

**UNIVERSIDADE REGIONAL DE BLUMENAU**  
**CENTRO DE CIÊNCIAS EXATAS E NATURAIS**  
**CURSO DE SISTEMAS DE INFORMAÇÃO – BACHARELADO**

**SISTEMA PARA GESTÃO DE FROTAS DE VEÍCULOS**

**RODRIGO ALEXANDRINO PEREIRA**

**BLUMENAU**  
**2016**

**RODRIGO ALEXANDRINO PEREIRA**

## **SISTEMA PARA GESTÃO DE FROTAS DE VEÍCULOS**

Trabalho de Conclusão de Curso apresentado ao curso de graduação em Sistemas de Informação do Centro de Ciências Exatas e Naturais da Universidade Regional de Blumenau como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

Prof. Matheus Carvalho Viana, Doutor - Orientador

**BLUMENAU  
2016**

# **SISTEMA PARA GESTÃO DE FROTAS DE VEÍCULOS**

Por

**RODRIGO ALEXANDRINO PEREIRA**

Trabalho de Conclusão de Curso aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II pela banca examinadora formada por:

Presidente: \_\_\_\_\_  
Prof. Matheus Carvalho Viana, Doutor – Orientador, FURB

Membro: \_\_\_\_\_  
Prof. Francisco Adell Péricas, Mestre – FURB

Membro: \_\_\_\_\_  
Prof. Luciana Pereira de Araújo, Mestre – FURB

Blumenau, 04 de Julho de 2016

Dedico este trabalho aos meus pais, minha irmã e meus avôs, que sempre me apoiaram e proporcionaram uma boa educação.

## **AGRADECIMENTOS**

A Deus por todas as bênçãos recebidas.

À minha família por apoiar e incentivar minhas decisões.

Aos meus amigos William Ribeiro e Saionara Martins por me receberem em sua casa quando precisei. Josimar Cuchi e Alexsandro Pires por acreditarem no meu potencial e permitirem que hoje esteja em Blumenau.

Ao meu orientador Matheus Viana, por me oferecer todo suporte necessário para realização desse trabalho.

Há mais pessoas que desistem do que pessoas que fracassam.

Henry Ford

## RESUMO

O presente trabalho tem como objetivo apresentar o desenvolvimento de um sistema *web* que visa auxiliar as pessoas e empresas a fazerem a gestão dos seus veículos. O sistema possibilita ao usuário realizar o controle de sua frota a partir da Internet e fornece relatórios de: consumo médio, distância percorrida, viabilidade, despesas e impacto financeiro. Portanto, ele permite acompanhar as operações da frota de qualquer lugar e hora, além de obter informações que auxiliam na tomada de decisões. O sistema foi desenvolvido com o uso de tecnologias *web*, da linguagem de programação Java e de *frameworks* para criação e manipulação da interface, entre eles estão o Bootstrap e AngularJS. O ambiente de desenvolvimento utilizado foi o Eclipse MARS em conjunto com o servidor de aplicação Tomcat na versão 7. O banco de dados PostgreSQL foi usado para a persistência de dados.

Palavras-chave: Gestão de frotas. Controle de veículos. Desenvolvimento web.

## **ABSTRACT**

This work aims to present the development of a web system that helps people and companies managing their vehicles. The system allow the user to perform fleet control through Internet and provide reports like: average consumption, distance traveled, feasibility, expenses and financial impact. Thus, it allows monitoring operations of the fleet anywhere and anytime. This system was developed using web technologies, using the Java programming language and making use of frameworks for creating and manipulating the interface, like Bootstrap and AngularJS. The development environment used was Eclipse Mars with the version seventh Tomcat application server. The PostgreSQL database was employed for data persistence.

**Key-words:** Fleet management. Vehicle control. Web development.

## LISTA DE FIGURAS

Figura 1 - Tela inicial do sistema .....	18
Figura 2 - Tela de relatório de reservas .....	18
Figura 3 - Tela de consulta de reservas .....	19
Figura 4 - Tela de relatórios do sistema .....	19
Figura 5 - Tela inicial do sistema Sofit.....	20
Figura 6 - Casos de uso do <code>Usuario</code> e <code>Administrador</code> .....	23
Figura 7 - Casos de uso do <code>Motorista</code> .....	23
Figura 8 - Diagrama de classes dos modelos.....	24
Figura 9 - Tela inicial do <code>pgAdmin</code> .....	26
Figura 10 - Tela inicial do eclipse <code>MARS</code> .....	27
Figura 11 - Tela de autenticação do sistema.....	28
Figura 12 - Tela inicial do sistema .....	29
Figura 13 - Mapa de navegação do cadastro de veículo.....	30
Figura 14 - Tela de cadastros.....	30
Figura 15 - Tela de visualização do veículo .....	33
Figura 16 - Tela de visualização do histórico do veículo .....	33
Figura 17 - Tela de visualização da agenda do veículo.....	34
Figura 18 - Mapa de navegação do controle da frota .....	35
Figura 19 - Tela de controle da frota .....	35
Figura 20 - Tela de cadastro de agenda .....	36
Figura 21 - Tela de controle de abastecimento.....	36
Figura 22 - Tela de cadastro de abastecimento .....	37
Figura 23 - Tela de cadastro de manutenção .....	37
Figura 24 - Tela de cadastro de quilometragem .....	38
Figura 25 - Tela de cadastro de multa .....	38
Figura 26 - Tela de geração de relatório.....	38
Figura 27 - Relatório de consumo médio .....	39
Figura 28 - Relatório de distância percorrida .....	40
Figura 29 - Tela de viabilidade de veículo .....	40
Figura 30 - Aviso de economia .....	40
Figura 31 - Aviso de despesas .....	41

Figura 32 - Tela de impacto financeiro .....	41
Figura 33 - Gráficos com resultados da pesquisa .....	43

## LISTA DE QUADROS

Quadro 1 - Requisitos funcionais .....	22
Quadro 2 - Requisitos não funcionais .....	22
Quadro 3 - Classes com métodos responsáveis por retornar as páginas de visualização .....	29
Quadro 4 - Configuração do local e terminação das páginas .....	29
Quadro 5 - Método cadastra da classe VeiculoController .....	31
Quadro 6 - Método cadastra da classe VeiculoDAO.....	31
Quadro 7 - Método getVeiculo .....	32
Quadro 8 - Método visualiza.....	32
Quadro 9 - Comparação com os trabalhos correlatos.....	42
Quadro 10 - Detalhamento do caso de uso UC03 - Manter veículos .....	49
Quadro 11 - Detalhamento do caso de uso UC05 - Manter manutenções.....	50
Quadro 12 - Detalhamento do caso de uso UC09 - Emitir relatório de viabilidade do veículo .....	50
Quadro 13 - Questionário de avaliação do sistema .....	51

## LISTA DE ABREVIATURAS E SIGLAS

ANFAVEA - Associação Nacional dos Fabricantes de Veículos Automotores

AMMVI - Associação dos Municípios do Médio Vale do Itajaí

CSS – *Cascading Style Sheets*

HTML – *HyperText Markup Language*

HTTP - *Hypertext Transfer Protocol*

JSON – *JavaScript Object Notation*

JSP – *Java Server Pages*

JVM – *Java Virtual Machine*

MVC – *Model View Controller*

REST – *Representational State Transfer*

RF – Requisitos Funcionais

RNF – Requisitos Não Funcionais

SOAP - *Simple Object Access Protocol*

UC – Casos de Uso

URL – *Uniform Resource Locator*

W3C – *World Wide Web Consortium*

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>13</b>
1.1 OBJETIVOS.....	13
1.2 ESTRUTURA.....	14
<b>2 FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>15</b>
2.1 GESTÃO DE FROTAS.....	15
2.2 SISTEMA WEB .....	16
2.3 TRABALHOS CORRELATOS .....	17
<b>3 DESENVOLVIMENTO .....</b>	<b>21</b>
3.1 LEVANTAMENTO DE INFORMAÇÕES .....	21
3.2 ESPECIFICAÇÃO .....	21
3.2.1 Diagrama de Casos de Uso .....	22
3.2.2 Diagrama de classes .....	23
3.3 IMPLEMENTAÇÃO .....	25
3.3.1 Técnicas e ferramentas utilizadas.....	25
3.3.2 Operacionalidade da implementação .....	28
3.4 RESULTADOS E DISCUSSÕES.....	41
<b>4 CONCLUSÕES.....</b>	<b>45</b>
4.1 EXTENSÕES .....	45
<b>APÊNDICE A – DESCRIÇÃO DOS CASOS DE USO .....</b>	<b>49</b>
<b>APÊNDICE B – QUESTIONÁRIO DE AVALIAÇÃO.....</b>	<b>51</b>

## 1 INTRODUÇÃO

De acordo com um estudo revelado pela Associação Nacional de Veículos Automotores (ANFAVEA), estima-se que a frota brasileira sofrerá um aumento de 140% chegando em 95,2 milhões em 2034 (ANFAVEA, 2014, p. 1). É um número expressivo e fundamental para a economia brasileira que chega a movimentar R\$150 bilhões anualmente (CNMCUT, 2010).

Há muito tempo os veículos deixaram de ser status e tornaram-se necessidade nas empresas, podendo gerar lucro ou prejuízo ao negócio, a depender da forma de utilização e gestão da frota. Uma empresa inicia sua frota com o primeiro veículo, motivo esse, que não dispensa uma gestão cuidadosa (ROTA EXATA, 2015).

Ao adquirir um veículo, as empresas tentam extrair o máximo de produtividade do mesmo, contudo, por muitas vezes não atingem o melhor custo/benefício devido a uma má gestão da frota. Entre os principais erros em uma boa gestão de frotas, encontram-se a falta de controle de manutenções, a má utilização por parte dos motoristas e a falta de interesse dos gestores (COLOMBO, 2012). Qualidade e produtividade formam a base para o aumento da eficiência, na administração de transportes e veículos. Para as empresas, isso levará à rentabilidade e competitividade, contribuindo para o seu crescimento sustentável. Para amenizar esses problemas, é necessário aplicar formas mais efetivas de controle das operações que são realizadas com os veículos da frota. Uma das formas para se alcançar isso é por meio de uma maior eficiência no registro e monitoramento dessas operações (COLOMBO, 2012). Atualmente, com a Internet disponível em diversos locais e com o uso de dispositivos móveis, é possível ter acesso a um sistema em diversos locais, sem necessidade de um terminal fixo.

Portanto, percebeu-se a necessidade de desenvolver um sistema web, que pode ser utilizado em qualquer local com acesso à Internet, para auxiliar na gestão e controle da frota. Com isso, busca-se obter um maior controle da frota e suas operações, reduzir as despesas relacionadas aos veículos e conseguir um maior retorno financeiro à empresa.

### 1.1 OBJETIVOS

O objetivo geral deste trabalho é desenvolver um sistema *web* com funções que permitam as empresas realizar uma e controle de frotas controlando a agenda, manutenção, abastecimento, quilometragem e multas dos veículos.

Os objetivos específicos do trabalho são:

- a) permitir o acompanhamento e controle dos veículos e motoristas, a partir da manipulação dos dados de viagens, abastecimento e manutenção dos mesmos;

- b) agilizar as atividades e operações referentes a gestão da frota por meio de um serviço disponível online e com módulos específicos para cada tipo de ação;
- c) fornecer informações relevantes sobre a frota, que auxiliem na tomada de decisões da empresa;
- d) possibilitar a utilização do sistema para uso pessoal.

## 1.2 ESTRUTURA

No primeiro capítulo tem-se a introdução ao tema principal deste trabalho com a apresentação da justificativa e dos objetivos.

No segundo capítulo apresenta-se a fundamentação teórica sobre gestão de frotas, sistema *web* e os trabalhos correlatos.

O terceiro capítulo apresenta o desenvolvimento do trabalho, detalhando o levantamento de informações, diagramas de Caso de uso e Classe, implementação do sistema e ao final os resultados e discussões.

No quarto capítulo tem-se as conclusões deste trabalho, bem como apresenta-se sugestões para trabalhos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA

Esse capítulo aborda assuntos a serem apresentados nas seções a seguir, tais como gestão de frotas, sistema *web* e os trabalhos correlatos.

### 2.1 GESTÃO DE FROTAS

De acordo com Valente, Passaglia e Novaes (1997), o termo “Gestão de Frotas” representa a atividade de reger, administrar ou gerenciar um conjunto de veículos pertencentes a uma mesma empresa. Esta tarefa tem uma abrangência bastante ampla e envolve diferentes serviços, como dimensionamento, especificação de equipamentos, roteirização, custos, manutenção e renovação de veículos, entre outros (VALENTE; PASSAGLIA; NOVAES, 1997).

A gestão da frota surgiu quando as empresas precisavam de vários veículos para fazer entregas, viajar para falar com clientes, transportar equipamentos, etc. Grande parte dos clientes localizam-se em diferentes lugares. Para os administradores convinha-lhes gerir bem o transporte de modo a reduzir os custos, os desperdícios e aumentar os lucros (CLEMENTE, 2008, p. 11). Também, de acordo com Clemente (2008, p. 11), a gestão de frotas consiste na utilização de métodos, técnicas e ferramentas que permitem às empresas eliminarem os riscos inerentes ao investimento dos seus veículos e, assim, aumentarem a produtividade e eficiência das suas operações.

