

**CENTRO UNIVERSITÁRIO PARA O DESENVOLVIMENTO DO ALTO VALE DO
ITAJAÍ - UNIDAVI**

LUCAS PORFÍRIO

**PROTÓTIPO DE SISTEMA PARA COLETA E EXIBIÇÃO DE DADOS DE
MÁQUINAS DE CORTE DE FRIOS**

**RIO DO SUL
2022**

**CENTRO UNIVERSITÁRIO PARA O DESENVOLVIMENTO DO ALTO VALE DO
ITAJAÍ - UNIDAVI**

LUCAS PORFÍRIO

**PROTÓTIPO DE SISTEMA PARA COLETA E EXIBIÇÃO DE DADOS DE
MÁQUINAS DE CORTE DE FRIOS**

Trabalho de Conclusão de Curso a ser apresentado ao curso de Sistemas da Informação, da Área das Ciências Naturais, da Computação e das Engenharias, do Centro Universitário para o Desenvolvimento do Alto Vale do Itajaí, como condição parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

Prof. Orientador: M. e. Marciel de Liz Santos

**RIO DO SUL
2022**

**CENTRO UNIVERSITÁRIO PARA O DESENVOLVIMENTO DO ALTO VALE DO
ITAJAÍ - UNIDAVI**

LUCAS PORFÍRIO

**PROTÓTIPO DE SISTEMA PARA COLETA E EXIBIÇÃO DE DADOS DE
MÁQUINAS DE CORTE DE FRIOS**

Trabalho de Conclusão de Curso a ser apresentado ao curso de Sistemas da Informação, da Área das Ciências Naturais, da Computação e das Engenharias, do Centro Universitário para o Desenvolvimento do Alto Vale do Itajaí- UNIDAVI, a ser apreciado pela Banca Examinadora, formada por:

Professor Orientador: M.e Marciel de Liz Santos

Banca Examinadora:

Prof. Fernando Andrade Bastos

Prof. Jeancarlo Visentainer

Rio do Sul, 28 de novembro de 2022.

Dedico este trabalho à meus pais Edson e Ivete, por terem se esforçado ao máximo para poder me proporcionar uma educação de qualidade.

AGRADECIMENTOS

Gostaria de agradecer ao meus pais, Edson Porfírio e Ivete de Oliveira Porfírio, por terem me proporcionado tanto amor e carinho, além de terem moldado um homem que tem em mente quais convicções deve seguir, e nunca desviar delas. Também agradecer que puderam me proporcionar um ensino de qualidade, coisa que não tiveram em sua juventude, mas não mediram esforços para me proporcionar.

Também dedico meus agradecimentos para todos os professores pelos quais tive o prazer de acompanhar as aulas no decorrer do curso, e em especial para o meu orientador e professor, Marciel de Liz Santos, ao qual me deu todo suporte para que eu construísse este trabalho.

RESUMO

Com os avanços da tecnologia, viu-se nas indústrias uma enorme necessidade de se adaptar a essas novas tecnologias. Surgiu então, a chamada indústria 4.0, que foca em integrar os equipamentos e máquinas já antes utilizados pela indústria, com um ecossistema totalmente interconectado, possibilitando assim uma visão de todo o escopo da linha de produção, auxiliando nas tomadas de decisões das empresas e melhorando o acesso as essas informações. Tendo isso em mente, observa-se que o mais importante a se fazer é encontrar uma maneira de coletar os dados que estão sendo gerados pela produção, buscando melhores informações que auxiliem na verificação e melhoramentos no desempenho da empresa. Visando todo este cenário descrito e demonstrar a possibilidade de coletar dados de equipamentos e máquinas, desenvolveu-se um protótipo de sistema simplificado que coleta dados de uma máquina fatiadora de frios e que são exibidos ao usuário em uma interface. O protótipo também oferece opção para o usuário exportar os dados gerados e assim poder efetuar técnicas de mineração de dados em ferramentas de sua preferência. Como uma máquina fatiadora de frios é um equipamento grande e caro, para a criação do protótipo e também de testes, foi desenvolvido um simulador da mesma, que é encarregado de gerar os dados e enviar os mesmos para o sistema. Para o desenvolvimento da estrutura do protótipo de sistema, foi utilizado o framework Laravel baseado na linguagem PHP, na comunicação com a máquina fatiadora foi utilizado o protocolo de conexão via socket e para a parte visual e de interface do sistema foi utilizado o framework VUE JS. Ao final deste trabalho, os objetivos foram atingidos, efetuando a coleta e a exibição dos dados ao usuário. Verificou-se a possibilidade de implementações futuras, como novas rotinas para gerenciamento das máquinas e a implementação de técnicas de projeção de dados futuros, para tornar cada vez mais o sistema completo e robusto, melhorando assim a análise dos dados por parte do usuário.

Palavras-Chave: Indústria, coleta de dados, tomadas de decisão.

ABSTRACT

With the advances in technology, an enormous need to adapt to these new technologies was perceived in the industries. Then came the so-called industry 4.0, which focuses on integrating the equipment and machines already used by the industry, with a fully interconnected ecosystem, thus enabling a view of the entire scope of the production line, helping companies to make decisions and improving access to these information. With this in mind, it is observed that the most important thing to do is to find a way to collect the data that is being generated by production, seeking better information that helps in the verification and improvements in the company's performance. Aiming at all this described scenario and demonstrating the possibility of collecting data from equipment and machines, a prototype of a simplified system was developed that collects data from a cold slicing machine and displays it to the user in an interface. The prototype also offers the option for the user to export the generated data and thus be able to perform data mining techniques in tools of their choice. As a cold slicing machine is a large and expensive equipment, for the creation of the prototype and also for tests, a simulator was developed, which is in charge of generating the data and sending them to the system. For the development of the structure of the prototype system, the Laravel framework based on the PHP language was used, the connection protocol via socket was used in the communication with the slicer machine and the VUE JS framework was used for the visual and interface part of the system. At the end of this work, the objectives were achieved, collecting and displaying the data to the user. The possibility of future implementations was verified, such as new routines for machine management and the implementation of future data projection techniques, to make the system more complete and robust, thus improving the data analysis by the user.

Keywords: Industry, machines, data collect, decision-making.

LISTA DE FIGURAS

| | |
|---|----|
| Figura 1 – Camadas modelo OSI e TCP/IP | 26 |
| Figura 2 – Posicionamento do Socket no protocolo TCP/IP | 28 |
| Figura 3 – Conexão Cliente Servidor | 28 |
| Figura 4 – Conexão Cliente Servidor | 30 |
| Figura 5 – Dashboard Engineering | 31 |
| Figura 6 - Dashboard Tractian..... | 32 |
| Figura 7 – Modelo relacional banco de dados | 37 |
| Figura 8 – Casos de uso | 38 |
| Figura 9 – Diagrama de atividade do login | 39 |
| Figura 10 – Diagrama de atividade do cadastro de usuário..... | 40 |
| Figura 11 – Diagrama de atividade do cadastro de máquinas | 41 |
| Figura 12 – Diagrama de atividade do cadastro de modelos de máquinas..... | 42 |
| Figura 13 – Diagrama de atividade do cadastro de produtos | 43 |
| Figura 14 – Diagrama de atividade do simulador da máquina de corte de frios | 44 |
| Figura 15 – Tela do simulador..... | 45 |
| Figura 16 – Fluxo básico simulador | 46 |
| Figura 17 – Tela de login..... | 47 |
| Figura 18 – Tela de cadastro..... | 48 |
| Figura 19 – Tela inicial usuário administrador..... | 49 |
| Figura 20 – Tela inicial usuário normal | 49 |
| Figura 21 – Tela de consulta de máquinas do usuário administrador | 50 |
| Figura 22 – Tela de consulta de máquinas com usuário normal | 51 |
| Figura 23 – Tela de cadastro de máquinas com usuário administrador..... | 52 |
| Figura 24 – Tela de cadastro de máquinas com usuário normal | 52 |

| | |
|---|----|
| Figura 25 – Tela de consulta de modelos de máquinas | 53 |
| Figura 26 – Tela de cadastro de modelos de máquinas | 54 |
| Figura 27 – Tela de consulta de produtos..... | 55 |
| Figura 28 – Tela de cadastro de produtos..... | 56 |
| Figura 29 – Tela de estatísticas da máquina | 57 |
| Figura 30 – Tela de estatísticas do produto | 58 |

LISTA DE QUADROS

| | |
|--|----|
| Quadro 1 – Tipos de descrições de requisitos | 17 |
| Quadro 2 – Métricas para especificar requisitos não funcionais | 18 |
| Quadro 3 – Tipos de bancos de dados | 19 |
| Quadro 4 – Estrutura de pastas Laravel | 21 |
| Quadro 5 - Padrão MVC..... | 21 |
| Quadro 6 - Lista de recursos suportados | 23 |
| Quadro 7 - Lista de recursos suportados | 32 |
| Quadro 8 - Requisitos funcionais | 35 |
| Quadro 9 - Requisitos não funcionais | 36 |

LISTA DE ABREVIATURAS E SIGLAS

| | |
|--------|--|
| AJAX | <i>Asynchronous JavaScript and XML</i> |
| ARP | <i>Address Resolution Protocol</i> |
| CSRF | <i>Cross-Site Request Forgery</i> |
| CSS | <i>Cascading Style Sheet</i> |
| CSV | <i>Comma-separated values</i> |
| DOM | <i>Document Object Model</i> |
| HTML | <i>HyperText Markup Language</i> |
| ICMP | <i>Internet Control Message Protocol</i> |
| IGMP | <i>Internet Group Management Protocol</i> |
| IoTs | <i>Internet of Things</i> |
| LAN | <i>Local Area Network</i> |
| MVC | <i>Model View Controller</i> |
| NoSQL | <i>No Standard Query Language</i> |
| OLTP | <i>Online Transaction Processing</i> |
| ORM | <i>Object-Relational Mapping</i> |
| OSI | <i>Open Systems Interconnection</i> |
| PHP | <i>Hypertext Preprocessor</i> |
| RARP | <i>Reverse Address Resolution Protocol</i> |
| ROM | <i>Read-Only Memory</i> |
| RPM | <i>Rotações por Minuto</i> |
| SCTP | <i>Stream Control Transmission Protocol</i> |
| SGBD | <i>Sistema Gerenciador de Banco de Dados</i> |
| SQL | <i>Standard Query Language</i> |
| TCP/IP | <i>Transmission Control Protocol/Internet Protocol</i> |
| UDP | <i>User Datagram Protocol</i> |
| WAN | <i>Wide Area Network</i> |
| XML | <i>Extensible Markup Language</i> |

SUMÁRIO

| | |
|---|-----------|
| 1. INTRODUÇÃO | 14 |
| 1.1 PROBLEMA DE PESQUISA | 14 |
| 1.2 OBJETIVOS | 14 |
| 1.2.1 Geral | 14 |
| 1.2.2 Específicos | 15 |
| 1.3 JUSTIFICATIVA | 15 |
| 2. REFERENCIAL TEÓRICO | 16 |
| 2.1 ENGENHARIA DE SOFTWARE | 16 |
| 2.1.1 Engenharia de Requisitos | 16 |
| 2.2 BANCO DE DADOS | 18 |
| 2.2.1 Tipos de banco de dados | 19 |
| 2.3.1 Variáveis | 20 |
| 2.3.2 Requisição, processamento e resposta | 20 |
| 2.4 LARAVEL | 20 |
| 2.4.1 Estrutura de pastas..... | 20 |
| 2.4.2 Padrão MVC (Model, View, Controller) | 21 |
| 2.4.3 Autenticação..... | 21 |
| 2.4.4 Elloquent | 22 |
| 2.5 POSTGRES | 22 |
| 2.5.1 Conformidade com padrão SQL | 22 |
| 2.6 VUE | 23 |
| 2.8.1 Estrutura | 24 |
| 2.10 PROTOCOLO TCP/IP | 25 |
| 2.10.1 Camada física e de enlace de dados | 26 |

| | |
|---|-----------|
| 2.10.2 Camada de rede | 26 |
| 2.10.3 Camada de transporte..... | 26 |
| 2.10.4 Camada de aplicativo | 27 |
| 2.11 SOCKET..... | 27 |
| 2.12 WEBSOCKET..... | 28 |
| 3. METODOLOGIA DA PESQUISA..... | 30 |
| 3.1 ESTADO DA ARTE | 31 |
| 3.1.1 Engineering | 31 |
| 3.1.2 Tractian | 31 |
| 3.1.3 Comparação protótipo com estado da arte | 32 |
| 4. RESULTADOS DO TC PROTÓTIPO DE SISTEMA PARA COLETA E EXIBIÇÃO DE DADOS DE MÁQUINAS DE CORTE DE FRIOS | 34 |
| 4.1 LEVANTAMENTO DE REQUISITOS | 34 |
| 4.1.1 Requisitos funcionais..... | 34 |
| 4.1.2 Requisitos não funcionais..... | 35 |
| 4.2 ESTRUTURA DO BANCO DE DADOS..... | 36 |
| 4.3 CASOS DE USO | 37 |
| 4.4 DIAGRAMA DE ATIVIDADE..... | 39 |
| 4.4.1 Diagrama de atividade do login..... | 39 |
| 4.4.2 Diagrama de atividade do cadastro de usuário | 40 |
| 4.4.3 Diagrama de atividade do cadastro de máquinas..... | 41 |
| 4.4.4 Diagrama de atividade do cadastro de modelos de máquinas..... | 42 |
| 4.4.4 Diagrama de atividade do cadastro de produtos | 43 |
| 4.4.4 Diagrama de atividade do simulador da máquina de corte de frios | 44 |
| 4.5 SIMULADOR | 45 |
| 4.6 TELAS DO SISTEMA..... | 46 |

| | |
|--|-----------|
| 4.6.1 Tela de login | 46 |
| 4.6.2 Tela de cadastro de usuário | 47 |
| 4.6.3 Tela inicial | 48 |
| 4.6.4 Tela de consulta de máquinas..... | 50 |
| 4.6.5 Tela de cadastro de máquinas | 51 |
| 4.6.6 Tela de consulta de modelos de máquinas..... | 53 |
| 4.6.7 Tela de cadastro de modelos de máquinas | 53 |
| 4.6.8 Tela de consulta de produtos | 54 |
| 4.6.9 Tela de cadastro de produtos..... | 55 |
| 4.6.10 Tela de estatísticas da máquina..... | 56 |
| 4.6.11 Tela de estatísticas do produto | 57 |
| 5. CONCLUSÃO..... | 59 |
| 5.1 SUGESTÕES DE MELHORIA FUTURAS | 59 |
| REFERÊNCIAS | 61 |

1. INTRODUÇÃO

O principal objetivo de qualquer empresa é produzir mais com menos recursos, e cada vez que se aprofunda neste assunto, surge uma nova forma de trabalhar, mais eficiente e moderna, que não só aumenta a produtividade como também a qualidade final do produto. Uma das mais recentes revoluções na indústria, foi a indústria 4.0, alterando a forma de como ela trabalha.

Na indústria 4.0 o foco está em obter um ambiente altamente conectado, fazendo com que muitos processos se automatizem e tomem decisões com base nos demais processos que estão sendo executados em simultâneo. Algumas áreas da indústria estão mais avançadas quando o quesito é coletar e gerenciar informações, entretanto outras ainda não iniciaram sua transformação para esta nova era. O primeiro passo a se dar em direção a interconectividade é a possibilidade de coletar os dados gerados pelos equipamentos da fábrica.

Com base nessa premissa, foi escolhida a área da indústria que trabalha com máquinas fatiadoras de frios, a fim de demonstrar como pode ser implementado a coleta e a exibição dos dados produzidos por este tipo de máquina. Antes estes dados estavam somente disponíveis via terminal e de forma desorganizada, impedindo um entendimento concreto dos mesmos.

Dessa forma, apresenta-se nesse trabalho um protótipo de sistema para gerenciamento e coleta, exibição e exportação de dados de máquinas de cortes de frios. Para a demonstração, criou-se um simulador de máquinas fatiadoras, responsável por simular a máquina e produzir os dados para o sistema. O sistema então, disponibiliza os dados coletados de forma gráfica e que podem ser exportados, dando ao usuário maior informação sobre o processo de produção, além de utilizar esses mesmos dados em diversas ferramentas de refinamento.

1.1 PROBLEMA DE PESQUISA

É possível efetuar a coleta e exibição de dados de máquinas de fatiamentos de frios?

1.2 OBJETIVOS

1.2.1 Geral

- Criar protótipo de sistema para coleta e exibição de dados de máquinas de corte de frios.

1.2.2 Específicos

- Levantar requisitos do sistema;
- Desenvolver o simulador da máquina de frios;
- Criar comunicação via socket;
- Desenvolver backend e frontend do sistema.
- Possibilitar a exportação dos dados coletados.

1.3 JUSTIFICATIVA

Para estar à frente no mercado, é necessário ter tomadas de decisões assertivas e estratégicas, de modo que se possa projetar produção, vendas e lucros meses à frente do dia atual, podendo assim criar planos específicos para situações específicas, estando preparado para diversos imprevistos. Com o intuito de tornar essas tomadas de decisões possíveis, são necessários uma grande quantidade dados precisos e, com base nisso, pode-se dizer que os dados são um dos ativos mais importantes de uma empresa.

As indústrias no geral, começaram somente recentemente a perceber a importância dos dados produzidos na área de produção para o seu crescimento, portanto estão mais do que nunca buscando soluções que possam satisfazer estes critérios, porém nem todas as soluções disponíveis no mercado dão a possibilidade de fazer essa coleta de dados das máquinas de forma automática, muitas das soluções, acabam por armazenar os dados localmente, sendo necessário coletar estes dados manualmente.

Hoje, com a ascensão da indústria 4.0 é possível integrar vários equipamentos em um sistema, obtendo os mais diversos dados em tempo real. Um ramo que tem grande volume de produção e grande parte automatizada é a indústria alimentícia. Todos os dias são processadas toneladas de alimentos e para obter um controle preciso da produção, somente aplicando técnicas da ciência de dados para esse controle.

