

ESTUDO E APLICAÇÃO DE LINGUAGENS DE PROGRAMAÇÃO UTILIZANDO O SOFTWARE CODESYS

Lucas Carvalho Souza¹
André Luiz Silva Pereira²

¹IFG/Jataí/Engenharia Elétrica - PIBITI, lucas.souza@ifg.edu.br

²IFG/Jataí /Engenharia Elétrica, andspa@gmail.com

Resumo

Atualmente a automação se tornou indispensável para o crescimento da sociedade moderna, possibilitando melhoria na produtividade e qualidade nos processos considerados repetitivos, proporcionando um aumento da produção e redução de custo com energia e matéria prima de um processo industrial. O principal equipamento deste processo é o CLP (Controlador Lógico Programável), sendo considerado como um computador projetado para trabalhar no ambiente industrial, que desempenha funções de controle através de softwares desenvolvidos pelo usuário. O grande problema dos CLPs é que cada fabricante possui software próprio, sem manter um determinado padrão ou modelo, utilizando-se de linguagens de programação diferentes que impossibilitam a portabilidade de código, ou seja, atualmente o programa desenvolvido para determinado equipamento não pode ser utilizado diretamente em outro. Visando uniformizar as várias metodologias de programação dos CLPs, a IEC (International Electro-Technical Commission) através da norma 61131-3 definiu sintática e semanticamente cinco linguagens de programação sendo elas: Ladder, Diagrama de Blocos de Funções, Sequenciamento Gráfico de Funções, Lista de Instruções e Texto Estruturado. Na atual conjuntura o software Codesys é um dos poucos softwares existentes a atender todos os requisitos da norma 61131-3.

Palavras-chave: Codesys, Linguagens de Programação, CLP, Automação, Programação

1. INTRODUÇÃO

A automação surgiu como um meio para simplificar a participação do trabalho humano sobre os processos industriais. Entende-se por automação um conceito e um conjunto de técnicas por meio das quais se constroem sistemas ativos capazes de atuar com uma eficiência ótima pelo uso de informações recebidas do meio sobre o qual atuam (SILVEIRA, 1998).

Durante a década de 60, foi desenvolvido a partir de uma demanda existente na indústria automobilística norte-americana, o *Programmable Logic Controller* (PLC) ou Controlador Lógico Programável (CLP). O CLP é a principal ferramenta de um processo industrial automatizado. Segundo a NEMA (National Electrical Manufacturers Association), o CLP é um aparelho eletrônico digital que utiliza uma memória programável para armazenar internamente instruções para implementar funções específicas, tais como lógica, sequenciamento, temporização, contagem e aritmética, controlando, por meio de módulos de entradas e saídas, vários tipos de máquinas ou processos.

Os Controladores Programáveis proporcionam para as fábricas modernas uma estrutura de processo automatizado, que se apoiam em dispositivos de hardware e software, combinados de forma organizada, que permite um controle total acerca das informações envolvidas. O Controlador Programável monitora o estado das entradas e saídas de acordo com às instruções programadas na memória, e energiza ou desenergiza as saídas, conforme o resultado lógico obtido através das instruções do programa. O programa pode ser entendido como uma sequência de instruções a serem executadas pelo CLP para realizar um determinado processo.

Existem no mercado de automação, nacional e mundial, vários fabricantes de CLP's, cada um com sua arquitetura, rede de comunicação, linguagem e software de programações diferentes. Uma Linguagem de programação pode ser entendida como um conjunto padronizado de instruções que o sistema computacional é capaz de reconhecer (FRANCHI e CAMARGO, 2008). O problema dos CLP's é que cada fabricante possui software próprio, sem manter um determinado padrão ou modelo, utilizando-se de linguagens de programação diferentes que impossibilitam a portabilidade de código, ou seja, atualmente o programa desenvolvido para determinado equipamento não pode ser utilizado diretamente em outro. Isto pode acarretar perda de tempo e elevação dos custos de produção.