Para Valente et al. (2003) a gestão de frotas envolve atividades como: gerenciar, comandar, administrar e planejar as atividades de um conjunto de veículos pertencentes a uma mesma empresa. Dário et al. (2012) comenta que o responsável por administrar a frota deve possuir conhecimento necessário para execução de atividades como:

- a) controlar custos operacionais, realizando o controle de gastos e receitas geradas pelos veículos;
- b) planejar manutenções, identificando a necessidade de cada veículo de modo a solucionar os problemas da melhor forma possível;
- c) dimensionar a frota, apontando a demanda da empresa e o número de veículos necessários para suprir essa necessidade.

Segundo D’Ávilla (2006), a produtividade no setor de transporte apresenta valores muito baixos quando comparados com outros setores da economia. Também, de acordo com o mesmo autor, o cenário atual caracteriza-se pela utilização de veículos com características inadequadas, má conservação das vias, congestionamento, etc. A má utilização de alguns

veículos inicia-se na aquisição. É comum gestores efetuarem a compra de veículos baseado apenas em critérios como preço de revenda, sendo que o fator de análise principal deve ser a adequação do veículo ao serviço (D'ÁVILLA, 2006).

Ao analisar o gasto de combustível, pneus e manutenções da frota, é possível identificar os pontos que mais demandam despesas, assim controlar os mesmos e ter uma gestão mais eficaz, contribuindo com a saúde econômica da empresa. Coelho (2011) comenta que isso contribui para uma melhor rentabilidade e competitividade, contribuindo para o crescimento sustentável.

## 2.2 SISTEMA WEB

A *web* é um dos serviços oferecidos na Internet, concebido para disponibilizar permanentemente informações a todos os usuários. Desde 1991, quando foi criada por Tim Berners-Lee, que também criou o *World Wide Web Consortium* (W3C), a *web* é utilizada em áreas cada vez mais diversas da atividade humana (informação, comunicação, educação, comércio, saúde, acesso a serviços públicos e privados, entre outras) e vem ganhando muita importância na vida cotidiana das pessoas (W3C, 2013a). Sistemas *web* são construídos utilizando diferentes linguagens como, por exemplo, HTML, CSS, JavaScript e Java.

Segundo a W3C (2016b, p. 7), "HTML é uma abreviação de *Hypertext Markup Language* - Linguagem de Marcação de Hipertexto. Resumindo em uma frase: o HTML é uma linguagem para publicação de conteúdo (texto, imagem, vídeo, áudio e etc) na Web" É utilizada para indicar a existência de parágrafos, figuras, tabelas e outras estruturas nas páginas *web*. HTML é uma linguagem de marcação, pois as estruturas definidas por ela são organizadas em *tags* (rótulos).

Conforme a W3C (2016c, p. 5), "O CSS formata a informação entregue pelo HTML. Essa informação pode ser qualquer coisa: imagem, texto, vídeo, áudio ou qualquer outro elemento criado." CSS é uma linguagem de marcação que define a aparência das páginas web. Por exemplo, enquanto HTML define a existência de uma tabela, CSS é usada para definir a localização, a borda, as cores e a formatação do texto dentro dessa tabela. É definida em folhas de estilos usada para a estilização da página web ou seja estrutura a informação para que ela seja consumida da melhor maneira possível (W3C, 2016c, p. 5).

De acordo com Grillo e Fortes (2008, p. 4), "*JavaScript* permite criar pequenos programas embutidos no próprio código de uma página HTML e capazes de gerar números, processar alguns dados, verificar formulários, alterar valor de elementos HTML e criar

elementos HTML." Uma vez que o interpretador de Javascript está incluído no navegador do cliente, isto o torna independente de sistemas operacionais, assim sendo executado em Windows, Macintosh, Linux e outros sistemas.

Java é uma linguagem de programação orientada a objeto e desenvolvida pela Sun Microsystems na década de 90, que atualmente, pertence a Oracle (BRITO, 2014). Segundo Cipoli (2015), o termo também se refere a diversas aplicações utilizadas no nosso cotidiano, entre elas o sistema operacional Android. Diferentemente do JavaScript, o Java é uma linguagem que precisa ser compilada e necessita da Máquina Virtual Java (JVM) instalada na máquina para que os programas sejam executados (CAELUM, 2015). No desenvolvimento de sistemas *web*, Java é utilizado para a programação no lado do servidor. Especificamente no desenvolvimento web, a linguagem Java pode ser utilizado na forma de Java Server Pages (JSP).

No desenvolvimento web é comum o uso de *frameworks* e bibliotecas para agilizar e prover maior qualidade à implementação dos sistemas. Alguns dos *frameworks* mais utilizados na indústria são o Bootstrap, AngularJS e o Spring. De acordo com Utterback (2014), Bootstrap é um conjunto de vários elementos com funções personalizáveis para projetos web. Esses elementos personalizáveis são uma combinação de HTML, CSS e Javascript. AngularJS é um *framework* Javascript mantido pelo Google. Nakamashi (2013) comenta que o principal objetivo do AngularJS é simplificar o desenvolvimento de aplicações web, trazendo mais dinamismo às páginas HTML e fazendo com que a parte da lógica seja executada na máquina do cliente. O Spring é um *framework* aberto e criado para embarcar a complexidade do desenvolvimento de aplicações corporativas. Uma de suas principais vantagens é possuir uma arquitetura dividida em módulos independentes. Isso permite ao desenvolvedor de um sistema web selecionar somente os módulos necessários (ANSELMO, 2016. p. 3).

### 2.3 TRABALHOS CORRELATOS

Entre os trabalhos correlatos encontrados, pode-se citar os trabalhos de conclusão de curso de Sorroche (2010) e Hoffmann (2010) ambos desenvolvidos na Universidade Regional de Blumenau, e o sistema comercial desenvolvido pela Sofit.

No trabalho realizado por Sorroche (2010), foi desenvolvido um sistema *web* para controle e gestão de uma frota de veículos da Associação dos Municípios do Médio Vale do Itajaí (AMMVI). Segundo Sorroche (2010, p. 24), o seu sistema tem como principal

funcionalidade gerenciar os dados das reservas e locações da frota. Além disso, contém funções para cadastro e geração de relatórios, tanto a nível de conferência quanto gerenciais. A Figura 1 ilustra a tela inicial do sistema com as permissões de administrador.

Figura 1 - Tela inicial do sistema

Código	Hora Inicial	Hora Final	Veículo	Colaborador	Transação	Observações
4	12:00:00	15:00:00	Gol-3	Roy Scheider	Reserva	
1	13:30:00	15:00:00	Palio-1	Roy Scheider	Reserva	Prefeitura

Sorroche (2010, p. 41) salienta que na geração do relatório gerencial de gastos é possível utilizar filtros, desse modo o relatório poderá ser separado por veículos ou por intervalos de data. A Figura 2 apresenta a tela de geração de relatórios de reservas dos veículos.

Figura 2 - Tela de relatório de reservas

Código	Colaborador	Veículo	Cidade	Status	Observações
1	Roy Scheider	Palio	Blumenau	Fechada	

De acordo com Sorroche (2010), o secretário executivo da AMMVI comentou que o sistema teve grande importância na gestão da frota, possibilitando o acompanhamento dos custos relacionados à frota, as manutenções efetuadas e o controle de utilização dos veículos pertencentes à organização.

No trabalho executado por Hoffmann (2010) foi desenvolvido um sistema *desktop*, com a premissa de controlar a frota de veículos da cooperativa de saúde Unimed Blumenau. Antes da elaboração do sistema, a cooperativa realizava o controle da sua frota através de planilhas eletrônicas onde eram criados novos documentos contendo os dados do respectivo ano. O sistema foi desenvolvido em Java, utilizando a ferramenta NetBeans e o banco de dados MySQL. O projeto apresenta módulos para registro e reservas de veículos,

acompanhamento das manutenções e relatórios de movimentação da frota. Na Figura 3 tem-se a tela principal do sistema, com a função de consultar as reservas da frota

Figura 3 - Tela de consulta de reservas

Modelo	Placa	Data Saida	Hora	Data Retorno	Hora	Destino	Condutor
Celta	MEC-3798	02/11/2010	12:00:00	03/11/2010	10:00:00	teste	Administração
Celta	MEC-3798	03/11/2010	11:00:00	03/11/2010	11:59:00	teste	Administração
Celta	MEC-3798	03/11/2010	12:00:00	05/11/2010	12:00:00	teste aiala	Administração

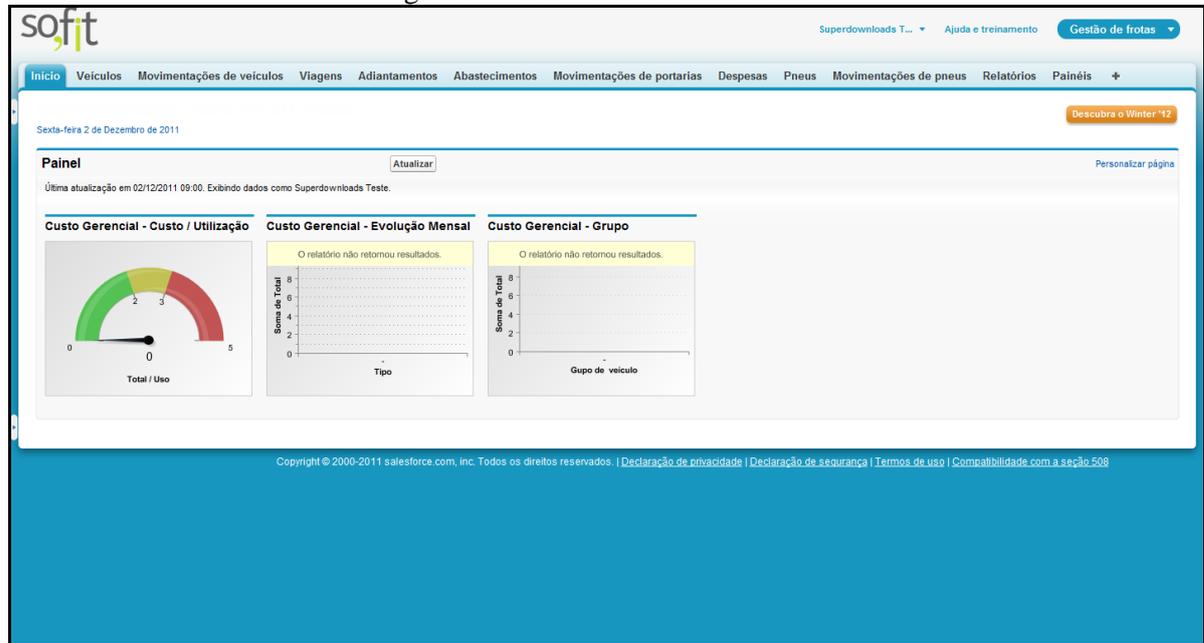
O menu para geração de relatórios conforme a Figura 4 apresenta os tipos de relatórios disponíveis, onde é possível selecionar o condutor, veículo e o período desejado.

Figura 4 - Tela de relatórios do sistema

Segundo Hoffman (2010, p. 45), o desenvolvimento desse sistema proporcionou maior agilidade nos processos, segurança e integridade dos dados cadastrados. O autor salienta que com cadastro dos condutores o sistema impossibilita que condutores com habilitação vencida tenham reservas de veículos liberadas evitando possíveis despesas com multas.

Entre os sistemas comerciais encontrados, pode-se citar o Sofit, que está no mercado há mais de 5 anos (SOFIT, 2015). A Figura 5 apresenta a tela inicial do sistema Sofit, contendo os atalhos para acesso a diversos módulos, com destaque para o que traz informações sobre custos dos veículos.

Figura 5 - Tela inicial do sistema Sofit



De acordo com os fundadores, Amaral et al. (2016), a Sofit é um sistema para gestão de manutenção de frotas com a premissa de reduzir em até 17% os gastos referentes a manutenção dos veículos. Conta com módulos de controle de veículos, visibilidade de gastos a curto e médio prazo, tempo de manutenção dos equipamentos da frota e painel de informações gerais sobre a frota. Com valores que variam entre R\$199 e R\$899, de acordo com o número de veículos, a empresa possui clientes expressivos, como LATAM, GERDAU e GVT.

### 3 DESENVOLVIMENTO

Neste capítulo são abordadas as etapas de desenvolvimento do sistema desenvolvido, constituído pelo levantamento das informações e sua especificação, formado pelos requisitos funcionais e não funcionais, diagramas de casos de uso, e diagramas de classes. Posteriormente são apresentadas as técnicas e ferramentas utilizadas, além da operacionalidade do sistema. Por fim é explanado os resultados e discussões.

#### 3.1 LEVANTAMENTO DE INFORMAÇÕES

O sistema proposto tem por objetivo auxiliar pequenas e médias empresas na gestão da frota dos seus veículos. O sistema é *opensource*, o código fonte do sistema está disponibilizado na *web* em um repositório do GitHub através do endereço [github.com/rodrigopereira/tcc](https://github.com/rodrigopereira/tcc), assim como as instruções para sua instalação e personalização. Com esse sistema, espera-se agilizar o processo de registros dos dados, reduzir o consumo de documentos físicos e obter relatórios sobre a atual situação da frota de veículos.

Os componentes do sistema se ajustam ao tamanho da tela do usuário, ou seja, é totalmente responsivo. Dessa forma, mantém-se uma boa aparência em qualquer dispositivo que irá acessá-lo. Os atalhos fornecidos para acesso às funções contêm ícones que auxiliam na identificação da ação a ser executada. São disponibilizados módulos para o cadastro de motoristas, veículos, quilometragem, manutenções corretivas e preventivas, multas e emissão de relatórios contendo informações relevantes sobre a situação da frota.

