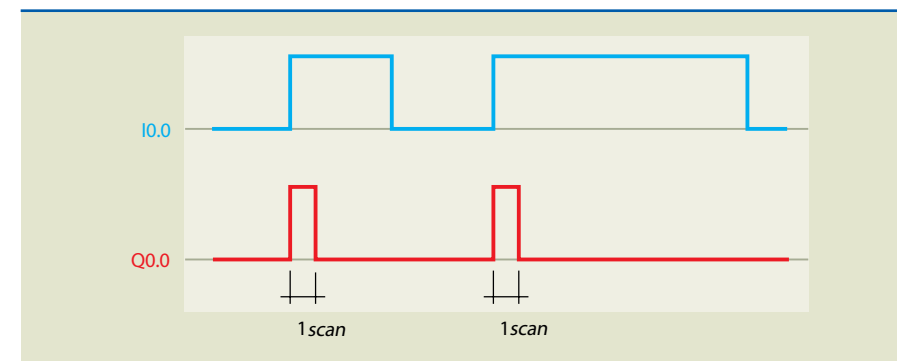


Um programa utilizando esse tipo de instrução pode ser visto na figura 5.29.

**Figura 5.29**  
Lógica utilizando contato por borda positiva e saída simples.



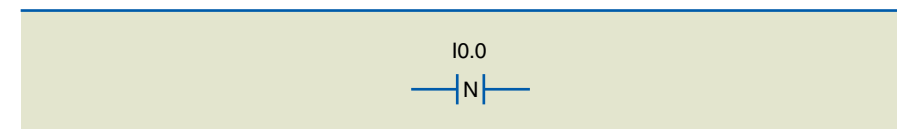
O diagrama de estado da entrada por borda positiva (I0.0) e da saída simples (Q0.0), apresentado na figura 5.30, demonstra a aplicação dessa instrução. Observe que, quando a entrada (I0.0) é acionada, ou seja, na passagem do nível lógico “0” para o “1”, na saída associada a essa entrada aparece um pulso com duração de 1 scan. Na descida, na passagem do nível lógico “1” para o “0”, nada acontece à saída.



**Figura 5.30**  
Diagrama de estado da entrada por borda positiva e da saída simples.

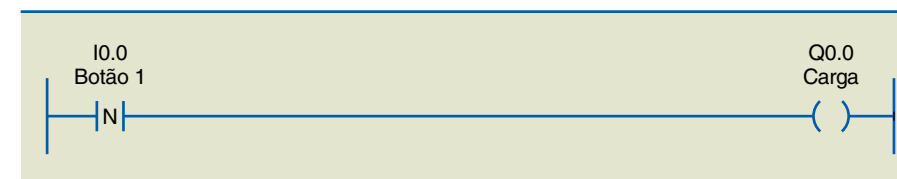
### 5.6 Contato por borda negativa

A instrução contato por borda negativa (figura 5.31) gera um pulso na saída associada a ela. Esse pulso tem o período de 1 scan e inicia-se quando a entrada faz a passagem do nível lógico “1” para o “0”.



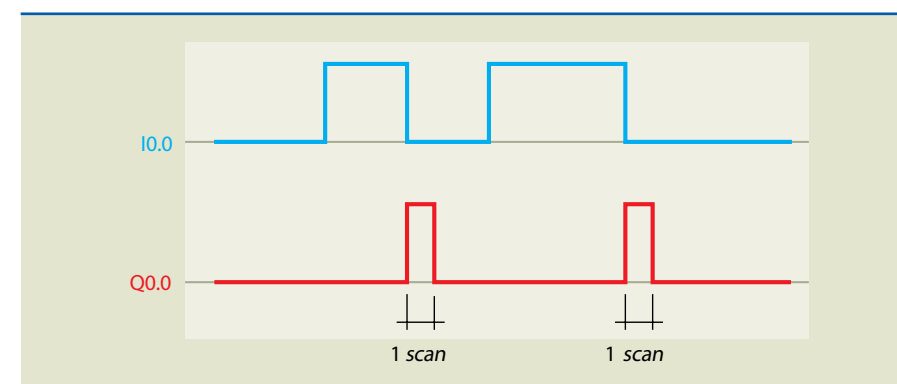
**Figura 5.31**  
Representação gráfica do contato por borda negativa.

Um programa utilizando esse tipo de instrução pode ser visto na figura 5.32.



**Figura 5.32**  
Lógica utilizando contato por borda negativa e saída simples.

O diagrama de estado da entrada por borda negativa (I0.0) e da saída simples (Q0.0), apresentado na figura 5.33, demonstra a aplicação dessa instrução. Observe que, quando a entrada (I0.0) é acionada, nada acontece; porém, ao ser desligada, na passagem do nível lógico “1” para o “0”, na saída associada a essa entrada aparece um pulso com duração de 1 scan.



**Figura 5.33**  
Diagrama de estado da entrada por borda negativa e da saída simples.



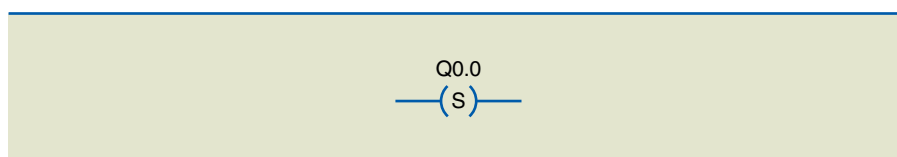
### 5.7 Saída set e saída reset

As instruções *set* e *reset* são utilizadas para memorização dos sinais de saída do CLP.

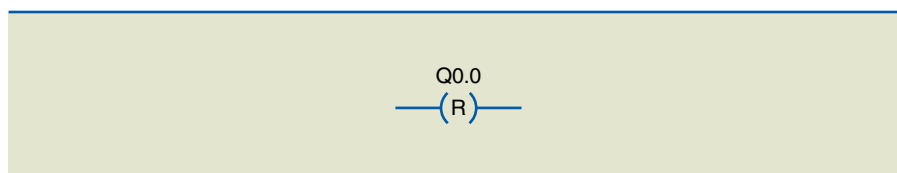
A instrução *set* (figura 5.34) serve para acionar e manter acionado um operando de saída quando, na entrada associada a ela, houver um pulso (passagem do nível lógico “0” para o “1”). Mesmo que a entrada associada à instrução *set* passe para o nível lógico “0” (transição do nível lógico “1” para o “0”), a saída permanecerá acionada.

A instrução *reset* (figura 5.35) serve para desacionar e manter desacionado um operando de saída quando, na entrada associada a ela, houver um pulso (passagem do nível lógico “0” para o “1”). A instrução *reset* permanecerá em “0” mesmo que a entrada associada a ela passe para o nível lógico “0” (transição do nível lógico “1” para o “0”).

**Figura 5.34**  
Representação gráfica saída set.

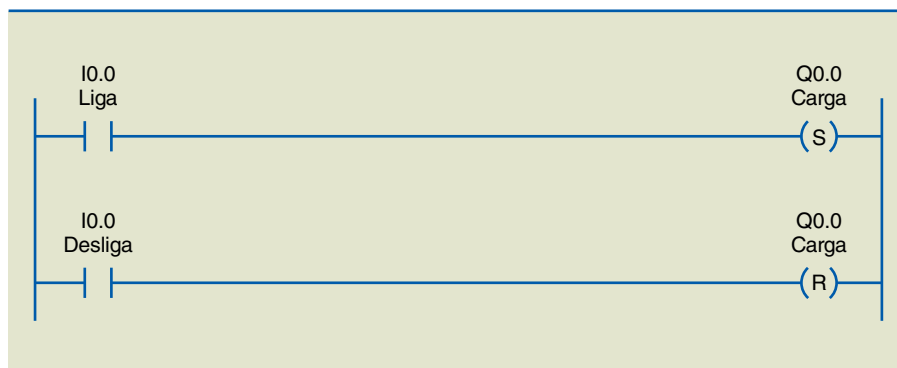


**Figura 5.35**  
Representação gráfica saída reset.

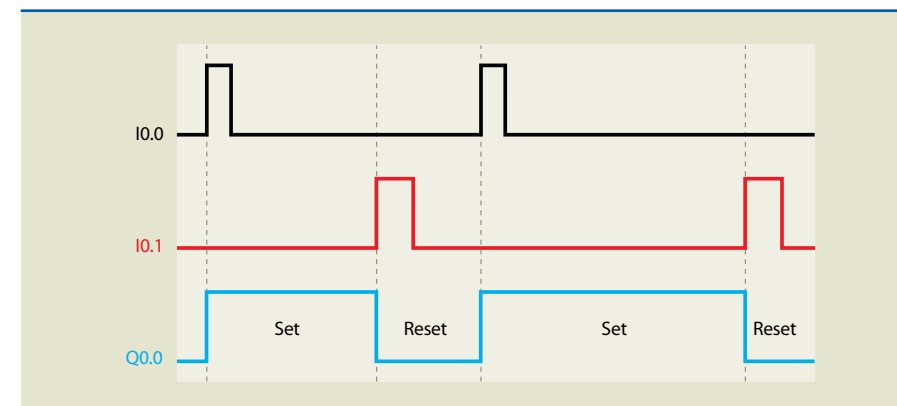


Em resumo, a saída *set* liga um operando quando a entrada associada a ele passa do nível lógico “0” para o “1”, e a saída *reset* desliga o operando quando a entrada associada a ele passa de “0” para “1” (figura 5.36).

**Figura 5.36**  
Lógica Ladder utilizando as saídas set e reset.



Analisando o diagrama de estado do programa da figura 5.36, pode-se notar que a saída *set*, carga (Q0.0), é acionada quando a entrada liga (I0.0) passa do nível lógico “0” para o “1” e permanece acionada mesmo quando a entrada liga (I0.0) passa para o nível lógico “0”. A saída carga (Q0.0) só será desligada quando a entrada desliga (I0.1) for acionada (figura 5.37).



**Figura 5.37**  
Saídas set e reset.

### 5.8 Memória ou flag

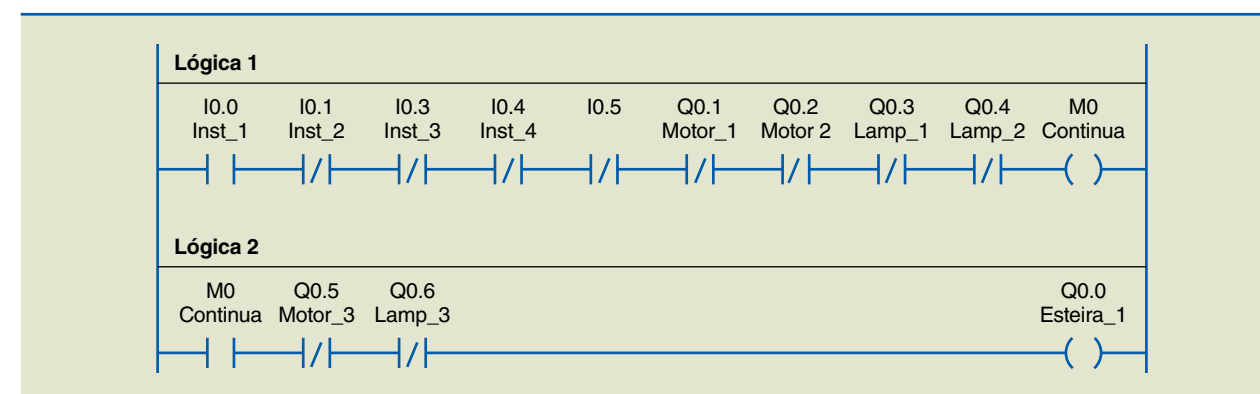
Essas instruções podem ser utilizadas como entradas ou saídas. Elas são bits de memória interna que, em nosso caso, podem ser endereçadas conforme a tabela 5.4.

Endereços	Tamanho	Descrição
M0 a M255	1 bit (nível “1” ou nível “0”)	Bits de memória
MR0 a MR255	1 bit (nível “1” ou nível “0”)	Bits retentivos
MB0 a MB255	1 byte (de 0 a 255)	Bytes de memória
MBR0 a MBR255	1 byte (de 0 a 255)	Bytes retentivos
MW0 a MW255	2 bytes (de 0 a 65 535)	Word de memória
MWR0 a MWR255	2 bytes (de 0 a 65 535)	Word retentivo

**Tabela 5.4**  
Endereços internos

A figura 5.38 mostra que as instruções não couberam na linha lógica 1. Nesse caso, foi necessária a utilização de um bit de memória (M0) como saída. Note que a representação gráfica é a mesma de uma saída simples. Na linha lógica 2, para dar sequência, utiliza-se o endereço de instrução (M0) como entrada. Observe que a representação gráfica é a mesma de uma entrada. Esse recurso economiza saída física do CLP e permite o uso de linhas longas, com muitas instruções.

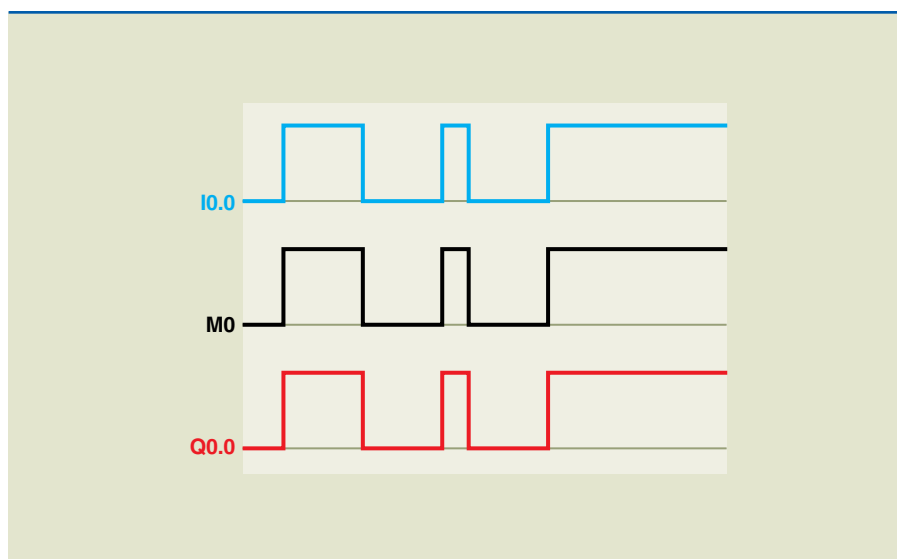
**Figura 5.38**  
Utilização de memória para sequência lógica.





O diagrama de estado da figura 5.39 demonstra a versatilidade da utilização de memória em linhas lógicas complexas. Quando a linha lógica 1 for verdadeira, a saída (M0) também será e atuará na lógica 2, que, se também for verdadeira, acionará a saída (Q0.0).

**Figura 5.39**  
Diagrama de estado utilizando memória.



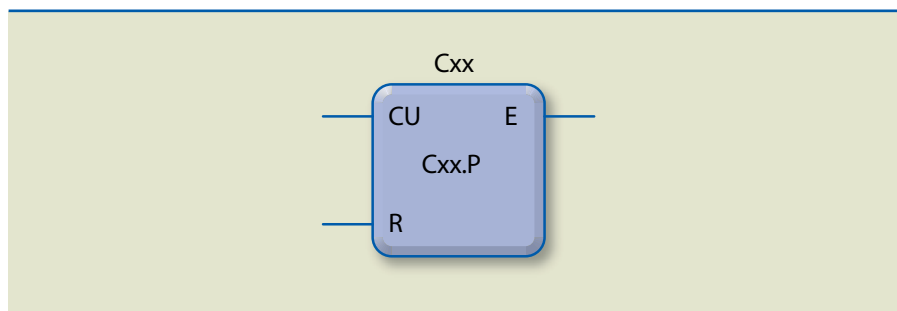
## 5.9 Contadores

Os contadores são usados quando se deseja contar o número de vezes que determinado evento ocorre – por exemplo, peças produzidas, operações realizadas etc.

### 5.9.1 Contador UP

O contador UP, denominado contador crescente (figura 5.40), incrementa uma unidade toda vez que o contato associado à entrada (CU) passa do estado lógico “0” para o “1”, até atingir o valor predeterminado (valor do *preset*). Quando o valor atual atingir o valor do *preset*, a saída (E) do contador será acionada, passando do nível lógico “0” para o “1”. Quando a entrada associada ao *reset* (R) do contador for acionada, passando do nível lógico “0” para o “1”, o valor atual do contador será zerado, podendo reiniciar a contagem assim que a entrada associada ao *reset* voltar ao estado inicial (nível lógico “0”).

**Figura 5.40**  
Símbolo gráfico do contador UP.



Observe na tabela 5.5 os parâmetros do contador UP.

**Tabela 5.5**  
Parâmetros do contador UP

	<p>Cxx: em que xx é o número do contador, de 0 a 31, definido pelo usuário.</p>
	<p>C01.P: valor do <i>preset</i>, definido pelo usuário. É um número inteiro na faixa de 0 a 65 535. C01.V: valor atual da contagem, definido por <i>software</i>. Utilizado para mostrar o número atual da contagem em uma IHM (interface homem-máquina), é incrementado por pulso aplicado na entrada (CU) do contador. É um número inteiro que varia de 0 a 65 535.</p>
	<p>CU: entrada do contador, definida pelo usuário. Recebe bit (“0” ou “1”) da entrada associada a ela.</p>
	<p>R: <i>reset</i>, definido pelo usuário. É ativado pela entrada associada a ele. Quando acionado, zera o valor atual da contagem, reiniciando-a assim que a entrada volta ao nível lógico “0”. Recebe bit (“0” ou “1”) da entrada associada a ele.</p>
	<p>E: saída do contador, definida por <i>software</i>. Quando o valor atual da contagem se iguala ao valor do <i>preset</i>, ela é ativada. Coloca nível lógico “1” no operando associado a ela.</p>

### Exemplo

Um exemplo de aplicação do contador UP é a limitação da quantidade de peças produzidas por uma máquina. Quando forem produzidas seis peças, o processo deve ser interrompido para sua retirada. Depois disso, deve ser reiniciado.



Solução:

Os materiais necessários são: dois botões de contato momentâneo com retorno por mola, um sensor indutivo e o CLP genérico. Os endereços, símbolos e comentários são apresentados na tabela 5.6.

Tabela 5.6

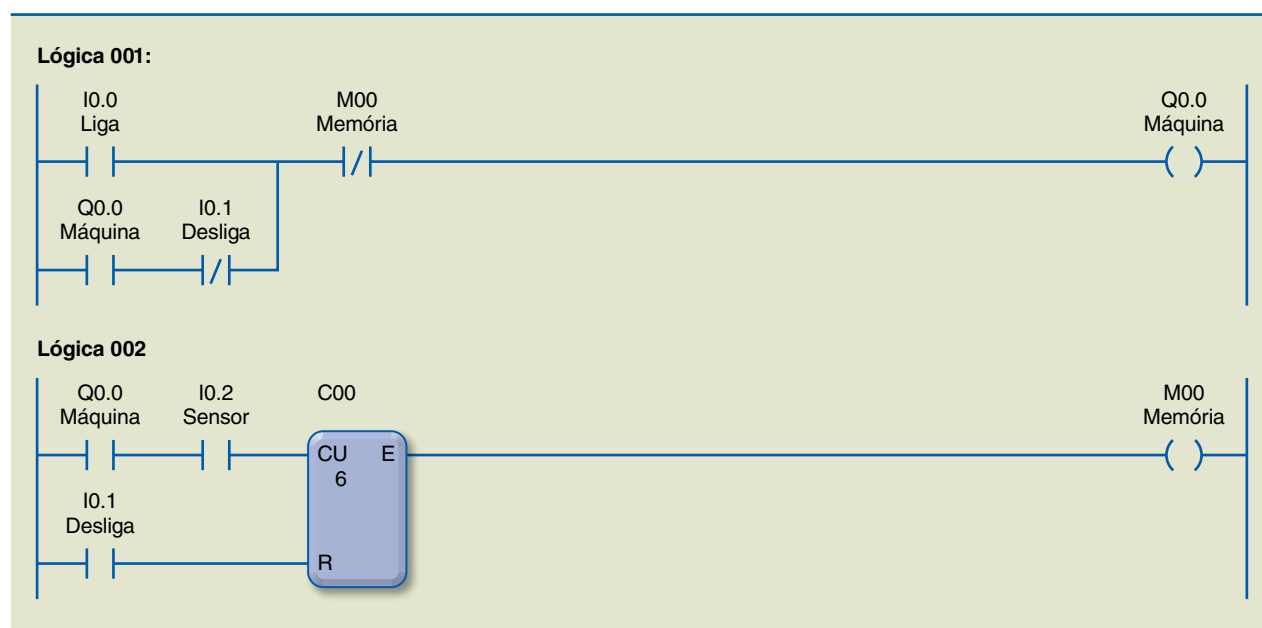
Endereços, símbolos e comentários

Endereço	Símbolo	Comentário
I0.0	Liga	Botão de contato momentâneo, retorno por mola.
I0.1	Desliga	Botão de contato momentâneo, retorno por mola.
I0.2	Sensor	Sensor indutivo colocado na entrada do contador UP.
Q0.0	Máquina	Saída do CLP aciona a máquina.
M0	Bit de memória	Quando acionado, desliga a máquina.

Figura 5.41

Diagrama Ladder.

A figura 5.41 mostra o programa em diagrama Ladder.



No exemplo, alguns importantes conceitos de comandos elétricos são utilizados na primeira linha (*rung*). A memória (M00) está associada em série com o endereço liga (I0.0). Essa linha só será verdadeira se a memória (M00) não estiver acionada, estabelecendo uma dependência de funcionamento, ou seja, a máquina (Q0.0) somente será acionada se a memória (M00) estiver desacionada.

Ainda na lógica 1, a segunda linha utiliza o conceito de selo elétrico executado pelo endereço de saída da máquina (Q0.0). Essa lógica depende apenas do endereço liga (I0.0) para acionar a saída, que se mantém fechada por meio do endereço

de saída da máquina (Q0.0) realocado como entrada. O desligamento da máquina depende do endereço desliga (I0.1) ou do endereço de memória (M00).

Na lógica 2, o contador foi parametrizado para contar até 6. Quando o valor atual atinge o valor do *preset*, a saída (E) do contador vai para nível lógico "1", acionando a memória (M00), também com nível lógico "1". Ao ser acionada, desliga o selo da lógica 1.

Os pulsos para a contagem são fornecidos pelo sensor (I0.2), mas somente são válidos quando a entrada de máquina (Q0.0) está acionada, ou seja, no nível lógico "1". Quando a memória (M00) é acionada, o sistema fica paralisado, podendo ser reiniciado ao acionar o operando desliga (I0.1), que resseta o contador e prepara a lógica "1" para ser acionada. O operando liga (I0.0) reinicia o processo. A figura 5.42 mostra o diagrama de estado.

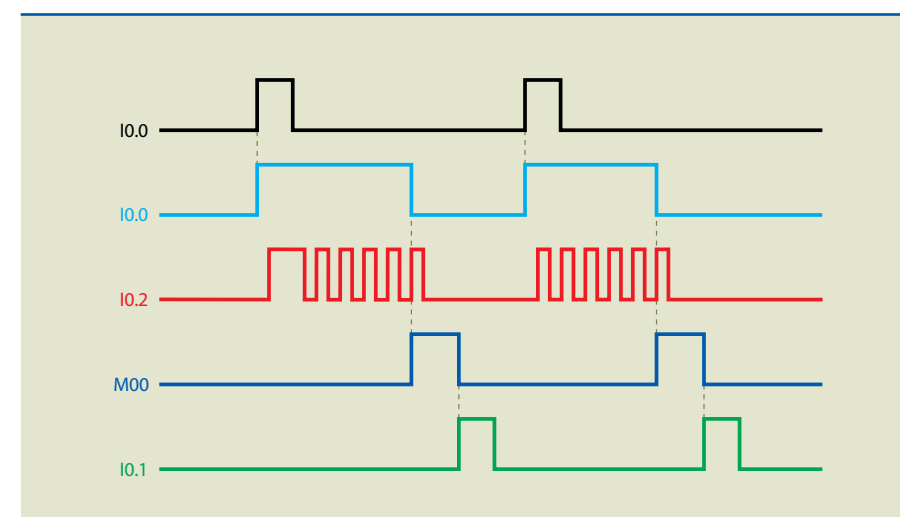


Figura 5.42

Diagrama de estado.

### 5.9.2 Contador DOWN

Nesse tipo de contador, o valor atual (Cxx.V) é carregado com o valor do *preset* (Cxx.P). O contador DOWN, denominado contador decrescente (figura 5.43), decremente uma unidade toda vez que o contato associado à entrada (CD) passa do estado lógico "0" para o "1". Quando o valor atual do contador chegar a zero, a saída (E) do contador será acionada, passando do nível lógico "0" para o "1". Quando a entrada associada ao *preset* (P) do contador for acionada, passando do nível lógico "0" para o "1", o valor atual do contador será carregado com o valor do *preset* e sua saída voltará ao estado inicial (nível lógico "0"), podendo reiniciar a contagem.

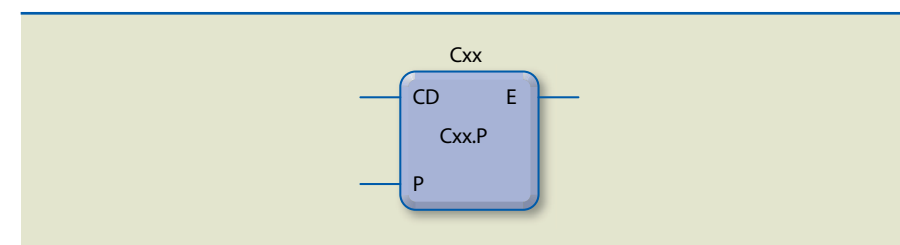


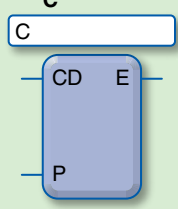
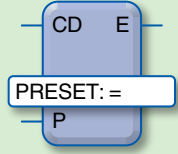
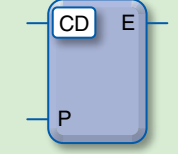
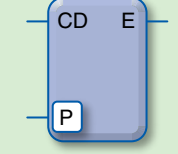
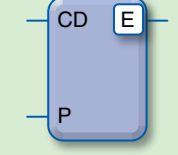
Figura 5.43

Símbolo gráfico do contador DOWN.



**Tabela 5.7**  
Parâmetros do contador DOWN

A tabela 5.7 apresenta os parâmetros do contador DOWN.

	<p>Cxx: em que xx é o número do contador, de 0 a 31, definido pelo usuário.</p>
	<p>Cxx.P: valor do <i>preset</i>, definido pelo usuário. É um número inteiro na faixa de 0 a 65 535. Cxx.V: valor atual da contagem, definido por <i>software</i>. Utilizado para mostrar o número atual da contagem em uma IHM (interface homem-máquina), é decrementado por pulso aplicado na entrada (CD) do contador. É um número inteiro que varia de 0 a 65 535.</p>
	<p>CD: entrada do contador, definida pelo usuário. Recebe bit ("0" ou "1") da entrada associada a ela.</p>
	<p>P: <i>preset</i>, definido pelo usuário. É ativado pela entrada associada a ele. Quando acionado, carrega com o valor do <i>preset</i> o valor atual. Recebe bit ("0" ou "1") da entrada associada a ele.</p>
	<p>E: saída do contador, definida por <i>software</i>. Quando o valor atual da contagem se iguala a zero, ela é ativada. Coloca bit ("0" ou "1") no operando associado a ela.</p>

**Exemplo**

Em um sistema produtivo, foi implantado um dispositivo que retira peças de uma esteira e as coloca em um recipiente que comporta seis peças. Quando o recipiente estiver cheio, o sistema deve parar a esteira e sinalizar ao operador para a retirada do recipiente com as peças produzidas. Quando o operador colocar um recipiente vazio na esteira, o sistema deve reiniciar.

*Solução:*

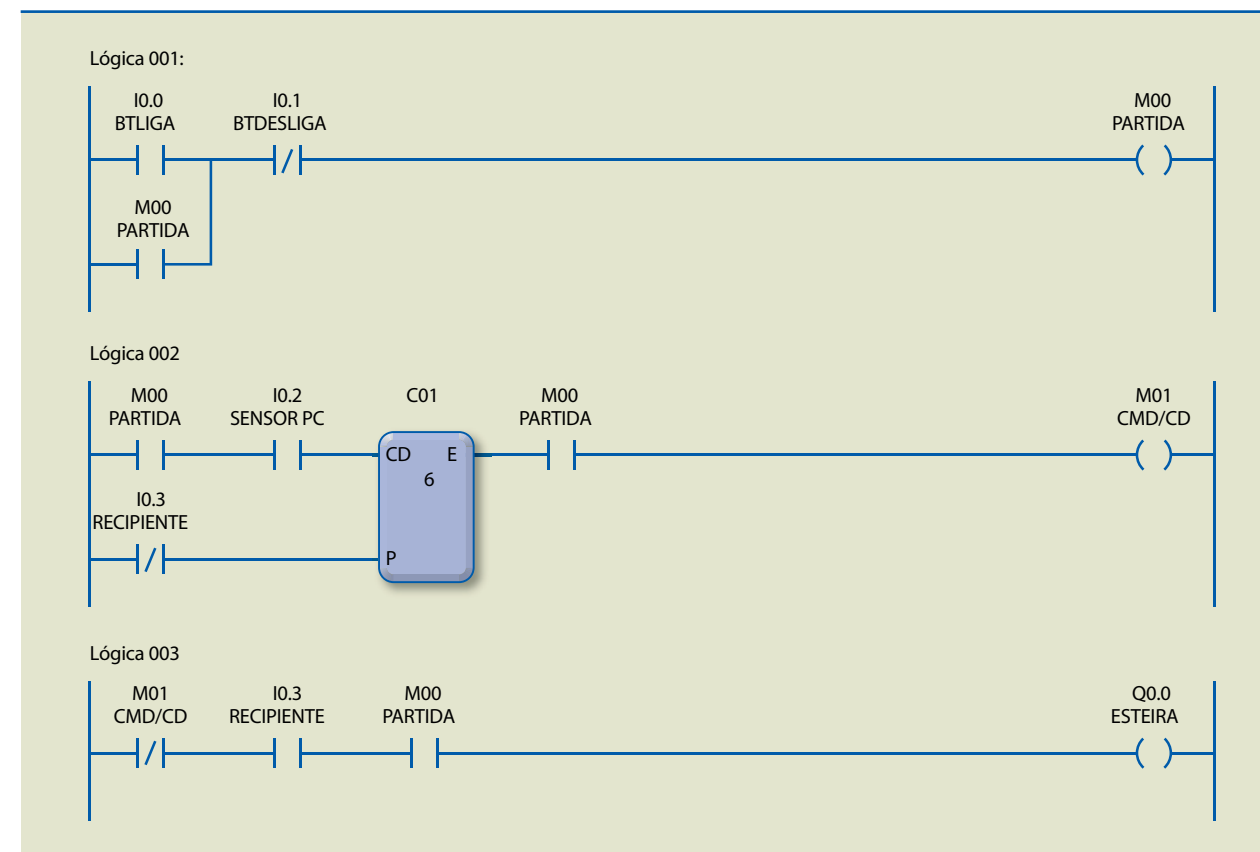
Os materiais necessários são: dois botões de contato momentâneo com retorno por mola, um sensor capacitivo, uma chave fim de curso e o CLP genérico. Os endereços, símbolos e comentários são apresentados na tabela 5.8.

Endereço	Símbolo	Comentário
I0.0	BTLiga	Botão de contato momentâneo, retorno por mola.
I0.1	BTDesliga	Botão de contato momentâneo, retorno por mola.
I0.2	Sensor	Sensor capacitivo colocado na entrada do contador DOWN.
I0.3	Chave	Chave fim de curso para detectar presença de recipiente.
M00	Partida	Bit de memória inicia o processo.
M01	CMD/CD	Bit de memória da saída do contador bloqueia/libera a esteira.
Q0.0	Esteira	Acionamento da esteira.

**Tabela 5.8**  
Endereços, símbolos e comentários

A figura 5.44 mostra o programa em diagrama Ladder.

**Figura 5.44**  
Diagrama Ladder.



O programa para esse exemplo foi feito com três lógicas, porém existem outras maneiras de executar a mesma tarefa. Essas lógicas também ilustram conceitos da programação Ladder.

A lógica 1 serve para ligar e desligar o sistema. O conceito de selo elétrico surge de novo e a memória é utilizada como recurso para economizar saída.

A lógica 2 apresenta o funcionamento do contador DOWN. O endereço *start* (M00) aparece como bloqueio, para que não haja contagem sem o sistema estar ligado. O endereço recipiente (I0.3), quando acionado, atualiza o valor do contador, preparando-o para uma nova contagem. O endereço CMD/CD (M01) atua diretamente sobre o funcionamento da esteira. Quando acionado, desliga a esteira e, quando desacionado, prepara o sistema para ser ligado.

A lógica 3 tem o funcionamento dependente de três endereços: (M01), (I0.3) e (M00). Quando os três são verdadeiros, a esteira (Q0.0) é acionada.

### 5.9.3 Contador rápido

O contador rápido executa a lógica que está programada no CLP em função da entrada rápida. No diagrama Ladder, a entrada rápida tem até oito contadores rápidos, que são habilitados por diferentes bits de entrada e podem executar a contagem de quantidades distintas de pulsos e ser ressetados por diferentes bits.

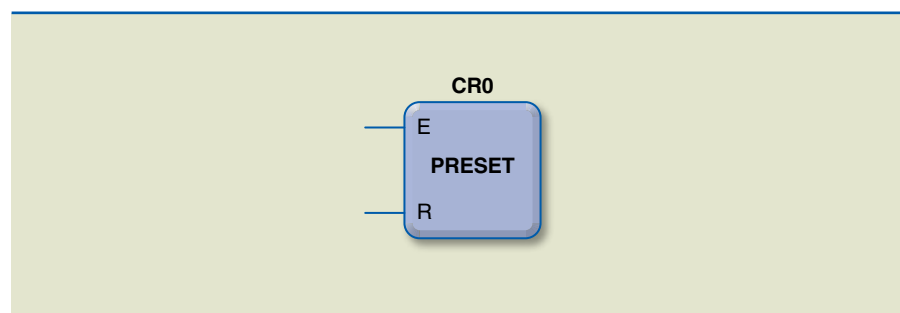
Nesse contador, a contagem dos pulsos é recebida pela entrada rápida a uma frequência máxima de 4 kHz. A entrada é pelos pinos CR0+ e CR0-. Os pulsos recebidos são incrementados no valor atual da contagem (CRx.V) no bloco do contador rápido.

O funcionamento do contador rápido (figura 5.45) se dá da seguinte maneira: quando a entrada *enable* (E) está habilitada, é feita uma comparação do valor atual (CRx.V) com o valor do *preset* (CRx.P) do bloco do contador rápido. Quando o valor atual for igual ou maior que o valor do *preset*, o bit relacionado ao bloco (CRx.Q) será acionado. Se a entrada *enable* (E) não estiver habilitada, não será efetuada a comparação, mas o valor atual da visualização continuará sendo incrementado a cada pulso recebido na entrada rápida.

O valor atual (CRx.V) do bloco do contador rápido pode ser zerado a qualquer instante com o acionamento do *reset* (R).

Figura 5.45

Símbolo gráfico do contador rápido.



Parâmetros do contador rápido:

- **CRx** — Número do bloco do contador rápido de 0 a 7, definido pelo usuário. Pode haver até oito blocos em apenas um programa.
- **CRx.P** — Valor do *preset* do contador, definido pelo usuário. É um número inteiro na faixa de 0 a 65 535.
- **CRx.V** — Valor atual da contagem, definido por *software*, incrementado a cada pulso recebido pela entrada (E) do contador. É um número inteiro na faixa de 0 a 65 535.
- **R** — *Reset* do contador, definido pelo usuário. Bit (0 ou 1).
- **E**: entrada *enable* do contador, definido pelo usuário. Bit (0 ou 1).
- **CRx.Q** — *Status* de saída do contador rápido. Bit (0 ou 1). Indica que o valor atual do contador rápido (CRx.V) já chegou ao valor do *preset* (CRx.P), acionando essa saída.

#### Exemplo

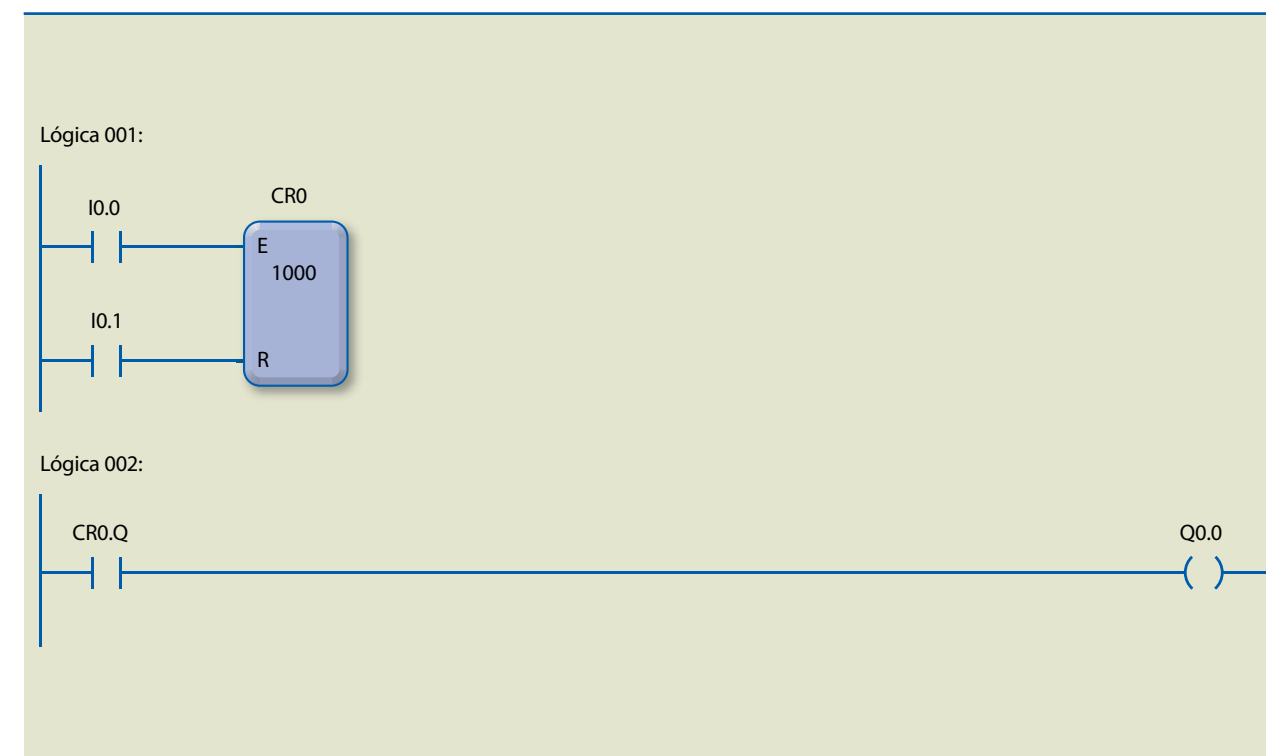
O programa deve fazer a contagem de pulsos da entrada rápida e acionar a saída (Q0.0) quando o contador rápido contar 1 000 pulsos.

*Solução:*

A figura 5.46 apresenta o programa em diagrama Ladder.

Figura 5.46

Diagrama Ladder.



O programador, ao utilizar os contadores UP, DOWN e Rápido, deve obedecer às regras descritas a seguir.

Na saída dos contadores, os parâmetros aceitos são valores com tamanho de 1 bit (nível lógico “0” ou “1”), ou seja, variáveis também com tamanho de 1 bit. Elas podem ser:

- Q<sub>xx</sub> (saídas digitais).
- M<sub>xx</sub> (bits de memória).
- MR<sub>xx</sub> (bits de memória retentiva).
- LT<sub>xx</sub> (LED da IHM).

No valor do *preset* dos contadores, os parâmetros aceitos são valores com tamanho de 1 *word* (valores inteiros de 0 a 65 535).

## 5.10 Temporizadores

Em sistemas automatizados, é comum incluir a variável tempo no processo. Nesses casos, o temporizador é utilizado para definir o intervalo de tempo entre duas operações, verificar se uma operação ocorre dentro do tempo esperado ou definir o tempo de duração de uma operação. Em geral, podem existir três tipos de temporizadores e em nosso CLP genérico há: temporizador na energização (TON), temporizador na desenergização (TOFF) e temporizador de pulso (TP). Os parâmetros e operandos aqui apresentados são aceitos nesses três tipos.

Ao utilizar os temporizadores TON, TOFF e de pulso, o programador deve obedecer às seguintes regras:

- **Txx** — Número do temporizador, de 0 a 31, definido pelo usuário.
- **Time base** — Base de tempo do temporizador (1 s, 0,1 s e 0,01 s), definida pelo usuário.
- **Txx.P** — Valor do *preset* do temporizador, definido pelo usuário. É um número inteiro na faixa de 0 a 65 535.
- **Txx.V** — Valor atual do temporizador, definido por *software*. É um número inteiro na faixa de 0 a 65 535.
- **Q** — *Status* da saída do temporizador, definido por *software*. É ativado quando o valor atual do temporizador se iguala ao valor do preset. Bit (0 ou 1).
- **E** — *Enable* do temporizador, definido pelo usuário. Quando ativado, faz a contagem do tempo. Bit (0 ou 1).

Na saída (Q) do temporizador, os operandos aceitos são aqueles com tamanho de 1 bit (nível lógico 0 ou 1). Essas variáveis podem ser:

- Q<sub>xx</sub> (saídas digitais).
- M<sub>xx</sub> (bits de memória).
- MR<sub>xx</sub> (bits de memória retentiva).
- LT<sub>xx</sub> (LED da IHM).

No valor do *preset* do temporizador, os parâmetros aceitos são valores com tamanho de 1 *word* (valor de 0 a 65 535).

### 5.10.1 Temporizador na energização (TON)

Esse tipo de temporizador (figura 5.47) causa retardo na energização de sua saída. Para isso, ele inicia a contagem do tempo a partir do instante em que a entrada *enable* (E) é habilitada, passando do nível lógico “0” para o “1”. Quando o valor atual do temporizador (Txx.V) se igualar ao tempo do *preset* (Txx.P), a saída do temporizador será acionada, passando do nível lógico “0” para o “1”. Se, a qualquer instante, a entrada *enable* (E) for desabilitada, passando do nível lógico “1” para o “0”, o valor atual do temporizador (Txx.V) será zerado e sua saída (Q) será desabilitada, retornando ao estado inicial, ou seja, nível lógico “0”.

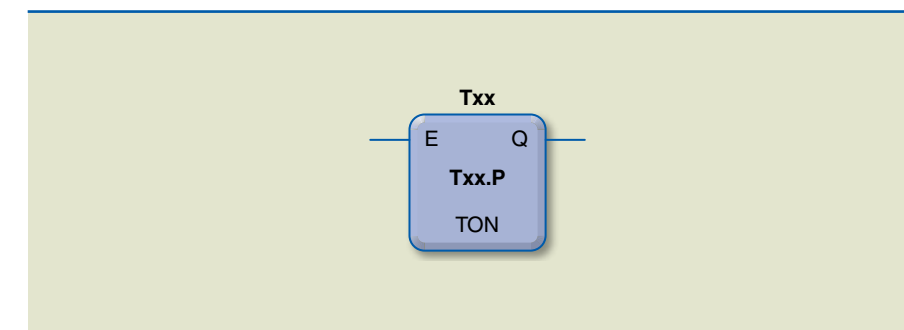


Figura 5.47

Símbolo gráfico do temporizador TON.

