

UNIVERSIDADE FEDERAL DE SANTA
CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE AUTOMAÇÃO E
SISTEMAS

Nicholas Wagner

**Desenvolvimento de aplicação para
equipamento de serialização em
plataforma de rastreabilidade de
medicamentos**

Florianópolis

2019

Nicholas Wagner

Desenvolvimento de aplicação para equipamento de serialização em plataforma de rastreabilidade de medicamentos

Relatório submetido à Universidade Federal de Santa Catarina como requisito para a aprovação na disciplina DAS 5511: Projeto de Fim de Curso do curso de Graduação em Engenharia de Controle e Automação. Orientador(a): Prof. Max Hering de Queiroz

Florianópolis

2019

Nicholas Wagner

Desenvolvimento de aplicação para equipamento de serialização em plataforma de rastreabilidade de medicamentos

Esta monografia foi julgada no contexto da disciplina DAS5511: Projeto de Fim de Curso e aprovada na sua forma final pelo Curso de Engenharia de Controle e Automação.

Florianópolis, 10 de Dezembro de 2019

Banca Examinadora:

David C. L. de Souza
Orientador na Empresa
Pollux Automation

Max Hering de Queiroz
Orientador no Curso
Universidade Federal de Santa Catarina

Ricardo José Rabelo
Avaliador
Universidade Federal de Santa Catarina

Vinicius Heck Peiter
Debatedor
Universidade Federal de Santa Catarina

Ricardo Fileti Marcon
Debatedor
Universidade Federal de Santa Catarina

Agradecimentos

Não posso deixar de agradecer àqueles que estiveram comigo ao longo dessa jornada. Se em várias vezes os desafios foram grandes e resisti em continuar, vocês permitiram que eu continuasse na luta.

Agradeço a Deus, primeiramente, pela sabedoria concedida e por guiar meus passos a cada novo desafio, é a Ele a quem dedico esta vitória.

Agradeço imensamente à minha família, que vibrou comigo em cada pequena conquista e me ouviu nos momentos em que não sabia a direção. Obrigado ao meu pai, meu grande incentivador; obrigado à minha mãe, minha grande conselheira e amiga, que viveu esses desafios junto comigo; e obrigado ao Thiago e à Juliana, meus grandes companheiros e amigos que estiveram ao meu lado em todos os momentos. Esta vitória é de todos nós, e sem vocês eu não teria aqui chegado.

Por fim, deixo meu agradecimento especial a todos os amigos e familiares que, de alguma forma, colaboraram ou torceram para este momento chegasse. Sou grato por tê-los ao meu lado, na certeza de que a torcida de vocês me fez chegar até aqui.

Agradeço também às equipes dos laboratórios LABMETRO e UFSCKite, pela oportunidade e aprendizado proporcionados durante os anos da graduação, bem como a toda a equipe do Departamento de Automação e Sistemas, em especial ao professor Max, o qual me orientou na realização deste trabalho.

E, à Pollux Automation e toda a sua equipe, agradeço pela oportunidade de desenvolver o meu Projeto de Fim de Curso e fazer parte de uma equipe que tanto me ensinou. Obrigado à todos pela paciência e pelos ensinamentos, em especial ao meu orientador David, que acompanhou cada passo da concretização deste projeto.

Resumo

A indústria farmacêutica brasileira estará sujeita a transformações ao longo dos próximos anos, fato que se deve à proposta de instalação do Sistema Nacional de Controle de Medicamentos (SNCM), o qual exige a gravação de informações de rastreabilidade em todos os medicamentos que circulam em território nacional. Nesse contexto, a empresa Pollux Automation desenvolveu a plataforma VPak-RT, oferecendo um conjunto de estações automáticas para a adequação da indústria à nova norma. Com o adiamento do prazo para a instalação do sistema, porém, a plataforma criada foi objeto de estudo para aplicação de melhorias. Neste Projeto de Fim de Curso (PFC), a estação de serialização, responsável pela impressão de códigos de rastreabilidade nas embalagens de medicamentos, é modelada e melhorias são propostas para a inserção de uma plataforma *web* desenvolvida pela Pollux que se adequa ao contexto da indústria 4.0. Apresenta-se por completo a mudança na estrutura hierárquica da estação, bem como os novos módulos de *software* implementados, com o objetivo de atingir o mesmo funcionamento da estação antiga, mas em uma solução não dependente de *softwares* proprietários, acessível remotamente e com interface gráfica *web* de acordo com as tecnologias atuais.

Palavras-chave: rastreabilidade farmacêutica; automação industrial; desenvolvimento *web*; indústria 4.0.

Abstract

In the coming years, the pharmaceutical industry will be subject of transformations due to the installation of the brazilian drug control system, which requires the traceability informations recorded on all medicines sold on the national territory. In this context, Pollux Automation is a company that developed the VPak-RT platform, offering several automatic machines that fulfill the new regulations proposed by ANVISA. With the recent delay in the laws to install the system, the developed system could be improved. At this PFC document the Vpak-RT serialization machine, which is responsible for printing the serialization codes at the drugs packaging, has its structure described, and some improvements are proposed to the station structure, inserting a new web framework developed by Pollux. In this document's chapters, this hierarchic change at the station modules will be presented, followed by the implementation description, resulting in a station that does not depends on proprietary softwares and is remotely accessible.

Keywords: Pharmaceutical traceability; industrial automation; web development; industry 4.0.

Lista de ilustrações

Figura 1 – Estrutura de aplicação de rastreabilidade de medicamentos	16
Figura 2 – Exemplo de código rastreável impresso em embalagem de medicamento	20
Figura 3 – Representação da estação de serialização	21
Figura 4 – Estações de agregação nas modalidades manual e semiautomática . . .	22
Figura 5 – Descrição de <i>hardware</i> da estação de serialização	23
Figura 6 – Exemplo de câmera Cognex	25
Figura 7 – Atuador pneumático para rejeição de embalagens	25
Figura 8 – Interface Humano Máquina (IHM) presente na estação de serialização .	26
Figura 9 – Computador industrial com interface gráfica de toque utilizado na estação de serialização	26
Figura 10 – Sequência de operações na estação de serialização	27
Figura 11 – Sequência de operações na estação de serialização	27
Figura 12 – Exemplo de interface gráfica de gerenciamento do Pollux DMS	29
Figura 13 – Exemplo de arquitetura de componentes na qual se insere o Pollux DMS	30
Figura 14 – Arquitetura de software atual dos componentes presentes na estação de serialização	32
Figura 15 – Proposta de arquitetura de software com inserção de PC Linux	34
Figura 16 – Arquitetura de software final, adotada buscando atender aos objetivos do projeto	36
Figura 17 – Diagrama de atividades completo do sistema	39
Figura 18 – Diagrama de atividades relativo à inicialização	41
Figura 19 – Diagrama de atividades relativo à produção	42
Figura 20 – Diagrama de atividades relativo à configuração do <i>setup</i>	44
Figura 21 – Diagrama de atividades relativo à simulação da produção em modo <i>bypass</i>	45
Figura 22 – Diagrama de atividades relativo ao teste dos componentes	45
Figura 23 – Diagrama de sequência relativo à inicialização e pré-produção	47
Figura 24 – Diagrama de sequência relativo à produção	48
Figura 25 – Estrutura de módulos de <i>software</i> da IHM e comunicação entre eles . .	50
Figura 26 – Máquina de estado para o módulo Controlador	51
Figura 27 – Máquina de estado para o módulo CLP	53
Figura 28 – Máquina de estados para o módulo de atualização dos números de série	54
Figura 29 – Máquina de estados para o módulo de comunicação do CLP com o DMS	58
Figura 30 – Exemplo de interface gráfica final desenvolvida para a nova aplicação da estação de serialização	63
Figura 31 – Exemplo de interface gráfica final desenvolvida para a nova aplicação da estação de serialização	63

Figura 32 – Exemplo de interface gráfica final desenvolvida para a nova aplicação
da estação de serialização 64

Lista de tabelas

Tabela 1 – Lista de estados para a maquina de estados da figura 26	51
Tabela 2 – Lista de transições para a maquina de estados da figura 26	52
Tabela 3 – Lista de estados para a maquina de estados da figura 26	53
Tabela 4 – Lista de transições para a maquina de estados da figura 26	54
Tabela 5 – Lista de estados para a maquina de estados da figura 26	54
Tabela 6 – Lista de transições para a maquina de estados da figura 26	55
Tabela 7 – Lista de estados para a maquina de estados da figura 26	58
Tabela 8 – Lista de transições para a maquina de estados da figura 26	59

Lista de abreviaturas e siglas

ANVISA - Agência Nacional de Vigilância Sanitária

CLP - Controlador Lógico Programável

DMS - *Digital Management Suite*

ERP - *Enterprise Resource Planning*

GTIN - *Global Trade Item Number*

HTTP - *Hypertext Transfer Protocol*

IC - *Industrial Computer*

IUM - identificador único de medicamento

OPC-UA - *Open Platform Communications - Unified Architecture*

PC - *Personal Computer*

PFC - Projeto de Fim de Curso

RDC - Resolução de Diretoria Colegiada

UML - *Unified Modelling Language*

Sumário

1	INTRODUÇÃO	15
1.1	Objetivos	17
1.2	Metodologia	17
1.3	Estrutura do documento	18
2	O SISTEMA DE SERIALIZAÇÃO E RASTREABILIDADE	19
2.1	Sistema Nacional de Controle de Medicamentos	19
2.2	O sistema V-Pak RT	20
2.3	A estação de serialização	22
2.3.1	<i>Hardware</i>	23
2.3.1.1	Estrutura mecânica ajustável	23
2.3.1.2	Esteiras e motores de passo	23
2.3.1.3	<i>Encoder</i>	24
2.3.1.4	Sensores de presença	24
2.3.1.5	Impressora	24
2.3.1.6	Câmera	24
2.3.1.7	Atuador pneumático	24
2.3.1.8	Computador industrial com interface gráfica de toque	25
2.3.2	Descrição do funcionamento	25
2.3.3	Interface Humano Máquina (IHM)	26
2.4	<i>Digital Management Suite (DMS)</i>	28
3	CONTEXTUALIZAÇÃO DO PROBLEMA	31
3.1	O problema da inserção do Pollux DMS	31
3.2	Arquitetura de software atual	32
3.3	Proposta de mudança na arquitetura de software	33
4	MODELAGEM DO SISTEMA DE CONTROLE E AUTOMAÇÃO	37
4.1	Diagramas de atividade	38
4.1.1	Inicialização	38
4.1.2	Produção	40
4.1.3	Manutenção	42
4.1.3.1	Configuração do <i>setup</i> de embalagens	43
4.1.3.2	Simulação da produção em modo <i>bypass</i>	43
4.1.3.3	Teste dos componentes	43
4.2	Diagramas de sequência	44

4.2.1	Inicialização e período pré-produção	46
4.2.2	Produção	46
5	IMPLEMENTAÇÃO DO SISTEMA DE CONTROLE E AUTOMAÇÃO	49
5.1	Interface Humano Máquina (IHM)	49
5.1.1	Controlador	50
5.1.2	Comunicação com CLP	51
5.1.3	Atualização dos números de série	52
5.1.4	Tratadores de requisições do CLP	53
5.1.5	Interface gráfica <i>web</i>	55
5.1.5.1	Tela inicial	55
5.1.5.2	Modo de produção	56
5.1.5.3	Modo de manutenção	56
5.2	Servidor central	57
5.3	Alterações no <i>software</i> do CLP	57
6	RESULTADOS E CONSIDERAÇÕES FINAIS	61
6.1	Aplicação desenvolvida	61
6.2	Conclusão	62
6.3	Próximos passos	64
6.4	Vantagens obtidas	65
6.5	Considerações Finais	65
	REFERÊNCIAS	67

1 Introdução

Atualmente, são cada vez mais frequentes os casos de falsificação e contrabando de medicamentos, por se tratarem de produtos de alto valor agregado e com pouca presença de fiscalização em inúmeros países. Segundo a Organização Mundial da Saúde, 1 a cada 10 medicamentos que circulam nos países em desenvolvimento é falsificado [1]. Esse cenário, porém, vem se alterando com o passar do tempo, visto que inúmeros países têm criado leis para a regulamentação da rastreabilidade de medicamentos em circulação.

A serialização e rastreabilidade dos produtos farmacêuticos consiste na impressão de um código único em cada embalagem, o qual deve ser lido em todos os pontos da cadeia produtiva, registrando todas as operações que ocorrem com determinado produto. Tal intervenção na cadeia produtiva dos produtos farmacêuticos dá ao consumidor a garantia de que o produto não é falsificado, além de permitir o rastreamento de cargas roubadas.

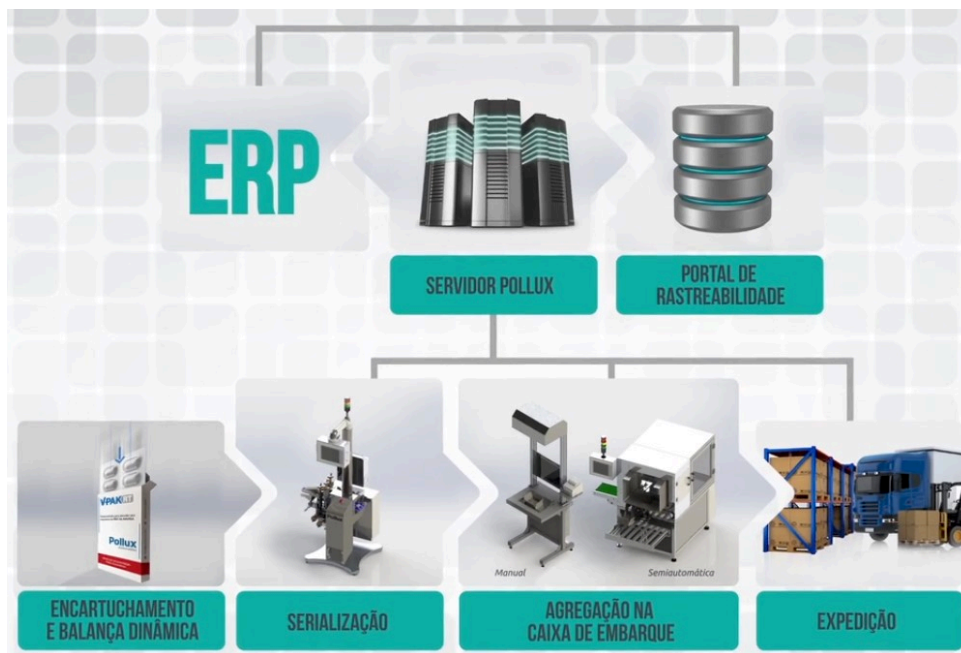
No Brasil, seguindo o movimento que tem surgido em outros países, o processo de regulamentação iniciou em 2009 e, ao longo dos anos, tem estabelecido as normas e prazos para a aplicação da mesma, apesar de já ter sofrido alguns atrasos. A primeira lei a respeito do assunto, de número 11.903/2009, teve sua regulamentação estabelecida pela primeira vez em resolução publicada pela ANVISA, a RDC nº 54/2013 e sofreu adiamento dos prazos de implantação por meio da Lei nº 13.410, de 2016 [2] [3] [4].

No contexto das soluções de serialização e rastreabilidade, insere-se a Pollux Automation, empresa sediada em Joinville que tem como objetivo desenvolver projetos capazes de aumentar a competitividade da indústria. Dentre os inúmeros trabalhos em desenvolvimento, encontra-se a criação de uma linha de produtos para a rastreabilidade de medicamentos na indústria farmacêutica. A plataforma lançada pela empresa, chamada de VPak-RT, conta com alguns módulos com diferentes estações como: serialização (impressão e conferência do código), agregação (embalagem de agrupamentos), *checkin* (entrada de carregamentos de medicamentos) e expedição. A figura 1 apresenta a estrutura completa da aplicação.

Nesse conjunto de dispositivos, a estação de serialização é responsável pela gravação dos identificadores únicos de medicamento (IUM) em cada uma das embalagens. Para isso, é dotada de um CLP, uma impressora, a qual registra os códigos dentro dos padrões a serem adotados pela ANVISA, e uma câmera, responsável pela checagem dos códigos impressos.

Após o lançamento da solução completa, porém, a regulamentação da ANVISA sofreu um adiamento do prazo para implantação em território nacional, passando de 2015 para 2021. Inúmeras inovações tecnológicas surgiram nos anos seguintes e novas soluções

Figura 1 – Estrutura de aplicação de rastreabilidade de medicamentos



Fonte: [5]

foram desenvolvidas pela Pollux Automation, como a plataforma centralizada Pollux DMS (*Digital Management Suite*). Nesse cenário, a plataforma VPak-RT foi alvo de retrabalho para atualização às novas soluções da empresa, até que a regulamentação entre em vigor, permitindo que o software anteriormente desenvolvido pudesse ser reimplementado no contexto do DMS.

Com a inserção do DMS como *framework* central da plataforma VPak-RT, o objetivo deste trabalho, é possível obter algumas vantagens: como a aplicação é executada em um sistema operacional Linux, dispõe-se de uma plataforma de código aberto robusta e com várias bibliotecas que podem ser utilizadas para aplicações conectadas à internet, o que elimina a dependência de tecnologias proprietárias e *frameworks* desenvolvidas pelos próprios fabricantes de componentes. Além disso, é possível alinhar o produto com as aplicações atuais da empresa, permitindo o suporte via acesso remoto.

Este documento, portanto, insere-se como o Projeto de Fim de Curso (PFC) desenvolvido acerca dessa plataforma, envolvendo a atualização da arquitetura de software e retrabalho em uma das estações do VPak-RT para que seu software seja baseado no Pollux DMS, a estação de serialização. São apresentadas todas as informações a respeito da plataforma existente, em conjunto com a modelagem de todo o sistema em diagramas UML e, por fim, a descrição das atividades que foram implementadas para tais adaptações.

