

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

DANRLEY SCHARF

**APLICAÇÃO WEB PARA GERENCIAMENTO DE ENTREGAS DE
TRANSPORTADORA**

TRABALHO DE CONCLUSÃO DE CURSO

**PATO BRANCO
2018**

DANRLEY SCHARF

**APLICAÇÃO WEB PARA GERENCIAMENTO DE ENTREGAS DE
TRANSPORTADORA**

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Conclusão de Curso 2, do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, da Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco, como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Prof. Vinicius Pegorini

**PATO BRANCO
2018**



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Pato Branco
Departamento Acadêmico de Informática
Curso de Tecnologia em Análise e Desenvolvimento
de Sistemas



TERMO DE APROVAÇÃO

TRABALHO DE CONCLUSÃO DE CURSO

APLICAÇÃO WEB PARA GERENCIAMENTO DE ENTREGAS DE TRANSPORTADORA

POR

DANRLEY SCHARF

Este trabalho de conclusão de curso foi apresentado no dia 13 de dezembro de 2018, como requisito parcial para obtenção do título de Técnico em Análise e Desenvolvimento de Sistemas, pela Universidade Tecnológica Federal do Paraná. O acadêmico foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Banca examinadora:

Prof MSc. Vinicius Pegorini
Orientador

Profª Drª Beatriz Terezinha Borsoi

Prof. Dr. Edilson Pontarolo

Prof. Dr. Edilson Pontarolo
Coordenador do Curso de Tecnologia em
Análise e Desenvolvimento de Sistemas

Profª Drª Beatriz Terezinha Borsoi
Responsável pela Atividade de Trabalho de
Conclusão de Curso

A Folha de Aprovação assinada encontra-se na Coordenação do Curso.

RESUMO

SCHARF, Danrley. Aplicação Web para gerenciamento de entregas de transportadora. 2018. 50f. Monografia (Trabalho de Conclusão de Curso) - Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2018.

O ramo de transportes tem um papel fundamental na economia brasileira, visto que os produtos consumidos pela população precisam ser distribuídos regularmente, visando não gerar desabastecimento, além disso, muitos dos produtos transportados são perecíveis e/ou prazo de validade curto. Com o crescimento do comércio eletrônico, as transportadoras precisam se especializar para manter um controle rigoroso de entregas de forma a transmitir segurança aos consumidores que o produto será entregue nas condições acordadas (prazo, qualidade quantidade) e também ao comércio eletrônico. A facilidade de acesso às tecnologias como *Global Positioning System* (GPS) e à internet pode facilitar o controle de encomendas e, ao mesmo, tempo reduzir os custos das transportadoras. O sistema desenvolvido neste trabalho utiliza recursos tecnológicos para a obtenção da localização atual do usuário, facilitando e agilizando o trabalho das transportadoras.

Palavras-chave: Transportadora. Sistema Web. GPS.

ABSTRACT

SCHARF, Danrley. Web application to manager shipping companies. 2018. 50f. Monografia (Trabalho de Conclusão de Curso) - Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2018.

The transport sector has a fundamental role in the Brazilian economy. The products consumed have been distributed regularly, in order to serve an entire population. With the growth of electronic commerce, the shipping companies need to specialize their services and thus maintain strict control of deliveries to offer a security job. The ease access to technologies as Global Positioning System (GPS) and internet access can facilitate the control of orders, and even reduce time and cost. The system developed in this work uses some technologies, such as obtaining the current location of the user, aiming to facilitate the work of shipping companies.

Keywords: Shipping company. Web Sistem. GPS.

LISTA DE FIGURAS

Figura 1 - Diagrama de casos de uso	24
Figura 2 - Diagrama de classes.....	26
Figura 3 - Diagrama de Entidades Relacionamentos	27
Figura 4 - Tela principal do sistema	32
Figura 5 - Lista de viagens	33
Figura 6 - Vincular encomendas à viagem	33
Figura 7 - Mensagem de confirmação	34
Figura 8 - Rastreamento da encomenda	34
Figura 9 - Obtendo a localização do motorista	35
Figura 10 - Relatório de encomendas por viagem.....	36
Figura 11 - Relatório de despesas por viagem.....	37
Figura 12 - Estrutura de pacotes	38
Figura 13 - Páginas da aplicação	38

LISTA DE QUADROS

Quadro 1 - Lista de Ferramentas e Tecnologias	17
Quadro 2 - Iterações de desenvolvimento.....	20
Quadro 3 - Requisitos Funcionais	23
Quadro 4 - Operação incluir dos casos de uso de cadastro.....	25
Quadro 5 - Operação alterar dos casos de uso de cadastro.....	25
Quadro 6 - Operação excluir dos casos de uso de cadastro.....	26
Quadro 7 - Campos da tabela Estados	27
Quadro 8 - Campos da tabela Cidades.....	28
Quadro 9 - Campos da tabela Entidades	28
Quadro 10 - Campos da tabela CentrosDistribuicao	28
Quadro 11 - Campos da tabela Cargos.....	28
Quadro 12 - Campos da tabela Funcionarios.....	29
Quadro 13 - Campos da tabela Permissoes_Funcionarios	29
Quadro 14 - Campos da tabela Permissoes.....	29
Quadro 15 - Campos da tabela Veiculos.....	30
Quadro 16 - Campos da tabela Viagens	30
Quadro 17 - Campos da tabela HistoricosViagens.....	30
Quadro 18 - Campos da tabela TiposDespesa	30
Quadro 19 - Campos da tabela Despesas	31
Quadro 20 - Campos da tabela Encomendas	31
Quadro 21 - Campos da tabela Encomendas_Viagens	31
Quadro 22 - Campos da tabela Entregas.....	32

LISTAGENS DE CÓDIGO

Listagem 1 - Configurações da conexão com a base de dados.....	39
Listagem 2 – Dependências importadas pela aplicação	40
Listagem 3 – Função que vincula a encomenda à viagem selecionada.....	41
Listagem 4 – Função que desvincula a encomenda à viagem selecionada.....	42
Listagem 5 – Tabela com os dados da viagem selecionada	42
Listagem 6 – Criação da tabela de apresentação de encomendas.....	43
Listagem 7 – Corpo da tabela com os dados das encomendas.....	43
Listagem 8 – Injeção de dependências e declaração de variáveis.....	44
Listagem 9 – Método de listagem da viagem e encomendas.....	44
Listagem 10 – Função que vincula as encomendas à viagem	45
Listagem 11 – Identificação dos dados selecionados	45
Listagem 12 – Teste para verificar se a encomenda já está vinculada à viagem.....	46
Listagem 13 – Método para desvincular encomenda da viagem.....	47

LISTA DE SIGLAS

CASE	<i>Computer-Aided Software Engineering</i>
GPS	<i>Global Positioning System</i>
IDE	<i>Integrated Development Environment</i>
JDT	<i>Java Development Tools</i>
MSSQL	<i>Microsoft SQL Server</i>
MySQL	<i>My Structure Query Language</i>
SDK	<i>Software Development Kit</i>
UML	<i>Unified Modeling Language</i>
WTP	<i>Web Tools Platform</i>

SUMÁRIO

1 INTRODUÇÃO	10
1.1 CONSIDERAÇÕES INICIAIS	10
1.2 OBJETIVOS	11
1.2.1 Objetivo Geral	11
1.2.2 Objetivos Específicos	11
1.3 JUSTIFICATIVA	12
1.4 ESTRUTURA DO TRABALHO	13
2 AS TRANSPORTADORAS E O CRESCIMENTO DO E-COMMERCE	14
2.1 TRANSPORTADORAS E SEUS DESAFIOS	14
2.2 O CRESCIMENTO DO E-COMMERCE	15
3 MATERIAIS E MÉTODOS	17
3.1 MATERIAIS	17
3.1.1 Astah Community:	17
3.1.2 DBDesigner	18
3.1.3 PostgreSQL	18
3.1.4 Eclipse IDE	18
3.1.5 Spring Framework	19
3.1.6 Java EE	19
3.2 MÉTODO.....	20
4 RESULTADO	22
4.1 ESCOPO DO SISTEMA	22
4.2 MODELAGEM DO SISTEMA	22
4.3 APRESENTAÇÃO DO SISTEMA	32
4.4 IMPLEMENTAÇÃO DO SISTEMA	37
5 CONSIDERAÇÕES FINAIS	48
REFERÊNCIAS	49

1 INTRODUÇÃO

Este capítulo apresenta as considerações iniciais, os objetivos e a justificativa da realização deste trabalho. O texto é finalizado com a apresentação dos capítulos subsequentes.

1.1 CONSIDERAÇÕES INICIAIS

O uso de sistemas web está cada vez mais presente no controle dos processos das empresas. E não é diferente no ramo dos transportes, pois essas empresas necessitam de comunicação contínua com seus centros de distribuição e os veículos de transporte.

Por meio de sistemas web as empresas de transportes conseguem fornecer dados como, por exemplo, a situação de uma encomenda enviada. Uma vez que o usuário poderá acessar as informações a partir de qualquer computador, *tablet* ou *smartphone*.

Os dispositivos móveis com suas funções de *Global Positioning System* (GPS) estão facilitando as rotinas dessas empresas. Um motorista pode em questão de segundos passar sua localização para o servidor, atualizando a cidade em que o caminhão se encontra e quais encomendas estão com ele, bem como uma previsão de chegada ao centro de distribuição.

Segundo Pirola (2016), o Brasil é o país que lidera a participação de compras no comércio eletrônico (e-commerce) entre os países latinos. Isso mostra a evolução e o crescimento constante do comércio eletrônico e que as pessoas estão preferindo a comodidade de suas casas para comprar produtos de bens duráveis, como eletrônicos, artigos de vestuário, entre outros. O aumento do comércio eletrônico exige, conseqüentemente, maior solicitação dos serviços logísticos ofertados pelas transportadoras.

As empresas que realizam vendas por meio de e-commerce necessitam de um controle rigoroso de entregas e que tanto as empresas como seus clientes possam consultar a situação de sua entrega. O acompanhamento permite o cliente acompanhar o cronograma previsto para entrega.

Esse contexto ressalta a importância de um sistema *web* para a transportadora realizar o gerenciamento da empresa e seus processos, bem como de um sistema para dispositivos móveis para o gerenciamento das encomendas. Nesse trabalho pretende-se desenvolver um sistema com soluções específicas para cada rotina, tais como cadastros de clientes, fornecedores, funcionários e veículos, gerenciamento das entregas com o controle dos gastos e lucros de cada viagem. Esses dados visam facilitar a execução das atividades administrativas no dia a dia destas empresas e, conseqüentemente, agilizar o processo de entrega das encomendas e a transparência para o cliente que realizou a compra do produto por meio de um site de *e-commerce*.

1.2 OBJETIVOS

A seguir estão os objetivos gerais e específicos deste trabalho.

1.2.1 Objetivo Geral

Desenvolver uma aplicação *web* responsiva, possibilitando o gerenciamento e a localização de encomendas de transportadoras.

