



UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA - CCET
DISCIPLINA: ARQUITETURA DE COMPUTADORES
ENGENHARIA ELÉTRICA

Nerval de Jesus Santos Junior
Discente

Prof. Dr. Denivaldo Lopes
Docente Responsável

Relatório da Unidade III sobre PLC

São Luís - MA
2023

Nerval de Jesus Santos Junior

Relatório da Unidade III sobre PLC

Engenharia Elétrica

Universidade Federal do Maranhão

Orientador: Prof. Dr. Denivaldo Lopes

São Luís - MA

2023

"Estamos aqui para fazer alguma diferença no universo, se não, por que estar aqui?"

(Steve Jobs)

Lista de ilustrações

Figura 1 – Circuito lógico da expressão	9
Figura 2 – Tabela verdade da Questão 1	10
Figura 3 – Continuação da Tabela verdade da Questão 1	11
Figura 4 – Mapa de karnaugh da Questão 1	12
Figura 5 – Circuito Logico da expressão simplificada	12
Figura 6 – Programação Ladder parte 1	13
Figura 7 – Programação Ladder parte 2	13
Figura 8 – simplificação da programação Ladder parte 1	14
Figura 9 – simplificação da programação Ladder parte 2	14
Figura 10 – Programa funcionando	15
Figura 11 – Mostrando o PLC	15
Figura 12 – Tabela verdade da questão 3 parte 1	16
Figura 13 – Tabela verdade da questão 3 parte 2	17
Figura 14 – Mapa de karnaugh da questão 3	18
Figura 15 – circuito do software Boolean TT	19
Figura 16 – Circuito no Logisim	20
Figura 17 – Circuito simplificado	20
Figura 18 – Programa em Ladder da questão 3	21
Figura 19 – Programado para 9:30 em Ladder	22
Figura 20 – Programa rodando no PLC no cli edit	23
Figura 21 – Analise combinacional	24
Figura 22 – Mapa k da expressão 1	24
Figura 23 – Mapa k da expressão 2	25
Figura 24 – Programa em Ladder da questão 4	25
Figura 25 – Rodando programa em Ladder da questão 4	26
Figura 26 – Rodando programa em Ladder da questão 4	26
Figura 27 – Tabela verdade parte 1 da questão 5	28
Figura 28 – Tabela verdade parte 2 da questão 5	29
Figura 29 – Circuito logico no logisim	30
Figura 30 – Circuito logico no logisim simplificado	31
Figura 31 – Programa em Ladder	32
Figura 32 – Programa em Ladder	33
Figura 33 – Programa em Ladder	34
Figura 34 – esquema do Tanque	34
Figura 35 – tabela verdade do esquema do Tanque	36
Figura 36 – continuação da tabela verdade	36

Figura 37 – Mapa k da questão 5 da válvula de entrada	38
Figura 38 – Mapa k da questão 5 da válvula de saída	38
Figura 39 – Mapa k da questão 5 de condições de erro	38
Figura 40 – circuito evs	39
Figura 41 – circuito eve	40
Figura 42 – circuito erro	41
Figura 43 – programa Ladder questão 6	42
Figura 44 – Rodando programa Ladder questão 6	42
Figura 45 – questão 6 no PLC	43

Sumário

1	INTRODUÇÃO	6
1.1	Objetivos	7
1.1.1	Específicos	7
1.2	Materias	7
2	METODOLOGIA	8
3	RESULTADOS	9
3.1	Questão 1	9
3.2	Questão 3	16
3.3	Questão 4	23
3.4	Questão 5	27
3.5	Questão 6	34
4	CONCLUSÃO	44
	REFERÊNCIAS	45

1 Introdução

A automação de processos industriais tem sido fundamental para a otimização da produção e aumento dos lucros das indústrias, e os Controladores Lógicos Programáveis (CLPs) são dispositivos especiais amplamente utilizados para controlar componentes e equipamentos industriais.

Neste relatório, apresentaremos os procedimentos e programas desenvolvidos durante uma aula prática de laboratório de aplicações com microcomputadores. Além disso, exibiremos os resultados obtidos com os experimentos e faremos uma conclusão geral sobre o trabalho realizado.

Os objetivos deste relatório são fornecer uma visão clara e objetiva dos procedimentos e programas desenvolvidos, explorar as funcionalidades e características dos CLPs, demonstrar o uso do ambiente de desenvolvimento para projetos e solucionar problemas encontrados durante o desenvolvimento desses projetos no ambiente Clic Edit ([WEG, s.d.](#)). Também abordaremos o uso da linguagem Ladder e a aplicação do PLC em diferentes contextos.

A linguagem de programação comumente empregada para programar CLPs é o Diagrama Ladder (LD), que permite a criação de lógicas de controle de forma intuitiva. No entanto, bugs em programas de LD podem ser extremamente caros e, em alguns casos, representar uma ameaça à segurança humana. Para lidar com esse desafio, propõe-se uma abordagem orientada a modelos para a verificação formal de programas de LD, utilizando técnicas de verificação de modelos ([SOUZA et al., 2012](#)). A verificação formal de programas de LD é uma área de pesquisa em constante evolução, e várias técnicas têm sido propostas para melhorar a segurança e confiabilidade desses programas. Uma dessas técnicas é o uso de redes de Petri temporizadas (TPN) para modelar programas de LD e verificar automaticamente propriedades comportamentais sobre esses modelos ([BENITEZ-PINA et al., 2017](#)). Essa abordagem permite a detecção de condições de corrida, um tipo comum de bug que pode ocorrer em programas de LD, e a redução significativa da ocorrência de bugs ([FRANCISCO, 2017](#)). Para a realização de experimentos com CLPs e a linguagem Ladder, é necessário o uso de um ambiente de desenvolvimento integrado (IDE) ([ASPAR; SHAIKH-HUSIN; KHALIL-HANI, 2018](#)). O sistema operacional Windows 7 de 64 bits e o ambiente de desenvolvimento integrado (IDE) WEG Clic 02 Edit são utilizados para o desenvolvimento dos programas ([ZHOU; TWISS, 1998](#)). No próximo capítulo, serão detalhados os procedimentos adotados para a realização dos experimentos, seguido pela metodologia utilizada.

1.1 Objetivos

O objetivo deste relatório é apresentar de forma clara e objetiva os procedimentos e programas que foram desenvolvidos em aula prática, dentro da disciplina de laboratório de arquitetura de computadores. Será feita também a exposição dos resultados obtidos com o experimento e uma conclusão geral a respeito do trabalho realizado.

1.1.1 Específicos

- Comparar os resultados obtidos por uma expressão booleana expandida e a sua versão simplificada por mapa de Karnaugh;
- Explorar as funcionalidades e características do PLC;
- Demonstrar a utilização do ambiente de desenvolvimento para projetos com o microcontrolador HCS12;
- Solucionar problemas encontrados durante o desenvolvimento de projetos no ambiente *Clik Edit*;
- Mostrar o uso da linguagem Ladder;
- Mostrar uso do PLC.

1.2 Materias

Hardware:

- Microcomputador (processador Core I7, RAM 8 GB, HD 1TB, SSD 258, Monitor 16");
- Kit de desenvolvimento para HCS12: APS12C128SLK [Ref1];
- Kit de desenvolvimento (placa de expansão): SLK0109UG [Ref2].

Software:

- Sistema operacional: Windows 10 64 bits;
- Ambiente de desenvolvimento integrado (IDE): WEG Clic 02 Edit.

2 Metodologia

Para a realização adequada do experimento e compreensão do mesmo, serão seguidos os seguintes passos: Gerar a tabela verdade e expressões booleanas correspondente ao problema. O software computacional Clic02 Edit será o ambiente de programação, na linguagem Ladder, para implementação da lógica de controle. O programa será carregado e executado no PLC, a fim de ser testado e observado os resultados finais.

Análise das expressões booleanas: Para cada questão, iremos analisar a expressão booleana fornecida e compreender sua lógica de funcionamento. Faremos uma avaliação inicial das variáveis e operações envolvidas para obter uma compreensão completa do problema.

Simplificação das expressões booleanas: Utilizando o mapa de Karnaugh, iremos simplificar as expressões booleanas fornecidas. Faremos agrupamentos lógicos e eliminação de termos redundantes para obter a forma mais compacta e simplificada possível das expressões.

Implementação em LADDER: Com base na expressão booleana simplificada, iremos desenvolver o programa em LADDER para o PLC da Weg "CLW-02/20VT-D". Utilizaremos os elementos de lógica disponíveis no software WEG Clic02 Edit para criar as redes de contatos e bobinas necessárias para implementar a lógica de controle.

Comparação das saídas: Após implementar o programa em LADDER com a expressão booleana original e a expressão simplificada, iremos realizar testes e comparar as saídas obtidas. Analisaremos se as duas implementações produzem resultados equivalentes ou se há diferenças significativas no comportamento do programa.

Obtenção da expressão simplificada: Para as questões 3 e 5, iremos obter a expressão booleana simplificada antes de implementar o programa em LADDER. Utilizaremos técnicas de simplificação lógica, como o mapa de Karnaugh, para reduzir as expressões a formas mais compactas e eficientes.

Essa metodologia nos permitirá abordar cada questão de forma sistemática, buscando soluções eficientes e comparando os resultados obtidos com as expressões originais e simplificadas. Dessa forma, poderemos verificar a eficácia da simplificação lógica e a correta implementação dos programas em LADDER para o PLC da Weg "CLW-02/20VT-D".

Utilizar a expressão logica das questões e usar o Logisim para fazer o circuito logico e simplificação. (C. Burch, 2011)

3 Resultados

Nesse capítulo será possível observar o método e a forma de como foi solucionada as questões escolhidas, bem como a solução da mesma e implementação em linguagem Ladder dentro do Clic Edit (WEG, s.d.).

3.1 Questão 1

Pede-se que seja feito um programa em LADDER para o PLC da Weg “CLW-02/20VT-D” que corresponda a seguinte expressão booleana. (pontos: 3,33)

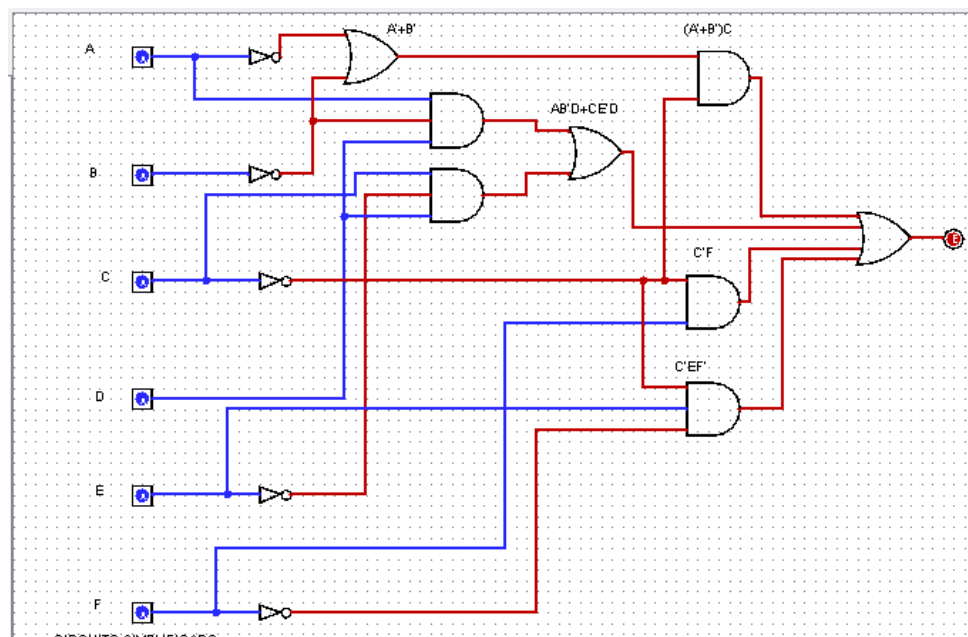
Expressão booleana

$$S = AB'D + CE'D + (A + B)' \cdot C' + C'F + C'EF' \quad (3.1)$$

Depois, simplifique esta expressão booleana e implemente o programa em LADDER da expressão simplificada. Compare a saída da expressão original e da expressão simplificada, colocando-as no mesmo programa LADDER. OBS: Deve-se encontrar a menor expressão booleana através do mapa de Karnaugh e depois implementá-la em LADDER.

O circuito lógico da expressão booleana na Figura 1 a seguir.

Figura 1 – Circuito lógico da expressão



Fonte: Autoria Própria.

A tabela verdade para a expressão na Figura 2 a seguir. Ela foi feita dentro do *software mobile Boolean algebra* (APPLE INC., 2023).

Figura 2 – Tabela verdade da Questão 1

A	B	C	D	f
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0

Fonte: Autoria Própria.

Continuação da tabela verdade para a expressão na Figura 3 a seguir.

Figura 3 – Continuação da Tabela verdade da Questão 1

A	B	C	D	f
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

Fonte: Autoria Própria.

O mapa de karnaugh para a expressão na Figura 4 a seguir.

