

**UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO**  
**CURSO: CIÊNCIA DA COMPUTAÇÃO**

# **Automação e Controle**

## **Aula 07 – Linguagem *Ladder***

**Prof<sup>a</sup> Danielle Casillo**

# A LINGUAGEM *LADDER*

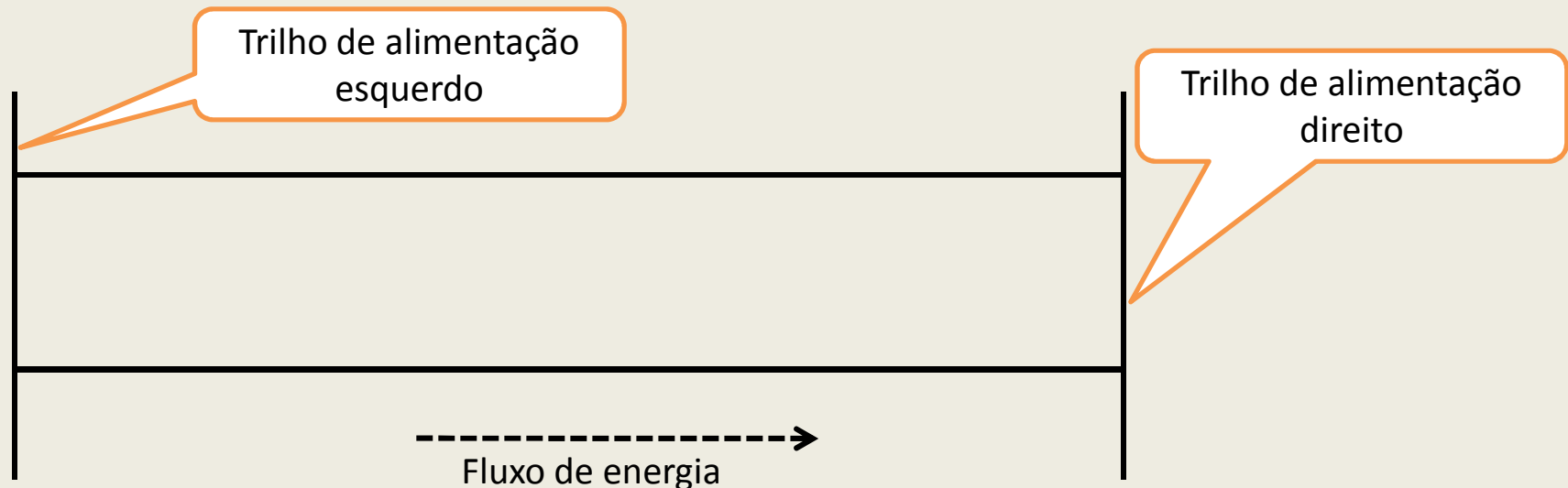
- Foi a primeira que surgiu para programação dos Controladores Lógicos Programáveis.
- Considerando que na época, os técnicos e engenheiros eletricitas eram normalmente os encarregados na manutenção no chão de fábrica, a linguagem *Ladder* deveria ser algo familiar para esses profissionais.
- Assim ela foi desenvolvida com os mesmos conceitos dos diagramas de comandos elétricos que utilizam bobinas e contatos.

# DIAGRAMA DE CONTATOS EM *LADDER*

- A função principal de um programa em linguagem *Ladder* é controlar o acionamento de saídas, dependendo da combinação lógica dos contatos de entrada.
- O diagrama de contatos *Ladder* é uma técnica adotada para descrever uma função lógica utilizando contatos e relés. Sua notação é bastante simples. Um diagrama de contatos é composto de duas barras verticais que representam os pólos positivos e negativo de uma bateria.

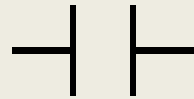
# DIAGRAMA DE CONTATOS EM *LADDER*

- A idéia por trás da linguagem *ladder* é representar graficamente um fluxo de “eletricidade virtual” entre duas barras verticais energizadas. Essa “eletricidade virtual” flui sempre do pólo positivo em direção ao negativo.



# A LINGUAGEM *LADDER*

- Instruções em linguagem *Ladder*



Contato NA



Contato NF







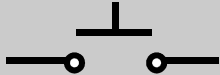



Saída (Relé)

- Símbolos semelhantes aos encontrados nos esquemas elétricos (contatos e bobinas).

# Relação “Dispositivos de Entrada” X “Lógica de Controle”

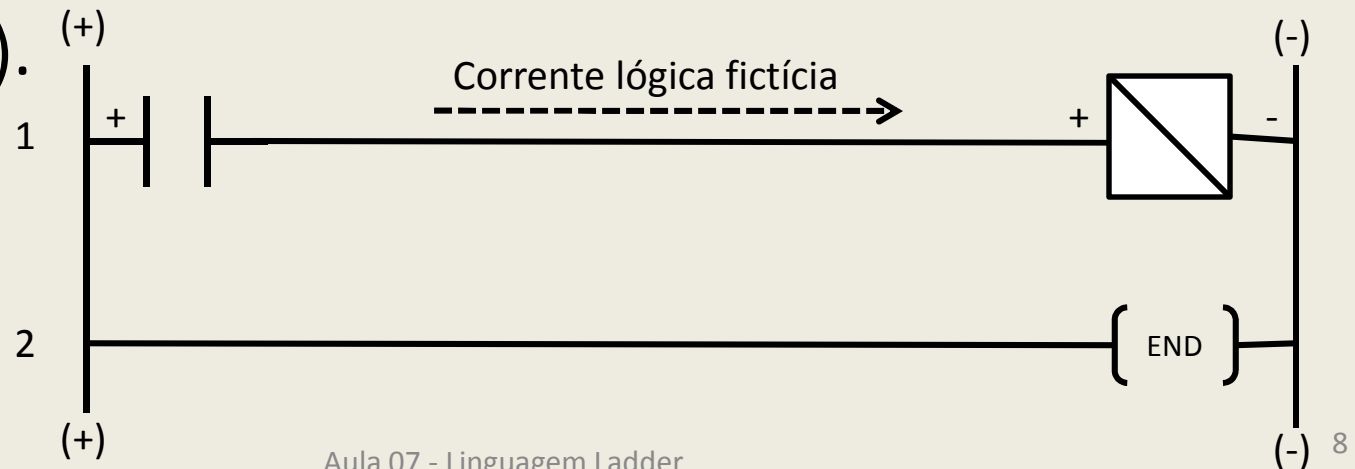
- Esta relação pode causar confusão inicial ao usuário durante a implementação de programas de aplicação para CLP's.
- Normalmente, faz-se a associação direta entre o elemento utilizado na lógica de controle e a condição do dispositivo de entrada, o que gera tal confusão.

# Relação “Dispositivos de Entrada” X “Lógica de Controle”

Dispositivo de Entrada	Tabela Imagem das Entradas	Elemento da Lógica de Controle	Atuação do Contato Lógico	Fluxo de Corrente Lógica
	0		Não	Não
	1		Sim	Sim
	0		Não	Sim
	1		Sim	Não

# CORRENTE LÓGICA FICTÍCIA

- Para que a bobina seja acionada (instrução executada), faz-se necessário “energizá-la logicamente”.
- Supondo que entre as barras verticais que “sustentam” toda a lógica de controle haja uma diferença de potencial (a barra da esquerda com potencial positivo e a barra da direita com potencial negativo).





# Relação “Dispositivos de Entrada” X “Lógica de Controle”

- Independente das características do dispositivo conectado ao módulo de entrada (contato NA ou NF), a lógica de controle pode ser implementada com contatos NA e/ou NF.

