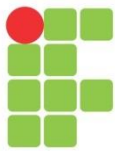


INSTITUTO FEDERAL  
MINAS GERAIS  
Campus Avançado Ipatinga

# *Automação Industrial*

## *Introdução à Automação Industrial*

**Curso: Engenharia Elétrica**  
**Prof. Sandro Dornellas**



**Carga Horária: 66,66 horas / 80 H/A (50 minutos)**

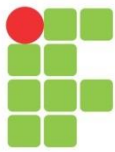
**Limite de Faltas: 20 H/A**

**Total de pontos a distribuir: 100**

Avaliação Teórica: 70

Avaliação Prática: 20

Listas de exercícios / Semana Tecnológica: 10



## **Ementa:**

Aspectos históricos e evolução tecnológica; Introdução ao processo de automação; Universo da automação industrial e perspectivas; Controle de sistemas a eventos discretos; Sensores discretos; Sistemas digitais de controle distribuído (SDCD); Redes industriais: arquiteturas e tecnologias; Controladores lógicos programáveis (CLP): arquitetura, programação; Interface Homem Máquina - IHM e sistemas supervisórios.

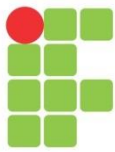


# Bibliografia

MORAES, Cícero Couto de; CASTRUCCI, Plínio. **Engenharia de automação industrial**. 2. ed. Rio de Janeiro: LTC, 2006. 360 p.

FRANCHI, Claiton Moro; CAMARGO, Valter Luís Arlindo de. **Controladores lógicos programáveis: sistemas discretos**. 2. ed. São Paulo: Érica, 2009. 352 p.

GROOVER, Mikell P. **Automação industrial e sistemas de manufatura**. 3. ed. Pearson Universidades, 2010. 592 p.



## Definição

Entende-se por automação qualquer sistema, apoiado em computadores, que substitua o trabalho humano em favor da segurança das pessoas, da qualidade dos produtos, da rapidez da produção ou da redução de custos, assim aperfeiçoando os complexos objetivos das indústrias e dos serviços.

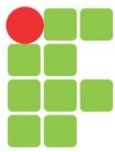
**Exemplos:** automação da mineração, da manufatura metálica, dos grandes processos químicos contínuos, automação bancária, metroviária, aeroportuária.



## Definição

A automação envolve a implantação de sistemas interligados e assistidos por redes de comunicação, compreendendo sistemas supervisórios e interfaces homem-máquina que possam auxiliar os operadores no exercício da supervisão e da análise dos problemas que porventura venham a ocorrer.





## História

O termo “automação” (*automation*) foi originalmente criado, em 1946, por um engenheiro da Ford Motor Company, para descrever a variedade de dispositivos automáticos de transferência e os mecanismos de alimentação que haviam sido instalados nas plantas de produção da empresa.

A primeira máquina-ferramenta de controle numérico foi criada e demonstrada em 1952, no MIT.



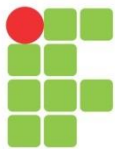
## História

No fim da década de 1960 e no início da de 1970, os computadores digitais começaram a ser conectados às máquinas-ferramenta.

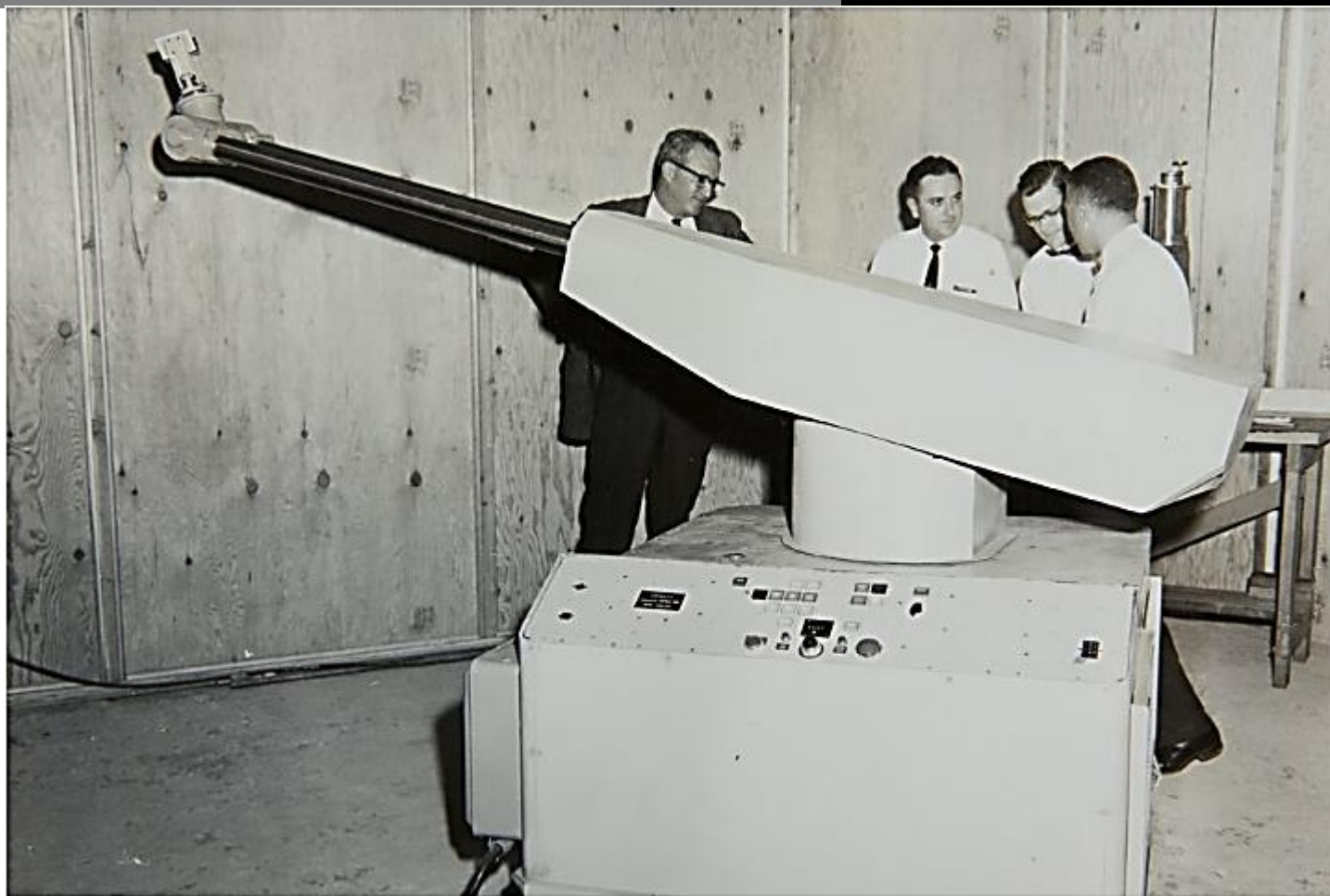
Em 1961, o primeiro robô industrial foi instalado para descarregar peças em uma operação de fundição.

No fim da década de 1960, o primeiro sistema de manufatura flexível dos EUA foi instalado para executar operações de máquina em uma variedade de peças.





# Introdução à Automação Industrial



O “Unimate”, primeiro robô industrial projetado e patenteado por George Devol.

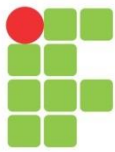


## História

Por volta de 1969, a General Motors desenvolveu o primeiro controlador lógico programável – CLP.



CLP da Modicon, década de 1980.



## História

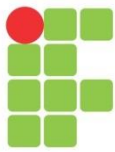
Os avanços no desenvolvimento de programas relacionados à automação também foram importantes:

- ❖ Linguagem FORTRAN (1955);
- ❖ Linguagem APT para controle numérico (CN) de máquinas-ferramenta (1961);
- ❖ SO UNIX (1969);
- ❖ Linguagem VAL para programação de robôs (1979);
- ❖ SO Windows (1985);
- ❖ Linguagem JAVA (1995);



## Introdução

A automação não resulta tão somente do objetivo de reduzir custos de produção, ela decorre de necessidades tais como maior nível de qualidade, expressa por especificações numéricas de tolerância, maior flexibilidade de modelos para o mercado, maior segurança pública e dos operários, menores perdas materiais e de energia, mais disponibilidade e qualidade da informação sobre o processo e melhor planejamento e controle da produção.



## Introdução

Os sistemas de controle e automação estão presentes em praticamente todas as atividades industriais, comerciais e de serviços, sendo a base da automação de processos industriais;

Os **processos industriais** são formados por equipamentos mecânicos, elétricos, eletrônicos, hidráulicos ou pneumáticos que, através de sucessivas operações utilizando matéria-prima e energia, resultará num produto final e resíduos;



## Arquitetura da Automação Industrial

A automação industrial exige a realização de muitas funções. A figura a seguir representa a chamada **Pirâmide de Automação**, com os diferentes níveis de automação encontrados em uma planta industrial.

Na base da pirâmide está frequentemente envolvido o CLP, atuando via inversores, conversores ou sistemas de partida suave sobre máquinas e motores e outros processos produtivos.



## Níveis de pirâmide

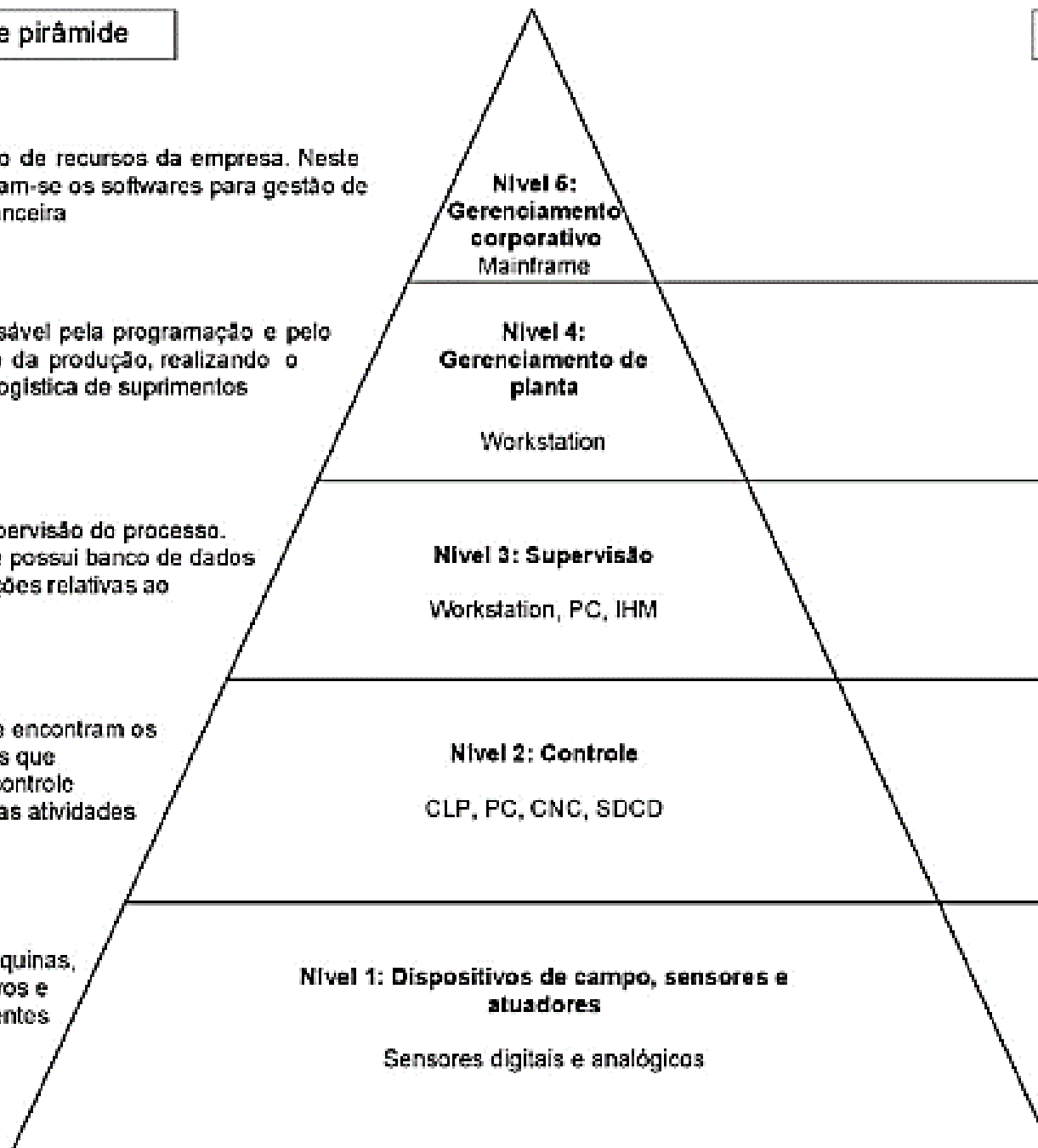
Administração de recursos da empresa. Neste nível encontram-se os softwares para gestão de vendas e financeira

Nível responsável pela programação e pelo planejamento da produção, realizando o controle e a logística de suprimentos

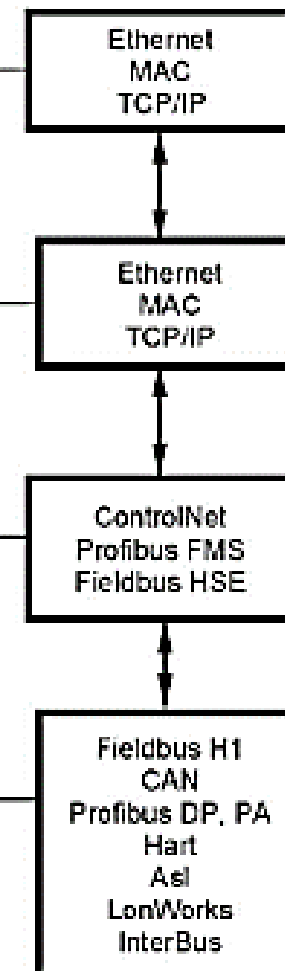
Permite a supervisão do processo. Normalmente possui banco de dados com informações relativas ao processo

Nível onde se encontram os equipamentos que executam o controle automático das atividades da planta

Nível das máquinas, dos dispositivos e dos componentes da planta



## Protocolos



Pirâmide da automação.



## Arquitetura da Automação Industrial

**Nível 1:** é o nível das máquinas, dispositivos e componentes (chão de fábrica). Ex.: máquinas de embalagem, linha de montagem ou manufatura.

**Nível 2:** é o nível dos controladores digitais, dinâmicos e lógicos, e de algum tipo de supervisão associada ao processo. Aqui se encontram concentradores de informações sobre o Nível 1, e as Interfaces Homem-Máquina (IHM).