1.2.2 Objetivos Específicos

- Facilitar o controle dos processos básicos da transportadora, como cadastros de clientes, fornecedores, funcionários e veículos, gerenciamento das entregas com o controle dos gastos e do lucro de cada viagem;
- Permitir que os consumidores consigam rastrear as suas respectivas encomendas;
- Disponibilizar a consulta de informações via aplicação Web de modo responsivo, possibilitando a consulta por dispositivos móveis;
- Possibilitar aos motoristas disponibilizarem informações sobre o transporte e sua localização.

1.3 JUSTIFICATIVA

Conforme informações do relatório WebShoppers (Ebit 2016a), houve um aumento no número de pessoas que realizam compras através de dispositivos móveis, o que está auxiliando na pesquisa de preços de produtos *online*:

“Atualmente, os consumidores têm amplo acesso a dados como preço, características técnicas e reputação de lojas e produtos. Usar o dispositivo móvel ao longo do processo de decisão de compra, enquanto assiste a um comercial de TV, caminha num shopping center ou dentro de lojas físicas, potencializa o acesso à informação, não necessariamente para comprar, mas sim para pesquisar e consultar preços. A democratização da informação mudou completamente a forma de comprar.” (Ebit 2016a)

Segundo o estudo realizado pela Revista Gestão & Tecnologia, entre os motivos para que o consumidor realize compras por meio de e-commerce, há algumas variáveis determinantes que o influenciam, como é possível verificar neste trecho do estudo:

“Entre as variáveis situacionais que influenciam a adoção do comércio eletrônico, destacam-se as pressões de tempo, muitas vezes associadas a longas jornadas de trabalho, pois uma das vantagens de se utilizar este modelo de negócios é justamente a economia de tempo e a flexibilidade de horário (Wolfenbarger & Gilly, 2001). Além disso, quando os pontos de venda de determinados produtos estão localizados a uma distância geográfica relativamente grande dos consumidores, ou quando estes não possuem condições de se locomover até o local de vendas, os mesmos podem optar pelo uso da Internet como forma de reduzir os obstáculos espaciais existentes (Perea et al., 2004).” (CERIBELI; OLIVEIRA; FELIPE, 2015, p.181)

Com um software web as transportadoras poderão melhorar o atendimento às empresas sejam tradicionais ou de comércio eletrônico, pois estará sendo automatizado o processo de rastreamento das encomendas.

Ao cadastrar ser cadastrada uma encomenda para ser entregue, tanto o vendedor (empresa) quanto o respectivo cliente (comprador) poderá consultar a

situação da encomenda. Isso pode reduzir custos das transportadoras, pois não necessitará manter uma central de atendimento para que possam ligar para saber a localização de encomendas. Reduzindo, assim, despesas com pessoal e operacionais dessa central de atendimento (água, luz, aluguel, telefone, entre outros).

1.4 ESTRUTURA DO TRABALHO

O referido trabalho está estruturado em quatro capítulos, sendo que o primeiro capítulo apresentou as considerações iniciais, os objetivos e a justificativa. O segundo capítulo apresenta o referencial teórico, contendo uma introdução sobre as principais necessidades das transportadoras e o crescimento do comércio eletrônico. O terceiro capítulo apresenta os materiais e o método utilizados a modelagem e o desenvolvimento do sistema. O quarto capítulo apresenta os diagramas desenvolvidos para a modelagem do sistema e a codificação das principais funcionalidades da aplicação. O quinto e último capítulo apresenta as considerações finais e, em seguida, estão as referências utilizadas no texto.

2 AS TRANSPORTADORAS E O CRESCIMENTO DO E-COMMERCE

2.1 TRANSPORTADORAS E SEUS DESAFIOS

Ao realizar uma compra de um produto por meio da internet, o consumidor pode não estar ciente de todas as etapas do processo envolvido na entrega daquele produto, para que tudo ocorra conforme prazos estabelecidos no momento em que a compra foi efetuada. Existe um processo de logística envolvendo a loja que efetuou a venda e a transportadora contratada para que a entrega seja realizada.

O consumidor sempre espera que a entrega seja realizada de forma pontual, com pouco ou nenhum atraso, independente dos problemas enfrentados pela transportadora no processo ou se a ela se encontra sobrecarregada (BOWERSOX et al., 2013).

Segundo Bowersox et al. (2013), embora a logística seja utilizada desde o começo da civilização, ela precisa se reinventar e aplicar novas práticas para se adaptar ao século XXI, gerando, assim, uma espécie de renascimento do processo logístico.

Assim como os produtos sofrem mudanças ao logo do tempo, visando melhorar sua qualidade, longevidade, atender necessidades e interesses com consumidores, o processo de logística também vem sofrendo alterações com o passar dos anos. Essas mudanças visam facilitar a entrega dos pedidos e reduzir custos para a transportadora até que os mesmos cheguem ao seu destino.

Caso o consumidor efetue uma compra e seu produto não chegue no prazo programado é a empresa que efetuou a venda que sofre os prejuízos da demora nas entregas. Muitas vezes, o consumidor não sabe nem qual é a transportadora que está realizando a entrega do seu produto.

Novaes (2014) destaca que o efeito negativo desse tipo de situação que prejudica a imagem da empresa, ainda não foi convenientemente avaliado no país, mas é, sem dúvida, significativo.

2.2 O CRESCIMENTO DO E-COMMERCE

O aumento no número de usuários de internet no Brasil aumentou muito desde o início dos anos 2000. Para Ceribeli, Oliveira e Felipe (*apud* Ecommerce, 2013), o número de pessoas com acesso à internet no país era de apenas 10 milhões no ano 2000. Informações do IBGE (2015) apontam, se forem consideradas as pessoas que utilizam a internet através de diversos equipamentos (computador, telefone móvel celular, *tablet* e outros), estimam que haviam no ano de 2013, 85,6 milhões de pessoas com acesso à internet, e 78,3 milhões de pessoas, se forem consideradas somente as pessoas que possuem acesso à internet através de computadores.

Com o aumento de pessoas com acesso à Internet, começaram a surgir também as primeiras empresas de **e-commerce** no Brasil, como o site Submarino que surgiu no ano de 1999. E esse tipo de comércio vem evoluindo cada vez mais no país. O Brasil é o país que lidera a participação de compras no e-commerce entre os países latinos, tendo 59,1% de participação, seguido por México (14,2%), Caribe (6,4%) e nossa vizinha Argentina (6,2%) (PIROLA, 2016).

Informações do relatório Webshoppers (Ebit 2016b) indicam que o valor médio gasto pelo brasileiro através de *e-commerce* foi de R\$ 403,46 no primeiro semestre de 2016, representando assim o aumento nominal de 7% em comparação com o primeiro semestre de 2015. Verificou-se ainda que 23,1 milhões de consumidores virtuais realizaram ao menos uma compra no primeiro semestre de 2016, representando um volume 31% maior que em 2015, fortalecido pela migração do consumidor do varejo físico para as compras *online*.

Ainda de acordo com o relatório do Webshoppers (Ebit 2016b), um dos principais motivos para a migração do varejo físico para o *e-commerce*, é a possibilidade de o consumidor comparar produtos e preços mais facilmente antes de efetuar a compra: Pesquisa especial da Ebit registrou-se que as pessoas demoram em média 16 dias para concretizar a compra, por exemplo, de um *smartphone*.

Esses dados mostram como o consumidor está buscando cada vez mais o *e-commerce*, fazendo com que essas empresas busquem serviços logísticos de qualidade para que possam suprir suas necessidades e entregar seus produtos aos clientes com rapidez e alto nível de confiabilidade. E, assim, o consumidor possa

realizar suas compras sem precisar se preocupar com atrasos na entrega da mercadoria.

3 MATERIAIS E MÉTODO

Este capítulo apresenta os materiais e o método utilizados para a realização deste trabalho.

3.1 MATERIAIS

O Quadro 1 apresenta as ferramentas e as tecnologias que foram utilizadas na modelagem e desenvolvimento da aplicação.

Ferramenta / Tecnologia	Versão	Finalidade
Astah Community	7.1.0	Modelagem Unified Modeling Language (UML) de casos de uso e diagrama de classes.
DBDesigner	4.0.5.6	Modelagem do banco de dados
PostgreSQL	9.5.1	Sistema Gerenciador do Banco de dados (SGBD)
Eclipse IDE	4.6	Integrated Development Environment (IDE) de desenvolvimento da aplicação Web
Spring Framework	1.5.10	Framework de desenvolvimento para plataforma Java
Java EE	7	Linguagem de programação.

Quadro 1 - Lista de Ferramentas e Tecnologias

3.1.1 Astah Community

Astah Community é uma ferramenta gratuita voltada para a modelagem de diagramas *Unified Modeling Language*. A ferramenta é conhecida por sua praticidade e simplicidade em elaborar diagramas, como por exemplo: diagramas de classe, casos de uso, sequência, atividade, comunicação, máquina de estado, componentes, implantação, estrutura de composição, objetos e pacotes (BRONDANI et al., 2016)

Além da versão Community, existem outras três versões da ferramenta: Astah UML, Astah Professional e Astah Share, que possuem outras funcionalidades específicas além da modelagem UML, porém, a licença para estas versões é comercial.

3.1.2 DBDesigner

O DBDesigner é uma ferramenta *Computer-Aided Software Engineering* (CASE) desenvolvida pela empresa *Fabulous Force Database Tools*. É uma ferramenta de uso livre e utilizada para a modelagem de dados visual.

O DBDesigner é uma excelente opção para soluções que utilizam quaisquer ferramentas de banco de dados, principalmente para MySQL, Oracle, *Microsoft SQL Server* e PostgreSQL. Por meio dessa ferramenta é possível modelar tabelas de forma gráfica, relacionamentos e ter a geração da modelagem, do projeto, da implementação e da manutenção integradas em apenas um ambiente (FABFORCE, 2016).

3.1.3 PostgreSQL

PostgreSQL é um sistema de banco de dados *open source* e objeto-relacional. A ferramenta é executada e compatível com todos os principais sistemas operacionais, incluindo Linux, UNIX e Windows. É altamente escalável tanto na enorme quantidade de dados que podem gerenciar e no número de usuários simultâneos que podem acomodar (POSTGRES, 2016).

3.1.4 Eclipse IDE

Segundo Faria et al (2010), o Eclipse é bastante utilizado, por ser de fácil uso e também por ser um software livre. Sua tecnologia é baseada em *plug-ins* e fornece um grande suporte aos programadores para realizar diferentes projetos de formas diversas.

A plataforma do Eclipse fornece vários pacotes de desenvolvimento, tais como:

a) Eclipse *Java Development Tool* (JDT), que é a base para qualquer *plug-in* na linguagem Java;

b) Eclipse Software Development Kit (SDK), que é o pacote de distribuição da *Integrated Development Environment* (IDE) Java;

c) Eclipse *Web Tools Platform* (WTP), que é usado para desenvolvimento de linguagem para web;

d) Compilador do JDT que é seu próprio compilador Java, que é mais rápido e de código aberto.

