



DISSERTAÇÃO DE MESTRADO PROFISSIONAL

**PROPOSTA DE SOLUÇÃO E ARQUITETURA PARA
AVALIAÇÃO DE CONDUÇÃO SEGURA DE USUÁRIOS
DE VEÍCULOS AUTOMOTORES PARA BENEFÍCIOS
EM SEGUROS VEICULARES**

Programa de Pós-Graduação Profissional em Engenharia Elétrica

DEPARTAMENTO DE ENGENHARIA ELÉTRICA
FACULDADE DE TECNOLOGIA
UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

DISSERTAÇÃO DE MESTRADO PROFISSIONAL

**PROPOSTA DE SOLUÇÃO E ARQUITETURA PARA
AVALIAÇÃO DE CONDUÇÃO SEGURA DE USUÁRIOS
DE VEÍCULOS AUTOMOTORES PARA BENEFÍCIOS
EM SEGUROS VEICULARES**

*Dissertação de Mestrado Profissional submetida ao Departamento de Engenharia
Elétrica como requisito parcial para obtenção
do grau de Mestre em Engenharia Elétrica*

Banca Examinadora

Prof Dr. Fábio Lúcio Lopes de Mendonça, _____
ENE/UnB
Orientador/Presidente

Prof Dra. Edna Dias Canedo, ENE/UnB _____
Examinador Interno

Prof Dr. Gilmar dos Santos Marques, UPIS _____
Examinador Externo

Prof Dr. Daniel Alves da Silva, PPEE/UnB _____
Suplente

FICHA CATALOGRÁFICA

CALDAS, EDÍLSON DE OLIVEIRA

PROPOSTA DE SOLUÇÃO E ARQUITETURA PARA AVALIAÇÃO DE CONDUÇÃO SEGURA DE USUÁRIOS DE VEÍCULOS AUTOMOTORES PARA BENEFÍCIOS EM SEGUROS VEICULARES [Distrito Federal] 2024.

xvi, 80 p., 210 x 297 mm (ENE/FT/UnB, Mestre, Engenharia Elétrica, 2024).

Dissertação de Mestrado Profissional - Universidade de Brasília, Faculdade de Tecnologia.

Departamento de Engenharia Elétrica

1. Avaliação de condução

2. Segurança no trânsito

3. Seguro veicular

4. Arquitetura de Aplicativos

I. ENE/FT/UnB

II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

CALDAS, E.O. (2024). *PROPOSTA DE SOLUÇÃO E ARQUITETURA PARA AVALIAÇÃO DE CONDUÇÃO SEGURA DE USUÁRIOS DE VEÍCULOS AUTOMOTORES PARA BENEFÍCIOS EM SEGUROS VEICULARES*. Dissertação de Mestrado Profissional, Departamento de Engenharia Elétrica, Publicação PPEE.MP.069, Universidade de Brasília, Brasília, DF, 80 p.

CESSÃO DE DIREITOS

AUTOR:

TÍTULO: PROPOSTA DE SOLUÇÃO E ARQUITETURA PARA AVALIAÇÃO DE CONDUÇÃO SEGURA DE USUÁRIOS DE VEÍCULOS AUTOMOTORES PARA BENEFÍCIOS EM SEGUROS VEICULARES

GRAU: Mestre em Engenharia Elétrica ANO: 2024

É concedida à Universidade de Brasília permissão para reproduzir cópias desta Dissertação de Mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. Do mesmo modo, a Universidade de Brasília tem permissão para divulgar este documento em biblioteca virtual, em formato que permita o acesso via redes de comunicação e a reprodução de cópias, desde que protegida a integridade do conteúdo dessas cópias e proibido o acesso a partes isoladas desse conteúdo. O autor reserva outros direitos de publicação e nenhuma parte deste documento pode ser reproduzida sem a autorização por escrito do autor.

Depto. de Engenharia Elétrica (ENE) - FT

Universidade de Brasília (UnB)

Campus Darcy Ribeiro

CEP 70919-970 - Brasília - DF - Brasil

DEDICATÓRIA

Dedico a minha esposa Alline César Caldas, que entendeu a importância particular que a formação em Mestre traria para a minha carreira profissional e também uma realização pessoal. Pelo impulsionamento para mais essa conquista em minha vida e também pelo perdão ao tempo que deixei de dedicar a nossa família para intensificar pesquisas e estudos para o sucesso da dissertação do tema.

Dedico as minhas filhas Ana Clara César e Giulia César Caldas, por entenderem a falta de atenção plena nos momentos em que me dedicava a pesquisa e escrita deste trabalho.

AGRADECIMENTOS

Agradeço a Deus por ter me dado saúde e persistência em buscar um sonho em ser Mestre e poder associar novas competências tecnológicas para o avanço em minha profissão e poder contribuir com a sociedade promovendo a inovação, educação e responsabilidade social.

Ao Professor Drº Fábio Lúcio Lopes de Mendonça, por ter apostado que minha tese teria um espaço significativo para desenvolvimento e aplicação junto ao meio acadêmico e a sociedade em geral, me levando com total orientação, paciência e aproveitando o máximo os meus conhecimentos anteriores..

Aos Professores PhD Rafael Timóteo e Dr Georges Daniel Amvame Nze por terem acreditado no meu potencial e vontade em desenvolver-se em um tema tecnológico totalmente novo a minha pessoa, a Redes de Comunicação.

Ao meu amigo Gerson do Nascimento Silva, PhD Student, que me incentivou e não permitiu em nenhum momento que desanimasse, visto que era difícil a divisão entre trabalho, família e pesquisa.

RESUMO

Neste trabalho, apresentamos uma proposta de solução e arquitetura para avaliação de condução segura de usuários de veículos automotores, com foco em benefícios em seguros veiculares, destacando a importância de promover a segurança no trânsito e reduzir acidentes, por meio de tecnologia de aplicativos para coletar dados do veículo e de smartfone do condutor. Esses dados seriam processados para avaliar a segurança da condução, fornecendo feedback direto ao condutor e possibilitando a obtenção de bônus em seguros veiculares, melhores condições de venda de veículos e defesas em possíveis crimes de trânsito. A discussão se estende ao potencial impacto positivo dessa iniciativa, não apenas na redução de acidentes, mas também na melhoria dos serviços de seguros veiculares e na fidelização de clientes e na relevância de investir em tecnologias e soluções inovadoras para promover a segurança no trânsito e melhorar a experiência dos condutores.

ABSTRACT

In this paper, we present a proposed solution and architecture for evaluating the safe driving of motor vehicle users, with a focus on vehicle insurance benefits, highlighting the importance of promoting traffic safety and reducing accidents, using application technology to collect data from the vehicle and the driver's smartphone. This data would be processed to assess driving safety, providing direct feedback to the driver and making it possible to obtain vehicle insurance bonuses, better conditions for selling vehicles and defenses in possible traffic crimes. The discussion extends to the potential positive impact of this initiative, not only in reducing accidents, but also in improving vehicle insurance services and customer loyalty, and the relevance of investing in innovative technologies and solutions to promote traffic safety and improve the driver experience.

SUMÁRIO

1	INTRODUÇÃO	1
1.1	MOTIVAÇÃO	2
1.2	OBJETIVO DO TRABALHO	3
1.2.1	OBJETIVO GERAL	3
1.2.2	OBJETIVO ESPECÍFICO	3
1.3	TRABALHOS PUBLICADOS	3
1.4	METODOLOGIA	4
1.5	ORGANIZAÇÃO DO TRABALHO	4
2	FUNDAMENTAÇÃO TEÓRICA E TRABALHOS RELACIONADOS	5
2.1	TECNOLOGIAS DE CONECTIVIDADE UTILIZADAS EM AUTOMÓVEIS	5
2.1.1	RÁDIOS AUTOMOTIVOS	5
2.1.2	CENTRAIS MULTIMÍDIA	6
2.2	TECNOLOGIAS DE TELEFONIA CELULAR	8
2.2.1	FUNCIONAMENTO DA TELEFONIA CELULAR	8
2.2.2	TECNOLOGIA DE TRANSMISSÃO DA TELEFONIA CELULAR	8
2.3	TECNOLOGIAS DE DESENVOLVIMENTO DE APPS	13
2.3.1	DIVERSIFICAÇÃO DAS PLATAFORMAS	13
2.3.2	CONSTRUÇÃO DE APP	15
2.3.3	TRABALHOS RELACIONADOS	23
3	ESTRUTURA DA PROPOSTA DO APP PARA AVALIAÇÃO DE CONDUÇÃO SE- GURA	25
3.1	ANÁLISE INICIAL	25
3.1.1	IDENTIFICAR A LACUNA NO MERCADO	27
3.1.2	PESQUISA DE MERCADO	32
3.1.3	ANÁLISE DE SOLUÇÕES	34
3.1.4	DEFINIR PÚBLICO-ALVO E PERSONAS	35
3.2	LEVANTAMENTO DE REQUISITOS	35
3.2.1	REALIZAR WORKSHOPS OU REUNIÕES PARA CAPTURAR REQUISITOS	37
3.2.2	CRIAR DOCUMENTOS DETALHADOS DE REQUISITOS	38
3.2.3	PRIORIZAR FUNCIONALIDADES E RECURSOS	39
3.3	ARQUITETURA E DESIGN	40
3.3.1	DEFINIR A ARQUITETURA DO APLICATIVO	41
3.3.2	CRIAR PROTÓTIPOS DE BAIXA FIDELIDADE PARA ESBOÇAR A INTERFACE	42
3.3.3	DESENVOLVIMENTO FRONTEND	43
3.3.4	DESENVOLVER UMA INTERFACE DO USUÁRIO (UI)	44
3.3.5	IMPLEMENTAR O DESIGN VISUAL E INTERAÇÕES DO USUÁRIO	45