#### Exemplo

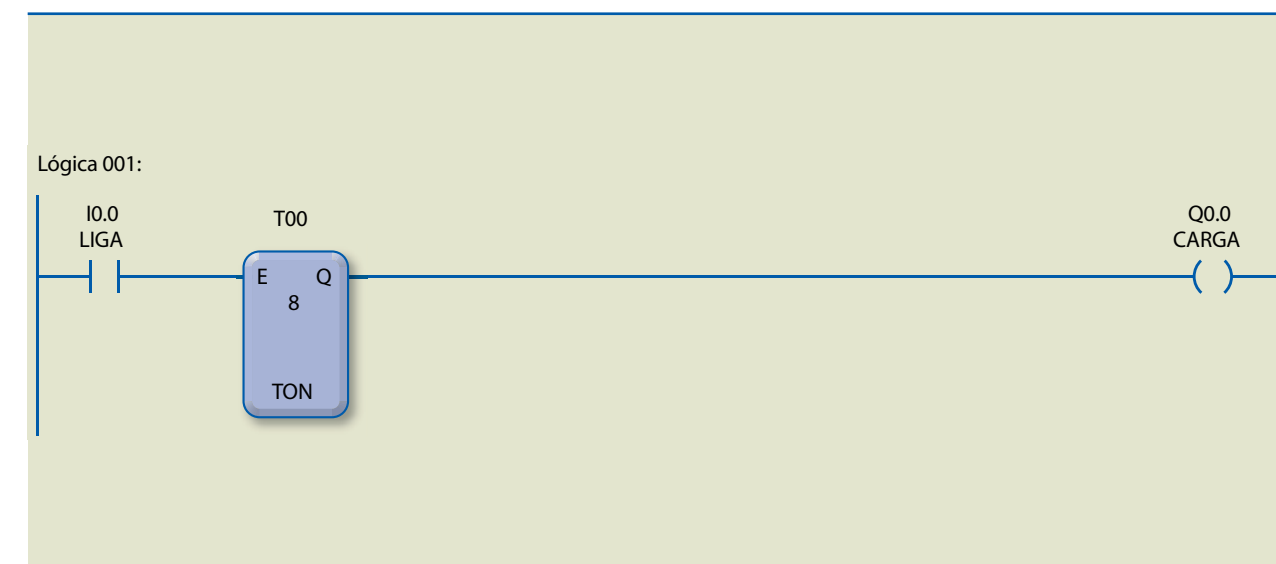
Deseja-se uma programação que acione uma carga que entre em funcionamento 8 segundos depois de o botão liga ser pressionado e desligue em qualquer instante em que o botão for desacionado.

*Solução:*

A figura 5.48 apresenta o programa em diagrama Ladder.

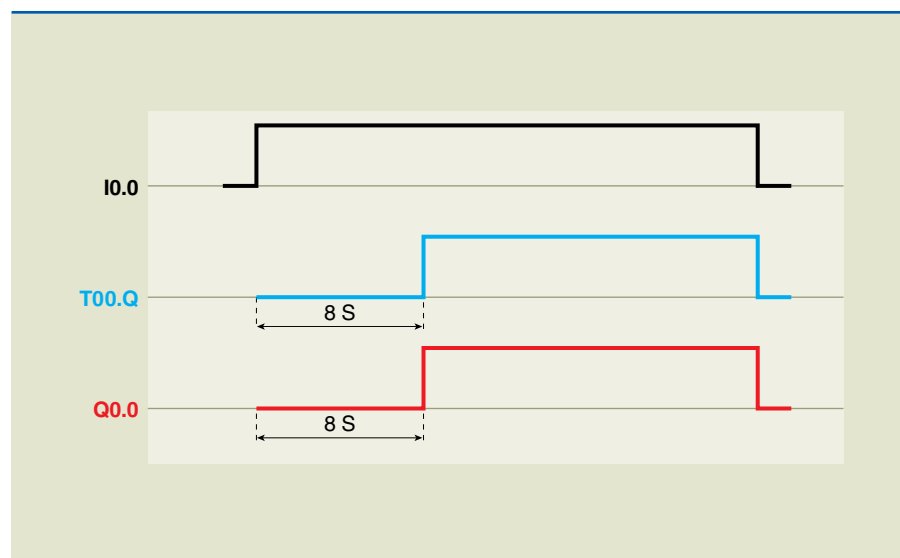
Figura 5.48

Diagrama Ladder.



A figura 5.49 mostra o diagrama de estado.

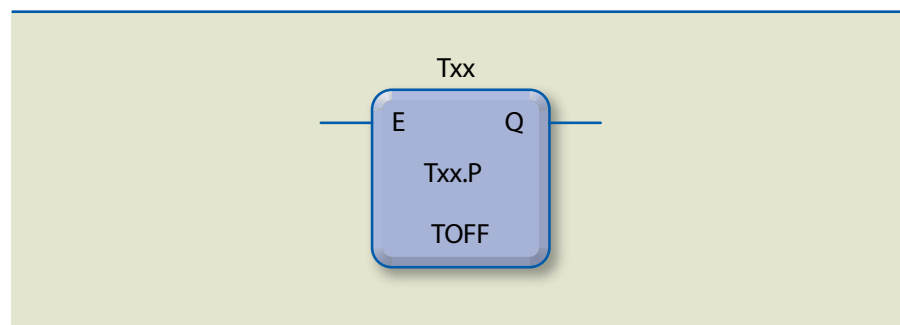
**Figura 5.49**  
Diagrama de estado.



### 5.10.2 Temporizador na desenergização (TOFF)

Nesse tipo de temporizador (figura 5.50), ao acionarmos a entrada *enable* (E), o valor atual do temporizador (Txx.V) zera e a saída (Txx.Q) passa para o nível lógico “1”, acompanhando a entrada. O retardo acontece na desenergização, ou seja, quando desacionamos a entrada *enable* (E), passando do nível lógico “1” para o “0”, inicia-se a contagem do tempo que causará o retardo na saída (Txx.Q). A partir do instante em que o valor atual do temporizador (Txx.V) se igualar ao tempo do *preset* (Txx.P), a saída do temporizador será desacionada, passando do nível lógico “1” para o “0”.

**Figura 5.50**  
Símbolo gráfico do temporizador TOFF.

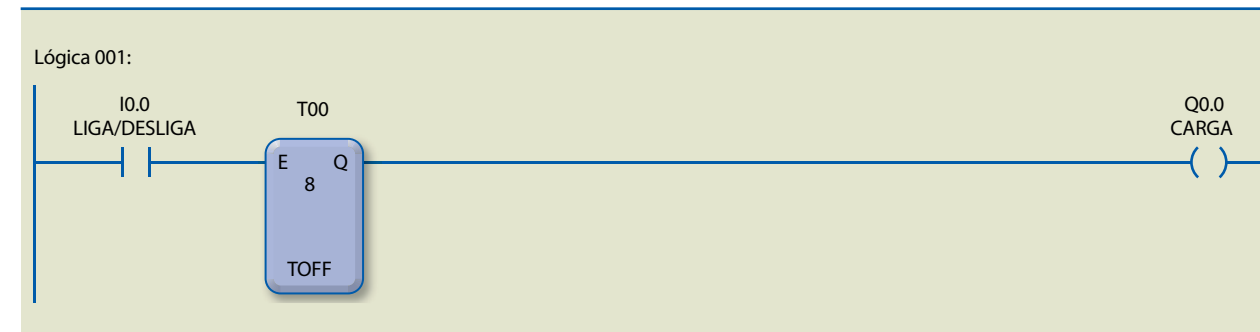


Exemplo

Deseja-se desacionar uma carga que interrompa seu funcionamento 8 segundos depois de o botão desliga ser acionado.

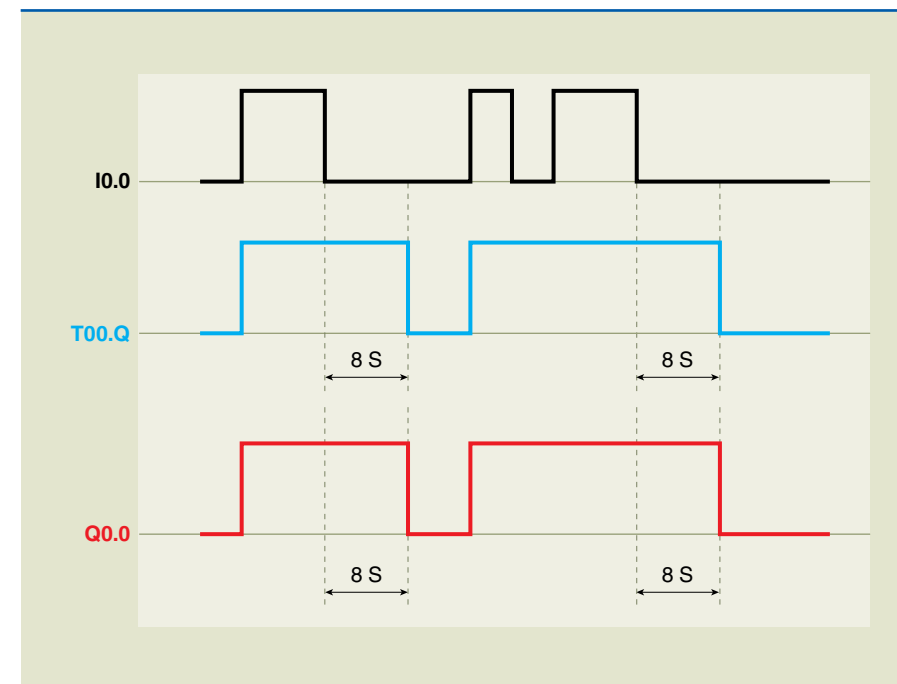
*Solução:*

A figura 5.51 apresenta o programa em diagrama Ladder.



**Figura 5.51**  
Diagrama Ladder.

A figura 5.52 mostra o diagrama de estado.



**Figura 5.52**  
Diagrama de estado.

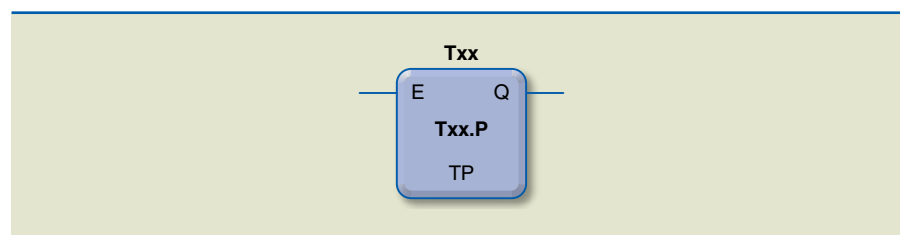
### 5.10.3 Temporizador de pulso (TP)

Nesse tipo de temporizador (figura 5.53), se, ao acionarmos a entrada *enable* (E), o pulso de entrada for menor que o tempo do *preset* (Txx.P) do temporizador, a saída será igual à entrada. Se a entrada permanecer acionada por tempo maior que o tempo do *preset* (Txx.P) do temporizador, a saída ficará acionada somente pelo tempo do *preset* (Txx.P), gerando um pulso na saída.

A partir do instante em que o valor atual do temporizador (Txx.V) se igualar ao tempo do *preset* (Txx.P), a saída (Q) do temporizador será desativada, passando do nível lógico “1” para o “0”. Um fato relevante é que existem pequenas variações no comportamento desse *timer*, dependendo do fabricante. Alguns modelos, por exemplo, mantêm a saída do *timer* ativada mesmo para pulsos curtos no *enable*, desligando após Txx.P. Dessa maneira, recomenda-se consultar o manual do fabricante.



**Figura 5.53**  
Símbolo gráfico do temporizador TP.



O funcionamento desse tipo de instrução pode ser esclarecido no próximo exemplo.

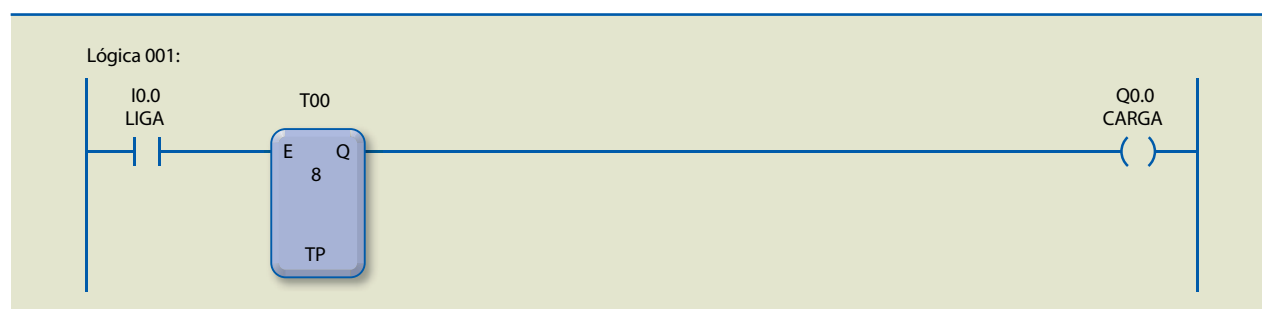
**Exemplo**

Deseja-se acionar uma carga que interrompa seu funcionamento 8 segundos depois de o botão liga ser acionado. Se desligarmos o botão liga a qualquer instante antes de decorridos os 8 segundos, ele interrompe o funcionamento de imediato.

*Solução:*

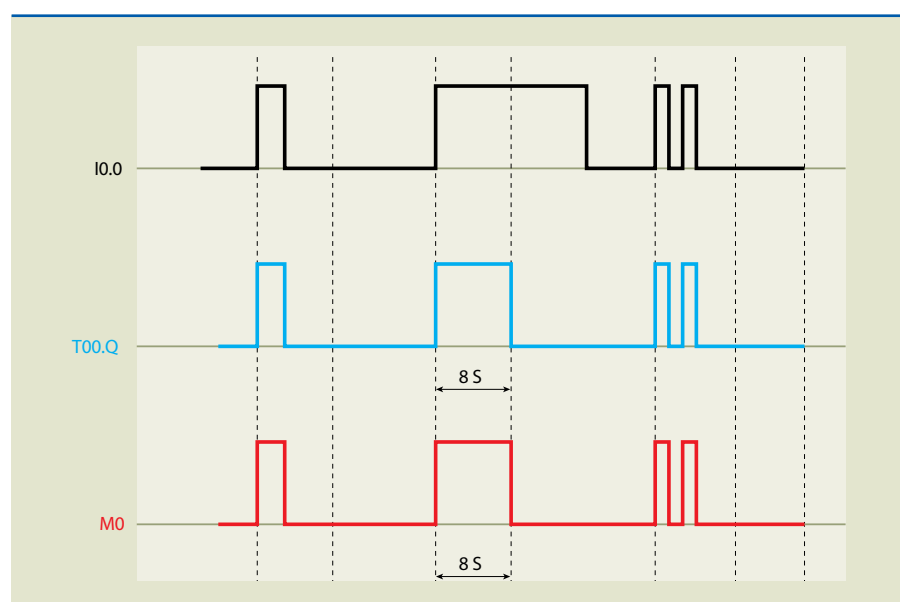
**Figura 5.54**  
Diagrama Ladder.

A figura 5.54 apresenta o programa em diagrama Ladder.



A figura 5.55 mostra o diagrama de estado.

**Figura 5.55**  
Diagrama de estado.



**5.11 Entradas e saídas analógicas: endereçamento**

Para que trabalhem com sinais analógicos, os CLPs necessitam de um conversor analógico-digital (A/D) nas entradas e, de modo similar, de um conversor digital-analógico (D/A) nas saídas.

A tabela 5.9 apresenta as características das entradas analógicas para o CLP genérico.

Entradas analógicas	
IA0	0 a 10V <sub>cc</sub> ou 0 a 20 mA
IA1	0 a 10V <sub>cc</sub> ou 0 a 20 mA
IA2	0 a 10V <sub>cc</sub>
IA3	0 a 10V <sub>cc</sub>
Resolução em tensão	12 bits (2,44 mV)
Impedância em tensão	10 kΩ
Resolução em corrente	12 bits (4,8 μA)
Impedância em corrente	500 Ω
Quantidade	4

**Tabela 5.9**  
Características das entradas analógicas

A tabela 5.10 mostra o endereçamento das entradas analógicas.

Quantidade de entradas	Endereço	Descrição
4	IA0 a IA3	12 bits de resolução referentes às entradas analógicas

**Tabela 5.10**  
Endereçamento das entradas analógicas

A tabela 5.11 apresenta as características das saídas analógicas.

Saídas analógicas	
QA0	0 a 10V <sub>cc</sub>
QA1	0 a 10V <sub>cc</sub>
Resolução	8 bits (39 mV)
Impedância	200 Ω
Quantidade	2

**Tabela 5.11**  
Características das saídas analógicas



A tabela 5.12 mostra o endereçamento das saídas analógicas.

Quantidade de saídas	Endereço	Descrição
2	QA0 e QA1	8 bits de resolução referentes às saídas analógicas

**Tabela 5.12**

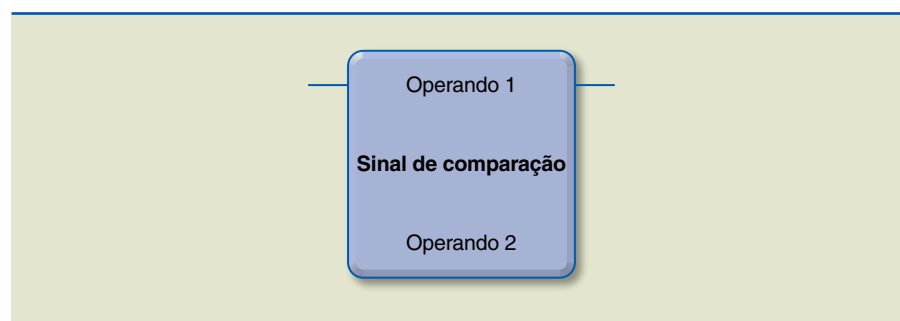
Endereçamento das saídas analógicas

## 5.12 Instruções de comparação

Em programação, muitas vezes é necessário comparar dois valores. Para isso, pode-se usar as instruções de comparação. Os comparadores utilizam dois operandos, que podem ser byte, *word* ou constante. O programa é realizado com os operandos 1 e 2 (figura 5.56). Caso os valores satisfaçam a condição de comparação e a entrada do comparador esteja habilitada, a saída do comparador será acionada, habilitando, assim, a saída do sistema.

**Figura 5.56**

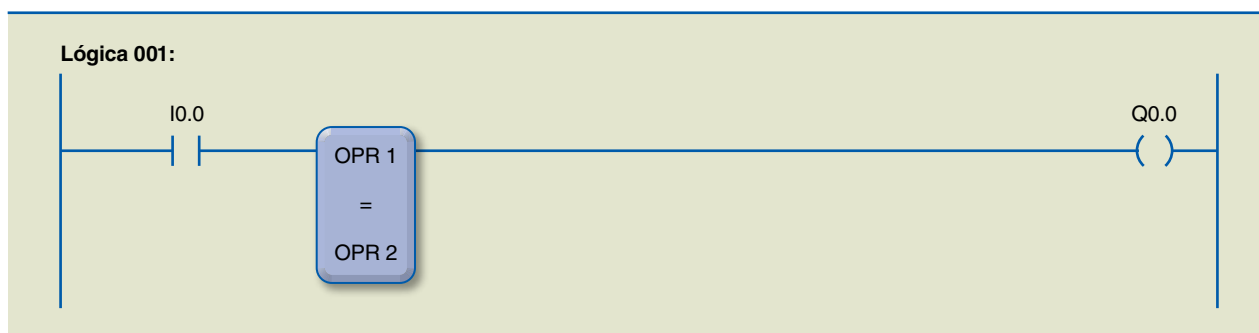
Símbolo gráfico do comparador.



As comparações que podem ser feitas são: igual, maior que, menor que, maior ou igual a, menor ou igual a e diferente.

**Figura 5.57**

Diagrama Ladder da instrução de comparação igual a (=).



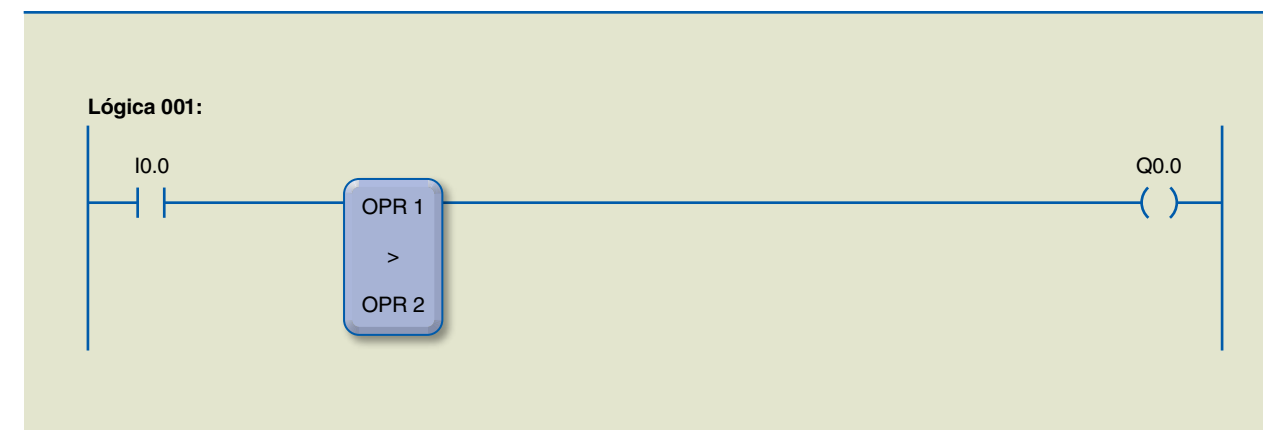
Nesse exemplo, quando a entrada I0.0 estiver habilitada, teremos a comparação entre o operando 1 e o operando 2. Se eles forem iguais, o resultado será nível lógico "1" e a saída será acionada. Se forem diferentes, o resultado será nível lógico "0" e a saída será desligada.

### 5.12.2 Maior que (>)

A figura 5.58 apresenta o programa da instrução maior que (>) em diagrama Ladder.

**Figura 5.58**

Diagrama Ladder da instrução maior que (>).



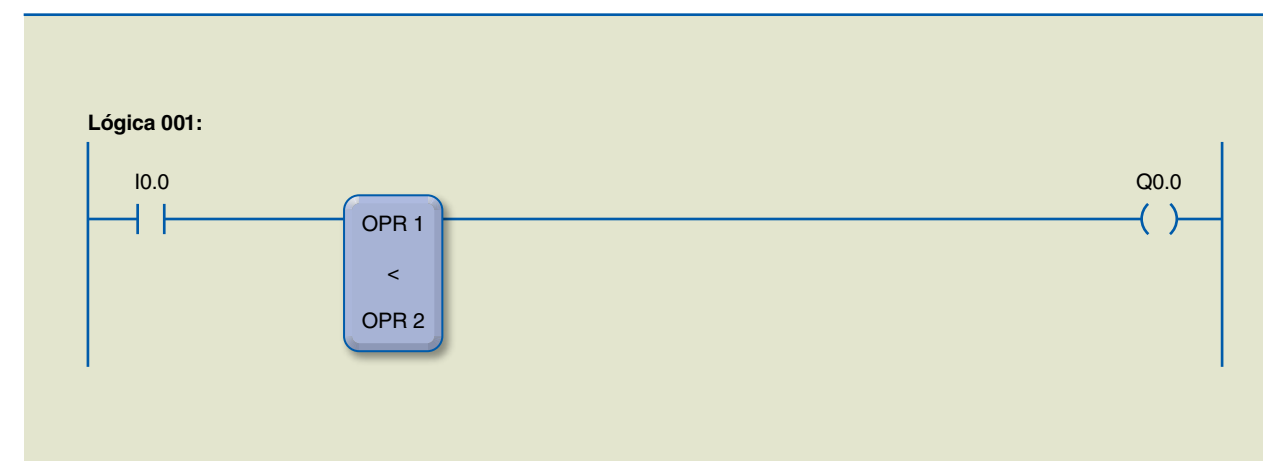
Nesse exemplo, quando a entrada I0.0 estiver habilitada, terá a comparação entre o operando 1 e o operando 2. Se o operando 1 for maior que o operando 2, o resultado terá nível lógico "1" e a saída será acionada. Se o operando 1 for menor que o operando 2, o resultado terá nível lógico "0" e a saída será desligada.

### 5.12.3 Menor que (<)

A figura 5.59 apresenta o programa da instrução menor que (<) em diagrama Ladder.

**Figura 5.59**

Diagrama Ladder da instrução menor que (<).



Nesse exemplo, quando a entrada I0.0 estiver habilitada, terá a comparação entre o operando 1 e o operando 2. Se o operando 1 for menor que o operando 2, o resultado terá nível lógico "1" e a saída será acionada. Se o operando 1 for maior ou igual ao operando 2, o resultado terá nível lógico "0" e a saída será desligada.



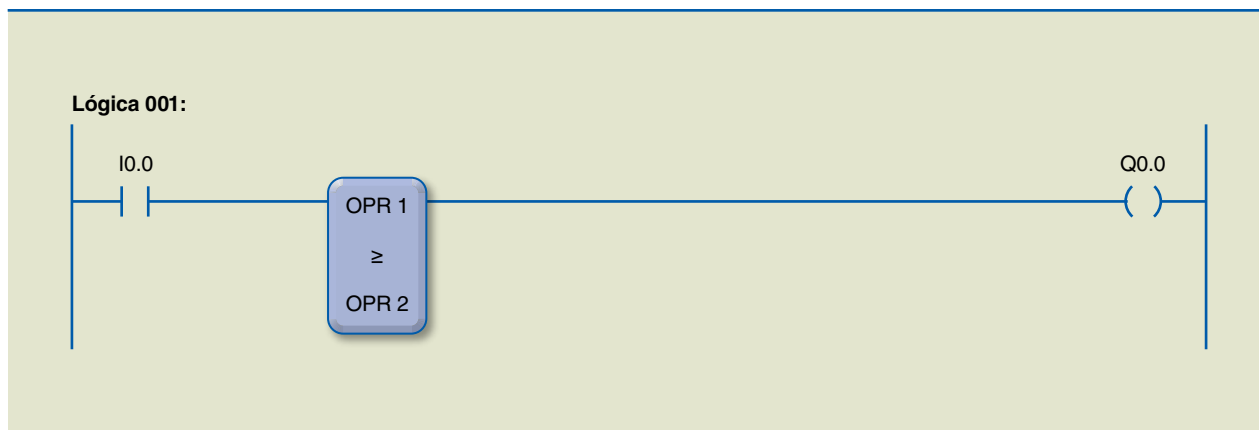


**Figura 5.60**

Diagrama Ladder da instrução maior ou igual a ( $\geq$ ).

### 5.12.4 Maior ou igual a ( $\geq$ )

A figura 5.60 apresenta o programa da instrução maior ou igual a ( $\geq$ ) em diagrama Ladder.



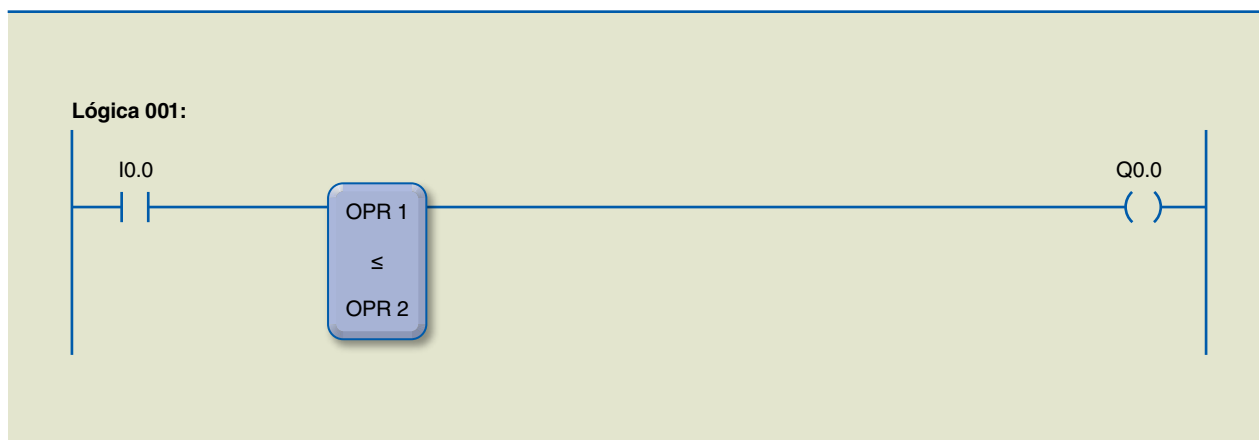
Nesse exemplo, quando a entrada I0.0 estiver habilitada, terá a comparação entre o operando 1 e o operando 2. Se o operando 1 for maior ou igual ao operando 2, o resultado terá nível lógico "1" e a saída será acionada. Se o operando 1 for menor que o operando 2, o resultado terá nível lógico "0" e a saída será desligada.

**Figura 5.61**

Diagrama Ladder da instrução menor ou igual a ( $\leq$ ).

### 5.12.5 Menor ou igual a ( $\leq$ )

A figura 5.61 apresenta o programa da instrução menor ou igual a ( $\leq$ ) em diagrama Ladder.



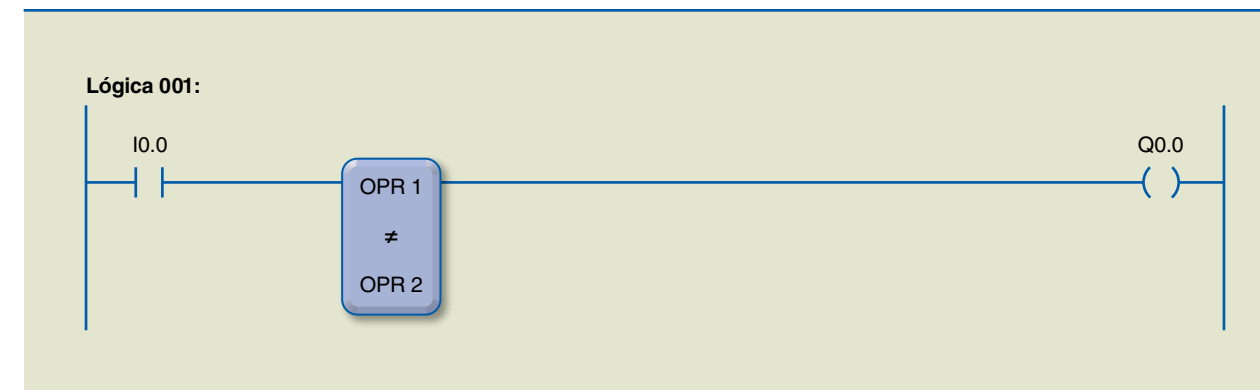
Nesse exemplo, quando a entrada I0.0 estiver habilitada, terá a comparação entre o operando 1 e o operando 2. Se o operando 1 for menor ou igual ao operando 2, o resultado será nível lógico "1" e a saída será acionada. Se o operando 1 for maior que o operando 2, o resultado será nível lógico "0" e a saída será desligada.

### 5.12.6 Diferente de ( $\neq$ )

A figura 5.62 apresenta o programa da instrução de comparação diferente de ( $\neq$ ) em diagrama Ladder.

**Figura 5.62**

Diagrama Ladder da instrução diferente de ( $\neq$ ).



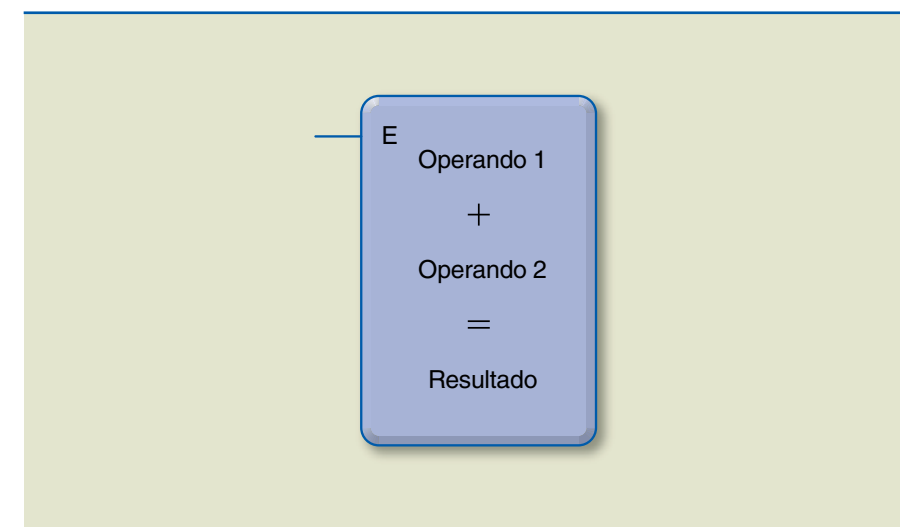
Nesse exemplo, quando a entrada I0.0 estiver habilitada, terá a comparação entre o operando 1 e o operando 2. Se os operandos forem diferentes, o resultado terá nível lógico "1" e a saída será acionada. Se o operando 1 for igual ao operando 2, o resultado terá nível lógico "0" e a saída será desligada.

## 5.13 Operações matemáticas

Essas instruções têm como função executar operações aritméticas entre dois operandos, colocando o resultado em um operando de resposta denominado RES.

### 5.13.1 Somador (+)

Quando essa instrução é habilitada por meio da entrada (E), executa-se a soma dos operandos ( $OPR1 + OPR2$ ), colocando o resultado em RES, conforme demonstra a figura 5.63.



**Figura 5.63**

Símbolo gráfico do somador.



Quando a entrada está habilitada, tem-se a execução da soma dos dois operandos ou constantes e, conseqüentemente, o valor passa para um terceiro operando.

**Operando 1** e **operando 2** são os valores que serão somados.

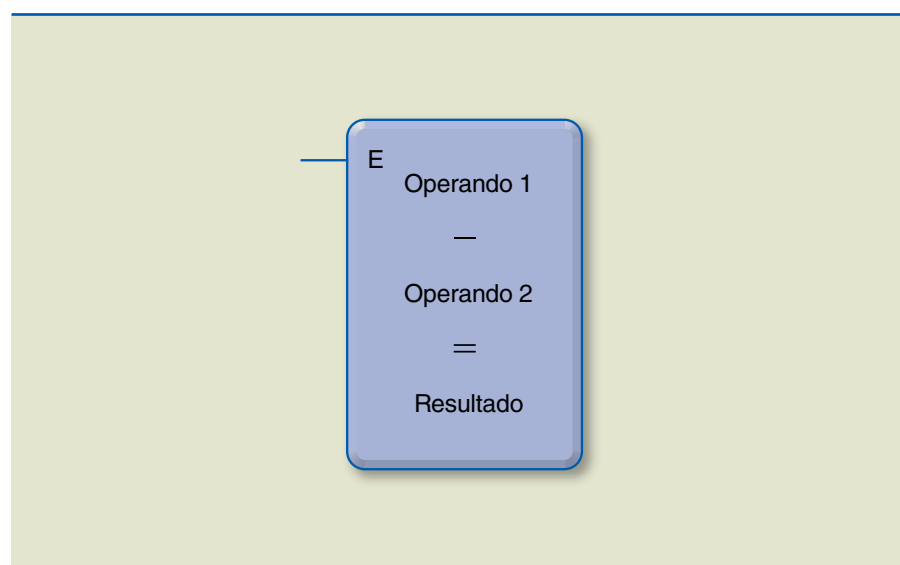
Os endereços permitidos são: bytes, *words* e constantes.

**Resultado** é o operando que receberá o resultado da soma.

### 5.13.2 Subtrator (–)

Quando essa instrução é habilitada por meio da entrada (E), executa-se a subtração dos operandos ( $OPR1 - OPR2$ ), colocando o resultado em RES, conforme demonstra a figura 5.64.

**Figura 5.64**  
Símbolo gráfico do subtrator:



**Operando 1** é o valor do qual será subtraído o valor do segundo operando e, conseqüentemente, o resultado passa para um terceiro operando.

**Operando 2** é o valor que será subtraído do primeiro operando.

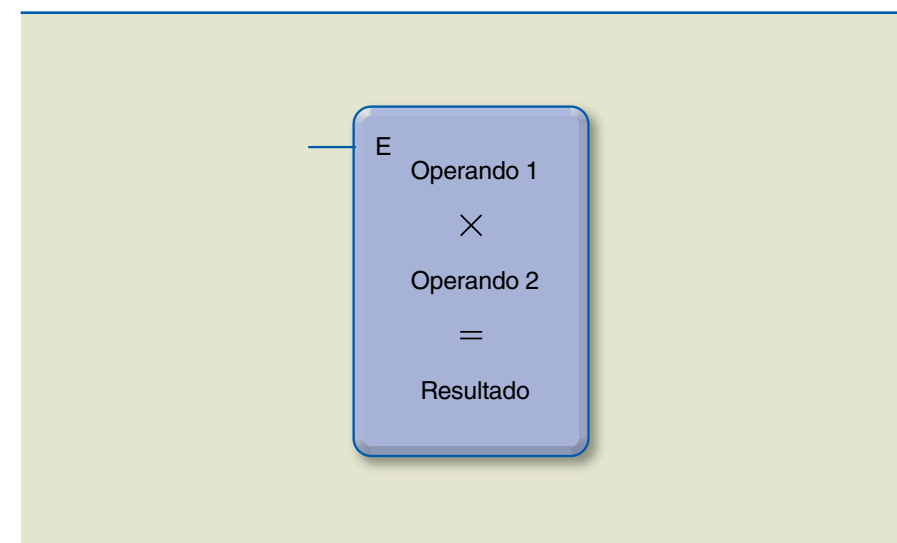
Os endereços permitidos são: bytes, *words* e constantes.

**Resultado** é o operando que receberá o resultado da subtração.

Os endereços permitidos são: bytes e *words*.

### 5.13.3 Multiplicador (×)

Quando essa instrução é habilitada por meio da entrada (E), executa-se a multiplicação dos operandos ( $OPR1 \times OPR2$ ), colocando o resultado em RES, conforme demonstra a figura 5.65.



**Figura 5.65**  
Símbolo gráfico do multiplicador:

Quando a entrada está habilitada, tem-se a execução da multiplicação dos dois operandos e, conseqüentemente, o valor passa para um terceiro operando.

**Operando 1** e **operando 2** são os valores que serão multiplicados.

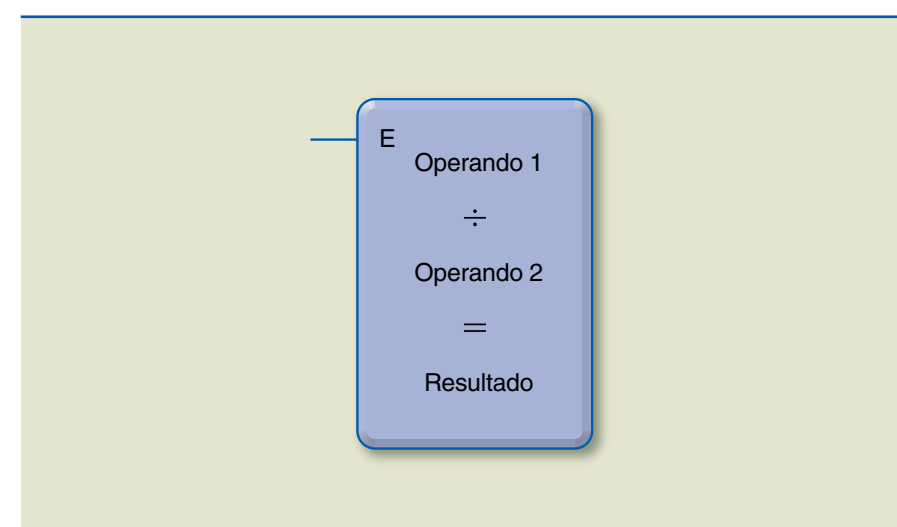
Os endereços permitidos são: bytes, *words* e constantes.

**Resultado** é o operando que receberá o resultado da multiplicação.

Os endereços permitidos são: bytes e *words*.

### 5.13.4 Divisor (÷)

Quando essa instrução é habilitada por meio da entrada (E), executa-se a divisão dos operandos ( $OPR1 \div OPR2$ ), colocando o resultado em RES, conforme demonstra a figura 5.66.



**Figura 5.66**  
Símbolo gráfico do divisor:



Quando a entrada está habilitada, tem-se a execução da divisão dos dois operandos e, conseqüentemente, o valor passa para um terceiro operando.

**Operando 1** e **operando 2** são os valores que serão divididos.

Os endereços permitidos são: bytes, *words* e constantes.

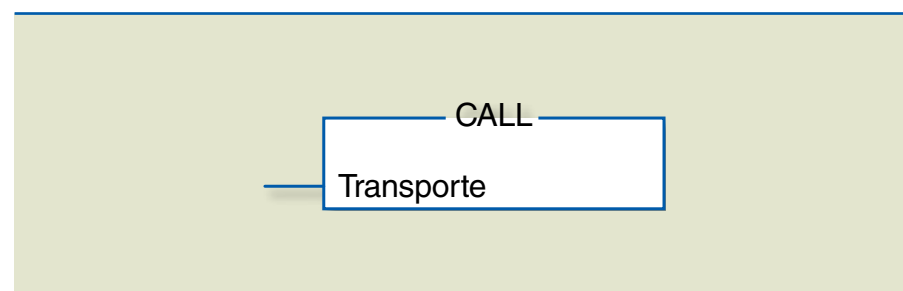
**Resultado** é o operando que receberá o resultado da divisão.

Os endereços permitidos são: bytes e *words*.

## 5.14 Funções especiais

### 5.14.1 CALL

A função especial CALL é mostrada na figura 5.67.

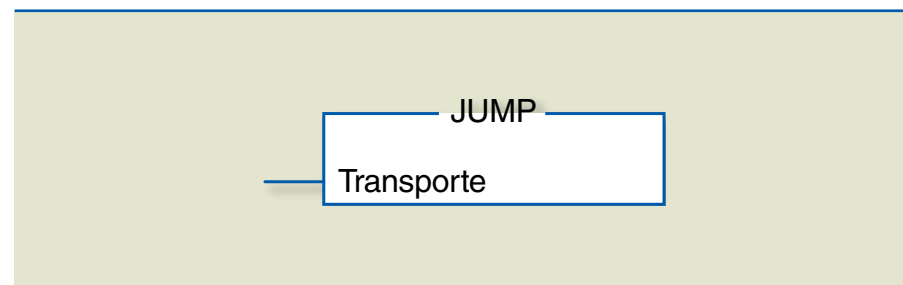


**Figura 5.67**  
Símbolo gráfico da  
função especial CALL.

Quando essa instrução é habilitada, o programa executa a sub-rotina indicada em CALL e, após a execução, retorna para o mesmo ponto do programa que chamou a sub-rotina.

### 5.14.2 JUMP

A função especial JUMP é mostrada na figura 5.68.

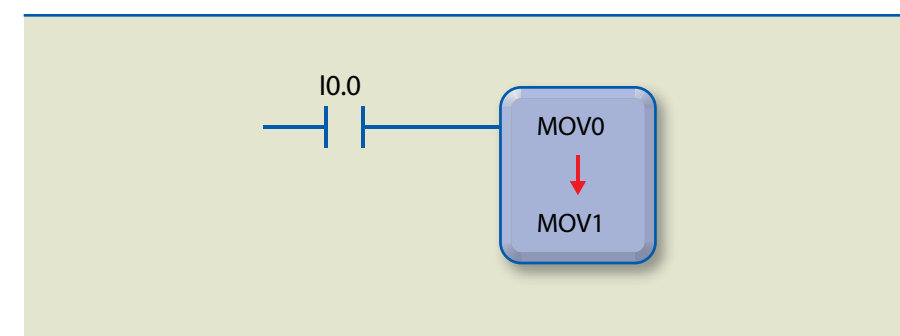


**Figura 5.68**  
Símbolo gráfico da  
função especial JUMP.

Quando essa instrução é habilitada, o programa, ao passar pela instrução, pula para a sub-rotina indicada em JUMP. Nessa instrução, não há retorno para a rotina que estava sendo executada; o programa continua na sub-rotina indicada.

### 5.14.3 MOVE

A figura 5.69 mostra a função especial MOVE.



**Figura 5.69**  
Símbolo gráfico da  
função especial MOVE.

Essa instrução possui duas variáveis: MOV0 (origem) e MOV1 (destino). Quando a entrada I0.0 passa do nível lógico "0" para o "1", a instrução MOVE é habilitada, transferindo o valor contido na variável MOV0 (origem) para a variável MOV1 (destino).



# Capítulo 6

## Projeto de automação utilizando o CLP

## 6.1 Definição dos pontos de entrada e saída

A primeira e talvez a mais importante etapa no processo de automação de um sistema é o levantamento dos pontos de monitoramento referentes à interligação entre o CLP, o equipamento, a máquina ou a planta a ser automatizada. Após a definição desses pontos de entrada e saída, devemos determinar o número de sensores e atuadores do projeto, para análise de sua viabilidade.

Uma vez de posse dessas informações e levando em conta que projetos novos sempre podem estar sujeitos a ajustes de último momento e a solicitações de contingência do cliente, costuma-se manter reserva de entradas e saídas além daquelas do projeto original. Tal adequação também possibilita compatibilizar diferentes modelos de cartões de CLPs comercialmente existentes. Por exemplo, em uma aplicação na qual são necessárias 30 entradas digitais, é comum utilizar cartões que tenham 32 entradas, garantindo duas entradas de reserva para o projeto.