1.1 Objetivos

Vista a necessidade de atualização da plataforma VPak-RT para as soluções mais recentes desenvolvidas pela Pollux Automation, as quais apresentam inúmeras vantagens, o objetivo geral deste trabalho é **projetar e implementar um novo sistema de controle e automação para a estação de serialização, baseado no sistema Pollux DMS**, realizando a migração do sistema implementado em uma plataforma proprietária e mantendo as mesmas funcionalidades. Alcançar esse objetivo geral será possível após atingir os seguintes objetivos específicos:

- Manter o mesmo funcionamento da estação já existente, com as comunicações entre os dispositivos ocorrendo de forma transparente ao usuário;
- Permitir o monitoramento e acesso remoto à estação, possível através da centralização das responsabilidades da mesma no dispositivo que executa o Pollux DMS;
- Operar a estação mesmo sem a conexão com o servidor central, pelo menos por um período de tempo limitado;
- Manter a coordenação com as estações anterior e subsequente, mesmo que estas não estejam atualizadas para a nova estrutura.

1.2 Metodologia

Diferente de um projeto em que o problema está em desenvolver uma aplicação por completo, aqui o problema a ser atacado é a transferência de uma aplicação existente e implementada em uma plataforma proprietária para um *software* da própria empresa, mantendo as mesmas funcionalidades e desenvolvendo uma nova interface gráfica. Por isso, a metodologia de modelagem deste segue os passos:

1. Compreensão, por parte do autor, de todas as tecnologias utilizadas pela empresa, bem como do Pollux DMS, a ser implementado na nova estação;
2. Modelagem do funcionamento da estação por meio de diagramas de atividade e de sequência, para capturar todos os fluxos de funcionamento da mesma. Tal atividade, além de realizada por meio de análise do código existente, envolverá a coleta de informações em reuniões com os desenvolvedores da primeira versão da plataforma;
3. Adaptação das funções existentes durante a implementação, buscando manter a maior quantidade de código possível da estação original, sem deixar de atender ao objetivo geral proposto para este projeto.

Para que isso seja possível, o acompanhamento das tarefas realizadas e prazos a serem cumpridos é realizado por meio da metodologia *SCRUM*, uma metodologia ágil de desenvolvimento de *software* na qual o tempo de desenvolvimento é subdividido em períodos curtos chamados de *sprints*, aos quais são associadas tarefas a serem cumpridas.

1.3 Estrutura do documento

Este documento é organizado em uma estrutura de capítulos na qual, progressivamente, o funcionamento do projeto original é apresentado, a modelagem de seu funcionamento e as técnicas utilizadas para a mesma são descritas e, por fim, as técnicas de implementação são apresentadas.

No capítulo 2, é apresentado todo o contexto da regulamentação proposta pela ANVISA para a criação do Sistema Nacional de Controle de Medicamentos (SNCM) em território nacional. Em seguida, o VPak-RT, solução desenvolvida pela Pollux para que a indústria farmacêutica se adeque à nova regulamentação, é apresentado e, em detalhes, a estação de serialização é descrita.

Em seguida, o capítulo 3 contextualiza toda a problemática a respeito da alteração na estrutura hierárquica do projeto para a inserção de um dispositivo de controle com sistema operacional Linux, capaz de executar o Pollux DMS.

No capítulo 4, então, toda a modelagem acerca do comportamento desejado na estação é descrito, resultado do conjunto de informações coletadas com base nas reuniões com os antigos desenvolvedores e análise do código já existente. Tal análise permite toda a preparação para a implementação, descrita no próximo capítulo.

O capítulo 5 descreve todas as atividades desenvolvidas e implementadas, bem como a abordagem adotada para o cumprimento dos objetivos geral e específicos propostos neste capítulo.

Por fim, no capítulo 6, são apresentados os resultados obtidos neste trabalho, quais os próximos passos para o refinamento da solução criada e as considerações finais do autor a respeito do trabalho desenvolvido e da sua relação com o curso de Engenharia de Controle e Automação.

2 O sistema de serialização e rastreabilidade

De forma a contextualizar o leitor a respeito dos tópicos abordados ao longo deste documento, este capítulo apresenta as questões referentes ao Sistema Nacional de Controle de Medicamentos, já sancionado em Lei e que deve entrar em vigor nos próximos anos, a fim de tornar rastreável toda a cadeia de produção e distribuição dos medicamentos em território nacional.

Além disso, são apresentadas as soluções da empresa Pollux Automation já existentes para a adequação das empresas ao sistema, por meio do sistema chamado VPak-RT. Dentre o conjunto de estações do mesmo, a estação de serialização será descrita em mais detalhes, pois será o problema abordado neste Projeto de Fim de Curso.

2.1 Sistema Nacional de Controle de Medicamentos

A Lei nº 13.410, de 2016, decretou o funcionamento e os prazos para implantação do Sistema Nacional de Controle de Medicamentos (SNCM) que, assim como em outros países do mundo, controlará a produção, embalagem, distribuição e comercialização de medicamentos em todo o território nacional [3].

Tal sistema de controle será implantado por meio da obrigatoriedade na rastreabilidade dos medicamentos comercializados no Brasil, a qual consiste no registro eletrônico de todas as operações realizadas sobre cada medicamento, como a embalagem, despacho, transporte e venda.

A rastreabilidade só é possível por meio do registro físico das informações de cada medicamento em sua embalagem utilizando um código identificador chamado de IUM (Identificador Único de Medicamento). Esse registro é composto, obrigatoriamente, por:

- Número de registro do medicamento junto à ANVISA;
- Número de série único, gerado por métodos não determinísticos;
- GTIN (número global do item comercial), registro global do produto apresentado como um código de barras;
- Data de validade do medicamento;
- Número do lote.

Além de registrar essas informações de forma legível na embalagem, os fabricantes devem, também, apresentá-las na forma de um código de barras bidimensional (*Datamatrix*)

Figura 2 – Exemplo de código rastreável impresso em embalagem de medicamento



Fonte: do autor

para a captura dos dados por parte dos sistemas eletrônicos. A Figura 2 apresenta um exemplo de registro das informações [4].

A norma não se aplica somente aos fabricantes, mas também aos distribuidores e varejistas, para que toda a cadeia de produção dos medicamentos possa ter seus dados armazenados e íntegros. Cada registro referente a um medicamento deverá permanecer armazenado em banco de dados por, no mínimo, um ano após o prazo de validade daquele respectivo produto.

Entre as vantagens de incluir a completa rastreabilidade da cadeia produtiva dos medicamentos em território nacional, estão:

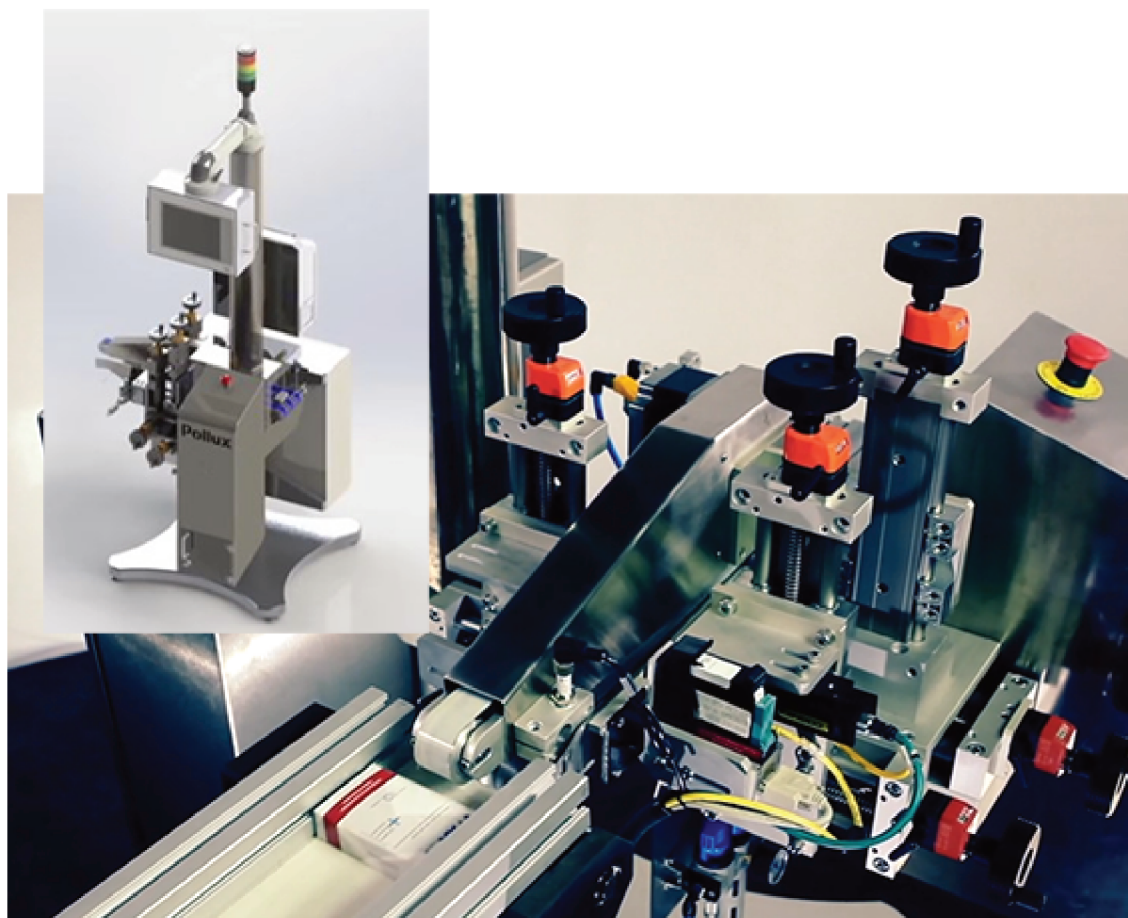
- Combate à falsificação de medicamentos, já que não será possível colocar em circulação aqueles que não possuírem devidamente registradas as informações sobre o IUM;
- Rastreamento completo de todas as movimentações na cadeia produtiva, sendo possível identificar produtos com defeito, extraviados ou incorretos;
- Maior confiabilidade do consumidor, que poderá consultar a autenticidade do produto, principalmente daqueles que possuem alto valor agregado;
- Eficiência na cadeia produtiva, por meio da análise dos dados coletados, dando uma resposta mais rápida a falhas.

Conforme instituiu a lei, a implantação do sistema será realizada após período experimental, no qual alguns lotes de medicamentos escolhidos pela ANVISA devem ter seus dados registrados e enviados para posterior análise. Após esse período, o prazo é de três anos para que o sistema entre em vigor.

2.2 O sistema V-Pak RT

No contexto da resolução RDC nº 54/2013 da Anvisa, criada com o intuito de regulamentar a rastreabilidade na fabricação de medicamentos, a empresa Pollux Automa-

Figura 3 – Representação da estação de serialização



Fonte: [5]

tion desenvolveu um produto que busca atender a tais normativas, tendo como cliente a indústria farmacêutica nacional e internacional.

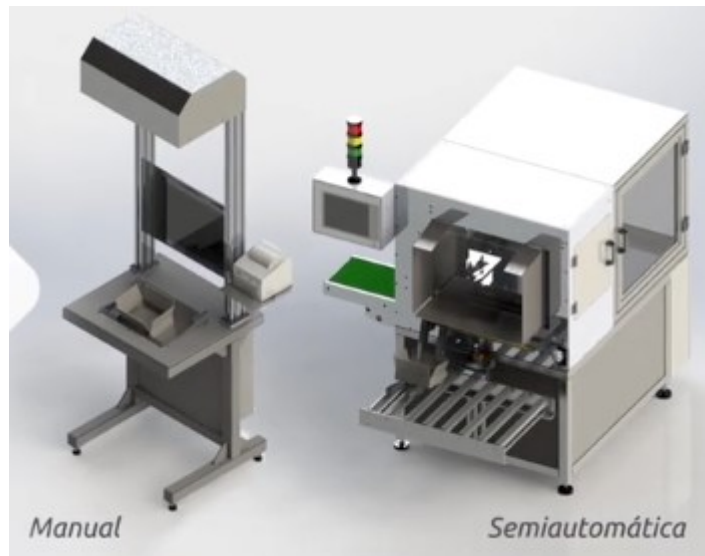
A plataforma, chamada de V-Pak RT, tem o intuito de automatizar a impressão dos identificadores de medicamentos em cada caixa e agregá-las em caixas maiores, as quais também devem ser identificadas e rastreáveis.

Para que isso seja possível, o sistema é composto por duas estações independentes: a estação de **serialização** e a estação de **agregação**.

Na primeira delas, a qual será abordada neste trabalho, os medicamentos já embalados em caixas individuais passam em alta velocidade por uma esteira, onde serão impressas todas as informações exigidas pela regulamentação da ANVISA. Além disso, uma câmera realiza a checagem da qualidade do código impresso e um ejetor de ar permite que as caixas sejam reprovadas, não avançando pela linha de produção.

Em seguida, a estação de agregação é responsável por reunir os medicamentos em caixas maiores, gerando um novo código de rastreabilidade que, por sua vez, será impresso na caixa agregadora. Essa estação existe em duas diferentes modalidades: uma

Figura 4 – Estações de agregação nas modalidades manual e semiautomática



Fonte: [5]

delas manual, onde o operador ordena os medicamentos e uma câmera faz a checagem de todos os códigos inseridos; e outra semiautomática, inserida na saída da estação de serialização, que realiza todo o processo de organização e encaixotamento com menor interferência do operador.

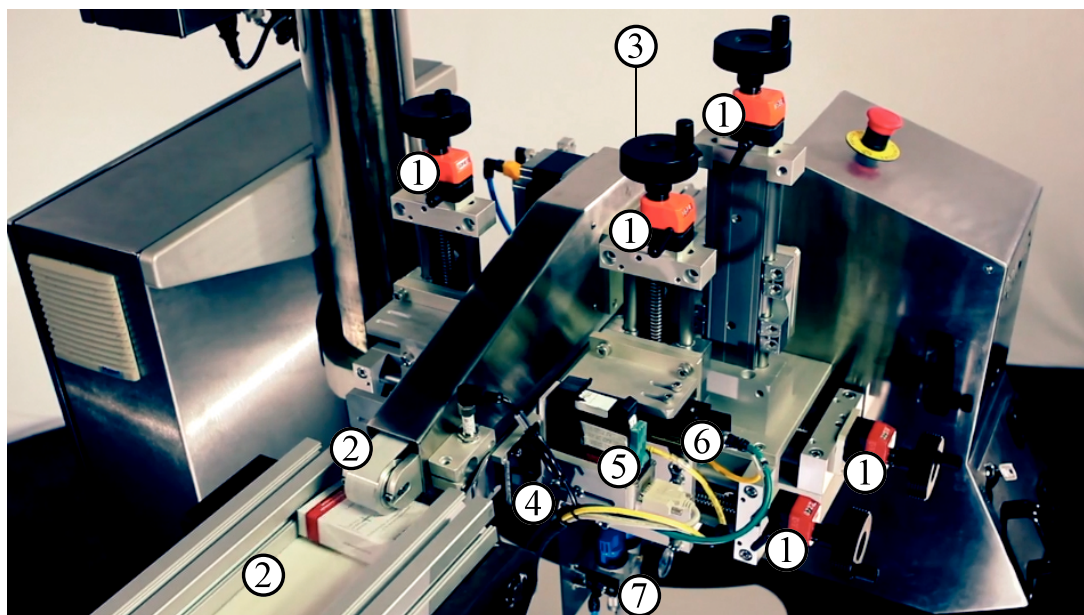
Além das estações de automatização do processo, o V-Pak RT conta com estações de controle da paletização e expedição das agregações, permitindo o completo monitoramento do processo desde a embalagem individual do medicamento até a saída dos *pallets* da indústria.

Ainda, a maioria dessas companhias do ramo farmacêutico possuem sistemas ERP (*Enterprise Resource Planning*) já instalados em sua linha de produção, os quais devem receber as informações de todos os medicamentos serializados e, também, geram os códigos e informações a serem impressos nas caixas. Por isso, o sistema da Pollux oferece a integração com esses sistemas de forma personalizada para cada um dos clientes.

2.3 A estação de serialização

Dentre as estações existentes no sistema V-Pak RT, a estação de serialização automática será descrita por completo, visto que, neste PFC, a mesma será alvo de análise para que possa ser migrado o *software* existente em plataforma proprietária para o sistema desenvolvido pela Pollux, o DMS.

Com a função de registrar os códigos em caixas de medicamentos individuais e garantir que esses códigos estejam legíveis e sem erros, a estação de serialização possui

Figura 5 – Descrição de *hardware* da estação de serialização

Fonte: [5]

diversos componentes para garantir que nenhum erro esteja presente nos medicamentos por ela processados. Para isso, alguns componentes de *hardware* são combinados, a serem descritos a seguir.

2.3.1 *Hardware*

A estação de serialização é composta pelos componentes descritos a seguir, os quais são numerados de acordo na figura 5.

2.3.1.1 Estrutura mecânica ajustável

No contexto da indústria farmacêutica, inúmeros medicamentos são produzidos em diferentes embalagens, as quais podem possuir diferentes tamanhos. Por isso, a estrutura mecânica da estação pode ser ajustada manualmente através dos fusos representados pelo número 1 na figura 5.

Cada um desses fusos possui indicador numérico de sua posição, permitindo que as configurações possam ser padronizadas quando o medicamento em produção é trocado.