Com base nessas informações e tendo como foco inicialmente modelos de máquinas fatiadoras de frios, justifica-se a criação de um sistema que efetue a comunicação com as máquinas industriais e possibilite não só a coleta, mas a visualização dos dados e sua exportação.

2. REFERENCIAL TEÓRICO

Os presentes tópicos foram baseados nas tecnologias e conceitos que serão utilizadas para o desenvolvimento do protótipo de sistema para coleta e exibição de dados de máquinas de corte de frios.

2.1 ENGENHARIA DE SOFTWARE

“A engenharia de software abrange um processo, um conjunto de métodos (práticas) e um leque de ferramentas que possibilitam aos profissionais desenvolverem software de altíssima qualidade.” (PRESSMAN; MAXIM, 2016, p.14).

É com a engenharia de software que se torna possível o desenvolvimento de sistemas complexos dentro do prazo e com qualidade. Ela organiza um trabalho para que não se torne caótico, mas também dá a possibilidade de abordar o software conforme as necessidades para o desenvolvimento. (PRESSMAN; MAXIM, 2016).

2.1.1 Engenharia de Requisitos

Requisitos de um sistema são descrições dos serviços que ele deve conter e restrições de sua operação. Os requisitos irão ter como base as necessidades dos clientes que irão utilizar o sistema. Engenharia de requisitos então é o processo de descoberta, análise, documentação, e conferências e restrições dessas necessidades do cliente. O requisito levantado pode ser uma descrição mais sucinta e superficial do serviço que o sistema deve oferecer, ou também uma descrição detalhada do serviço a ser oferecido. (SOMMERVILLE, 2019).

Para Sommerille (2019), deve-se separar as descrições dos requisitos em dois tipos, requisitos para usuários, que tem uma descrição mais superficial e geral do serviço, e requisitos de sistema, que contém uma descrição mais detalhado do que o sistema irá fazer. Segue abaixo Quadro 1 explicando sobre os dois tipos de descrição de requisitos.

Quadro 1 – Tipos de descrições de requisitos

| | |
|-----------------------------|---|
| Requisito de usuário | Requisitos de usuário são declarações, em uma linguagem natural somada a diagramas, dos serviços que se espera que o sistema forneça para os usuários e das limitações sob as quais ele deve operar. Esses requisitos podem variar de declarações amplas das características necessárias do sistema até descrições precisas e detalhadas da sua funcionalidade. |
| Requisito de sistema | Os requisitos de sistema são descrições mais detalhadas das funções, dos serviços e das restrições operacionais do sistema de software. O documento de requisitos de sistema (chamado às vezes de especificação funcional) deve definir exatamente o que deve ser implementado. Pode fazer parte do contrato entre o adquirente do sistema e os desenvolvedores de software |

Fonte: Elaborado a partir de Sommerville (2019).

Os requisitos podem ser classificados em requisitos funcionais e requisitos não funcionais. Os requisitos funcionais descrevem como o sistema deve reagir a determinadas ações e situações, já os requisitos não funcionais são descrições de restrições que o sistema irá apresentar, e que se aplicam constantemente de forma geral no sistema, em vez de em pontos específicos como é feito nos requisitos funcionais. (SOMMERVILLE, 2019).

2.1.1.1 Requisitos funcionais

Requisitos funcionais tem como objetivo descrever o que o sistema deve fazer, e estão ligados ao tipo de software que está sendo desenvolvido, dos usuários, e do método utilizado para organizar e escrever os requisitos. Devem detalhar as funções do sistemas, entradas, saídas e exceções, podendo variar entre descrições mais gerais do que o sistema deve fazer ou então descrições mais específicas. (SOMMERVILLE, 2019).

As especificações de um sistema deve ser completa, tendo todos os serviços e informações definidos e coerentes, porém somente se é possível alcançar essa coerência em sistemas menores, pois existem menos chances de serem cometidos erros pelo fato do escopo ser menor, coisa que dificulta em um grande sistema, justamente pela quantidade de requisitos. (SOMMERVILLE, 2019).

2.1.1.2 Requisitos não funcionais

Requisitos não funcionais não possuem relação direta com os serviços específicos fornecidos aos usuários, geralmente especificam ou restringem as características do sistema de maneira geral. Podem estar relacionados com propriedades do sistema, como confiabilidade, tempo de resposta e uso de memória, também podem ser restrições ao sistema como capacidades dos dispositivos de entrada e saída. (SOMMERVILLE, 2019).

Os requisitos não funcionais surgem das necessidades dos usuários, que se devem a restrições orçamentárias, políticas organizacionais, necessidade de interoperabilidade com outros sistemas de software ou hardware, ou fatores externos, como normas de segurança (safety) ou legislação relativa à privacidade. (SOMMERVILLE, 2019, p.91).

Segundo Sommerville (2019, p.94), “Sempre que possível, os requisitos não funcionais devem ser escritos de forma quantitativa para que possam ser testados objetivamente.”. Segue Quadro 2 com as métricas para especificar as propriedades não funcionais do sistema.

Quadro 2 – Métricas para especificar requisitos não funcionais

| | |
|--------------------------|---|
| Velocidade | Transações processadas/segundo; Tempo de resposta do usuário/evento; Tempo de atualização da tela. |
| Tamanho | Megabytes/número de chips de ROM |
| Facilidade de uso | Tempo de treinamento; Número de quadros de ajudas. |
| Confiabilidade | Tempo médio até a falha; Probabilidades de indisponibilidade; Taxa de ocorrência de falhas; Disponibilidade. |
| Robustez | Tempo para reiniciar após a falha; Porcentagem de eventos causando falhas; Probabilidade de corromper dados em uma falha. |
| Portabilidade | Porcentagem de declarações dependentes do sistema-alvo; Número de sistemas-alvo. |

Fonte: Elaborado a partir de Sommerville (2019).

2.2 BANCO DE DADOS

Conforme Teorey et al. (2014), a menor unidade de dados identificável que tem um significado no mundo real é o item de dados, e ao agrupar e relacionar esses itens de dados tem-se um registro, e ao obter uma coleção de registros do mesmo tipo se obtém um arquivo. Nos sistemas de banco de dados, os itens de dados são chamados de coluna ou atributo, os registros são chamados de linha ou tupla, e os arquivos são chamados de tabelas.

Com base nessas premissas define-se que um banco de dados é uma coleção de dados armazenados e relacionados, para que possam atender as necessidades de vários usuários. Essas coleções de dados podem ser representadas em tabelas.

Para Teorey et al. (2014), um sistema gerenciador de banco de dados (SGBD) é um sistema de software que tem como objetivo manipular bancos de dados, e deve possuir uma visão lógica (esquema, subesquema), visão física (métodos de acesso, clustering de dados), linguagem de definição de dados, linguagem de manipulação de dados e ferramentas como gerenciamento de transação, controle de concorrência, integridade de dados, recuperação de falhas e segurança.

2.2.1 Tipos de banco de dados

Conforme Oracle (2022), existem muitos tipos de bancos de dados, e o melhor banco de dados depende de como os dados serão utilizados. Segue abaixo quadro 3 com os tipos de bancos de dados.

Quadro 3 – Tipos de bancos de dados

| | |
|---|---|
| Bancos de dados relacionais | Os bancos de dados relacionais se tornaram dominantes na década de 1980. Os itens em um banco de dados relacional são organizados como um conjunto de tabelas com colunas e linhas. A tecnologia de banco de dados relacional fornece a maneira mais eficiente e flexível de acessar informações estruturadas. |
| Bancos de dados orientados a objetos | As informações em um banco de dados orientado a objetos são representadas na forma de objetos, como na programação orientada a objetos. |
| Bancos de dados distribuídos | Um banco de dados distribuído consiste em dois ou mais arquivos localizados em sites diferentes. O banco de dados pode ser armazenado em vários computadores, localizados no mesmo local físico ou espalhados por diferentes redes. |
| Data warehouses | Um repositório central de dados, um data warehouse é um tipo de banco de dados projetado especificamente para consultas e análises rápidas. |
| Bancos de Dados NoSQL | Um NoSQL, ou banco de dados não relacional, permite que dados não estruturados e semiestruturados sejam armazenados e manipulados (em contraste com um banco de dados relacional, que define como todos os dados inseridos no banco de dados devem ser compostos). Os bancos de dados NoSQL se tornaram populares à medida que os aplicativos web se tornaram mais comuns e mais complexos. |
| Bancos de dados gráficos | Um banco de dados gráfico armazena dados em termos de entidades e os relacionamentos entre entidades. |
| Bancos de dados OLTP | Um banco de dados OLTP é um banco de dados rápido e analítico projetado para um grande número de transações realizadas por vários usuários. |

Fonte: Elaborado a partir de Oracle (2022).

2.3 PHP

Para Bento (2016), o PHP possibilita pré-processar páginas HTML, podendo assim alterar os comportamentos das páginas antes de envia-las para o navegador. Também permite capturar dados do usuário com formulários ou de outras maneiras.

2.3.1 Variáveis

Para definir uma variável no PHP é necessário adiciona um \$, seguido por alguma sequência de caracteres, contanto que seja iniciado por alguma letra do alfabeto ou um underline. Outra característica das variáveis do PHP é que ela são case sensitive, ou seja, existe diferença entre letras maiúsculas ou minúsculas, por exemplo, \$Ontem é diferente de \$ONTEM. (LOBO, 2007).

2.3.2 Requisição, processamento e resposta

O PHP é uma linguagem de server-side, ou seja, seu processamento é feito no lado do servidor, retornando então somente o código HTML para o lado do cliente. Isto possibilita que retorne para o cliente somente o que o programador possibilitar, em o conteúdo restrito aos usuários fica gravado no banco de dados. (LOBO, 2007).

2.4 LARAVEL

Laravel é um framework de aplicação web, sendo um framework Web uma estrutura que fornece um ponto de partida para criar seu aplicativo, permitindo que você se concentre em desenvolver. (LARAVEL, 2022).

Segundo Laravel (2022), graças à natureza de fácil dimensionamento do PHP e ao suporte integrado do Laravel para sistemas de cache rápidos e distribuídos, o Laravel se torna incrivelmente escalável. Os aplicativos Laravel foram dimensionados para lidar facilmente com centenas de milhões de requisições por mês.

2.4.1 Estrutura de pastas

A estrutura padrão da aplicação Laravel destina-se a fornecer um ótimo ponto de partida para aplicativos grandes e pequenos. Mas o desenvolvedor é livre para organizar sua aplicação como desejar. O Laravel não impõe quase nenhuma restrição sobre onde qualquer classe está localizada, contanto que o Composer possa carregar automaticamente a classe. (LARAVEL, 2022).

Turini (2015) menciona quatro pastas como as principais da estrutura do Laravel: App, config, public, vendor, conforme apresentado no Quadro 4.

Quadro 4 – Estrutura de pastas Laravel

| | |
|---------------------|--|
| Pasta App | Nela ficam seus modelos, views e controllers, que serão bem detalhados no próximo capítulo. Em poucas palavras, é onde boa parte do seu código vai ficar. Ela possui uma série de subdiretórios, como Commands, Console, Http, Events, entre outros. |
| Pasta Config | Como o nome já indica, é onde ficam os arquivos de configuração do seu projeto. Se você precisar alterar as configurações de cache, e-mail, banco de dados, entre outras, já sabe onde encontrar. |
| Pasta Public | É a pasta pra onde seu web server vai apontar. Lá fica o arquivo index.php, que aponta para sua aplicação. Além disso, é comum colocarmos os arquivos css, imagens, javascript e todos os demais arquivos públicos nesse diretório. |
| Pasta Vendor | É onde fica o source code do Laravel, plugins e outras dependências. Tudo que você usar de terceiros (bibliotecas, frameworks etc.) deve ficar nela. |

Fonte: Elaborado a partir de Turini (2015).

2.4.2 Padrão MVC (Model, View, Controller)

Para Turini (2015, p. 20), “A grande ideia desse padrão arquitetural é que você separe suas regras de negócio em 3 camadas, cada uma com sua responsabilidade muito bem definida.”, conforme apresentado no Quadro 5.

Quadro 5 - Padrão MVC

| | |
|-------------------|--|
| Model | Model é a camada onde ficam nossas regras de negócio, nossas entidades e classes de acesso ao banco de dados. |
| View | View é a responsável por apresentar as páginas e outros tipos de resultado para o usuário (ou mesmo para outros sistemas, que se comunicam). É a resposta que o framework envia para o navegador, que normalmente é um HTML. |
| Controller | Controller é quem cuida de receber as requisições web e decidir o que fazer com elas. Nessa camada definimos quais modelos devem ser executados para determinada ação e para qual view vamos encaminhar a resposta. Em outras palavras, essa camada quem faz o link entre todas as outras. |

Fonte: Elaborado a partir de Turini (2015).

2.4.3 Autenticação

As instalações de autenticação do Laravel são compostas por "guardas" e "provedores". Os guardas definem como os usuários são autenticados para cada solicitação, já os provedores definem como os usuários são recuperados de seu armazenamento. (LARAVEL, 2022).

2.4.4 Eloquent

O Laravel inclui o Eloquent, um mapeador relacional de objetos(ORM), que facilita interagir com o banco de dados. Utilizando o Eloquent, cada tabela do banco de dados possui um "Modelo" correspondente, que é usado para interagir com essa tabela. Além de buscar registros da tabela do banco de dados, os modelos do Eloquent também permitem inserir, atualizar e excluir registros da tabela. (LARAVEL, 2022).

2.5 POSTGRES

O PostgreSQL é um poderoso sistema de banco de dados relacional de objeto de código aberto que usa e tem como base a linguagem SQL, combinada com muitos recursos que armazenam e dimensionam com segurança as cargas de trabalho de dados mais complicadas. As origens do PostgreSQL remontam a 1986 como parte do projeto POSTGRES na Universidade da Califórnia em Berkeley e tem mais de 30 anos de desenvolvimento ativo na plataforma principal. (POSTGRESQL, 2022).

2.5.1 Conformidade com padrão SQL

Conforme (POSTGRESQL, 2022), o PostgreSQL está em conformidade com pelo menos 170 dos 179 recursos obrigatórios para conformidade com SQL:2016 Core. Até o momento, nenhum banco de dados relacional atende totalmente a esse padrão. Os recursos suportados pelo PostgreSQL são apresentados no Quadro 6.

Quadro 6 - Lista de recursos suportados

| | |
|---|--|
| Tipos de dados | Primitivos: Inteiro, Numérico, String, Booleano; Estruturado: Data/Hora, Matriz, Intervalo/Multirange, UUID; Documento: JSON/JSONB, XML, valor-chave (Hstore); Geometria: Ponto, Linha, Círculo, Polígono; Personalizações: Composto, Tipos personalizados. |
| Integridade de dados | Único, não nulo; Chaves Primárias; Chaves estrangeiras; Restrições de exclusão; Bloqueios Explícitos, Bloqueios Consultivos. |
| Simultaneidade, Desempenho | Indexação: B-tree, Multicoluna, Expressões, Parcial; Indexação avançada: GiST, SP-Gist, KNN Gist, GIN, BRIN, índices de cobertura, filtros Bloom; Planejador/otimizador de consultas sofisticado, varreduras somente de índice, estatísticas de várias colunas; Transações, transações aninhadas (via pontos de salvamento); Controle de simultaneidade de várias versões (MVCC); Paralelização de consultas de leitura e construção de índices de árvore B; Particionamento de tabela; Todos os níveis de isolamento de transação definidos no padrão SQL, incluindo Serializable; Compilação just-in-time (JIT) de expressões. |
| Confiabilidade, recuperação de desastres | Registro de gravação antecipada (WAL); Replicação: Assíncrona, Síncrona, Lógica; Recuperação pontual (PITR), esperas ativas; Espaços de tabela |
| Segurança | Autenticação: GSSAPI, SSPI, LDAP, SCRAM-SHA-256, Certificado e mais; Sistema robusto de controle de acesso; Segurança em nível de coluna e linha; Autenticação multifator com certificados e um método adicional. |
| Extensibilidade | Funções e procedimentos armazenados; Linguagens de procedimento: PL/PGSQL, Perl, Python (e muito mais); Expressões de caminho SQL/JSON; Wrappers de dados estrangeiros: conecte-se a outros bancos de dados ou fluxos com uma interface SQL padrão; Interface de armazenamento personalizável para tabelas; Muitas extensões que fornecem funcionalidades adicionais, incluindo PostGIS. |
| Internacionalização, Pesquisa de Texto | Suporte para conjuntos de caracteres internacionais, por exemplo, por meio de agrupamentos ICU; Agrupamentos que não diferenciam maiúsculas de minúsculas e não diferenciam acentos; Pesquisa de texto completo. |

Fonte: Elaborado a partir de Postgresql (2022).

2.6 VUE

Vue é um framework utilizado para construir interfaces de usuário de forma progressiva, de modo que se possa ser adotado de forma incremental, diferente de frameworks monolíticos. Sua biblioteca tem foco na parte visual, e dá a possibilidade de trabalho em conjunto com outras bibliotecas. (VUE.JS, 2022).

O sistema de componentes também é outro importante conceito no Vue, por ser uma abstração que proporciona a construção de aplicações de larga escala compostas por pequenos componentes, auto-contidos e frequentemente reutilizáveis. Se nós pensarmos sobre isso, quase qualquer tipo de interface de uma aplicação pode ser abstraída em uma árvore de componentes. (VUE.JS, 2022, n. p.).

“Uma característica do JavaScript, e por consequência também do Vue.js, é a possibilidade de ser utilizado em qualquer projeto que possua front-end. Independente da linguagem de programação escolhida, é possível usar o Vue.js.” (INCAU, 2017, p. 2).

2.7 CSS3

Segundo Caelum ([20--] século certo), o CSS é uma linguagem separada do HTML, e tem como objetivo cuidar da estilização da página. Tem como vantagem em relação ao HTML, sua robustez para estilização, além de manter distintas as regras de estilo do conteúdo HTML.