Para uniformizar as várias metodologias de programação dos CLP's, a *International Electrotechnical Commission* (IEC) publicou a norma IEC 61131-3 (1993), a qual define regras de semântica e sintaxe para as linguagens de programação mais comuns, que podem ser empregadas em dispositivos comerciais, facilitando serviços de manutenção e desenvolvimento de projetos para que os fabricantes possam expandir e adaptar estas regras para suas próprias implementações em CLP's. A norma define cinco linguagens de programação, sendo elas: Ladder, Diagrama de Blocos de Funções, Sequenciamento Gráfico de Funções, Lista de Instruções e Texto Estruturado.

Atualmente o software de programação Codesys, é um dos poucos softwares existentes padronizados de acordo com a IEC 61131-3, que oferece uma ferramenta de automação avançada, contendo inúmeras funções pré-definidas e métodos para desenvolvedores de aplicação. Com o intuito de elaborar uma apostila didática, o trabalho teve como objetivo realizar um estudo do software Codesys. Para isso, teve-se o propósito de dominar as cinco linguagens de programação estabelecidas pela IEC 61131-3, simulando uma aplicação de um processo industrial para cada linguagem de programação. A elaboração da apostila didática tem o intuito de disseminar conhecimento na área de automação industrial para o ensino técnico profissionalizante, alunos de graduação e profissionais de indústria.

2. Norma IEC 61131

Em 1993, a *International Electrotechnical Commission* (IEC) publicou a terceira parte da norma IEC 61131, a IEC 61131-3 estabelecendo um padrão global para programação de Controladores Lógicos Programáveis. Essa norma estabelece que para um CLP estar de acordo com os padrões por ela estabelecidos, deve possuir no mínimo duas linguagens gráficas e duas linguagens de texto para sua programação. Com isso, definiu-se uma interface padrão permitindo que usuários com diferentes habilidades e formações, criem programas.



Hoje, a IEC 61131-3 é o único padrão mundial para programação de controle industrial que consiste na definição das linguagens de programação:

- Textuais:

IL - *Instruction List* ou Lista de Instruções.

ST - *Structured text* ou Texto Estruturado.

- Gráficas:

LD - *Ladder Diagram* ou Diagrama Ladder.

FBD - *Function Block Diagram* ou Diagrama de Blocos de Funções.

SFC – *Sequential Function Chart* ou Sequenciamento Gráfico de Funções.

3. PROGRAMAÇÃO NO SOFTWARE CODESYS

O Codesys é uma plataforma de software especialmente projetado para atender as necessidades diferentes de projetos modernos de automação industrial. O software foi desenvolvido com os requisitos da norma IEC 61131-3. Ele oferece fácil utilização para apoiar o usuário no desenvolvimento da programação.

O programa foi adquirido gratuitamente da fabricante alemã *3S-Smart Software Solutions*, sendo instalado em um microcomputador.

A Tabela 1, lista os softwares dos principais fabricantes de CLP, e faz uma análise quanto aos princípios de portabilidade e reaproveitamento da norma IEC 61131-3.

| Avaliação de Softwares de CLP's Segundo Critérios da Norma IEC 61131-3 | | | | | | | |
|--|------------|----------------|----------------|----------------|--------------------------------|-----|------------|
| Fabricante | Software | Modularização | Estruturação | Reutilização | Linguagens | POU | Modelo IEC |
| WEG | CLIC 02 | insatisfatório | insatisfatório | insatisfatório | Apenas LD e FDB | não | não |
| Siemens | STEP 7 | insatisfatório | insatisfatório | insatisfatório | Apenas LD e IL | não | não |
| Schneider | Twido Soft | insatisfatório | insatisfatório | insatisfatório | Apenas LD e IL | não | não |
| Schneider | Zélio Soft | insatisfatório | insatisfatório | insatisfatório | Apenas LD e SFC | não | não |
| 3 S | CoDeSys | OK | OK | OK | Todas as 5 linguagens da norma | sim | sim |

Tabela 1: Avaliação de Softwares de CLP's Segundo Critérios da Norma IEC 61131-3

Conceitos utilizados na tabela:

- Modularização – decomposição de qualquer sistema, complexo ou não, em partes menores capazes de serem gerenciáveis;

- Estruturação – forma hierárquica utilizada para a programação em níveis facilitando a modularização e reutilização de blocos;
- Reutilização – de funções, de blocos funcionais ou programas;
- POU – módulo que integra programas, blocos funcionais e funções, chamado de Unidades de Organização de Programas – *Program Organization Units*, essa mesclagem de elementos na POU dão o poder da reutilização de códigos desde o nível micro ao macro.

O software permite fazer uma compilação do programa, e testá-lo virtualmente, dispensando o uso de um CLP para a fase de testes. Pode-se monitorar as variáveis do processo a qualquer momento, sendo possível a criação de uma Interface Homem Máquina (IHM) a fim de interagir com o sistema, verificando as variáveis e as condições de processamento. A Figura 1 retrata a edição IHM no software Codesys.

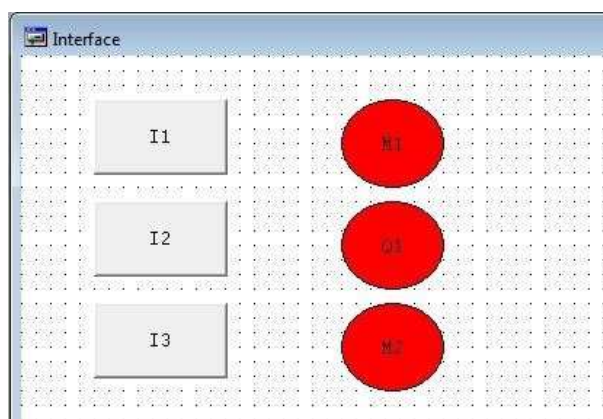


Figura 1: Edição IHM no software Codesys

Todas as simulações no trabalho, utilizando as cinco linguagens de programação, foram feitas no software Codesys de forma sucinta para um aprendizado básico do usuário que utilizará a apostila. A simulação a ser implementada consiste na contagem de objetos em uma esteira rolante, que deverá ser acionada ao ligar o processo. Ao contar 4 objetos, o processo deverá parar por um tempo de cinco segundos (tempo de retirada dos objetos). Contados cinco segundos o processo inicia novamente. Caso o processo parar durante a contagem dos objetos, o mesmo deverá recomeçar a contagem quando for energizado novamente. A seguir serão apresentadas as linguagens de programação de acordo com a interface do software Codesys.

3.1 DIAGRAMA LADDER

A linguagem gráfica Ladder é a mais popular, de fácil entendimento e visualização. Baseada em diagrama de elétrico de contatos e graficamente próxima do entendimento do mantenedor.



Nessa linguagem utiliza-se bobinas e contatos. Os símbolos de contatos programados em uma linha representam as condições que serão avaliadas de acordo com a lógica. Como resultado determinam o controle de uma saída, que normalmente é representada pelo símbolo de uma bobina (FRANCHI e CAMARGO, 2008). O Diagrama Ladder (LD) baseia-se em uma série de redes, uma rede é limitada nos lados direito e esquerdo por uma linha vertical. No meio está um diagrama de circuito composto de contatos, bobinas e linhas de conexão (Altus, 2009).

O ambiente de programação em LD no software pode ser visualizado na Figura 2.

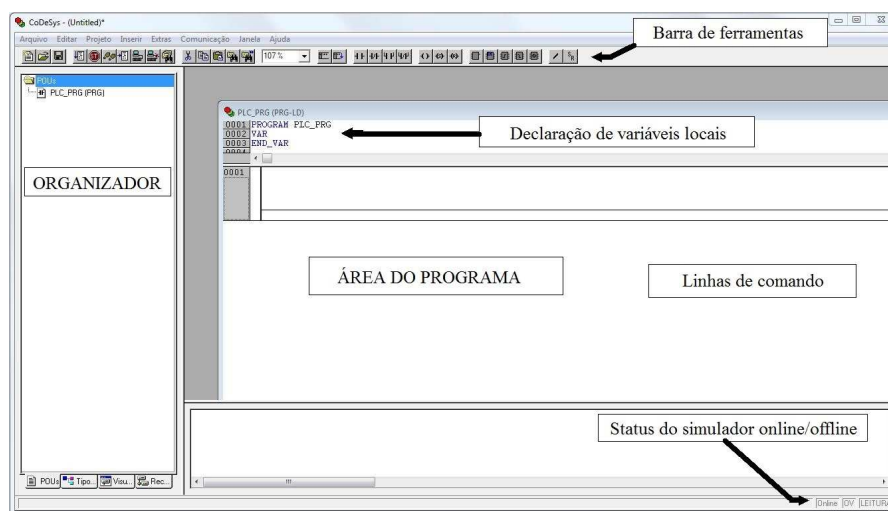


Figura 2: Tela de edição em LD

Após o desenvolvimento do processo utilizando o Diagrama Ladder, o programa pode ser visto na Figura 3:

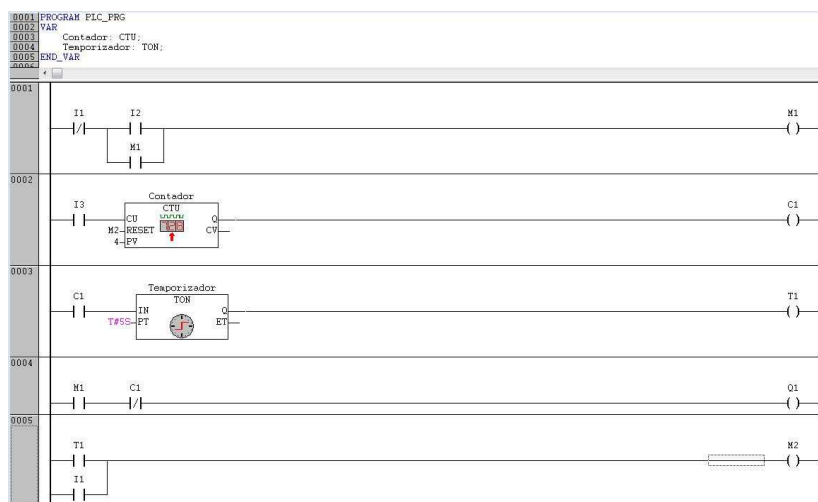


Figura 3: Programação desenvolvida em LD



3.2 DIAGRAMA DE BLOCOS DE FUNÇÕES

É uma linguagem gráfica baseada em diagramas lógicos de circuitos, de fácil entendimento, desde que o programador esteja familiarizado com representações gráficas das lógicas de Boole. Um bloco funcional é representado por um retângulo com entradas no seu lado esquerdo e saídas no direito.

As entradas e saídas são conectadas a esses blocos criando uma malha de interconexões que possibilita a obtenção dos mesmos resultados de outras lógicas de programação. A linguagem é próxima a Ladder, com exceção da representação gráfica e algumas funcionalidades específicas de cada linguagem, o que pode ser observado na Figura 4 que representa a edição da simulação feita em Blocos de Funções.

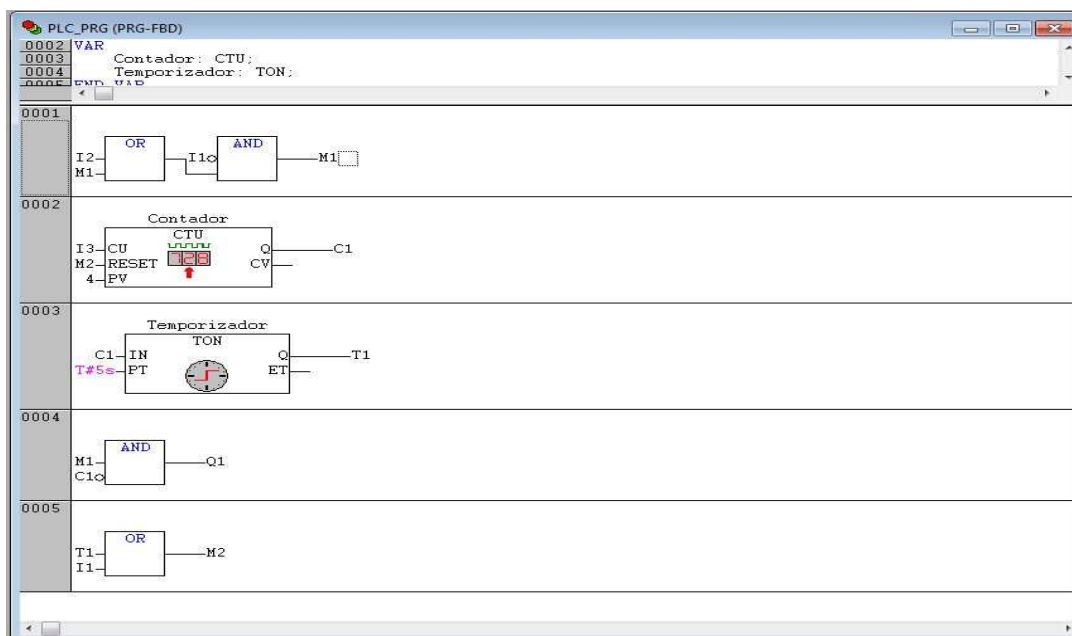


Figura 4: Programação desenvolvida em FBD

Essa linguagem 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, normalmente os blocos são construídos utilizando a linguagem de texto estruturado (FRANCHI e CAMARGO, 2008).

3.3 SEQUENCIAMENTO GRÁFICO DE FUNÇÕES

É uma linguagem gráfica muito poderosa, proporciona uma representação das seqüências do processo controlado na forma de um diagrama. O Sequenciamento Gráfico de Funções (SFC) é utilizado para dividir um problema de controle, permitindo uma visão geral do processo e facilitando o diagnóstico.



O SFC é descendente direto do Grafcet, fornece os meios para estruturar uma unidade de organização de um programa num conjunto de etapas separadas por transição. A cada transição está associada uma repetitividade que terá de ser satisfeita para que a transposição da transição ocorra, para que o programa evolua para a etapa seguinte (FRANCHI e CAMARGO, 2008).

De forma prática, podemos analisar o funcionamento de uma edição da simulação produzida em SFC observando a Figura 5. Os retângulos representam os passos a serem executados no controle do processo e, entre alguns retângulos, a condição necessária para que se chegue ao novo passo. Sendo assim, podemos garantir que determinado passo nunca ocorra sem que uma transição esteja concluída.

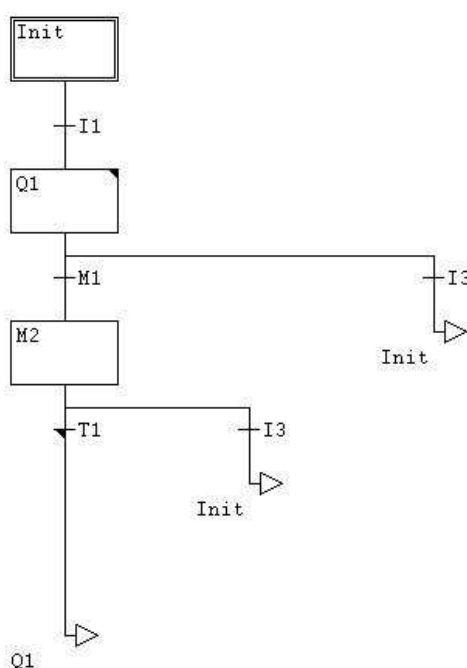


Figura 5: Programação desenvolvida em SFC

3.4 LISTA DE INSTRUÇÕES

A Lista de Instruções é baseada na linguagem assembly e de característica essencialmente sequencial, caracterizando-se por instruções que possuem um operador e, dependendo do tipo de operação, podem incluir um ou mais operandos, separados por vírgulas (FRANCHI, 2008).

De certo modo, podemos dizer que a linguagem é a transcrição do diagrama de relés (LD), ou seja, a passagem de uma linguagem gráfica para uma linguagem escrita. A Tabela 2 apresenta as instruções mais comuns empregadas nessa linguagem.

A linguagem utiliza uma única instrução por linha de programação, como pode ser observado na Figura 6 que representa a edição da simulação em Lista de Instruções. Faz-se

necessário um controle muito eficiente na utilização dos registradores e interfaces de entrada e saída para não haver falhas durante a programação.

| Operador | Modificador | Operando | Descrição/ significado |
|----------|-------------|----------------|---|
| LD | N | | Carrega o operando para o acumulador |
| ST | N | | Armazena o conteúdo do acumulador no local especificado pelo operando |
| S | | BOOL | Faz com que o valor do operando seja 1 |
| R | | BOOL | Faz com que o valor do operando seja 0 |
| AND | N, (| | Função booleana AND |
| & | N, (| | Função booleana AND |
| OR | N, (| | Função booleana OR |
| XOR | N, (| | Função booleana OU - Exclusivo |
| ADD | | | Soma |
| SUB | | | Subtração |
| MUL | | | Multiplicação |
| DIV | | | Divisão |
| GT | (| | Comparação (Greater Than) maior que (>) |
| GE | (| | Comparação (Greater or Equal) maior ou igual que (>=) |
| EQ | (| | Comparação (Equal to) igual a (=) |
| NE | (| | Comparação (Not Equal) diferente de (< >) |
| LT | (| | Comparação (Less Than) menor que (<) |
| LE | (| | Comparação (Less or Equal) menor ou igual que (<=) |
| JMP | C, N | Nome do rótulo | Desvia para o rótulo Nome_do_Rótulo |
| CAL | C, N | Nome da Função | Invoca a execução de um bloco de funções |
| RET | C, N | | Retorna de uma função ou bloco de função. |

Tabela 2: Lista de comandos da linguagem IL

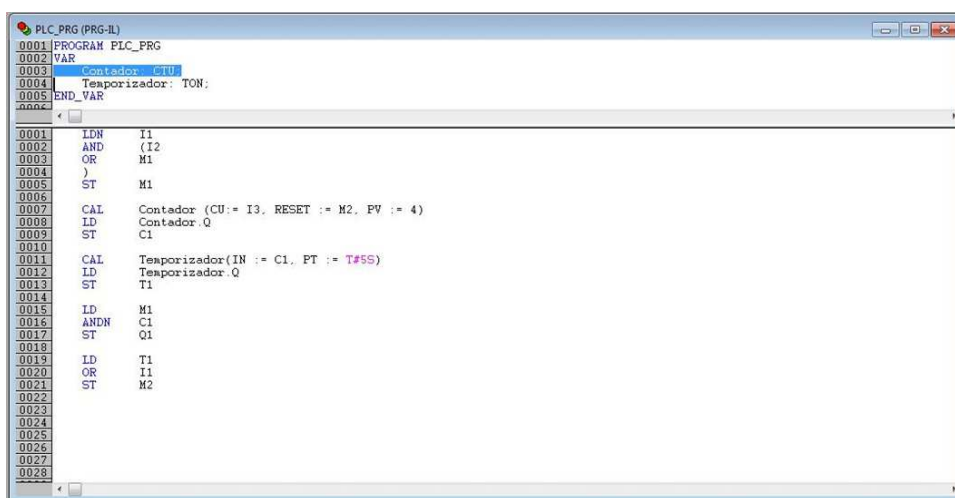


Figura 6: Programação desenvolvida em IL



3.5 TEXTO ESTRUTURADO

A linguagem de programação Estruturada, ou Structured Text (ST), é uma linguagem textual muito próxima das linguagens de alto nível como PASCAL, ou linguagem C. Podemos escrever expressões matemáticas e lógicas em uma linha, ou utilizar comandos de decisão, laço, contagem etc. Contém os elementos essenciais de uma linguagem de programação moderna, incluindo as instruções condicionais (IF, THEN, ELSE e CASE OF) e instruções de iterações (FOR, WHILE e REPEAT). Permite a programação de funções e blocos que ficariam de difícil solução com outras linguagens.

A linguagem possibilita a utilização de mais de uma instrução por linha, agiliza e facilita a tarefa dos programadores em projetos mais complexos. Esta linguagem é muito utilizada por quem vem da área de Informática devido à sua similaridade com as linguagens de programação de alto-nível. A tela de programação para a linguagem ST é demonstrada na figura 7 abaixo.

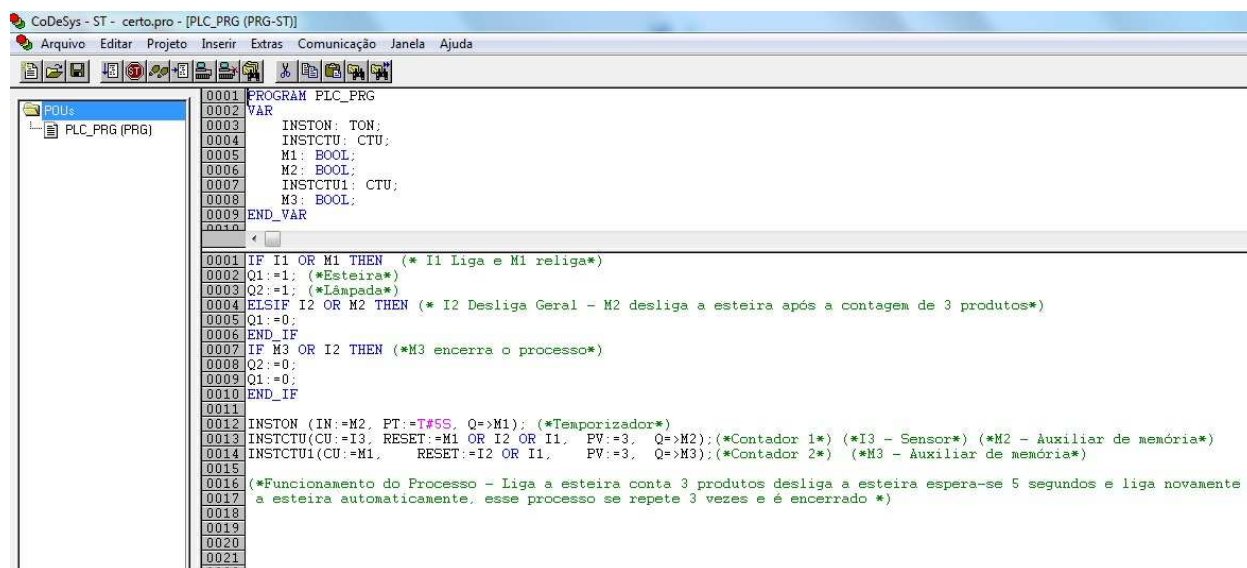


Figura 7: Programação desenvolvida em ST

4. CONSIDERAÇÕES FINAIS

A proposta desse trabalho era elaborar uma apostila didática, sendo que para realização da tarefa foi necessário dominar as cinco linguagens de programação estabelecidas pela norma IEC 61131-6. O software Codesys, em seu uso, demonstrou ser uma ferramenta versátil para a implementação de linguagens de programação para CLP's, obedecendo ao padrão global estabelecido pela norma, permitindo que pessoas com diferentes habilidades e formações, criem programas em estágios diferentes para automatização de algum processo.

O material didático produzido foi elaborado implementando um processo em cada linguagem de programação, sendo possível identificar que as linguagens gráficas são utilizadas



em processos de menor complexidade, enquanto que as textuais proporcionam uma maior flexibilidade para o programador. Para cada situação existe uma linguagem, porém o programador é que vai decidir em qual programar.

5. REFERÊNCIAS BIBLIOGRÁFICAS

FRANCHI, Claiton Moro; CAMARGO, Valter Luís Arlindo. **Controladores Lógicos Programáveis: Sistemas Discretos**. 1. ed. São Paulo: Érica, 2008.

SILVEIRA, Paulo Rogério da; SANTOS, Winderson E. dos Santos. **Automação e Controle Discreto**. 1. ed. São Paulo: Érica, 1998.

PAREDE, Ismael Moura, Luiz Carlos da Cunha. **Eletrônica: automação industrial**. São Paulo: Fundação Padre Anchieta, 2011.

INTERNATIONAL ELECTROTECHNICAL COMMISSION. **IEC 61131-3: Programmable controllers: Programming languages**. Genebra, 2013.

NATIONAL ELECTRICAL MANUFACTURERS ASSOCIATION. **NEMA IA 2.1-2005: Programmable Controllers Part 1: General Information**. Rosslyn, 2005.

GOEKING, Weruska. **Da máquina a vapor aos softwares de automação**. Disponível em: <<http://www.osetoreletrico.com.br/web/a-revista/343-xxxx.html>>. Acesso em: 11/05/2015.