A informação de quais sensores e quantas interfaces existem disponíveis nos módulos de CLP a serem utilizados servirá também como documentação prévia formatada ao programador. O programador deverá vincular cada sensor ou atuador utilizado a um endereço de entrada ou saída do CLP, definindo o endereço para a leitura e escrita de sensores e atuadores que vão compor o projeto.

A tabela 6.1 exemplifica um mapeamento de dispositivos de entrada e saída já definidos de acordo com a programação desenvolvida no projeto.

**Tabela 6.1**

Mapa de entradas e saídas

Endereço do CLP	Símbolo	Descrição
I0.0	B1	Chave liga – Tipo NA
I0.1	B2	Chave desliga – Tipo NF
I0.2	BE	Chave emergência – Tipo NF
I0.3	S7	Contato relé térmico – NF
Q0.2	M1	Contator – Motor M1

## 6.2 Descritivo de funcionamento

Com o mapa de variáveis de entrada e saída definido, o próximo passo é entender o funcionamento desejado do processo. O processo pode ser:

- **Descritivo** – O projeto é descrito função por função de forma textual para que todas as possibilidades de funcionamento da automação estejam previstas no documento.
- **Gráfico** – O projeto é descrito por fluxos de operação (fluxograma), definindo a sequência lógica na qual o programa deve ser executado. Em geral, é apresentado quando se sabe o objetivo a ser alcançado, mas não exatamente quais as formas de atuação e aquisição de informação da planta.

- **Diagrama elétrico de comandos** – Costuma ser utilizado em situações de *retrofitting*, ou seja, quando se tem uma máquina com uma automação baseada em comandos de relés e se deseja atualizar a automação com CLP. Algumas vezes, esquemas elétricos ainda existentes da máquina a ser atualizada possuem documentação em diagrama elétrico. Cabe à pessoa que assume o projeto executar a conversão do projeto em lógicas de programa.
- **Montado pelo projetista** – É usado quando não existe documentação da operação ou a documentação encontra-se incompleta. Nesse caso, é importante observar o funcionamento do equipamento, bem como conversar com os operadores, com o objetivo de entender não só o funcionamento em regime normal, mas também os possíveis erros de operação existentes e como corrigir essas falhas.

Em todos os procedimentos citados, deve-se elaborar uma documentação detalhada do processo antes de iniciar o trabalho de programação. Vale ressaltar que qualquer situação prevista durante a confecção do programa que, no entanto, não conste da documentação recebida deve ser esclarecida de forma oficial, ou seja, por meio de documento firmado entre as partes, para a decisão da situação. Com base nesse acordo, pode-se concluir a tarefa em aberto ou o projeto em questão.

## 6.3 Elaboração de programas

Diversas técnicas podem ser utilizadas para apresentar uma sequência lógica aos programadores que iniciam seus trabalhos com CLP. Esta seção apresenta duas formas para facilitar o entendimento do sequenciamento a ser seguido na elaboração de um programa.

### 6.3.1 Ligar, manter ligado e desligar

Trata-se de uma forma de elaborar programas utilizando selos lógicos em cada linha do programa. Esse procedimento facilita o trabalho dos programadores sem muita experiência em programação de CLPs, por ser de fácil entendimento e possibilitar a construção de lógicas simples.

De início, vamos levar em conta as seguintes variáveis:

- Variável A: responsável por ligar determinada saída digital.
- Variável B: responsável por desligar determinada saída digital.
- Variável C: saída digital ligada por A e desligada por B.

Completando o descritivo, consideremos que as variáveis A e B são energizadas quando são pressionadas as botoeiras A e B, respectivamente.

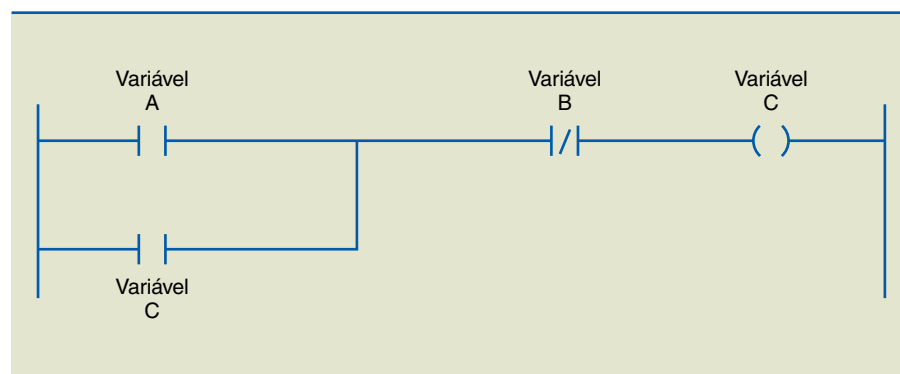
A leitura do estado da variável C é justamente o que define se essa variável deve manter-se ativa ou não.

Apresentam-se, a seguir, a linha de programação (figura 6.1) que executa essa tarefa e sua análise.



**Figura 6.1**

Linha de programação básica.



Quando a botoeira A é acionada, considerando que a botoeira B encontra-se em repouso, a variável A torna-se verdadeira, garantindo a continuidade lógica para o acionamento da variável C. Dado que a variável C agora é verdadeira, sua leitura em paralelo com a variável A ainda garante a continuidade lógica na variável C. Essa situação agora se mantém independente do *status* da variável A. No entanto, basta que a variável B seja falsa, isto é, que a botoeira B seja acionada, para interromper a continuidade lógica na variável C e desligar todo o circuito. Para quem conhece o funcionamento de um selo lógico, não existe muita novidade nesse descritivo.

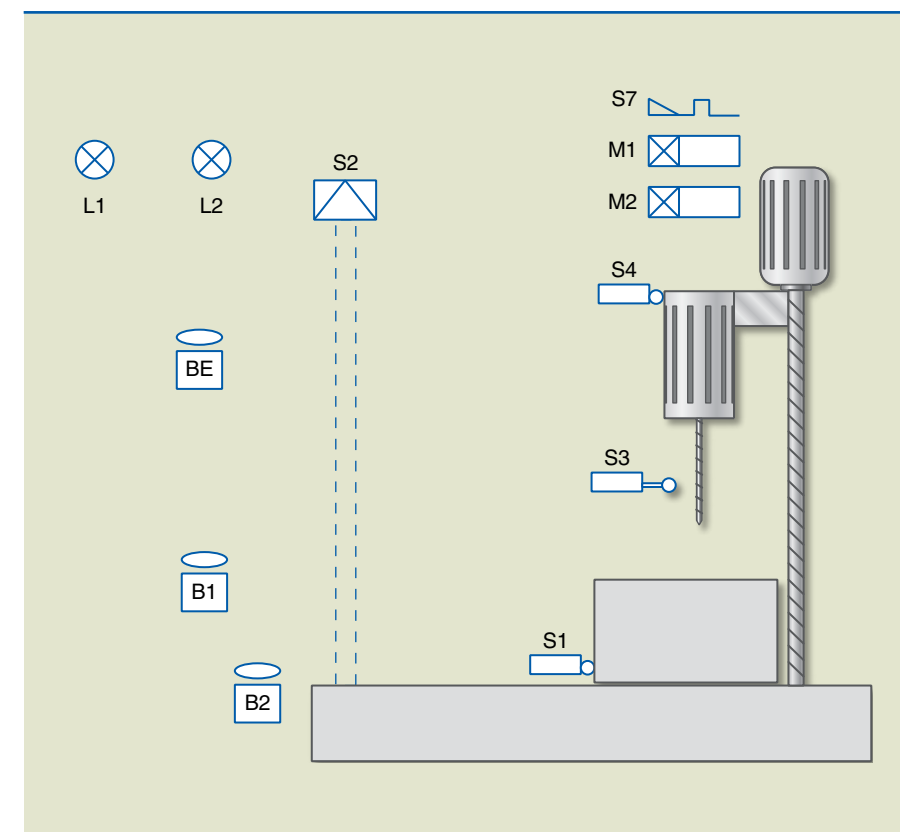
Agora analisemos essas variáveis de maneira mais crítica. Consideremos as condições para que a variável C não seja somente uma botoeira, mas sim um conjunto de operações, sensores ou estados de memória que, quando todos forem verdadeiros, devam executar determinada ação ou determinado acionamento. Um exemplo clássico é o acionamento de sistemas mecânicos como prensas e furadeiras.

Apresenta-se, a seguir, o descritivo de funcionamento de um projeto desse tipo de forma didática, não levando em conta detalhes mais aprofundados que devem ser observados em sistemas de segurança de máquinas operatrizes com operação humana.

O projeto deve obedecer às seguintes condições:

Para que a furadeira desça e execute sua função de furar a peça, ambas as mãos do operador devem estar acionando as botoeiras B1 e B2. O sensor de peça tem de ser acionado, indicando que existe uma peça a ser furada. A área de segurança protegida por uma cortina de luz, sensor S2, não pode ser invadida. O sensor de início de curso S4 deve estar acionado, indicando que a furadeira se encontra na posição inicial do processo. A botoeira de emergência e o sensor S7 de sobrecarga do motor não podem estar acionados. Nessas condições, a lâmpada L1 deve acender e o contator de acionamento de descida do motor deve ser acionado. Quando o sensor S3 é acionado, o contator que propicia a descida da furadeira deve ser desligado e o contator que propicia a subida do conjunto devr ser acionado. O motor da furadeira tem de ser desligado quando o sensor S4 for acionado novamente. A qualquer momento em que uma das botoeiras do operador for desacionada, S7 apresentar sobrecarga, S1 detectar a

ausência de peça ou a cortina de luz for invadida, o processo de descida ou subida com o motor da furadeira acionado deve ser interrompido. Esse processo somente pode ser reiniciado com as condições iniciais de operação garantidas (subindo quando interrompido na subida e descendo quando interrompido na descida). A figura 6.2 mostra o esquema de um sistema de furação de peças em que tal conceito pode ser estudado.

**Figura 6.2**

Sistema de furação de peças.

Podemos destacar como dispositivos de entrada:

- B1 – Botoeira esquerda de acionamento.
- B2 – Botoeira direita de acionamento.
- S1 – Sensor de peça.
- S2 – Cortina de segurança.
- S3 – Sensor de fim de curso.
- S4 – Sensor de início de curso.
- BE – Botão de emergência.
- M1 – Contator de avanço da furadeira.
- M2 – Contator de recuo da furadeira.
- S7 – Sensor de sobrecarga.
- L1 – Lâmpada de indicação ligada.
- L2 – Lâmpada de indicação desligada.

Apresentadas as variáveis de entrada e saída, o mapa de variáveis pode obedecer à descrição mostrada na figura 6.3.



**Figura 6.3**

Mapa de variáveis de entrada e saída.

Endereço	Símbolo	Comentário
I0.0	BE	Botão Emergencia
I0.1	B1	Botão Esquerdo de Acionamento
I0.2	B2	Botão Direito de Acionamento
I0.3	SENSOR 1	Sensore de Peça
I0.4	SENS FIM	Sensor de Fim de Curso
I0.5	SENS INIC	Sensor de Inicio de Curso
Q0.0	MOT AVANÇ	Contator de Avanço do Motor
Q0.1	MOT RECUO	Contator de Recuo do Motor
I0.6	SENS SOBR	Sensor de Sobrecarga
Q0.2	LAMP 1	Lampada Sistema Ligado
Q0.3	LAMP 2	Lampada Sistema Desligado
I0.7	SENS CORT	Sensor da Cortina de Luz

Uma vez configurado o mapa de entradas e saídas, pode-se iniciar a elaboração do programa. Vale ressaltar que existem várias soluções para o mesmo problema, e cada programador pode encontrar outras formas de elaboração. Seguindo a ideia do “ligar, manter ligado e desligar”, as linhas de programa podem ser descritas como apresentado a seguir.

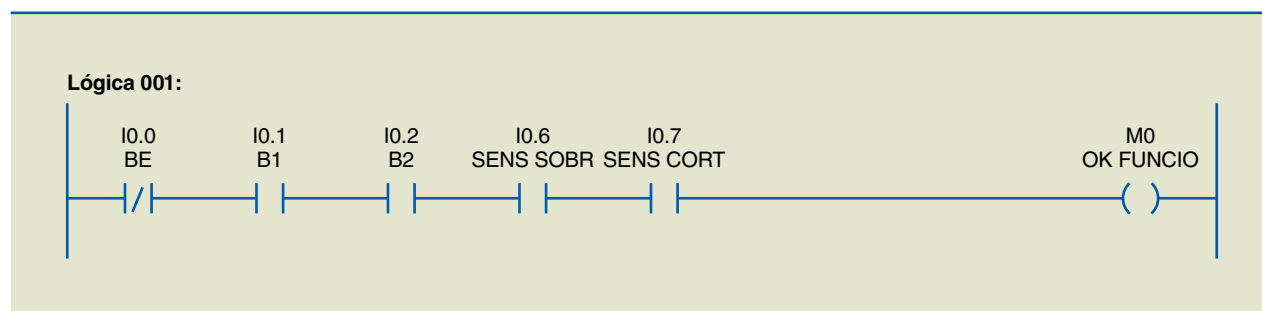
A primeira lógica desenvolvida (figura 6.4) tem a função de energizar a memória zero ( $M_0$ ) quando todas as condições iniciais de funcionamento da ferramenta são verdadeiras, ou seja:

- Não existe um botão de emergência pressionado (BE).
- O operador está pressionando B1 e B2 simultaneamente.
- O sensor de sobrecarga não abriu, ou seja, não está em sobrecarga (SENS SOBR).
- A área do sensor de cortina de segurança S2 não foi invadida (SENS CORT).

Temos a seguinte leitura dessa lógica: todas as variáveis lidas para garantir a continuidade lógica do processo são essenciais, ou seja, a situação necessária para desligar a memória ( $M_0$ ) é a própria ausência de qualquer um dos sinais lidos. As instruções apresentadas são necessárias para manter energizada a variável, não sendo preciso selar essa saída nem programar lógica específica para desativar o selo.

**Figura 6.4**

Primeira lógica: energizar  $M_0$ .

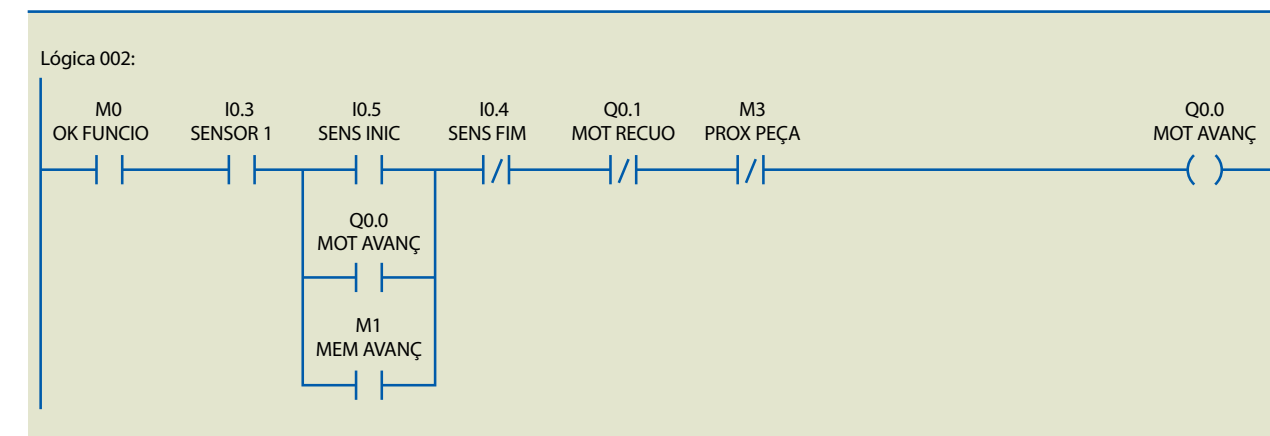


A lógica 2 (figura 6.5) trata do procedimento de descida do conjunto furadeira. Garantidas as condições iniciais pela memória ( $M_0$ ), existindo a peça a ser furada, detectada pelo sensor 1 (I0.3), o sensor de fim de curso (I0.4) não está acionado, pois não haverá movimento vertical do conjunto, uma vez que ele já se encontra na posição final e o contator ligado ao recuo (Q0.1 – MOT RECUO) não está acionado. A memória ( $M_3$ ), que indica próxima peça, não pode estar habilitada (essa variável será explicada na lógica 6). Garantidas essas condições, estando o mecanismo em sua posição inicial, indicada pelo sensor início (I0.5), a saída (Q0.0) será acionada, energizando o contator que propicia o avanço do conjunto e, em paralelo, o acionamento da furadeira.

Toda essa análise garante o acionamento dos motores de descida e da furadeira, mas não que eles permanecerão ligados. Dado o início da descida do conjunto, o sensor de início já não estará ativo, pois o conjunto não estará mais na posição inicial. Dessa maneira, é necessário criar uma lógica que faça com que, uma vez iniciado o processo de descida e furação da peça, ele continue nesse sentido até que o sensor de fim de curso seja acionado. O selo lógico, em paralelo com esse sensor feito com a própria saída (Q0.0), é que garante a continuidade do processo. Existe ainda uma terceira instrução, a memória ( $M_1$ ), comentada na lógica 4, que assegura o procedimento de descida em caso de interrupção momentânea do processo. O sensor de fim de curso (I0.4) exerce a função do “desligar” no esquema “ligar, manter ligado e desligar”.

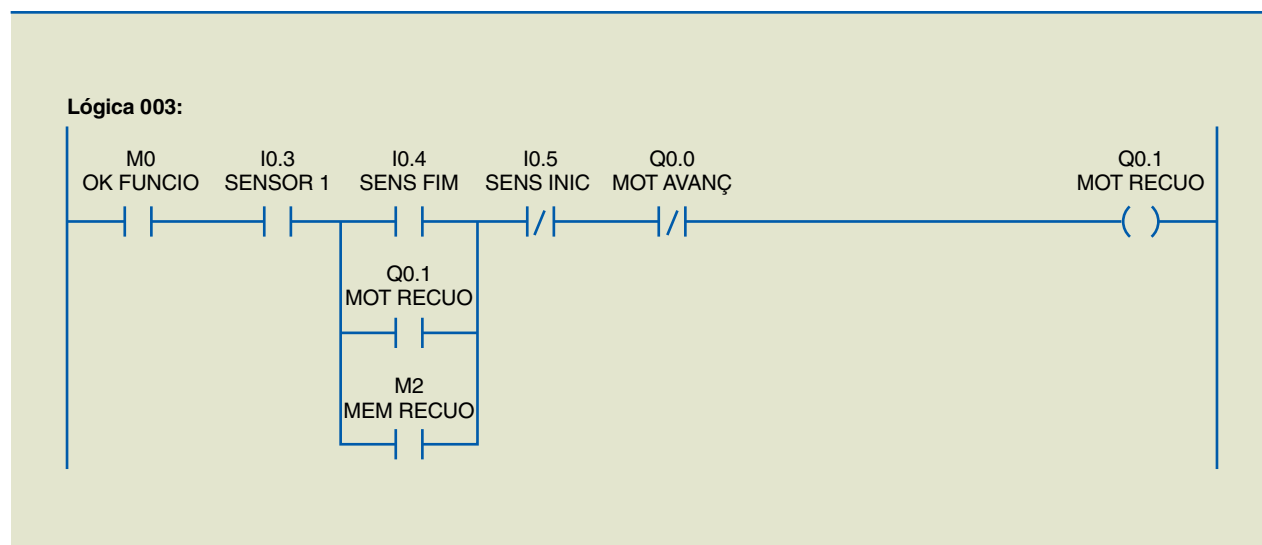
**Figura 6.5**

Procedimento de descida do conjunto furadeira.



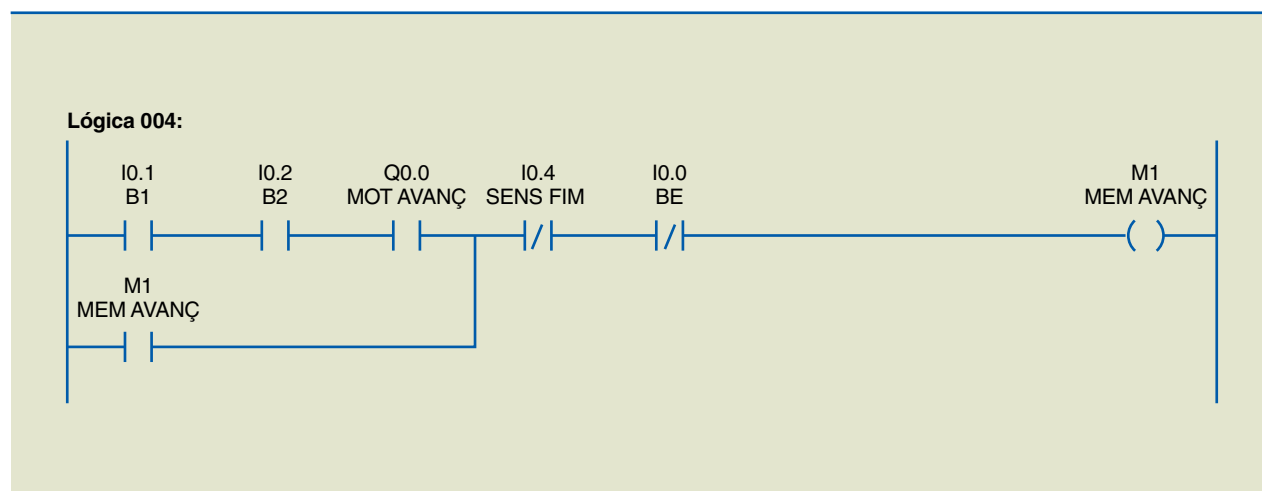
A lógica 3 (figura 6.6) apresenta uma solução para o procedimento de subida do conjunto. Ainda garantidas as condições iniciais pela memória ( $M_0$ ) e existindo a peça a ser furada, detectada pelo sensor 1 (I0.3), a lógica aguarda que o sensor de fim de curso (I0.4) seja acionado. Quando acionado o sensor, essa instrução garante a interrupção da continuidade lógica na saída (Q0.0) do procedimento de descida do conjunto. Uma vez que a descida esteja desabilitada na lógica 2, a mesma instrução que examina se essa variável está em zero na lógica 3 assegura, com o acionamento do sensor de fim de curso, o acionamento do contator responsável pela subida do conjunto. Fazendo novamente a análise do “ligar, manter ligado e desligar”, todas essas instruções garantem o início do processo de subida, mas não que ele se mantenha assim depois de iniciado, pois o sensor de fim de curso é automaticamente desabilitado quando esse processo se inicia. Por isso, é

**Figura 6.6**  
Procedimento de subida  
do conjunto furadeira.



A lógica 4 (figura 6.7) tem a função específica de memorizar se o conjunto está na situação de descida no instante em que algum evento que interrompa o processo ocorrer. Se as botoeiras B1 e B2 forem acionadas e o contator de descida energizado, a memória de avanço (M1) será energizada. Tal sequência faz parte da lógica “ligar”. Essa memória permanecerá nessa condição graças ao selo lógico feito pelo próprio endereço (M1), exercendo a função do “manter ligado” até que o sensor de fim de curso (I0.4) ou o botão de emergência (I0.0) seja acionado, ambos (sensor de fim de curso e botão de emergência) fazendo parte da lógica “desligar”, da proposta inicial comentada. A memória (M1) garante que o procedimento de descida, uma vez interrompido na lógica 2, continue até que uma nova condição de parada ocorra ou até que o conjunto termine o procedimento, acionando, assim, o sensor de fim de curso.

**Figura 6.7**  
Memória de avanço (M1).

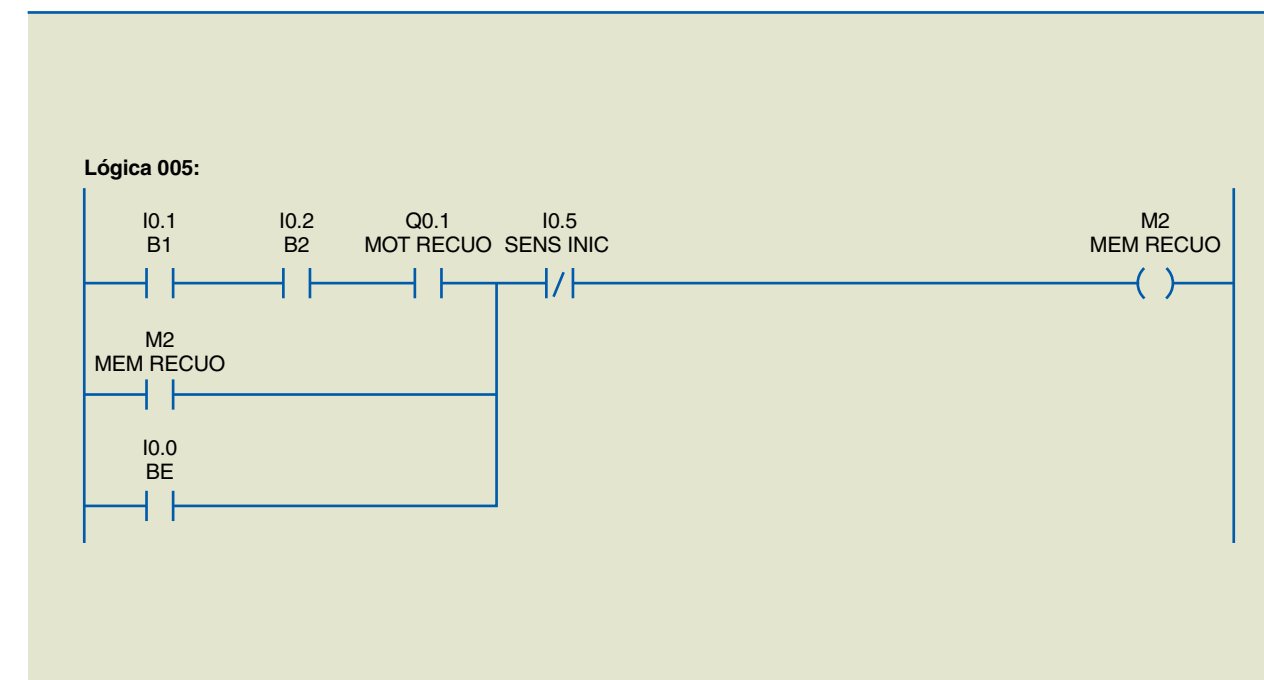


necessário que o selo lógico esteja em paralelo com esse sensor, fazendo com que o processo de subida continue até que a lógica seja interrompida com o acionamento do sensor de início de curso (I0.5). Esse sensor exerce a função do “desligar” no esquema “ligar, manter ligado e desligar”.

A descrição da lógica 5 (figura 6.8) é semelhante à da lógica 4, porém, faz com que o conjunto recue quando as condições de operação estiverem restabelecidas. A principal diferença entre as duas lógicas é que o botão de emergência garante, na 5, que a memória seja acionada e, na 4, que ela seja desligada.

É interessante que a máquina retorne à posição inicial mesmo que o processo ainda não tenha sido completado. Voltando à análise do “ligar, manter ligado e desligar”, as botoeiras pressionadas com o acionamento do motor de recuo ou o acionamento do botão de emergência asseguram que a memória seja energizada. A própria memória de recuo faz com que ela se mantenha energizada, e o acionamento do sensor de início de curso garante o desligamento da memória.

**Figura 6.8**  
Memória de recuo (M<sub>2</sub>).

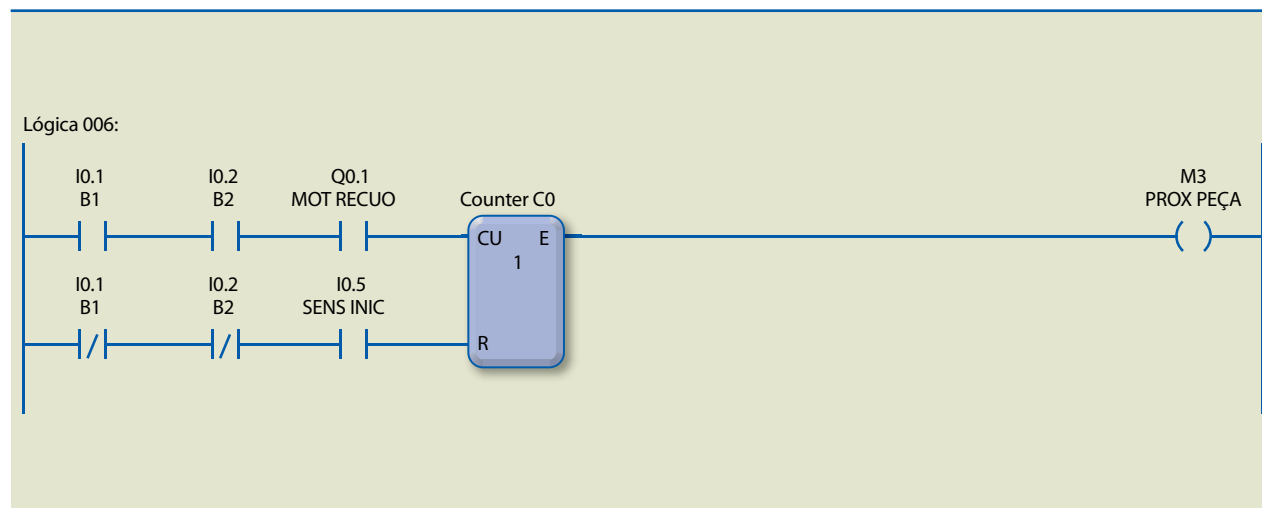


A lógica 6 (figura 6.9) bloqueia o processo de descida, depois que o conjunto desceu e subiu, executando a função de furar a peça. Conforme comentado na lógica 2, o endereço de memória (M<sub>3</sub>) será verdadeiro se os botões B1 e B2 e também o contador de recuo estiverem acionados. Essas três instruções verdadeiras garantem continuidade lógica no contador (C01), predefinido para totalizar um pulso de entrada e acionar sua saída uma vez que a situação esteja atendida.

A saída *enable* do contador será verdadeira enquanto o sensor de início de processo não estiver acionado e as duas botoeiras liberadas, garantindo que o operador utilizará as mãos para substituir a peça que foi furada por outra que ainda não foi. Ou seja, a ausência das mãos do operador nas botoeiras B1 e B2, mais o fato de a máquina estar na posição inicial faz com que o *reset* do contador seja acionado e que a memória (M3) volte ao estado desligado, permitindo um novo ciclo de processo.







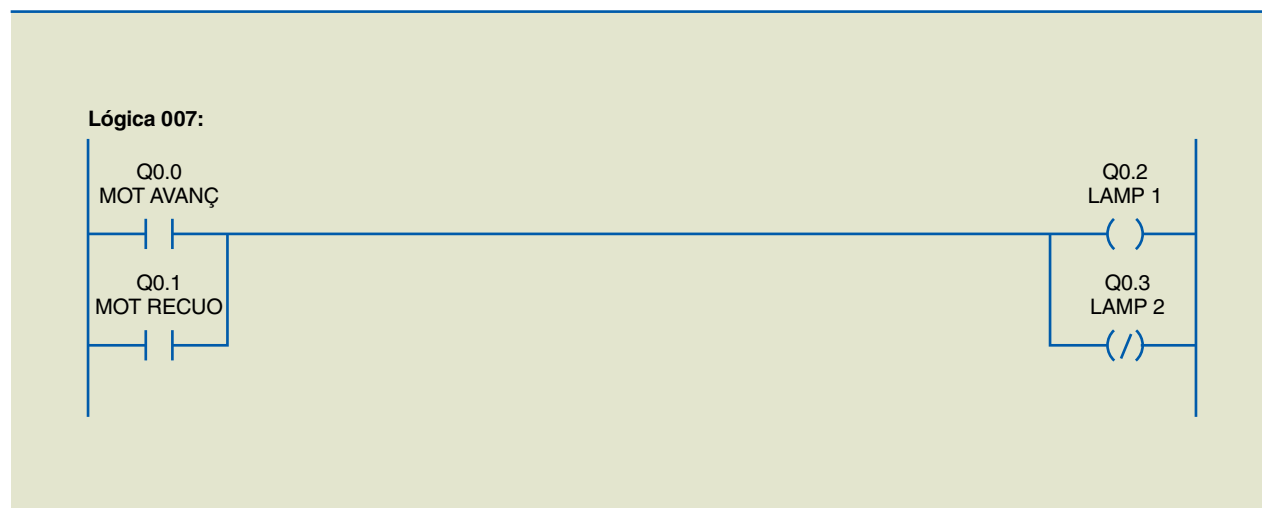
**Figura 6.9**

Bloqueio do processo de descida.

A lógica 7 (figura 6.10) é um simples complemento que garante o acionamento da lâmpada 1 (Q0.2) quando um dos contatores estiver acionado, seja na descida, seja na subida. Com a saída complementar (Q0.3) ligada à lâmpada 2, esta permanecerá desligada enquanto a lâmpada 1 estiver acionada. Se a lâmpada 1 estiver apagada, a lâmpada 2 será acionada, informando que a máquina está parada nesse instante.

**Figura 6.10**

Acionamento das lâmpadas.



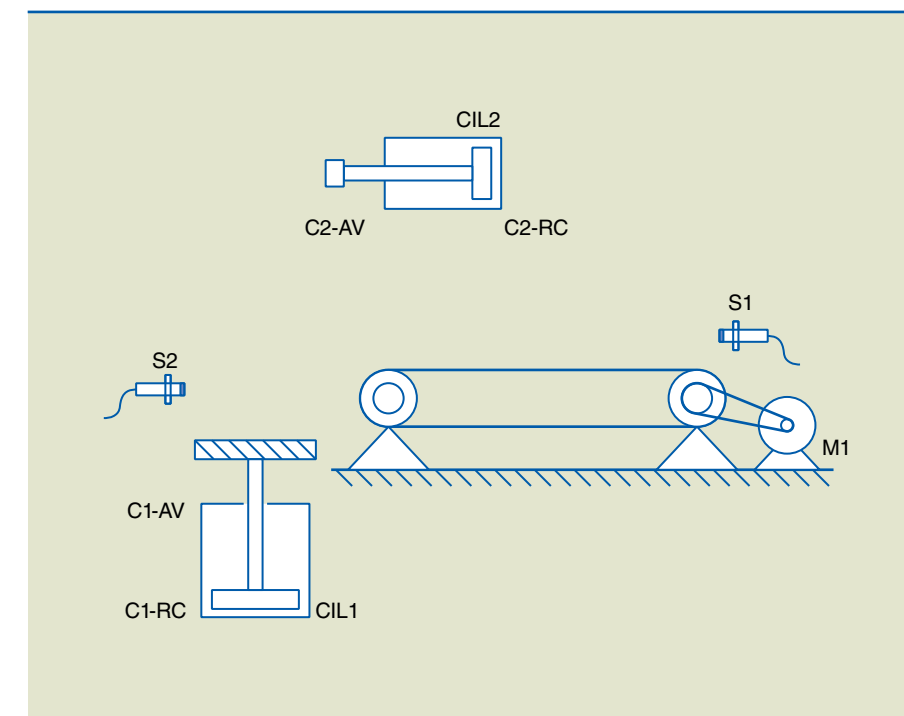
### 6.3.2 Passos e transições

A técnica de programação de passos e transições é muito útil na elaboração de programas de natureza essencialmente sequencial, ou seja, quando os acionamentos dependem do ponto da sequência em que o sistema se encontra. Essa técnica baseia-se no estudo do que deve ocorrer e por quanto tempo, caso o controlador receba alguma informação do processo. É recomendada quando a lógica com intertravamento puro se mostra muito extensa, exigindo a criação e manipulação de diversas memórias auxiliares, o que torna o programa como um todo difícil de entender e, não raras vezes, pouco confiável.

Essa técnica consiste basicamente em definir bits que correspondam aos passos da sequência de funcionamento ou estados do programa. Os passos são níveis lógicos que determinadas entradas ou saídas devem encontrar de modo a habilitar uma nova etapa do processo. Para que esse novo passo seja obtido, é necessário que ocorram transições, até que um novo passo seja alcançado.

Exemplificando de forma prática: a válvula que abastece um reservatório só deve ser desligada quando ele estiver cheio, ou seja, o primeiro passo é o reservatório vazio e a primeira transição é encher o reservatório abrindo a válvula. O segundo passo é o reservatório cheio, que provoca uma segunda transição, que é fechar a válvula.

Vários são os processos que podem ser montados com essa técnica, entre eles projetos de esteira transportadora de caixas com cilindros expulsores. Com o sistema descrito na figura 6.11, pode-se aprofundar a análise dessa técnica de programação.



**Figura 6.11**

Esteira transportadora de caixas com cilindros expulsores.

Sequência de trabalho do sistema:

- Quando uma caixa for detectada pelo sensor S1, o motor M1 deve ser acionado e o transporte da caixa iniciado.
- Quando a caixa se aproximar do sensor S2, o motor deve ser desligado e o avanço do cilindro CIL1 acionado.
- Quando o cilindro CIL1 chegar à sua posição final, o avanço do cilindro CIL2 deve ser acionado.
- Quando o cilindro CIL2 chegar à sua posição final, os cilindros CIL1 e CIL2 devem retornar, habilitando o sistema para um novo ciclo de operação.



Para esse sistema, desenvolveremos a sequência de passos da tabela 6.2.

Passo	Acionamentos realizados	Mudança de passo (transição)	Passo seguinte	Descrição
1 (passo inicial)	Nenhum	Sensor S1 atuado	2	Aguarda caixa ser depositada na cabeceira da esteira.
2	Movimento da esteira – motor M1	Sensor S2 atuado	3	Movimenta a esteira e aguarda a caixa chegar à cabeceira esquerda, mantendo o movimento da esteira.
3	Avanço do cilindro elevador CIL1	Avanço do cilindro CIL1 concluído – sensor C1-AV atuado	4	Para o movimento da esteira e aciona a elevação da caixa.
4	Mantém avanço do cilindro elevador CIL1 Avanço do cilindro expulsor CIL2	Avanço do cilindro CIL1 concluído – sensor C1-AV atuado Avanço do cilindro CIL2 concluído – sensor C2-AV atuado	5	Mantém a elevação da caixa e aciona o cilindro de expulsão.
5	Recuo dos cilindros de elevação e expulsão – recuo dos cilindros CIL1 e CIL2	Recuo do cilindro CIL1 concluído – sensor C1-RC atuado Recuo do cilindro CIL2 concluído – sensor C2-RC atuado	1	Recua cilindros pneumáticos e retorna ao passo inicial.

**Tabela 6.2**

Sequência de passos

Pela sequência, é possível perceber que cada passo do programa corresponde a um ponto em que dada ação deve ser tomada, seja a execução ou a interrupção de uma instrução, seja a espera de determinada condição. Dessa maneira, durante todo o ciclo de funcionamento do programa, apenas um estado da sequência permanece ativo por vez, garantindo o comportamento preciso e confiável do sistema.

Podemos utilizar a definição de que cada passo do programa corresponde a um bit. Os passos são executados de forma sequencial; o primeiro deles deve ser o primeiro da sequência lógica definida no programa e pré-requisito de análise para os subsequentes. Nessa técnica, é comum a padronização com o uso das instruções *set* e *reset*, vistas no capítulo 5, ou seja, em cada transição, devemos ressetar o passo dado e setar o seguinte.

Como exemplo, implementaremos o programa da esteira transportadora de caixas. O objetivo é descrever o programa em Ladder do sistema, utilizando a técnica de passos e transições para o desenvolvimento da sequência de operações.

Cada passo do programa corresponderá a uma memória do CLP. Portanto, o passo inicial para programar é a alocação das memórias que corresponderão aos passos. No exemplo apresentado, há apenas cinco passos, logo serão necessárias

apenas cinco memórias. Recomenda-se reservar maior quantidade de memórias que os passos do programa, mesmo que muitas delas não sejam utilizadas, pois servirão para futuras expansões no sistema. Reserva-se, assim, os primeiros 16 bits da memória para os passos do sistema. A tabela 6.3 mostra a alocação dos passos e das memórias.

Passo	Símbolo	Memória
1	PAS1	M0
2	PAS2	M1
3	PAS3	M2
4	PAS4	M3
5	PAS5	M4
...	...	...
14	PAS14	M13
15	PAS15	M14
16	PAS16	M15

**Tabela 6.3**

Alocação dos passos e das memórias do sistema

Uma vez definidos os sensores e atuadores do sistema, tal informação também deve estar definida no mapa de entradas e saídas, conforme mostra a tabela 6.4.

**Tabela 6.4**

Mapa de entradas e saídas (I/O)

Sensores	Descrição	Memória
S1	Sensor de caixa 1 (caixa presente = nível lógico 1)	I0.0
S2	Sensor de caixa 2 (caixa presente = nível lógico 1)	I0.1
C1-AV	Sensor cilindro 1 (posição máxima de avanço = nível lógico 1)	I0.2
C2-AV	Sensor cilindro 2 (posição máxima de avanço = nível lógico 1)	I0.3
C1-RC	Sensor cilindro 1 (posição máxima de recuo = nível lógico 1)	I0.4
C2-RC	Sensor cilindro 2 (posição máxima de recuo = nível lógico 1)	I0.5
Atuadores	Descrição	Memória
M1	Contator de acionamento do motor 1 da esteira (liga motor em 1)	Q0.0
CIL1	Válvula solenoide para o cilindro 1 (avanço em 1, recuo em 0)	Q0.1
CIL2	Válvula solenoide para o cilindro 2 (avanço em 1, recuo em 0)	Q0.2

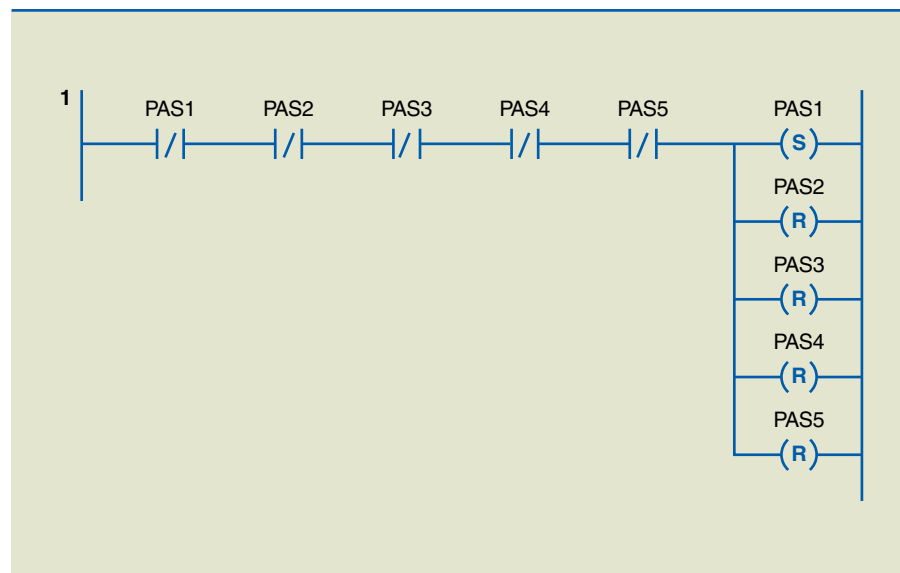


Declaradas as variáveis e as memórias vinculadas aos passos do programa, pode-se começar a escrevê-lo.

A primeira linha (figura 6.12) é utilizada para fazer o programa assumir o passo inicial quando nenhum outro passo estiver selecionado. Portanto, quando nenhum passo estiver selecionado, aciona-se o passo 1.

**Figura 6.12**

Programa para assumir o passo inicial.



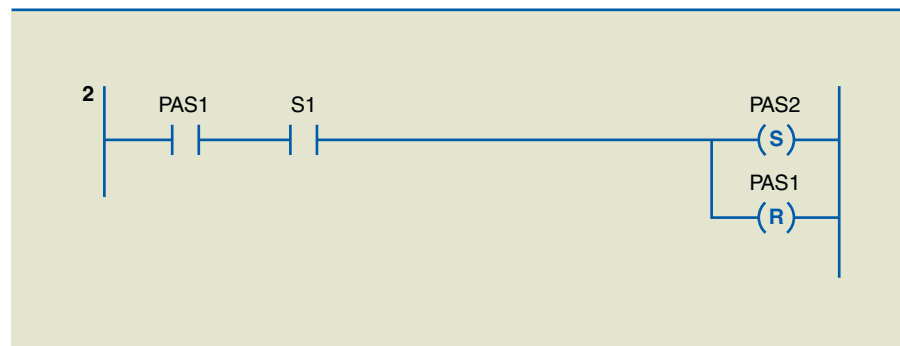
Geralmente, os sistemas possuem um botão de *reset* geral, que retorna o sistema à posição inicial. Esse comando de *reset* geral costuma ser colocado na linha acima, como condição para retorno dos passos ao ponto inicial, e em paralelo com o conjunto de contatos NF em série (PAS1 a PAS5).