2.8 HTML5

Conforme Limongi (2014), HTML é basicamente uma linguagem de hipertexto projetada para mostrar páginas Web nos navegadores.

A Hipertext Markup Language ou HTML, se trata de uma linguagem para a manipulação de hipertextos baseada em marcas. Basicamente é um conjunto de comandos que se intercalam dentro de um texto para cumprir certas funções de formatação, comando e relações. Adicionalmente serve para incorporar outros tipos de conteúdos como áudio, vídeo e imagens. (LIMONGI, 2014, p.17).

A grande vantagem do HTML, é que todas as páginas da internet são construídas com base nessa linguagem. (LIMONGI, 2014).

2.8.1 Estrutura

Os documentos HTML consistem em uma árvore de elementos e texto. Cada elemento é indicado na origem por uma tag inicial, como "<body>", e uma tag final, como "</body>". (DEV.W3C, 2011).

Segundo DEV.W3C(2011), os elementos podem conter atributos que controlam como os elementos vão se comportar. Esses atributos são colocados dentro da tag inicial e consistem em um nome e um valor, separados pelo caractere "=" . O valor do atributo pode permanecer sem aspas se não contiver espaços ou qualquer um dos " ' ` = < ou > . Caso contrário, ele deve utilizar aspas simples ou duplas. Se o atributo recebe uma string vazia, não necessita do caractere "=", nem mesmo do valor.

DEV.W3C(2011) descreve que, compiladores de HTML utilizam as tags e as transformam em uma árvore DOM (*Document Object Model*). Uma árvore DOM é uma representação na memória de um documento. Essa árvore DOM pode ser manipulada a partir de scripts na página. Scripts (normalmente em JavaScript) são pequenos programas que podem ser incorporados usando a tag script ou usando atributos de conteúdo do manipulador de eventos.

2.9 JAVASCRIPT

Para W3SCHOOLS (2016), um script é um código de programa que não precisa de pré-processamento antes de ser executado. Em navegadores da Web, o script geralmente se refere ao código do programa escrito em JavaScript que é executado pelo navegador quando uma página é baixada ou em resposta a um evento acionado pelo usuário.

Conforme W3SCHOOLS (2016), os scripts podem tornar as páginas Web mais dinâmicas. Permite atualizar uma nova versão da página sem recarregar todo o conteúdo da mesma, método esse chamado de DHTML (HTML Dinâmico), e também pode adicionar ou enviar conteúdo da página via requisições para o servidor, chamado de AJAX (JavaScript e XML assíncrono).

A interface de script mais básica desenvolvida é o DOM (*Document Object Model*), que permite que programas e scripts acessem e atualizem dinamicamente o conteúdo, a estrutura e o estilo dos documentos. (W3SCHOOLS, 2016).

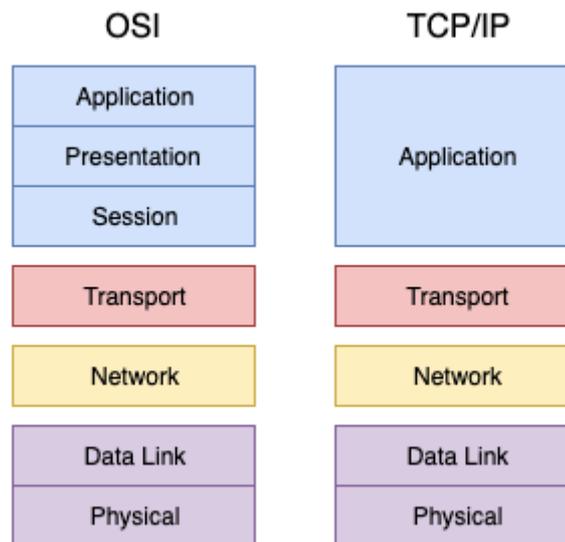
2.10 PROTOCOLO TCP/IP

De acordo com Tedesco (2019), uma rede de computadores funciona sob protocolos, e o protocolo mais conhecido e utilizado é o TCP/IP, este que faz uma junção dos protocolos IP (da camada de rede), TCP e UDP (da camada de transporte), HTTP (da camada de aplicação) entre outros.

O conjunto de protocolos TCP/IP foi desenvolvido antes do modelo OSI. Portanto, as camadas no conjunto de protocolos TCP/IP não correspondem exatamente às que aparecem no modelo OSI. O conjunto de protocolos TCP/IP é constituído de cinco camadas: física, enlace de dados, rede, transporte e aplicativo. As quatro primeiras camadas fornecem padrões físicos, interface de rede, interconexão em rede e funções de transporte que correspondem às quatro primeiras camadas do modelo OSI. Entretanto, as três camadas superiores do modelo OSI são representadas no TCP/IP por uma única, chamada de camada de aplicativo. (FOROUZAN, 2010, p.30).

Para Tedesco (2020, n.p.), “O modelo OSI possui sete camadas enquanto o TCP/IP é dividido em cinco.”. A Figura 1 exemplifica a equivalência do modelo TCP/IP para o modelo OSI.

Figura 1 – Camadas modelo OSI e TCP/IP



Fonte: Elaborado a partir de Tedesco (2019).

2.10.1 Camada física e de enlace de dados

“Nas camadas física e de enlace de dados, o TCP/IP não define nenhum protocolo específico. Ele suporta todos os protocolos-padrão e patenteados. Na interconexão em rede TCP/IP, uma rede pode ser local (LAN) ou remota (WAN).”. (FOROUZAN, 2010, p.31).

2.10.2 Camada de rede

“Na camada de rede (ou, mais precisamente, a camada de interconexão em rede), o TCP/IP suporta o IP (*Internetworking Protocol*). Por sua vez, o IP usa quatro protocolos de apoio: ARP, RARP, ICMP e IGMP.”. (FOROUZAN, 2010, p.31).

2.10.3 Camada de transporte

Segundo Forouzan (2010), a camada de transporte é representada pelos protocolos TCP, UDP e SCTP, que são responsáveis pelo envio de uma mensagem de um processo para outro programa.

O mais simples dos dois protocolos de transporte do TCP/IP é o UDP (*User Datagram Protocol*), sendo um protocolo de processo a processo que adiciona apenas endereços de porta,

controle de erros de soma de verificação e informações de comprimento nos dados da camada superior. (FOROUZAN, 2010).

O TCP (*Transmission Control Protocol*) é um protocolo de transporte de fluxo confiável, sendo fluxo, orientado a conexões, ou seja, se faz necessário que haja uma conexão estabelecida entre as duas partes da transmissão, para que então haja o envio dos dados. No envio de cada transmissão, é feito uma divisão do fluxo de dados em unidades menores chamada de segmentos. (FOROUZAN, 2010).

Cada segmento inclui uma sequência numérica, para reordenação depois de recebidos, junto a um número de reconhecimento para os segmentos recebidos. Os segmentos são transportados na Internet dentro de datagramas IP. Na extremidade receptora, o TCP reúne cada datagrama à medida que chega e reordena a transmissão com base na sequência numérica. (FOROUZAN, 2010, p.32).

“O novo SCTP (*Stream Control Transmission Protocol*) fornece suporte para novos aplicativos, como telefonia de IP. Trata-se de um protocolo de camada de transporte que combina os bons recursos do UDP e do TCP.”. (FOROUZAN, 2010, p.32).

2.10.4 Camada de aplicativo

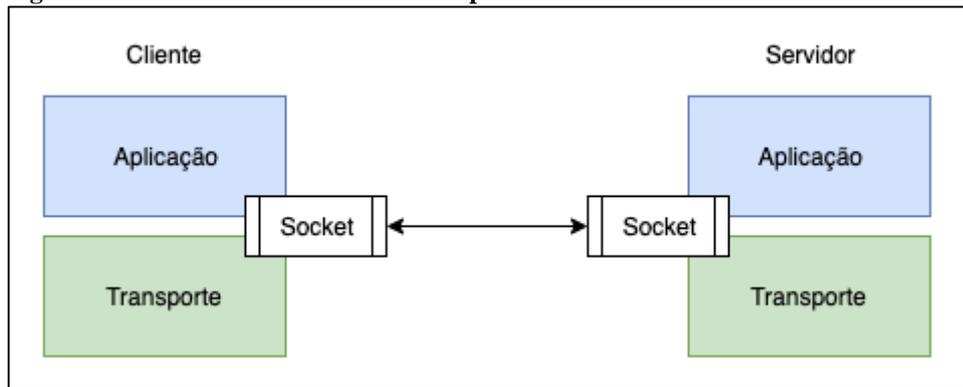
“A camada de aplicativo no TCP/IP é equivalente às camadas de sessão, apresentação e de aplicativo, combinadas do modelo OSI. Muitos protocolos são definidos nessa camada.”. (FOROUZAN, 2010, p.32).

2.11 SOCKET

Para Tedesco (2019), um Socket fornece a comunicação entre duas pontas (fonte e destino), entre dois processos que estejam ou na mesma máquina (Unix Socket) ou na mesma rede (TCP/IP Socket). Estando na rede, um socket é representado por ip:porta, por exemplo: 127.0.0.1:4477 (IPv4).

Tedesco (2019), afirma que considerando o modelo de rede OSI, no protocolo TCP/IP, os sockets de rede estão alocados entre a camada de aplicação e a de transporte. A figura 2 exemplifica o posicionamento do sockets nas camadas.

Figura 2 – Posicionamento do Socket no protocolo TCP/IP



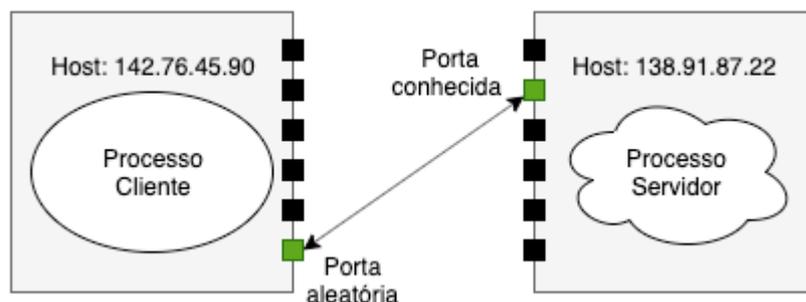
Fonte: Elaborado a partir de Tedesco (2019).

Ainda para Tedesco (2019), os Sockets são responsáveis pela abstração da comunicação entre as partes, dando a sensação de que a comunicação está sendo direta, porém ela passa pelas camadas de rede antes de chegar ao seu destino.

“Todo cliente deve conhecer o socket do servidor (conjunto ip e porta) para se comunicar, mas o servidor só vai conhecer o socket do cliente quando este realizar uma conexão com ele, ou seja, a conexão no modelo cliente-servidor é sempre iniciada pelo cliente.” (TEDESCO, 2019, n.p.).

É possível verificar na Figura 3, um exemplo da comunicação entre o cliente e o servidor, onde “...a porta do servidor precisa ser previamente conhecida pelo cliente, enquanto que pro servidor não importa qual é a porta do cliente, ele vai conhecê-la quando a conexão dele com o cliente for estabelecida.”. (TEDESCO, 2019, n.p.).

Figura 3 – Conexão Cliente Servidor



Fonte: Elaborado a partir de Tedesco (2019).

2.12 WEBSOCKET

Tedesco (2020) afirma que, WebSocket é um tipo de protocolo que permite a comunicação bidirecional e simultânea entre cliente e servidor, diferenciando-o do protocolo Http, que tem sua transmissão de forma unidirecional, onde o cliente deve enviar uma

requisição, e o servidor retornar uma resposta, encerrando então a conexão. Encontra-se então uma vantagem nas conexões de WebSocket em relação ao Http, em aplicações que necessitem de grande volume de dados trafegando simultaneamente, pois com uma única conexão aberta é possível efetuar todos os envios e recebimentos de dados, até que alguma das partes encerre a conexão. Deve-se ter em mente que um WebSocket é diferente de um Socket, pois o WebSocket é um protocolo que roda em cima de sockets TCP, já o socket é a comunicação entre duas portas em uma rede.

3. METODOLOGIA DA PESQUISA

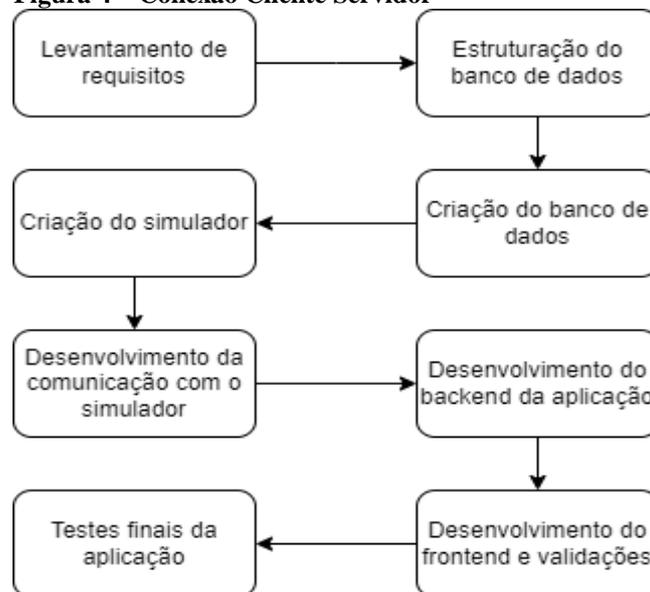
O trabalho apresentado caracteriza-se como pesquisa aplicada e descritiva, pois tem como objetivo desenvolver um protótipo de aplicação para poder gerenciar dados de máquinas industriais de cortes de frios via socket. Para responder a seguinte pergunta: É possível efetuar a coleta e exibição de dados de máquinas de fatiamentos de frios?

Para realização do trabalho fora desenvolvido um simulador de máquinas de cortes de frios, onde o mesmo gera dados condizentes com os que seriam gerados em uma máquina física.

Esta máquina virtual, assim como uma física, se conecta com o protótipo de sistema via socket de rede, que é amplamente utilizado para comunicação de dispositivos tecnológicos, como exemplo disto se tem os IoTs (*Internet of Things*), que utilizam da comunicação de dispositivos com sistemas a fim de coletar dados.

A aplicação tem como foco efetuar a comunicação e coletar os dados das máquinas, para que posteriormente possam ser exportados estes dados, a fim de que seja possível utiliza-los em ferramentas de refinamento e mineração de dados, ajudando na tomada de decisões da empresa que possuir a aplicação. Os processos de desenvolvimento que foram realizados para criar o projeto estão descritos na Figura 4.

Figura 4 – Conexão Cliente Servidor



Fonte: Acervo do autor (2022).

3.1 ESTADO DA ARTE

No estado da arte será apresentado possíveis soluções já existentes no mercado para lidar com a coleta de dados que mais se assemelham com o projeto desenvolvido.

3.1.1 Engineering

O sistema da Engineering é uma das alternativas do mercado para a coleta de dados de máquinas. Cobre uma ampla variedade de tipos de protocolos de rede, o que facilita a possibilidade de trabalhar com variadas máquinas. Possui rotinas de exibição de dados, bem como relatórios e análise dos dados. Estão no Brasil há mais de 13 anos, sendo atualmente uma das mil maiores empresas do Brasil. Seu foco geralmente se dá para indústrias maiores, não abrangendo pequenos produtores. Na Figura 5 é possível verificar uma das telas de visualização de dados da Engineering.

Figura 5 – Dashboard Engineering



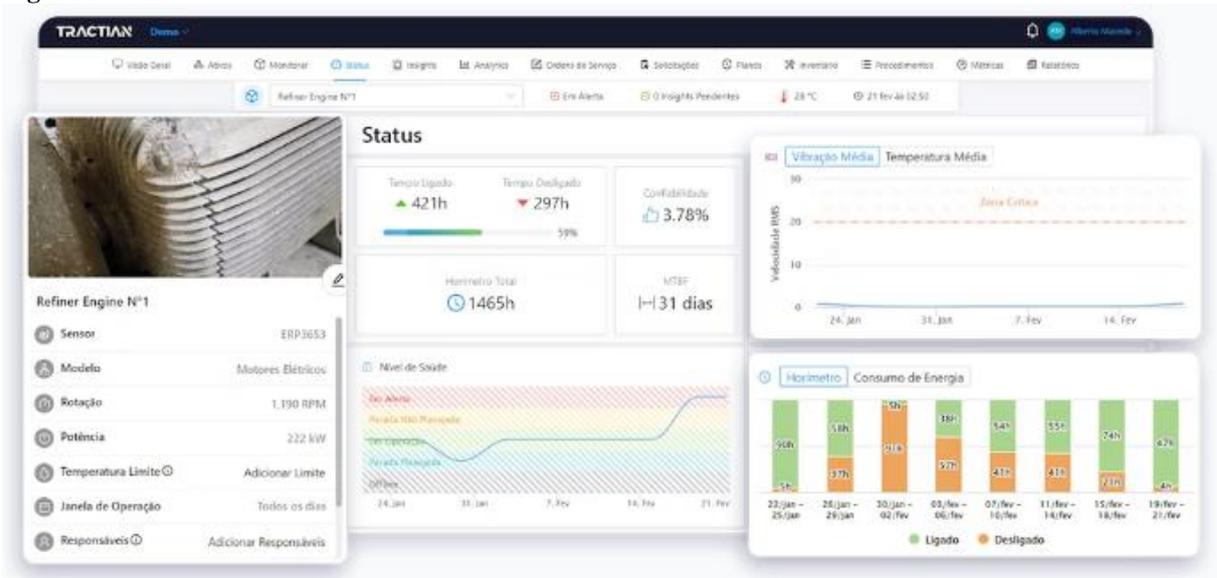
Fonte: Elaborado a partir de Engineering ([20--]).

3.1.2 Tractian

O Tractian é um sistema focado no gerenciamento das máquinas industriais de modo a avaliar seus status. Pode assim agendar manutenções preventivas, ou mesmo então uma manutenção corretiva com base nos dados colhidos da máquina. Os dados coletados são

colhidos por sensores que devem ser instalados nas máquinas. Estes sensores terão informações como vibração, temperatura, dentre outras, que facilitam avaliar o estado da máquina. Porém, este sistema não efetua a coleta da produção da máquina. Na Figura 6 é possível visualizar um painel de gestão de dados da Tractian.

