UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Instituto de Ciência e Tecnologia
Câmpus de Sorocaba

KARINNE SOUZA RIBEIRO

Virtualização de uma estação de trabalho de um sistema de manufatura automatizado

Sorocaba
2022

KARINNE SOUZA RIBEIRO

Virtualização de uma estação de trabalho de um sistema de manufatura automatizado

Projeto Final de Conclusão de Curso apresentado ao Instituto de Ciência e Tecnologia de Sorocaba, Universidade Estadual Paulista (UNESP), como parte dos requisitos para obtenção do grau de Bacharel em Engenharia de Controle e Automação.

Orientador: Prof. Dr. Eduardo Paciencia Godoy

Sorocaba
2022

R484v

Ribeiro, Karinne Souza

Virtualização de uma estação de trabalho de um sistema de manufatura automatizado / Karinne Souza Ribeiro. -- Sorocaba, 2022

60 p. : il., tabs., fotos

Trabalho de conclusão de curso (Bacharelado - Engenharia de Controle e Automação) - Universidade Estadual Paulista (Unesp), Instituto de Ciência e Tecnologia, Sorocaba

Orientador: Eduardo Paciencia Godoy

1. Virtualização. 2. Gêmeos Digitais. 3. Comissionamento Virtual. 4. Visual Components. 5. FMS. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca do Instituto de Ciência e Tecnologia, Sorocaba. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

Virtualização de uma estação de trabalho de um sistema de manufatura automatizado

KARINNE SOUZA RIBEIRO

BANCA EXAMINADORA:

Prof. Dr. Eduardo Paciencia Godoy
Orientador/UNESP-Campus de Sorocaba

Prof. Dr. Fernando Pinhabel Marafão
UNESP- Campus de Sorocaba

Eng. Régis de Goes Vieira
UNESP – Campus de Sorocaba

Maio de 2022

AGRADECIMENTOS

Gostaria de agradecer, primeiramente, a todos os professores da universidade que me proporcionaram os conhecimentos para me formar engenheira. Agradeço especialmente meu professor orientador Eduardo Paciencia Godoy que me ajudou a desenvolver um tema tão recente e importante para os dias de hoje, principalmente no ramo industrial.

Gostaria de agradecer, também, ao Regis Vieira por todo o apoio durante o desenvolvimento do meu trabalho e por sempre estar disponível para me auxiliar; à cooperação em pesquisa com os professores da Poli-USP Fabrício Junqueira e Marcos Iris Pessoa, pela disponibilização da licença do Visual Components; e à a Festo pela disponibilização de acesso e aos desenhos 3D da estação de trabalho.

Por fim, deixo meus agradecimentos à minha família que me encorajou em toda a minha trajetória, independente dos desafios que apareceram.

RIBEIRO, K. S. **Virtualização de uma estação de trabalho de um sistema de manufatura automatizado.** Trabalho de Graduação (Engenharia de Controle e Automação) – Instituto de Ciência e Tecnologia de Sorocaba, UNESP - Universidade Estadual Paulista, Sorocaba, 2022

RESUMO

Revoluções tecnológicas cada vez mais têm impactado nos processos industriais, tornando-se um dos grandes pilares do desenvolvimento. A quarta revolução industrial ou Indústria 4.0 permitiu a difusão de conceitos como virtualização, comissionamento virtual e gêmeos digitais, que podem atuar de forma conjunta melhorando os processos, a produtividade e eficiência dos processos industriais. Visando implementar estes novos conceitos e integrá-los, este trabalho consiste em virtualizar uma estação de separação (Sorting) de peças de um sistema de manufatura flexível (FMS) da Festo, a qual separa em três canaletas peças cilíndricas de acordo com suas respectivas cores (preto, prata e vermelho). Assim, o modelo virtualizado da estação tem como objetivo reproduzir o funcionamento real da estação a partir de uma emulação de suas características dinâmicas e de interação entre seus elementos. O modelo virtualizado da estação foi desenvolvido com o software Visual Components e permite a integração e comunicação com softwares externos para execução de seu controle lógico e monitoramento de operação. Neste trabalho, o controle lógico do modelo foi desenvolvido através de sua programação num controlador lógico programável virtual da Siemens (PLCSim). O monitoramento da operação da estação através de variáveis do processo em dashboard foi desenvolvido com o software Node-RED. A integração entre o modelo virtualizado e os softwares externos foi realizada via comunicação Ethernet (Wi-Fi) TCP/IP (NetToPLCSim). A validação do modelo virtualizado foi realizada através da verificação de sua operação em cenários de uso da estação de trabalho. Os resultados demonstraram o uso do modelo para comissionamento virtual de sistemas de automação e potencial futuro para o desenvolvimento de um gêmeo digital da estação.

Palavras-chave: Virtualização, FMS, Visual Components, Simatic Step7, Node-Red

RIBEIRO, K. S. **Virtualization of a workstation of an automatized manufacturing system**. Graduation Project (bachelor's degree in Control and Automation Engineering) – Sorocaba Institute of Science and Technology, UNESP – São Paulo State University, Sorocaba, 2022.

ABSTRACT

Technological revolutions have increasingly impacted industrial processes, becoming one of the great pillars of development. The fourth industrial revolution or Industry 4.0 allowed the diffusion of concepts such as virtualization, virtual commissioning and digital twins, which can work together to improve processes, productivity and efficiency of industrial processes. Aiming to implement these new concepts and integrate them, this work consists of virtualizing a parts sorting station of a flexible manufacturing system (FMS) from Festo, which separates cylindrical parts into three channels according to their respective colors. (black, silver and red). Thus, the virtualized model of the station aims to reproduce the real functioning of the station from an emulation of its dynamic characteristics and the interaction between its elements. The station's virtualized model was developed with Visual Components software and allows the integration and communication with external software to execute its logic control and operation monitoring. In this work, the logic control of the model was developed through its programming in a Siemens virtual programmable logic controller (PLCSim). The monitoring of the station's operation through process variables in a dashboard was developed with Node-RED software. The integration between the virtualized model and the external software was performed via Ethernet (Wi-Fi) TCP/IP (NetToPLCSim) communication. The validation of the virtualized model was performed by verifying its operation in scenarios of workstation usage. The results demonstrated the use of the model for virtual commissioning of automation systems and future potential for the development of a digital twin of the station.

Keywords: Virtualization, FMS, Visual Components, Simatic Step7, Node-Red.

LISTA DE ABREVIATURAS E SIGLAS

CLP/PLC - *Controlador Lógico Programável*

TCP/IP - *Transmission Control Protocol/ Internet Protocol*

HTTP – HyperTextTransferProtocol

SMTP - *Simple Mail Transfer Protocol*

FTP - *File Transport Protocol*

NA/CNA - *Contatos ou Contatores Normalmente Abertos*

NF/CNF - *Contatos ou Contatores Normalmente Fechados*

VC - *Visual Componentes*

FMS - *Sistema de Manufatura Flexível*

IoT – *Internet das coisas*

I/O – Input/Output

KPI – Key Performance Indicator

C.V. – Comissionamento Virtual

API - *Interface de Programação de Aplicações*

IDE - *Ambiente de Desenvolvimento Integrado*

PWM - Modulação por Largura de Pulso

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1. INTRODUÇÃO

1.1 Justificativa

A humanidade tem como objetivo constante aprimorar processos e implementar novas tecnologias em prol de aperfeiçoar as atividades que impactam diretamente em seu desenvolvimento e bem-estar. Não é à toa que empresas bilionárias como *Instagram*, *Google*, *Facebook* buscam aplicar inteligência artificial para refinar as buscas e anúncios que aparecem para seus usuários, ou mesmo a gigante da tecnologia, a *Microsoft*, que vem criando aplicativos, *Power Apps*, capazes de serem integrados resultando em uma plataforma de desenvolvimento e automatização de processos, ou mesmos grandes nomes da indústria como *ABB*, *Weg* e *Siemens* trabalhando incessantemente para se diferenciar no mercado trazendo novidades de digitalização, como monitoramentos avançados de diversas variáveis de seus produtos.

Pode-se dizer que os avanços tecnológicos sempre foram vistos como uma revolução para as eras passadas, mas não se imaginava que era possível obter um avanço tão rápido e capaz de atingir todas as esferas da sociedade, desde usuários de aplicativos de entretenimento até grandes produtores agrícolas, automobilísticos e de maquinários pesados.

E não é necessário ir muito longe na história para perceber a diferença da velocidade dos avanços tecnológicos. Tome como exemplo a primeira revolução industrial que ocorreu no século XVIII, quando a primeira máquina à vapor foi desenvolvida. Quase um século depois, foi possível iniciar a produção em massa devido à descoberta da energia elétrica, a qual ficou em discussão por anos para determinar qual seria o melhor tipo de corrente a ser aplicada, a corrente contínua ou alternada.

Apenas em meados do século XX, houve um grande passo com o início da aplicação dos semicondutores, que possibilitou desenvolver computadores e processadores mais robustos, e implementar automação de processos com supervisão humana aliada ao potencial da internet. Um dos grandes exemplos da tecnologia dessa era e que revolucionou a forma de comunicação, por exemplo, foi o primeiro celular fabricado pela Motorola em 1983, e que desde então, vem se tornando objetos essenciais da humanidade.

Com esses avanços constantes e a necessidade evoluir mais e mais, em 2011 surgiu o primeiro termo relacionado à Indústria 4.0, que seria marcada: pela era do *Big Data*, pela Inteligência Artificial, pela integração de sistemas, pela robótica avançada, pela digitalização

e comissionamento virtual de sistemas, pelos gêmeos digitais dentre muitos outros. Segundo *Qin et al. (2016),* a quarta revolução poderia garantir um desenvolvimento real das empresas gerando um ambiente de produção confiável, impulsionados por interpretações e coleta de dados de forma inteligente, levando a tomadas de decisões corretas, algo de extrema importância para todas as indústrias, uma vez que "tempo é dinheiro" e decisões erradas requerem tempo para solucioná-las.

Mas não se pode esquecer que junto à necessidade de obter uma maior escala de produção e rentabilidade, veio a necessidade humana. O mundo está extremamente volátil quanto ao consumo. Cada vez mais os produtos duram menos, e cada vez mais as empresas nos incentivam a comprar o produto de última geração e desfrutar das novas tecnologias. Dessa forma, a indústria possui todas as variáveis para apostar 100% na Indústria 4.0: a necessidade de consumo e a ânsia pelo dinheiro.

No entanto, por ser uma tecnologia nova no mercado e que requer um alto investimento, cerca de 80% dos negócios ainda estão em estágios iniciais de transformação digital, uma vez que a aplicação dessas tecnologias é mais eficiente em estágios mais avançados de desenvolvimento.

Nesse contexto, visando possibilitar a aplicação de novas tecnologias e com baixo custo, além de explorar o potencial da Indústria 4.0, o projeto atual propõe implementar o conceito de virtualização e de comissionamento virtual em um processo de manufatura que consiste em separar materiais de acordo com três cores: vermelho, prata e preto.

1.2 Objetivos

O trabalho tem como objetivo virtualizar uma estação de separação (*Sorting*) de peças de um Sistema Flexível de Manufatura. Com o auxílio de um Controlador Lógico Programável, um desenho 3D desenvolvido pela Festo e com o protocolo de comunicação TCP/IP, será possível desenvolver um sistema no qual um CLP virtual irá simular o funcionamento da bancada enquanto o modelo 3D irá reproduzir em tempo real o seu comportamento. Algumas variáveis serão monitoradas e dispostas em dashboards no Node-Red. Além disso, tem como objetivo secundário explorar os conceitos da indústria 4.0, dentre eles a virtualização, comissionamento virtual e gêmeos digitais.

Por fim, o projeto espera possibilitar que seus resultados sejam utilizados como base para explorar a aplicação dessas novas tecnologias como comissionamento e gêmeos digitais em

experimentos de laboratório pelos alunos do curso de Engenharia de Controle e Automação da Unesp Sorocaba.

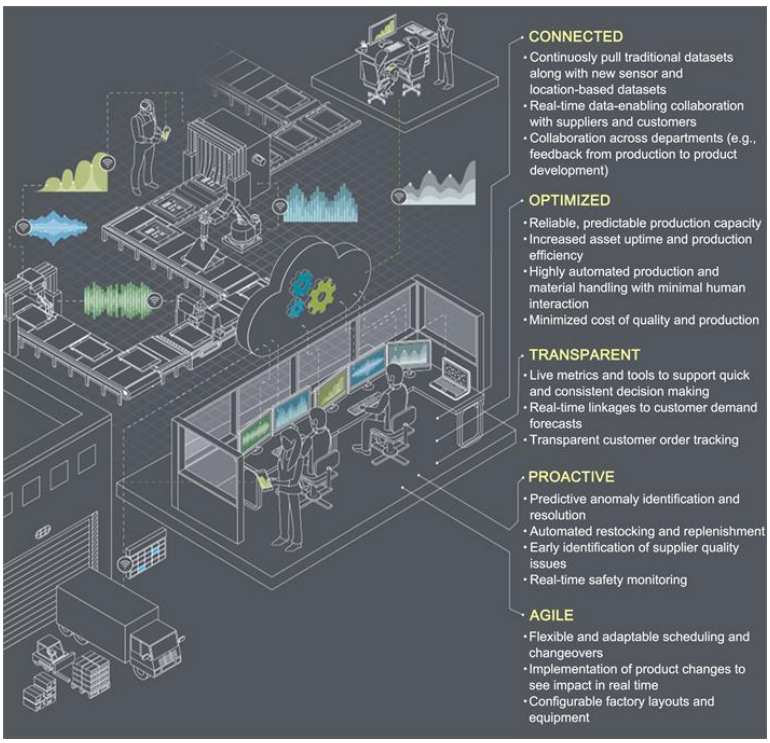
1.3 Estrutura do Trabalho

Para ampliar os conceitos a respeito do tema desenvolvido, inicialmente serão abordados conceitos sobre a indústria 4.0 e suas tecnologias. Em seguida, será detalhado o desenvolvimento do projeto, destacando as ferramentas utilizadas e os processos realizados. Por fim, serão discutidos os resultados obtidos frente às expectativas iniciais do projeto, além de possíveis soluções a serem implementadas a partir do projeto apresentado.

2. CONCEITOS DA INDÚSTRIA 4.0

Com o início do século XXI e com as evoluções da tecnologia, a indústria deparou-se com um novo conceito de manufatura. Segundo F. Basile, L. Ferreira (2020), sistemas de manufatura modernos consistem em estações de trabalho automatizadas e integradas com robôs, os quais possuem capacidade de manusear, segregar e armazenar produtos. Essa relação máquina-ambiente faz com que as indústrias se tornem cada vez mais "inteligentes" e se utilizem dos recursos da indústria 4.0 para tratar uma grande quantidade de dados que levarão a tomadas de decisões mais precisas, uma vez que agora, é possível prever diversos cenários. De acordo com Couzon P. *et al.*(2020), existem cinco fatores chaves que definem se uma indústria é inteligente, iniciando-se pela conectividade, em seguida pela otimização de processos, transparência, proatividade e por fim pela agilidade, como detalhado pela Figura 1. E não é à toa que esses *KPIs* foram determinados para definir uma indústria como "inteligente", pois são todos essenciais para o crescimento e desenvolvimento de uma empresa e podem ser fatais para o sucesso se não acompanharem a evolução do processo como um todo.

Figura 1: Cinco *KPIs* que definem uma indústria "inteligente".



Fonte: American Journal of Operations Research[1].

