

**UNIVERSIDADE FEDERAL DA FRONTEIRA SUL
CAMPUS CHAPECÓ
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

KAREN RUVET MENTGES

**AUTOMAÇÃO DE UM PASTEURIZADOR DE LEITE
COMO APLICAÇÃO DA INDÚSTRIA 4.0: UM SISTEMA
DE CONTROLE E MONITORAMENTO WEB**

**CHAPECÓ
2025**

KAREN RUVÉR MENTGES

**AUTOMAÇÃO DE UM PASTEURIZADOR DE LEITE
COMO APLICAÇÃO DA INDÚSTRIA 4.0: UM SISTEMA
DE CONTROLE E MONITORAMENTO WEB**

Trabalho de Conclusão de Curso apresentado ao Curso de Ciência da Computação da Universidade Federal da Fronteira Sul (UFFS), como requisito para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Luciano L. Caimi

**CHAPECÓ
2025**

Mentges, Karen Ruver

AUTOMAÇÃO DE UM PASTEURIZADOR DE LEITE COMO
APLICAÇÃO DA INDÚSTRIA 4.0: UM SISTEMA DE CON-
TROLE E MONITORAMENTO WEB / Karen Ruver Mentges
- 2025.

74 f.

Orientador: Luciano L. Caimi

Trabalho de Conclusão de Curso (Graduação)
- Universidade Federal da Fronteira Sul, Curso
de Ciência da Computação, Chapecó, SC, 2025.

1. Automação Industrial 2. Indústria 4.0
3. Sistema Web 4. Modbus TCP/IP 5. Pasteuriza-
ção de Leite I. Caimi, Luciano L., orient.
II. Universidade Federal da Fronteira Sul. III.
Título.

KAREN RUVET MENTGES

**AUTOMAÇÃO DE UM PASTEURIZADOR DE LEITE COMO APLICAÇÃO DA
INDÚSTRIA 4.0: UM SISTEMA DE CONTROLE E MONITORAMENTO WEB**

Trabalho de Conclusão de Curso apresentado ao Curso de Ciência da Computação da Universidade Federal da Fronteira Sul (UFFS), como requisito para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Luciano L. Caimi

Este Trabalho de Conclusão de Curso foi avaliado e aprovado pela banca avaliadora em: 10/12/2025.

BANCA AVALIADORA

Luciano L. Caimi - UFFS

Giancarlo D. Salton - UFFS

Adriano S. Padilha - Sanick Ltda

RESUMO

A Indústria 4.0 marcou uma evolução significativa para as empresas, mas nem todos os setores foram igualmente beneficiados. Pequenas agroindústrias de leite ainda enfrentam barreiras financeiras e tecnológicas para adotar tais inovações. Este trabalho apresenta o desenvolvimento de uma solução de automação e supervisão web para um Pasteurizador Lento Elétrico Compacto, visando modernizar a produção de pequenos e médios laticínios com baixo custo. A metodologia envolveu a substituição de controles eletromecânicos e analógicos por um sistema digital baseado em Controlador Lógico Programável (CLP) e Interface Homem-Máquina (IHM), utilizando lógica Ladder para o controle preciso dos ciclos de pasteurização, refrigeração e trato. Para habilitar a supervisão remota, foi desenvolvida uma arquitetura de comunicação IoT proprietária, composta por um Gateway em Python que realiza a aquisição de dados via protocolo Modbus TCP/IP e os persiste em banco de dados. O sistema é completado por uma aplicação Web Full-Stack (Node.js e React.js), que permite o monitoramento em tempo real, gestão de usuários, geração de relatórios e espelhamento remoto da IHM via noVNC. Os resultados demonstram a viabilidade técnica e econômica da integração entre automação industrial e sistemas web, proporcionando padronização operacional, rastreabilidade do processo e maior competitividade para o produtor, alinhando equipamentos convencionais aos conceitos de Indústria 4.0.

Palavras-Chave: Automação Industrial, Indústria 4.0, Sistema Web, Modbus TCP/IP, Pasteurização de Leite.

ABSTRACT

Industry 4.0 has marked a significant evolution for companies, but not all sectors have been equally benefited. Small dairy agroindustries still face financial and technological barriers to adopting such innovations. This work presents the development of an automation and web supervision solution for a Compact Electric Slow Pasteurizer, aiming to modernize the production of small and medium-sized dairy plants at low cost. The methodology involved replacing electromechanical and analog controls with a digital system based on a Programmable Logic Controller (PLC) and Human-Machine Interface (HMI), using Ladder logic for precise control of pasteurization, cooling, and treatment cycles. To enable remote supervision, a proprietary IoT communication architecture was developed, consisting of a Python Gateway that performs data acquisition via Modbus TCP/IP protocol and persists them in a database. The system is completed by a Full-Stack Web application (Node.js and React.js), which allows real-time monitoring, user management, report generation, and remote mirroring of the HMI via noVNC. The results demonstrate the technical and economic feasibility of integrating industrial automation and web systems, providing operational standardization, process traceability, and greater competitiveness for the producer, aligning conventional equipment with Industry 4.0 concepts.

Keywords: Industrial Automation, Industry 4.0, Web System, Modbus TCP/IP, Milk Pasteurization.

LISTA DE FIGURAS

1.1	Consumo aparente per capita de leite e derivados no Brasil de 2000 a 2023 em litros/habitante (Fonte: CILeite (2023))	11
1.2	Visão geral do sistema de automação (Fonte: Autor)	21
2.1	Fluxo do processo do leite pasteurizado (MILAGRES, 2022)	24
2.2	Pirâmide da automação industrial (NEPIN, 2023)	28
2.3	Esquema do pasteurizador alternativo de leite com seus componentes (BARBOSA et al., 2018)	30
2.4	Fluxograma do processo do protótipo (BARBOSA et al., 2018)	31
2.5	Fluxograma detalhado da área 1000 - Recepção de Leite e Resfriamento (MILAGRES, 2022)	32
2.6	Fluxograma detalhado da área 2000 - Estocagem e Pasteurização de Leite (MILAGRES, 2022)	33
2.7	Arquitetura da rede de automação (MILAGRES, 2022)	34
3.1	Pasteurizador lento - vista frontal (Fonte: Autor)	37
3.2	Arquitetura do sistema de automação (Fonte: Autor)	41
3.3	Localização dos dispositivos no pasteurizador lento (Fonte: Autor)	42
3.4	Fluxograma da automação (Fonte: Autor)	43
3.5	Função da pasteurização (Fonte: Autor)	44
3.6	Função da refrigeração (Fonte: Autor)	45
3.7	Função do trato parte 1 (Fonte: Autor)	46
3.8	Função do trato parte 2 (Fonte: Autor)	46
3.9	CCW (Fonte: Autor)	47
3.10	DOPSoft (Fonte: Autor)	49
3.11	Tela inicial da IHM (Fonte: Autor)	50
3.12	Tela de andamento da IHM (Fonte: Autor)	50
3.13	Acionamento do envio das mensagens do tipo status (Fonte: Autor)	52
3.14	Envio das mensagens do tipo status (Fonte: Autor)	52
3.15	Erro no envio das mensagens do tipo status (Fonte: Autor)	52
3.16	Acionamento do envio de andamento (Fonte: Autor)	53
3.17	Conteúdo das mensagens de status (Fonte: Autor)	53
4.1	Diagrama EER (Fonte: Autor)	60

4.2	Layout do PDF (Fonte: Autor)	61
4.3	Tela inicial (Fonte: Autor)	62
4.4	Tela de relatórios (Fonte: Autor)	63

LISTA DE SIGLAS

ASI – Atuador Sensor Interface

CLP – Controlador Lógico Programável

IHM – Interface Homem-Máquina

IOT – Internet das Coisas

IIOT – Internet Industrial das Coisas

LCD – Liquid Crystal Display

LED – Diodo Emissor de Luz

SIM – Serviço de Inspeção Municipal

SIE – Serviço de Inspeção Estadual

SIF – Serviço de Inspeção Federal

SISBI-POA – Sistema Brasileiro de Inspeção de Produtos de Origem Animal

UHT – Ultra High Temperature

VCA – Volts em Corrente Alternada

VCC – Volts em Corrente Contínua

SUMÁRIO

1	INTRODUÇÃO	11
1.1	CONTEXTUALIZAÇÃO	11
1.2	PROBLEMA	12
1.3	OBJETIVO GERAL	13
1.4	OBJETIVOS ESPECÍFICOS	13
1.5	JUSTIFICATIVA	14
1.6	METODOLOGIA	14
1.6.1	FASE I: ANÁLISE, ESPECIFICAÇÃO E REQUISITOS	15
1.6.2	FASE II: PROJETO DA ARQUITETURA	16
1.6.3	FASE III: FASE DE IMPLEMENTAÇÃO E INTEGRAÇÃO	18
1.6.4	FASE IV: TESTES E VALIDAÇÃO	20
1.6.5	FASE V: ESCRITA DA MONOGRAFIA E APRESENTAÇÃO DO TCC	20
1.7	VISÃO GERAL DO SISTEMA PROPOSTO	20
1.7.1	CAMADA DE CONTROLE E AUTOMAÇÃO	20
1.7.2	CAMADA DE INTEGRAÇÃO (GATEWAY)	21
1.7.3	CAMADA DE SUPERVISÃO E MONITORAMENTO	21
2	FUNDAMENTAÇÃO TEÓRICA E TRABALHOS RELACIONADOS	23
2.1	FLUXO DO PROCESSO DE INDUSTRIALIZAÇÃO DO LEITE	23
2.2	INDÚSTRIA 4.0	25
2.3	AUTOMAÇÃO INDUSTRIAL	27
2.4	MODBUS TCP/IP	28
2.5	TRABALHOS RELACIONADOS	30
2.5.1	AUTOMAÇÃO DE UM PROCESSO ALTERNATIVO DA PASTEURIZAÇÃO DO LEITE UTILIZANDO AÇÃO DE CONTROLE ON-OFF	30
2.5.2	PROPOSTA DE AUTOMAÇÃO DE UMA PLANTA DE PRODUÇÃO DE IOGURTE	31
3	AUTOMAÇÃO DO PASTEURIZADOR	36
3.1	PASTEURIZADOR LENTO ELÉTRICO COMPACTO DE LEITE PARA BEZERRAS	36

3.1.1	PROCESSOS REALIZADOS PELO EQUIPAMENTO	36
3.1.2	OPERAÇÃO DO EQUIPAMENTO	37
3.2	REQUISITOS DA AUTOMAÇÃO	38
3.2.1	REQUISITOS FUNCIONAIS	38
3.2.2	REQUISITOS NÃO FUNCIONAIS	39
3.3	ARQUITETURA DE HARDWARE	39
3.4	ARQUITETURA DE SOFTWARE	42
3.4.1	PROGRAMAÇÃO DO CLP	46
3.4.2	PROGRAMAÇÃO DA IHM	48
3.4.3	GATEWAY MODBUS TCP/IP	51
4	SUPERVISÃO DA AUTOMAÇÃO: SISTEMA WEB	56
4.1	O SISTEMA WEB	56
4.2	REQUISITOS DO SISTEMA WEB	56
4.2.1	REQUISITOS FUNCIONAIS	57
4.2.2	REQUISITOS NÃO FUNCIONAIS	57
4.3	ARQUITETURA DO SISTEMA WEB	58
4.3.1	BANCO DE DADOS	58
4.3.2	BACK-END	59
4.3.3	FRONT-END	61
4.3.4	GATEWAY - COMUNICAÇÃO ENTRE CLP E SISTEMA WEB	63
4.3.5	MODBUS TCP/IP	64
4.3.6	FUNCIONALIDADES	69
5	CONSIDERAÇÕES FINAIS	71
5.1	TRABALHOS FUTUROS	72
	REFERÊNCIAS	73

1. INTRODUÇÃO

1.1 Contextualização

A indústria de processamento de leite é altamente impactada pelo crescimento da população brasileira e pelo aumento do consumo de leite e seus derivados. O mercado precisou se expandir para atender à demanda e, para isso, foram adotadas tecnologias que auxiliam no processo, com o objetivo de agilizar a produção e reduzir os custos (SIQUEIRA, 2019).

A Figura 1.1 apresenta o crescimento do consumo de leite de 2000 a 2013, e a partir desse intervalo é observada uma estabilização do consumo, em média, 180 litros por habitante. O fator econômico é o principal responsável por essa estabilização (SIQUEIRA, 2019). Segundo a EMBRAPA (2025), houve uma redução do número de produtores, da ordem de 13% entre os anos de 2006 e 2017, e também uma forte redução do número de vacas ordenhadas, da ordem de 33% entre os anos de 2011 e 2023.

Entretanto, é importante destacar que a produtividade do animal no período de 2000 a 2023 cresceu 104% ao atingir 2.254 litros/vaca/ano. A Embrapa explica que, mesmo com a estagnação do consumo de leite e derivados no Brasil, a demanda é suprida com a exportação, principalmente pela Argentina e Uruguai. O gráfico abaixo evidencia a importância da presença dos produtos lácteos na alimentação da população. Pode-se destacar que o leite é uma das principais fontes de proteínas, sais minerais e gorduras, rico em cálcio e fósforo.

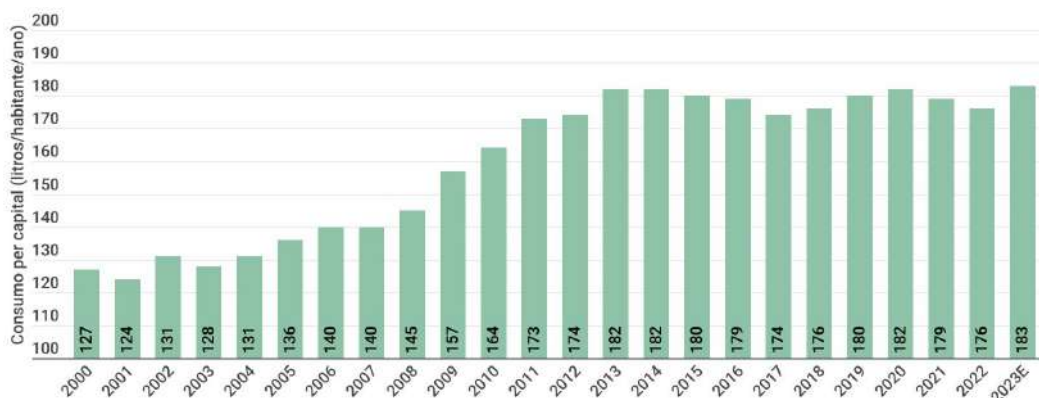


Figura 1.1 – Consumo aparente per capita de leite e derivados no Brasil de 2000 a 2023 em litros/habitante (Fonte: CILeite (2023))

O Brasil possui agroindústrias de leite espalhadas por todo o território, nas quais ocorre a industrialização do leite, transformando-o em produtos como leite pas-

teurizado, leite UHT, iogurte, queijo, requeijão, nata, entre outros. Cada derivado segue um procedimento de produção específico que, além dos ingredientes e da receita, exige o uso de equipamentos adequados para cada etapa do processo.

A escolha dos equipamentos depende principalmente da quantidade de leite industrializado por dia. Essa informação é utilizada para calcular a capacidade dos equipamentos necessários para atender à demanda da agroindústria. Outro ponto a considerar é a tecnologia dos equipamentos: quanto maior for a produção, busca-se melhores tecnologias; para pequenas produções, o custo-benefício é mais relevante. Os equipamentos selecionados neste estudo possuem sistemas de controle eletromecânicos e analógicos típicos das tecnologias da Indústria 2.0, limitando o monitoramento e transferência de dados.

Cabe ressaltar que, para a comercialização do produto, ainda é necessário realizar uma etapa (MAPA, 2017): submeter o produto ao serviço de inspeção. A escolha do serviço dependerá da abrangência comercial desejada: SIM (Serviço de Inspeção Municipal), SIE (Serviço de Inspeção Estadual), SIF (Serviço de Inspeção Federal) ou SISBI-POA (Serviço Brasileiro de Inspeção de Produtos de Origem Animal).

1.2 Problema

Pequenos produtores de leite e pequenos laticínios enfrentam dificuldades para entrar no mercado de industrialização do leite devido à alta competitividade e à necessidade de altos investimentos. Assim, iniciam suas produções de maneira bastante manual, o que impacta negativamente tanto a produtividade quanto a lucratividade (EMBRAPA, 2024).

Para reduzir o trabalho manual em um laticínio, é recomendada a aquisição de equipamentos específicos para o produto a ser processado. Por exemplo, o pasteurizador é utilizado em todas as linhas de industrialização do leite, enquanto o tanque para fabricação de queijo faz parte apenas da linha de queijos. A aquisição desses equipamentos contribui para a otimização da produção, que é comumente o objetivo do empresário que realiza esses investimentos.

Mesmo com esses equipamentos, a produção ainda não será automatizada em nível suficiente para dispensar a necessidade de pessoas executando os processos. Entretanto, para realizar uma automação completa em um laticínio, os custos se tornam inacessíveis para quem trabalha com pequenas produções. Por isso, é necessário desenvolver técnicas e ferramentas que adaptem os equipamentos à Indústria 4.0, focando em custo-benefício para pequenos laticínios.

A indústria tem sido uma grande apoiadora da automação de equipamentos para a industrialização de alimentos, uma vez que seu objetivo é atender às demandas do mercado. Ao longo dos anos, as melhorias implementadas desde a criação dessas tecnologias têm se sofisticado a tal ponto que os equipamentos agora funcionam de forma independente, eliminando a necessidade de um operador. Isso também resulta em maior uniformidade no processo, devido à utilização de padrões estabelecidos.

A Indústria 4.0 tem como objetivo melhorar a eficiência e produtividade dos processos, onde, a partir da Internet das Coisas (IoT) e armazenamento em nuvem, são desenvolvidos projetos automatizados para empresas, apresentando informações aos gestores em tempo real das suas produções. Essas informações podem ser utilizadas na tomada de decisões, auxiliando as empresas a reduzir os custos e aumentar os lucros.

No Brasil, é necessária a obtenção do serviço de inspeção como SIM, SIE, SIF ou SISBI-POA para a comercialização de produtos (MAPA, 2017). Nesse contexto, a aquisição de equipamentos tornou-se obrigatória, pois assegura que o produto final atenda aos requisitos exigidos pela vigilância sanitária e outras organizações. É importante destacar que a automação dos equipamentos contribui ainda mais para atender esses requisitos, uma vez que a linha de produção segue sempre um padrão pré-definido para a execução dos processos.

1.3 Objetivo Geral

Desenvolver a automação de um pasteurizador lento de leite, juntamente com o desenvolvimento de um sistema de controle e monitoramento Web, para auxiliar os operadores e gestores dentro do fluxo de produção de um laticínio.

1.4 Objetivos Específicos

- Especificar os requisitos funcionais e não funcionais do sistema de automação e supervisão, com base na análise da operação manual do pasteurizador;
- Projetar e implementar a lógica de controle (CLP/IHM), garantindo a rastreabilidade e a aderência aos padrões de qualidade exigidos;
- Desenvolver um gateway de comunicação eficiente e de baixo custo, baseado no protocolo Modbus TCP/IP, para a transferência contínua de dados operacionais;
- Implementar a arquitetura full-stack para o Sistema Web de Monitoramento Remoto, habilitando a visualização e o histórico de dados operacionais (I4.0);

- Validar a funcionalidade e a integração do sistema, verificando a precisão dos dados coletados e a conformidade do processo automatizado com os requisitos de pasteurização.

1.5 Justificativa

Como mostrado na Figura 1.1, o mercado de industrialização do leite tem apresentado um crescimento significativo devido ao aumento do consumo de lácteos nos últimos anos. Para acompanhar esse crescimento, os grandes laticínios têm investido na automação de seus maquinários, com o objetivo de otimizar processos e aumentar a produção. No entanto, os pequenos laticínios ainda operam com uma alta dependência de trabalho manual, o que eleva os custos de produção e dificulta sua competitividade no mercado (EMBRAPA, 2024).

A automação desses equipamentos pode contribuir de forma significativa para os processos realizados no laticínio do cliente, proporcionando maior segurança, agilidade, autonomia, redução da mão de obra e melhorias na qualidade do produto final. Para as empresas fabricantes de equipamentos, a automação favorecerá sua permanência no mercado, impulsionando novas oportunidades de vendas, além de resolver uma necessidade crucial de seus clientes (GROOVER, 2015).