3.1.5 Spring Framework

O Spring é um *framework open source* para a plataforma Java não intrusivo, baseado nos padrões de projeto Inversão de Controle (IoC) e injeção de dependência.

No Spring o contêner se encarrega de "instanciar" classes de uma aplicação Java e definir as dependências entre elas utilizando um arquivo de configuração em formato XML, inferências do *framework*, o que é chamado *de auto-wiring* ou ainda anotações nas classes, métodos e propriedades. Dessa forma o Spring permite o baixo acoplamento entre classes de uma aplicação orientada a objetos.

Esse *framework* oferece diversos módulos que podem ser utilizados de acordo com as necessidades do projeto, como módulos voltados para desenvolvimento *Web*, persistência, acesso remoto e programação orientada a objetos.

3.1.6 Java EE

Java é uma linguagem de programação orientada a objetos. A plataforma Java Enterprise Edition (Java EE) é o padrão em software corporativo baseado na comunidade. O Java EE é desenvolvido usando o Java Community Process, com contribuições de especialistas do setor, organizações comerciais e de código aberto, grupos de usuários Java e inúmeras pessoas. Cada versão integra novos recursos que se alinham às necessidades do setor, melhora a portabilidade do aplicativo e aumenta a produtividade do desenvolvedor (ORACLE. 2018).

3.2 MÉTODO

A modelagem e a implementação do sistema têm como base o modelo sequencial linear descrito por Pressman (2011). O modelo sequencial foi utilizado para denominar os processos (ou fases) que serão executados durante o projeto. Estes processos foram organizados em ciclos iterativos de modelagem e implementação.

O Quadro 2 apresenta os processos (fluxos de trabalho) e as iterações que serão realizadas.

Iterações \ Processos	1ª Iteração	2ª Iteração	3ª Iteração
Requisitos	Definição dos requisitos básicos.	Complemento dos requisitos.	Revisão dos requisitos.
Análise	Definição do diagrama de casos de uso. Descrição das operações de cadastro de casos de uso.	Ajustes nos casos de uso e nas operações de cadastro.	Revisão dos casos de uso e das funcionalidades.
Projeto	Definição do diagrama de entidades e relacionamentos.	Ajuste de campos e atributos de tabelas. Descrição dos campos das tabelas.	Revisão da modelagem.
Implementação	Estudo das tecnologias e testes de implementação visando identificar a melhor forma de fazê-lo.	Implementação	Implementação da solução para dispositivos móveis do sistema.
Testes	Não haverá testes nesta iteração.	Realização dos testes. .	Realização dos testes de consulta de entregas através das soluções.

Quadro 2 - Iterações de desenvolvimento

A seguir estão descritas as etapas (identificadas como processos no Quadro 2) definidas para o desenvolvimento das soluções e as principais atividades de cada uma dessas etapas.

a) Requisitos

O levantamento dos requisitos iniciou tendo como objetivo desenvolver um software que possibilite à transportadora realizar o rastreamento de suas entregas, bem como controlar os gastos e os lucros com suas viagens. Na fase inicial foi realizada uma breve pesquisa sobre o funcionamento de uma transportadora e suas necessidades para o rastreamento das encomendas. Foram observadas também as funcionalidades ofertadas por sistemas já existentes na área, verificando assim os objetivos principais do software para empresas deste ramo.

Os requisitos foram complementados à medida que as iterações ocorriam. Várias alterações foram realizadas com o objetivo de definir os requisitos do sistema de forma funcional e adequada para o uso da transportadora.

b) Análise do sistema

Com base nos requisitos estabelecidos foram definidos os casos de uso. Esses casos de uso foram documentados gerando informações para a definição do banco de dados. Estes casos de uso foram modificados conforme as necessidades surgiam.

c) Projeto

Com base nos casos de uso foi definido o diagrama de entidades e relacionamentos do banco de dados, com a elaboração das tabelas e dos seus campos, sendo os tipos e tamanho de dados definidos. Esse diagrama foi alterado várias vezes, à medida que ajustes e complementos nos requisitos eram realizados.

d) Implementação

A implementação foi realizada utilizando a ferramenta Eclipse. Foi desenvolvida uma estrutura padrão para as telas do sistema. Inicialmente foi desenvolvida a solução web com as principais funcionalidades do sistema, e posteriormente a solução para dispositivos móveis. Após isso, foram desenvolvidos os relatórios.

e) Testes

Os testes foram informais e realizados à medida que as iterações forem ocorrendo. Esses testes foram baseados na verificação do código e na forma de interação com as soluções desenvolvidas.

4 RESULTADO

Este capítulo apresenta o que foi obtido como resultado deste trabalho que é a modelagem e a implementação de um sistema para gerenciamento de entregas de encomendas por transportadoras.

4.1 ESCOPO DO SISTEMA

O sistema se destina ao controle de entregas das encomendas realizadas por meio de comércio eletrônico. O objetivo do sistema é possibilitar o rastreamento das encomendas até a sua chegada ao consumidor final. O sistema possuirá duas soluções: uma *web* que será direcionada para uso dos funcionários da transportadora e uma para dispositivos móveis, que possibilitará ao cliente consultar o histórico da encomenda e, também, possibilitará ao motorista registrar os dados das viagens.

A solução web possuirá os principais cadastros, como entidades (clientes e fornecedores), funcionários, veículos, centros de distribuição, viagens e encomendas, e também será possível analisar os dados repassados pela outra solução. Por sua vez, na solução para dispositivos móveis o motorista poderá realizar, quando houver conexão com o servidor do sistema (internet), os cadastros das despesas de viagens, históricos de localização, bem como as tentativas de entrega de cada encomenda.

4.2 MODELAGEM DO SISTEMA

O Quadro 3 apresenta a listagem dos requisitos funcionais identificados para o sistema.

Identificação	Nome	Responsável	Descrição
RF01	Manter entidades	Funcionário	O cadastro de entidades será composto por cidade, nome, CPF ou CNPJ, data de nascimento ou fundação, endereço, número, bairro, CEP, complemento e o tipo da entidade.
RF02	Cadastrar encomenda	Funcionário	O cadastro de encomenda possuirá, entidades de origem e destino, descrição, chave de rastreio, peso, valor, data de

			coleta, previsão de entrega, altura, largura, profundidade e se a encomenda é frágil.
RF03	Manter centros de distribuição	Funcionário	O cadastro do centro de distribuição possuirá um id, cidade e descrição.
RF04	Cadastrar viagem	Funcionário	O cadastro da viagem terá centros de distribuição de origem e destino, funcionário, veículo, data de saída, data de chegada e se é uma viagem de entrega ou viagem entre centros de distribuição.
RF05	Cadastrar histórico da viagem	Motorista	O histórico da viagem possuirá uma viagem, a data do histórico, uma descrição, uma localização escrita e as coordenadas recuperadas através do GPS.
RF06	Cadastrar despesas	Motorista	A despesa possuirá um tipo de despesa, uma viagem, descrição, data e valor.
RF07	Realizar entrega	Motorista	A entrega conterà uma encomenda, data de entrega, observações e se a entrega foi realizada.
RF08	Consultar encomenda	Cliente	O cliente poderá visualizar as viagens e tentativas de entrega de sua encomenda, através da chave de rastreo e CPF ou CNPJ da entidade de origem ou destino. Esta consulta poderá ser realizada através do navegador de internet ou através de aplicativo.
RF09	Emitir Relatórios	Funcionário	Exemplos de relatórios que serão emitidos: - Relatório de extrato de viagem; - Relatório de despesas; - Relatório de lucratividade.

Quadro 3 - Requisitos Funcionais

O diagrama de casos de uso apresentado na Figura 1 apresenta as funcionalidades essenciais do sistema que são realizadas pelos seus atores funcionário, motorista e cliente.

O usuário do tipo funcionário será o responsável pelos cadastros de entidades, encomendas, centros de distribuição, viagens e emissão de relatórios. O usuário do tipo motorista será o responsável pelos cadastros de históricos de viagens, despesas e entregas das encomendas. O cliente apenas poderá consultar a encomenda e visualizar as viagens e tentativas de entrega realizadas.

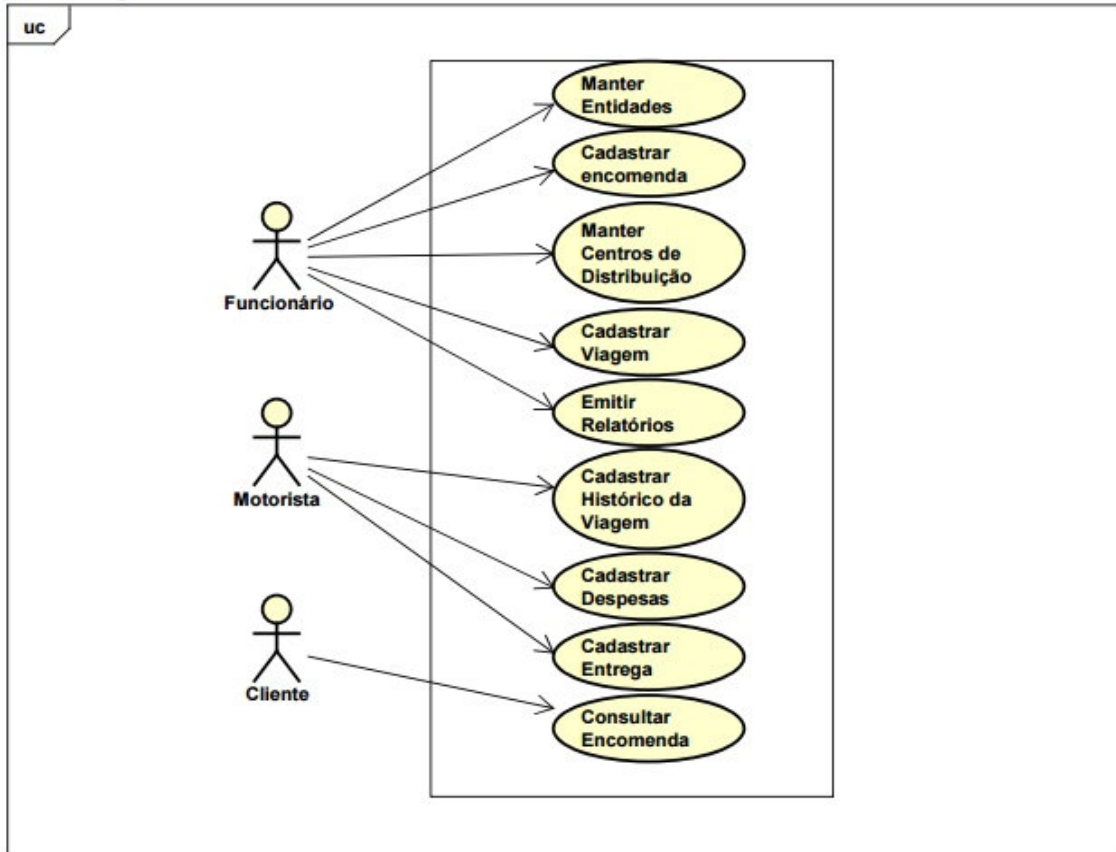


Figura 1 - Diagrama de casos de uso

O Quadro 4 apresenta a operação incluir dos casos de uso de cadastros para todas as funcionalidades constantes na Figura 1 que se referem aos cadastros realizados no sistema.

Caso de Uso:

Cadastrar entidades, encomendas, centros de distribuição, viagem, histórico da viagem, despesas e entregas.

Descrição:

Usuário irá efetuar um cadastro, incluindo os dados necessários a serem preenchidos em determinado cadastro.

Evento Iniciador:

Acessar o menu do cadastro que será inserido.

Atores:

Funcionário e Motorista

Pré-condição:

Ter acesso à solução web do sistema ou possuir o aplicativo instalado em seu dispositivo móvel.

Sequência de Eventos:

1. O usuário acessará a inclusão de um novo cadastro, pelo menu inicial escolhendo o cadastro de um item o qual deseja incluir.
2. Em seguida pressiona o botão incluir.
3. O usuário insere os dados necessários para determinado cadastro.
4. Em seguida pressiona o botão salvar.

Pós-condição:

Dados do cadastro inserido salvo no banco de dados.

Nome do fluxo alternativo (extensão)	Descrição
4 Dados não são válidos	4.1 No momento de incluir os dados, o sistema faz a verificação e constata que há dados inválidos, como CPF/CNPJ e falta de preenchimento de campos obrigatórios. É emitida uma mensagem e permanece no formulário de cadastro para edição.

Quadro 4 - Operação incluir dos casos de uso de cadastro

O Quadro 5 apresenta a operação de alteração dos casos de uso de cadastro.

Caso de Uso:

Alterar os cadastros de entidades, encomendas, centros de distribuição, viagem, histórico da viagem, despesas e entregas.

Descrição:

Usuário irá alterar um cadastro já incluso.

Evento Iniciador:

Acessar o cadastro que será alterado.

Atores:

Funcionário e Motorista.

Pré-condição:

Já possuir dados inclusos no cadastro a ser alterado.

Sequência de Eventos:

1. O usuário poderá alterar um cadastro já incluso, pelo menu inicial selecionando o cadastro que deseja alterar.
2. Em seguida seleciona o registro que deseja alterar.
3. O formulário abrirá em modo de edição para o usuário realizar as alterações necessárias.
4. Em seguida pressiona o botão salvar.

Pós-condição:

Dados do cadastro alterado salvo no banco de dados.

Nome do fluxo alternativo (extensão)	Descrição
4 Dados não são válidos	4.1 No momento de incluir os dados, o sistema faz a verificação e constata que há dados inválidos, como CPF/CNPJ e falta de preenchimento de campos obrigatórios. É emitida uma mensagem e permanece no formulário de cadastro para edição.

Quadro 5 - Operação alterar dos casos de uso de cadastro

O Quadro 6 detalha a operação de exclusão dos casos de uso de cadastro.

Caso de Uso:

Excluir os cadastros de entidades, encomendas, centros de distribuição, viagem, histórico da viagem, despesas e entregas.

Descrição:

Usuário irá excluir um cadastro já incluso.

Evento Iniciador:

Acessar o cadastro que será excluído.

Atores:

Funcionário e Motorista.

Pré-condição:

Já possuir dados inclusos no cadastro a ser excluído.

Sequência de Eventos:

1. O usuário poderá excluir um cadastro já incluso, pelo menu inicial selecionado o cadastro que deseja alterar.
2. Em seguida seleciona o registro que deseja excluir.
3. O usuário pressiona o botão excluir.

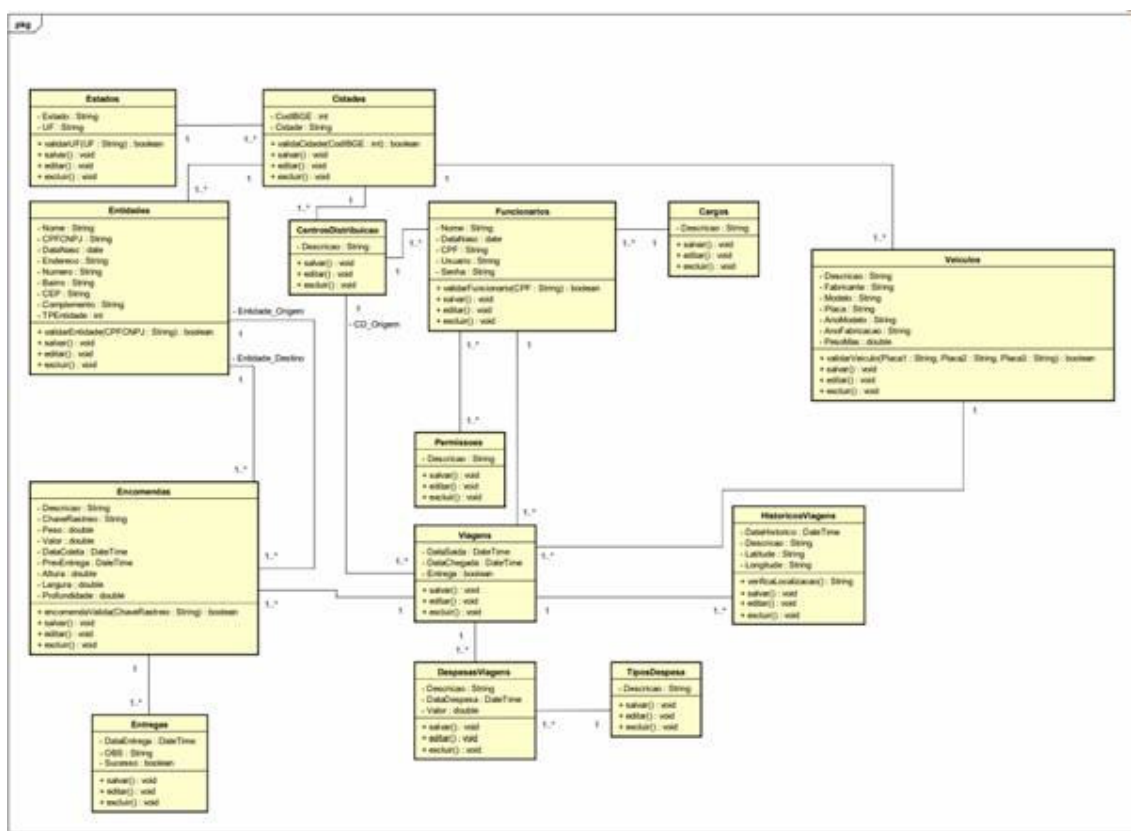
Pós-condição:

Dados do cadastro foram excluídos do banco de dados.

Nome do fluxo alternativo (extensão)	Descrição
3 Confirmação de exclusão	3.1 O sistema irá emitir uma mensagem confirmando se o usuário deseja realmente excluir determinado registro.
4 Não possuir dependências	4.1 O sistema irá emitir uma mensagem informando que o registro possui dependências e desta forma não poderá excluir o registro selecionado.

Quadro 6 - Operação excluir dos casos de uso de cadastro

A Figura 2 exibe o diagrama de classes que apresenta a forma que as classes do sistema foram implementadas.

**Figura 2 - Diagrama de classes**

A Figura 3 apresenta o diagrama de entidades e relacionamentos que representam a forma que o banco de dados está projetado.

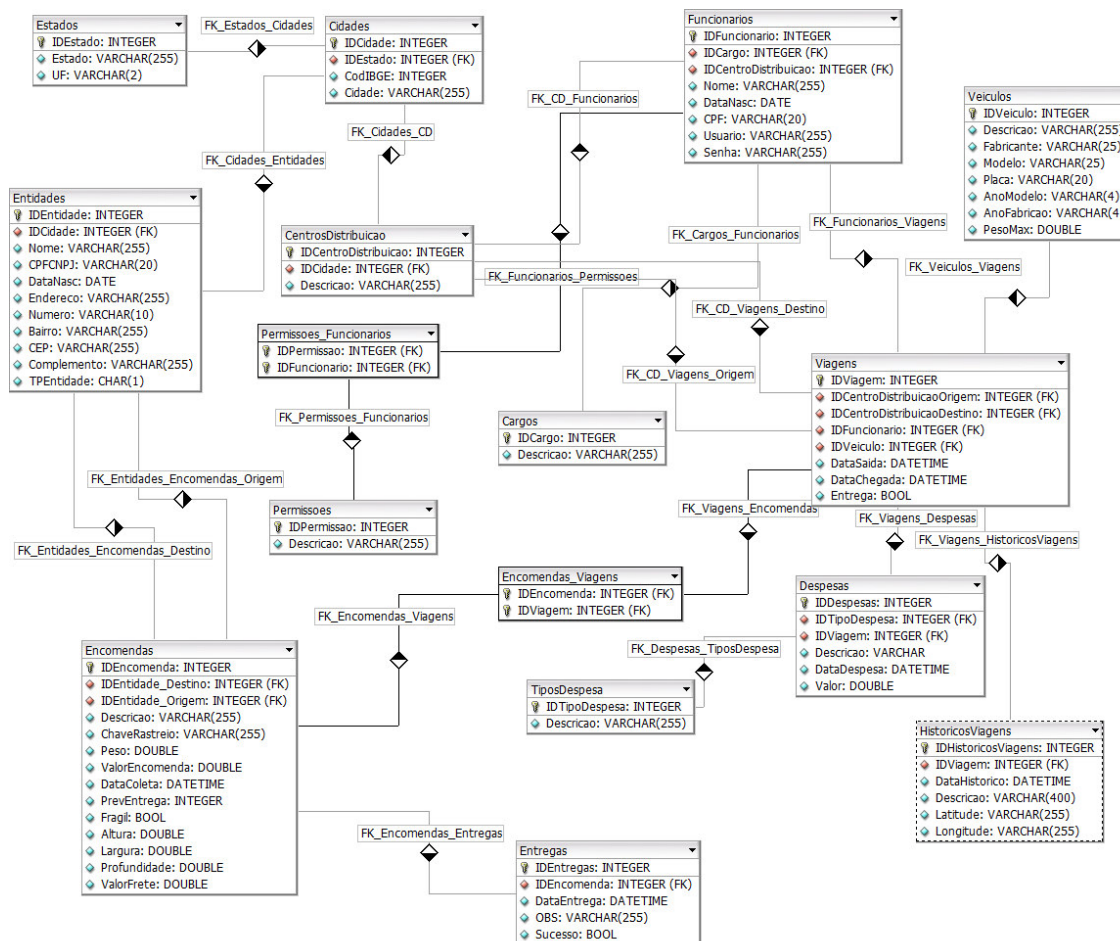


Figura 3 - Diagrama de Entidades Relacionamento

As chaves primárias e estrangeiras existentes em cada tabela são identificadas pelo prefixo “ID” no nome das respectivas colunas, visando assim facilitar a identificação e padronizar o desenvolvimento.

No Quadro 7 estão os campos da tabela Estados. Um estado poderá estar relacionado com muitas cidades. Os estados já estarão previamente inseridos no banco de dados.

Campo	Tipo	Nulo	Chave Primária	Chave estrangeira
Id	Integer	Não	Sim	Não
Estado	Varchar	Não	Não	Não
UF	Varchar	Não	Não	Não

Quadro 7 - Campos da tabela Estados

No Quadro 8 estão os campos da tabela Cidades.

Campo	Tipo	Nulo	Chave Primária	Chave estrangeira
Id	Integer	Não	Sim	Não
IDEstado	Integer	Não	Não	Sim
CodIBGE	Integer	Sim	Não	Não
Cidade	Varchar	Não	Não	Não

Quadro 8 - Campos da tabela Cidades

No Quadro 9 estão os campos da tabela Entidades.

Campo	Tipo	Nulo	Chave Primária	Chave estrangeira
Id	Integer	Não	Sim	Não
IDCidade	Integer	Não	Não	Sim
Nome	Varchar	Não	Não	Não
CPF CNPJ	Integer	Não	Não	Não
DataNasc	Date	Não	Não	Não
Endereco	Varchar	Não	Não	Não
Numero	Varchar	Não	Não	Não
Bairro	Varchar	Não	Não	Não
Cep	Varchar	Não	Não	Não

Quadro 9 - Campos da tabela Entidades

No Quadro 10 estão os campos da tabela CentrosDistribuicao.

Campo	Tipo	Nulo	Chave Primária	Chave estrangeira
Id	Integer	Não	Sim	Não
IDCidade	Integer	Não	Não	Sim
Descricao	Varchar	Não	Não	Não

Quadro 10 - Campos da tabela CentrosDistribuicao

No Quadro 11 estão os campos da tabela Cargos.

Campo	Tipo	Nulo	Chave Primária	Chave estrangeira
Id	Integer	Não	Sim	Não
Descricao	Varchar	Não	Não	Não

Quadro 11 - Campos da tabela Cargos

No Quadro 12 estão os campos da tabela Funcionarios.

Campo	Tipo	Nulo	Chave Primária	Chave estrangeira
Id	Integer	Não	Sim	Não
IDCargo	Integer	Não	Não	Sim
IDCentroDistribuicao	Integer	Não	Não	Sim
Nome	Varchar	Não	Não	Não
DataNasc	Date	Não	Não	Não
CPF	Varchar	Não	Não	Não
Usuario	Varchar	Não	Não	Não
Senha	Varchar	Não	Não	Não

Quadro 12 - Campos da tabela Funcionarios

No Quadro 13 estão os campos da tabela Permissoes_Funcionarios. Esta tabela será a responsável pelo relacionamento “muitos para muitos” entre as tabelas funcionários e permissões.

Campo	Tipo	Nulo	Chave Primária	Chave estrangeira
Id	Integer	Não	Sim	Não
IDPermissao	Integer	Não	Não	Sim
IDFuncionario	Integer	Não	Não	Sim

Quadro 13 - Campos da tabela Permissoes_Funcionarios

No Quadro 14 estão os campos da tabela Permissoes. Nesta tabela serão armazenadas as permissões de acesso do usuário.

Campo	Tipo	Nulo	Chave Primária	Chave estrangeira
Id	Integer	Não	Sim	Não
Descricao	Varchar	Não	Não	Não

Quadro 14 - Campos da tabela Permissoes

No Quadro 15 estão os campos da tabela Veiculos.

Campo	Tipo	Nulo	Chave Primária	Chave estrangeira
Id	Integer	Não	Sim	Não
Descricao	Varchar	Não	Não	Não
Fabricante	Varchar	Não	Não	Não
Modelo	Varchar	Não	Não	Não

Placa	Varchar	Não	Não	Não
AnoModelo	Varchar	Sim	Não	Não
AnoFabricacao	Varchar	Sim	Não	Não
PesoMax	Double	Sim	Não	Não

Quadro 15 - Campos da tabela Veiculos

No Quadro 16 estão os campos da tabela Viagens.

Campo	Tipo	Nulo	Chave Primária	Chave estrangeira
ID	Integer	Não	Sim	Não
ID_cddOrigem	Integer	Sim	Não	Sim
ID_cddDestino	Integer	Sim	Não	Sim
ID_funcionario	Integer	Não	Não	Sim
ID_veiculo	Integer	Não	Não	Sim
DataSaida	DateTime	Sim	Não	Não
DataChegada	DateTime	Sim	Não	Não
Entrega	Boolean	Não	Não	Não

Quadro 16 - Campos da tabela Viagens

No Quadro 17 estão os campos da tabela HistoricosViagens.

Campo	Tipo	Nulo	Chave Primária	Chave estrangeira
Id	Integer	Não	Sim	Não
IDViagem	Integer	Não	Não	Sim
DataHistorico	DateTime	Não	Não	Não
Descricao	Varchar	Não	Não	Não
Latitude	Varchar	Não	Não	Não
Longitude	Varchar	Sim	Não	Não

Quadro 17 - Campos da tabela HistoricosViagens

No Quadro 18 estão os campos da tabela TiposDespesa.

Campo	Tipo	Nulo	Chave Primária	Chave estrangeira
Id	Integer	Não	Sim	Não
Descricao	Varchar	Não	Não	Não

Quadro 18 - Campos da tabela TiposDespesa

No Quadro 19 estão os campos da tabela Despesas.

Campo	Tipo	Nulo	Chave Primária	Chave estrangeira
Id	Integer	Não	Sim	Não
ID_viagem	Integer	Não	Não	Sim
ID_tipoDespesa	Integer	Não	Não	Sim
Descricao	Varchar	Não	Não	Não
DataDespesa	DateTime	Não	Não	Não
Valor	Double	Não	Não	Não

Quadro 19 - Campos da tabela Despesas

No Quadro 20 estão os campos da tabela Encomendas.

Campo	Tipo	Nulo	Chave Primária	Chave estrangeira
Id	Integer	Não	Sim	Não
IDEntidade_Destino	Integer	Não	Não	Sim
IDEntidade_Origem	Integer	Não	Não	Sim
Descricao	Varchar	Não	Não	Não
ChaveRastreio	Varchar	Não	Não	Não
Peso	Double	Não	Não	Não
ValorEncomenda	Double	Não	Não	Não
DataColeta	DateTime	Não	Não	Não
PrevEntrega	Integer	Não	Não	Não
Altura	Double	Não	Não	Não
Largura	Double	Não	Não	Não
Profundidade	Double	Não	Não	Não
Valor	Double	Não	Não	Não

Quadro 20 - Campos da tabela Encomendas

No Quadro 21 estão os campos da tabela Encomendas_Viagens. Esta tabela será a responsável pelo relacionamento “muitos para muitos” entre as tabelas encomendas e viagens.

Campo	Tipo	Nulo	Chave Primária	Chave estrangeira
Id	Integer	Não	Sim	Não
IDEncomenda	Integer	Não	Sim	Sim
IDViagem	Integer	Não	Sim	Sim

Quadro 21 - Campos da tabela Encomendas_Viagens

No Quadro 22 estão os campos da tabela Entregas.

Campo	Tipo	Nulo	Chave Primária	Chave estrangeira
Id	Integer	Não	Sim	Não
IDEncomenda	Integer	Não	Não	Sim
DataEntrega	DateTime	Não	Não	Não
Obs	Varchar	Não	Não	Não
Sucesso	Boolean	Não	Não	Não

Quadro 22 - Campos da tabela Entregas

4.3 APRESENTAÇÃO DO SISTEMA

Ao acessar o sistema o usuário é direcionado para a tela principal que exibirá um menu de interação, apresentado na Figura 4. O menu principal é responsivo, ou seja, adequa-se à tela do dispositivo que está sendo utilizado, podendo ser acessado tanto por computadores como *tablet ou smartphone*.

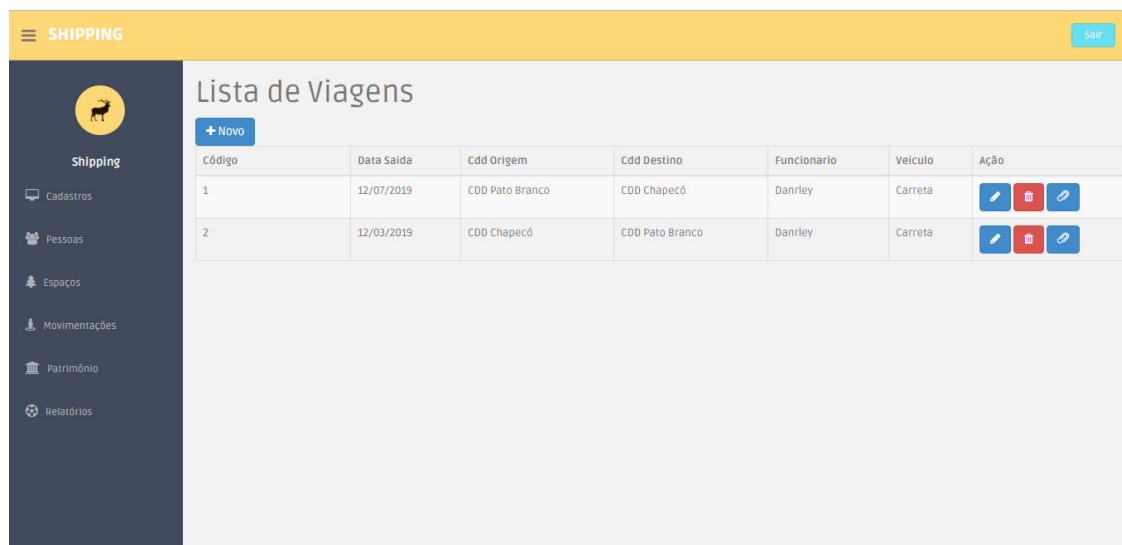


Figura 4 - Tela principal do sistema

No menu de cadastros estão disponíveis todas as rotinas de cadastros básicos do sistema, como cidades, estados e tipos de despesas. Logo abaixo há o menu Pessoas, que terá cadastros de entidades, funcionários e cargos. No menu Espaços há o cadastro de centros de distribuições da transportadora. No menu Movimentações estão as principais telas do sistema, como os cadastros de viagens, encomendas, entregas, entre outras. Logo abaixo, no menu Patrimônio, há a possibilidade de cadastro de veículos da transportadora. E por último há o menu

Relatórios, onde o usuário poderá visualizar os dados das movimentações do sistema.

Na Figura 5 é apresentada como exemplo a tela de lista de viagens. Nessa listagem o usuário poderá editar ou excluir uma viagem, ou ainda selecioná-la para anexar encomendas à mesma.

Código	Data Saída	Cdd Origem	Cdd Destino	Funcionario	Veiculo	Ação
1	26/11/2018	CDD Chapecó	CDD Campinas	Danrley	Carreta	  
2	26/11/2018	CDD Chapecó	CDD Campinas	Danrley	Carreta	  

Figura 5 - Lista de viagens

Caso seja selecionada a opção para anexar encomendas, o usuário será redirecionado para a tela apresentada na Figura 6. Nessa tela será apresentado ao usuário os dados principais da viagem selecionada e logo abaixo uma lista de encomendas para serem vinculadas.