Figura 6 - Dashboard Tractian



Fonte: Elaborado a partir de Tractian([20--]).

3.1.3 Comparação protótipo com estado da arte

No Quadro 7 são feitas algumas comparações das funcionalidades entre o protótipo desenvolvido e os outros sistemas no mercado listados anteriormente.

Quadro 7 - Lista de recursos suportados

| Funcionalidade | Protótipo | Engineering | Tractian |
|--|-----------|-------------|----------|
| Coleta de dados da produção | X | X | |
| Coleta de dados para manutenção e prevenção das máquinas | | X | X |
| Gráficos em tempo real | X | X | X |
| Específico para máquinas fatiadoras de frios | X | | |
| Cadastro de máquinas | X | X | X |
| Precificação dos produtos produzidos | X | | |
| Exportação de dados | X | | X |

Fonte: Acervo do autor (2022).

É possível notar que os demais sistemas possuem funcionalidades em comum ao protótipo, porém por se tratar de um sistema focado especificamente para máquinas fatiadoras de frios, o protótipo acaba por ser mais completo quando seu objetivo for trabalhar com estas máquinas, principalmente quando o objetivo for a coleta de dados da produção, pois além de

garantir a coleta, é possível vincular preços aos produtos que estão sendo fatiados pelas máquinas, enriquecendo assim a base dos dados coletados, possibilitando análises mais amplas pelos usuários.

O único tópico que não é atendido é a coleta de dados para manutenção e prevenção, pois este é um recurso que para ser implementado necessitaria de sensores específicos para isso implantados nas máquinas, portanto sairia um pouco da área das fatiadoras em si e entraria para uma área mais ampla de coleta de dados de máquinas baseadas nestes sensores. Uma possível estratégia seria implementar em um desenvolvimento futuro uma rotina que iria se basear na produção ou tempo de trabalho da máquina para enviar possíveis avisos sobre a saúde da máquina.

4. RESULTADOS DO TC PROTÓTIPO DE SISTEMA PARA COLETA E EXIBIÇÃO DE DADOS DE MÁQUINAS DE CORTE DE FRIOS

Neste tópico serão apresentados os processos utilizados para a elaboração do projeto, bem como os resultados obtidos com seu desenvolvimento.

4.1 LEVANTAMENTO DE REQUISITOS

O sistema deve disponibilizar ao usuário a possibilidade de se cadastrar e acessar sua conta de forma segura, mantendo uma autenticação de quem o acessa. Todos os novos usuários cadastrados no sistema por padrão não terão privilégios de administradores, sendo necessário conceder estes privilégios de forma manual no banco de dados.

Os usuários administradores podem cadastrar modelos de máquinas, que são especificações padrões referentes as máquinas, para posteriormente ter-se a facilidade de cadastrar novas máquinas somente vinculando a elas uma especificação prévia.

O cadastro de novas máquinas será feito somente por usuários administradores, onde será necessário vincular um modelo à máquina. Ao ser cadastrada, a máquina gera um código serial único, que a identifica.

Para usufruir das funcionalidades gráficas do sistema, o usuário deve vincular uma máquina a sua conta, utilizando o código serial único referente a máquina.

A fim de gerar dados referentes receita gerada pelas máquinas, o usuário poderá cadastrar produtos e atribuir preços a eles, estes que terão seu histórico armazenado, para manter uma fidelidade de dados.

A produção das máquinas será registrada, possibilitando ao usuário visualizar os dados de forma intuitiva no sistema e posteriormente resgatar estes dados para projeções próprias.

4.1.1 Requisitos funcionais

Os requisitos funcionais foram desenvolvidos pensando tanto no sistema, como na comunicação das máquinas com o servidor, conforme é apresentado no Quadro 8.

Quadro 8 - Requisitos funcionais

| Número | Nome | Descrição |
|--------|----------------------------------|---|
| RF 01 | Login | Tela de login para o usuário acessar o sistema. |
| RF 02 | Cadastro de usuários | Tela de cadastro de usuários. |
| RF 03 | Layout padrão | Tela com layout base que será utilizado nas demais rotinas do sistema. |
| RF 04 | Tela inicial | Tela apresentada ao acessar o sistema. |
| RF 05 | Cadastro de modelos | Tela de cadastro de modelos de máquinas. |
| RF06 | Consulta de modelos | Tela de consulta de modelos de máquinas. |
| RF 07 | Cadastro de máquinas | Tela de cadastro de máquinas. |
| RF08 | Consulta de máquinas | Tela de consulta de máquinas. |
| RF 09 | Cadastro de produtos | Tela de cadastro de produtos. |
| RF 10 | Consulta de produtos | Tela de consulta de produtos. |
| RF 11 | Estatísticas da máquina | Tela com gráficos dos dados da máquina. |
| RF 12 | Estatísticas do produto | Tela com gráficos dos dados do produto. |
| RF 13 | Gráfico diário de produção | Gráfico com a produção total em quilogramas (Kg) dos últimos 10 dias. |
| RF 14 | Gráfico diário de receita | Gráfico com a receita total em reais (R\$) dos últimos 10 dias. |
| RF 15 | Gráfico de lucro total | Gráfico em forma de rosca com o lucro total em reais (R\$). |
| RF 16 | Gráfico de produção total | Gráfico em forma de rosca com a produção total em quilogramas (Kg). |
| RF 17 | Gráfico da pesagem em tempo real | Gráfico com a produção em gramas (g), em tempo real de uma máquina. |
| RF 18 | Exportação de dados | Ação para exportar os dados do sistema que foram gerados por uma máquina para um arquivo. |
| RF 19 | Rotina de notificação | Mensagens de <i>feedback</i> para o usuário ao executar uma ação. |
| RF 20 | Coleta de dados da máquina | Coletar os dados gerados pelo simulador das máquinas de cortes de frios. |
| RF 21 | Simulador de máquinas | Simular os dados que seriam gerados por máquinas físicas. |

Fonte: Acervo do autor (2022).

4.1.2 Requisitos não funcionais

No Quadro 9 estão especificados requisitos não funcionais que complementam os requisitos funcionais propostos ao sistema.

Quadro 9 - Requisitos não funcionais

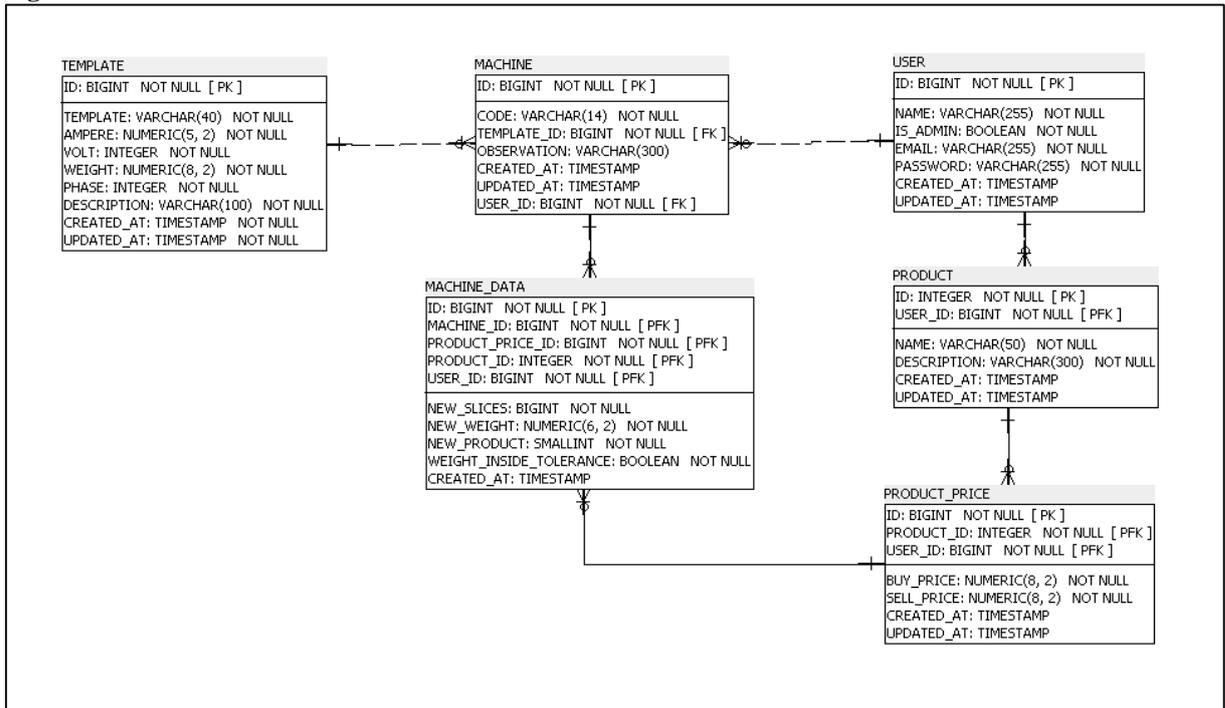
| Número | Nome | Descrição |
|--------|---|---|
| RNF 01 | Banco de dados | Utilização do banco de dados relacional postgres para efetuar o armazenamento dos dados do sistema. |
| RNF 02 | Servidor único para conexão com as máquinas | Os dados enviados pelas máquinas irão para um servidor a parte do sistema do usuário. |
| RNF 03 | Senhas fortes | Senhas devem possuir pelo menos 8 caracteres, contendo dentre eles pelo menos 1 caractere maiúsculo, um número e um caractere especial. |
| RNF 03 | Perfis de usuário | Deve haver um perfil para usuários administradores, e um perfil para usuários normais. |
| RNF 04 | Autenticação de usuário via token | Utilização tokens CSRF para validar as requisições dos usuários no sistema. |
| RNF 05 | Cards de exibição | Utilização de cards para exibição no layout padrão. |
| RNF 06 | Paginação da consulta | Utilização de uma paginação de dados nas telas de consulta. |
| RNF 07 | Validação de campos | Campos dos formulários que implementarem alguma validação devem ser responsivos e apresentar mensagem quando houver algum dado em desacordo. |
| RNF 08 | Formato dos arquivos exportados | Os arquivos exportados do sistema contendo os dados coletados das máquinas serão de formato csv. |
| RNF 10 | Tipo de notificação | O sistema deve possuir três tipos de notificação, sendo elas, notificação de sucesso, de alerta e de erro. Devem ser apresentadas em forma de pop-up. |
| RNF 11 | Gráfico iterativos | Gráficos iterativos que podem ser manipulados pelo usuário para facilitar sua exibição. |
| RNF 12 | Múltipla coleta de dados | Sistema deve dar suporte à conexão de mais de uma máquina simultânea para coleta de dados. |
| RNF 13 | Tempo de resposta de exibição dos dados coletados | Tempo de resposta do gráfico em tempo real deve estar abaixo dos 3 segundos de latência. |

Fonte: Acervo do autor (2022).

4.2 ESTRUTURA DO BANCO DE DADOS

A Figura 7 mostra como foi estruturado o modelo dados do sistema, sendo a tabela template, a responsável por armazenar os dados referentes as especificações dos modelos de máquinas, a tabela machine se encarrega do cadastro de novas máquinas e a qual usuário elas pertencem, a tabela user armazena os novos usuários, a tabela product guarda os produtos do sistema, a tabela product_price armazena o histórico dos preços dos produtos, e a tabela machine_data arquivo os dados coletados das máquinas em operação.

Figura 7 – Modelo relacional banco de dados

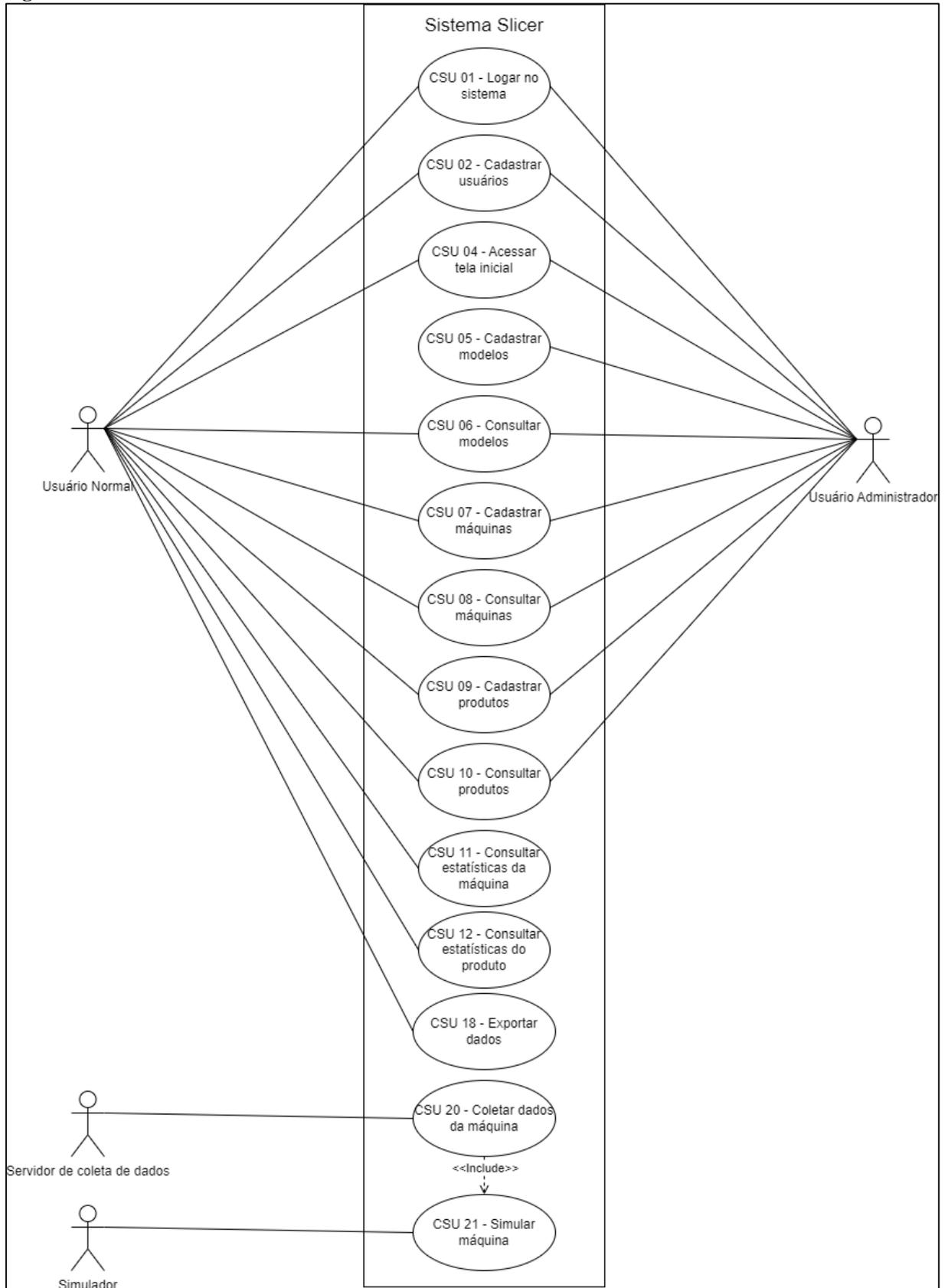


Fonte: Acervo do autor (2022).

4.3 CASOS DE USO

Os casos de uso representam de que forma espera-se que o sistema seja utilizado e quais agentes estão aptos a utilizarem determinada rotina. É possível verificar os casos de uso do protótipo na Figura 8.

Figura 8 – Casos de uso



Fonte: Acervo do autor (2022).

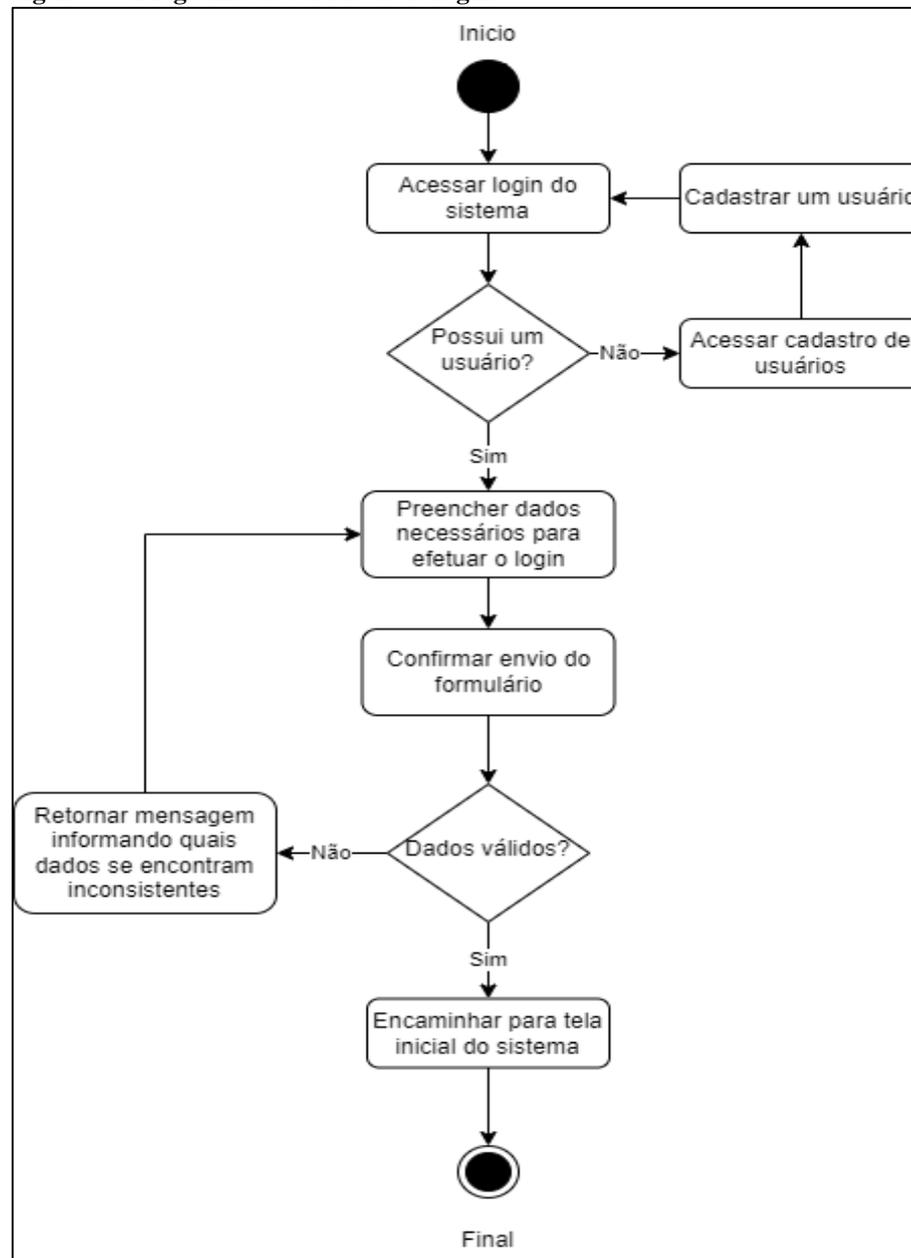
4.4 DIAGRAMA DE ATIVIDADE

Os diagramas de atividade apresentados nas figuras a seguir tem como objetivo demonstrar como o sistema deve se comportar quando sendo utilizado pelos usuários.

