

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO DE JOINVILLE
CURSO DE ENGENHARIA MECATRÔNICA

PABLO VINÍCIUS ZABEL

CONTROLE SUPERVISÓRIO DE SISTEMAS A EVENTOS DISCRETOS: UMA
ABORDAGEM BASEADA EM CLP E SIMULAÇÃO

Joinville
2024

PABLO VINÍCIUS ZABEL

CONTROLE SUPERVISÓRIO DE SISTEMAS A EVENTOS DISCRETOS: UMA
ABORDAGEM BASEADA EM CLP E SIMULAÇÃO

Trabalho apresentado como requisito parcial para obtenção do título de bacharel em Engenharia Mecatrônica, no Curso de Engenharia Mecatrônica, do Centro Tecnológico de Joinville, da Universidade Federal de Santa Catarina.

Orientador: Prof. Dr. Benjamin Grando Moreira

Joinville
2024

RESUMO

Este trabalho de conclusão de curso, intitulado "Avaliação da Síntese e Controle de Sistemas a Eventos Discretos Utilizando CLP", investiga métodos de controle supervísório aplicados a Sistemas a Eventos Discretos (SEDs) utilizando Controladores Lógicos Programáveis (CLPs). Inspirado no estudo de Oliveira (2021), que avaliou o software CIF3 para simulação de SEDs, este projeto expande essa abordagem com a introdução da linguagem ladder, simulações em CLPs e um simulador fabril. Foram avaliadas diferentes ferramentas, incluindo OpenPLC e Factory IO, para selecionar aquelas que melhor atendem aos requisitos de compatibilidade e funcionalidade para simulação de processos industriais. A metodologia envolveu a remodelagem de autômatos e a configuração de ambientes de simulação, seguida pela execução de simulações para validar a eficácia das soluções propostas. Os resultados demonstraram a viabilidade do uso de CLPs e ferramentas associadas para o controle supervísório de SEDs, culminando em recomendações de uso.

Palavras-chave: Sistemas a Eventos Discretos. Controle Supervísório. Controladores Lógicos Programáveis. Automação Industrial. Simulação.

ABSTRACT

This thesis, titled "Evaluation of the Synthesis and Control of Discrete Event Systems Using PLC," investigates supervisory control methods applied to Discrete Event Systems (DES) using Programmable Logic Controllers (PLCs). Inspired by Oliveira (2021) study on the CIF3 software for DES simulation, this project expands the approach by introducing ladder language, simulations in PLCs, and a factory simulator. Various tools, including OpenPLC and Factory IO, were evaluated to select those that best meet compatibility and functionality requirements for industrial process simulation. The methodology involved remodelling automata and configuring simulation environments, followed by simulations to validate the proposed solutions' effectiveness. Results demonstrated the feasibility of using PLCs and associated tools for supervisory control of DES, culminating in usage recommendations.

Keywords: Discrete Event Systems. Supervisory Control. Programmable Logic Controllers. Industrial Automation. Simulation.

AGRADECIMENTOS

Agradeço à minha família pelo apoio emocional, incentivo e motivação.

Ao meu orientador, Prof. Benjamin Grando Moreira, pela atenção, dedicação e paciência.

Aos meus amigos pelo apoio e pelas discussões inspiradoras.

"O sucesso é ir de fracasso em fracasso sem perder o entusiasmo."

Winston Churchill

LISTA DE ILUSTRAÇÕES

Figura 1 – Autômato.	17
Figura 2 – Diagrama da comunicação Modbus.	23
Figura 3 – Diagrama da metodologia.	26
Figura 4 – Opções de exportação nativas do ESCET.	28
Figura 5 – Plantas do sistema de operação da esteira.	33
Figura 6 – Modelo de especificação do sistema.	34
Figura 7 – Autômato supervisor para controle do sistema simulado.	35
Figura 8 – Fluxo da metodologia.	36
Figura 9 – Debugger do OpenPLC.	38
Figura 10 – Condição inicial dos estados.	39
Figura 11 – Status dos eventos não controláveis.	39
Figura 12 – Transições dos eventos não controláveis.	40
Figura 13 – Transições dos eventos controláveis.	42
Figura 14 – Escrita na saída.	43
Figura 15 – Escrita no contador.	43
Figura 16 – Configuração do dispositivo no OpenPLC Runtime.	44
Figura 17 – Painel do OpenPLC Runtime.	45
Figura 18 – Simulação do sistema no CIF3.	46
Figura 19 – Etapas para realização da simulação no CIF3.	46
Figura 20 – Etapas para realização da simulação no Factory IO.	47
Figura 21 – Simulação do sistema no Factory IO.	48
Figura 22 – Partes individuais da simulação.	49

LISTA DE TABELAS

Tabela 1 – Comandos ladder abordados pela IEC 61131-3.	19
Tabela 2 – Tabela de Variáveis	37

LISTA DE SIGLAS

CIF	Compositional Interchange Format
CLP	Controlador Lógico Programável
CPU	Unidade central de processamento
ESCET	Eclipse Supervisory Control Engineering Toolkit
FBD	Function Block Diagram
IA	Inteligência artificial
IL	Instruction List
IoT	Internet das coisas
LD	Ladder Diagram
MBAP	Modbus Application Protocol
MBE	Model-Based Engineering
MDE	Model-Driven Engineering
SCADA	Supervisory Control and Data Acquisition
SED	Sistema a eventos discretos
SFC	Sequential Function Chart
ST	Structured Text
TCC	Trabalho de Conclusão de Curso
TCP	Protocolo de Controle de Transmissão
TCS	Teoria de Controle Supervisório
XML	Extensible Markup Language

SUMÁRIO

1	INTRODUÇÃO	11
1.1	Objetivo Geral	12
1.2	Objetivos Específicos	12
1.3	Estrutura do texto	12
2	FUNDAMENTAÇÃO TEÓRICA	13
2.1	Sistemas a eventos discretos	13
2.1.1	Modelagem e Análise de Sistemas a Eventos Discretos	14
2.1.2	Aplicações dos Sistemas a Eventos Discretos	14
2.1.3	Importância na Indústria Moderna	15
2.1.4	Desafios e Perspectivas	15
2.2	Teoria de Controle Supervisório	16
2.3	Controladores Lógicos Programáveis	17
2.4	Linguagem ladder	19
2.5	Ferramentas Utilizadas	19
2.5.1	DESTool	20
2.5.2	FlexFact	21
2.5.3	Compositional Interchange Format 3	21
2.5.4	OpenPLC	22
2.5.5	Factory IO	22
2.5.6	Comunicação Modbus TCP	23
2.5.6.1	Integração entre OpenPLC e Factory IO via Modbus TCP	23
3	METODOLOGIA	25
3.1	Seleção de ferramentas	26
3.2	Procedimentos Metodológicos	28
3.2.1	Transcrição do autômato supervisor para CLP	28
3.2.2	Configuração do ambiente de simulação	29
3.2.3	Execução das simulações	29
3.3	Análise dos resultados	30
3.4	Limitações e soluções	30
4	SIMULAÇÃO DESENVOLVIDA	32
4.1	Plantas e especificações	32
4.2	Transcrição em ladder	35
4.2.1	Instruções ladder para representar o supervisor	38
4.3	Comunicação Modbus	44

4.4	Simulação	45
4.5	Dificuldades encontradas	49
5	CONCLUSÕES	51
	REFERÊNCIAS	53

1 INTRODUÇÃO

A automação industrial tem se desenvolvido com a integração de tecnologias que aprimoram a eficiência e o controle de processos produtivos. No centro dessa evolução estão os Sistemas a Eventos Discretos (SEDs), que se destacam pelo gerenciamento de eventos e transições de estado em ambientes industriais. Esses sistemas possibilitam a automação precisa de processos que dependem de eventos específicos, como mudanças de estado ou ações de controle, sendo aplicáveis em situações que exigem alta confiabilidade e respostas rápidas (CASSANDRAS; LAFORTUNE, 1999).

Neste contexto, o uso de Controladores Lógicos Programáveis (CLPs) é adotado, oferecendo uma plataforma para implementar soluções de controle. Juntamente com a linguagem ladder, utilizada na programação de CLPs, que é conhecida por sua representação gráfica intuitiva, facilitando a programação e o diagnóstico, especialmente em aplicações industriais complexas (LEWIS, 1998).

O objetivo deste trabalho é desenvolver uma metodologia para a simulação de controle supervisorio de SEDs utilizando CLPs, com foco na aplicação prática em ambientes de simulação industrial. Inspirado no estudo de Oliveira (2021), que explorou o software CIF3 para simulação de SEDs, esta pesquisa amplia essa abordagem ao incorporar o uso da linguagem ladder e de ferramentas de simulação como OpenPLC e Factory IO. Essas ferramentas foram escolhidas por sua capacidade de simular processos industriais de forma realista, permitindo que os usuários visualizem e interajam com os sistemas em um ambiente seguro e controlado.

O OpenPLC, um controlador lógico programável de código aberto, permite o desenvolvimento e a implementação de lógicas de controle em um ambiente simulado que replica condições industriais reais (ALSABBAGH; KIM; LANGENDÖRFER, 2023). A integração com o Factory IO possibilita a criação de simulações de fábricas virtuais onde sistemas automatizados podem ser testados e validados antes da implementação física, permitindo a análise e otimização das lógicas de controle (SALAH et al., 2023). Esta abordagem visa validar a eficácia dessas ferramentas na simulação de processos industriais para sua adoção em ambientes educacionais e industriais.

Segundo Felder e Brent (2004), a simulação é uma ferramenta bastante utilizada na educação em engenharia e automação industrial, permitindo que estudantes e profissionais em formação experimentem cenários práticos sem os riscos associados ao manuseio de equipamentos reais. Com o uso de ferramentas de

simulação, os alunos podem interagir de maneira segura, o que é essencial para o aprendizado. A possibilidade de visualizar operações complexas e de realizar testes em um ambiente controlado torna o aprendizado mais envolvente e eficaz, contribuindo para a retenção de conhecimento (BANKS et al., 2005).

A relevância deste estudo reside na crescente demanda por soluções de automação que sejam adaptáveis e de fácil implementação, respondendo às necessidades de um setor em constante transformação. Ao explorar o uso de CLPs e SEDs, este trabalho contribui para o desenvolvimento de novas abordagens na automação industrial, promovendo a eficiência e a inovação.

1.1 Objetivo Geral

Elaborar uma abordagem didática de simulação de controle supervisório de Sistemas a Eventos Discretos.

1.2 Objetivos Específicos

Para atingir o objetivo principal, são propostos os seguintes objetivos específicos:

- a. Avaliar recursos para execução do controle supervisório de sistemas a eventos discretos;
- b. Aplicar em simulação o controle supervisório;
- c. Elaborar recomendações sobre o processo.

1.3 Estrutura do texto

A estrutura deste documento é organizada de forma a proporcionar uma visão clara e detalhada do desenvolvimento do projeto. Primeiramente, é apresentada uma fundamentação teórica sobre SEDs e TCS, seguida pela descrição das ferramentas utilizadas. A metodologia adotada, incluindo a transcrição de autômatos na linguagem ladder e a configuração dos ambientes de simulação, é então detalhada. Finalmente, os resultados obtidos são discutidos, abordando as dificuldades encontradas e as soluções implementadas, culminando em uma série de conclusões e recomendações para futuras pesquisas e aplicações práticas.

2 FUNDAMENTAÇÃO TEÓRICA

A simulação é uma ferramenta aplicada no ensino de engenharia e automação industrial, permitindo a compreensão prática de conceitos. Segundo Felder e Brent (2004), ela facilita a aplicação de teorias em um ambiente controlado, onde é possível observar o comportamento de sistemas e como os conceitos teóricos se traduzem em operações .

O uso de simulações evita riscos associados ao manuseio de equipamentos reais, além de reduzir os custos com materiais e equipamentos. Os alunos podem experimentar e cometer erros sem consequências, o que é importante para o aprendizado (BANKS et al., 2005).

Ferramentas de simulação tornam o aprendizado mais interativo, transformando os alunos em participantes ativos no processo. A interação com sistemas simulados promove envolvimento e retenção do conhecimento (PRENSKY, 2001).

Simulações permitem a reprodução de cenários impraticáveis em ambientes reais, como falhas em sistemas industriais, para que os alunos aprendam a diagnosticar e corrigir problemas sem risco de danos ao equipamento (ALESSI; TROLLIP, 2001).

A prática com simulações auxilia os estudantes a desenvolver habilidades técnicas, operando e controlando sistemas, programando controladores lógicos programáveis (CLPs) e entendendo a dinâmica dos processos (SIEMENS, 2005).

As ferramentas de simulação podem ser ajustadas para diferentes níveis de complexidade, atendendo a diversos perfis de alunos, desde iniciantes até profissionais em formação continuada.

2.1 Sistemas a eventos discretos

Os sistemas a eventos discretos (SED) são uma classe especial de sistemas de controle e automação que respondem a eventos específicos que ocorrem em momentos determinados, em vez de processos contínuos que ocorrem ao longo do tempo. Segundo Cassandras e Lafortune (1999), esses sistemas são utilizados em uma ampla variedade de aplicações industriais e tecnológicas, desde o controle de processos de fabricação até a automação de sistemas de segurança e logística.

SEDs podem ser modelados usando teoria de autômatos, que descrevem formalmente o comportamento do sistema em termos de estados, eventos e transições entre estados. De acordo com Ramadge e Wonham (1989), os autômatos são uma

ferramenta poderosa para modelagem e análise de SEDs, permitindo aos engenheiros entender melhor o comportamento dinâmico do sistema e projetar estratégias de controle eficazes. Um dos modelos de autômatos mais comuns é o autômato finito determinístico (AFD), que possui um conjunto finito de estados e transições determinísticas entre esses estados em resposta a eventos discretos.

2.1.1 Modelagem e Análise de Sistemas a Eventos Discretos

A modelagem e análise de SEDs são processos utilizados para entender o comportamento dinâmico desses sistemas e projetar estratégias de controle eficazes. Uma abordagem comum para modelagem de SEDs é identificar os eventos discretos que podem ocorrer no sistema, bem como os estados possíveis em que o sistema pode se encontrar. Segundo Baier e Katoen (2008), com base nessa descrição, os engenheiros podem construir modelos de autômatos que representam o comportamento dinâmico do sistema em resposta a eventos discretos.

A análise de SEDs envolve a verificação de propriedades do sistema, como alcançabilidade, vivacidade e previsibilidade. Segundo Lafortune (2008), a alcançabilidade refere-se à capacidade do sistema de atingir determinados estados a partir de um estado inicial dado, enquanto a vivacidade refere-se à capacidade do sistema de continuar respondendo a eventos discretos indefinidamente. A previsibilidade refere-se à capacidade do sistema de garantir que determinados eventos ocorram dentro de um intervalo de tempo especificado.