Para adaptar os equipamentos à Indústria 4.0, é necessário realizar um estudo voltado ao desenvolvimento de uma solução tecnológica que seja economicamente viável para pequenos e médios laticínios. É importante destacar que, ao longo dos anos, os dispositivos de processamento e comunicação baratearam significativamente, permitindo que o processo de automação e comunicação se expandisse sem que os custos se tornassem elevados e inacessíveis. Isso facilitou a adoção de tecnologias avançadas em diversos setores.

Neste contexto, este estudo é baseado em uma necessidade enfrentada pela empresa Suck Milk, localizada em Nova Erechim – SC. Ela atua no mercado desde 1995 e é especializada na fabricação de equipamentos para a agroindústria leiteira. A empresa busca a automação de seus equipamentos para inseri-los na Indústria 4.0 e, conseqüentemente, obter mais eficiência e qualidade nos processos de industrialização do leite.

1.6 Metodologia

Conforme mencionado anteriormente, este trabalho é baseado em uma necessidade enfrentada pela empresa Suck Milk. Assim, foi definido juntamente com a

empresa que será realizada a automação do Pasteurizador Lento Elétrico Compacto de Leite para Bezerras, com o envio das informações do equipamento para um servidor Web. Juntamente, será desenvolvido um sistema de controle e monitoramento Web, que irá receber e armazenar as informações do equipamento, apresentar essas informações em relatórios e também permitir o controle do equipamento por meio de uma conexão Ethernet.

1.6.1 Fase I: Análise, Especificação e Requisitos

A fase inicial de pesquisa e contextualização foi fundamentada em coleta de dados primários e workshops junto à empresa fabricante, visando compreender a operação da agroindústria leiteira e as características técnicas dos equipamentos. Esta análise subsidiou a definição do problema e dos requisitos do projeto, priorizando soluções que garantissem a otimização da produção e a melhoria da experiência do usuário no processo.

Com o projeto técnico em mãos, foram analisadas a composição, o passo a passo da fabricação e a localização de cada componente dos equipamentos. Assim, facilita-se a compreensão do funcionamento de cada processo desempenhado, onde é possível identificar os componentes e a sua responsabilidade dentro do processo.

Em conjunto com a empresa, ficou decidido que inicialmente seria realizada a automação do Pasteurizador Lento Elétrico Compacto de Leite para Bezerras, com a inserção de um CLP (Controlador Lógico Programável) para controlar os periféricos e gerenciar os comandos necessários ao funcionamento do equipamento. Além disso, será instalado um sensor de temperatura para capturar os dados essenciais que servirão como variáveis de referência no código de execução do CLP.

O pasteurizador é composto por chaves contactoras, interruptores, disjuntor e fios elétricos, componentes fundamentais para a montagem do painel elétrico do equipamento. No painel, também encontramos um programador de horário analógico, que define o tempo de aquecimento e o tempo restante é destinado ao resfriamento do leite, além de três termostatos: o primeiro define a temperatura de aquecimento, o segundo a temperatura de resfriamento e o terceiro a temperatura ideal para o trato das bezerras.

Com a automação desse equipamento, o programador e os termostatos serão substituídos por um CLP e uma IHM (Interface Homem-Máquina), permitindo que o processo de pasteurização, refrigeração e trato ocorra de forma automática. As configurações necessárias estarão disponíveis na IHM, facilitando a visualização das informações e eliminando a necessidade de acionamento manual dos comandos.

A partir disso, foram identificados os componentes que devem ser trocados para implementar o CLP no pasteurizador. Foi concluído que o painel atual irá permanecer da mesma forma, pois irá se comunicar bem com o CLP, e com a adição da IHM é possível fazer a remoção dos três termostatos e o programador de horário, que ficam dispostos na parte de fora do painel para o manuseio do operador.

Com a adição do CLP e da IHM, acontece o desenvolvimento do código que realiza o acionamento de cada componente de acordo com o processo da pasteurização, seguindo os protocolos que a empresa já possui. Para esse desenvolvimento, se faz necessário o estudo completo sobre os componentes já existentes no painel elétrico e os que serão adicionados.

1.6.2 Fase II: Projeto da Arquitetura

A fase de Projeto da Arquitetura concentrou-se na seleção tecnológica. Para o controle local, foi definida uma arquitetura baseada em CLP da Rockwell Automation para garantir robustez industrial e elaborado o fluxograma de processos da automação. A arquitetura de software do sistema de supervisão foi projetada como full-stack (Node.js/React.js) para assegurar escalabilidade e responsividade na interface. O ponto central desta fase foi o projeto da arquitetura de comunicação (*gateway*), definindo o Mapa de Endereçamento Modbus para a troca de dados entre o CLP e o servidor, conforme detalhado no Capítulo 4.

A definição dos processos do equipamento é de grande importância para o projeto, é essa ideia que irá guiar a programação de toda a automação. A partir de uma reunião da equipe de projetos e desenvolvimento, foram formalizados os processos e os requisitos que a empresa possui sobre o pasteurizador.

A partir desse levantamento, foram definidos os cinco processos presentes na automação e monitoramento:

1. Pasteurização → Refrigeração → Trato;
2. Refrigeração → Pasteurização → Trato;
3. Pasteurização;
4. Refrigeração;
5. Trato.

Os processos 1 e 2 são automáticos, então as trocas de fluxo são realizadas após a finalização de cada etapa, levando em consideração o horário programado

para o início do trato, que é escolhido antes do equipamento iniciar a operação. Os processos 3, 4 e 5 são manuais, assim, eles iniciam e finalizam o processo sem troca de fluxo. O fluxograma (Figura 3.4) ilustra o fluxo de cada processo, com o objetivo de auxiliar na programação do CLP. Esse fluxograma será detalhado no Capítulo 3.

A definição da arquitetura do Sistema Web é o que viabiliza a supervisão remota do pasteurizador, um requisito fundamental para a aplicação dos conceitos da Indústria 4.0. A arquitetura foi projetada em um modelo de três camadas distribuídas, complementada por uma camada de *Gateway* de Comunicação dedicada à integração com o hardware industrial.

1. Camada de Aquisição e Gateway

A principal decisão de projeto nesta fase foi a criação de um *Gateway* de Comunicação para mediar a comunicação entre o CLP e a Camada de Dados. Este módulo foi desenvolvido em Python e opera via Modbus TCP/IP.

Função: O *Gateway* é responsável por estabelecer a conexão com o CLP, realizar a leitura contínua (polling) dos registradores de memória definidos no Mapa de Endereçamento Modbus e traduzir os dados binários para o formato adequado. Para este módulo, não foi utilizada nenhuma biblioteca existente, sendo o desenvolvimento do mesmo uma das contribuições do trabalho.

Mecanismo: A leitura de dados é realizada de forma assíncrona, garantindo que o ciclo de controle do CLP não seja comprometido pelo tráfego de rede do sistema de supervisão. Após a aquisição, o *gateway* insere os dados de tempo real (e.g., temperatura, status) no Banco de Dados.

2. Camada de Lógica de Negócios e Persistência

Esta camada é o coração do sistema de supervisão e é responsável pelo tratamento e disponibilização dos dados.

Tecnologia *back-end*: A plataforma Node.js em conjunto com o framework Express foi selecionada para o *back-end* (servidor de aplicação), devido à sua capacidade de lidar com requisições concorrentes e escalabilidade, essenciais para o consumo contínuo de dados.

API RESTful: O *back-end* expõe uma API que padroniza o acesso aos dados para o *front-end* e, futuramente, para outros sistemas. Isso garante a separação de responsabilidades e a manutenção modular do sistema.

Persistência de Dados: O Banco de Dados foi definido para armazenar o histórico dos ciclos de pasteurização, eventos e alarmes, permitindo a análise retrospectiva do processo.

3. Camada de Apresentação (Interface)

A camada de apresentação foi projetada com foco na usabilidade e acessibilidade remota.

Tecnologia *front-end*: A biblioteca React.js foi escolhida para o desenvolvimento do *front-end*, permitindo a criação de uma Single Page Application (SPA). Esta arquitetura assegura uma experiência fluida e responsiva ao usuário, com carregamento otimizado de informações.

Funcionalidade: O design da interface foi focado na visualização de dados, fornecendo ao gestor uma visão clara e em tempo real do estado operacional do pasteurizador.

1.6.3 Fase III: Fase de Implementação e Integração

A fase de implementação e integração envolveu o desenvolvimento de todos os componentes de hardware e software. No ambiente industrial, o CLP foi programado com a lógica de controle utilizando a linguagem Ladder, enfatizando a segurança e a rastreabilidade das operações. A IHM foi desenvolvida para oferecer uma interface intuitiva ao operador. Paralelamente, a integração CLP-Web foi implementada através do servidor Python (*Gateway*), garantindo que os dados operacionais fossem transferidos e persistidos de forma confiável, permitindo o consumo pela API REST do *back-end* e a visualização no *front-end*.

Inicialmente, a programação do CLP, com a estruturação do código em ladder para o CLP, foi realizada de forma que os fluxos do programa ficavam expostos e lineares, seguindo o fluxograma desenvolvido.

Para a realização da programação do CLP, é necessário baixar o programa específico fornecido pelo fabricante (Connected Components Workbench). Após a instalação completa, ainda era preciso configurar alguns dados básicos para os testes no dispositivo. Em seguida, foram criadas as primeiras linhas do programa, as restantes foram criadas por partes, e a cada parte eram feitos testes para entender o comportamento do programa.

Após a conclusão da criação do código base para o CLP, foram realizados diversos testes e ajustes conforme a necessidade para que o equipamento operasse da forma correta, seguindo o fluxograma. Finalizados todos esses ajustes, foram realizados novos testes para ter a comprovação do correto funcionamento de todos os processos do equipamento, além das demais funções criadas. Testado e aprovado em conjunto com o engenheiro, a programação do CLP é concluída.

A programação da IHM tem como objetivo obter um design moderno e minimalista, mantendo a funcionalidade e usabilidade. As telas da IHM foram desenvolvidas no Figma. Essas telas foram utilizadas como fundo para inserir os dados dinâmicos e botões em cima, facilitando a criação do layout. A criação ocorreu conforme a necessidade durante o desenvolvimento do programa, e ao todo foram criadas 67 telas, incluindo as telas do manual.

O desenvolvimento e os testes foram realizados em conjunto com o CLP, para entender o comportamento de comunicação entre os dois. Assim como no CLP, diversos ajustes foram feitos após a conclusão do código base, além da inserção de mais funções para compor o programa final.

Ao fim, foram realizados novos testes para ter a comprovação do correto funcionamento de todos os processos do equipamento, além das demais funções criadas. Testado e aprovado em conjunto com o engenheiro, a programação da IHM é concluída.

Na automação, foi mantido apenas um sensor de temperatura. Os dados obtidos pelo sensor são de extrema importância para os processos realizados pelo equipamento, então esse valor é armazenado no banco de dados. As variáveis contidas dentro da lógica do programa do CLP também necessitam ser armazenadas no banco de dados, por exemplo, processo selecionado, etapa em execução, temperaturas programadas, status dos atuadores, entre outros dados.

Esse envio de dados é realizado via protocolo Modbus pelo CLP ao *Gateway* desenvolvido em Python, responsável por receber, organizar os dados e inserir no banco a partir de requisições SQL assíncronas.

Os dados registrados no banco podem ser acessados via Sistema Web por meio de relatórios. O cliente poderá visualizar os dados registrados dos seus equipamentos, e a empresa fabricante poderá visualizar os dados de todos os equipamentos cadastrados no sistema, através do número de série do equipamento. Esses relatórios podem contribuir significativamente para a empresa, pois apresentam dados relevantes para a manutenção do equipamento caso ocorra um problema na operação.

Finalmente, as informações da automação devem ser disponibilizadas no Sistema Web. A ideia para o Sistema Web é permitir que o operador possa monitorar e controlar o equipamento sem a necessidade de estar em frente ao equipamento. E para o gestor, permitir que possa obter dados do funcionamento do equipamento para acompanhar os resultados e tomar decisões a partir dele. Para a empresa fabricante, este registro de dados é de grande valia, pois pode colaborar com manutenções e melhorias futuras nos equipamentos.

Para o desenvolvimento do código-fonte foram selecionados: React.js para o *front-end*, Node.js para o *back-end*, Express e Sequelize para realizar a comunicação entre o banco de dados e o *back-end*, Python para o servidor Modbus, e por fim, MySQL para o banco de dados.

1.6.4 Fase IV: Testes e validação

Por fim, a fase de testes e validação foi executada para verificar a aderência do sistema aos requisitos especificados. Foram realizados testes de integração para confirmar a correta comunicação e precisão dos dados transferidos via Modbus TCP/IP.

Em seguida, o sistema completo foi submetido a testes de sistema para validar a estabilidade e precisão do controle de temperatura durante o ciclo de pasteurização completo, bem como a transferência de dados e a visualização dos dados do processo e etapas da automação no Sistema Web.

1.6.5 Fase V: Escrita da monografia e apresentação do TCC

Esta fase abrange a escrita final e formalização da monografia, garantindo que todo o processo metodológico, desde a especificação até a validação, esteja devidamente documentado, culminando com a defesa pública do Trabalho de Conclusão de Curso perante a banca examinadora.

1.7 Visão Geral do Sistema Proposto

O sistema proposto estabelece a interconexão entre o equipamento industrial (o domínio físico) e a infraestrutura de Tecnologia da Informação (o domínio digital). A arquitetura do sistema foi projetada em camadas distintas, visando modularidade, confiabilidade e facilidade de manutenção.

O sistema pode ser dividido em três componentes principais interligados, conforme ilustrado na Figura 1.2.

1.7.1 Camada de Controle e Automação

Esta camada é responsável pela execução em tempo real das funções de controle do pasteurizador. É composta pelos sensores (temperatura) e atuadores (resistências, válvulas) conectados a um Controlador Lógico Programável (CLP). O



Figura 1.2 – Visão geral do sistema de automação (Fonte: Autor)

CLP executa a lógica de controle desenvolvida (malha de controle de temperatura e sequenciamento de processo), substituindo o controle manual ou analógico anterior. Uma Interface Homem-Máquina (IHM) é integrada nesta camada para permitir a operação e visualização local dos parâmetros do processo.

1.7.2 Camada de Integração (Gateway)

Atuando como a ponte de comunicação entre o ambiente industrial (CLP) e a rede de dados (TI), o *Gateway* é um módulo de software desenvolvido em Python. Utilizando o protocolo Modbus TCP/IP, o *gateway* realiza o polling periódico dos registradores do CLP para aquisição de dados. Este componente garante que a transferência de dados para a nuvem não interfira na performance crítica da lógica de controle. Após a aquisição, o *gateway* persiste os dados operacionais no Banco de Dados.

1.7.3 Camada de Supervisão e Monitoramento

Esta camada é responsável pelo armazenamento, processamento e apresentação dos dados coletados remotamente. É composta por uma API RESTful (*back-end*), responsável pela lógica de negócios e consulta ao Banco de Dados, e pela Aplicação Web (*front-end*). A aplicação Web (desenvolvida com React) permite ao usuário (operador ou gestor) monitorar o estado do pasteurizador, visualizar o histórico de eventos e analisar o desempenho dos ciclos de pasteurização, acessível de qualquer dispositivo conectado à internet a partir do servidor Web.

Dessa forma, o sistema garante a precisão e padronização do processo de pasteurização (CLP) ao mesmo tempo que oferece a rastreabilidade e a capacidade analítica exigidas pelos paradigmas da Indústria 4.0 (Sistema Web).

Estrutura do Trabalho

O presente Trabalho de Conclusão de Curso está organizado em cinco capítulos, dispostos de maneira lógica para apresentar o desenvolvimento e a validação do sistema de automação e monitoramento remoto.

O **capítulo 1** (Introdução) apresenta o contexto da pesquisa na agroindústria de laticínios, define o problema a ser resolvido, estabelece os objetivos e justifica a relevância do estudo, culminando na apresentação da metodologia empregada no desenvolvimento.

O **capítulo 2** (Fundamentação Teórica) provê o suporte conceitual para o desenvolvimento. Ele aborda os pilares da pesquisa, as tecnologias da Indústria 4.0 e os conceitos de Automação e Redes Industriais (Modbus TCP/IP) e apresenta os trabalhos relacionados.

O **capítulo 3** (Automação do Pasteurizador) detalha o projeto e a implementação do controle local. Neste item é realizada a apresentação do equipamento (Pasteurizador Lento), à especificação de seus Requisitos de Hardware e Software, e à descrição da Lógica de Controle implementada no Controlador Lógico Programável (CLP) e na Interface Homem-Máquina (IHM).

O **capítulo 4** (Supervisão da Automação: Sistema Web) foca na solução de monitoramento remoto. São apresentadas a Arquitetura do Sistema Web, a escolha das tecnologias full-stack (Node.js/React.js) e o projeto do *Gateway* de Comunicação (Python), que estabelece a ponte entre o ambiente industrial (OT) e o ambiente de tecnologia da informação (IT).

Por fim, o trabalho é concluído no **capítulo 5** (Conclusão), onde são sintetizadas as contribuições do projeto e apresentadas sugestões para trabalhos futuros, visando a expansão e o aprimoramento da solução.

2. FUNDAMENTAÇÃO TEÓRICA E TRABALHOS RELACIONADOS

Neste capítulo serão apresentados fundamentos e conceitos utilizados no trabalho, bem como alguns trabalhos relacionados que exploram problemas e metodologias semelhantes. O objetivo das seções não é detalhar as características, métodos e conceitos envolvidos, mas fornecer ao leitor um conhecimento introdutório para facilitar o entendimento do trabalho em desenvolvimento.

2.1 Fluxo do processo de industrialização do leite

Os produtos como leite pasteurizado, iogurte, bebida láctea, queijo, manteiga, nata e outros são obtidos por meio da industrialização do leite. Cada um deles é produzido através de um processo específico, que deve seguir um conjunto de ações ordenadas para obter o produto no final. No decorrer desta seção, serão exemplificadas as ações realizadas no processo de obtenção do leite pasteurizado.

Para a industrialização do leite, é necessário o uso de equipamentos específicos, de acordo com o tipo de produto desejado, a capacidade de produção diária e o tipo de rede elétrica. Para a linha de leite, são indicados os seguintes equipamentos: resfriador, pasteurizador, gerador de água quente, tanque banca de gelo, desnata-deira, embaladeira e câmara fria; e para a linha de queijo: resfriador, pasteurizador, gerador de água quente, tanque banca de gelo, tanque para fabricação de queijo (para grandes produções, utiliza-se a queijomatic), mesas, prensa pneumática com formas, tanque para salga e câmara fria.

Para exemplificar, a seguir é descrito brevemente o processo do leite pasteurizado (Figura 2.1). O processo de industrialização do leite inicia-se com a coleta por um caminhão isotérmico nas propriedades próximas ao laticínio. Ao chegar à plataforma de recepção, são realizadas análises de qualidade e, após aprovação, o leite é resfriado e armazenado a uma temperatura entre 4 °C e 5 °C. Em seguida, o leite passa pelo processo de desnate, onde a gordura é separada de acordo com o tipo de leite a ser produzido, como integral, semidesnatado ou desnatado (SILVA; SILVA; FERREIRA, 2012).

Após o desnate, o leite é pasteurizado, um processo que envolve aquecimento e resfriamento para eliminar microrganismos patogênicos. Dependendo do volume de processamento, usa-se um pasteurizador lento ou um pasteurizador rápido. Em seguida, o leite é transferido para a embaladeira automática, onde é embalado e,

posteriormente, armazenado em uma câmara fria até a comercialização. Após a pasteurização, o leite pode ser usado para fabricação de derivados do leite, por exemplo: queijos, iogurte, nata, manteiga, entre outros. Pode também ser utilizado para o trato dos animais (SILVA; SILVA; FERREIRA, 2012).



Figura 2.1 – Fluxo do processo do leite pasteurizado (MILAGRES, 2022)

No fluxo acima, são apresentadas as etapas do processo de leite pasteurizado, juntamente com os equipamentos utilizados em cada etapa. Para a fabricação desses equipamentos, utiliza-se o aço inox, uma matéria-prima que possui resistência à corrosão e à proliferação de microrganismos, é de fácil limpeza, além de apresentar alta durabilidade.

Esses equipamentos e maquinários ficam dispostos nas agroindústrias de modo a atender esse fluxo. Para a construção de uma agroindústria, é necessária uma gama de documentos, incluindo memoriais e plantas baixas que descrevem como será o projeto do início ao fim, devendo estar de acordo com as normas do Ministério da Agricultura, Pecuária e Abastecimento (MAPA). Dependendo da norma, pode ser solicitado um conjunto adicional de documentos para complementar o projeto. Sem a liberação do projeto, a construção não pode ser iniciada.

