

UNIVERSIDADE FEDERAL DO PARANÁ

ANDRE EDMIR MOREIRA RADOVANSKI  
GIOVANNE MOMESSO  
INGRID GOMES GUILHEM  
LUIZA ARAÚJO ZAPPELINI

AidAuto  
SISTEMA GERENCIADOR AUTOMOTIVO

CURITIBA  
2016

UNIVERSIDADE FEDERAL DO PARANÁ

ANDRE EDMIR MOREIRA RADOVANSKI  
GIOVANNE MOMESSO  
INGRID GOMES GUILHEM  
LUIZA ARAÚJO ZAPPELINI

AidAuto  
SISTEMA GERENCIADOR AUTOMOTIVO

Trabalho apresentado como requisito parcial à  
obtenção do grau de tecnólogo em Análise e  
Desenvolvimento de Sistemas, Setor de  
Educação Profissional e Tecnológica da  
Universidade Federal do Paraná.

Orientadora: Profa. Rafaela Mantovani Fontana

CURITIBA  
2016

## TERMO DE APROVAÇÃO

ANDRE EDMIR MOREIRA RADOVANSKI  
GIOVANNE MOMESSO  
INGRID GOMES GUILHEM  
LUIZA ARAÚJO ZAPPELINI

AidAuto  
SISTEMA GERENCIADOR AUTOMOTIVO

Trabalho de conclusão de curso aprovado como requisito parcial para obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas pela Universidade Federal do Paraná.

### BANCA EXAMINADORA

---

Professora Rafaela Mantovani Fontana  
Departamento de Informática UFPR  
Professora Orientadora

---

Professor Luis Antonio Pereira Neves  
Departamento de Informática UFPR

---

Professor Pedro Rodrigues Torres Junior  
Departamento de Informática UFPR

Curitiba  
2016

## **AGRADECIMENTOS**

Agradecemos a todos os profissionais do meio acadêmico pelos conhecimentos passados, em especial, a Professora e Orientadora Rafaela por todo o incentivo, apoio e auxílio durante todo o processo de desenvolvimento do projeto.

Agradecemos também aos nossos pais, familiares e amigos pelo amor, paciência e tolerância nos momentos de ausência.

## RESUMO

A manutenção preventiva em veículos é importante, pois evita acidentes causados por falhas em equipamentos. Ajudar o condutor a identificar o momento certo de realizar tal preservação, monitorar o desempenho de seu veículo e controlar gastos são, também, atividades fundamentais. Levando em conta as funcionalidades de aplicativos concorrentes no mercado, foi desenvolvido um aplicativo gestor automotivo na plataforma Android para *Smartphones* que ajuda o usuário a manter o funcionamento correto do veículo, por meio de notificações de manutenções, monitoramento do desempenho por meio de dados fornecidos e controle de gastos. Possui diferenciais como: reconhecimento de imagem, planejamento de viagem e cálculo do melhor combustível. A metodologia de desenvolvimento utilizada foi o Scrum. O projeto é dividido em *Sprints*, com reuniões de planejamento, reuniões diárias, de revisão do *Sprint* e retrospectiva. O resultado alcançado foi um aplicativo para dispositivos móveis que ajuda o usuário a manter um bom funcionamento do veículo e estar em dia com os pagamentos, conforme o objetivo proposto no início do projeto.

## **ABSTRACT**

*Vehicle preventive maintenance is important, because it prevents accidents caused by equipment failure. To Help the driver to identify the exact moment to perform such preservation, monitor the performance of your vehicle and control expenses are, also, fundamental activities. Taking into account the features of competitors currently on the market, this work was developed an automotive manager application in the Android platform for smartphones that will help the user to keep the right operation of the vehicle, by notifying the maintenances, monitoring the performance by data provided and control of expenses. It has some differences such as: image recognition, trip planning and calculating the best fuel. The developing methodology used was Scrum. The project was divided into sprints, with a planning meeting, daily meetings, sprints review and a retrospective. The result achieved was an application for mobile devices that helps the user to keep a good operation of the vehicle and be on time with payments, according to the objective proposed in the beginning of the project.*

## LISTA DE FIGURAS

Figura 1 - ARQUITETURA ANDROID EM CAMADAS FONTE: ADAPTADO DE DESENVOLVEDORES ANDROID(2012) .....	15
Figura 2 - FLUXOGRAMA METODOLOGIA DE DESENVOLVIMENTO. FONTE: AUTORIA PRÓPRIA.....	21
Figura 3 – FLUXOGRAMA DE TELAS FONTE: AUTORIA PRÓPRIA .....	32
Figura 4 - TELA INICIAL .....	33
Figura 5 – TELAS MENU .....	34
Figura 6 – TELAS MANTER VEÍCULO.....	36
Figura 7 - TELAS MANTER CONDUTOR.....	37
Figura 8 - TELAS ABASTECIMENTO.....	38
Figura 9 - TELA DE RECONHECIMENTO DE IMAGEM .....	39
Figura 10 - TELAS DE MANUTENÇÕES EVENTUAIS .....	41
Figura 11 - TELA DE ESCOLHA DO TIPO DE MANUTENÇÃO PROGRAMADA .....	42
Figura 12 - TELAS DE MANUTENÇÕES PROGRAMADAS .....	43
Figura 13 - TELAS DE DESPESAS EXTRAS .....	44
Figura 14 - TELAS DE LOCAIS .....	45
Figura 15 - TELAS DE IMPOSTOS.....	46
Figura 16 - TELAS DE SEGURO.....	47
Figura 17 - TELAS DE FINANCIAMENTO .....	49
Figura 18 - TELAS DE MULTAS.....	50
Figura 19 - TELAS DE TELEFONES ÚTEIS.....	51
Figura 20 - TELAS DE VIAGEM .....	52
Figura 21 - TELA DE LISTA DE ABASTECIMENTOS DE VIAGEM.....	53
Figura 22 - TELA DE LISTA DE DESPESAS EXTRAS DE VIAGEM .....	54
Figura 23 - TELA DE FECHAR VIAGEM .....	55
Figura 24 - TELAS DE RELATÓRIO.....	56
Figura 25 - TELA MELHOR COMBUSTIVEL .....	57
Figura 26 - PROTÓTIPO TELA INICIAL E MENU.....	77
Figura 27 - PROTÓTIPO TELA MANTER VEÍCULO .....	79
Figura 28 - PROTÓTIPO TELA MANTER CONDUTOR .....	80
Figura 29 - PROTÓTIPO TELA CADASTRAR ABASTECIMENTO.....	81
Figura 30 - PROTÓTIPO TELAS MANTER MANUTENÇÕES EVENTUAIS .....	82
Figura 31 - PROTÓTIPO TELAS MANTER MANUTENÇÕES PROGRAMADAS .....	84
Figura 32 - PROTÓTIPO TELAS MANTER IMPOSTOS.....	85
Figura 33 - PROTÓTIPO TELAS MANTER DESPESAS EXTRAS .....	86
Figura 34 - PROTÓTIPO TELAS MANTER SEGURO .....	87
Figura 35 - PROTÓTIPO TELAS MANTER FINANCIAMENTO .....	88
Figura 36 - PROTÓTIPO TELAS MANTER MULTAS .....	89
Figura 37 - PROTÓTIPO TELA RELATÓRIO .....	90
Figura 38 - PROTÓTIPO TELAS CADASTRAR VIAGEM.....	91
Figura 39 - PROTÓTIPO TELAS GERENCIAR VIAGEM.....	92
Figura 40 - PROTÓTIPO TELAS MANTER TELEFONES ÚTEIS .....	93
Figura 41 - PROTÓTIPO TELAS MANTER LOCAIS .....	94
Figura 42 - PROTÓTIPO TELA MELHOR COMBUSTÍVEL .....	95
Figura 43 - RESULTADO TESTE CADASTRAR VEÍCULO .....	96

Figura 44 - RESULTADO TESTE CADASTRAR CONDUTOR .....	96
Figura 45 - RESULTADO TESTE CADASTRAR MANUTENÇÃO PROGRAMADA ....	97
Figura 46 - RESULTADO TESTE MANUTENÇÃO EVENTUAL.....	97
Figura 47 - RESULTADO TESTE CADASTRAR ABASTECIMENTO .....	97
Figura 48 - RESULTADO TESTE CADASTRAR DESPESA EXTRA .....	97
Figura 49 - RESULTADO TESTE CADASTRAR IMPOSTO .....	97
Figura 50 - RESULTADO TESTE CADASTRAR MULTA.....	97
Figura 51 - RESULTADO TESTE CADASTRAR SEGURO .....	97
Figura 52 - RESULTADO TESTE CADASTRAR FINANCIAMENTO .....	97
Figura 53 - RESULTADO TESTE CADASTRAR LOCAL .....	97
Figura 54 - RESULTADO TESTE CADASTRAR TELEFONES ÚTEIS .....	97
Figura 55 - RESULTADO TESTE INICIAR VIAGEM.....	97
Figura 56 - RESULTADO TESTE FECHAR VIAGEM .....	97
Figura 57 - RESULTADO TESTE MELHOR COMBUSTÍVEL .....	97
Figura 58 - RESULTADO TESTE GERAR RELATÓRIO .....	97

## LISTA DE TABELAS

Tabela 1 - TABELA DE COMPARAÇÃO DAS FUNCIONALIDADES. FONTE: AUTORIA PRÓPRIA.....	19
Tabela 2 - LINHA DO TEMPO DAS SPRINTS. AUTORIA: PRÓPRIA.....	27

## Sumário

1	INTRODUÇÃO.....	11
1.1	OBJETIVO.....	12
1.1.1	OBJETIVO GERAL.....	12
1.1.2	OBJETIVOS ESPECÍFICOS.....	12
1.2	JUSTIFICATIVA .....	13
1.3	ESTRUTURA DO DOCUMENTO .....	14
2	FUNDAMENTAÇÃO TEÓRICA .....	15
2.1	O que é o Android.....	15
2.1.1	Arquitetura do Android.....	15
2.2	Como construir um aplicativo em Android.....	16
2.3	Aplicativos de Manutenção Preventiva de Veículos .....	17
2.3.1	Carrorama .....	17
2.3.2	RoadTrip.....	18
2.3.3	Fuelio .....	18
2.3.4	FuelBuddy .....	19
3	METODOLOGIA.....	21
3.1	Requisitos.....	22
3.2	Metodologia de Análise .....	23
3.2.1	Diagrama de Caso de Uso.....	23
3.2.2	Diagrama de Classes .....	23
3.2.3	Diagrama de Sequência .....	24
3.2.4	Modelagem de Dados.....	24
3.3	Metodologia de Desenvolvimento do Projeto.....	24
3.3.1	Divisão das <i>Sprint</i> .....	27
3.4	Tecnologia.....	30
3.4.1	Android Studio: .....	30
3.4.2	SQLite: .....	30
3.4.3	Linguagem de Programação Java: .....	30
3.4.4	Tesseract OCR:.....	30
4	APRESENTAÇÃO DO APLICATIVO .....	32
4.1	Tela Inicial .....	33
4.2	Menu .....	34
4.3	Manter Veículo .....	34
4.4	Manter Condutor.....	36

4.5	Abastecimento.....	37
4.6	Manutenções Eventuais.....	40
4.7	Manutenções Programadas.....	41
4.8	Despesas Extras .....	43
4.9	Locais .....	44
4.10	Imposto.....	46
4.11	Seguro.....	47
4.12	Financiamento .....	48
4.13	Multas.....	49
4.14	Telefones.....	50
4.15	Viagens .....	51
4.16	Relatório .....	55
4.17	Melhor Combustível.....	56
5	CONSIDERAÇÕES FINAIS.....	58
6	REFERÊNCIAS BIBLIOGRÁFICAS.....	60
	APÊNDICE A – REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS.....	63
	APÊNDICE B – DIAGRAMA GERAL DE ATIVIDADES .....	64
	APÊNDICE C – DIAGRAMA DE CASOS DE USO .....	67
	APÊNDICE D – DIAGRAMA DE CLASSE .....	68
	APÊNDICE E – DIAGRAMA DE SEQUÊNCIA - MANTER VEÍCULO .....	69
	APÊNDICE F – DIAGRAMA DE SEQUÊNCIA - MANTER VIAGEM.....	70
	APÊNDICE G – DIAGRAMA DE SEQUÊNCIA - FECHAR VIAGEM.....	71
	APÊNDICE H – DIAGRAMA DE SEQUÊNCIA – DESPESAS VIAGEM.....	72
	APÊNDICE I – DIAGRAMA DE SEQUÊNCIA – RECONHECIMENTO DE IMAGEM ..	73
	APÊNDICE J – DIAGRAMA DE SEQUÊNCIA – RELATÓRIO.....	74
	APÊNDICE K – DIAGRAMA ENTIDADE-RELACIONAMENTO .....	75
	APÊNDICE L – PROTÓTIPOS DE TELA: APLICATIVO AIDAUTO .....	77
	APÊNDICE M – DESCRIÇÃO DOS CASOS DE USO .....	78
	APÊNDICE N – TESTES CAIXA PRETA.....	96

## 1 INTRODUÇÃO

A manutenção preventiva em automóveis tem como objetivo minimizar as chances de acidentes causados por desgaste de peças. Isso porque, quando os itens como pneus, rodas e freios estão desgastados, o risco de acidentes aumenta para passageiros e motoristas.

De acordo com o Departamento Estadual de Trânsito do Paraná (Detran-PR), os riscos de acidentes em carros que não fazem manutenções periódicas é três vezes maior que o normal. Um exemplo de situação agravada pela falta de manutenção é o travamento do motor ocasionado pelo rompimento da correia dentada do veículo. Este componente é responsável pela sincronia de peças no motor, controlando a entrada e saída de gases do cilindro (DETRAN, 2016).

Segundo matéria publicada na Revista SóCarrão, se a troca da peça acontece de forma preventiva, possui um custo aproximado de R\$ 150,00 em um carro popular. No entanto, quando ocorre o rompimento com o veículo em movimento, problemas em outras peças relacionadas podem acontecer, como por exemplo, danos nos pistões, cabeçote, emperramentos, entre outros, chegando a um valor médio de R\$ 3000,00. (SÓCARRÃO, 2009).

Além da redução de custos devido à troca antecipada de peças, outras vantagens em fazer manutenção preventiva são: aumento da vida útil do veículo, melhora no desempenho e funcionamento, maior segurança e o aumento da confiabilidade, mantendo seu funcionamento e prevenindo futuros acidentes. Estas revisões planejadas também proporcionam vantagens na economia de combustível, visto que as condições de alguns itens de revisão são relacionadas a estes quesitos.

Portanto, adotar uma estratégia para se realizar a manutenção periódica e estabelecer um período de manutenção são atividades chave para a redução de custos e manutenção da confiabilidade operacional, cujo

cálculo leva em consideração a probabilidade de um equipamento funcionar normalmente em um período determinado de tempo. (CITISYSTEMS, 2016).

Com base nessa necessidade, a proposta deste trabalho é criar um aplicativo para dispositivos móveis que auxilie no gerenciamento das atividades do veículo, por meio de notificações, prevenindo incidentes causados por falta de manutenção.

No mercado tecnológico já existem aplicativos que são gestores automotivos, porém muitos deles não oferecem as funcionalidades necessárias, e os que oferecerem, são pagos. Pensando nisso, desenvolveu-se um aplicativo gratuito que auxilia o usuário a acompanhar e gerenciar o desempenho de seu veículo, e se diferencia por oferecer planejamento de gastos, reconhecimento de imagem na leitura do odômetro e acompanhamento de manutenções preventivas.

## **1.1 OBJETIVO**

### **1.1.1 OBJETIVO GERAL**

Desenvolver um aplicativo para *smartphones*<sup>1</sup> a fim de ajudar o usuário a manter o funcionamento correto do veículo, por meio de notificações de manutenções, monitoramento do desempenho através de dados fornecidos, controle de gastos, gerenciamento de custos em viagens mediante cadastro de despesas e abastecimentos realizados ao longo do trajeto.

### **1.1.2 OBJETIVOS ESPECÍFICOS**

---

<sup>1</sup> Smartphone: telefone inteligente com tecnologia avançada que inclui programas executados equivalente aos computadores

- Avaliar as interfaces e as funcionalidades de aplicativos automotivos disponíveis no mercado;
- Identificar os requisitos necessários para a construção de um novo aplicativo para gestão de manutenção de veículos, se adequando ao perfil dos usuários finais;
- Determinar as principais funcionalidades, considerando suas relevâncias;
- Analisar periodicidade sobre manutenções em veículos.

## 1.2 JUSTIFICATIVA

Muitos acidentes veiculares ocorrem todos os dias pela falta de manutenção. Para diminuir a probabilidade que isso ocorra é recomendado realizar manutenções preventivas, reduzindo riscos de acidente para motorista, passageiros e outros ao redor.

Em matéria publicada no jornal Gazeta do Povo, Fernanda Leitóles ressalta a importância de manter as manutenções em dia. Para evidenciar esse problema a Ecovia, concessionária de pedágio, realizou uma pesquisa onde constatou que nas principais rodovias do Paraná, BR-277, PRs 407 e 508, a média de veículos com circulação interrompida, devido à falha técnica, é de 72 veículos por dia, ou seja, três veículos por hora (GAZETA DO POVO, 2010).

Por isso há necessidade de se gerenciar atividades importantes do veículo, como por exemplo, a troca de óleo, filtro de ar, calibragem e reposição de pneus, aumentando significativamente a segurança e garantindo um bom funcionamento do automóvel.

Além da segurança, outro fator relevante é o controle de gastos. Calcular o consumo de combustível, despesas com impostos e obter uma visão total dos custos pode ajudar a gerenciar despesas e prever um futuro gasto com o veículo.

Tendo em vista esses fatores, a proposta deste projeto é o desenvolvimento de um aplicativo para *smartphone* Android que facilite o gerenciamento de manutenção preventiva e acompanhe as despesas do carro.

Pretende atender às expectativas dos usuários, fornecendo, além dos dados de consumo por quilometragem, o momento exato da manutenção, reduzindo riscos.

### **1.3 ESTRUTURA DO DOCUMENTO**

Este trabalho está dividido em cinco capítulos, incluindo este introdutório, contendo o objetivo geral, objetivos específicos e justificativa. No capítulo 2, Fundamentação Teórica, são expostos os conteúdos e teorias que embasaram o desenvolvimento do projeto, assim como uma análise e comparação dos concorrentes do mercado mais populares. O capítulo 3, Metodologia, apresenta o método utilizado para o desenvolvimento deste projeto, incluindo informações detalhadas sobre a metodologia ágil de desenvolvimento, requisitos funcionais e não funcionais diagramas de análise e de modelagem de dados. No capítulo 4, Apresentação do sistema, há o manual de instalação e manual do usuário, trazendo todas as informações necessárias para utilização do aplicativo. No capítulo 5, Considerações Finais, é apresentado o resumo do trabalho, os objetivos atingidos, questões relevantes e sugestões a trabalhos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem como objetivo apresentar os conteúdos e teorias que embasaram a elaboração do projeto proposto, introduzir o conceito da plataforma utilizada no sistema, e apresentar outros *softwares*<sup>2</sup> com o mesmo propósito do aplicativo desenvolvido.

### 2.1 O que é o Android

Android é o nome de um sistema operacional de código aberto, baseado em uma versão do Kernel Linux. Esse sistema foi desenvolvido para operar em dispositivos móveis, como celulares (*smartphones*), *netbooks*<sup>3</sup> e *tablets*<sup>4</sup>. Ele é distribuído pela loja GooglePlay e desenvolvido pela OpenHandset Alliance, uma aliança entre várias empresas, dentre elas a Google (BORDIN, 2012).

#### 2.1.1 Arquitetura do Android

O sistema Android possui uma arquitetura baseada em camadas. De Aplicações, Camada de *Frameworks*<sup>5</sup> de Aplicações, Bibliotecas, Android em Tempo Real e Núcleo Linux, conforme Figura 1.

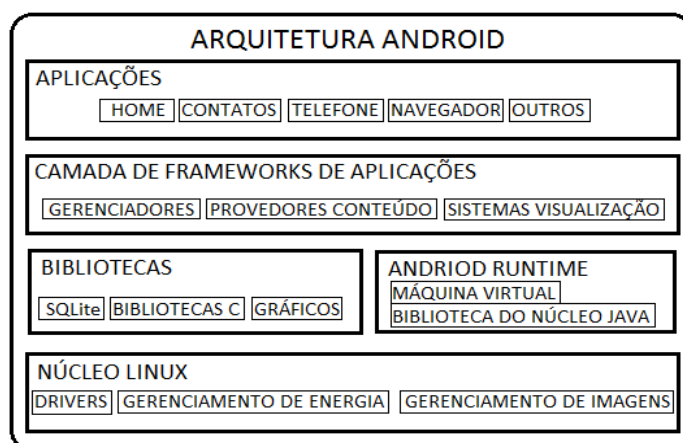


Figura 1 - ARQUITETURA ANDROID EM CAMADAS  
FONTE: ADAPTADO DE DESENVOLVEDORES ANDROID (2012)

<sup>2</sup> Software: conjunto de componentes lógicos de um computador ou sistema de processamento de dados

<sup>3</sup> Netbook: computador portátil

<sup>4</sup> Tablet: computador portátil pequeno com tela sensível ao toque

<sup>5</sup> Frameworks: junção de códigos genéricos comuns entre projetos de software

A camada Aplicação é onde encontram-se as aplicações do dispositivo, como por exemplo: *Home*<sup>6</sup>, Contatos, Telefone, Navegador entre outros.

Logo abaixo, existe a camada de *Frameworks* de aplicações, escrita na linguagem Java. Esta fornece as funcionalidades para criação de aplicativos, como por exemplo, o gerenciamento de atividades, janelas, pacotes, telefone, recursos, localização e notificação; provedores de conteúdo, sistemas de visualização e serviços XMPP (DESENVOLVEDORES ANDROID, 2012).

A terceira camada é a de Bibliotecas, que possui: banco de dados SQLite, suporte a formatos de áudio, vídeo e imagens, camada de segurança SSL, Gráficos 2D e 3D, rede de internet, biblioteca de linguagem C (LIBC) e de fontes.

As aplicações Java do Android são executadas em uma máquina virtual, localizada na camada Android em tempo de execução. Nesta camada localizam-se as bibliotecas padrão do núcleo Java e a Máquina Virtual Dalvik.

Conforme Figura 1, o Android tem como base da sua arquitetura o Linux Kernel (Núcleo do Linux), provedor de serviços para as aplicações executadas, onde encontram-se os *Drivers*<sup>7</sup>, gerenciamento de energia e de memória.

## 2.2 Como construir um aplicativo em Android

Para desenvolver um aplicativo é necessário possuir o Kit Desenvolvedor Java JDK (*Java Development Kit*) versão 5 ou superior. O Kit de Desenvolvedor de *Software* Android SDK (*Software Development Kit*) contém as ferramentas e emuladores que permitem o desenvolvimento e teste dos aplicativos. O SDK pode ser obtido por meio do site <<http://developer.android.com/adk/index.html>>.

O Android Studio, que é um Ambiente de Desenvolvimento Integrado, conhecido como IDE (*Integrated Development Environment*), foi utilizado para desenvolver o aplicativo proposto, pois possui um editor de código que auxilia na maior produtividade do desenvolvedor de aplicativos Android. Além disso, também oferece compatibilidade com módulos no Android Studio. A ferramenta

---

<sup>6</sup> Home: página inicial

<sup>7</sup> Driver: controlador de dispositivo

já é pré-configurada com uma imagem do emulador otimizada. O Gerenciador de dispositivos virtual atualizado e simplificado fornece perfis de dispositivos pré-definidos para dispositivos Android comuns (DESENVOLVEDORES ANDROID, 2012).

Uma vez apresentada a estrutura utilizada no desenvolvimento do aplicativo AidAuto, serão expostos na seção 2.3 outros *softwares* com o mesmo propósito.

## 2.3 Aplicativos de Manutenção Preventiva de Veículos

Como já especificado no capítulo 1, há diversos tipos de aplicativos similares no mercado que têm como objetivo gerenciar funcionalidades automotivas. Para esta análise foram escolhidos quatro aplicativos com maior número de *downloads*<sup>8</sup>: Carrorama, FuelBuddy, Fuelio e Road Trip. O método de análise foi embasado em suas funcionalidades. Segue abaixo uma breve descrição de cada aplicativo concorrente:

### 2.3.1 Carrorama

Este aplicativo requer a criação de um *login*<sup>9</sup> possibilitando o cadastro de mais de um carro para o mesmo motorista, e permite mais de um motorista para o mesmo veículo.

O aplicativo possui um esquema de menus para navegar entre as opções de benefícios, dicas e utilitários. Na opção de compra é possível adquirir uma versão sem propaganda. As opções de dicas são direcionadas a segurança no trânsito.

Dentro do ícone “utilitários”, há a possibilidade de identificar qual o melhor combustível baseado na relação custo x benefício, por meio de um cálculo simples que independe da marca e do modelo do veículo.

---

<sup>8</sup> Download: Ato de fazer cópia de uma informação que se encontra em um computador remoto

<sup>9</sup> Login: autenticação através de usuário e senha

Também é possível informar os valores gastos com abastecimento, manutenções, impostos e seguro. Essas informações devem ser inseridas manualmente sem crítica de consistência de dados.

Todas as informações podem ser visualizadas por meio do relatório de histórico, e o usuário deve informar a validade das manutenções para receber um alerta de quando será necessária a próxima manutenção.

### **2.3.2 RoadTrip**

O aplicativo RoadTrip não requer *login*<sup>12</sup>. No primeiro acesso é necessário informar os dados do veículo, incluindo a quantidade de quilômetros registrados no odômetro, definir os parâmetros de unidade de moeda e combustível. Finalizando o cadastro do veículo com a criação de um apelido.

As opções de menu são limitadas aos dados de consumo de combustível. A cada abastecimento é necessário informar a data do abastecimento, odômetro, litros abastecidos, valor pago por litro e o tipo de combustível.

O aplicativo calcula o valor total do abastecimento e informa, por meio de um relatório, a média de consumo dos últimos 100 quilômetros percorridos. As informações podem ser visualizadas via gráfico de barras ou em tabela.

### **2.3.3 Fuelio**

O aplicativo Fuelio não requer *login* nem dados do veículo. As informações são cadastradas baseadas em um veículo exemplo. Na sequência é possível renomear o veículo padrão para um apelido.

Entre as opções de menu, é possível cadastrar o consumo de combustível e também outras informações de despesas que servirão de base para informar por meio de um relatório histórico as estatísticas de consumo na forma de tabela ou na forma de um gráfico de linha mostrando a evolução do consumo ao longo do tempo. À medida que o rendimento de combustível piora, o proprietário percebe que está na hora de fazer manutenção.

O sistema se diferencia ao oferecer uma sessão de mapas onde mostra os locais de abastecimento salvos para que seja possível comparar o rendimento de combustível de cada posto.

No menu configurações é possível informar as unidades de moeda, formato de data e distância.

#### 2.3.4 FuelBuddy

O aplicativo FuelBuddy não requer *login* nem dados do veículo. As informações são cadastradas baseadas em um veículo exemplo. Na sequência é possível alterar o nome do veículo padrão.

No primeiro acesso é necessário informar os campos que serão preenchidos, bem como as respectivas unidades. As informações do odômetro podem ser inseridas por reconhecimento de voz ou de digitação.

As funcionalidades estão divididas no menu de abastecimento, serviço, compras e viagens. Os dados históricos serão mostrados por meio de um relatório de consumo médio. Dentro da opção de serviços é possível cadastrar lembretes para realização de manutenções (troca de óleo e balanceamento).