# Tipos de Dados

- Além dos pontos de entrada e saída discretas, há outros elementos utilizados na implementação da lógica de controle.
- Embora cada PLC utilize nomenclatura, representação gráfica (linguagem *Ladder*) e forma de endereçamento próprias, a equivalência entre os tipos de dados disponíveis em CPU's distintas proporciona rápida adaptação ao usuário.

# Entradas Digitais: I

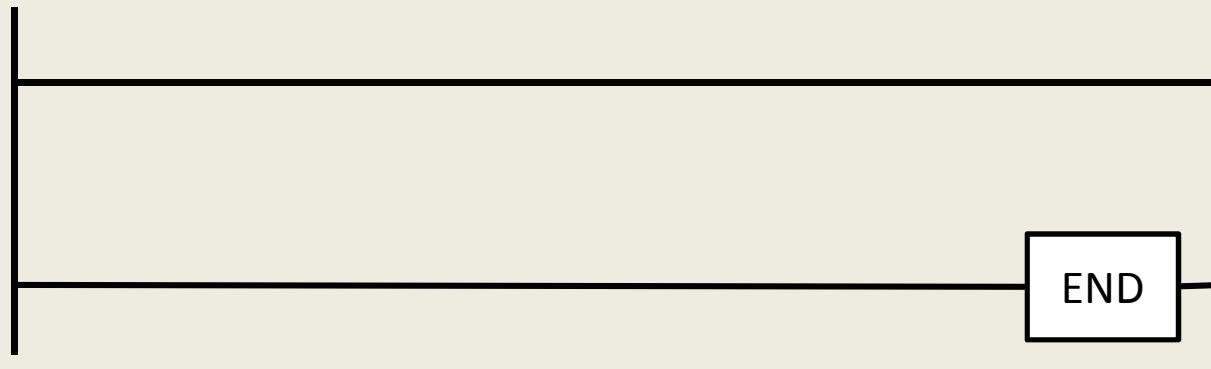
- As entradas digitais são identificadas por I nos controladores da HI Tecnologia. É identificado através de uma numeração sequencial que inicia-se em zero, seu valor final depende da quantidade de placas de I/O ou do modelo do controlador.
- Normalmente, estão associadas às instruções booleanas de entrada (NA e NF).
- Para o PLC eZAP900, temos disponíveis 12 entradas digitais.

# Saídas Digitais: O

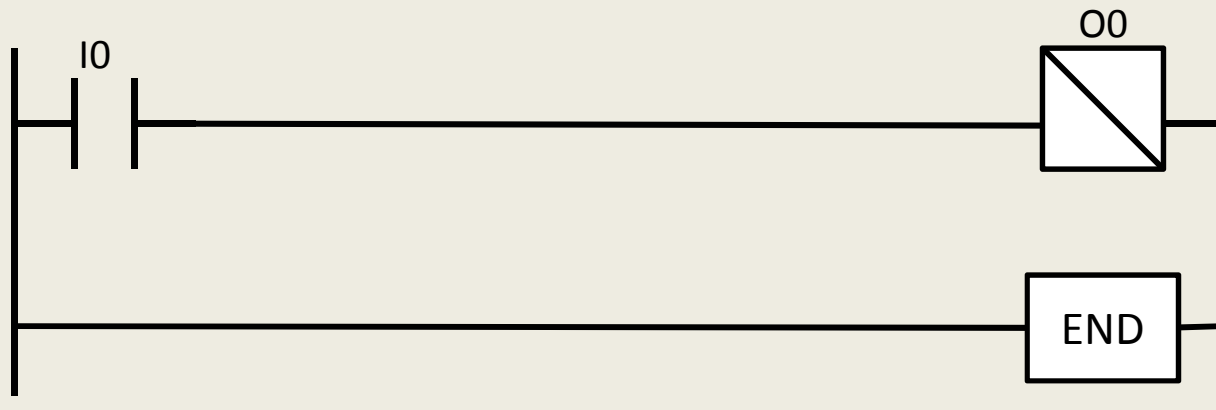
- As saídas digitais são identificadas por **O** nos controladores da HI Tecnologia.
- Normalmente, associadas às instruções booleanas de saída (bobinas), mas podem ser utilizadas também em instruções booleanas de entrada (NA e NF) conforme a necessidade.
- Para o PLC eZAP900, temos disponíveis 12 saídas digitais.

# Instrução END

- Todo programa em *Ladder* deve ter uma Instrução **END**, indicando o seu final. Toda instrução localizada após a instrução **END** não será executada. A não-existência da instrução **END** ocasiona erro.



# PRIMEIRO PROGRAMA: Botão Liga/Desliga



- Acredita-se inicialmente que a saída **O0** estará acionada quando a entrada **I0** estiver aberta, tal qual indicado na linguagem Ladder.
- A verdade é exatamente oposta a esta idéia, ou seja, a saída **O0** só estará acionada quando a entrada **I0** estiver fechada.

# Funções Lógicas em *Ladder*

- As funções lógicas são estudadas em todos e quaisquer elementos. A combinação entre os contatos NA e NF servem de importante orientação para o projetista e programador de circuitos lógicos.