Ferramentas matemáticas robustas e precisas são exigidas para a modelagem e o controle de SEDs. Nesse contexto, a Teoria de Controle Supervisório (TCS) surge como um paradigma. A TCS se baseia na criação de um sistema supervisor capaz de monitorar o comportamento do SED e aplicar as ações de controle necessárias para garantir o cumprimento das especificações predefinidas (WONHAM, 2011). A engenharia de sistemas moderna, oferece ferramentas computacionais sofisticadas para auxiliar na modelagem e simulação de SEDs. O Model-Based Engineering (MBE) ou Model-Driven Engineering (MDE) se destaca como uma metodologia que utiliza modelos para representar os diferentes aspectos do sistema, auxiliando no processo de desenvolvimento e análise (BRUCKNER; DORNER; ERNST, 2009).

2.1.2 Aplicações dos Sistemas a Eventos Discretos

SEDs encontram uma ampla gama de aplicações em diversos setores industriais e tecnológicos. Segundo Gomes e Veloso (2019), em sistemas de controle de processos industriais, os SEDs são usados para monitorar e controlar máquinas, linhas de produção e sistemas de logística. Eles são responsáveis por coordenar o movimento de equipamentos, acionar dispositivos de manipulação e garantir a sincronização

entre diferentes etapas do processo. Além disso, os SEDs desempenham funções em sistemas de segurança e controle de acesso.

Conforme mencionado por Cassandras e Lafortune (1999), sistemas de alarme em residências e edifícios comerciais podem utilizar SEDs para detectar intrusões, acionar sirenes e notificar as autoridades relevantes. Da mesma forma, sistemas de controle de acesso baseados em cartões ou biometria empregam SEDs para autorizar ou negar o acesso a áreas restritas com base em eventos discretos, como a apresentação de um cartão de acesso válido ou uma correspondência biométrica positiva.

2.1.3 Importância na Indústria Moderna

Segundo Bemporad, Giua e Rubagotti (1999), esses sistemas desempenham funções na automação de processos, aumentando a eficiência operacional, melhorando a qualidade do produto e reduzindo os custos de produção. Ao responder de forma rápida e precisa a eventos específicos, os SEDs ajudam a minimizar o tempo de inatividade, otimizar o uso de recursos e garantir a conformidade com padrões de qualidade e segurança. Além disso, os SEDs são utilizados na implementação de sistemas de controle supervisão. Segundo Ramadge e Wonham (1989), nesses sistemas, um supervisor monitora e controla as operações de um sistema distribuído composto por múltiplos dispositivos e subsistemas. Por meio do controle supervisão, os SEDs garantem que o sistema global opere de acordo com as especificações e requisitos predefinidos, adaptando-se dinamicamente a mudanças nas condições do ambiente ou nas demandas operacionais.

2.1.4 Desafios e Perspectivas

Embora os SED ofereçam benefícios na automação e controle de processos, eles também enfrentam desafios. Conforme observado por Baier e Katoen (2008), um dos principais desafios é lidar com a complexidade crescente dos sistemas industriais modernos, que podem envolver milhares de dispositivos e subsistemas interconectados. Além disso, a integração de SEDs com outras tecnologias emergentes, como Internet das Coisas (IoT) e inteligência artificial (IA), apresenta desafios adicionais em termos de interoperabilidade e segurança. No entanto, apesar dos desafios, os SED continuam a desempenhar um papel na indústria moderna. Segundo Lafortune (2008), com o avanço da tecnologia e o desenvolvimento de novas técnicas de modelagem e controle, espera-se que os SEDs se tornem ainda mais poderosos e versáteis, impulsionando a automação e a eficiência em uma ampla variedade de aplicações industriais e tecnológicas.

2.2 Teoria de Controle Supervisório

A Teoria de Controle Supervisório foi introduzida por Ramadge e Wonham (1989) como uma abordagem formal para controlar SEDs. A TCS visa desenvolver controladores que garantam que um SED opere de acordo com especificações desejadas, prevenindo comportamentos indesejáveis e assegurando a eficiência do sistema.

Segundo Ramadge e Wonham (1989) p. 206, "A TCS fornece um framework para o projeto de controladores que restringem o comportamento de um sistema a eventos discretos de modo que satisfaça um conjunto de especificações desejadas". A TCS utiliza modelos formais, como autômatos, para representar tanto o sistema a ser controlado (planta) quanto as especificações de controle (supervisor). A planta é modelada como um autômato que descreve todos os possíveis comportamentos do sistema, enquanto o supervisor desabilita os eventos controláveis para garantir que somente os comportamentos desejados ocorram.

Para exemplificar, considere um sistema de controle de acesso a uma sala segura. A planta pode ser modelada como um autômato com estados representando as portas abertas e fechadas, e eventos como "solicitação de entrada" e "entrada concedida". O supervisor atua monitorando as solicitações e concedendo ou negando acesso com base em regras de segurança:

Estados (Q)	Eventos (Σ)
Fechado	SolicitacaoEntrada, EntradaConcedida, FecharPorta
Aberto	

A função de transição δ é definida como:

$$\delta(\text{Fechado}, \text{SolicitacaoEntrada}) = \text{Aberto}$$

$$\delta(\text{Aberto}, \text{EntradaConcedida}) = \text{Fechado}$$

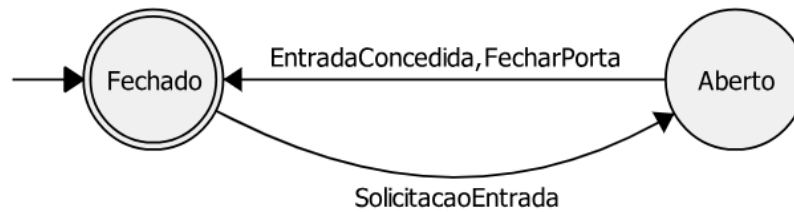
$$\delta(\text{Aberto}, \text{FecharPorta}) = \text{Fechado}$$

O estado inicial q_0 é Fechado. O supervisor atua para garantir que apenas acessos autorizados ocorram:

- Se uma SolicitacaoEntrada é recebida e as condições de segurança são atendidas, o estado transita para Aberto.
- Uma EntradaConcedida faz o estado retornar para Fechado.
- Ação de FecharPorta mantém o estado em Fechado.

Na figura 1 é possível observar o autômato do supervisor.

Figura 1 – Autômato.



Fonte: Autor (2024).

Neste cenário, o supervisor monitora as solicitações de entrada e garante que a transição para o estado Aberto só ocorra se todas as condições de segurança forem satisfeitas. Isso impede acessos não autorizados e assegura a operação segura do sistema. A função de transição δ é utilizada pelo supervisor para verificar as condições de cada evento e determinar se a transição de estado é permitida.

No contexto de sistemas supervisórios segundo a teoria de Ramadge e Wonham (1989), eventos controláveis são aqueles que o supervisor pode influenciar diretamente, como ações de operadores ou comandos do sistema, que normalmente são representados por setas ou arcos cortados, diferenciando-se dos eventos não controláveis. Os Eventos não controláveis ocorrem independentemente da intervenção do supervisor e incluem fatores externos como falhas e condições ambientais. A distinção entre esses dois tipos de eventos permite que o supervisor adapte sua estratégia para acomodar eventos não controláveis enquanto otimiza a resposta aos eventos controláveis. Para mais detalhes sobre a TCS sugere-se consultar o capítulo 2.3 do trabalho de Oliveira (2021).

A TCS proporciona uma abordagem sistemática e formal para o desenvolvimento de controladores robustos para SEDs. Isso permite que sistemas complexos sejam gerenciados de forma eficiente e segura, garantindo que os requisitos de operação sejam atendidos e que os comportamentos indesejáveis sejam evitados.

2.3 Controladores Lógicos Programáveis

Bolton (2015) diz que os Controladores Lógicos Programáveis (CLPs) são dispositivos eletrônicos utilizados na automação industrial para controlar máquinas e processos. Introduzidos nos anos 1960, os CLPs substituíram sistemas de controle baseados em relés e temporizadores por soluções programáveis .

Segundo a norma IEC 61131-3, os CLPs são "sistemas eletrônicos digitais que utilizam memórias programáveis para o armazenamento interno de instruções de usuário específicas para a implementação de funções específicas como lógica, sequenciamento, temporização, contagem e operações aritméticas para controlar, através de módulos de entrada/saída digitais e analógicos, vários tipos de máquinas

e processos"(INTERNATIONAL ELECTROTECHNICAL COMMISSION, 2003). Essa definição destaca a flexibilidade dos CLPs em lidar com diferentes tipos de processos e aplicações industriais.

Um CLP típico é composto por uma unidade central de processamento (CPU), módulos de entrada e saída (I/O), uma fonte de alimentação e interfaces de comunicação (BOLTON, 2015). A CPU executa o programa armazenado na memória, processando sinais recebidos dos dispositivos de entrada e gerando sinais de controle para os dispositivos de saída. Essa arquitetura modular permite que os CLPs sejam configurados e expandidos para atender às necessidades específicas de cada aplicação (BRYAN; BRYAN, 1997).

Os CLPs são usados em diversos setores industriais, como manufatura, petroquímica, automotiva e alimentícia, devido à sua robustez e capacidade de operar em ambientes adversos. Eles são utilizados para o controle de processos contínuos e discretos, como linhas de produção, sistemas de transporte, máquinas de embalagem e processos de mistura (BOLTON, 2015).

De acordo com John e Tiegelkamp (2010), algumas das vantagens dos CLPs incluem a flexibilidade de programação, a facilidade de manutenção e diagnóstico, e a capacidade de integração com uma variedade de dispositivos e sistemas. O avanço das tecnologias digitais tem tornado os CLPs mais poderosos e versáteis. A inclusão de recursos de rede e comunicação, junto com a integração com plataformas de IoT, posiciona os CLPs como componentes centrais na Indústria 4.0, aumentando a conectividade e a inteligência dos processos industriais (BOLTON, 2015).

A comunicação entre Controladores Lógicos Programáveis (CLPs) é realizada por meio de protocolos como Modbus, Profibus e Ethernet/IP, que facilitam a troca de dados entre dispositivos e sistemas industriais (LEWIS, 1998). De acordo com John e Tiegelkamp (2010), esses protocolos permitem a integração de equipamentos variados em uma linha de produção, possibilitando que cada dispositivo envie e receba sinais de controle. Além disso, a comunicação em redes industriais suporta o monitoramento e o controle em tempo real dos processos, permitindo ajustes automáticos e precisos conforme as mudanças nas condições operacionais (INTERNATIONAL ELECTROTECHNICAL COMMISSION, 2003).

A norma IEC 61131-3 define cinco linguagens de programação para CLPs: Ladder Diagram (LD), Function Block Diagram (FBD), Structured Text (ST), Instruction List (IL) e Sequential Function Chart (SFC) (INTERNATIONAL ELECTROTECHNICAL COMMISSION, 2003). Segundo Lewis (1998) a linguagem ladder, também conhecida como lógica de contatos, é a mais utilizada devido à sua semelhança com os diagramas elétricos tradicionais, facilitando a transição de técnicos e engenheiros de sistemas baseados em relés para sistemas programáveis.

2.4 Linguagem ladder

A linguagem ladder, também conhecida como lógica de contatos, é uma linguagem de programação gráfica amplamente utilizada em Controladores Lógicos Programáveis (CLPs) devido à sua semelhança com os esquemas de circuitos elétricos de relés. Desenvolvida inicialmente para facilitar a transição de técnicos e engenheiros que trabalhavam com relés eletromecânicos para sistemas programáveis, a linguagem ladder representa a lógica de controle por meio de "degraus" que conectam contatos e bobinas (LEWIS, 1998). Cada degrau na linguagem ladder simula uma operação lógica que pode ser executada pelo CLP, tornando-a intuitiva para profissionais com formação em eletricidade e automação industrial. De acordo com a norma IEC 61131-3 citada na seção 2.3, a linguagem ladder é uma das cinco linguagens padrão para programação de CLPs, sendo amplamente adotada devido à sua clareza visual e facilidade de diagnóstico e manutenção (INTERNATIONAL ELECTROTECHNICAL COMMISSION, 2003). Além disso, a flexibilidade da linguagem ladder permite a integração de operações como temporização, contagem e aritmética, aumentando sua utilidade em aplicações industriais (BOLTON, 2015). Na Tabela 1 são apresentadas as funções básicas da linguagem ladder abordadas pela IEC 61131-3.

Tabela 1 – Comandos ladder abordados pela IEC 61131-3.

Símbolo	Nome	Descrição
- -	Contato NA	Contato normalmente aberto
- / -	Contato NF	Contato normalmente fechado
- P -	Contato BS	Contato borda de subida
- N -	Contato BD	Contato borda de descida
-(S)-	Liga bobina	É ligada quando a linha está energizada e permanece ligada até que seja desligada em outra parte do código.
-(R)-	Desliga bobina	É desligada quando a linha está energizada e permanece desligada até que seja ligada em outra parte do código.
-()-	Energiza bobina	Permanece ligada enquanto a linha está energizada.

Fonte: Autor (2024).

2.5 Ferramentas Utilizadas

Nesta seção, são discutidas as ferramentas avaliadas ao longo desta pesquisa. Cada ferramenta foi considerada por suas capacidades e adequação ao contexto da automação de sistemas a eventos discretos. Embora nem todas tenham sido implementadas na versão final, a análise das mesmas oferece uma visão sobre as

opções disponíveis e suas potencialidades no desenvolvimento de soluções. A seguir, têm-se as ferramentas destacando suas características principais e justificando sua relevância para este estudo.

2.5.1 DESTool

O DESTool¹ é uma ferramenta projetada para o estudo e análise de Sistemas de Eventos Discretos (SEDs). Ele utiliza autômatos para a modelagem de sistemas, permitindo aos usuários criar modelos detalhados e precisos. A partir desses modelos, uma variedade de operações pode ser executada, como a composição de plantas, o cálculo de supervisores, e a verificação das propriedades de um modelo específico. O DESTool não se limita a uma interface gráfica para a representação dos autômatos. Ele também suporta a execução de operações por meio de scripts, utilizando a biblioteca libFaudes, escrita em C++ e disponível como código aberto. Isso amplia significativamente a flexibilidade do software, permitindo aos usuários automatizar tarefas complexas e personalizar suas análises de acordo com necessidades específicas. Entre as operações relacionadas a autômatos, são destacadas a composição de autômatos, a gestão de multitarefas, e o diagnóstico de falhas. Essas funcionalidades permitem o desenvolvimento e teste de sistemas de automação.