Na opção de compras devem ser informadas as despesas de impostos e estacionamento para que o sistema possa gerar um relatório dos custos do veículo. Em ambas as opções o aplicativo irá enviar mensagens com lembretes de manutenções para o e-mail cadastrado.

A Tabela 1, a seguir, as comparações entre as principais funcionalidades dos aplicativos descritos e o proposto neste trabalho, nomeado AidAuto.

Nome do App	Login	Cadastrar veículo	Cadastrar condutor	Cadastrar abastecimento	Cadastrar manutenções	Gerar relatório	Reconhecimento OCR	Planejamento de viagem
Carrorama	✓	✓	✓	✓	✓	✓	✗	✗
FuelBuddy	✗	✓	✗	✓	✓	✗	✗	✗
Fuelio	✗	✓	✗	✓	✗	✓	✗	✗
RoadTrip	✗	✓	✗	✓	✗	✗	✗	✗
AidAuto	✗	✓	✓	✓	✓	✓	✓	✓

Tabela 1 - TABELA DE COMPARAÇÃO DAS FUNCIONALIDADES. FONTE: AUTORIA PRÓPRIA

A Tabela 1 mostra que o aplicativo desenvolvido apresenta, além das funcionalidades em comum, um diferencial no reconhecimento de imagem, planejamento e divisão de custos de viagem.

Para auxiliar no desenvolvimento do projeto, elaborou-se um diagrama que representa fluxo das principais atividades do aplicativo, tais como: cadastrar um veículo, cadastrar abastecimento, inserir despesas, cadastrar manutenções, cadastrar locais, cadastrar planejamento de viagem, gerar relatório e calcular o melhor combustível. Este diagrama encontra-se no APÊNDICE B – DIAGRAMA GERAL DE ATIVIDADES.

Este capítulo apresentou a arquitetura Android utilizada para o desenvolvimento do aplicativo. Ademais, realizou-se uma análise dos concorrentes de mercado, bem como uma comparação entre os mesmos.

O capítulo seguinte tem a proposta de apresentar a metodologia utilizada para o processo de construção do projeto, divisão de entrega e atividades propostas, bem como seus diagramas e requisitos.

### 3 METODOLOGIA

Este capítulo tem por objetivo apresentar a metodologia utilizada no desenvolvimento do projeto e seus requisitos, bem como o planejamento utilizado para a concepção do aplicativo.

O método utilizado para a construção deste projeto foi o Scrum, que se iniciou com o levantamento de requisitos funcionais e não funcionais, a partir da análise de sistemas já existentes, entrevistas e estudo do manual de veículos de diferentes marcas e modelos. Conforme a Figura 2, a metodologia de desenvolvimento consiste em levantar e especificar as funcionalidades necessárias para o desenvolvimento do projeto, sendo necessário implementá-las e validá-las.

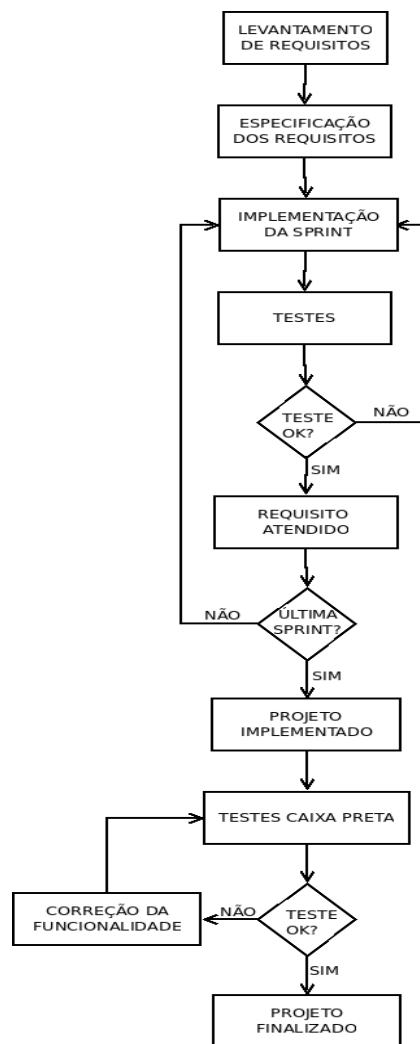


Figura 2 - FLUXOGRAMA METODOLOGIA DE DESENVOLVIMENTO. FONTE: AUTORIA PRÓPRIA

As funcionalidades especificadas foram divididas em oito etapas de desenvolvimento, onde em cada etapa foram realizados testes para comprovar a correta implementação dos requisitos. Caso os testes sejam satisfatórios e o resultado obtido conforme esperado, a próxima etapa de desenvolvimento é iniciada. Caso contrário, a etapa deve ser reimplementada.

Após a finalização destas etapas, a implementação do projeto é finalizada. A partir deste ponto, apenas testes de caixa preta são realizados na interface do *software*, examinando as funcionalidades do sistema sem se preocupar com a estrutura interna (PRESSMAN, 1995). Os testes caixa preta acima descritos localizam-se no APÊNDICE N – TESTES CAIXA PRETA.

### **3.1 Requisitos**

O levantamento de requisitos foi realizado por duas diferentes estratégias: (a) Análise de sistemas existentes e (b) Estudo de manuais de veículos.

#### **a) Análise dos sistemas já existentes**

Foi realizada a análise de quatro sistemas diferentes, RoadTrip, Carrorama, FuelBuddy e Fuelio como base para o desenvolvimento do aplicativo que se adequasse às necessidades dos veículos, conforme apresentado no capítulo 2 seção 2.3.

#### **b) Estudo de manuais de veículos**

O manual do proprietário possui informações importantes para o funcionamento correto do veículo, como por exemplo: as datas de revisões que variam de modelo para modelo, as marcas de peças e produtos recomendados pelo fabricante, e até saber identificar avisos no painel como óleo e temperatura fora do normal.

Com base no levantamento de requisitos, foram identificados os principais requisitos funcionais e não funcionais para o sistema proposto, descritos no APÊNDICE A – REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS.

## 3.2 Metodologia de Análise

Para o auxílio da visualização das funcionalidades do *software*, foi utilizada uma Linguagem Unificada de Modelagem, a UML (em inglês Unified Modeling Language) ao longo das *sprints*<sup>10</sup> de desenvolvimento.

A linguagem UML possui diferentes tipos de diagramas, que, em conjunto, determinam uma visão geral e aspectos do sistema (MARTINEZ, 2015).

Foram criados os seguintes diagramas UML: Diagrama de Caso de Uso, Diagrama de Classe, Diagrama de Sequência e Diagrama de Atividades. Foi criado também a representação física do banco de dados.

### 3.2.1 Diagrama de Caso de Uso

O diagrama de caso de uso que se encontra no APÊNDICE C – DIAGRAMA DE CASOS DE USO descreve a interação entre o usuário e as funcionalidades do sistema, documentando os requisitos funcionais, ou seja, descreve o que o sistema faz do ponto de vista do usuário (RIBEIRO, 2015).

As especificações encontram-se no APÊNDICE M – DESCRIÇÃO DOS CASOS DE USO.

### 3.2.2 Diagrama de Classes

O diagrama de classes, localizado no APÊNDICE D – DIAGRAMA DE CLASSE, tem como principal objetivo mostrar as classes do sistema com seus atributos e métodos, além de mostrar o relacionamento entre elas, apresentando uma visão estática da organização das classes e sua estrutura lógica (SILVA, 2016).

---

<sup>10</sup> Sprint: representa um ciclo de trabalho no Scrum

### 3.2.3 Diagrama de Sequência

O diagrama de sequência é baseado no diagrama de caso de uso. Ele mostra a ordem temporal em que os eventos ocorrem, as condições que devem ser contempladas e a ordem dos métodos a serem disparados.

Os diagramas de sequência desse projeto são baseados nas descrições do caso de uso, nos quais são mostrados: Diagrama de sequência Manter veículo (APÊNDICE E – DIAGRAMA DE SEQUÊNCIA - MANTER VEÍCULO), Manter viagem (APÊNDICE F – DIAGRAMA DE SEQUÊNCIA - MANTER VIAGEM), Fechar viagem (APÊNDICE G – DIAGRAMA DE SEQUÊNCIA - FECHAR VIAGEM), Despesas viagem (APÊNDICE H – DIAGRAMA DE SEQUÊNCIA – DESPESAS VIAGEM) e Reconhecimento de imagens (APÊNDICE I – DIAGRAMA DE SEQUÊNCIA – RECONHECIMENTO DE IMAGEM).

### 3.2.4 Modelagem de Dados

A modelagem de dados explora as estruturas orientadas a dados, desde modelos conceituais de alto nível até modelos de dados e físicos, tornando-se essencial em qualquer projeto de *software*, pois é nesta fase que se identificam os tipos de entidades (NETO, 2014).

O diagrama entidade-relacionamento, também conhecido por DER, descreve a estrutura lógica do banco de dados. Utiliza-se de elementos como entidade (objeto do mundo real), atributos (características da entidade) e relacionamentos como as entidades relacionam-se (RODRIGUES, 2016).

O diagrama físico de entidade-relacionamento deste projeto se encontra no APÊNDICE K – DIAGRAMA ENTIDADE-RELACIONAMENTO

## 3.3 Metodologia de Desenvolvimento do Projeto

Conforme já citado, a técnica utilizada para o desenvolvimento deste projeto foi o Scrum.

O Scrum é um *framework* simples para gestão de projetos complexos, utilizando metodologia de desenvolvimento ágil. Possui uma abordagem iterativa

e incremental para otimizar a previsibilidade e controlar riscos. Essa ferramenta é utilizada para definir padrões de gestão e planejamento de projetos de software, acelerando o seu desenvolvimento a partir da melhoria contínua processo, mantendo uma comunicação em tempo real e subdividindo o projeto em etapas (SCHWABER; SUTHERLAND, 2013).

Esse *framework* sugere a divisão da equipe em papéis para aumentar a eficiência, criatividade e versatilidade; a realização de eventos para criar rotinas e reuniões mais definidas; e a utilização de artefatos que maximizam transparência entre os integrantes da equipe.

São três pilares fundamentais que apoiam a implementação do controle de processo: Transparência, Inspeção e Adaptação (SCRUM ALLIANCE, 2014).

- **Transparência:** dos procedimentos entre os integrantes da equipe, dos requisitos de entrega e do status do projeto, garantindo visibilidade dos processos que afetam o resultado final.
- **Inspeção:** inspeções constantes nas atividades, com frequência suficiente para ser possível detectar variações inaceitáveis. Essa inspeção também pode ocorrer no processo nas reuniões diárias e nas revisões.
- **Adaptação:** do processo e das mudanças do produto, preservado valores e práticas.

Os papéis do Scrum foram adaptados ao projeto da seguinte forma: desenvolvedores Android e analistas de documentos e diagramas. Desta forma, a equipe determinou a melhor maneira de realizar o trabalho e atingir a meta estabelecida.

O projeto é dividido em etapas, chamadas eventos. Estes possuem duração máxima, inspeções criteriosas, redução de reuniões e criação de rotinas com o objetivo de minimizar perdas no processo de planejamento. Estes são divididos em: *sprints*, reunião de planejamento, reunião diária, reunião de revisão do *sprint* e reunião de retrospectiva (SCHWABER; KEN, 2001).

De acordo com Schwaber (2013), estas reuniões otimizam a comunicação entre os membros da equipe, facilitando as tomadas de decisão e melhorando o nível

de conhecimento do time de desenvolvimento (SCHWABER; SUTHERLAND, 2013).

- *Sprints*: períodos de tempo em que alguns itens selecionados da lista de funcionalidades necessárias serão construídos e entregues. Uma nova *sprint* é iniciada após o término da *sprint* anterior. Neste projeto foram realizadas oito *sprints*, que serão descritas na sessão 3.3.1.
- Reuniões de planejamento: onde serão definidos os itens a serem desenvolvidos durante a *Sprint*. Estas foram realizadas semanalmente, com acompanhamento da professora orientadora.
- Reuniões diárias: evento com 15 minutos de duração com objetivo de analisar o progresso do dia anterior, onde os integrantes do time de desenvolvimento devem responder a três questões: O que foi feito desde a última reunião? O que será feito na próxima reunião? O que está impedindo a conclusão do trabalho?

Todos os dias, após a etapa desenvolvida, era realizada uma conferência em tempo real com todos os integrantes da equipe, para atualizar informações.

Os artefatos do Scrum servem para aumentar a transparência e possibilitar inspeção, gerando maior sucesso na entrega de cada etapa, divididos em *backlog*<sup>11</sup> do produto, *backlog* da *sprint* e o incremento do produto.

O *backlog* do produto é a lista de funcionalidades necessárias, requisitos e adaptações ordenadas por prioridade. É gerado na definição do projeto, e tem a função de descrever o trabalho previsto para o produto final.

Já o *backlog* da *sprint* é o planejamento do que será entregue no próximo incremento, implementado a cada duas semanas, tornando visível o trabalho a ser realizado pelo time.

Por fim, o incremento do produto é conjunto de itens desenvolvidos durante um *sprint*, que devem estar em condições de uso. Seu resultado avaliado pode ajudar nas próximas *sprints*.

---

<sup>11</sup> Backlog: resumo histórico

Levando em consideração a metodologia ágil acima descrita, a Tabela 2 - LINHA DO TEMPO DAS SPRINTS. AUTORIA: PRÓPRIA apresenta a linha do tempo de todas as *sprints* necessárias para a elaboração do projeto. A divisão das tarefas dentre os integrantes do grupo foi a seguinte: todo o grupo foi responsável pela primeira *sprint*, após esta André e Giovane ficaram responsáveis pela parte de desenvolvimento do projeto, a parte de programação, e Ingrid e Luiza com a parte escrita do trabalho. A seguir, a meta e tarefas desenvolvidas em cada *sprint* são apresentadas.

ATIVIDADE	29/01 a 22/03	23/03 a 01/04	02/04 a 15/04	16/04 a 29/04	30/04 a 13/05	14/05 a 27/05	28/05 a 10/06	11/06 a 17/06	18/06 a 28/06
SPRINT 1									
SPRINT 2									
SPRINT 3									
SPRINT 4									
SPRINT 5									
SPRINT 6									
SPRINT 7									
SPRINT 8									
ENTREGA									

Tabela 2 - LINHA DO TEMPO DAS SPRINTS. AUTORIA: PRÓPRIA

### 3.3.1 Divisão das *Sprint*

O projeto foi dividido em oito *sprints*, sendo definido uma meta, objetivo a ser alcançado ao final de cada etapa, conforme abaixo:

*Sprint* 1 - 29/01/2016 até 22/03/2016

- Escolha do tema do trabalho
- Planejamento das telas
- Definição do Design padrão
- Lista de Requisitos Funcionais
- Análise dos softwares concorrentes
- Pesquisa sobre manutenção preventiva de veículos
- Diagrama de Classe
- Introdução do documento
- Modelo lógico do banco de dados

- Diagrama de Casos de Uso
- Especificações dos casos de uso
- Planejamento macro das atividades do projeto
- Diagrama de Atividades

META: Definição do escopo.

*Sprint 2- 23/03/2016 até 01/04/2016*

- Capítulos 1 e 2 do documento
- Versão inicial do capítulo 3
- Implementação do layout do sistema
- CRUD<sup>12</sup> de veículo
- Protótipo das telas

META: Protótipo de telas.

*Sprint 3 - 02/04/2016 até 15/05/2016*

- CRUD Abastecimento (Sem OCR)
- CRUD Locais
- CRUD Telefones
- Diagrama de sequência (veículos)
- Versão Revisada capítulos 1 e 2

META: CRUD Abastecimento implementado.

*Sprint 4 - 16/04/2016 até 29/04/2016*

- CRUD Despesas
- CRUD Multas
- CRUD Financiamento

---

<sup>12</sup> CRUD: Create (criar), Read (ler), Update (atualizar) e Delete (excluir).

- CRUD Impostos
- Versão Revisada capítulo 3

META: Implementar CRUDS secundários.

*Sprint 5 - 30/04/2016 até 13/05/2016*

- Implementação OCR
- CRUD Condutor

META: Implementação OCR.

*Sprint 6 - 14/05/2016 até 27/05/2016*

- CRUD Manutenções
- CRUD Seguro
- CRUD Viagem
- Alertas
- Diagrama de Sequência (OCR)

META: CRUD Manutenção implementado.

*Sprint 7 - 28/05/2016 até 10/06/2016*

- Relatório
- Melhor Combustível
- Diagrama de sequência (Viagem, Relatório e Melhor Combustível)
- Versão final capítulos 1, 2 e 3
- Capítulos 4 e 5

META: Gerar relatório.

*Sprint 8 - 11/06/2016 até 17/06/2016*

- Revisão geral da documentação

- Versão final da parte escrita
- Testes do aplicativo

META: Testes do aplicativo.

## 3.4 Tecnologia

Nesta Seção serão descritas as tecnologias e ferramentas utilizadas no desenvolvimento do sistema.

### 3.4.1 Android Studio:

O Android Studio é um ambiente de desenvolvimento integrado para desenvolver para plataforma Android. É disponibilizado gratuitamente desde maio de 2013, porém a sua primeira compilação estável foi lançada em dezembro de 2014. Sua instalação é simples, bastando fazer o *download* e executar.

### 3.4.2 SQLite:

O SQLite é uma biblioteca em linguagem C que implementa um banco de dados SQL. Não é uma biblioteca de cliente usada para conectar em um grande servidor de banco de dados, é o próprio servidor. Essa biblioteca lê e escreve diretamente para e do arquivo do banco de dados no disco. Não possui licença, é totalmente disponibilizado pelos autores.

### 3.4.3 Linguagem de Programação Java:

Java é uma linguagem de programação orientada à objetos, desenvolvida na década de 90 na empresa Microsystems. É uma linguagem que possibilita a utilização de *frameworks* e ambientes para facilitar o desenvolvimento das aplicações. Possui uma comunidade de desenvolvedores, onde reúnem-se grupos denominados JUG's (*Java User Groups*, ou grupos de usuários Java).

### 3.4.4 Tesseract OCR:

Tesseract é uma biblioteca para reconhecimento ótico de caracteres, desenvolvido na linguagem C. Utiliza como entrada de dados uma imagem

binaria, realiza uma serie de análises, delimitações e o reconhecimento. Logo após, os caracteres são classificados. Possui código aberto, e está hospedado atualmente no GitHub (TESSERACT – OCR, 2016).

Esta aplicação contém o Tesseract Open Source OCR Engine. Originalmente desenvolvido na Hewlett Packard Laboratories Bristol e na Hewlett Packard Co, Greeley Colorado, todo o código nesta distribuição está agora licenciado sob a Licença Apache.

Este capítulo apresentou a metodologia de desenvolvimento do projeto e a divisão de tarefas em *sprints*, bem como os diagramas de análise e as tecnologias utilizadas. O próximo capítulo explicará o funcionamento das telas do aplicativo.



## 4.1 Tela Inicial

Esta tela possui *spinner*<sup>13</sup> de veículos para escolha das informações a serem exibidas, tais como: último abastecimento, com botão para cadastrar novo abastecimento; os alertas sendo: verde quando está para vencer com menos de 5 dias, amarelo quando está no dia de vencimento e vermelho para quando está vencido, ao clicar no botão ok o usuário indica que tal alerta foi atendido; e dicas de seguranças e manutenções, que ao clicar no botão atualizar uma nova dica irá aparecer como mostrado na Figura 4 - TELA INICIAL.

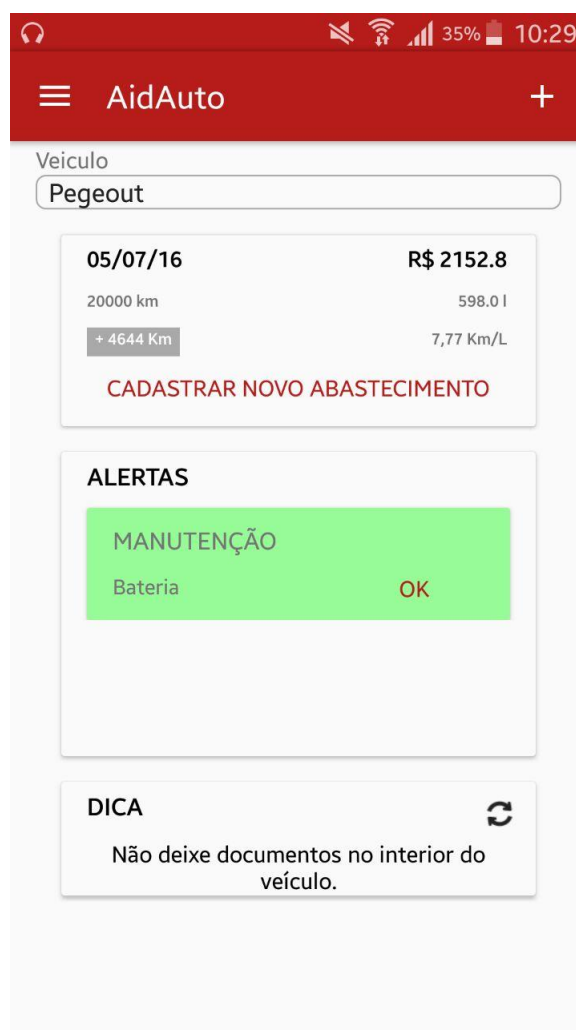


Figura 4 - TELA INICIAL

<sup>13</sup> Spinner: opções de escolha

## 4.2 Menu

O menu é aberto ao pressionar o ícone no canto superior esquerdo ou deslizando o dedo da extrema esquerda para a direita. Conforme mostrado na figura Figura 5 – TELAS MENU, ele é dividido em sub-menus contendo início, dados, gastos, combustível, manutenções, viagens e relatórios.

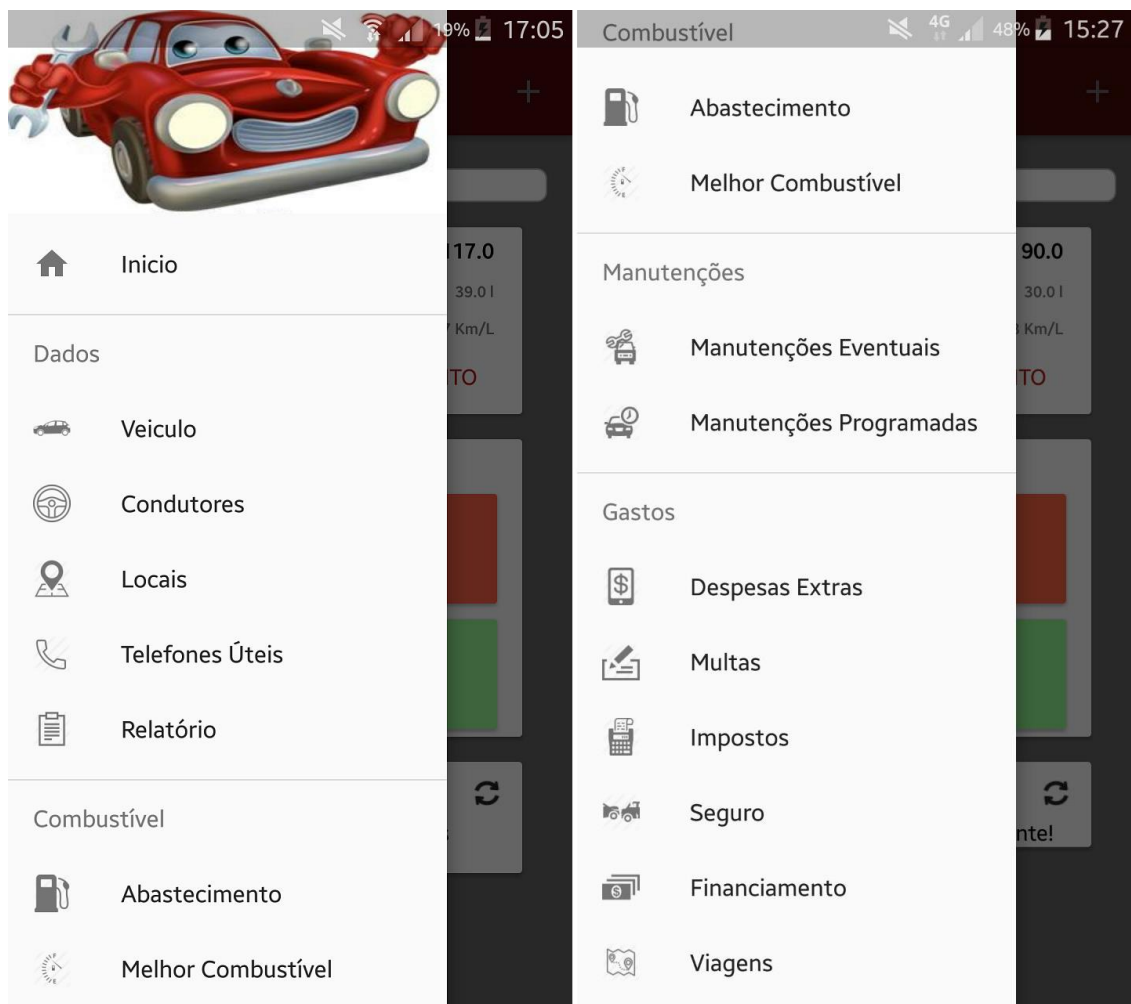


Figura 5 – TELAS MENU

## 4.3 Manter Veículo

Ao pressionar a opção dados-veículo, o sistema apresenta as informações referentes aos veículos cadastrados. Cada *card*<sup>14</sup> possui nome e ano do veículo, além das opções de editar e excluir.

Ao clicar no botão de adicionar, localizado no canto inferior direito, a tela de formulário é aberta. Nesta tela é possível adicionar o nome do veículo, marca,

<sup>14</sup> Card: bloco de tela, cartão

modelo, ano, placa, número do chassi, renavan, tipo de combustível, que é escolhido através de um *combobox*<sup>15</sup> com as opções: gasolina, etanol, diesel e *flex*<sup>16</sup>, e o tipo de veículo escolhido por *combobox*<sup>30</sup> com as opções: carro, motocicleta, caminhão e ônibus.

Para salvar as informações inseridas, basta pressionar o *check-mark*<sup>17</sup> localizado no canto superior direito. Após gravar os conteúdos inseridos, o sistema é redirecionado para a tela de lista.

Para editar as opções de veículo já salvo basta clicar no botão editar. Dessa forma, os campos são apresentados preenchidos e aptos a serem alterados. Ao concluir a dição, basta pressionar o botão *check-mark*.

Para excluir é necessário pressionar o botão excluir no *card* do veículo. Sendo assim, o sistema apaga permanentemente o registro. Como visto na figura Figura 6.