O sistema também permite visualizar o histórico dos veículos e motoristas, consultar o consumo médio, distância percorrida, viabilidade, despesas totais com combustível, multas, revisões e manutenções, impacto financeiro da frota sobre a empresa. Essas funções fornecem ao gestor informações sobre as despesas relacionados aos veículos.

#### 3.2 ESPECIFICAÇÃO

Nesta seção são apresentadas as especificações do sistema, com os Requisitos Funcionais (RF) e Requisitos Não Funcionais (RNF), diagrama de casos de uso e diagrama de classes. Para a criação dos diagramas utilizou-se a ferramenta de modelagem Astah. A descrição dos principais Casos de Uso (UC) é apresentada no Apêndice A.

No Quadro 1 são apresentados os RF's previstos para o sistema e sua rastreabilidade, ou seja, vinculação com o caso de uso associado.

Quadro 1 - Requisitos funcionais

Requisitos Funcionais	Caso de Uso
RF01: O sistema deve permitir o usuário e o motorista a efetuar <i>login</i> no sistema.	UC01
RF02: O sistema deve permitir o usuário e o motorista alterar a senha do <i>login</i> .	UC02
RF03: O sistema deve permitir o usuário manter veículos.	UC03
RF04: O sistema deve permitir o usuário manter motoristas.	UC04
RF05: O sistema deve permitir o usuário e o motorista manter manutenção preventiva e corretiva do veículo.	UC05
RF06: O sistema deve permitir o usuário emitir relatório de histórico do veículo.	UC06
RF07: O sistema deve permitir o usuário emitir relatório de histórico do motorista.	UC07
RF08: O sistema deve permitir o administrador manter usuários e suas permissões.	UC08
RF09: O sistema deve permitir o administrador emitir relatório de viabilidade dos veículos.	UC09
RF10: O sistema deve permitir o usuário emitir relatórios de manutenções.	UC10
RF11: O sistema deve permitir o usuário e o motorista manter abastecimentos.	UC11
RF12: O sistema deve permitir o usuário manter multas.	UC12
RF13: O sistema deve permitir o usuário manter fornecedores.	UC13
RF14: O sistema deve permitir o usuário e o motorista manter quilometragens.	UC14
RF15: O sistema deve permitir o administrador emitir relatório de consumo médio.	UC15
RF16: O sistema deve permitir o administrador emitir relatório de despesas.	UC16
RF17: O sistema deve permitir o administrador emitir relatório de distância percorrida.	UC17
RF18: O sistema deve permitir o administrador emitir relatório de impacto financeiro.	UC18

O Quadro 2 lista os RNF's previstos para o sistema

Quadro 2 - Requisitos não funcionais

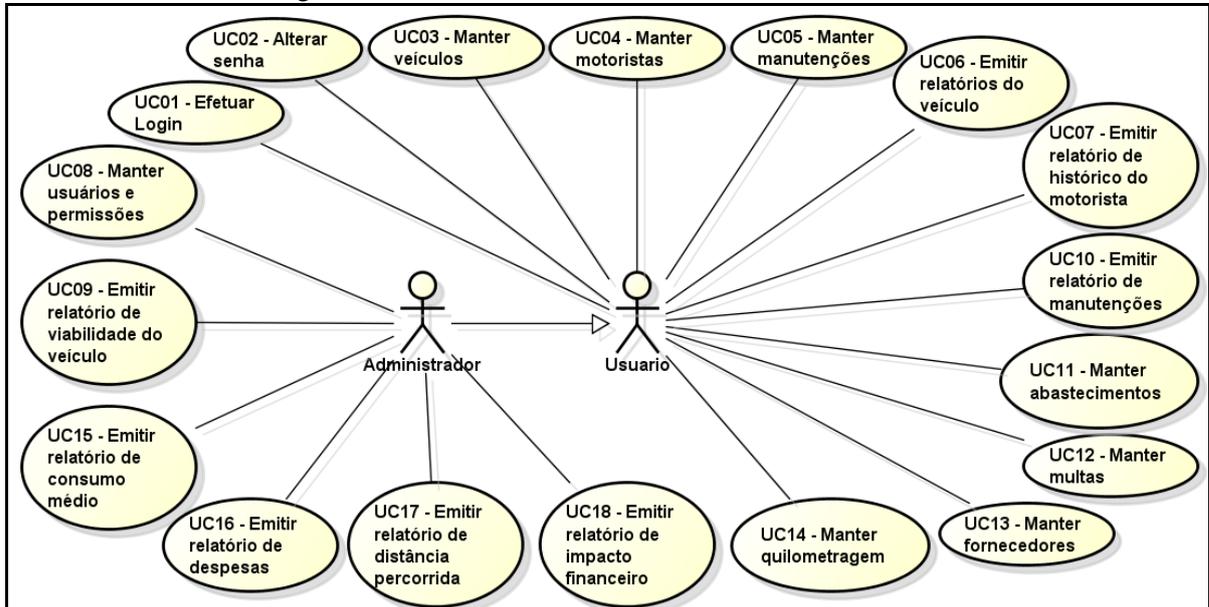
Requisitos Não Funcionais
RNF01: O sistema deverá ser implementado em Java 1.8.
RNF02: O sistema deverá ser executado em plataforma web.
RNF03: O sistema deverá ser responsivo.
RNF04: O sistema deverá utilizar o <i>framework</i> Bootstrap para criação da interface.
RNF05: O sistema deverá utilizar banco de dados Postgresql 9.4.
RNF06: O sistema deverá ser acessível via Mozilla Firefox (versão 29.0 ou superior) e Google Chrome (versão 33 ou superior).
RNF07: O sistema deverá ser implementado na arquitetura MVC utilizando REST.
RNF08: O sistema deverá utilizar AngularJS como <i>framework</i> Javascript.
RNF09: JasperReport para geração de relatório

### 3.2.1 Diagrama de Casos de Uso

Na Figura 6 tem-se o diagrama de casos de uso do sistema proposto dos atores *Usuario* e *Administrador*. O ator *Usuario* tem por responsabilidade a interação com o

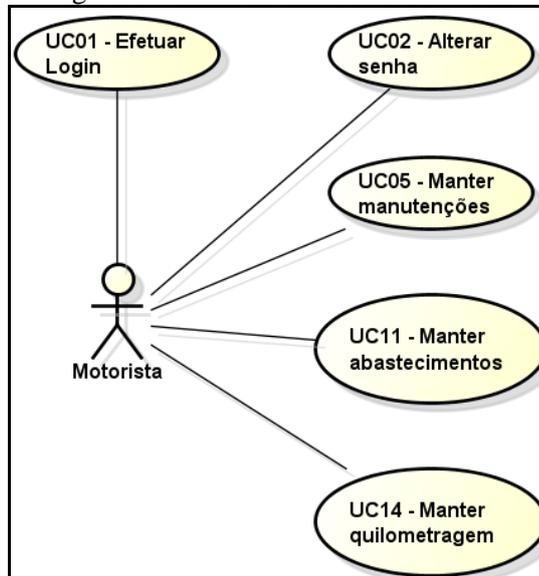
cadastro dos dados e o Administrador, além das funções de usuário, pode cadastrar novos usuários, manter suas permissões e gerar relatórios da frota.

Figura 6 - Casos de uso do Usuário e Administrador



A Figura 7 mostra o diagrama com os casos de uso do ator Motorista, que além de controlar suas informações como usuário, tem a função de registrar as operações realizadas com os veículos no momento em que elas ocorrem.

Figura 7 - Casos de uso do Motorista



### 3.2.2 Diagrama de classes

Na Figura 8 é apresentado o diagrama de classes da camada de modelo que compõe o sistema. Essas classes mantêm as informações do sistema.



- d) *Motorista* – herda as propriedades da classe *Funcionario*, registrando as informações dos usuários que utilizam a frota e registram suas atividades que são realizadas com ela. Possui um atributo da classe *CNH*, contendo o número, validade e categoria da habilitação;
- e) *Multa* – registra as ocorrências de infrações de trânsito da frota durante o exercício de uma atividade com um veículo da frota. Possui um motorista, um veículo e as informações de pagamento da infração;
- f) *Agenda* – registra os agendamentos de atividades da frota, possuindo um veículo e a opção de fornecedor;
- g) *Manutencao* – registra as atividades de manutenção da frota. Pode ser *corretiva* ou *preventiva*, contém o status da manutenção variando entre *andamento* e *finalizado*;
- h) *Veiculo* – registra os dados dos veículos da frota;
- i) *Km* – utilizada no controle de viagens dos veículos, contém informações de datas e horários das viagens, distância percorrida e os respectivos motoristas;
- j) *Pagamento* – registra os gastos referentes à frota, como pagamentos das multas, abastecimentos e manutenções.

### 3.3 IMPLEMENTAÇÃO

Nesta seção são expostas as ferramentas e técnicas adotadas na implementação e operacionalidade do sistema.

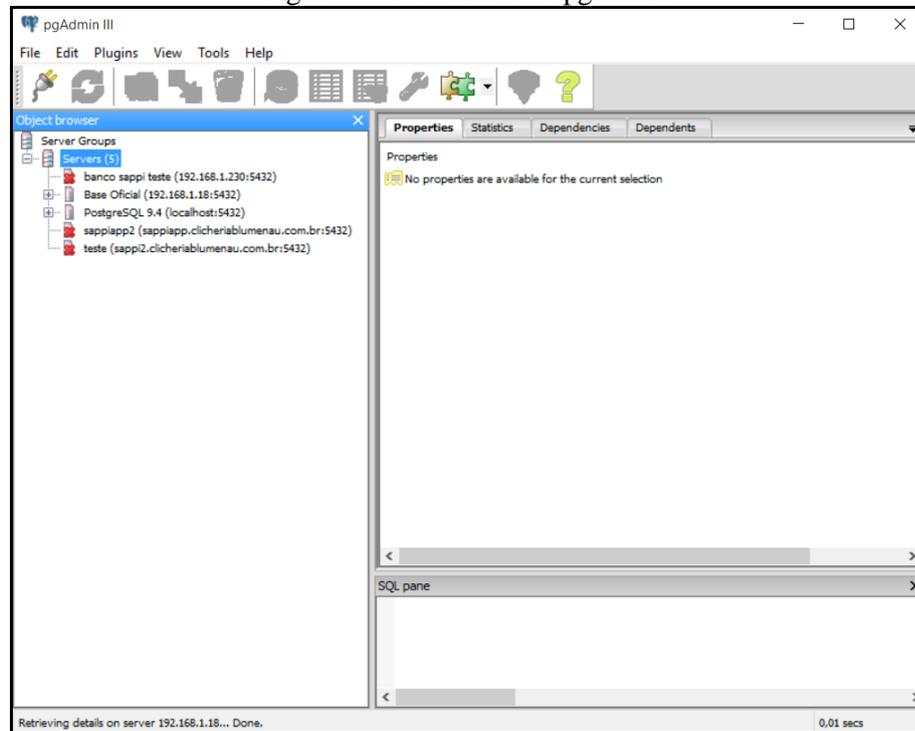
#### 3.3.1 Técnicas e ferramentas utilizadas

No desenvolvimento do sistema foram adotados o padrão de arquitetura de software *Model-View-Controller* (MVC) e os protocolos definidos na arquitetura *Representational State Transfer* (REST). De acordo com Lima (2012, p. 25), REST é uma abstração das melhores práticas dos estilos Cliente/Servidor e sistema em camadas. A transferência de dados entre os módulos cliente (*front-end*) e servidor (*back-end*) ocorre no formato *JavaScript Object Notation* (JSON). O MVC tem como objetivo separar dados ou lógica de negócios (Modelo) da interface do usuário (Visualização) e do fluxo da aplicação (Controlador) (BAPTISTELLA, 2016).

O código do sistema no módulo servidor foi feito em Java utilizando o sistema gerenciador de banco de dados PostgreSQL para a persistência e manutenibilidade dos dados.

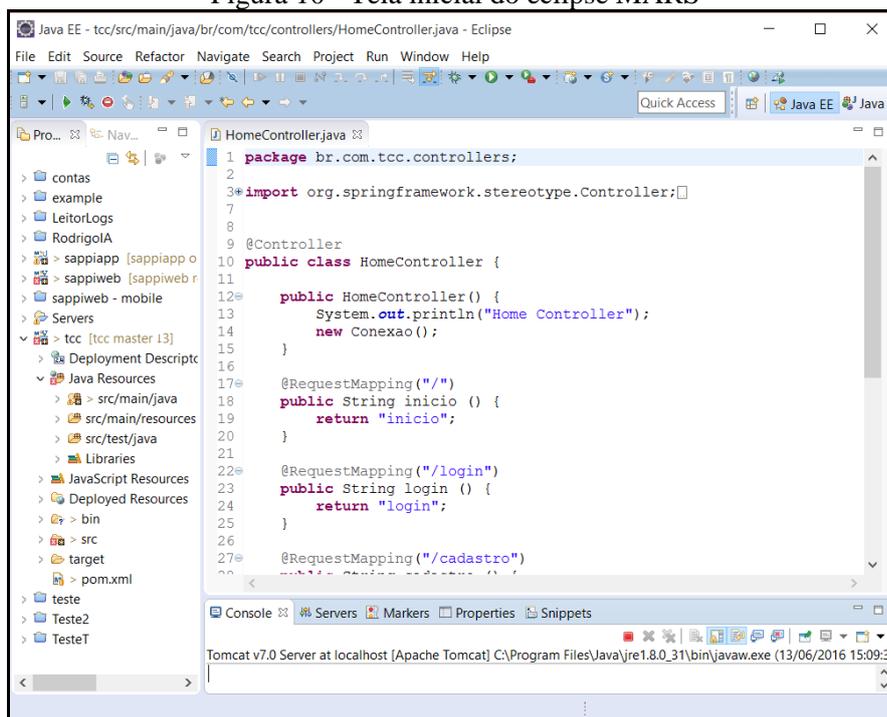
O PostgreSQL oferece um gerenciador gráfico padrão, chamado pgAdmin, no qual é possível elaborar as consultas, criar e modelar as tabelas e administrar os dados cadastrados. A Figura 9 mostra a tela inicial do pgAdmin.