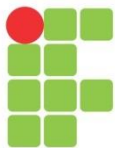




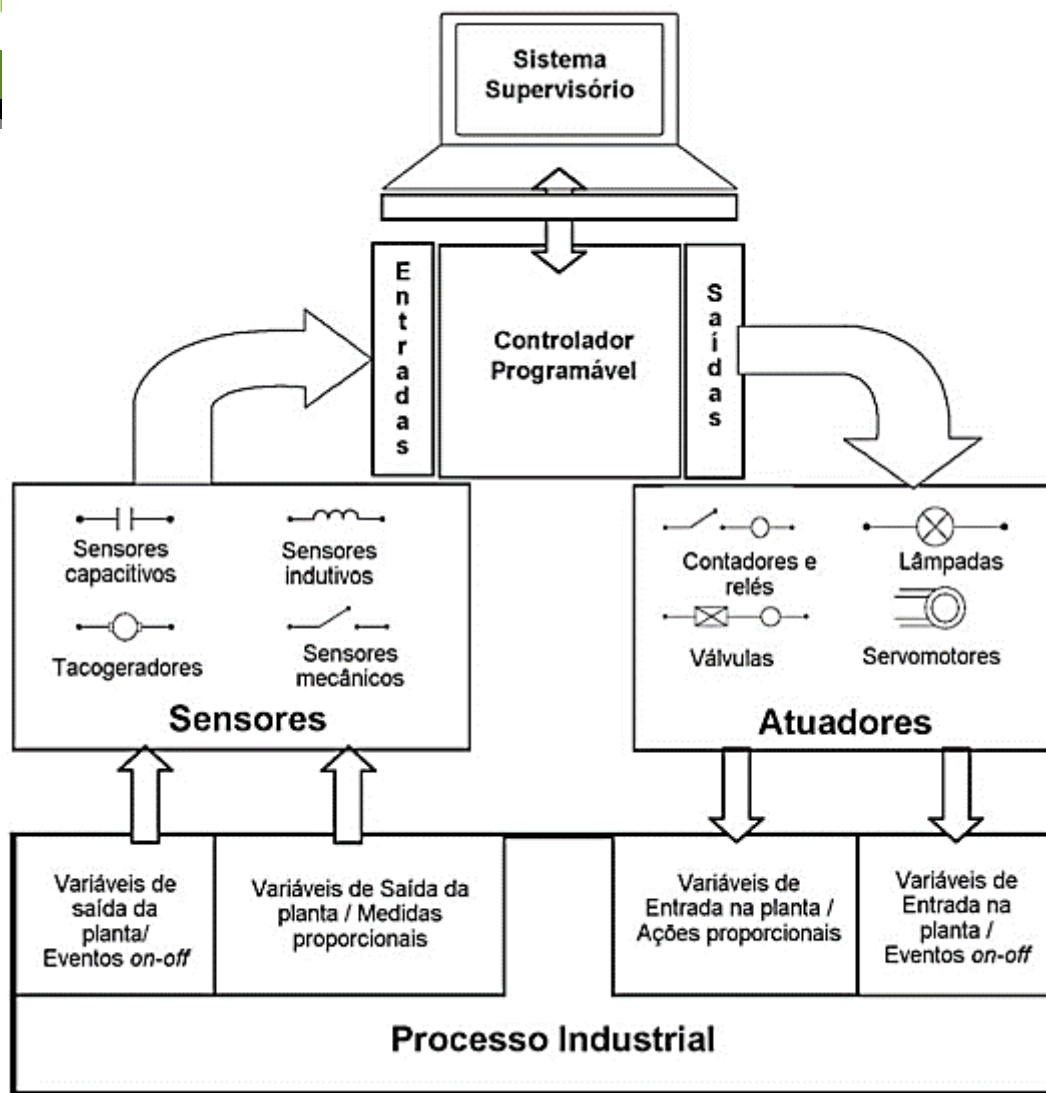
## Arquitetura da Automação Industrial

**Nível 3:** permite o controle do processo produtivo da planta; normalmente é constituído por bancos de dados com informações dos índices de qualidade da produção, relatórios e estatísticas de processo, índices de produtividade, algoritmos de otimização da operação produtiva. Ex.: avaliação e controle da qualidade em processo químico ou alimentício; supervisão de um laminador a frio.

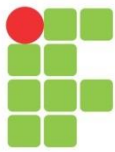
**Nível 4:** é o nível responsável pela programação e pelo planejamento da produção, realizando o controle e a logística dos suprimentos. Ex.: controle de suprimentos e estoques em função da sazonalidade e da distribuição geográfica.



# Introdução à Automação Industrial



Níveis 1, 2 e 3 da automação industrial.



## Arquitetura da Automação Industrial

**Nível 5:** é o nível responsável pela administração dos recursos da empresa, em que se encontram os softwares para gestão de vendas e gestão financeira; é também onde se realizam a decisão e o gerenciamento de todo o sistema.



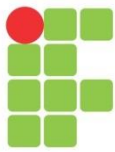


## Níveis de complexidade da automação

Os sistemas de automação podem ser distinguidos em 3 níveis de complexidade quanto à meios de projeto e realização física:

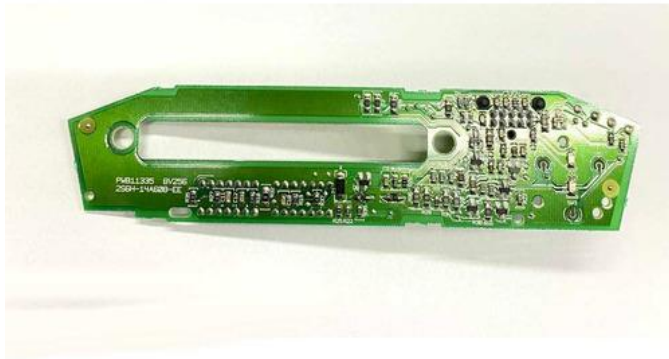
**Complexidade “menor”:** compreendem as automações especializadas (embarcadas), como aparelhos de TV, ar-condicionado, eletrodomésticos, automóveis, etc.

Utilizam-se de microcontroladores de pequenas memórias, dedicados, montados em placas de circuito impresso no interior dos equipamentos.



## Níveis de complexidade da automação

Programa-se diretamente em “linguagem de máquina” e grava-se em memórias tipo ROM.



(a) Circuito interno de um ar condicionado de automóvel. (b) circuito interno de um aparelho de TV.



## Níveis de complexidade da automação

**Complexidade “maior”:** compreendem os grandes sistemas de automação, envolvem áreas extensas e muitos computadores de vários tipos e capacidades. Exemplos: sistemas de controle de voo, de controle metroviário, de defesa militar.

Sua programação envolve engenharia de software em tempo real, programas de aplicação comercial e científica, programas em C, Ada e outras linguagens específicas de tempo real.





## Níveis de complexidade da automação



(a) Sistema de controle de voo. (b) Sistema de defesa aérea e circulação operacional militar.



## Níveis de complexidade da automação

**Complexidade “média”:** englobam as automações industriais e de serviços, sendo este nível o foco da disciplina. Exemplos: sistemas transportadores industriais, manufaturas, processos químicos, térmicos, controle de nível, etc.

Comportam-se muito bem com o emprego dos CLPs e suas linguagens de programação, seus softwares e aplicativos para controle.



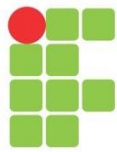


## Projeto de automação

O desenvolvimento de um projeto de automação abrange duas modalidades distintas.

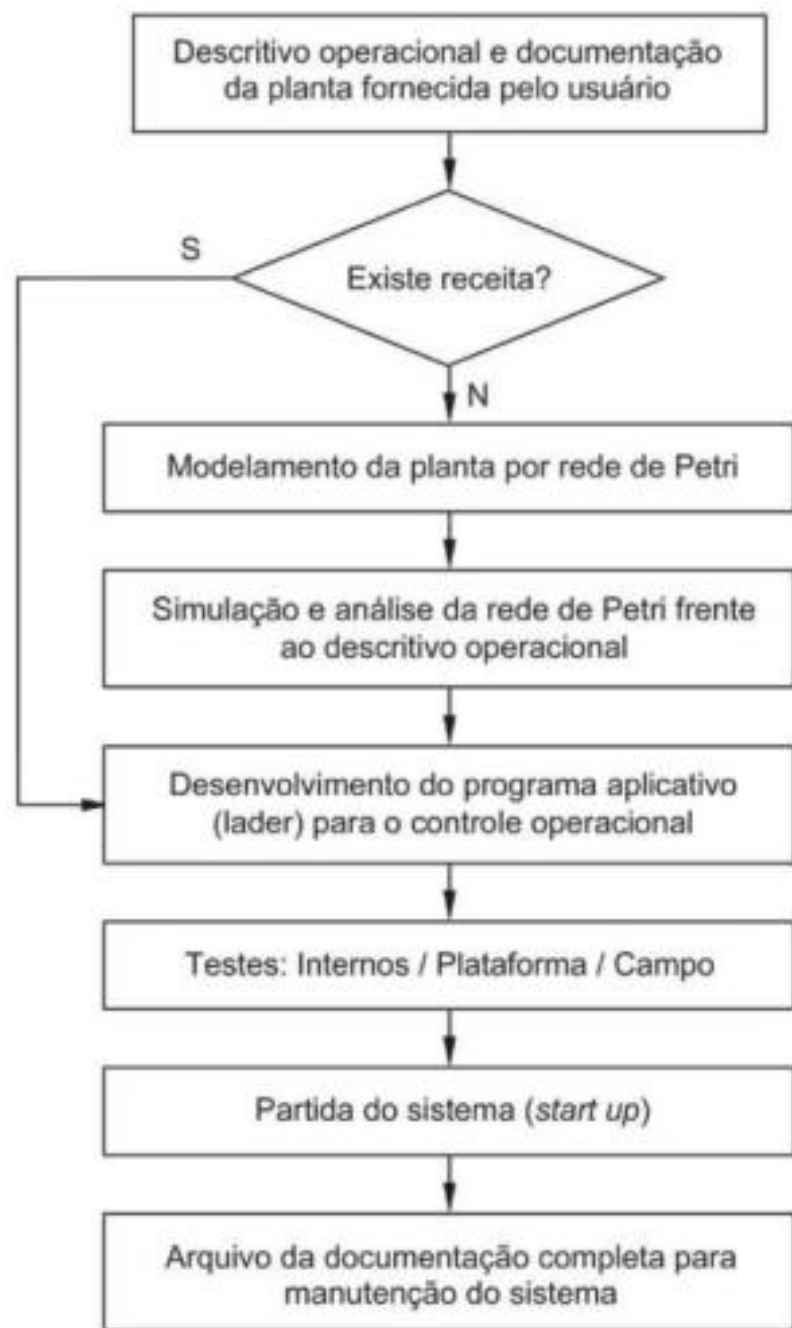
Na **primeira**, o usuário sabe exatamente todas as ações que devem ser automatizadas, definindo o que deve ocorrer em cada circunstância.

Na **segunda**, o usuário somente define o resultado final desejado, cabendo ao engenheiro de projeto definir toda a lógica das ações.



## Projeto de Automação

Diagrama de blocos descritivo das etapas para implementação da automação.





## Controlador Lógico Programável - CLP

A vantagem de utilizar sistemas que envolvam diretamente a informatização é a possibilidade da expansão utilizando recursos de fácil acesso; nesse contexto, são de extraordinária importância os **controladores lógicos programáveis (CLPs)**, que tomam a automação industrial uma realidade onipresente.



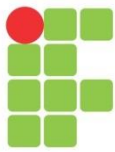
CLPs atuais.



## Controlador Lógico Programável - CLP

O Controlador Lógico Programável – CLP (*Programmable Logic Controller – PLC*) é um dispositivo digital que controla máquinas e processos.

Utiliza uma memória programável para armazenar instruções e executar funções específicas: energização/desenergização, temporização, contagem, sequenciamento, operações matemáticas e manipulação de dados.



## Controlador Lógico Programável - CLP

O CLP vem substituir circuitos de relés que integravam o antigo painel industrial. Para efetuar uma modificação da lógica dos comandos, por qualquer motivo, era necessário um rearranjo na montagem, cansativo, demorado e dispendioso.

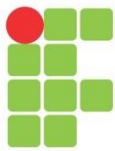
Modificação que, às vezes, implicava uma reforma total dos painéis elétricos. Com o CLP basta modificar o programa, mantendo o hardware.



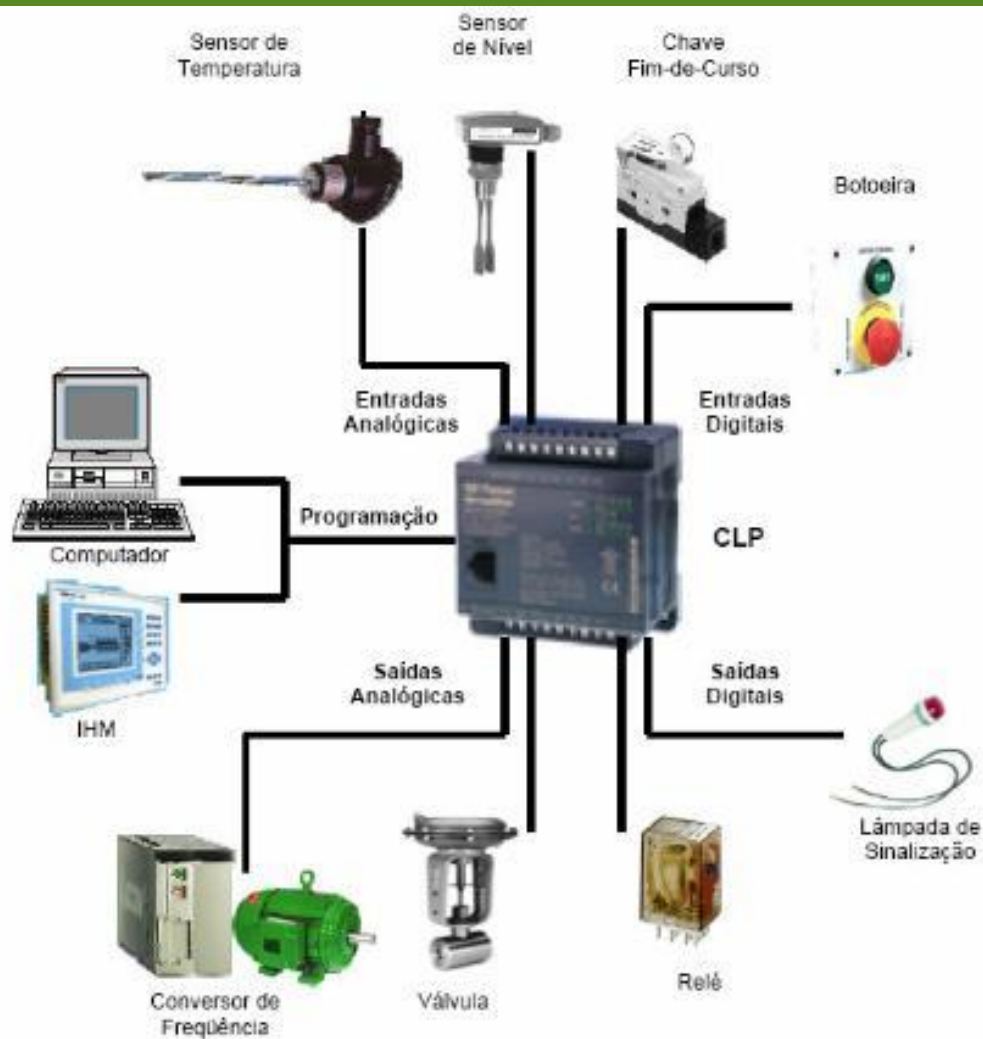
## Controlador Lógico Programável - CLP



Painéis elétricos à base de relés.

