



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

CAIO AUGUSTO MOREIRA

**LINGUAGEM LADDER APLICADA A PROFISSIONAIS DA CIÊNCIA
DA COMPUTAÇÃO**

Assis/SP
2020

CAIO AUGUSTO MOREIRA

**LINGUAGEM LADDER APLICADA A PROFISSIONAIS DA CIÊNCIA
DA COMPUTAÇÃO**

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Bacharelado em Ciência da Computação.

Orientador: Prof. Esp. Célio Desiró

Assis/SP
2020

FICHA CATALOGRÁFICA

Moreira, Caio Augusto.

Linguagem Ladder aplicada para profissionais da ciência da computação / Caio Augusto Moreira.
Fundação Educacional do Município de Assis –FEMA – Assis, ano. 2020 . 69 p

1. CLP, **2.** Ladder, **3.** Ciência da Computação, **4.** Automação Industrial.

CDD:
Biblioteca da FEMA

LINGUAGEM LADDER APLICADA A PROFISSIONAIS DA CIÊNCIA DA COMPUTAÇÃO

CAIO AUGUSTO MOREIRA

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Bacharelado em Ciência da Computação, avaliado pela seguinte comissão examinadora:

Orientador: Prof. Esp. Célio Desiró

Examinador: Prof. MSc. Guilherme de Cleve Farto

Assis/SP
2020

DEDICATÓRIA

Dedico este trabalho a Deus que vem me abençoando a cada dia

AGRADECIMENTOS

Agradeço em primeiro lugar a **DEUS** que sempre tem me abençoado e capacitado.

Agradeço meu pai **Edson Renato Moreira** e minha mãe **Rita de Cássia Tragueta Moreira**, que sempre acreditaram em mim e trabalharam muito para que meus sonhos se concretizem.

Agradeço aos meus irmãos **Edson Renato Moreira Filho** e **João Moreira Neto**, que sempre estiveram ao meu lado.

Agradeço a minha namorada **Rafaela Marocolo Felicetti**, por sempre estar ao meu lado e fazer meus dias mais felizes.

Agradeço ao meu orientador **Prof. Esp. Célio Desiró**, por me orientar e colaborar nesta fase acadêmica tão importante.

Agradeço ao professor examinador **Prof. MSc. Guilherme de Cleve Farto**, por avaliar meu trabalho e dar sugestões construtivas.

RESUMO

Com a evolução do tempo vem se concretizando a quarta onda de avanço tecnológico aplicado à indústria, conhecida como Indústria 4.0, que se inicia com a consolidação das ferramentas da tecnologia da informação, uso de simulações, uso da computação em nuvem e com o aprimoramento dos sensores e com conectividade além dos CLPs. Para demonstrar que a linguagem *Ladder* pode ser utilizada por profissionais e acadêmicos de Ciência da Computação, foi demonstrado neste trabalho em duas etapas, a primeira é comparar as disciplinas da grade do curso de Ciência da Computação com os conceitos de *Ladder*, a segunda etapa é utilizar os conceitos de *Ladder* para desenvolver um programa de controle de um tanque de nível. Após todos os resultados apresentados, é perfeitamente cabível afirmar que os acadêmicos e profissionais da área de tecnologia da informação poderão desenvolver programas para CLPs em *Ladder* e assumir o protagonismo da indústria 4.0.

Palavras-chave: CLP, Ladder, Automação Industrial.

ABSTRACT

With the evolution of time, the fourth wave of technological advancement applied to industry has come to fruition, known as Industry 4.0, which begins with the consolidation of information technology tools, the use of simulations, the use of cloud computing and the improvement of sensors and connectivity beyond PLCs. To demonstrate that the Ladder language can be used by Computer Science professionals and academics, it was demonstrated in this work in two steps, the first is to compare the disciplines of the Computer Science course with the concepts of Ladder, the second step is use Ladder concepts to develop a level tank control program. After all the results presented, it is perfectly reasonable to state that academics and professionals in the information technology area will be able to develop programs for PLCs in Ladder and assume the role of industry 4.0.

Keywords: PLC, Ladder, Industrial Automation.

LISTA DE FIGURAS

Figura 1: Arquitetura do CLP	22
Figura 2: CLP com Interface IHM	24
Figura 3: Fluxograma do funcionamento da CPU do CLP	25
Figura 4: Estrutura da programação em <i>Ladder</i>	27
Figura 5 Aplicação do temporizador TON.....	28
Figura 6: Aplicação do Contador Crescente/Decrescente	29
Figura 7: Tela Inicial <i>CODESYS</i>	30
Figura 8: Ambiente de programação do <i>CODESYS</i>	31
Figura 9: Ambiente 3D <i>FACTORY I/O</i>	32
Figura 10: Tela de aplicação Elipse SCADA	33
Figura 11: Tanque de nível que será controlado.....	35
Figura 12 - Porta Lógica AND com o diagrama Ladder correspondente	38
Figura 13 - Porta Lógica OR com o diagrama Ladder correspondente	38
Figura 14 - Porta Lógica NOT com o diagrama Ladder correspondente	38
Figura 15 - Porta Lógica NAND com o diagrama Ladder correspondente	39
Figura 16 - Porta Lógica NOR com o diagrama Ladder correspondente.....	39
Figura 17 - Porta Lógica XOR com o diagrama Ladder correspondente.....	39
Figura 18 - Porta Lógica SET RESET com o diagrama Ladder correspondente.....	40
Figura 19 - Declaração de Variáveis	41
Figura 20 - Início do sistema	42
Figura 21 - Conversão Real para Int	42
Figura 22 - Bloco Move.....	43
Figura 23 - Bloco PID.....	43
Figura 24 - Acionamento Válvula de Descarga	44
Figura 25 - Fechamento da Válvula de Descarga.....	44
Figura 26 - Menu Sistema Supervisório.....	45
Figura 27 - Tela de Processo.....	46
Figura 28 - Tela do Gráfico de Nível	46
Figura 29 - Tela de Alarmes.....	47
Figura 30 - Tanque de Nível com Setpoint em 45	48
Figura 31 - Tela de Processo com setpoint em 45	49
Figura 32 - Alarme de Nível Baixo.....	49
Figura 33 - Gráfico de nível com setpoint em 45	50
Figura 34 - Variação do Nível do Tanque	50
Figura 35 - Variação da Tela de Processo	51
Figura 36 - Variação Gráfico de Nível	51
Figura 37 - Avisos de Alarmes de Nível Alto	52

LISTA DE TABELAS

Tabela 1: Relação de comandos <i>ladder</i> com comandos elétricos	26
Tabela 2: Linguagem <i>Ladder</i> com Portas Lógicas.....	27

LISTA DE ABREVIATURAS E SIGLAS

A/D: Analógico para Digital

CLP: Controlador Lógico Programável

CPU: Unidade Central de Processamento

D/A: Digital para Analógico

IHM: Interação Homem Máquina

I/O: Input e Output(Entrada e Saída)

V: Volt

Sumário

1 INTRODUÇÃO.....	14
1.2 OBJETIVOS	15
1.2.1 Objetivos Gerais	15
1.2.2 Objetivos Específicos.....	16
1.3 PÚBLICO ALVO.....	16
1.4 JUSTIFICATIVAS	17
1.5 MOTIVAÇÃO.....	17
1.6 PERSPECTIVA DE CONTRIBUIÇÃO	18
1.7 METODOLOGIA	18
1.7.1 Procedimento	18
1.7.2 Software.....	19
1.8 ESTRUTURA DO TRABALHO	19
2. CONTROLADOR LÓGICO PROGRAMAVEL (CLP).....	21
2.1 HISTÓRICO.....	21
2.2 ARQUITETURA DO CLP	22
2.3 FUNCIONAMENTO DO CLP.....	25
3. LINGUAGEM LADDER	26
3.1 CONCEITOS TEÓRICOS DA LINGUAGEM <i>LADDER</i>	26
3.2 FUNCIONALIDADES DA LINGUAGEM <i>LADDER</i>	28
3.2.1 Temporizadores	28
3.2.2 Contador.....	29
4. SOFTWARES UTILIZADOS	30
4.1 CODESYS	30
4.2 FACTORY I/O	32
4.3 ELIPSE SCADA	32
5. PROPOSTA DO TRABALHO	34
5.1 TECNOLOGIA E RECURSOS ADOTADOS.....	34
6. ANÁLISE DE DISCIPLINAS COM SIMILARIDADE A LINGUAGEM LADDER DA GRADE CURRICULAR DO CURSO DE CIÊNCIA DA COMPUTAÇÃO	36
6.1 ANÁLISES DAS DISCIPLINAS	36

6.1.1 Algoritmos e Estruturas de Dados I	36
6.1.2 Introdução a Lógica	37
6.1.3 Eletrônica Digital.....	37
6.2 COMPARAÇÕES DOS CONCEITOS DE PORTAS LÓGICAS COM OS DIAGRAMAS EM <i>LADDER</i>	37
6.3 CONCLUSÕES SOBRE AS SEMELHANÇAS DE CONCEITOS ENTRE O <i>LADDER</i> E PORTAS LÓGICAS	40
7. DESENVOLVIMENTO DA PROGRAMAÇÃO DO TANQUE DE NÍVEL.....	41
7.1 PROGRAMAÇÃO <i>LADDER</i> NO <i>CODESYS</i>	41
7.2 DESENVOLVIMENTO DO SISTEMA SUPERVISÓRIO	45
8. RESULTADOS	48
8.1 PARÂMETRO DE SETPOINT EM 45	48
8.2 PARÂMETRO DE SETPOINT VARIANDO DE 45 A 250	50
9 CONCLUSÃO	53
9.1 TRABALHOS FUTUROS	53
10 REFERÊNCIAS	54
APÊNDICE A – Instalação Codesys	58
APÊNDICE B – Instalação Factory I/O	64
APÊNDICE C – Instalação Elipse SCADA	67
APÊNDICE D – Programa Básico em Ladder “Hello Word”.....	69

1 INTRODUÇÃO

A sociedade tem passado por avanços tecnológicos que impulsionaram a produtividade industrial desde o início da Primeira Revolução Industrial, na qual foram aplicadas as fábricas com motores a vapor. Na etapa seguinte a eletrificação levou à produção em larga escala e, finalmente na terceira revolução, acontece a automatização da produção com o uso da tecnologia de informação.

Desde 2011 vem se concretizando a quarta onda de avanço tecnológico aplicado à indústria, conhecida como Indústria 4.0, que se inicia com a consolidação das ferramentas da tecnologia da informação, uso de simulações integrando um modelo virtual da planta com a planta propriamente dita, uso da computação em nuvem e com o aprimoramento dos sensores e com conectividade além dos CLPs (HEIDRICH et al, 2017).

Os Controladores Lógicos Programáveis (CLPs) são equipamentos eletrônicos utilizados em sistemas de automação e controle industrial. São ferramentas úteis em sistema de controle, sendo utilizados em alta escala no mercado industrial. Uma característica que torna os CLPs flexíveis é a possibilidade que eles oferecem de se desenvolver e alterar facilmente a lógica para acionamento das saídas em função das entradas (PIRES, 2002).

O CLP necessita para seu funcionamento a presença de um programa que define a lógica do controle usado no experimento. Esta lógica possui vários tipos de linguagens possíveis para sua programação (PIRES, 2002).

Uma destas linguagens, conhecida como *Ladder*, é baseada na simbologia gráfica e seus arranjos. Esta linguagem representa o fluxo da corrente elétrica. Possui símbolos que representam contatos ON/OFF de entradas que representam dados do mundo real e símbolos que representam o comportamento de uma saída quanto à mudança das entradas. Outros símbolos representam um macro, ou seja, um conjunto de operações (ALMEIDA, 2003).

Há um grande interesse, por parte dos alunos da computação em aprender sobre programação de sistemas controlados, mas a grande barreira está na complexidade

da programação envolvida nos mesmos, muito diferente das linguagens de alto nível (SAMPAIO, 2011).

Outra motivação para este interesse, é que os CLPs e os microprocessadores têm muitas características em comum com um *personal computer* (PC), como por exemplo, portas seriais, portas paralelas (ALMEIDA, 2003).

Este trabalho irá realizar a programação de um CLP utilizando a linguagem *Ladder*, com objetivo de demonstrar a integração da linguagem para os profissionais e acadêmicos de Ciência da Computação. .

1.2 OBJETIVOS

1.2.1 Objetivos Gerais

O presente projeto tem como objetivo geral demonstrar que a programação de CLPs utilizando a linguagem *Ladder* pode ser utilizada por profissionais e acadêmicos de Ciência da Computação.

Estes resultados serão atingindo em duas etapas, a primeira é comparar as disciplinas da grade do curso de Ciência da Computação com os conceitos de *Ladder*, a segunda etapa é utilizar os conceitos de *Ladder* para desenvolver um programa para controlar o nível de um tanque.

Com essas duas etapas concluídas, pode-se demonstrar que os profissionais de ciência da computação podem desenvolver programas em *Ladder* e utilizar esses conhecimentos para programar em CLPs.

1.2.2 Objetivos Específicos

Pretende-se com este trabalho demonstrar que a programação de CLPs utilizando a linguagem *Ladder* pode ser utilizado por cientistas da computação. Para se tornar possível a realização deste projeto foram definidos os seguintes objetivos específicos:

- Realizar o levantamento da parte teórica do CLP,
- Realizar o levantamento da parte teórica da linguagem *Ladder*,
- Demonstrar a relação da linguagem *Ladder* com algumas disciplinas constantes na grade do curso de Ciência da Computação,
- Desenvolver um programa em *Ladder* utilizando o software *CODESYS*,
- Desenvolver um ambiente gráfico da cena de um tanque de nível utilizando o software *Factory IO* para simular o comportamento do CLP,
- Desenvolver um sistema supervisorio utilizando o software *Elipse SCADA*,
- Fazer a conexão entre a programação e o CLP simulado,
- Fazer a integração entre o sistema conectado do CLP com o supervisorio desenvolvido,
- Documentar os resultados obtidos.

1.3 PÚBLICO ALVO

Este trabalho tem como público alvo acadêmico e profissional de Ciência da Computação, trazendo a realidade da linguagem *Ladder* para o cotidiano. Com isto será possível desenvolver programas para CLPs, com a possibilidade de estes profissionais integrarem novas oportunidades de trabalho no setor industrial participando maciçamente da 4ª revolução industrial.

1.4 JUSTIFICATIVAS

Atualmente estamos diante da Quarta Revolução Industrial, intitulada de “Indústria 4.0”, que se caracteriza pela incorporação de tecnologias emergentes ao ambiente de trabalho, proporcionando ganhos substanciais de produtividade, transformando a natureza do trabalho e gerando impactos nas esferas políticas, econômicas e sociais, com a implementação das novas tecnologias e do conceito de Indústria 4.0.

Devido a isso os empregos passarão por grandes mudanças. Tais mudanças serão responsáveis pela extinção de tradicionais postos de trabalho, mas também pela criação de novas oportunidades (ABRAMOVAY, 2017).

Com essas novas oportunidades surgindo no setor industrial, passou a ser muito importante a integração do profissional de Ciência da Computação com a linguagem *Ladder*. Assim o profissional poderá utilizar seu conhecimento computacional e desenvolver programas para CLPs que é o cérebro da parte produtiva automatizada de uma indústria, e atuar em um novo ramo de atuação completamente inserido no setor produtivo de uma indústria.

1.5 MOTIVAÇÃO

A motivação para realização deste trabalho vem no desenvolvimento do profissional cientista da computação que, além do conhecimento que carrega nas mais diversas tecnologias de programação do mundo, conseguirá atuar no ramo de automação industrial programando em CLPs, trazendo uma integração entre a engenharia elétrica e a tecnologia da informação.

1.6 PERSPECTIVA DE CONTRIBUIÇÃO

A perspectiva de contribuição com este trabalho é primeiramente despertar o interesse dos profissionais e acadêmicos de Ciência da Computação na tecnologia *Ladder* e assim ingressar no mercado de automação. Com isso, abrem-se novos horizontes para a atuação dos profissionais formados no curso de Ciência da Computação.

1.7 METODOLOGIA

1.7.1 Procedimento

A proposta e objetivos deste trabalho acadêmico serão alcançados por meio de pesquisas teóricas e desenvolvimento de um programa demonstrativo em *Ladder*, este programa será conectado no *software Factory IO* que tem a função de simular um CLP e diversas cenas para o seu uso, logo após também será utilizado o *software Elipse Scada* que tem a função de ser o sistema supervisor da cena que será controlada pelo CLP.

A cena escolhida será de um tanque de nível, que tem as válvulas de encher e de descarga que serão controladas através do CLP, quando o fluido do tanque estiver fora do valor determinado às válvulas entraram em ação automaticamente para deixar o tanque com o nível que foi ajustado anteriormente.