Figura 9 - Tela inicial do pgAdmin



O Eclipse MARS foi utilizado como ambiente de desenvolvimento, devido à facilidade de integração das tecnologias citadas anteriormente e por ser gratuito. Na Figura 10 é ilustrado a tela inicial do Eclipse MARS.

Figura 10 - Tela inicial do eclipse MARS



De modo a facilitar o gerenciamento e *download* dos *drivers* necessários para a execução do projeto foi utilizado o *framework* Maven. O Maven é uma ferramenta de integração de projetos, responsável por gerenciar dependências e controlar versão. (DClick, 2010). Para a criação e mapeamento dos controladores e configurações de segurança foi adotado o *framework* Spring 3.\* com suas versões MVC e Security. Como aplicativo servidor foi utilizado o Tomcat na versão 7. O lado cliente do sistema foi desenvolvido em HTML, em conjunto com as tecnologias CSS, JavaScript, Bootstrap, AngularJS e JSP.

O *framework* Spring foi utilizado no projeto para gerenciar as camadas de visualização e controle e criar o mecanismo de autenticação e acesso ao sistema. Como foi citado na seção 2.2, esse *framework* disponibiliza diversos módulos que podem ser utilizados de acordo com a necessidade do sistema (ANSELMO, 2016. p. 3). Os módulos adotados para este projeto são: *Spring Framework MVC*, utilizado para desacoplar as camadas de modelo, visualização e controle; e *Spring Security*, utilizado na implementação das operações de segurança.

No *framework* Spring, o mapeamento das classes Java que atuam como controladores se dá pela tag `@Controller` e dos métodos controladores dessas classes é feito por meio da anotação `@RequestMapping`. Esta tag indica o *Uniform Resource Locator* (URL) que posteriormente será invocado pelo usuário, onde será executado o método logo abaixo da anotação.

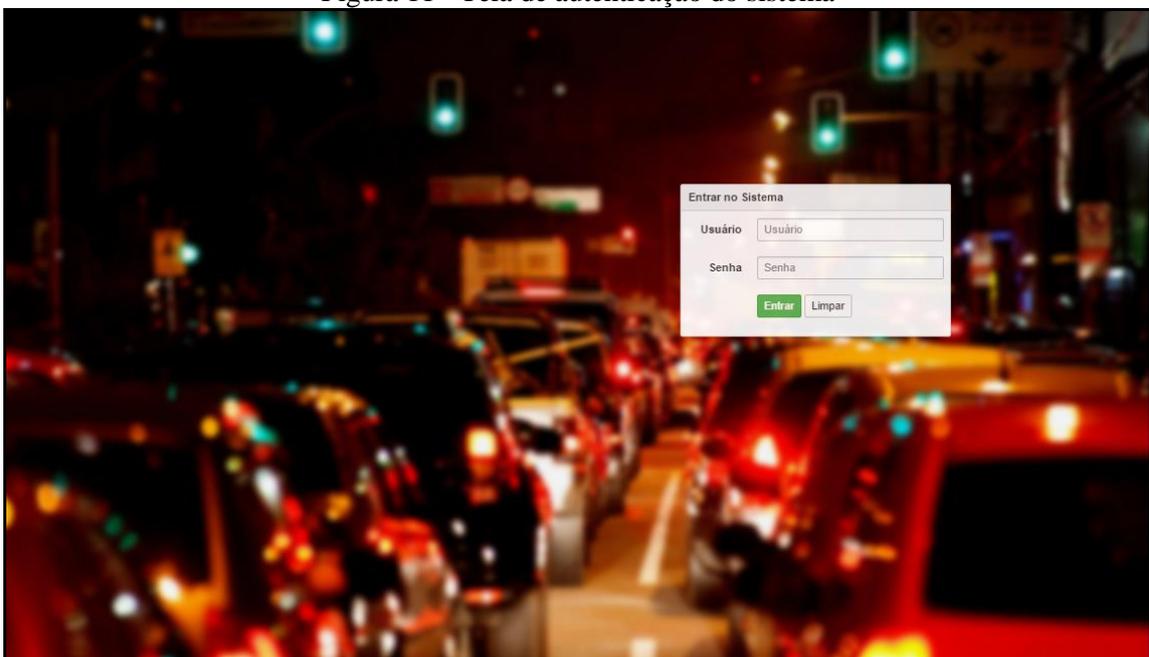
Cada controlador possui características que indicam o tipo de requisição que é aceita e o tipo de dados que vai ser consumido. Neste projeto foi adotado o conceito REST, cujos

protocolos são POST, GET, PUT e DELETE. De acordo com Coutinho (2015), REST possibilita a criação de *web services*, cujas diferenças em relação ao modelo *Simple Object Access Protocol* (SOAP) estão na utilização semântica dos métodos HTTP (GET, POST, PUT e DELETE). Uma das principais características do REST é a acessibilidade, na qual é possível utilizar a mesma URL para diversos métodos diferindo apenas o protocolo aceito. No sistema proposto foi adotada a prática mais comum no desenvolvimento de sistemas web (COUTINHO, 2015), que foi utilizar GET para recuperar, POST para criar, PUT para alterar e DELETE para apagar.

### 3.3.2 Operacionalidade da implementação

O sistema é acessado por meio de uma URL disponibilizada aos usuários. A tela de autenticação apresenta um formulário para o preenchimento do usuário e senha do utilizador. Após preenchimento dos dados é realizada a comprovação das informações no banco de dados e direcionado a tela inicial do sistema. A Figura 11 mostra a tela de autenticação, solicitando as credenciais do usuário.

Figura 11 - Tela de autenticação do sistema



A Figura 12 ilustra a tela inicial do sistema que é apresentada após a autenticação correta do usuário. Essa tela contém o menu com acesso as funcionalidades separadas em 4 opções: Cadastros, Controle de Frota, Relatórios e Sair.

Figura 12 - Tela inicial do sistema



Cada opção contém uma URL única que dá acesso a determinada página do sistema. A URL é definida pela anotação `@RequestMapping`. Ao clicar em uma opção do menu é enviada uma requisição ao Spring e conforme o método correspondente é retornado o local e o nome da página específica exibindo o resultado no navegador. O Quadro 3 apresenta a classe `HomeController` onde ficam os controladores de visualização

Quadro 3 - Classes com métodos responsáveis por retornar as páginas de visualização

```

@Controller
public class HomeController {
    public HomeController() {
        new Conexao();
    }
    @RequestMapping("/")
    public String inicio () {
        return "inicio";
    }
    @RequestMapping("/login")
    public String login () {
        return "login";
    }
    @RequestMapping("/cadastro")
    public String cadastro () {
        return "cadastro/cadastro";
    }
    @RequestMapping("/frota")
    public String frota () {
        return "frota/frota";
    }
    @RequestMapping("/relatorio")
    public String relatorio () {
        return "relatorio/relatorio";
    }
}

```

A pasta raiz onde se encontra a estrutura dos arquivos é definido por meio da classe `InternalResourceViewResolver` no arquivo de configuração do Spring através da `tag prefix`, e a terminação dos arquivos é definida pela `tag suffix` demonstrada no Quadro 4.

Quadro 4 - Configuração do local e terminação das páginas

```

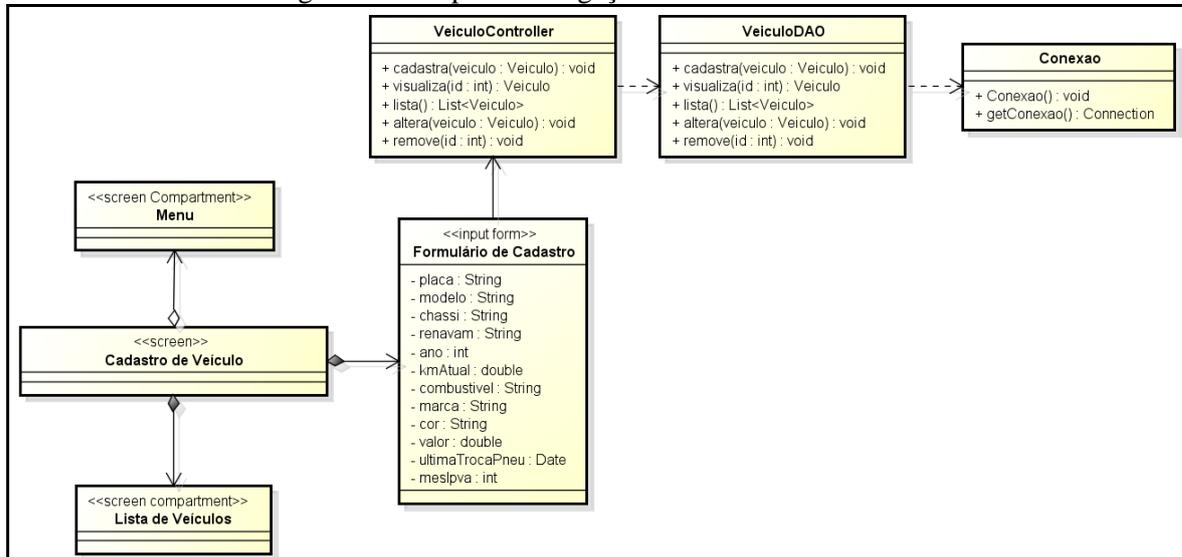
<bean
class="org.springframework.web.servlet.view.InternalResourceViewResolver">
    <property name="prefix" value="/WEB-INF/views/" />
    <property name="suffix" value=".jsp" />
</bean>

```

No menu `Cadastros`, mostrado na Figura 12, o usuário pode incluir, alterar ou excluir os veículos, usuários e fornecedores. Para listar os itens cadastrados, basta selecionar a opção no menu acima da tabela. Por padrão, são listados os veículos cadastrados. A Figura 13

apresenta um mapa de navegação do cadastro de veículo mostrando o fluxo de telas e a modelagem dos dados no servidor com os métodos e suas dependências.

Figura 13 - Mapa de navegação do cadastro de veículo



A inclusão de um veículo se dá pelo preenchimento do formulário disponibilizado a direita e, posteriormente, clicando no botão `Cadastrar`. Caso o usuário queira limpar o formulário, basta pressionar o botão `Cancelar`. A Figura 14 apresenta a tela de cadastro de veículo.

Figura 14 - Tela de cadastros

The screenshot shows the application interface with a table of vehicles and a registration form.

Placa	Descrição	Ano	Km Atual	Combustível	Marca	Cor
MIT4539	Biz	2010	46750	Gasolina	Honda	Preto
ARI4694	Gol	2010	40100	Gasolina	Volkswagen	Prata
DSO3154	Saveiro	2015	0	Gasolina	Volkswagen	Branco
MIT1322	Uno	2010	23000	Gasolina		

The registration form on the right includes the following fields:

- Placa
- Descrição
- Chassi
- Renavam
- Ano
- Km Atual
- Combustível
- Marca
- Cor
- Valor Pago
- Últ. Troca de Pneu
- Mês Pgto IPVA

Buttons: `Cadastrar` and `Cancelar`.

Após preencher o formulário de cadastro de veículo, o botão `Cadastrar` envia uma requisição para o método `cadastra` da classe `VeiculoController` passando os dados do veículo como parâmetro. O Quadro 5 apresenta o método `cadastra`.

Quadro 5 - Método cadastra da classe VeiculoController

```

@RequestMapping(method = RequestMethod.POST, consumes="application/json")
public void cadastra (@RequestBody Veiculo veiculo) {
    daoVeiculo.cadastra(veiculo);
}

```

Este método recebe requisições do tipo POST, com um objeto da classe `Veiculo` em formato JSON como parâmetro, após isso é invocado o método `cadastra` da classe `VeiculoDAO` instanciado na variável `daoVeiculo` responsável por fazer a persistência dos veículos no banco. O Quadro 6 apresenta o método `cadastra` responsável por inserir o cadastro no banco de dados. No início do método é criada uma variável `String` com nome `sql`, essa variável recebe o código `sql` a ser executado no banco de dados. Posteriormente, é instanciada uma variável `connection` responsável pela conexão com o banco de dados e executado o código `sql`.