3.4	DESENVOLVIMENTO BACKEND	47
3.4.1	PARÂMETROS DE PROCESSAMENTOS DE ANOMALIAS	48
3.4.2	DESENVOLVER A LÓGICA DE NEGÓCIOS E FUNCIONALIDADES PRINCIPAIS	52
3.4.3	IMPLEMENTAR A INTEGRAÇÃO COM SERVIÇOS EXTERNOS, SE NECESSÁRIO	53
3.4.4	CONFIGURAR O SERVIDOR E BANCO DE DADOS	57
3.5	POLIMENTO E REFINAMENTO	59
3.5.1	REFINAR O CÓDIGO E A INTERFACE COM BASE NOS TESTES DE USABILIDADE	60
3.5.2	OTIMIZAR O DESEMPENHO E A SEGURANÇA DO APLICATIVO	60
3.6	IMPLANTAÇÃO E LANÇAMENTO	60
3.6.1	PREPARAR A VERSÃO FINAL PARA PUBLICAÇÃO NA GOOGLE PLAY STORE	61
3.6.2	CONFIGURAR DESCRIÇÃO, IMAGENS E METADADOS NA LOJA	62
3.6.3	LANÇAR O APLICATIVO	63
3.7	MONITORAMENTO PÓS-LANÇAMENTO	63
3.7.1	MONITORAR MÉTRICAS DE DESEMPENHO	63
3.7.2	COLETAR FEEDBACKS E BUGS REPORTADOS PELOS USUÁRIOS	66
3.7.3	PLANEJAR ATUALIZAÇÕES E MELHORIAS CONTÍNUAS	66
4	EXPERIMENTO E RESULTADOS	68
4.1	TESTES UNITÁRIOS E DE INTEGRAÇÃO	68
4.1.1	TESTAR A INTEGRAÇÃO ENTRE FRONTEND E BACKEND	69
4.1.2	CORRIGIR BUGS E PROBLEMAS IDENTIFICADOS	70
4.2	TESTES DE USABILIDADE	71
4.2.1	REALIZAR TESTES DE USABILIDADE COM USUÁRIOS REAIS OU GRUPOS DE FOCO	71
4.2.2	COLETAR FEEDBACKS E REALIZAR AJUSTES NA INTERFACE E FUNCIONALIDADES	71
4.3	RESULTADOS ESPERADOS DA IMPLEMENTAÇÃO DO APLICATIVO DE MONITORA- MENTO DE CONDUTORES	72
5	CONCLUSÃO	74
	REFERÊNCIAS BIBLIOGRÁFICAS	76
	APÊNDICES	80

LISTA DE FIGURAS

2.1	Primeiro rádio desenvolvido e instalado por George Frost.	6
2.2	Ilustração da instalação do rádio automotivo.	6
2.3	DVD com tela e entrada para pendrive.	7
2.4	Central Multimídia.	7
2.5	ERB e redes celulares conectadas.	9
2.6	Representação da rede de telefonia em funcionamento.	9
2.7	Tecnologia 2G - cdmaOne.....	10
2.8	Funcionamento da rede com Tecnologia 4G.	11
2.9	Possibilidades que a Tecnologia 5G pode proporcionar.	11
2.10	Arquitetura de funcionamento da rede com Tecnologia 5G.	12
2.11	Fluxo básico de requisitos.	16
2.12	Exemplo de interface com o usuário.	17
2.13	Diferença de Sintaxe entre o Java e o Kotlin.	18
2.14	Interface do Android Studio.	19
2.15	Execução no console.....	20
2.16	Interface do Usuário da tecnologia Django.....	21
2.17	Interface do Usuário do Rails.....	22
2.18	Mapa de funcionalidades para um APP.	22
2.19	Mensagem de rejeição de APP pela PLayer Store Google.....	23
3.1	Estrutura da proposta do desenvolvimento do APP - parte 1	25
3.2	Estrutura da proposta do desenvolvimento do APP - parte 2	26
3.3	Fluxograma de respostas da pesquisa de lacuna de mercado	28
3.4	Persona 1 e 2.....	36
3.5	Público-Alvo	36
3.6	Levantamento de Requisitos	37
3.7	Matriz de priorização MOSCOW	40
3.8	Arquitetura Básica	41
3.9	Arquitetura detalhada da proposta do Aplicativo	42
3.10	Proposta de front end da plataforma Safe Driving.....	45
3.11	Telas de Interface do usuário	46
3.12	Gráfico de Aceleração bruscas	51
3.13	Risco de alta velocidade em curvas	52
3.14	Tempo de parada obrigatório	52

LISTA DE TABELAS

3.1	Pesquisa de Lacuna de Mercado - Gênero.....	28
3.2	Pesquisa de Lacuna de Mercado - Faixa etária.....	28
3.3	Pesquisa de Lacuna de Mercado - Faixa etária + Gênero Feminino	29
3.4	Pesquisa de Lacuna de Mercado - Faixa etária + Gênero Masculino	29
3.5	Pesquisa de Lacuna de Mercado - Posse de veículos Automotor	29
3.6	Pesquisa de Lacuna de Mercado - Posse de veículos + Gênero	29
3.7	Pesquisa de Lacuna de Mercado - Seguro Veicular	30
3.8	Pesquisa de Lacuna de Mercado - Seguro Veicular + Gênero	30
3.9	Pesquisa de Lacuna de Mercado - Uso de Apps ou programa de benefícios.....	30
3.10	Pesquisa de Lacuna de Mercado - Uso de Apps ou programa de benefícios + Gênero.....	30
3.11	Pesquisa de Lacuna de Mercado - Uso da proposta de App.....	31
3.12	Pesquisa de Lacuna de Mercado - Uso da proposta de App + Gênero	31
3.13	Pesquisa de Lacuna de Mercado - Consideraria o uso da proposta de App	31
3.14	Pesquisa de Lacuna de Mercado - Consideraria o uso da proposta de App + Gênero	32
3.15	Pesquisa de Lacuna de Mercado - Usaria mesmo que sabendo que ele está avaliando sua forma de dirigir	32
3.16	Pesquisa de Lacuna de Mercado - Usaria mesmo que sabendo que ele está avaliando sua forma de dirigir + Gênero	32
3.17	Pesquisa de Mercado	33
3.18	Tecnologias escolhidas para futuro desenvolvimento.....	34
3.19	Algoritmo 1 - parâmetros para Anomalias de condução	49
3.20	Algoritmo 2 – Parâmetros de bônus de Score	49
3.21	Algoritmo 3 – Parâmetros pertinentes a cada anomalia	50
3.22	Anomalias de condução.....	53

1 INTRODUÇÃO

Ao longo dos anos, diversas aplicações foram desenvolvidas para penalizar os condutores menos educados e estão implementadas nas cidades conectadas, como radares com câmeras inteligentes para identificar o comportamento dos condutores ao dirigirem seus veículos, aplicando multas com valores significativos e perda de pontos em suas habilitações. Esse é um caminho punitivo e em algumas situações se faz necessário, porém não é educativo, pois o condutor irá se familiarizar com o posicionamento dessas tecnologias e irá agir de forma aparentemente responsável nas imediações das instalações destas aplicações. Para sermos efetivos e termos resultados consideráveis, precisamos que os condutores vivenciem um processo de educação contínua no trânsito, onde perceba vantagens aplicadas a ele e a todo ecossistema, proporcionando a autocobrança em ser um condutor responsável e seguro durante o uso de seu automóvel. Através de nossa proposta de arquitetura de aplicação, focada em avaliar as ações dos condutores, podendo pontuar ou penalizar de acordo a sua dirigibilidade, extraindo esses dados do Smartphone e do sistema de controle eletrônico do veículo, o SAFE DRIVING, a educação contínua se torna possível, visto que os veículos automotores ficaram mais modernos e com mais tecnologias embarcadas, com diversas interfaces de conectividade entre os usuários e o veículo por meio de seus smartphones.

Essa evolução levou os veículos a serem comparados a um smartphones sobre rodas e por meio dessa interatividade entre os veículos e smartphones, tornou possível que os usuários podasse escolher o caminho mais rápido para um determinado local e até mesmo agendar uma revisão preventiva por meio de seu veículo em uma concessionária mais próxima ou de sua preferência. O entretenimento também foi uma das opções relevantes ao conectar os smartphones aos veículos, pois permitiu acessos a músicas, mensagens, ligações e diversas outras funcionalidades sem que o usuário encoste no aparelho. Porém sabemos que o próprio smartphone é um dos principais causadores de acidentes no trânsito brasileiro, pois proporciona distração ao conduzir o veículo, uma vez que, o usuário deixa de interagir com o ecossistema do trânsito de forma segura. Isso acontece quando a tecnologia de seu veículo ou de seu smartphone não são compatíveis com as necessidades de segurança observadas durante a utilização do veículo. Quando atendem essas necessidades, entra o modelo mental do condutor que ainda não se adaptou aos novos padrões tecnológicos, causando acidentes, muitos deles fatais. Algumas das interfaces disponíveis em veículos ou smartphones já são programadas para não permitir a operação da aplicação quando o veículo estiver em movimento, mas outras aplicações estão sobrepostas a essas condições, como aplicativos de mensagens, por exemplo. Também sabemos que inúmeros fatores na condução podem causar acidentes, como excesso de velocidade, consumo de álcool e até mesmo a não observação de pausa para descanso ao estar em um tempo maior que quatro horas de condução.

