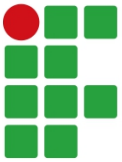


INSTITUTO FEDERAL
São Paulo
Campus Campinas

Gabriel Leme Scolari Fiorenza

DescompliCar: Estudo e Desenvolvimento de Software para auxílio no controle de gastos com manutenção de veículos

CAMPINAS
2024

Gabriel Leme Scolari Fiorenza

Estudo e Desenvolvimento de Software para auxílio no controle de manutenção de veículos

Trabalho de Conclusão de Curso apresentado como parte dos requisitos para obtenção do diploma do Curso Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia Campus Campinas.

Orientador: Prof. Me. Everton Josué Meyer da Silva.

CAMPINAS
2024

FICHA CATALOGRÁFICA

Ficha Catalográfica
Instituto Federal de São Paulo – Campus Campinas
Biblioteca "Pedro Augusto Pinheiro Fantinatti"
Tatiane Salles - CRB8/8946

Fiorenza, Gabriel Leme Scolari
F518d DescompliCar: estudo e desenvolvimento de software para auxílio no controle de gastos com manutenção de veículos / Gabriel Leme Scolari Fiorenza. – Campinas, SP: [s.n.], 2024.
60 f. : il.

Orientador: Me. Everton Josué Meyer da Silva
Trabalho de Conclusão de Curso (graduação) – Instituto Federal de Educação, Ciência e Tecnologia de São Paulo Campus Campinas. Curso de Tecnologia em Análise e Desenvolvimento de Sistemas, 2024.

1. Software - Desenvolvimento. 2. Automóveis - Manutenção e reparos. 3. Orçamento pessoal. 4. Motoristas. I. Instituto Federal de Educação, Ciência e Tecnologia de São Paulo Campus Campinas, Curso de Análise e Desenvolvimento de Sistemas. II. Título.

ATA DE DEFESA

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
São Paulo

Ministério da Educação
Instituto Federal de Educação, Ciência e Tecnologia de São Paulo
Campus Campinas
FUC CURSO SUP TEC ADS

ATA N.º 33/2024 - TADS-CMP/DAE-CMP/DRG/CMP/IFSP

Ata de Defesa de Trabalho de Conclusão de Curso - TADS

Na presente data, realizou-se a sessão pública de defesa do Trabalho de Conclusão de Curso intitulado DescompliCar: Estudo e Desenvolvimento de Software para auxílio no controle de gastos com manutenção de veículos, apresentado(a) pelo(a) aluno(a) Gabriel Leme Scolari Fiorenza (CP3018261) do **CURSO DE TECNÓLOGO EM ANÁLISE DE SISTEMAS** (campus Campinas). Os trabalhos foram iniciados às 15:00 h pelo(a) Professor(a) presidente da banca examinadora, constituída pelos seguintes membros:

Membros	Instituição	Presença (Sim/Não)
Everton Josué Meyer da Silva (Presidente/Orientador)	IFSP	sim
Carlos Eduardo Beluzo (Examinador 1)	IFSP	sim
Fábio Feliciano de Oliveira (Examinador 2)	IFSP	sim

Observações:

A banca examinadora, tendo terminado a apresentação do conteúdo da monografia, passou à arguição do(a) candidato(a). Em seguida, os examinadores reuniram-se para avaliação e deram o parecer final sobre o trabalho apresentado pelo(a) aluno(a), tendo sido atribuído o seguinte resultado:

[X] Aprovado(a) [ ] Reprovado(a)

Proclamados os resultados pelo presidente da banca examinadora, foram encerrados os trabalhos e, para constar, eu lavrei a presente ata que assino em nome dos demais membros da banca examinadora.

IFSP Campus Campinas - 12/12/2024

Documento assinado eletronicamente por:

- **Everton Josue Meyer da Silva**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 13/12/2024 08:44:49.
- **Fabio Feliciano de Oliveira**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 13/12/2024 19:06:47.
- **Carlos Eduardo Beluzo**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 17/12/2024 08:25:31.

Este documento foi emitido pelo SUAP em 12/12/2024. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse https://suap.ifsp.edu.br/autenticar-documento/ e forneça os dados abaixo:

Código Verificador: 862045
Código de Autenticação: d4afc60850

AGRADECIMENTOS

Expresso minha gratidão a todos os professores e colaboradores do IFSP - Câmpus Campinas, cuja dedicação e apoio foram fundamentais para minha formação profissional e pessoal. Em particular, agradeço ao meu orientador, Prof. Mr. Everton Josué da Silva, pela confiança depositada, pela liberdade concedida e pela valiosa parceria na elaboração desta monografia.

Aos meus pais, Ana Karina e Ricardo e meu irmão Guilherme, que me apoiaram incondicionalmente durante meus estudos. A Ana Julia por estar comigo me incentivando em todos os momentos.

RESUMO

Este trabalho apresenta o desenvolvimento de um software denominado DescompliCar, projetado para auxiliar proprietários de veículos na gestão financeira e manutenção preventiva. O sistema oferece funcionalidades como registro e monitoramento de custos, categorização de despesas, notificações automáticas, geração de relatórios e visualização do histórico de manutenções. O desenvolvimento seguiu o método empírico e a metodologia ágil Scrum, priorizando usabilidade e acessibilidade. Para o front-end, foram aplicadas tecnologias como React e Tailwind CSS, proporcionando uma interface responsiva e intuitiva. No back-end, o Firebase foi utilizado devido à sua escalabilidade, segurança e sincronização em tempo real, garantindo estabilidade e proteção aos dados dos usuários. A aplicação foi testada com usuários finais para identificar melhorias na interface e corrigir eventuais falhas, assegurando que o software atendesse às necessidades práticas e emocionais dos usuários. Com isso, o DescompliCar se destaca como uma solução eficaz para facilitar o planejamento financeiro e a organização de manutenções, contribuindo para a redução de custos e promovendo maior segurança no uso de veículos.

Palavras-chave: manutenção preventiva; gestão financeira; usabilidade; Firebase; desenvolvimento web.

ABSTRACT

This paper presents the development of a software program called DescompliCar, designed to help vehicle owners with financial management and preventive maintenance. The system offers features such as recording and monitoring costs, categorizing expenses, automatic notifications, generating reports and viewing maintenance history. Development followed the empirical method and the agile Scrum methodology, prioritizing usability and accessibility. For the front-end, technologies such as React and Tailwind CSS were applied, providing a responsive and intuitive interface. For the back-end, Firebase was used due to its scalability, security and real-time synchronization, ensuring stability and protection for user data. The application was tested with end users to identify improvements to the interface and correct any flaws, ensuring that the software met users' practical and emotional needs. As a result, DescompliCar stands out as an effective solution for facilitating financial planning and the organization of maintenance, contributing to cost reduction and promoting greater safety in the use of vehicles.

Keywords: preventive maintenance; financial management; usability; Firebase; web development.

LISTA DE FIGURAS

SUMÁRIO

1 INTRODUÇÃO

No mundo moderno, a necessidade de acesso imediato a informações tornou-se essencial, impulsionada pelo avanço tecnológico e pela conectividade constante proporcionada por dispositivos móveis. Dados atualizados em tempo real são cruciais para decisões mais assertivas, seja em transações financeiras, planejamento pessoal ou gestão de bens. No contexto automotivo, isso é particularmente importante, uma vez que falhas no controle de manutenções ou esquecimentos podem resultar em altos custos financeiros, riscos de segurança e perda de eficiência.

Nesse cenário, o desenvolvimento de um software web que centralize e otimize o gerenciamento de gastos e manutenções de veículos apresenta uma solução prática e eficaz. O sistema proposto permite aos usuários registrar e acompanhar despesas de manutenção, receber notificações automáticas de revisões necessárias e acessar relatórios detalhados para análise financeira. Com isso, o objetivo principal é simplificar a organização, reduzir custos inesperados e aumentar a confiabilidade no planejamento de cuidados com o veículo.

Entre os benefícios gerados estão a automação de processos, como lembretes de renovação de seguros e impostos, e a categorização de despesas, que permite uma visão clara e estruturada dos gastos. Esses recursos não apenas facilitam o controle financeiro, mas também promovem maior segurança e eficiência no uso dos veículos, fornecendo insights que ajudam os usuários a tomarem decisões informadas e a longo prazo.

Para assegurar a eficiência do sistema, foi adotada uma abordagem que combina testes unitários, realizados pelo próprio desenvolvedor, e de usabilidade com testes de usuários supervisionados. Isso garante um software funcional, seguro e alinhado às expectativas do público-alvo. O foco na experiência do usuário busca oferecer uma interface intuitiva, com respostas rápidas e informações claras, atendendo tanto às necessidades técnicas quanto ao conforto do usuário final.

Dessa forma, o estudo explora não apenas as técnicas de desenvolvimento de software, mas também a compreensão das necessidades específicas dos proprietários de veículos. O resultado foi um sistema robusto e acessível, que une tecnologia e praticidade para proporcionar uma gestão automotiva eficiente e confiável.

2 JUSTIFICATIVA

A gestão de veículos é uma necessidade crescente para proprietários, que buscam soluções práticas e acessíveis para organizar informações relacionadas à manutenção, custos e documentos.

Contudo, a análise de ferramentas disponíveis revela lacunas significativas. Conforme identificado em estudos apresentados nas figuras 1 e 2, os softwares existentes atendem predominantemente concessionárias, focando em dados relacionados à venda de veículos, ou apresentam limitações que comprometem sua usabilidade.

Entre essas limitações, destacam-se a indisponibilidade em múltiplos dispositivos, versões gratuitas restritivas, e a presença de anúncios excessivos, que poluem o design e dificultam o uso.

Figura 1 - Tabela comparativa de Softwares parte 1

SOFTWARES	Suporte Web	Login	Cadastro	Suporte Smatphone	Gestão Gastos Gerais	Gestão Gastos Individual	Mais de 1 veiculo
Descomplicar	☑	☑	☑	☑	☑	☑	☑
Checar	☐	☐	☐	☐	☐	☐	☐
Drivvo	☑	☑	☑	☑	☑	☑	☑
Motolog	☐	☑	☑	☑	☑	☑	☑
Car Expenses	☐	☐	☐	☑	☑	☑	☐
Fuelio	☐	☐	☐	☑	☑	☑	☑
Niun Auto	☐	☑	☑	☑	☑	☑	☑
Dahuj	☑	☐	☐	☑	☐	☐	☐
Caoa	☑	☐	☐	☑	☐	☐	☐
Sim Veiculos	☑	☐	☐	☑	☐	☐	☐

Fonte: Elaborado pelo autor (2024)

Figura 2 - Tabela comparativa de Softwares parte 2

SOFTWARES	Facilidade de uso	Intuitividade	Claresa de Uso	Gratuidade Parcial	Gratuidade Plena	Apresenta propagandas	Tem acesso a suporte
Descomplicar	☑	☑	☑	☑	☑	☐	☐
Checar	☐	☐	☐	☐	☐	☐	☐
Drivvo	☑	☑	☑	☑	☐	☑	☑
Motolog	☑	☑	☑	☑	☐	☑	☑
Car Expenses	☑	☐	☑	☐	☑	☑	☑
Fuelio	☑	☑	☑	☑	☐	☑	☑
Niun Auto	☑	☑	☑	☑	☐	☐	☑
Dahuj	☑	☑	☑	☐	☑	☐	☑
Caoa	☑	☑	☑	☐	☑	☐	☑
Sim Veiculos	☑	☑	☑	☐	☑	☐	☑

Fonte: Elaborado pelo autor (2024)

Este trabalho propôs o desenvolvimento de um software direcionado especificamente aos proprietários de veículos, oferecendo uma solução que prioriza

a simplicidade, acessibilidade e usabilidade. Ao contrário das ferramentas tradicionais, que se concentram em funcionalidades específicas, este software busca entregar uma experiência centrada no usuário final, facilitando a gestão de manutenções, custos e documentos veiculares.

A proposta diferencia-se por sua interface amigável, desenvolvida com princípios de design UX/UI para atender usuários de diferentes níveis de familiaridade tecnológica. Com uma navegação intuitiva, o software reduz a curva de aprendizado, permitindo que informações como a data da última revisão ou o vencimento de documentos sejam acessadas de forma rápida e clara. Além disso, notificações personalizadas serão integradas para alertar os usuários sobre eventos importantes, como prazos de manutenção e renovação de documentos, evitando atrasos e imprevistos.

O impacto prático da solução também se mostra significativo ao centralizar dados em uma plataforma acessível em qualquer dispositivo, permitindo aos usuários organizar custos futuros, planejar financeiramente e evitar surpresas. Isso resulta em uma gestão mais eficiente, economia de tempo, e redução de despesas imprevistas.

O desenvolvimento do projeto seguiu a metodologia ágil Scrum, com sprints e revisões regulares que possibilitaram aprimoramentos contínuos tanto na funcionalidade quanto no layout. Dessa forma, o software permaneceu simples de usar, sem abrir mão de recursos essenciais.