2.3.1.2 Esteiras e motores de passo

Para movimentar a embalagem ao longo do dispositivo, dois motores de passo são responsáveis por mover os conjuntos de esteiras representadas pelo número 2 na figura 5, os quais estão presentes tanto na parte superior como inferior, garantindo a fixação do medicamento para o processo de impressão.

2.3.1.3 Encoder

Para o cálculo da posição de cada embalagem ao longo da estação, um *encoder* é acoplado às esteiras inferiores. O mesmo encontra-se no final da linha, representado pelo número 3 na figura 5.

2.3.1.4 Sensores de presença

Ao longo da linha, três sensores de presença são utilizados para:

- Indicar a entrada de uma embalagem na estação e calcular a sua posição a partir dali;
- Garantir que uma embalagem rejeitada não passou para a próxima estação, detectando a presença no final da linha;
- Garantir que uma embalagem descartada caiu no compartimento de rejeito, detectando a passagem pela entrada do compartimento.

O sensor presente na entrada, o qual pode ser visualizado na figura 5, é indicado pelo número 4.

2.3.1.5 Impressora

Responsável pela impressão das informações e do *datamatrix* na embalagem, a impressora possui um cabeçote de impressão no início da estação, indicado pelo número 5 na figura 5.

A estação dispõe de uma impressora industrial responsável pelo controle das informações e impressão das mesmas no *layout* desejado.

2.3.1.6 Câmera

A checagem pela integridade do código impresso na embalagem é realizada por uma câmera da fabricante *Cognex*, a qual já possui embarcado o processamento gráfico para identificar a qualidade do código. Essa câmera é representada pelo número 6 na figura 5. A figura 6 apresenta em detalhes a câmera utilizada.

2.3.1.7 Atuador pneumático

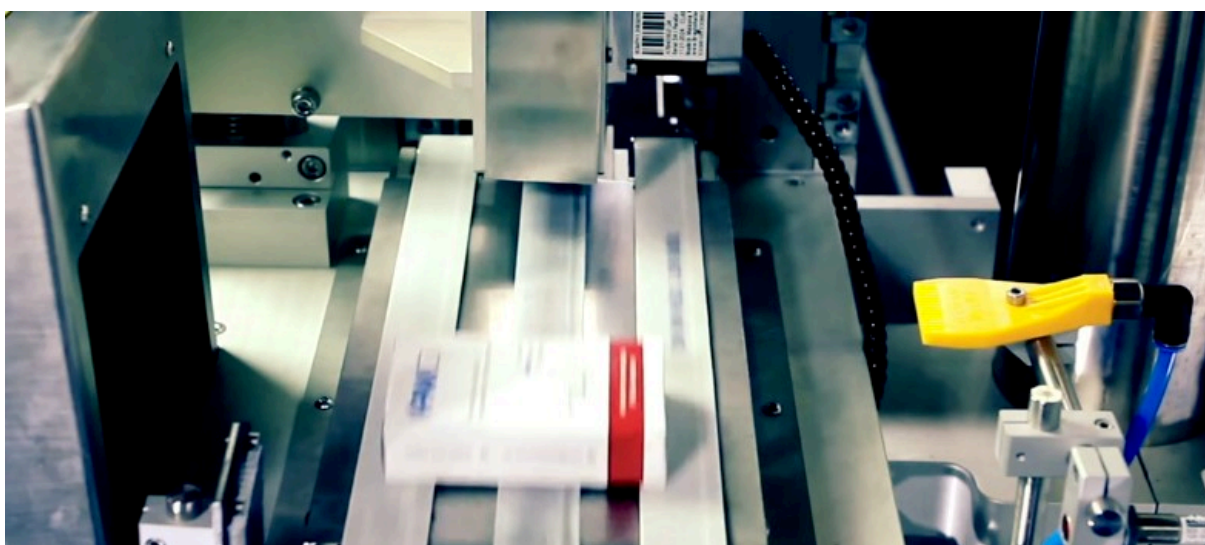
A rejeição das embalagens é realizada por um atuador pneumático, o qual empurra os medicamentos para fora do fluxo da esteira com a pressão do ar comprimido. A figura 7 apresenta esse atuador em detalhes.

Figura 6 – Exemplo de câmera Cognex



Fonte: do autor

Figura 7 – Atuador pneumático para rejeição de embalagens



Fonte: [5]

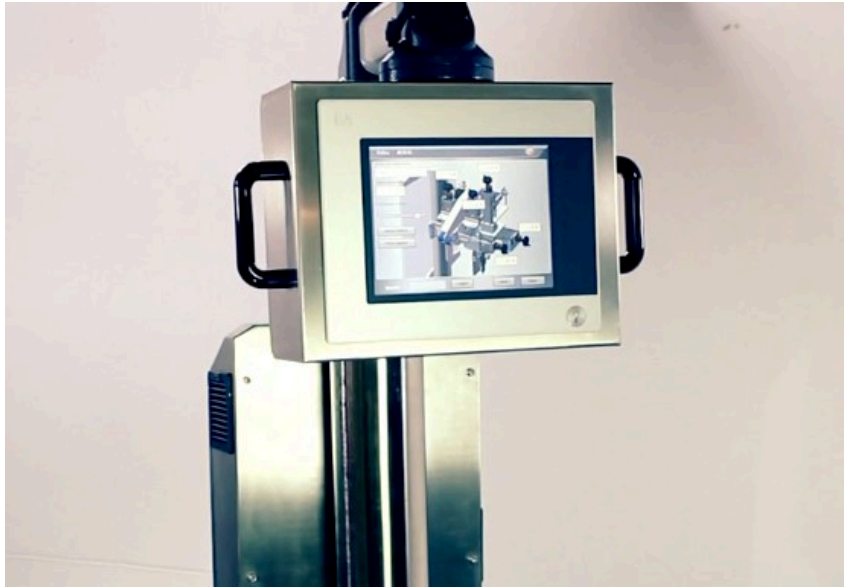
2.3.1.8 Computador industrial com interface gráfica de toque

Além desses componentes físicos, o sistema conta com uma unidade controladora industrial, responsável por gerenciar as lógicas de funcionamento da máquina e gerar a interface gráfica de toque ao operador (figura 8). Este componente é apresentado na figura 9.

2.3.2 Descrição do funcionamento

Apresentada nas figuras 10 e 11, a operação da estação de serialização inicia com a checagem da estação anterior e posterior à de serialização, garantindo que todas as esteiras se movem (1). Em seguida, quando o sensor de presença na entrada detecta a chegada de uma caixa (2), a mesma é puxada em velocidade maior que a esteira anterior, garantindo que produtos em sequência que estejam adjacentes sejam separados. Conforme

Figura 8 – Interface Humano Máquina (IHM) presente na estação de serialização



Fonte: [5]

Figura 9 – Computador industrial com interface gráfica de toque utilizado na estação de serialização



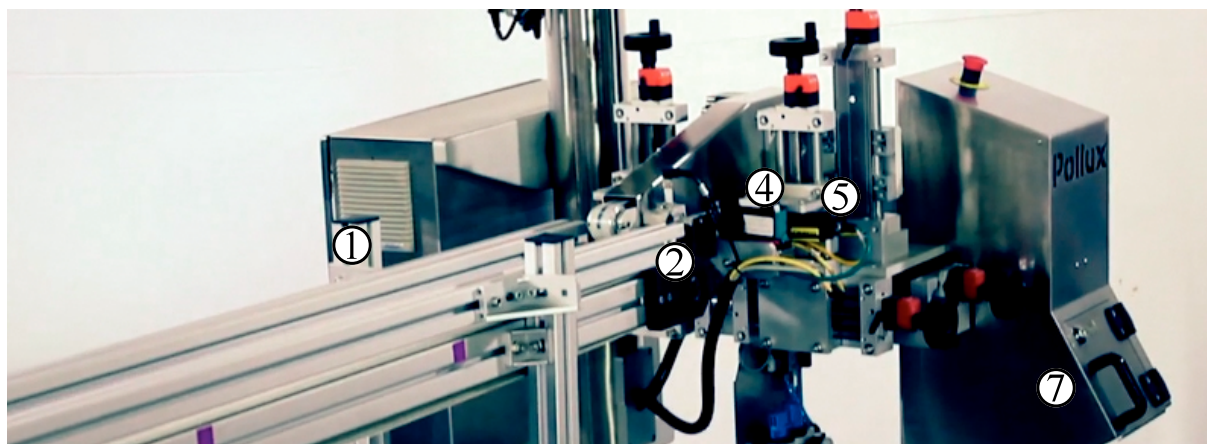
Fonte: [5]

a posição do objeto na esteira é calculada a partir de um *encoder* (3), o mesmo recebe, então, o código de rastreabilidade por meio da impressora (4), tem a impressão checada pela câmera (5) e pode ser rejeitado pelo atuador pneumático (6) em um compartimento para descarte (7). Antes de sair, outro sensor detecta se nenhum medicamento rejeitado passou para o próximo equipamento (8).

2.3.3 Interface Humano Máquina (IHM)

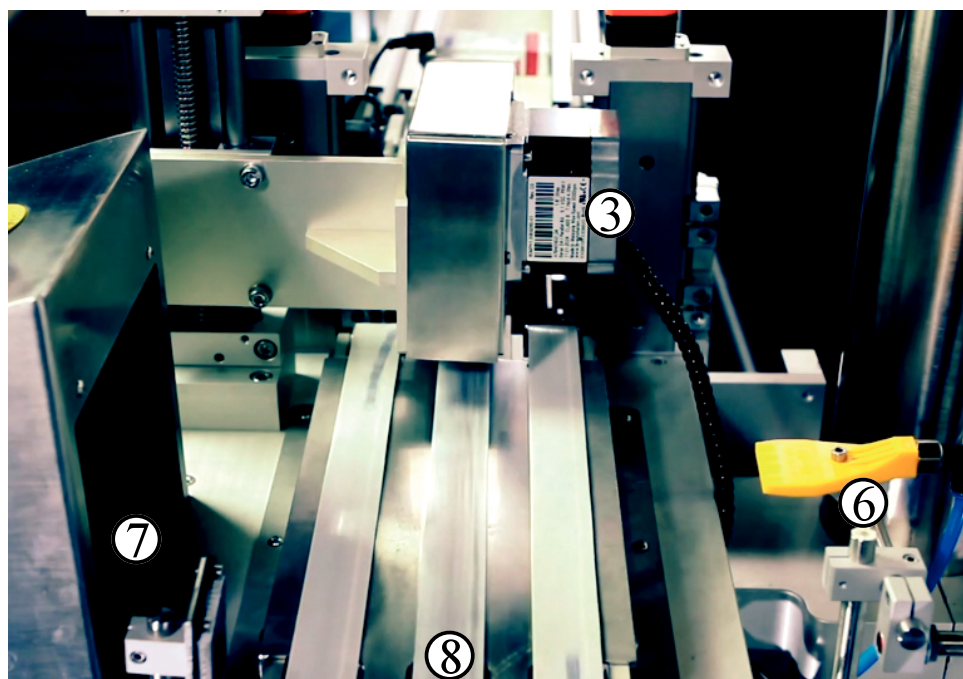
Para receber os comandos do operador, a estação contém uma IHM com toque na tela (figura 8) na qual toda a operação pode também ser monitorada. Dentre as funções desempenhadas por esse componente, estão:

Figura 10 – Sequência de operações na estação de serialização



Fonte: [5]

Figura 11 – Sequência de operações na estação de serialização



Fonte: [5]

- Início da produção, checagem das informações acerca do lote a ser produzido e configuração mecânica manual relativa ao medicamento a ser produzido;
- Autenticação do operador para autorização do início da produção, incluindo checagem *double-blind* ao iniciar um novo lote;
- Identificação e reconhecimento de alarmes ocorridos na estação;
- Acesso ao modo de testes dos componentes, sendo possível controlar cada uma das entradas e saídas manualmente;
- Acesso ao modo de produção com *bypass*, permitindo a passagem de medicamentos sem rastreabilidade, descartando-os.

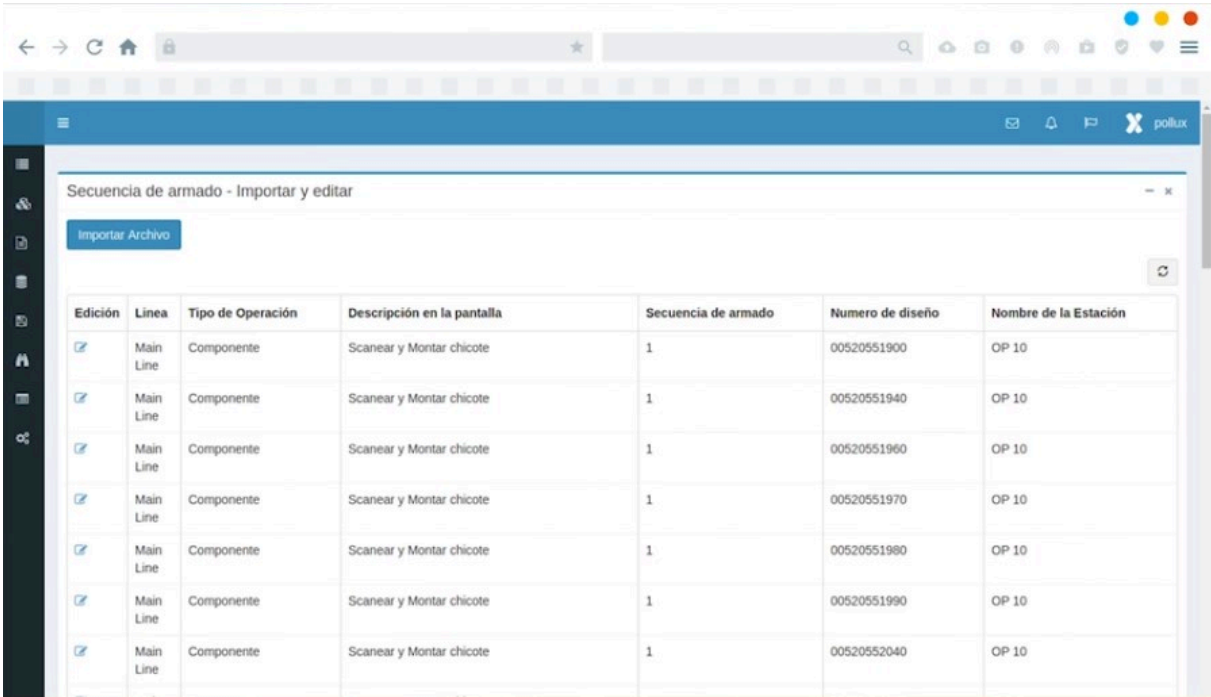
2.4 *Digital Management Suite* (DMS)

Como forma de descrever os conceitos a serem utilizados no desenvolvimento deste trabalho, esta seção tem como objetivo apresentar o *software* DMS da Pollux, o qual será utilizado nas modificações a serem realizadas na estação de serialização.

O *Digital Management Suite*, ou DMS, é um *software* criado pela Pollux que passou a ser utilizado em todas as suas soluções de automação. Desenvolvido na linguagem de programação Python, esse sistema é utilizado para a integração dos componentes industriais e apresenta as informações por meio de interface gráfica *web* ao usuário. A captura de tela mostrada na figura 12 apresenta a interface padrão de desenvolvimento oferecida pelo Pollux DMS, sendo as telas todas desenvolvidas dentro desse padrão.

De forma geral, a plataforma é uma base para o desenvolvimento de soluções completas para a indústria 4.0, funcionando como um coordenador dos diversos componentes no nível da planta, mas também comunicando-se com as soluções na nuvem por meio da internet industrial. Assim, esse sistema é capaz de ser executado tanto nos servidores de gerenciamento da indústria como nas estações individuais. Definido como uma *framework* completa para o desenvolvimento de software para a indústria, é capaz de receber funcionalidades de tratamento de dados em larga escala, modularidade e independência das estações.

O esquema da figura 13 apresenta um exemplo de arquitetura na qual se insere o Pollux DMS, para melhor visualização da sua função dentro de uma solução de automação industrial.



The screenshot displays a web browser window with the Pollux DMS interface. The main content area is titled "Secuencia de armado - Importar y editar". Below the title is a blue button labeled "Importar Archivo". A table with seven columns is shown below the button. The columns are: "Edición", "Linea", "Tipo de Operación", "Descripción en la pantalla", "Secuencia de armado", "Numero de diseño", and "Nombre de la Estación". The table contains seven rows of data, each representing an assembly sequence step.

