



**UFOP**

Universidade Federal  
de Ouro Preto

**Universidade Federal de Ouro Preto  
Instituto de Ciências Exatas e Aplicadas  
Departamento de Computação e Sistemas**

**GarageParts: Sistema de Controle de  
Consumo, Deslocamento e  
Manutenções de Veículos Pessoais**

**José Wilson da Silva Júnior**

**TRABALHO DE  
CONCLUSÃO DE CURSO**

ORIENTAÇÃO:  
Bruno Rabello Monteiro

**Dezembro, 2019  
João Monlevade–MG**

**José Wilson da Silva Júnior**

**GarageParts: Sistema de Controle de Consumo,  
Deslocamento e Manutenções de Veículos  
Pessoais**

Orientador: Bruno Rabello Monteiro

Monografia apresentada ao curso de Sistemas de Informação do Instituto de Ciências Exatas e Aplicadas, da Universidade Federal de Ouro Preto, como requisito parcial para aprovação na Disciplina “Trabalho de Conclusão de Curso II”.

**Universidade Federal de Ouro Preto**

**João Monlevade**

**Dezembro de 2019**

S586g

Silva Júnior, José Wilson da.

GarageParts [manuscrito]: sistema de controle de consumo, deslocamento e manutenções de veículos pessoais / José Wilson da Silva Júnior. - 2019.

44f.: il.: color; tabs.

Orientador: Prof. MSc. Bruno Rabello Monteiro.

Monografia (Graduação). Universidade Federal de Ouro Preto. Instituto de Ciências Exatas e Aplicadas. Departamento de Computação e Sistemas de Informação.

1. Sistemas de informação. 2. Desenvolvimento Web. 3. Aplicações Web. 4. Sistemas de coleta automática de dados. I. Monteiro, Bruno Rabello. II. Universidade Federal de Ouro Preto. III. Título.

CDU: 004.777

Catálogo: [ficha.sisbin@ufop.edu.br](mailto:ficha.sisbin@ufop.edu.br)



## FOLHA DE APROVAÇÃO

José Wilson da Silva Júnior

**GarageParts: Sistema de Controle de Consumo, Deslocamento e Manutenções de Veículos Pessoais**

Membros da banca

Bruno Rabello Monteiro - Mestre - Decsi - Ufop  
Rafael Frederico Alexandre - Doutor - Decsi - Ufop  
Mateus Ferreira Satler - Doutor - Decsi - Ufop

Versão final

Aprovado em 16 de dezembro de 2019

De acordo

Bruno Rabello Monteiro



Documento assinado eletronicamente por **Bruno Rabello Monteiro, PROFESSOR DE MAGISTERIO SUPERIOR**, em 16/01/2020, às 14:18, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site [http://sei.ufop.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **0032317** e o código CRC **ACA080E9**.

# Agradecimentos

Agradeço a Deus, acima de tudo, por todas as bem-aventuranças que me foram concedidas durante minha vida. Agradeço aos meus pais, José Wilson e Geralda, por toda a dedicação de uma vida voltada a construção da educação de seus filhos enquanto seres humanos e pessoas de caráter, e também por todo o suporte nessa caminhada. Agradeço ao meu irmão Wilson José pelo companheirismo, compreensão e apoio. Agradeço a todos os familiares e amigos pela torcida, bons conselhos, participação e boas energias. Agradeço ao prof. Bruno, que foi meu orientador neste trabalho, por toda a contribuição e paciência desde o início do projeto. Agradeço aos meus colegas da universidade que de alguma forma participaram da minha jornada, de maneira especial ao Samuel, Edeilson, Carla e Camilla que se mostraram grandes companheiros nos estudos. Agradeço também aos meus companheiros da Van que tornaram o traslado diário entre a minha cidade e a universidade mais agradável e menos exaustivo. Agradeço aos meus colegas de trabalho pela paciência e compreensão, nos momentos críticos dessa dupla jornada de trabalho e graduação. Por fim, agradeço imensamente a todos os professores que compartilharam com maestria seus conhecimentos, tornando possível o desenvolvimento desse trabalho.

*“Science is more than a body of knowledge; it is a way of thinking.”*

— Carl Sagan (1934 – 1996),  
*in: The Demon-Haunted World: Science as a Candle in the Dark.*

# Resumo

Os sistemas de informações fazem parte do cotidiano da sociedade contemporânea, em particular, os sistemas *web* se tornaram muito populares com o avanço da tecnologia de informação. Através destes sistemas, tarefas do dia a dia, antes feitas manualmente, se tornaram mais práticas, rápidas e automatizadas. Seguindo este contexto, no presente trabalho é desenvolvido um sistema *web* responsivo, denominado GarageParts, que objetiva fornecer um meio funcional para o controlar o consumo, o deslocamento e as manutenções de veículos pessoais. O sistema permite o cadastro dos veículos e todos os dados relacionados ao consumo e manutenção. O sistema armazena um histórico de cada veículo, permite a geração de relatórios, gera notificações sobre as manutenções e permite a transferência dos dados nas vendas dos veículos.

**Palavras-chaves:** Sistemas de Informação. Desenvolvimento *web*. Gerenciamento de Veículos Pessoais.

# Abstract

Information systems are part of the daily life of contemporary society, in particular, web systems have become very popular with the advancement of information technology. Through these systems, daily tasks, previously done manually, have become more practical, faster and automated. Following this context, the present work develops a responsive web system, called GarageParts, which aims to provide a functional means to control the consumption, displacement and maintenance of personal vehicles. The system allows the registration of vehicles and all data related to consumption and maintenance. The system stores a history of each vehicle, allows the generation of reports, generates notifications about maintenance and allows the transfer of data on vehicle sales.

**Key-words:** Information Systems. Web development. Personal Vehicle Management.

# Lista de ilustrações

Figura 1 – Atividades de um sistema de informação . . . . .	15
Figura 2 – Exemplo de uso da API da Tabela FIPE. . . . .	17
Figura 3 – Interface do programa Habil Pessoal + Veiculos . . . . .	18
Figura 4 – Interface do aplicativo Drivvo . . . . .	19
Figura 5 – Menu de opções e formulário de abastecimento do aplicativo Carango .	20
Figura 6 – Caso de uso gestão pessoal de veículos com GarageParts . . . . .	29
Figura 7 – Diagrama de Entidade-Relacionamento do sistema GarageParts. . . . .	30
Figura 8 – Página home do sistema GarageParts . . . . .	32
Figura 9 – Formulário de cadastro de novo usuário . . . . .	32
Figura 10 – Tela principal da área restrita do usuário denominada Linha do Tempo.	33
Figura 11 – Tela do gerenciamento de perfil do usuário. . . . .	33
Figura 12 – Formulário de cadastro do veículo. . . . .	34
Figura 13 – Tela do gerenciamento de Veículos. . . . .	35
Figura 14 – Linha do tempo com eventos. . . . .	35
Figura 15 – Tela formulário de cadastro de abastecimento. . . . .	36
Figura 16 – Tela gerenciamento de abastecimentos. . . . .	37
Figura 17 – Tela gerenciamento de notificações. . . . .	37
Figura 18 – Tela cadastro e gerenciamento de manutenções do serviços. . . . .	38
Figura 19 – Interface de transferencia do veículo, seleção do comprador. . . . .	39

# Lista de tabelas

Tabela 1 – Comparação entre as aplicações analisadas. . . . .	21
Tabela 2 – Requisitos funcionais do sistema GarageParts. . . . .	23
Tabela 3 – Requisitos não-funcionais. . . . .	26
Tabela 4 – Associação dos casos de uso aos requisitos funcionais. . . . .	27
Tabela 5 – Comparação entre as aplicações analisadas e o GarageParts. . . . .	41

# Lista de abreviaturas e siglas

**API** *Application Programming Interface*

**CSV** *Comma-separated values*

**CSS** *Cascading Style Sheets*

**FIPE** Fundação Instituto de Pesquisas Econômicas

**GPL** GNU General Public License

**HTTP** *Hypertext Transfer Protocol*

**Json** *JavaScript Object Notation*

**MIT** *Massachusetts Institute of Technology*

**MVC** *Model-View-Controller*

**PHP** *PHP: Hypertext Preprocessor*

**PRO** *Professional*

**REST** *Representational State Transfer*

**RUP** *Rational Unified Process*

**SGBD** Sistemas Gerenciado de Bancos de Dados

**SQL** *Structured Query Language*

**SI** Sistemas de Informação

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>12</b>
1.1	Objetivo	13
1.2	Organização do Trabalho	13
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA</b>	<b>14</b>
2.1	Tecnologias Utilizadas	14
2.2	Trabalhos Relacionados	16
<b>3</b>	<b>ESPECIFICAÇÃO DO SISTEMA</b>	<b>22</b>
3.1	Análise de Requisitos	22
3.2	Casos de Uso	27
3.3	Diagramas de Entidade-Relacionamento	28
<b>4</b>	<b>O SOFTWARE GARAGEPARTS</b>	<b>31</b>
4.1	Cadastro e Gerenciamento de Veículos	34
4.2	Cadastro e Gerenciamento de Abastecimentos	35
4.3	Cadastro e Gerenciamento de Serviços	36
4.4	Cadastro e Gerenciamento de Seguros	38
4.5	Transferência de Veículos	38
<b>5</b>	<b>RESULTADOS</b>	<b>40</b>
5.1	Trabalhos Futuros	42
	<b>REFERÊNCIAS</b>	<b>43</b>

# 1 Introdução

A sociedade contemporânea está a cada dia mais próxima dos sistemas de informação, dos computadores, dos *smartphones*, e um dos motivos é a oferta e preço dos dispositivos e sistemas. Com o avanço da tecnologia, a busca por ferramentas que automatizem tarefas e proporcionem redução do tempo gasto em atividades diárias se torna desejável.

Uma das tarefas que pode ser automatizada é o controle e gerenciamento dos gastos com os veículos particulares por parte dos seus donos. Proprietários de veículos, sejam eles carros ou motos, por muitas vezes não consideram todos os custos que os veículos podem gerar, como por exemplo, as manutenções periódicas, os abastecimentos, as trocas de pneus e o seguro.

Para a realização dessa tarefa, a ferramenta comumente utilizada é a planilha eletrônica, e em alguns casos, planilhas manuscritas. Segundo Wernke (2015) “(...) em que pese cada consumidor utilizar critérios de ordem objetiva ou subjetiva por ocasião da compra, os adquirentes deveriam considerar diversos fatores ligados aos custos totais de propriedade de veículo”.

Neste contexto, é comum que os proprietários sempre busquem maneiras para elencar os custos das manutenções, dos impostos, dos abastecimentos e da aquisição de acessórios para seus veículos, na tentativa de prever quanto recurso será necessário para para cada um destes itens.

Tendo em vista este comportamento, é possível encontrar alguns sistemas que permitem o controle financeiro dos gastos com os veículos. Entretanto, algumas buscas preliminares evidenciaram que a maioria das aplicações não disponibilizam suas funcionalidades inteiramente de forma gratuita e também, quando permitem o controle de gastos com abastecimento, não o fazem em relação aos impostos, por exemplo.

Aplicações como o Cobli<sup>1</sup>; o SofitView<sup>2</sup>; o MasterFrota<sup>3</sup>; o Hábil<sup>4</sup>; e o sistema Frota Fácil<sup>5</sup> correspondem a grandes sistemas comerciais, voltados para o mercado empresarial de grandes frotas veiculares, fugindo portanto, do escopo deste trabalho.

Outras, por sua vez, possuem um alto número de anúncios e propagandas (que servem pra financiar os custos do desenvolvimento e manutenção das ferramentas) e tornam fatídico ou inviável o uso da ferramenta para o resolver este problema.