3 OBJETIVOS

3.1 Objetivo Geral

Desenvolver um software multiplataforma que auxilie proprietários de veículos na gestão financeira e na organização de manutenções preventivas, promovendo planejamento, economia e segurança.

3.2 Objetivos Específicos

- Implementar funcionalidades para registro e monitoramento de custos veiculares, incluindo gráficos e relatórios personalizados que auxiliem no planejamento financeiro.
- Desenvolver um sistema de categorização de despesas (manutenção, combustível, seguros, impostos) e visualização de padrões de gastos por categoria e período.
- Criar um módulo para registro e consulta de histórico de manutenções, abrangendo datas, serviços realizados e custos associados.
- Desenvolver notificações automáticas para lembretes importantes, como revisões, pagamento de impostos e renovação de seguros.
- Garantir a integração com uma base de dados segura e escalável, como o Firebase, para armazenar e proteger as informações do usuário.
- Criar relatórios de gastos, permitindo a análise de custos dos veículos em períodos selecionados.
- Desenvolver um dashboard intuitivo, que centralize informações relevantes, como manutenções pendentes, despesas recentes e lembretes.

4 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os conceitos e teorias que embasam o desenvolvimento do DescompliCar. Inicialmente, aborda-se a importância da manutenção preventiva de veículos, destacando sua relevância para a segurança e economia.

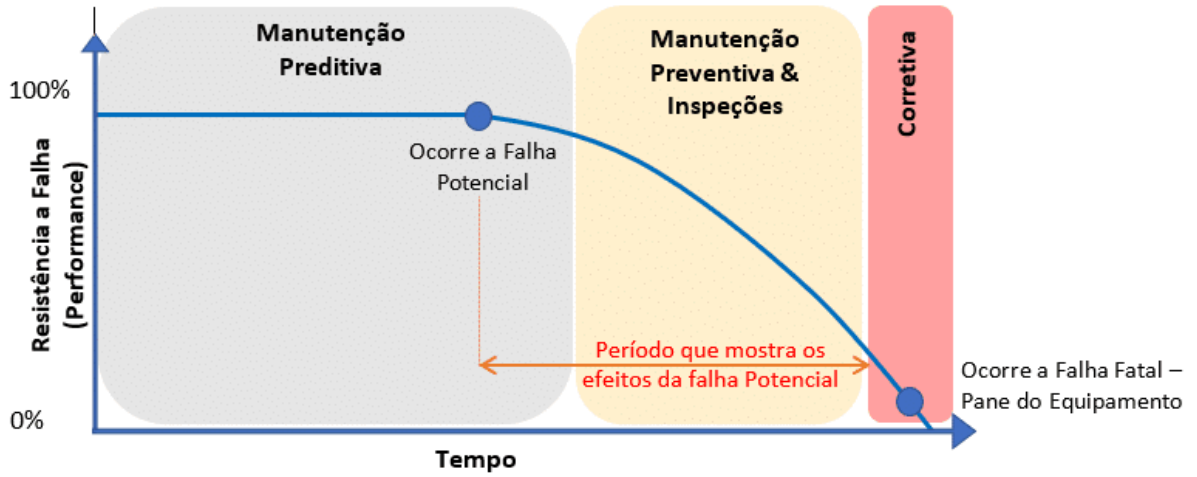
4.1 Manutenção Preventiva de Veículos

A manutenção preventiva de veículos é uma prática essencial que vai além da simples prevenção de acidentes. Ela desempenha um papel fundamental na redução de custos futuros, garantindo maior segurança e eficiência operacional. De acordo com Silva et al. (2023), a falta de manutenção adequada pode comprometer significativamente a segurança dos ocupantes do veículo, além de gerar despesas excessivas com reparos corretivos, frequentemente mais dispendiosos do que as ações preventivas.

Essa abordagem consiste em um conjunto de ações realizadas de forma regular e planejada para manter o veículo em condições ideais de funcionamento. Essas práticas incluem inspeções periódicas, substituição de peças desgastadas e verificações de sistemas críticos, como freios, suspensão e pneus. Estudos mostram que uma manutenção preventiva adequada pode reduzir consideravelmente os riscos de falhas mecânicas inesperadas, que muitas vezes resultam em acidentes ou despesas imprevistas.

Além disso, a manutenção preventiva contribui para a longevidade do veículo, permitindo que ele opere de forma eficiente por mais tempo. Segundo dados levantados pela Abecom, sintetizados na Figura 3, equipamentos submetidos a manutenções regulares apresentam uma redução significativa em falhas mecânicas graves e em custos acumulados com reparos ao longo de sua vida útil.

Figura 3 - Relação existente entre a curva PF e a manutenção preventiva



Fonte: Abecon (2021)

Essa prática também beneficia a sociedade como um todo, pois veículos em boas condições geram menor impacto ambiental, com emissões reduzidas, e promovem maior segurança no trânsito, diminuindo o risco de acidentes causados por falhas técnicas. Ademais, a utilização de peças e serviços de qualidade reforça a eficácia dessas ações, garantindo resultados duradouros.

Portanto, a manutenção preventiva deve ser encarada não como um custo adicional, mas como um investimento estratégico. Essa visão não apenas preserva o patrimônio do proprietário, mas também contribui para um trânsito mais seguro e sustentável. A conscientização sobre sua importância e a incorporação de ferramentas que auxiliem no planejamento e execução dessas ações são fundamentais para garantir seus benefícios a longo prazo.

4.2 Necessidades dos Usuários na Manutenção de Veículos

Manter um veículo em boas condições apresenta desafios significativos para muitos proprietários. Entre as principais dificuldades estão a falta de organização no acompanhamento das manutenções e o desconhecimento técnico sobre os períodos adequados para intervenções preventivas. Esses fatores não apenas elevam os custos de manutenção corretiva, como também podem comprometer a segurança do veículo e de seus ocupantes.

Outro problema recorrente é a vulnerabilidade dos usuários à desinformação ou mesmo a fraudes em serviços mecânicos. Muitos proprietários, devido à falta de conhecimento técnico, não conseguem identificar quando uma peça realmente precisa ser substituída ou se estão sendo induzidos a realizar trocas desnecessárias, resultando em gastos excessivos.

De acordo com dados da Polícia Federal, entre janeiro e abril de 2023, foram registradas mais de 4.400 infrações relacionadas a condutores que dirigiam veículos em mau estado de conservação. Além disso, estima-se que veículos sem manutenção adequada sejam responsáveis por cerca de 30% dos acidentes nas estradas, evidenciando a gravidade do problema. Esse cenário reforça a necessidade de ferramentas que ajudem os usuários a gerenciar informações sobre a manutenção de forma clara e confiável.

Nesse contexto, um software que centralize o histórico de manutenções e ofereça previsões baseadas em registros anteriores pode atender de maneira eficaz às necessidades desses usuários. Ele não apenas ajudaria a organizar e planejar manutenções futuras, mas também proporcionaria maior transparência, permitindo que os proprietários tomem decisões mais informadas, reduzindo o risco de imprevistos mecânicos e custos desnecessários.

Além disso, ao oferecer relatórios personalizados e alertas automatizados para manutenções preventivas, o sistema contribui para a economia de tempo e recursos, bem como para a segurança do veículo. Assim, atende tanto à necessidade de organização quanto à de confiabilidade nas decisões de manutenção.

4.3 Usabilidade em Software para Usuários Finais

Segundo Soares (2004, p. 34), “Uma das características que distingue um software com qualidade, em termos de usabilidade, é a sua adequação à funcionalidade do usuário, sem exigir para o seu uso que o usuário tenha que se adaptar a ele.” Partindo desse ponto de vista, o desenvolvimento do software proposto será conduzido com ênfase nas necessidades dos usuários, sempre considerando o *feedback* obtido durante as etapas de testes. A usabilidade, de acordo com a norma ISO 9241-11, pode ser definida como a "capacidade de um sistema ser usado por determinados usuários para alcançar objetivos específicos com eficácia, eficiência e satisfação em um contexto de uso especificado". Essa definição destaca a importância de projetar sistemas que facilitem a interação, proporcionando uma experiência fluida e intuitiva.

Soares (2004) enfatiza que a qualidade de um software está diretamente ligada à sua capacidade de se ajustar às necessidades dos usuários, promovendo autonomia e reduzindo barreiras. Para isso, é essencial priorizar o design centrado no usuário, analisando os comportamentos e fluxos de trabalho do público-alvo. Por exemplo, em um software para controle financeiro, recursos como categorização automática de despesas ou geração de relatórios claros são projetados para atender às práticas comuns dos usuários, aumentando a eficiência do sistema sem exigir adaptação significativa. Nesse sentido, Aragão (2001) reforça que “Um sistema realmente efetivo é aquele que é projetado a partir do ponto de vista do operador e não da perspectiva de uma simbiose operador/máquina.” Essa perspectiva direciona o desenvolvimento para a criação de interfaces que complementem o fluxo natural de trabalho do usuário, eliminando complexidades desnecessárias.

A coleta de *feedback* em cada etapa do desenvolvimento é uma prática essencial para garantir a evolução do software em conformidade com as expectativas dos usuários. Estudos conduzidos por Nielsen (1994) mostram que até cinco usuários são suficientes para identificar a maior parte dos problemas de usabilidade em um sistema. Aplicar esse tipo de metodologia em testes iterativos permite ajustes rápidos e eficazes no design. Por exemplo, sistemas de comércio eletrônico frequentemente utilizam barras de progresso em formulários longos, uma funcionalidade que surgiu da observação de que usuários abandonavam o preenchimento por falta de informações sobre o tempo necessário para concluir o

processo.

Outro aspecto fundamental para a usabilidade é a acessibilidade. A Web Content Accessibility Guidelines (WCAG) sugere práticas que tornam os sistemas utilizáveis por pessoas com deficiência, incluindo o uso de leitores de tela, teclas de atalho e contrastes de cores adequados. Essas diretrizes destacam como o design inclusivo amplia a base de usuários e melhora a experiência geral. Um exemplo bem-sucedido é o Google Maps, que implementou descrições por áudio para usuários com deficiência visual, tornando o serviço mais acessível e reforçando seu compromisso com a inclusão.

O design centrado no usuário também exige um foco constante na simplicidade e clareza. Conforme defendido por Soares (2004) e Aragão (2001); um sistema eficaz deve minimizar a necessidade de aprendizado, oferecendo uma interface que guie o usuário naturalmente. Para alcançar esse objetivo, práticas como prototipagem rápida e realização de testes de usabilidade em diferentes fases do desenvolvimento permitem que o software evolua continuamente. Durante essas avaliações, dificuldades como confusão com termos, frustrações com o fluxo de navegação ou problemas de interação tornam-se oportunidades para refinar o design. Além disso, a aplicação de técnicas de design visual, como feedback imediato e disposição lógica dos elementos, fortalece a confiança do usuário e reduz a ocorrência de erros.

A interação constante com os usuários durante o desenvolvimento não apenas melhora a funcionalidade do sistema, mas também promove um vínculo de confiança entre o público-alvo e a equipe de desenvolvimento. Quando os usuários percebem que suas necessidades são valorizadas e atendidas, a satisfação com o software aumenta, e eles se tornam colaboradores ativos no processo de aprimoramento. Isso reflete a visão de que a tecnologia deve se adaptar às pessoas, e não o contrário, criando sistemas que são ao mesmo tempo eficientes e agradáveis de usar.

Por fim, ao alinhar os princípios de usabilidade de Soares (2004), Aragão (2001) e Moraes e Mont'Alvão (2000); torna-se evidente que um software eficaz é aquele que respeita as jornadas e os contextos de uso do usuário, proporcionando uma experiência sem obstáculos e intuitiva. A aplicação de práticas como testes iterativos, design centrado no usuário e inclusão de recursos acessíveis garante que o software atenda tanto às necessidades funcionais quanto às expectativas

emocionais do público-alvo. Com uma abordagem contínua de aprimoramento, é possível entregar um sistema que, além de atender aos requisitos técnicos, promove satisfação, eficiência e facilidade de uso, assegurando sua relevância e utilidade a longo prazo.

4.4 Metodologia Scrum e Desenvolvimento Ágil

A metodologia Scrum é amplamente reconhecida e utilizada no desenvolvimento de software como parte das práticas de desenvolvimento ágil. Conforme Sutherland (2014), o Scrum organiza equipes em ciclos iterativos e incrementais chamados sprints, com duração fixa, geralmente de duas a quatro semanas. Cada sprint tem como objetivo a entrega de funcionalidades utilizáveis do sistema, promovendo uma evolução constante do produto e permitindo ajustes frequentes com base no *feedback* dos stakeholders.

Uma característica essencial do Scrum é sua flexibilidade para lidar com mudanças nos requisitos ao longo do projeto. Essa adaptabilidade, segundo Sutherland (2014), possibilita que equipes respondam rapidamente a alterações, maximizando a entrega de valor. Essa abordagem o diferencia de metodologias tradicionais, como o modelo Waterfall, que segue um fluxo linear e rígido, dificultando adaptações durante o ciclo de desenvolvimento. Além disso, reuniões regulares, como o planejamento de sprint, os encontros diários (daily scrums) e as retrospectivas, garantem o alinhamento contínuo da equipe com os objetivos definidos e promovem um ciclo de melhoria contínua.