4.4.1 Diagrama de atividade do login

A Figura 9 apresenta o fluxo em que o login do usuário deve seguir para obter êxito ao entrar no sistema.

Figura 9 – Diagrama de atividade do login

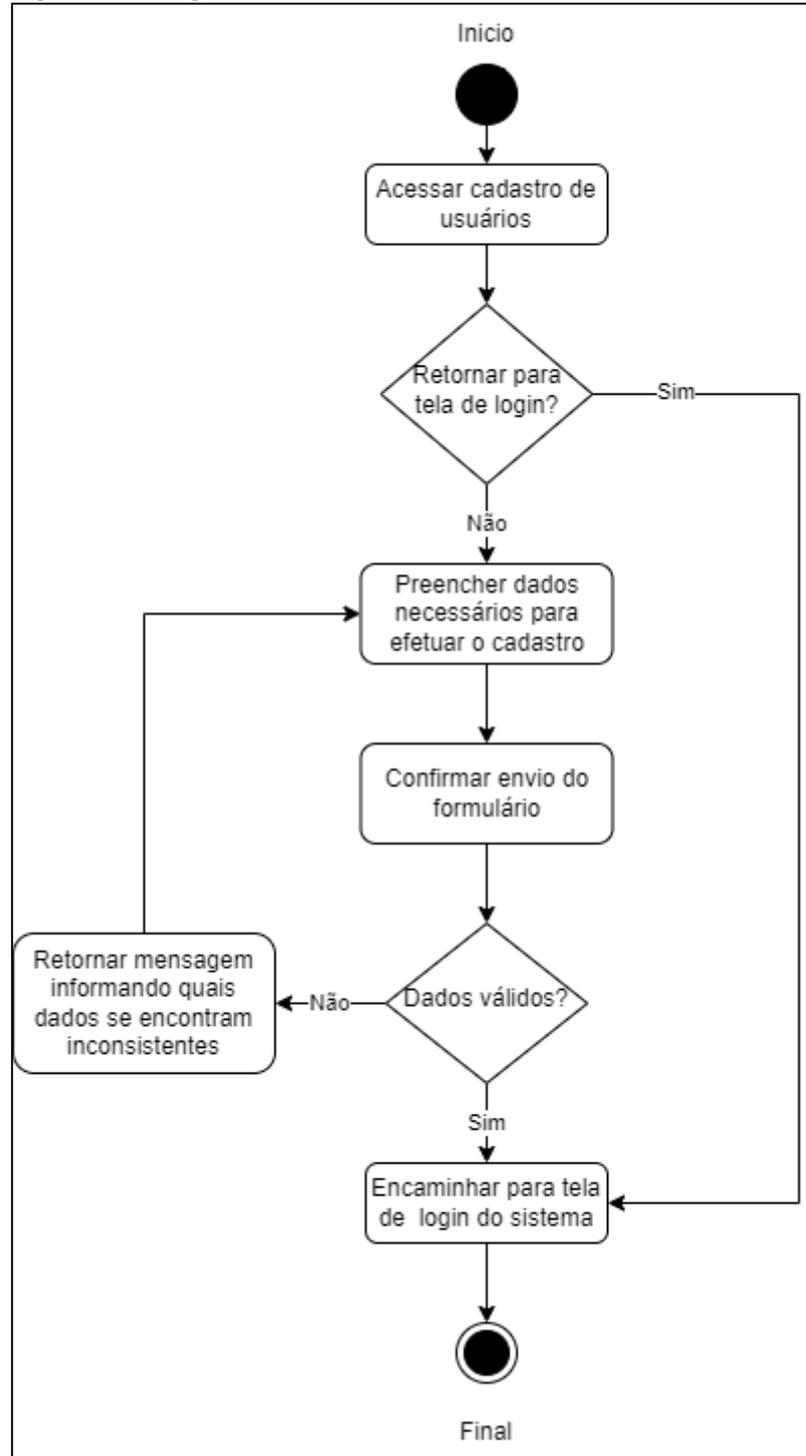


Fonte: Acervo do autor (2022).

4.4.2 Diagrama de atividade do cadastro de usuário

A Figura 10 apresenta o fluxo em que o cadastro de usuário deve seguir e suas possíveis exceções, para inserir um novo usuário ao sistema.

Figura 10 – Diagrama de atividade do cadastro de usuário

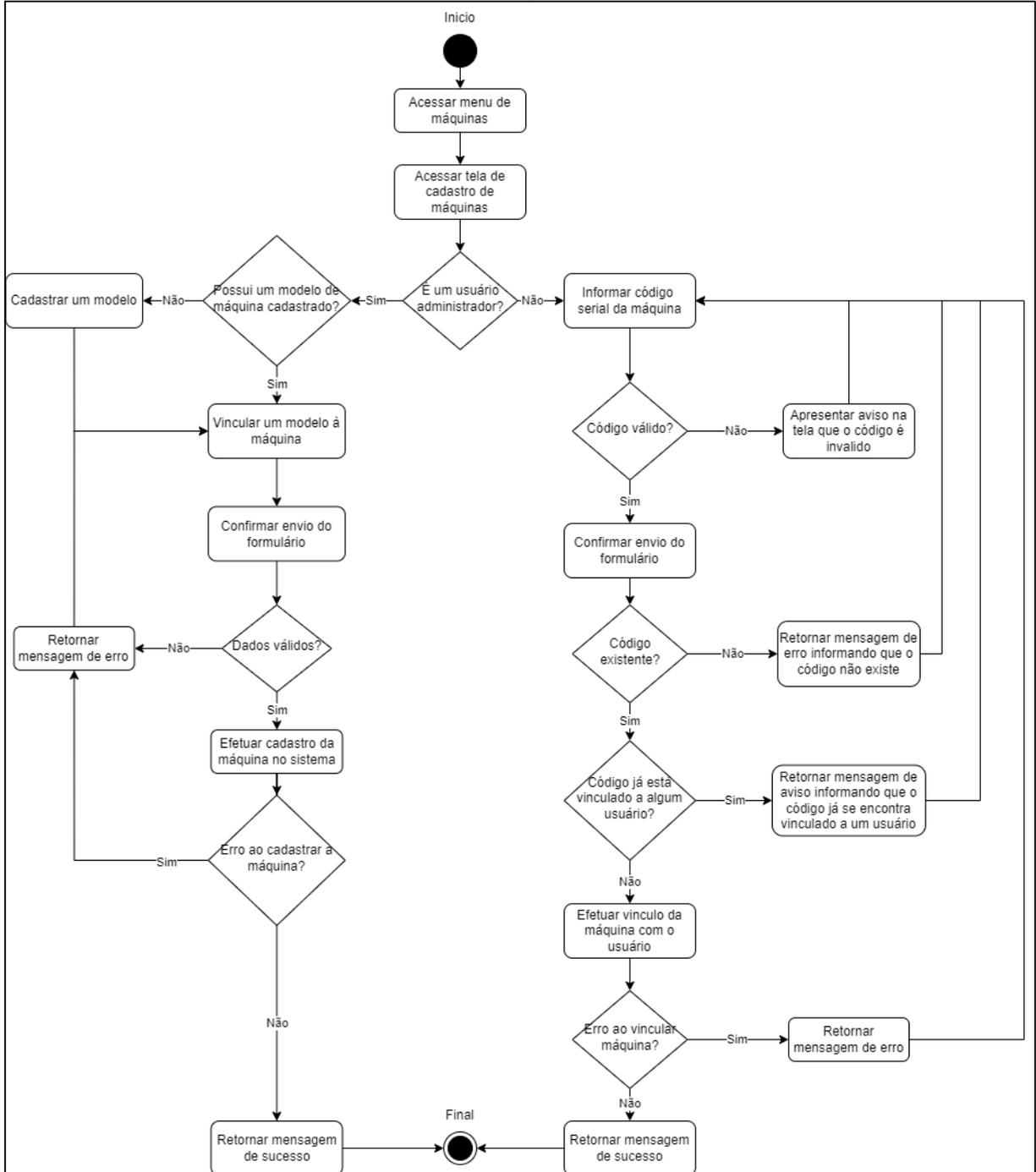


Fonte: Acervo do autor (2022).

4.4.3 Diagrama de atividade do cadastro de máquinas

A Figura 11 apresenta qual fluxo em que o cadastro de máquinas no sistema deve seguir e suas possíveis exceções.

Figura 11 – Diagrama de atividade do cadastro de máquinas

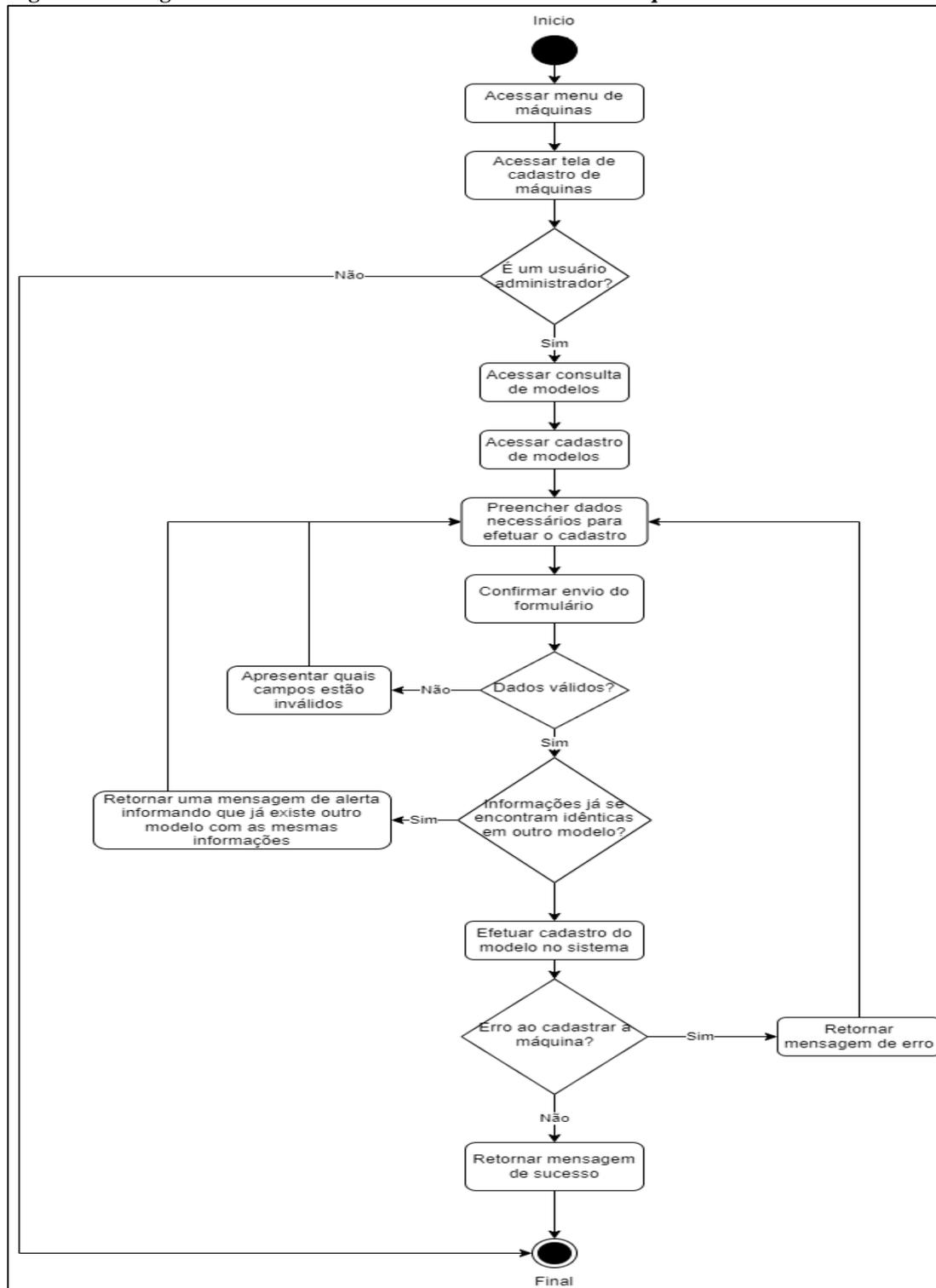


Fonte: Acervo do autor (2022).

4.4.4 Diagrama de atividade do cadastro de modelos de máquinas

A Figura 12 apresenta o fluxo em que o cadastro de modelos de máquinas no sistema deve seguir e suas possíveis exceções.

Figura 12 – Diagrama de atividade do cadastro de modelos de máquinas

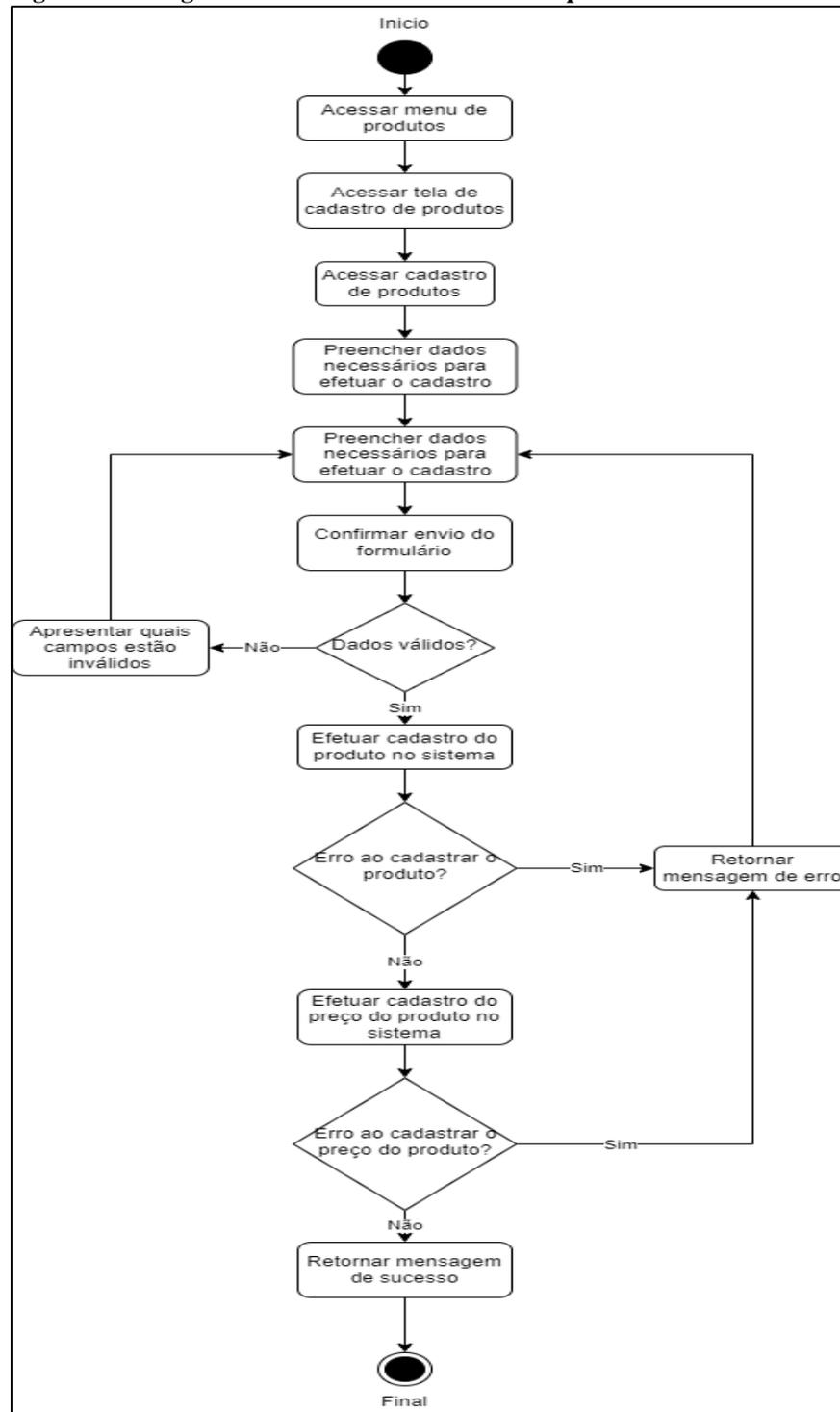


Fonte: Acervo do autor (2022).

4.4.4 Diagrama de atividade do cadastro de produtos

A Figura 13 apresenta o fluxo em que o cadastro de produtos no sistema deve seguir e suas possíveis exceções.