## A. Função “E” (AND)

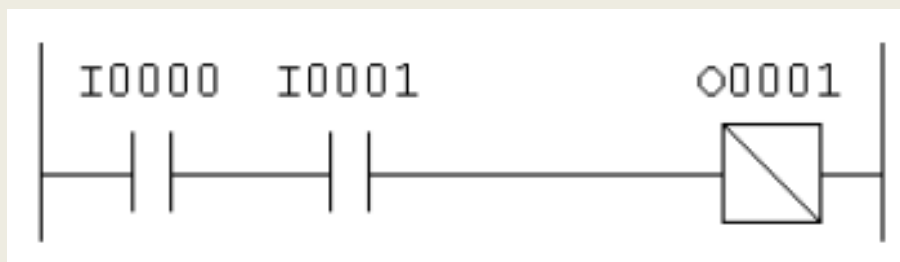


Tabela verdade

I1	I2	Q1
0	0	0
0	1	0
1	0	0
1	1	1

# Funções Lógicas em *Ladder*

## B. Função “OU” (OR)

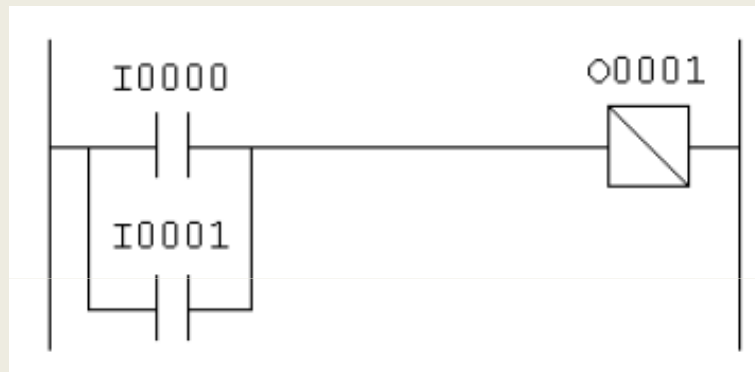
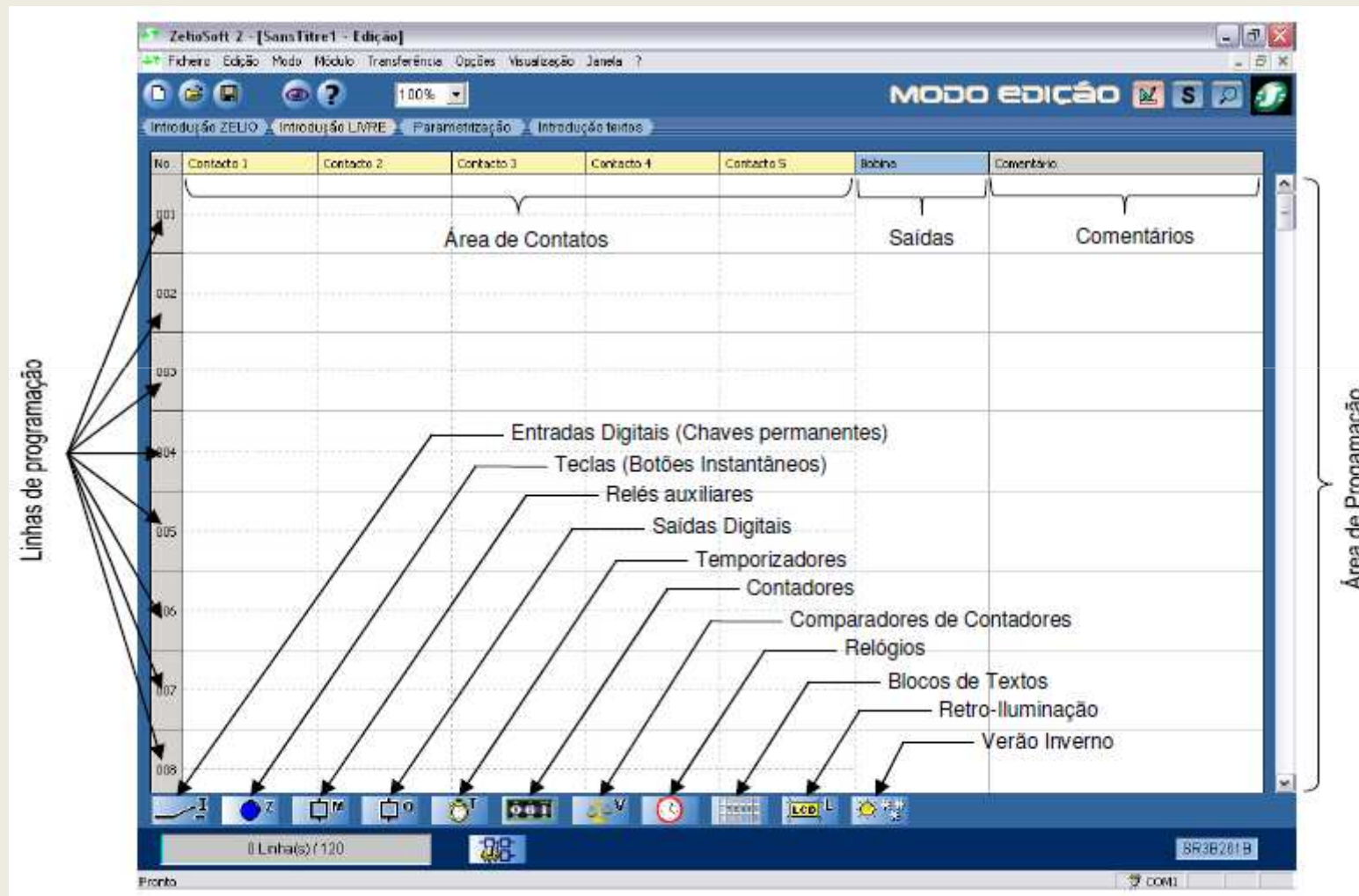


