

**CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS
GERAIS CAMPUS TIMÓTEO**

Luiz Eduardo Langkammer Murta

**Sistema de simulação para identificação e separação de
produtos utilizando CODESYS e Plataforma Factory IO**

Timóteo

2023

Luiz Eduardo Langkammer Murta

Sistema de simulação para identificação e separação de produtos utilizando CODESYS e Plataforma Factory IO

Monografia apresentada à Coordenação de Engenharia de Computação do Campus Timóteo do Centro Federal de Educação Tecnológica de Minas Gerais para obtenção do grau de Bacharel em Engenharia de Computação.

Orientador: Dr. Elder de Oliveira Rodrigues

Timóteo

2023

Lulz Eduardo Langkammer Murta

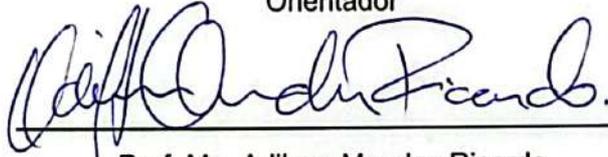
Sistema de simulação para identificação e separação de produtos utilizando CODESYS e Plataforma Factory IO

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Computação do Centro Federal de Educação Tecnológica de Minas Gerais, campus Timóteo, como requisito parcial para obtenção do título de Engenheiro de Computação.

Trabalho aprovado. Timóteo, 27 de novembro de 2023:



Prof. Dr. Elder de Oliveira Rodrigues
Orientador



Prof. Me. Adilson Mendes Ricardo
Professor Convidado



Prof. Dra. Viviane Cota Silva
Professora Convidada

Timóteo

2023

Agradecimentos

Agradeço primeiramente a Deus por abençoar e me guiar rumo aos meus objetivos. Aos meus pais e irmã que me apoiaram, me deram forças e condições para continuar sempre estudando e me dedicando. Aos meus amigos que estiveram comigo nesta jornada compartilhando experiência e se ajudando a superar os desafios no caminho, em especial o Arthur que esteve comigo e me motivou a seguir adiante. Ao meu amor Dayane, que sempre me motivou, me orientou e me deu forças para seguir em toda esta jornada. E por último, mas não menos importante, ao professor Elder que acreditou e confiou no meu trabalho desde o princípio, me orientou e me forneceu os conhecimentos necessários para alcançar meu objetivo final neste trabalho.

Resumo

A automação industrial vem desempenhando cada dia mais um papel crucial no aprimoramento do processo de fabricação para que, dessa forma, seja possível a produção de maneira mais rápida e mais barata, ou seja, maximizando a eficiência. Inicialmente focada na atuação em pequenas partes de um sistema, hoje ela é capaz de atuar em processos completos e mais complexos. O CLP (Controlador Lógico Programável) é um elemento de grande importância nesses sistemas, uma vez que é utilizado para efetuar o controle de sistemas focados no ambiente industrial possuindo não só uma maior robustez para suportar as intempéries típicas destes locais, como também recursos técnicos para a interação com os mais diversos sensores, atuadores, sistemas supervisórios e IHMs (Interface Homem-Máquina). No entanto, colocar em prática os conhecimentos a respeito do uso destes equipamentos pode se mostrar um desafio, uma vez que os equipamentos envolvidos possuem um valor elevado de custo para aquisição. Assim, este trabalho tem como objetivo desenvolver um sistema de automação industrial simulado para identificação, contagem e separação de caixas por tamanho ou cor. O sistema proposto utiliza do *software* Factory IO para construção e simulação do ambiente industrial e é controlado por um CLP virtual disponibilizado no CODESYS (*Controller Development System*), juntamente com uma IHM desenvolvida e executada no mesmo *software*, a comunicação entre estes elementos se dá por meio do protocolo OPC (*Open Platform Communications*). O projeto do sistema inclui a arquitetura geral, a seleção dos sensores adequados, a programação Ladder do CLP e o desenho da IHM. Após o desenvolvimento do projeto foi possível contar com um ambiente configurado no Factory IO mediante o objetivo do trabalho bem como uma lógica de controle e IHM alinhados para correto funcionamento do sistema. Assim, foi permitida a execução de uma simulação envolvendo a geração aleatória de uma série de produtos para identificação e separação que foram posteriormente validados através da correta contagem apresentada pela IHM e o visual no Factory IO que apresentava cada elemento em seu lugar ao fim do processo. O resultado da simulação feita indica o sucesso na comunicação entre simulador e CLP virtual através do protocolo OPC e no processo proposto através das observações feitas ao fim da contagem e separação dos produtos, com possibilidades de incremento quanto ao uso de um CLP real, como o S7-1200 da Siemens.

Palavras-chave: CLP, Factory IO, CODESYS, Ladder, IHM.

Abstract

Industrial automation has been playing an increasingly crucial role in enhancing the manufacturing process, making it possible to produce faster and more cost-effectively, thereby maximizing efficiency. Initially focused on operating in small parts of a system, it can now handle complete and more complex processes. The PLC (Programmable Logic Controller) is a prevalent element in this context, as it is used to control systems in the industrial environment. It not only provides greater robustness to withstand the typical harsh conditions of these locations but also offers technical features for interaction with various sensors, actuators, supervisory systems, and HMIs (Human Machine Interface). However, putting knowledge into practice regarding the use of these devices can be challenging, given the high cost of acquiring the involved equipment. Therefore, the objective of this work is to develop a simulated industrial automation system for the identification, counting, and sorting of boxes by size or color. The proposed system uses Factory IO software for the construction and simulation of the industrial environment and is controlled by a virtual PLC provided in CODESYS (*Controller Development System*), along with an HMI developed and executed in the same software. The communication between these elements occurs through the OPC (*Open Platform Communications*) protocol. The system project includes the overall architecture, selection of suitable sensors, ladder programming of the PLC, and the design of the HMI. After project development, a configured environment in Factory IO was achieved according to the work's objective, along with control logic and an HMI aligned for the correct operation of the system. Thus, a simulation involving the random generation of a series of products for identification and separation was executed. These results were subsequently validated through the correct counting displayed by the HMI and the visual representation in Factory IO, which showed each element in its place at the end of the process. The simulation results indicate successful communication between the simulator and the virtual PLC through the OPC protocol. The proposed process, as observed at the end of product counting and separation, was successful, with possibilities for improvement with the use of a real PLC, such as the Siemens S7-1200.

Keywords: PLC, Factory IO, CODESYS, Ladder, IHM.

Lista de Figuras

Figura 1 - Estrutura básica CLP.....	10
Figura 2 - Ciclo de funcionamento CLP.....	10
Figura 3 - Relação entre função e símbolo em ladder.....	13
Figura 4 - Função lógica “AND” em Ladder.....	14
Figura 5 - Função lógica “OR” em Ladder.....	14
Figura 6 - Função lógica “NOT” em Ladder.....	15
Figura 7 - Função lógica “XOR” em Ladder.....	15
Figura 8 - Exemplo de Bloco de CTU no CODESYS.....	16
Figura 9 - Exemplo de Bloco de CTD no CODESYS.....	16
Figura 10 - Janela de instalação personalizada do CODESYS.....	17
Figura 11 - Exemplo de IHM.....	19
Figura 12 - Exemplo de simulação montada no Factory IO.....	20
Figura 13 - Seleção de drivers de comunicação no Factory IO.....	21
Figura 14 - Configuração da conexão OPC UA no Factory IO.....	21
Figura 15 - Sensor óptico.....	22
Figura 16 - Sensor óptico difuso no Factory IO.....	23
Figura 17 - Sensor de visão no Factory IO.....	23
Figura 18 - Relação entre objeto e sinal gerado pelo sensor de visão no Factory IO...24	
Figura 19 - Sensor Reflexivo no Factory IO.....	25
Figura 20 - Classificador de Rodas Pop-Up no Factory IO.....	26
Figura 21 - Classificador de Braço Articulado no Factory IO.....	27
Figura 22 - Emissor no Factory IO.....	27
Figura 23 - Cenário para identificação das caixas no Factory IO.....	28
Figura 24 - Relatório de carga em Excel.....	29
Figura 25 - Fluxograma da estação de transporte.....	30
Figura 26 - Fluxograma da estação de classificação.....	30
Figura 27 - Esquema do ambiente a ser controlado.....	31
Figura 28 - Maquete com as conexões com o CLP.....	32
Figura 29 - Fluxograma metodológico do projeto.....	34
Figura 30 - Tipos de caixa diferentes no tamanho.....	35
Figura 31 - Tipos de matéria prima diferentes na cor.....	36
Figura 32 - Visualização do Conjunto 1 e Conjunto 2 lado a lado.....	37
Figura 33 - Disposição dos sensores difusos ópticos no Factory IO.....	38
Figura 34 - Passagem do Produto G no portal de identificação no Factory IO.....	38
Figura 35 - Posição do sensor de visão no portal de identificação no Factory IO.....	39
Figura 36 - Separação por Classificador de Braço Articulado no Factory IO.....	40
Figura 37 - Separação por Classificador de Rodas Pop-Up no Factory IO.....	40
Figura 38 - Sistema completo montado no Factory IO.....	41
Figura 39 - IHM montada no CODESYS.....	43

Figura 40 - Lógica de identificação do Produto G em Ladder.....	48
Figura 41 - Lógica de identificação do Produto ML em Ladder.....	49
Figura 42 - Lógica de identificação do Produto M em Ladder.....	49
Figura 43 - Lógica de identificação do Produto P em Ladder.....	49
Figura 44 - Lógica de identificação e atuação para o Produto G em Ladder.....	50
Figura 45 - Lógica de reset de atuação para o Conjunto 1 em Ladder.....	51
Figura 46 - Lógica de identificação do Produto Azul em Ladder.....	51
Figura 47 - Lógica de identificação do Produto Verde em Ladder.....	51
Figura 48 - Lógica de identificação do Produto Cinza em Ladder.....	52
Figura 49 - Lógica de identificação e atuação para o Conjunto 2 em Ladder.....	53
Figura 50 - Lógica de reset de atuação para o Conjunto 2 em Ladder.....	54
Figura 51 - IHM no CODESYS ao iniciar código no CLP.....	55
Figura 52 - Factory IO após início de uma simulação.....	56
Figura 53 - Factory IO durante separação do primeiro produto.....	57
Figura 54 - IHM durante separação do primeiro produto.....	57
Figura 55 - Factory IO durante separação do Produto Cinza.....	58
Figura 56 - Factory IO após paralisação das esteiras.....	59
Figura 57 - IHM após paralisação das esteiras.....	59

Lista de Tabelas

Tabela 1 - Sinais de entrada para uso no Factory IO e CODESYS.....	43
Tabela 2 - Sinais de saída para uso no Factory IO e CODESYS.....	45

Lista de Abreviaturas

CLP	Controlador lógico programável
CODESYS	<i>Controller Development System</i>
CTD	<i>Counter Down</i>
CTU	<i>Counter Up</i>
HMI	<i>Human Machine Interface</i>
IHM	Interface Homem Máquina
NA	Normalmente aberto
NF	Normalmente fechado
IDE	Ambiente de Desenvolvimento Integrado
IEC	<i>International Electrotechnical Commission</i>
OLE	<i>Object Linking and Embedding</i>
OPC	<i>Open Platform Communication</i>
OPC UA	<i>Open Platform Communication Unified Architecture</i>
PLC	<i>Programmable Logic Controller</i>
UCP	Unidade Central de Processamento

Sumário

1 Introdução	7
1.1 Problema	7
1.2 Justificativa	8
1.2 Objetivos	8
2 Fundamentação teórica	9
2.1 CLP	9
2.1.1 Entradas/Saídas	11
2.1.2 Linguagem	12
2.1.3 Programação	16
2.2 Protocolo OPC	18
2.3 IHM	18
2.4 Factory IO	19
2.4.1 Sensor Difuso Óptico	21
2.4.2 Sensor De Visão	22
2.4.3 Sensor Reflexivo	24
2.4.4 Classificador De Rodas Pop-Up	24
2.4.5 Classificador De Braço Articulado	25
2.4.6 Emissor	26
3 Trabalhos relacionados	28
4 Procedimentos metodológicos	33
4.1 Métodos e Materiais	33
5 Desenvolvendo o sistema de controle	35
5.1 Construção do ambiente industrial	35
5.1.1 Mecanismos de identificação	35
5.1.2 Mecanismos de separação	39
5.1.3 Sistema completo	41
5.2 Construção IHM	42
5.2.1 Visualização da contagem	43
5.2.2 Visualização da identificação	43
5.2.3 Botões de ação	44
5.3 Lógica de controle	44
5.3.1 Identificação e separação por tamanho	49
5.3.2 Identificação e separação por cor	51
5.3.3 Ações da IHM	54
6 Resultados	56
7 Considerações finais	61
7.1 Trabalhos futuros	61
Referências Bibliográficas	62

1 Introdução

A automação industrial desempenha um papel fundamental no aprimoramento dos processos de fabricação, permitindo maior eficiência, produtividade e qualidade na indústria moderna. Segundo Silveira (2011, *online*), “O objetivo principal da automação industrial é criar mecanismos que sejam capazes de produzir o melhor produto com o menor custo”.

Os sistemas automatizados inicialmente focavam em apenas sistemas fechados e atuavam sobre eles de forma isolada dos demais processos. Com a evolução ao longo dos anos passou de sistemas individuais para sistemas abertos que abrangem múltiplos processos, otimizando o funcionamento de toda uma planta. De modo geral, um sistema automatizado é composto por duas partes principais: a parte operacional e a parte de controle.

A parte operacional da automação industrial é responsável por atuar diretamente nos processos. Envolvendo elementos como dispositivos de acionamento e pré-acionamento, tais como motores, cilindros, compressores, válvulas e pistões, além de dispositivos de detecção, como sensores indutivos, capacitivos, de visão e ultrassônicos. Esses componentes permitem o movimento e a operação das máquinas de acordo com o objetivo desejado.

Por outro lado, a parte de controle é a parte programável do sistema, normalmente implementada por meio de um CLP. No passado, a lógica de controle era feita utilizando relés eletromagnéticos, temporizadores, placas eletrônicas e módulos lógicos. No entanto, com o avanço dos componentes eletrônicos e o aumento do volume de dados, tornou-se comum o uso de CLPs e computadores industriais para o controle de máquinas e processos (SILVEIRA, 2011).

1.1 Problema

A automação industrial conta com diversos desafios clássicos de automatização, sendo um deles a identificação, contagem e separação eficiente de caixas ou produtos. Essa tarefa, quando realizada manualmente, pode ser suscetível a erros humanos, além de ser demorada e exigir recursos significativos.

O desafio central a ser abordado é a criação de algoritmos e estratégias de controle eficientes que permitam ao sistema identificar, contar e separar as caixas de acordo com seu tamanho ou cor, utilizando os dados obtidos pelos sensores disponíveis no Factory IO. É necessário garantir a precisão na identificação das características das caixas e a correta atuação dos dispositivos de acionamento para a separação adequada.

Outro aspecto dos problemas atacados pelo projeto está relacionado ao fato de que colocar em prática os estudos sobre CLP pode ser um desafio devido à sua natureza industrial e aos altos custos dos equipamentos necessários para o controle. Assim, não somente a obtenção do CLP pode ser complicada para projetos acadêmicos como também os sensores e atuadores compartilham das mesmas características industriais.

Portanto, a presente monografia tem como base as seguintes perguntas: Como colocar em prática os estudos sobre CLP uma vez que sua natureza industrial exige componentes de valor elevado para montagem de um sistema de controle real? Como implementar um sistema de controle para identificação e separação de produtos em ambiente industrial simulado?

1.2 Justificativa

A presente monografia tem como justificativa a oportunidade de integração entre um CLP simulado e um *software* de simulação gráfica robusto como o Factory IO, realizando o controle de um processo importante na indústria. Além de retratar tal processo de identificação e separação de produtos, é possível sintetizar o conhecimento do uso do CLP de uma maneira visual sem grandes custos com sensores e demais equipamentos físicos fornecidos pelo simulador.

1.2 Objetivos

Este trabalho de conclusão de curso tem como objetivo desenvolver e implementar um sistema de identificação e contagem de produtos por tamanho ou cor, seguido de sua separação automatizada. O controle partirá de um CLP virtual e atuará sobre um ambiente gerado no *software* Factory IO, além de contar com uma IHM para acompanhamento e controle da simulação.

Também objetivam-se mais especificamente:

1. Definir um modelo de arquitetura para o controle.
2. Criar lógica de identificação e contagem dos produtos por tamanho ou cor no CODESYS.
3. Criar lógica de separação dos produtos em esteiras no CODESYS.
4. Montar uma representação visual do processo físico no *software* Factory IO.
5. Desenvolver IHM para visualização e controle do processo físico no CODESYS.
6. Executar simulação completa com os elementos desenvolvidos e integrados.

2 Fundamentação teórica

Visando alcançar o objetivo de desenvolver um sistema de identificação e separação de caixas por tamanho ou cor, por meio do uso do *software* de simulação industrial Factory IO e um CLP virtual, serão apresentados os conceitos teóricos necessários para entendimento e construção deste.

2.1 CLP

O CLP, do inglês PLC (*Programmable Logic Controller*) é um dispositivo de controle muito utilizado no ambiente industrial. Ele consiste em um aparelho eletrônico digital que possui uma memória interna destinada à persistência de blocos de instrução de programas bem como de variáveis internas necessárias para a lógica de controle (ZANCAN, 2011).

