

UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

Trabalho de Conclusão de Curso

Laboratório Virtual de Sistemas de Automação Industrial

Evellyne Alves de Oliveira

Campina Grande - PB

Maio de 2021

Evellyne Alves de Oliveira

Laboratório Virtual de Sistemas de Automação Industrial

Trabalho de Conclusão de Curso submetido à Coordenação de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para a obtenção do grau de Engenharia Eletricista.

Área de Concentração: Controle e Automação
Universidade Federal de Campina Grande - UFCG
Centro de Engenharia Elétrica e Informática - CEEI
Departamento de Engenharia Elétrica - DEE
Coordenação de Graduação em Engenharia Elétrica - CGEE

George Acioli Júnior, D.Sc.
(Orientador)

Campina Grande - PB
Maio de 2021

Evellyne Alves de Oliveira

Laboratório Virtual de Sistemas de Automação Industrial

Trabalho de Conclusão de Curso submetido à Coordenação de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para a obtenção do grau de Engenheira Eletricista.

Aprovada em ____ / ____ / ____

Professor Avaliador

Universidade Federal de Campina Grande
Avaliador

George Acioli Júnior

Universidade Federal de Campina Grande
Orientador

Campina Grande - PB

Maio de 2021

Este trabalho é dedicado aos meus pais, Elio e Malena, que sempre lutaram para que eu tivesse um futuro digno, e a todos os meus estimados amigos pela paciência e companheirismo ao longo de toda essa jornada.

Agradecimentos

a Deus, sobre todas as coisas, pois a Ele tudo devo e por tudo sou eternamente grata. A nossa senhora de Fátima que atendeu meus pedidos e todas as vezes que pedi para ela passa na frete das minhas decisões e ações ela estava lá.

Aos meus pais, Elio e Malena , que desde cedo batalharam para, dentro de suas possibilidades, oferecerem sempre o melhor para os seus filhos. Aos meus irmãos, Alinny e Alexandre que são os bens mais preciosos que tenho na vida.

Agradeço também a toda minha família, que com todo apoio e carinho contribuíram para que eu chegasse até aqui.

Agradeço ao professor George Acioli por aceitar orientar este trabalho, pelos conhecimentos transmitidos, pelas orientações e contribuições.

Agradeço também aos professores Péricles Rezende e Rafael Bezerra por um dia me receberem no LIEC, me direcionando para a ênfase que escolhi seguir. O conhecimento adquirido no LIEC foi fundamental para a minha formação.

Aos grandes amigos da UFCG, Lizandra, Stayner, Giordano, Giovanni, Marcus, Laecio, Vagne, Valmir e Samuel pessoas com quem muito aprendi, e passei com certeza, os melhores e os piores momentos do curso.

Além deles, sou grato pela amizade cada vez mais viva dos conterrâneos, Victor Hugo, Paulo Roberto, Lucas de Lima e Paulo César. Estes a quem considero desde o tempo do IFCE.

Enfim, agradeço a todos que de alguma forma, passaram pela minha vida e contribuíram para a construção de quem sou hoje.

Resumo

O projeto desenvolvido, visando a situação de pandemia provocada pelo corona vírus, idealizou para Laboratório de Automação do LIEC a construção de práticas virtuais para os alunos de Engenharia Elétrica. As Aulas elaboradas permitirão aos alunos abordarem tópicos de instrumentação e controle de manufaturas através da simulação de instrumentos utilizados em aplicações industriais e de malhas de controle. Além disso, essas atividades podem ser usadas para o aprendizado na área de Redes Industriais e para o desenvolvimento de aplicações de *Softwares* Supervisório. A montagem dessas cenas auxilia no processo de aprendizado e na evolução da educação técnica, estimulando a aquisição do conhecimento. Por meio das práticas virtuais, os alunos podem experimentar a sensação de atuar sobre um processo industrial, mas em um ambiente distante deste e com seu principal uso como atividade didática complementar. Assim, partindo dessa premissa é essencial que o engenheiro se familiarize com essa aproximação técnica, que é de fundamental importância para sua formação.

Palavras-chave: Laboratório Virtual, Manufatura, Simulação, Automação.

Abstract

©.

The developed project, aiming at the pandemic situation caused by the corona virus, idealized for LIEC's Automation Laboratory the construction of virtual practices for Electrical Engineering students. The elaborated classes will allow students to approach instrumentation and manufacturing control topics through the simulation of instruments used in industrial applications and control loops. In addition, these activities can be used for learning in the area of Industrial Networks and for the development of Supervisory *Softwares* applications. The assembly of these scenes helps in the learning process and in the evolution of technical education, stimulating the acquisition of knowledge. Through virtual practices, students can experience the sensation of acting on an industrial process, but in an environment far from it and with its main use as a complementary didactic activity. Thus, based on this premise, it is essential that the engineer becomes familiar with this technical approach, which is of fundamental importance for his training.

Keywords: Virtual Laboratory, Manufacturing, Simulation, Automation.

Lista de ilustrações

Figura 1 – O sistema de controle de processos por computador.	7
Figura 2 – Quadro Elétrico.	10
Figura 3 – Botão de Emergência.	11
Figura 4 – Botões.	11
Figura 5 – Potenciômetro.	12
Figura 6 – Display.	12
Figura 7 – Esteira Transportadora.	13
Figura 8 – Tanque de Nível.	15
Figura 9 – Esteira com Contagem de Caixas.	16
Figura 10 – Equipamento de montagem.	17
Figura 11 – Controle de Nível.	19
Figura 12 – Célula.	21
Figura 13 – Matéria Prima.	22
Figura 14 – Tampa do Produto.	22
Figura 15 – Base do Produto.	23
Figura 16 – Produto final.	23
Figura 17 – Estação de Distribuição.	24
Figura 18 – Estação de Usinagem.	25
Figura 19 – Estação de Classificação de Peças.	27
Figura 20 – Estação Montagem de Peças.	29
Figura 21 – Painel de Controle.	31
Figura 22 – Sistema SCADA.	33
Figura 23 – Câmeras	38
Figura 24 – Cenas	39
Figura 25 – Cenas	39
Figura 26 – Tags	40
Figura 27 – Novo Projeto	41
Figura 28 – Janela para seleção do dispositivo	42
Figura 29 – Árvore hierárquica inicial da configuração do projeto	42
Figura 30 – Janela para edição de POU's	43
Figura 31 – Painel de controle	49
Figura 32 – Variáveis	50
Figura 33 – Comunicação via OPC	51
Figura 34 – PLC	51
Figura 35 – PLC	52
Figura 36 – Contatos	53

Figura 37 – Acionamento dos botões	54
Figura 38 – Acionamento do motor	54

Lista de tabelas

Tabela 1 – Tags da prática 1	13
Tabela 2 – Tags da prática 2	14
Tabela 3 – Tags da prática 3	16
Tabela 4 – Tags da prática 4	18
Tabela 5 – Tags da prática 5	20
Tabela 6 – Tags da Estação de Distribuição	24
Tabela 7 – Tags da Estação de Usinagem.	26
Tabela 8 – Codificação dos itens.	27
Tabela 9 – Tags Empurrador pneumático.	28
Tabela 10 – Tags Estação Montagem de Peças	30

Lista de abreviaturas e siglas

CAD	Computer Aided Design
CLP	Controladores Lógicos Programáveis
CNC	Computador Numérico
E/S	Entrada e saídas
GD	Gêmeos Digitais
I/O	Input/Output
IIoT	Industrial Internet of Thing
IP	Internet Protocol
NA	Normalmente Aberto
NF	Normalmente Fechado
LIEC	Laboratório de Automação Industrial
PID	Controlador proporcional integral derivativo
POU	Unidades de Organização de Programa
SA	Sistema Automatizado
SAI	Sistema de Automação Industrial
SCADA	Supervisory Control and Data Acquisition
SFM	Sistema Flexível de Manufatura

Sumário

1	INTRODUÇÃO	1
1.1	Justificativa	2
1.2	Objetivos	2
1.3	Metodologia e Organização do Trabalho	3
2	FUNDAMENTAÇÃO TEÓRICA	4
2.1	Conceitos Básicos de Automação Industrial	4
2.2	Principais Elementos de Operação	6
2.3	Manufatura	7
2.4	Conceito de Gêmeos Digitais	8
3	PRÁTICAS VIRTUAIS	10
3.1	Operadores	10
3.2	Prática Virtual 1: Esteira Transportadora	12
3.3	Prática Virtual 2: Acionamento do Tanque de Nível	14
3.4	Prática Virtual 3: Esteira com Contagem de Caixas	15
3.5	Prática Virtual 4: Equipamentos de Montagem	17
3.6	Prática Virtual 5: Controle de Nível	19
4	CÉLULA DE MANUFATURA VIRTUAL DIRECIONADA AO EN- SINO DE AUTOMAÇÃO	21
4.1	Descrição da Célula	21
4.1.1	Estação de Distribuição de Matéria Prima	24
4.1.2	Estação de Usinagem	25
4.1.3	Estação Classificação de Peças	27
4.1.4	Estação de Montagem de Peças	29
4.2	Visão Geral	30
4.3	Sistema de Supervisão	31
5	CONCLUSÕES	34
	REFERÊNCIAS BIBLIOGRÁFICAS	35
	ANEXO A – GUIA DO LABORATÓRIO VIRTUAL	36
A.1	Apresentação do Factory I / O	36
A.1.1	Instalação	36

A.1.2	Navegando	37
A.1.2.1	Câmera Orbit	38
A.1.2.2	Câmera Fly	38
A.1.2.3	Câmera First Person	38
A.1.3	Cenas	38
A.1.4	Peças	39
A.1.5	Tags	39
A.2	Projeto no CODESYS	40
A.2.1	Instalação	40
A.2.2	Criação de um novo projeto	40
A.2.3	Criação de POUs	42
A.2.4	Edição de POUs	43
A.2.5	Compilação do projeto	43
A.2.6	Instruções mais utilizadas	43
A.2.6.1	Sinal de Entrada (Contato)	43
A.2.6.2	Sinal de Saída Comum (Bobina)	44
A.2.7	Sinal de Saída Retentivo (Set / Reset)	44
A.2.7.1	Temporizadores	44
A.2.7.2	Contadores	46
A.3	Acionamento da esteira transportadora	49
A.3.1	Configurando o factory I / O	49
A.3.2	Configuração da comunicação OPC	50
A.3.2.1	Codesys	50
A.3.2.2	Factory I / O	52
A.3.3	Programação no Codesys	53

1 Introdução

Os laboratórios tem um importante papel na formação do aluno de engenharia, uma vez que é um lugar onde várias teorias são testadas e possibilita a adaptação do estudante às mais diversas condições que podem ser favoráveis ou desfavoráveis quando futuramente encontradas no exercício da profissão. As disciplinas desenvolvidas em laboratórios, buscam aplicar os conceitos teóricos aprendidos em sala de aula, proporcionando ao aluno uma consolidação desses conceitos além de apresentar equipamentos relacionados a operações e processos industriais, em escala de laboratório, propiciando-lhe um melhor entendimento dos processos bem como seu princípio de funcionamento.

Tendo em vista a situação de pandemia provocada pelo corona vírus, os laboratórios virtuais e laboratórios remotos fornecem uma excelente solução para os problemas enfrentados por universidades de engenharias. Além disso, pode proporcionar para alguns laboratórios que não possuem uma estrutura completa a possibilidade de práticas a partir de simulações realistas.

De maneira a contribuir no processo de aprendizado e na melhoria da educação técnica, levando em consideração o cenário atual, é possível com a utilização de alguns softwares que os alunos experimentem a sensação de atuar sobre um processo real, mas em um ambiente virtual. Permitindo assim, a realização das atividades essenciais do laboratório da disciplina de sistemas de automação industrial.

A parte prática da disciplina de sistemas de automação industrial, atualmente é composta principalmente por: programação e configuração de controladores lógicos programáveis (CLP), instrumentação industrial (Sensores e Atuadores) e sistemas de manufatura. Por meio de um conjunto de *softwares*, como o Factory I / O, Codesys e Elipse SCADA é possível realizar os experimentos da disciplina de forma remota.

Este trabalho tem a finalidade de apresentar a capacidade desses *softwares* de simular as práticas realizadas atualmente no laboratório da disciplina de SAI e realizar a simulação e programação em *ladder* de uma linha de produção industrial.

1.1 Justificativa

Este trabalho consolidará o que vêm sendo desenvolvido no Laboratório de Sistemas de Automação Industrial (SAI), Departamento de Engenharia Elétrica da Universidade Federal de Campina Grande, na aplicação de conceitos de automação industrial para a formação de estudantes e pesquisadores de graduação, na ênfase de Controle e Automação.

A proposta de simulações de cenas virtuais modulares com arquiteturas de Controle e Supervisão genérica e aberta, de baixo custo, permitirá o desenvolvimento de experimentos práticos completos e complexos, dentro de um ambiente de aprendizagem que exemplifica tarefas rotineiras (Movimentação, Classificação, Montagem, Inspeção e Supervisão), e que ocorrem normalmente em sistemas automatizados industriais.

1.2 Objetivos

O objetivo do trabalho consiste em obter meios remotos para aplicação dos experimentos realizados no laboratório de automação industrial do LIEC, por meio dos seguintes *softwares* de automação:

- Factory I / O;
- Codesys;
- Elipse SCADA.

Eles iram ajudar o aluno com práticas, como programação em *ladder*, configuração de CLPs, sistemas de manufatura e sistema supervisorio. É objetivo também a criação de uma documentação em formato de guia experimental referente ao uso dos *softwares* e como aplicação a utilização do conjunto de *softwares* para a simulação de uma linha de produção industrial.

Dos objetivos específicos, busca-se:

- Aprofundar os estudos no âmbito da Automação Industrial.
- Desenvolver habilidades na linguagem de programação *ladder*.
- Desenvolver atividades voltadas ao controle de processos industriais utilizando *softwares* de automação.
- Desenvolver um guia para que os futuros alunos possam utilizar os *softwares* no desenvolvimento de projetos na disciplina de Sistemas de Automação Industrial.

1.3 Metodologia e Organização do Trabalho

O trabalho está estruturado em 5 capítulos, incluindo este introdutório, conforme a seguir.

O **Capítulo 2** apresenta uma fundamentação teórica, referente ao tema pesquisado, mediante livros, artigos científicos, dissertações, teses, tutoriais e manuais de instruções.

O **Capítulo 3** é feita a descrição de todas as práticas virtuais proposta para o laboratório virtual. Cada prática contém um tópico de descrição da cena, de programação em *ladder* e um exercício proposto.

No **capítulo 4** é apresentada uma célula de manufatura virtual direcionada ao ensino de automação. Será elaborado uma simulação computacional nos *softwares* Codesys, Factory I / O, e Elipse SCADA para realizar a construção, programação e supervisão da célula virtual.

O **Capítulo 5** apresenta as conclusões do trabalho em um formato do resumo dos resultados alcançados e propostas futuras visando complementar as atividades desenvolvidas.

2 Fundamentação Teórica

2.1 Conceitos Básicos de Automação Industrial

Automação é todo processo que realiza tarefas e atividades de forma autônoma ou que auxilia o homem em suas tarefas do dia-a-dia. As antigas rodas d'água, os pilões e os moinhos são considerados sistemas automatizados. Com o advento das máquinas, principalmente após a chegada da máquina a vapor, a automação estabeleceu-se dentro das indústrias e, como consequência imediata, a elevação da produtividade e da qualidade dos produtos e dos serviços. Ainda assim a automação era muito dependente do homem, pois havia máquinas automáticas espalhadas pela fábricas, mas sem integração entre elas.

Um conceito mais abrangente de automação pode ser definido como a integração de conhecimentos substituindo a observação, os esforços e as decisões humanas por dispositivos (mecânicos, elétricos e eletrônicos, entre outros) e *softwares* concebidos por meio de especificações funcionais e tecnológicas, com uso de metodologias. [Rosário \(2009\)](#)

A automação industrial na maioria da vezes processa-se da seguinte maneira: um computador recebe os sinais provenientes dos vários instrumentos de medidas da fábrica, compara tais medidas com os valores ideais e realiza operações matemáticas com a finalidade de gerar sinais de correção, que instruirão os dispositivos de controle acerca da alteração mais apropriada para cada instante, com o intuito de conduzir a uma produção ótima sob um determinado ponto de vista, seja ele qualitativo ou quantitativo.

Com isso, a automação industrial pode ser dividida em três classes: a fixa, a flexível e a programável. [Rosário \(2009\)](#)

- A automação fixa é utilizada quando o volume de produção é elevado; nesta forma, a linha de produção de produção é composta de diversas máquinas de Comando - CN, chamadas estações de trabalho. Nessa estações é realizado um conjunto de operações e, à medida que são terminadas, as peças são transferidas a outras estações. Dessa forma a produção possui uma linha de produção fixa, voltada apenas para a concepção de um determinado tipo de produto.

- A automação flexível é utilizada para um volume de produção média, decorrente da maior interação da área de engenharia mecânica com tecnológicas eletrônicas embarcadas e sistemas de informação. Nesta forma, a automação, aliada à flexibilidade, possibilita que sejam fabricados diversos produtos ao mesmo tempo, utilizando-se o mesmo sistema de produção.
- A automação programável é utilizada para um volume de produção relativamente baixo e diversificado, ou seja, a produção é efetuada em pequenos lotes. Nesta forma de produção os equipamentos devem ser reprogramados a cada novo lote.

Sistema automatizado (S.A.) é um conjunto de elementos dinamicamente relacionados entre si, formando uma atividade para atingir um objetivo, operando sobre entradas (informação, energia, ou matéria) e fornecendo saídas (informação, energia ou matéria) processadas. O conceito de automação pode ser exemplificado através de uma analogia com o corpo humano, que é possui de sensores (visão, paladar, olfato, tato, ouvidos), os atuadores (braços, mãos, pernas), um controlador (cérebro), e o esqueleto (estrutura mecânica). [SARAMAGO \(2002\)](#)

2.2 Principais Elementos de Operação

Para implementar a automação e o controle de processos, o computador de controle deve coletar dados do processo de produção e transmitir sinais a ele. As variáveis e os parâmetros de processo são classificados como contínuos ou discretos. [GROOVER \(2010\)](#)

Uma variável contínua (ou parâmetro) é a que se mantém ininterrupta conforme o tempo procede, pelo menos durante a operação de produção. Uma variável contínua é de modo geral considerada analógica, o que significa que pode assumir qualquer valor dentro de um determinado intervalo; ela não é restrita a um conjunto discreto de valores. As operações de produção tanto nas indústrias de processo como nas de produção de peças discretas são caracterizadas por variáveis contínuas. Exemplos incluem força, temperatura, vazão, pressão e velocidade.

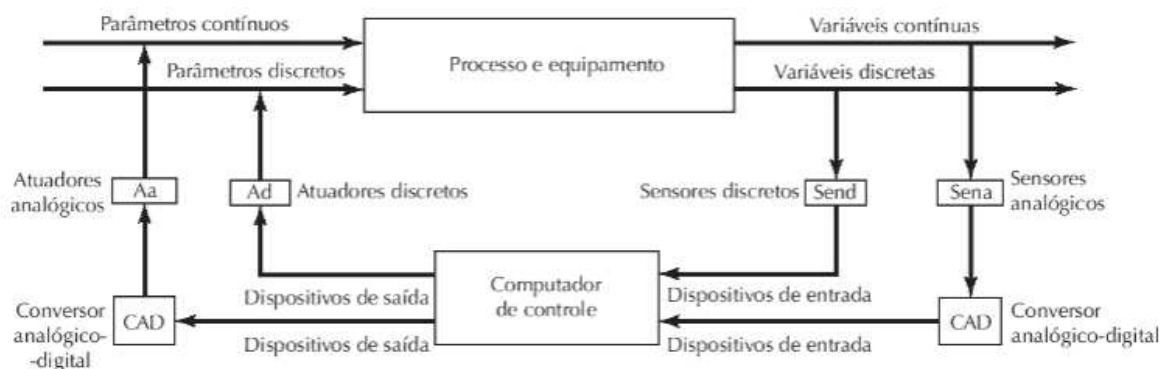
Uma variável discreta (ou parâmetro) pode assumir apenas certos valores em um dado intervalo. O tipo mais comum de variável discreta é a binária, o que quer dizer que ela pode assumir um de dois valores possíveis, ligado ou desligado, aberto ou fechado, e assim por diante. Exemplos de variáveis e parâmetros discretos binários em produção incluem interruptor fim-de-curso aberto ou fechado, motor ligado ou desligado e peça de trabalho presente ou ausente em uma instalação.

O computador digital opera com dados digitais (binários), enquanto alguns dados do processo são contínuos ou analógicos. É preciso acomodar essa diferença na interface entre o processo e o computador. Os componentes necessários à implementação dessa interface são os seguintes [GROOVER \(2010\)](#):

- Sensores para medir as variáveis contínuas e discretas do processo;
- Atuadores que acionam os parâmetros contínuas e discretos dos processos;
- Dispositivos que convertem sinais analógicos em dados digitais;
- Dispositivos que convertem dados digitais em sinais analógicos;
- Dispositivos de entrada/saída para dados discretos.

A Figura 1 apresenta a configuração geral do sistema de controle de processos por computador e como essas cinco categorias de componentes são utilizadas na criação da interface do processo com o computador. Esse modelo representa o arranjo geral da maioria dos sistemas de manuseio de material e dos sistemas de manufatura.

Figura 1 – O sistema de controle de processos por computador.



Fonte: GROOVER (2010).

2.3 Manufatura

Podemos definir sistema de manufatura como uma coleção de equipamentos e recursos humanos integrados, cuja função é realizar uma ou mais operações de processamento e/ou montagem na matéria-prima, na peça ou em um conjunto inicial de peças. O equipamento integrado inclui máquinas e ferramentas de produção, dispositivos e manuseio de material e de posicionamento de trabalho e sistemas de computador. Os recursos humanos são necessários em tempo integral ou periodicamente para manter o sistema em operação. No sistema de manufatura, o trabalho de agregação de valor é realizado nas peças e nos produtos. GROOVER (2010)

Um Sistema Flexível de Manufatura (SFM) é um agrupamento de estações de trabalho parcial ou totalmente independentes, monitorado por um controlador central, interligadas através de um sistema automatizado de transporte (ou manuseio). Sua implantação é indicada quando se produz uma alta variedade de peças, em volumes de produção, baixo e médio. MARTINS (1999)

MOREIRA (S.d.) define sistema de manufatura celular baseado no conceito de tecnologia de grupo por meio de formação de famílias de peças e células de manufatura, como é explicado a seguir:

- Família de peças é constituída de características de atributos similares seja ela geométrica e/ou processo de fabricação.
- A célula de manufatura é constituída por um agrupamento de máquinas e/ou equipamentos capazes de processar uma dada família de peças.

Dentre as principais vantagens de um sistema de manufatura celular podemos destacar o menor ciclo e custo de fabricação, número de operadores; a simplificação do fluxo de fabricação e controle de produção; a redução de refugos e retrabalho, “startup” do processo e de transporte e movimentação e a melhoria da qualidade do produto. [ROJAS \(2004\)](#)

2.4 Conceito de Gêmeos Digitais

A tecnologia chamada de gêmeos digitais (GDs) consiste na digitalização de componentes, produtos e sistemas físicos com a decorrente geração de modelos computacionais, por meio de descrições físicas e funcionais, que visam replicar a realidade por meio de simulações das condições de uso e fabricação destes mesmos produtos, visando as áreas de desenvolvimento de produto e processos industriais. [SCHLEICH \(2017\)](#)

De acordo com [GE \(2008\)](#), os três elementos que compõem um gêmeo digital são:

- Modelo do ativo: Descreve detalhadamente sua estrutura e componentes, de forma hierarquizada.
- Análises: Preveem e descrevem seu comportamento, possibilitando a automatização de comportamentos.
- Base de Conhecimentos: Dados acumulados a partir da captação de variáveis relacionadas ao ativo do mundo real, apresentados de forma intuitiva.

[PARROT \(2017\)](#) define GDs como um perfil digital em evolução do comportamento histórico e atual de um objeto físico ou processo que ajuda a otimizar o desempenho de uma organização.

A principal diferença dos GDs em relação aos CADs – desenho assistido por computador, ou em inglês computer-aided design – ou das próprias ferramentas da IIoT, consiste que o primeiro é encapsulado pelo software, e embora já tenha se provado moderadamente útil e eficiente em sua proposta, não possui a interação com o ambiente; o segundo, por sua vez, é um sistema responsável por mensurar ou diagnosticar objetos reais, falhando em demonstrar interações entre componentes ou o ciclo de vida do produto [PARROT \(2017\)](#).

A coleta de informações ao nível físico é viabilizada pela utilização de uma diversidade de sensores, espalhados pelo chão de fábrica, cada qual responsável por prover ao sistema um conjunto de informações a respeito de uma variável física, normalmente associada a um equipamento ou etapa do processo produtivo, por exemplo, pressão, temperatura, medição de espessura, dentre outras. Juntamente com estes sensores, conectam-se atuadores, ou dispositivos capazes de causar alguma alteração no meio físico como motores, cilindros pneumáticos, válvulas, impressoras 3D, robôs, etc. A comunicação entre sensores (provedores de informação) e dispositivos que utilizam as informações produzidas, sejam atuadores, computadores ou usuários, ocorre por uma rede, baseada em arquiteturas abertas como ethernet e IP, que interconecta todo este conjunto. Dentro do paradigma da Indústria 4.0, esta rede chama-se Industrial Internet (ou Internet Industrial), e o conjunto de dispositivos (sensores e atuadores) é nomeado Industrial Internet of Things (IIoT) [MADEIRA \(2016\)](#).

Uma vez digitalizado, as ações executadas no objeto físico também são realimentadas para o gêmeo digital, tais como manutenções, mudanças de especificações, substituição de peças, de tal forma que possa se obter uma representação fiel do equipamento. Assim, ao interagir com o gêmeo digital, pode-se ter uma comparação dos estados “conforme construído” versus “conforme mantido” [PARROT \(2017\)](#).

Neste trabalho a solução encontrada para representar os gêmeos digitais para o laboratório virtual é o aplicativo *Factory I/O*, uma ferramenta que permite a rápida construção de processos de fabricação virtuais utilizando peças industriais comuns, tais como esteiras, sensores, elevadores, manipuladores e outros. O acionamento dos equipamentos de campo é realizado por meio de sinais externos, que podem ser obtidos por meio de placas de aquisição de dados, comunicação com CLPs compatíveis ou por meio de protocolos de comunicação.

3 Práticas Virtuais

Este capítulo é voltado para a construção de práticas virtuais para o laboratório de automação industrial. Com essas práticas os alunos poderão exercitar a programação e configuração de controladores lógicos programáveis (CLP), conhecer mais sobre instrumentação industrial (Sensores e Atuadores) e sistemas de manufatura.

O FACTORY I/O é o software escolhido para realizar essas práticas. Ele foi criado para o ensino e desenvolvimento de aplicações em automação industrial. Altamente flexível, esta ferramenta possibilita criar plantas industriais virtuais que podem ser controladas por tecnologias externas, tais como Soft PLCs, CLPs, microcontroladores, etc.

3.1 Operadores

Antes de começar a apresentar as práticas virtuais é de suma importância listar os principais operadores que serão usados em todas as simulações construídas no software.

- **Quadro Elétrico**

Usado para simular um quadro elétrico com operadores.

Figura 2 – Quadro Elétrico.



Fonte: [Factoryio \(2020\)](#).

- **Botão de Emergência**

Botão normalmente fechado de duas posições e não iluminado. Normalmente usado em eventos de emergência.

Figura 3 – Botão de Emergência.



Fonte: [Factoryio \(2020\)](#).

- **Botões**

Botões de pressão iluminado disponível em três cores diferentes (verde: iniciar, amarelo: reiniciar e vermelho: parar). Os botões de pressão pode ser usado como uma ação momentânea ou alternativa, de acordo com a configuração selecionada.

Figura 4 – Botões.

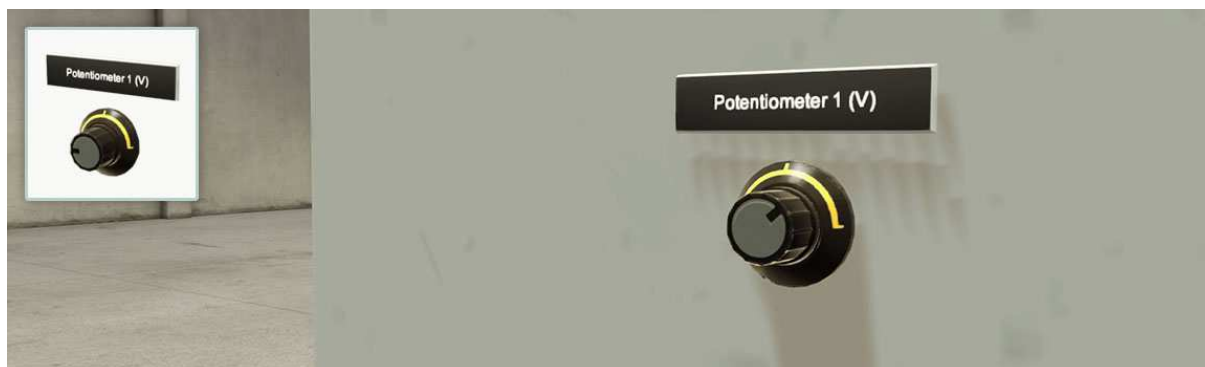


Fonte: [Factoryio \(2020\)](#).

- **Potenciômetro**

Usado para gerar um valor analógico proporcional à rotação do botão. A configuração selecionada pode definir a faixa de valores.

Figura 5 – Potenciômetro.

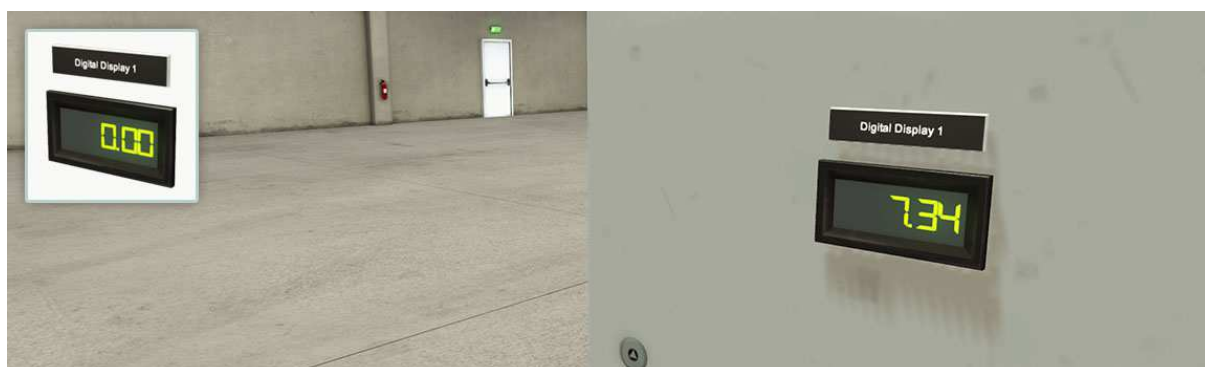


Fonte: [Factoryio \(2020\)](#).

- **Display**

Permite que valores numéricos sejam exibidos durante a simulação. A configuração selecionada define o intervalo de valores e o tipo de dados.

Figura 6 – Display.



Fonte: [Factoryio \(2020\)](#).

3.2 Prática Virtual 1: Esteira Transportadora

Essa primeira prática tem como objetivo principal familiarizar os alunos com os softwares de simulação e programação. A figura 7 mostra a cena virtual a ser configurada, ela consiste em uma esteira que transporta uma caixa do sensor A para um sensor B. Na tabela 1 estão as descrições dos atuadores e sensores presente na cena.

Tabela 1 – Tags da prática 1

Tag	E / S Controlador	Tipo	Descrição
Sensor_A	Entrada	Bool	Nível lógico zero quando o feixe de luz é interrompido.
Sensor_B	Entrada	Bool	Nível lógico zero quando o feixe de luz é interrompido.
Esteira_1	Saída	Bool	Gira a correia (direção da seta).
Esteira_2	Saída	Bool	Gira a correia (direção da seta).

Figura 7 – Esteira Transportadora.



Fonte: [Factoryio \(2020\)](#).

Para facilitar a programação do código em *ladder* é apresentado abaixo uma descrição narrativa de como a simulação deve funcionar.

Descrição narrativa para a programação no codesys:

- Acionar a esteira 1 ao apertar o botão start ;
- A esteira 2 será acionada quando a caixa passar pelo sensor_A;
- A esteira 1 só é desligada quando a caixa terminar de passar pelo sensor_A;
- A esteira 2 é desligada quando a caixa chegar ao sensor_B;
- Botão stop e emergência para o processo em qualquer instante, quando acionados.

3.3 Prática Virtual 2: Acionamento do Tanque de Nível

O objetivo desta segunda prática é a utilização dos temporizadores. Os temporizadores são uma das partes mais importante na programação de CLPs. As funções TON e TOF são fundamentais para a utilização correta do tempo, nas mais diversas lógicas de programação. A figura 8 mostra a cena virtual a ser configurada, ela consiste em um tanque de nível incluindo duas válvulas de controle e um sensor de nível capacitivo que pode ser usado para controlar o fluxo de líquido dentro e fora do tanque. As válvulas de controle são controladas por atuadores pneumáticos que podem ser posicionados com sinais entre 0 e 10 V. Os sensores capacitivos também podem ser usados para detectar níveis de líquido.

Tabela 2 – Tags da prática 2

Tag	E / S Controlador	Tipo	Descrição
Medidor de Nível	Entrada	Float	[0,10]V: posicionamento da válvula de enchimento
Medidor de Fluxo	Entrada	Float	[0,10]V: posicionamento da válvula de descarga.
Válvula de Enchimento	Saída	Bool	Abre a válvula de enchimento.
Válvula de Descarga	Saída	Bool	Abre a válvula de descarga

Figura 8 – Tanque de Nível.



Fonte: [Factoryio \(2020\)](#).

Para facilitar a programação do código em *ladder* é apresentado abaixo uma descrição narrativa de como a simulação deve funcionar.

Descrição narrativa para a programação no codesys:

- Primeiro passo é acionar a válvula de enchimento por um determinado tempo ao pressionar um botão;
- Depois se deve acionar a válvula de descarga por um determinado tempo ao pressionar um segundo botão ;
- O tempo de acionamento das válvulas devem aparecer em displays.

3.4 Prática Virtual 3: Esteira com Contagem de Caixas

A prática três tem como objetivo apresentar um elemento que tem muita importância em atividades na indústria que necessitam de uma ferramenta para contagens. O contador é um importante bloco de lógica em muitas programações em CLPs. A figura

abaixo mostra uma esteira que carrega e descarrega três caixas, um painel com os botões necessários para o controle da cena e um display para mostrar a contagem das caixas.

Tabela 3 – Tags da prática 3

Tag	E / S Controlador	Tipo	Descrição
Sensor_A	Entrada	Bool	Nível lógico zero quando o feixe de luz é interrompido.
Sensor_B	Entrada	Bool	Nível lógico zero quando o feixe de luz é interrompido.
Sensor_C	Entrada	Bool	Nível lógico zero quando o feixe de luz é interrompido.
Esteira_1	Saída	Bool	Gira a correia (direção da seta).
Esteira_2	Saída	Bool	Gira a correia (direção da seta).

Figura 9 – Esteira com Contagem de Caixas.



Fonte: [Factoryio \(2020\)](#).

Descrição narrativa para a programação no codesys:

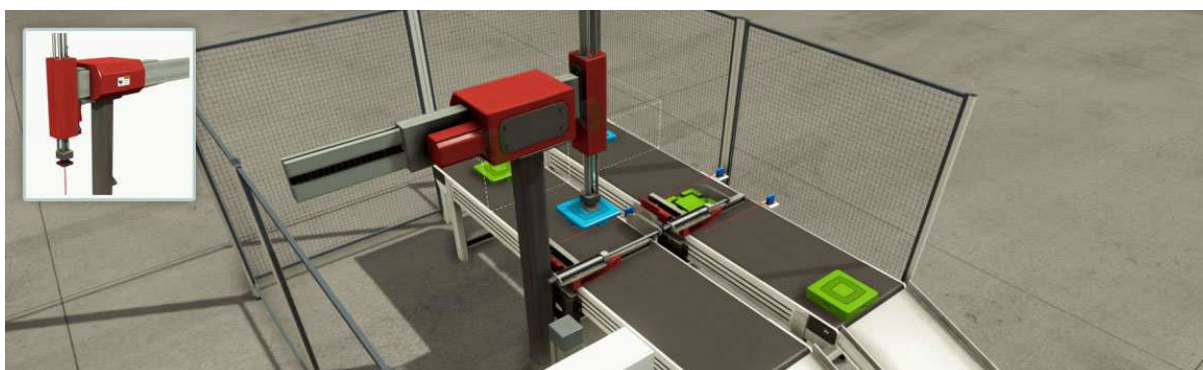
- Acionar a esteira 1 ao pressionar o botão start e o sensor_A em nível lógico 0;
- A esteira 1 para e a esteira 2 será acionada quando três caixas passarem pelo sensor_B;
- Por fim, quando as três caixas passarem pelo sensor_C, a esteira 2 para enquanto a esteira 1 volta a funcionar.
- A quantidade de caixas deve aparecer no display do painel.
- Botão stop e emergência para o processo em qualquer instante, quando acionados.

3.5 Prática Virtual 4: Equipamentos de Montagem

A prática virtual quatro tem o objetivo de propor ao aluno um exercício com as variáveis analógicas. A simulação consiste em montar peças feitas de tampas e bases usando um pick and place de dois eixos. Para garantir um ajuste correto, as bases e tampas devem ser devidamente alinhadas por barras de posicionamento. As características do equipamento são listadas abaixo:

- Curso do eixo X: 1,125 m
- Curso do eixo Z: 0,625 m
- Velocidade do braço e do seletor: 2 m / s
- Rotação do eixo Z em incrementos de 90 ° (braço e garra)

Figura 10 – Equipamento de montagem.



Fonte: [Factoryio \(2020\)](#).

A figura 4 mostra como a estação de montagem deve ser montada no Factory I/O e na tabela quatro estão todas as tags e descrições para facilitar a construção do código em *ladder*.

Tabela 4 – Tags da prática 4

Tag	E / S Controlador	Tipo	Descrição
Posição X	Entrada	Float	[0,10]V: posição atual ao longo do eixo X.
Posição Z	Entrada	Float	[0,10]V: posição atual ao longo do eixo Z.
Rotação	Entrada	Bool	Rotação do equipamento.
Sensor_ventosa	Entrada	Bool	Sensor ventosa giratória.
Sensor_item	Entrada	Bool	Sensor para detectar o item próximo a ventosa.
Sensor_Tampa	Entrada	Bool	Sensor para detectar a chegada da tampa
Sensor_Base	Entrada	Bool	Sensor para detectar a chegada da base
Ponto de ajuste X	Saída	Float	[0,10]V: definir a posição alvo ao longo do eixo X.
Ponto de ajuste Z	Saída	Float	[0,10]V: definir a posição alvo ao longo do eixo Z.
Giro horário	Saída	Bool	Rotação do equipamento sentido horário.
Giro anti-horário	Saída	Bool	Rotação do equipamento sentido anti-horário.
Giro ventosa horário	Saída	Bool	Rotação da ventosa sentido horário.
Giro ventosa anti-horário	Saída	Bool	Rotação da ventosa sentido anti-horário.
Ventosa	Saída	Bool	Ativar a ventosa.
Esteira_Tampa	Saída	Bool	Esteira que fornece tampas
Esteira_Base	Saída	Bool	Esteira que fornece bases
Esteira_Peça	Saída	Bool	Esteira com a peça montada
Barra_Tampa	Saída	Bool	Barra que posiciona a tampa
Barra_Base	Saída	Bool	Barra que posiciona a base

Descrição narrativa para a programação no codesys:

- Ao pressionar o botão start as esteiras iniciais devem fornecer uma tampa e uma base para a montagem da peça e os eixos X e Z são posicionados no estado inicial;
- Após os sensores detectarem a chegada da base e da tampa, as barras de posicionamento serão acionadas para fixar os objetos.
- A posição em torno do eixo X é definida no centro da tampa;

- Quando o sensor_item detectar a tampa, será acionada a ventosa;
- A posição em torno do eixo X é definida novamente para posicionar a tampa em cima da base;
- A posição em torno do eixo Z é definida para encaixar a tampa na base, montando assim a peça;
- A barra de posicionamento libera a peça e os eixos X e Z são posicionados no estado inicial aguardando um novo processo de montagem.
- Botão stop e emergência para o processo em qualquer instante, quando acionados.

3.6 Prática Virtual 5: Controle de Nível

A finalidade da prática virtual cinco é uso do bloco PID, disponível no codesys, para o controle de nível e fluxo. Com o controlador e com as especificações da cena virtual, listada abaixo, é possível encontrar um controlador.

- Altura: 3 m;
- Diâmetro: 2 m;
- Raio do tubo de descarga: 0,125 m;
- Máx. vazão de entrada: $0,25 \text{ m}^3 / \text{s}$;
- Máx. fluxo de saída: $0,3543 \text{ m}^3 / \text{s}$;
- Sensores capacitivos podem detectar líquido.

Figura 11 – Controle de Nível.



Fonte: [Factoryio \(2020\)](#).

Tabela 5 – Tags da prática 5

Tag	E / S Controlador	Tipo	Descrição
medidor de nível	Entrada	Float	[0, 10] V: valor do medidor de nível.
medidor de fluxo	Entrada	Float	[0, 10] V: valor do fluxômetro (10 V = 0,3543 m ³ / s).
Válvula de enchimento	Saída	Float	[0, 10] V: posicionamento da válvula de enchimento.
Válvula de descarga	Saída	Float	[0, 10] V: posicionamento da válvula de descarga.

Descrição narrativa para a programação no codesys:

- Ao definir o valor de setpoint pelo potenciômetro as válvulas de encher e de esvaziar o tanque devem ser abertas;
- O nível deve estabilizar no valor desejado.
- Um display deve mostrar a variação do nível

4 Célula de Manufatura Virtual Direcionada ao Ensino de Automação

4.1 Descrição da Célula

Este capítulo apresentará uma célula que foi construída para tentar simular uma cena que requer a maioria dos blocos de função em *ladder*. Em outras palavras, tentou-se desenvolver uma planta que permita o tratamento da maior parte dos conceitos teóricos analisados na disciplina de Automação Industrial, que é ministrada no Laboratório de Instrumentação Eletrônica e Controle (LIEC).

Figura 12 – Célula.



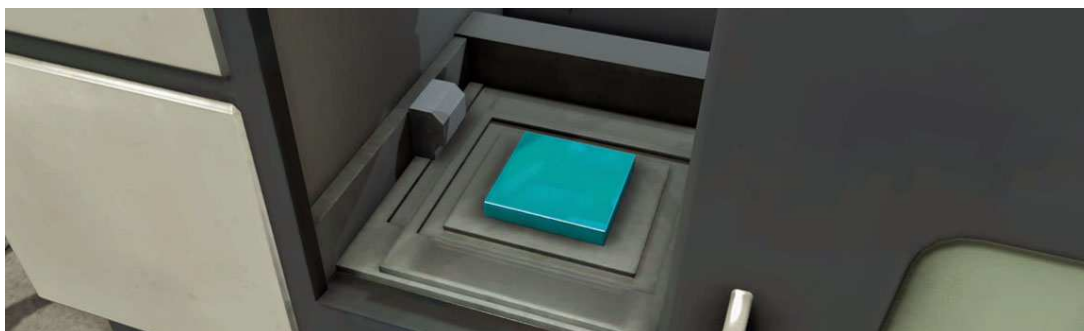
Fonte: [Factoryio \(2020\)](#).

Uma vez que FACTORY I / O oferece itens que podem ser produzidos, esta cena foi elaborada para fazer a distribuição de materiais, usinagem, classificação por cores e montagem das peças. Uma sucessão de matéria prima será incorporada ao sistema, que terá que fazer todos os processos citados. Abaixo são apresentados todos os objetos necessários para montar a peça final.

- **Matéria Prima:**

A matéria-prima é utilizada para produzir tampas e bases de produtos no centro de suinagem . Disponível em três tipos diferentes de material: plástico azul ou verde e metal.

Figura 13 – Matéria Prima.



Fonte: [Factoryio \(2020\)](#).

- **Tampa do Produto:**

Peça que pode ser montada em uma base de produto para produzir um peça final.

Figura 14 – Tampa do Produto.



Fonte: [Factoryio \(2020\)](#).

- **Base do Produto:**

Peça que pode ser montada em uma tampa de produto para produzir um produto final.

Figura 15 – Base do Produto.

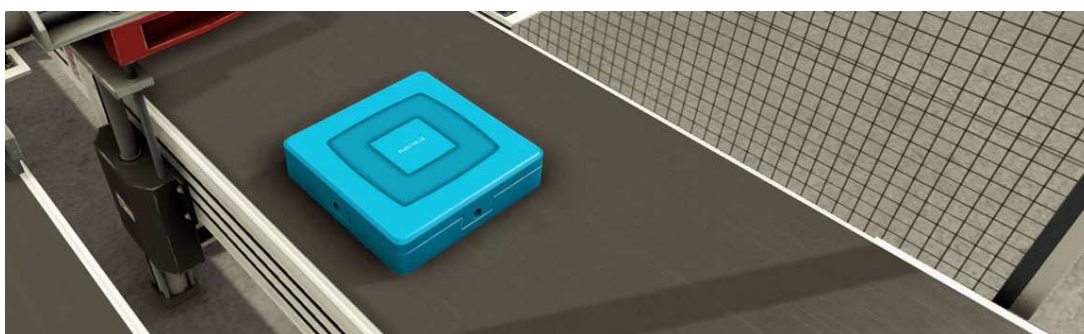


Fonte: [Factoryio \(2020\)](#).

- **Produto Final:**

Criado pela montagem de uma Base e Tampa.

Figura 16 – Produto final.



Fonte: [Factoryio \(2020\)](#).

Como pode ser visto na figura 12, a planta apresenta simetrias em algumas de suas partes. Devido a isso, existem grupos de sensores e atuadores que cumprem ou realizam a mesma função. Isso significa que, por exemplo, localizar e analisar os sensores e / ou atuadores presentes em uma das estações será suficiente, sendo assim não é necessário analisar os do outro, pois, como foi comentado, sua localização e análise seriam totalmente idêntica.

A seguir, estações que compõem a célula serão descritas e no anexo B estará toda a programação em *ladder* desenvolvida para o seu funcionamento.

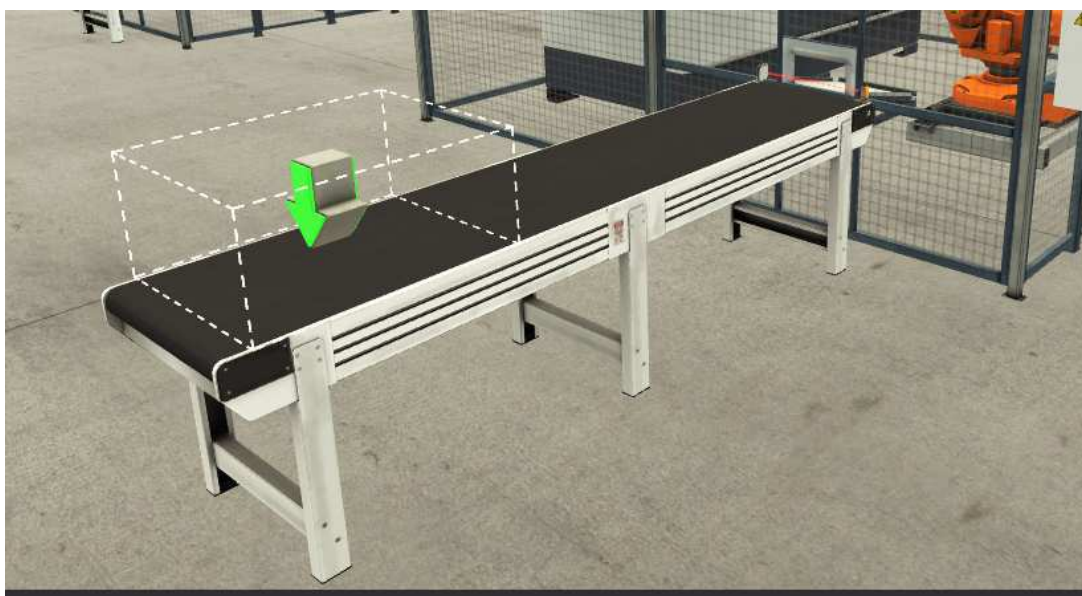
4.1.1 Estação de Distribuição de Matéria Prima

A alimentação de matéria prima no sistema se dá na estação de distribuição. Isso acontece por meio de um emissor de peças e uma esteira. Enquanto um material ainda está dentro do emissor, nenhum outro material é emitido.

O objetivo da estação de distribuição é então:

- Retirar a peça do emissor;
- Posicionar a peça na estação seguinte.

Figura 17 – Estação de Distribuição.



Fonte: [Factoryio \(2020\)](#).

Após o processo de emitir a peça, ela é encaminhada por meio da esteira até a próxima estação. Ao final da esteira a peça corta o feixe de luz do sensor difuso para que a próxima estação possa ser acionada.

Tabela 6 – Tags da Estação de Distribuição

Tag	E / S Controlador	Tipo	Descrição
Esteira_1	Saída	Bool	Esteira que transporta a peça bruto emissor para a próxima estação.
Sensor_Detecta_Material	Entrada	Bool	Sensor difuso que aciona a estação de modelagem toda vez que uma peça bruta passar por ele.

4.1.2 Estação de Usinagem

Esta estação possui duas máquinas de CNC, uma das máquinas tem a função de usinar a matéria prima em uma tampa e a outra máquina tem a função de usinar a matéria prima em uma base, assim dois itens são fornecidos para a montagem do produto final.

Após a matéria prima ser posicionada pela estação de distribuição, o braço robótico da estação de usinagem é acionado devido o corte do feixe de luz do sensor difuso da estação anterior. O braço pega a matéria e posiciona dentro da máquina de CNC. A máquina fecha sua porta e começa o processo de usinagem, depois de alguns segundos dois itens (tampa e base) são fornecidos. Após a usinagem a porta da máquina CNC é aberta e o braço robótico pega o objeto modelado e posiciona ele na próxima estação, cortando novamente um feixe de luz de um novo sensor, para acionar o próximo processo.

Objetivo da estação de usinagem é:

- Usinar a matéria prima
- Fornecer para a próxima estação uma uam tampa e uma base.

Figura 18 – Estação de Usinagem.



Fonte: [Factoryio \(2020\)](#).

A operação de produção pode ser interrompida a qualquer momento, reiniciada ou redefinida. A interação da estação de usinagem pode ser feita por meio do quadro elétrico ao lado da porta de proteção:

Emergência: aciona uma parada de emergência, interrompendo o braço robótico e o CNC. Após uma parada de emergência, a estação só pode ser reiniciada desarmando o botão de emergência e pressionando o botão reiniciar.

Iniciar: inicia a estação.

Parar: para o braço robótico e o CNC. Para reiniciar, é necessário pressionar o botão iniciar ou reiniciar.

Reiniciar: reinicia a estação.

Dois luzes de pilha colocadas no topo da proteção fornecem informações sobre o status atual.

Luz verde: não ocupado.

Luz amarela: ocupado.

Luz vermelha: tem erro (item incorreto detectado no compartimento de entrada).

Tabela 7 – Tags da Estação de Usinagem.

Tags	E / S Controlador	Tipo	Descrição
Produto	Saída	Bool	Define Verdadeiro para produzir tampas; Falso para produzir bases.
Start	Saída	Bool	Iniciar
Stop	Saída	Bool	Parar
Reset	Saída	Bool	Reiniciar
Estação (Ocupada)	Entrada	Bool	Indica se um item está sendo processado.
Estação (Erro)	Entrada	Bool	Indica se um item inválido foi detectado no compartimento de entrada.
CNC1 (Porta)	Entrada	Bool	Verdadeiro quando a porta está aberta.
CNC1 (Usinagem)	Entrada	Float	Progresso de usinagem (0-100).

4.1.3 Estação Classificação de Peças

Esta estação possui um sensor de visão responsável pelo reconhecimento das matérias-primas, tampas de produtos, bases de produtos e suas respectivas cores. Este sensor pode ser configurado para detectar mais de um tipo de peça, selecionando a configuração adequada:

Totalmente digital: retorna quatro entradas digitais indicando qual item foi detectado;

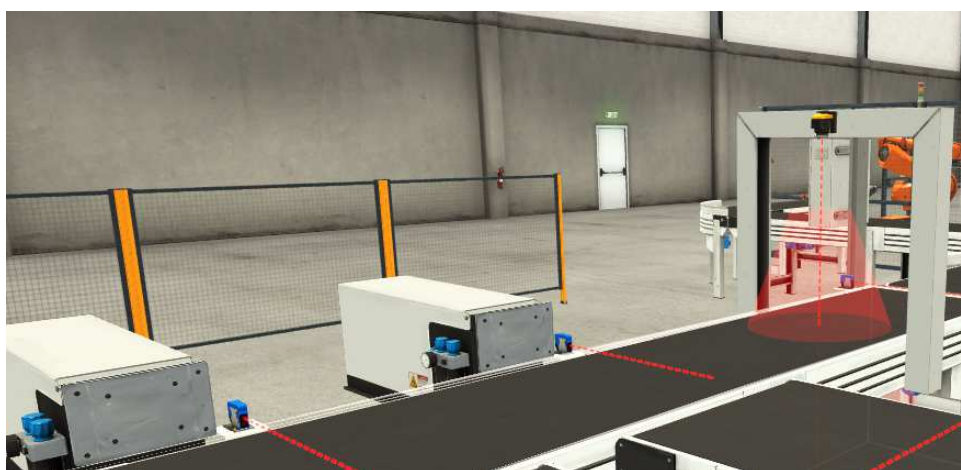
Tudo Numérico: retorna um valor que codifica o item detectado;

All ID: retorna um valor único (aleatório) que identifica o item detectado. Pode ser usado de forma semelhante aos leitores de código de barras ou RFID.

Tabela 8 – Codificação dos itens.

Item	Digital	Numérico	ID
Nenhum	0000	0	0
Material Azul	1000	1	ID
Tampa Azul	0100	2	ID
Base Azul	1100	3	ID
Material Verde	0010	4	ID
Tampa Verde	0110	5	ID
Base Verde	1110	6	ID

Figura 19 – Estação de Classificação de Peças.



Fonte: [Factoryio \(2020\)](#).

Outro elemento importante para esta estação é o classificador empurrador pneumático, equipado com dois sensores indicando os limites frontal e traseiro. Também inclui uma válvula servo que pode ser usada para definir e medir a posição da haste. O controle pode ser feito por valores digitais ou analógicos de acordo com a configuração selecionada.

Tabela 9 – Tags Empurrador pneumático.

Tags	E / S Controlador	Tipo	Descrição
Mover_Peça	Saída	Float	Mover para frente
Peça_Posição	Saída	Float	Mover para trás
Front Limit	Entrada	Bool	Limite frontal
Back Limit	Entrada	Bool	Limite traseiro

Objetivo da estação de classificação de peças é:

- Separar peças por cores e tipos;
- Posicionar a peça na esteira correta para a montagem do produto final.

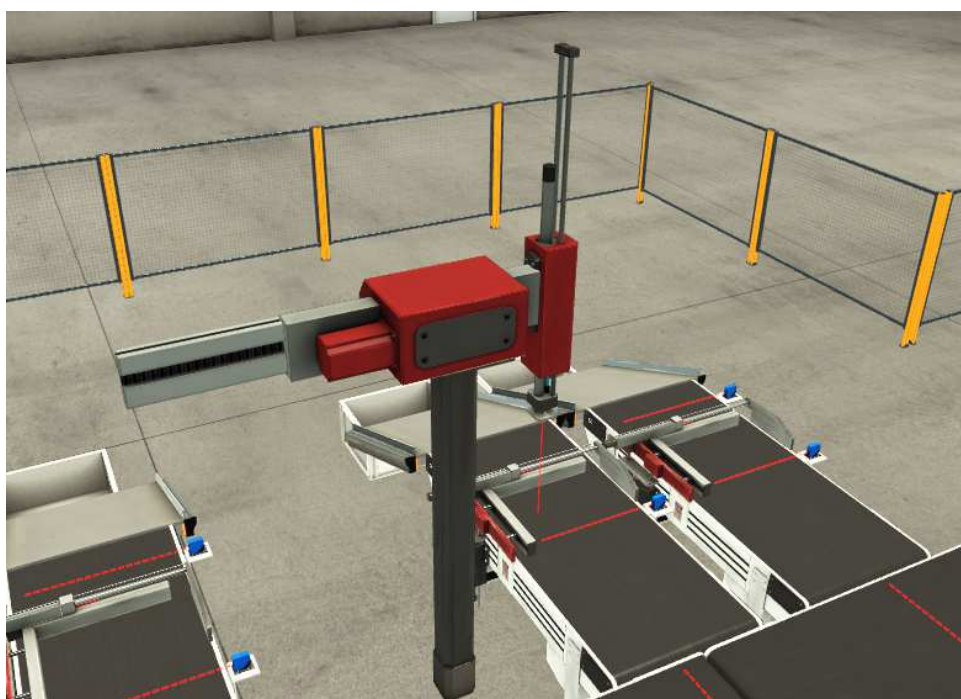
4.1.4 Estação de Montagem de Peças

Após a tampa ser posicionada pela estação de classificação, a esteira é acionada levando a tampa para o montador. Ao chegar na região de posicionamento um sensor difuso será cortado e as barras iram posicionar a tampa e a base do produto, logo depois o montador irá se movimentar pelo eixo z para segurar a tampa com a ventosa. Após esse movimento o montador retorna a posição inicial e faz um movimento pelo eixo x em direção a esteira que está posicionada a base do produto final.

Objetivo da estação de modelagem é:

- Posicionar a tampa e a base;
- Montar o produto final;
- Liberar o produto final.

Figura 20 – Estação Montagem de Peças.



Fonte: [Factoryio \(2020\)](#).

Tabela 10 – Tags Estação Montagem de Peças

Tag	E / S Controlador	Tipo	Descrição
Posição X	Entrada	Float	[0,10]V: posição atual ao longo do eixo X.
Posição Z	Entrada	Float	[0,10]V: posição atual ao longo do eixo Z.
Rotação	Entrada	Bool	Rotação do equipamento.
Sensor_ventosa	Entrada	Bool	Sensor ventosa giratória.
Sensor_item	Entrada	Bool	Sensor para detectar o item próximo a ventosa.
Sensor_Tampa	Entrada	Bool	Sensor para detectar a chegada da tampa
Sensor_Base	Entrada	Bool	Sensor para detectar a chegada da base
Ponto de ajuste X	Saída	Float	[0,10]V: definir a posição alvo ao longo do eixo X.
Ponto de ajuste Z	Saída	Float	[0,10]V: definir a posição alvo ao longo do eixo Z.
Giro horário	Saída	Bool	Rotação do equipamento sentido horário.
Giro anti-horário	Saída	Bool	Rotação do equipamento sentido anti-horário.
Giro ventosa horário	Saída	Bool	Rotação da ventosa sentido horário.
Giro ventosa anti-horário	Saída	Bool	Rotação da ventosa sentido anti-horário.
Ventosa	Saída	Bool	Ativar a ventosa.
Esteira_Tampa	Saída	Bool	Esteira que fornece tampas
Esteira_Base	Saída	Bool	Esteira que fornece bases
Esteira_Peça	Saída	Bool	Esteira com a peça montada
Barra_Tampa	Saída	Bool	Barra que posiciona a tampa
Barra_Base	Saída	Bool	Barra que posiciona a base

4.2 Visão Geral

Devido a dimensão do sistema a ser controlado, um processo rigorosamente organizado foi seguido para obter uma solução para o problema de sequência das estações.

O desenvolvimento do processo acompanhou o projeto da planta, uma vez que cada parte adicionada depende das anteriores, sendo de grande dificuldade a correção de erros nas etapas finais. Portanto, começou pela área de entrada de produtos, seguindo a área de usinagem, classificação, montagem e terminando na saída do produto final.

O sistema foi dividido em estações que possuem unidades menores definidas por cada um dos dispositivos controláveis (esteiras, pick and place, empurradores), de modo que cada estação corresponde a uma POU no CODESYS.

A operação do sistema completo é governada pelo painel de controle mostrado abaixo:

Figura 21 – Painel de Controle.



Fonte: [Factoryio \(2020\)](#).

O painel possui botoeiras para ligar, desligar e resetar a linha de produção, um botão de emergência que para todo o sistema, uma chave seletora para trocar o funcionamento do sistema de manual para automático e por fim displays que mostram a contagem de peças finalizadas.

4.3 Sistema de Supervisão

A sigla SCADA em inglês significa "Supervisory Control and Data Acquisition" e, em poucas palavras, é um sistema que visa facilitar a interação do usuário com o chão

de fábrica (máquina, equipamento, linha de produção, processo de fabricação, etc). A facilidade fica clara não só ao usuário que está realizando a interação, mas também ao desenvolvedor da aplicação.

Para explicar o que é facilitar a interação, dois aspectos do sistema de supervisão são apresentados a seguir:

- "como os dados do chão de fábrica chegam ao sistema supervisório"
- "como os dados do chão de fábrica são apresentados ao usuário"

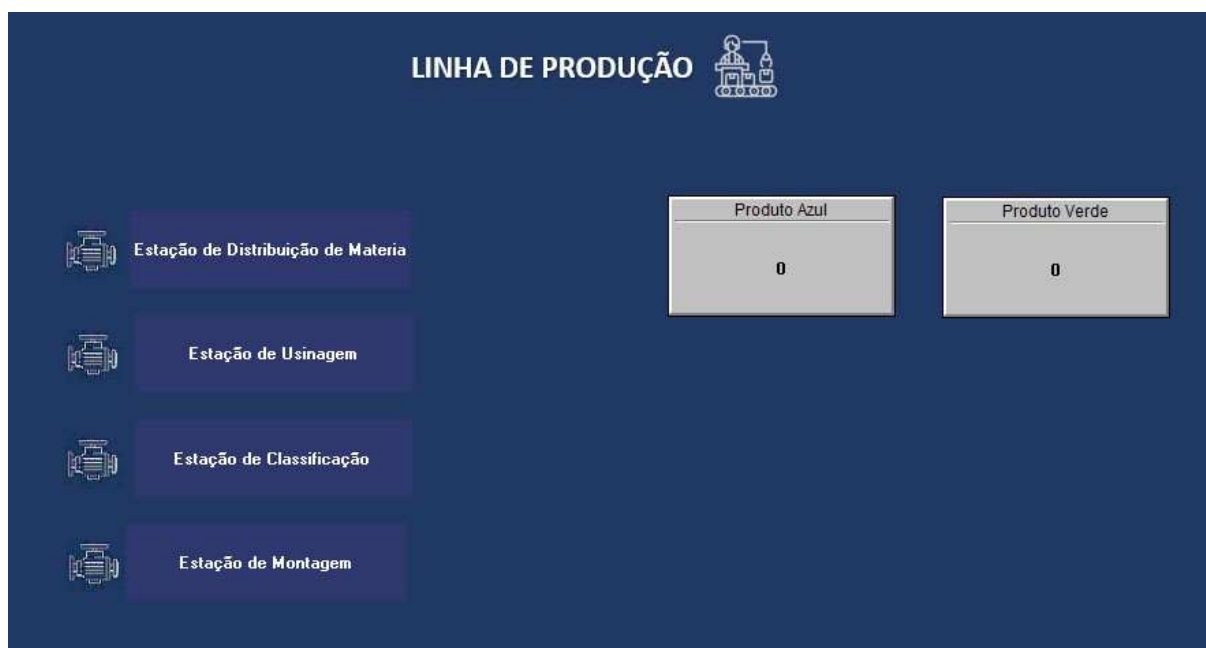
O primeiro aspecto pode inviabilizar (financeiramente) a utilização de um ambiente de desenvolvimento de software convencional (por exemplo: Visual Studio .NET, Eclipse, entre outros), embora a fonte de dado possa ser outra (banco de dados, por exemplo), os dados são normalmente coletados do processo através de um driver de comunicação (software).

Como o desenvolvimento de um driver de comunicação pode ser uma atividade que depende de conhecimento específico, tanto do protocolo de comunicação, como do ambiente ou linguagem a ser utilizada no desenvolvimento, o fato dos sistemas supervisórios já virem com uma solução de drivers de comunicação para facilitar acesso aos dados, torna-se um diferencial em relação aos ambientes de desenvolvimento de software convencionais.

O segundo aspecto também é um diferencial tanto do ponto de vista da ferramenta de desenvolvimento. como da aplicação que é gerada. Existem recursos comuns a muitos dos supervisórios (alarmes, relatórios, telas sinóticas, acesso a banco de dados, históricos, etc) que facilitam a criação da aplicação e também melhoram o resultado final.

Neste trabalho para desenvolver um sistema SCADA para a célula de manufatura foi utilizado o Elipse SCADA. Com ele foi desenvolvido uma tela para o monitoramento das estações e das contagem de produtos finais como é mostrado na figura 22.

Figura 22 – Sistema SCADA.



Fonte: Autor.

A supervisão funciona da seguinte forma:

- Ao iniciar o processo de cada estação será indicado em verde a estação em funcionamento.
- Os dois contadores iram contabilizar a quantidade de peças azuis e verdes finalizadas.
- A comunicação dos sensores e atuadores da célula para o sistema supervisorio é feita via OPC.

5 Conclusões

Após a conclusão deste Projeto de Graduação Final, as seguintes conclusões são tiradas:

- A presença de atrasos na sincronização das informações entre CODESYS e FACTORY I / O constituiu um dos principais desafios ao nível do design. Inevitavelmente, há uma série de atrasos que afetam a comunicação entre os dois programas, por isso devem ser levados em consideração na hora de realizar a elaboração da planta. Para resolver esses problemas, foi escolhido colocar os sensores um pouco à frente de seu lugar, simulando o atraso para alcançar a operação adequada.
- Em relação ao FACTORY I / O, pode-se notar que o número máximo de itens que são recuperados do servidor é limitado, composto por 128 itens.

As conclusões anteriores, a priori, poderiam ser entendidas como algo negativo, embora na realidade ajudou a desenvolver capacidades em termos de resolução de problemas. Tem sido gratificante encontrar alternativas que diluam esses problemas e, de uma maneira geral, posso destacar os aspectos positivos:

- Trabalhar com CODESYS e, portanto, aumentar o leque de conhecimento a respeito, tem sido uma tarefa enriquecedora, visto que a maioria dos fabricantes de PLC's já optaram por este programa, havendo, portanto, uma infinidade de CLP's que estão programados por meio do CODESYS.
- O estudo da comunicação via OPC tem sido algo a destacar. É uma alternativa que reduz custos e que, a nível acadêmico, apresenta muitas vantagens.
- O fato de solucionar os contra tempos que vem surgindo ao longo desses meses tem sido bastante satisfatório. Eu melhorei em termos de depuração de código e tenho sido capaz de aprender algo com cada um dos problemas que foram apresentados a mim.
- Focando nas aulas apresentadas, posso comentar que ganhei habilidade na programação e na construção de cenas de forma didática.

A programação das demais cenas que o Factory I/O oferece fica como sugestão para trabalhos futuros além disso, é possível a integração de mais estações na célula de manufatura construída neste trabalho, um exemplo disso é a estação de armazenamento que poderia ficar logo após a estação de montagem.

Referências Bibliográficas

- Factoryio. *Documentação*. 2020. Disponível em: <<https://factoryio.com/>>. Citado 16 vezes nas páginas 10, 11, 12, 13, 15, 16, 17, 19, 21, 22, 23, 24, 25, 27, 29 e 31.
- GE. *System 1 - Visão geral do sistema: Apostila de treinamento prático*. 2008. Citado na página 8.
- GROOVER, M. *Automação Industrial e Sistemas de Manufatura*. [S.l.]: Editora Pearson, 2010. ISBN 9788576058717. Citado 2 vezes nas páginas 6 e 7.
- MADEIRA, D. *IIoT - Industrial Internet of Things*. 2016. Disponível em: <<https://www.embarcados.com.br/iiot-industrial-internet-of-things/>>. Citado na página 9.
- MARTINS, P. G. *Administração da Produção*. [S.l.]: Saraiva, 1999. Citado na página 7.
- MOREIRA, P. M. C. M. Implantação de mudanças organizacionais: Do arranjo funcional à manufatura celular e fms. *Saber Eletrônica*, S.d. Citado na página 7.
- PARROT, A. *Industry 4.0 and the digital twin*. *Deloitte University Press*. 2017. Citado 2 vezes nas páginas 8 e 9.
- ROJAS, J. H. C. *Metodologia de modelagem, simulação e programação off-line de robos e mecanismos mecatronicos integrados e direcionados a celulas de manufatura flexiveis*. 2004. Disponível em: <<http://www.bibliotecadigital.unicamp.br/document/?code=vtls000373840&opt=3>>. Citado na página 8.
- ROSÁRIO, J. M. *Automação Industrial*. [S.l.]: Baraúna SE Ltda, 2009. ISBN 978-85-7923-000-4. Citado na página 4.
- SARAMAGO, M. A. P. *Integração de dispositivos inteligentes utilizando conceitos de domotica direcionados a automação hospitalar*. 2002. Disponível em: <<http://www.bibliotecadigital.unicamp.br/document/?code=vtls000271587&opt=4>>. Citado na página 5.
- SCHLEICH, B. *Cyclone II Device Handbook Shaping the digital twin for design and production engineering*. *CIRP Annals-Manufacturing Technology*. 2017. Citado na página 8.

ANEXO A – Guia do Laboratório Virtual

A.1 Apresentação do Factory I / O

Factory I / O é uma simulação de fábrica 3D para aprender tecnologias de automação. Projetado para ser fácil de usar, permite construir rapidamente uma fábrica virtual usando uma seleção de peças industriais comuns. Inclui muitas cenas inspiradas em aplicações industriais típicas, desde níveis iniciantes a avançados.

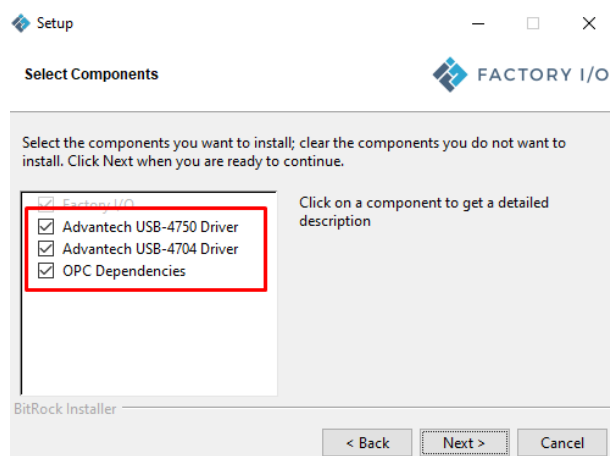
O cenário mais comum é usar o Factory I / O como uma plataforma de treinamento de CLP, uma vez que o CLP são os controladores mais comuns encontrados em aplicações industriais. No entanto, também pode ser usado com microcontroladores, SoftPLC, Modbus, entre muitas outras tecnologias.

A.1.1 Instalação

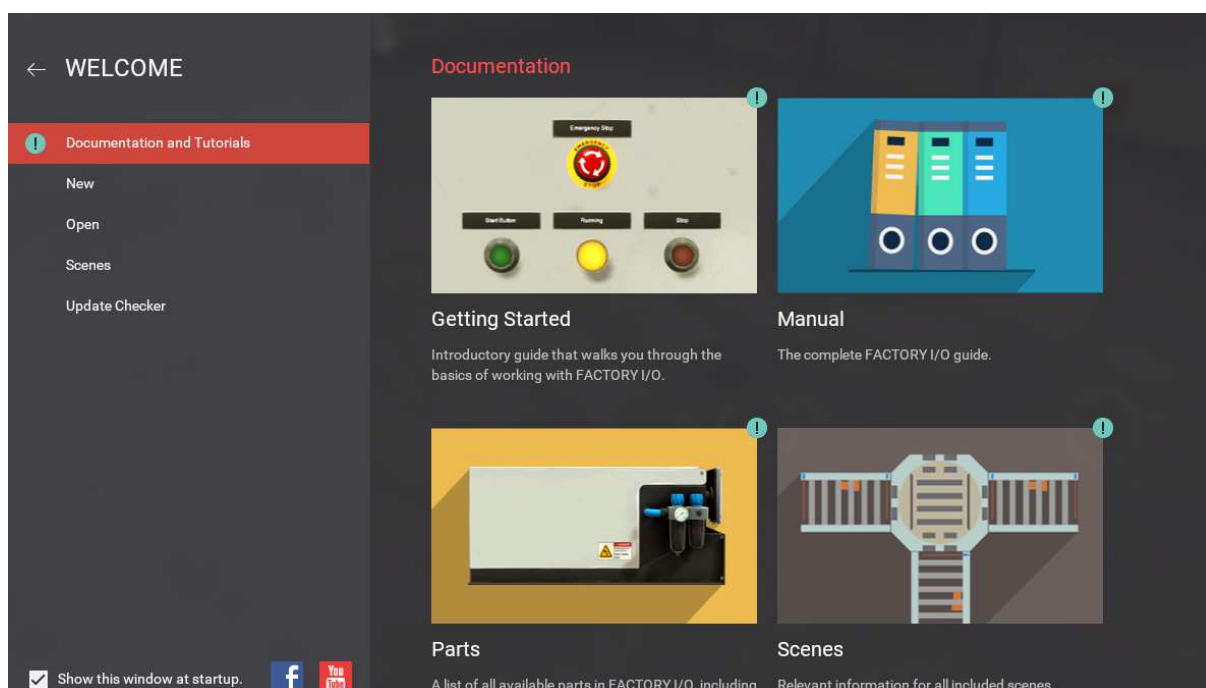
1. Execute o instalador. Se solicitado pelo Controle de Conta de Usuário, clique em Sim . Siga as instruções do instalador.



2. Na janela de componentes, deixe os drivers Advantech marcados. Deixe a opção OPC Core Components marcada se você for usar Factory I / O com o driver OPC Client Data Access.



3. Agora é a hora de abrir o Factory I / O. Se nenhuma licença for encontrada, vá a pasta do instalador e copie o arquivo **Assembly-CSharp**, depois cole na pasta **Managed**, ela pode ser encontrada seguindo o caminho: **Arquivos de Programas > Real Games > Factory IO > Factory IO_Data > Managed**



A.1.2 Navegando

Uma das habilidades mais importantes a serem aprendidas no Factory I / O é como usar as câmeras. As câmeras são usadas para navegar no espaço 3D e são a chave para interagir com as peças ou criar novas cenas. Você pode usar três tipos de câmeras: Orbit , Fly e First Person.

Acostume-se a cada tipo testando as ações conforme descrito abaixo. Lembre-se de que cada câmera foi projetada para uma finalidade funcional específica.

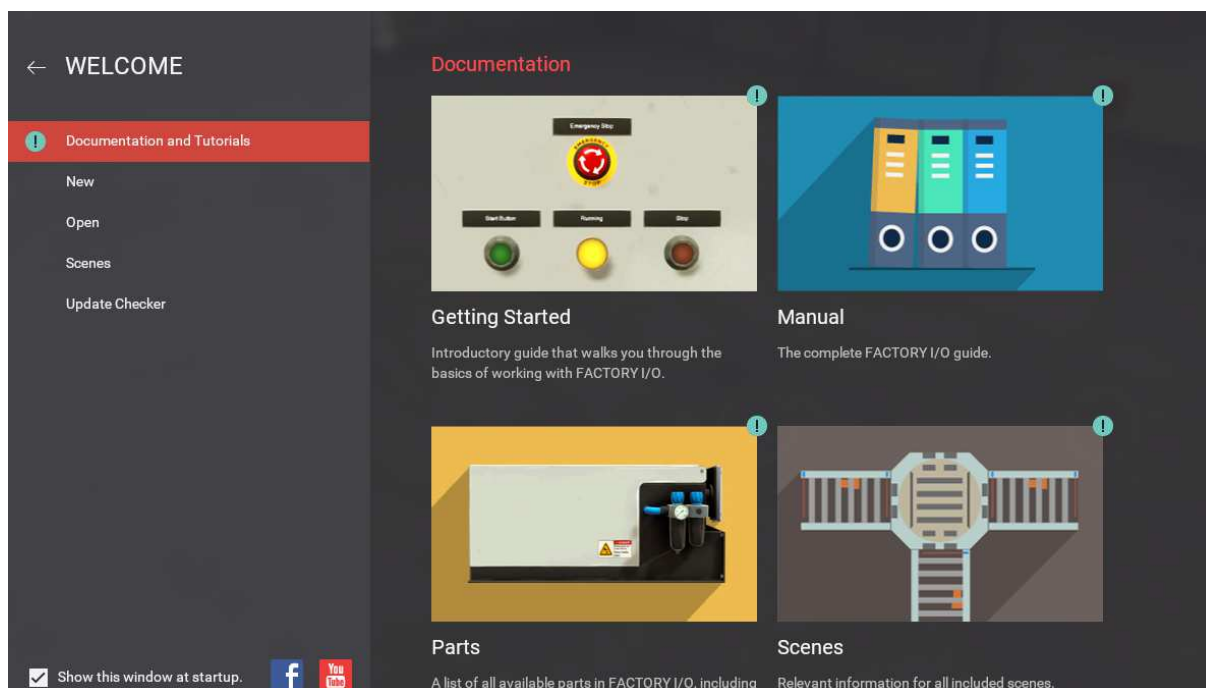


Figura 23 – Câmeras

A.1.2.1 Câmera Orbit

A câmera Orbit é a câmera padrão e deve ser usada na construção de uma cena. Esta é a única câmera que permite que você se mova pelas partes sem colidir com elas.

Ele funciona girando em torno de um ponto de interesse (indicado por um ponto branco) que você define clicando duas vezes em uma peça. Depois que o ponto de interesse é definido, você gira a câmera ao redor, segurando o botão direito do mouse e arrastando o mouse.

A.1.2.2 Câmera Fly

A câmera Fly é usada para se mover livremente no espaço 3D. Esta câmera colide com partes da cena, mas não é detectada pelos sensores.

A.1.2.3 Câmera First Person

A câmera First Person representa uma pessoa com 1,8 m de altura. Deve ser usado ao simular uma pessoa em uma fábrica virtual. Ele colide com partes da cena e, por padrão, não é detectado pelos sensores.

A.1.3 Cenas

O factory I / O vem com 21 cenas , que podem ser acessadas na guia Cenas . Você também pode usar qualquer uma dessas cenas como ponto de partida para uma nova, abrindo e salvando-a com um nome personalizado.

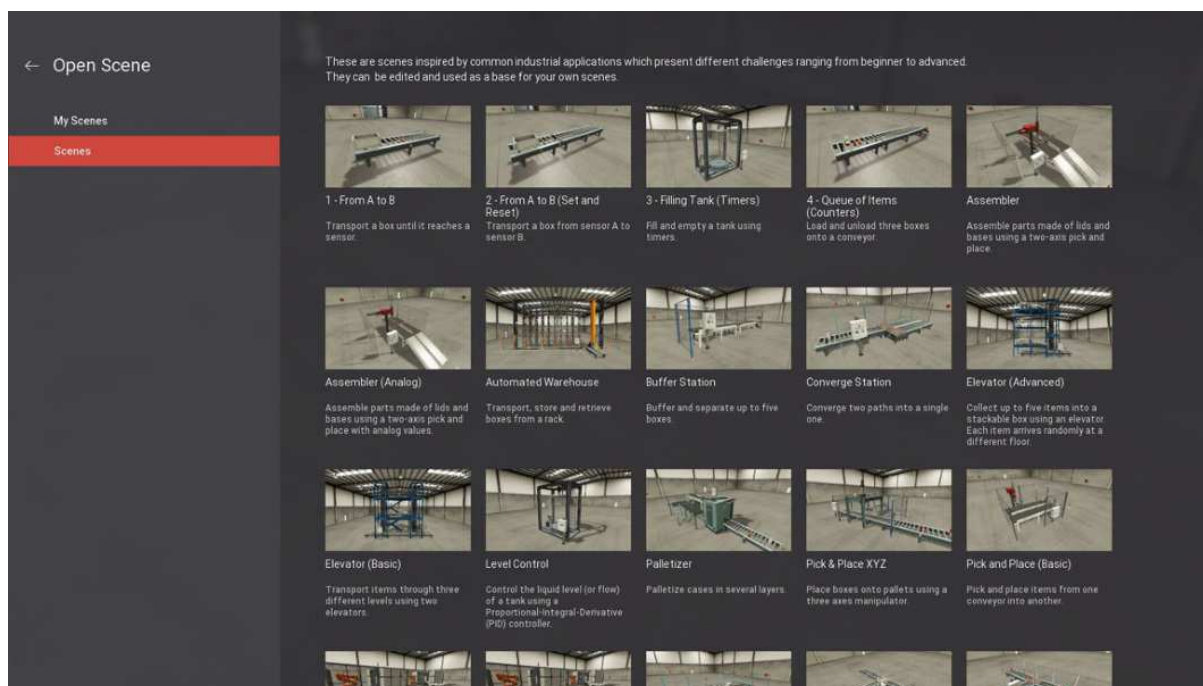


Figura 24 – Cenas

A.1.4 Peças

O factory I / O inclui uma grande variedade de peças inspiradas nos equipamentos industriais mais comuns. Você pode criar uma fábrica virtual colocando e organizando essas peças juntas.

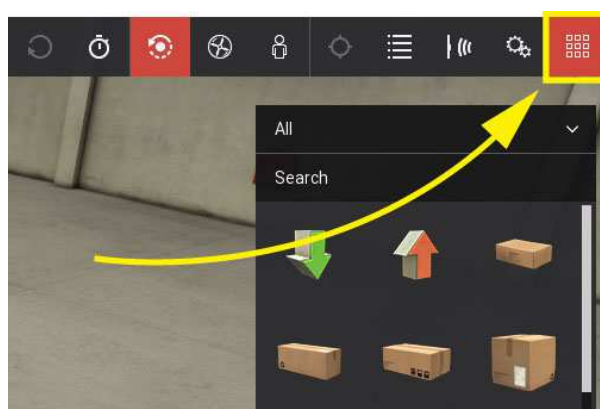


Figura 25 – Cenas

A.1.5 Tags

Antes de controlar uma cena com um controlador externo (um CLP, por exemplo), é recomendável testá-la manualmente. Dessa forma, você pode garantir que o layout da cena funcione conforme o esperado. Mas primeiro, você precisa aprender o que são Tags e como elas podem ser usadas para controlar partes.

Qualquer peça que seja um sensor ou atuador possui pelo menos uma etiqueta. As tags são compostas por um nome e um valor e podem ser de dois tipos diferentes: Tags de sensores e Tags de atuadores . Eles podem conter três tipos de dados diferentes: Booleano para valores on / off, Float para valores analógicos (números reais) e Inteiro para dados específicos.

Os valores das tags podem ser Forçados a qualquer momento, permitindo que você faça o papel do controlador. Para controlar sua cena manualmente.

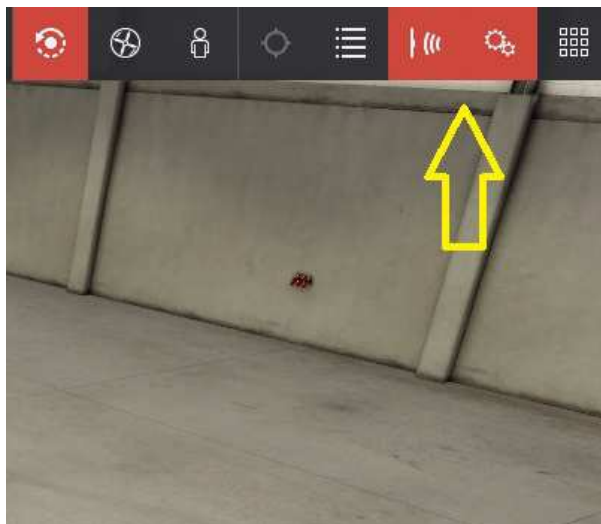


Figura 26 – Tags

A.2 Projeto no CODESYS

CODESYS é um aplicativo para a programação de projetos de controle e automação. As etapas para o desenvolvimento de um projeto neste aplicativo são as seguintes:

A.2.1 Instalação

1. Todas as versões dos CODESYS estão disponíveis no seguinte link: <http://applications.eaton.eu/sdlc/?lx=11>
2. Para baixar a versão Demo clique em:
Software > XSOFT-CODESYS > 3.5.11 > XSOFT-CODESYS V3.5.11
3. Feito o download, execute o instalador e siga as instruções.

A.2.2 Criação de um novo projeto

1. Depois da instalação do aplicativo abra o CODESYS por meio do menu iniciar (XSOFT-CODESYS V3.5.11).

2. Crie um novo projeto em New Project... Ctrl+N. Na janela que abrir, conforme mostra a figura 5, selecione a opção Standard project, defina o nome do projeto (Name), o diretório para armazená-lo (Location) e selecione OK.
3. Uma nova janela será aberta (fig 6): selecione o dispositivo CODESYS Control Win V3 (este dispositivo permite que o aplicativo execute o projeto em um ambiente virtual, sem a necessidade de definir uma configuração real de hardware de CLP), e a linguagem do programa PLC PRG. No nosso caso, será o Ladder. Por default, o aplicativo cria uma árvore hierárquica (fig.7) com uma tarefa principal, denominada PLC PRG, que é uma POU do tipo Program, cujo nome poderá ser alterado posteriormente.
4. Salve seu projeto, e não se esqueça de salvá-lo ao final de cada uma das próximas etapas.

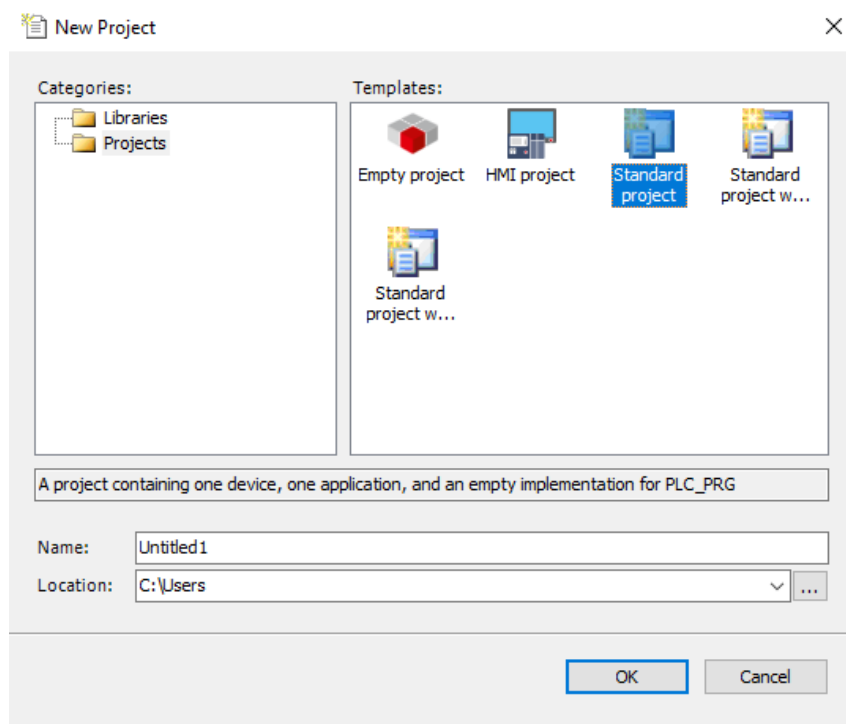


Figura 27 – Novo Projeto

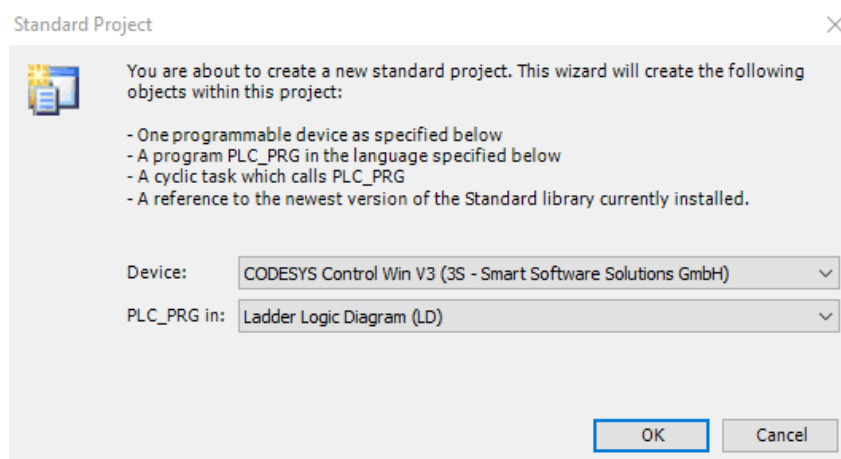


Figura 28 – Janela para seleção do dispositivo

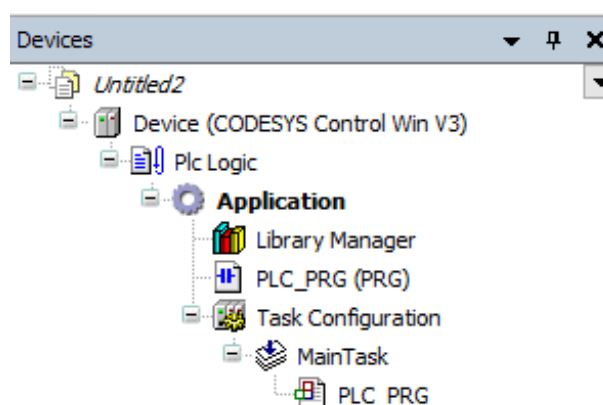


Figura 29 – Árvore hierárquica inicial da configuração do projeto

A.2.3 Criação de POU's

Você pode criar quantas POU's forem necessárias para o seu projeto. O termo POU em português significa "Unidades de Organização de Programa". No Codesys a primeira aba à esquerda é das POU's, todos os programas de um projeto ficam nesta aba, um projeto sempre deve ter um programa chamado PLC_PRG. Para criar uma POU, pressione o botão direito do mouse em Application e utilize o caminho Add Object → POU. Defina o nome da POU, o seu tipo (Program, Function Block ou Function) e a linguagem de implementação. Lembre-se que se a POU for do tipo Function é preciso definir também o tipo de valor de retorno da função.

A.2.4 Edição de POU

Para editar um POU basta clicar em seu nome que o editor da linguagem selecionada na criação da POU será aberta. Independente da linguagem selecionada, a janela do editor é dividida em duas partes. A parte superior é destinada à declaração das variáveis locais, e a parte inferior ao algoritmo da POU, como mostrado na figura 8.

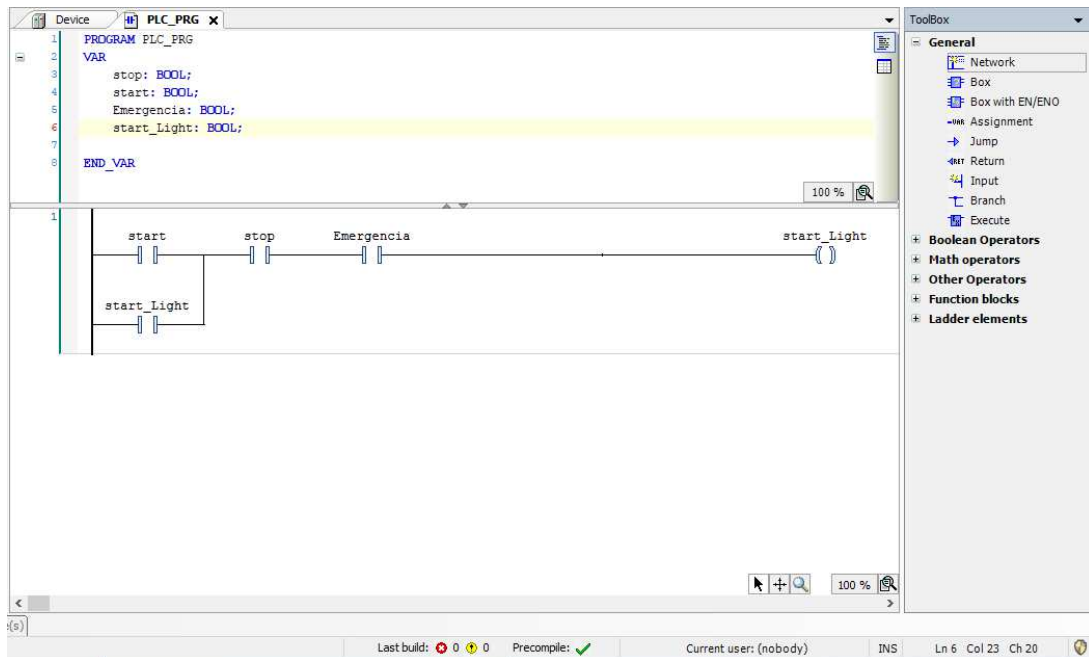


Figura 30 – Janela para edição de POU

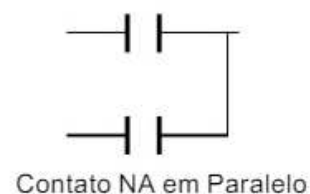
A.2.5 Compilação do projeto

Compile seu projeto antes de testá-lo. Para isso siga o caminho Build → Build F11. Analise os erros e as mensagens que forem exibidas, e corrija o projeto caso seja necessário.

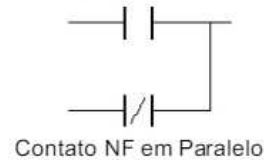
A.2.6 Instruções mais utilizadas

A.2.6.1 Sinal de Entrada (Contato)

- **Relé NA:** tais dispositivos fecham os seus contatos quando são energizados. A denominação NA, normalmente aberto, se refere à consideração do que os relés eletromecânicos, deste tipo, estariam abertos se estivessem fora do circuito.

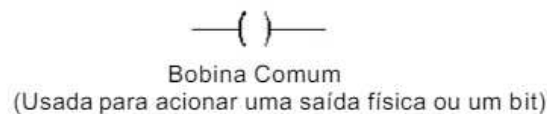


- **Relé NF:** tais dispositivos abrem os seus contatos quando são energizados (Fig.4.3). A denominação NF, normalmente fechado, se refere à consideração de que os relés eletromecânicos, deste tipo, estariam fechados se estivessem fora do circuito.



A.2.6.2 Sinal de Saída Comum (Bobina)

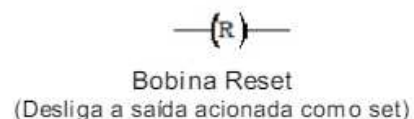
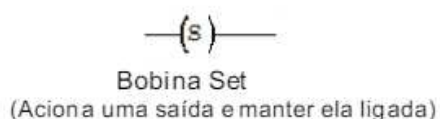
Esta instrução serve para habilitar uma saída, que tanto pode corresponder a uma saída física do CLP, como a um bit. A saída será habilitada quando a condição, onde está localizada a instrução se tornar verdadeira. Quando a condição falsa, a saída será desabilitada.



NOTA: Há situações em que é necessário manter uma saída energizada, mesmo quando a entrada venha a ser desligada. O circuito utilizado para essa finalidade é chamado de “selo” ou trava. Os selos são combinações entre elementos destinados a manter uma saída ligada (Retentiva), quando se utilizam botoeiras de pressão (ou de contato momentâneo).

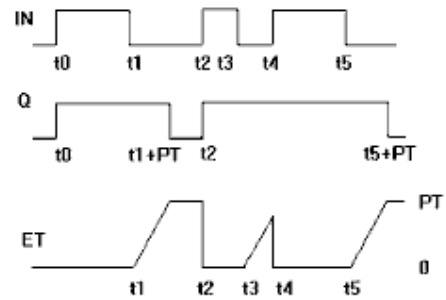
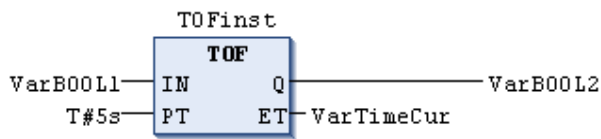
A.2.7 Sinal de Saída Retentivo (Set / Reset)

Uma maneira de fazer a auto retenção de uma saída (bobina) é pela instrução set. A instrução set liga uma saída e a mantém ligada mesmo que o contato da entrada deixe de conduzir. Para desligar a saída é utilizada a instrução reset.



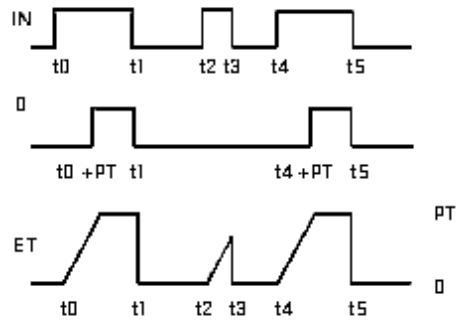
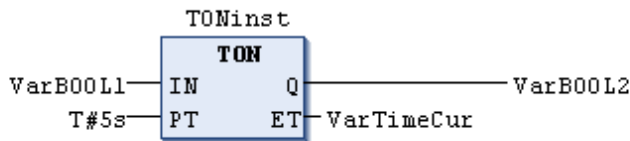
A.2.7.1 Temporizadores

- **TOF(IN, PT, Q, ET):**



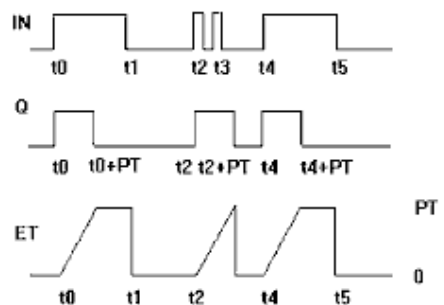
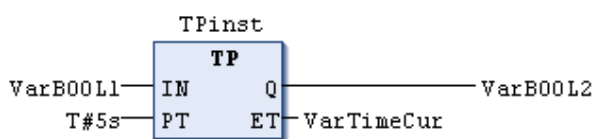
O temporizador começa quando a entrada IN muda para FALSE (logica 0). Quando o tempo que foi definido pela entrada PT (Preset Time) expira, a saída Q do temporizador muda de 1 para 0. A saída ET é o tempo decorrido uma vez que o temporizador foi iniciado.

- TON(IN, PT, Q, ET):



O temporizador começa quando a entrada IN muda para TRUE (logica 1). Quando o tempo que foi definido pela entrada PT(Preset Time) expira, a saída Q do temporizador muda de 0 para 1. A saída ET é o tempo decorrido uma vez que o temporizador foi iniciado.

- TP(IN, PT, Q, ET):

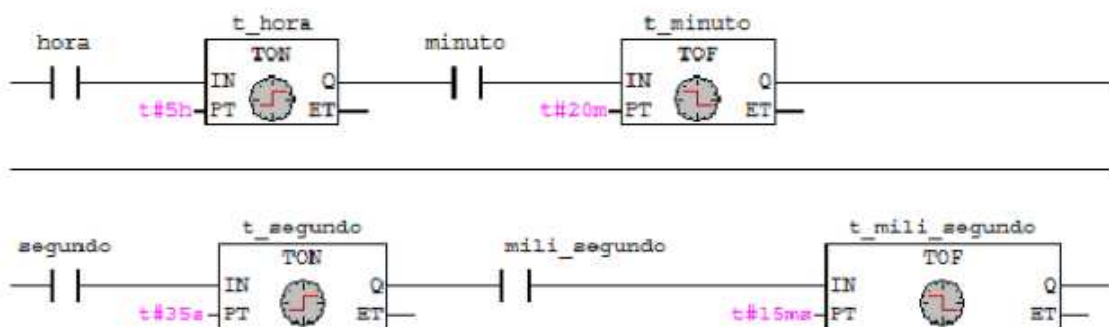


O temporizador começa quando a entrada IN muda para TRUE (lógica 1). A saída Q também será TRUE (lógica 1) pelo período de tempo que foi definido pela entrada PT (Preset Time). A saída Q irá permanecer acionada pelo período de tempo especificado, mesmo que a entrada IN for derrubada (O valor assumido será FALSE = lógico 0). A saída ET é o tempo decorrido para o temporizador uma vez que o tempo foi iniciado. Q será verdadeiro quando IN mudar para TRUE e ET deverá ser menor que o tempo PT, Caso contrário será FALSE.

Resumindo: O temporizador TP será TRUE (lógica 1) assim que for iniciado, mas somente por um pulso de duração que foi especificado no PT.

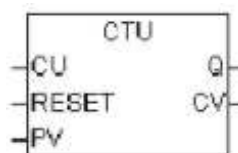
A declaração de tempo pode consistir de dias (indicado por "d"), horas ("h"), minutos ("m"), segundos ("s") e milissegundos ("ms"). Nota: Aqui os dados de tempo devem ser indicados em ordem de magnitude (d antes de h, h antes de m, m antes de s e s antes de ms).

- **Exemplo de declaração de tempo:**



A.2.7.2 Contadores

- **CTU**



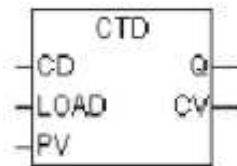
O bloco CTU é um contador incremental (Crescente).

- **RESET:** Se o parâmetro de entrada RESET = VERDADEIRO (Lógica 1), o contador é inicializado com o valor de contagem em 0 e então o tempo é executado para cima.

- **PV (Valor Pré-Definido):** PV é um valor de 16 bits inteiro, ou seja, entre -32768 e +32767. Este valor é utilizado para definir o limite superior para o contador, na qual Q muda para VERDADEIRO.
- **CU:** CU (Contagem Crescente) é o pulso de entrada da contagem, ou seja, a contagem é aumentada de 1 em 1 quando ha uma mudança de falso para verdadeiro (contanto que 0 não é atingido).
- **CV:** CV (Valor Atual) é o valor atual (contagem) para o contador. Esta contagem é aumentada de 1 em 1 quando ha um pulso positivo que está presente no valor CU para a entrada.
- **Q:** Q é VERDADEIRO (Lógica 1), quando o valor atual (contagem) CV é o mesmo que PV.

As variáveis CU, RESET e Q são do tipo BOOL, enquanto o PV e CV são do tipo INT.

- **CTD**

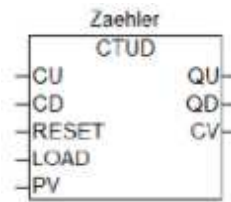


O bloco CTD é um contador decremental (decrecente).

- **LOAD:** Se o parâmetro LOAD = VERDADEIRO (Lógica 1), o contador é inicializado com o valor de PV e a contagem é então executada para baixo.
- **PV (Valor Pré-Definido):** PV é um valor de 16 bits inteiro, ou seja, entre -32768 e +32767. O contador é inicializado com este valor, quando a entrada LOAD = VERDADEIRO,
- **CD:** CD (Contagem Decrescente) é o pulso de entrada da contagem, ou seja, a contagem de CV é diminuída de 1 em 1 quando há uma mudança de falso para verdadeiro (contanto que 0 não é atingido).
- **CV:** CV (Valor Atual) é o valor atual (contagem) para o contador. Esta contagem é diminuída de 1 em 1 quando ha um pulso positivo que está presente no valor CD de entrada.
- **Q:** Q é VERDADEIRO (Lógica 1), quando o valor atual (contagem) CV é 0.

As variáveis CD, LOAD e Q são do tipo BOOL, enquanto o PV e CV são do tipo INT.

- CTUD



O bloco CTUD é um contador incremental e decremental. Este contador é uma combinação de um CTU (Crescente) e um contador CTD (Decrescente).

- **RESET:** Se o parâmetro de entrada RESET = VERDADEIRO (Lógica 1), o contador é inicializado com o valor de contagem em 0 e então o tempo é executado para cima ou para baixo.
- **LOAD:** Se o parâmetro LOAD = VERDADEIRO (Lógica 1), o contador é inicializado com o valor de PV e a contagem é então a contagem é executada para cima ou para baixo.
- **PV (Valor Pré-Definido):** PV é um valor de 16 bits inteiro, ou seja, entre -32768 e +32767 (somente um valor maior que 0 é realmente significativo). O contador é inicializado com este valor, quando a entrada LOAD = VERDADEIRO.
- **CD:** CD (Contagem Decrescente) é o pulso de entrada da contagem, ou seja, a contagem de CV é diminuída de 1 em 1 quando há uma mudança de falso para verdadeiro (contanto que 0 não é atingido).
- **CU:** CU (Contagem Crescente) é o pulso de entrada da contagem, ou seja, a contagem é aumentada de 1 em 1 quando ha uma mudança de falso para verdadeiro (contanto que 0 não é atingido).
- **CV:** CV (Valor Atual) é o valor atual (contagem) para o contador. Esta contagem é aumentada de 1 em 1 quando ha um pulso positivo que está presente no valor CD para a entrada diminuída de 1 em 1 quando há um pulso positivo que esta presente no valor CU para a entrada
- **QD:** QD é à saída binária inferior do contador. Esta saída é TRUE (Lógica 1) quando o valor atual CV = 0.
- **QU:** QU é à saída binária superior do contador. Esta saída é TRUE (Lógica 1) quando o valor atual CV \geq PV (valor pré-definido).

As variáveis CU, CD, RESET, LOAD, QU e QD são do tipo BOOL, enquanto o PV e CV são do tipo INT.

A.3 Acionamento da esteira transportadora

Nesta seção será apresentada todas as configurações necessária para simular e programar uma esteira transportadora

A.3.1 Configurando o factory I / O

Clique em scenes na página inicial do aplicativo e escolha a cena 1-From A to B. Nela terá a simulação de uma esteira que transporta uma caixa, um sensor óptico na extremidade da esteira e um motor.