Entre os benefícios do Scrum, destacam-se sua capacidade de promover iterações rápidas, foco na entrega incremental de valor e a colaboração contínua entre os membros da equipe e os stakeholders. Sutherland (2014) ressalta que o Scrum não apenas aumenta a produtividade, mas também reduz significativamente os riscos ao permitir ajustes constantes no planejamento e na execução das tarefas. Isso melhora a comunicação e facilita a identificação precoce de problemas, aumentando a eficiência em projetos complexos. Ao contrário do modelo Waterfall, que segue um fluxo sequencial rígido, o Scrum permite revisão e melhorias constantes durante o ciclo de desenvolvimento, tornando-se especialmente eficaz em projetos sujeitos a mudanças frequentes nos requisitos ou incertezas quanto ao escopo inicial.

A aplicação prática do Scrum foi essencial para o sucesso do desenvolvimento deste projeto. A metodologia foi utilizada como base para o gerenciamento do trabalho, promovendo entregas contínuas e alinhadas aos requisitos estabelecidos. O projeto foi dividido em sprints de duas semanas, com objetivos claros e mensuráveis definidos a partir de um backlog do produto, que priorizava os requisitos funcionais e não funcionais do sistema. Durante o planejamento de cada sprint, foram escolhidas as tarefas a serem realizadas, focando sempre no desenvolvimento de funcionalidades específicas.

Ao final de cada sprint, foi realizada uma revisão, onde as funcionalidades desenvolvidas foram testadas, simulando uma entrega para o cliente, validando as funcionalidades e identificando pontos onde cabiam melhorias, cujo as quais entravam para a próxima sprint para serem implementadas.

Com essa abordagem, o projeto foi conduzido de forma organizada com entregas incrementais que proporcionaram maior controle sobre o escopo e o cronograma. A aplicação do Scrum foi fundamental para lidar com mudanças nos requisitos e garantir que o sistema desenvolvido atendesse às expectativas e objetivos definidos. Essa metodologia possibilitou a entrega de um produto final alinhado com as necessidades dos usuários e com as diretrizes do projeto, conforme destacado por Sutherland (2014).

4.5 Segurança e Privacidade de Dados

A segurança e a privacidade de dados são pilares fundamentais no desenvolvimento de sistemas, especialmente em um contexto de crescente digitalização e armazenamento de informações na nuvem. A proteção de dados pessoais tem ganhado atenção global devido a regulamentações como o GDPR (General Data Protection Regulation) na União Europeia e a LGPD (Lei Geral de Proteção de Dados) no Brasil, que estabelecem diretrizes rigorosas sobre o tratamento de informações pessoais. Essas regulamentações buscam assegurar que dados sensíveis sejam manipulados de forma ética, segura e transparente, prevenindo vazamentos e o uso indevido de informações confidenciais.

No contexto deste projeto, que utiliza o Firebase como plataforma para armazenamento e gerenciamento de dados, práticas robustas de segurança foram adotadas para proteger as informações dos usuários. O Firebase oferece uma

infraestrutura baseada em nuvem, projetada com padrões elevados de segurança. Conforme descrito em sua documentação oficial, o Firebase utiliza criptografia em trânsito (SSL/TLS) e em repouso para proteger os dados, além de controles de acesso baseados em regras configuráveis pelo desenvolvedor. Essas regras permitem definir permissões específicas para leitura e escrita, de acordo com os perfis dos usuários e os níveis de acesso necessários para o sistema.

A aplicação de boas práticas de segurança inclui a autenticação forte dos usuários, como o uso do Firebase Authentication, que suporta múltiplos métodos de login, incluindo autenticação multifator (MFA). Essas medidas aumentam a proteção contra acessos não autorizados e garantem maior confiabilidade na identificação dos usuários. Adicionalmente, políticas de retenção de dados foram configuradas para minimizar o armazenamento desnecessário, em conformidade com os princípios do GDPR e da LGPD.

As consequências de falhas na segurança de dados podem ser devastadoras, tanto para os usuários quanto para os responsáveis pelo sistema. Vazamentos de informações sensíveis podem comprometer a privacidade, gerar prejuízos financeiros e danos à reputação das empresas. Além disso, incidentes dessa natureza podem acarretar sanções legais e multas, conforme estipulado pelas legislações vigentes. Portanto, a implementação de um sistema seguro e confiável é indispensável para garantir a integridade das informações e a confiança dos usuários.

Em suma, a segurança e a privacidade de dados foram priorizadas durante o desenvolvimento deste projeto, com a aplicação de tecnologias modernas e práticas recomendadas pelo Firebase. O compromisso com a proteção das informações reflete a importância de alinhar os objetivos do sistema às regulamentações de proteção de dados, assegurando que os usuários possam utilizar o sistema de forma segura e confiável.

5 METODOLOGIA

Este capítulo detalha as etapas práticas envolvidas no desenvolvimento do *DescompliCar*, software voltado para a gestão financeira e manutenção de veículos. A metodologia aplicada combina princípios do desenvolvimento ágil com práticas de engenharia de software, com o objetivo de garantir um processo iterativo e eficiente. A escolha de metodologias e ferramentas foi orientada pelas necessidades específicas do projeto, priorizando acessibilidade, usabilidade e escalabilidade.

A metodologia adotada permitiu dividir o desenvolvimento em fases bem definidas, abrangendo desde o levantamento de requisitos até a implementação e testes finais. Para organizar e estruturar o progresso, foi utilizada uma abordagem inspirada no Scrum, com ciclos curtos de desenvolvimento e validação contínua por meio de feedbacks de usuários. Cada etapa foi planejada para assegurar que as funcionalidades implementadas atendessem aos requisitos técnicos e práticos dos usuários, refletindo uma abordagem centrada no usuário.

No contexto do *DescompliCar*, a aplicação da metodologia não apenas garantiu o cumprimento dos objetivos funcionais, mas também assegurou a qualidade do sistema em aspectos como responsividade, integração com banco de dados em tempo real e entrega de uma interface intuitiva. Dessa forma, este capítulo explora como as escolhas metodológicas contribuíram para a construção de um software robusto, acessível e alinhado às demandas dos usuários finais.

5.1 Metodologia Ágil: Aplicação do Scrum

A abordagem ágil Scrum foi escolhida para guiar o desenvolvimento do DescompliCar devido à sua flexibilidade e foco em entregas incrementais. Essa metodologia permitiu a divisão do projeto em etapas gerenciáveis, denominadas sprints, facilitando o acompanhamento do progresso e a adaptação às necessidades identificadas ao longo do processo.

5.1.1 Sprints e Entregas Incrementais

O projeto foi estruturado em sprints, com o objetivo de desenvolver funcionalidades específicas, como, por exemplo, a tela de login. No início de cada sprint, foi realizado um planejamento detalhado sobre os requisitos que cada funcionalidade deveria atender, definindo-se os passos necessários para sua implementação. Nas sprints iniciais, o foco principal foi o design da interface, sem a preocupação de adicionar funcionalidades. Posteriormente, as sprints passaram a priorizar a implementação das funcionalidades principais, como o sistema de cadastro e login de usuários, culminando na integração com o Firebase.

Para o desenvolvimento deste projeto, utilizando os princípios da metodologia Scrum, as sprints foram organizadas da seguinte forma:

- **MVP (Produto Mínimo Viável):** Nesta sprint, foram desenvolvidos os conceitos das principais telas das histórias levantadas (Dashboard, Veículos, Manutenções, Despesas, Relatórios). Para permitir testes iniciais, foram utilizados dados mockados, o que possibilitou identificar possíveis melhorias a serem realizadas após os testes.
- **Telas Externas:** Durante esta sprint, foram desenvolvidas as telas de login e de cadastro utilizando dados mockados. Também foi criada a Home da aplicação, permitindo finalizar um MVP com dados mockados para a realização da primeira etapa de testes.
- **Integração de Cadastro com o Firebase:** Nesta sprint, foi realizada a integração das funcionalidades de login e cadastro com o Firebase Authentication, permitindo o armazenamento dos cadastros e a realização do login, incluindo a opção de autenticação com contas do Google.
- **Integração de Dados do Usuário com o Firebase:** Nessa etapa, a tela de dados do usuário foi integrada ao Firestore Database, e foram definidos os dados do usuário a serem tratados e armazenados.
- **Integração dos Dados dos Veículos com o Firebase:** Durante esta sprint, foram definidos os dados veiculares a serem armazenados e realizada sua integração com o Firestore Database.

- **Integração dos Dados de Manutenção com o Firebase:** Foram definidos os dados relacionados às manutenções e realizada a integração desses dados com o Firestore Database.
- **Integração dos Dados de Despesas com o Firebase:** Nesta etapa, foram definidos e integrados os dados de despesas ao Firestore Database, consolidando o armazenamento dessas informações.
- **Integração da Tela de Relatórios com o Firebase:** Durante esta sprint, os dados mockados exibidos nos relatórios foram substituídos pelos dados reais cadastrados no Firestore Database.
- **Integração da Tela de Dashboard:** Nesta etapa, os dados mockados da tela de Dashboard foram substituídos pelos dados reais registrados no Firestore Database.
- **Correções:** Foi dedicada uma sprint para correção de bugs encontrados durante os testes, além da implementação de melhorias textuais visando facilitar a interpretação pelos usuários.
- **Implantação de Responsividade:** Durante esta sprint, foi implementada e refinada a responsividade do software, permitindo seu uso tanto em computadores quanto em smartphones, por meio do navegador.
- **Hospedagem:** Por fim, foi realizada a hospedagem do software utilizando o Firebase Hosting, disponibilizando o sistema para acesso público, concluindo as etapas de desenvolvimento do projeto.

5.1.2 Entregas das Sprints

Durante o processo de desenvolvimento deste projeto, não foi possível contar com um cliente específico, como ocorre em práticas tradicionais do Scrum. A ausência de um cliente real inviabilizou a entrega formal e frequente dos incrementos do produto ao cliente para validação. Para contornar essa limitação, optou-se por substituir as entregas reais por simulações do funcionamento do software, utilizando o Vite como ferramenta de desenvolvimento e execução. Por meio dessa abordagem, cada incremento do software foi analisado e testado, permitindo uma avaliação criteriosa das funcionalidades desenvolvidas em cada sprint.

Esse processo de simulação consistiu no acesso ao sistema em execução, avaliando tanto o comportamento das funcionalidades implementadas quanto a usabilidade das interfaces criadas. Durante as análises, foi possível identificar e corrigir eventuais bugs, que poderiam comprometer o desempenho e a experiência do usuário. Além disso, a avaliação permitiu a identificação de oportunidades de melhoria nas funções existentes, incluindo ajustes textuais que pudessem tornar o sistema mais intuitivo e acessível.

Essa abordagem foi essencial para manter o foco na qualidade e na adequação do sistema às necessidades dos usuários finais. Mesmo sem a presença de um cliente real para *feedback* direto, a simulação e os testes sistemáticos possibilitaram um ciclo contínuo de ajustes e refinamentos, contribuindo para que o produto final fosse mais robusto, funcional e alinhado às expectativas de um público-alvo genérico, mas exigente em termos de eficiência e facilidade de uso. Assim, as entregas das sprints não apenas cumpriram seu papel como etapas de desenvolvimento, mas também consolidaram um processo iterativo de melhoria contínua.

5.2 Ferramentas e Tecnologias Utilizadas

O desenvolvimento do *DescompliCar* foi realizado com o emprego de tecnologias modernas, selecionadas de forma criteriosa para atender às necessidades específicas do projeto. Essas ferramentas foram escolhidas com o objetivo de viabilizar uma solução eficiente, responsiva e escalável, alinhada aos requisitos do sistema. A seguir, são apresentadas as principais tecnologias utilizadas e suas respectivas aplicações.

5.2.1 React: Implementação da Interface do Usuário

O React foi escolhido como biblioteca principal para a construção da interface do usuário devido à sua capacidade de criar componentes reutilizáveis e interfaces dinâmicas. Sua abordagem baseada em componentes possibilitou dividir o sistema em partes menores e independentes, como o dashboard, o módulo de relatórios e o formulário de cadastro. Isso não apenas acelerou o desenvolvimento, mas também

facilitou a manutenção do código.

A reatividade do React permitiu que as atualizações na interface ocorressem de maneira fluida, garantindo uma experiência de usuário intuitiva e responsiva. A biblioteca também foi integrada com outras ferramentas, como o React Router DOM, para gerenciar a navegação entre páginas, proporcionando uma experiência contínua e sem interrupções.

O React foi selecionado por sua ampla adoção na indústria, documentação robusta e suporte à criação de interfaces complexas de forma eficiente. A possibilidade de integração com outras bibliotecas e a facilidade em implementar uma interface responsiva foram fatores determinantes para sua utilização.