De acordo com a Confederação Nacional dos Transportes (1), no período de 2007 a 2022 foram registrados 1.982.059 acidentes de trânsito nas rodovias federais brasileiras (1) por diversos motivos com números alarmantes acumulados de vítimas (mortos e feridos), no total são 970.674 pessoas, o que nos dá uma média de 7 acidentes a cada 10 km. Esse número pode ficar ainda maior, pois diversos acidentes, com ou sem vítimas, não são registrados por estarem em ruas e avenidas não monitoradas pelos órgãos competentes, trazendo uma falsa sensação de redução de acidentes ou melhoria de perspectiva da segurança

nas estradas. Fato é que cada acidente gerado, representa um custo elevado aos cofres do estado, seja com reabilitação das vítimas, indenizações ou a recuperação de infraestrutura das vias. Esses custos estão na ordem de R\$ 250,26 milhões (1) durante o período. Além da população em geral e o Governo estarem interessados na redução destes números de acidentes, existem outros atores que atuam com ações preventivas, pois afetam diretamente o seu negócio. Um desses atores é a Companhia de Seguros (CS), pois cada acidente gerado, seja ele de qualquer grandeza, significa um desembolso em seus cofres e redução de sua lucratividade.

Um dos principais desafios que as CS tem enfrentado é o desenho de uma persona ideal de seu cliente, persona que não ocasiona acidentes de trânsito e que realmente seja o perfil declarado no momento da aquisição das apólices de seguros para seus veículos. Para uma melhor definição e não muito assertiva, as CS normalmente tem alguns estereótipos traçados a partir de estatísticas de acidentes de trânsito e precificam seus serviços conforme esses dados, adotando um contexto de classificação de grupos, o que foge a grande maioria de seus clientes, principalmente no que tange a acidentes de pequena monta (2). Como exemplo, podemos trazer a faixa etária, onde as CS aplicam o maior custo aos clientes com idade entre 18 e 25 anos. Os estudos realizados anualmente pela Confederação Nacional dos Transporte (1) apontam divergências sobre esses conceitos, pois são os clientes com idade maiores que 45 anos que mais se envolvem em acidentes no Brasil, totalizando 707.596 acidentes no período entre 2007 e 2022 e são o clientes onde a CS tem a menor aplicação de custos nas apólices de seguros veiculares.

1.1 MOTIVAÇÃO

Em meio a este cenário, encontramos a oportunidade de gerar educação no trânsito através dos aplicativos (App). Por sua vez, os App's vêm ganhando cada vez mais popularidade entre os usuários de Smartphone. Mais de 78% de usuários utilizaram algum tipo de aplicativo em 2022 (3). Eles se tornaram aliados na execução de atividades diárias com excelência e total rastreabilidade comparativas. Os App permitem aos usuários saberem se realizaram uma tarefa da melhor forma que no dia ou mês anterior, seja uma tarefa pessoal, profissional ou até mesmo um olhar a distância para membros da família. Em meio a toda a facilidade, outros App's surgiram para controlar gastos, ampliar rendimentos e monitorar scores referente a compras e pagamentos realizados por pessoas físicas, possibilitando as Instituições financeiras (IF) avaliar com velocidade o perfil do usuário, permitindo tomadas de decisão sem burocracias, como a cessão de um empréstimo, por exemplo.

O Desenvolvimento de uma arquitetura para criação de um App pode ser uma solução vantajosa para quem gostaria de se apresentar como um condutor responsável e seguro, além de ter benefícios ao compartilhar suas ações e dados, em consonância com A Lei Geral de Proteção de Dados Pessoais (LGPD), Lei nº 13.709/2018 (4), com CS para adquirir prêmios de seguros.

Para as CS, seria uma forma de atrair mais clientes e realizar cotações de apólices de forma justa e personalizada, pois deixaria de atuar com um histórico estatístico para atuar com dados pessoais de condução. As CS também conseguiriam melhorar sua lucratividade, pois gastariam menos recursos com reparações de veículos e indenizações por perda total, possibilitando melhores prêmios de seguros e fidelizando seu

cliente a sua marca.

A possibilidade de redução de acidentes nas cidades e estradas brasileiras é um dos maiores motivadores a serem considerados, pois são centenas de milhares de vidas perdidas no trânsito, o que além de causar prejuízos físicos e financeiros, também destroem vidas promissoras e famílias envolvidas direta e indiretamente nessas fatalidades, fatalidades que poderiam ser evitadas por meio de conscientização e principalmente educação no trânsito.

1.2 OBJETIVO DO TRABALHO

1.2.1 Objetivo Geral

Neste Trabalho, o objetivo principal é desenvolver e testar uma arquitetura para criação de um aplicativo denominado *SAFE DRIVING* que busca combinar dados dos Smartphones por meio de acelerômetros, relógio, mapas e giroscópios com os dados dos veículos extraído por meio de sua tomada On-board diagnósticos (OBD-II) como rotação de motor, controle de estabilidade, controle de tração, anti-bloqueio de freios, velocidade do veículo, tempo de aceleração e frenagem, conforme em (5). A combinação desses dados, permitirá a detecção de um comportamento agressivo ou não de um motorista de forma eficiente, oportunizando a sua segurança e a de todo ecossistema do trânsito.

1.2.2 Objetivo Específico

Para chegar no objetivo geral da proposta, foram elencados os seguintes objetivos específicos:

- Estudar métodos de Coleta dados de funcionamento do veículo e dados dos sensores do dispositivo Android do usuário;
- Elaborar a especificação de gateway de aplicativo móvel para receber dados coletados e transmitir para servidor *SAFE DRIVING* utilizando protocolo Hypertext Transfer Protocol Secure;
- Especificar inferências, nas camadas de aplicação e do servidor;
- Propor arquitetura de funcionamento do aplicativo *SAFE DRIVING*.

1.3 TRABALHOS PUBLICADOS

Durante o estudo para construção desta dissertação, um artigo foi publicado referente ao tema de Avaliação de Condução Segura de Veículos Automotores.

- APLICATIVO PARA AVALIAÇÃO DE CONDUÇÃO SEGURA DE USUÁRIOS DE VEÍCULOS AUTOMOTORES POR MEIO DE INTELIGÊNCIA ARTIFICIAL PARA BENEFÍCIOS EM SEGUROS VEICULARES

1.4 METODOLOGIA

Para alcançar os objetivos desta dissertação, na fase inicial fizemos a identificação e caracterização do problema, além do levantamento das hipóteses de solução por meio de desenvolvimento de arquitetura para criação de App. Em seguida foi realizada a verificação de infraestrutura e equipamentos que possibilitassem a aquisição e o processamento de dados obtidos em pesquisas com usuários voluntários.

Com os dados da pesquisa coletados e a personificação de cada usuário voluntário, foi realizado um feedback coletivo para que todos pudessem opinar sobre os resultados apontados, bem como as experiências e comportamentos assumidos após a utilização do App.

Após todas as informações fornecidas por possíveis usuários compiladas, foi especificado a infraestrutura de funcionamento, bem como a arquitetura proposta para o funcionamento do App.

1.5 ORGANIZAÇÃO DO TRABALHO

Este trabalho é composto por cinco capítulos, incluindo este primeiro de introdução.

O segundo capítulo trata das tecnologias e técnicas utilizadas para realizar as pesquisas, desenvolvimento e testes da arquitetura para criação do App em questão. Aqui traremos todos os recursos utilizados, bem como os seus avanços tecnológicos e seu atual funcionamento durante a aplicação neste trabalho.

O terceiro capítulo apresenta a arquitetura proposta, as instalações e configurações a serem realizadas, a disposição dos ativos de rede e por fim a arquitetura final.

O quarto capítulo apresenta as propostas os testes serem realizadas pelos usuários voluntários, também serão apresentadas as diferentes funcionalidades utilizadas nos testes, bem como as análises dos resultados obtidos e experiências e comportamentos expressados por eles.

O quinto capítulo conclui este trabalho trazendo a consolidação dos resultados e a validação da proposta, além da indicação de trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA E TRABALHOS RELACIONADOS

Neste capítulo estão presentes o conjunto de soluções e utilizadas para alcançar os objetivos da proposta. Abordamos as tecnologias de conectividade utilizadas em automóveis, como rádios automotivos, rádios FM, rádios com reprodução de mídias físicas e centrais multimídia. Além disso, também são discutidas as tecnologias de telefonia celular, o funcionamento da telefonia celular e a tecnologia de transmissão da telefonia celular. Outro ponto abordado é o desenvolvimento de aplicativos, incluindo a diversificação das plataformas, a construção de aplicativos e trabalhos relacionados. Essa seção do documento fornece uma base teórica e contextualiza as tecnologias que serão utilizadas na proposta de solução e arquitetura para avaliação de condução segura de usuários de veículos automotores para benefícios em seguros veiculares.

2.1 TECNOLOGIAS DE CONECTIVIDADE UTILIZADAS EM AUTOMÓVEIS

Se pensarmos nos automóveis atuais e observarmos suas tecnologias, é fácil compará-los a um smartfone sobre rodas, pois a conectividade atual permite uma grande interação entre o condutor e o veículo através de seu aparelho Smartfone. Mas se você ler novamente este trabalho daqui a 10 (dez) anos, perceberá que as tecnologias aqui relatadas estão ultrapassadas e talvez seja até mesmo parte de sua aula de história da tecnologia. Então vamos a um breve histórico para contextualizar as tecnologias de conectividade utilizada nos automóveis.