Em seguida, são apresentadas as ações presentes no processo da industrialização do leite para obtenção do leite pasteurizado, baseadas no material realizado pelo Instituto Federal do Paraná (SILVA; SILVA; FERREIRA, 2012), de uma forma mais detalhada do que anteriormente:

Recepção do Leite

Inicialmente, um caminhão com tanque isotérmico coleta o leite nas propriedades mais próximas do laticínio. Após finalizar a coleta, o caminhão dirige-se à plataforma de recepção do laticínio, onde é coletada uma amostra do leite para a realização das seguintes análises: temperatura, alizarol, acidez, crioscopia, pH, densidade, gordura, proteína, lactose e extrato seco. Após a confirmação das análises e a liberação para uso, o leite é transferido por uma bomba sanitária até o resfriador, que o resfria e mantém a uma temperatura de 4 °C a 5 °C. O leite permanecerá armazenado no resfriador até o início da próxima etapa.

Desnate

Dando continuidade ao processo, o leite é encaminhado por tubulações do resfriador até a desnatadeira, onde ocorre o processo de desnate. A desnatadeira separa a gordura por meio de centrifugação e diferença de densidade dos dois produtos. Na desnatadeira, há duas saídas: uma para o leite desnatado e outra para a gordura, que pode ser utilizada posteriormente na fabricação de derivados. A quantidade de gordura a ser retirada do leite depende do tipo de leite que o laticínio deseja comercializar: leite integral possui teor de gordura superior ou igual a 3%, leite semidesnatado possui teor de gordura entre 0,6% e 2,9%, e o leite desnatado possui teor de gordura inferior a 0,5%.

Pasteurização

Após o desnate, o leite passa pela pasteurização, um processo térmico utilizado para eliminar microrganismos patogênicos. Esse processo pode ser realizado de duas formas: conforme estabelecido pelo MAPA, lenta, em que o leite é aquecido entre 62 °C e 65 °C pelo período de 30 minutos; ou rápida, em que o leite é aquecido entre 72 °C e 75 °C pelo período de 15 a 20 segundos. Essa etapa é essencial para evitar riscos à saúde pública. Em seguida, o leite é resfriado entre 4 °C e 5 °C para a estocagem.

Envase e estocagem

O leite é transferido através de uma bomba para a embaladeira automática, que envasa em embalagens plásticas flexíveis, conhecidas como “barriga mole”. Após esse processo, o leite é levado para uma câmara fria, onde permanecerá estocado e resfriado até a comercialização.

2.2 Indústria 4.0

A Indústria 4.0 é o termo usado para a Quarta Revolução Industrial que começou por volta de 2010. Ele foi utilizado formalmente pela primeira vez na Feira de Hannover, em 2011, na Alemanha, em um contexto que destacava a crescente automação, digitalização e o uso de tecnologias avançadas no setor industrial.

A Indústria 4.0 procura desenvolver indústrias inteligentes e automatizadas, onde a tecnologia desempenha um papel central na coleta e armazenamento de dados, permitindo que as empresas tenham controle e acesso às informações em tempo

real. Esses dados são obtidos por meio de sensores instalados em pontos estratégicos para o monitoramento.

Conforme Ortiz, Marroquin e Cifuentes (2020), a Indústria 4.0 tem nove pilares que compõem a base da transformação digital nas indústrias. Cada um desses pilares tem sua importância na formação de uma indústria inteligente. A seguir, estão listados:

- Internet das Coisas (IoT): é a incorporação de sensores e atuadores nas máquinas, conectando-as a uma rede para o controle dos processos industriais em tempo real;
- Big Data e Data Analytics: refere-se ao grande volume de dados produzidos por diferentes fontes, que precisam ser processados e armazenados. A partir desses dados, é possível realizar análises para criar estratégias de negócios e tomar decisões;
- Realidade Aumentada: é a combinação da realidade com elementos virtuais, funcionando como uma ferramenta para auxiliar o homem a obter mais informações sobre algo à sua volta, ou realizar tarefas;
- Segurança Cibernética: é responsável pela proteção de dados armazenados em servidores conectados à rede, bloqueando ameaças à integridade da rede, sejam físicas ou virtuais;
- Robôs Colaborativos: são robôs inteligentes que interagem com o homem e outras máquinas, por meio de sensores e IA, aprendem a realizar tarefas e tomar decisões em diferentes situações. A implantação destes robôs trará um aumento na eficiência da produção, enquanto os custos irão diminuir;
- Manufatura Aditiva: abrange as tecnologias que permitem a criação de peças complexas do zero a partir de um modelo virtual, adicionando material em camadas ao objeto, como no caso da impressão 3D;
- Computação em Nuvem: é o armazenamento de dados diretamente na internet, facilitando o acesso ao usuário e garantindo a segurança dos dados por meio de login. Essa opção é vantajosa para pequenas empresas, que evitam os altos custos de instalação e manutenção de um servidor físico;
- Inteligência Artificial (IA): é o uso de programas computacionais para tomada de decisões e execução de tarefas, essa área de estudo tem o objetivo de reproduzir o pensamento humano em robôs;

- Redes 5G: é a quinta geração para redes móveis e de banda larga, mais rápida e eficiente na transmissão de dados entre dispositivos.

O uso das tecnologias mencionadas possibilita a automação completa dos ambientes de trabalho. No entanto, sabe-se que os investimentos ainda são altos, o que se torna inviável, especialmente para pequenas agroindústrias.

2.3 Automação industrial

A automação industrial é uma tecnologia para transformar operações manuais em automáticas com o uso de máquinas, equipamentos, softwares e outras tecnologias. Seu objetivo é diminuir a dependência da mão de obra gerada na indústria, aumentando a produtividade e a padronização dos produtos fabricados.

De acordo com Roggia e Fuentes (2016), desde a pré-história, o homem vem desenvolvendo mecanismos e invenções com o intuito de reduzir o esforço físico e auxiliar na realização de atividades. A criação da roda para transporte de cargas e os moinhos movidos por vento ou força animal são exemplos das automações desse período.

Com o início da revolução industrial na Inglaterra, a sociedade tomou conhecimento do que seria a automação industrial e, ao longo dos anos, outros países também adotaram as tecnologias até então desenvolvidas. Os avanços tecnológicos obtidos nas fases da revolução impactaram o modelo de trabalho com produções de larga escala, e gradativamente o trabalho manual foi substituído por máquinas automatizadas.

A pirâmide da automação industrial (Figura 2.2) é uma representação visual da organização hierárquica dos cinco níveis de controle e trabalho. O primeiro nível, conhecido como “chão de fábrica”, é composto por máquinas e dispositivos como atuadores, sensores, transmissores e outros elementos que integram as operações em campo. No segundo nível, encontram-se equipamentos responsáveis pelo controle automatizado das atividades. Esses equipamentos devem possuir um nível de automação que pode ser alcançado por meio da implantação de CLPs.

O terceiro nível é responsável por supervisionar e otimizar o processo produtivo, utilizando bancos de dados e algoritmos de otimização. No quarto nível, é realizado o processo de controle fabril, a programação e o planejamento da produção. Já o quinto nível é dedicado à administração dos recursos da empresa, onde, por meio de softwares, ocorre a organização e o controle dos dados.



Figura 2.2 – Pirâmide da automação industrial (NEPIN, 2023)

2.4 Modbus TCP/IP

O protocolo Modbus TCP/IP é uma estrutura de mensagens utilizada para estabelecer uma comunicação cliente/servidor entre dispositivos conectados em diferentes tipos de barramentos ou redes (MODBUS.ORG, 2012). O cliente envia uma requisição para um servidor, que processa a solicitação e retorna uma resposta.

A sua estrutura é composta por quatro tipos específicos de dados:

- Coils (Bobinas): dado booleano usado para saídas digitais que podem ser controladas, o acesso é de leitura e escrita;
- Discrete Inputs (Entradas Discretas): dado booleano usado para entradas digitais, o acesso é apenas de leitura;
- Holding Registers (Registradores de Retenção): dado de 16 bits usado para parâmetros de configuração e saídas analógicas, o acesso é de leitura e escrita;
- Input Registers (Registradores de Entrada): dado de 16 bits usado para valores de sensores e medições, o acesso é apenas de leitura.

O Modbus possui duas categorias de funções: funções públicas são pré-definidas e universalmente aceitas em todos os dispositivos Modbus, e funções definidas pelo usuário são personalizadas por fabricantes individuais e podem não ser universalmente reconhecidas (WEVOLVE.COM, 2023).

Na Tabela 2.1 são apresentadas as funções públicas, a função 16 foi utilizada no envio de mensagens desse projeto devido à necessidade de escrever em múltiplos registradores:

Código (Hex)	Função	Objetivo Principal
01 (01)	Read Coils	Ler o estado (ON/OFF) de uma ou mais saídas digitais.
02 (02)	Read Discrete Inputs	Ler o estado de uma ou mais entradas digitais.
03 (03)	Read Holding Registers	Ler o valor de um ou mais registradores de retenção.
04 (04)	Read Input Registers	Ler o valor de um ou mais registradores de entrada.
05 (05)	Write Single Coil	Escrever o estado (ON ou OFF) em uma única saída digital.
06 (06)	Write Single Register	Escrever um valor numérico em um único registrador de retenção.
15 (0F)	Write Multiple Coils	Escrever o estado em um bloco de saídas digitais.
16 (10)	Write Multiple Registers	Escrever valores numéricos em um bloco de registradores de retenção.

Tabela 2.1 – Códigos de função do protocolo Modbus TCP/IP (Fonte: Adaptado de Modbus.org (2012) e ProSoft (2005))

Esse protocolo é bastante utilizado em automações industriais por sua simplicidade e fácil implementação. Destaca-se que esse protocolo possui código aberto, o que viabilizou a aceitação entre os fabricantes de componentes.

A sua aplicabilidade em CLPs e IHMs é ampla e muito utilizada. Este projeto tem o objetivo de conectar os equipamentos na rede local para realizar o monitoramento e controle dos processos. Assim, o Modbus TCP/IP é essencial, ao realizar a comunicação entre esses dispositivos através da Ethernet, permitindo a leitura e escrita de dados, desde que seja corretamente configurado e programado. Essa comunicação entre os equipamentos e o servidor é de grande importância, ao permitir o armazenamento dos dados de operação do equipamento e a geração de relatórios e estatísticas.

2.5 Trabalhos Relacionados

2.5.1 AUTOMAÇÃO DE UM PROCESSO ALTERNATIVO DA PASTEURIZAÇÃO DO LEITE UTILIZANDO AÇÃO DE CONTROLE ON-OFF

Conforme apresentado na seção 2.1, o processo de pasteurização do leite é obrigatório para a sua comercialização, ao eliminar microrganismos patogênicos prejudiciais à saúde do indivíduo.

Nesse contexto, o trabalho proposto em Barbosa et al. (2018) tem como objetivo automatizar um protótipo de pasteurização rápida de leite. Devido ao alto custo de equipamentos industriais fabricados em aço inox, foram utilizados materiais alternativos para reduzir custos, considerando que o público-alvo do estudo são pequenos produtores.

Para a construção do protótipo, foram utilizados os seguintes materiais: sensor de temperatura LM 35, microcontrolador Basic Step 1, display LCD 16x2, minicondutor auxiliar 110 VCA-16 A, placa StepLab, relé de 5 Vcc, transistor BC 548 ($\beta = 100$), resistores de carvão de $\frac{1}{4}$ W com precisão de 5% (330 Ω , 1 k Ω , 10 k Ω), fonte de alimentação 12 Vcc, cabo serial DB9, aquecedor resistivo 110 VCA x 1000 W, bomba injetora (bomba de água veicular do limpador de para-brisas) e um LED 3 mm, além de fios, solda, conectores, cola e recipientes diversos (BARBOSA et al., 2018).

Esses componentes foram dispostos conforme a Figura 2.3 apresentada a seguir:

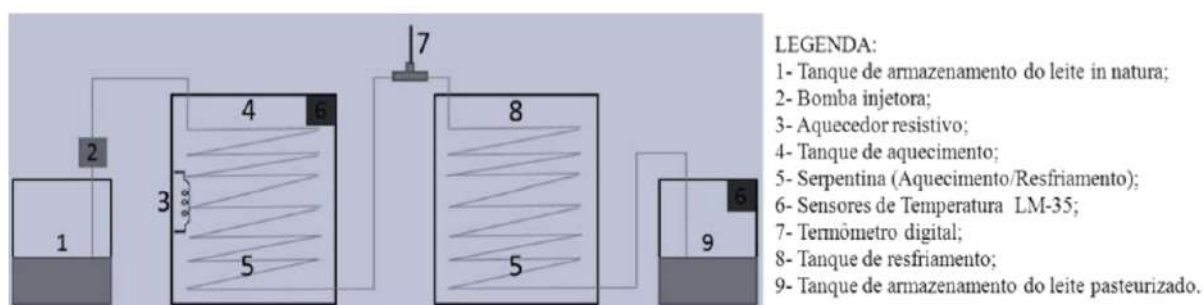


Figura 2.3 – Esquema do pasteurizador alternativo de leite com seus componentes (BARBOSA et al., 2018)

Testes prévios de vazão, comprimento da mangueira, temperatura da água e voltagem da bomba injetora foram realizados para determinar o tempo necessário para atingir a temperatura de pasteurização do leite. O processo de pasteurização automática foi realizado em batelada, onde o leite *in natura* era bombeado do tanque de repouso para o tanque de água quente até preencher completamente a serpen-

tina. Em seguida, a bomba era desligada, mantendo o leite em contato com a água quente por 135 segundos. Após isso, a bomba era novamente acionada, conduzindo o leite aquecido para a serpentina do tanque de resfriamento, onde permanecia por 300 segundos. Por fim, a bomba era acionada novamente para conduzir o leite já pasteurizado ao reservatório final (BARBOSA et al., 2018).

A Figura 2.4 apresenta a sequência operacional do protótipo para a pasteurização do leite:

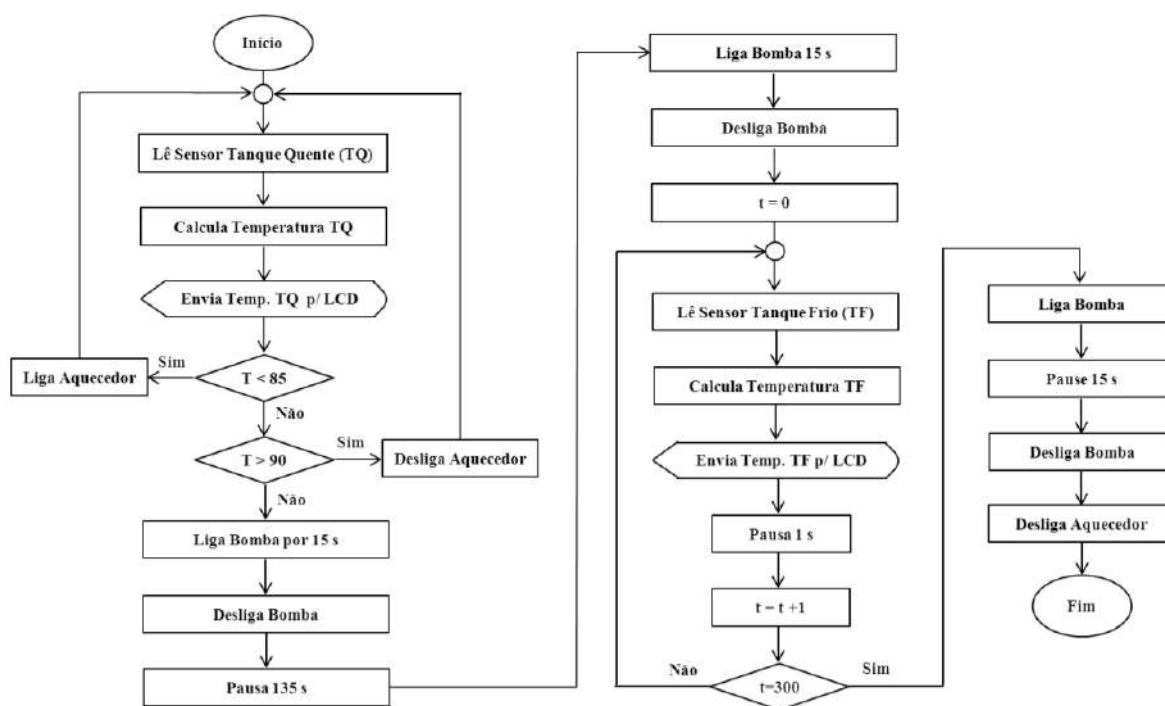


Figura 2.4 – Fluxograma do processo do protótipo (BARBOSA et al., 2018)

Este projeto demonstrou-se eficiente, pois o algoritmo conseguiu acionar a bomba injetora e o aquecedor resistivo, utilizando as leituras dos sensores de temperatura e a sequência programada no *Basic Step* como referência. Devido ao material utilizado, foi necessário um tempo maior para realizar a pasteurização. Enquanto o tempo de referência é de 15 a 20 segundos, o tempo obtido foi de 135 segundos para aquecimento e 300 segundos para resfriamento, indicando que o sistema desenvolvido ainda precisa de melhorias para obter resultados mais próximos do esperado.

2.5.2 PROPOSTA DE AUTOMAÇÃO DE UMA PLANTA DE PRODUÇÃO DE IOGURTE

O iogurte é um derivado do leite que pode ser produzido em uma agroindústria. Inicialmente, é realizada a pasteurização do leite e o desnatado; em seguida,

adiciona-se o soro e realiza-se a fermentação. Por fim, são adicionados a polpa e o aroma, e o produto é envasado.

Em Milagres (2022), foi proposta a automação de uma planta de produção de iogurte fermentado que abrange todos os processos em uma agroindústria de leite. A linha de produção começa com a **Recepção de Leite e Resfriamento**, onde o operador deve realizar a entrega do leite, conectando manualmente a mangueira na válvula de recepção do caminhão. Para reduzir erros operacionais, o projeto propõe a instalação de uma IHM nesse ambiente, assim o operador deve realizar confirmações de abertura manual da válvula do caminhão. O sistema então verifica se o engate da mangueira foi realizado corretamente.

Em seguida, o leite passa por um tanque desaerador (TQ-1001) para remover a espuma. Esse tanque é equipado com uma chave de nível (alto: 1000-LSH-01 e baixo: 1000-LSL-01) que identifica a presença de leite, permitindo o acionamento da bomba centrífuga (1000-BC-01), que transporta o leite pelas tubulações até os filtros (1000-FI-01 e 1000-FI-02). Foi desenvolvido um método em que um sensor de pressão identifica a saturação do filtro. Caso o filtro esteja saturado, o sistema alterna automaticamente a entrada do leite entre os filtros e emite um alerta para o operador, que então realiza a limpeza do filtro.

O leite é encaminhado para o trocador de calor (1000-TC-01) que realiza um controle PID da válvula moduladora de água gelada (1000-VA-07) para o leite atingir a temperatura desejada. Após a descarga do leite do caminhão, é necessário realizar o “flush” do leite com água para limpar as mangueiras. O turbidímetro (1000-AT-01) detecta a transição de leite para água na tubulação e fecha a válvula de entrada do tanque (2000-VA-01).

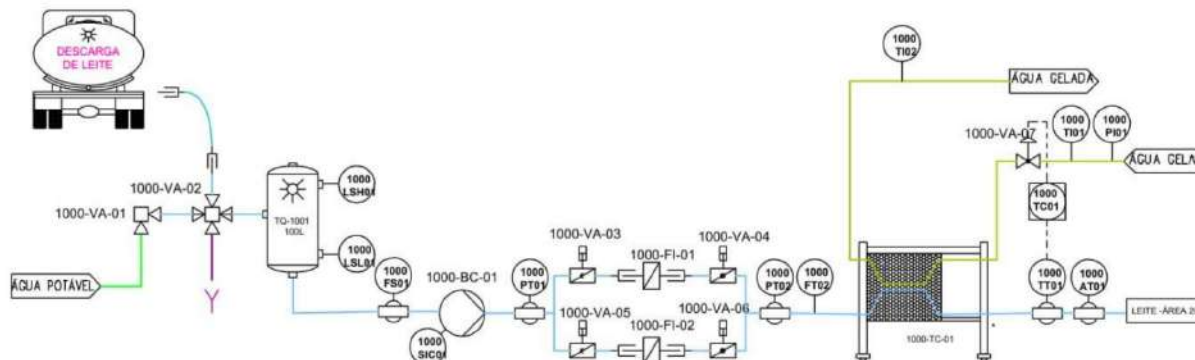


Figura 2.5 – Fluxograma detalhado da área 1000 - Recepção de Leite e Resfriamento (MILAGRES, 2022)

A próxima etapa na linha de produção é a **Estocagem e Pasteurização do Leite**, onde o leite é armazenado no tanque TQ-2001, esse por sua vez também possui controle de nível (alto: 2000-LSH-01 e baixo: 2000-LSL-01), transmissor de nível (2000-LT-01), transmissor de temperatura (2000-TT-01) para auxiliar no controle de qualidade do leite, e um agitador (2000-AG-01) para manter o leite homogêneo. Seguindo o fluxo, o leite é enviado ao pasteurizador, onde é pasteurizado e desnatado, resultando em leite pasteurizado desnatado e creme de leite.