[1] Disponível em: https://www.scirp.org/html/1-1040746_103411.htm?pagespeed=noscript#ref9

2.1 Internet das Coisas

Internet das Coisas ou *IoT* vem sendo um dos termos mais falados no contexto da Indústria 4.0 e é utilizado para descrever uma rede de objetos físicos incorporados a sensores, softwares e outras tecnologias que conectam e trocam dados com outros dispositivos e sistemas pela Internet. Esses gadgets já estão sendo utilizados em muitas aplicações domésticas, facilitados pelo uso de inteligências artificiais como Alexa, da *Amazon*, ou o próprio *Google*, onde são sempre chamados pelo usuário para executarem alguma tarefa, como apagar as luzes de um ambiente, ligar para um colega, fechar um portão, entre diversas outras funcionalidades. Mas apesar de ser aplicado para uso domésticos, o *IoT* vem ganhando espaço em indústrias, como fabricação, transporte, energia, agricultura, permitindo que as organizações analisem e atuem em dados, permitindo tomadas de decisões inteligentes e em tempo real. Dessa forma, o *IoT* traz *insights* com o poder de transformar os negócios e reduzir custos por meio de aprimoramentos, como simplificação de processos operacionais e mecânicos e de redução de energia e de materiais desperdiçados. Atualmente, estima-se que a quantidade de aparelhos conectados deve duplicar em 2025, atingindo mais de 20 bilhões de conexões.

Além de possuir benefícios voltados para eficiência energética, economia financeira e de ativos, a Internet das Coisas também permite executar ações estratégicas que visam melhorar o desempenho da planta industrial, como por exemplo monitoramento remoto, manutenções preditivas, gerenciamento de instalações e fabricação com eficiência.

Uma vez que com o passar dos anos a mão de obra humana para realizar trabalhos repetitivos e de alto risco vem sendo substituída por maquinários ou robôs, a necessidade de monitoramento remoto tem se tornado cada vez maior. Empresas de grande porte como a Siemens, tem apresentado soluções de digitalização capazes de monitorar em tempo real a saúde dos equipamentos, seus alarmes e falhas com o objetivo de prevenir paradas não planejadas. Isso se deve pois, com a introdução dos maquinários inteligentes, o homem deixou de trabalhar em campo e passou a utilizar com mais frequência os escritórios, tornando necessário desenvolver uma forma de monitorar o funcionamento dos processos à distância. E aliada a isso, começaram a surgir mecanismos capazes de prever e programar melhores períodos de paradas para realizar manutenções preditivas com o intuito de manter a planta totalmente funcional e não ser surpreendido por falhas, que podem ocorrer a qualquer momento. Com isso, as indústrias aumentam cada vez mais sua capacidade de gerenciar de forma inteligente e fabricar seus produtos com eficiência.

2.2 Comissionamento Virtual

Dentre os conceitos da indústria 4.0 e da necessidade de supervisionar sistemas, uma vez que o homem se encontra cada vez mais distante das plantas industriais, não se pode deixar de destacar o comissionamento virtual (C.V), suas facilidades e benefícios.

O estudo sobre o comissionamento virtual iniciou-se há cerca de vinte anos, como forma de tornar possível integrar dispositivos reais à modelos de simulação antes que os sistemas reais fossem implementados. Segundo Carlsson *et al* (2012), esse tipo de estudo tem como objetivo verificar a lógica de controle desenvolvida antes de realizar a instalação física da programação. Essa etapa requer que o desenvolvedor ou o técnico responsável por realizar esse comissionamento tenha um conhecimento mais profundo de toda a linha de produção.

Silva *at al* (2017) define que existem três tipos de C.V: no nível de máquina ou equipamentos individuais; no nível de linha de produção ou no nível de sistema de produção. Note que o último tipo engloba os dois primeiros, sendo o modelo mais completo de virtualização. Para atingir esse tipo de comissionamento, alguns requisitos mínimos devem ser atendidos de acordo com Portelinha R. (2014), como a modelagem e simulação do sistema em análise (modelo elétrico, físico), o entendimento dos elementos físicos que pertencerão ao sistema (modelo cinemático) e um software de integração para realizar a interação entre os elementos físicos e virtuais.

Como forma de entender o dinamismo e a eficiência de aplicar o comissionamento virtual, alguns estudos e experimentos já foram realizados. Desenvolvido por ZÄH *et al* (2006), dois grupos de estudantes foram designados a automatizar um processo de fabricação de latas comissionados a um PLC SIEMENS S7-300. O primeiro grupo não utilizou nenhum software de simulação de controle ou da planta, enquanto o segundo desenvolveu todo o modelo de forma virtualizada para depois comissionar de forma física o CLP. Para o segundo caso, o tempo para resolver a tarefa foi 75% menor em relação ao primeiro. Nesse ponto vale ressaltar que não apenas o tempo reduzido de desenvolvimento é relevante, mas também o quanto que esse procedimento pode prever falhas e antecipar quanto a possíveis danos que poderiam ocorrer na planta.

No entanto, apesar dos comissionamentos virtuais trazerem estes benefícios, não se exclui a possibilidade de realizar o comissionamento real e aplicar alguns testes in loco. Isso porquê alguns comportamentos de equipamentos eletrônicos, como o tempo de acionamento de sensores, diodos, IGBTs, entre diversos outros, pode variar dependendo do estresse

elétrico/eletrônico/mecânico em que está atuando, podendo ter interferências nos resultados previamente obtidos através de simulações.

Por fim, além de facilitar o comissionamento de novos equipamentos, pode-se realizar estudos mais profundos de melhorias de layout de produção, no sensoriamento, no tempo de execução de atividades em cada estação de uma linha de manufatura, na possibilidade de digitalizar uma área que possui maquinários mais críticos, tendo como resultado uma planta otimizada e com melhor capacidade de gerenciar sua produção de forma a atender à demandas dos mercado que estão cada vez mais intensas e imediatas.

2.3 Gêmeos Digitais

As novas implementações realizadas na indústria a partir da virtualização abriram diversas possibilidades para o mundo digital. Cada vez mais as empresas estão buscando uma forma de migrar seus documentos físicos como projetos, cases de sucesso, contratos, entre diversos outros, para a nuvem. Para isso, os grandes dominadores do mercado como *Amazon, Google, Microsoft* vêm trazendo ferramentas para gerenciar essa grande quantidade de dados fora do ambiente do seu cliente, armazenando-os em nuvem. Isso só é possível graças a tecnologia do *Big Data*, que permite um elevado tráfego de dados em uma alta velocidade. Mas e se armazenar dados não é suficiente para uma indústria? E se for necessário explorar mais a planta industrial a fim de ter no detalhe cada ponto de execução? Surge então o termo gêmeos digitais

Gêmeos digitais podem ser definidos por realizar representações virtuais em tempo real de objetos, processos e sistemas, de forma a auxiliar no monitoramento do processo. De acordo com a *GlobalLogic*, os gêmeos trazem cinco principais benefícios para as empresas, entre eles avaliação de risco e tempo de produção acelerados, manutenção preditiva, monitoramento remoto em tempo real, melhor colaboração em equipe e melhor tomada de decisão financeira. Analise de forma individual esses benefícios e verifique como estão conectados a partir de uma cadeia colaborativa, ou seja, um benefício é devido ao outro e assim por diante. Quando um sistema é capaz de realizar um monitoramento remoto, técnicos especializados naquele processo podem verificar o funcionamento dos componentes de forma que caso perceba falhas constantes, possa planejar uma parada para realizar uma manutenção preditiva/preventiva. Esse tipo de tomada de decisão baseado na análise de risco pode prevenir que várias ações em cadeia prejudiquem a empresa, como elevados gastos com reparo e com horas de máquina parada, resultando em prejuízos, instabilidade no

mercado financeiro e para os colaboradores. Por isso, esse tipo de tecnologia está sendo cada vez mais implementado das indústrias.

Quanto à sua estrutura, os gêmeos digitais são constituídos por elementos primários de armazenamento de dados, iniciando-se em dados históricos que mostram o desempenho das máquinas, usualmente armazenados nos próprios logs dos computadores; os dados em tempo real, que enviam atualizações contínuas de seus sinais de entrada e de saída e os dados do futuro, que a partir do *machine learning* podem prever situações e indicar a melhor tomada de decisão.

Esse tipo de mercado vem chamando a atenção de empresas dominantes, como da Rolls-Royce que melhorou a eficiência dos seus motores a jato a partir da análise da atuação do piloto e das condições de operação do motor em voo.

Assim, compreender sobre essas tecnologias da indústria 4.0 e procurar implementa-las está se tornando cada vez mais necessário não apenas para melhorar os processos internos, mas também para se manter no mercado e não ser passado para trás pelos seus concorrentes.

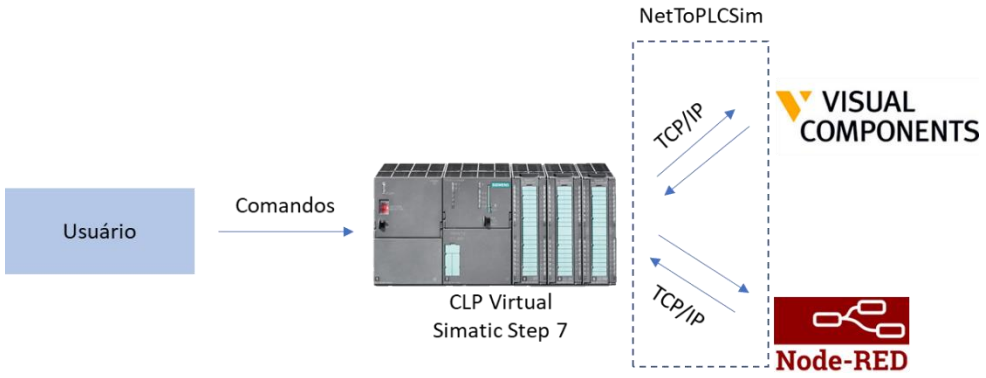
3. VIRTUALIZAÇÃO DA ESTAÇÃO DA FMS

O trabalho em questão fez uso de ferramentas de simulação, tais como Simatic Step7, o qual utiliza linguagem de baixo nível (*Ladder*) para a programação do CLP e o Visual Components para reproduzir o funcionamento de uma cadeia produtiva a partir de componentes 3D. Além disso, foram utilizados serviços como o Node-RED e a ferramenta NetToPLCSim para realizar a comunicação entre os softwares utilizando o protocolo TCP/IP.

3.1 Proposta do Trabalho

O presente trabalho visa aplicar e ampliar os conceitos de indústria 4.0 em um sistema de manufatura flexível de forma a reproduzir, virtualmente, o funcionamento de uma de suas estações, utilizando ferramentas já disponíveis no mercado. Com o software da Siemens que permite a programação do CLP virtual e a integração com os sistemas de monitoramento é possível obter um sistema capaz de ser controlado a distância e ao mesmo tempo ser analisado *step by step* de sua cadeia produtiva, como mostra a fluxo da Figura 2.

Figura 2: Arquitetura do Sistema.

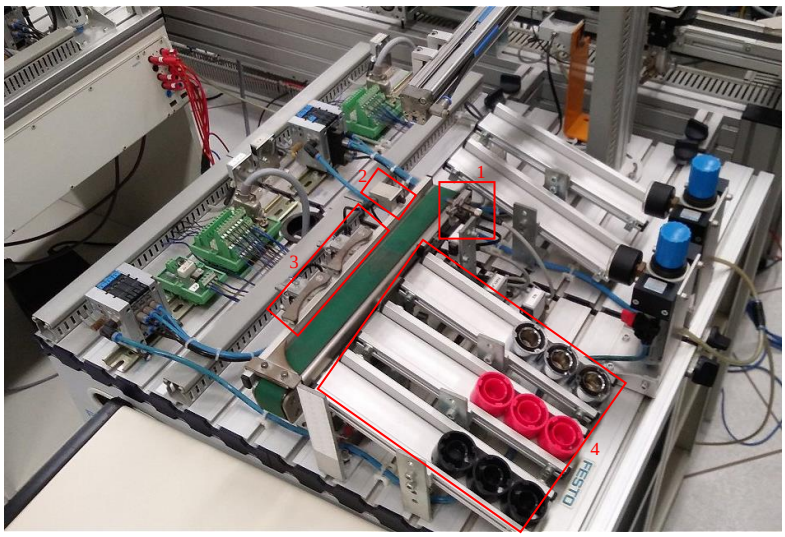


Fonte: Autor

Inicialmente, o usuário deve compreender a unidade da FMS a ser implementada no projeto. De maneira simplificada, a estação é denominada de *sorting* (ordenação) e é responsável por separar peças cilíndricas de cores vermelha, prata e preta a partir do sensoriamento presente no início do sistema. Os sensores são de natureza capacitiva e

fotoelétrica e atuam em pares booleanos de forma que (0,1), (1,1) e (0,0) representam peças de cores vermelha, prata e preta, respectivamente. A estação tem limite de 5 peças por unidade de armazenamento, sendo que a partir do momento em que estiver completa, mais nenhuma peça da respectiva cor poderá ser enviada para seleção, mantendo o sistema em *stand by*. Além disso, a estação apresenta 3 componentes mecânicos, um dos quais deve segurar a peça para verificação de sua cor e dois "braços" que irão direcionar os produtos para seus locais de armazenamento. A Figura 3 mostra cada um desses componentes, destacados por retângulos vermelhos enumerados de 1 a 4, onde tem-se os sensores, o pino de avanço, os braços mecânicos e as unidades de armazenamento, respectivamente, da estação real localizada na Unesp, no Campus de Sorocaba.

Figura 3: Estação Sorting real da FMS no laboratório de automação da Unesp.



Fonte: Automação das Estações do FMS Festo[2].

Assim, para que o sistema realize as separações das peças é necessário desenvolver uma programação no CLP da FMS, definindo os bits no qual deseja-se mapear o funcionamento utilizando nomenclaturas como Ix.x ou Qx.x, onde x indica um número real de 0 a 9, e I e Q representam bits de entrada e saída respectivamente. Essa leitura dos dados pelo processador permitirá acionar, em tempo real, os bits conforme a estação recebe as peças, resultando em inicializações e paradas, reconhecimento sensorial, acionamento dos braços mecânicos e verificação das unidades de armazenamento.

[2] Disponível em: https://alanmmolina.github.io/projects/FMSFesto

No entanto, como o presente estudo não foi realizado presencialmente no laboratório, para efeitos de simulação, o usuário deverá simular os bits de comando a partir de um CLP virtual disponibilizado pelo Simatic Step 7.

Com o processo definido e executado pelo PLC, seja de forma remota ou presencial, para que seja aplicada umas das definições da indústria 4.0, como virtualização, por exemplo, é necessário reproduzir as etapas de forma visual para que o usuário possa acompanhar o funcionamento da estação como um todo.

Apesar de parecer complexo, diversas ferramentas possibilitam desenhar e desenvolver sistemas a partir de um modelo real, aplicando conhecimentos mais avançados em desenho técnico ou utilizando modelos pré-definidos, por exemplo. No caso deste estudo, será utilizada uma renderização da estação, fornecida pela Festo, de forma a obter uma reprodução o mais próxima possível do real. Todo o processo será monitorado pelo software Visual Components, o qual já possui ferramentas que permitem estas edições além de se conectar com outros softwares a partir de protocolos de comunicação, como TCP/IP ou mesmo OPC UA.

Por fim, para finalizar o projeto e permitir uma integração de forma que seja possível verificar o funcionamento do sistema e acioná-lo remotamente a partir de uma dashboard, algumas variáveis do sistema, consideradas as mais críticas, devem estar visíveis para o usuário, que poderá tomar decisões como parar a produção, esvaziar uma das unidades de armazenamento ou reiniciar o sistema em caso de falhas, aproximando o funcionamento do sistema para o conceito dos gêmeos digitais tão abordado nesta última revolução industrial. Neste caso, o Node-RED será utilizado para realizar leitura e escrita de dados no CLP também utilizando protocolos de comunicação, como o TCP/IP.

3.2 Materiais e Métodos

Como mencionado anteriormente, o projeto necessita de um software de desenvolvimento de programação base, duas ferramentas para visualização e uma única ferramenta para realizar a integração entre todos. Foram utilizados então, respectivamente, o Simatic Step 7, Visual Componenets, Node-Red e NetToPLCSim.