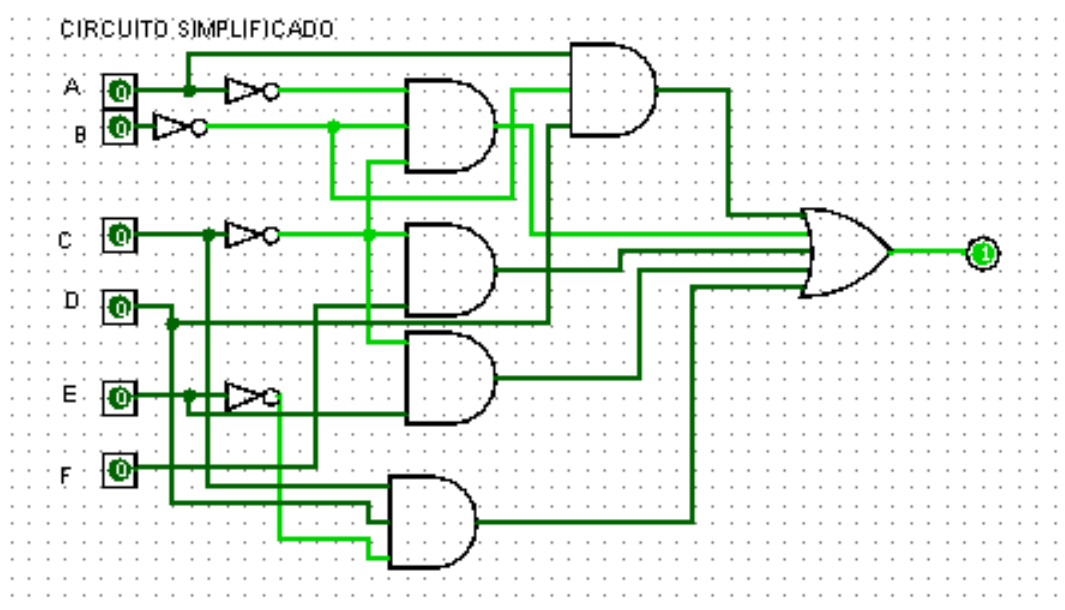
Figura 4 – Mapa de karnaugh da Questão 1



Fonte: Autoria Própria.

A seguir é possível ver o circuito simplificado na Figura 5.

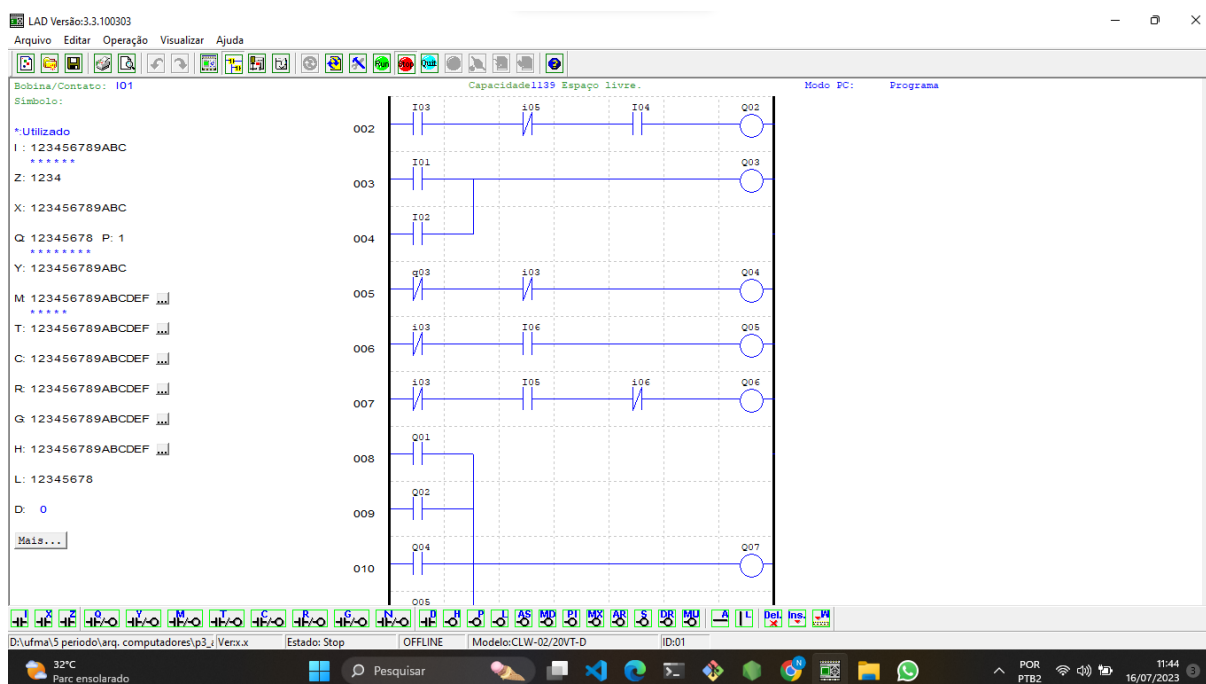
Figura 5 – Circuito Logico da expressão simplificada



Fonte: Autoria Própria.

A seguir é possível ver o programa em Ladder da questão 1

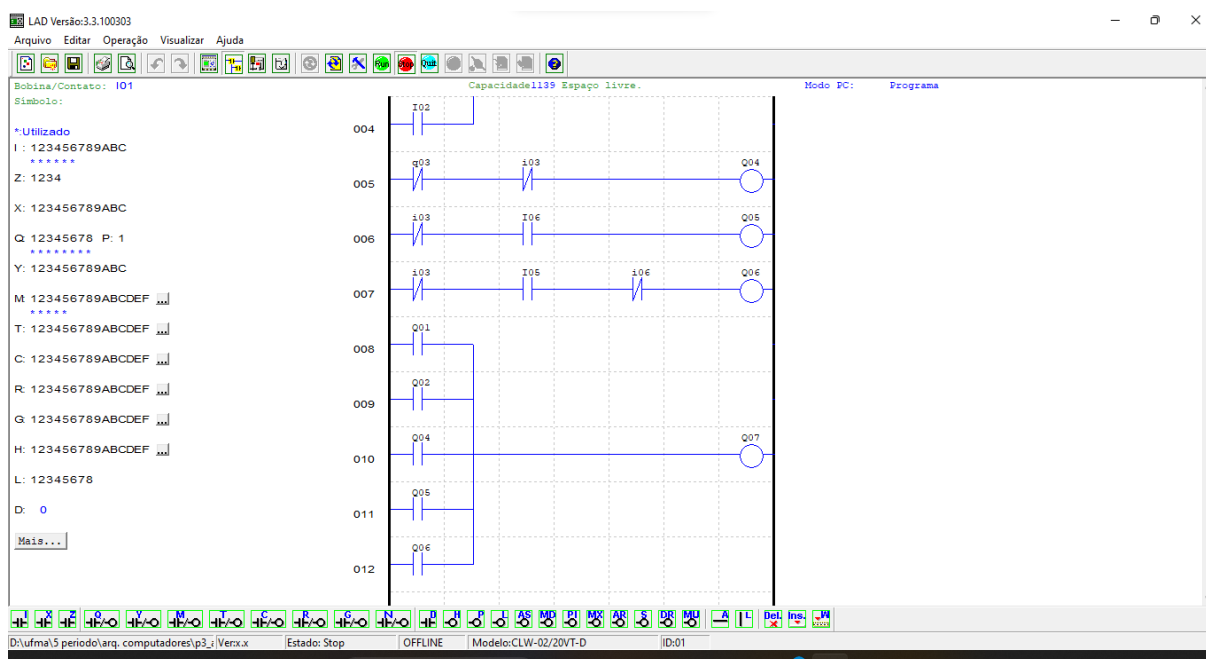
Figura 6 – Programação Ladder parte 1



Fonte: Autoria Própria.

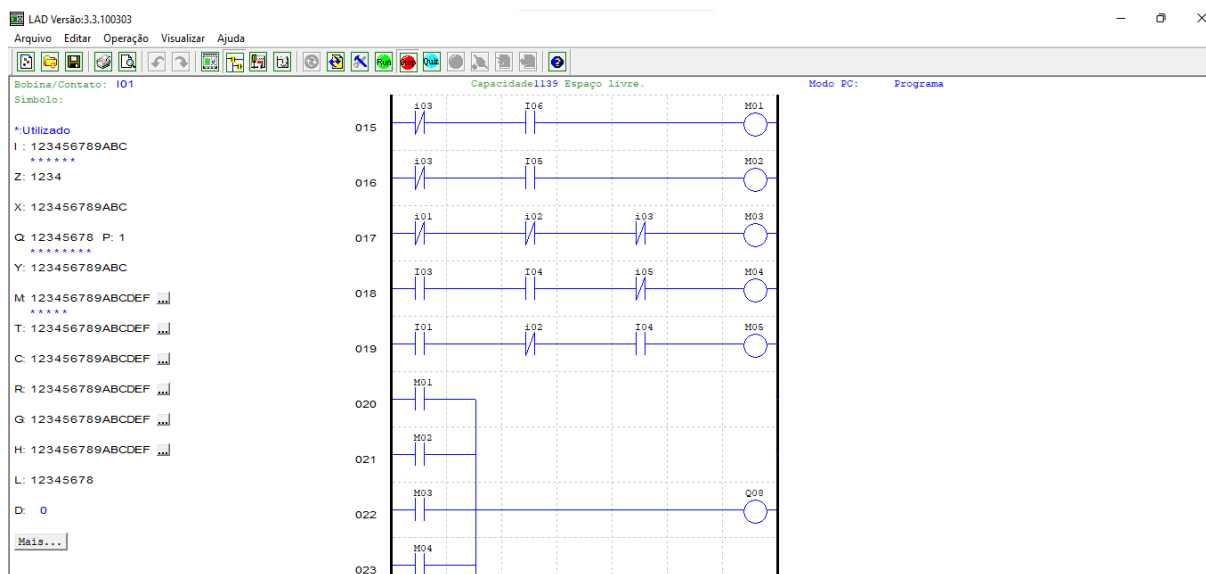
Na imagem 7 é possível ver toda a continuação do circuito de maneira mais completa

Figura 7 – Programação Ladder parte 2



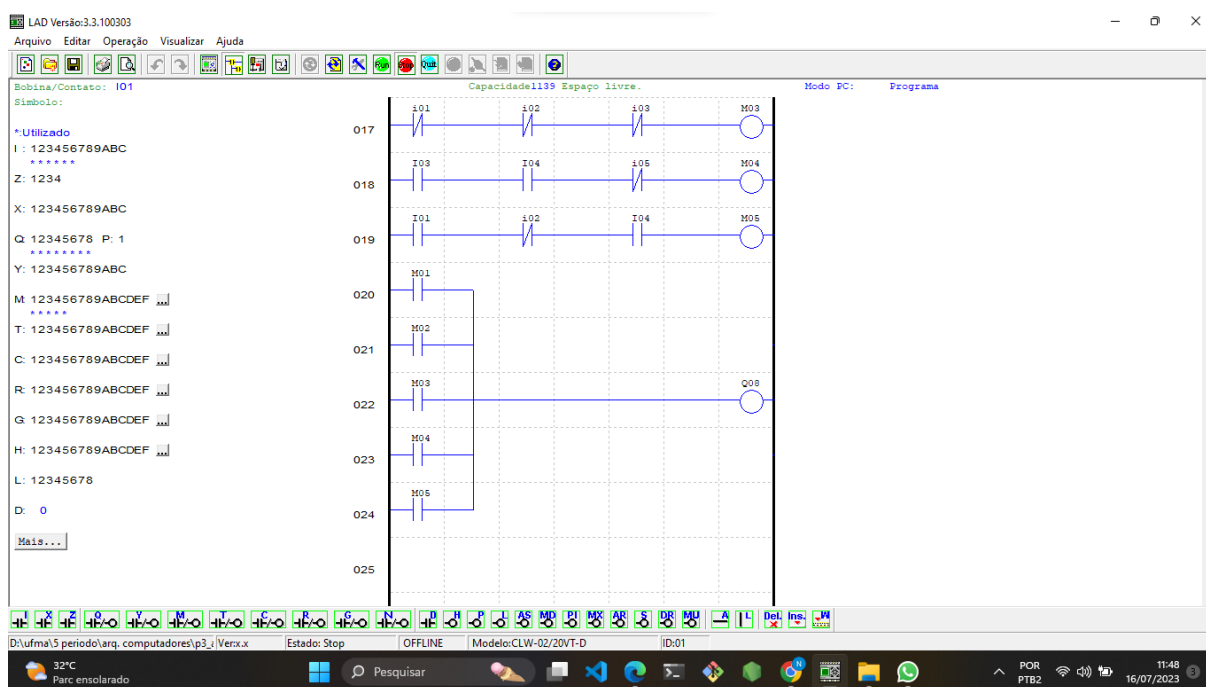
Fonte: Autoria Própria.

Figura 8 – simplificação da programação Ladder parte 1



Fonte: Autoria Própria.

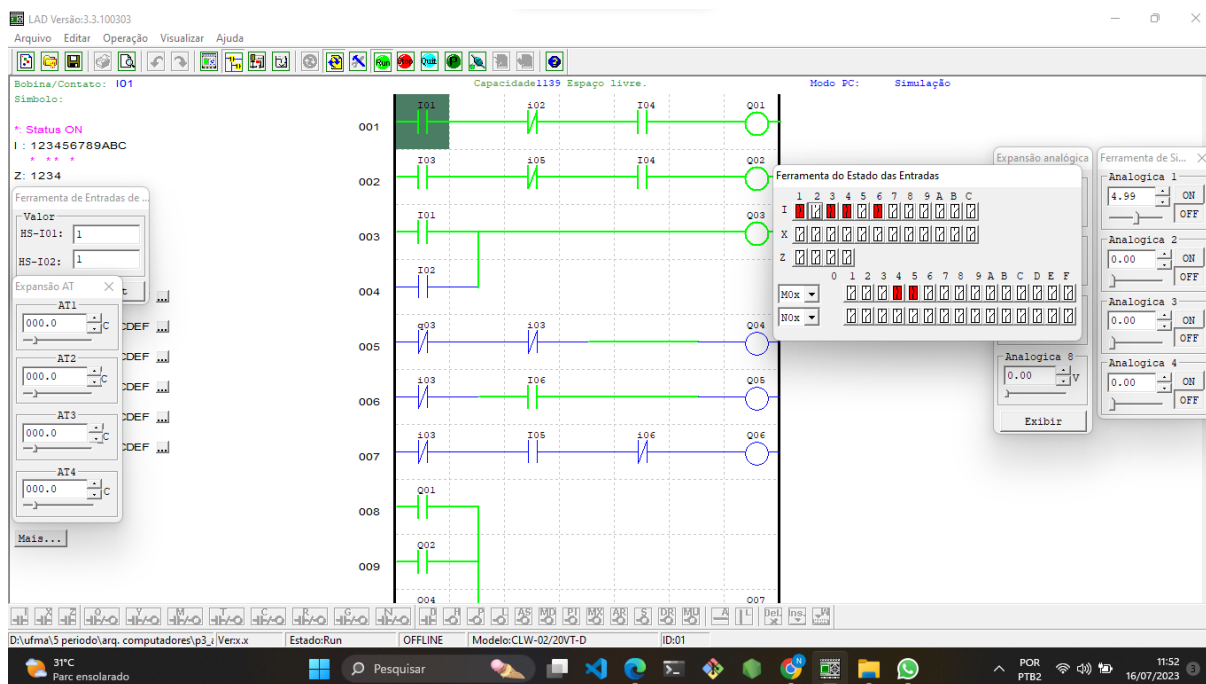
Figura 9 – simplificação da programação Ladder parte 2



Fonte: Autoria Própria.