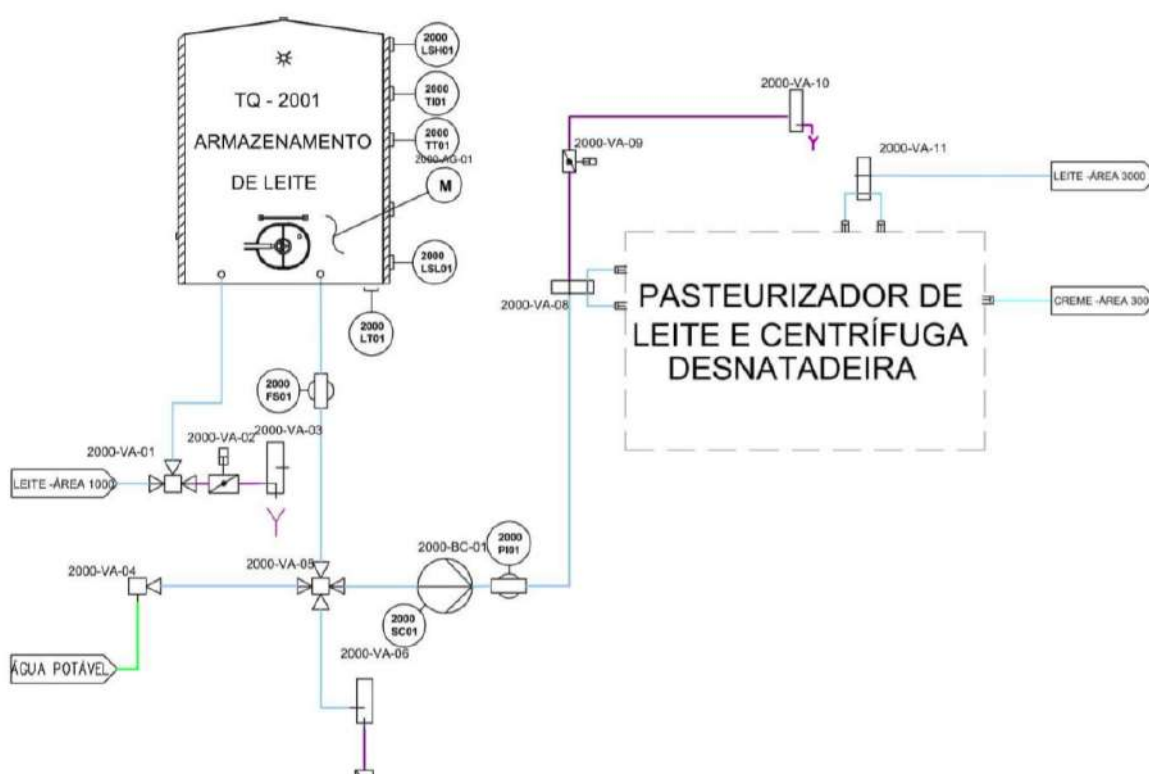


Figura 2.6 – Fluxograma detalhado da área 2000 - Estocagem e Pasteurização de Leite (MILAGRES, 2022)

No estudo, também é detalhada a automação das etapas de **Preparação de Iogurte, Fermentação, Base Branca e Envase**.

Para o desenvolvimento do projeto, foi utilizado um CLP da Rockwell Automation para comunicar-se com as válvulas da planta descrita acima, utilizando a rede ASI, que opera no formato mestre-escravo. O CLP também se comunica com os inversores de frequência via rede Ethernet.

Optou-se por uma topologia em anel, na qual a comunicação é unidirecional entre os dispositivos conectados à rede, com os dados viajando de nó em nó até o destino. Para reduzir distorções e atenuações nos sinais transmitidos, cada estação

conta com um repetidor. A comunicação com os instrumentos e os motores de partida direta é realizada via cartões de saída/entrada de sinais analógicos e digitais.

O esquema abaixo exemplifica como foi montada a automação para a planta de produção de iogurte:

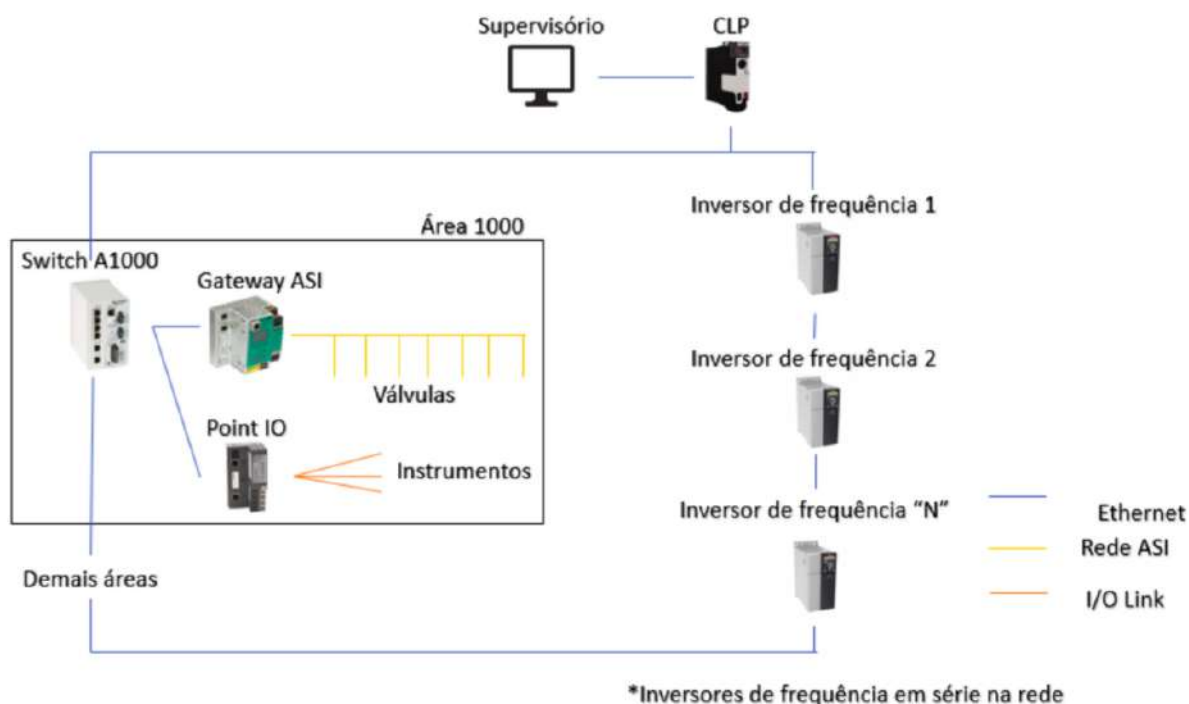


Figura 2.7 – Arquitetura da rede de automação (MILAGRES, 2022)

O iogurte é um produto extremamente sensível a contaminações físico-químicas, sendo necessário buscar soluções sanitárias de controle e monitoramento da planta. Com isso, a automação conforme proposta nesse trabalho utiliza instrumentação Endress+Hauser e controladores Rockwell Automation, fornecedores de referência no mercado, que apresentam soluções de alto nível e confiabilidade.

Este trabalho focou em automatizar uma planta para reduzir falhas em tarefas manuais realizadas pelos operadores. A planta automatizada contribui para evitar riscos à qualidade do produto, e identificar e notificar o operador que está realizando as tarefas corretamente.

Observações sobre os trabalhos relacionados

O primeiro artigo aborda a automação de um protótipo de pasteurização de leite, cujo objetivo era atender à demanda de pequenos produtores. Para isso, foram utilizados materiais alternativos ao inox devido ao custo elevado. Essa automação contempla o processo completo realizado na pasteurização, onde foi detalhada cada

etapa estudada no artigo. A partir dele, é possível obter a lista de componentes utilizados, o esquema elétrico e o fluxograma que define a lógica de operação.

As informações apresentadas, juntamente com os resultados obtidos, serão de grande importância para realizar a automação do pasteurizador lento compacto para bezerras. Elas servem como base para a criação do projeto de automação desse estudo, oferecendo uma ideia inicial que pode orientar o desenvolvimento em um caminho semelhante. Contudo, o artigo propõe a automação de um protótipo e esse estudo irá trabalhar com um equipamento industrial já disponível no mercado.

O segundo artigo apresenta uma proposta de automação de uma planta de produção de iogurte fermentado, onde são detalhados os componentes e equipamentos utilizados em cada etapa da produção do iogurte. Como o leite pasteurizado é um produto extremamente sensível a contaminações, o controle de acionamentos automáticos é essencial para garantir a qualidade no processo de manipulação e prevenir falhas.

Além disso, o artigo descreve a arquitetura da rede de automação, especificando o uso de um CLP e inversores no sistema, onde cada área/etapa é composta por: um switch, um gateway ASI, um Point IO, válvulas e instrumentos. Nesse estudo, serão realizados comandos de forma automática por meio de um CLP, e o artigo poderá contribuir para a elaboração dessa solução.

3. AUTOMAÇÃO DO PASTEURIZADOR

Neste capítulo será apresentada a automação do pasteurizador. Inicialmente, será descrito o equipamento da forma como ele funciona sem automação. Após a descrição do equipamento, apresentaremos os requisitos funcionais e não funcionais do sistema de automação. A seguir, iremos detalhar a arquitetura de hardware proposta para a automação e, por último, a arquitetura de software.

3.1 Pasteurizador Lento Elétrico Compacto de Leite para Bezerras

O pasteurizador tem como objetivo aproveitar o leite de descarte, reduzindo significativamente a proliferação de bactérias e agentes patógenos causadores de doenças. Após a pasteurização, esse leite será utilizado no trato dos bezerros.

Construído em aço inox, o equipamento possui sistema automático de pasteurização e camisa tripla, composta por três camadas de inox. Entre essas camadas, é utilizada água ou espuma expansiva, sendo esta responsável pelo isolamento térmico do equipamento.

O equipamento conta com agitador elétrico para mistura do leite e controladores digitais de temperatura. O pasteurizador tem registro de 2 polegadas para escoamento de líquidos e tampa removível para limpeza. O aquecimento é realizado por meio de resistência elétrica, enquanto o resfriamento ocorre por meio de uma unidade de frio com fundo expandido. Um quadro elétrico contendo botoeiras, contoras e temporizadores é acoplado ao equipamento e é responsável por coordenar os processos no pasteurizador. A Figura 3.1 mostra a visão frontal do pasteurizador.

3.1.1 Processos Realizados pelo Equipamento

O primeiro processo é a pasteurização, que realiza o aquecimento do leite entre 62 °C e 65 °C, em seguida, permanece em repouso por 30 minutos. A pasteurização é um tratamento térmico que tem o objetivo de eliminar microrganismos patogênicos.

O segundo processo é a refrigeração, que realiza o resfriamento do leite a 3 °C. Após atingir essa temperatura, é iniciado o ciclo de descanso, 30 minutos de descanso com 5 minutos de agitação. A refrigeração tem como objetivo inibir ou retardar o crescimento de microrganismos, conservando o leite por mais tempo. A agitação tem o intuito de homogeneizar a temperatura do leite.



Figura 3.1 – Pasteurizador lento - vista frontal (Fonte: Autor)

O terceiro processo é o aquecimento para o trato, que realiza o aquecimento do leite a aproximadamente 37 °C, de acordo com o que o operador programou. Esse processo proporciona uma temperatura semelhante à corporal da vaca, tornando a ingestão mais confortável para os bezerros.

3.1.2 Operação do Equipamento

Primeiramente, o operador regula o programador de horário analógico levantando os pinos referentes a 1 hora e 30 minutos para iniciar o processo de pasteurização. Em seguida, seleciona a chave seletora na posição automática, que dá início ao processo de pasteurização e refrigeração do leite automaticamente.

Para a realização do processo do trato, 30 minutos antes, o operador seleciona a chave seletora na posição manual e define a temperatura de trato no termostato. Inicia-se o aquecimento até a temperatura definida para o trato, na sequência é realizada a retirada do leite do tanque para tratar os bezerros.

Nessa versão do equipamento, não é possível ao operador definir um horário mais aproximado, deve ser realizado um cálculo manual e definido o programador de horário analógico com esse intervalo de tempo calculado. Caso o operador esqueça o leite no pasteurizador e o programador de horário atingir o ponto onde a pasteurização inicia, o equipamento realizará a pasteurização novamente, o que não deve ocorrer.

Também deve ser destacado que o processo de trato é realizado manualmente, assim o operador deve reservar um tempo para dar o comando para esse processo. Após atingir a temperatura de trato, o processo é finalizado e o leite estará pronto para tratar as bezerras.

Esse equipamento não se enquadra nos conceitos da Indústria 4.0, dessa forma, não é possível realizar o monitoramento e controle da máquina remotamente. Além disso, o equipamento não registra os dados do andamento do processo.

3.2 Requisitos da Automação

Considerando as características do processo, as falhas identificadas no funcionamento e a necessidade de atualização para atender às demandas do mercado atual, a seguir são apresentados os requisitos funcionais e não funcionais da automação do pasteurizador.

3.2.1 Requisitos Funcionais

- RF01 - Controle de temperatura: o sistema deve ser capaz de aquecer ou resfriar o leite até uma temperatura alvo (ex.: 62 °C) e mantê-la estável por um período de tempo predefinido (ex.: 30 minutos);
- RF02 - Monitoramento em tempo real: o sistema deve ler continuamente a temperatura e exibir ao operador o processo em execução, a etapa atual e a temperatura medida em tempo real;
- RF03 - Acionamento de atuadores: o sistema deve comandar o acionamento e desligamento dos atuadores (resistências de aquecimento, unidade de frio e agitador) de acordo com a lógica de controle estabelecida;
- RF04 - Alertas: o sistema deve identificar e alertar o operador sobre o status do equipamento durante o processo, como a finalização da etapa de pasteurização ou trato, e também quando o horário selecionado para o início do trato não é suficiente para a operação completa do equipamento;
- RF05 - Definição de ciclo: o sistema deve permitir que o operador inicie ou encerre o ciclo de operação diretamente pela interface da IHM;
- RF06 - Captação e envio de dados: o sistema deve realizar o envio dos dados de operação do equipamento para o servidor em nuvem;

- RF07 - Comunicação com Sistema de Supervisão/Web: a IHM deve se conectar com o Sistema Web para poder realizar o controle do equipamento remotamente;

3.2.2 Requisitos Não Funcionais

- RNF01 - Precisão: o sistema deve garantir leituras de temperatura com margem de erro inferior a $\pm 0,5$ °C, assegurando o controle adequado do processo;
- RNF02 - Confiabilidade: o sistema deve operar continuamente por longos períodos (24/7) sem falhas críticas ou necessidade de reinicialização manual;
- RNF03 - Manutenibilidade: o código-fonte do CLP e da IHM deve ser bem documentado e modular para facilitar futuras atualizações ou correções;
- RNF04 - Usabilidade: a interface da IHM deve ser intuitiva e de fácil compreensão, permitindo que um operador com treinamento básico consiga acompanhar e controlar o processo com segurança;
- RNF05 - Robustez: o sistema deve ser capaz de suportar variações de energia, pequenas falhas de comunicação ou reinicializações sem perda de dados importantes;
- RNF06 - Segurança operacional: o sistema deve impedir o acionamento simultâneo de atuadores conflitantes (como aquecimento e refrigeração), garantindo a integridade dos componentes e a segurança do operador.

3.3 Arquitetura de Hardware

A arquitetura de hardware do sistema proposto é composta por dispositivos responsáveis pelo controle, monitoramento e comunicação dos processos automatizados.

Considerando a arquitetura e os requisitos, segue o conjunto de dispositivos que compõem a arquitetura do sistema:

- CLP: Rockwell Automation Micro820 2080-LC20-20QBB, porta Ethernet incorporada e porta serial RS-232/485 não isolada, (12) entradas digitais de 24 Vcc, (7) saídas de fonte de 24 Vcc, (1) saída analógica, (4) entradas analógicas configuráveis com saída de referência de tensão do termistor, RTC incorporado, alimentação de 24 Vcc.

- IHM: Delta Electronics DOP-107EV, 7 polegadas, resolução 800×480, porta Ethernet incorporada, porta USB slave, 3 portas COM, alimentação de 24 Vcc.
- Sensor de temperatura: Ageon Sensor NTC em Inox, com faixa de operação de -50 °C a 100 °C.
- Transmissor de temperatura: Novus TxRail-USB, configuração através do conector USB micro-B, entrada configurável para termopar J, K, T, E, N, R, S, B, Pt100, Pt1000, NTC, 0-50 mV e 0-100 mV, saída 4-20 mA ou 0-10 V.
- Resistências: Technik Resistência 3 elementos, flange de 2", potência 8000 W, tensão 220 V.
- Unidade de frio: Elgin Unidade Condensadora UCM2150, potência máxima de 1,5 HP, Monofásico, 220 V, gás R22.
- Agitador: Varivelox Motorreductor VXRSA 15, potência máxima de 0,05 hp, tensão de 220/254 V, relação de redução de 1:37, 1100 RPM motor, 30 RPM saída.
- Switch: Stratix 2000 5T Port Unmanaged 1783-US5T, 5 portas Ethernet incorporadas, plug and play, taxa de dados 10/100 Mbps, alimentação de 24 Vcc.

A escolha desses dispositivos foi influenciada pela experiência da empresa com o uso dos mesmos, e também, os adicionados na automação foram selecionados de acordo com a necessidade levantada a partir do fluxo de processos e do Sistema Web para monitoramento e controle. Futuramente, esses dispositivos serão reavaliados, a fim de encontrar opções melhores e de menor custo.

A Figura 3.2 apresenta a arquitetura do sistema de automação com os dispositivos listados acima.

O controle dos processos do equipamento é realizado pelo Controlador Lógico Programável (CLP), o qual gerencia a leitura do sensor de temperatura, as variáveis do ambiente que compõem a lógica, e controla os atuadores conforme a programação implementada, seguindo o fluxograma apresentado na Figura 3.4.

O sensor de temperatura utilizado é do tipo NTC, responsável por monitorar as variações térmicas do leite durante as etapas de pasteurização, refrigeração e trato. O CLP utiliza o valor da leitura desse sensor para comparar com os valores programados para cada etapa, evitando o uso de múltiplos sensores para a realização desse controle, o que reduz o custo do sistema e simplifica a manutenção sem comprometer a confiabilidade das medições. Para este sensor, foi utilizado um transmissor de temperatura TxRail-USB, configurado para converter o sinal do sensor em tensão de

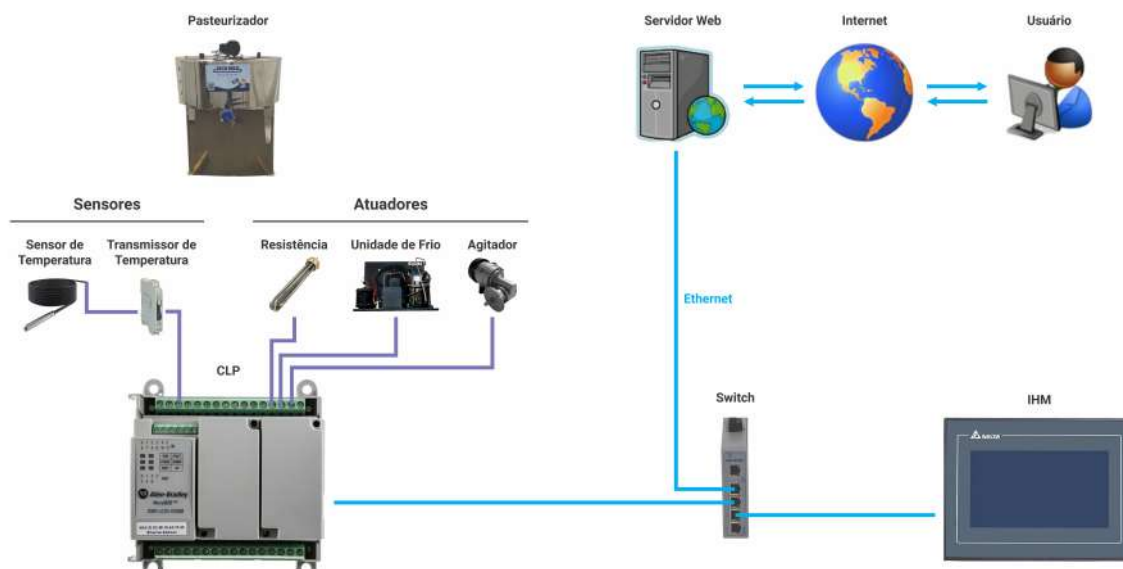


Figura 3.2 – Arquitetura do sistema de automação (Fonte: Autor)

0–10 V, a qual é lida pela entrada analógica do CLP. Nessa etapa, os requisitos RF01 e RNF01 são cumpridos.