Código	Descrição	Chave de Rastreio	Anexar
1	Caixa Pequena	123456	 
2	Caixa Média	654321	 
3	Caixa Grande	123	 

Figura 6 - Vincular encomendas à viagem

Ao clicar para vincular a encomenda à viagem, é exibida ao usuário uma mensagem de confirmação, conforme é apresentado na Figura 7. Caso o usuário realize a confirmação a encomenda será vinculada à viagem, ou o processo será cancelado caso o usuário desista da operação em clique em “Cancelar”.

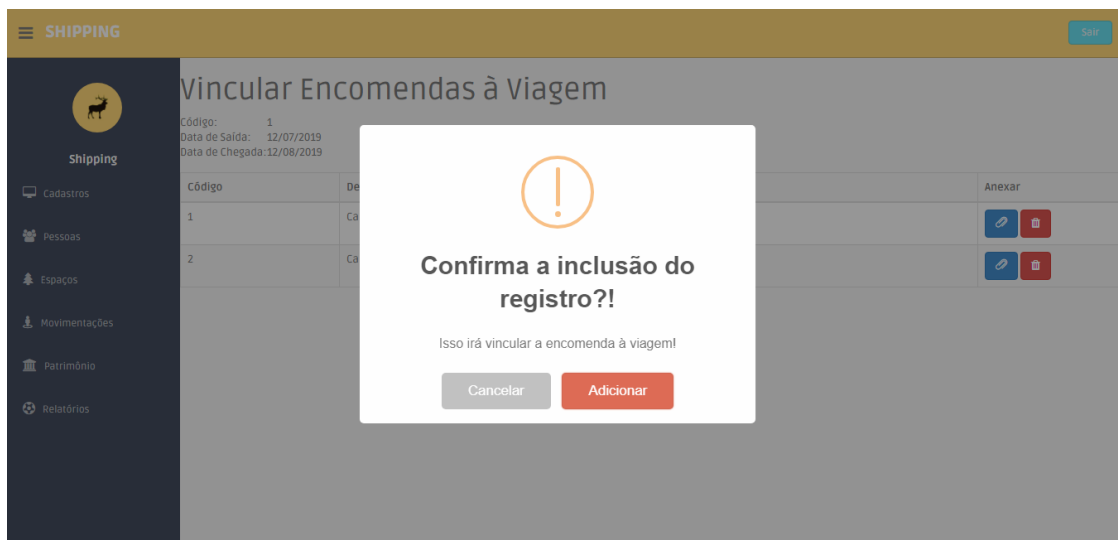


Figura 7 - Mensagem de confirmação

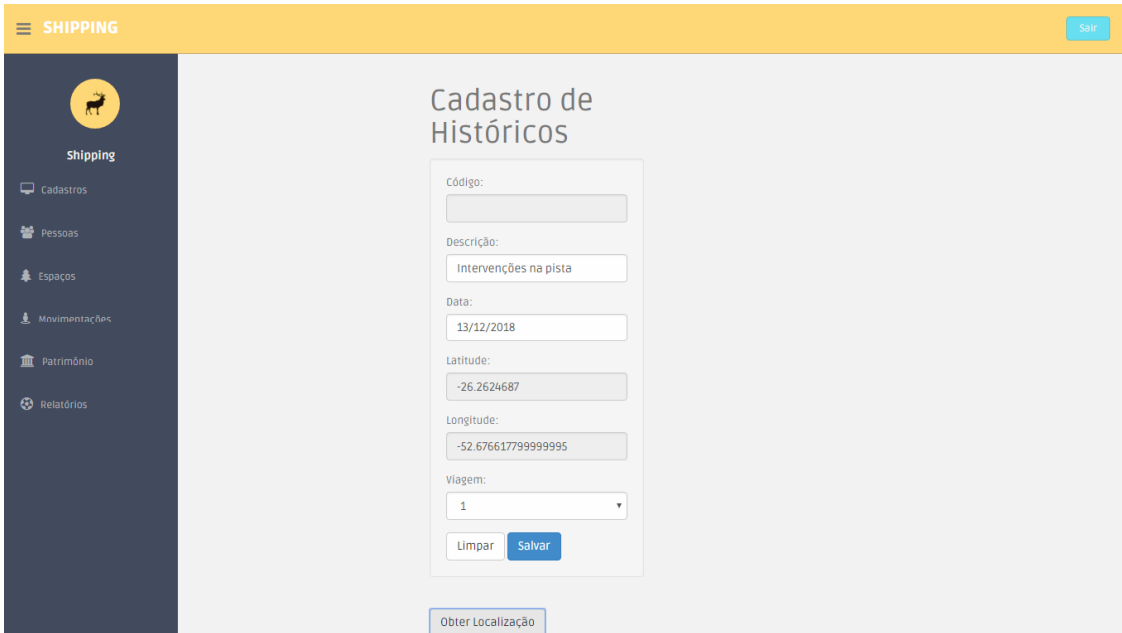
Além das funcionalidades destinadas ao usuário do sistema, tem ainda a tela de rastreamento de encomendas onde através da chave de rastreio o consumidor que solicitou o transporte da encomenda poderá verificar todas as viagens e tentativas de entrega realizadas pela transportadora. Esta tela com os dados do rastreamento é apresentada na Figura 8.



Figura 8 - Rastreamento da encomenda

Na tela de rastreamento são apresentadas as informações principais da encomenda, como remetente e destinatário, e logo abaixo o histórico de viagens e tentativas de entrega realizadas.

Na Figura 9 ainda é apresentada a tela de cadastro de históricos de viagens. Nesta tela o motorista poderá realizar o cadastro de um histórico da viagem, podendo relatar possíveis situações que podem ocasionar um atraso, como trânsito lento ou acidente na pista, por exemplo. Através desta tela o motorista poderá ainda informar o local exato que ele se encontra no momento deste cadastro, apenas clicando no botão “Obter localização”. Ao clicar neste botão o sistema irá obter a localização exata do dispositivo, utilizando recursos *javascript* e funções específicas do navegador.



The screenshot shows a web application interface for 'SHIPPING'. On the left is a dark sidebar with a 'Shipping' logo and a menu containing 'Cadastrados', 'Pessoas', 'Espaços', 'Movimentações', 'Patrimônio', and 'Relatórios'. The main content area is titled 'Cadastro de Históricos' and contains a form with the following fields: 'Código:' (empty), 'Descrição:' (containing 'Intervenções na pista'), 'Data:' (containing '13/12/2018'), 'Latitude:' (containing '-26.2624687'), 'Longitude:' (containing '-52.67661779999995'), and 'Viagem:' (a dropdown menu with '1' selected). Below the form are 'Limpar' and 'Salvar' buttons. At the bottom of the form area is a separate 'Obter Localização' button.

Figura 9 - Obtendo a localização do motorista

Todas as telas de listagens e de cadastros do sistema seguem o mesmo padrão de visualização e exibirão os principais dados de cada cadastro. O usuário poderá editar um registro utilizando do botão de ação com um ícone similar a um lápis, ou realizar a exclusão por meio do botão de ação com um ícone similar a uma lixeira.

Após a realização dos cadastros e movimentações das encomendas, o usuário poderá ainda obter relatórios dos principais cadastros realizados, bem como suas movimentações de despesas e receitas com as encomendas. Na Figura 10 é apresentado como exemplo o relatório de viagens por encomenda, no qual o usuário poderá ter o controle de quais encomendas pertencem a uma determinada viagem.

Relatório de Encomendas por Viagem	
Descrição Encomenda	Valor
Viagem: 1 Data Saída: 12/07/2019	
CDD Origem: CDD Pato Branco	
Caixa Média	60.00
Faturamento total viagem:	60.00
Viagem: 2 Data Saída: 12/03/2019	
CDD Origem: CDD Chapecó	
Caixa Média	60.00
Caixa Pequena	30.00
Faturamento total viagem:	90.00
	Página: 1 de 1

Figura 10 - Relatório de encomendas por viagem

Na Figura 11 é apresentado o relatório de despesas por viagem. Neste relatório o usuário poderá ter um maior controle de todas as despesas que cada viagem gerou, e assim comparar com o relatório apresentado na Figura 10 e verificar se determinada gerou lucro ou prejuízos.

Despesas por viagem	
Descrição Despesa	Valor
Viagem: 1	
Veículo: Carreta	
Combustível	150.00
Borracharia	30.00
Despesas totais viagem:	180,00
Viagem: 2	
Veículo: Carreta	
Combustível	100.00
Borracharia	30.00
Mecânica	50.00
Despesas totais viagem:	180,00

Página: 1 de 1

Figura 11 - Relatório de despesas por viagem

4.4 IMPLEMENTAÇÃO DO SISTEMA

Para iniciar o desenvolvimento da aplicação foi definida a estrutura de pacotes para organizar o projeto. Esta estrutura é apresentada na Figura 12. Foi utilizada a estrutura *Model-View-Controller* (MVC). Nos pacotes *controller*, *model* e *repository* foram desenvolvidas as funções e métodos do *backend*, as classes do sistema, as conexões com o banco de dados e o mapeamento das páginas da aplicação.

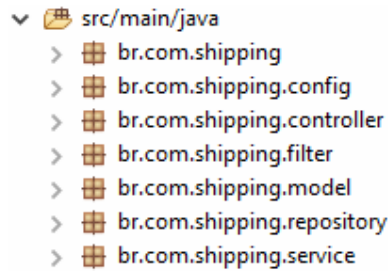


Figura 12 - Estrutura de pacotes

Nos pacotes *config*, *filter* e *service* estão definidas a parte de segurança do sistema, sendo configuradas quais páginas da aplicação poderão ser acessadas pelos respectivos tipos de usuários, por exemplo.

Na Figura 13, está a apresentação das pastas onde estão localizadas as páginas que contêm a parte gráfica (*View*) da aplicação.

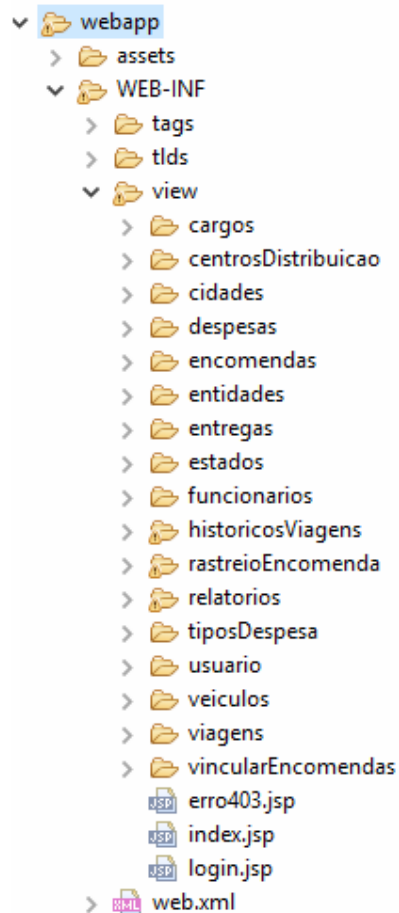


Figura 13 - Páginas da aplicação

Iniciando pela configuração da conexão, no arquivo responsável pela conexão foram definidos os parâmetros de conexão com o banco de dados, conforme podemos visualizar na Listagem 1.

```
spring.jpa.hibernate.ddl-auto:update
spring.jpa.show-sql:true
spring.jpa.hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect
spring.datasource.url=jdbc:postgresql://localhost:5432/shipping
spring.datasource.username=postgres
spring.datasource.password=123456
spring.datasource.driverClassName=org.postgresql.Driver
server.port=8085
```

Listagem 1 - Configurações da conexão com a base de dados

A Listagem 1 apresenta os padrões de conexão com o banco de dados PostgreSQL, sendo definido o diretório de conexão com a base de dados, os dados de autenticação, o *driver* utilizado e a porta na qual a aplicação será executada.

Ainda na parte estrutural da aplicação, a Listagem 2 todas as dependências que foram importadas para o correto funcionamento da aplicação. As principais são referentes ao Spring Framework, como a que é listada da linha número 42 à linha número 44, que é responsável pela segurança de acesso ao sistema pelo recurso Spring Security.

```
1<dependencies>
2    <dependency>
3        <groupId>org.springframework.boot</groupId>
4        <artifactId>spring-boot-starter-web</artifactId>
5    </dependency>
6    <dependency>
7        <groupId>org.springframework.boot</groupId>
8        <artifactId>spring-boot-starter-tomcat</artifactId>
9    </dependency>
10   <dependency>
11       <groupId>org.apache.tomcat.embed</groupId>
12       <artifactId>tomcat-embed-jasper</artifactId>
13       <scope>provided</scope>
14   </dependency>
15   <dependency>
16       <groupId>javax.servlet</groupId>
17       <artifactId>jstl</artifactId>
18   </dependency>
19   <dependency>
20       <groupId>org.springframework.boot</groupId>
21       <artifactId>spring-boot-starter-data-jpa</artifactId>
22   </dependency>
23   <dependency>
24       <groupId>org.postgresql</groupId>
25       <artifactId>postgresql</artifactId>
26       <scope>provided</scope>
27   </dependency>
28   <dependency>
29       <groupId>org.springframework.boot</groupId>
30       <artifactId>spring-boot-devtools</artifactId>
```



```

31     </dependency>
32     <dependency>
33         <groupId>org.springframework.boot</groupId>
34         <artifactId>spring-boot-starter-data-rest</artifactId>
35     </dependency>
36     <dependency>
37         <groupId>org.projectlombok</groupId>
38         <artifactId>lombok</artifactId>
39         <scope>provided</scope>
40     </dependency>
41     <dependency>
42         <groupId>org.springframework.boot</groupId>
43         <artifactId>spring-boot-starter-security</artifactId>
44     </dependency>
45     <dependency>
46         <groupId>org.apache.commons</groupId>
47         <artifactId>commons-lang3</artifactId>
48         <version>3.3.2</version>
49     </dependency>
50     <dependency>
51         <groupId>commons-fileupload</groupId>
52         <artifactId>commons-fileupload</artifactId>
53         <version>1.3.1</version>
54     </dependency>
55     <dependency>
56         <groupId>commons-io</groupId>
57         <artifactId>commons-io</artifactId>
58         <version>2.4</version>
59     </dependency>
60     <dependency>
61         <groupId>org.json</groupId>
62         <artifactId>json</artifactId>
63     </dependency>
64     <dependency>
65         <groupId>net.sf.jasperreports</groupId>
66         <artifactId>jasperreports</artifactId>
67         <version>6.2.0</version>
68     </dependency>
69 </dependencies>

```

Listagem 2 – Dependências importadas pela aplicação

Para exemplificar a criação do *front* é apresentada a estrutura da página de vínculo de encomendas às viagens, de maneira detalhada. Na Listagem 3 é apresentada a primeira função *Java Script* criada nesta tela, que realiza o vínculo do registro de encomenda à viagem.

```

function adicionar(idViagem, idEncomenda) {
    swal({
        title: "Confirma a inclusão do registro?!",
        text: "Isso irá vincular a encomenda à viagem!",
        type: "warning",
        showCancelButton: true,
        confirmButtonColor: "#DD6B55",
        cancelButtonText: "Cancelar",
        confirmButtonText: "Adicionar",
        closeOnConfirm: false
    },
    function() {

```

```

var destino = '<c:url value="/vincularEncomendas/adicionar/">' + idViagem
+ '/' + idEncomenda;
$.ajax({
  type : 'GET',
  url : destino,
  success : function(data) {
    if (data.situacao == "OK"){
      swal("Vinculado!", "Registro vinculado com sucesso.", "success");
    }else{
      swal("Erro!", "Falha ao vincular registro.", "error");
    }
  },//Fim success
  error : function() {
    swal("Erro!", "Falha ao vincular registro.", "error");
  }
}); //Fim ajax
}); //Fim swal
}
</script>

```

Listagem 3 – Função que vincula a encomenda à viagem selecionada

A função apresentada neste código recebe como parâmetro o identificador do registro da viagem e da encomenda selecionada e solicita ao usuário uma confirmação da ação realizada. Caso seja confirmado pelo usuário, é repassado através da variável *destino* a página de redirecionamento que o sistema deverá acessar para realizar tal ação, concatenada com o código de identificação dos registros da viagem e da encomenda. Essa página acessada será mapeada pelo *backend* através da classe *EncomendaViagensController*, cujo o funcionamento é apresentado, após a apresentação do *front*.

Na Listagem 4 está o código da função que desvinculará a encomenda da viagem. O funcionamento é semelhante ao da função apresentada na Listagem 3, apenas é repassado um diretório diferente para a variável *destino*, que irá redirecionar para o método respectivo no *backend*.

```

<script type="text/javascript">
  function remove(idViagem,idEncomenda) {
    swal({
      title: "Confirma a exclusão do registro?!",
      text: "Isso irá desvincular a encomenda da viagem!",
      type: "warning",
      showCancelButton: true,
      confirmButtonColor: "#DD6B55",
      cancelButtonText: "Cancelar",
      confirmButtonText: "Remover",
      closeOnConfirm: false
    },
    function(){
      var destino = '<c:url
value="/vincularEncomendas/remove/">' + idViagem + '/' + idEncomenda;
$.ajax({
  type : 'GET',

```

```

url : destino,
success : function(data) {
if (data.situacao == "OK"){
swal("Desvinculado!", "Registro desvinculado com sucesso.", "success");
}else{
swal("Erro!", "Falha ao remover registro.", "error");
}
}, //Fim success
error : function() {
swal("Erro!", "Falha ao remover registro.", "error");
}
}); //Fim ajax
}); //Fim swal
}
</script>

```

Listagem 4 – Função que desvincula a encomenda à viagem selecionada

Na Listagem 5 é apresentada a continuação do desenvolvimento da página de vínculo de encomendas. A tag `<jsp:body>` indica o início do código dos elementos visuais do corpo da página e logo abaixo é exibido o código para criação de uma tabela na qual serão apresentados os dados da viagem selecionada.

```

<jsp:body>
<section class="wrapper">
<div class="row">
<h1>Vincular Encomendas à Viagem</h1>
<div class="spacer"></div>
<div class="form-group">
<table>
<tr>
<td>Código: </td>
<td>${viagens.id}</td>
</tr>
<tr>
<td>Data de Saída: </td>
<td><fmt:formatDate pattern = "dd/MM/yyyy" value =
"${viagens.dataSaida}" /></td>
</tr>
<tr>
<td>Data de Chegada: </td>
<td><fmt:formatDate pattern = "dd/MM/yyyy" value =
"${viagens.dataChegada}" /></td>
</tr>
</table></div>

```

Listagem 5 – Tabela com os dados da viagem selecionada

Na sequência é criada uma nova tabela, que será utilizada para exibir os dados das encomendas que serão listados e poderão ser vinculadas à viagem, conforme está sendo apresentado na Listagem 6. No atributo `<thead>` são definidos os dados do cabeçalho da tabela, indicando que tipo de conteúdo será apresentado pela respectiva coluna da tabela.

```

<div class="spacer"></div>
  <table id="dados" data-id-field="id" data-unique-id="id"
    class="table table-striped table-bordered display"
    style="width: 100%">
    <thead>
      <tr>
        <th class="col-md-2" >Código</th>
        <th>Descrição</th>
        <th>Chave de Rastreo</th>
        <th class="col-md-2">Anexar</th>
      </tr>
    </thead>

```

Listagem 6 – Criação da tabela de apresentação de encomendas

A Listagem 7 apresenta o código responsável pela apresentação das encomendas. Na função *forEach* o sistema realiza um *loop* entre os registros de encomendas existentes e recupera os dados dos campos sinalizados. A variável *var* indica o ponteiro do registro atual dentro da lista definida em *items*. Na última coluna do corpo da tabela há o código dos *links* que farão a chamada das funções *adicionar* e *remove* apresentadas nas Listagens 3 e 4, respectivamente. Após a exibição de todos os registros do *forEach*, há a *tag* de finalização *smo*, além da finalização também do corpo da tabela. Como esta tabela não possui rodapé, então já é também finalizada a tabela.

```

<tbody>
  <c:forEach var="encomenda" items="{encomendas}" >
    <tr id="row_{$encomenda.id}" >
      <td>{$encomenda.id}</td>
      <td>{$encomenda.descricao}</td>
      <td>{$encomenda.chaveRastreio}</td>
      <td>
        <a class="edit btn-primary" title="Anexar"
          href="javascript:adicionar({viagens.id},{$encomenda.id})">
          <i class="li_clip" aria-hidden="true"></i></a>
          <a class="remove btn-danger" title="Remover"
            href="javascript:remove({viagens.id},{$encomenda.id})">
            <i class="fa fa-trash" aria-hidden="true"></i></a>
        </td>
      </tr>
    </c:forEach>
</tbody>

```

Listagem 7 – Corpo da tabela com os dados das encomendas

Nas próximas listagens será apresentado o código fonte da classe *EncomendasViagensController*, que é responsável pelo *backend* da manutenção de vínculos das encomendas com as respectivas viagens. Na Listagem 8 estão as anotações *@Controller* e *@RequestMapping* que fazem parte do *Spring framework*,

sendo o primeiro responsável por receber requisições e enviar respostas ao usuário e o segundo pelo mapeamento das páginas de navegação. Há ainda, nessa listagem. A injeção de dependências de chaves estrangeiras utilizando a anotação *@Autowired* e a declaração da variável *contador*, que será utilizada e explicada nas próximas Listagens.

```
@Controller
@RequestMapping("/vincularEncomendas")
public class Encomendas_ViagensController {

    @Autowired
    private Encomendas_ViagensRepository encomendas_ViagensRepository;
    @Autowired
    private EncomendasRepository encomendasRepository;
    @Autowired
    private ViagensRepository viagensRepository;
    int contador;
```

Listagem 8 – Injeção de dependências e declaração de variáveis

Na Listagem 9 é apresentada a função responsável pela listagem dos dados da viagem selecionada, bem como das encomendas que ainda não possuem entrega realizada. Quando a página é acessada com apenas um parâmetro adicional, representando o código identificador da viagem, a anotação *@RequestMapping* executa a função *form*, que carrega do banco de dados os dados da viagem e de encomendas, filtrando-as pelo atributo *entregue*, retornando apenas aquelas com conteúdo falso, ou seja, que ainda não foram possuem registro de entrega com sucesso. Em seguida retorna uma *url* que irá direcionar para a página de vínculo das encomendas à viagem.

```
@RequestMapping(value = "/form/{id}", method = RequestMethod.GET)
public String form(@PathVariable Long id, Model model){
    model.addAttribute("encomendas",
        encomendasRepository.findByEntregue(false));
    model.addAttribute("viagens", viagensRepository.findOne(id));
    return "vincularEncomendas/form";
}
```

Listagem 9 – Método de listagem da viagem e encomendas

Após o usuário selecionar a opção de vincular uma encomenda, o servidor, através das anotações *@RequestMapping* e *@ResponseBody* recebe esta requisição e executa a função *adicionar*, conforme o código apresentado pela Listagem 10.

```

@RequestMapping(value = "/adicionar/{idViagem}/{idEncomenda}", method =
RequestMethod.GET,
    produces="application/json")
    @ResponseBody
    public String adicionar(@PathVariable Long idViagem,
        @PathVariable Long idEncomenda, @Valid Encomendas_Viagens
encomendas_viagens, BindingResult erros, Model model){
        JSONObject retorno = new JSONObject();
        try{
            if (erros.hasErrors()){
                retorno.put("situacao", "ERRO");
                retorno.put("mensagem", "Falha ao salvar registro!");
            }
        }
    }

```

Listagem 10 – Função que vincula as encomendas à viagem

Para o tratamento de possíveis erros que possam ocorrer ao salvar o registro no banco de dados, foi utilizado o método *TryCatch* nesta função. Após instanciar o objeto *JSON* e entrar no método citado, é realizada uma verificação de erros e se não houver nenhum erro o processo de salvamento continua, conforme apresentado pela Listagem 11.

```

else{
    contador = 0;
    Viagens viagem = viagensRepository.findOne(idViagem);
    Encomenda encomenda = encomendasRepository.findOne(idEncomenda);
    Encomendas_Viagens encomendasViagens = new Encomendas_Viagens();
    encomendasViagens.setViagem(viagem);
    encomendasViagens.setEncomenda(encomenda);
    for(Encomendas_Viagens ev : encomendas_ViagensRepository.findAll()){
        if((ev.getViagem() == viagem) && (ev.getEncomenda() == encomenda)){
            contador++;
        }
    }
}

```

Listagem 11 – Identificação dos dados selecionados

Ainda no método adicionar será inicializado o valor da variável contador, identificada a viagem e encomendas recebidas e então é iniciada uma estrutura de repetição dentre os registros já existentes, para verificar se a encomenda já possui registro de vínculo com determinada viagem. Se possuir, a variável *contador* é incrementada. A Listagem 12 apresenta a codificação na qual é testado o valor da variável *contador*.

```

if(contador > 0){
    retorno.put("situacao", "ERRO");
    retorno.put("mensagem", "Registro já foi vinculado!");
} else{
    encomendas_ViagensRepository.save(encomendasViagens);
    retorno.put("situacao", "OK");
    retorno.put("mensagem", "Registro salvo com sucesso!");
}
}

```

```

} catch (Exception ex){
    retorno.put("situacao", "ERRO");
    retorno.put("mensagem", "Falha ao salvar registro!");
}

return retorno.toString();
}

```

Listagem 12 – Teste para verificar se a encomenda já está vinculada à viagem

Caso o valor do contador seja maior que zero, o servidor retornará um erro para o cliente, indicando que a encomenda já possui vínculo com a viagem. Se o valor do contador não for maior que zero, então o registro é salvo e é retornada para o cliente a mensagem indicando o sucesso do vínculo. Após isso há, ainda, a codificação para o tratamento de erros ao salvar, caso ocorram.

Na Listagem 13 é apresentada a função *remover*. Esta função tem funcionamento muito semelhante à função *adicionar*. Primeiro é realizado o mapeamento do diretório, em seguida o tratamento de erros com a função *TryCatch*, identificação da encomenda e da viagem. Também é verificado se o registro que está em processo de exclusão possui o vínculo da encomenda com a viagem e, em caso positivo, o registro é excluído e é enviada resposta ao cliente de que a operação foi realizada com sucesso.

```

@RequestMapping(value = "/remover/{idViagem}/{idEncomenda}", method =
RequestMethod.GET,
                produces="application/json")
@ResponseBody
public String remover(@PathVariable Long idViagem, @PathVariable Long
idEncomenda,
                    @Valid Encomendas_Viagens encomendas_viagens, BindingResult erros,
Model model){

JSONObject retorno = new JSONObject();
try{
    if (erros.hasErrors()){
        retorno.put("situacao", "ERRO");
        retorno.put("mensagem", "Falha ao remover registro!");
    }else{
        Encomendas encomenda = encomendasRepository.findOne(idEncomenda);
        Viagens viagem = viagensRepository.findOne(idViagem);
        for(Encomendas_Viagens ev :
encomendas_ViagensRepository.findByEncomenda(encomenda)){
            if(ev.getViagem().getId() == viagem.getId()){
                if( ev.equals(null) ){
                    retorno.put("situacao", "ERRO");
                    retorno.put("mensagem", "Este registro não está vinculado!");
                }else{
                    encomendas_ViagensRepository.delete(ev);
                    retorno.put("situacao", "OK");
                    retorno.put("mensagem", "Registro excluído com sucesso!");
                }
            }
        }
    }
}
}

```

```
}  
    } catch (Exception ex){  
        retorno.put("situacao", "ERRO");  
        retorno.put("mensagem", "Falha ao remover registro!");  
    }  
    return retorno.toString();  
}
```

Listagem 13 – Método para desvincular encomenda da viagem

5 CONSIDERAÇÕES FINAIS

. Os cadastros básicos do sistema, as movimentações da encomenda até seu destinatário e os relatórios já foram implementados de acordo com as funcionalidades definidas no projeto. Também foram desenvolvidas as funcionalidade de obtenção da localização atual do dispositivo nos cadastros de históricos da viagem e a funcionalidade para rastreamento da encomenda pelo cliente da transportadora, sem que haja necessidade de efetuar o acesso ao sistema da transportadora.

Como não foi possível o desenvolvimento da aplicação para dispositivos móveis, foi então desenvolvida a aplicação principal de forma que a mesma funcione com fluidez e responsividade, de maneira que os menus e telas da aplicação sejam redimensionados automaticamente de acordo com o formato e tamanho da tela do dispositivo, o que possibilita assim o acesso à aplicação principal também por dispositivos móveis.

O resultado final obtido da aplicação foi satisfatório, pois permite o gerenciamento das viagens e encomendas de uma transportadora por completo, desde a sua postagem pelo remetente até o recebimento da mesma pelo destinatário, bem como o rastreamento da entrega das encomendas pelo cliente.

Entre as perspectivas futuras está a de continuidade no desenvolvimento e aprimoramento de todas as funcionalidades propostas, sempre melhorando e atualizando-as com as principais novidades existentes.

REFERÊNCIAS

Android Studio. **Configure compilações de forma ilimitada**. Disponível em: <<https://developer.android.com/studio/features.html>>. Acesso em: 12 out. 2016.

BOWERSOX, Donald J.; CLOSS, David J.; COOPER, M. Bixby; BOWERSOX, John C. **Gestão Logística da Cadeia de Suprimentos**. AMGH Editora, 2013.

BRONDANI, Camila Hubner; AREND, Cesar Frantz; SOUZA, Darcielle Aparecida Zilio de; PIRES, José Carlos Puiati. **Guia prático de utilização da ferramenta Astah Community 6.1**. Disponível em: <<http://pt.scribd.com/doc/139768773/AstahCommunity#>> Acesso em: 12 out. 2016.

CERIBELI, Harrison B.; OLIVEIRA, Raoni I.; FELIPE, Israel J. S. **Um estudo dos determinantes da decisão dos e-consumidores de comprarem no comércio eletrônico**. Revista Gestão & Tecnologia, Pedro Leopoldo, v. 15, n. 1, p. 174-199, Janeiro-Abril 2015. Disponível em: <<http://ssrn.com/abstract=2600228>>. Acesso em: 14 set. 2016.

Ebit. **2016a**. Relatório Webshoppers 2016, ed. 33. Disponível em: <<http://portal.ebit.com.br/Webshoppers>>. Acesso em: 12 out. 2016.

Ebit. **2016b**. Relatório Webshoppers 2016, ed. 34. Disponível em: <<http://portal.ebit.com.br/Webshoppers>>. Acesso em: 28 set. 2016.

FABFORCE. 2016. **General Information**. Disponível em: <<http://fabforce.eu/dbdesigner4/index.php>>. Acesso em: 28 set. 2016.

FARIA, Fernanda B.; LIMA, Priscila da S. N.; DIAS, Luiz G.; SILVA, Andrea A.; COSTA, Mayara P. da; BITTAR, Thiago J. **Evolução e Principais Características do IDE Eclipse**. Goiás: ENACOMP, 2010. Disponível em: <http://www.enacomp.com.br/2010/cd/artigos/completos/enacomp2010_23.pdf>. Acesso em: 12 out. 2016.

IBGE. **Acesso à internet e à televisão e posse de telefone móvel celular para uso pessoal : 2013**. Rio de Janeiro: IBGE, 2015. Disponível em: <<http://biblioteca.ibge.gov.br/index.php/biblioteca-catalogo?view=detalhes&id=293373>>. Acesso em: 06 out. 2016.

NOVAES, Antonio. **Logística e gerenciamento da cadeia de distribuição – 4ª Edição**. Elsevier Brasil, 2014.

Oracle. 2018. **Overview**. Disponível em: <<https://www.oracle.com/technetwork/java/javaee/overview/index.html>>. Acesso em: 06 dez. 2018.

PIROLA, Antonio L. R. **Comércio Eletrônico: Conhecimentos básicos para iniciantes**. Clube de autores, 2016.

PostgreSQL. 2016. **About**. Disponível em: <<https://www.postgresql.org/about/>>. Acesso em: 11 nov. 2016.

PRESSMAN, Roger S. **Engenharia de Software: Uma Abordagem Profissional**. Editora Bookman, 2011.

SPRING. 2018. **Overview**. Disponível em: <<https://spring.io/projects/spring-boot>>. Acesso em: 06 dez. 2018.