Figura 13 – Diagrama de atividade do cadastro de produtos

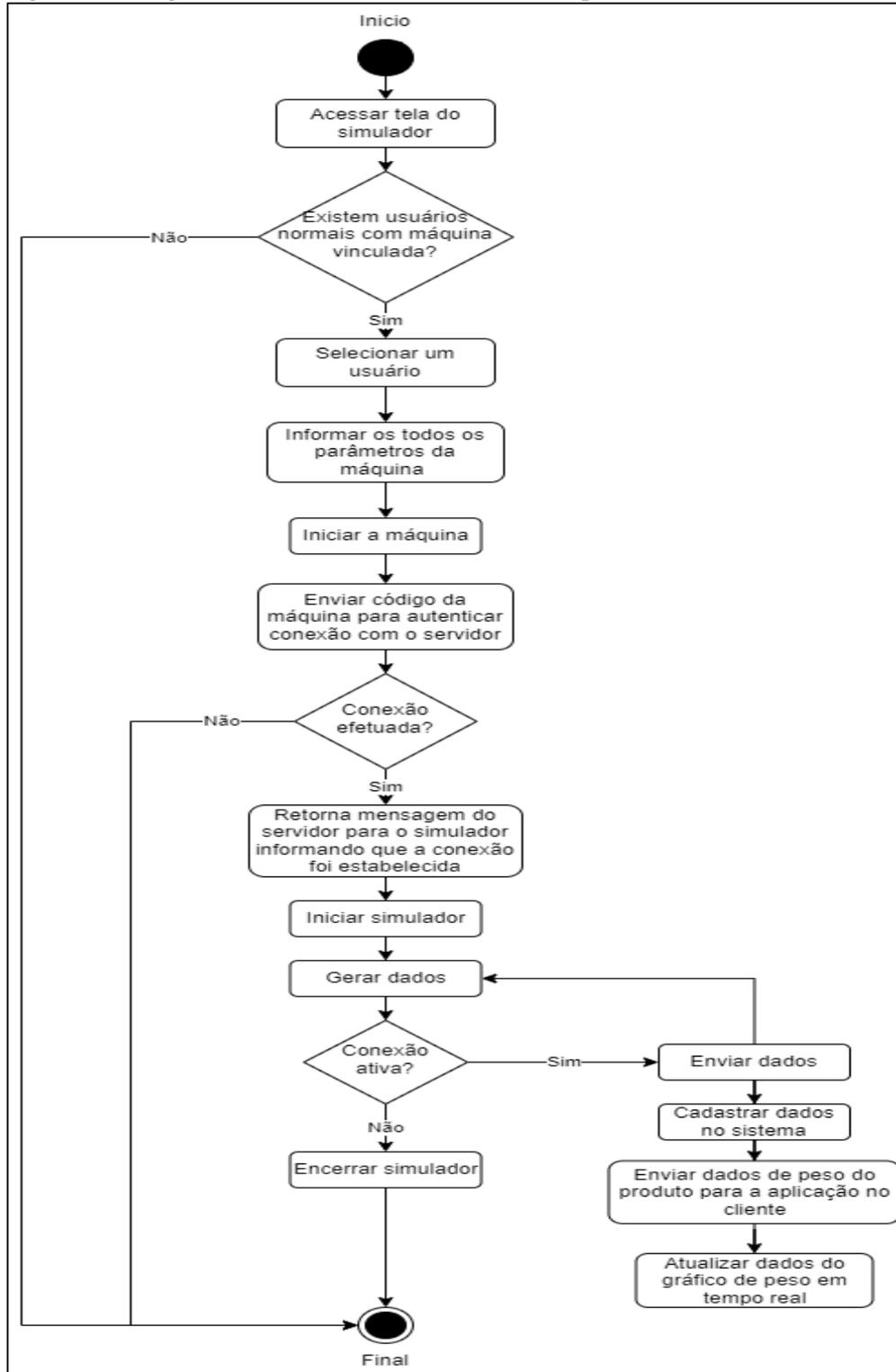


Fonte: Acervo do autor (2022).

4.4.4 Diagrama de atividade do simulador da máquina de corte de frios

A Figura 14 apresenta o fluxo em que o cadastro de modelos de máquinas no sistema deve seguir e suas possíveis exceções.

Figura 14 – Diagrama de atividade do simulador da máquina de corte de frios



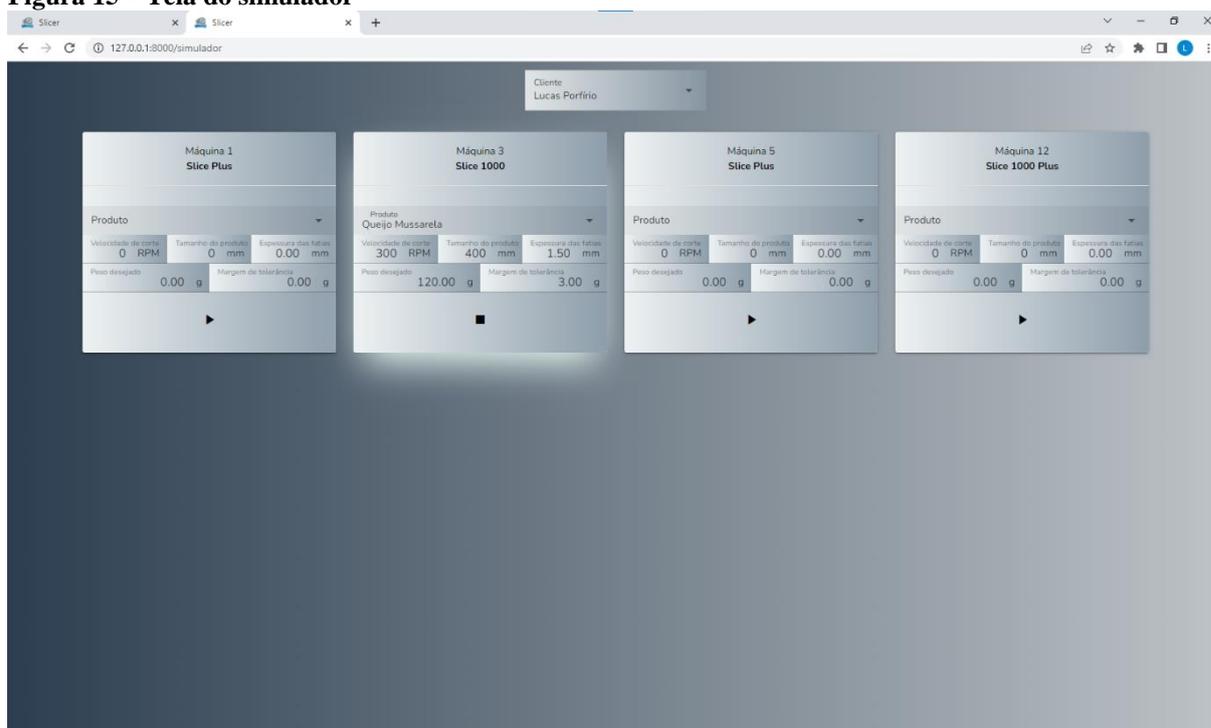
Fonte: Acervo do autor (2022).

4.5 SIMULADOR

Fatiadoras são máquinas muito grandes, pesadas e caras, além de necessitarem de um ambiente adequado para sua instalação e funcionamento. Devido a esses empecilhos, foi desenvolvido um simulador, que irá representar as máquinas de cortes de frios, podendo assim gerar os dados necessários e efetuar a comunicação com o sistema.

Com o intuito de poder configurar os parâmetros do simulador, foi criada uma tela onde podem ser filtradas as máquinas vinculadas a um usuário normal conforme Figura 15. Nesta tela é possível configurar qual o tipo de produto que a máquina irá produzir, a velocidade do corte das fatias em RPM, qual o tamanho do produto a ser fatiado em milímetros, a espessura das fatias em milímetros, o peso desejado para cada pilha de fatias, e a margem de tolerância aceitável para o peso da pilha de fatias.

Figura 15 – Tela do simulador



Fonte: Acervo do autor (2022).

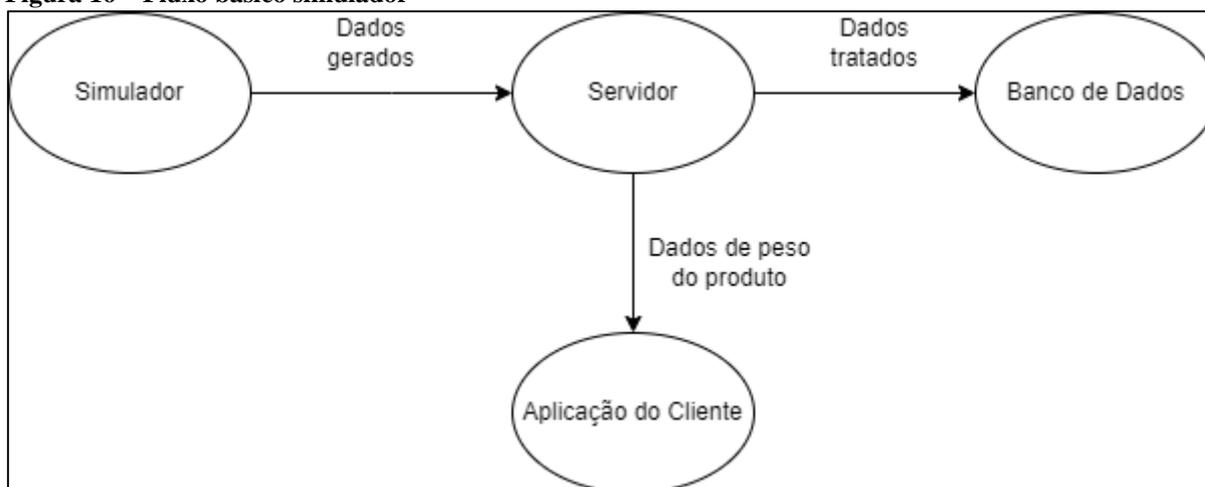
Os parâmetros informados são utilizados para gerar dados no intervalo de um segundo como, número de novas fatias produzidas, peso da pilha de fatias, se o peso da pilha se encontra dentro da margem de tolerância aceitável, e se foi iniciado o corte de uma nova peça na máquina.

Ao iniciar uma máquina no simulador, ela cria uma conexão socket com o servidor enviando o seu código serial de maneira a autenticar que é uma máquina cadastrada no sistema.

Estando com a conexão estabelecida ela começa a enviar os dados gerados pelo simulador, assim como o código do produto selecionado, no intervalo de um segundo entre cada envio, simulando como seria o funcionamento em uma máquina física.

O servidor ao receber os dados, irá tratá-los e enviá-los ao banco de dados, bem como com o horário em que foi feito o cadastro. Os dados de peso gerados serão enviados também para a aplicação do cliente, gerando assim o gráfico em tempo real do peso dos produtos. Um fluxo básico do funcionamento pode ser visualizado na figura 16.

Figura 16 – Fluxo básico simulador



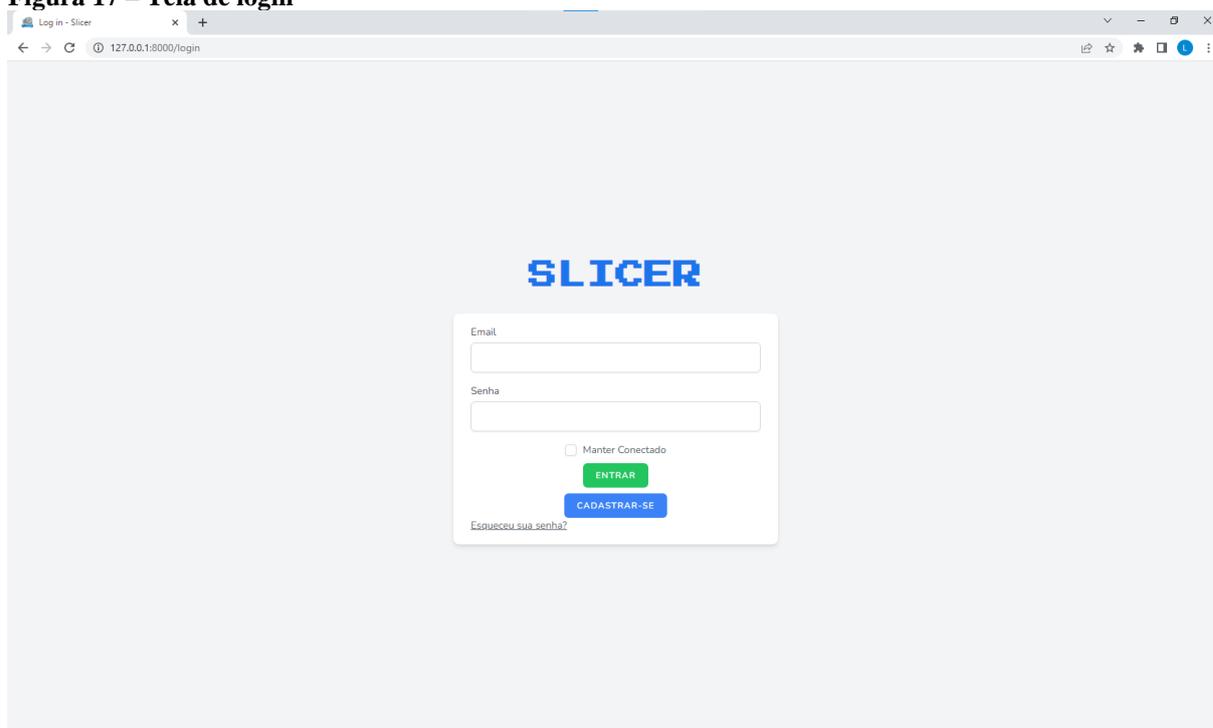
Fonte: Acervo do autor (2022).

4.6 TELAS DO SISTEMA

As telas do protótipo de sistema foram construídas de forma manter o usuário sempre no mesmo ecossistema, utilizando de um layout base, que fará com que o usuário não perceba a mudança de páginas.

4.6.1 Tela de login

Deve ser a tela inicial para qualquer usuário que não esteja registrado no sistema ou não esteja logado no momento. Ao acessar esta tela o usuário tem a possibilidade de acessar sua conta já cadastrada ou então seguir para a rotina de cadastro de usuários. Também existe a possibilidade do usuário manter sua conta conectada ao selecionar esta opção na tela. A tela de login está representada na Figura 17.

Figura 17 – Tela de login

Fonte: Acervo do autor (2022).

4.6.2 Tela de cadastro de usuário

Ao acessar esta tela o usuário tem a possibilidade de cadastrar sua conta no sistema, ou então caso já tenha uma conta, voltar para a tela de login. Será acessada pela tela de login ao usuário clicar em “Cadastrar-se”.

Os usuários cadastrados são por padrão usuários não administradores e os usuários administradores somente terão seus privilégios concedidos manualmente via banco de dados. É possível visualizar a tela de cadastro na Figura 18.

Figura 18 – Tela de cadastro

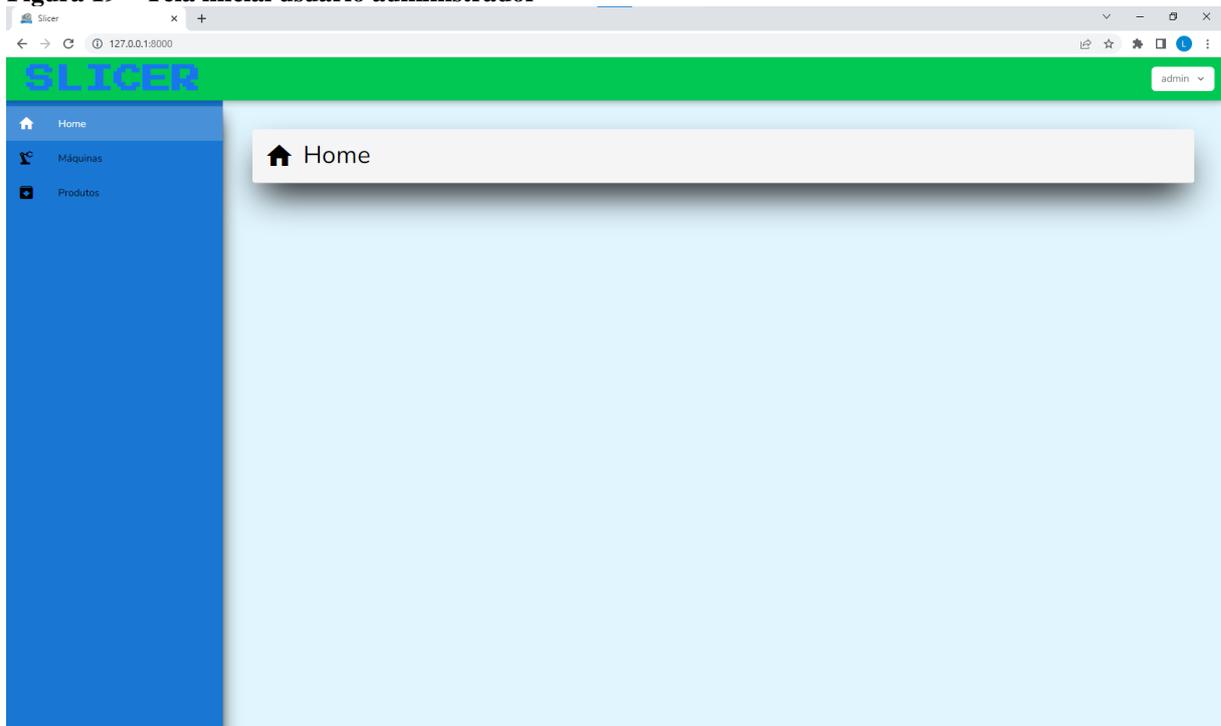
The image shows a web browser window displaying the registration page for 'SLICER'. The browser's address bar shows the URL '127.0.0.1:8000/register?email=&password=&remember=false'. The page content includes the 'SLICER' logo in a blue, pixelated font. Below the logo is a registration form with the following fields: 'Nome', 'Email', 'Senha', and 'Confirme a senha'. At the bottom of the form, there is a link that says 'já possui uma conta?' and a green button labeled 'CADASTRAR'.