Após essa linha de instruções, será iniciada a construção dos passos do programa propriamente ditos. Para simplificar o entendimento, vamos escrever as linhas dos passos na ordem: passo 1, passo 2, passo 3 etc.

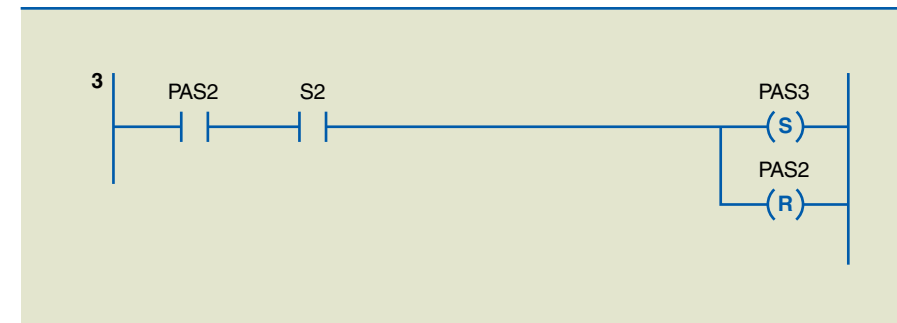
O passo 1 (figura 6.13) apresenta a linha de transição do passo 1 para o próximo passo – no caso, passo 2. Tal transição é feita quando o sensor S1 está atuado. Portanto, o passo 2 é acionado quando o programa está no passo 1 e o sensor S1 ativo. O passo 2 é acionado e o passo 1 ressetado.

**Figura 6.13**

Transição do passo 1 para o passo 2.



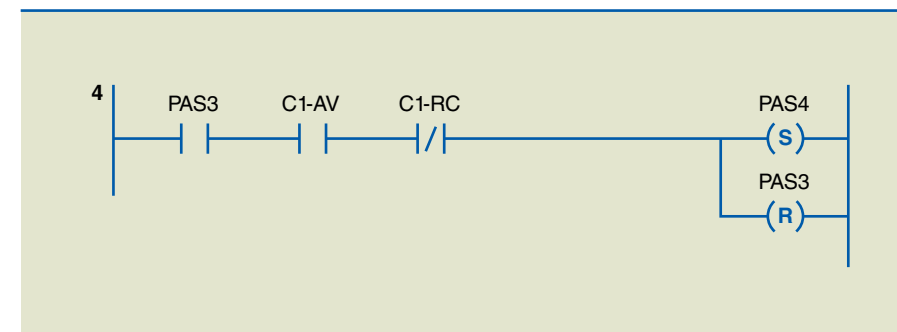
A figura 6.14 ilustra a linha que representa a transição do passo 2 para o passo 3. Essa transição é iniciada quando, estando no passo 2, se detecta o acionamento do sensor S2. Lembre-se de que essa transição representa o abandono do passo 2 e o acionamento do passo 3.



**Figura 6.14**

Transição do passo 2 para o passo 3.

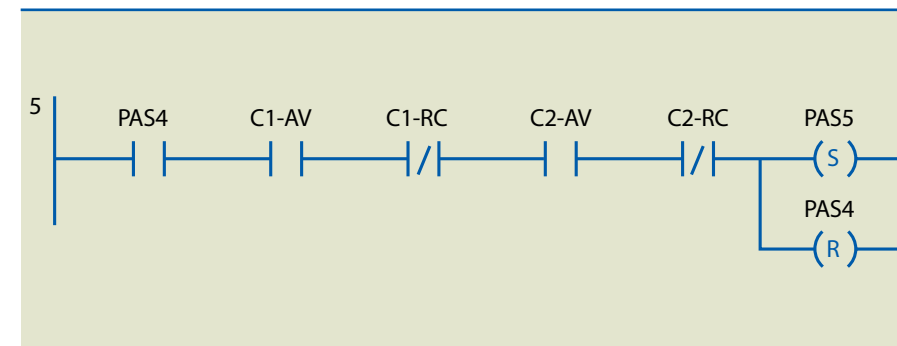
Seguindo com a lógica, desenvolveremos a linha que representa a transição do passo 3 para o passo 4 (figura 6.15). Essa transição ocorre com a conclusão do movimento de avanço do cilindro CIL1 no passo 3, ou seja, quando o sensor C1-RC está desligado e o sensor C1-AV é acionado durante o passo 3.



**Figura 6.15**

Transição do passo 3 para o passo 4.

A transição do passo 4 para o passo 5 (figura 6.16) é semelhante à última linha, mas envolve o avanço dos cilindros CIL1 e CIL2. Essa transição ocorre quando se tem a confirmação de que CIL1 e CIL2 estão avançados no passo 4. Considerando que o reconhecimento de avanço de determinado cilindro é dado pelo acionamento de seu sensor de avanço mais o desacionamento de seu sensor de recuo, a linha para a transição será representada como na figura 6.16.



**Figura 6.16**

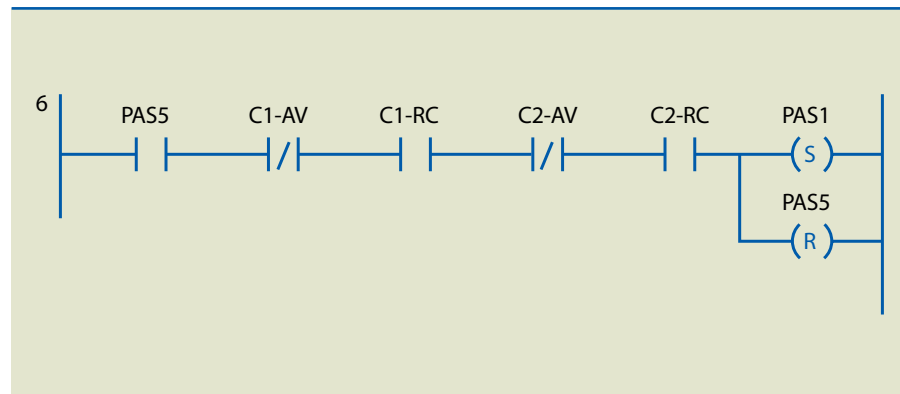
Transição do passo 4 para o passo 5.



Finalizadas as linhas de transição, deve-se realizar a transição do passo 5 de volta ao passo 1 (figura 6.17), a qual ocorre quando CIL1 e CIL2 estão recuados. Portanto, o acionamento do passo 1 pode ser escrito com o reconhecimento de ambos os cilindros recuados durante o passo 5, sendo o reconhecimento de recuo de um cilindro o acionamento de seu sensor de recuo com o desligamento de seu sensor de avanço. A figura 6.17 ilustra a linha para essa transição.

**Figura 6.17**

Transição do passo 5 para o passo 1.

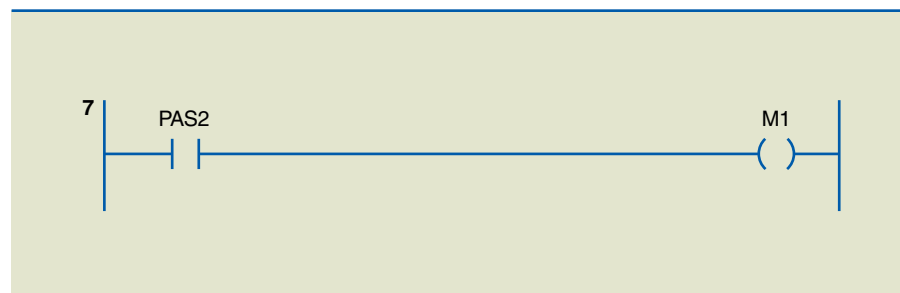


Toda a sequência de transições é concluída com a linha da figura 6.17, restando somente elaborar as linhas de acionamento de cada dispositivo em seu respectivo passo.

O motor M1 é acionado no passo 2, conforme mostra a figura 6.18.

**Figura 6.18**

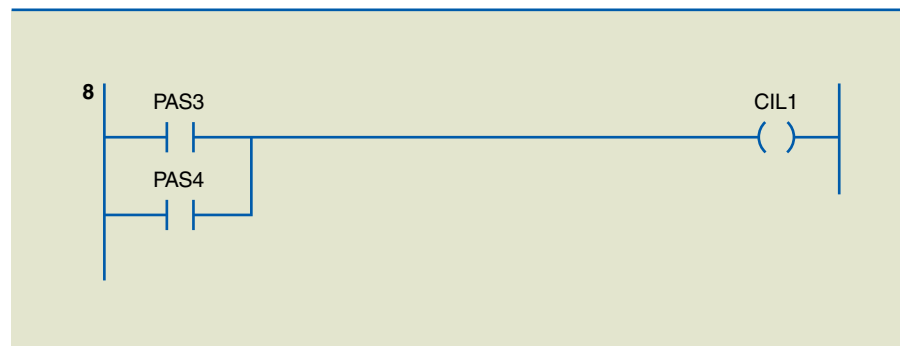
Acionamento do motor M1.



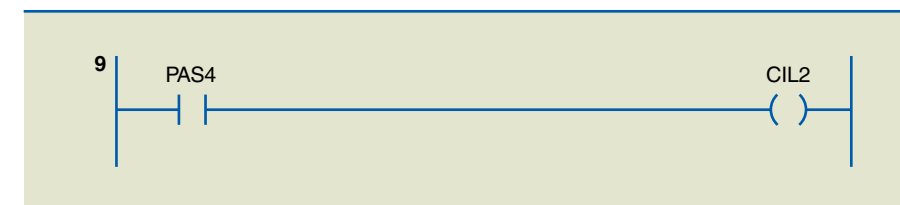
O cilindro CIL1 é acionado nos passos 3 e 4, como ilustra a figura 6.19.

**Figura 6.19**

Acionamento do cilindro CIL1.



O cilindro CIL2 é acionado no passo 4, tendo como última linha a da figura 6.20.



**Figura 6.20**

Acionamento do cilindro CIL2.

## 6.4 Testes, simulações e alterações

É grande o número de *softwares* de programação de CLPs que permitem a simulação dos programas elaborados, porém, são genéricos e limitados. No procedimento de teste e simulação de um programa, devem ser consideradas com rigor as características dos dispositivos de entrada que serão utilizados no projeto físico. Uma das causas de erros de programação é não levar em conta a variedade de tipos de dispositivos de entrada, como chaves e botões pulsantes, chaves e botões que possuem contatos normalmente fechados, chaves de duas posições (que se mantêm na posição) etc.

Os *softwares* de simulação normalmente disponibilizam chaves biestáveis para o acionamento de entradas digitais. Essas chaves são de operação retentiva e podem confundir os que se iniciam nessa tarefa de programação Ladder. Considerando que as chaves disponibilizadas pelos *softwares* são retentivas, a simulação de uma botoeira não retentiva, por exemplo, nada mais é do que o acionamento e o desacionamento da chave disponibilizada pelo *software*, ou seja, a geração de um pulso na entrada digital em teste.

Cabe alertar que ocorrem erros de simulação quando essa chave é acionada e esquecida pelo programador, simulando que o operador da máquina está o tempo todo com a botoeira pressionada. Essa situação pode camuflar possíveis erros de lógica, como a necessidade de inserção de selos lógicos na programação.

Outro erro comum é usar instruções invertidas quando se precisa elaborar um programa com chaves ou sensores NF. Para garantir a continuidade lógica de uma linha de programa quando o dispositivo NF está em situação normal de operação, deve-se utilizar a instrução NA (—|—), pois é a chave que garante energia à entrada digital e, por consequência, torna a instrução NA verdadeira na lógica construída.

Simular um dispositivo NF em um *software* de simulação é iniciar o programa com a chave fechada. Quando pressionada, a chave NF se abre, interrompendo a continuidade lógica na linha de programação que utiliza uma instrução NA monitorando tal variável.

Simulações devem ser feitas com cautela e dificilmente substituem um teste final na liberação do projeto.



### 6.5 Liberação de projeto

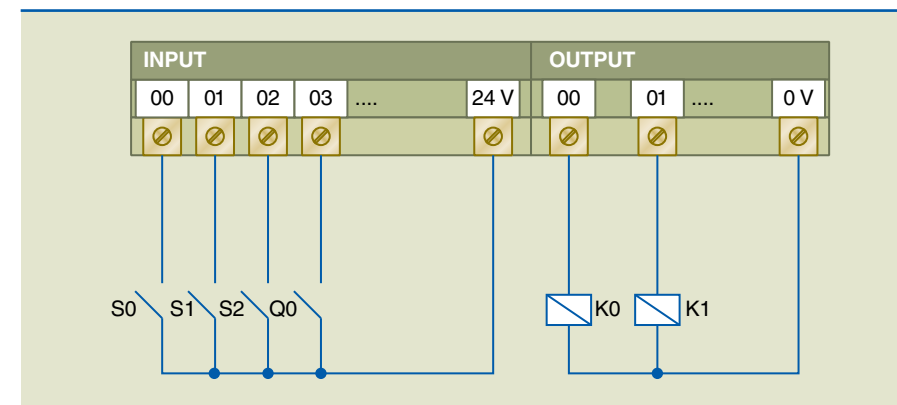
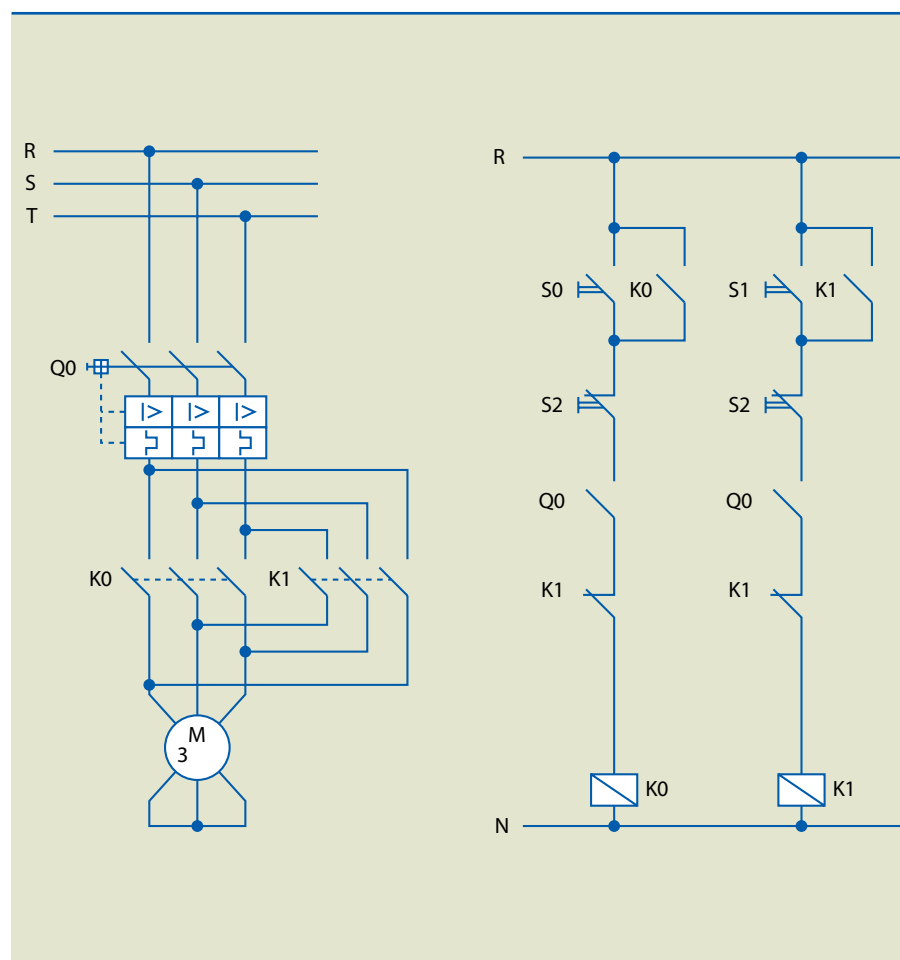
O sucesso na liberação de um projeto está diretamente atrelado à qualidade da produção do projeto, no que se refere à identificação dos componentes, à sua descrição técnica, aos diagramas unifilares indicando claramente em que porta do CLP cada sensor ou atuador vai ser ligado etc. Todo o processo de teste, incluindo o monitoramento das variáveis, a energização de contatores e o reconhecimento de sensores, deve ser feito antes de o circuito de potência ser energizado. Tal processo pode durar dias, semanas ou meses, mas é de vital importância para garantir que todas as possíveis falhas sejam investigadas e, se existirem, corrigidas. Essa etapa garante a integridade física dos operadores e do processo durante os testes e durante a operação do sistema.

### 6.6 Projetos para estudo

#### 6.6.1 Motor trifásico com reversão

Será elaborado um programa para o CLP que implemente a partida direta de um motor trifásico com reversão. O esquema elétrico é apresentado na figura 6.21, e a ligação no CLP, na figura 6.22. O mapeamento das entradas e saídas utilizado pode ser observado na figura 6.23.

**Figura 6.21**  
Esquema elétrico de partida de motor trifásico com reversão.



**Figura 6.22**  
Esquema elétrico de ligação no CLP do projeto de motor trifásico com reversão.

Tabela de Símbolos

Endereço	Símbolo	Comentário
I0.0	LIGA S0	
I0.1	LIGA S1	
I0.2	DESL S2	
I0.3	REL Q0	
Q0.0	MOTOR S0	
Q0.1	MOTOR S1	

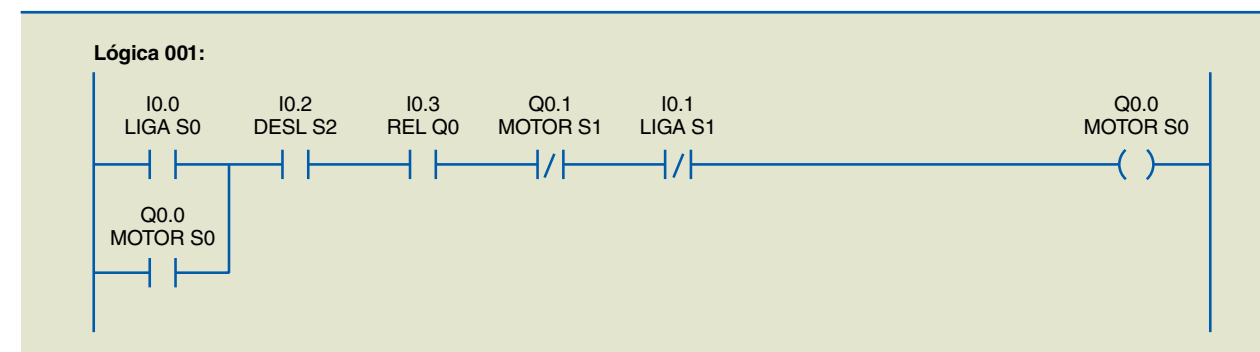
Inserir todos    Apagar linha    Imprimir    X Fechar

**Figura 6.23**  
Mapeamento das entradas e saídas do projeto de motor trifásico com reversão.

A solução proposta é descrita a seguir.

Lógica 1 (figura 6.24) – (I0.0) inicia todo o processo de energizar a saída (Q0.0). Isso ocorrerá desde que: a chave S2 (tipo NF) não esteja pressionada; o relé térmico (I0.3) (NF) não esteja atuado; o botão S1 de reversão não esteja pressionado; e o contator (Q0.1) que aciona o motor no sentido anti-horário não esteja acionado. Uma vez atuada a saída (Q0.0), o selo lógico sobre a chave S0 é mantido até que uma das demais condições citadas não seja mais válida.

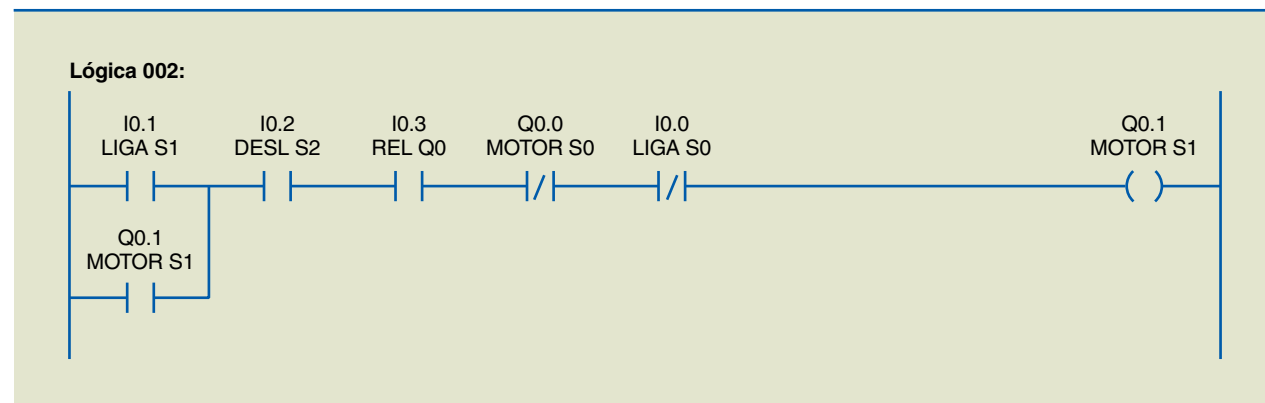
**Figura 6.24**  
Acionamento do motor no sentido horário com S0.



Lógica 2 (figura 6.25) – De maneira análoga à lógica 1, o que define esse sentido de rotação é a chave S1 ligada ao endereço (I0.1). Isso ocorrerá desde que: a chave S2 (tipo NF) não esteja pressionada; o relé térmico (I0.3) não esteja atuado; o botão S0 de reversão não esteja pressionado; e o contator (Q0.0) que aciona o motor no sentido horário não esteja acionado. Uma vez atuada a saída (Q0.1), o selo lógico sobre a chave S1 é mantido até que uma das demais condições citadas não seja mais válida.

**Figura 6.25**

Acionamento do motor no sentido anti-horário com S1.

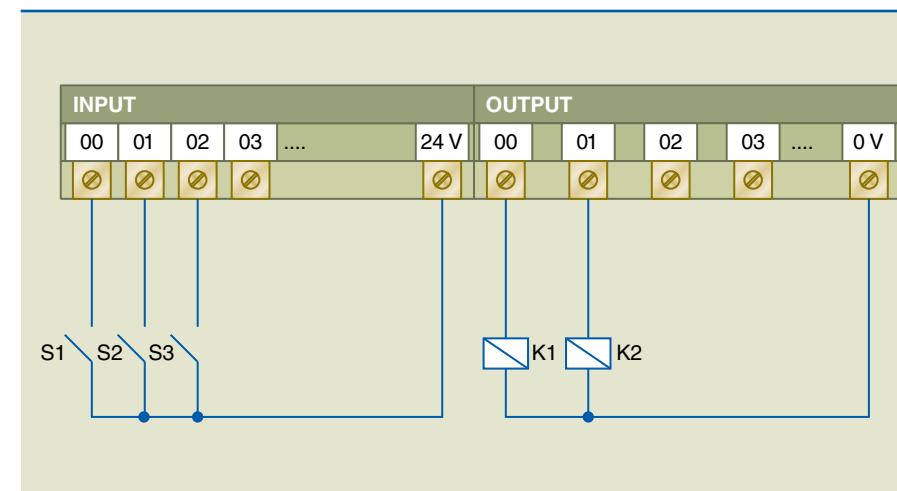
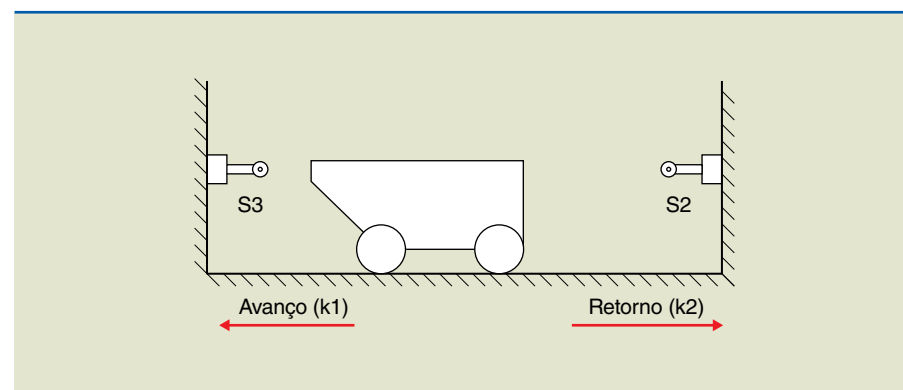


### 6.6.2 Carro transportador

Será desenvolvido o programa do CLP para controle de um carro transportador. O carro transportador deve funcionar da seguinte maneira: o operador pressiona o botão S1 para dar o comando de avanço do carro. Na posição inicial (recuada), o sensor fim de curso S2 permanece atuado. Uma vez pressionado o botão S1, o carro inicia seu movimento de avanço, por meio do contator K1. Ao atingir o fim da trajetória, o sensor fim de curso S3 é acionado, momento no qual o movimento de avanço é interrompido e o movimento de recuo é acionado pelo contator K2, automaticamente. O carro transportador continua seu movimento de retorno até que o sensor fim de curso S2 seja acionado de novo, quando o carro transportador deve parar. Caso o carro transportador encontre-se parado no meio da trajetória, o operador tem de pressionar o botão S1 para que ele recue. A representação do processo é ilustrada na figura 6.26; e a ligação dos dispositivos ao CLP, na figura 6.27. O mapeamento das entradas e saídas utilizado pode ser observado na figura 6.28.

**Figura 6.26**

Representação do processo.



**Figura 6.27**

Esquema elétrico de ligação no CLP do projeto de carro transportador.

Endereço	Símbolo	Comentário
I0.0	BOTAO S1	
I0.1	SENSOR S2	
I0.2	SENSOR S3	
Q0.0	K1 CONT	
Q0.1	K2 CONT	

**Figura 6.28**

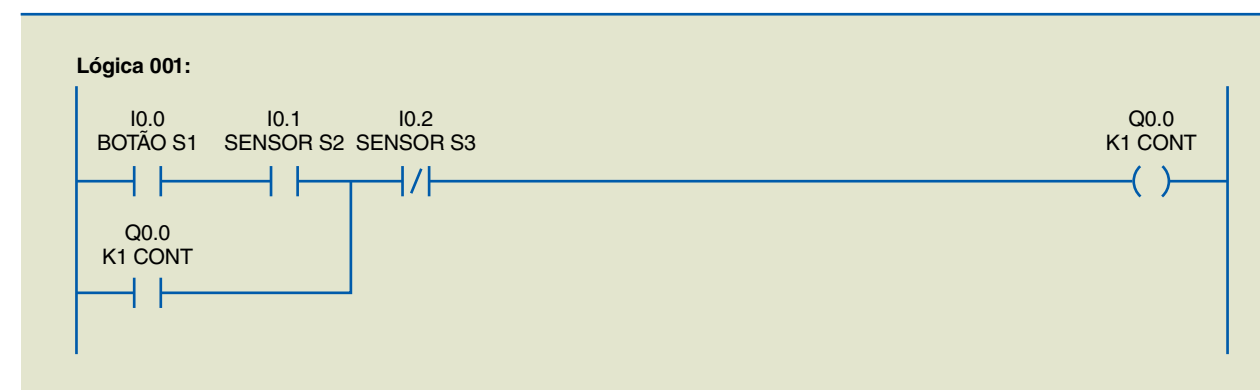
Mapeamento das entradas e saídas do projeto de carro transportador.

A solução proposta é descrita a seguir.

Lógica 1 (figura 6.29) – O acionamento do motor no sentido avanço ocorre quando o botão S1 é pressionado, desde que o sensor S2 esteja acionado, até o momento de acionamento de S3. O selo lógico é garantido por (Q0.0).

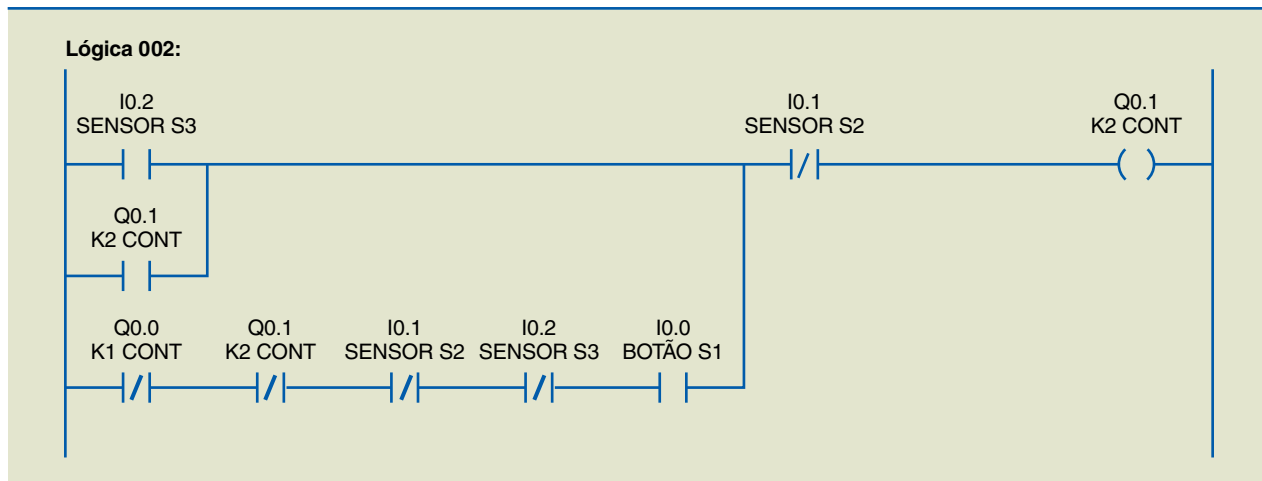
**Figura 6.29**

Lógica 1.



Lógica 2 (figura 6.30) – (Q0.1) o contator responsável pelo retorno do carro é acionado quando, em movimento, o carro aciona o sensor S3. Após interrupção de energia, quando o carro se encontra no meio do caminho, deve-se pressionar S1 para que ele retorne à posição inicial. O selo lógico é mantido pela própria saída (Q0.1), até que o carro atinja a posição inicial.

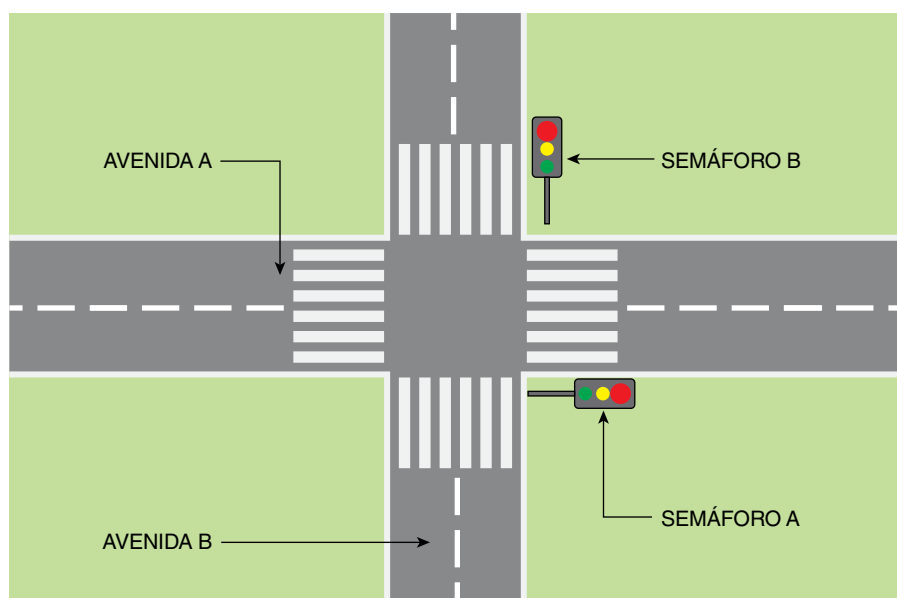
**Figura 6.30**  
Lógica 2.



### 6.6.3 Semáforos

Será desenvolvido um programa no CLP para controle dos semáforos de um cruzamento de duas avenidas (figura 6.31). A mudança dos sinais dos semáforos é feita por temporização. Os semáforos permanecem em cada estado por um período de 5 segundos. Os estados dos sinais são apresentados na tabela 6.5; o esquema de ligação dos dispositivos no CLP, na figura 6.32; e o mapeamento das saídas utilizado, na figura 6.33.

**Figura 6.31**  
Cruzamento e semáforos.



**Tabela 6.5**

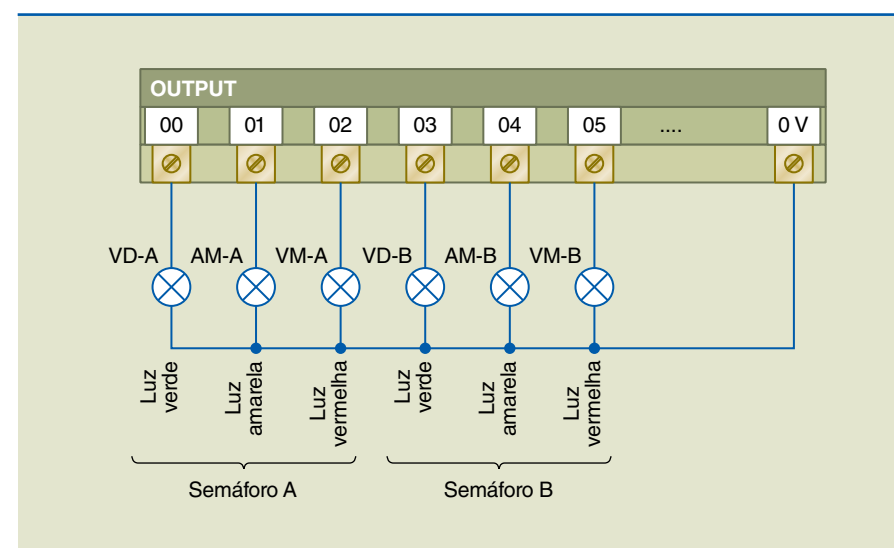
Tabela de estados dos semáforos

Estado	Semáforo avenida A	Semáforo avenida B
1	Verde	Vermelho
2	Amarelo	Vermelho
3	Vermelho	Verde
4	Vermelho	Amarelo

Ao sair do estado 4, os semáforos retornam ao estado 1.

**Figura 6.32**

Esquema elétrico de ligação no CLP do projeto de semáforos.



**Figura 6.33**

Mapeamento das saídas do projeto de semáforos.

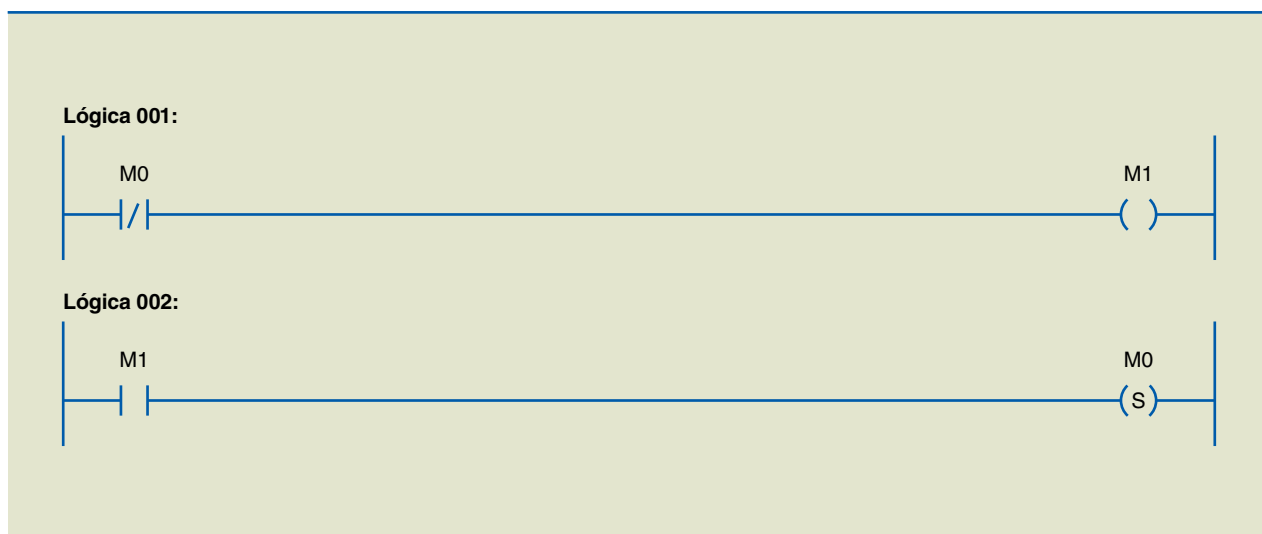
Endereço	Símbolo	Comentário
Q0.0	VERM A	lampada Vermelha Rua A
Q0.1	AMAR A	lampada Amarela Rua A
Q0.2	VERD A	lampada Verde Rua A
Q1.0	VFRM B	lampada Vermelha Rua B
Q1.1	AMAR B	lampada Amarela Rua B
Q1.2	VERD B	lampada Verde Rua B



A solução proposta é descrita a seguir.

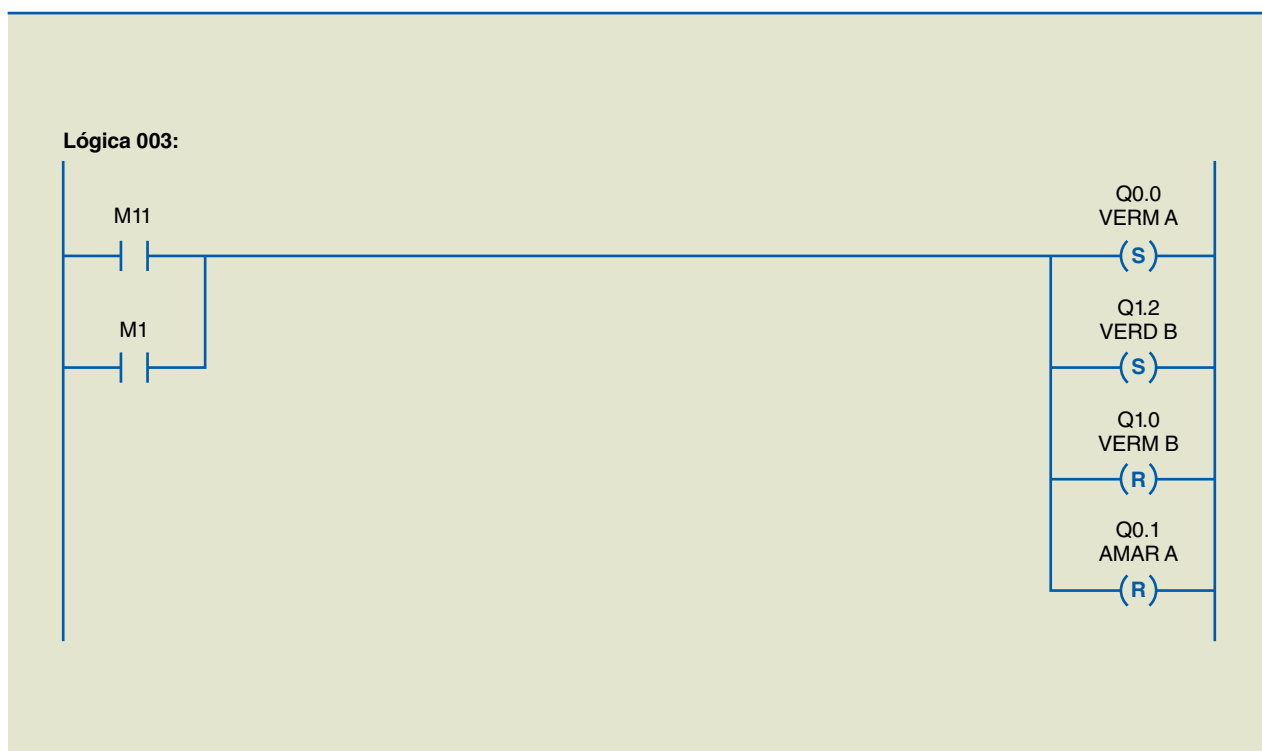
**Lógicas 1 e 2** – Geram um pulso de início do processo em (M1) sempre que o CLP for iniciado em condições normais de operação (figura 6.34). Preveem situação de queda de energia, não sendo necessário um botão de *start* para o processo ser inicializado.

**Figura 6.34**  
Lógicas 1 e 2.



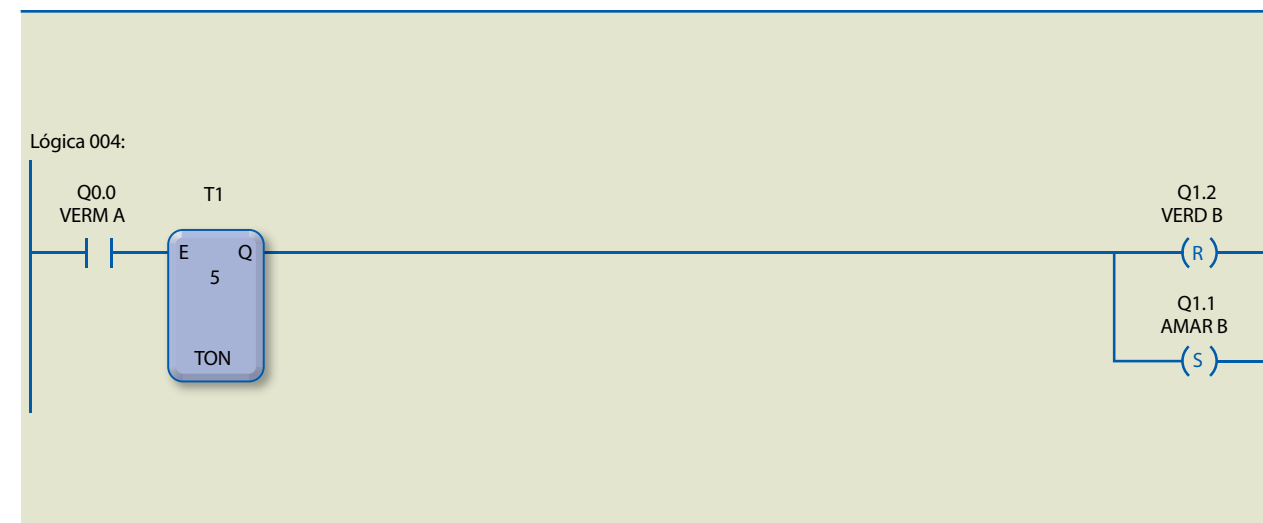
**Figura 6.35**  
Lógica 3.

**Lógica 3** – Uma vez que o processo seja iniciado por (M1) ou reiniciado por (M11), a lógica define a situação inicial de ligar a lâmpada vermelha do semáforo A e a verde do B, apagando a vermelha do B e a amarela do A (figura 6.35).



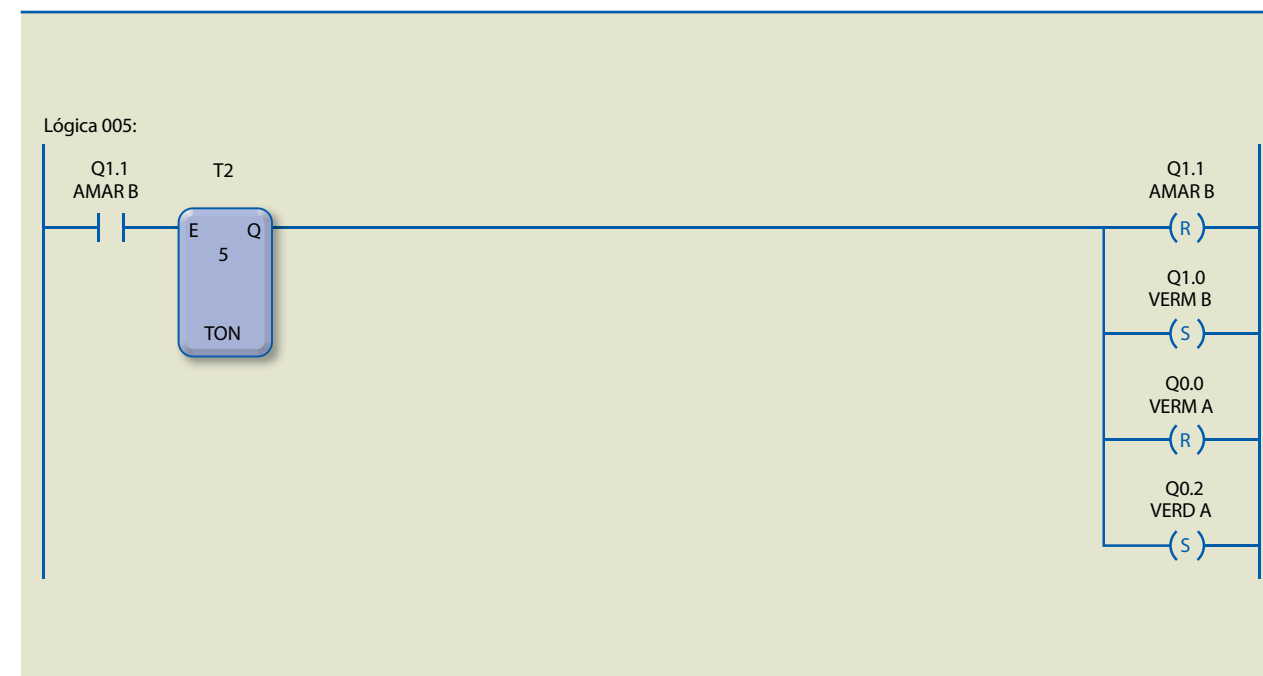
**Lógica 4** – Uma vez acionada a lâmpada vermelha do semáforo A, contam-se 5 segundos para que a lâmpada verde do B se apague e a amarela do B seja acionada (figura 6.36).

**Figura 6.36**  
Lógica 4.



**Lógica 5** – Uma vez acionada a lâmpada amarela do semáforo B, contam-se 5 segundos e a lógica, então, segue apagando a lâmpada amarela do B e a vermelha do A e acendendo a vermelha de B e a verde de A (figura 6.37).

**Figura 6.37**  
Lógica 5.



**Lógica 6** – Uma vez acionada a lâmpada vermelha do semáforo B, contam-se 5 segundos para que a lâmpada verde do A se apague e a amarela do A seja acionada (figura 6.38).





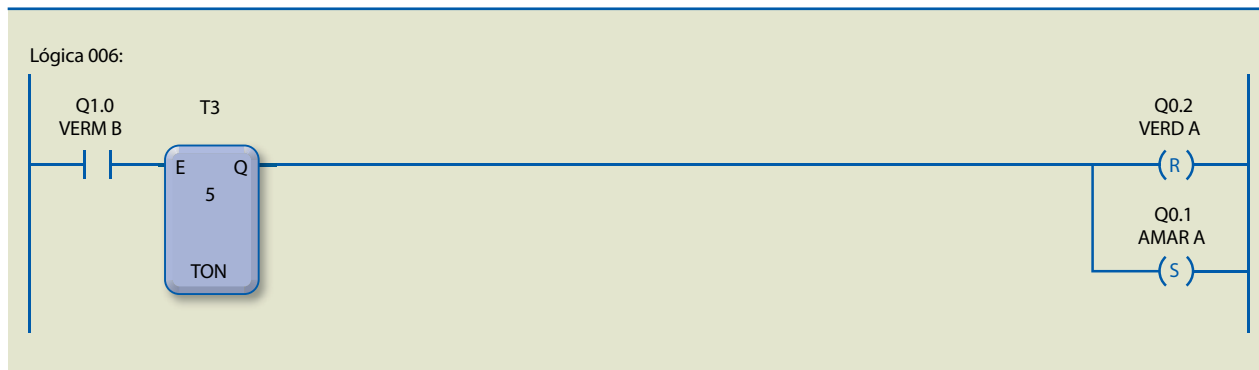


Figura 6.38

Lógica 6.

**Lógica 7** – Ao final de um ciclo, após 5 segundos que a lâmpada amarela do semáforo A foi acionada, a memória (M11) aciona o processo da lógica 3, reiniciando o ciclo por tempo indeterminado (figura 6.39).

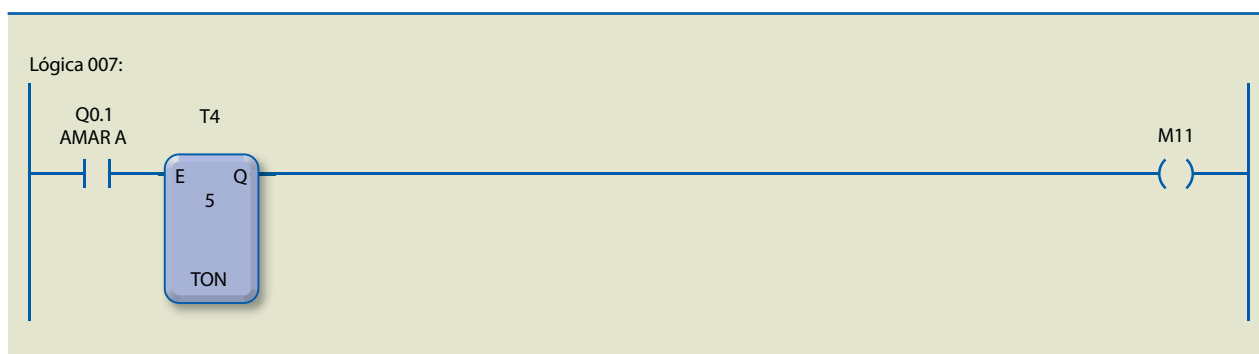


Figura 6.39

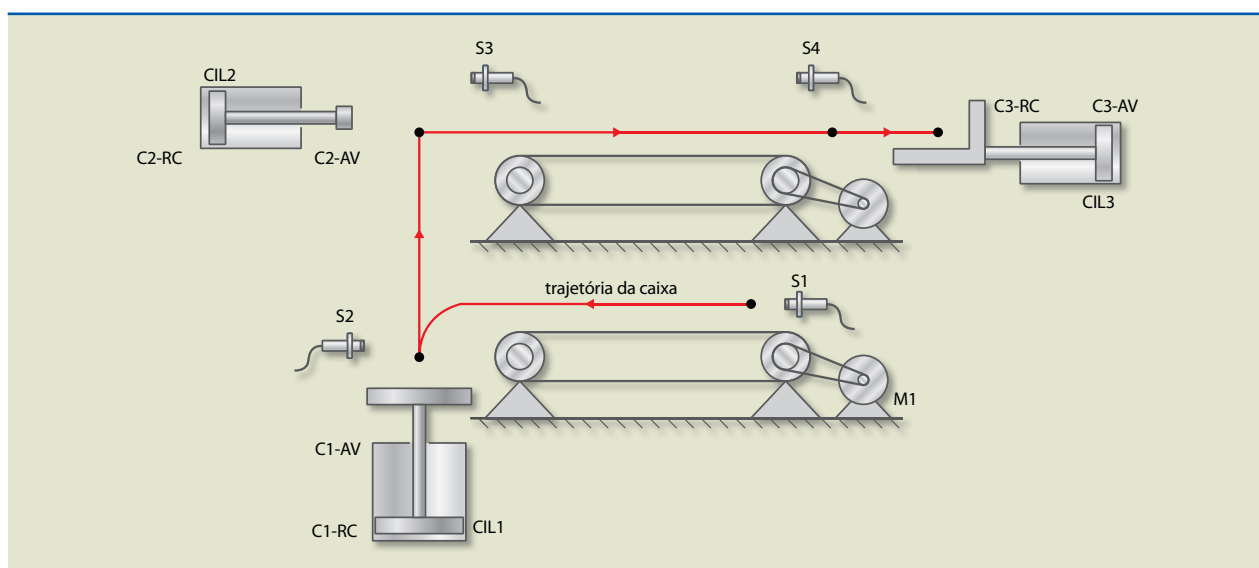
Lógica 7.

### 6.6.4 Transportadora versão 2

Figura 6.40

Transportadora versão 2.

O sistema da figura 6.40, composto por duas esteiras transportadoras e três cilindros pneumáticos, foi projetado para o transporte de caixas.



O esquema de ligação no CLP é apresentado na figura 6.41; e o mapeamento das entradas e saídas utilizado, na figura 6.42.

Figura 6.41

Esquema de ligação no CLP do projeto de transportadora versão 2.

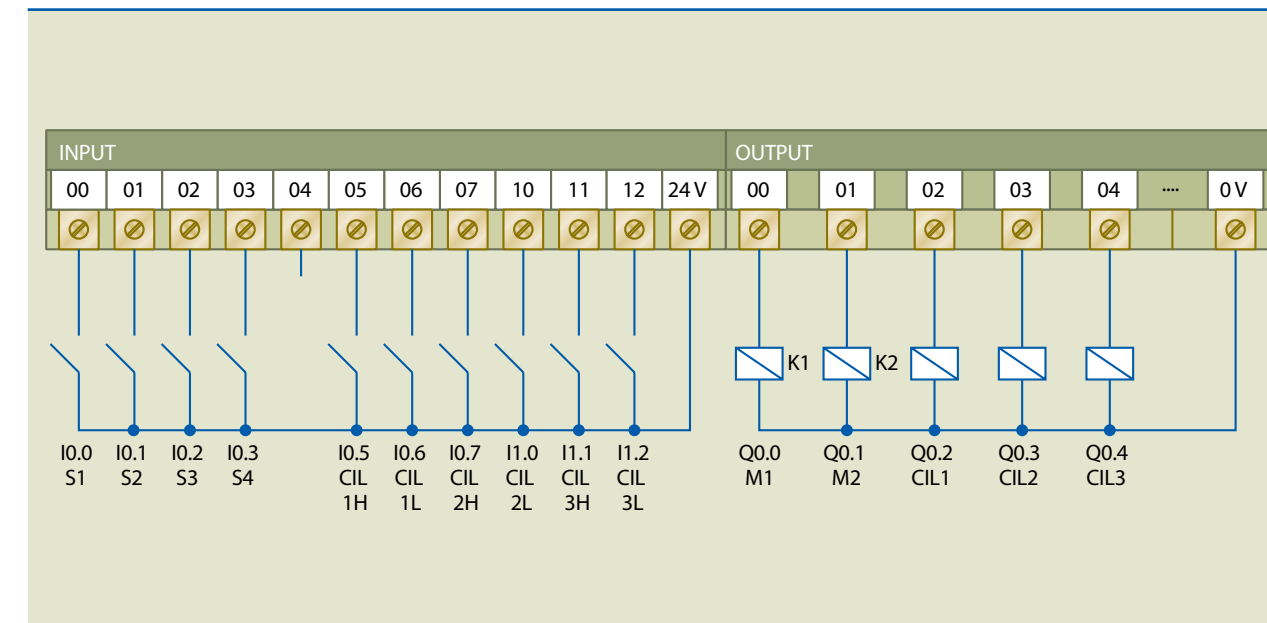


Figura 6.42

Mapeamento das entradas e saídas do projeto de transportadora versão 2.

Endereço	Símbolo	Comentário
M0	PASSO 1	
M1	PASSO 2	
M2	PASSO 3	
M3	PASSO 4	
M4	PASSO 5	
M5	PASSO 6	
M6	PASSO 7	
M7	PASSO 8	
M8	PASSO 9	
M9	PASSO 10	
I0.0	SENS S1	
I0.1	SENS S2	
I0.2	SENS S3	
I0.3	SENS S4	
I0.5	CIL 1 H	
I0.6	CIL 1 L	
I0.7	CIL 2 H	
I1.0	CIL 2 L	
I1.1	CIL 3 H	
I1.2	CIL 3 L	
Q0.0	MOT M1	
Q0.1	MOT M2	
Q0.2	1 CILINDR	
Q0.3	2 CILINDR	
Q0.4	3 CILINDR	
M20	TEMP 105	



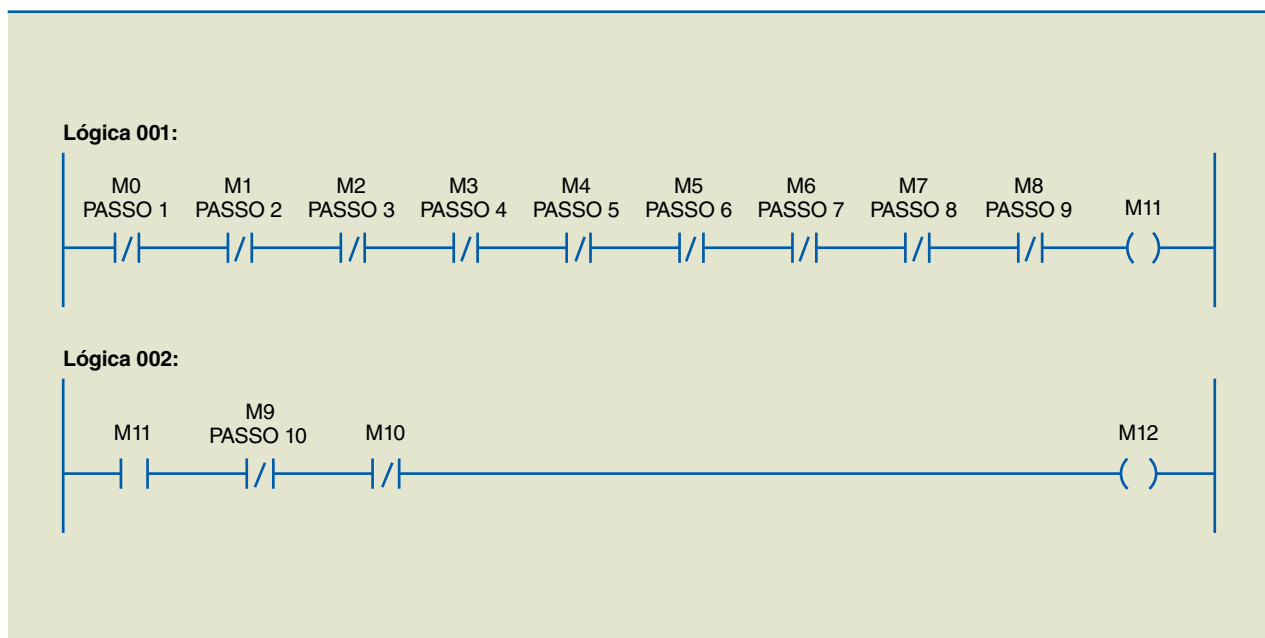
Resumo dos 10 passos e dos respectivos eventos de transição para o projeto transportadora versão 2:

- 10 – Aguarda sensor S1.
- 1 – Liga motor M1 e aguarda sensor S2.
- 2 – Desliga motor M1, avança cilindro CIL1 e aguarda cilindro CIL1 avançado.
- 3 – Mantém avanço de cilindro CIL1, avança cilindro CIL2, aguarda cilindro CIL2 avançado e aguarda sensor S3.
- 4 – Liga motor M2 e aguarda ausência de sensor S3.
- 5 – Desliga motor M2, recua cilindro CIL1, recua cilindro CIL2, aguarda cilindro CIL1 recuado e aguarda cilindro CIL2 recuado.
- 6 – Liga motor M2 e aguarda sensor S4.
- 7 – Desliga motor M2, avança cilindro CIL3, aguarda cilindro CIL3 avançado e aguarda ausência de sensor S4.
- 8 – Espera 10 segundos.
- 9 – Recua cilindro CIL3 e aguarda cilindro CIL3 recuado.
- 10 – Aguarda sensor S1 (e volta para 1 início).

Com essas informações, pode-se desenvolver o programa em Ladder utilizando como auxiliar o diagrama de funcionamento do sistema. O programa atende ao solicitado, de maneira que se observam os detalhes descritos a seguir.

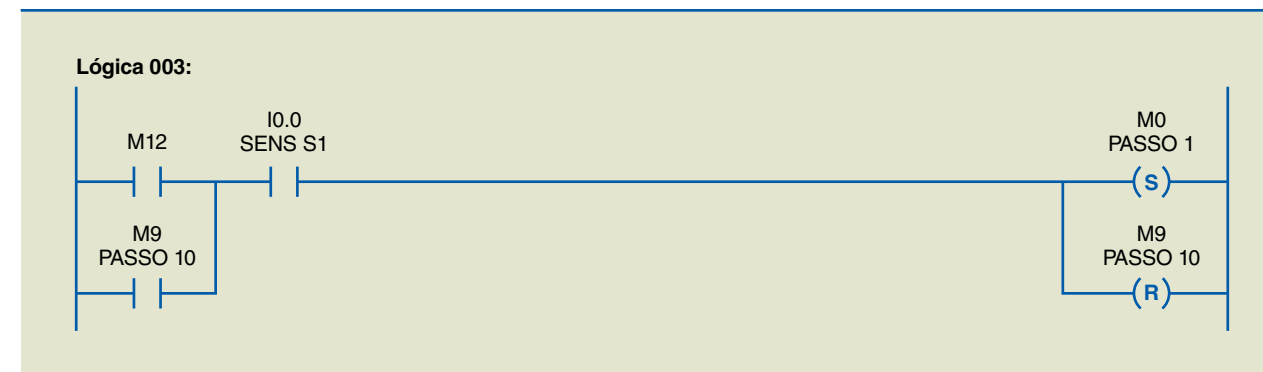
**Lógicas 1 e 2** – Definem que, se nenhum passo está ativo, o primeiro passo do processo é o que deve entrar em execução. O que determina isso é a saída da memória (M12). Na figura 6.43 estão representadas duas linhas para essa função. É preciso ficar atento, pois, se o número de passos for superior ao número de instruções-limite por linha de lógica, será possível cascatear todas as instruções de uma primeira linha armazenando seu resultado em uma memória. Nessas condições, em uma segunda linha, adiciona-se essa memória como pré-requisito para o acionamento da segunda.

Figura 6.43 Lógicas 1 e 2.



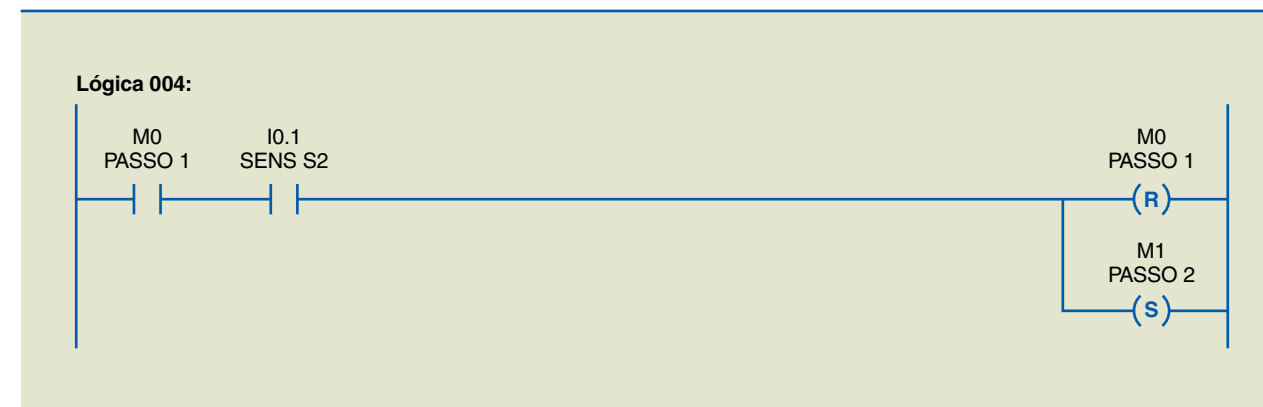
**Lógica 3** – A transição para o passo 1 ocorrerá com S1 acionado, descartando a possibilidade de o passo 10 ser executado, desde que a memória (M12) de início do processo esteja ativa ou que o passo 10 tenha sido executado (figura 6.44).

Figura 6.44 Lógica 3.



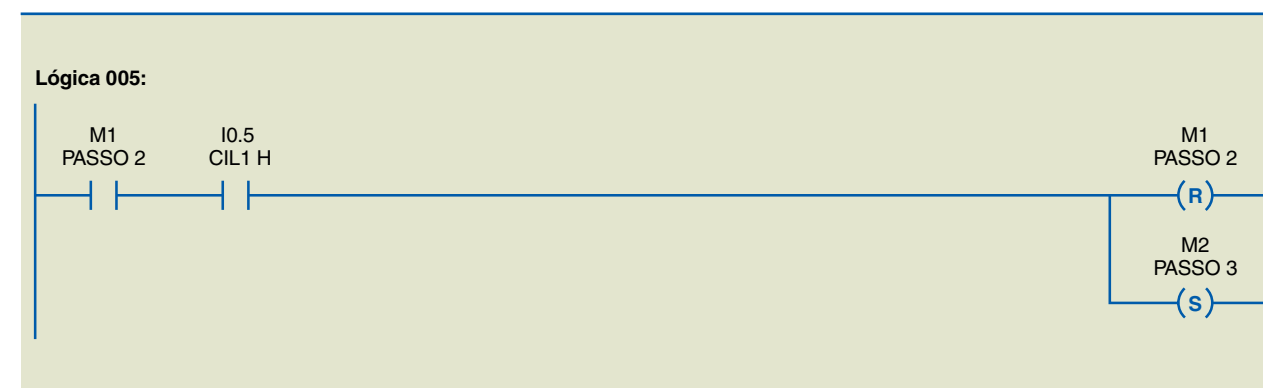
**Lógica 4** – A transição para o passo 2 ocorrerá com S2 acionado, desde que o passo 1 esteja ativo (figura 6.45). Isso descartará ao final a possibilidade de execução do passo 2.

Figura 6.45 Lógica 4.



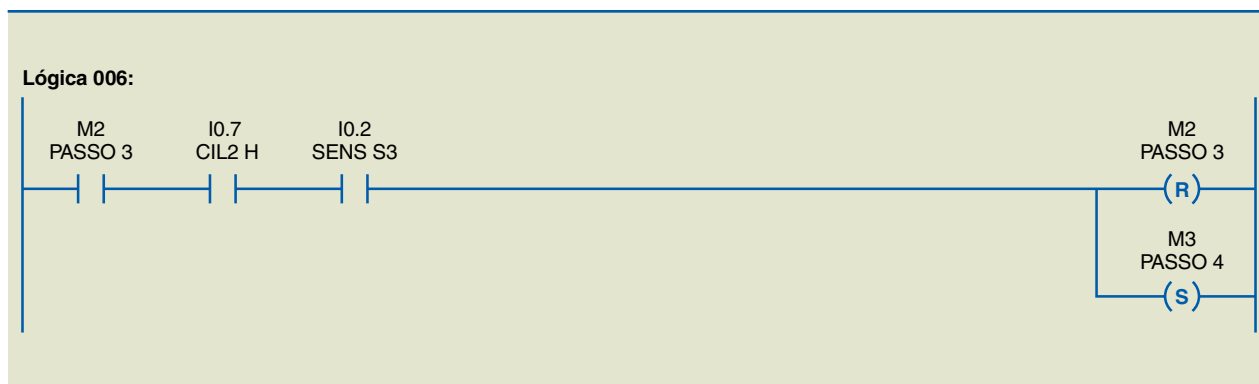
**Lógica 5** – A transição para o passo 3 ocorrerá com CIL1 H acionado, desde que o passo 2 esteja ativo (figura 6.46). Isso descartará ao final a possibilidade de execução do passo 3.

Figura 6.46 Lógica 5.



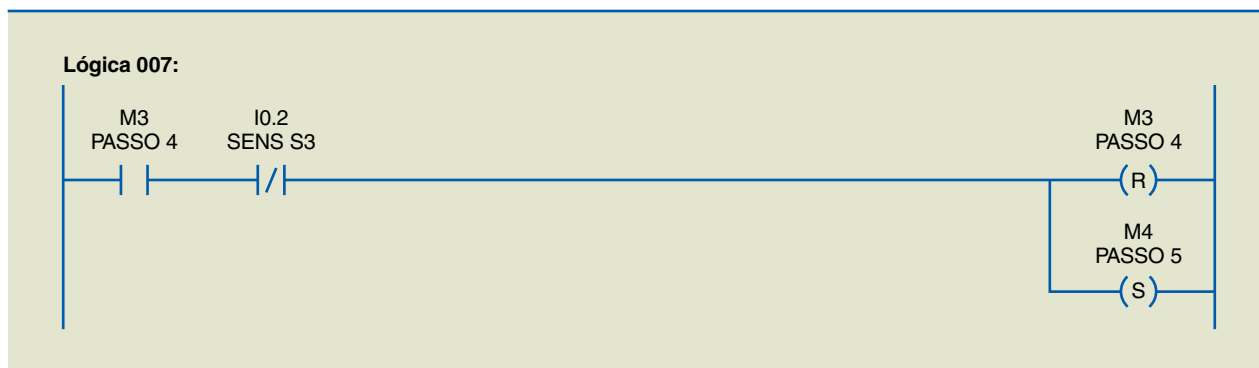
**Figura 6.47**  
Lógica 6.

**Lógica 6** – A transição para o passo 4 ocorrerá com CIL2 H e S3 acionados, desde que o passo 3 esteja ativo (figura 6.47). Isso descartará ao final a possibilidade de execução do passo 4.



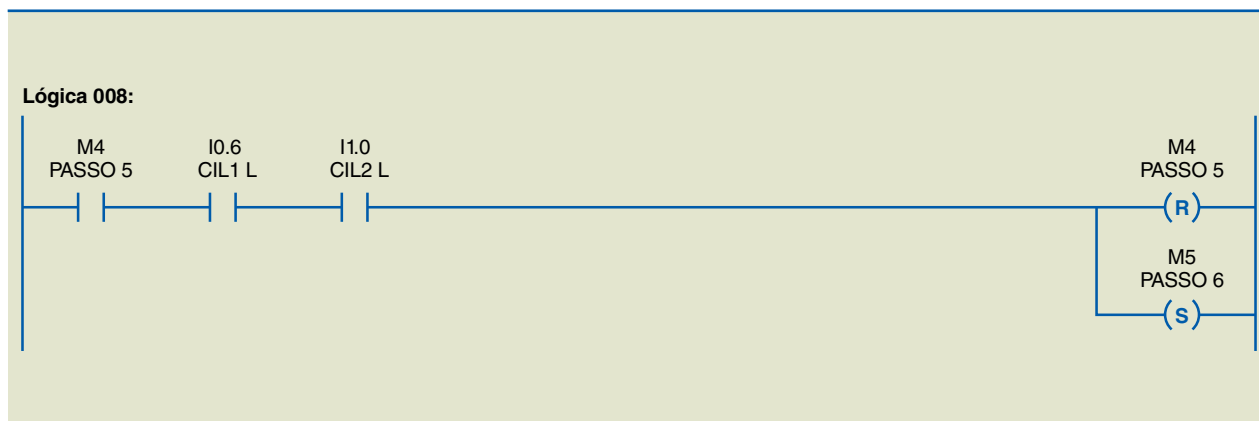
**Figura 6.48**  
Lógica 7.

**Lógica 7** – A transição para o passo 5 ocorrerá com S3 inativo, desde que o passo 4 esteja ativo (figura 6.48). Isso descartará ao final a possibilidade de execução do passo 5.



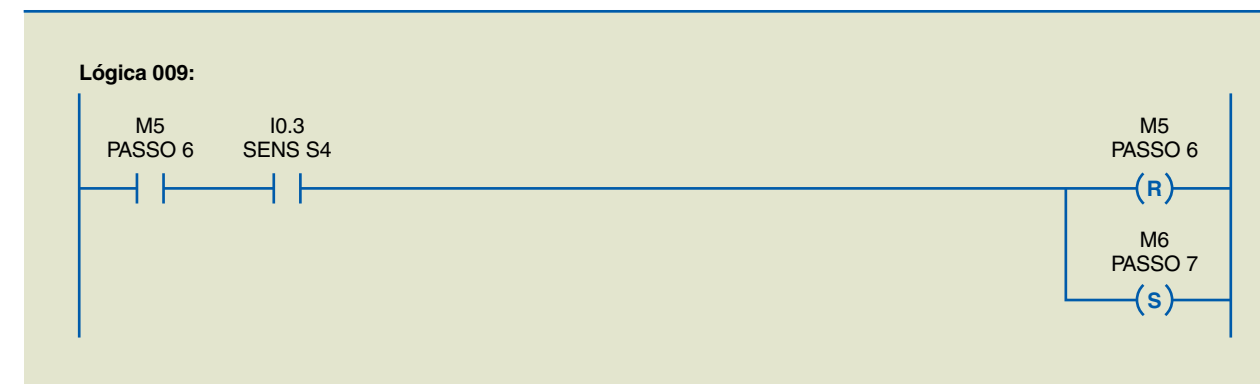
**Figura 6.49**  
Lógica 8.

**Lógica 8** – A transição para o passo 6 ocorrerá com CIL1 L e CIL2 L acionados, desde que o passo 5 esteja ativo (figura 6.49). Isso descartará ao final a possibilidade de execução do passo 6.



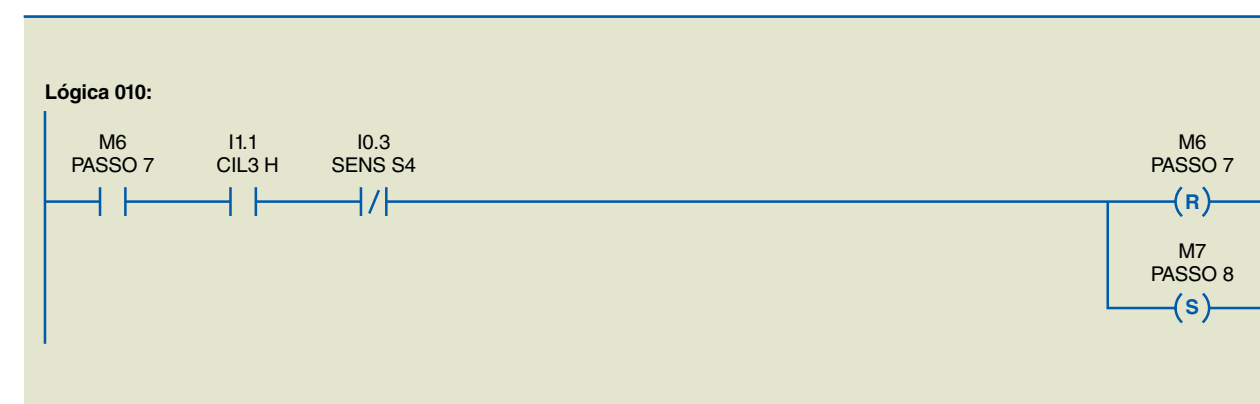
**Lógica 9** – A transição para o passo 7 ocorrerá com S4 acionado, desde que o passo 6 esteja ativo (figura 6.50). Isso descartará ao final a possibilidade de execução do passo 7.

**Figura 6.50**  
Lógica 9.



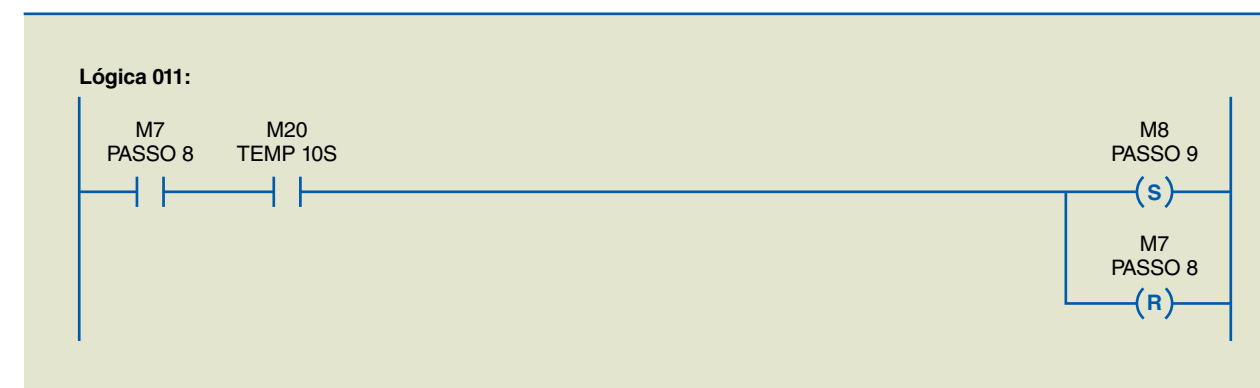
**Lógica 10** – A transição para o passo 8 ocorrerá com CIL3 H ativo e S4 inativo, desde que o passo 7 esteja ativo (figura 6.51). Isso descartará ao final a possibilidade de execução do passo 8.

**Figura 6.51**  
Lógica 10.



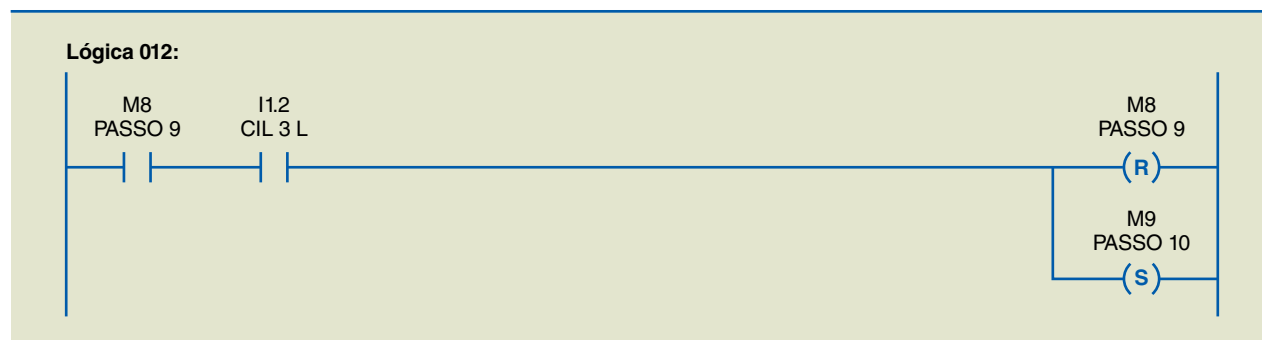
**Lógica 11** – A transição para o passo 9 ocorrerá com TEMP 10S acionado, desde que o passo 8 esteja ativo (figura 6.52). Isso descartará ao final a possibilidade de execução do passo 9.

**Figura 6.52**  
Lógica 11.



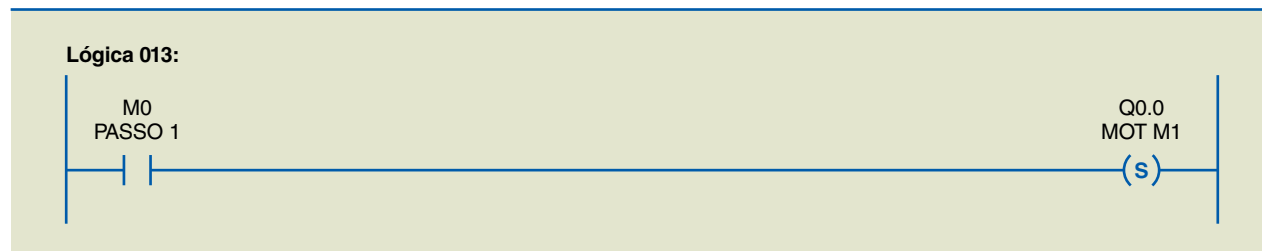
**Figura 6.53**  
Lógica 12.

**Lógica 12** – A transição para o passo 10 ocorrerá com CIL3 L acionado, desde que o passo 9 esteja ativo (figura 6.53). Isso descartará ao final a possibilidade de execução do passo 10 e permitirá que o processo seja reiniciado na lógica 3.



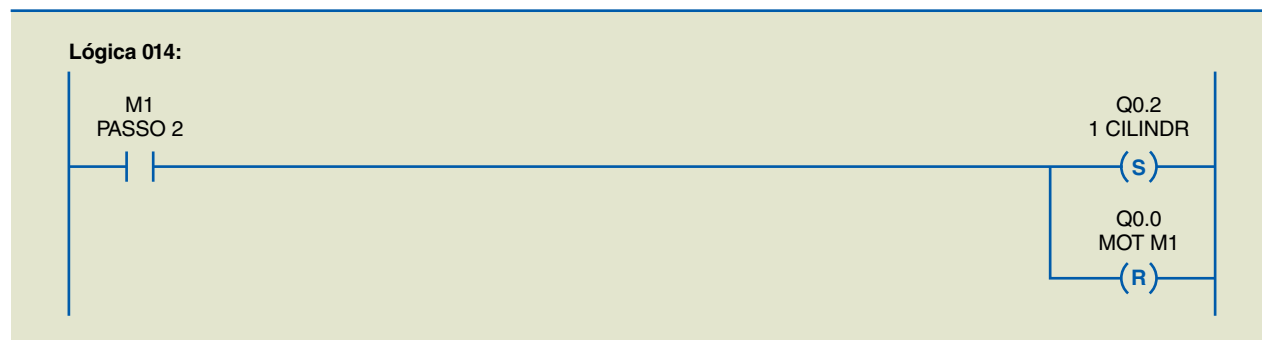
**Figura 6.54**  
Lógica 13.

**Lógica 13** – O passo 1 consiste em acionar o motor M1 (figura 6.54).



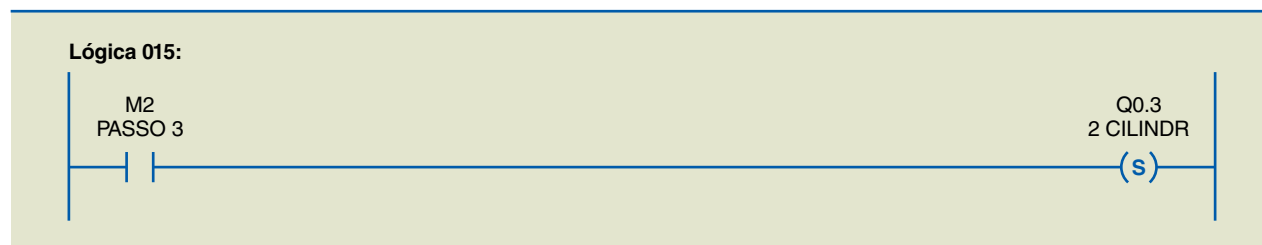
**Figura 6.55**  
Lógica 14.

**Lógica 14** – O passo 2 consiste em acionar o cilindro 1 e desligar o motor M1 (figura 6.55).



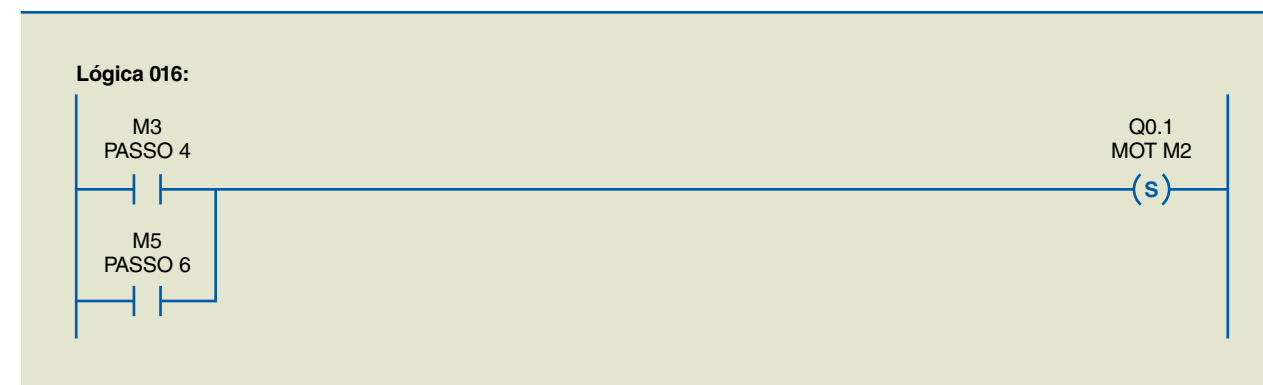
**Figura 6.56**  
Lógica 15.

**Lógica 15** – O passo 3 consiste em acionar o cilindro 2 (figura 6.56).



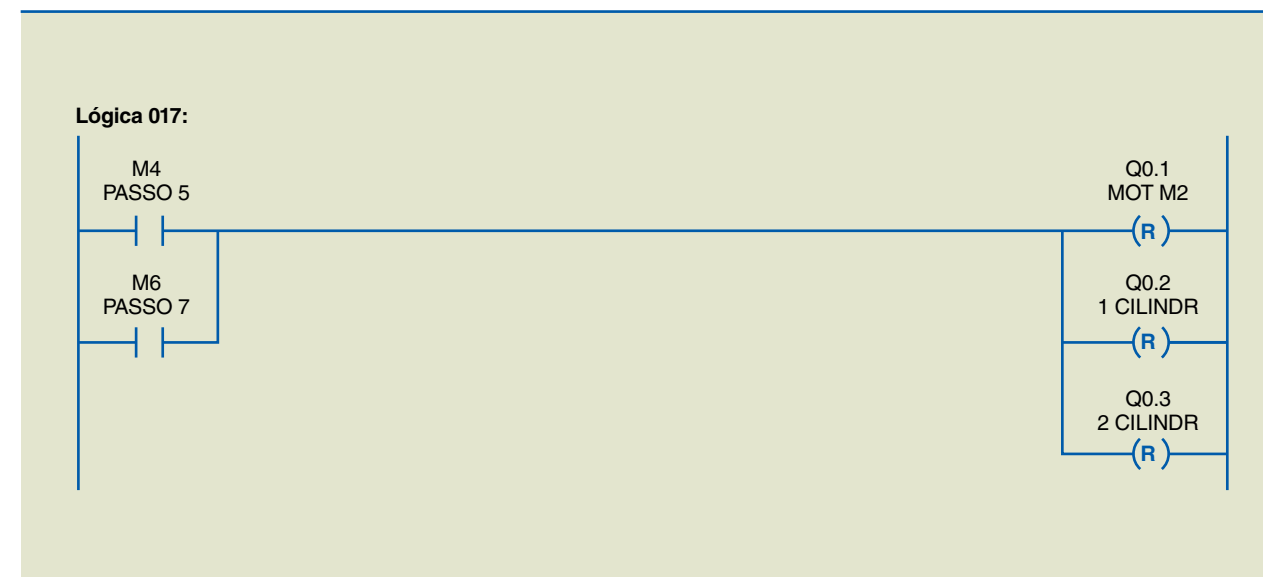
**Lógica 16** – O passo 4 ou 6 consiste em acionar o motor M2 (figura 6.57).

**Figura 6.57**  
Lógica 16.



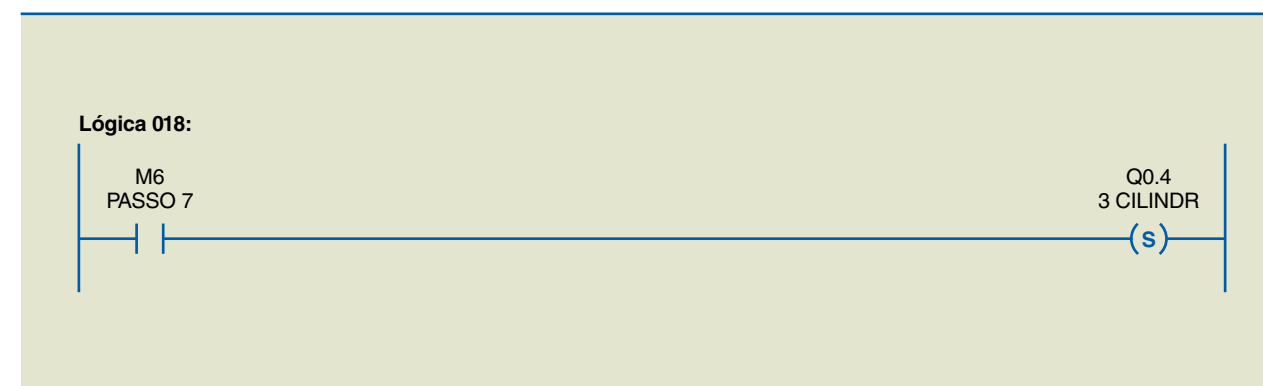
**Lógica 17** – O passo 5 ou 7 consiste em desligar o motor M2, recuar o cilindro 1 e recuar o cilindro 2 (figura 6.58).

**Figura 6.58**  
Lógica 17.



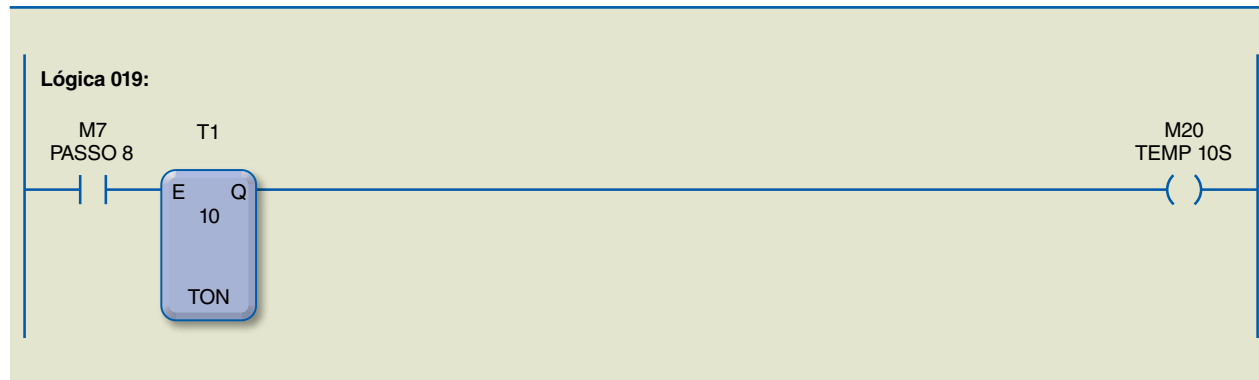
**Lógica 18** – O passo 7 consiste em acionar o cilindro 3 (figura 6.59).

**Figura 6.59**  
Lógica 18.



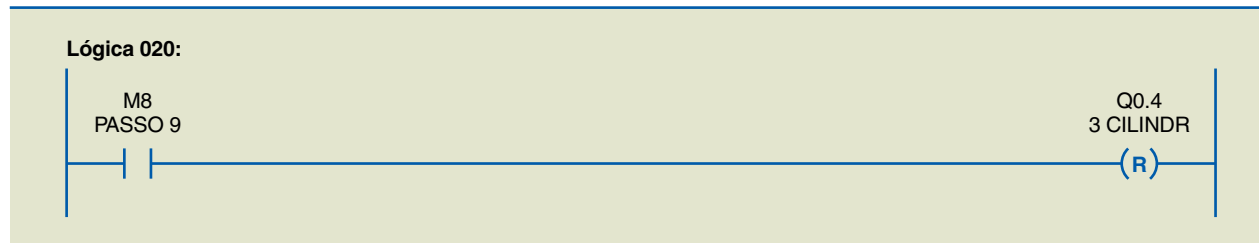
**Figura 6.60**  
Lógica 19.

**Lógica 19** – O passo 8 consiste em acionar uma bobina auxiliar depois de esperar 10 segundos (figura 6.60).



**Figura 6.61**  
Lógica 20.

**Lógica 20** – O passo 9 consiste em recuar o cilindro 3 (figura 6.61).



# Capítulo 7

## Interface

## Homem-Máquina



Interface homem-máquina (IHM), ou, em inglês, *machine-man interface* (MMI), é um dispositivo que se comunica diretamente com o CLP, utilizado para visualizar dados do processo que executem funções de monitoramento, controle de máquinas, processos e instalações industriais. É por meio da IHM que o operador pode interagir com o sistema controlado, exercendo uma função extremamente importante, pois, com as informações recebidas da IHM, ele pode realizar intervenções que mantenham a segurança, o bom funcionamento e a disponibilidade de serviços. Seus principais componentes são: visor (*display*), teclas e botões para navegação ou inserção de dados, barramentos para placas de expansão, portas de comunicação e *software*.

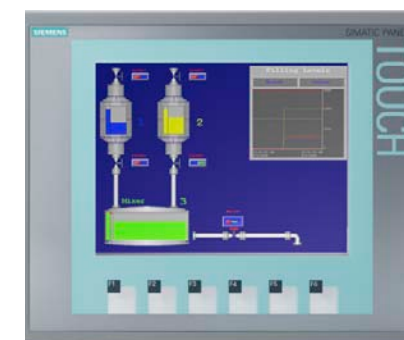
Existem diversas tecnologias de controle, e o interfaceamento entre este e o comando proporciona excelente integração do operador com a máquina. O controle pode ser por botoeiras, *thumbwheels* e chaves seletoras para a entrada de dados, além de sinaleiros, alarmes, instrumentos indicadores de painel e anunciadores para o monitoramento do processo. A opção com os dispositivos citados requer o projeto e a montagem de enormes painéis, com grande quantidade de fios conectados à planta, diminuindo a flexibilidade no caso de ampliações da planta e alterações no processo.

Mesmo nas IHMs mais simples, o operador pode monitorar e alterar a maioria dos operandos do CLP. Por meio de telas pré-programadas, ele tem acesso às variáveis de interesse, atuando quando necessário, pois visualiza os alarmes e as falhas ocorridas.

Há IHMs simples, com poucas linhas de caracteres alfanuméricos, LEDs e botões, assim como modelos sofisticados, com *displays* gráficos coloridos e tela sensível ao toque (*touch screen*). Os modelos mais simples são adequados para o controle de pequenos equipamentos e os avançados permitem que se criem telas coloridas exibindo diagramas de partes do processo, com indicação das variáveis de interesse e possibilidade de atuação por botões virtuais. A função de IHM também pode ser executada por um computador pessoal ou industrial e um programa supervisor.

Com as IHMs, é fácil acrescentar novas telas ou criar novas estratégias de interação operador-máquina, por meio de reprogramação. Além disso, não há necessidade de refazer a fiação entre a IHM e o processo, nem de acrescentar sinaleiros e instrumentos ao painel. Com relação à fiação, basta um par de fios para conectá-la por interface serial ou rede de dados ao CLP.

**Figura 7.1**  
Simatic Panels, da Siemens.



Normalmente, as IHMs são programadas pelo mesmo *software* que programa o CLP. Elas possuem uma ou mais entradas para comunicação, pois, na maioria dos casos, são configuradas como escravas do CLP.

## 7.1 Tipos de IHM

Existem muitos fabricantes de IHM, porém o *software* de programação é do proprietário, não havendo padronização. Isso gera alguns inconvenientes, como o tempo gasto pelo programador na aprendizagem e sua dificuldade em utilizar diversos ambientes.

Os diversos tipos de IHMs disponíveis no mercado variam em tamanho, número e funções de teclas, tipo e tamanho do *display*, resolução da tela, tipo de interface, grau de proteção, memória etc. Seria como dizer que para cada aplicação existe uma IHM.





**Figura 7.2**

Tipos de IHM.

Os tipos mais comuns de IHM são:

- Frontais com teclado e *display* de sete segmentos.
- Frontais de LCD com teclado de funções.
- Terminais de vídeo.
- Terminais de vídeo com tela sensível ao toque.

As IHMs costumam ser utilizadas para:

- Indicar falhas em máquinas e processos.
- Mostrar *status* das máquinas e processos.
- Permitir intervenção do operador no processo.
- Monitorar variáveis de processo, como contagem de componentes, tempos de fabricação, temperatura e pressão.

Esses dispositivos dividem-se em duas categorias:

- **IHMs passivas** – Monitoram eventos ou *status* das máquinas e não interferem no processo. Geralmente não possuem processador interno.
- **IHMs ativas** – Fazem parte do processo e possuem processador interno. Alguns modelos armazenam parte do *software* aplicativo e conectam-se a diversos CLPs em rede.

## 7.2 Configuração e programação da IHM

As IHMs podem ser configuradas para enviar sinais de atuação ou simplesmente monitorar um sinal, trabalhando no modo de desenvolvimento ou no modo de operação.

No modo de desenvolvimento, são criadas as telas gráficas e animações representativas do processo. No modo de operação, são apresentadas janelas animadas que mostram o andamento do processo, além de possibilitar a captura e o armazenamento de dados, a criação de gráficos de tendências e a geração de alarmes.

De modo geral, a programação dos CLPs e das IHMs é feita com ferramentas de plataforma amigável. A maioria dos *softwares* existentes no mercado possui assistentes de parametrização das funções de comunicação em rede, biblioteca de objetos com gráfico de barras, indicadores de medidas, seletores, potenciômetros, curvas de tendências e símbolos do tipo *bitmap*, entre outros.

A área de trabalho relacionada à IHM corresponde à área de trabalho do *software* de programação em que se localizam as principais ferramentas de parametrização e programação da IHM. O acesso a essa área varia de acordo com o fabricante do *software* de programação, porém é possível definir a existência de três grupos principais: navegação, evento e alarme.

### 7.2.1 Navegação

Normalmente, utiliza-se um conjunto de páginas para compor determinada aplicação. Ele pode ser usado para mudanças entre processos, seleção de área de visualização de dados, modos de operação e também para intervenção direta do operador no sistema. A área de navegação tem como principal objetivo a visualização de parâmetros do CLP. Em geral, entre telas do mesmo grupo, é criado um *link*, como mostra o exemplo da figura 7.3, em que as teclas Opção 01 e Opção 02 selecionam o modo de operação do processo.



**Figura 7.3**

Opções exibidas na tela da IHM.

### 7.2.2 Evento

Qualquer mudança nas condições de funcionamento de uma máquina ou processo pode significar um evento, o qual, normalmente, consiste na passa-



gem de um bit qualquer do CLP de “0” para “1”. Esse evento deve ser mostrado ao operador, em telas denominadas telas de evento, conforme ilustra a figura 7.4. O menu de configuração da mensagem desse evento é apresentado na figura 7.5. Quando ocorrer um evento, a página associada a ele será exibida na IHM. Dependendo do tipo de mensagem recebida, o operador tem de executar tarefas, como reiniciar ou interromper o processo, alimentar ou retirar peça produzida etc. Para sair da tela de evento, ele deve concluir a tarefa e pressionar a tecla indicada. A IHM, então, volta para a tela de navegação anterior ao evento ocorrido. As telas de evento também são organizadas em conjuntos, porém, essa organização só tem efeito para o operador da IHM, e nenhum para o CLP.

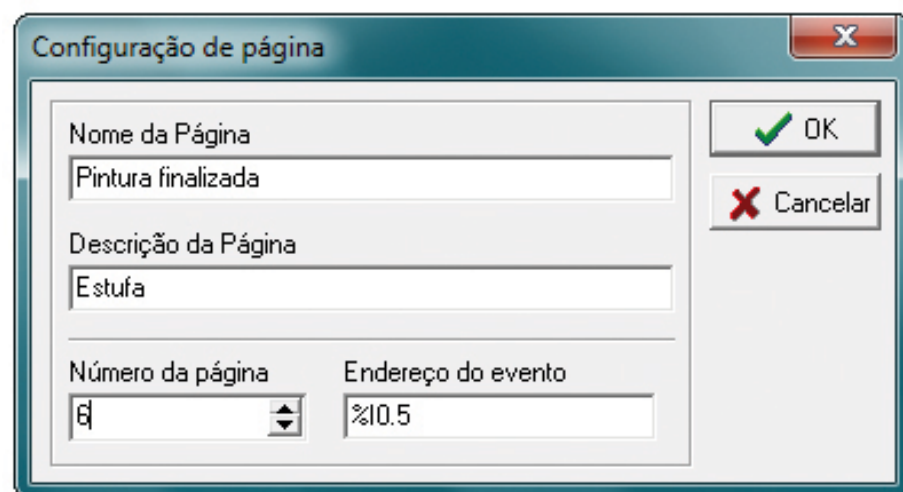
**Figura 7.4**

Telas de evento:  
(a) secagem,  
(b) aquecimento e  
(c) resfriamento.



**Figura 7.5**

Exemplo de programação de tela de evento.



### 7.2.3 Alarme

A IHM possui as funções que o operador necessita para comandar e alterar o funcionamento de máquinas ou processos. A tela de alarme mostra que algo imprevisto ou anormal está ocorrendo na máquina, e tal situação pode requerer ações corretivas do operador, evitando um agravamento.

Quando um alarme dispara, a tela correspondente àquele alarme começa a piscar. Para evitar qualquer risco à segurança, a informação exibida na tela deve ser suficientemente legível, explícita e de interpretação imediata. De acordo com o teor da mensagem recebida, o operador pode ter de intervir rapidamente, acionando um ou vários botões ou teclas. Essas ações serão facilitadas se os botões ou as teclas estiverem devidamente identificados ou marcados, com símbolos normalizados. Caso ocorram dois ou mais alarmes simultâneos, as telas desses alarmes serão mostradas alternadamente na tela da IHM até a intervenção do operador.

Um alarme é caracterizado por um bit e permanecerá ligado enquanto o bit estiver em “1”. Se o bit for para “0”, a tela de alarme será automaticamente desativada, sem a intervenção do operador.

**Figura 7.6**

Sinalização: tela de alarme piscando.



**Figura 7.7**

Sinalização com dois alarmes, que são mostrados alternadamente.





### 7.3 Considerações sobre a aplicação de IHMs

A aplicação de IHM a um sistema deve levar em conta o conjunto operador, máquina e processo a ser automatizado. Do ponto de vista do operador, consideram-se aspectos baseados em ergonomia, grau de intervenção, nível de segurança, proteção a informações etc.; do ponto de vista do CLP, aspectos ligados à estrutura de dados, à estrutura do programa, à colocação em funcionamento, à evolução etc.; do ponto de vista do processo, aspectos relacionados à sua operação, à falhas possíveis, à estratégias de recuperação de falhas, à estratégias de manutenção preventiva etc.

### 7.4 Etapas de um projeto com CLP e IHM

O projeto que utiliza CLP e IHM deve ser executado de acordo com as etapas apresentadas a seguir, para que sejam atingidos os principais objetivos da automação industrial.

**1ª etapa** – São especificados os tipos e a quantidade de sensores e atuadores do processo. Essas especificações definem a segunda etapa.

**2ª etapa** – São escolhidos os controladores lógicos programáveis, com definição do fabricante, modelo, processador e módulos que serão utilizados.

**3ª etapa** – São definidas as redes de comunicação, os protocolos e o *bus*.

**4ª etapa** – É elaborada a definição da IHM, que compreende:

- **Definição da lista das páginas** – Página inicial, sumário, página do processo, janelas *pop-up*, gráficos etc.
- **Definição da navegação entre páginas** – Botões, zonas sensíveis ao toque etc.
- **Definição das informações a visualizar nas páginas** – Sinalizadores, valores, curvas, diagramas, alarmes etc.
- **Definição dos comandos a enviar ao CLP** – Botões, zona sensível ao toque, digitação de valores etc.

A resposta às questões a seguir dá subsídios para a correta escolha da IHM:

1. Quais são as informações necessárias e quem vai utilizá-las?
2. De que maneira essas informações devem ser disponibilizadas?
3. Como o usuário espera que as informações sejam apresentadas?
4. Em que momento a informação deve ser disponibilizada ao usuário?
5. Existe operador com necessidades especiais?
6. Será necessária a utilização de som?
7. Quais opções devem ser disponibilizadas ao operador?

Sabendo o modelo a ser usado e atendendo aos itens que determinam a IHM correta, a próxima etapa é a programação da IHM, cujos passos, de maneira geral, são:

1. Criar as telas em um microcomputador utilizando um *software* para CLP ou próprio para IHM.
2. Fazer o *download* das telas criadas para o CLP ou IHM.

3. Conectar a unidade ao CLP.

4. Ler e escrever na IHM, usando as posições de memória para obter a entrada e a atualização das telas.

5. Testar exaustivamente as telas de navegação, eventos e alarme.

Ao utilizarmos *softwares* de programação para a criação de páginas de aplicação (sumário, processo, gráficos, alarmes, receitas etc.), muitas vezes temos de protegê-los. Essas páginas ou alguns de seus campos podem ser protegidos ou bloqueados por uma senha (*password*) contra acessos indevidos ou usuários mal-intencionados.

Os sistemas automatizados necessitam de acesso à informação certa, no instante certo e no lugar certo. A correta especificação da IHM estabelece a comunicação entre o homem e a máquina, utilizando as soluções de mercado de modo simples para o operador e evitando erros de atuação.



# Capítulo 8

## Comunicação em sistemas automatizados

**S**inônimo de agilidade e precisão, a automação industrial passou a ser uma necessidade na maioria das indústrias. Ela pode, por exemplo, coletar dados instantaneamente em uma linha de produção e levá-los à mesa da diretoria, possibilitando uma rápida tomada de decisão em uma negociação – a informação útil e necessária no momento certo.

Outra aplicação é o monitoramento da produção, pois não adianta receber ao final do expediente a informação de que a linha de produção está com um problema de precisão em uma das peças se já se comprometeu a qualidade dos itens produzidos. É essencial que essa informação chegue rapidamente aos responsáveis pelo setor produtivo da indústria para o reparo ser solicitado sem comprometer toda a produção.

No chão de fábrica, informações qualitativas e, muitas vezes, quantitativas do processo são transmitidas aos operadores e monitores da planta industrial instantaneamente por meio de painéis e sinaleiros. Podem ser usados, por exemplo: uma lâmpada (figura 8.1), para indicar que certa operação está ocorrendo ou não; uma sirene, para alertar os funcionários sobre a ocorrência de uma situação potencialmente perigosa; ou um *display* numérico, para mostrar níveis de produção ou informações de processo, como a temperatura com que determinado forno está operando naquele exato instante.

**Figura 8.1**

Lâmpadas de sinalização.



ALTERALTER/SHUTTERSTOCK

**Figura 8.2**



ALTERALTER/SHUTTERSTOCK

Os sinalizadores podem também avisar os operadores da necessidade de intervenções de emergência no processo: o esvaziamento de um reservatório em momento de emergência, a parada de uma máquina por medida de segurança durante um procedimento de manutenção etc. Essas atuações de emergência podem ser executadas por meio de painéis elétricos com botoeiras de comandos (figura 8.3).

**Figura 8.3**

Painéis elétricos com botoeiras.



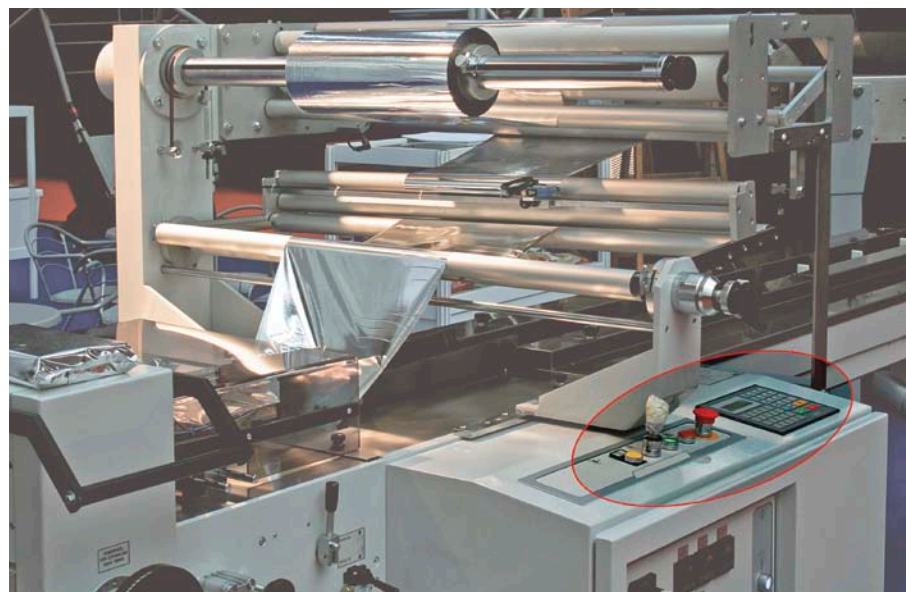
DEJAN LAZAREVIC /SHUTTERSTOCK



As IHMs são úteis, ainda, na sinalização e atuação de equipamentos do processo produtivo (figura 8.4).

**Figura 8.4**

IHM monitorando localmente um processo.



Assim como a informação local é importante em certas áreas de uma empresa, a informação centralizada de processos mais complexos, como os de refinaria, siderurgia e indústrias químicas, também são de grande importância. Nesse tipo de sistema, as informações sobre o que está ocorrendo na planta são enviadas para pontos seguros e distantes do local do processo. Elas são centralizadas em salas de controle (figura 8.5) e monitoradas, 24 horas por dia, por operadores treinados, que devem saber exatamente o que fazer em situações que estejam fora do programado.

**Figura 8.5**

Sala de controle.



Aos gestores do processo, não interessa saber a temperatura do forno, se a bomba necessita de manutenção ou se a prensa está com vazamento de óleo, mas sim os recursos disponíveis (energia, matéria-prima, água etc.), seu consumo, a produção, a carteira de pedidos, a quantidade de produtos estocados, entre outros. Nesse caso, os dados do chão de fábrica devem ser filtrados, organizados e, posteriormente apresentados aos gestores para que analisem as informações e possam tomar decisões.

Como vimos, as informações precisam circular entre os vários níveis do processo produtivo. Este capítulo trata das redes de comunicação empregadas em automação e suas características.

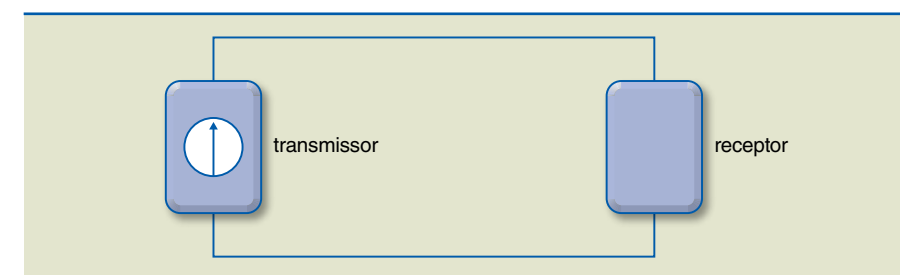
## 8.1 Redes de comunicação

A necessidade de exportar dados do processo local para pontos distantes, seja por questões de segurança ou de redução do custo operacional, seja simplesmente por comodidade ou simplificação no processo, levou à criação das redes industriais. Elas garantem a transmissão de informações relativas ao processo, de maneira rápida, em ambiente eletromagneticamente ruidoso, entre dois ou mais pontos.

As redes industriais surgiram com o advento da computação na indústria, pois até então, o monitoramento a grandes distâncias só podia ser feito com sistemas analógicos baseados em *loop* de corrente.

No sistema de *loop* de corrente, o transmissor é uma fonte de corrente (figura 8.6) e a amplitude da corrente é proporcional ao sinal que está sendo transmitido. A grande vantagem da transmissão em corrente é que, independentemente da queda de tensão nos condutores, a corrente é a mesma em qualquer ponto do *loop*. Assim, a corrente no receptor é a mesma que no transmissor. Se empregarmos uma fonte de tensão, o sinal recebido pelo receptor será o sinal enviado, menos a queda de tensão na resistência do fio, mais o ruído induzido pela radiação eletromagnética dos equipamentos.

De início, utilizou-se o padrão de 0 a 20 mA; posteriormente, o de 4 a 20 mA, que oferece como vantagem a possibilidade de detecção de interrupção no *loop*. Se a corrente for nula, o receptor reconhece que o cabo está interrompido.



**Figura 8.6**

Circuito de um *loop* de corrente.

Com o passar do tempo e o avanço da tecnologia, surgiram diversos meios físicos que possibilitaram reduzir distâncias entre pontos de monitoramento,



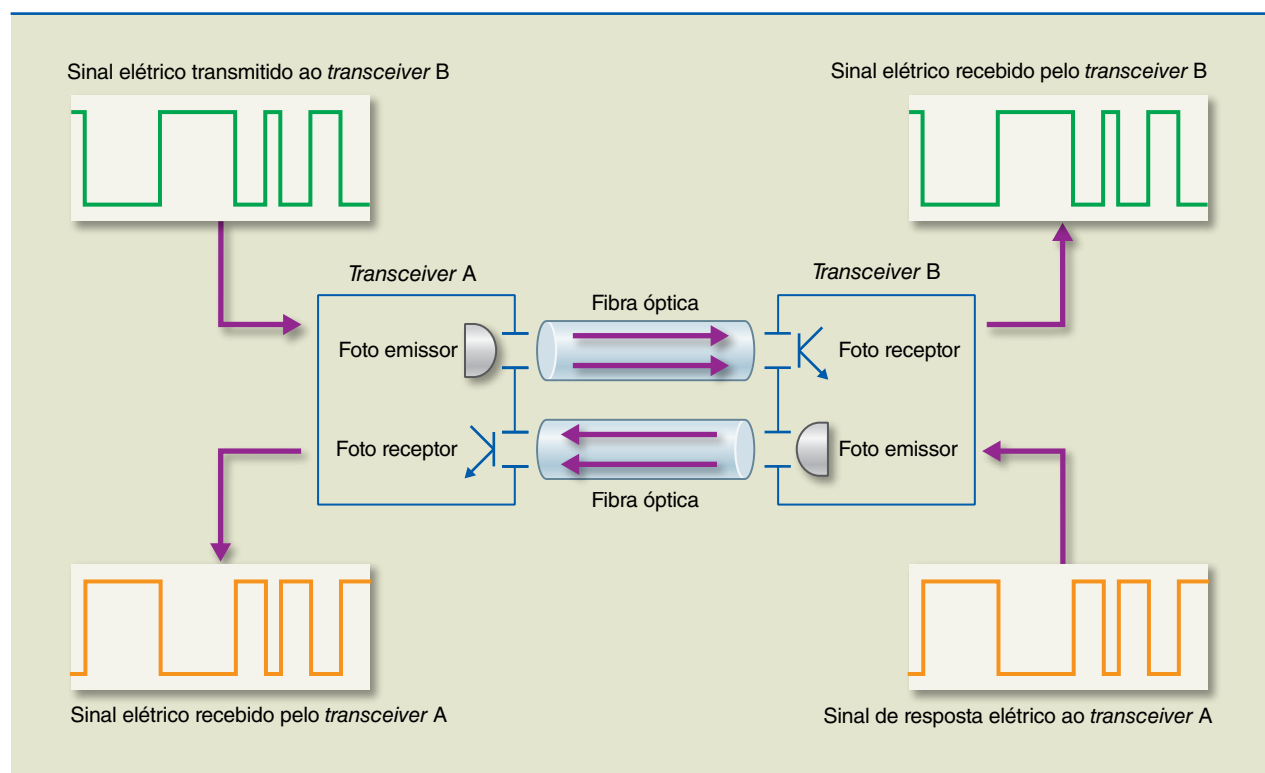
assim como a ideia de incluir vários pontos de medição em uma mesma rede de comunicação. No entanto, grande parte das comunicações feitas em ambientes industriais é baseada em comunicação serial com um ou mais pares de fios metálicos.

Além da forma como as informações são enviadas e recebidas pelo meio físico (protocolo de comunicação), as redes diferem entre si pelo próprio meio físico (e sua imunidade a campos de indução eletromagnética, ambientes corrosivos etc.). Sabemos que capas emborrachadas com núcleos isolados envoltos em filme metálico garantem rigidez mecânica ao cabo, estanqueidade e imunidade eletromagnética.

Atualmente, a quantidade de dados que podem ser enviados em um único par de fios (que, no passado, só possibilitava o envio de informações de um único transdutor) é imensa. A integridade dos dados é garantida pela utilização de protocolos de comunicação específicos.

Outro meio físico utilizado em redes de comunicação industrial é a fibra óptica. Sua aplicação é de maneira serial, ou seja, uma única fibra é capaz de enviar informações, bit a bit, a um receptor no ponto de monitoramento. As vantagens do uso da fibra óptica, em comparação com o par de fios trançados, são a imunidade ao ruído eletromagnético e a maior capacidade de transmissão. A figura 8.7 ilustra um *link* bidirecional com duas fibras.

**Figura 8.7**  
Rede de comunicação em fibra óptica.



No mercado existem equipamentos que convertem sinais elétricos em ópticos, chamados de *transceivers* ou conversores de fibra óptica. No entanto, alguns fa-

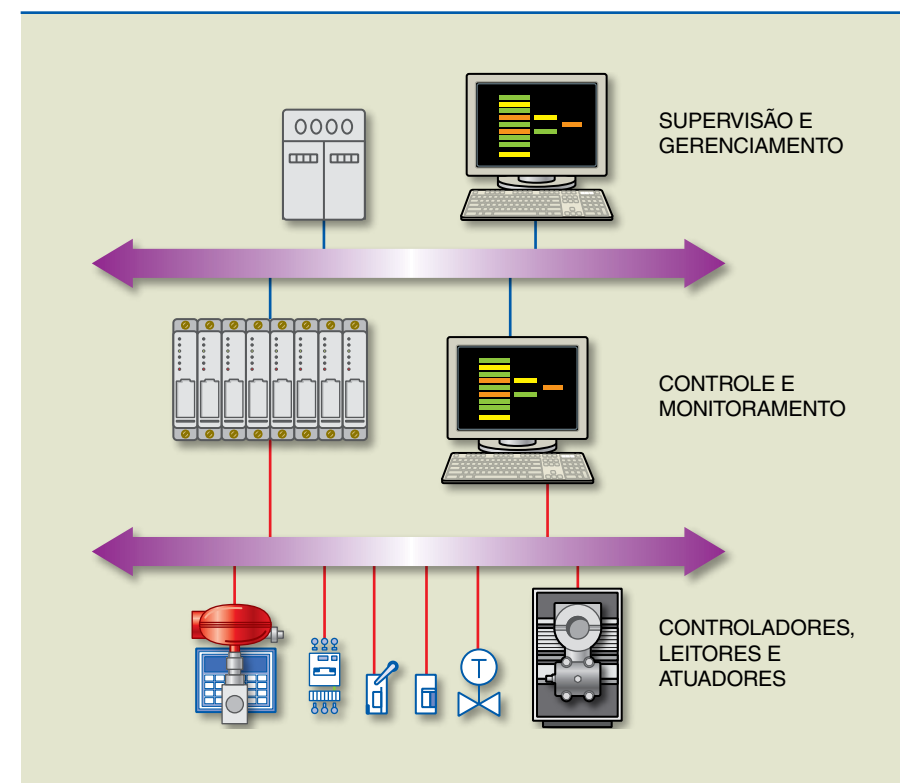
bricantes de *hardware* para automação já possuem interfaces de comunicação com circuitos ópticos em seu *hardware* original, como mostram os exemplos da figura 8.8.

**Figura 8.8**  
Transceivers.



Radiofrequência, ou sistema *wireless*, é o meio de comunicação que tem sido utilizado na indústria em processos de monitoramento e atuação secundária, por suas características e questões de segurança. Esse sistema depende de grande conhecimento técnico dos que trabalham na implantação da rede e será tratado na seção 8.3.

A rede de comunicação industrial normalmente é estruturada em níveis, com funções distintas, conforme a figura 8.9.



**Figura 8.9**  
Níveis hierárquicos de uma rede de comunicação industrial.



O nível mais alto é responsável por supervisionar as operações, efetuar análises críticas quanto à produtividade e à abrangência de produtos, além de tratar de questões de otimização. As informações que chegam a esse nível permitem que gerentes e supervisores de área programem paradas ou fluxos produtivos de acordo com as diretrizes da empresa. Os computadores nas salas de controle centralizam as informações e conectam as redes industriais às redes corporativas, podendo gerar dados para faturamento, gráficos de qualidade e índices de produção, graças à possibilidade um mesmo computador ter interfaces físicas de redes industriais e redes Ethernet.

O nível intermediário possui os dispositivos responsáveis pela operação da planta. Nele estão os CLPs, os computadores industriais e as IHMs, comunicando-se entre si e com os sensores e atuadores. Parte dessa comunicação é feita pelos barramentos de campo (em inglês, *fieldbus*).

O terceiro e último nível é composto pelos sensores e atuadores ligados diretamente ao processo. Estes são os maiores responsáveis por coletar dados da produção e informar como, quando e onde determinadas ações devem ocorrer.

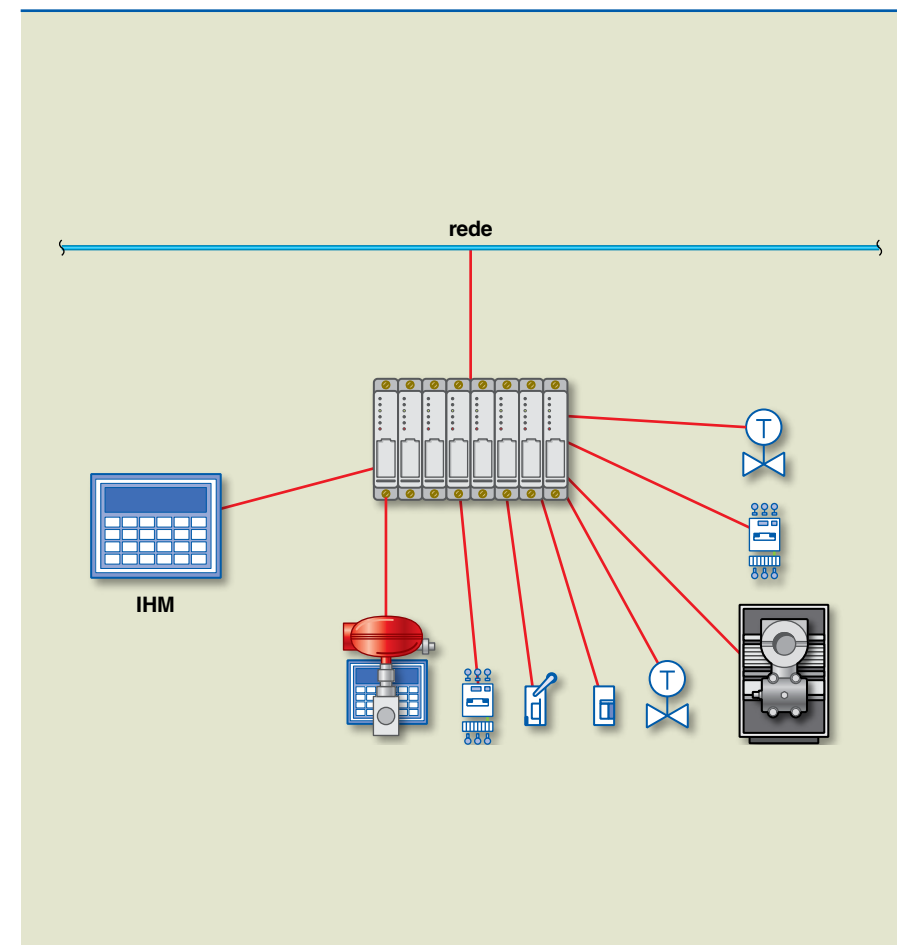
A comunicação entre todos os dispositivos e componentes desses três níveis é imprescindível para o objetivo final da automação: realizar operações de maneira automática e segura para os operadores envolvidos no menor tempo possível, gerando maior lucratividade para a empresa.

Considerando a interligação dos equipamentos de controle e os dispositivos de ponta, ou seja, sensores e atuadores, é possível definir duas topologias básicas de instalações.

Na primeira topologia, normalmente chamada de CLP central, cada sensor e atuador são ligados diretamente a um CLP. Nessa configuração, o CLP se mantém conectado à rede de comunicação industrial para enviar informações do processo ao nível superior, apontando dados produtivos e disponibilizando informações instantâneas nas IHMs da planta (figura 8.10). Desse modo, cada dispositivo é ligado ao CLP por um par de fios.

Em grandes instalações, isso pode representar enorme concentração de cabos e longas distâncias a serem percorridas pela fiação, tornando necessários proteções mecânicas e suportes. A manutenção corretiva pode ser um problema, dependendo da instalação que foi feita. Por ser um modelo relativamente simples de implementar para baixa densidade de pontos, é ainda muito utilizado em indústrias de pequeno e médio portes ou em plantas maiores, porém antigas.

A segunda topologia é a rede descentralizada de controle (figura 8.11), utilizada em processos mais complexos ou com necessidade de cobertura de grandes áreas e distâncias e alta densidade de pontos de monitoramento. Nesse tipo de planta, é comum encontrar barramentos de campo, que nada mais são do que redes de comunicação específicas para processos industriais. Por meio dos barramentos de campo, os CLPs podem conversar entre si e com os dispositivos que possuem entradas e saídas para atuar no processo.



**Figura 8.10**  
Modelo de ligação  
com CLP central.

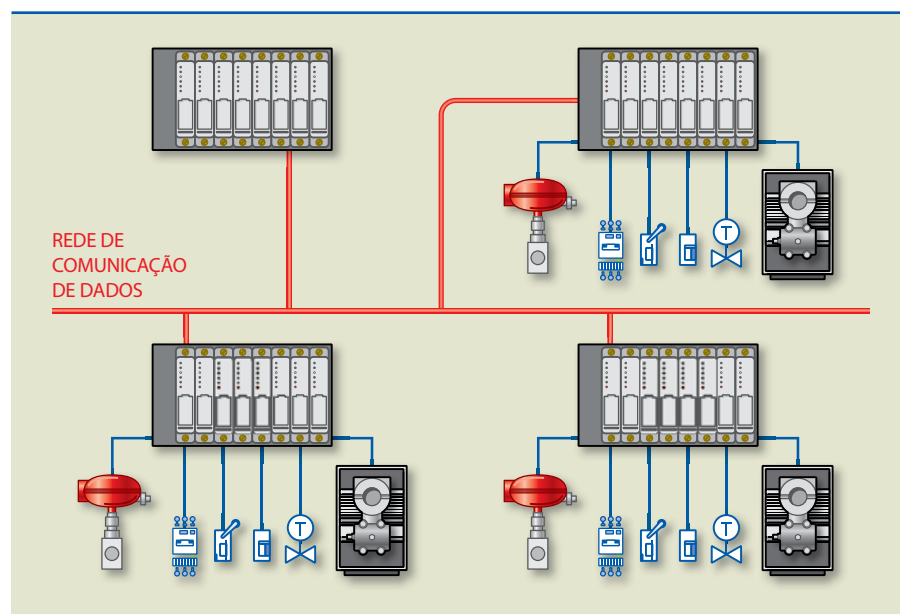
Também chamados de controladores de campo, blocos de entrada e saída ou simplesmente blocos de I/O (*input/output*), esses dispositivos podem coletar dados dos sensores ligados a eles e disponibilizar a informação em “pacotes” ao CLP. Tais pacotes, uma vez recebidos e decodificados pelo CLP, permitem que este saiba o que ocorreu na planta industrial, possibilitando a análise e atuação sobre a situação, seja por atuadores diretamente ligados a ele, seja por outros blocos de saída ligados ao barramento. Cada CLP pode tornar-se responsável por uma parte do processo e enviar aos demais apenas informações de interesse comum. Essa topologia de vários CLPs interligados será abordada na seção 8.4.

A característica de aquisição descentralizada por barramentos de comunicação, que agora passamos a chamar de redes industriais, reduz significativamente a quantidade de cabos entre as localidades da planta (figura 8.11). Dessa maneira, com o emprego de tecnologia mais avançada e de maior custo, é possível simplificar a instalação, além de facilitar a manutenção e o remanejamento do processo.

Diversos são os fabricantes e as tecnologias utilizadas na comunicação no chão de fábrica. Normalmente, a tecnologia física está atrelada ao protocolo de comunicação que vai ser usado na rede industrial, assunto abordado a seguir.



**Figura 8.11**  
Modelo de ligação  
descentralizada.



## 8.2 Protocolos de comunicação

Diversos são os protocolos de comunicação utilizados na indústria. Pode-se dizer que um protocolo de comunicação é um idioma no qual dois ou mais equipamentos conversam. Fazendo uma analogia para melhor entendimento, uma pessoa que fala apenas em português não consegue conversar com outra que fale em russo. O mesmo ocorre com CLPs que trabalham com protocolos de comunicação diferentes.

O protocolo de comunicação é um conjunto de regras definidas em que consta o formato no qual a mensagem deve ser transmitida entre os participantes da rede. A simbologia e os caracteres usados para questionar sobre certo evento ou responder a determinada questão e a velocidade com que essa informação flui de um lado a outro, por exemplo, são detalhes que devem ser padronizados em um protocolo.

Os protocolos necessitam de um meio físico que permita a interligação desses dispositivos na velocidade e condições exigidas pelo fabricante, possibilitando a construção de uma rede de comunicação. Tais informações normalmente são transmitidas ou questionadas de forma serial, ou seja, bit a bit, por meio de sinal elétrico ou óptico. Em geral, protocolo e meio físico são ditados pelo fabricante.

No início dos anos 1980, surgiram diversas tecnologias não padronizadas de diferentes fabricantes, cada um deles definindo a sua. Na década seguinte, com o crescimento desse mercado, a padronização começou a ser considerada, pois os clientes queriam trabalhar com o melhor produto do fabricante A com o melhor produto do fabricante B, e ambos os fabricantes tinham protocolos distintos. Foi nesse período que surgiram as primeiras iniciativas de padronização.

É interessante citar os fatores de análise que costumam ser utilizados na escolha de um protocolo de comunicação industrial em novas instalações:

- Custo do *hardware*.
- Número de pontos de entrada e saída permitidos pela interface elétrica.
- Distância física máxima suportada pela interface elétrica.
- Tempo de resposta.
- Facilidade de parametrização.
- Meio físico: possui imunidade a ruído ou permite a utilização de interface óptica (para locais que não permitem sinais elétricos, como plantas à prova de explosão).

Alguns exemplos de fabricantes de equipamentos para automação industrial e os protocolos que normalmente disponibilizam em seus CLPs:

- **Rockwell** – protocolos Devicenet; Controlnet; Ethernet/IP.
- **Siemens** – protocolos AS-I; Profibus DP, PA, FMS; Profinet.
- **Schneider** – protocolos Modbus; Modbus Plus; Modbus TCP.
- **Outros fabricantes** – protocolos Interbus; CAN; SDS; DNP3; ARCnet.

Outras grandes empresas de automação industrial, como Honeywell, ABB, Emerson Electric, Invensys, Yokogawa, Mitsubishi, GE e Omron, também empregam a maioria dos protocolos de comunicação citados.

A seguir, são apresentadas as propriedades dos principais protocolos usados atualmente.

### 8.2.1 Protocolo Modbus

Criado em 1979 para a comunicação entre controladores da Modicon, é de simples configuração e manutenção. Por ser um protocolo aberto, desde o início teve fácil penetração no mercado e continua sendo bem-aceito, por sua simplicidade e certa robustez na troca de dados entre os dispositivos. Três versões são mais comuns: o Modbus RTU, o Modbus Plus e o Modbus TCP/IP.

O Modbus RTU (*Remote Terminal Unit*, ou Unidade Terminal Remota) traz o conceito de visualização remota de dados. Normalmente, é utilizado para a comunicação dos CLPs com os blocos de E/S e com os instrumentos eletrônicos inteligentes, como relés de proteção, controladores de processo, atuadores de válvulas, transdutores de energia e outros. O meio físico usado é o serial RS-232 ou RS-485 (padrão serial elétrico que trabalha com comunicação diferencial entre dois fios), podendo também ser empregado com conversores seriais – fibra óptica em uma velocidade de até 230 kbps (quilobits por segundo).

Muitas vezes, o RTU é utilizado em CLPs de pequeno porte, fornecendo informações do processo a *softwares* de supervisão em salas de controle, ou mesmo em IHMs locais que possuem o protocolo Modbus também nativo. Esse protocolo permite que um CLP mestre interogue até 32 blocos de E/S, considerados dispositivos escravos.

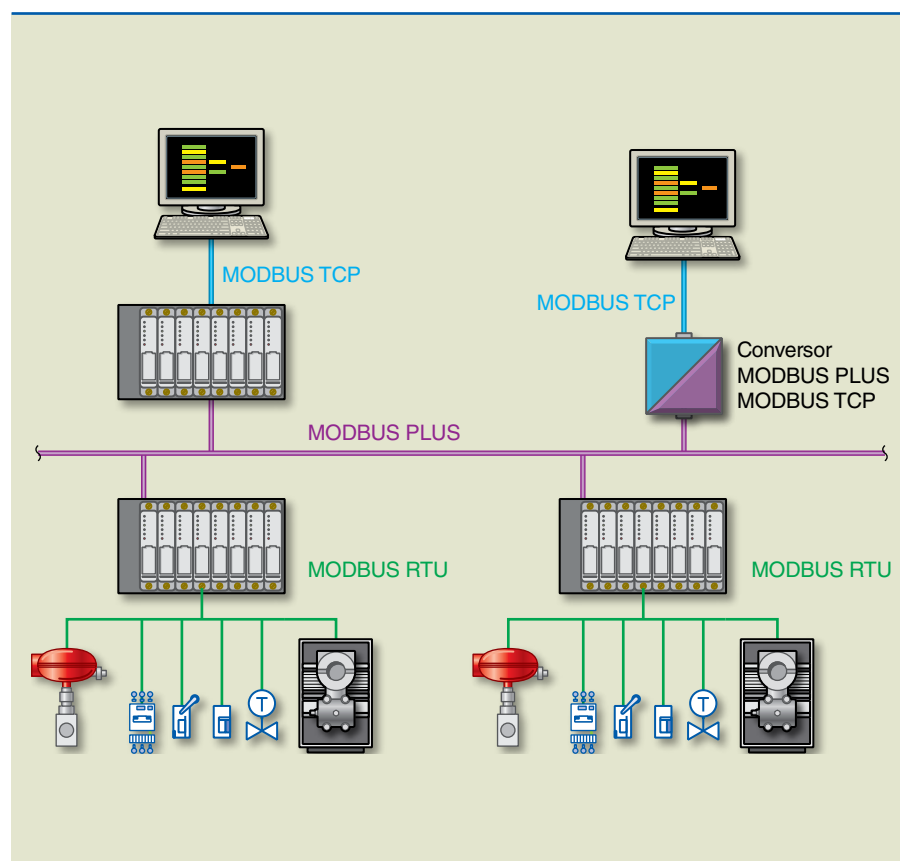


O Modbus Plus é uma versão do protocolo RTU melhorada em alguns pontos, possibilitando que CLPs que trabalham em Modbus conversem entre si durante a execução do processo. Também pode trabalhar com blocos de E/S e outros dispositivos de ponta, como relés de proteção e diversos controladores. O meio físico é o serial RS-485 com taxas de transmissão de até 1 Mbps (megabits por segundo).