Dessa maneira rodando o programa ao satisfazer condições descritas no enunciado é possível é as saídas de acordo com a tabela verdade.

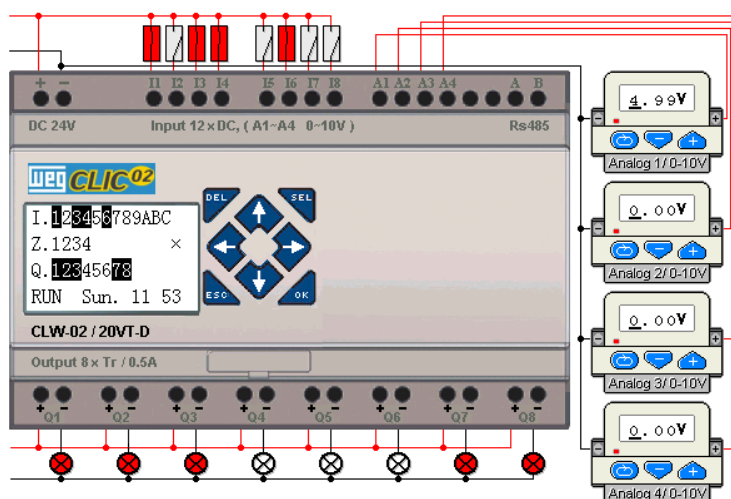
Figura 10 – Programa funcionando



Fonte: Autoria Própria.

Na Figura 11 a seguir é possível ver uma disposição diferente do PLC

Figura 11 – Mostrando o PLC



Fonte: Autoria Própria.

3.2 Questão 3

Um determinado sistema é acionado através da expressão S2 uma vez por dia as 09:30 e deve permanecer ligado por 20 minutos, desde que a expressão booleana S1 seja verdadeira (pontos: 3,33).

$$S1 = ABE + BCD'F' + (B + A'C')D + A'F' + BCF + (D' + AB)E'F' \quad (3.2)$$

$$S2 = S1 \quad (3.3)$$

OBS: Utilize o relógio de tempo real do PLC para que S2 seja avaliado todo dia as 09:30, e seja verdadeiro se, somente se, S1 for verdadeiro. Deve-se obter a expressão simplificada antes de implementar o programa em LADDER.

A tabela da verdade foi feita dentro do Software Excel como podemos ver a seguir. Nela temos 6 entradas e as saídas satisfazem as condições estabelecidas.

Figura 12 – Tabela verdade da questão 3 parte 1

Tabela verdade:						
A	B	C	D	E	F	S1
0	0	0	0	0	0	1
0	0	0	0	0	1	0
0	0	0	0	1	0	0
0	0	0	0	1	1	0
0	0	0	1	0	0	0
0	0	0	1	0	1	0
0	0	0	1	1	0	0
0	0	0	1	1	1	0
0	0	1	0	0	0	1
0	0	1	0	0	1	0
0	0	1	0	1	0	0
0	0	1	0	1	1	0
0	0	1	1	0	0	0
0	0	1	1	0	1	0
0	0	1	1	1	0	0
0	0	1	1	1	1	0
0	1	0	0	0	0	1
0	1	0	0	0	1	0
0	1	0	0	1	0	0
0	1	0	0	1	1	0
0	1	0	1	0	0	1
0	1	0	1	0	1	1
0	1	0	1	1	0	1
0	1	0	1	1	1	1
0	1	1	0	0	0	1
0	1	1	0	0	1	1
0	1	1	0	1	0	1
0	1	1	0	1	1	1
0	1	1	1	0	0	1
0	1	1	1	0	1	1
0	1	1	1	1	0	1
0	1	1	1	1	1	1
1	0	0	0	0	0	0
1	0	0	0	0	1	0
1	0	0	0	1	0	0
1	0	0	0	1	1	0
1	0	0	1	0	0	0
1	0	0	1	0	1	0
1	0	0	1	1	0	0
1	0	0	1	1	1	0
1	0	1	0	0	0	1
1	0	1	0	0	1	0
1	0	1	0	1	0	0
1	0	1	0	1	1	0
1	0	1	1	0	0	0
1	0	1	1	0	1	0
1	0	1	1	1	0	0
1	0	1	1	1	1	0
1	1	0	0	0	0	1
1	1	0	0	0	1	0
1	1	0	0	1	0	0
1	1	0	0	1	1	0
1	1	0	1	0	0	1
1	1	0	1	0	1	1
1	1	0	1	1	0	1
1	1	0	1	1	1	1
1	1	1	0	0	0	1
1	1	1	0	0	1	1
1	1	1	0	1	0	1
1	1	1	0	1	1	1
1	1	1	1	0	0	1
1	1	1	1	0	1	1
1	1	1	1	1	0	1
1	1	1	1	1	1	1

Fonte: Autoria Própria.

A continuação da tabela verdade da questão 3 ilustrada a seguir, pois o tamanho da imagem caso fosse menor dificultaria a leitura do leitor.

Figura 13 – Tabela verdade da questão 3 parte 2

1	0	0	0	0	0	1
1	0	0	0	0	1	0
1	0	0	0	1	0	1
1	0	0	0	1	1	0
1	0	0	1	0	0	1
1	0	0	1	0	1	1
1	0	0	1	1	0	1
1	0	0	1	1	1	1
1	0	1	0	0	0	1
1	0	1	0	0	1	0
1	0	1	0	1	0	1
1	0	1	0	1	1	0
1	0	1	1	0	0	1
1	0	1	1	0	1	0
1	0	1	1	1	0	1
1	0	1	1	1	1	0
1	1	0	0	0	0	1
1	1	0	0	0	1	0
1	1	0	0	1	0	1
1	1	0	0	1	1	1
1	1	0	1	0	0	1
1	1	0	1	0	1	1
1	1	0	1	1	0	1
1	1	0	1	1	1	1
1	1	1	0	0	0	1
1	1	1	0	0	1	1
1	1	1	0	1	0	1
1	1	1	0	1	1	1
1	1	1	1	0	0	1
1	1	1	1	0	1	1
1	1	1	1	1	0	1
1	1	1	1	1	1	1
1	1	1	1	1	0	1
1	1	1	1	1	1	1

Fonte: Autoria Própria.

Para o Mapa de karnaugh da questão 3 foi utilizado novamente o software Boolean algebra como mostrado na Figura 14 (APPLE INC., 2023).

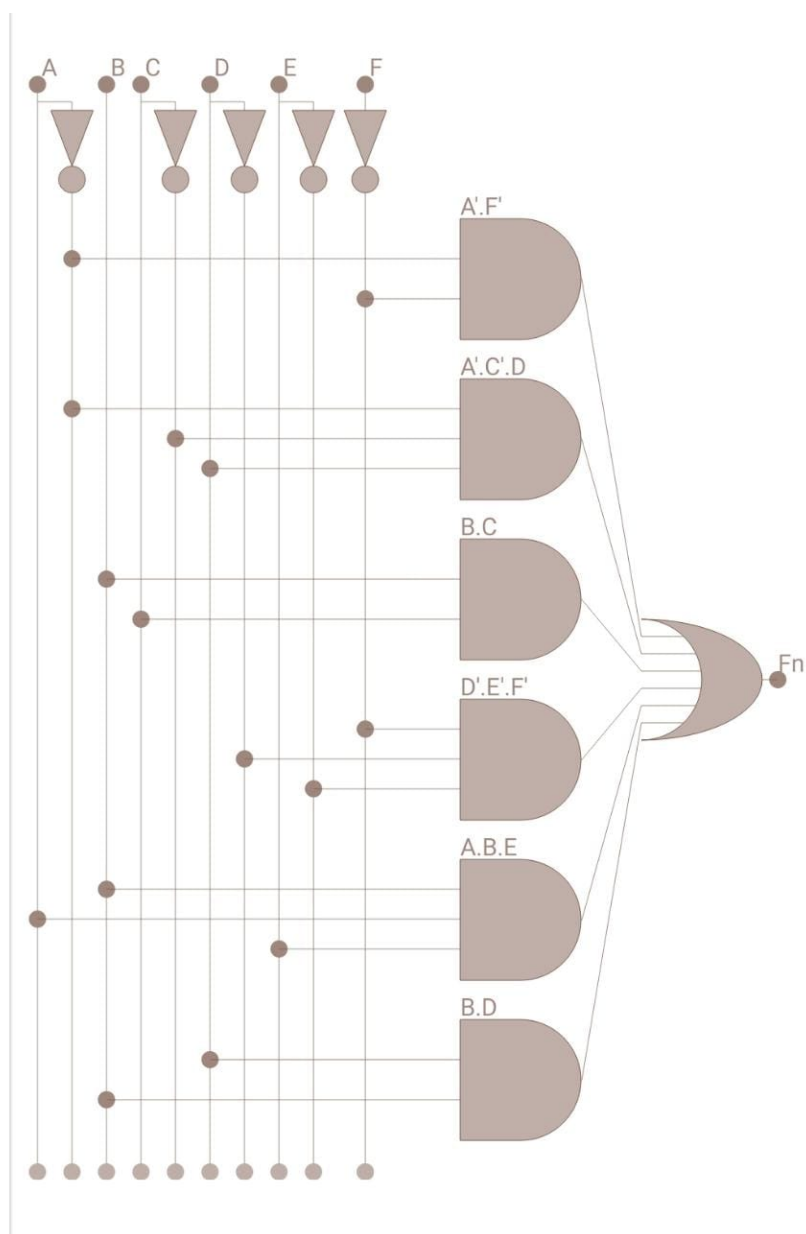
Figura 14 – Mapa de karnaugh da questão 3



Fonte: Autoria Própria.

Na Figura 15 foi utilizado o software mobile Boolean TT para confecção do circuito lógico ([BOOLEANTT.HAZEAPPS.COM](https://booleanTT.hazeapps.com), 2023).

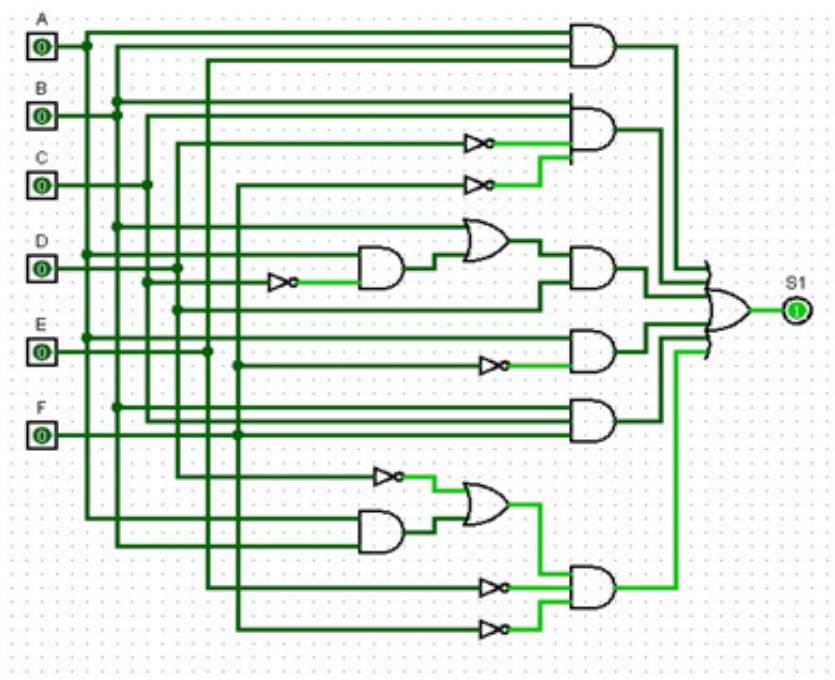
Figura 15 – circuito do software Boolean TT



Fonte: Autoria Própria.

Na figura 16 é possível ver o circuito feito dentro do logisim para ajudar na confecção em Ladder (C. Burch, 2011).

Figura 16 – Circuito no Logisim

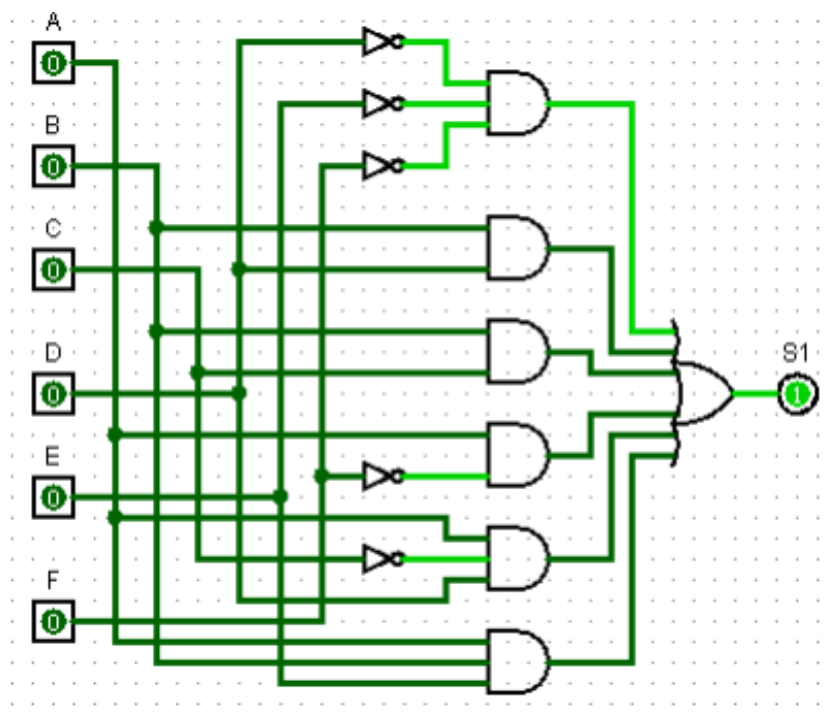


Fonte: Autoria Própria.