Quadro 6 - Método cadastra da classe VeiculoDAO

```

public void cadastra(Veiculo veiculo){
    String sql = "insert into veiculo(placa, modelo, ano, chassi,
renavam, marca, cor, combustivel, km, valor, ultimatrocapi, mesipva)
values (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";

    try (Connection connection = Conexao.getConexao()) {

        PreparedStatement stmt = connection.prepareStatement(sql);
        stmt.setString(1, veiculo.getPlaca());
        stmt.setString(2, veiculo.getModelo());
        stmt.setInt(3, veiculo.getAno());
        stmt.setString(4, veiculo.getChassi());
        stmt.setString(5, veiculo.getRenavam());
        stmt.setString(6, veiculo.getMarca());
        stmt.setString(7, veiculo.getCor());
        stmt.setString(8, veiculo.getCombustivel());
        stmt.setDouble(9, veiculo.getKm());
        stmt.setDouble(10, veiculo.getValor());
        stmt.setTimestamp(11,
Data.retornaData(veiculo.getUltimaTrocaPneu()));
        stmt.setInt(12, veiculo.getMesIpva());

        stmt.execute();
        stmt.close();
        connection.close();

    } catch (SQLException e) {
        throw new RuntimeException(e);
    }
}

```

Para acessar um veículo cadastrado, é necessário dar 2 cliques rápidos em cima da linha do item escolhido. Ao executar esta ação, é disparada uma requisição para o método `getVeiculo`, apresentado no Quadro 7, passando o `id` do veículo escolhido.

Quadro 7 - Método `getVeiculo`

```

@RequestMapping(value="/{id}", method = RequestMethod.GET)
public Veiculo getVeiculo (@PathVariable int id) {
    return daoVeiculo.visualiza(id);
}

```

O método `getVeiculo` recebe requisições do tipo GET retornando um objeto `Veiculo`. Esse objeto é criado a partir dos dados retornados do banco de dados. O método `visualiza` efetua uma conexão com o banco de dados e realiza a seleção das informações baseado no `id` recebido como parâmetro. O Quadro 8 apresenta o método `visualiza`.

Quadro 8 - Método `visualiza`

```

public Veiculo visualiza(int id) {

    String sql = "select * from veiculo where id=?";
    try (Connection connection = Conexao.getConexao()) {
        PreparedStatement stmt = connection.prepareStatement(sql);
        stmt.setInt(1, id);
        ResultSet rs = stmt.executeQuery();
        Veiculo veiculo = new Veiculo();

        while (rs.next()) {
            veiculo.setId(rs.getInt("id"));
            veiculo.setPlaca(rs.getString("placa"));
            veiculo.setModelo(rs.getString("modelo"));
            veiculo.setAno(rs.getInt("ano"));
            veiculo.setChassi(rs.getString("chassi"));
            veiculo.setRenavam(rs.getString("renavam"));
            veiculo.setMarca(rs.getString("marca"));
            veiculo.setCor(rs.getString("cor"));
            veiculo.setCombustivel(rs.getString("combustivel"));
            veiculo.setKm(rs.getInt("km"));
            veiculo.setValor(rs.getDouble("valor"));
            veiculo.setUltimaTrocaPneu(
                rs.getTimestamp("ultimatrocapneu"));
            veiculo.setMesIpva(rs.getInt("mesipva"));
            veiculo.setHistorico(listaHistorico(rs.getInt("id")));
            veiculo.setAgenda(listaAgenda(rs.getInt("id")));
        }
        stmt.close();
        connection.close();

        return veiculo;
    } catch (SQLException e) {
        throw new RuntimeException(e);
    }
}

```

Após o retorno da requisição, é apresentada a tela com as informações do veículo cadastrado e mais 3 botões com as opções de salvar as alterações, excluir o item e sair da tela. A Figura 15 apresenta a tela de informações do veículo.

Figura 15 - Tela de visualização do veículo

Visualizar Veículo

Dados Histórico Agenda

Placa: ARI4894 Descrição: Gol

Chassi: Renavam:

Ano: 2010 Km Atual: 40100 Combustível: Gasolina

Marca: Volkswagen Cor: Prata Valor Pago: 35000

Últ. Troca de Pneu: 13/03/2016 Mês Pgto IPVA: 1

Salvar Excluir Fechar

As informações do veículo são separadas em 3 tipos: Dados, Histórico e Agenda. A opção Dados apresentada na Figura 15 vem definida por padrão e mostra as informações básicas do veículo. A opção Histórico, apresentada na Figura 16, lista os últimos 15 registros em que o veículo foi utilizado. Por fim, a opção Agenda, apresentada na Figura 17, mostra os últimos 15 agendamentos do veículo. Todos os cadastros do sistema utilizam o mesmo padrão do cadastro de veículo.

Figura 16 - Tela de visualização do histórico do veículo

Visualizar Veículo

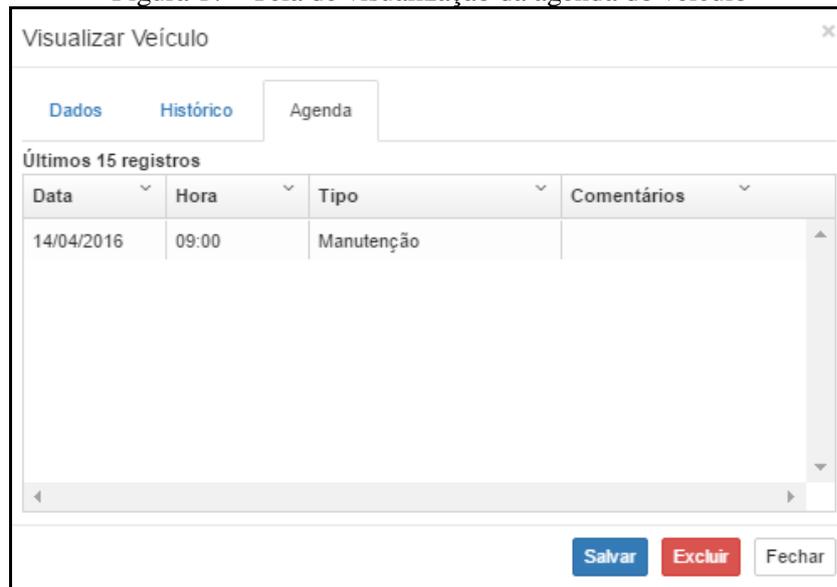
Dados Histórico Agenda

Últimos 15 registros

Tipo	Data
Cadastrou Agenda	14/04/2016 21:48
Cadastrou Multa	02/05/2016 23:24

Salvar Excluir Fechar

Figura 17 - Tela de visualização da agenda do veículo



Data	Hora	Tipo	Comentários
14/04/2016	09:00	Manutenção	

Ao acessar o menu Controle de Frota, (Figura 12), é apresentada uma tela, mostrada na Figura 19, com opções para controle de Agenda, Abastecimento, Manutenção, Km e Multa. A Agenda já aparece selecionada por padrão, listando os compromissos agendados dos veículos cadastrados. Também é possível cadastrar um novo agendamento. O mapa de navegação de acesso ao Controle de Frota apresentado na Figura 18 mostra o fluxo da Agenda até o servidor. A <<screen>> Controle de Frota é separada em 5 opções de controle, cada opção está dentro de uma <<screen compartment>> que por sua vez possui um formulário de cadastro (<<input form>>) e uma lista de itens cadastrados. Os fluxos do Abastecimento, Manutenção, Km e Multa são simplificados por serem semelhantes ao fluxo da Agenda.

Figura 18 - Mapa de navegação do controle da frota

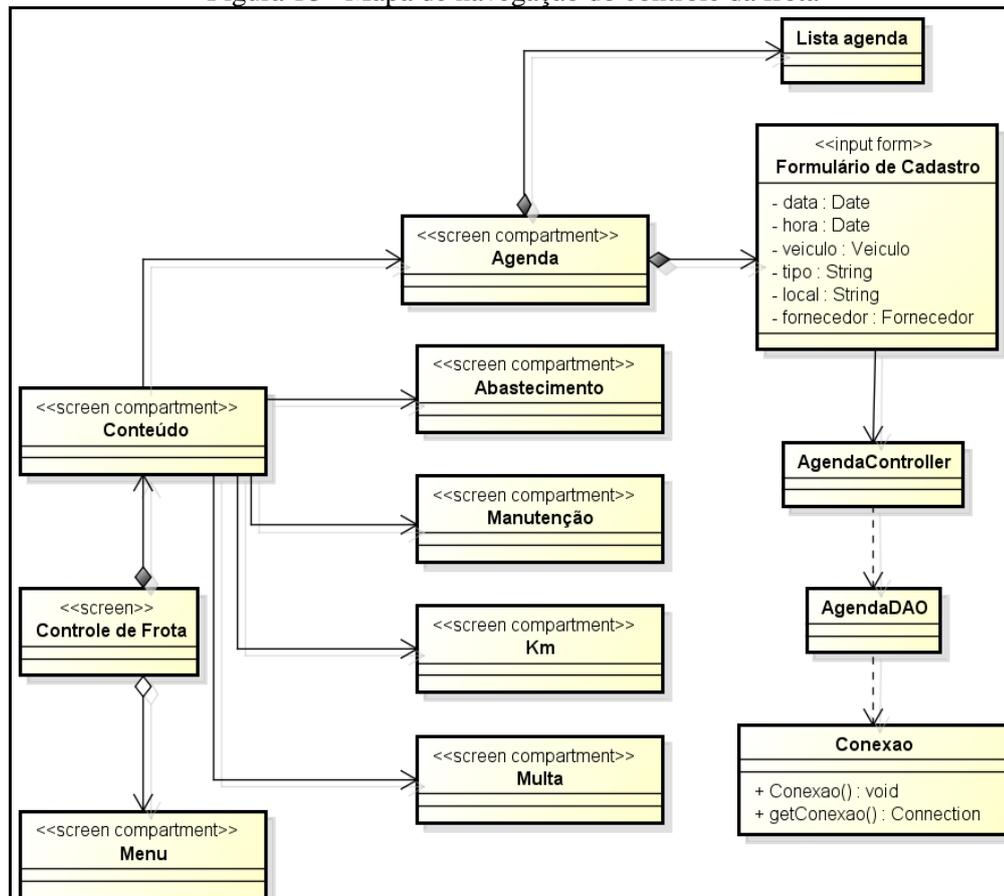
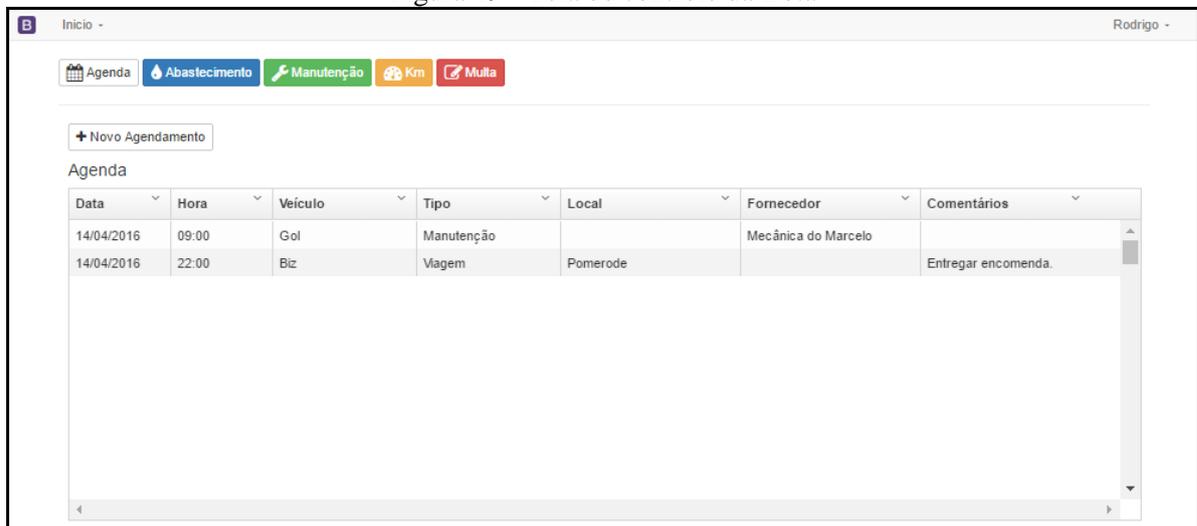


Figura 19 - Tela de controle da frota



A Figura 20 apresenta a tela de cadastro de agenda. Nesta tela é necessário preencher um formulário com data e hora do agendamento, veículo, tipo da agenda (manutenção, abastecimento e viagem.), local, fornecedor e descrição.