5.2.2 Firebase: Gerenciamento do Banco de Dados em Tempo Real

O Firebase foi utilizado como solução de back-end, fornecendo um banco de dados em tempo real para armazenar informações como manutenções, despesas e relatórios dos usuários. A sincronização automática dos dados permitiu que as atualizações fossem refletidas instantaneamente para todos os dispositivos conectados, garantindo uma experiência consistente. Além disso, o Firebase também foi utilizado para autenticação de usuários, incluindo login social com Google, o que simplificou a gestão de contas e ampliou a acessibilidade.

Outro recurso explorado foi o Firebase Hosting, que viabilizou a hospedagem do sistema, garantindo alta disponibilidade e tempos de carregamento rápidos por meio de uma rede de distribuição de conteúdo (CDN).

O Firebase foi escolhido devido à sua escalabilidade, segurança e simplicidade de integração com aplicações web. Sua capacidade de gerenciar dados em tempo real e autenticação segura permitiu concentrar esforços no desenvolvimento das funcionalidades do sistema, sem a necessidade de gerenciar servidores complexos.

5.2.3 Tailwind CSS: Criação de uma Interface Visual Consistente

Para a estilização da interface do *DescompliCar*, foi utilizado o Tailwind CSS, uma biblioteca de classes utilitárias amplamente reconhecida por sua flexibilidade e

eficiência no desenvolvimento de layouts modernos. Essa ferramenta permitiu a criação de um design visualmente consistente e responsivo, garantindo que a interface fosse atrativa e funcional em diferentes dispositivos. A abordagem baseada em utilitários do Tailwind CSS reduziu significativamente a necessidade de escrita de CSS personalizado, acelerando o processo de desenvolvimento e proporcionando maior uniformidade entre os elementos visuais do sistema.

A aplicação do Tailwind CSS também simplificou a personalização de componentes como botões, formulários e gráficos, permitindo que esses elementos fossem ajustados de forma precisa para atender às demandas específicas do projeto. Além disso, sua integração nativa com React facilitou a implementação de estilos diretamente nos componentes, melhorando a produtividade da equipe. Outro benefício notável foi a capacidade de garantir que o layout permanecesse responsivo, adaptando-se perfeitamente a diferentes tamanhos de tela, desde desktops até dispositivos móveis, e assegurando uma experiência de uso fluida e acessível para todos os usuários.

A escolha do Tailwind CSS foi motivada por sua abordagem utilitária, que reduz a necessidade de escrever CSS personalizado. Sua integração direta com o React também agilizou a aplicação de estilos, garantindo uma aparência moderna e uniforme em toda a aplicação.

5.2.4 Outras Tecnologias

- **TypeScript**: Garantiu maior confiabilidade no código ao adicionar tipagem estática, reduzindo erros durante o desenvolvimento.
- **Vite**: Facilitou o processo de build e desenvolvimento com tempos de carregamento mais rápidos.
- **React Router DOM**: Gerenciou a navegação entre páginas, proporcionando uma experiência contínua ao usuário.
- **Tanstack React Query**: Otimizou a sincronização de dados e o cache, melhorando o desempenho do sistema.
- **Lucide React**: Forneceu uma biblioteca de ícones intuitivos para enriquecer a interface.
- **Zod**: Validou entradas do usuário, garantindo a consistência e a segurança dos dados enviados ao sistema.

5.2.5 Justificativa das Escolhas

As ferramentas selecionadas para o desenvolvimento do *DescompliCar* foram cuidadosamente integradas para complementar o ecossistema React, resultando em um processo de desenvolvimento mais ágil e eficiente. Cada tecnologia desempenhou um papel fundamental na simplificação de tarefas técnicas e no aprimoramento da experiência do usuário. Por exemplo, bibliotecas como o Tailwind CSS e o Tanstack React Query ofereceram soluções especializadas que reduziram a complexidade do código e permitiram que a equipe se concentrasse no desenvolvimento das funcionalidades principais do sistema.

Além disso, a utilização dessas tecnologias modernas não apenas otimizou o tempo de desenvolvimento, mas também garantiu a entrega de um produto final robusto e escalável. O React, em conjunto com ferramentas como o TypeScript, trouxe maior confiabilidade ao código e facilitou a manutenção futura do sistema. Por outro lado, a integração com o Firebase possibilitou a implementação de recursos avançados, como a sincronização em tempo real, autenticação segura e armazenamento confiável, sem a necessidade de gerenciar servidores complexos. Em conjunto, essas escolhas tecnológicas garantiram que o *DescompliCar* atendesse às demandas de seus usuários de forma eficiente e consistente, contribuindo diretamente para o alcance dos objetivos do projeto.

5.3 Etapas Do Desenvolvimento

O desenvolvimento do *DescompliCar* seguiu um fluxo estruturado e sistemático, composto por etapas interdependentes que garantiram a organização do projeto e a qualidade do sistema entregue. Cada fase foi planejada para abordar aspectos cruciais do processo, desde a definição dos requisitos até a realização de testes finais, assegurando que o software atendesse às necessidades dos usuários e aos objetivos do projeto.

5.3.1 Levantamento de Requisitos

O levantamento de requisitos constituiu uma etapa essencial no desenvolvimento do *DescompliCar*, uma vez que norteou todas as fases subsequentes do projeto. Essa fase envolveu uma análise detalhada de softwares

existentes no mercado e estudos de caso relacionados à gestão de manutenção de veículos, com o objetivo de identificar boas práticas e lacunas em soluções já disponíveis. Além disso, foram realizadas consultas com potenciais usuários finais, especialmente proprietários de veículos, para compreender suas demandas e desafios cotidianos. Essa abordagem garantiu que o sistema fosse projetado para atender tanto às necessidades práticas, como organização e controle financeiro, quanto às expectativas relacionadas à usabilidade e acessibilidade.

Com base nas análises e nos feedbacks obtidos, foram definidos os requisitos do sistema, divididos em funcionais e não funcionais. Esses requisitos orientaram o desenvolvimento, assegurando que o software atendesse aos objetivos gerais e específicos do projeto de maneira eficaz.

5.3.1.1 Requisitos Funcionais

Os requisitos funcionais especificam as funcionalidades essenciais que o sistema deve implementar para cumprir suas finalidades principais:

- **Gerenciamento de Gastos**:
    - Registro e monitoramento de custos relacionados à manutenção, combustível, seguros e impostos.
    - Categorização de despesas com visualização da distribuição por categoria.
    - Comparação de custos entre períodos e veículos.
- **Histórico de Manutenções**:
    - Registro detalhado de manutenções realizadas, incluindo datas, serviços executados e custos.
    - Visualização e edição de histórico de dados do veículo.
- **Alertas e Notificações**:
    - Notificações automáticas para eventos como manutenção preventiva, renovação de documentos e pagamento de impostos.
- **Relatórios Financeiros**:
    - Geração de gráficos e relatórios personalizados para acompanhamento financeiro.
- **Dashboard**:

    - Resumo em tempo real de despesas, manutenções pendentes e próximos eventos.
- **Autenticação e Cadastro de Usuários**:
    - Login via email/senha e integração com serviços de autenticação (ex.: Google).
    - Suporte a múltiplos veículos por conta.
- **Acessibilidade Multiplataforma**:
    - Interface responsiva para acesso em dispositivos móveis, tablets e desktops.
- **Colaboração e Compartilhamento**:
    - Possibilidade de compartilhamento de dados entre usuários (ex.: famílias ou equipes de manutenção).

5.3.1.2 Requisitos Não Funcionais

Os requisitos não funcionais descrevem características que asseguram a qualidade do sistema em aspectos de desempenho, segurança e escalabilidade:

- **Desempenho**:
    - Resposta rápida na geração de gráficos e relatórios.
    - Atualizações em tempo real via Firebase.
- **Usabilidade**:
    - Interface intuitiva baseada em *feedback* de usuários finais.
    - Navegação clara e autoexplicativa, com textos e ícones acessíveis.
- **Segurança**:
    - Armazenamento de dados protegido por autenticação e regras de segurança no Firebase.
    - Conexão HTTPS para comunicação segura.
- **Escalabilidade**:
    - Suporte ao aumento de usuários e volume de dados sem perda de desempenho.
- **Manutenibilidade**:
    - Uso do GitHub para controle de versões, garantindo histórico de alterações e recuperação de versões anteriores.
- **Compatibilidade**:

- Suporte aos principais navegadores tanto em computadores quanto em dispositivos móveis.

O levantamento desses requisitos garantiu que o desenvolvimento desse projeto fosse orientado por objetivos claros, focados em proporcionar uma solução robusta, escalável e alinhada às necessidades dos usuários. Ao atender a esses requisitos, o *DescompliCar* se posicionou como um sistema confiável e eficiente para a gestão de gastos e manutenção de veículos.

5.3.2 Prototipagem: Criação e validação de protótipos

A etapa de prototipagem foi fundamental no desenvolvimento do *DescompliCar*, pois permitiu visualizar e validar as funcionalidades e o design do sistema antes da implementação completa. Essa fase foi iniciada com a criação de wireframes de baixa fidelidade, que delinearam a estrutura básica das interfaces, como o dashboard, os formulários de registro de despesas e manutenções, e o módulo de relatórios. Esses protótipos iniciais serviram como base para refinamentos, ajudando a alinhar aquilo que estava sendo desenvolvido com as necessidades dos usuários e os requisitos levantados.

Posteriormente, os protótipos de alta fidelidade foram elaborados, incorporando elementos visuais mais próximos da versão final do sistema, como paletas de cores, tipografia e ícones. Essa etapa foi guiada por princípios de design responsivo, garantindo que as interfaces fossem funcionais e visualmente consistentes. O Tailwind CSS foi um aliado importante nessa fase, pois permitiu a aplicação rápida de estilos, possibilitando ajustes em tempo real durante a validação.

A validação dos protótipos envolveu teste com usuário final, que avaliou aspectos como clareza na navegação, disposição das informações e entendimento das funcionalidades. Os feedbacks coletados foram utilizados para implementar melhorias, como ajustes nos textos de botões, reposicionamento de elementos e simplificação de fluxos de uso. Esse processo iterativo garantiu que a interface final fosse intuitiva, atendendo às expectativas do público-alvo e contribuindo para a eficiência do sistema.

A prototipagem, portanto, desempenhou um papel essencial na construção do

DescompliCar, permitindo identificar e corrigir potenciais problemas ainda nas etapas iniciais do desenvolvimento. Dessa forma, assegurou-se que o produto final fosse tecnicamente robusto e visualmente alinhado às melhores práticas de usabilidade.

5.3.3 Implementação: Desenvolvimento das funcionalidades principais

A etapa de implementação foi central no desenvolvimento do DescompliCar, marcada pela construção e integração das principais funcionalidades do sistema. Essa fase teve como base o levantamento de requisitos e os protótipos validados, garantindo que cada módulo fosse desenvolvido de maneira alinhada às necessidades dos usuários finais e aos objetivos do projeto.

O processo de implementação foi conduzido de forma incremental, com foco na entrega de um MVP (Produto Mínimo Viável) que incorporasse as funcionalidades essenciais. Após realizada a “entrega” do MVP, marcada no contexto desse projeto, pela realização de teste de uso, foram desenvolvidos os módulos de autenticação e cadastro de usuários, utilizando o Firebase Authentication para implementar login com email e senha, bem como integração com o Google. Esse módulo garantiu a segurança e a personalização da experiência para os usuários.

Em seguida, o desenvolvimento avançou para funcionalidades específicas, como o gerenciamento de gastos, registro de manutenções e geração de relatórios financeiros. O React foi utilizado para estruturar o front-end, permitindo a criação de componentes reutilizáveis e dinâmicos, como o dashboard e os gráficos de análise. O Firebase desempenhou um papel crucial no back-end, proporcionando armazenamento de dados em tempo real e sincronização instantânea, o que foi essencial para a atualização imediata de informações em múltiplos dispositivos.

Paralelamente, o *Tailwind CSS* foi empregado para garantir que a interface fosse responsiva e visualmente consistente, independentemente do dispositivo utilizado. Ferramentas complementares, como o *React Query*, otimizaram a manipulação de dados e o desempenho do sistema, enquanto o *TypeScript* assegura maior confiabilidade no código, minimizando erros durante o desenvolvimento.

Cada funcionalidade foi implementada e testada de forma iterativa, horas simulando o uso de um usuário final, horas realizando testes supervisionados por usuários finais, permitindo a identificação e correção de falhas antes do avanço para

o próximo módulo. Essa abordagem incremental assegurou a construção de um sistema coeso e funcional, capaz de atender às demandas práticas e expectativas dos usuários finais.

A integração de todas as funcionalidades, incluindo o dashboard, categorização de despesas e levantamento de relatórios, assim como a hospedagem do sistema no firebase hosting, culminou na entrega de um software robusto, escalável e alinhado às melhores práticas de desenvolvimento web.

Por fim, para marcar a conclusão do projeto, foi realizada uma última bateria de testes no software já hospedado, garantindo que todas as funcionalidades estivessem operando corretamente no ambiente de produção. Esses testes finais validaram a estabilidade, a usabilidade e o desempenho do sistema, assegurando que ele atendesse aos requisitos estabelecidos. Com isso, o DescompliCar foi disponibilizado para acesso público, permitindo que qualquer usuário o utilize em diferentes dispositivos de maneira eficiente e segura.