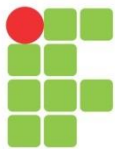


# Introdução à Automação Industrial

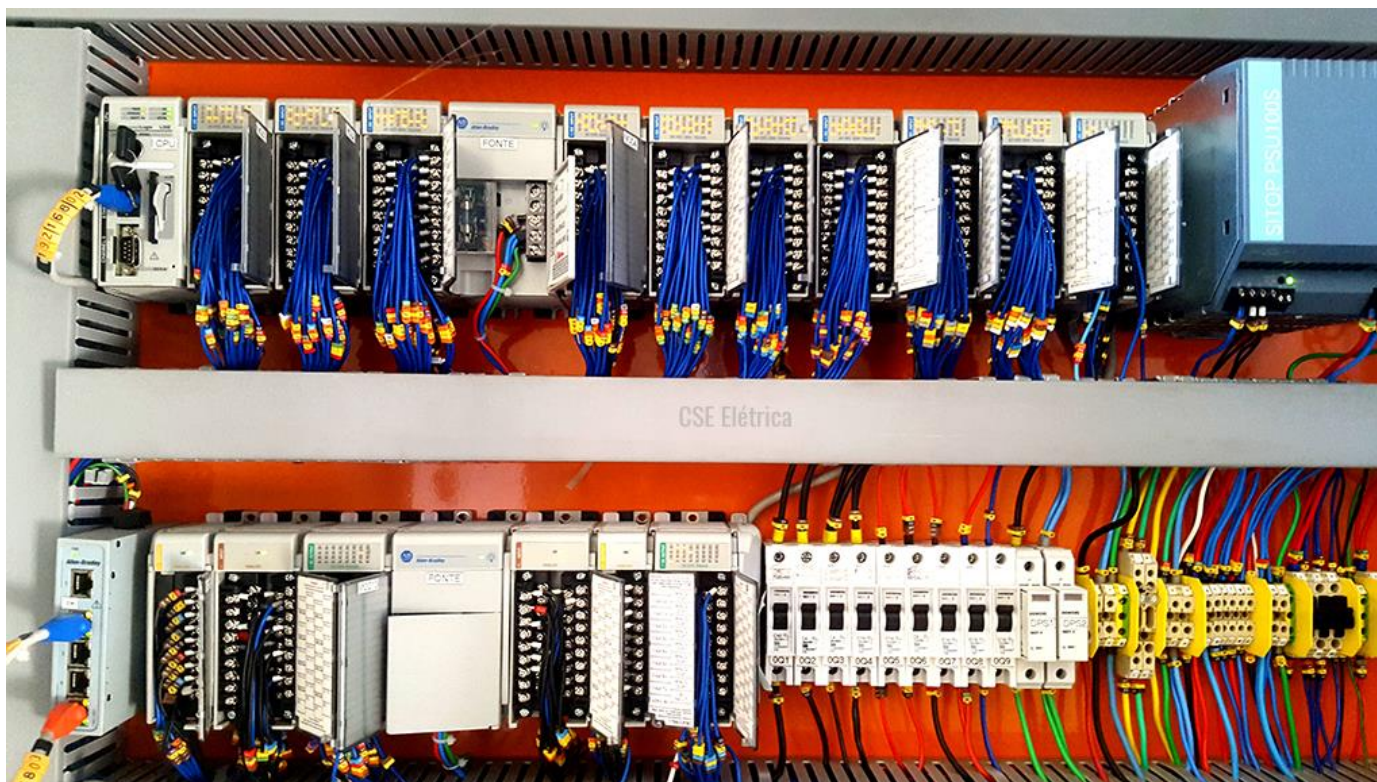


Periféricos conectados à um único CLP.





## Controlador Lógico Programável - CLP



Painel elétrico com CLPs e módulos de expansão.





## Controlador Lógico Programável - CLP

Os CLPs, de extraordinária importância prática nas indústrias, caracterizam-se por:

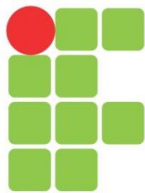
- Robustez adequada aos ambientes industriais (geralmente não incluem telas);
- Programação por meio de computadores pessoais (PCs);
- Linguagens amigáveis para o projetista de automação;
- Permitir tanto o controle lógico quanto o controle dinâmico (PID);
- Incluir modelos capazes de conexões em grandes redes de dados.



## Controlador Lógico Programável - CLP

Um CLP automatiza uma grande quantidade de ações com precisão, confiabilidade, rapidez e pouco investimento.

Informações de entrada são analisadas, decisões são tomadas, comandos são transmitidos, tudo concomitantemente com o desenrolar do processo.



INSTITUTO FEDERAL  
MINAS GERAIS  
Campus Avançado Ipatinga

# *Automação Industrial*

## *Controladores Lógicos Programáveis - CLPs*

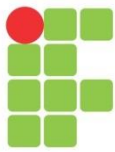
**Curso: Engenharia Elétrica**  
**Prof. Sandro Dornellas**



## Definição

Um CLP é definido pelo IEC (*International Electrotechnical Commission*) como:

"Sistema eletrônico operando digitalmente, projetado para uso em um ambiente industrial, que usa uma memória programável para a armazenagem interna de instruções orientadas para o usuário para implementar funções específicas, tais como lógica, sequencial, temporização, contagem e aritmética, para controlar, através de entradas e saídas digitais ou analógicas, vários tipos de máquinas ou processos. O CLP e seus periféricos associados são projetados para serem facilmente integráveis em um sistema de controle industrial e facilmente usados em todas suas funções previstas."



## Utilização dos CLP's

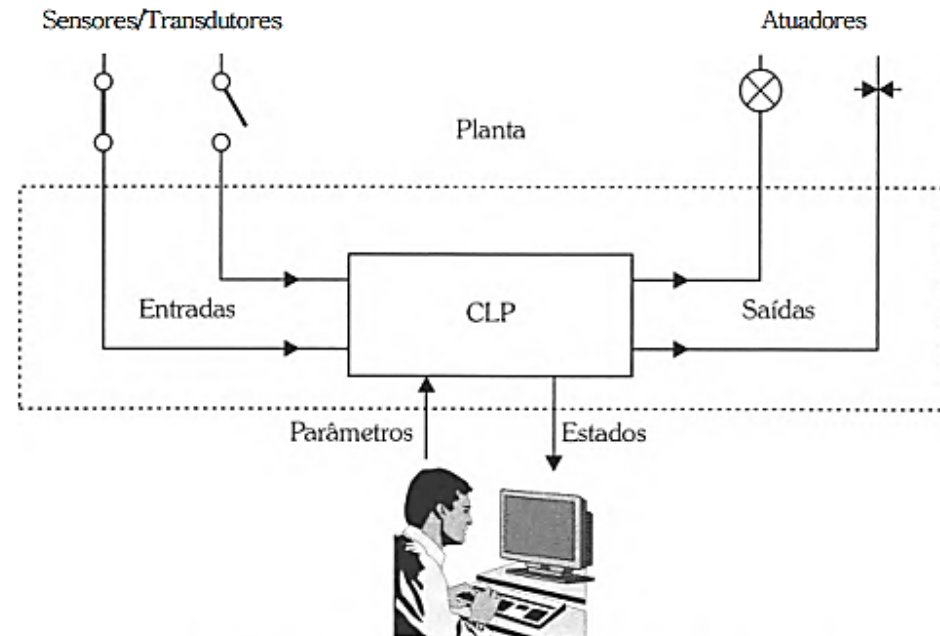
Toda planta industrial necessita de algum tipo de controlador para garantir uma operação segura e economicamente viável.

Desde o nível mais simples, em que pode ser utilizado para controlar o motor elétrico de um ventilador para regular a temperatura de uma sala, até um grau de complexidade elevado, controlando a planta de um reator nuclear para produção de energia elétrica.

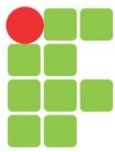


## Utilização dos CLP's

Embora existam tamanhos e complexidades diferentes, todos os sistemas de controle podem ser divididos em três partes com funções bem definidas: os transdutores (sensores), os controladores e os atuadores.



Representação dos sistemas de controle.



## Utilização dos CLP's

**Sensores/transdutores:** transdutor é um dispositivo que converte uma condição física do elemento sensor em um sinal elétrico para ser utilizado pelo CLP através da conexão às entradas do CLP. Um exemplo típico é um botão de pressão momentânea, em que um sinal elétrico é enviado do botão de pressão ao CLP, indicando sua condição atual.



Botoeiras e sensores industriais.



## Utilização dos CLP's

**Atuadores:** sua função é converter o sinal elétrico oriundo do CLP em uma condição física, normalmente ligando ou desligando algum elemento. Os atuadores são conectados às saídas do CLP. Um exemplo típico é fazer o controle do acionamento de um motor através do CLP. Neste caso a saída do CLP vai ligar ou desligar a bobina do contator que o comanda.



Exemplos de atuadores: motores, eletroválvulas, cilindros pneumáticos.





## Utilização dos CLP's

**Controladores:** de acordo com os estados das suas entradas, o controlador utiliza um programa de controle para calcular os estados das suas saídas. Os sinais elétricos das saídas são convertidos no processo através dos atuadores. O operador pode interagir com o controlador por meio dos parâmetros de controle. Alguns controladores podem mostrar o estado do processo em uma tela ou em um display.



CLP “Clic 02” da WEG.



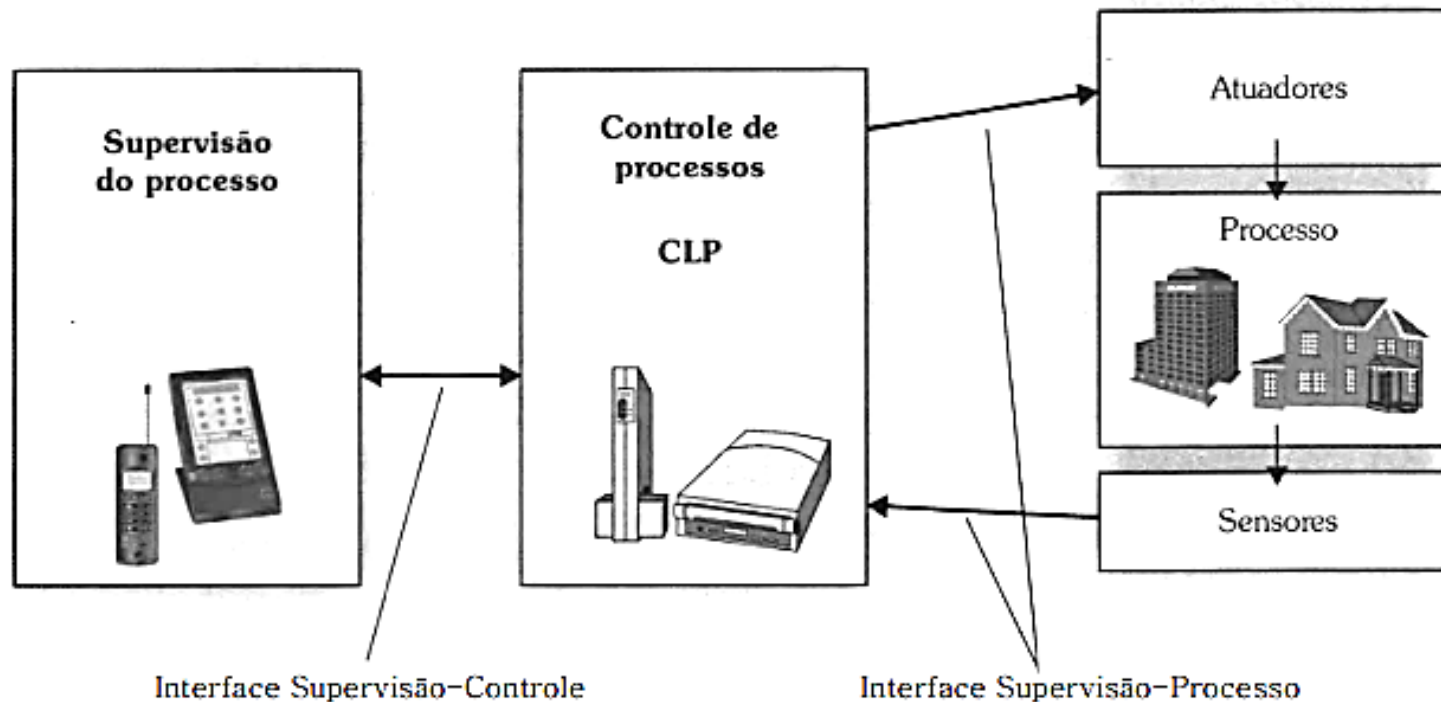
## Utilização dos CLP's

As atuais funções de controle existentes em uma planta industrial são normalmente distribuídas entre um número de controladores programáveis, os quais poder ser instalados próximo dos equipamentos a serem controlados.

Os diferentes controladores são usualmente conectados via rede local (LAN) a um sistema supervisório central, o qual gerencia as diversas informações do processo controlado, tais como alarmes, receitas e relatórios.



## Utilização dos CLP's



Etapas de supervisão e controle utilizando CLP's.



## Características (e benefícios) do sistema com CLP

Características do sistema com CLP	Benefícios
Uso de componentes de estado sólido	Alta confiabilidade
Memória programável	Simplifica mudanças Flexibiliza o controle
Pequeno tamanho	Necessita de um espaço mínimo para instalação
Microprocessador	Capacidade de comunicação Alto nível de performance Alta qualidade dos produtos Possibilidade de trabalhar com muitas funções simultaneamente
Contadores/temporizadores via <i>software</i>	Facilidade para alterar <i>presets</i> Elimina <i>hardware</i>
Controle de relés via <i>software</i>	Reduz custo em hardware/cabeamento Redução de espaço



## Características (e benefícios) do sistema com CLP

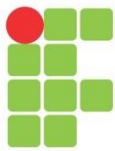
Arquitetura modular	Flexibilidade para instalação Facilmente instalado Redução de custos de <i>hardware</i> Expansibilidade
Variedades de interfaces de I/O	Controle de uma grande variedade de I/O Elimina um controle dedicado
Estações remotas de I/O	Elimina cabeamentos longos
Indicadores de diagnóstico	Reduz tempo de manutenção Sinaliza a operação correta/incorreta do sistema de controle
Interfaces modulares de I/O	Facilita a manutenção Facilita o cabeamento
Variáveis de sistema alocadas na memória de dados	Facilita gerenciamento/manutenção Podem ser colocadas na forma de um relatório de saída



## Arquitetura de um CLP

Um CLP é constituído basicamente de:

- Fonte de alimentação;
- Unidade Central de Processamento (UCP);
- Memórias dos tipos fixo e volátil;
- Dispositivos de entrada e saída;
- Terminal de programação.