Figura 20 - Tela de cadastro de agenda



Novo Agendamento

Data:  Hora:  Veículo:

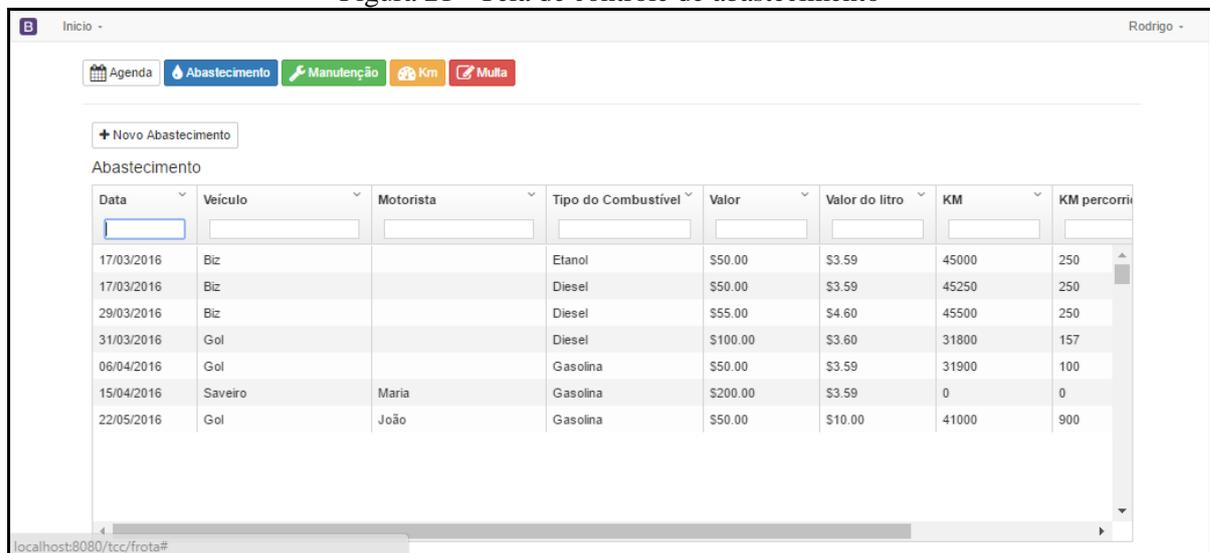
Tipo:  Local:  Fornecedor:

Descrição:

Salvar Fechar

A Figura 21 apresenta a tela de controle de abastecimento. Nessa tela há uma tabela com todos os abastecimentos cadastrados e um botão para cadastro. No cabeçalho de cada coluna dessa tabela, existe um campo para o usuário filtrar os dados cadastrados. No cadastro de abastecimento mostrado na Figura 22, é necessário preencher a data, veículo, motorista, fornecedor, tipo do combustível (gasolina, etanol ou diesel), valor, preço do litro, forma de pagamento (dinheiro, débito, crédito, cheque) e a quilometragem atual, a partir desses dados o formulário irá calcular a distância percorrida desde o último abastecimento e a quantia abastecida em litros.

Figura 21 - Tela de controle de abastecimento



Início - Rodrigo -

Agenda Abastecimento Manutenção Km Multa

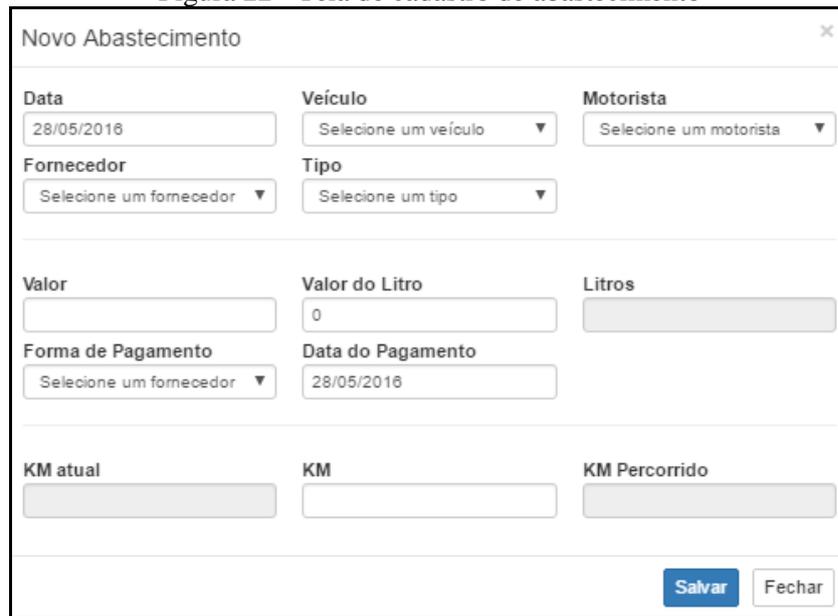
+ Novo Abastecimento

Abastecimento

Data	Veículo	Motorista	Tipo do Combustível	Valor	Valor do litro	KM	KM percorri
<input type="text"/>							
17/03/2016	Biz		Etanol	\$50.00	\$3.59	45000	250
17/03/2016	Biz		Diesel	\$50.00	\$3.59	45250	250
29/03/2016	Biz		Diesel	\$55.00	\$4.60	45500	250
31/03/2016	Gol		Diesel	\$100.00	\$3.60	31800	157
06/04/2016	Gol		Gasolina	\$50.00	\$3.59	31900	100
15/04/2016	Saveiro	Maria	Gasolina	\$200.00	\$3.59	0	0
22/05/2016	Gol	João	Gasolina	\$50.00	\$10.00	41000	900

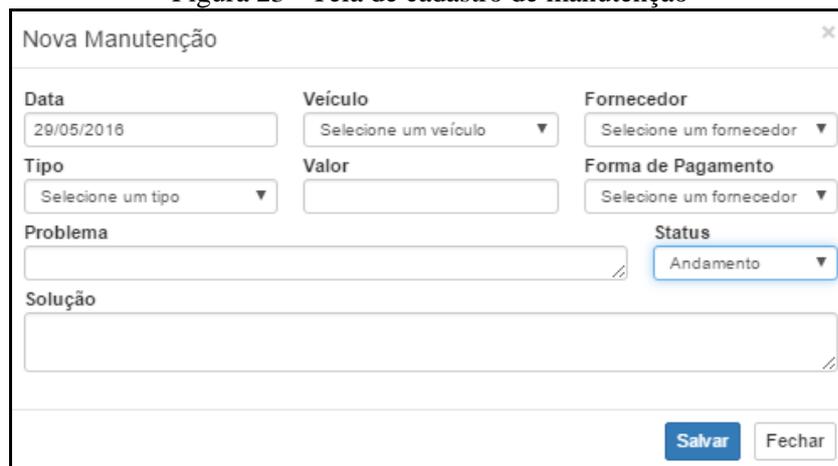
localhost:8080/tcc/frota#

Figura 22 - Tela de cadastro de abastecimento



As telas de controle de manutenção, quilometragem e multa apresentam tabelas com os itens cadastrados e botões para cadastro de novos itens. No cabeçalho de cada coluna das tabelas, existe um campo para o usuário filtrar os dados cadastrados. As telas são similares ao do controle de abastecimento mostrado na Figura 21. O formulário para cadastro de manutenção apresentado na Figura 23, requer os seguintes dados: data, veículo, fornecedor, tipo (preventiva ou corretiva), valor, forma de pagamento, problema, status (andamento ou finalizado) e a descrição da solução do problema.

Figura 23 - Tela de cadastro de manutenção



A tela para cadastro de quilometragem apresentado na Figura 24, requer os seguintes dados: veículo, data e hora da saída e chegada, motorista, quilometragem atual e destino, o campo de quilometragem anterior é automaticamente preenchido com a quilometragem atual do veículo selecionado.

Figura 24 - Tela de cadastro de quilometragem

A tela para cadastro de multa apresentado na Figura 25, requer os seguintes dados: data, hora, veículo, data do vencimento, pontuação, valor, motorista e descrição.

Figura 25 - Tela de cadastro de multa

Ao acessar o menu Relatórios, mostrado na Figura 12, é apresentada uma tela mostrada na Figura 26, com opções para geração de relatório de Consumo Médio, Distância Percorrida, Viabilidade, Despesas e Impacto Financeiro. Para gerar um relatório é necessário informar a data inicial, data final e definir um veículo ou selecionar a opção de Todos.

Figura 26 - Tela de geração de relatório

Ao gerar o relatório de consumo médio (Figura 27), são mostrados a distância percorrida e o combustível utilizado. Além disso, é calculado o consumo médio do período,

dividindo a distância percorrida pelo combustível utilizado. No rodapé do relatório é calculado o consumo médio de todos os veículos, caso seja selecionado a opção `Todos`.

Figura 27 - Relatório de consumo médio

 <b>Sistema para Gestão de Frotas</b> Universidade Regional de Blumenau rodrigopereira@outlook.com		Consumo Médio
Data Inicial:	31/05/2016	
Data Final:	30/06/2016	
<b>Veículo: Biz</b>		
Distância percorrida no período:	700,00	
Combustível utilizado no período:	10,85	
Consumo Médio no período:	64,49	
<hr/>		
<b>Veículo: Gol</b>		
Distância percorrida no período:	600,00	
Combustível utilizado no período:	50,14	
Consumo Médio no período:	11,97	
<hr/>		
<b>Veículo: Uno</b>		
Distância percorrida no período:	540,00	
Combustível utilizado no período:	41,78	
Consumo Médio no período:	12,92	
<hr/>		
<b>Distância Total Percorrida:</b>	1840,00	
<b>Combustível Total</b>	102,78	
<b>Consumo Médio Total:</b>	29,79	

Ao gerar o relatório de distância percorrida mostrado na Figura 28, é apresentado os últimos 5 destinos de viagem do veículo. Em cada registro é apontado a quilometragem de chegada, data e hora de saída e chegada, destino e quilometragem percorrida pelo veículo de forma individual. Ao final, caso a opção `todos` esteja selecionada é somado a quilometragem percorrida de todos os veículos no período.

Figura 28 - Relatório de distância percorrida

SGF		Sistema para Gestão de Frotas		Distância Percorrida		
Universidade Regional de Blumenau		rodrigopereira@outlook.com				
Data Inicial:	31/05/2016					
Data Final:	30/06/2016					
<b>Veículo: Biz</b>						
Motorista	Km de Chegada	Data e Hora de Saída		Data e Hora de Chegada		Destino
João	47500.0	31/05/2016	19.00	31/05/2016	20.00	Mercado
Pedro	47600.0	01/06/2016	22.54	02/06/2016	19.00	Cliente Marcelo
Maria	48000.0	04/06/2016	19.00	05/06/2016	18.00	Fornecedor
<b>Km total do veículo:</b>		700.0				
<b>Veículo: Gol</b>						
Motorista	Km de Chegada	Data e Hora de Saída		Data e Hora de Chegada		Destino
João	41600.0	01/06/2016	19.00	01/06/2016	20.00	Casa da Maria
<b>Km total do veículo:</b>		600.0				
<b>Veículo: Uno</b>						
Motorista	Km de Chegada	Data e Hora de Saída		Data e Hora de Chegada		Destino
Pedro	23540.0	01/06/2016	19.00	01/06/2016	19.00	Furb
<b>Km total do veículo:</b>		540.0				
<b>Km total da frota:</b>		1840.0				

O opção de Viabilidade apresenta a tela mostrada na Figura 29, em que o usuário informa a quilometragem diária do veículo a ser substituído, preço do combustível, consumo médio e o consumo médio do veículo a ser adquirido. A partir desses dados, é calculada a viabilidade da compra mostrando um aviso de Economia (Figura 30), ou Despesas (Figura 31). Também existe a opção N° de Veículos, em que o usuário informa a quantia de veículos a ser adquirido. O resultado do cálculo será multiplicado por esse número.

Figura 29 - Tela de viabilidade de veículo

**Análise de Viabilidade** x

N° de Veículos:

Quilometragem Diária:

Preço do Combustível:

Consumo Médio atual:

Consumo Médio do Veículo Novo:

Figura 30 - Aviso de economia

**Economia!**

Em 1 mês você terá a economia de: \$131.63

Em 1 ano você terá a economia de: \$1,579.60

O aviso de despesas, mostrado na Figura 31, apresenta as despesas mensais e anuais da possível compra. A última opção da tela de Relatório, mostrada na Figura 26, nomeada como Impacto Financeiro, apresenta uma tela (Figura 32), em que o usuário informa o faturamento anual da empresa. A partir deste valor, é calculado o percentual que as despesas com a frota representa para a empresa. Também é mostrado o total de despesas até a data e um gráfico separado em 3 opções: Total de Abastecimento, Total de Manutenção e Total de Multa.

Figura 31 - Aviso de despesas

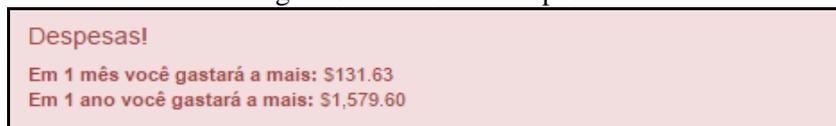
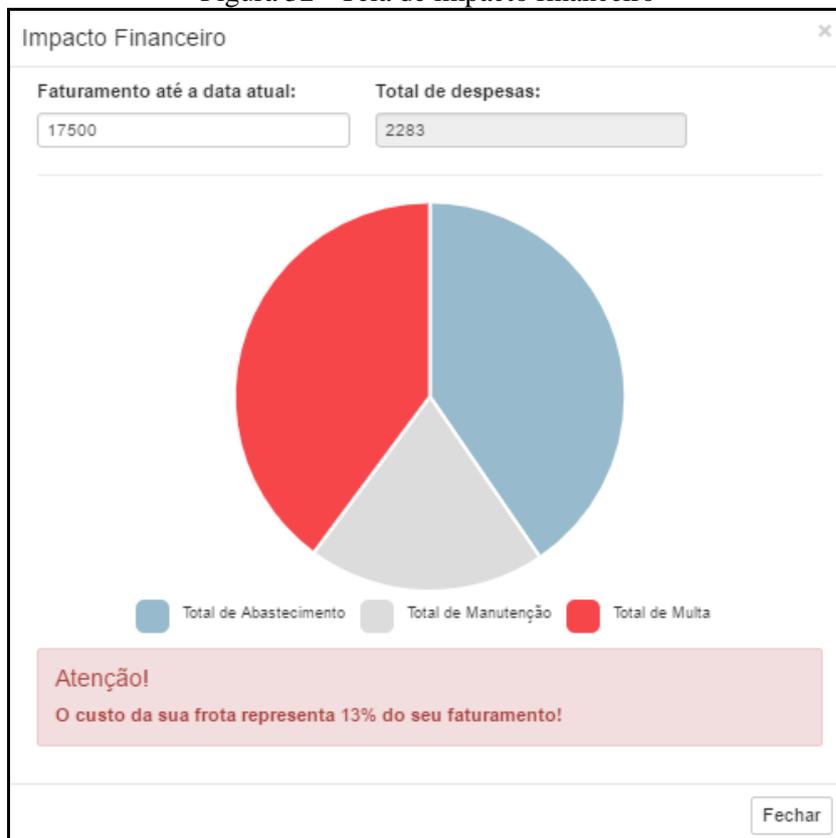


Figura 32 - Tela de impacto financeiro



### 3.4 RESULTADOS E DISCUSSÕES

O presente trabalho apresentou o desenvolvimento de um sistema web com objetivo de auxiliar as empresas a realizarem a gestão de sua frota e pessoas físicas a controlarem seus veículos de forma gratuita. As instruções para implementação do sistema estão disponíveis em um repositório online, acessível em <https://github.com/rodrigopereira/tcc>. Dessa forma, o objetivo geral e o quarto objetivo específico do trabalho foram alcançados.