2.1.1 Rádios Automotivos

A conectividade não seria possível nos dias de hoje, se um jovem de 18 anos chamado de George Frost instalasse o primeiro rádio em seu veículo Ford modelo T, no dia 5 de maio de 1922, conforme (6) O gesto histórico em que ele deu partida no veículo e ligou o rádio, e saiu andando pelas ruas de Nova York ficou marcado no mundo automotivo. Frost era o presidente de um rádio clube e com sua genialidade e conhecimento inventou o então modelo de rádio aplicado em seu veículo. O objetivo de sua instalação foi ouvir músicas e notícias enquanto conduzia seu veículo, para que as suas viagens solitárias não ficassem tão monótonas. Os rádios, de maneira geral, ainda estavam em desenvolvimento e não eram tão compactos como conhecemos nos dias de hoje. A figura 2.1 mostra os primeiros rádio utilizado no automóvel Ford modelo T.

Rapidamente, os rádios ganharam a Europa e a Mercedes-Benz adotou a instalação dos rádios em seus veículos. O jornal alemão Berliner Illustrierte Zeitungm relatou em um artigo escrito em agosto de 1922, que a moda mais recente na América era a utilização de rádios automotivos. O artigo trazia imagens de



Figura 2.1: Primeiro rádio desenvolvido e instalado por George Frost.
Fonte: getty images

instalação do rádio bem como sua antena passando pelo para-brisa. Na figura 2.2 ilustra as imagens publicadas no artigo.



Figura 2.2: Ilustração da instalação do rádio automotivo.
Fonte: Portal Terra

2.1.2 Centrais Multimídia

Conforme as mídias iriam evoluindo, também o auto-radio acompanhava para trazer a melhor experiência do usuário. nos anos 2000, os rádios já permitiam a inserção de pendrive para executar arquivos de áudio em MP-3, mesmo junto aos CDs e posteriormente os auto rádios evoluíram para a leitura de DVDs com pequenas telas de execução de vídeo, o que tornou as possibilidades do usuário quase infinitas para a época. Na figura 2.3 é ilustrada o auto-radio compatível com DVD e pendrive.

Com criação das centrais multimídias nos veículos, foi possível evoluir para conexão de smartphones, neste momento a experiência de ouvir musica se tornava infinita, pois os dispositivos permitiam o acesso



Figura 2.3: DVD com tela e entrada para pendrive.

Fonte: Sony

as músicas na rede de internet ou mesmo que em seus armazenamentos internos, que por sua vez era muito maior que os DVDs (7). A conexão bluetooth permitiram além de tocar músicas , toca de dados entre o Smartfone e a central multimídia, o que levou a utilização de aplicativos como de Navegação, mensagens e ligações (8). A figura 2.4 demonstra uma central multimídia de um veículo.



Figura 2.4: Central Multimídia.

Fonte: Extra online

A evolução dos rádios até a centrais multimídias permitiram a conexão direta do condutor com veículo por meio de seu smartfone, possibilitando infinitas oportunidades de desenvolvimento de aplicações e tecnologias de apoio a condução segura e interativa (9).

2.2 TECNOLOGIAS DE TELEFONIA CELULAR

As tecnologias da telefonia celular estão em constante evolução, todas as fabricantes buscam atender as mais variadas necessidades de seus clientes, seja por meio de espaços de armazenamentos, concessões de aplicativos, melhores fotos e vídeos. Nos dias atuais o que menos importa é o objetivo principal de seu surgimento, a ligação telefônica (10).

O início das pesquisas voltadas a telefonia móvel foi na União Soviética em 1935, quando o cientista e professor Dimitriy Vasiliyevich Ageev defende uma dissertação sobre recepção de rádio por divisão de códigos. Esse foi intitulado como “Os Métodos para Tratamento de Ruído na Recepção de Rádio”, o artigo, apresentado no Instituto de Comunicações Eletrotécnicas de Leningrado, descrevia os procedimentos-base para o que hoje se conhece como tecnologia de Código de Divisão de Acesso Múltiplo ou do inglês Code Division Multiple Access (CDMA).

2.2.1 Funcionamento da telefonia celular

A telefonia celulares funciona enviando e recebendo sinais de raio de baixa potência, ou seja, os sinais são enviados e recebidos a partir de estações rádio-base (ERB) conectadas a transmissores e receptores de rádio, essas são popularmente conhecidas como antenas. As ERB são conectadas às redes de telefonia móvel e fixa, por ali, passam o sinal/chamada. Ela oferece cobertura para uma determinada área geográfica denominada “célula”. As células ficam alinhadas umas próximas às outras, semelhante a uma colmeia e é por isso que as redes de telefonia móvel são chamadas de redes “celulares”. A localização da ERB dentro da célula é determinada por uma série de fatores como limitações físicas e topográficas da região, número e altura de árvores e prédios, a capacidade de transmissão das células, número estimado de chamadas e a frequência de rádio dentro da qual a ERB irá operar. Basicamente, o aparelho celular precisa “enxergar” a ERB, ou seja, o sinal de rádio emitido pelo telefone para a ERB deverá ser contínuo. É necessário analisar a localização de instalação da ERB para ampliar o alcance máximo de sua área de cobertura. As Montanhas, árvores e prédios altos podem interferir nessa linha de “visão” e prejudicar o bom funcionamento das redes celulares. Na figura 2.5 ilustramos uma rede celular.

Na prática, ao realizar uma ligação, o celular se comunica com a ERB mais próxima, por sua vez, a ERB encaminha a sua ligação para a Central de Comutação e Controle (CCC) e dependendo do destino da chamada, ela é encaminhada para a ERB ou para o CCC de outra região, assim o usuário emissor que realizou a chamada irá se comunicar com o usuário receptor desejado por meio de seu celular que vai receber a ligação. A figura 2.6 representa a rede celular interligada com a rede de telefonia fixa e a central de comutação e controle em funcionamento.

2.2.2 Tecnologia de transmissão da telefonia celular

Em 1983, William Lee, conhecido engenheiro da AT&T, retomou o pioneirismo para a gigante de telecomunicações e desenvolveu a primeira tecnologia aplicada a transmissão de telefonia celular, a "Tec-

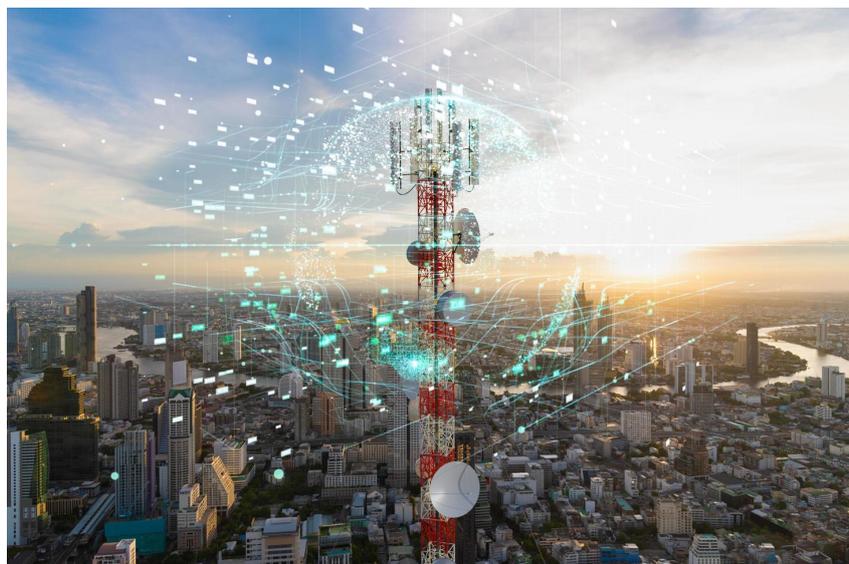


Figura 2.5: ERB e redes celulares conectadas.

Fonte: Vecteezy

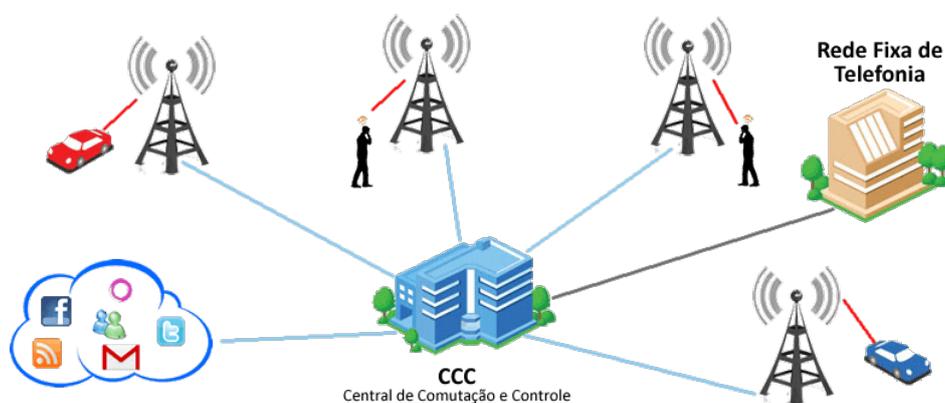


Figura 2.6: Representação da rede de telefonia em funcionamento.

Fonte: davidpiner.com

nologia 1G" ou primeira geração. Ela era formada por sistemas analógicos e só era possível fazer transmissão de voz. No entanto, com a baixa capacidade de tráfego e alto custo, não houve, em um primeiro momento, uma grande popularização da tecnologia, por sua vez, também não houve uma padronização dos sinais, sendo criados diversos padrões diferentes. Com as deficiências encontradas na 1G, foi desenvolvida a Tecnologia 2G ou segunda geração. A segunda geração, foi uma tecnologia difundida na década de 1990 em que o telefone celular começou a ser mais utilizado pela população. O sinal mudou de analógico para um sinal digital com as tecnologias mais utilizadas Code Division Multiple Access (Acesso Múltiplo por Divisão de Código - CDMA) e a Global System for Mobile Communications (Sistema Global para Comunicações Móveis - GSM) (11). A figura 2.7 ilustra o funcionamento da rede cdmaOne.