5.3.4 Testes: Realização de testes unitários e de usabilidade com usuário final

A etapa de testes desempenhou um papel essencial no desenvolvimento do DescompliCar, garantindo a funcionalidade, a usabilidade e a confiabilidade do sistema. Para isso, foram realizados testes técnicos, como os unitários e de integração, além de testes de usabilidade com usuários finais, que forneceram *feedbacks* valiosos para aprimorar o software.

Os testes unitários foram conduzidos com o objetivo de verificar a correta funcionalidade de cada módulo isoladamente. Durante essa fase, funcionalidades como o sistema de autenticação, o registro de manutenções e o cálculo de relatórios financeiros foram avaliadas individualmente. Erros técnicos detectados, como falhas na validação de entradas de dados e inconsistências na lógica de algumas operações, foram corrigidos para assegurar o desempenho esperado. Os testes também incluíram a análise do console do navegador, onde mensagens de erro e logs foram utilizados para identificar problemas que não eram visíveis na interface do usuário.

Após os testes unitários, foram realizados testes de usabilidade com usuários finais, selecionados por não possuírem conhecimentos técnicos aprofundados, simulando o público-alvo do software. Durante esses testes supervisionados, os

participantes foram convidados a executar tarefas específicas, como adicionar informações de manutenção, consultar o histórico de despesas e navegar no dashboard. A interação dos usuários com o sistema foi monitorada para identificar possíveis dificuldades, como confusões geradas por textos de botões ou pela organização visual da interface.

Os testes de usabilidade contaram com a participação de quatro usuários, incluindo o próprio autor e mais três participantes com diferentes perfis:

- Um usuário homem, na faixa dos 50 anos, proprietário de dois veículos e com conhecimento básico do uso da internet.
- Uma usuária mulher, na faixa dos 20 anos, sem veículo e com conhecimento básico do uso da internet.
- Um usuário homem, na faixa dos 20 anos, proprietário de um veículo e com conhecimento avançado sobre o uso da internet, incluindo noções de desenvolvimento de software.

Os *feedbacks* obtidos revelaram melhorias necessárias, incluindo alterações em textos e botões para torná-los mais claros e intuitivos. Por exemplo, o botão de registro de manutenções, inicialmente intitulado "Nova Manutenção", foi renomeado para "Adicionar Manutenção", alinhando-se melhor às expectativas dos usuários. Além disso, foram realizados ajustes na disposição visual de informações e no fluxo de navegação para proporcionar uma experiência mais fluida e acessível.

A combinação de testes técnicos e de usabilidade permitiu validar não apenas a funcionalidade, mas também a adequação do software às necessidades práticas do público-alvo. Essa abordagem iterativa assegurou que o DescompliCar fosse entregue como um sistema confiável, intuitivo e capaz de atender às expectativas dos usuários finais.

5.4 Desafios e Soluções

O desenvolvimento do *DescompliCar* enfrentou desafios técnicos e operacionais em diferentes etapas do projeto, exigindo adaptações e soluções criativas para assegurar o cumprimento dos objetivos propostos. Entre os principais desafios, destacam-se a integração das tecnologias utilizadas, os ajustes com base no *feedback* dos usuários e limitações técnicas relacionadas ao uso do Firebase.

5.4.1 Integração de Tecnologias

A integração entre o front-end, desenvolvido com React, e o back-end, baseado no Firebase, foi uma das etapas mais complexas. Essa integração envolveu a sincronização de funcionalidades como autenticação, armazenamento de dados e atualização em tempo real, o que demandou testes rigorosos e ajustes constantes. Além disso, desafios específicos, como a configuração de login social e a manipulação de dados dinâmicos no dashboard, exigiram a aplicação de boas práticas e uma abordagem incremental.

Para superar essas dificuldades, o projeto adotou a integração modular, priorizando funcionalidades críticas como autenticação e cadastro. O uso de ferramentas como React Query foi essencial para gerenciar a comunicação eficiente entre as interfaces e o banco de dados, garantindo estabilidade e desempenho consistente.

5.4.2 Ajustes com Base no Feedback dos Usuários

Os testes de usabilidade com usuários finais revelaram a necessidade de ajustes tanto no design quanto na clareza de algumas funcionalidades. Por exemplo, dificuldades foram identificadas na interpretação de botões e mensagens, como no caso do texto do botão de registro de manutenções, que inicialmente causava confusão. Para corrigir isso, o texto foi reformulado para "Adicionar Manutenção", tornando a funcionalidade mais clara e alinhada às expectativas dos usuários.

Além disso, foram realizados aprimoramentos na organização visual das informações e no design da interface, utilizando o Tailwind CSS para ajustes rápidos e consistentes. Esses ajustes resultaram em um sistema mais intuitivo e fácil de usar, promovendo uma experiência positiva para o público-alvo.

5.4.3 Limitações Técnicas do Firebase

Durante o desenvolvimento, foram identificadas limitações específicas no uso do Firebase que impactaram algumas funcionalidades. Um dos desafios foi a impossibilidade de implementar a troca de e-mail cadastrada pelos usuários no sistema de autenticação do Firebase. Como solução alternativa, foi estabelecido que, caso o usuário deseje alterar o e-mail vinculado, será necessário excluir a conta e criar um novo cadastro, respeitando as limitações impostas pela plataforma.

Outro obstáculo foi a inviabilidade da configuração de disparos automáticos de notificações por e-mail baseados em datas. Embora funcionalmente importante para alertas e lembretes, essa funcionalidade foi descartada devido aos custos elevados associados ao uso de verificações contínuas no Firebase, que poderiam ultrapassar R$1.000 mensais.

6 RESULTADOS

Este capítulo apresenta os principais resultados alcançados durante o desenvolvimento do *DescompliCar*. A partir da aplicação das metodologias descritas no capítulo anterior e do uso das tecnologias selecionadas, foi possível entregar um sistema funcional, acessível e alinhado às necessidades dos usuários finais.

Os resultados incluem a implementação bem-sucedida das principais funcionalidades do software, como o registro e monitoramento de gastos, alertas de manutenção e geração de relatórios financeiros personalizados. Além disso, o *feedback* recebido durante os testes de usabilidade permitiu aprimorar a interface e corrigir inconsistências, contribuindo para uma experiência de usuário intuitiva e eficiente.

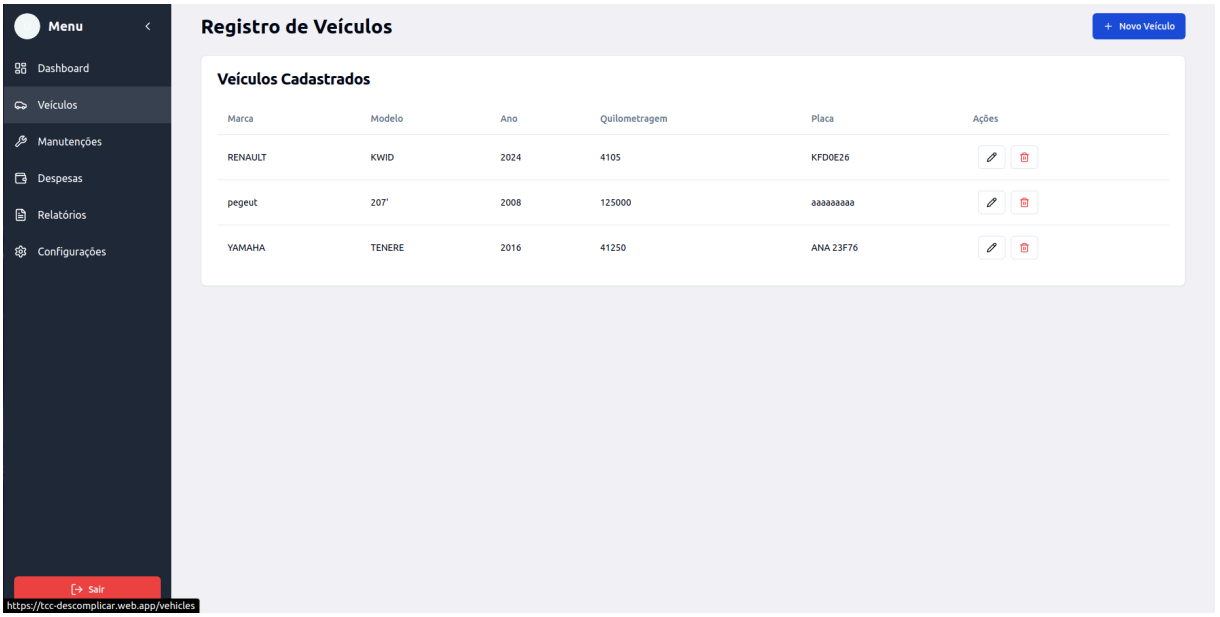
O sistema final reflete o cumprimento dos objetivos propostos, destacando-se como uma solução prática e escalável para a gestão financeira e organização de manutenções veiculares. Neste capítulo, são detalhados os resultados obtidos, as melhorias implementadas e o impacto das escolhas técnicas no desempenho e usabilidade do software.

6.1 Funcionalidades Implementadas

O DescompliCar foi desenvolvido com um conjunto de funcionalidades que visam atender às necessidades de proprietários de veículos, proporcionando uma gestão eficiente de despesas e manutenção. Abaixo, estão listadas as principais funcionalidades implementadas no sistema, conforme definido nos requisitos levantados e validado ao longo do processo de desenvolvimento:

- **Cadastro e gerenciamento de veículos**
    - Conforme demonstrado na Figura 4, essa funcionalidade permite ao usuário registrar os dados básicos de identificação dos veículos na plataforma, como marca, modelo, ano, quilometragem e placa.
    - Oferece opções para editar ou excluir registros existentes, garantindo um controle eficiente das informações cadastradas.

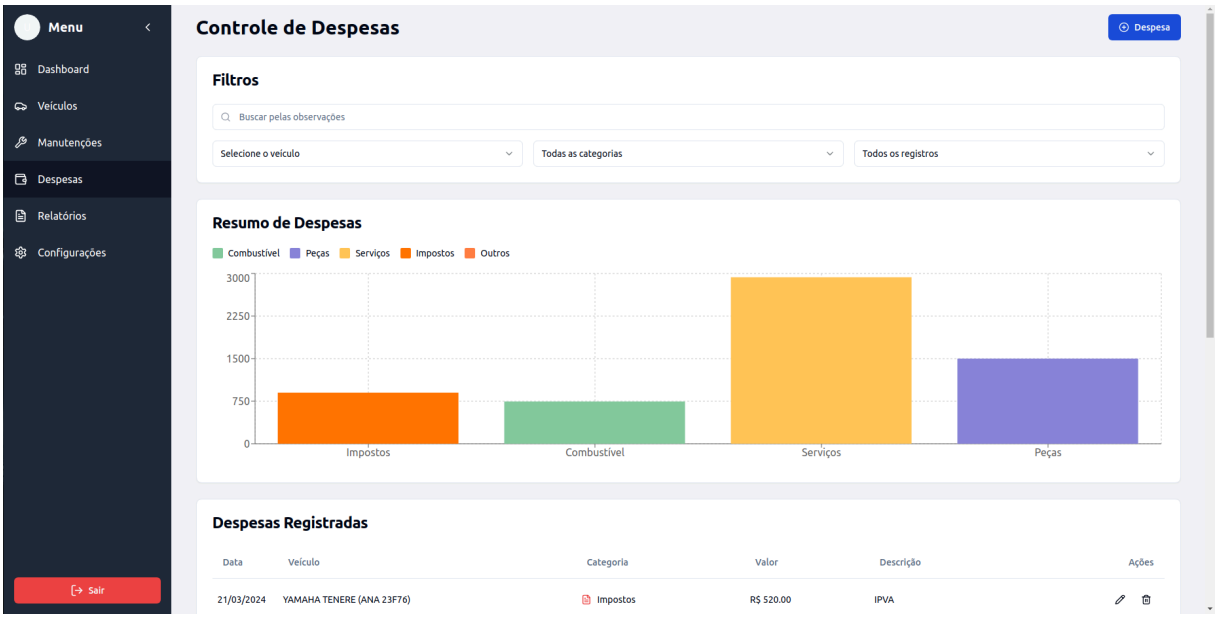
Figura 4 - Tela de Veículos do Software



Fonte: Elaborado pelo autor (2024)

- **Registro de Despesas**:
  - Conforme demonstrado na Figura 5, permite ao usuário cadastrar e monitorar os custos relacionados à manutenção, combustível, seguros e impostos.
  - As despesas são categorizadas manualmente, facilitando a visualização e análise personalizada pelo usuário.
  - visualização de métricas básicas personalizáveis de gastos com o veículo.