Uma das características mais notáveis do DESTool é sua capacidade de realizar a síntese de supervisores de várias formas. O software oferece métodos diversos para a criação de supervisores, incluindo o supervisor monolítico, o supervisor modular, o supervisor reduzido, e o supervisor baseado em normalização. Cada um desses métodos possui suas próprias vantagens e é adequado para diferentes tipos de sistemas e requisitos de controle. A flexibilidade na síntese de supervisores permite que os usuários escolham a abordagem mais eficiente e eficaz para suas necessidades específicas. Além dessas funcionalidades, o DESTool também se destaca por sua capacidade de realizar animações interativas. Essa funcionalidade, chamada de “Animation”, permite que os usuários explorem o comportamento dos sistemas em um ambiente dinâmico e visual. Durante a animação, os usuários podem ativar e desativar eventos da planta passo a passo, observando em tempo real como essas alterações afetam o estado do sistema e as transições entre estados. Isso proporciona uma compreensão profunda do funcionamento do sistema e das interações entre seus componentes.

A “Animation” oferece uma visualização clara e detalhada do supervisor, mostrando o estado atual das plantas e quais transições estão habilitadas ou desabilitadas. Isso é particularmente útil para identificar e resolver problemas, otimizar o desempenho do sistema, e garantir que o comportamento do sistema esteja alinhado

¹ <<https://www.fgdes.tf.fau.de/destool/>>

com os objetivos de controle desejados.

2.5.2 FlexFact

O FlexFact que é desenvolvido pelo mesmo grupo criador do DESTool (FGDES), se trata de uma ferramenta avançada para a simulação de processos industriais, destacando-se pela sua capacidade de modelar ambientes de manufatura. Esta plataforma permite a criação de simulações realistas que replicam operações industriais, facilitando o teste e a validação de lógicas de controle antes da implementação em ambiente real. Esta ferramenta é especialmente útil para engenheiros e pesquisadores que buscam otimizar o desempenho de sistemas de automação, oferecendo uma interface intuitiva e recursos abrangentes para análise e depuração.

2.5.3 Compositional Interchange Format 3

O Compositional Interchange Format 3² (CIF3) é uma ferramenta computacional desenvolvida para facilitar o projeto, análise e implementação de sistemas de controle complexos. Sua versatilidade e eficácia o tornam uma escolha popular entre os engenheiros de controle e automação em todo o mundo.

O CIF3 é baseado em uma abordagem de modelagem composicional, que permite aos engenheiros construir modelos hierárquicos de sistemas complexos a partir de componentes individuais. De acordo com Murray, Sastry e Zexiang (1994), essa abordagem facilita a reutilização de modelos e a integração de diferentes subsistemas em um único sistema global. Além disso, o CIF3 suporta uma variedade de formalismos de modelagem, incluindo autômatos finitos, redes de Petri e modelos de estado espaço, o que o torna adequado para uma ampla gama de aplicações na engenharia de controle e automação.

O CIF3 encontra uma ampla gama de aplicações na engenharia de controle e automação, desde o projeto de sistemas de controle supervisão até a modelagem de sistemas híbridos complexos. Segundo Ramadge e Wonham (1989), o CIF3 é amplamente utilizado em indústrias de processos, como petróleo e gás, química e alimentos, para desenvolver estratégias de controle avançadas e otimizar o desempenho operacional. Além disso, o CIF3 é utilizado em aplicações de automação de manufatura, robótica e sistemas de transporte inteligentes.

O CIF3, originalmente desenvolvido pela Universidade de Tecnologia de Eindhoven, foi posteriormente integrado ao projeto ESCET (Eclipse Supervisory Control Engineering Toolkit), uma plataforma colaborativa mantida pela comunidade Eclipse. O ESCET reúne um conjunto de ferramentas para o controle supervisão, incluindo o CIF3, visando facilitar o desenvolvimento, simulação e verificação de controladores

² <<https://eclipse.dev/escet/cif/>>

supervisórios em um ambiente integrado. Essa integração amplia as capacidades do CIF3, permitindo seu uso em um contexto mais amplo de engenharia de controle.

2.5.4 OpenPLC

O OpenPLC³ é uma plataforma avançada projetada para programação de Controladores Lógicos Programáveis (CLPs) de código aberto. Essa ferramenta permite a criação detalhada de lógicas de controle através de linguagens de programação padrão da indústria, facilitando a modelagem e implementação de sistemas de automação industrial. Com o OpenPLC, os usuários podem desenvolver e testar diferentes estratégias de automação, desde simples sistemas de controle até aplicações mais complexas em ambientes industriais. A flexibilidade da plataforma permite a integração com hardware específico e a execução de scripts de processos industriais em tempo real, garantindo a precisão e eficiência no desenvolvimento de projetos.

Além da programação convencional, o OpenPLC suporta a criação de lógicas de controle utilizando linguagens de script e bibliotecas especializadas. Esta abordagem versátil permite adaptar as funcionalidades do sistema de automação conforme as necessidades específicas de cada projeto, garantindo compatibilidade e escalabilidade. Uma característica destacada do OpenPLC é sua capacidade de oferecer visualizações interativas das operações de controle, permitindo aos usuários monitorar e ajustar variáveis em tempo real, o que auxilia na depuração e otimização de processos industriais, proporcionando uma interface para análise e ajuste de parâmetros.

2.5.5 Factory IO

O Factory IO⁴ é uma ferramenta voltada para simulação de processos industriais em ambientes virtuais. Desenvolvida para oferecer uma experiência realista, permite a criação e simulação de fábricas virtuais, onde usuários podem modelar e testar sistemas de automação industrial de forma precisa e interativa. Com o Factory IO, é possível criar layouts detalhados de fábricas, incluindo máquinas, sensores, atuadores e esteiras transportadoras. A interface intuitiva permite aos usuários configurar e monitorar o comportamento de cada componente, facilitando a simulação de operações industriais complexas em tempo real.

Além da modelagem gráfica, o Factory IO oferece suporte para integração com controladores lógicos programáveis (CLPs), permitindo a execução de lógicas de controle programadas. Esta funcionalidade permite o desenvolvimento e teste de sistemas de automação antes da implementação física, o que pode ajudar na eficiência e redução de custos no processo de produção. Da mesma forma que o OpenPLC citado

³ <<https://autonomylogic.com>>

⁴ <<https://factoryio.com>>

anteriormente, o Factory IO tem a capacidade de oferecer feedback visual instantâneo durante a simulação, reforçando a otimização e a identificação de potenciais problemas de projeto.

2.5.6 Comunicação Modbus TCP

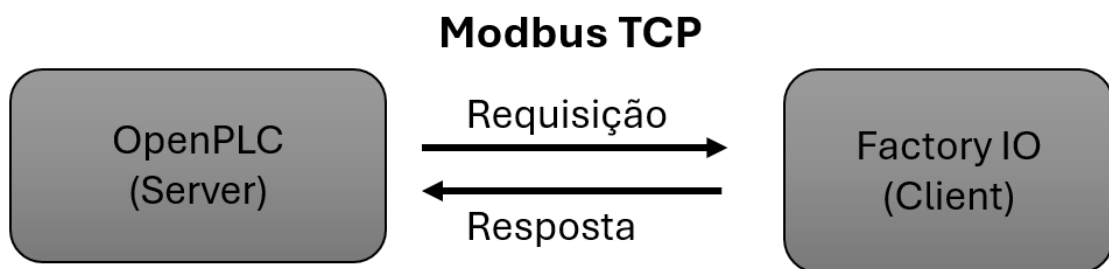
Modbus TCP é um protocolo de comunicação de rede amplamente utilizado na automação industrial, adaptado do protocolo Modbus tradicional. Ele opera sobre redes Ethernet utilizando TCP/IP, permitindo a comunicação entre dispositivos como CLPs, sensores e atuadores. Cada dispositivo em uma rede Modbus TCP possui um endereço IP exclusivo e utiliza a porta 502 para comunicação padrão (ORGANIZATION, 2006).

A estrutura de uma mensagem Modbus TCP inclui um cabeçalho MBAP, um código de função e os dados específicos da função. Operando no modo cliente-servidor, o cliente inicia a comunicação enviando comandos para ler ou escrever dados no servidor (SIEMENS, 2018). O Modbus TCP é amplamente utilizado em sistemas SCADA para monitoramento e controle de equipamentos, devido à sua simplicidade, facilidade de implementação e compatibilidade com diversos dispositivos de automação (SIEMENS, 2018).

2.5.6.1 Integração entre OpenPLC e Factory IO via Modbus TCP

A integração entre OpenPLC e Factory IO é feita via Modbus TCP, oferecendo uma solução para simulação em ambientes industriais virtuais. O Modbus atua como um protocolo de comunicação robusto, facilitando a troca de dados em tempo real entre o controlador (OpenPLC) e a simulação de fábrica (Factory IO). Isso permite que o usuário desenvolva e valide sequências de controle diretamente na interface do Factory IO, observando como essas lógicas interagem com o ambiente virtual da fábrica (ALVES J., 2020). Na figura 2 é ilustrada a comunicação entre as ferramentas.

Figura 2 – Diagrama da comunicação Modbus.



Fonte: Autor (2024).

Essa comunicação bidirecional proporciona benefícios, como a capacidade de testar e otimizar programas de automação antes da implementação física. Os

usuários podem monitorar variáveis de processo, acionar dispositivos virtuais e simular condições de operação diversas, tudo de forma segura e controlada dentro do ambiente virtual do Factory IO. Além disso, a integração via Modbus entre OpenPLC e Factory IO oferece uma representação visual imersiva e interativa das operações industriais. Os usuários podem observar o comportamento do sistema em tempo real, realizar ajustes nas lógicas de controle e avaliar o desempenho do código antes de sua aplicação prática, promovendo eficiência e redução de custos no ciclo de desenvolvimento de automação industrial.

3 METODOLOGIA

Este projeto foi inspirado no trabalho de conclusão de curso (TCC) de Oliveira (2021), intitulado "Avaliação do software CIF3 para projetos e simulação de sistemas a eventos discretos". Oliveira (2021) investigou a eficácia do software CIF3 na automação industrial, utilizando Sistemas a Eventos Discretos (SEDs) e a Teoria de Controle Supervisório (TCS). O estudo compreendeu duas simulações específicas: um estudo de caso de uma esteira industrial e um processo mais complexo de envasamento de cerveja. O autor adotou metodologias didáticas para o uso do CIF3, como a organização por MVC (Model-View-Controller) e a utilização de autômatos para monitorar a localização do supervisor. Concluiu-se que o CIF3 é adequado para projetos e simulações de SEDs, destacando-se pela constante evolução e adição de novos recursos. Modelos e plantas foram inicialmente desenvolvidos no software DESTool e, posteriormente, transferidos para o CIF3, onde foram executadas as simulações com base no supervisor calculado pelo DESTool.

Este trabalho propõe uma abordagem alternativa para a simulação de SEDs, adaptando as metodologias do estudo de Oliveira (2021) para outros contextos e ferramentas. Utilizando uma das simulações propostas e criadas por Oliveira (2021), este projeto propõe remodelá-lo na linguagem ladder para simulação tanto em Controladores Lógicos Programáveis (CLPs) quanto em software simulador de processo fabril, com o objetivo de facilitar a simulação de sistemas industriais e propor um conjunto de recursos para uma abordagem didática de simulação do controle supervisório de SEDs.

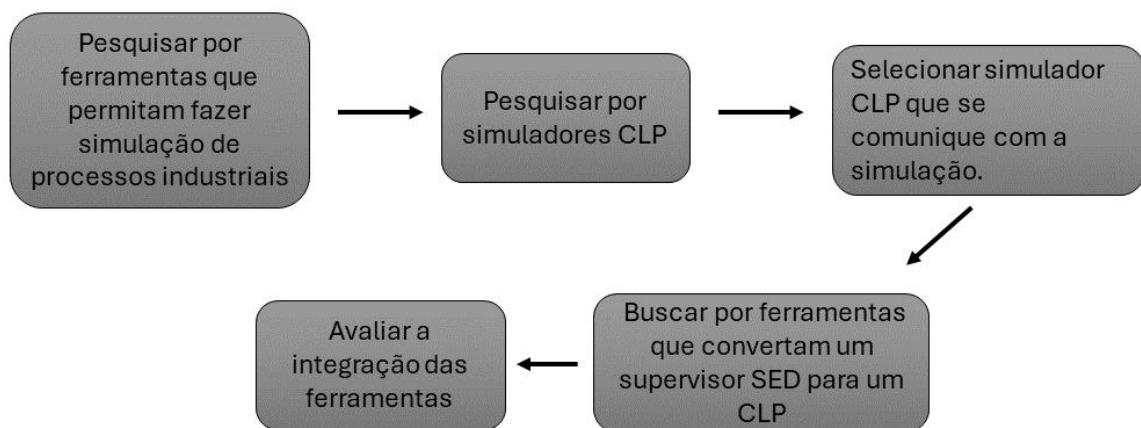
Para a implementação da abordagem proposta, foram seguidas uma série de etapas metodológicas, detalhadas a seguir:

1. Pesquisar por ferramentas que permitam fazer simulação de processos industriais: O objetivo é identificar softwares que simulem ambientes industriais de forma realista, proporcionando um ambiente de testes seguro e controlado para avaliar diferentes cenários operacionais e de controle.
2. Pesquisar por simuladores CLP: Visa encontrar simuladores que possam programar e testar lógicas de controle utilizadas em CLPs, garantindo que o comportamento esperado do sistema seja replicado de forma precisa antes de sua implementação em campo.

3. Selecionar simulador CLP que se comunique com a simulação: O objetivo é escolher um simulador que seja compatível com a ferramenta de simulação industrial, possibilitando a troca de dados em tempo real e a integração das operações de controle com o ambiente simulado.
4. Buscar por ferramentas que convertam um supervisor SED para um CLP: Mesmo que não tenha sido possível, a tentativa tinha como objetivo automatizar a conversão de modelos de sistemas a eventos discretos (SED) para lógicas de CLP, otimizando o processo de desenvolvimento e implementação de sistemas de controle.
5. Avaliar a integração das ferramentas: Objetiva verificar a compatibilidade e a eficiência da comunicação entre as diferentes ferramentas selecionadas, assegurando que a simulação e o controle funcionem de forma integrada e coerente, facilitando a validação e a otimização do sistema como um todo.

Na figura 3 é possível observar os passos necessários para se atingir o objetivo.

Figura 3 – Diagrama da metodologia.



Fonte: Autor (2024).

3.1 Seleção de ferramentas

No processo de desenvolvimento deste projeto, foram consideradas diversas ferramentas de Controladores Lógicos Programáveis (CLP), entre as quais se destacam CODESYS, TwinCAT e OpenPLC. Cada uma dessas ferramentas possui características distintas e oferece diferentes funcionalidades que poderiam potencialmente beneficiar o projeto. No entanto, após uma análise de suas capacidades e compatibilidades, é escolhido o OpenPLC.