Tabela Verdade

I1	I2	Q1
0	0	0
0	1	1
1	0	1
1	1	1

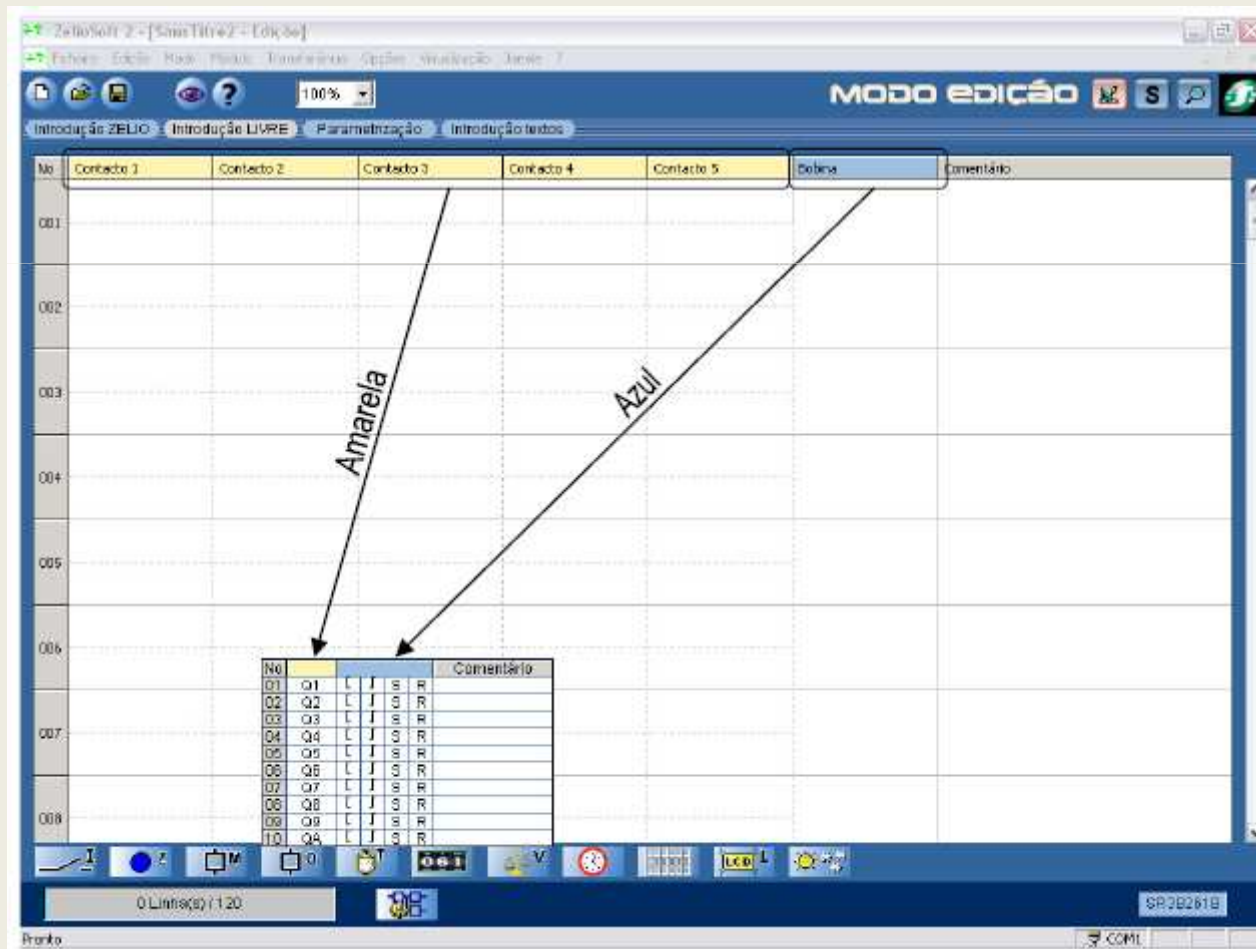


# Ambiente de Programação Zelio Soft 2



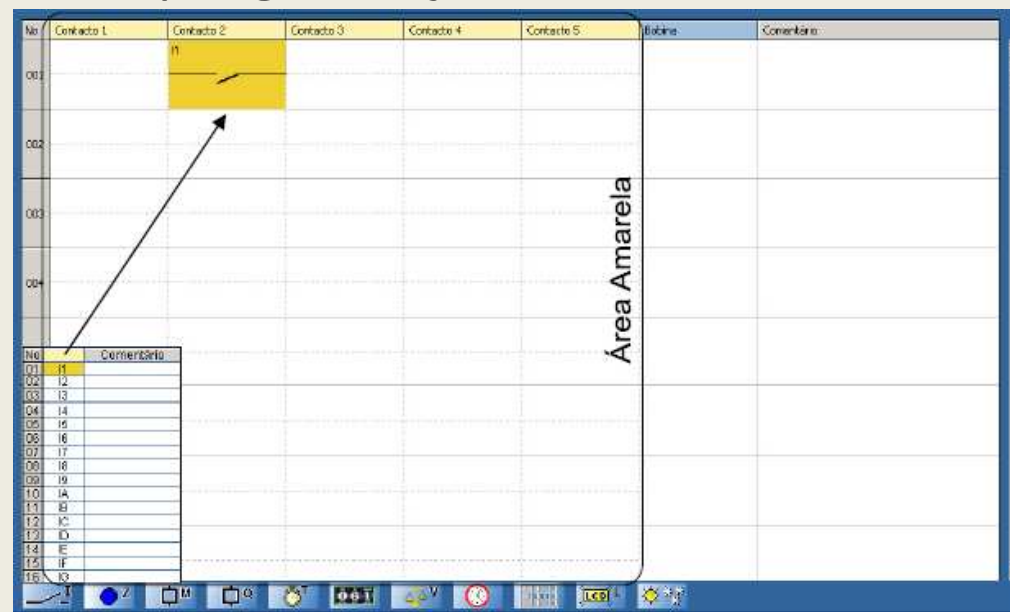
# Ambiente de Programação Zelio Soft 2

- Identificação das áreas Amarelas e azuis:



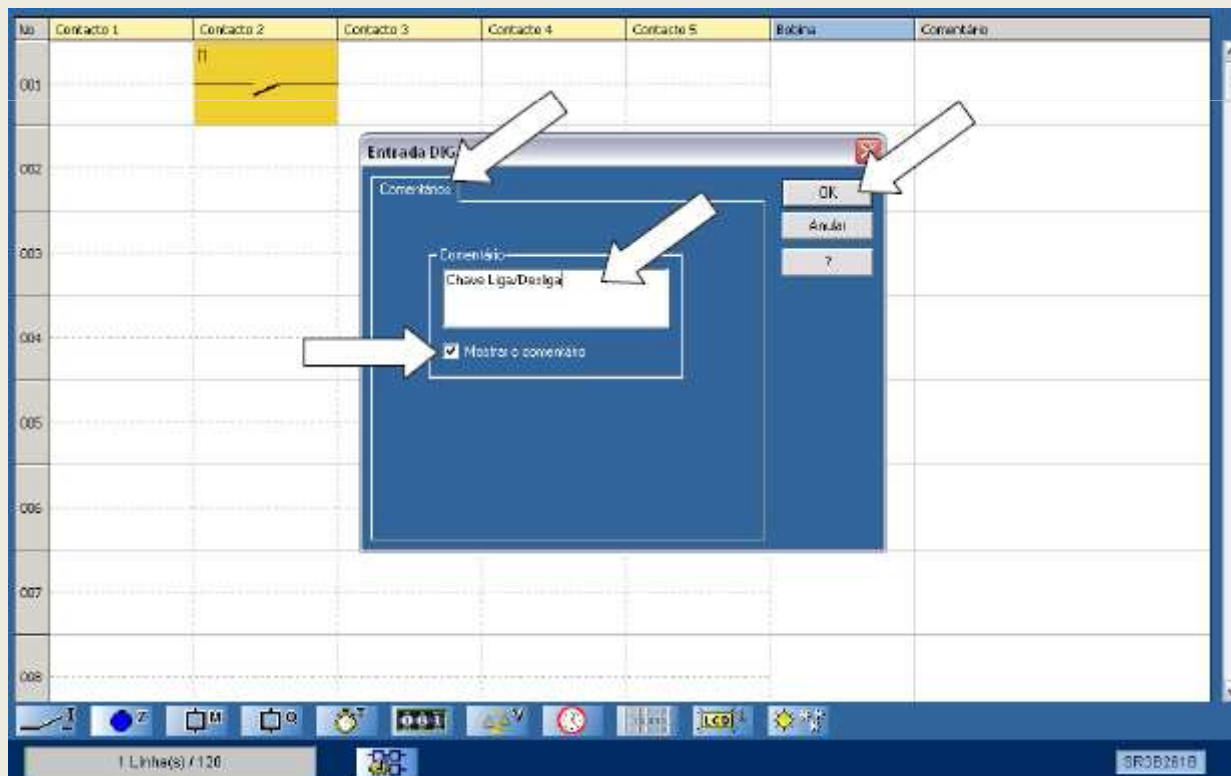
# Exemplo de Programação no Zelio Soft 2

- Acender uma lâmpada
  - Inicialmente Seleccionamos um contato de chave permanente (Entrada Digital), para isso, selecione o botão do elemento Entrada Digital indicado na figura, clique na descrição “I1” e arraste para a área amarela da área de programação:



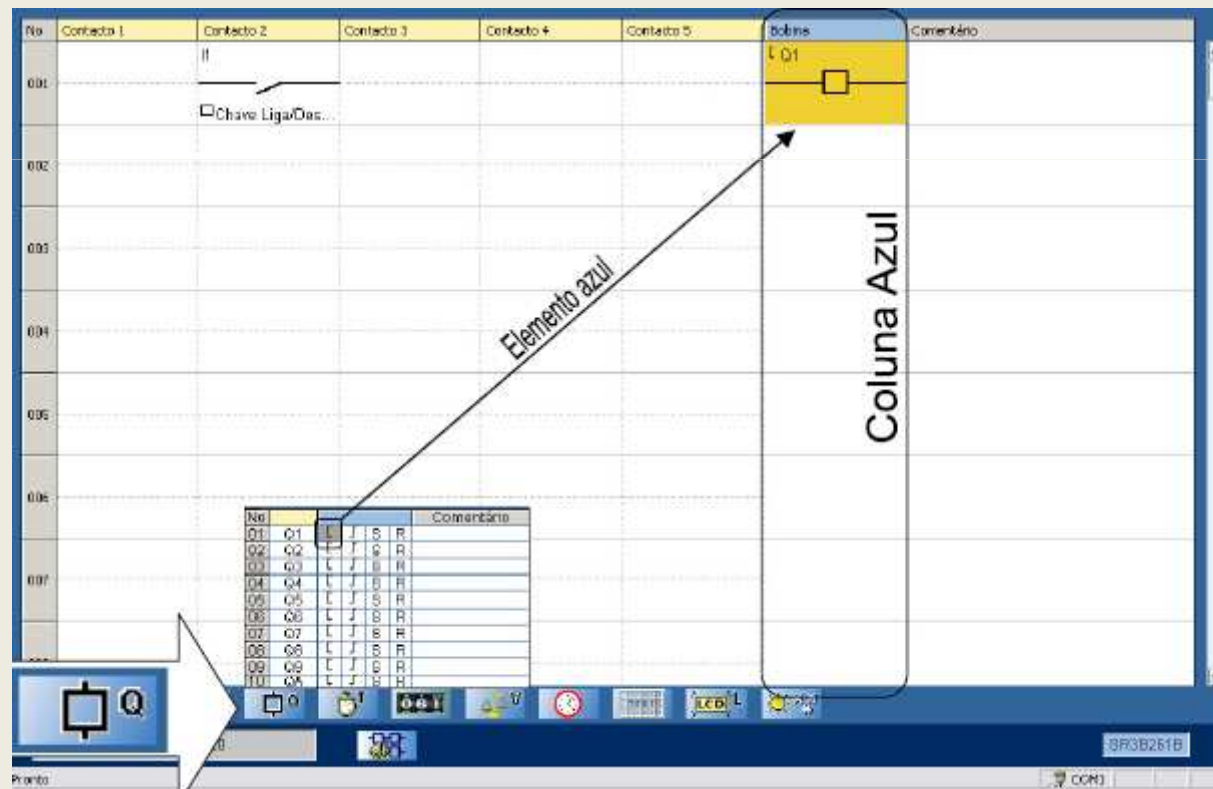
# Exemplo de Programação no Zelio Soft 2

- Acender uma lâmpada
  - Para adicionar um nome ao contato, clique duas vezes sobre o contato.



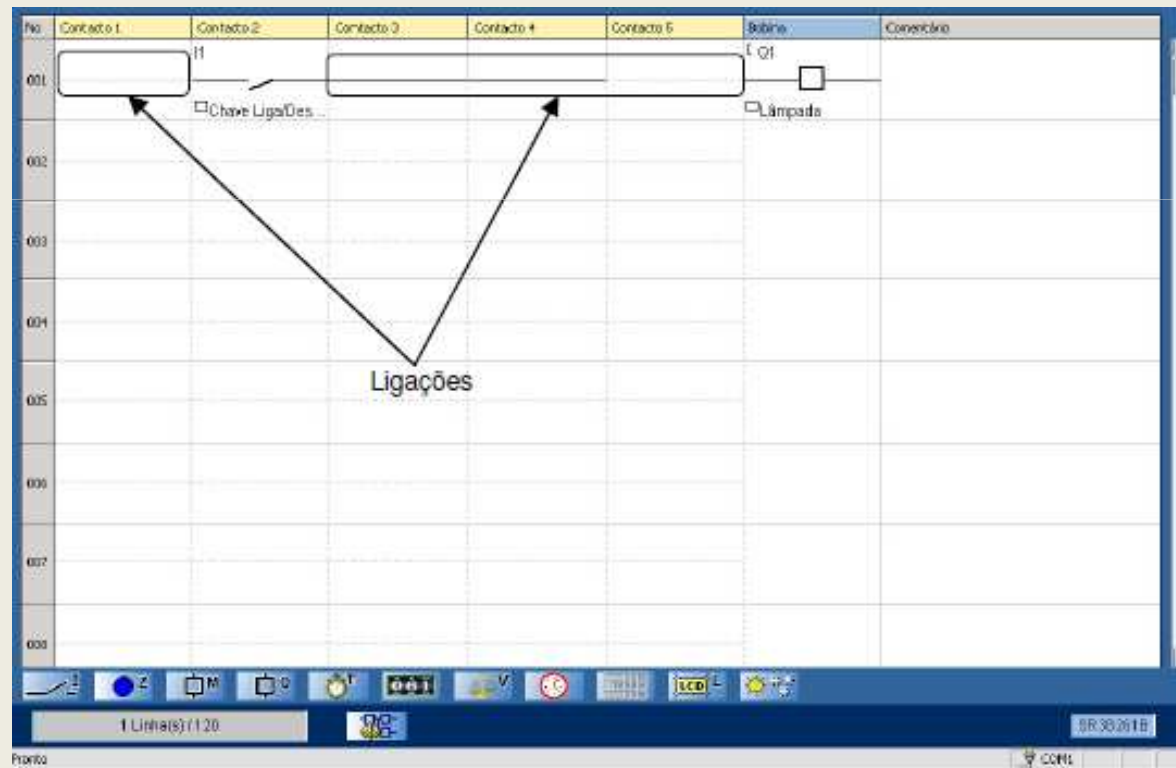
# Exemplo de Programação no Zelio Soft 2

- Acender uma lâmpada
  - Adicione ao programa uma Saída Digital



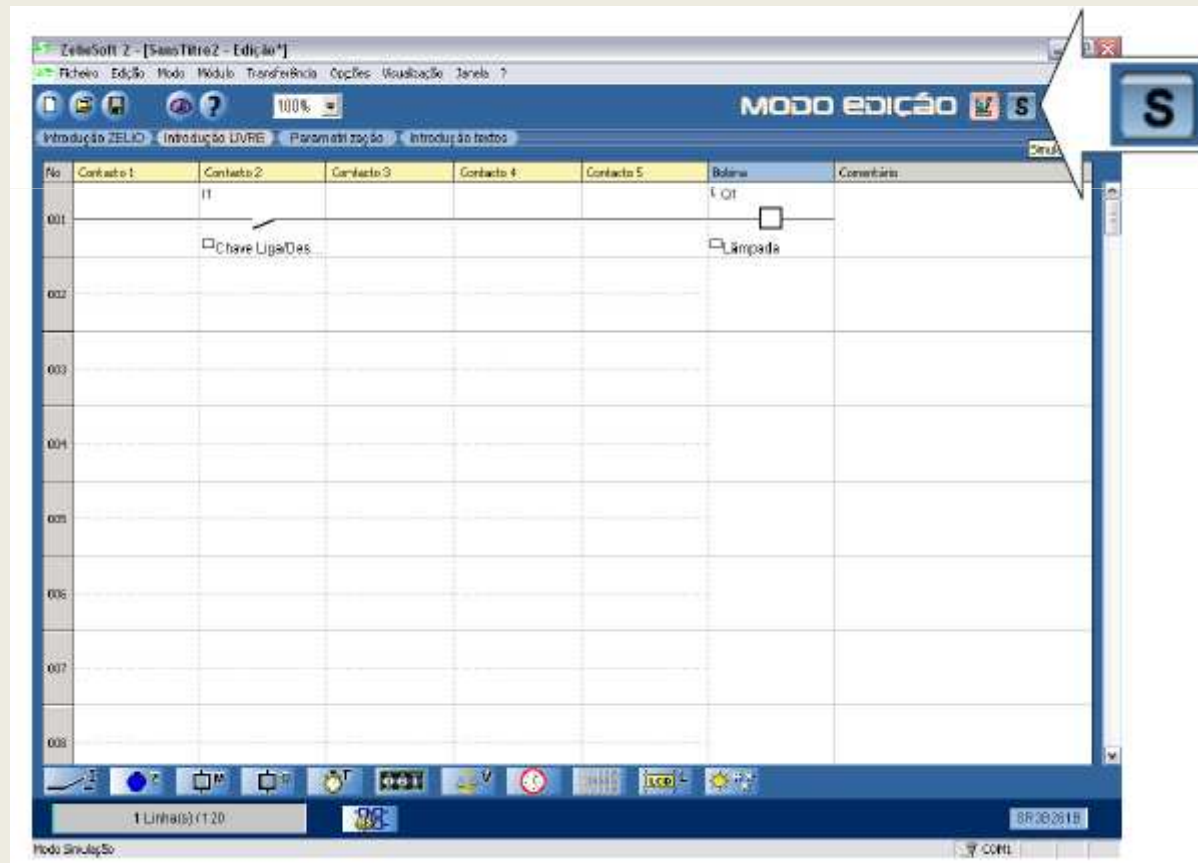
# Exemplo de Programação no Zelio Soft 2

- Acender uma lâmpada
- Feche as ligações.



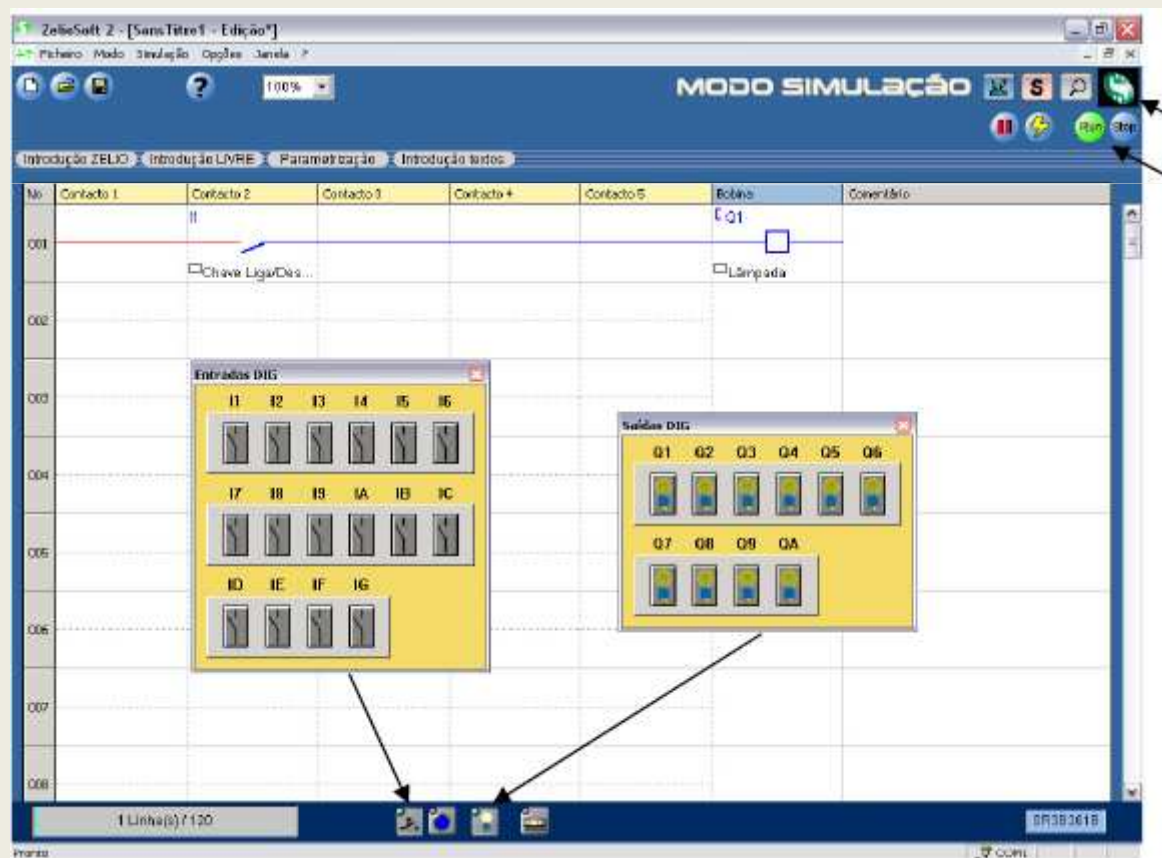
# Exemplo de Programação no Zelio Soft 2

- Acender uma lâmpada
  - Feito isso, simule conforme indicado na figura.



# Exemplo de Programação no Zelio Soft 2

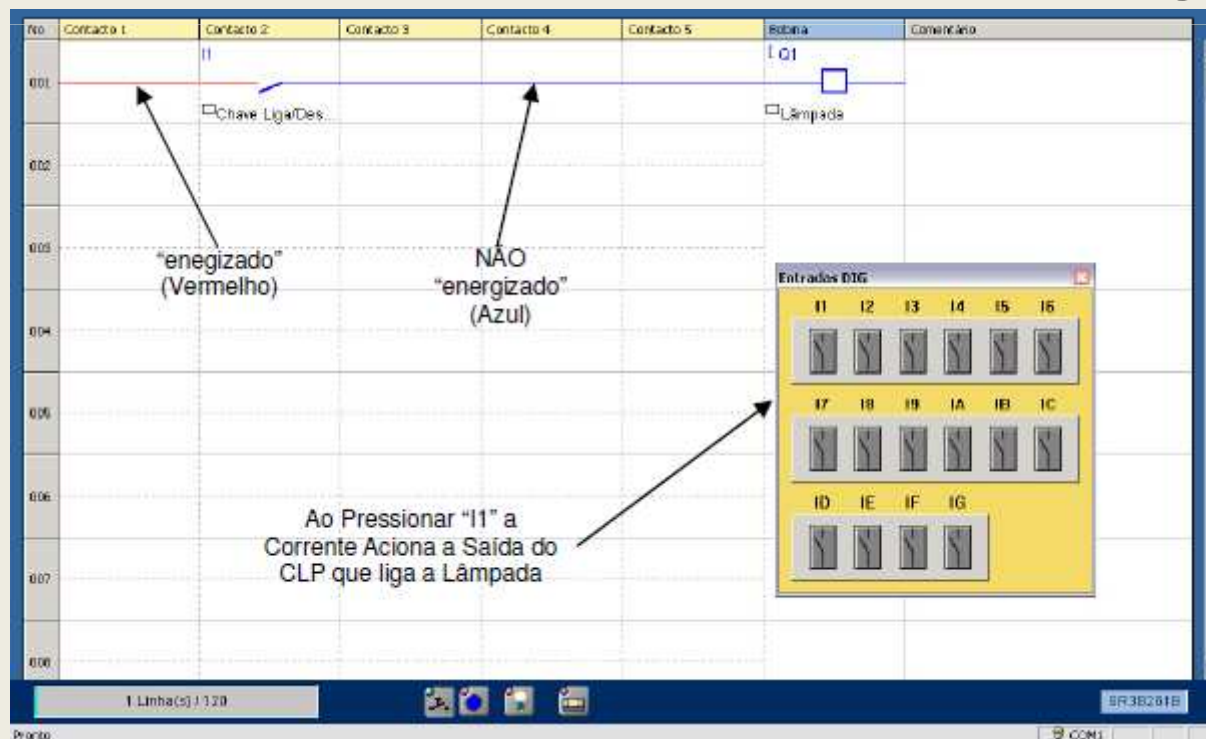
- Acender uma lâmpada
  - Para iniciar a simulação, clique na tecla “Run”.





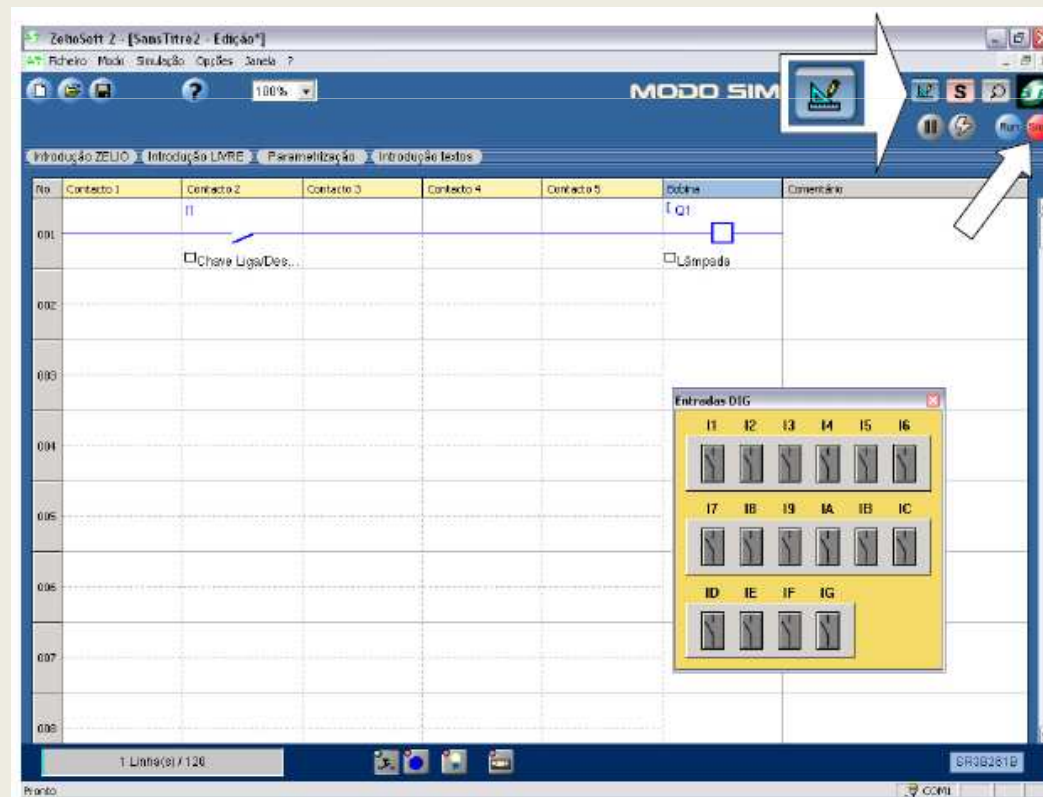
# Exemplo de Programação no Zelio Soft 2

- Acender uma lâmpada
  - Quando a simulação estiver em andamento, observe que o local identificado em vermelho, mostra onde o circuito está **“energizado”**. Os locais identificados em azul, mostra onde o circuito **NÃO** está **“energizado”**.



# Exemplo de Programação no Zelio Soft 2

- Acender uma lâmpada
- Para parar a simulação clique na tecla “Stop”. Para sair do modo de simulação e voltar ao modo de programação, basta clicar na tecla Edição.

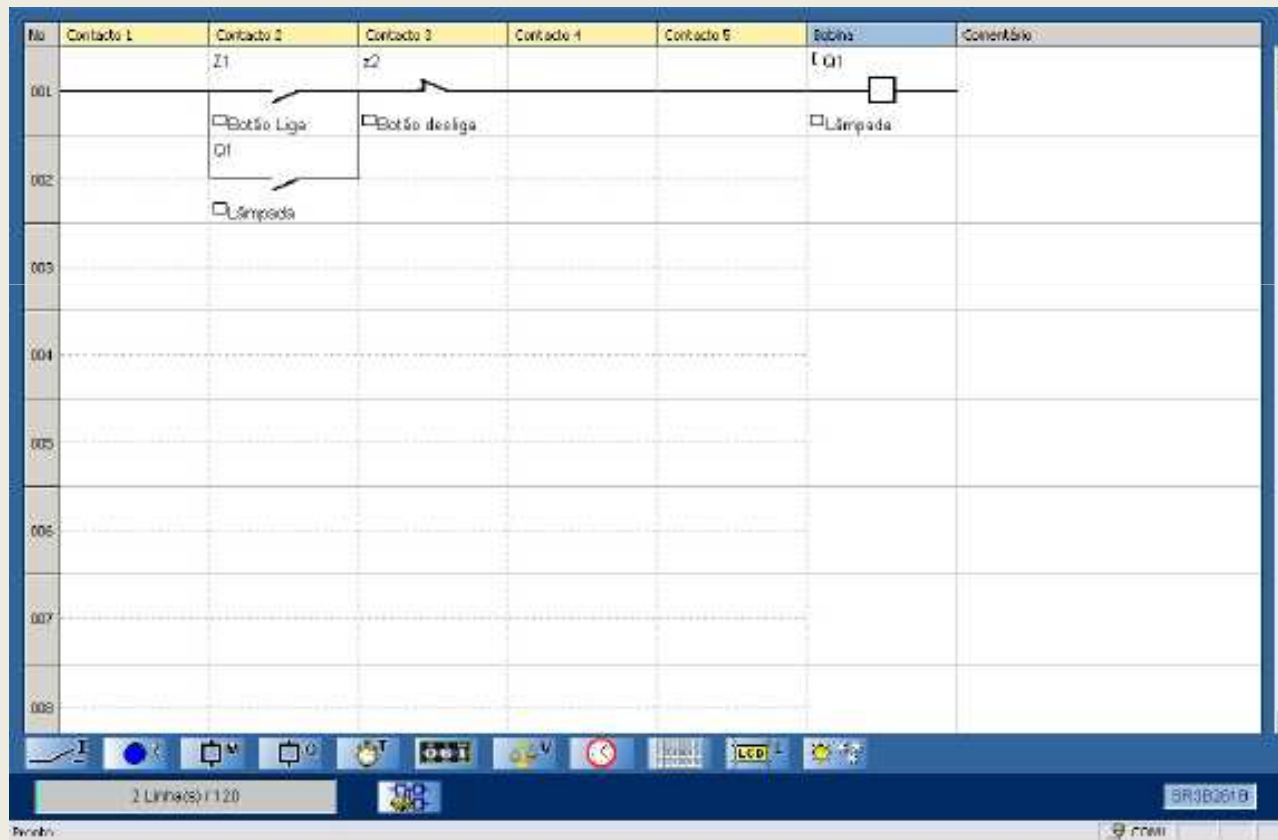


# Exemplo de Programação no Zelio Soft 2

- Acender uma lâmpada
  - Salve seu programa na pasta “Meus documentos”, crie uma nova pasta com seu nome e salve o seu programa nesta pasta.
  - **Atenção:** Para melhor organização os arquivos, trabalhem sempre no mesmo computador.
  - **Aproveitando:** Lembrem-se de desligar os computadores quando sair e também de deixar a bancada organizada. Por último, desligue o No-break.

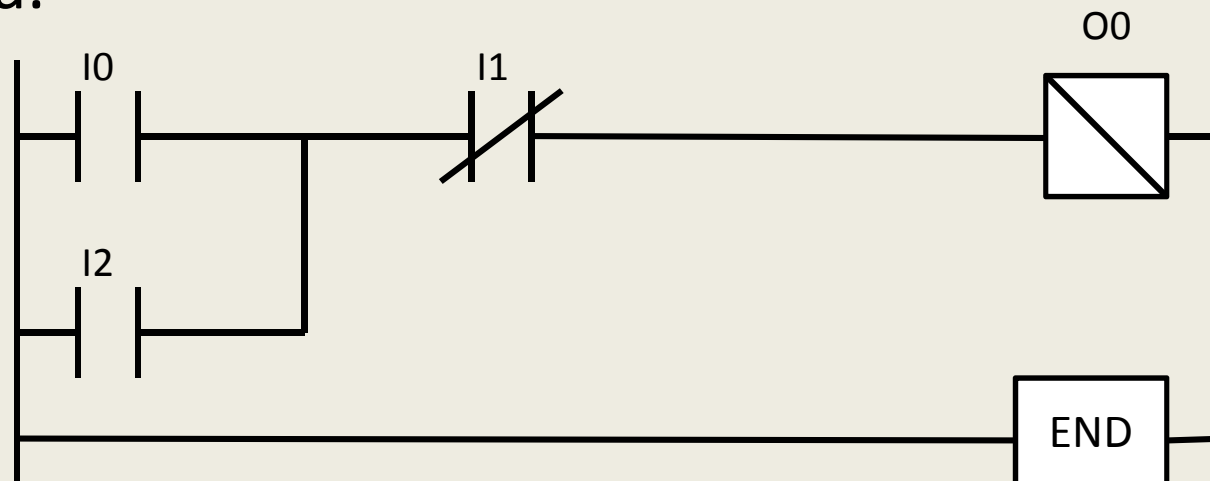
# Exemplo de Programação no Zelio Soft 2