---

<sup>15</sup> Combobox: opções de escolha

<sup>16</sup> Flex: veículo com opção de combustível flexível

<sup>17</sup> Check-mark: marcador de validação

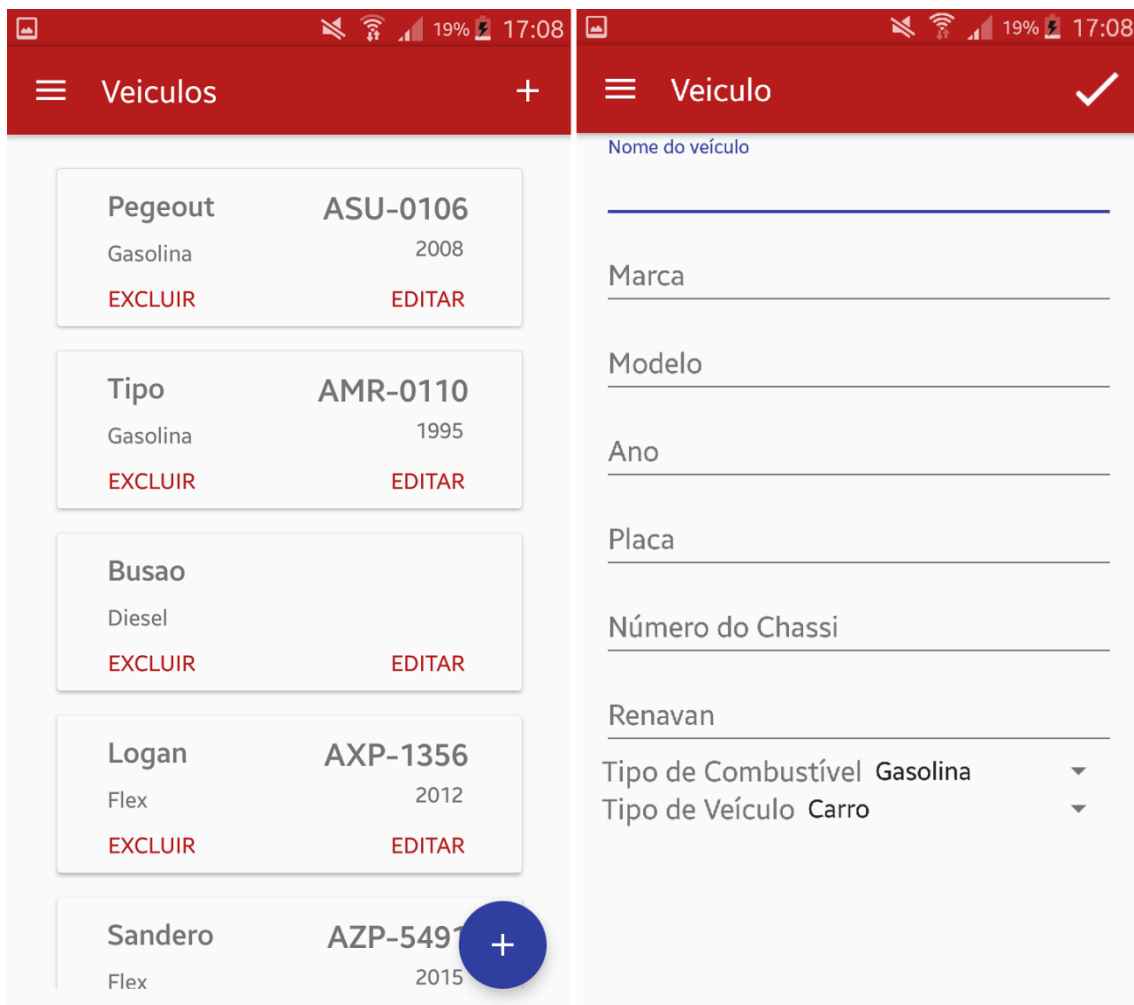


Figura 6 – TELAS MANTER VEÍCULO

#### 4.4 Manter Condutor

Ao pressionar a opção dados-condutor, o sistema apresenta as informações de condutores para o veículo que está em foco no topo da tela.

Para adicionar um condutor é necessário pressionar o botão adicionar, localizado no canto inferior esquerdo onde é encaminhado para a tela de formulário que o usuário deve preencher o nome do condutor, número do RG, número do CPF, data de nascimento, data da primeira habilitação, data de vencimento da CNH, número e categoria da CNH. O usuário também pode selecionar a opção alerta para o vencimento da habilitação.

Para editar os registros de um condutor já salvo basta clicar no botão editar. Neste caso, os campos são apresentados preenchidos e aptos a serem alterados. Ao concluir a edição, basta pressionar o botão *check-mark*.

Para excluir é necessário pressionar o botão excluir no *card* do condutor. Sendo assim, o sistema apaga permanentemente o registro. Segue abaixo a Figura 7 com as telas descritas acima.

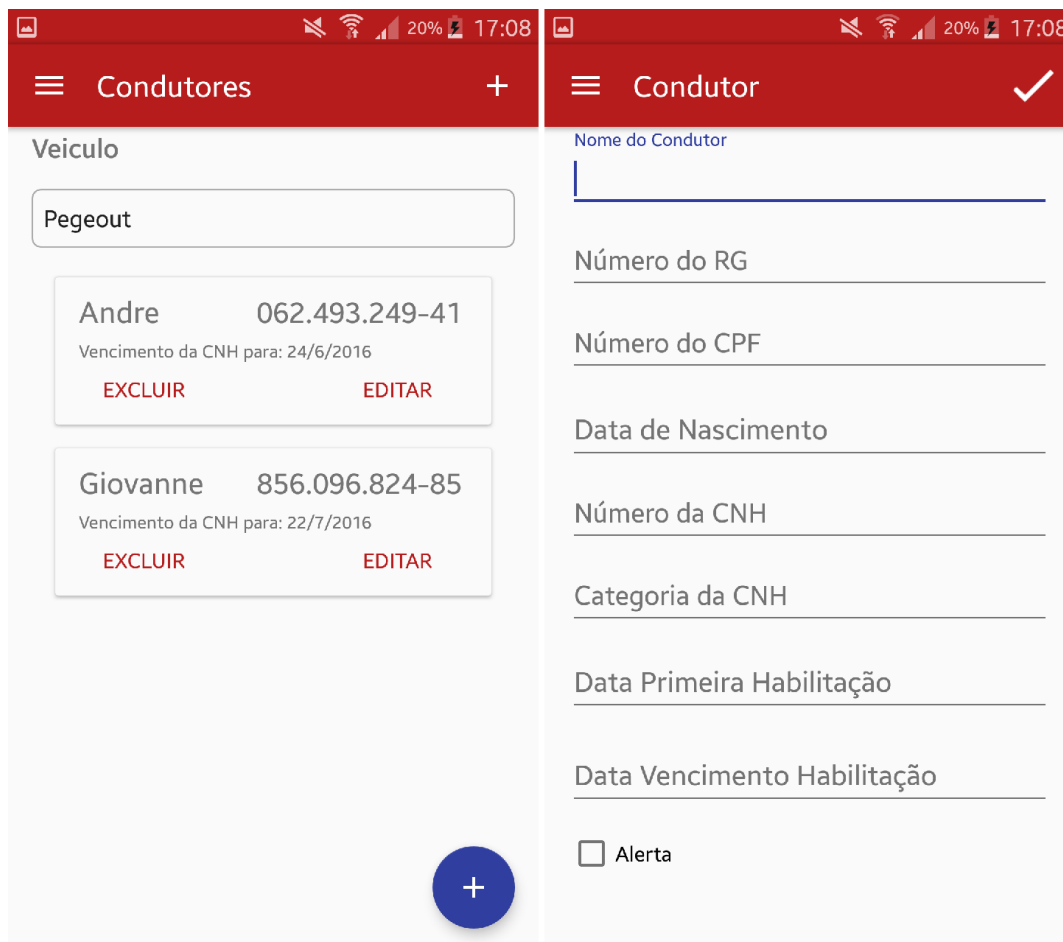


Figura 7 - TELAS MANTER CONDUTOR

## 4.5 Abastecimento

Ao pressionar a opção combustível-abastecimento, o sistema apresenta as informações de todos os abastecimentos cadastrados, como mostrado na figura Figura 8.

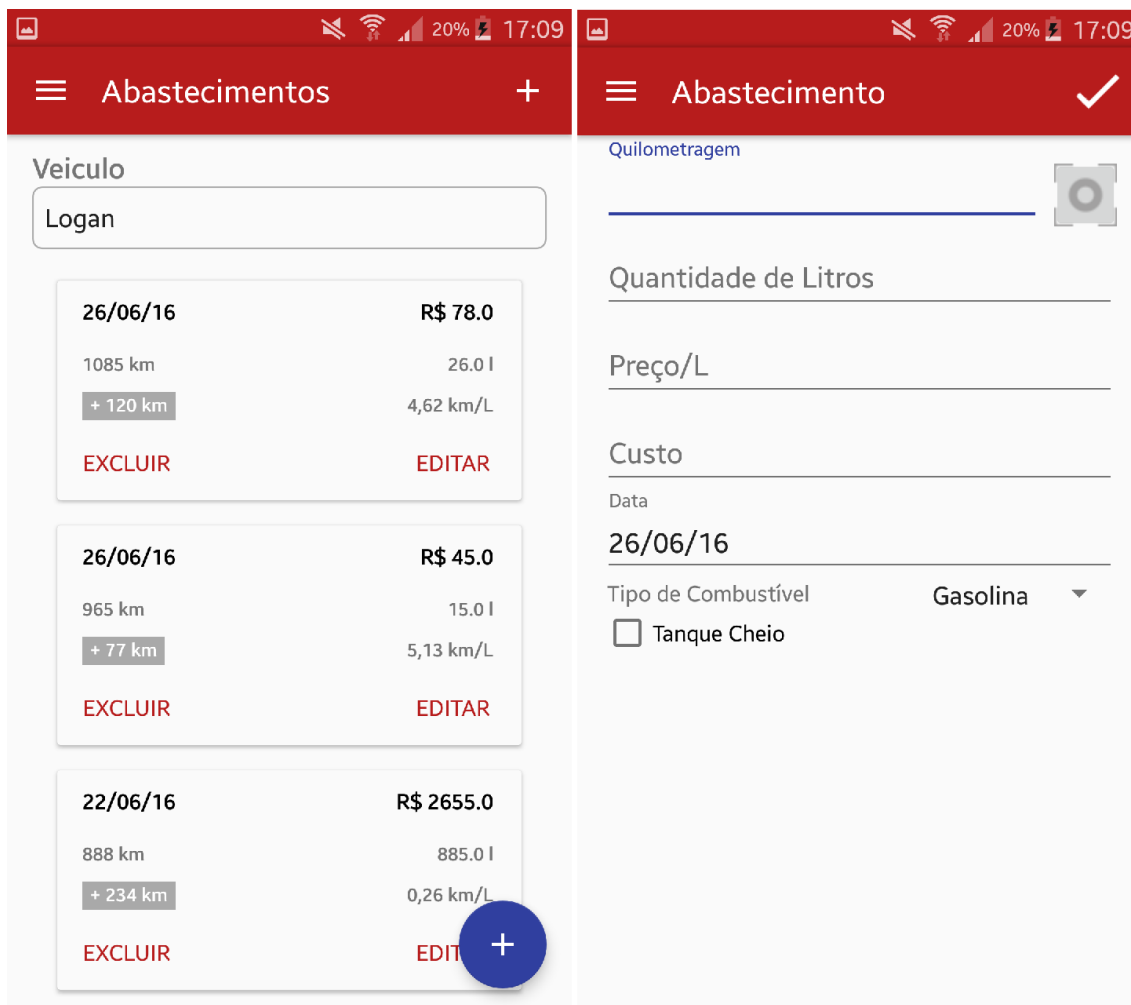


Figura 8 - TELAS ABASTECIMENTO

Para adicionar um abastecimento, é preciso pressionar o botão adicionar, localizado no canto inferior esquerdo. Nesta tela é possível cadastrar a data do abastecimento, o valor por litro, valor total e a quantidade de litros abastecidos.

É possível utilizar o leitor de imagens para a leitura do odômetro, conforme Figura 9 clicando no botão ao lado do campo de texto quilometragem. O aplicativo abrirá a câmera do dispositivo, e deve ser redimensionado o campo de leitura de imagem. Clicar no botão para focalizar e após, clicar em captura da imagem.

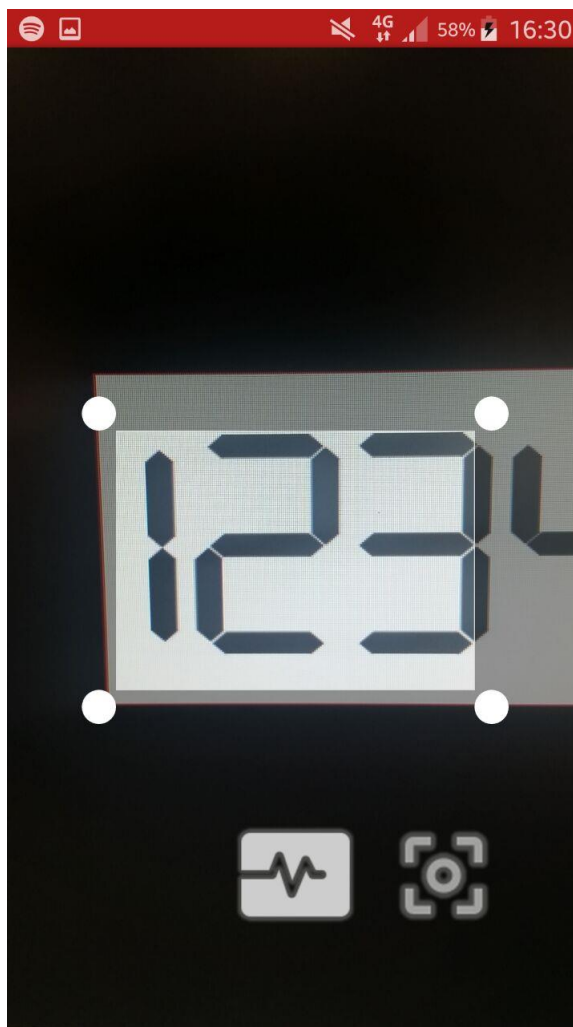


Figura 9 - TELA DE RECONHECIMENTO DE IMAGEM

Para salvar as informações inseridas, basta pressionar o *check-mark* localizado no canto superior direito. Após gravar os conteúdos inseridos, o sistema é redirecionado para a tela de lista.

Para editar um abastecimento já salvo basta clicar no botão editar. Dessa forma, os campos são apresentados preenchidos e aptos a serem alterados. Ao concluir a edição, basta pressionar o botão *check-mark*.

Para excluir é necessário pressionar o botão excluir no *card* do abastecimento. Sendo assim, o sistema apaga permanentemente o registro.

## 4.6 Manutenções Eventuais

Ao pressionar a opção manutenção-eventual, o sistema apresenta as informações de todas as manutenções eventuais cadastrados.

Para adicionar uma manutenção, é preciso pressionar o botão adicionar, localizado no canto inferior esquerdo. Nesta tela é possível cadastrar a data da manutenção, título, odômetro, valor e local, sendo que os locais já cadastrados são sugeridos.

Para salvar as informações inseridas, basta pressionar o *check-mark* localizado no canto superior direito. Após gravar os conteúdos inseridos, o sistema é redirecionado para a tela de lista.

Para editar uma manutenção eventual já salva basta clicar no botão editar. Dessa forma, os campos são apresentados preenchidos e aptos a serem alterados. Ao concluir a edição, basta pressionar o botão *check-mark*.

Para excluir é necessário pressionar o botão excluir no *card* da manutenção eventual. Sendo assim, o sistema apaga permanentemente o registro, como visto na Figura 10.

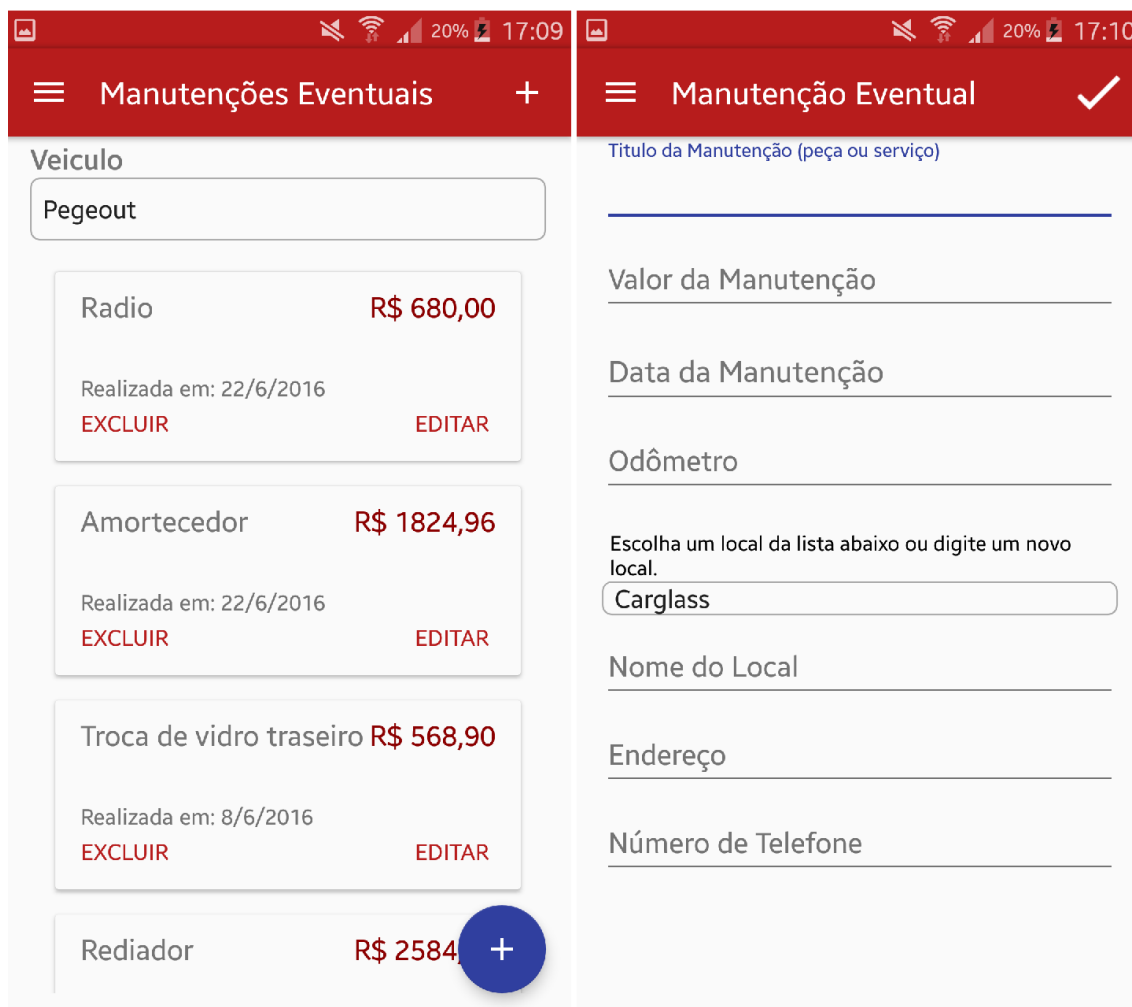


Figura 10 - TELAS DE MANUTEÇÕES EVENTUAIS

## 4.7 Manutenções Programadas

Ao pressionar a opção manutenção-programada, o sistema apresenta as informações de todas as manutenções programadas cadastradas como mostrado na Figura 12 - TELAS DE MANUTENÇÕES PROGRAMADAS.

Para adicionar uma manutenção, é preciso pressionar o botão adicionar, localizado no canto inferior esquerdo. Esta tela traz todas as opções de manutenções programadas, sendo escolhida a partir de uma lista, conforme Figura 11. É possível também cadastrar a data da manutenção, título, odômetro, valor e local, sendo que os locais já cadastrados são sugeridos, e dependendo da escolha feita o sistema gera automaticamente a data de vencimento e o odômetro.



Figura 11 - TELA DE ESCOLHA DO TIPO DE MANUTENÇÃO PROGRAMADA

Para salvar as informações inseridas, basta pressionar o *check-mark* localizado no canto superior direito. Após gravar os conteúdos inseridos, o sistema é redirecionado para a tela de lista.

Para editar uma manutenção programada já salva basta clicar no botão editar. Dessa forma, os campos são apresentados preenchidos e aptos a serem alterados. Ao concluir a edição, basta pressionar o botão *check-mark*.

Para excluir é necessário pressionar o botão excluir no *card* da manutenção programada. Sendo assim, o sistema apaga permanentemente o registro.

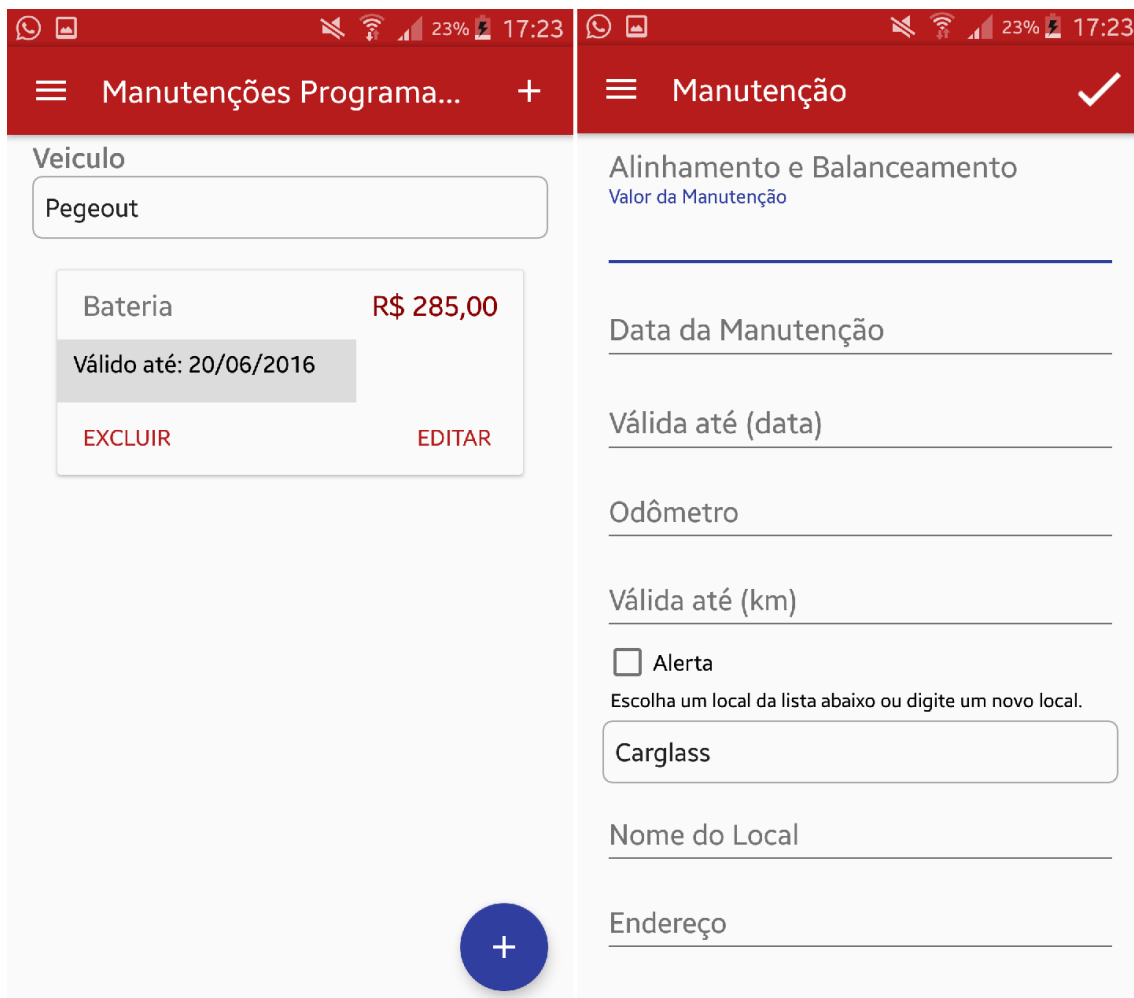


Figura 12 - TELAS DE MANUTENÇÕES PROGRAMADAS

## 4.8 Despesas Extras

Ao pressionar a opção despesas extras, o sistema apresenta as informações de todas as despesas extras eventuais cadastrados como na figura Figura 13.

Para adicionar uma despesa extra, é preciso pressionar o botão adicionar, localizado no canto inferior esquerdo. Nesta tela é possível cadastrar o título, o valor e a data que a despesa foi realizada.

Para salvar as informações inseridas, basta pressionar o *check-mark* localizado no canto superior direito. Após gravar os conteúdos inseridos, o sistema é redirecionado para a tela de lista.

Para editar uma despesa extra eventual já salva basta clicar no botão editar. Dessa forma, os campos são apresentados preenchidos e aptos a serem editados.

Para excluir é necessário pressionar o botão excluir no *card* da despesa extra. Sendo assim, o sistema apaga permanentemente o registro.

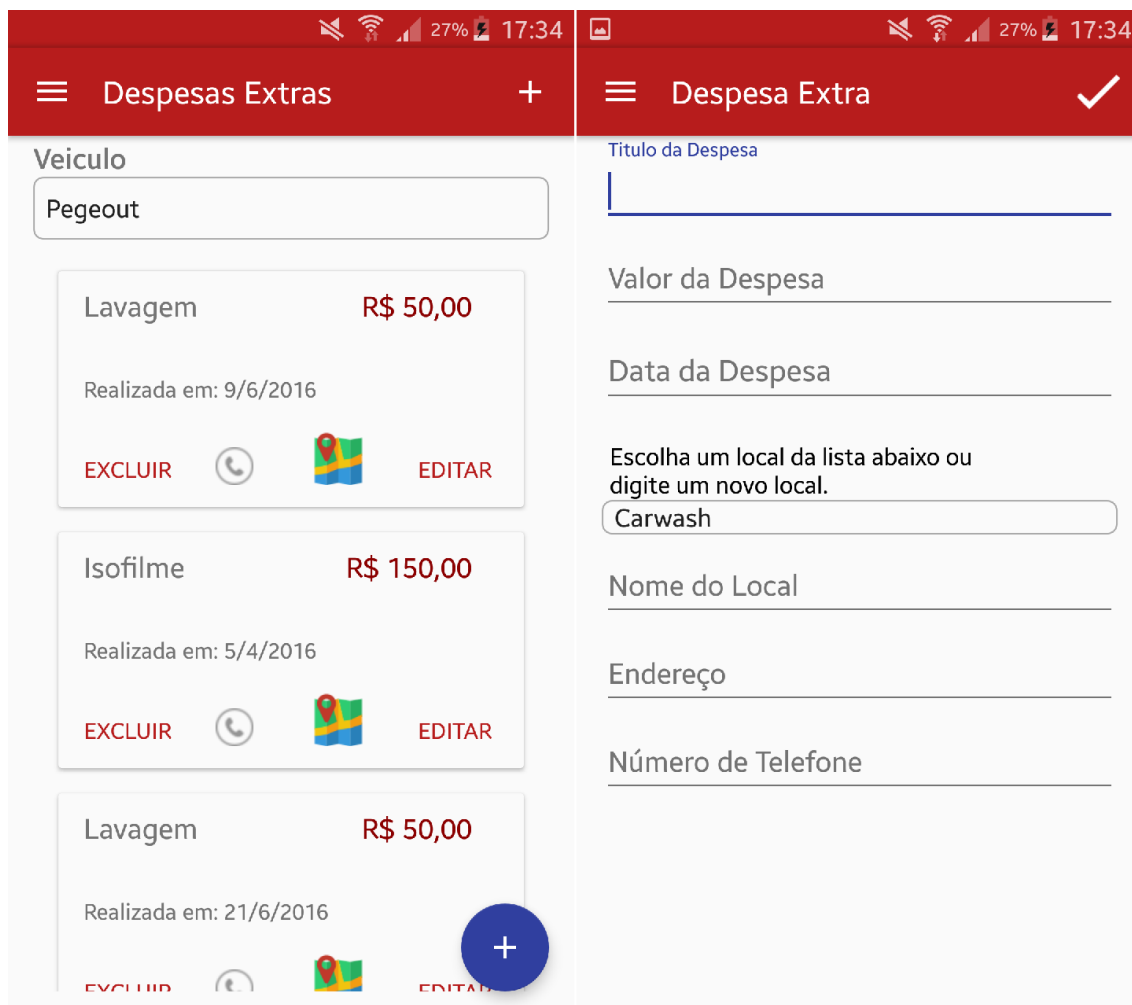


Figura 13 - TELAS DE DESPESAS EXTRAS

## 4.9 Locais

Ao pressionar a opção dados-locais, o sistema apresenta as informações de todos os locais cadastrados.

