

FEITEP – EDUCAÇÃO PRESENCIAL E A DISTÂNCIA

ENGENHARIA DA COMPUTAÇÃO

ALLAN CHAVIER FELINTRO

**DESENVOLVIMENTO DE UM SISTEMA DE GERENCIAMENTO DE
ARMAZÉM (*WAREHOUSE MANAGEMENT SYSTEM*)**

**MARINGÁ
2023**

ALLAN CHAVIER FELINTRO

**DESENVOLVIMENTO DE UM SISTEMA DE GERENCIAMENTO DE
ARMAZÉM (*WAREHOUSE MANAGEMENT SYSTEM*)**

Trabalho de Conclusão de Curso apresentado à
FEITEP – Faculdade De Educação Presencial e
a Distância, como parte dos requisitos para
obtenção do título de Bacharel em Engenharia
da Computação.

Orientador(a): Ma. Cláudia Tupan Rosa

MARINGÁ
2023

FEITEP – Educação Presencial e a Distância

ALLAN CHAVIER FELINTRO

**DESENVOLVIMENTO DE UM SISTEMA DE GERENCIAMENTO DE
ARMAZÉM (*WAREHOUSE MANAGEMENT SYSTEM*)**

Trabalho de Conclusão de Curso apresentado à
FEITEP – Faculdade De Ensino Presencial e a
Distância, como parte dos requisitos para
obtenção do título de Bacharel em Engenharia
da Computação.

Orientador(a): Ma. Cláudia Tupan Rosa

BANCA EXAMINADORA

Professor Orientador: Ma. Cláudia Tupan Rosa – FEITEP

Professor: Me. Guilherme Américo Rosa – FEITEP

Professor: Me. Fábio Splendor – FEITEP

Data de Aprovação: _____ de _____ de 2023.

AGRADECIMENTOS

Agradeço a minha companheira Talita, pelo amor, pela dedicação, pelo respeito, pelo apoio incondicional, pelo incentivo, pelos conselhos e pela paciência durante esta etapa tão importante.

Aos meus pais, que mesmo distantes, apoiaram-me, incentivaram-me e oraram por mim.

A minha orientadora, pela paciência, pelas dicas, pelo tempo aplicado na orientação e pela parceria durante toda a produção do trabalho.

A todos os professores que compartilharam um pouco de seu conhecimento comigo e contribuíram para minha formação.

“Mares calmos não formam bons marinheiros”
Autor desconhecido

RESUMO

O uso de sistemas da informação, para registro e controle de processos, é uma realidade amplamente observada no contexto corporativo. O número de empresas que utilizam recursos de *software* cresce, enquanto os tradicionais papel e caneta caem em desuso. Entender as particularidades do negócio, ao qual planeja-se aplicar uma solução de *software* pode ser um desafio complexo. Prontamente, este trabalho propôs o desenvolvimento de um Sistema de Gerenciamento de Armazém (*Warehouse Management System* – WMS), capaz de registrar, exibir e controlar as operações de um centro de distribuição (CD). Para obter o entendimento do negócio, no qual o WMS é aplicado, foi realizado um levantamento bibliográfico abordando o setor logístico, o CD e o WMS em si. Em seguida, abordou-se ferramentas e conceitos tecnológicos utilizados no desenvolvimento do sistema. Tendo alcançado o entendimento necessário, foi elaborado um levantamento de requisitos de *software* para determinar quais operações do CD seriam atendidas pela aplicação. Nesta etapa, também foram definidos, os recursos usados no ambiente de desenvolvimento de *software*. Com as definições elaboradas, iniciou-se, então, a etapa de construção do sistema, produzindo o código-fonte, demonstrando sua estrutura geral e seu modelo de banco de dados. Após sua construção, seu funcionamento básico foi descrito e demonstrado. Por fim, foram realizados testes em ambiente local. Os resultados demonstraram-se positivos, pois o *software* produzido atendeu as operações do CD propostas inicialmente.

Palavras-chave: Sistema de Gerenciamento de Armazém. Logística. Centro de Distribuição. Desenvolvimento de *Software*. Banco de Dados.

ABSTRACT

The use of information systems to record and control processes is a widely observed reality in the enterprise environment. The number of companies using software resources are growing, while traditional pen and paper are falling into disuse. Understanding the details of the business to which you plan to apply a software solution can be a complex challenge. Promptly, this paper proposed the development of a Warehouse Management System (WMS), capable of recording, displaying and controlling warehouse operations. To obtain an understanding of the business in which the WMS is applied, a bibliographic survey was carried out covering the logistics sector, the warehouse and the WMS itself; then, technological tools and concepts used in the development of the system were discussed. Having achieved the necessary understanding, a software requirements gathering was prepared to determine which warehouse operations would be supported by the application. At this stage, the resources used in the software development environment were also defined. With the definitions elaborated, the system construction stage began, producing the source code, demonstrating its general structure and database model. After its construction, its basic functioning was described and demonstrated. Finally, tests were carried out in a local environment. The results were positive, as the software produced met the warehouse operations initially proposed.

Key words: Warehouse Management System. Logistics. Warehouse. Software Development. Database.

SUMÁRIO

1	INTRODUÇÃO	13
1.1	OBJETIVOS	15
1.1.1	Objetivo geral	15
1.1.2	Objetivos específicos	15
2	A LOGÍSTICA E O CENTRO DE DISTRIBUIÇÃO	16
2.1	ATRIBUIÇÕES DA LOGÍSTICA GERAL.....	18
2.2	O CENTRO DE DISTRIBUIÇÃO.....	19
3	WAREHOUSE MANAGEMENT SYSTEM	24
4	BANCO DE DADOS	29
4.1	SISTEMA GERENCIADOR DE BANCO DE DADOS	31
4.2	BANCO DE DADOS RELACIONAL	32
4.3	BANCO DE DADOS NÃO-RELACIONAL	35
5	DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS	38
5.1	FRONT-END.....	39
5.2	BACK-END	42
5.2.1	<i>Application Programming Interface</i>	44
5.3	FRAMEWORK.....	45
6	MATERIAIS E MÉTODOS	47
6.1	DEFINIÇÃO DO TIPO DE PESQUISA.....	47
6.2	DESENVOLVIMENTO DA PESQUISA	48
6.2.1	Recursos e Ambiente de Desenvolvimento.....	49
7	RESULTADOS E DISCUSSÕES	50
7.1	LEVANTAMENTO DE REQUISITOS.....	50
7.2	ESTRUTURA DO SISTEMA	52
7.3	MODELAGEM DO BANCO DE DADOS	56
7.4	FUNCIONAMENTO DA APLICAÇÃO	58

7.5 TESTES EM AMBIENTE LOCAL.....	60
8 CONSIDERAÇÕES FINAIS	69
REFERÊNCIAS.....	71

LISTA DE FIGURAS

Figura 1 – O subsistema armazém e suas relações.....	21
Figura 2 – Fluxo de um produto através do CD.....	22
Figura 3 – Divisão de um CD em ruas.....	25
Figura 4 – Visão dos lados direito e esquerdo de uma rua.	25
Figura 5 – Leitura de um código de barras.....	27
Figura 6 – Diagrama simplificado de um sistema de banco de dados.....	32
Figura 7 – Exemplo de relação ou tabela.	34
Figura 8 – Trecho de instrução SQL para criação de uma tabela.	34
Figura 9 – Dados armazenados em documentos com codificação JSON.....	35
Figura 10 – Exemplo de conjuntos de chave e valor.	36
Figura 11 – Exemplo de armazenamento em grafos.....	36
Figura 12 – Exemplo de código HTML.	40
Figura 13 – Formulário de <i>login</i>	40
Figura 14 – Exemplo de estilos escritos em CSS.....	41
Figura 15 – Formulário de <i>login</i> após aplicação do CSS.....	41
Figura 16 – Relação entre o <i>Front-end</i> e o <i>Back-end</i>	43
Figura 17 – Exemplo de código feito em <i>Java</i>	44
Figura 18 – Fluxo de funcionamento de uma API.....	45
Figura 19 – Modelo Espiral de desenvolvimento de software.....	48
Figura 20 – Diagrama de casos de uso.....	51
Figura 21 – Arquitetura geral do sistema.....	52
Figura 22 – Estrutura do cadastro de Produto.....	53
Figura 23 – Estrutura da operação Movimentação.....	54
Figura 24 – Classe de modelo da entidade Produto.....	55
Figura 25 – Geração do banco de dados pelo <i>Hibernate</i>	55
Figura 26 – Modelagem do banco de dados.	56
Figura 27 – Página de cadastro de endereço.....	58
Figura 28 – Página de movimentação de <i>packs</i>	59
Figura 29 – Corpo padrão da requisição de registro de pedidos.	60
Figura 30 – Tela de boas-vindas do sistema.....	61
Figura 31 – Preenchimento do cadastro de empresa.....	61

Figura 32 – Empresa registrada no banco de dados.....	62
Figura 33 – Início da alteração da empresa.	62
Figura 34 – Alteração realizada no cadastro da empresa.	63
Figura 35 – Tabela endereço antes da movimentação.....	64
Figura 36 – Início da operação movimentação.....	64
Figura 37 – Tabela endereço após a movimentação.....	65
Figura 38 – Movimentação efetuada.	65
Figura 39 – Seleção da operação de recebimento.	66
Figura 40 – Tela de seleção de pedido.	66
Figura 41 – Tela de recebimento de produtos.	67
Figura 42 – Produtos recebidos.....	67
Figura 43 – Retorno para a tela seleção de pedidos.	68

LISTA DE QUADROS

Quadro 1 – O banco de dados de uma adega de vinhos.	30
Quadro 2 – Levantamento de requisitos de software.	50

LISTA DE SÍMBOLOS E ABREVIACÕES

API	<i>Application Programming Interface</i>
ASLOG	Associação Brasileira de Logística
CD	Centro de Distribuição
CNPJ	Cadastro Nacional da Pessoa Jurídica
CSS	<i>Cascading Style Sheets</i>
DTO	<i>Data Transfer Object</i>
DUN	<i>Distribution Unit Number</i>
EDI	<i>Electronic Data Interchange</i>
EAN	<i>European Article Number</i>
HTML	<i>HyperText Markup Language</i>
IBM	<i>International Business Machines</i>
JDK	<i>Java Development Kit</i>
JPA	<i>Java Persistence API</i>
JRE	<i>Java Runtime Environment</i>
JSON	<i>Javascript Object Notation</i>
JVM	<i>Java Virtual Machine</i>
MVC	Modelo-Visão-Controlador
NoSQL	<i>Not Only SQL</i>
QR	<i>Quick Response</i>
RFID	<i>Radio Frequency Identification</i>
SGBD	Sistema Gerenciador de Banco de Dados
SQL	<i>Structured Query Language</i>
WMS	<i>Warehouse Management System</i>
WCS	<i>Warehouse Control Systems</i>
XML	<i>Extensible Markup Language</i>

1 INTRODUÇÃO

A tecnologia, cada vez mais presente, promove melhorias significativas em processos empresariais, tornando-os cada vez mais rápidos e menos propensos a erros. Agilizar processos e registrar suas informações, controlar o sistema de produção em fábricas e armazéns, fornecer dados em tempo real aos executivos são exemplos de melhorias geradas pelo uso da tecnologia em favor da organização (NOGUEIRA, 2018).

Segundo Nazário (1999), no passado, o controle das operações em armazéns era feito de maneira rudimentar, com papel e caneta. A cada nova operação, um registro era realizado em fichas, tornando o trabalho e a passagem de informações lentos e propensos a inadequações.

A entrega de um produto ao cliente no prazo acordado é uma dificuldade comum no Brasil, onde com frequência um prazo é firmado com o vendedor e não cumprido, devido às lacunas em seus sistemas de informação e em suas operações (NOVAES, 2007).

De acordo com Barros (2005b), com a tecnologia sendo bem utilizada, operações que antes eram morosas, ganham rapidez e precisão e, conseqüentemente, tornam-se mais confiáveis. Além disso, gera-se o ganho da possibilidade de rápidas tomadas de decisão e de melhorias no nível de serviço.

Dentre várias tecnologias que agregam melhorias nas operações logísticas, Nunes e Platt (2008, *apud* PLATT, 2015) mencionam: código de barras, EDI – *Electronic Data Interchange* ou Intercâmbio Eletrônico de Dados, WMS – *Warehouse Management System* ou Sistema de Gerenciamento de Armazém.

Os sistemas WMS fornecem recursos eletrônicos, os quais conectam e controlam as operações básicas de um armazém desde a entrada de produtos, até sua entrega ao cliente final (BALLOU, 2006).

Segundo Platt (2015), o WMS, é uma tecnologia avançada que permite aprimorar a organização e as operações em armazéns, trazendo benefícios como aumento da capacidade produtiva e redução do tempo gasto nas operações básicas da instalação.

Este sistema, manipula dados dos produtos trabalhados e dos colaboradores ativos na operação, direcionando pessoas e máquinas, de acordo com a demanda de

pedidos, e controla todo este fluxo de ponta a ponta. Ainda, pode ser integrado com outros sistemas relacionados com a operação (PLATT, 2015).

Neste âmbito, o objeto deste estudo foi o desenvolvimento de um sistema do tipo WMS, que seja capaz de controlar as operações básicas de um armazém logístico. Vislumbra-se que ele seja capaz de monitorar, controlar e distribuir tarefas, manter cadastro de produtos e registros das operações em banco de dados, receber pedidos e os disparar para a produção.

A seguir, será apresentada a divisão deste trabalho.

No segundo capítulo, será introduzido o conceito de logística na sua concepção geral, descreve o panorama histórico, o cenário atual e suas atribuições como um setor empresarial. Na sequência, aborda-se os tipos de instalações relacionadas ao processo logístico, dando ênfase no centro de distribuição, no qual o sistema WMS tem papel fundamental.

No terceiro capítulo, conceitua-se o sistema WMS, abordando seu panorama histórico, conceitos gerais, sua função, suas integrações e os benefícios de utilizá-lo.

No quarto capítulo, será descrita a ferramenta banco de dados, com seu panorama histórico, seu funcionamento e suas características importantes. Esta ferramenta, teve papel imprescindível no desenvolvimento e funcionamento do sistema produzido nesta pesquisa.

No quinto capítulo, aborda-se brevemente, o assunto *software*, explora-se as tecnologias aplicadas em seu desenvolvimento no meio corporativo e descreve-se as camadas nas quais uma aplicação comercial é dividida.

No sexto capítulo, descreve-se a metodologia empregada na produção desta pesquisa, os materiais utilizados, e o passo a passo da produção do objeto de estudo.

No sétimo capítulo, apresenta-se os resultados da pesquisa, incluindo o levantamento de requisitos de *software*, a modelagem, a descrição da estrutura e das funcionalidades do sistema, bem como a interface de usuário e seu funcionamento.

Por fim, no oitavo capítulo, apresenta-se o que foi concluído a partir da produção da pesquisa, apresentando informações a respeito dos testes feitos no *software*. Ao final do capítulo, foi sugerido melhorias futuras para aperfeiçoamento do sistema.

1.1 OBJETIVOS

1.1.1 Objetivo geral

Desenvolver um sistema WMS capaz de registrar, exibir e controlar as operações dentro de um armazém.

1.1.2 Objetivos específicos

- Elaborar revisão bibliográfica sobre as operações logísticas que o sistema produzido irá contemplar e as tecnologias empregadas em seu desenvolvimento.
- Realizar levantamento de requisitos de *software* e seus casos de uso.
- Projetar a arquitetura do sistema.
- Modelar as entidades do banco de dados.
- Desenvolver o código-fonte da aplicação que contém as regras de negócio (*back-end*).
- Desenvolver o código da interface de usuário do sistema (*front-end*).
- Realizar testes de funcionamento da aplicação em ambiente local.

2 A LOGÍSTICA E O CENTRO DE DISTRIBUIÇÃO

Em seu início, a concepção de logística estava intimamente conectada à movimentos militares. Surgindo em torno de 1670, no início de uma nova estruturação do exército francês, a logística era responsável por planejar, transportar, armazenar suprimentos e abastecer as tropas francesas. O objetivo era fornecer os suprimentos necessários (como armas, ferramentas, pontes, alimentos, entre outros) para que as tropas avançassem sem impedimentos e com força total. Este termo foi usado então, para caracterizar o conhecimento de movimentação e do suprimento de forças militares (CHIAVENATO, 2014).

Para Ching (2010), durante a Segunda Guerra Mundial, a ideia de logística foi muito empregada pelo Exército Americano, a fim de atender os propósitos militares nos conflitos que ocorriam na época, englobando todo o processo de provimento de insumos durante as operações, desde a compra de materiais, até a entrega dos suprimentos às tropas.

De acordo com Novaes (2007), seguindo uma determinada tática, líderes militares, os quais comandavam o progresso de tropas, precisavam que suprimentos como: equipamento bélico, alimentos e atendimento médico fossem entregues no local e no tempo adequado, para isto, era necessário ter sob seu comando, um esquadrão designado a providenciar a entrega dos suprimentos no campo de batalha no momento certo. Estes eram os esquadrões militares logísticos, áreas de suporte que operavam sem o prestígio das campanhas, discretamente na retaguarda possibilitando o mantimento das tropas no campo de batalha.

Em paralelo aos acontecimentos do período militar, ocorreram operações semelhantes às do exército no ramo empresarial. Organizações precisavam enviar seus produtos aos armazéns ou ao comércio que atendia seus clientes; necessitavam manter estoque de matéria-prima para que sua produção procedesse, e devido divergências entre pedidos e produção, precisavam manter em estoque, bens concluídos (NOVAES, 2007).

Iniciou-se, então, o desenvolvimento da logística nas organizações empresariais, com o surgimento de novos processos de armazenagem e distribuição de itens aos clientes como características do fluxo de insumos dentro e fora da organização (CHIAVENATO, 2014).