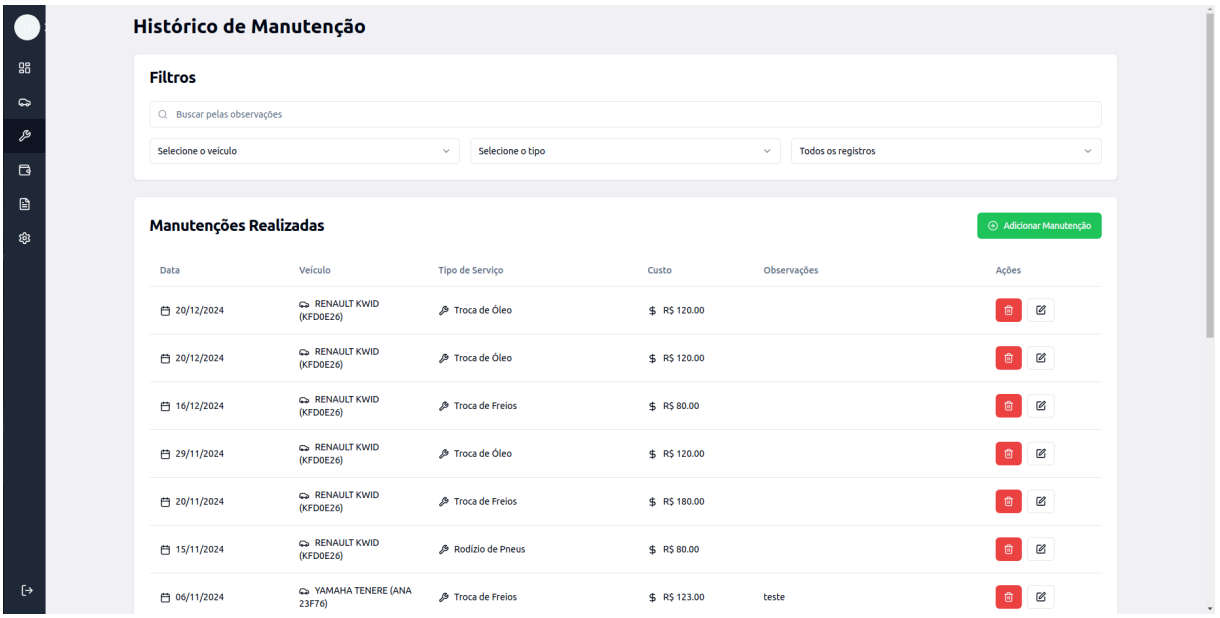
Figura 5 - Tela de Despesas do software



Fonte: Elaborado pelo autor (2024)

- **Histórico de Manutenções**:
  - Conforme demonstrado na Figura 6, contempla o registro detalhado de manutenções realizadas, incluindo datas, serviços executados e custos.
  - O histórico pode ser visualizado e editado conforme necessário.

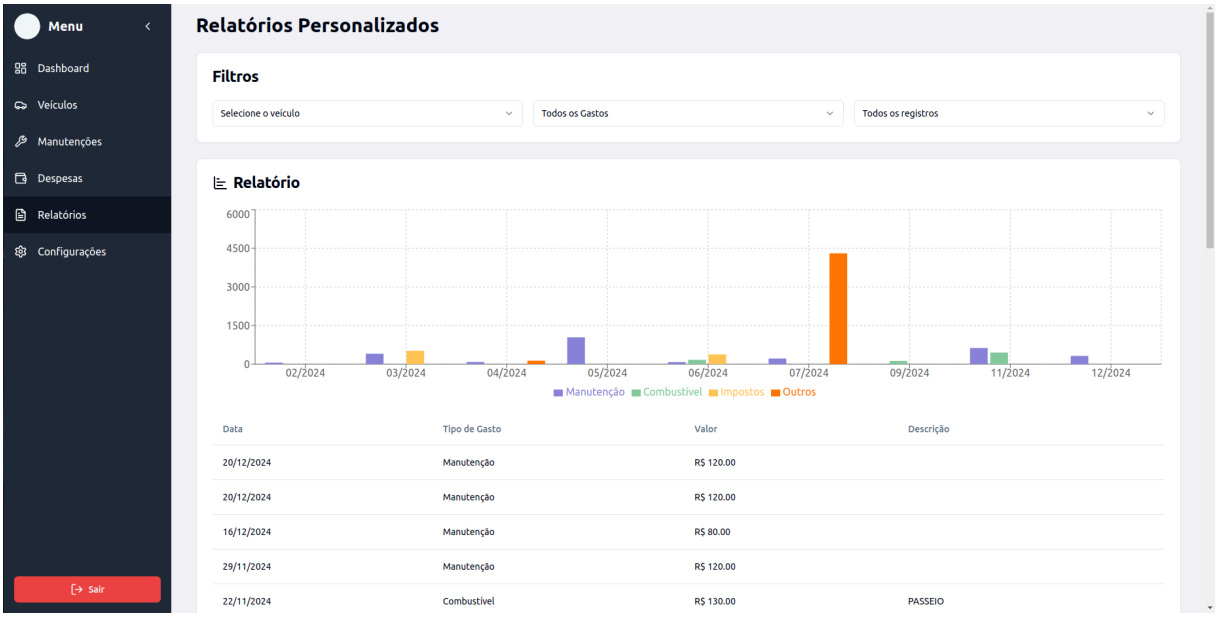
Figura 6 - Tela de manutenções do Software



Fonte: Elaborado pelo autor (2024)

- **Relatórios Financeiros**:
  - Conforme demonstrado na figura 7, possibilita a geração de gráficos e relatórios personalizados usando os filtros que auxiliam no acompanhamento financeiro e na comparação de custos entre períodos ou veículos.

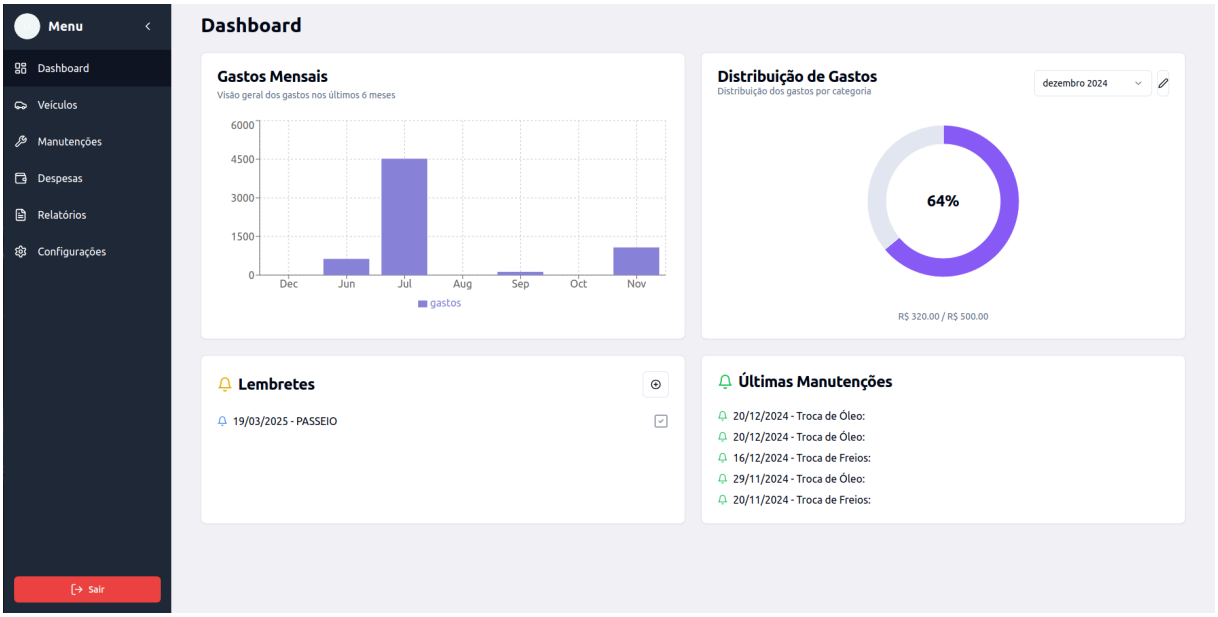
Figura 7 - Tela de relatórios do software



Fonte: Elaborado pelo autor (2024)

- **Dashboard**:
  - Conforme demonstrado na Figura 8, exibe um resumo em tempo real das principais informações, como manutenções recentes, manutenções pendentes e próximos eventos registrados.

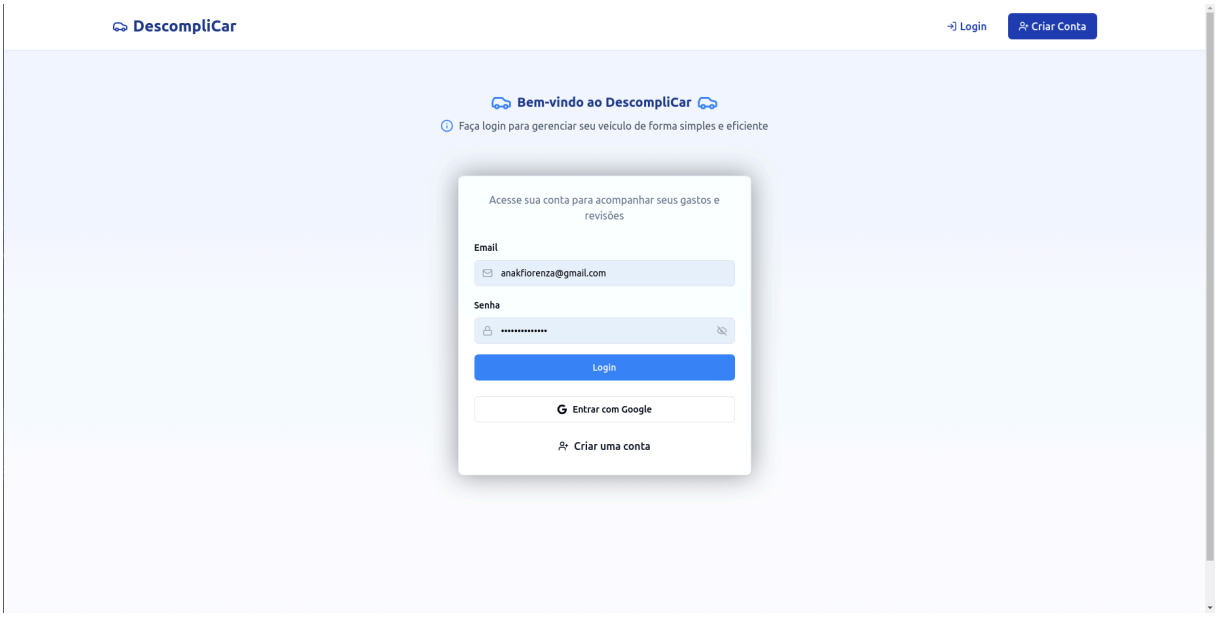
Figura 8 - Tela de Dashboard do software



Fonte: Elaborado pelo autor (2024)

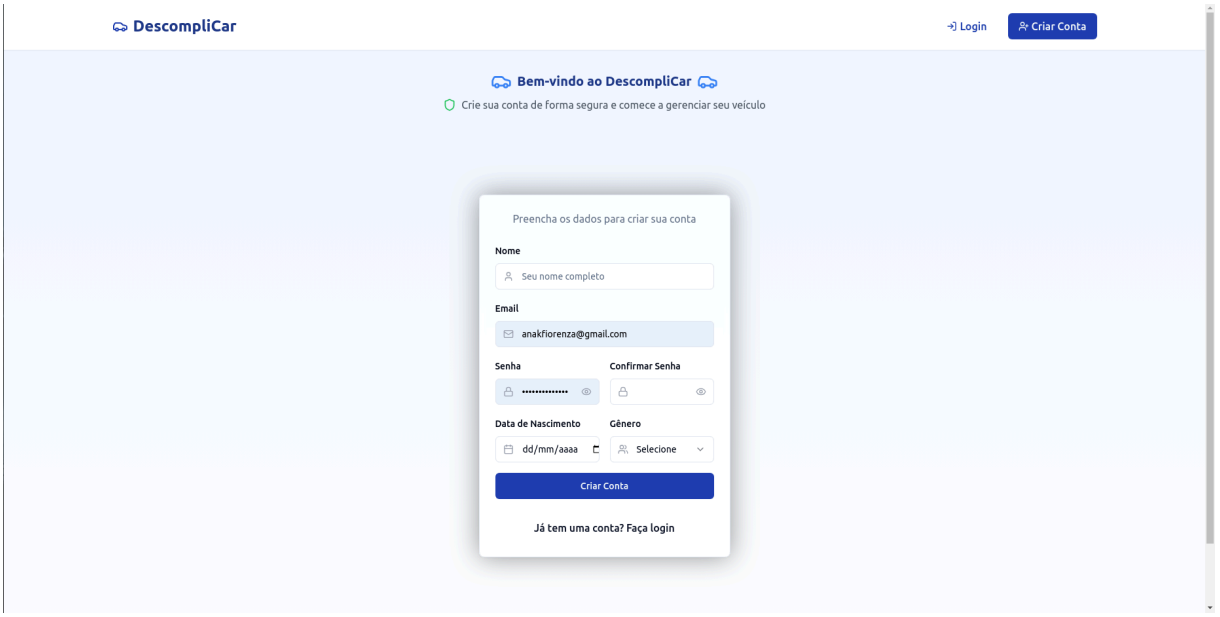
- **Autenticação de Usuários**:
  - Conforme demonstrado nas Figuras 9 e 10, apresenta um sistema de login e cadastro com email/senha, integrado ao Google Authentication para maior acessibilidade.
  - Suporte a múltiplos dispositivos com a mesma conta de usuário.
  - Suporte a login com a conta Google, agilizando o processo de cadastro.