A decisão de utilizar o OpenPLC é motivada principalmente por sua interface simples e intuitiva, que facilita a configuração e a operação, além de sua integração com simuladores de processos industriais. Essa integração é essencial para garantir que a simulação se aproxime ao máximo da realidade operacional. O OpenPLC é compatível com uma ampla gama de dispositivos de hardware de baixo custo, como Raspberry Pi e Arduino, e suporta as linguagens de programação padrão da IEC 61131-3. Esta flexibilidade permite aos usuários testar programas de lógica de controle em ambientes virtuais antes de implementá-los em hardware real, tornando-o uma ferramenta versátil para pesquisa e desenvolvimento, conforme observado por Alsabbagh, Kim e Langendörfer (2023).

Um exemplo de utilização do OpenPLC é o trabalho recente de Afenu, Asiri e Saxena (2024) onde o OpenPLC serve como o controlador lógico programável principal, executando a lógica de controle predefinida e controlando um sistema de semáforo que atua como dispositivo escravo. A comunicação entre o OpenPLC e o sistema de semáforo é realizada usando o protocolo Modbus TCP, permitindo a troca de dados em tempo real entre os sistemas.

Para a simulação dos processos fabris, foram avaliadas opções de ferramentas especializadas, destacando-se o FlexFact e Factory IO. Após uma análise das características de cada uma, optou-se pela utilização do Factory IO. Esta escolha foi baseada na compatibilidade do Factory IO com o OpenPLC, além de seus recursos e funcionalidades abrangentes, que permitem uma simulação mais realista e detalhada dos processos industriais.

O Factory IO se destaca por sua capacidade de criar ambientes de simulação interativos e visualmente ricos, o que facilita a visualização e o teste das lógicas de controle implementadas. Por exemplo, Salah et al. (2023) utilizaram o Factory I/O para simular uma máquina de enchimento de iogurte, replicando os processos de enchimento, escaneamento e remoção, o que facilita a identificação de áreas para otimização e melhoria da eficiência e produtividade do sistema.

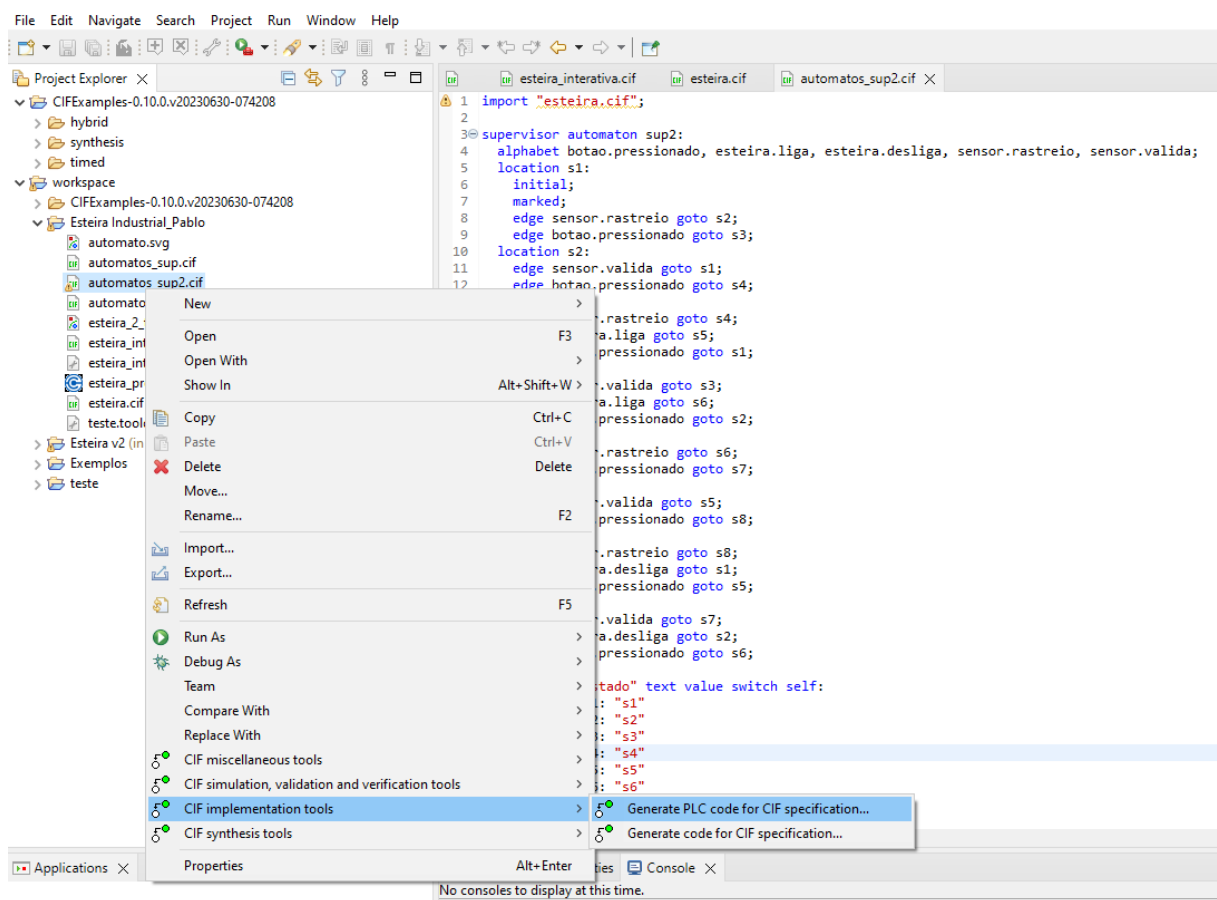
Segundo Guerra et al. (2022) vários artigos de conferências relataram a aplicação bem-sucedida do Factory I/O para ensinar programação de CLP em cursos universitários em todo o mundo, por exemplo, na Universidade Claude Bernard (Lyon, França), na Universidade Pan-Americana (Aguascalientes, México), na Universidade Western Kentucky (EUA) e na Universidade de Bagdá (Irão). Assim, a combinação do OpenPLC com o Factory IO resulta em uma plataforma robusta e eficiente para simulação e controle de sistemas industriais, assegurando que os objetivos do projeto sejam alcançados de maneira eficaz e precisa.

3.2 Procedimentos Metodológicos

3.2.1 Transcrição do autômato supervisor para CLP

A transcrição de um dos autômatos desenvolvidos no trabalho de Oliveira (2021) foi realizada utilizando a linguagem ladder com o objetivo de facilitar a simulação dos Sistemas a Eventos Discretos (SEDs). Inicialmente, havia sido planejado utilizar a função nativa de exportação de arquivos no formato .cif do Eclipse ESCET que pode ser observada na Figura 4 permitindo assim uma transição automática entre as etapas de modelagem e simulação.

Figura 4 – Opções de exportação nativas do ESCET.



Fonte: Autor (2024).

Diversos problemas técnicos, identificados durante a implementação, impediram o uso direto desta funcionalidade. Entre os problemas encontrados, destacam-se incompatibilidades de formato e dificuldades na tradução direta das especificações de controle do autômato para o código ladder. Esses obstáculos, que serão detalhadamente discutidos na seção 4.5, forçaram a buscar alternativas viáveis para assegurar a integridade e a eficácia do sistema simulado.

Diante desses desafios, foram exploradas e avaliadas outras metodologias para alcançar os objetivos propostos. As alternativas foram pesquisadas, considerando

a complexidade e os requisitos específicos do projeto. Decidiu-se por implementar manualmente as transformações necessárias para adaptar os autômatos à linguagem ladder. Este processo manual envolveu uma reinterpretação das especificações de controle e a codificação das lógicas de operação, onde as nuances e particularidades do controle supervisorio foram traduzidas e operacionalizadas no ambiente de simulação.

3.2.2 Configuração do ambiente de simulação

Após a avaliação de diversas opções disponíveis, foram selecionadas as ferramentas OpenPLC e Factory IO para a condução das simulações necessárias ao projeto. A escolha do OpenPLC foi justificada pela sua interface amigável e intuitiva, que facilita significativamente o processo de configuração e operação. Além disso, o OpenPLC destaca-se por sua excelente capacidade de integração com diversos simuladores, o que amplia as possibilidades de aplicação e teste dentro do ambiente de simulação.

Por outro lado, o Factory IO foi selecionado devido à sua compatibilidade com o OpenPLC, bem como pelos seus recursos que permitem a criação de simulações industriais realistas. O Factory IO é reconhecido por oferecer um conjunto de funcionalidades que suportam a modelagem e simulação de processos industriais, tornando-o uma ferramenta que atende às exigências específicas deste projeto.

O ambiente de simulação foi configurado utilizando o Factory IO, o qual, após testes preliminares, demonstrou ser a opção mais compatível e capaz de fornecer os recursos necessários para cumprir os objetivos do projeto de maneira eficaz. A configuração incluiu a integração perfeita entre o OpenPLC e o Factory IO, permitindo uma comunicação fluida e eficiente entre as duas plataformas. Esta integração assegurou que todas as variáveis e estados necessários fossem corretamente monitorados e controlados, proporcionando uma base sólida para a execução das simulações e a análise dos resultados obtidos. Dessa forma, a combinação do OpenPLC com o Factory IO oferece uma solução poderosa e versátil, que atende plenamente às necessidades técnicas e operacionais do projeto.

3.2.3 Execução das simulações

Foram conduzidas várias simulações utilizando o código CLP previamente gerado e o ambiente de simulação configurado. Este processo envolveu a execução de múltiplos cenários de teste para que o sistema operasse de maneira consistente e previsível sob diferentes condições. O ambiente de simulação, configurado utilizando as ferramentas selecionadas, permitiu a replicação de condições industriais reais, proporcionando uma plataforma robusta para a análise do desempenho do sistema.

Os resultados obtidos a partir dessas simulações foram analisados e

comparados com os dados e conclusões previamente alcançados por Oliveira (2021). Esta comparação teve o objetivo de avaliar a eficácia da nova abordagem proposta no presente estudo. Ao contrastar os resultados, buscou-se identificar melhorias, consistências e quaisquer divergências significativas que pudessem fornecer insights sobre a performance e a viabilidade do sistema implementado. Além de validar os aspectos técnicos da nova metodologia, a análise comparativa destacou benefícios adicionais e áreas potenciais de aprimoramento.

3.3 Análise dos resultados

Os resultados das simulações foram analisados com o objetivo de avaliar a eficácia do controle supervisorio de Sistemas a Eventos Discretos (SEDs) utilizando Controladores Lógicos Programáveis (CLPs). Esta análise verifica se a nova abordagem proposta no presente estudo atende aos requisitos de desempenho e confiabilidade esperados.

O projeto tirou proveito da experiência adquirida e dos resultados obtidos no trabalho de Oliveira (2021). Utilizando as conclusões de Oliveira (2021) como ponto de partida, este estudo foi capaz de incorporar novas ferramentas e ajustar técnicas existentes para atender a diferentes contextos e necessidades específicas. Este processo de adaptação e expansão permitiu não apenas validar as metodologias anteriores, mas também inovar e melhorar as práticas de controle supervisorio de SEDs, proporcionando uma contribuição significativa ao campo da automação industrial.

3.4 Limitações e soluções

As limitações encontradas durante o desenvolvimento das simulações e as soluções adotadas são discutidas, com o intuito de fornecer um panorama completo das dificuldades enfrentadas e dos métodos empregados para superá-las. Este exame crítico é essencial para compreender os desafios inerentes ao processo e as estratégias eficazes para contorná-los.

Uma das principais dificuldades identificadas foi a necessidade de transição manual dos autômatos do formato CIF para a linguagem ladder. Embora este processo tenha se mostrado bastante trabalhosa e exigente em termos de tempo e esforço, ele se revelou viável para garantir a integridade e a funcionalidade dos sistemas simulados. A transição manual envolveu a reinterpretação meticulosa das especificações dos autômatos e sua codificação precisa na linguagem ladder, o que assegurou que todas as nuances e detalhes dos modelos originais fossem preservados.

Esta abordagem permitiu manter a consistência e a precisão das simulações, proporcionando uma base sólida para a validação dos sistemas de controle. As soluções adotadas para essas limitações também incluíram a utilização de ferramentas

auxiliares para verificação e validação dos códigos gerados, garantindo que as lógicas de controle fossem implementadas corretamente e funcionassem conforme esperado. A documentação do processo e das soluções encontradas também facilitou a replicação e a aplicação futura das metodologias desenvolvidas.

Em suma, a discussão das limitações e das soluções adotadas não só esclarece os obstáculos superados, mas também oferece lições e recomendações para futuros trabalhos na área, promovendo o contínuo aprimoramento das práticas de simulação e controle de Sistemas a Eventos Discretos utilizando CLPs.

4 SIMULAÇÃO DESENVOLVIDA

Nesta seção é detalhado o funcionamento do software através da aplicação prática de um cenário industrial envolvendo a operação de uma esteira transportadora.

Como citado no capítulo 3, Este estudo de caso foi inspirado em uma das simulações realizadas por Oliveira (2021). Uma vez que o supervisor obtido para o exemplo teve que ser traduzido manualmente para o ladder, foi escolhido o exemplo mais simples da proposta de Oliveira (2021), mas suficiente para o propósito deste TCC.

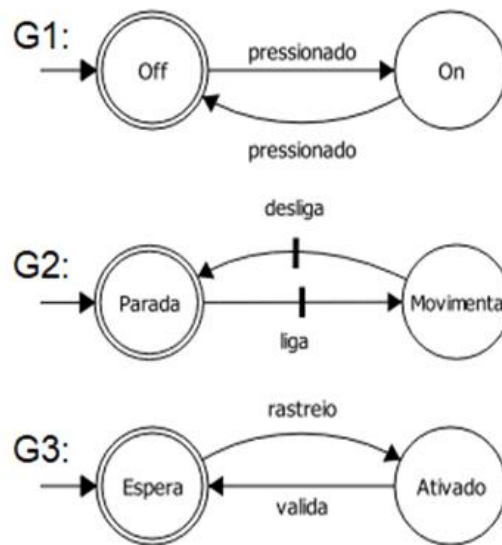
A simulação consiste no transporte de caixas ao longo da esteira, desde o ponto de carga até o ponto de descarga, onde um sensor será utilizado para contabilizar a quantidade de caixas transportadas. Além disso, a aplicação é projetada para permitir a interação do usuário com o sistema em tempo real, possibilitando o controle manual da esteira por meio de comandos para ligar e desligar o equipamento. Esse nível de interação visa proporcionar uma compreensão mais profunda do controle supervisorio e da eficiência do software na gestão de processos industriais automatizados, permitindo a observação direta do impacto das ações do usuário sobre o funcionamento da esteira e a precisão do sistema de contagem de caixas.

Nas subseções a seguir são apresentados detalhamentos sobre aspectos dos autômatos e transcrição para ladder.