A tecnologia GSM permitiu a inserção de novas tecnologias em seu protocolo de funcionamento como a General Packet Radio Service (Serviço de Rádio de Pacote Geral - GPRS). Essa tecnologia proporcionou o aumento na velocidade de transferência de dados e permitiu ao usuário se manter sempre conectado à internet. Essa tecnologia ainda é utilizada em algumas regiões do Brasil.

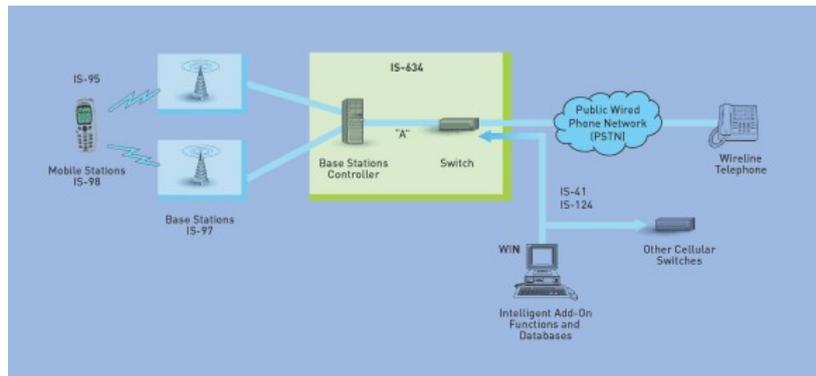


Figura 2.7: Tecnologia 2G - cdmaOne.

Fonte: páginas.fe.up

A Tecnologia 3G ou terceira geração, é a evolução do 2G. Ela veio para implementar novos padrões de comunicações definidos pela União Internacional de Comunicações (ITU). Sua principal característica é o aumento significativo da capacidade de usuários de serviços de voz e dados comparados com tecnologia anterior. Nos dias atuais é a rede em que a maioria dos brasileiros estão conectados. Novos padrões foram inseridos com a 3G.

A evolução tecnológica 4G nos permite avançar no desenvolvimento da arquitetura proposta, bem como os testes e comprovações de nosso trabalho, pois consegue gerar a conectividade necessária para a troca de dados entre o veículo e o smartfone e posteriormente, entre o smartfone e o servidor.

A tecnologia de 4ª geração trouxe o Long Term Evolution (Evolução de Longo Prazo - LTE). Esse padrão é para que as operadoras que utilizam o GSM e o WCDMA estão se adaptando pois esta nova tecnologia de rádio permite velocidades de 150 Mbps de downlink e 50 Mbps de uplink com taxas máximas. O LTE tem compatibilidade com o GSM e HSPA, o que significa que ao mudarem para esta tecnologia não precisam descontinuar a tecnologia anterior.

O 4G está baseada totalmente em IP, sendo um sistema e uma rede, alcançando a convergência entre as redes de cabo e sem fio e computadores, dispositivos eletrônicos e tecnologias da informação para prover velocidades de acesso entre 100 Megabit/s em movimento e 1 Gigabit/s em repouso, mantendo uma qualidade de serviço (QoS) de ponta a ponta (ponto-a-ponto) de alta segurança para permitir oferecer serviços de qualquer tipo, regularmente e independentemente da localização. A figura 2.8 ilustra o funcionamento da rede 4G.

Com o crescimento do mercado de tecnologia, a tecnologia 5G veio para revolucionar a rede de dados e trazer soluções de conectividade. O objetivo é expandir a rede de conexão móvel para o máximo de dispositivos possíveis. Como exemplo, temos os carros, eletrodomésticos, telemedicina, agricultura, educação e nas demais áreas da Internet of Things (Internet das Coisas - IoT). A rede 5G utilizará um espectro de onda maior do que as gerações de internet móvel anteriores, ocupando as frequências entre 600 MHz e 700 MHz, 26 GHz e 28 GHz, 38 GHz e 42 GHz com latência de até 1 ms com velocidades de até 1Gb/s. Com a utilização dessa tecnologia, a recepção e envio de dados se torna mais rápida e precisa, o que aponta para uma evolução do trabalho realizado. Na figura 2.9 estão ilustradas as possibilidades proporcionadas pela tecnologia 5G (12).

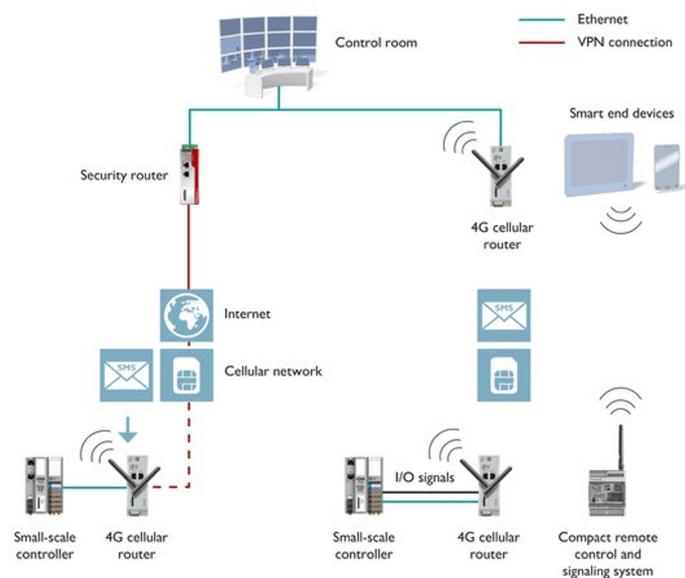


Figura 2.8: Funcionamento da rede com Tecnologia 4G.

Fonte: Phoenix Contact



Figura 2.9: Possibilidades que a Tecnologia 5G pode proporcionar.

Fonte: Sindinfor

O 5G funciona por meio de ERB que emitem ondas de rádio que para se conectar com os dispositivos móveis. Um núcleo de rede (core), uma espécie de servidor, gerencia todas as conexões antes de entregar o tráfego para a internet. Os ERBs de 5G podem ser instalados nas mesmas estações utilizadas pelo 4G,

mas, a tecnologia exige maior número de ERB por conta da utilização de espectros mais altos, pois quanto maior a frequência, menor o alcance de sinal. Assim os ERBs se comunicam com o núcleo de rede através de fibra óptica, radiofrequências ou até mesmo via satélite, dependendo da região e do projeto de cada operadora. Na figura 2.10 estão ilustradas a arquitetura de funcionamento da tecnologia 5G.

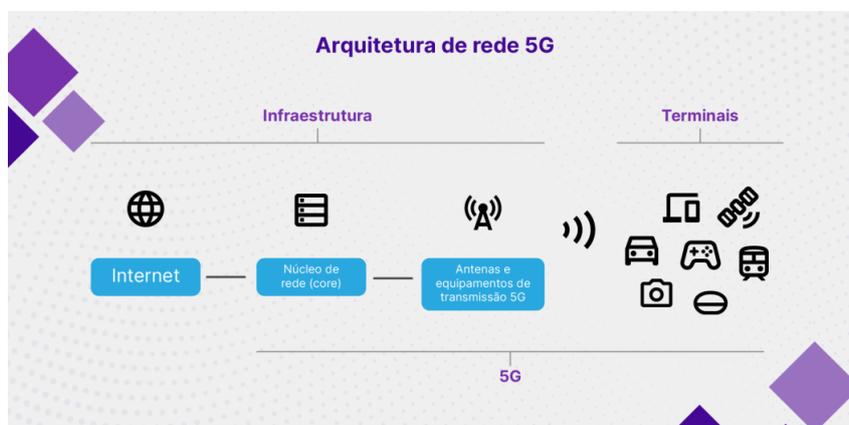


Figura 2.10: Arquitetura de funcionamento da rede com Tecnologia 5G.

Fonte: Tecnoblog

Por trás do 5G, existem diferentes recursos de rede, que podem ser utilizados dependendo do objetivo de cada operadora. Os principais recursos são:

- Enhanced Mobile Broadband (Banda larga móvel melhorada - eMBB) - Esse recurso permite a maior largura de banda disponível para usuários, com velocidades de download que podem ultrapassar a casa de 1 Gb/s;
- Ultra-Reliable Low Latency Communications (Comunicações ultra fiáveis de baixa latência - URLLC) - Esse recurso garante menor latência na conexão com os dispositivos, com maior confiabilidade da rede para casos de usos sensíveis a atrasos, como operação de máquinas, cirurgias remotas, veículos autônomos e streaming de jogos;
- Massive Machine Type Communications (Comunicações de tipo máquina massiva - mMTC) - Esse recurso permite o suporte a grandes quantidades de dispositivos de Internet das Coisas com consumo de energia reduzido, como sensores, medidores e aplicações para cidades inteligentes;
- Network Slicing (Fatiamento de redes) - Esse recurso permite dividir o 5G em várias sub-redes, permitindo utilização industrial e privativa no mesmo espectro de uma rede comercial com diferentes níveis de prioridade e qualidade de serviço.