S1 simplificada:

$$S1 = ABE + D'E'F' + BD + BC + AF' + AC'D \quad (3.4)$$

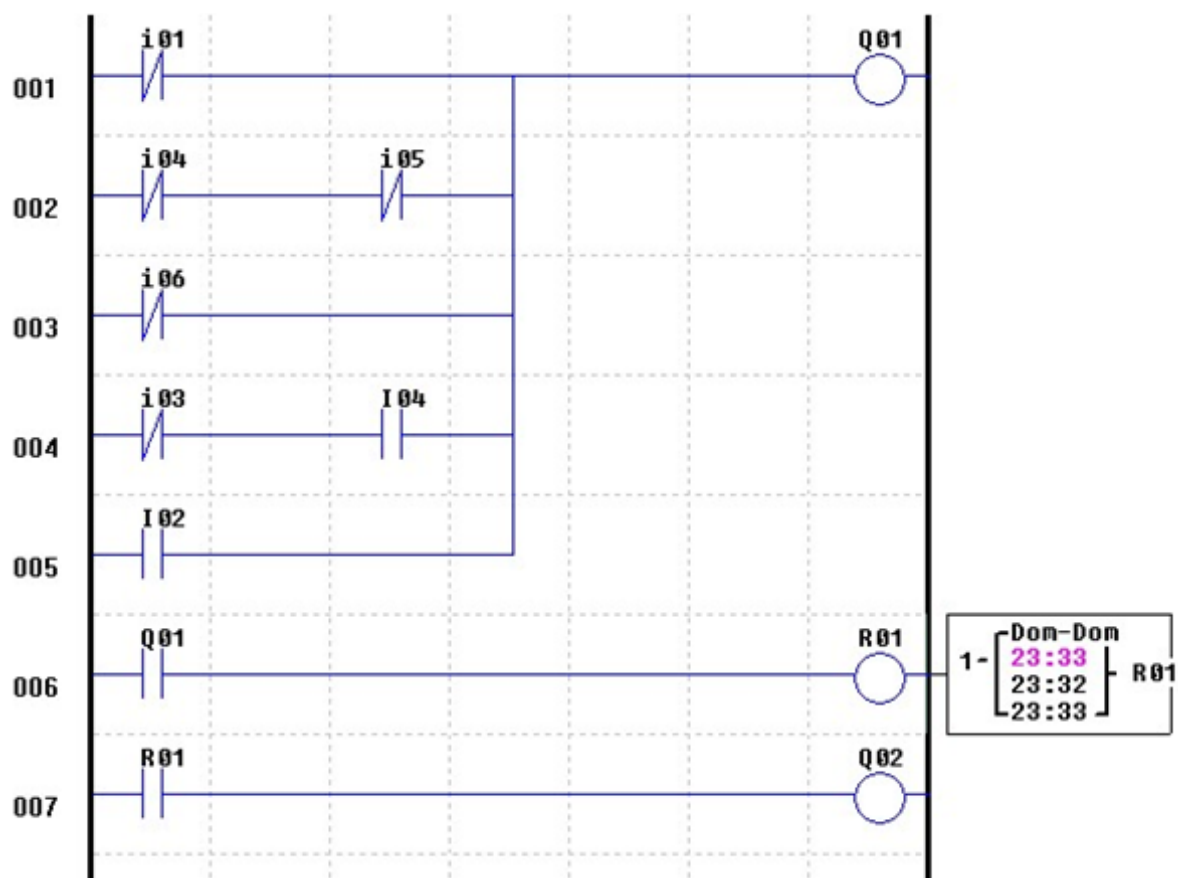
Figura 17 – Circuito simplificado



Fonte: Autoria Própria.

O programa em Ladder da questão está mostrado na Figura 18 (WEG, s.d.).

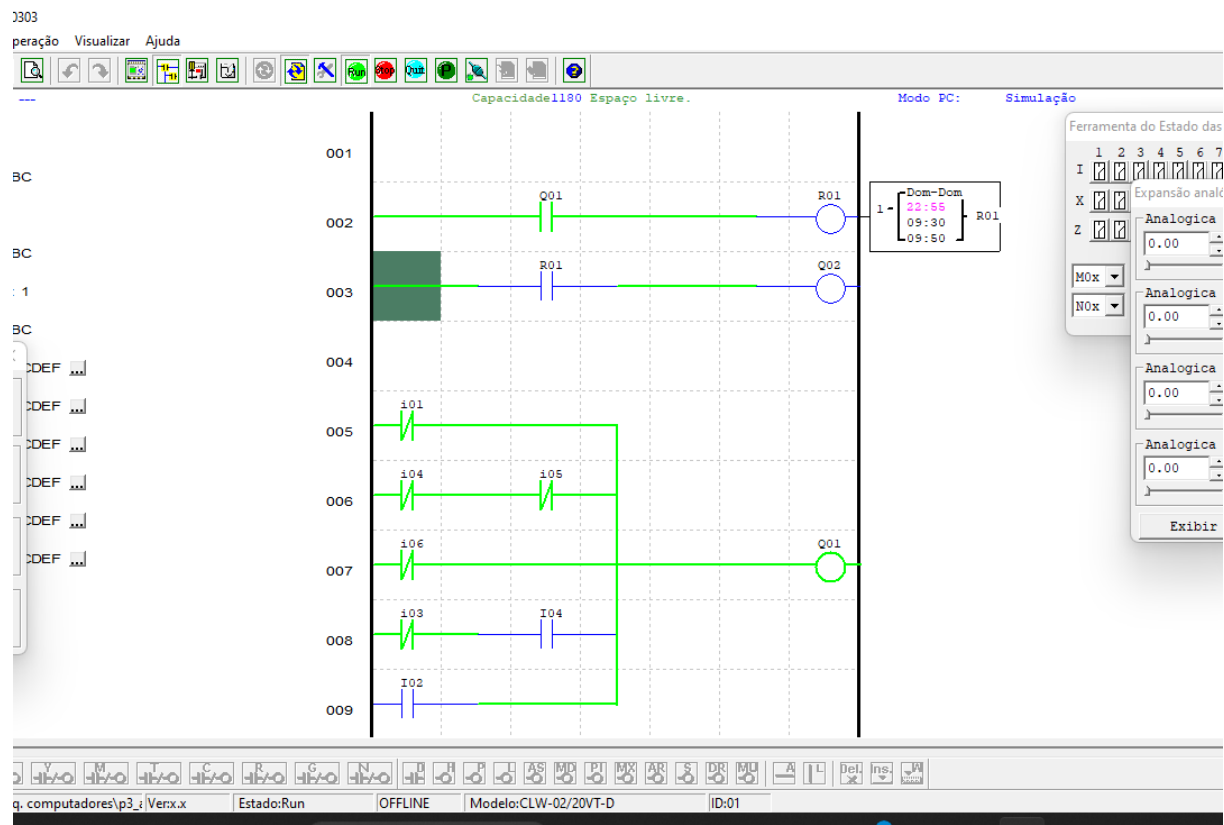
Figura 18 – Programa em Ladder da questão 3



Fonte: Autoria Própria.

Ao rodar pode-se ser feito o teste das condições descritas no enunciado da questão se baseando na tabela da verdade.

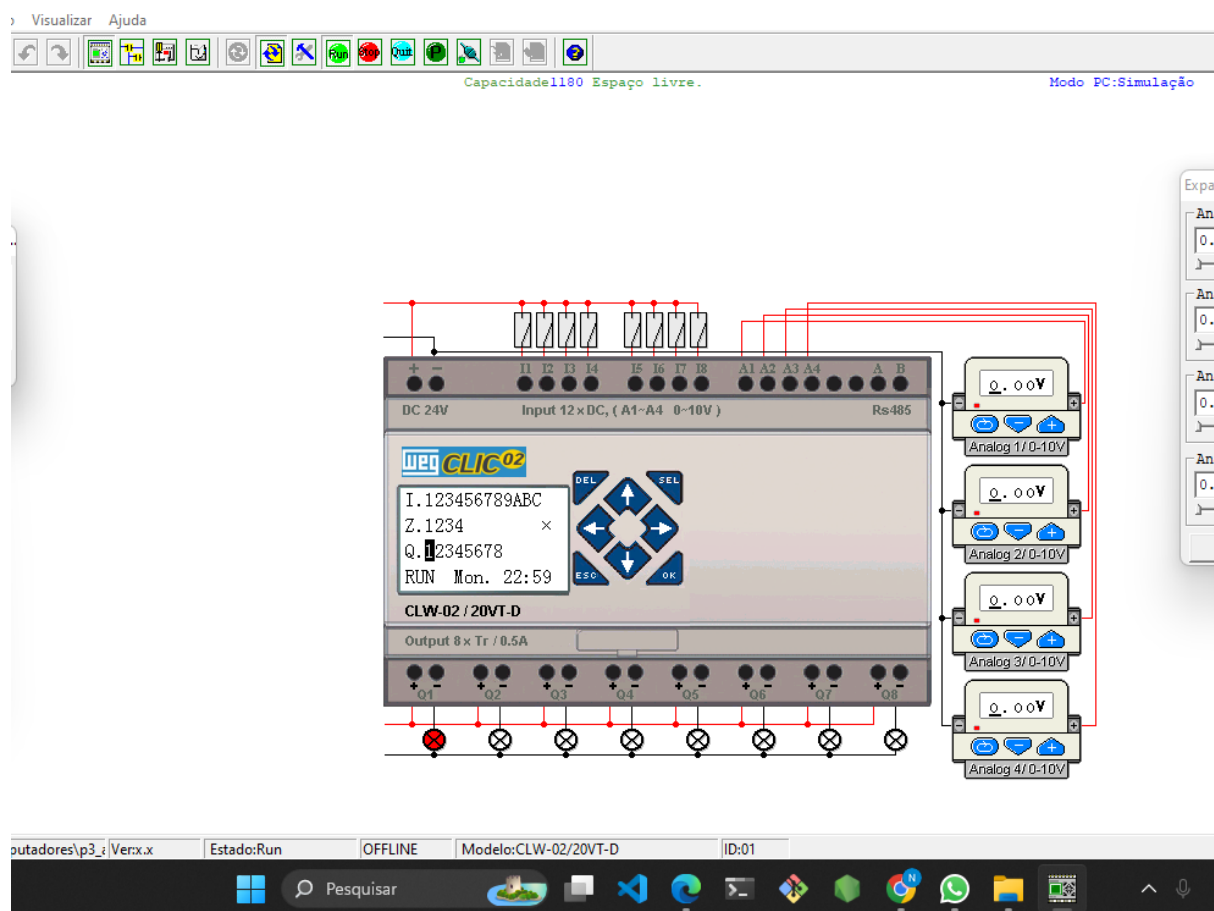
Figura 19 – Programado para 9:30 em Ladder



Fonte: Autoria Própria.

O programa Ladder no PLC mostrado na Figura 20

Figura 20 – Programa rodando no PLC no cliq edit



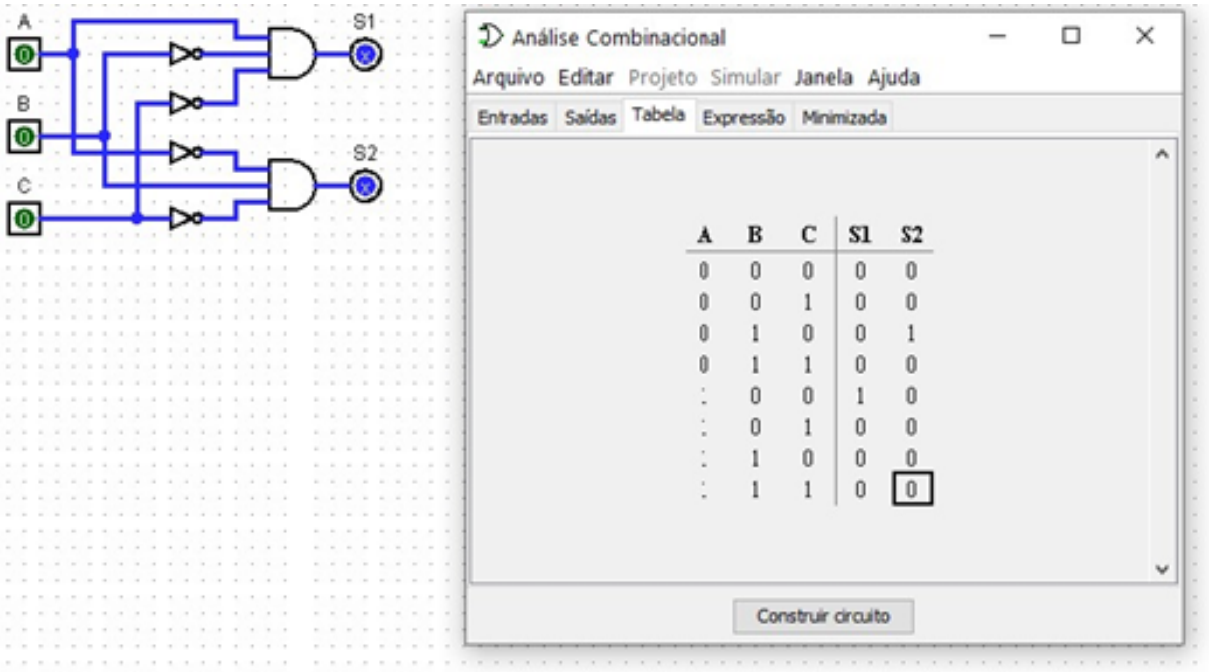
Fonte: Autoria Própria.