Para Novaes (2007), no entendimento dos executivos, estas atividades eram fundamentais, mas não agregavam valor ao seu produto e a divisão logística em si era apontada apenas como uma despesa para a organização, sem possibilidade de trazer ganhos estratégicos.

De acordo com Ballou (2013), o início de muitas noções logísticas ocorreu durante a Segunda Guerra Mundial. Em 1945, o armazenamento de mercadorias e o transporte, já eram atividades presentes nas empresas e lideradas por um único gestor. Nessa perspectiva, empresas do ramo de alimentos iniciaram o movimento, porém até a década de 50, não haviam estudos concisos para orientar o avanço da área.

Nos anos após a Segunda Guerra Mundial, segundo Novaes (2007), observou-se um desenvolvimento ininterrupto da área de logística. A introdução de novas técnicas e suporte tecnológico, que surgiram da necessidade de melhorias dos resultados organizacionais, por exemplo: o tempo de entrega, abastecimento de produtos no mercado, a redução de despesas entre outros; trouxeram grandes avanços no que se refere a logística (CHIAVENATO, 2014).

A logística, na sua origem, foi equivocadamente caracterizada e limitada apenas ao armazenamento e transporte de produtos, todavia, atualmente ela é apontada como área estratégica de empresas em suas cadeias produtivas, englobando várias atividades e gerando grandes vantagens competitivas para as organizações (NOVAES, 2007).

Chiavenato (2014) descreve que, recentemente, a logística é a área que administra instalações empresariais como armazéns e gerencia atividades como: armazenagem, distribuição, balanços de estoque, deslocamento de materiais no interior de fábricas e a conclusão do pedido ao cliente final.

Conforme mencionado por Ballou (2013), dentre as caracterizações dadas à logística no ramo empresarial, pode-se destacar:

A logística empresarial trata de todas atividades de movimentação e armazenagem que facilitam o fluxo de produtos desde o ponto de aquisição da matéria-prima até o ponto de consumo final, assim como dos fluxos de informação que colocam os produtos em movimento, com o propósito de providenciar níveis de serviço adequados aos clientes a um custo razoável (BALLOU, 2013, p. 24).

2.1 ATRIBUIÇÕES DA LOGÍSTICA GERAL

Ballou (2013) categoriza as atividades básicas da logística a nível de importância, dividindo-as em atividades primárias (Transportes, Manutenção de estoques e Processamento de pedidos) e atividades secundárias (Armazenagem, Manuseio de materiais, Embalagem de proteção, Obtenção, Programação do produto e Manutenção de informação). Esta divisão origina-se da combinação de dois fatores, a proporção do custo da atividade em relação ao custo logístico total e ao nível de contribuição para o cumprimento das metas da logística.

Existe uma relação das atividades primárias com as atividades secundárias, onde atividades secundárias são dispostas sob o manto das primárias, pois ao serem executadas, contribuem diretamente com as primárias. As relações são as seguintes: Transportes engloba as atividades secundárias de Manuseio de materiais e Embalagem, Processamento de Pedidos engloba Obtenção e Armazenagem e Manutenção de Estoques que engloba Manutenção de informações e Programação do Produto (BALLOU, 2013).

Algumas das atividades primárias e secundárias serão descritas a seguir, pois farão parte das funcionalidades do sistema a ser implementado pela proposta dessa pesquisa.

Processamento de pedidos: engloba uma série de atividades, nas quais observa-se variações no tempo total de execução e nas atividades que a compõe dependendo do tipo de pedido tratado. A exemplo, um pedido aberto a uma indústria, terá atividades e tempo diferentes de um pedido feito a um varejista. Algumas das atividades que a compõem são: emissão do pedido pelo cliente, recebimento do pedido pelo fornecedor, entrada do pedido na produção, atendimento do pedido e relatório (BALLOU, 2006).

Manutenção dos estoques: está relacionada com atividades que permitem aumentar a disponibilidade dos produtos frente aos pedidos (BALLOU, 2013).

Armazenagem: trabalha a gestão do armazém de forma geral, o qual pode englobar atividades como cálculo e dimensionamento do armazém, determinação do leiaute do estoque, parametrização do depósito (BALLOU, 2006).

Manuseio de materiais: está relacionada com a Armazenagem e dá suporte a Manutenção dos estoques, refere-se à movimentação dentro do armazém, na qual

conglomera atividades como: a transferência do produto da área de recebimento à área de armazenagem e, desta última até a área de expedição ou carregamento (BALLOU, 2013).

Programação do produto: trata-se do ponto inicial da distribuição do produto, em que se define a quantidade do produto a ser entregue ao cliente e o local de produção (BALLOU, 2013).

Manutenção de informações: apoia a boa administração fornecendo informações fundamentais aos gestores, ou seja, registros sobre as operações em geral, cadastro de clientes, posição dos estoques, quantidade de vendas, entre outras (BALLOU, 2013).

2.2 O CENTRO DE DISTRIBUIÇÃO

Independentemente de o produtor ou a fábrica estarem localizados há 10 ou 2.500 quilômetros de distância de um comércio, um cliente que o procura espera que seu produto esteja disponível para compra, sendo um dos papéis principais da logística, isto é, o fornecimento de meios para que o produto chegue às prateleiras quando for preciso. Em outras palavras, este papel também pode ser descrito como a condução da movimentação de produtos oriundos de muitos produtores, cada vez mais espalhados pelo país de operação e também pelo mundo, com destino aos consumidores finais em regiões cada vez mais afastadas (LACERDA, 2000).

Ainda, para o autor, fornecer os melhores resultados no que se refere a possibilitar o acesso de produtos em estoque e ao seu respectivo tempo de entrega, faz parte do escopo da estruturação de sistemas que possibilitem o abastecimento de prateleiras localizadas em áreas afastadas do produtor ou da fábrica, e realizar tal estruturação com o menor custo possível é uma das premissas básicas do processo logístico. Com base neste cenário, o foco do processo é direcionado às instalações destinadas à armazenagem de produtos e também as formas que elas colaboram para a melhora dos resultados propostos em relação ao nível de serviço.

Segundo Alvarenga e Novaes (2000), tipos comuns de instalações são: o armazém (ou depósito), cujo objetivo é a estocagem e a remessa de produtos oriundos de uma planta industrial e o CD (Centro de Distribuição), que de acordo com Lacerda

(2000), é destinado a atender rapidamente às demandas de determinada área geográfica afastada de grandes regiões produtoras.

A ASLOG (Associação Brasileira de Logística) define um CD como sendo um depósito responsável por gerir os estoques de mercadorias e a sua distribuição física (BECKEDORFF, 2013).

Para Alvarenga e Novaes (2000), as principais funções do armazém são: Armazenagem, que consiste em estocar produtos diversos por tempo e quantidade que variam de acordo com a necessidade e estratégia da empresa (premissa básica do armazém); Consolidação, que consiste em unir ou agregar frações de cargas que foram previamente recebidas de diversos fornecedores e armazenadas no estoque e consolidá-las em um grande carregamento seguindo quantidades e sortimento de acordo com o pedido; Desconsolidação que pode ser descrita como o inverso da consolidação, na qual grandes cargas são recebidas e divididas em pequenas cargas para serem distribuídas para diversos destinos.

Executando as funções descritas anteriormente, o armazém se torna peça fundamental da cadeia logística empresarial, sendo ele por si só, um sistema ou subsistema dentro do sistema logístico global da organização, sendo interessante designar diretamente as funções e divisões deste subsistema com base no seu papel dentro do sistema global mencionado (ALVARENGA; NOVAES, 2000).

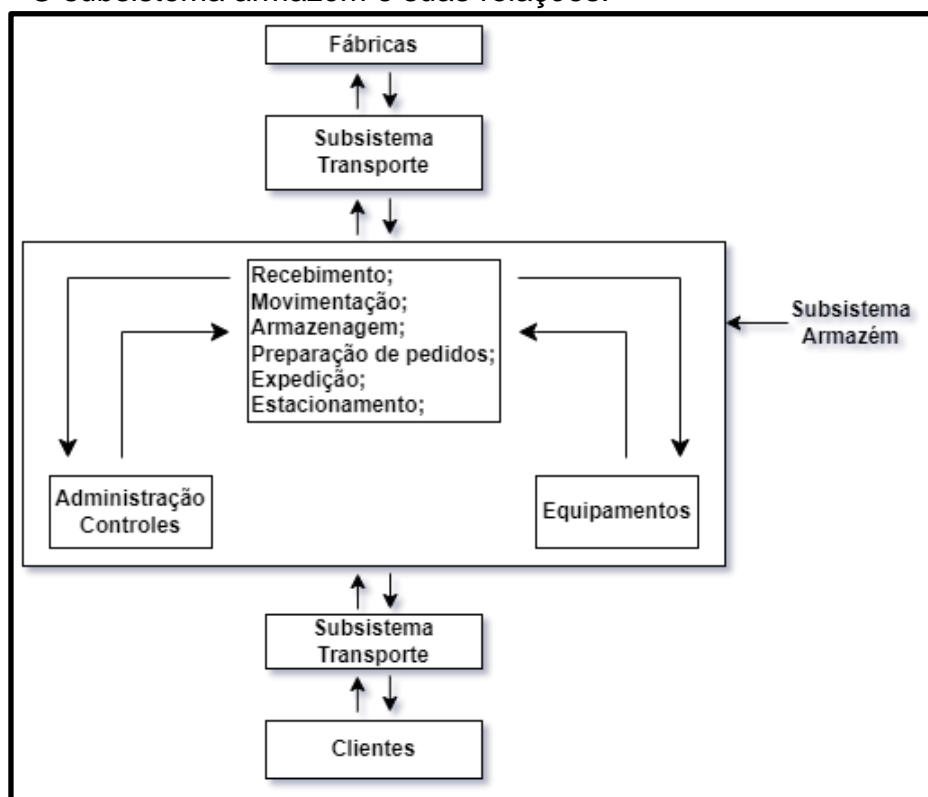
Os autores ainda descrevem, que são partes integrantes do subsistema os setores a seguir:

- Recebimento: fornecedores enviam suas cargas com destino ao armazém, onde a área de recebimento é responsável por descarregar a mercadoria, realizar a conferência da carga (pedido x mercadoria) e enviar ao local de estocagem;
- Movimentação: depois de recebidas, as mercadorias precisam ser movimentadas até seu respectivo local de armazenagem para serem estocadas. Posteriormente, elas podem ser movimentadas de seu local de armazenagem até um local designado para a consolidação dos pedidos. Este processo é nomeado de movimentação;
- Armazenagem: componente do subsistema designado à estocagem das mercadorias em diferentes estágios do processo logístico. Nesta etapa, as mercadorias podem ficar estocadas por períodos variados. Assim, cada período pode variar, de acordo com a estratégia adotada pela empresa;

- Preparação dos pedidos: nesta parte do subsistema, designa-se um local apropriado para a preparação dos pedidos do cliente (seja ele o consumidor ou o ponto de venda), onde os produtos são trazidos de seus locais de estocagem, acondicionados em invólucros como contêineres, caixas, paletes entre outros. E este invólucro é identificado, externamente, com os dados do cliente;
- Embarque ou Expedição: após a finalização da preparação do pedido, a mercadoria estará pronta para ser carregada no veículo destinado a transportá-la até o cliente, esta tarefa é de responsabilidade do componente Expedição;
- Circulação externa e estacionamento: Importante componente do subsistema, o qual destina-se espaço físico voltado a carga e descarga para que esses processos não sejam feitos em vias públicas como em alguns casos de empresas que não dispõe deste componente.

A Figura 1 ilustra o esquema do subsistema armazém e suas relações com outros subsistemas.

Figura 1 – O subsistema armazém e suas relações.



Fonte: Adaptado de Alvarenga e Novaes (2000).

Vários são os fatores que levam uma empresa a dispor capital para a construção e implementação de um armazém ou CD em seu sistema logístico. Alguns

desses fatores estão diretamente relacionados à busca constante pela melhoria do nível de serviço e da competitividade das organizações (LACERDA, 2000). Em suma, a implementação de um CD ao longo de um sistema logístico empresarial, se torna importante, pois o mesmo funcionará como centro de fornecimento intermediário entre os fornecedores e os consumidores finais, isso traz melhoria direta no tempo de reposição de produtos (BARROS, 2005b).

Segundo Lacerda (2000), o CD além de melhorar o tempo de atendimento de determinadas demandas dos consumidores, também contribui para a redução dos custos logísticos, já que atuam como consolidadores de cargas. Ao invés do atendimento a vários clientes ser feito diretamente por uma unidade produtora, resultando em veículos percorrendo grandes distâncias com cargas fracionadas, as entregas poderiam ser realizadas em um CD com grandes quantidades em amplos carregamentos consolidados e a entrega ao cliente ser feita pelo CD de maneira fracionada, com menor distância percorrida e, conseqüentemente, menor custo logístico para o sistema.

No CD, grandes carregamentos são recebidos diariamente, de diversos fornecedores. Depois, são armazenados e separados, nas quantidades e no sortimento (variedade) que cada ponto de venda necessita ou pretende receber. Na sequência, é consolidada em novos carregamentos e expedidos ao respectivo ponto de venda. A Figura 2 descreve este fluxo (BECKEDORFF, 2013).

Figura 2 – Fluxo de um produto através do CD.



Fonte: Adaptado de Beckedorff (2013).

Segundo Bowersox *et al.* (2014), alguns autores fazem distinção entre os termos armazém e CD, entretanto, existem muitas semelhanças e poucas diferenças em suas funções. Portanto, nesta pesquisa, o termo CD será adotado para descrever a

instalação do sistema logístico, pois, é o termo que mais se adequa às operações que serão atendidas pelo *software* proposto.

No intuito de controlar todo o fluxo dentro de uma planta do sistema logístico, pode-se utilizar o sistema WMS, que de acordo com Ballou (2006), fornece recursos eletrônicos relevantes para a gestão e controle dos processos logísticos. Este sistema será o tema do próximo capítulo.

3 WAREHOUSE MANAGEMENT SYSTEM

Para Barros (2005a), O sistema WMS originou-se da necessidade de otimizar o trânsito de informações e a movimentação de produtos no interior de um CD, é um recurso de gerenciamento, o qual permite o controle e execução de operações como: Recebimento, Armazenagem, Separação e Expedição de produtos. É classificado como boa opção para melhorar atividades dentro de um CD, uma vez que, também possui formas de melhorar a ocupação e uso do espaço físico, e coordenar a movimentação e a distribuição dos produtos e também um rápido e preciso fluxo de informações (BARROS, 2005a).

Beckedorff (2013) menciona que o surgimento do sistema WMS fornece ganhos de rendimento nos processos do CD, direcionando atividades, conforme a disposição da mão de obra, ou seja, de acordo com colaboradores disponíveis no momento para executá-las. O sistema é capaz de localizar um colaborador dentro da planta e direcioná-lo ao local da atividade, gerando um ganho na produtividade quando diferentes atividades são alternadas e direcionadas aos demais colaboradores.

A aplicação do WMS iniciou com base nos antigos sistemas WCS – *Warehouse Control Systems* (Sistemas de Controle de Armazéns), que até meados da década de 1970 eram capazes apenas de gerenciar transações de entradas e saídas de estoque. Iniciou-se então, o surgimento dos primeiros sistemas gerenciadores de endereçamento, que possibilitaram a localização de materiais dentro do CD por meio de endereços. Os produtos, dessa forma, deixaram de ter um local de armazenagem fixo e ganharam dinamicidade em relação à Armazenagem, podendo, assim, serem armazenados em locais diferentes, conforme a necessidade e disponibilidade de espaço (BARROS, 2005a).

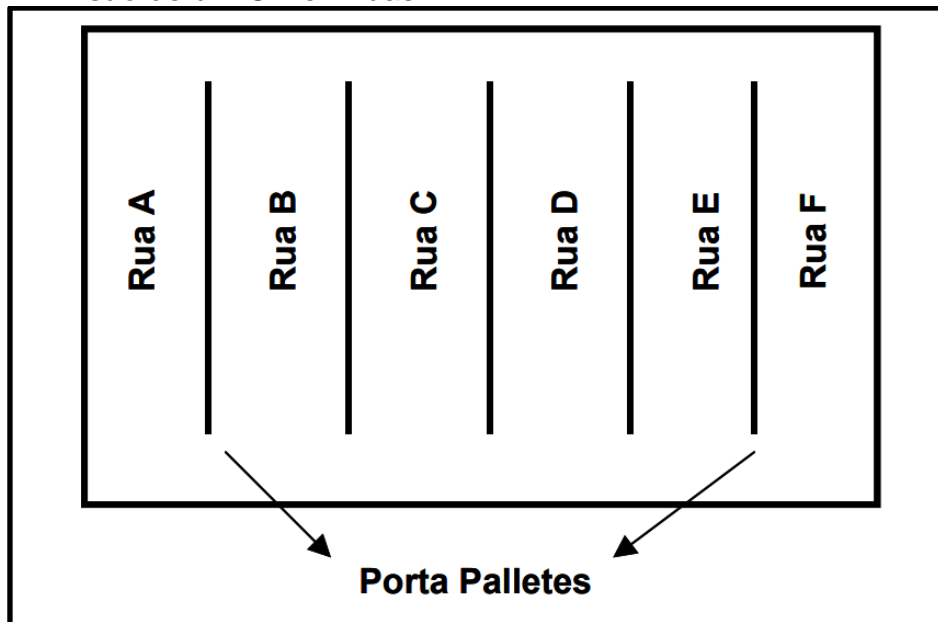
Nogueira (2018) descreve que, no intuito de otimizar o espaço físico e gerenciar a entrada ou saída de novos produtos em estoque, recomenda-se utilizar um esquema de endereçamento com alternância numérica, por exemplo: A1-001, A1-003, A1-005, e assim por diante.

A designação do esquema de endereçamento está diretamente relacionada à forma de organização dos materiais armazenados, sendo indispensável a

determinação do leiaute, pois baseando-se no leiaute pode-se definir o melhor esquema de endereçamento (BARROS, 2005b).

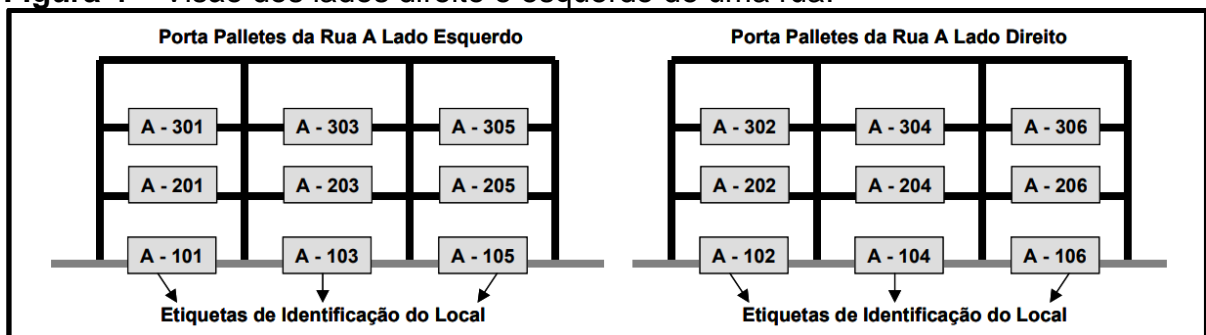
Um esquema de endereçamento muito presente em CDs é a divisão da planta em “ruas” nomeadas com letras, as quais possuem níveis de estocagem numerados que comportam paletes. A numeração segue um padrão ímpar, do lado esquerdo, e par, do lado direito, da “rua”. Desse modo, cada endereço (ou apartamento) recebe um número, de acordo com a sua posição na rua e o andar ocupado, semelhante a um prédio com apartamentos (101, 201 e assim por diante). As três coordenadas (rua, número e altura) compõem o esquema de endereçamento do CD. As figuras 3 e 4 ilustram o esquema descrito (FRANKLIN, 2003 apud BARROS, 2005b).

Figura 3 – Divisão de um CD em ruas.



Fonte: Barros (2005b).

Figura 4 – Visão dos lados direito e esquerdo de uma rua.



Fonte: Barros (2005b).

Segundo Delage (2023), o sistema WMS trabalha em três grandes frentes, sendo elas: Controle de estoque, Execução das atividades e Fornecimento de informações. Esse sistema controla todas as pendências operacionais fornecendo otimização referente ao uso de recursos, e provê interfaces com dados e notificações que ocorrem em tempo real, de maneira que os líderes de operações possam tomar decisões e se adiantar mediante a possíveis falhas. Com a utilização deste *software*, todo o fluxo de mercadorias se torna rastreável, ordenado e com rígida verificação, sendo estas, características que evitam perdas e erros operacionais.

O sistema também opera integrando *hardware*, *software* e periféricos para coordenar o ambiente, as operações, o uso de equipamentos e a mão de obra dos CDs. Ademais, fornece diversas funcionalidades para a organização, como: prover dados em tempo real, fornecer o plano de recebimento de pedidos, prover interface a uma série de operações do CD – separação dos pedidos, conferência dos produtos na entrada e na saída, auditorias pontuais e realização de inventários –, gerenciar o equipamento de movimentação, gerar relatórios operacionais e gerenciais. Alguns benefícios de utilizá-lo são: otimização da produtividade, melhora da ocupação do espaço físico, redução no tempo de separação dos pedidos, aumento da acurácia das operações em geral (NOGUEIRA, 2018).

Segundo Finco (2020), os sistemas do tipo WMS, usualmente são concebidos para melhorar o gerenciamento e automatização dos processos logísticos, principalmente nas operações de CDs. Estes sistemas, quando aplicados na operação logística, otimizam processos de estocagem, expedição, gestão de estoques, separação de pedidos (ou *picking*) e, conseqüentemente, reduz custos e tempo de execução.

Outras melhorias também são geradas ao utilizar um sistema WMS, por exemplo: melhor gestão operacional (o sistema informa as tarefas pendentes), reduz o tempo ocioso da mão de obra, melhora o caminho percorrido pela equipe de separação, torna o estoque mais denso reduzindo a distâncias percorridas (BARROS, 2005a).

O *software* também atua no controle de prazo de validade de produtos, organização da paletização de produtos, etiquetagem, integração com outros sistemas para trânsito de dados. O sistema atua com informações coletadas a partir de tecnologias de rastreamento como código de barras, RFID – *Radio Frequency*

Identification (Identificação Por Radiofrequência) e código QR – *Quick Response* (Resposta Rápida ou QR Code) (EQUIPE TOTVS, 2022).

Muito utilizado pelas organizações, o código de barras opera para agilizar o lançamento de informações de operações com produtos físicos em *softwares* gerenciais. Possibilita a redução de margem de erro em relação ao registro manual de informações e também uma coleta imediata de dados de um produto. Ainda pode se integrar aos sistemas gerenciais próprios e também se conectar aos sistemas de outras empresas pertencentes a cadeia de suprimentos (PLATT, 2015).

A utilização desta tecnologia requer equipamento específico para impressão de etiquetas e sua leitura. Sua aplicação possui padronização mundial para cada tipo de produto ou objetivo de identificação. Alguns exemplos: EAN-13 (*European Article Number*), utilizados para produtos destinados ao consumidor final; DUN-14 (*Distribution Unit Number*), impressos ou etiquetados nas embalagens que contém várias unidades dos produtos oriundas das unidades produtoras; EAN-128, aplicados em paletes no interior de depósitos (BECKEDORFF, 2013). A Figura 5 apresenta a leitura de um código de barras, o código lido pelo periférico é convertido nos seus respectivos caracteres e é inserido no sistema gerencial.

Figura 5 – Leitura de um código de barras.



Fonte: GS1 Brasil (2016).

Levando em consideração o contexto da utilização de *software* para gestão de operações em um centro de distribuição, a aplicação de um banco de dados em conjunto com o sistema utilizado se faz necessário, pois de acordo com Pinto (2022), fornece recursos para o armazenamento de informações e gestão organizada dos dados.

4 BANCO DE DADOS

Sendo considerado peça fundamental na vida cotidiana da sociedade, o banco de dados é utilizado em inúmeras atividades humanas do dia a dia. Realizar alguma transação bancária, alugar um quarto, comprar uma passagem de avião, registrar um evento em agenda, são exemplos de interações que o ser humano tem com o banco de dados. Em todos esses exemplos, um sistema computadorizado acessa um banco de dados e apresenta seu conteúdo ao usuário (ELMASRI; NAVATHE, 2011).

De acordo com Ramakrishnan E Gehrke (2011), o volume de dados e informações disponibilizados às pessoas é enorme, e a ideia de que esses dados podem agregar valor organizacional é largamente validada. No intuito de coletar grandes quantidades de dados, os usuários precisam de recursos que forneçam uma forma adequada de coletar, gerenciar e organizar essas informações. Seria contraproducente seguir no caminho oposto, e, neste contexto, o que poderia ser considerado um ativo organizacional acaba se tornando um passivo, no qual o preço para se obter e se gerenciar ultrapassa muito o valor por ele agregado.

O banco de dados é um conjunto de informações que possui relevância para a organização que o possui. É um programa de computador cuja premissa é fornecer interface para depositar e resgatar informações de forma rápida, organizada e vantajosa. Esses sistemas são concebidos para se armazenar e gerenciar grandes e complexos conjuntos de dados (SILBERSCHATZ; KORTH; SUDARSHAN, 2012).

Para Date (2003), o banco de dados corresponde a um grande armário de arquivos eletrônico, no qual os usuários de uma aplicação podem enviar comandos para que a aplicação requirite que o banco de dados realize operações com os dados existentes, como: buscar, excluir, alterar e também incluir novos dados ao seu conjunto. O Quadro 1 representa um exemplo de estrutura gerada e gerenciada pelo banco de dados, a tabela, sendo organizada em colunas (representam atributos) e linhas (representam registros).

Dep	Vinho	Produtor	Ano	Garrafas	Pronto
2	Chardonnay	Buena Vista	2001	1	2003
3	Chardonnay	Geyser Peak	2001	5	2003
6	Chardonnay	Simi	2000	4	2002
12	Joh. Riesling	Jekel	2002	1	2003
21	Fumé Blanc	Ch. St. Jean	2001	4	2003
22	Fumé Blanc	Robt. Mondavi	2000	2	2002
30	Gewürztraminer	Ch. St. Jean	2002	3	2003
43	Cab. Sauvignon	Windsor	1995	12	2004
45	Cab. Sauvignon	Geyser Peak	1998	12	2006
48	Cab. Sauvignon	Robt. Mondavi	1997	12	2008
50	Pinot Noir	Gary Farrell	2000	3	2003
51	Pinot Noir	Fetzer	1997	3	2004
52	Pinot Noir	Dehlinger	1999	2	2002
58	Merlot	Clos du Bois	1998	9	2004
64	Zinfandel	Cline	1998	9	2007
72	Zinfandel	Rafanelli	1999	2	2007

Quadro 1 – O banco de dados de uma adega de vinhos.

Fonte: Date (2013).

Sendo o principal responsável pelo armazenamento e gerenciamento de dados, o banco de dados possui aplicação em diversas áreas, como: vendas (para dados de consumidores, itens e compras), recursos humanos (para dados de colaboradores, folha de pagamento, benefícios e remuneração), bancos (para dados de clientes, transações, serviços e operações com cartão de crédito), instituições de ensino (para dados de alunos, cursos, notas e matrículas) (SILBERSCHATZ; KORTH; SUDARSHAN, 2012).

Elmasri e Navathe (2011) afirma que, um banco de dados reproduz um conjunto de aspectos do mundo real, em que as mudanças, nesses aspectos, refletem diretamente nas informações do banco. Para ser conciso e seguro, ele precisa ser a imagem fiel do conjunto no qual representa, ou seja, toda modificação no conjunto do mundo real precisa ser refletida no banco de dados o mais rápido possível.

O banco de dados não se limita apenas ao armazenamento de dados, mas também atua no gerenciamento de toda a estrutura que comporta essas informações. Esta gestão cobre desde a configuração das estruturas nas quais os dados serão registrados e nas regras e recursos que irão prevenir a manipulação indevida e garantir a segurança das informações no intuito de manter os dados íntegros e precisos (SILBERSCHATZ; KORTH; SUDARSHAN, 2012).

4.1 SISTEMA GERENCIADOR DE BANCO DE DADOS

Date (2013) descreve que, em um sistema de banco de dados, há um componente importante, o qual atua como uma ponte entre o usuário do sistema e o banco de dados propriamente dito, ele é denominado SGBD (Sistema de Gerenciamento de Banco de Dados).

O SGBD pode ser definido como um conjunto de programas de computador que fornece ferramentas e interface ao usuário que o permite conceber e conservar o banco de dados, possibilitando o usuário definir, manipular, construir, e compartilhar as informações com outros usuários e aplicações (ELMASRI; NAVATHE, 2011).

Para Silberschatz, Korth e Sudarshan (2012), este recurso é um conjunto de dados correlacionados e um kit de *software* que possibilita o acesso a esses dados. Ramakrishnan e Gehrke (2011) afirmam que é uma coleção de *softwares* concebidos para facilitar a execução de tarefas que sem ele seriam complexas, tais como: fornecer proteção contra manipulações concorrentes, garantir restauração dos dados a um estado sólido em caso de falha do sistema.

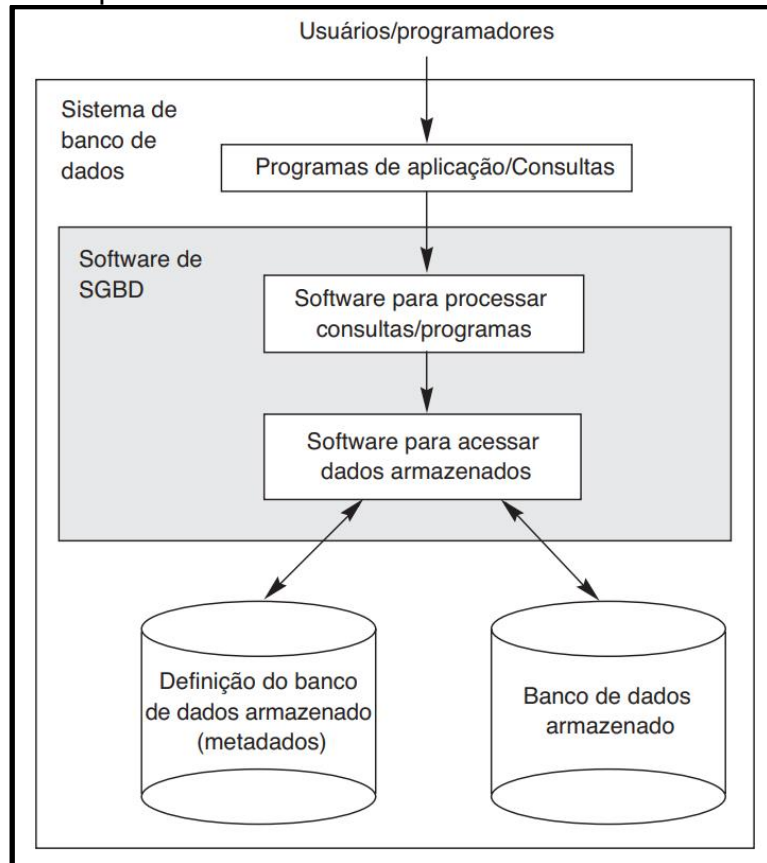
O SGBD fornece importantes funcionalidades ao sistema de banco de dados, como: proteção contra falhas de *hardware* e *software*, controle de acessos não validados e maliciosos. Araujo et al. (2023) complementa que ele fornece um conjunto de segurança que impede acessos não permitidos ao banco de dados e possui procedimentos de *backup* (cópias de segurança) e recuperação em situações de falhas. O conjunto de segurança é formado por normas, as quais mapeiam os usuários que podem acessar os dados e limitam quais deles cada indivíduo pode acessar (ELMASRI; NAVATHE, 2011).

De acordo com Date (2003), quaisquer comandos dados ao banco de dados são previamente validados pelo SGBD, requisições para inserir ou remover registros, recuperar e atualizar dados previamente armazenados são algumas das funcionalidades fornecidas e validadas pelo *software*. Ser uma camada de isolamento entre o detalhamento do banco de dados e o usuário é, portanto, uma responsabilidade do SGBD.

A Figura 6 descreve um sistema de banco de dados, na qual observa-se a divisão em camadas. Usuários acessam os dados do banco, por meio de aplicações,

que se comunicam com o banco, de dados de acordo com a gestão e validação do SGBD (ELMASRI; NAVATHE, 2011).

Figura 6 – Diagrama simplificado de um sistema de banco de dados.



Fonte: Elmasri e Navathe (2011).

Date (2003) afirma que profissionais da área, utilizam erroneamente o termo “banco de dados” quando, na verdade, estão se referindo ao SGBD, sendo importante fazer a distinção, assegurando que o SGBD se trata de um conjunto, o qual contém o banco de dados em si e o kit de ferramentas para construí-lo e acessá-lo.

4.2 BANCO DE DADOS RELACIONAL

Proposto em 1970, o modelo de dados relacional possibilitou uma grande evolução para a área de banco de dados e superou os dois modelos de dados dominantes da época (hierárquico e em rede). Depois de pouco tempo da sua apresentação, foram desenvolvidos esboços de SGBDs relacionais em projetos na

corporação IBM – *International Business Machines* e Universidade de Berkeley na Califórnia e, logo, em seguida, já haviam produtos de banco de dados relacionais sendo oferecidos por fabricantes (RAMAKRISHNAN; GEHRKE, 2011).

Sendo o modelo principal para aplicações comerciais de banco de dados, o modelo relacional assumiu este posto por sua facilidade de desenvolvimento se comparados com modelos anteriores, como: o hierárquico e de redes (SILBERSCHATZ; KORTH; SUDARSHAN, 2012).

Elmasri e Navathe (2011) afirmam que um banco de dados relacional, é um conjunto de relações, no qual cada relação é, de maneira informal, semelhante a uma tabela. Ao se representar uma relação por uma tabela, cada linha representa um conjunto de dados inter-relacionados, e condiz com uma entidade ou acontecimento do mundo real. Neste modelo, os nomes das colunas e da tabela auxiliam na interpretação dos dados em cada linha.

Dentro da nomenclatura formal do modelo relacional, denominam-se a linha como Tupla, o nome da coluna como Atributo e a tabela como Relação. O tipo de dado que uma coluna comporta é representado por um conjunto de valores denominado domínio (ELMASRI; NAVATHE, 2011).

Date (2003) afirma que, os tipos de dados, também chamados de domínio, delimitam quais valores podem ser inseridos em uma coluna, alguns exemplos comuns são: números inteiros (-3, -2, -1, 1, 2, 3), caracteres (letras do alfabeto), números decimais (1,5; π ; $\frac{1}{4}$).

Os principais benefícios do banco de dados relacional em relação aos modelos anteriores são a representação simplificada dos dados e a facilidade para construir consultas complexas. A forma de representação dos dados permite que usuários de todos os níveis (até iniciantes) compreendam o conteúdo do banco de dados (RAMAKRISHNAN; GEHRKE, 2011).

A Figura 7 expõe a forma, na qual, os dados são dispostos, onde os cabeçalhos das colunas descrevem os nomes dos campos (ou atributos) e cada linha representa um registro. Observa-se também o comportamento da tipagem, na qual demonstra que cada coluna comporta apenas um tipo de dados (RAMAKRISHNAN; GEHRKE, 2011).

Figura 7 – Exemplo de relação ou tabela.

The diagram shows a table with the following structure:

CAMPOS (ATRIBUTOS, COLUNAS)				
Nomes de campo				
<i>id-aluno</i>	<i>nome</i>	<i>login</i>	<i>idade</i>	<i>média</i>
50000	Dave	dave@cs	19	3,3
53666	Jones	jones@cs	18	3,4
53688	Smith	smith@ee	18	3,2
53650	Smith	smith@math	19	3,8
53831	Madayan	madayan@music	11	1,8
53832	Guldu	guldu@music	12	2,0

Labels in the diagram:

- CAMPOS (ATRIBUTOS, COLUNAS)**: Points to the header row.
- Nomes de campo**: Points to the field names in the header.
- TUPLAS (REGISTROS, LINHAS)**: Points to the data rows.

Fonte: Ramakrishnan e Gehrke (2011).

Nos SGBDs relacionais, a visualização de seu conteúdo é realizada no formato de tabelas, e nada além disso. Ao executar uma consulta, os dados retornados formam novas tabelas geradas a partir de dados existentes. O motivo de serem chamados relacionais tem premissa no termo *relação*, que matematicamente caracterizam uma tabela (DATE, 2003).

Elmasri e Navathe (2011) afirmam que, a maioria dos bancos de dados relacionais comerciais, utilizam a Linguagem de Consulta Estruturada, ou SQL (*Structured Query Language*), a fim de descrever comandos ao banco de dados, para por exemplo: efetuar consultas, atualizações, definir visões (ou *views*) sobre o banco, parametrizar segurança e autorização, gerenciar o controle de transações, entre outras operações. Ela também fornece diretrizes para embutir comandos SQL em linguagens de programação de uso geral como Java, Cobol ou C/C++. A Figura 8 exibe uma instrução SQL para criação de uma tabela de dados, seus atributos e restrições.

Figura 8 – Trecho de instrução SQL para criação de uma tabela.

```
CREATE TABLE FUNCIONARIO
(Pnome          VARCHAR(15)          NOT NULL,
Minicial        CHAR,
Unome           VARCHAR(15)          NOT NULL,
Cpf             CHAR(11),            NOT NULL,
Datanasc        DATE,
Endereço        VARCHAR(30),
Sexo            CHAR,
Salario         DECIMAL(10,2),
```

Fonte: Elmasri e Navathe (2011).

4.3 BANCO DE DADOS NÃO-RELACIONAL

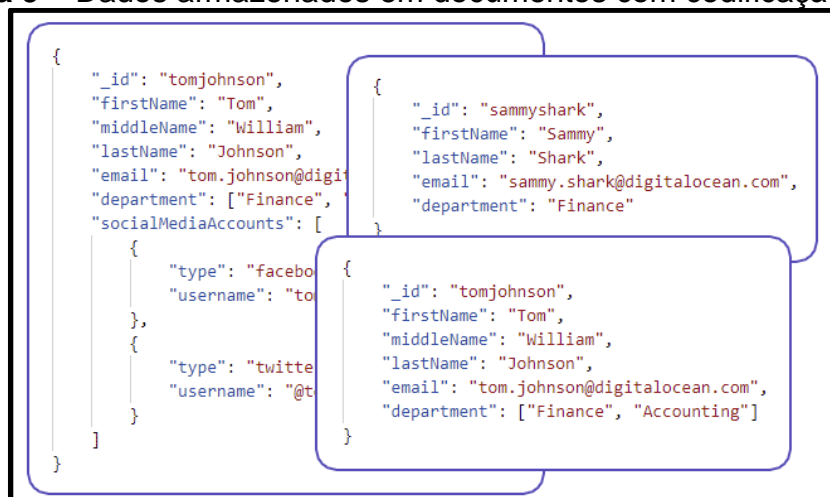
Tejada (2023) relata que o banco de dados não-relacional, diferente do relacional, não utiliza tabelas para o armazenamento de seus dados. Seguindo no caminho inverso dos modelos tradicionais, esses bancos seguem um modelo direcionado para o requisito de armazenamento de dados do software, o qual prestará suporte.

Como não se limita a tabelas de linhas e colunas, o banco de dados não-relacional fornece maior versatilidade no que diz respeito ao armazenamento de dados, pois é otimizado para o requisito de dados da aplicação a qual dará suporte. Esse modelo não requer rigidez no armazenamento de dados como no modelo relacional, onde alterações na estrutura do banco de dados não impactam nos dados existentes (KONDADO TEAM, 2022).

De acordo com Tejada (2023), ao invés das tradicionais tabelas, os bancos de dados não relacionais possuem outras formas de armazenar os dados, conforme são descritas a seguir:

- Documentos: os dados são armazenados em um conjunto de caracteres com alguma codificação como: JSON (*Javascript Object Notation*, ou Notação de Objetos *Javascript*), ou XML (*Extensible Markup Language*, ou Linguagem de marcação Extensível). Um exemplo é descrito na Figura 9, na qual os dados utilizam a codificação JSON.

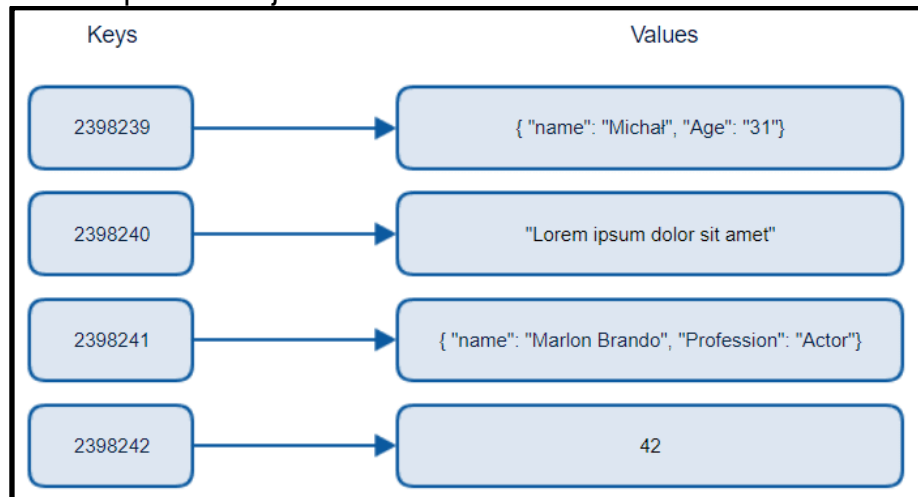
Figura 9 – Dados armazenados em documentos com codificação JSON.



Fonte: Awari (2022).

- Chave e valor: os dados são armazenados em conjuntos de chave e valor, onde a chave atua como identificador único e os dados em si podem ter diversos tipos. A Figura 10 mostra como é o modelo de chave e valor.

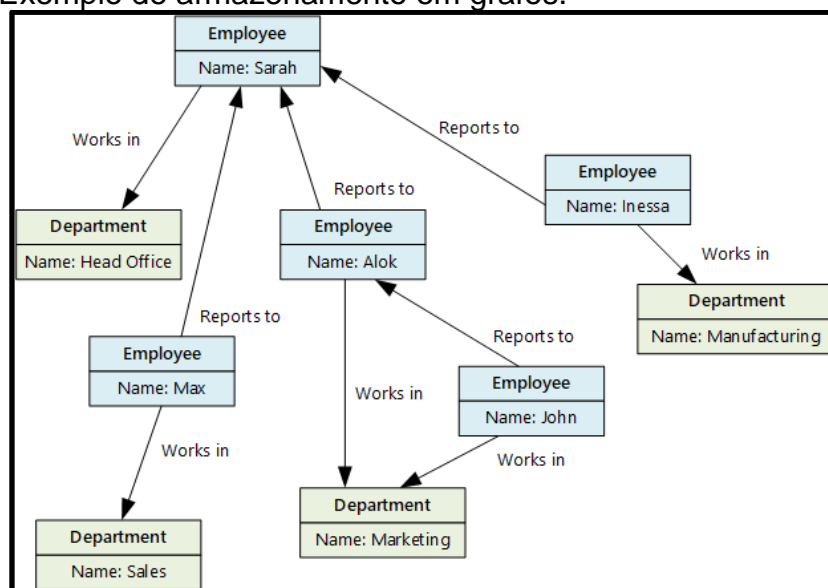
Figura 10 – Exemplo de conjuntos de chave e valor.



Fonte: Awari (2022).

- Grafos: os dados são representados e armazenados no formato de nós conectados por arestas, onde eles representam entidades e arestas representam o relacionamento que elas possuem. A Figura 11 descreve o modelo de armazenamento de dados em grafo.

Figura 11 – Exemplo de armazenamento em grafos.



Fonte: Tejada (2023).

Outra característica do modelo não-relacional, é o não uso da linguagem SQL para consultas e definições, fato que originou o termo *NoSQL* (*Not Only SQL*, não somente SQL), o qual é usado para descrever bancos de dados não relacionais (AWARI, 2022).

5 DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS

Barboza (2022) relata que, a área de desenvolvimento de *software* impulsionou vários setores do mercado mundial, mudando drasticamente os procedimentos e todo o dia a dia de diversas empresas. Esta mudança foi viabilizada pelo trabalho dos programadores, que cada vez mais, tem a sua disposição, recursos que possibilitam a agilidade na entrega de aplicações comerciais.

Softwares não são limitados por propriedades materiais, estão além da esfera das leis da física e não se restringem aos processos de produção, são abstratos e não possuem natureza material. Isto implica em um potencial sem limites e descomplica a área da engenharia de software. Contudo, essa falta de limitações os tornam produtos complicados de se entender e caros para se alterar (SOMMERVILE, 2011).

De acordo com Silva (2010), com o passar dos anos, verificou-se que a complexidade da programação de aplicações comerciais aumentou e gerou a necessidade de se desenvolver *softwares* de simples manutenção, pouco acoplados, fáceis de testar e rápidos de se desenvolver. Logo, a aplicação de ferramentas e padrões específicos de desenvolvimento se tornou uma necessidade inerente ao negócio, pois seriam utilizados recursos previamente validados.

Para haver um melhor entendimento sobre alguns assuntos, precisa-se primeiramente entender o que é um *software*. *Software* é uma coleção de códigos escritos em uma determinada linguagem de programação, ele é usado para efetuar inúmeras tarefas pré-determinadas em algum dispositivo computacional (DEVMEDIA, 2021).

Para Sommerville (2011), o uso do *software* é fundamental para o funcionamento do mundo moderno, pois vários serviços e infraestruturas são gerenciados por recursos informatizados. Ainda, grande parte dos eletrodomésticos e eletroeletrônicos são controlados por um sistema computacional. Algumas áreas são completamente informatizadas, como: a manufatura, a distribuição e o setor financeiro. Na esfera do entretenimento, o emprego do *software* é intenso em áreas como a produção musical, jogos, cinema e televisão.

O software é geralmente separado em duas partes principais, a interface de usuário (ou *front-end*) e a camada de funcionamento interno (ou *back-end*)

(DEV MEDIA, 2021). As seções seguintes descrevem brevemente essa divisão e também alguns termos comuns da área de tecnologia.

5.1 FRONT-END

Amazon (2023a) afirma que, em linhas gerais, atribui-se o termo *front-end* para designar interface gráfica do usuário. Ele é composto por tudo que os usuários veem e interagem, como componentes visuais, botões, listas suspensas, gráficos e textos. É responsável por fornecer ao usuário meios de interagir com as informações de determinada aplicação.

O *front-end*, é a camada visível ao usuário do sistema, tudo que é possível visualizar e interagir pertence a esta camada (DEV MEDIA, 2021). Esta parte do sistema se encarrega de fornecer uma boa experiência ao usuário, provendo a forma, o desenho, o conteúdo, as interações e respostas que um *site* ou sistema pode apresentar às pessoas (MACHADO, 2021).

De acordo com Amazon (2023a), existem três linguagens principais que determinam as interações de usuário com a camada do *front-end* de uma aplicação: HTML – *HyperText Markup Language* (Linguagem de Marcação de Hipertexto), CSS – *Cascading Style Sheets* (Folhas de Estilo em Cascatas) e *Javascript*.

Com o HTML, é possível codificar estruturas para páginas web, definindo elementos e sua hierarquia, parametrizando cores e posição de componentes, fontes do texto, e outros aspectos estruturais do *site*, pode-se dizer que ele é encarregado de formar o esqueleto da página (REDAÇÃO TERA, 2023).

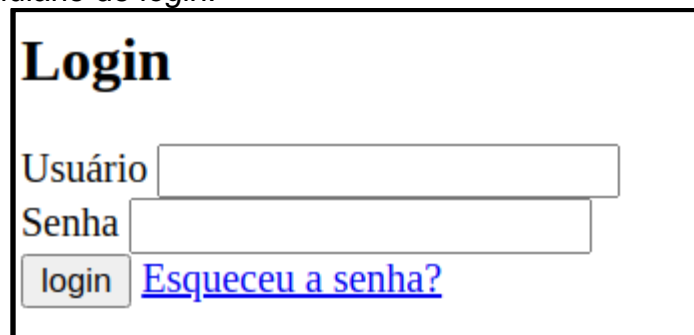
A Figura 12 apresenta um exemplo de código HTML, o qual gera um formulário simples de *login*. Na Figura 13, o resultado do código da Figura 12 é exibido ao executá-lo no navegador.

Figura 12 – Exemplo de código HTML.

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="style.css">
  <title>Login</title>
</head>
<body>
<div id="login">
  <form class="card">
    <div class="card-header">
      <h2>Login</h2>
    </div>
    <div class="card-content">
      <div class="card-content-area">
        <label for="usuario">Usuário</label>
        <input type="text" id="usuario" autocomplete="off">
      </div>
      <div class="card-content-area">
        <label for="password">Senha</label>
        <input type="password" id="password" autocomplete="off">
      </div>
    </div>
    <div class="card-footer">
      <input type="submit" value="login" class="submit">
      <a href="#" class="recuperar_senha">Esqueceu a senha?</a>
    </div>
  </form>
</div>
</body>
</html>
```

Fonte: Método Programar (2023).

Figura 13 – Formulário de *login*.



The image shows a visual mockup of the login form. At the top is the word "Login" in a large, bold, black serif font. Below it are two input fields: the first is labeled "Usuário" and the second is labeled "Senha". Both labels are in a black serif font. Below the "Senha" field is a button with the text "login" in a black serif font. To the right of the button is a blue, underlined link that says "Esqueceu a senha?". The entire form is enclosed in a thin black rectangular border.

Fonte: Método Programar (2023).

Kruger (2023) relata que, aplicando o CSS, é possível prover estilo e aparência agradáveis para páginas estruturadas em HTML, seu uso possibilita a aplicação de cores, animações e efeitos visuais, o que torna a página mais dinâmica dos pontos de vista visual e interativo.

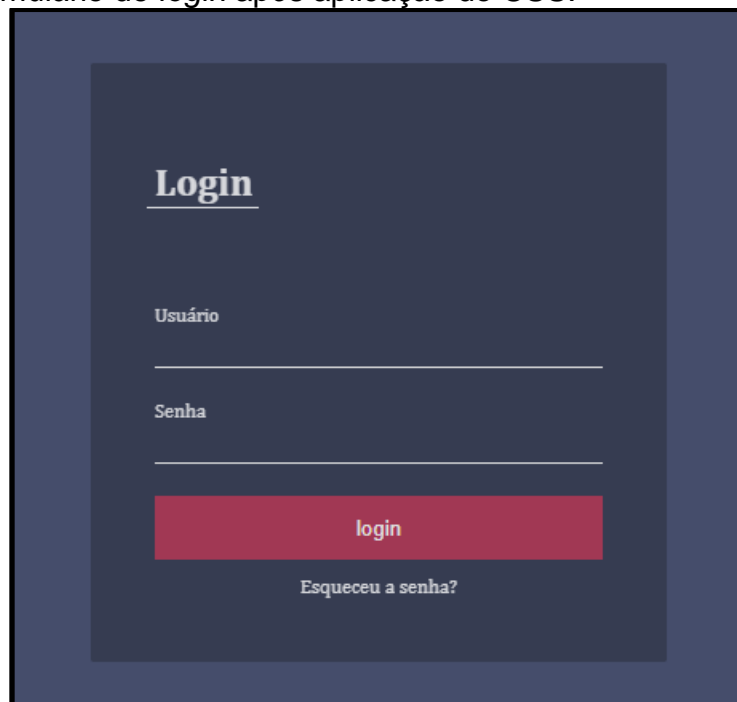
A Figura 14 mostra, um trecho de código, produzido em CSS para aplicar estilo no formulário da Figura 13. A Figura 15 apresenta o resultado desta aplicação.

Figura 14 – Exemplo de estilos escritos em CSS.

```
body {  
  padding: 0;  
  margin: 0;  
  background-color: #454d6b;  
}  
  
#login {  
  display: flex;  
  align-items: center;  
  justify-content: center;  
  height: 100vh;  
  font-family: Cambria, Cochin, Georgia, Times, 'Times New Roman', serif;  
}  
  
.card {  
  background-color: rgba(19, 19, 19, 0.3);  
  padding: 40px;  
  border-radius: 2px;  
  width: 280px;  
}
```

Fonte: Método Programar (2023).

Figura 15 – Formulário de *login* após aplicação do CSS.



The image shows a login form with a dark blue background and a lighter blue border. The form contains a title "Login", two input fields for "Usuário" and "Senha", a red "login" button, and a link "Esqueceu a senha?".

Fonte: Método Programar (2023).

Sendo parte fundamental da camada *front-end*, o *Javascript* é uma linguagem utilizada para prover às páginas *web* dinamicidade e interatividade, com ele é possível manipular o HTML e o CSS de tal maneira que o usuário consiga interagir com a página de várias maneiras. Sua aplicação fornece meios de criação de efeitos visuais,

transições dinâmicas, validação de formulários e outras funcionalidades (ESPINOSA, 2023).

5.2 BACK-END

Enquanto a camada do *front-end* fornece a forma e apresentação de um sistema, a camada denominada *back-end* atua nos bastidores, fornecendo suporte ao funcionamento correto do software, enviando e processando dados no servidor e mantendo a conexão com o banco de dados, sendo considerada a seção mais complexa de uma aplicação *web* (REDAÇÃO TERA, 2023).

Devmedia (2021) descreve o *back-end* como a camada principal de uma aplicação. Sendo invisível ao usuário, ele trabalha processando os dados e realizando as funções que o *software* se dispõe a fazer e se encarrega de prover a “inteligência” desse mecanismo.

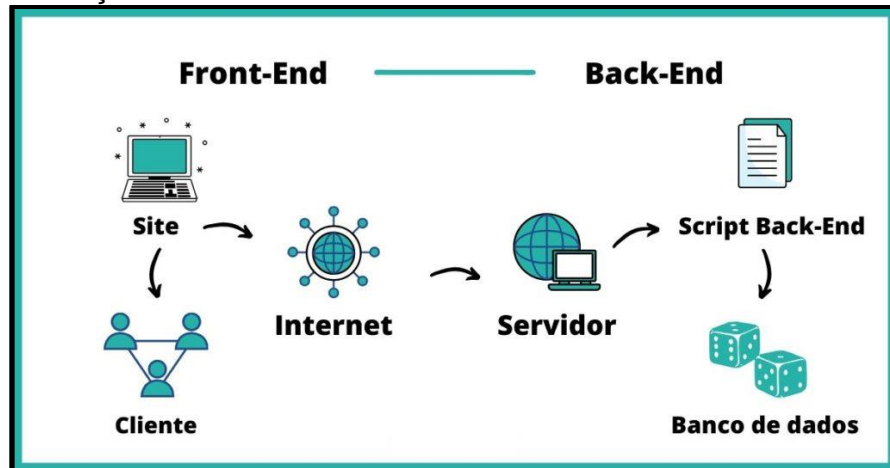
Segundo Machado (2021) a camada do *back-end* engloba o servidor, o banco de dados e a aplicação. Quando um *site* da *web* é acessado o usuário recebe os dados necessários para visualização da página oriundos do servidor, inclusive, o armazenamento correto dessas informações e a garantia de segurança do site são responsabilidades do *back-end*.

Em um ambiente *web*, o servidor é encarregado de atender às solicitações do cliente e fornecer o que foi requisitado. Uma das funções do *back-end* é cuidar da lógica por trás do servidor. Dessa forma, ele atua administrando alguns pontos, como: a conexão com o banco de dados, os tópicos de segurança e a conexão com outras aplicações intermediada por APIs (*Application Programming Interface*, ou Interface de Programação de Aplicação) (REDAÇÃO TERA, 2023).

O *back-end* provê um caminho entre as informações oriundas do navegador cujo destino é o banco de dados e também o caminho inverso, esta camada ainda se encarrega de validar se os dados processados estão corretos e aplicar as devidas regras de negócio do sistema. A Figura 16 representa, esta ponte que o *back-end* provê, onde o cliente acessa um determinado *site* ou sistema disponibilizado na internet por um servidor que contém as regras de negócio e validações representadas

pelos *scripts back-end* e após as informações passar por esses *scripts* então podem ser armazenadas no banco de dados (ANTUNES, 2020).

Figura 16 – Relação entre o *Front-end* e o *Back-end*.



Fonte: Antunes (2020).

Para Machado (2021), toda a parte interna de um sistema é desenvolvida com a utilização de linguagens de programação, por exemplo, o Java que é uma das linguagens utilizadas e que possui uma plataforma de desenvolvimento de *software* própria.

Segundo Bessa (2023) o Java como plataforma, fornece um ambiente de desenvolvimento completo, sendo composta por: uma linguagem de programação de alto nível orientada a objetos, uma máquina virtual (JVM – *Java Virtual Machine*, ou Máquina Virtual Java) que assegura o correto funcionamento multiplataforma, um ambiente de execução (JRE – *Java Runtime Environment*) e um kit de desenvolvimento (JDK – *Java Development Kit*).

Sendo uma das linguagens prediletas das corporações, o Java é uma linguagem de programação muito empregada na implementação de sistemas feitos para operar em dispositivos que possuam comunicação via internet (DEITEL, 2017).

A Figura 17 expõe, um exemplo de código feito na linguagem de programação Java, implementação que visa retornar ao usuário uma lista de produtos.

Figura 17 – Exemplo de código feito em *Java*.

```
public class ProdutoDAO {
    public List<Produto> lista(){
        // Implementação qualquer que devolve uma lista de produtos
    }
    public void produtosResumido(){
        List<Produto> prs = lista();
        for (Produto p : prs) {
            String dse = p.getDescricao().trim();
            StringBuilder sb = new StringBuilder();
            for (int i = 0; i < dse.length(); i++) {
                if(i > 15) {
                    sb.append(" ... ");
                    break;
                }
            }
            sb.append(dse.charAt(i));
        }

        String da = sb.toString();
        String vf = p.getValor().toString().replace(".", ",");
        String df = p.getId() + " - " + da + " - R$ " + vf;
        System.out.println(df);
    } }
}
```

Fonte: Felipe (2017).

Ewally (2023) relata que, embora possuam funções distintas, as camadas *back-end* e *front-end* de um *software* estão estreitamente conectadas, fornecendo sincronia para o funcionamento das aplicações.

A conexão entre camadas é o que permite sua troca de dados e pode ser programada por meio de uma API, tema da próxima seção (GOCACHE, 2022).

5.2.1 Application Programming Interface

Segundo Melo (2021) APIs são mecanismos de *software* que possibilitam duas aplicações distintas se integrarem e trocarem informações mediadas por um grupo de regras e protocolos. Ao implementar uma API, um determinado sistema provê um canal de comunicação para que outros serviços tenham acesso às suas funcionalidades, sem precisarem saber como elas foram implementadas.

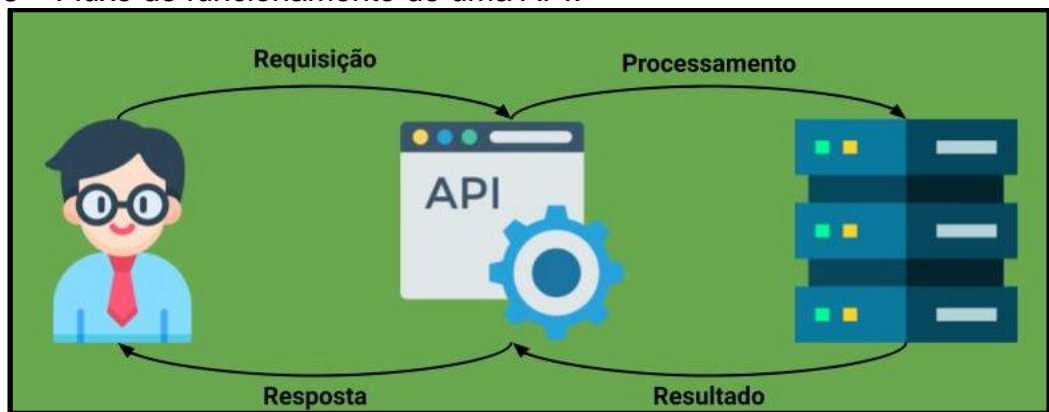
Uma API é um conjunto de recursos que fornece um meio de comunicação padronizado para aplicações sem que uma tenha conhecimento da maneira que a

outra foi implementada. Ela opera de maneira semelhante a um contrato entre duas entidades. Caso uma das partes efetue uma requisição devidamente estruturada, haverá uma resposta da outra parte de acordo com os termos do contrato (RED HAT, 2023).

Amazon (2023b) relata que, de forma geral, uma API pode ser descrita na forma de cliente e servidor. Diante disso, o *software* que realiza a requisição é denominado cliente e a aplicação que responde a esta requisição é chamada de servidor.

A Figura 18 demonstra o fluxo completo de uma API, onde ela recebe uma requisição do cliente, processa as informações recebidas junto a aplicação, recebe o resultado, gera e envia a resposta ao cliente. Lembrando que o cliente pode ser o navegador *web*, um aplicativo móvel, um programa *desktop*, entre outros.

Figura 18 – Fluxo de funcionamento de uma API.



Fonte: Veras (2022).

5.3 FRAMEWORK

Conforme Barboza (2022), algumas das ferramentas em destaque no desenvolvimento de *software* são os *frameworks*, que são basicamente conjuntos de código previamente desenvolvidos que podem ser aplicados na programação de sistemas de vários tipos e que facilitam o processo de desenvolvimento.

Um *framework* serve como uma estrutura sólida para o desenvolvimento de aplicações com objetivos específicos, nas quais, há um custo elevado de programação. Com ele, a construção de sistemas é realizada a partir de uma estrutura

pré-definida, adaptando apenas as individualidades, de acordo com o projeto (KRIGER, 2022).

Para Fontoura (2023), os *frameworks* podem ser aplicados em vários projetos, como: sistemas *web*, aplicativos móveis e jogos. Eles são estruturas capazes de oferecer uma gama de regras e padrões que auxiliam na construção de aplicativos e sistemas de maneira ordenada e coesa e agilizam este processo.

Algumas vantagens são obtidas ao se utilizar um *framework* no processo de desenvolvimento de *software*, tais como: redução no tempo de programação, padronização do código, segurança, diminuição de erros e melhora nos custos de produção (BARBOZA, 2022).

Segundo Kriger (2022) as vantagens de aplicar o uso do *framework* em um projeto de *software* são: aumento da produtividade, redução de erros no código, manutenção simplificada, maior segurança, padronização do projeto, otimização dos custos e tempo de produção.

6 MATERIAIS E MÉTODOS

Esta pesquisa foi desenvolvida seguindo os seguintes passos: o início se deu com uma revisão bibliográfica acerca dos conceitos necessários para o completo entendimento do objeto de estudo. Na sequência, foi efetuado um levantamento de requisitos para o desenvolvimento do *software* proposto com base na revisão bibliográfica elaborada. Posteriormente, foi projetada a arquitetura do sistema e feito a modelagem do banco de dados. Ainda, foram desenvolvidos os códigos para a interface de usuário e para as regras de negócio da aplicação. Por fim, foram efetuados testes para validação do correto funcionamento do *software*.

6.1 DEFINIÇÃO DO TIPO DE PESQUISA

Do ponto de vista dos objetivos, esta pesquisa se enquadra como uma pesquisa exploratória, pois tem como proposta aprimorar o conhecimento do objeto de estudo, envolveu um levantamento bibliográfico e foi motivada a viabilizar o entendimento do problema proposto, no intuito de levantar hipóteses e torná-lo mais claro (GIL, 2002).

Ainda, o autor relata que a maior parte das pesquisas exploratórias se enquadram como pesquisas bibliográficas e baseado nos procedimentos técnicos aplicados para desenvolver esta pesquisa, ela pode ser classificada como uma pesquisa bibliográfica, pois foi desenvolvida alicerçada em materiais bibliográficos previamente concebidos, principalmente de livros e artigos científicos.

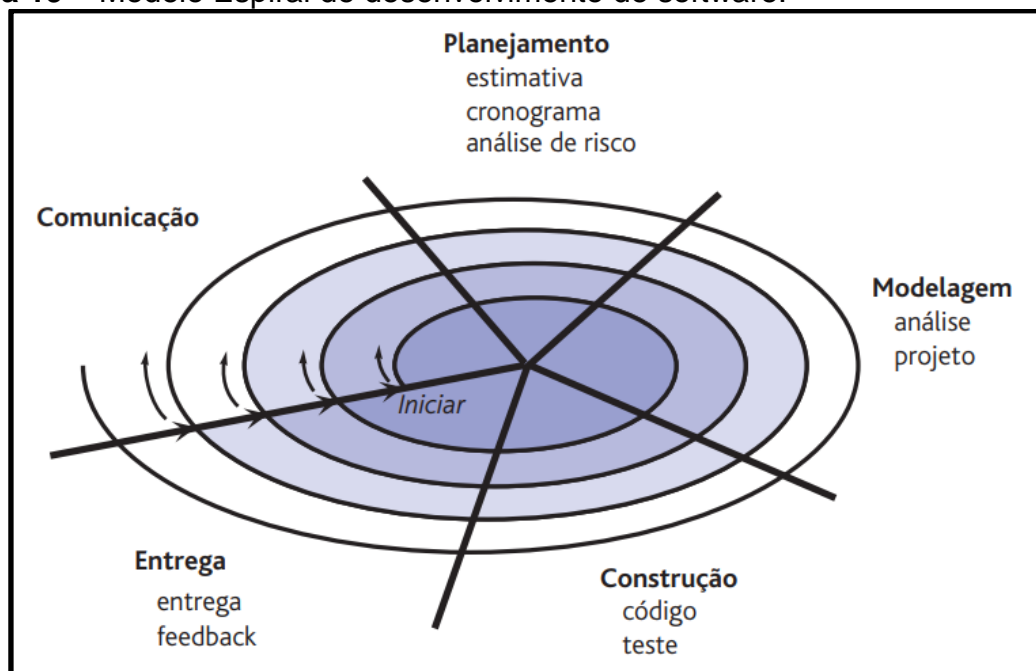
Quanto ao método, a pesquisa se caracteriza como qualitativa, pois busca desenvolver um produto, testá-lo e entendê-lo, levando em consideração a sua totalidade e não apenas atributos isolados (ALYRIO, 2009).

6.2 DESENVOLVIMENTO DA PESQUISA

Para o desenvolvimento do sistema proposto por esta pesquisa, as etapas foram definidas com base no modelo em Espiral (PRESSMAN; MAXIM, 2016), conforme pode ser verificado a seguir:

- **Levantamento de requisitos:** verificação da revisão bibliográfica produzida para mapear as operações do CD que o sistema irá atender, bem como seus casos de uso e atores.
- **Modelagem das entidades do sistema:** desenvolvimento das entidades do sistema, definição de seus atributos e criação do banco de dados.
- **Desenvolvimento do código do *back-end*:** implementação em código de toda a lógica e regras de negócio do sistema.
- **Desenvolvimento do código do *front-end*:** desenvolvimento da interface de usuário do sistema.
- **Testes:** realização de testes das funcionalidades do sistema, no intuito de validar seu correto funcionamento.

Figura 19 – Modelo Espiral de desenvolvimento de software.



Fonte: Pressman e Maxim (2016).

Apesar das etapas serem listadas em ordem, o processo seguiu a lógica do modelo Espiral (PRESSMAN; MAXIM, 2016), no qual, em alguns pontos, foi necessário que ciclos de desenvolvimento e testes fossem realizados não necessariamente em sequência.

6.2.1 Recursos e Ambiente de Desenvolvimento

Para desenvolver esta pesquisa, foram utilizados os seguintes recursos no ambiente de desenvolvimento de *software*:

- **IntelliJ IDEA *Ultimate*:** ambiente de desenvolvimento de *software* integrado, o qual fornece vários recursos interessantes que facilitam o trabalho do programador (autopreenchimento de código incompleto, terminal de comando para subida de aplicação e testes, suporte a linguagens de *back-end* e *front-end* o que permite o desenvolvimento tanto da lógica de negócio quanto da interface de usuário em uma única “estação de trabalho”);
- **MySQL 8.0 *Community*:** um dos SGBDs relacionais de uso comercial mais utilizados no mundo, possui uma interface gráfica amigável que simplifica a configuração e utilização do banco de dados;
- **Postman:** ferramenta utilizada para testes de API, a qual fornece uma interface para escrever e enviar requisições para APIs disponibilizadas na web e receber suas respostas;
- **Java 17:** uma das linguagens de programação orientada a objetos mais utilizadas no mundo, segura, confiável, robusta e adequada para o desenvolvimento de aplicações *web* e *desktop*;
- **Spring Framework:** *framework* flexível e leve que trabalha em conjunto com o Java no *back-end* da aplicação. Fornece recursos interessantes para um rápido desenvolvimento de aplicações, como: controles de transações, acesso ao banco de dados, segurança, autenticação e integração com o *front-end*;
- **Thymeleaf:** gerador de *templates* para aplicações *web* e *desktop*. Fornece recursos a escrita de código, que permite a integração entre o *front-end* e o *back-end*.

7 RESULTADOS E DISCUSSÕES

Nesta seção, são apresentados os resultados obtidos, fornecendo uma visão geral do sistema implementado, juntamente com

7.1 LEVANTAMENTO DE REQUISITOS

Com base na bibliografia revisada, foi elaborado um mapeamento das principais operações logísticas presentes em um CD. Em seguida, foi realizado um levantamento de requisitos, a fim de verificar as principais funcionalidades do sistema. Na sequência, foi produzido um diagrama de casos de uso, com o intuito de nortear o desenvolvimento do sistema WMS.

Os requisitos foram divididos em duas categorias:

1. Importante (I): imprescindível para o sistema;
2. Recomendável (R): pode ser excluído em casos de limitações técnicas ou de cronograma.

Esses requisitos ainda foram classificados como Funcionais (F) e Não Funcionais (NF). Além disso, o campo “*Status*” foi inserido no quadro para checagem do desenvolvimento de cada requisito.

	Requisito	Categoria	Classe	Status
1	Armazenar dados de clientes	I	F	OK
2	Armazenar dados de fornecedores	I	F	OK
3	Armazenar dados de produtos	I	F	OK
4	Armazenar dados das operações logísticas	I	F	OK
5	Armazenar dados do CD (endereços, área de recebimento e expedição)	I	F	OK
6	Registrar operações por meio da interface de usuário	I	F	OK
7	Fornecer visão das entidades registradas no banco de dados em tela	R	F	OK
8	Implementar scripts para popular o banco de dados para eventuais testes	R	NF	OK
9	Registrar pedidos (de compra e de venda) por meio de API	R	F	OK

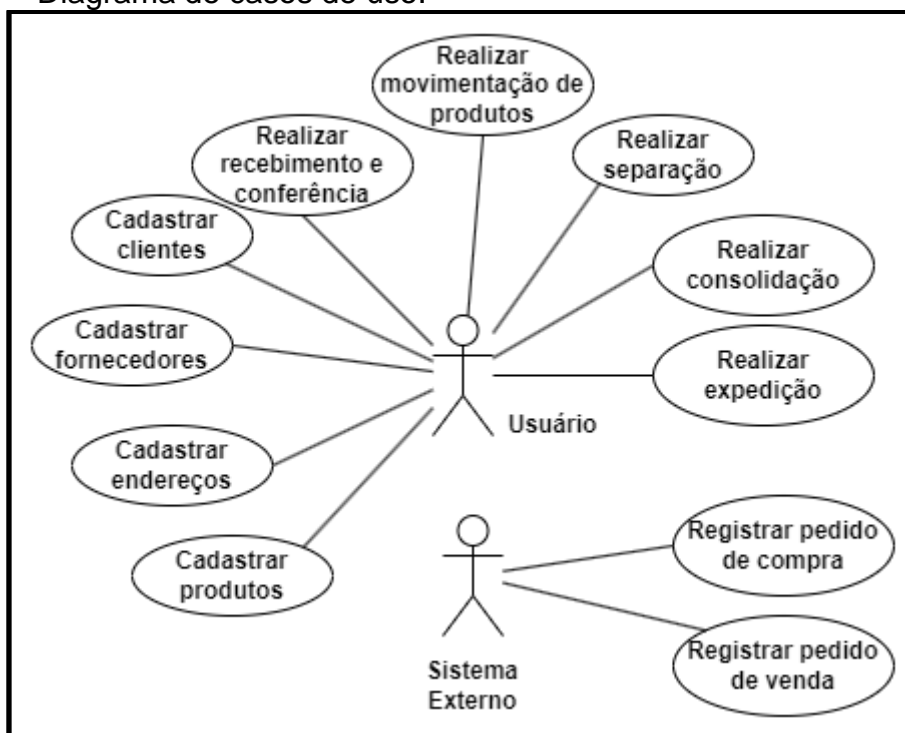
Quadro 2 – Levantamento de requisitos de software.

Fonte: Elaborado pelo autor (2023).

A Figura 20 descreve as funcionalidades propostas na elaboração do sistema. Todas essas funcionalidades foram planejadas para atender as operações básicas de um CD (recebimento, armazenagem, separação, consolidação e expedição). É

importante ressaltar que, o processo de conferência de mercadorias está vinculado a operação de recebimento, a consolidação da carga está associada a operação de expedição e a armazenagem está mapeada como movimentação. Em termos de lógica de negócio, estes processos relacionados serão acessados e efetuados em um único fluxo dentro do sistema.

Figura 20 – Diagrama de casos de uso.



Fonte: Elaborado pelo autor (2023).

Na imagem, observa-se que o diagrama possui uma divisão em dois atores (usuário e sistema externo), para os quais as funções do sistema foram atribuídas.

O ator usuário, que representa algum funcionário do CD, é responsável por operar o sistema, ao passo que efetua as operações logísticas. Ademais, ele também se encarrega de realizar manutenção nos cadastros de produtos, empresas (clientes e fornecedores) e endereços. A manutenção dos cadastros é necessária para o bom funcionamento do sistema e a prevenção de erros.

O ator sistema externo, que representa um *software* integrado ao WMS, poderá solicitar o registro de pedidos de compra e de venda, executando requisições padronizadas a uma API disponibilizada pelo sistema para registro do pedido.

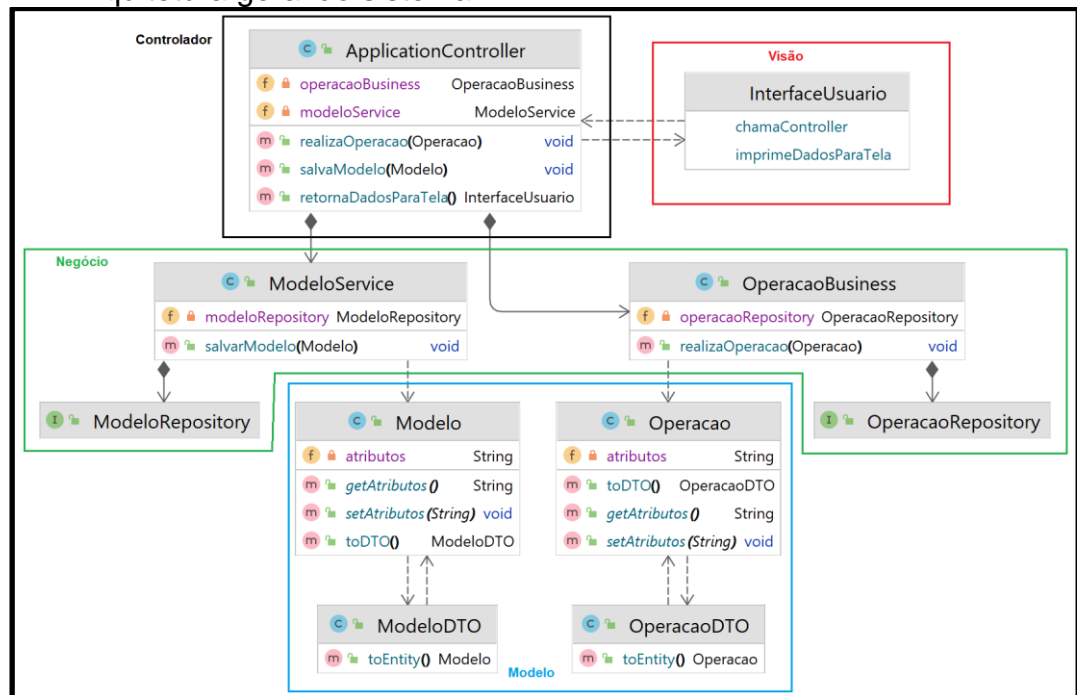
Após a definição das funcionalidades, foi projetada a estrutura do sistema e modelado o banco de dados. Tais processos serão descritos nas próximas seções.

7.2 ESTRUTURA DO SISTEMA

O sistema foi estruturado, utilizando os recursos do Spring Framework na criação do projeto e no desenvolvimento do código. Além disso, sua arquitetura foi construída com base no modelo MVC (Modelo-Visão-Controlador), a qual é facilitada pelo uso do Spring Framework. No modelo MVC, separa-se o *software* em camadas de tal maneira que reduza o acoplamento, ou seja, cada camada trabalha com pouca ou nenhuma dependência das outras.

A Figura 21 caracteriza a arquitetura geral do sistema, onde observa-se a divisão em 4 camadas (Visão, Controlador, Negócio e Modelo) e 6 tipos de classes Java (cujos tipos são definidos pelo respectivo sufixo), sendo elas: *Controller* (controlador), *Service* (serviço), *Business* (negócios), *Repository* (repositório), classes modelo (Modelo e Operacao) e, por fim, os DTOs (*Data Transfer Object*, ou Objeto de Transferência de Dados).

Figura 21 – Arquitetura geral do sistema.



Fonte: Elaborado pelo autor (2023).

A troca de dados entre a interface de usuário e a aplicação ocorre por meio da classe *Controller*, que recebe uma requisição previamente mapeada a um método,

aciona a cadeia de negócio, recebe os dados de retorno e os envia para a interface de usuário.

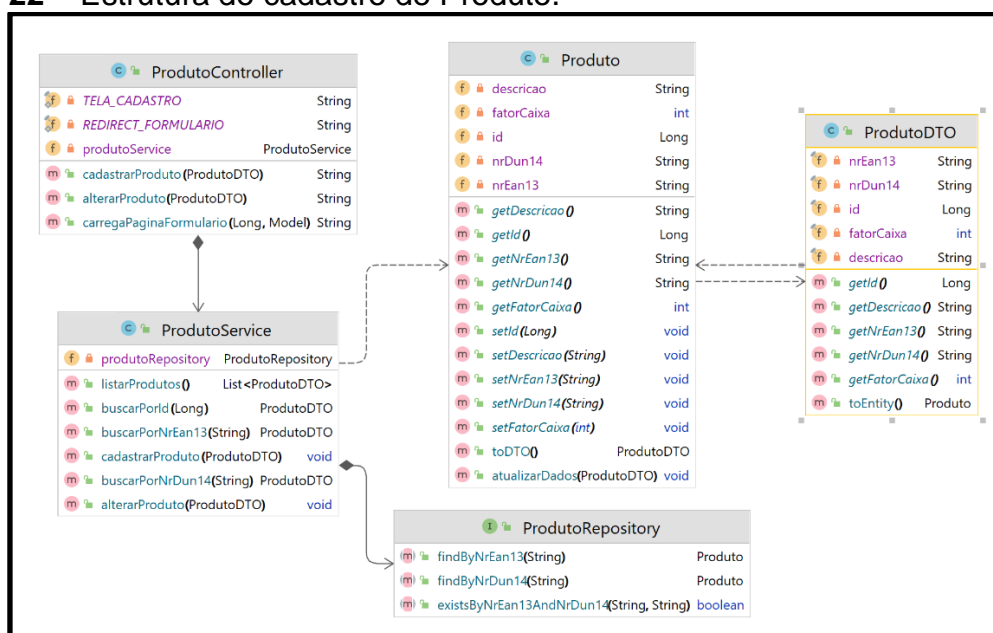
As classes *Repository* definem e fornecem métodos de acesso aos dados registrados no banco de dados da aplicação.

As classes *Service* e *Business* são responsáveis por toda a lógica de negócio do sistema e pelo acionamento dos métodos de acesso aos dados das classes *Repository*. A diferença entre elas, é que o maior volume de regras de negócio está concentrado nas classes *Business* que são utilizadas para realizar operações complexas no sistema, enquanto que as classes *Service* aplicam pouca ou nenhuma regra de negócio em seus fluxos e são utilizadas em funcionalidades mais simples com acesso ao banco de dados como em cadastros por exemplo.

As classes Modelo representam objetos do mundo real, contendo um atributo para cada característica importante e possuem mapeamento com tabelas do banco de dados, já as classes DTO apenas replicam os dados das classes modelo sem possuir mapeamento com o banco de dados, desta forma, são usadas apenas no recebimento de dados no *Controller* e na sua impressão na interface de usuário.

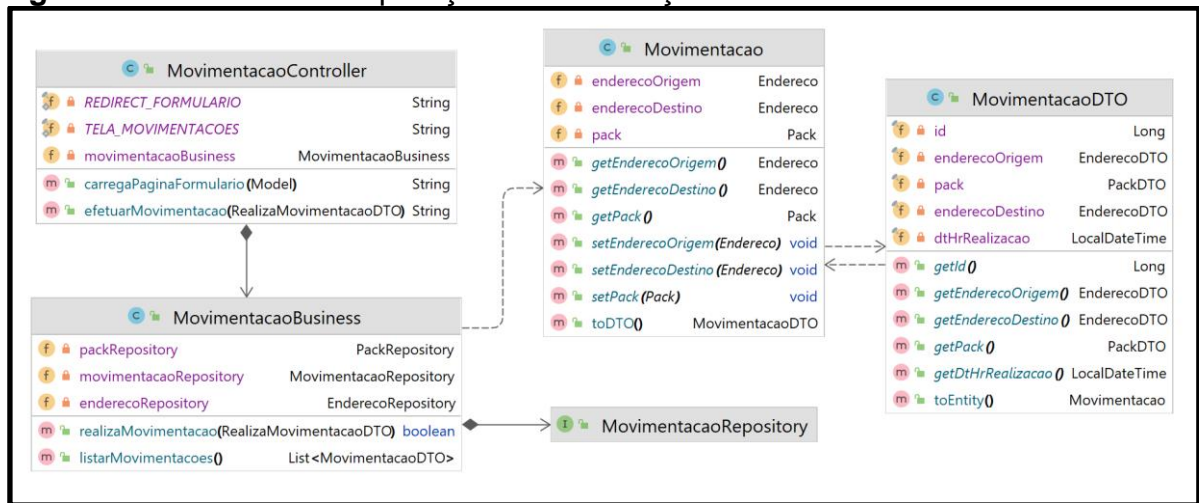
As figuras 22 e 23 mostram o uso desta arquitetura, nas quais, observa-se o uso da classe *Service* para o cadastro de Produto, contendo apenas métodos para acessar e escrever no banco de dados, e da classe *Business* para a operação Movimentação contendo um método para realizar a operação em si.

Figura 22 – Estrutura do cadastro de Produto.



Fonte: Elaborado pelo autor (2023).

Figura 23 – Estrutura da operação Movimentação.



Fonte: Elaborado pelo autor (2023).

O *front-end* foi desenvolvido na forma de páginas utilizando HTML para definir a estrutura, CSS para dar estilo e o *Thymeleaf* para troca de dados com o *back-end*.

Utilizando o *Thymeleaf*, foi possível mapear as classes da camada de modelo e seus atributos na forma de formulários e com isso permitir a manipulação de dados pelo usuário.

A construção do banco de dados foi elaborada com base nas classes Modelo do Java. Este processo foi feito por meio do mapeamento objeto-relacional fornecido pelo conjunto de duas ferramentas, a JPA (Java *Persistence API*, ou API de persistência Java) e o *Hibernate* (biblioteca que implementa os recursos padronizados pela JPA), eliminando a necessidade de escrever comandos SQL manualmente para criação e atualização da base de dados e das tabelas.

O conjunto *JPA–Hibernate* foi fornecido pelo módulo de persistência de dados do *Spring Framework*, denominado *Spring Data JPA*.

Para esta tarefa ser executada corretamente, foi necessário mapear as classes modelo com as tabelas, mediante a anotações conforme mostra a Figura 24. No código demonstrado, observa-se, a anotação `@Table`, definindo a classe `Produto` como uma tabela de nome produto, também se observa a anotação `@Column` vinculada aos atributos da classe, esta anotação define que cada atributo relacionado a ela é uma coluna da tabela produto, por fim, a anotação `@Id` define o identificador único da tabela. Todas as classes da camada de modelo receberam o mapeamento objeto-relacional, onde as anotações usadas, são fornecidas pela JPA.

Figura 24 – Classe de modelo da entidade Produto.

```
@Entity
@Table(name = "produto")
public class Produto {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @Column(name = "ds_descricao", nullable = false)
    private String descricao;

    @Column(name = "nr_ean13", nullable = false, length = 13, unique = true)
    private String nrEan13;

    @Column(name = "nr_dun14", nullable = false, length = 14, unique = true)
    private String nrDun14;

    @Column(name = "fator_caixa", nullable = false)
    private int fatorCaixa;
```

Fonte: Elaborado pelo autor (2023).

Com esse mapeamento, o *Hibernate* é capaz de criar o banco de dados, as tabelas, as colunas, os identificadores e as restrições logo após a subida da aplicação, conforme o trecho de *log* exibido na Figura 25. É válido destacar, que, o uso deste recurso é recomendado pelos desenvolvedores somente em ambiente de desenvolvimento e de testes, sendo orientado a criação manual do banco de dados para o ambiente de produção.

Figura 25 – Geração do banco de dados pelo *Hibernate*.

```
Hibernate:
create table produto (
  id bigint not null auto_increment,
  ds_descricao varchar(255) not null,
  fator_caixa integer not null,
  nr_dun14 varchar(14) not null,
  nr_ean13 varchar(13) not null,
  primary key (id)
) engine=InnoDB
Hibernate:
create table pedido (
  nr_pedido bigint not null auto_increment,
  dt_hr_emissao datetime(6) not null,
  status_pedido varchar(255),
  tipo_pedido varchar(255),
  id_empresa bigint not null,
  primary key (nr_pedido)
) engine=InnoDB
Hibernate:
alter table pedido
add constraint FK5vdgv51dbwnfce3ba3v557u70
foreign key (id_empresa)
references empresa (id)
Hibernate:
alter table pedido_produto
add constraint FKKhqu4f43gvaqaeqrqf1o6v0e
foreign key (nr_pedido)
references pedido (nr_pedido)
Hibernate:
alter table pedido_produto
add constraint FK323jgu8b2khvfx160jci8jkhL
foreign key (id_produto)
references produto (id)
```

Fonte: Elaborado pelo autor (2023).

Inicialmente, 3 tabelas foram definidas para gestão de estoque do CD, sendo elas: *produto*, *pack* e *container*.

A tabela *produto*, recebe o cadastro de produtos, com atributos de descrição, código de barras unitário (*nr_ean13*), código de barras de embalagem (*nr_dun14*) e o fator de caixa.

As tabelas *pack* e *container* possuem função semelhante, mas se apresentam em contextos diferentes. No estoque físico, ambos representam a unidade de armazenamento definida pela gestão do CD, por exemplo, palete, cuba, carrinho. Um *pack* é gerado durante a operação de recebimento. Ele comporta uma quantidade qualquer de um único produto e pode ser armazenado em um endereço. Já o *container* é gerado pela operação de separação de produtos, e pode comportar vários produtos em qualquer nível de sortimento e quantidade. No final do processo de separação, um endereço é solicitado para indicar onde o *container* finalizado ficará alocado até que os processos de consolidação e expedição sejam iniciados.

Na definição do esquema de endereçamento do CD, foi estabelecido um modelo numérico em que as ruas, prédios e apartamentos são identificadas por números mapeados pela tabela “endereço”, além disso, esta tabela também recebe um identificador externo referente ao número do *pack* que ocupa o respectivo espaço físico.

Durante o uso do sistema, os endereços são apresentados concatenando os números de rua, apartamento e prédio, separados por ponto, obtendo assim, o formato “1.2.2”, que representa o endereço Rua 1, Predio 2, Apartamento 2.

O cadastro de clientes e fornecedores gerou a tabela *empresa*, que recebe e-mail, número de CNPJ (Cadastro Nacional da Pessoa Jurídica), telefone, razão social (nome da pessoa jurídica) e o tipo (se é cliente ou fornecedor).

Todas as operações logísticas mapeadas no sistema geraram suas respectivas tabelas, sendo elas: *recebimento*, *movimentacao*, *separacao* e *expedicao* (sem os caracteres “ç” e “ã” devido a uma limitação do *MySQL* que não permite caracteres acentuados nos nomes das tabelas e das colunas). Suas tabelas possuem colunas comuns para data e hora de realização, status da operação (que pode ser pendente, em andamento ou concluída).

As tabelas *separacao* e *expedicao* relacionam-se com a tabela *container* por meio de tabelas de relacionamento, pois em ambas operações, contêineres são processados, portanto, os relacionamentos são necessários.

Já a tabela recebimento possui relação com o *pack*, pois vários *packs* são gerados em uma operação de recebimento.

Uma entidade de pedido foi criada para registrar dados de pedidos de compra e de venda recebidos de outros sistemas conforme foi descrito na seção 7.1. Esta tabela se relaciona com empresa por meio do identificador de empresa e com produto, mediante a uma tabela de relacionamento, pois um pedido possui vários produtos e uma empresa. A relação pedido_produto ainda possui uma coluna para quantidade do respectivo produto.

7.4 FUNCIONAMENTO DA APLICAÇÃO

A interface de usuário do sistema, foi desenvolvida, utilizando-se um *template* fornecido pelo *Thymeleaf*, no qual, ao lado esquerdo da página, há um menu lateral de navegação. Já no lado direito, o conteúdo da página selecionada é carregado, este *template* é apresentado na Figura 27.

Figura 27 – Página de cadastro de endereço.

[Cad. de Produtos](#)
[Cad. de Endereços](#)
[Cad. de Empresas](#)
[Recebimento](#)
[Movimentação](#)
[Separação](#)
[Expedição](#)

Cadastro de endereços

Número da rua:

Número do prédio:

Número do apartamento:

[Cadastrar](#)

Endereços cadastrados

#	Rua	Prédio	Apartamento	Ações
---	-----	--------	-------------	-------

Fonte: Elaborado pelo autor (2023).

De maneira geral, as telas seguem um formato padrão, nas quais há um formulário para o usuário inserir as informações, um ou mais botões para efetuar ações, e uma tabela mostrando os dados cadastrados ou as operações realizadas. A Figura 28 expõe o formato da página da operação Movimentação.

Figura 28 – Página de movimentação de *packs*.

Fonte: Elaborado pelo autor (2023).

A operação do sistema segue este padrão, na qual o usuário insere as informações solicitadas nos formulários, e utiliza os botões para confirmar o registro da operação. Durante as atividades, algumas informações extras podem aparecer na página, por exemplo, os produtos recebidos no processo de Recebimento, ou, os contêineres gerados no processo de Separação, desta forma, o usuário terá ciência do progresso de sua atividade e dos produtos que já manuseou.

A condição inicial, para o usuário efetuar as operações logísticas, exceto a movimentação, é haver um pedido previamente registrado, seja ele de compra para a operação Recebimento, ou de venda para as operações Separação e Expedição. Diferentemente das ações que o usuário pode realizar, o registro de pedidos é efetuado por meio de requisições, oriundas de sistemas externos. Prontamente, o sistema disponibiliza uma API que é capaz de receber requisições contendo os dados necessários para o registro de pedidos.

Os dados devem ser enviados no corpo da requisição no formato JSON, contendo o CNPJ da empresa, o tipo do pedido (VENDA ou COMPRA) e um conjunto de pares de código de barras do produto e a sua quantidade. O corpo da requisição é descrito na Figura 29.

Figura 29 – Corpo padrão da requisição de registro de pedidos.

```
{
  "nrCnpj": "11111111111111",
  "produtos": [
    {
      "nrEan13": "7891001123456",
      "qtde": 10
    },
    {
      "nrEan13": "7891231456789",
      "qtde": 10
    },
    {
      "nrEan13": "7894564789456",
      "qtde": 10
    }
  ],
  "tipoPedido": "COMPRA"
}
```

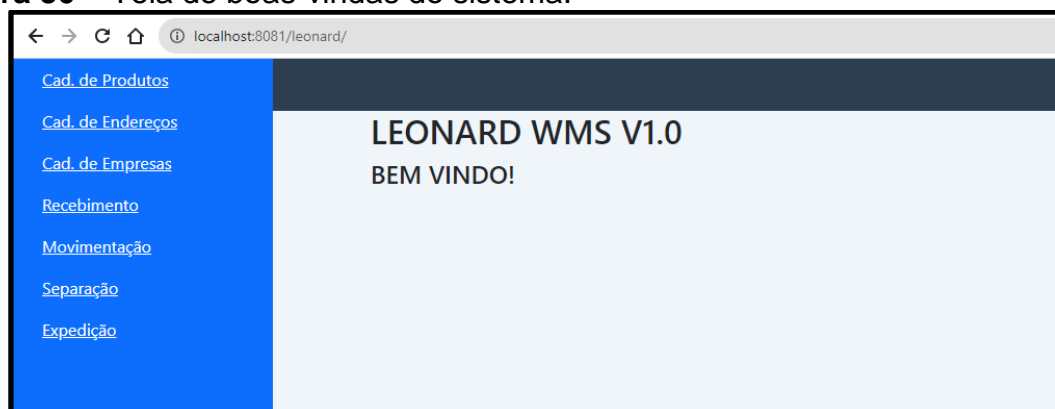
Fonte: Elaborado pelo autor (2023).

7.5 TESTES EM AMBIENTE LOCAL

Como padrão, para realização de testes locais, a porta de rede 8081 foi associada ao sistema, de maneira que, para acessá-lo, é necessário inserir o endereço eletrônico local, seguido da porta 8081 e do sufixo “/leonard” na barra de navegação de qualquer navegador *web*, formando o seguinte endereço: “localhost:8081/leonard”.

Após o acesso inicial, o sistema recebe o usuário em sua tela de boas-vindas, conforme demonstrado na Figura 30. Na tela inicial, apresenta-se a mensagem de boas-vindas, e prontamente, o menu lateral é carregado na página, para que o usuário comece sua operação.

Figura 30 – Tela de boas-vindas do sistema.



Fonte: Elaborado pelo autor (2023).

No primeiro acesso, recomenda-se que, produtos, empresas e endereços sejam cadastrados, pois, a manutenção desses cadastros, é de grande importância para funcionamento do sistema e da operação logística.

Para cadastrar as entidades, é necessário clicar, em uma das opções do menu lateral, nomeadas com o prefixo “Cad.” (abreviatura de cadastro), que o sistema será direcionado para a tela de cadastro desejada. Em seguida, precisa-se preencher, o formulário impresso na tela, com as informações solicitadas. Então, basta clicar, no botão azul indicado como “Cadastrar”, para o sistema realizar o registro da entidade no banco de dados. Este processo é exibido nas figuras 31 e 32.

Figura 31 – Preenchimento do cadastro de empresa.

#	CNPJ	Razão Social	E-mail	Telefone	Ações
---	------	--------------	--------	----------	-------

Fonte: Elaborado pelo autor (2023).

Figura 32 – Empresa registrada no banco de dados.

The screenshot shows a web browser window with the URL `localhost:8081/leonard/empresa/formulario`. The left sidebar contains a menu with the following items: [Cad. de Produtos](#), [Cad. de Endereços](#), [Cad. de Empresas](#) (highlighted with a red arrow), [Recebimento](#), [Movimentação](#), [Separação](#), and [Expedição](#). The main content area is titled 'Cadastro de empresas' and contains the following form fields:

- CNPJ:
- Razão Social:
- E-mail:
- Telefone:
-

Below the form, there is a table titled 'Empresas cadastradas:' with the following data:

#	CNPJ	Razão Social	E-mail	Telefone	Ações
2	11111111111111	Empresa Teste	teste@teste.com	44988887777	Editar

Fonte: Elaborado pelo autor (2023).

Quando houver alterações nos dados cadastrais, pode-se utilizar o botão “Editar” na coluna de ações, para realizar manutenção no cadastro. Desta forma, os dados serão enviados novamente ao formulário, e o usuário poderá alterar, conforme for necessário, e em seguida, clicar no botão “Salvar” para gravar a alteração no banco de dados. Observa-se, esta alteração, nas figuras 33 e 34.

Figura 33 – Início da alteração da empresa.

The screenshot shows the 'Cadastro de empresas' form in the 'Editar' mode. The form fields are pre-filled with the data from the table in Figure 32:

- CNPJ:
- Razão Social:
- E-mail:
- Telefone:
-

Below the form, there is a table titled 'Empresas cadastradas:' with the following data:

#	CNPJ	Razão Social	E-mail	Telefone	Ações
2	11111111111111	Empresa Teste	teste@teste.com	44988887777	Editar

Fonte: Elaborado pelo autor (2023).

Figura 34 – Alteração realizada no cadastro da empresa.

Cadastro de empresas

CNPJ:

Razão Social:

E-mail:

Telefone:

[Cadastrar](#)

Empresas cadastradas:

#	CNPJ	Razão Social	E-mail	Telefone	Ações
2	11111111111111	Empresa Teste 12	teste12@teste12.com	449999999999	Editar

Fonte: Elaborado pelo autor (2023).

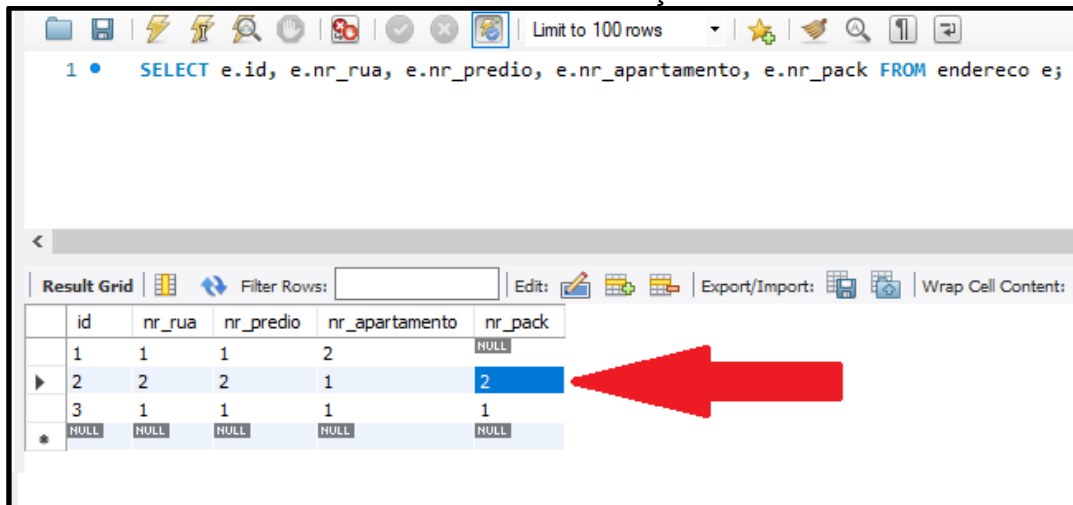
O procedimento descrito, ocorre da mesma maneira nos cadastros de produto e de endereço.

De maneira semelhante, as operações logísticas, também são registradas em formulários fornecidos pelo sistema, tendo como diferença, em relação aos cadastros, não ser possível realizar alterações após o usuário confirmar ou finalizar a operação, clicando no botão “Efetuar”, pois, entende-se que, uma vez concluída, não há como mudar as variáveis de uma operação e nem como desfazê-la. Em casos atípicos, como movimentar um *pack* para o endereço errado, será necessário, realizar a operação novamente, inserindo os dados corretamente.

Para o usuário realizar uma movimentação, é preciso que haja algum endereço desocupado no estoque. O fluxo desta operação, é exposto nas figuras 35, 36, 37 e 38, nas quais a movimentação do *pack* de número 2, é realizada do endereço 2.2.1, para o endereço 1.1.2. A cada estágio do processo, consultas ao banco de dados são exibidas, de maneira que comprovem as mudanças de endereço feitas.

Inicialmente, observa-se na Figura 35, o *pack* de nº 2, estocado no endereço 2.2.1.

Figura 35 – Tabela endereco antes da movimentação.



	id	nr_rua	nr_predio	nr_apartamento	nr_pack
1	1	1	1	2	NULL
2	2	2	2	1	2
3	1	1	1	1	1
*	NULL	NULL	NULL	NULL	NULL

Fonte: Elaborado pelo autor (2023).

Verifica-se que o usuário efetua a operação pelo respectivo formulário, inserindo os dados solicitados, como mostra a Figura 36.

Figura 36 – Início da operação movimentação.



Movimentação

Número do Pack:

Endereço destino:

Rua:

Predio:

Apartamento:

Efetuar

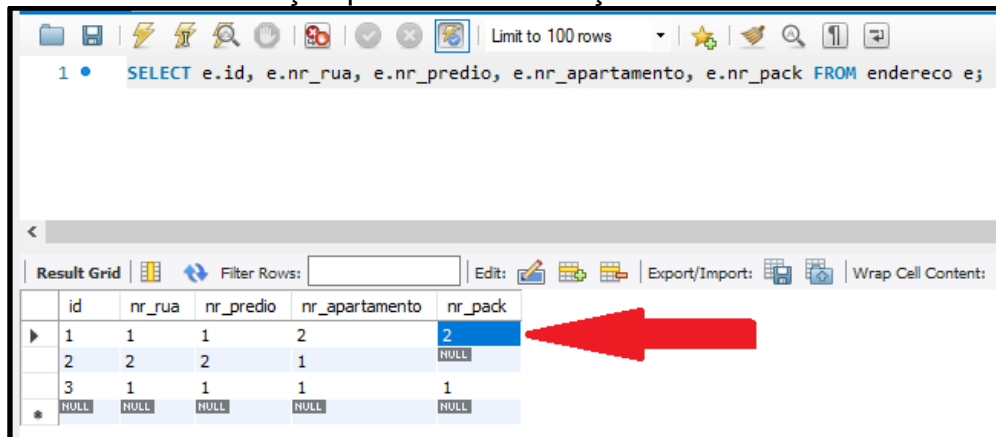
Operações Realizadas

#	Endereço Origem	Endereço Destino	Número Pack
1	1.1.2	1.1.1	1

Fonte: Elaborado pelo autor (2023).

Em seguida, na Figura 37, mostra-se a consulta ao banco de dados, feita logo após o usuário efetuar a operação via sistema.

Figura 37 – Tabela endereço após a movimentação.



	id	nr_rua	nr_predio	nr_apartamento	nr_pack
1	1	1	2	2	2
2	2	2	1	1	NULL
3	1	1	1	1	1
*	NULL	NULL	NULL	NULL	NULL

Fonte: Elaborado pelo autor (2023).

Por fim, a Figura 38 confirma o registro da operação, expondo a tela do sistema após o usuário efetuá-la.

Figura 38 – Movimentação efetuada.



Movimentação

Número do Pack:

Endereço destino:

Rua:

Predio:

Apartamento:

Efetuar

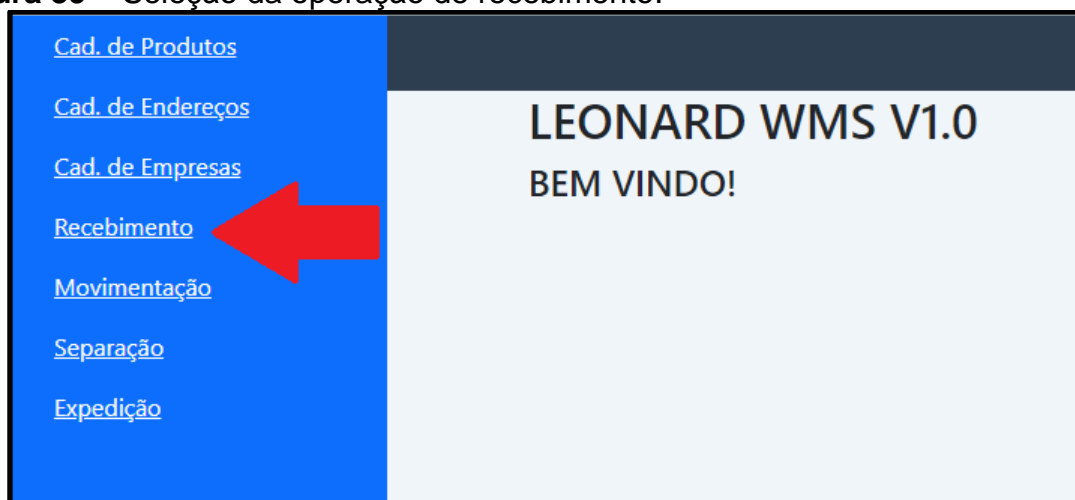
Operações Realizadas

#	Endereço Origem	Endereço Destino	Número Pack
1	1.1.2	1.1.1	1
2	2.2.1	1.1.2	2

Fonte: Elaborado pelo autor (2023).

Na operação de recebimento, o processo ocorre da seguinte maneira, após o sistema receber um pedido de compra, este pedido é disponibilizado ao usuário após clicar na opção “Recebimento” do menu lateral, conforme as figuras 39 e 40.

Figura 39 – Seleção da operação de recebimento.



Fonte: Elaborado pelo autor (2023).

Figura 40 – Tela de seleção de pedido.

Pedidos pendentes de recebimento:							
Nº do Pedido	Data de Emissão	Fornecedor	Tipo	Status	Nº de produtos	Volumes	Ações
1	19/11/2023 16:35	Tarara Ltda.	COMPRA	ABERTO	3	90	Receber
2	19/11/2023 16:47	Empresa Teste S.A.	COMPRA	ABERTO	3	90	Receber

Pedidos recebidos:							
Nº do Pedido	Data de Emissão	Fornecedor	Tipo	Status	Nº de produtos	Volumes	
Vazio.							

Fonte: Elaborado pelo autor (2023).

A tela de seleção de pedido fornece ao usuário, dados do pedido, como: data de emissão, nome do fornecedor, tipo, *status*, quantidade de produtos e quantidade de volumes.

O usuário poderá iniciar o processo de recebimento do pedido clicando no botão “Receber”, ao lado direito da página.

A tela de recebimento de produtos mostra a lista de produtos do pedido de compra e suas respectivas quantidades, ela também fornece dois campos para o usuário inserir a quantidade recebida e o número do *pack*, e um botão para confirmar o recebimento do produto.

Após o usuário preencher os campos e clicar no botão “Receber”, os produtos são mostrados na parte inferior da página, confirmando seu recebimento. As figuras 41 e 42 demonstram este processo passo a passo.

Figura 41 – Tela de recebimento de produtos.

Recebimento

Nº do Pedido: 1

Status da operação: EM_ANDAMENTO

#	Descrição	EAN13	DUN14	Qtde Pedido	Qtde Física	Pack	Ações
2	Arroz Branco Tipo 1 5kg	7891231456789	17891231456789	25	<input type="text" value="25"/>	<input type="text" value="1"/>	<button>Receber</button>
3	Iogurte Sabor Ameixa Zero Lactose	7894564789456	17894564789456	50	<input type="text" value="50"/>	<input type="text" value="2"/>	<button>Receber</button>
1	Leite UHT 1L Integral	7891001123456	17891001123456	15	<input type="text" value="Qtde Física"/>	<input type="text" value="Nº do Pack"/>	<button>Receber</button>

Packs recebidos:

Nº do Pack	Id Produto	Descrição	EAN13	DUN14	Qtde	Ações

Fonte: Elaborado pelo autor (2023).

Figura 42 – Produtos recebidos.

Recebimento

Nº do Pedido: 1

Status da operação: EM_ANDAMENTO

#	Descrição	EAN13	DUN14	Qtde Pedido	Qtde Física	Pack	Ações
2	Arroz Branco Tipo 1 5kg	7891231456789	17891231456789	25	<input type="text" value="Qtde Física"/>	<input type="text" value="Nº do Pack"/>	<button>Receber</button>
3	Iogurte Sabor Ameixa Zero Lactose	7894564789456	17894564789456	50	<input type="text" value="Qtde Física"/>	<input type="text" value="Nº do Pack"/>	<button>Receber</button>
1	Leite UHT 1L Integral	7891001123456	17891001123456	15	<input type="text" value="Qtde Física"/>	<input type="text" value="Nº do Pack"/>	<button>Receber</button>

Packs recebidos:

Nº do Pack	Id Produto	Descrição	EAN13	DUN14	Qtde	Ações
1	2	Arroz Branco Tipo 1 5kg	7891231456789	17891231456789	25	<button>Estornar Recebimento</button>
2	3	Iogurte Sabor Ameixa Zero Lactose	7894564789456	17894564789456	50	<button>Estornar Recebimento</button>

Fonte: Elaborado pelo autor (2023).

Caso o usuário observe algum erro durante o processo de recebimento dos produtos, ele pode clicar em “Estornar Recebimento”, para poder refazer.

Após o último produto ser recebido, o processo encerra-se automaticamente pelo sistema, e o usuário é redirecionado a tela de seleção de pedidos. Neste momento, observa-se o pedido recebido na parte inferior da tela, no qual o *status* foi alterado para “RECEBIDO”, confirmando que a operação foi finalizada, conforme é descrito na Figura 43.

Figura 43 – Retorno para a tela seleção de pedidos.



Pedidos pendentes de recebimento:							
Nº do Pedido	Data de Emissão	Fornecedor	Tipo	Status	Nº de produtos	Volumes	Ações
2	19/11/2023 17:09	Empresa Teste S.A.	COMPRA	ABERTO	3	90	Receber

Pedidos recebidos:						
Nº do Pedido	Data de Emissão	Fornecedor	Tipo	Status	Nº de produtos	Volumes
1	19/11/2023 17:09	Tarara Ltda.	COMPRA	RECEBIDO	3	90

Fonte: Elaborado pelo autor (2023).

8 CONSIDERAÇÕES FINAIS

A proposta da presente pesquisa foi desenvolver uma aplicação capaz de gerenciar as operações de um CD, utilizando recursos e ferramentas de desenvolvimento de *software* capazes de descomplicar o processo de estruturação e codificação do sistema.

Considerando a necessidade de se entender o funcionamento interno de um CD, foi realizada uma revisão bibliográfica a respeito do assunto. A partir desta revisão, um levantamento de requisitos de *software* foi elaborado, contendo um diagrama de casos de uso e um quadro de requisitos funcionais e não funcionais. Dessa forma, o levantamento serviu de base para o início do desenvolvimento do projeto.

Em seguida, foi necessário selecionar a linguagem de programação e as ferramentas necessárias para o desenvolvimento do sistema. Após esta definição, iniciou-se, então, a elaboração do *software*.

Inicialmente, a linguagem Java se mostrou bastante verbosa, exigindo muita escrita de código para operações simples. Por outro lado, a aplicação dos recursos fornecidos pelo *Spring Framework*, simplificou muito este processo, possibilitando a produção rápida de um código simples e enxuto. O *framework* possibilitou, ainda, boa integração entre *front-end* e *back-end*. Ele também forneceu, uma estrutura de *software* robusta e pouco acoplada, se mostrando eficiente e permitindo que erros fossem facilmente corrigidos e novas funcionalidades fossem rapidamente desenvolvidas.

O *Spring Data JPA*, possibilitou o mapeamento objeto-relacional entre as classes modelo e as tabelas do banco de dados e, ainda, a criação do banco de dados pelo conjunto JPA–Hibernate, sendo favorável ao processo de desenvolvimento da aplicação.

Com o uso do *Thymeleaf*, foi possível integrar a interface de usuário com a aplicação, de maneira que informações da tela fossem processadas pelo sistema e que as operações do CD fossem devidamente registradas no banco de dados, gerando assim, boa comunicação com o *Spring Framework*.

Uma *API* foi disponibilizada pelo sistema, para que outras aplicações possam solicitar a criação de pedidos, representando uma solicitação de compra ou de venda de produtos.

Tendo em vista que, os requisitos de *software* foram completamente atendidos, entende-se que a aplicação elaborada atendeu a proposta do trabalho, pois as funcionalidades desenvolvidas cobriram as operações básicas de um CD.

Como trabalhos futuros, sugere-se, a implementação de relatórios logísticos como: estoque sumarizado por produto e ocupação total do armazém; melhoria na interface de usuário com a aplicação do CSS; implantação de medidas de segurança fornecidas pelo *Spring Security*, como: autenticação de usuário e permissão de acesso conforme o tipo de usuário; e, por fim, a implementação de validações de dados com mensagens de erro ao usuário.

REFERÊNCIAS

ALVARENGA, Antonio Carlos; NOVAES, Antonio Galvão N. **Logística aplicada: suprimento e distribuição física**. 3. ed. São Paulo: Blucher, 2000.

ALYRIO, Rovigati Danilo. **Métodos e Técnicas de Pesquisa em Administração**. Rio de Janeiro: Fundação CECIERJ, 2009. 281 p.

AMAZON. **Qual é a diferença entre front-end e back-end no desenvolvimento de aplicações?** 2023a. Disponível em: <https://aws.amazon.com/pt/compare/the-difference-between-frontend-and-backend/>. Acesso em: 05 set. 2023.

AMAZON. **O que é uma API?** 2023b. Disponível em: <https://aws.amazon.com/pt/what-is/api/>. Acesso em: 19 set. 2023.

ANTUNES, Daniel. **Desenvolvimento Back-end**: um guia para empreendedores. 2020. Disponível em: <https://www.gobacklog.com/blog/back-end-guia-para-empresarios/>. Acesso em: 12 set. 2023.

ARAUJO, Camila Chicherchio de et. al. **Gerenciamento de Banco de Dados: Análise Comparativa de SGBDs**. 2023. Disponível em: <https://www.devmedia.com.br/gerenciamento-de-banco-de-dados-analise-comparativa-de-sgbd-s/30788>. Acesso em: 14 jun. 2023.

AWARI (Brasil). **Banco de Dados Não Relacional (NoSQL)**: o que é e principais tipos. 2022. Disponível em: <https://awari.com.br/banco-de-dados-nao-relacional/>. Acesso em: 17 jun. 2023.

BALLOU, Ronald H. **Gerenciamento da cadeia de suprimentos/logística empresarial**. 5 ed. Porto Alegre: Bookman, 2006.

BALLOU, Ronald H.. **Logística empresarial**: transportes, administração de materiais e distribuição física. São Paulo: Atlas, 2013. 388 p.

BARBOZA, Victor. **Programação**: o que são frameworks e por que utilizá-los?. 2022. Disponível em: <https://izap.com.br/blog/programacao-o-que-sao-frameworks-e-por-que-utiliza-los/>. Acesso em: 15 ago. 2023.

BARROS, Monica Coutinho. **WMS no gerenciamento de depósitos, armazéns e centros de distribuição**. 2005a. Disponível em: <https://www.ilos.com.br/web/wms-no-gerenciamento-de-depositos-armazens-e-centros-de-distribuicao>. Acesso em: 02 abr. 2023.

BARROS, Monica Coutinho. **Warehouse Management System (WMS)**: conceitos teóricos e implementação em um centro de distribuição. 2005b. 132 f. Dissertação (Mestrado) - Curso de Pós-Graduação em Engenharia Industrial, Departamento de Engenharia Industrial, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2005.

BECKEDORFF, Irzo Antonio. **Logística de suprimentos e distribuição**. Indaial: Uniasselvi, 2013. 235 p.

BESSA, André. **Java**: o que é, linguagem e um guia para iniciar na tecnologia. 2023. Disponível em: <https://www.alura.com.br/artigos/java>. Acesso em: 19 set. 2023.

BOWERSOX, D. J. et al. **Gestão logística da cadeia de suprimentos**. 4. ed. Porto Alegre: AMGH, 2014. 472 p.

CHIAVENATO, Idalberto. **Gestão da produção**: uma abordagem introdutória. 3. ed. Barueri: Manole, 2014.

CHING, Hong Yuh. **Gestão de estoques na cadeia de logística integrada - Supply Chain**. 4. ed. São Paulo: Atlas, 2010. 237 p.

DATE, Christopher J.. **Introdução a sistemas de banco de dados**. Rio de Janeiro: Elsevier, 2003. Tradução de Daniel Vieira.

DEITEL, Paul; DEITEL, Harvey. **Java: como programar**. 10. ed. São Paulo: Pearson, 2017. 934 p.

DELAGE. **Quais as funcionalidades do sistema WMS?** 2023. Disponível em: <https://delage.com.br/blog/quais-as-funcionalidades-do-sistema-wms/>. Acesso em: 30 maio 2023.

DEVMEDIA. **O que é front-end e back-end?** 2021. Disponível em: <https://www.devmedia.com.br/o-que-e-front-end-e-back-end/43216>. Acesso em: 19 ago. 2023.

ELMASRI, Ramez; NAVATHE, Shamkant B.. **Sistemas de banco de dados**. 6. ed. São Paulo: Pearson, 2011. 788 p.

EQUIPE TOTVS. **WMS: o que é, funcionalidades e vantagens**. 2022 Disponível em: <https://www.totvs.com/blog/atacadista-distribuidor/wms/>. Acesso em: 30 maio 2023.

ESPINOSA, Luiz. **JavaScript para iniciantes: fundamentos, desenvolvimento front-end e como evitar os erros mais comuns**. 2023. Disponível em: <https://www.dio.me/articles/javascript-para-iniciantes-fundamentos-desenvolvimento-front-end-e-como-evitar-os-erros-mais-comuns>. Acesso em: 07 set. 2023.

EWALLY. **Back-end: o que é, para que serve e quais suas linguagens?** 2023. Disponível em: <https://www.ewally.com.br/blog/ajudando-sua-empresa/backend>. Acesso em: 12 set. 2023.

FELIPE, Alex. **Deixando o código mais simples com refatoração no Java**. 2017. Disponível em: <https://www.alura.com.br/artigos/deixando-o-codigo-mais-simples-com-refatoracao-no-java>. Acesso em: 19 ago. 2023.

FINCO, Nina. **WMS na logística: o que é, para que funciona e como aplicar**. 2020. Disponível em: <https://www.cobli.co/blog/o-que-e-wms/>. Acesso em: 30 maio 2023.

FONTOURA, Augusto. **Framework**: qual sua função, tipos e importância! 2023. Disponível em: <https://www.fm2s.com.br/blog/framework-funcao-e-importancia>. Acesso em: 16 ago. 2023.

FRANKLIN, Ronaldo. **Conhecimentos de Movimentação e Armazenagem**. Rio de Janeiro. E-Quality Núcleo de Treinamento e Pesquisa da Consultoria InfoJBS, 2003.

GIL, Antônio Carlos. **Como Elaborar Projetos De Pesquisa**. 4. ed. São Paulo: Atlas, 2002. 175 p.

GOCACHE. **O que é Web API? E seus principais usos**. 2022. Disponível em: <https://www.gocache.com.br/seguranca/o-que-e-web-api/>. Acesso em: 19 set. 2023.

KONDADO TEAM (Brasil). **O que é um banco de dados não relacional?** 2022. Disponível em: <https://kondado.com.br/blog/blog/2022/11/10/o-que-e-um-banco-de-dados-nao-relacional/>. Acesso em: 17 jun. 2023.

KRIGER, Daniel. **O QUE SÃO FRAMEWORKS E POR QUE SÃO IMPORTANTES PARA OS DEVS**. 2022. Disponível em: <https://kenzie.com.br/blog/framework/>. Acesso em: 15 ago. 2023.

KRIGER, Brunno. **Front End**: entenda o que é, para que serve, como aprender essa especialidade! 2023. Disponível em: <https://kenzie.com.br/blog/front-end/>. Acesso em: 07 set. 2023.

LACERDA, Leonardo. **Armazenagem Estratégica**: analisando novos conceitos. 2000. Disponível em: <https://www.ilos.com.br/web/armazenagem-estrategica-analisando-novos-conceitos>. Acesso em: 02 abr. 2023.

MACHADO, Amanda. **Qual a diferença entre front-end e back-end?** 2021. Disponível em: <https://tecnoblog.net/responde/qual-a-diferenca-entre-front-end-e-back-end/>. Acesso em: 03 set. 2023.

MELO, Diego. **O que é uma API? [Guia para iniciantes]**. 2021. Disponível em: <https://tecnoblog.net/responde/o-que-e-uma-api-guia-para-iniciantes/>. Acesso em: 19 set. 2023.

MÉTODO PROGRAMAR. **Como Criar Uma Página de Login Com HTML e CSS**. 2023. Disponível em: <https://metodoprogramar.com.br/como-criar-uma-pagina-de-login-com-html-e-css/>. Acesso em: 07 set. 2023.

NAZÁRIO, Paulo. **A importância de sistemas de informação para a competitividade logística**. 1999. Disponível em: <https://www.ilos.com.br/web/a-importancia-de-sistemas-de-informacao-para-a-competitividade-logistica>. Acesso em: 28 mar. 2023.

NOGUEIRA, Amarildo de Souza. **Logística Empresarial: um guia prático de operações logísticas**. 2. ed. São Paulo: Atlas, 2018. 250 p. Disponível em: <https://app.minhabiblioteca.com.br/#/books/9788597015553>. Acesso em: 31 maio 2023.

NOVAES, Antonio Galvão. **Logística e gerenciamento da cadeia de distribuição**. 3. ed. Rio de Janeiro: Elsevier, 2007.

NUNES, Rogério da Silva; PLATT, Allan Augusto. **Gestão Logística**. Florianópolis: Departamento de Ciências da Administração/UFSC, 2008. 114 p.

PINTO, André. **Entenda qual a importância do banco de dados e conheça suas vantagens**. 2022. Disponível em: <https://blog.vibetecnologia.com/banco-de-dados/>. Acesso em: 21 jun. 2023.

PLATT, Allan Augusto. **Logística e cadeia de suprimentos**. 3. ed. Florianópolis: Departamento de Ciências da Administração/UFSC, 2015. 116 p.

PRESSMAN, Roger S.; MAXIM, Bruce R. **Engenharia de software: uma abordagem profissional**. 8. ed. Porto Alegre: AMGH, 2016. 940 p.

RAMAKRISHNAN, Raghu; GEHRKE, Johannes. **Sistemas de Gerenciamento de Banco de Dados**. 3. ed. Porto Alegre: McGraw-Hill, 2011. 884 p.

REDAÇÃO TERA. **Front-end e Back-end**: entenda essas duas áreas do desenvolvimento. 2023. Disponível em: <https://blog.somostera.com/carreiras-digitais/front-end-e-back-end>. Acesso em: 07 set. 2023.

RED HAT. **O que é API?** 2023. Disponível em: <https://www.redhat.com/pt-br/topics/api/what-are-application-programming-interfaces>. Acesso em: 19 set. 2023.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S.. **Sistema de Banco de Dados**. 6. ed. Rio de Janeiro: Elsevier, 2012. 861 p.

SILVA, Fabrício Costa. **Desenvolvimento De Um Sistema De Controle De Tcc Para A Uesb Utilizando Js e Spring**. 2010. 50 f. TCC (Graduação) - Curso de Ciência da Computação, Universidade Estadual do Sudoeste da Bahia, Vitória da Conquista, 2010.

SOMMERVILLE, Ian. **Engenharia de Software**. 9. ed. São Paulo: Pearson Prentice Hall, 2011. 529 p.

TEJADA, Zoiner. **Dados não relacionais e NoSQL**. 2023. Disponível em: <https://learn.microsoft.com/pt-br/azure/architecture/data-guide/big-data/non-relational-data>. Acesso em: 17 jun. 2023.

VERAS, Davi. **O que é uma API? [Definição e exemplos]**. 2022. Disponível em: <https://blog.cod3r.com.br/o-que-e-uma-api/>. Acesso em: 19 set. 2023.