

Universidade Federal de Campina Grande - UFCG  
Centro de Engenharia Elétrica e Informática - CEEI  
Departamento de Engenharia Elétrica - DEE

Lucas Dantas Pereira

## **Relatório de Estágio Supervisionado**

Campina Grande, Brasil

Setembro de 2024

Lucas Dantas Pereira

## **Relatório de Estágio Supervisionado**

Relatório de Estágio Supervisionado submetido à Coordenação de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande, Campus Campina Grande, como parte dos requisitos necessários para obtenção do título de Graduado em Engenharia Elétrica.

Universidade Federal de Campina Grande - UFCG

Centro de Engenharia Elétrica e Informática - CEEI

Departamento de Engenharia Elétrica - DEE

Orientador: George Acioli Júnior, D.Sc.

Campina Grande, Brasil

Setembro de 2024

Lucas Dantas Pereira

## Relatório de Estágio Supervisionado

Relatório de Estágio Supervisionado submetido à Coordenação de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande, Campus Campina Grande, como parte dos requisitos necessários para obtenção do título de Graduado em Engenharia Elétrica.

Trabalho aprovado em:            /            /

---

**George Acioli Júnior, D.Sc.**  
Orientador

---

**Eisenhaver de Moura Fernandes,**  
**D.Sc.**  
Convidado

Campina Grande, Brasil  
Setembro de 2024

# Agradecimentos

Agradeço primeiramente a Deus por me permitir trilhar meu caminho até aqui. Apesar de todos os dias de estresse e noites sem dormir, também vivi alegrias, conquistas e fiz amizades que levarei para a vida toda.

Agradeço à minha família por sempre estar ao meu lado, proporcionando apoio e suporte, além de todas as condições necessárias para que eu pudesse continuar minha jornada.

Aos meus professores da graduação pelos ensinamentos, em especial, ao meu orientador Dr. George Acioli Júnior por me acompanhar durante o desenvolvimento desse estágio. Também, ao professor Eisenhower de Moura Fernandes, por me supervisionar durante esse processo.

Aos amigos que me acompanham desde a escola e aos que fiz durante o caminho que me apoiam até esse momento.

Por fim, agradeço a mim mesmo por me permitir concluir a graduação, por toda a dedicação e por, mesmo nos momentos difíceis, sempre ter seguido em frente e perseverado sem desistir.

*"Nós somos o que repetidamente fazemos. Excelência, então, não é um ato, mas um hábito."*

*Aristóteles*

# Resumo

Este relatório de estágio descreve o desenvolvimento de um sistema supervisorio para o Sistema Modular de Produção (MPS), desenvolvido pela empresa alemã FESTO, e a criação de um guia de utilização do CODESYS. O estágio foi realizado no Laboratório de Instrumentação Eletrônica e Controle (LIEC) da Universidade Federal de Campina Grande (UFCG), com duração de 227 horas, entre 30 de julho e 20 de setembro de 2024, sob a orientação do professor George Acioli Júnior e supervisão do professor Eisenhower de Moura Fernandes. O guia de utilização foi estruturado para proporcionar aos alunos de graduação uma compreensão prática das funcionalidades do CODESYS e sua conexão com o CLP das estações, abrangendo desde o download e instalação do software até a programação do CLP. A implementação das funcionalidades foi feita de forma a garantir que os alunos possam replicar os passos apresentados com êxito.

**Palavras-chaves:** SCADA; Sistemas Supervisórios; CLP; Visão computacional; LADDER; PYTHON.

# Abstract

This internship report describes the development of a supervisory system for the Modular Production System (MPS), developed by the German company FESTO, and the creation of a user guide for CODESYS. The internship was carried out at the Laboratory of Electronic Instrumentation and Control (LIEC) at the Federal University of Campina Grande (UFCG), lasting 227 hours, from July 30 to September 20, 2024, under the guidance of Professor George Acioli Júnior and the supervision of Professor Eisenhower de Moura Fernandes. The user guide was designed to provide undergraduate students with a practical understanding of CODESYS functionalities and its connection to the station's PLC, covering everything from the software download and installation to PLC programming. The implementation of the functionalities was carried out in such a way that students can successfully replicate the steps presented.

**Key-words:** SCADA; Supervisory Systems; PLC; COMPUTER VISION; LADDER; PYTHON .

# Lista de ilustrações

|   |    |
|---|----|
| Figura 1 – Fachada do LIEC . . . . .  | 2  |
| Figura 2 – Estações do sistema modular de produção disponíveis no LIEC. . . . . | 3  |
| Figura 3 – Peças processadas pelo MPS. . . . .                                  | 4  |
| Figura 4 – Componentes do Sistema de Controle e Interface de Operação. . . . .  | 5  |
| Figura 5 – Adaptações Realizadas . . . . .                                      | 7  |
| Figura 6 – Arquitetura do Sistema SCADA . . . . .                               | 11 |
| Figura 7 – Supervisorio Desenvolvido. . . . .                                   | 13 |
| Figura 8 – Visão computacional . . . . .  | 14 |
| Figura 9 – <i>Download</i> do CODESYS . . . . .                                 | 19 |
| Figura 10 – Download do Target . . . . .  | 20 |
| Figura 11 – Criando um Projeto Padrão . . . . .                                 | 21 |
| Figura 12 – Configurando o Projeto Padrão . . . . .                             | 22 |
| Figura 13 – Janela Inicial do Projeto . . . . .                                 | 22 |
| Figura 14 – CLPs Conectados . . . . .   | 23 |
| Figura 15 – Acessando a área de Devices . . . . .                               | 24 |
| Figura 16 – Versão do Device . . . . .  | 24 |
| Figura 17 – Update de Dervice . . . . .   | 25 |
| Figura 18 – Conectando o CLP ao Gateway . . . . .                               | 26 |
| Figura 19 – CLPs Conectados . . . . .   | 26 |
| Figura 20 – Criando Gateway . . . . .   | 27 |
| Figura 21 – Configuração do gateway. . . . .                                    | 27 |
| Figura 22 – Adição de um dispositivo ao gateway. . . . .                        | 28 |
| Figura 23 – Configuração da comunicação com o CLP. . . . .                      | 28 |
| Figura 24 – Habilitação da comunicação com o CLP. . . . .                       | 29 |
| Figura 25 – CLP da FESTO. . . . .   | 29 |
| Figura 26 – Painel de uma estação do MPS. . . . .                               | 30 |
| Figura 27 – Acessando a Área de Dispositivos. . . . .                           | 31 |
| Figura 28 – Adicionando os Dispositivos. . . . .                                | 31 |
| Figura 29 – Visualização dos módulos do CLP adicionados. . . . .                | 32 |
| Figura 30 – Visualização dos módulos do CLP adicionados. . . . .                | 32 |
| Figura 31 – Tela do programa principal. . . . .                                 | 35 |
| Figura 32 – Inserindo nova linha no diagrama Ladder . . . . .                   | 35 |
| Figura 33 – Blocos disponíveis na linguagem Ladder. . . . .                     | 36 |
| Figura 34 – Inserção de um elemento no diagrama Ladder. . . . .                 | 36 |
| Figura 35 – Inserção de um elemento no diagrama Ladder. . . . .                 | 37 |
| Figura 36 – Exemplo de programação Ladder. . . . .                              | 38 |

|  |    |
|--|----|
| Figura 37 – Upload da Programação. . . . .   | 39 |
| Figura 38 – Programação Funcionando. . . . . | 39 |

# Lista de abreviaturas e siglas

|       |   |
|-------|---|
| LIEC  | Laboratório de Instrumentação Eletrônica e Controle |
| UFMG  | Universidade Federal de Campina Grande              |
| DEE   | Departamento de Engenharia Elétrica                 |
| SCADA | Supervisory Control And Data Acquisition            |
| CLP   | Controladores Lógicos Programáveis                  |
| RTU   | Unidades Terminais Remotas                          |
| IHM   | Interface Homem-Máquina                             |
| MPS   | Sistema Modular de Produção                         |

# Sumário

|            |  |           |
|------------|--|-----------|
| <b>1</b>   | <b>INTRODUÇÃO</b>  | <b>1</b>  |
| <b>2</b>   | <b>LOCAL DO ESTÁGIO</b>  | <b>2</b>  |
| <b>3</b>   | <b>FUNDAMENTAÇÃO TEÓRICA</b>   | <b>3</b>  |
| <b>3.1</b> | <b>Módulo FESTO</b>  | <b>3</b>  |
| 3.1.1      | Componetes do Módulo   | 4         |
| 3.1.1.1    | Peças  | 4         |
| 3.1.1.2    | Sistema de Controle e Interface de Operação                              | 4         |
| 3.1.2      | Estações   | 5         |
| 3.1.2.1    | Estação de Distribuição  | 5         |
| 3.1.2.2    | Estação de separação   | 6         |
| 3.1.2.3    | Estação de Classificação   | 6         |
| 3.1.3      | Integração da Visão Computacional ao Sistema MPS: Motivação e Adaptações | 6         |
| <b>3.2</b> | <b>Sistemas Supervisórios</b>  | <b>7</b>  |
| <b>3.3</b> | <b>SCADA</b>   | <b>9</b>  |
| 3.3.1      | SCADA no Controle de Processos   | 9         |
| <b>4</b>   | <b>ATIVIDADES REALIZADAS</b>   | <b>12</b> |
| <b>4.1</b> | <b>Desenvolvimento do Sistema Supervisório para o MPS</b>                | <b>12</b> |
| 4.1.0.1    | Atividades de Estudo   | 12        |
| 4.1.1      | Desenvolvimento do Sistema Supervisório                                  | 12        |
| <b>4.2</b> | <b>Desenvolvimento do Guia Utilização para o CODESYS</b>                 | <b>14</b> |
| <b>5</b>   | <b>CONCLUSÃO</b>   | <b>16</b> |
| <b>5.1</b> | <b>Perspectivas para o Futuro</b>  | <b>16</b> |
|            | <b>REFERÊNCIAS</b>   | <b>17</b> |
|            | <b>ANEXO A – GUIA DE UTILIZAÇÃO DO CODESYS</b>                           | <b>18</b> |
| <b>A.1</b> | <b>Introdução</b>  | <b>18</b> |
| <b>A.2</b> | <b>Primeiros Passos</b>  | <b>18</b> |
| <b>A.3</b> | <b>Download do CODESYS</b>   | <b>19</b> |
| <b>A.4</b> | <b>Download do Target</b>  | <b>19</b> |
| <b>A.5</b> | <b>Começando o Projeto</b>   | <b>20</b> |
| <b>A.6</b> | <b>Criando um Novo Projeto</b>   | <b>20</b> |

|            |  |           |
|------------|--|-----------|
| <b>A.7</b> | <b>Conectando ao CODESYS</b> . . . . .           | <b>23</b> |
| A.7.1      | Versões Atuais . . . . .                         | 25        |
| A.7.2      | Versões mais antigas . . . . .                   | 27        |
| <b>A.8</b> | <b>Configuração dos Cartões do CLP</b> . . . . . | <b>29</b> |
| <b>A.9</b> | <b>Programação</b> . . . . .                     | <b>34</b> |

# 1 Introdução

Neste documento são descritas as atividades desenvolvidas pelo discente Lucas Dantas Pereira durante o Estágio Supervisionado de 180 horas. O estágio foi realizado no Laboratório de Instrumentação Eletrônica e Controle (LIEC), no período de 30 de julho de 2024 até 20 de setembro de 2024, sob a orientação do professor George Acioli Júnior e a supervisão do professor Eisenhower de Moura Fernandes.

Este estágio supervisionado teve como objetivo o cumprimento da disciplina Estágio Curricular, parte integrante da grade do curso de Engenharia Elétrica da Universidade Federal de Campina Grande (UFCG). A disciplina é fundamental para a formação profissional, proporcionando ao aluno a oportunidade de aplicar na prática os conhecimentos adquiridos ao longo do curso.

A principal atividade desenvolvida durante o estágio foi a criação de um sistema supervisor para o Sistema Modular de Produção fabricado pela empresa alemã FESTO e também confecção de um guia de utilização do CODESYS para esse sistema. Esse guia foi elaborado para auxiliar os alunos na operação e programação da planta.

## 2 Local do Estágio

O Laboratório de Instrumentação Eletrônica e Controle (LIEC) faz parte do Departamento de Engenharia Elétrica (DEE) do Centro de Engenharia Elétrica e Informática (CEEI) da Universidade Federal de Campina Grande (UFCG), localizado no campus de Campina Grande, Paraíba. Fundado em dezembro de 1975, o LIEC desempenha um papel importante no desenvolvimento de pesquisas e projetos inovadores nas áreas de instrumentação eletrônica, controle e automação.

O laboratório ocupa um prédio de aproximadamente 900 m<sup>2</sup> e conta com uma infraestrutura moderna que inclui oito laboratórios bem equipados, duas salas de apoio técnico, uma sala de reuniões e diversas salas destinadas a alunos e professores. O corpo técnico do LIEC é composto por professores doutores, alunos de pós-graduação e graduação, todos empenhados em atividades de pesquisa e desenvolvimento que contribuem significativamente para o avanço da engenharia elétrica.

A figura 1 apresenta a fachada do prédio onde se localiza o LIEC, destacando sua infraestrutura que oferece um ambiente propício para a realização de atividades acadêmicas e de pesquisa.

Figura 1 – Fachada do LIEC



Fonte: Laboratório de Instrumentação Eletrônica e Controle (2024)

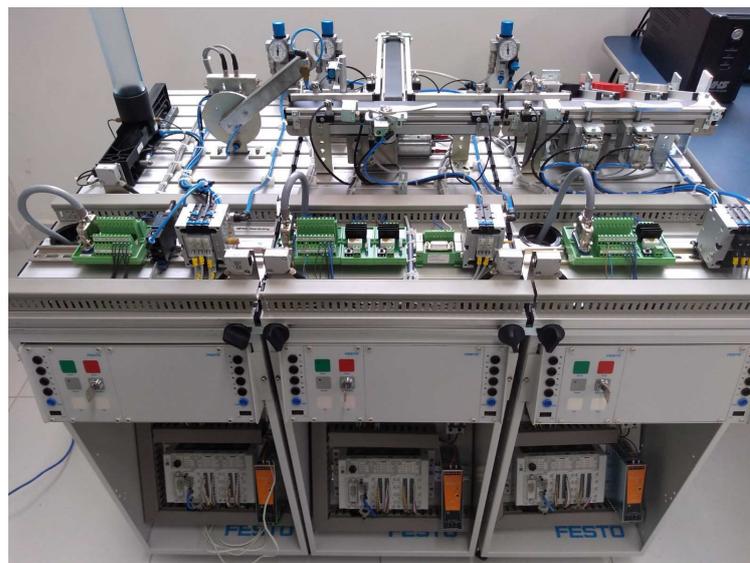
## 3 Fundamentação Teórica

Este capítulo apresenta os conceitos teóricos fundamentais que sustentam as atividades desenvolvidas durante o estágio.