Os atuadores são responsáveis pelo aquecimento do leite (resistências), pelo resfriamento do leite (unidade de frio) e pela agitação do leite (agitador). Todos são controlados pelo CLP, que os aciona e desaciona conforme a lógica programada. Os atuadores, em conjunto com a lógica do CLP, cumprem com o requisito RF03.

Os comandos enviados pelo operador ao CLP são realizados através da Interface Homem-Máquina (IHM), que apresenta o monitoramento em tempo real do equipamento: temperatura atual, temperaturas programadas, processo selecionado, etapa do processo e atuadores. A IHM permite que o operador faça alterações nos parâmetros de operação do equipamento, além da seleção ou parada de um processo. Essa funcionalidade foi solicitada nos requisitos RF02 e RF05.

Para a comunicação entre o CLP e a IHM, foi instalado um switch no painel elétrico, que utiliza o protocolo Ethernet para realizar a troca de dados entre os dois dispositivos. Esse mesmo switch também é utilizado para permitir a comunicação do equipamento com o Sistema Web desenvolvido. Para o cumprimento dos requisitos RF06 e RF07, é necessário que o switch esteja conectado à rede local de internet.

Na Figura 3.3 é apresentada a localização dos dispositivos descritos abaixo:

- Sensor de temperatura;
- Atuadores: resistências, unidade de frio e agitador;
- Painel elétrico: CLP, IHM, switch, TxRail-USB e outros componentes elétricos;

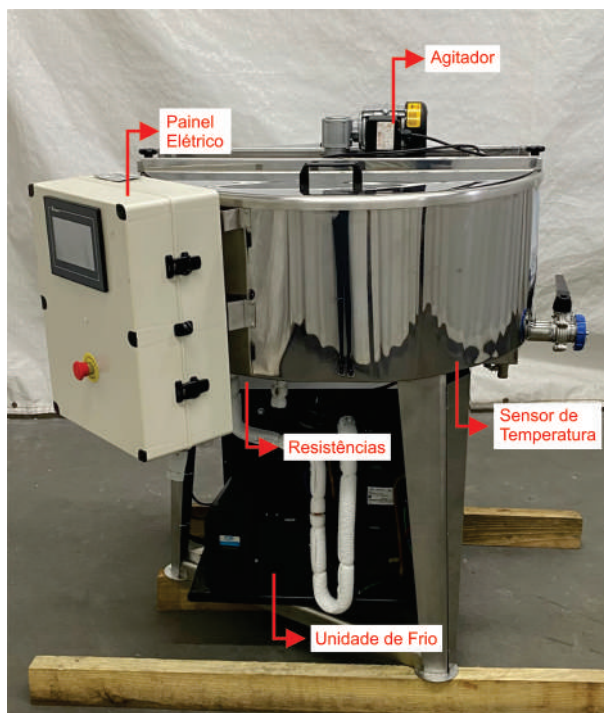


Figura 3.3 – Localização dos dispositivos no pasteurizador lento (Fonte: Autor)

3.4 Arquitetura de Software

A arquitetura de software referente à automação é baseada na programação do CLP e da IHM. Cada um desses dispositivos possui o seu próprio software para a programação, e para iniciar esse processo, foi necessário planejar os processos que o equipamento deveria realizar. Na Figura 3.4 está apresentado o fluxograma que explica a lógica de programação dos processos do equipamento realizado no CLP:

No fluxograma, são definidos cinco processos:

1. Pasteurização → Refrigeração → Trato;
2. Refrigeração → Pasteurização → Trato;
3. Pasteurização;
4. Refrigeração;
5. Trato.

No **fluxo do processo 1** é apresentado o processo de **Pasteurização** → **Refrigeração** → **Trato**: inicia após a definição do horário para o início do trato, em seguida, aquece o leite até atingir 62 °C. Quando a temperatura é atingida, o sistema passa a repousar o leite por 30 minutos. Após esse repouso, é verificada a lógica de

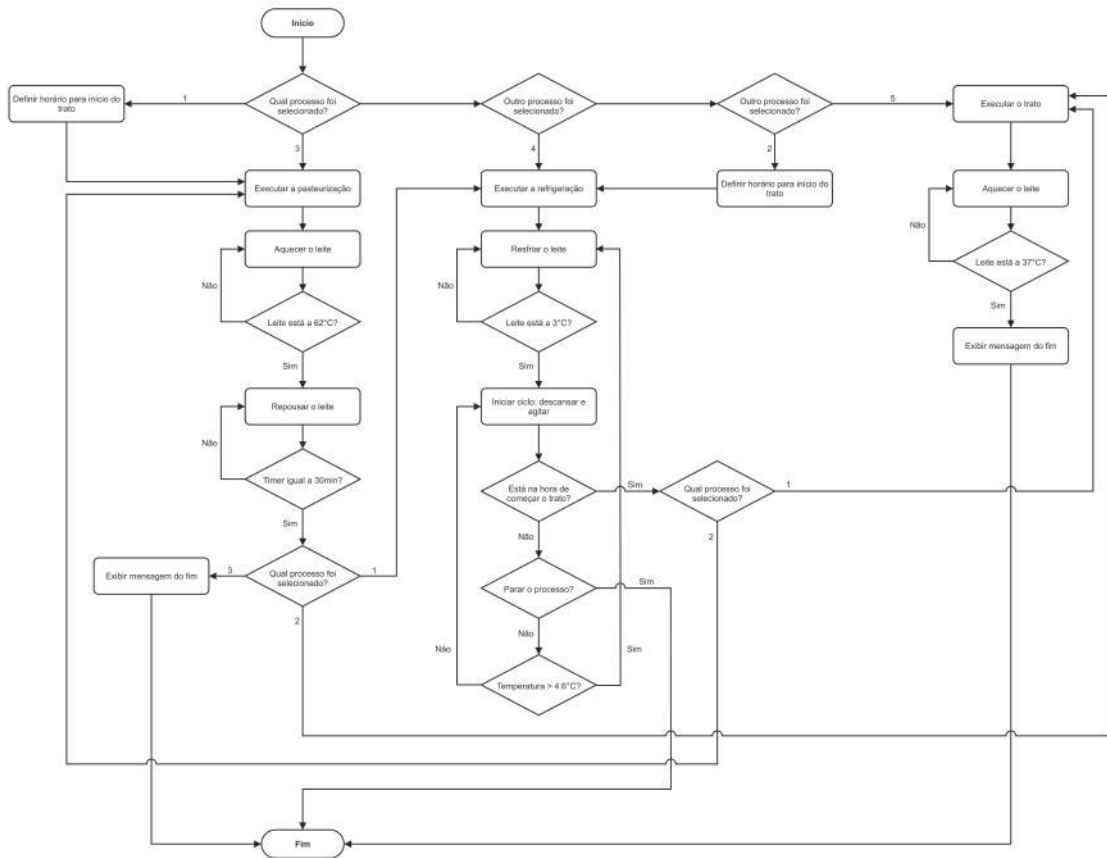


Figura 3.4 – Fluxograma da automação (Fonte: Autor)

fluxo e passado para a refrigeração, resfriando o leite até atingir 3 °C. Quando a temperatura é atingida, inicia-se um ciclo contínuo de descanso (20 minutos) e agitação (5 minutos). Durante o ciclo, o sistema verifica:

- Se é hora de iniciar o trato: o fluxo passa para o trato (essa hora é calculada a partir da hora definida pelo operador, é necessário iniciar com 45 minutos de antecedência para que dê tempo de realizar o processo);
- Se foi acionado o "Parar o processo": então é finalizado o processo completo;
- Se a temperatura ultrapassa 4,6 °C: então o sistema volta a resfriar o leite até 3 °C.

Assim que o fluxo é passado para o trato, é feito o aquecimento do leite até atingir 37 °C. Quando a temperatura é atingida, o sistema exibe uma mensagem e finaliza o processo.

No **fluxo do processo 2** é apresentado o processo de **Refrigeração** → **Pasteurização** → **Trato**: inicia após a definição do horário para o início do trato, em seguida, resfria o leite até atingir 3 °C. Quando a temperatura é atingida, inicia-se um ciclo contínuo de descanso (20 minutos) e agitação (5 minutos). Durante o ciclo, o sistema verifica:

- Se é hora de iniciar o trato: o fluxo passa para a pasteurização (essa hora é calculada a partir da hora definida pelo operador, é necessário iniciar com 150 minutos de antecedência para que dê tempo de realizar o processo);
- Se foi acionado o "Parar o processo": então é finalizado o processo completo;
- Se a temperatura ultrapassa 4,6 °C: então o sistema volta a resfriar o leite até 3 °C.

Assim que o fluxo é passado para a pasteurização, inicia-se o aquecimento do leite até atingir 62 °C. Quando a temperatura é atingida, o sistema passa a repousar o leite por 30 minutos. Após esse repouso, o fluxo é passado para o trato, resfriando o leite até atingir 37 °C. Quando a temperatura é atingida, o sistema exibe uma mensagem e finaliza o processo.

No **fluxo do processo 3** é apresentado o processo de **Pasteurização**: inicia aquecendo o leite até atingir 62 °C. Quando a temperatura é atingida, o sistema passa a repousar o leite por 30 minutos. Após esse repouso, é exibida uma mensagem e o processo é finalizado.

Na Figura 3.5 é ilustrada uma parte do código em ladder realizado no CLP da função de pasteurização, nesse trecho é dado o início à etapa de pasteurização de acordo com o processo selecionado, representado como modo na figura. Em seguida, há um módulo para verificar se a temperatura é maior ou igual à temperatura programada, isso ocorre enquanto a variável `inicio_past` está acionada, quando é atingida a temperatura esperada, a etapa é finalizada e a variável `final_past` é acionada.



Figura 3.5 – Função da pasteurização (Fonte: Autor)

No **fluxo do processo 4** é apresentado o processo de **Refrigeração**: inicia resfriando o leite até atingir 3 °C. Quando a temperatura é atingida, inicia-se um ciclo contínuo de descanso (20 minutos) e agitação (5 minutos). Durante o ciclo, o sistema verifica:

- Se foi acionado o "Parar o processo": então é finalizado o processo completo;
- Se a temperatura ultrapassa 4,6 °C: então o sistema volta a resfriar o leite até 3 °C.

Na Figura 3.6 é ilustrada uma parte do código em ladder realizado no CLP da função de refrigeração, nesse trecho é dado o início à etapa de refrigeração de acordo com o processo selecionado, representado como modo na figura. Em seguida, há um módulo para verificar se a temperatura é menor ou igual à temperatura programada, isso ocorre enquanto a variável `inicio_refrig` está acionada, quando é atingida a temperatura esperada, a etapa é finalizada e a variável `final_refrig` é acionada.



Figura 3.6 – Função da refrigeração (Fonte: Autor)

No **fluxo do processo 5** é apresentado o processo de **Trato**: inicia verificando a temperatura atual, se for superior a 37 °C irá resfriar e se for inferior irá aquecer o leite até atingir 37 °C. Quando a temperatura é atingida, o sistema exibe uma mensagem e finaliza o processo.

Na Figura 3.7 é ilustrada uma parte do código em ladder realizado no CLP da função de trato, nesse trecho é dado o início à etapa de trato de acordo com o processo selecionado, representado como modo na figura. Em seguida, tem duas linhas de código para realizar a análise da temperatura. Na primeira linha, há um módulo para verificar se a temperatura é menor ou igual à temperatura programada, caso seja verdadeiro, é acionada a variável `trato_aquec`, que irá aquecer o leite.

Na segunda linha, há um módulo para verificar se a temperatura é maior que a temperatura programada, caso seja verdadeiro, é acionada a variável `trato_refrig`, que irá resfriar o leite. Isso ocorre enquanto a variável `inicio_trato` está acionada.



Figura 3.7 – Função do trato parte 1 (Fonte: Autor)

Na Figura 3.8 é ilustrada outra parte do código em ladder realizado no CLP da função de trato, nesse trecho realiza o aquecimento ou o resfriamento do leite, de acordo com a variável acionada no trecho anterior (trato_aquec ou trato_refrig). Quando é atingida a temperatura esperada, a etapa é finalizada e a variável final_trato é acionada.



Figura 3.8 – Função do trato parte 2 (Fonte: Autor)

3.4.1 Programação do CLP

Para a realização da programação do CLP, foi necessário instalar o software próprio para o modelo da marca escolhido: Micro820 2080-LC20-20QBB da Rockwell Automation. O software Connected Components Workbench (CCW) é o responsável pela configuração e programação do CLP, nele foi criada a lógica descrita no fluxograma da Figura 3.4.

Para esse desenvolvimento, foi selecionada a linguagem de programação Ladder. Entretanto, conforme foi crescendo o projeto, tanto em tamanho quanto em complexidade, foi constatado que ficou difícil para realizar uma futura manutenção. Assim, após a finalização desse projeto, será realizada a programação em Texto Es-

truturado para comparar a organização dos códigos, focando em uma estrutura mais simples e bem organizada.

Importante ressaltar que o CLP e a IHM se comunicam por meio de um cabo Ethernet, assim foi necessário configurar um IP fixo para cada um. Essa configuração é feita através do software CCW para o CLP e DOPSoft para a IHM.

A Figura 3.9 é uma captura do software CCW, representando como são organizadas as ferramentas disponíveis para utilizar na programação do CLP. O programa aberto está em ladder, onde é possível identificar os módulos lógicos que compõem a etapa de pasteurização. Nesse programa ainda estão as outras funções necessárias para a operação completa do equipamento.

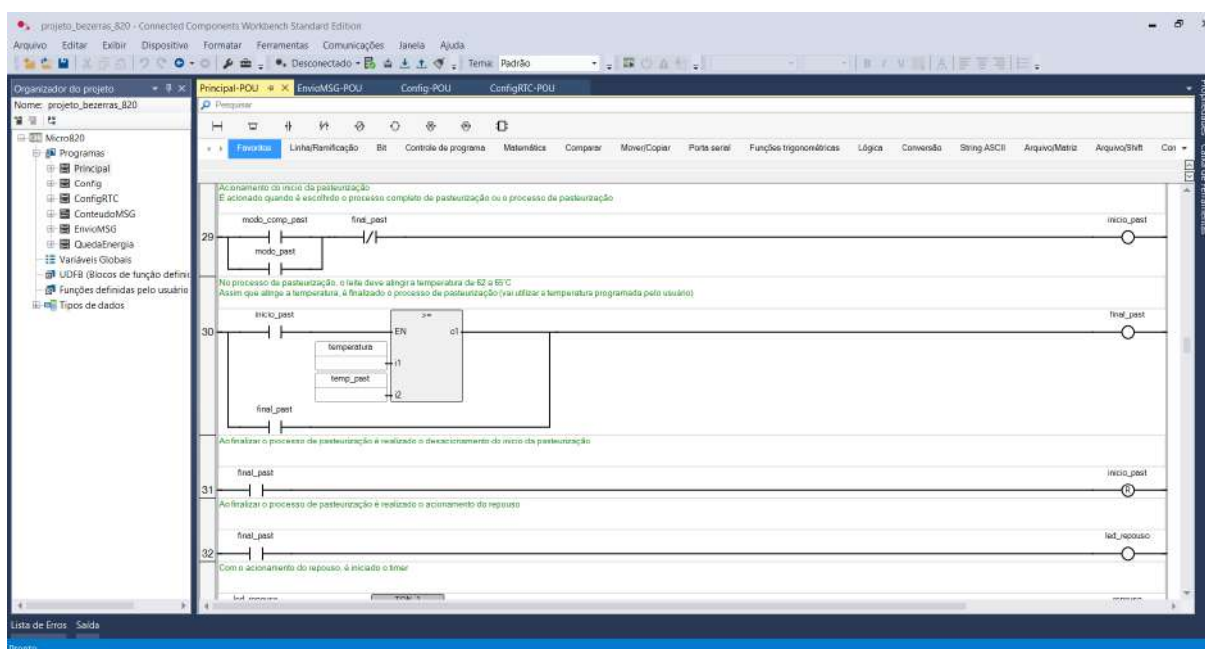


Figura 3.9 – CCW (Fonte: Autor)

Os requisitos RF01 e RNF01 exigem o uso de um sensor de temperatura de qualidade e devidamente calibrado. Essa calibração será feita por um laboratório de calibração certificado pelo Inmetro, que irá garantir que o sensor do equipamento esteja funcionando corretamente e enviando valores precisos de leitura para o CLP.

O requisito RF03 foi inserido ao longo da lógica de programação, onde esse acionamento dos atuadores é obrigatório para que os processos sejam realizados. Por exemplo, na pasteurização, é imprescindível o aquecimento e agitação do leite, assim as resistências e o agitador devem ser acionados enquanto o leite não atingir a temperatura programada.

Para o cumprimento do requisito RF06, foi utilizado o protocolo MODBUS para realizar o envio de mensagens com os dados do equipamento para o Sistema

Web, que possui um servidor MODBUS desenvolvido em Python, o qual recebe a mensagem, reorganiza os dados e cadastra no banco de dados do Sistema Web. A programação do MODBUS será detalhada melhor na seção 3.4.3.

Os requisitos não funcionais foram levados em conta durante a programação do fluxo dos processos do equipamento, considerando as diversas possibilidades que deveriam ser atendidas ou evitadas. Essa comprovação foi obtida nos testes contínuos realizados no equipamento com a automação feita e as modificações no projeto concluídas.

3.4.2 Programação da IHM

Para a realização da programação da IHM, foi necessário instalar o software próprio para o modelo da marca escolhido: DOP-107EV da Delta. O software DOPSoft é o responsável pela configuração e programação da IHM, nele foram criadas as telas para a operação do equipamento.

Essas telas não foram criadas 100% no DOPSoft, o layout foi criado no Figma, com o objetivo de criar um modelo moderno, minimalista e de fácil usabilidade, conforme o requisito RNF04. A lógica, os valores e os botões foram configurados no DOPSoft. Para trazer essas informações, foi necessário colocar todas as variáveis como globais no CLP e mapeá-las também na IHM.

A Figura 3.10 é uma captura do software DOPSoft, representando como são organizadas as ferramentas disponíveis para utilizar na programação da IHM. Foram criadas diversas telas para esse equipamento, a tela em aberto é a inicial, onde é possível identificar as temperaturas, status dos atuadores, botões de processos, configurações e manual de operação.

O monitoramento em tempo real descrito no RF02 precisa que o CLP mande as informações do programa para a IHM, essa comunicação é feita via Ethernet. As telas foram desenvolvidas de acordo com a necessidade dessas informações, então na tela de andamento de um processo, o operador pode acompanhar o status do equipamento.

No requisito RF04, foram desenvolvidas três telas na IHM para alertar o operador quando ocorre a finalização da etapa de pasteurização ou trato, e também quando o horário selecionado para o início do trato não é suficiente para a operação completa do equipamento.

A Figura 3.11 é a tela inicial da IHM, a tela é dividida em duas partes, na parte esquerda estão: a temperatura atual, valor captado do sensor; temperatura para pasteurização, programada pelo operador; temperatura para refrigeração, programada

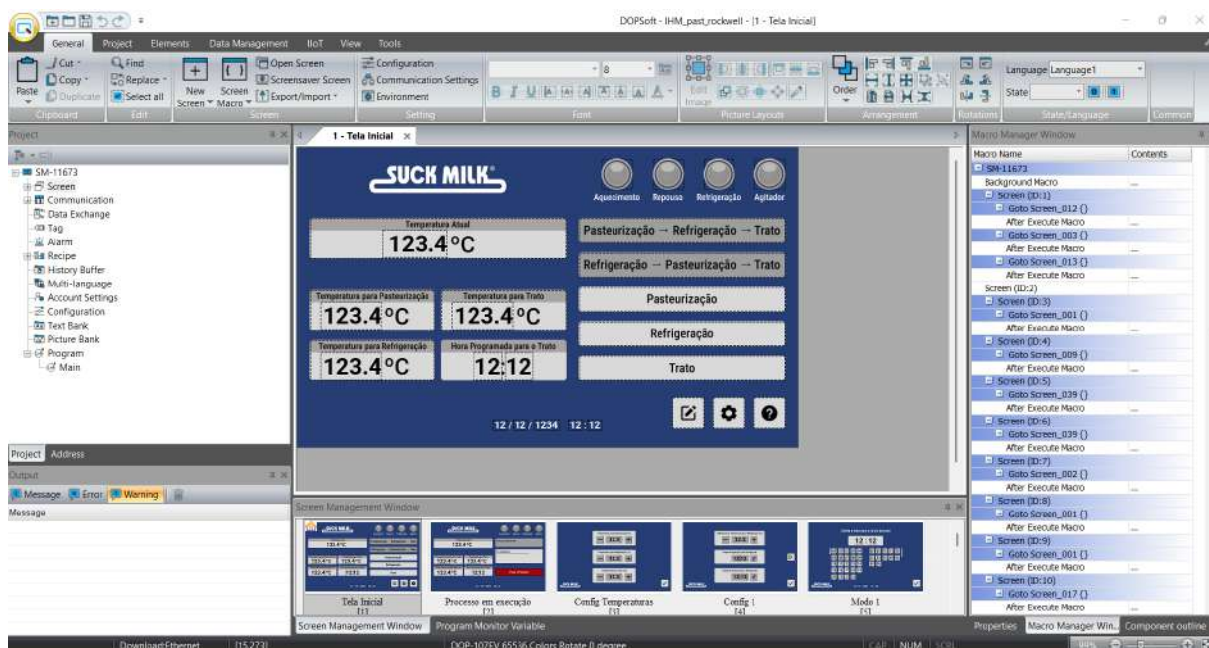


Figura 3.10 – DOPSoft (Fonte: Autor)

pelo operador; temperatura para trato, programada pelo operador; e hora programada, escolhida pelo operador após a seleção do processo 1 ou 2.

Na parte direita da figura, há os status dos atuadores: o aquecimento que é referente às resistências, o repouso que ocorre após a pasteurização, a refrigeração referente a unidade de frio e o agitador; e logo embaixo tem 5 botões, um para cada processo que o equipamento pode executar. Conforme o requisito RF05, esses botões foram criados para o operador poder selecionar o ciclo de operação desejado.

Após a seleção de um dos 5 processos, ocorre a troca da tela inicial pela tela de andamento. A Figura 3.12 é a tela de andamento da IHM, na parte esquerda estão os mesmos dados da tela inicial, e na parte direita tem os status dos atuadores, e logo abaixo está apresentado o processo selecionado e a etapa em andamento. Esses dados cumprem com o requisito RF02, realizando o monitoramento em tempo real do equipamento.

E, por fim, há o botão para parar o processo em andamento, essa opção permite que o operador encerre o ciclo de operação, assim como definido no requisito RF05.

No capítulo seguinte, serão apresentados os requisitos funcionais e não funcionais, e a arquitetura desenvolvida para o Sistema Web, responsável pelo monitoramento e controle do equipamento escolhido para esse estudo, o Pasteurizador Lento Elétrico Compacto de Leite para Bezerras. Nesse capítulo, o detalhamento é de cu-

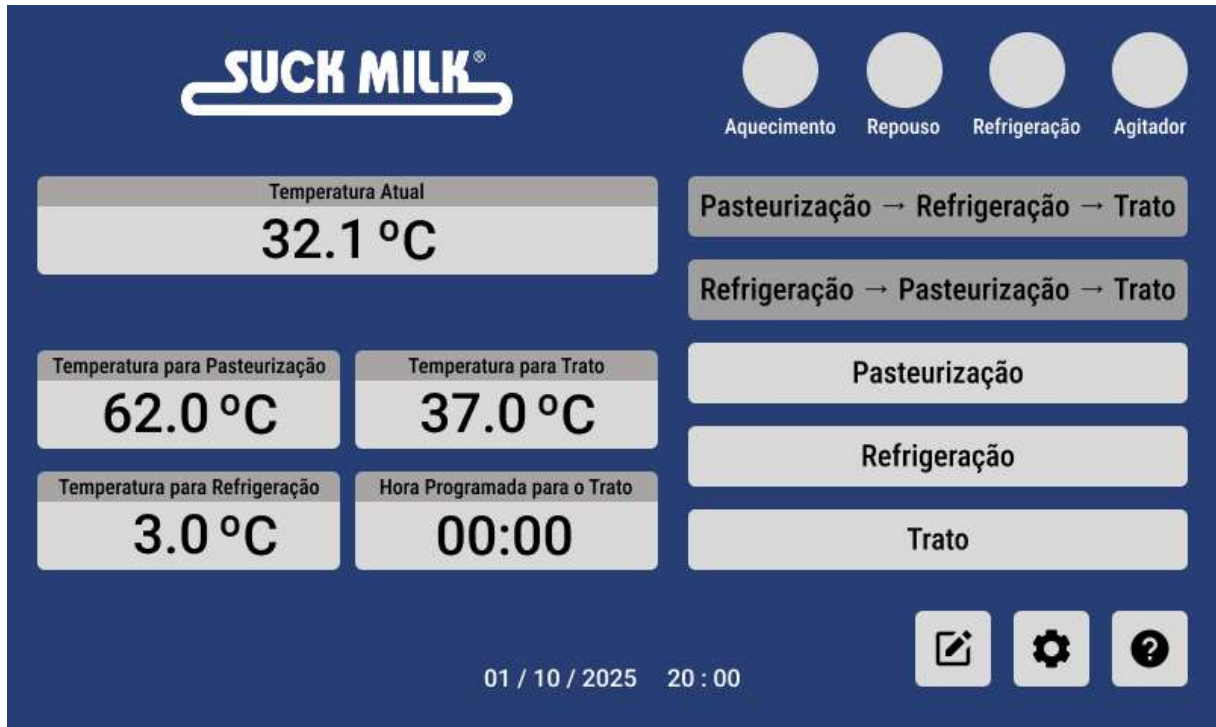


Figura 3.11 – Tela inicial da IHM (Fonte: Autor)



Figura 3.12 – Tela de andamento da IHM (Fonte: Autor)

nho técnico, focado em fornecer uma ideia base para o leitor de como o sistema foi desenvolvido e como o funcionamento cumpre com os requisitos propostos.

3.4.3 Gateway Modbus TCP/IP

No lado da automação, o CLP envia os valores das variáveis para o Sistema Web durante o andamento do processo executado pelo equipamento. Esse envio é realizado através do protocolo Modbus TCP/IP, onde o equipamento está conectado via cabo Ethernet à rede local, fazendo uso da internet para o envio desses dados. Para a configuração do Modbus TCP/IP no CLP, foram criados dois programas: o primeiro em ladder com a lógica para o envio das mensagens, e o segundo em texto estruturado com o conteúdo das mensagens.

Inicialmente, foram elencados 3 tipos de envios de mensagens importantes para o cliente e para a empresa fabricante:

- Status: dados do funcionamento do equipamento em operação;
- Configurações: dados das configurações definidas pela IHM;
- Erros: problemas registrados durante a operação do equipamento.

No programa em ladder, foram programadas as lógicas para o envio das mensagens, uma para cada tipo descrito acima. A seguir, será detalhado o envio das mensagens do primeiro tipo: status. Na Figura 3.13 é apresentada a lógica utilizada para o acionamento do envio das mensagens do tipo status. Para indicar que o processo foi iniciado, são utilizadas as variáveis `inicio_past`, `inicio_refrig` e `inicio_trato`, dessa forma o CLP irá informar ao Sistema Web quando foi iniciado um processo no equipamento. Importante destacar que o contato de cada variável contém o P, que irá captar apenas na borda de subida, quando a variável passar de OFF para ON.

Para indicar que o processo foi finalizado, são utilizadas as variáveis `aviso_past`, `aviso_trato`, `parar_processo` e `parada_emergencia`, dessa forma o CLP irá informar ao Sistema Web quando foi finalizado um processo no equipamento. Por fim, há o desacionamento da mensagem de início e de andamento, assim que é finalizado o processo do equipamento.

Na Figura 3.14, é realizado o envio propriamente dito da mensagem, essa mensagem é referente ao momento em que é iniciado o processo. Caso haja um erro nesse envio, será realizada a contagem de 30 segundos em um timer, para tentar enviar a mensagem novamente. O módulo Modbus utilizado para esse envio é o `MSG_MODBUS2`, e as variáveis utilizadas nele serão explicadas na decorrência dessa seção.

Na Figura 3.15 é realizada a conferência do erro da mensagem, que aciona o timer. Nessa lógica, foi definido que seriam feitas cinco tentativas. Caso não ob-



Figura 3.13 – Acionamento do envio das mensagens do tipo status (Fonte: Autor)



Figura 3.14 – Envio das mensagens do tipo status (Fonte: Autor)

tenha sucesso, as mensagens são finalizadas (se o cliente não conectar na rede, o equipamento não enviará essas mensagens, resultando nesse erro).

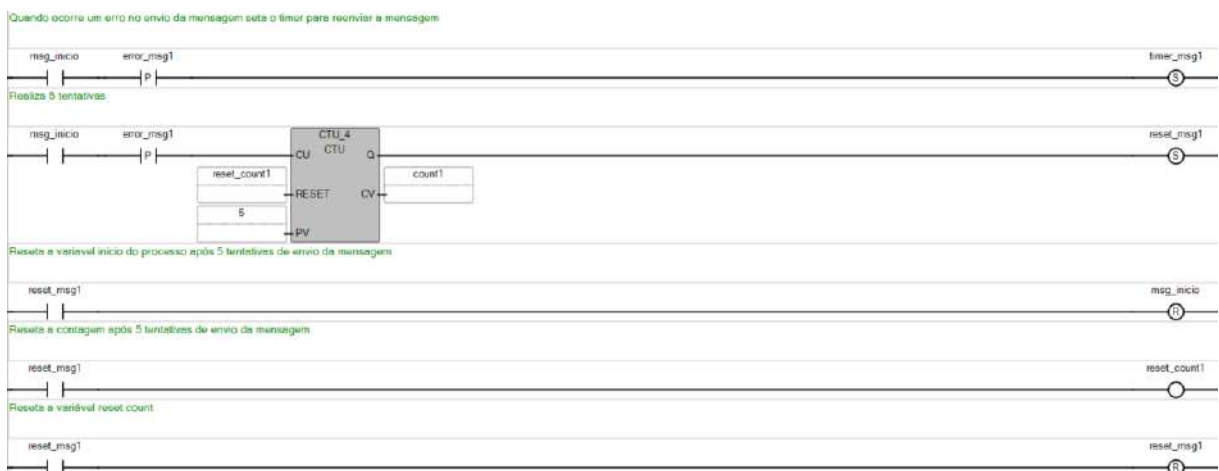


Figura 3.15 – Erro no envio das mensagens do tipo status (Fonte: Autor)

Se a mensagem for enviada com sucesso, a variável `msg_inicio` é desacionada e a variável `msg_andamento` é acionada, assim inicia-se o envio das mensagens durante o processo do equipamento. Esse envio permanece ativo durante o andamento do processo e é terminado quando o processo é finalizado, e então é enviada a mensagem de finalização. Conforme demonstrado na Figura 3.16.



Figura 3.16 – Acionamento do envio de andamento (Fonte: Autor)

No programa em texto estruturado, foram configuradas as estruturas das mensagens, conforme a Figura 3.17.

```

1 // LocalCfg (Configuração do lado CLP - origem da mensagem Modbus)
2 msgLocalCfg.Channel := 4; // Canal 4 = porta Ethernet do Micro820
3 msgLocalCfg.TriggerType := 0; // 0 = trigger por borda (executa quando há borda no trigger).
4 msgLocalCfg.Cmd := 16; // 3 = Read Holding Registers; 16 = Write Miltiple Registers
5 msgLocalCfg.ElementCnt := 17; // Número de registradores Modbus que serão lidos/enviados
6
7
8 // TargetCfg (Configuração do destino - servidor Python que recebe os dados)
9 msgTargetCfg.Addr := 1; // Identificador interno do destino (ID da configuração)
10 msgTargetCfg.NodeAddress[0] := 192; // Endereço IP do servidor destino (192.168.68.139)
11 msgTargetCfg.NodeAddress[1] := 168;
12 msgTargetCfg.NodeAddress[2] := 68;
13 msgTargetCfg.NodeAddress[3] := 68;
14 msgTargetCfg.Port := 1502; // Porta Modbus/TCP padrão (502)
15 msgTargetCfg.UnitId := 1; // Unit ID (endereço de escravo no Modbus/TCP, geralmente 1 ou 255)
16 msgTargetCfg.MsgTimeout := 2000; // Tempo máximo (ms) de espera por resposta do servidor
17 msgTargetCfg.ConnTimeout := 3000; // Tempo (ms) para estabelecer a conexão TCP
18 msgTargetCfg.ConnClose := false; // false = mantém conexão aberta; true = fecha após cada requisição
19
20
21 // LocalAddr - Tabela Statuses (Endereço e conteúdo dos dados enviados/armazenados)
22 msgLocalAddr1[1] := ANY_TO_WORD(0); // Flag: 0=statuses, 1=configs, 3=errors
23 msgLocalAddr1[2] := ANY_TO_WORD(11673); // numero_serie
24 msgLocalAddr1[3] := ANY_TO_WORD(temperatura_dint);
25 msgLocalAddr1[4] := ANY_TO_WORD(status);
26 msgLocalAddr1[5] := ANY_TO_WORD(processo);
27 msgLocalAddr1[6] := ANY_TO_WORD(timer1);
28 msgLocalAddr1[7] := ANY_TO_WORD(timer2);
29 msgLocalAddr1[8] := ANY_TO_WORD(timer3);
30 msgLocalAddr1[9] := ANY_TO_WORD(horas);
31 msgLocalAddr1[10] := ANY_TO_WORD(minutos);
32 msgLocalAddr1[11] := ANY_TO_WORD(0);
33 msgLocalAddr1[12] := ANY_TO_WORD(0);
34 msgLocalAddr1[13] := ANY_TO_WORD(0);
35 msgLocalAddr1[14] := ANY_TO_WORD(0);
36 msgLocalAddr1[15] := ANY_TO_WORD(0);
37 msgLocalAddr1[16] := ANY_TO_WORD(0);
38 msgLocalAddr1[17] := ANY_TO_WORD(0);

```

Figura 3.17 – Conteúdo das mensagens de status (Fonte: Autor)

A variável `msgLocalCfg` é referente à configuração local - cliente Modbus no CLP. A seguir serão detalhadas as configurações necessárias para o funcionamento do envio de mensagens:

- Channel: número da porta de Ethernet local, para o Micro820 o padrão é a porta 4;
- TriggerType: tipo de disparo da mensagem, nesse caso foi escolhido o tipo 0, que corresponde ao disparo único da mensagem, quando a entrada do módulo passa de False para True;
- Cmd: código da função Modbus, o valor 16 é referente à escrita de múltiplos registradores, cada opção existente foi descrita na seção 2.4;
- ElementCnt: número de registradores que serão escritos na mensagem, como a maior mensagem das três que foram definidas é de 16 campos, então o número de registradores foi definido como 17, onde o primeiro campo representa o tipo da mensagem para o servidor em Python conseguir colocar na tabela certa.

A variável msgTargetCfg é referente à configuração de destino - servidor Modbus Python. A seguir serão detalhadas as configurações necessárias para o funcionamento do envio de mensagens:

- Addr: endereço dos dados Modbus do dispositivo de destino, por padrão o valor é 1;
- NodeAddress[4]: endereço IP do servidor de destino, nesse caso o IP utilizado é o 192.168.68.68, esse IP é dividido em 4 campos na variável;
- Port: número da porta TCP de destino, a porta padrão é 502, porém essa porta precisa de autorização do nível administrador, então para o projeto foi escolhida a porta 1502;
- UnitId: identificador da unidade usada para comunicação com dispositivos escravos por uma ponte Modbus, o intervalo de valor aceito é 0-255 e o valor escolhido é 1;
- MsgTimeout: tempo limite da mensagem em milissegundos, a partir do teste foi analisada a demora no envio da mensagem e para isso foi definido o valor de 2000 ms, para que tivesse tempo suficiente;
- ConnTimeout: tempo limite do estabelecimento da conexão TCP em milissegundos, também foi definido através da realização de testes, o valor escolhido é 3000 ms;
- ConnClose: Comportamento de fechamento da conexão TCP, o valor escolhido é False, assim a conexão não é fechada na conclusão da mensagem.

A variável `msgLocalAddr` é referente aos dados locais a serem enviados, nesse caso, são os dados do status do equipamento. O primeiro campo é utilizado para identificar a tabela em que deve ser inserida essa mensagem, do lado do servidor Python. A flag pode receber 3 valores: 0 - tabela "statuses", 1 - tabela "configs" ou 3 - tabela "errors".

Na sequência, tem os 16 registradores, em cada um é colocado os valores das variáveis para inserir no banco de dados. A tabela "statuses" possui os seguintes campos:

- `serialNumber`;
- `temperature`;
- `status`;
- `process`;
- `restTimer`;
- `stirringTimer`;
- `breakTimer`;
- `scheduledTime`.

Na Figura 3.17 são identificados os 17 registradores, o primeiro identifica a tabela, e o restante são os campos da tabela "statuses", esses campos estão descritos acima, com um adendo, o `scheduledTime` é dividido em dois registradores, um para as horas e outro para os minutos. Os 7 últimos registradores excedem os campos da tabela, assim são preenchidos com 0, e são ignorados no lado do servidor Python. A função `ANY_TO_WORD(...)` é utilizada para converter o valor da entrada para o tipo de dado `WORD` (um inteiro de 16 bits não assinado), que é o formato padrão dos registradores Modbus.

Os programas em ladder e texto estruturado, em conjunto, realizam os comandos e configurações necessários para o envio das mensagens do lado do CLP. No próximo capítulo, será apresentado e detalhado o Sistema Web criado para realizar o monitoramento e controle do CLP remotamente. Serão apresentados os requisitos funcionais e não funcionais, e a arquitetura do sistema, contemplando o banco de dados, o *back-end*, o *front-end*, o *gateway* e o protocolo Modbus TCP/IP.

4. SUPERVISÃO DA AUTOMAÇÃO: SISTEMA WEB

Neste capítulo será apresentado o Sistema Web. Inicialmente, será descrita a ideia do sistema, em seguida serão apresentados os requisitos funcionais e não funcionais do Sistema Web. Por fim, será detalhada a arquitetura proposta para esse sistema.

4.1 O Sistema Web

Na Indústria 4.0, os equipamentos ficam interligados e se comunicam entre si, permitindo o controle remoto da operação. Para este projeto, o objetivo é utilizar a tecnologia Ethernet para comunicar e controlar cada equipamento disposto em uma agroindústria leiteira. Esses equipamentos devem estar conectados a uma rede local de internet e a um computador que irá realizar a comunicação com o servidor online da empresa fabricante.

Essa conexão permitirá o envio e o armazenamento de dados no servidor destinado a esse sistema. Essas informações serão disponibilizadas ao cliente por meio do Sistema Web, que conta com espelhamento da IHM e geração de relatórios para a gestão da produção. Esses dados também contribuem para fomentar melhorias e realizar manutenções nos equipamentos.

Além de este sistema proporcionar mais funcionalidades e comodidade para o cliente e a empresa fabricante, ele se torna um diferencial entre os equipamentos ofertados no mercado. Esse também é um dos objetivos que devem ser alcançados com esse projeto, uma vez que, para uma empresa se manter no mercado, ela necessita estar se reinventando sempre.

4.2 Requisitos do Sistema Web

Considerando as necessidades da automação do equipamento, as melhorias identificadas no fluxo de operação e a importância de garantir um acompanhamento eficiente pelo sistema, a seguir são apresentados os requisitos funcionais e não funcionais do Sistema Web desenvolvido para a automação do pasteurizador.