Como apresentado na figura 9, adicione as seguintes peças na cena:

- Column
- Electric Switchboard
- Start Button
- Stop Button
- Emergency Stop



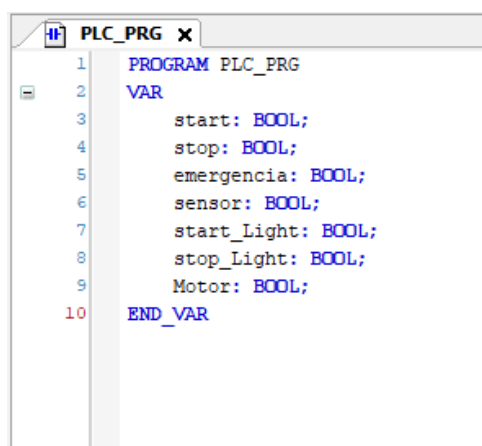
Figura 31 – Painel de controle

A.3.2 Configuração da comunicação OPC

Antes de fazer a programação em Ladder do processo, é necessário estabelecer a comunicação de todas as variáveis encontradas na cena do factory I / O para o codesys seguindo os seguintes passos:

A.3.2.1 Codesys

1. Criar um projeto no codesys como descrito no item 2.1.
2. Clicar em PLC_PRG(PRG) e declarar todos os sensores e atuadores da cena do factory I / O (fig 10).



```
1 PROGRAM PLC_PRG
2 VAR
3     start: BOOL;
4     stop: BOOL;
5     emergencia: BOOL;
6     sensor: BOOL;
7     start_Light: BOOL;
8     stop_Light: BOOL;
9     Motor: BOOL;
10 END_VAR
```

Figura 32 – Variáveis

3. Adicionar o suporte OPC para a comunicação, para isso você irá clicar com o botão direito do mouse em Application → Add Object → symbol configuration → Support OPC UA Features → Add (fig 11)

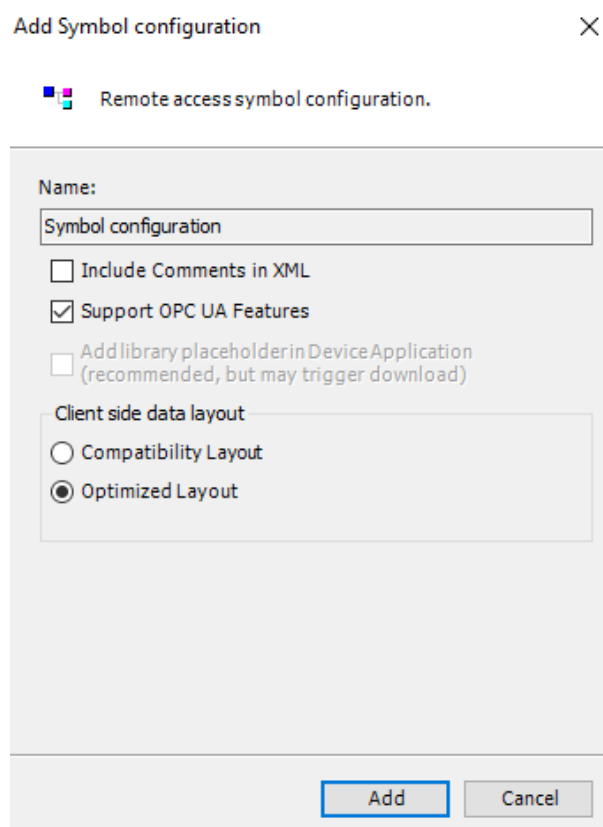


Figura 33 – Comunicação via OPC

4. Selecionar todas as variáveis do PLC_PRG no symbol configuration. Em seguida clique em Build.
5. Clique com o botão direito do mouse no ícone CODESYS Control Win PLC (Systray) e selecione Iniciar PLC .

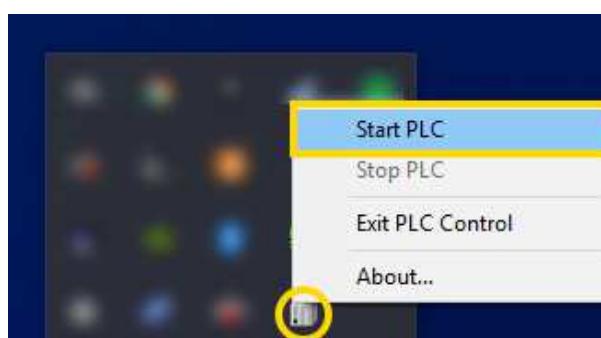


Figura 34 – PLC

6. Para configurar o servidor OPC, inicie o aplicativo OPC Configurator, localizado no menu iniciar, e altere a taxa de atualização para 10 ms.

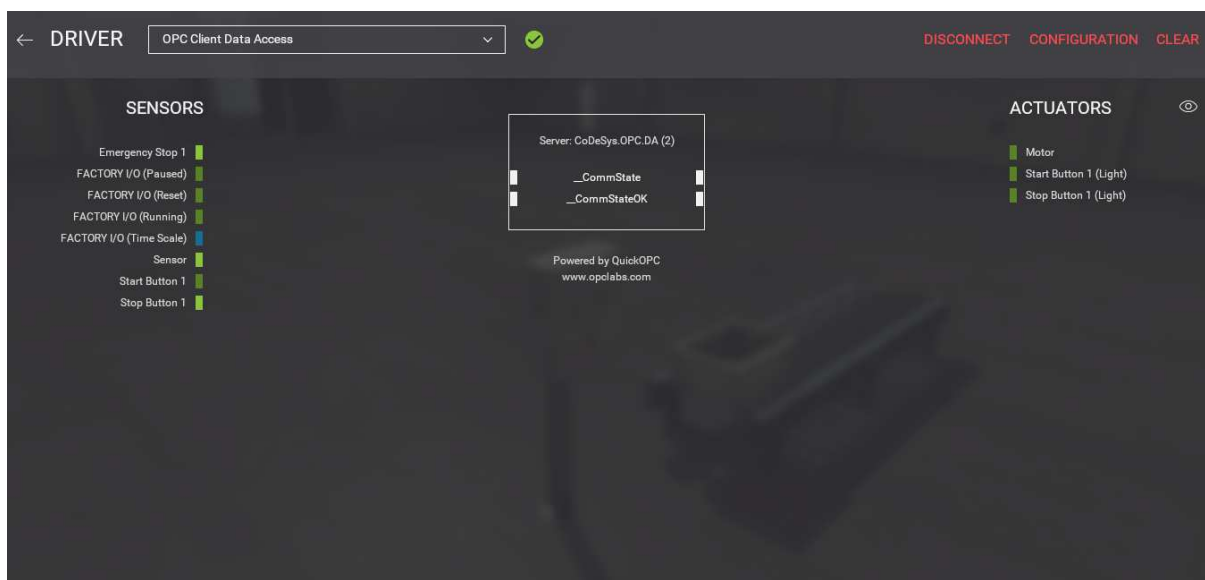
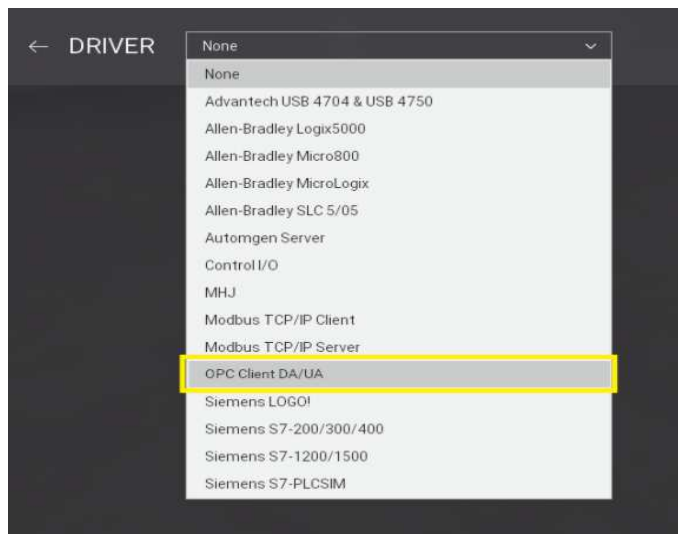


Figura 35 – PLC

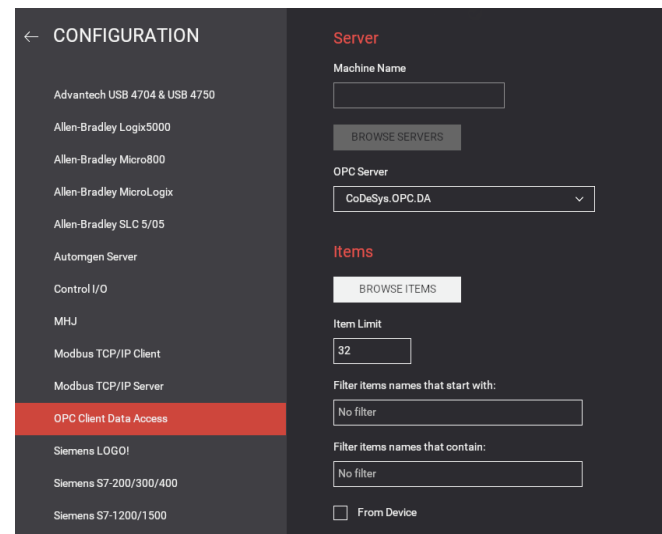
7. Por fim, clique em login (Alt+F8) no ambiente do codesys.

A.3.2.2 Factory I / O

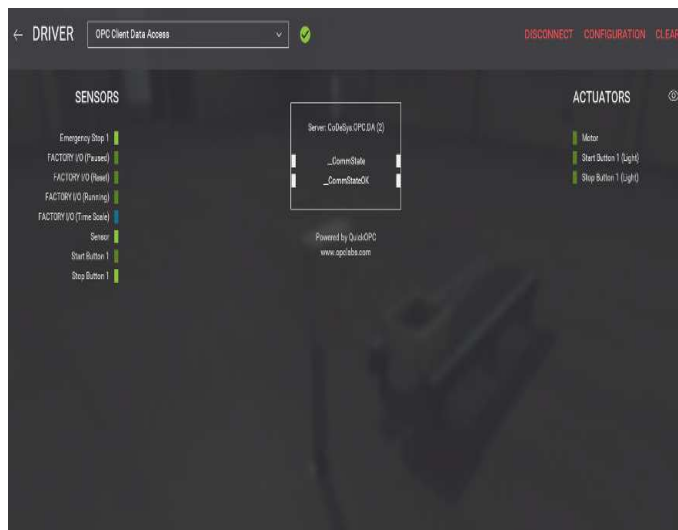
1. Abra a cena e clique em File→Driver.
2. Selecione OPC Client DA na lista Driver e clique em CONFIGURATION.
3. Selecione DA: CoDeSys.OPC.DA na lista OPC Server. Em seguida, clique em BROWSE.
4. Mapeie as etiquetas dos sensores e atuadores, conforme mostrado na imagem d.
5. Por fim, feche a janela Driver e inicie a simulação (pressione F5).



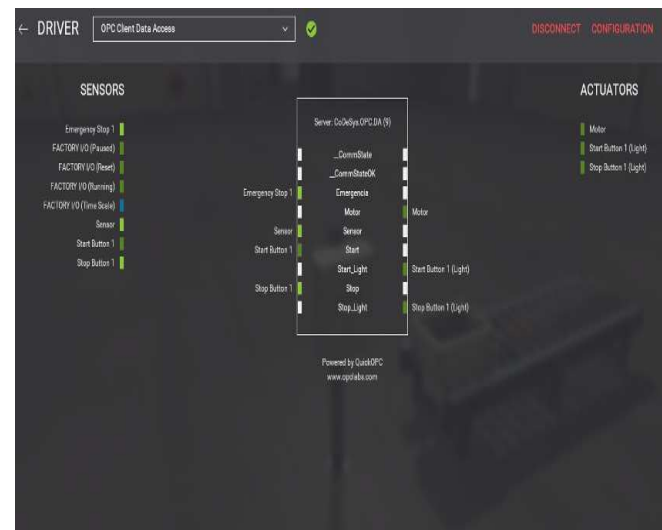
(a) Driver



(b) Configuração



(c) Ambiente



(d) Etiquetar

A.3.3 Programação no Codesys

Realizada toda a parte de comunicação entre o codesys e o factory I / O, podemos agora fazer a programação em ladder no ambiente do codesys para colocar em funcionamento a simulação da esteira transportadora.

temos os seguintes contatos disponíveis na parte superior do codesys para a programação em ladder:

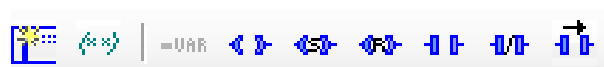


Figura 36 – Contatos

Como um diagrama de comandos elétricos, iremos acionar com programação em

ladder os botões start, stop e emergência do painel de controle presente na simulação, da seguinte forma:

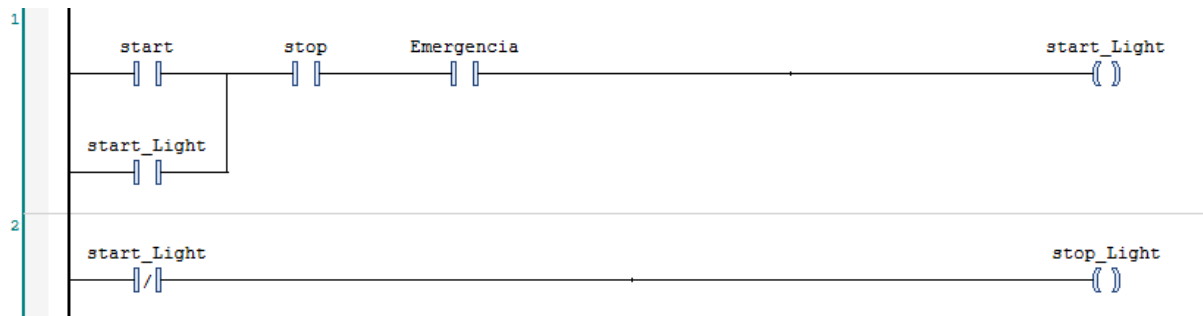


Figura 37 – Acionamento dos botões

O próximo passo será o acionamento do motor e do sensor da esteira.

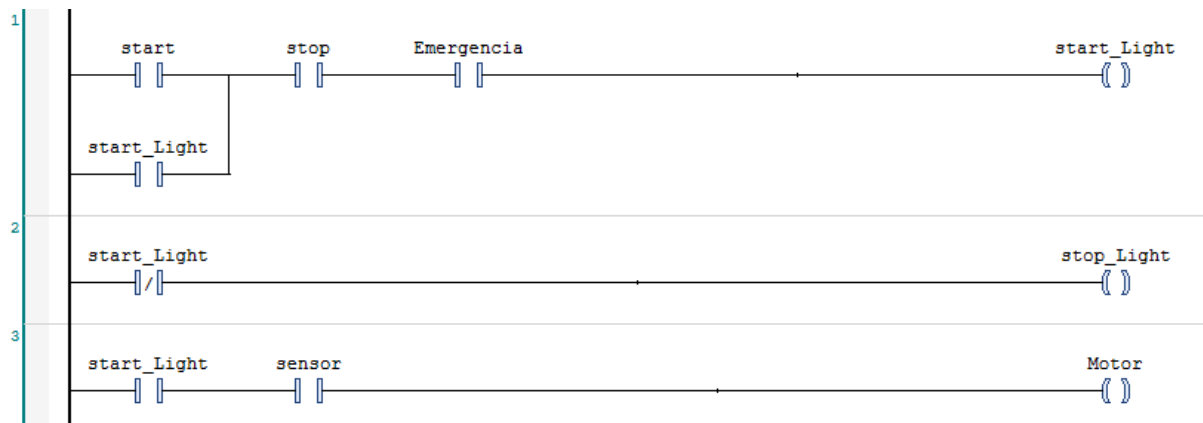


Figura 38 – Acionamento do motor

Por fim, Compile seu projeto clicando em Build (F11) e em seguida clique em login.