## Arquitetura de um CLP

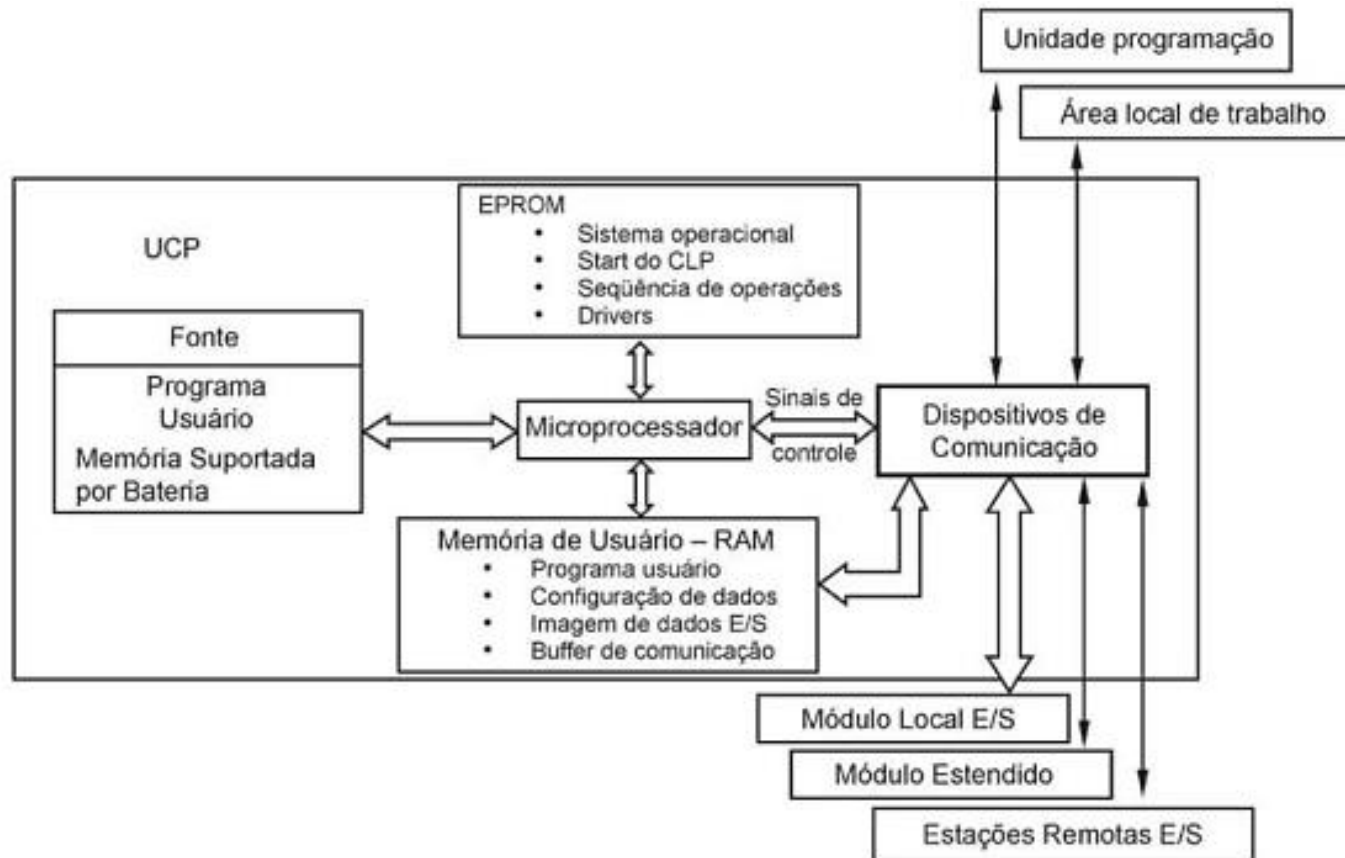
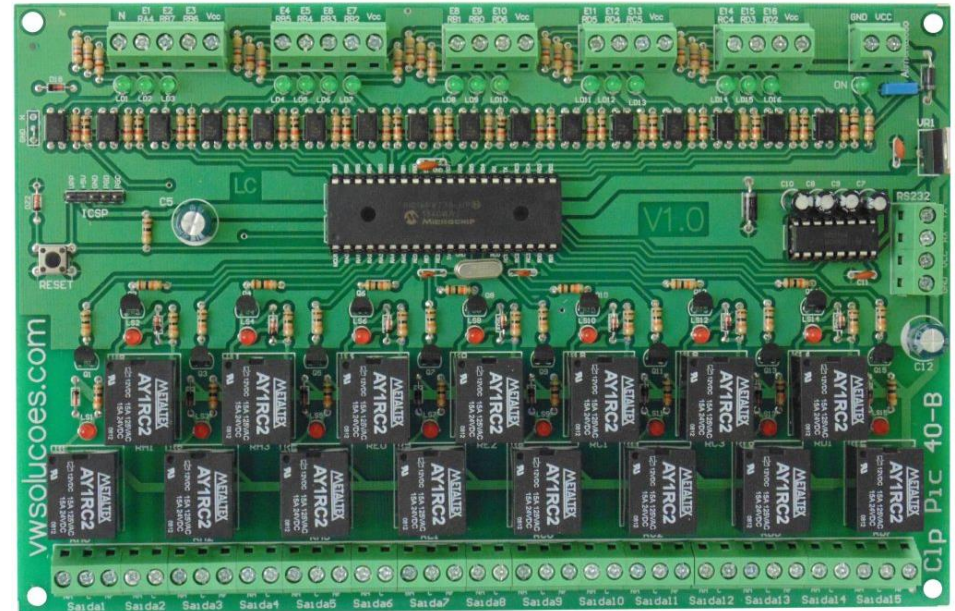
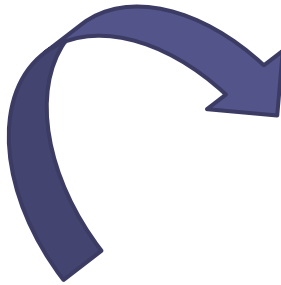


Diagrama de blocos da UCP do CLP.





## Arquitetura de um CLP



Estrutura interna de um CLP.





## Arquitetura de um CLP

**Fonte de alimentação:** Converte corrente alternada em contínua para alimentar o controlador. Certos modelos de CLPs são projetados para operarem com uma tensão de alimentação de 220 V, outros trabalham com tensão de alimentação contínua de 24 V. Caso falte energia, há uma bateria que impede a perda do programa do usuário. Ao retomar a energia, o programa se reinicia.

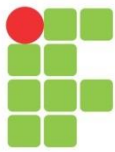
Existem dois tipos de fontes:

- Source: fonte de energia interna ao controlador;
- Sink: fonte de energia externa ao controlador.



## Arquitetura de um CLP

**UCP – Unidade Central de Processamento:** Responsável pela execução do programa do usuário e pela atualização da memória de dados e da memória-imagem das entradas e saídas. Ela recebe os sinais digitais e analógicos dos sensores do campo conectados aos módulos de entrada, e também recebe os comandos via comunicação em rede (quando for o caso). Em seguida executa as operações lógicas, as operações aritméticas e avançadas e atualiza os cartões de saída.



## Arquitetura de um CLP

**Memória EPROM:** Contém programa monitor elaborado pelo fabricante que faz o *start-up* do controlador, armazena dados e gerencia a sequência de operações. Esse tipo de memória não é acessível ao usuário do controlador programável.

**Memória de dados:** Volátil. Encontram-se aqui dados referentes ao processamento do programa do usuário, isto é, uma tabela de valores manipuláveis (variáveis).

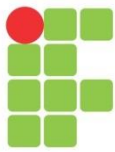


## Arquitetura de um CLP

**Memória do usuário:** Armazena o programa aplicativo do usuário. A CPU processa esse programa e atualiza a memória de dados internos e a de imagem E/S.

A memória possui dois estados:

- RUN: em operação, com varredura cíclica;
- PROG: parado, quando se carrega o programa aplicativo no CLP.



## Arquitetura de um CLP

**Memória-imagem das E/S:** Memória que reproduz o estado dos periféricos de entrada e saída.

A correspondência entre níveis 0 e 1 e níveis de tensão varia conforme a necessidade; por exemplo, pode ocorrer nível 0 para 0 volt e nível 1 para 115 volts CA.



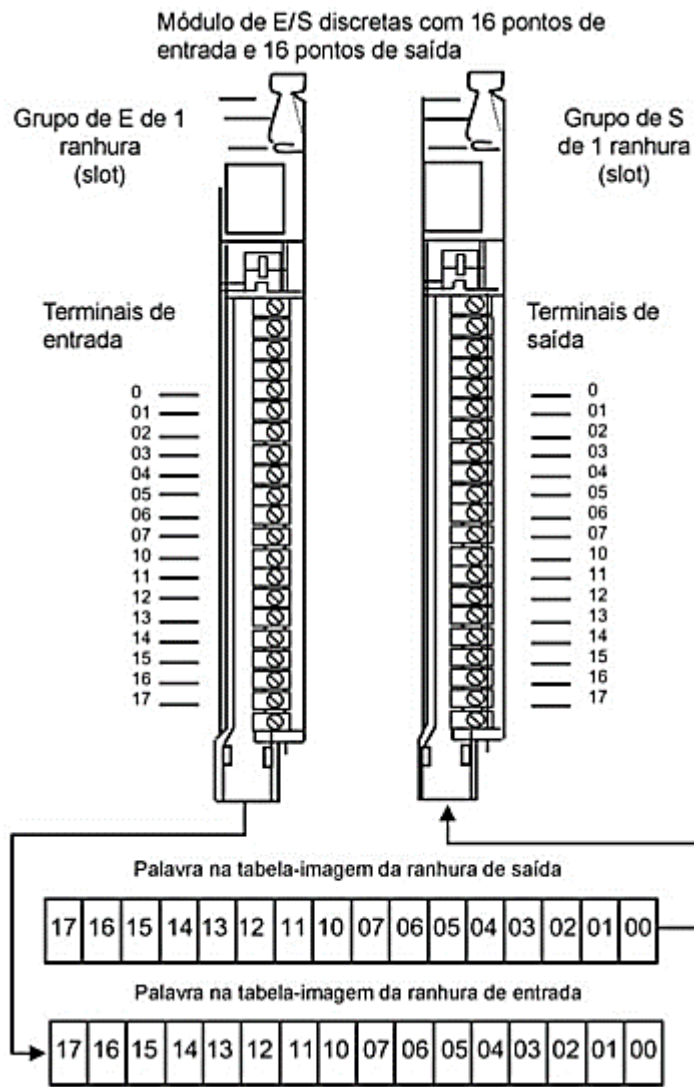
## Arquitetura de um CLP

**Módulos de Entrada e Saída (E/S):** módulos responsáveis pela interface da CPU com o mundo exterior, adaptando os níveis de tensão e corrente e realizando a conversão dos sinais no formato adequado.

Também conhecidos como módulos de I/O. (Input/Output).



# Controladores Lógico Programáveis – CLP's



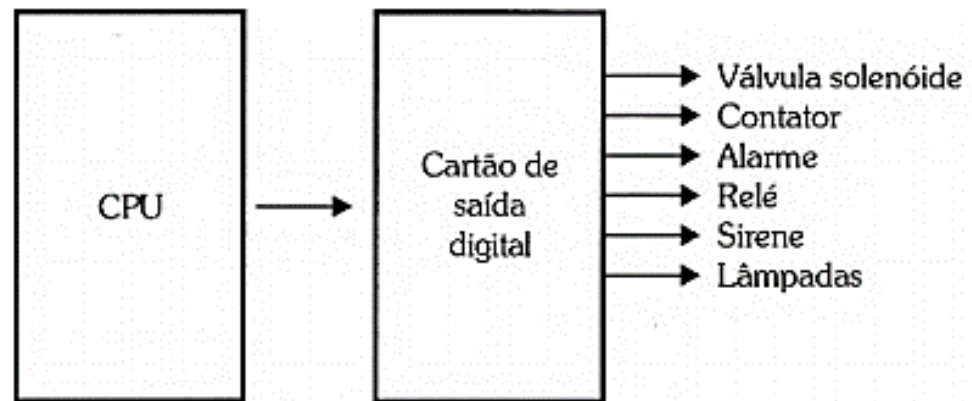
Módulo de E/S.



## Arquitetura de um CLP

**Módulos de Saídas digitais do CLP:** Basicamente, os módulos de saída digitais dos controladores são acionados por três métodos:

- ✓ Saída a relé;
- ✓ Saída a TRIAC;
- ✓ Saída a transistor.



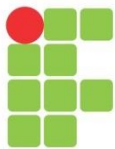




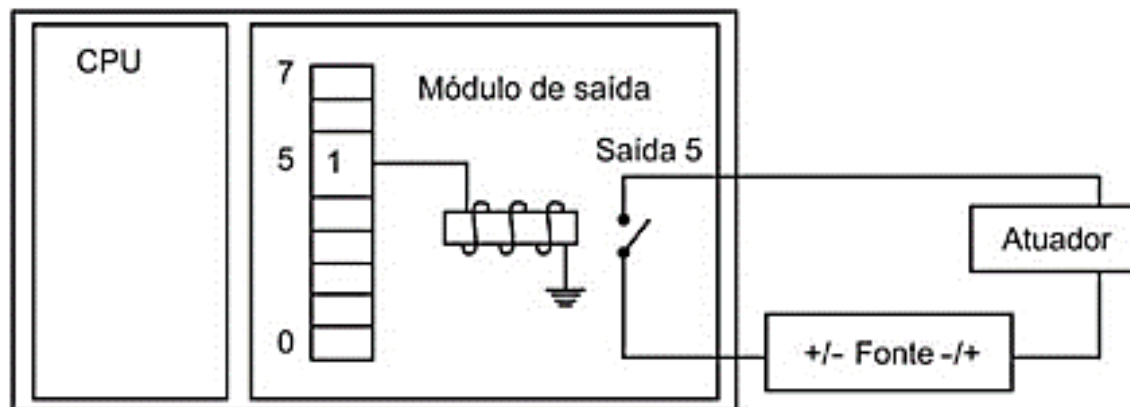
## Arquitetura de um CLP

- ❑ Saída a Relé: quando ativado o endereço da palavra-imagem de saída, um solenoide correspondente a ele é ativado, fechando-se o contato na borneira de saída do controlador.

A grande vantagem desse tipo de saída está na robustez do módulo, que é praticamente imune a qualquer tipo de transiente de rede. Porém, sua vida útil (até 300k) e capacidade de corrente (5 A) são limitados.



## Arquitetura de um CLP

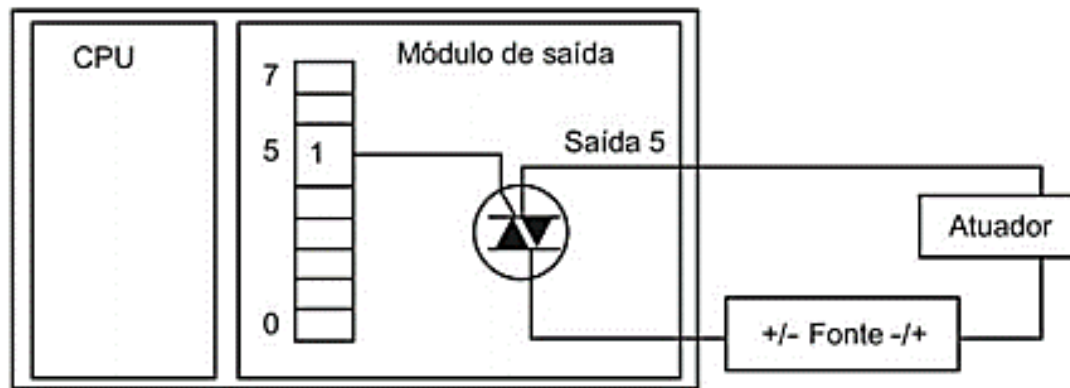


Módulo de saída a relé.