O Modbus TCP/IP é utilizado para comunicação entre sistemas de supervisão, CLPs e alguns modelos de blocos de E/S. O protocolo Modbus é encapsulado no protocolo TCP/IP e transmitido por redes Ethernet. Trata-se de uma adaptação do protocolo Modbus RTU utilizando interface Ethernet, em vez das tradicionais interfaces seriais. Com isso, os limites de distância estão diretamente relacionados ao alcance das interfaces Ethernet.

A topologia usada em redes Modbus pode ser observada na figura 8.12.

**Figura 8.12**  
Topologia utilizada em redes Modbus.

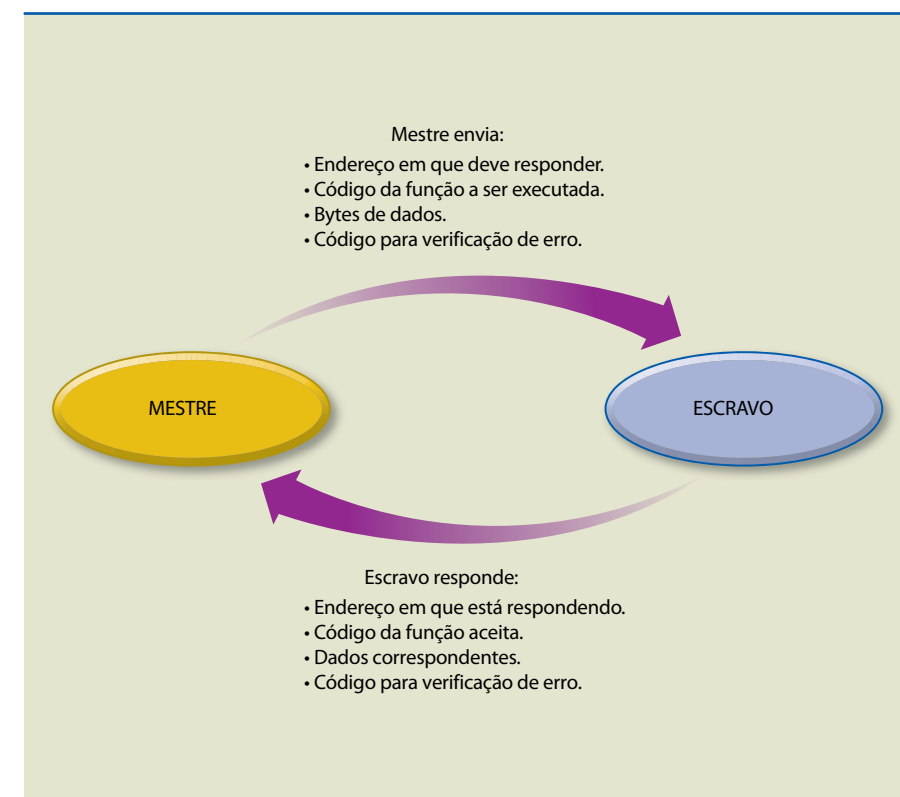


Durante a comunicação em uma rede Modbus, o protocolo determina como o dispositivo mestre deve questionar o dispositivo escravo, ou seja, instruções para verificar se o escravo está ativo ou não, se o escravo possui valores discretos (digitais) ou analógicos e como decodificar a informação recebida. Em resumo, a programação do mestre corresponde a dizer quais blocos ele vai ler, em quais endereços e qual conteúdo deve esperar como resposta. Isso tem de ocorrer de maneira coordenada, pois os blocos ou dispositivos escravos, por sua vez, devem

estar parametrizados com os valores de endereço esperados, ser realmente dispositivos discretos ou analógicos e estar preparados para responder de acordo com a mensagem formatada pelo protocolo, conforme previamente programado no dispositivo mestre Modbus.

O dispositivo mestre dessa rede pode enviar mensagens para um escravo específico ou uma única mensagem para que todos os dispositivos da rede leiam e entendam o que foi pedido. Esse tipo de envio em massa a todos os escravos da rede normalmente contempla ações que devem ser sincronizadas no tempo. Um exemplo é a leitura de temperatura em 64 pontos analógicos distintos (8 blocos de E/S com 8 entradas de temperatura cada um). No instante em que o escravo recebe a informação de ler e armazenar temporariamente esse valor, ele o fará para posterior consulta do mestre e então consolidará toda a informação recolhida ao longo de alguns ciclos de varredura. A comunicação entre um mestre e um escravo obedece a algo semelhante ao fluxo apresentado na figura 8.13.

**Figura 8.13**  
Fluxo de comunicação entre mestre e escravo no Modbus.



Na transmissão do mestre, os bytes de dados têm informações para o escravo, incluindo qual registrador inicial deve responder e a quantidade de registros que têm de ser lidos, ao passo que o campo de verificação de erro permite ao escravo validar os dados que recebeu do mestre. Na resposta do escravo, o código de função é repetido de volta para o mestre, juntamente com os bytes de dados que possuem a informação solicitada ao escravo. Se alguma falha na aquisição do dado ocorrer, o código da função é modificado para indicar esse erro e os bytes de dados conterão um código que descreverá o erro. A verificação de erro permite ao mestre validar os dados recebidos.



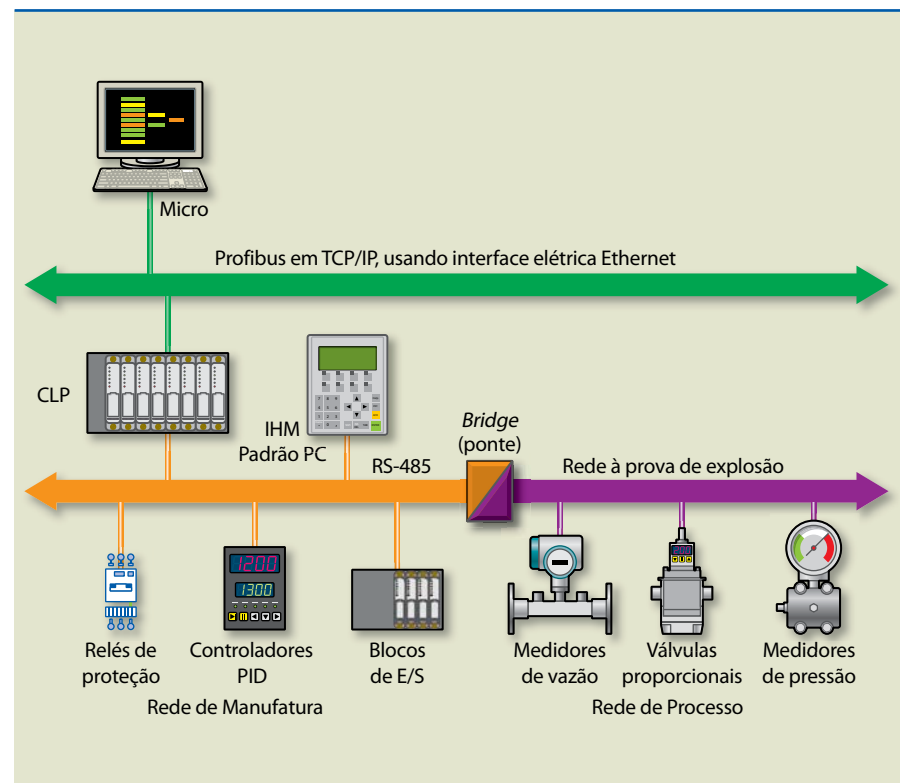


## 8.2.2 Protocolo Profibus

O Profibus foi desenvolvido em 1987 na Alemanha por uma associação de 21 companhias, a fim de criar um barramento de campo. Em 1995, foi fundada a Profibus International, englobando 22 entidades espalhadas pelo mundo, com o objetivo de promover esse protocolo de comunicação aberto.

O Profibus possibilita interligar não só a parte de controle da planta por meio do gerenciamento dos blocos de E/S e outros CLPs, como também dispositivos de outros controles externos ao processo, mas pertinentes à planta industrial. A figura 8.14 dá uma ideia de como e onde é utilizado o protocolo Profibus na comunicação entre dispositivos de uma planta industrial.

**Figura 8.14**  
Interligação de dispositivos utilizando o protocolo Profibus.



O protocolo Profibus pode ser usado em níveis superiores de acesso, na parte de administração da informação de chão de fábrica, por meio do Profibus Ethernet, também conhecido como Profinet; em níveis inferiores, gerenciando ou controlando dispositivos de processo, como relés de proteção, controladores PID, blocos de E/S, válvulas, medidores e outros; e em conversores eletro-ópticos, empregando fibras ópticas como meio de transmissão e recepção de dados em áreas classificadas, ou seja, que tenham risco de explosão.

O protocolo subdivide os elementos em dispositivos de manufatura e de processo, de acordo com o tipo de dados que serão enviados e recebidos por meio das interfaces.

Fornecedores de *hardware* também disponibilizam conversores, conhecidos como *gateways* (portas de acesso), que servem de dispositivos de interligação entre os padrões a serem utilizados, por exemplo, o conversor Ethernet para RS-485 e conversores RS-485 para fibra óptica. Esses *gateways* decodificam a informação de um meio físico e a transmitem em outro, mantendo a rede sempre atualizada e os caminhos alternativos disponíveis para alguns fluxos de dados que o processo exige.

Assim como outros protocolos, o Profibus é uma família, com pequenas variações entre os diversos tipos. Destacam-se:

- **Profinet (Profibus for Ethernet)** – Utilizado na comunicação entre CLPs e PCs usando Ethernet-TCP/IP por meio de uma interface Ethernet.
- **Profibus FMS (Fieldbus Message Specification, ou Especificação de Mensagem no Barramento de Dados)** – Também empregado em comunicação entre CLPs e PCs, porém, por meio de interfaces seriais dedicadas de alta velocidade.
- **Profibus DP (Decentralized Peripherals, ou Periféricos Descentralizados)** – Usado na comunicação com *drivers* de motores, blocos de E/S, transdutores, analisadores, controladores, entre outros.
- **Profibus PA (Process Automation, ou Automação de Processos)** – Utilizado na comunicação com transmissores de pressão, nível, vazão, temperatura e válvulas de controle. Usa como meio físico as interfaces elétricas com nível de potência adequado para permitir sua instalação em áreas classificadas.
- **AS-I (Actuator Sensor Interface, ou Interface Sensor/Atuador)** – Empregado na comunicação com sensores e atuadores discretos (*on/off*). Utiliza um *gateway* entre o Profibus DP e o AS-I para acesso a esses atuadores.

A seguir, são apresentados detalhes das interfaces de comunicação físicas usadas no Profibus.

### Padrão RS-485

- Cabo par trançado com malha.
- Taxas de 9 600 bps (distância de 1 200 metros) a 12 Mbps (distância de 100 metros).
- Topologia física em barramento com terminadores.
- Até 32 dispositivos.
- Com um repetidor, permite mais 32 dispositivos.
- Com o uso máximo de quatro repetidores, chega a 126 dispositivos em uma mesma rede.
- Conectores de interligação tipo DB9 para padrão IP20.
- Conectores circulares específicos para IP65 ou IP66.

### Padrão MBP (Manchester Coding Bus Powering, ou Barramento de Código Manchester)

A codificação Manchester é uma das formas de garantir a integridade de um sinal transmitido em uma rede de comunicação.



- Cabo par trançado com malha.
- Transmissão síncrona com taxa de 31,25 kbps.
- Topologia física em barramento e/ou árvore.
- Segmentos com até 32 dispositivos.
- Com um repetidor, permite mais 32 dispositivos.
- Com o uso máximo de quatro repetidores, chega a 126 dispositivos em uma mesma rede.
- Intrinsecamente seguro.

### Fibra óptica

É utilizada em ambientes com muito ruído elétrico, com necessidade de vencer grandes distâncias ou ainda para redundância. Entre suas características estão ser imune à interferência eletromagnética e permitir o isolamento entre duas redes distintas, ou seja, se um lado da rede interligada por fibra óptica receber uma descarga elétrica e se danificar, a rede do outro extremo da comunicação óptica não será danificada. Os conversores de fibra decodificam o sinal RS-485 em óptico.

### 8.2.3 Protocolo CAN

O protocolo CAN (*Controller-Area Network*, ou Rede de Controladores de Área) foi desenvolvido pela Bosch na década de 1980, para a interligação dos sistemas de controle eletrônicos dos automóveis. Originalmente usado para a troca de dados em veículos motorizados, exigindo extrema velocidade, nos anos 1990 passou a ser utilizado também na comunicação de dispositivos industriais, em barramentos internos de máquinas e equipamentos mais complexos e em dispositivos que demandavam a velocidade que a interface possibilitava, além de ter várias utilizações em IHMs.

Esse protocolo permite o acesso a múltiplos mestres, ou seja, todos os dispositivos da rede podem transmitir solicitações de consulta ou respostas a qualquer outro dispositivo. Dessa maneira, não existe um endereço específico para cada dispositivo, ou seja, o transmissor envia a mensagem para todos os nós simultaneamente.

No entanto, cada mensagem contém certo número de bits identificadores usados para definir a prioridade no acesso ao meio. Tal liberdade de tráfego pode gerar colisões e, nesse caso, as regras de prioridade devem ser definidas pelo programador. Existe um bit específico na transmissão que define o nível de prioridade que o pacote tem no tráfego. Se um pacote de informação possui prioridade 5 e outro, prioridade 2, em uma ordem crescente o pacote com prioridade 2 seria transmitido, enquanto o de prioridade 5 teria de aguardar o término dessa transmissão. Desse modo, identificadores com menor valor binário são de prioridade mais alta. A arquitetura multimestre tem efeito positivo na disponibilidade da rede, pois a falha de um controlador não leva ao colapso do barramento.

Outros protocolos que utilizam a especificação do CAN:

- Devicenet, ou Rede de Dispositivos.
- *Smart Distributed System* (SDS), ou Sistema Inteligente Distribuído.

- CAN Open.
- CAN Kingdom.
- SAE J1939.

As especificações do protocolo CAN detalham partes da camada física e da camada de enlace dos dados a serem transmitidos entre os controladores. A parte física, como conectores e cabos, é aberta aos usuários dessa tecnologia. A principal utilização do CAN em aplicações industriais é em processos cíclicos e sincronizados, ou seja, uma vez definida a plataforma CAN de comunicação, cabe aos usuários e programadores definir qual sequência e quais prioridades devem existir entre esses dispositivos.

Os protocolos Devicenet, SDS, CAN Open, SAE J1939 e CAN Kingdom, mesmo baseados no CAN, possuem características bem definidas no que se refere a especificações de cabeamento e conexão dos dispositivos que compõem essa rede.

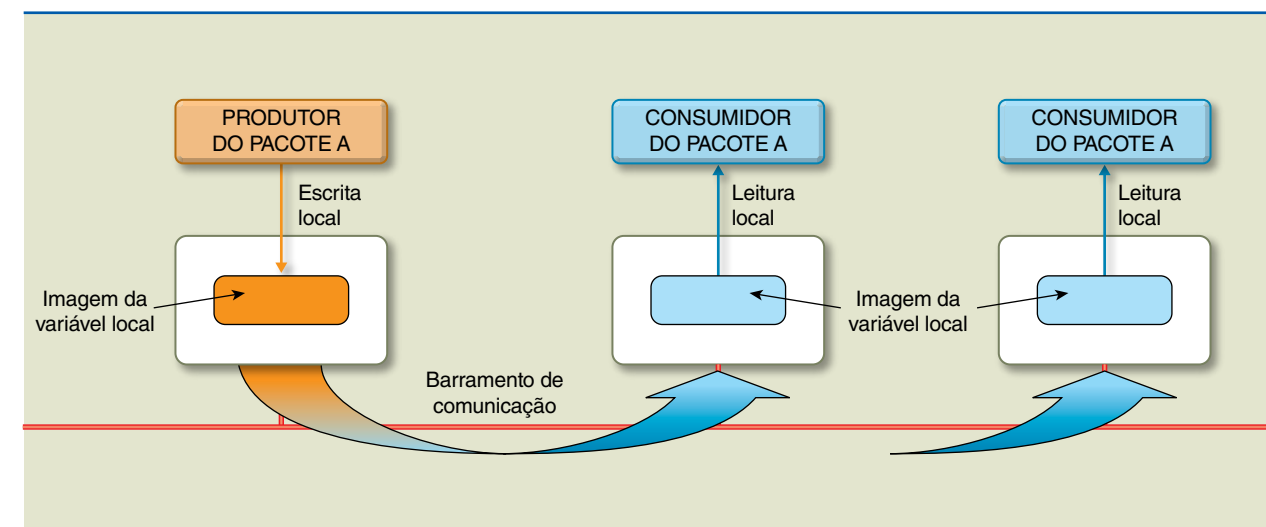
As principais características das mensagens no protocolo CAN são:

- Segmentação para transmissão de mais bytes.
- Identificador de 11 bits com 2048 tipos diferentes de mensagens.
- Taxas de 1 Mbps até 40 metros e 50 kbps em 1 km.
- Barramento com até 64 nós.
- Uso da tecnologia do *chip* CAN.
- Identificador de dados (0 a 8 bytes).

A rede CAN trabalha no modelo produtor-consumidor, que, em resumo, trata a questão da comunicação com a seguinte ideia: se alguém precisa de uma informação, é necessário providenciá-la; de outro lado, se uma informação é disponibilizada na rede, alguém vai precisar dela. Esse raciocínio define que não existe um único mestre ou uma forma de comunicação predefinidos. A figura 8.15 mostra um caso em que um pacote de dados (produto) é enviado à rede e um ou mais dispositivos o utilizam para processar ou interagir localmente no processo (consomem o pacote).

**Figura 8.15**

Geração e consumo de produtos na rede CAN.

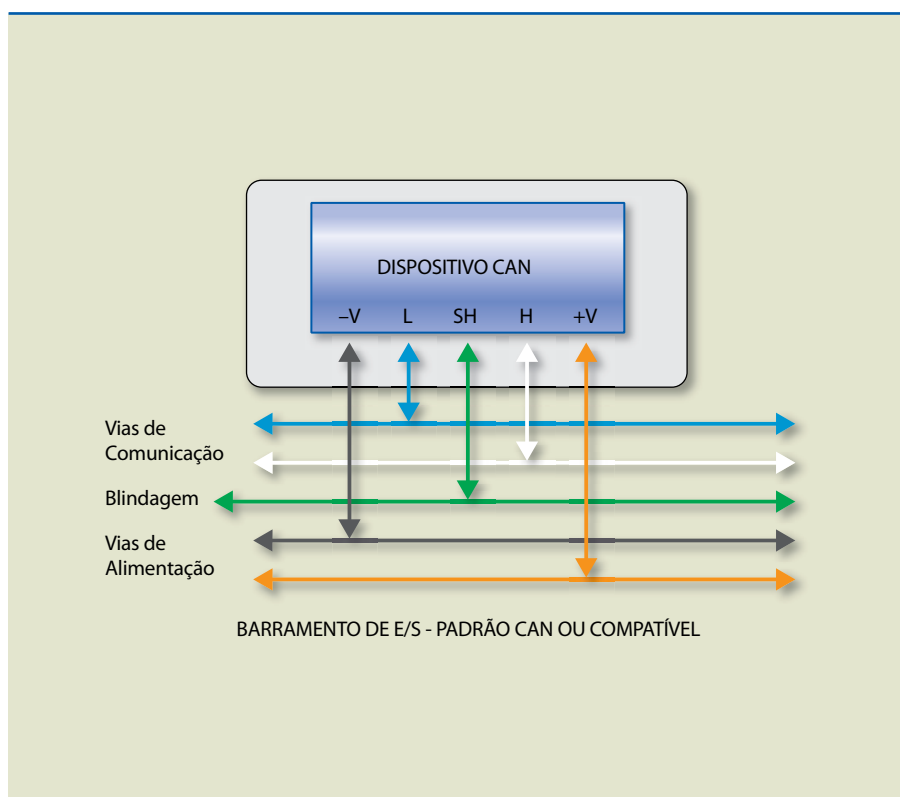


### 8.2.4 Protocolo Devicenet

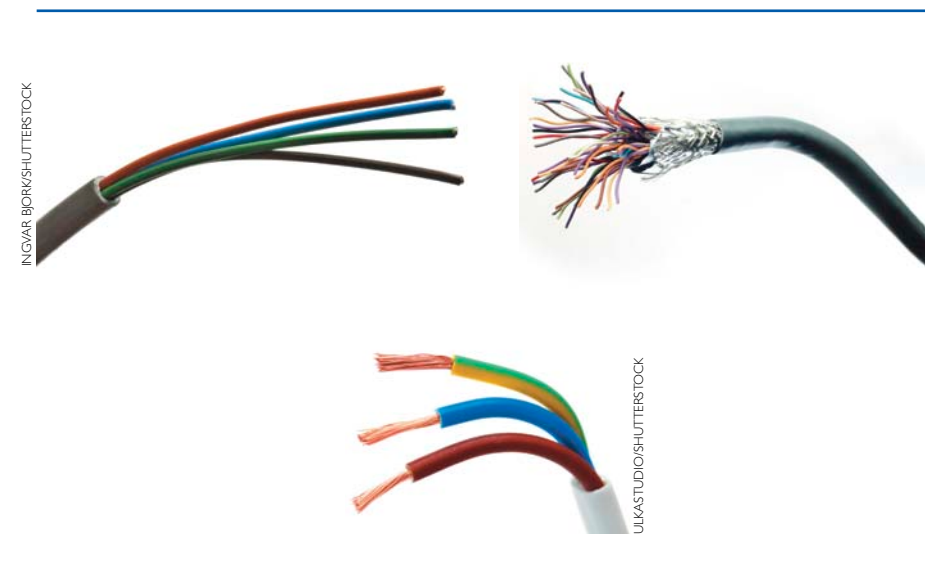
Desenvolvido pela Allen-Bradley, empresa do grupo Rockwell Automation e lançado em 1994, o Devicenet, ou Rede de Dispositivos, oferece uma arquitetura integrada no formato produtor-consumidor, pois baseia-se no protocolo CAN. Por se tratar de um protocolo aberto, outros fabricantes também utilizam esse padrão com o intuito de compatibilizar seus produtos a essa rede muito difundida nas indústrias.

Consiste em uma arquitetura de rede que permite a leitura, a atuação e a configuração de dispositivos ligados a ela (figura 8.16). Esse protocolo é utilizado na interligação de CLPs e IHMs e na comunicação com atuadores *on/off*, inversores de frequência e chaves de partida eletrônica.

**Figura 8.16**  
Barramento de comunicação Devicenet, compatível com CAN.

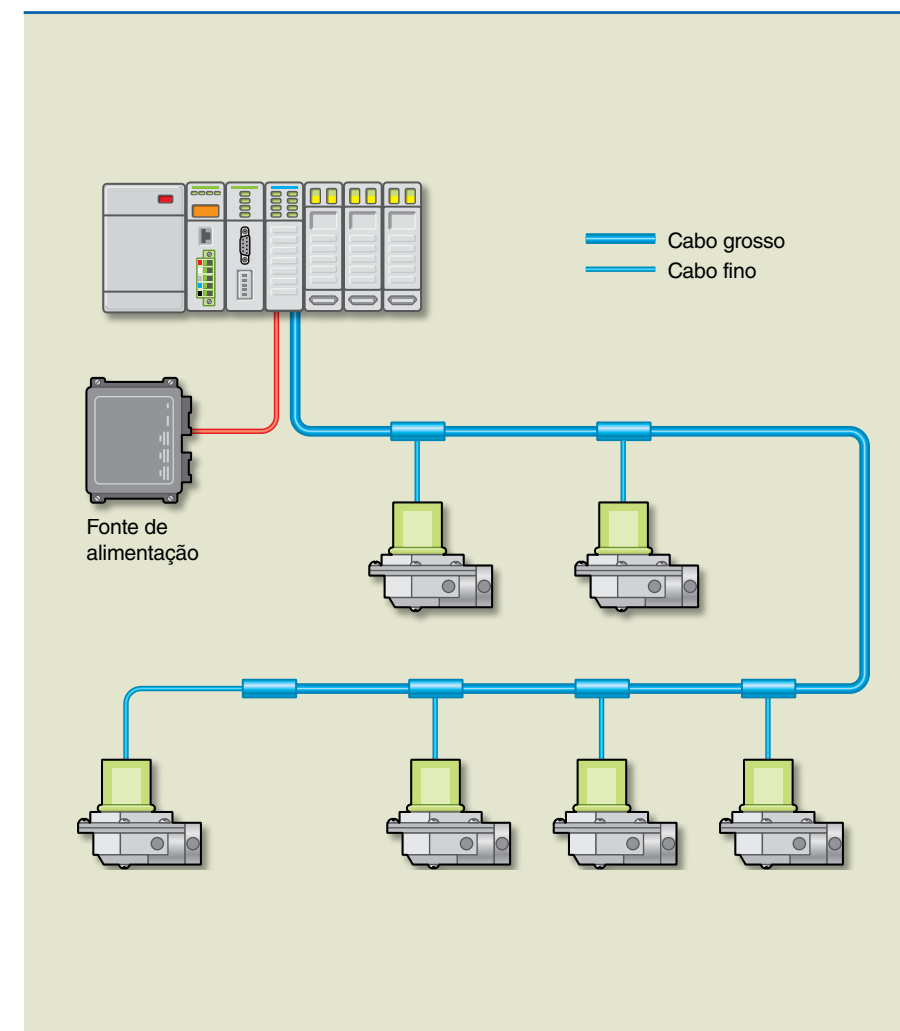


A rede Devicenet utiliza dois pares de fios em seu barramento, um deles para comunicação e o outro para alimentação dos dispositivos. Usa corrente contínua na alimentação, e os cabos possuem blindagem externa aos pares com fita de alumínio e uma segunda blindagem geral por malha trançada com fio de dreno. Existe uma padronização nas cores dos fios: o par de alimentação é vermelho (V+) e preto (V-), e o par de comunicação, branco para o sinal chamado de CAN High e azul para o CAN Low. Com diversos modelos de cabos para diferentes aplicações na indústria, os cabos são conhecidos como tronco ou grosso (diâmetro de 12,5 mm), fino (diâmetro de 7 mm) e *flat*, cada qual com aplicações específicas de utilização. Os cabos podem ser visualizados na figura 8.17.



**Figura 8.17**  
Modelos de cabos para Devicenet.

A topologia de distribuição na figura 8.18 exemplifica como os dispositivos são instalados em uma rede Devicenet na indústria.



**Figura 8.18**  
Exemplo de rede Devicenet e especificação dos cabos.



Outras características da rede Devicenet são:

- Topologia física em barramento.
- Linhas tronco e derivações com resistores de terminação de  $121 \Omega$ . Têm a função de evitar a reflexão dos sinais e devem ser ligadas entre os terminais de comunicação (par azul e branco).
- Conexão de até 64 elementos.
- Inserção e remoção de nós a quente, ou seja, sem a necessidade de desconectar a alimentação da rede.
- Taxas de transmissão de dados da ordem de 125 a 500 kbps.

Existe uma relação entre o comprimento de cabo e a velocidade de transmissão de dados que deve ser observada em novas instalações ou mesmo na manutenção e substituição de cabos. A tabela 8.1 apresenta esses dados para consulta.

**Tabela 8.1**

Relação tipo de cabo *versus* taxa de transmissão e velocidade de barramento

Tipo de cabo	Função do cabo	Taxa de transmissão		
		125 kbps	250 kbps	500 kbps
Cabo grosso	Tronco	500 m	250 m	100 m
Cabo fino	Tronco	–	100 m	–
Cabo <i>flat</i>	Tronco	380 m	200 m	75 m
Cabo fino	Derivação individual	–	6 m	–
Cabo fino	Soma de derivações	156 m	78 m	39 m

A configuração dos dispositivos, também chamados de nós, é baseada em folhas de dados eletrônicos (EDS, *Electronic Data Sheets*). Fornecidas pelos fabricantes dos dispositivos Devicenet, essas folhas contêm descrições eletrônicas de como o dispositivo opera e quais os dados e formatos que lhe é permitido trocar. O Devicenet também possui uma versão Ethernet, o Controlnet, que possibilita a comunicação de dados com dispositivos hierarquicamente superiores à rede de controle.

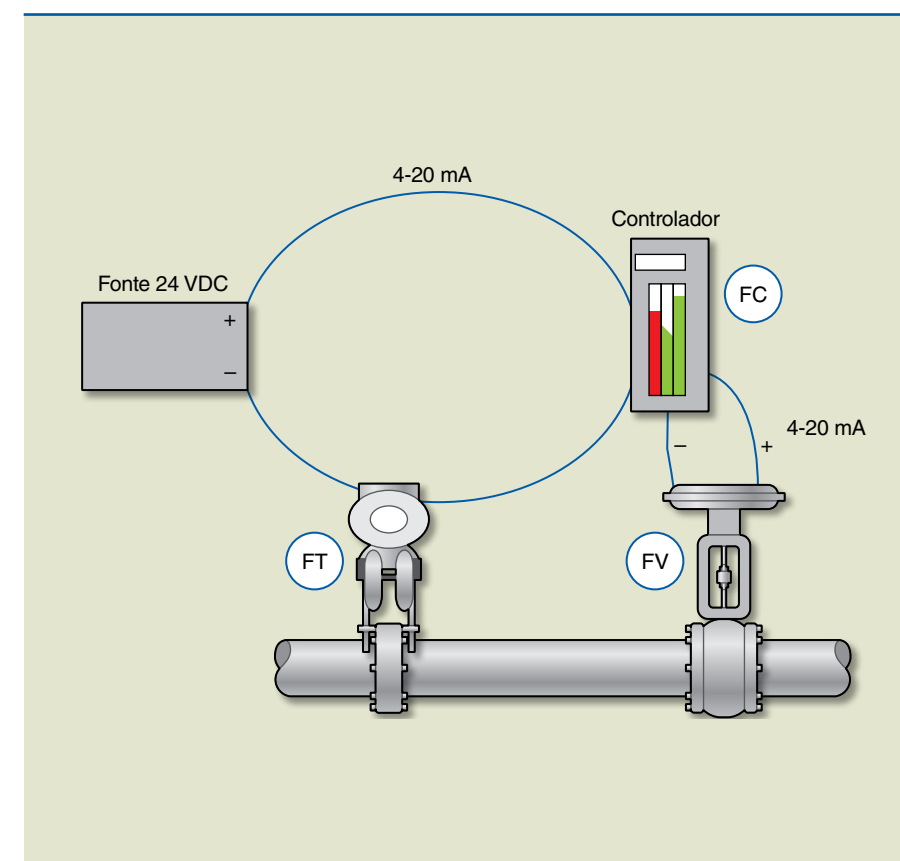
### 8.2.5 Protocolo Hart

O protocolo de comunicação Hart (*Highway Addressable Remote Transducer*, ou, em tradução não literal, Larga Comunicação para Transdutores Remotos Endereçáveis) é um padrão industrial para comunicação com instrumentos de processo “inteligentes”. Criado pela Rosemount na década de 1980 e mantido pela Hart Communication Foundation desde 1993, esse protocolo permite comunicação digital simultânea com o sinal analógico de corrente de 4 a 20 mA, ou seja, a grandeza informada digitalmente por um sinal modulado em FSK (*frequency shift keying*, ou modulação por chaveamento de frequência) é proporcional à variação de corrente na malha, de 4 a 20 mA, utilizando o mesmo par de cabos.

É importante destacar algumas características do Hart:

- Possibilita o acesso remoto a todos os parâmetros e diagnósticos do instrumento.
- Permite o suporte a instrumentos multivariáveis, ou seja, instrumentos que trabalham com mais de uma grandeza no mesmo dispositivo.
- Disponibiliza informações do estado do dispositivo durante o monitoramento deste.
- É compatível com a fiação existente para 4-20 mA.
- Utiliza um padrão aberto de comunicação.
- Possibilita que instrumentos de campo sejam calibrados e aferidos simultaneamente (medidores de pressão, vazão, temperatura, nível, entre outros).

O padrão de comunicação 4-20 mA é largamente utilizado como forma de transmitir uma informação do processo para um elemento de controle ou registro. Apenas uma informação pode ser transmitida, ou seja, tem a característica de uma rede ponto a ponto.



**Figura 8.19**

Sistema de vazão: exemplo de ligação do protocolo Hart.

O sistema de controle de vazão da figura 8.19 exemplifica o tipo de ligação entre o dispositivo de medição e o controlador nesse protocolo. Esse sistema é composto por um transmissor multivariável de vazão (FT), um controlador e indicador de vazão (FC) e uma válvula de controle (FV); todos eles utilizam o padrão 4-20 mA para a transmissão da informação.



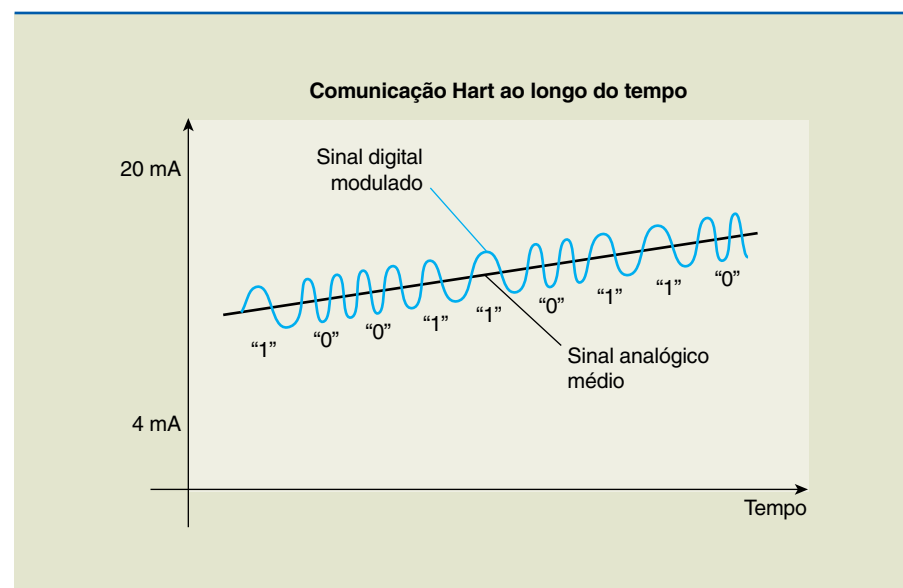
A vazão calculada dentro do transmissor multivariável é enviada ao controlador por um sinal de 4-20 mA. O controlador tem por função comparar o valor recebido e atuar na válvula de abertura de acordo com a vazão desejada. O controle da válvula também é feito enviando um sinal de 4-20 mA. Mesmo que o transmissor seja capaz de trabalhar com quatro grandezas físicas predefinidas pelo instalador do dispositivo, apenas uma delas pode ser transmitida pelo sinal de 4-20 mA.

Detalhando um pouco mais a comunicação feita por meio do protocolo Hart, apresenta-se a figura 8.20, na qual podemos observar como o sinal digital é incorporado ao sinal analógico. Ao sinal de corrente entre 4 e 20 mA é adicionado um sinal modulado em frequência, com amplitude máxima predefinida, em que o valor médio dessa corrente não interfere na proporcionalidade do sinal de 4-20 mA. A comunicação é feita em uma taxa de 1 200 kbps.

A modulação é feita com sinais senoidais, com amplitudes de corrente da ordem de 0,5 mA de pico. Considerando que o valor médio de uma senoide é zero, verifica-se que a modulação aplicada ao sinal de corrente não tem influência no valor real transmitido.

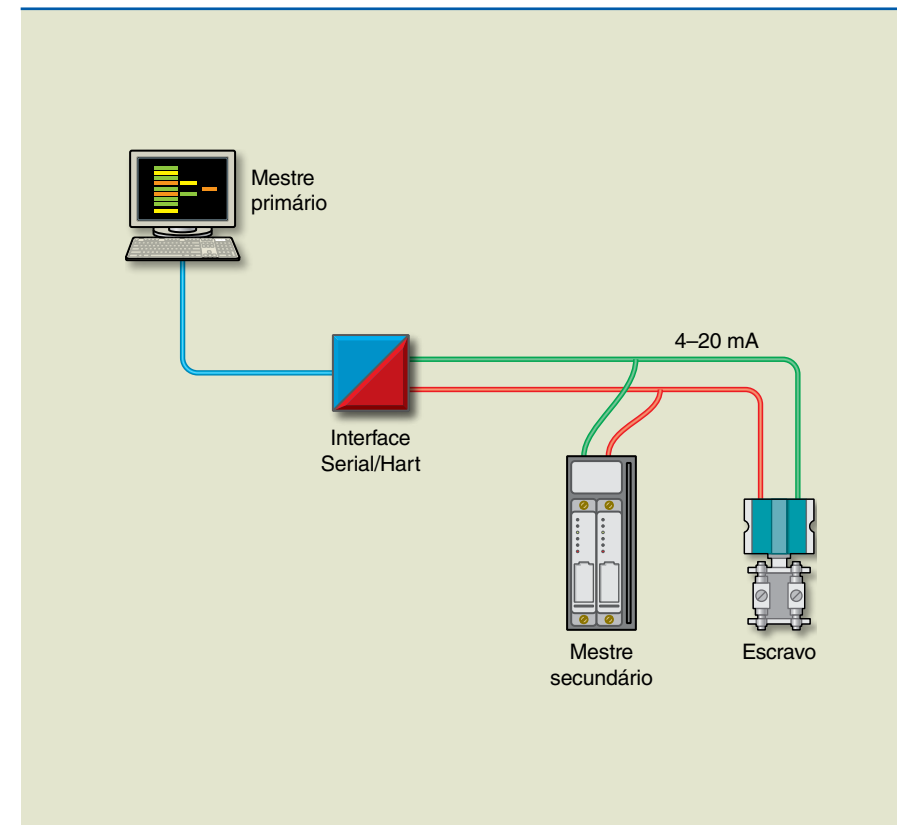
**Figura 8.20**

Modulação aplicada ao sinal de corrente no protocolo Hart.



Em aplicações práticas, o Hart é usado em comunicações mestre-escravo, ou seja, um dispositivo questiona e o outro responde ao questionamento. O protocolo também permite a utilização de mais de um mestre, porém, o segundo só assume o controle da comunicação quando percebe que o principal não está operando.

Nessa topologia, em geral, o mestre primário é o computador de supervisão e o secundário, um terminal de calibração e programação que pode ser ligado ao próprio barramento de 4 a 20 mA. Obviamente, o computador deve possuir uma interface que permita a leitura do sinal analógico da rede. A figura 8.21 apresenta uma topologia de rede multimestre conforme descrito.



**Figura 8.21**

Topologia de ligação multimestre no protocolo Hart.

O dispositivo mestre possui a capacidade de provocar variações de corrente na linha, modulando o sinal; o escravo faz a parte da demodulação, transformando a corrente em tensão e aplicando essa variação proporcional no dispositivo que está controlando. Durante a resposta, esse processo se inverte: o dispositivo escravo da rede é que define o valor de corrente, que, por sua vez, é interpretado pelo mestre, que vai analisar, comparar e atuar em algum ponto do sistema quando necessário.

A comunicação em Hart é definida com os seguintes parâmetros:

- 1 bit de início e 8 bits de dados.
- 1 bit de paridade ímpar e 1 bit de parada.
- 40 bits para endereços universais (redes de maior porte) ou 8 bits para endereços curtos (redes de menor porte).
- 6 bits para o código de identificação do fabricante.
- 8 bits para o código do tipo de dispositivo.
- 24 bits para o código identificador do dispositivo.
- 4 bits para endereços de 1 a 15.

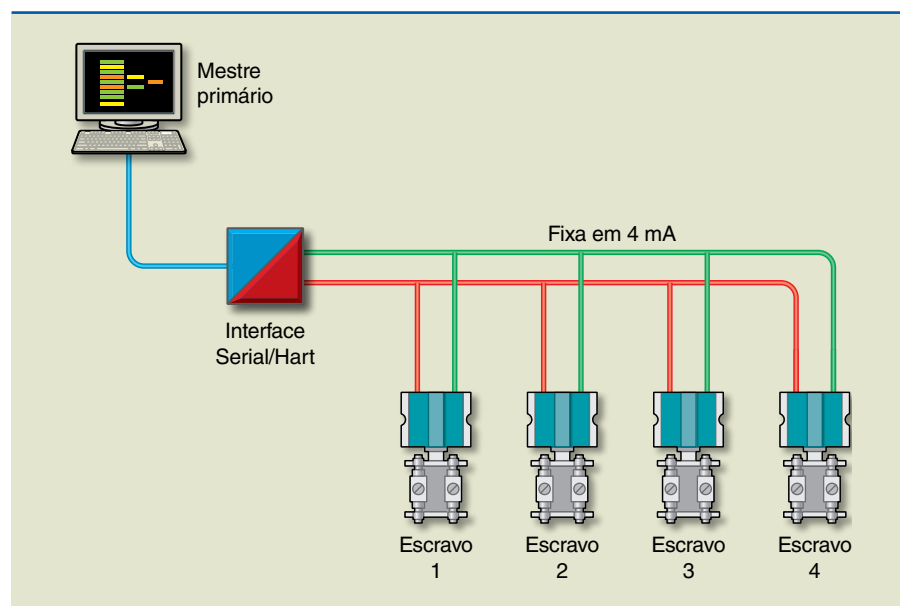
A rede Hart permite também a comunicação somente em sinais digitais (modulados em FSK). Nessa situação, a corrente utilizada é fixa em 4 mA e comporta até 15 dispositivos interligados. É possível interligar dispositivos com até 3 000 metros de distância utilizando um par de cabos, com tempo de leitura médio de 0,5 segundo por dispositivo.



As figuras 8.22 e 8.23 apresentam exemplos clássicos da utilização desse protocolo em automação de sistemas. A figura 8.22 ilustra um exemplo de mestre-escravos utilizando mais de um controle Hart atrelados a um computador que efetua a supervisão do sistema. Na figura 8.23, aparece um sistema multiplexado em que várias leituras ponto a ponto são possíveis graças ao chaveamento sincronizado dos pontos a serem monitorados.

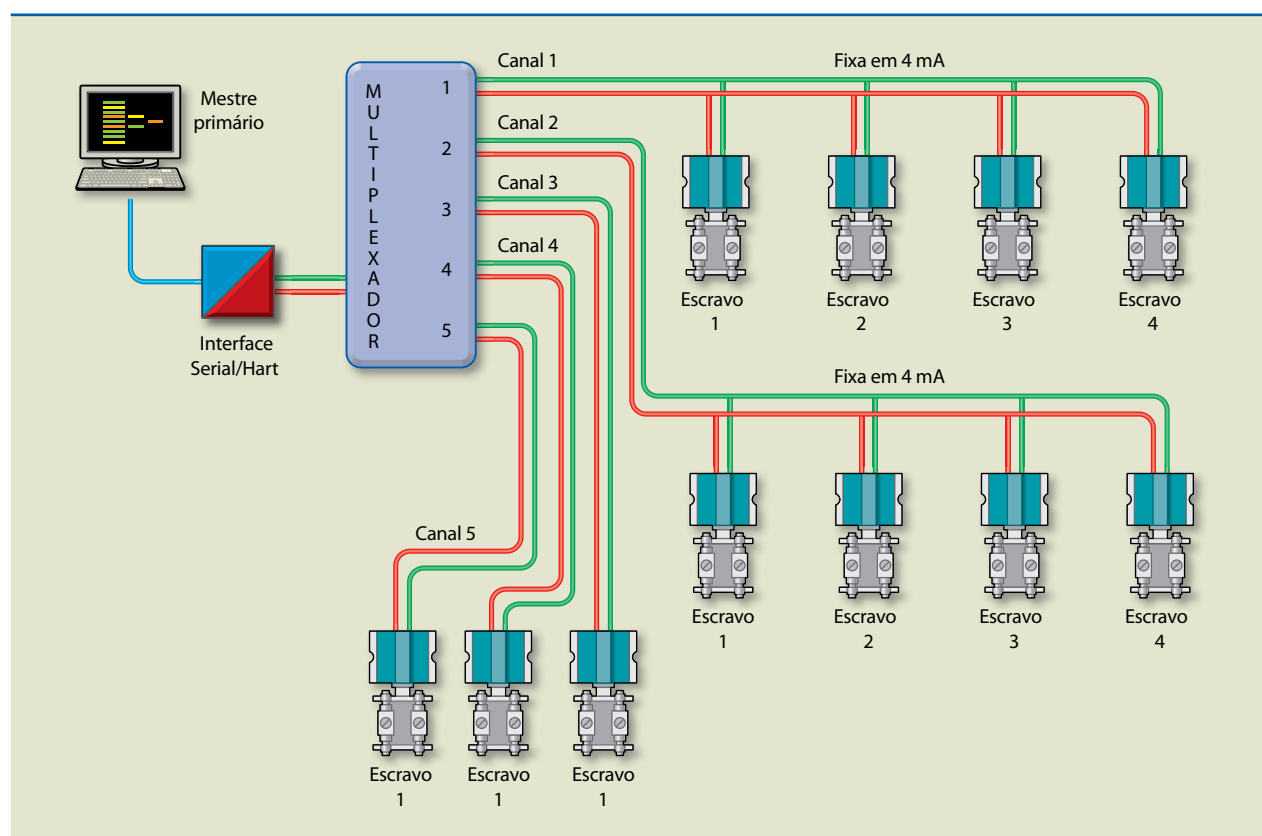
**Figura 8.22**

Mestre e escravos na topologia Hart.