3.3 Questão 4

Um motor deve ser acionado através de duas botoeiras: sentido horário e sentido anti-horário. Além disto, há uma botoeira de emergência que faz o motor parar. Implemente um programa em LADDER que faça o acionamento do motor, sendo que a saída S1 aciona no sentido horário e a saída S2 aciona no sentido anti-horário (pontos: 3,33).

Na Figura 21 a seguir é possível ver a Análise combinacional utilizada para confecção do circuito em Ladder.

Figura 21 – Análise combinacional



Fonte: Autoria Própria.

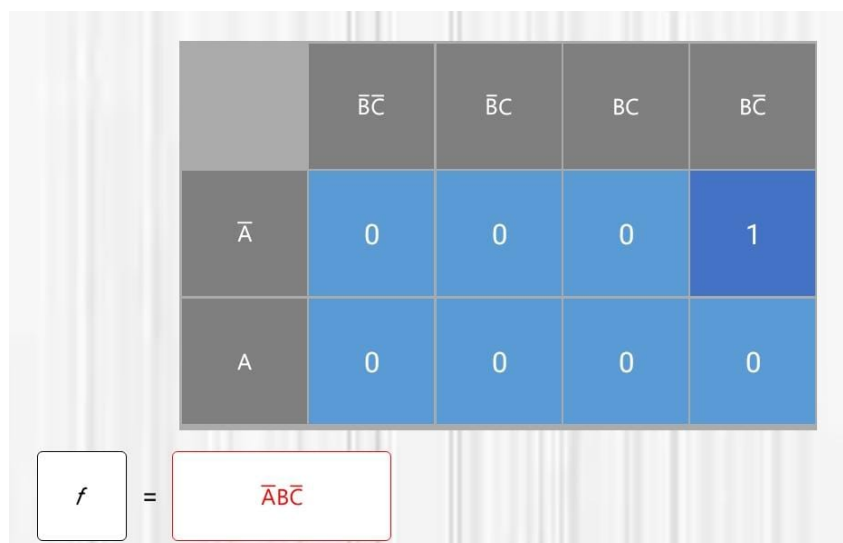
Para o Mapa k de cada expressão foi utilizado o software Boolean Algebra ([APPLE INC., 2023](#)).

Figura 22 – Mapa k da expressão 1



Fonte: Autoria Própria.

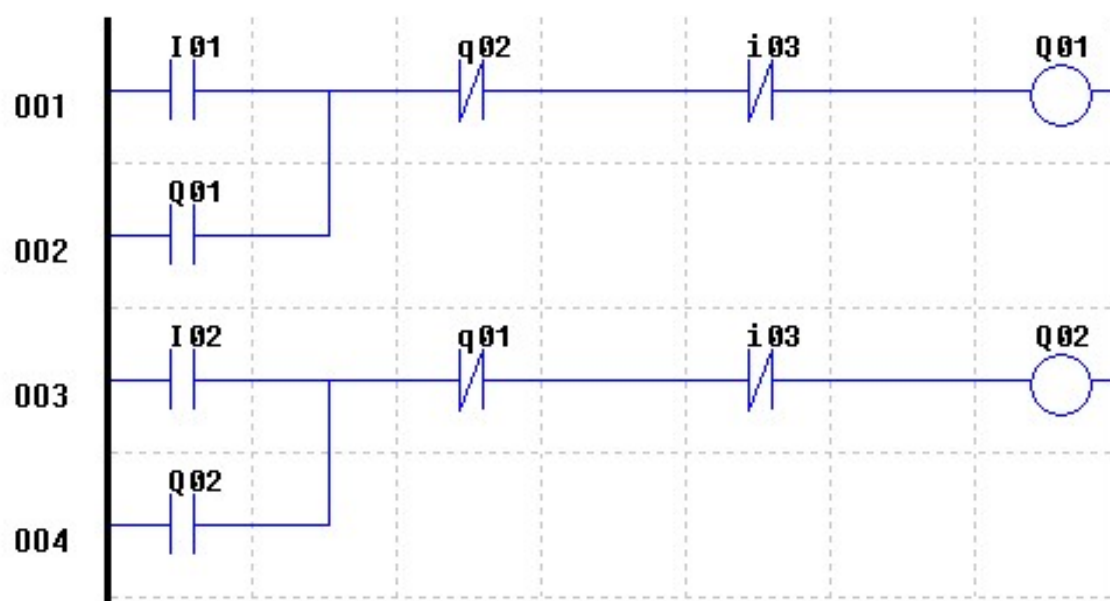
Figura 23 – Mapa k da expressão 2



Fonte: Autoria Própria.

A programação em Ladder está na Figura 24 a seguir.

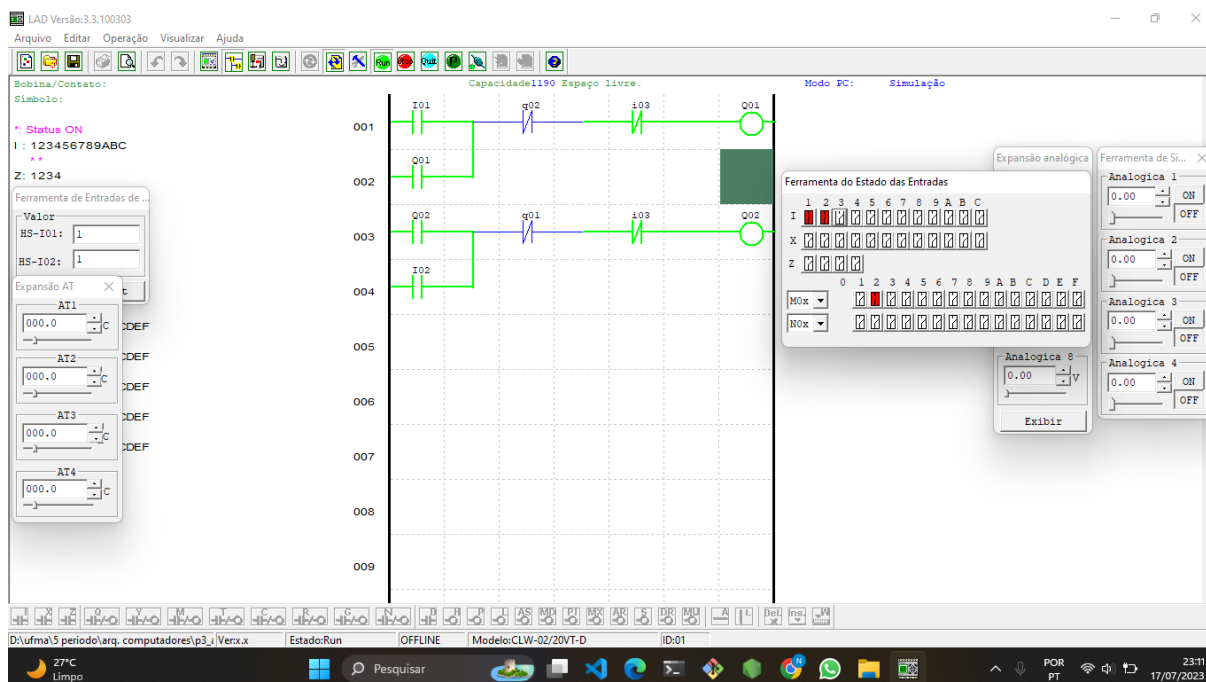
Figura 24 – Programa em Ladder da questão 4



Fonte: Autoria Própria.

Rodando o programa podemos testar de acordo com as condições estabelecidas.

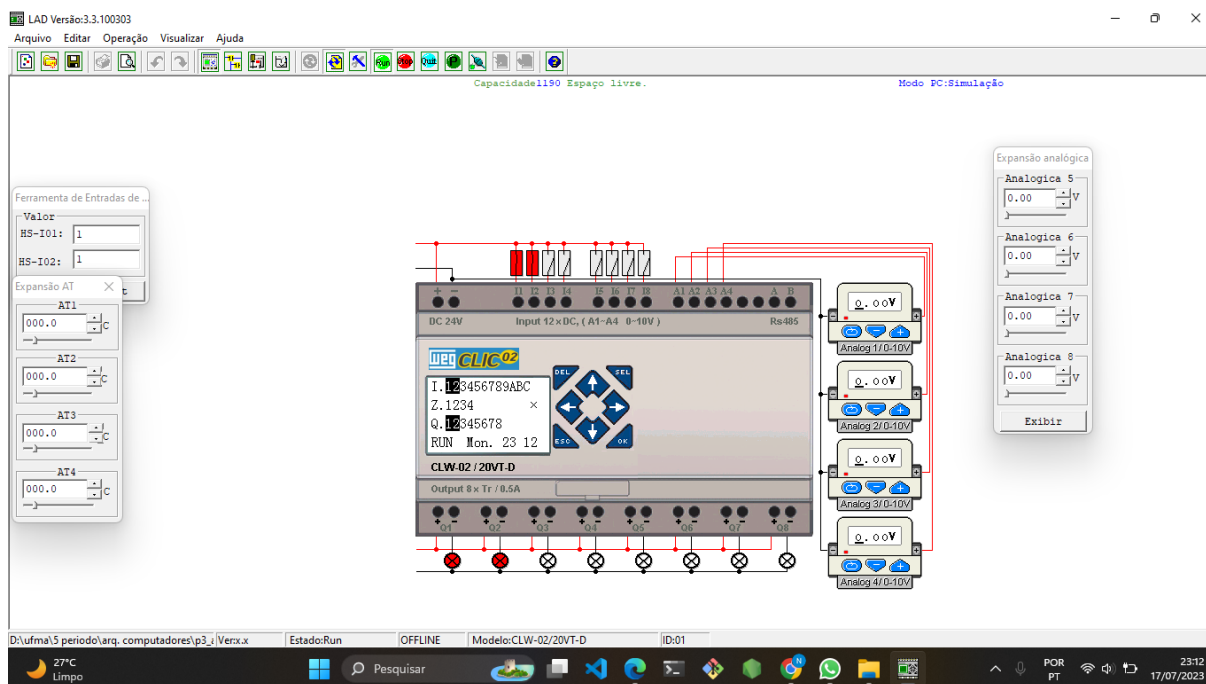
Figura 25 – Rodando programa em Ladder da questão 4



Fonte: Autoria Própria.

o programa no PLC na Figura 26 a seguir (WEG, s.d.).

Figura 26 – Rodando programa em Ladder da questão 4



Fonte: Autoria Própria.

3.4 Questão 5

Um motor deve ser acionado toda vez que a expressão booleana $S1$ for verdadeira por 6 vezes consecutivas. Quando o motor for desligado, a contagem deve ser reiniciada (pontos: 2,5).

$$S1 = A'BD + B'CDF + (B' + AE)D \quad (3.5)$$

Tabela verdade parte 1 da questão 5 na Figura 27 foi feita no Software Excel([Microsoft Corporation, 2018](#)).

Figura 27 – Tabela verdade parte 1 da questão 5

Tabela verdade:						
A	B	C	D	E	F	S1
0	0	0	0	0	0	0
0	0	0	0	0	1	0
0	0	0	0	1	0	0
0	0	0	0	1	1	0
0	0	0	1	0	0	1
0	0	0	1	0	1	1
0	0	0	1	1	0	1
0	0	0	1	1	1	1
0	0	1	0	0	0	0
0	0	1	0	0	1	0
0	0	1	0	1	0	0
0	0	1	0	1	1	0
0	0	1	1	0	0	1
0	0	1	1	0	1	1
0	0	1	1	1	0	1
0	0	1	1	1	1	1
0	1	0	0	0	0	0
0	1	0	0	0	1	0
0	1	0	0	1	0	0
0	1	0	0	1	1	0
0	1	0	1	0	0	1
0	1	0	1	0	1	1
0	1	0	1	1	0	1
0	1	0	1	1	1	1
0	1	1	0	0	0	0
0	1	1	0	0	1	0
0	1	1	0	1	0	0
0	1	1	0	1	1	0
0	1	1	1	0	0	1
0	1	1	1	0	1	1
0	1	1	1	1	0	1
0	1	1	1	1	1	1
1	0	0	0	0	0	0
1	0	0	0	0	1	0
1	0	0	0	1	0	0
1	0	0	0	1	1	0
1	0	0	1	0	0	1
1	0	0	1	0	1	1
1	0	0	1	1	0	1
1	0	0	1	1	1	1
1	0	1	0	0	0	0
1	0	1	0	0	1	0
1	0	1	0	1	0	0
1	0	1	0	1	1	0
1	0	1	1	0	0	1
1	0	1	1	0	1	1
1	0	1	1	1	0	1
1	0	1	1	1	1	1
1	1	0	0	0	0	0
1	1	0	0	0	1	0
1	1	0	0	1	0	0
1	1	0	0	1	1	0
1	1	0	1	0	0	1
1	1	0	1	0	1	1
1	1	0	1	1	0	1
1	1	0	1	1	1	1
1	1	1	0	0	0	0
1	1	1	0	0	1	0
1	1	1	0	1	0	0
1	1	1	0	1	1	0
1	1	1	1	0	0	1
1	1	1	1	0	1	1
1	1	1	1	1	0	1
1	1	1	1	1	1	1

Fonte: Autoria Própria.