Figura 9 - Tela de login do software



Fonte: Elaborado pelo autor (2024)

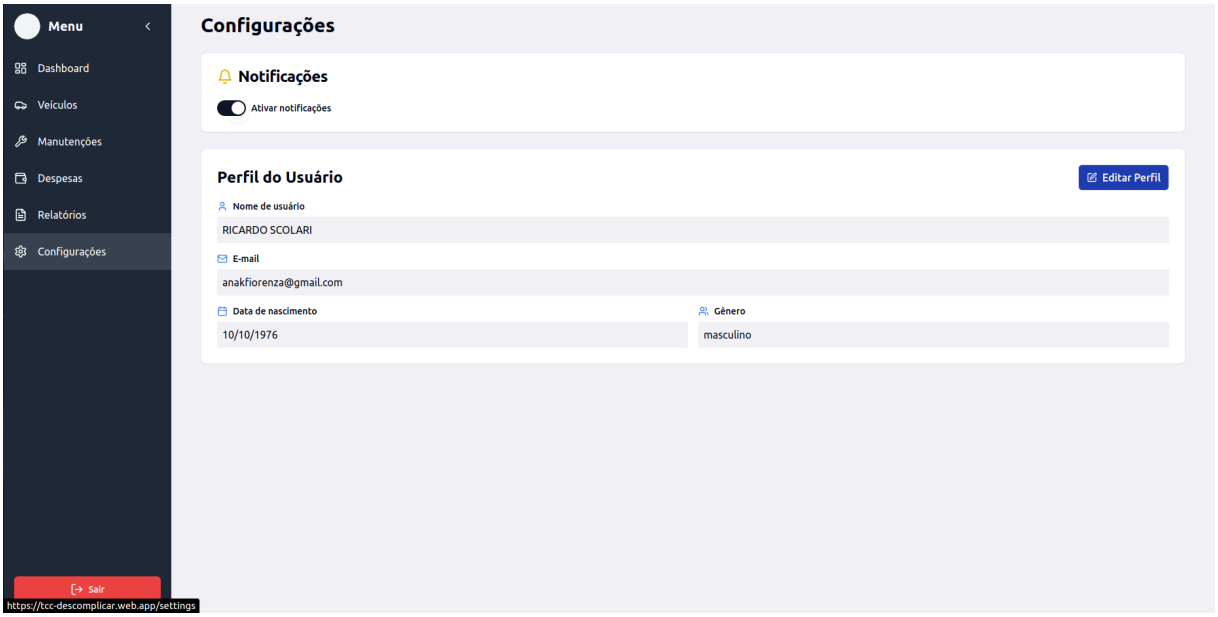
Figura 10 - Tela de cadastro do Software



Fonte: Elaborado pelo autor (2024)

- **Configuração de dados do usuário**
    - Conforme demonstrado na figura 11, exibe e permite gerenciar os dados registrados desse usuário, por exemplo, nome, e-mail, gênero, etc.

Figura 11 - Tela de configurações do usuário



Fonte: Elaborado pelo autor (2024)

- **Interface Responsiva**:
    - Conforme demonstrado nas figuras 4 à 11 o design contempla acesso via computadores e telas grades
    - Conforme demonstrado nas figuras 12 à 16, apresenta um design adaptado para acesso em dispositivos móveis, tablets e desktops, garantindo uma experiência consistente em diferentes plataformas.

Figura 12 - Página Home no smartphone

Fonte: Elaborado pelo autor (2024)

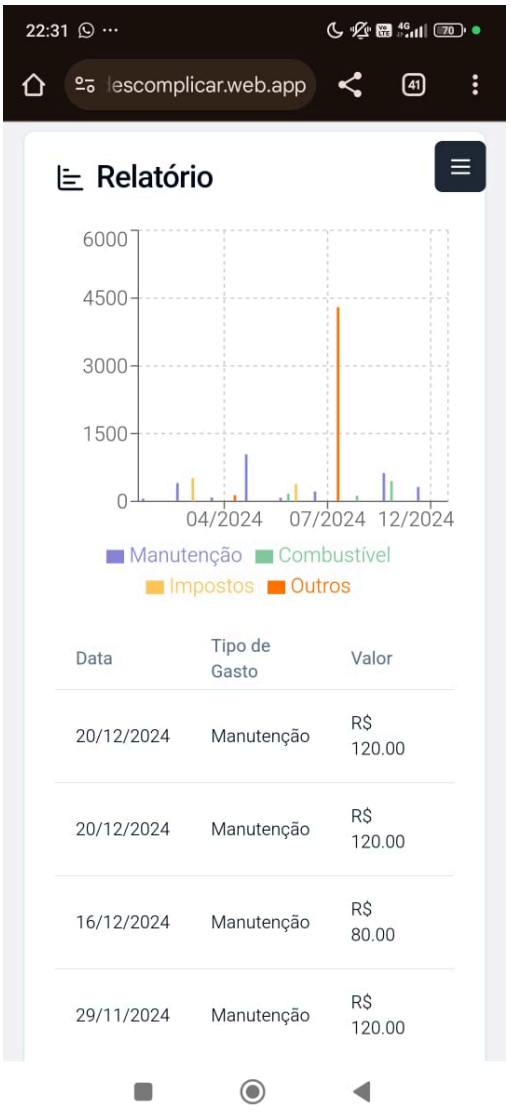
Figura 13 - Tela Dashboard no Smartphone

Fonte: Elaborado pelo autor (2024)

Figura 14 - Tela de Despesas no Smartphone

Fonte: Elaborado pelo autor (2024)

Figura 15 - Tela de relatório no smartphone



Fonte: Elaborado pelo autor (2024)

Figura 16 - Tela de veículos no smartphone

Fonte: Elaborado pelo autor (2024)

Essas funcionalidades foram desenvolvidas e validadas para atender aos objetivos propostos, garantindo que o *DescompliCar* se posicione como uma solução prática, escalável, intuitiva e multiplataforma para a gestão de veículos.

6.2 Validação e *Feedback*

A validação do *DescompliCar* foi realizada por meio de testes de usabilidade supervisionados, envolvendo usuários finais que representavam o público-alvo do sistema. Os participantes foram selecionados com base em sua familiaridade com veículos, mas sem conhecimento técnico prévio sobre o funcionamento do software, o que garantiu uma avaliação mais próxima da experiência de uso real. Essa etapa foi essencial para verificar a eficácia, clareza e funcionalidade do sistema, além de identificar dificuldades enfrentadas pelos usuários e melhorias necessárias.

Os testes permitiram coletar *feedbacks* valiosos, que refletiram as experiências práticas dos participantes. Entre os pontos destacados estavam sugestões para aprimorar a clareza de botões e a organização visual do dashboard, além de ajustes para facilitar a navegação. Essas informações foram fundamentais para guiar as melhorias implementadas no software, assegurando que ele fosse intuitivo, funcional e alinhado às expectativas de seu público-alvo antes de sua disponibilização final.

6.2.1 *Feedback* Recebido

Os usuários forneceram *feedbacks* valiosos durante a interação com o sistema. Entre os principais pontos destacados estão:

- **Clareza na Interface**: Alguns usuários relataram dificuldades em compreender o propósito de certos botões, como o botão de registro de manutenções, cujo texto inicial "Nova Manutenção" gerava dúvidas.
- **Organização Visual**: Sugestões foram feitas para melhorar a disposição das informações no dashboard, de forma a facilitar a visualização dos dados mais relevantes.
- **Facilidade de Navegação**: De maneira geral, os usuários elogiaram a simplicidade da interface e a responsividade do sistema, destacando a rápida adaptação às funcionalidades disponíveis.

6.2.2 Melhorias Implementadas com Base no *Feedback*

Com base nos *feedbacks* recebidos, foram implementadas as seguintes melhorias:

- **Ajustes nos Textos dos Botões**: O botão "Nova Manutenção" foi renomeado para "Adicionar Manutenção", tornando sua funcionalidade mais clara e alinhada às expectativas dos usuários.
- **Reorganização do Dashboard**: Elementos visuais foram reposicionados para destacar informações prioritárias, como manutenções pendentes e próximos eventos.
- **Refinamento da Interface Visual**: Foram realizados ajustes no design, utilizando o Tailwind CSS para melhorar a disposição e a estética dos elementos, garantindo uma navegação mais intuitiva.

6.3 Síntese dos Resultados

Os resultados obtidos com o desenvolvimento do *DescompliCar* evidenciam sua eficácia como uma solução prática para a gestão financeira e de manutenção de veículos. As funcionalidades implementadas abordaram diretamente os problemas identificados, como a falta de organização dos usuários e a ausência de ferramentas acessíveis e intuitivas no mercado. O sistema se destacou por centralizar informações, deixar os dados de forma visualmente agradável com gráficos e afins e oferecer relatórios detalhados, contribuindo para a economia de tempo e recursos, além de promover maior segurança no planejamento das manutenções.

Entretanto, para consolidar uma análise mais ampla, é necessário avaliar os benefícios do DescompliCar em comparação com outros sistemas existentes e considerar as limitações enfrentadas ao longo do projeto. Além disso, deve-se explorar as possibilidades de evolução do software, com foco em novas funcionalidades e melhorias futuras. Essas discussões serão aprofundadas no próximo capítulo, onde os resultados serão analisados de forma crítica, destacando o impacto do sistema, suas restrições e o potencial para inovações.

7 DISCUSSÃO

Este capítulo apresenta uma análise crítica dos resultados alcançados no desenvolvimento do DescompliCar, explorando suas contribuições, limitações e potenciais melhorias. Inicialmente, é realizada uma comparação com sistemas similares analisados durante a fase de levantamento de requisitos, destacando as vantagens e os diferenciais do software em relação às soluções existentes. Em seguida, são discutidas as limitações identificadas, incluindo restrições técnicas e funcionalidades não implementadas devido a desafios enfrentados ao longo do projeto. Por fim, são propostas possibilidades de evolução, considerando o aprimoramento do sistema e a inclusão de novas funcionalidades que possam ampliar seu impacto e alcance.

Esse capítulo tem como objetivo contextualizar os resultados do DescompliCar no cenário atual de ferramentas de gestão de manutenção veicular, proporcionando uma visão abrangente sobre as contribuições do projeto e seu potencial futuro.

7.1 Análise de Sistemas Existentes

Este capítulo analisa estudos de caso para identificar boas práticas e lacunas em sistemas relacionados à gestão de manutenção de veículos. A partir dessa análise, foram definidas metas para atender às necessidades do usuário, compreendendo melhor o processo de desenvolvimento de sistemas e comparando soluções estudadas com o **DescompliCar**.

7.1.1 Estudo de Caso 1: Software para Auxílio na Manutenção de Veículos

- **Descrição**: Software desenvolvido para Android que auxilia donos de carros na manutenção e controle de gastos.
- **Pontos Fortes**:
    - Facilidade de uso para usuários Android.
    - Funcionalidades voltadas ao registro e monitoramento de despesas.
- **Limitações**:
    - Restrito ao ambiente Android, limitando sua acessibilidade.

    - Dificuldade para atualizações rápidas e sincronização entre dispositivos.
- **Impacto no DescompliCar**: Inspirou a adoção de uma abordagem multiplataforma, permitindo atualizações rápidas e sincronização instantânea entre dispositivos.

7.1.2 Estudo de Caso 2: Drivvo

- **Descrição**: Aplicativo popular para gerenciamento financeiro e manutenção de veículos, disponível em Android, iOS e web.
- **Pontos Fortes**:
    - Funcionalidades como registro de despesas, notificações preventivas e relatórios financeiros.
    - Disponibilidade multiplataforma (Android, iOS e web).
- **Limitações**:
    - Dependência de backups manuais para sincronização em algumas versões.
    - Muitas funcionalidades disponíveis apenas na versão paga.
- **Impacto no DescompliCar**: Influenciou a priorização de gratuidade plena e integração com o Firebase para sincronização automática.

7.1.3 Conclusão da Análise

Os estudos de caso evidenciaram a importância de acessibilidade, usabilidade e escalabilidade nos sistemas de gestão veicular. O **DescompliCar** foi projetado com base nesses aprendizados, destacando-se pela:

- **Acessibilidade Multiplataforma**: Implementação responsiva via Firebase Hosting, garantindo uso eficiente em qualquer dispositivo.
- **Facilidade de Atualização**: Atualizações em tempo real com o comando “firebase deploy”.
- **Escalabilidade e Sincronização**: Uso do Realtime Database e Firestore para dados atualizados e consistentes em todos os dispositivos.