### 3.1 Módulo FESTO

O Sistema Modular de Produção (MPS - *Modular Production System*) fabricado pela empresa alemã FESTO é composto por estações que realizam funções específicas de manufatura em escala laboratorial. Três módulos foram adquiridos pelo LIEC e podem ser combinados para executar tarefas mais complexas. O MPS é utilizado para o treinamento de pessoal em diferentes áreas relacionadas à produção, tais como: planejamento, montagem, programação, comissionamento, operação e manutenção (RAMOS, 2019).

Figura 2 – Estações do sistema modular de produção disponíveis no LIEC.



Fonte: Ramos (2019)

Podemos observar na Figura 2 os três módulos combinados. O primeiro módulo à esquerda é a estação de distribuição, responsável por alimentar a estação seguinte com peças. No meio, temos a estação de separação, que identifica se o orifício da peça está voltado para cima ou para baixo, separando-as em dois caminhos distintos. Por fim, à direita, temos a estação de classificação, responsável por categorizar e separar as peças.

### 3.1.1 Componentes do Módulo

#### 3.1.1.1 Peças

As peças representam produtos em uma linha de produção de manufatura e possuem uma configuração cilíndrica. Uma das bases desses componentes apresenta um orifício que identifica sua posição durante o processamento. O MPS inclui três categorias distintas de componentes: preto, vermelho e metálico, conforme ilustrado na Figura 3.

Figura 3 – Peças processadas pelo MPS.



Fonte: Ramos (2019)

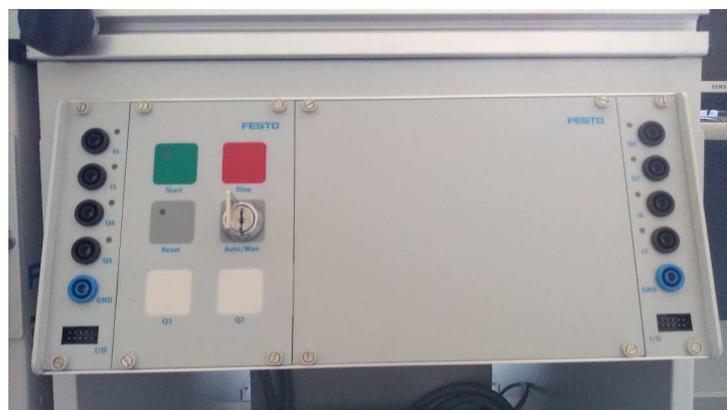
#### 3.1.1.2 Sistema de Controle e Interface de Operação

Cada estação utiliza um controlador lógico programável (CLP) para seu gerenciamento. Além disso, cada estação possui um painel de controle que permite a interação entre o operador e o sistema por meio de botões, lâmpadas e uma interface com o CLP, contendo entradas e saídas (E/S) para comunicação. O CLP utilizado, fabricado pela FESTO, possui uma estrutura modular que inclui a CPU CPX-CEC-C1-V3 e dois cartões de E/S digitais CPX-AB-8-KL-4POL, responsáveis por receber sinais de entrada e enviar sinais de saída para controlar atuadores e sinalizadores. Esta configuração pode ser visualizada na Figura 4, que apresenta o CLP com os módulos e o painel de controle (RAMOS, 2019).

Figura 4 – Componentes do Sistema de Controle e Interface de Operação.



a) CLP da FESTO



b) Painel de uma estação do MPS

Fonte: Ramos (2019)

### 3.1.2 Estações

#### 3.1.2.1 Estação de Distribuição

A estação de distribuição é o módulo inicial do sistema de manufatura, encarregada de fornecer as peças para as próximas estações. Esta estação é composta por dois subsistemas principais: um depósito, onde as peças são armazenadas, e um atuador rotativo equipado com uma ventosa, que transporta as peças do depósito para a estação subsequente (PEREIRA, 2024).

### 3.1.2.2 Estação de separação

A estação de separação é um módulo intermediário, responsável por determinar o trajeto das peças com base em sua posição. Um sensor localizado no início da esteira principal verifica a orientação da peça, gerando um sinal lógico distinto dependendo de o orifício da peça estar voltado para cima ou para baixo (PEREIRA, 2024).

### 3.1.2.3 Estação de Classificação

A estação de classificação é o módulo final do MPS, responsável por separar as peças de acordo com seu tipo: preta, vermelha ou metálica. Para isso, são utilizados dois sensores: um sensor óptico difuso, que identifica peças vermelhas, e um sensor indutivo, que detecta peças metálicas. As peças pretas são identificadas quando nenhum dos sensores gera um sinal lógico alto. Dessa maneira, é possível distinguir entre peças vermelhas, metálicas e pretas, permitindo a separação adequada de cada uma (PEREIRA, 2024).

## 3.1.3 Integração da Visão Computacional ao Sistema MPS: Motivação e Adaptações

Compreendendo o funcionamento de cada estação individualmente e em conjunto, torna-se possível propor uma nova solução para a classificação das peças. No contexto da automação industrial, a visão computacional emerge como uma alternativa viável e eficiente. Utilizando tecnologias avançadas de captura e análise de imagens, é possível realizar todas as classificações necessárias para o funcionamento adequado do MPS, substituindo os métodos tradicionais baseados em sensores (PEREIRA, 2024).

A implementação da visão computacional no MPS permite substituir tanto o sensor responsável pela seleção das peças quanto os dois sensores encarregados da classificação. Esta substituição não apenas simplifica o sistema, mas também aumenta a precisão e a eficiência do processo. Além disso, a visão computacional possibilita uma infinidade de novas formas de operação, como a adição de novos tipos de peças sem a necessidade de *hardware* adicional. Esta flexibilidade permite uma adaptação mais rápida a novas exigências de produção e uma maior capacidade de resposta a mudanças nas características das peças, melhorando significativamente a qualidade do controle no sistema de manufatura (PEREIRA, 2024).

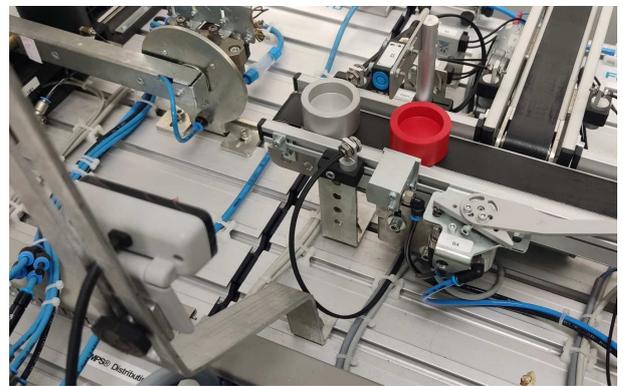
Para incorporar a classificação das peças por meio da visão computacional ao sistema MPS, foram realizadas modificações específicas na estação de separação. Identificada como o primeiro ponto de contato das peças após serem posicionadas pela estação de distribuição, essa estação foi considerada estratégica para a instalação de uma câmera. Assim, um suporte foi fixado na bancada, permitindo posicionar uma *webcam* para capturar imagens em tempo real. Além disso, o suporte inclui um ajuste de altura para o

posicionamento da *webcam* e um orifício para acoplar uma luminária na parte superior, proporcionando uma iluminação adequada da esteira e garantindo uma qualidade de imagem otimizada. Essas adaptações foram cruciais para possibilitar a integração eficaz da visão computacional ao sistema MPS, permitindo a captura e análise precisas das peças em movimento durante o processo de classificação (PEREIRA, 2024). Como pode ser observado na Figura 5.

Figura 5 – Adaptações Realizadas



a) Suporte com Webcam e Luminária



b) Posicionamento do Suporte

Fonte: Pereira (2024)

## 3.2 Sistemas Supervisórios

Um sistema supervisório (também conhecido como SCADA – Supervisory Control and Data Acquisition) é um conjunto de ferramentas e *software* utilizado para monitorar, controlar e supervisionar processos industriais ou infraestruturas em tempo real. Ele permite que operadores humanos interajam com equipamentos como sensores, atuadores e controladores (CLPs, por exemplo) de forma centralizada e visual. O sistema recebe

dados de campo, os processa e os exibe em interfaces gráficas, facilitando a tomada de decisão, identificação de falhas e o controle remoto de dispositivos (PAIOLA, 2016).

O objetivo de um sistema supervisório é fornecer ao usuário a capacidade de exercer controle sobre um dispositivo específico e confirmar seu desempenho de acordo com a ação direcionada. O nome comumente utilizado é Sistema de Controle e Aquisição de Dados Supervisório, ou SCADA (*Supervisory Control And Data Acquisition*) (ACKERMAN; BLOCK, 1992).

A partir da segunda metade do século XX, os sistemas supervisórios começaram a ser implementados, possibilitando que operadores pudessem monitorar e controlar processos industriais à distância. Isso foi possível graças ao desenvolvimento das Interfaces Homem-Máquina (IHM), que ofereciam uma forma visual e interativa de operar máquinas e equipamentos remotamente, trazendo mais eficiência e segurança para a gestão de processos complexos (FIGUEIRÊDO, 2016).

Os sistemas supervisórios são compostos principalmente por três elementos essenciais: telas, base de dados e *drivers* de comunicação. Utilizando diferentes protocolos de comunicação, esses sistemas conseguem adquirir dados de dispositivos de campo, como CLPs, por meio de *drivers* específicos. Os dados coletados são armazenados em uma base de dados, permitindo sua utilização para a atualização e animação das telas do supervisório. Essas telas exibem, de forma visual e em tempo real, diversas informações sobre o processo monitorado, facilitando o acompanhamento e controle das operações industriais (FIGUEIRÊDO, 2016).

Ao tornarem suas operações mais eficientes e confiáveis por meio da visualização em tempo real dos dados de processo, as indústrias começaram a basear seus relatórios e ferramentas de gestão nos dados fornecidos pelos sistemas supervisórios. Essa necessidade, combinada com o avanço dos computadores e a crescente competitividade entre fabricantes, impulsionou a evolução da tecnologia SCADA (Supervisory Control and Data Acquisition), que passou a incorporar melhorias constantes e novas funcionalidades.

Entre as principais funcionalidades dos sistemas supervisórios modernos, destacadas por (PAIOLA, 2016), estão:

- Identificação da dinâmica do processo: permite análises avançadas que auxiliam no gerenciamento da produção, oferecendo maior controle e otimização.
- Facilidade de integração: os sistemas supervisórios podem se integrar com diversas outras plataformas na rede de automação, como PIMS (Plant Information Management System), MES (Manufacturing Execution Systems) e ERP (Enterprise Resource Planning). Isso é viabilizado pela implementação de arquiteturas orientadas a serviços (SOA).

- **Acesso remoto:** possibilita visualização, operação e até desenvolvimento remoto de aplicativos, facilitando a supervisão e manutenção de sistemas distribuídos em grandes distâncias.
- **Tecnologias móveis:** permite o uso de dispositivos como smartphones e tablets, permitindo que operadores e gestores acompanhem e interajam com o sistema de qualquer lugar, seja no chão de fábrica, em salas de controle ou até de suas casas.

### 3.3 SCADA

O sistema supervisorio mais conhecido e difundido no meio industrial é o sistema SCADA. É usado para aquisição de dados (geralmente feita por Controladores Lógico Programáveis) e um sistema de computador digital de uso geral onde é rodado um *software* para o controle supervisorio (QUESADA, 2017).

O sistema opera a partir da coleta de dados dos controladores do processo, que são então processados pelos algoritmos. O usuário determina o *setpoint*, que é enviado ao controlador. Em resposta, o controlador realiza os cálculos necessários para ajustar as saídas do sistema conforme o *setpoint* definido e envia as informações ao sistema supervisorio. O supervisorio acompanha o desempenho do sistema em tempo real, exibindo as informações ao usuário, que pode ajustar o *setpoint* quando necessário (QUESADA, 2017).

#### 3.3.1 SCADA no Controle de Processos

SCADA é um sistema de automação amplamente utilizado para controle remoto e/ou coleta de dados sobre o status dos ativos de campo de qualquer sistema, através de sensores localizados em locais remotos. Esses dados são transmitidos para um local central, seja para controle ou monitoramento, e com base nas informações coletadas, comandos supervisorios são emitidos para os controladores no campo, geralmente conhecidos como dispositivos de campo (SAMI, 2019). O sistema SCADA geralmente inclui os seguintes componentes:

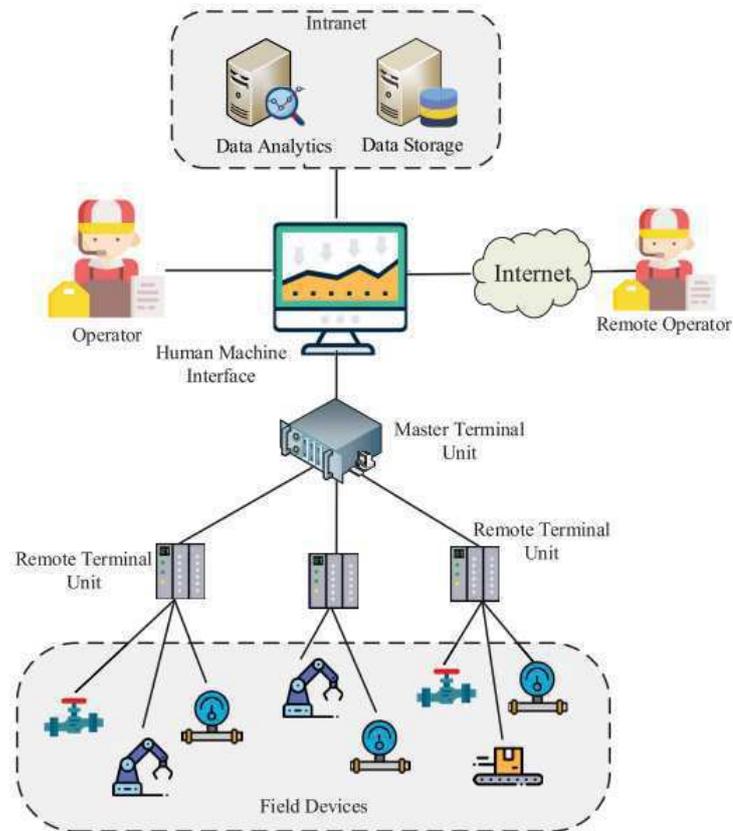
- **Sensores:** Dispositivos para medir dados, também conhecidos como dispositivos de campo.
- **Processadores Locais:** Equipamentos que coletam dados e se comunicam com os instrumentos e equipamentos operacionais do local, como CLP e Unidades Terminais Remotas (RTU).
- **Computadores Host:** Utilizam conexões de rede com fio ou sem fio.