A tecnologia de 5ª geração possui três classificações diferentes quanto a disponibilidade de serviços, a 5G Standalone (SA), a 5G Non-Standalone (NSA) e a 5G compartilhamento dinâmico de espectros (DSS) que são detalhadas a seguir:

- 5G Standalone - A rede 5G SA é uma arquitetura nativa do 5G. Ela possui frequência dedicada e núcleo de rede independente de tecnologias anteriores, desenvolvida desde o início com base nos

padrões do 5G. Ela não depende de elementos criados para o 4G, permitindo uma arquitetura totalmente independente e otimizada para a nova geração;

- 5G Non-Standalone - A rede 5G NSA depende de uma implementação que utiliza elementos da rede 4G existente. Ela possui frequência dedicada, mas compartilha o mesmo núcleo de rede da tecnologia 4G. Isso faz com que ela dependa das infraestruturas do 4G para funcionar, com o 5G sendo utilizado principalmente para fornecer velocidade mais rápida e menor latência, mas sem todo o potencial do 5G SA;
- 5G DSS - a rede 5G DSS utiliza compartilhamento dinâmico de espectro é, de fato, outra tecnologia, com padrão 5G NR (New Radio) em formato NSA (non-standalone). Isso significa que o 5G DSS é uma nova tecnologia, que se enquadra como 5G NR nos padrões da 3GPP e depende do core existente do 4G LTE para funcionar, embora tenha portadoras dedicadas para a tecnologia (assim como qualquer 5G NSA, independente de compartilhamento de espectro) ela é uma rede 5G que não tem espectro dedicado para funcionar, e compartilha a capacidade do 4G, 3G e/ou 2G.

2.3 TECNOLOGIAS DE DESENVOLVIMENTO DE APPS

Com a necessidade de se criar novas formas de interação entre os usuários e os smartphones, foram sendo desenvolvidos aplicações compatíveis com cada sistema operacional aplicado por cada fabricante de smartphones. Com a inteligência aplicada ao aparelho, oportunidades diferentes de realizar e atender ligações ficaram evidentes e essa deixou de ser a função primária. Hoje, por causa das diversas funcionalidades proporcionadas pelos Apps, os usuários podem realizar uma reunião WEB com pessoas de outro país estando na condição de passageiro de um transporte que também foi chamado por um aplicativo. Nesta seção vamos conhecer as plataformas e as ferramentas que serão importantes para o desenvolvimento da arquitetura proposta para a construção do App **Safe Driving**.

2.3.1 Diversificação das plataformas

As plataformas de aplicativos foram desenvolvidas para fornecer aos usuários a navegação e o download de aplicativos, jogos, filmes disponíveis e muito mais. Sabemos que as duas lojas online mais populares são Google Play para Android e App Store para iOS (13). Porém existem outras lojas que oferecem os mesmos serviços de disponibilização de aplicativos, conforme veremos a seguir:

- Amazon Appstore

Plataforma da Amazon tem o objetivo de relacionar aplicativos conforme o perfil de compras do usuário em função de aquisições anteriores. Ela oferta um sistema de pagamento e como recompensa disponibiliza Amazon Coins, moeda de troca dentro da loja, que gera maior vantagem e grandes descontos para compras a serem realizadas na loja. A Amazon Appstore apresentou mais um recurso “App Gratuito do Dia”, ou seja, um App pago, porém naquele dia específico ele está disponível gratuito para o cliente. Os Apps ofertados pela loja não estão ligados a um sistema operacional Amazon, geralmente são IOS ou Android.

- BlackBerry World

A app store oficial para smartphones e tablets BlackBerry é uma vitrine e serviço de conteúdo multi-mídia onde os usuários podem fazer download de aplicativos disponível em seu dispositivo, jogos e temas são exemplos para uso nos aparelhos. O serviço foi estabelecido em 1º de abril de 2009 pela BlackBerry. Existem grandes lojas de aplicativos de diferentes sistemas operacionais, em 21 de janeiro de 2013, a BlackBerry publicou que reconstruía o BlackBerry App World para simplificar o BlackBerry World. Mas o BlackBerry World é um mercado centralizado, a coleção varia em dispositivos BlackBerry OS 7, dispositivos BlackBerry 10 e BlackBerry PlayBook.

- Firefox Marketplace

Lançado em 1998 por membros do Netscape Communications Group, a organização Mozilla é uma empresa de capital aberto e sem um proprietário específico e tem a Firefox Marketplace como a sua loja oficial de aplicativos para dispositivos Firefox OS. Ela armazena os aplicativos e recursos projetado para a plataforma da web Firefox usando tecnologias, idiomas e ferramentas padrão da Web. A Firefox Marketplace permite que qualquer pessoa publique Open Web Apps. Os produtos da Mozilla incluem o navegador Firefox, sistema operacional móvel Firefox OS, e-mail Mozilla Thunderbird, mecanismo de layout Gecko, ChatZilla, Mozilla Mail & Newsgroups, sistema de rastreamento de bugs Bugzilla, ChatZilla e outros recursos.

- Ovi (Nokia)

A Ovi by Nokia, formalmente conhecido como Nokia Ovi Suite é uma loja de aplicativos para usuários da Nokia utilizada para conectar seus aparelhos com o Microsoft Windows. Nokia Ovi Suite é a marca dos serviços de Internet da Nokia e os serviços do Ovi podem ser usados em um dispositivo móvel, computador com Ovi Suite ou via web. O objetivo principal da Nokia era concentrar nas cinco principais áreas de serviços, jogos, mapas, mídia, mensagens e música. Os Serviços Ovi eram abertos e foram adquiridos pela Nokia, mas em seguida foram comprados pela Microsoft Mobile.

- Aplicativos Samsung Galaxy

A Samsung Galaxy Apps é uma plataforma de distribuição digital, desenvolvida e mantida pela Samsung. A loja de aplicativos usada para dispositivos fabricados pela Samsung Electronics que foi lançada em setembro de 2009. Este desenvolvimento foi para seus usuários terem uma loja exclusiva de aplicativos e dispensasse a necessidade de recorrerem ao Google Play.

- Centro de Software Ubuntu

O Ubuntu Software center é um repositório com gráfico de alto nível criado para o sistema operacional Linux, oferecendo aos usuários muitos softwares excelentes por padrão. Ele é um software gratuito que possibilita a instalação de software em um computador que executa o Ubuntu OS. Permite navegar e instalar milhares de aplicativos gratuitos e pagos disponíveis.

- Windows Store

A Windows Store é uma plataforma de distribuição digital da Microsoft. A Windows Store era conhecida como Microsoft Store. Ela é um marketplace de lojas de varejo e um site de compras online de aplicativos e softwares. O objetivo inicial era entregar ao usuários aplicativos para Windows 8 e Windows Server 2012, porém com o Windows 10, a Microsoft juntou todas as suas plataformas de distribuição como Windows Marketplace, Windows Phone Store, Xbox Video e Xbox Music na Microsoft Store.

Com diversas plataformas disponíveis para os usuários, a escolha de uma delas para realizar desenvolvimentos vem sendo um desafio cada vez maior, porém neste trabalho, nos baseamos em estudos de consumos para definir a plataforma de distribuição digital da Google play com melhor aderência ao mercado, de acordo com (14).

2.3.2 Construção de APP

A construção de um aplicativo (APP) pode variar dependendo do sistema operacional para o qual será desenvolvido, os mais comuns são iOS, Android e Windows Phone, e das tecnologias que podemos escolher utilizar. No geral, algumas etapas são padronizadas para a construção de um APP. a seguir vamos descreve-las demonstrando as ações realizadas neste trabalho.

2.3.2.1 Definição de requisitos

Para criar um APP, é necessário definir os requisitos para garantir que ele atenda às necessidades dos usuários e cumpra seus objetivos.É importante ter uma compreensão clara dos recursos e funcionalidades que ele irá disponibilizar. A seguir, apresentaremos alguns requisitos comuns a serem considerados na criação de um APP.

- **Objetivo do aplicativo:** Devemos identificar claramente o objetivo do aplicativo, assim devemos reunir informações específicas sobre a usabilidade e propósito, bem como o quantitativo de usuários poderão utiliza-los.
- **Plataforma:** Determinar em qual ou quais plataformas o aplicativo será desenvolvido, levando em consideração o objetivo do Aplicativo.
- **Design de interface do usuário (UI):** Definir uma interface intuitiva e fácil de usar que seja atraente visualmente e ofereça uma experiência positiva ao usuário.
- **Funcionalidades principais:** Listar as funcionalidades essenciais que o aplicativo deve ter para cumprir seu objetivo principal. Por exemplo, se for um aplicativo de listas de compras, pode incluir recursos como adicionar itens à lista, marcar itens comprados e compartilhar a lista com outras pessoas.
- **Segurança:** Garantir a segurança das informações dos usuários através da implementação de medidas adequadas, como criptografia de dados e autenticação segura.

- Performance e velocidade: Assegurar que o aplicativo seja rápido e responsivo, evitando lentidão ou travamentos durante o uso.
- Compatibilidade com dispositivos: Certificar-se de que o aplicativo funcione corretamente em diferentes tamanhos de tela e resoluções, adaptando-se aos diversos dispositivos móveis existentes no mercado.
- Integração com serviços externos: Se necessário, permitir integrações com outros serviços ou APIs para expandir as capacidades do aplicativo (15).
- Suporte e manutenção: Planejar a disponibilidade de suporte técnico e atualizações regulares para corrigir bugs, adicionar novos recursos ou garantir a compatibilidade com as atualizações do sistema operacional.
- Testes: Realizar testes em diferentes cenários para verificar se o aplicativo funciona corretamente, é intuitivo e oferece uma experiência satisfatória ao usuário.

Dependendo do serviço a ser oferecido, outros requisitos podem surgir e serem considerado no desenvolvimento. A definição completa dos requisitos dependerá das necessidades específicas do aplicativo e dos usuários-alvo. Na figura 2.11 demonstramos o fluxo básicos de requisitos.