**Figura 8.23**

Sistema multiplexado utilizando Hart.



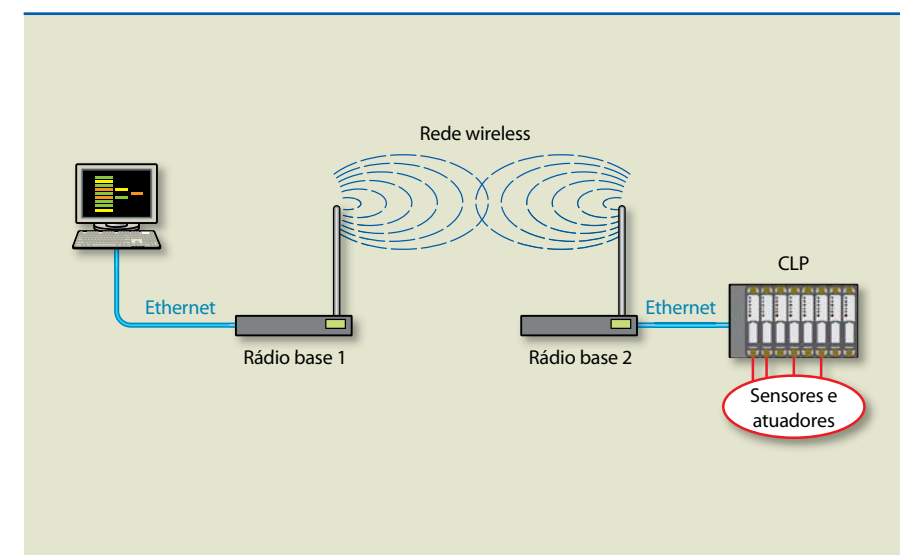
Por fim, vale destacar a existência de transmissores compatíveis com o protocolo Hart que têm por função também o controle de PID (proporcional, integral e derivativo). Com eles, é possível montar redes independentes, apenas com o controlador e a válvula de controle.

### 8.3 Wireless na automação

A comunicação *wireless* (sem fio) é caracterizada por um sistema de radiofrequência, comumente utilizada em 900 MHz ou 2,4 GHz, que estabelece *links* entre dois ou mais pontos. Esse sistema de rádio recebe o sinal digital de uma interface elétrica, decodifica e condiciona o sinal em ondas de rádio e o transmite a um segundo rádio. O segundo rádio capta o sinal transmitido e o codifica e condiciona a uma interface elétrica que deve estar ligada a um dispositivo destinatário. Vale ressaltar que essa comunicação é bidirecional, ou seja, o rádio que transmite o pacote de dados de perguntas também deve receber um pacote de dados de respostas, assim como o rádio que recebe o pacote de dados de perguntas tem de transmitir outro pacote de dados de respostas.

Sistemas de comunicação *wireless* trabalham com interfaces-padrão, normalmente do tipo serial ou Ethernet. Como a grande maioria dos protocolos apresentados pode trabalhar com essas interfaces, criou-se a cultura de utilização desse sistema de transmissão e recepção de dados também em áreas industriais. Para isso, esses rádios, comercialmente chamados de rádios bases ou repetidoras, têm se adequando aos padrões exigidos em ambientes industriais.

O sistema *wireless* pode ser utilizado em topologias ponto a ponto (figura 8.24), em que somente dois pontos de sinais são decodificados em sinais de rádio; ou ponto-multiponto (figura 8.25), em que vários pontos permanecem interligados, não havendo restrição de acesso entre eles.

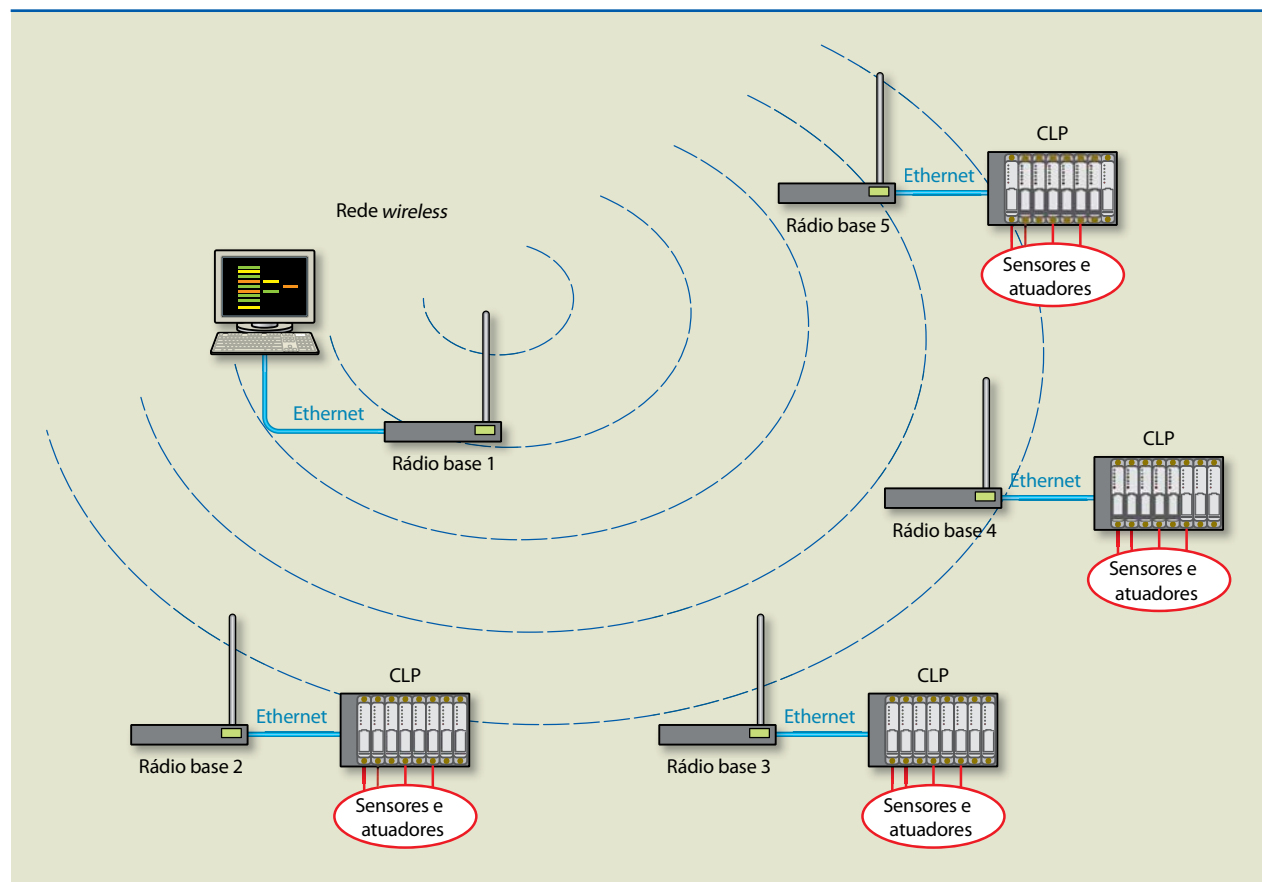


**Figura 8.24**

Topologia ponto a ponto em comunicações wireless.

Para que não haja problemas de comunicação em redes *wireless*, é importante conhecer dois aspectos básicos: tipos de antena e área de cobertura.





**Figura 8.25**

Topologia ponto-multiponto em comunicações wireless.

Os tipos de antena a serem utilizados estão diretamente ligados ao tipo de comunicação que deve ser estabelecido.

Basicamente, as antenas omnidirecionais (figura 8.26) possibilitam cobertura radial a partir do ponto em que estão fixas, e as direcionais (figura 8.27) possuem transmissão e recepção em determinado ângulo ao qual estão direcionadas. Ambos os modelos de antenas também são classificados pela potência de transmissão, que está relacionada com a área ou distância que se deseja cobrir com esse sinal.

**Figura 8.26**

Exemplo de antena omnidirecional.



WIKIMEDIA.ORG



VERBIGALEX/SHUTTERSTOCK

**Figura 8.27**

Exemplo de antena direcional.

A área de cobertura desejada é um detalhe importante. Em telecomunicações, existe um estudo denominado *site survey*. Relativamente complexo, ele analisa a amplitude do sinal na área desejada em relação ao melhor ponto de transmissão do sinal. Esse estudo costuma ser realizado quando a automação demanda que o sinal esteja disponível em equipamentos móveis, como pontes rolantes, veículos com automação embarcada e computadores de mão para apontamento de produção na planta. Tal análise de cobertura é feita normalmente por *softwares* do próprio fabricante do rádio, que, ligado a um computador móvel, efetua leituras de potência e qualidade do sinal transmitido. Com essa informação mapeada na planta em estudo, é possível efetuar a análise de viabilidade econômica, bem como se existe ou não a necessidade de instalação de rádios repetidores de sinal para aumentar a área de cobertura.

As principais razões da não utilização de redes wireless em larga escala na automação industrial são a disponibilidade do sinal, que pode ser comprometida por fatores externos ou mesmo pela qualidade dos produtos, e a segurança dos dados. Vale ressaltar que, em relação à disponibilidade do sinal, a tecnologia dos rádios têm evoluído muito nos últimos anos e, por conta disso, a estabilidade da comunicação já é uma realidade em muitas empresas. Quanto à segurança dessas redes, com o avanço da tecnologia do rádio, tem avançado também a tecnologia de encriptação de sinal, ou seja, de criação de chaves de segurança, que, uma vez definidas entre os dispositivos, reduzem a probabilidade de invasão dessas redes.

## 8.4 Sistemas multimestre

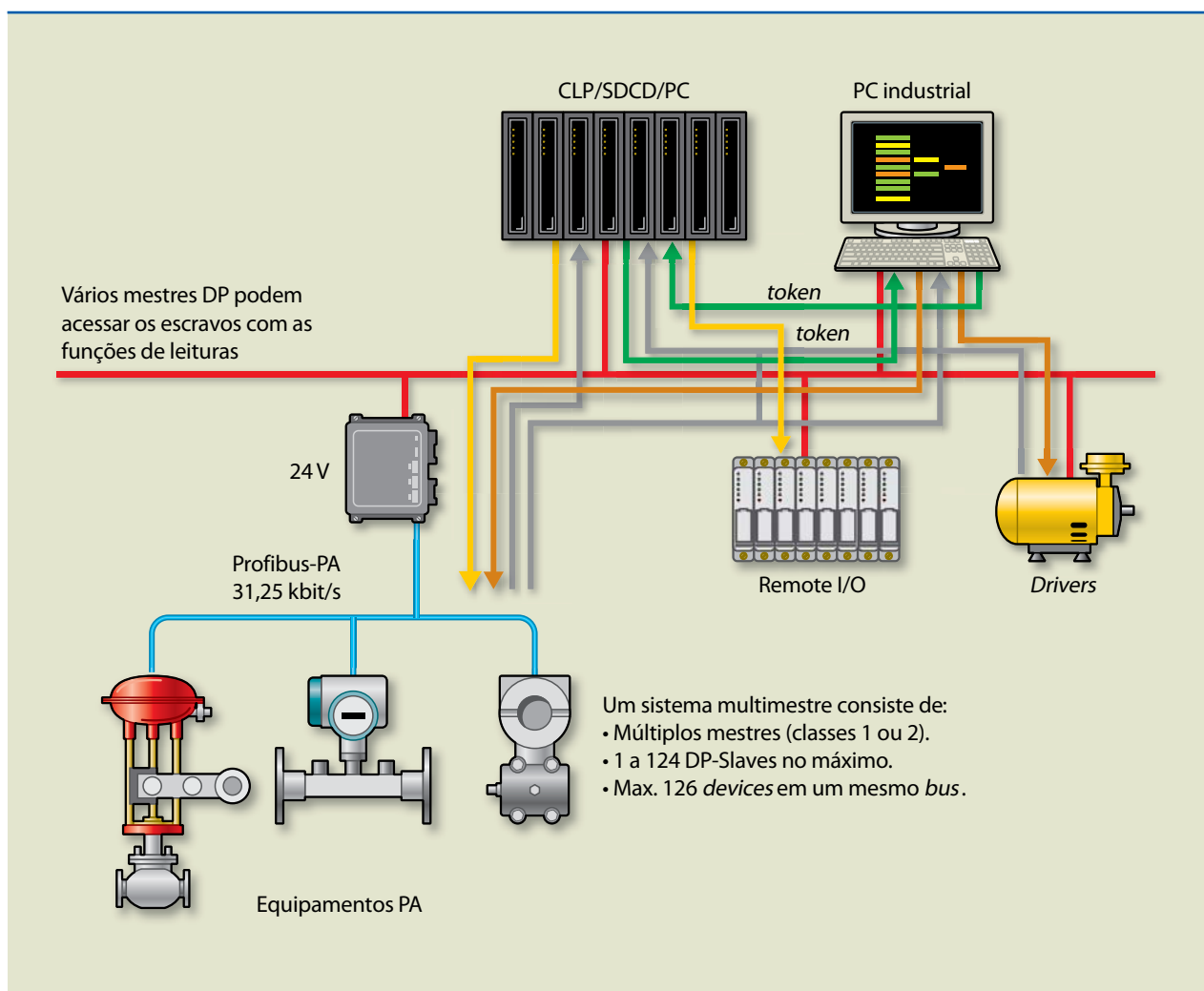
Muitos dos sistemas de automação necessitam de um controle mais apurado do que ocorre na planta produtiva. Processos químicos, de fundição, plantas de combustíveis e tantos outros processos industriais requerem alto índice de se-



gurança, visando não apenas a proteger a planta e os operadores, como também a minimizar perdas de matéria-prima. É comum que tais processos possuam sistemas que, em caso de falhas do controlador mestre, um controle secundário, tão poderoso quanto o primário, assuma o controle da planta produtiva.

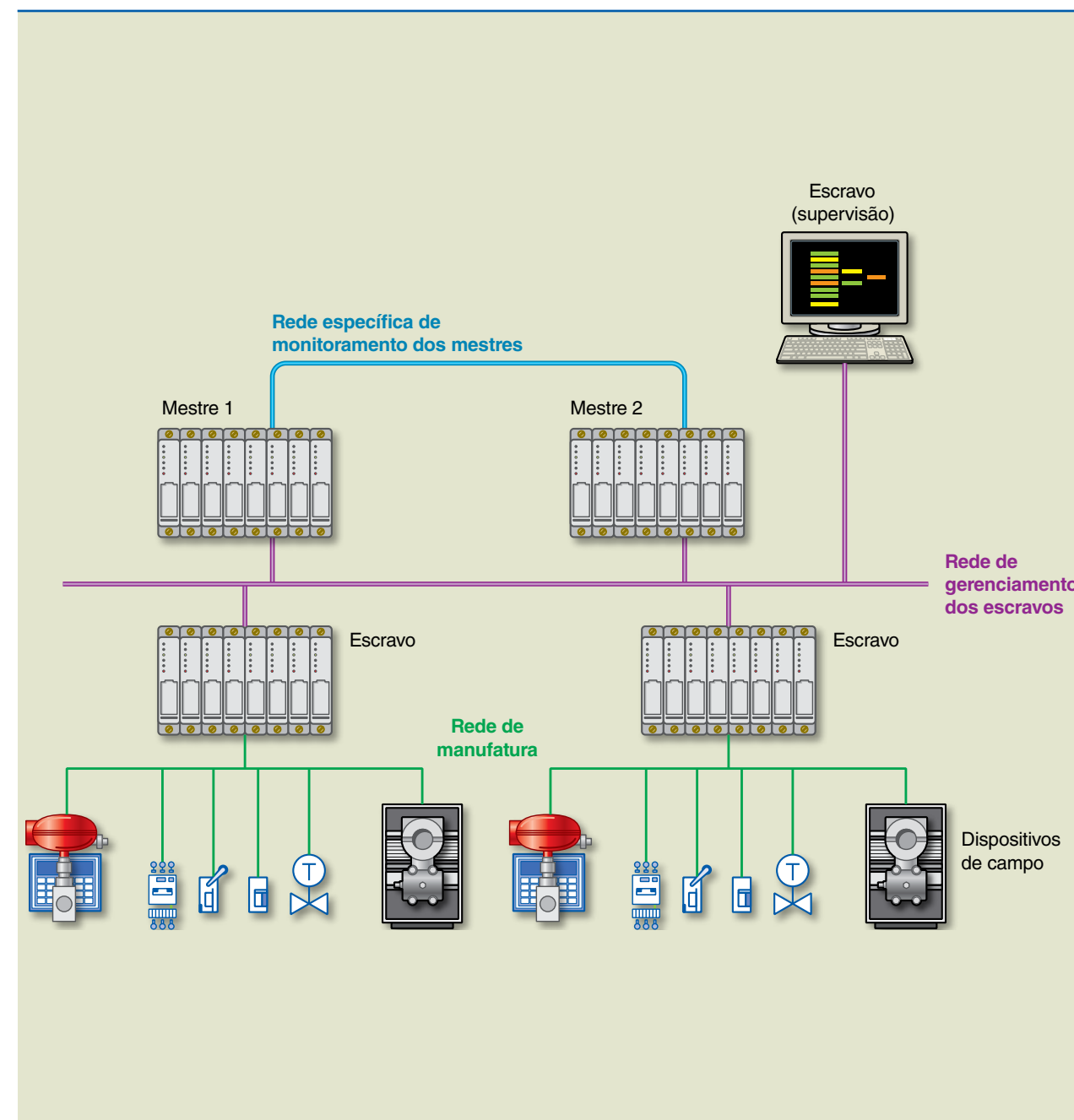
Nessa situação, além da necessidade de um *hardware* robusto, é importante que a rede de comunicação de dados possua a capacidade de “chavear” para um mestre secundário, em caso de falha do primário. Além do protocolo de comunicação, interligações de *hardware* possibilitam a rápida detecção da falha e a transferência de responsabilidade entre mestres. Um exemplo clássico sem grandes necessidades físicas é a utilização da topologia apresentada no protocolo Profibus DP. Nela, o próprio barramento de dados informa que um mestre deixou a operação, permitindo que o segundo assuma, conforme apresenta a figura 8.28. Nessa figura, é possível notar que os mestres podem ter acesso à leitura de todos os escravos (setas em cinza). Na atuação (setas em vermelho), apenas um deles pode assumir a tarefa. Usando o protocolo *Token Ring* (em verde), somente o mestre que detém o *Token* pode assumir o uso da rede. Dessa maneira, deve haver um esquema de arbitragem para que se decida quando o *Token* será passado ao outro mestre.

**Figura 8.28**  
Topologia multimestre Profibus DP.



Outras técnicas utilizadas em protocolos se valem do monitoramento de uma variável do mestre primário por meio de uma rede de dados secundária, específica para essa finalidade. Nessa topologia, um CLP mestre secundário, por exemplo, monitora um bit ou um byte do CLP primário, que deve ser atualizado constantemente. Caso o CLP primário tenha algum problema de processamento, não poderá atualizar essa variável e, por consequência, essa falha, ao ser percebida pelo CLP secundário, fará com que ele assuma o controle até que a situação seja regularizada. Essa técnica possibilita que sistemas de menor porte executem funções similares quando o barramento de comunicação principal não permitir fazer. A figura 8.29 exemplifica o esquema de interligação entre os dispositivos.

**Figura 8.29**  
Topologia multimestre genérica.





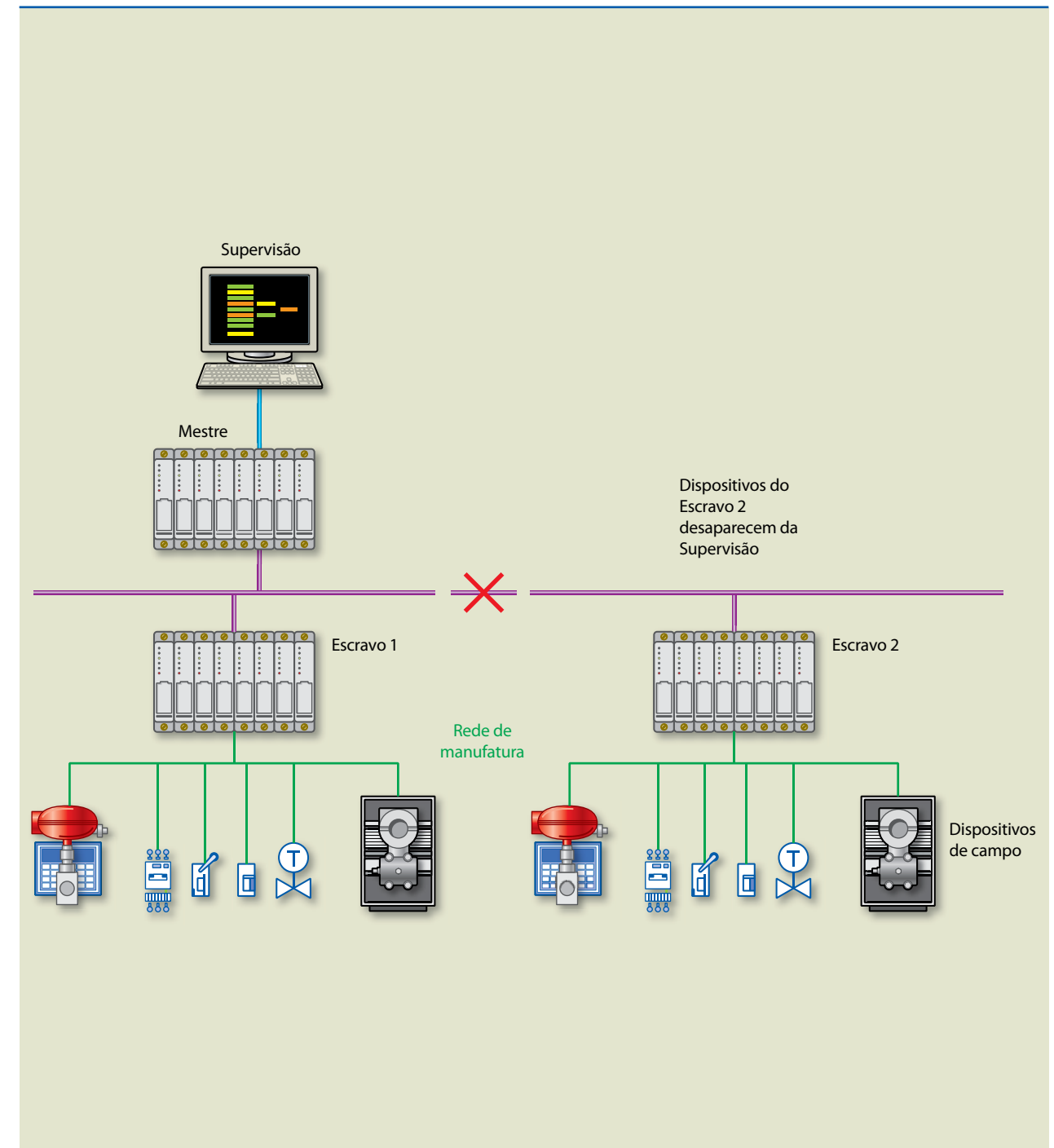
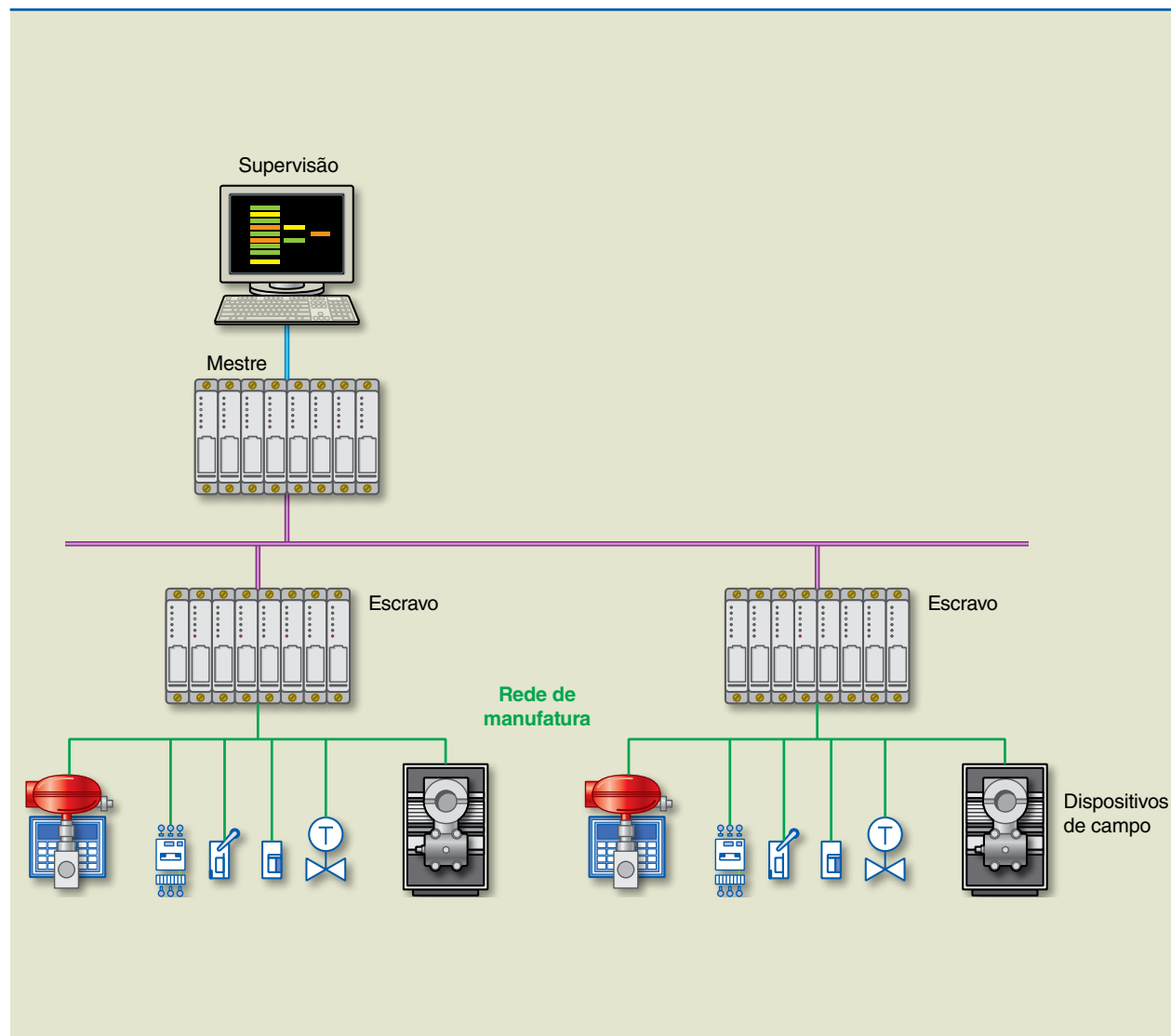
## 8.5 Topologias utilizadas na indústria

Nesta seção, são discutidas várias topologias de rede de comunicação utilizadas em automação industrial, com ênfase à nomenclatura empregada no chão de fábrica. Um aspecto importante a ser abordado é a continuidade do serviço no caso de falhas nas redes de comunicação, seja por quebra de cabos, seja por falhas nos transmissores da linha. As topologias são apresentadas com um breve descritivo de funcionamento. Vale ressaltar que todas elas permitem a utilização de interfaces em fibra óptica.

- **Rede em barramento** – Definida por uma linha comum de dados, permite que os dispositivos sejam ligados em paralelo ao longo dessa linha (figura 8.30). Foi uma das primeiras redes de comunicação utilizadas e tem como ponto fraco o fato de, quando rompido um dos cabos do par do barramento que a configura, todos os dispositivos a partir desse ponto ficam comprometidos (figura 8.31).

**Figura 8.30**

Rede em barramento.



**Figura 8.31**

Rede em barramento com falha no cabo.

- **Rede em anel** – Definida pela interligação bidirecional entre os dispositivos que a compõem (figura 8.32). O dispositivo de distribuição deve possuir certa “inteligência”, ou seja, deve detectar que o sinal que chega à sua direita é o mesmo que chega à sua esquerda e, portanto, deve desconsiderar um deles. Em caso de falha de um dos ramos da rede (figura 8.33), o dispositivo detecta que o sinal chegou somente por um dos lados e o utiliza, também informando ao *hardware* que gerencia o sistema que existe uma falha na rede de comunicação.



Figura 8.32 Rede em anel.

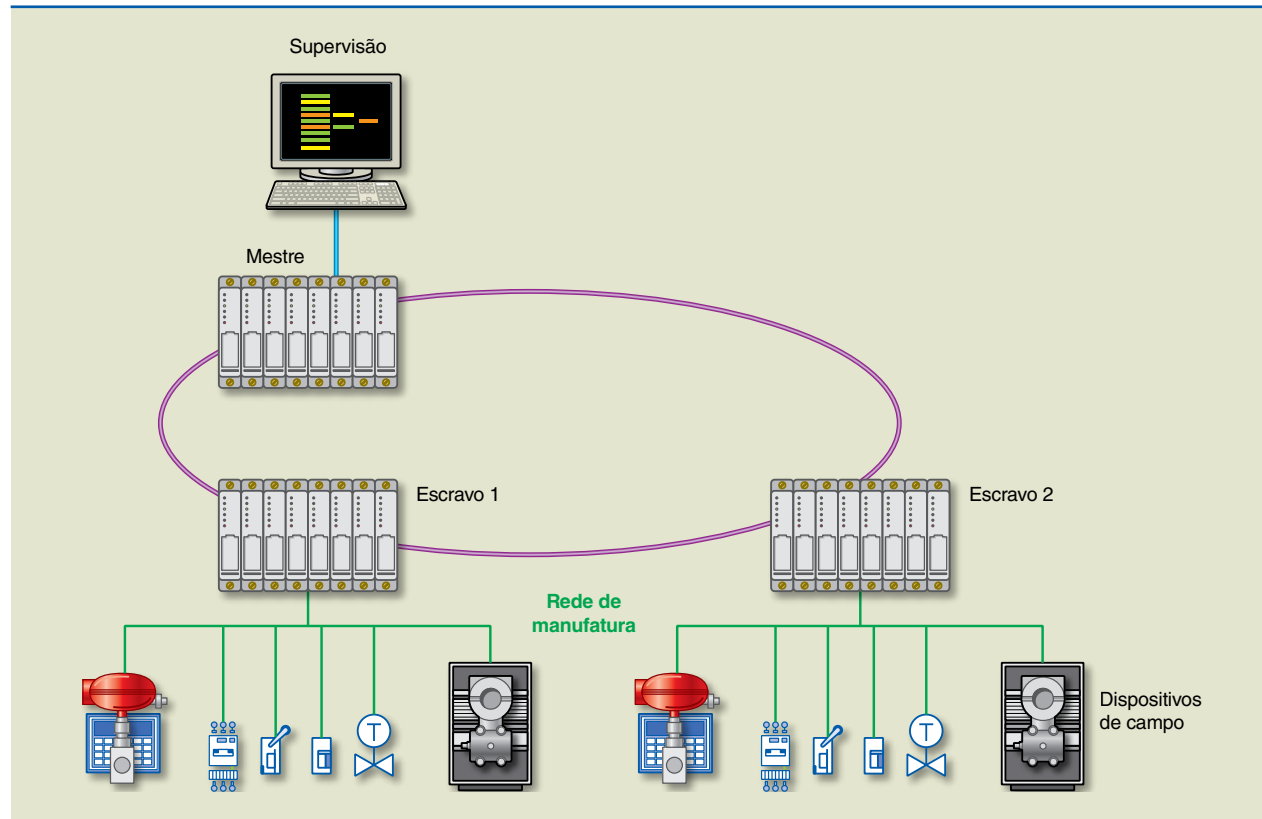
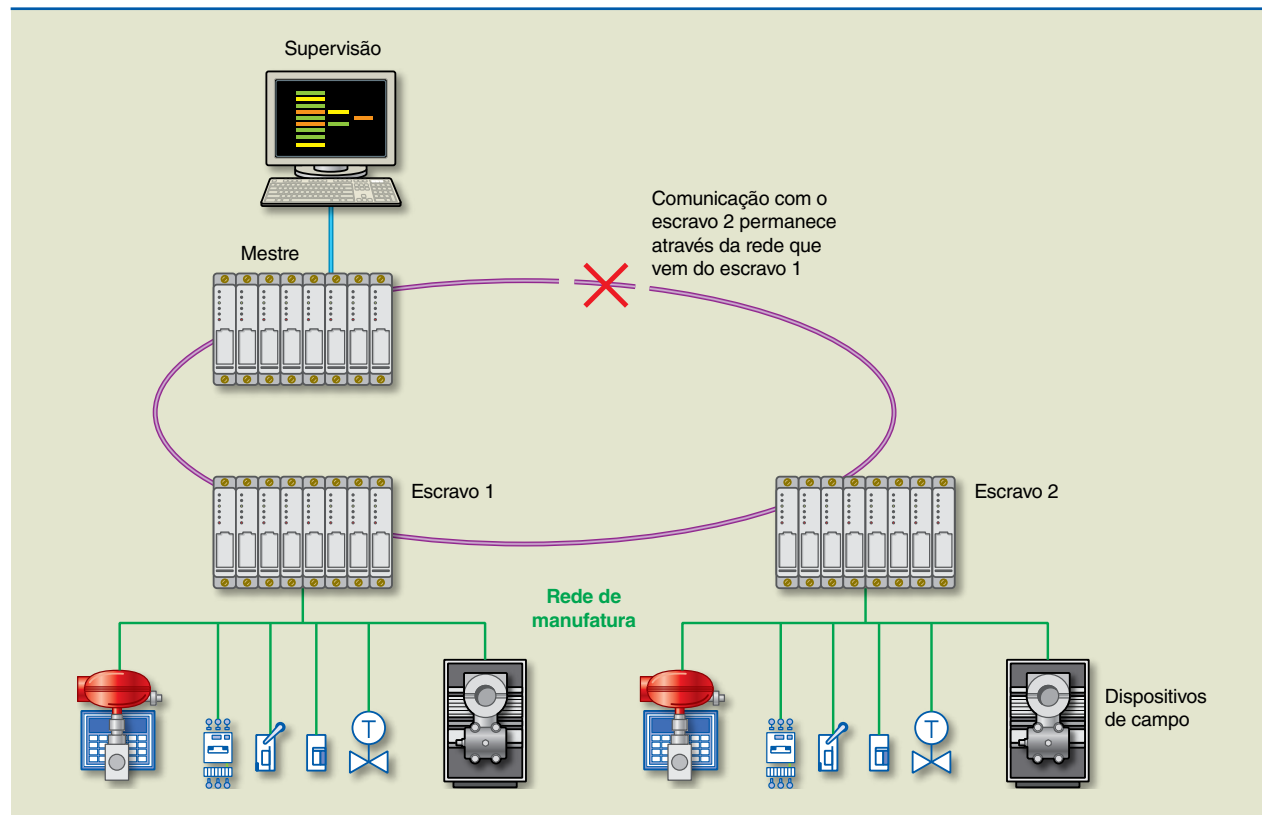
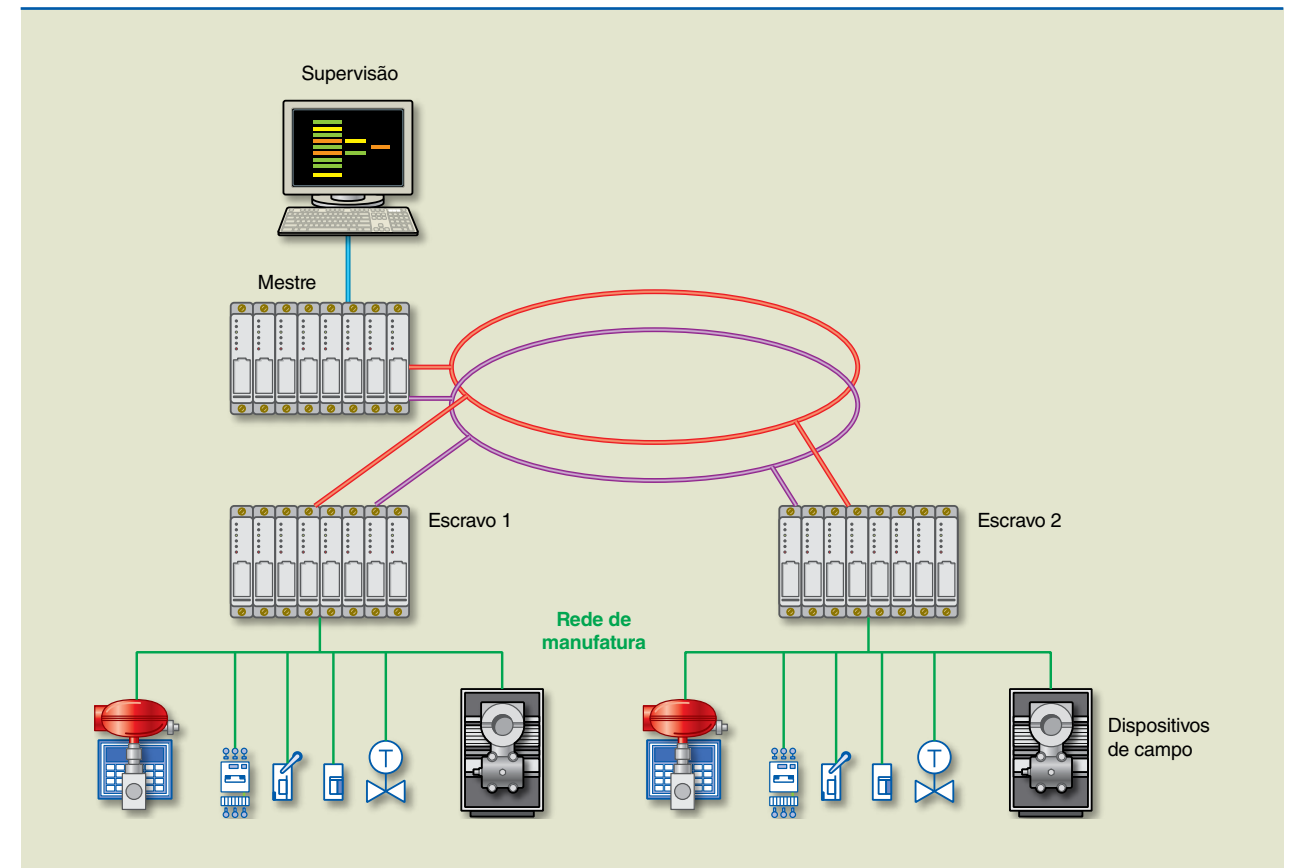


Figura 8.33 Rede em anel com ramo com falha no cabo.



- **Rede em anel redundante** – Asemelha-se à rede em anel (figura 8.34). Os dispositivos de distribuição possuem redundância na comunicação para o caso de falha do dispositivo que a gerencia. Praticamente não existe uma situação de falha na rede de comunicação que faça com que os dispositivos parem de se comunicar. Se houver falha de um dos dispositivos de distribuição, outro assume o controle de maneira automática e, em geral, transparente ao processo. É utilizada em situações muito críticas de disponibilidade. Gerencia redes de interface tipo Ethernet.

Figura 8.34 Rede em anel redundante.



The background features a light gray grid on a darker gray background. Two glowing, neon-like curves are visible: a blue curve that dips downwards and a red curve that rises to a peak. The text is centered over the red curve.

# **Referências**

---

# **bibliográficas**

---

ALVARENGA, B.; MÁXIMO, A. *Curso de física*. Vol. 3. 3ª ed. São Paulo: Scipione, 2007.

ANZENHOFER, K. et al. *Eletrotécnica para escolas profissionais*. 3ª ed. São Paulo: Mestre Jou, 1980.

CARVALHO, G. *Máquinas elétricas*. 2ª ed. revisada. São Paulo: Érica, 2008.

CAVALCANTI, P. J. M. *Fundamentos de eletrotécnica*. 21ª ed. Rio de Janeiro: Freitas Bastos, 2001.

CAVALIN, G.; CERVELIN, S. *Instalações elétricas prediais*. 18ª ed. São Paulo: Érica, 2008.

CHIQUETO, M. J. *Física na escola de hoje*. Vol. 3. São Paulo: Scipione, 1987.

COMO FUNCIONA: enciclopédia de ciência e técnica. São Paulo: Abril Cultural, 1974.

COTRIM, A. A. M. B. *Instalações elétricas*. 4ª ed. São Paulo: Pearson-Makron Books, 2008.

GOMES, A. T. *Telecomunicações: transmissão e recepção*. São Paulo: Érica, 1995.

MIYOSHI, E. M.; SANCHES, C. A. *Projetos de sistemas de rádio*. São Paulo: Érica, 2002.

MORAES, A. F. de. *Rede de computadores: fundamentos*. São Paulo: Érica, 2004.

RIBEIRO, J. J. A. *Comunicações ópticas*. São Paulo: Érica, 2011.

SAMPAIO, M. A. de; QUEIROZ, W. J. L. de. *Ondas eletromagnéticas e teoria de antena*. São Paulo: Érica, 2010.

SOARES NETO, V. *Telecomunicações: sistemas de modulação*. São Paulo: Érica, 2005.

SVERZUT, J. U. *Redes GSM, GPRS, EDGE e UMTS: evolução a caminho da quarta geração (4G)*. 2ª ed. São Paulo: Érica, 2007.



## SITES

ABNT: <<http://www.abnt.org.br>>. Acesso em: 29 jun. 2011.

Cortex: [www.cortex.com.br](http://www.cortex.com.br)

Energibrás: <<http://www.energibras.com.br>>. Acesso em: 29 jun. 2011.

GTD – Geração, Transmissão e Distribuição de Energia Elétrica:

<<http://www.dee.ufc.br/~rleao/GTD/1Introducao.pdf>>. Acesso em: 29 jun. 2011.

Instalações elétricas: <<http://www.instalacoeseltricas.com/teoria.asp?id=6>>.

Acesso em: 29 jun. 2011.

Lista de Exercícios - Interferência: <<http://www.if.ufrgs.br/tex/fisica-4/lista1.htm>>. Acesso em: 29 jun. 2011.

WEG: <<http://www.weg.net/br>>. Acesso em: 29 jun. 2011.



**CENTRO PAULA SOUZA DO GOVERNO DE SÃO PAULO**





## Excelência no ensino profissional

Administrador da maior rede estadual de educação profissional do país, o Centro Paula Souza tem papel de destaque entre as estratégias do Governo de São Paulo para promover o desenvolvimento econômico e a inclusão social no Estado, na medida em que capta as demandas das diferentes regiões paulistas. Suas Escolas Técnicas (Etecs) e Faculdades de Tecnologia (Fatecs) formam profissionais capacitados para atuar na gestão ou na linha de frente de operações nos diversos segmentos da economia.

Um indicador dessa competência é o índice de inserção dos profissionais no mercado de trabalho. Oito entre dez alunos formados pelas Etecs e Fatecs estão empregados um ano após concluírem o curso. Além da excelência, a instituição mantém o compromisso permanente de democratizar a educação gratuita e de qualidade. O Sistema de Pontuação Acrescida beneficia candidatos afrodescendentes e oriundos da Rede Pública. Mais de 70% dos aprovados nos processos seletivos das Etecs e Fatecs vêm do ensino público.

O Centro Paula Souza atua também na qualificação e requalificação de trabalhadores, por meio do Programa de Formação Inicial e Educação Continuada. E ainda oferece o Programa de Mestrado em Tecnologia, recomendado pela Capes e reconhecido pelo MEC, que tem como área de concentração a inovação tecnológica e o desenvolvimento sustentável.