- Selo de chave-instantânea



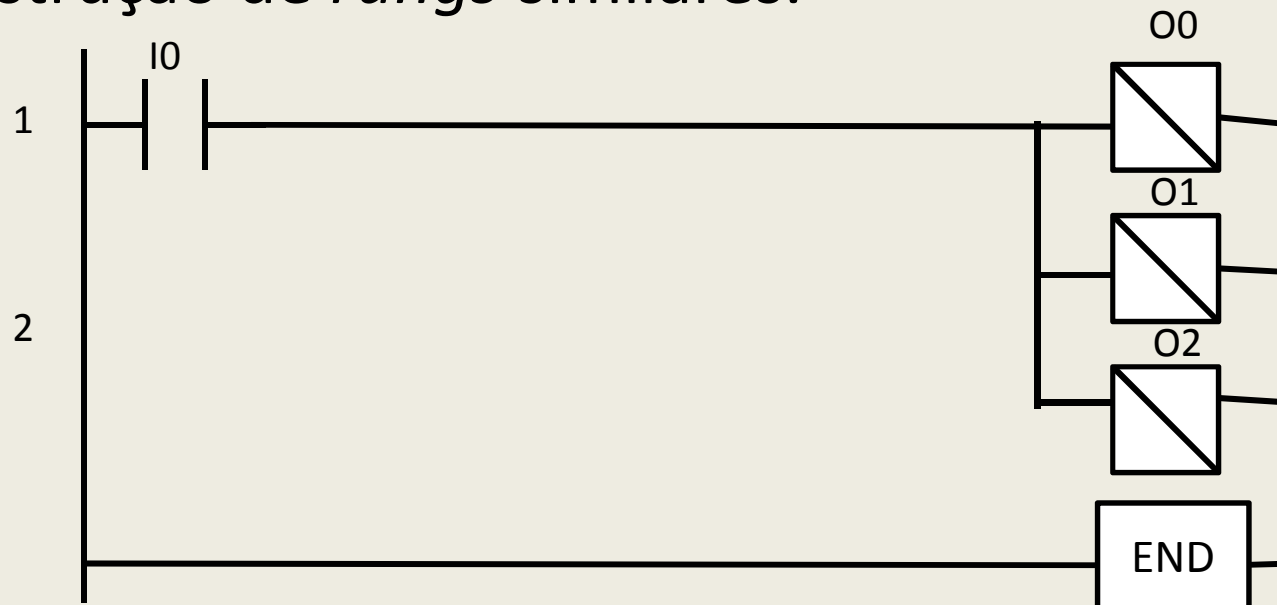
# A linguagem *Ladder*

- O nome *Ladder* deve-se a representação da linguagem se parecer com uma escada (*ladder*), na qual duas barras verticais paralelas são interligadas pela lógica de controle, formando degraus (*rungs*) da escada.



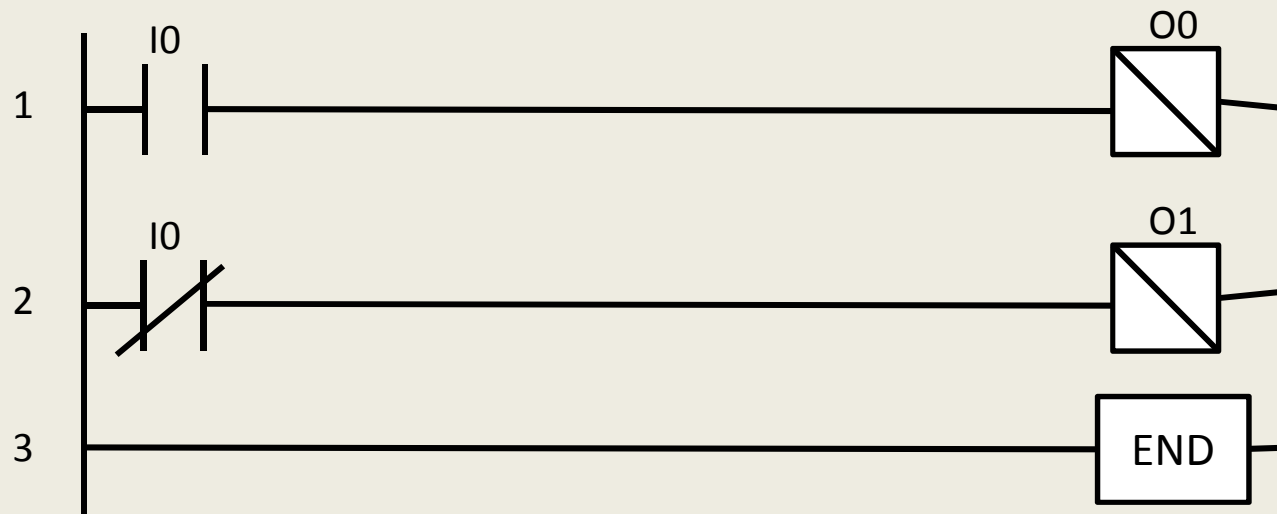
# Implementação da Lógica de Controle

- A linguagem *Ladder* permite o acionamento de vários elementos de saída (bobinas, temporizadores, contadores, etc.) simultaneamente, por meio da mesma lógica de controle, sem necessidade de construção de *rungs* similares.



# Exemplo

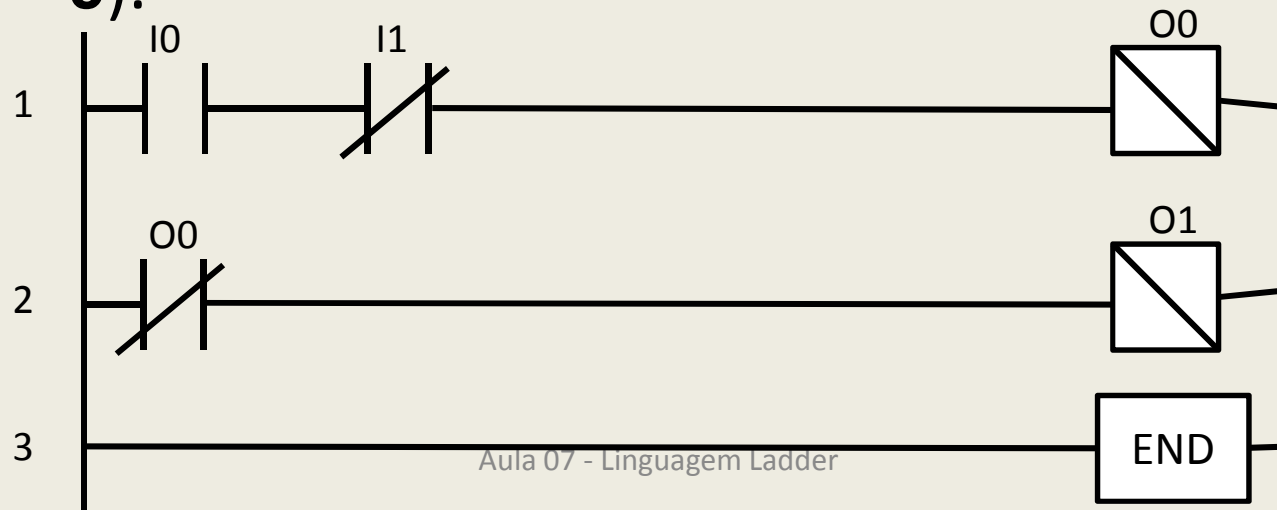
- Utilização de contatos NA e NF referenciados ao mesmo ponto de entrada.



- Se **I0 = 1** (entrada atuada), será acionada a saída **O0**, se **I0 = 0** (entrada não atuada), será acionada a saída **O1**.

# Exemplo

- Pode-se observar a utilização do tipo de dado **O** (**O0**) associado a uma instrução de entrada (contato NF). Neste caso, a saída **O0** é acionada a partir da condição das entradas **I0** e **I1** (**I0** = **1** e **I1** = **0**). Caso esta condição não seja satisfeita, a saída **O0** não é acionada ocasionando o acionamento da saída **O1** (**O0** = **0**).





# Relés de Controle (auxiliar): R

- Trata-se de bits internos à CPU, não tendo conexão a dispositivos externos de entrada ou saída. São úteis na definição das lógicas. É identificado através de uma numeração sequencial que inicia-se em zero e é limitado pela quantidade de memória disponível pela CPU.