O sistema apresentado neste trabalho foi desenvolvido na plataforma web, podendo ser acessado de qualquer dispositivo com acesso à Internet. É dividido em 3 módulos, entre os quais: cadastros, controle de frota e relatórios. O módulo de cadastros permite ao usuário cadastrar todos os motoristas, veículos e fornecedores que serão utilizados no módulo de controle de Frota, este módulo conta com opções para registros dos agendamentos, manutenções, abastecimentos, quilometragem e multa dos veículos. Desse modo, o primeiro e segundo objetivos específicos do trabalho, que dizem respeito ao controle da frota e dos motoristas e a facilitação do acompanhamento das operações desses, são atingidos. O módulo de relatórios é responsável por agrupar os dados cadastrados e reporta-los ao usuário através de uma informação concisa, o sistema disponibiliza 3 relatórios: consumo médio, distância percorrida e despesas.

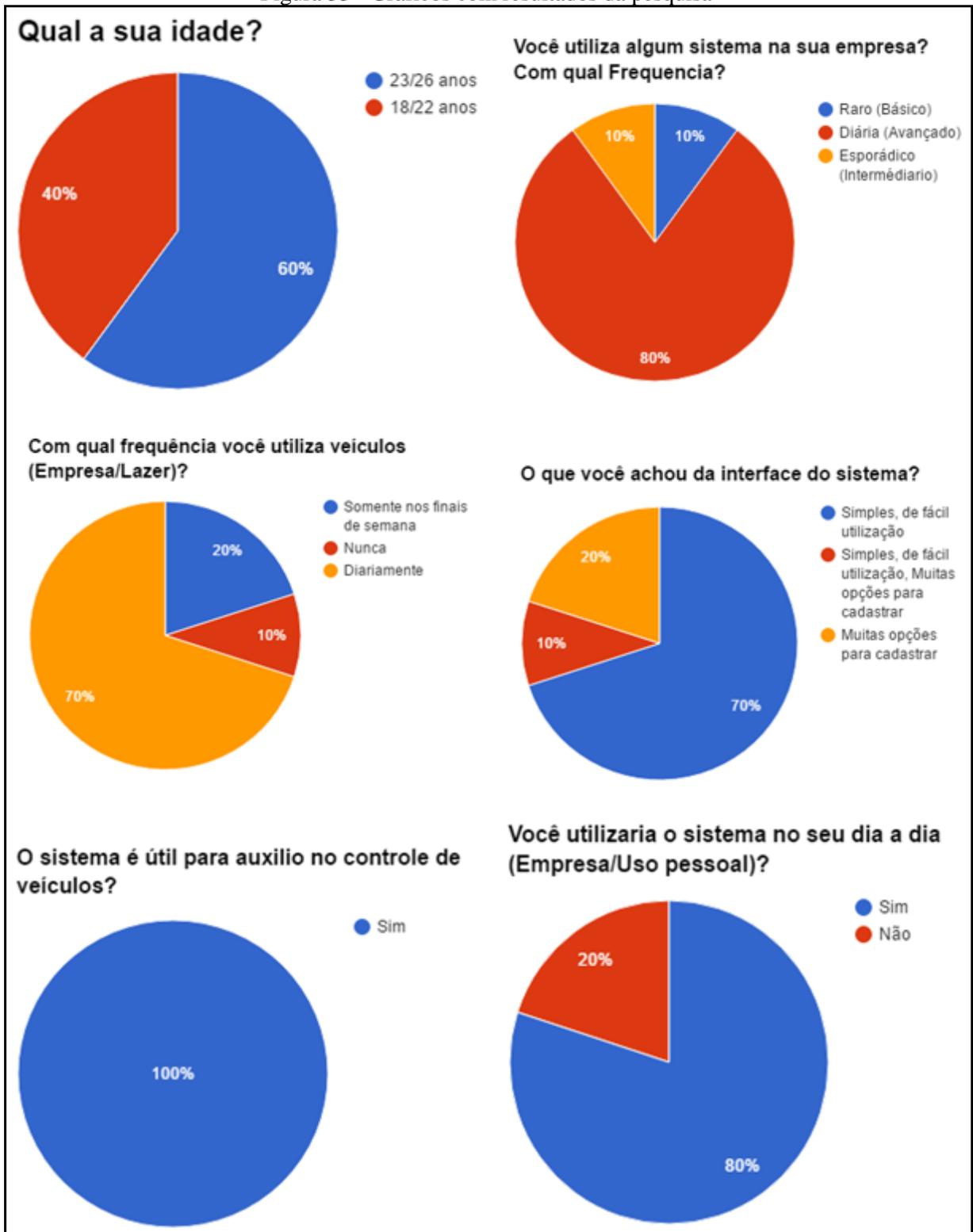
Em comparação com os trabalhos correlatos desenvolvidos por Sorroche (2010) e Hoffman (2010), o sistema desenvolvido neste trabalho possibilita a implementação do mesmo por qualquer empresa, abrangendo a maioria das funções desenvolvidas no projeto dos dois autores. Pode-se citar como um dos diferenciais, a geração de relatórios de consumo médio, viabilidade na aquisição de novos veículos e o impacto financeiro da sobre o faturamento da empresa. Conseqüentemente contempla o terceiro objetivo específico do trabalho, que é fornecer dados elaborados para auxiliar a tomada de decisões. O uso do sistema proposto neste trabalho é livre. Essa característica é o grande diferencial em comparação ao sistema Sofit, desenvolvido por Amaral et al(2016). O Quadro 9 apresenta um comparativo entre o sistema desenvolvido e os trabalhos correlatos.

Quadro 9 - Comparação com os trabalhos correlatos

Característica	Este projeto	Trabalho de Sorroche	Trabalho de Hoffman	Sofit
Livre	Sim	Não informado	Não informado	Não
Controle de Multas	Sim	Não	Não	Sim
Viabilidade de Veículo	Sim	Não	Não	Sim
Uso pessoal	Sim	Não	Não	Parcial

Foi realizado uma pesquisa em que o sistema foi testado por 10 participantes. Após os testes foi aplicado um questionário com 6 perguntas com objetivo de analisar o perfil do participante e captar a experiência do mesmo com o sistema. O questionário é apresentado no Apêndice B. Na Figura 33 são apresentados os gráficos com o resultado da pesquisa.

Figura 33 - Gráficos com resultados da pesquisa



Esse estudo de caso foi organizado da seguinte forma:

- a pesquisa foi realizada para verificar se há dificuldades no uso do sistema quanto a sua interface;
- dez pessoas participaram da pesquisa com faixa etária de 17 a 30 anos. Os

- participantes possuem diferentes níveis de conhecimento sobre gestão de frotas;
- c) os participantes acessaram individualmente o sistema a partir de um computador servidor e tiveram 2 dias para testar todas as funcionalidades. Em seguida, os participantes responderam a um questionário, apresentado no Apêndice B, com questões sobre a interface e utilidade do sistema.

Baseado nos dados apresentados na Figura 33, nota-se que a idade média dos participantes varia entre 18 a 26 anos, sendo que 60% (6 participantes) possuem 23 anos ou mais. Também se percebe que 80% dos avaliadores utilizam diariamente algum sistema em sua empresa, isso demonstra experiência na utilização de sistemas. Foi constatado que 90% dos participantes tem contato direto com algum tipo de veículo, sendo que 70% utiliza-os diariamente. A interface do sistema mostrou-se amigável, pois nenhum participante teve dificuldades em acessar as páginas e realizar as operações. Entre os 10 participantes, 8 declaram que utilizariam o sistema, sendo para uso pessoal ou na sua empresa, sendo que todos os participantes julgaram que o sistema desenvolvido é útil na realização de um controle de frotas.

## 4 CONCLUSÕES

Com objetivo de auxiliar os usuários na gestão de suas frotas, foi proposto neste trabalho um sistema *web* que possibilita o controle de operações relacionadas aos agendamentos, manutenções, abastecimentos, viagens e multas dos veículos. Baseando-se nesses dados, foi possível desenvolver um módulo responsável pela emissão de diversos relatórios, contendo informações sobre a situação da frota e visando prover informações úteis aos responsáveis pela gestão dos veículos.

O ambiente de desenvolvimento mostrou-se capaz de trabalhar em harmonia com as diversas tecnologias utilizadas neste trabalho. A integração nativa com o Tomcat foi fundamental para execução do sistema, possibilitando a criação de *builds* automáticos sem a necessidade de fazer um *deploy* manual a cada nova versão. O Bootstrap e AngularJS foram capazes de satisfazer todas as necessidades no desenvolvimento do sistema, pois ambos contam com largo suporte da comunidade e com grande quantidade de bibliotecas abertas disponíveis para cada necessidade.

Por ser um sistema web baseado em tecnologias responsivas, o mesmo pode ser acessado a partir de qualquer dispositivo com acesso à Internet. Isso garante uma vantagem ao sistema, pois as operações podem ser registradas no momento em que ocorrem, ficando instantaneamente disponíveis aos gestores.

A pesquisa e implementação permitiu ainda a oportunidade de aprimorar o conhecimento nas tecnologias e arquiteturas apresentadas e entender o processo de gestão de frotas de maneira mais ampla.

### 4.1 EXTENSÕES

Como sugestões de melhora ao sistema desenvolvido tem-se:

- a) disponibilizar um módulo para rastreamento dos veículos;
- b) disponibilizar um módulo para controle de pneus;
- c) criar relatórios relacionados ao motorista;
- d) disponibilizar uma opção para envio dos relatórios ao e-mail dos usuários;
- e) permitir a consulta de multas diretamente ao órgão responsável;
- f) emitir mensagens com boas práticas na gestão de frotas;
- g) disponibilizar um módulo para auditoria do sistema..

## REFERÊNCIAS

- AMARAL, G.; et al. **A empresa**. [S. l.], 2016. Disponível em: < <http://www.sofit4.com.br/pt/empresa/sofit-software-para-gestao-e-manutencao-de-frota/>>. Acesso em: 24 maio 2016.
- ANFAVEA, ASSOCIAÇÃO NACIONAL DOS FABRICANTES DE VEÍCULOS AUTOMOTORES. **Anfavea apresenta estudo sobre o mercado brasileiro de autoveículos em 2034**. São Paulo: Anfavea, 2014. Disponível em: < [http://www.anfavea.com.br/docs/06%2011%2014\\_PressRelease\\_Estudo\\_2034.pdf/](http://www.anfavea.com.br/docs/06%2011%2014_PressRelease_Estudo_2034.pdf/)>. Acesso em: 28 fev. 2016.
- ANSELMO, Fernando. **Spring framework**. Espirito Santo: Brasília: X25, 2016. Disponível em: < <http://www.fernandoans.site50.net/curso/curso05/Curso/01-SpringFramework.pdf/>>. Acesso em: 21 abr. 2016.
- BAPTISTELLA, Adriano José. **Abordando a arquitetura mvc, e design patterns: observer, composite, strategy**. [S. l.], 2016. Disponível em: < <http://www.linhadecodigo.com.br/artigo/2367/abordando-a-arquitetura-mvc-e-design-patterns-observer-composite-strategy.aspx/>>. Acesso em: 31 maio 2016.
- BRITO, Edivaldo. **Java: entenda para que serve o software e os problemas da sua ausência**. [S. l.], 2014. Disponível em: < <http://www.techtudo.com.br/dicas-e-tutoriais/noticia/2014/11/java-entenda-para-que-serve-o-software-e-os-problemas-da-sua-ausencia.html/>>. Acesso em: 24 fev. 2016.
- CAELUM. **O que é java**. [S. l.], 2015. Disponível em: < <http://www.caelum.com.br/apostila-java-orientacao-objetos/o-que-e-java/>>. Acesso em: 05 jul. 2016.
- CIPOLI, Pedro. **O que é java, jre, jvm e jdk?**. [S. l.], 2015. Disponível em: < <http://corporate.canaltech.com.br/o-que-e-programacao/O-que-e-Java-JRE-JVM-e-JDK/>>. Acesso em: 24 fev. 2016.
- CLEMENTE, Quebo kenge. **Gestão de frotas de veículos rodoviários**. Lisboa: Universidade Técnica de Lisboa, 2008. Disponível em: < <https://fenix.tecnico.ulisboa.pt/downloadFile/3779572158698/16--view.pdf/>>. Acesso em: 05 jul 2016.
- CNMCUT. **Industria automobilista movimentada r\$150 bilhões por ano no Brasil**. São Bernardo do Campo: CNMCUT, 2010. Disponível em: < <http://www.cnmcut.org.br/conteudo/industria-automobilistica-movimentada-r-150-bilhoes-por-ano-no-brasil>>. Acesso em: 05 jul. 2016.
- COELHO, Leandro Callegari. **Gestão de frotas de veículos – logística descomplicada**. São Paulo, 2011. Disponível em: < <http://www.logisticadescomplicada.com/gestao-de-frotas-de-veiculos/>>. Acesso em: 05 jul. 2016.
- COLOMBO, Silas. **Conheça os 10 erros mais comuns na gestão de manutenção de frota**. [S. l.], 2012. Disponível em: < <http://www.transportabrasil.com.br/2012/07/conheca-os-10-erros-mais-comuns-na-gestao-de-manutencao-de-frota/>>. Acesso em: 15 maio 2015.
- COUTINHO, Paulo César. **Primeiros passos com serviços rest**. [S. l.], 2015. Disponível em: < <http://www.devmedia.com.br/primeiros-passos-com-os-servicos-rest/28912#/>>. Acesso em: 1 maio 2016.