Fonte: Acervo do autor (2022).

4.6.3 Tela inicial

Esta tela tem comportamentos diferentes dependendo de qual usuário a está acessando. Quando com um usuário administrador, a tela não possuirá nenhuma interação em sua parte central, conforme Figura 19.

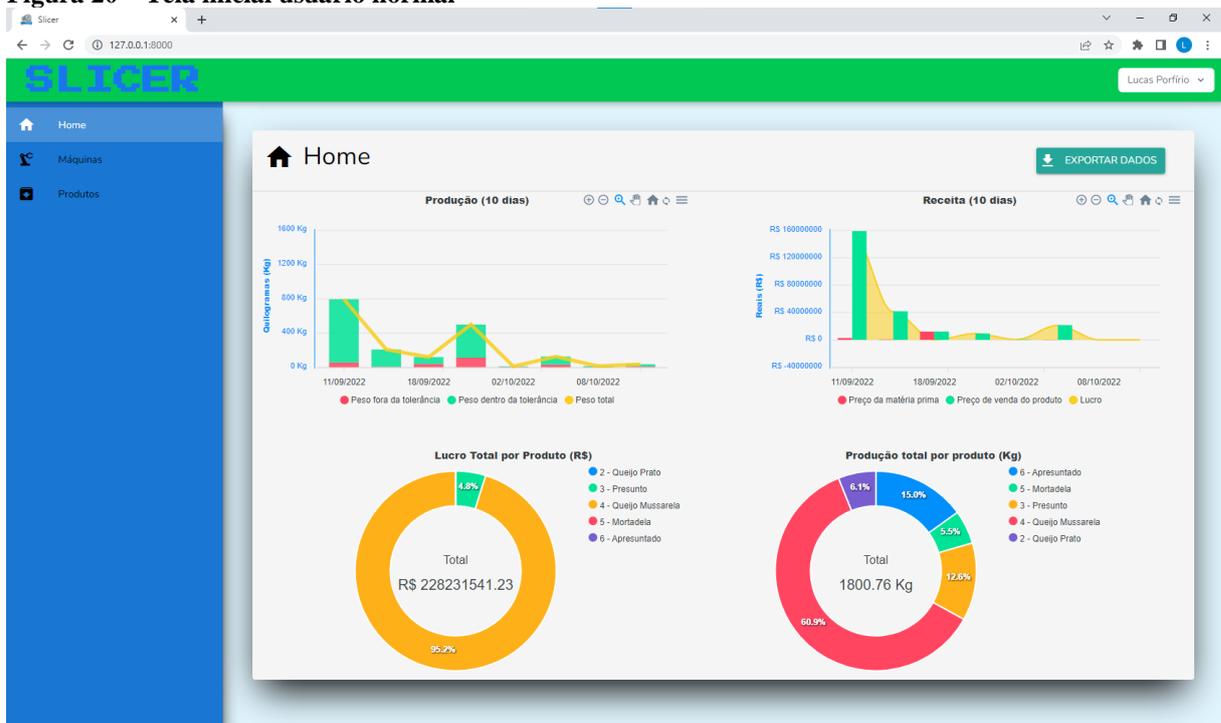
Figura 19 – Tela inicial usuário administrador



Fonte: Acervo do autor (2022).

Quando com um usuário normal irá disponibilizar uma ação de exportar dados e terá gráficos referentes as máquinas vinculadas ao usuário conforme Figura 20.

Figura 20 – Tela inicial usuário normal



Fonte: Acervo do autor (2022).

4.6.4 Tela de consulta de máquinas

Ao acessar a rotina são exibidos os registros das máquinas cadastradas no sistema em forma de uma tabela. Seu acesso é feito através do menu lateral da tela principal ao clicar em “Máquinas”.

Quando com um usuário administrador, estão visíveis todas as máquinas cadastradas no sistema e disponíveis as ações básicas de edição, conforme Figura 21.

Figura 21 – Tela de consulta de máquinas do usuário administrador

| ID | Modelo | Voltagem | Amperagem | Fases | Peso | |
|----|-----------------|----------|-----------|------------|------------|--|
| 1 | Slice Plus | 380 V | 100.00 A | Trifásico | 1000.00 Kg | |
| 3 | Slice 1000 | 110 V | 110.00 A | Monofásico | 11.11 Kg | |
| 4 | Slice Plus | 380 V | 100.00 A | Trifásico | 1000.00 Kg | |
| 5 | Slice Plus | 380 V | 100.00 A | Trifásico | 1000.00 Kg | |
| 7 | Slice Plus | 380 V | 100.00 A | Trifásico | 1000.00 Kg | |
| 8 | Slice 1000 | 110 V | 110.00 A | Monofásico | 11.11 Kg | |
| 9 | Slice Plus | 380 V | 100.00 A | Trifásico | 1000.00 Kg | |
| 11 | Slice Plus | 380 V | 100.00 A | Trifásico | 1000.00 Kg | |
| 12 | Slice 1000 Plus | 220 V | 110.00 A | Monofásico | 110.00 Kg | |
| 13 | Slice 2700 | 110 V | 110.00 A | Monofásico | 110.00 Kg | |

Fonte: Acervo do autor (2022).

Quando com um usuário normal, são apresentadas somente máquinas vinculadas ao usuário logado, removida a ação de alterar e adicionado a ação de estatísticas da máquina, conforme apresentado na Figura 22.

Figura 22 – Tela de consulta de máquinas com usuário normal

| ID | Modelo | Voltagem | Amperagem | Fases | Peso |
|----|-----------------|----------|-----------|------------|------------|
| 1 | Slice Plus | 380 V | 100.00 A | Trifásico | 1000.00 Kg |
| 3 | Slice 1000 | 110 V | 110.00 A | Monofásico | 11.11 Kg |
| 5 | Slice Plus | 380 V | 100.00 A | Trifásico | 1000.00 Kg |
| 12 | Slice 1000 Plus | 220 V | 110.00 A | Monofásico | 110.00 Kg |

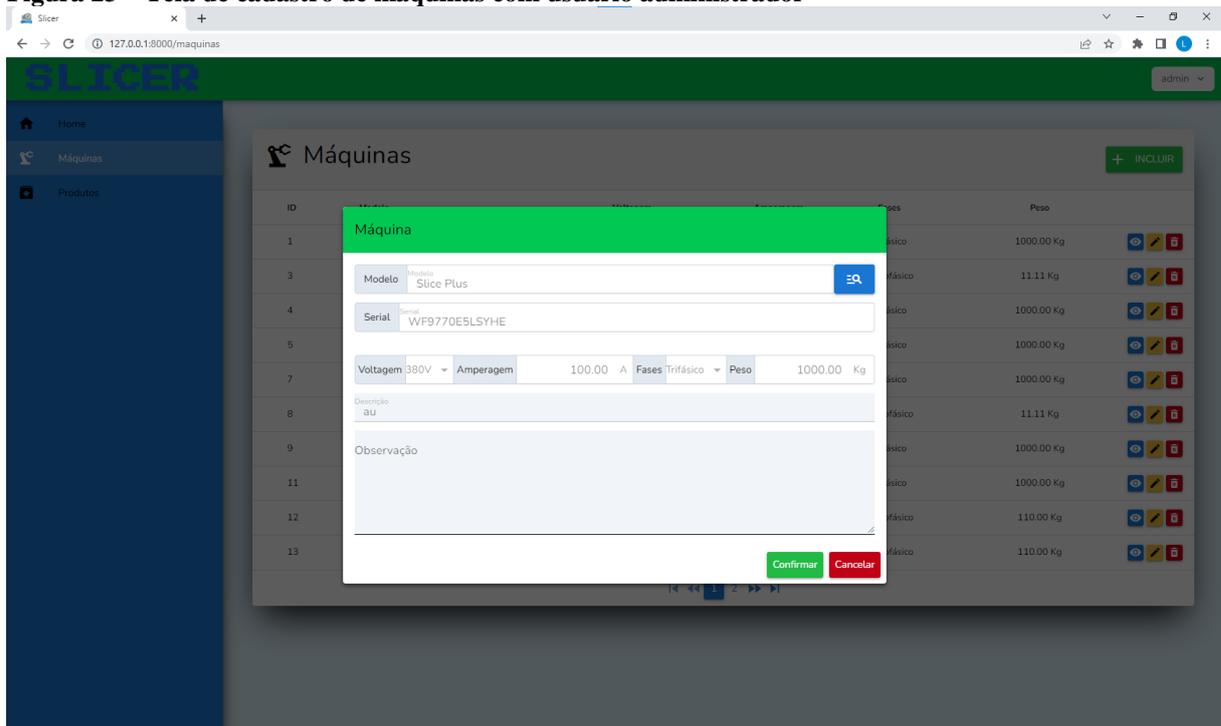
Fonte: Acervo do autor (2022).

4.6.5 Tela de cadastro de máquinas

Quando com um usuário administrador, é possível escolher qual modelo deseja vincular a máquina consultando pela lupa ao lado do campo, que leva o usuário a consulta de modelos. Ao confirmar irá gerar um código serial único vinculado a máquina. A Figura 23 demonstra a aparência da tela.

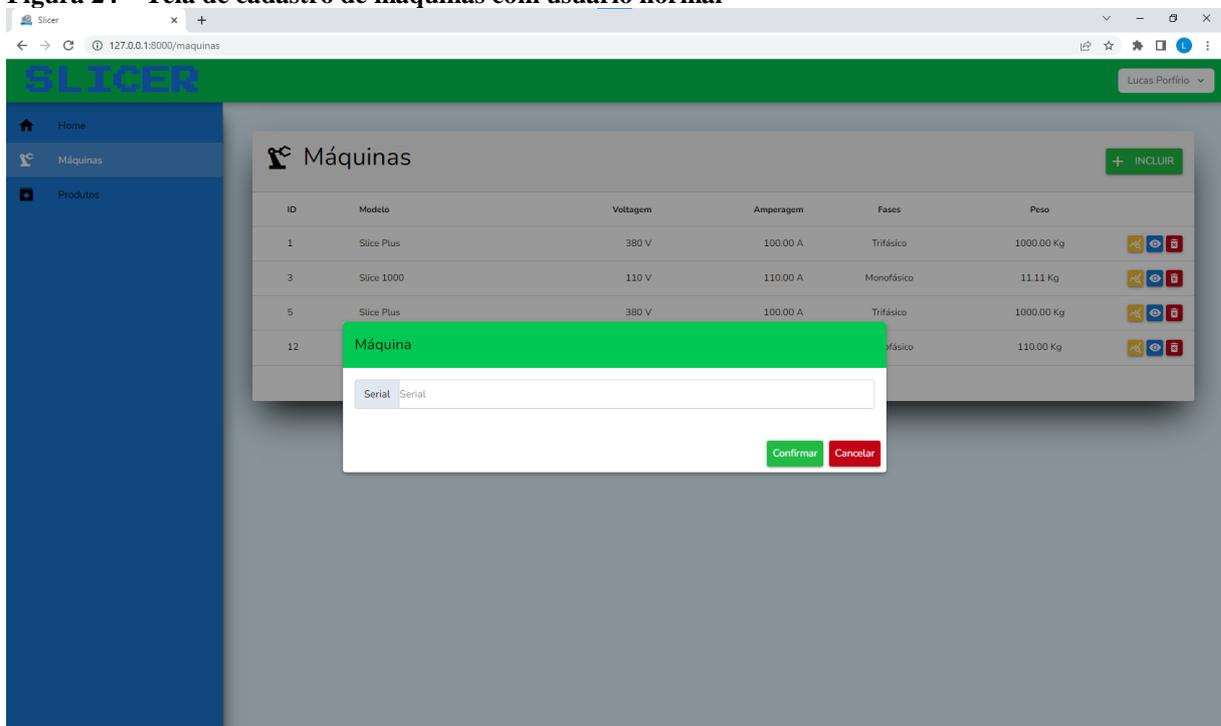
Já quando com um usuário normal, ao incluir uma máquina virá somente uma janela pedindo para informar o serial da máquina que deseja vincular à conta, conforme Figura 24. Após isso o usuário pode utilizar a tela de cadastro somente para visualizar os dados da máquina conforme Figura 23.

Figura 23 – Tela de cadastro de máquinas com usuário administrador



Fonte: Acervo do autor (2022).

Figura 24 – Tela de cadastro de máquinas com usuário normal

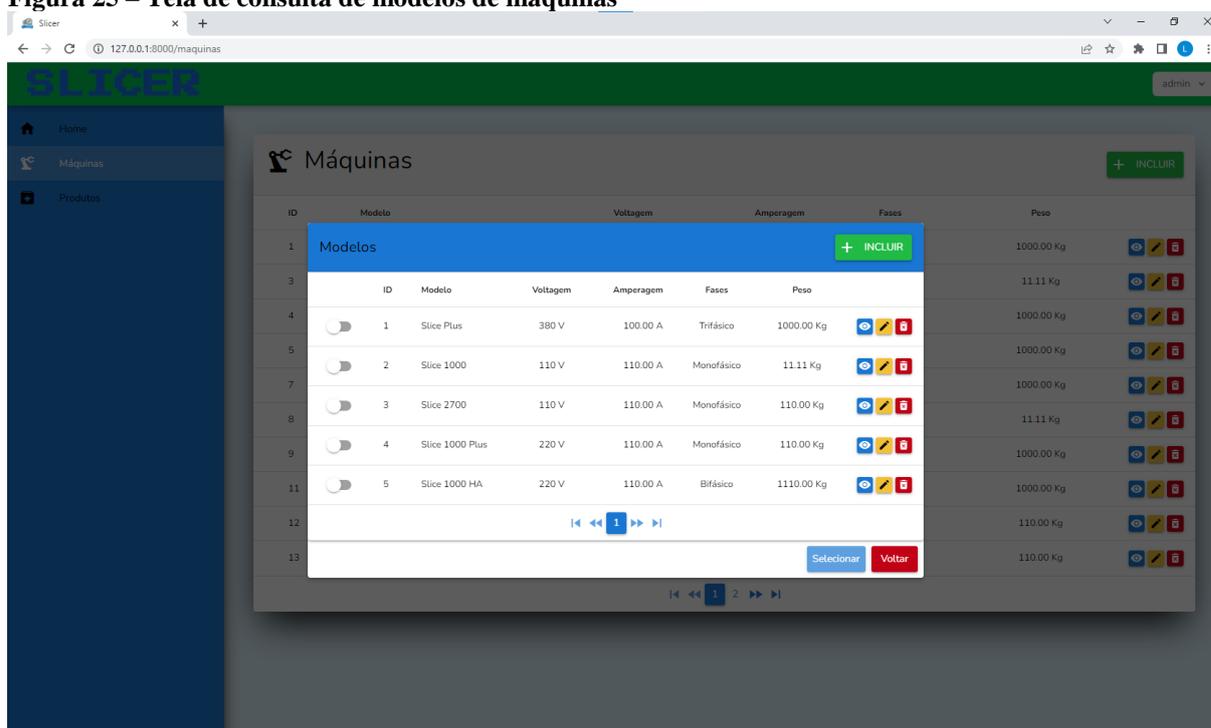


Fonte: Acervo do autor (2022).

4.6.6 Tela de consulta de modelos de máquinas

Seu acesso será feito através da tela de cadastro de máquinas e só pode ser utilizada por usuários administradores. Ao selecionar algum dos modelos, os dados do mesmo são enviados para preencher a tela de cadastro de máquinas com as informações referentes ao modelo. A Figura 25 demonstra a aparência da tela de consulta dos modelos.

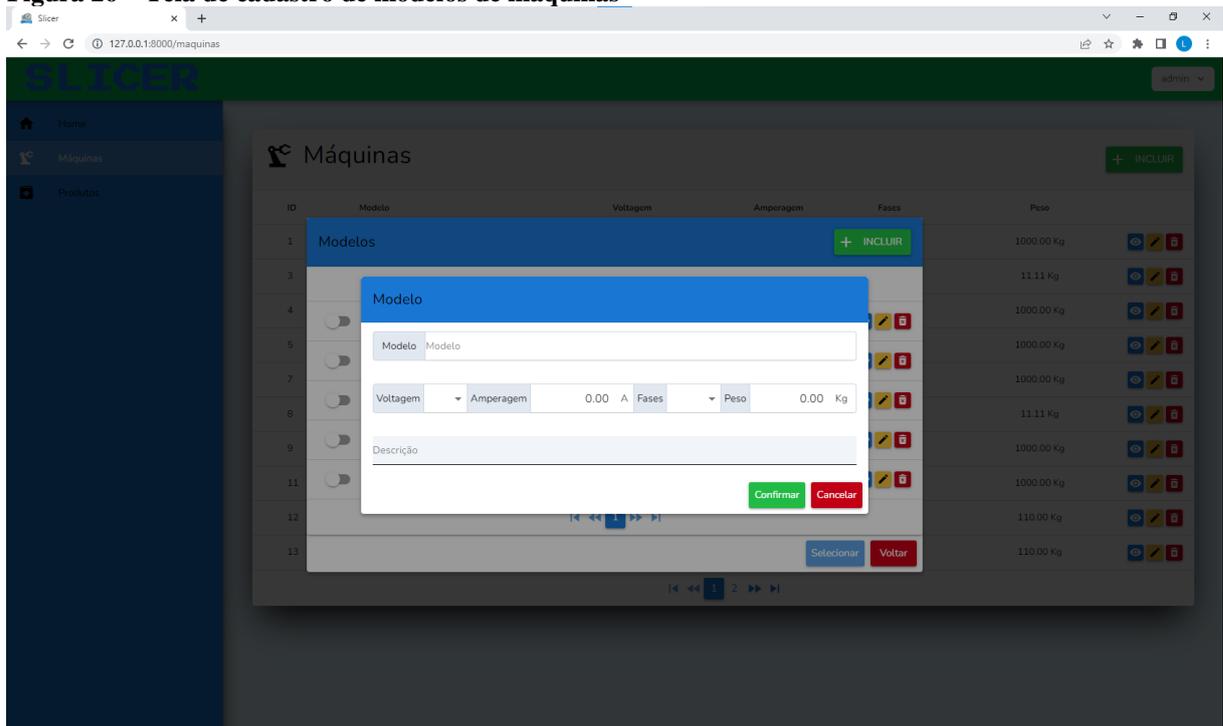
Figura 25 – Tela de consulta de modelos de máquinas



Fonte: Acervo do autor (2022).

4.6.7 Tela de cadastro de modelos de máquinas

A Figura 26 apresenta a tela onde são cadastradas as especificações referentes as máquinas, para que assim ao cadastrar uma nova máquina ao sistema, seja possível utilizar configurações padrões. Não deve ser possível cadastrar um modelo de máquina com as mesmas informações já existentes em outro modelo.

Figura 26 – Tela de cadastro de modelos de máquinas

Fonte: Acervo do autor (2022).

4.6.8 Tela de consulta de produtos

Ao acessar a tela são exibidos os registros dos produtos cadastrados no sistema vinculados ao usuário logado, conforme Figura 27. Seu acesso deve ser feito através do menu lateral da tela principal ao clicar em “Produtos”.

Quando com um usuário administrador, não será apresentada a ação de estatística, sendo esta exclusiva para usuário normais pois contém vínculos com máquinas.

Figura 27 – Tela de consulta de produtos

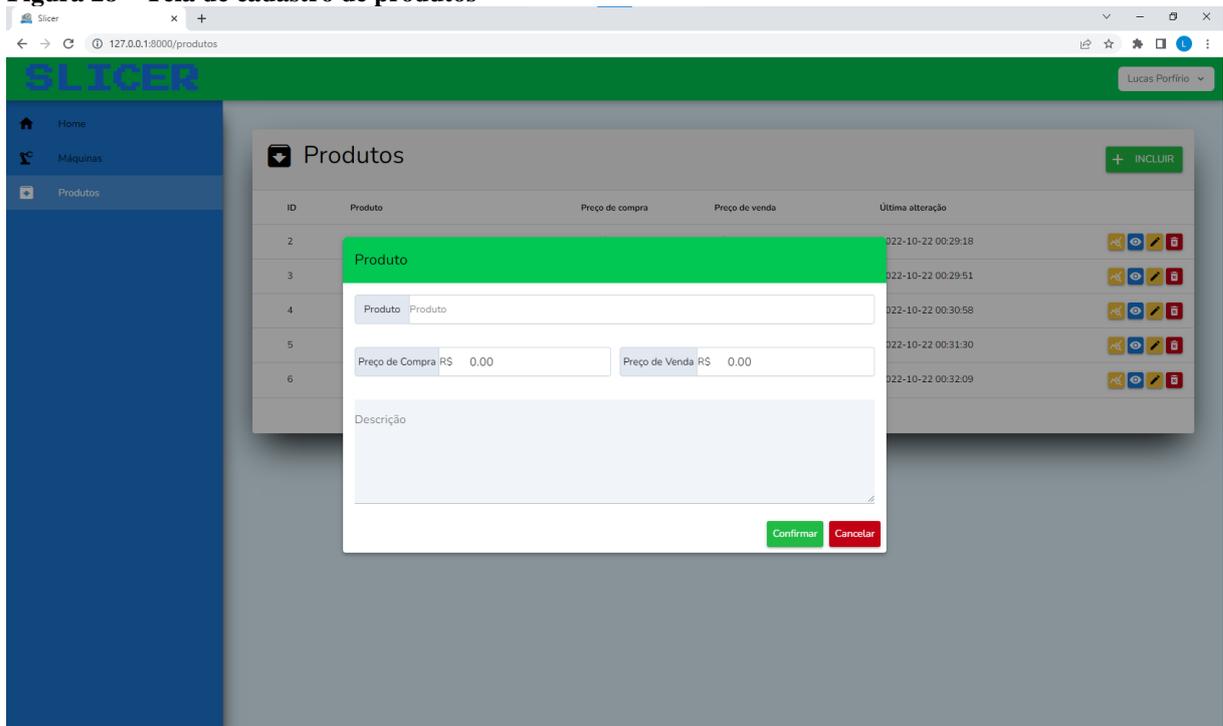
| ID | Produto | Preço de compra | Preço de venda | Última alteração |
|----|------------------|-----------------|----------------|---------------------|
| 2 | Queijo Prato | R\$20.00 | R\$30.00 | 2022-10-22 00:29:18 |
| 3 | Presunto | R\$10.00 | R\$20.00 | 2022-10-22 00:29:51 |
| 4 | Queijo Mussarela | R\$30.00 | R\$35.00 | 2022-10-22 00:30:58 |
| 5 | Mortadela | R\$7.00 | R\$12.00 | 2022-10-22 00:31:30 |
| 6 | Apresuntado | R\$5.00 | R\$8.00 | 2022-10-22 00:32:09 |

Fonte: Acervo do autor (2022).

4.6.9 Tela de cadastro de produtos

A Figura 28 apresenta a tela de cadastro de produtos, que tem como principal foco adicionar uma precificação ao produtos.

Ao cadastrar ou alterar um produto, os valores de compra e venda do produto são armazenados em uma tabela separada do banco de dados. Assim sempre que for alterado o valor de um produto, tem-se o histórico dos produtos anteriores e as datas de alteração.

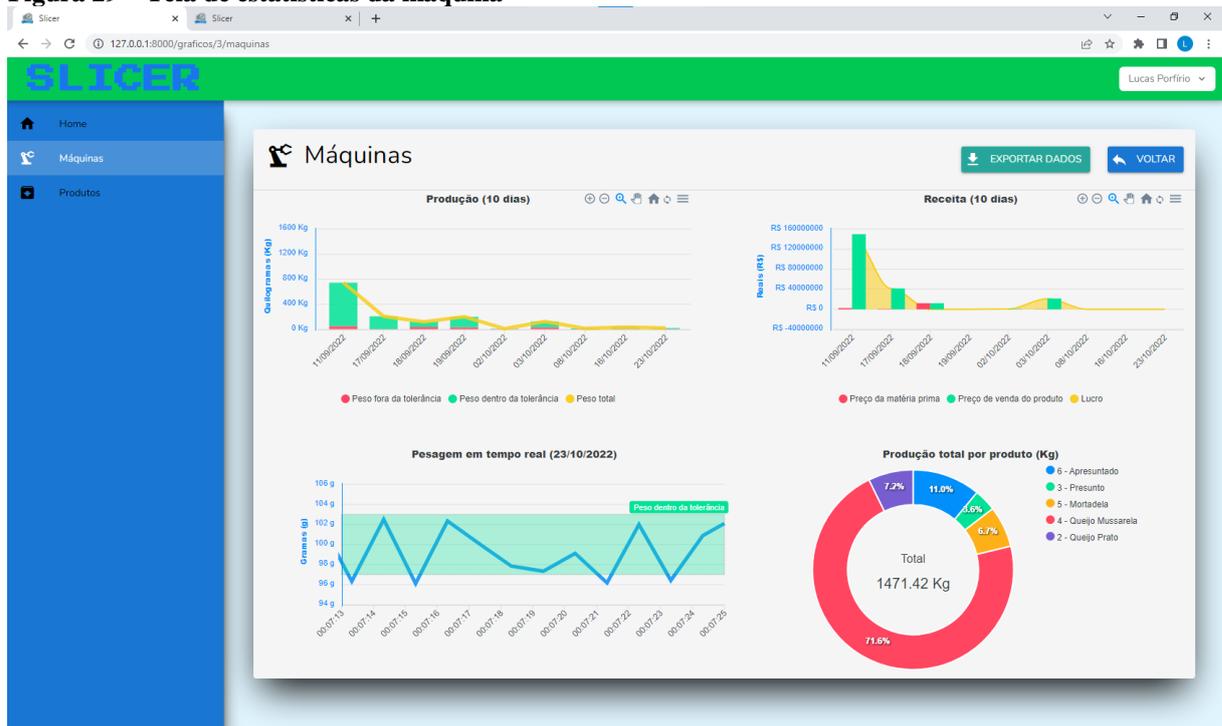
Figura 28 – Tela de cadastro de produtos

Fonte: Acervo do autor (2022).

4.6.10 Tela de estatísticas da máquina

A tela de estatísticas da máquina, conforme apresentada na Figura 29, reúne gráficos da produção e lucro da máquina, bem como a sua produção em tempo real. Os gráficos apresentados são baseados nos dados coletados da máquina acessada. Caso o usuário queira, ele tem a possibilidade de exportar esses dados para um arquivo csv.

Figura 29 – Tela de estatísticas da máquina

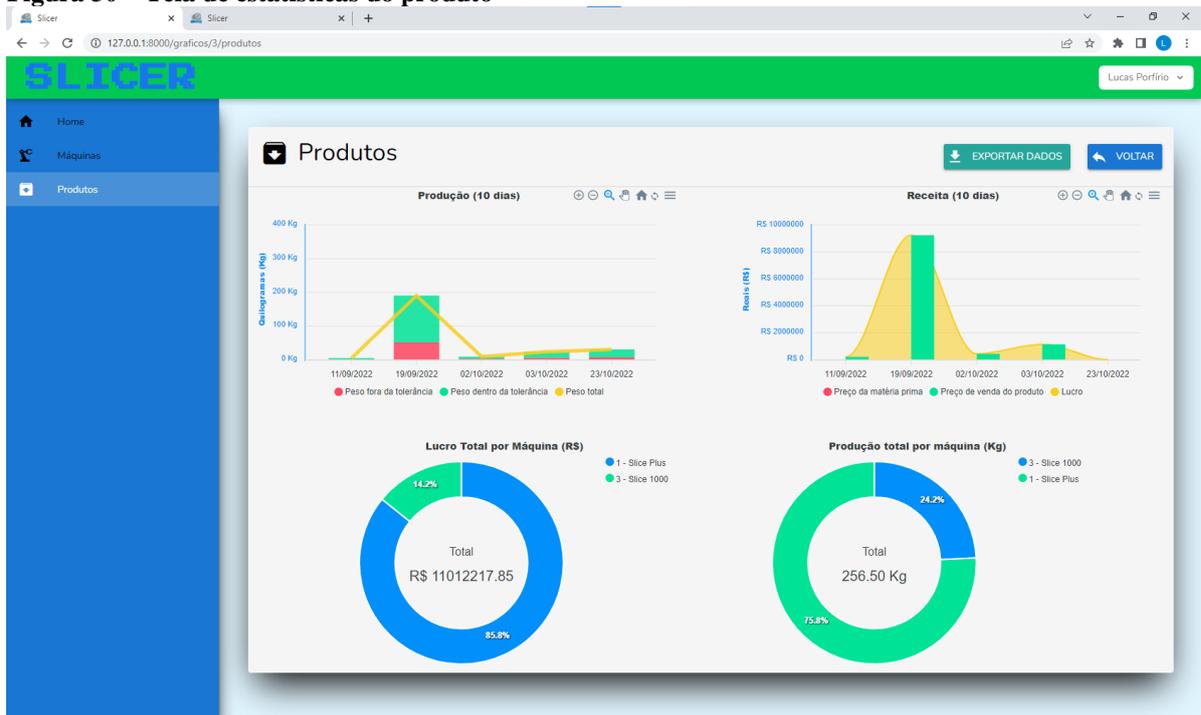


Fonte: Acervo do autor (2022).

4.6.11 Tela de estatísticas do produto

A tela de estatísticas do produto apresenta gráficos referentes a produção e o lucro do produto em relação as máquinas do sistema, conforme Figura 30. Caso o usuário queira, ele tem a possibilidade de exportar os dados relacionados ao produto para um arquivo csv.

Figura 30 – Tela de estatísticas do produto



Fonte: Acervo do autor (2022).

5. CONCLUSÃO

Conforme apresentado no trabalho, atualmente o maior diferencial que uma empresa pode ter sobre suas concorrentes é a possibilidade de tomar medidas mais assertivas em suas decisões de mercado, e para que essa possibilidade exista é necessário que estas decisões sejam baseadas em dados reais e precisos. Na indústria, muitos destes dados estão nas linhas de produção o que dificulta o acesso a essas informações. Portanto, o trabalho buscou apresentar uma solução de coleta de dados de uma indústria com máquinas de fatiamentos de frios.

Ao final do trabalho, verificou-se que todos os objetivos traçados foram alcançados com sucesso. Onde o levantamento de requisitos foi o pontapé inicial, nele foram avaliados quais dados deveriam ou poderiam ser coletados das máquinas, bem como, a maneira com que seriam exibidos em tela. Assim, como também de que maneira seria feita essa coleta. Em seguida houve o desenvolvimento do simulador da máquina, que foi uma parte essencial para o projeto, pois sem ele não se tornaria possível a coleta dos dados, já que era totalmente inviável conseguir uma máquina física para efetuar os testes. Outra parte muito importante, foi o desenvolvimento da comunicação com a máquina, pois somente desta maneira foi possível coletar os dados gerados e armazená-los de forma que fossem úteis posteriormente. Por fim, houve o desenvolvimento do sistema como um todo, onde foi possível implementar todas as regras do negócio e seus requisitos funcionais e não funcionais. O protótipo então coleta, exibe e disponibiliza a exportação dos dados gerados pela máquina fatiadora de frios.

O tema abordado no trabalho visou contribuir com o aumento de produtividade na indústria, que por sua vez irá reduzir o custo final do produto, aumentar os lucros da empresa e gerar possíveis novos empregos, beneficiando assim a sociedade como um todo. Este trabalho também serviu como um crescimento profissional, pois durante o desenvolvimento do mesmo foram utilizadas várias novas tecnologias, e foi-se aprofundado nas que já possuía domínio, além de ser uma experiência muito enriquecedora em ter a possibilidade de passar por todas as partes de desenvolvimento de um projeto.

5.1 SUGESTÕES DE MELHORIA FUTURAS

Com o fim do protótipo obteve-se um resultado muito satisfatório baseada na ideia inicial do projeto, porém observou-se também a possibilidade de futuramente implementar novos recursos ao projeto, o tornando mais robusto e completo. Uma dessas possibilidades é a de se aprofundar mais no sistema em si, adicionando mais maneiras de se gerenciar máquinas e produtos, como por exemplo, adicionar uma rotina para agendar revisões ou manutenções nas

máquinas. Outro ponto que somaria muito ao projeto, seria a possibilidade de adicionar rotinas para análise de dados já dentro do sistema, sem a necessidade de um software terceiro, como por exemplo adicionar uma rotina para apresentar uma projeção futura dos lucros ou produção, ou então um ranking com as melhores máquinas ou produtos tornando ainda mais visível os dados gerados pelas máquinas.

REFERÊNCIAS

- BENTO, Evaldo Junior. **Desenvolvimento Web com PHP e MySQL**. São Paulo: Casa do Código, 2016. 363 p.
- CAELUM. **Desenvolvimento Web com HTML, CSS e JavaScript**. [S. L.]: Caelum Ensino e Inovação, [20--]. 270 p.
- DEV.W3C. **Documento de software**. 2011. Disponível em: <https://dev.w3.org/html5/spec-LC>. Acesso em: 15 abr. 2022.
- ENGINEERING. **Coleta de dados da manufatura**. [20--]. Disponível em: https://www.engusa.com/pt_br/solution/data-collection. Acesso em: 07 out. 2022.
- FOROUZAN, Behrouz A.. **Protocolo TCP/IP**. 3. ed. Porto Alegre: Amgh Editora Ltda, 2010. 864 p.
- INCAU, Caio. **Vue.js construa aplicações incríveis**. São Paulo: Casa do Código, 2017. 175 p.
- TEDESCO, Kennedy. **Aplicações em tempo real com PHP usando WebSockets**. 2020. Disponível em: <https://www.treinaweb.com.br/blog/uma-introducao-a-tcp-udp-e-sockets>. Acesso em: 13 maio 2022.
- TEDESCO, Kennedy. **Uma introdução a TCP, UDP e Sockets**. 2019. Disponível em: <https://www.treinaweb.com.br/blog/uma-introducao-a-tcp-udp-e-sockets>. Acesso em: 10 maio 2022.
- LARAVEL. **Documento de software**. 2022. Disponível em: <https://laravel.com/docs/9.x>. Acesso em: 01 abr. 2022.
- LIMONGI, Alfredo. **HTML a partir do zero**. [S. L.]: Createspace Independent Publishing Platform, 2014. 102 p.
- LOBO, Edson Junio Rodrigues. **Criação de sites em PHP**. São Paulo: Digerati Books, 2007. 126 p.
- ORACLE. **O que é um Banco de Dados?**. 2022. Disponível em: <https://www.oracle.com/br/database/what-is-database/>. Acesso em: 29 mai. 2022.
- POSTGRESQL. **Documento de software**. 2022. Disponível em: <https://www.postgresql.org/about/>. Acesso em: 10 abr. 2022.
- PRESSMAN, Roger S.; MAXIM, Bruce R.. **Engenharia de software** uma abordagem profissional. 8. ed. Porto Alegre: Amgh Editora Ltda, 2016. 940 p.
- SOMMERVILLE, Ian. **Engenharia de software**. 10. ed. São Paulo: Pearson Education do Brasil Ltda, 2019. 757 p.
- TEOREY ET AL. **Projeto e modelagem de banco de dados**. Rio de Janeiro: Elsevier Editora Ltda, 2014. 309 p.
- TRACTIAN. **Gestão de Manutenção e Monitoramento Online para Indústrias de Alimentos e Bebidas**. [20--]. Disponível em: <https://tractian.com/industria/alimenticia>. Acesso em: 10 out. 2022.
- TURINI, Rodrigo. **PHP e Laravel: crie aplicações web como um verdadeiro artesão**. [S.L.]: Casa do Código, 2015.
- VUE.JS. **Documento de software**. 2022. Disponível em: <https://br.vuejs.org/v2/guide/index.html>. Acesso em: 26 jul. 2022.

W3SCHOOLS. **Documento de software.** 2016. Disponível em:
<https://www.w3.org/standards/webdesign/script> Acesso em: 16 abr. 2022.