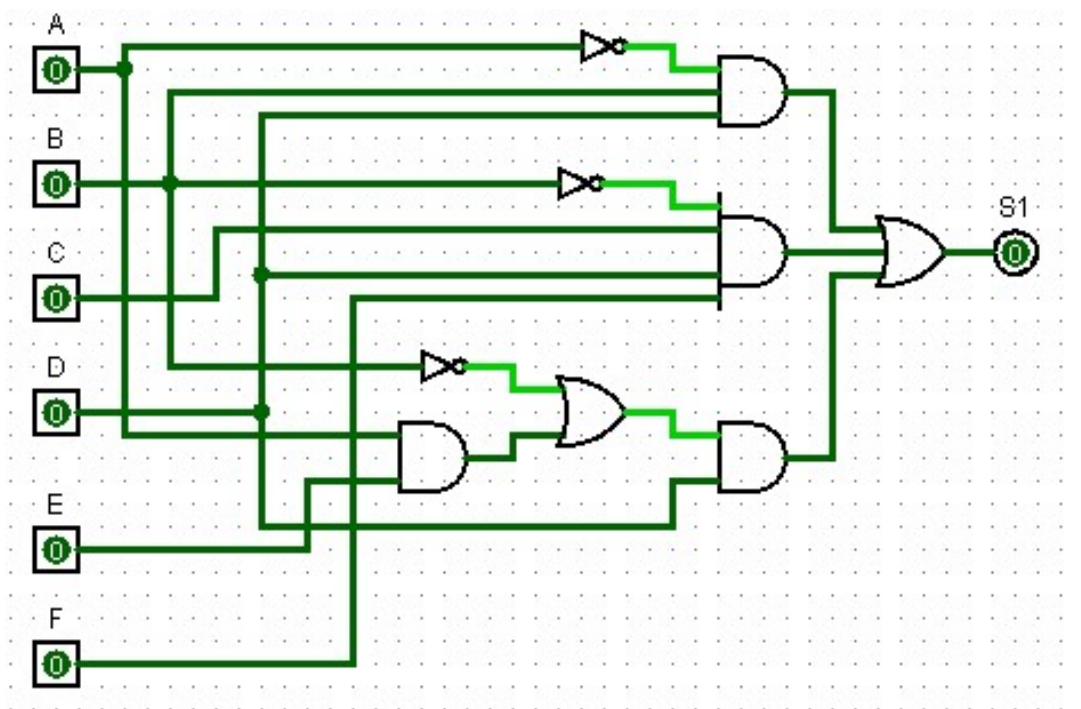
Figura 28 – Tabela verdade parte 2 da questão 5

1	0	0	0	0	0	0
1	0	0	0	0	1	0
1	0	0	0	1	0	0
1	0	0	0	1	1	0
1	0	0	1	0	0	1
1	0	0	1	0	1	1
1	0	0	1	1	0	1
1	0	0	1	1	1	1
1	0	1	0	0	0	0
1	0	1	0	0	1	0
1	0	1	0	1	0	0
1	0	1	0	1	1	0
1	0	1	1	0	0	1
1	0	1	1	0	1	1
1	0	1	1	1	0	1
1	0	1	1	1	1	1
1	1	0	0	0	0	0
1	1	0	0	0	1	0
1	1	0	0	1	0	0
1	1	0	0	1	1	0
1	1	0	1	0	0	0
1	1	0	1	0	1	0
1	1	0	1	1	0	1
1	1	0	1	1	1	1
1	1	1	0	0	0	0
1	1	1	0	0	1	0
1	1	1	0	1	0	0
1	1	1	0	1	1	0
1	1	1	1	0	0	0
1	1	1	1	0	1	0
1	1	1	1	1	0	0
1	1	1	1	1	1	0
1	1	1	1	1	0	1
1	1	1	1	1	1	1

Fonte: Autoria Própria.

De acordo com a Expressão booleana o circuito digital formado está na Figura a seguir.

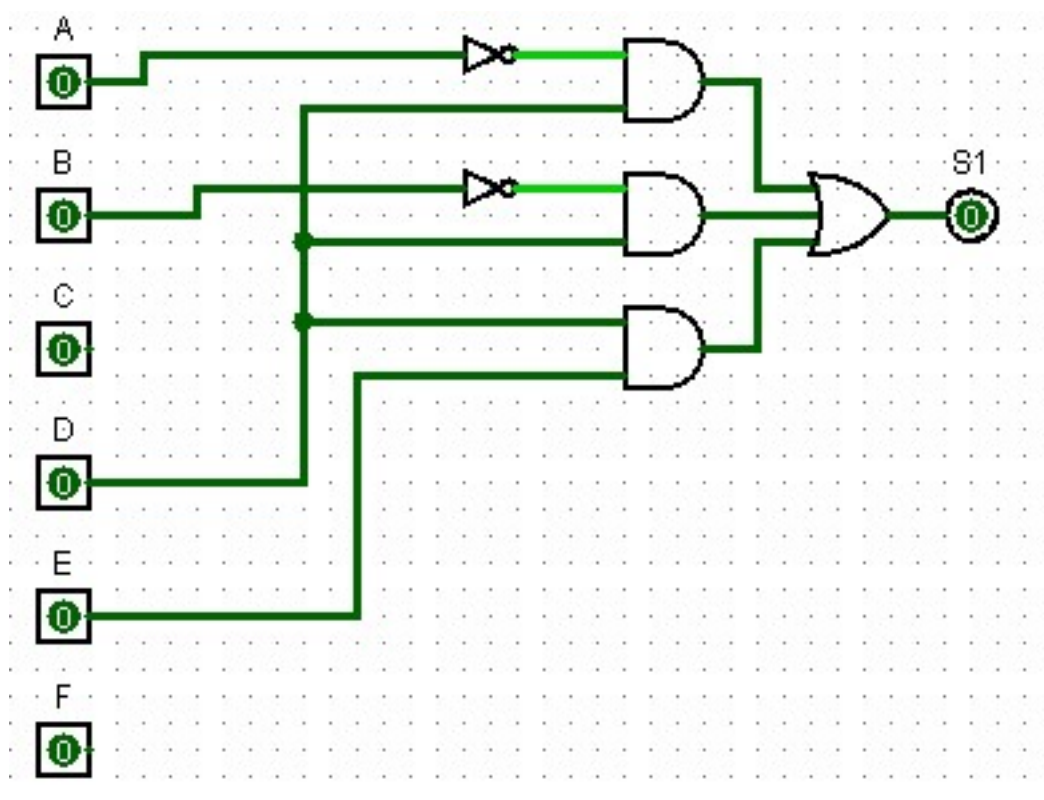
Figura 29 – Circuito logico no logisim



Fonte: Autoria Própria.

$$S1 = A'D + B'D + ED \quad (3.6)$$

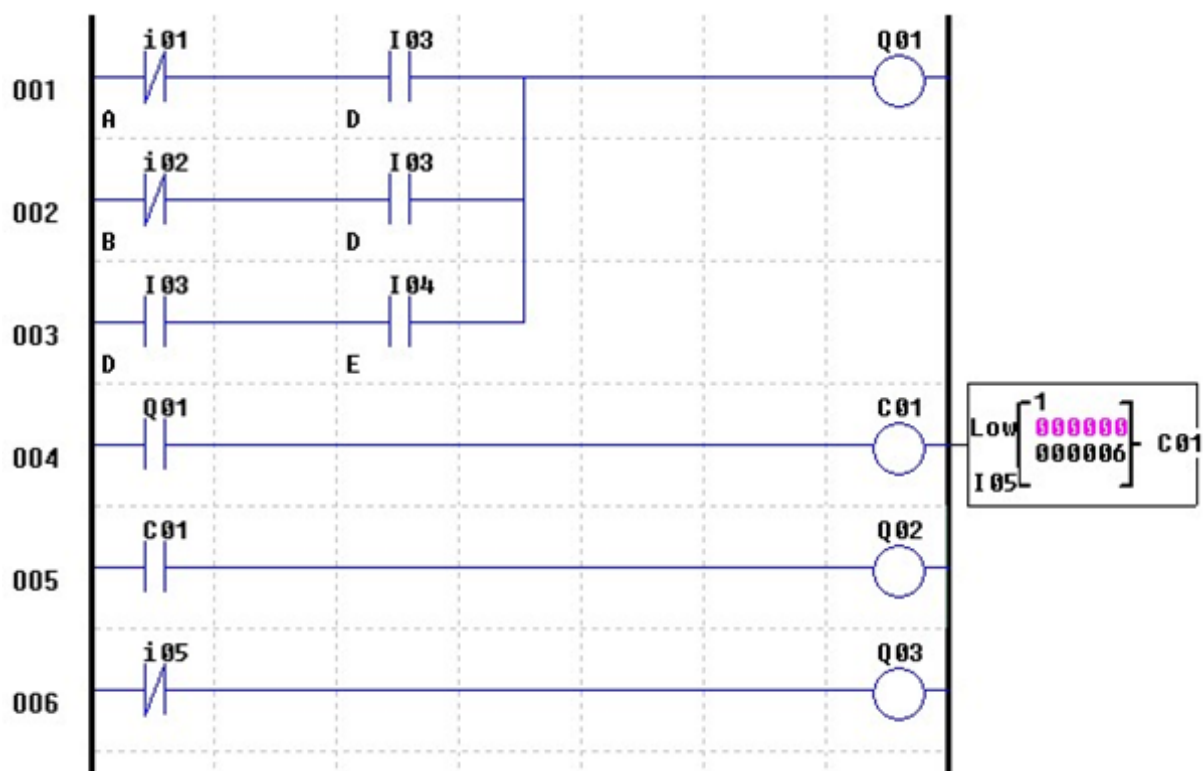
Figura 30 – Circuito logico no logisim simplificado



Fonte: Autoria Própria.

Na Figura 31 é possível ver o tempo colocado de acordo com a condição.

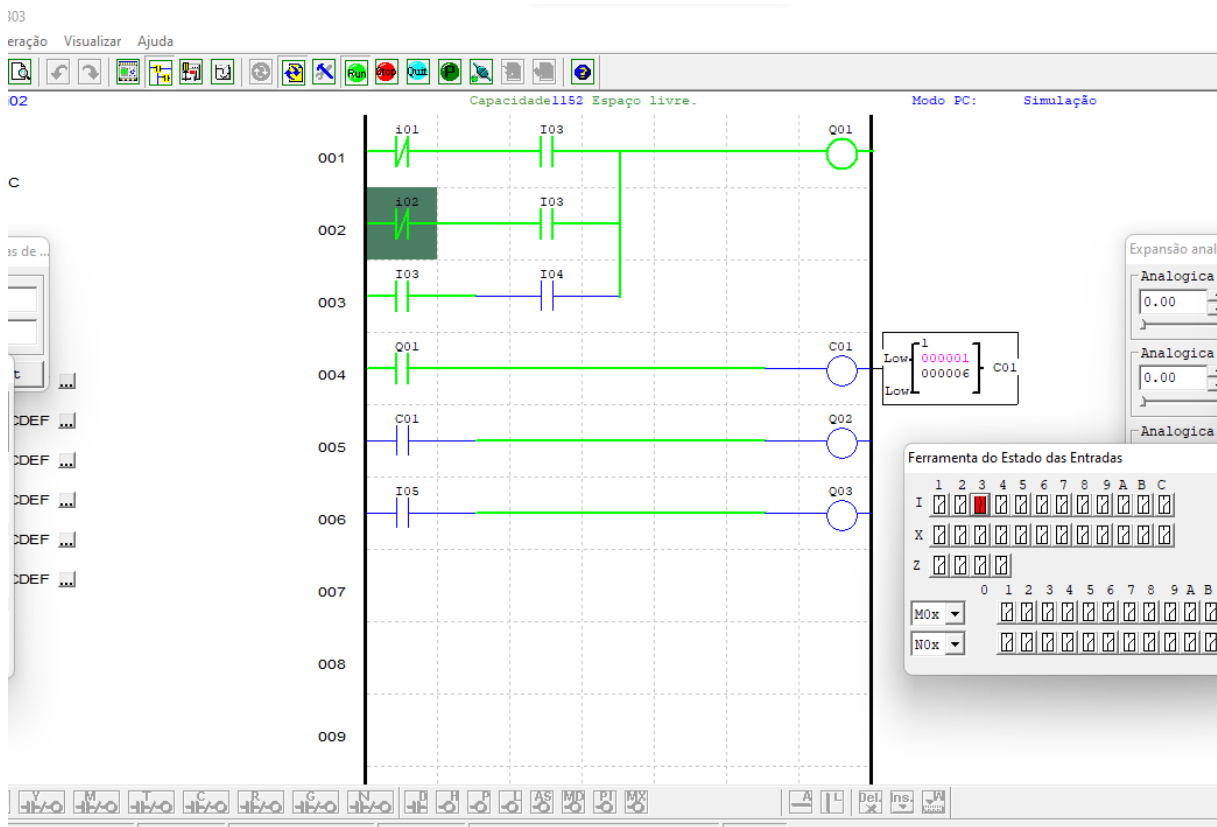
Figura 31 – Programa em Ladder



Fonte: Autoria Própria.

Rodando o programa é possível clicando nos botões que estão normalmente abertos ou fechados e observar as condições descritas no enunciado.