Para adicionar um local é preciso pressionar o botão adicionar, localizado no canto inferior esquerdo. Nesta tela é possível adicionar o nome, endereço e telefone do local.

Para salvar as informações inseridas, basta pressionar o *check-mark* localizado no canto superior direito. Após gravar os conteúdos inseridos, o sistema é redirecionado para a tela de lista.

Para editar um local já salvo basta clicar no botão editar. Dessa forma, os campos são apresentados preenchidos e aptos a serem alterados. Ao concluir a edição, basta pressionar o botão *check-mark*.

O ícone mapa direciona o endereço para o aplicativo de mapas do próprio celular.

Para excluir é necessário pressionar o botão excluir no *card* do local. Sendo assim, o sistema apaga permanentemente o registro como mostra a figura Figura 14.

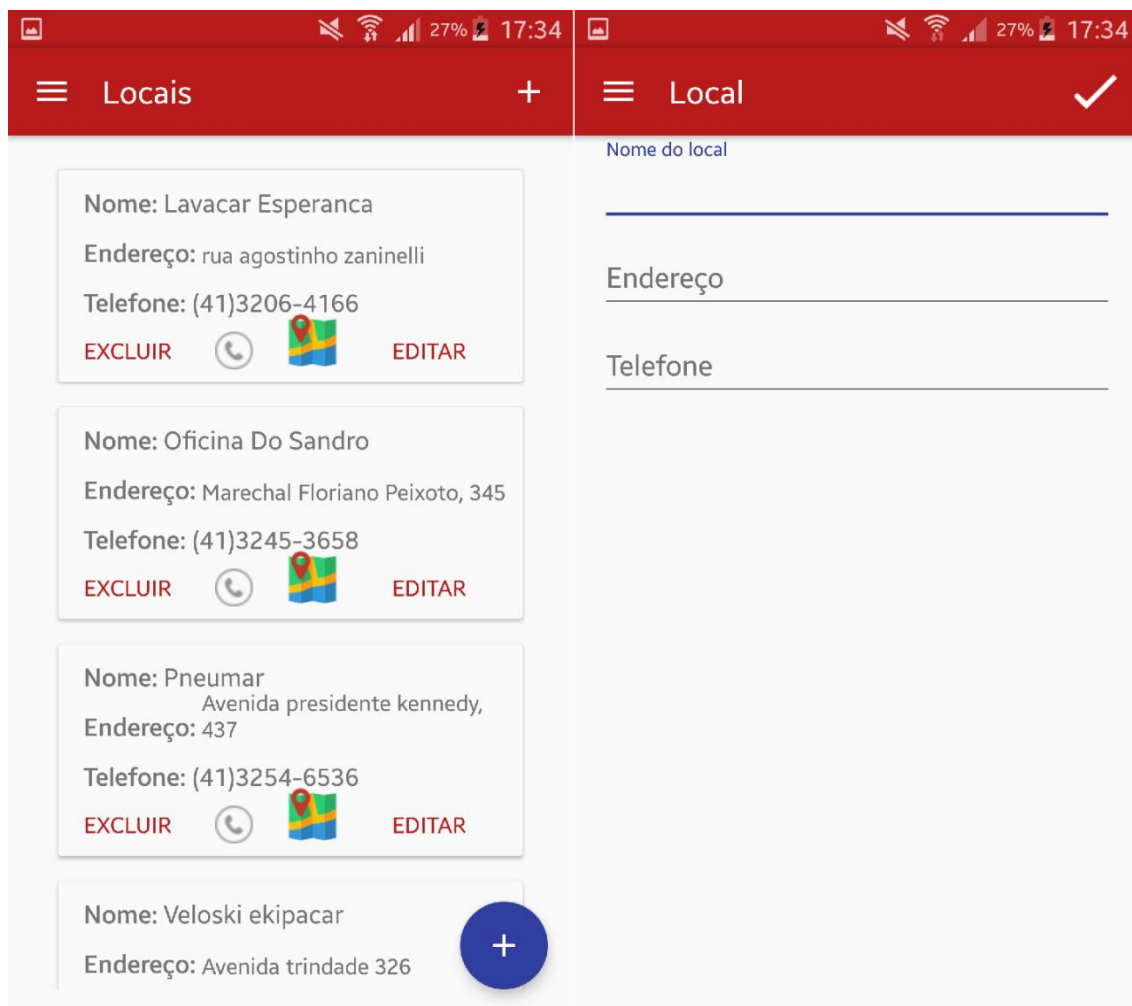


Figura 14 - TELAS DE LOCAIS

## 4.10 Imposto

Ao pressionar a opção gastos-imposto, o sistema apresenta as informações de todos os impostos cadastrados como segue a Figura 15.

Figura 15 - TELAS DE IMPOSTOS

Para adicionar um imposto, é preciso pressionar o botão adicionar, localizado no canto inferior esquerdo. Nesta tela é possível escolher o tipo de imposto, através de um *spinner*, que pode ser DPVAT<sup>18</sup>, IPVA<sup>19</sup> ou licenciamento. É possível também adicionar o ano, a referência, o valor total, o valor das parcelas, o número de parcelas e a data do primeiro pagamento.

Para salvar as informações inseridas, basta pressionar o *check-mark* localizado no canto superior direito. Após gravar os conteúdos inseridos, o sistema é redirecionado para a tela de lista.

<sup>18</sup> DPVAT: Seguro de danos pessoais causados por veículos automotores de vias terrestres

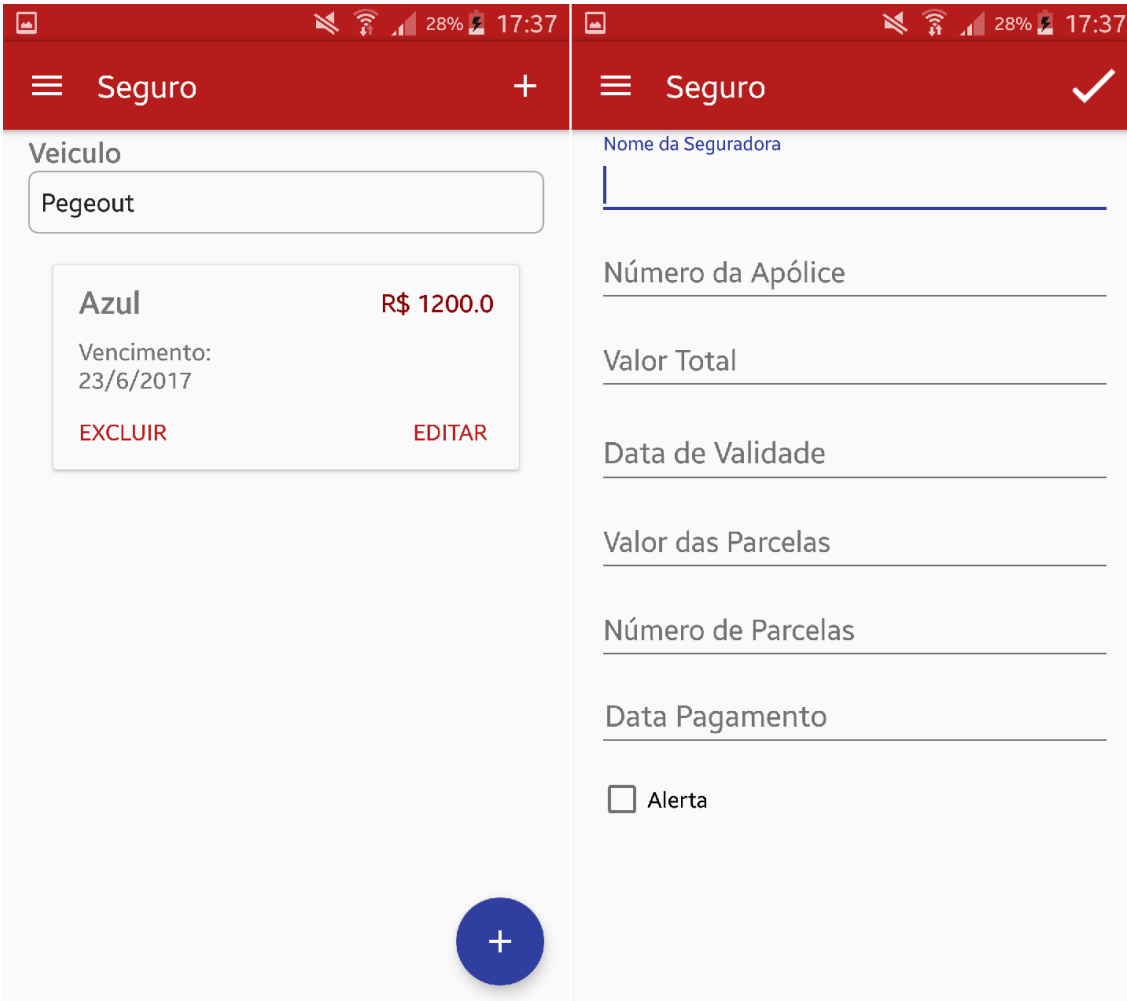
<sup>19</sup> IPVA: Imposto sobre a propriedade de veículos automotores

Para editar um imposto já salvo basta clicar no botão editar. Dessa forma, os campos são apresentados preenchidos e aptos a serem alterados. Ao concluir a edição, basta pressionar o botão *check-mark*.

Para excluir é necessário pressionar o botão excluir no *card* do imposto. Sendo assim, o sistema apaga permanentemente o registro.

#### 4.11 Seguro

Ao pressionar a opção gastos-seguro, o sistema apresenta as informações de todos os seguros cadastrados como mostra a Figura 16.



The image displays two side-by-side screenshots of a mobile application interface for managing insurance policies. Both screenshots show a red header bar with the text 'Seguro' and a navigation icon (hamburger menu) on the left. The left screenshot has a '+' icon on the right, while the right screenshot has a checkmark icon. The left screenshot shows a 'Veiculo' section with a text input field containing 'Pegeout'. Below it is a card for an insurance policy: 'Azul' with a value of 'R\$ 1200.0' and a due date of 'Vencimento: 23/6/2017'. At the bottom of the card are two buttons: 'EXCLUIR' and 'EDITAR'. A blue circular button with a white '+' sign is located at the bottom center of the left screenshot. The right screenshot shows a form for adding a new insurance policy with the following fields: 'Nome da Seguradora', 'Número da Apólice', 'Valor Total', 'Data de Validade', 'Valor das Parcelas', 'Número de Parcelas', 'Data Pagamento', and an 'Alerta' checkbox.

Figura 16 - TELAS DE SEGURO

Para adicionar um seguro, é preciso pressionar o botão adicionar, localizado no canto inferior esquerdo. Nesta tela é possível adicionar o nome, número da apólice, valor total, data de validade, data de pagamento, valor e número das parcelas além do alerta.

Para salvar as informações inseridas, basta pressionar o *check-mark* localizado no canto superior direito. Após gravar os conteúdos inseridos, o sistema é redirecionado para a tela de lista.

Para editar um seguro já salvo basta clicar no botão editar. Dessa forma, os campos são apresentados preenchidos e aptos a serem alterados. Ao concluir a edição, basta pressionar o botão *check-mark*.

Para excluir é necessário pressionar o botão excluir no *card* do seguro. Sendo assim, o sistema apaga permanentemente o registro.

#### **4.12 Financiamento**

Ao pressionar a opção gastos-financiamento, o sistema apresenta as informações de todos os financiamentos cadastrados.

Para adicionar um financiamento, somente um por veículo, é preciso pressionar o botão adicionar, localizado no canto inferior esquerdo. Nesta tela é possível adicionar o nome da financeira, valor total, valor parcelas, número de parcela, número de parcelas, dia do vencimento e número de parcelas pagas.

Para salvar as informações inseridas, basta pressionar o *check-mark* localizado no canto superior direito. Após gravar os conteúdos inseridos, o sistema é redirecionado para a tela de lista.

Para editar um financiamento já salvo basta clicar no botão editar. Dessa forma, os campos são apresentados preenchidos e aptos a serem alterados. Ao concluir a edição, basta pressionar o botão *check-mark*.

Para excluir é necessário pressionar o botão excluir no *card* do financiamento. Sendo assim, o sistema apaga permanentemente o registro, assim como mostra a Figura 17.

The image displays two side-by-side screenshots of a mobile application interface for financial management. Both screens have a red header with the text 'Financiamento'. The left screen features a 'Veiculo' section with a text input field containing 'Pegeout'. Below this is a card for 'Bradesco' with a value of 'R\$ 17000,00', a due date of '12/6/2016', and a note 'Faltam 43 parcela(s)'. At the bottom of the card are 'EXCLUIR' and 'EDITAR' buttons. A large blue '+' button is positioned at the bottom center of the screen. The right screen shows a form for 'Financiamento' with several input fields: 'Nome da Financeira', 'Valor Total', 'Valor Parcelas', 'Número de Parcelas', 'Dia do Vencimento', and 'Número de Parcelas Pagas'. There is also a checkbox labeled 'Alerta'.

Figura 17 - TELAS DE FINANCIAMENTO

#### 4.13 Multas

Ao pressionar a opção gastos-multas, o sistema apresenta as informações de todas as multas cadastradas.

Para adicionar uma multa, é preciso pressionar o botão adicionar, localizado no canto inferior esquerdo. Nesta tela é possível adicionar o título da multa, data de quando ocorreu, valor, data de vencimento e descrição.

Para salvar as informações inseridas, basta pressionar o *check-mark* localizado no canto superior direito. Após gravar os conteúdos inseridos, o sistema é redirecionado para a tela de lista.

Para editar uma multa já salva basta clicar no botão editar. Dessa forma, os campos são apresentados preenchidos e aptos a serem alterados. Ao concluir a edição, basta pressionar o botão *check-mark*.

Para excluir é necessário pressionar o botão excluir no *card* da multa. Sendo assim, o sistema apaga permanentemente o registro, como mostra a Figura 18.

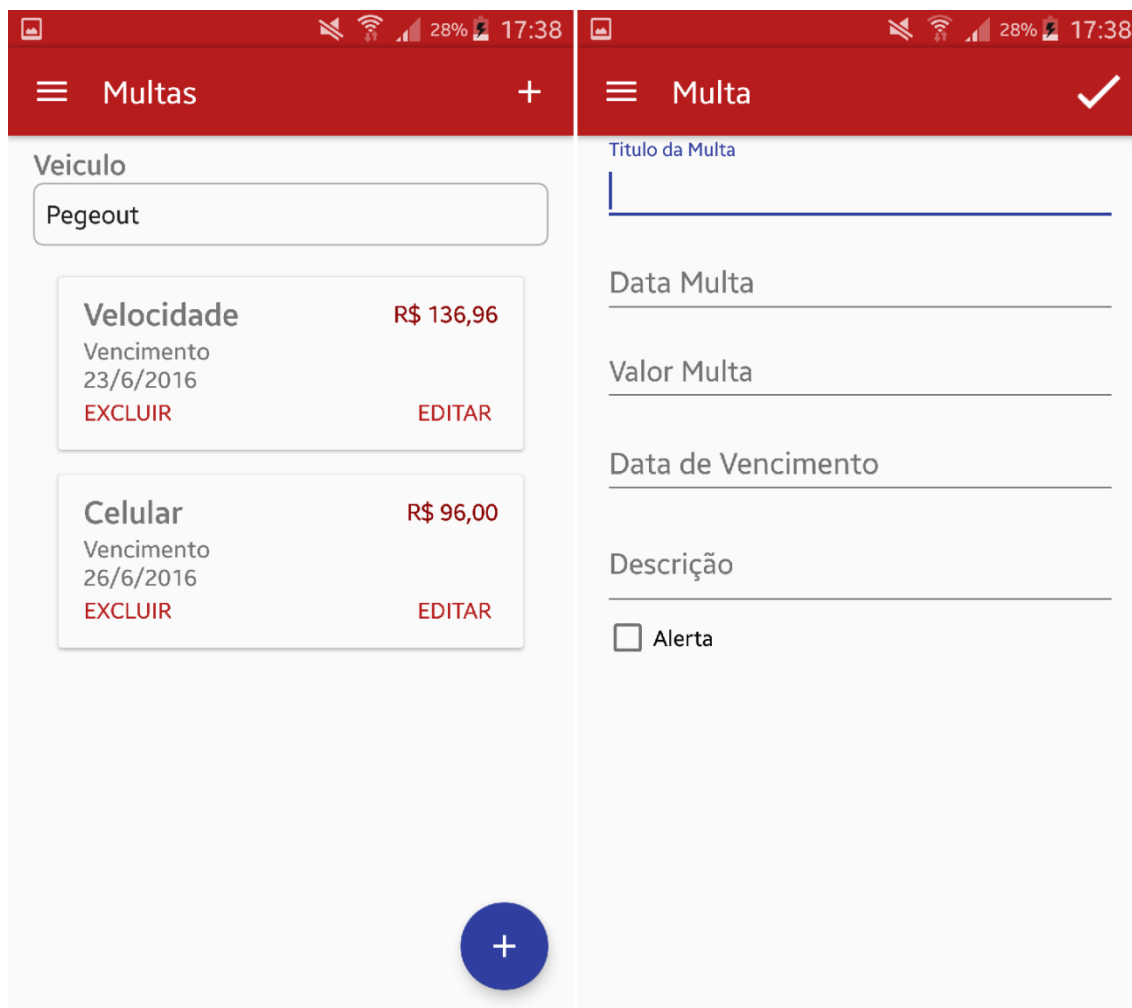


Figura 18 - TELAS DE MULTAS

#### 4.14 Telefones

Ao pressionar a opção dados-telefone, o sistema apresenta as informações de todos os telefones cadastrados.

Para adicionar um telefone é preciso pressionar o botão adicionar, localizado no canto inferior esquerdo. Nesta tela é possível adicionar o nome e telefone do contato.

Para salvar as informações inseridas, basta pressionar o *check-mark* localizado no canto superior direito. Após gravar os conteúdos inseridos, o sistema é redirecionado para a tela de lista.

Para editar um telefone já salvo basta clicar no botão editar. Dessa forma, os campos são apresentados preenchidos e aptos a serem alterados. Ao concluir a edição, basta pressionar o botão *check-mark*.

Para excluir é necessário pressionar o botão excluir no *card* do telefone. Sendo assim, o sistema apaga permanentemente o registro como segue na figura Figura 19.

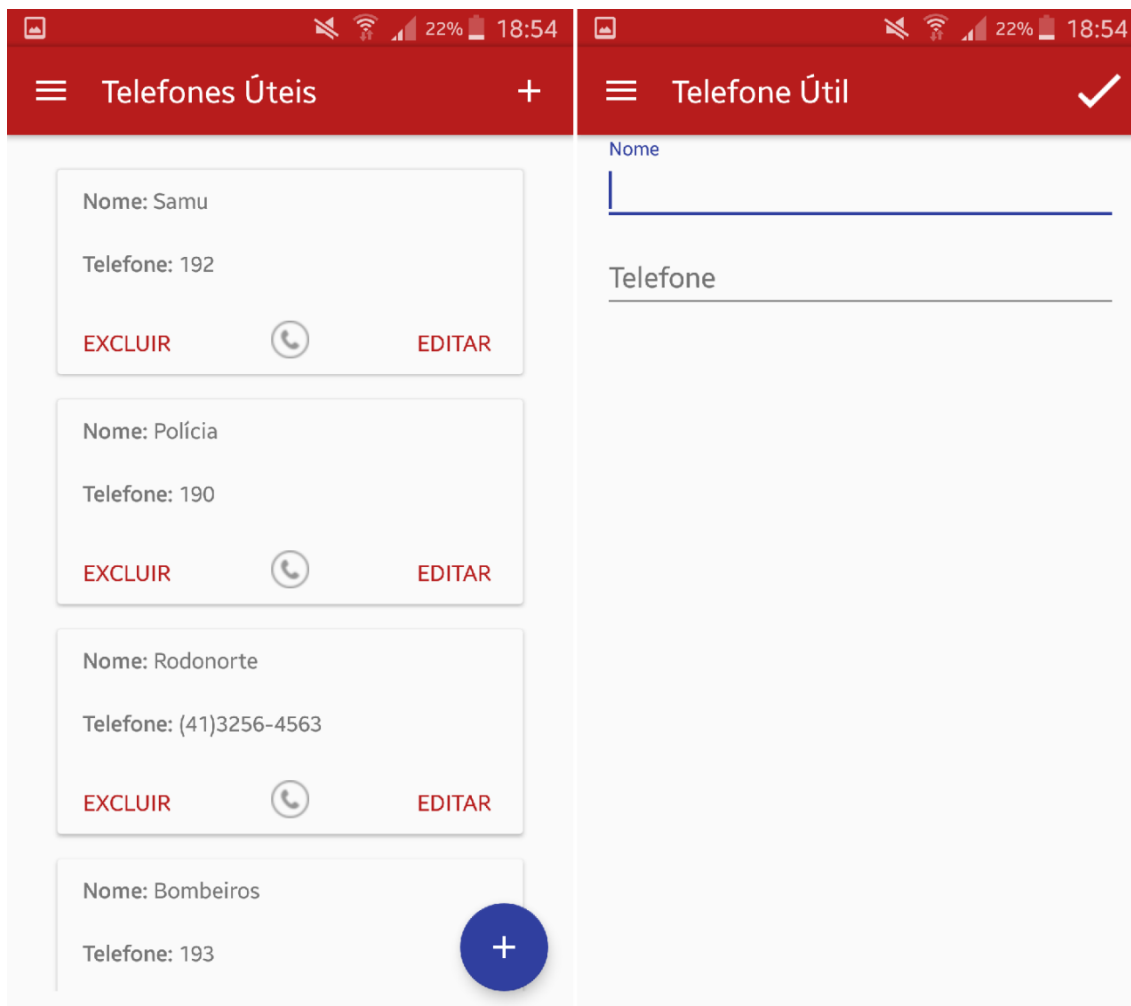


Figura 19 - TELAS DE TELEFONES ÚTEIS

#### 4.15 Viagens

Ao pressionar a opção viagens, o sistema apresenta as informações de todas as viagens cadastradas.

Para adicionar uma viagem, é preciso pressionar o botão adicionar, localizado no canto inferior esquerdo. Nesta tela é inserido o nome da viagem e a distância até o destino. Com essas informações o sistema calcula qual será o

valor estimado de gasto baseado nas médias de gastos de combustíveis salvos no sistema.

O *card* viagens possui também as opções de fechar viagem, adicionar abastecimento e adicionar despesas.

Para salvar as informações inseridas, basta pressionar o *check-mark* localizado no canto superior direito. Após gravar os conteúdos inseridos, o sistema é redirecionado para a tela de lista.

Para editar uma viagem já salvo basta clicar no botão editar. Dessa forma, os campos são apresentados preenchidos e aptos a serem alterados. Ao concluir a edição, basta pressionar o botão *check-mark*.

Para excluir é necessário pressionar o botão excluir no *card* da viagem. Sendo assim, o sistema apaga permanentemente o registro como mostra na figura Figura 20.

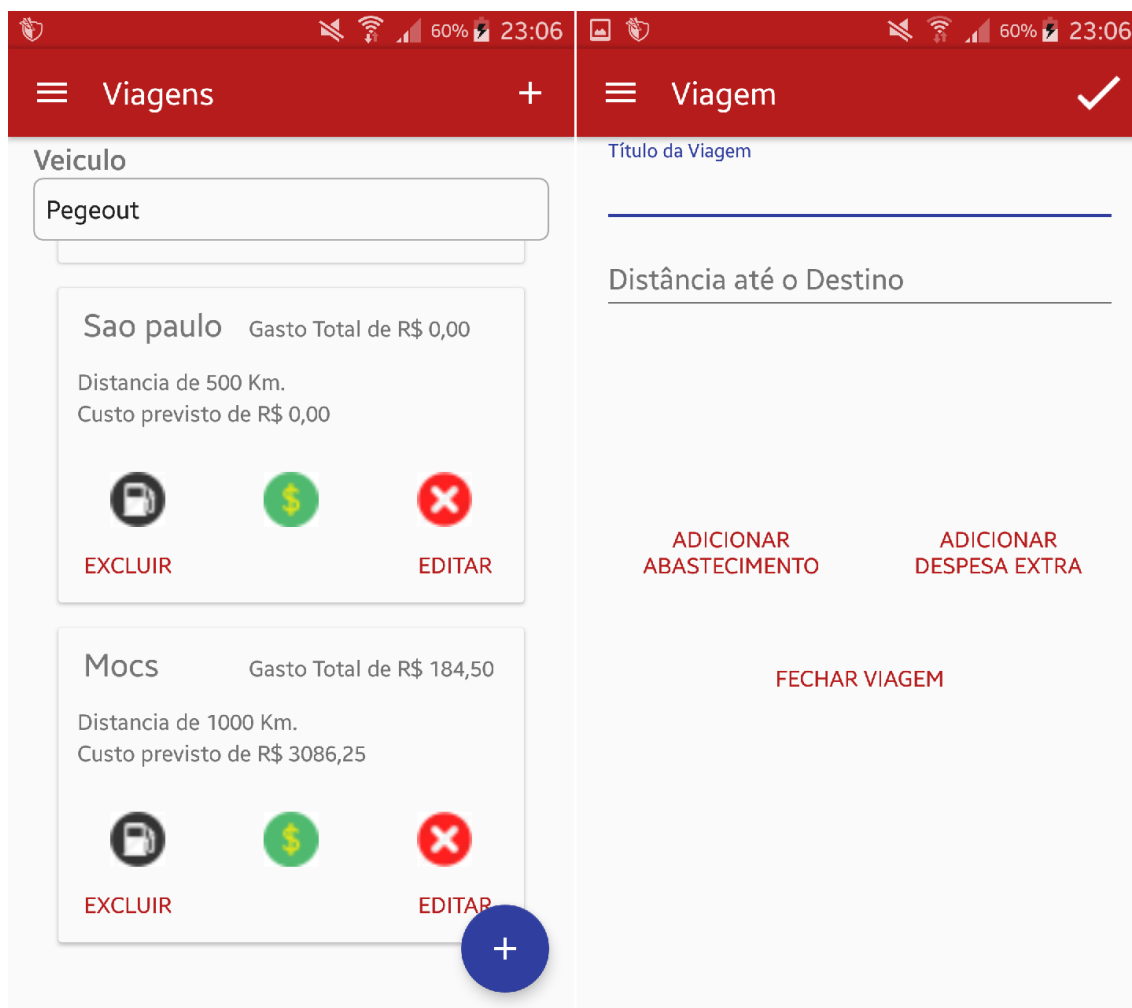


Figura 20 - TELAS DE VIAGEM

Para adicionar um abastecimento de viagem, pode-se entrar pelo *card* clicando no botão com a bomba de combustível ou entrando em editar viagem, e após adicionar abastecimento. O sistema abrirá uma tela com todas os abastecimentos cadastrados na viagem. Ao clicar no botão de adicionar no canto inferior direito da tela o usuário será direcionado para a tela de cadastrar um novo abastecimento. Após um novo abastecimento cadastrado o sistema redireciona novamente para a tela de lista de abastecimentos de viagem. O usuário pode retornar para a tela de viagens clicando no botão no canto superior direito da tela. Como visto na Figura 21.

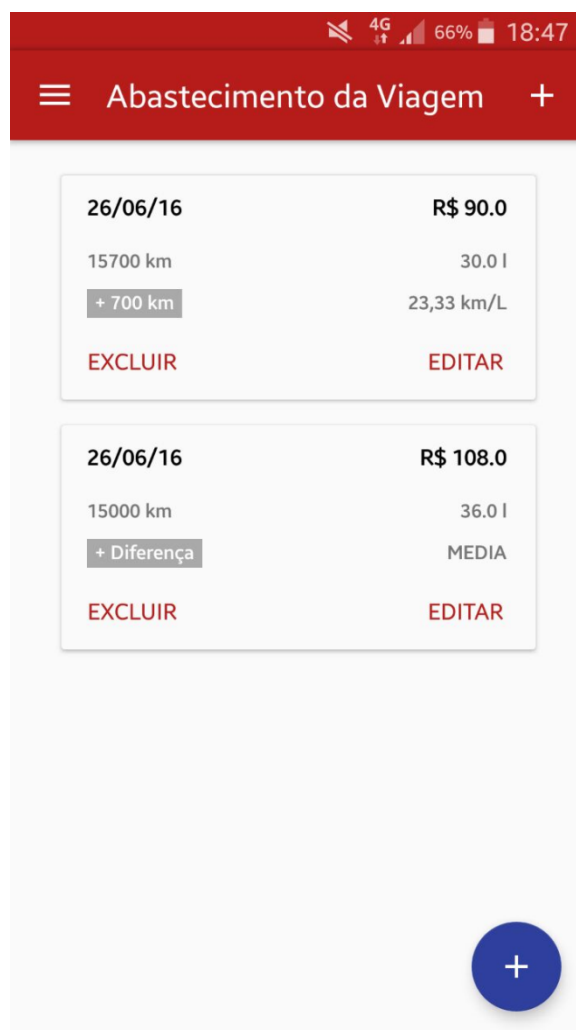
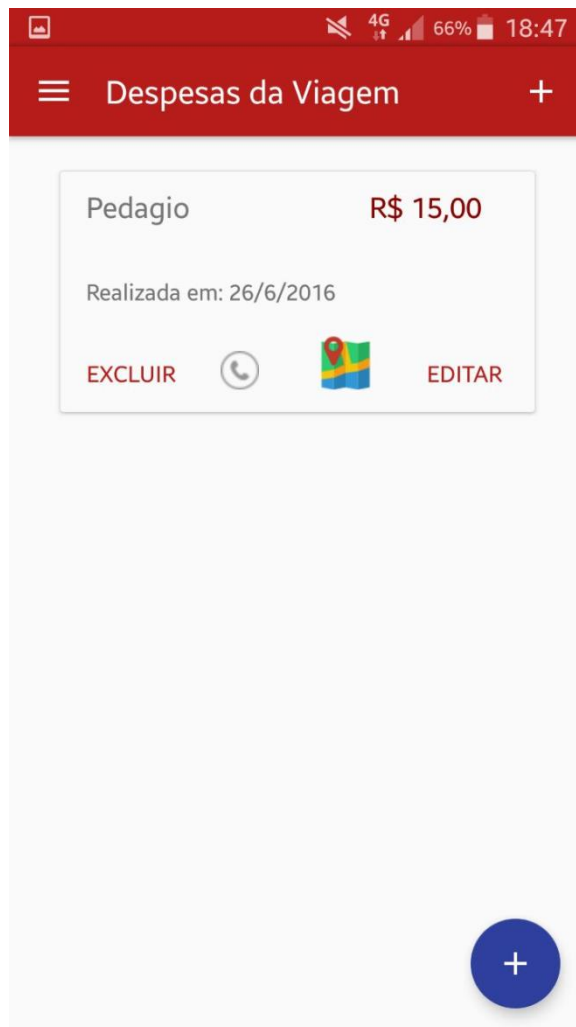


Figura 21 - TELA DE LISTA DE ABASTECIMENTOS DE VIAGEM

Para cadastrar uma despesa de viagem o usuário pode clicar no botão com um cifrão verde no *card* ou pelo caminho editar viagem – adicionar despesa extra. O usuário será direcionado para a tela de listas de despesas extras de viagem. Para adicionar uma nova despesa deve-se clicar no botão de adicionar

no canto inferior direito da tela. O sistema irá abrir a tela de cadastrar despesa extra. Após cadastramento da nova despesa o usuário será redirecionado para a tela de despesas extras de viagem. Ao clicar no canto superior direito volta-se para a tela de viagens, assim como mostra a figura Figura 22.



**Figura 22 - TELA DE LISTA DE DESPESAS EXTRAS DE VIAGEM**

Para encerrar uma viagem o usuário pode clicar no botão vermelho com um x no *card* ou em editar viagem – fechar viagem. A tela de fechar viagem como mostra a Figura 23 traz as informações de despesas extras e abastecimentos gastos durante a viagem. O usuário ainda pode escolher uma porcentagem de desgaste do veículo através do seek-bar<sup>20</sup> para a viagem, que começa com um valor padrão de 5% do valor gasto na viagem. O usuário pode ainda escolher o número de companhias na viagem. O sistema vai automaticamente calcular o total gasto na viagem e quanto cada uma das companhias deve pagar. Ao clicar

---

<sup>20</sup> Seek-bar: barra de progresso

no botão fechar viagem o sistema irá deixar disponibilizado para consulta um relatório da viagem. Como mostra a Figura 23.

The screenshot shows a mobile application interface for closing a trip. At the top, there is a red header with a hamburger menu icon, the text 'Fechar Viagem', and a plus sign. Below the header, the screen is divided into several sections:

- Two columns of expense data: 'Gasto total com combustível' (198.0) and 'Gasto total com outras despesas' (15.0).
- A section for 'Número de Companhias da Viagem' with a text input field containing the number '2'.
- A slider control for 'Taxa para desgaste de veículo' (Vehicle wear tax) set to 5%.
- A red button labeled 'CANCELAR FECHAMENTO DA VIAGEM'.
- A section titled 'Total de Gastos' (Total Expenses) showing a value of 223,65.
- A final calculation: 'Valor por pessoa: R\$111,82'.

Figura 23 - TELA DE FECHAR VIAGEM

#### 4.16 Relatório

Ao pressionar a opção relatório, o sistema apresenta a tela onde é necessário selecionar, através do *spinner*, qual será o carro e qual o período do relatório (data-início e data-fim).

O usuário deve então escolher as opções que deseja incluir no relatório, como por exemplo, desempenho e outras despesas. Na opção outras despesas, o sistema permite inserir outras informações, tais como: manutenções, despesas extras, multas, impostos, seguro e financiamento.

Após a escolha dos dados do relatório, o sistema abrirá uma nova tela com o relatório requisitado. Conforme a Figura 24.

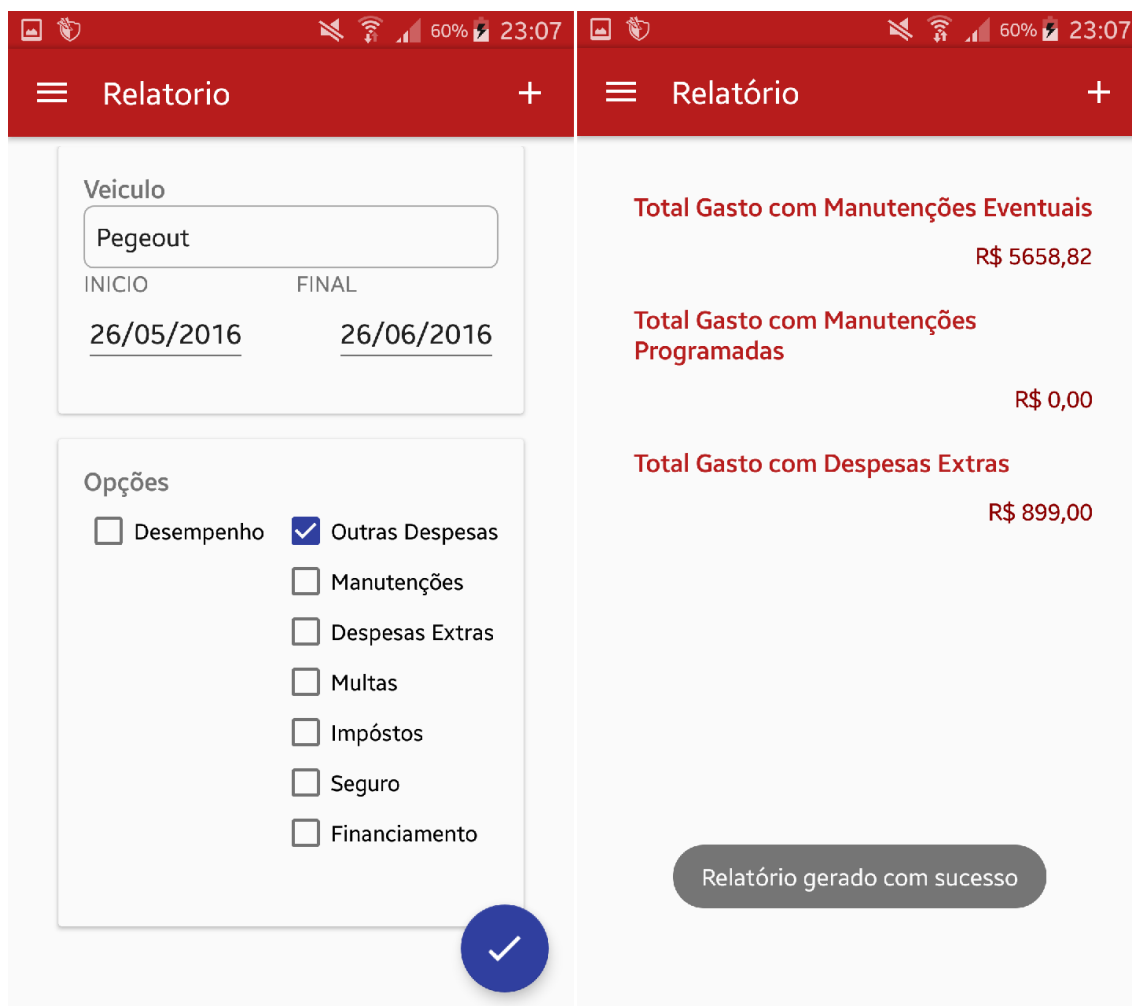
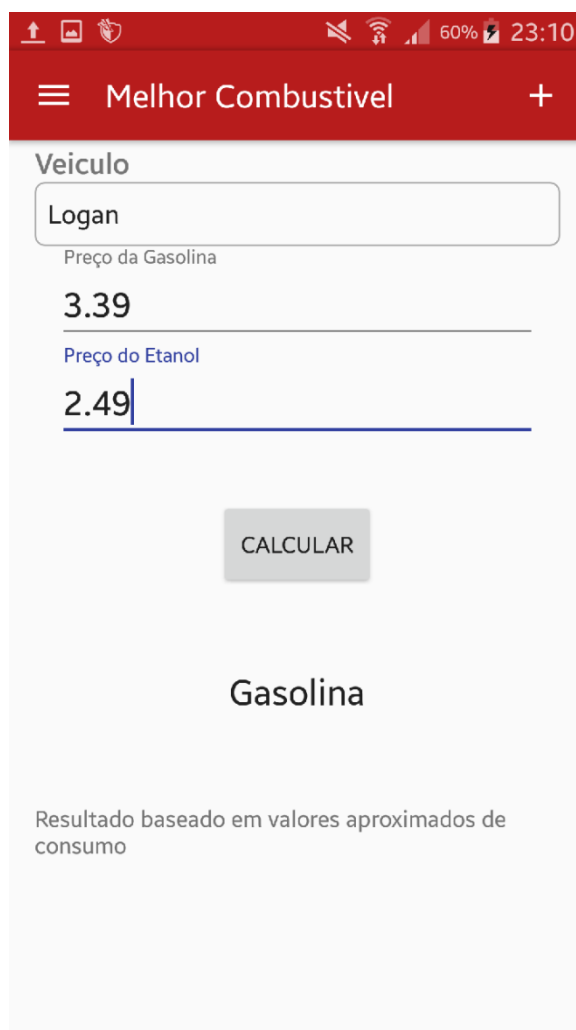


Figura 24 - TELAS DE RELATÓRIO

#### 4.17 Melhor Combustível

Ao pressionar a opção combustível-melhor combustível, o sistema apresenta a tela onde é necessário selecionar, através do *spinner*, qual será o carro e a partir dessa escolha o melhor combustível é calculado. Este cálculo é feito com base no histórico das médias, como mostra a Figura 25 - TELA MELHOR COMBUSTIVEL.



**Figura 25 - TELA MELHOR COMBUSTIVEL**

Este capítulo apresentou as telas e funcionalidades do aplicativo AidAuto. O capítulo seguinte irá expor uma visão geral do documento e sugestões para trabalhos futuros.

## 5 CONSIDERAÇÕES FINAIS

Esse projeto se iniciou com a necessidade de gerenciar manutenções veiculares. Para tanto, realizou-se uma análise dos principais *softwares* do mercado, e também feito o levantamento dos requisitos para a construção de um novo aplicativo que se adeque ao perfil dos usuários finais.

Dentre esses requisitos, identificou-se a necessidade de notificar o momento exato de realizar as manutenções preventivas, visando reduzir danos maiores ao veículo, diminuindo assim custos e tempo de oficina.

Após as análises, concluiu-se também a necessidade em gerenciar contas fixas como IPVA (Imposto sobre a Propriedade de Veículos Automotores), licenciamento, seguro (obrigatório e particular), parcelamento e financiamento do veículo.

Tendo em vista esses fatores, desenvolveu-se um aplicativo para dispositivos móveis a fim de ajudar o usuário a manter um bom funcionamento do veículo e estar em dia com os pagamentos, conforme o objetivo proposto no início do projeto.

Os objetivos específicos foram atingidos, pois foram avaliadas as interfaces e funções de aplicativos concorrentes, conforme seção 2.3. A partir desta análise, foram determinadas as principais funcionalidades, levando em conta sua relevância. Foi realizada também uma pesquisa sobre manutenções em veículos, com o intuito de notificar o usuário o momento certo de realizar a prevenção no automóvel.

O levantamento de requisitos necessários foi realizado baseado nas funcionalidades dos aplicativos descritos na seção 2.3, com diferenças como: reconhecimento de imagem, planejamento de viagem e cálculo do melhor combustível.

É importante ressaltar que melhorias podem ser implementadas futuramente no reconhecimento de imagem do odômetro, a biblioteca utilizada apresenta dificuldades ao ler imagens, tais como: a câmera não pode estar no modo paisagem; quando o fundo é escuro, a letra não pode ser clara e a

distância da foto não pode ultrapassar 70 centímetros. Além disso, os números a serem lidos precisam ser sete segmentos. Para solucionar este problema, é necessário treinar a biblioteca de padrões utilizada.

Além disso, pode-se citar como sugestão de trabalhos futuros, a integração do dispositivo móvel com o carro, chamado AndroidAuto. Este dispositivo foi projetado para minimizar as distrações e manter o foco do motorista na estrada, pois possui controles integrados no volante e novas ações de voz. Para a utilização de tal funcionalidade, é necessário ter um veículo compatível com o AndroidAuto e um smartphone Android com a versão 5.0 ou superior.

Espera-se ajudar as pessoas a manter seu veículo em bom estado evita acidentes que podem ser causado por falhas de equipamentos, mau funcionamento de peças ou até mesmo falta de manutenção.

## 6 REFERÊNCIAS BIBLIOGRÁFICAS

ANDROID STUDIO – **A IDE Oficial para Android** (The Official IDE for Android). Disponível em: <<http://developer.android.com/intl/pt-br/sdk/index.html>>. Acesso em 4 de maio de 2016.

BEDANI, Janaína. Engenharia de Software 2 - **Técnicas para levantamento de Requisitos**. Disponível em <<http://www.devmedia.com.br/engenharia-de-software-2-tecnicas-para-levantamento-de-requisitos/9151>>. Acesso em 19 de maio de 2016.

BORDIN, Maycon Viana. **Introdução a Arquitetura Android**. Disponível em: <<http://goo.gl/hla47Z>>. Acesso em 2 de junho de 2016.

CITISYSTEMS. **Confiabilidade e disponibilidade de máquinas: um exemplo prático**. Disponível em: <<http://www.citisystems.com.br/confiabilidade-disponibilidade-maquinas/>>. Acesso em 1 de abril de 2016.

COMPULETRA. **Porque fazer Manutenção Preventiva**. Disponível em: <<http://www.compuletra.com.br/vistoria-pro/porque-fazer-manutencao-preventiva/>>. Acesso em 13 de abril de 2016.

DESENVOLVEDORES ANDROID (ANDROID DEVELOPERS) - **O Guia do Desenvolvedor** (The developer's guide). Disponível em: <<https://developer.android.com/guide/index.html>>. Acesso em 03 julho 2016.

DETRAN, 2016. **Não fazer manutenção do veículo triplica risco de acidente**. Disponível em: <<http://www.detran.pr.gov.br/modules/noticias/article.php?storyid=1949>>. Acesso em 15 de maio de 2016.

GAZETA DO POVO, 2010. **Falta de manutenção nos veículos facilita a ocorrência de acidentes**. Disponível em: < <http://goo.gl/UNywL6> >. Acesso em 15 de maio de 2016.

LECHETTA, Ricardo R. **Google Android - aprenda a criar aplicações para dispositivos móveis com o SDK**. São Paulo : Novatec Editora, 2013.

NET MARKETSHARE – **Tecnologia de Marketing de estatística compartilhada para internet** (Market Share Statistics for Internet

Technologies). Disponível em: <<http://goo.gl/gq8rGw>>. Acesso em 05 de junho de 2016.

NETO, Arilo Cláudio Dias. **Introdução à modelagem de dados - Artigo Revista SQL Magazine 86**. 2014. Disponível em: <<http://www.devmedia.com.br/introducao-a-modelagem-de-dados-artigo-revista-sql-magazine-86/20398>>. Acesso em 5 de junho de 2016.

NUDELMAN, Greg. **Padrões de Projeto para o Android – Soluções de Projetos de Interação para Desenvolvedores**. São Paulo : Novatec Editora, 2013.

PRESSMAN, Roger S. **Engenharia de Software**. São Paulo : Pearson Education do Brasil, 1995.

RIBEIRO, Leandro. **O que é UML e Diagramas de Caso de Uso: Introdução Prática à UML**. Disponível em: <<http://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408>>. Acesso em 23 de maio de 2016.

RODRIGUES, Joel. **Modelo Entidade Relacionamento (MER) e Diagrama Entidade- Relacionamento (DER)**. Disponível em: <<http://goo.gl/lwDfuE>>. Acesso em: 23 de maio de 2016.

SCHWABER; SUTHERLAND, 2013 - **Guia do Scrum**. Disponível em: <<http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-Portuguese-BR.pdf>>. Acesso em 27 de março de 2016.

SCHWABER, Ken. **Desenvolvimento Ágil de Software usando Scrum (Agile Software Development With Scrum)**. Prentice Hall, 2001.

SCRUM ALLIANCE, 2014 – **O Guia do Scrum**. Disponível em: <<https://www.scrumalliance.org/why-scrum/scrum-guide>>. Acesso em 26 de março de 2016.

SILVA, Paulo César Barreto. **Artigo SQL Magazine 63 - Utilizando UML: Diagrama de Classes**. Disponível em <<http://www.devmedia.com.br/artigo-sql-magazine-63-utilizando-uml-diagrama-de-classes/12251>>. Acesso em 19 de maio de 2016.

SOCARRÃO, 2009. **Correia dentada: Barata para trocar, cara se quebrar.** <<http://revista.socarrao.com.br/geral/correia-dentada-barata-para-trocar-cara-se-quebrar>>. Acesso em 11 de maio de 2016.

TESSERACT – OCR, 2016. **Tesseract Código Aberto OCR.** Disponível em <<https://github.com/tesseract-ocr/tesseract/wiki>> Acesso em 05 de junho de 2016.

TISELVAGEM. **Desenvolvendo para Android: Arquitetura Android.** Disponível em: <<http://www.tiselvagem.com.br/geral/desenvolvendo-para-android-arquitetura-android/>>. Acesso em 30 de maio de 2016.

UOL CARROS - **Tecnologia e bom senso avisam: tem muita coisa inútil no seu carro.** Disponível em: <<http://goo.gl/2DM8to>>. Acesso em 1 de abril de 2016.

## APÊNDICE A – REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS

Com base no levantamento de requisitos, foram identificados os principais requisitos funcionais e não funcionais para o sistema proposto, conforme descrito abaixo.

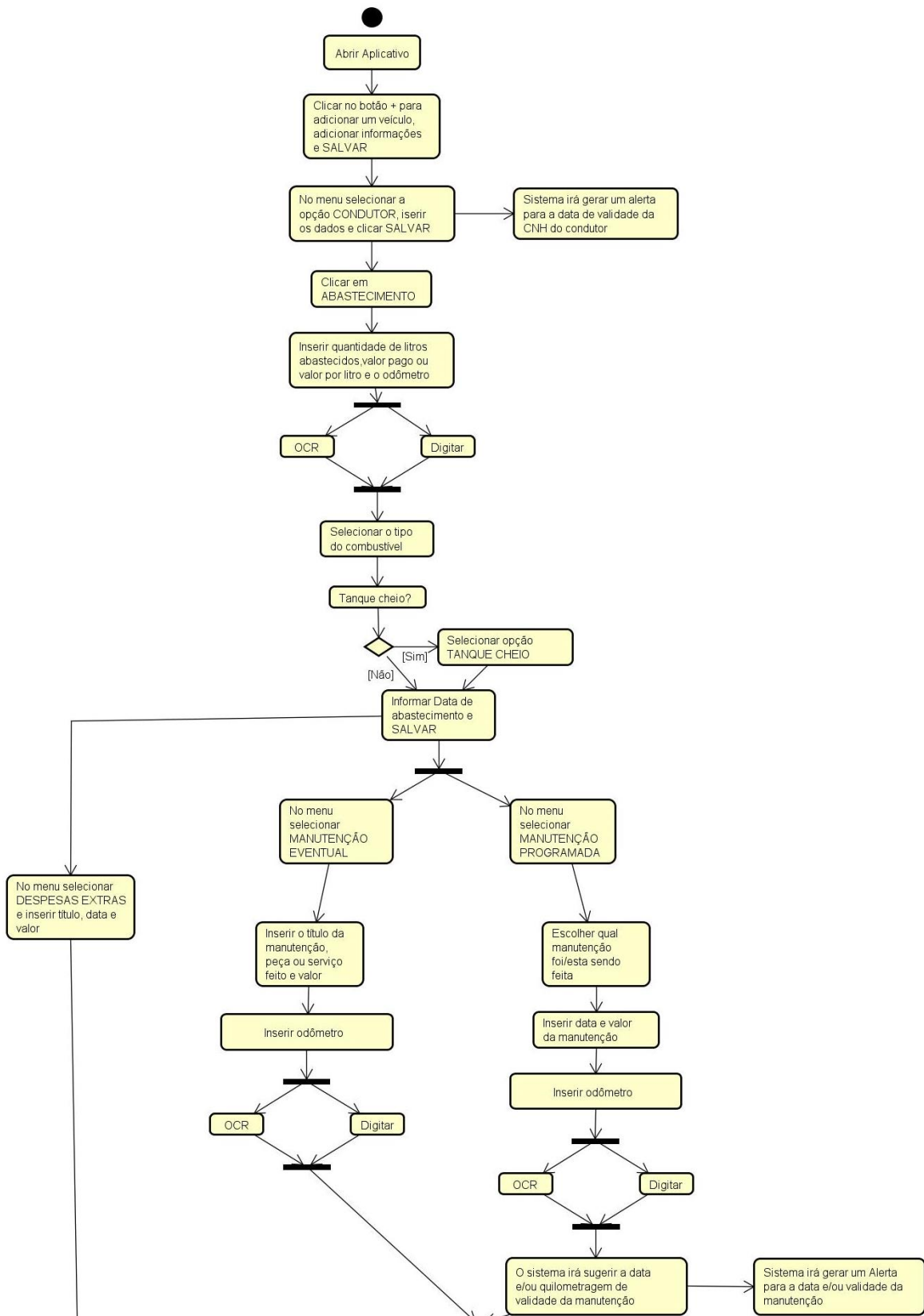
### a) Requisitos Funcionais

- Cadastrar veículo;
- Cadastrar condutor;
- Cadastrar abastecimento;
- Reconhecimento de imagem para leitura do odômetro;
- Cadastrar manutenção eventual e programada;
- Cadastrar despesas extras;
- Cadastrar impostos;
- Cadastrar seguro;
- Cadastrar financiamento;
- Gerar relatório;
- Calcular planejamento de viagem;
- Cadastrar telefones úteis;
- Cadastrar locais;
- Calcular o melhor combustível;
- Gerar alerta;
- Mostrar dicas.

### b) Requisitos não Funcionais

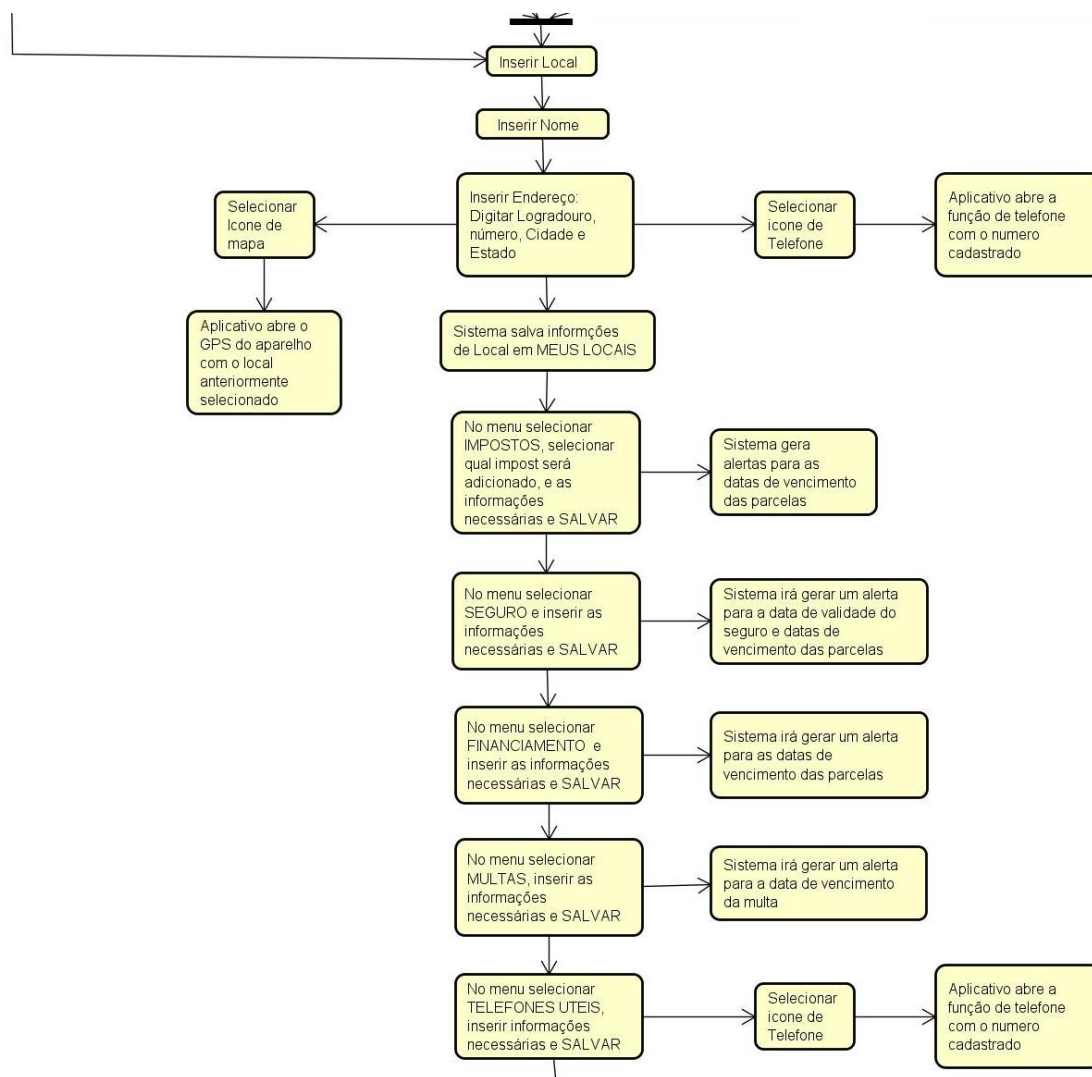
- Possuir interface amigável e intuitiva;
- Tratamento de erros;
- Utilizar o ambiente de desenvolvimento *Android Studio* para desenvolver o Software;
- Tempo de resposta abaixo de 3 segundos;
- Utiliza o banco de dados SQLite;
- Linguagem de programação Java.

## APÊNDICE B – DIAGRAMA GERAL DE ATIVIDADES



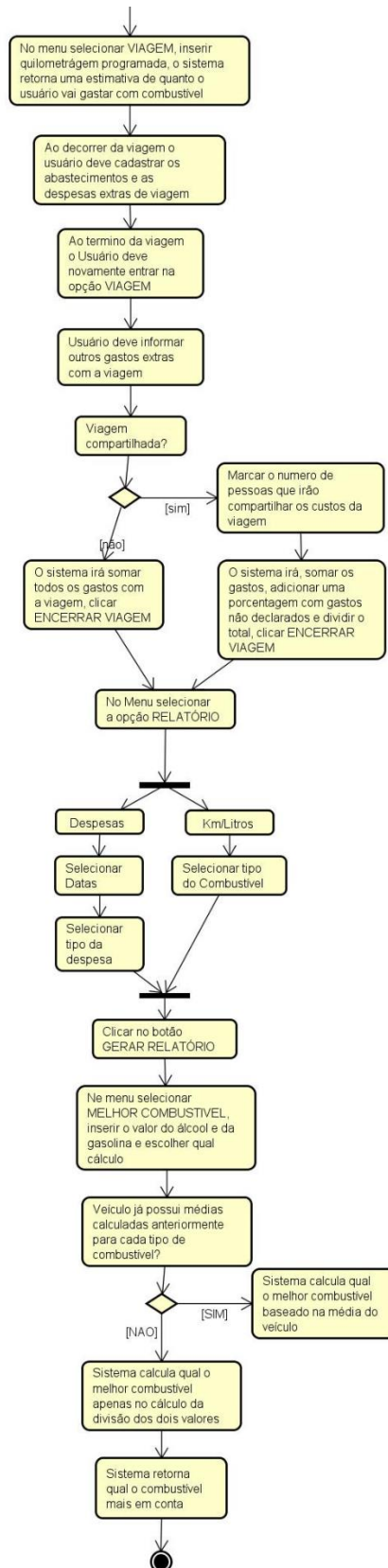
Continua na próxima página.

## APÊNDICE B – DIAGRAMA GERAL DE ATIVIDADES – CONTINUAÇÃO

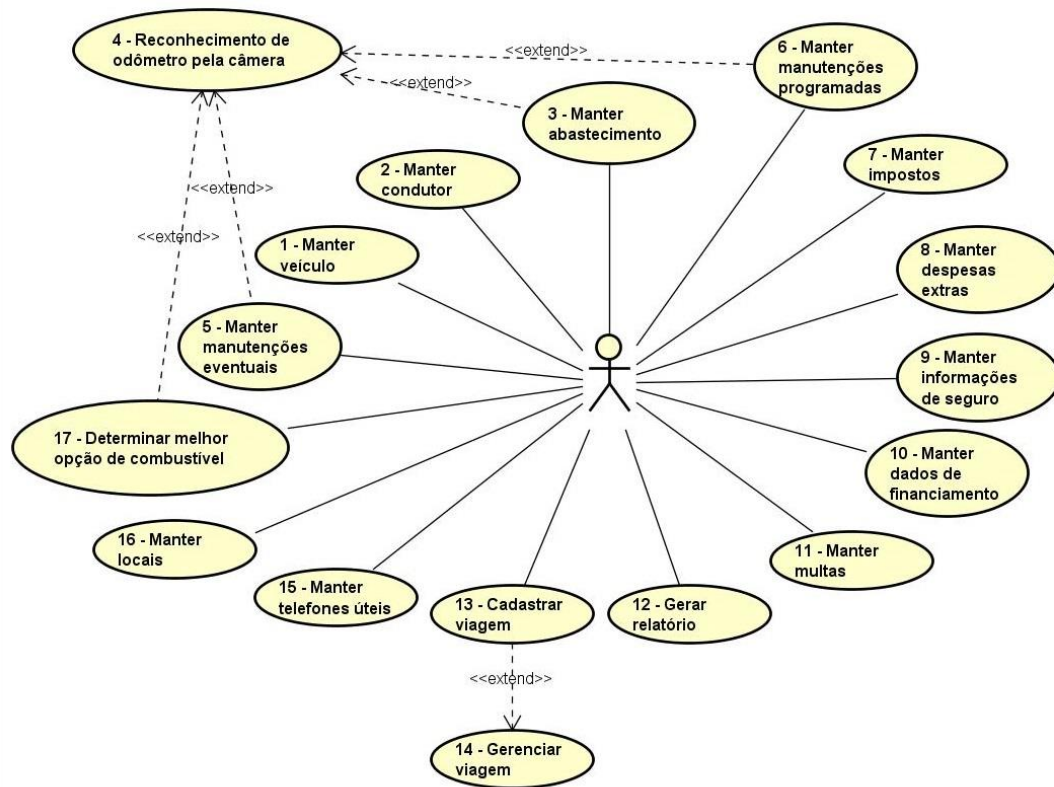


Continua na próxima página.

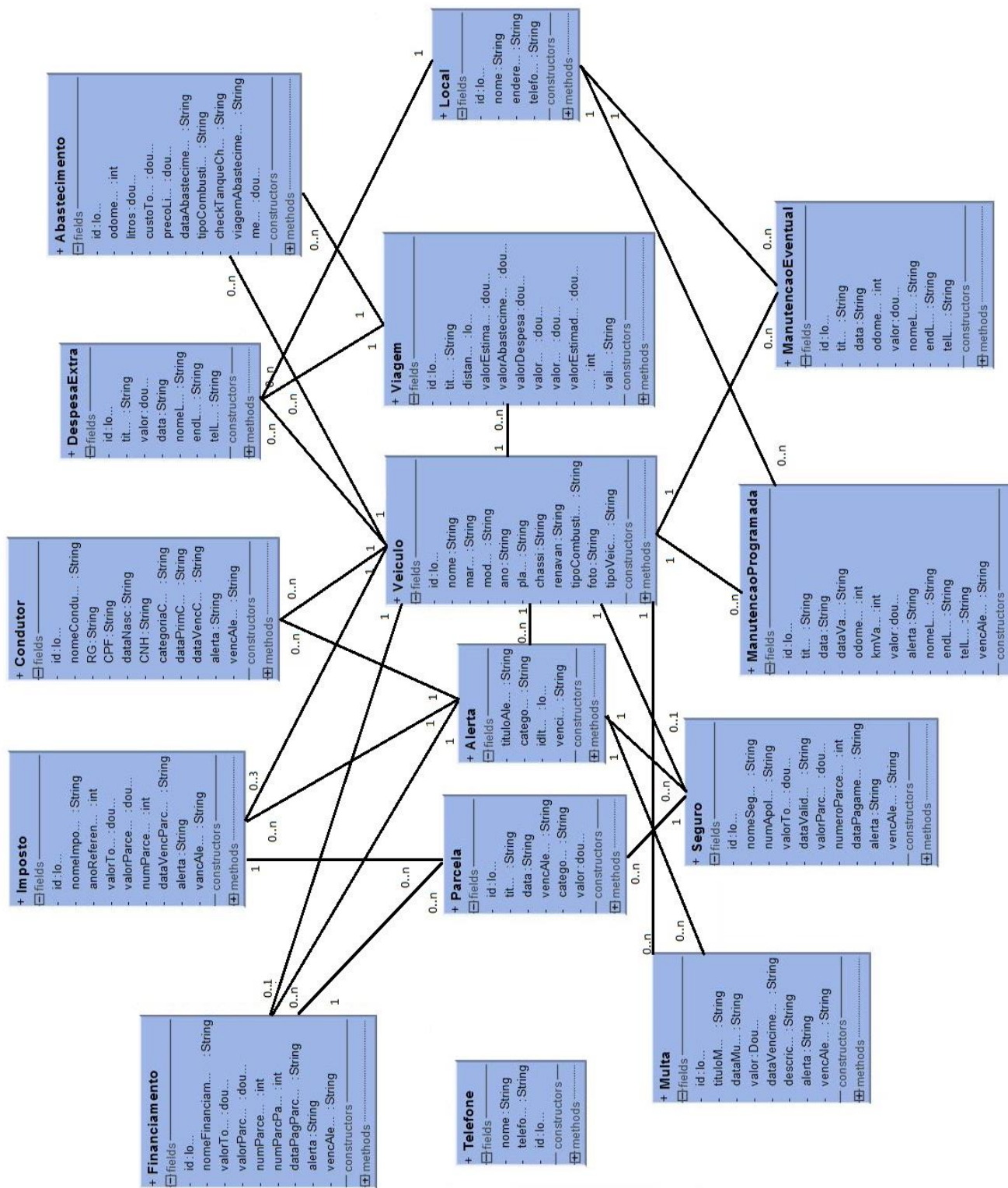
## APÊNDICE B – DIAGRAMA GERAL DE ATIVIDADES – CONTINUAÇÃO



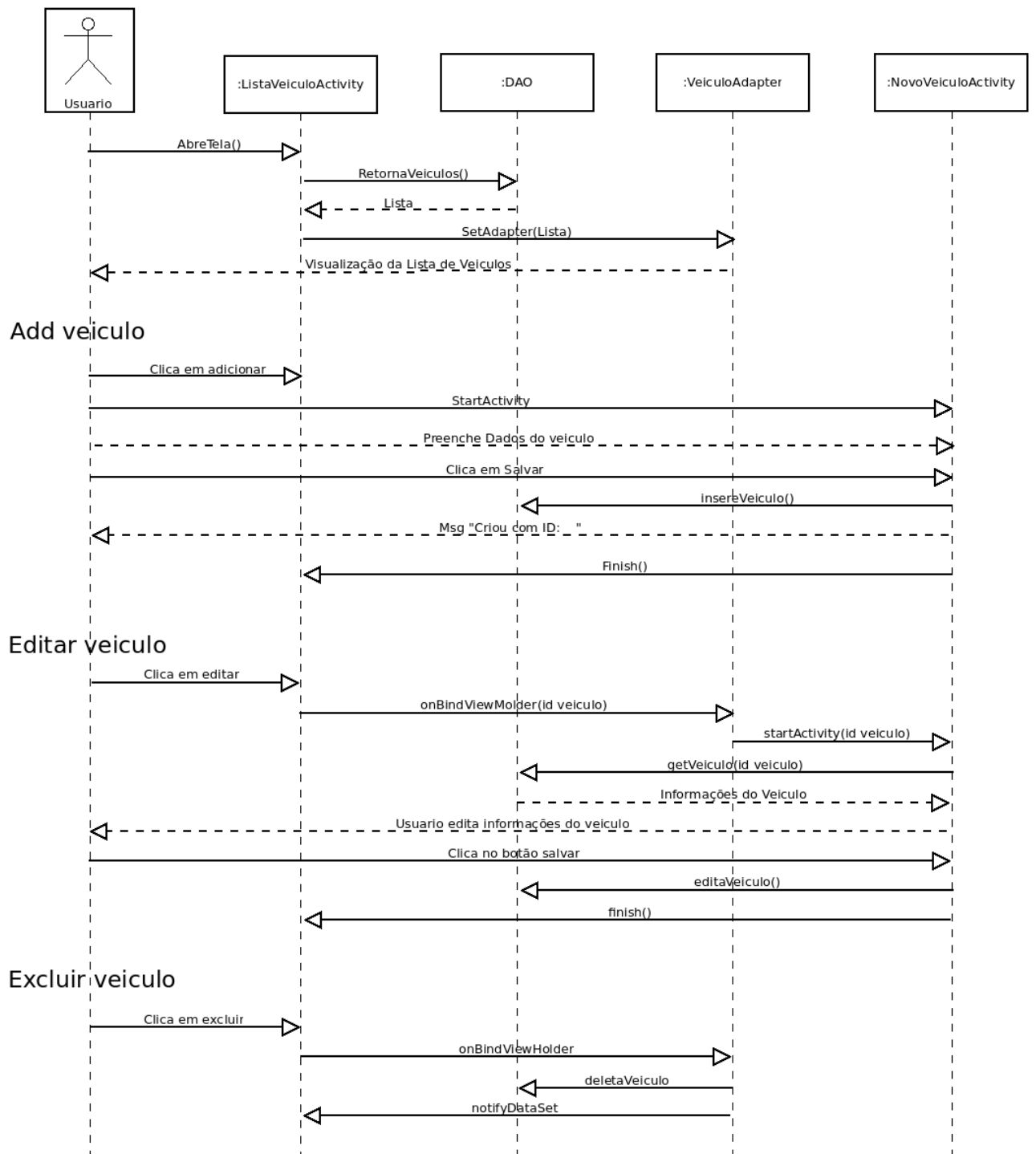
## APÊNDICE C – DIAGRAMA DE CASOS DE USO



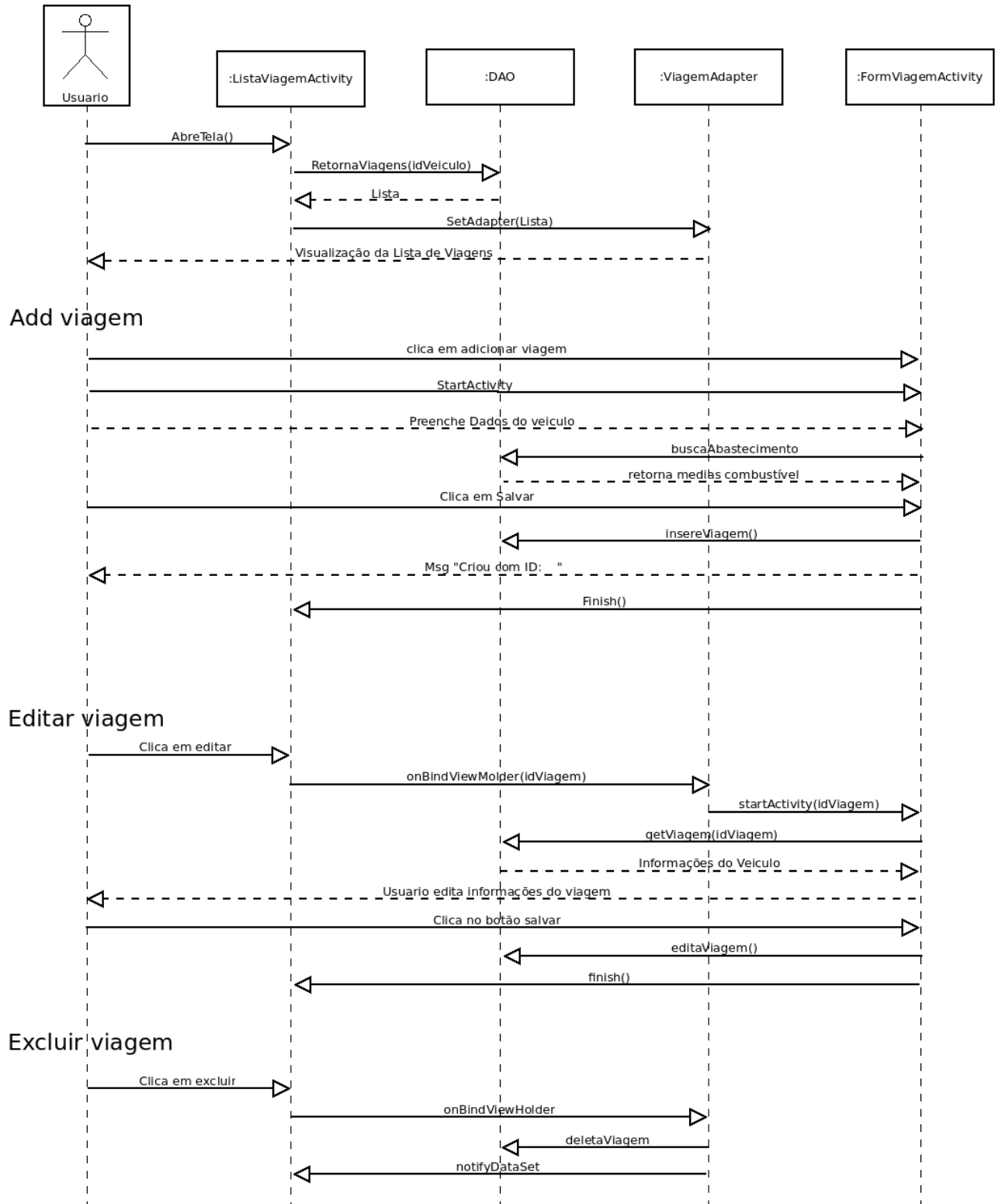
# APÊNDICE D – DIAGRAMA DE CLASSE



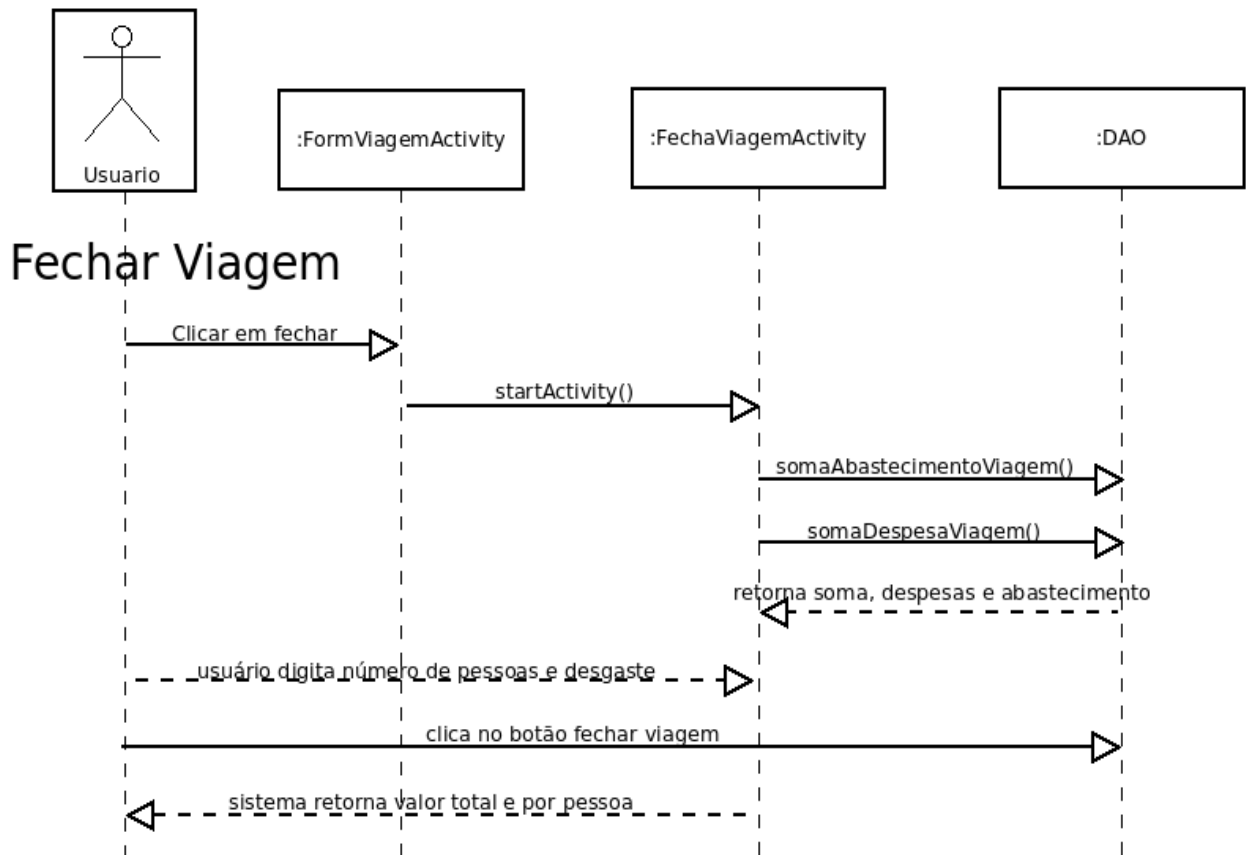
## APÊNDICE E – DIAGRAMA DE SEQUÊNCIA - MANTER VEÍCULO



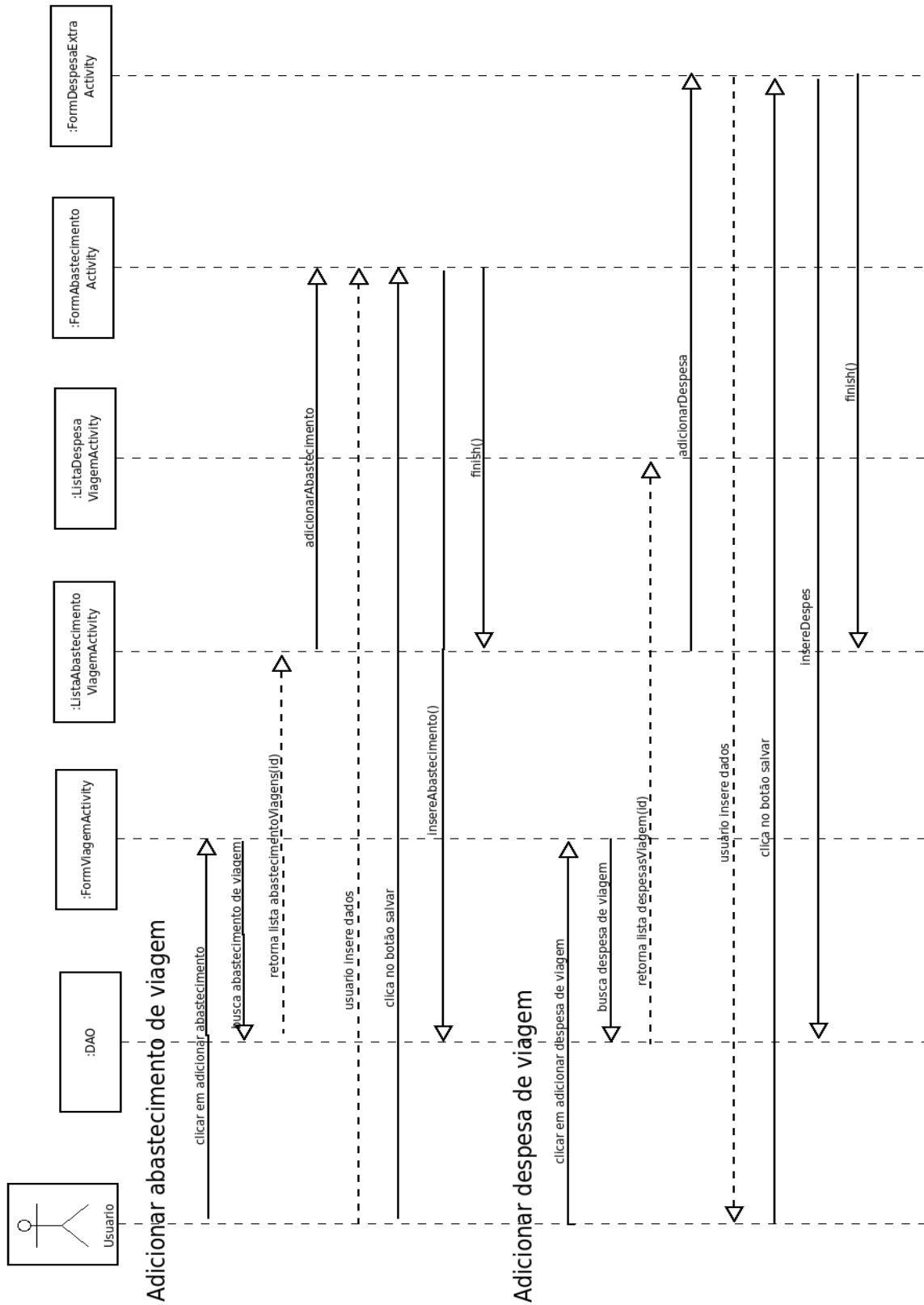
## APÊNDICE F – DIAGRAMA DE SEQUÊNCIA - MANTER VIAGEM



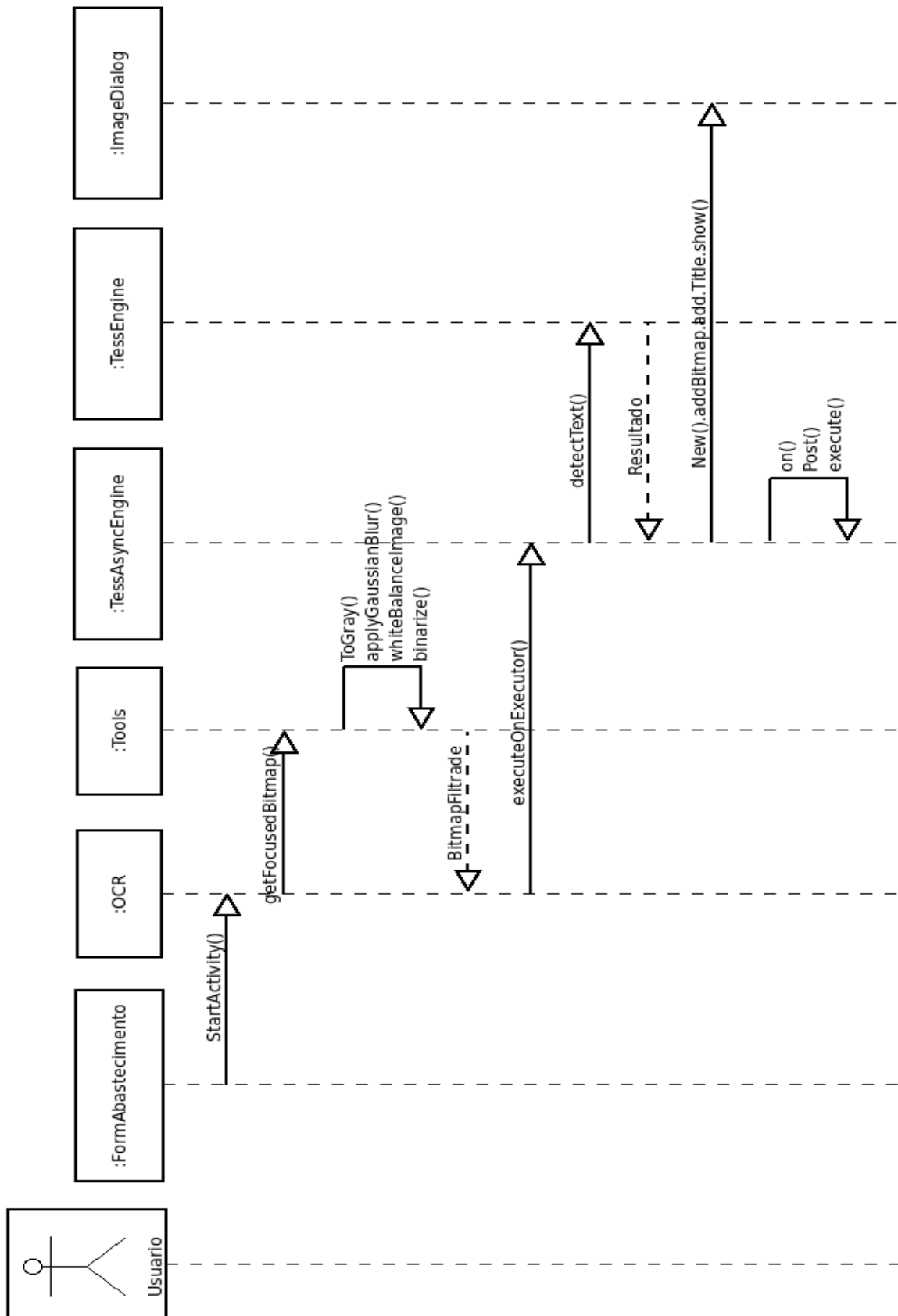
## APÊNDICE G – DIAGRAMA DE SEQUÊNCIA - FECHAR VIAGEM



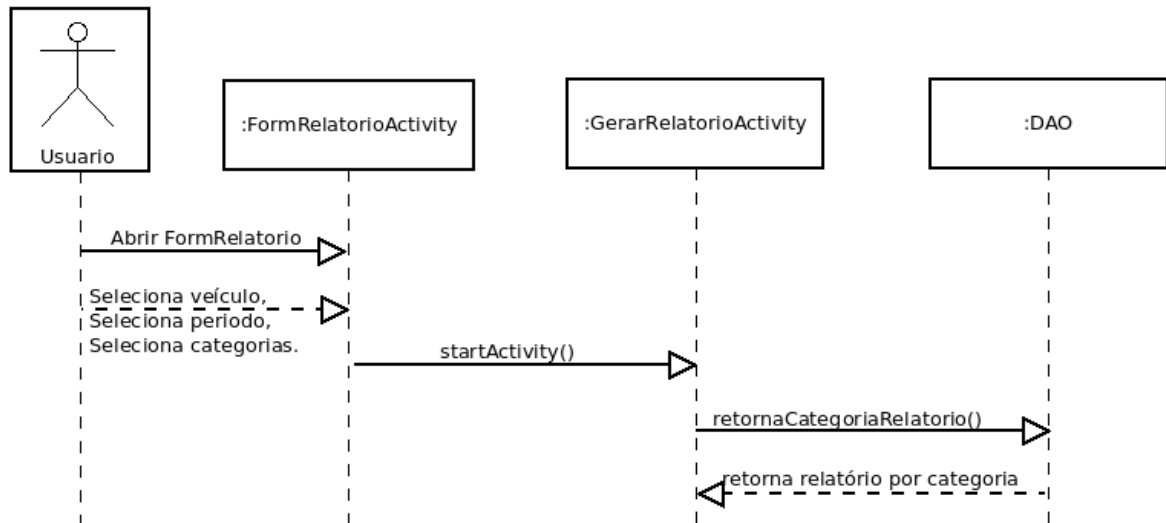
## APÊNDICE H – DIAGRAMA DE SEQUÊNCIA – DESPESAS VIAGEM



## APÊNDICE I – DIAGRAMA DE SEQUÊNCIA – RECONHECIMENTO DE IMAGEM



## APÊNDICE J – DIAGRAMA DE SEQUÊNCIA – RELATÓRIO



## APÊNDICE K – DIAGRAMA ENTIDADE-RELACIONAMENTO



## APÊNDICE L – PROTÓTIPOS DE TELA: APLICATIVO AIDAUTO



Figura 26 - PROTÓTIPO TELA INICIAL E MENU

## APÊNDICE M – DESCRIÇÃO DOS CASOS DE USO

### UC 01 – Manter Veículo

**Sumário:** Gerenciar veículos através da inclusão, alteração e remoção.

**Ator Principal:** Usuário

**Pré-condição:** N/A

#### Fluxo Principal

1. O aplicativo fornece a lista de veículos cadastrados
2. Pressionando o ícone + o formulário cadastro de um novo veículo é exibido com os campos: Nome, marca, modelo, ano, placa, número do chassi, renavan, tipo de combustível e uma imagem.
3. Segurando o dedo sobre um veículo da lista o aplicativo fornece as opções de edição (**FA1**) e exclusão (**FA2**) do veículo em foco.

#### FA1 Fluxo Alternativo (Edição)

- a. O usuário solicita a edição dos dados do veículo.
- b. O aplicativo exibe um formulário com os campos para a alteração (Nome, marca, modelo, ano, placa, número do chassi, renavan, tipo de combustível e imagem).
- c. O usuário altera os campos com os novos dados.
- d. O aplicativo verifica se o campo nome não está em branco.
- e. O usuário clica no botão salvar e encerra o processo.

#### FA2 Fluxo Alternativo (Exclusão)

- a. O usuário solicita a exclusão dos dados do veículo.
- b. O aplicativo exibe uma mensagem confirmando a ação do usuário.
- c. Após a confirmação o aplicativo exclui os dados do veículo.



Figura 27 - PROTÓTIPO TELA MANTER VEÍCULO

#### UC 02 – Manter Condutor

**Sumário:** Gerenciar condutores através da inclusão, alteração e remoção.

**Ator Principal:** Usuário

**Pré-condição:** N/A

#### Fluxo Principal

1. O aplicativo fornece a lista de condutores cadastrados
2. Pressionando o ícone + o formulário cadastro de um novo condutor é exibido com os campos: Nome, RG, CPF, data de nascimento, número da CNH, categoria, data da primeira habilitação e data de vencimento.
3. Segurando o dedo sobre um condutor da lista o aplicativo fornece as opções de edição (**FA1**) e exclusão (**FA2**) do condutor em foco.

#### FA1 Fluxo Alternativo (Edição)

- a. O usuário solicita a edição dos dados de um condutor.
- b. O aplicativo exibe um formulário com os campos para a alteração (Nome, RG, CPF, data de nascimento, número da CNH, categoria, data da primeira habilitação e data de vencimento).
- c. O usuário altera os campos com os novos dados.
- d. O aplicativo verifica se o campo nome não está em branco.
- e. O usuário clica no botão salvar e encerra o processo.

#### FA2 Fluxo Alternativo (Exclusão)

- a. O usuário solicita a exclusão dos dados do condutor.
- b. O aplicativo exibe uma mensagem confirmando a ação do usuário.
- c. Após a confirmação o aplicativo exclui os dados do condutor.

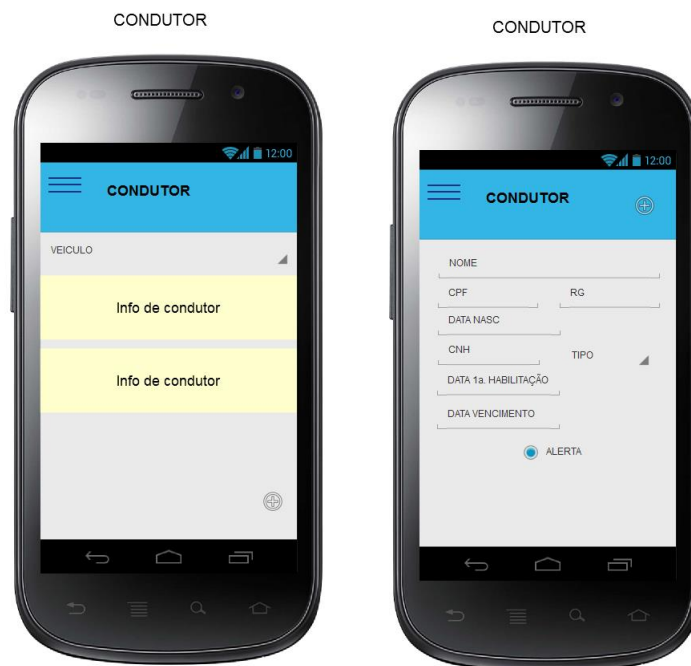


Figura 28 - PROTÓTIPO TELA MANTER CONDUTOR

### UC 03 – Manter Abastecimento

**Sumário:** Gerenciar abastecimentos através de inclusão, edição e exclusão

**Ator:** Usuário

**Pré-condição:** N/A

#### Fluxo Principal

1. O aplicativo fornece a lista de abastecimentos cadastrados
2. Pressionando o ícone + o formulário cadastro de um novo abastecimento é exibido com os campos: Quilometragem atual, quantidade de litros abastecido, preço por litro, custo total, tipo de combustível e data do abastecimento.
3. Segurando o dedo sobre um abastecimento da lista o aplicativo fornece as opções de edição (**FA1**) e exclusão (**FA2**) do abastecimento em foco.

#### FA1 Fluxo Alternativo (Edição)

- a. O usuário solicita a edição dos dados de um abastecimento.
- b. O aplicativo exibe um formulário com os campos para a alteração (Quilometragem Atual, quantidade de litros abastecido, preço por litro, custo total, tipo de combustível e data do abastecimento).
- c. O usuário altera os campos com os novos dados.
- d. O usuário clica no botão salvar e encerra o processo.

#### FA2 Fluxo Alternativo (Exclusão)

- a. O usuário solicita a exclusão dos dados do abastecimento.
- b. O aplicativo exibe uma mensagem confirmando a ação do usuário.
- c. Após a confirmação o aplicativo exclui os dados do abastecimento.



Figura 29 - PROTÓTIPO TELA CADASTRAR ABASTECIMENTO

#### UC 04 – Reconhecimento de odômetro pela câmera

**Sumário:** Reconhecer a quilometragem registrada do odômetro do veículo utilizando a câmera do dispositivo.

**Ator:** Usuário

**Pré-condição:** O dispositivo possuir câmera.

##### Fluxo Principal

1. O usuário clica no botão para iniciar um reconhecimento de odômetro pela câmera.
2. O usuário posiciona o foco sobre o número que consta no odômetro.
3. O usuário clica no botão de analisar para começar o reconhecimento.
4. O aplicativo retorna o número reconhecido.
5. Caso não seja o número correspondente: **FE1**.
6. Caso não seja reconhecido nenhuma informação: **FE2**.

##### FE1 Fluxo de Exceção (Retorno não compatível)

- a. O usuário solicita novo reconhecimento.

##### FE2 Fluxo de Exceção (Não reconhece nenhuma informação)

- a. O aplicativo exibe uma mensagem que não foi possível realizar o reconhecimento.
- b. O usuário solicita novo reconhecimento

## UC 05 – Manter Manutenções Eventuais

**Sumário:** Gerenciar manutenções eventuais através de inclusão, edição e exclusão.

**Ator:** Usuário

**Pré-condição:** N/A

### Fluxo Principal

1. O aplicativo fornece a lista de manutenções eventuais cadastradas
2. Pressionando o ícone + o formulário cadastro de uma nova manutenção eventual é exibido com os campos: Título da manutenção, peça ou serviço que foi realizado, valor do serviço e o local da oficina.
3. Segurando o dedo sobre uma manutenção eventual da lista o aplicativo fornece as opções de edição (**FA1**) e exclusão (**FA2**) da manutenção eventual em foco.

### FA1 Fluxo Alternativo (Edição)

- a. O usuário solicita a edição dos dados de uma manutenção eventual.
- b. O aplicativo exibe um formulário com os campos para a alteração (Título da manutenção, peça ou serviço que foi realizado, valor do serviço e o local da oficina.).
- c. O usuário altera os campos com os novos dados.
- d. O usuário clica no botão salvar e encerra o processo.

### FA2 Fluxo Alternativo (Exclusão)

- a. O usuário solicita a exclusão dos dados da manutenção eventual.
- b. O aplicativo exibe uma mensagem confirmando a ação do usuário.
- c. Após a confirmação o aplicativo exclui os dados da manutenção eventual.

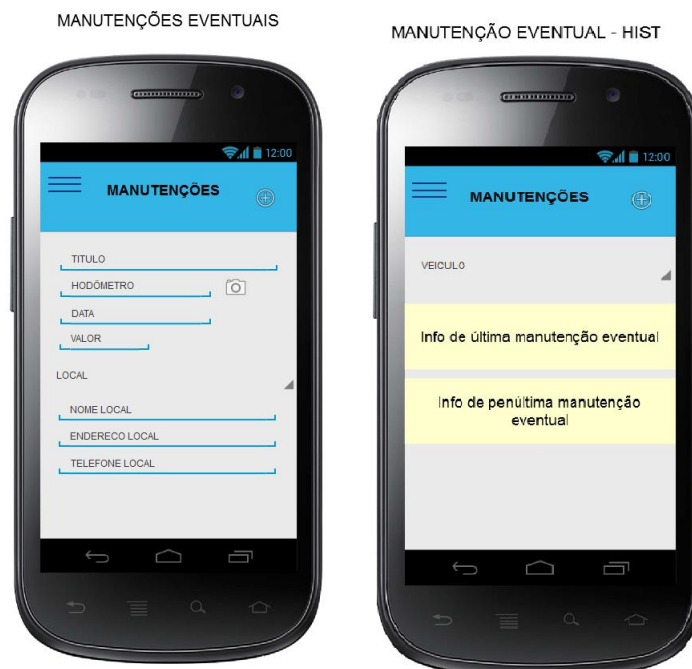


Figura 30 - PROTÓTIPO TELAS MANUTENÇÕES EVENTUAIS

## UC 06 – Manter Manutenções Programadas

**Sumário:** Gerenciar manutenções programadas através de inclusão, edição e exclusão.

**Ator:** Usuário

**Pré-condição:** N/A

#### **Fluxo Principal**

1. O aplicativo fornece a lista de manutenções programadas cadastradas
2. Pressionando o ícone + o formulário cadastro de uma nova manutenção programada é exibido com os campos: Título da manutenção, categoria da manutenção, valor, data, quilometragem atual e local onde foi efetuada a manutenção. De acordo com a categoria, o aplicativo sugere que criar um alerta com base nas sugestões que o aplicativo fornece ou o usuário poderá criar seu próprio alerta.
3. Segurando o dedo sobre uma manutenção programada da lista o aplicativo fornece as opções de edição (**FA1**) e exclusão (**FA2**) da manutenção programada em foco.

#### **FA1 Fluxo Alternativo (Editar)**

- a. O usuário solicita a edição dos dados de uma manutenção programada.
- b. O aplicativo exibe um formulário com os campos para a alteração (Título da manutenção, categoria da manutenção, valor, data, quilometragem atual e local onde foi efetuada a manutenção). O usuário pode alterar as informações do alerta, podendo ativa-lo ou desativa-lo.
- c. O usuário altera os campos com os novos dados.
- d. O usuário clica no botão salvar e encerra o processo.

#### **FA2 Fluxo Alternativo (Excluir)**

- a. O usuário solicita a exclusão dos dados da manutenção programada.
- b. O aplicativo exibe uma mensagem confirmando a ação do usuário.
- c. Após a confirmação o aplicativo exclui os dados da manutenção programada.



Figura 31 - PROTÓTIPO TELAS MANter MANUTENÇÕES PROGRAMADAS

### UC 07 – Manter Impostos

**Sumário:** Gerenciar o pagamento de impostos através de inclusão, edição e exclusão.

**Ator:** Usuário

**Pré-condição:** N/A

#### Fluxo Principal

1. O aplicativo exibe uma lista com os impostos cadastrados.
2. O usuário escolhe a categoria de imposto entre: IPVA (**FA1**), Licenciamento e Seguro Obrigatório (DPVAT).
3. Pressionando o ícone + o formulário cadastro de um novo imposto é exibido com os campos: Ano de referência, valor e vencimento. Usuário pode habilitar alarmes para as datas de vencimento.
4. Segurando o dedo sobre um imposto da lista o aplicativo fornece as opções de edição (**FA2**) e exclusão (**FA3**) do imposto em foco.

#### FA1 Fluxo Alternativo (IPVA)

- a. Além dos campos Ano de Referência, valor e vencimento, o usuário pode cadastrar a quantidade de parcelas, o valor de cada uma e a data de vencimento de cada parcela, podendo habilitar alarmes para cada vencimento.

#### FA2 Fluxo Alternativo (Edição)

- a. O usuário solicita a edição dos dados de um imposto.
- b. O aplicativo exibe um formulário com os campos para a alteração (Ano de Referência, valor e data de vencimento). O usuário pode alterar as informações do alerta, podendo ativa-lo ou desativa-lo.

- c. O usuário altera os campos com os novos dados.
- d. O usuário clica no botão salvar e encerra o processo.

#### FA3 Fluxo Alternativo (Exclusão)

- a. O usuário solicita a exclusão dos dados do imposto.
- b. O aplicativo exibe uma mensagem confirmando a ação do usuário.
- c. Após a confirmação o aplicativo exclui os dados do imposto.

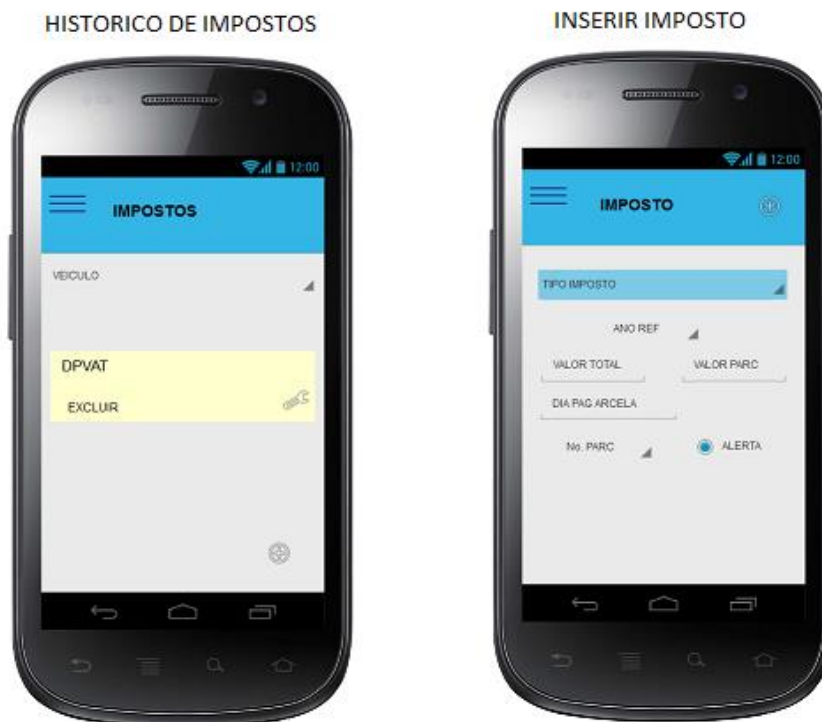


Figura 32 - PROTÓTIPO TELAS MANTER IMPOSTOS

#### UC 08 – Manter Despesas Extras

**Sumário:** Gerenciar os gastos com despesas extras através de inclusão, edição e exclusão de registros.

**Ator:** Usuário

**Pré-condição:** N/A

#### Fluxo Principal

1. O aplicativo fornece uma lista com as despesas extras cadastradas.
2. Pressionando o ícone + o formulário cadastro de uma nova despesa extra é exibido com os campos: Título da despesa, valor, data e local onde foi efetuada a despesa.
3. Segurando o dedo sobre uma despesa extra da lista o aplicativo fornece as opções de edição (**FA1**) e exclusão (**FA2**) da despesa extra em foco.

#### FA1 Fluxo Alternativo (Editar)

- a. O usuário solicita a edição dos dados de uma despesa extra.
- b. O aplicativo exibe um formulário com os campos para a alteração (Título da despesa extra, valor, e o local).
- c. O usuário altera os campos com os novos dados.
- d. O usuário clica no botão salvar e encerra o processo.

**FA2 Fluxo Alternativo (Exclusão)**

- a. O usuário solicita a exclusão dos dados da despesa extra.
- b. O aplicativo exibe uma mensagem confirmando a ação do usuário.
- c. Após a confirmação o aplicativo exclui os dados da despesa extra.

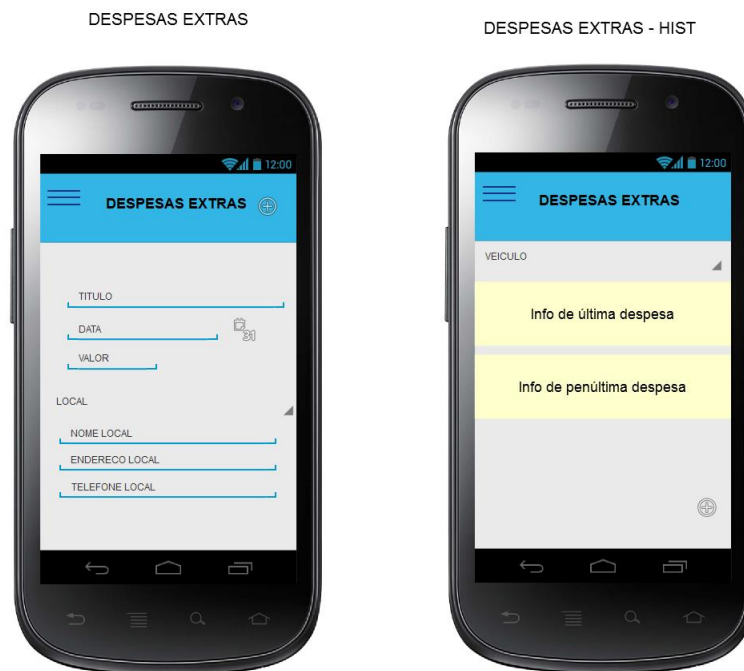


Figura 33 - PROTÓTIPO TELAS MANTER DESPESAS EXTRAS

**UC 09 – Manter Informações de Seguro**

**Sumário:** Gerenciar informações sobre seguro através de inclusão, edição e exclusão.

**Ator:** Usuário

**Pré-condição:** N/A

**Fluxo Principal**

1. Pressionando o ícone + o formulário para cadastro de informações do seguro é exibido com os campos: Nome da seguradora, número da apólice, o valor total, a data de validade, valor das parcelas, número de parcelas e data de pagamento das parcelas.
2. O usuário pode criar alertas para a data de vencimento das parcelas, no momento em que as informações do seguro forem preenchidas.
3. Segurando o dedo sobre as informações do seguro, o aplicativo fornece as opções de edição (**FA1**) e exclusão (**FA2**) do seguro do veículo em foco.

**FA1 Fluxo Alternativo (Editar)**

- a. O usuário solicita a edição dos dados do seguro.
- b. O aplicativo exibe um formulário com os campos para a alteração (Nome da seguradora, número da apólice, o valor total, a data de validade, valor das parcelas, número de parcelas e data de pagamento das parcelas).
- c. O usuário altera os campos com os novos dados.
- d. O usuário clica no botão salvar e encerra o processo.

**FA2 Fluxo Alternativo (Exclusão)**

- a. O usuário solicita a exclusão dos dados do seguro.
- b. O aplicativo exibe uma mensagem confirmando a ação do usuário.
- c. Após a confirmação o aplicativo exclui os dados do seguro do veículo em foco.



Figura 34 - PROTÓTIPO TELAS MANTER SEGURO

#### UC 10 – Manter dados de Financiamento

**Sumário:** Gerenciar as informações sobre o financiamento do veículo através de inclusão, edição e exclusão das informações de financiamento.

**Ator:** Usuário

**Pré-condição:** N/A

#### Fluxo Principal

1. Pressionando o ícone + o formulário para cadastro de informações do financiamento é exibido com os campos: Nome da instituição financeira, o valor total, o número de parcelas, a quantidade de parcelas pagas e a data de vencimento das parcelas.
2. O usuário pode criar alertas para a data de vencimento das parcelas, no momento em que as informações do financiamento forem preenchidas.
3. Segurando o dedo sobre as informações do seguro, o aplicativo fornece as opções de edição (**FA1**) e exclusão (**FA2**) do financiamento do veículo em foco.

#### FA1 Fluxo Alternativo (Editar)

- a. O usuário solicita a edição dos dados do financiamento.
- b. O aplicativo exibe um formulário com os campos para a alteração (Nome da seguradora, número da apólice, o valor total, a data de validade, valor das parcelas, número de parcelas e data de pagamento das parcelas).
- c. O usuário altera os campos com os novos dados.
- d. O usuário clica no botão salvar e encerra o processo.

**FA2 Fluxo Alternativo (Exclusão)**

- a. O usuário solicita a exclusão dos dados do financiamento.
- b. O aplicativo exibe uma mensagem confirmando a ação do usuário.
- c. Após a confirmação o aplicativo exclui os dados do financiamento do veículo em foco.

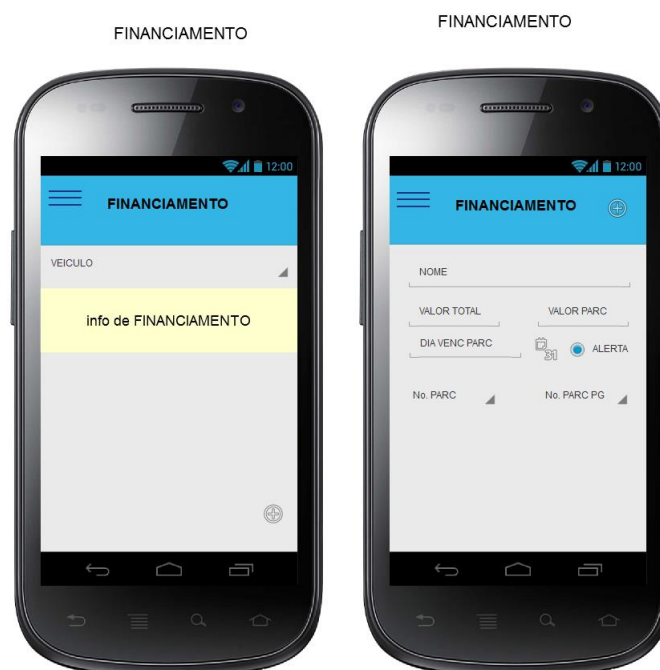


Figura 35 - PROTÓTIPO TELAS MANTER FINANCIAMENTO

**UC 11 – Manter Multas**

**Sumário:** Gerenciar cadastro de multas do veículo e/ou do condutor.

**Ator:** Usuário

**Pré-condição:** N/A

**Fluxo Principal**

1. O aplicativo fornece uma lista com as multas cadastradas de todos os veículos e condutores.
2. Pressionando o ícone + o formulário cadastro de uma nova multa é exibido com os campos: Data da infração, valor, dia do vencimento e descrição da multa.
3. Segurando o dedo sobre uma multa da lista o aplicativo fornece as opções de edição (**FA1**) e exclusão (**FA2**) da multa em foco.

**FA1 Fluxo Alternativo (Edição)**

- a. O usuário solicita a edição dos dados da multa.
- b. O aplicativo exibe um formulário com os campos para a alteração (Data da infração, valor, dia do vencimento e descrição da multa).
- c. O usuário altera os campos com os novos dados.
- d. O usuário clica no botão salvar e encerra o processo.

**FA2 Fluxo Alternativo (Exclusão)**

- a. O usuário solicita a exclusão dos dados da multa.

- b. O aplicativo exibe uma mensagem confirmando a ação do usuário.
- c. Após a confirmação o aplicativo exclui os dados da multa do veículo e/ou do condutor.



Figura 36 - PROTÓTIPO TELAS MANTER MULTAS

#### UC 12 – Gerar Relatórios

**Sumário:** Gerar relatório de acordo com as necessidades do usuário.

**Ator:** Usuário

**Pré-condição:** Ter ao menos um veículo cadastrado no sistema.

#### Fluxo Principal

1. O usuário seleciona os campos de seu interesse para o relatório e clica em gerar relatório.
2. O usuário visualiza o relatório.

## RELATÓRIO

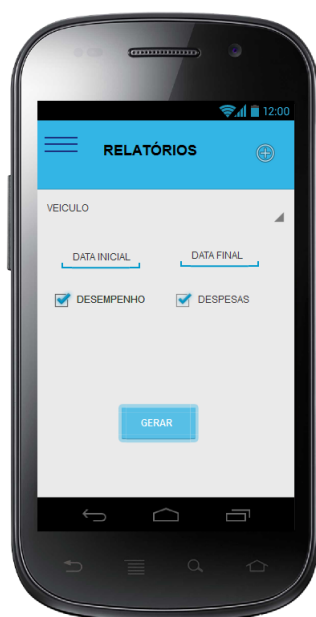


Figura 37 - PROTÓTIPO TELA RELATÓRIO

**UC 13 – Cadastrar Viagens**

**Sumário:** Gerenciar gastos com viagens.

**Ator:** Usuário

**Pré-condição:** N/A

**Fluxo Principal**

1. O usuário seleciona a opção viagem.
2. Pressionando o ícone, o formulário de cadastro de uma nova viagem é exibido, disponibilizando os campos: Título da viagem e a distancia a ser percorrida.
3. O sistema busca a média recente dos valores de combustível.
4. O aplicativo retorna o quanto será gasto em combustível de acordo com médias previamente cadastradas e o preço informado.
5. Mantendo pressionado sobre manutenção programada da lista o aplicativo fornece as opções de edição **(FA1)** exclusão **(FA2)** e adicionar custos adicionais da rota em foco **(FA3)**.

**FA1 Fluxo Alternativo (Edição)**

- a. O usuário solicita a edição dos dados da viagem.
- b. O aplicativo exibe um formulário com os campos para a alteração (Título da viagem, distância a ser percorrida e capacidade do tanque de combustível).
- c. O usuário altera os campos com os novos dados.
- d. O usuário clica no botão salvar e encerra o processo.

**FA2 Fluxo Alternativo (Exclusão)**

- a. O usuário solicita a exclusão dos dados da viagem.
- b. O aplicativo exibe uma mensagem confirmando a ação do usuário.
- c. Após a confirmação o aplicativo exclui os dados da viagem.

**FA3 Fluxo Alternativo (Adicionar custos adicionais)**

- a. O usuário solicita a adição de custos adicionais para a viagem.
- b. O usuário cadastra dados de custo como: adicionar abastecimentos durante a viagem e adicionar despesas.
- c. Marcando a opção de “fechar viagem”, o usuário informa quantas pessoas participaram da viagem e divide os custos da viagem entre os ocupantes.

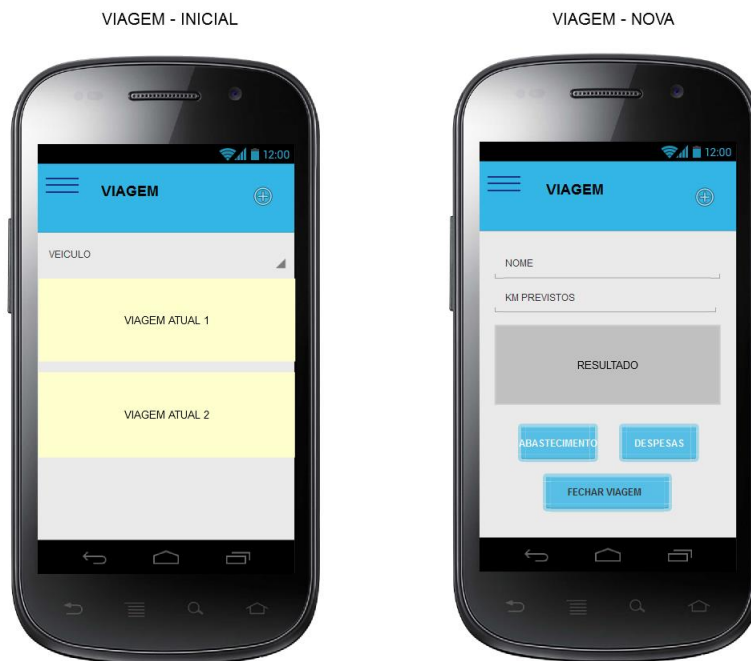


Figura 38 - PROTÓTIPO TELAS CADASTRAR VIAGEM

#### UC 14 – Gerenciar Viagem

**Sumário:** Adicionar custos adicionais à viagem.

**Ator:** Usuário

**Pré-condição:** (UC 13) Ter viagem cadastrada.

#### Fluxo Principal

1. O usuário cadastra dados de custo como: Abastecimentos durante a viagem e custos.
2. Marcando a opção de “fechar viagem”, o usuário informa quantas pessoas participaram da viagem e divide os custos da viagem entre os ocupantes.
3. O usuário inclui uma taxa de desgaste do veículo.
4. O sistema retorna um relatório de viagem onde traz o total gasto de combustível, os gastos com despesa e os gastos gerais.

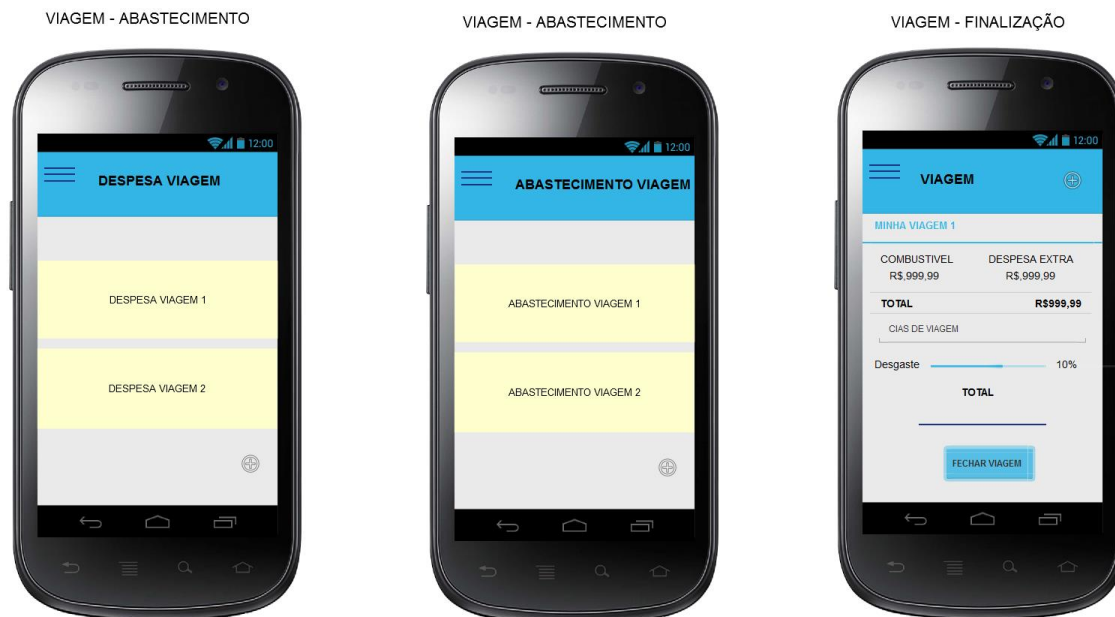


Figura 39 - PROTÓTIPO TELAS GERENCIAR VIAGEM

### UC 15 – Manter Telefones Úteis

**Sumário:** Gerenciar a lista de telefones úteis através de inclusão, edição e exclusão.

**Ator:** Usuário

**Pré-condição:** N/A

#### Fluxo Principal

1. O aplicativo fornece a lista de números telefônicos úteis cadastrados.
2. Pressionando o ícone + o formulário cadastro de um novo número telefônico é exibido com os campos: Nome e número.
3. Segurando o dedo sobre um número telefônico da lista o aplicativo fornece as opções de edição (**FA1**) e exclusão (**FA2**) do telefone em foco.

#### FA1 Fluxo Alternativo (Edição)

- a. O usuário solicita a edição dos dados de um telefone.
- b. O aplicativo exibe um formulário com os campos para a alteração (Nome e número).
- c. O usuário altera os campos com os novos dados.
- d. O usuário clica no botão salvar e encerra o processo.

#### FA2 Fluxo Alternativo (Exclusão)

- a. O usuário solicita a exclusão do telefone.
- b. O aplicativo exibe uma mensagem confirmando a ação do usuário.
- c. Após a confirmação o aplicativo exclui o telefone.



Figura 40 - PROTÓTIPO TELAS MANTER TELEFONES ÚTEIS

#### UC 16 – Manter Locais

**Sumário:** Gerencia locais de interesse do usuário através de inclusão, edição e exclusão.

**Ator:** Usuário

**Pré-condição:** N/A

#### Fluxo Principal

1. O aplicativo fornece a lista com os locais cadastrados tanto pelos casos de uso: **UC05 – Manter Manutenções Programadas**, **UC06 – Manter Manutenções Eventuais** e por este caso de uso.
2. Pressionando o ícone + o formulário cadastro de um novo local é exibido com os campos: Nome, endereço e telefone.
3. Segurando o dedo sobre um local da lista o aplicativo fornece as opções de edição (**FA1**) e exclusão (**FA2**) do local em foco.

#### FA1 Fluxo Alternativo (Edição)

- a. O usuário solicita a edição dos dados de um local.
- b. O aplicativo exibe um formulário com os campos para a alteração (Nome, endereço e telefone).
- c. O usuário altera os campos com os novos dados.
- d. O usuário clica no botão salvar e encerra o processo.

#### FA2 Fluxo Alternativo (Exclusão)

- a. O usuário solicita a exclusão do local.
- b. O aplicativo exibe uma mensagem confirmando a ação do usuário.
- c. Após a confirmação o aplicativo exclui o local.

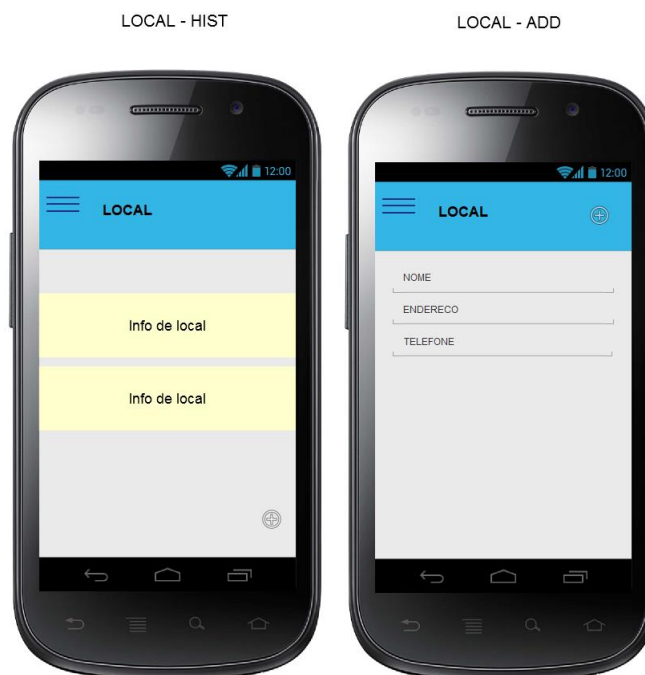


Figura 41 - PROTÓTIPO TELAS MANTER LOCAIS

#### UC 17 – Determinar Melhor Opção de Combustível (Etanol x Gasolina)

**Sumário:** Determinar qual combustível reverterá um melhor custo benefício para veículos bicompostíveis.

**Ator:** Usuário

**Pré-condição:** Para uma média personalizada para cada veículo, o usuário precisa ter cadastrado abastecimentos completos com ambos os combustíveis a serem comparados.

#### Fluxo Principal

1. O usuário informa o preço por litro de ambos os combustíveis.
2. Se o aplicativo não tiver as informações necessárias para calcular a razão personalizada de acordo com a média do veículo, o cálculo será estático (o etanol rende em média 30% menos que a gasolina).



Figura 42 - PROTÓTIPO TELA MELHOR COMBUSTÍVEL

## APÊNDICE N – TESTES CAIXA PRETA

### Descrição do caso:

Caso 1: Cadastrar Veículo.

### Procedimento da execução:

- Usuário preenche campos: Nome e Placa;
- Clica no botão.

### Resultado esperado:

Sistema retorna mensagem inserida com sucesso' e retorna no *card* de veículos cadastrados.

**Resultado encontrado:** Figura 43



Figura 43 - RESULTADO TESTE CADASTRAR VEÍCULO

### Descrição do caso:

Caso 2: Cadastrar Condutor.

### Procedimento da execução:

- Usuário preenche campos: Nome do Condutor, Número do RG, Número do CPF, Número da CNH e Categoria da CNH;
- Clica no botão '*Check-mark*'.

### Resultado esperado:

Sistema retorna mensagem 'Cadastrado com sucesso' e retorna na tela 'Condutores' para o *card* com o nome do condutor.

**Resultado encontrado:** Figura 44.



Figura 44 - RESULTADO TESTE CADASTRAR CONDUTOR

**Descrição do caso:**

Caso 3: Cadastrar Manutenção Programada.

**Procedimento da execução:**

- Usuário seleciona a opção 'Manutenções Programadas';
- Seleciona o botão '+';
- Seleciona a opção desejada de manutenção;
- Insere o valor da manutenção;
- Seleciona uma data válida no campo 'Data da Manutenção';
- Seleciona uma data válida no campo 'Válida até';
- Clica no botão 'Check-mark'.

**Resultado esperado:**

O sistema retorna 'Inserido com sucesso', em seguida retorna para o *card* com a manutenção cadastrada.

**Resultado encontrado:** Figura 45.

**Descrição do caso:**

Caso 4: Cadastrar Manutenção Eventual.

**Procedimento da execução:**

- Usuário seleciona a opção 'Manutenções Eventuais';
- Preenche campo 'Título da Manutenção';
- Preenche campo 'Valor da Manutenção';
- Seleciona data válida no campo 'Data da Manutenção';
- Preenche o campo 'Odômetro';
- Clica no botão 'Check-mark'.

**Resultado esperado:**

Figura 45 - RESULTADO TESTE CADASTRAR MANUTENÇÃO PROGRAMADA



Figura 46 - RESULTADO TESTE MANUTENÇÃO EVENTUAL

O sistema retorna 'Inserido com sucesso', em seguida retorna para o *card* manutenção eventual cadastrada.

**Resultado encontrado:** Figura 46.

**Descrição do caso:** Caso 5: Cadastrar abastecimento.

**Procedimento da execução:**

- Usuário clica na opção 'Abastecimento';
- Seleciona o botão '+';
- Preenche o campo 'Quilometragem';
- Preenche o campo 'Quantidade de Litros';
- Preenche o campo 'Preço/L';
- Preenche o campo 'Custo';
- Seleciona o tipo de Combustível;
- Clica no Botão '*Check-mark*'.

**Resultado esperado:**

O sistema retorna 'Abastecimento Cadastrado', em seguida retorna na tela 'Abastecimentos' para o *card* do abastecimento cadastrado.

**Resultado encontrado:** Figura 47.

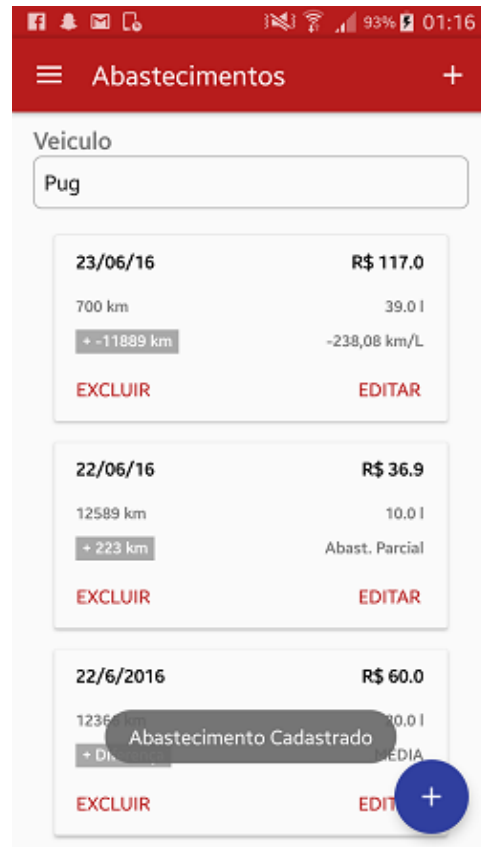


Figura 47 - RESULTADO TESTE CADASTRAR ABASTECIMENTO

**Descrição do caso:** Caso 6: Inserir Despesas Extras.

**Procedimento da execução:**

- Usuário seleciona a opção 'Despesa Extra';
- Preenche campo 'Valor da Despesa';
- Seleciona data válida no campo 'Data da Despesa';
- Clica no botão 'Check-mark'.

**Resultado esperado:** O sistema retorna 'Inserido com sucesso', em seguida retorna tela de Despesas Extras para o *card* com as despesas.

**Resultado encontrado:** Figura 48.

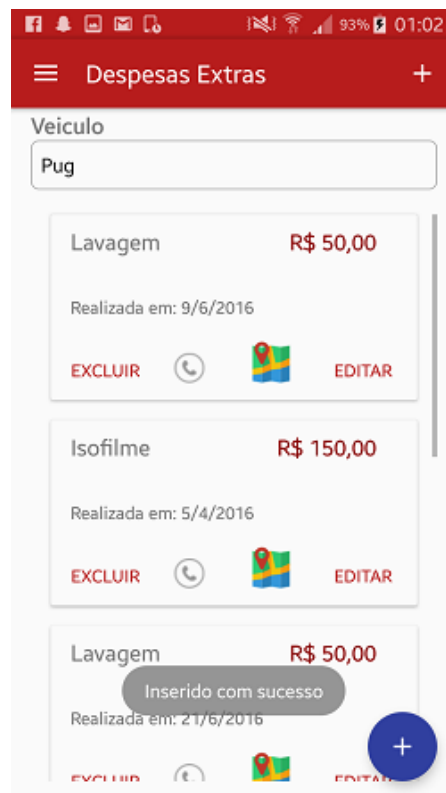


Figura 48 - RESULTADO TESTE CADASTRAR DESPESA EXTRA

**Descrição do caso:** Caso 7: Inserir Imposto.

**Procedimento da execução:**

- Usuário seleciona a opção 'Imposto';
- Seleciona o botão '+';
- Preenche o campo 'Ano Referência';
- Preenche o campo 'Valor Total';
- Preenche o campo 'Valor das Parcelas';
- Preenche o campo 'Número de Parcelas';
- Preenche o campo 'Dia do vencimento das parcelas';
- Clica no botão 'Check-mark'.

**Resultado esperado:**

O sistema retorna 'Inserido com sucesso', em seguida, retorna para o *card* com o imposto cadastrado.

**Resultado encontrado:** Figura 49.

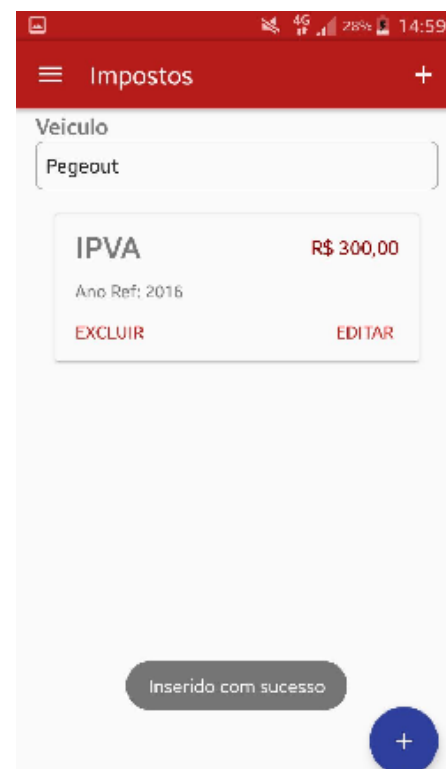


Figura 49 - RESULTADO TESTE CADASTRAR IMPOSTO

**Descrição do caso:** Caso 8: Cadastrar Multa.

**Procedimento da execução:**

- Usuário seleciona a opção 'Multa';
- Seleciona o botão '+';
- Preenche o campo 'Título da Multa';
- Preenche o campo 'Data da Multa';
- Preenche o campo 'Valor da Multa';
- Preenche o campo 'Data de Vencimento';
- Preenche o campo 'descrição';
- Clica no botão 'Check-mark'.

**Resultado esperado:** O sistema retorna 'Inserido com sucesso', em seguida retorna para o *card* com a multa cadastrada.

**Resultado encontrado:** Figura 50.



Figura 50 - RESULTADO TESTE CADASTRAR MULTA

**Descrição do caso:** Caso 9: Cadastrar Seguro.

**Procedimento da execução:**

- Usuário seleciona a opção 'Seguro';
- Seleciona o botão '+';
- Preenche o campo 'Título do Seguro';
- Preenche o campo 'Data do Seguro';
- Preenche o campo 'Valor do Seguro';
- Preenche o campo 'Data de Vencimento';
- Preenche o campo 'descrição';
- Clica no botão 'Check-mark'.

**Resultado esperado:** O sistema retorna 'Inserido com sucesso', em seguida retorna para o *card* com o seguro cadastrado.

**Resultado encontrado:** Figura 51.

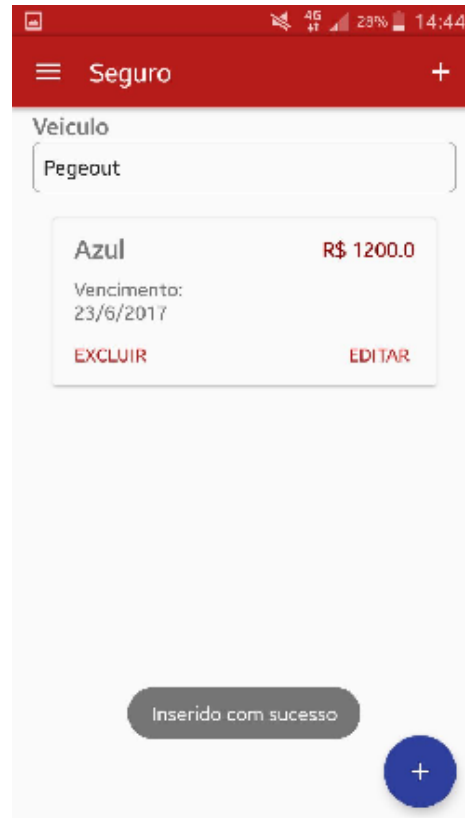


Figura 51 - RESULTADO TESTE CADASTRAR SEGURO

**Descrição do caso:** Caso 10: Cadastrar Financiamento.

**Procedimento da execução:**

- Usuário seleciona a opção 'Financiamento';
- Seleciona o botão '+';
- Preenche o campo 'Nome da Financeira';
- Preenche o campo 'Valor Total';
- Preenche o campo 'Valor das Parcelas';
- Preenche o campo 'Número das parcelas';
- Preenche o campo 'Dia do Vencimento';
- Preenche o campo 'Número das parcelas Pagas';
- Clica no botão 'Check-mark'.

**Resultado esperado:**

O sistema retorna 'Financiamento cadastrado com sucesso' em seguida retorna para o *card* com o financiamento cadastrado.

**Resultado encontrado:** Figura 52.

**Descrição do caso:** Caso 11: Cadastrar Local.

**Procedimento da execução:**

- Usuário seleciona a opção 'Locais';
- Seleciona o botão '+';
- Preenche o campo 'Nome do Local';
- Preenche o campo 'Endereço';
- Preenche o campo 'Telefone';
- Clica no botão 'Check-mark'.

**Resultado esperado:**

O sistema retorna 'Inserido com sucesso' em seguida retorna para o *card* com local cadastrado.



Figura 52 - RESULTADO TESTE CADASTRAR FINANCIAMENTO



Figura 53 - RESULTADO TESTE CADASTRAR LOCAL

**Resultado encontrado:** Figura 53.

**Descrição do caso:** Caso 12: Cadastrar Telefones Úteis.

**Procedimento da execução:**

Usuário seleciona a opção 'Telefones Úteis';

Seleciona o botão '+';

Preenche o campo 'Nome';

Preenche o campo 'Telefone';

Clica no botão 'Check-mark'.

**Resultado esperado:**

O sistema retorna 'Inserido com sucesso' em seguida retorna para o *card* com os telefones uteis já cadastrados.

**Resultado encontrado:** Figura 54.



Figura 54 - RESULTADO TESTE CADASTRAR TELEFONES ÚTEIS

**Descrição do caso:** Caso 13: Iniciar Viagem.

**Procedimento da execução:**

Usuário seleciona a opção 'Viagens';

Seleciona o botão '+';

Preenche o campo 'Título da viagem';

Preenche o campo 'Distância até o destino';

Clica no botão 'Check-mark'.

**Resultado esperado:**

O sistema retorna 'Inserido com sucesso', em seguida calcula o valor estimado de gasto baseado nas médias de gastos de combustíveis salvos e retorna para o *card* de viagens.

**Resultado encontrado:** Figura 55.



Figura 55 - RESULTADO TESTE INICIAR VIAGEM

**Descrição do caso:** Caso 14: Finalizar Viagem

**Procedimento da execução:**

- Usuário preenche campo informando o número de pessoas que estão dividindo a gasolina;
- Preenche campos 'valor de desgaste';
- Clica no botão 'Check-mark'.

**Resultado esperado:**

O sistema retorna 'Viagem fechada com Sucesso', em seguida apresenta *card* de viagens.

**Resultado encontrado:** Figura 56.



Figura 56 - RESULTADO TESTE FECHAR VIAGEM

**Descrição do caso:** Caso 15: Calcular Melhor Combustível.

**Procedimento da execução:**

- Usuário seleciona a opção 'Melhor Combustível';
- Seleciona o botão '+';
- Preenche o campo 'Preço da Gasolina';
- Preenche o campo 'Preço do Etanol';
- Clica no botão 'Calcular'.

**Resultado esperado:**

Sistema retorna na tela o nome do melhor combustível, em seguida retorna para o *card* com o nome do melhor combustível.

**Resultado encontrado:** Figura 57.

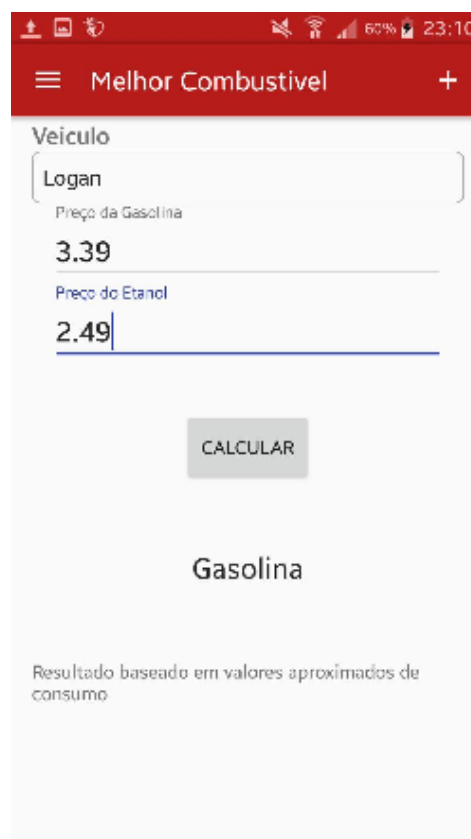


Figura 57 - RESULTADO TESTE MELHOR COMBUSTÍVEL

**Descrição do caso:** Caso 16: Gerar Relatório.

**Procedimento da execução:**

- Usuário seleciona a opção 'Relatório';
- Seleciona Data de Início';
- Seleciona Data Final';
- Seleciona o *check box* de opções;
- Clica no botão '*Check-mark*'.

**Resultado esperado:**

O sistema retorna 'Relatório gerado com sucesso' em seguida retorna para o *card* com relatório gerado.

**Resultado encontrado:**

Figura 58.



**Figura 58 - RESULTADO TESTE GERAR RELATÓRIO**