4.1 Plantas e especificações

Na Figura 5, são apresentadas as plantas da operação da esteira industrial selecionadas para a simulação. A planta identificada como G1 representa o botão de controle, que alterna entre os estados Off e On a cada evento de pressão do botão. A planta G2 corresponde à esteira, que inicia sua operação no estado Parada e muda para o estado Movimenta quando ocorre o evento liga, retornando ao estado Parada com o evento desliga. A planta G3 descreve o funcionamento do sensor, que começa no estado Espera. Quando ocorre um evento de rastreamento, o sensor passa para o estado Ativado, e posteriormente, o evento valida é responsável por incrementar o contador, registrando cada caixa que passa pelo sensor. Essas plantas detalham a lógica de operação dos componentes do sistema, proporcionando uma visão clara de como os eventos e estados interagem para simular o funcionamento real da esteira industrial e garantir a precisão do sistema de contagem de caixas.

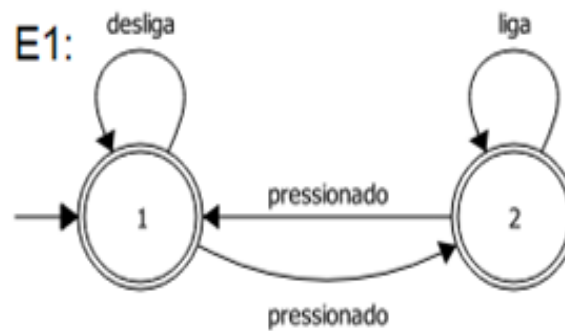
Figura 5 – Plantas do sistema de operação da esteira.



Fonte: Oliveira (2021).

O modelo de especificação para este sistema é ilustrado na Figura 6. Esta especificação descreve o comportamento operacional da esteira transportadora em resposta às interações do usuário com o botão de acionamento. Conforme mostrado na figura, a esteira entra no estado de movimento, identificado como estado 2, quando o botão de acionamento é pressionado, este estado sinaliza ao sistema que a esteira pode ser ligada, transportando as caixas ao longo de sua extensão. Adicionalmente, a especificação inclui a lógica para que a esteira retorne ao estado inicial de parada, identificado como estado 1 quando o botão é pressionado novamente. Este mecanismo de alternância entre os estados de movimento e parada é essencial para garantir a operação controlada da esteira, permitindo que o usuário inicie e interrompa o transporte das caixas conforme necessário. A figura serve como uma representação visual desta lógica operacional, facilitando a compreensão dos processos envolvidos no controle da esteira e assegurando que todos os estados e transições sejam corretamente implementados no sistema de simulação.

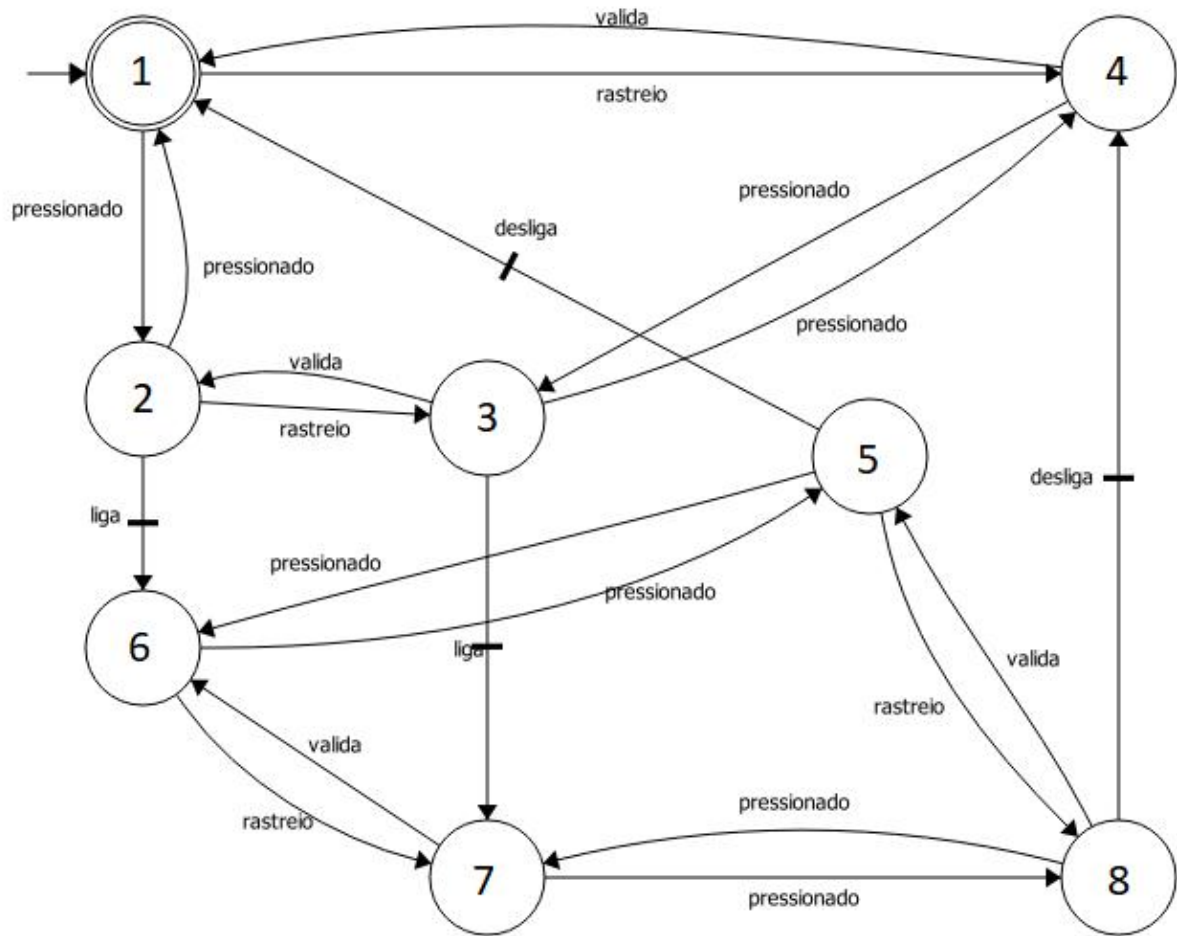
Figura 6 – Modelo de especificação do sistema.



Fonte: Oliveira (2021).

A partir das plantas G1, G2 e G3, bem como da especificação de controle E1, foi realizado o cálculo do supervisor responsável pelo controle do sistema. As plantas G1, G2 e G3 fornecem a estrutura e a lógica necessárias para a modelagem dos componentes individuais do sistema, enquanto a especificação de controle E1 detalha os requisitos e as condições operacionais que devem ser atendidos para garantir o funcionamento adequado da esteira e dos seus elementos associados. Com base nessas informações, o supervisor é calculado para assegurar que o sistema opere conforme esperado, monitorando e controlando as transições entre os diferentes estados das plantas. A Figura 7 apresenta o supervisor obtido, oferecendo uma visualização de como o sistema de controle gerencia as operações da esteira. Esta figura apresenta o comportamento dinâmico do sistema e ilustra como o supervisor regula as interações entre os eventos e estados descritos nas plantas e na especificação de controle. Destaca-se que o supervisor é obtido a partir das plantas, da especificação de controle e da TCS.

Figura 7 – Autômato supervisor para controle do sistema simulado.



Fonte: Autor (2024).

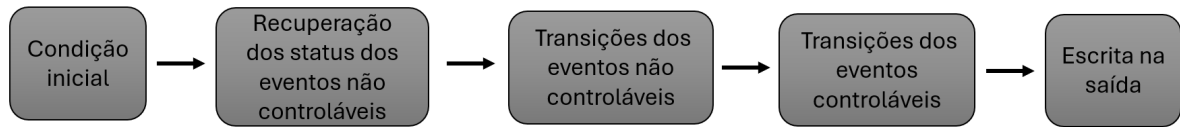
Para mais detalhes do projeto do supervisor, ver as seções 4.2 e 4.5 de Oliveira (2021).

4.2 Transcrição em ladder

Para o desenvolvimento dessa etapa foi utilizado como referência, a dissertação de Cruz (2011), que detalha a metodologia para garantir o funcionamento do diagrama ladder baseado no supervisor além de sugerir métodos para evitar o comportamento não desejado do programa, como por exemplo o efeito avalanche e o problema da escolha, que não se aplicam ao presente trabalho.

Na figura 8 tem-se um fluxograma das etapas utilizadas para fazer a transcrição para ladder, baseado na metodologia utilizada por Cruz (2011), sendo que o presente trabalho se baseia nessa metodologia, com adaptações. Além disso, destaca-se que a transcrição foi feita de forma manual.

Figura 8 – Fluxo da metodologia.



Fonte: Autor (2024).

O software OpenPLC foi utilizado na construção do supervisor no formato ladder, conforme mencionado anteriormente. Este software foi escolhido devido à sua versatilidade e facilidade de integração com diversos simuladores de processos industriais. A construção do diagrama ladder representa graficamente a lógica de controle do sistema.

Na tabela 2, são detalhadas as variáveis utilizadas nesta construção. Esta figura inclui não apenas a lista completa das variáveis, mas também especifica os tipos de cada uma e sua localização na memória do CLP virtual. Este nível de detalhamento é essencial para garantir que o supervisor funcione corretamente, permitindo a correta atribuição de valores e a execução de comandos dentro do ambiente de simulação. A descrição detalhada das variáveis e suas propriedades facilita a compreensão da estrutura do diagrama ladder, assegurando que os elementos necessários para o controle supervisorório estejam configurados e acessíveis.

Tabela 2 – Tabela de Variáveis

Nome	Tipo	Localização	Documentação
Esteira	BOOL	%QX100.0	Esteira
Q	BOOL		Variável auxiliar necessária para funcionamento do contador
Preset	INT		Variável auxiliar necessária para funcionamento do contador
Display	INT	%QW100	Variável que guarda o valor do contador para mostrar no display
e1	BOOL		Estado 1
e2	BOOL		Estado 2
e3	BOOL		Estado 3
e4	BOOL		Estado 4
e5	BOOL		Estado 5
e6	BOOL		Estado 6
e7	BOOL		Estado 7
e8	BOOL		Estado 8
pressionado_2	BOOL	%IX100.0	Auxiliar
pressionado_1	BOOL		Evento gerado através do botão
liga	BOOL		Evento Liga Esteira
desliga	BOOL		Evento Desliga Esteira
valida_2	BOOL		Auxiliar
valida_1	BOOL	%IX100.1	Evento Valida Sensor
rastreio_2	BOOL		Auxiliar
rastreio_1	BOOL		Evento Rastreia Sensor
CTU1	CTU		Contador do número de caixas
M1	BOOL		Bit de memória auxiliar
M0	BOOL		Bit de memória auxiliar

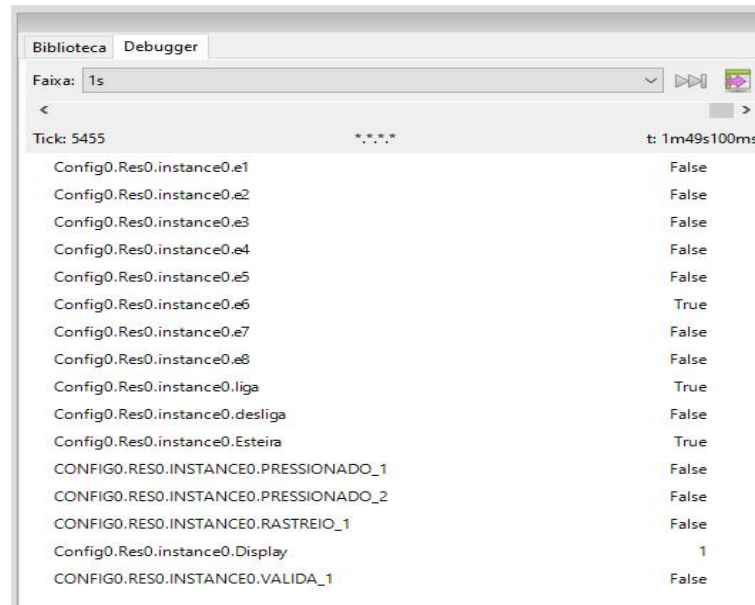
Fonte: Autor (2024).

O diagrama ladder do supervisor, que foi desenvolvido para controlar as operações do sistema, foi dividido em etapas com o objetivo pedagógico de facilitar o entendimento da sequência lógica e fluxo de execução. Com o mesmo objetivo, foram adicionados atrasos propositais nas transições de estados e eventos, para que se possa observar o comportamento do supervisor através do debugger. Como pode ser observado na Figura 9 o funcionamento em tempo real do diagrama implementado, onde na esquerda têm-se o nome das variáveis e na direita o seu status. Na figura em questão, é possível observar pelos status das variáveis que o evento liga foi habilitado, ativando o estado 6 e ligando a esteira, de acordo como o que era esperado quando

observado o supervisor da figura 7.

Essa etapa oferece uma visualização do diagrama ladder, permitindo a análise e a compreensão de todos os aspectos da lógica de controle implementada. Através dos diagramas a seguir, é possível observar como cada componente e variável interagem, destacando-se a estrutura hierárquica e a sequência lógica das instruções que o supervisor executa.

Figura 9 – Debugger do OpenPLC.



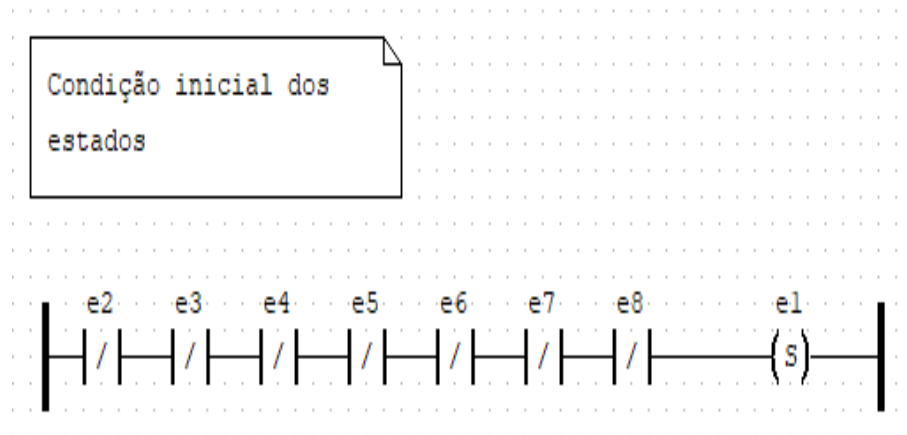
Fonte: Autor (2024).

Destaca-se que em Oliveira (2021), o recurso de acompanhamento dos estados conforme o debugger do OpenPLC também é citado. No caso, o CIF3 possui o recurso State Visualizer e que permite o acompanhamento da evolução dos estados.

4.2.1 Instruções ladder para representar o supervisor

A primeira etapa do processo mostrado na Figura 8 é a condição inicial, sendo esta mostrada na Figura 10. Nesta etapa, os estados estão desativados, com exceção do estado 1, que é ativado, garantindo que o sistema comece sua operação de maneira consistente.

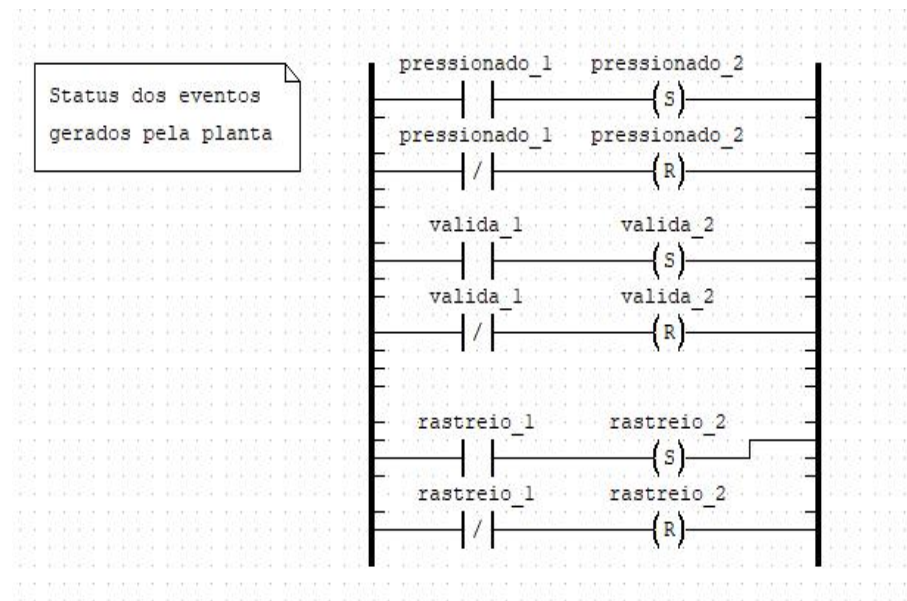
Figura 10 – Condição inicial dos estados.



Fonte: Autor (2024).

A Figura 11 apresenta os eventos não controláveis do sistema. Estes eventos são aqueles que ocorrem independentemente das ações do controlador, como a ativação de sensores ou a detecção de condições específicas no ambiente de operação.

Figura 11 – Status dos eventos não controláveis.



Fonte: Autor (2024).

Para entender como o sistema reage a esses eventos, observa-se a Figura 12 que apresenta as transições correspondentes aos eventos não controláveis. É preciso analisar as transições para garantir que o sistema responda corretamente a todos os eventos possíveis, mantendo a operação segura e eficiente. Quando observado o primeiro segmento da figura, percebe-se que se o estado 1 (e1) estiver ativo e ocorrer o evento pressionado, o estado 2 (e2) será ativado, a variável pressionado_1 será desativada, por sua vez desativando o pressionado_2 e por último será desativado o

estado 1 (e1) completando a transição do estado 1 para o estado 2, respeitando o que foi descrito pelo supervisor da Figura 7.

Figura 12 – Transições dos eventos não controláveis.

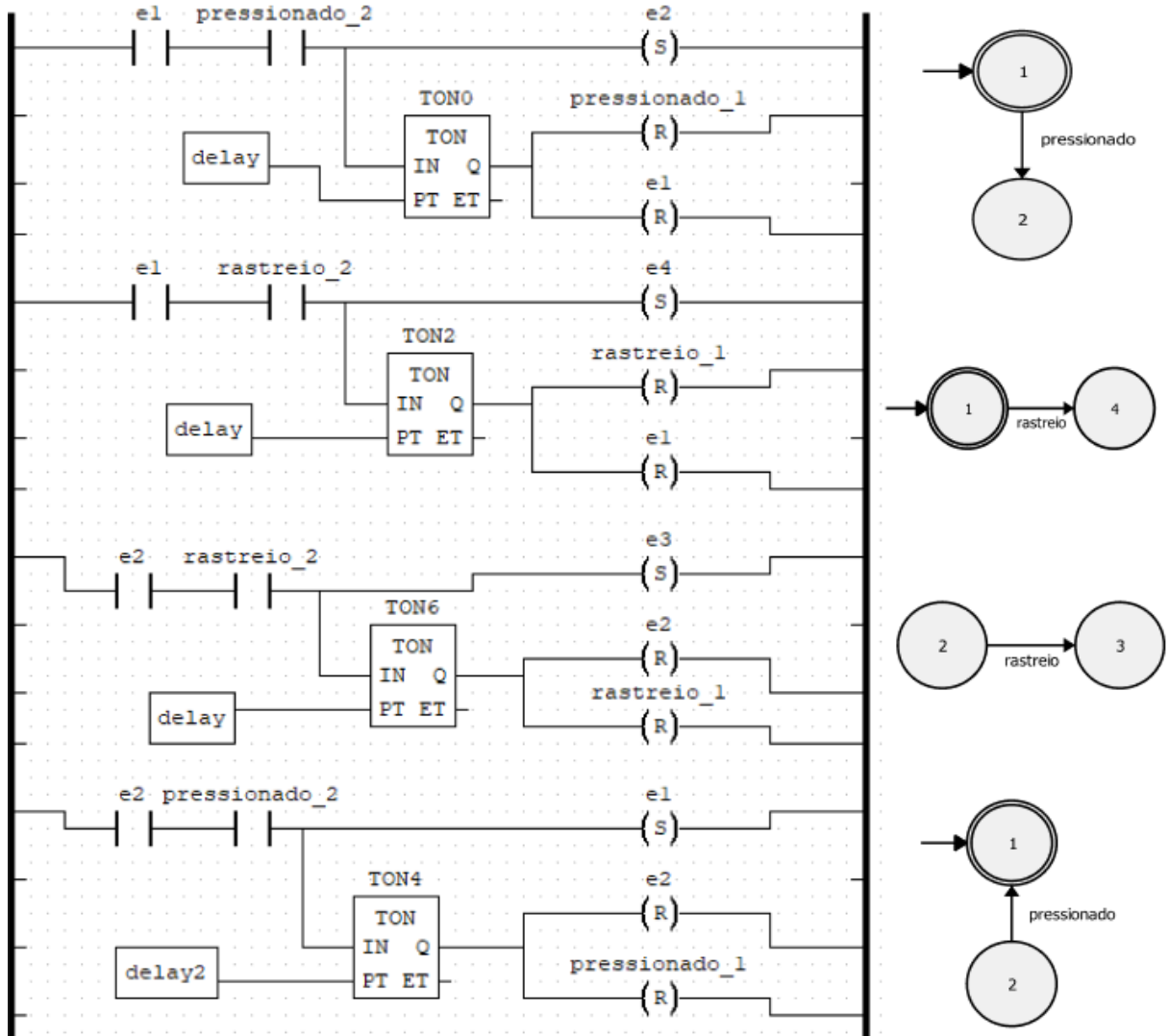
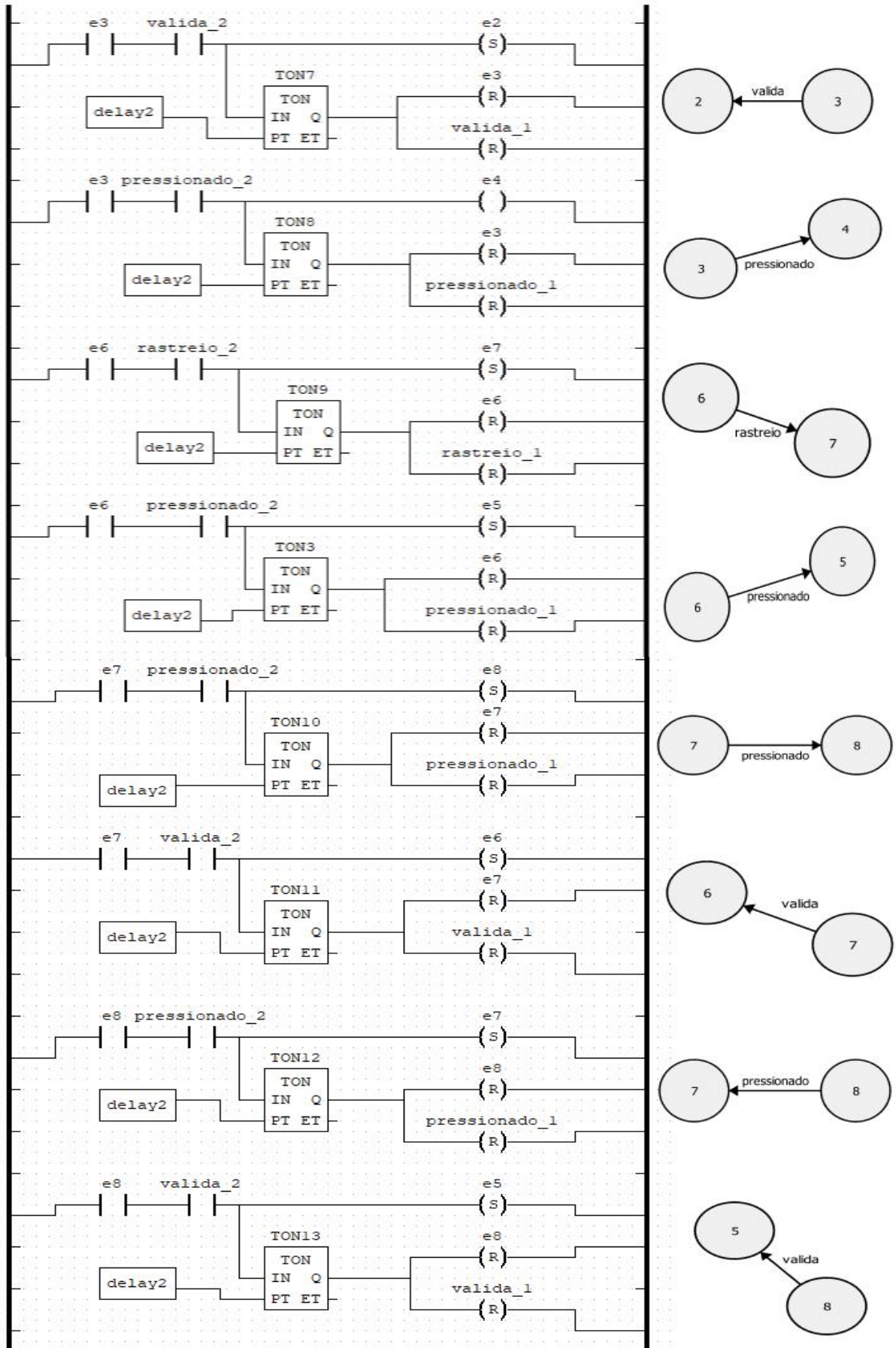
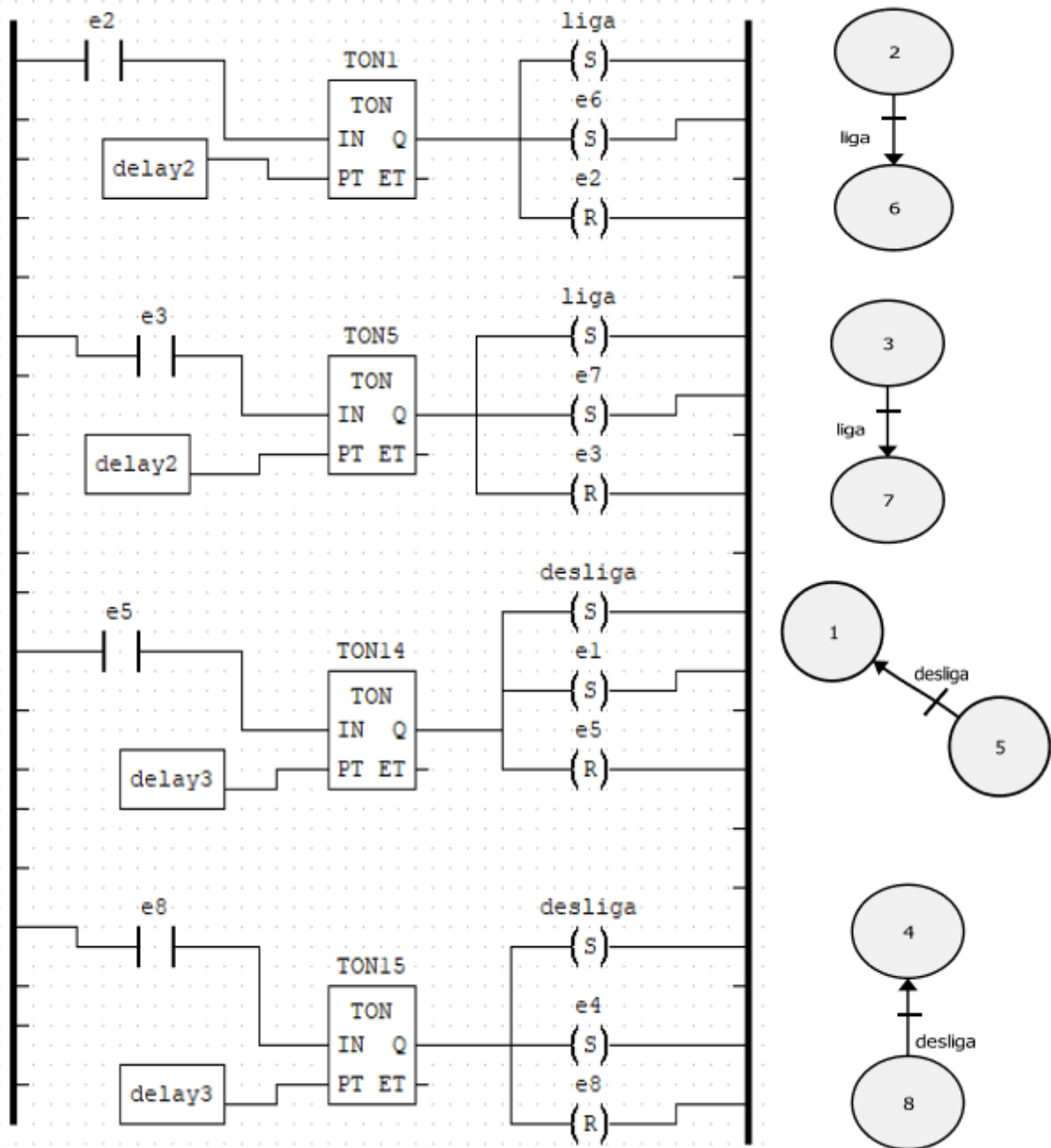


Figura 12 - Continuação.