7.2 Comparação com Estudos de Casos

Este capítulo destaca como o DescompliCar se posiciona em relação a outros sistemas de gestão veicular, avaliando funcionalidades, usabilidade, acessibilidade e modelo de negócio.

7.2.1 Drivvo

- **Semelhanças**:
  - Registro de despesas, notificações preventivas e relatórios financeiros.
- **Diferenciais do DescompliCar**:
  - Gratuidade plena.
  - Sincronização automática baseada no Firebase.
  - Interface responsiva e multiplataforma.

7.2.2 Fuelio

- **Foco**: Rastreamento de custos com combustível, sendo intuitivo para esse propósito.
- **Limitações**: Gestão limitada a combustível e falta de integração com outras despesas.
- **Diferenciais do DescompliCar**: Sistema abrangente que cobre despesas como manutenção, impostos e seguros, consolidando informações em relatórios customizáveis.

7.2.3 Motolog

- **Foco**: Gestão de veículos empresariais.
- **Limitações**: Interface voltada para frotas, menos adaptada ao uso pessoal.
- **Diferenciais do DescompliCar**: Interface intuitiva para uso pessoal e pequenos negócios.

7.2.4 Car Expenses

- **Foco**: Gestão detalhada de gastos.
- **Limitações**: Interface menos intuitiva e suporte parcial a dispositivos móveis.
- **Diferenciais do DescompliCar**: Layout responsivo e suporte completo para smartphones, tablets e desktops.

7.2.5 Sistemas Básicos (Caoa e Sim Veículos)

- **Foco**: Funcionalidades limitadas a cadastro e login.
- **Limitações**: Ausência de ferramentas robustas de gestão financeira.
- **Diferenciais do DescompliCar**: Dashboard completo com visão consolidada de dados financeiros e notificações automáticas.

7.2.6 Consolidação dos Diferenciais

O **DescompliCar** se destaca pelos seguintes aspectos:

- **Acessibilidade Multiplataforma**: Interface responsiva e suporte completo a dispositivos diversos.
- **Gratuidade e Transparência**: Todos os recursos disponíveis gratuitamente, sem propagandas.
- **Abrangência Funcional**: Integração de todas as categorias de despesas.
- **Experiência do Usuário (UX)**: Interface simplificada, intuitiva e personalizável.
- **Integração e Sincronização**: Dados consistentes e atualização automática com o Firebase.

7.3 Limitações do Projeto

Embora o *DescompliCar* tenha alcançado resultados significativos em termos de funcionalidades e usabilidade conforme discutido ao longo deste capítulo, o projeto enfrentou algumas limitações que refletem desafios técnicos e estratégicos durante o desenvolvimento. Estas limitações, em parte, estão relacionadas à complexidade de certas implementações e ao escopo estabelecido inicialmente, além de restrições impostas por ferramentas e plataformas utilizadas.

Uma das limitações mais notáveis foi a impossibilidade de permitir a troca do e-mail cadastrado no sistema, devido à ausência de uma funcionalidade nativa no Firebase Authentication que atendesse a essa necessidade de maneira viável. Como resultado, os usuários ficam limitados a utilizar o e-mail inicialmente cadastrado ou, alternativamente, excluir sua conta e criar uma nova. Essa restrição pode impactar a experiência de longo prazo de alguns usuários, especialmente aqueles que desejarem alterar suas informações de login por questões pessoais ou organizacionais.

Além disso, a configuração de disparos automáticos de notificações baseadas em datas inseridas pelos usuários foi outro desafio enfrentado. Apesar de sua relevância para o público-alvo, a implementação dessa funcionalidade no Firebase demandava o uso de serviços pagos, como funções em nuvem constantemente em execução. Isso poderia gerar custos elevados, inviabilizando a sua implementação no software.

Outras funcionalidades planejadas, como a expansão para suporte a múltiplos idiomas e maior personalização de relatórios financeiros, também não foram implementadas nesta versão do sistema devido a limitações de tempo e recursos disponíveis.

Essas restrições, contudo, não apenas destacam as áreas que podem ser aprimoradas, mas também abrem caminho para um planejamento estratégico de evolução do sistema. Nesse sentido, o *DescompliCar* apresenta um potencial significativo para futuras melhorias, permitindo que novas funcionalidades sejam incorporadas para atender a um público mais amplo e atender às demandas identificadas ao longo do desenvolvimento e uso do software.

Essa perspectiva de evolução será detalhada no próximo tópico, onde serão discutidas sugestões de aprimoramentos e novas funcionalidades que poderão ampliar ainda mais o impacto e a relevância do sistema no cotidiano dos usuários.

7.4 Trabalhos Futuros

Com base nas limitações identificadas e nas análises realizadas, diversas possibilidades de evolução podem ser consideradas para aprimorar o DescompliCar e expandir suas funcionalidades. Essas sugestões têm o objetivo de indicar caminhos para futuras melhorias, atendendo a demandas ainda não contempladas e

ampliando o potencial de uso do sistema para diferentes perfis de usuários.

Uma das possibilidades seria a implementação de suporte para múltiplos idiomas, visando tornar o sistema mais acessível a usuários de diferentes nacionalidades. Essa funcionalidade poderia ampliar o alcance do DescompliCar para contextos internacionais, onde o gerenciamento de veículos é igualmente relevante.

Outra sugestão envolve a personalização avançada de relatórios financeiros, permitindo que os usuários configurem filtros e categorizem despesas de maneira mais detalhada. Essa funcionalidade poderia atender a públicos com necessidades específicas, como empresas que gerenciam frotas ou usuários que possuem múltiplos veículos.

Também pode ser avaliada a integração com sistemas externos, como aplicativos de rastreamento de quilometragem e plataformas de manutenção automotiva. Essa integração poderia otimizar a experiência do usuário ao automatizar o preenchimento de dados e reduzir o esforço manual.

Adicionalmente, a criação de um fórum para compartilhamento de dados e conversas entre usuários apresenta-se como uma evolução estratégica. Essa funcionalidade permitiria a troca de experiências, dicas e soluções entre os utilizadores do sistema, fomentando uma comunidade ativa e colaborativa em torno do DescompliCar.

Outra melhoria importante seria a implementação de um mecanismo para leitura automática de notas de manutenção veicular. Por meio de tecnologias de reconhecimento de texto (OCR), o sistema poderia extrair informações relevantes de notas fiscais e registrá-las diretamente no banco de dados, reduzindo o trabalho manual do usuário e aumentando a precisão dos registros.

Por fim, a revisão da funcionalidade de troca de e-mail cadastrada diretamente pelo usuário, atualmente limitada pelas configurações do Firebase Authentication, apresenta-se como uma oportunidade de estudo. A exploração de alternativas técnicas no backend pode ser uma solução viável para aumentar a flexibilidade e a autonomia do usuário.

8 CONCLUSÃO

O desenvolvimento do **DescompliCar** teve como objetivo principal criar uma ferramenta técnica eficaz para a gestão de manutenção e controle financeiro de veículos. Por meio da aplicação da metodologia Scrum, foi possível organizar o trabalho em sprints, garantindo entregas incrementais e validações contínuas durante o desenvolvimento.

A escolha de tecnologias como React para o front-end e Firebase para o back-end mostrou-se adequada para atender aos requisitos técnicos do projeto. O React possibilitou a criação de interfaces dinâmicas e responsivas, enquanto o Firebase forneceu uma infraestrutura segura e escalável, com sincronização em tempo real, facilitando o armazenamento e a recuperação de dados.

As funcionalidades implementadas, como registro de despesas, geração de relatórios e categorização de gastos, foram projetadas para atender a necessidades práticas de usuários finais, garantindo eficiência e usabilidade. Testes realizados com usuários permitiram identificar e corrigir problemas técnicos, bem como ajustar aspectos da interface para melhorar a experiência de uso.

Embora tenham sido enfrentadas limitações, como restrições na personalização de notificações e na gestão de alterações no sistema de autenticação do Firebase, estas não comprometeram a funcionalidade geral do sistema. As soluções adotadas atenderam aos requisitos do projeto, mantendo o foco na entrega de um software tecnicamente robusto e funcional.

O projeto cumpriu seus objetivos, apresentando um sistema funcional e modular, com potencial para expansões futuras. Os resultados obtidos demonstram a viabilidade técnica do uso das tecnologias escolhidas e da abordagem metodológica empregada, fornecendo uma base sólida para novos desenvolvimentos ou aprimoramentos

REFERÊNCIAS

ABECom. **O que é manutenção preventiva**. Abecom. Disponível em: https://www.abecom.com.br/o-que-e-manutencao-preventiva/. Acesso em: 21 nov. 2024.

ARAGÃO, C. R. V. **A percepção do usuário sobre o fator usabilidade das páginas da web voltadas para o comércio eletrônico**. 2001. Dissertação (Mestrado em Engenharia de Produção) - Programa de Pós-Graduação em Engenharia de Produção, UFSC, Florianópolis. Disponível em: https://repositorio.ufsc.br/xmlui/bitstream/handle/123456789/80278/185214.pdf?sequence=1&isAllowed=y. Acesso em: 23 nov. 2024

BRASIL. ***Lei nº 13.709, de 14 de agosto de 2018.*** Dispõe sobre a proteção de dados pessoais e altera a Lei nº 12.965, de 23 de abril de 2014 (Marco Civil da Internet). Diário Oficial da União: seção 1, Brasília, DF, p. 1, 15 ago. 2018.

CARSUGHI, Claudia. **Falta de manutenção dos veículos é responsável por 30% dos acidentes de trânsito**. UOL. 26 fev. 2024. Disponível em: https://carsughi.uol.com.br/2024/02/falta-de-manutencao-dos-veiculos-e-responsavel-por-30-dos-acidentes-de-transito/. Acesso em: 23 nov. 2024.

ESPINHA, Roberto Gil. **SCRUM**: o que é, como funciona e como aplicar essa metodologia ágil. Artia Blog, 05 abr. 2024. Disponível em: https://artia.com/blog/scrum/. Acesso em: 26 ago. 2024.

EUROPEAN UNION. *General Data Protection Regulation (GDPR) – Regulation (EU) 2016/679.* **Official Journal of the European Union**, Brussels, 27 Apr. 2016. Disponível em: https://eur-lex.europa.eu/eli/reg/2016/679/oj. Acesso em: 28 nov. 2024.

GITHUB, INC. **Documentação oficial do GitHub**. Disponível em: https://docs.github.com/pt. Acesso em: 30 mar. 2024.

GOOGLE LLC. **Site oficial do Google Firebase**. Disponível em: https://firebase.google.com/docs?hl=pt. Acesso em: 30 mar. 2024.

MENEZES, Gustavo Campos; FERNANDES, Camila Caroline Martins; SILVA, Camila Maria Rodrigues; GAIA, Rebeca Larissa Silva. **CHECAR - SOFTWARE PARA AUXÍLIO NA MANUTENÇÃO DE VEÍCULOS**. In: CONFERÊNCIAS CEFET-MG, 27ª Mostra Específica de Trabalhos e Aplicações. Disponível em: https://www.conferencias.cefetmg.br/index.php/27META/27META/paper/view/3068. Acesso em: 30 mar. 2024.

META PLATFORMS, INC. **Site Oficial do React**. Disponível em: https://react.dev/learn. Acesso em: 30 mar. 2024.

NIELSEN, Jakob. **Usability engineering**. San Francisco: Morgan Kaufmann, 1994.
INTERNATIONAL ORGANIZATION FOR STANDARDIZATION (ISO). **ISO 9241-11: Ergonomic requirements for office work with visual display terminals (VDTs) — Guidance on usability**. 1998. Disponível em: https://www.iso.org/standard/16883.html. Acesso em: 23 nov. 2024.

SILVA, Luciano Lourenço da; SANTOS, Edcarlos Damião dos; TORREÃO, Aety Augusto Castello Branco. **A importância da manutenção de veículos automotores**. 2004. 22 f. (Engenharia mecânica) – Centro Universitário dos Guararapes, São Paulo, 2004. Disponível em: https://repositorio-api.animaeducacao.com.br/server/api/core/bitstreams/a843e24e-b79c-4dc6-88ad-ce39834ae088/content. Acesso em: 30 mar. 2024.

SOARES, Leônidas Garcia. **Avaliação de usabilidade, por meio do índice de satisfação dos usuários, de um software gerenciador de websites**. 2004. 156 f. (Mestrado Profissionalizante em Engenharia) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2004. Disponivel em: https://lume.ufrgs.br/handle/10183/4622

SOARES, Marcelo M. **Ergonomia das interfaces homem-computador**. São Paulo: Editora Blucher, 2004.

SUTHERLAND, Jeff. **Scrum**: a arte de fazer o dobro do trabalho na metade do tempo. Rio de Janeiro: Leya, 2014.

WORLD WIDE WEB CONSORTIUM (W3C). **Web Content Accessibility Guidelines (WCAG) 2.1. 2018**. Disponível em: https://www.w3.org/TR/WCAG21/. Acesso em: 23 nov. 2024.