- **Comunicações de Curta Distância:** Entre processadores locais e instrumentos.
- **Computador Host como Ponto Central:** Para monitoramento e controle humano dos processos, armazenamento de bancos de dados, exibição de gráficos de controle estatístico e relatórios.
- **Comunicações de Longa Distância:** Entre processadores locais.

Os sistemas SCADA tradicionais foram projetados para operar de forma independente e dependiam de redes isoladas e protocolos proprietários para garantir a segurança do sistema. Portanto, os projetos iniciais do SCADA nunca incorporaram recursos de segurança. Nos últimos anos, devido à expansão dos negócios e à necessidade de monitoramento centralizado de software distribuído, os sistemas SCADA evoluíram para sistemas abertos sofisticados e complexos conectados à Internet usando tecnologia avançada (YADAV; PAUL, 2021).

Na figura 6 ilustra a arquitetura do sistema SCADA, os dispositivos de campo interagem diretamente com o ambiente físico, coletando dados e executando ações de controle. Estes incluem válvulas, sensores e medidores, essenciais para o monitoramento operacional. As Unidades de Controle Remoto (RTUs) recebem dados dos Dispositivos de Campo e enviam comandos de volta. Elas se comunicam com a Unidade Terminal Mestre (MTU), que coordena a comunicação entre as RTUs e os operadores. A MTU processa os dados recebidos e envia comandos conforme necessário (PLIATSIOS et al., 2020).

Figura 6 – Arquitetura do Sistema SCADA



Fonte: [Pliatsios et al. \(2020\)](#)

A IHM é a plataforma gráfica que permite aos operadores visualizar dados em tempo real, configurar alarmes e tomar decisões baseadas em informações operacionais. A intranet conecta os diversos componentes do sistema, facilitando a troca de dados entre servidores de armazenamento e análise, além da IHM. A análise de dados permite identificar padrões e anomalias, enquanto o armazenamento garante o acesso a informações históricas. Os operadores monitoram e controlam o sistema através da IHM, e o Operador Remoto pode acessar o sistema pela internet para supervisão à distância. A internet facilita a conectividade e o gerenciamento remoto dos sistemas SCADA distribuídos geograficamente ([PLIATSIOS et al., 2020](#)).

## 4 Atividades Realizadas

Durante o estágio supervisionado no LIEC, foram desenvolvidos estudos, seguidos da aplicação prática, para a criação de um sistema supervisório para o Sistema Modular de Produção da FESTO, além do desenvolvimento de um Guia de Utilização do CODESYS para aplicação no MPS. Este capítulo detalha o desenvolvimento, a estruturação, a implementação de funcionalidades, bem como os testes e a validação do guia.

### 4.1 Desenvolvimento do Sistema Supervisório para o MPS

Para viabilizar o monitoramento em tempo real e facilitar a operação das estações, foi desenvolvido um sistema supervisório para o MPS, além de permitir a integração com novas tecnologias por meio da rede industrial implementada para a comunicação entre o sistema supervisório e a planta.

#### 4.1.0.1 Atividades de Estudo

Para a realização do estágio, foi necessário estudar e aprofundar conhecimentos fundamentais. Esse processo incluiu a leitura dos manuais do MPS, fornecidos pela FESTO, e a revisão de conceitos previamente abordados sobre sistemas supervisórios, além de noções básicas relacionadas a redes industriais.

#### 4.1.1 Desenvolvimento do Sistema Supervisório

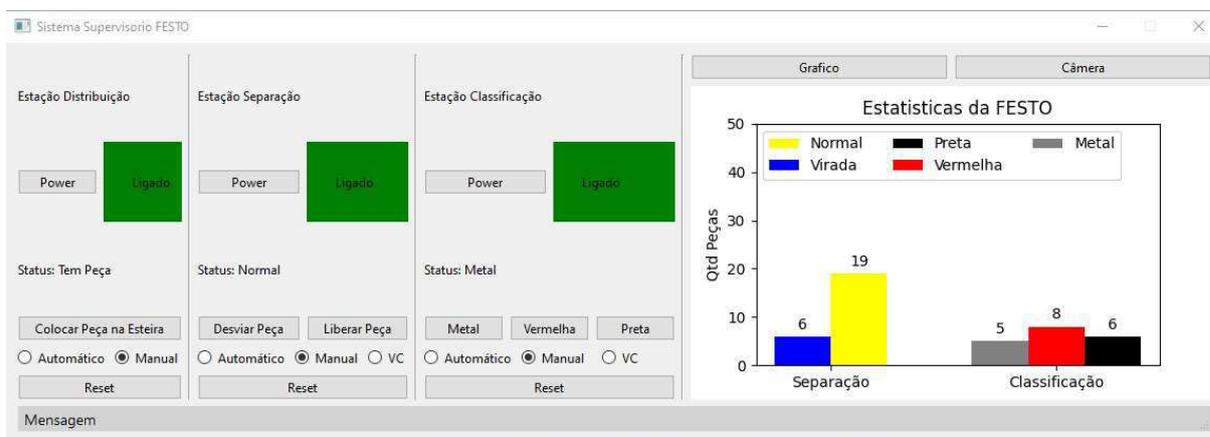
Antes de qualquer modificação, foi realizada uma análise da programação LADDER utilizada em cada estação, desenvolvida no *software* CODESYS, com o objetivo de compreender seu funcionamento. Com esse entendimento, tornou-se necessário adaptar a programação original para permitir a interação do usuário com o sistema, já que, anteriormente, o único acesso à planta era feito por meio do painel de controle. Para isso, foram criados contatos auxiliares na programação, possibilitando a comunicação digital com o sistema supervisório.

Com o auxílio do CODESYS, que oferece uma ferramenta para a criação de um servidor OPC UA integrado à programação e hospedado no CLP, foi possível realizar testes de comunicação entre o CLP e um computador. Para esses testes, utilizou-se o *software* UAexpert, que simula um cliente OPC UA, permitindo a validação da interação entre os dispositivos. A partir dessa configuração, foram implementados três servidores OPC UA, um em cada estação do MPS, viabilizando a comunicação entre o sistema supervisório e as estações.

Para o desenvolvimento do sistema supervisorio, foi utilizada a linguagem *Python*, amplamente reconhecida por sua versatilidade e facilidade de uso. Python oferece suporte nativo para o desenvolvimento de interfaces gráficas por meio de bibliotecas como *PySide6*, além de contar com uma extensa variedade de bibliotecas dedicadas ao aprendizado de máquina, como *TensorFlow* e *Scikit-learn*. Essa flexibilidade permitiu a integração do MPS com algoritmos de visão computacional, otimizando o monitoramento e controle de processos, e possibilitando a automação avançada através de técnicas de análise de imagens e dados em tempo real.

O *software* do sistema supervisorio, apresentado na figura 7, é dividido em quatro regiões principais. As três primeiras regiões correspondem ao supervisorio de cada estação do MPS: Distribuição, Separação e Classificação. A quarta região exibe as estatísticas do processo, representadas graficamente, permitindo o monitoramento em tempo real do processamento das peças no MPS e facilitando a análise de desempenho do sistema. Além disso, nesta região é possível acessar a câmera instalada na planta, cuja finalidade é realizar o processamento das peças utilizando algoritmos de visão computacional.

Figura 7 – Supervisorio Desenvolvido.



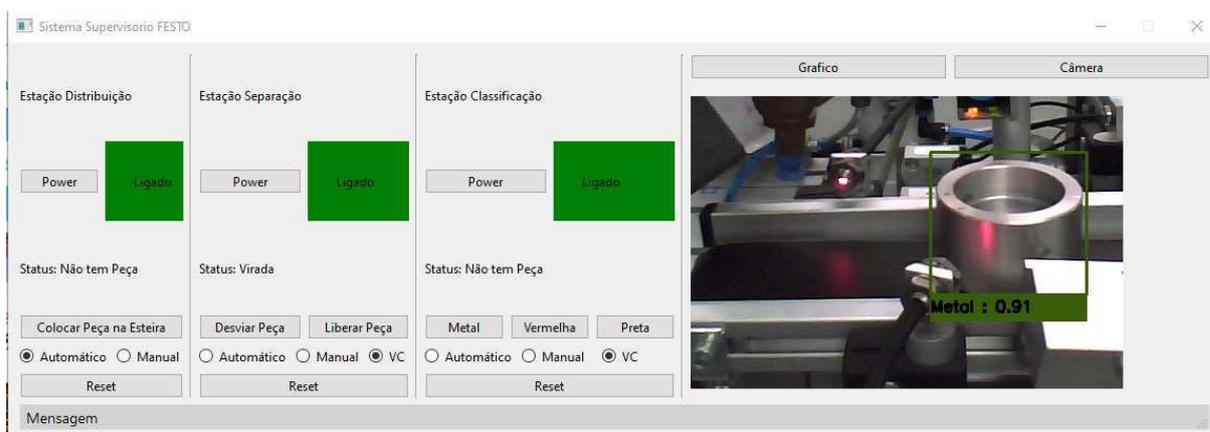
Fonte: autoria própria.

Na interface de cada estação, há um botão para Ligar/Desligar a estação, acompanhado por um indicador que informa o status atual, indicando se a estação está ligada ou desligada. Além disso, cada estação possui uma indicação específica: na estação de Distribuição, o indicador mostra se há peças no estoque; na estação de Separação, informa se a peça está com o orifício voltado para cima ou para baixo; e na estação de Classificação, identifica se a peça é preta, vermelha ou de metal.

Além dos indicadores, a interface disponibiliza botões para o controle manual dos atuadores de cada estação, caso o modo manual esteja ativado. No modo automático, as estações realizam o processamento das peças de forma autônoma, utilizando os sensores para garantir a operação precisa e eficiente conforme o fluxo de trabalho estabelecido.

Ao selecionar o modo de operação "VC", o sistema ativa o modo de visão computacional, no qual a câmera instalada na estação de separação captura imagens das peças e processa-as por meio de um algoritmo de visão computacional previamente treinado para detecção e identificação. Os resultados gerados pelo algoritmo são utilizados pelo *software* supervisorio para o processamento das peças, oferecendo uma alternativa aos sensores tradicionais. Isso proporciona maior flexibilidade no monitoramento do processo, como ilustrado na Figura 8.

Figura 8 – Visão computacional .



Fonte: autoria própria.

## 4.2 Desenvolvimento do Guia Utilização para o CODESYS

O guia foi desenvolvido com o objetivo principal de fornecer aos alunos uma compreensão prática das funcionalidades do CODESYS para a programação do CLP CPX-CEC-C1-V3, disponível no MPS. A estrutura do guia abrange desde a instalação do CODESYS e seus *drivers* até a criação de um projeto e a conexão com o CLP para iniciar a programação.

O guia começa com uma introdução que oferece uma visão geral sobre o CODESYS como ferramenta de programação para sistemas de automação industrial. Em seguida, detalha-se o processo de instalação do CODESYS e dos *drivers* necessários para a comunicação com o CLP CPX-CEC-C1-V3, garantindo que o ambiente de desenvolvimento esteja corretamente configurado.

Na seção de criação de projetos, os usuários são guiados através das etapas para configurar um novo projeto no CODESYS, incluindo a seleção do dispositivo correto (CPX-CEC-C1-V3), a configuração das variáveis e a criação de um programa simples em linguagem LADDER. Exemplos práticos, como a implementação de um botão liga/desliga e a criação de contatos auxiliares, são fornecidos para que os usuários possam se familiarizar com a interface e as funcionalidades básicas do *software*.

Por fim, o guia inclui uma atividade prática onde os usuários podem aplicar os conceitos aprendidos, desenvolvendo um programa completo que controle uma estação do MPS. Essa atividade reforça a compreensão dos principais conceitos e ferramentas do CODESYS, permitindo que os usuários adquiram confiança na programação de CLPs.

A metodologia utilizada para desenvolver este guia combinou pesquisa bibliográfica sobre programação de CLPs e a consulta a manuais e documentação técnica do CODESYS e do CLP CPX-CEC-C1-V3. Além disso, foram realizados testes práticos para validar cada um dos exemplos e procedimentos apresentados no guia, assegurando que os usuários possam replicar com sucesso as instruções fornecidas.

Os procedimentos de teste e validação foram essenciais para garantir a correta implementação das funcionalidades descritas, desde a criação de projetos até a conexão e monitoramento do CLP. Testes funcionais foram realizados para verificar a operação de variáveis e a comunicação com o sistema MPS, assegurando a confiabilidade do guia.

## 5 Conclusão

As atividades realizadas durante o estágio proporcionaram um aprendizado significativo, especialmente na aplicação prática de conceitos teóricos de *SCADA* e *HMI*. A criação do sistema supervisório e do guia laboratorial contribuiu tanto para o meu desenvolvimento profissional quanto para os recursos didáticos do LIEC.

O guia foi estruturado para abordar todos os aspectos essenciais do *software* CO-DESYS, incluindo a criação de projetos, a utilização do CLP CPX-CEC-C1-V3 e a configuração dos módulos de entrada/saída. Dessa forma, proporciona aos usuários o conhecimento necessário para desenvolver e aprimorar suas habilidades em automação industrial.

### 5.1 Perspectivas para o Futuro

Considerando os desafios encontrados neste trabalho, existem algumas perspectivas para a continuidade e melhorias do sistema, tais como:

- Confeccionar e realizar o treinamento com novas peças para o MPS, incluindo a concepção de peças defeituosas para separação;
- Utilizar softwares especializados de sistemas supervisórios, como Elipse ou InTouch.