Figura 2.11: Fluxo básico de requisitos.

Fonte:Próprio Autor

2.3.2.2 Design da interface do usuário

O design da interface do usuário para apps envolve a criação e organização visual dos elementos presentes na tela, com o objetivo de proporcionar uma experiência intuitiva e agradável para os usuários. Isso inclui a escolha de cores, tipografia, ícones, layout e interações. Ela deve ser focado na usabilidade, facilitando a navegação e fornecendo feedback claro aos usuários. Além disso, é importante considerar as diretrizes específicas de cada plataforma em que o app será utilizado, como iOS ou Android, para garantir consistência e familiaridade aos usuários. Podemos utilizar ferramentas como Adobe XD, Sketch ou Figma para criar designs interativos ou buscar em páginas específicas de layout para APP. Na figura 3.11 está ilustrada um exemplo de layout de APP.



Figura 2.12: Exemplo de interface com o usuário.

Fonte:Free Vector

2.3.2.3 Plataforma de desenvolvimento

Observando as necessidades do projeto, podemos optar por desenvolver nativamente utilizando as linguagens específicas da plataforma, como a Swift/Objective-C para iOS e a Java/Kotlin para Android, ou usar uma abordagem híbrida/cross-platform, usando frameworks como React Native, Flutter ou Xamarin. Como a proposta deste trabalho é desenvolver a arquitetura para construção do App Sefe Driving em Android, utilizaremos o Java/Kotlin para Android como especificação.

- Java/Kotlin:

O Kotlin é uma linguagem de programação moderna e com tipagem estática desenvolvida pela JetBrains (16). Foi projetado para ser completamente interoperável com o Java, podendo facilmente converter um código Java existente para Kotlin e vice-versa. Essa interoperabilidade é uma das principais razões pelas quais o Kotlin se tornou popular entre os desenvolvedores Android. Ele foi lançado pela JetBrains em 2011 e teve sua versão 1.0 lançada em 2016 com o objetivo de oferecer uma alternativa melhor ao Java, eliminando suas limitações e adicionando recursos mais modernos. Uma das principais vantagens do Ko-

tlin é sua concisão. Ele possui uma sintaxe limpa e enxuta em comparação com o Java, tornando o código mais legível e fácil de escrever. O Kotlin possui algumas diferenças importantes em relação ao Java. Por exemplo, em vez de usar a palavra-chave "class" para definir uma classe, usamos a palavra-chave "class". Além disso, o Kotlin possui recursos extras, como a declaração de variáveis por meio da palavra-chave "val"(imutáveis) ou "var"(mutáveis). O Kotlin também permite a criação de arquivos de código independentes, chamados de "arquivos de script", que podem ser executados diretamente sem a necessidade de uma classe principal. Além disso, o Kotlin também traz recursos poderosos, como nullability de tipos, extensões de funções e suporte à programação funcional. A diferença entre sintaxe de linguagem entre o Kotlin e o Java é expressa na fig 2.13.

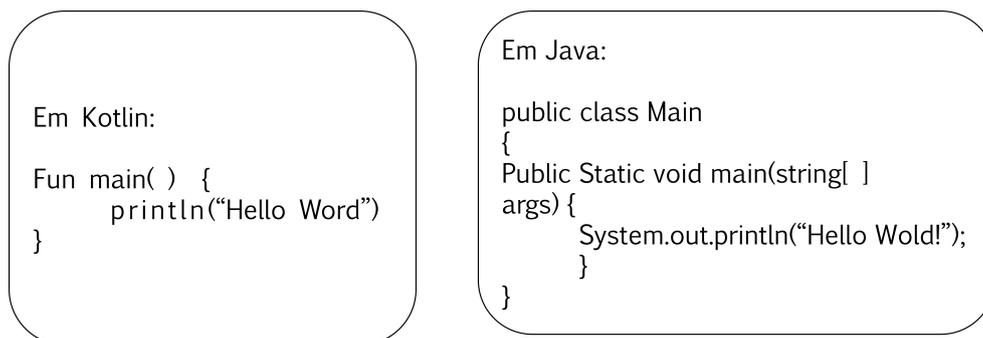


Figura 2.13: Diferença de Sintaxe entre o Java e o Kotlin.

Fonte:waldo.com

A nullability de tipos é um recurso importante do Kotlin que ajuda a evitar erros comuns relacionados a valores nulos. Com o Kotlin, podemos especificar se uma variável pode ou não ser nula, o que ajuda a evitar erros de NullPointerException que são comuns no Java. As extensões de funções são outro recurso interessante do Kotlin. Elas permitem adicionar novas funções a classes existentes sem precisar modificá-las ou herdar delas. Isso torna o código mais modular e fácil de manter. Outra característica do Kotlin é o suporte à programação funcional. Ele permite escrever código mais conciso e expressivo, aproveitando conceitos como funções de ordem superior, lambdas e imutabilidade de dados.

As Vantagens de usar Kotlin em relação a outras linguagens de programação, em especial para o desenvolvimento Android, é a oferta de maior produtividade, pois sua sintaxe é limpa e possui recursos avançados, possibilitando a escrita de códigos com maior velocidade e com menores erros. O Kotlin elimina a necessidade de escrever código redundante, tornando o desenvolvimento mais eficiente. Ele também oferece recursos de segurança aprimorados, como a nullability de tipos, que ajudam a evitar erros comuns, tornando o código mais robusto.

Existem várias ferramentas de desenvolvimento e IDEs disponíveis para trabalhar com Kotlin. A JetBrains, criadora do Kotlin, oferece o IntelliJ IDEA, que é uma IDE poderosa e amplamente utilizada por desenvolvedores Kotlin. Além disso, o Android Studio, o ambiente de desenvolvimento oficial do Android, também possui suporte completo ao Kotlin. Além dessas IDEs, podemos usar outras ferramentas populares, como o VS Code, o Eclipse e o NetBeans, com plugins e extensões para suporte ao Kotlin. Neste trabalho a ferramenta de desenvolvimento especificada é o Android Studio.

2.3.2.4 Configuração do ambiente de desenvolvimento

Para configurar o ambiente se faz necessário a instalação dos softwares para o desenvolvimento do APP, como os Ambientes de desenvolvimento integrado (IDE), por exemplo Xcode para iOS e Android Studio para Android, emuladores/simuladores e outras dependências relevantes. Neste ambiente vamos escrever todos os códigos e desenvolver todas as funcionalidades dos APPs e conforme mencionado anteriormente, o ambiente de desenvolvimento especificado será o Android Studio (17).

O Android Studio é projetado especificamente para criar aplicativos Android. Desenvolvido pelo Google, ele oferece uma ampla gama de recursos e ferramentas poderosas que ajudam os desenvolvedores a escrever, depurar e testar seus aplicativos de forma eficiente. Com ele temos acesso a um editor de código avançado, um depurador robusto, emuladores de dispositivos virtuais, bem como muitos outros recursos que tornam mais fácil desenvolver aplicativos Android. Além disso, o Android Studio também suporta várias linguagens de programação, como Java e Kotlin, permitindo a escolha de linguagem com a qual estão mais familiarizados.

Recursos oferecidos pelo Android Studio ajudam a criar aplicativos Android de alta qualidade. O Editor de código avançado realiza realce de sintaxe, preenchimento automático e navegação rápida entre classes e métodos. Ele vem com um depurador integrado que permite aos desenvolvedores identificar e corrigir problemas em seus aplicativos durante o desenvolvimento. Também possui um emuladores de dispositivo virtual, permitindo testar seus aplicativos, que simulam diferentes dispositivos Android, permitindo que eles verifiquem como seus aplicativos se comportam em diferentes tamanhos de tela e versões do Android.

O Android Studio possui um gerenciador de pacotes integrado que facilita a adição e atualização de bibliotecas e dependências em seus projetos. Ele oferece também, ferramentas avançadas de design de interface que permitem criar interfaces de usuário atraentes e intuitivas para seus aplicativos. A interface do usuário é demonstrada na figura 2.14.

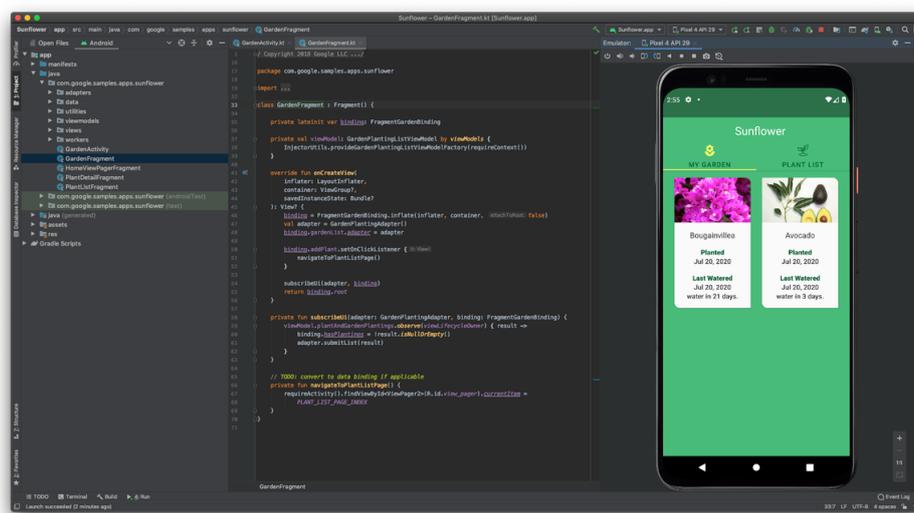


Figura 2.14: Interface do Android Studio.