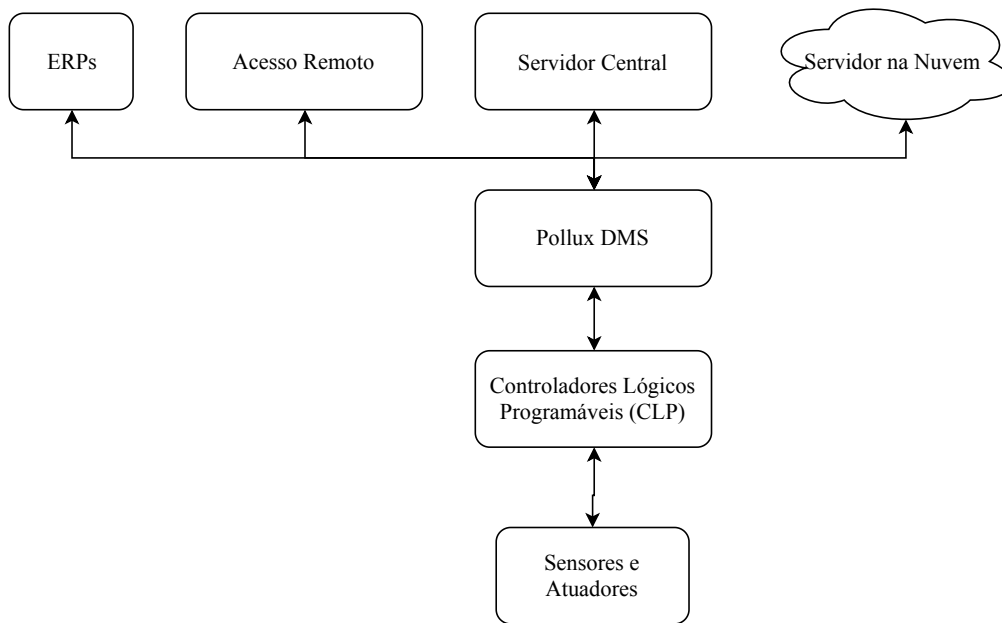
Edición	Linea	Tipo de Operación	Descripción en la pantalla	Secuencia de armado	Numero de diseño	Nombre de la Estación
<input checked="" type="checkbox"/>	Main Line	Componente	Scanear y Montar chicote	1	00520551900	OP 10
<input checked="" type="checkbox"/>	Main Line	Componente	Scanear y Montar chicote	1	00520551940	OP 10
<input checked="" type="checkbox"/>	Main Line	Componente	Scanear y Montar chicote	1	00520551960	OP 10
<input checked="" type="checkbox"/>	Main Line	Componente	Scanear y Montar chicote	1	00520551970	OP 10
<input checked="" type="checkbox"/>	Main Line	Componente	Scanear y Montar chicote	1	00520551980	OP 10
<input checked="" type="checkbox"/>	Main Line	Componente	Scanear y Montar chicote	1	00520551990	OP 10
<input checked="" type="checkbox"/>	Main Line	Componente	Scanear y Montar chicote	1	00520552040	OP 10

Fonte: Pollux Automation

Figura 12 – Exemplo de interface gráfica de gerenciamento do Pollux DMS

Fonte: do autor

Figura 13 – Exemplo de arquitetura de componentes na qual se insere o Pollux DMS



3 Contextualização do problema

Após a introdução dos conceitos gerais relativos ao funcionamento da estação de serialização, este capítulo busca a contextualização a respeito do problema existente na plataforma, a qual já está validada, porém desatualizada quanto às tecnologias desenvolvidas pela empresa Pollux. Para isso, será apresentada a estrutura hierárquica atual dos componentes presentes na estação, será justificada a necessidade de melhoria e, em seguida, apresentadas algumas abordagens possíveis para alcançar o modelo final de componentes em uma nova estação de serialização.

3.1 O problema da inserção do Pollux DMS

Nos capítulos anteriores deste documento o sistema VPak-RT foi apresentado integralmente e, por meio das informações dispostas, percebe-se que o mesmo já apresenta a robustez e validação necessários para um produto final, tanto que já atua em linhas de produção da indústria farmacêutica. A sua instalação em clientes, portanto, não requer modificações no sistema existente.

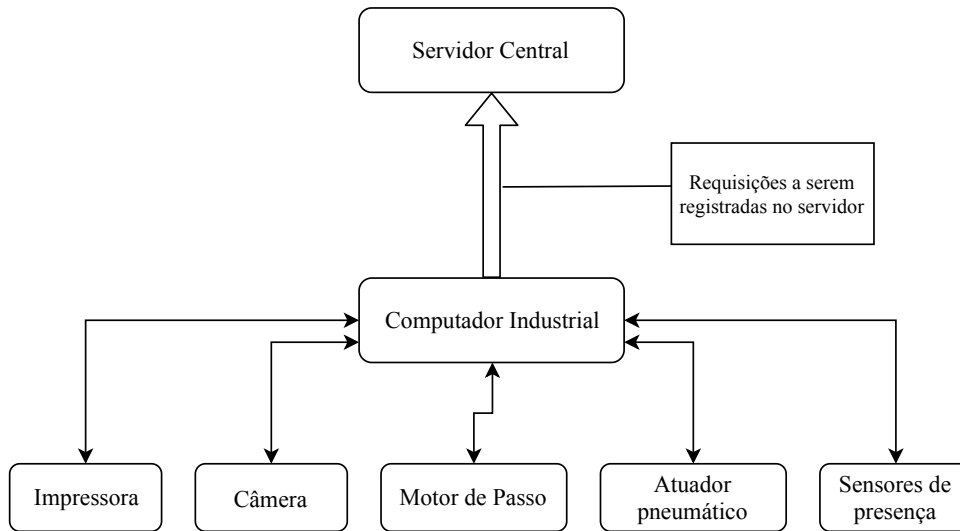
Por outro lado, é necessário destacar que essa solução foi desenvolvida no ano de 2014. Naquele contexto, as normas da ANVISA estavam próximas do prazo para serem aplicadas no mercado nacional, mas ainda sem regulamentação clara. A primeira Lei a respeito do assunto, de 2009 [2] carecia de maiores informações sobre o funcionamento do SNCM.

Por isso, o prazo inicialmente previsto para a implantação veio a ser prorrogado pelo governo federal visando a melhor especificação do funcionamento de tal sistema. Já se sabe que a regulamentação definitiva surgiu apenas em 2016, com a Lei nº 13.410 [3], concedendo à Pollux Automation um período de tempo maior para o refinamento do produto criado.

Agora, no ano de 2019, a empresa de Joinville já conta com a sua nova plataforma desenvolvida para a indústria 4.0, o Pollux DMS, o qual está presente na grande maioria de suas soluções e fornece inúmeras vantagens aos produtos que a usam, como o uso de sistemas operacionais Linux, a independência de *softwares* proprietários das fabricantes de componentes industriais, a modularidade e o acesso remoto por parte da empresa.

Por isso, viu-se necessária a reformulação da estrutura de *hardware* e, consequentemente, de *software* das estações existentes no VPak-RT, de forma que o controle de toda a estação não seja mais dependente de tecnologias proprietárias como, por exemplo, os ambientes de desenvolvimento de CLPs, mas sim da *framework* criada pela Pollux.

Figura 14 – Arquitetura de software atual dos componentes presentes na estação de serialização



Fonte: do autor

Surge, então, o problema associado à reformulação da aplicação presente em cada uma das estações, de forma a realizar as modificações citadas sem descartar o que já foi desenvolvido.

Nesse contexto, este Projeto de Fim de Curso atacará o problema da mudança na estrutura de *software* proposta para uma das estações existentes — a de serialização. Nas seções que seguem, a antiga e a nova estrutura proposta são comparadas. Dessa forma, a implementação poderá ser descrita nos capítulos subsequentes.

3.2 Arquitetura de software atual

Na sua versão atual, a estação de serialização conta com uma unidade de processamento central, responsável por todas as tarefas executadas na máquina. Apesar de definido como um IC (*Industrial Computer*), o componente possui comportamento similar ao de um CLP, executando os programas em ciclos e atualizando suas entradas e saídas ao final de cada execução. O *software* proprietário fornecido pela fabricante do componente permite programá-lo em linguagens conhecidas no meio industrial, como Ladder e C++.

Em conjunto com o módulo de entrada/saída e o *driver* dos motores de passo, o IC representa o controlador central da estação. Ele que contém todas as lógicas de execução, comunicação com o servidor, gerenciamento de alarmes, emergência e a Interface Humano Máquina. A figura 14 exemplifica a estrutura atual do sistema, indicando as comunicações entre os componentes e suas respectivas responsabilidades. Não especificados na figura, os protocolos de comunicação utilizados estão entre os mais utilizados no contexto da indústria 4.0, como as requisições HTTP (*Hypertext Transfer Protocol*), *Modbus* e *OPC-UA*.

Nesse modelo apresentado e presente atualmente na estação de serialização, não suporta-se a inserção do Pollux DMS, pois a mesma é uma aplicação com interface *web* executada na linguagem de programação Python. Por isso, propõe-se a alteração nos componentes da estação para a inserção de um sistema operacional baseado em Linux.

3.3 Proposta de mudança na arquitetura de software

Como já descrita em capítulos anteriores, a inserção de um dispositivo com sistema operacional Linux na estação, capaz de executar o Pollux DMS, traz inúmeras vantagens à solução, principalmente no contexto das soluções para a indústria 4.0. Dentre algumas das vantagens já citadas, estão a flexibilidade de uma interface *web* e a independência de *softwares* proprietários.

Por outro lado, o sistema Pollux DMS, programado em linguagem Python, não suportaria apenas a cópia de toda a lógica desenvolvida no IC, exigindo o retrabalho de toda a solução já desenvolvida e aumentando o tempo de entrega da nova estação.

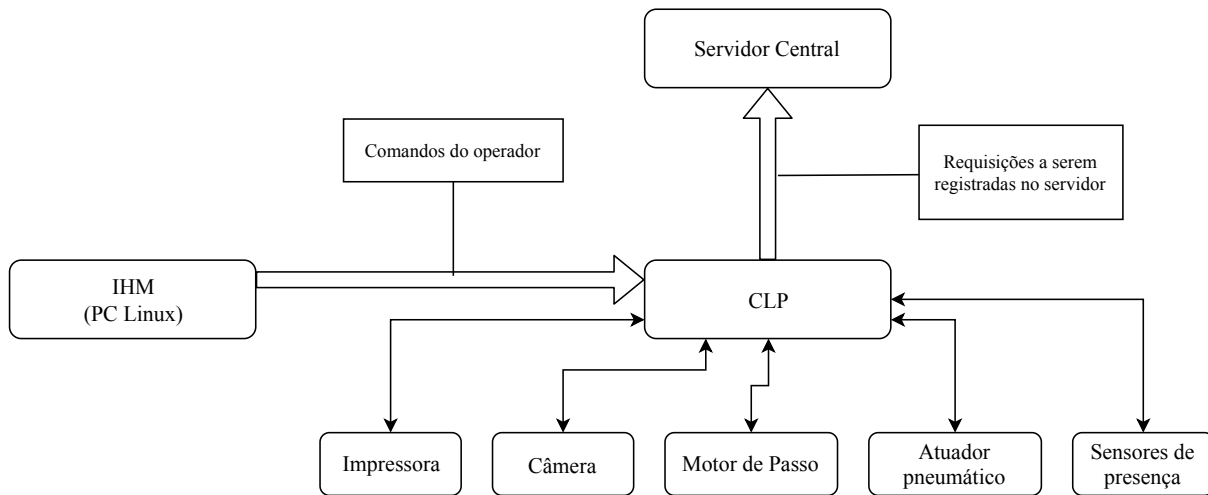
Além disso, um dispositivo como um PC, dotado de um sistema operacional Linux, não é capaz de garantir a confiabilidade e robustez que um equipamento industrial como um CLP possui. Por isso, opta-se pela troca do computador industrial por um CLP da mesma fabricante, permitindo, também, que parte da lógica já desenvolvida e validada seja reutilizada.

A nova estrutura de *hardware* proposta, em conjunto com o requisito de preservar a maior quantidade possível da solução já implementada e validada para reduzir o tempo de entrega, cria um problema quanto à responsabilidade de cada um dos novos componentes na hierarquia de controle da estação.

A fim de apresentar o progresso do desenvolvimento que levou à arquitetura final, uma primeira solução proposta, a qual não foi adotada na implementação final, é apresentada nesta seção. Nesta proposta inicial, define-se que:

- Um PC com sistema operacional Linux exercerá o papel de Interface Humano Máquina, e deverá gerenciar os comandos do operador, a autenticação do usuário e o reconhecimento de alarmes;
- O CLP preservará a lógica de comunicação com os componentes existente no IC da versão anterior da estação, enviando os comandos à impressora, câmera, esteira, etc. Além disso, realiza a comunicação com o servidor central, exigindo os dados a serem impressos, informações do lote em produção e enviando informações de *log* das ações realizadas na máquina.

Figura 15 – Proposta de arquitetura de software com inserção de PC Linux



Fonte: do autor

A figura 15 esquematiza a arquitetura de software proposta nessa estrutura e quais as comunicações que cada um dos componentes realiza, de forma que se ilustre ao leitor as conexões e mensagens trocadas entre os componentes.

O problema existente em tal estrutura, porém, é que o novo PC inserido exercerá unicamente o papel de IHM, não desempenhando as funções de um dispositivo intermediário entre CLP e servidor que possa realizar o controle e monitoramento completo da estação. Se acessado remotamente, por exemplo, o mesmo não teria acesso às comunicações realizadas com o servidor, já que as mesmas são responsabilidade do CLP.

Por isso, uma nova iteração sobre as estruturas dos componentes busca atingir os seguintes objetivos:

- Manter o controle central da aplicação, no que se refere aos comandos de início, parada, interface gráfica e comunicação com o servidor na nova IHM, para que todo o gerenciamento da operação possa ser feito de forma remota e o CLP dependa apenas da IHM para funcionar, não mais se comunicando com o servidor central;
- Manter o máximo de funções já implementadas no CLP, para que se reduza o tempo de entrega, retirando apenas as ações de gerenciamento, registro (*log*) e interpretação dos comandos do operador.

Como a retirada de toda a lógica de comunicação com o servidor existente no CLP exigiria grandes mudanças em seu *software*, a solução foi manter a mesma estrutura da comunicação realizada por requisições ao servidor, porém com as mesmas sendo enviadas à IHM e por ela tratadas. A IHM, por sua vez, é responsável por requisitar ao servidor as informações. Assim, em uma possível situação de falha temporária na rede interna, CLP e

IHM ainda podem se comunicar e manter a operação, enquanto o envio de informações ao servidor pode aguardar até que a falha seja corrigida.

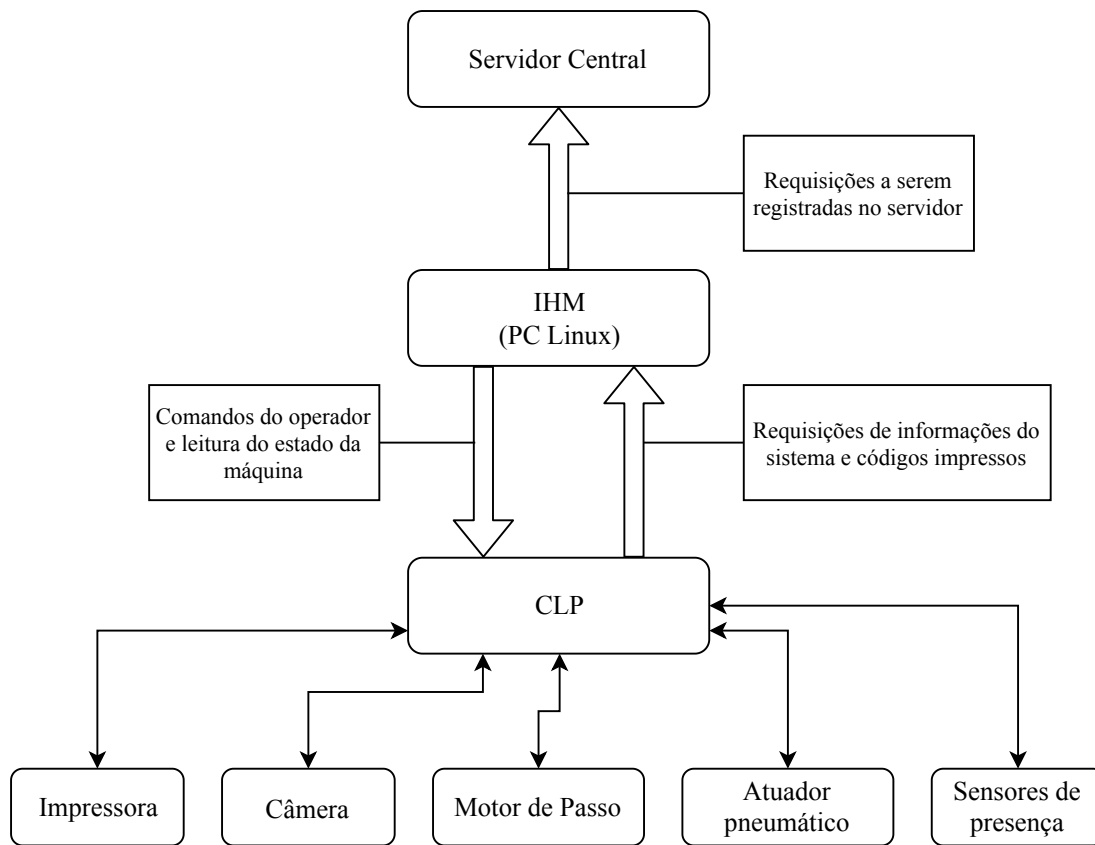
Dessa forma, a proposta de melhoria estabelece uma estrutura onde cada um dos componentes tem parte das responsabilidades no funcionamento da estação:

- O CLP exerce todo o controle e coordenação sobre os componentes físicos, sujeito aos comandos do operador executados no PC Linux;
- O CLP mantém sua lógica de comunicação com o servidor, porém passa a enviar as requisições ao PC Linux. Dentre as requisições realizadas, destacam-se: pedido de novos números de série para impressão e informações do lote atual.
- O PC Linux deve tratar as requisições vindas do CLP e é responsável por enviá-las ao servidor, quando possível e necessário. Além disso, é responsável por gerenciar todos os comandos do operador e controlar a operação do CLP.

A arquitetura descrita é esquematizada na figura 16, a fim de apresentar visualmente as comunicações e conexões entre os componentes, onde cada componente é apresentado em conjunto com as mensagens que troca.

Nesse caso, em um exemplo de acesso remoto realizado pela empresa, todas as operações ocorrendo no CLP e definidas pelo operador podem ser monitoradas, pois o controle das mesmas é de responsabilidade do PC Linux, o qual representa um novo intermediário entre o CLP e o servidor.