4.2.1 Requisitos Funcionais

- RF01 – Tela de login: o Sistema Web deve possuir uma tela de login, garantindo que somente usuários autorizados acessem os dados;
- RF02 – Acesso ao banco de dados: o sistema deve conectar-se ao servidor em nuvem e listar as informações armazenadas conforme o que foi programado;
- RF03 – Tela de configurações: o sistema deve disponibilizar uma tela específica para administradores, onde os administradores possam visualizar os usuários e equipamentos cadastrados;
- RF04 – Listagem de equipamentos: na tela inicial, o sistema deve exibir uma lista dos equipamentos vinculados ao usuário logado;
- RF05 – Espelhamento da IHM: como principal funcionalidade, o sistema deve permitir o espelhamento da IHM, possibilitando que o usuário controle seus equipamentos remotamente;
- RF06 – Geração de relatórios: o sistema deve possibilitar a geração de relatórios com o histórico de status e configurações dos equipamentos, tanto para visualização na plataforma quanto para exportação em PDF;
- RF07 – Comunicação entre CLP e sistema: o sistema deve conter um programa responsável por receber as mensagens enviadas pelo CLP, organizar as informações e inseri-las no banco de dados;
- RF08 – Visualizar dados do equipamento: o sistema deve apresentar as informações principais de cada equipamento da listagem, tais como nome, número de série, processo selecionado, etapa em andamento e status;
- RF09 – Cadastros: o sistema deve permitir que administradores cadastrem e editem usuários e equipamentos, esses usuários serão utilizados para o login no sistema, e os equipamentos servem para definir cada um com seus dados. Esses dados são utilizados para identificar cada um nos registros e também para fazer o espelhamento da IHM desse equipamento.

4.2.2 Requisitos Não Funcionais

- RNF01 – Confiabilidade: o sistema deve operar de forma estável, sem quedas ou perda de informações, garantindo dados atualizados e consistentes;

- RNF02 – Segurança: o sistema deve ser desenvolvido com foco em segurança, evitando acessos não autorizados e vazamento de dados;
- RNF03 – Manutenibilidade: o código-fonte deve ser estruturado e documentado, facilitando futuras atualizações e correções;
- RNF04 – Desempenho: o sistema deve responder às ações do usuário em tempo hábil, mantendo boa experiência de uso mesmo com múltiplas requisições simultâneas;
- RNF05 – Compatibilidade: o Sistema Web deve ser compatível com os principais navegadores e dispositivos modernos.

4.3 Arquitetura do Sistema Web

A seguir, serão apresentadas as partes que compõem a arquitetura do sistema de monitoramento e controle da automação. O detalhamento abrange o banco de dados, *back-end*, *front-end*, comunicação entre CLP e Sistema Web, Modbus TCP/IP e as funcionalidades do sistema. A linha de pensamento e a explicação ocorrerão de acordo com a ordem de desenvolvimento realizada para esse projeto.

4.3.1 Banco de Dados

O banco de dados foi iniciado com a ideia de que o CLP iria enviar os dados do funcionamento do equipamento para o servidor. Assim, foi desenhada a tabela *statuses*, que contém os registros das informações do andamento do processo. Em seguida, foram realizados testes para verificar a configuração e a comunicação do banco com o sistema.

Essa tabela possui as seguintes informações: número de série do equipamento, temperatura atual, etapa em andamento, processo selecionado, timer de repouso, timer de agitação, timer de descanso e hora programada. Por fim, há a data de criação e edição do registro. Esses dados foram considerados obrigatórios e contribuem para que o cliente tenha o acompanhamento do processo.

Com o andamento do projeto, foi identificado que seria interessante também registrar as configurações que o cliente definiu no equipamento. Por esse motivo, foi criada a tabela *configs*, que contém as configurações do equipamento. Esses dados são enviados pelo CLP sempre que são alterados na IHM.

Essa tabela possui as seguintes informações: número de série do equipamento, temperatura de pasteurização, temperatura de refrigeração, temperatura de

trato, tempo de agitação, tempo de descanso, tempo de repouso, diferencial da temperatura para refrigeração, e a data e hora. Por fim, há a data de criação e edição do registro. Esses dados foram considerados obrigatórios e contribuem para a empresa fabricante ter o conhecimento das configurações utilizadas pelo cliente.

Para a tela de login, é obrigatório possuir o cadastro de usuários para poder acessar o sistema, assim, foi criada a tabela *users*, que contém os seguintes dados: nome, login, senha, tipo e data de criação e edição do registro.

As funcionalidades de monitoramento e controle do equipamento necessitam dos dados desse equipamento, então foi criada a tabela *equipments*, que contém todas as informações para serem utilizadas no sistema. Nessa tabela estão as seguintes informações: número de série do equipamento, nome, capacidade, IP do CLP, IP da IHM, senha da IHM e ID do usuário. Por fim, há a data de criação e edição do registro.

O ID do usuário da tabela *equipments* é utilizado como função de posse, atribuindo, assim, o cliente ao equipamento. Inicialmente, seria somente para referenciar o cliente ao equipamento e auxiliar na identificação por parte da empresa. Entretanto, outras funcionalidades do sistema demonstraram que seria interessante fazer o uso desses dados. Por exemplo, a listagem de equipamentos na tela inicial, que busca somente os equipamentos atribuídos ao usuário que realizou o login e que estão conectados ao computador, limita bastante a lista de equipamentos e reduz o tempo de execução do SQL.

Por último, foi criada a tabela *errors*, responsável por registrar os acontecimentos de erros encontrados durante o andamento do processo. Essa funcionalidade será desenvolvida futuramente e tem o objetivo de contribuir com a empresa para melhorias e manutenções.

Na Figura 4.1 é apresentado o diagrama EER do banco de dados desenvolvido em MySQL para esse projeto:

4.3.2 Back-end

O desenvolvimento do *back-end* do Sistema Web foi realizado em Node.js. Inicialmente, foram criados os modelos para as tabelas do banco de dados com o Sequelize, em seguida foram realizadas as configurações do banco no projeto e iniciada a estruturação do arquivo *app.js*, responsável pelas operações do CRUD (Create/Criar, Read/Ler, Update/Atualizar e Delete/Deletar). Para a criação de rotas e requisições HTTP, foi utilizado o framework Express.

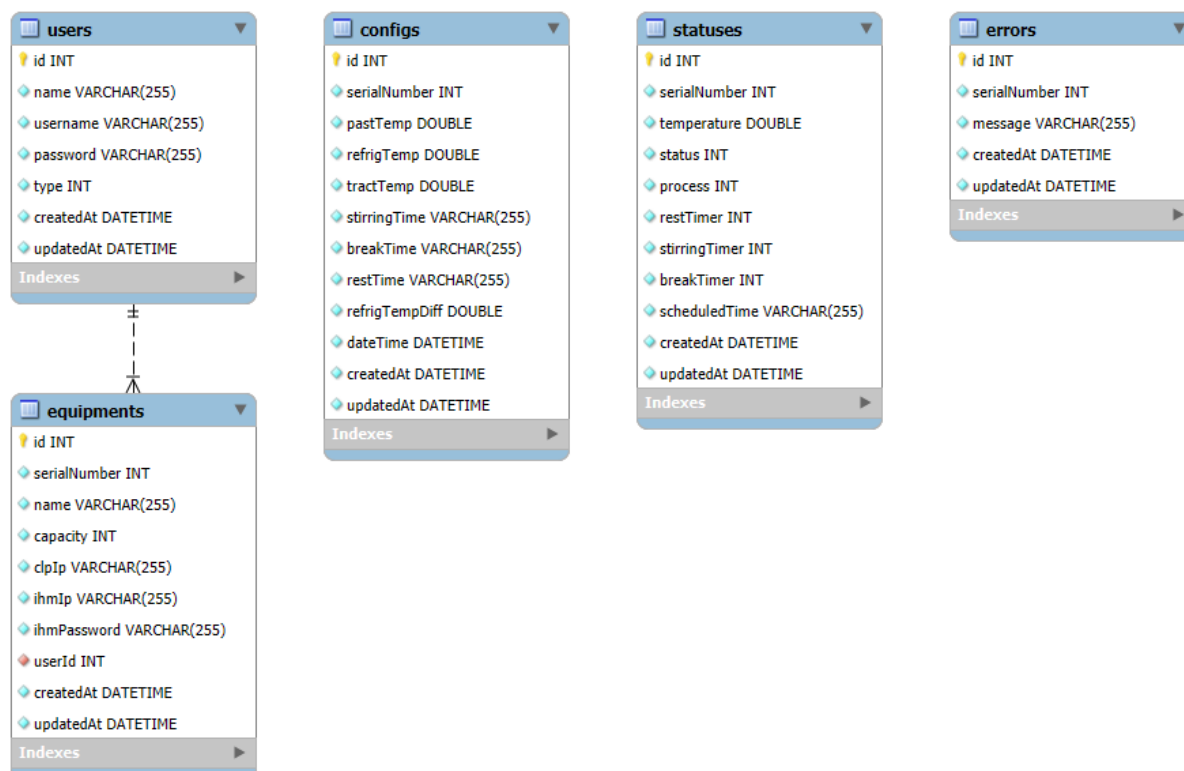


Figura 4.1 – Diagrama EER (Fonte: Autor)

Após a criação das requisições para cada tabela, foi realizada a função de autenticação de login, para a qual é necessário realizar uma busca para encontrar um usuário cadastrado com o mesmo username inserido, em seguida, é comparada a senha inserida com a senha cadastrada, por fim, o sistema retorna o usuário com um token de expiração para esse login, caso o username e a senha não confirmem, o sistema irá retornar uma mensagem de erro.

Conforme o projeto foi sendo ampliado e aprimorado, aumentou-se a necessidade de intensificar a segurança do sistema. Para isso, foram desenvolvidas diversas funcionalidades, tais como a autenticação baseada em token temporário, na qual o logout é realizado automaticamente após a expiração do token. Além disso, as páginas de gerenciamento de equipamentos e usuários são restritas a administradores, sendo realizada a verificação do tipo do usuário autenticado. Também foram implementadas rotas autenticadas e rotas administrativas, ambas protegidas por mecanismos de redirecionamento e controle de acesso, validados por meio do encapsulamento global da segurança. Essas funcionalidades foram desenvolvidas tanto no *back-end* quanto no *front-end*.

A geração de relatórios em PDF foi desenvolvida com o JsPDF, sendo uma ferramenta própria para criar PDFs com JavaScript. No código, é configurado o do-

cumento em modo paisagem, inserindo o logotipo da empresa fabricante, o título do relatório, informações do equipamento, período selecionado, data e usuário logado. Em seguida, os registros são organizados em tabela e estilizados de acordo com o solicitado pela empresa. A Figura 4.2 demonstra como é o layout do PDF gerado:

Id	N° série	Temperatura	Processo	Status	Timer Repouso	Timer Agitação	Timer Descanso	Hora Programada	Data Criação	Data Edição
1	11673	19.8 °C	Sem trabalho		0	0	0	00:00	07/10/2025 10:08:47	07/10/2025 10:08:47
2	11673	19.7 °C	Sem trabalho		0	0	0	00:00	07/10/2025 10:09:46	07/10/2025 10:09:46
3	11673	19.7 °C	Sem trabalho		0	0	0	00:00	07/10/2025 10:10:46	07/10/2025 10:10:46
4	11673	19.8 °C	Sem trabalho		0	0	0	00:00	07/10/2025 10:11:46	07/10/2025 10:11:46
5	11673	19.8 °C	Sem trabalho		0	0	0	00:00	07/10/2025 10:12:46	07/10/2025 10:12:46
6	11673	19.7 °C	Sem trabalho		0	0	0	00:00	07/10/2025 10:13:46	07/10/2025 10:13:46
7	11673	19.7 °C	Sem trabalho		0	0	0	00:00	07/10/2025 10:19:46	07/10/2025 10:19:46
8	11673	19.7 °C	Sem trabalho		0	0	0	00:00	07/10/2025 10:20:46	07/10/2025 10:20:46
9	11673	19.7 °C	Sem trabalho		0	0	0	00:00	07/10/2025 10:21:47	07/10/2025 10:21:47
10	11673	19.7 °C	Sem trabalho		0	0	0	00:00	07/10/2025 10:22:47	07/10/2025 10:22:47
11	11673	19.8 °C	Refrigeração	Refrigeração	0	0	0	00:00	07/10/2025 10:23:47	07/10/2025 10:23:47
12	11673	19.9 °C	Sem trabalho		0	0	0	00:00	07/10/2025 10:24:47	07/10/2025 10:24:47
13	11673	19.8 °C	Sem trabalho		0	0	0	00:00	07/10/2025 10:25:48	07/10/2025 10:25:48

Figura 4.2 – Layout do PDF (Fonte: Autor)

Para complementar o projeto, alguns arquivos foram criados para auxiliar em algumas tarefas no código, por exemplo, encontrar uma porta de conexão disponível para uma IHM, calcular o período selecionado para os relatórios, gerar opções de mês/ano a partir de 2025 para os relatórios, entre outros.

4.3.3 Front-end

O desenvolvimento do *front-end* do Sistema Web foi realizado em React.js. Inicialmente, foi criada a tela de login, seguindo um layout já existente e realizando a personalização deste para atender o esperado pela empresa fabricante. Em seguida, a tela de cadastro de usuários foi criada, permitindo que os administradores do sistema pudessem criar quantos usuários fossem necessários.

Entretanto, somente o cadastro dos usuários não é suficiente, é necessário poder listar, cadastrar, editar e excluir esses registros. Assim, foi criada uma tela para gerenciar os usuários. Essa tela contempla a listagem dos usuários com o campo de pesquisa para filtrar a listagem, um botão para realizar o cadastro dos usuários, e abaixo a listagem dos usuários com botões ao lado para editar ou excluir esse registro.

Para o cadastro e edição dos usuários, foi desenvolvido um pop-up, com os campos obrigatórios de acordo com o definido na tabela do banco de dados. Após

desenvolver todo o código para o gerenciamento dos usuários, foi replicado e alterado para atender à demanda do gerenciamento dos equipamentos. A lógica da estruturação é a mesma, mas diferem quanto aos campos e as requisições HTTP.

Com os cadastros concluídos, o próximo passo foi criar a tela inicial. Na Figura 4.3 é apresentada a tela inicial, nela foi desenvolvida uma estrutura bastante complexa. Iniciamos com a listagem de equipamentos localizada na parte esquerda da tela, onde é utilizado como filtro o usuário logado, e também essa listagem somente trará os equipamentos que estão conectados à rede local, dessa forma, é feita uma busca por IP da IHM para realizar essa identificação.

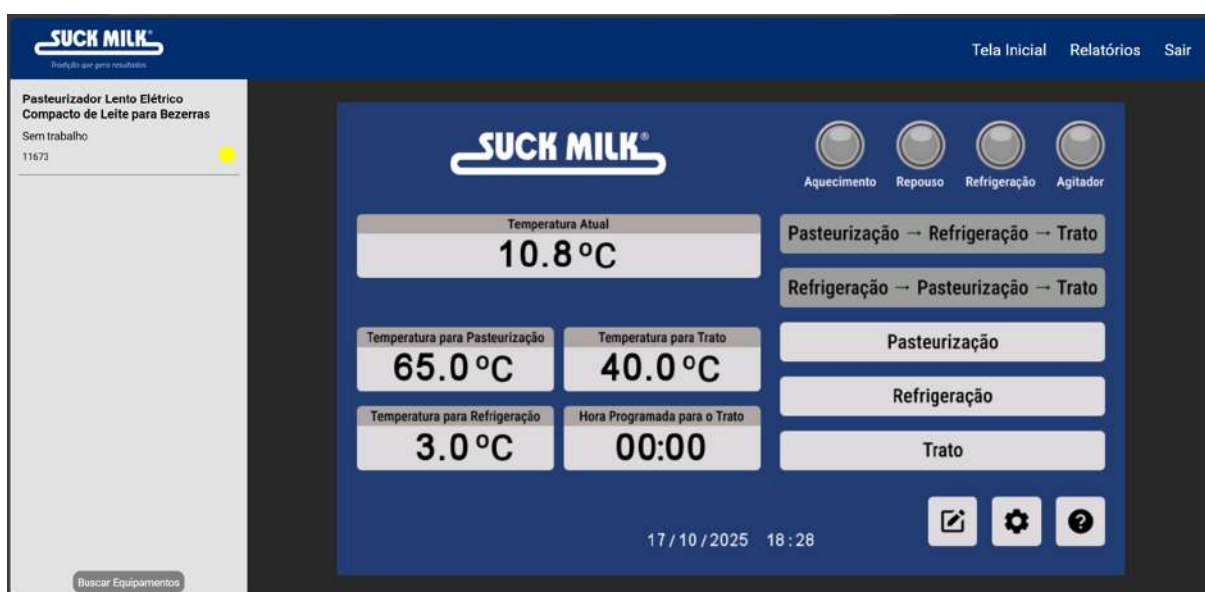


Figura 4.3 – Tela inicial (Fonte: Autor)

Cada item dessa lista é um botão, e ao clicar nele será feita uma requisição para o sistema realizar o espelhamento da IHM com o auxílio do noVNC, protocolo que permite acessar e controlar remotamente outro computador; nesse caso, a IHM que está conectada ao CLP. Para essa conexão, é necessário utilizar o programa comentado na seção anterior, para encontrar uma porta disponível a fim de iniciar a comunicação com a IHM.

Em seguida, as configurações são definidas dentro do VNCViewer, criado com o noVNC, é definida a parte de autenticação da conexão e o layout que deve ser apresentado quando é espelhada a tela da IHM pelo computador/notebook. Essa funcionalidade gerou a ideia de criar o Sistema Web, sendo responsável por permitir ao operador monitorar e controlar o equipamento, sem a necessidade de estar em frente ao painel elétrico.

Por fim, foi criada a tela para geração de relatórios, que será utilizada pelos clientes e pelo suporte técnico da empresa fabricante para acompanhar o funciona-

mento dos equipamentos. Nessa tela é apresentada a listagem dos dados de acordo com os filtros: modo, número de série do equipamento e mês/ano de referência. Na Figura 4.4 é possível identificar esses filtros. Para gerar um relatório, primeiro é preciso selecionar todos os filtros e clicar em pesquisar. Os dados serão apresentados na tela para a verificação. Ao lado do botão de pesquisar, há o botão do PDF. Ao pressionar, o relatório é gerado em PDF e o navegador realiza automaticamente o download.

Id	Nº série	Temperatura	Processo	Status	Timer Repouso	Timer Agitação	Timer Descanso	Hora Programada	Data Criação	Data Edição
55	11673	17.4 °C	Sem trabalho		0	0	0	00:00	07/10/2025 11:08:48	07/10/2025 11:08:48
56	11673	17.4 °C	Refrigeração → Pasteurização → Trato	Refrigeração	0	0	0	19:00	07/10/2025 11:09:46	07/10/2025 11:09:46
57	11673	17.4 °C	Refrigeração → Pasteurização → Trato	Refrigeração	0	0	0	19:00	07/10/2025 11:10:46	07/10/2025 11:10:46
58	11673	17.2 °C	Refrigeração → Pasteurização → Trato	Refrigeração	0	0	0	19:00	07/10/2025 11:11:47	07/10/2025 11:11:47
59	11673	16.9 °C	Refrigeração → Pasteurização → Trato	Refrigeração	0	0	0	19:00	07/10/2025 11:12:47	07/10/2025 11:12:47
60	11673	16.6 °C	Refrigeração → Pasteurização → Trato	Refrigeração	0	0	0	19:00	07/10/2025 11:13:47	07/10/2025 11:13:47
61	11673	16.3 °C	Sem trabalho		0	0	0	00:00	07/10/2025 11:14:48	07/10/2025 11:14:48
62	11673	16.2 °C	Refrigeração	Refrigeração	0	0	0	00:00	09/10/2025 18:09:52	09/10/2025 18:09:52

Figura 4.4 – Tela de relatórios (Fonte: Autor)

4.3.4 Gateway - Comunicação entre CLP e Sistema Web

O sistema foi desenvolvido e testado localmente, de modo que a comunicação entre o CLP, a IHM e o Sistema Web é realizada por meio de um cabo Ethernet. Essa tecnologia será utilizada posteriormente por questões de segurança da operação do equipamento. Antes de lançar esse produto, o Sistema Web será hospedado em um servidor em nuvem para que usuários de diferentes locais possam acessar.

O CLP utiliza a conexão com a internet para enviar as mensagens ao servidor, esse envio é feito por meio do Modbus TCP/IP, que será detalhado na seção 4.3.5.