## Arquitetura de um CLP

Saída a Triac: nesse caso, o elemento acionador é um triac (estado sólido). Pela própria característica do componente, esse elemento é utilizado quando a fonte é de corrente alternada. Vida útil (10 M) e capacidade de corrente (1 A).

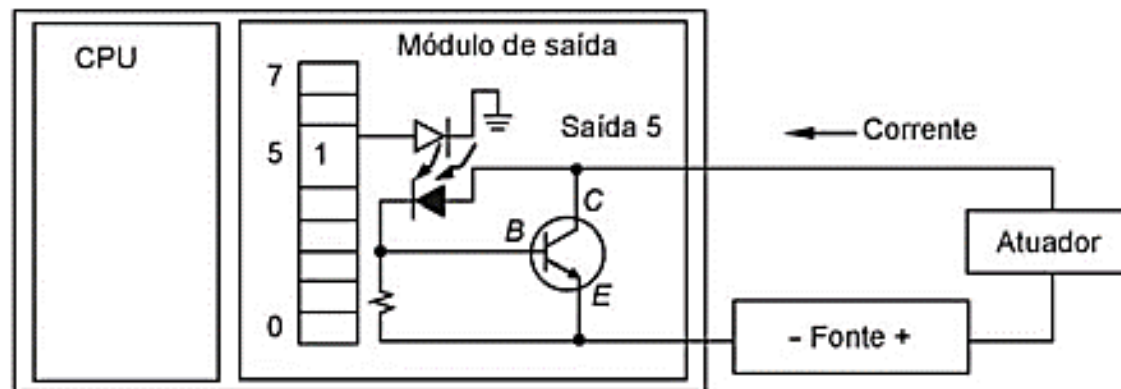


Módulo de saída a triac.



## Arquitetura de um CLP

Saída a Transistor: o elemento acionador pode ser um transistor comum ou do tipo efeito de campo (FET). Esse tipo de módulo, normalmente o mais usado, é recomendado quando são utilizadas fontes em corrente contínua. 10 M, 1 A.

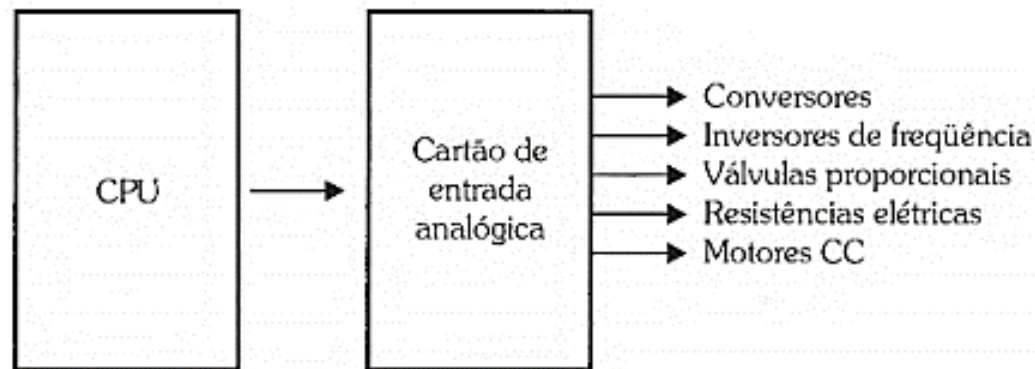


Módulo de saída a transistor.



## Arquitetura de um CLP

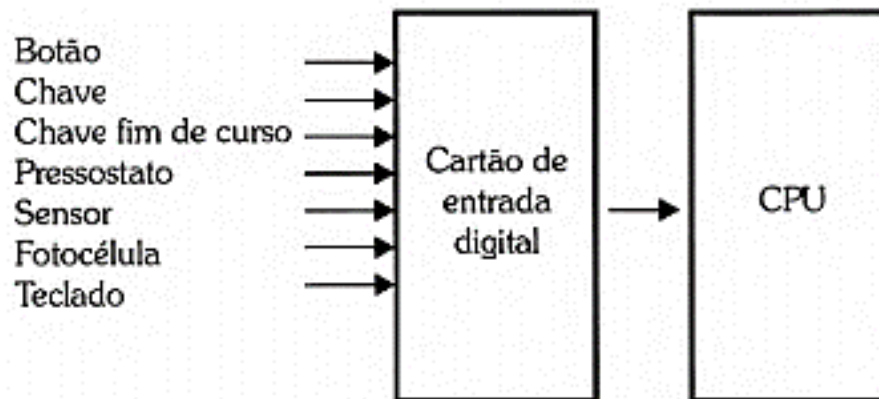
**Módulos de Saídas analógicas do CLP:** Esse tipo de interface converte valores numéricos (binários) em sinais de saída proporcionais em tensão (0 – 10 V) ou corrente (4 – 20 mA).

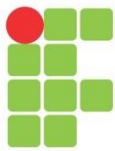




## Arquitetura de um CLP

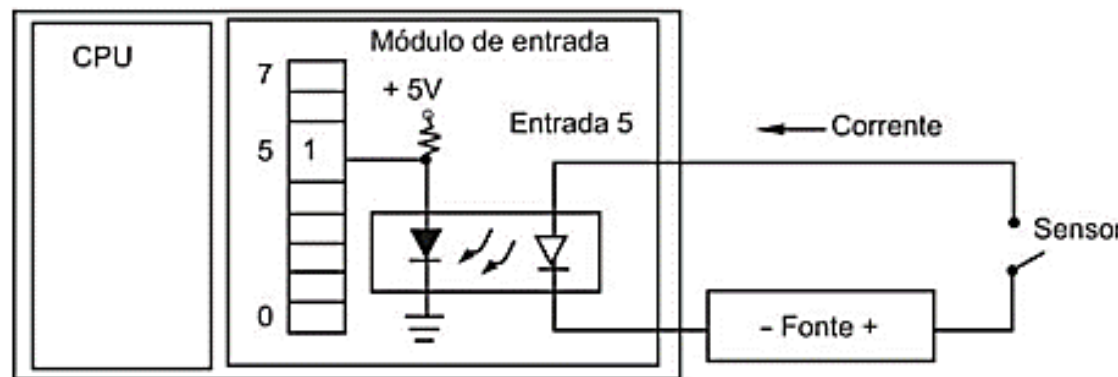
**Módulos de Entradas digitais do CLP:** responsáveis por fazer a interface entre os elementos de sinais de entrada e o CLP. Nesse tipo de módulo, os valores de entradas podem assumir apenas valores discretos (0 ou 1, ON/OFF).





## Arquitetura de um CLP

Esses módulos contêm optoisoladores (ou optoacopladores) em cada um dos circuitos. Quando um circuito externo é fechado através do seu sensor, um diodo emissor de luz (LED) sensibiliza o componente de base, fazendo circular corrente interna no circuito de entrada correspondente.



Módulo de entrada a optoacoplador.



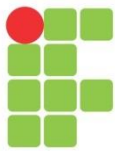
## Arquitetura de um CLP

As entradas digitais do módulo podem ainda ser do tipo **fonte** ou **dreno**. Essa informação é fundamental para escolher o tipo de sensor que fará a interface com o CLP.

## Interface com sensor PNP

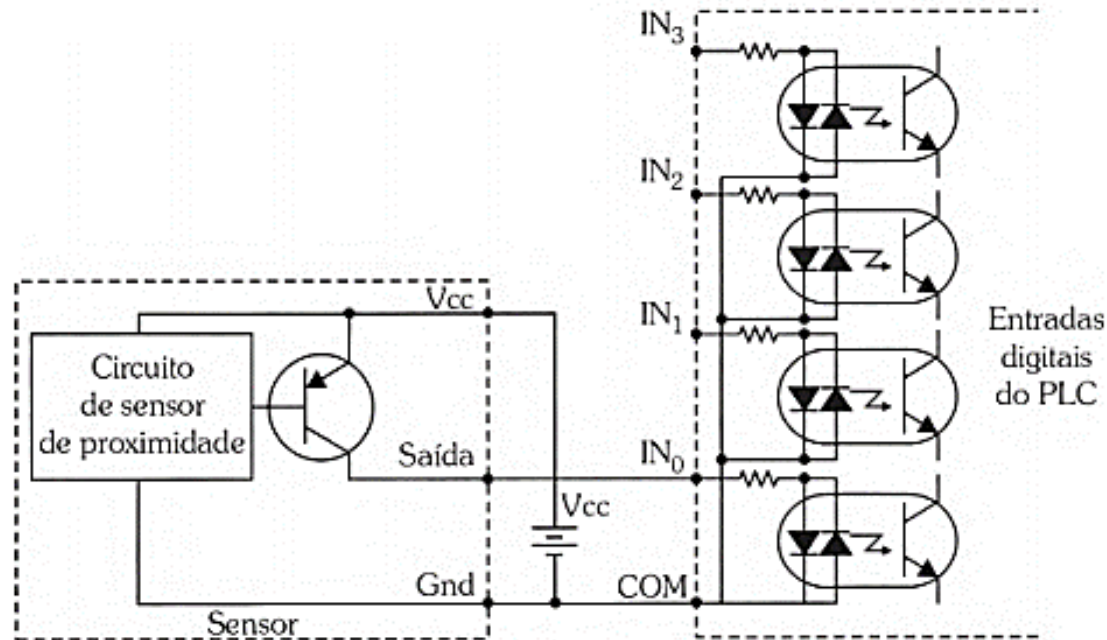
A saída transistorizada PNP tem o emissor conectado a  $V_{cc}$  e o coletor aberto, pode-se dizer que essa interface exibe uma lógica positiva (o dispositivo manda um sinal positivo para indicar que está ativado).





## Arquitetura de um CLP

### Interface com sensor PNP



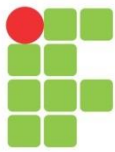
Sensor PNP conectado a uma entrada tipo dreno.



## Arquitetura de um CLP

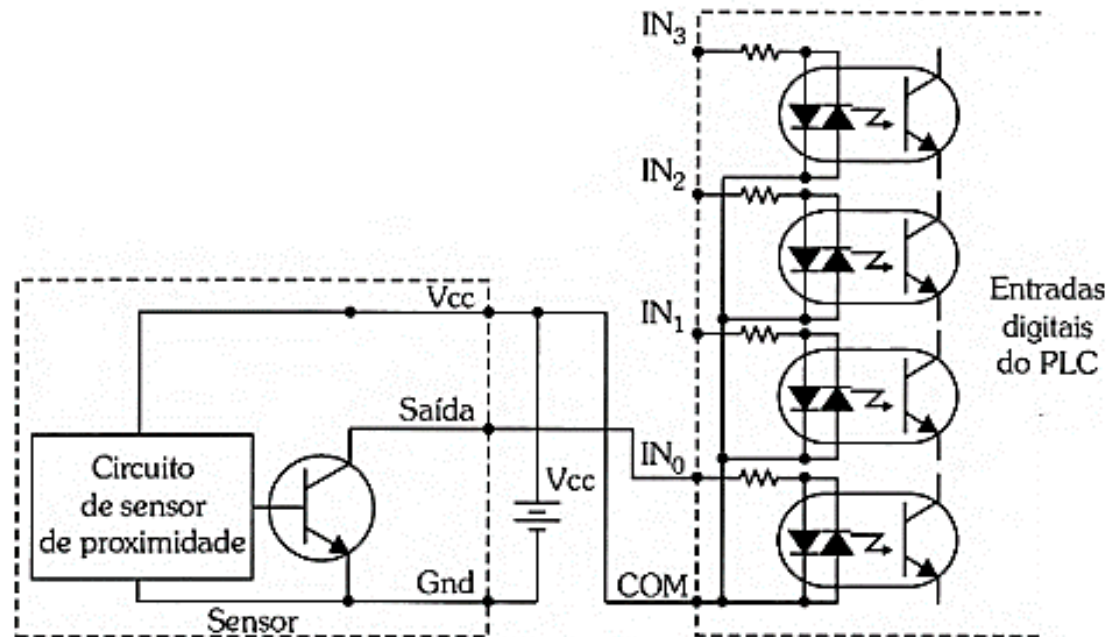
### Interface com sensor NPN

A saída transistorizada NPN por sua vez, tem o emissor conectado a GND e o coletor aberto, exibindo uma lógica negativa (o dispositivo manda um sinal negativo para indicar que está ativado).



## Arquitetura de um CLP

### Interface com sensor NPN



Sensor NPN conectado a uma entrada tipo fonte.



## Arquitetura de um CLP

### Interface de entrada de dados

Os canais de entrada fornecem isolamento e condicionamento de sinais para que sensores e atuadores possam ser conectados diretamente sem necessidade de outro circuito intermediário.

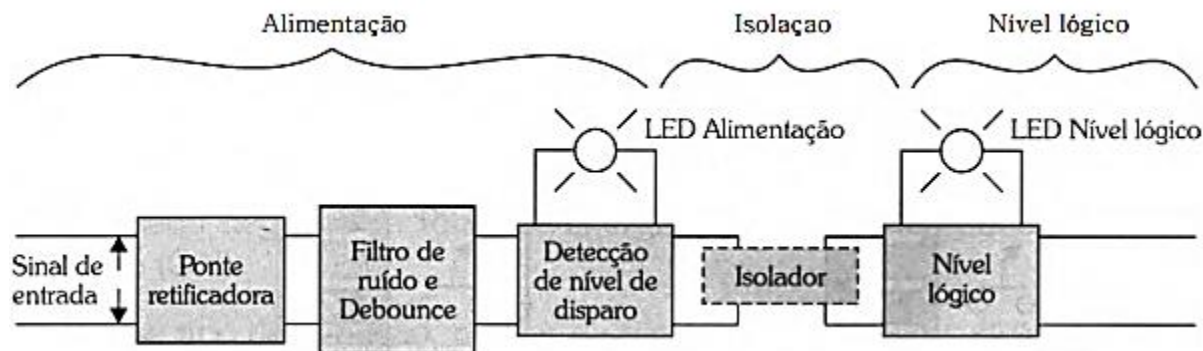
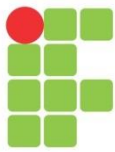
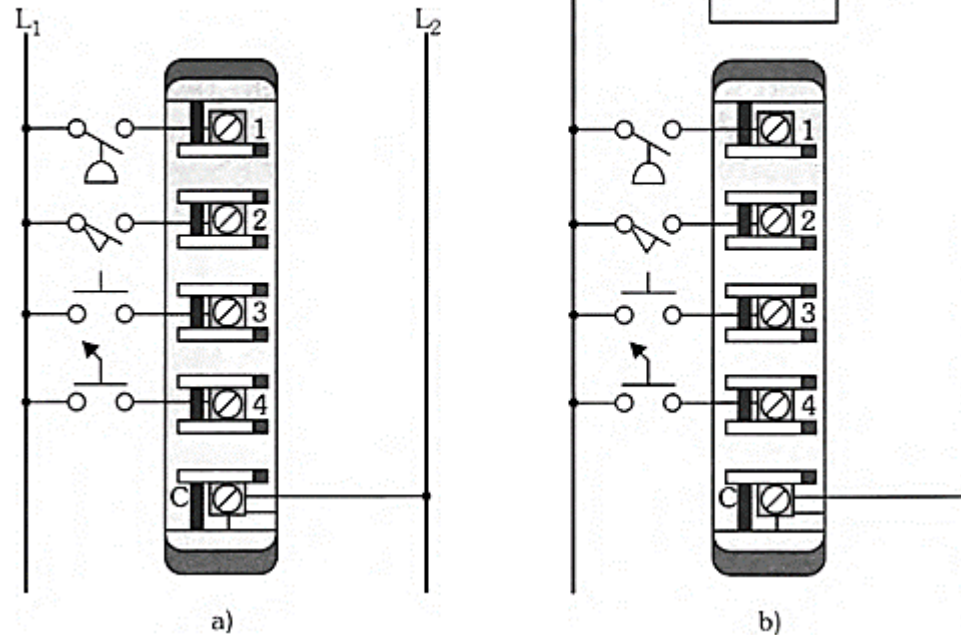


Diagrama de blocos para uma interface típica de entrada de dados.