Figura 16 – Arquitetura de software final, adotada buscando atender aos objetivos do projeto



Fonte: do autor

4 Modelagem do sistema de controle e automação

Apresentado o contexto do problema a ser abordado neste PFC — a criação de um sistema de controle e automação responsável pelo gerenciamento da estação de serialização em um sistema operacional Linux —, este capítulo tem como objetivo apresentar toda a modelagem realizada antes da real implementação da solução. Para isso, esta fase inicial do Projeto de Fim de Curso compreendeu a elaboração de diagramas, em conjunto com a equipe de desenvolvimento, para a especificação de todas as tarefas a serem desempenhadas pelo novo sistema, bem como a definição das comunicações realizadas entre os dispositivos.

Neste capítulo, serão apresentados os diagramas de atividade a respeito do funcionamento da máquina, destacando-se o papel de cada componente por meio de cores. Em seguida, serão elaborados os diagramas de sequência, os quais permitem a separação das responsabilidades dos componentes, definição das mensagens trocadas entre eles e fornecimento da base para a implementação.

Os diagramas de atividade apresentados no início deste capítulo representam o funcionamento desejado da estação de serialização e, provavelmente, já foram elaborados anteriormente, quando a primeira versão da estação estava sendo desenvolvida. Porém, vale destacar a importância de iterar novamente na criação dos mesmos: além de descrever em detalhes todas as operações e fornecer a base para a elaboração dos diagramas de sequência, o processo foi útil para a compreensão e familiarização do autor com o funcionamento. Além disso, a alteração ou criação de novas especificações pode ser claramente definida nesta etapa do processo.

Para a elaboração dos diagramas presentes neste capítulo fez-se uso da *Unified Modelling Language* (UML), a qual colabora na especificação, visualização e documentação de *softwares* por meio de modelos abstratos e padronizados que podem ser adaptados para qualquer tipo de *software* e *hardware*, sendo amplamente utilizada em projetos de sistemas [6] [7].

Por fim, destaca-se a ausência de diagramas importantes para o desenvolvimento de software, como os diagramas de classe [7]. Como, no contexto deste PFC, foi utilizada uma metodologia de desenvolvimento ágil baseada numa *framework* que já conta com a estrutura de classes para o desenvolvimento das máquinas de estado, tais diagramas não foram necessários na modelagem da nova aplicação e, por isso, não estão incluídos neste documento.

4.1 Diagramas de atividade

A fase inicial de interpretação e modelagem do problema envolve a coleta de informações a respeito da lógica de funcionamento da estação, já que a mesma possui uma sequência de operações e a intenção é que se preserve sua lógica de negócio. Para isso, foram realizadas reuniões com os desenvolvedores da versão original a fim de coletar informações sobre a operação por meio de diagramas de atividade.

O resultado dessa fase de coleta de informações é apresentado, dentro dos padrões da linguagem UML e de forma simplificada, na figura 17. Buscando especificar melhor cada uma das partes do diagrama e evitar apenas apresentá-lo em sua forma completa, as próximas seções apresentam as funcionalidades de forma isolada e descrevem a sua respectiva operação.

O funcionamento da estação de serialização passa por um conjunto de ações na inicialização, onde são estabelecidas as conexões iniciais. Após esse período, o comando do usuário na IHM pode levar a dois fluxos de atividade, sendo eles: (a) modo de produção, que consiste na operação da estação; (b) modo de manutenção, onde o operador pode testar os componentes, definir novas configurações e simular a produção.

4.1.1 Inicialização

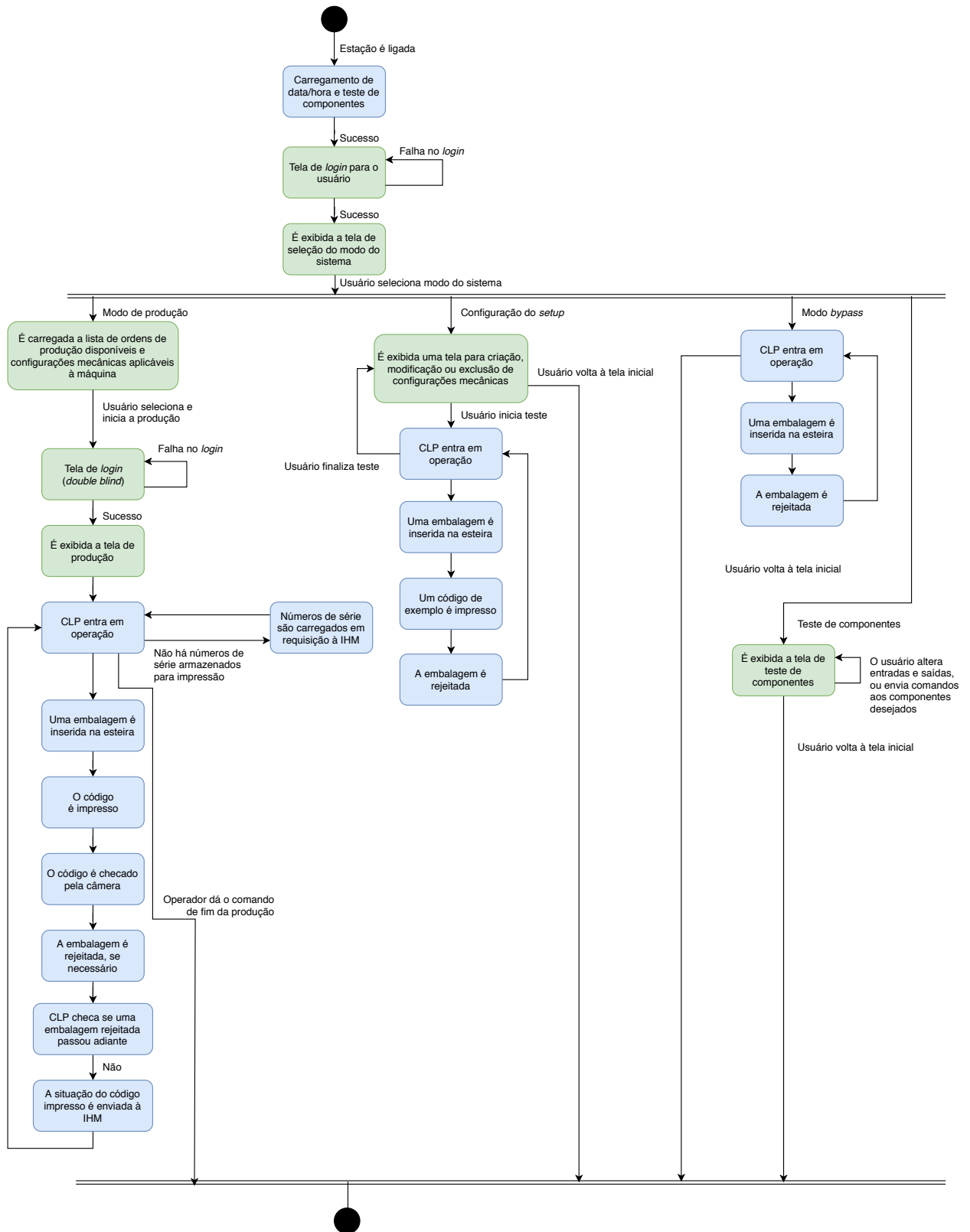
O período de inicialização da estação consiste nas primeiras ações executadas logo após a mesma ser ligada. Em especial, algumas tarefas de *setup* devem ser executadas:

- Estabelecimento da comunicação entre o CLP e PC Linux;
- Carregamento das informações do sistema, como data e hora, por parte do CLP;
- Estabelecimento de conexão com os componentes que exigem de protocolos mais complexos de comunicação, como impressora e câmera.

Essas ações de inicialização são necessárias para o funcionamento, pois a comunicação entre PC Linux e CLP, por exemplo, é obrigatória para o envio de comandos do operador. Por outro lado, a conexão do PC Linux com o servidor não necessita de inicialização, só se tornando necessária quando for preciso carregar informações do mesmo, e por isso não está listada nas ações de inicialização.

Caso a conexão seja estabelecida, o CLP carrega do PC Linux as informações do sistema, que são a data e a hora atuais, pois o dispositivo não conta com sincronização automática com a *internet* e precisa desses dados para registrar as movimentações referentes aos números de série.

Figura 17 – Diagrama de atividades completo do sistema



Fonte: do autor

Ainda, alguns componentes são controlados pelo CLP por protocolos de comunicação específicos, mais complexos que apenas entradas e saídas binárias. Na estrutura atual de funcionamento, a câmera e a impressora usam de tal processo.

Apenas após o processo de inicialização dos componentes e a verificação de que o CLP está pronto para operar, o usuário poderá realizar o *login* na IHM. Tal autenticação é realizada junto ao servidor e retorna as permissões associadas ao tipo de usuário.

Feitas as conexões iniciais e o *login*, a tela inicial de escolha do modo de funcionamento é exibida, permitindo ao usuário ir ao modo de produção ou às opções de manutenção. As opções dispostas são apresentadas a seguir:

- Acessar o modo de produção;
- Configurar o *setup* para diferentes embalagens;
- Simular a produção no modo *bypass*;
- Testar os componentes individualmente.

A figura 18 apresenta, isoladamente, o diagrama de atividades relativo à inicialização. Nela, são apresentados, na cor verde, os passos de responsabilidade do PC Linux; e na cor azul, os passos de responsabilidade do CLP.

4.1.2 Produção

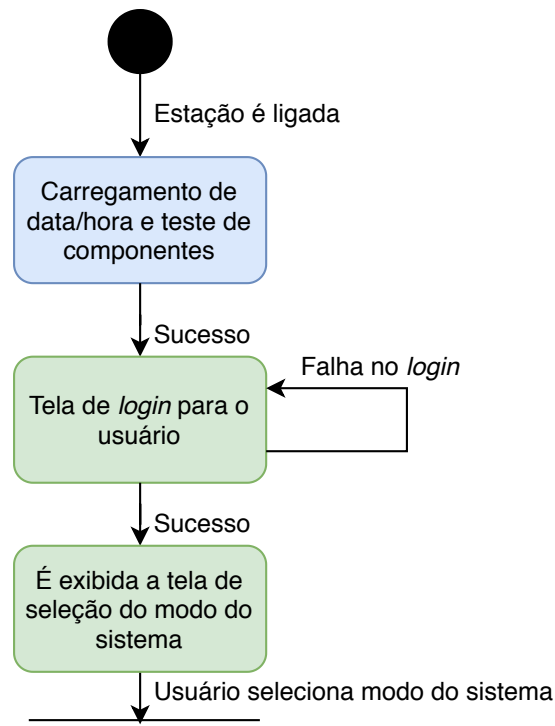
Quando selecionado o modo de produção, a função principal do sistema será executada, que é serializar as embalagens e garantir a qualidade da impressão.

Para que isso seja possível, é necessário que o operador selecione, na IHM, qual a ordem de produção que deseja serializar e qual a configuração mecânica da estação associada ao medicamento que será aplicada, sendo que todas essas informações devem ser registradas no banco de dados da estação. Feito isso, o botão de início da produção estará disponível para iniciar o processo.

A estação está, então, pronta para iniciar a produção, mas a operação só será autorizada pela autenticação de um supervisor — usuário com permissões superiores ao operador — que deve inserir suas credenciais na interface gráfica. Esse método de dupla checagem antes de iniciar a produção é chamado de *double-blind*.

Agora, a máquina inicia sua operação e a tela passa a exibir todas as informações relativas à produção. Com isso, a esteira está continuamente ativada e as embalagens que forem inseridas receberão a impressão e serão checadas. O fim da produção só se dá por meio do comando de parada do operador.

Figura 18 – Diagrama de atividades relativo à inicialização



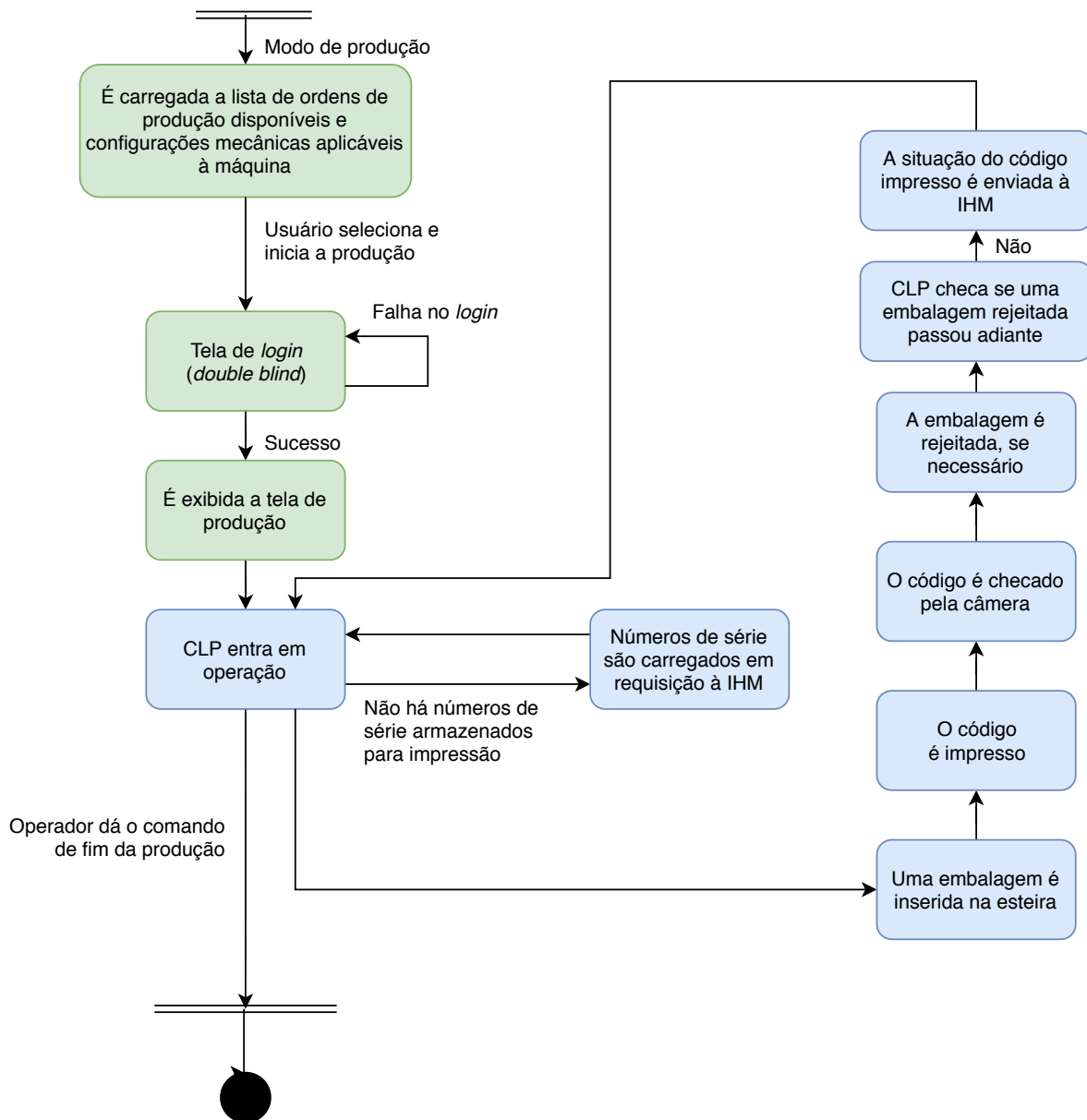
Fonte: do autor

Enquanto não ocorre o comando de parada por parte do operador, a estação de serialização continua executando as seguintes operações:

1. O CLP verifica quantos números de série (únicos e gerados aleatoriamente) ele possui armazenados. Se forem necessários mais códigos, ele os requisita ao PC Linux;
2. Quando uma embalagem é detectada na entrada da estação, as operações de serialização são, sequencialmente, executadas;
3. O código é impresso na embalagem pela impressora;
4. A qualidade do código impresso é checada pela câmera;
5. O produto é rejeitado ou não pelo atuador pneumático, dependendo da qualidade do código impresso;
6. A situação do número de série impresso, seja ele válido ou descartado, é repassada imediatamente ao PC Linux. Esta, por sua vez, armazena a informação e repassa ao servidor.

Esse conjunto de operações é representado, individualmente, na figura 19, iniciando a partir do momento em que o modo de operação é selecionado e terminando no comando

Figura 19 – Diagrama de atividades relativo à produção



Fonte: do autor

de parada do sistema. Nela são apresentados, na cor verde, os passos de responsabilidade do PC Linux; e na cor azul, os passos de responsabilidade do CLP.

4.1.3 Manutenção

A estação de serialização possui algumas funções não associadas à produção, as quais serão chamadas como modo de manutenção neste documento. A seguir, o diagrama de atividades de cada uma delas é descrito e apresentado.

4.1.3.1 Configuração do *setup* de embalagens

Como já descrito no capítulo 2, a estação conta com fusos de ajuste manual com indicadores numéricos de posição, para que o operador possa registrar as posições relativas ao *setup* de cada embalagem para posterior troca. Em algum momento, porém, pode ser necessário alterar o registro de tais configurações, seja pela alteração nas dimensões de uma embalagem ou pelo desgaste da máquina.

Para isso, a tela de alteração das configurações permite a consulta às configurações existentes, a alteração das mesmas e a simulação da operação para testar a passagem das embalagens.

Ao acessar essa funcionalidade, a IHM carrega do servidor uma lista de todos os *setups* existentes, os quais podem ser filtrados por tipo de produto. Nessa tela, é possível realizar o conjunto de operações CRUD (*Create, Read, Update, Delete*) dos *setups*.

O teste de cada uma das configurações pode ser iniciado pelo usuário. Quando isso ocorre, o CLP inicia a esteira e opera como se estivesse em produção. Porém, os códigos impressos são apenas números de teste e todas as embalagens são, obrigatoriamente, rejeitadas, evitando que um produto não registrado passe adiante na linha de produção.