3.2.1. Simatic Step 7

O SIMATIC STEP 7 é uma ferramenta de engenharia, da Siemens, abrangente capaz de configurar, programar, testar e diagnosticar todas as gerações de controladores SIMATIC,

sejam baseados em PLC ou PC. Uma vez que a unidade controladora da estação da FMS pertence ao mesmo fabricante do software, decidiu-se utilizá-lo para maior compatibilidade de processos e acionamentos.

Em suas versões mais recentes, permite realizar comunicações utilizando diversos protocolos, como TCP/IP, OPC/UA, Ethernet, Profibus, entre outros. No entanto, como na estação física da universidade é utilizada uma versão mais antiga da família dos controladores, a S7-300, será utilizada apenas a comunicação via TCP/IP.

O TCP/IP é um dos principais protocolos de envio e recebimento de dados, permitindo que dois computadores de comuniquem. De acordo com as definições TCP significa *Transmission Control Protocol* (Protocolo de Controle de Transmissão) e IP significa *Internet Protocol* (Protocolo de Internet). Separado em quatro camadas, o TCP/IP representa um conjunto de protocolos, que são subdivididos em: aplicação, transporte, rede e interface. A aplicação é utilizada para enviar e receber informações de outros programas, podendo encontrar protocolos como SMTP (*Simple Mail Transfer Protocol* – realiza envio de mensagens de correio eletrônico através da internet), FTP (*File Transport Protocol* – realiza a transmissão de arquivos entre computadores), e HTTP (*Hypertext Transfer Protocol* – lida com a comunicação entre um servidor web e um navegador web) e que em seguidas são enviadas para as camadas inferiores. Na camada de transporte, é verificado a integridade dos dados recebidos e que em seguida são divididos em pacotes, os quais serão processados pela camada da rede, anexando-os ao endereço virtual (IP) do computador destinatário. Por fim, na camada de interface, os dados são enviados pela rede, usualmente utilizando a Ethernet.

Por ser um software voltado para a engenharia e controle de CLP, será utilizada a linguagem Ladder para desenvolver a programação da estação. Esta linguagem utiliza os princípios de contatos normalmente abertos (CNA) e normalmente fechados (CNF) que quando combinados, geram uma relação inversa de controle da variável, ou seja, se esta variável for acionada a partir de outras duas do tipo NA e NF, caso elas sejam ativadas simultaneamente, não permitirão que a saída receba o sinal lógico um, mantendo-a inativa. E isso ocorre porque o contato NA permite a passagem do sinal (tensão) apenas quando for acionado, de forma física ou virtual, enquanto o CNF atua de forma contrária. Na maioria dos casos, os CNFs foram utilizados para cessar uma ação ou uma atividade da bancada e os contatos NA para dar início ou realizar uma ação, como por exemplo para indicar a chegada de uma peça na estação.

Além disso, foram utilizadas outras lógicas de programação, como de *timers* que controlam por quanto tempo uma saída deve permanecer ativa, ou depois de quanto tempo será permitida a passagem de sinal, de forma a permitir uma fluidez no funcionamento do sistema e impedir que erros ocorram devido a leituras incorretas dos sinais das variáveis.

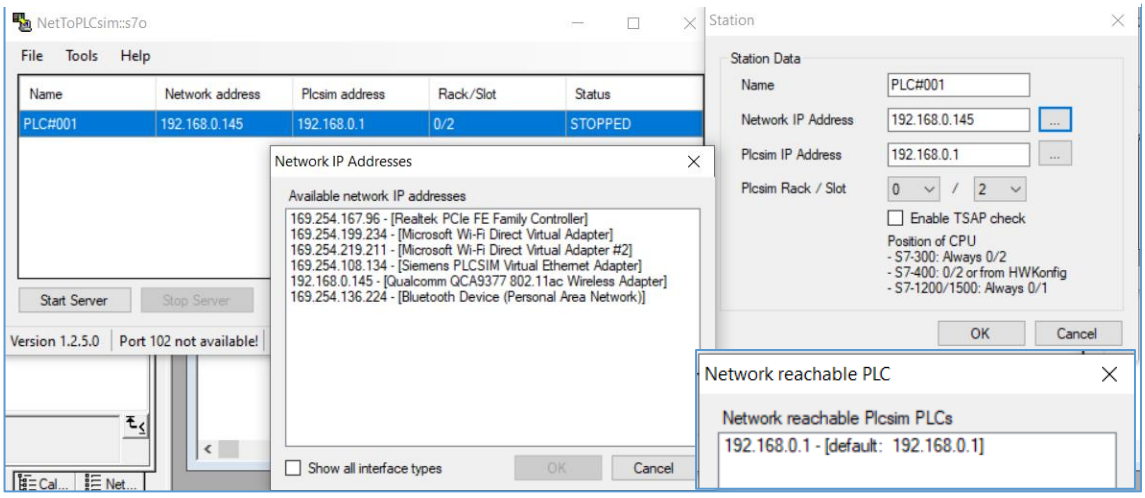
Por fim, também é foram utilizados contadores decrescentes para contabilizar a quantidade de itens armazenados em cada unidade de armazenamento, de forma que quando fosse atingida a capacidade máxima, seria realizada uma trava no sistema de produção.

3.2.2 NetToPLCSim

O NetToPCLSim é um emulador de rede que possibilita uma extensão de funcionalidades disponíveis nos CLPs Siemens. Basicamente, esse software atua como um conversor dos protocolos S7-Protocol e IsoOnTCP, permitindo a conexão com diferentes supervisórios, como esquematizado na Figura 2.

Para funcionar, o NetToPLCSim deve estar conectado no mesmo IP da rede e com a porta 102 liberada, como configurado na Figura 4. Quanto ao endereço do PLCSim, é necessário habilitar a função no Simatic Step 7. Para isso deve-se configurar o CLP virtual no menu principal acessando em *Opções*, *Set PG/PC Interface*, e selecionar a opção *PLCSIM.TCPIP.1*, e então será fornecido um IP do próprio PLC. A opção de Rack/slot deve ser preenchida conforme as configurações de barramento do controlador. No caso da série S7-300 utilizada, a posição da CPU será sempre 2. A partir de então, é possível conectar o CLP a outros softwares e monitorar o comportamento de todas as variáveis de interesse definidas.

Figura 4: Configuração do NetToPLCSim.



Fonte: Autor

3.2.3 Visual Components

A Visual Components (VC) é uma desenvolvedora de software de simulação 3D para manufatura, utilizado para aplicações que incluem planejamento de layout, simulação de produção, programação off-line e verificação de PLC. Fundada em 1999, na Finlândia, a companhia possuía a filosofia de tornar a tecnologia de simulações e projetos fácil de ser utilizada e acessível a organização de todos os tamanhos. Assim, em 2016, foi introduzido no mercado a família de produtos Visual Components 4.0, construído em uma nova arquitetura e plataforma de software, projetados para aproveitar o hardware de computação moderno e os processadores de 64 bits. *APIs* abertas foram incluídas no design para permitir fácil personalização e desenvolvimento de aplicativos de terceiros.

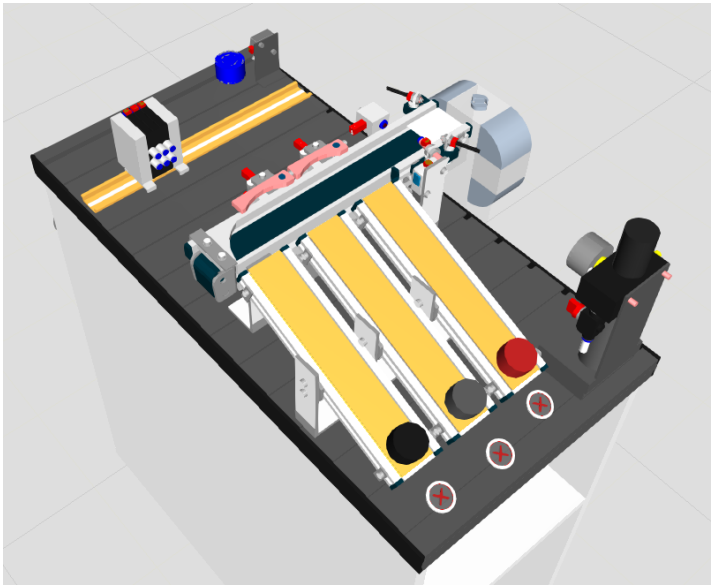
Então, para desenvolver a virtualização da estação foi utilizado o VC Premium versão 4.4 a qual permite os usuários usarem componentes prontos para criar, construir e simular processos de produção. Ele também possui recursos para educação de robôs, conectividade PLC, realidade virtual, modelagem de componentes e programação off-line, além de ferramentas para aplicações de robótica sofisticadas. Apesar de possuir esta robustez para criar componentes, foi importado para o ambiente de simulação um modelo pronto da bancada fornecido pela Festo.

O modelo, incialmente, é considerado pelo software como um único componente, devendo ser configurado e readaptado pelo usuário, “extraindo” os links e componentes de forma a possibilitar a configuração individual de cada um deles. Além disso, vale destacar a necessidade de nomear os componentes de forma a facilitar a identificação em etapas futuras de desenvolvimento. Para realizar essa extração, pode-se utilizar a função “extrair” disponibilizada na aba de modelagem e apontar como links os componentes que se deseja modelar.

Além de possibilitar a criação de componentes, o VC permite que o usuário expanda a aplicação utilizando alguns princípios de comportamentos cinemáticos, como exemplo definir juntas translacionais ou rotacionais. Estes tipos de comportamentos permitem que a simulação se torne mais dinâmica e mais próxima de representar o comportamento da bancada. A **Figura 5** ilustra como seria o modelo completo de modelagem da bancada no ambiente de desenvolvimento do Visual Components.

Aqui, vale ressaltar que estes comportamentos podem ser definidos devido às diversas funcionalidades contidas na aba “Modelagem” da versão Premium do Software. Além de ser possível transformar as juntas e extrair componentes, existem “Comportamentos” *(Behaviors)* que podem ser aplicados de forma a atribuir funções aos itens, como por exemplo, realizar a leitura de um sinal booleano a partir de um script em Python adicionado ao componente. Todas essas funcionalidades de modelagem que se aplicam à bancada *Sorting* serão explicadas nas seções seguintes.

Figura 5: Modelagem da bancada realizada no Visual Components.



Fonte: Autor

3.2.4 Node-RED

Node-RED é uma ferramenta de programação para integração de dispositivos, APIs e outros aplicativos baseados na web. É possível comunicar e gerenciar ações e estados a partir de nós configuráveis. Os nós podem ser construídos e instalados individualmente no ambiente, e aqueles que já foram configurados podem ser exportados e importados para outros fluxos posteriormente. Seu ambiente é baseado em navegador e pode ser oferecido localmente ou através de outro provedor via IP.

Os fluxos no ambiente Node-RED são bem simples de utilizar e apresentam diversas funcionalidades que permitem converter e manipular variáveis de entrada e saída, além de possuir bibliotecas para construção de dashboards que podem ser para consultas de status ou

funcionamento do sistema, assim como para realizar acionamentos a partir de escrita de dados no controlador, por exemplo.

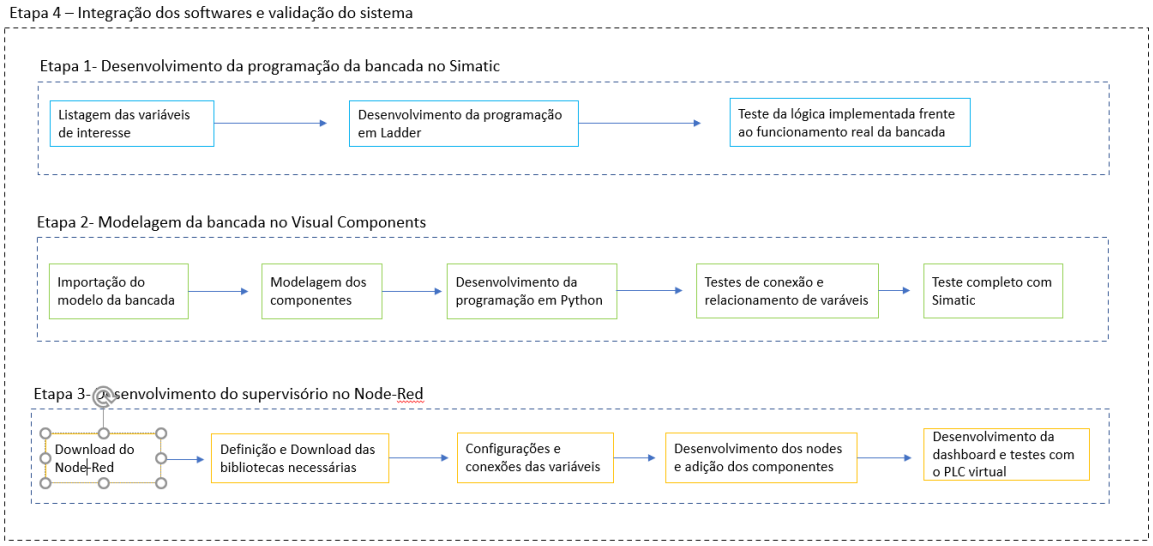
Por ser uma ferramenta que realiza a integração com outros dispositivos/softwares, a ideia é representar o funcionamento das variáveis definidas no Simatic, de forma a verificar qual a cor da peça que chegou à estação, se as unidades de armazenamentos estão completas ou se é necessário reiniciar o sistema, por exemplo. Para isso, foi necessário realizar o download do Node-Red disponível gratuitamente na internet de forma a utilizá-lo como um servidor local via web. Uma das maneiras de acessá-lo é abrindo o prompt de comando do sistema operacional (para o presente estudo foi utilizado o Windows 10) e digitar "node-red", pressionando *enter* em seguida. Após inicializado, digitando como endereço web *locallhost:1880 (*endereço padrão definido pela ferramenta) foi possível acessar a ferramenta.

Para desenvolver o projeto, foram utilizadas as bibliotecas *node-red-contrib-s7* para leitura dos dados do CLP*, node-red-dashboard e node-red-contrib-ui-led,* para visualização dos dados.

4. DESENVOLVIMENTO

Para explorar e detalhar como a modelagem de todos os softwares foram realizadas, foi definido um diagrama para cada etapa de desenvolvimento, como ilustrado pela Figura 6. Por se tratarem de softwares com funcionamento distintos, mas por dependerem da ferramenta responsável por realizar o controle da bancada, o ponto de partida definido pela etapa 1 é iniciar pela programação em Ladder. Em seguida, na etapa 2 é necessário modelar todos os componentes do ambiente do Visual Components, os quais devem ser conectados ao Simatic para testar o funcionamento e relação entre as variáveis de controle. Já na etapa 3, é desenvolvida a dashboard no Node-Red e realizado o teste de conexão com o Simatic. Por fim, na etapa 4, é realizada a integração de todos os softwares das etapas 1 a 3, com o objetivo de verificar se o objetivo inicial do projeto foi atingido, mostrando o resultado na seção RESULTADOS.

Figura 6: Etapas de desenvolvimento para realizar a modelagem completa do sistema.



Fonte: Autor

4.1 Etapa 1: Programação da Automação da Estação

Após compreender como a bancada se comporta, para que seja possível realizar qualquer tipo de controle, é necessário desenvolver a programação em Ladder. Para iniciar, foram realizadas configurações básicas de hardware, como definir a CPU, a estação a ser programada, no caso SIMATIC 300 *Station* e selecionar o trilho (*Rail*) pertencente a família,

como a *CPU 314C-2 PN/DP – 6ES731463H040AB0 – V3.3.* Após isso, o controlador está pronto para ser programado a partir do OB1 gerado pelo programa. Ademais, como o usuário será responsável por definir o *status* dos bits de entrada e saída, o painel de comando (PLCSim) deve ser, também, configurado no menu principal em *Opções*, *Set PG/PC Interface*, e selecionada a opção *PLCSIM.TCPIP.1*.