---

<sup>1</sup> <https://cobli.co>

<sup>2</sup> <https://www.sofit4.com.br/>

<sup>3</sup> <https://masterfrota.com.br>

<sup>4</sup> <http://www.habilpessoal.com.br/>

<sup>5</sup> <https://frota-facil.softonic.com.br/>

Diante do exposto, foram escolhidos três *softwares* que mais se alinharam com a proposta deste trabalho. Estes três sistemas foram comparados, com o intuito de encontrar as características de cada um e as funcionalidades presentes ou não. Os resultados obtidos foram incorporados ao projeto e serviram como ponto de partida para a análise de requisitos para construção do GarageParts.

## 1.1 Objetivo

O objetivo deste trabalho é o projeto e a implementação de um sistema que permita o controle financeiro dos gastos com veículos pessoais. Pretende-se que esse sistema esteja disponível na plataforma *web*, seja gratuito e responsivo, de modo a suportar o acesso através dos *browsers* de navegação em computadores de mesa, computadores portáteis ou mesmo dispositivos móveis, tais como *smartphones* e *tablets*.

Especificamente pretende-se que a aplicação permita que seus usuários possam cadastrar veículos de qualquer marca e modelo, dentre todos os disponíveis no mercado automotivo nacional, através da [API](#) da tabela [FIPE](#) utilizada pelo sistema.

Além disso, é desejável que a aplicação possa registrar manutenções preditivas e/ou corretivas dos veículos, registrar e consultar o histórico de gastos de cada veículo, utilizar de filtros por categorias para melhor visualizar relatórios e informações históricas já cadastradas no sistema, realizar o acompanhamento de gastos com combustível, incluindo históricos de preços, de cada abastecimento realizado, registrar informações sobre o seguro veicular, compartilhar as informações de veículos cadastrados, e por fim, transferir o histórico de informações dos veículos para outros usuários em caso de venda do veículo.

## 1.2 Organização do Trabalho

Este trabalho é composto por 5 capítulos, neste primeiro capítulo, foi feita uma apresentação do tema do trabalho, com a descrição do problema, a motivação e os objetivos do trabalho.

No capítulo [2](#) é feita uma revisão bibliográfica das tecnologias utilizadas neste trabalho. Também são apresentados os trabalhos correlatos encontrados na literatura.

O capítulo [3](#) apresenta a especificação do sistema, contendo os diagramas de casos de uso, diagrama entidade-relacionamento e análise de requisitos do GarageParts.

O capítulo [4](#) apresenta o GaraParts, seus recursos e funcionalidades. Por fim, o capítulo [5](#) mostra as conclusões obtidas após a construção do sistema e as sugestões de trabalhos futuros.

## 2 Revisão Bibliográfica

Este capítulo apresenta uma revisão dos conceitos e tecnologias envolvidas no desenvolvimento do sistema de informação proposto neste trabalho, bem como realiza uma análise comparativa de três programas de propósito correlato ao da ferramenta desenvolvida.

Laudon e Laudon (2015) apresentam a definição de um Sistema de Informação (SI) como uma coleção de componentes que se relacionam internamente coletando, processando, armazenando e distribuindo informações com a finalidade de apoiar a tomada de decisão. E complementam que três atividades em um SI são necessárias para controlar operações, analisar problemas e criar novos produtos, são elas: entrada, processamento e saída.

Ainda segundo Laudon e Laudon (2015, p. 13):

A entrada captura ou coleta dados brutos de dentro da organização ou de seu ambiente externo. O processamento converte esses dados brutos em uma forma mais significativa. A saída transfere as informações processadas às pessoas que as utilizarão ou às atividades nas quais serão empregadas.

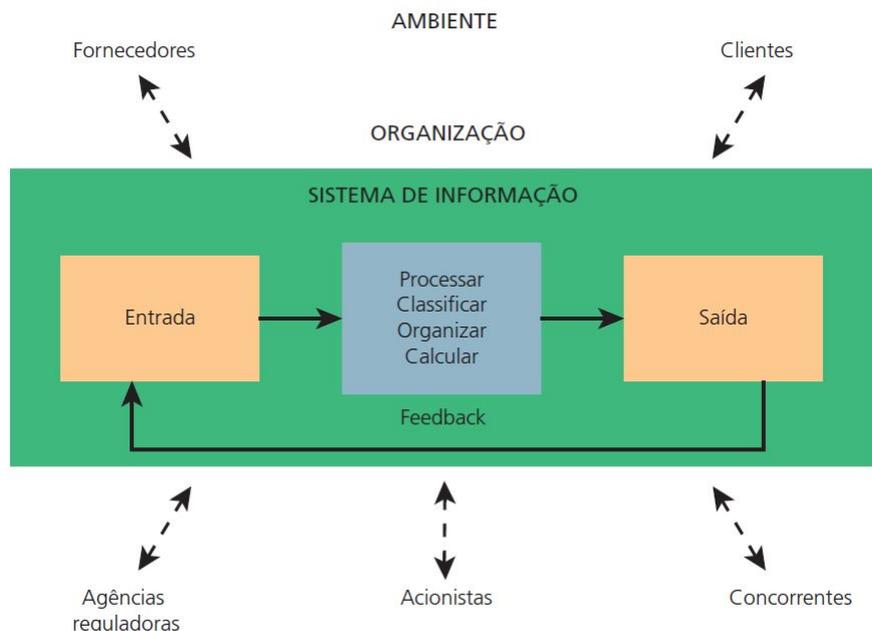
A Figura 1 ilustra a representação das atividades de entrada, processamento e saída em interação em um ambiente organizacional. Neste trabalho os conceitos apreentados sobre SI podem ser aplicados de modo que a entrada, são as informações dos veículos e serviços cadastradas pelo usuário, enquanto o processamento é feito pelo sistema GarageParts, resultando em saídas tais como: informações históricas das manutenções, abastecimentos e serviços.

O capítulo 3 tratará da especificação do sistema GarageParts, bem como apresentará os artefatos de Engenharia de Software concebidos na etapa de projeto. A seção 2.1 apresenta as tecnologias utilizadas no desenvolvimento do sistema.

### 2.1 Tecnologias Utilizadas

Para este projeto definiu-se a arquitetura MVC, que de acordo com Sommerville (2011) separa os elementos de um sistema e define as suas responsabilidades, permitindo mudá-los de forma independente. Neste contexto, o *model* representa os dados, a *view* representa a forma como os dados são expostos aos usuários, e por fim o *controller* manipula e roteia as requisições dos usuários.

Um dos primeiros requisitos do sistema foi ser um sistema *web*. Logo foi necessário a definição de uma linguagem de programação que atendesse esse objetivo. Nesse intuito,



Fonte: [Laudon e Laudon \(2015, p. 15\)](#)

Figura 1 – Atividades de um sistema de informação

definiu-se como *framework* de desenvolvimento o Laravel ([LARAVEL, 2019](#)). O Laravel utiliza a linguagem de programação **PHP**, além de ser livre e *open source*, e tem por principal objetivo proporcionar um ambiente estruturado, que incorpora a arquitetura **MVC**, possibilitando a criação de trabalhos de forma ágil.

O Laravel possui ferramentas de produtividade que elevam a experiência do desenvolvimento, algumas delas são:

- Sistema de *templates Blade*: tem por objetivo reduzir a quantidade de código **PHP** e aumentar o reuso. Seus principais benefícios são a herança e as seções.
- *Eloquent e QueryBuilder*: o Eloquent representa cada tabela do banco de dados através de uma classe *Model* que serve de interface para interação com esta tabela. Já o *QueryBuilder* é um construtor de *queries* e permite executar a maioria das operações de banco de dados.
- *Artisan Console*: fornece uma interface de linha de comando para facilitar o desenvolvimento da aplicação, criação de novos elementos e classes.

Outra tecnologia utilizada foi o AdminLTE ([ADMINLTE, 2019](#)), que é um *template* para painéis administrativos e painéis de controle *open source*. Ele é baseado na estrutura de **CSS Bootstrap 3**, sob a licença **MIT** e fornece painéis e *dashboards* que podem ser configurados de acordo com a aplicação desenvolvida. Também possui ampla documentação

e é compatível com a grande maioria dos *frameworks*. A escolha do AdminLTE deu-se pela versatilidade e pelas opções de componentes responsivos que podem ser usados na construção do *front-end* na camada da *view*.

Dado a estrutura dos dados necessários para aplicação foi pensado um Sistema de Gerenciamento de Banco de Dados ou **SGBD** Relacional. Um SGBD é definido por Elmasri e Navathe (2005, p. 4) como:

(...) uma coleção de programas que permite aos usuários criar e manter um banco de dados. O SGBD é, portanto, um sistema de software de propósito geral que facilita os processos de definição, construção, manipulação e compartilhamento de banco de dados entre vários usuários e aplicações.

Elmasri e Navathe (2005) acrescentam que o **SGBD** provê importantes funções como a proteção e manutenção do banco de dados por grandes períodos, que inclui proteção do sistema contra falhas e também provê a segurança contra acessos não autorizados.

O **SGBD** escolhido foi o MySQL (MYSQL, 2019). As características que pautaram a escolha foram: ser *open source*, licenciado sob **GPL**; uso próximo do padrão da linguagem **SQL**, além de facilidades de conexão com as linguagens de programação para **web**, como a **PHP**, já escolhida.

Este projeto ainda utiliza uma API que realiza consultas na tabela FIPE (FORTUNA, 2019), disponibilizada de forma gratuita através do repositório na plataforma `gitHub`<sup>1</sup>. De modo resumido a API fornece preços médios de veículos do mercado nacional, o que justifica sua escolha, através de um serviço **RESTful HTTP Json**.

Para utilizar a API o sistema realiza consultas através de requisições como: **GET: <https://parallelum.com.br/fipe/api/v1/carros/marcas>**, onde a ação `marcas` sem nenhum parâmetro, retorna uma lista de todos os carros das marcas disponíveis na tabela FIPE em formato **Json** como exemplificado na Figura 2. Todos os demais requests disponíveis através da API podem ser consultados em Fortuna (2019).

Realizada essa breve descrição das tecnologias escolhidas para a realização deste trabalho, a seção 2.2 apresenta os trabalhos relacionados encontrados na literatura.

## 2.2 Trabalhos Relacionados

A metodologia adotada neste trabalho pode ser vista em Wazlawick (2004) e compreende a fase de concepção, que incorpora o estudo da viabilidade como primeira etapa. Para suprir esta fase optou-se por uma pesquisa de aplicações similares.

<sup>1</sup> <https://github.com/deividfortuna/fipe>

## Marcas

Primeiro liste as marcas do tipo de veículo que você deseja, através da ação marcas e sem nenhum parâmetro:

### Request

GET: <https://parallelum.com.br/fipe/api/v1/carros/marcas>

Você pode alterar na url "carros" para "motos" ou "caminhoes" de acordo com a sua necessidade.

### Reponse

```
▼ Array (5)
  0: Object
    nome: "Acura"
    codigo: "1"
  1: Object
    nome: "Agrale"
    codigo: "2"
  2: Object
    nome: "Alfa Romeo"
    codigo: "3"
  3: Object
    nome: "AM Gen"
    codigo: "4"
  4: Object
    nome: "Asia Motors"
    codigo: "5"
```

Fonte: Fortuna (2019)

Figura 2 – Exemplo de uso da API da Tabela FIPE.

Alguns dos sistemas encontrados são *softwares* proprietários e possuem direitos reservados. Para esses casos, a obtenção do acesso se fez possível através da concessão por parte dos fabricantes dos sistemas, após a apresentação de uma justificativa para a finalidade das pesquisas.