# Referências

- ACKERMAN, W. J.; BLOCK, W. R. Understanding supervisory systems. *IEEE Computer Applications in Power*, IEEE, v. 5, n. 4, p. 37–40, 1992. Citado na página 8.
- FIGUEIRÊDO, A. F. *Especificação arquitetural do simulador SimuLIHM e desenvolvimento de módulo de comunicação com sistemas supervisórios*. 63 p. Trabalho de Conclusão de Curso - Monografia — Centro de Engenharia Elétrica e Informática, Campina Grande, Paraíba, Brasil, 2016. Acesso em: 18 set. 2024. Disponível em: <<http://dspace.sti.ufcg.edu.br:8080/jspui/handle/riufcg/18586>>. Citado na página 8.
- Laboratório de Instrumentação Eletrônica e Controle. *LIEC - Laboratório de Instrumentação Eletrônica e Controle*. 2024. Acesso em: 31 jul. 2024. Disponível em: <<https://liec.dee.ufcg.edu.br/>>. Citado na página 2.
- PAIOLA, C. E. G. O papel do supervisório no atual contexto tecnológico. *Intech*, p. 6–18, 2016. Acesso em: 18 set. 2024. Disponível em: <[http://www.aquarius.com.br/Boletim/InTech132\\_artigo.pdf](http://www.aquarius.com.br/Boletim/InTech132_artigo.pdf)>. Citado na página 8.
- PEREIRA, L. D. *Desenvolvimento de um sistema de classificação de peças em ambiente industrial utilizando visão computacional e a arquitetura YOLO*. Campina Grande, Paraíba, Brasil, 2024. Curso de Bacharelado em Engenharia Elétrica, Centro de Engenharia Elétrica e Informática. Disponível em: <<http://dspace.sti.ufcg.edu.br:8080/jspui/handle/riufcg/37283>>. Citado 3 vezes nas páginas 5, 6 e 7.
- PLIATSIOS, D. et al. A survey on scada systems: secure protocols, incidents, threats and tactics. *IEEE Communications Surveys & Tutorials*, IEEE, v. 22, n. 3, p. 1942–1976, 2020. Citado 2 vezes nas páginas 10 e 11.
- QUESADA, R. C. *Controle e automação de processos industriais*. Londrina: Editora e Distribuidora Educacional SA, 2017. Citado na página 9.
- RAMOS, E. T. G. *Relatório de estágio supervisionado*. Campina Grande, Paraíba, Brasil, 2019. Curso de Bacharelado em Engenharia Elétrica, Centro de Engenharia Elétrica e Informática. Disponível em: <<http://dspace.sti.ufcg.edu.br:8080/jspui/handle/riufcg/20730>>. Citado 9 vezes nas páginas 3, 4, 5, 27, 28, 29, 30, 33 e 34.
- SAMI, A. *Scada (supervisory control and data acquisition)*. 2019. Citado na página 9.
- YADAV, G.; PAUL, K. Architecture and security of scada systems: A review. *International Journal of Critical Infrastructure Protection*, Elsevier, v. 34, p. 100433, 2021. Citado na página 10.

# ANEXO A – Guia de Utilização do CODESYS

## A.1 Introdução

A introdução de um Controlador Lógico Programável (CLP) em um ambiente de automação exige conhecimento técnico e compreensão dos processos iniciais de configuração. Esses controladores são peças-chave para o funcionamento eficiente de sistemas automatizados, possibilitando a integração e o controle preciso de diferentes equipamentos. Para quem está começando com um CLP, é crucial seguir os passos corretos desde o início para garantir que o dispositivo seja configurado adequadamente, evitando problemas e assegurando o desempenho esperado.

Neste guia, abordaremos detalhadamente as etapas necessárias para configurar e iniciar um projeto utilizando o CLP **CPX-CEC-C1-V3**. Desenvolvido pela empresa alemã FESTO, conhecida por sua excelência em soluções de automação, este CLP oferece diversas funcionalidades que podem otimizar os processos industriais. Aqui, você aprenderá como fazer o *download* do *software* Codesys e de seu *driver*, realizar as configurações iniciais e começar a programar seu CLP, permitindo que você tire o máximo proveito dessa poderosa ferramenta de automação.

## A.2 Primeiros Passos

CODESYS (*Controller Development System*) é um *software* de código aberto amplamente utilizado como ferramenta de programação para sistemas de automação. Baseado na norma internacional IEC 61131-3, o CODESYS é ideal para o desenvolvimento de projetos com Controladores Lógico Programáveis (CLPs), oferecendo uma interface intuitiva e recursos técnicos avançados que facilitam o uso e tornam a programação mais eficiente.

A plataforma permite a visualização de processos industriais através de uma Interface Homem-Máquina (IHM). Além disso, algumas versões do software oferecem a possibilidade de monitoramento remoto via *tablets* e *smartphones*, o que amplia a flexibilidade e a capacidade de acompanhamento dos processos automatizados.

O CODESYS suporta cinco linguagens de programação: Texto Estruturado, Lista de Instruções, Diagrama de Fluxo, Linguagem Ladder e Diagrama de Blocos. Isso permite que o usuário escolha e combine as linguagens mais adequadas para cada aplicação,

aumentando a funcionalidade do sistema de controle. Como uma ferramenta integrada à Indústria 4.0, o software se destaca por sua interface aberta, robustas opções de segurança e pela facilidade na troca de dados entre redes IIoT, além de possibilitar o desenvolvimento de controladores de borda.

### A.3 Download do CODESYS

O *download* do CODESYS está disponível no site <https://store.codesys.com/en/>. Para realizar o *download*, é necessário criar uma conta no site. Após o cadastro, o software estará disponível para *download* de forma gratuita.

O driver do CLP CPX-CEC-C1-V3 é compatível com as versões mais recentes do CODESYS. No entanto, para garantir a compatibilidade com a versão *target* instalada nos CLPs da FESTO adquiridos pelo LIEC, é necessário baixar a versão do software para sistemas de 32 bits, conforme ilustrado na figura 9.

Figura 9 – *Download* do CODESYS



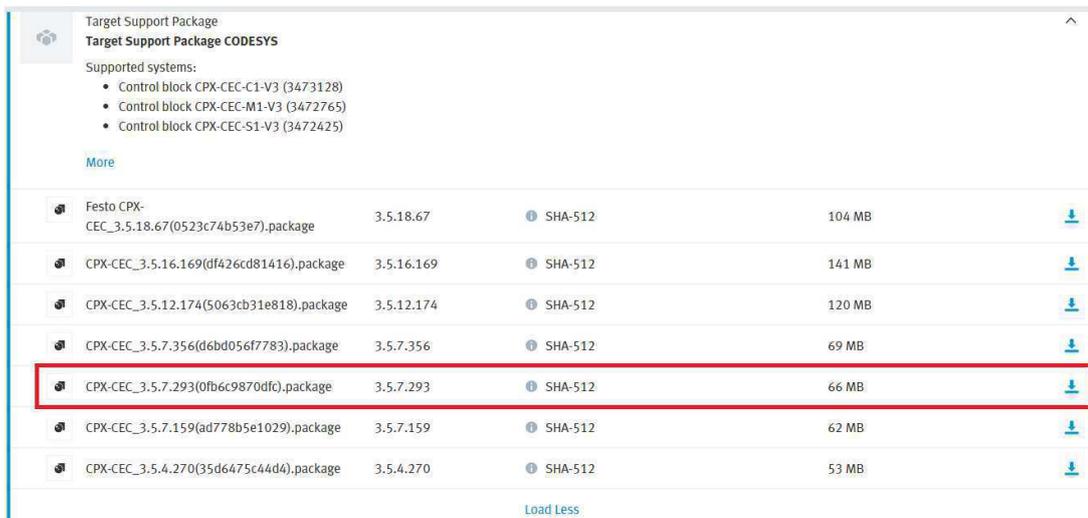
Fonte: autoria própria.

### A.4 Download do Target

Após realizar a instalação padrão do CODESYS, é necessário fazer o *download* do *target* específico para o CLP, disponibilizado pela FESTO. Esse *target* é um pacote de *drivers* responsável por permitir que o *software* reconheça o CLP, além de adicionar todos os módulos e bibliotecas necessários para o pleno funcionamento do controlador. O *download* do *Target* está disponível em: <https://www.festo.com/gb/en/a/3473128/>. Acessando a área de **downloads** e em seguida **software**.

A versão a ser instalada do *target* é a **3.5.7.293** como pode ser visto na figura 10

Figura 10 – Download do Target



Target Support Package  
Target Support Package CODESYS

Supported systems:

- Control block CPX-CEC-C1-V3 (3473128)
- Control block CPX-CEC-M1-V3 (3472765)
- Control block CPX-CEC-S1-V3 (3472425)

[More](#)

|  |                  |                |              |                                 |
|--|------------------|----------------|--------------|---------------------------------|
| Festo CPX-CEC_3.5.18.67(0523c74b53e7).package  | 3.5.18.67        | SHA-512        | 104 MB       | <a href="#">Download</a>        |
| CPX-CEC_3.5.16.169(df426cd81416).package       | 3.5.16.169       | SHA-512        | 141 MB       | <a href="#">Download</a>        |
| CPX-CEC_3.5.12.174(5063cb31e818).package       | 3.5.12.174       | SHA-512        | 120 MB       | <a href="#">Download</a>        |
| CPX-CEC_3.5.7.356(d6bd056f7783).package        | 3.5.7.356        | SHA-512        | 69 MB        | <a href="#">Download</a>        |
| <b>CPX-CEC_3.5.7.293(0fb6c9870dfc).package</b> | <b>3.5.7.293</b> | <b>SHA-512</b> | <b>66 MB</b> | <b><a href="#">Download</a></b> |
| CPX-CEC_3.5.7.159(ad778b5e1029).package        | 3.5.7.159        | SHA-512        | 62 MB        | <a href="#">Download</a>        |
| CPX-CEC_3.5.4.270(35d6475c44d4).package        | 3.5.4.270        | SHA-512        | 53 MB        | <a href="#">Download</a>        |

[Load Less](#)

Fonte: autoria própria.

Após a instalação do *target*, o CODESYS estará pronto para estabelecer a comunicação com o CLP, permitindo que você inicie a programação e configuração do controlador sem problemas.

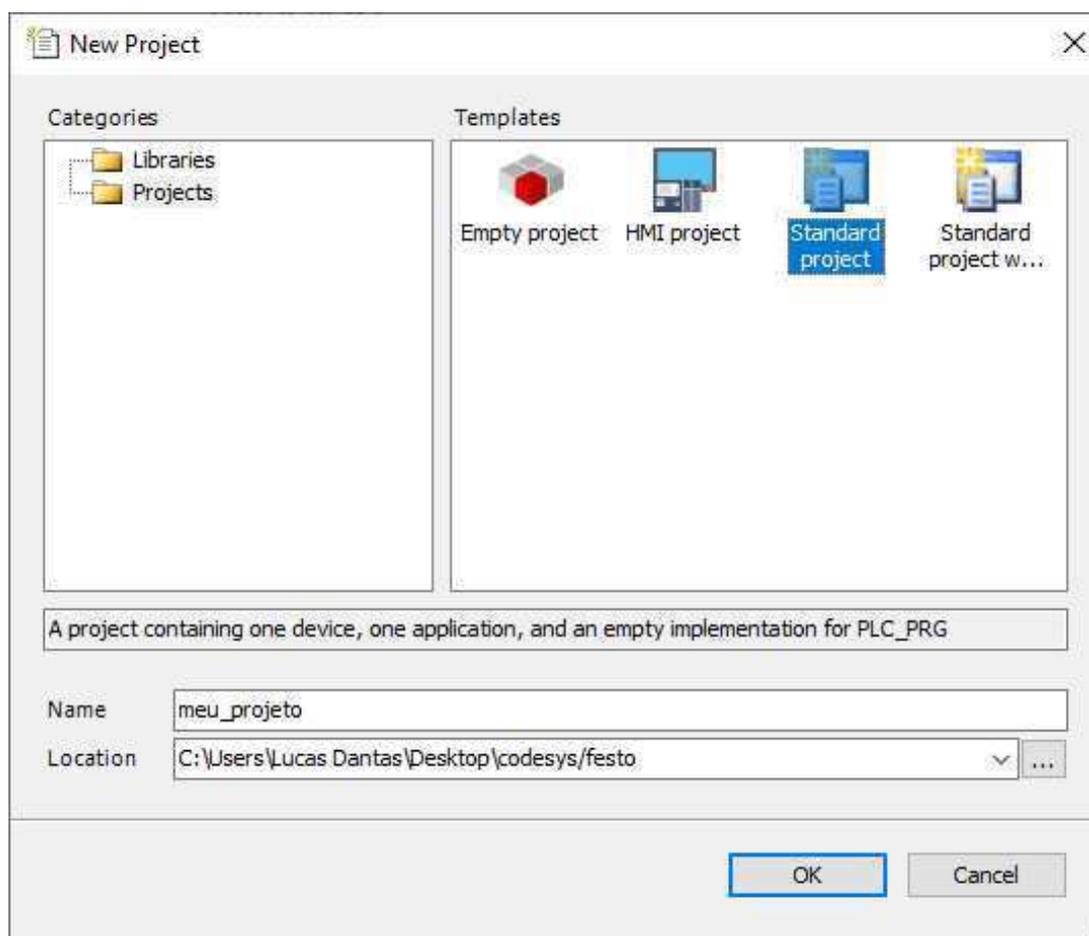
## A.5 Começando o Projeto

Agora que todos os requisitos anteriores foram concluídos, podemos iniciar o CODESYS e criar nosso primeiro projeto.

## A.6 Criando um Novo Projeto

Para começar, abra o CODESYS e clique em **New Project....** Uma janela será exibida para a criação de um novo projeto. Selecione **Standard Project** para criar um projeto padrão, insira um nome para o seu projeto, escolha a pasta de destino como pode ser visto na figura 11. por fim, clique em **OK** para finalizar a configuração.