DARIO, Marcos. **Práticas, indicadores da manutenção e custos na gestão de pneus: estudo em uma empresa de transportes**. Piracicaba: Universidade Metodista de Piracicaba, 2012. Disponível em: < [https://www.unimep.br/phpg/bibdig/pdfs/docs/11102013\\_143049\\_marcos\\_dario.pdf](https://www.unimep.br/phpg/bibdig/pdfs/docs/11102013_143049_marcos_dario.pdf) >. Acesso em: 05 jul. 2016.

D'ÁVILLA, Célio Antônio. **Curso técnico em transportes**. CEFETES, 2006. Disponível em: < [ftp://ftp.ifes.edu.br/cursos/Transportes/CelioDavilla/Assuntos%20Diversos/Gest%E3o%20de%20frotas/GEST%C3O%20DE%20FROTAS%202006\\_1\\_C%C9LIOf.pdf/](ftp://ftp.ifes.edu.br/cursos/Transportes/CelioDavilla/Assuntos%20Diversos/Gest%E3o%20de%20frotas/GEST%C3O%20DE%20FROTAS%202006_1_C%C9LIOf.pdf/)>. Acesso em: 05 jul. 2016.

DClick. **O que é o maven, e seus primeiros passos com a ferramenta**. [S. l.], 2010. Disponível em: < <http://www.dclick.com.br/2010/09/15/o-que-e-o-maven-e-seus-primeiros-passos-com-a-ferramenta/>>. Acesso em: 10 abr. 2016.

GRILLO, Filipi Del Nero; FORTES, Renata Pontin de Mattos. **Aprendendo javaScript**. São Paulo: Universidade de São Paulo, 2008. Disponível em: < [http://www.icmc.usp.br/CMS/Arquivos/arquivos\\_enviados/BIBLIOTECA\\_113\\_ND\\_72.pdf/](http://www.icmc.usp.br/CMS/Arquivos/arquivos_enviados/BIBLIOTECA_113_ND_72.pdf/)>. Acesso em: 05 jul 2016.

HOFFMANN, Ricardo. **Sistema para reserva de veículos de uma cooperativa de saúde**. Blumenau: Universidade Regional de Blumenau, 2010. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) - Curso de Sistemas de Informação, Universidade Regional de Blumenau, Blumenau. Disponível em: < <http://dsc.inf.furb.br/arquivos/tccs/monografias/TCC2010-2-17-VF-RicardoHoffmann.pdf/>>. Acesso em: 24 maio 2016.

LIMA, Jean Carlos Rosário. **Web services (soap x rest)**. São Paulo: Faculdade de Tecnologia de São Paulo, 2012. Disponível em: < <http://www.fatecsp.br/dti/tcc/tcc00056.pdf/>>. Acesso em: 24 maio 2016.

NAKAMASHI, Diego. **Introdução ao AngularJS**. [S. l.], 2013. Disponível em: < <http://vizir.com.br/2013/09/introducao-ao-angularjs/>>. Acesso em: 20 abr. 2016.

ROTA EXATA. **Infográfico: 3 passos para gestão dos veículos**. [S. l.], 2015. Disponível em: < <https://www.rotaexata.com.br/blog/3-passos-para-gestao-dos-veiculos/>>. Acesso em: 05 jul 2016.

SOFIT. **Sofit completa 5 anos**. [S. l.], 2015. Disponível em: <http://www.sofit4.com.br/blog/2015/10/sofit-completa-5-anos/>>. Acesso em: 16 jun. 2016.

SORROCHE, Ricardo. **Sistema de informações para controle de frota de veículos**. Blumenau: Universidade Regional de Blumenau, 2010. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) - Curso de Sistemas de Informação, Universidade Regional de Blumenau, Blumenau. Disponível em: < <http://dsc.inf.furb.br/arquivos/tccs/monografias/TCC2010-1-20-VF-RicardoSorroche.pdf/>>. Acesso em: 24 maio 2016.

UTTERBACK, Benjamin. **O que é bootstrap – verdades e mitos: parte 1 de 2**. [S. l.], 2014. Disponível em: < <https://www.prestashop.com/blog/pt/2014/03/06/o-que-e-o-bootstrap-verdades-e-mitos-parte-1-de-2/>>. Acesso em: 20 abr. 2016.

VALENTE, A.M.; PASSAGLIA, E.; NOVAES, A.G. **Gerenciamento de transporte e frotas**. São Paulo: Editora Pioneira Thomson Learning, 1997.

**W3C. Cartilha de acessibilidade na web.** [S. l.], 2013. Disponível em: <<http://www.w3c.br/pub/Materiais/PublicacoesW3C/cartilha-w3cbr-acessibilidade-web-fasciculo-I.html>>. Acesso em: 05 jul. 2016.

\_\_\_\_\_. **Css.** [S. l.], 2016. Disponível em: <<http://www.w3c.br/pub/Cursos/CursoCSS3/css-web.pdf>>. Acesso em: 05 jul. 2016.

\_\_\_\_\_. **Html5.** [S. l.], 2016. Disponível em: <<http://www.w3c.br/pub/Cursos/CursoHTML5/html5-web.pdf>>. Acesso em: 05 jul. 2016.

## APÊNDICE A – Descrição dos Casos de Uso

Este Apêndice apresenta a descrição dos principais casos de uso. No Quadro 10 apresenta-se o caso de uso UC03 – Manter veículos.

Quadro 10 - Detalhamento do caso de uso UC03 – Manter veículos

UC03 – Manter veículos	
<b>Ator</b>	Usuário
<b>Objetivo</b>	Permitir ao usuário, cadastrar, consultar, editar e excluir veículos.
<b>Pré-condições</b>	Em caso de edição, consulta e exclusão é necessário que a veículo já tenha sido cadastrado. Em caso de cadastro é necessário que todos os dados sejam preenchidos.
<b>Cenário principal</b>	<ol style="list-style-type: none"> <li>1. O usuário seleciona a opção “Cadastros”;</li> <li>2. O usuário seleciona a opção “Veículos”;</li> <li>3. Sistema apresenta tela para funções de veículo;</li> <li>4. O usuário opta por cadastrar um veículo;</li> <li>5. O usuário informa todos os dados do veículo;</li> <li>6. O usuário seleciona a opção “Cadastrar”;</li> <li>7. Sistema cadastra o veículo.</li> </ol>
<b>Cenário alternativo</b>	<p>No passo 4, o usuário opta por consultar um veículo.</p> <ol style="list-style-type: none"> <li>4.1 O usuário preenche os filtros;</li> <li>4.2 O usuário seleciona o veículo;</li> <li>4.3 Sistema apresenta as informações do veículo.</li> </ol>
<b>Cenário alternativo</b>	<p>Após o passo 4.3, o usuário opta por editar os dados do veículo.</p> <ol style="list-style-type: none"> <li>4.3.1 O usuário altera os dados do veículo;</li> <li>4.3.2 O usuário seleciona a opção “Salvar”;</li> <li>4.3.3 Sistema salva os dados do veículo.</li> </ol>
<b>Cenário alternativo</b>	<p>Após o passo 4.3, o usuário opta por excluir o veículo.</p> <ol style="list-style-type: none"> <li>4.3.1 O usuário seleciona a opção “Excluir”;</li> <li>4.3.2 Sistema remove o veículo.</li> </ol>
<b>Pós-condições</b>	Um veículo foi cadastrado, visualizado, editado ou excluído.

No Quadro 11 apresenta-se o caso de uso UC05 – Manter manutenções.

Quadro 11 - Detalhamento do caso de uso UC05 - Manter manutenções

UC05 – Manter manutenções	
<b>Ator</b>	Usuário, Motorista
<b>Objetivo</b>	Permitir ao usuário e Motorista, cadastrar, consultar, editar e excluir manutenções.
<b>Pré-condições</b>	Em caso de edição, consulta e exclusão é necessário que a manutenção já tenha sido cadastrado. Em caso de cadastro é necessário que todos os dados sejam preenchidos.
<b>Cenário principal</b>	<ol style="list-style-type: none"> <li>1. O usuário seleciona a opção “Controle de frota”;</li> <li>2. O usuário seleciona a opção “Manutenção”;</li> <li>3. Sistema apresenta tela para funções de manutenção;</li> <li>4. O usuário opta por cadastrar uma manutenção;</li> <li>5. O usuário seleciona a opção “Nova manutenção”;</li> <li>6. O usuário informa todos os dados da manutenção;</li> <li>7. O usuário seleciona a opção “Cadastrar”;</li> <li>8. Sistema cadastra a manutenção.</li> </ol>
<b>Cenário alternativo</b>	<p>No passo 4, o usuário opta por consultar uma manutenção.</p> <ol style="list-style-type: none"> <li>4.1 O usuário preenche os filtros;</li> <li>4.2 O usuário seleciona a manutenção;</li> <li>4.3 Sistema apresenta as informações da manutenção.</li> </ol>
<b>Cenário alternativo</b>	<p>Após o passo 4.3, o usuário opta por editar os dados da manutenção.</p> <ol style="list-style-type: none"> <li>4.3.1 O usuário altera os dados da manutenção;</li> <li>4.3.2 O usuário seleciona a opção “Salvar”;</li> <li>4.3.3 Sistema salva os dados da manutenção.</li> </ol>
<b>Cenário alternativo</b>	<p>Após o passo 4.3, o usuário opta por excluir a manutenção.</p> <ol style="list-style-type: none"> <li>4.3.1 O usuário seleciona a opção “Excluir”;</li> <li>4.3.2 Sistema remove a manutenção.</li> </ol>
<b>Pós-condições</b>	Uma manutenção foi cadastrada, visualizada, editada ou excluída.

No Quadro 12 apresenta-se o caso de uso UC09 - Emitir relatório de viabilidade do veículo.

Quadro 12 - Detalhamento do caso de uso UC09 - Emitir relatório de viabilidade do veículo

UC09 – Emitir relatório de viabilidade do veículo	
<b>Ator</b>	Administrador
<b>Objetivo</b>	Permitir ao administrador emitir um relatório de viabilidade de compra de veículo.
<b>Cenário principal</b>	<ol style="list-style-type: none"> <li>1. O administrador seleciona a opção “Relatórios”;</li> <li>2. O administrador seleciona a opção “Viabilidade”;</li> <li>3. Sistema apresenta tela para viabilidade;</li> <li>4. O administrador informa a quilometragem diária;</li> <li>5. O administrador informa o valor do combustível;</li> <li>6. O administrador informa o consumo médio do novo veículo;</li> <li>7. O administrador informa o consumo médio do veículo a ser substituído;</li> <li>8. Sistema mostra o resultado da viabilidade.</li> </ol>
<b>Cenário alternativo</b>	<p>No passo 8, caso a viabilidade seja negativa.</p> <ol style="list-style-type: none"> <li>8.1 Sistema mostra uma mensagem em vermelho com os gastos prováveis.</li> </ol>
<b>Cenário alternativo</b>	<p>No passo 8, caso a viabilidade seja positiva.</p> <ol style="list-style-type: none"> <li>8.1 Sistema mostra uma mensagem em verde com a economia provável.</li> </ol>
<b>Pós-condições</b>	Foi analisado a viabilidade de um veículo.

## APÊNDICE B – Questionário de avaliação

O Quadro 13 mostra o questionário utilizado pelos usuários para a avaliação do sistema.

Quadro 13 - Questionário de avaliação do sistema

- 1) Qual a sua idade?
  - a) 18/22 anos
  - b) 23/26 anos
  - c) 27 ou mais
  
- 2) Você utiliza algum sistema na sua empresa? Com qual Frequência?
  - a) Nenhuma
  - b) Raro (Básico)
  - c) Esporádico (Intermediário)
  - d) Diariamente (Avançado)
  
- 3) Com qual frequência você utiliza veículos (Empresa/Lazer)?
  - a) Diariamente
  - b) As vezes no trabalho
  - c) Somente nos finais de semana
  - d) Nunca
  
- 4) O que você achou da interface do sistema?
  - a) Simples, de fácil utilização
  - b) Muitas opções para cadastrar
  - c) Difícil de encontrar as opções
  
- 5) Você utilizaria o sistema no seu dia a dia (Empresa/Use pessoal)?
  - a) Sim
  - b) Não
  
- 6) O sistema útil para o auxílio no controle de veículos?
  - a) Sim
  - b) Não
  
- 7) Deixe sua opinião sobre o sistema: