

UNIVERSIDADE FEDERAL DO PARANÁ

AMAURI DE MELO BARROS

**AUTOCASHER: SOFTWARE PARA GESTÃO
DE GASTOS AUTOMOTIVOS**

CURITIBA

2021

AMAURI DE MELO BARROS

**AUTOCASHER: SOFTWARE PARA GESTÃO
DE GASTOS AUTOMOTIVOS**

Monografia apresentada como requisito parcial à obtenção do grau de Especialista em Engenharia de Software, no Curso de Pós-Graduação em Engenharia de Software, Setor de Educação Profissional e Tecnológica (SEPT), da Universidade Federal do Paraná.

Orientador: Prof. Dr. Razer A. N. R. Montano

CURITIBA

2021



MINISTÉRIO DA EDUCAÇÃO
SETOR DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
CURSO DE PÓS-GRADUAÇÃO ENGENHARIA DE
SOFTWARE - 40001016231E1

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em ENGENHARIA DE SOFTWARE da Universidade Federal do Paraná foram convocados para realizar a arguição da Monografia de Especialização de **AMAURI DE MELO BARROS** intitulada: **AUTOCASHER: SOFTWARE PARA GESTAO DE GASTOS AUTOMOTIVOS**, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua aprovação no rito de defesa. A outorga do título de especialista está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 10 de Junho de 2021.

RAZER ANTHOM NIZER ROJAS MONTAÑO

Presidente da Banca Examinadora (UNIVERSIDADE FEDERAL DO PARANÁ)

JAIMÉ WOJCIECHOWSKI

Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

RESUMO

Atualmente no Brasil circulam cerca de 39 milhões de automóveis que possuem uma média de idade de 10 anos. Para garantir a segurança durante a circulação desses automóveis, são recomendadas manutenções preventivas realizadas a cada 10.000km ou 1 ano, preferivelmente o que acontecer primeiro. A gestão das manutenções e outros gastos automotivos como abastecimentos e despesas diárias se faz cada vez mais necessária. Entretanto, a população brasileira não possui o costume de gerir renda e dívidas, prejudicando sua saúde financeira e, conseqüentemente, a saúde dos automóveis. A população brasileira está entre as que mais utilizam celular diariamente, e, para muitas famílias, a única forma de acesso à internet é através do aparelho. Portanto, o objetivo deste projeto é desenvolver um software para celulares que facilite a gestão de gastos e manutenções automotivas, exibindo métricas, estatísticas, lembretes e históricos de registros. As tecnologias usadas na construção dessa ferramenta foram a linguagem de programação orientada a objetos Java, Spring Boot como *framework* do *backend*, PostgreSQL como Banco de Dados e Android na aplicação móvel. O software foi nomeado como Autocasher.

Palavras-chave: Gestão. Automóvel. Estatística. Abastecimento. Manutenção. Prevenção.

ABSTRACT

Currently in Brazil, about 39 million cars circulate with an average age of 10 years. To ensure safety during the circulation of these cars, preventive maintenance is recommended every 10,000 km or 1 year, preferably whichever comes first. The management of maintenance and other automotive expenses such as refuels and daily expenses is increasingly necessary. However, the Brazilian population does not have the habit of managing income and debts, damaging their financial health and, consequently, the health of their automobiles. The Brazilian population is among those who use cell phones most every day, and for many families, the only way to access the internet is through the device. Therefore, the objective of this project is to develop a software for cell phones that facilitates the management of expenses and automotive maintenance, displaying metrics, statistics, reminders and record histories. The technologies used in the construction of this tool were the object-oriented programming language Java, Spring Boot as the backend framework, PostgreSQL as Database and Android in the mobile application. The software was named Autocasher.

Key-words: Management. Car. Statistics. Refuel. Maintenance. Prevention.

LISTA DE FIGURAS

FIGURA 1 - HABITANTES POR VEÍCULO NO BRASIL.....	16
FIGURA 2 - HABITANTES ECONOMICAMENTE ATIVOS POR VEÍCULO NO BRASIL	17
FIGURA 3 - PIRÂMIDE ETÁRIA (BRASIL 2012 E 2018)	18
FIGURA 4 - LINGUAGENS DE PROGRAMAÇÃO MAIS POPULARES EM 2020	19
FIGURA 5 - FUNCIONAMENTO DA JAVA VIRTUAL MACHINE.....	20
FIGURA 6 - DESENVOLVEDORES QUE TÊM INTERESSE EM CONTINUAR UTILIZANDO.....	22
FIGURA 7 - BANCOS DE DADOS MAIS UTILIZADOS ENTRE DESENVOLVEDORES	23
FIGURA 8 - EXEMPLOS DE ÍCONES USADOS NO MATERIAL DESIGN	25
FIGURA 9 - UTILIZAÇÃO DO TRELLO PARA GESTÃO DO PROJETO	31
FIGURA 10 - UTILIZAÇÃO DO TRELLO PARA GESTÃO DO PROJETO	32
FIGURA 11 - GRÁFICO DE GANTT	35
FIGURA 12 - ARQUITETURA DO AUTOCASHER.....	39
FIGURA 13 - TELA DE INÍCIO.....	39
FIGURA 14 - TELA DE ESTATÍSTICAS	40
FIGURA 15 - TELA DE ABASTECIMENTOS.....	41
FIGURA 16 – TELA DE EDITAR ABASTECIMENTO	42
FIGURA 17 - TELA DE CALCULAR MELHOR COMBUSTÍVEL.....	42
FIGURA 18 - TELA DE MANUTENÇÕES	43
FIGURA 19 - TELA EDITAR MANUTENÇÃO	44
FIGURA 20 - TELA DE VISUALIZAR LEMBRETES	45
FIGURA 21 - TELA DE EDITAR LEMBRETE	45
FIGURA 22 - E-MAIL DE LEMBRETE RECEBIDO.....	46
FIGURA 23 - TELA DE VISUALIZAR GASTOS	47
FIGURA 24 - TELA DE EDITAR GASTO	47
FIGURA 25 - DIAGRAMAS DE CASOS DE USO	55
FIGURA 26 - VISUALIZAR ABASTECIMENTOS	56
FIGURA 27 - MANTER ABASTECIMENTO.....	58
FIGURA 28 - CALCULAR MELHOR COMBUSTÍVEL.....	59

FIGURA 29 - VISUALIZAR ESTATÍSTICAS	60
FIGURA 30 - VISUALIZAR GASTOS.....	62
FIGURA 31 - MANTER GASTO.....	64
FIGURA 32 - VISUALIZAR LEMBRETES	65
FIGURA 33 - MANTER LEMBRETE	67
FIGURA 34 - VISUALIZAR MANUTENÇÕES	68
FIGURA 35 - MANTER MANUTENÇÃO	70
FIGURA 36 - DIAGRAMA DE CLASSES MODEL	72
FIGURA 37 - DIAGRAMA DE CLASSES ANDROID.....	72
FIGURA 38 - DIAGRAMA DE CLASSES BACK END.....	73
FIGURA 39 - DIAGRAMA DE SEQUÊNCIA MANTER ABASTECIMENTO	74
FIGURA 40 - DIAGRAMA DE SEQUÊNCIA VISUALIZAR ESTATÍSTICAS	75
FIGURA 41 - DIAGRAMA DE SEQUÊNCIA MANTER GASTO.....	75
FIGURA 42 - DIAGRAMA DE SEQUÊNCIA MANTER LEMBRETE	76
FIGURA 43 - DIAGRAMA DE SEQUÊNCIA MANTER MANUTENÇÃO	77
FIGURA 44 - DIAGRAMA DE MODELO LÓGICO DE DADOS.....	78

LISTA DE QUADROS

QUADRO 1 - COMPARATIVO DE FUNCIONALIDADES DOS SOFTWARES SEMELHANTES	29
QUADRO 2 - SPRINTS E ATIVIDADES	34

SUMÁRIO

1	INTRODUÇÃO	11
1.1	PROBLEMA	12
1.2	OBJETIVOS	12
1.2.1	Objetivo geral	13
1.2.2	Objetivos específicos	13
1.3	JUSTIFICATIVA	13
1.4	ESTRUTURA DO DOCUMENTO	14
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	FUNDAMENTAÇÃO TEÓRICA DO NEGÓCIO	15
2.1.1	Manutenção preventiva	15
2.1.2	Perfil do dono de automóvel	16
2.2	FUNDAMENTAÇÃO TEÓRICA DAS TECNOLOGIAS	18
2.2.1	Java	18
2.2.2	Spring Boot	21
2.2.3	PostgreSQL	22
2.2.4	Unified Modeling Language (UML)	23
2.2.5	Material Design	25
2.2.6	Método Ágil	26
2.2.6.1	Scrum	26
2.3	SOFTWARES SEMELHANTES	28
3	MATERIAIS E MÉTODOS	31
3.1	ADAPTAÇÃO DO SCRUM AO PROJETO	31
3.2	MATERIAIS	32
3.2.1	Hardware	32
3.2.2	Software	33
3.3	CRONOGRAMA	33
3.4	DESENVOLVIMENTO DO PROJETO	35
3.4.1	Sprint 0	35
3.4.2	Sprint 1	35
3.4.3	Sprint 2	36
3.4.4	Sprint 3	36

3.4.5	Sprint 4	36
3.4.6	Sprint 5	36
3.4.7	Sprint 6	37
3.4.8	Sprint 7	37
4	APRESENTAÇÃO DOS RESULTADOS	38
4.1	ARQUITETURA DO SISTEMA.....	38
4.2	SOFTWARE	39
4.3	CONSIDERAÇÕES DA APRESENTAÇÃO DOS RESULTADOS.....	48
5	CONSIDERAÇÕES FINAIS	49
5.1	TRABALHOS FUTUROS.....	49
5.1.1	Autenticação	49
5.1.2	Funcionamento <i>offline</i>	50
5.1.3	Melhoria da interface gráfica	50
	REFERÊNCIAS.....	51
	APÊNDICE A – LISTA DE HISTÓRIAS DE USUÁRIO	54
	APÊNDICE B – DIAGRAMAS DE CASOS DE USO	55
	APÊNDICE C – ESPECIFICAÇÕES DE CASOS DE USO	56
	APÊNDICE D – DIAGRAMAS DE CLASSE.....	72
	APÊNDICE E – DIAGRAMAS DE SEQUÊNCIA	74
	APÊNDICE F – MODELO LÓGICO DE DADOS	78

1 INTRODUÇÃO

A frota brasileira de automóveis cresceu, em média, 2,23% nos últimos três anos. A projeção para 2020 é de 38,91 milhões de automóveis circulantes no país. A idade média da frota de automóveis atingiu 9 anos e 6 meses em 2018, e a projeção é que chegue a 10 anos em 2020. (SINDIPEÇAS, 2019).

Nesse contexto, a importância da correta gestão e manutenção desses veículos é destacada, visto que, ao fim da garantia de fábrica dos carros, proprietários perdem a obrigatoriedade de realizar manutenções preventivas. O prazo recomendado para realizar manutenções preventivas é frequentemente encontrado no manual do carro, porém uma recomendação geral indica que seja feita a cada 12 meses ou a cada 10 mil quilômetros rodados – preferencialmente o que ocorrer primeiro (FORD, 2018).

A manutenção preventiva de automóveis possui benefícios como redução de custos no longo prazo, aumento da vida útil do veículo e o aumento da segurança no trânsito. Segundo artigo publicado pela Agência de Notícias do Paraná (2016), uma pesquisa realizada no Brasil em 2014 pelo Instituto Scaringella Trânsito, associa a falta de manutenções preventivas à 27% dos acidentes urbanos e rodoviários no país.

Entre as técnicas para aumentar a vida útil do carro estão, por exemplo, a atenção aos óleos e fluídos, cuidado na escolha do combustível, uso de peças e serviços genuínos e atenção às manutenções. (TRANSPORTE ECONÔMICO, 2018).

Apesar dos benefícios que um automóvel pode trazer ao seu proprietário, ele possui diversos gastos associados. Para realizar manutenções preventivas, abastecer com combustíveis de boa qualidade, pagar os impostos e dívidas em dia, é importante que exista um planejamento financeiro adequado para que isso não impacte negativamente o orçamento familiar. O orçamento familiar é uma ferramenta para gerir receitas, gastos e investimentos, e é estruturado a fim de atingir objetivos. (PEREIRA, 2011).

Entretanto, as famílias brasileiras não têm o hábito de realizar um orçamento familiar efetivo, mostrando dificuldade em gerir suas rendas e dívidas,

além de não possuírem o costume de investir em seu futuro e saúde financeira. (SOUZA E TORRALVO, 2004).

Pensando em estimular a gestão de gastos e manutenções, propõe-se um sistema para gestão de gastos automotivos, auxiliando o usuário a manter em dia as manutenções preventivas de seu veículo, ter controle sobre as peças, serviços e abastecimentos realizados.

1.1 PROBLEMA

A concepção desse trabalho se deu devido aos seguintes problemas recorrentes a motoristas de automóveis: dificuldade em lembrar quando foi realizada a última manutenção em seu veículo, dúvida ao escolher o melhor combustível para veículo *flex*, incerteza da data de vencimento e valor de sua apólice de seguro e necessidade de lembrar quando deve realizar a próxima manutenção preventiva.

O ritmo acelerado na rotina dos proprietários de automóveis aliado à ausência de um sistema facilitado para registrar suas manutenções e despesas contribuem para a falta de controle da manutenção de seu automóvel, gerando, portanto, gastos inesperados com manutenções emergenciais. Dessa forma, a falta de controle de manutenções pode afetar a gestão de dívidas e do orçamento familiar.

A utilização de um aplicativo móvel, com armazenamento em nuvem, para a gestão de manutenções, dívidas e gastos com automóveis promove ao motorista o controle sobre o registro consistente destes artefatos, contribuindo para o sucesso do orçamento familiar, além do aumento da vida útil do carro e da segurança no trânsito.

1.2 OBJETIVOS

Tendo em vista o contexto exposto acima, a seguir serão apresentados os objetivos gerais e específicos deste trabalho.

1.2.1 Objetivo geral

O objetivo geral do trabalho é desenvolver um software que permita a gestão de gastos automotivos, controlar manutenções realizadas e programadas e registrar abastecimentos de combustível.

1.2.2 Objetivos específicos

Os objetivos específicos do trabalho são:

- a) Permitir ao motorista o registro do histórico de abastecimentos;
- b) Permitir ao motorista calcular o combustível mais vantajoso com base nos preços;
- c) Permitir ao motorista o registro de gastos;
- d) Permitir ao motorista o registro de manutenções;
- e) Permitir ao motorista o registro de lembretes;
- f) Permitir ao motorista visualizar métricas e estatísticas de seus registros.

1.3 JUSTIFICATIVA

A forma de acesso e gestão de informação mais comum é através de celulares. Em 2019 existiam 420 milhões de dispositivos digitais no Brasil, incluindo *smartphones*, computadores, *notebooks* e *tablets*. Destes dispositivos, 230 milhões são *smartphones*, somando mais de um aparelho por habitante (ÉPOCA, 2019). Em 2017, 49% das famílias brasileiras dependiam do aparelho para acessar a rede, e, dos lares pesquisados, 19% utilizavam exclusivamente o celular. (AGÊNCIA BRASIL, 2018).

Em 2018 o Brasil ocupava o 5º lugar no ranking mundial de tempo de uso diário no aparelho celular. No acumulado diário, os brasileiros passam mais de três horas por dia utilizando o dispositivo, sendo 50% desse período gasto em redes sociais, seguido por aplicativos de reprodução de vídeo com 15% e jogos, acumulando 10% (APP ANNIE, 2019).

Ainda segundo a pesquisa realizada pela consultoria App Annie, o brasileiro possui, em média, 70 aplicativos instalados, sendo 30 destes utilizados pelo usuário constantemente.

Os brasileiros acessam constantemente seus aparelhos, 24% dos usuários dizem acessar praticamente o tempo todo, e 42% várias vezes em uma hora. Metade dos usuários diz checar seu celular ao acordar, e 63% antes de ir para a cama (PAGBRASIL, 2019).

Levando em consideração o perfil do dono de automóvel no Brasil, juntamente com as estatísticas de aplicativos móveis no país, esse trabalho visa desenvolver uma aplicação voltada à adultos economicamente ativos com mais de 30 anos e que possuem carro e celular com acesso à internet.

1.4 ESTRUTURA DO DOCUMENTO

Este documento apresenta as informações do desenvolvimento do projeto e está estruturado conforme descrito abaixo.

O Capítulo 2 tem como objetivo apresentar a contextualização teórica do sistema desenvolvido, detalhando os assuntos importantes para sua confecção. Entre os temas abordados estão uso de celular no Brasil, uso de aplicativos, manutenção automotiva e orçamento familiar. Serão comentadas também as tecnologias e *frameworks* usados.

O Capítulo 3 objetiva relatar os materiais e métodos, suas definições técnicas e características individuais.

O Capítulo 4 apresenta o software desenvolvido, suas descrições e detalhamento das funcionalidades. Além das descrições, serão apresentadas as imagens do produto finalizado em funcionamento, permitindo melhor compreensão das descrições escritas.

O Capítulo 5 tem por objetivo apresentar as considerações finais do projeto. São levados em consideração os objetivos propostos e observa-se se foram cumpridos satisfatoriamente, se houve pontos de falha e quais as possibilidades de trabalhos a serem realizados em momentos futuros.

Ao final do documento pode-se observar os artefatos gerados como definição do sistema.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão apresentados os assuntos, tecnologias e *frameworks* relevantes ao projeto devidamente referenciados à bibliografia fundamental para o desenvolvimento.

A fim de facilitar a separação dos temas, a fundamentação foi separada em duas categorias principais de Negócio e Tecnologia.

2.1 FUNDAMENTAÇÃO TEÓRICA DO NEGÓCIO

A seguir são abordados os temas relevantes do Negócio proposto, tratando de estatísticas, dados e estudos sobre o cenário brasileiro de manutenção automotiva, uso de celular e de aplicativos.

2.1.1 Manutenção preventiva

Para Moubray (1997), manutenção é garantir que ativos físicos continuem funcionando conforme a expectativa de seu usuário. Segundo o autor, falhas por falta de manutenção impactam cada vez mais a segurança e o meio ambiente, e devido à crescente dependência da sociedade em ativos físicos, cresce proporcionalmente o custo para ter e operar tais equipamentos. O pesquisador defende que para assegurar máximo retorno no investimento representado pela compra de um equipamento, este deve funcionar eficientemente pelo tempo que for desejado.

Barbosa (2017) explica que algumas das estratégias de manutenção são:

- Manutenção corretiva por opção;
- Manutenção preventiva sistemática;
- Manutenção baseada em inspeção visual e ou sensitiva;
- Manutenção baseada em inspeção instrumentada primeira linha;
- Manutenção baseada na inspeção por intrusão;
- Manutenção baseada em inspeção preditiva;
- Manutenção baseada em análise proativa de falha (RCM);
- Manutenção autônoma.

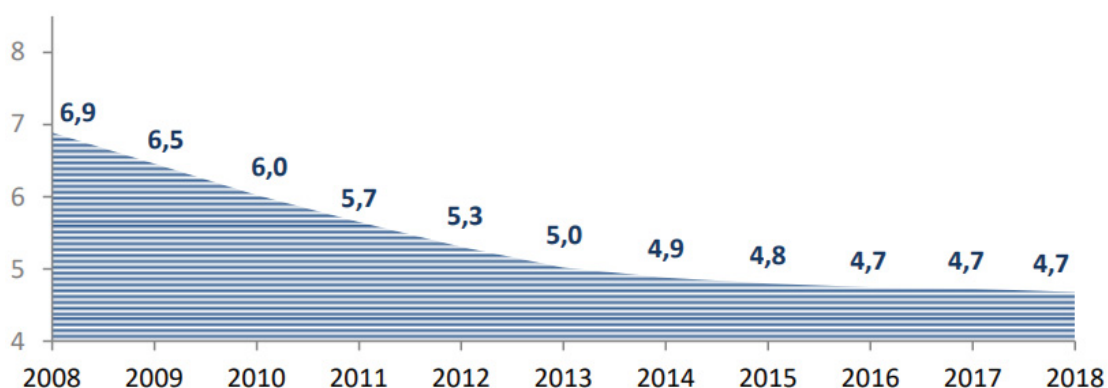
Segundo artigo publicado pela Asea Brown Boveri (ABB) em 2020, manutenção preventiva é uma ação realizada em intervalos pré-determinados ou seguindo critérios pré-determinados, visando reduzir o risco de falha ou perda de performance do equipamento. Os ciclos de manutenção são planejados de acordo com a necessidade da retirada de serviço do equipamento.

Uma das vantagens da manutenção preventiva é assegurar o funcionamento contínuo do equipamento com paradas programadas e previsíveis para consertos. A aplicação desse tipo de manutenção na rotina do dono de automóveis permite-o planejar o momento mais oportuno para realizar o serviço em seu carro (WYREBSKI, 1997).

2.1.2 Perfil do dono de automóvel

De acordo com uma pesquisa realizada pelo Sindicato Nacional da Indústria de Componentes para Veículos Automotores (SINDIPEÇAS) em conjunto com a Associação Brasileira da Indústria de Autopeças (ABIPEÇAS) em 2019, em 2018 no Brasil existia uma proporção de 4,7 habitantes por veículo circulante. Essa relação, juntamente com o histórico anual, pode ser visualizada na FIGURA 1.

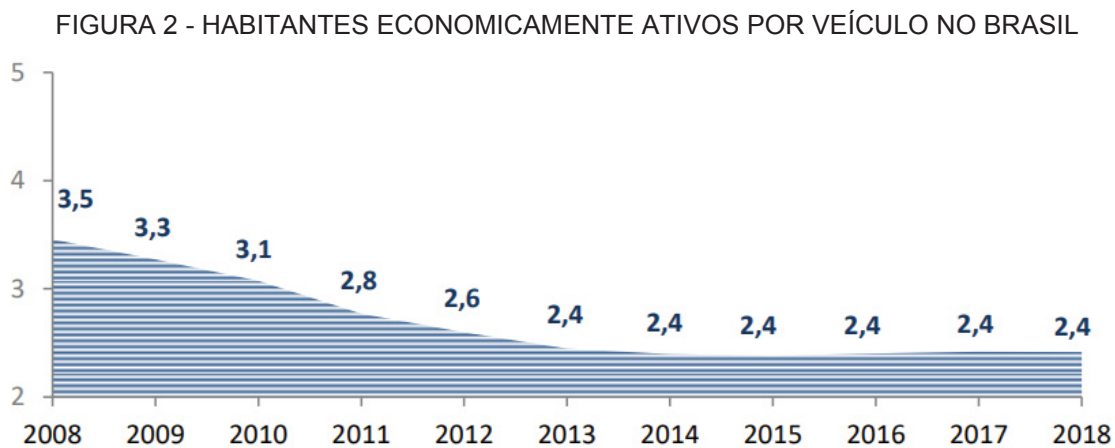
FIGURA 1 - HABITANTES POR VEÍCULO NO BRASIL



FONTE: SINDIPEÇAS (2019)

Entretanto, ao considerar a População Economicamente Ativa (PEA), que soma pessoas acima de 16 anos, ocupadas ou em busca de empregos, a

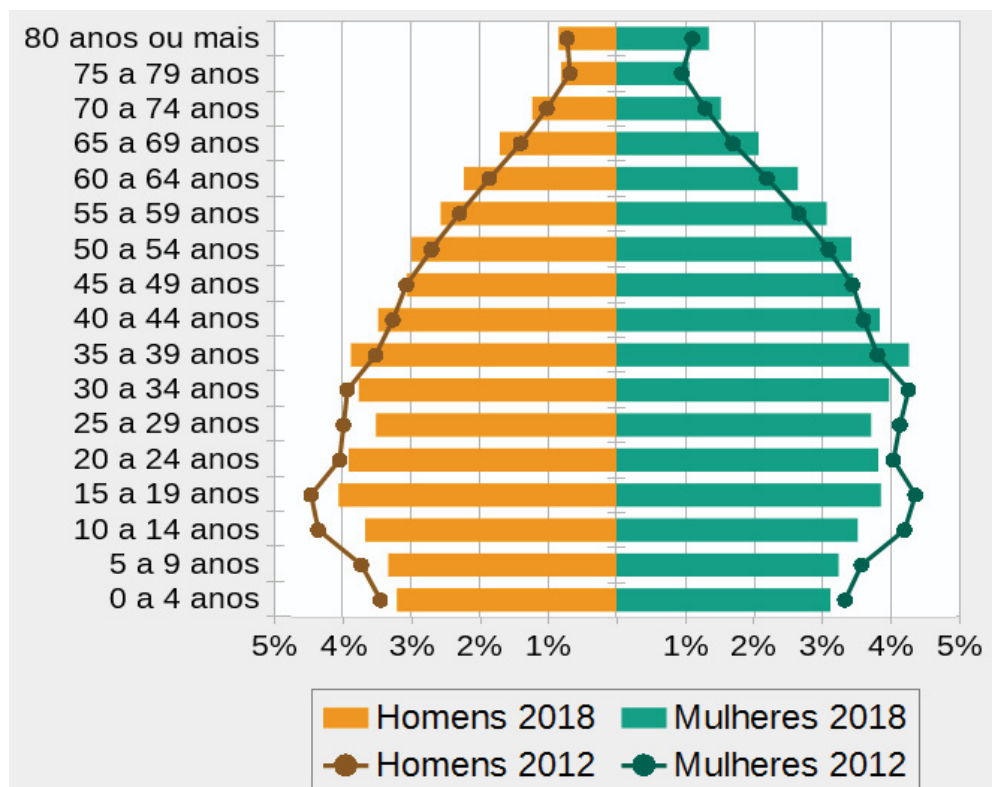
proporção indica expressivo declínio, descendo de 4,7 para 2,4 habitantes por veículo circulante no país. Esse fenômeno pode ser observado na FIGURA 2.



FONTE: SINDIPEÇAS (2019)

Segundo levantamento realizado pelo Instituto Brasileiro de Geografia e Estatística (IBGE) em 2019, a população economicamente ativa no Brasil era de 105,2 milhões, e, segundo outro estudo apresentado pelo IBGE em 2018, a população acima de 30 anos de idade no Brasil era de 57,1%. A pirâmide etária do Brasil em 2018 pode ser observada na FIGURA 3.

FIGURA 3 - PIRÂMIDE ETÁRIA (BRASIL 2012 E 2018)



FONTE: IBGE (2018)

Pode-se observar na pirâmide etária do Brasil que a população está ficando mais velha, e, portanto, sua população economicamente ativa também está ficando mais velha. Dessa forma, o desenvolvimento desse projeto define como foco a faixa específica da população economicamente ativa e mais velha entre 30 e 64 anos.

2.2 FUNDAMENTAÇÃO TEÓRICA DAS TECNOLOGIAS

A seguir serão abordadas as tecnologias relevantes para o desenvolvimento deste trabalho, tratando dos aspectos técnicos e conceituais das ferramentas utilizadas.

2.2.1 Java

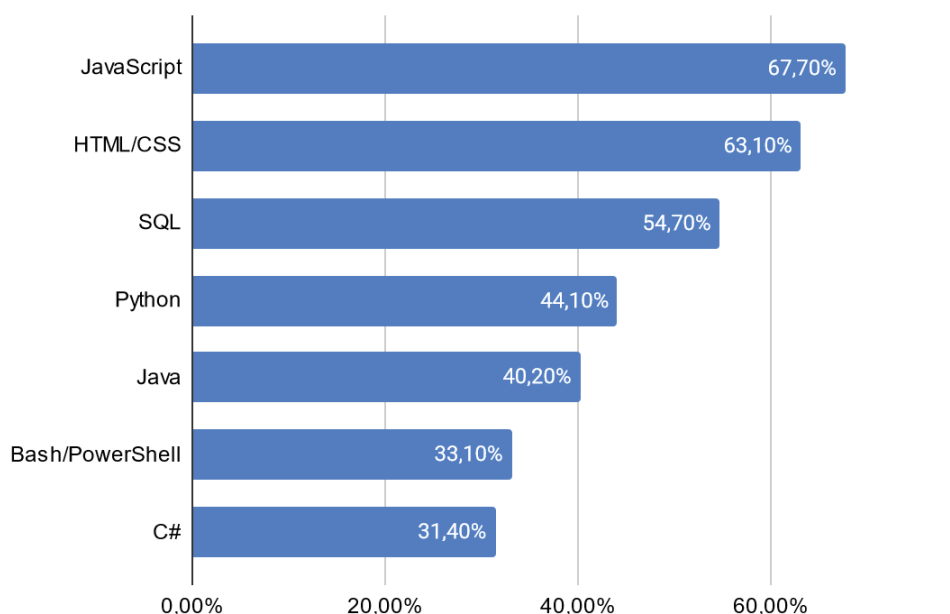
Java é uma linguagem de programação orientada a objetos. Foi desenvolvida pela Sun Microsystems em 1995. A *Java Runtime Environment*

(JRE) agrega a Máquina Virtual Java (JVM), plataformas e classes base, suportando diversas bibliotecas Java (ORACLE, 2021). O *Java Development Kit* inclui o JRE e é a ferramenta usada para compilar, documentar e gerar os pacotes de programas Java.

A linguagem Java é considerada de alto nível, significando que ela omite os detalhes da representação de máquina. A tecnologia possui um sistema automático de gestão de armazenamento, tipicamente usando um *garbage collector*, evitando assim problemas de segurança causados por falhas de alocação de memória, como acontece com os comandos *free* e *delete*, das linguagens C e C++, respectivamente (GOSLING et al., 2015).

Segundo levantamento realizado pelo site Stack Overflow em 2020, Java é a 5º linguagem mais utilizada pelos desenvolvedores, significando que 40,2% dos profissionais utilizam Java em suas atividades. A FIGURA 4 mostra o ranking das linguagens mais populares segundo o Stack Overflow em 2020.

FIGURA 4 - LINGUAGENS DE PROGRAMAÇÃO MAIS POPULARES EM 2020

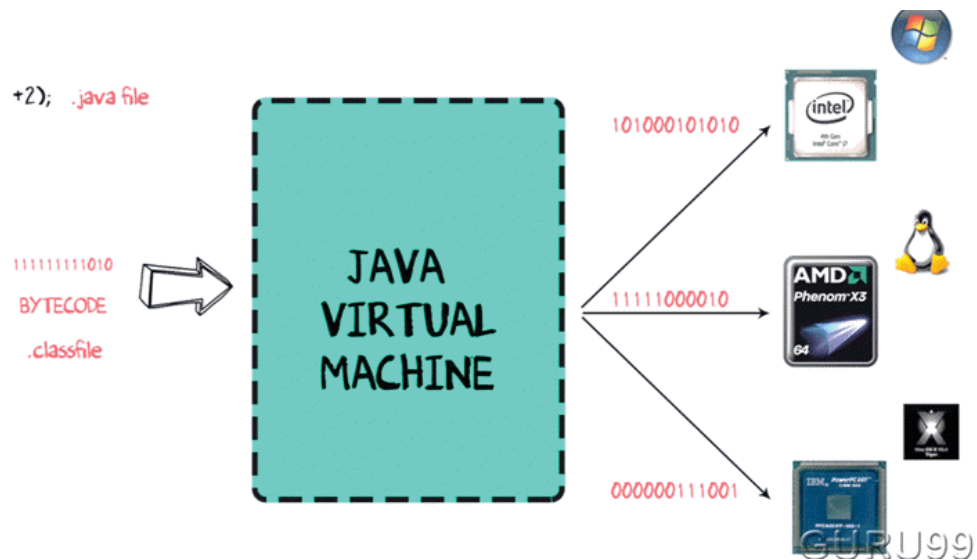


FONTE: Adaptado de Stack Overflow (2020)

A linguagem é similar a C++ com recursos simplificados e avançados. A tecnologia é concorrente, significando que pode executar diversos comandos ao invés de somente sequencial. É baseada em classes e orientada a objetos. O

compilador Java gera um formato de código chamado *bytecode* que é interpretado pela JVM em qualquer sistema operacional, permitindo a linguagem a característica de independente, podendo ser executada em todas as plataformas que suportam Java (GURU99, 2021). A FIGURA 5 mostra o funcionamento da JVM.

FIGURA 5 - FUNCIONAMENTO DA JAVA VIRTUAL MACHINE



FONTE: Guru99 (2021)

Java é usado como funcionalidade central em diversas tecnologias presentes no mercado, possibilitando o desenvolvimento de aplicações nos domínios de Desenvolvimento *Web*, Testes Contínuos, Android, *Big Data* e *frameworks* múltiplos, como Spring e Hibernate (EDUREKA, 2020). Abaixo são listados exemplos de tecnologias que utilizam Java, juntamente com seu propósito geral.

- Spring: *Framework web*
- Hibernate: *Framework* de mapeamento objeto/relacional
- Selenium: *Framework* de testes
- Hadoop: Plataforma de *Big Data*
- Android: Sistema Operacional *mobile*

2.2.2 Spring Boot

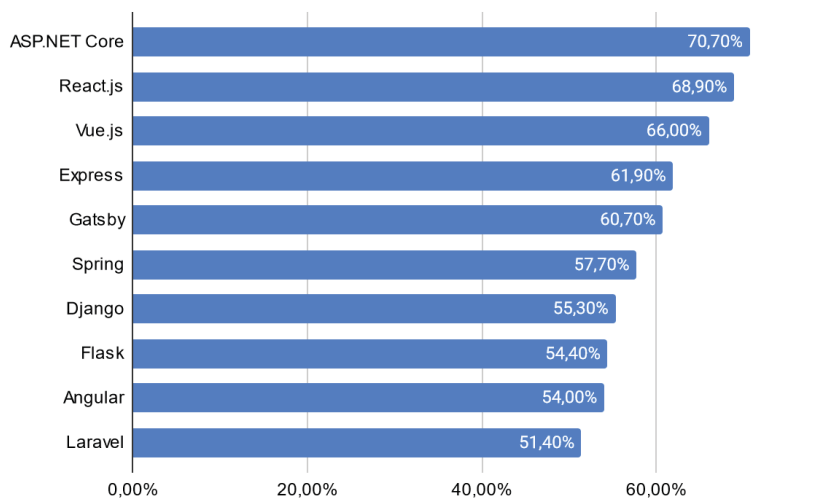
Spring é um *framework open-source* Java que fornece um modelo de configuração e programação facilitado para aplicações *web* na *Java Enterprise Edition* em diversas plataformas (SPRING, 2020).

O Spring possui tecnologias como injeção de dependência, eventos, validação e *data binding*. Possibilita testes através do uso de *mock objects* e *frameworks* como TestContext e Spring MVC Test. A ferramenta entrega a funcionalidade de acesso de dados, utilizando transações, suporte a Data Access Object, e Java Database Connectivity. Além disso, integra *frameworks web* como Spring MVC e Spring WebFlux, podendo ser utilizado em diversas linguagens como Java, Kotlin e Groovy (SPRING, 2021).

Spring Boot é um *framework* criado para facilitar o processo de iniciar um novo projeto Spring, simplificando a configuração inicial, gestão de dependências e evitando desperdício de tempo com códigos desnecessários e repetitivos (INFOQ, 2014). O *framework* configura automaticamente o projeto Spring com opções padronizadas personalizáveis. Por exemplo, se foi escolhido um banco de dados MySQL mas a conexão ainda não foi criada, o Spring Boot automaticamente configura um banco de dados em memória.

De acordo com pesquisa realizado pelo Stack Overflow em 2020, Spring é um dos *frameworks* para *web* mais populares entre os desenvolvedores que usaram a tecnologia. A FIGURA 6 ilustra o ranking dos *frameworks* para *web* mais estimados segundo o Stack Overflow em 2020.

FIGURA 6 - DESENVOLVEDORES QUE TÊM INTERESSE EM CONTINUAR UTILIZANDO



FONTE: Adaptado de Stack Overflow (2020)

2.2.3 PostgreSQL

PostgreSQL é um sistema de gerenciamento de banco de dados (SGBD) livre e de código aberto que estende a linguagem SQL combinada com diversas características para armazenar de forma segura e escalável dados complexos. A tecnologia foi criada em 1986, como parte do projeto POSTGRES na Universidade da Califórnia, em Berkeley, nos Estados Unidos (POSTGRESQL, 2021).

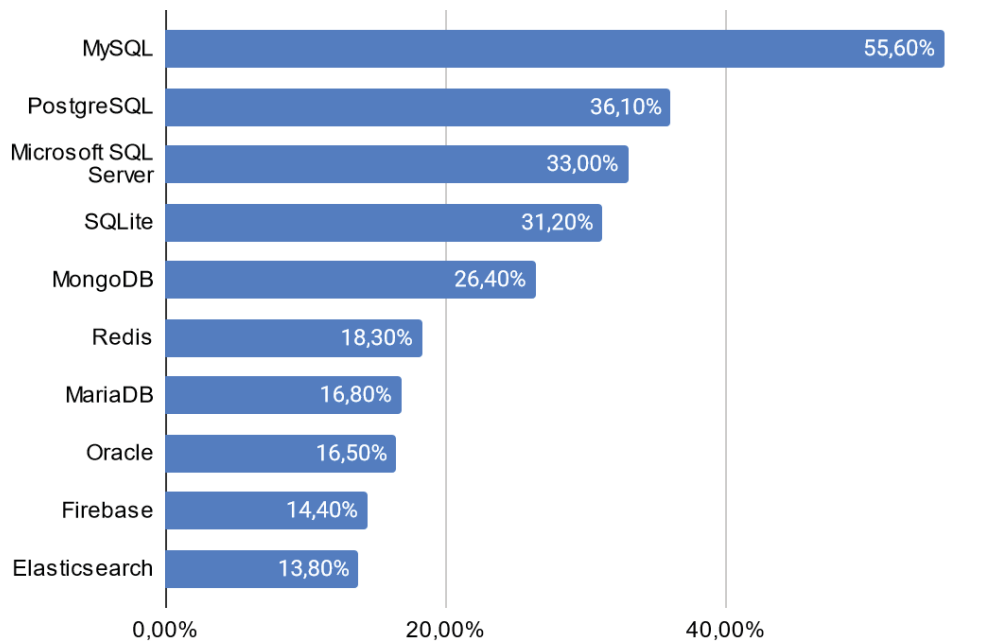
De acordo com a documentação oficial da ferramenta, algumas das principais características do PostgreSQL são:

- Tipos de dados;
- Integridade de dados;
- Concorrência e Performance;
- Confiabilidade e Recuperação de Desastres;
- Segurança;
- Extensibilidade;

Segundo levantamento realizado pelo Stack Overflow em 2020, PostgreSQL é o segundo banco de dados mais usados entre os

desenvolvedores, atrás somente do MySQL. De acordo com a pesquisa, o PostgreSQL é utilizado por 36,1% das pessoas que responderam à pesquisa. A FIGURA 7 mostra a lista dos bancos de dados mais utilizados em 2020 entre os desenvolvedores:

FIGURA 7 - BANCOS DE DADOS MAIS UTILIZADOS ENTRE DESENVOLVEDORES



FONTE: Adaptado de Stack Overflow (2020)

2.2.4 Unified Modeling Language (UML)

Segundo Booch et al. (1999), modelagem é uma técnica de engenharia comprovada e aceita pela indústria. Segundo os autores, um modelo é uma simplificação da realidade. Um bom modelo inclui elementos que possibilitam visões simplificadas e detalhadas dos planos de um projeto, permitindo alcançar diversos objetivos como:

- Modelos auxiliam na visualização de um sistema como ele é ou como deseja-se que ele seja;
- Modelos permitem a especificação da estrutura ou comportamento de um sistema;

- Modelos fornecem um padrão que guia a construção de um sistema;
- Modelos documentam as melhores decisões tomadas.

De acordo com os autores, existem diversas maneiras de realizar a modelagem de *softwares*. As mais comuns são com perspectiva de algoritmo e com perspectiva de orientação a objetos. A mais tradicional, a visão de algoritmo, leva desenvolvedores a focar em problemas de controle e na decomposição de grandes algoritmos em pedaços menores. Conforme os requisitos mudam e o sistema cresce, essa forma de modelagem gera manutenções mais difíceis. Em contrapartida, a forma que foca em orientação a objeto possui como principal componente os objetos ou classes, onde cada um possui uma identidade, estado e comportamento. Essa é a principal forma pois provou ao longo do tempo ser de grande valor na construção dos mais variados tipos de sistemas.

Para Guedes (2011), a *Unified Modeling Language* (UML) é uma linguagem visual usada ao modelar softwares baseados no paradigma de orientação a objetos. A UML permite entendimento facilitado de sistemas, componentes e funcionalidades através de representações visuais e intuitivas. Essa linguagem define artefatos visuais que auxiliam na documentação e modelagem de softwares, sendo divididos em três categorias:

- **Diagramas estruturais:** Diagrama de Classes, Diagrama de Objetos, Diagrama de Componentes, Diagrama de Pacotes, Diagrama de Estrutura Composta, Diagrama de Instalação e Diagrama de Perfil;
- **Diagramas comportamentais:** Diagrama de Casos de Uso, Diagrama de Atividades e Diagrama de Transição de Estados;
- **Diagramas de Interação:** Diagrama de Sequência, Diagrama de Interatividade, Diagrama de Colaboração ou Comunicação e Diagrama de Tempo.

Utilizando estes artefatos, desenvolvedores podem representar diversos aspectos de um software independente de linguagens, *frameworks* ou metodologias.

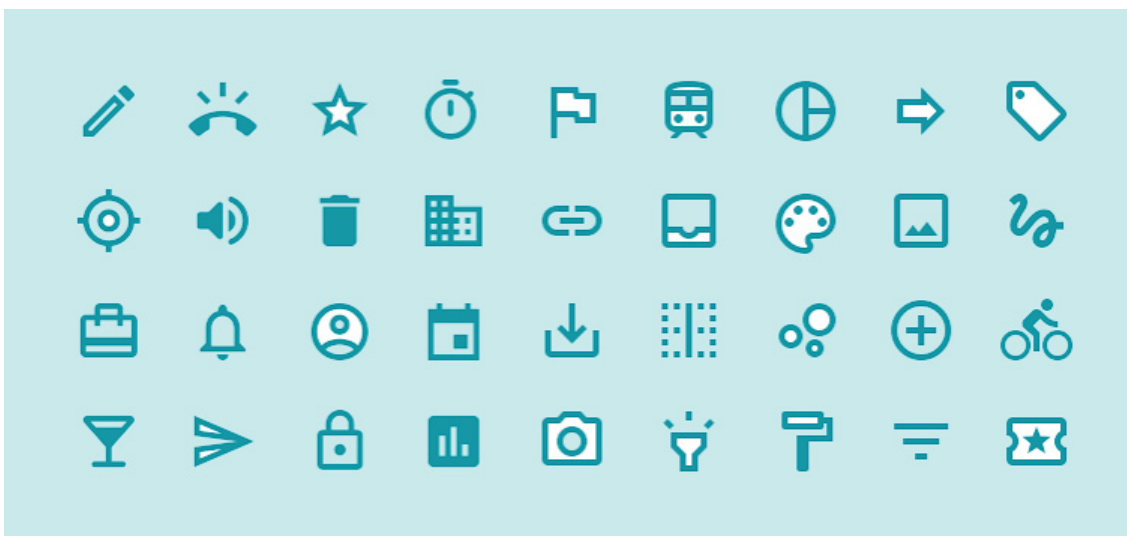
De acordo com Booch et al. (1999), a UML é usada primariamente em sistemas de software, se mostrando efetiva em contextos como sistemas

empresariais, serviços financeiros e bancários, telecomunicações, transporte, entre outros. Entretanto, a UML não se limita à modelagem de software. Ela possui capacidade de representar conceitos como fluxos de trabalho no ambiente jurídico, estrutura e comportamento de sistemas de saúde e até no design de hardware.

2.2.5 Material Design

Material Design é um conceito de *design* desenvolvido pelo Google e apresentado em 2014. É um sistema de princípios e ferramentas que suportam boas práticas no *design* de interface com usuário (GOOGLE, 2020). O estilo visual e de experiência de usuário está presente na maioria dos produtos da empresa como o Google Documentos, Planilhas, Youtube, Gmail e Maps. A FIGURA 8 ilustra alguns dos ícones utilizados no Material Design.

FIGURA 8 - EXEMPLOS DE ÍCONES USADOS NO MATERIAL DESIGN



FONTE: Google (2020)

Segundo o Google (2020), o Material Design objetiva criar uma linguagem visual que une elementos clássicos do bom *design* com a inovação e tecnologia, unificar a experiência do usuário entre as várias plataformas e aparelhos e customizar a linguagem permitindo expandir e flexibilizar os fundamentos da inovação e expressão de uma marca.

Os princípios de design do Material Design abordam diversos temas e oferecem exemplos para Ambiente, *Layout*, Navegação, Cor, Tipografia, Som, Iconografia, Forma, Movimento, Interação e Comunicação.

2.2.6 Método Ágil

Em 2001, a partir da reunião de um grupo de profissionais e representantes de metodologias como Extreme Programming, Scrum e DSDM, foi publicado o Manifesto Ágil. Este manifesto lista doze princípios para projetos que aplicam a metodologia ágil e objetiva torná-los uma alternativa a desenvolvimentos de software orientados a documentação pois propõe quebrar o tradicional longo ciclo de desenvolvimento, tipicamente chamado de método cascata, em ciclos menores chamados de iterações. Segundo Beck et al., os 4 valores da metodologia são:

Estamos descobrindo maneiras melhores de desenvolver software, fazendo-o nós mesmos e ajudando outros a fazerem o mesmo. Através deste trabalho, passamos a valorizar:

- Indivíduos e interações mais que processos e ferramentas
- Software em funcionamento mais que documentação abrangente
- Colaboração com o cliente mais que negociação de contratos
- Responder a mudanças mais que seguir um plano

Ou seja, mesmo havendo valor nos itens à direita, valorizamos mais os itens à esquerda (BECK et al., 2001, p. 1).

Segundo Fowler (2001), os valores acima foram definidos de forma que a primeira parte de cada valor indica a preferência, e a segunda parte indica um aspecto que, apesar de importante, possui menos prioridade. Para o autor, essa diferença é o ponto central da agilidade pois auxilia projetos de software a definirem suas prioridades a respeito de processos, ferramentas, interações e indivíduos.

2.2.6.1 Scrum

Scrum é um *framework* para ajudar times e organizações a gerar valor com soluções adaptáveis para problemas complexos (SCHWABER, 2020). De

forma resumida, a metodologia necessita de um *Scrum Master* para fomentar um ambiente que:

1. Um *Product Owner* insere o trabalho de um problema complexo no *Product Backlog*.
2. O *Scrum Team* transforma uma seleção de tarefas em um incremento de valor ao produto em uma *Sprint*.
3. O *Scrum Team* e seus *stakeholders* analisam os resultados e se ajustam para o próximo *Sprint*.
4. Repete

Para Schwaber (2020), Scrum é propositalmente incompleto, somente definindo as partes necessárias para implementá-lo. O *framework* não possui instruções detalhadas, somente propõe regras para guiar relações e interações. Os valores do método são: compromisso, foco, abertura, respeito e coragem. Segundo o autor, o *Scrum Team* é dividido em:

- **Desenvolvedores:** Responsáveis por criar incrementos em cada *Sprint*;
- **Product Owner:** Responsável por maximizar o valor do produto resultante do trabalho do *Scrum Team*;
- **Scrum Master:** Encarregado de verificar que o Scrum está sendo corretamente utilizado, facilitando entendimento da teoria e prática.

Os *Scrum Events* são definidos pelo autor como uma oportunidade para revisar e adaptar o andamento do projeto, criando transparência para os envolvidos. Abaixo são listados os *Scrum Events*:

- **Sprint:** Período de duração fixa de 1 mês ou menos onde ideias são transformadas em valor;
- **Sprint Planning:** Evento que inicia a *Sprint* e define o trabalho a ser executado durante o período. Esse plano é criado de forma colaborativa por todo o *Scrum Team*;
- **Daily Scrum:** Evento diário de 15 minutos para os desenvolvedores do time discutirem o progresso e adaptar o trabalho conforme necessário;

- ***Sprint Review***: Evento que ocorre ao final da *Sprint* e permite ao *Scrum Team* discutir o que foi realizado e o que mudou, possibilitando o planejamento para o próximo *Sprint*;
- ***Sprint Retrospective***: Evento que ocorre ao final da *Sprint* e objetiva planejar formas de aumentar qualidade e efetividade ao revisar indivíduos, interações, processos e ferramentas.

2.3 SOFTWARES SEMELHANTES

Durante a idealização e definição do software apresentado realizou-se um estudo dos produtos mais populares disponíveis no mercado. Foram observados os objetivos de cada aplicação, suas principais funcionalidades, forma de implementação e experiência de usuário. Pode ser observado que o sistema proposto possui diversas características semelhantes aos outros softwares estudados. Abaixo serão introduzidos os aplicativos estudados, e, em seguida, é apresentada o QUADRO 1, comparando as principais funcionalidades dos produtos em questão.

Drivvo é um aplicativo disponível gratuitamente na Play Store, loja de aplicativos para Android, e objetiva auxiliar o usuário a cuidar da preservação e gestão de custos automotivos (DRIVVO, 2021). Possui um histórico dos registros armazenados, filtro de registros, registro de manutenções, percurso, despesa, abastecimentos, veículos, além de diversas funcionalidades que são bloqueadas somente para usuários registrados, e outras somente para usuários da Pro, sua versão paga. Devido a sua diversa gama de funcionalidade e opções, é possível que um usuário menos experiente se sinta desconfortável ao utilizá-lo.

Fuelio é uma aplicação para smartphones disponível gratuitamente na Play Store e tem por objetivo ser uma ferramenta para gestão de veículo, abastecimentos, custos e quilometragem (FUELIO, 2021). Possui um histórico de registros, registro de veículos, manutenções, despesas, abastecimentos, lembretes, mapa de postos de combustível e calculadora de viagem. Apesar de apresentar uma interface simplificada, algumas de suas principais funcionalidades estão acessíveis somente após diversos cliques, possivelmente tornando-as menos conhecidas pela sua base de usuários.

Simply Auto é um aplicativo disponível gratuitamente na Play Store e é uma ferramenta para gestão de quilometragem e veículos. Possui um histórico de registros, registro de veículos, manutenções, percursos, despesas, abastecimentos, cálculo de preços de viagens (SIMPLY AUTO, 2021). Possui funcionalidades que estão disponíveis somente para usuários registrados e algumas opções somente para usuários da versão paga. Sua tradução para português do Brasil possui diversos problemas, dificultando o entendimento de algumas funções do software.

O aplicativo proposto, Autocasher, possui como diferencial sua experiência de usuário facilitada. Sua interface é baseada em cores que classificam cada tipo de registro, possui uma gama simplificada de funcionalidades que estão sempre visíveis, possibilitando o entendimento facilitado do usuário, sendo dessa forma mais acessível para utilizadores com menos experiência tecnológica. O QUADRO 1 exibe o comparativo de funcionalidades dos produtos acima mencionados.

QUADRO 1 - COMPARATIVO DE FUNCIONALIDADES DOS SOFTWARES SEMELHANTES

Funcionalidades	Autocasher	Drivo	Fuelio	Simply Auto
Visualização de histórico	X	X	X	X
Filtro de histórico	X	X		
Registro de veículos		X	X	X
Funcionalidades para usuário registrado		X		X
Funcionalidades para versão paga		X		X
Registro de Receita		X		
Registro de Manutenção	X	X	X	X
Registro de Percurso			X	X
Registro de Despesa	X	X	X	X
Registro de Abastecimento	X	X	X	X
Lembretes	X	X	X	
Relatórios	X	X	X	X
Calculadora melhor combustível	X	X		
Cálculo preço de viagem			X	X
Localização de postos de combustíveis		X	X	X
Mapa			X	X

FONTE: O Autor (2021)

Ao observar o quadro acima, nota-se que o aplicativo proposto evita a implementação de funcionalidades que já são atendidas em aplicações populares como o Google Maps, dessa forma o Autocasher foca na gestão das principais funções visando a uma melhor experiência de usuário.

O próximo capítulo deste documento apresenta os materiais e métodos utilizados durante a confecção do projeto.

3 MATERIAIS E MÉTODOS

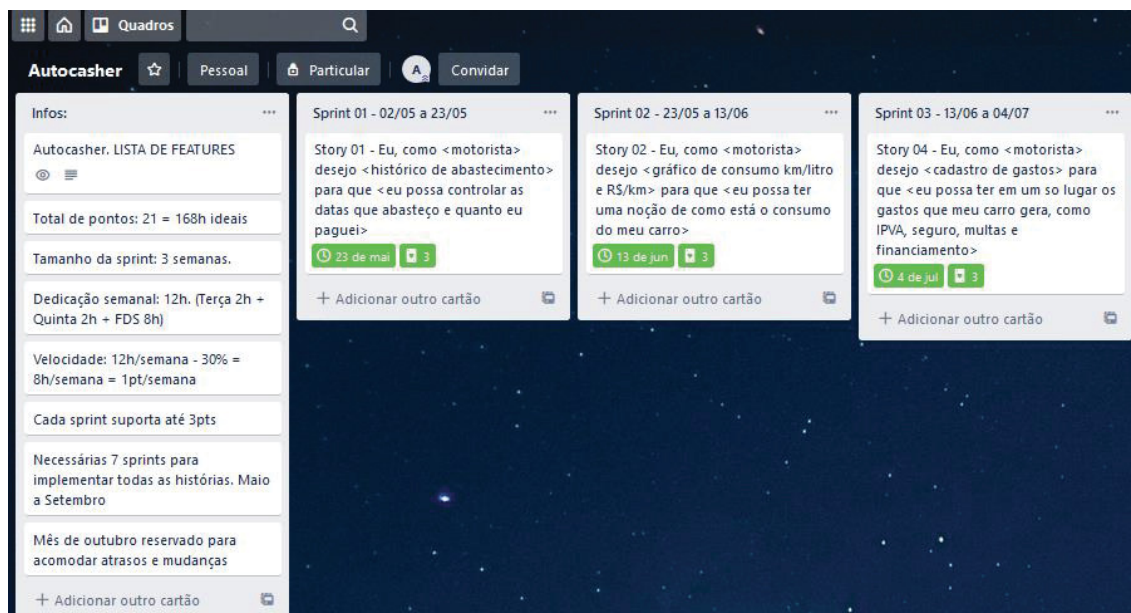
Neste capítulo serão apresentados os materiais e métodos relevantes ao projeto e utilizados durante seu desenvolvimento.

3.1 ADAPTAÇÃO DO SCRUM AO PROJETO

Este projeto utilizou uma adaptação do Scrum em seu desenvolvimento. Foram planejados Sprints de desenvolvimento considerando o tempo disponível do autor e foi fixado um período de 3 semanas por Sprint. Por conta deste projeto não ser realizado em equipe, não foram planejadas cerimônias Scrum. O autor assumiu os papéis de *Product Owner*, *Scrum Master* e time de desenvolvimento.

A gestão das Sprints foi feita utilizando o Trello, uma ferramenta online para gestão de projetos de forma visual (TRELLO, 2020), separando em cartões e listas, semelhante à estrutura de um quadro branco e adesivos *post-its*. A FIGURA 9 exibe como realizou-se a gestão das Sprints utilizando a ferramenta Trello, exibindo algumas informações do planejamento Scrum e as Sprints 1 a 3.

FIGURA 9 - UTILIZAÇÃO DO TRELLO PARA GESTÃO DO PROJETO



FONTE: O Autor (2021).

A FIGURA 10 mostra o planejamento das Sprints 4 a 7 utilizando a ferramenta Trello:

FIGURA 10 - UTILIZAÇÃO DO TRELLO PARA GESTÃO DO PROJETO



FONTE: O Autor (2021).

3.2 MATERIAIS

Neste item são apresentados os hardwares e softwares utilizados durante a construção do sistema proposto.

3.2.1 Hardware

Durante o desenvolvimento deste projeto foram utilizados dois computadores, sendo um desktop e outro notebook, ambos utilizando Windows 10 64bit. Suas especificações podem ser conferidas a seguir:

- Desktop
 - Memória RAM: 16 GB;
 - Processador: AMD Ryzen 5 3400G
 - Placa de vídeo: AMD Radeon Vega Graphics
 - Armazenamento: 256 GB SSD + 1 TB HDD
- Notebook

- Memória RAM: 8GB
- Processador: i3
- Armazenamento: 128GB SSD

3.2.2 Software

Os softwares utilizados durante a confecção de todas as etapas do software podem ser conferidos abaixo:

- **Android Studio v4.1.3:** Desenvolvimento da aplicação Android. Disponível em: <https://developer.android.com/studio>
- **Astah UML v8.3.0:** Desenvolvimento dos diagramas UML. Disponível em: <https://astah.net/downloads/>
- **Microsoft Word – Office 365:** Confecção do documento escrito desse projeto. Disponível em: <https://www.office.com/>
- **Mozilla Firefox v88.0:** Pesquisas e visualização de materiais de suporte para o desenvolvimento. Disponível em: <https://www.mozilla.org/pt-BR/firefox/new/>
- **Netbeans v11.1:** Desenvolvimento da aplicação *backend*. Disponível em: <https://netbeans.apache.org/download/nb111>
- **Sublime Text v3.2.2:** Suporte para o desenvolvimento de software. Disponível em: <https://www.sublimetext.com/3>

3.3 CRONOGRAMA

O planejamento de atividades do projeto foi realizado conforme o planejamento de Sprints. Os artefatos e diagramas UML foram confeccionados anteriormente, no período classificado como *Sprint 0*. A lista de artefatos e diagramas é a seguinte:

- Lista de Histórias de Usuário (Apêndice A);
- Diagrama de Casos de Uso (Apêndice B);
- Especificações dos Casos de Uso, com interfaces (Apêndice C);
- Diagramas de Classe (Apêndice D);
- Diagramas de Sequência (Apêndice E);
- Modelo Lógico de Dados (Apêndice F).

O QUADRO 2 mostra a lista de sprints, datas e atividades desenvolvidas.

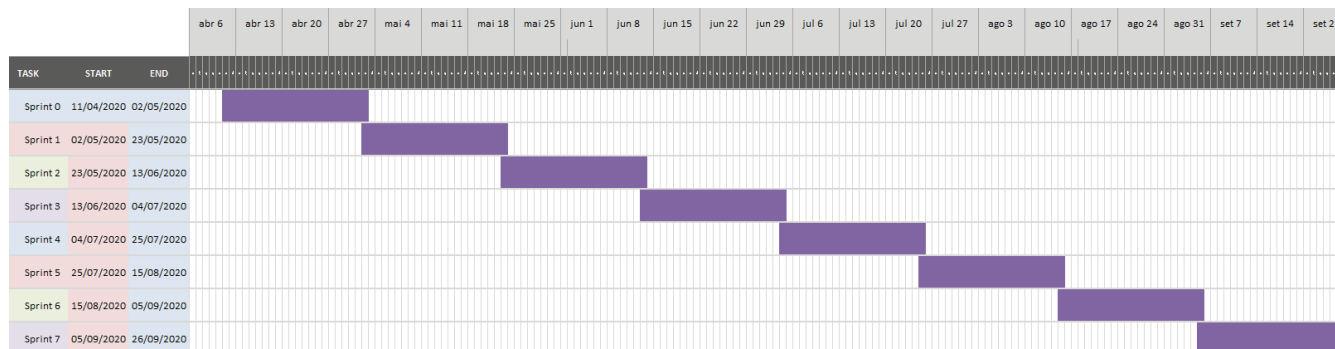
QUADRO 2 - SPRINTS E ATIVIDADES

Sprint	Início	Fim	Atividades
0	11/abr	02/mai	Artefatos: Histórias de usuário, Diagrama de Casos de Uso, Especificação dos Casos de Uso, Interfaces Prototipadas, Diagrama de Classes, Diagramas de Sequência e Modelo Lógico de Dados
1	02/mai	23/mai	Story 01 - Eu, como <motorista> desejo <histórico de abastecimento> para que <eu possa controlar as datas que abasteço e quanto eu paguei>
2	23/mai	13/jun	Story 02 - Eu, como <motorista> desejo <gráfico de consumo km/litro e R\$/km> para que <eu possa ter uma noção de como está o consumo do meu carro>
3	13/jun	04/jul	Story 04 - Eu, como <motorista> desejo <cadastro de gastos> para que <eu possa ter em um só lugar os gastos que meu carro gera, como IPVA, seguro, multas e financiamento>
4	04/jul	25/jul	Story 05 - Eu, como <motorista> desejo <previsão de gastos> para que <eu possa visualizar quanto vou gastar mensalmente e anualmente com despesas do carro>
5	25/jul	15/ago	Story 06 - Eu, como <motorista> desejo <registro de realização de serviços> para que <eu possa controlar quais serviços já realizei, quando, e qual foi o preço. para saber se algum mecânico está me enganando e cobrando muito caro>
6	15/ago	05/set	Story 07 - Eu, como <motorista> desejo <lembretes de manutenções regulares> para que <eu pare de esquecer de realizar as manutenções na hora certa>
7	05/set	26/set	Story 03 - Eu, como <motorista> desejo <visualizar qual combustível é mais vantajoso> para que <eu economize dinheiro com os preços de combustíveis tão instáveis atualmente> Story 08 - Eu, como <motorista> desejo <programar lembretes de manutenção> para que <eu pare de esquecer de manutenções>

FONTE: O Autor (2021)

Na FIGURA 11 observa-se o gráfico de Gantt do andamento desse projeto.

FIGURA 11 - GRÁFICO DE GANTT



FONTE: O Autor (2021)

3.4 DESENVOLVIMENTO DO PROJETO

Baseando-se na estimativa de esforços durante a definição de histórias de usuários, foram planejadas 8 *Sprints* com duração de 3 semanas e esforço de 12h em cada semana. Nas seções abaixo são descritas as atividades realizadas ao longo das *Sprints*.

3.4.1 Sprint 0

Na Sprint 0 foram desenvolvidos os artefatos e diagramas UML: Lista de Histórias de Usuário (Apêndice A), Diagrama de Casos de Uso (Apêndice B), Especificações dos Casos de Uso com interfaces (Apêndice C), Diagramas de Classe (Apêndice D), Diagramas de Sequência (Apêndice E) e Modelo Lógico de Dados (Apêndice F).

3.4.2 Sprint 1

Na Sprint 1 foi desenvolvido o módulo de registro de abastecimentos na aplicação. Foram desenvolvidas as funcionalidades de registrar, modificar e apagar, juntamente com as telas. Foram realizados testes para preparar a integração desses módulos com os dos próximos sprints, além de testes unitários em cada endpoint da API.

3.4.3 Sprint 2

Na Sprint 2 foi desenvolvida a funcionalidade de visualizar relatórios gráficos do módulo de abastecimentos. Foram pesquisadas boas práticas de experiência do usuário visando exibir de maneira intuitiva e útil as estatísticas. Testou-se bibliotecas gráficas da linguagem para implementação do gráfico.

Após os testes de bibliotecas, foi implementada a funcionalidade, juntamente com a tela e sua integração com os módulos. Foram realizados testes unitários e de integração.

3.4.4 Sprint 3

Na Sprint 3 foi implementado o módulo de registro de gastos. Foram desenvolvidas as funções de registrar, modificar e apagar. Realizou-se testes de integração e de unidade na API.

3.4.5 Sprint 4

Na Sprint 4 foi desenvolvida a funcionalidade de visualizar relatório sobre os gastos cadastrados. Implementou-se a função de visualizar os dados separados por tipo e mês. Foram feitos testes de integração da funcionalidade com o resto do sistema, além de testes unitários de visualização do relatório.

3.4.6 Sprint 5

Na Sprint 5 implementou-se o módulo de registrar manutenções. Foram pesquisados os tipos de manutenções, peças e características mais comuns. Com base nas pesquisas, foram desenvolvidas as funcionalidades de registrar, modificar e apagar uma manutenção, além das telas e sua integração com o restante do sistema. Foram realizados testes de unidade e de integração.

3.4.7 Sprint 6

Na Sprint 6 foi implementado o módulo de lembretes. Foram desenvolvidas as funções de registrar, modificar e apagar um lembrete, juntamente das telas e sua integração com o restante do sistema. Realizou-se testes de integração e de unidade.

3.4.8 Sprint 7

Na Sprint 7 foi implementada a função de visualizar o melhor combustível. Foram implementadas as funcionalidades, juntamente com suas respectivas telas e integrações com o sistema. Foram feitos testes de unidade e de integração.

4 APRESENTAÇÃO DOS RESULTADOS

Neste capítulo será apresentado o software desenvolvido, Autocasher. A seguir, será representada a arquitetura do sistema, e, em seguida, serão listadas todas as funcionalidades do software de forma descritiva e visual, através de sua interface gráfica.

4.1 ARQUITETURA DO SISTEMA

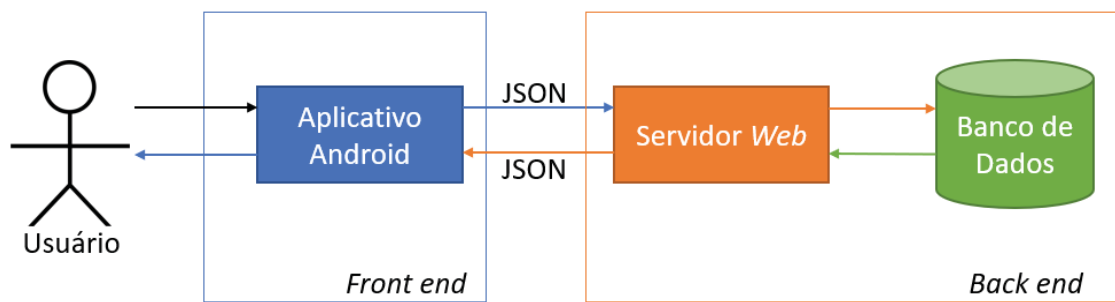
O sistema foi dividido em duas partes: *Front end* e *Back end*. Essas partes se comunicam através da internet utilizando requisições com o protocolo HTTP e notação JSON. O protocolo de camada de aplicação *Hypertext Transfer Protocol* (HTTP) é utilizado para comunicação entre clientes e servidores *web*, aplicando o modelo cliente-servidor (MOZILLA, 2021). A notação em formato texto *JavaScript Object Notation* (JSON) é baseada no *JavaScript* e permite troca de dados com leitura intuitiva (JSON, 2021).

O *front end* é responsável por permitir interação e *input* do usuário com o sistema e suas funcionalidades e exibir as informações obtidas ao executar requisições no *back end*. Foi desenvolvido com a linguagem de programação Java e bibliotecas como MPAndroidChart para exibição de gráficos e *retrofit2* para envio de requisições HTTP.

O *back end* é responsável por executar as requisições recebidas do cliente *front end*, realizar o processamento das informações, comunicar-se com o banco de dados para realizar a persistência de dados e responder ao cliente as informações necessárias. Foi implementado utilizando a linguagem Java com o *framework Spring Boot* e banco de dados PostgreSQL.

A Figura 14 apresenta a arquitetura do Autocasher.

FIGURA 12 - ARQUITETURA DO AUTOCASHER



FONTE: O Autor (2021)

4.2 SOFTWARE

O Autocasher exibe primeiramente a tela de início, onde é possível visualizar a logo, nome do software, nome do desenvolvedor, nome da universidade e botão “COMEÇAR” conforme visto na FIGURA 13.

FIGURA 13 - TELA DE INÍCIO



FONTE: O Autor (2021)

Ao entrar no sistema é apresentada primeiramente a tela de Estatísticas. São exibidas métricas e estatísticas de todos os registros realizados pelo usuário durante o período selecionado, como mostra a FIGURA 14. O período padrão é 90 dias, podendo variar entre: 15 dias, 30 dias, 90 dias, 1 ano, 2 anos e 5 anos. Nessa tela são apresentadas as estatísticas de quantidade de abastecimentos, manutenções, lembretes e gastos. É calculado o preço médio pago por litro, o consumo médio registrado, a data do lembrete mais próximo e o total de despesas registradas, sendo a soma dos valores de abastecimentos, manutenções e gastos. Em seguida são apresentados os gráficos de despesas separadas por tipo, soma de gastos por mês, soma de gastos por tipo e o cálculo do consumo separado por mês.

FIGURA 14 - TELA DE ESTATÍSTICAS



FONTE: O Autor (2021)

A partir deste ponto, o usuário pode navegar para as outras telas do sistema através da barra de navegação presente na parte inferior da tela. As outras telas são: Abastecimento, Manutenção, Lembrete e Gasto. Em cada uma dessas telas pode-se visualizar a lista de registros do tipo selecionado, além de criar, editar e deletar um registro.

Ao selecionar a tela de Abastecimento, será exibida a FIGURA 15. Nessa tela é apresentada a lista de abastecimentos registrados durante o período selecionado, além dos botões para registrar um novo abastecimento (FIGURA 16), editar ou deletar um abastecimento existente e calcular o melhor combustível (FIGURA 17). Na tela de calcular o melhor combustível, o usuário preenche o valor do litro de gasolina e álcool, e o sistema retorna o melhor combustível.

FIGURA 15 - TELA DE ABASTECIMENTOS



FONTE: O Autor (2021)

FIGURA 16 – TELA DE EDITAR ABASTECIMENTO

21:46

Editar Abastecimento

Quantidade Litros

38.0

Valor do litro

\$ 4.9

Data

25/abr./2021

Info adicional

Odômetro

50795

Tanque cheio

Abastecimento anterior em falta

SALVAR

DESCARTAR

FONTE: O Autor (2021)

FIGURA 17 - TELA DE CALCULAR MELHOR COMBUSTÍVEL

21:46

Abastecimento

15 dias

Mais novos

ABASTECIMENTO 0,00

CALCULAR MELHOR COMBUSTÍVEL

PREÇO GASOLINA (L)

4.99

PREÇO ÁLCOOL (L)

3.40

CALCULAR

Melhor combustível:

GASOLINA

Abastecimento

FONTE: O Autor (2021)

Selecionando a tela de Manutenção, será apresentada a FIGURA 18. Nessa tela é possível visualizar a lista de manutenções realizadas durante o período selecionado e ordenadas conforme a seleção de ordenação. O botão para criar uma manutenção pode ser localizado na parte inferior direita da tela. A FIGURA 19 mostra a tela de criar ou editar uma manutenção.

FIGURA 18 - TELA DE MANUTENÇÕES



FONTE: O Autor (2021)

FIGURA 19 - TELA EDITAR MANUTENÇÃO

21:48

✕ Editar Manutencao

Tipo de manutenção
📄 Emergência

Valor
\$ 70.0

Data
📅 29/mar./2021

Local
📍 Posto Shell

Info adicional
📄 Garantia 2 anos

Peças
🔧 Palhetas parabrisa

SALVAR

DESCARTAR

FONTE: O Autor (2021)

Ao selecionar a tela Lembrete, será apresentada a FIGURA 20. Nessa interface são listados os lembretes agendados para o período selecionado, além dos agendados para datas futuras. O botão para agendar um novo lembrete pode ser localizado na parte inferior direita da interface. A FIGURA 21 exibe a interface da criação de novos lembretes ou edição de lembretes existentes.

Ao registrar um lembrete, este é registrado em uma base de dados armazenada na nuvem. Foi adicionado à essa base um script que verifica diariamente se já foi alcançada a data agendada para o lembrete. Caso positivo, será enviado um e-mail de lembrete ao usuário, conforme visto na FIGURA 22.

FIGURA 20 - TELA DE VISUALIZAR LEMBRETES



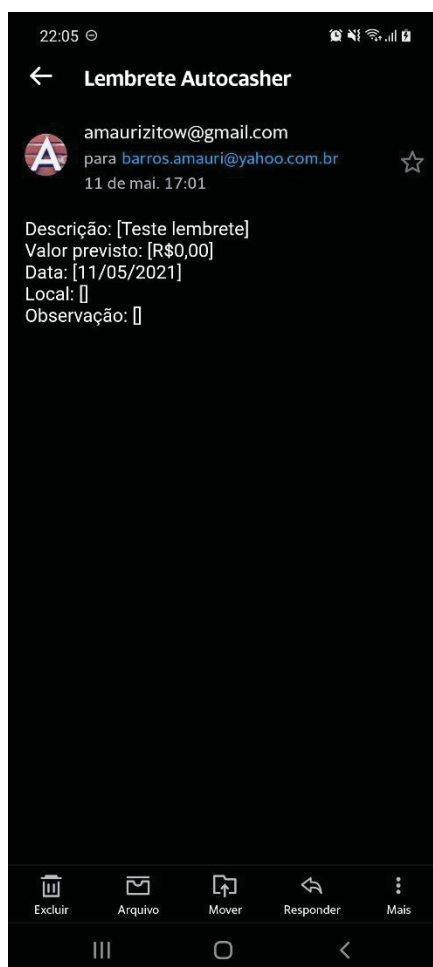
FONTE: O Autor (2021)

FIGURA 21 - TELA DE EDITAR LEMBRETE



FONTE: O Autor (2021)

FIGURA 22 - E-MAIL DE LEMBRETE RECEBIDO



FONTE: O Autor (2021)

Ao escolher a tela de Gasto, será exibida a FIGURA 23. Nessa interface o usuário pode visualizar a lista de gastos registrados durante o período selecionado e ordenado conforme a ordenação escolhida. O botão para criar um Gasto é localizado na parte inferior direita da tela. A FIGURA 24 exhibe a tela usada para criação de um novo gasto ou edição de um gasto existente.

FIGURA 23 - TELA DE VISUALIZAR GASTOS



FONTE: O Autor (2021)

FIGURA 24 - TELA DE EDITAR GASTO



FONTE: O Autor (2021)

4.3 CONSIDERAÇÕES DA APRESENTAÇÃO DOS RESULTADOS

As telas apresentadas neste capítulo demonstram a implementação final do escopo de funcionalidades do projeto. Na seção anterior constam descrições e explicações das telas e suas respectivas ilustrações visuais.

O *layout* e design das telas foi definido pensando na facilidade de uso e intuitividade. As cores e ícones presentes em cada tela visam oferecer ao usuário uma forma de associar intuitivamente a função da tela selecionada. Os tipos de registros e suas cores associadas são:

- Abastecimentos: laranja;
- Manutenções: verde;
- Lembretes: roxo;
- Gastos: rosa;
- Estatísticas gerais: azul.

A barra de navegação localizada na parte inferior das telas exibe os ícones associados à cada tela, e, após o clique em um ícone, este é expandido e apresenta seu nome abaixo, indicando a tela selecionada.

5 CONSIDERAÇÕES FINAIS

Neste trabalho foi desenvolvido o software proposto, Autocasher. Esse sistema visa facilitar a gestão de gastos automotivos através de uma aplicação para celulares *Android* onde é possível registrar abastecimentos, gastos, lembretes e manutenções e visualizar estatísticas como consumo mensal, gastos mensais, média de preço de combustível, entre outros. Além disso, é possível calcular o combustível mais vantajoso.

Algumas das principais dificuldades encontradas durante o processo de desenvolvimento do projeto foram relacionadas à sua implementação técnica. A complexidade técnica para implementar as diversas funcionalidades e integrar suas tecnologias se mostrou um grande desafio, propiciando dessa forma um importante aprendizado para o futuro profissional do autor.

Outra situação que impactou o desenvolvimento do projeto foi a pandemia do Covid-19, gerando momentos de incerteza e dificuldades para a sociedade como um todo. Em vista desse fato, a data de entrega do projeto foi adiada, fator de vital importância para a qualidade da entrega final desse projeto, permitindo um trabalho mais cuidadoso e aprendizados proveitosos.

5.1 TRABALHOS FUTUROS

Embora esse projeto tenha atingido seus objetivos propostos de forma satisfatória, algumas funcionalidades relevantes deixaram de ser implementadas em detrimento das funcionalidades do escopo principal, sendo, dessa forma, uma oportunidade para trabalhos futuros. Algumas dessas oportunidades são:

5.1.1 Autenticação

Durante a definição de escopo do projeto a funcionalidade de autenticação não foi contemplada. Dessa forma, o projeto foi desenvolvido conforme inicialmente planejado. Ao perceber que a funcionalidade de autenticação poderia ser adicionada ao projeto, seu desenvolvimento já estava em estágio avançado, tornando essa inclusão inviável.

Portanto, sugere-se como trabalho futuro a inclusão de um sistema de *login* para que vários usuários possam utilizar a ferramenta e ter acesso somente a registros que lhe dizem respeito.

5.1.2 Funcionamento *offline*

Para a utilização do Autocasher, o usuário precisa necessariamente possuir conexão com a internet, visto que a aplicação depende de um *webservice*. Uma proposta de melhoria é a implementação de funcionamento *offline*, permitindo ao usuário registrar e visualizar o aplicativo com dados armazenados localmente. Na presença de conexão à internet, a aplicação faria a sincronização desses dados com os dados armazenados na nuvem.

5.1.3 Melhoria da interface gráfica

Apesar da tentativa de utilização do Material Design durante a confecção das interfaces, percebeu-se que a metodologia não foi utilizada de forma correta. Dessa forma, propõe-se a melhoria da interface gráfica visando uma melhor adaptação ao Material Design.

REFERÊNCIAS

ABB Group. **4 types of maintenance strategy, which one to chose?** Disponível em: <<https://new.abb.com/medium-voltage/service/maintenance/feature-articles/4-types-of-maintenance-strategy-which-one-to-choose>>. Acesso em: 12/05/2021

AGÊNCIA Brasil. **Celular se torna principal forma de acesso à internet no Brasil.** Brasília, 2018 em: <<https://agenciabrasil.ebc.com.br/geral/noticia/2018-07/celular-se-torna-principal-forma-de-acesso-internet-no-brasil>>. Acesso em: 10/05/2021

AGÊNCIA Brasil. **Brasil é o 5º país em ranking de uso diário de celulares no mundo.** Brasília, 2019 em: <<https://agenciabrasil.ebc.com.br/geral/noticia/2019-01/brasil-foi-5o-pais-em-ranking-de-uso-diario-de-celulares-no-mundo>>. Acesso em: 12/05/2021

AGÊNCIA DE NOTÍCIAS DO PARANÁ (AEN). **Não fazer manutenção do veículo triplica risco de acidente.** Paraná, 2016 em: <<http://www.aen.pr.gov.br/modules/debaser/visualizar.php?audiovideo=1&xfid=64835&tit=Nao-fazer-manutencaodo-veiculo-triplicarisco-de-acidente>>. Acesso em: 10/05/2021

APP ANNIE. **The State of Mobile 2019.** Estados Unidos, 2019 em: <<https://www.appannie.com/en/go/state-of-mobile-2019/>>. Acesso em: 12/05/2021

BARBOSA, Juarez. **Estratégia de Manutenção.** Brasil, 2017 em: <<https://consultoriaengenharia.com.br/engenharia-de-manutencao/estrategia-de-manutencao/>>. Acesso em: 12/05/2021

BECK et al. **Manifesto para Desenvolvimento Ágil de Software.** Estados Unidos, 2001 em: <<https://agilemanifesto.org/iso/ptbr/manifesto.html>>. Acesso em: 12/05/2021

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **The unified modeling language user guide.** Reading: Addison Wesley, 1999.

DRIVVO. **O Drivvo.** Disponível em: <<https://www.drivvo.com/pt/>>. Acesso em: 12/05/2021

EDUREKA. **What is Java?** A Beginner's Guide to Java and it's Evolution. Disponível em: <<https://www.edureka.co/blog/what-is-java/>>. Acesso em: 12/05/2021

ÉPOCA Negócio. **Brasil tem 230 milhões de smartphones em uso.** Disponível em: <<https://epocanegocios.globo.com/Tecnologia/noticia/2019/04/brasil-tem-230-milhoes-de-smartphones-em-uso.html>>. Acesso em: 10/05/2021

FORD Motor Company. **FORD FOCUS Manual do proprietário**. Brasil, 2018 em: <https://www.ford.com.br/content/dam/Ford/website-assets/latam/br/servico-ao-cliente/manuais/2019/manuais-do-proprietario/Manual_Proprietario_Focus_2019.pdf>. Acesso em: 10/05/2021

FOWLER, M.; HIGHSMITH, J. **The Agile Manifesto**. Estados Unidos, 2001 em: <<http://users.jyu.fi/~mieijala/kandimateriaali/Agile-Manifesto.pdf>>. Acesso em: 20/05/2021

FUELIO. **About**. Disponível em: <<https://www.fuel.io/>>. Acesso em: 12/05/2021

GOSLING, J.; JOY, B.; STEELE, G.; BRACHA, G.; BUCKLEY, A. **The Java Language Specification: Java SE 8 Edition**. Redwood City: Oracle America, 2015.

GUEDES, Gilleanes T. A. **UML 2: Uma Abordagem Prática**. São Paulo: Novatec, 2011.

GURU99. **What is Java?** Definition, Meaning & Features of Java Platforms. Disponível em: <<https://www.guru99.com/java-platform.html>>. Acesso em: 12/05/2021

INFOQ. **Exploring Micro-Frameworks: Spring Boot**. Disponível em: <<https://www.infoq.com/articles/microframeworks1-spring-boot/>>. Acesso em: 12/05/2021

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA (IBGE). **PNAD Contínua**. Disponível em: <<https://agenciadenoticias.ibge.gov.br/agencia-sala-de-imprensa/2013-agencia-de-noticias/releases/24109-pnad-continua-taxa-de-desocupacao-e-de-12-4-e-taxa-de-subutilizacao-e-de-24-6-no-trimestre-encerrado-em-fevereiro-de-2019>>. Acesso em: 12/05/2021

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA (IBGE). **Características gerais dos domicílios e dos moradores**: 2018. Rio de Janeiro, 2019 em: <<https://biblioteca.ibge.gov.br/index.php/biblioteca-catalogo?view=detalhes&id=2101654>>. Acesso em: 12/05/2021

JSON. **Introdução ao JSON**. Disponível em: <<https://www.json.org/json-pt.html>>. Acesso em: 12/05/2021

MOUBRAY, J. **Reliability-centered maintenance**. Nova York: Butterworth-Heinemann, 1997.

MOZILLA. **Visão Geral do HTTP**. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/HTTP>>. Acesso em: 12/05/2021

ORACLE. **What is Java technology and why do I need it?** Disponível em: <https://java.com/en/download/help/whatis_java.html>. Acesso em: 12/05/2021

PAGBRASIL. **Mobile no Brasil:** estatísticas de uso e perfil do usuário. Brasil, 2019 em: <<https://www.pagbrasil.com/pt-br/noticias/mobile-no-brasil/>>. Acesso em: 12/05/2021

POSTGRESQL. **What is PostgreSQL?** Disponível em: <<https://www.postgresql.org/about/>>. Acesso em: 12/05/2021

SIMPLY AUTO. **Features.** Disponível em: <<https://simplyauto.app/>>. Acesso em: 12/05/2021

SINDICATO NACIONAL DA INDÚSTRIA DE COMPONENTES PARA VEÍCULOS AUTOMOTORES (SINDIPEÇAS). **Relatório da Frota Circulante.** São Paulo, 2019 em: <[http://www.automotivebusiness.com.br/abinteligencia/pdf/R_FrotaCirculante_2019\(v4\).pdf](http://www.automotivebusiness.com.br/abinteligencia/pdf/R_FrotaCirculante_2019(v4).pdf)>. Acesso em: 10/05/2021

SCRUM. **What is Scrum?** Disponível em: <<https://www.scrum.org/resources/what-is-scrum>>. Acesso em: 12/05/2021

SCHWABER, K.; SUTHERLAND, J. **The Scrum Guide.** Estados Unidos, 2020 em: <<https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>>. Acesso em: 21/05/2021

SPRING. **Spring Framework.** Califórnia, 2021 em: <<https://spring.io/projects/spring-framework>>. Acesso em: 12/05/2021

STACK Overflow. **2020 Developer Survey.** Disponível em: <<https://insights.stackoverflow.com/survey/2020>>. Acesso em: 20/05/2021

TRANSPORTE Econômico. **5 dicas para aumentar a vida útil do seu veículo.** Disponível em: <<https://transporteeconomico.com.br/5-dicas-para-aumentar-vida-util-do-seu-veiculo/>>. Acesso em: 22/06/2020

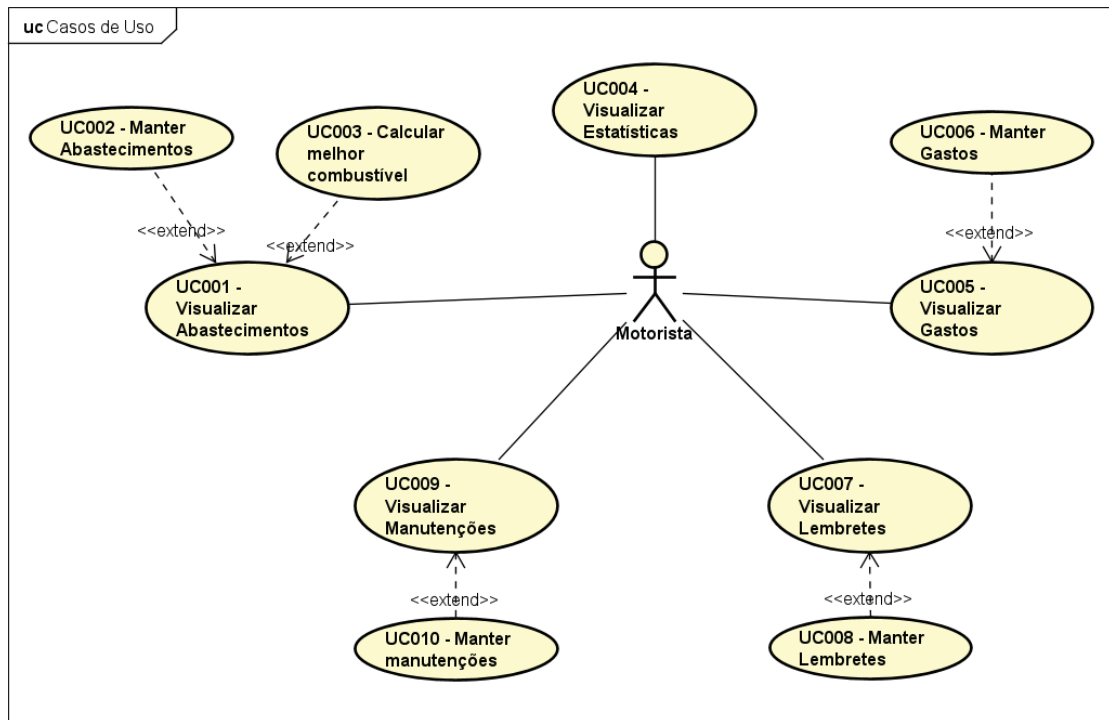
TRELLO. **Sobre o Trello.** Disponível em: <<https://trello.com/about>>. Acesso em: 12/05/2021

APÊNDICE A – LISTA DE HISTÓRIAS DE USUÁRIO

- Story 01 - Eu, como <motorista> desejo <histórico de abastecimento> para que <eu possa controlar as datas que abasteço e quanto eu paguei>
- Story 02 - Eu, como <motorista> desejo <gráfico de consumo km/litro e estatísticas> para que <eu possa ter uma noção de como está o consumo do meu carro e minhas despesas>
- Story 03 - Eu, como <motorista> desejo <visualizar qual combustível é mais vantajoso> para que <eu economize dinheiro com os preços de combustíveis tão instáveis atualmente>
- Story 04 - Eu, como <motorista> desejo <cadastro de gastos> para que <eu possa ter em um só lugar os gastos que meu carro gera, como IPVA, seguro, multas e financiamento>
- Story 05 - Eu, como <motorista> desejo <previsão de gastos> para que <eu possa visualizar quanto vou gastar mensalmente e anualmente com despesas do carro>
- Story 06 - Eu, como <motorista> desejo <registro de realização de serviços> para que <eu possa controlar quais serviços já realizei, quando, e qual foi o preço. para saber se algum mecânico está me enganando e cobrando muito caro>
- Story 07 - Eu, como <motorista> desejo <lembretes de manutenções regulares> para que <eu pare de esquecer de realizar as manutenções na hora certa>
- Story 08 - Eu, como <motorista> desejo <programar lembretes de manutenção> para que <eu pare de esquecer de manutenções>

APÊNDICE B – DIAGRAMAS DE CASOS DE USO

FIGURA 25 - DIAGRAMAS DE CASOS DE USO



FONTE: O Autor (2021)

APÊNDICE C – ESPECIFICAÇÕES DE CASOS DE USO

Especificação do Caso de Uso 001 - Visualizar Abastecimentos

Data View:

FIGURA 26 - VISUALIZAR ABASTECIMENTOS



FONTE: O Autor (2021)

Pré-condições: O motorista selecionou a opção “Abastecimento” na barra de navegação.

Pós-condições: O sistema exibe a lista de abastecimentos registrados durante o período selecionado e ordenados conforme ordenação selecionada.

Ator Primário: Motorista

Fluxo de Eventos Principal

1. O sistema carrega os abastecimentos previamente registrados durante o período selecionado;

2. O sistema ordena a lista de abastecimentos conforme seleção de ordenação;
3. O sistema apresenta a tela;
4. O motorista visualiza os abastecimentos registrados (A1)(A2)(A3)(A4);
5. Fim do caso de uso;

Fluxos Alternativos

A1: Visualizar puxando tela para cima

1. O sistema apresenta mais abastecimentos do que a tela comporta;
2. O motorista puxa a tela para cima usando gesto de navegação padrão na tela;
3. O sistema exibe os próximos abastecimentos;
4. O motorista visualiza os abastecimentos registrados;
5. Fim do caso de uso;

A2: Visualizar detalhes do abastecimento

1. O motorista clica em um dos abastecimentos da lista;
2. O sistema expande o item da lista, exibindo mais detalhes do abastecimento, juntamente com opções de editar e deletar;
3. O motorista visualiza os detalhes do abastecimento;
4. Fim do caso de uso;

A3: Manter abastecimento

1. O motorista clica em um dos abastecimentos da lista;
2. O sistema expande o item da lista, exibindo mais detalhe do abastecimento, juntamente com opções de editar e deletar;
3. O motorista visualiza os detalhes do abastecimento;
4. O motorista clica no botão para editar o abastecimento (UC002);
5. Fim do caso de uso;

A4: Calcular melhor combustível

1. O motorista clica no botão para calcular o melhor abastecimento (UC003);
2. Fim do caso de uso;

Especificação do Caso de Uso 002 - Manter Abastecimento

Data View:

FIGURA 27 - MANTER ABASTECIMENTO

21:46

✕ Editar Abastecimento

Quantidade Litros

38.0

Valor do litro

\$ 4.9

Data

25/abr./2021

Info adicional

Odômetro

50795

Tanque cheio

Abastecimento anterior em falta

SALVAR

DESCARTAR

FONTE: O Autor (2021)

Pré-condições: O motorista clicou no botão para criar abastecimento ou editar abastecimento existente.

Pós-condições: O sistema exibe a lista de abastecimentos registrados.

Ator Primário: Motorista

Fluxo de Eventos Principal

1. O sistema exibe a tela;
2. O motorista preenche os campos Quantidade Litros, Valor do litro, Data, Info adicional, Odômetro, Tanque cheio e Abastecimento anterior em falta;
3. O usuário clica no botão Salvar para finalizar o registro (A1);
4. O sistema valida os dados preenchidos;
5. O sistema persiste os dados no banco;
6. Fim do caso de uso;

Fluxos Alternativos

A1: O motorista clica em voltar

1. O motorista seleciona o botão de descartar;
2. O sistema exibe a tela de histórico de abastecimentos;
3. Fim do caso de uso;

Especificação do Caso de Uso 003 – Calcular Melhor Combustível**Data View:**

FIGURA 28 - CALCULAR MELHOR COMBUSTÍVEL



FONTE: O Autor (2021)

Pré-condições: O motorista clica no botão Visualizar Melhor Combustível**Pós-condições:** O sistema calcula o melhor combustível e exibe.**Ator Primário:** Motorista**Regra de Negócio:** Se preço do álcool for menor do que 68% do preço da gasolina, o melhor combustível é o álcool. Se o preço for igual ou maior, o melhor combustível é a gasolina.

Fluxo de Eventos Principal

1. O sistema exibe a tela;
2. O motorista preenche os campos Preço Gasolina e Preço Álcool;
3. O usuário clica no ícone para calcular (A1);
4. O sistema calcula o melhor combustível;
5. O sistema exibe o melhor combustível;
6. Fim do caso de uso;

Fluxos Alternativos

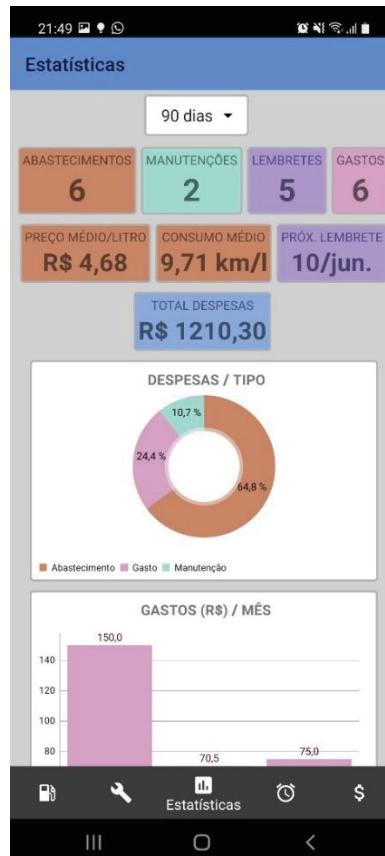
A1: O motorista clica fora da tela de Visualizar Melhor Combustível

1. O motorista clica fora da tela de Visualizar Melhor Combustível;
2. O sistema exibe a tela de histórico de abastecimentos;
3. Fim do caso de uso;

Especificação do Caso de Uso 004 - Visualizar Estatísticas

Data View:

FIGURA 29 - VISUALIZAR ESTATÍSTICAS



FONTE: O Autor (2021)

Pré-condições: O motorista realizou registros durante o período selecionado.

Pós-condições: O sistema calcula as estatísticas dos registros registrados durante o período selecionado e exibe.

Ator Primário: Motorista

Fluxo de Eventos Principal

1. O sistema carrega os dados relativos aos registros cadastrados durante o período selecionado;
2. O sistema calcula as estatísticas como consumo, média de preço de combustíveis, total de despesas e total de gastos;
3. O sistema exibe os gráficos relativos às estatísticas calculadas;
4. O motorista visualiza os gráficos apresentados (A1);

Fluxos Alternativos

A1: Visualizar puxando tela para cima

1. O sistema apresenta mais gráficos do que a tela comporta;
2. O motorista puxa a tela para cima usando gesto de navegação padrão na tela;
3. O sistema exibe os próximos gráficos;
4. O motorista visualiza os gráficos;
5. Fim do caso de uso;

Especificação do Caso de Uso 005 - Visualizar Gastos

Data View:

FIGURA 30 - VISUALIZAR GASTOS



FONTE: O Autor (2021)

Pré-condições: O motorista selecionou a opção “Gasto” na barra de navegação.

Pós-condições: O sistema exibe a lista de gastos registrados durante o período selecionado e ordenados conforme ordenação selecionada.

Ator Primário: Motorista

Fluxo de Eventos Principal

1. O sistema carrega os gastos já registrados;
2. O sistema apresenta a tela (Tela 1);
3. O motorista visualiza os gastos já registrados (A1)(A2)(A3);
4. Fim do caso de uso;

Fluxos Alternativos

A1: Visualizar mais gastos puxando tela para cima

1. O sistema apresenta mais gastos do que a tela comporta;

2. O motorista puxa a tela para cima usando gesto de navegação padrão na tela;
3. O sistema exibe os próximos gastos;
4. O motorista visualiza os gastos registrados;
5. Fim do caso de uso;

A2: Visualizar detalhes do gasto

1. O motorista clica em um dos gastos da lista;
2. O sistema expande o item da lista, exibindo mais detalhes do gasto, juntamente com opções de editar e deletar;
3. O motorista visualiza os detalhes do gasto;
4. Fim do caso de uso;

A3: Manter gasto

1. O motorista clica em um dos gastos da lista;
2. O sistema expande o item da lista, exibindo mais detalhes do gasto, juntamente com opções de editar e deletar;
3. O motorista visualiza os detalhes do gasto;
4. O motorista clica no botão para editar o gasto (UC006);
5. Fim do caso de uso;

Especificação do Caso de Uso 006 - Manter Gasto**Data View:**

FIGURA 31 - MANTER GASTO

22:01

✕ Editar Gasto

Tipo de gasto
 ⓘ Pedágio

Valor
 \$ 45.0

Data
 📅 13/mai./2021

Local
 📍 Litoral Paraná

Info adicional
 ⓘ Preço subiu

Odômetro
 🚗 0.0

SALVAR

DESCARTAR

FONTE: O Autor (2021)

Pré-condições: O motorista clicou no botão de criar gasto ou editar gasto existente.

Pós-condições: O sistema exibe a tela de histórico de gastos.

Ator Primário: Motorista

Fluxo de Eventos Principal

1. O sistema exibe a tela;
2. O motorista preenche os campos Tipo de Gasto, Valor, Data, Local, Info adicional e Odômetro;
3. O usuário clica no ícone para finalizar o registro (A1);
4. O sistema valida os dados preenchidos (E1);
5. O sistema persiste os dados no banco;
6. Fim do caso de uso;

Fluxos de Exceção

E1: Falta preencher campos obrigatórios

1. O sistema exibe a mensagem “Favor preencher os campos obrigatórios”;
2. O motorista preenche os campos obrigatórios faltantes;
3. O sistema valida os dados preenchidos;
4. O sistema persiste os dados no banco;
5. Fim do caso de uso;

Fluxos Alternativos

A1: O motorista clica em voltar

1. O motorista seleciona o botão de voltar;
2. O sistema exibe a tela de histórico de gastos;
3. Fim do caso de uso;

Especificação do Caso de Uso 007 - Visualizar Lembretes

Data View:

FIGURA 32 - VISUALIZAR LEMBRETES



FONTE: O Autor (2021)

Pré-condições: O motorista selecionou a opção “Lembrete” na barra de navegação.

Pós-condições: O sistema exibe a lista de lembretes registrados durante o período selecionado e ordenados conforme ordenação selecionada.

Ator Primário: Motorista

Fluxo de Eventos Principal

1. O sistema carrega os lembretes já registrados;
2. O sistema apresenta a tela (Tela 1);
3. O motorista visualiza os lembretes já registrados (A1)(A2)(A3);
4. Fim do caso de uso;

Fluxos Alternativos

A1: Visualizar mais gastos puxando tela para cima

1. O sistema apresenta mais lembretes do que a tela comporta;
2. O motorista puxa a tela para cima usando gesto de navegação padrão na tela;
3. O sistema exibe os próximos lembretes;
4. O motorista visualiza os lembretes registrados;
5. Fim do caso de uso;

A2: Visualizar detalhes dos lembretes

1. O motorista clica em um dos lembretes da lista;
2. O sistema expande o item da lista, exibindo mais detalhes do lembrete, juntamente com opções de editar e deletar;
3. O motorista visualiza os detalhes do lembrete;
4. Fim do caso de uso;

A3: Manter lembrete

1. O motorista clica em um dos lembretes da lista;
2. O sistema expande o item da lista, exibindo mais detalhes do lembrete, juntamente com opções de editar e deletar;
3. O motorista visualiza os detalhes do lembrete;
4. O motorista clica no botão para editar o lembrete (UC008);
5. Fim do caso de uso;

Especificação do Caso de Uso 008 - Manter Lembrete

Data View:

FIGURA 33 - MANTER LEMBRETE

21:50

✕ Editar Lembrete

Descrição
Trocar pneus

Valor previsto
\$ 1000.0

Data
10/jun./2021

Local
Carrefour

Observação

SALVAR

DESCARTAR

FONTE: O Autor (2021)

Pré-condições: O motorista clicou no botão de criar lembrete ou editar lembrete existente.

Pós-condições: O sistema exibe a lista de lembretes.

Ator Primário: Motorista

Fluxo de Eventos Principal

1. O sistema exibe a tela;
2. O motorista preenche os campos Descrição, Valor previsto, Data, Local e Observação;
3. O usuário clica no ícone para finalizar o registro (A1);
4. O sistema valida os dados preenchidos (E1);
5. O sistema persiste os dados no banco;
6. Fim do caso de uso;

Fluxos de Exceção

E1: Falta preencher campos obrigatórios

1. O sistema exibe a mensagem “Favor preencher os campos obrigatórios”;
2. O motorista preenche os campos obrigatórios faltantes;
3. O sistema valida os dados preenchidos;
4. O sistema persiste os dados no banco;
5. Fim do caso de uso;

Fluxos Alternativos

A1: O motorista clica em voltar

1. O motorista seleciona o botão de voltar;
2. O sistema exibe a tela de lembretes;
3. Fim do caso de uso;

Especificação do Caso de Uso 009 - Visualizar Manutenções

Data View:

FIGURA 34 - VISUALIZAR MANUTENÇÕES



FONTE: O Autor (2021)

Pré-condições: O motorista selecionou a opção “Manutenção” na barra de navegação.

Pós-condições: O sistema exibe a lista de manutenções registradas durante o período selecionado e ordenadas conforme ordenação selecionada.

Ator Primário: Motorista

Fluxo de Eventos Principal

1. O sistema carrega as manutenções já registradas;
2. O sistema apresenta a tela (Tela 1);
3. O motorista visualiza as manutenções já registradas (A1)(A2)(A3);
4. Fim do caso de uso;

Fluxos Alternativos

A1: Visualizar mais manutenções puxando tela para cima

1. O sistema apresenta mais manutenções do que a tela comporta;
2. O motorista puxa a tela para cima usando gesto de navegação padrão na tela;
3. O sistema exibe as próximas manutenções;
4. O motorista visualiza as manutenções registradas;
5. Fim do caso de uso;

A2: Visualizar detalhes da manutenção

1. O motorista clica em uma das manutenções da lista;
2. O sistema expande o item da lista, exibindo mais detalhes da manutenção, juntamente com opções de editar e deletar;
3. O motorista visualiza os detalhes da manutenção;
4. Fim do caso de uso;

A3: Manter manutenção

1. O motorista clica em uma das manutenções da lista;
2. O sistema expande o item da lista, exibindo mais detalhes da manutenção, juntamente com opções de editar e deletar;
3. O motorista visualiza os detalhes da manutenção;
4. O motorista clica no botão para editar a manutenção (UC010);
5. Fim do caso de uso;

Especificação do Caso de Uso 010 - Manter Manutenção

Data View:

FIGURA 35 - MANTER MANUTENÇÃO

21:48

✕ Editar Manutencao

Tipo de manutenção
📌 Emergência

Valor
\$ 70.0

Data
📅 29/mar./2021

Local
📍 Posto Shell

Info adicional
📌 Garantia 2 anos

Peças
🔧 Palhetas parabrisa

SALVAR

DESCARTAR

FONTE: O Autor (2021)

Pré-condições: O motorista clicou no botão para criar manutenção ou editar manutenção existente.

Pós-condições: O sistema exibe a lista de manutenções.

Ator Primário: Motorista

Fluxo de Eventos Principal

1. O sistema exibe a tela;
2. O motorista preenche os campos Tipo de Manutenção, Valor, Data, Local, Info adicional, Peças;
3. O usuário clica no ícone para finalizar o registro (A1);
4. O sistema valida os dados preenchidos (E1);
5. O sistema persiste os dados no banco;
6. Fim do caso de uso;

Fluxos de Exceção

E1: Falta preencher campos obrigatórios

1. O sistema exibe a mensagem “Favor preencher os campos obrigatórios”;
2. O motorista preenche os campos obrigatórios faltantes;
3. O sistema valida os dados preenchidos;
4. O sistema persiste os dados no banco;
5. Fim do caso de uso;

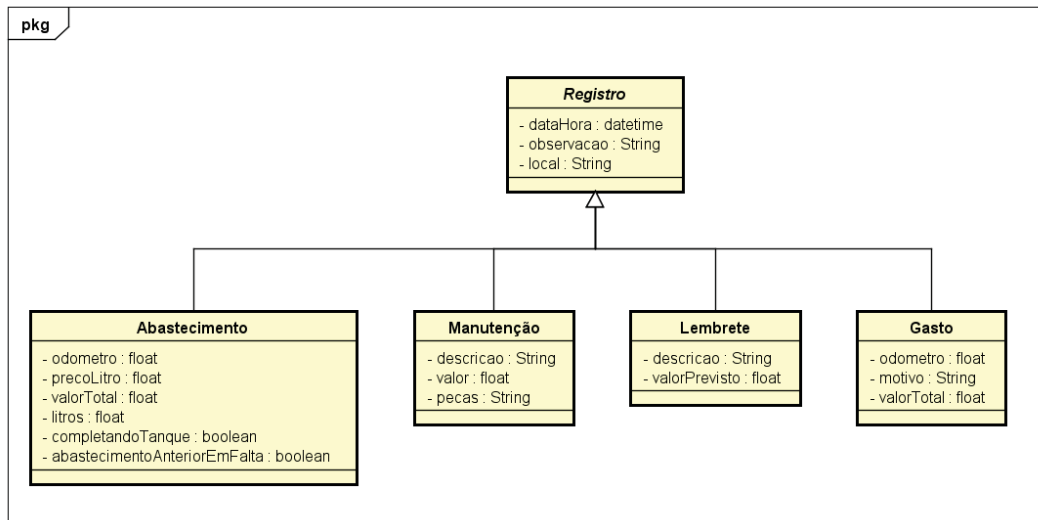
Fluxos Alternativos

A1: O motorista clica em voltar

1. O motorista seleciona o botão de voltar;
2. O sistema exibe a tela de manutenções;
3. Fim do caso de uso;

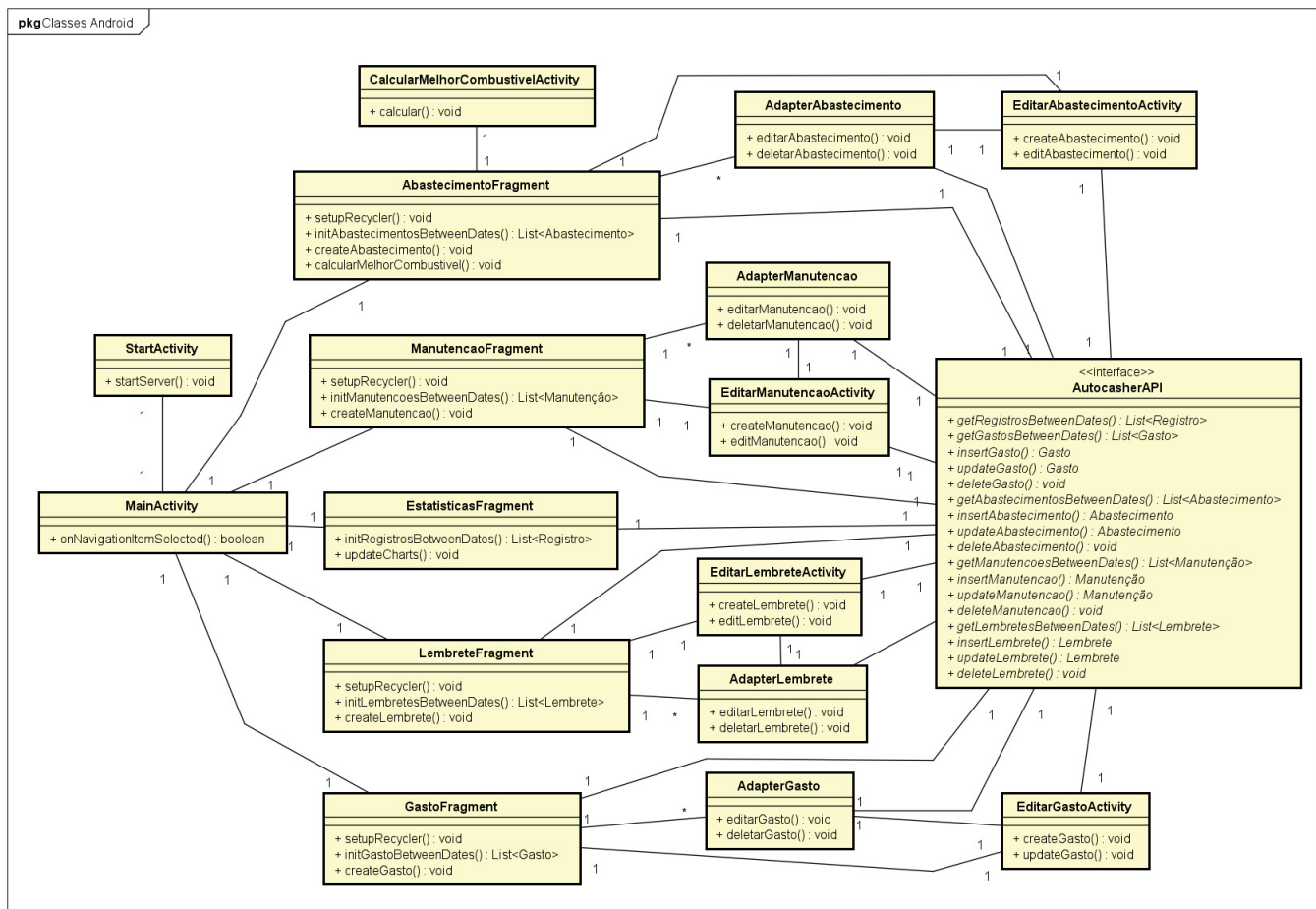
APÊNDICE D – DIAGRAMAS DE CLASSE

FIGURA 36 - DIAGRAMA DE CLASSES MODEL



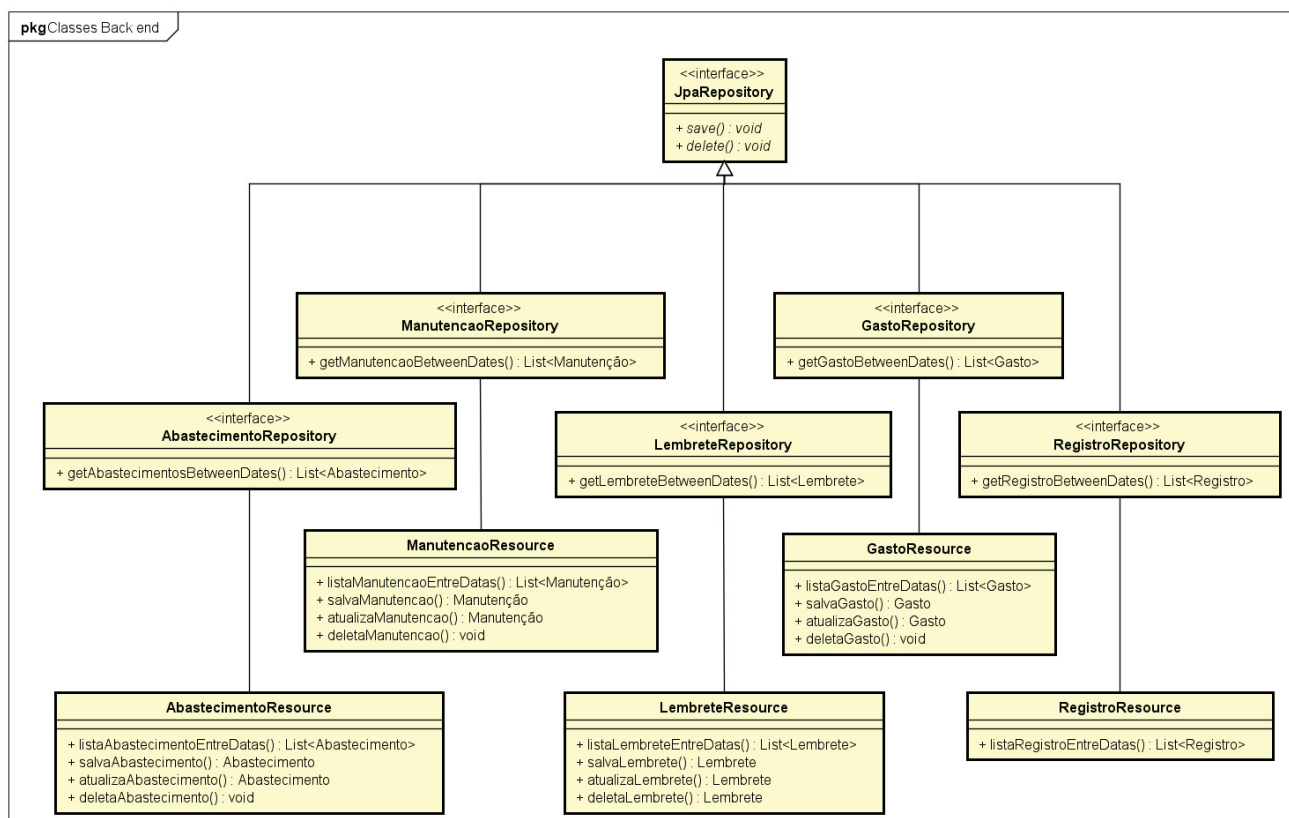
FONTE: O Autor (2021)

FIGURA 37 - DIAGRAMA DE CLASSES ANDROID



FONTE: O Autor (2021)

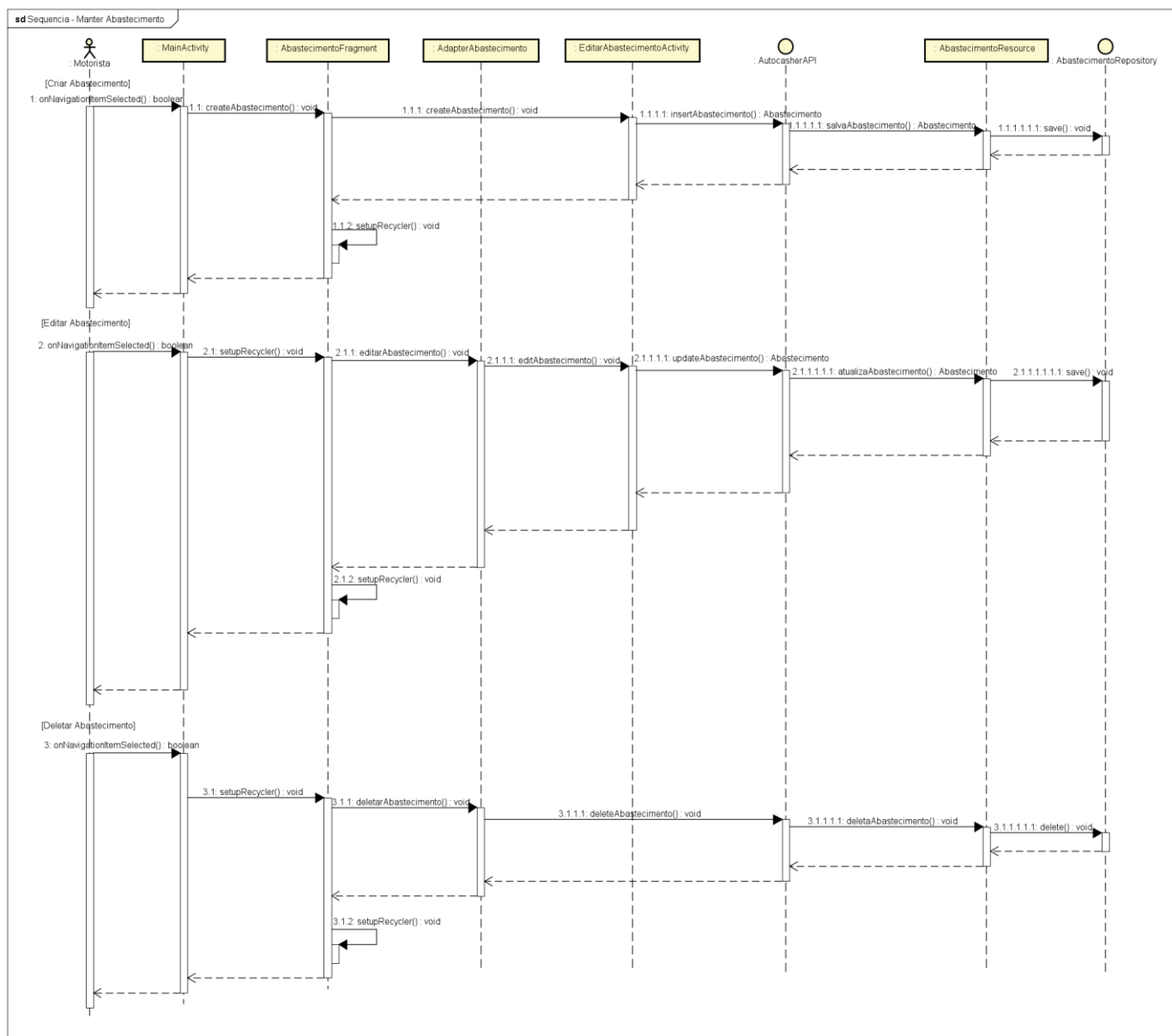
FIGURA 38 - DIAGRAMA DE CLASSES BACK END



FONTE: O Autor (2021)

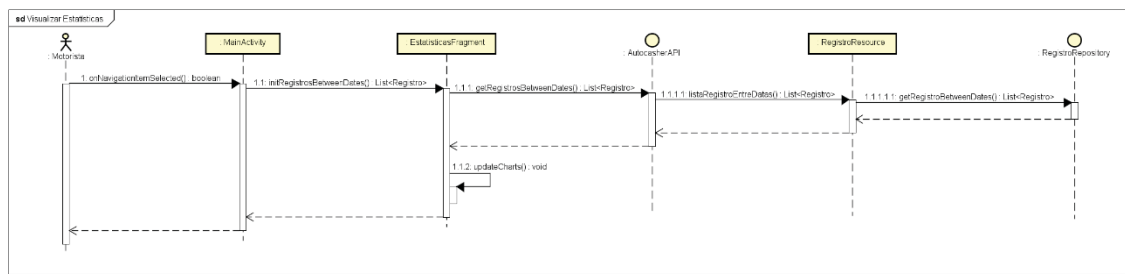
APÊNDICE E – DIAGRAMAS DE SEQUÊNCIA

FIGURA 39 - DIAGRAMA DE SEQUÊNCIA MANTER ABASTECIMENTO



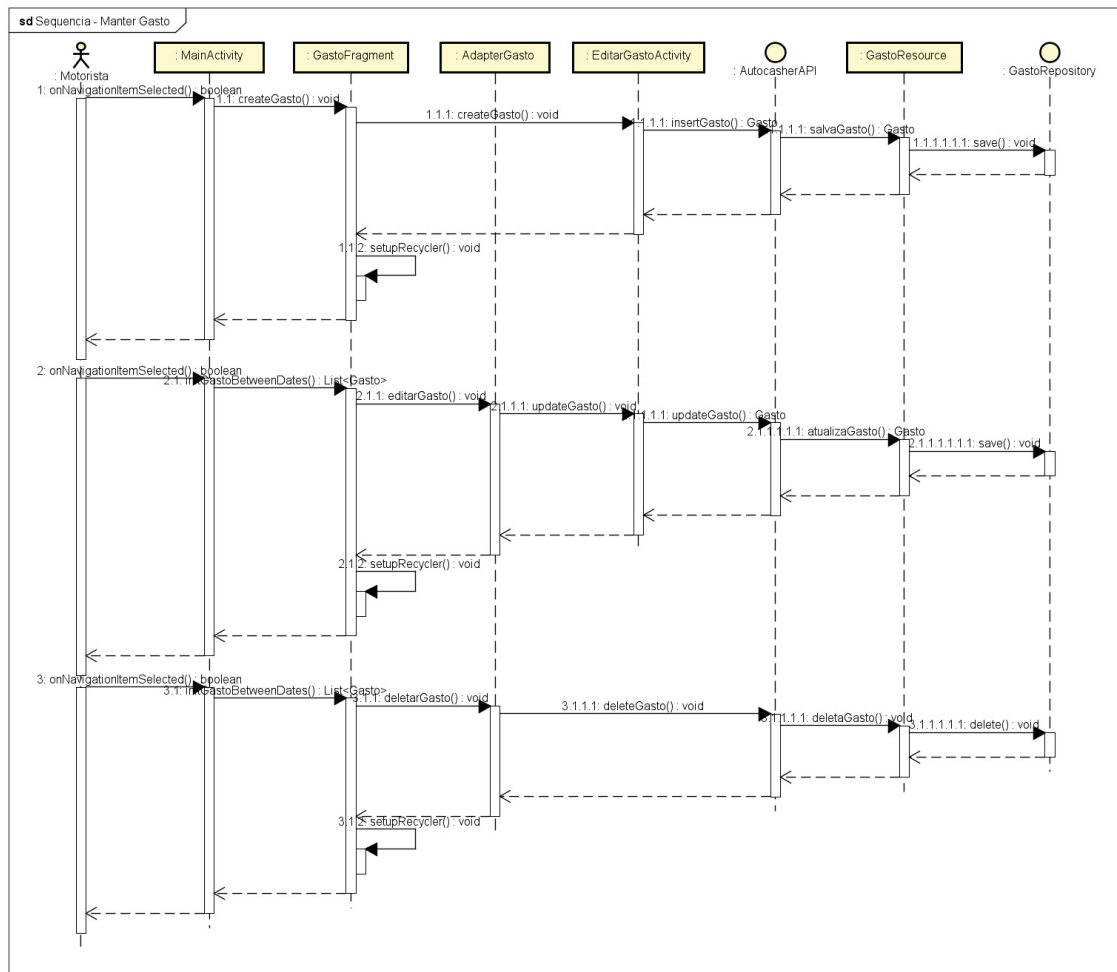
FONTE: O Autor (2021)

FIGURA 40 - DIAGRAMA DE SEQUÊNCIA VISUALIZAR ESTATÍSTICAS



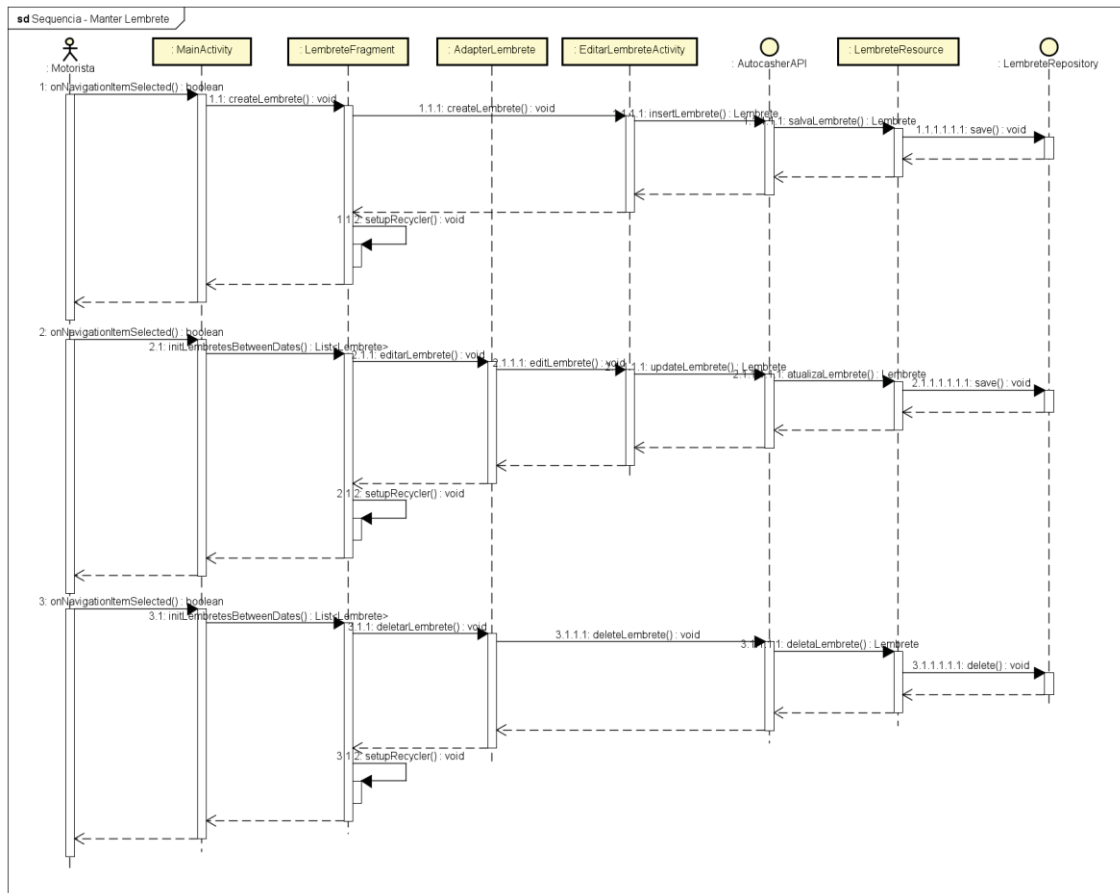
FONTE: O Autor (2021)

FIGURA 41 - DIAGRAMA DE SEQUÊNCIA MANTER GASTO



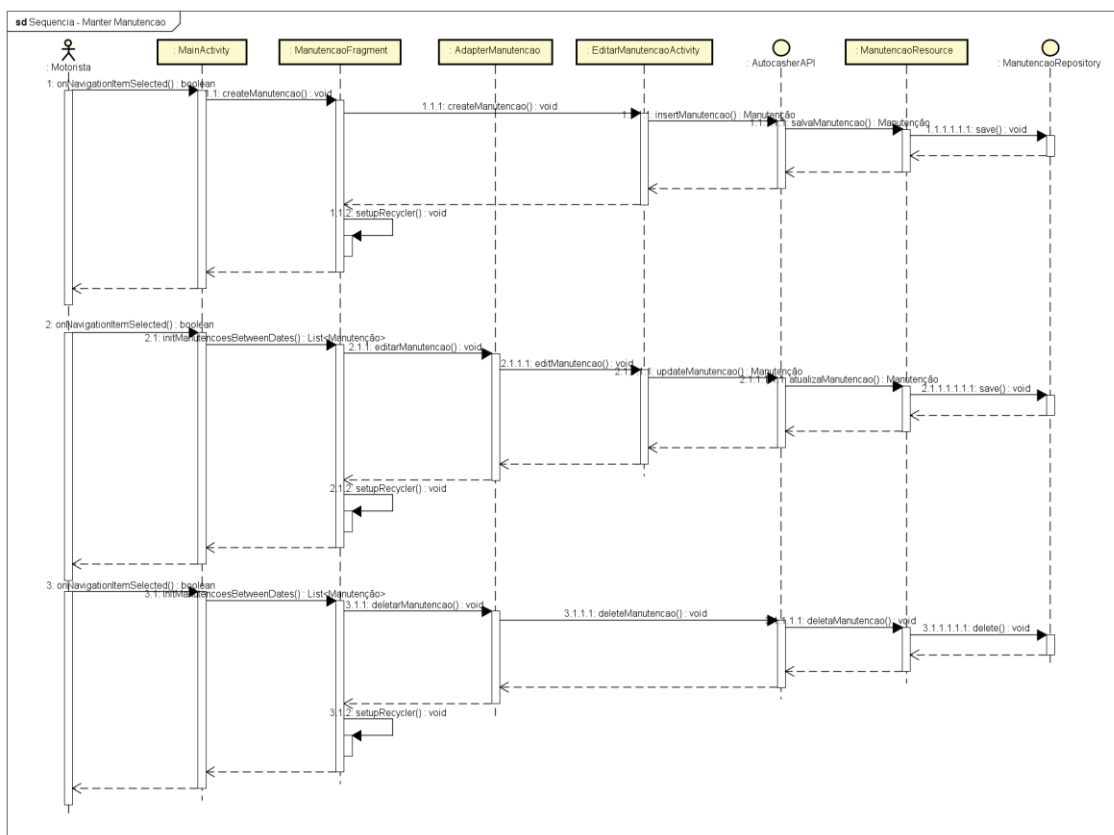
FONTE: O Autor (2021)

FIGURA 42 - DIAGRAMA DE SEQUÊNCIA MANTER LEMBRETE



FONTE: O Autor (2021)

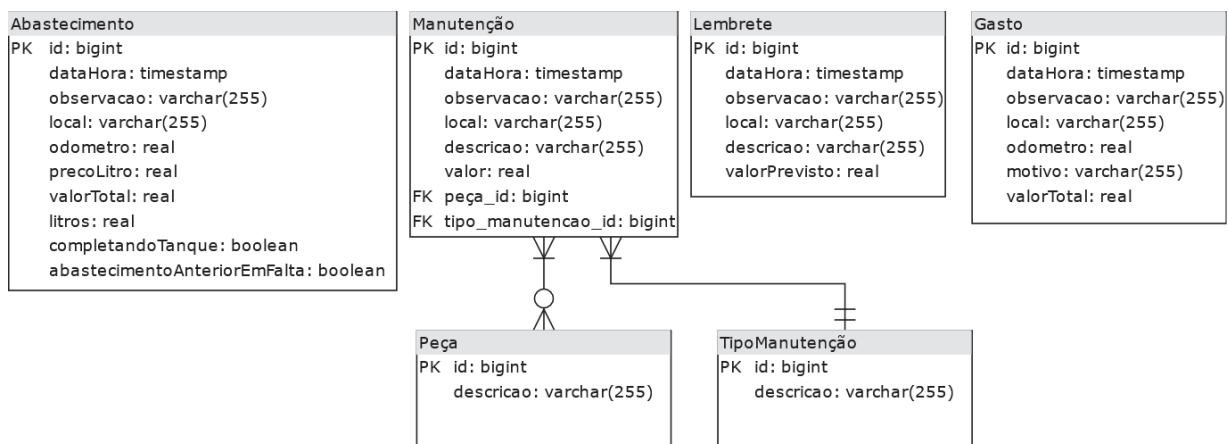
FIGURA 43 - DIAGRAMA DE SEQUÊNCIA MANTER MANUTENÇÃO



FONTE: O Autor (2021)

APÊNDICE F – MODELO LÓGICO DE DADOS

FIGURA 44 - DIAGRAMA DE MODELO LÓGICO DE DADOS



FONTE: O Autor (2021)