Essa operação de configuração é representada, individualmente, na figura 20. Nela são apresentados, na cor verde, os passos de responsabilidade do PC Linux; e na cor azul, os passos de responsabilidade do CLP.

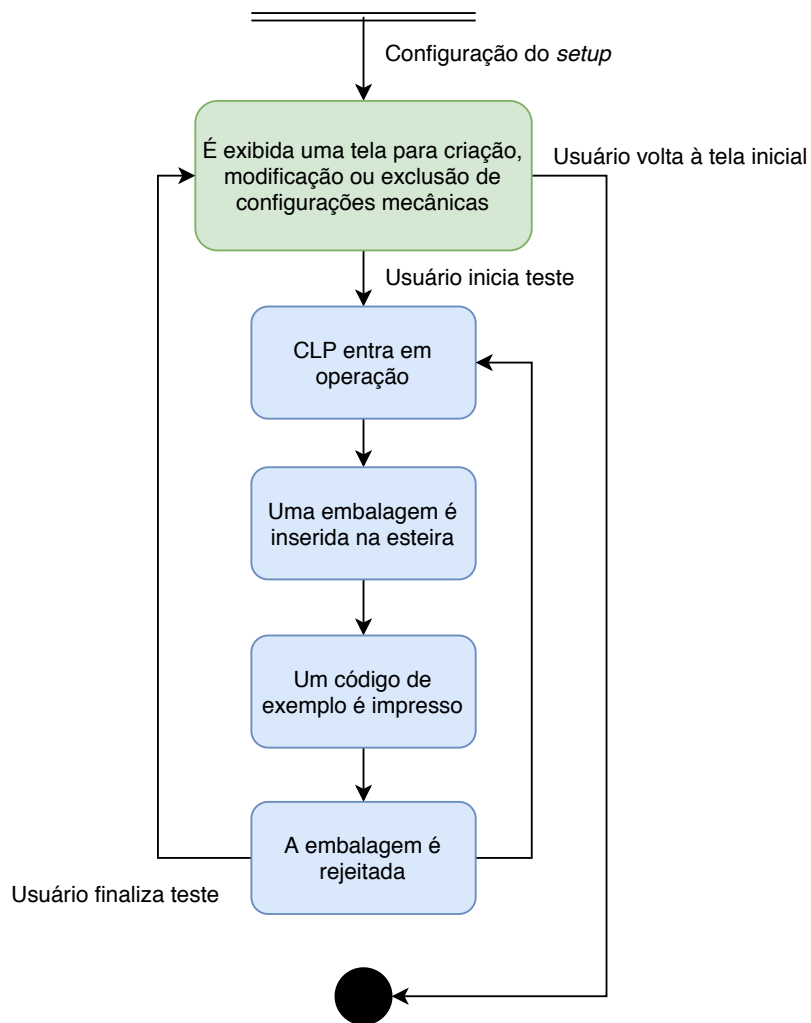
4.1.3.2 Simulação da produção em modo *bypass*

De forma similar ao teste de operação para configuração dos *setups*, a simulação com *bypass* inicia a esteira na estação para a passagem de embalagens. Nessa opção, porém, nenhuma impressão ou checagem visual é realizada e todos os produtos devem ser rejeitados. O diagrama que descreve essa operação é apresentado na figura 21. Nela, são apresentados, na cor verde, os passos de responsabilidade do PC Linux; e na cor azul, os passos de responsabilidade do CLP.

4.1.3.3 Teste dos componentes

De modo a fornecer ao operador a possibilidade de testar todos os componentes presentes na estação, existe a tela de teste dos componentes. Por meio de *checkboxes*, é possível modificar manualmente o estados de todos os componentes da estação.

Ainda, se o componente não pode ser testado apenas com a alternância entre nível lógico baixo e alto, existe a opção de enviar os comandos a cada um deles. Um exemplo é a esteira, a qual pode ser inicializada, entrar em movimento ou parar.

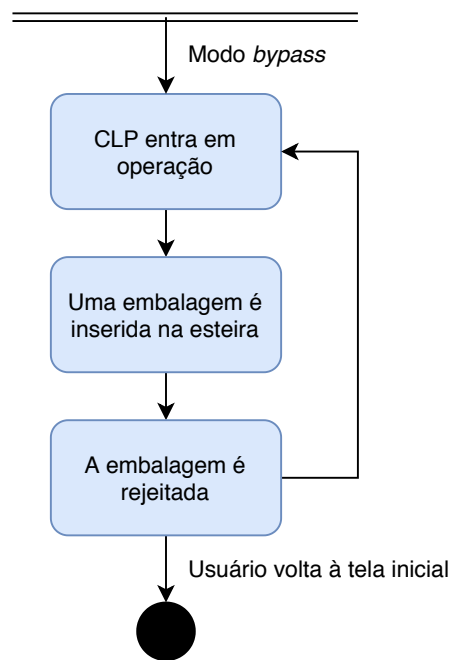
Figura 20 – Diagrama de atividades relativo à configuração do *setup*

Fonte: do autor

O diagrama associado ao teste é apresentado na figura 22. Nela, são apresentados, na cor verde, os passos de responsabilidade do PC Linux.

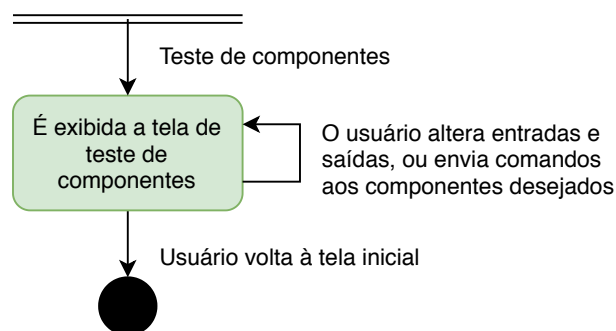
4.2 Diagramas de sequência

Os diagramas de atividade, apresentados na seção anterior, são capazes de descrever toda a sequência de operações em todos os modos de operação do sistema. Porém, para melhor especificação de quais mensagens são trocadas entre os cada um desses componentes, um diagrama de sequência em linguagem UML pode ser capaz de modelar melhor todas as comunicações a serem implementadas, sendo particularmente úteis para representar a sequência de eventos e respostas em um caso de uso [7]. Por isso, as reuniões incluíram a elaboração, em conjunto com os antigos desenvolvedores, desses diagramas a serem adotados na nova estrutura implementada na estação.

Figura 21 – Diagrama de atividades relativo à simulação da produção em modo *bypass*

Fonte: do autor

Figura 22 – Diagrama de atividades relativo ao teste dos componentes



Fonte: do autor

Como esse tipo de diagrama pode representar apenas um caso de uso, o diagrama de sequência aqui apresentado só será elaborado para o modo de produção. Além disso, o período cíclico desse modo será isolado em um segundo diagrama, buscando facilitar a compreensão do mesmo.

Tais modelos tornam claros os papéis de cada um dos componentes, principalmente na relação entre IHM e CLP, na qual ambos terão iniciativa na comunicação. Cada uma das seções a seguir apresenta e descreve os diagramas elaborados.

4.2.1 Inicialização e período pré-produção

Neste diagrama, está compreendido o período de inicialização, incluindo o *setup* do CLP e *login*, e a definição dos parâmetros associados à produção a ser iniciada.

No momento em que a máquina é ligada, o CLP realiza uma requisição das informações do sistema contendo a data e a hora atuais. Em seguida, o mesmo estabelece a conexão com a câmera e a impressora.

Em paralelo, o operador insere os dados de *login* na IHM, a qual realiza a autenticação e recebe as permissões do servidor. Após esse período, como já descrito na seção anterior, o usuário poderá escolher entre o modo de produção ou as diferentes opções de manutenção na IHM. Neste diagrama, é considerado que o mesmo escolhe o modo de produção.

Antes do início da produção, o usuário deve selecionar, em sequência, a ordem de produção a ser iniciada e a configuração mecânica associada que foi aplicada na estação.

Finalizado o processo de inicialização da produção, resta apenas o processo de *double-blind*, com a autenticação do supervisor autorizando o início da produção por meio de uma requisição ao servidor. Feito isso, a IHM envia ao CLP o comando de início da produção. Tal comando é, também, representado no próximo diagrama, a fim de facilitar a compreensão da continuidade de ambos.

O diagrama que representa tais comunicações é apresentado na figura 23.

4.2.2 Produção

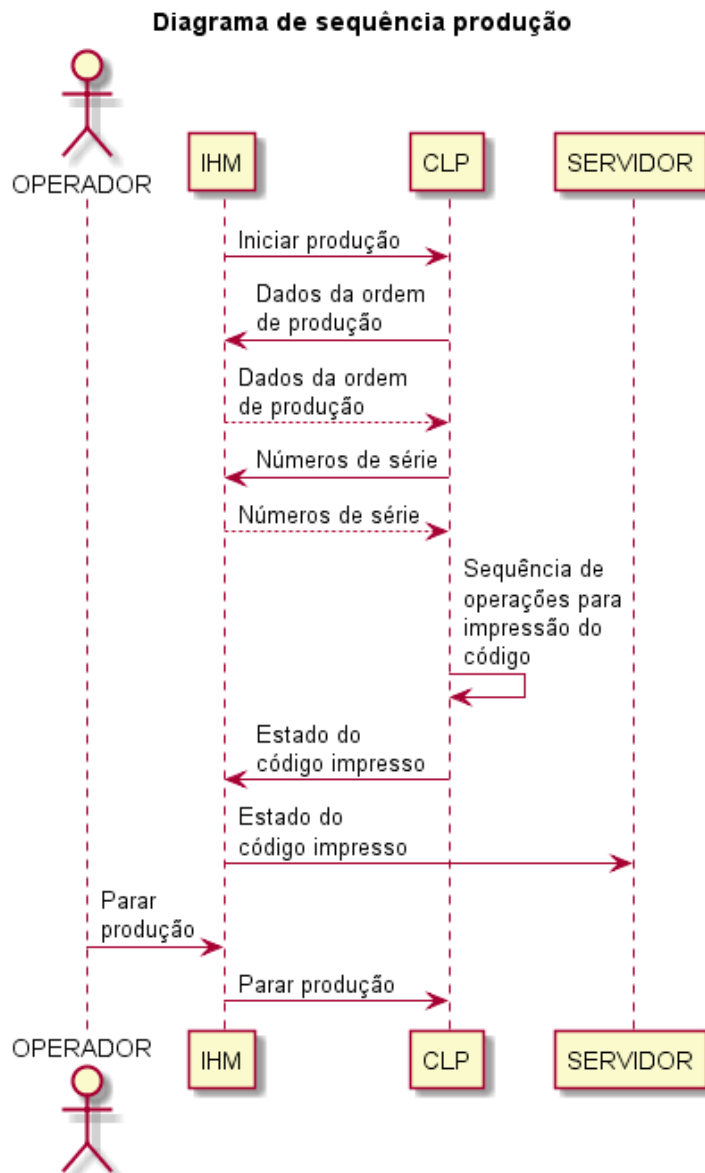
Este diagrama tem como objetivo apresentar a sequência de comunicações realizada durante a produção de embalagens. A fim de facilitar a compreensão e conectá-lo com o diagrama anterior, este apresenta em seu início o mesmo comando de início da produção enviado ao CLP.

Após receber o comando de início, o CLP, por sua vez, passará a tomar iniciativa nas comunicações com o PC Linux, a começar pela requisição dos dados da ordem de produção que será produzida. Logo em seguida, o CLP verifica se possui armazenados em seu *buffer* quaisquer números de série para a impressão e, se não possuir, irá também realizar uma requisição ao PC Linux pedindo um lote de códigos. Sempre que necessário durante a sua operação, a lista de números de série será atualizada mediante requisição do CLP.

Com todas as informações necessárias para entrar em produção, os programas a serem executados na câmera e impressora são carregados.

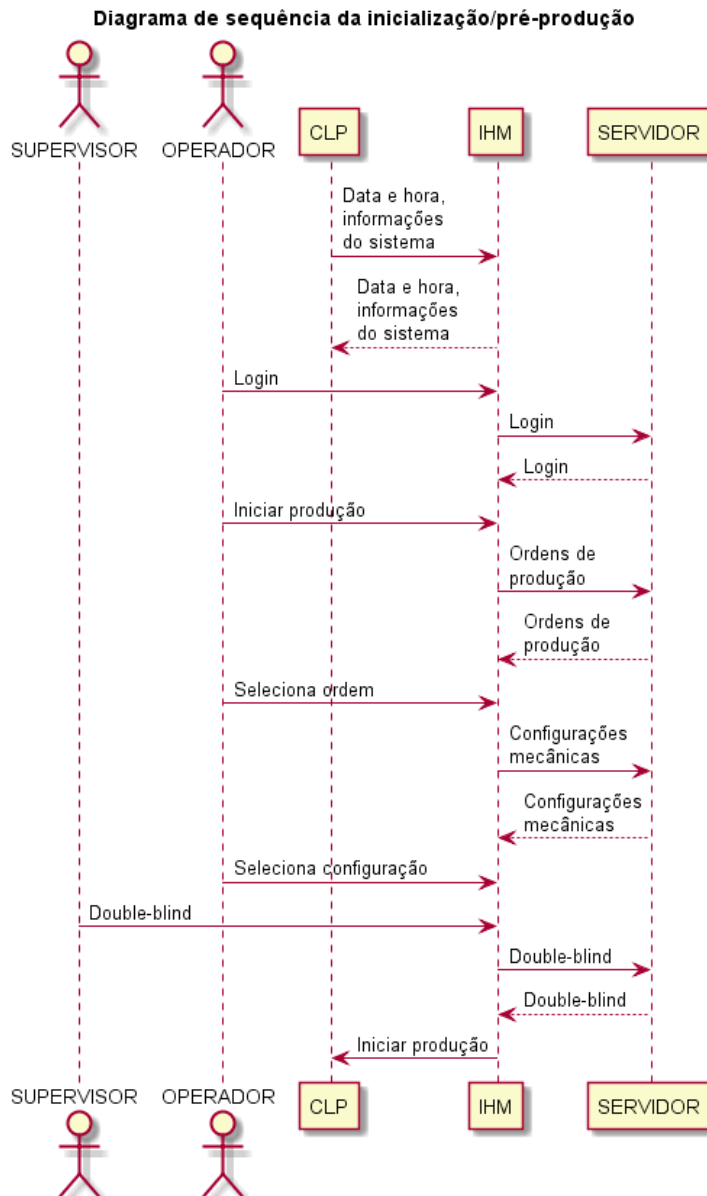
Feito o *setup* de todos os componentes, o CLP passa a ler o nível lógico do sensor de entrada e, se detectada a entrada de um objeto, comunica-se com cada um dos componentes

Figura 23 – Diagrama de sequência relativo à inicialização e pré-produção



Fonte: do autor

Figura 24 – Diagrama de sequência relativo à produção



Fonte: do autor

ao longo da esteira para que a impressão, checagem visual e rejeito sejam feitos nas posições corretas. A fim de simplificar o diagrama, essas operações de escrita são realizadas do CLP para o próprio CLP, em uma única operação. Dessa forma, a lógica interna do CLP de tratamento da posição da embalagem não deixa de ser representada neste diagrama.

Por fim, ao final da passagem de uma embalagem pela linha, o estado do número de série impresso é enviado ao PC Linux, a qual o repassa para o servidor.

Esse conjunto de operações ocorre continuamente, até que o comando de parada é executado pelo operador na IHM.

O diagrama da figura 24 apresenta as comunicações descritas nesta seção.

5 Implementação do sistema de controle e automação

Nos capítulos anteriores a este, foram apresentadas a contextualização do problema existente na plataforma VPak-RT existente na empresa Pollux Automation e proposta uma solução para a sua melhoria, seguida da modelagem em alguns diagramas. Feitos esses passos, a sequência é a implementação do sistema de controle e automação nos dispositivos, a qual será descrita neste capítulo.

A implementação, por sua vez, exige a modelagem de uma estrutura de *software*, para que, então, possa ser feito o desenvolvimento do código. Por isso, em cada seção, serão abordadas as propostas de implementação relativas a cada um dos componentes, baseadas na linguagem de programação e capacidades do *hardware*, seguidos de uma breve descrição de suas respectivas funções. Ainda assim, apenas os modelos do que foi desenvolvido serão apresentados, sem a exibição dos códigos.

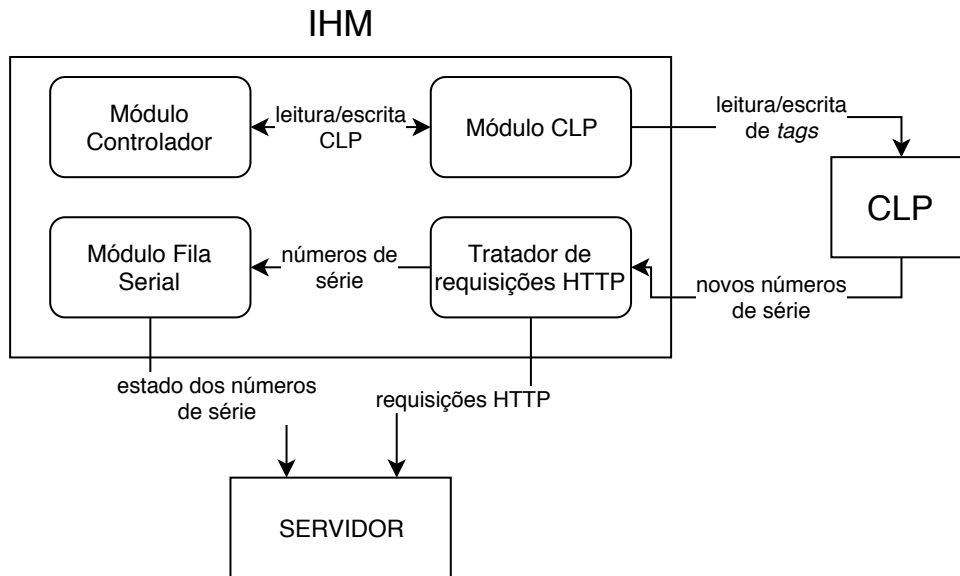
Os três dispositivos dos quais será descrita a implementação desenvolvida são: (a) Interface Humano Máquina (IHM), executada em um PC Linux; (b) servidor central e (c) CLP.

5.1 Interface Humano Máquina (IHM)

O módulo de Interface Humano Máquina (IHM) é o componente deste projeto no qual é desenvolvida a maior quantidade de módulos de *software*, visto que o CLP e o servidor central já possuem grande parte de suas funcionalidades implementadas.

Como já descrito anteriormente, este será um PC com sistema operacional Linux, o qual é capaz de executar a aplicação Pollux DMS, desenvolvida na linguagem de programação Python. Por ser uma aplicação já estabelecida nos projetos desenvolvidos pela empresa, a mesma já contém inúmeros trechos de código que podem ser reutilizados como, por exemplo, um sistema para a execução de máquinas de estado, um padrão para a implementação de interfaces gráficas *web* e um módulo de comunicação *Modbus* com CLPs de diversos fabricantes.

No sistema presente no Pollux DMS, a implementação de máquinas de estado pode ser realizada de forma modular, com a possibilidade de comunicação entre elas para que possam progredir em conjunto. Por isso, a estrutura adotada será a de desenvolver um módulo principal, chamado de **controlador**, o qual é responsável por coordenar a operação de todos os outros módulos, sendo eles: o **CLP**, responsável pela gerência da

Figura 25 – Estrutura de módulos de *software* da IHM e comunicação entre eles

Fonte: do autor

comunicação com o Controlador Lógico Programável; e a **fila serial**, responsável por manter atualizados, no servidor, o estado dos números de série recebidos do CLP. A figura 25 ilustra a estrutura de funcionamento dos módulos de *software* presentes na IHM.

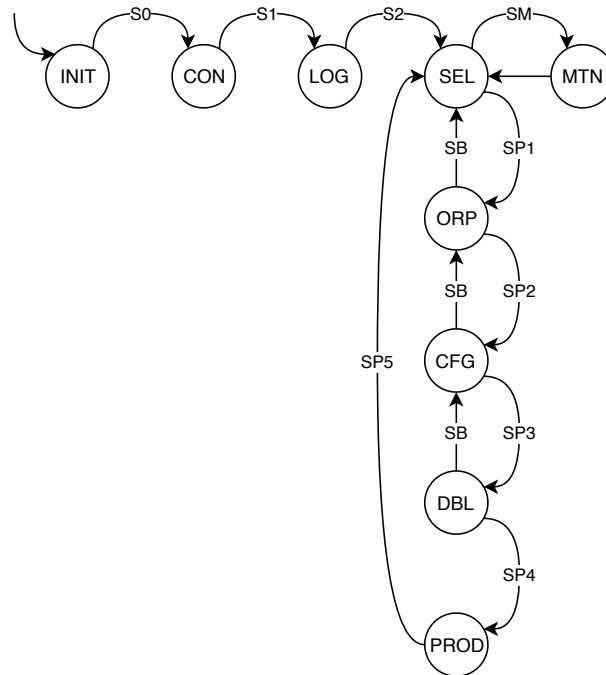
5.1.1 Controlador

O módulo controlador, além de responsável pela coordenação de todos os outros componentes, também interpreta os comandos oriundos da interface gráfica de toque disponível ao operador. Por isso, pode-se considerar que o mesmo, se representado em uma máquina de estados, apresentará comportamento semelhante ao do diagrama de atividades da figura 17. Mesmo que, naquele diagrama, são apresentadas as funções relativas ao CLP, este módulo também terá de incluir um estado que aguarde o sucesso dessas operações de comunicação para, então, prosseguir com os estados da IHM, já que ambos são interdependentes.

Apresentada na figura 26 e descrita pela tabela de estados 1 e pela tabela de transições 2, a máquina de estados é descrita pelo seguinte comportamento: ao ser inicializada, esta aguarda a conexão do CLP e uma checagem do seu *status*, garantindo que esteja funcional, mensagem que será originada no módulo do CLP; em seguida, é checado o *login* do operador para, então, ser aberta a tela de seleção entre o modo de produção ou os modos de manutenção.

Enquanto no modo de produção, a máquina de estados passa, inicialmente, pelas mensagens da interface gráfica indicando a seleção da ordem de produção, da configuração mecânica da estação e do comando de início, o qual é seguido de um novo *login* do

Figura 26 – Máquina de estado para o módulo Controlador



Fonte: do autor

supervisor (*double-blind*). Iniciada a produção, a responsabilidade pelo gerenciamento de todos os componentes é do CLP e, por isso, apenas um estado representa que o modo de produção encontra-se ativo, aguardando o comando de parada do operador.

Ainda, o módulo controlador pode ser direcionado ao modo de manutenção onde, sob o comando do operador na interface gráfica, é possível testar os comandos de componentes como a esteira e a impressora, ou mudar o nível lógico aplicado sobre as diversas saídas do CLP.

Tabela 1 – Lista de estados para a maquina de estados da figura 26

INIT	Inicialização da máquina
CON	Aguarda teste de conexão com o CLP
LOG	<i>Login</i> do usuário
SEL	Aguarda seleção do modo de funcionamento
ORP	Tele de seleção da ordem de produção a ser produzida
CFG	Tela de seleção da configuração mecânica a ser aplicada na máquina
DBL	<i>Double-blind</i>
PROD	Sistema em produção

5.1.2 Comunicação com CLP

Se as funções de comunicação com o CLP fossem integralmente implementadas no controlador principal, o mesmo poderia ficar extenso e com muitos trechos de código

Tabela 2 – Lista de transições para a máquina de estados da figura 26

Nome	Est. origem	Est. destino	Descrição
S0	INIT	CON	Máquina de estados inicializada normalmente
S1	CON	LOG	Conexão com o CLP estabelecida
S2	LOG	SEL	<i>Login</i> realizado com sucesso
SM	SEL	MTN	Modo de manutenção selecionado
SP1	SEL	ORP	Modo de produção selecionado
SP2	ORP	CFG	Ordem de produção selecionada
SP3	CFG	DBL	Configuração mecânica selecionada e aplicada
SP4	DBL	PROD	<i>Double-blind</i> realizado com sucesso
SP5	PROD	SEL	Comando de fim da produção
SB	*	*	Comando de retorno do operador

repetidos. Por isso, o módulo de comunicação com o CLP representa uma abstração dessas mensagens, trocadas entre a plataforma em Python com o CLP. Mesmo com a existência de um protocolo de comunicação entre ambos, o qual já garante o envio das mensagens, opta-se pela implementação de uma lógica que garanta o reconhecimento da mensagem enviada para o CLP.

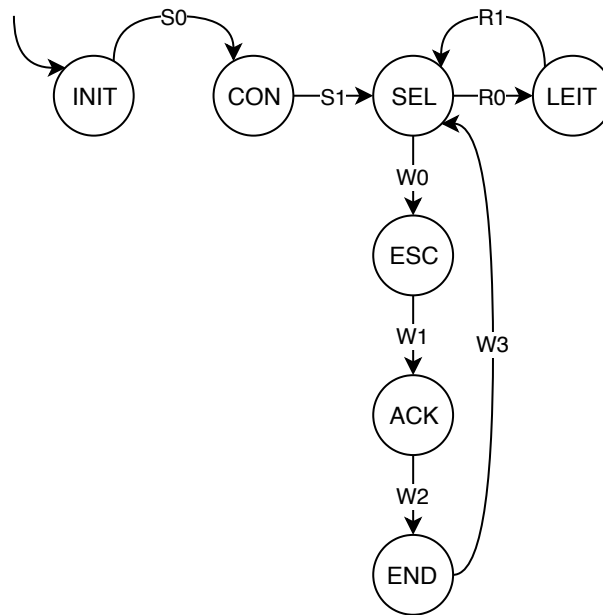
Nesse sistema, as mensagens enviadas para o CLP são escritas em *tags* (endereços de memória de 1 *bit*), onde cada um desses endereços representa um comando para a lógica do controlador. Em um endereço adicional, o CLP indicará o reconhecimento do comando com um *bit* 1, o qual deverá ser lido pela aplicação na IHM. Assim que reconhecer que o CLP identificou o comando, a IHM escreve novamente na *tag* de comando um *bit* 0, liberando a sequência da operação.

A máquina de estados que rege a comunicação com o CLP é uma implementação dessa lógica. Inicialmente, ocorre o estabelecimento da comunicação com o CLP e a checagem de seu *status*, o qual é enviado para o módulo controlador. Em seguida, um estado de espera é responsável por receber as mensagens contendo os comandos a serem enviados para o CLP e realizar o processo de comunicação apresentado anteriormente. Tal sequência de estados é apresentada na figura 27, seus estados são descritos na tabela 3 e suas transições na tabela 4.

5.1.3 Atualização dos números de série

Isolado dos módulos anteriores, é criado um módulo responsável por manter atualizados os números de série criados no servidor central. Após enviado ao CLP, cada número de série pode ter uma situação a ele atribuída, podendo esse já estar impresso, mas em

Figura 27 – Máquina de estado para o módulo CLP



Fonte: do autor

Tabela 3 – Lista de estados para a maquina de estados da figura 26

INIT	Inicialização da máquina
CON	Estabelecimento da conexão com o CLP
SEL	Aguarda comando oriundo do controlador
LEIT	Realiza leitura de <i>tag</i> do CLP
ESC	Realiza escrita de <i>tag</i> no CLP
ACK	Aguarda o CLP reconhecer o comando enviado
END	Envia o fim da escrita ao CLP e aguarda o reconhecimento

uma embalagem rejeitada ou tornando-se válido. Essas informações sempre precisam ser atualizadas no servidor mas, como as mesmas são registradas na IHM, é possível continuar a operação sem conexão com o servidor central, atualizando-os posteriormente.

Para que isso seja possível, este módulo monitora constantemente os códigos registrados no banco de dados que ainda não foram atualizados no servidor, baseado em sua última data de atualização. Toda vez que um novo código é atualizado, esta máquina de estados se encarrega de atualizá-lo no servidor e, caso não exista conexão, um temporizador é acionado para aguardar a próxima tentativa de envio. Tal lógica é apresentada na figura 28, acompanhada da tabela de estados 5 e da tabela de transições 6.

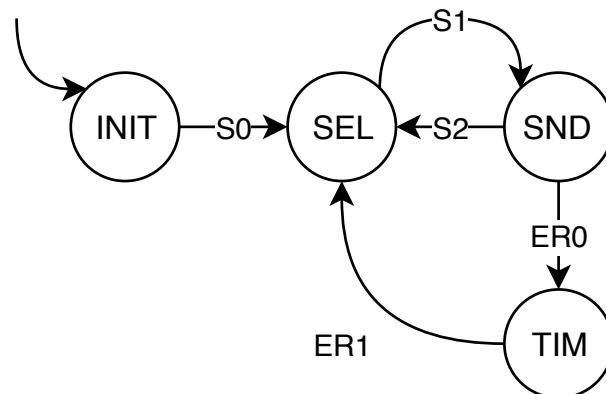
5.1.4 Tratadores de requisições do CLP

Além das máquinas de estado, há a necessidade de tratar as requisições de dados originadas do CLP, já que ambos os dispositivos tomam iniciativa nessa comunicação. Para

Tabela 4 – Lista de transições para a máquina de estados da figura 26

Nome	Est. origem	Est. destino	Descrição
S0	INIT	CON	Máquina de estados inicializada normalmente
S1	CON	SEL	Conexão com o CLP estabelecida
R0	SEL	LEIT	Comando de leitura de <i>tag</i> recebido
R1	LEIT	SEL	<i>Tag</i> lida com sucesso e valor retornado à origem
W0	SEL	ESC	Comando de escrita em <i>tag</i> recebido
W1	ESC	ACK	Escrita realizada com sucesso
W2	ACK	END	Escrita reconhecida pelo CLP
W3	END	SEL	Fim da comunicação reconhecido pelo CLP

Figura 28 – Máquina de estados para o módulo de atualização dos números de série



Fonte: do autor

Tabela 5 – Lista de estados para a máquina de estados da figura 26

INIT	Inicialização da máquina
SEL	Em espera, aguardando a chegada de novos números de série
SND	Atualiza números de série ainda não enviados ao servidor
TIM	Aciona <i>timer</i> para próxima tentativa

isso, são criados três tratadores de requisições, os quais não são regidos por máquinas de estado, responsáveis por tratar imediatamente os seguintes dados:

- Requisição de informações sobre data e hora na inicialização do CLP;
- Requisição de informações sobre a ordem de produção selecionada pelo operador ao iniciar a produção;

Tabela 6 – Lista de transições para a máquina de estados da figura 26

Nome	Est. origem	Est. destino	Descrição
S0	INIT	SEL	Máquina de estados inicializada normalmente
S1	SEL	SND	Novo número de série recebido E temporizador permite o envio de nova requisição
S2	SND	SEL	Números de série atualizados com sucesso
ER0	SND	TIM	Erro ao atualizar números de série no servidor
ER1	TIM	SEL	<i>Timer</i> para a próxima requisição iniciado com sucesso

- Requisição de lista de números de série a serem impressos pelo CLP;
- Envio de atualização do *status* de um número de série, enquanto em produção.

5.1.5 Interface gráfica web

Uma das principais características a serem desenvolvidas na IHM é a própria interface gráfica que, anteriormente desenvolvida em uma *software* proprietário, deve agora ser reprojeta para o ambiente *web* existente no Pollux DMS.

Tal tarefa, por outro lado, não necessita da criação de uma plataforma para a renderização de páginas *web*, pois essa funcionalidade já existe no sistema da Pollux, bastando que os componentes sejam organizados visualmente.

O DMS faz uso das tecnologias *Bootstrap*, um *framework* capaz de formatar o conteúdo de forma padronizada, para a apresentação de seus componentes, bastando que os conteúdos sejam inseridos na linguagem de programação *Python*, a qual irá renderizar os arquivos HTML.

As seções a seguir apresentam uma breve descrição das telas que serão desenvolvidas. Algumas delas, como a de *login* e de informações do sistema, já estão presentes no Pollux DMS por padrão e, por isso, não serão apresentadas no contexto deste trabalho. O resultado final da interface gráfica desenvolvida é apresentado no capítulo 6.

5.1.5.1 Tela inicial

Apresentada após a autenticação do operador, a tela inicial apresenta algumas informações básicas sobre a estação como, por exemplo, o estado da conexão com o CLP e o servidor, o último lote produzido e a versão do sistema no servidor. Além disso, os

comandos para as operações que podem ser realizadas devem ser apresentados de forma clara, sendo eles: início do modo de produção e início do modo de manutenção.

5.1.5.2 Modo de produção

O modo de produção contém, como especificado em capítulos anteriores, uma série de operações antes do início da operação, o qual pode ser modelado pelas seguintes telas:

- Para a seleção da ordem de produção, a lista de dados obtida do servidor é apresentada em uma tabela, sendo possível pressionar na linha contendo a ordem que se deseja produzir;
- Em seguida, de forma similar, a lista de configurações mecânicas é apresentada em uma tabela para que o operador pressione naquela que deseja aplicar;
- A configuração mecânica selecionada deve ser apresentada de forma intuitiva ao operador para que o mesmo aplique as correções nos componentes físicos. Para isso, uma imagem representando a estação de serialização é apresentada, acompanhada dos indicadores numéricos associados a cada um dos fusos ajustáveis;
- Ainda, antes da produção, a tela de autenticação do supervisor é apresentada, a qual já está implementada no Pollux DMS;
- Por fim, a tela de produção precisa apresentar os dados em tempo real ao usuário de forma clara. Para isso, uma barra de progresso indica a quantidade de embalagens produzidas em relação ao total esperado do lote. Ainda, informações a respeito do lote em produção e do número de números de série disponíveis são apresentados.

5.1.5.3 Modo de manutenção

A tela de manutenção do sistema consiste na apresentação de diversos botões para o teste de cada um dos componentes, separados em três categorias: teste de entradas e saídas, teste da esteira e teste da impressora, os quais podem ser selecionados pelo operador.

No teste de entradas e saídas, uma lista com todos os componentes a serem testados é apresentada, acompanhados de caixas de seleção que indicam o estado atual da saída. Assim, o operador pode visualizar qual o nível lógico forçado na saída e se a mesma está sendo apresentada corretamente no *hardware*.

Para o teste dos componentes mais complexos, porém, uma tela específica é desenvolvida para melhor compreensão do operador sobre o teste. Em uma tela para o teste de cada componente, além de botões para o envio dos comandos associados, é possível

visualizar qual o estado de cada componente segundo o CLP e se os mesmos estão se alterando com o envio desses comandos.

5.2 Servidor central

O servidor central é a aplicação presente no topo da hierarquia de componentes desenvolvidos pela Pollux, e agrupa todas as informações a respeito das estações presentes no VPak-RT, assim como realiza a comunicação com servidores externos ao sistema, como ERPs (*Enterprise Resource Planning*) ou bancos de dados proprietários das empresas.

Essa aplicação, porém, já encontra-se em estágio avançado de desenvolvimento e não será alvo de grandes mudanças no contexto deste PFC. Algumas novas funcionalidades, porém, devem ser implementadas, devido às requisições realizadas pela IHM sobre informações necessárias ao funcionamento da estação de serialização. O servidor central deve responder às solicitações de: (a) lista de números de série para impressão; (b) lista de ordens de produção disponíveis para serialização; e (c) configurações mecânicas da máquina para cada tipo de medicamento.