## Arquitetura de um CLP

### Interface de entrada de dados



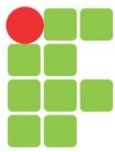
Cartões de entrada digital: a) alimentação em CA; b) alimentação em CC.



## Arquitetura de um CLP

**Módulos de Entradas analógicas do CLP:** permitem que o CLP manipule grandezas analógicas que são enviadas por sensores eletrônicos. Essas grandezas analógicas tratadas por esses módulos são normalmente tensão (0 a 10; 0 a 5; 1 a 5 Vcc; diferenciais: -5 a 5; -10 a 10 Vcc) e corrente (0 a 20; 4 a 20 mA).





## Arquitetura de um CLP

Outra questão importante é a resolução do módulo de entrada analógica, que é medido em bits. Um cartão com maior número de bits permite melhor representação da grandeza analógica.

Exemplo: um cartão de entrada analógica de 0 a 10 Vcc com resolução de 8 bits permite uma resolução de 39,1 mV.

$$Res. = \frac{URL - LRL}{2^N} = \frac{10\text{ V} - 0\text{ V}}{2^8} = 39,1\text{ mV}$$

**N**-> número de bits   **URL** -> Upper Range Limit   **LRL** -> Lower Range Limit



## Endereçamento

Os métodos de endereçamento de entradas e saídas nos CLPs são bastante semelhantes. Palavras ou bits podem ser endereçados. A imagem das entradas e saídas (I/O) é realizada da seguinte maneira.

Por exemplo: I:12/04 ou O:02/06

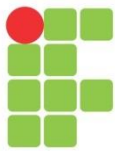
- a) A primeira letra refere-se ao fato de a variável estar indexada como entrada ou como saída, ou seja, I (input) para a palavra de entrada e O (output) para a palavra de saída.



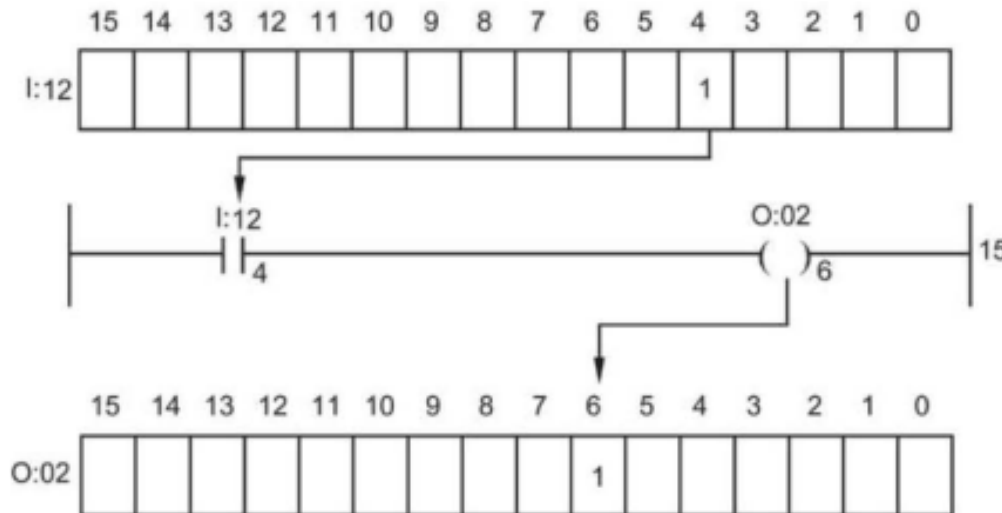


## Endereçamento

- b) Os dois dígitos após o ponto duplo, ":", correspondem à localização que o respectivo módulo de entrada ou saída (pino) ocupa no controlador programável ou na sua expansão. No exemplo, os módulos imagem são respectivamente, 12 para a entrada e 02 para a saída.
- c) Os dois dígitos após a barra inclinada, "/", correspondem ao endereço do bit da imagem da palavra de entrada ou saída.

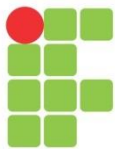


## Endereçamento

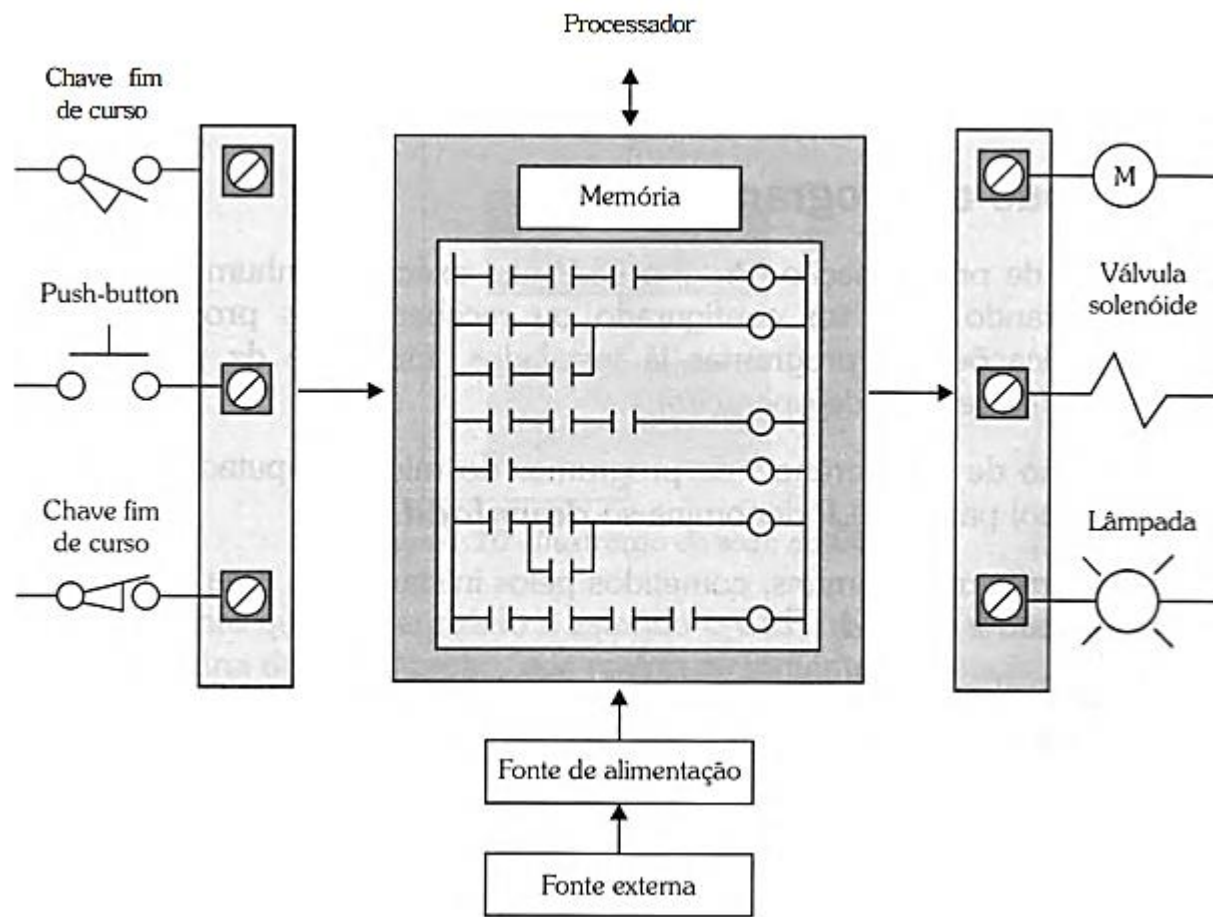


Endereços de entrada e de saída.

Nesse caso, o endereço de entrada corresponde ao bit 4 do pino nº 12 de entrada do controlador, que, ativado pela linha 15 do programa aplicativo, habilita a saída correspondente ao bit 6 do pino nº 02 de saída do CLP.



# Controladores Lógico Programáveis – CLP's



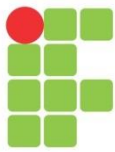
Interação das interfaces de E/S de dados.



## Terminal de Programação

É um periférico que serve de meio de comunicação entre o usuário e o controlador, nas fases de implementação do software aplicativo. Pode ser um computador (PC) ou um dispositivo portátil composto de teclado e display; quando instalado, permite:

- autodiagnostico;
- alterações on-line;
- programação de instruções;
- monitoração;
- gravação e apagamento da memória.



## Ciclo de execução (*scan*) em operação normal (Modo RUN)

Em um ciclo, o CLP realiza as seguintes etapas básicas:

- ✓ atualização das entradas;
- ✓ processamento das instruções do programa;
- ✓ atualização das saídas.

A varredura é processada em ciclo fechado:

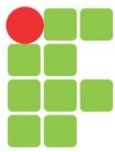




## Ciclo de execução (*scan*) em operação normal (Modo RUN)

O controlador lê a porta de entrada, gravando a informação na imagem de entrada. Em seguida ocorre o processamento, e, por fim, ele copia a imagem de saída na porta de saída.

Para o primeiro ciclo, a imagem das variáveis de entrada é zerada. O processamento desenvolve-se a partir dessa situação, atualizando a palavra e a imagem de saída. Toda vez que a varredura da imagem de entrada se efetua, a palavra de entrada é atualizada.



## Ciclo de execução (*scan*) em operação normal (Modo RUN)

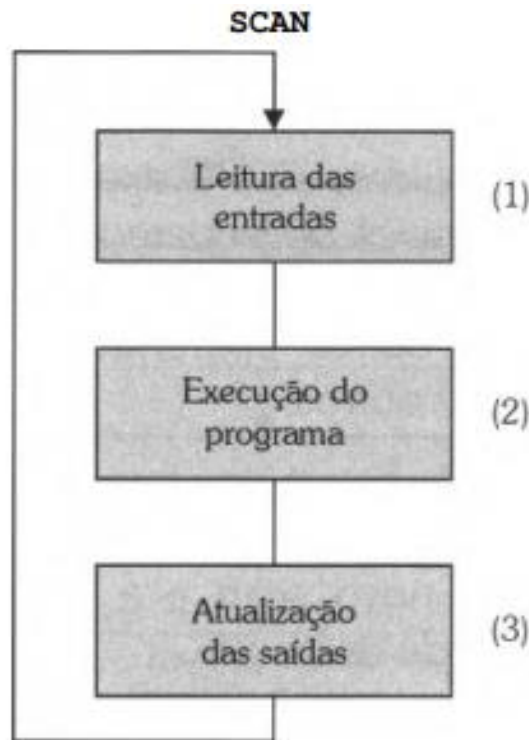


Ilustração do *scan* do CLP.



## Ciclo de execução (*scan*) em operação normal (Modo RUN)

No momento que é energizado e estando o CLP no modo de execução, é executada uma rotina de inicialização, que realiza as seguintes operações:

- ✓ Limpeza da memória-imagem, para operandos não retentivos;
- ✓ Teste de memória RAM;
- ✓ Teste de executabilidade do programa.

Logo após a CPU inicia uma leitura sequencial das instruções em laço fechado (*loop*).





## Modo de programação (Prog)

No modo de programação (Prog) o CLP não executa nenhum programa, isto é, fica aguardando para ser configurado ou receber novos programas ou até receber modificações de programas já instalados. Esse tipo de programação é chamado de *off-line* (fora de operação).

A operação de transferência de programas do microcomputador (ou terminal de programação) para o CLP denomina-se ***download***; já a operação para fazer a coleta de um programa armazenado no CLP para o PC é chamada de ***upload***.



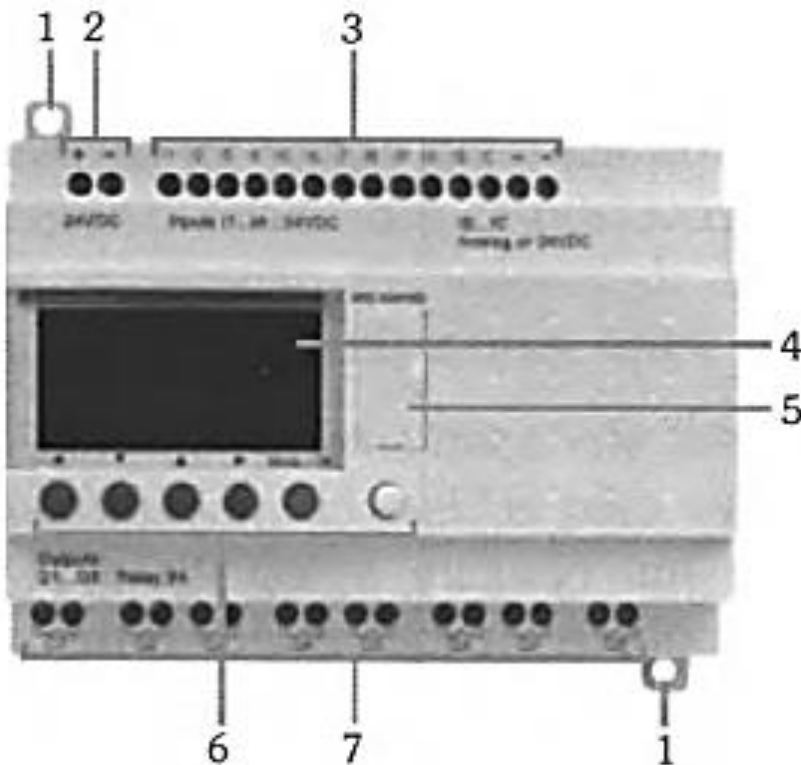
## Modelos de CLP's

**CLP's compactos**: Possuem incorporados em uma única unidade: a fonte de alimentação, a CPU e os módulos de E/S, ficando o usuário com acesso somente aos conectores do sistema E/S. Esse tipo de estrutura normalmente é empregado para CLPs de pequeno porte. Atualmente suportam uma grande variedade de módulos especiais (normalmente vendidos como opcionais), tais como:

- Entradas e saídas analógicas;
- Contadores rápidos;
- Módulos de comunicação;
- Interfaces Homem/Máquina (IHM);
- Expansões de I/O.



## Modelos de CLP's



1. Dois terminais de fixação.
2. Dois terminais para fonte de alimentação.
3. Terminais para conexão das entradas.
4. Display LCD com 4 linhas e 18 caracteres.
5. Slot para cartão de memória ou conexão com PC ou interface de comunicação com modem.
6. Seis botões para programação e entrada de parâmetros.
7. Terminais para conexão das saídas.

CLP compacto Zelio Logic da Schneider Electric.



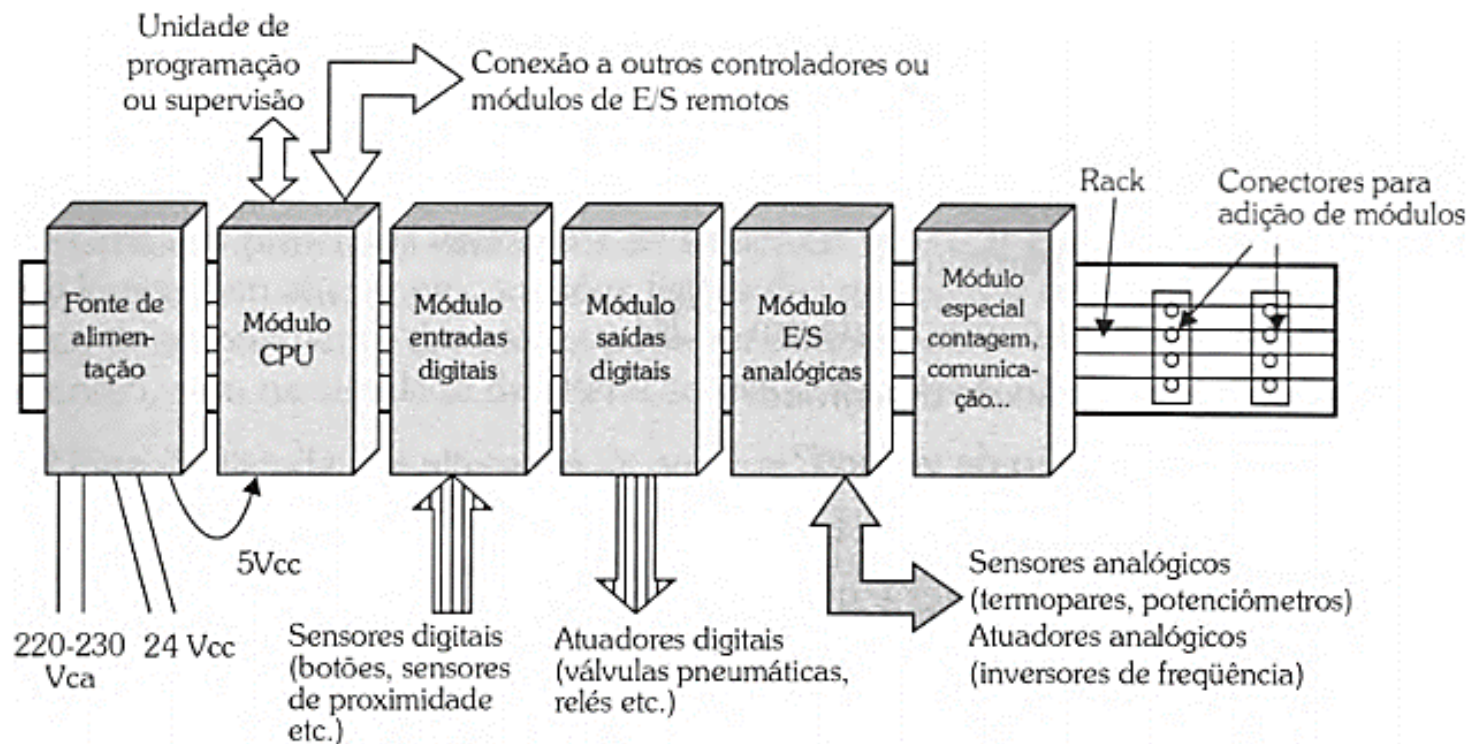
## Modelos de CLP's

**CLP's modulares**: Esses CLPs são compostos por uma estrutura modular, em que cada módulo executa uma determinada função. Podemos ter processador e memória em um único módulo com fonte separada ou então as três partes juntas em um único gabinete.

O sistema de entrada/saída é decomposto em módulos de acordo com suas características. Eles são colocados em posições predefinidas (racks), formando uma configuração de médio e grande porte.



## Modelos de CLP's



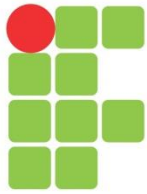
Arquitetura de um CLP modular.



## Modelos de CLP's



CLP modular Simatic S7-300 da Siemens.



INSTITUTO FEDERAL  
MINAS GERAIS  
Campus Avançado Ipatinga

# *Automação Industrial*

## *Linguagens de programação para CLPs*

**Curso: Engenharia Elétrica**  
**Professor Sandro Dornellas**





## Linguagens de programação

Um item fundamental para utilização de um controlador lógico programável é a seleção da linguagem a ser utilizada, a qual depende de diversos fatores, entre eles:

- Disponibilidade da linguagem no CLP.
- Grau de conhecimento do programador.
- Solução a ser implementada.
- Nível da descrição do problema.
- Estrutura do sistema de controle.





## Linguagens de programação

Visando atender aos diversos segmentos da indústria, incluindo seus usuários, e uniformizar as várias metodologias de programação dos controladores industriais, a norma IEC 61131-3 definiu sintática e semanticamente cinco linguagens de programação:

Texto Estruturado (ST)	Textuais
Lista de Instruções (IL)	
Diagrama de Blocos e Funções (FDB)	Gráficas
Linguagem <i>Ladder</i>	
Seqüenciamento Gráfico de Funções (SFC)	

Linguagens segundo a norma IEC 61131-3.



## Linguagens textuais (IL e ST)

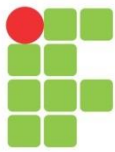
Linguagens textuais como Lista de Instruções e Texto Estruturado consistem em uma sequência de comandos padronizados correspondentes à funções específicas. Correções de programas são complexas e demandam mais tempo de aprendizado.

```
IF Posicao = 10 THEN
  Velocidade := 0 ;
  DesligaMotor := TRUE ;
ELSE IF Posicao = 20
  Velocidade := 50 ;
  DesligaMotor := FALSE ;
ELSE
  Velocidade := 100 ;
  DesligaMotor := TRUE ;
END_IF
```

(a) Exemplo código em ST.

0001	LD	I1	(*LE ENTRADA 1*)
0002	ANDN	I2	(*LÓGICA E NÃO*)
0003	ST	Q1	(*COLOCA RESULTADO NA SAÍDA*)
0004			
0005	LD	I3	
0006	OR	(	IW4
0007	GE	500	)
0008	ST	Q2	
0009			

(b) Exemplo de código em IL.



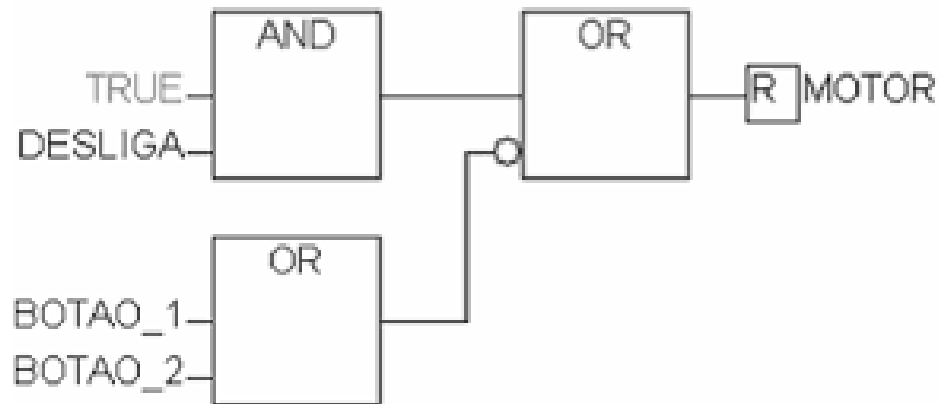
## **FBD - *Function Block Diagram***

Muito popular na Europa, cujos elementos são expressos por blocos interligados utilizando a linguagem de texto estruturado.

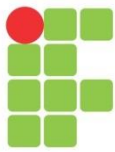
Permite um desenvolvimento hierárquico e modular do software, uma vez que podem ser construídos blocos de funções mais complexos a partir de outros menores e mais simples.



## FBD - *Function Block Diagram*



Exemplo de programação utilizando FBD.



## **SFC/Grafcet – *Sequential Function Chart***

Linguagem gráfica que permite a descrição de ações sequenciais, paralelas e alternativas existentes numa aplicação de controle.

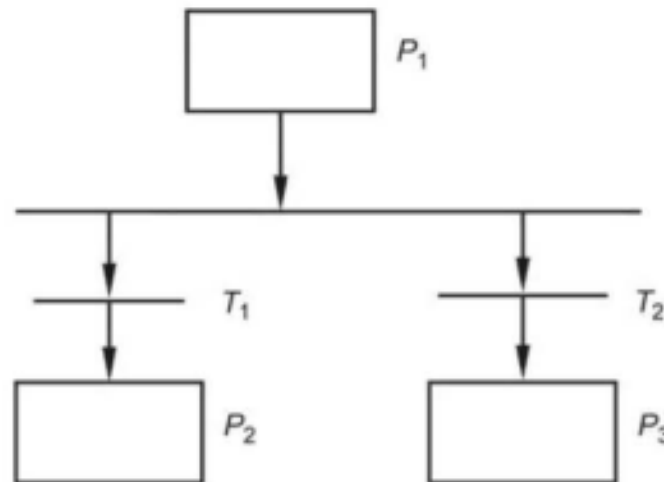
Isso permite uma visualização objetiva e rápida da operação e do desenvolvimento da automação implementada. É uma linguagem gráfica que se originou das Redes de Petri.

O SFC é programado em PASSOS P<sub>1</sub>, P<sub>2</sub>, P<sub>3</sub> (estados operacionais definidos) e TRANSIÇÕES T<sub>1</sub>, T<sub>2</sub> (condições definidas).

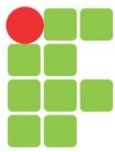


## SFC/Grafcet – *Sequential Function Chart*

Os passos contêm as ações booleanas, e as transições contêm os eventos necessários para autorizar a mudança de um passo a outro.



Exemplo de representação SFC.



## ***LD – Ladder Diagram***

Linguagem gráfica baseada na lógica de relés e contatos elétricos para a realização de circuitos de comandos de acionamentos. Por ser a primeira linguagem utilizada pelos fabricantes, é a mais difundida e encontrada em quase todos os CLPs da atual geração.

Bobinas e contatos são símbolos utilizados nessa linguagem. Os símbolos de contatos programados em uma linha representam as condições que serão avaliadas de acordo com a lógica.



## LD – *Ladder Diagram*

Como resultado determinam o controle de uma saída, que normalmente é representado pelo símbolo de uma bobina.

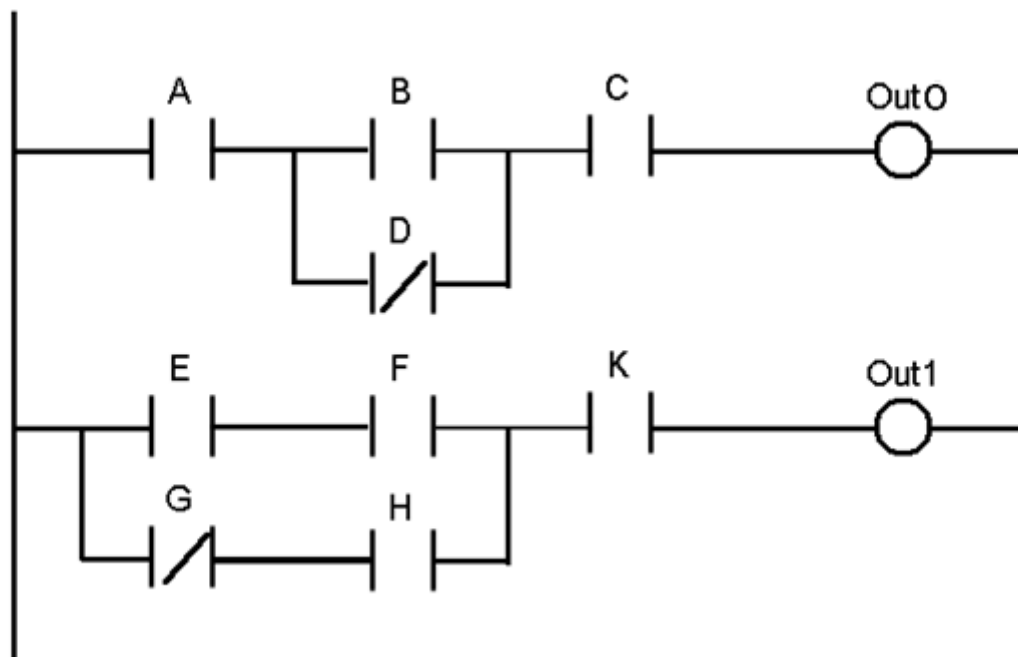
Instrução	Representação
Contato normalmente aberto — NA	-     -
Contato normalmente fechado — NF	-   /   -
Bobina	-( )-
Bobina inversa (acionada, desenergiza)	-(I)-
Bobina set	-(S)-
Bobina reset	-(R)-

Instruções para diagrama *Ladder*.





## LD – *Ladder Diagram*

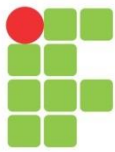


Exemplo de programação em diagrama *Ladder*.



## Vantagens da linguagem *Ladder*

- Possibilidade de uma rápida adaptação do pessoal técnico (semelhança com diagramas elétricos convencionais com lógica a relés);
- Possibilidade de aproveitamento do raciocínio lógico na elaboração de um comando feito com relés;
- Fácil recomposição do diagrama original a partir do programa de aplicação;
- Fácil visualização dos estados das variáveis sobre o diagrama Ladder, permitindo uma rápida depuração e manutenção do software;
- Documentação fácil e clara;



## Vantagens da linguagem *Ladder*

- Símbolos padronizados e mundialmente aceitos pelos fabricantes e usuários;
- Técnica de programação mais difundida e aceita industrialmente.

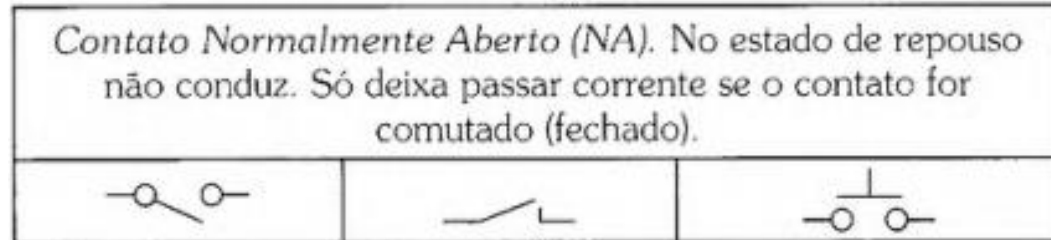
## Desvantagens

- Sua utilização em programas extensos ou com lógicas mais complexas é bastante difícil;
- Programadores não familiarizados com a operação de relés tendem a ter dificuldades com essa linguagem;
- Edição mais lenta.