Deste modo, foram obtidas licenças *trial*, que as empresas concedem para quem deseja uma demonstração das funcionalidades dos sistemas. Essas licenças possuem limitações de funcionalidades ou tempo de uso. O resultado dessa pesquisa pode ser visto na Tabela 1, apresentada ao fim desta seção, que compara algumas funcionalidades dos sistemas encontrados.

Um dos primeiros sistemas a ser analisado foi o Hábil Pessoal + Veículos (HáBIL, 2019), que consiste na união de um programa para gestão financeira e um *software* para gerenciamento de veículos pessoais. Esse sistema foi lançado em 2010, pela empresa Koinonia Software<sup>2</sup>, e inclui um módulo de Controle Pessoal de Veículos, com gastos com veículos, controle de viagens, manutenções pendentes, gráficos, média de km/litro, imagens e fotos, além de estar integrado ao módulo de controle financeiro.

Para utilizar o sistema é necessário realizar o registro no site do fabricante<sup>3</sup>. O registro é gratuito e oferece as funcionalidades para a gestão de até dois veículos. O Hábil Pessoal + Veículos está disponível para a plataforma *Windows*. Existe também uma versão avançada que inclui todas as funcionalidades da versão gratuita, sem propagandas e sem limite para a quantidade de veículos gerenciados.

Entre as funcionalidades não encontradas no sistema estão a transferência de

<sup>2</sup> <http://www.habilpessoal.com.br/sobre.asp>

<sup>3</sup> <http://www.habilpessoal.com.br/registro.asp>

propriedade dos veículos, a não realização de *backup*, nem a exportação dos dados da aplicação. Outra deficiência está na não realização de consultas aos preços de mercado dos veículos da tabela FIPE. A Figura 3 apresenta um interface do programa Hábil Pessoal + Veículos.



Fonte: Hábil (2019)

Figura 3 – Interface do programa Hábil Pessoal + Veículos

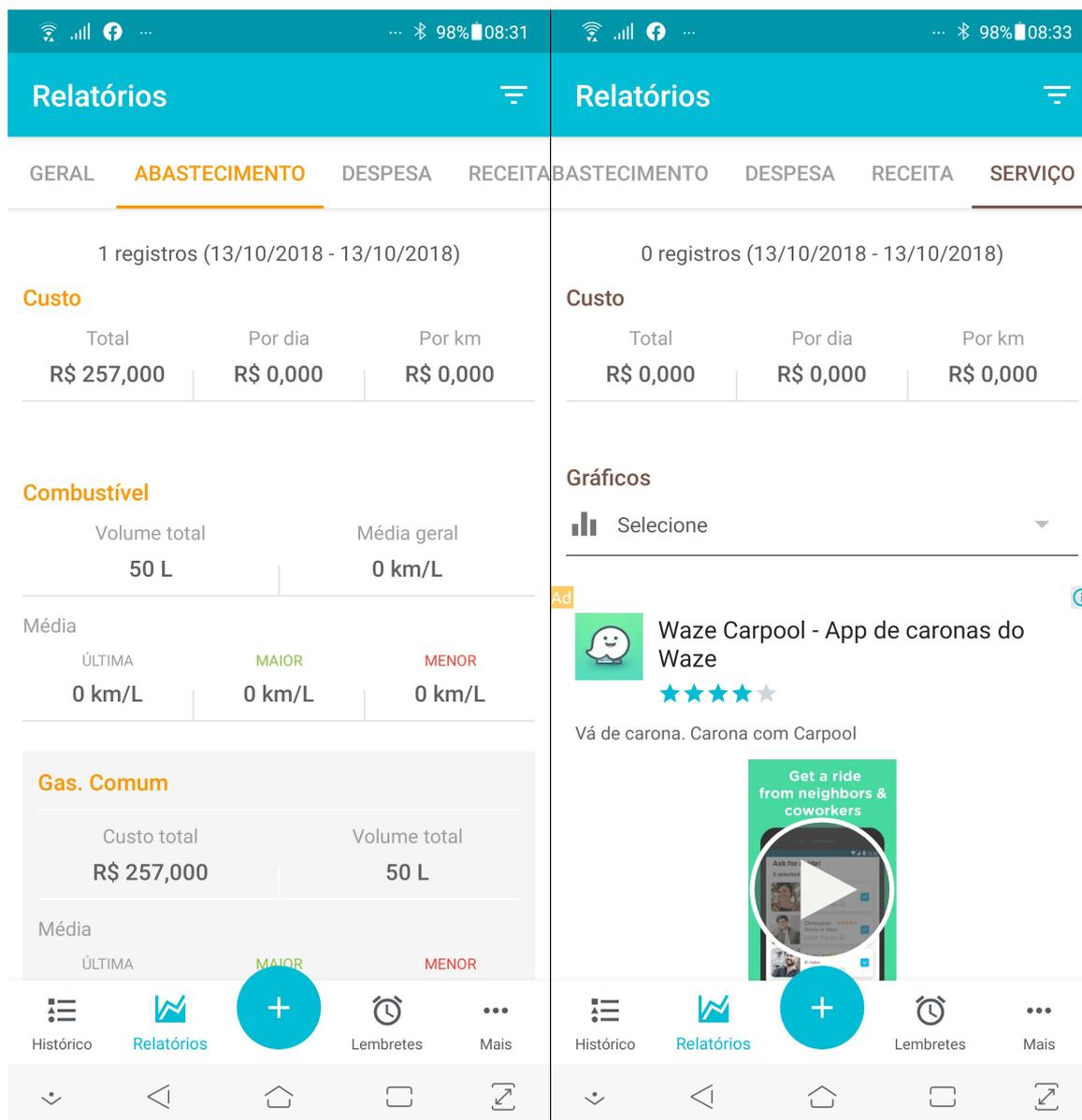
Outro sistema encontrado foi o Drivvo<sup>4</sup> (DRIVVO, 2019). Ele é um aplicativo gestor de finanças para veículos de uso pessoal ou para fins profissionais. O aplicativo possibilita o controle de gasto com combustível, manutenções e despesas.

O Divvo está disponível apenas para as plataformas *Android* (ANDROID, 2019) e *iOS* (APPLE, 2019). Possui uma versão gratuita que dispõe de anúncios e limitação para algumas funcionalidades. É possível ter acesso a versão PRO, por meio de aquisição através das lojas *Play Store* e *App Store*.

A versão gratuita, utilizada para comparação neste trabalho não possui *backup* dos dados na nuvem, não permite sincronizar dados entre dispositivos, possui anúncios, não permite exportar os dados referente aos seus veículos (em formatos como CSV/Excel). Este aplicativo não integra consultas a Tabela FIPE, portanto não exibe valor de referência dos veículos utilizados pelo mercado de automóveis. Não realiza a transferência da propriedade em caso de venda dos veículos para outros usuários do sistema. A Figura 4 apresenta a interface do aplicativo Drivvo.

O Carango (CARANGO, 2019) também é um aplicativo que permite o gerenciamento de informações relevantes sobre veículos. Este aplicativo permite acompanhamento

<sup>4</sup> <http://www.drivvo.com/pt/>



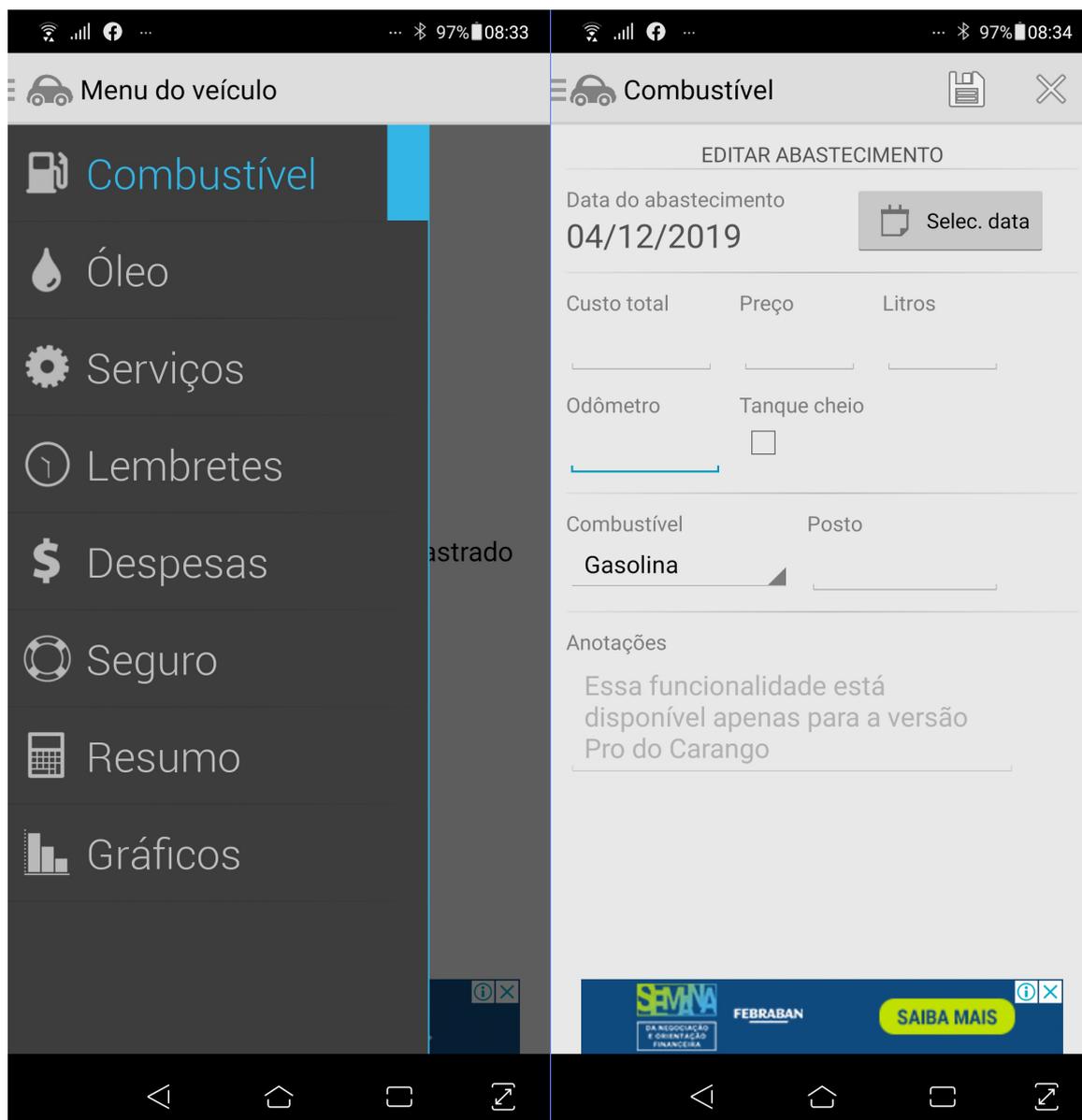
Fonte: Drivvo (2019)

Figura 4 – Interface do aplicativo Drivvo

da quilometragem do veículo, o consumo de combustível, as manutenções, os serviços e os custos de vários tipos de veículos. Seu *download* está disponível através da loja de aplicativos *Play Store* da plataforma *mobile Android*.

Este aplicativo possui uma versão gratuita que apresenta funcionalidades limitadas e anúncios. Também é possível ter acesso a versão **PRO** do aplicativo livre de anúncios e com todas as funcionalidades disponíveis, mediante a compra. A **Figura 5** apresenta a interface do Carango.

O Carango não dispõe de versões para outras plataformas e sistemas operacionais. A aplicação também não possui integração com a tabela **FIPE**, portanto não exhibe o valor



Fonte: Carango (2019)

Figura 5 – Menu de opções e formulário de abastecimento do aplicativo Carango

de referência dos veículos utilizados pelo mercado de automóveis. Também não é possível realizar a transferência da propriedade do veículo entre os usuários do aplicativo, ou anexar documentos no cadastro do serviço. Sua versão gratuita não dispõe de exportação das informações do aplicativo (como arquivos CSV ou similares) e não dispõe de sistema de *backup* das informações.

A Tabela 1 mostra a comparação das características encontradas nas aplicações.

Tabela 1 – Comparação entre as aplicações analisadas.

<b>Comparação entre sistemas do mercado</b>			
<b>Características</b>	<b>Habil Pessoal + Veículos</b>	<b>Drivvo</b>	<b>Carango</b>
<b>Tipo de licença</b>	Gratuito (2 veículos)	Gratuito com funcionalidades limitadas e anúncios	Gratuito com funcionalidades limitadas e anúncios
<b>Plataforma</b>	<i>Windows</i>	<i>Android e iOS</i>	<i>Android</i>
<b>Limite de veículos cadastrados</b>	2 veículos ( <i>trial</i> )	Ilimitado	Ilimitado
<b>Controle de abastecimentos</b>	Sim	Sim	Sim
<b>Controle de quilometragem dos veículos</b>	Sim	Sim	Sim
<b>Controle de manutenções</b>	Sim	Sim	Sim
<b>Lembretes / Pendencias</b>	Sim	Sim	Sim
<b>Consulta valor Tabela FIPE</b>	Não	Não	Não
<b>Exportar dados da aplicação</b>	Versão Completa	Versão PRO	Versão PRO
<b>Cadastro de receitas</b>	Não	Sim	Não
<b>Backup das informações da aplicação</b>	Versão Completa	Versão PRO	Versão PRO
<b>Galeria de Fotos</b>	Sim	Não	Não
<b>Tranferência de veículos e informações históricas entre usuários</b>	Não	Não	Não

Fonte: Elaborado pelo próprio autor.

## 3 Especificação do Sistema

A metodologia adotada para especificar, desenvolver e testar o sistema proposto neste trabalho é inspirada no *framework* de processo da engenharia de *software* denominado RUP, descrita em Wazlawick (2004). De modo resumido, essa metodologia propõe as etapas: Concepção, Elaboração, Construção e Transição.

A fase de *concepção* incorpora o estudo da viabilidade e divide parte da análise de requisitos com a fase de *elaboração*. Esta por sua vez, incorpora a maior parte da análise de requisitos, análise de domínio e projeto.

O resultado dessas fases são apresentados neste capítulo, com os artefatos gerados na forma de requisitos funcionais e não-funcionais, diagramas de casos de uso e diagrama de entidade-relacionamento. A ferramenta AstahUML (ASTAH, 2019) foi usada na elaboração dos diagramas de casos de uso, já para o diagrama entidade-relacionamento foi escolhida a ferramenta brModelo (CANDIDO, 2017).

Como resultado da fase de *concepção* segue a descrição do sistema proposto.

O sistema web deve ser responsivo, voltado para o controle de consumo, deslocamento e manutenções de veículos pessoais. O objetivo do sistema é possibilitar aos seus usuários um maior controle das informações relevantes como histórico de manutenções e abastecimento, deslocamento e gastos de seus automóveis. A partir das informações de uso, manutenções preventivas e corretivas cadastradas pelos usuários, deverão ser gerados relatórios de gastos com peças, abastecimentos e mão de obra. O sistema deverá calcular automaticamente média de consumo de combustível, gerar notificações de pendências de acordo com o plano de manutenção agendadas ou rotineiras para cada veículo cadastrado pelo usuário. O sistema deverá possibilitar o cadastro de vários usuários, porém todas as informações referentes aos veículos serão restritas e particulares, portanto um usuário não poderá em seu perfil ter acesso a informações inseridas por outros usuários.

### 3.1 Análise de Requisitos

A análise de requisitos é fundamental para por se tratar de uma etapa associada ao processo da descoberta de quais funcionalidades o sistema deverá possuir para atender ao objetivo proposto.

A análise de requisitos divide os requisitos em dois tipos, conforme as definições de

(WAZLAWICK, 2004):

- Requisitos funcionais: correspondem a todas as tarefas (funcionalidades) que o sistema deve executar.
- Requisitos não-funcionais: são as restrições colocadas para definir a forma como o sistema deve realizar os requisitos funcionais.

A Tabela 2 exibe a lista dos requisitos funcionais contempladas no sistema GarageParts. A estrutura da tabela foi baseada em Wazlawick (2004), com as devidas adaptações, a coluna Código apresenta sempre a letra F seguido de um número iniciando a partir do 1. O campo Nome apresenta uma especificação curta do requisito em questão. Já a coluna Descrição se refere a uma descrição longa do requisito com maior detalhamento.

Tabela 2 – Requisitos funcionais do sistema GarageParts.

<b>Requisitos Funcionais</b>			
<b>Código</b>	<b>Nome</b>	<b>Descrição</b>	<b>Categoria</b>
<b>F1</b>	Cadastrar conta de usuário	O sistema deve permitir que seus usuários possam se cadastrar através de uma conta de e-mail válida, nome completo e senha.	Evidente
<b>F2</b>	Editar perfil do usuário	Após cadastrado e autenticado, o sistema deve permitir que seus usuários alterem informações como nome, apelido ou senha cadastrada em sua conta de usuário.	Evidente
<b>F3</b>	Cadastrar veículos	Usuário deverá efetuar o login no sistema e fornecer as informações do veículo como placa, código renavan, cor, marca e modelo, valor da compra, data da compra e quilometragem.	Evidente
<b>F4</b>	Cadastrar serviços	O sistema deverá permitir o cadastro de serviços agendados ou executados para cada veículo quando necessário.	Evidente
Continua na próxima página			

Tabela 2 – continuação da página anterior.

Requisitos Funcionais			
Código	Nome	Descrição	Categoria
<b>F5</b>	Cadastrar manutenções por tipo	O sistema deve categorizar o tipo de manutenção que é possível incluir nos serviços, dividindo-os em mecânicos, hidráulicos, elétricos e de pneus.	Evidente
<b>F6</b>	Cadastro de abastecimentos	O cadastro de abastecimento requer que exista ao menos um veículo cadastrado, no ato do cadastramento deverão ser informados: quantidade, data, valor e quilometragem do veículo.	Evidente
<b>F7</b>	Cadastro de seguro	O sistema deve incluir a funcionalidade de cadastrar apólices de seguro para cada um dos veículos do proprietário.	Evidente
<b>F8</b>	Gerenciamento de veículos	Após cadastrados os veículos, o sistema deve permitir que o usuário proprietário do veículo possa gerenciar seus veículos em uma área apropriada do sistema oferecendo a possibilidade de editar, excluir ou transferir um veículo.	Evidente
<b>F9</b>	Gerenciamento de serviços	Após cadastrados os serviços, o sistema deve permitir que o usuário proprietário do veículo possa gerenciar os serviços possibilitando listar, filtrar, editar ou excluir os serviços.	Evidente
<b>F10</b>	Gerenciamento de seguros	Após cadastrados os seguros, o sistema deve permitir ao usuário proprietário renovar um seguro, exibir o histórico dos seguros anteriores ou excluir o seguro.	Evidente

Continua na próxima página

Tabela 2 – continuação da página anterior.

Requisitos Funcionais			
Código	Nome	Descrição	Categoria
<b>F11</b>	Gerenciamento de abastecimentos	Após cadastrados os abastecimentos, o sistema deve permitir que o usuário proprietário do veículo possa gerenciar os abastecimentos em uma área apropriada do sistema possibilitando listar, filtrar, editar ou excluir os serviços.	Evidente
<b>F12</b>	Gerenciamento de Pendências	O sistema deve ser capaz de identificar os serviços agendados, renovação de seguros, limite de quilometragem ou data de vencimento das manutenções e indicar para o usuário a qual item em questão o lembrete se refere.	Evidente
<b>F13</b>	Linha do tempo	O sistema deve registrar e exibir as interações do usuário com a aplicação através de eventos em ordem cronológica, categorizando-os em "veículos, serviços, abastecimentos e transferências" de acordo com o contexto da execução de cada evento.	Evidente
<b>F14</b>	Transferir a propriedade do veículo	O sistema deve permitir que seus usuários realizem a transferência da propriedade de seus veículos entre si, de modo que o histórico de serviços, abastecimentos, seguros e demais informações relacionadas o acompanhem a transferência.	Evidente
<b>F15</b>	Consulta na tabela FIPE	O sistema deve consultar a API da tabela FIPE referente ao mês atual e exibir ao usuário informações atualizadas referente ao veículo sempre que requisitado.	Oculto

Fonte: Elaborado pelo próprio autor.

Na sequência é apresentada a Tabela 3 listando os requisitos não-funcionais que

representam algumas das restrições impostas aos requisitos funcionais.

Tabela 3 – Requisitos não-funcionais.

Requisitos Não-funcionais			
Código	Nome	Restrição	Categoria
NF1	Controle de acesso	O sistema deverá assegurar que apenas usuários autenticados através do login (e-mail) e senha, consigam ter acessos as informações de seus veículos. Também deverá garantir que um usuário não consiga acessar as informações cadastradas por outro usuário.	Segurança
NF2	Sistema Web	O sistema será desenvolvido em plataforma web e deverá ser compatível com os navegadores mais conhecidos no mercado como exemplo, Safari <sup>1</sup> , Mozilla Firefox <sup>2</sup> , Edge <sup>3</sup> , Opera <sup>4</sup> e Google Chrome <sup>5</sup> .	Implementação
NF3	Sistema responsivo	O sistema deverá ser implementado de forma responsiva, possibilitando que a sua interface se ajuste em monitores e telas de tamanhos variados e alcançar compatibilidade com dispositivos como laptops, desktops, tablets e smartphones.	Interface
NF4	Arquitetura MVC	O sistema deverá ser implementado de seguindo as características da arquitetura MVC, tal que o sistema seja modular, escalável, facilite o reaproveitamento de código e possibilite uma boa performance.	Implementação
Continua na próxima página			

<sup>1</sup> <https://www.apple.com/br/safari/>

<sup>2</sup> <https://www.mozilla.org/pt-BR/firefox/new/>

<sup>3</sup> <https://www.microsoft.com/pt-br/windows/microsoft-edge>

<sup>4</sup> <https://www.opera.com/pt-br>

<sup>5</sup> <https://www.google.com/intl/pt-BR/chrome/>

Tabela 3 – continuação da página anterior.

Requisitos Não-funcionais			
Código	Nome	Restrição	Categoria
NF5	Persistência de dados	O sistema deverá garantir que após os dados que serem cadastrados e salvos, não sejam perdidos, mesmo que o dispositivo pelo qual a aplicação esteja sendo acessada se desconecte da <i>internet</i> .	Confiabilidade

Fonte: Elaborado pelo próprio Autor.

## 3.2 Casos de Uso

Os casos de uso fornecem uma interface para que desenvolvedores, usuários finais e especialistas do domínio possam chegar a uma compreensão em comum. Também auxiliam na validação da arquitetura a medida que evolui mediante seu desenvolvimento. Conforme apresentado em [Booch, Rumbaugh e Jacobson \(2006\)](#) um caso de uso é uma descrição do conjunto sequencial das ações que especificam o comportamento de um sistema ou da suas partes.

Na [Figura 6](#) é apresentando o diagrama que representa o contexto das interações entre os atores apresentados na forma de bonecos "Proprietário", "API Tabela Fipe" e o sistema GarageParts. As elipses representam os casos de uso.

A Tabela 4 relaciona cada caso de uso, representados no diagrama, com os requisitos funcionais correspondentes definidos na sessão 3.1, acrescentando uma breve descrição.

Tabela 4 – Associação dos casos de uso aos requisitos funcionais.

Relacionamento casos de uso e Requisitos funcionais			
Nome	Ator	Descrição	Referências
Cadastrar Veículo	Proprietário, API Tabela FIPE	O proprietário deve inserir as informações do veículo, enquanto a marca, modelo e o ano do veículo serão consultadas na API da Tabela FIPE.	F4, F13, 15.

Continua na próxima página

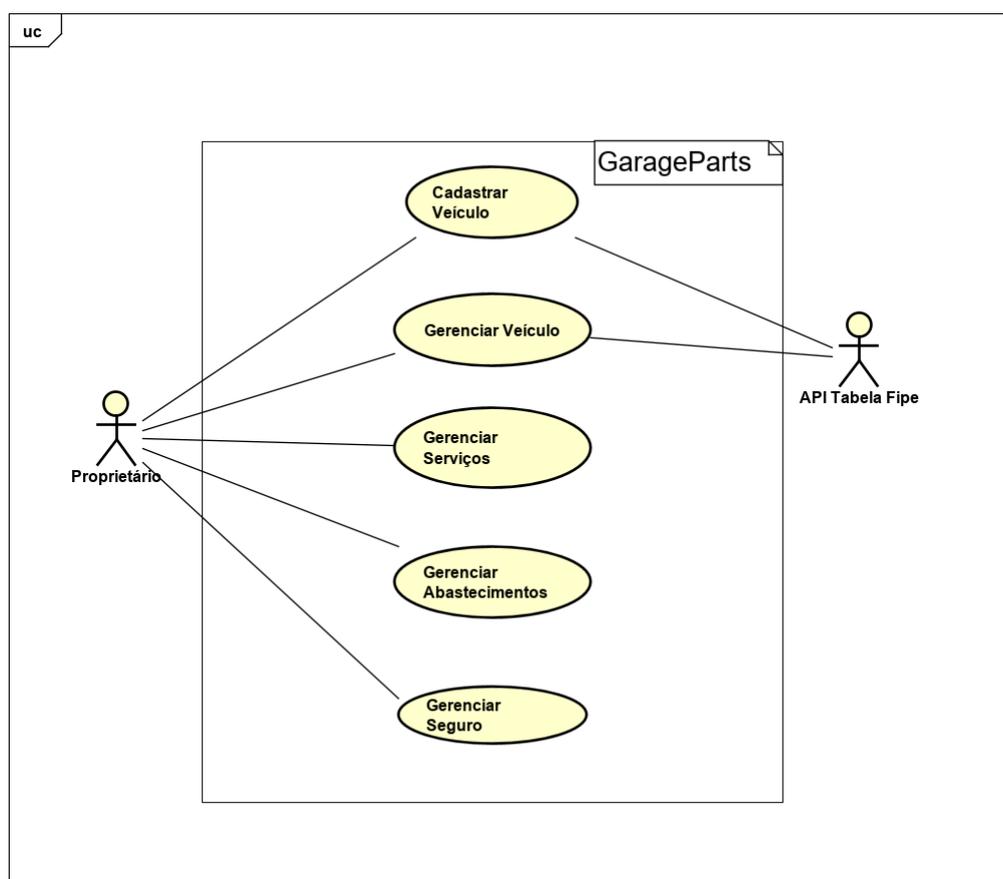
Tabela 4 – continuação da página anterior.

Relacionamento casos de uso e Requisitos funcionais			
Nome	Ator	Descrição	Referências
<b>Gerenciar Veículo</b>	<b>Proprietário, API Tabela FIPE</b>	O proprietário pode editar, transferir ou consultar as informações do veículo, incluindo o valor de mercado referente a Tabela FIPE.	F3, F8, F12, F13, F15.
<b>Gerenciar Serviços</b>	<b>Proprietário</b>	O proprietário poderá cadastrar, editar, listar, filtrar ou excluir serviços e manutenções sempre que necessário. Quando for pertinente os serviços poderão ser agendados para gerar notificações posteriormente.	F4, F5, F8, F9, F12, F13.
<b>Gerenciar Abastecimentos</b>	<b>Proprietário</b>	O proprietário pode registrar sempre que necessário novos abastecimentos. Quando conveniente também pode consultar as informações históricas dos antigos abastecimentos.	F6, F11, F13.
<b>Gerenciar Seguro</b>	<b>Proprietário</b>	O proprietário pode cadastrar seguros para manter o controle sobre a vigência do seguro. O proprietário poderá também a qualquer momento, renovar as apólices de seguro expiradas e consultar os contratos antigos que foram cadastrados.	F7, F10, F12, F13.

Fonte: Elaborado pelo próprio Autor.

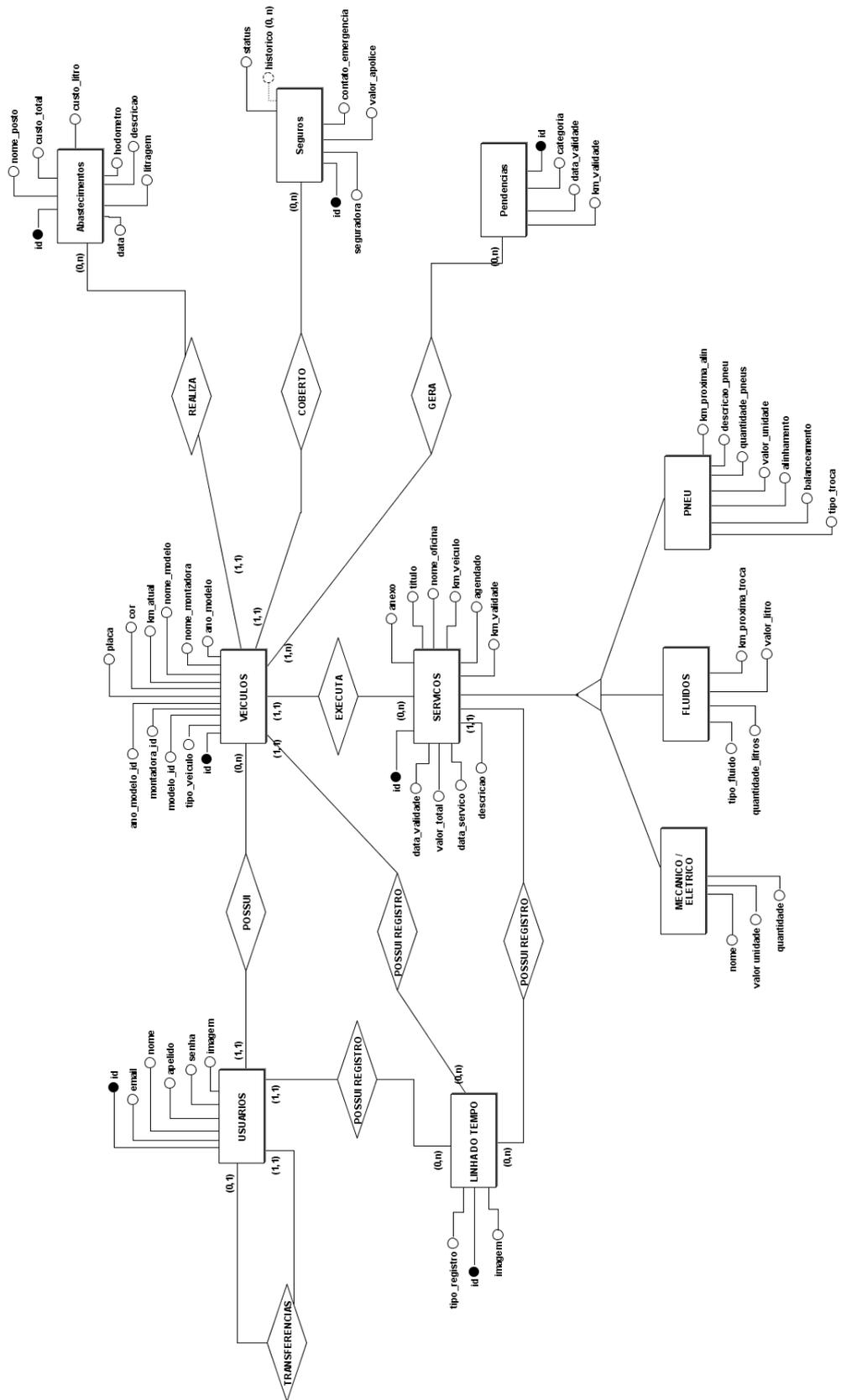
### 3.3 Diagramas de Entidade-Relacionamento

O Diagrama Entidade-Relacionamento é tido como notação fundamental para a modelagem de dados conceitual. [Rezende \(2006\)](#) define que o "principal propósito do DER é representar os objetos de dados e suas relações, sendo que cada entidade termina representada pelo menos por uma tabela de dados". A [Figura 7](#) apresenta o Diagrama de Entidade-Relacionamento desenvolvido para o GarageParts.



Fonte: Elaborado pelo próprio autor

Figura 6 – Caso de uso gestão pessoal de veículos com GarageParts



Fonte: Elaborado pelo próprio autor

Figura 7 – Diagrama de Entidade-Relacionamento do sistema GarageParts.

## 4 O Software GarageParts

Após a conclusão das fases de *concepção* e *elaboração*, que resultaram nas funcionalidades do sistema, o GarageParts foi implementado.

O programa permite cadastrar e gerenciar as informações referentes a dois tipos de veículos: Motos e Carros, de acordo com as marcas e modelos disponíveis no mercado e referenciados: na Tabela [FIPE](#).

Para cada veículo é possível cadastrar os itens: serviço(s) agendado(s) ou já executado(s) (sendo que os serviços devem ser cadastrados antes de incluir as manutenções); as manutenções, por sua vez, são classificadas em "mecânicas, elétricas, hidráulicas, pneus e mão de obra"; abastecimento(s) e seguro veicular. Cabe ressaltar que todos estes itens só podem ser incluídos no sistema caso o usuário tenha adicionado ao menos um veículo.

O desenvolvimento do sistema, projetado para o ambiente *web*, seguiu a arquitetura [MVC](#), utilizando o [SGBD MySQL versão 5.7](#)<sup>1</sup> na camada de persistência de dados, [AdminLTE 2.4](#)<sup>2</sup> na construção das *views*, o [Laravel 5.5](#)<sup>3</sup> provendo a interação entre as camadas de *model* e *view* realizando o roteamento por meio da camada de *controller*. Como IDE de desenvolvimento foi utilizado o Visual Studio Code v1.4<sup>4</sup>.

Considerando que o sistema está devidamente hospedado, *online* e operacional, quando seu endereço é acessado através de um *browser* de Internet, a página inicial do sistema é exibida, expondo o *layout* do site, com uma barra de navegação fixada na parte superior da tela. Nesta barra de navegação se encontram dois botões, um para cadastro do usuário e outro para o *login*, uma vez que o usuário já completou seu cadastro. A [Figura 8](#) ilustra a página inicial do sistema.

Para obter acesso aos recursos implementados no sistema, primeiro o usuário deverá realizar o cadastro de sua conta ( [Figura 9](#)). Após preenchimento e submissão de todas as informações do formulário de cadastro de usuário, o sistema confirmará a criação do usuário e o redirecionará para a área restrita do usuário ou alertará da falha na criação e solicitará a correção dos dados informados.

A [Figura 10](#) exibe a tela da área restrita do usuário, após o processo de autenticação ser concluído. Todas as funcionalidades do sistema estarão disponíveis e sem restrição, portanto, para todas as ações retratadas adiante, deve-se considerar que o usuário realizou o *login* no sistema com êxito.

<sup>1</sup> <https://dev.mysql.com/downloads/mysql/5.7.html>

<sup>2</sup> <https://adminlte.io/docs/2.4/installation>

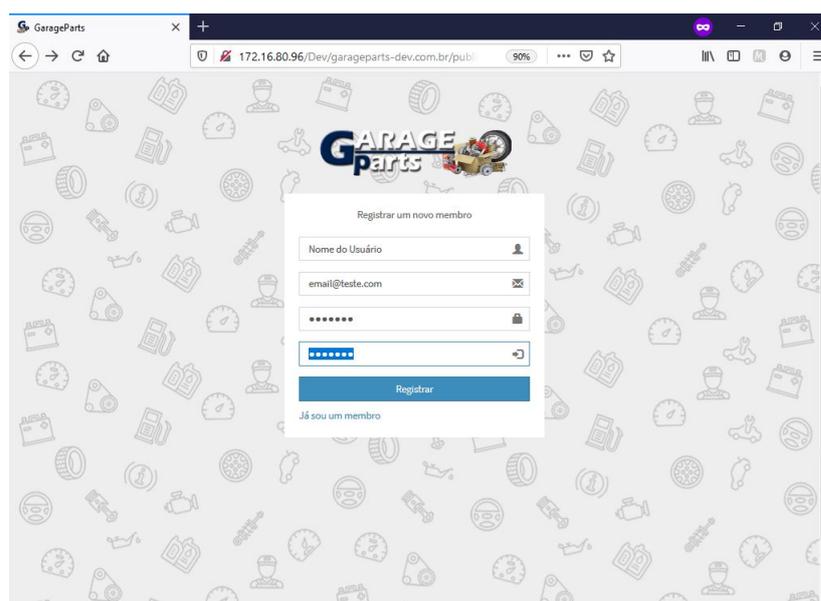
<sup>3</sup> <https://laravel.com/docs/5.5/releases>

<sup>4</sup> <https://code.visualstudio.com/updates/v1.40>



Fonte: Elaborado pelo próprio autor.

Figura 8 – Página home do sistema GarageParts

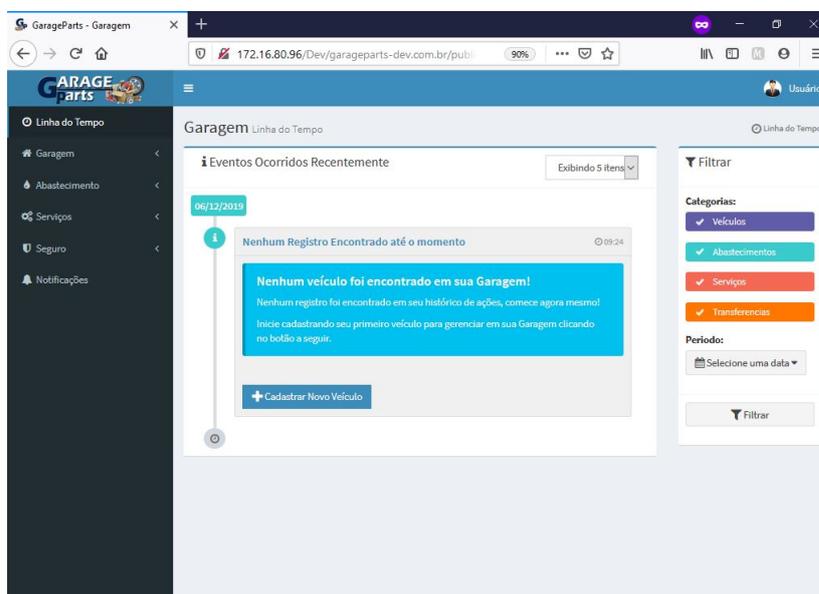


Fonte: Elaborado pelo próprio autor.

Figura 9 – Formulário de cadastro de novo usuário

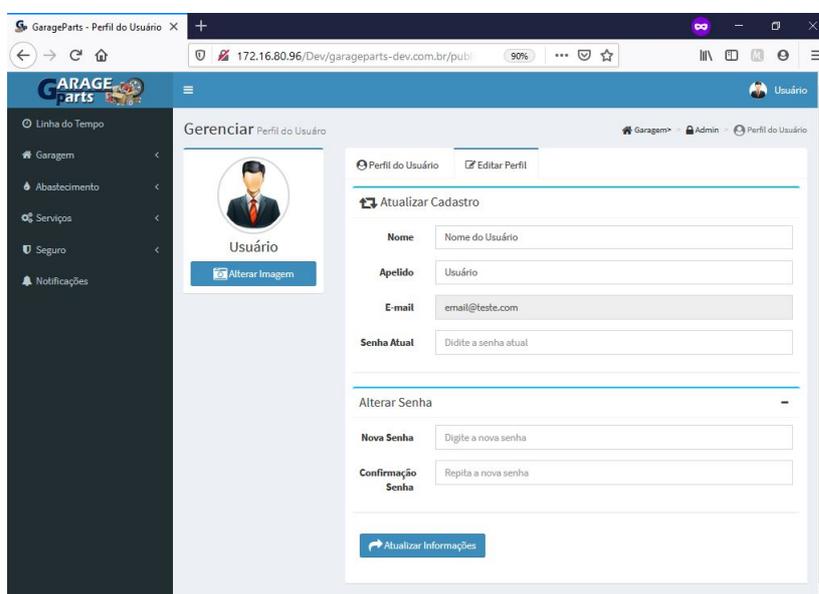
A interface apresentada ainda na [Figura 10](#) é intuitiva e autoexplicativa, o que torna a experiência de uso e navegação no sistema direta. A esquerda está posicionado um menu com as principais funcionalidades do sistema: Linha do Tempo; Garagem; Abastecimentos; Serviços; Seguro; e por fim, Notificações. Na parte superior, encontra-se a barra de navegação na cor azul, onde está posicionada a logomarca do sistema GarageParts que é um link para redirecionar o usuário para a tela principal, contendo a Linha do Tempo. À direita está posicionado um menu *dropdown* com a imagem do usuário e seu

nome de usuário cadastrado. Já a Figura 11 mostra a tela de gerenciamento do perfil de usuário.



Fonte: Elaborado pelo próprio autor.

Figura 10 – Tela principal da área restrita do usuário denominada Linha do Tempo.



Fonte: Elaborado pelo próprio autor.

Figura 11 – Tela do gerenciamento de perfil do usuário.

## 4.1 Cadastro e Gerenciamento de Veículos

Retornando a Timeline, conforme [Figura 10](#), uma mensagem é exibida no corpo da interface, pois o usuário ainda não possui nenhum veículo cadastrado. Para usufruir de fato dos recursos proporcionados pelo sistema o primeiro passo é cadastrar um veículo, para tanto, o usuário poderá clicar no botão abaixo da mensagem ou utilizar o menu "Garagem > Cadastrar Veículos".

O usuário será redirecionado para o formulário referente ao cadastro das informações do veículo. A [Figura 12](#) mostra os campos necessários para a realização do cadastro do veículo. Ao selecionar o tipo do veículo, o sistema retorna uma lista com as marcas dos fabricantes presentes na Tabela Fipe. De forma similar, após selecionar a marca, todos os modelos de veículos também são preenchidos conforme dados presentes na Tabela. O sistema faz checagem quanto a duplicação de placas e do código renavan.

A imagem mostra a interface de usuário para o cadastro de um novo veículo. O formulário contém os seguintes campos:

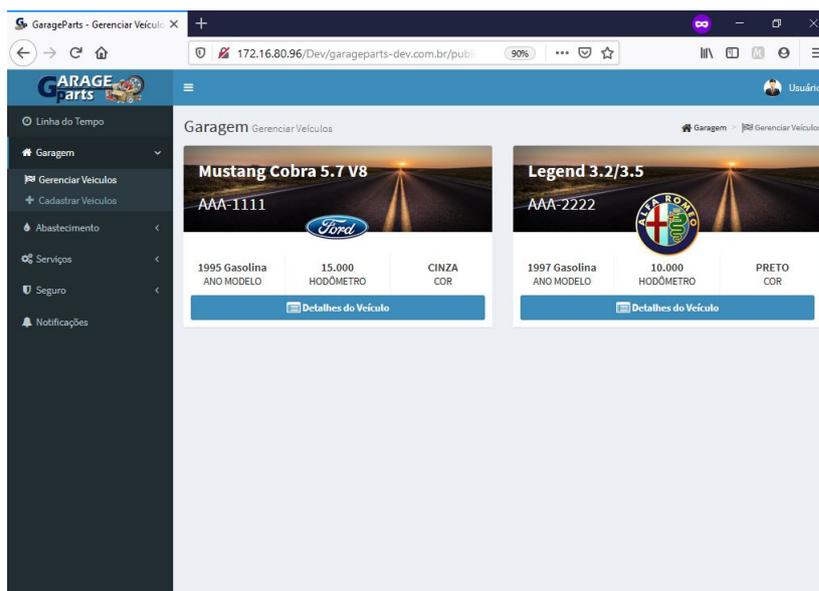
- Tipo do veículo\***: Menu suspenso para selecionar o tipo de veículo.
- Marca e Modelo**: Seção com subcampos:
  - Marca\***: Menu suspenso para selecionar a montadora.
  - Modelo\***: Menu suspenso para selecionar o modelo.
  - Ano do Modelo\***: Menu suspenso para selecionar o ano do modelo.
- Informações do Veículo**: Seção com subcampos:
  - Placa\***: Campo de texto com o exemplo "Ex: AAA-9999".
  - Renavan**: Campo de texto com o placeholder "Digite o renavan".
  - Cor\***: Campo de texto com o placeholder "Cor do veículo".
- Informações da Aquisição**: Seção com subcampos:
  - Hodômetro\***: Campo de texto com o placeholder "Digite o Km inicial do veículo".

Fonte: Elaborado pelo próprio autor.

Figura 12 – Formulário de cadastro do veículo.

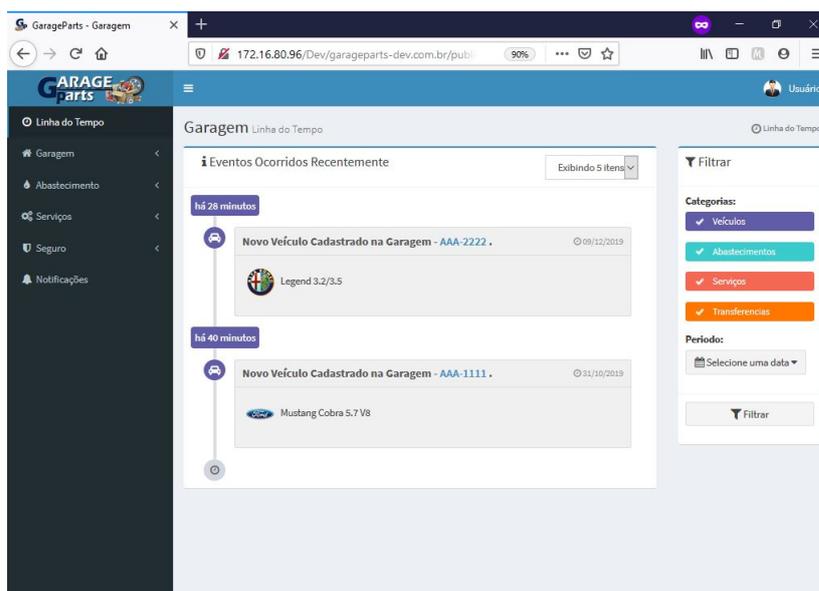
A [Figura 13](#) exibe a interface referente ao menu "Garagem > Gerenciar Veículos". Na opção "Detalhes do Veículo" pode-se selecionar as funcionalidades referente ao gerenciamento de cada veículo de forma particular.

Após realizar o cadastro do veículo, os primeiros eventos são gerados na Linha do Tempo e ficam dispostos em ordem decrescente (do mais recente para o mais antigo). É possível filtrar os eventos da Linha do Tempo pela categoria desejada ([Figura 14](#)).



Fonte: Elaborado pelo próprio autor.

Figura 13 – Tela do gerenciamento de Veículos.



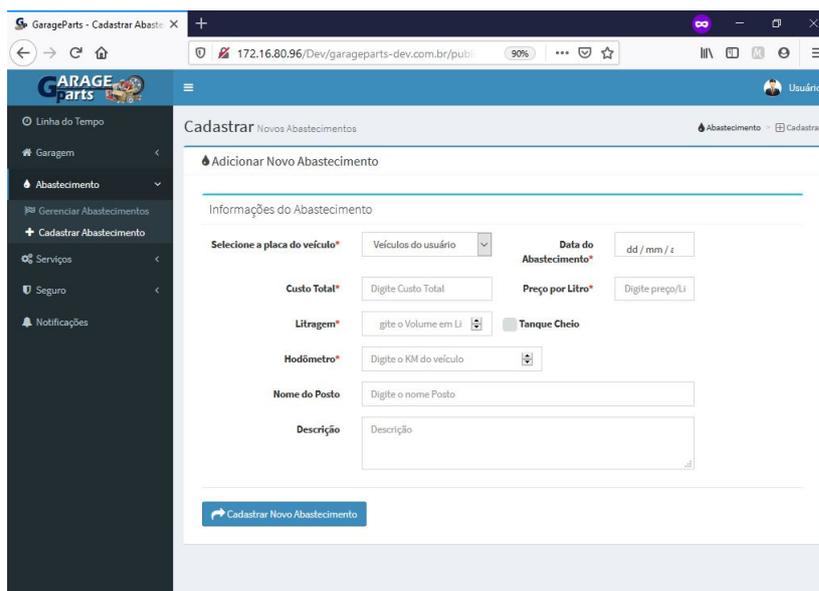
Fonte: Elaborado pelo próprio autor.

Figura 14 – Linha do tempo com eventos.

## 4.2 Cadastro e Gerenciamento de Abastecimentos

Com ao menos um veículo cadastrado, é possível incluir novos abastecimentos. No menu "Abastecimentos > Cadastrar Abastecimento". A [Figura 15](#) exibe a interface do formulário e as informações necessárias para inserir registros de abastecimento.

Cada abastecimento é realizado para um veículo, em uma data. Também são

The image shows a web browser window displaying the 'GarageParts' application. The page title is 'Cadastrar Novo Abastecimento'. The interface is in Portuguese and features a dark sidebar on the left with navigation options like 'Garagem', 'Abastecimento', 'Gerenciar Abastecimentos', 'Cadastrar Abastecimento', 'Serviços', 'Seguro', and 'Notificações'. The main content area is titled 'Cadastrar' and contains a form for adding a new refueling record. The form fields include: 'Selecione a placa do veículo\*' (a dropdown menu), 'Data do Abastecimento\*' (a date picker), 'Custo Total\*' (a text input), 'Preço por Litro\*' (a text input), 'Litragem\*' (a text input with a unit selector), 'Tanque Cheio' (a checkbox), 'Hodômetro\*' (a text input with a unit selector), 'Nome do Posto' (a text input), and 'Descrição' (a text area). A blue button at the bottom of the form is labeled 'Cadastrar Novo Abastecimento'.

Fonte: Elaborado pelo próprio autor.

Figura 15 – Tela formulário de cadastro de abastecimento.

necessárias as informações do custo por litro, da quantidade de litros abastecidos, a quilometragem atual do hodômetro. Pode-se informar se o tanque foi completo e, de maneira opcional, o nome do posto e uma descrição do mesmo. Ao submeter as informações, o sistema realiza uma verificação da quilometragem informada, e caso ela seja maior que a quilometragem que consta no cadastro do veículo, a mesma será atualizada, deste modo será possível acompanhar o avanço de quilometragem dos veículos gerenciados.

A [Figura 16](#) exibe uma lista dos abastecimentos já realizados e suas informações. Nesta interface é possível filtrar os abastecimentos cadastrados através do painel.

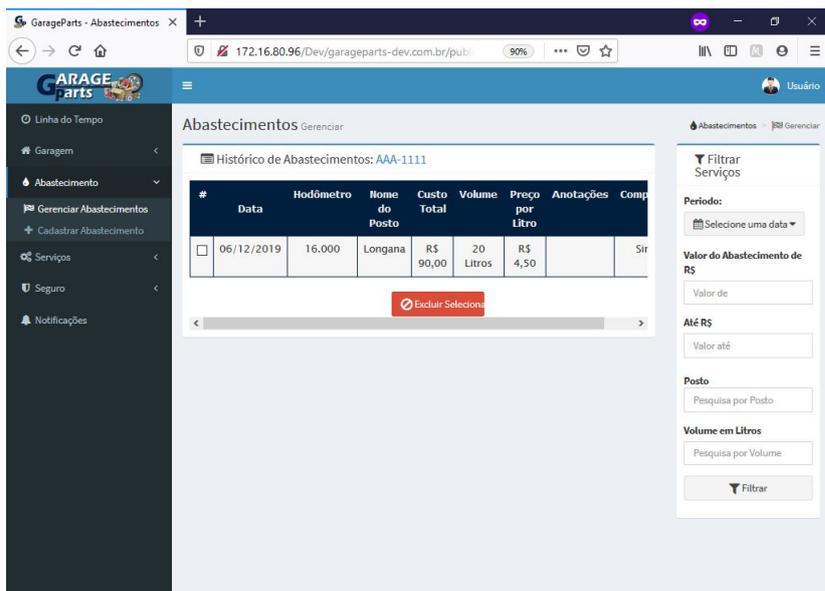
### 4.3 Cadastro e Gerenciamento de Serviços

Serviços podem ser cadastrados a qualquer momento para um veículo. Os serviços podem ser de dois tipos: normal ou agendado. Para incluir um serviço normal ou já executado as informações requeridas são as mesmas, com exceção do item "Agendado" disponível no formulário de cadastro de serviços.

Caso a opção "Agendado" seja selecionada, é gerado um registro na tabela de pendências, relacionando o usuário, o veículo e o serviço a ser executado. As pendências de um veículo podem ser acessadas no menu "Notificações" ([Figura 17](#)).

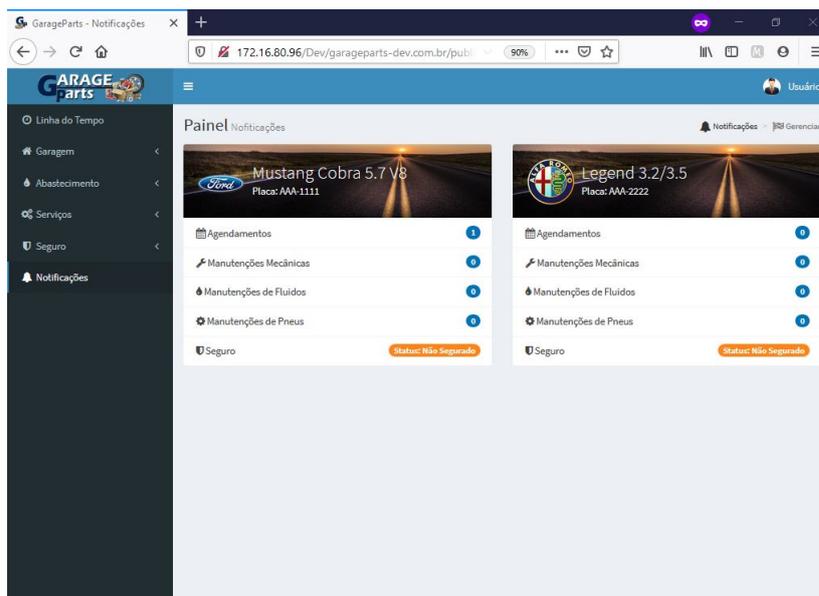
Após a efetivação do cadastro de um serviço, é necessário que seja feito o acesso a tela de detalhamento do serviço, no menu "Serviços > Gerenciamento de Serviços.

Para cada serviço é possível anexar uma nota fiscal (formato .pdf) e editar as



Fonte: Elaborado pelo próprio autor.

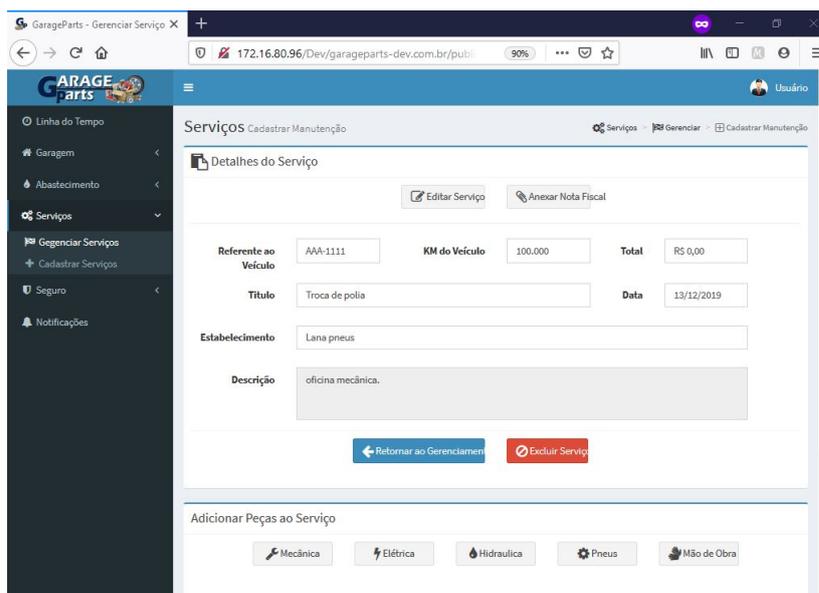
Figura 16 – Tela gerenciamento de abastecimentos.



Fonte: Elaborado pelo próprio autor.

Figura 17 – Tela gerenciamento de notificações.

informações do próprio serviço. Por fim, as manutenções devem ser inseridas obedecendo as categorias já existentes. A cada manutenção cadastrada, o valor do serviço é atualizado, levando em consideração o custo de cada peça multiplicado pela quantidade de peças, vide [Figura 18](#).



Fonte: Elaborado pelo próprio autor.

Figura 18 – Tela cadastro e gerenciamento de manutenções do serviços.

## 4.4 Cadastro e Gerenciamento de Seguros

O gerenciamento de seguro não permite que mais de um contrato de seguro esteja ativo para um mesmo veículo. Portanto, é possível cadastrar novos seguros apenas para veículos sem qualquer tipo de contrato.

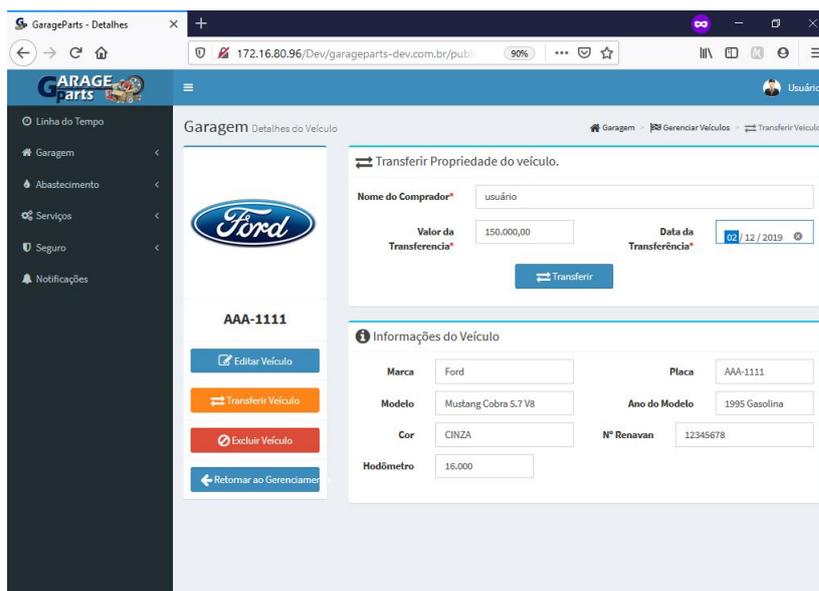
O registro de um novo seguro gera uma confirmação do gerenciamento de notificações, e exibe o *status* do seguro, ou como ativo ou como expirado. Caso seja detectado um seguro próximo do vencimento, um *link* para a informação de renovação do seguro é disponibilizado no painel de notificações (Figura 17).

## 4.5 Transferência de Veículos

O sistema GarageParts implementa a funcionalidade de transferência de veículos, não presente nas ferramentas estudadas na fase *concepção* e *elaboração*. Essa funcionalidade realiza a transferência de propriedade de um veículo, com todo o histórico de informações de serviços, abastecimentos, seguros e agendamentos, para um outro usuário do sistema.

O proprietário do veículo, exclusivamente, é quem pode realizar a transferência em si. A opção "Transferir Veículo", presente nas telas de detalhamento ou edição dos veículos cadastrados, inicia o procedimento. Então, é requerido o nome do usuário para o qual a transferência será realizada. Portanto, o usuário comprador também deverá estar cadastrado no GarageParts.

A Figura 19 exibe as informações solicitadas para transferência.



Fonte: Elaborado pelo próprio autor.

Figura 19 – Interface de transferência do veículo, seleção do comprador.

Quando a transferência é realizada, o veículo e todas as informações a ele vinculadas são enviadas para o novo proprietário, e o antigo proprietário não consegue mais visualizar as informações.

## 5 Resultados

O presente trabalho apresentou o processo de desenvolvimento do sistema GarageParts, um *software* capaz de registrar e gerenciar as informações de gastos e consumo de veículos, além de manter e transferir informações históricas entre os proprietários, auxiliando na gestão do controle de manutenções e comprovação da procedência dos automóveis cadastrados. O código fonte do sistema pode ser visto no repositório GarageParts<sup>1</sup>.

Para atender aos objetivos específicos, foram observadas as características de: automatização de tarefas manuais; e o armazenamento persistente dos dados dos veículos dos usuários, por meio de uma ferramenta *web* gratuita e responsiva.

Na fase de transição não foram realizados testes sistematizados da Engenharia de Software disponíveis na literatura. No entanto com a finalidade de garantir que os objetivos deste trabalho tenham sido alcançados, foram executados testes de funcionalidade, teste alpha e posteriormente, testes beta com um grupo de usuários. Os testes foram conduzidos de forma exaustiva e os resultados relatados pelos usuários serviram para evidenciar possíveis erros nas funcionalidades do sistema e inconformidades dos requisitos propostos.

Com o *software* desenvolvido e após ser testado, pode-se concluir que o objetivo de automatização de tarefas manuais e persistência de dados foram satisfeitas, assim como as funcionalidades que possibilitam a realização dessas tarefas, como o cadastro e gerenciamento de veículos, abastecimentos, seguros e serviços; o gerenciamento de notificações para serviços agendados ou com autonomia e validade prevista; galeria de fotos e transferência da propriedade dos veículos gerenciados.

O GarageParts também atende ao requisito de ser responsivo e compatível com os principais navegadores do mercado, característica obtida com o auxílio do AdminLTE. Este comportamento do sistema permite que o GarageParts possa ser acessado a partir de qualquer dispositivo do mercado equipado com um navegador de internet, tornando-o uma ferramenta multiplataforma.

Por outro lado, o GarageParts por ser um sistema *web* requer conexão com a Internet para que seja possível acessar e utilizar de suas funcionalidades, o que pode ser facilmente contornável nos dias atuais, através do uso de *Internet* móvel como o caso das conexões 4G ou similares, disponíveis para computadores, *tablets* e *smartphones*.

Por fim para consolidar a proposta, especificações e projeto do sistema proposto neste trabalho é apresentada a Tabela 5 que compara as funcionalidades dos sistemas já apresentados na sessão 2.2 e o GarageParts.

---

<sup>1</sup> <https://github.com/JoseWilsonJr/GarageParts>

Tabela 5 – Comparação entre as aplicações analisadas e o GarageParts.

Comparação entre sistemas do mercado				
Características	Habil	Drivvo	Carango	GarageParts
Tipo de licença	Gratuito (2 veículos)	Gratuito com funcionalidades limitadas e anúncios	Gratuito com funcionalidades limitadas e anúncios	Gratuito com funcionalidades ilimitadas e sem anúncios
Plataforma	<i>Windows</i>	<i>Android e iOS</i>	<i>Android</i>	<i>Web</i>
Limite de veículos cadastrados	2 veículos ( <i>trial</i> )	Ilimitado	Ilimitado	Ilimitado
Controle de abastecimentos	Sim	Sim	Sim	Sim
Controle de quilometragem dos veículos	Sim	Sim	Sim	Sim
Controle de manutenções	Sim	Sim	Sim	Sim
Lembretes / Pendencias	Sim	Sim	Sim	Sim
Consulta valor Tabela FIPE	Não	Não	Não	Sim
Exportar dados da aplicação	Versão Completa	Versão PRO	Versão PRO	Não
Cadastro de receitas	Não	Sim	Não	Não
<i>Backup</i> das informações da aplicação	Versão Completa	Versão PRO	Versão PRO	Sim
Galeria de Fotos	Sim	Não	Não	Sim
Tranferência de veículos e informações históricas entre usuários	Não	Não	Não	Sim

Fonte: Elaborado pelo próprio autor.

## 5.1 Trabalhos Futuros

Mediante a finalização das atividades deste trabalho, foi possível observar algumas possibilidades de expansão e aprimoramento para o sistema desenvolvido.

Como trabalho futuro é possível que o sistema seja submetido a testes sistematizados da Engenharia de Software, para que sejam testados o desempenho da ferramenta em situações que haja uso massivo e simultâneo da ferramenta, afim de medir o desempenho do sistema em situações de uso em larga escala.

O GarageParts foi inicialmente idealizado como uma ferramenta para uso em caráter não corporativo, portanto, a iniciativa do desenvolvimento *web* teve o propósito de tornar a ferramenta compatível com qualquer computador ou dispositivo móvel. Porém, por este motivo, há a necessidade destes dispositivos estarem sempre conectados a Internet quando seus usuários estiverem fazendo uso da ferramenta. Neste contexto, uma possível melhoria seria o desenvolvimento de um aplicativo *mobile* para contornar a falta de Internet em localidades remotas, possibilitando o cadastro das informações referentes aos veículos em modo *offline*, e posteriormente, a sincronização destes dados quando o dispositivo estiver conectado .

Fugindo um pouco do escopo deste trabalho, outra possibilidade visionária seria o desenvolvimento de uma rede social com o foco em automóveis e temas relacionados. Neste contexto, o objeto principal desta rede seriam os veículos de seus usuários. O objetivo seria criar um ambiente para atender o nicho de mercado voltado para o comércio de carros, peças, serviços e colecionadores, de modo a prover a interação, a troca de experiências e compartilhamento de conteúdo entre seus usuários em um ambiente específico para este fim.

# Referências

- ADMINLTE. *Documentação AdminLTE*. 2019. Disponível em: <<https://adminlte.io/themes/AdminLTE/documentation/index.html>>. Acesso em: 30 out 2019. Citado na página 15.
- ANDROID. *O Sistema Operacional móvel mais conhecido do mundo*. 2019. Disponível em: <<https://www.mysql.com/about/legal/licensing/oem/>>. Acesso em: 23 dez 2019. Citado na página 18.
- APPLE. *iOS13. Muitas novidades para ver. Incluindo um jeito novo de ver*. 2019. Disponível em: <<https://www.apple.com/br/ios/ios-13/>>. Acesso em: 23 dez 2019. Citado na página 18.
- ASTAH. *Astah Pro/UML/Viewer Ver8.2 Released!* 2019. Disponível em: <<http://astah.net/>>. Acesso em: 30 out 2019. Citado na página 22.
- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. *UML: guia do usuário*. [S.l.]: Elsevier Brasil, 2006. Citado na página 27.
- CANDIDO, C. H. *Ferramenta de Modelagem de Bancos de Dados Relacionais brModelo v3*. 2017. Disponível em: <<http://www.sis4.com/brModelo/>>. Acesso em: 06 dez 2019. Citado na página 22.
- CARANGO. *Carango - Controle de Veículos*. 2019. Disponível em: <[https://play.google.com/store/apps/details?id=br.com.carango&hl=pt\\_BR](https://play.google.com/store/apps/details?id=br.com.carango&hl=pt_BR)>. Acesso em: 30 out 2019. Citado 2 vezes nas páginas 18 e 20.
- DRIVVO. *Drivvo - Gerencie seu veículo (Gastos e Receitas)*. 2019. Disponível em: <<http://www.drivvo.com/pt/>>. Acesso em: 30 out 2019. Citado 2 vezes nas páginas 18 e 19.
- ELMASRI, R.; NAVATHE, S. B. *Sistemas de banco de dados*. [S.l.]: Pearson Addison Wesley São Paulo, 2005. Citado na página 16.
- FORTUNA, D. *FIPE API HTTP REST*. 2019. Disponível em: <<http://deividfortuna.github.io/fipe/>>. Acesso em: 23 dez 2019. Citado 2 vezes nas páginas 16 e 17.
- HÁBIL. *Hábil Pessoal + Veículos*. 2019. Disponível em: <<http://www.habilpessoal.com.br/registro.asp>>. Acesso em: 30 out 2019. Citado 2 vezes nas páginas 17 e 18.
- LARAVEL. *Laravel - The PHP Framework for Web Artisans*. 2019. Disponível em: <<https://laravel.com/>>. Acesso em: 30 out 2019. Citado na página 15.
- LAUDON, K. C.; LAUDON, J. P. *Sistemas de informação gerenciais*. [S.l.]: Pearson Prentice Hall, 2015. ISBN 978-85-4300-585-0. Citado 2 vezes nas páginas 14 e 15.
- MYSQL. *Commercial License for OEMs, ISVs and VARs*. 2019. Disponível em: <<https://www.mysql.com/about/legal/licensing/oem/>>. Acesso em: 30 out 2019. Citado na página 16.

- 
- REZENDE, D. A. *Engenharia de software e sistemas de informação*. [S.l.]: Brasport, 2006. Citado na página 28.
- SOMMERVILLE, I. *Engenharia de software*. [S.l.]: Pearson, 2011. Citado na página 14.
- WAZLAWICK, R. S. *Análise e Projeto de Sistemas de Informação Orientados a Objetos*. [S.l.]: Elsevier, 2004. Citado 3 vezes nas páginas 16, 22 e 23.
- WERNKE, R. Custo total de propriedade de automóveis de uso particular: aplicação comparativa em dois modelos populares (1.0). *Revista Brasileira de Contabilidade*, n. 211, p. 30–45, 2015. Citado na página 12.