1.7.2 Software

- **CodeSys**– Compilador *Ladder*.
Software não gratuito, mas apresenta versão demonstrativa para download com limitação de recursos, sem expiração de tempo.
Disponível em: <https://www.codesys.com>
- **Factory IO** – Simulador de CLP e também de cenas para sua utilização.
Software não gratuito, mas apresenta versão demonstrativa para download com todos os recursos presentes na versão paga, com duração de 30 dias.
Disponível em: <https://factoryio.com/>
- **Eclipse SCADA** – Simulador para criação de sistema supervisório.
Software não gratuito, mas apresenta versão demonstrativa para download com limitação de recursos, sem expiração de tempo.
Disponível em: <https://www.elipse.com.br/>

1.8 ESTRUTURA DO TRABALHO

A estrutura deste trabalho será composta das seguintes partes:

- **Capítulo 1 – Introdução:** Neste capítulo é contextualizada a área de estudo e apresentarão os objetivos, público alvo, justificativas, motivação, perspectivas de contribuição e metodologia de pesquisa para o desenvolvimento deste trabalho.
- **Capítulo 2 – Controlador Lógico Programável (CLP):** Neste capítulo, introduz-se sobre o que é e como funciona um CLP.
- **Capítulo 3 – Linguagem *Ladder*:** Neste capítulo, introduz-se sobre o que é e como se estrutura a linguagem *Ladder*.
- **Capítulo 4 – Softwares Utilizados:** Neste capítulo, é apresenta e explica sobre os softwares que serão utilizados no trabalho.

- **Capítulo 5 – Proposta de Trabalho:** Neste capítulo, apresenta sobre a importância da linguagem *Ladder* para o cientista da computação e apresenta como será feito o trabalho.
- **Capítulo 6 – Análise de Disciplinas com Similaridade a Linguagem *Ladder* Referente à Grade Curricular do Curso de Ciência da Computação:** Neste capítulo, apresenta uma análise da grade do curso de ciência de computação com o intuito de realizar uma comparação das semelhanças dos conceitos da linguagem *Ladder* com os conteúdos estudados.
- **Capítulo 7 – Desenvolvimento da Programação do Tanque de Nível:** Neste capítulo, apresenta todo o desenvolvimento da programação realizada em *Ladder* para controle do tanque de nível.
- **Capítulo 8 – Resultados:** Neste capítulo, apresentam os resultados obtidos da programação do tanque de nível e do sistema supervisorio.
- **Capítulo 9 – Conclusão:** Neste capítulo, apresentam-se as vantagens do uso da linguagem *Ladder* para profissionais e acadêmicos de ciência da computação e sugestões para trabalhos futuros.
- **Referências.**

2. CONTROLADOR LÓGICO PROGRAMAVEL (CLP)

CLP é um computador especializado para desempenhar a função de automação, controle, monitoramento de máquinas e de processos industriais de vários tipos e das mais diversas complexidades.

2.1 HISTÓRICO

Na década de 60, o aumento da competitividade fez com que a indústria automotiva melhorasse o desempenho de suas linhas de produção, aumentando tanto a qualidade como a produtividade. Fazia-se necessário encontrar uma alternativa para os sistemas de controle a relés. Uma saída possível, imaginada pela General Motors, seria um sistema baseado no computador (GEORGINI, 2002).

Assim, em 1968, a Divisão Hydramatic da General Motors determinou os critérios para o projeto do CLP, sendo que o primeiro dispositivo a atender às especificações foi desenvolvido pela Goulds Modicon em 1969 (GEORGINI, 2002).

Com a criação do CLP's devido a necessidade de mercado, estes equipamentos começaram a mostrar uma versatilidade e praticidade muito maior que os painéis elétricos antigos, apesar de, no início, serem rústicos na sua estrutura de programação (ANTONELLI, 1998).

A cada dia foi se realizando mudanças significativas e aprimoramentos, como por exemplo, a variedade de tipos de entrada e saídas, aumento da velocidade de processamento, a inclusão de blocos lógicos complexos para tratamento das informações e principalmente o modo de programação e a interface com o usuário (ANTONELLI, 1998).

Os CLPs podem-se dividir de acordo com o sistema de programação utilizada (CASILLO, 2011):

1ª Geração: Programação em *Assembly*. Faz-se necessário conhecer o hardware do equipamento, ou seja, a eletrônica do projeto do CLP.

2ª Geração: Surgem as linguagens de programação de nível médio. É desenvolvido o “Programa monitor”, que transforma para linguagem de máquina o programa inserido pelo usuário.

3ª Geração: Os CLPs passam a ter uma entrada de programação que era feita através de um teclado, ou programador portátil, conectado ao mesmo.

4ª Geração: É introduzida uma entrada para comunicação serial, e a programação passa a ser feita através de microcomputadores. Com esse advento, surge a possibilidade de testar o programa antes de esse ser transferido ao módulo do CLP.

5ª Geração: Os CLPs de quinta geração vêm com padrões de protocolo de comunicação para facilitar a interface com equipamentos de outros fabricantes, e, também, com Sistemas Supervisórios e Redes Internas de comunicação.

2.2 ARQUITETURA DO CLP

Segundo Nogueira (2010) o CLP é projetado seguindo um padrão de arquitetura, composto basicamente de: uma CPU, memória, dispositivos de entrada e saída, e uma IHM (interface homem máquina), como apresentado pela a figura e detalhado logo embaixo.

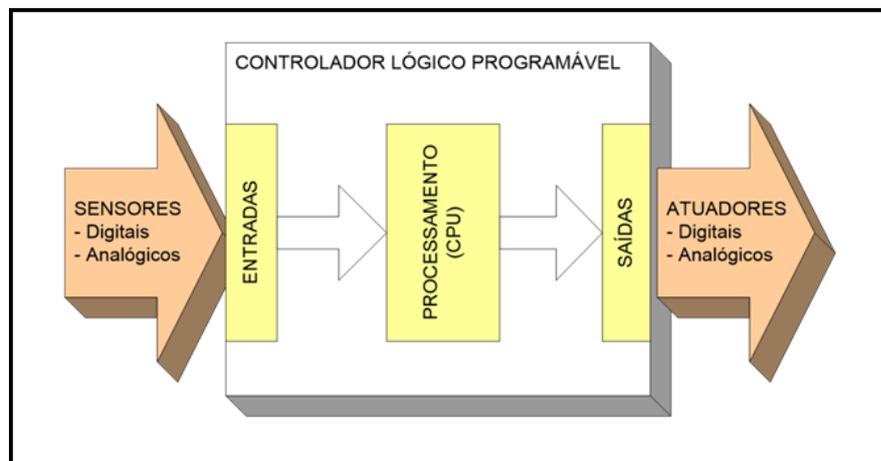


Figura 1: Arquitetura do CLP

Fonte: <https://alfacombrasil.com/2019/02/11/clp-o-que-e-e-como-funciona/>

Unidade de Entrada- Esse dispositivo fornece as conexões entre os dispositivos de campo e a unidade central de processamento (CPU). Podem ter um ou mais canais de aquisição de dados que codifica o sinal analógico ou digital de diversos níveis de tensão, provenientes de botoeiras, chaves, sensores, termostatos, pressostatos, termopares, encoders, tensões, correntes.

Entradas Digitais: São entradas que recebem sinais que assumem apenas 2 níveis, 0 e 1, 0v ou 5v, 0v ou 24v, 0v ou 220v. Estes sinais podem vir de chaves fim de curso, botões de painéis elétricos, sensores do tipo ON/OFF, etc.

Entradas Analógicas: São entradas que recebem sinais que podem assumir vários valores dentro de uma faixa determinada de tensão ou controle. Estes sinais podem vir de sensores de temperatura, velocidade, nível, e que sejam proporcionais, ou seja, enviam um sinal que varia de 0v a 10v, por exemplo, para informar a temperatura exata do processo naquele instante.

Unidade de Saída – Da mesma forma que a unidade de entrada, a unidade de saída fornece as conexões entre os dispositivos de campo e CPU. Esse módulo irá comutar as tensões de controle fornecidas, necessárias para acionar vários dispositivos, como conectoras, solenóides, atuadores dentre outros.

Saídas Digitais: São saídas que enviam sinais que podem assumir apenas 2 níveis de tensão, 0v ou 24v, por exemplo, e podem ser utilizados para acionar um motor, uma bomba, etc.

Saídas Analógicas: São saídas que enviam sinais que podem assumir vários níveis de tensão dentro de uma determinada faixa, por exemplo, 0v a 10v.

Unidade Central de Processamento (CPU): É o centro nervoso do sistema, responsável pelo gerenciamento e processamento das informações, é composto de microprocessador ou microcontrolador. Ela recebe os sinais digitais e os sinais analógicos dos sensores do campo conectado aos módulos de entrada e também recebe os comandos e o dado via comunicação de rede. Em seguida executa as operações lógicas, as operações aritméticas e avançadas como as de controle de malha programada na memória do usuário e atualiza os cartões de saída digital e analógica.

Memória: Podemos dividir em três partes: memória básica, memória de dados, memória de usuário.

Memória básica: Contém um conjunto de programas armazenados permanentemente, com o objetivo de controlar e supervisionar as atividades do sistema.

Memória de dados: Também conhecida como memória rascunho, podendo ser volátil ou não, a cada ciclo de varredura a memória de dados é atualizada, nela são armazenado todos os dados de controle do sistema.

Memória de usuário: É a memória destinada ao armazenamento das instruções de programação, ou seja, o programa de usuário (ele é o responsável para controlar a máquina ou a operação do processo).

Interface Homem-Máquina (IHM): são utilizados principalmente para a introdução e visualização de dados e mensagens. Permite a interação do homem com a máquina.

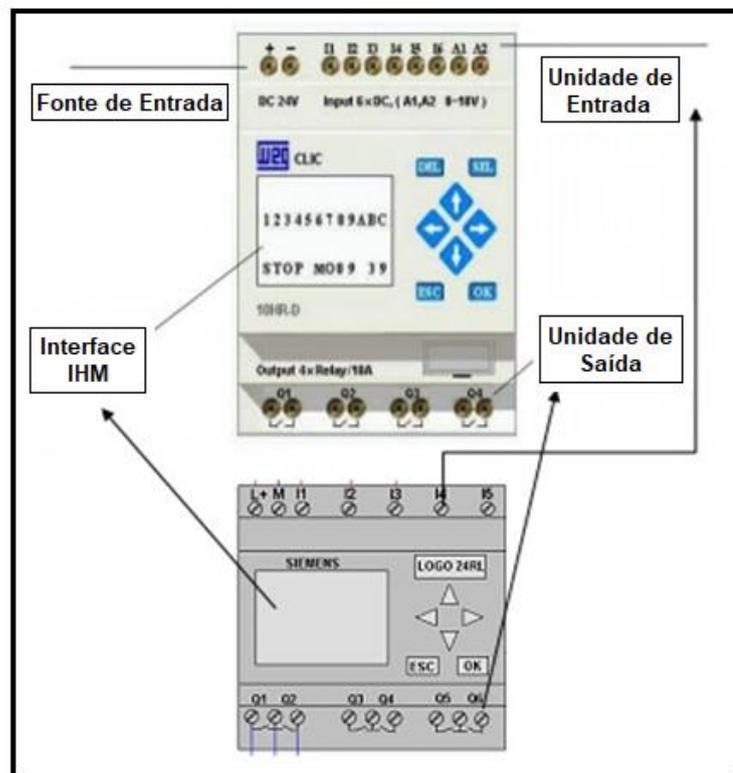


Figura 2: CLP com Interface IHM

Fonte: <http://engmecatonico.blogspot.com/2011/05/arquitetura-basica-do-clp.html>

2.3 FUNCIONAMENTO DO CLP

O CLP contém sinais de entrada e saídas que podem ser digitais ou analógicos. Os módulos de entrada e saídas são compostos de grupos de bits, associados em conjunto de 8 bits (1 byte) ou conjunto de 16 bits, de acordo com o tipo da CPU. As entradas analógicas são módulos conversores A/D, que convertem um sinal de entrada em um valor digital, normalmente de 12 bits (4096 combinações). As saídas analógicas são módulos conversores D/A, ou seja, um valor binário é transformado em um sinal analógico (STOCLER, 2005).

Os sinais dos sensores são aplicados às entradas do controlador e a cada ciclo (varredura) todos esses sinais são lidos e transferidos para a unidade de memória interna denominada memória imagem de entrada. Estes sinais são associados entre si e aos sinais internos. Ao término do ciclo de varredura, os resultados são transferidos à memória imagem de saída e então aplicados aos terminais de saída (SEVERO, 2006). Este ciclo está representado na figura 3.

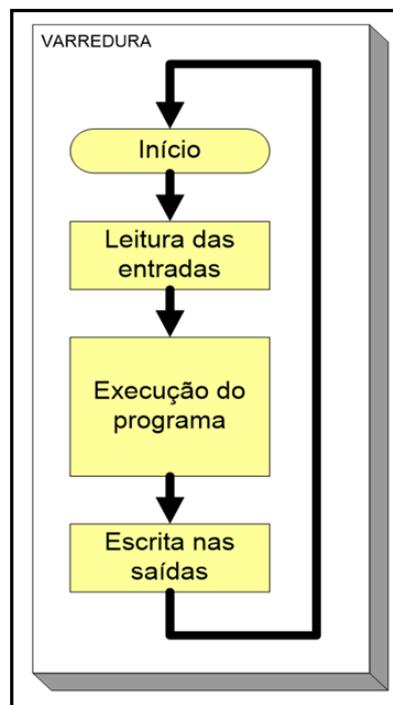


Figura 3: Fluxograma do funcionamento da CPU do CLP

Fonte: <https://alfacombrasil.com/2019/02/11/clp-o-que-e-e-como-funciona/>

3. LINGUAGEM LADDER

Este capítulo irá apresentar os conceitos teóricos da linguagem *Ladder* de como ela surgiu, as suas funcionalidades e alguns exemplos de blocos.

3.1 CONCEITOS TEÓRICOS DA LINGUAGEM LADDER

A linguagem *Ladder* foi a primeira que surgiu na programação dos Controladores Lógica Programável (CLPs), pois sua funcionalidade procurava imitar os antigos diagramas elétricos, utilizados pelos técnicos e engenheiros e pessoas habituadas com diagramas e esquemas elétricos, mas, devido a sua simplicidade, acabou sendo difundida entre profissionais das mais diversas áreas (NOGUEIRA, 2010).

Tipo	Símbolo	Equipamento elétrico
Contato aberto		
Contato fechado		
Saída		

Tabela 1: Relação de comandos *ladder* com comandos elétricos
Fonte: Nogueira 2010

A linguagem *Ladder* consiste na lógica matemática binária que possui apenas dois valores que são representados por: 0 e 1. A partir desses dois símbolos se constrói então uma base numérica binária. A partir desses conceitos foram criadas as portas lógicas, que são circuitos utilizados para combinar níveis lógicos digitais de formas específicas (SILVA, 2007).

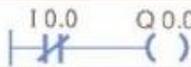
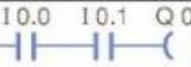
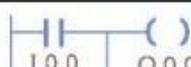
Portas Lógicas	Símbolo	Expressão	Ladder
NOT	A —  — S	$S = \bar{A}$	
AND	A —  — S	$S = A \cdot B$	
OR	A —  — S	$S = A + B$	

Tabela 2: Linguagem *Ladder* com Portas Lógicas

Fonte: <http://controleeautomacaoindustrial3.blogspot.com/2013/08/aula-23-logicas-booleanas.html>

O nome *Ladder* deve-se à representação da linguagem se parecer com uma escada (do inglês, *ladder*), na qual duas barras verticais paralelas são interligadas pela Lógica de Controle, formando os degraus (do inglês, *rungs*) da escada. Portanto, a cada lógica de controle existente no programa de aplicação dá-se o nome de *rung*, a qual é composta por colunas e linhas (CARVALHO, 2011). Os degraus são divididos em duas zonas conforme ilustra a figura 4. A zona da esquerda pode ser chamada de zona de testes e é a região que contém a lógica da função booleana. A zona da direita é chamada de zona de ação e representa a saída da função booleana.

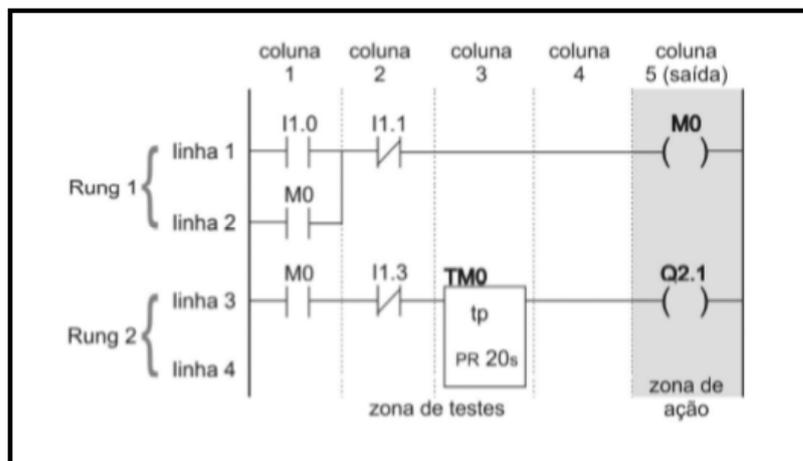


Figura 4: Estrutura da programação em *Ladder*

Fonte: Nogueira 2010

Fazendo uma avaliação genérica e de forma elétrica sobre a linguagem *Ladder*, seu funcionamento consiste em criar uma lógica de acordo com os requisitos do problema, de modo que uma corrente fictícia chegue à zona de ação.

3.2 FUNCIONALIDADES DA LINGUAGEM *LADDER*

A linguagem *Ladder* contém outras funcionalidades além dos blocos de contato aberto e fechado, sendo assim possível desenvolver uma lógica de programação mais completa e eficiente para a solução de problemas propostos.

3.2.1 Temporizadores

Segundo (SILVA, 2007), um dos elementos mais utilizados na lógica de programação *Ladder* são os temporizadores. Existem quatro tipos de temporizadores:

Temporizador de pulso (TP): aciona após um único pulso, após esse pulso ele aciona, e desativa após o tempo desejado.

Temporizador de atraso no desligamento (TOF): é usado para contar um determinado tempo, e após esse tempo, desliga o sinal.

Temporizador com atraso no acionamento (TON): é usado para contar um determinado tempo, e após esse tempo, libera o sinal.

Temporizador com atraso no acionamento retentivo (TONR): é usado para acumular o tempo sempre que o dispositivo for energizado e mantém o tempo corrente quando a energia é desligada do dispositivo.



Figura 5 Aplicação do temporizador TON

Fonte: PIRES 2002

3.2.2 Contador

Segundo (MAITELLI, 2002), um contador é um componente simples aplicado para contar pulsos. São utilizados para indexar, incrementar ou decrementar valores, existem quatro tipos de contadores que são:

CTUD – Contador Crescente e Decrescente: Tem a capacidade de no mesmo bloco realizar operação crescente e decrescente,

CTU – Contador Crescente: Realiza uma contagem ascendente até o valor máximo definido;

CTD – Contador Decrescente: Realiza uma contagem descendente a partir do valor acumulado (ACC) até o valor mínimo definido;

RES (Reset): Faz o zeramento da contagem.

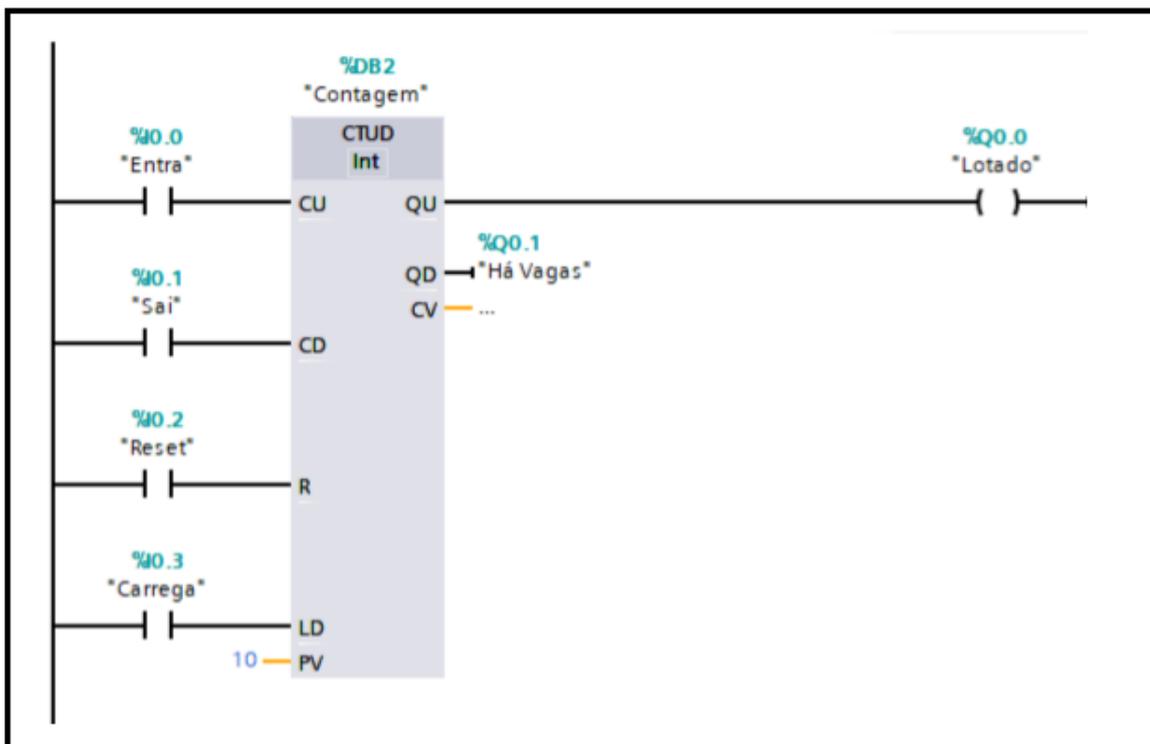


Figura 6: Aplicação do Contador Crescente/Decrescente
Fonte: PIRES 2002

4. SOFTWARES UTILIZADOS

Para desenvolver o trabalho é necessário a utilização de alguns softwares que são: *CODESYS* que será o compilador para a implementação da linguagem *Ladder*, o *FACTORY I/O* que será o ambiente 3D para simulação dos equipamentos e também será utilizado o *ELIPSE SCADA* que tem a função de ser o sistema supervisor de controle do processo.

4.1 CODESYS

Segundo (CODESYS, 2020), o *codesys* é desenvolvido e comercializado pela *3S-Smart Software Solutions GmbH*, uma empresa alemã situada na cidade de Kempten, na Baviera. A versão 1.0 foi lançada em 1994 e atualmente encontra-se na versão 3.5. Sua interface de desenvolvimento é gratuita e não é orientado a um tipo específico de hardware de controlador programável ou sistema embarcado, inclusive não dependendo do fabricante do hardware. É amplamente utilizado na indústria para diversos tipos de automações, variando de lógicas simples à robótica, controle de movimento.

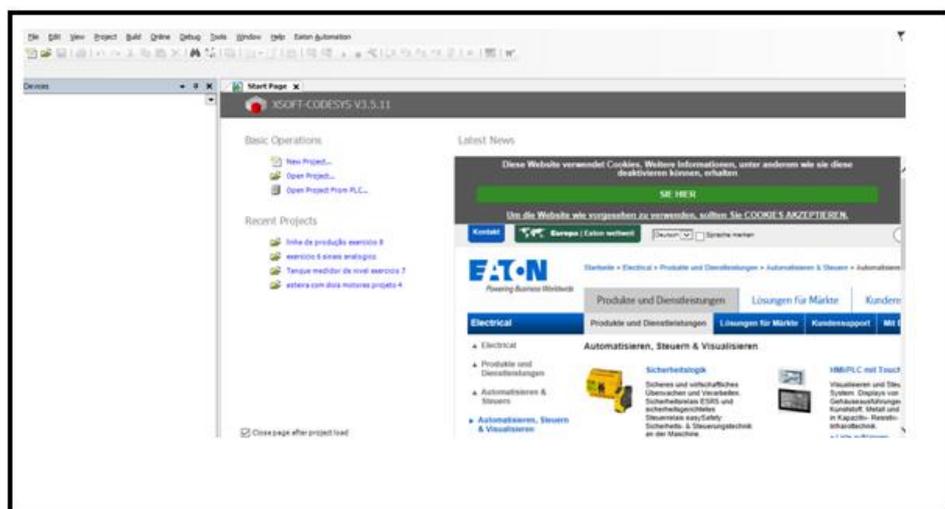


Figura 7: Tela Inicial CODESYS

Todas as cinco linguagens de desenvolvimento de CLPs estão disponíveis na interface de desenvolvimento do *codesys* e podem ser utilizadas em uma mesma aplicação. São elas:

- **ST (Structured Text) Texto estruturado;**
- **IL (Instruction List) Lista de instruções;**
- **SFC (Sequential Flow Chart) Diagrama de fluxo;**
- **LD (Ladder) Linguagem ladder;**
- **FBD (Function Block Diagram) Diagrama de bloco.**

O seu funcionamento é feito através do compilador interno que monta o código da máquina (código binário) a partir da aplicação desenvolvida e transfere ao dispositivo de destino, a partir do qual é possível testar o sistema utilizando os avançados recursos de depuração (*debug*), como a monitoração de variáveis, *breakpoints*, osciloscópio e *data-logger*. Na ausência de um dispositivo também é possível executar a simulação da aplicação.

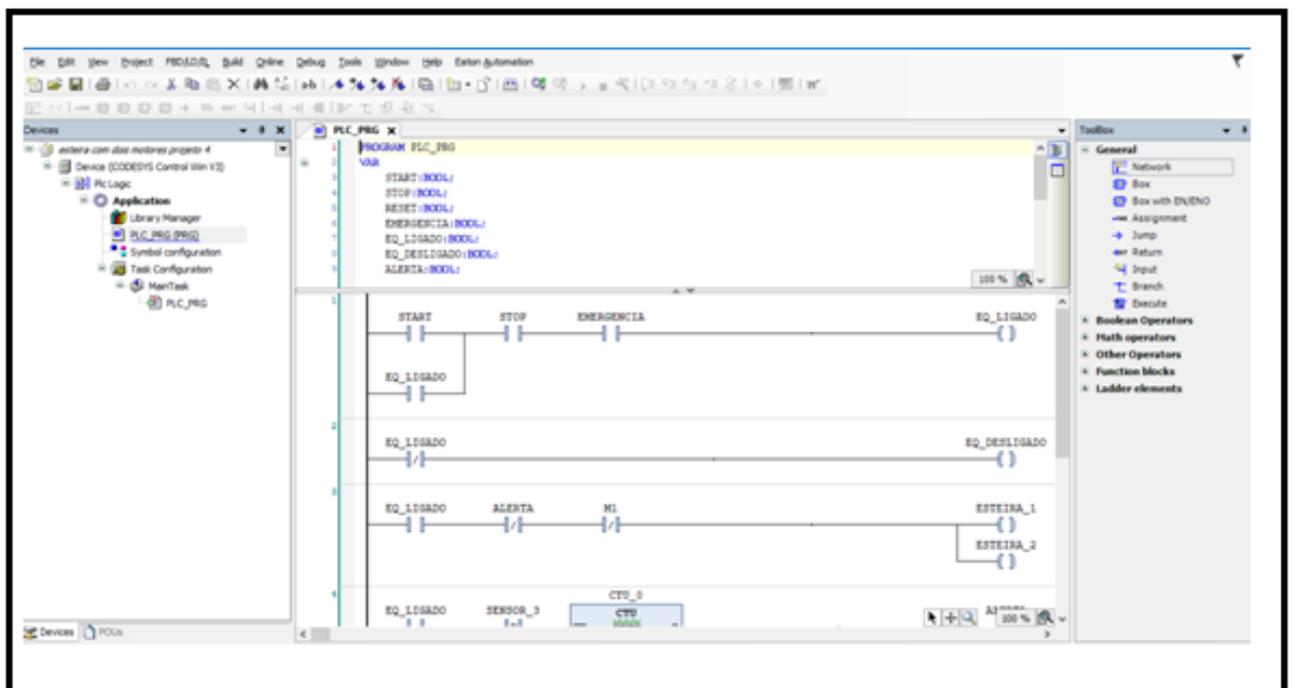


Figura 8: Ambiente de programação do CODESYS

4.2 FACTORY I/O

Factory I/O é uma simulação de fábrica 3D para aprender tecnologias de automação. Projetado para ser fácil de usar, permite construir rapidamente uma fábrica virtual usando uma seleção de peças industriais comuns. O cenário mais comum é usar o *Factory I/O* como uma plataforma de treinamento de CLP, uma vez que o CLP são os controladores mais comuns encontrados em aplicações industriais (FACTORY I/O, 2020).

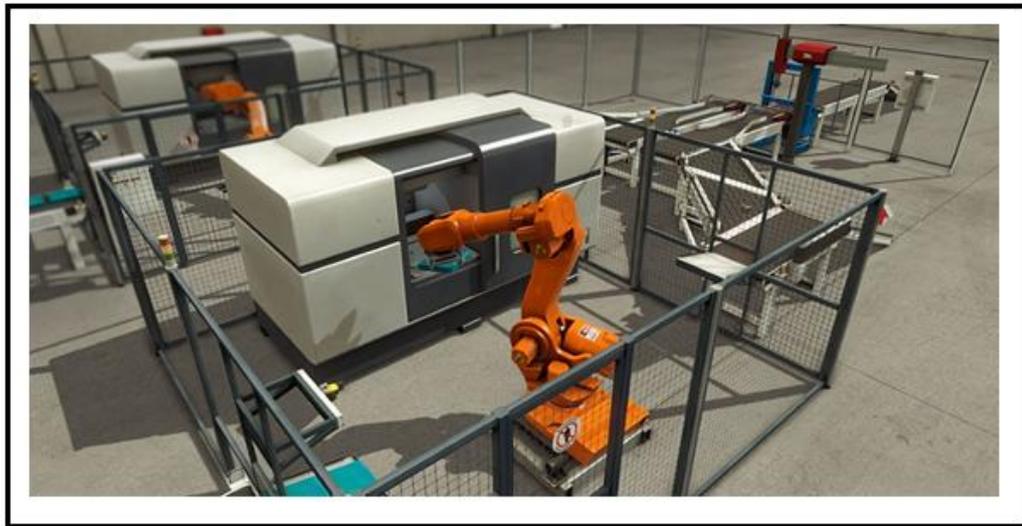


Figura 9: Ambiente 3D *FACTORY I/O*

4.3 ELIPSE SCADA

O *ELIPSE SCADA* foi desenvolvida pela empresa Elipse software, a sigla *SCADA* é uma expressão em inglês para *Supervisory Control and Data Acquisition*, ou seja, controle supervisorio e aquisição de dados. A aquisição de dados representa a obtenção de informações e seu armazenamento, com esses dados pode se realizar o monitoramento e análise de variáveis envolvidas em um controle de processo.

Esses sistemas servem como uma interface entre operadores e processos em diversas situações como máquinas industriais, controle de processos (ELIPSE, 2020).

A partir dos sistemas SCADA pode se idealizar tanto projetos simples como automação e controle de um tanque de abastecimento quanto os painéis de controle das grandes geradoras e distribuidoras de energia elétrica e das grandes plataformas de exploração de petróleo. Os elementos básicos de um SCADA são os *drivers* de comunicação com os equipamentos, uma interface homem máquina (IHM), e um registro contínuo dos dados. As IHM's em geral disponibilizam elementos gráficos típicos das aplicações industriais como botões, ícones, motores para facilitar a visualização do que está sendo monitorado e ou controlado no processo para os operadores (ELIPSE 2020).

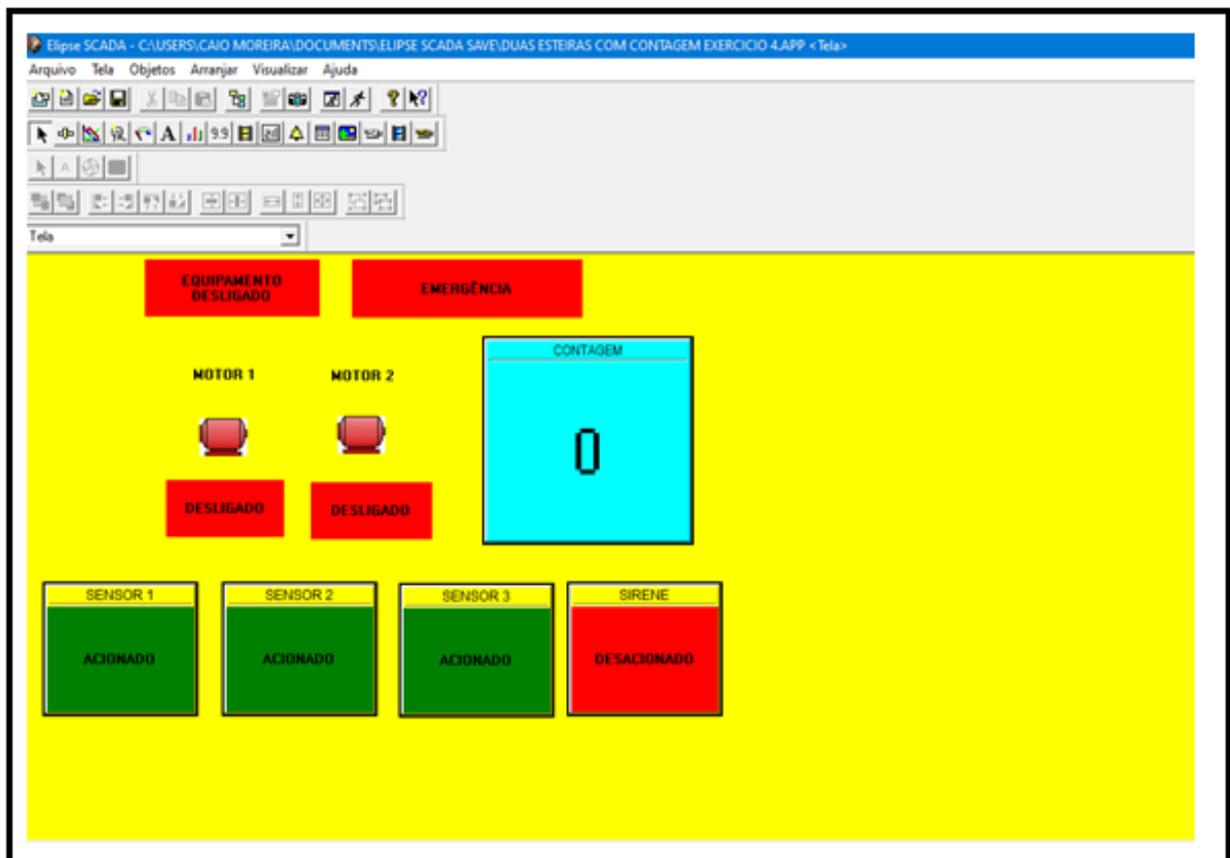


Figura 10: Tela de aplicação Elipse SCADA

5. PROPOSTA DO TRABALHO

Este trabalho tem como objetivo demonstrar que a programação de CLPs utilizando a linguagem *Ladder* pode ser utilizada por profissionais e acadêmicos de ciência da computação, com isso abrir o mercado de automação industrial para os profissionais de ciência da computação.

Para chegar nesse objetivo este trabalho será dividido em duas etapas, a primeira etapa será uma análise da grade curricular do curso de Ciência da Computação com o intuito que estas disciplinas podem ser utilizadas como base para o estudo da linguagem *Ladder* pelos acadêmicos e profissionais da área.

Segunda etapa deste trabalho irá desenvolver um programa em *Ladder* para realizar o controle de um tanque de nível, fazendo que este tanque controle seu nível automaticamente com a quantidade de fluido pré-estipulada e todo este processo sendo monitorado por um sistema supervisório.

5.1 TECNOLOGIA E RECURSOS ADOTADOS

Em um primeiro momento será desenvolvido um programa para controlar o nível de um tanque com válvulas de encher e esvaziar automáticas, este programa será conectado em uma cena de tanque de nível no ambiente virtual do software *Factory I/O*, o *Factory I/O* tem a função de simular o CLP que fará o controle do sistema e também de todos os equipamentos elétrico-mecânicos da cena.

Logo após esta etapa será desenvolvido um sistema supervisório de controle em tempo real do tanque de nível, para isso será necessário à utilização do software *Elipse SCADA*.



Figura 11: Tanque de nível que será controlado

6. ANÁLISE DE DISCIPLINAS COM SIMILARIDADE A LINGUAGEM LADDER DA GRADE CURRICULAR DO CURSO DE CIÊNCIA DA COMPUTAÇÃO

Este capítulo tem o intuito de analisar a grade currículo do Curso de Ciência da Computação e encontrar matérias com similaridade com a linguagem *Ladder* conforme o desenvolvimento teórico da linguagem *Ladder* apresentado no capítulo 3 deste trabalho.

6.1 ANÁLISES DAS DISCIPLINAS

Conforme apresentado no capítulo 3 neste trabalho, a linguagem *Ladder* É baseado em funções lógicas, uma combinação de contatos abertos e fechados muito similar com o princípio de funcionamento das portas lógicas.

Analisando a grade do curso de Ciência da Computação encontra-se a disciplina de Algoritmo e Estruturas de Dados I, Introdução a Lógica e Eletrônica Digital.

6.1.1 Algoritmos e Estruturas de Dados I

Analisando o conteúdo programático e os objetivos da disciplina de Algoritmos e Estruturas de Dados I, nota-se que o aluno aprenderá os conceitos de programação estruturada e, entre estes conceitos estão os relacionados à definição de variáveis com os tipos apropriados para armazenamento dos valores e o uso correto dos operadores e conectivos lógicos, utilizados nos comandos condicionais e iterativos (FEMA, 2020).

Todos estes conceitos serão utilizados para iniciar a programação em *Ladder*, pois uma escolha mal feita das variáveis acarretará em uma falha total na funcionalidade do programa.

6.1.2 Introdução a Lógica

Analisando o conteúdo programático e os objetivos da disciplina de Introdução a Lógica, nota-se que o aluno aprenderá sobre os conceitos de conectivos lógicos e terá uma introdução sobre as funções booleanas e de portas lógicas (FEMA, 2020).

Toda esta base de conhecimento aplicado na linguagem *Ladder* vai possibilitar que o aluno comece a criar combinações de lógicas para resolver diferentes tipos de problemas.

6.1.3 Eletrônica Digital

Analisando o conteúdo programático da disciplina de Eletrônica Digital, o aluno irá se dedicar mais intensivamente sobre as aplicações dos estudos digitais para as resoluções de problemas, para isso o aluno irá estudar mais profundamente as funções booleanas e os conceitos de portas lógicas (FEMA, 2020).

Após a conclusão desta disciplina o aluno terá plenas condições em se aprofundar no desenvolvimento de programas em *Ladder*, pois o conceito de portas lógicas e de funções booleanas e a base da linguagem *Ladder*.

6.2 COMPARAÇÕES DOS CONCEITOS DE PORTAS LÓGICAS COM OS DIAGRAMAS EM *LADDER*

E quando se fala em lógica, logo vêm à mente funções lógicas como “E” ou “AND” e “OU” ou “OR”, muito conhecidas na eletrônica digital. Esta mesma lógica, com algumas mudanças nos símbolos, também pode ser usada na estruturação de programas a serem desenvolvidos em *ladder*.

- Lógica AND - Executa função lógica “AND”, ou seja, somente se as entradas A e B estiverem em 1 e a saída S será acionada.



Figura 12 - Porta Lógica AND com o diagrama Ladder correspondente

- Lógica OR - Executa função lógica “OR”, ou seja, para que a saída S seja acionada basta que uma das entradas A ou B esteja em 1.



Figura 13 - Porta Lógica OR com o diagrama Ladder correspondente

- Lógica NOT - Executa função lógica “NOT”, ou seja, nega ou inverte o sinal de entrada.

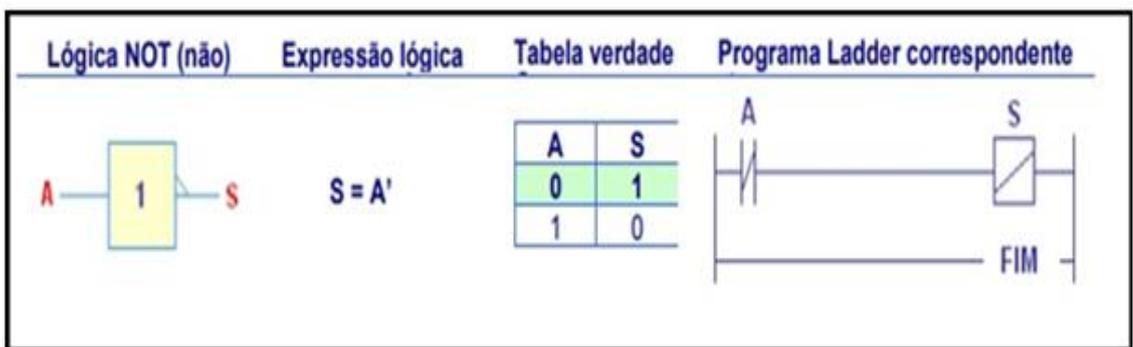


Figura 14 - Porta Lógica NOT com o diagrama Ladder correspondente

- Lógica NAND - Executa função lógica “NAND”, ou seja, nega ou inverte as saídas da função AND.



Figura 15 - Porta Lógica NAND com o diagrama Ladder correspondente

- Lógica NOR - Executa função lógica “NOR”, ou seja, nega a função OR, invertendo assim, suas saídas.



Figura 16 - Porta Lógica NOR com o diagrama Ladder correspondente

- Lógica XNOR – A saída recebe sinal 1 quando a entrada A ou a entrada B estiveram com nível de sinal em 1.



Figura 17 - Porta Lógica XOR com o diagrama Ladder correspondente

- Lógica SET e RESET - Set” significa Ligar e “Reset” desligar. Seu funcionamento é simples uma vez setado (nível lógico (1) em A) ele comuta a saída S, ou seja, vai para (1) e somente volta para nível baixo (0) se for resetado.



Figura 18 - Porta Lógica SET RESET com o diagrama Ladder correspondente

6.3 CONCLUSÕES SOBRE AS SEMELHANÇAS DE CONCEITOS ENTRE O LADDER E PORTAS LÓGICAS

Após analisar a ementa da disciplina de eletrônica digital do curso de ciência da computação pode se comprovar que os docentes tem uma grande capacidade de compreensão das portas lógicas.

Com a grande semelhança do conceito de portas lógicas e linguagem *Ladder*, se torna plausível que os estudantes e profissionais de ciência da computação consigam desenvolver programas e sistemas em *Ladder*.

Para comprovar essa afirmação no próximo capítulo será desenvolvido um sistema em *Ladder* de controle do fluxo do nível de um tanque.

7. DESENVOLVIMENTO DA PROGRAMAÇÃO DO TANQUE DE NÍVEL

Neste capítulo será apresentado o desenvolvimento da programação *Ladder* de um tanque de nível incluindo todo o sistema de supervisão do processo e o acompanhamento virtual da cena.

7.1 PROGRAMAÇÃO LADDER NO CODESYS

O primeiro processo para o desenvolvimento do algoritmo em *Ladder* de controle do tanque de nível foi realizar a definição das variáveis conforme a figura abaixo.

```

1 PROGRAM PLC_PRG
2 VAR
3 MEDIDOR_NIVEL:REAL;
4 MEDIDOR_FLUXO:REAL;
5 VALVULA_DESCARGA:REAL;
6 VALVULA_ENCHER:REAL;
7 SETPOINT:REAL;
8 SP:INT;
9 FV:INT;
10 START:BOOL;
11 STOP:BOOL;
12 RESET:BOOL;
13 START_LIGHT:BOOL;
14 STOP_LIGHT:BOOL;
15 RESET_LIGHT:BOOL;
16 SP_REAL:REAL;
17 SP_INT: INT;
18 FV_REAL: REAL;
19 FV_INT: INT;
20 PID_0: PID :=(KP :=1, TN := 1.5, TV:=0, CLOCK := (ET := TIME#250MS));
21 M1: BOOL;
22
23 M2: BOOL;
24 END_VAR

```

Figura 19 - Declaração de Variáveis

As variáveis foram definidas em diversos formatos como Real, Inteiro, Booleano e PID (Proporcional-Integral-Derivativo).

- Real – Tem como objetivo representar valores que contém casa decimal.
- Inteiro – Tem como objetivo representar valores de números inteiros.
- Booleano – Tem como objetivo representar uma condição verdadeira ou falsa.
- PID – Representar o bloco de controle do processo.

Logo após definir as variáveis, foi implementada a lógica de acionamento do sistema conforme a figura 20 abaixo.

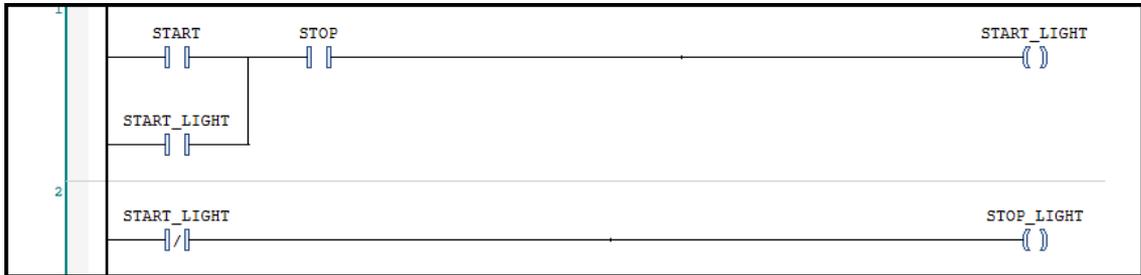


Figura 20 - Início do sistema

Com o sistema das válvulas de encher ativo conforme a figura acima é necessário a realização do controle do fluxo que contém no tanque.

Para a realização esse procedimento tem que se levar em conta que a cena 3D do tanque utiliza um sensor de nível do tipo analógico com valores de tensão variando de 0 a 10 V e a régua de medição do tanque vai de 0 a 300 centímetros.

A variável MEDIDOR_NIVEL recebe o valor lido pelo sensor fazendo se necessário uma conversão numérica para uma escala de 0 a 300 centímetros.

Para essa conversão foi utilizado um bloco multiplicador e logo após um bloco de conversão de real para inteiro, pois esse valor de nível será utilizado no bloco de PID e o bloco de PID só recebe valores do tipo inteiro.

Este mesmo procedimento vai ser realizado para a variável SETPOINT.

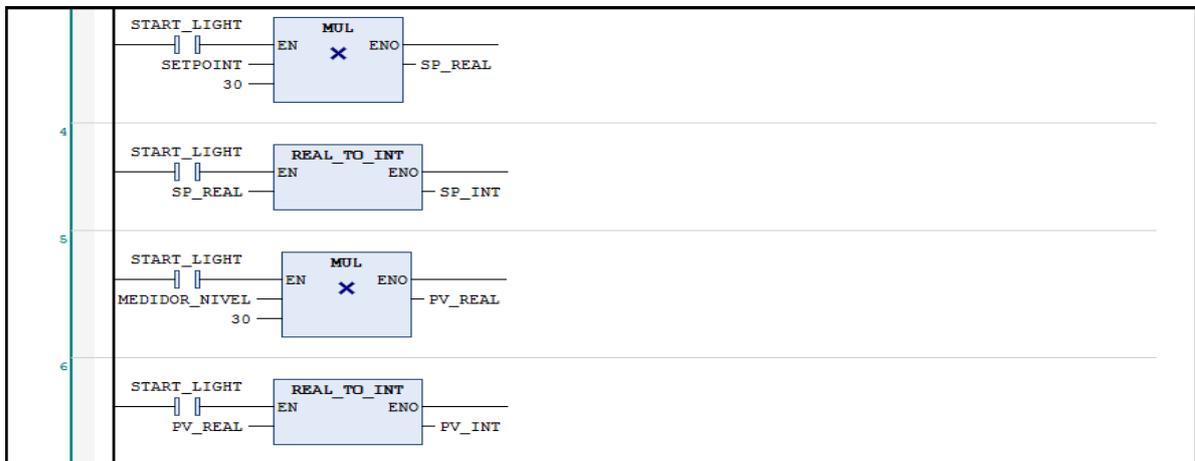


Figura 21 - Conversão Real para Int

Para os valores das variáveis MEDIDOR_NIVEL e SETPOINT aparecerem no visor digital da cena do tanque no software *FACTORY I/O* e necessário à utilização do bloco *move*, que tem a função de mover a informação de uma variável e mandar para outra conforme a figura abaixo.



Figura 22 - Bloco Move

O bloco PID (Proporcional-Integral-Derivativo) será usado para calcular o valor de atuação sobre o processo a partir das informações dos valores das variáveis PV_INT e SP_INT em sua entrada, essas informações serão responsáveis pelo acionamento da variável VALVULA_ENCHER na saída do bloco.

A velocidade de acionamento da válvula de encher e definida pelo componente Y_MIN e Y_MAX que atua de 0 a 10, sendo da seguinte forma o valor 0 representa válvula fechada e 10 representa a válvula 100 % aberta.

O componente RESET representa a paralisação do bloco.

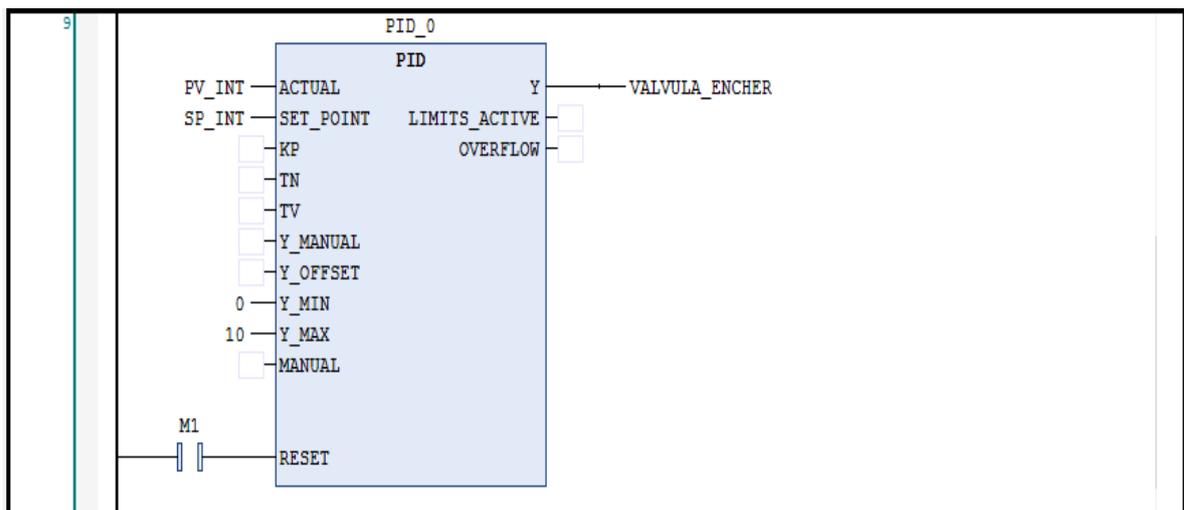


Figura 23 - Bloco PID

Após realizar o controle de enchimento do tanque com o bloco PID (Proporcional-Integral-Derivativo), agora será necessário trabalhar com a variável VALVULA_DESCARGA que tem como objetivo de esvaziar o fluido do tanque para que o controle do nível seja mais preciso e nunca ocorra enchimento acima do normal.

Para o desenvolvimento dessa lógica foi necessário à criação de uma variável auxiliar M1 e de um bloco de comparação, esse bloco de comparação irá verificar se o nível de fluido existente é maior que o valor do setpoint determinado, caso essa condição for verdadeira a variável M1 aciona o reset do bloco PID (Proporcional-Integral-Derivativo) fechando a válvula de encher.

Será necessária a utilização de mais um bloco *move* para regular a porcentagem de abertura da válvula de descarga para que o fluido excessivo escoe.



Figura 24 - Acionamento Válvula de Descarga

Agora para não permitir que escoe fluido excessivo será necessário a criação de mais uma variável auxiliar M2 e mais um bloco de comparação, o bloco de comparação irá comparar se o nível de fluido do tanque e menor ou igual ao valor de setpoint, se essa condição for atendida será atribuída o valor 0 para fechar a valvular de descarga.

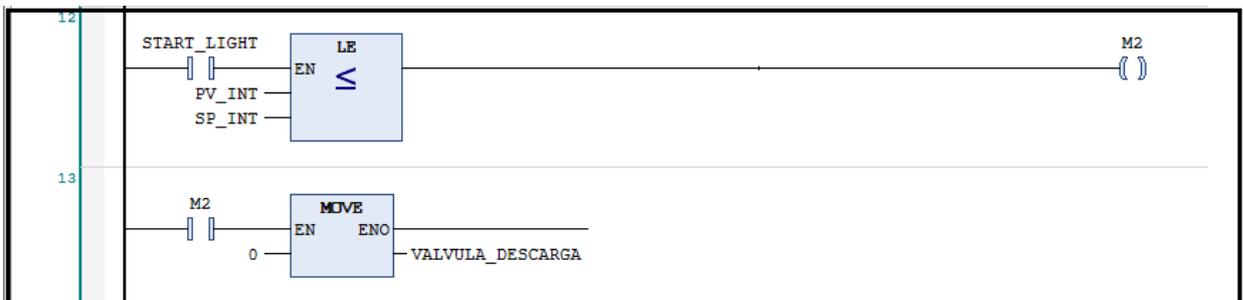


Figura 25 - Fechamento da Válvula de Descarga

7.2 DESENVOLVIMENTO DO SISTEMA SUPERVISÓRIO

A princípio foi desenvolvido uma tela inicial contendo as informações do trabalho e o menu de navegação para as telas de monitoramento.



Figura 26 - Menu Sistema Supervisório

A tela de Processo é responsável pelas informações básica do processo, que contém:

- Botões de navegação para as outras telas.
- Monitoramento de sistema ligado ou desligado.
- Medidor de fluxo que representa o enchimento do tanque.
- Medidor do nível desejado de SETPOINT.
- Medição do nível que representa a medição em tempo real de fluido no tanque.
- Medidor da válvula de descarga que mostra a taxa de fluido que está sendo descarregado por segundo.

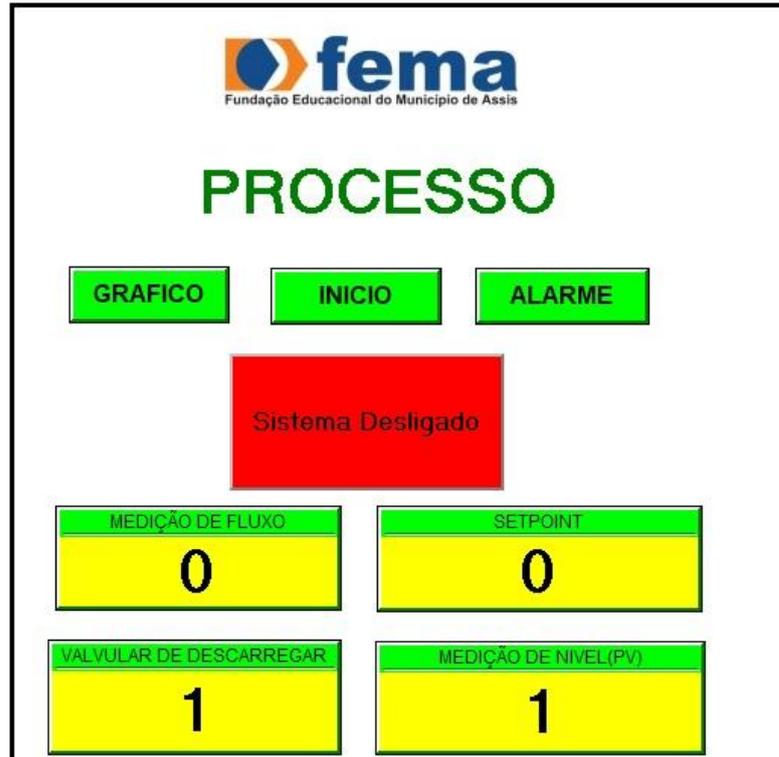


Figura 27 - Tela de Processo

A tela de Gráfico é responsável por traçar um gráfico entre o nível de fluido atual no tanque pelo nível do valor ajustado de setpoint.

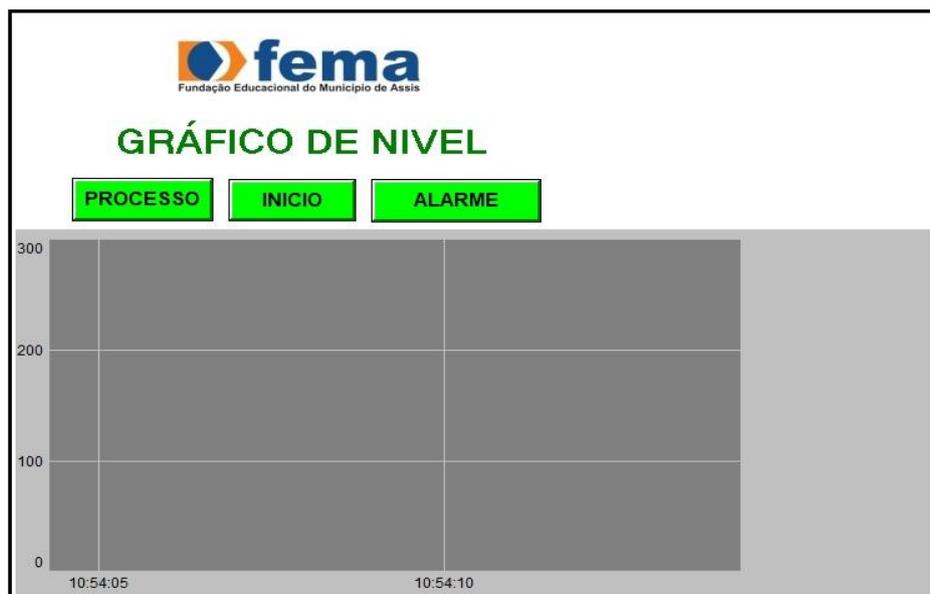


Figura 28 - Tela do Gráfico de Nível

A tela de Alarmes é responsável por informar e alertar sobre os níveis de fluido do tanque avisando se está muito baixo ou muito alto.

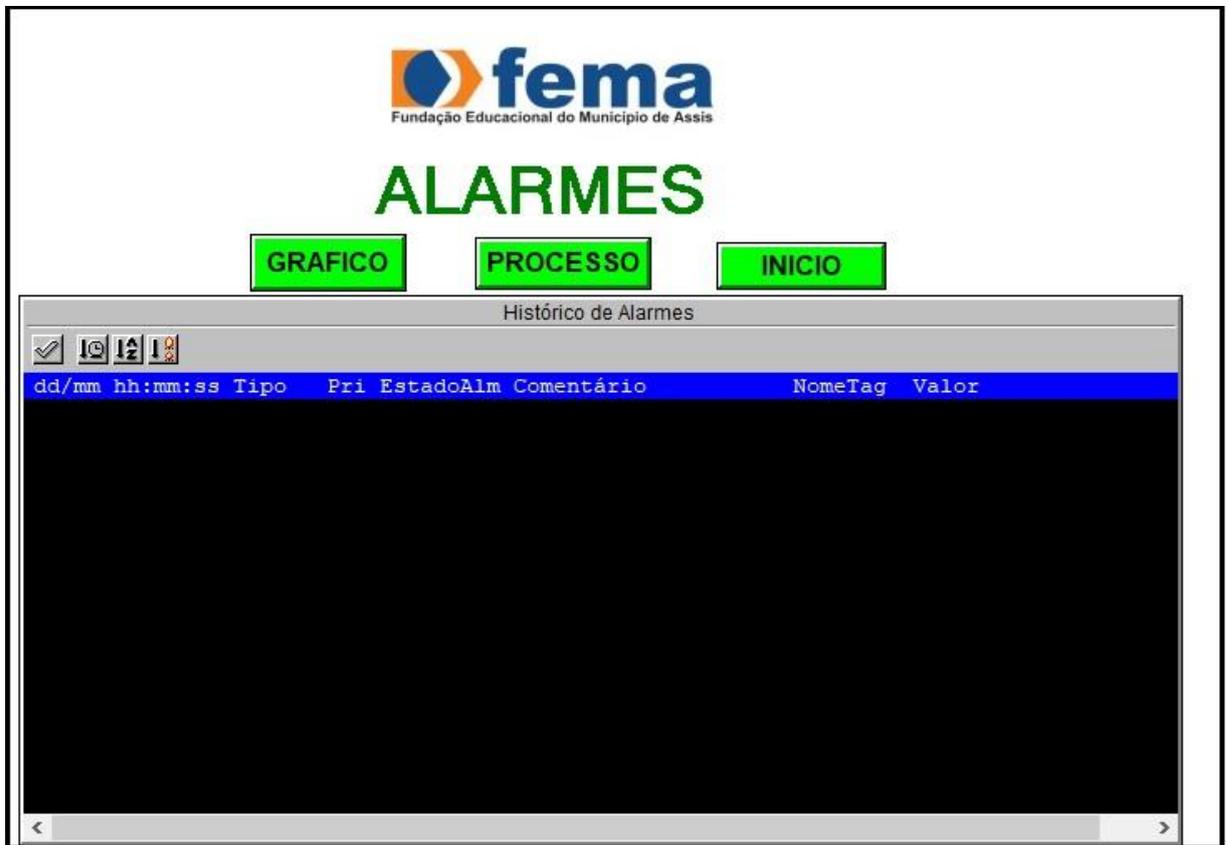


Figura 29 - Tela de Alarmes

8. RESULTADOS

Neste capítulo será apresentado os resultados óbitos pelo monitoramento do tanque de nível com diferentes parâmetros de SETPOINT.

8.1 PARÂMETRO DE SETPOINT EM 45

O primeiro parâmetro está com o SETPOINT ajustado em 45, à imagem da figura xx abaixo mostra o painel de controle do tanque com o ajuste do SETPOINT em 45 e o nível de fluido sendo regulando automaticamente para o nível próximo a 45.



Figura 30 - Tanque de Nível com Setpoint em 45

O sistema de monitoramento supervisorio registra na sua tela de controle os dados de nível, SETPOINT, medição de fluxo e sistema ligado.

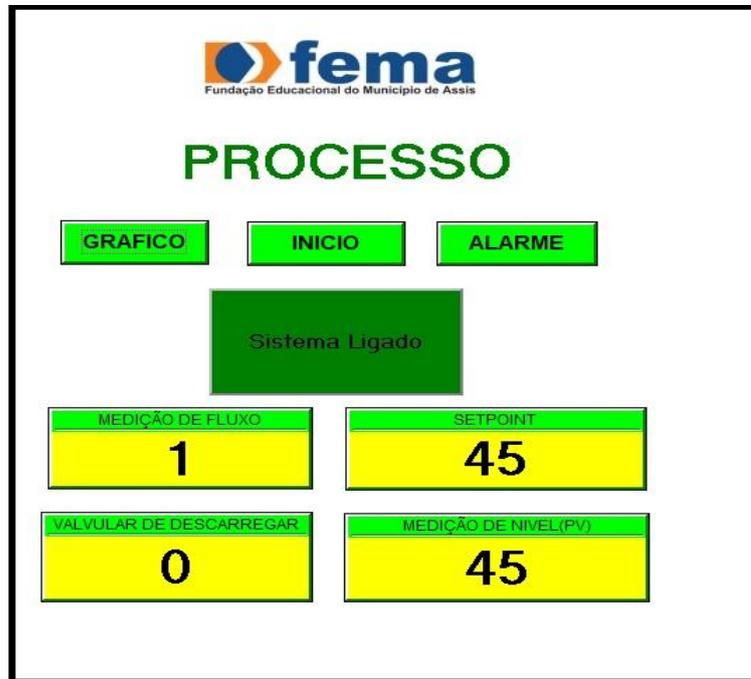


Figura 31 - Tela de Processo com setpoint em 45

A tela de alarmes está registrando um alarme de nível baixo pois o valor do nível do tanque está registrado em 44.

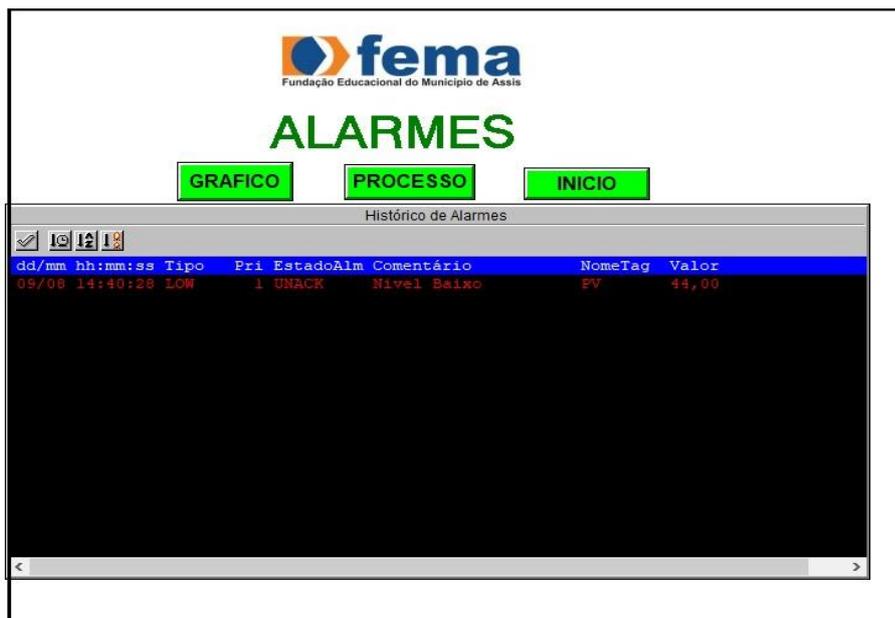


Figura 32 - Alarme de Nível Baixo

O gráfico de nível mostra que o nível de fluido de tanque está no mesmo valor que o parâmetro de SETPOINT ajustado pelo operador.

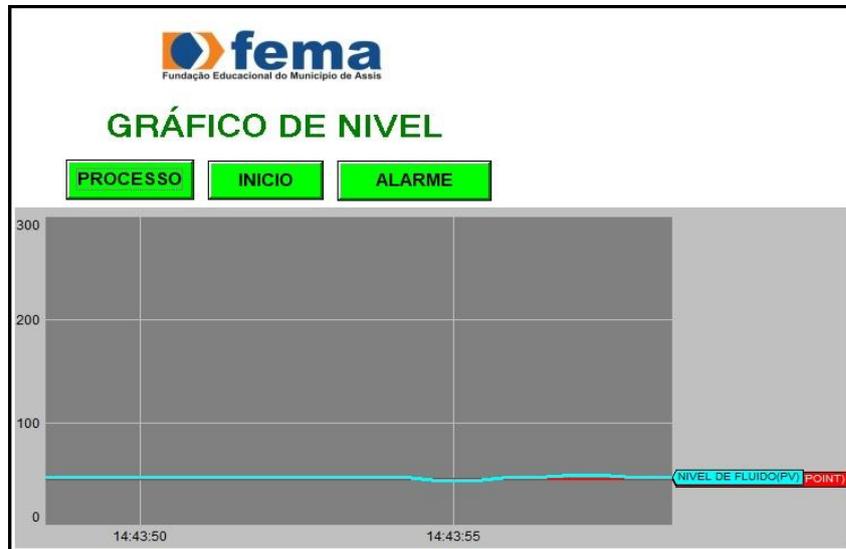


Figura 33 - Gráfico de nível com setpoint em 45

8.2 PARÂMETRO DE SETPOINT VARIANDO DE 45 A 250

Neste teste o parâmetro de SETPOINT será alterado de forma brusca para o valor de 250, com isso a válvula de encher irá atuar bruscamente no tanque.



Figura 34 - Variação do Nível do Tanque

O sistema de processo registra o valor de SETPOINT em 252 e a medição de nível em 76 com a válvula de medição de fluxo subindo em 2 unidades por segundo.

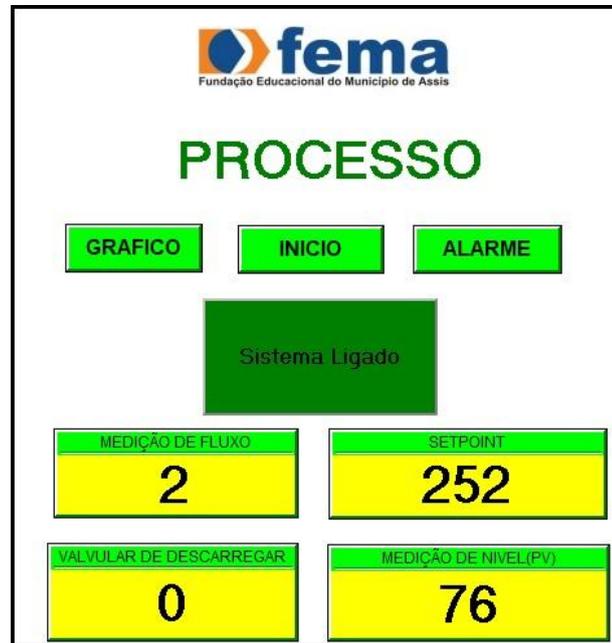


Figura 35 - Variação da Tela de Processo

A tela do gráfico de nível mostra o valor de SETPOINT em 250 com uma linha constante no eixo x e o valor do nível crescente até se igualar com o valor de SETPOINT.

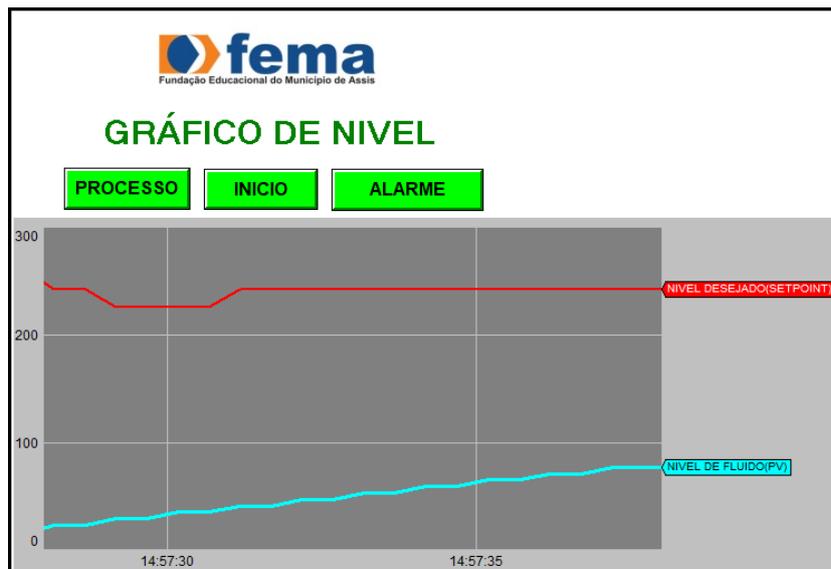


Figura 36 - Variação Gráfico de Nível

A tela de alarmes mostra um aviso de nível de tanque cheio em 253, mostrando um aviso para baixar o fluxo de fluido no tanque.

fema
Fundação Educacional do Município de Assis

ALARMES

GRAFICO PROCESSO INICIO

Histórico de Alarmes

dd/mm	hh:mm:ss	Tipo	Pri	EstadoAlm	Comentário	NomeTag	Valor
09/08	15:00:19	HIGH	1	UNACK	Nível Alto	PV	253,0
09/08	14:59:42	RET	1			PV	51,00
09/08	14:59:38	RET	1			SP	157,0
09/08	14:59:26	LOW	1	UNACK	Nível Baixo	PV	49,00
09/08	14:59:16	LOLO	1	UNACK	SETPOINT Baixo	SP	0,00
09/08	14:59:13	RET	1			PV	55,00
09/08	14:59:02	LOW	1	UNACK	Nível Baixo	PV	24,00

Figura 37 - Avisos de Alarmes de Nível Alto

9 CONCLUSÃO

Tendo em vista o que foi abordado no presente trabalho, a linguagem *Ladder* poderá ser usada pelos profissionais e alunos de ciência da computação, devido ao fato desses profissionais carregarem um grande embasamento de portas lógicas que são semelhantes aos conceitos de *Ladder*, além dos conhecimentos de matemática e lógica que foi ofertado por toda a graduação.

Utilizando os conhecimentos de porta lógica foi desenvolvido um programa em *Ladder* para controlar o nível de um tanque. Após todos os resultados apresentados, é perfeitamente cabível afirmar que os acadêmicos e profissionais da área de tecnologia da informação poderão desenvolver programas para CLPs em *Ladder*.

E devido a grande evolução da indústria 4.0, o setor industrial está buscando por profissionais que tenham conhecimentos lógicos e matemáticos para automatizar seus processos como comprovado neste trabalho os acadêmicos e profissionais de ciência da computação estão capacitados para assumir esse protagonismo na indústria 4.0.

9.1 TRABALHOS FUTUROS

O conteúdo sobre linguagem *Ladder* e a programação de CLPs é um tema muito amplo para ser explorado, devido à fato apresento sugestões para temas futuros de trabalhos:

- Utilização de Sistemas Supervisórios com dispositivos móveis
- Desenvolvimento de Sistema Supervisório.
- Integração de Redes de CLPs
- Controle de Processos Indústrias com Raspberry PI.

10 REFERÊNCIAS

ALMEIDA, De Alexandre Sidnei, **PROTÓTIPO DE AMBIENTE LADDER PARA OS MICROCONTROLADORES 8051 E PIC16F873**,2003 .77 f. Trabalho de Conclusão de Curso (Bacharel em Ciência da Computação) – Universidade Regional de Blumenau,2003.

ANOTONELLI. P .L. **Introdução aos Controladores Lógicos Programáveis**, 1998, Descrito em <<http://www.ejm.com.br/download/Introducao%20CLP.pdf>>, Acesso em: 28 de fev . 2020.

ABRAMOVAY, Ricardo. **“Inteligência artificial pode trazer desemprego e fim da privacidade.”** *Folha de São Paulo*. Matéria de 2.4.2017. Disponível em: <<http://www1.folha.uol.com.br/ilustrissima/2017/04/1871569-inteligencia-artificial-pode-trazer-desemprego-e-fim-da-privacidade.shtml>>. Acesso em 4 de novebro. 2019.

BRETTEL, M. et al. How Virtualization, Decentralization and Network Building Change the Manufacturing Landscape: An Industry 4.0 Perspective. **International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering**, v. 8, n. 1, p. 37-44, Novembro 2014.

CASSILO, Danielle. **Automação e controle**. Universidade Federal Rural Do SemiÁrido (UFERSA). Disponível em: <<http://www2.ufersa.edu.br/portal/view/uploads/setores/166/arquivos/Automacao>>. Acesso em: 28 de fevereiro de 2020.

CODESYS. **Manual CODESYS:** Disponível:
 <<https://www.codesys.com/fileadmin/data/Downloads/Broschueren/CODESYS-Engineering-en.pdf>> Acesso em 08 de fevereiro de 2020.

ELIPSE. **Manual do Elipse ESCADA.** Disponível
 em<<https://www.elipse.com.br/downloads/?cat=69&key=&language=ptbr#header-main>>, Acesso em: 08 de fevereiro de 2020.

FACTORY I/O. **Instrução de FACTORY I/O:** Disponível:
 <<https://docs.factoryio.com/>> acesso em 08 de fevereiro de 2020.

FEMA: **Grade Curricular do Curso de Ciência da Computação:** Disponível
 <https://cepein.femanet.com.br/extrafema/planoCurso.jsp?cod_disc=0138&cod_curso=BCC2&serie=02> acesso em 18 de julho de 2020.

FRANCHI, C.M., **Controladores Lógicos Programáveis – Sistemas Discretos.** 1. ed. São Paulo: Érica, 2008.

F. HEIDRICH, J. F. B. Facó, C. F. B. Reis, “**O Impacto Competitivo na Indústria Brasileira com a Aplicação dos Conceitos da Indústria 4.0**”, SIMPOI, 2017.

GEORGINI, M. **Automação Aplicada - Descrição e Implementação de Sistemas Seqüenciais com PLCs.** 6 Ed. Editora Érica. João Marcelo Georgini 30 DE JULHO DE 2009 – ARRUMAR

GEORGINI, Marcelo. **Automação Aplicada.** 3ª Edição. São Paulo: Editora Érica Ltda, 2002.

MIGOTTO, Matheus W. **Desenvolvimento de Sistemas e Experimentos Acadêmicos para Auxílio Didático a Práticas Laboratoriais nas Áreas de Mecatrônica, Controle e Automação Industrial.** Universidade Estadual de Londrina – UEL – PR, 2009.

SAMPAIO, R Leonardo, **Validação Visual de Programas Ladder Baseada em Modelos**, 2011,69 f Trabalho de Conclusão de Curso (Bacharel em Ciência da Computação) – Universidade Federal de Campina Grande,2011.

SEVERO, Bernardo da S. F. **Curso de Controladores Lógicos Programáveis.** Laboratório de Engenharia Elétrica – LEE -Universidade do Estado do Rio de Janeiro – RJ, 2006.

SILVEIRA, P.R., SANTOS, W.E. **Automação e controle discreto.** São Paulo: Érica, 2008

S. NOGUEIRA, Pedro Guilherme. **Coleta de dados de chão de fábrica via rede serial RS-485 para um software MES utilizando CLPs.** 2010. Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica) - Universidade Estadual de Londrina

STOCLER, Geraldo. **Controlador Lógico Programável.** Centro de formação profissional “José Inácio Peixoto” – Unidade Operacional – SENAI – MG, 2005.

SILVA, M. E. **Controladores Lógicos Programáveis- Ladder.** EEP Escola de Engenharia de Piracicaba. 2007. Apostila.

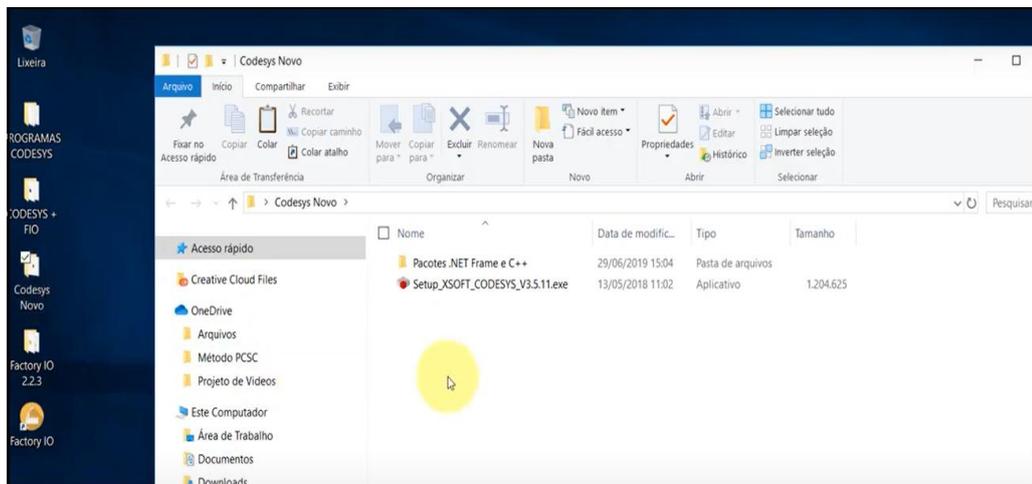
PIRES, J. Norberto. **Automação Industrial.** 1 ed . Lisboa: Editora ETEP (Edições Técnicas e Profissionais), 2002;

VALÉRIO, Marcus. **CLP – Controlador Lógico Programável**. ETEP faculdades, 2002

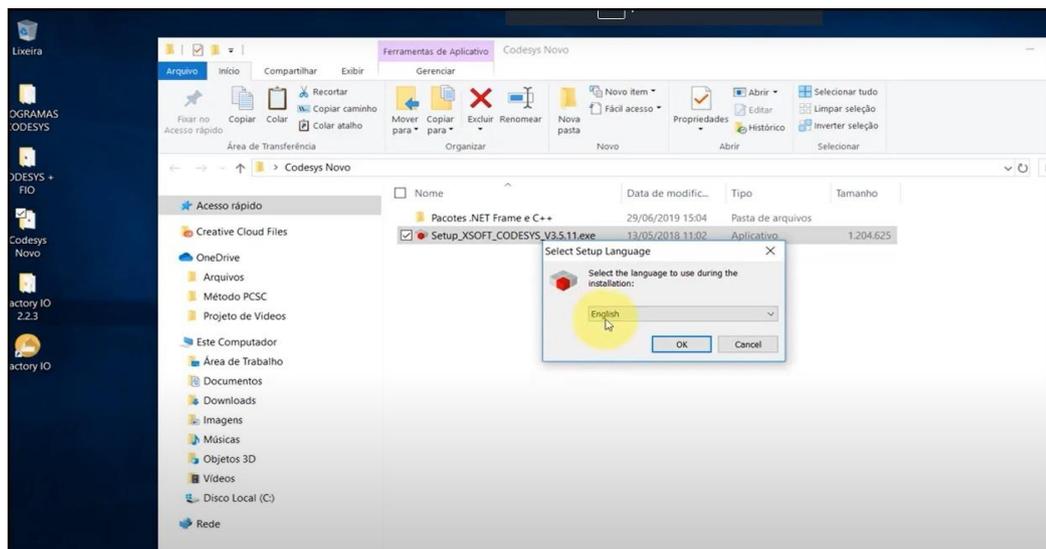
APÊNDICE A – Instalação Codesys

Este apêndice tem como objetivo auxiliar os leitores deste trabalho a realizar a instalação do software *Codesys*.

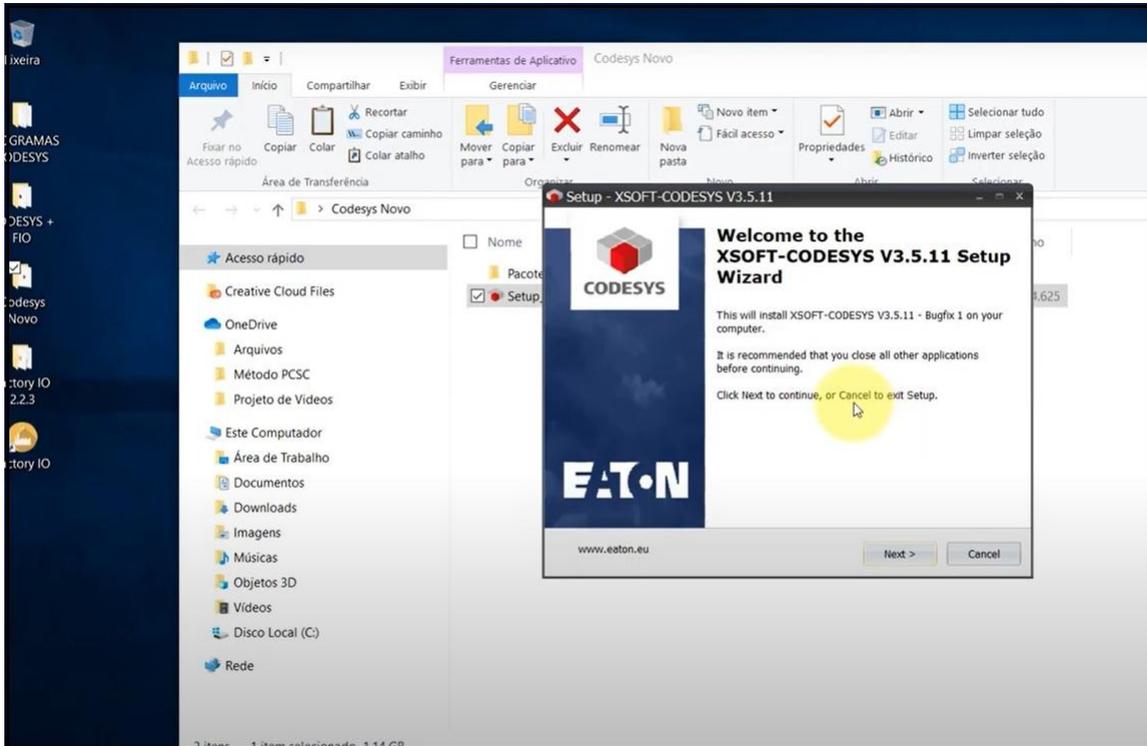
Passo 1



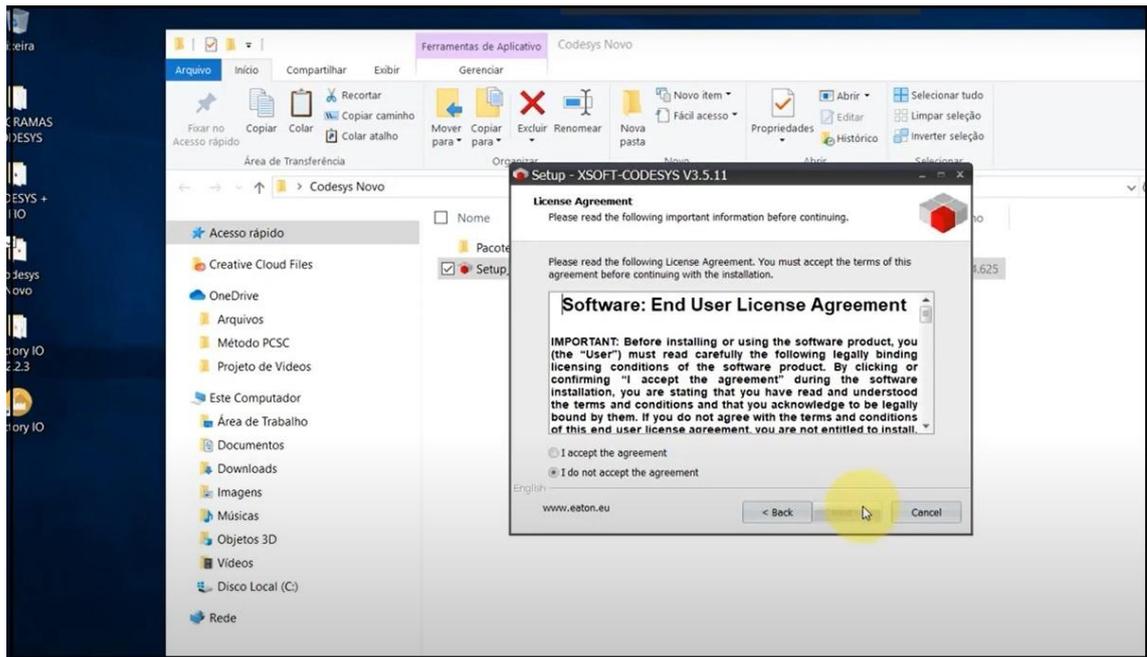
Passo 2



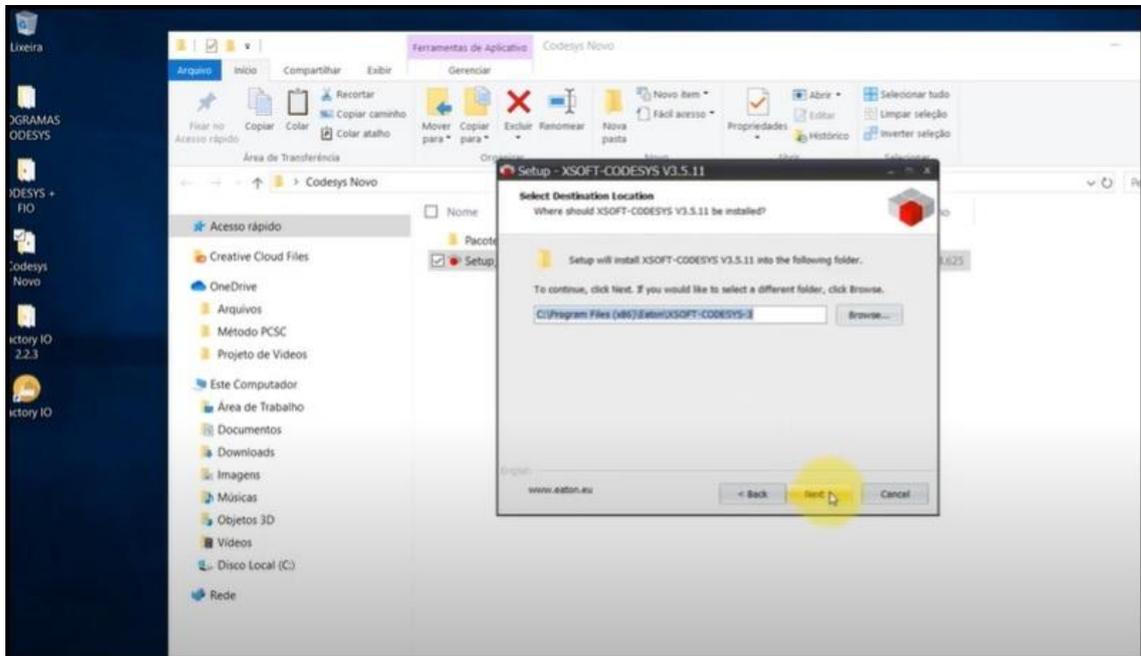
Passo 3



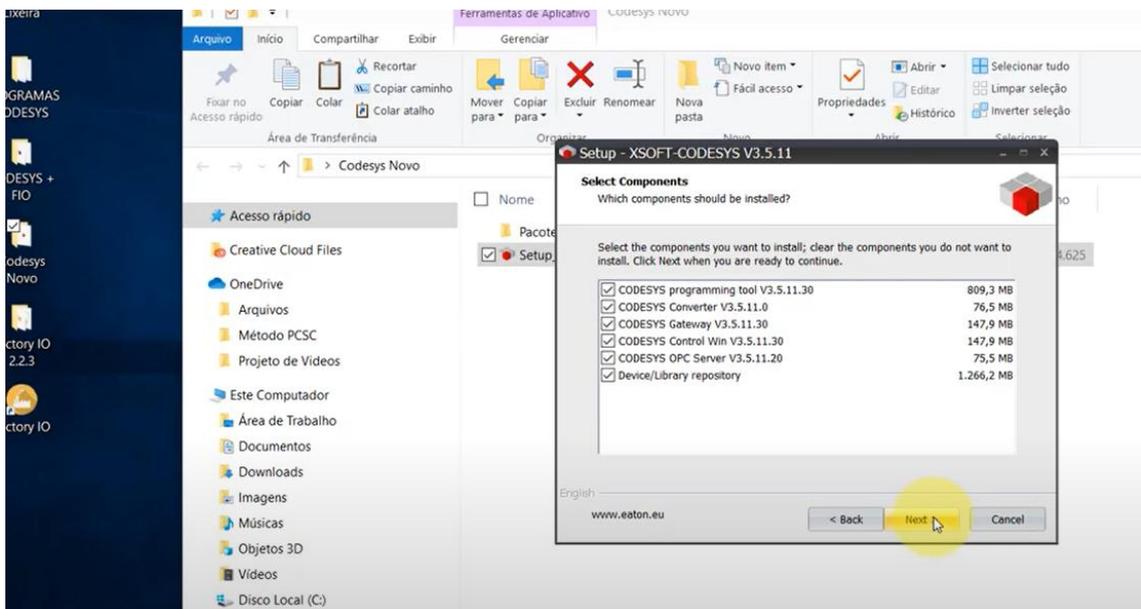
Passo 4:



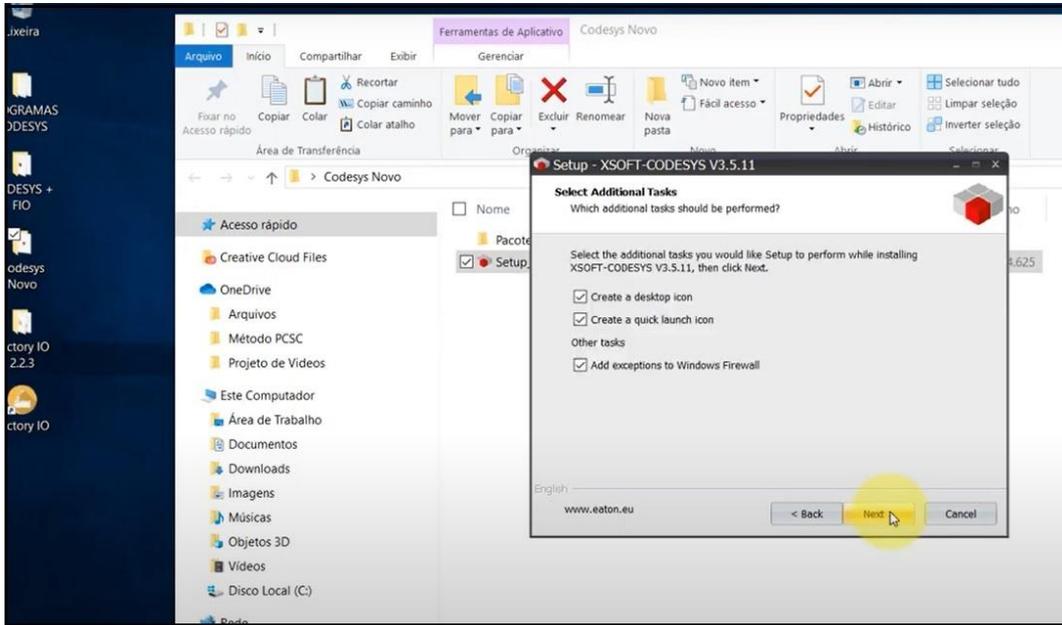
Passo 5:



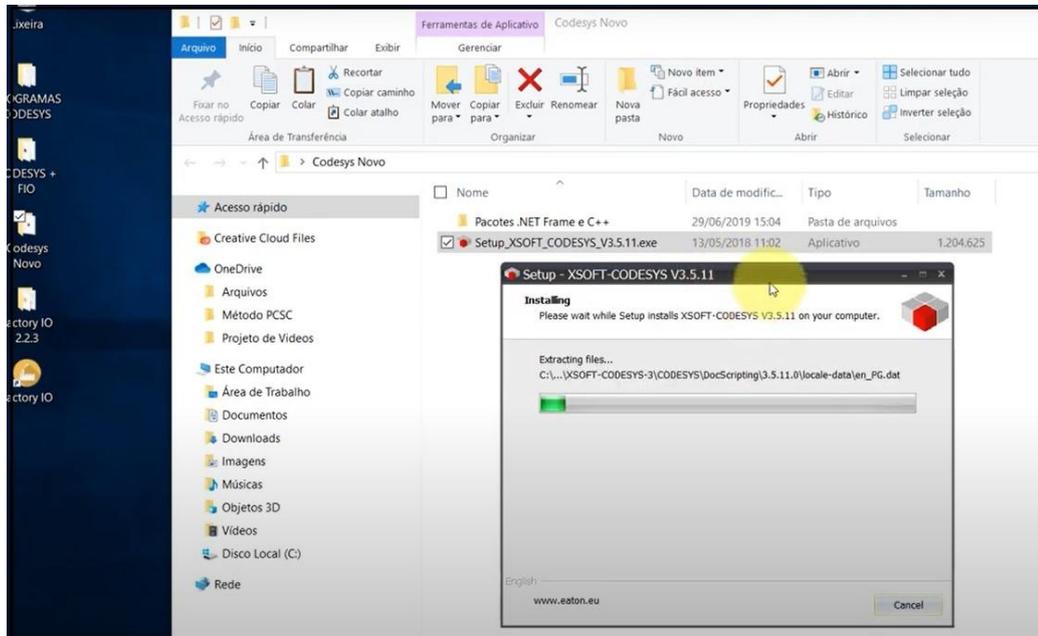
Passo 6:



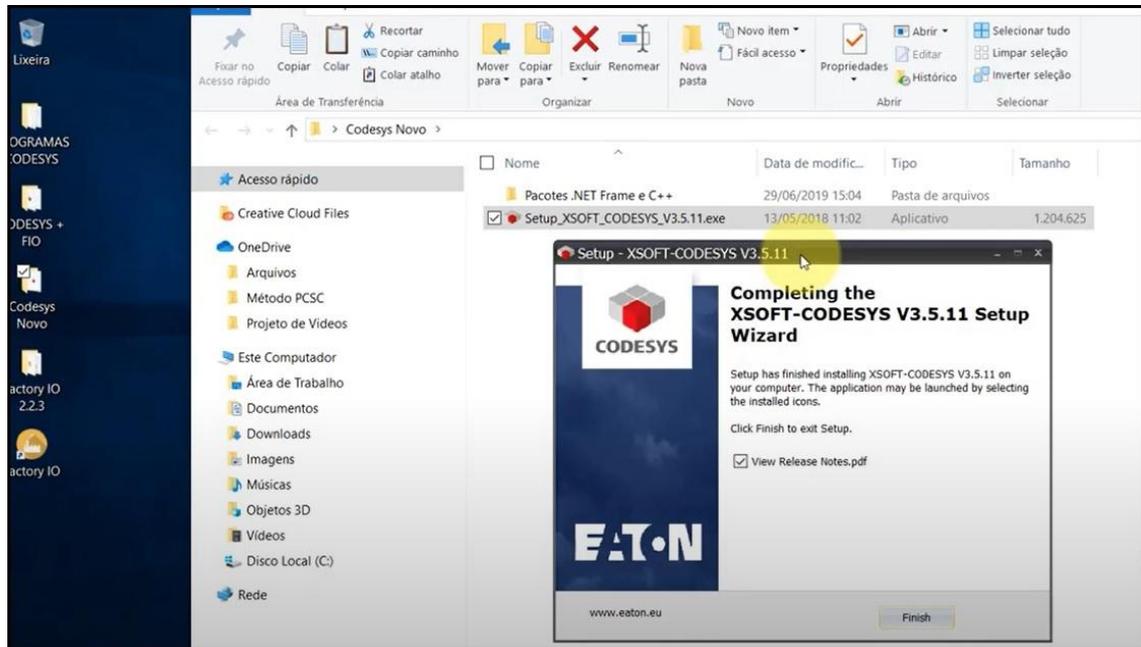
Passo 7:



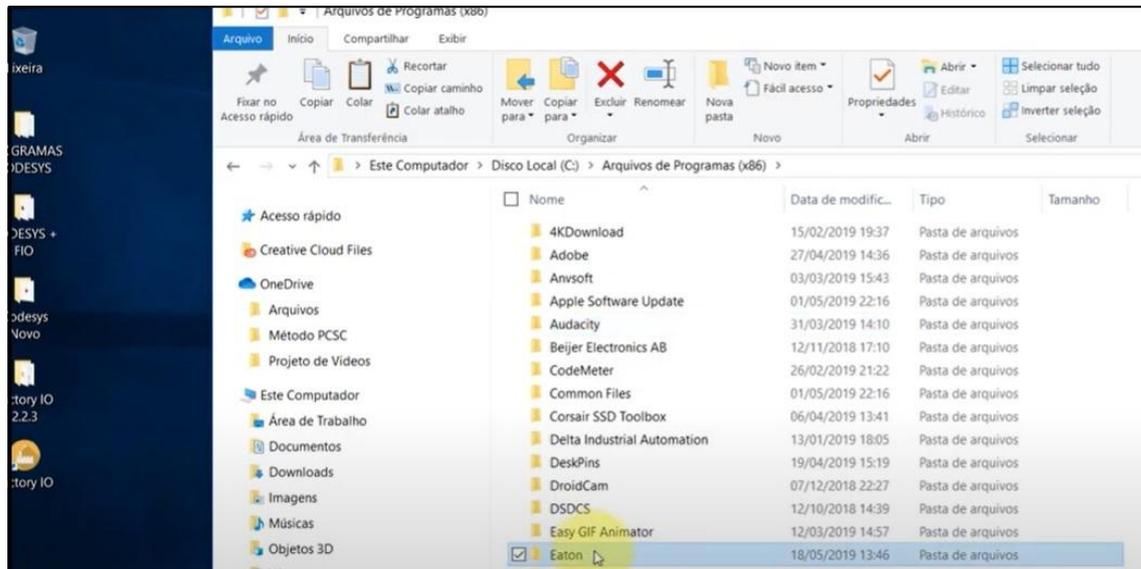
Passo 8:



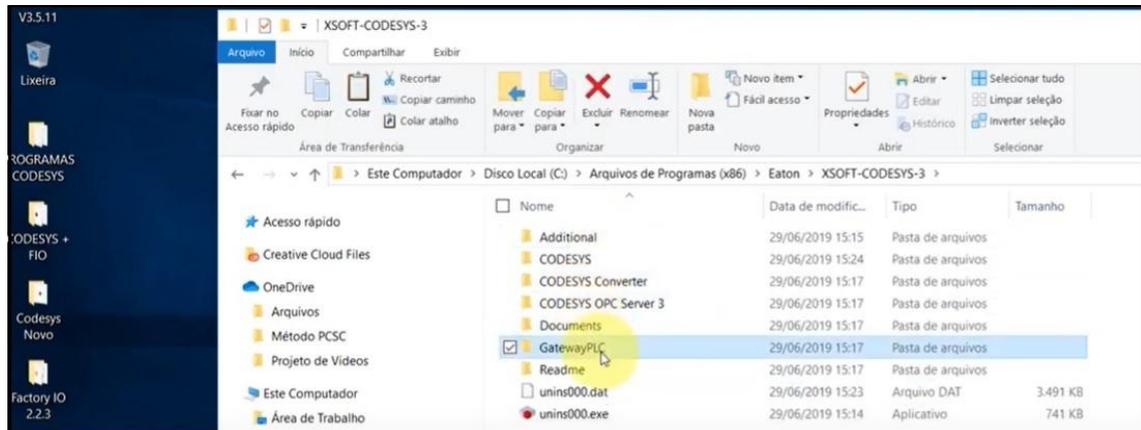
Passo 9:



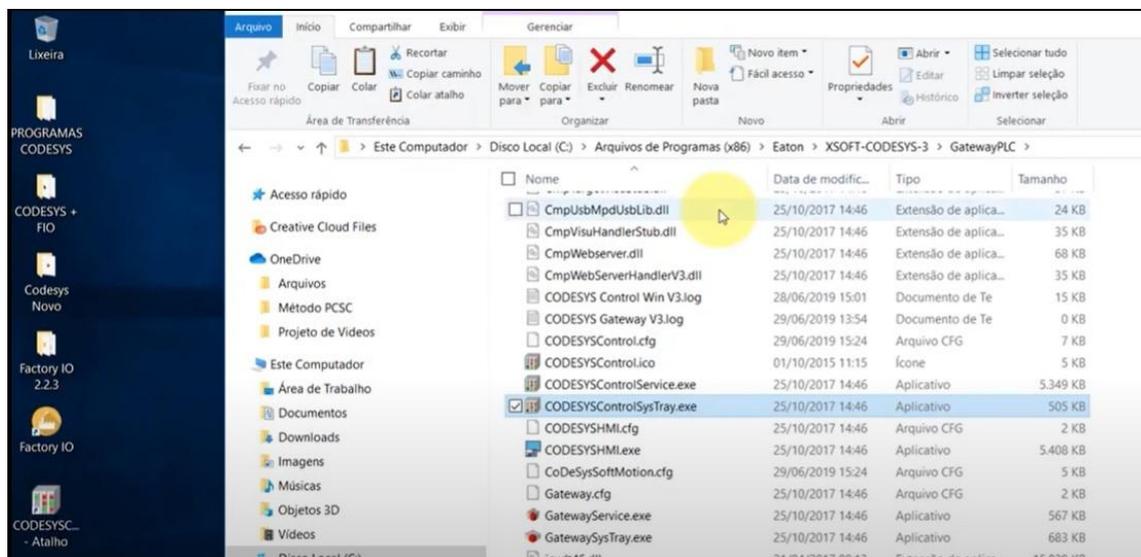
Passo 10: Para este passo é necessário entrar no c: e procurar a pasta Eaton



Passo 11: Procurar a pasta GateWayPLC



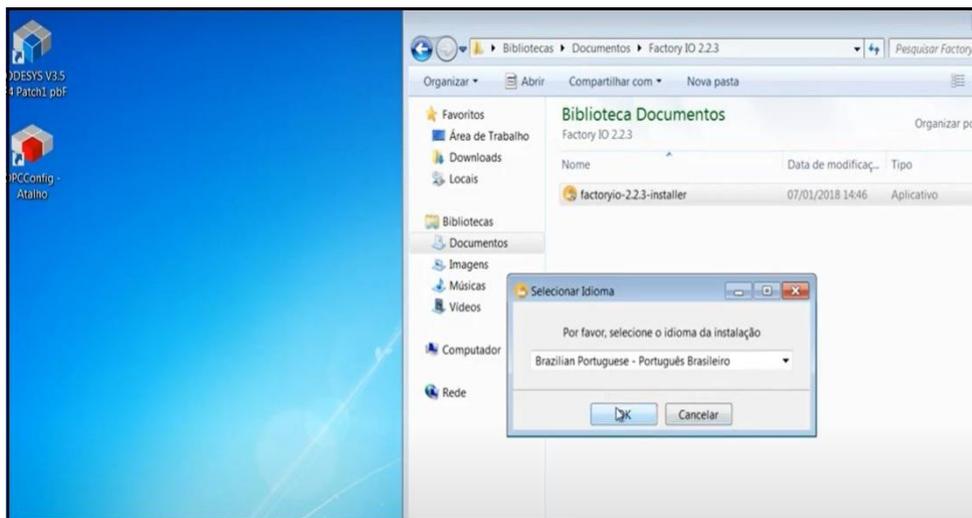
Passo 12: Neste passo é necessário encontrar o item CODESYSControlSysTray.exe e criar um atalho para a área de trabalho, este ícone é responsável pela comunicação do Codesys com o Factory I/O.



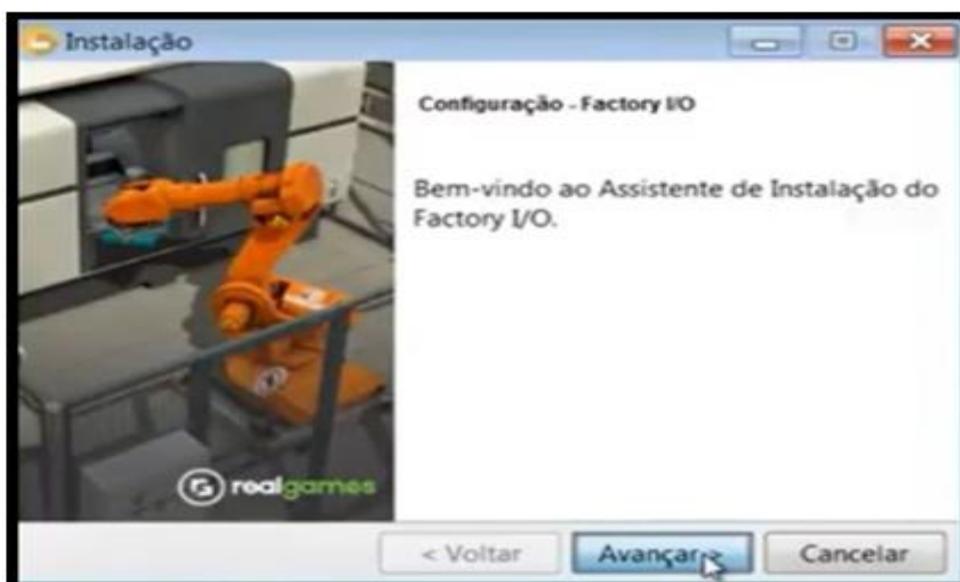
APÊNDICE B – Instalação Factory I/O

Este apêndice tem como objetivo auxiliar os leitores deste trabalho a realizar a instalação do software *Factory I/O*.

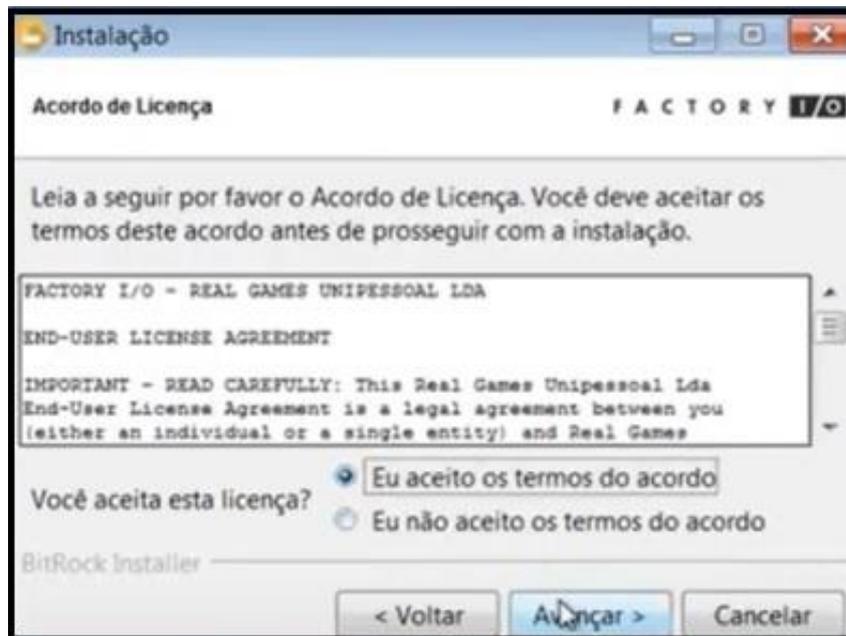
Passo 1:



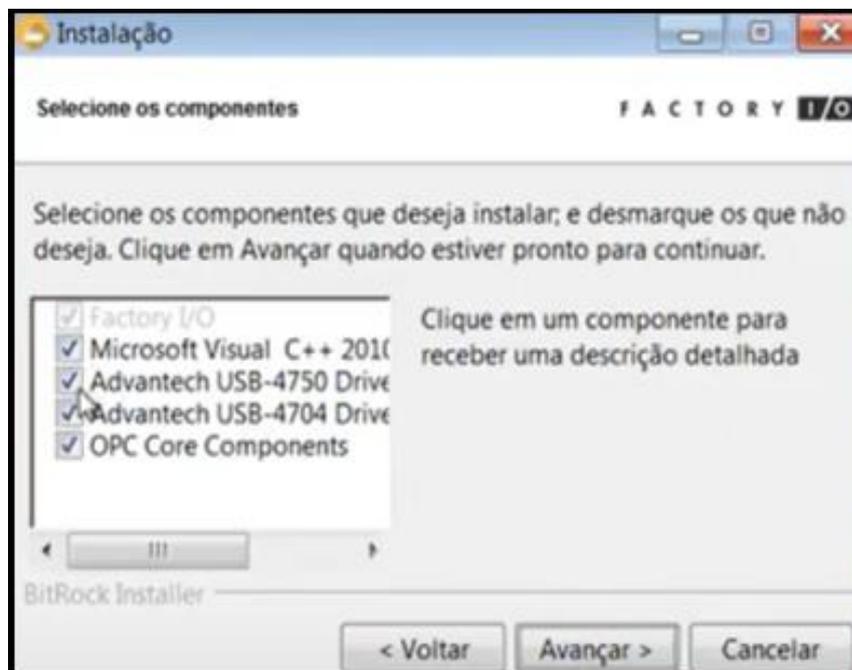
Passo 2:



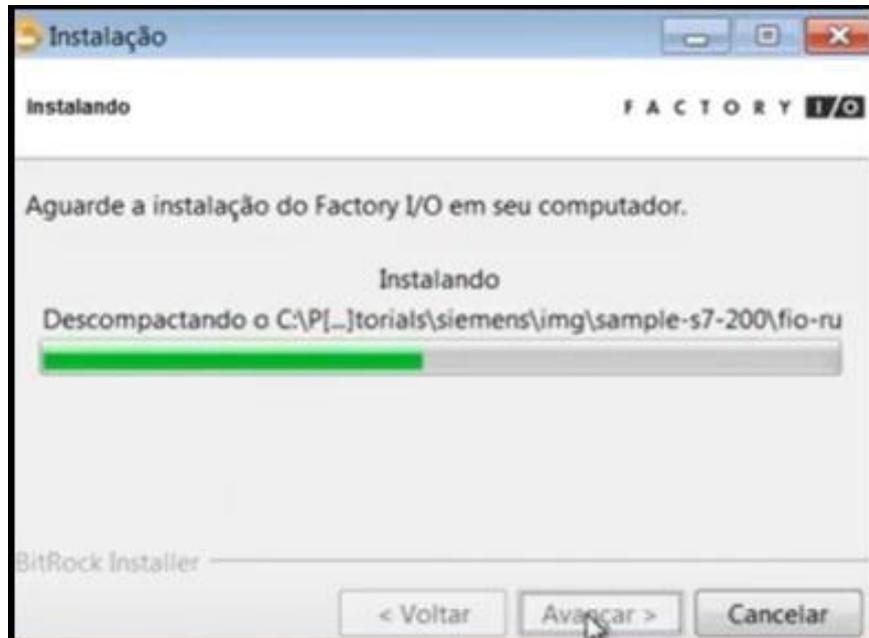
Passo 3:



Passo 4



Passo 5:



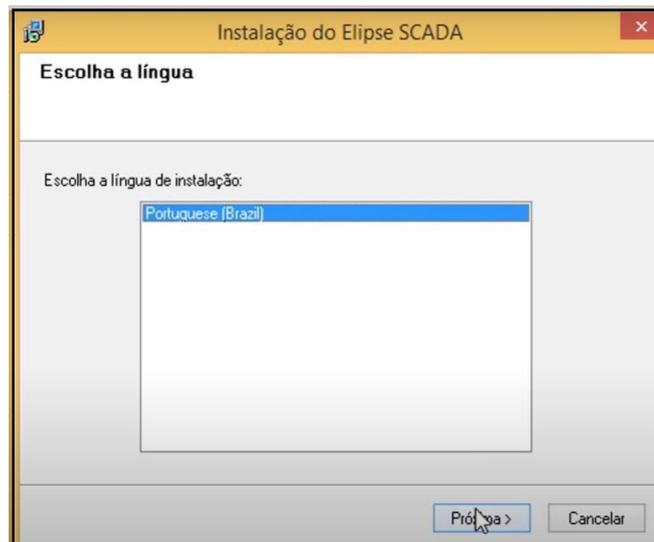
Passo 6 :



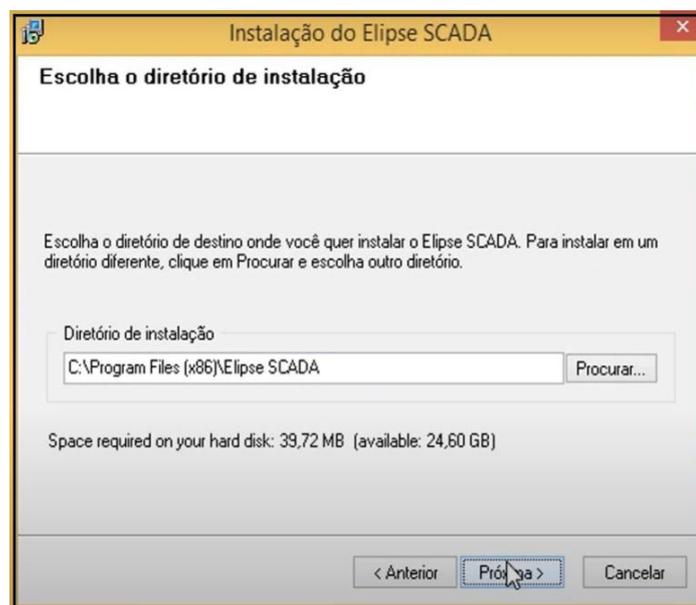
APÊNDICE C – Instalação Elipse SCADA

Este apêndice tem como objetivo auxiliar os leitores deste trabalho a realizar a instalação do software *Elipse Scada*.

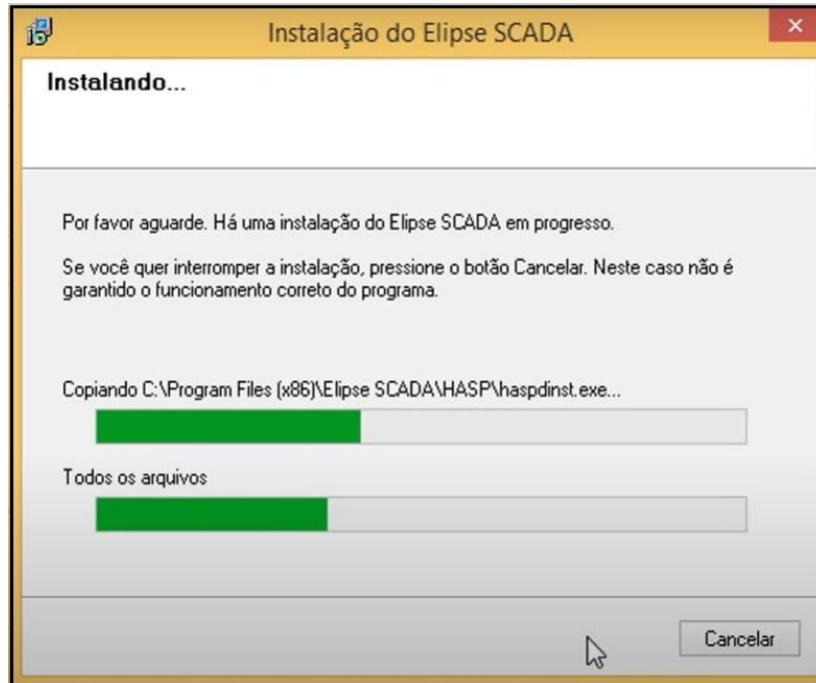
Passo 1:



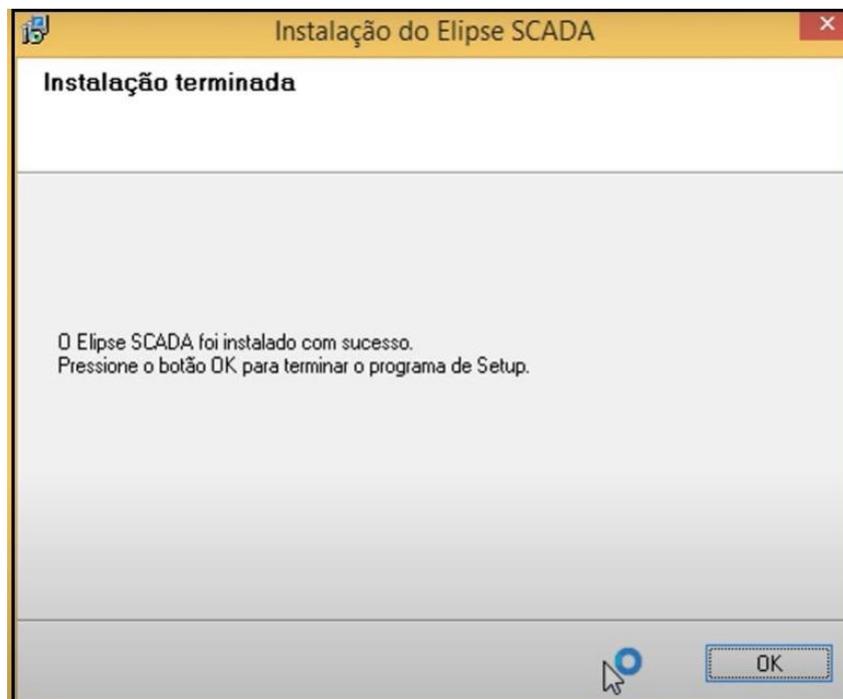
Passo 2 :



Passo 3:



Passo 4:



APÊNDICE D – Programa Básico em Ladder “Hello Word”

Este apêndice irá demonstrar a aplicação mais básica de um diagrama em *ladder*, será demonstrada a programação de uma saída lógica, este comando seria como um “HELLO WORD” em qualquer linguagem de programação comum.