Segundo Cravo (2023b, *online*), “O primeiro projeto de CLP foi desenvolvido pela General Motors, na década de 60, nos Estados Unidos. Buscava-se uma alternativa ao uso dos painéis de relés, que eram amplamente usados para controle”. Diferentemente desta solução antiga que contava com substituição ou reorganização de partes físicas para alteração da lógica de controle, o CLP é capaz de contar com tais alterações por meio de *software*. Assim, diminui-se o custo e o tempo para programação do controle, permitindo maior flexibilidade.

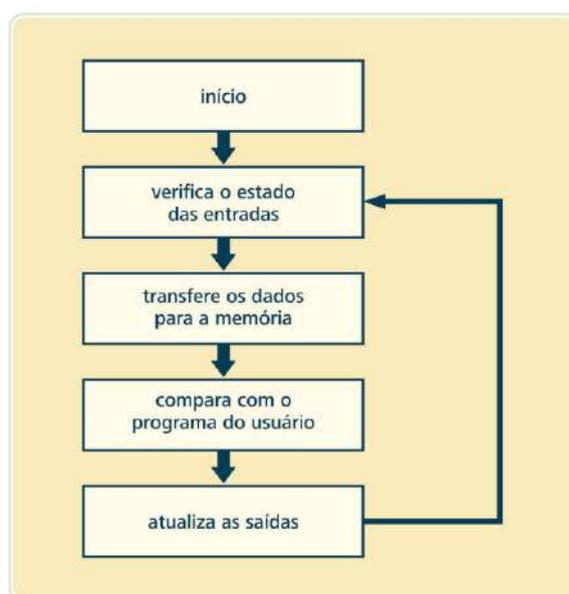
Para a realização do controle de um processo, o CLP conta com uma estrutura básica de componentes que é constante dentre as diversas fabricantes do mercado. Pode-se considerar em um nível mais alto as entradas que levam à unidade central de processamento que por sua vez termina nas saídas. A Figura 1 ilustra o fluxo citado.

Nas entradas do CLP são conectados sensores capazes de medir as grandezas físicas envolvidas no sistema para que estas encaminhem os dados à Unidade Central de Processamento (UCP). Uma vez na UCP, esta irá processar as informações conforme a lógica do programa atualmente em execução, gerando resultados encaminhados a um conjunto de saídas. Ligados a estas saídas temos os elementos atuadores que irão interagir com o sistema de forma ativa. Dentre os atuadores podemos citar bobinas, motores, LEDs, esteiras (ZANCAN, 2011).

Figura 1 - Estrutura básica CLP

Fonte: Autor

A Figura 2 mostra um diagrama do processo de funcionamento de um CLP durante a sua execução, podendo ser possível identificar alguns passos sequenciais que são seguidos e repetidos constantemente.

Figura 2 - Ciclo de funcionamento CLP

Fonte: (ZANCAN, 2011)

No passo determinado como início o CLP faz uma verificação do estado de seus componentes internos, da presença de um programa na memória e autoriza a execução do programa caso exista.

O ciclo de execução começa com uma verificação do estado das entradas do CLP, avaliando suas tensões para interpretação dos valores atuais. Logo em seguida os dados lidos das entradas são persistidos na memória de dados interna para posterior consulta do CLP.

Uma vez com os dados em memória, a lógica programada pode enfim atuar e processar as informações das entradas lidas e armazenadas gerando enfim um conjunto de valores que são direcionados às suas respectivas saídas. Assim, o processo voltará para a leitura das entradas para que o ciclo continue (ZANCAN, 2011).

2.1.1 Entradas/Saídas

Como já discutido anteriormente, duas partes básicas de um CLP e que garantem sua interface com o mundo real são as suas entradas e saídas e por isso discutiremos aqui as suas definições mais profundamente. Estas partes ditam a maneira como o CLP irá interpretar o sistema que deseja controlar por meio das entradas, e ditam a maneira como ele irá atuar sobre o sistema por meio das saídas para alcançar o resultado desejado.

Inicialmente concebido para o controle de variáveis discretas, o CLP adquiriu capacidades expandidas, incluindo o controle de variáveis analógicas, o que lhe permite comandar servomecanismos e máquinas multiferramentais de maneira precisa. Isso não apenas ampliou sua gama de aplicações, mas também contribuiu para a redução de custos de fiação e instalação nas indústrias (EGÍDIO *et. al*, 2011).

Levando em consideração tal evolução, podemos dividir as entradas e saídas em duas categorias sendo elas entradas/saídas digitais e entradas/saídas analógicas.

As entradas digitais são capazes de identificar a presença ou não de um sinal não se importando com sua amplitude

Apesar das variáveis físicas, tais como temperatura, pressão, força, massa, etc, terem comportamento analógico, a maioria dos processos são controlados através de informações digitais, provindas de sensores, botoeiras, chaves fim de curso, termostatos, pressostatos, etc, tornando as entradas digitais as mais presentes e as mais utilizadas em CLPs (ZANCAN, 2011, p.23).

Por outro lado, as entradas analógicas levam em consideração a amplitude do sinal recebido salvando em memória, como no fluxo anteriormente descrito, o valor do exato momento da leitura. Assim, garante uma leitura mais precisa, utilizada geralmente para sensores de temperatura e pressão, por exemplo (ZANCAN, 2011).

As saídas digitais, assim como as entradas trabalham com a presença ou não de um sinal, neste caso o acionamento ou não da saída, resultando em uma atuação por parte do dispositivo atuador ligado a ela. Assim, conforme a lógica de controle, uma saída possuirá um valor de tensão condicional.

Por fim, as saídas analógicas de um CLP geralmente são utilizadas devido à necessidade de maior precisão na atuação.

Os sinais elétricos das saídas analógicas poderão ser de tensão ou corrente, cuja faixa de valores mais utilizada é, respectivamente, de 0 a 10 Vcc e 4 mA a 20 mA. Dessa forma, os atuadores receberão das saídas analógicas sinais elétricos variáveis, não apenas energizando os equipamentos, mas, principalmente, definindo a intensidade de sua atuação no processo (ZANCAN, 2011, p.26).

Como exemplo podemos ter o controle de velocidade de uma esteira podendo ser acionada para movimentação mais rápida ou mais lenta dependendo da necessidade.

2.1.2 Linguagem

Uma das principais vantagens do CLP sobre o uso de painéis de relés é a sua facilidade de programação, pois a mudança de sua lógica de controle é realizada a nível de *software*, enquanto no segundo são necessárias trocas ou reorganização de componentes físicos. Para isso o CLP conta com algumas linguagens de programação.

Em consonância com a norma IEC 61131-3, que estabelece um conjunto de linguagens "padronizadas" para programar CLPs, as linguagens mais comuns incluem o Histograma de Contatos (*Ladder Diagram*) e outras, como o Diagrama de Blocos Funcionais (*Functional Block Diagram*), o Mapa de Sequência Funcional (*Sequential Function Chart*), o Texto Estruturado (*Structured Text*) e a Lista de Instruções (*Instruction List*). Esta norma tem como objetivo promover a padronização entre diferentes fornecedores de CLPs, permitindo uma maior portabilidade de programas entre várias plataformas (EGÍDIO *et. al*, 2011).

No contexto deste trabalho, focaremos especificamente na linguagem *Ladder Diagram*, ou somente Ladder, que é uma das opções oferecidas pela norma IEC 61131-3 e foi escolhida para o desenvolvimento do sistema devido a sua ampla utilização e presença em exemplos práticos.

Como consequência da própria maneira antiga de realizar o controle dos processos utilizando os painéis de relé, a linguagem Ladder carrega consigo marcas desse passado. Silveira (2016, *online*) comenta sobre esta relação:

A linguagem ladder foi a primeira linguagem de programação desenvolvida para os CLPs e, como a criação destes foi uma necessidade de substituição do controle de sistemas com relés lógicos, nada mais natural

que a linguagem ladder fosse similar aos diagramas utilizados para documentar a lógica por relês.

Na Figura 3 serão apresentados os principais operadores com os quais a linguagem conta e suas simbologias no desenvolvimento e leitura de um programa.

Figura 3 - Relação entre função e símbolo em ladder

Símbolo	Descrição
	Contato normalmente aberto (NA ou NO). Transfere energia se a chave estiver ligada (fechada).
	Contato normalmente fechado (NF ou NC). Transfere energia se a chave estiver desligada (aberta).
	Transição positiva. Se o estado do símbolo mudar de desligado para ligado, este contato transfere energia na linha até que haja um novo scan do controlador.
	Transição negativa. Se o estado do símbolo mudar de ligado para desligado, este contato transfere energia na linha até que haja um novo scan do controlador.
	Bobina ou Saída. Se todos os contatos na linha transferirem energia, a bobina é energizada (ligada). Caso contrário, ela permanece desligada.
	Bobina ou Saída Negada. Se todos os contatos na linha transferirem energia, a bobina é desenergizada (desligada). Caso contrário, ela permanece ligada.
	Bobina Set. Se a linha transferir energia para este elemento, ele fica energizado, mesmo que a linha deixe de estar energizada.
	Bobina Reset. Se a linha transferir energia para este elemento, ele fica desenergizado e permanece desenergizado mesmo que ocorra alteração de energia na linha.
	Bobina de Detecção Positiva. Se as condições antes desta bobina mudar de desligado para ligado, esta bobina é ligada para um scan do controlador.
	Bobina de Detecção Negativa. Se as condições antes desta bobina mudar de ligado para desligado, esta bobina é ligada para um scan do controlador.
	Bobina de Memória Retentiva. É como a bobina comum exceto pelo fato de que ela retem o seu estado de ligada ou desligada mesmo que o CLP pare ou se desenergize.
	Bobina Set de Memória. Funciona como a bobina set, porém ela retem o seu estado mesmo que o controlador pare ou perca a energia, ficando armazenado na memória.
	Bobina Reset de Memória. Funciona como a bobina reset, porém ela retem o seu estado mesmo que o controlador pare ou perca a energia, ficando armazenado na memória.

Fonte: (SILVEIRA, 2016)

Estes operadores permitem a criação de funções lógicas fundamentais a partir de suas combinações, como "AND", "OR", "XOR" e "NOT". A função "AND" é utilizada para combinar múltiplas condições, exigindo que todas sejam verdadeiras para que a saída seja ativada. Assim para sua montagem, como ilustrado na Figura 4, são adicionados contatos

NA em um mesmo segmento para as entradas Sensor1 e Sensor2 que por fim leva à bobina Atuador1.

Figura 4 - Função lógica "AND" em Ladder



Fonte: (Autor)

A função "OR" ativa a saída se pelo menos uma das condições for verdadeira. Na Figura 5, essa função é montada a partir da adição de contatos NA para as entradas Sensor1 e Sensor2 em segmentos paralelos, assim a ativação de qualquer uma delas resulta na ativação da bobina Atuador1.

Figura 5 - Função lógica "OR" em Ladder



Fonte: (Autor)

A função "NOT" inverte o estado lógico da condição, ativando a saída quando a condição é falsa e vice-versa. Como mostrado na Figura 6, um contato NF é utilizado para fazer esta inversão devido a sua lógica de funcionamento. Assim, o acionamento da bobina Atuador1 somente ocorre quando a entrada Sensor1 estiver em nível lógico baixo.

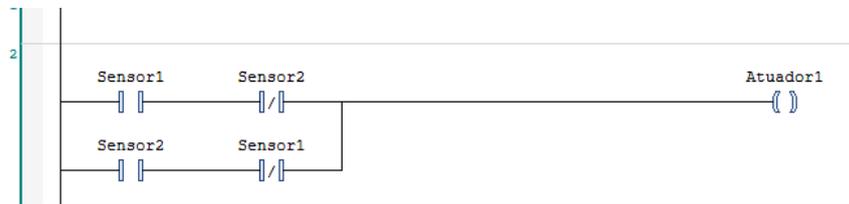
Figura 6 - Função lógica "NOT" em Ladder



Fonte: (Autor)

Por fim a função "XOR" ativa a saída se uma, e apenas uma, das condições for verdadeira. Para isso, como mostrado na Figura 7, utilizamos de uma combinação das funções "AND", "OR" e "NOT" para produzir o efeito desejado.

Figura 7 - Função lógica "XOR" em Ladder



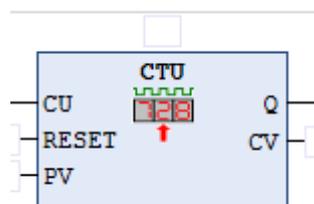
Fonte: (Autor)

A inclusão de blocos funcionais em um diagrama Ladder é uma prática amplamente adotada na programação de CLPs. Essa abordagem se torna especialmente valiosa quando se trata de combinar a lógica sequencial ou combinacional com funções mais complexas que seriam desafiadoras de implementar apenas com elementos de Ladder convencionais. Por exemplo, embora seja possível realizar a contagem de eventos diretamente em Ladder, a utilização de blocos contadores e temporizadores simplifica significativamente a tarefa e permite um controle mais eficaz dos valores de contagem e do tempo decorrido (EGÍDIO *et. al*, 2011).

Levando em consideração o uso de blocos funcionais de contagem, no CODESYS temos disponíveis dois tipos de contador para uso por padrão: CTU (*Counter Up*) e CTD (*Counter Down*).

O CTU, se caracteriza por sua contagem incremental (1, 2, 3...). Figura 8.

Figura 8 - Exemplo de Bloco de CTU no CODESYS



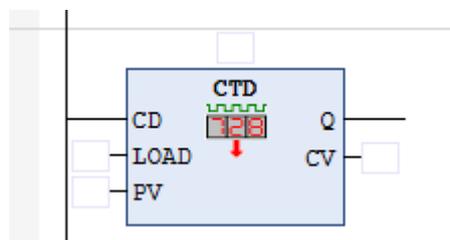
Fonte: Autor

Os impulsos originados na entrada CU desse contador causam incremento na memória interna de contagem que fica disponível na saída CV. A entrada PV por sua vez determina qual o valor objetivo do contador, assim ao o contador atingir este número a

saída Q será acionada. Por fim, ele pode ser reiniciado por meio do acionamento da entrada Reset.

O contador CTD, se caracteriza por sua contagem decrescente(3, 2, 1). Figura 9.

Figura 9 - Exemplo de Bloco de CTD no CODESYS



Fonte: Autor

Os impulsos originados na entrada CD desse contador causam decremento na memória interna de contagem que também fica disponível na saída CV. A entrada PV determina o valor inicial de contagem, assim acionando a saída Q ao atingir o valor 0. Por fim, ele pode ser reiniciado por meio do acionamento da entrada LOAD que volta o contador para o valor PV.

2.1.3 Programação

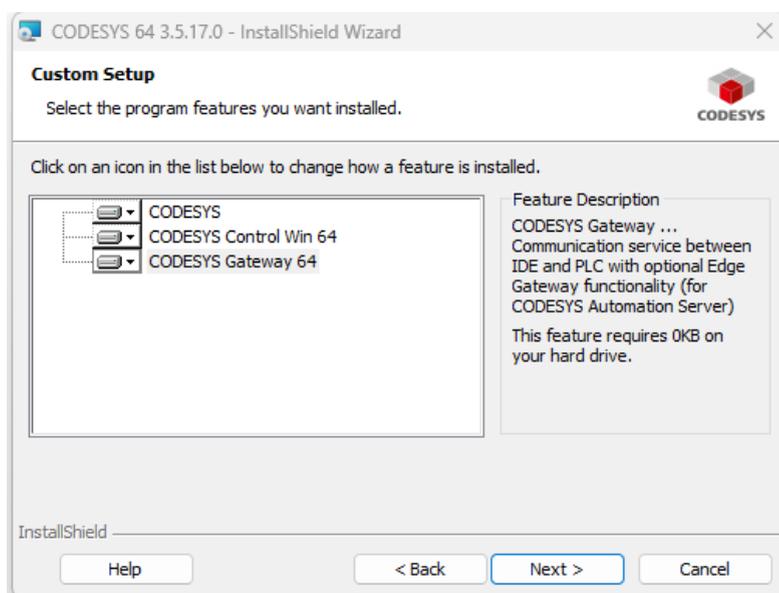
Para realizar a programação em quaisquer que sejam as linguagens escolhidas, compilação e envio do código para um CLP real ou simulado é desejável uma plataforma que permita tais ações de maneira simplificada e iterativa. No mundo dos CLPs contamos com diversas marcas fabricantes e estes por sua vez podem disponibilizar *softwares* próprios para integração de seus produtos.

Fabricantes, como por exemplo a Siemens, possuem *softwares* próprios desenvolvidos e otimizados para a programação e configuração de CLPs da marca, neste caso o Tia Portal. No entanto, estas plataformas não se limitam a apenas proprietários e focados em apenas uma marca, um exemplo é o CODESYS que conta com os recursos acima citados e foi o escolhido para utilização neste trabalho.

O CODESYS é um ambiente de desenvolvimento integrado (IDE) independente de fornecedor de código aberto, projetado para programação de CLPs e sistemas de automação. Ele é amplamente utilizado por uma variedade de fabricantes de CLPs e oferece uma plataforma flexível, versátil e padronizada para programação. O CODESYS suporta múltiplas linguagens de programação e permite a portabilidade de projetos entre diferentes plataformas de hardware (CRAVO, 2023a).

Esse *software* é gratuito, um ponto extremamente relevante na sua escolha para o projeto e possui um processo de configuração inicial bastante simples. Ao iniciar o processo de instalação do CODESYS, adquirido no site da própria empresa desenvolvedora, temos a opção de fazer uma instalação completa ou escolher dentre os recursos que desejamos adquirir. Dentre eles, mostrados na Figura 10, temos a IDE CODESYS, o CLP virtual fornecido na versão demonstrativa CODESYS *Control Win* 64 bits e o serviço de comunicação entre a IDE e o CLP, CODESYS *Gateway* 64.

Figura 10 - Janela de instalação personalizada do CODESYS



Fonte: Autor

O CODESYS *Control Win* é um recurso capaz de transformar um computador em um CLP industrial com uso real e com escalabilidade quase ilimitada dependendo da capacidade da máquina. Este CLP conta com entradas e saídas digitais e analógicas, protocolos de comunicação padrão IEC 61131-3 e sua configuração é feita diretamente na IDE de desenvolvimento mediado pelo CODESYS *Gateway* (CODESYS, 2022a).

Portanto, tendo o CODESYS *Control Win* instalado na máquina é possível ter um CLP virtual em operação com possibilidade de comunicação com softwares como o Factory IO através do protocolo OPC. Vale ressaltar que este recurso não se trata de um simulador e possui suas próprias características de operação.

2.2 Protocolo OPC

A troca de informações diária na indústria exige padrões de comunicação eficazes para garantir a conectividade e o armazenamento de dados de maneira eficiente. Para atender a essa demanda, o protocolo OPC foi desenvolvido como um conjunto de padrões e especificações para comunicação industrial. Baseado no OLE (*Object Linking and Embedding*) da Microsoft, o OPC permite a integração de dados entre equipamentos de chão de fábrica e sistemas de controle, visando à segurança e facilidade (CRAVO, 2023d).

A criação deste protocolo teve início em 1995 quando um consórcio de empresas se uniu para criar um padrão em colaboração com a Microsoft, com o objetivo de possibilitar o acesso a dados em tempo real no sistema operacional Windows. Esse esforço resultou na formação de um grupo sem fins lucrativos, gerenciado pela OPC Foundation (Fonseca, 2002).

Segundo Fonseca (2002, *online*) “Basicamente, o padrão OPC estabelece as regras para que sejam desenvolvidos sistemas com interfaces padrões para comunicação dos dispositivos de campo (CLPs, sensores, balanças, etc.) com sistemas de monitoração, supervisão e gerenciamento (SCADA, MES, ERP, etc.)”.

Com a evolução do protocolo e novos desafios na segurança e modelagem dos dados tratados por ele, a OPC Foundation desenvolveu o OPC UA para atacar estes pontos. Segundo a própria OPC Foundation(2023, *online*) “A Arquitetura Unificada OPC (UA), lançada em 2008, é uma arquitetura orientada a serviços independente de plataforma que integra todas as funcionalidades das especificações individuais do OPC Clássico em um único *framework* extensível.”.

Este novo protocolo carrega algumas características importantes e que a guiaram no caminho de substituir o uso do OPC clássico. Dentre estes pontos temos a compatibilidade com o protocolo antigo, tornando assim mais fácil a migração entre eles. Outro ponto relevante é a independência da plataforma que inicialmente se restringia ao Windows e a melhoria na segurança por meio de autenticação e criptografia.

2.3 IHM

IHM pode ser definido como um computador industrial capaz de permitir uma comunicação entre o operador de chão de fábrica e o sistema de controle operante. Assim o IHM traduz os sinais elétricos e valores internos do sistema em uma interface intuitiva e amigável a quem a usa, podendo não só interpretar os valores como enviar comandos ao controlador (PAHC, 2021)

Uma segunda definição agora trazida por Cravo (2023c, *online*) descreve a IHM como “Um equipamento com uma tela (que pode ser *touch screen* ou não), cujo objetivo é facilitar a comunicação e interação entre o homem e os sistemas ou dispositivos atuantes na produção inteligente, nas plantas de produção”.

As IHMs no ambiente industrial, além de proporcionar a maior interatividade dos operadores com o sistema de controle, também conta com um design robusto para lidar com o ecossistema nocivo no qual estão inseridas. Característica também compartilhada pelos CLPs. A Figura 11 ilustra um exemplo desta interface.

Figura 11 - Exemplo de IHM



Fonte: (CRAVO, 2023c)

No CODESYS temos disponível a ferramenta CODESYS *Visualization* através da qual pode-se montar interfaces para visualização e controle dos processos no mesmo ambiente onde é desenvolvida a lógica de controle. A ferramenta oferece elementos de interface tipicamente encontrados em IHMs e a capacidade de personalizar elementos genéricos (CODESYS, 2022b).

Portanto, é possível que tenhamos uma comunicação em tempo real entre um CLP e uma IHM gerada e executada pelo CODESYS *Visualization* via protocolos como OPC. No entanto, ao utilizar o CODESYS *Control Win* não são necessárias configurações de comunicação uma vez que ambos estão tendo sua operação mediada pela própria IDE.

2.4 Factory IO

Para simular o ambiente industrial foi escolhido o *software* Factory IO que conta com uma gama de elementos industriais que tornam a aplicação próxima da realidade sem que haja grande custo na aquisição dos componentes reais.

O Factory IO é um simulador 3D industrial para aprendizado de tecnologias de automação. O uso mais comum é como uma plataforma para treinamento de CLPs, uma vez que são os controladores mais encontrados neste ambiente (FACTORY IO, 2023a). A Figura 12 mostra uma das cenas criadas no *software*.

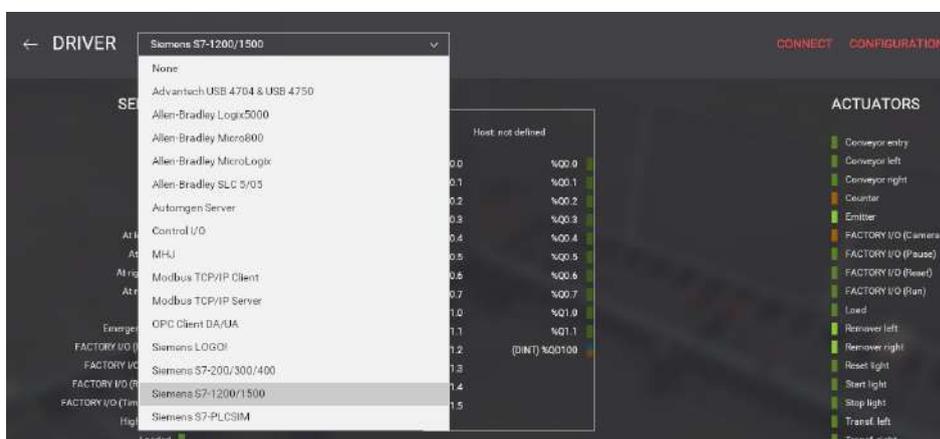
Figura 12 - Exemplo de simulação montada no Factory IO



Fonte: (FACTORY IO, 2023)

O *software* oferece uma variedade de *drivers* de entrada e saída, permitindo a conexão entre o CLP e o ambiente simulado no Factory IO. Esses *drivers* incluem suporte para CLP S7-1200 da Siemens, PLC-SIM (um simulador de CLP da mesma empresa) e também *drivers* para clientes OPC, essenciais para a comunicação com o CODESYS, que possui um servidor OPC UA. A Figura 13 apresenta a área de seleção e os *drivers* disponíveis.

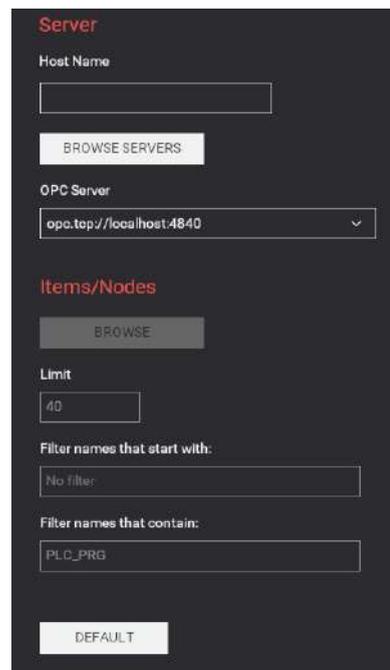
Figura 13 - Seleção de *drivers* de comunicação no Factory IO



Fonte: (FACTORY IO, 2023)

Para conectar ao servidor OPC UA do CODESYS é necessário que se escolha a opção OPC Client DA/UA. Uma vez selecionado, na aba de configuração, devem ser inseridas as informações necessárias para que o Factory IO encontre o servidor corretamente e as entradas e saídas disponibilizadas conforme Figura 14.

Figura 14 - Configuração da conexão OPC UA no Factory IO



The image shows a dark-themed configuration window for OPC UA. It is divided into two main sections: 'Server' and 'Items/Nodes'.
In the 'Server' section, there is a 'Host Name' input field, a 'BROWSE SERVERS' button, and an 'OPC Server' dropdown menu currently showing 'opc.tcp://localhost:4840'.
In the 'Items/Nodes' section, there is a 'BROWSE' button, a 'Limit' input field with the value '40', a 'Filter names that start with:' input field containing 'No filter', and a 'Filter names that contain:' input field containing 'PLD_PRG'.
At the bottom of the window is a 'DEFAULT' button.

Fonte: Autor

Por padrão o servidor irá utilizar a porta 4840 e o protocolo `opc.tcp`, assim poderá ser localizado pela url "`opc.tcp://localhost:4840`". Mais abaixo podem ser definidos limites à quantidade de itens (sinais de entrada e saída) que é desejado que seja descoberto bem como aplicação de filtro ao nome destes itens para mais fácil identificação dos elementos desejados.

Como já dito, o *software* conta com diversos elementos simulados para montagem do ambiente desejado, desde sensores, atuadores e componentes específicos ao seu contexto de uso. Adiante iremos especificar tanto o exemplo real, quando aplicado, quanto o que é trazido de forma simulada para uso no presente projeto.

2.4.1 Sensor Difuso Óptico

O sensor difuso óptico funciona a partir da emissão de um raio de luz invisível que, ao rebater em um objeto sólido, retorna ao receptor causando o acionamento do sensor. Ele conta com um emissor e um receptor no mesmo dispositivo de modo que um feixe de luz é

emitido pelo emissor e recebido de forma difusa pelo receptor ao encontrar um objeto sólido no caminho (ADIR; SOUZA, 2018). A Figura 15 mostra um exemplo deste sensor

Figura 15 - Sensor óptico



Fonte: <https://www.elegraz.com.br/sensor-fotoeletrico-difuso>

No Factory IO encontramos uma simulação deste sensor como mostrado na Figura 16. Este é capaz de detectar qualquer objeto sólido, dentro das definições de objeto do próprio simulador, e funciona com apenas um bit de informação. Quando está detectando um objeto seu valor é 1 ou o equivalente a 24v para o CLP e quando não detecta permanece em 0 ou 0V (FACTORY IO, 2023b).

Um sensor real possui em sua especificação a distância máxima de detecção enquanto na simulação somos capazes de regular qual será esta distância entre 0-1,6 metros.

Figura 16 - Sensor óptico difuso no Factory IO



Fonte: (Factory IO, 2023)

2.4.2 Sensor De Visão

O sensor de visão simulado pelo Factory IO não é inspirado em um sensor real em específico e sim atende às necessidades do próprio contexto do simulador. Assim sua

detecção é focada no formato do material e sua cor dentro das possibilidades apresentadas no *software*. Na Figura 17 vemos uma representação deste sensor.

Figura 17 - Sensor de visão no Factory IO



Fonte: (FACTORY IO docs, 2023)

Este sensor é capaz de detectar 3 categorias de objetos do simulador, *Raw materials* (matéria-prima), *Product Lids* (tampa do produto) e *Product bases* (base do produto) e suas respectivas cores, cinza, verde ou azul. Ele pode retornar 4 bits de informação (*ALL Digital*) permitindo a identificação do objeto ou de maneira analógica (*ALL numerical*) retornar apenas um número identificando o objeto ou também de maneira analógica (*ALL ID*) retornar um valor único aleatório que relaciona ao objeto detectado segundo a Figura 18 (FACTORY IO, 2023b).

Figura 18 - Relação entre objeto e sinal gerado pelo sensor de visão no Factory IO

Item	All Digital	All Numerical	All ID
	Bit 0 1 2 3	Value	Value
None	0 0 0 0	0	0
Blue Raw Material	1 0 0 0	1	ID
Blue Product Lid	0 1 0 0	2	ID
Blue Product Base	1 1 0 0	3	ID
Green Raw Material	0 0 1 0	4	ID
Green Product Lid	1 0 1 0	5	ID
Green Product Base	0 1 1 0	6	ID
Metal Raw Material	1 1 1 0	7	ID
Metal Product Lid	0 0 0 1	8	ID
Metal Product Base	1 0 0 1	9	ID

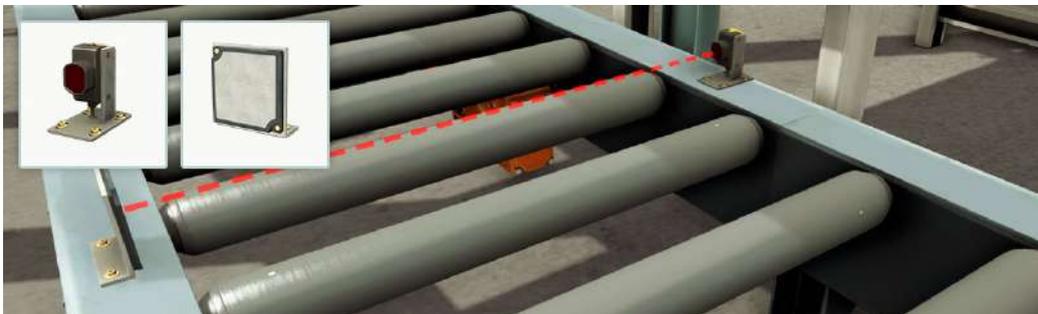
Fonte: (FACTORY IO, 2023)

2.4.3 Sensor Reflexivo

Um sensor reflexivo, conforme descrito por Blog Sense (2019, *online*), é um tipo de sensor que inclui tanto o transmissor quanto o receptor em um único invólucro. A detecção de objetos com esse sensor ocorre por meio da reflexão da luz através de um espelho prismático colocado em frente ao sensor. Quando um objeto interrompe o feixe de luz refletido pelo espelho, a detecção é acionada. O espelho usado neste sensor possui prismas em toda a sua superfície, que refletem uma grande porção da luz polarizada de volta para o sensor de forma paralela, permitindo uma considerável distância entre o sensor e o espelho. Um exemplo prático de aplicação deste sensor reflexivo é a detecção de caixas em linhas transportadoras.

No Factory IO, temos uma representação simulada do sensor reflexivo, como ilustrado na Figura 19, composta por duas partes: o próprio sensor e um refletor designado para refletir o feixe emitido pelo sensor. Quando o feixe é refletido pelo refletor, indicando a ausência de qualquer obstrução, seu valor é 1, equivalente a 24V para o CLP. No entanto, quando um objeto interrompe o trajeto do feixe, o valor do sensor cai para 0, correspondendo a 0V (FACTORY IO, 2023b).

Figura 19 - Sensor Reflexivo no Factory IO



Fonte: (FACTORY IO, 2023)

2.4.4 Classificador De Rodas Pop-Up

O *Pop-Up Wheel Sorter*, ou Classificador De Rodas Pop-Up, é um sistema de classificação que utiliza uma esteira transportadora com rodas, rolos ou correias incorporados, que podem subir, para transferir itens a um ângulo de 30 ou 90 graus para outra esteira transportadora adjacente. Esse tipo de sistema é mais eficaz ao lidar com produtos uniformes, como itens em caixas ou carrinhos, tornando-o uma escolha ideal para tarefas de classificação durante o processo de empacotamento (ROMAINE, 2020).

No Factory IO temos uma simulação deste classificador, conforme mostrado na Figura 20, que possui um funcionamento semelhante ao seu exemplo real. Este

classificador é composto por diversas rodas capazes de girar em um ângulo de 45° e pode assim, desviar os objetos para a direita ou esquerda, ou apenas permitir que sigam adiante dependendo das configurações de suas variáveis de controle.

Ele conta com três variáveis de controle rotuladas “+”, “Left” e “Right”. Elas por sua vez podem ser combinadas para causar os comportamentos desejados:

- Seguir adiante: Para acionamento de maneira que o produto siga apenas adiante deve-se colocar “+” em nível lógico alto para que assim as rodas se levantem e girem para frente.
- Desviar para direita: Para acionamento de maneira que o produto seja direcionado para a saída à direita do classificador deve-se colocar em nível lógico alto “+” e “Right” de modo que o segundo gira as rodas para a direita.
- Desviar para esquerda: Para acionamento de maneira que o produto seja direcionado para a saída à esquerda do classificador deve-se colocar em nível lógico alto “+” e “Left” de modo que o segundo gira as rodas para a esquerda.
- Acionar “Right” e “Left” simultaneamente faz com que as rodas retornem para a posição inicial girando para frente.

Figura 20 - Classificador de Rodas Pop-Up no Factory IO



Fonte: (FACTORY IO, 2023)

2.4.5 Classificador De Braço Articulado

O Classificador de Braço Articulado consiste em um braço articulado posicionado ao longo de uma esteira transportadora. Este braço quando ativado é capaz de se projetar sobre a esteira desviando o caminho do produto para uma outra esteira transportadora. Os classificadores de braço articulado são amplamente utilizados em ambientes de classificação de remessas, especialmente no manuseio de bagagens e pacotes (ROMAINE, 2020).

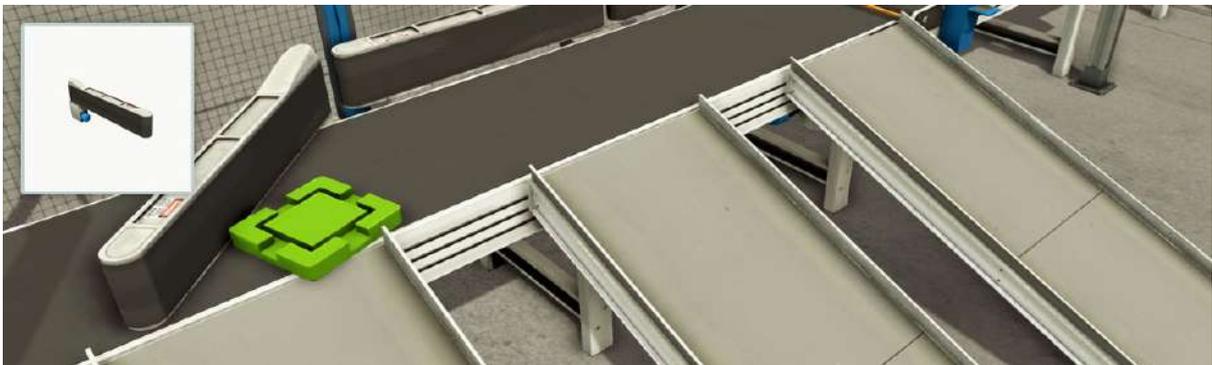
A Figura 21 mostra o classificador presente no Factory IO baseado no comportamento deste atuador real. Ele possui, além da articulação que permite sua

movimentação para sobre a esteira, uma esteira para facilitar o processo de separação quando um produto passa em contato com ele.

Para seu funcionamento são disponibilizadas três variáveis discretas: “Turn”, “+” e “-”. Colocar “Turn” em nível lógico alto faz com que a articulação entre em ação posicionando o classificador sobre a esteira transportadora. A variável “+” por sua vez determina se sua esteira deve girar e em sentido horário enquanto “-” determina se deve girar e em sentido anti-horário.

Por fim, o lado para o qual o braço deve virar é definido manualmente na configuração do elemento no Factory IO, permitindo assim que esse classificador fique à direita ou à esquerda da esteira transportadora.

Figura 21 - Classificador de Braço Articulado no Factory IO



Fonte: (FACTORY IO, 2023)

2.4.6 Emissor

Para facilitar a inserção de produtos no sistema durante a simulação no Factory IO é disponibilizado o elemento Emissor. Este não é um componente real sendo específico para o contexto do simulador para que assim, produtos sejam gerados em tempo real no ambiente para uso no processo.

O emissor é uma ferramenta que libera itens para serem usados em uma cena, com opções para escolher o tipo de item (*Pallets, Materials, Box*), o tempo entre as emissões, a quantidade total e a possibilidade de posicionamento e orientação aleatórios. É possível ativar ou desativar o emissor e este só produz um novo objeto quando não há outro já presente em sua área de emissão (FACTORY IO, 2023c). A Figura 22 mostra este elemento e sua área pontilhada de geração.

Figura 22 - Emissor no Factory IO



Fonte: (FACTORY IO, 2023)

3 Trabalhos relacionados

Neste capítulo iremos tratar dos trabalhos encontrados durante a revisão bibliográfica que auxiliaram no desenvolvimento deste trabalho.

1. Sistema de análise e controle de produtos em um processo de carregamento:

Gouvêa (GOUVÊA, 2021) apresenta um sistema de identificação e contagem de caixas por tamanho, seguido da geração de relatório em excel. Para simular o ambiente industrial a ser controlado o autor utilizou do *software* Factory IO integrado com o *software* CODESYS para codificação e simulação do CLP.

O ambiente foi montado com o objetivo de fazer a identificação um a um das caixas geradas e logo em seguida removidas do ambiente por meio dos elementos Emissor e Removedor do Factory IO, a Figura 23 mostra o projeto final no *software*.

Figura 23 - Cenário para identificação das caixas no Factory IO



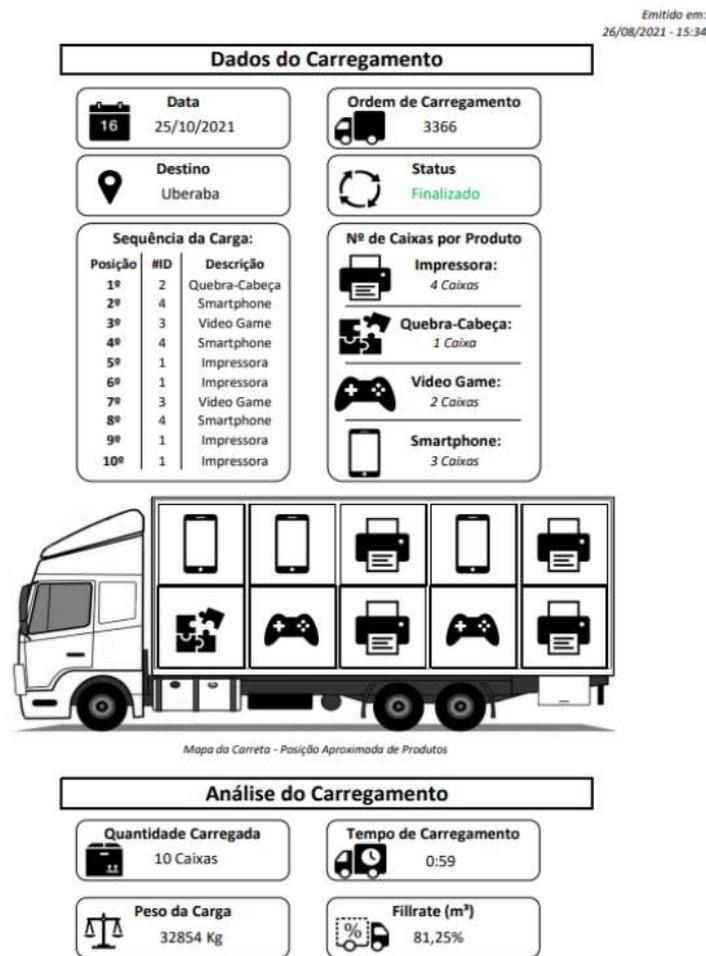
Fonte: (GOUVÊA, 2021)

Por meio do uso de linguagens como Ladder e Structured Text ele foi capaz de criar uma lógica de controle que identificou os produtos que passaram na esteira transportadora e exportou os dados para o excel. Assim, ao final os dados modelados utilizando da linguagem VBA um relatório foi montado com base em diversas informações colhidas pelo controlador durante sua execução, como mostra a Figura 24.

Segundo Gouvêa (2021, p.4):

O objetivo do projeto é desenvolver um sistema de controle, monitoramento e análise capaz de facilitar o processo de carregamento de carretas e fornecer dados importantes de forma automatizada como: ordem de produtos embarcados, tipo e quantidade de cada produto e tempo de carregamento. Essas informações serão coletadas e apresentadas na forma de um relatório, com um mapa da carreta, os dados do carregamento e uma análise gerada automaticamente. Com isso, espera-se obter assertividade entre produtos planejados e embarcados, facilidade de descarga e até mesmo abrir portas para outras análises.

Figura 24 - Relatório de carga em Excel



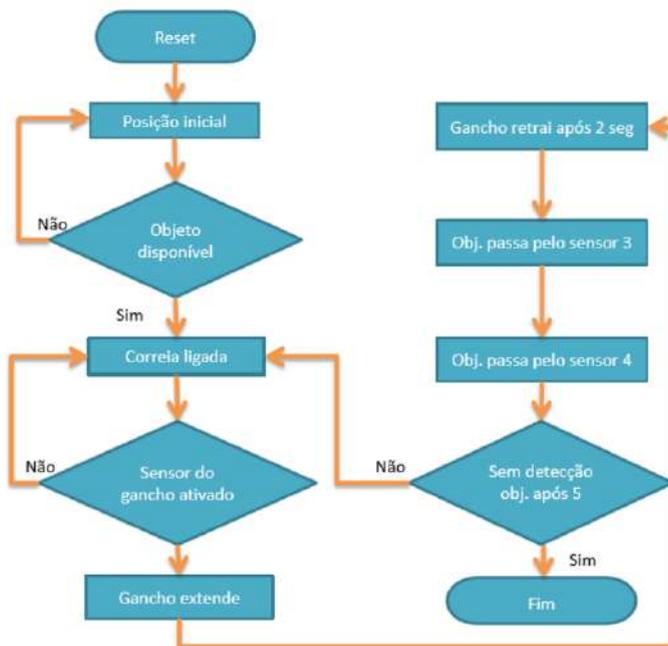
Fonte: (GOUVÊA, 2021)

2. Aplicação de um sistema automatizado de uma linha de produção MPS 500 utilizando CLP S7-1200:

Lage (2017) propõe um sistema de controle utilizando da Estação Modular de Produção (MPS-500) da marca FESTO e um CLP S7-1200 da Siemens para realização de um processo envolvendo a identificação e a separação de objetos por sua cor.

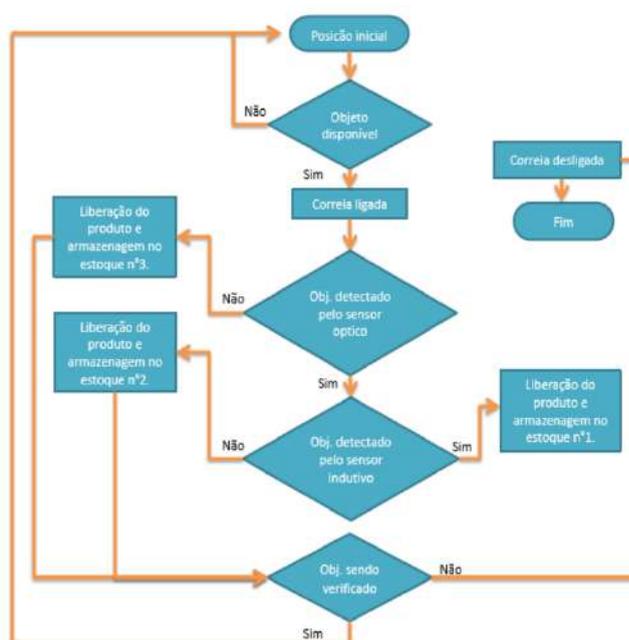
Por meio da associação de MPS-500 foi construído um sistema físico capaz de, por meio do controle do CLP, transportar objetos e separá-los mediante a sua identificação. Este sistema é dividido em duas etapas constituindo uma estação em cada uma delas sendo a estação de transporte e a estação de classificação. A primeira tem como objetivo apenas a organização e transporte inicial dos objetos, enquanto a segunda realiza a identificação e classificação. A Figura 25 mostra um fluxograma do funcionamento da estação de transporte, já a Figura 26 mostra o fluxograma do funcionamento da estação de classificação.

Figura 25 - Fluxograma da estação de transporte



Fonte: (Lage, 2017)

Figura 26 - Fluxograma da estação de classificação



Fonte: (LAGE, 2017)

Para desenvolvimento do trabalho foi utilizado o *software* Tia Portal da Siemens para que fosse realizada a programação e a configuração do CLP

disponível no laboratório. Foi feito o uso de variáveis digitais na linguagem Ladder, contato também com blocos temporizadores para construção da lógica de controle. Segundo Lage (2017, p.15) o objetivo do trabalho é:

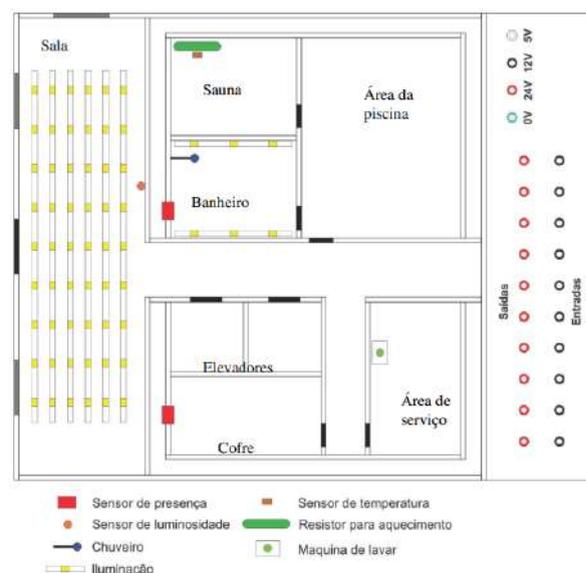
Programar três estações da plataforma FESTO, com o propósito de gerar um sistema que simule uma linha de produção. O programa final deverá ser capaz de pegar um objeto, levá-lo até seu armazém, identificando-o e separando-o por suas respectivas cores.

3. Projeto de Domótica Utilizando CLP:

Neste trabalho de Medeiros (MEDEIROS, 2014) é explorada a aplicação do Controlador Lógico Programável (CLP) na automação predial, visando a integração e centralização do sistema, conhecido como domótica. A domótica busca proporcionar conforto e eficiência energética, reduzindo o consumo de eletricidade e aquecimento. O estudo destaca a importância da domótica e dos CLPs, culminando na implementação do CLP em uma maquete de um imóvel, resultando em uma miniatura automatizada com sensores de baixo custo.

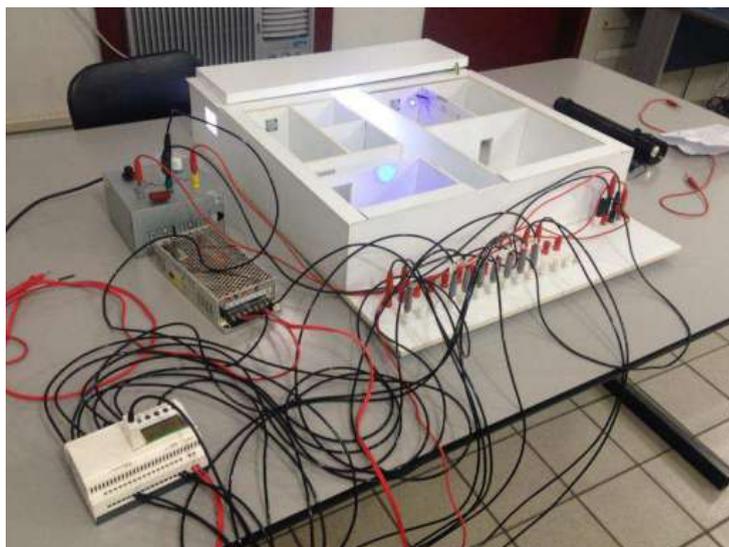
Para a execução deste trabalho foi utilizado um CLP real Zelio SR3B261BD fabricado pela Schneider Eletric e o ambiente a ser controlado por ele envolveu a criação de uma maquete e o uso de sensores, como por exemplo de movimento e luminosidade, e atuadores como leds e resistores para aquecimento. A Figura 27 mostra o esquema do ambiente e a Figura 28 a maquete montada com as conexões no CLP.

Figura 27 - Esquema do ambiente a ser controlado



Fonte: (MEDEIROS, 2014)

Figura 28 - Maquete com as conexões com o CLP



Fonte: (MEDEIROS, 2014)

A linguagem de programação Ladder, compatível com o CLP, foi a escolha do autor para a implementação do projeto. Por meio dessa linguagem, foi possível desenvolver as lógicas de controle para os ambientes selecionados, empregando combinações de diversas funções lógicas, tais como "AND", "OR" e "NOT".

Segundo Medeiros (2014, p.2):

Devido ao potencial de utilização dos CLPs no controle e automação, este trabalho tem o objetivo de familiarizar os métodos de utilização e de programação desse dispositivo. Uma maneira de explorar a maioria das funcionalidades do CLP é a de automatizar os diversos sistemas de um imóvel.

4 Procedimentos metodológicos

A presente pesquisa é exploratória do ponto de vista dos objetivos; aplicada quanto à natureza; experimental quanto aos procedimentos técnicos (WAZLAWICK, 2014).

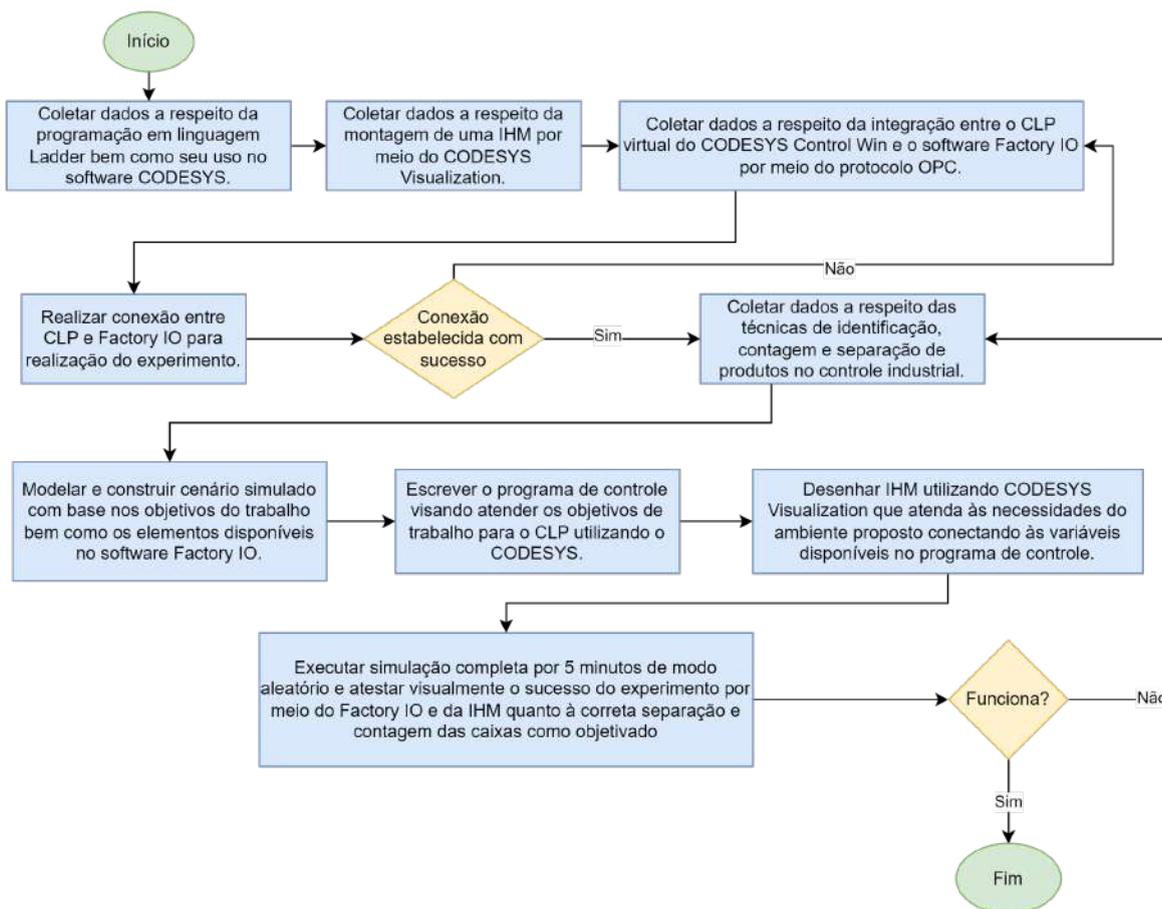
4.1 Métodos e Materiais

Os métodos consistem em:

1. Coletar dados a respeito da programação em linguagem Ladder bem como seu uso no *software* CODESYS.
2. Coletar dados a respeito da montagem de uma IHM por meio do CODESYS *Visualization*.
3. Coletar dados a respeito da integração entre o CLP virtual do CODESYS *Control Win* e o *software* Factory IO por meio do protocolo OPC.
4. Realizar conexão entre CLP e Factory IO.
5. Coletar dados a respeito das técnicas de identificação, contagem e separação de produtos no controle industrial.
6. Modelar e construir cenário simulado com base nos objetivos do trabalho bem como os elementos disponíveis no *software* Factory IO.
7. Escrever o programa de controle visando atender os objetivos de trabalho para o CLP utilizando o CODESYS.
8. Desenhar IHM utilizando CODESYS *Visualization* que atenda às necessidades do ambiente proposto conectando às variáveis disponíveis no programa de controle.
9. Executar simulação completa por 5 minutos de modo aleatório e atestar visualmente o sucesso do experimento por meio do Factory IO e da IHM quanto à correta separação e contagem das caixas como objetivado.

O diagrama da Figura 29 mostra com mais detalhes o fluxo acima descrito.

Figura 29 - Fluxograma metodológico do projeto



Fonte: Autor

5 Desenvolvendo o sistema de controle

A etapa de desenvolvimento deste trabalho visa a criação de um ambiente industrial simulado utilizando o Factory IO, que será controlado por um CLP virtual por meio do CODESYS *Control Win* e com visualização por meio da IHM criada no Codesys *Visualization*. Com o objetivo de implementar a lógica de identificação e separação de caixas com base em seu tamanho ou cor, é necessário entender a construção de cada elemento, os quais serão detalhados a seguir.

5.1 Construção do ambiente industrial

O Factory IO possibilita a construção de ambientes industriais simulados, oferecendo uma gama de recursos para montagem. Ele inclui elementos como esteiras transportadoras, sensores, dispositivos de emissão de produtos e mecanismos de remoção, permitindo a criação de cenários virtuais realistas para experimentação e aprendizado em automação industrial.

Tomando como base o objetivo do projeto a criação do ambiente no Factory IO foi dividida em duas etapas, assim como o processo como um todo, sendo elas a identificação e a separação dos produtos.

5.1.1 Mecanismos de identificação

Com base nos elementos disponíveis no *software* foram escolhidos uma série de produtos alvos de identificação durante o processo. Estes produtos irão ser gerados pelo elemento Emissor do Factory IO de maneira aleatória em intervalos de tempo definidos.

O primeiro conjunto de produtos, nomeado Conjunto 1, se diferencia no tamanho mantendo a mesma cor, sendo 4 tipos de caixas em que as dimensões variam tanto em altura quanto em largura, como mostra a Figura 30. Para melhor identificação destes elementos iremos nomeá-los com relação ao seu tamanho segundo a mesma Figura 28, assim teremos a caixa grande (Produto G), caixa média larga (Produto ML), caixa média (Produto M) caixa pequena (Produto P).

Figura 30 - Tipos de caixa diferentes no tamanho

Fonte: Autor

No Factory IO temos disponível os elementos do tipo matéria-prima (*Raw Material*), tampa de produto (*Product Lid*) e base de produto (*Product Base*) de maneira que cada um destes elementos têm suas versões em três cores distintas, azul, verde e cinza e se diferenciam em seu formato. Como estamos apenas interessados na diferenciação por cor e não de formato, será escolhido o conjunto de elementos do tipo matéria prima apenas.

Para melhor identificação ao longo do texto iremos nomeá-los em relação a sua cor, portanto teremos matéria prima azul (Produto Azul), matéria prima verde (Produto Verde), matéria prima cinza (Produto Cinza). Assim, o segundo conjunto de produtos, nomeado Conjunto 2, se diferencia na cor mantendo o mesmo tamanho, sendo 3 diferentes objetos. A Figura 31 ilustra estes elementos.

Figura 31 - Tipos de matéria prima diferentes na cor

Fonte: Autor

Como mostrado na Figura 32 os produtos do Conjunto 2 não variam de tamanho e não apresentam tamanho semelhante a quaisquer elementos do Conjunto 1, assim também ocorre com os produtos do Conjunto 1 que não variam de cor e não possuem quaisquer cores dos elementos do Conjunto 2. Desta maneira a identificação dos produtos de um conjunto não interfere na identificação dos produtos de outro, sendo uma característica imposta pela limitação do software, que não fornece meios para variar ambas as propriedades em um único produto, e que será importante no momento de desenvolver as respectivas lógicas de identificação.

Figura 32 - Visualização do Conjunto 1 e Conjunto 2 lado a lado



Fonte: Autor

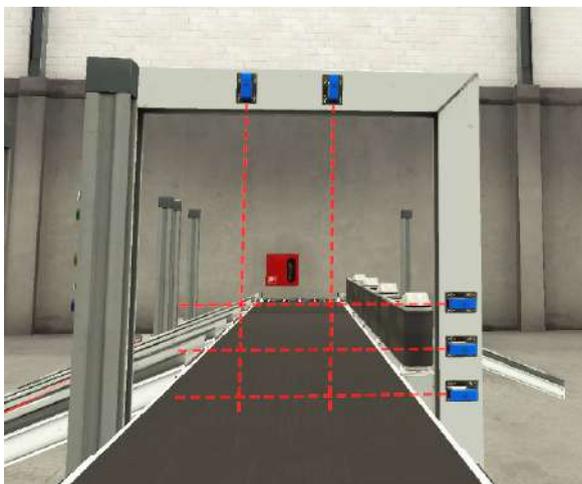
Após a determinação dos elementos a serem transportados pela esteira, é possível estabelecer um procedimento de identificação. Seguindo a abordagem apresentada por Gouvêa (2021), optou-se por empregar o sensor difuso óptico disponibilizado pelo Factory IO para a detecção dos quatro tamanhos de caixas em questão.

O sensor difuso óptico possui a capacidade de discernir a presença ou ausência de um objeto em sua faixa de atuação, a qual pode ser configurada no *software*. Este dispositivo emite um sinal digital, registrando o valor "0" quando não há objetos em seu campo de detecção e "1" quando identifica a presença de um objeto. Nesse sentido, serão dispostos dois sensores de forma vertical sobre um pedaço da esteira para a medição da largura das caixas, enquanto três sensores horizontais serão posicionados para medir a altura das caixas.

A combinação das informações obtidas por esses sensores permite a determinação do elemento que está passando pelos sensores naquele momento. A disposição dos

sensores e a sua área de atuação são exemplificadas na Figura 33, enquanto a Figura 34 representa a passagem do Produto G pelo portal de sensores, resultando no acionamento de todos eles.

Figura 33 - Disposição dos sensores difusos ópticos no Factory IO



Fonte: Autor

Figura 34 - Passagem do Produto G no portal de identificação no Factory IO

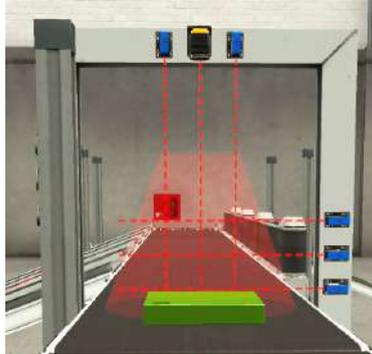


Fonte: Autor

Para a identificação do Conjunto 2 o qual possui variação nas cores utilizaremos o sensor de visão disponível no Factory IO. O sensor de visão é capaz diferenciar entre os elementos: matéria-prima, tampa de produto e base de produto, e suas respectivas cores, verde, azul e cinza. Como resultado o sensor retorna um valor digital de 4 bits

representando o objeto identificado. O sensor será colocado verticalmente no mesmo portal onde estarão os sensores difusos como mostra a Figura 35.

Figura 35 - Posição do sensor de visão no portal de identificação no Factory IO



Fonte: Autor

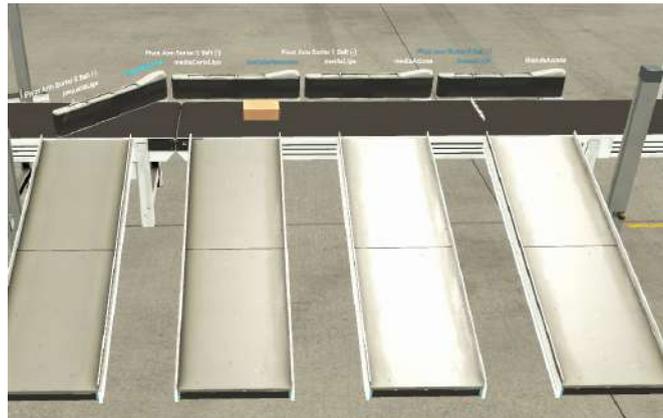
5.1.2 Mecanismos de separação

O sistema conta com 7 produtos distintos passíveis de identificação como descrito anteriormente, logo estes elementos devem ser encaminhados para 7 locais diferentes mediante suas características. Após a primeira etapa todos os produtos irão seguir por uma esteira de transporte, nomeada Esteira Principal, ao longo da qual serão realizadas as devidas separações. Para isto serão utilizados dois métodos diferentes com base nos equipamentos disponíveis no Factory IO.

O primeiro equipamento escolhido para a separação é o Classificador De Braço Articulado. Ele fica posicionado na lateral de uma esteira e conta com três variáveis digitais, para acionamento, para indicação do sentido de atuação de sua esteira e o acionamento do braço para efetuar a separação. Uma vez que este equipamento está acionado ele é capaz de mudar a direção para onde um objeto sobre a esteira deve ir, podendo ser encaminhado para outra plataforma de transporte exclusiva para os elementos que ele está separando.

O classificador de braço articulado foi designado para a separação das do Conjunto 1, exigindo assim 4 destes dispostos ao longo da Esteira Principal e 4 rampas para encaminhamento exclusivo de cada tipo de produto. A Figura 36 mostra a disposição dos braços ao longo da esteira, das rampas e exemplifica o acionamento de um dos braços para separação.

Figura 36 - Separação por Classificador de Braço Articulado no Factory IO

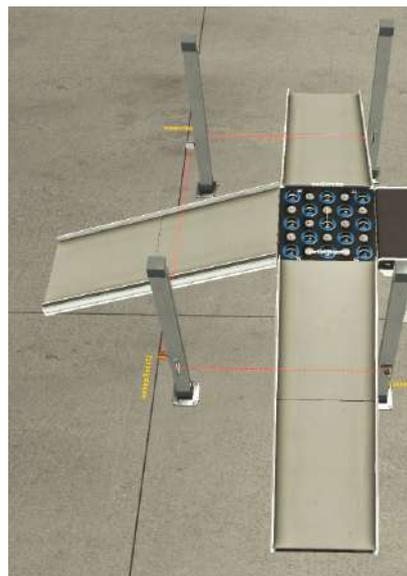


Fonte: Autor

Visando reduzir o tamanho do sistema, uma vez que para separar as matérias-primas seria necessário aumentar a Esteira Principal e adicionar mais 3 Classificadores de Braço Articulado, e apresentar mais de uma forma de separação baseado nos mecanismos oferecidos pelo *software*, o Conjunto 2 utilizará do Classificador De Roda Pop-Up.

Este classificador conta com 3 variáveis digitais para acionamento e definição da direção de encaminhamento dos produtos. Assim é possível adicionar mais 3 rampas nas laterais deste equipamento que é capaz de direcionar os produtos para 3 direções diferentes como mostra a Figura 37.

Figura 37 - Separação por Classificador de Rodas Pop-Up no Factory IO



Fonte: Autor

Por fim, como medida para indicar que o produto avançou com sucesso para a região seguinte, permitindo que um novo elemento inicie o processo de identificação, serão implementados sensores reflexivos nas rampas.

O sensor reflexivo, diferentemente do sensor difuso óptico, tem a capacidade de identificar a passagem de um objeto através da interrupção do seu feixe de luz. Nesse cenário, o sensor constantemente retorna o valor digital "1" na ausência de objetos em seu campo de detecção e "0" quando um objeto está presente.

Para as rampas utilizadas na separação por meio do classificador de roda pop-up, que se encontram em direções distintas, foi necessário alocar um sensor em cada rampa. Por outro lado, as rampas empregadas na separação por meio do braço articulado estão em uma mesma direção, o que possibilitou a cobertura adequada por um único sensor. As Figura 36 e Figura 37 demonstram os sensores e suas áreas de atuação representados em pontilhado vermelho.

5.1.3 Sistema completo

Após definir os equipamentos e suas disposições em ambas as etapas do processo a ser simulado é necessário realizar a junção em um único sistema a ser controlado com um mecanismo de geração de produtos.

Com o propósito de incorporar os 7 tipos de produtos no sistema, permitindo que eles transitem pelas etapas do processo, será implementada uma esteira secundária, equipada com um elemento Emissor. A responsabilidade pela geração aleatória e em intervalos predefinidos dos 7 tipos de produtos está neste emissor. A esteira secundária encaminhará os produtos gerados até a Esteira Principal, onde seguirão o restante do percurso. A Figura 38 apresenta uma visão geral do sistema montado por completo

Figura 38 - Sistema completo montado no Factory IO



Fonte: Autor

5.2 Construção IHM

A visualização do processo físico será feita por meio do *software* Factory IO que permitirá uma prova em tempo real do funcionamento do sistema atestando visualmente a identificação e separação de cada produto ao longo do tempo. No entanto, uma IHM se faz necessária a fim de permitir o acompanhamento das variáveis internas do sistema, como valor da contagem, e controle sobre alguns processos para que possamos realizar ações em um cenário mais próximo ao real sem a necessidade de pausas nos simuladores.

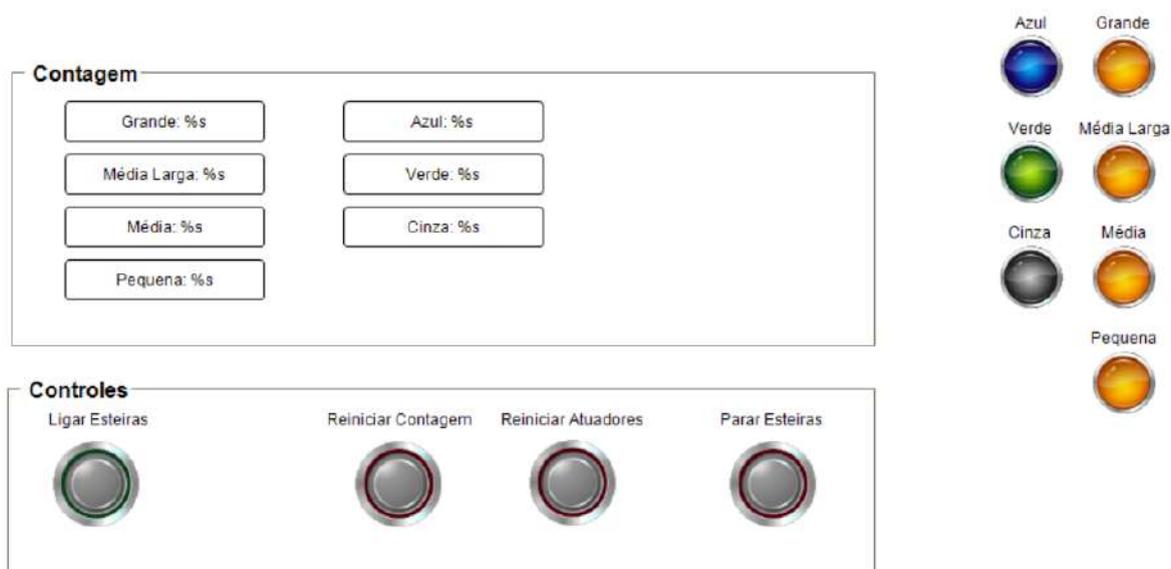
Um exemplo da importância da presença de uma IHM está em ações realizadas repetidamente durante o desenvolvimento do trabalho que consistia na pausa da simulação do cenário do Factory IO para manutenção ou alteração da posição de componentes a fim de se adequar ao objetivo do projeto. O funcionamento do CLP virtual é independente da existência de uma simulação ativa no FACTORY IO, ou seja, pausar o *software* não faz com que as variáveis internas do CLP retornem ao valor padrão do início da execução e muito menos reinicie a contagem dos contadores do controlador. Esta situação causa a necessidade de desligamento forçado do CLP para que este volte a seu estado inicial de execução e entre em acordo com o que se estava sendo visto no ambiente 3D. Ações essas e outras que podem ser evitadas por meio da implementação da IHM explicada a seguir.

Será utilizado aqui o CODESYS *Visualization* que é capaz de nos fornecer elementos típicos de uma IHM para visualização do processo por meio do design da interface e associação com as variáveis internas do programa que está rodando no CLP simulado.

O monitoramento e controle por meio da interface terá como alvo a contagem dos produtos durante o processo, a operação dos atuadores e a identificação através dos sensores. A interface apresentará os seguintes itens, conforme Figura 39:

- Visualização da contagem
- Visualização da identificação, e
- Botões de ação

Figura 39 - IHM montada no CODESYS



Fonte: Autor

5.2.1 Visualização da contagem

Para exibir em tela a contagem de cada tipo de produto que passa pelo sistema, será utilizado o elemento Retângulo Arredondado (*Rounded Rectangle*). Para cada um dos produtos do Conjunto 1 e 2, totalizando 7 produtos, será adicionado um Retângulo Arredondado.

A propriedade de Texto (*Text*) será preenchida com o valor “Grande: %s” se tomarmos como exemplo o desejo de exibição da contagem dos Produtos G. Já a propriedade Variável do Texto (*Text Variable*) apontará para a variável interna do CLP capaz de armazenar o valor da contagem segundo a lógica que será descrita mais à frente.

A fim de tornar uma visualização mais clara e objetiva em meio a outras informações disponíveis na interface, os Retângulos Arredondados foram colocados dentro de um Grupo (*Group Box*).

5.2.2 Visualização da identificação

Olhando para o momento em que o produto passa pelo portal e é identificado pelos sensores presentes será utilizado o elemento LED para sinalização momentânea do objeto distinguido. Acima deste será adicionado o elemento Rótulo (*Label*) para correta associação do LED ao produto que ele representa.

A propriedade Variável (*Variable*) aponta para a variável interna do CLP capaz de indicar se o referido produto está passando pelo portal de identificação naquele momento.

5.2.3 Botões de ação

Além da visualização das informações é desejado que se possa atuar no sistema de controle mediante a alguma falha ou necessidade durante o processo. Para isso foi escolhido o elemento Botão de Pressionar (*Push Switch*) para que atue como um sinal a ser processado pela lógica de controle. A propriedade Variável deste elemento terá como valor o rótulo dado ao sinal que ele permite emitir, portanto, ao ser pressionado ele emitirá o sinal 1 e quando solto o sinal 0.

A série de Botões será colocada dentro de um Grupo para identificação da área reservada a ações de controle. Quatro elementos serão inseridos com as seguintes finalidades e com os seguintes rótulos inseridos logo acima na interface para identificação:

- Ligar Esteiras: Será associado a um sinal responsável por ligar todas as esteiras do sistema simultaneamente.
- Reiniciar Contagem: Será associado a um sinal responsável por zerar a contagem de todos os 7 produtos.
- Reiniciar Atuadores: Será associado a um sinal responsável por retornar todos os atuadores do sistema para o estado inicial aguardando a identificação do produto.
- Parar Esteiras: Será associado a um sinal responsável por interromper o funcionamento de todas as esteiras e separadores visando uma pausa completa de todo o sistema.

5.3 Lógica de controle

Tomando como base o ambiente montado no Factory IO, incluindo o modo de funcionamento dos elementos disponíveis e a sua disposição ao longo do processo, seguimos para a implementação da lógica de controle. Para isso foi utilizado o *software* CODESYS.

A Tabela 1 apresenta todos os sinais de entrada que o Factory IO e a IHM são capazes de fornecer durante o processo de simulação. Nessa tabela, apresenta-se o rótulo utilizado para designar as variáveis associadas a esses sinais, com a finalidade de utilização tanto no simulador quanto no CODESYS e IHM, além de servir como referência ao longo deste trabalho. Também são apresentados os detalhes relativos ao componente físico responsável pela emissão do sinal e à interpretação da leitura.

Tabela 1 - Sinais de entrada para uso no Factory IO e CODESYS

Sinais de Entrada Factory IO / CODESYS		
Rótulo (FactoryIO /CODESYS)	Elemento	Interpretação do sinal
Altura1	Sensor Difuso Óptico (Factory IO)	Quando emitido valor 1 constata-se que uma caixa de no mínimo altura baixa está presente
Altura2	Sensor Difuso Óptico (Factory IO)	Quando emitido valor 1 constata-se que uma caixa de no mínimo altura média está presente
Altura3	Sensor Difuso Óptico (Factory IO)	Quando emitido valor 1 constata-se que uma caixa de altura mais alta está presente
Largura1	Sensor Difuso Óptico (Factory IO)	Quando emitido valor 1 constata-se que uma caixa de no mínimo largura pequena está presente
Largura2	Sensor Difuso Óptico (Factory IO)	Quando emitido valor 1 constata-se que uma caixa de no mínimo largura pequena está presente
Cor1	Sensor de Visão (Factory IO)	Compõe os um dos 4 bits gerados pelo sensor sendo o bit menos significativo. Seu valor em conjunto dos demais identifica um objeto quando a tipo e cor
Cor2	Sensor de Visão (Factory IO)	Compõe os um dos 4 bits gerados pelo sensor sendo o segundo bit menos significativo. Seu valor em conjunto dos demais identifica um objeto quando a tipo e cor
Cor3	Sensor de Visão (Factory IO)	Compõe os um dos 4 bits gerados pelo sensor, sendo o terceiro bit bit menos significativo. Seu valor em conjunto dos demais identifica um objeto quando a tipo e cor
Cor4	Sensor de Visão (Factory IO)	Compõe os um dos 4 bits gerados pelo sensor sendo o bit mais significativo. Seu valor em

		conjunto dos demais identifica um objeto quando a tipo e cor
SensorReset0	Sensor Reflexivo (Factory IO)	Quando emite valor 0 constata-se que um objeto desceu por uma das rampas de separação por tamanho indicando fim da etapa
SensorReset1	Sensor Reflexivo (Factory IO)	Quando emite valor 0 constata-se que um objeto desceu pela rampa de separação para cor azul indicando fim da etapa
SensorReset2	Sensor Reflexivo (Factory IO)	Quando emite valor 0 constata-se que um objeto desceu pela rampa de separação para cor verde indicando fim da etapa
SensorReset3	Sensor Reflexivo (Factory IO)	Quando emite valor 0 constata-se que um objeto desceu pela rampa de separação para cor cinza indicando fim da etapa
LigaEsteiras	Botão (IHM)	Emite constantemente o valor 0, mas enquanto pressionado emite valor 1
ReiniciarAtuadores	Botão (IHM)	Emite constantemente o valor 0, mas enquanto pressionado emite valor 1
ReiniciarContadores	Botão (IHM)	Emite constantemente o valor 0, mas enquanto pressionado emite valor 1
PararEsteiras	Botão (IHM)	Emite constantemente o valor 0, mas enquanto pressionado emite valor 1

Fonte: Autor

Já na Tabela 2 é apresentada uma tabela similar que descreve os sinais de saída disponíveis nos ambientes configurados. Por meio dessas variáveis, será possível definir o funcionamento dos atuadores do sistema.

Tabela 2 - Sinais de saída para uso no Factory IO e CODESYS

Sinais de Saída		
Rótulo (FactoryIO / CODESYS)	Elemento	Interpretação do sinal
EsteiraEntrada	Esteira (4m) (Factory IO)	Quando recebido valor 1 a esteira irá girar carregando os objetos sobre ela na direção da esteira principal
EsteiraPrincipal1	Esteira (6m) (Factory IO)	Representa a esteira principal, quando recebido valor 1 ela irá girar carregando os objetos sobre ela ao longo do percurso de identificação e separação
EsteiraPrincipal2	Esteira (2m) (Factory IO)	Representa a esteira principal, quando recebido valor 1 ela irá girar carregando os objetos sobre ela ao longo do percurso de identificação e separação. Idealmente deve funcionar em concordância com a EsteiraPrincipal1
CorLiga	Classificador de rodas pop-up (Factory IO)	Quando recebido 1 as rodas se levantam e começam a girar transportando os objetos sobre elas adiante
CorDireita	Classificador de rodas pop-up (Factory IO)	Quando recebido 1 as rodas se direcionam para direita a fim de que em conjunto com CorLiga transporte os objetos sobre elas com desvio para direita
CorEsquerda	Classificador de rodas pop-up (Factory IO)	Quando recebido 1 as rodas se direcionam para esquerda a fim de que em conjunto com CorLiga transporte os objetos sobre elas com desvio para esquerda
GrandeAcciona	Classificador de Braço Articulado (Factory IO)	Quando recebido 1 o braço irá se projetar sobre a esteira a fim de desviar os objetos transportados por ela para a primeira rampa destinada às caixas grandes
MediaLargaAcciona	Classificador de Braço Articulado (Factory IO)	Quando recebido 1 o braço irá se projetar sobre a esteira a fim de desviar os objetos

		transportados por ela para a segunda rampa destinada às caixas médias largas
MediaAciona	Classificador de Braço Articulado (Factory IO)	Quando recebido 1 o braço irá se projetar sobre a esteira a fim de desviar os objetos transportados por ela para a terceira rampa destinada às caixas médias
PequenaAciona	Classificador de Braço Articulado (Factory IO)	Quando recebido 1 o braço irá se projetar sobre a esteira a fim de desviar os objetos transportados por ela para a quarta rampa destinada às caixas pequenas
Grande	LED (IHM)	Quando recebido 1 ele irá acender na cor configurada
MediaLarga	LED (IHM)	Quando recebido 1 ele irá acender na cor configurada
Media	LED (IHM)	Quando recebido 1 ele irá acender na cor configurada
Pequena	LED (IHM)	Quando recebido 1 ele irá acender na cor configurada
Azul	LED (IHM)	Quando recebido 1 ele irá acender na cor configurada
Verde	LED (IHM)	Quando recebido 1 ele irá acender na cor configurada
Cinza	LED (IHM)	Quando recebido 1 ele irá acender na cor configurada

Fonte: Autor

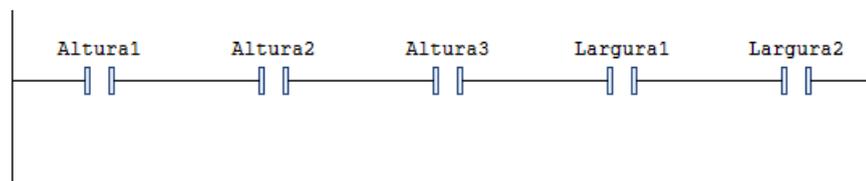
Devido a completa distinção entre o Conjunto 1 e o Conjunto 2 e os mecanismos de separação escolhidos para cada caso, a lógica terá particularidades isoladas, logo, a seguir serão descritas as regras de controle impostas para cada conjunto de elementos e por fim o comportamento dos botões da IHM.

5.3.1 Identificação e separação por tamanho

No processo de tratamento de caixas de tamanhos variados, os sensores ópticos difusos desempenham um papel fundamental. Para diferenciar os quatro tipos de produtos, utilizamos uma abordagem de associação de sinais desses sensores na linguagem ladder.

O Produto G é o maior entre todos e é capaz de acionar tanto os três sensores horizontais quanto os dois verticais. Para identificar corretamente a presença desse produto, utilizamos uma lógica da seguinte maneira: Altura1 AND Altura2 AND Altura3 AND Largura1 AND Largura2. A Figura 40 demonstra essa lógica na linguagem Ladder.

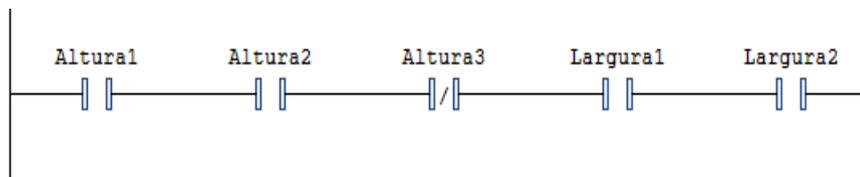
Figura 40 - Lógica de identificação do Produto G em Ladder



Fonte: Autor

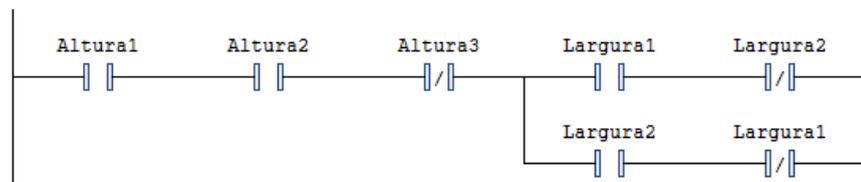
O Produto ML possui a mesma largura do Produto G, mas uma altura menor. Portanto, ela pode acionar os dois sensores horizontais mais baixos e os dois sensores verticais. Para identificar a presença deste produto, utilizamos a lógica da seguinte forma: Altura1 AND Altura2 AND !Altura3 AND Largura1 AND Largura2 conforme Figura 41.

Figura 41 - Lógica de identificação do Produto ML em Ladder



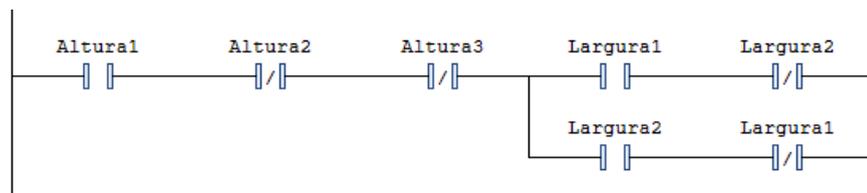
Fonte: Autor

O Produto M possui a mesma altura que o Produto ML, mas uma largura menor, acionando os dois sensores horizontais mais baixos e um sensor vertical. Portanto, para identificar corretamente a presença deste produto, utilizamos uma lógica da seguinte maneira: Altura1 AND Altura2 AND !Altura3 AND (Largura1 XOR Largura2) conforme Figura 42.

Figura 42 - Lógica de identificação do Produto M em Ladder

Fonte: Autor

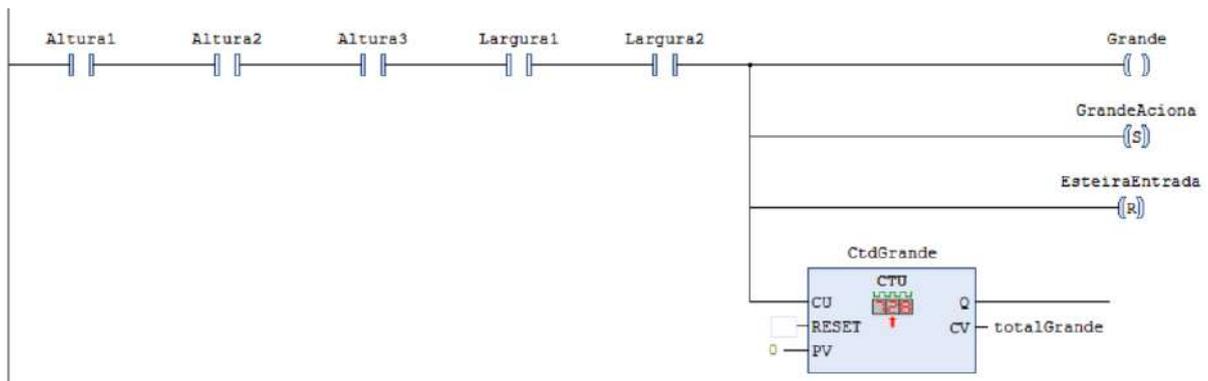
O Produto P é capaz de acionar apenas o sensor horizontal mais baixo e um dos sensores de largura. Para identificar a presença deste produto, utilizamos uma lógica da seguinte maneira: $\text{Altura1 AND !Altura2 AND !Altura3 AND (Largura1 XOR Largura2)}$, conforme Figura 43.

Figura 43 - Lógica de identificação do Produto P em Ladder

Fonte: Autor

Quando um desses produtos é identificado, o sistema deve seguir para a etapa de separação, logo, devem ocorrer alterações nas variáveis dos atuadores e contadores relacionados à separação e contagem. Tomando o Produto G como exemplo, quando sua expressão lógica é verdadeira, uma Bobina é acionada para operar na variável "Grande", uma Bobina Set é acionada para operar na variável "GrandeAciona", uma Bobina Reset atua na variável "EsteiraEntrada", e, simultaneamente, um pulso de contagem é gerado no contador associado a essa expressão. Conseqüentemente, o contador incrementa seu valor, o separador de braço articulado destinado ao desvio dos Produtos G é ativado, o led indicador de identificação acende, e a esteira de entrada é parada, interrompendo o fornecimento de produtos para o portal de identificação. A Figura 44 apresenta o código em linguagem ladder para este processo no Produto G.

Figura 44 - Lógica de identificação e atuação para o Produto G em Ladder



Fonte: Autor

Ao desviar o produto para uma das rampas de destino, o sensor reflexivo com área de atuação sobre as quatro rampas tem seu feixe interrompido pela passagem da caixa. O sinal fornecido por esse sensor indica que a separação foi concluída e o sistema pode retornar para a etapa de identificação. Portanto, acionamos uma Bobina Reset para operar nas variáveis "AcionaGrande", "AcionaMediaLarga", "AcionaMedia" e "AcionaPequena", bem como uma Bobina Set para operar na variável "EsteiraEntrada". Conseqüentemente, os Separadores de Braço Articulado que estiverem ativos retornam à posição inicial, e a esteira de entrada retoma o fornecimento de produtos para o portal de identificação, reiniciando o processo. A Figura 45 ilustra esta última expressão em linguagem ladder.

Figura 45 - Lógica de reset de atuação para o Conjunto 1 em Ladder



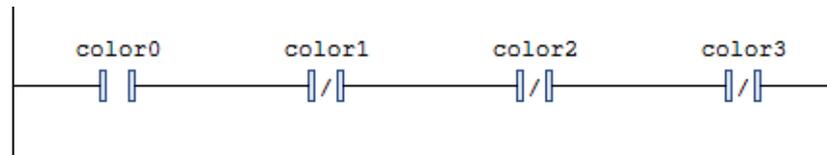
Fonte: Autor

5.3.2 Identificação e separação por cor

No processo de tratamento de caixas de cores variadas, será utilizado apenas um sensor de visão disponível no simulador. Este sensor retorna quatro bits de sinal digital que fornecem informações para identificação direta dos produtos, especialmente no que se refere à diferenciação por cor, que é de interesse neste escopo.

Para identificar os diferentes produtos no portal de identificação, foram utilizados os quatro sinais fornecidos pelo sensor. O Produto Azul, por exemplo, é representado pelo valor binário 1000, que é a combinação dos quatro bits retornados pelo sensor. Portanto, a lógica de identificação para a cor azul será: Cor1 AND !Cor2 AND !Cor3 AND !Cor4, conforme representado na Figura 46.

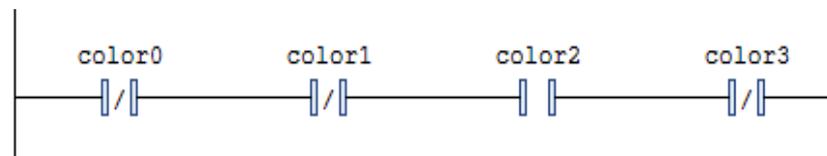
Figura 46 - Lógica de identificação do Produto Azul em Ladder



Fonte: Autor

Da mesma forma, o Produto Verde é representado pelo valor binário 0010, e a lógica para identificação será: !Cor1 AND !Cor2 AND Cor3 AND !Cor4, como ilustrado na Figura 47.

Figura 47 - Lógica de identificação do Produto Verde em Ladder



Fonte: Autor

Por fim, o Produto Cinza é representado pelo valor binário 1110, e a lógica de identificação correspondente será: Cor1 AND Cor2 AND Cor3 AND !Cor4, como apresentado na Figura 48.

Figura 48 - Lógica de identificação do Produto Cinza em Ladder



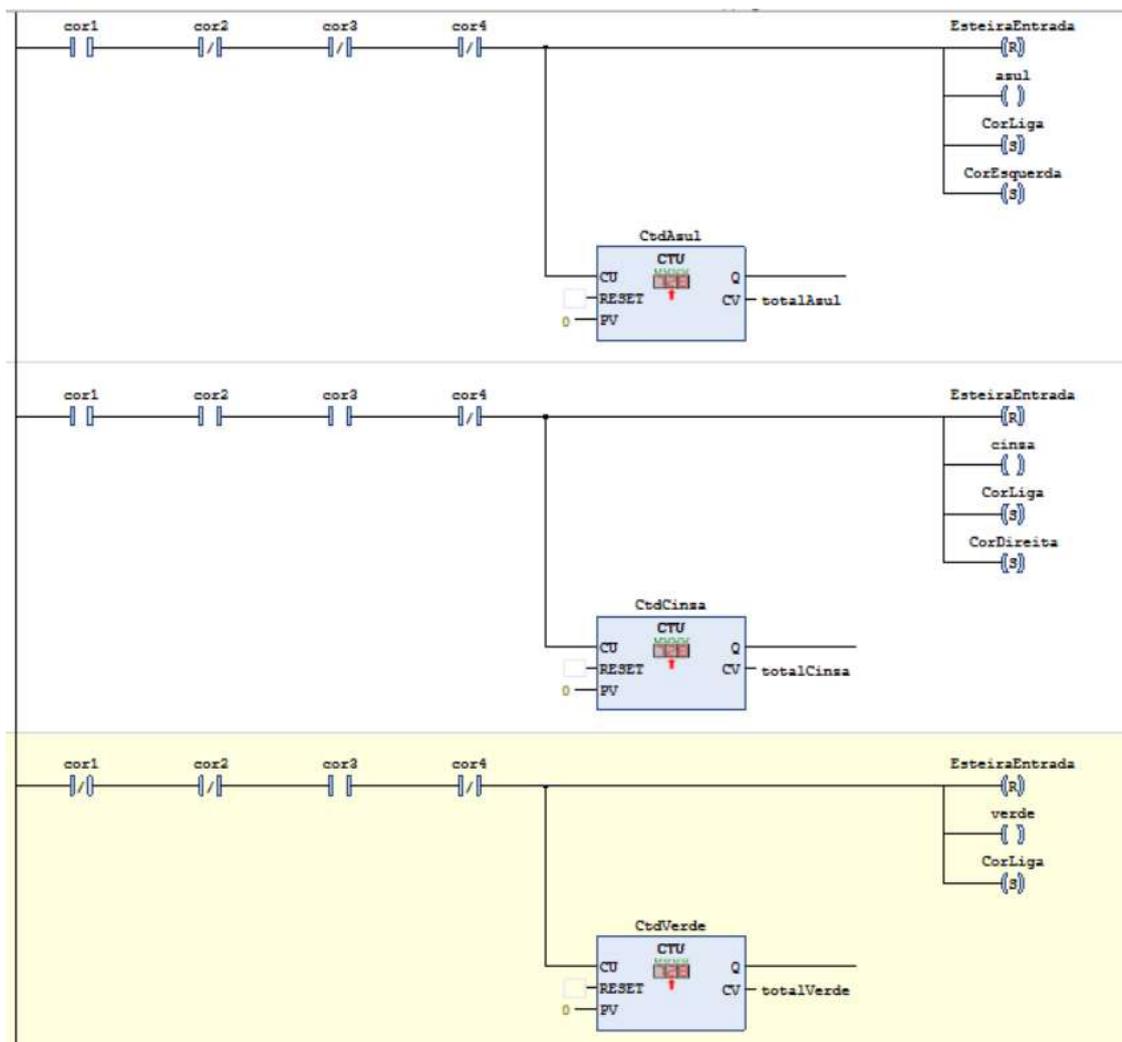
Fonte: Autor

Assim como no processo de identificação por tamanho, a identificação de qualquer uma dessas matérias-primas resultará em alterações nas variáveis que controlam os atuadores e contadores relacionados à separação e contagem desses produtos. Considerando o Produto Azul como exemplo, quando sua expressão lógica se torna

verdadeira, uma Bobina é acionada para alterar a variável "Azul", uma Bobina Set é acionada para operar nas variáveis "CorLiga" e "CorEsquerda", uma Bobina Reset opera na variável "EsteiraEntrada", e simultaneamente, um pulso de contagem é gerado no contador associado a essa expressão. Isso faz com que o contador incremente seu valor, o Led indicador de identificação acende, o Classificador de Roda Pop-Up ative as rodas e as direcione para a esquerda, e a esteira de entrada é interrompida, cessando o fornecimento de produtos para o portal de identificação.

É importante destacar que os comportamentos são diferentes para as outras duas cores. Para a separação do Produto Cinza, a variável "CorEsquerda" é substituída por "CorDireita" na lógica descrita acima. Já para o Produto Verde, nenhuma variável de direção é acionada, permitindo que o produto siga em frente. A Figura 49 apresenta a lógica completa para os três casos.

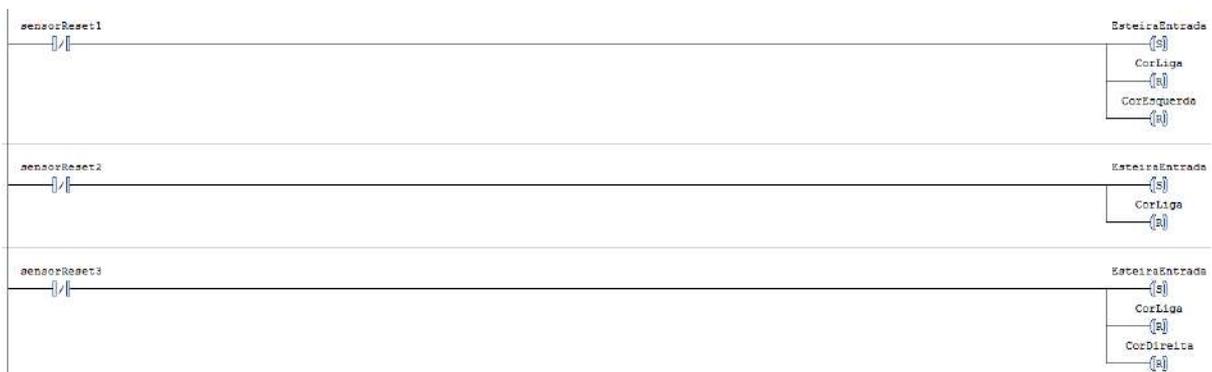
Figura 49 - Lógica de identificação e atuação para o Conjunto 2 em Ladder



Fonte: Autor

Ao direcionar o produto para uma das três rampas de destino, os sensores reflexivos em cada uma delas identificam a passagem do objeto alvo da separação. Utilizando um contato NF, detectamos o fim da separação, e o sistema retorna à etapa de identificação da seguinte maneira: para todas as rampas, uma Bobina Reset age sobre a variável "CorLiga" e uma Bobina Set age sobre a variável "EsteiraEntrada". Mais especificamente, a rampa destinada à cor cinza também envolve o *reset* na variável "CorDireita", enquanto a rampa destinada à cor azul envolve o *reset* na variável "CorEsquerda". O resultado será o retorno do fornecimento de produtos ao portal de identificação por parte da esteira de entrada e, além do desligamento das rodas do classificador, o direcionamento destas volta a ser para frente como o padrão. A Figura 50 ilustra esta etapa de finalização.

Figura 50 - Lógica de reset de atuação para o Conjunto 2 em Ladder



Fonte: Autor

5.3.3 Ações da IHM

A IHM descrita anteriormente conta com quatro botões de ação sendo eles responsáveis por gerar um sinal digital "1" enquanto pressionados e 0 enquanto soltos. Assim a variável associada ao botão será capaz de afetar os atuadores do sistema de maneira a satisfazer a referida funcionalidade e sua lógica de controle se dá por meio do uso de um contato NA devido a seu modo de funcionamento.

Para o botão rotulado como Ligar Esteiras será associada a variável "LigaEsteiras" e portanto o acionamento do botão resultará em uma ação da Bobina Set sobre a variável "EsteiraEntrada", iniciando assim o encaminhamento dos produtos até o portal de identificação.

Para o botão rotulado como Reiniciar Atuadores, será associada a variável "ReiniciarAtuadores" e portanto o acionamento resultará em uma ação da Bobina Reset sobre as variáveis "AcionaGrande", "AcionaMediaLarga", "AcionaMedia", "AcionaPequena", "CorLiga", "CorEsquerda" e "CorDireita". Isso fará com que todos os atuadores parem de ter efeito de separação no sistema.

Para o botão rotulado como Parar Esteiras, será associada a variável “PararEsteiras” e portanto o acionamento resultará em uma ação da Bobina Reset sobre as variáveis “EsteiraEntrada”, “EsteiraPrincipal1”, “EsteiraPrincipal2”. Como consequência todas as esteiras irão parar a fim de representar uma ação de emergência.

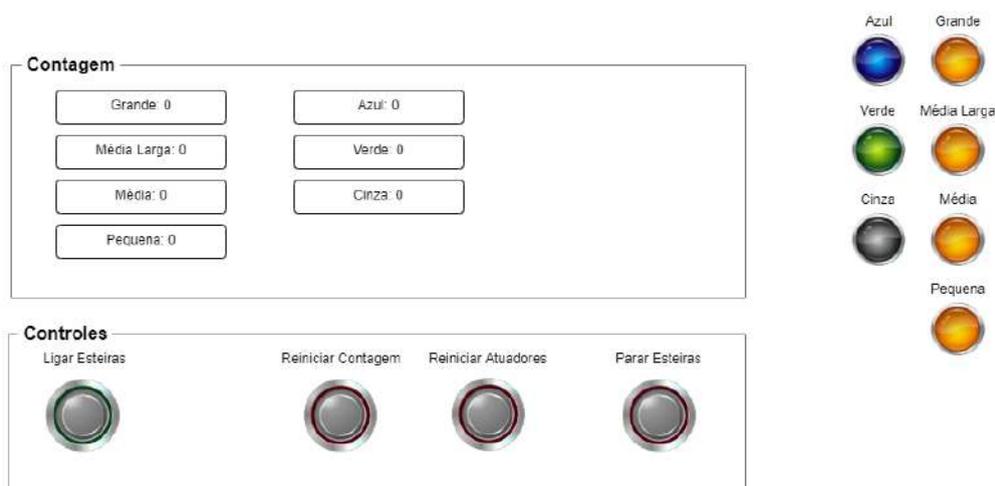
Por fim, para o botão rotulado como Reiniciar Contadores, será associada a variável “ReiniciarContadores” que por sua vez não terá um fluxo em específico que resulte na ação de uma bobina como os demais casos. Neste caso esta variável estará ligada à entrada *reset* de todos os contadores, logo, ao apresentar o valor digital “1” causará um *reset* nas contagens de todos os contadores.

6 Resultados

Após a construção do ambiente industrial no *software* Factory IO, o desenvolvimento da lógica de controle dentro do CODESYS, construção da IHM e as devidas configurações para comunicação entre as partes através do protocolo OPC, a simulação está pronta para uma execução completa.

O código desenvolvido no CODESYS foi enviado ao CLP virtual para que fosse dado início à sua execução, assim, simultaneamente, o *software* também foi capaz de inicializar a IHM para visualização e controle dos parâmetros propostos no desenvolvimento do projeto. Todos os contadores apresentaram contagem zerada, uma vez que o processo de simulação no Factory IO ainda permanecia desligado e nenhum produto havia sido identificado. A Figura 51 ilustra a IHM no momento de inicialização do CLP.

Figura 51 - IHM no CODESYS ao iniciar código no CLP



Fonte: Autor

Com a conexão estabelecida entre CODESYS e Factory IO a simulação do ambiente industrial estava pronta para dar início. Uma vez iniciada pode-se ver a esteira de entrada permanecendo parada enquanto o Emissor gera o primeiro produto aleatório, Produto M, como mostrado na Figura 52.

Figura 52 - Factory IO após início de uma simulação



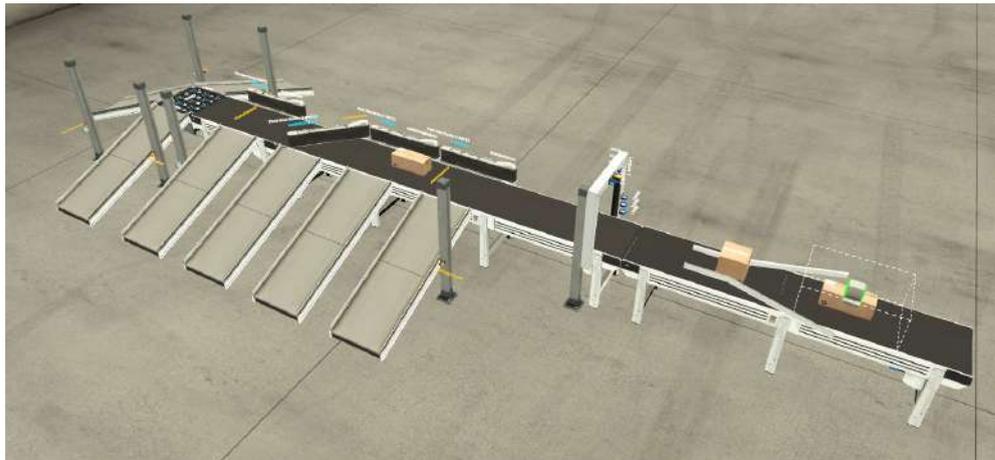
Fonte: Autor

Para que o sistema funcionasse por completo foi necessário acionar o botão Ligar Esteiras na IHM fazendo com que a esteira de entrada encaminhasse o primeiro produto até o portal de identificação que foi capaz de reconhecer o Produto M. Esta ação causou como consequência os seguintes efeitos:

- Acendimento do Led correspondente à expressão lógica de identificação deste produto enquanto este ainda passava pelo portal, se apagando logo que saiu por completo.
- Desligamento da esteira de entrada.
- Incremento no valor mostrado da contagem dos Produtos M tornando-se 1
- Acionamento do Classificador de Braço Articulado responsável por enviar este tipo de produto até a rampa de destino.

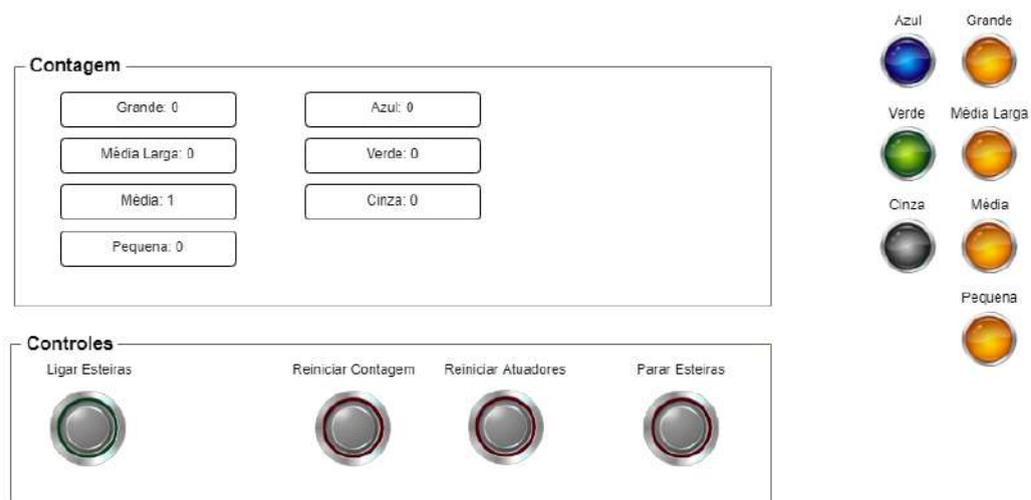
Tendo descido pela rampa e cruzado o Sensor Reflexivo outros dois efeitos foram causados retornando o sistema para a etapa de identificação, sendo eles o recolhimento do Classificador de Braço Articulado antes acionado e a ligação da esteira de entrada que passou a transportar o próximo produto. É válido ressaltar que ao o primeiro produto sair da área de atuação do Emissor, este foi capaz de gerar um novo produto aleatório que passou a ser transportado também pela esteira de entrada, mas permaneceu parado ao esta desligar devido ao reconhecimento do que estava à sua frente. A Figura 53 mostra o sistema durante a etapa de separação deste Produto M e a Figura 54 a IHM neste mesmo instante.

Figura 53 - Factory IO durante separação do primeiro produto



Fonte: Autor

Figura 54 - IHM durante separação do primeiro produto



Fonte: Autor

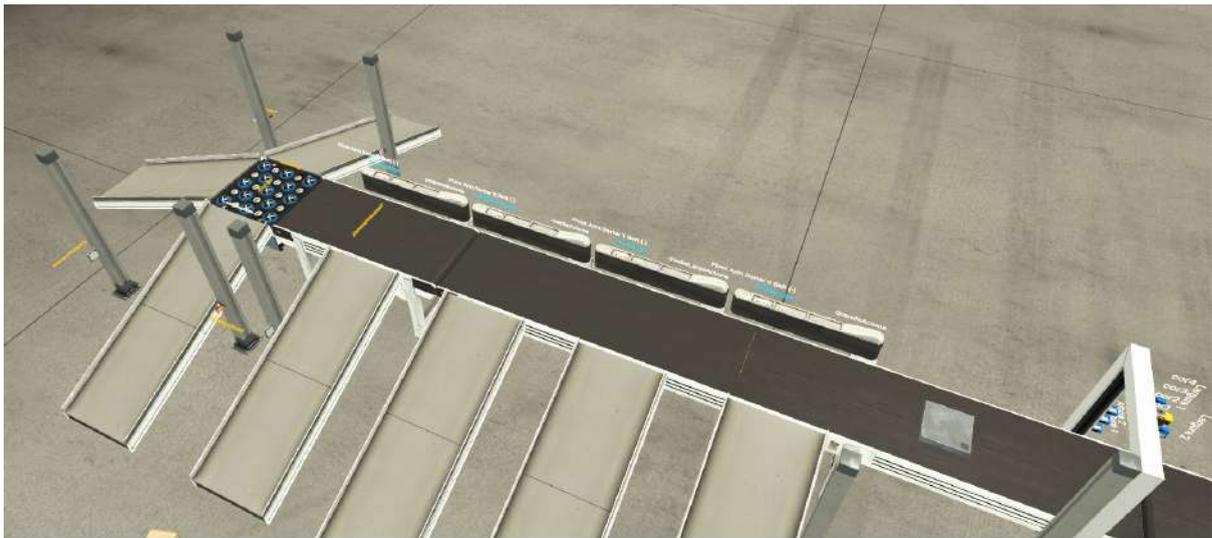
Com a simulação ainda em andamento foi o momento de ser identificado um produto do Conjunto 2, um Produto Cinza. Ao passar pelo portal e ser devidamente reconhecido pelo sensor de visão e lógica de controle impostas foram observados alguns efeitos consequentes:

- Acendimento do Led correspondente à expressão lógica de identificação deste produto enquanto este ainda passava pelo portal, se apagando logo que saiu por completo.
- Desligamento da esteira de entrada.
- Incremento no valor mostrado da contagem dos Produtos Cinza tornando-se 1

- Acionamento do Classificador de Rodas Pop Up que passou a girar e direcionou suas rodas para a direita.

Uma vez tendo descido pela rampa de destino e cruzado o Sensor Reflexivo o sistema retornou para a etapa de identificação tendo o Classificador desligado as rodas e as direcionado novamente para frente no sentido original das esteiras e a esteira de entrada ligado novamente. A Figura 55 mostra o sistema durante a etapa de separação deste Produto Cinza.

Figura 55 - Factory IO durante separação do Produto Cinza

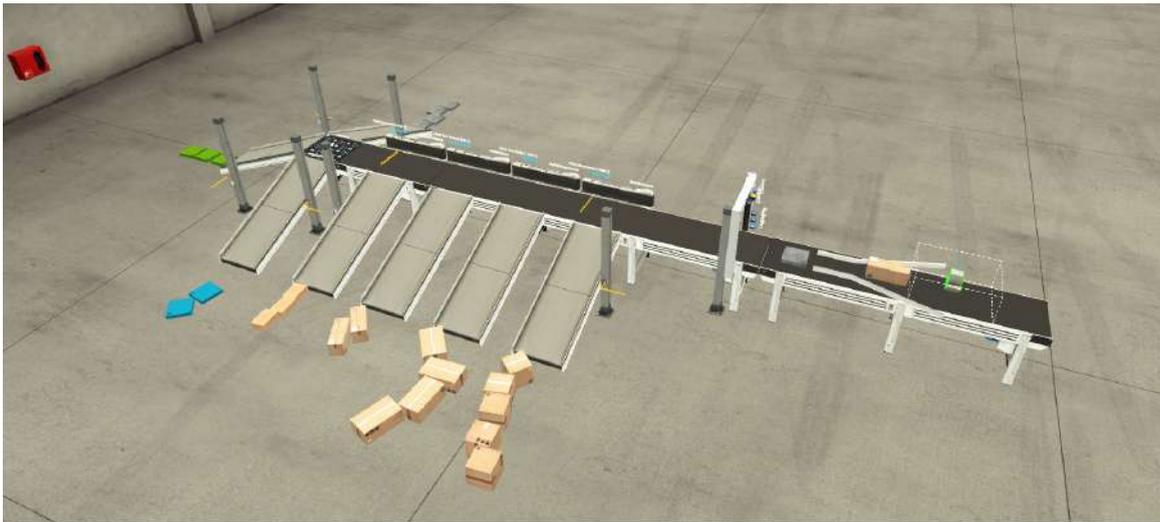


Fonte: Autor

A simulação permaneceu ativa por 5 minutos sob supervisão visual para que fosse avaliado sucesso da separação dos produtos bem como a correta contagem de cada um individualmente. Para interromper o processo permitindo assim uma análise do sistema após esse período de execução foi acionado o botão Parar Esteiras na IHM fazendo com que todas as esteiras transportadoras parassem.

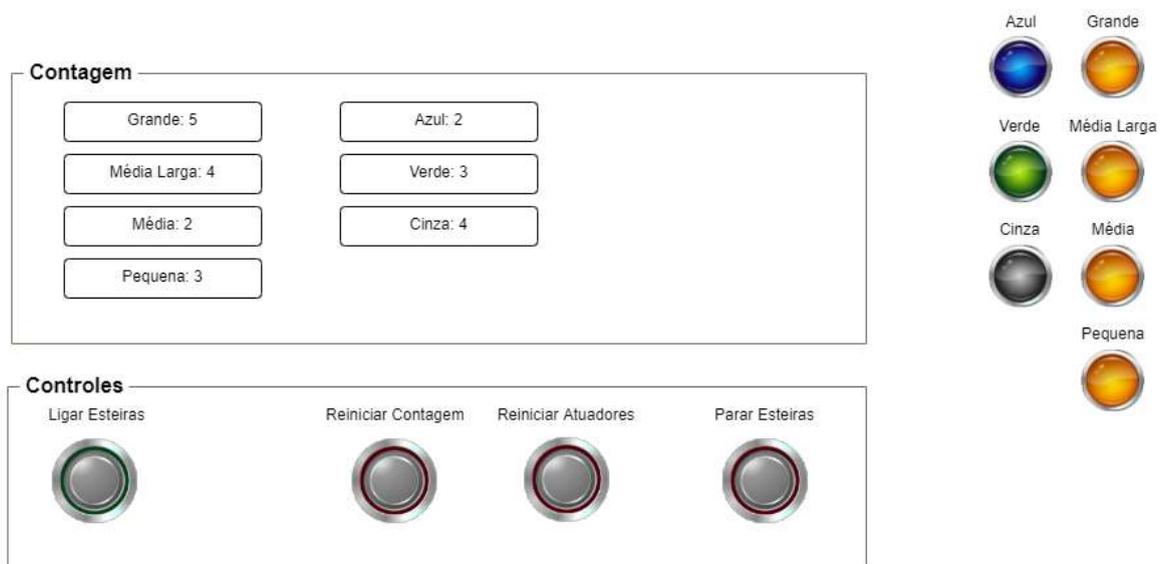
Assim como vemos na Figura 56 que mostra o ambiente no Factory IO após a pausa das esteiras e a Figura 57 que mostra a IHM neste mesmo momento, tivemos um total de 5 Produtos G, 4 Produtos ML, 2 Produtos M, 3 Produtos P, 2 Produtos Azul, 3 Produtos Verde, 4 Produtos Cinza, separados corretamente em suas respectivas regiões e com a contagem real de cada um.

Figura 56 - Factory IO após paralisação das esteiras



Fonte: Autor

Figura 57 - IHM após paralisação das esteiras



Fonte: Autor

Por fim, fomos capazes de reiniciar a simulação através dos botões de controle Reiniciar Contagem, que ao ser apertado tornou 0 o valor de contagem de todos os produtos, Reiniciar Atuadores que fez com que o Classificador de Braço Articulado fosse desativado e Ligar Esteiras para que todo o sistema se iniciasse como da primeira vez descrito acima. Vale ressaltar aqui a necessidade de remoção manual das caixas no Factory IO ainda dentro da simulação utilizando da interação com os objetos em tempo real que nos é fornecida pelo *software*.

7 Considerações finais

O objetivo deste trabalho envolveu o desenvolvimento e implementação de um sistema de identificação e contagem de produtos por tamanho ou cor, seguido de sua separação automatizada.

Para isso foi necessária a montagem de um cenário no *software* Factory IO que fosse capaz de simular o processo desejado, assim este cenário refletiria nos recursos disponíveis para o futuro desenvolvimento da lógica de controle.

A lógica por sua vez foi feita em linguagem Ladder no CODESYS para, em conjunto com o disposto no simulador industrial, realizar o controle do sistema segundo os objetivos. Por fim, visando um controle maior sobre todo o processo e visualização por meio de valores inteiros de contagem objetivou-se também a criação de uma IHM também no CODESYS por meio do CODESYS *Visualization*.

Uma vez que foi possível integrar todas as partes do projeto por meio da comunicação OPC e realizar diversas simulações, pode-se concluir que o projeto correspondeu às expectativas de funcionamento, sendo capaz de contar adequadamente a quantidade de produtos de cada tipo e separá-los em seus devidos destinos.

7.1 Trabalhos futuros

Os resultados deste trabalho trazem ideias para trabalhos futuros a serem realizados em cima do que foi aqui desenvolvido sendo assim:

- Gerar relatórios em txt ou Excel quanto aos valores coletados de contagem dos produtos e métricas como tempo gasto. Estes valores podem ser utilizados em uma ferramenta como Power BI para criação do relatório contando com uma lógica para repetição da simulação para maior amostra de dados.
- Utilizar um CLP e IHM real para execução da lógica desenvolvida e comunicação com o Factory IO, exemplo, PLC S7-1200 da Siemens.
- Utilização de entradas/saídas analógicas do CLP para permitir o controle de elementos do Factory IO, como a esteira por exemplo, por meio de valores analógicos, neste caso ajustando a velocidade da esteira e não apenas ditar se está ou não funcionando como no caso digital.
- Considerar o sistema como um Objeto de Aprendizagem e criar um material didático para ensino de conteúdos relacionados.
- Avaliar a aplicação das ferramentas Factory IO e CODESYS na simulação e estudo de sistemas usados na metalurgia.

Referências Bibliográficas

Bega, Egídio Alberto. et al. **Instrumentação Industrial**. 3ª edição. Rio de Janeiro: Interciência, 2011.

Blog Sense, 2019. **Sensores fotoelétricos - O que são? Quais os tipos?** Disponível em: <<https://www.blog.sense.com.br/2019/02/sensores-fotoeletricos-o-que-sao-quais.html>>. Acesso em: 02, nov. 2023.

CODESYS, 2022a. **CODESYS Control Win SL** Disponível em: <<https://store.codesys.com/en/codesys-control-win-sl-bundle.html>>. Acesso em: 03, nov 2023.

CODESYS, 2022b. **CODESYS Visualization**. Disponível em: <<https://www.codesys.com/products/codesys-visualization.html>>. Acesso em: 03, nov 2023.

Cravo, Edilson. **Codesys: O que é e quais são as aplicações**. Kalatec, 2023a. Disponível em: <<https://blog.kalatec.com.br/codesys/>>. Acesso em: 11, jun. 2023.

Cravo, Edilson. **O que é controlador lógico programável(CLP) e como funciona**. Kalatec, 2023b. Disponível em: <<https://blog.kalatec.com.br/controlador-logico-programavel/>>. Acesso em: 22, maio. 2023.

Cravo, Edilson. **O que é IHM e para que serve? Veja vantagens e aplicações!**. Kalatec, 2023c. Disponível em: <<https://blog.kalatec.com.br/o-que-e-ihm/>>. Acesso em: 23, maio 2023.

Cravo, Edilson. **OPC UA: o que é, como funciona e uso na automação industrial**. Kalatec, 2023d. Disponível em: <<https://blog.kalatec.com.br/opc-ua/>>. Acesso em: 03, nov 2023.

Factory I/O, 2023a. **About**. Disponível em <<https://docs.factoryio.com/>>. Acesso em: 19, maio. 2023.

Factory I/O, 2023b. **Sensors**. Disponível em <<https://docs.factoryio.com/manual/parts/sensors/>>. Acesso em: 18, maio. 2023.

Factory I/O, 2023c. **Emitter**. Disponível em <<https://docs.factoryio.com/manual/parts/emitter/>>. Acesso em: 02, nov. 2023.

Fonseca, Marcos de Oliveira. **Comunicação OPC – Uma abordagem prática**. VI Seminário de Automação de Processos, Associação Brasileira de Metalurgia e Materiais, 2002. Disponível em: <<http://www.cpdee.ufmg.br/~luizt/seixas/PaginaSDA/Download/DownloadFiles/OPCMarcosFonseca.PDF>>.

Gouvêa, Luiz Felipe Bueno. **Sistema de análise e controle de produtos em um processo de carregamento**. Universidade Federal de Uberlândia, 2021. Disponível em: <<https://repositorio.ufu.br/handle/123456789/33845>>

Lage, Francine Avelar Damasceno. **Aplicação de um sistema automatizado de uma linha de produção MPS 500 utilizando CLP S7-1200**. Universidade Federal de Ouro Preto, 2017. Disponível em: <<https://www.monografias.ufop.br/handle/35400000/634>>

Medeiros, Eddy Lopes. **Projeto de Domótica Utilizando CLP**. Universidade Federal de Campina Grande, 2014. Disponível em: <<http://dspace.sti.ufcg.edu.br:8080/xmlui/bitstream/handle/riufcg/18366/EDDY%20LOPES%20MEDEIROS%20-%20TCC%20ENG.%20EL%c3%89TRICA%202014.pdf?sequence=1&isAllowed=y>>

PAHC, 22, abr. 2021. **Programação CLP e IHM: tudo o que você precisa saber**. Disponível em: <<https://www.pahcautomacao.com.br/programacao-clip-e-ihm-tudo-o-que-voce-precisa-saber/#:~:text=O%20que%20%C3%A9%20IHM%3F.obter%20feedback%20sobre%20seu%20status>>. Acesso em: 20, maio. 2023.

Romaine, Ed. Conveyco, 2020. **Sortation System Types, Applications & Benefits**. Disponível em: <<https://www.conveyco.com/blog/pros-cons-popular-sortation-systems/>>. Acesso em: 02, nov 2023.

Silveira, Cristiano Bertulucci. Citisystems, 26, ago. 2011. **O que é automação industrial**. Disponível em: <<https://www.citisystems.com.br/o-que-e-automacao-industrial/>>. Acesso em: 10, jun 2023.

Silveira, Cristiano Bertulucci. Citisystems, 25, jul. 2016. **Como funciona a linguagem ladder**. Disponível em: <<https://www.citisystems.com.br/linguagem-ladder/>>. Acesso em: 23, maio 2023.

Somariva, Adir Felipe; Souza, Giovani Batista. **Sensor óptico difuso PNP**. 7º Simpósio de Integração Científica e Tecnológica do Sul Catarinense, 2018.

WAZLAWICK, R. S. Metodologia de pesquisa para ciência da computação. [s.n.], 2014. OCLC: 902734376. ISBN 9788535277838. Disponível em: <<https://raulwazlawick.paginas.ufsc.br/livros/metodologia-de-pesquisa-para-ciencia-da-computacao/>>.

Zancan, Marcus Daniel. **Controladores Programáveis**. Universidade Federal de Santa Maria, 2011. Disponível em: <https://www.ufsm.br/app/uploads/sites/413/2018/11/17_controladores_programaveis.pdf>