

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
Faculdade de Engenharia Elétrica
Graduação em Engenharia de Controle e Automação

LUIZ FELIPE BUENO GOUVÊA

**SISTEMA DE ANÁLISE E CONTROLE DE PRODUTOS EM UM
PROCESSO DE CARREGAMENTO**

UBERLÂNDIA

2021

LUIZ FELIPE BUENO GOUVÊA

**SISTEMA DE ANÁLISE E CONTROLE DE PRODUTOS EM UM
PROCESSO DE CARREGAMENTO**

Trabalho de Conclusão de Curso de Engenharia de Controle e Automação da Universidade Federal de Uberlândia - UFU - Câmpus Santa Mônica, como requisito para a obtenção do título de Graduação em Engenharia de Controle e Automação.

Universidade Federal de Uberlândia – UFU
Faculdade de Engenharia Elétrica

Orientador: Prof. Dr. Renato Santos Carrijo

UBERLÂNDIA
2021

Bueno Gouvêa, Luiz Felipe

Sistema de análise e controle de produtos em um processo de carregamento/ **Luiz Felipe Bueno Gouvêa.**
– **UBERLÂNDIA, 2021**- 83 p, 30 cm.

Orientador: Prof. Dr. Renato Santos Carrijo

Trabalho de Conclusão de Curso – Universidade Federal de Uberlândia–UFU Faculdade de Engenharia Elétrica. **2021.**

Inclui bibliografia.

1. Automação Comercial
 2. Business Intelligence
 3. Comunicação OPC
 4. CODESYS
 5. Factory IO
 6. Simulação
- I. Orientador Prof. Dr. Renato Santos Carrijo. II. Universidade Federal de Uberlândia. III. Faculdade de Engenharia Elétrica. IV. Engenharia de Controle e Automação.

LUIZ FELIPE BUENO GOUVÊA

**SISTEMA DE ANÁLISE E CONTROLE DE PRODUTOS EM UM
PROCESSO DE CARREGAMENTO**

Trabalho de Conclusão de Curso de Engenharia de Controle e Automação da Universidade Federal de Uberlândia - UFU - Câmpus Santa Mônica, como requisito para a obtenção do título de Graduação em Engenharia de Controle e Automação.

Uberlândia (MG), 09 de novembro de 2021.

Banca de Avaliação:

Prof. Dr. Renato Santos Carrijo
Orientador

Prof. Dr. Renato Ferreira Fernandes Jr
Membro

Prof. Dr. Fábio Vincenzi Romualdo da Silva
Membro

AGRADECIMENTOS

Agradeço primeiramente à Deus por me guiar nessa jornada de aprendizados.

Agradeço aos meus pais Santos e Enádia pela minha criação e educação, pelo apoio e incentivo em todas as decisões tomadas e pelas conquistas que me trouxeram até aqui.

Agradeço a todos os professores da FEELT pelos ensinamentos, em especial ao meu professor e orientador Renato Santos Carrijo, pela orientação para que esse trabalho fosse concluído com o êxito esperado.

Agradeço à Marcela pelo carinho, pela companhia e por sempre se dispor a me ajudar.

Agradeço por fim, aos meus amigos que estiveram comigo durante todo o período da faculdade, nos trabalhos, estudos e nos momentos de lazer, em especial aos meus amigos de curso de Controle e Automação: Elcio Rezende, Felipe Matos, Gustavo Sales e Igor Hasparyk (Engenharia Elétrica).

"A força não vem da capacidade física, vem da vontade indomável." - Mahatma Gandhi

RESUMO

Este trabalho apresenta o desenvolvimento de um sistema de controle e monitoramento de carregamento em um processo logístico, com o objetivo de reduzir prejuízos com erros humanos, reduzir o esforço físico dos operadores e fornecer dados e análises da carga de forma automatizada. O sistema foi desenvolvido utilizando conceitos de engenharia, automação comercial, inteligência de negócios, simulação e protocolos de comunicação.

O projeto foi desenvolvido utilizando ferramentas como o CODESYS, para a simulação do controle e monitoramento do processo, o Factory IO, para a simulação do processo físico de um carregamento, e o Microsoft Excel para a exportação de um relatório com dados e análises do carregamento simulado. Os resultados obtidos com o desenvolvimento do sistema e da ferramenta de tratamento e análise dos dados também são apresentados.

Palavras chave: Automação Comercial, Business Intelligence, Comunicação OPC, CODESYS, Factory IO, Simulação.

ABSTRACT

This work presents the development of a load control and monitoring system in a logistic process, aiming to reduce losses due to human errors, reduce the physical effort by the operators and provide data and an automated analysis of the load. The system was developed using engineering, business automation, business intelligence, simulation and communication protocols concepts.

The project was developed using tools such as CODESYS, for the simulation of the control and monitoring of the process, Factory IO, for the simulation of the physical shipment process, and Microsoft Excel to export a report with data and analysis of the simulated shipment. The results obtained with the development of the system and creation of the data treatment and analysis tool are also presented.

Keywords: *Business Automation, Business Intelligence, CODESYS, Factory IO, OPC Communication, Simulation.*

LISTA DE ABREVIATURAS E SIGLAS

BI – Business Intelligence

CLP – Controlador Lógico Programável

CPU – Central Processing Unit

CSV – Comma Separated Values

CTD – Count Down

CTU – Count Up

CTUD – Count Up and Down

FDB – Function Block Diagram

IDE – Integrated Development Environment

IHM – Interface Homem-Máquina

IL – Instruction List

IoT – Internet of Things

KPI – Key Performance Indicator

LD – Ladder

OPC – OLE for Process Control

PDF – Portable Document Format

PLC – Programmable Logic Controller

QRCODE – Quick Response Code

SFC – Sequential Flow Chart

SQL – Structured Query Language

ST – Structured Text

TI – Tecnologia da Informação

VBA – Virtual Basic for Applications

WMS – Warehouse Management System

LISTA DE FIGURAS

Figura 1 - Funcionamento do CLP.	9
Figura 2 - Diagrama de funcionamento da linguagem SFC.	10
Figura 3 - Lógicas E, OU e NOT em Ladder.	12
Figura 4 - Simbologia em Ladder.	12
Figura 5 - Lógica de acionamento em Ladder.	13
Figura 6 - Operação lógica XOR em Ladder.	13
Figura 7 - Exemplo de contador crescente.	14
Figura 8 - Sintaxe da função IF em ST.	15
Figura 9 - Exemplo de combinação de linguagens usando ST.	15
Figura 10 - Cena exemplo de Linha de Produção - Factory IO.	17
Figura 11 - Exemplo de Dashboard desenvolvido no Excel.	19
Figura 12 - Configuração de variáveis compartilhadas no OPC.	22
Figura 13 - Configuração de tipo de comunicação no Factory IO.	22
Figura 14 - Configuração das variáveis na comunicação OPC (Factory IO).	23
Figura 15 - Exemplo da utilização de funções do Excel no VBA.	24
Figura 16 - Arquitetura do projeto.	26
Figura 17 - Simulação do processo no Factory IO.	27
Figura 18 - Caixas utilizadas para simulação de tipos de produtos diferentes.	28
Figura 19 - Disposição dos sensores no Factory IO.	29
Figura 20 - Identificação do Produto 1 (Impressora).	30
Figura 21 - Identificação do Produto 2 (Quebra-Cabeça).	30
Figura 22 - Identificação dos Produtos 3 e 4 (Video Game e Smartphone).	31
Figura 23 - Sensor de contagem total de caixas e sensor de fim de esteira.	31
Figura 24 - Lógica do processo em SFC.	33
Figura 25 - Lógica dos contadores em Ladder.	34
Figura 26 - Acionamento de sensores de largura diferentes pelo Produto 3.	35
Figura 27 - Lógica em ST de preenchimento do Array dos produtos.	36
Figura 28 - Lógica em ST de preenchimento do Array da carga.	37
Figura 29 - Lógica de exportação do Array da carga.	38
Figura 30 - Tela de controle e monitoramento – CODESYS.	39
Figura 31 - Lógica de controle de status do processo.	41
Figura 32 - Dados da carga exportados em txt.	42
Figura 33 - Configuração de variáveis OPC no CODESYS.	43
Figura 34 - Configuração de variáveis OPC no Factory IO.	43
Figura 35 - Janela de entrada do Excel.	44
Figura 36 - Arquitetura do código em VBA.	45
Figura 37 - Dados do carregamento importados no Excel.	46
Figura 38 - Modelo padrão de relatório desenvolvido no Excel.	48
Figura 39 - Tabela de flexão de texto.	49

Figura 40 - Cálculo de Peso e Fillrate da Carga.....	49
Figura 41 - Imagens dos produtos utilizados no Mapa da Carreta.	50
Figura 42 - Exemplo de preenchimento do Mapa da Carreta.	51
Figura 43 - Lógica de exportação do relatório em PDF utilizando VBA.....	52
Figura 44 - Exemplo de relatório exportado em PDF.	53
Figura 45 - Lógica de limpeza dos dados em VBA.....	54
Figura 46 - Simulação do processo em andamento no Factory IO.	56
Figura 47 - Interface de controle e monitoramento do processo em andamento.	57
Figura 48 - Interface de interação com o usuário.	59
Figura 49 - Relatório final do processo.....	60

LISTA DE TABELAS

Tabela 1 - Qualificadores de saídas em SFC.....	10
---	----

SUMÁRIO

1. INTRODUÇÃO	3
1.1 Objetivo	4
1.1.1 Objetivos Específicos	4
1.2 Justificativa	5
1.3 Organização desta monografia	6
1.4 Trabalhos publicados	6
2. REVISÃO BIBLIOGRÁFICA	7
2.1 Automação Comercial	7
2.2 Simulação de Controladores Lógicos Programáveis	8
2.3 Business Intelligence (BI)	17
2.4 Comunicação OPC	20
2.5 Virtual Basic for Applications aplicado em Excel	23
3. METODOLOGIA	26
3.1 Lógica de desenvolvimento do projeto	27
3.2 Simulação no software Factory IO	27
3.3 Lógica de simulação no software CODESYS	32
3.3.1 Lógica principal do processo	32
3.3.2 Lógica de identificação e contagem dos produtos	34
3.3.3 Lógica de exportação de dados	36
3.4 Sistema de controle e monitoramento no CODESYS	39
3.5 Transferência de dados usando OPC	42
3.6 Tratamento de dados e geração de relatório usando VBA	44
4. RESULTADOS	55
5. CONCLUSÃO	62
6. BIBLIOGRAFIA	66
7. APÊNDICE	68

1. INTRODUÇÃO

No século XVIII, a busca pelo aumento de produção, a execução de tarefas em menor tempo e o aumento dos lucros impulsionaram a primeira Revolução Industrial, criando-se o primeiro conceito de automação, em que eram utilizadas máquinas a vapor, para que a produção ganhasse mais ritmo. A partir de 1870, a adoção da eletricidade e da divisão do trabalho na indústria deu início à Segunda Revolução Industrial, que também apresentou ao mundo a produção em massa através das linhas de montagem, idealizadas e implementadas por Henry Ford.

A terceira Revolução Industrial, ocorrida no século XX, foi impulsionada pela Segunda Guerra Mundial quando se desenvolveram avançadas técnicas de eletrônica para automação dos processos de produção através da introdução dos controladores lógico programáveis (PLC) e da tecnologia da informação (TI) no chão de fábrica. Essas mudanças trouxeram ganhos significativos para a indústria como redução de custos de produção, aumento de produtividade, diminuição de perdas e falhas no ciclo produtivo e consequentemente aumento de lucros e crescimento das empresas (SILVEIRA, 2018).

No cenário atual, a tecnologia e a automação já não estão mais inseridas somente na indústria, mas no dia-a-dia de todas as empresas, conhecida como Automação Comercial, que diz respeito à utilização de ferramentas e métodos tecnológicos para a automatização de tarefas corporativas. O desenvolvimento da automação inicia com a implantação de equipamentos e a substituição de procedimentos e rotinas manuais por automáticos, até chegar à utilização de ferramentas, que possibilitam mais controle e melhor administração do negócio, obtendo maior rentabilidade, competitividade, produtividade e confiabilidade (ISHIZAKI, 2009).

Por fim estamos entrando na era da Indústria 4.0, em que existe o aumento da informatização nos meios de produção, onde as estruturas físicas estão cada vez mais integradas com as redes de informação digital, proporcionando a integração de um grande número de sistemas em todos os níveis de produção, sendo possível encontrar soluções com a menor quantidade de operações nas atividades (MASLARIC, 2016).

Nesse contexto, as empresas têm buscado cada vez mais o uso de tecnologias e de automações que proporcionem integração de informações no processo, redução de

custos e desperdícios, aumento de produtividade, assertividade e controle sobre vendas e faturamentos, vantagens a serem entregues pelo presente projeto.

Tendo por base o objetivo de controlar os produtos em um processo de carregamento, visando assertividade nas quantidades e tipos de materiais enviados ao cliente, a redução de prejuízos ocasionados por erros humanos e do aumento de produtividade, o presente trabalho apresentará um sistema de controle de produtos em um processo de carregamento.

1.1 Objetivo

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.

1.1.1 Objetivos Específicos

- Desenvolver uma proposta de arquitetura para aplicação do projeto.
- Criar uma lógica no software CODESYS para simular a passagem de caixas em um contador.
- Desenvolver uma visualização virtualizada do processo físico no software Factory IO.
- Estabelecer uma comunicação utilizando protocolo OPC entre o CODESYS e o software de simulação gráfica Factory IO.
- Desenvolver um sistema de controle e monitoramento no CODESYS com acionamento e inserção de dados.
- Desenvolver uma estratégia para exportação dos dados simulados e importação dos mesmos no Excel.

- Desenvolver um código em VBA para tratar os dados e apresentá-los na forma de um relatório contendo:
 1. Data
 2. Destino
 3. Número do Carregamento
 4. Lista de Produtos
 5. Quantidade de Produtos
 6. Tempo de Carregamento
 7. Mapa da Carreta
 8. Peso da Carga
 9. Fillrate da Carreta

1.2 Justificativa

O presente projeto tem por motivação a oportunidade de reduzir prejuízos com erros de carregamento a partir de um controle visual e assertivo da carga, para que o desvio possa ser identificado com antecedência, diminuindo o custo e tempo gastos com investigações e tratativas. Além disso, a presença de uma esteira reduz o tempo e esforço físico do operador no trânsito de pallets até o interior da carreta. O projeto também tem por motivação abrir portas para análises importantes para a logística como produtividade, taxa de ocupação da carreta e peso, utilizando automação.

No ambiente logístico, nota-se que a automação tem ganhado espaço significativo buscando assertividade, facilidade e principalmente produtividade. Desde aplicações simples como planilhas com Macros em VBA automáticas até o uso de robôs na armazenagem e separação de cargas em porta-pallets, impactam diretamente no tempo e qualidade de serviços prestados.

Tendo em vista a importância de trazer para o meio acadêmico discussões e problemas enfrentados não só no ambiente industrial, mas também em outras áreas, esse trabalho tem por finalidade apresentar um sistema capaz de fornecer informações importantes de uma carga, de forma assertiva e segura, além de ajudar na disseminação de pesquisas aplicadas nesse meio.

1.3 Organização desta monografia

Esta monografia possui 7 capítulos, sendo estruturada da seguinte forma: o capítulo 1 apresenta as justificativas e objetivos do projeto, o capítulo 2 apresenta a revisão de trabalhos e estudos nessa área, o capítulo 3 apresenta o desenvolvimento do projeto, os capítulos 4 e 5 apresentam os resultados e conclusões, o capítulo 6 apresenta a bibliografia de apoio utilizada para o desenvolvimento do projeto e, por fim, o capítulo 7 apresenta o código desenvolvido em VBA no Excel.

1.4 Trabalhos publicados

GOUVEA, L. F. B; CARRIJO, R. S. **SISTEMA DE CONTROLE E MONITORAMENTO EM UM PROCESSO DE CARREGAMENTO**. XIX Conferência de Estudos em Engenharia Elétrica, Uberlândia: dezembro de 2021.

2. REVISÃO BIBLIOGRÁFICA

O objetivo desse capítulo é apresentar e fundamentar os principais fundamentos para a compreensão desse trabalho. Serão abordados conceitos de Automação Comercial, Simulação de Controladores Lógicos Programáveis, Business Intelligence, Comunicação OPC e Virtual Basic for Applications aplicado no Excel.

2.1 Automação Comercial

A Automação Comercial é um processo que utiliza a tecnologia para automatizar tarefas que são feitas de forma manual, através do uso de hardwares e softwares em áreas comerciais. Esses recursos visam entregar maior praticidade, segurança, qualidade e eficiência ao processo.

Antigamente, todos os controles eram feitos de forma manual, em cadernos, quadros e muitas vezes as informações se perdiam, principalmente quando era necessário armazená-las por muito tempo. Por meio da Automação Comercial, as empresas conseguem lidar com a imensa quantidade de dados gerados no dia a dia, além de terem acesso à informação necessária de forma prática e rápida (ARAÚJO, s.d.).

Os sistemas de automação são importantes em uma análise de dados detalhada, pois fornecem, de forma eficaz e assertiva, informações valiosas por meio de KPIs (indicadores-chave de performance) e até mesmo dados em tempo real. Além disso, o uso da tecnologia no meio comercial permite que o foco da empresa esteja voltado para o atendimento rápido e para a satisfação do cliente, reduzindo o tempo gasto com cálculos, cadastros de consumidores, custos, etc.

No setor logístico, em que o projeto proposto está inserido, pequenas empresas ainda fazem uso de processos e controles manuais que podem ser otimizados através da automação, buscando maior assertividade, eficiência, flexibilidade e organização.

Um exemplo de automação na área da logística são softwares do tipo WMS, sigla para Warehouse Management System, utilizados para melhorar a separação dos pedidos e o trabalho de controle de estoque nos armazéns.

Este projeto tem como objetivo utilizar o conceito de Automação Comercial para atender uma área ainda pouco explorada pela automação, o processo de carregamento. Esse processo se inicia após duas outras etapas: a separação, em que o operador faz a

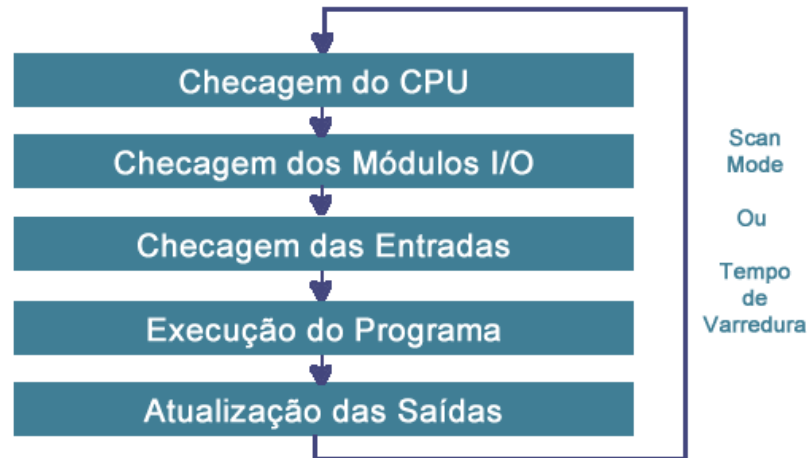
coleta dos produtos no porta-pallets e os posiciona na doca de carregamento, e a conferência, em que o operador compara a quantidade separada com a quantidade exigida no pedido (lista de separação da carga). Em seguida se inicia o processo de carregamento em que o operador faz o trânsito dos pallets separados, utilizando uma paleteira, até o interior da carreta onde o pallet é desmontado e as caixas posicionadas uma a uma. Esse tipo carga é conhecido como carga “batida”, em que as caixas são arrumadas, individualmente, uma sobre a outra e em suas próprias embalagens. Outro tipo de carga existente é a paletizada em que caixas de produtos iguais ou similares são agrupadas em cima de um pallet e envolvidas com um *stretch* (plástico fino) para sua fixação (JULIO, 2019).

2.2 Simulação de Controladores Lógicos Programáveis

Os Controladores Lógicos Programáveis são dispositivos comumente muito utilizados no ambiente industrial, eles possuem uma memória programável para armazenar internamente instruções e implementar funções específicas, tais como lógica, sequenciamento, temporização e contagem, controlando e monitorando outros equipamentos e processos por meio de módulos de entradas e saídas (ZANCAN, 2011).

Para que esse controle seja feito corretamente é preciso que o processo que se deseja controlar seja monitorado. Para isso, o CLP recebe sinais elétricos de dispositivos de entrada conectados a ele como sensores, chaves, dispositivos inteligentes, transdutores, medidores, módulos de comunicação, etc. A Unidade Central de Processamento (CPU) analisa as informações recebidas, percorre a lógica programada pelo usuário e ativa ou desativa dispositivos que estão conectados à saída do CLP e que interagem diretamente com o processo como atuadores, motores, turbinas, esteiras, etc. A Figura 1 mostra o funcionamento do CLP.

Figura 1 - Funcionamento do CLP.



A lógica de programação dos CLPs é baseada em cinco linguagens: LD (Ladder), FDB (Function Block Diagram), IL (Instruction List), SFC (Sequential Flow Chart) e ST (Structured Text). As linguagens ST e IL são textuais, pois dependem de comandos em forma de texto, já as demais são gráficas, pois existem blocos (símbolos) que contém comandos e informações prontos.

A simulação é uma técnica que se fundamenta em princípios do ensino baseado em tarefas e se utiliza da reprodução parcial ou total destas tarefas em um modelo artificial, conceituado como simulador. Em suma, o simulador tenta reproduzir um processo, equipamento ou mecanismo artificialmente (FILHO, 2007).

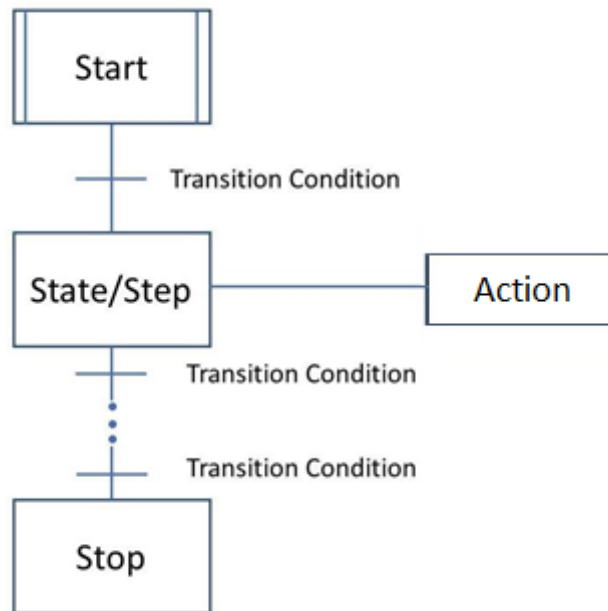
Existem programas como o CODESYS, TIA Portal, e Outseal, capazes de simular processos utilizando as linguagens de programação usadas em sistemas controlados por CLPs. Também existem softwares gráficos de simulação em 3D como o Factory IO, que podem ser conectados a PLCs, via protocolo OPC, e auxiliam no desenvolvimento de projetos de automação. Para o presente projeto será utilizado o software CODESYS para a simulação do processo, por ser um software já utilizado pelo autor durante a graduação. As linguagens utilizadas serão SFC, para a lógica principal do projeto, Ladder e ST, utilizadas nas lógicas específicas de cada etapa.

A linguagem SFC é comumente utilizada e mais fácil de representar cenários onde há vários estados de operação. Uma vantagem é sua capacidade de ajudar a quebrar processos grandes e complexos em partes menores que são fáceis de entender. Nesta

linguagem de programação, as operações são descritas como passos (*steps*) separados e conectados sequencialmente (DEVASIA, 2020).

Na programação em SFC, os passos ou *steps* são representados por caixas retangulares. Uma linha reta vertical conecta os diferentes passos, e cada passo tem uma saída (ação) correspondente. Além disso, existe uma condição de transição os passos, de forma que o próximo passo só é executado quando a transição anterior a ele é satisfeita. Em alguns casos a transição também é representada por um retângulo pequeno. A Figura 2 mostra um diagrama de funcionamento de uma lógica em SFC.

Figura 2 - Diagrama de funcionamento da linguagem SFC.



Dentro de cada passo existem *actions* (acionamentos), representadas por um retângulo e conectadas horizontalmente ao passo. Para o controle de cada tipo de ação, são utilizados os chamados qualificadores ligados às variáveis acionadas, conforme a Tabela 1.

Tabela 1 - Qualificadores de saídas em SFC.

N	Non-stored	The action active as long as the step.
----------	------------	--

R	overriding Reset	The action is deactivated.
S	Set (Stored)	The action is activated and remains active until a Reset.
L	time Limited	The action is activated for a certain time.
D	time Delayed	The action becomes active after a certain time as long as the step is still active.
P	Pulse	The action is executed just one time if the step is active.
SD	Stored and time Delayed	The action is activated after a certain time and remains active until a Reset.
DS	Delayed and Stored	The action is activated after a certain time as long as the step is still active and remains active up to a Reset.
SL	Stored and time Limited	The is action is activated for a certain time.

As transições devem ter o valor *TRUE* ou *FALSE*, para isso podem estar vinculadas a variáveis booleanas ou conter uma série de instruções com um resultado booleano. A transição “Inicio.t>T#5s”, por exemplo, está referenciada ao passo “Inicio”, e só é satisfeita (se torna *TRUE*) quando o timer de execução do passo atinge 5 segundos.

Na programação em Ladder, utilizam-se operações lógicas de acionamento de chaves e bobinas, assim como na lógica de relé com contatos. As operações realizadas pelo CLP são baseadas em três funções lógicas fundamentais de lógica Ladder: E, OU e NÃO. Essas funções combinam variáveis binárias para formar instruções e cada função tem uma regra que determina o resultado da instrução (*TRUE* ou *FALSE*). A Figura 3 apresenta as principais lógicas utilizadas em Ladder.

Figura 3 - Lógicas E, OU e NOT em Ladder.

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

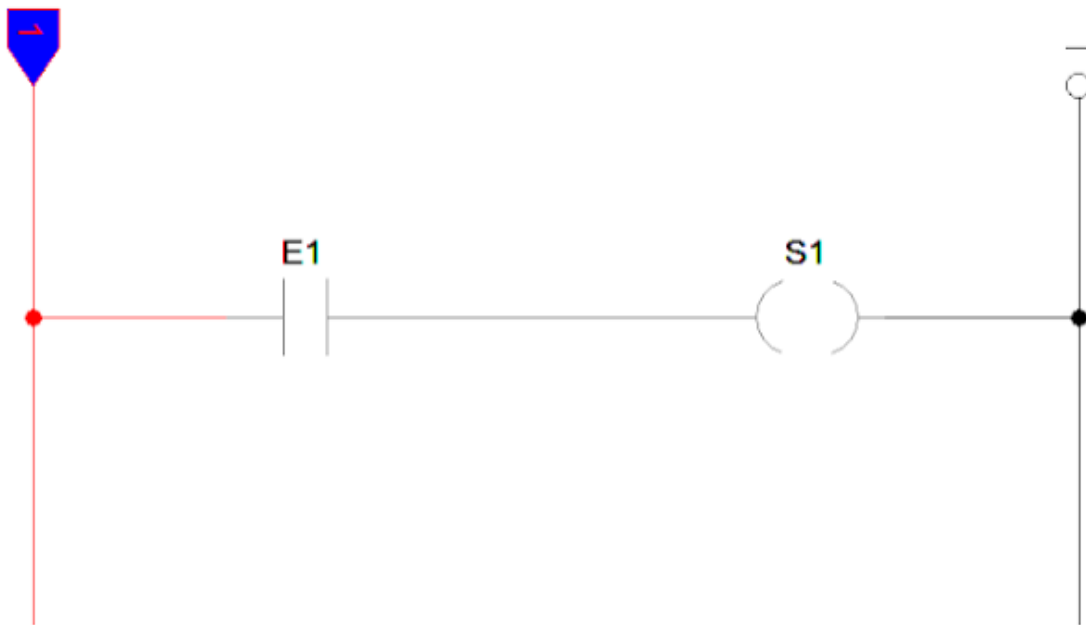
Em Ladder, cada linha horizontal existe uma lógica de associação em que os contatos representam as entradas e as bobinas as saídas, conforme Figura 4.

Figura 4 - Simbologia em Ladder.



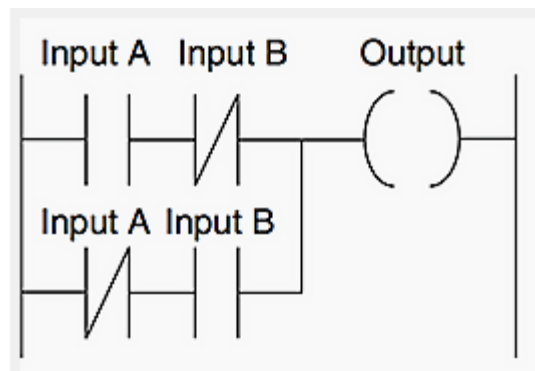
A Figura 5 demonstra uma lógica simples de acionamento em Ladder, em que o acionamento da chave E1 aciona a bobina S1.

Figura 5 - Lógica de acionamento em Ladder.



A combinação de lógicas operacionais permite a execução de lógicas mais complexas como por exemplo a operação XOR, em que uma e somente uma dentre duas variáveis deve estar acionada para que a saída seja acionada, conforme Figura 6.

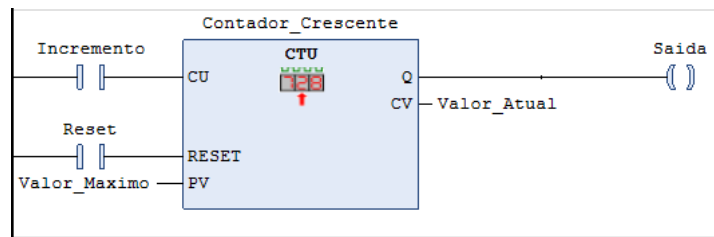
Figura 6 - Operação lógica XOR em Ladder.



Além dos dispositivos de entrada e saída mencionados também existem contadores, timers, comparadores, somadores, entre outros blocos que auxiliam nas lógicas de controle.

Os contadores são componentes simples aplicados para contar pulsos. São utilizados para indexar, incrementar ou decrementar valores. Usualmente existem três tipos de contadores CTU, CTD e CTUD: crescente, decrescente e crescente/decrescente (SILVEIRA, 2016).

Figura 7 - Exemplo de contador crescente.



Conforme visto na Figura 7 (CTU) os contadores são compostos por uma entrada (CU), uma saída (Q), uma variável de contagem atual (CV), uma variável de valor final da contagem (PV) e um *Reset*.

A entrada é responsável acionamento do contador, quando alternada de *FALSE* para *TRUE*, incrementa ou decrementa a variável Valor_Atual (CV), que por sua vez é comparada com a variável Valor_Maximo e, quando iguais, acionam a saída do contador. O *Reset*, por sua vez é responsável por reiniciar o contador.

Nesse projeto também será utilizada a biblioteca *File Access*, já inclusa no CODESYS 3.5, composta por blocos como FILE.Open, FILE.Write e FILE.Close em Ladder capazes de exportar dados para arquivos TXT.

Na programação em ST, utilizam-se comandos textuais assim como em PHP, Python e C, pois a linguagem foi desenvolvida para se parecer com a sintaxe de programação de alto nível com loops, variáveis, condições e operadores (Peter, 2015).

A utilização da programação textual é vantajosa em alguns casos por tomar menos espaço que uma programação em blocos, ser mais fácil de ler e de entender a lógica desenvolvida. Como qualquer outra linguagem textual, a programação em ST possui uma

sintaxe, isso significa que você tem que escrever de uma certa maneira para que o compilador entenda.

A principal função a ser utilizada nesse projeto é a função IF, uma função de declaração com condições em que o compilador valida uma determinada condição e, caso verdadeira, executa a tarefa imposta por aquela condição, caso falsa, valida a próxima condição e assim por diante, até a última condição. Existe uma sintaxe especial para instruções IF, conforme Figura 8.

Figura 8 - Sintaxe da função IF em ST.

```
IF <expressão booleana> THEN
    <comandos>;/*Executa caso a expressão booleana for verdadeira*/
ELSE
    <comandos>;/* Executa caso a expressão booleana for falsa*/
END_IF;
```

Uma grande vantagem da utilização da programação em ST é poder combinar diferentes linguagens: é possível ter blocos de função contendo funções escritas em ST. Também é possível comparar variáveis e até mesmo acionar dispositivos declarados em lógicas de outras linguagens, conforme Figura 9.

Figura 9 - Exemplo de combinação de linguagens usando ST.

```
IF Simulacao_SFC.Emissor = 0 THEN
    Status_Carregamento:= 'Aguardando';
END_IF
```

No exemplo acima a lógica desenvolvida em ST verifica se a variável booleana “Emissor”, declarada na lógica SFC, é falsa, e caso a validação seja verdadeira, atualiza a variável “Status_Carregamento” com o texto “Aguardando”.

O *Array* é uma estrutura de dados que contém um grupo de elementos, tipicamente esses elementos possuem o mesmo *data type* (tipo de dado como inteiro ou string). São comumente usados na programação para organizar os dados para que possam ser encontrados facilmente (TECH TERMS, s.d.). Nesse projeto o *Array* será utilizado na linguagem ST para organizar e armazenar as informações do carregamento, sendo elas:

lista de produtos carregados, data, número do carregamento, destino, tempo e status do carregamento.

Cada uma das linguagens mencionadas possui suas particularidades e algumas tarefas podem ser executadas mais facilmente em uma do que em outra, como por exemplo os contadores. A linguagem Ladder possui blocos de contadores prontos, portanto pode ser inviável a criação de uma lógica de contagem utilizando ST. Diante disso e da possibilidade de combinação das lógicas, para o presente projeto serão utilizadas as linguagens ST, Ladder e SFC combinadas afim de simular o controle do processo como um todo.

Para a simulação do processo físico será utilizado o software Factory IO, por permitir construir rapidamente um processo industrial usando uma seleção de peças e dispositivos industriais comuns, além de permitir conexão OPC. Segundo o próprio site, o cenário mais comum é utilizar o Factory IO como uma plataforma de treinamento de controle usando CLP, uma vez que são os controladores mais comuns encontrados em aplicações industriais (Factory IO Docs, s.d.).

O software conta uma grande seleção de peças inspiradas nos equipamentos industriais mais comuns que podem ser inseridas e organizadas em um ambiente 3D para criar um sistema. A fim de introduzir o usuário e apresentar o funcionamento do software também existem cenas prontas para simulação, já inclusas na instalação do programa, como por exemplo a cena de Linha de Produção, conforme Figura 10.

Figura 10 - Cena exemplo de Linha de Produção - Factory IO.



Todas as peças e dispositivos inseridos na simulação possuem variáveis específicas e podem ser nomeados pelo usuário afim de facilitar seu controle e acionamento.

O Factory IO é uma excelente ferramenta de estudo, pois permite a simulações virtuais de processos, normalmente feitas em laboratórios durante a graduação, além de permitir a comunicação externa com controladores reais e por isso será utilizada nesse projeto.

2.3 Business Intelligence (BI)

O Business Intelligence, ou BI, surgiu originalmente na década de 80 como um sistema de compartilhamento de informações nas organizações. Ele se desenvolveu para auxiliar na tomada de decisões por meio da coleta, armazenamento e análise de dados das atividades ou operações dentro de um negócio. Sua aplicação possibilita que empresas tomem melhores decisões, pois apresenta dados atuais e históricos da companhia, identificando os pontos de maior fragilidade e de maior destaque da empresa

ao longo do tempo através de indicadores, análises comportamentais, identificação de tendências, de erros, entre outros.

Na era da nova economia global, líderes corporativos são cada vez mais pressionados a entregarem inovação, diferenciação e desempenho e investem em ferramentas de BI pois entendem que a informação tem peso significativo na criação de valor de seus negócios, a partir da possibilidade do monitoramento e gestão da performance, permitindo tomadas de decisão mais efetivas (MORAIS, 2019).

O Business Intelligence engloba uma grande variedade de ferramentas e aplicativos que tornam possível a coleta, organização e análise dos dados. A ferramenta mais utilizada é o Power BI, desenvolvida pela Microsoft, que inicialmente era um suplemento do Excel. Nele é possível criar relatórios que podem ser disponibilizados online e que permitem fazer a conexão com diversas fontes de dados, como o Excel, SQL Server, arquivos CSV, etc.

O Excel é uma aplicação desenvolvida pela Microsoft em 1985 para desenvolver e editar planilhas. Essa ferramenta ainda é muito utilizada pela maioria das empresas para o tratamento de dados e desenvolvimento de análises através de fórmulas, gráficos, códigos de identificação.

O BI auxilia na organização e estruturação de um planejamento estratégico para construir vantagens competitivas sustentáveis que agreguem valor ao negócio. As vantagens de empregar ferramentas de BI são inúmeras, dando mais velocidade à tomada de decisões assertivas, potencializando a eficiência das operações e otimizando processos.

O Business Intelligence (BI) no Excel é uma excelente solução para a análise de dados de forma ágil e organizada. Por ser uma plataforma de uso intuitivo, permite cruzar as informações de interesse e criar diversos cenários, o que torna mais simples a análise preditiva e até mesmo a possibilidade de solucionar problemas rapidamente. Além disso, o Business Intelligence no Excel facilita a criação de relatórios dinâmicos, com gráficos ilustrativos, deixando as reuniões corporativas mais atrativas e de fácil compreensão.

Figura 11 - Exemplo de Dashboard desenvolvido no Excel.



A Figura 11 mostra um exemplo de dashboard no Excel utilizando BI, apresentando de forma clara e simplificada dados e resultados de uma empresa. Tais análises são muito úteis na tomada de decisões, na implementação de melhorias e mudanças no processo e na análise de dados históricos.

Para o presente projeto será utilizado o software Excel para o tratamento dos dados e geração do relatório, por meio de um código em VBA. Serão utilizadas algumas ferramentas, dentre várias presentes no Excel para desenvolvimento desse projeto, abordadas em seguida.

Para a criação de relatórios, o Excel permite a utilização de símbolos, imagens, fórmulas, caixas de texto, entre outras ferramentas capazes de auxiliar na exposição dos dados de forma rápida e clara. Um exemplo disso é a utilização de caixas de texto para apresentação de um dado presente em alguma célula.

Ao vincular uma caixa de texto à uma célula no Excel utilizando a barra de fórmulas, o conteúdo da caixa de texto é preenchido com o texto presente na célula

vinculada e, caso o valor ou texto da célula seja alterado, a caixa de texto é atualizada automaticamente, sem a necessidade de um preenchimento manual por parte do usuário.

A fim de melhorar a ilustração e o entendimento dos dados, o Excel possui uma biblioteca de símbolos, que podem ser utilizados para contextualizar os dados graficamente, facilitando seu entendimento.

A Câmera é uma ferramenta muito útil na atualização de imagens automaticamente, ela “filma” uma determinada parte da sua planilha e apresenta a imagem numa "caixa de imagem". É possível variar conforme algum critério o lugar que a câmera irá apontar para fazer essa transmissão. Com isso, é possível, por exemplo, trazer logos diferentes em relatórios, trazer imagens de produtos conforme a escolha dele, trazer gráficos diferentes conforme a necessidade, melhorar a aparência de dashboards, entre outros.

Outro ponto importante na criação de relatórios e dashboards é a segurança e veracidade das informações. Para proteger os dados das planilhas contra modificações o Excel possui a ferramenta de revisão, em que o usuário é capaz de proteger a planilha, bloqueando células, imagens, caixas de texto, para que outros usuários não sejam capazes de editar, remover, adicionar ou até mesmo selecionar informações na mesma.

Com a utilização dessas e de outras diversas ferramentas presentes no Excel, é possível desenvolver dashboards e relatórios capazes de apresentar de maneira clara e simples os dados e informações desejados.

2.4 Comunicação OPC

Há algumas décadas, a comunicação entre equipamentos, hardwares e dispositivos era feita pelos Drivers Proprietários. Eram programas de computador e protocolos específicos de cada fabricante, que não permitiam conectividade entre dispositivos, essa limitação impulsionou o surgimento do padrão OPC.

OPC, sigla para OLE for Process Control onde OLE significa Object Linking and Embedding, é um protocolo padrão desenvolvido para possibilitar a interconectividade entre dispositivos com protocolos diferentes no meio industrial. A intermediação da comunicação entre cliente e equipamento é realizada por um servidor OPC, que possui

drivers referentes aos equipamentos e faz a “tradução” dos dados para que os dois consigam se comunicar.

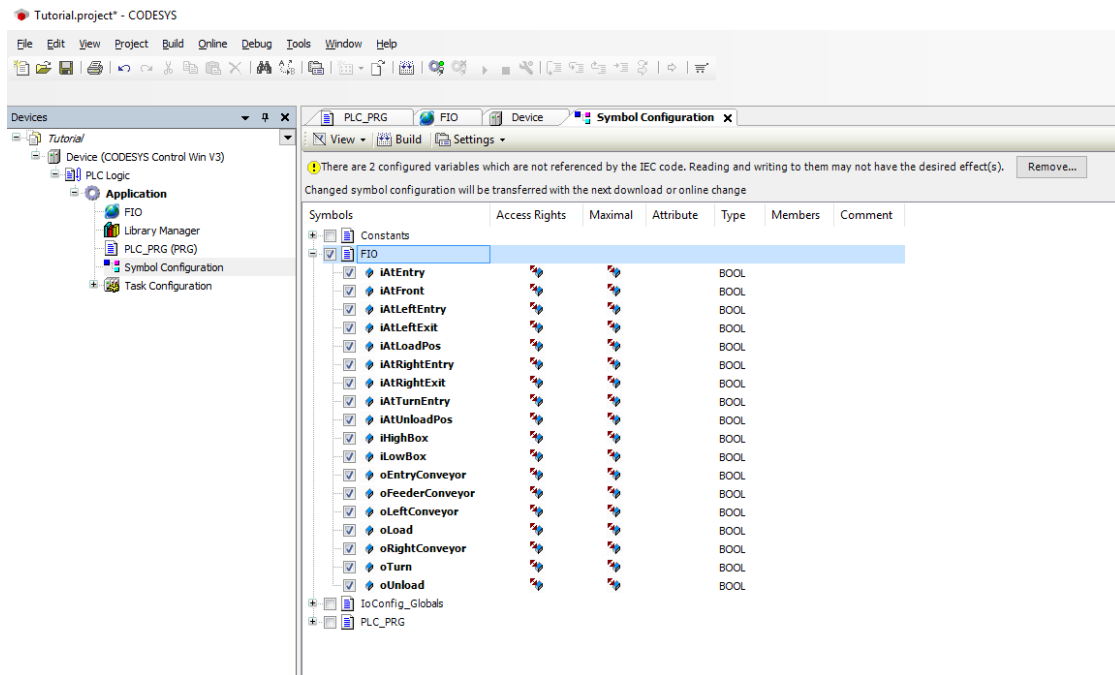
O principal ponto positivo dessa tecnologia é a sua capacidade de realizar multitarefas, permitindo a comunicação com dispositivos variados. Hoje, o OPC já está inserido no conceito de Indústria 4.0 e sua tecnologia Publisher/Subscriber já traz uma solução para IoT, buscando comunicações mais ágeis.

O Factory IO é um programa de simulação 3D para desenvolvimento de sistemas de automação, ele faz uso do OPC na comunicação com PLCs e outros softwares de simulação.

Para o projeto proposto será utilizada a comunicação OPC na comunicação e transferência de dados entre o CODESYS e o Factory IO, afim de simular o processo sistemicamente (CODESYS simulando o controlador) e visualmente (Factory IO simulando o processo físico). O principal objetivo da utilização dessa comunicação é sua facilidade de adaptação em projetos futuros e aplicações reais, já que a comunicação OPC está presente em IHMs e Sistemas Supervisórios.

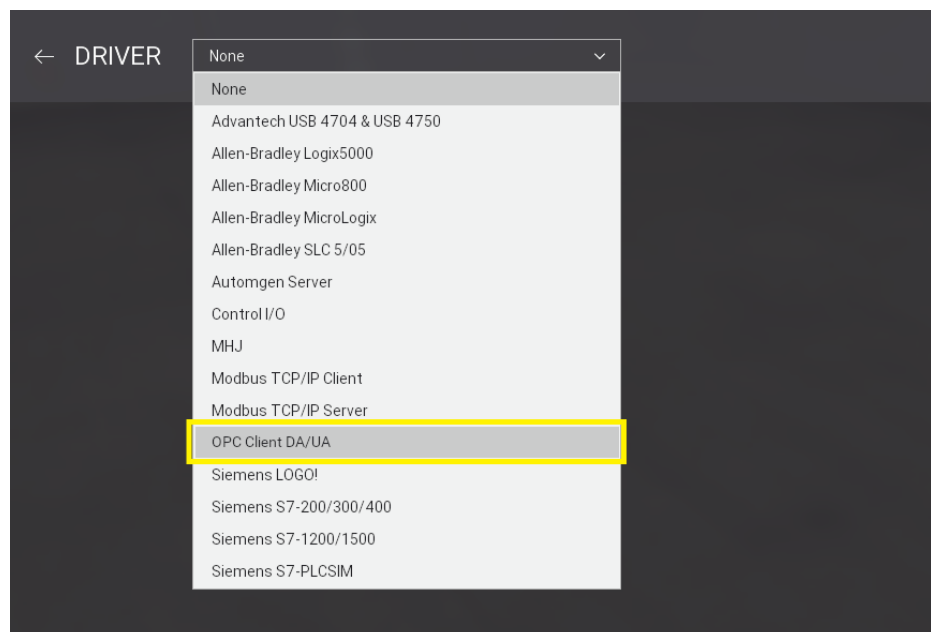
Para que a comunicação OPC no CODESYS aconteça por meio do CODESYS OPC Server é necessário utilizar o CODESYS Control Win V3. Também é importante que as variáveis a serem compartilhadas sejam configuradas no CODESYS, para isso é necessário adicionar o objeto *Symbol Configuration* com suporte à comunicação OPC UA e selecionar todas as variáveis desejadas, conforme Figura 12.

Figura 12 - Configuração de variáveis compartilhadas no OPC.



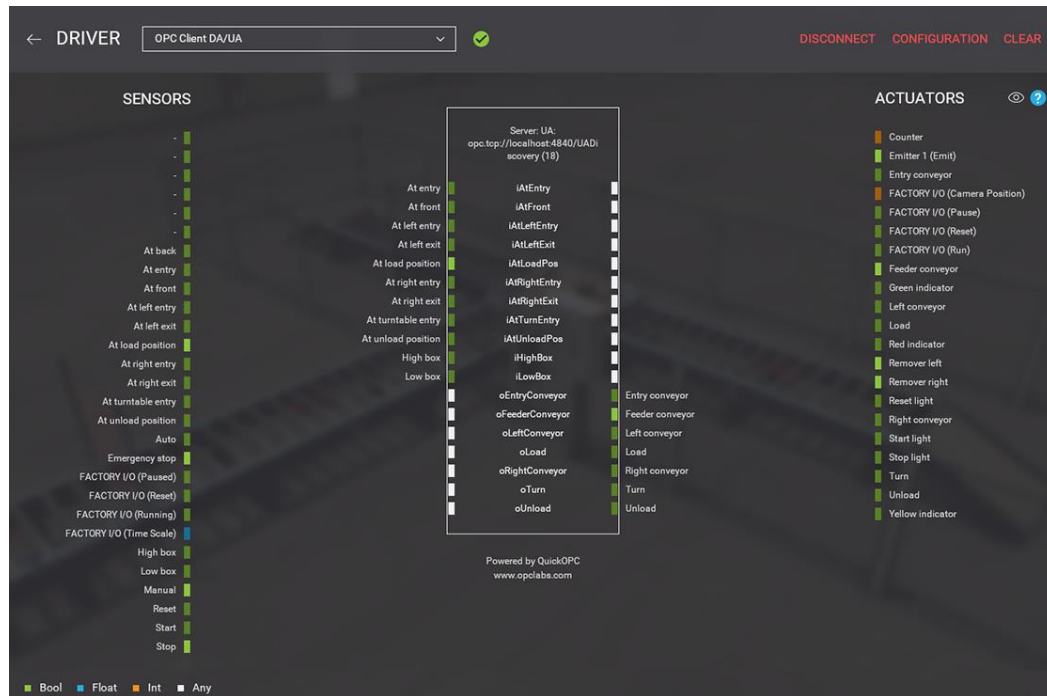
A comunicação no Factory IO acontece por meio da janela de Drivers, nela é possível escolher o controlador com o qual deseja-se comunicar e protocolos como Modbus e OPC, conforme Figura 13.

Figura 13 - Configuração de tipo de comunicação no Factory IO.



Após selecionar o OPC Server é necessário vincular as variáveis de entrada e saída de peças e dispositivos utilizados no Factory IO com as respectivas variáveis compartilhadas na comunicação OPC pelo CODESYS, conforme Figura 14.

Figura 14 - Configuração das variáveis na comunicação OPC (Factory IO).



Com isso será possível simular a lógica do processo desenvolvida no CODESYS e visualizar as etapas e o acionamento dos dispositivos em tempo real no Factory IO.

2.5 Virtual Basic for Applications aplicado em Excel

Virtual Basic for Applications, também conhecido como VBA, é uma ferramenta que pode ser usada no Microsoft Excel para gerar mais opções de controle e edição de planilhas. O VBA atua como uma linguagem de programação, permitindo a criação de macros e a automatização de diversos processos dentro das planilhas e tabelas desenvolvidas no Excel (GARRETT, 2018).

Segundo o próprio site da Microsoft Docs, a grande vantagem da programação em VBA é que é que quase todas as operações que podem ser executadas com um mouse,

teclado ou uma caixa de diálogo também podem ser realizadas usando o VBA com a mesma facilidade, centenas de vezes, afinal a automatização de tarefas repetitivas é um dos usos mais comuns dessa ferramenta.

Assim como a linguagem ST, o VBA também possui sua própria sintaxe, porém por ser uma linguagem orientada a objetos, possui algumas particularidades para cada aplicativo do Office. Por exemplo, os objetos Document e Dictionary são específicos para o modelo de objeto do Word, enquanto os objetos Workbook, Worksheet e Range são específicos para o modelo de objeto do Excel (Askari, s.d.).

O VBA possui inúmeras funções cujas sintaxes podem ser consultadas no site da Microsoft Docs, onde também são disponibilizados exemplos de aplicação a fim de auxiliar o desenvolvedor.

Um exemplo de função simples e muito utilizada é a função MsgBox, que envia uma mensagem para o usuário na forma de uma caixa de mensagem. Através dela é possível apresentar ao usuário resultado de equações e textos vinculados a variáveis ou somente transmitir uma mensagem.

Além da utilização das funções já existentes, uma grande facilidade dessa linguagem é poder utilizar as funções existentes no Excel no código VBA, conforme Figura 15.

Figura 15 - Exemplo da utilização de funções do Excel no VBA.

```
myVar = Application.WorksheetFunction.Average(5, 7, 9)
myVar2 = Application.WorksheetFunction.StDev(3, 7, 11)
```

Nesse exemplo foram utilizadas as fórmulas Average e StDev do Excel, utilizadas para calcular a média e o desvio padrão respectivamente. Também vale ressaltar que os resultados das fórmulas podem ser armazenados em uma variável, conforme visto na figura acima, e não só ser passado para uma célula da planilha no Excel.

Para o presente projeto, será usada a programação em VBA para realizar o tratamento dos dados recebidos do CODESYS e a apresentação de uma análise em forma de relatório. Para isso, serão utilizadas funções existentes no Excel para análise e

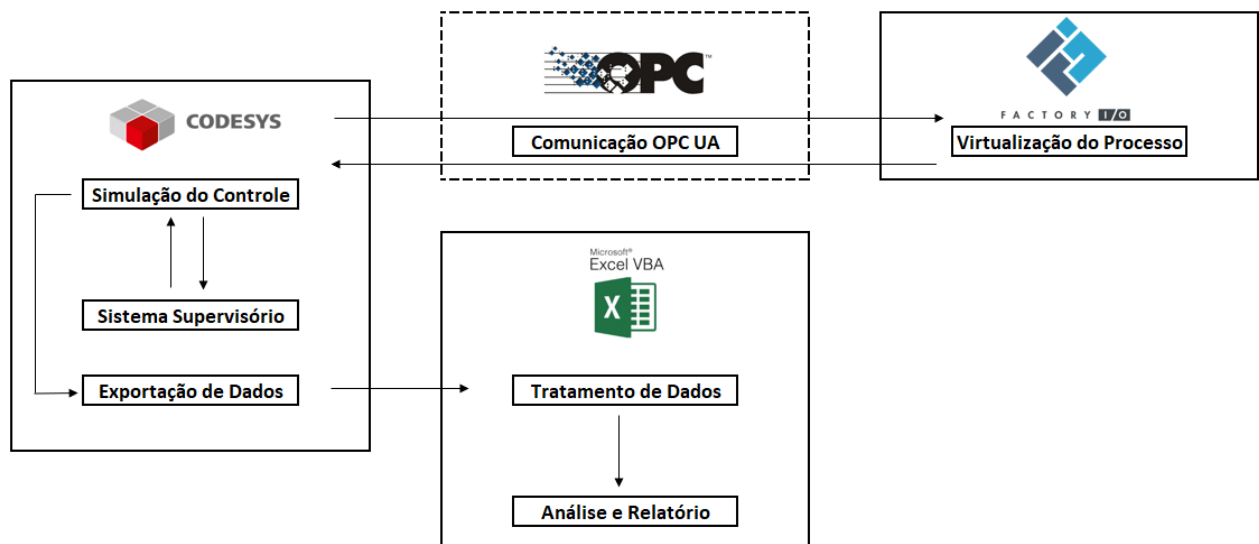
tratamento dos dados e funções do VBA para importação dos dados e exportação do relatório no formato PDF.

3. METODOLOGIA

Esse trabalho propõe desenvolver um sistema capaz de fornecer um controle assertivo de produtos carregados em uma carreta, visando reduzir o número de erros de carregamento em uma empresa de logística.

Para o desenvolvimento do projeto serão necessárias algumas etapas, como a criação de uma lógica de simulação do carregamento, a transferência dos dados coletados para o Excel, tratamento de dados e finalmente a geração de um relatório com uma análise das informações coletadas, utilizando VBA. A figura a seguir ilustra a arquitetura do projeto e as ferramentas utilizadas.

Figura 16 - Arquitetura do projeto.



De acordo com a Figura 16, para que a simulação do processo e visualização ocorram simultaneamente, o CODESYS foi conectado ao Factory IO por meio de uma comunicação OPC. Assim, quando o usuário inicia a simulação do carregamento no CODESYS o Factory IO também inicia automaticamente. Após a execução de toda a lógica de simulação, visualizada em tempo real no sistema supervisório dentro do CODESYS, os dados coletados são exportados do CODESYS para o Microsoft Excel, tratados e apresentados na forma de um relatório no através de uma Macro em VBA. Finalmente, o usuário terá a opção de exportar o arquivo em PDF finalizando o processo.

3.1 Lógica de desenvolvimento do projeto

O objetivo principal do projeto é apresentar um relatório com dados detalhados de um carregamento específico, para tal foi necessário desenvolver uma simulação do processo, em que uma esteira leva diferentes tipos de produtos até uma carreta. Diante disso foi preciso criar algumas lógicas para identificação de diferentes tipos de produtos, contagem de caixas, tratamento e exportação dos dados e para o desenvolvimento de um sistema supervisorio. Cada uma das lógicas estão detalhas as seguir.

3.2 Simulação no software Factory IO

O Factory IO é um software de simulação de automação de sistemas em 3D que permite a criação de um processo industrial utilizando dispositivos, sensores e máquinas comuns no ambiente industrial. Para a simulação desse projeto foram utilizados emissores, esteiras, sensores e removedores, conforme Figura 17.

Figura 17 - Simulação do processo no Factory IO.



O processo se inicia com o acionamento do emissor e das esteiras pelo CODESYS, via comunicação OPC, apresentada nos próximos tópicos. O emissor foi configurado para emitir um total de 10 produtos, entre 4 tipos diferentes de maneira aleatória, com uma variação de 1 a 6 segundos entre as emissões. Nessa simulação foram utilizadas 4 caixas de tamanhos distintos para simular tipos de produtos diferentes em um carregamento, conforme Figura 18.

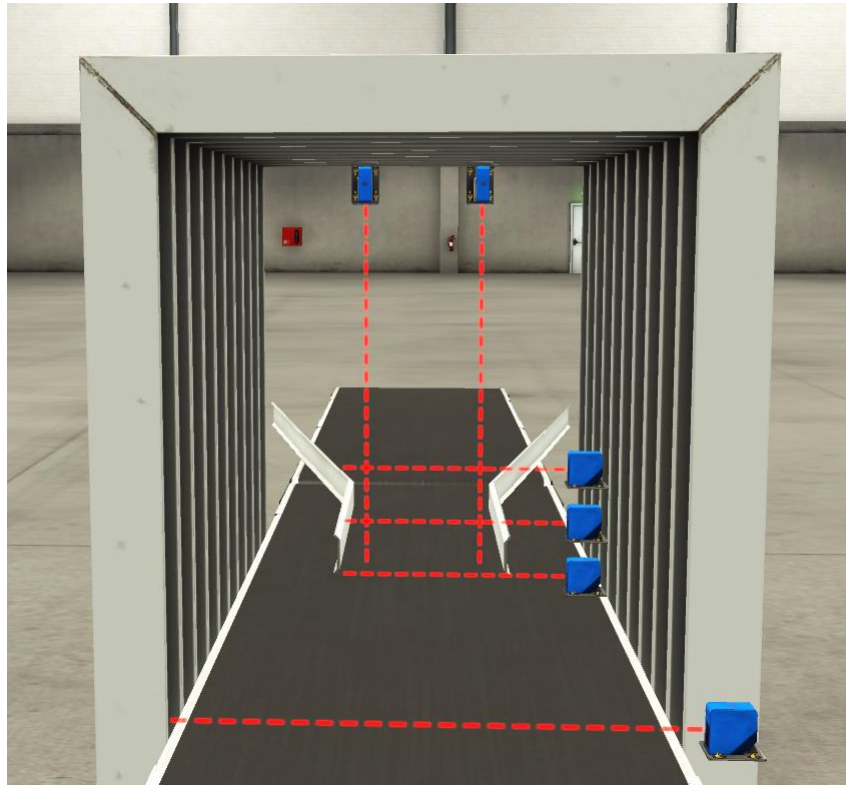
Figura 18 - Caixas utilizadas para simulação de tipos de produtos diferentes.



Para melhor identificação durante o monitoramento e análise final da carga, foram escolhidos quatro produtos reais diferentes, representados por cada uma das caixas. O Produto 1 foi considerado uma Impressora, o Produto 2 foi considerado um Quebra-Cabeça, o Produto 3 foi considerado um Vídeo Game e o Produto 4 foi considerado um Smartphone (da esquerda para a direita respectivamente).

Após o acionamento das esteiras e do emissor, o produto passa por 5 sensores difusos, dispostos estrategicamente para identificação de caixas (produtos) com tamanhos diferentes. Foram posicionados 2 sensores verticalmente, para identificar a largura e 3 sensores horizontalmente para identificar a altura das caixas, conforme a Figura 19.

Figura 19 - Disposição dos sensores no Factory IO.



Os sensores utilizados trabalham como sensores fotoelétricos, são acionados quando identificam qualquer material sólido dentro de uma distância determinada. Como as caixas possuem tamanhos diferentes, cada uma aciona uma quantidade diferente de sensores. Dessa forma é possível identificar qual caixa (tipo do produto) está na esteira, através de uma lógica no CODESYS, apresentada no próximo tópico.

Na imagem abaixo é possível perceber que a maior caixa, nomeada nesse projeto como Produto 1 (Impressora), aciona todos os sensores, o Produto 2 (Quebra-Cabeça) aciona apenas 4 deles, Figura 21. O mesmo ocorre para os Produtos 3 e 4 (Video Game e Smartphone), de acordo com seu tamanho, conforme Figura 22.

Figura 20 - Identificação do Produto 1 (Impressora).



Figura 21 - Identificação do Produto 2 (Quebra-Cabeça).

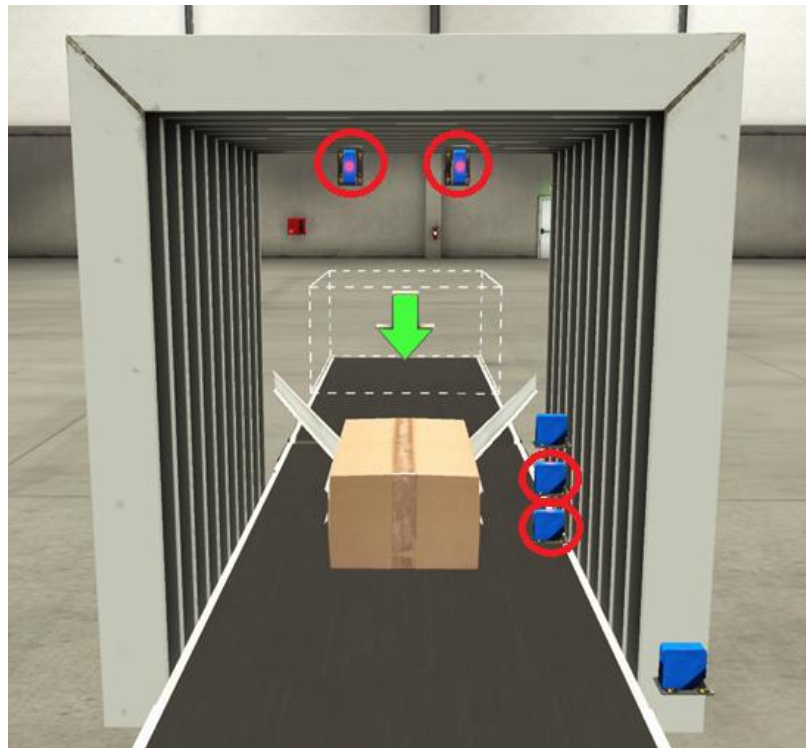
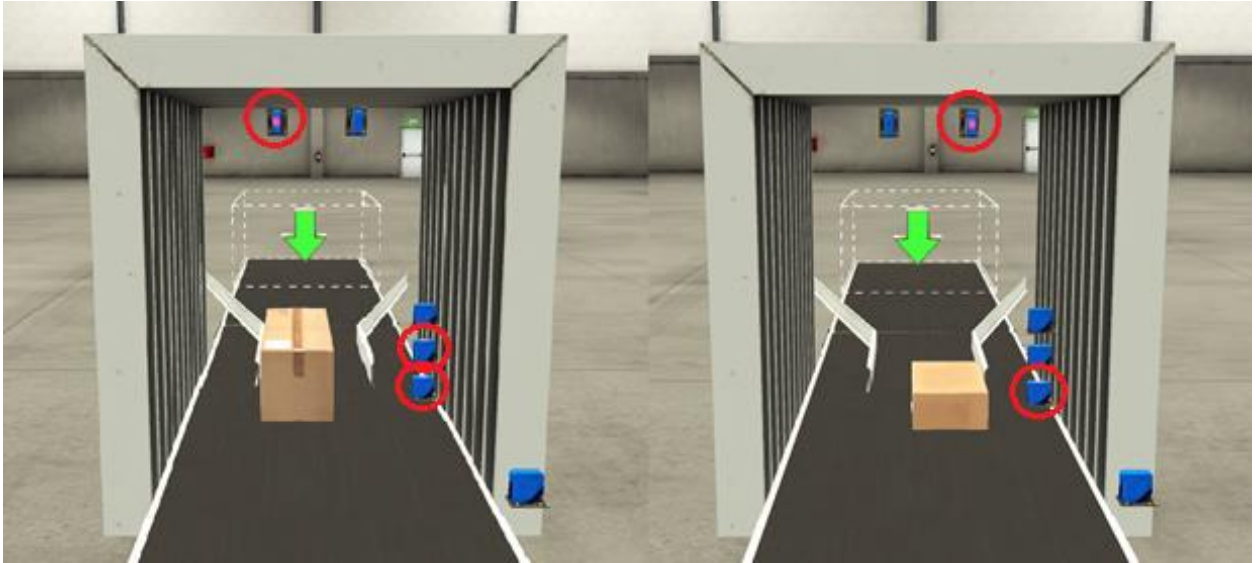
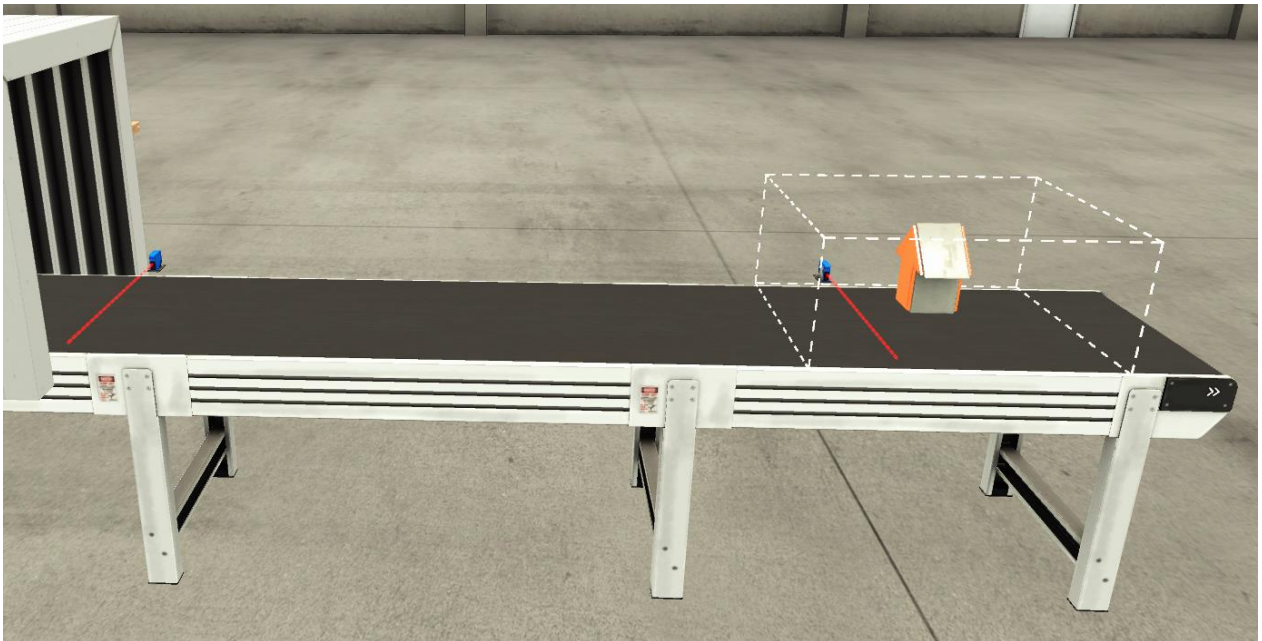


Figura 22 - Identificação dos Produtos 3 e 4 (Video Game e Smartphone).



Também foram utilizados outros 2 sensores para contagem do número total de caixas e identificação do fim da esteira, conforme Figura 23.

Figura 23 - Sensor de contagem total de caixas e sensor de fim de esteira.



Em suma, o processo de simulação desenvolvido no Factory IO se inicia com a geração de uma caixa pelo emissor, em seguida ela é transportada pela esteira,

passando pelos sensores de identificação, pelo sensor de contagem total de caixas e até ser retirada ao atingir o removedor no fim da esteira.

3.3 Lógica de simulação no software CODESYS

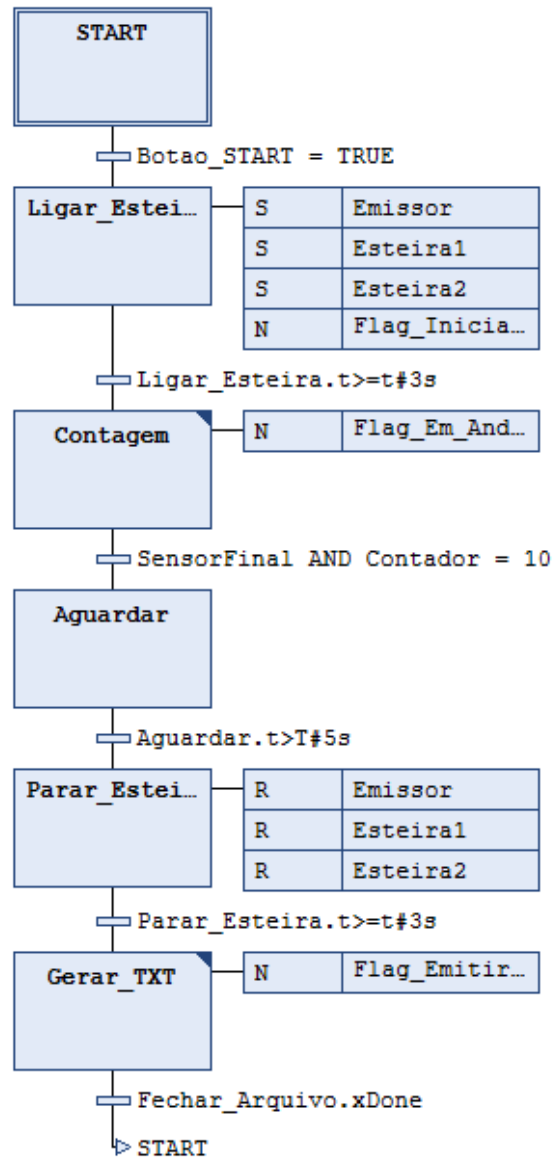
O software CODESYS faz uso de linguagens como Ladder, SFC, ST e FDB, utilizadas na programação de CLPs, para simular o controle de um processo através de uma lógica escrita ou em blocos.

Na primeira etapa de desenvolvimento desse projeto uma lógica em SFC (em blocos) simula o processo geral, já as lógicas em Ladder e ST, inseridas dentro dos blocos em SFC, simulam os processos específicos: sistema supervisorio, contagem de produtos e exportação de dados.

3.3.1 Lógica principal do processo

Na linguagem SFC a programação é feita em blocos, cada bloco (chamado de passo ou step) possui ações, que atuam sobre as variáveis declaradas, e é seguido de uma transição que, se executada, aciona um novo passo. A lógica principal do projeto foi desenvolvida em SFC contendo 6 passos (steps), sendo eles *START*, *Ligar_Esteira*, *Contagem*, *Aguardar*, *Parar_Esteira* e *Gerar_TXT*, seguidos de 6 transições conforme a Figura 24.

Figura 24 - Lógica do processo em SFC.

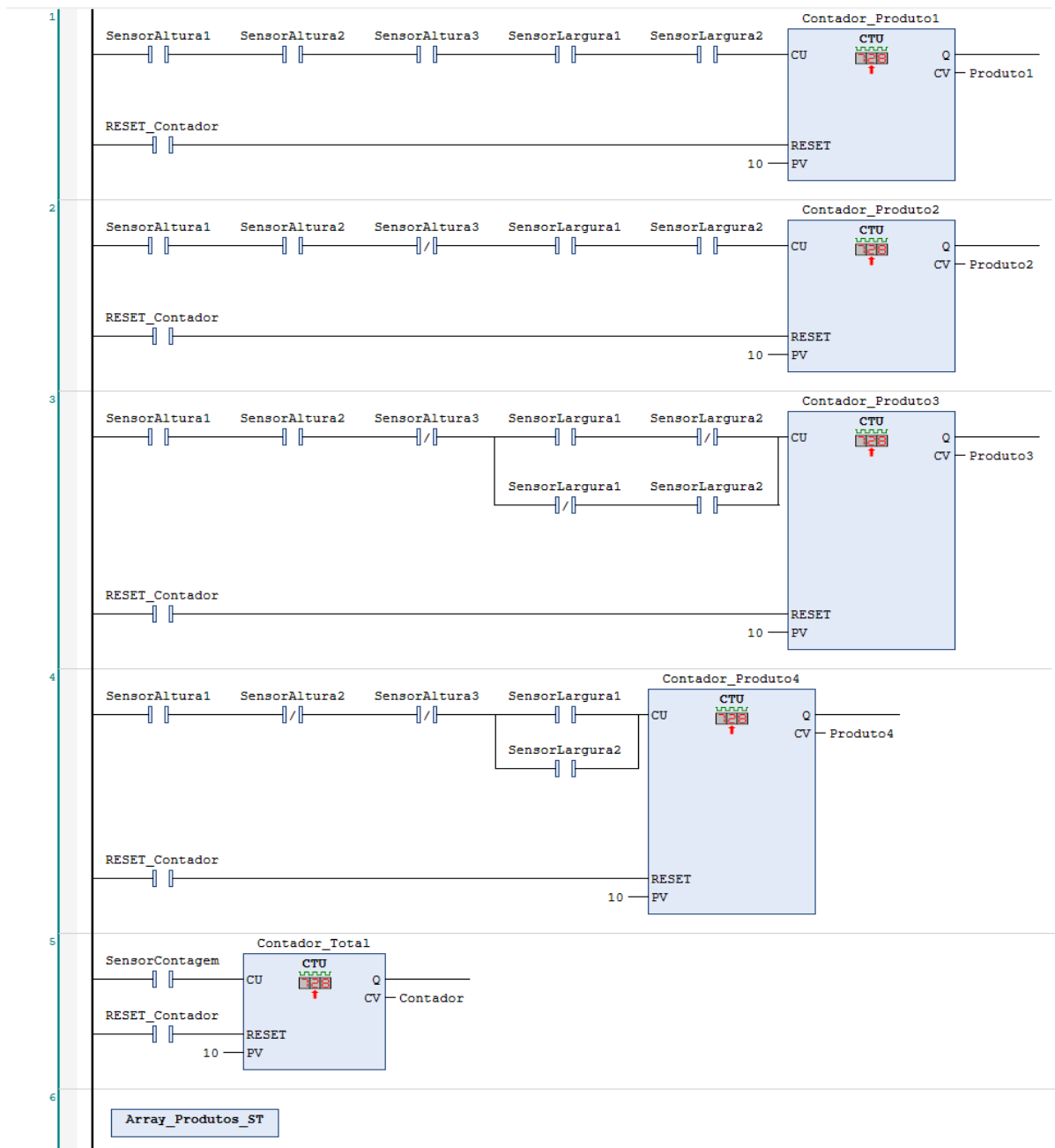


O passo “START” (step) inicia o processo e aguarda até que a o “Botão_START” seja acionado. Quando isso acontece, a transição é executada e inicia o passo “Ligar_Esteira”. O passo “Ligar_Esteira” executa suas ações, e aguarda 3 segundos, até a transição ser executada, passando para o próximo passo e assim por diante. A lógica é finalizada quando a última transição (Fechar_Arquivo.xDone) é executada retornando ao passo “START”.

3.3.2 Lógica de identificação e contagem dos produtos

Os passos “Contagem” e “Gerar_TXT” possuem identificações diferentes dos demais por se tratarem de passos compostos por uma lógica em outra linguagem. Dentro do passo Contagem foi criada uma lógica em Ladder para fazer a identificação e contagem dos produtos simulados no Factory IO conforme Figura 25.

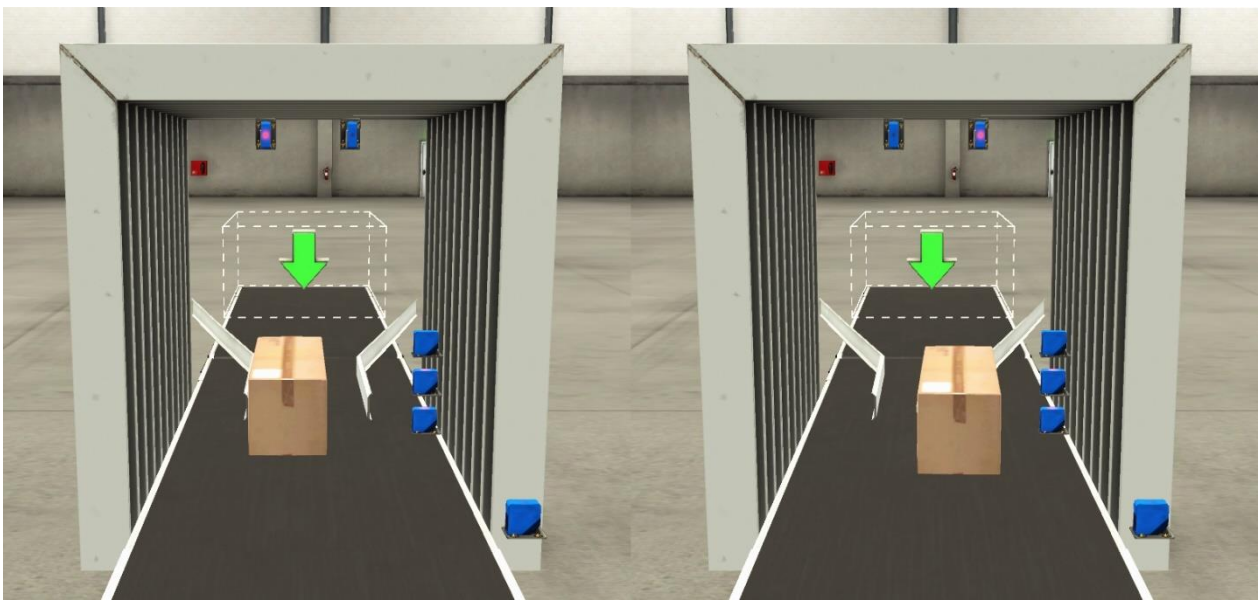
Figura 25 - Lógica dos contadores em Ladder.



Foram criados 4 contadores diferentes (Contadores_Produto1 até 4), para cada um dos tipos de produtos simulados no Factory IO. Os Contadores_Produto1 até 4 são incrementados de acordo com o produto presente na esteira simulada no Factory IO, quando o Produto 1 passa pelos sensores, por exemplo, todos os sensores são acionados, dessa forma o Contador_Produto1 é incrementado em 1. O acionamento dos sensores de cada item foi apresentado no desenvolvimento da simulação no CODESYS (tópico anterior).

Os Produto 3 e 4 são caixas estreitas e por isso acionam apenas 1 dos sensores de largura, mas não necessariamente sempre o mesmo sensor, conforme Figura 26.

Figura 26 - Acionamento de sensores de largura diferentes pelo Produto 3.



Diante disso foi necessário criar uma estratégia utilizando um operador lógico XOR (ou exclusivo) para a identificação do Produto 3 e uma lógica OU para a identificação do Produto 4, conforme visto na Figura 25.

O Contador_Total é incrementado após identificar qualquer produto, dessa forma ele também valida a quantidade final de produtos da carga.

Dentro desse passo também foi desenvolvida uma lógica em ST (Array_Produtos_ST) para preenchimento de um *Array* de acordo com a ordem de passagem dos produtos pelo contador, conforme Figura 27.

Figura 27 - Lógica em ST de preenchimento do Array dos produtos.

```

(*Reset de variaveis*)
IF Simulacao_SFC.Contador = 0 THEN
  i:= 0;
  Produto1_aux:= 0;
  Produto2_aux:= 0;
  Produto3_aux:= 0;
  Produto4_aux:= 0;
  Contador_aux:= 0;
END_IF

(*Logica para Identificar se o produto foi contado e inseri-lo no ARRAY*)
IF Simulacao_SFC.Contador = Contador_aux + 1 THEN
  (*Produto 1*)
  IF Simulacao_SFC.Produto1 = Produto1_aux + 1 THEN
    Carga[i]:=1;
    i:= i+ 1;
    Produto1_aux:= Produto1_aux + 1;
  END_IF
  (*Produto 2*)
  IF Simulacao_SFC.Produto2 = Produto2_aux + 1 THEN
    Carga[i]:=2;
    i:= i+ 1;
    Produto2_aux:= Produto2_aux + 1;
  END_IF
  (*Produto 3*)
  IF Simulacao_SFC.Produto3 = Produto3_aux + 1 THEN
    Carga[i]:=3;
    i:= i+ 1;
    Produto3_aux:= Produto3_aux + 1;
  END_IF
  (*Produto 4*)
  IF Simulacao_SFC.Produto4 = Produto4_aux + 1 THEN
    Carga[i]:=4;
    i:= i+ 1;
    Produto4_aux:= Produto4_aux + 1;
  END_IF

  Contador_aux:= Contador_aux + 1;
END_IF;

```

A cada incrementação do contador total de produtos é feita uma validação da quantidade de produtos contados e a quantidade presente no *Array* de cada produto, dessa forma o produto contado é inserido no *Array* de produtos.

3.3.3 Lógica de exportação de dados

Após o término da contagem a esteira é desligada e os dados dos produtos são passados para o *Array* da carga, conforme Figura 28, juntamente com as informações de

data, ordem de carregamento, destino, tempo de carregamento e status, dados coletados no sistema de controle e monitoramento abordado no próximo tópico.

Figura 28 - Lógica em ST de preenchimento do Array da carga.

```
(*Passagem dos dados do ARRAY de produtos em ordem para o ARRAY da Carga a ser exportado*)
IF Aux = 0 THEN
  IF i<=9 THEN
    Dados_Carga:= CONCAT(Dados_Carga,',');
    Dados_Carga:= CONCAT(Dados_Carga,INT_TO_STRING(Array_Produtos_ST.Carga[i]));
    i:= i+1;
  END_IF

  (*Escrita das informacoes de Data, Destino, Numero de Carregamento, Tempo e Status da carga.*)
  IF i = 10 THEN
    Dados_Carga:= CONCAT(Dados_Carga,'$N');
    Dados_Carga:= CONCAT(Dados_Carga,'Data:');
    Dados_Carga:= CONCAT(Dados_Carga,',');
    Dados_Carga:= CONCAT(Dados_Carga,DATE_TO_STRING(Simulacao_SFC.Data));

    Dados_Carga:= CONCAT(Dados_Carga,'$N');
    Dados_Carga:= CONCAT(Dados_Carga,'Destino:');
    Dados_Carga:= CONCAT(Dados_Carga,',');
    Dados_Carga:= CONCAT(Dados_Carga,Simulacao_SFC.Destino);

    Dados_Carga:= CONCAT(Dados_Carga,'$N');
    Dados_Carga:= CONCAT(Dados_Carga,'Ordem de Carregamento:');
    Dados_Carga:= CONCAT(Dados_Carga,',');
    Dados_Carga:= CONCAT(Dados_Carga,Simulacao_SFC.Num_Carregamento);

    Dados_Carga:= CONCAT(Dados_Carga,'$N');
    Dados_Carga:= CONCAT(Dados_Carga,'Tempo de Carregamento:');
    Dados_Carga:= CONCAT(Dados_Carga,',');
    Dados_Carga:= CONCAT(Dados_Carga,TIME_TO_STRING(Supervisorio_ST.Tempo_Carregamento));

    Dados_Carga:= CONCAT(Dados_Carga,'$N');
    Dados_Carga:= CONCAT(Dados_Carga,'Status:');
    Dados_Carga:= CONCAT(Dados_Carga,',');
    Dados_Carga:= CONCAT(Dados_Carga,'Finalizado');

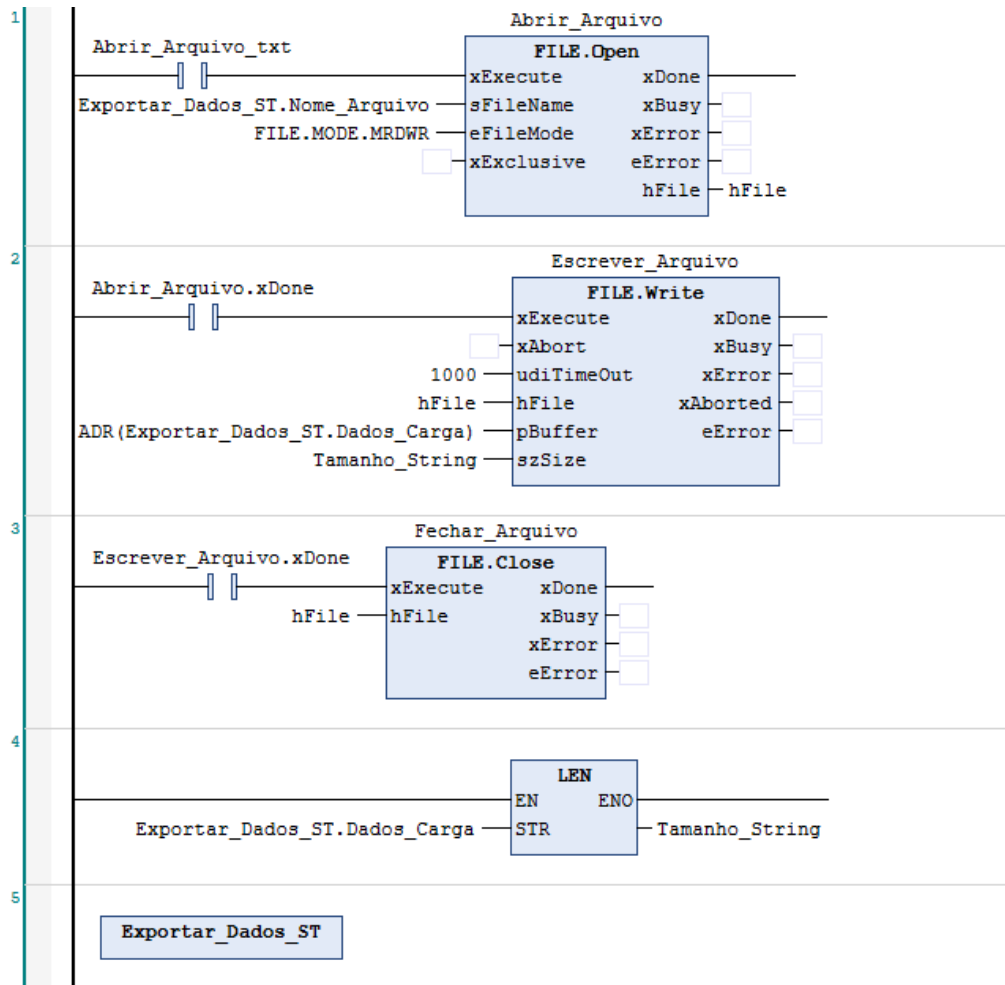
    Nome_Arquivo:= CONCAT('D:\Dropbox\Faculdade\TCC\Simulacao\Controle Carregamento Final\Cargas\'
    ,CONCAT(Simulacao_SFC.Num_Carregamento,'.txt')); (*Atualiza nome do arquivo*)

    Aux:= 1;
  END_IF
END_IF
```

A fim de facilitar a importação e separação dos dados no Excel, também foram inseridos os símbolos “\$N”, para saltar linha, e “,” para saltar coluna, no *Array* da carga.

O passo final é a exportação dos dados presentes no *Array* da carga para um arquivo TXT, para isso foi desenvolvida uma lógica em Ladder (Figura 29) utilizando a biblioteca *File Access*, inclusa no CODESYS 3.5.

Figura 29 - Lógica de exportação do Array da carga.



A lógica utiliza três blocos: *File.Open*, *File.Write* e *File.Close*, com a função de abrir, escrever e fechar o arquivo, respectivamente, de forma que a saída do primeiro bloco está ligada na entrada do segundo, dessa forma as ações são executadas em ordem. Essa tarefa se inicia após o acionamento do botão “Emitir Dados” no sistema supervisor, discutido no próximo tópico. Com o término da exportação dos dados o flag “xDone” do bloco “Fechar_Arquivo” finaliza o processo.

3.4 Sistema de controle e monitoramento no CODESYS

É de extrema importância a presença de um sistema de controle e monitoramento em processos automatizados, um exemplo disso são as IHMs, amplamente utilizadas em processos industriais. Pensando nisso, foi desenvolvido uma janela de controle e monitoramento utilizando a ferramenta de visualização do CODESYS, conforme Figura 30.

Figura 30 - Tela de controle e monitoramento – CODESYS.

Bem Vindo!

Insira os dados do carregamento e pressione Iniciar.

Iniciar



Data

← August 2021 →

Mo	Tu	We	Th	Fr	Sa	Su
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Ordem de Carregamento

Destino

Monitoramento

STATUS DO CARREGAMENTO

Aguardando



Esteira



Contador

Impressora: 0

Quebra-Cabeça: 0

Video Game: 0

Smartphone: 0

Total: 0

Emitir Dados

A biblioteca de visualização, já inclusa no CODESYS 3.5, possui ícones, medidores, *input boxes*, botões, LEDs, entre outras ferramentas comumente utilizadas

em IHMs. No controle desenvolvido foram utilizados: botão, *Date picker*, *text field*, *Lamp*, *Labels* e algumas formas geométricas.

O processo se inicia quando o usuário pressiona o botão “Iniciar”, que aciona a variável “Botão_START”, mencionada no desenvolvimento da simulação do projeto anteriormente. Ele também deve preencher os campos “Ordem de Carregamento”, “Destino” e selecionar a data no *Date picker*, pois estão vinculados a variáveis que serão inseridas no *Array* da carga no processo de exportação dos dados. Assim que o botão “Iniciar” é pressionado a lógica desenvolvida no CODESYS se inicia e é possível monitorar o processo através da janela Monitoramento, Figura 30.

O LED “Esteira” está vinculado à variável da esteira na lógica, portanto permanece ligado enquanto a variável estiver acionada.

O LED “Contador” é acionado toda vez que uma caixa passa pelo sensor de contagem total de caixas durante a simulação no Factory IO.

As caixas de texto “Impressora”, “Quebra-Cabeça”, “Video Game” e “Smartphone” representam as quantidades dos produtos e são atualizadas em tempo real de acordo com a contagem de cada um.

O “Status do Carregamento” é atualizado de acordo com a progressão da lógica do processo, para isso foram criadas variáveis *flag*, acionadas em cada etapa. Também foi desenvolvido um código em ST para cálculo do tempo de carregamento, utilizando a função TIME, e controle do status de acordo com o acionamento de cada *flag*, Figura 31.

Figura 31 - Lógica de controle de status do processo.

```

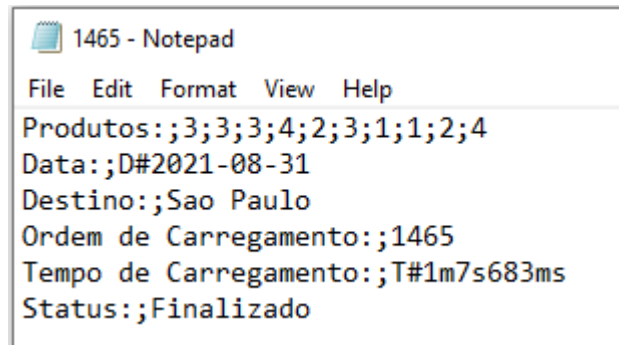
1
2 IF Simulacao_SFC.Emissor = 0 THEN
3     Status_Carregamento:= 'Aguardando';
4 END_IF
5
6 IF Simulacao_SFC.Flag_Iniciando THEN
7     Status_Carregamento:= 'Iniciando';
8     Tempo_Inicio := TIME();
9 END_IF
10
11 IF Simulacao_SFC.Flag_Em_Andamento THEN
12     Status_Carregamento:= 'Em andamento';
13 END_IF
14
15 IF Simulacao_SFC.Flag_Emitir_Dados THEN
16     Status_Carregamento:= 'Aguardando Emissão de Dados';
17     IF AUX = 0 THEN
18         Tempo_Fim := TIME();
19         Tempo_Carregamento := Tempo_Fim - Tempo_Inicio; (*Calculo do tempo de Carregamento*)
20         AUX := 1;
21     END_IF
22 END_IF
23
24 IF Simulacao_SFC.Fechar_Arquivo.xDone THEN
25     Status_Carregamento:= 'Finalizado!';
26 END_IF
27
28

```

No passo “Ligar_Esteira”, por exemplo, o flag “Flag.Iniciando” é acionado e a variável “Status_Carregamento” é atualizada com a *string* “Iniciando”, alterando o texto de status na visualização.

Após o término do processo de contagem o status é atualizado como “Aguardando Emissão de Dados” até que o usuário pressione o botão “Emissão de Dados”, que aciona a lógica de exportação de dados em Ladder, vista na Figura 29. Quando isso acontece, a lógica exporta um arquivo no formato TXT com os dados do *Array* da carga, nomeado com o número da ordem de carregamento, conforme Figura 32.

Figura 32 - Dados da carga exportados em txt.



```
1465 - Notepad
File Edit Format View Help
Produtos:;3;3;3;4;2;3;1;1;2;4
Data:;D#2021-08-31
Destino:;Sao Paulo
Ordem de Carregamento:;1465
Tempo de Carregamento:;T#1m7s683ms
Status:;Finalizado
```

3.5 Transferência de dados usando OPC

O OPC é um protocolo de comunicação padrão desenvolvido para possibilitar a interconectividade entre dispositivos com protocolos diferentes. A grande vantagem do uso desse protocolo é sua adaptabilidade em diversos equipamentos nos dias atuais, ideal para desenvolvimento de trabalhos futuros.

Nesse trabalho foi utilizado o OPC para a transferência de dados entre o CODESYS e o Factory IO em tempo real, a fim de simular o processo sistemicamente (CODESYS simulando o controlador) e visualmente (Factory IO simulando o processo físico). Foi utilizado o CODESYS OPC Server, já incluso no CODESYS 3.5, capaz de comunicar e transferir dados do CODESYS para outros CLPs, IHMs e outros softwares, como o Factory IO.

Para a configuração foi necessário adicionar o objeto *Symbol Configuration* no CODESYS e selecionar as variáveis desejadas na comunicação, conforme Figura 33.

Figura 33 - Configuração de variáveis OPC no CODESYS.

Symbols	Access Rights	Maximal	Attribute	Type	Members	Comment
<ul style="list-style-type: none"> [-] [✓] [📄] Simulacao_SFC <ul style="list-style-type: none"> [✓] [🔗] Emissor [✓] [🔗] Esteira1 [✓] [🔗] Esteira2 [✓] [🔗] SensorAltura1 [✓] [🔗] SensorAltura2 [✓] [🔗] SensorAltura3 [✓] [🔗] SensorContagem [✓] [🔗] SensorFinal [✓] [🔗] SensorLargura1 [✓] [🔗] SensorLargura2 						
				BOOL		
				BOOL		
				BOOL		
				BOOL		
				BOOL		
				BOOL		
				BOOL		
				BOOL		
				BOOL		
				BOOL		

As mesmas variáveis foram vinculadas aos atuadores (esteira, emissor, removedor, LEDs) e sensores na janela de *Driver* no Factory IO, conforme Figura 34.

Figura 34 - Configuração de variáveis OPC no Factory IO.

SENSORS

- FACTORY I/O (Paused) [🟢]
- FACTORY I/O (Reset) [🟢]
- FACTORY I/O (Running) [🟢]
- FACTORY I/O (Time Scale) [🟦]
- Sensor Contagem [🟢]
- Sensor Final [🟢]
- SensorAltura1 [🟢]
- SensorAltura2 [🟢]
- SensorAltura3 [🟢]
- SensorLargura1 [🟢]
- SensorLargura2 [🟢]

Server:
OPCUAServer@DESKTOP-OJH
99JT (JA) (61)

- SerialNumber
- RevisionCounter
- Manufacturer
- Model
- DeviceManual
- DeviceRevision
- SoftwareRevision
- HardwareRevision
- Emissor [🟢]
- Esteira1 [🟢]
- Esteira2 [🟢]
- SensorAltura1 [🟢]
- SensorAltura2 [🟢]
- SensorAltura3 [🟢]
- SensorContagem [🟢]
- SensorFinal [🟢]
- SensorLargura1 [🟢]
- SensorLargura2 [🟢]

ACTUATORS

- Emissor [🟢]
- Esteira1 [🟢]
- Esteira2 [🟢]
- FACTORY I/O (Camera Position) [🟠]
- FACTORY I/O (Pause) [🟢]
- FACTORY I/O (Reset) [🟢]
- FACTORY I/O (Run) [🟢]
- Removedor [🟢]
- Emissor [🟢]
- Esteira1 [🟢]
- Esteira2 [🟢]

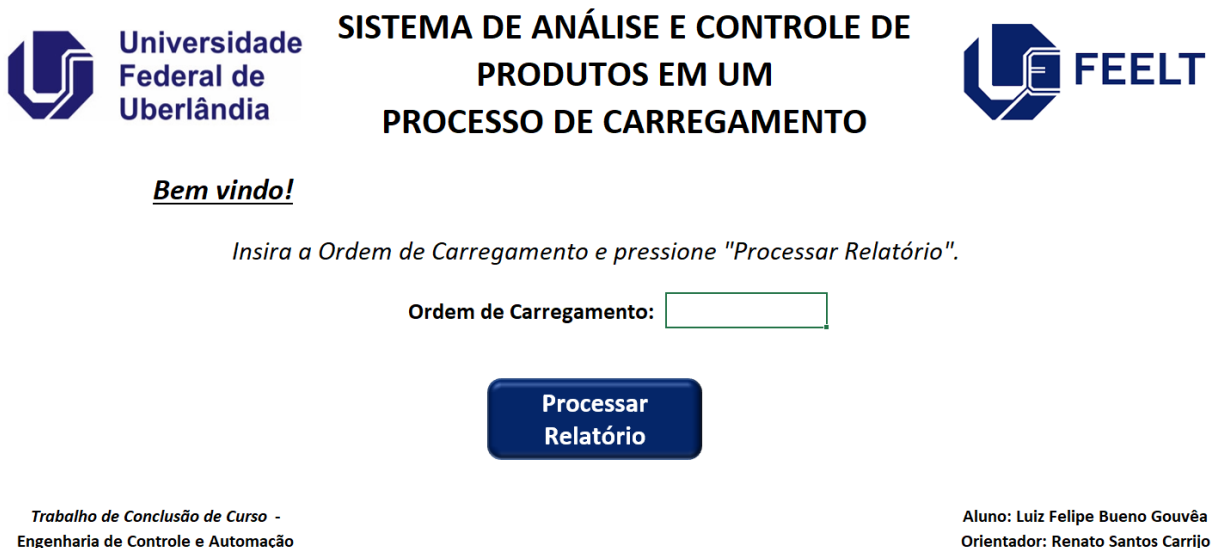
Com todas as variáveis configuradas foi iniciado o CODESYS OPC Server e feita a conexão do Factory IO e da lógica do CODESYS para iniciar a simulação.

3.6 Tratamento de dados e geração de relatório usando VBA

O Excel é uma ferramenta utilizada para criação, edição e controle de dados através de planilhas e gráficos. O Virtual Basic for Applications é uma ferramenta, hospedada dentro do Excel, capaz de realizar tarefas repetitivas de forma rápida e automática, através de uma lógica de programação textual. Nesse projeto, o VBA foi responsável pela importação, tratamento dos dados (Figura 32) e a geração de um relatório analítico.

Para isso foi desenvolvida uma janela de entrada (aba), na forma de um dashboard, onde o usuário insere a ordem de carregamento e pressiona o botão “Processar Relatório”, conforme Figura 35.

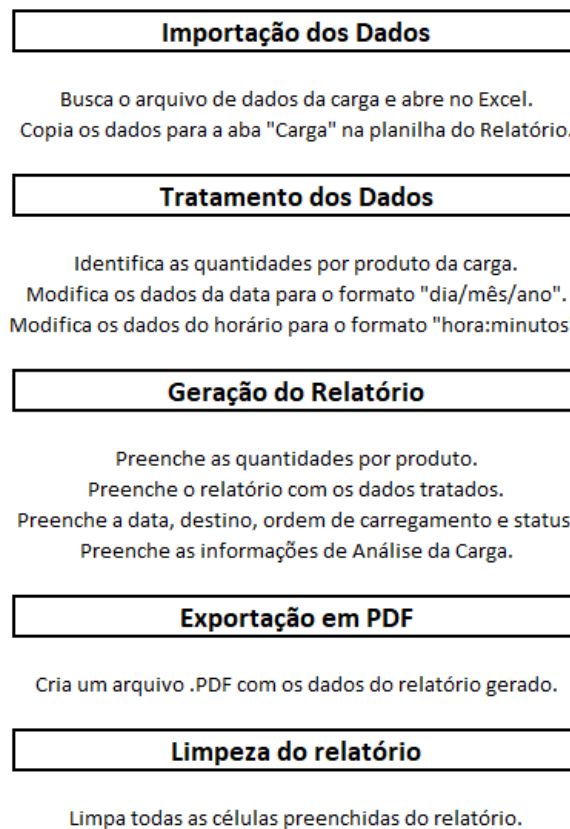
Figura 35 - Janela de entrada do Excel.



O botão “Processar Relatório” foi vinculado a uma macro, e quando pressionado, executa a lógica desenvolvida em VBA que importa, trata, faz análises e apresenta os

dados do carregamento no formato de um relatório. Para isso, a lógica foi dividida em alguns passos conforme a arquitetura mostrada na Figura 36.

Figura 36 - Arquitetura do código em VBA.



O código completo desenvolvido para importação, tratamento de dados e geração do relatório será apresentado no fim desse trabalho como APÊNDICE.

O primeiro passo desenvolvido foi a criação de uma lógica para a importação dos dados para a planilha de resultados no Excel. Nesse passo foi utilizada a função *Application.DisplayAlerts* para desabilitar alertas e mensagens durante a execução de uma macro, isso diminui o tempo de processamento da macro, pois faz com que a macro seja executada sem atualizar a tela a cada linha do código.

Para a importação dos dados foi utilizada a função *Workbooks.OpenText*, que busca e abre o arquivo TXT exportado pelo CODESYS em uma nova planilha do Excel.

Em seguida, ela é copiada, colada na aba Carga da planilha de resultados e a planilha importada é fechada.

Conforme explicado no tópico de exportação dos dados no CODESYS, o símbolo “;” separa os dados em colunas diferentes no Excel. Dessa forma os dados são importados conforme Figura 37.

Figura 37 - Dados do carregamento importados no Excel.

	A	B	C	D	E	F	G	H	I	J	K
1	Produtos:	3	3	3	4	2	3	1	1	2	4
2	Data:	D#2021-08-31									
3	Destino:	Sao Paulo									
4	Ordem de Carregamento:	1465									
5	Tempo de Carregamento:	T#1m7s683ms									
6	Status:	Finalizado									

O segundo passo foi a criação de uma lógica para o tratamento dos dados importados: a quantidade de produtos por tipo, a data e o tempo do carregamento.

Para a aquisição da quantidade de produtos por tipo foram criadas quatro variáveis, uma para cada produto, atreladas à fórmula *Countlf* (ContSe), que conta a quantidade de vezes em que o número do produto, referente a cada variável, aparece em toda a linha 1.

Para a data foi necessário remover os caracteres “D#” e modificar os dados para o formato português. As fórmulas utilizadas foram *Right* (Direita), que extrai dez caracteres da célula B2 da direita para a esquerda, e *Format* (Formatar), que converte os dados extraídos para o formato “dd/mm/yyyy” (dia, mês e ano). Em seguida, a data convertida é escrita na célula B15.

Para o horário foi necessário remover os caracteres “T#” e modificar os dados para o formato de horário digital. Nesse projeto, os dados em milissegundos “ms” foram desconsiderados. As fórmulas utilizadas foram *Substitute* (Substituir), que remove os caracteres “ms”, *Mid* (Ext.Texto), que extrai os minutos e segundos do texto, e *Find* (Procurar), que identifica em qual posição dentro do texto se encontram os caracteres “m” e “s”.



Em suma, a lógica de formatação de horário remove os caracteres “ms”, identifica em quais posições do texto se encontram os caracteres “m” e “s” e extrai os números

anteriores a eles (referentes aos minutos e segundos). Quando a simulação ocorre em menos de 1 minuto, a fórmula não encontra o caractere “m” e retorna um erro. Para contornar esse problema, foi utilizada a fórmula *IfError* (SeErro), que executa uma segunda lógica e extrai apenas o número de segundos.

Os dados dos minutos e segundos extraídos são formatados utilizando a fórmula *Time* (Tempo) e inseridos na célula B19.


Após o tratamento dos dados, foi necessário desenvolver uma lógica para preenchimento do relatório final do carregamento. Para isso foi desenvolvido um modelo padrão de relatório, modificado de acordo com os dados do carregamento, conforme Figura 38.

Figura 38 - Modelo padrão de relatório desenvolvido no Excel.


Dados do Carregamento

Data




16


Ordem de Carregamento



Destino




Status





Sequência da Carga:


Posição	#ID	Descrição
1º		
2º		
3º		
4º		
5º		
6º		
7º		
8º		
9º		
10º		


Nº de Caixas por Produto

 **Impressora:**

 **Quebra-Cabeça:**

 **Video Game:**


 **Smartphone:**




Mapa da Carreta - Posição Aproximada de Produtos

Análise do Carregamento

Quantidade Carregada




Tempo de Carregamento




Peso da Carga

0 Kg



Fillrate (m³)

0,00%



As informações de Data, Ordem de Carregamento, Destino, Status e Sequência da Carga foram vinculados às células na aba Carga, portanto assim que os dados são importados e tratados, o relatório é atualizado automaticamente.

O preenchimento do Número de Caixas por produto concatena a quantidade de cada produto, calculada anteriormente, com o texto “Caixa” ou “Caixas”. Para isso, foi utilizada a fórmula *Concat* (Concatenar), que concatena o número de caixas de cada

produto com o resultado da fórmula *VLookUp* (ProcV), que retorna o texto de uma planilha auxiliar, de acordo com a quantidade de caixas, conforme Figura 39.

Figura 39 - Tabela de flexão de texto.

	A	B
1	Quantidade	Flexionador
2	0	
3	1	Caixa
4	2	Caixas

A fórmula retorna um texto vazio quando o número de caixas é zero, “Caixa” quando o número de caixas é 1 e “Caixas” quando o número de caixas é maior ou igual a 2.

As informações de Quantidade Carregada, Tempo de Carregamento, Peso da Carga e Fillrate apresentadas na Análise do Carregamento estão vinculadas a células nas abas Carga e Auxiliar e são atualizadas automaticamente após o tratamento dos dados. A Quantidade Carregada é calculada a partir da soma das quantidades de cada produto.

Durante a simulação, o tempo de carregamento foi medido em minutos e segundos, porém afim de aproximar o resultado de um número real, o Tempo de Carregamento apresentado no relatório final foi convertido para horas e minutos (minutos considerados como horas e os segundos como minutos).

O Peso da Carga e Fillrate foram calculadas na aba Auxiliar, conforme Figura 40.

Figura 40 - Cálculo de Peso e Fillrate da Carga.




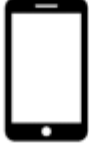
6	Produtos	Peso (Produto Kg)	Dimensao (Produto m3)
8	1	15	0,125
9	2	10	0,09375
10	3	8	0,0625
11	4	3	0,03125
12	Total	-	-
13	Maximo (Carreta)	120	1
14			
15	Carga		
16	Fillrate	0,00%	
17	Peso	0 Kg	

Produtos	Peso (Carga Kg)	Dimensao (Carga m3)
1	0	0
2	0	0
3	0	0
4	0	0
Total	0	0
Porcentagem (Carreta)	0,00	0

Foram utilizadas as medidas de peso e dimensão das caixas fornecidas pelo Factory IO. A fim de simular um processo real, para o cálculo do peso e *fillrate* foram utilizadas as medidas de peso máximo da carga igual a 41,5 toneladas e *fillrate* máximo de 10 caixas do Produto 1 (maior caixa). Diante disso, a planilha calcula automaticamente os dois valores, de acordo com a quantidade de cada tipo de produto simulado na carga.

Finalmente, para o preenchimento do mapa da carreta, foi utilizada uma lógica utilizando a ferramenta Câmera do Excel. Para isso, foi criada uma aba (Produtos) com cada uma das imagens dos produtos, conforme Figura 41.

Figura 41 - Imagens dos produtos utilizados no Mapa da Carreta.

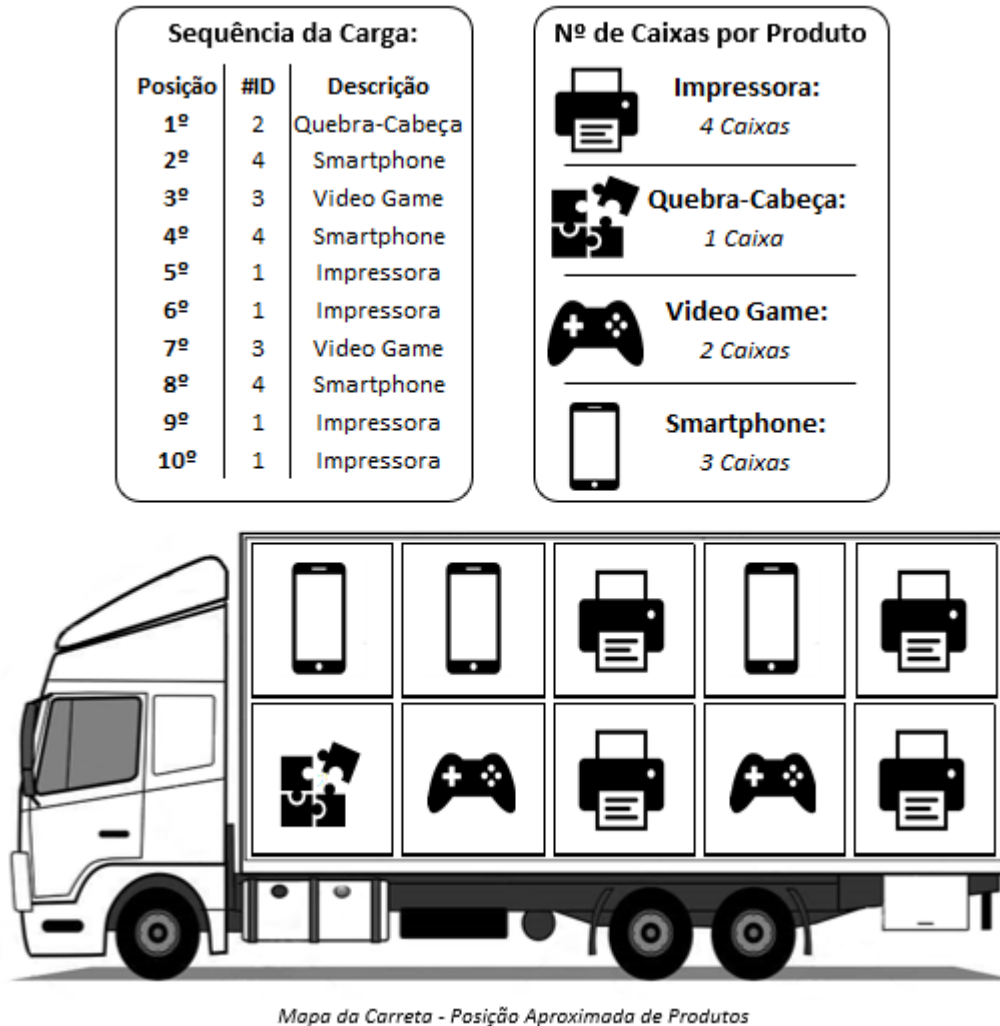
	A	B
1		
2		
3		
4		
5		

Utilizando a ferramenta Câmera, foram feitas 10 cópias da célula A1 (quadrados com a imagem do produto) e inseridas no Mapa da Carreta, representando a ordem de entrada dos produtos, do primeiro ao último, da esquerda para a direita respectivamente.

De acordo com o produto em cada posição, o Excel preenche automaticamente cada quadrado com a respectiva imagem utilizando a fórmula Indireto.

Em suma, de acordo com o tipo produto o Excel importa a respectiva imagem da aba Produtos, para cada posição de entrada na carreta, conforme exemplo na Figura 42.

Figura 42 - Exemplo de preenchimento do Mapa da Carreta.



Conforme exemplo acima, de acordo com a Sequência da Carga, os primeiros produtos a entrarem na carreta (passarem pela esteira) foram os Produtos 2 e 4 (Quebra-Cabeça e Smartphone), portanto os dois se encontram na parte esquerda no mapa da carreta, seguidos do Video Game e Smartphone, e assim por diante.

Ainda foram desenvolvidas outras duas lógicas para exportação do relatório final em PDF automaticamente e para limpeza do relatório.

Para a exportação do relatório em PDF foi utilizada a função *ExportAsFixedFormat*, na qual o nome do arquivo exportado foi definido como a ordem de carregamento inserida pelo usuário, conforme Figura 43.

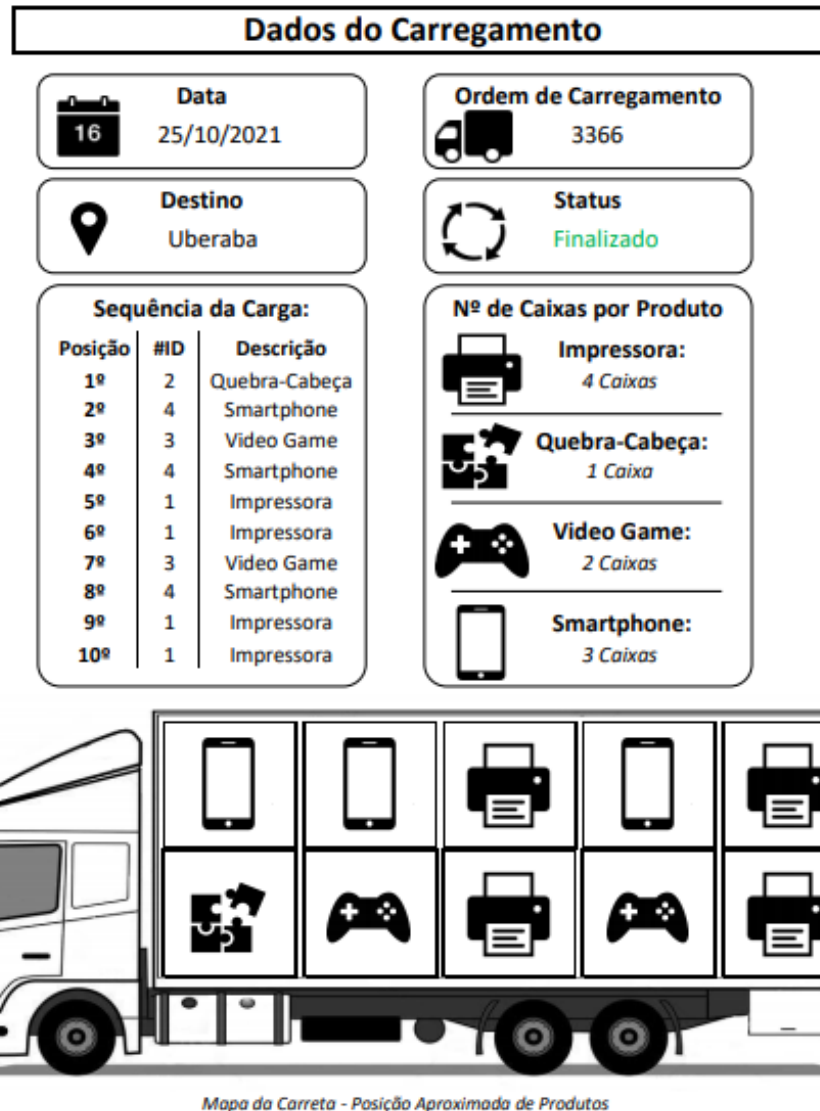
Figura 43 - Lógica de exportação do relatório em PDF utilizando VBA.

```
Sub Exportar_PDF()  
,  
'Desativa a atualizacao visual da tela entre comandos  
Application.ScreenUpdating = False  
  
'Exporta o Relatorio como PDF  
ActiveSheet.ExportAsFixedFormat Type:=xlTypePDF, Filename:= _  
"D:\Dropbox\Faculdade\TCC\Simulacao\Controle Carregamento Final\Cargas\" & _  
Range("Dashboard!$I$12").Value & ".pdf", IgnorePrintAreas:=False  
  
MsgBox ("Relatório Exportado com Sucesso!")  
End Sub
```

Quando a exportação é concluída com sucesso, o Excel apresenta uma caixa de diálogo (MsgBox) com a mensagem “Relatório Exportado com Sucesso!”. Pensando em uma aplicação real do projeto, foram inseridas data e hora da emissão do arquivo, dados importantes para controles logísticos, no canto superior direito do arquivo PDF gerado, conforme Figura 44.

Figura 44 - Exemplo de relatório exportado em PDF.

Emitido em:
26/08/2021 - 15:34



Para a limpeza dados do relatório foi desenvolvida uma lógica que apaga o conteúdo das células preenchidas durante a importação e tratamento dos dados, conforme Figura 45.

Figura 45 - Lógica de limpeza dos dados em VBA.

```
Sub Limpar_Relatorio()  
  
'Desativa a atualizacao visual da tela entre comandos  
Application.ScreenUpdating = False  
  
'Aba Carga  
Sheets("Carga").Visible = True  
Sheets("Carga").Select  
Cells.Select  
Selection.ClearContents  
Range("A1").Select  
Sheets("Carga").Visible = False  
  
'Aba Relatorio  
Sheets("Relatorio").Select  
Range("G16").Select  
Selection.ClearContents  
Range("G19").Select  
Selection.ClearContents  
Range("G22").Select  
Selection.ClearContents  
Range("G25").Select  
Selection.ClearContents  
Range("B4").Select  
  
'Aba Dashboard  
Sheets("Dashboard").Select  
Range("I12").Select  
Selection.ClearContents  
Range("I12").Select  
  
End Sub
```

Após o fim da execução da lógica de limpeza dos dados, o usuário é levado de volta à janela de entrada da planilha (Figura 35) para que uma nova ordem de carregamento seja inserida.

4. RESULTADOS

Após a separação, coleta dos produtos no porta-pallet, e a conferência, comparação e validação das quantidades de produtos separados com o pedido (lista de separação da carga), se inicia o processo de carregamento em que o operador faz o trânsito da carga separada até o interior da carreta, desmonta o pallet e organiza as caixas individualmente.

Diante de um processo manual, os objetivos do trabalho eram desenvolver um sistema capaz simular a automatização, o controle e monitoramento do carregamento em tempo real e, após o término do processo, fazer uma análise da carga e gerar um relatório automático com os dados daquele carregamento. Para isso o trabalho foi dividido em pequenas etapas: virtualização do processo físico utilizando o Factory IO V2.4.3, simulação do controle e monitoramento do processo utilizando o CODESYS V3.5, transferência de dados utilizando OPC, tratamento de dados e exportação do relatório final utilizando VBA aplicado no Microsoft Excel 2019. Todas as etapas foram desenvolvidas ao longo de um período de 2 meses e uma semana em que foram feitas pequenas modificações e melhorias na programação.

4.1 Resultados da simulação

Durante um processo de carregamento manual o operador deve transitar com os pallets da doca de carregamento até o interior da carreta, onde ele é desmontado e as caixas são organizadas individualmente uma sobre a outra. O uso do Factory IO teve como objetivo simular a proposta de automatização desse processo, em que o operador desmonta o pallet e alimenta uma esteira, que leva as caixas até o interior da carreta, reduzindo o tempo e o esforço físico gastos pelo operador, e os prejuízos com a queda de produtos durante o trânsito dos pallets.

A simulação no software Factory IO foi capaz que demonstrar a automatização do processo proposta com equipamentos comuns no meio industrial, conforme Figura 46.

Figura 46 - Simulação do processo em andamento no Factory IO.



Em suma, o processo se inicia com o emissor, que simula a atividade do operador de pegar a caixa no pallet e colocar na esteira, por se tratar de um processo manual, foi configurado um tempo variável entre 1 e 6 segundos entre as emissões no Factory IO, a fim de aproximar a simulação de valores reais. O emissor também foi configurado para emitir um total de dez caixas, variando entre quatro caixas de tamanhos diferentes, de forma aleatória, a fim de simular cargas diversas. Em seguida a esteira transporta as caixas passando por sensores, que acionam com a presença de qualquer objeto sólido dentro de uma determinada distância, simulando um contador de caixas identificadas pelo tamanho. Finalmente as caixas são removidas ao atingirem o fim da esteira, simulando a retirada da caixa da esteira pelo operador no interior da carreta. O Factory IO foi capaz de virtualizar a automatização do processo proposto e se mostrou uma ótima ferramenta de simulação de processos de automação em 3D.

O CODESYS fez uso da linguagem SFC para a divisão e ordenação do controle em etapas maiores: acionamento dos dispositivos (esteira e emissor), contagem, desligamento dos dispositivos e exportação dos dados. As etapas de acionamento e desligamento foram controladas em SFC. Dentro da etapa de contagem foi utilizada a linguagem Ladder para identificação das caixas pelo tamanho utilizando uma lógica de

associação de sensores, a fim de simular um contador com capacidade de diferenciação das caixas. Nessa mesma etapa foi utilizada a linguagem ST para o preenchimento de um *Array* com os dados simulados. Dentro da etapa de exportação dos dados foi utilizada a biblioteca File Access, com blocos em linguagem Ladder para exportar os dados do *Array* para um arquivo TXT.

O CODESYS foi capaz de simular o controle de um CLP, utilizando três linguagens diferentes (SFC, Ladder e ST), e simular uma interface de controle e monitoramento, semelhante às IHMs amplamente utilizadas na indústria, conforme Figura 47.

Figura 47 - Interface de controle e monitoramento do processo em andamento.

Bem Vindo!

Insira os dados do carregamento e pressione Iniciar.

Iniciar

Data

← August 2021 →

Mo	Tu	We	Th	Fr	Sa	Su
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Ordem de Carregamento

150921

Destino

Uberlandia

Monitoramento

STATUS DO CARREGAMENTO

Em andamento

Esteira **Contador**

Impressora: 2
Quebra-Cabeça: 1
Video Game: 1
Smartphone: 2
Total: 6

Emitir Dados

A interface de controle coletou informações da carga (data, ordem de carregamento e destino), inseridas pelo operador antes do início do carregamento, através de caixas de texto e caixas de seleção. A interface de monitoramento apresentou o andamento processo, atualizando automaticamente o acionamento dos dispositivos, o status do carregamento e as quantidades de cada produto em tempo real, a partir da simulação no Factory IO.

O CODESYS mostrou as vantagens do uso da automação permitindo o controle e monitoramento em tempo real do processo. Durante um processo manual de carregamento, por exemplo, pode ser difícil identificar quais produtos já foram embarcados na carreta e em quais quantidades sem uma nova conferência da carga. O uso de um sistema de monitoramento como o proposto, permite a visualização em tempo real do estado do processo e as quantidades de cada produto embarcado. Diante disso, é possível comparar a quantidade solicitada no pedido (lista de separação) e a quantidade embarcada durante o ato do carregamento, possibilitando a parada do processo e correção de quantidades e/ou produtos divergentes.

A utilização do OPC UA possibilitou a comunicação entre o CODESYS e o Factory IO, permitindo o controle e monitoramento do processo em tempo real na tela de supervisor. A comunicação se mostrou eficiente, pois não houve perda de dados e o tempo de resposta foi ideal para o monitoramento do processo desejado. Além disso, a comunicação OPC está inserida na Indústria 4.0, em que as informações estão cada vez mais integradas e mais facilmente acessíveis, como proposto nesse monitoramento, e é amplamente utilizada na transferência de dados entre dispositivos com drivers diferentes, ideal para aplicações em processos reais.

4.2 Resultados do tratamento de dados e exportação de relatório

Para a apresentação dos resultados no Excel foi desenvolvida uma interface de interação com o usuário, característica de sistemas automatizados, em que ele insere o número do carregamento simulado e pressiona um botão para executar a lógica de importar, tratar os dados e apresentar o relatório automaticamente, conforme Figura 48.

Figura 48 - Interface de interação com o usuário.

 **Universidade
Federal de
Uberlândia**

**SISTEMA DE ANÁLISE E CONTROLE DE
PRODUTOS EM UM
PROCESSO DE CARREGAMENTO**

 **FEELT**

Bem vindo!

Insira a Ordem de Carregamento e pressione "Processar Relatório".

Ordem de Carregamento:

**Processar
Relatório**

Trabalho de Conclusão de Curso -
Engenharia de Controle e Automação

Aluno: Luiz Felipe Bueno Gouvêa
Orientador: Renato Santos Carrijo

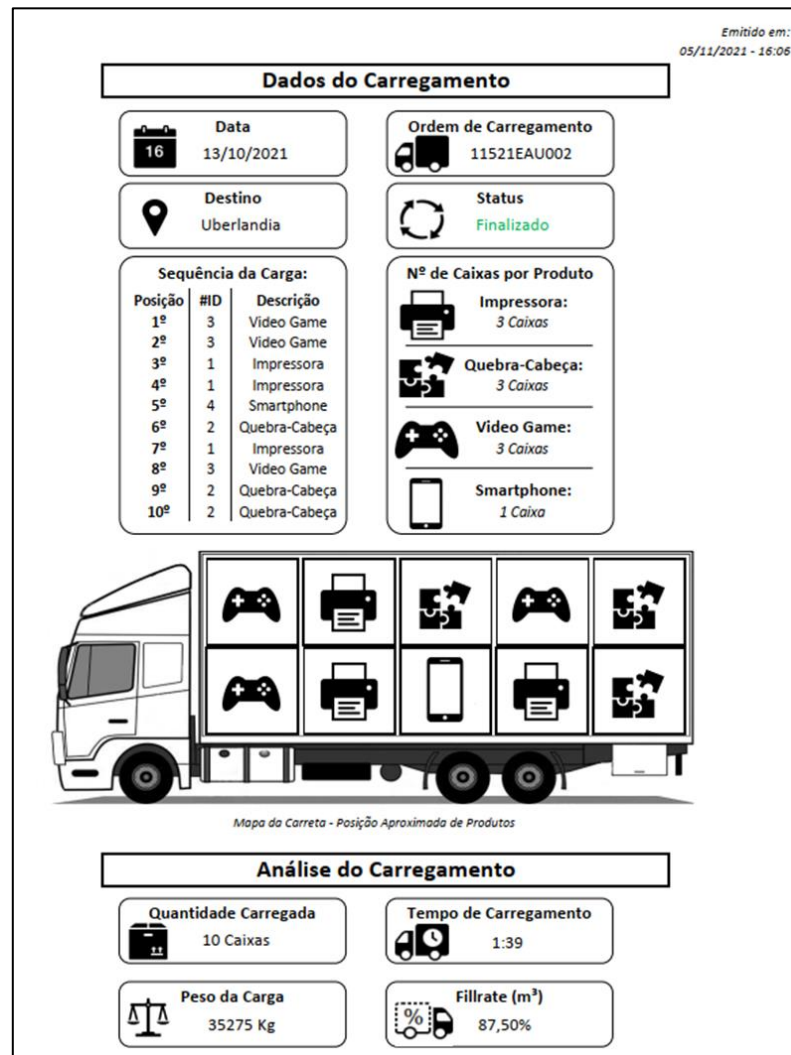
A lógica desenvolvida atingiu os objetivos esperados, apresentando todas as informações necessárias de forma fácil, o usuário só precisa inserir a ordem de carregamento e clicar para processar, organizada, todos os dados da carga se encontram separados e identificados no relatório, e rápida, o preenchimento de todas as informações ocorre em poucos segundos, tornando suscetível sua aplicação em um processo real. Diante disso, espera-se reduzir o número de erros com preenchimento de informações, facilitar e agilizar a busca por informações da carga.

A análise dos dados do carregamento fez uso de informações coletadas na simulação para estimar os valores de peso, *fillrate* e tempo de carregamento próximos dos reais. Com a configuração de um período variável de emissões e a emissão de caixas aleatórias foi possível simular cargas diferentes. Na análise da carga o tempo entre as emissões influencia no tempo total do carregamento, pois o tempo de carregamento diz respeito ao tempo entre a emissão da primeira caixa até a última caixa ser removida, e a aleatoriedade das caixas emitidas influencia no cálculo do *fillrate* e do peso da carga, pois para cada produto foram determinados dimensões e peso específicos. Diante disso, foi possível simular diversas cargas com análises distintas.

Pensando em uma aplicação real do projeto, foram adicionados botões para facilitar a exportação do relatório em PDF automaticamente e para limpeza e importação de novos dados. O relatório gerado também conta com a data e horário de sua emissão, informações importantes em controles logísticos.

O relatório atingiu os objetivos específicos esperados, fornecendo os dados da carga, as quantidades por produto, um mapa da carreta e uma análise de peso, fillrate e tempo de carregamento. Essas informações podem ser utilizadas na análise de KPIs, na criação de um histórico de carregamentos, na mitigação de erros de embarque e até mesmo auxiliar na descarga da carreta, já que o mapa mostra a posição aproximada de cada produto, conforme Figura 49.

Figura 49 - Relatório final do processo.



Os dados de sequência da carga e número de caixas por produto permitem uma nova conferência da carga ao término carregamento, pois o operador é capaz de comparar as quantidades planejadas, conferidas antes do carregamento, com as quantidades carregadas, identificadas pelo contador e impressas no relatório. O uso do relatório também pode ser importante e facilitar a identificação de excesso de peso na carga.

Durante o desenvolvimento do projeto foram enfrentados alguns problemas mencionados a seguir. O Factory IO demonstrou ser uma excelente ferramenta de simulação de processos de automação, porém houve dificuldade na identificação de tipos diferentes de produtos. O software possui poucos sensores com configurações limitadas, portanto foi necessário criar uma lógica utilizando 5 sensores fotoelétricos dispostos a fim de diferenciar as caixas pelo tamanho, conforme Figura 19. Diante desse mesmo problema, também foi necessário criar uma estratégia de identificação no CODESYS, de acordo com o acionamento de cada sensor, utilizando operações lógicas E, OU e XOR, conforme Figura 25.

Outro problema enfrentado foi a separação e organização dos dados importados no Excel, pois todos os dados eram importados juntos dentro de uma única célula. Para contornar o problema, foi desenvolvida uma lógica (Figura 28) para preenchimento do *Array* intercalando dados com os símbolos “;” ou “\$N”, utilizados para saltar coluna e saltar linha respectivamente. Apesar dos problemas enfrentados, o projeto foi capaz de simular um cenário de carregamento com dados próximos aos vistos em processos logísticos reais.

Além de melhores resultados com sua aplicação em campo mencionados, o projeto também busca trazer ganhos para o meio acadêmico, já que a integração das ferramentas utilizadas permite a criação de sistemas de automação de forma virtual, permitindo simulações complexas que normalmente são feitos em laboratórios e requerem equipamentos industriais.

5. CONCLUSÃO

Esse trabalho propôs a criação de um sistema de controle e monitoramento de carregamento, para isso foram desenvolvidas simulações de um processo real em que uma esteira transporta produtos, passando por um contador. Após o término do carregamento, as informações coletadas pelo controlador são exportadas e apresentadas na forma de um relatório.

A simulação do processo real foi feita no software Factory IO, a simulação do controle do processo foi feita no software CODESYS, o protocolo OPC permitiu a comunicação entre os dois softwares em tempo real, o tratamento de dados e desenvolvimento do relatório foi feito no Excel utilizando programação em VBA.

Os resultados das simulações se mostraram satisfatórios, não houveram falhas na identificação dos produtos na esteira, a comunicação OPC permitiu a transferência de dados entre o CODESYS e o Factory IO em tempo real e os dois softwares foram capazes de mostrar separadamente as etapas do processo:

- Físico: funcionamento e integração de equipamentos industriais como esteiras e sensores.
- Controle: funcionamento de uma lógica de controle utilizada em CLPs.
- Monitoramento: visualização de uma janela de interface com o usuário, semelhante a uma IHM.

Com a implantação desse sistema em um processo real, espera-se obter aumento da produtividade, reduzindo o tempo e esforço gastos pelo operador com o deslocamento de pallets até o interior da carreta, e melhor controle, utilizando o sistema supervisor é possível monitorar o processo em tempo real, fazer estimativas de tempo até o fim do carregamento, identificar os produtos já carregados e suas quantidades, entre outros.

Os resultados do relatório foram apresentados de forma organizada e em poucos segundos, necessitando apenas que o usuário insira a ordem de carregamento e pressione um botão. Com a implantação desse sistema em um processo real, espera-se obter assertividade, pois a precisão de contagem e apresentação das quantidades por produto permitem a conferência de produtos planejados e os produtos carregados, e melhorias de gestão e planejamento, o uso de indicadores como o tempo de

carregamento e o fillrate são de extrema importância no setor logístico e impactam nos resultados do processo.

Esse trabalho fez uso de diferentes disciplinas estudadas durante a graduação como CLP, foram utilizadas três linguagens diferentes (LD, ST e SFC), eletrônica digital, foram utilizadas operações lógicas E, OU, XOR, sistemas supervisórios, foi desenvolvida uma janela de controle e monitoramento, instrumentação industrial, foram utilizados sensores e atuadores na simulação do processo físico, redes industriais, foi utilizado o protocolo de comunicação OPC e lógicas de programação. Diante disso, conclui-se que o projeto abre portas para aplicação em atividades acadêmicas, no desenvolvimento de simulações de processos de automação. Também espera-se incentivar a criação de uma disciplina de graduação que explore a integração de sistemas de simulação de processos como o proposto, possibilitando o controle de dispositivos industriais inexistentes em laboratórios acadêmicos.

Em suma, o trabalho cumpriu com os objetivos específicos planejados e apresentou resultados rápidos e precisos:

- ***Desenvolver uma proposta de arquitetura para aplicação do projeto.***

A arquitetura foi desenvolvida, conforme Figura 16.

- ***Criar uma lógica no software CODESYS para simular a passagem de caixas em um contador.***

A lógica de controle foi desenvolvida utilizando três linguagens: Ladder, SFC e ST.

- ***Desenvolver uma visualização virtualizada do processo no software Factory IO.***

A simulação do processo físico apresentou virtualmente uma esteira, que transporta caixas passando por um contador.

- ***Estabelecer uma comunicação utilizando protocolo OPC entre o CODESYS e o software de simulação gráfica Factory IO.***

O protocolo OPC foi utilizado para realizar a comunicação e transferência de dados em tempo real entre o CODESYS e o Factory IO, durante a simulação.

- ***Desenvolver um sistema de controle e monitoramento no CODESYS com acionamento e inserção de dados.***

Foi desenvolvida uma interface de controle e monitoramento utilizando a ferramenta de visualização do CODESYS em que o usuário é capaz inserir os dados de data, ordem de carregamento, e destino, visualizar o status do processo e as quantidades já carregadas.

- ***Desenvolver uma estratégia para exportação dos dados simulados e importação dos mesmos no Excel.***

Foi desenvolvida uma lógica em ST para o preenchimento de um array e uma lógica em Ladder, utilizando a biblioteca *FileAccess*, para a exportação dos dados do array para um arquivo .TXT.

- ***Desenvolver um código em VBA para tratar os dados e apresentá-los na forma de um relatório contendo: Data, Destino, Número do Carregamento, Lista de Produtos, Quantidade de Produtos, Tempo de Carregamento, Mapa da Carreta, Peso da Carga, Fillrate da Carreta.***

Foi desenvolvida uma lógica em VBA de importação, tratamento e exportação de dados em formato de relatório com todas as informações citadas acima.

Além disso, foram criadas outras duas lógicas adicionais, uma para exportar o relatório gerado em PDF automaticamente, outra para limpar o relatório e retornar à janela inicial.

Diante dos resultados desse trabalho, espera-se abrir portas para o desenvolvimento de projetos e sistemas de automação no setor logístico, que ainda possui poucos estudos e aplicações de automação comparados ao meio industrial, além prosseguir com implementações e criação de novas análises a partir das apresentadas nesse projeto.

Como trabalho futuro espera-se:

- Ampliar a aplicação do relatório à uma situação de descarga, possibilitando que o sistema valide as informações de quantidade e posição na carreta presentes no relatório de carga, e ainda gere um novo relatório com os dados da carga descarregada. Também será possível desenvolver uma lógica de comparação entre carga carregada e carga descarregada.
- A partir das informações geradas na análise da carga, espera-se desenvolver uma análise de indicadores a partir do histórico de cargas.
- Expandir a simulação para um número variável de caixas, sendo também necessário adaptar o mapa da carreta para um modelo duas dimensões.
- Expandir a aplicação do projeto a outros meios de transporte, criando análises específicas e mapeamento de cargas em aeronaves e navios.

6. BIBLIOGRAFIA

ARAÚJO, P. **AUTOMAÇÃO COMERCIAL: O QUE É E BENEFÍCIOS**. Clientar CRM, (s.d). Disponível em: <https://www.clientarcrm.com.br/automacao-comercial-erp-crm/>. Acesso em: agosto de 2021.

ASKARI, H. **Visual Basic for Applications**. MINE 325 – Mine Planning and Design. Disponível em: https://sites.ualberta.ca/~jbb/files/Chapter01_VBA.pdf. Acesso em: agosto de 2021.

DEVASIA, A. (01 de outubro de 2020). **An Overview of Sequential Function Chart (SFC) Programming**. Control Automation. Disponível em: <https://control.com/technical-articles/an-overview-of-sequential-function-chart-sfc-programming/>. Acesso em: 2021.

DREUX, M; AZEVEDO, F. U. B. **Macros para Excel na prática**. 4ª reimpressão. Rio de Janeiro: Elsevier, 2009.

Factory IO. **About Factory IO**. Factory IO Docs. Disponível em: <https://docs.factoryio.com/>. Acesso em: agosto de 2021.

FILHO, A. P. et al. **SIMULAÇÃO: DEFINIÇÃO**. Ribeirão Preto: Medicina, junho de 2007.

FONSECA, M. **Comunicação OPC - Uma abordagem prática**. VI Seminário de Automação de Processos, Associação Brasileira de Metalurgia e Materiais. Vitória: outubro de 2002.

GARRETT, F. (15 de maio de 2018). **O que é e como usar VBA no Excel?** TECHTUDO. Disponível em: <https://www.techtudo.com.br/dicas-e-tutoriais/noticia/2016/03/o-que-e-e-como-usar-vba-no-excel.html>. Acesso em: 2021.

ISHIZAKI, A. M. **PROJETO DE AUTOMAÇÃO COMERCIAL PARA RESTAURANTE DE PEQUENO PORTE**. Dezembro de 2009. 33p. (Engenharia Elétrica) - Universidade São Francisco, Campinas, 2009.

JULIO. (12 de abril de 2019). **Qual a diferença entre carga batida e carga paletizada?** Haas Madeiras. Disponível em: <https://www.haasmadeiras.com.br/carga-batida-e-carga-paletizada/>. Acesso em: agosto de 2021.

MASLARIC, M. et al. **Logistics Response to the Industry 4.0: the Pshysical Internet**. De Gruyter Open: Open Engineering, 2016.

MICROSOFT. (14 de agosto de 2019). **Introdução ao VBA no Office**. Microsoft Docs. Disponível em: <https://docs.microsoft.com/pt-br/office/vba/library-reference/concepts/getting-started-with-vba-in-office>. Acesso em: agosto de 2021.

MORAIS, B. C. **Sistema Integrado de Análise e Predição de Indicadores de Desempenho em um Processo Industrial**. Dezembro de 2019. 91p. (Engenharia Elétrica) – Universidade Federal de Uberlândia, Uberlândia, 2019.

Peter. **Structured Text Programming: A Step by Step Guide (With Examples)**. PLC Academy. Disponível em: <https://www.plcacademy.com/structured-text-tutorial/>. Acesso em: agosto de 2021.

SHENDE, A. et al. **Truck Load and Monitoring Analysis for Vehicle Routing System**. IRE Journals. Vol 1, 9, 266-270, março de 2018.

SILVEIRA, C. B. **A Função dos Contadores na Lógica LADDER**. Citisystems. Disponível em: <https://www.citisystems.com.br/contadores-logica-ladder-clp/>. Acesso em: 22 de agosto de 2021.

SILVEIRA, C. C. **APLICAÇÃO DE CONCEITOS DA INDÚSTRIA 4.0 EM SERVIÇOS: UM ESTUDO DE CASO DO SETOR FINANCEIRO**. 2018. 60p. (Engenharia de Produção) – Universidade Federal Fluminense, Niterói, 2018.

TAMBURO, F. S. et al. **AUTOMAÇÃO E SUPERVISÃO DE PLANTA DIDÁTICA VIRTUAL**. 2018. 266p. (Engenharia de Controle e Automação) - Universidade Tecnológica Federal do Paraná, Curitiba, 2018.

TECH TERMS. **Array Definition**. Tech Terms. Disponível em: <https://techterms.com/definition/array>. Acesso em: novembro de 2021.

ZANCAN, M. D. **Controladores Programáveis**. 3ª Edição. Santa Maria: Universidade Federal de Santa Maria: Colégio Técnico Industrial de Santa Maria, 2010.

7. APÊNDICE

O código abaixo foi desenvolvido em VBA aplicado no Microsoft Excel, utilizado para importar, tratar e preencher o relatório da carga automaticamente.

```
Sub Gerar_Relatorio()
```

```
'Definicao de Variaveis
```

```
Dim Produto1 As Integer
```

```
Dim Produto2 As Integer
```

```
Dim Produto3 As Integer
```

```
Dim Produto4 As Integer
```

```
Dim Flex As String 'Flexionador da palavra Caixa
```

```
Dim Data As String
```

```
'Desativa a atualizacao visual da tela entre comandos
```

```
Application.ScreenUpdating = False
```

```
'Desativa os alertas
```

```
Application.DisplayAlerts = False
```

```
'Importa arquivo.Txt
```

```
Workbooks.OpenText Filename:="D:\Dropbox\Faculdade\TCC\Simulacao\Controle  
Carregamento Final\Cargas\" & Range("Dashboard!$I$12").Value & ".txt", _
```

```
Origin:=437, StartRow:=1, DataType:=xlDelimited, TextQualifier:= _
```

```
xlDoubleQuote, ConsecutiveDelimiter:=False, Tab:=True, Semicolon:=True, _
```



```
Comma:=False, Space:=False, Other:=False, FieldInfo:=Array(Array(1, 1), _  
Array(2, 1), Array(3, 1), Array(4, 1), Array(5, 1), Array(6, 1), Array(7, 1), Array(8, 1), _  
Array(9, 1), Array(10, 1), Array(11, 1)), TrailingMinusNumbers:=True  
'Copia os dados do arquivo para a aba "Carga"  
Range("A1:Z40").Select  
Selection.Copy  
Windows("Processar Relatorio.xlsm").Activate  
Worksheets("Carga").Activate  
Range("A1").Select  
Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks:=False,  
Transpose:=False  
Application.DisplayAlerts = False  
Workbooks(Range("Dashboard!$I$12").Value & ".txt").Close SaveChanges:=False  
  
'Quantidade por Produto  
With Sheets("Carga")  
Produto1 = Application.CountIf(.Range("1:1"), "1")  
Produto2 = Application.CountIf(.Range("1:1"), "2")  
Produto3 = Application.CountIf(.Range("1:1"), "3")  
Produto4 = Application.CountIf(.Range("1:1"), "4")  
End With  
  
'Formatacao de Data  
Worksheets("Carga").Activate
```

```
Data = Format(Right(Range("B2"), 10), "dd/mm/yyyy")
```

```
Range("A13").Value = "Dados Formatados"
```

```
Range("A15").Value = "Data"
```

```
Range("B15").Value = Data
```

```
'Formatacao de Horario
```

```
Range("A18").Value = "Tempo de Carregamento"
```

```
Range("C5").Value = "=SUBSTITUTE(B5,""ms"", """)"
```

```
Range("B18").Value = "=MID(C5,FIND("#",C5,1)+1,IFERROR(FIND("m",C5,1),3)-  
FIND("#",C5,1)-1)"
```

```
Range("C18").Value = "=MID(C5,IFERROR(FIND("m",C5,1)+1,3),FIND("s",C5,1)-  
IFERROR(FIND("m",C5,1)+1,3))"
```

```
Range("B17").Value = "Minutos"
```

```
Range("C17").Value = "Segundos"
```

```
Range("A19").Value = "Conversao"
```

```
Range("B19").Value = "=TIME(IF(B18=""",0,B18),C18,)"
```

```
'Preenchimento do Relatorio
```

```
'Quantidade por Produto
```

```
Worksheets("Relatorio").Activate
```

```
Range("G16").Select
```

```
Flex = Application.WorksheetFunction.VLookup(Produto1, Range("Auxiliar!$A$1:$B$4"), 2, 1)
```

```
ActiveCell.Value = Produto1 & Flex
```

```
Range("G19").Select
```

```
Flex = Application.WorksheetFunction.VLookup(Produto2, Range("Auxiliar!$A$1:$B$4"), 2, 1)
```

```
ActiveCell.Value = Produto2 & Flex
```

```
Range("G22").Select
```

```
Flex = Application.WorksheetFunction.VLookup(Produto3, Range("Auxiliar!$A$1:$B$4"), 2, 1)
```

```
ActiveCell.Value = Produto3 & Flex
```

```
Range("G25").Select
```

```
Flex = Application.WorksheetFunction.VLookup(Produto4, Range("Auxiliar!$A$1:$B$4"), 2, 1)
```

```
ActiveCell.Value = Produto4 & Flex
```

```
'Analise
```

```
Produtos_Total = WorksheetFunction.Sum(Produto1, Produto2, Produto3, Produto4)
```

```
Range("Carga!$A$14").Value = "Total de Produtos"
```

```
Range("Carga!$B$14").Value = Produtos_Total & " Caixas"
```

```
Range("B4").Select
```

```
End Sub
```