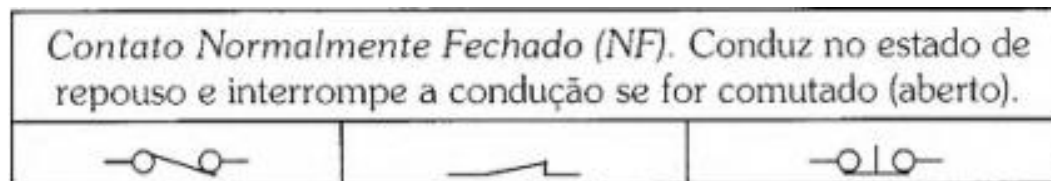


## Lógica de contatos

Um contato na linguagem *Ladder* equivale à uma chave (botoeira) elétrica, a qual pode estar em duas posições: aberta ou fechada.



(a) Contatos elétricos NA.



(b) Contatos elétricos NF.

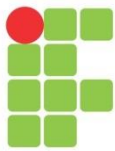


## Lógica de contatos

Os contatos representam os elementos de entrada (sensores, botoeiras, etc) do sistema de controle.









Fabricante	Contato Normalmente Fechado (NF)	Contato Normalmente Aberto (NA)
IEC 61131-3		
Allen-Bradley		
Siemens S7		
GE Fanuc		

Símbolos *Ladder* para contatos, utilizados por alguns fabricantes de CLPs.

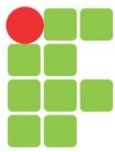


## Bobinas (relés)

Bobinas na linguagem *Ladder* são equivalentes aos relés elétricos, quando circula corrente pelo circuito a bobina é energizada, caso contrário, é deserenergizada. Representam os elementos de saída (atuadores) do sistema de controle.

Fabricante	Bobina	Bobina negada
IEC 61131-3		
Allen-Bradley		Não disponível
GE Fanuc		
Modicon Quantum		
Siemens S7		Não disponível

Símbolos *Ladder* para bobinas, utilizados por alguns fabricantes de CLPs.



## Diagrama de contatos *Ladder*

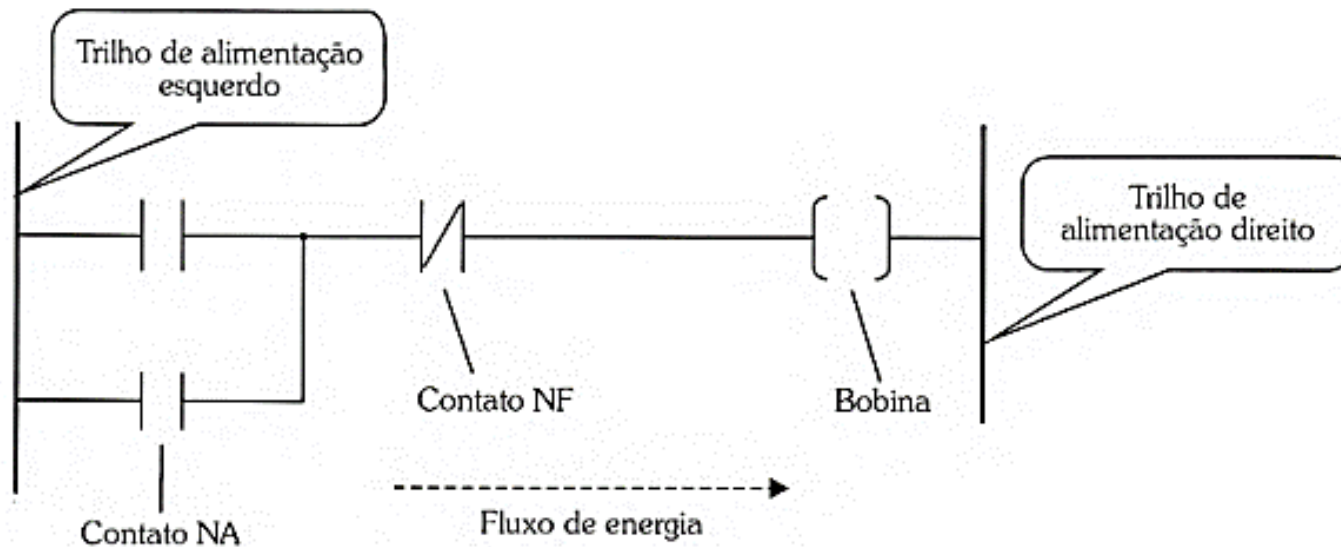
A ideia principal de um programa em linguagem Ladder é controlar o acionamento de saídas, dependendo da combinação lógica dos contatos de entrada.

Um diagrama de contatos é composto de duas barras verticais que representam os polos positivo e negativo de uma bateria.

Uma linha no diagrama é composta de um conjunto de condições de entrada (representado por contatos NA e NF) e uma instrução de saída no final da linha (representada pelo símbolo de uma bobina).

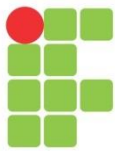


## Diagrama de contatos *Ladder*



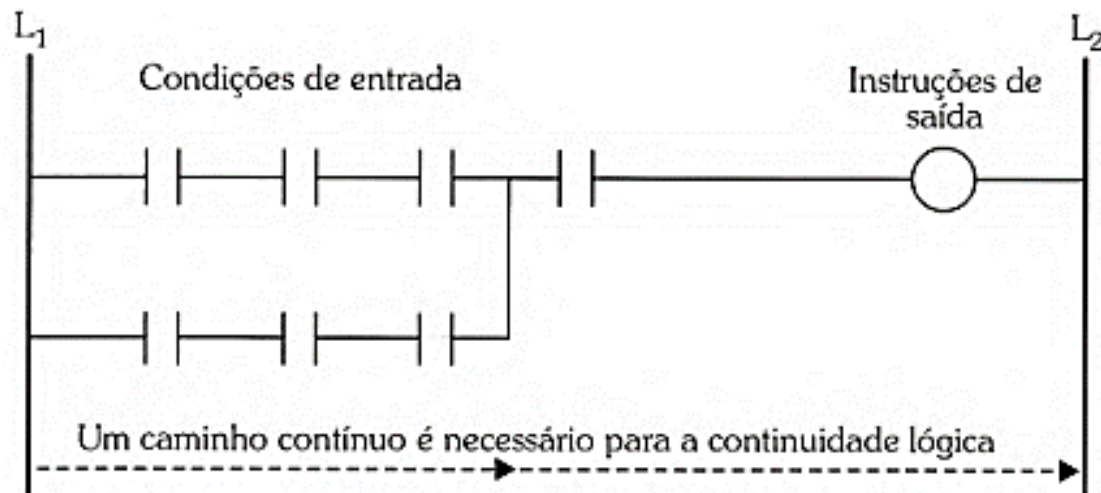
Estrutura típica de uma linha (degrau) em linguagem *Ladder*.





## Diagrama de contatos *Ladder*

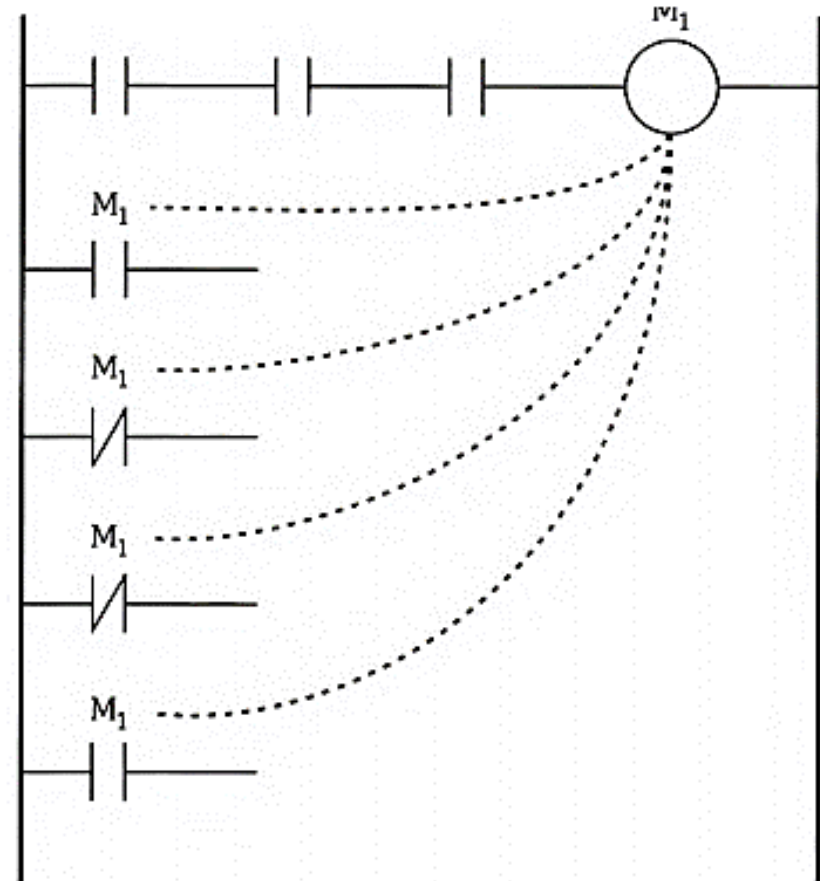
Uma linha é verdadeira, ou seja, energiza uma saída ou um bloco funcional, quando os contatos permitem um fluxo "virtual de eletricidade", ou seja, existe uma continuidade entre a barra da esquerda e a da direita.

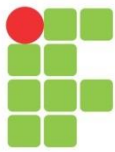




## Repetição de contatos

Os programas em *Ladder* podem ter quantos contatos normalmente abertos ou fechados desejar. Isso significa que um mesmo contato pode ser repetido várias vezes.

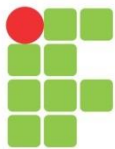




## Repetição de bobinas

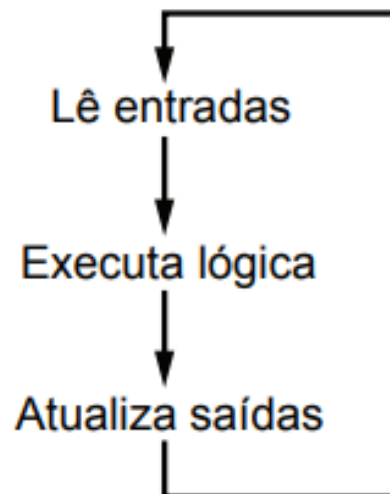
Embora alguns modelos de CLP permitam que uma mesma saída (bobina) seja repetida, é desaconselhável fazê-lo porque a repetição de uma saída em degraus diferentes vai tomar muito confusa a lógica do programa e, por consequência, dificultar o entendimento de quem assumir a manutenção desse programa.

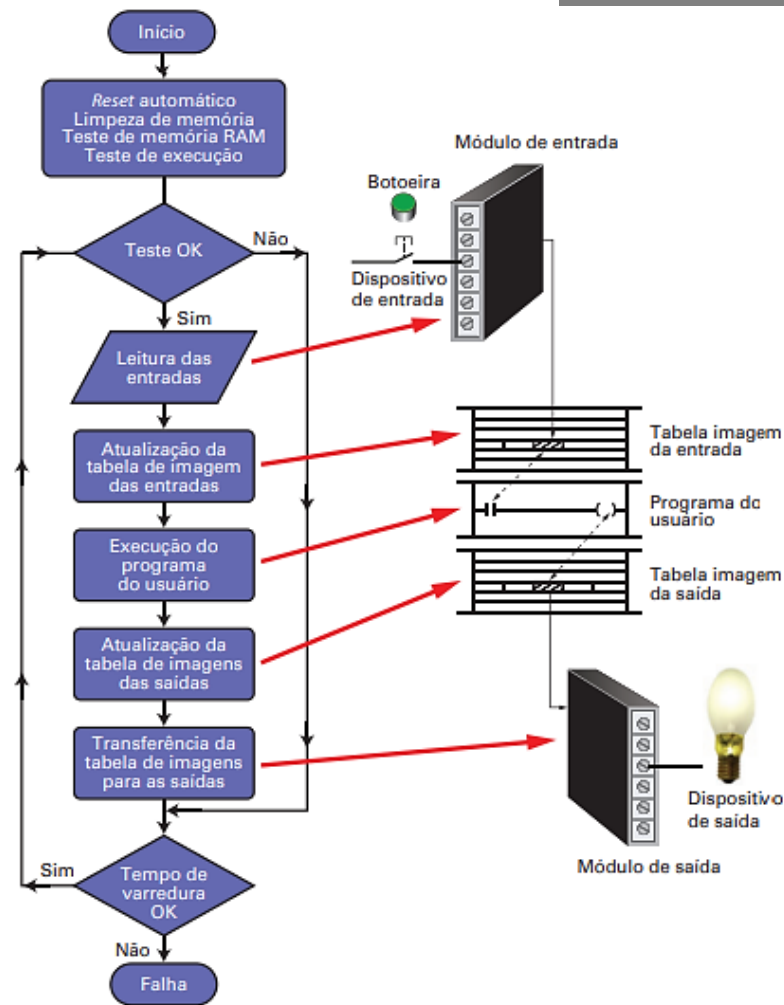
Recomenda-se, portanto, que uma bobina (saída) não seja repetida.



## Princípio de Funcionamento

*Loop infinito:*





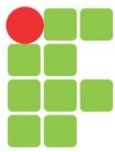


## Princípio de Funcionamento

A CPU realiza a leitura de todos os pontos de entrada e armazena-os na tabela de imagem das entradas.

Cada ponto de entrada corresponde a uma posição de memória específica (um bit de um determinado byte/word).

A tabela de imagem das entradas é acessada pela CPU durante a execução do programa de aplicação.



## Princípio de Funcionamento

Após a execução deste segmento em um determinado scan, a leitura das entradas será realizada apenas no scan seguinte, ou seja, se o status (condição) de um determinado ponto de entrada mudar após a leitura das entradas, ele só terá influência na execução do programa de aplicação no scan seguinte, quando será percebida tal alteração.



## Princípio de Funcionamento

Neste segmento, a CPU executa as instruções do programa de aplicação, que definem a relação entre a condição das entradas e a atuação das saídas, ou seja, definem a lógica de controle a ser realizada.

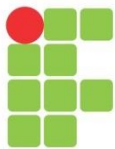
Constrói, assim, uma nova tabela de imagem das saídas, gerada a partir da lógica executada.





## Princípio de Funcionamento

Após a execução do programa de aplicação, o conteúdo da tabela de imagem das saídas, construída de acordo com a lógica executada, é enviado aos pontos de saída correspondentes.



		BIT							
		7	6	5	4	3	2	1	0
B y t e	0	0	0	0	0	0	0	1	0
	1	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0

Ender. de  
Bit

I0.0

I0.1

I0.7

I1.0

Ender. de  
Byte

IB0 = Byte 0

IB1 = Byte 1

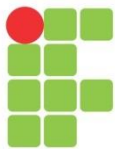
Ender. de  
Word

IW0 = Byte 0 + Byte 1

IW1 = Byte 1 + Byte 2

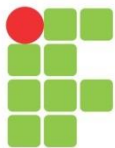
IW2 = Byte 2 + Byte 3

Forma de Endereçamento: I byte.bit



		BIT							
		7	6	5	4	3	2	1	0
B y t e	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0
	2	0	0	0	0	1	0	0	0
	3	0	0	0	0	0	0	0	0

Q2.3



		BIT							
		7	6	5	4	3	2	1	0
B y t e	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	0	0
	7	0	0	0	0	0	0	0	0

MO.0

MB1

MW2

MW1

MW0

MW2

MD0 - Bytes 0, 1, 2 e 3

MD4 - Bytes 4, 5, 6 e 7