A solicitação de uma lista de números de série é requisitada pela IHM que, por sua vez, recebeu a requisição do CLP. Isso ocorre quando os códigos disponíveis para impressão estão se esgotando e novos números aleatórios precisam ser gerados. Como esses números precisam ser únicos em todo o sistema, eles devem ser gerados no servidor e repassados à estação.

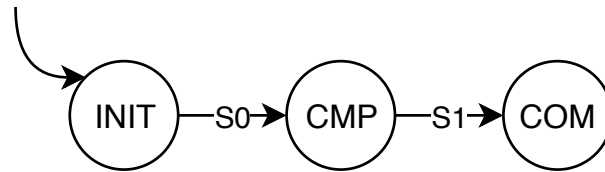
Além disso, no início da produção, o operador deve selecionar qual a ordem de produção que terá suas embalagens impressas, além de visualizar a configuração mecânica da máquina utilizada. Por isso, as requisições oriundas da IHM são tratadas no servidor, o qual é responsável por manter o banco de dados com essas informações.

5.3 Alterações no *software* do CLP

As funcionalidades desenvolvidas no contexto do *software* DMS, apresentadas anteriormente, substituem as funções da interface gráfica desenvolvida anteriormente para o CLP. Por isso, mesmo que a proposta deste projeto seja reduzir o número de alterações no software do controlador lógico, ainda serão necessárias mudanças para tratar da substituição dos comandos do operador pela comunicação com o Pollux DMS.

Para que isso seja possível, um novo módulo é desenvolvido em substituição ao módulo de Interface Humano Máquina existente. Suas funcionalidades, apesar de similares, diferem na forma em que os comandos do operador são interpretados, sendo que o primeiro deles interpreta os comandos oriundos da interface gráfica, enquanto o segundo faz a constante leitura das *tags* disponíveis via protocolo de comunicação à IHM.

Figura 29 – Máquina de estados para o módulo de comunicação do CLP com o DMS



Fonte: do autor

A linguagem de programação utilizada no CLP é o *C++*, organizada de forma a ser executada de forma cíclica, assim como no funcionamento da linguagem *Ladder*, utilizada frequentemente em controladores do tipo.

Nesse contexto, o módulo desenvolvido substituiu aquele que desempenhava a coordenação geral do CLP no que se trata da comunicação com a IHM e dos comandos desempenhados pelo operador. Apresentado na forma de uma máquina de estados na figura 29, acompanhado da tabela de estados 7 e da tabela de transições 8, o novo módulo realiza, em ordem, as seguintes tarefas:

1. Estabelece a conexão inicial com o sistema central, presente na IHM, requisitando a data e hora atuais do sistema. A resposta dessa requisição já define que a conexão está estabelecida;
2. Estabelece a conexão com todos os componentes;
3. O módulo passa a monitorar os comandos originados da IHM, como especificado na seção 5.1.2. Os comandos recebidos são, então, enviados aos outros módulos de *software C++* previamente desenvolvidos, os quais se comportam como se os comandos tivessem sido recebidos da interface gráfica presente na versão anterior da estação.

Tabela 7 – Lista de estados para a maquina de estados da figura 26

INIT	Estabelecimento de comunicação com o novo sistema de controle e automação em plataforma DMS
CMP	Estabelecimento de comunicação com os componentes físicos
COM	Interpretação de comandos originados na interface gráfica

Dessa forma, foi possível verificar a vantagem de a versão anterior do *software* ter sido desenvolvido em módulos individualizados e independentes, pois não houve necessidade na alteração da lógica de coordenação dos componentes físicos, apenas substituindo o interpretador de comandos da IHM originalmente desenvolvido.

Tabela 8 – Lista de transições para a maquina de estados da figura 26

Nome	Est. origem	Est. destino	Descrição
S0	INIT	CMP	Dados de data e hora recebidos, comunicação estabelecida
S1	CMP	COM	Teste de componentes finalizado

6 Resultados e Considerações Finais

O Projeto de Fim de Curso abordado neste documento teve como objetivo o desenvolvimento de uma nova aplicação para a estação de serialização existente na plataforma VPak-RT, substituindo a interface gráfica original, desenvolvida em *software* proprietário, pelo sistema criado pela Pollux, o *Digital Management Suite*, o qual permite a comunicação com aplicações na nuvem, o acesso remoto e conta com uma linguagem de programação com vasta quantidade de recursos.

Após a apresentação, modelagem e descrição das atividades implementadas no sistema, este capítulo apresenta os resultados obtidos e as vantagens na mudança da estrutura hierárquica do sistema de controle e automação da estação de serialização. Além disso, alguns pontos foram identificados para a possibilidade de melhoria nos desenvolvimentos futuros do projeto.

Por fim, são apresentadas as considerações finais a respeito do trabalho desenvolvido, incluindo uma síntese pessoal a respeito das atividades, do atendimento às expectativas e do aprendizado durante o período de estágio.

6.1 Aplicação desenvolvida

Assim como descrito em capítulos anteriores, o objetivo deste projeto não era a criação de uma nova funcionalidade, mas sim a mudança da arquitetura de software da estação de serialização, a qual exigiu que funções já implementadas fossem recriadas. A análise dos resultados, porém, não pode ser verificada pela presença de novas funções, mas sim do funcionamento da estação sendo similar ao da versão anterior.

Para a verificação desses resultados, foram realizados testes na estação de serialização presente na própria empresa, de forma a validar todos os novos componentes desenvolvidos e as diferentes ações que o operador pode tomar. Como o desenvolvimento de *software* avançou em conjunto com a montagem física de uma nova estação, a ser enviada para o cliente, algumas etapas de teste envolveram diferentes configurações de componentes que pudessem validar as funções, as quais são listadas a seguir.

O primeiro item a ser desenvolvido foi a substituição do módulo que interpretava os comandos do operador na versão anterior da IHM pelo novo código em C++, o qual interpreta os comandos oriundos do PC Linux. Como ainda não havia a implementação do DMS, o CLP foi testado de forma isolada com a abstração dos comandos enviados por meio da própria interface gráfica de desenvolvimento. Dessa forma, independentemente da estação que envie os comandos ao CLP, foi possível garantir o funcionamento do mesmo

de forma isolada.

Em seguida, todo o desenvolvimento relativo à nova interface gráfica e lógica de controle e automação presente no PC Linux foram desenvolvidos com base na *framework* do Pollux DMS, tanto no sistema presente na estação como no servidor. Nesse momento, a estação estava ainda em montagem e, por isso, os testes foram realizados todos no próprio PC de desenvolvimento. Para que os comandos do CLP pudessem ser interpretados, foi criado um simulador de leitura e escrita no CLP, o qual repete o comportamento desejado do mesmo sem a necessidade do componente físico. Assim, da mesma forma que com os testes do CLP, a aplicação do Pollux DMS foi completamente testada.

Por fim, os módulos de comunicação entre CLP e o Pollux DMS puderam ser testados após a montagem da nova estação. Como, anteriormente, já se havia garantido o funcionamento isolado de cada uma das estações, esses testes envolveram apenas a solução de erros relativos às comunicações, com a execução das aplicações em cada um dos respectivos componentes em rede: servidor, IHM e CLP.

Nesse contexto, após os testes em conjunto com a equipe de desenvolvimento da empresa, o objetivo pode ser considerado como cumprido, já que a estação continua funcionando dentro dos modelos propostos no capítulo 4.

O desenvolvimento da nova interface gráfica representa a alteração mais visível ao operador, já que a mesma passou de um ambiente de desenvolvimento proprietário para uma interface *web* desenvolvida com auxílio do *framework Bootstrap*.

Nas novas telas desenvolvidas, foi possível aprimorar a experiência do usuário em comparação com as telas presentes na versão anterior do sistema. O resultado final de algumas delas presentes na interface *web* do sistema são apresentados nas figuras 30, 31 e 32.

6.2 Conclusão

Pelo conteúdo apresentado ao longo do desenvolvimento deste trabalho e os resultados obtidos ao longo da seção 6.1, pode-se concluir que os objetivos gerais e específico deste projeto foram atingidos, ao se obter uma nova estação de serialização com a alteração de componentes proposta e apresentando o mesmo funcionamento já existente na versão anterior.

Frente aos resultados obtidos, porém, alguns pontos de melhoria puderam ser levantados para que sejam desenvolvidos no futuro pela Pollux. Esses pontos não foram incluídos no escopo deste trabalho para que não fossem alterados os prazos relativos à entrega do projeto como um produto para a empresa. Assim, a seção 6.3 a seguir apresenta tais pontos passíveis de melhoria futura.

Figura 30 – Exemplo de interface gráfica final desenvolvida para a nova aplicação da estação de serialização

Batch Number	Product Number	Product Name	Fabrication Date	Expiration Date	Quantity
200020	4321	Remedio2	2019-11-01T00:00:00	2020-11-01T00:00:00	500
200019	1234	Remedio1	2019-11-01T00:00:00	2020-11-01T00:00:00	1000

Exibindo 1 até 2 de 2 linhas

Copyright © 1998 - 2019 Pollux Automation Todos os direitos reservados

Fonte: do autor

Figura 31 – Exemplo de interface gráfica final desenvolvida para a nova aplicação da estação de serialização

BATCH QUANTITY 0

APPROVED 0

REJECTED 0

STATUS Running

Batch Number: --
Product Name: --

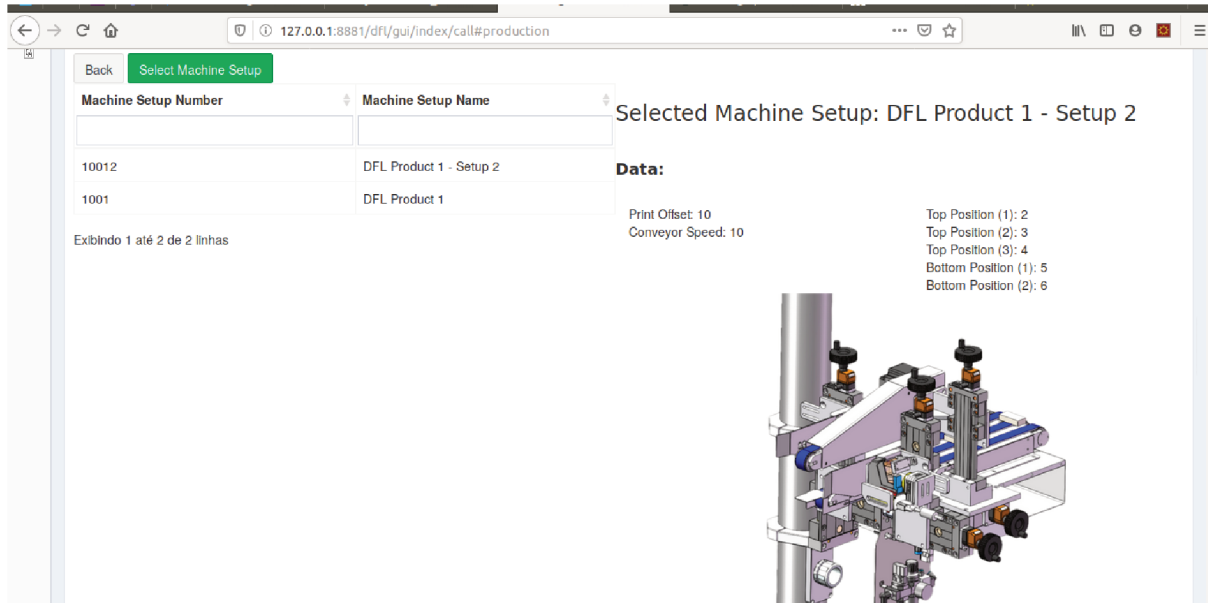
Fabrication Date: --
Expiration Date: --

Finish Production

Copyright © 1998 - 2019 Pollux Automation Todos os direitos reservados

Fonte: do autor

Figura 32 – Exemplo de interface gráfica final desenvolvida para a nova aplicação da estação de serialização



Fonte: do autor

6.3 Próximos passos

Ao longo do desenvolvimento do projeto, notou-se que algumas decisões a respeito da estrutura de *software* adotada são passíveis de melhoria para momentos futuros do projeto. O principal deste estes, a ser destacado a seguir, é a responsabilidade de cada um dos componentes na comunicação e coordenação da operação.

Na estrutura hierárquica desenvolvida neste projeto, quando analisada a comunicação entre CLP e IHM, percebe-se que ambos os dispositivos tomam iniciativa no envio de informações: a IHM é responsável por enviar os comandos do operador e checar o estado do CLP, como apresentado na seção 5.1.2; por outro lado, o CLP é responsável por requisitar as informações de data e hora em sua inicialização, bem como realiza a requisição de mais números de série para a impressão, para a IHM.

Tal abordagem se apresenta funcional, mas pode gerar erros quando ambos os dispositivos estão em estados diferentes da operação, sendo necessário implementar o tratamento desse erro em ambos para que o fluxo não fique incorreto.

Como proposta de melhoria futura, a centralização das responsabilidades de comunicação no dispositivo central, a IHM, pode eliminar a possibilidade dos erros supracitados.

6.4 Vantagens obtidas

Dentre as vantagens atingidas com a mudança de estrutura na estação de serialização, pode-se destacar: a mudança da operação central para um sistema operacional Linux e o uso do *software* desenvolvido pela Pollux, o DMS.

Com a inserção do sistema operacional Linux, utiliza-se de uma plataforma estável e capaz de ser executada em diferentes dispositivos, facilitando futuras substituições e modificações na estrutura e possibilitando o uso de diversos *softwares* desenvolvidos para a plataforma. Além disso, surge a possibilidade de acesso remoto à estação, facilitando o suporte aos clientes por parte da empresa Pollux e a entrega de novas versões sem a necessidade de um técnico no local.

Ainda, a outra grande vantagem é o uso do Pollux DMS. Como a plataforma está presente em inúmeros outros clientes e é constantemente atualizada, há a garantia de que o sistema rodando na estação é estável. Além disso, por ser desenvolvido em linguagem *Python*, de código aberto, há a possibilidade de uso de inúmeras bibliotecas disponíveis para a mesma, além de eliminar a dependência do *software* proprietário, o qual pode não fornecer todas as funcionalidades desejadas, caso venham a ser implementadas.

6.5 Considerações Finais

Durante o período de atividades desenvolvidas na empresa Pollux Automation, em Joinville, considero que o aprendizado foi de grande proporção, utilizando de muitos conhecimentos obtidos durante a graduação para a compreensão e implementação de funcionalidades em diversas soluções para a indústria. Dentre elas, algumas estiveram muito presentes durante o período de trabalho, são elas:

- O funcionamento do *software* de CLPs, bem como a implementação de funções em linguagem *C++* adaptada a esse tipo de plataforma;
- A compreensão de protocolos de comunicação industriais, como *Modbus*, além de outros protocolos mais presentes na indústria 4.0, como as requisições HTTP;
- A compreensão de conceitos de modelagem de máquinas de estado;
- Conhecimentos a respeito da interpretação de diagramas elétricos, sensores e atuadores;
- Programação em linguagens orientadas a objetos.

Além disso, o fato deste trabalho ter sido realizado em ambiente empresarial permitiu que as experiências obtidas durante o curso fossem complementadas com a

dinâmica de trabalho em equipe, gerenciamento de prazos, atendimento ao cliente e obtenção de informações a respeito das soluções a serem desenvolvidas para o cliente, qualidades de grande valor para o desenvolvimento pessoal e profissional.

Por fim, destaca-se a qualidade de exigir a apresentação deste estágio final na forma de um projeto fechado, permitindo que todos os processos de um projeto sejam vistos do início ao fim em uma empresa, diferente de estágios que exigem atividades individuais e isoladas.

Referências

- 1 1 in 10 medical products in developing countries is substandard or falsified. 2017. Acesso em: Novembro de 2019. Disponível em: <<https://www.who.int/news-room/detail/28-11-2017-1-in-10-medical-products-in-developing-countries-is-substandard-or-falsified>>. Citado na página 15.
- 2 LEI nº 11.903 de 2009. 2009. Acesso em: Outubro de 2019. Disponível em: <http://www.planalto.gov.br/ccivil_03/_Ato2007-2010/2009/Lei/L11903.htm>. Citado 2 vezes nas páginas 15 e 31.
- 3 LEI nº 13.410 de 2013. 2016. Acesso em: Outubro de 2019. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2016/lei/L13410.htm>. Citado 3 vezes nas páginas 15, 19 e 31.
- 4 RESOLUçãO de Diretoria Colegiada, RDC nº54/2013. 2019. Acesso em: Outubro de 2019. Disponível em: <http://portal.anvisa.gov.br/documents/10181/2724161/RDC_54_2013_COMP.pdf/96e4684d-e243-4577-95b3-dad90ea94730>. Citado 2 vezes nas páginas 15 e 20.
- 5 V-PAK RT - Solução completa para Serialização e Rastreabilidade de Medicamentos. 2018. Acesso em: Novembro de 2019. Disponível em: <<https://www.youtube.com/watch?v=Xb0zbmjADWY>>. Citado 7 vezes nas páginas 16, 21, 22, 23, 25, 26 e 27.
- 6 WHAT is UML. 2019. Acesso em: Novembro de 2019. Disponível em: <<https://www.uml.org/what-is-uml.htm>>. Citado na página 37.
- 7 WAZLAWICK, R. S. *Análise e design orientado a objetos para sistemas de informação*. 3ª edição. ed. [S.l.]: Elsevier, 2015. Citado 2 vezes nas páginas 37 e 44.