Assim como em todos os tipos de linguagem, programar requer pensar em diversas possibilidades de combinações de variáveis que, de alguma forma, possam alterar o funcionamento do sistema indevidamente. Com o intuito de evitar esse tipo de interrupção, as variáveis da programação foram dispostas em uma tabela indicando se seriam variáveis booleanas de entrada (Ix.x) ou saída (Qx.x). Dessa forma, foi possível obter uma tabela de símbolos dispostos na Tabela 1, representando apenas os bits de maior relevância para o funcionamento da estação e tendo como parâmetro as declarações das variáveis.

Tabela 1:Tabela de símbolos com algumas das variáveis de entrada e de saída.

SÍMBOLO	ENDEREÇO		TIPO DE VARIÁVEL	DESCRIÇÃO BREVE
LIGA_ESTACAO	I	0.0	BOOL	Liga a estação para permitir o início do processo
PRESENÇA_PEÇA	I	0.1	BOOL	Sinal entre estações que permite a inicialização
SENSOR_FOTOELETRICO	I	0.4	BOOL	Verifica se a peça colocada é vermelha ou prata.
SENSOR_CAPACITIVO	I	0.5	BOOL	Verifica se a peça colocada é prata
SEGURA_PECA	Q	0.2	BOOL	Segura a peça para verificar qual é a cor
ATIVA_SERVO_1	Q	1.0	BOOL	Aciona o Servo para levar a peça vermelha à sua canaleta
R1	Q	1.4	BOOL	Restart Vermelho
ATIVA_SERVO_2	Q	2.0	BOOL	Aciona o Servo para levar a peça prata à sua canaleta
R2	Q	2.4	BOOL	Restart Prata
ATIVA_PECA_PRETA	Q	3.2	BOOL	Sinaliza que a peça é preta
R3	Q	3.4	BOOL	Restart Preto

Fonte: Autor

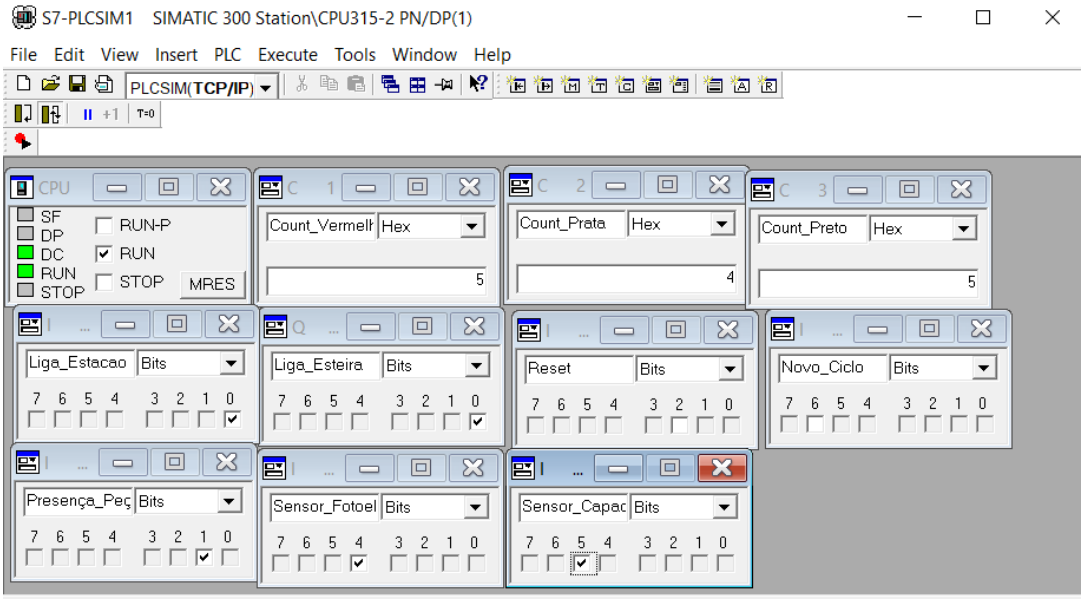
Note que definir as variáveis no formato Ix.x e Qx.x é essencial para que o software reconheça o tipo de dado a ser lido/escrito além de conseguir salvar esse memorial em uma tabela de símbolos.

Como forma de facilitar e identificar a qual parte do sistema pertence a variável em questão, foi utilizado o bit 0.x como bit geral de acionamento da bancada, bit 1.x, 2.x e 3.x

como ações relacionadas com as peças das cores vermelha, prata e preta respectivamente, como por exemplo o acionamento do primeiro braço mecânico (servo motor 1) enviando a peça vermelha para a primeira canaleta (bit referência Q1.0).

Utilizando então o CLP virtual, a partir do painel de comando, foi possível simular o funcionamento do sistema, selecionando os bits de entrada e de saída e observando se os contadores estavam realizando a contagem correta, se os bits referentes a alguns processos, como acionamento dos braços mecânicos estava ocorrendo no momento certo e em tempo hábil de segregar as peças e se os bits de reset estavam reiniciando a contagem das peças. A Figura 7 ilustra o painel de comando e como as variáveis foram selecionadas. Neste caso, note que ambos os sensores de reconhecimento de peça estão acionados, mostrando que uma peça prata chegou à estação e dessa forma ocupou uma posição da unidade de armazenamento, restando apenas 4 espaços.

Figura 7: Acionamento das variáveis para verificar o funcionamento do sistema.



Fonte: Autor

Apenas após confirmar o funcionamento completo da estação, como verificar casos em que as unidades de armazenamentos estivessem completas, se os contadores estavam sendo decrescidos em uma unidade e se os bits mais relevantes como reset e chegada da peça estavam atuando propriamente no sistema, é que se pôde seguir para a implementação visual no VC.

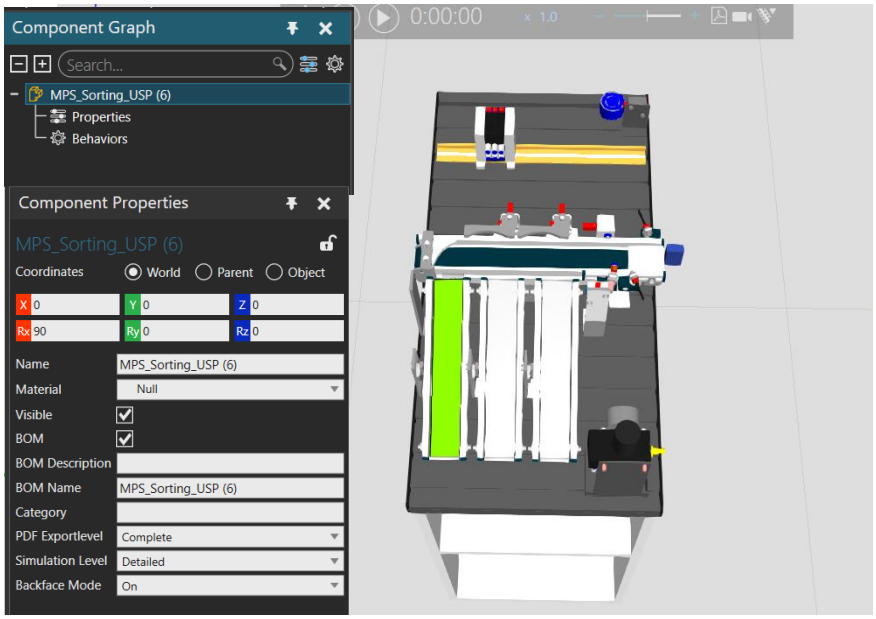
4.2 Etapa 2: Modelagem da Estação de Trabalho

Com o CLP programado e atuando de forma funcional, foi possível seguir para a etapa 2 de desenvolvimento, que é composta por modelar os componentes no ambiente do Visual Components, desenvolver uma programação em Python capaz de receber e compreender os comandos enviados pelo Simatic e por fim realizar a integração entre ambos os softwares.

4.2.1 Importação da Estação

Como mencionado anteriormente, o VC permite que o usuário desenvolva seus componentes, desenhando-os de forma a melhor encaixar no seu projeto, mas também permite a utilização de modelos prontos. Dessa forma, para o presente trabalho, optou-se por utilizar uma renderização do desenho 3D da bancada disponibilizada pela Festo.

Figura 8: Modelo original da Festo importado no ambiente do VC.



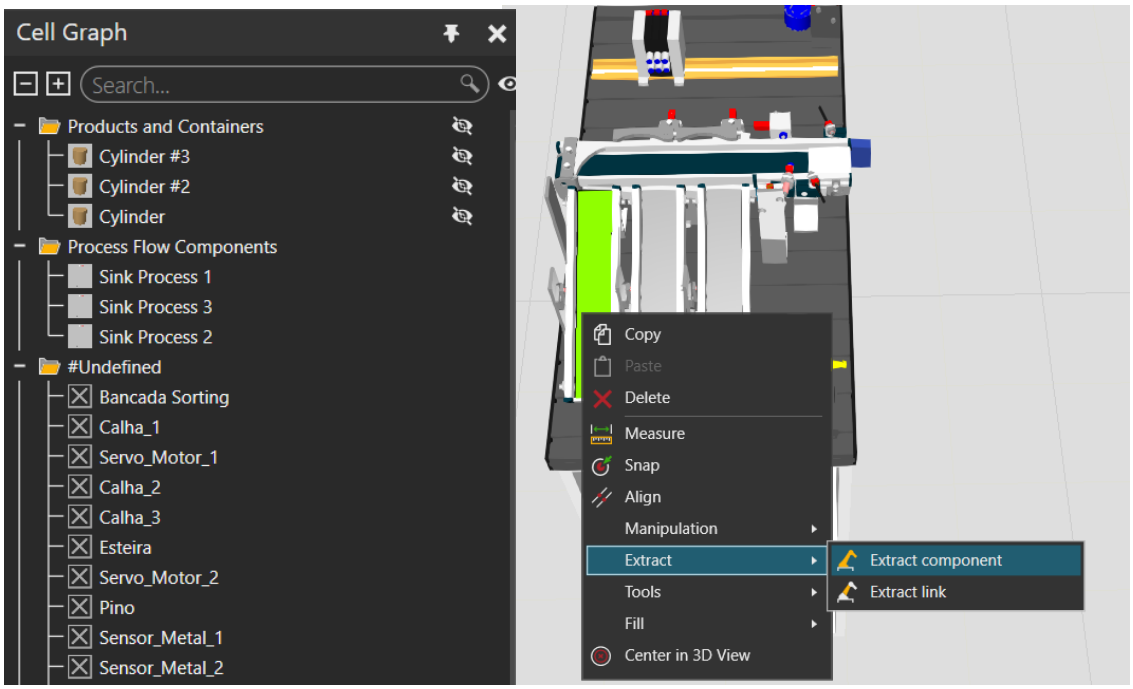
Fonte: Autor

Para importar o desenho da bancada para o modelo de desenvolvimento, deve-se ter um arquivo com extensão do tipo .stp, em seguida acessar a aba *Modeling*, selecionar a opção *Import – Geometry* e escolher o arquivo para ser importado para o VC. Nesse momento é possível realizar algumas alterações de orientação dos eixos X, Y e Z, além de possibilitar alterar unidades de medida. No entanto, foram mantidas as configurações padrões e obteve-

se o modelo representado pela Figura 8. Note que no painel de propriedade dos componentes existem campos de coordenadas para que o usuário consiga alterar a orientação do modelo ou mesmos dos itens que o compõem. No caso, foi necessário que a bancada fosse rotacionada em 90 graus em Rx de forma a estar orientada para cima. Outro detalhe importante são os componentes gráficos que compõem a bancada. Nesses campos é possível verificar quais comportamentos ou propriedades estão atribuídos ao sistema.

Por fim, note que os componentes são considerados itens pertencentes à bancada. Para que seja possível configura-los é necessário extraí-los da bancada como ilustra a Figura 9. Então, para cada componente de interesse para o funcionamento do sistema, como as esteiras de rolagem, braços mecânicos, trava inicial e sensores, foi realizada a sua extração transformando-o em um componente. A relação de todos os itens pode ser verificada na aba *Process* do VC.

Figura 9: Extração dos componentes da bancada para configurações e modelagens.



Fonte: Autor

Após serem extraídos, os componentes foram nomeados de acordo com sua função obtendo uma lista com os braços mecânicos, denominados de *Servo Motores 1 e 2*, as 3 canaletas, denominadas *Calha 1, 2 e 3*, os *sensores capacitivos e fotoelétricos* e a trava de

segurança, denominada *Pino*. A seguir será detalhado como foi realizada a modelagem de cada um deles.

4.2.2 Servos Motores

Servo motores são dispositivos eletromecânicos utilizados em aplicações onde se deseja movimentar um objeto de forma precisa e controlada, garantindo sua rotação em posições, ângulos e velocidades específicas. Para tal, utiliza de sua junta rotacional que gira em torno de uma linha imaginária estacionária, denominada de eixo de rotação.

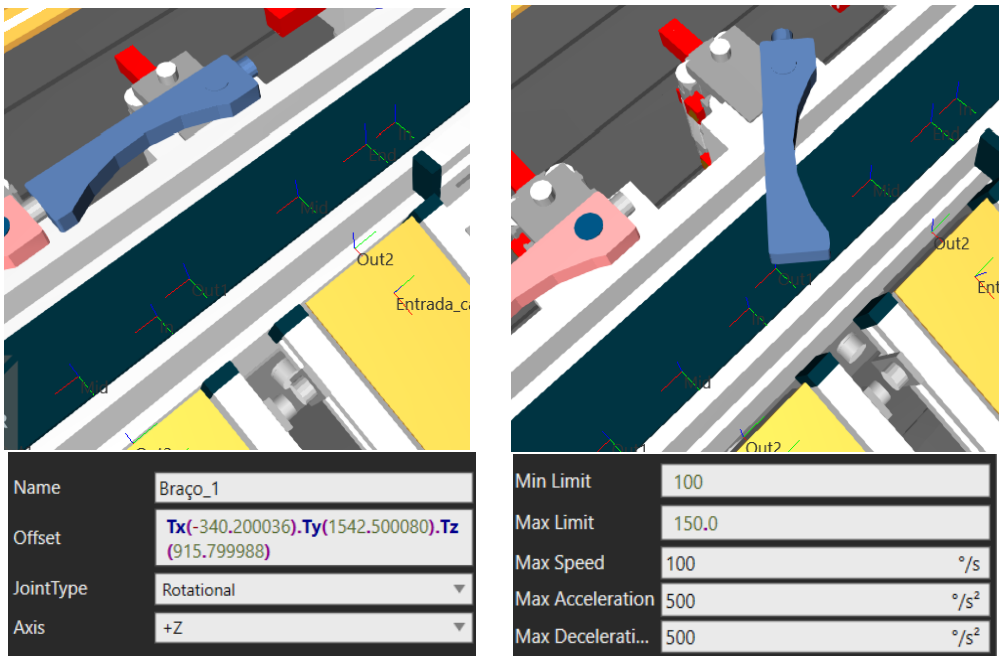
Na estação da bancada da Festo, os servos motores são utilizados para enviar as peças para suas devidas unidades de armazenamento. Assim, quando os sensores capacitivo e fotoelétrico enviam os sinais para o processador seguindo o padrão booleano descrito na seção anterior, os servos irão agir de acordo com a lógica implementada, ou seja, caso o par booleano seja (0,1), o Servo_Motor_1 será acionado direcionando a peça vermelha para a canaleta 1, enquanto que se o par for (1,1) o Servo_Motor_2 enviará a peça para a canaleta central e caso o par seja (0,0), nenhum servo será acionado, permitindo que a peça preta seja armazenada na última unidade da estação. Para conferir aos elementos do ambiente do VC esse dinamismo, foi necessário utilizar a aba *Modeling* para transformar estes servos em juntas rotacionais.

Então, inicialmente, para o braço mecânico 1, denominado Servo_Motor_1, foi necessário adicionar um “Comportamento” de *Servo Controller*, que será responsável por realizar o movimento do componente. Em seguida, para definir os ângulos iniciais e máximos de atuação, foi necessário extrair um “link” do componente Servo_Motor_1. Anteriormente, esse componente havia sido extraído da bancada original da Festo para que fosse possível realizar modificações e atribuir comportamentos. Mas agora, é preciso configurar em um nível abaixo da hierarquia do material, e por isso, utiliza-se da ferramenta de extração de link.

A partir de então, é possível escolher o tipo de junta a ser atribuída a esse link, que neste caso é a junta rotacional e escolher o eixo no qual o componente deve atuar. A direção do eixo varia de acordo com a orientação em que seu objeto está posicionado no ambiente de simulação, que neste caso foi definido o eixo +Z. Por fim, para determinar a amplitude de atuação do servo motor, utilizou-se a ferramenta *Interact* que permite simular a

movimentação do componente, definindo que o valor mínimo que o servo deve atingir é de 100º e máximo de 150º, obtendo as movimentações indicadas na Figura 10.

Figura 10: Modelagem e movimentação do Servo Motor 1, indicando seus valores mínimos e máximos de atuação



Fonte: Autor

Note que além do grau mínimo e máximo definidos, é possível escolher a aceleração e desaceleração de acionamento do servo motor. Apesar de não ser aplicado no presente estudo, em sistemas reais de manufatura, os servos motores podem ser acionados por sistemas pneumáticos ou controlados a partir de *PMW* e apresentarem velocidades distintas de avanço e recuo, dessa forma, o VC permite que estes parâmetros sejam ajustados.

Os mesmos processos foram aplicados para o braço mecânico 2, denominado de Servo_Motor_2, responsável por levar as peças pratas à canaleta central.

4.2.3 Trava/Pino para reconhecimento da peça

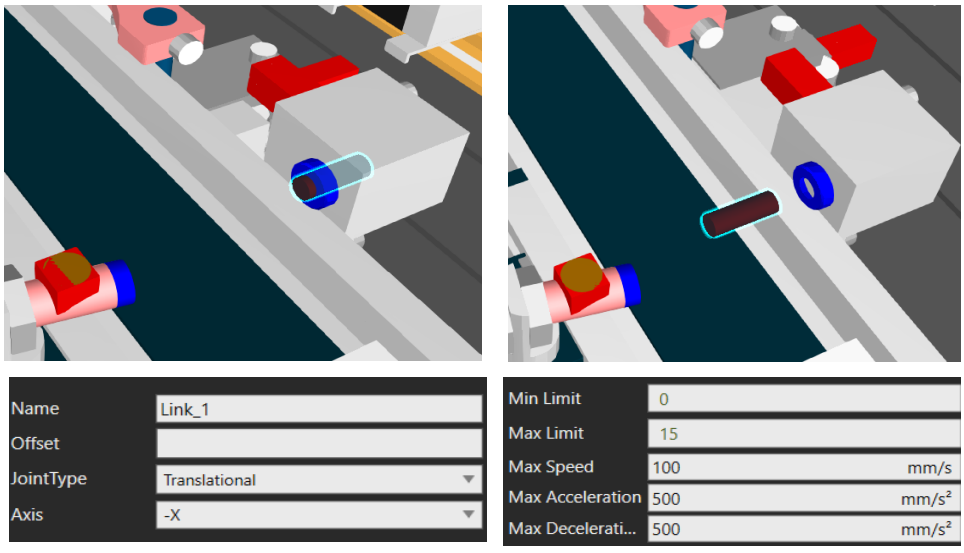
As tecnologias de sensoriamento e reconhecimento de materiais evoluiu muito nos últimos anos. Hoje é possível detectar, registrar e analisar objetos por toda a superfície terrestre apenas por captação de sinais. No entanto, apesar de ser considerada uma tecnologia que consegue gerar resultados em *real time*, os dados precisam de um determinado tempo de processamento para gerar uma informação.

Este tipo de situação pode ser verificado na bancada da Festo, pois os sensores precisam detectar a presença do componente e verificar qual o tipo de material presente. Como na prática o sensoriamento não ocorre de forma imediata, é necessário utilizar um pino ou trava para segurar esses componentes até que os sensores reconheçam o tipo de peça.

Assim, quando a estação recebe um sinal de que um novo componente será inserido na linha de produção, o pino translada ao longo de seu eixo impossibilitando a passagem da peça. Para que isso seja visualmente possível de ocorrer no VC, um processo de modelagem, parecido com o descrito para os servos motores, deve ser implementado.

Dessa forma, foi necessário incluir um comportamento de *Servo Controller* para que o pino avance e recue e em seguida, extraí-lo como um link para também modelar o componente em um nível inferior do material. Agora, como o pino deve transladar, o tipo de junta a ser definida foi a translacional ao longo do eixo -X e os limites foram definidos a partir da ferramenta “Interact” obtendo valore mínimos e máximos de 0 e 15 mm, respectivamente. A modelagem pode ser verificada na Figura 11.

Figura 11: Modelagem e movimentação do Pino com seus valores mínimos e máximos.



Fonte: Autor

4.2.4 Esteiras de Rolagem

Em uma linha de produção, para transportar os materiais por toda a cadeia produtiva, usualmente são utilizadas esteiras de rolagem de forma que possam ser controladas individualmente a fim de atender às necessidades de cada processo. No caso da bancada *Sorting*, o sistema é composto por 4 esteiras: a esteira principal, responsável por transportar o material por todo o sistema, e as outras três que devem levar suas respectivas peças às unidades de armazenamento.

Aqui, vale ressaltar que os componentes *default* do VC apresentam comportamentos já definidos e no caso de conectá-los com outros através da ferramenta "PNP", há uma comunicação entre os mesmos e a simulação fica fluida.

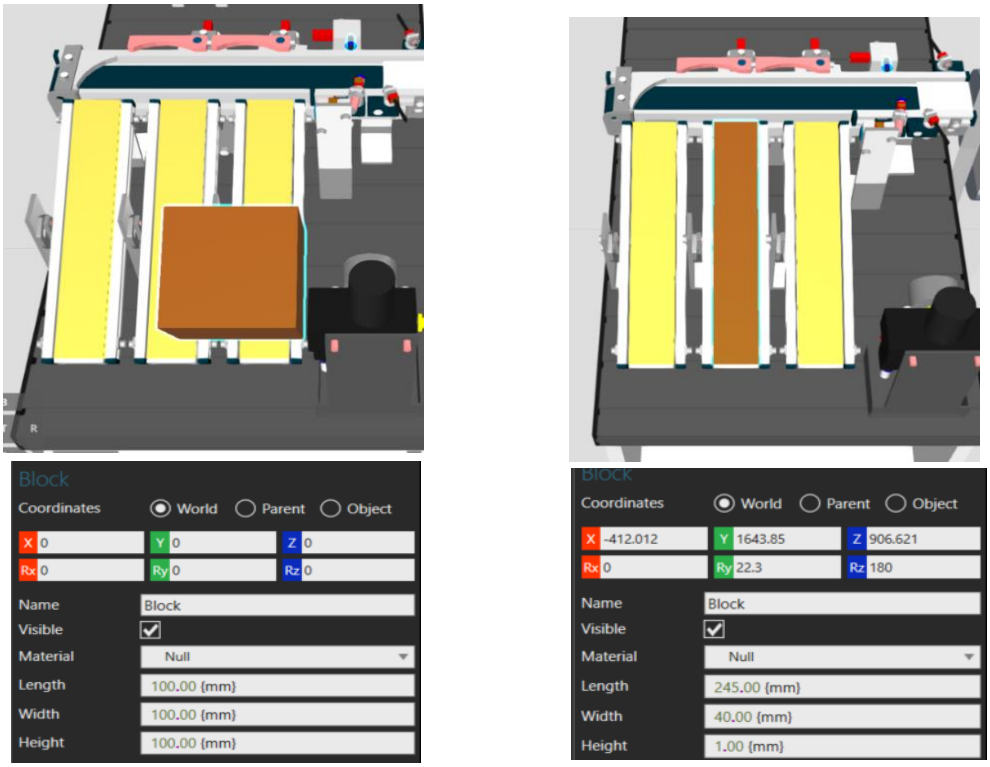
No caso da bancada, como a esteira principal será conectada a um gerador de peças, denominado de *Feed Conveyor*, e às Calhas 1,2,3 os próprios comportamentos dos componentes farão com que as peças percorram pela superfície da esteira sem necessidade de nenhuma configuração específica, uma vez que ela será utilizada apenas como meio de transporte e não como um fim de curso. Isso já não ocorre para as esteiras adjacentes, uma vez que a peça deve fluir por toda a extensão da esteira e ser armazenada.

Uma das formas de atribuir esse comportamento, foi utilizar dois artifícios da aba *Modeling*, adicionando um novo *Feature* denominado *box* (caixa), redefinindo seus padrões de altura, comprimento e largura para se encaixarem perfeitamente na canaleta da bancada, e em seguida, utilizar a ferramenta *Wizard* para transformar essa nova caixa em uma esteira. Com isso, ao conectar os novos meios de transporte à esteira principal, a peça fluirá por todo o compartimento.

A Figura 12 ilustra a inserção da funcionalidade "caixa" no ambiente de desenvolvimento, sendo a imagem à esquerda o material ainda em seu estado original e à direita já modelado de acordo com os padrões da bancada.

O processo foi aplicado para as três esteiras adjacentes à principal e renomeadas como Calha 1,2 e 3, responsáveis pelo armazenamento das peças vermelha, prata e preta, respectivamente.

Figura 12: Modelagem das esteiras/unidades de armazenamento.



Fonte: Autor

4.2.5 Advanced Feeder

Em um sistema de produção, as estações de processamento podem ser alimentadas por robôs ou mesmo pelas próprias esteiras sequenciais, de forma a obter um processo contínuo e atender os requisitos dos projetos.

Quando se fala de uma FMS, é válido lembrar que ela é composta por diversas estações formando um sistema de manufatura flexível, ou seja, os componentes que são processados interagem todas as estações de execução alimentando a subsequente. No entanto, como para o presente estudo foi realizado apenas a virtualização de uma única estação, foi necessário simular o fornecimento de peças a partir de um *Feeder* ou alimentador contido em uma das bibliotecas padrões do VC.

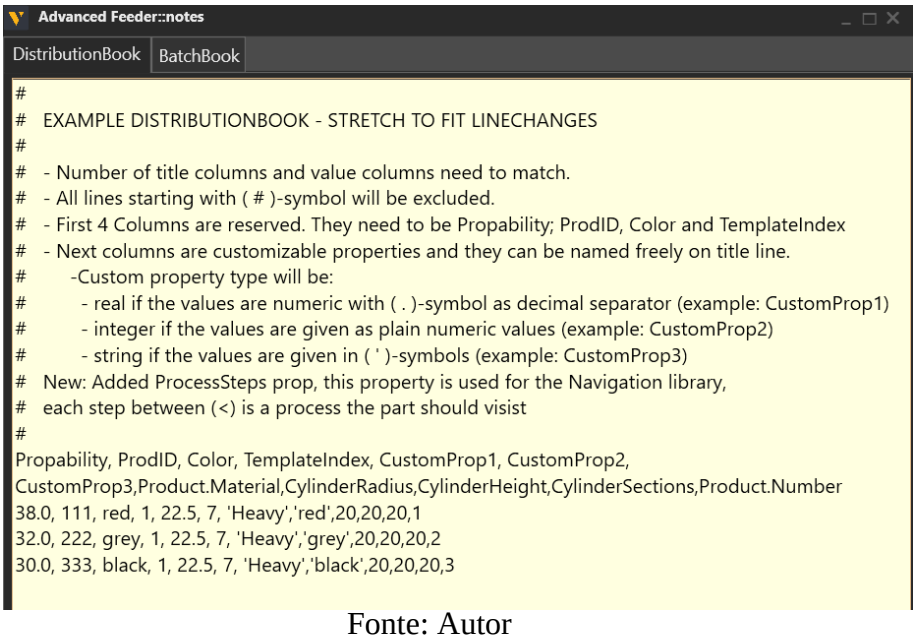
Os *Feeders* são customizáveis e capazes de reproduzir distintos componentes, como caixas, cilindros, garrafas, entre outros, além de possibilitar a alteração do tipo e cor do material gerado. Dessa forma, utilizou-se *o Advanced Feeder* que pode ser customizado a partir de um script como mostra a Figura 13.

O script de edição permite utilizar dois tipos de padrão de distribuição: o *DistributionBook* e o *BatchBook*, que significam livros de distribuição e de lotes, respectivamente. Como a ideia é simular a chegada aleatória de peças, será utilizada a primeira opção.

Então foi necessário definir algumas variáveis padrões do script como Probabilidade, *ProdID*, Cor, *TemplateIndex*, Propriedade de Customização 1,2 e 3. A primeira variável deve ser definida de acordo com a distribuição que se deseja obter, como por exemplo na Figura 13, foi determinado que as peças vermelha, prata e preta seriam produzidas da proporção de 20/50/30%, respectivamente. A variável *ProdID* foi mantida padrão; a Cor foi alterada para atender aos requisitos da bancada e as demais foram definidas de forma a produzir o mesmo tipo de material.

Note que até então não se sabe qual o tipo de componente será criado, por isso foram adicionadas as propriedades *Product.Material, CylinderRadius, CylinderHeight, CylinderSections e Product.Number*. Assim, cilindros com 20mm de altura, raio e seção serão gerados atendendo aos *match codes* padrões definidos inicialmente.

Figura 13: Script customizável do *Advanced Feeder* para produzir peças aleatórias.



Fonte: Autor

4.2.6 *Split Conveyors*

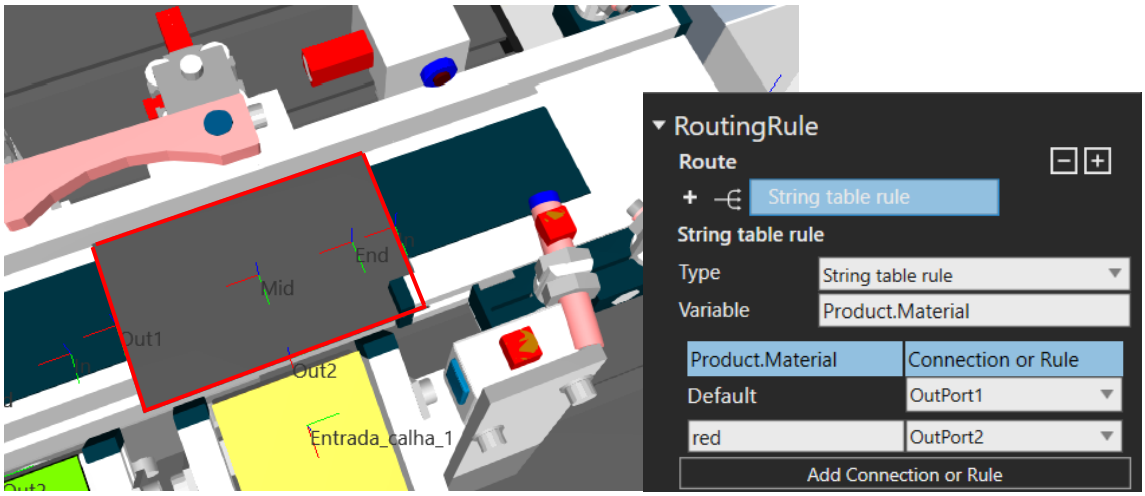
Analisando todas as configurações realizadas até o momento, o sistema, em tese, seria capaz reproduzir o funcionamento da estação real, como por exemplo utilizar os servos

motores para enviar as peças para suas unidades de armazenamento. No entanto, estes componentes mecânicos não conseguem agir sobre outros materiais no ambiente de desenvolvimento, ou seja, ele acaba por ser apenas um ícone visual que avança e retrai em determinados momentos, mas não impede que a peça continue fluindo sobre a esteira principal.

Para que a simulação possa separar essas peças junto à atuação dos braços mecânicos, foi necessário utilizar outro componente *default* da biblioteca do VC denominado de *Split Conveyor*.

Os *Split Conveyors* têm a capacidade de rotear os componentes de acordo com as suas propriedades. Por exemplo, na Figura 13 nota-se que uma propriedade *Product.Material* foi atribuída a cada componente recebendo o valor de *red, grey* e *black* os quais serão utilizados como referência pelo *Split Conveyor* para definir qual a "rota" cada um deve seguir, como ilustrado pela Figura 14.

Figura 14: Regras de roteamento no Split Conveyor (destacado por um retângulo vermelho) utilizando a propriedade Product.Metrial do Advanced Feeder.



Fonte: Autor

Note que as regras de roteamento possuem saídas padrões, ou seja, para a figura exemplificada que utiliza como referência a peça vermelha, caso a peça que chegue não possua o *match code Product.Material* igual a *red*, então deverá seguir pela *Out1* (esteira principal), mas caso contrário, deverá seguir pelo *Out2* fluindo pela unidade de armazenamento desejada. Para cada uma das peças foi utilizada a regra disposta na Tabela 2.

Tabela 2: Lógica de roteamento dos *Split Conveyos*

	Peça Vermelha	Peça Prata	Peça Preta
Split Conveyor 1	Saída 1	Saída 2	Saída 2
Split Conveyor 2	Saída 2	Saída 1	Saída 2
Split Conveyor 3	Saída 2	Saída 2	Saída 1

Fonte: Autor

4.2.7 *Sink Process*

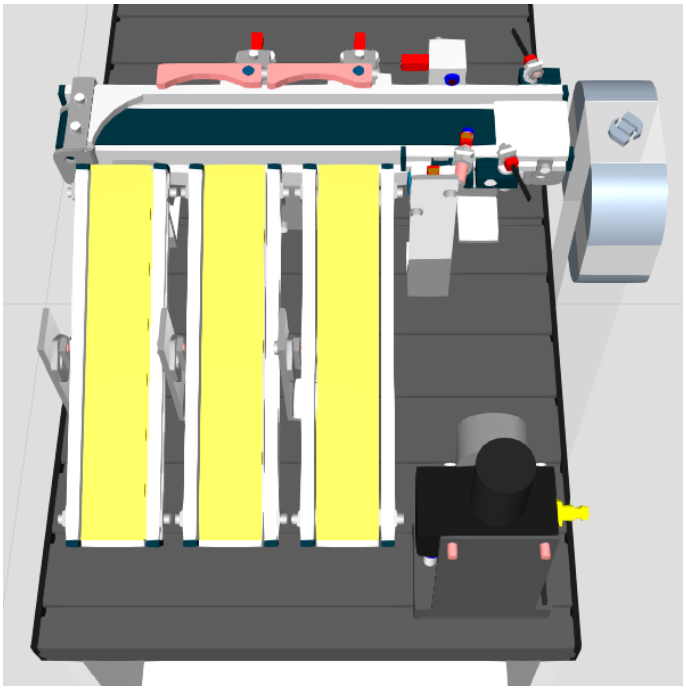
As esteiras modeladas anteriormente permitem apenas que os produtos fluam por elas, mas não possibilitam que os mesmos se acumulem ao atingir o final do seu processo. Nesse caso, nunca seria possível verificar quantas peças foram produzidas ou quantas estão armazenadas. Esse tipo de comportamento poderia afetar uma linha de produção real, pois, caso as calhas tenham limites de armazenamento de produtos, isso não seria visível e representaria uma falsa visão da realidade, podendo causar acúmulo de produtos além de prejudicar a linha de produção como um todo.

Para resolver esse problema, uma opção é utilizar um componente da biblioteca do VC denominada *Sink Process,* o qual possibilita parar os produtos ao atingirem o final do processo. Na **Figura 5**, o *Sink* é representado pelos ícones em "x" no final de cada esteira. Dessa forma, conforme as peças são geradas pelo *Feeder*, elas ficam acumuladas formando um fila em cada uma das unidades de processamento.

Por fim, para ter uma simulação mais clara, após testar o funcionamento dos componentes e sua modelagem, é possível deixá-los invisível, como é o caso dos *Splits Conveyors* e dos *Sink Process*, resultando em uma bancada representada pela Figura 15.

Com os componentes da bancada totalmente modelados, deve-se ressaltar a necessidade de adicionar *Behaviors* de sinais booleanos para cada um deles. Esses comportamentos serão utilizados para receber sinais enviados pelo Simatic e executar ações definidas através de um script em *Python*. Assim, além de adicionados para cada componentes, foram renomeados de acordo com sua função na bancada, como por exemplo "Ativa_Servo_1", responsável por sinalizar se o braço mecânico 1 está sendo acionado devido a chegada de uma peça vermelha à estação.

Figura 15: Bancada final com os componentes totalmente modelados.



Fonte: Autor

4.2.8 Programação do Modelo em Python

Com os "comportamentos" adicionados a todos os componentes foi possível iniciar o desenvolvimento da programação em *Pyhton*. Diferentemente da modelagem para os componentes individuais, o *behavior* "*Python Script*" deve estar contido na própria bancada, com o objetivo de conseguir processar os dados de todos os componentes. Caso fosse adicionado dentro de um componente específico, a leitura de dados ficaria restrita. Vale destacar que a programação em *python* será utilizada como um meio de traduzir os sinais vindos do Simatic Step 7 e converte-los em movimentos dos componentes.

Vejamos o exemplo da Figura 16. O código *python* precisa de um "caminho" para acessar o funcionamento de outros itens. Para isso, deve-se utilizar um padrão na inicialização do código como *findComponent e findBehaviour*. Note que o sinal será encontrado apenas se todos os sinais booleanos foram adicionados aos seus respetivos componentes. Para exemplificar, no item referente ao Servo Motor 1, um comportamento do tipo booleano Ativa_Servo_1 irá retornar um valor à variável declarada em python denominada ativa_servo1.

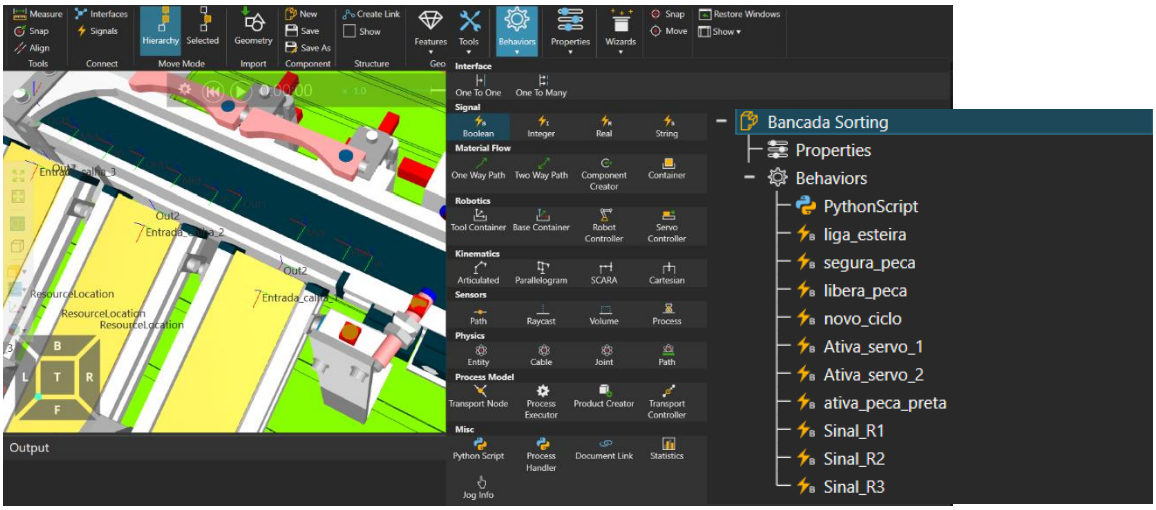
Figura 16: Definição padrão para acessar os sinais booleanos das variáveis no VC.

```
from vcScript import *


app = getApplication()
sim = app.Simulation
comp = getComponent()

#Inicia o funcionamento do sistema
peca_na_esteira=app.findComponent("Conveyor").findBehaviour("liga_esteira") #Verifica se há peça presente na estação e se a esteira deve ser ligada.

#Processo que vai ativar ou desativar o pino de acordo com a presença da peça
segura_pino= app.findComponent("Pino").findBehaviour("segura_peca") #Avança o pino para permitir a leitura dos sensores
libera_pino= app.findComponent("Pino").findBehaviour("libera_peca") #Recua o pino depois que a estação verifica qual a cor da peça
reset_vermelho=app.findComponent("Calha_1").findBehaviour("Sinal_R1") #Verifica se a quantidade máxima de peças vermelhas foi atingida
reset_prata=app.findComponent("Calha_2").findBehaviour("Sinal_R2") #Verifica se a quantidade máxima de peças pratas foi atingida
reset_preto=app.findComponent("Calha_3").findBehaviour("Sinal_R3") #Verifica se a quantidade máxima de peças pretas foi atingida

#Processo que vai ativar o primeiro, segundo ou nenhum servo de acordo com a cor da peça que for selecionada pelos sensores iniciais
ativa_servo1=app.findComponent("Servo_Motor_1").findBehaviour("Ativa_servo_1") #Ativa o primeiro servo se a peça for vermelha
ativa_servo2=app.findComponent("Servo_Motor_2").findBehaviour("Ativa_servo_2") #Ativa o segundo servo se a peça for prata
peca_preta_ativa=app.findComponent("Conveyor").findBehaviour("ativa_peca_preta") #Não ativa nenhum servo se a peça for preta
servo1 = app.findComponent("Servo_Motor_1").findBehaviour("Servo Controller_1") #Verifica o comportamento do Servo Motor 1 - definido como junta rotacional
servo2 = app.findComponent("Servo_Motor_2").findBehaviour("Servo Controller_2") #Verifica o comportamento do Servo Motor 2 - definido como junta rotacional
parada= app.findComponent("Pino").findBehaviour("Servo Controller_Pino") #Verifica o comportamento do Pino/Trava - definido como junta translacional

#Processo que permite definir a velocidade da esteira, seja ela mínima (0) ou máxima (200) permitindo a passagem da peça.
path=app.findComponent("Conveyor").findBehaviour("Path_HIDE_")
Speed = app.findComponent("Conveyor").getProperty('ConveyorSpeed')
```

Fonte: Autor

Uma vez declaras no script, os mesmos comportamentos dos sinais booleanos de cada variável devem ser adicionados à bancada geral, como ilustra a Figura 17, com o objetivo de garantir a leitura correta dos sinais.

Figura 17: Modelagem da bancada para acessar os sinais booleanos dos componentes.



Fonte: Autor

A partir desse momento, o restante da programação foi realizada pensando no funcionamento da bancada como um todo, como por exemplo, recuo dos servos motores e do pino às suas respectivas posições inicias; chamadas de funções para realizar determinadas ações; adição de laços condicionais utilizando o sinal booleano como condição (*True* ou *False*), entre outros.

No caso de referenciar-se aos valores booleanos como condicional, é necessário utilizar, por exemplo, o *nome_da_variável.Value == True/False*. Caso contrário, o VC não conseguirá acessar os valores e a programação não terá ação sobre a simulação.

4.2.9 Integração com o Simatic Step 7

Com o objetivo de validar as modelagens realizadas e a programação em python, é necessário realizar a integração com o Simatic Step 7. De forma simplificada, o VC, a partir de uma conexão via TCP/IP, deve ser capaz de receber todos os sinais tanto de entrada quanto de saída emitidos pelo Simatic e representar visualmente o funcionamento da bancada. Para isso, é necessário conectar ambos os softwares através de configurações internas do VC e através do NetToPCLSim.

Inicialmente, acessando a aba de "Conectividade" do Visual Components, é possível escolher o tipo de PLC e o protocolo de comunicação a ser utilizado. Uma vez que o controlador é da Siemens, foi selecionada a opção de conectividade com o Siemens S7 e adicionado as configurações compatíveis com o CLP, ou seja, endereço do IP sendo o mesmo da rede e os campos de Rack e Slot definidos como 0/2.

Note que o IP deve ser o mesmo utilizado no campo *Netword Address* no NetToPLCSim para que uma conexão seja estabelecida, como mostra a Figura 18. Note que para realizar a configuração os ícones referentes ao servidor (*Server to Simulation* e *Simulation do Server*) devem estar inativos.

Em seguida, para testar a comunicação, deve-se inicializar o servidor no NetToPLCSim com as configurações detalhadas na seção 3.2.2 NetToPLCSim e em seguida acionar os ícones referentes ao servidor no VC. No entanto, o ícone Simulation to server deve permanecer inativo, pois deseja-se que a transmissão de dados seja realizada do Simatic (*Server*) para a simulação, e não o inverso.

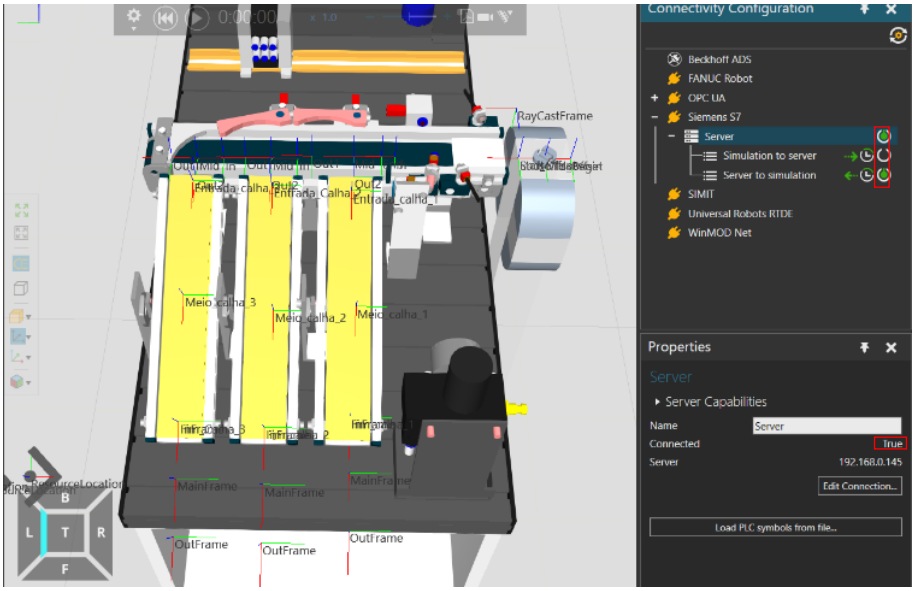
Por fim, ao selecionar o botão *Test connection* uma mensagem deve aparecer na tela indicando uma conexão bem sucedida, e o status do servidor passará para *True*, como indicado na Figura 19.

Figura 18: Configuração da conectividade entre VC e Simatic.

Fonte: Autor

Com a conexão estabelecida, é necessário conectar as variáveis de ambos ambientes de desenvolvimento.

Figura 19: Acionamento da simulação e conexão com o Simatic Step 7.



Fonte: Autor

Para isso, foi utilizado um modelo padrão em formato excel (xlsx), como mostra a Tabela 3, que deverá ser importado no VC selecionando a opção *Load PLC symbols from file...* A tabela no excel deve conter as quatro colunas indicadas de forma e os endereços

lógicos devem ser declarados utilizando sempre o símbolo % antes do endereço Yx.x, sendo Y uma entrada ou saída (I/Q) e x um número de 0 a 9.

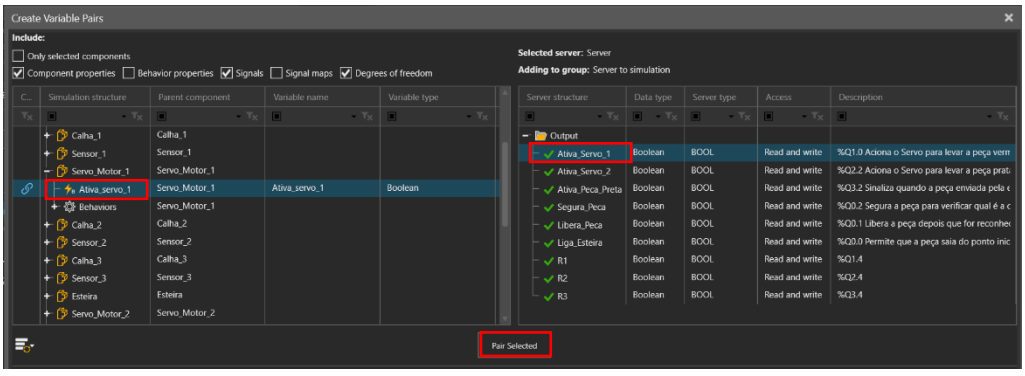
Tabela 3: Tabela de símbolos importada no VC.

NAME	PATH	DATA TYPE	LOGICAL ADDRESS
ATIVA_SERVO_1	Default tag table	BOOL	%Q1.0
ATIVA_SERVO_2	Default tag table	BOOL	%Q2.2
ATIVA_PECA_PRETA	Default tag table	BOOL	%Q3.2
SEGURA_PECA	Default tag table	BOOL	%Q0.2
LIBERA_PECA	Default tag table	BOOL	%Q0.1
LIGA_ESTEIRA	Default tag table	BOOL	%Q0.0
R1	Default tag table	BOOL	%Q1.4
R2	Default tag table	BOOL	%Q2.4
R3	Default tag table	BOOL	%Q3.4

Fonte: Autor

Com a tabela de símbolos adicionada, as variáveis devem ser conectadas a partir da aba *Connectivity* do VC, selecionando a opção *Server to Simulation* e em seguida *Add variables*, localizado na barra superior de comandos. Em seguida, as variáveis do ambiente do Visual Components (à esquerda) foram conectados às variáveis do Simatic (à direita). Para isto, basta clicar em ambas as variáveis de interesse e selecionar *Pair Selected*, como ilustrado na Figura 20.

Figura 20: Inserindo e conectando as variáveis do VC com o Simatic



Fonte: Autor

Esse processo foi realizado para todas as variáveis, obtendo a relação ilustrada na Figura 21. A partir desse momento, toda vez que uma variável no Simatic foi acionada ou

desligada, um espelho do processo será visualizado no VC, alterando as variáveis de *False/True* e vice-versa.

Figura 21: Variáveis do VC e do Simatic conectadas a partir de uma tabela de símbolos importadas pela aba "Connectivity".

Fonte: Autor

Com esse passo finalizado, o *pyhton Script* conseguirá acessar os valores booleanos em tempo real do software Simatic, completando a integração entre os ambientes de desenvolvimento, permitindo a aplicação e teste do sistema como um todo.

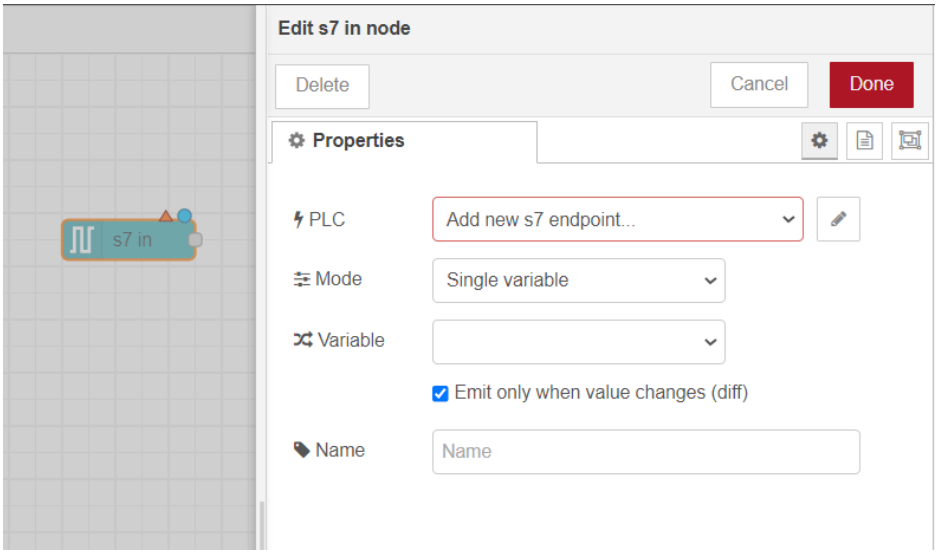
4.3 Etapa 3: Desenvolvimento do Monitoramento e Supervisão

Quando se pensa em um sistema interativo, deseja-se ter um ambiente no qual é possível verificar o funcionamento de uma cadeia produtiva e ao mesmo tempo realizar comandos virtualmente, permitindo escrever no controlador dados de entrada. Dessa forma, na terceira etapa do desenvolvimento do projeto, foi realizada uma dashboard no Node-RED integrada junto ao Simatic.

4.3.1 Configuração do Node-Red

Com o Node-RED configurado como explicado na seção 3.2.4 Node-RED foi possível iniciar a integração com o Simatic. Selecionando uma das variáveis de entrada disponibilizadas pela biblioteca *node-red-contrib-s7*, foi possível editar as configurações do nó desejado, como ilustra a figura Figura 22. Neste momento, é necessário adicionar o IP da rede e mantê-lo igual ao configurado no *NetToPLCSim*, de forma que as ferramentas consigam transferir os dados, realizando leituras e escritas no controlador.

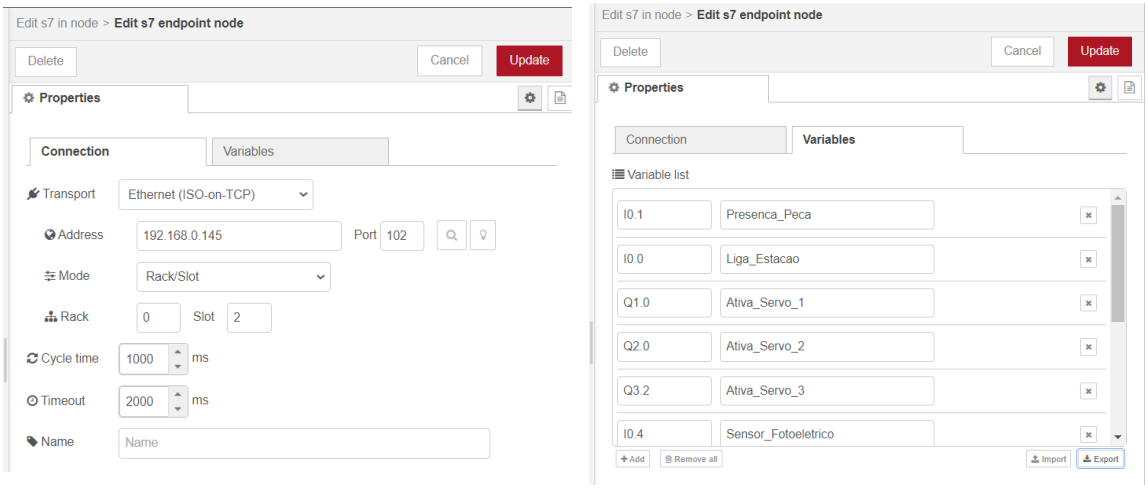
Figura 22: Configuração para conectar o Node-Red com o CLP virtual.



Fonte: Autor

Após definida a conexão, ao selecionar o lápis ao lado da configuração do CLP, foi possível escolher o tipo de comunicação a ser estabelecida, no caso Ethernet (ISO-on-TCP), indicando qual o slot e a porta a serem utilizadas. Por fim, na aba "Variables", foi possível adicionar todas as variáveis que se gostaria de monitorar, indicando o bit de entrada e de saída, como indicado na Figura 23. Note que as variáveis representadas possuem nomes similares aos atribuídos no Simatic, com o objetivo de facilitar a organização do trabalho.

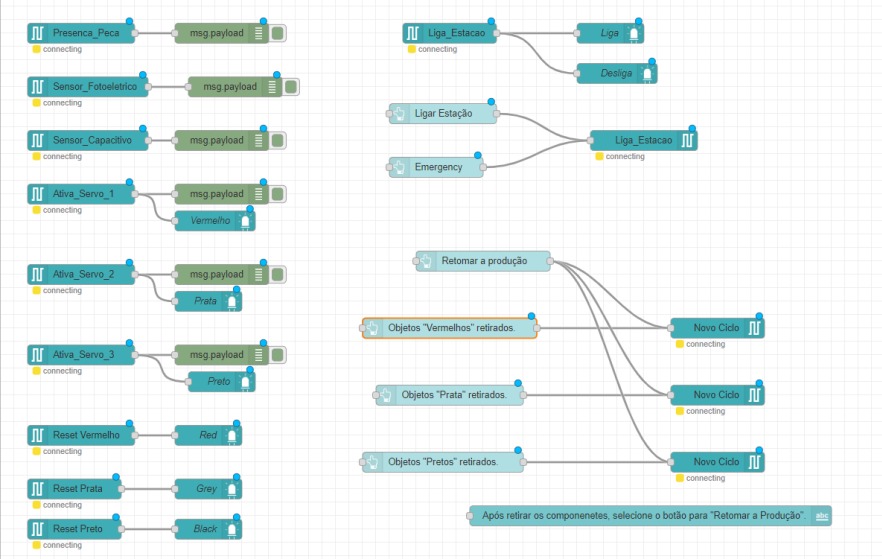
Figura 23: Configuração dos slots de conexão e adição das variáveis.



Fonte: Autor

Com essas configurações finalizadas, foi possível adicionar os restantes das variáveis no ambiente de desenvolvimento, relacionando-as com os blocos de “msg.payload”, responsáveis por sinalizar as alterações dos status das variáveis além de conectar aos ícones ilustrativos, como os leds, como ilustra a Figura 24. Note que existem dois tipos de conexões sendo realizadas. No quadrante esquerdo, as variáveis estão sendo conectadas a leds, de forma a imprimir no dashboard o bit recebido pelo controlador do Simatic. Já no quadrante à direita, as variáveis estão recebendo comandos através de botões que serão utilizados para escrever os dados no CLP.

Figura 24: Ambiente Node-Red desenvolvido.



Fonte: Autor

Por fim, para obter um ambiente organizado, esses componentes foram manipulados e configurados em *tabs* onde foi possível redimensiona-los e agrupa-los. Para visualiza-los, basta utilizar a função *Deploy* do próprio ambiente de desenvolvimento obtendo o resultado mostrado na Figura 25.

Figura 25: Dashboard no Node-Red

Fonte: Autor

Para validar então a conexão e o comportamento das variáveis, foi iniciada uma comunicação com o servidor no NetToPLCSim, verificando a atualização das variáveis na dashboard, sejam elas como dados de leitura, representadas pelos Leds ou como dados de entrada, representados pelos botões.

Por fim, basta então realizar a integração entre todos os softwares, a partir do NetToPLCSim, atingindo a quarta etapa de desenvolvimento do trabalho, onde os resultados serão apresentados na seção RESULTADOS.

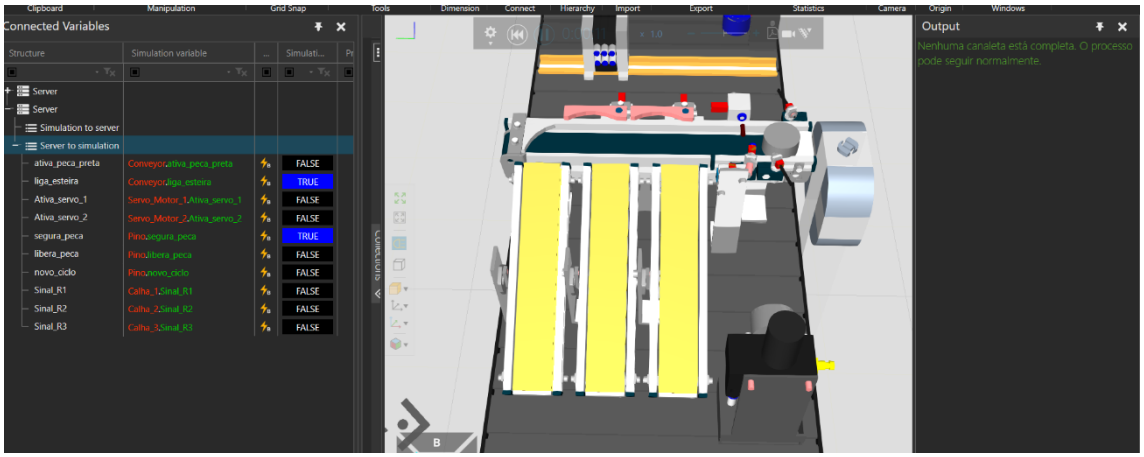
5. RESULTADOS

Com todos os softwares configurados e conectados através do NetToPLCSim, foi possível obter uma virtualização completa da bancada Sorting. Além de simular em tempo real as etapas de execução do processo de separação das peças, as ferramentas são interativas mostrando mensagens informativas, alterações dos status das variáveis *(True/False)* além de componentes visuais que permitem que o usuário acompanhe e interfira no processo de forma remota a partir de botões na própria dashboard do Node-Red.

5.1 Etapa 4: Validação do Modelo Virtualizado

Para validar o projeto foi iniciada a simulação no Visual Components obtendo o funcionamento do sistema como ilustrado pelas Figura 26 a Figura 29. A Figura 26 mostra a chegada da primeira peça à unidade de processamento, lembrando que a produção das peças é realizada de forma aleatória pelo *Advanced Feeder*. A partir desse momento, foram selecionados os bits no CLP (Figura 27) referentes à presença da peça na esteira, ligando-a e acionando o pino. Note que simultaneamente os status dos bits são alterados para *True*, e uma mensagem é emitida indicando que nenhuma unidade de armazenamento está completa e o processo pode continuar.

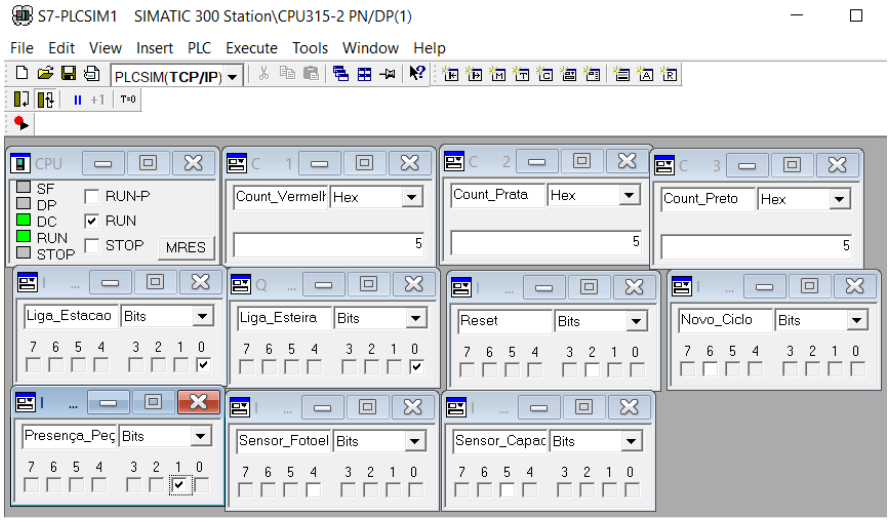
Figura 26: Chegada a peça ao sistema de seleção e acionamento dos bits no CLP.



Fonte: Autor

Como o processo acabou de iniciar, os contadores que indicam a quantidade de espaços restantes nas unidades de armazenamento ainda estão com seus valores máximos de 5 unidades.

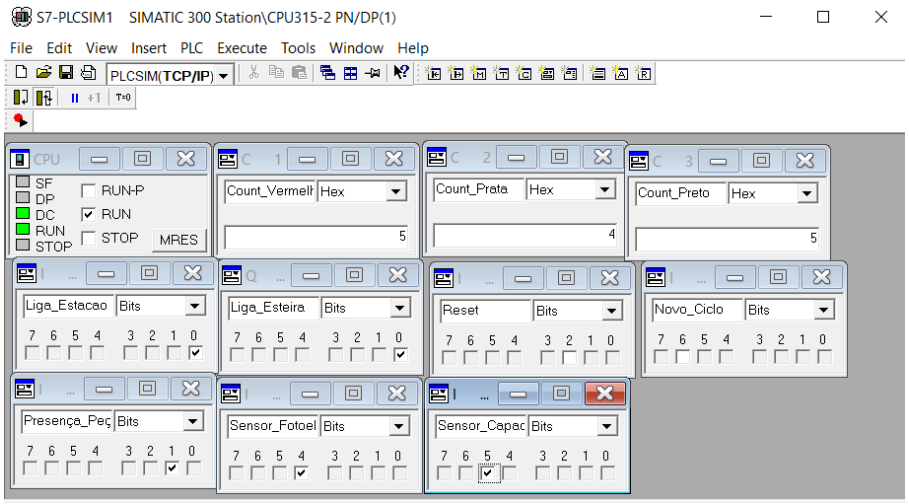
Figura 27: Acionamento dos bits no CLP de acordo com a ação ocorrida no VC.



Fonte: Autor

Como a primeira peça é de cor prata, os bits referentes aos sensores Fotoelétrico e Capacitivo foram acionados formando o par (1,1) enviando para o VC um status *True* para o sinal booleano Ativa_Servo_2 o qual irá indicar visualmente a atuação do mesmo sobre a peça, como ilustra a Figura 28.

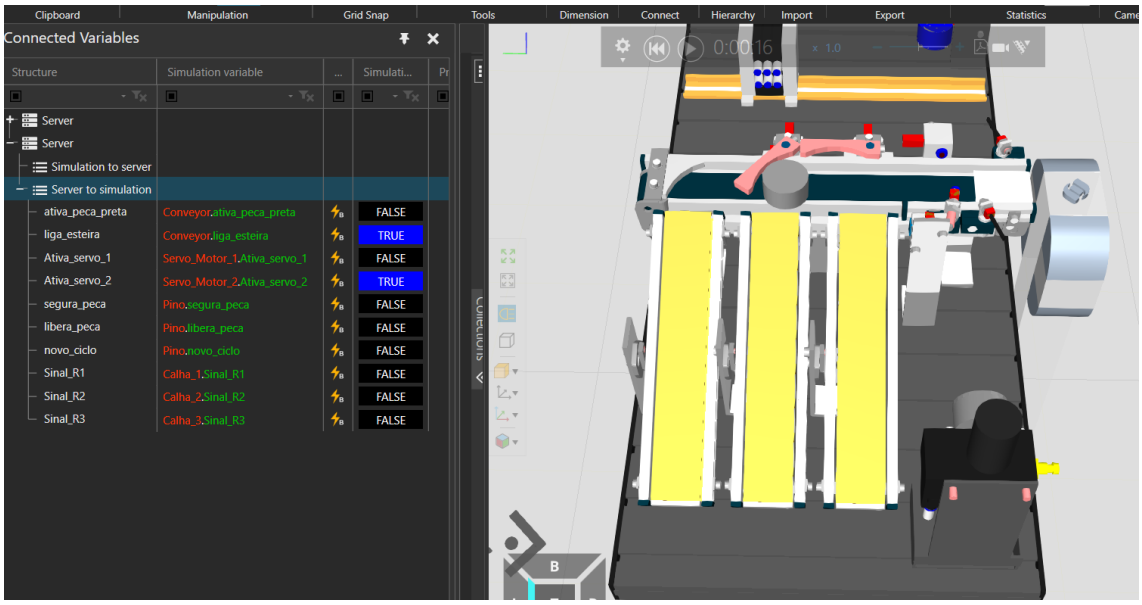
Figura 28: Reconhecimento sensorial da peça a partir do CLP virtual.



Fonte: Autor

Note que agora o bit Count_Prata foi decrescido em 1 unidade (Figura 28), o pino inicial foi recuado e seu *status* foi alterado para *False* e o braço mecânico 2 atuou para enviar a peça prata à sua respectiva unidade de armazenamento, tendo seu status alterado para *True.*

Figura 29: Separação da peça prata utilizando o servo motor 2.



Fonte: Autor

Com isso, todo o processo pode ser acompanhado via Node-Red, como ilustra a Figura 30. Neste caso é possível verificar que a estação está ligada, que a peça a ser processada é a prata e que nenhuma unidade de armazenamento está completa, permitindo a sequência do processamento.

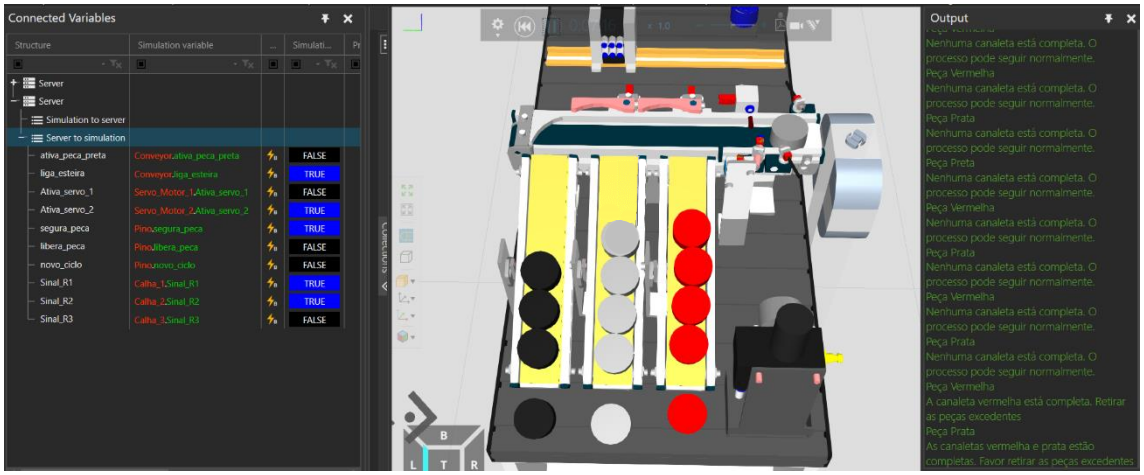
Figura 30: Dashboard de acionamento no Node-Red.

Fonte: Autor

Por fim, testou-se o funcionamento para caso alguma das unidades de armazenamento estivessem cheias não liberando a passagem da respectiva peça. As Figura 31 e Figura 32 ilustram essa situação. Note, agora, que ambas as canaletas das peças vermelha e prata estão cheias e dessa forma se qualquer uma das duas cores chegarem ao sistema, nenhuma poderá seguir. O VC então emite mensagens de que o sistema está saturado e que as peças excedentes devem ser retiradas, não permitindo a passagem a peça prata, no caso. Vale

ressaltar também que os status dos bits de capacidade máxima R1 e R2 estão ativados, sendo mais um indicativo que o sistema está completo.

Figura 31: Sistema saturado com duas unidades de armazenamento completas.



Fonte: Autor

Por fim, o Node-Red indica este saturamento do sistema acendendo dois leds da cor vermelha para indicar a necessidade de retirada das peças, como ilustrado pela Figura 32.

Figura 32: Dashboard no Node-Red indicando o saturamento do sistema.

Fonte: Autor

Dessa forma, o sistema foi testado como um todo, analisando as possibilidades de chegada de peças de forma aleatória assim como a atuação dos componentes, tanto no ambiente de desenvolvimento da programação em Ladder quanto nos ambientes de visualização, obtendo os resultados esperados definidos inicialmente no projeto.

5.1.1 Vídeo de Demonstração de Operação do Modelo Virtualizado

Finalmente, um vídeo do funcionamento do sistema foi gravado e está disponível através do link https://youtu.be/6exv_E5dHTU. No vídeo é possível acompanhar o acionamento das variáveis no CLP além de acompanhar as alterações em tempo real no Visual Components e no Node-Red. Neste mesmo link, é possível acessar um vídeo mais detalhado sobre o funcionamento do sistema, mostrando as configurações em cada um dos softwares.

6. CONSIDERAÇÕES FINAIS

O trabalho, em sua totalidade, funcionou de acordo com os objetivos traçados incialmente. No entanto, durante o desenvolvimento surgiram algumas dificuldades de implementação e conexão entre os softwares.

Mesmo com a robustez do Visual Components apresentada nas seções anteriores, o software possui algumas deficiências que acabaram por dificultar o desenvolvimento e andamento do projeto. Apesar de ser possível importar o modelo para o ambiente, as modelagens dos componentes e os artifícios para configurá-los, como a inserção de "Comportamentos" não são intuitivos. Por ser um software relativamente novo no mercado, ainda não existem muitos trabalhos desenvolvidos, então foi necessário realizar pesquisas ou publicar as dificuldades em fóruns de discussão do próprio VC.

Por ser utilizado a partir de uma licença flutuante, houveram períodos de instabilidade de conexão, o que acabou por interferir no tempo de comunicação entre o próprio software e o Simatic, resultando em falhas na execução do código em Python e levando a interrupção do processamento do VC, que era fechado automaticamente pois o software não estava respondendo. Talvez se fosse utilizado um outro tipo de protocolo de comunicação ou outro ambiente de desenvolvimento de programação, a comunicação teria sido mais efetiva, pois o Simatic necessita de um outro software, o NetToPLCSim, para conseguir se conectar ao VC, o que pode ter complicado a efetividade da comunicação.

Ademais, foram encontradas também algumas dificuldades no momento de acessar as variáveis através do código Python. Foi necessário então buscar artifícios para que o código pudesse retornar estes valores, e para isso, todas as varáveis que deveriam ser monitoradas foram adicionadas na configuração da bancada, como mostrado na Figura 17. Por fim, o ambiente de desenvolvimento da programação também requiriu uma atenção maior quanto a identação dos laços condicionais, pois a *IDE* não indica se existem problemas espaçamento e por vezes acabava por não realizar o comando do laço.

Apesar das dificuldades, principalmente com o VC, após totalmente configurado foi possível obter um processamento e simulação fluidos.

7. CONCLUSÃO

O objetivo inicial deste trabalho era virtualizar uma estação de uma FMS, presente no laboratório da Unesp, em Sorocaba, que tem como principal função separar as peças de acordo com suas cores em unidades de armazenamento distintas. Para isso, foi necessário realizar diversos estudos quanto às novas tecnologias da indústria 4.0, como a digitalização, comissionamento virtual e gêmeos digitais. Apesar de parecerem tópicos complexos de se implementar, as tecnologias disponíveis no mercado, a nível de software, permitem constante aprimoramento e desenvolvimento de sistemas. Uma vez que a integração do sistema foi completa e de baixo custo, o projeto poderia ser ampliado para outras aplicações, podendo ser utilizado como base para realizar a virtualização das outras estações da FMS ou ainda o aplicar em novos produtos, ambientes de produção ou sistemas.

Apesar de totalmente funcional e interativo com o usuário, o sistema poderia ser aprimorado utilizando uma relação inversa entre os componentes, ou seja, a peça ao chegar no VC seria de cor neutra e apenas quando o sinal fosse enviado pelo CLP, definindo se seria da coloração vermelha, prata ou preta, só então seria liberada adquirindo esta mesma cor. Esse tipo de funcionamento permitiria uma aproximação do conceito de gêmeos digitais pois estaria na íntegra espelhando a estação *sorting*. No entanto, devido a restrições do Visual Components não foi possível aplicar essa solução, mantendo então a comunicação e definição das cores das peças através da relação VC para Simatic, como explicado ao longo deste projeto.

Por fim, pode-se concluir que, uma vez que este trabalho foi realizado de forma remota, o comissionamento virtual é essencial para desenvolver projetos, analisar problemas, sejam eles relacionados ao funcionamento dos componentes, ou da lógica de programação ou até mesmo do tempo de comunicação entre os softwares, e conseguir atingir um modelo que possa ser monitorado e controlado à distância por qualquer usuário. Ademais, deixa aberta a possibilidade de implementação no próprio Campus da Unesp, para realizar um comissionamento real e então possibilitar a criação de gêmeos digitais, o qual poderá trazer informações sobre o comportamento da bancada, como falha de sensoriamento, liberação precoce da peça entre outros, alguns fatores que quando explorados no mundo real das indústrias e de suas de linhas de produção, poderiam acarretar em diversos prejuízos para uma empresa e que poderiam ser evitados com tantas soluções e potenciais tecnológicos da Indústria 4.0.

REFERÊNCIAS BIBLIOGRÁFICAS

BASILE F., FERRARA L; From supervisory Control to PLC code: a way to speed-up Constructive/Virtual Commissioning of Manufacturing Systems. **15th IFAC Workshop on Discrete Event Systems WODES 2020**.DOI: https://doi.org/10.1016/j.ifacol.2021.04.061. Acesso em 24 de março de 2022.

CARLSSON, H. *et al*; Methods for Reliable Simulation-Based PLC Code Verification; **IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS**, vol. 8; 2012.

CRUZ, F. *et al*, Comissionamento Virtual: Ferramenta de Validação de Programas de Controle de Sequência em Sistemas Automatizados de Manufatura; **XI Simpósio de Excelência em Gestão e Tecnologia**. Disponível em: https://www.aedb.br/seget/arquivos/artigos14/662059.pdf. Acesso em 11 abr 2022.

DELOITTE, A.G. (2015) Industry 4.0 Challenges and Solutions for the Digital Transformation and Use of Exponential Technologies. **McKinsey Global Institute**. Acesso em 24 de março de 2022.

MICROSOFT. O que é a IoT? **MICROSOFT**. Disponível em: https://azure.microsoft.com/pt-br/overview/internet-of-things-iot/what-is-the-internet-of-things/#overview. Acesso em 10 de abr de 2022.

ORACLE. O que é IoT? **ORACLE**. Disponível em: https://www.oracle.com/br/internet-of-things/what-is-iot/. Acesso em 10 abr de 2022

PORTELINHA, R.; Comissionamento virtual aplicado à automação de processos de manufatura: organização do conhecimento. Dissertação de Mestrado. **Instituto Tecnológico de Aeronáutica**. São José dos Campos, 2014. 99 pgs.

QIN, J.; LIU, Y.; GORSVENOR, R. A Categorical Framework of Manufacturing for Industry 4.0 and Beyond. **Procedia CIRP 2016**, 52, 173–178. DOI: https://doi.org/10.1016/j.procir.2016.08.005. Acesso em 23 março 2022.

SILVA, J. *et al;* Comissionamento Virtual aplicado nos Sistemas de Manufatura Atuais: Revisão da Literatura; **Instituto Federal de Educação, Ciência e Tecnologia de São Paulo**. Disponível em: https://www.theacademicsociety.net/_files/ugd/c16819_6ce08305a6b84ee2b74100ce887e0670.pdf. Acessado em 11 de abril de 2022.

TECMUNDO. O que é TCP/IP. **TECMUNDO**. Disponível em: https://www.tecmundo.com.br/o-que-e/780-o-que-e-tcp-ip-.htm. Acesso em 15 de abril de 2022.

THOR, O.; O que é um gêmeo digital? Uma representação virtual em tempo real. **IT FORUM**; Disponível em: https://itforum.com.br/noticias/o-que-e-um-gemeo-digital-uma-representacao-virtual-em-tempo-real/. Acesso em 11 de abril de 2022.

VISUAL COMPONENTS. Introducing Visual Components 4.4. **VISUAL COMPONENTS**. Disponível em: https://www.visualcomponents.com/resources/blog/introducing-visual-components-4-4/. Acesso em 05 abr 2022.

ZÄH M. F., WÜNSCH G., HENSEL, T. LINDWORSKY, A.; Use of virtual commissioning: An experiment. **ZWF Zeitscchrift fuer Wirtschaftlichen Fabrikbetrieb**; 2006.