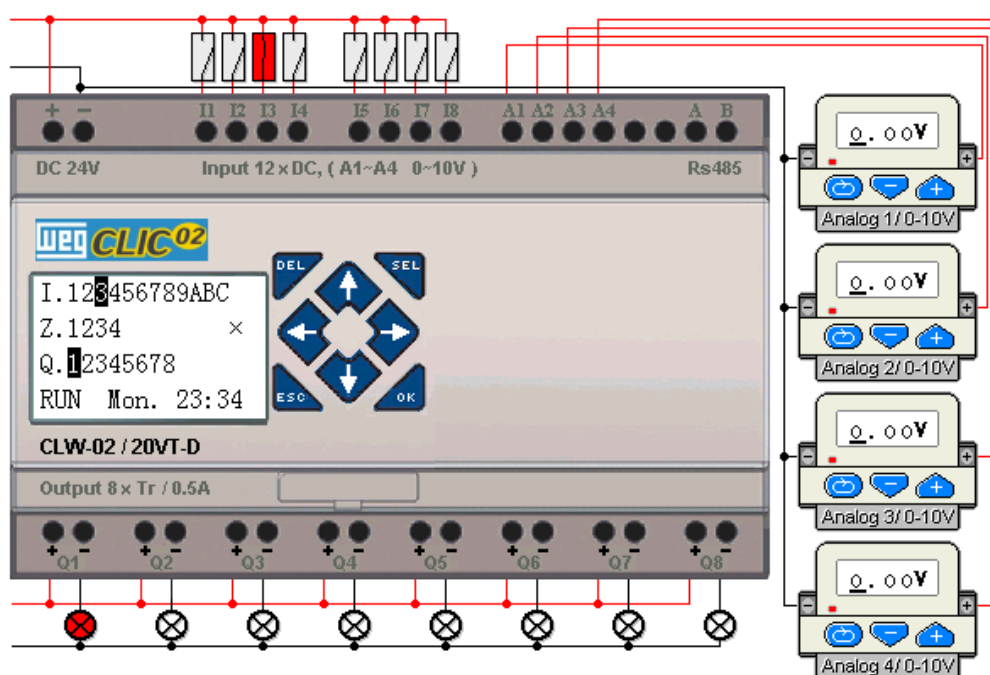
Figura 32 – Programa em Ladder



Fonte: Autoria Própria.

O PLC da questão 5 na Figura 33

Figura 33 – Programa em Ladder

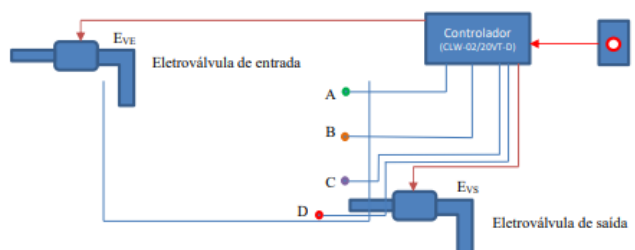


Fonte: Autoria Própria.

3.5 Questão 6

Solicita-se que seja concebido um sistema baseado no PLC da Weg “CLW-02/20VT-D” e na linguagem LADDER para implementar o controle da seguinte planta. (Pontos: 3,33)

Figura 34 – esquema do Tanque



Fonte: Autoria Própria.

Um reservatório para armazenar um determinado produto líquido deve ser controlado da seguinte forma:

- Ao pressionar o botão I, o sistema deve ser acionado;
- Uma vez que o sistema tenha sido acionado, a eletroválvula Evs deve ser ligada se, e somente se, o líquido estiver no nível A ou nível B. Se o líquido estiver no nível D, então a eletroválvula Evs deve ser desligada até que o nível B seja atingido;
- A eletroválvula Eve deve ser ligada se, e somente se, o botão I estiver ligado e o nível do líquido estiver no nível D ou C ou B. No nível A, a eletroválvula Eve deve ser desligada;
- Caso o sistema esteja ligado e seja pressionado o botão I para desligá-lo, então a Eletroválvula Eve deve ser desligada. Em seguida, todo o líquido deve ser escoado pela Eletroválvula Evs, até que o sensor D acuse reservatório vazio (sensor desativado).

OBS: Utilize as seguintes entradas e saídas do PLC:

- I1 como entrada para I;
- I2 como entrada para A;
- I3 como entrada para B;
- I4 como entrada para C;
- I5 como entrada para D;
- Q1 como saída para Evs;
- Q2 como saída para Eve;
- Q3 como sinal de alerta para mau funcionamento do sistema, por exemplo: sensor A e B ativos, mas sensor C desativado; ou sensor A ativado, mas sensores B e C desativados; ou B ativado, mas sensor C desativado.

OBS: Faça a tabela da verdade, depois obtenha a expressão simplificada pelo mapa de Karnaugh, faça o fluxograma da solução do problema, escreva o programa em linguagem LADDER para o PLC da Weg “ $CLW_{stx}02/20VT - D$ ”.

Figura 35 – tabela verdade do esquema do Tanque

PB0(I)	PB3(D)	PB2(C)	PP1(B)	PP0(A)	PB4(Evs)	PB5(Eve)	PB7(Erros)
0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0
0	0	0	1	0	0	0	0
0	0	0	1	1	0	0	0
0	0	1	0	0	0	0	0
0	0	1	0	1	0	0	0
0	0	1	1	0	0	0	0
0	0	1	1	1	0	0	0
0	1	0	0	0	1	0	0
0	1	0	0	1	1	0	0
0	1	0	1	0	1	0	0
0	1	0	1	1	1	0	0
0	1	1	0	0	1	0	0
0	1	1	0	1	1	0	0
0	1	1	1	0	1	0	0
0	1	1	1	1	1	0	0
1	0	0	0	0	0	1	0
1	0	0	0	1	0	0	1
1	0	0	1	0	0	0	1
1	0	0	1	1	0	0	1
1	0	1	0	0	0	0	1

Fonte: Autoria Própria.

Figura 36 – continuação da tabela verdade

1	0	1	0	1	0	0	1
1	0	1	1	0	0	0	1
1	0	1	1	1	0	0	1
1	1	0	0	0	0	1	0
1	1	0	0	1	0	0	1
1	1	0	1	0	0	0	1
1	1	0	1	1	0	0	1
1	1	1	0	0	0	1	0
1	1	1	0	1	0	0	1
1	1	1	1	0	1	1	0
1	1	1	1	1	1	0	0

Fonte: Autoria Própria.

O terceiro experimento consiste em fazer o controle de um tanque conforme o enunciado. A seguir é possível ver a imagem de ilustração do tanque. Tabela verdade e obtenção da expressão simplificada: O primeiro passo é construir a tabela verdade que

relaciona as entradas (sensores de nível A, B, C e D e o interruptor I) com as saídas (ligamento das eletroválvulas Evs e Eve). Com base na tabela verdade, é possível obter a expressão simplificada por meio do mapa de Karnaugh. Essa expressão simplificada será utilizada para controlar o ligamento das eletroválvulas.

Entrada dos sinais dos sensores de nível e interruptor I: Os sensores de nível A, B, C e D, juntamente com o interruptor I, são utilizados como entradas para o programa. Os sensores de nível são lidos pelas portas PB0, PB1, PB2 e PB3 do microcontrolador HCS12, respectivamente, enquanto o interruptor I é lido pela porta PB0. Esses valores são interpretados como sinais de entrada no programa para determinar as condições de controle.

Ligamento das eletroválvulas de saída (Evs e Eve): Com base nas condições especificadas, o programa controla o ligamento das eletroválvulas de saída Evs (ligada à porta PB4) e Eve (ligada à porta PB5). O ligamento das eletroválvulas é determinado pelas condições estabelecidas na premissa, como o nível do líquido e o estado do interruptor I. O programa liga ou desliga as eletroválvulas de acordo com essas condições.

Sinal de alerta para mau funcionamento do sistema: Além do controle das eletroválvulas, o programa também utiliza a porta PB7 para emitir um sinal de alerta em caso de mau funcionamento do sistema. Isso é feito com base nas condições especificadas, como sensores ativados ou desativados de forma inadequada. O programa monitora essas condições e emite um sinal de alerta, se necessário. O programa em linguagem C para o microcontrolador HCS12 deve ser implementado seguindo o fluxograma que representa a solução do problema. A implementação detalhada desse programa dependerá das especificações e recursos disponíveis no microcontrolador HCS12, bem como das bibliotecas e funções disponíveis na linguagem de programação utilizada.

Na Tabela 3 abaixo conseguimos ver os resultados da tabela verdade que soluciona o problema da questão 5. As saídas e entradas estão descritas como pedidas pela questão.

Figura 37 – Mapa k da questão 5 da válvula de entrada

*Mapa de Karnaugh de Eve(PB5) :		$IC'B'A'+IDCA'$			
I		CD			
		00	01	11	10
AB	00	1	1	1	0
	01	0	0	1	0
	11	0	0	0	0
	10	0	0	0	0

Fonte: Autoria Própria.

Figura 38 – Mapa k da questão 5 da válvula de saída

*Mapa de Karnaugh de Evs(PB4) :		$I'D+DCB$			
		CB			
		00	01	11	10
ID	00	0	0	0	0
	01	1	1	1	1
	11	0	0	1	0
	10	0	0	0	0

Fonte: Autoria Própria.

Figura 39 – Mapa k da questão 5 de condições de erro

*Mapa de Karnaugh de Erros(PB7) :		$ID'C+IC'B+IB'A$			
I		CD			
		00	01	11	10
AB	00	0	0	0	1
	01	1	1	0	1
	11	1	1	0	1
	10	1	1	1	1

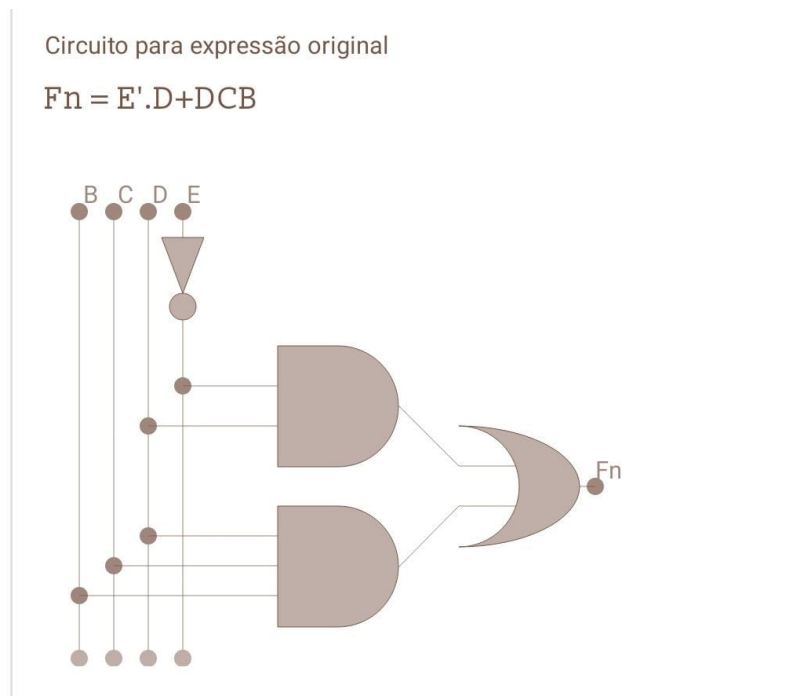
Fonte: Autoria Própria.

Para fazer o circuito foi através das expressões booleana para cada caso foi utilizado o software Boolean TT ([BOOLEANTT.HAZEAPPS.COM](https://boolean.tt.hazeapps.com), 2023).

evs:

$$I'D + DCB \quad (3.7)$$

Figura 40 – circuito evs

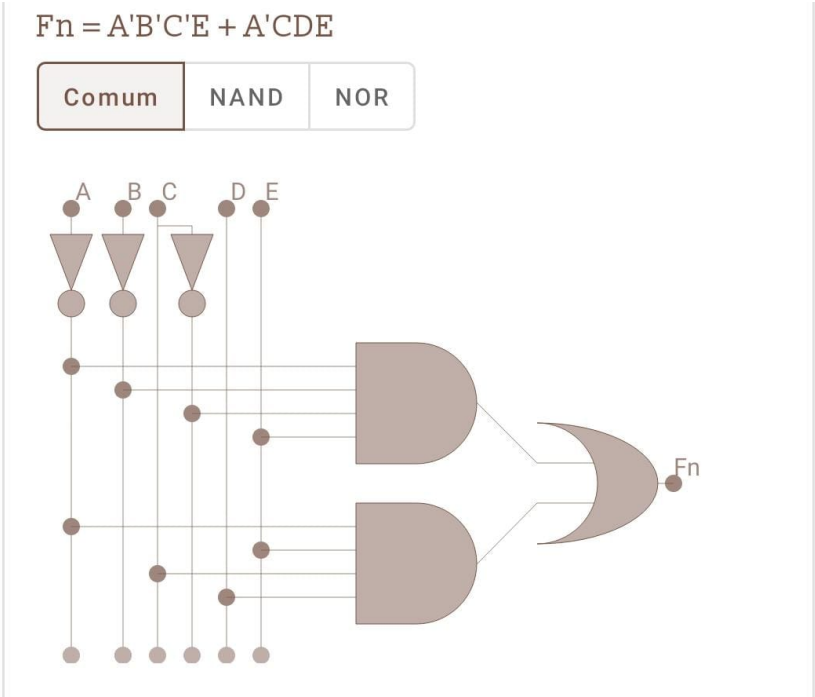


Fonte: Autoria Própria.

eve:

$$I(C'B'A' + DCA') \quad (3.8)$$

Figura 41 – circuito eve

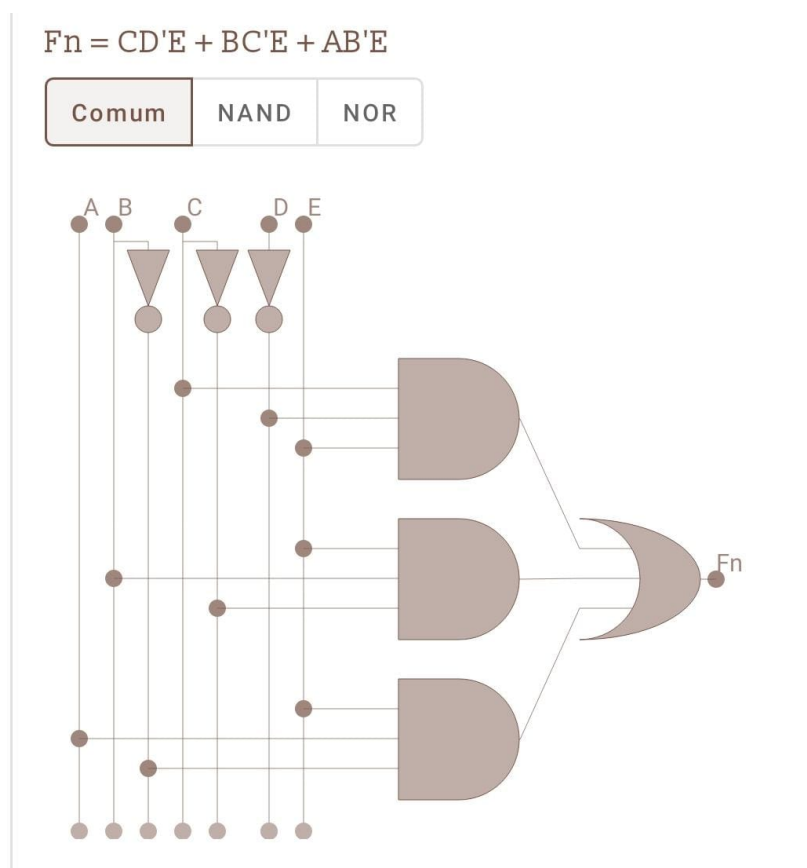


Fonte: Autoria Própria.

erro:

$$I(D'C + C'B + B'A) \tag{3.9}$$

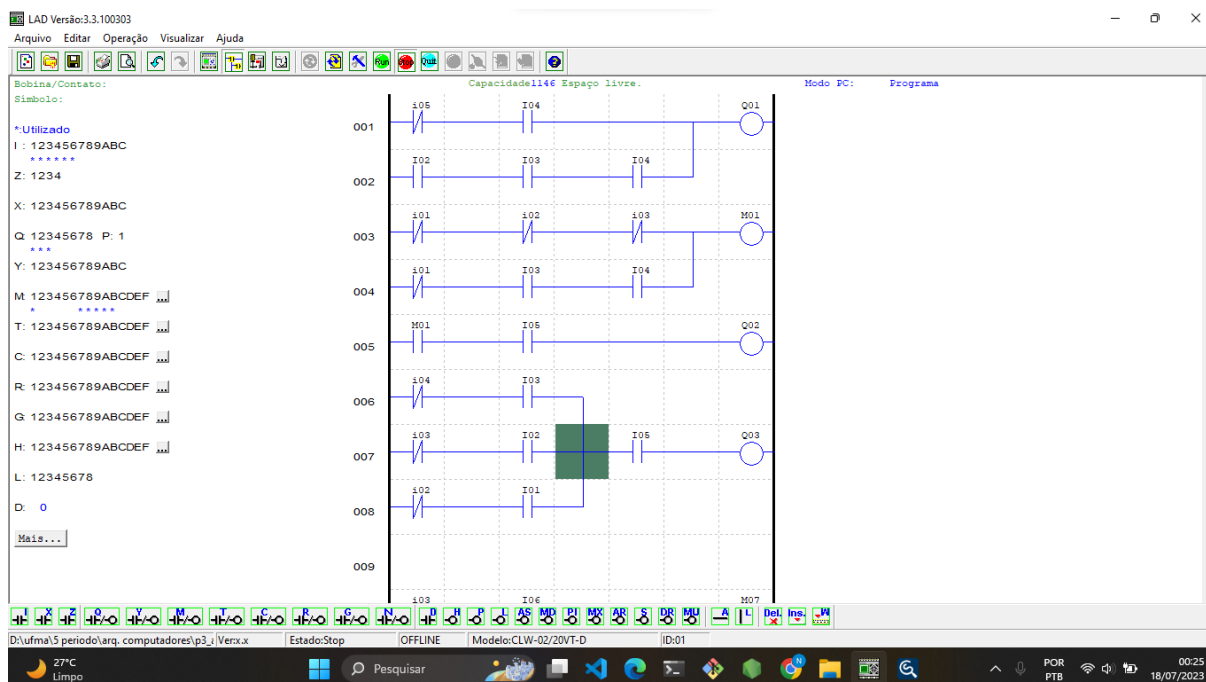
Figura 42 – circuito erro



Fonte: Autoria Própria.

o programa em Ladder da questão 6:

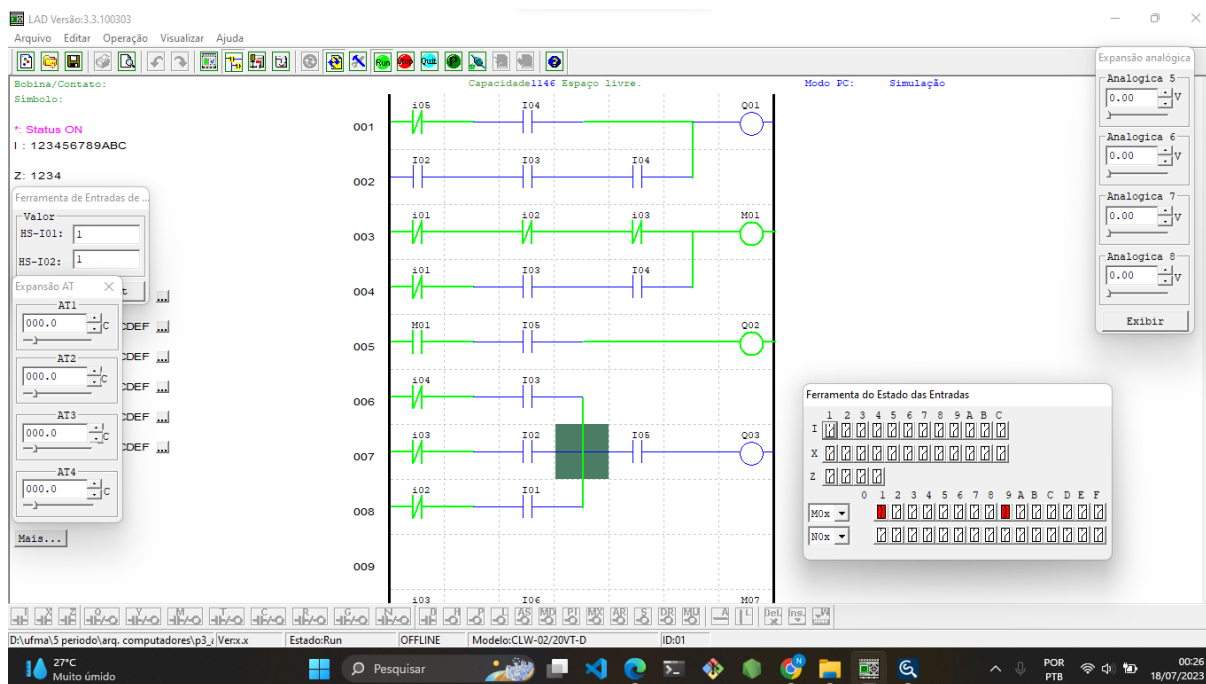
Figura 43 – programa Ladder questão 6



Fonte: Autoria Própria.

Rodando o programa em Ladder.

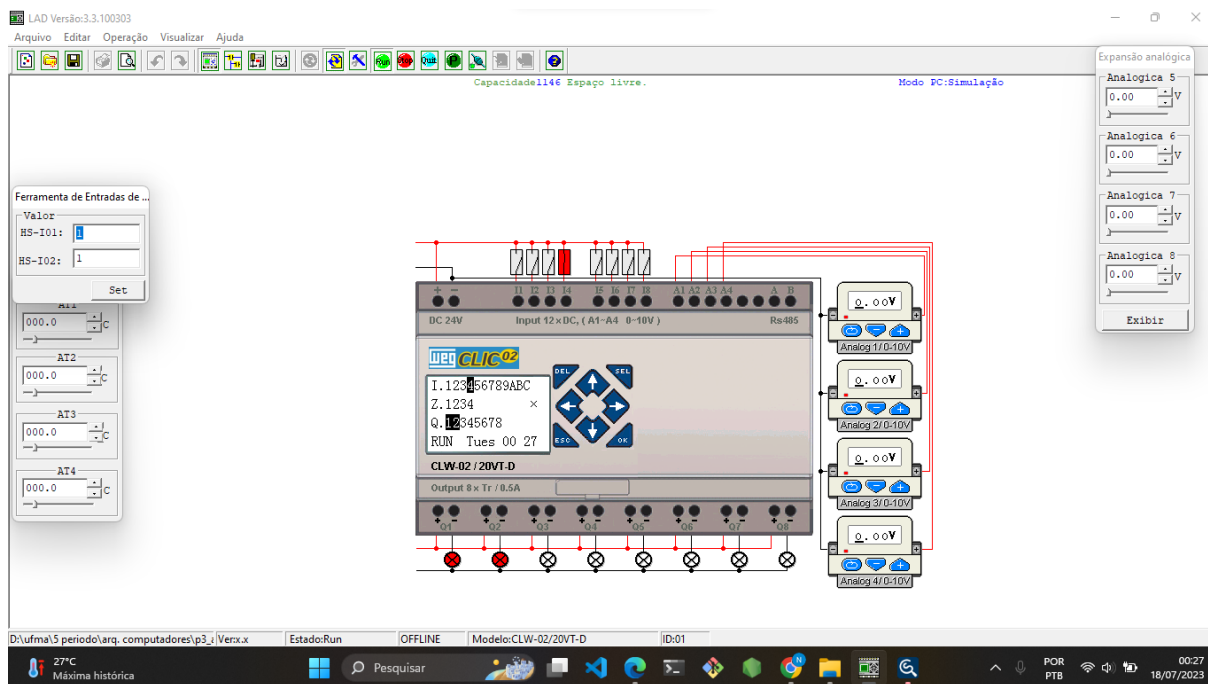
Figura 44 – Rodando programa Ladder questão 6



Fonte: Autoria Própria.

Visualizando o PLC.

Figura 45 – questão 6 no PLC



Fonte: Autoria Própria.

4 Conclusão

Ao longo deste trabalho, foram exploradas as funcionalidades e características do PLC, juntamente com a apresentação de soluções para os problemas encontrados durante o desenvolvimento de projetos utilizando o ambiente de desenvolvimento Clik Edit. Os experimentos e questões foram solucionados com sucesso, resultando nos resultados esperados obtidos nas simulações dentro do Click Edit.

Durante o trabalho, foram abordados diferentes aspectos do PLC, desde sua arquitetura e recursos até a implementação de soluções práticas para problemas no simulador de programação Ladder. Foram exploradas técnicas e estratégias para lidar com erros de compilação, depuração de código, otimização de desempenho e integração de bibliotecas externas. Através dos experimentos foi possível compreender de forma clara e prática os conhecimentos adquiridos de forma teórica na disciplina de arquitetura de computadores.

Todos os passos seguidos e as dificuldades para concluir os projetos serviram de grande aprendizado de como é o dia a dia do engenheiro eletricista e como deve-se agir diante de tais situações. Com os experimentos trazidos pelo professor ficou evidente as diversas aplicações de microcontroladores no cotidiano de diversas empresas e como é possível aproveitar situações vivenciadas na indústria para implementar tecnologias que auxiliem um usuário e dê a ele uma maior praticidade para realizar suas tarefas.

Em resumo, os objetivos propostos foram alcançados com sucesso, demonstrando a eficácia do microcontrolador PLC e do ambiente Click Edit na criação e desenvolvimento de sistemas embarcados. Os resultados obtidos nas simulações confirmaram as expectativas e destacaram a importância dessas tecnologias no avanço da tecnologia em diversas áreas industriais. Assim, foi possível compreender de forma clara e prática os conhecimentos adquiridos de forma teórica em outras disciplinas. Todos os passos seguidos e as dificuldades para concluir os projetos serviram de grande aprendizado de como é o dia a dia do engenheiro eletricista e como deve-se agir diante de tais situações. Com os experimentos trazidos pelo professor ficou evidente as diversas aplicações de PLC no cotidiano de diversas empresas e como é possível aproveitar situações vivenciadas na indústria para implementar tecnologias que auxiliem um usuário e dê a ele uma maior praticidade para realizar suas tarefas.

Referências

APPLE INC. *Boolean Algebra*. 2023. <<https://apps.apple.com/us/app/boolean-algebra/id1512829475>>. Acesso em: 18 de julho de 2023. Citado 3 vezes nas páginas 10, 17 e 24.

ASPAR, Z.; SHAIKH-HUSIN, N.; KHALIL-HANI, M. Algorithm to convert signal interpreted petri net models to programmable logic controller ladder logic diagram models. *Indonesian Journal of Electrical Engineering and Computer Science*, v. 10, n. 3, p. 905–916, 2018. Citado na página 6.

BENITEZ-PINA, I. F.; LAMAR-CARBONELL, S.; SILVA, R. M. d.; EIGI-MIYAGI, P.; SILVA, J. R. Design of automatic control system based on unified timed hybrid petri net. *Dyna*, v. 84, n. 200, p. 80–89, 2017. Citado na página 6.

BOOLEANTT.HAZEAPPS.COM. *BooleanTT - Beta - Online Boolean Algebra*. 2023. <<https://booleantt.hazeapps.com/>>. Acesso em: 18 de julho de 2023. Citado 2 vezes nas páginas 18 e 39.

C. Burch. *Logisim: Software de simulação de circuitos digitais*. 2011. Versão estável: 2.7.1. Disponível em: <<http://www.cburch.com/logisim/pt/index.html>>. Citado 2 vezes nas páginas 8 e 19.

FRANCISCO, B. I. Design of automatic control system based on unified timed hybrid petri net. *DYNA*, v. 84, 2017. Disponível em: <<https://www.redalyc.org/journal/496/49650910010/html/>>. Citado na página 6.

Microsoft Corporation. *Microsoft Excel*. 2019. ed. Redmond, 2018. Citado na página 27.

SOUZA, M. F. d. et al. Modelagem e verificação de programas de clp escritos em diagrama ladder. 2012. Citado na página 6.

WEG. *WEG. Clic02 Edit*. s.d. Versão 3.41. Disponível em: <<http://www.weg.net>>. Citado 4 vezes nas páginas 6, 9, 21 e 26.

ZHOU, M.; TWISS, E. Design of industrial automated systems via relay ladder logic programming and petri nets. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, IEEE, v. 28, n. 1, p. 137–150, 1998. Citado na página 6.