A seguir, na Figura 13, são exploradas as transições dos eventos controláveis, aqueles que são diretamente manipulados pelo controlador. Estas transições determinam como o sistema deve atuar em resposta a comandos específicos, como ligar ou desligar a esteira transportadora. Fazendo novamente um paralelo com a Figura 7, pode-se observar que se o estado 2 (e2) estiver ativo, a transição liga será habilitada, assim como o estado 6 (e6), e o estado 2 (e2) será desativado, gerando assim a situação demonstrada na Figura 9.

Figura 13 – Transições dos eventos controláveis.

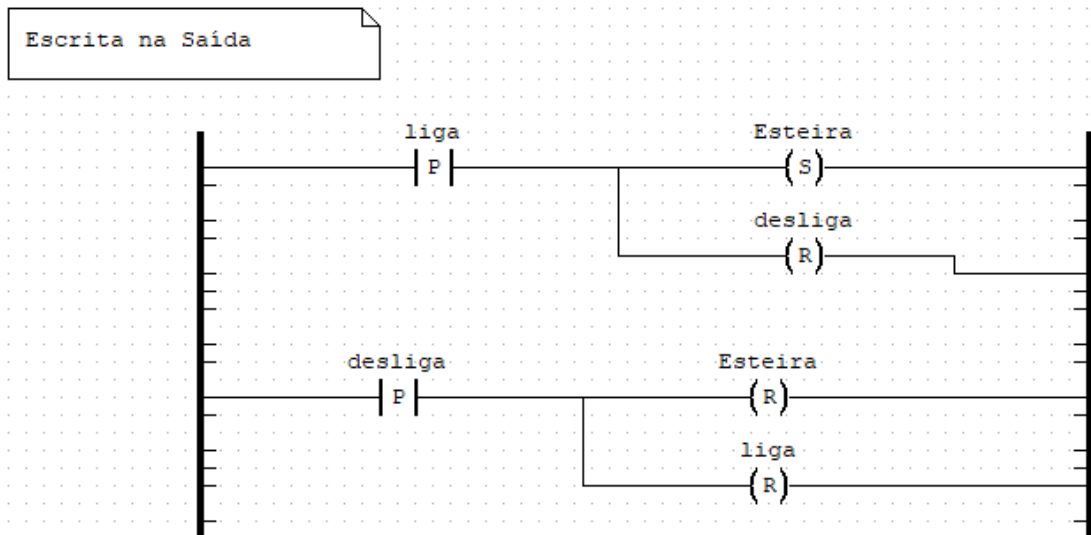


Fonte: Autor (2024).

Por fim, a Figura 14 ilustra a escrita na saída (esteira), onde as ações finais

são executadas. Esta etapa envolve a ativação e desativação da esteira através de um contato borda de subida para que com apenas um pulso seja ativada/desativada a saída esteira e seja desativado o evento desliga/liga, caso fosse utilizado um contato normalmente aberto, quando o evento ocorresse, a bobina ficaria desativada indefinidamente, impedindo o evento concorrente de ocorrer.

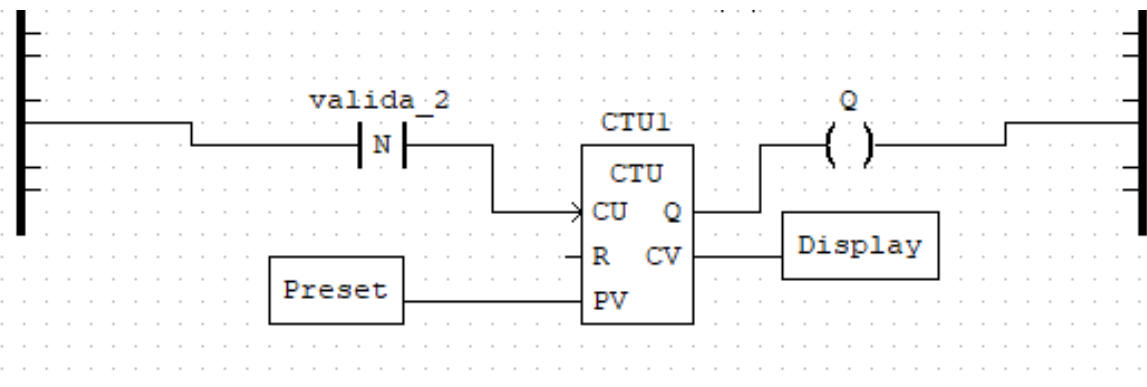
Figura 14 – Escrita na saída.



Fonte: Autor (2024).

Já a Figura 15, representa o incremento no contador de caixas passadas pelo sensor. O funcionamento deste contador não é uma ação do supervisor e sim o que ocorre em estados associados com esse processo, sendo esse associado com o estado Ativado de G3 (mostrado anteriormente na Figura 5).

Figura 15 – Escrita no contador.



Fonte: Autor (2024).

Através desta sequência de figuras e suas explicações, é possível compreender o fluxo completo do diagrama ladder, desde a inicialização dos estados até a execução das ações finais. Cada etapa pode ser monitorada pelo debugger (Figura 9) quando o

modo de simulação está ativo, e ajustada para melhorar a eficiência e segurança na operação do sistema.

4.3 Comunicação Modbus

Após a conclusão do diagrama ladder, é possível gerar automaticamente um programa que pode ser posteriormente compilado no ambiente OpenPLC Runtime. Este processo de compilação permite transformar o diagrama ladder em um código executável que o controlador lógico programável pode interpretar e executar. No OpenPLC Runtime, é realizada então a configuração de um dispositivo virtual, que inclui a definição precisa de endereços de memória para todas as entradas e saídas do sistema. Esta etapa é ilustrada na Figura 16, que fornece uma representação visual detalhada da configuração do dispositivo virtual. É através deste dispositivo virtual que a comunicação com o software de simulação Factory IO é estabelecida, utilizando o protocolo Modbus TCP. Este protocolo de comunicação é amplamente utilizado na automação industrial devido à sua robustez e eficiência, permitindo a troca de dados em tempo real entre o OpenPLC e o Factory IO. Desta forma, a configuração correta do dispositivo virtual e a comunicação via Modbus TCP garantem que a simulação possa replicar fielmente as operações do sistema controlado, proporcionando uma plataforma de testes robusta e precisa.

Figura 16 – Configuração do dispositivo no OpenPLC Runtime.

The screenshot displays the 'Edit slave device' configuration page in the OpenPLC Runtime. The left sidebar contains navigation options: Dashboard, Programs, Slave Devices (highlighted), Monitoring, Hardware, Users, Settings, and Logout. The main content area is titled 'Edit slave device' and includes the following configuration fields:

- Device Name:** Factory_IO
- Device Type:** Generic Modbus TCP Device
- Slave ID:** 1
- IP Address:** 192.168.56.1
- IP Port:** 502

On the right side, there are five sections for configuring Modbus registers, each with 'Start Address' and 'Size' input fields:

- Discrete Inputs (%IX100.0):** Start Address: 0, Size: 4
- Coils (%QX100.0):** Start Address: 0, Size: 4
- Input Registers (%IW100):** Start Address: 0, Size: 2
- Holding Registers - Read (%IW100):** Start Address: 0, Size: 2
- Holding Registers - Write (%QW100):** Start Address: 0, Size: 4

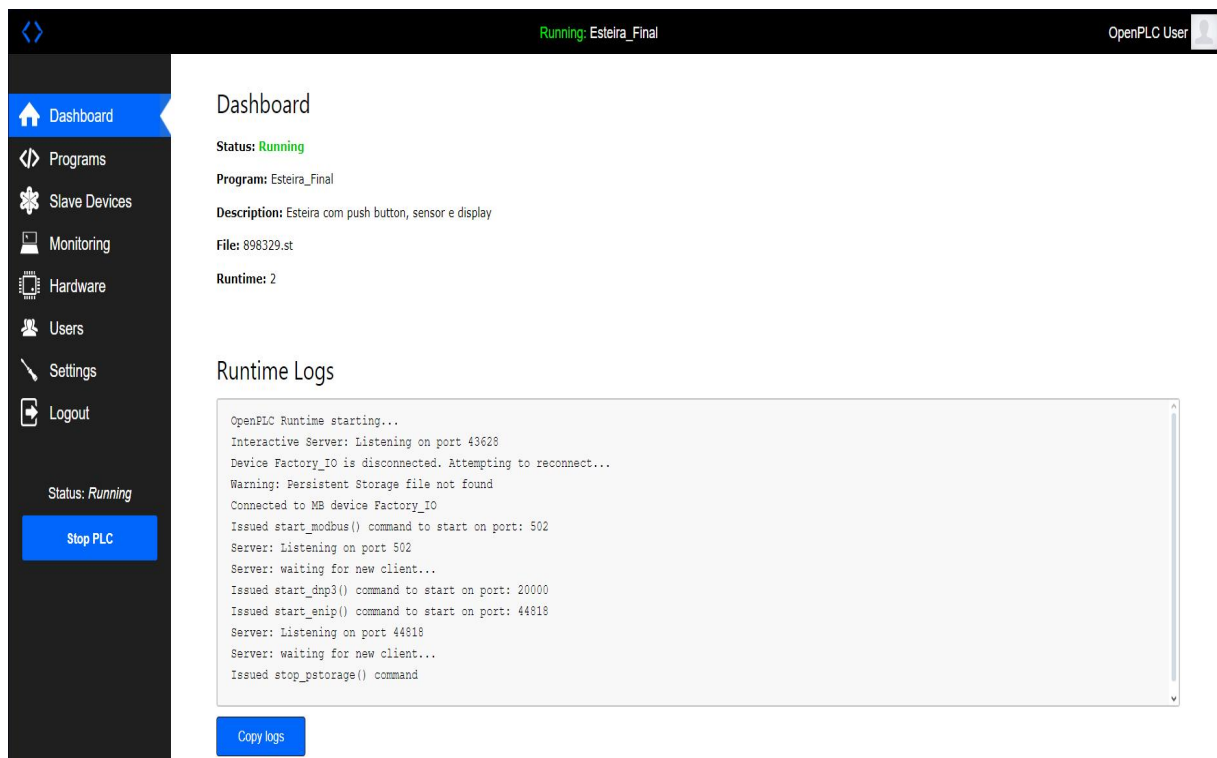
The top status bar shows 'Stopped: Esteira_Final' and 'OpenPLC User'.

Fonte: Autor (2024).

No ambiente OpenPLC Runtime, após a configuração inicial e a compilação do programa, é possível dar início à operação do Controlador Lógico Programável (CLP). Através do dashboard, conforme ilustrado na Figura 17, os usuários podem

monitorar de forma contínua e em tempo real o status do CLP. Este painel de controle oferece uma visão abrangente e detalhada de várias métricas operacionais, permitindo o acompanhamento do desempenho do sistema de maneira eficiente. Além disso, o dashboard fornece acesso ao log de comunicação, onde são registrados os eventos e intercâmbios de dados entre o CLP e o software de simulação Factory IO. Este log é uma ferramenta para a análise e a resolução de problemas, pois documenta cada interação e pode ajudar a identificar rapidamente quaisquer discrepâncias ou falhas na comunicação. Assim, a utilização do dashboard no OpenPLC Runtime não só facilita o monitoramento do status do CLP, mas também assegura que os usuários possam manter um controle rigoroso sobre as operações de comunicação entre os sistemas, garantindo a integridade e a precisão da simulação industrial.

Figura 17 – Painel do OpenPLC Runtime.

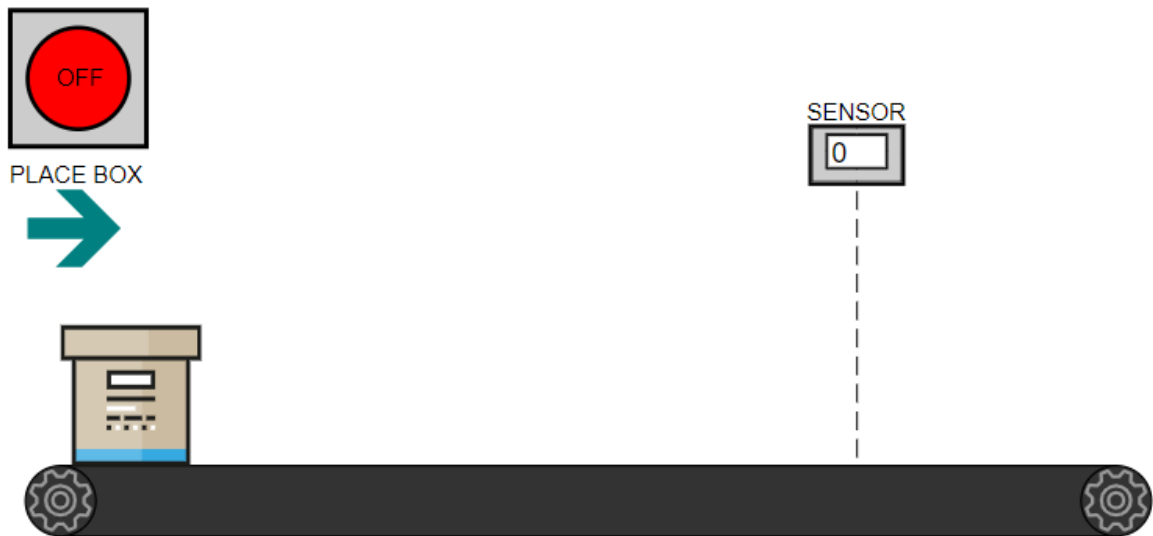


Fonte: Autor (2024).

4.4 Simulação

No trabalho de Oliveira (2021), foram elaboradas duas simulações com o objetivo de avaliar a capacidade do software CIF3 (Atualmente com o nome de ESCET), sendo a primeira (Figura 18) um estudo de caso de uma esteira industrial, onde a simulação apresenta os principais recursos do CIF3 para realização de uma simulação de SEDs, e a segunda de um processo de envasamento de cerveja.

Figura 18 – Simulação do sistema no CIF3.

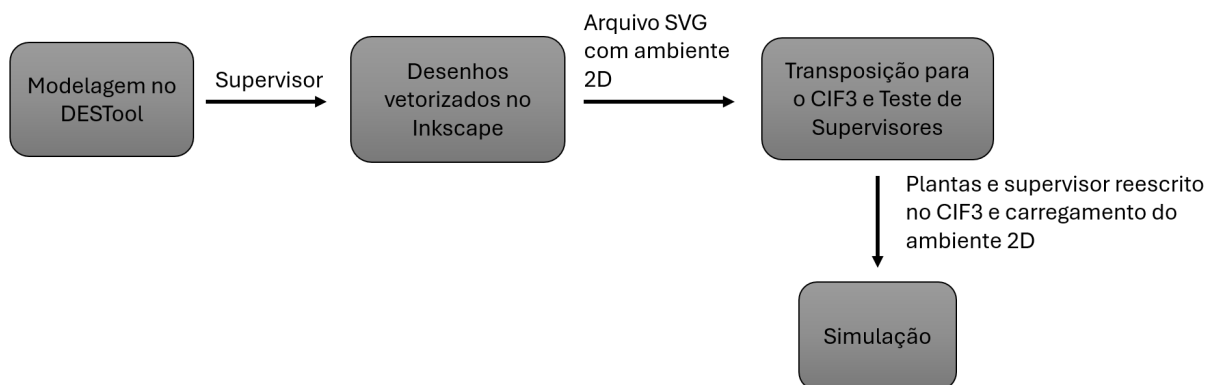


Fonte: Oliveira (2021).