A IHM está conectada ao *switch*, o qual está conectado à rede local, onde também é encontrado o computador do cliente. Essa comunicação é feita por meio de cabo Ethernet e é necessária para a funcionalidade de espelhamento da IHM realizada pelo Sistema Web. O sistema identifica as IHMs conectadas ao computador do usuário autenticado, e não o CLP. Essa funcionalidade foi detalhada na seção 4.3.3.

4.3.5 Modbus TCP/IP

Foi desenvolvido um servidor Modbus TCP/IP em Python que atua como gateway de comunicação entre o CLP e o Sistema Web. Esse servidor recebe as mensagens enviadas pelo CLP, realiza a leitura dos registradores e organiza os dados na estrutura esperada pelo banco de dados relacional MySQL. Em seguida, armazena esses dados em tabelas, no primeiro registrador há um valor que indica a tabela na qual os dados devem ser inseridos, podendo ser status, configurações ou erros. Essa solução de registro de mensagens é de grande valia para a rastreabilidade do produto e a manutenção preventiva ou corretiva do equipamento.

Mapeamento Modbus e Estrutura dos Registradores

A comunicação entre o CLP e o gateway Modbus é realizada por meio da função Modbus *Write Multiple Registers* (código de função 16). O CLP escreve a cada 30 segundos um conjunto de dados em registradores do tipo *Holding Register*, esses dados são referentes ao funcionamento do equipamento, informações sobre o andamento do processo e configurações.

O servidor Modbus desenvolvido em Python realiza a leitura desses registradores por meio da função *Read Holding Registers* (código de função 03) a cada 2 segundos. Após a leitura, o gateway reorganiza os dados recebidos e armazena as informações no banco de dados. Em seguida, os registradores são zerados do lado do servidor, evitando leituras duplicadas e garantindo a integridade dos dados registrados.

A Tabela 4.1 apresenta o mapeamento dos registradores Modbus utilizados, indicando o endereço de cada registrador, a função Modbus empregada, a descrição do dado transmitido e a operação realizada:

Programação Servidor Python

A seguir, detalha-se este programa com o objetivo de compreender melhor o seu desenvolvimento. A primeira função do programa prepara um conjunto de conexões MySQL para serem utilizadas de forma rápida e assíncrona no restante do sistema. Foi desenvolvido assim para poder escalar o programa facilmente, futuramente haverá várias conexões ao mesmo tempo.

O código abaixo define uma função assíncrona denominada *create_pool*, responsável por criar um objeto do tipo *pool*, que gerencia diversas conexões simultâneas com o MySQL. Os parâmetros necessários para a criação desse *pool* são:

Endereço	Função Modbus (CLP / Gateway)	Descrição	Operação
0	16 / 03	Tipo da mensagem (0 = status)	Escrita / Leitura
1	16 / 03	Número de série do equipamento	Escrita / Leitura
2	16 / 03	Temperatura do sensor (°C)	Escrita / Leitura
3	16 / 03	Etapa do processo	Escrita / Leitura
4	16 / 03	Processo selecionado	Escrita / Leitura
5	16 / 03	Temporizador de repouso (s)	Escrita / Leitura
6	16 / 03	Temporizador de agitação (s)	Escrita / Leitura
7	16 / 03	Temporizador de descanso (s)	Escrita / Leitura
8	16 / 03	Hora programada – horas	Escrita / Leitura
9	16 / 03	Hora programada – minutos	Escrita / Leitura
10–16	16 / 03	Reservado	Escrita / Leitura

Tabela 4.1 – Mapeamento dos registradores Modbus utilizados no gateway

- *host*: endereço do servidor MySQL;
- *port*: porta do MySQL;
- *user*: usuário do banco de dados;
- *password*: senha do usuário;
- *db*: nome do banco de dados a ser utilizado;
- *minsize*: número mínimo de conexões abertas;
- *maxsize*: número máximo de conexões abertas;
- *autocommit*: habilita o commit automático das operações SQL.

```

async def create_pool():
    return await aiomysql.create_pool(
        host=MYSQL_HOST,
        port=MYSQL_PORT,
        user=MYSQL_USER,
        password=MYSQL_PASSWORD,
        db=MYSQL_DATABASE,
        minsize=1,
        maxsize=10,
        autocommit=True
    )

```

Quadro 4.1 – Função responsável pela criação do pool de conexões assíncronas com o banco de dados MySQL

A segunda função do programa recebe os dados já organizados que vieram do CLP, cria um SQL com esses dados e salva no MySQL de forma assíncrona usando o pool de conexões. Também foi criado um comando para que a resposta dessa inserção seja registrada em um arquivo de log.

O código abaixo define uma função assíncrona denominada *insert_status* que recebe o *pool* criado na função anterior, e o *data*, valores dos registradores enviados pelo CLP. Nessa função, é utilizado o bloco *try*, que encapsula toda a operação de inserção dos dados no banco, e *except*, que captura qualquer exceção que possa ocorrer durante a execução do *try*. Em caso de erro, uma mensagem é registrada no arquivo de log, permitindo que em outro momento sejam verificados os erros para uma correção.

Dentro do *try*, é solicitada uma conexão ao *pool* de forma assíncrona, em seguida, cria-se um cursor para executar o comando SQL. Na variável *sql* é definido o SQL para fazer o insert dos dados na tabela do banco de dados, nesse caso é referente à tabela *statuses* e seus campos. Após a definição do SQL, é realizada a execução com os dados recebidos do CLP, mapeando as variáveis com os campos da tabela. Por fim, é realizado o registro dos dados inseridos no arquivo de log.

```

async def insert_status(pool, data):
    try:
        async with pool.acquire() as conn:
            async with conn.cursor() as cur:
                sql = """ INSERT INTO statuses
                    (
                        serialNumber, temperature, status,
                        process,
                        restTimer, stirringTimer, breakTimer,
                        scheduledTime
                    ) VALUES (%s, %s, %s, %s, %s, %s, %s, %s) """
                await cur.execute(sql, (
                    data['serialNumber'], data['temperature'],
                    data['status'],
                    data['process'], data['restTimer'], data['
                        stirringTimer'],
                    data['breakTimer'], data['scheduledTime']
                ))
                log.info(f"[STATUS] Salvo: {data!r}")
    except Exception as e:

```

```
log.error("Erro ao salvar STATUS: %s", e)
```

Quadro 4.2 – Função responsável pela inserção assíncrona dos dados de status no banco de dados MySQL

A terceira função do programa é composta por um loop, que lê os dados do CLP a cada 2 segundos. Esses dados vêm do envio de mensagens via Modbus TCP/IP do CLP, o programa recebe esses dados, reorganiza-os em um objeto (*data_status*) e realiza a inserção no banco MySQL. Assim que concluída a leitura, é registrado no log. Caso haja algum erro, também será registrado no log.

O código abaixo define uma função assíncrona denominada *loop_clp*, que recebe o identificador do CLP (*slave_id*), o contexto Modbus que armazena os registradores desse CLP (*context*) e o (*pool*) que foi criado na primeira função do programa. O bloco *while True* define que a função será executada continuamente, sendo controlado o tempo de ativo com o bloco *await asyncio.sleep(2)*, que pausa a execução do loop por 2 segundos.

Em seguida, são utilizados os blocos *try* e *except*, assim como na segunda função do código. Dentro do *try*, é realizada a leitura dos 17 registradores, essa leitura é definida pelo código 3 do Modbus (*Read Holding Registers*), e o resultado dessa leitura é armazenado na variável *values*. É verificado se há dados nessa variável e o primeiro registrador para identificar a tabela em que essa mensagem será inserida.

Nesse exemplo, é apresentado quando o primeiro registrador é igual a zero, então os dados serão inseridos na tabela *statuses*. É criada uma variável *data_status* para organizar os dados recebidos, facilitando assim a inserção no banco de dados. Após esse mapeamento, o insert no banco de dados é executado, e por fim, todos os registradores são zerados, com o objetivo de não processar os mesmos dados novamente.

```
async def loop_clp(slave_id, context, pool):
    while True:
        try:
            values = context[slave_id].getValues(3, 0, count=17)

            if any(values):
                if values[0] == 0: # STATUSES
                    data_status = {
                        'serialNumber': values[1],
                        'temperature': values[2]/10,
                        'status': values[3],
                        'process': values[4],
```

```

        'restTimer': values[5],
        'stirringTimer': values[6],
        'breakTimer': values[7],
        'scheduledTime': f"{values[8]:02d}:{
            values[9]:02d}"
    }
    await insert_status(pool, data_status)

    # Zera os registradores
    context[slave_id].setValues(3, 0, [0] * 17)

except Exception as e:
    log.error(f"Erro no loop do CLP {slave_id}: {e}")

await asyncio.sleep(2)

```

Quadro 4.3 – Função responsável pela leitura periódica dos dados do CLP e inserção no banco de dados

Finalizando, a quarta função do programa é a *main*, responsável por inicializar as funções anteriormente descritas. Inicialmente, o programa cria 10 CLPs simulados, cada um com 17 registradores (todos zerados) e junta-os em um contexto Modbus. Cria o pool de conexões MySQL e cria uma tarefa assíncrona para cada CLP, rodando os 10 em paralelo. Concluindo, inicia um servidor Modbus TCP na porta 1502.

O código abaixo define uma função assíncrona denominada *main*, responsável por inicializar todas as funções apresentadas anteriormente. Primeiramente, é definido o número de CLPs, nesse caso 10, e depois cria uma variável *slaves* com chave (identificador do CLP) e valor (memória interna de um escravo Modbus), essa variável é utilizada na criação do contexto global dos dados de um servidor Modbus (*ModbusServerContext*), esse módulo é responsável por gerenciar todos os *Modbus-SlaveContext*.

Em seguida é iniciado o *pool* de conexões com o banco de dados, e criada uma tarefa independente para cada CLP, cada uma dessas tarefas executa o *loop_clp* que realiza a leitura contínua dos dados enviados pelo CLP. Por fim, é inicializado o servidor Modbus TCP/IP no endereço 0.0.0.0 e na porta 1502, mantendo o servidor em execução contínua com o módulo *server.serve_forever*.

```

async def main():
    NUM_CLPS = 10

```

```

slaves = {
    i: ModbusSlaveContext(hr=ModbusSequentialDataBlock(1, [0]
        * 17))
    for i in range(1, NUM_CLPS + 1)
}

context = ModbusServerContext(slaves=slaves, single=False)
pool = await create_pool()

# Cria uma tarefa para cada CLP
for i in range(1, NUM_CLPS + 1):
    asyncio.create_task(loop_clp(i, context, pool))

log.info(f"Servidor Modbus TCP rodando na porta 1502 para {
    NUM_CLPS} CLPs...")
server = ModbusTcpServer(context, address=("0.0.0.0", 1502))
await server.serve_forever()

if __name__ == "__main__":
    asyncio.run(main())

```

Quadro 4.4 – Função principal responsável pela inicialização do servidor Modbus TCP, criação do pool de conexões e gerenciamento das tarefas dos CLPs

Dessa forma, o CLP envia a mensagem para o IP e porta do servidor configurados no módulo MSG_MODBUS2 dentro do CCW, posteriormente o servidor Python recebe essa mensagem e realiza as funções acima, gravando os dados no banco. Com esses dados gravados, é possível realizar a geração de relatórios e estatísticas. O desenvolvimento desse servidor Modbus cumpre com o requisito RF07 e constitui uma contribuição acadêmica desse trabalho de conclusão de curso.

4.3.6 Funcionalidades

As funcionalidades presentes no sistema são:

- Espelhamento da IHM: o sistema lista os equipamentos conectados à rede local buscando pelo IP da IHM, nessa lista, cada registro é um botão que ao clicar, abre-se uma tela que reproduz o mesmo conteúdo contido na IHM, mostrando a operação do equipamento em tempo real. Esse espelhamento permite o monitoramento e controle do equipamento, e com isso o requisito RF05 é cumprido.

- Geração de relatórios: o sistema armazena os dados enviados pelo CLP no banco de dados. Para o cliente obter o acesso a esses dados, é necessário realizar a geração do relatório, que também pode ser gerado em PDF. Essa funcionalidade permite que o cliente possa gerenciar sua produção. A geração de relatórios cumpre com o requisito RF06.
- Cadastros: o sistema permite o cadastro de usuários e equipamentos, conforme mencionado no requisito RF09. Essa funcionalidade é pensada exclusivamente para os administradores do sistema realizarem, esse gerenciamento é utilizado em todas as telas do Sistema Web.

Todos os requisitos funcionais e não funcionais estabelecidos foram cumpridos no decorrer do desenvolvimento do Sistema Web.

No próximo capítulo, serão apresentados os resultados e conclusões desse trabalho.

5. CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo automatizar um equipamento industrial a partir do zero, inserindo e programando um CLP e uma IHM no projeto, a fim de controlar os processos existentes, bem como o desenvolvimento de um Sistema Web de supervisão para monitoramento e controle remoto do equipamento. Para isso, foi selecionado o Pasteurizador Lento Elétrico Compacto de Leite para Bezerras, e foi realizado um estudo sobre suas características; processos de pasteurização, refrigeração e trato; e dispositivos presentes no painel elétrico. Foram levantados os problemas do modelo atual, com o intuito de resolvê-los e propor uma solução alinhada aos conceitos da Indústria 4.0.

A automação do pasteurizador possibilitou maior precisão no controle das etapas do processo, garantindo padronização operacional do equipamento, atendendo à demanda do cliente de otimizar os processos e permitir que o operador não precise configurar e iniciar manualmente cada etapa. A utilização de um CLP integrado a uma IHM permitiu ao operador acompanhar dados de cada etapa do processo, realizar comandos de operação e configurar as variáveis de acordo com o protocolo que ele preferir. Outro aspecto relevante é a comunicação via Modbus TCP/IP, que possibilitou o envio das informações do CLP para o banco de dados do servidor dentro do Sistema Web. Essas informações são de grande valia para o cliente e para a empresa fabricante.

O Sistema Web implementado mostrou-se uma solução eficaz para a supervisão da automação, permitindo o acesso remoto às informações da operação, o espelhamento da IHM para controle do equipamento, o gerenciamento de usuários e equipamentos, bem como a geração de relatórios. Para atender aos requisitos funcionais e não funcionais, a estruturação da arquitetura do sistema contou com banco de dados relacional, *back-end* bem estruturado e organizado, e *front-end* moderno, minimalista e responsivo. Todas as ideias iniciais para o sistema foram desenvolvidas e superaram as expectativas da empresa fabricante.

O resultado final deste projeto agrega valor tanto para o cliente que compra esse equipamento quanto para a empresa fabricante que vende e presta suporte técnico. A solução desenvolvida contribui para a automatização da produção de lácteos, reduzindo a necessidade de intervenção manual do operador, facilitando o acompanhamento dos processos e dados obtidos, uma vez que as normas da vigilância sanitária devem ser atendidas do início ao fim da produção. Esse novo produto, criado a partir da automação, será um diferencial competitivo dentro desse mercado disputado de agroindústrias leiteiras.

Por fim, o trabalho demonstra que a integração entre automação industrial e sistemas web é uma alternativa viável e eficiente para o aprimoramento de processos produtivos, podendo ser aplicada em outros equipamentos e outras áreas de atuação. Este projeto também contribui para a evolução visual e organizacional das automações e sistemas industriais aplicados no mercado atual.

5.1 Trabalhos Futuros

Este projeto representa somente o primeiro passo para a empresa, pois a proposta consiste em automatizar a linha completa de equipamentos e o objetivo final é obter um laticínio 100% automatizado. Para atingir essa expectativa, será realizado o desenvolvimento dos CLPs e IHMs para cada equipamento, com base no que foi desenvolvido durante este projeto, visto que cada equipamento tem processos diferentes que devem seguir o protocolo da empresa.

Outro aspecto relevante é o Sistema Web, que foi desenvolvido já pensando nos outros equipamentos que irão integrar esse ambiente. O operador poderá supervisionar mais de um equipamento no sistema, permitindo o controle da produção por meio de equipamentos Suck Milk. Esse sistema ainda precisa ser adaptado para celulares, porém, será somente para monitoramento e não controle, devido à instabilidade na rede de energia da maioria dos clientes. Nesse quesito, os dados deverão ser obtidos do banco de dados, apenas para um acompanhamento da situação do processo que está em andamento no equipamento.

Outra melhoria que pode ser realizada é a implementação completa do módulo de tratamento e análise de erros; essa parte deve ser desenvolvida no CLP e no Sistema Web. Registrando e apresentando ao operador as falhas ocorridas durante o processo, esses dados contribuem para manutenções e melhorias do equipamento.

No Sistema Web, deve ser desenvolvida uma tela para a inserção de gráficos e estatísticas. Esses dados podem ser filtrados por equipamento ou envolver todos os equipamentos do cliente. Quando for realizado o lançamento do produto, esse sistema será hospedado em nuvem, a fim de permitir o acesso dos clientes em nível nacional e internacional. Por fim, deve ser reanalisada toda a estruturação do código com o objetivo de realizar melhorias que podem contribuir com o bom funcionamento do projeto.

REFERÊNCIAS BIBLIOGRÁFICAS

BARBOSA, C. M. D. et al. AutomaÇÃO de um processo alternativo da pasteurizaÇÃO do leite utilizando aÇÃO de controle on-off. *Revista de Engenharia e Tecnologia*, UEPG, v. 10, n. 1, p. 41–52, 2018.

CILEITE. *Leite em Números - Consumo*. 2023. Disponível em: <https://www.cileite.com.br/leite_numeros_consumo>. Acesso em: Set. 2024.

EMBRAPA. *Anuário Leite 2024: avaliação genética multirracial*. 2024. Disponível em: <<https://www.infoteca.cnptia.embrapa.br/infoteca/bitstream/doc/1164754/1/Anuario-Leite-2024.pdf>>. Acesso em: Dez. 2025.

EMBRAPA. *Anuário Leite 2025: produção de leite e as mudanças climáticas*. 2025. Disponível em: <<https://www.infoteca.cnptia.embrapa.br/infoteca/bitstream/doc/1176413/1/Anuario-Leite-2025.pdf>>. Acesso em: Dez. 2025.

GROOVER, M. P. *Automation, Production Systems, and Computer-Integrated Manufacturing*. 2015. Disponível em: <https://api.pageplace.de/preview/DT0400.9781292076126_A24270953/preview-9781292076126_A24270953.pdf>. Acesso em: Dez. 2025.

MAPA. *Decreto nº 9.013, de 29 de março de 2017 (RIIS-POA)*. 2017. Disponível em: <<https://www.gov.br/agricultura/pt-br/assuntos/inspecao/produtos-animal/arquivos-publicacoes-dipoa/perguntas-e-respostas-decreto-9-013-de-2017-regulamento-de-inspecao-industrial-e-sanitaria>>. Acesso em: Dez. 2025.

MILAGRES, F. L. Proposta de automação de uma planta de produção de iogurte. Universidade Federal de Viçosa, p. 10, 2022.

MODBUS.ORG, M.-I. . *Modbus Application Protocol Specification, V1.1b3*. 2012. Disponível em: <<https://www.modbus.org/file/secure/modbusprotocolspecification.pdf>>. Acesso em: Nov. 2025.

NEPIN. *Entendendo o conceito de Pirâmide da Automação Industrial*. 2023. Disponível em: <<https://www.nepin.com.br/blog/solucoes-industriais/entendendo-o-conceito-de-piramide-da-automacao-industrial/>>. Acesso em: Nov. 2024.

ORTIZ, J. H.; MARROQUIN, W. G.; CIFUENTES, L. Z. Industry 4.0: Current status and future trends. In: *Industry 4.0 - Current Status and Future Trends*. London, UK: IntechOpen, 2020. v. 5, p. 13–28.

PROSOFT. *Introduction to Modbus TCP/IP*. 2005. Disponível em: <https://www.prosoft-technology.com/kb/assets/intro_modbustcp.pdf>. Acesso em: Nov. 2025.

ROGGIA, L.; FUENTES, R. C. *Automação Industrial*. Santa Maria, RS: e-Tec/MEC, 2016.

SILVA, G.; SILVA, A. M. A. D.; FERREIRA, M. P. de B. *Derivados do Leite*. Recife, PE: e-Tec/MEC, 2012.

SIQUEIRA, K. B. *O mercado consumidor de leite e derivados*. 2019. Disponível em: <<https://www.infoteca.cnptia.embrapa.br/infoteca/bitstream/doc/1110792/1/CT120MercadoConsumidorKenny.pdf>>. Acesso em: Dez. 2025.

WEVOLVER.COM. *Modbus TCP/IP*. 2023. Disponível em: <<https://www.wevolver.com/article/modbus-tcp-ip>>. Acesso em: Nov. 2025.