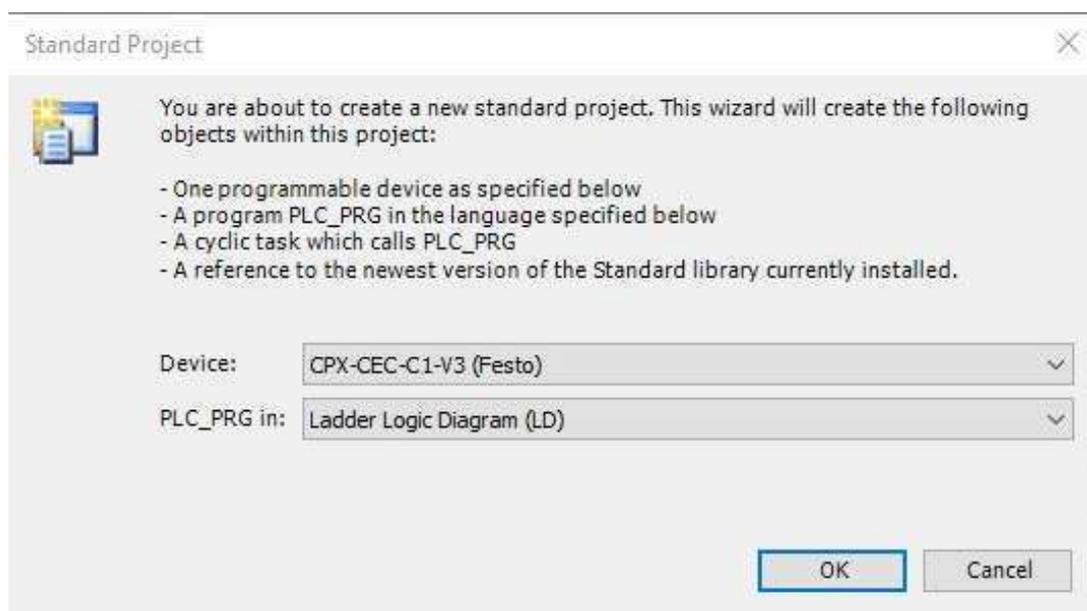
Figura 11 – Criando um Projeto Padrão



Fonte: autoria própria.

Na figura 12, é exibida a janela para a seleção do dispositivo e da linguagem de programação desejada. No primeiro campo, selecione o **CLP CPX-CEC-C1-V3 (Festo)**. A linguagem de programação pode ser escolhida de acordo com sua preferência; para exemplificação, será utilizada a linguagem **Ladder**.

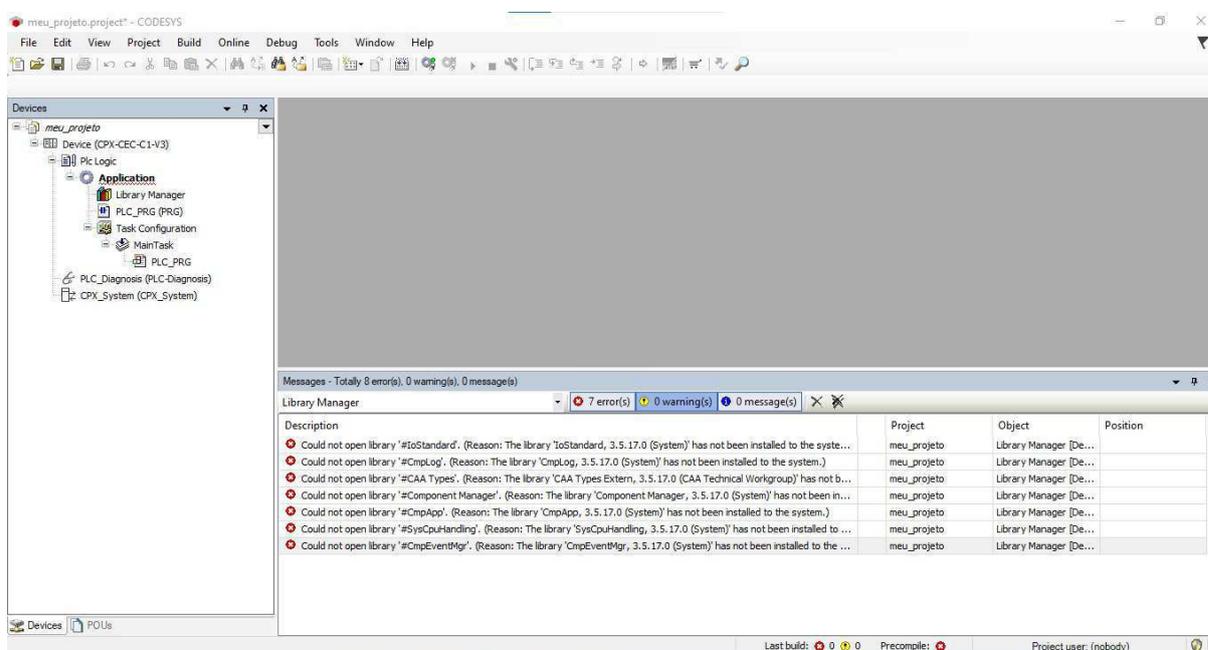
Figura 12 – Configurando o Projeto Padrão



Fonte: autoria própria.

Após a conclusão dessas etapas, o projeto será criado, e o software apresentará a interface mostrada na figura 13.

Figura 13 – Janela Inicial do Projeto

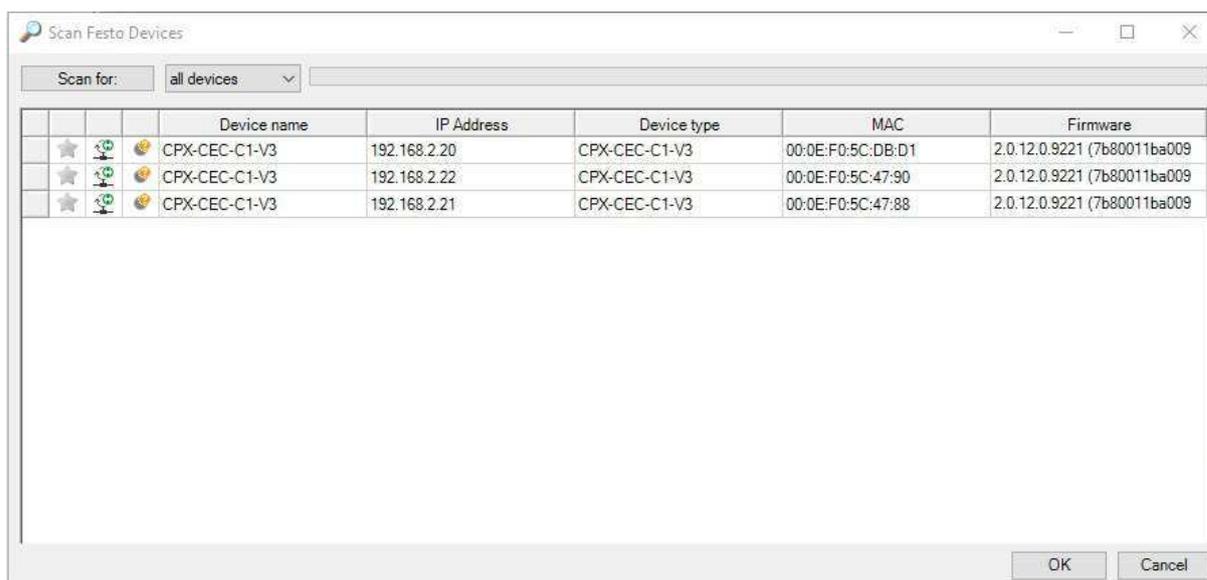


Fonte: autoria própria.

## A.7 Conectando ao CODESYS

Podemos verificar as conexões com os CLPs clicando no ícone da lupa na barra superior à direita (observe que, em algumas versões do CODESYS, essa opção pode não estar disponível). Isso abrirá a janela exibida na figura 14.

Figura 14 – CLPs Conectados

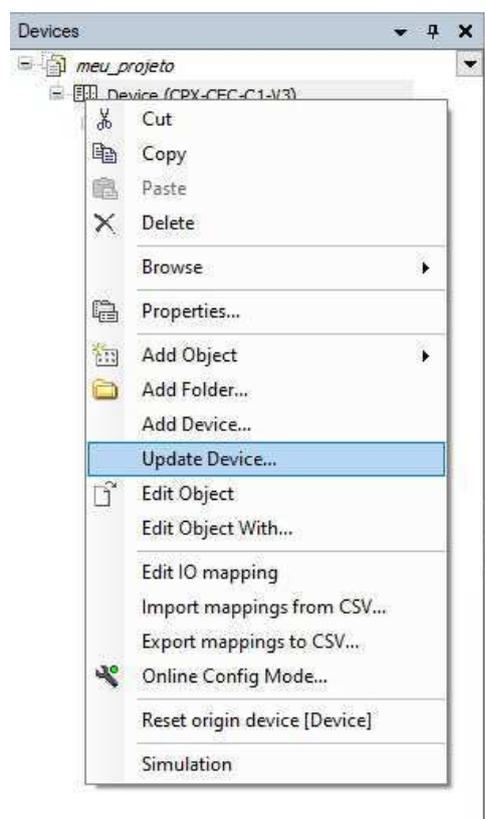


Fonte: autoria própria.

Nessa janela, é possível visualizar todos os CLPs da FESTO conectados à rede, juntamente com detalhes dos dispositivos, como o endereço IP e o endereço MAC. Se nenhum CLP conectado for exibido, verifique a conexão do computador e dos CLPs com a rede.

Antes de começarmos a programar o CLP é importante verificar se a versão do Target selecionada no CODESYS está igual a versão utilizada no CLP, para isso, na área de *devices* clique com botão direito do mouse em **Device (CPX-CEC-C1-V3)** e selecione a opção **Update Device...** como pode ser visto na figura 15

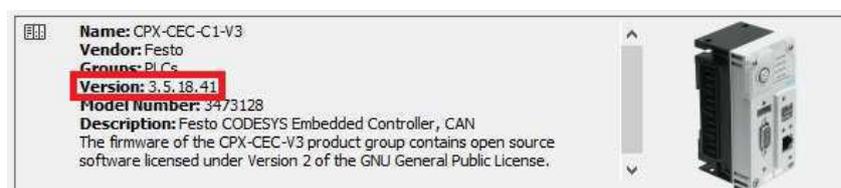
Figura 15 – Acessando a área de Devices



Fonte: autoria própria.

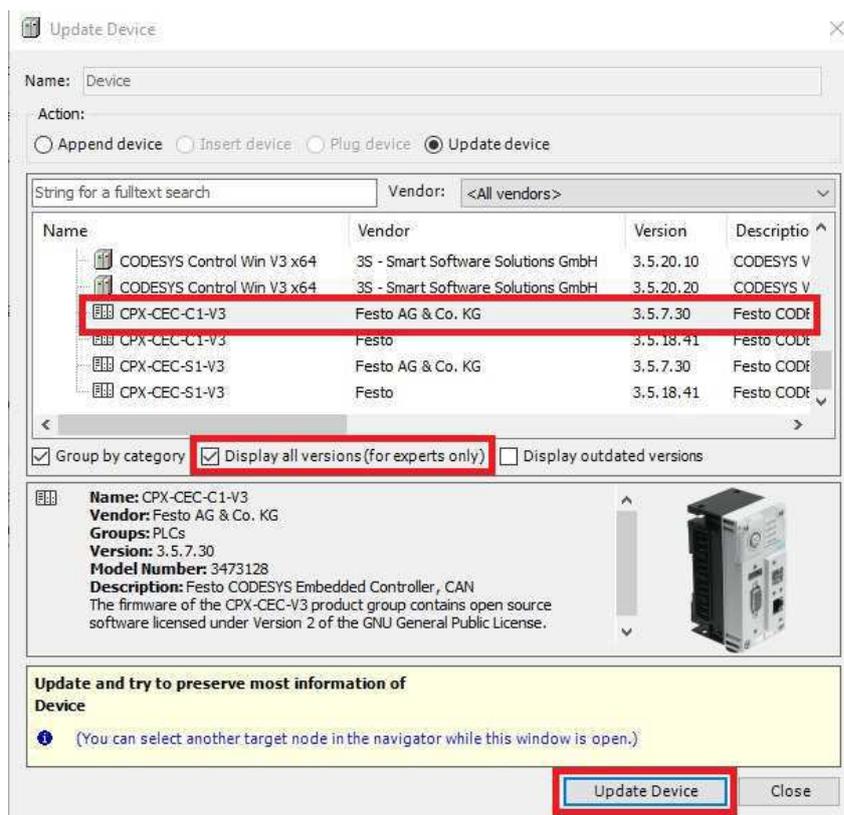
Nessa janela verifique se a versão do *Target* como pode ser visto na figura 16, caso esteja em uma versão diferente da versão 3.5.7.30 marque a opção **Display all versions** e selecione o **CPX-CEC-C1-V3** com a versão **3.5.7.30** como pode ser visto na figura 17 e finalize clicando em **Update Device**.

Figura 16 – Versão do Device



Fonte: autoria própria.

Figura 17 – Update de Drevice



Fonte: autoria própria.

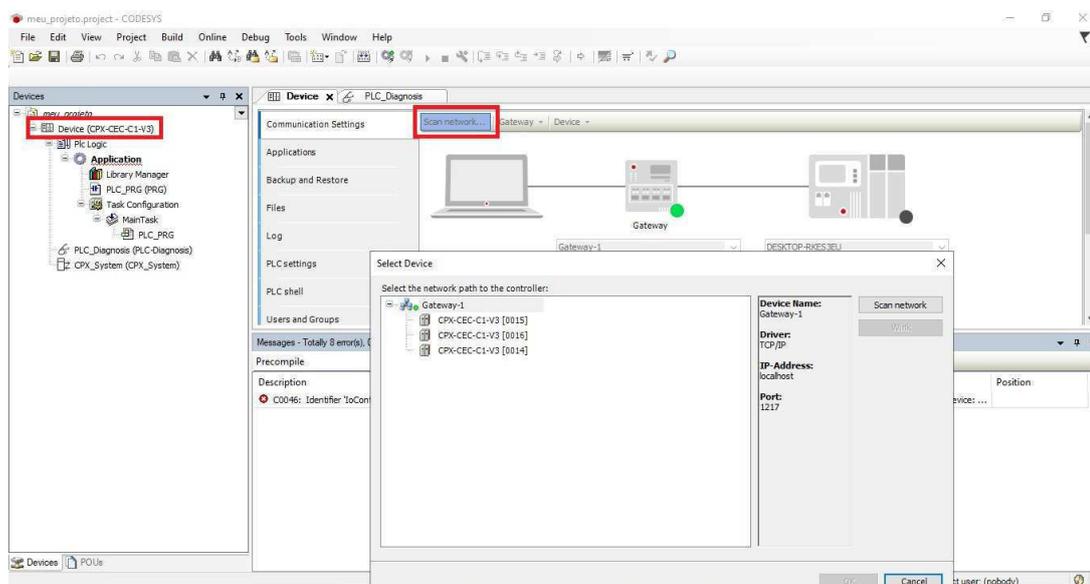
Caso a versão 3.5.7.30 não esteja disponível volte na seção A.4 deste guia e verifique se realizou a instalação correta do *Target*.

Agora, podemos conectar o CODESYS ao CLP da FESTO. Para facilitar o processo, dividiremos esta seção em duas partes: a seção A.7.1 abordará as versões mais recentes do CODESYS, enquanto a seção A.7.2 tratará das versões mais antigas.

### A.7.1 Versões Atuais

Dê um duplo clique em **Device (CPX-CEC-C1-V3)** e, em seguida, clique em **Scan network...** Isso abrirá uma janela onde você poderá escolher o dispositivo a ser conectado, conforme mostrado na figura 18. Nessa janela, é possível observar três dispositivos disponíveis para conexão: **CPX-CEC-C1-V3 [0014]**, **CPX-CEC-C1-V3 [0015]** e **CPX-CEC-C1-V3 [0016]**. Esses dispositivos correspondem, respectivamente, aos CLPs das estações de distribuição, separação e classificação.

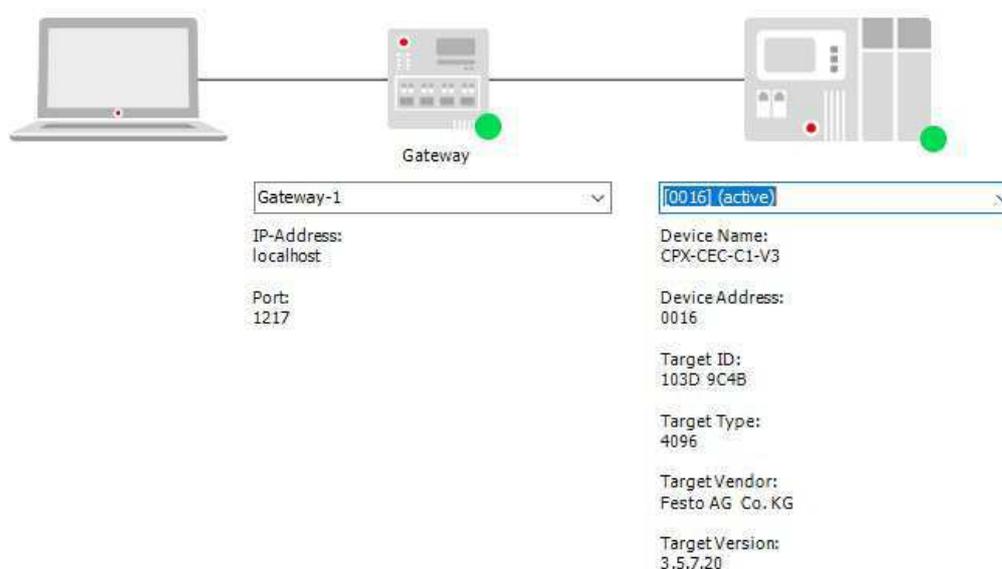
Figura 18 – Conectando o CLP ao Gateway



Fonte: autoria própria.

Selecione o CLP que deseja programar e clique em **OK**, para exemplificação foi escolhido o **CPX-CEC-C1-V3 [0016]** da estação de classificação. Após essa etapa, você poderá verificar no CODESYS as informações exibidas, conforme mostrado na figura 19.

Figura 19 – CLPs Conectados

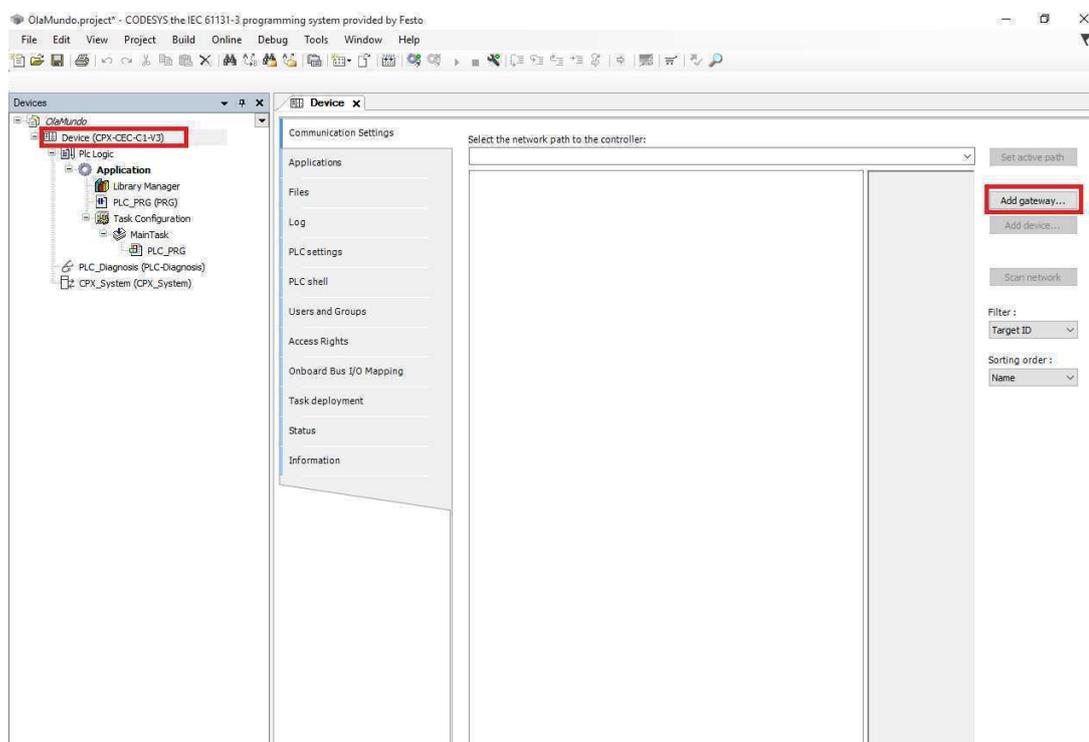


Fonte: autoria própria.

## A.7.2 Versões mais antigas

Dê um duplo clique em **Device (CPX-CEC-C1-V3)**. Em seguida, do lado direito da mesma figura clique em **Add gateway** como pode ser visto na figura 20.

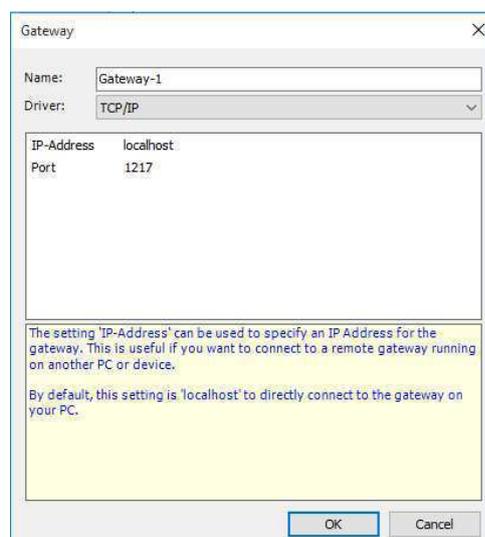
Figura 20 – Criando Gateway



Fonte: Ramos (2019).

A seguinte janela será aberta:

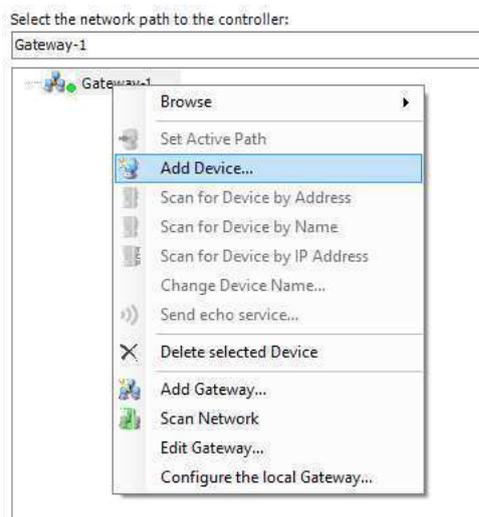
Figura 21 – Configuração do gateway.



Fonte: Ramos (2019).

Configure um nome e protocolo de comunicação para o gateway e clique em **OK**. Feito isso, o próximo passo é adicionar o CLP ao gateway. Para isso, clique com o botão direito sobre o gateway criado e selecione a opção **Add Device**, conforme ilustrado na figura 22.

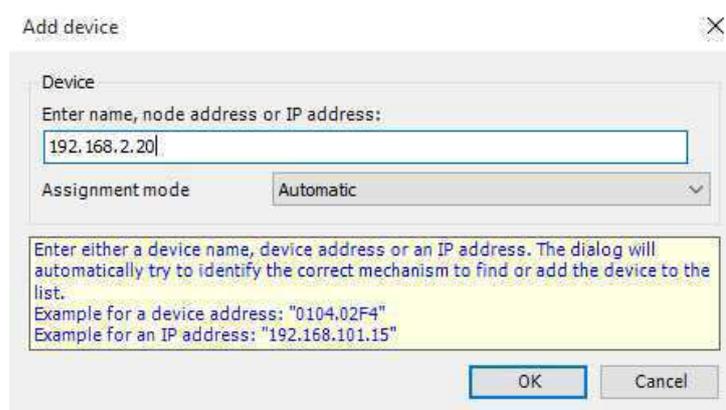
Figura 22 – Adição de um dispositivo ao gateway.



Fonte: Ramos (2019).

Na janela que se abrir, digite o identificador do dispositivo e clique em OK. Se estiver utilizando TCP/IP, basta inserir o endereço IP da conexão do CLP. No LIEC, os IPs dos CLPs que controlam as estações de distribuição, separação e classificação são, respectivamente: 192.168.2.20, 192.168.2.21 e 192.168.2.22, conforme visto na seção A.7 ao clicar no ícone da lupa.

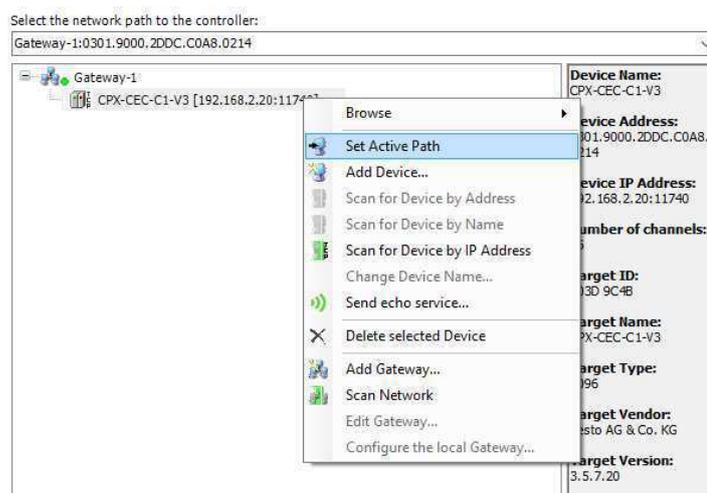
Figura 23 – Configuração da comunicação com o CLP.



Fonte: Ramos (2019).

Para habilitar a comunicação com o CLP, clique com o botão direito sobre o dispositivo adicionado e selecione a opção **Set Active Path**.

Figura 24 – Habilitação da comunicação com o CLP.



Fonte: Ramos (2019).

## A.8 Configuração dos Cartões do CLP

Cada estação do Sistema modular de produção (MPS) produzido pela FESTO adquirido pelo LIEC utiliza um CLP para cada módulo. O CLP é apresentado na Figura 25. Sua estrutura é modular e inclui a **CPU CPX-CEC-C1-V3** e dois cartões de E/S digitais **CPX-AB-8-KL-4POL**, cada um contendo oito entradas e oito saídas, que são utilizados para interface com a plataforma e o painel.

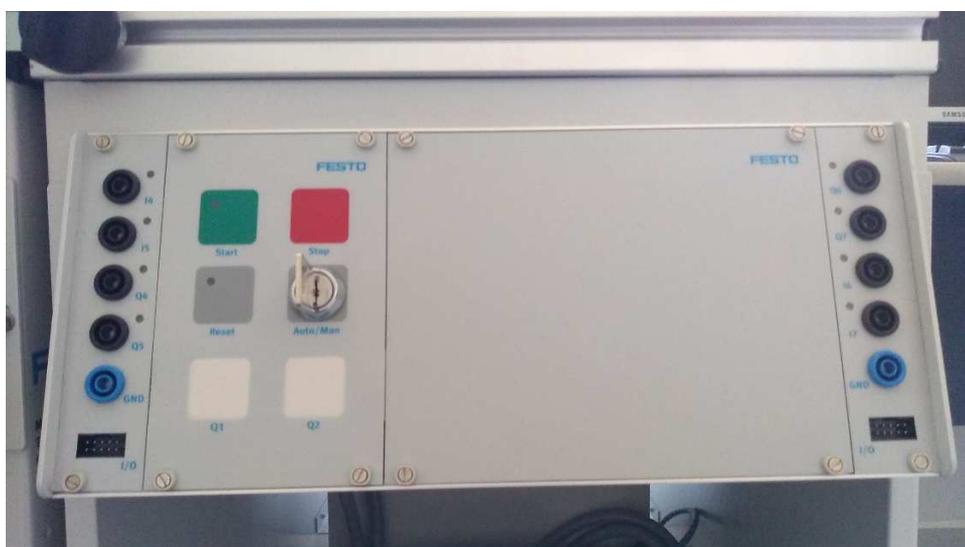
Figura 25 – CLP da FESTO.



Fonte: Ramos (2019).

O primeiro cartão de E/S digitais é responsável por conectar os sensores e atuadores da estação, enquanto o segundo cartão está ligado ao painel de controle, mostrado na figura 26. Esse painel permite a interação entre o operador e o sistema, com botões que enviam comandos para a bancada e LEDs que indicam os estados de operação. Ele é composto por três botões (start, stop e reset), uma chave seletora (auto/man), duas lâmpadas indicadoras (Q1 e Q2), quatro bits de entradas (I4 a I7) e quatro bits de saídas (Q4 a Q7). Esses bits de E/S possibilitam a conexão da estação com dispositivos externos, como por exemplo novos sensores e atuadores, permitindo o desenvolvimento de melhorias nas estações.

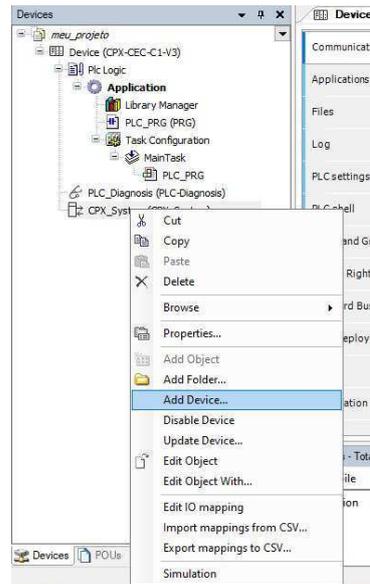
Figura 26 – Painel de uma estação do MPS.



Fonte: Ramos (2019).

Após a configuração da comunicação visto na seção anterior, é necessário configurar os cartões do CLP. Para isso, observando a região destacada no lado esquerdo da tela ilustrada na Figura 27, clique com o botão direito em **CPX\_System**. Em seguida clique em **Add Device....**

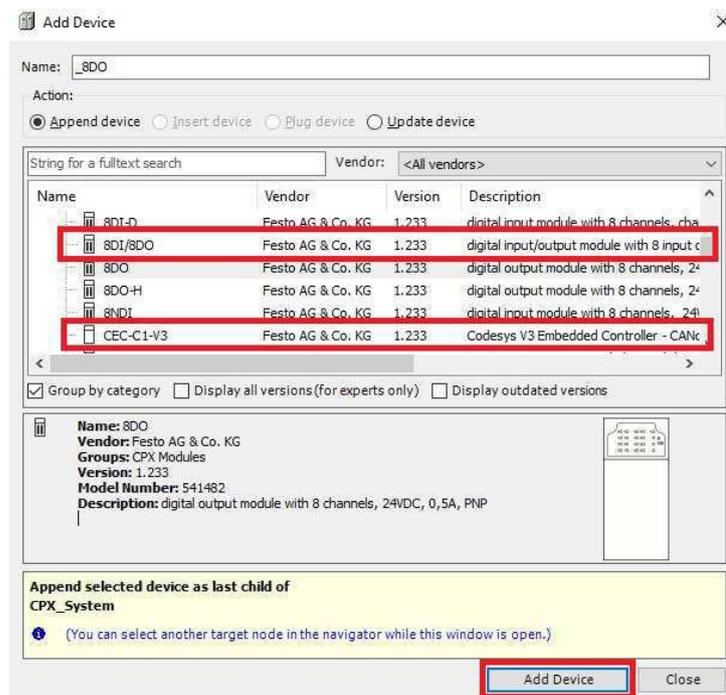
Figura 27 – Acessando a Área de Dispositivos.



Fonte: autoria própria.

Uma janela será aberta para selecionar os dispositivos a serem adicionados. Como mencionado anteriormente, o CLP da estação é composto por uma CPU e dois cartões de E/S, representados por 8DI/8DO. Selecione a CPU **CEC-C1-V3** e clique em **Add Device**. Em seguida, selecione **8DI/8DO** e clique duas vezes em **Add Device**, conforme mostrado nas marcações da figura 28.

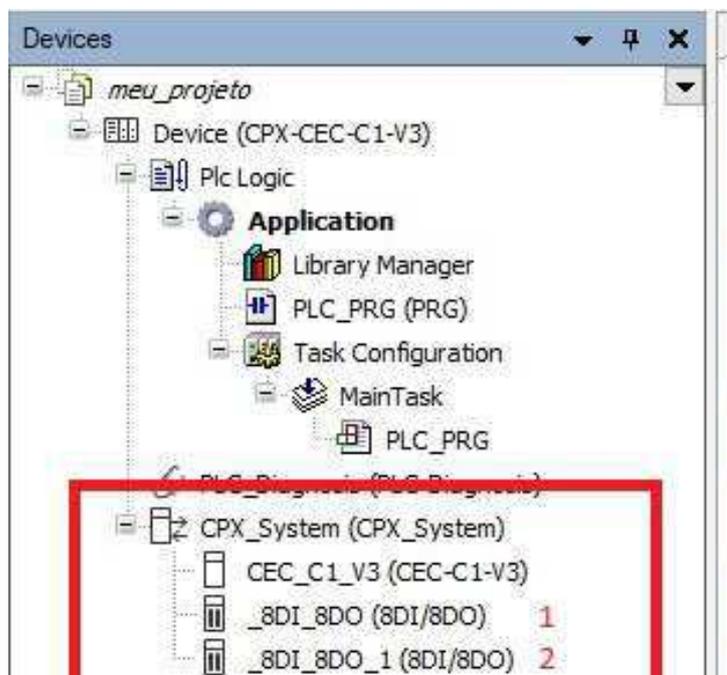
Figura 28 – Adicionando os Dispositivos.



Fonte: autoria própria.

Ao realizar esse procedimento, a CPU e os cartões de E/S do CLP serão exibidos conforme a Figura 29. O primeiro cartão está conectado aos sensores e atuadores da estação, enquanto o segundo está ligado aos botões e LEDs do painel.

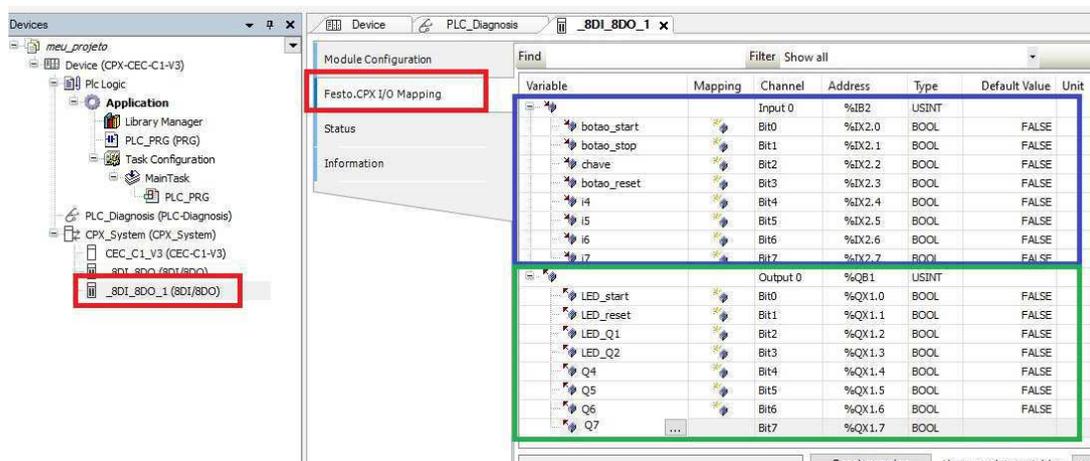
Figura 29 – Visualização dos módulos do CLP adicionados.



Fonte: autoria própria.

Ao dar um duplo clique em um dos cartões de E/S, a tela apresentada na Figura 30 será exibida. Ao clicar em **Festo.CPX I/O Mapping** são exibidas as entradas e saídas do cartão, onde podem ser definidos apelidos para facilitar sua utilização durante a programação, a região destacada em azul contém as entradas e a região destacada em verde contém as saídas

Figura 30 – Visualização dos módulos do CLP adicionados.



Fonte: autoria própria.

Para viabilizar a programação da lógica de funcionamento das estações, é essencial definir as conexões elétricas entre os sensores e atuadores de cada sistema com os bits de entrada e saída dos cartões do CLP. O mapeamento dessas E/S pode ser visualizado em detalhes nas tabelas (1)-(4).

Tabela 1 – Mapa de E/S do painel de controle.

| Bit do Cartão de E/S | Entrada         | Saída              |
|----------------------|-----------------|--------------------|
| 0                    | Botão Start     | LED do botão Start |
| 1                    | Botão Stop (NF) | LED do botão Reset |
| 2                    | Chave Auto/Man  | Lâmpada Q1         |
| 3                    | Botão Reset     | Lâmpada Q2         |
| 4                    | I4              | Q4                 |
| 5                    | I5              | Q5                 |
| 6                    | I6              | Q6                 |
| 7                    | I7              | Q7                 |

Fonte: Ramos (2019).

Tabela 2 – Mapa de E/S da estação de distribuição.

| Bit do Cartão de E/S | Entrada                                 | Saída                         |
|----------------------|---|-------------------------------|
| 0                    | -                                       | Recuo do atuador cilíndrico   |
| 1                    | Posição do atuador cilíndrico em avanço | Ativação do vácuo na ventosa  |
| 2                    | Posição do atuador cilíndrico em recuo  | Pulso de ar na ventosa        |
| 3                    | Vacuostato                              | Retorno do atuador cilíndrico |
| 4                    | Posição do atuador rotativo em recuo    | Avanço do atuador cilíndrico  |
| 5                    | Posição do atuador rotativo em avanço   | -                             |
| 6                    | Presença de peça no armazém (NF)        | -                             |
| 7                    | Receptor IP_FI                          | -                             |

Fonte: Ramos (2019).

Tabela 3 – Mapa de E/S da estação de separação.

| Bit do Cartão de E/S | Entrada                                 | Saída                        |
|----------------------|---|------------------------------|
| 0                    | Presença de peça no início da esteira   | Acionamento da esteira 1     |
| 1                    | Presença de peça no sensor de distância | Acionamento da esteira 2     |
| 2                    | Sensor de distância                     | Avanço do atuador cilíndrico |
| 3                    | Módulo de desvio recuado                | Avanço do módulo de desvio   |
| 4                    | Sensor óptico B5 (NF)                   | -                            |
| 5                    | Senso óptico B6 (NF)                    | -                            |
| 6                    | -                                       | -                            |
| 7                    | Receptor IP_FI                          | Transmissor IP_N_FO          |

Fonte: Ramos (2019).

Tabela 4 – Mapa de E/S da estação de classificação.

| Bit do Cartão de E/S | Entrada                               | Saída                        |
|----------------------|---------------------------------------|------------------------------|
| 0                    | Presença de peça no início da esteira | Acionamento da esteira       |
| 1                    | Peça metálica detectada               | Avanço do módulo de desvio 1 |
| 2                    | Peça vermelha detectada               | Avanço do módulo de desvio 2 |
| 3                    | Fila cheia nos escorregadores         | Avanço do atuador cilíndrico |
| 4                    | Módulo de desvio 1 recuado            | -                            |
| 5                    | Módulo de desvio 1 avançado           | -                            |
| 6                    | Módulo de desvio 2 recuado            | -                            |
| 7                    | Módulo de desvio 2 avançado           | Transmissor IP_N_FO          |

Fonte: Ramos (2019).

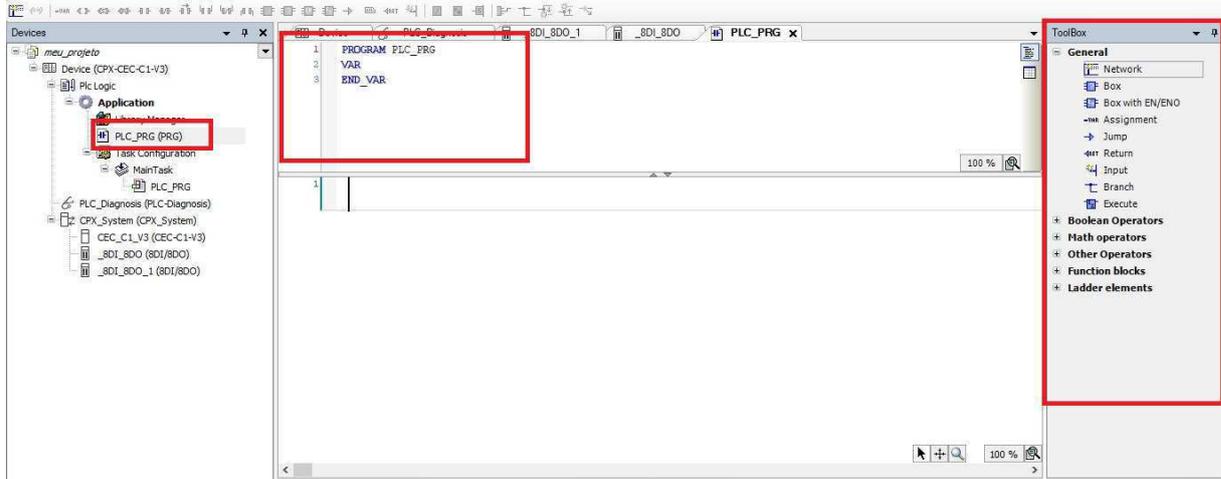
Após o preenchimento correto dos dois cartões de E/S, é possível iniciar a programação das estações.

## A.9 Programação

Para iniciar a programação no CODESYS, primeiro abra o programa principal dando um duplo clique em **PLC\_PRG**. Isso exibirá a tela mostrada na Figura 31,

composta por uma área para definição de variáveis, uma área para criação dos diagramas, e uma caixa de ferramentas à direita com a lista de blocos disponíveis para programação.

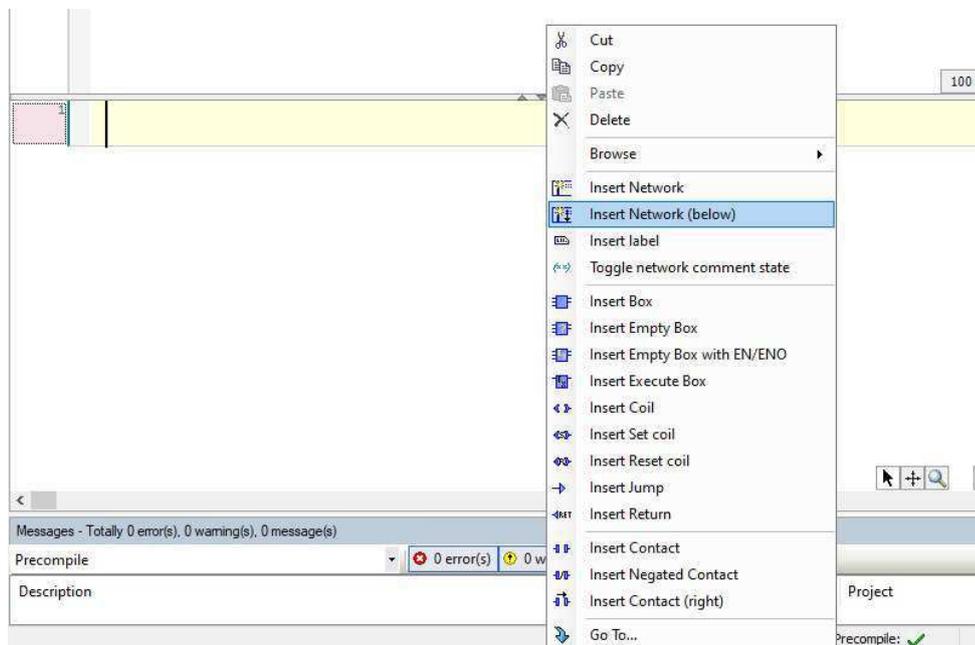
Figura 31 – Tela do programa principal.



Fonte: autoria própria.

Na área de criação dos diagramas Ladder, apresentada na Figura 50, a inserção de linhas é feita clicando com o botão direito e selecionando a opção **Insert Network**.

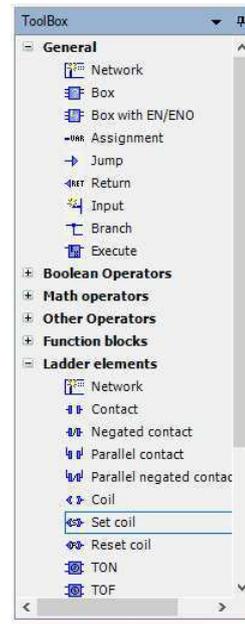
Figura 32 – Inserindo nova linha no diagrama Ladder



Fonte: autoria própria.

A lista de elementos para programação é apresentada na Figura 33. Na aba Ladder elements podem ser inseridos blocos customizados criados utilizando as unidades de organização do programa

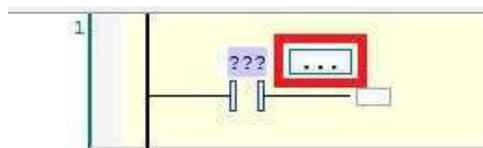
Figura 33 – Blocos disponíveis na linguagem Ladder.



Fonte: autoria própria.

Para adicionar um elemento, arraste-o da lista de blocos para a área de criação do diagrama. Para criar uma nova variável, insira o nome no campo "???". Na janela que aparecer, você poderá definir detalhes como o tipo da variável. Se preferir usar variáveis já existentes, como as dos cartões de E/S do CLP, clique no botão "..." conforme mostrado na Figura 34.

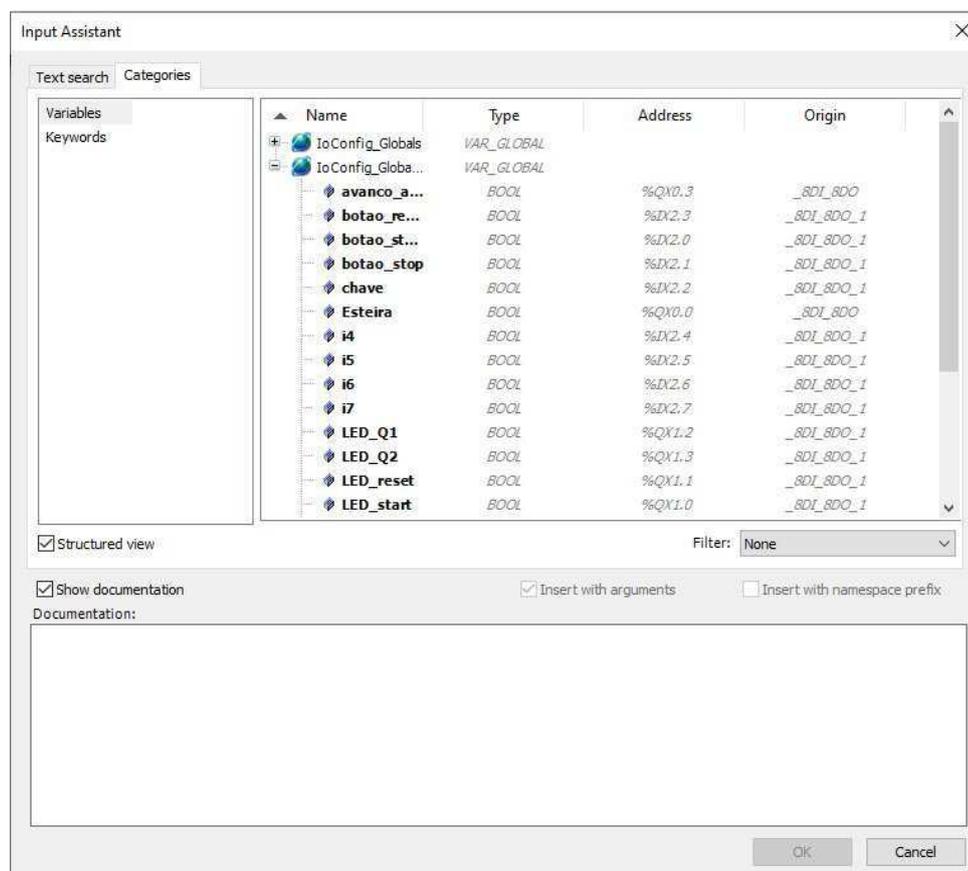
Figura 34 – Inserção de um elemento no diagrama Ladder.



Fonte: autoria própria.

Uma janela semelhante a mostrada na Figura 35 será aberta. Nela é possível visualizar as entradas e saídas dos cartões do CLP que foram mapeadas. Basta selecionar a desejada e clicar em **OK**.

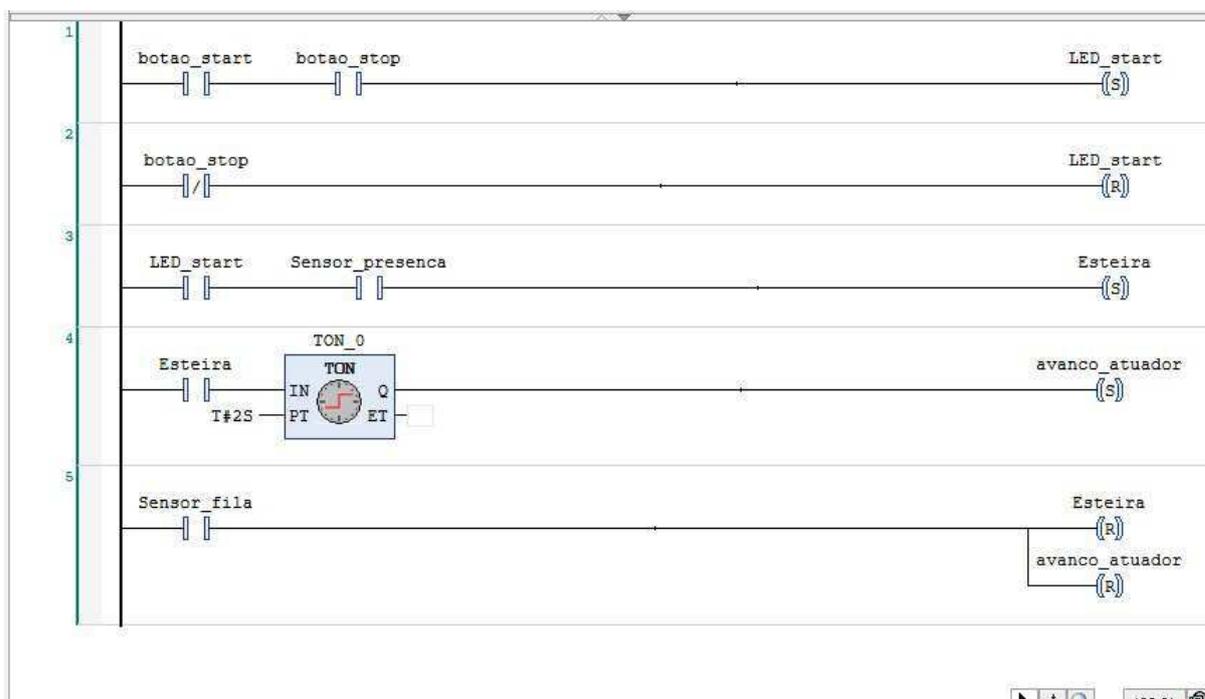
Figura 35 – Inserção de um elemento no diagrama Ladder.



Fonte: autoria própria.

Com esses conceitos básicos aprendidos, já é possível iniciar a programação das estações. Na Figura 36, é mostrado um exemplo de um código básico desenvolvido para a estação de classificação. Esse código verifica se a estação foi acionada pelo botão de ligar e, quando uma peça chega, ativa a esteira e é aberto o pistão permitindo que a peça se desloque até o armazém.

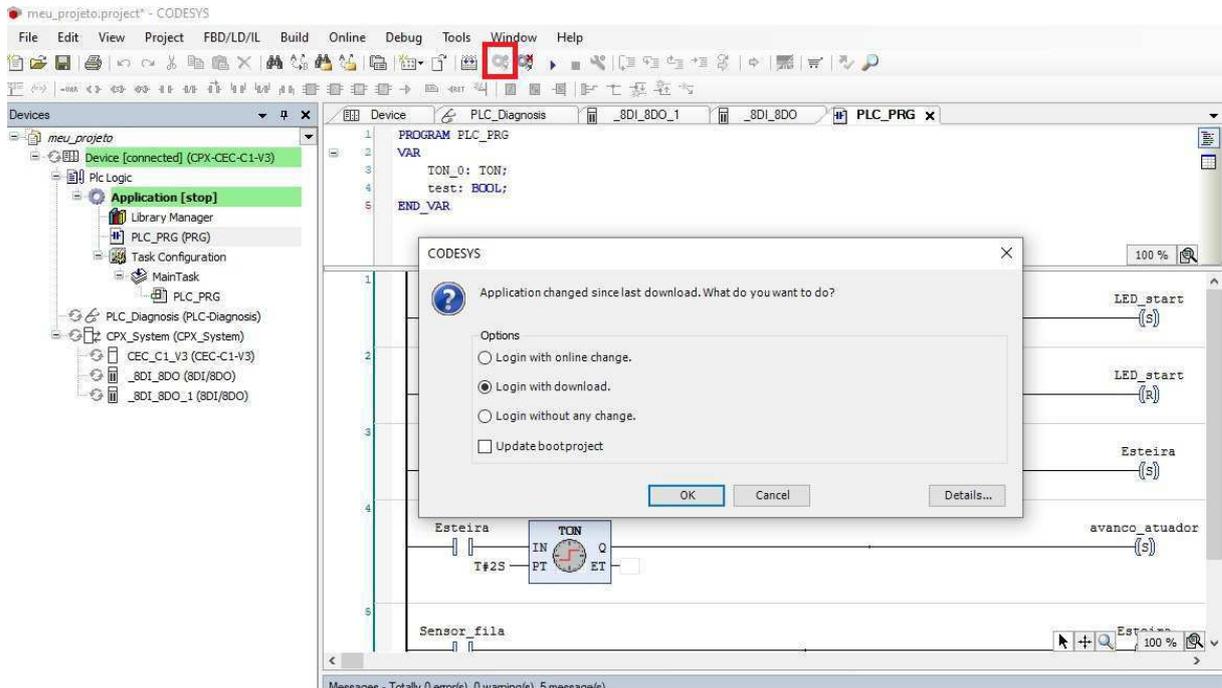
Figura 36 – Exemplo de programação Ladder.



Fonte: autoria própria.

Após finalizar a programação da estação, é necessário fazer o *upload* do código para o CLP. Para isso, clique no ícone de **Login** na barra superior. Em seguida, uma janela será exibida; marque a opção **Login with download** para transferir a programação para o CLP e clique em **OK**. Se desejar que a programação seja salva na memória não volátil do CLP, marque a opção Update boot project. Isso garantirá que o programa permaneça salvo, mesmo em caso de queda de energia. Esse procedimento está ilustrado na Figura 37.

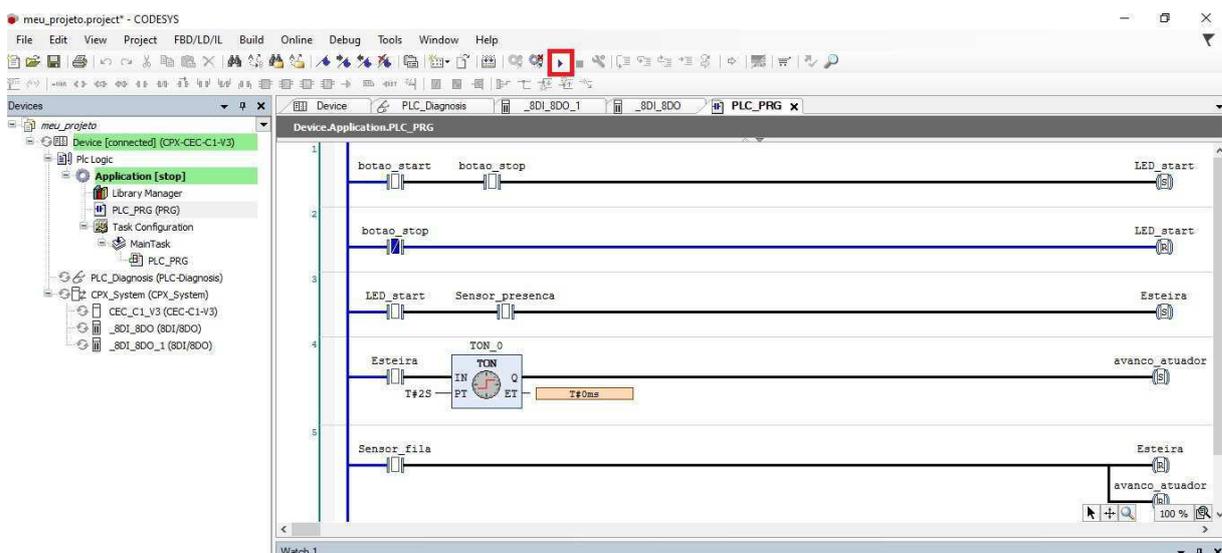
Figura 37 – Upload da Programação.



Fonte: autoria própria.

Para finalizar, clique no ícone de **Start** para que o CLP inicie a execução do programa. A região de programação agora exibirá o código em funcionamento no CLP em tempo real, permitindo que você acompanhe sua execução. Esse processo pode ser visualizado na Figura 38.

Figura 38 – Programação Funcionando.



Fonte: autoria própria.

Concluimos, assim, o guia básico para a programação das estações da FESTO.

Com os conhecimentos adquiridos neste guia, é possível realizar a programação dos três módulos do sistema. Caso seja necessário modificar a programação, clique no botão de "Logout", localizado ao lado do ícone de "Login", para reabrir o ambiente de programação e efetuar as alterações desejadas.