Os passos seguidos para a realização da simulação estão ilustrados no diagrama da Figura 19 e são descritos a seguir:

- Modelagem no DEStool: Criação dos modelos de Sistemas a Eventos Discretos (SED) para o sistema de operação da esteira.
- Desenhos vetorizados no Inkscape: Criação gráfica de cada componente utilizado na simulação em formato vetorizado no software Inkscape.
- Transposição para o CIF3 e Teste de Supervisores: Implementação das plantas para o botão, a esteira e o sensor, com uma especificação para a operação da esteira apenas com o botão pressionado.
- Simulação: Realização da simulação no software CIF3, com interação do usuário através dos eventos controláveis.

Figura 19 – Etapas para realização da simulação no CIF3.

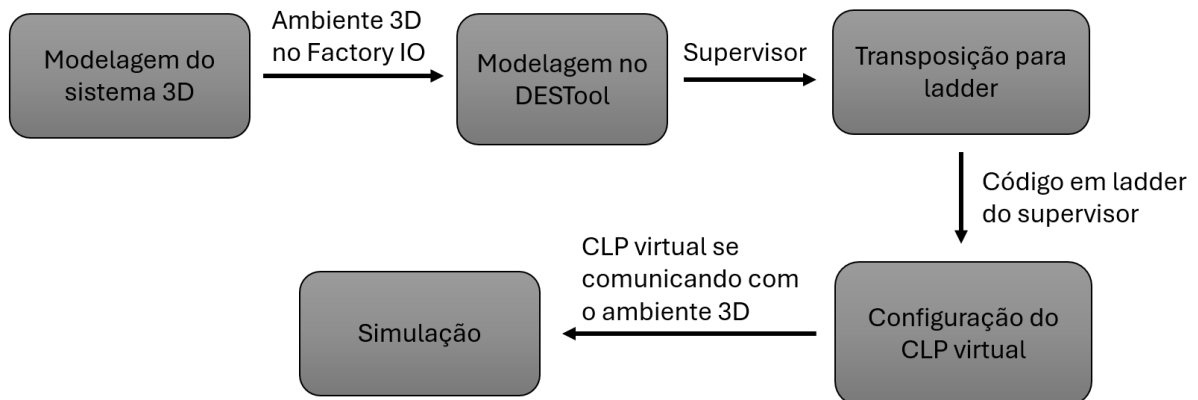


Fonte: Autor (2024).

Para fins de comparação na Figura 20 é possível observar o diagrama referente a simulação do presente trabalho e as descrições das etapas a seguir:

- Modelagem do sistema 3D: No software Factory IO é montada a representação gráfica da simulação a partir de objetos nativos da ferramenta.
- Modelagem no DEStool: Criação dos modelos de Sistemas a Eventos Discretos (SED) para o sistema de operação da esteira.
- Transposição para ladder: Etapa onde foi recriado o supervisor do sistema em ladder no software OpenPLC.
- Configuração do CLP virtual: No OpenPLC Runtime, configura-se a quantidade e o endereçamento das entradas e saídas do CLP. No Factory IO, atribuem-se essas entradas e saídas aos objetos da simulação.
- Simulação: Realização da simulação no software Factory IO, com interação do usuário através dos elementos controláveis.

Figura 20 – Etapas para realização da simulação no Factory IO.



Fonte: Autor (2024).

Na figura 21 têm-se o resultado final do presente trabalho, que se trata da simulação do mesmo sistema utilizado na simulação do trabalho citado de Oliveira (2021), no software Factory IO.



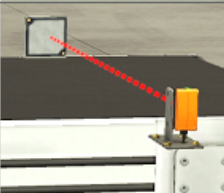
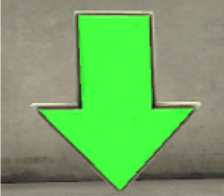


Figura 21 – Simulação do sistema no Factory IO.



Fonte: Autor (2024).

A descrição dos elementos utilizados na construção da simulação está apresentada na Figura 22. Esta figura é composta por uma tabela que organiza as informações em três colunas distintas, a primeira coluna, intitulada 'Imagem', exibe visualmente a imagem de cada elemento, a segunda coluna, denominada 'ID', fornece a identificação textual (ID) de cada elemento e a terceira coluna, intitulada 'Descrição', contém uma breve descrição de cada elemento. Essa tabela é similar à descrição do editor XML para o desenho vetorizado apresentada por Oliveira (2021).

Figura 22 – Partes individuais da simulação.

Imagem	ID	Descrição
	Botao	Botão liga/deliga para operar a esteira.
	Display	Display que mostra a quantidade de caixas passadas na esteira.
	Sensor	Sensor para detectar a presença da caixa.
	Emissor	Emissor que gera uma nova caixa a cada 15 segundos
	Esteira	Estrutura da esteira
	Caixa	Caixa que percorre a esteira

Fonte: Autor (2024).

4.5 Dificuldades encontradas

Apesar de o Eclipse ESCET oferecer suporte à exportação de autômatos para o formato de arquivo XML, foi encontrada uma incompatibilidade direta com o OpenPLC, o que impossibilitou a importação automática dos autômatos criados. Os simuladores CODESYS e TwinCAT, apesar de suportarem o formato XML, não foram capazes de replicar integralmente o comportamento dos autômatos na linguagem ladder, resultando em discrepâncias na funcionalidade simulada. Diante dessas limitações, optou-se pela transcrição manual dos autômatos do formato CIF para a linguagem ladder. Embora mais trabalhosa, essa abordagem demonstrou-se viável para manter a integridade e

funcionalidade dos sistemas simulados. A escolha pelo OpenPLC foi motivada pela sua interface intuitiva e pela eficiente integração com o simulador fabril Factory IO, utilizando o protocolo Modbus TCP para comunicação.

O OpenPLC é reconhecido por sua capacidade de transformar hardware simples em controladores lógicos programáveis completos, utilizando-se de uma variedade de protocolos de comunicação, incluindo Modbus TCP, o que facilita a interação com sistemas de automação industrial. Essa flexibilidade e adaptabilidade tornaram o OpenPLC uma escolha ideal para este estudo, permitindo uma integração eficiente entre o software de supervisão e o ambiente de simulação industrial. Essa abordagem superou as limitações encontradas com a exportação automática de arquivos do ESCET, e também proporcionou uma solução funcional para simulação de sistemas industriais complexos. O trabalho continua explorando a integração entre softwares de supervisão e simuladores de CLP, visando otimizar os processos de desenvolvimento e simulação na indústria.

5 CONCLUSÕES

Este trabalho explorou o uso de Controladores Lógicos Programáveis (CLPs) e a linguagem ladder no controle supervísório de Sistemas a Eventos Discretos (SEDs) em ambientes industriais. A simulação de um processo utilizando o software CIF3 e o Factory IO demonstrou a aplicação prática dessas ferramentas na automação.

O estudo de caso, focado na simulação de uma esteira transportadora, permitiu avaliar a integração dos componentes do sistema e a capacidade de resposta dos controladores às mudanças de estado dos processos. Os resultados indicaram que o uso do CIF3, juntamente com o OpenPLC, oferece uma plataforma adequada para o desenvolvimento e teste de estratégias de controle, proporcionando um ambiente virtual seguro para experimentação.

As simulações facilitaram o processo de programação dos CLPs e melhoraram a compreensão sobre o funcionamento dos sistemas de controle, resultando em maior eficiência operacional e redução de erros durante a implementação. Além disso, a pesquisa evidenciou a importância da validação de modelos e do uso de simulações como ferramentas educacionais, auxiliando no aprendizado de conceitos de automação e controle.

Este trabalho destaca-se pela proposta de uso de simulações mais realistas em comparação ao estudo de Oliveira (2021). A incorporação da linguagem ladder e o uso de ferramentas como OpenPLC e Factory IO proporcionam uma representação mais precisa dos processos industriais. Essas ferramentas permitem aos usuários visualizar e interagir com os sistemas em um ambiente seguro e controlado, facilitando a aplicação prática dos conceitos de controle supervísório em cenários industriais.

A abordagem adotada também contribui para fins didáticos, oferecendo aos estudantes uma experiência prática e segura, essencial para a compreensão e retenção dos conhecimentos em automação industrial. Esta metodologia valida a eficácia das soluções propostas e reforça o aprendizado prático na área de automação industrial.

Como o arquivo de exportação do supervisor do ESCET é um arquivo no padrão XML, uma sugestão é desenvolver métodos automáticos para a transformação da exportação para um padrão compatível com o OpenPLC. Sendo um arquivo XML estruturado por meio de tags que definem elementos e atributos específicos, como estados e eventos, ele pode ser interpretado por scripts ou ferramentas de software para extrair as informações necessárias e reformatá-las de acordo com os requisitos de outro sistema, como o OpenPLC. Automatizando esse processo, seria possível

traduzir os modelos de sistemas a eventos discretos para a linguagem ladder de forma rápida, reduzindo o tempo de desenvolvimento e a possibilidade de erros na transcrição manual.

REFERÊNCIAS

- AFENU, D. S.; ASIRI, M.; SAXENA, N. Industrial control systems security validation based on mitre adversarial tactics, techniques, and common knowledge framework. **Electronics**, MDPI, v. 13, n. 5, p. 917, 2024. Disponível em: <<https://www.mdpi.com/2079-9292/13/5/917>>.
- ALESSI, S. M.; TROLLIP, S. R. **Multimedia for Learning: Methods and Development**. [S.l.]: Allyn & Bacon, 2001. 197-199 p.
- ALSABBAGH, W.; KIM, C.; LANGENDÖRFER, P. Investigating the security of openplc: Vulnerabilities, attacks, and mitigation solutions. **IEEE Access**, 2023. Disponível em: <https://www.researchgate.net/publication/373862064_Investigating_the_Security_of_OpenPLC_Vulnerabilities_Attacks_and_Mitigation_Solutions>.
- ALVES J., F. M. . C. P. Integration of openplc and factory io for industrial automation simulations. **Journal of Industrial Automation**, p. 123–134, 2020.
- BAIER, C.; KATOEN, J. **Principles of Model Checking**. [S.l.]: MIT Press, 2008.
- BANKS, J. et al. **Discrete-Event System Simulation**. 4th. ed. [S.l.]: Pearson, 2005. 54-55 p.
- BEMPORAD, A.; GIUA, A.; RUBAGOTTI, M. **Supervisory Control of Discrete Event Systems Using Linear Programming**. [S.l.]: Springer, 1999.
- BOLTON, W. **Programmable Logic Controllers**. [S.l.]: Newnes, 2015.
- BRUCKNER, S.; DORNER, K.; ERNST, R. **Model-based Engineering of Embedded Systems: The SPES Perspective**. [S.l.]: Springer Science Business Media, 2009.
- BRYAN, L. A.; BRYAN, E. A. **Programmable Controllers: Theory and Implementation**. [S.l.]: Industrial Text Video Company, 1997.
- CASSANDRAS, C. G.; LAFORTUNE, S. **Introduction to Discrete Event Systems**. [S.l.]: Springer, 1999.
- CRUZ, L. L. D. **Metodologia para Implementação de Controle Supervisório Modular Local em Controladores Lógicos Programáveis**. 2011. Acesso em: 20 jun. 2024. Disponível em: <https://www.researchgate.net/publication/281004189_METODOLOGIA_PARA_IMPLEMENTACAO_DE_CONTROLE_SUPERVISORIO_MODULAR_LOCAL_EM_CONTROLADORES_LOGICOS_PROGRAMAVEIS>.
- FELDER, R. M.; BRENT, R. The intellectual development of science and engineering students: Part 1, models and challenges. **Journal of Engineering Education**, v. 93, n. 4, p. 269–277, 2004.

GOMES, T.; VELOSO, M. **Event-based Control and Signal Processing: Principles of Event-based Control Systems**. [S.l.]: CRC Press, 2019.

GUERRA, M. et al. Teaching automation with factory i/o under a competency-based curriculum. **Multimedia Tools and Applications**, v. 81, n. 2, p. 12863–12886, 2022. Disponível em: <<https://link.springer.com/article/10.1007/s11042-022-12863-8>>.

INTERNATIONAL ELECTROTECHNICAL COMMISSION. **IEC 61131-3: Programming Languages**. Geneva, Switzerland, 2003.

JOHN, K. H.; TIEGELKAMP, M. **IEC 61131–3: Programming Industrial Automation Systems**. [S.l.]: Springer, 2010.

LAFORTUNE, S. **Introduction to Discrete Event Systems**. [S.l.]: Springer, 2008.

LEWIS, R. W. **Programming Industrial Control Systems Using IEC 1131-3**. [S.l.]: IET, 1998.

MURRAY, R. M.; SASTRY, S. S.; ZEXIANG, L. **A Mathematical Introduction to Robotic Manipulation**. [S.l.]: CRC Press, 1994.

OLIVEIRA, B. d. **Avaliação do software CIF3 para projetos e simulação de sistemas a eventos discretos**. 2021. Acesso em: 13 jun. 2024. Disponível em: <<https://repositorio.ufsc.br/handle/123456789/228297>>.

ORGANIZATION, M. **Modbus Messaging on TCP/IP Implementation Guide V1.0b**. 2006. Acesso em: 29 jun. 2024. Disponível em: <http://www.modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b.pdf>.

PRENSKY, M. Digital game-based learning. **McGraw-Hill**, p. 106–110, 2001.

RAMADGE, P. J.; WONHAM, W. M. **Supervisory Control of a Class of Discrete Event Processes**. [S.l.]: SIAM Journal on Control and Optimization, 1989.

SALAH, B. et al. 3d simulation of a yogurt filling machine using grafcet studio and factory io: Realization of industry 4.0. **Transactions of FAMENA**, v. 47, n. 3, p. 15–30, 2023. Disponível em: <<https://hrcak.srce.hr/en/clanak/441658>>.

SIEMENS, G. **Connectivism: A Learning Theory for the Digital Age**. [S.l.: s.n.], 2005. v. 2. 3-10 p.

SIEMENS, G. **SIMATIC Modbus/TCP Communication Manual**. 2018. Acesso em: 27 jun. 2024. Disponível em: <<https://support.industry.siemens.com/cs/document/109742748/simatic-modbus-tcp-communication>>.

WONHAM, W. M. **Supervisory Control of Discrete-Event Systems**. [S.l.]: Springer, 2011.