Fonte: Xcom

2.3.2.5 Desenvolvimento do backend

O App a ser desenvolvido pode exigir algum tipo de processamento no servidor ou integração com APIs externas. Para configurar o backend, deverá ser utilizadas tecnologias como Node.js, Python Django ou Ruby on Rails.

O Node.js é um ambiente de execução JavaScript que permite executar aplicações desenvolvidas com a linguagem de forma autônoma, sem depender de um navegador. Com ele, é possível criar praticamente qualquer tipo de aplicações web, desde servidores para sites estáticos e dinâmicos, até APIs e sistemas baseados em microserviços. A principal característica do Node.js é sua execução ser single-thread, ou seja, os recursos computacionais são alocados apenas uma vez pelo tempo que a aplicação estiver sendo executada. Essa thread única é chamada de “Event Loop”. Ela trata todas as requisições como eventos, de maneira assíncrona e não-bloqueável, eliminando a necessidade de filas de processamento e tornando as aplicações mais eficientes e responsivas (18).

O Node.js não é um framework JavaScript, como o React ou o Vue.js, mas sim um ambiente de execução em uma máquina virtual própria para interpretar e executar os scripts de forma autônoma, baseada no “motor” V8, criado pelo Google para o navegador Chrome. Devido ao baixo consumo de recursos e à capacidade de processar várias requisições simultaneamente, o Node.js é muito utilizado em aplicações que requerem uma alta escalabilidade, ou seja, que possam crescer sem perder qualidade ou aumentar os custos. Outra vantagem é a possibilidade de manter o ecossistema de aplicações e toda a base de código em uma só linguagem de programação, no caso, o JavaScript. Na figura 2.15 podemos visualizar a execução no console.

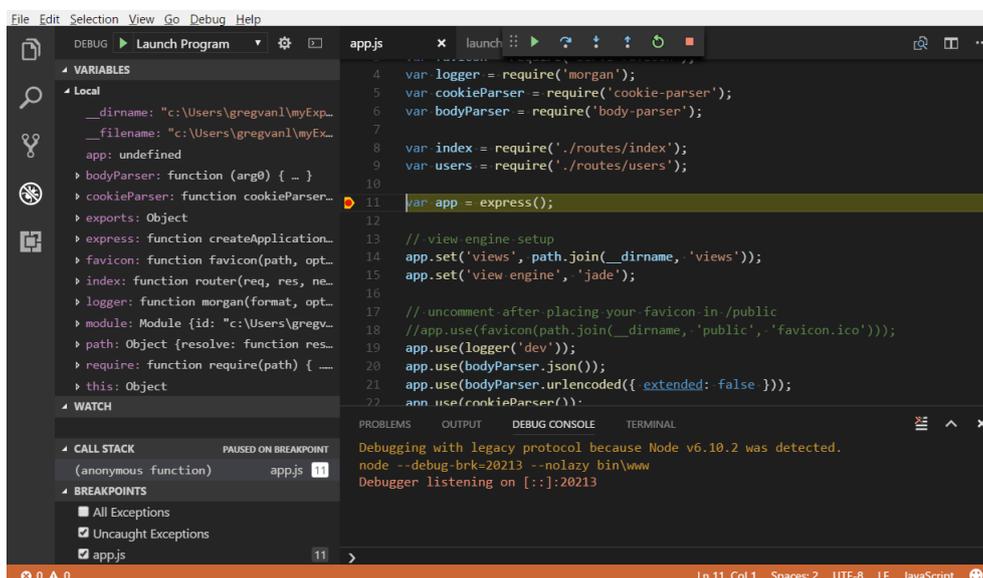


Figura 2.15: Execução no console.

Fonte: code.visual studio

Outra tecnologia utilizada é o Django. O Django é um framework web gratuito e de código aberto escrito em Python para desenvolvimento web. Ele permite a construção de aplicações web de alto desempenho e é usado para resolver problemas comuns no processo de desenvolvimento de aplicações web, como autenticação, rotas, object relational mapper (ORM) e até migrations. O Django funciona com base

no modelo Model-View-Template(MVT), onde Model representa os dados da aplicação e tem ligação direta com um banco de dados. É a camada de abstração feita para manipular, incluir ou excluir dados. View é uma função Python feita para receber uma requisição e enviar uma resposta em retorno. É aqui onde os dados são extraídos e produzem uma resposta e Template é a camada onde o modelo da aplicação é produzido através de artefatos compreendidos pelo navegador web. Esta é a parte que diz respeito a tudo que o usuário final é capaz de visualizar em seu dispositivo. O Django foi criado com o objetivo de permitir o rápido desenvolvimento de sites seguros e de fácil manutenção. Ele é muito utilizado em projetos que requerem escalabilidade e possui um eficiente sistema de segurança. A figura 2.16 demonstra a interface do usuário na tecnologia Django.

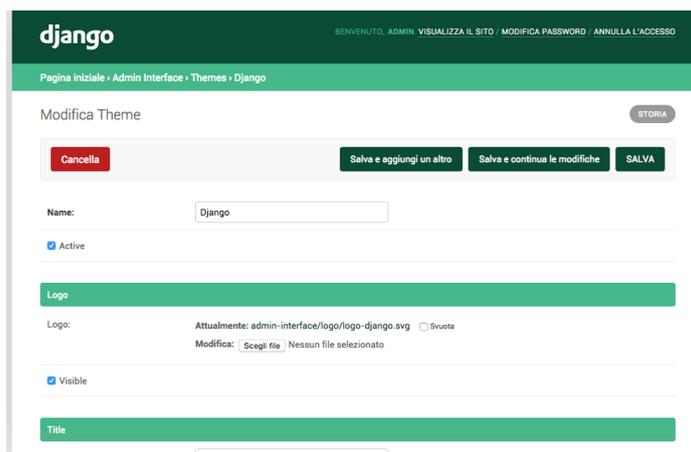


Figura 2.16: Interface do Usuário da tecnologia Django.

Fonte: GitHub

Uma tecnologia também utilizada é o Ruby on Rails, Ruby on Rails, também conhecido como Rails, é um framework de desenvolvimento de aplicações web escrito em Ruby. Ele foi lançado em 2004 e é livre e multiplataforma. O Rails permite a criação de aplicações com base em estruturas pré-definidas, o que facilita e acelera o desenvolvimento. Ele segue o padrão arquitetural Model-View-Controller (MVC). Onde Model é o gerenciamento do comportamento dos dados, incluindo recursos como consultas ao banco de dados, validações, notificações, etc. O View gerencia a saída gráfica e textual da parte da aplicação visível ao usuário final e o Controller interpreta as entradas do mouse e teclado do usuário, comandando a visão e o modelo para se alterarem de forma apropriada. O Rails é composto por várias bibliotecas que podem ser usadas em projetos Ruby de forma independente. Por exemplo, o Active Records é um Object Relational Mapping (ORM) que permite a integração de objetos Ruby com sistemas de banco de dados. Diversas aplicações populares foram desenvolvidas usando Rails, incluindo GitHub, Shopify, Airbnb, Twitch, SoundCloud, Hulu, Zendesk, Square, entre outros. Na figura 2.17 demonstramos a interface do usuário do Rails.

2.3.2.6 Implementação das funcionalidades

A implementação de funcionalidades são imputadas por meio da escrita dos códigos. Essas diferentes funcionalidades definidas nos requisitos. Isso pode incluir a programação de recursos como login, cadastro de usuários, integração com banco de dados, envio de notificações push e demais funcionalidades de acordo

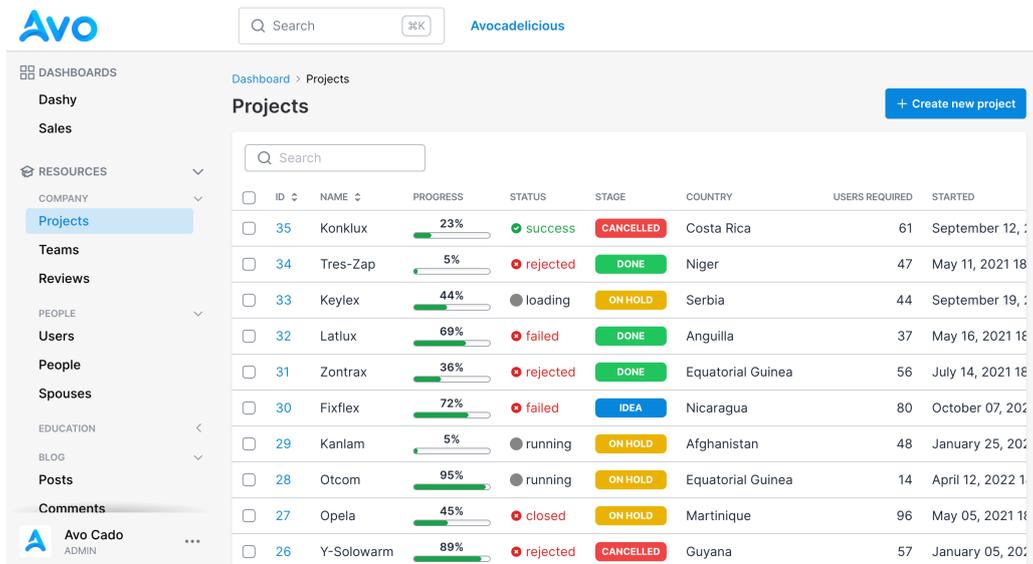


Figura 2.17: Interface do Usuário do Rails.

Fonte: Avo Admin For Rails

com as necessidades do APP. A figura 2.18 demonstra um mapa de funcionalidades a ser implementado em um app.

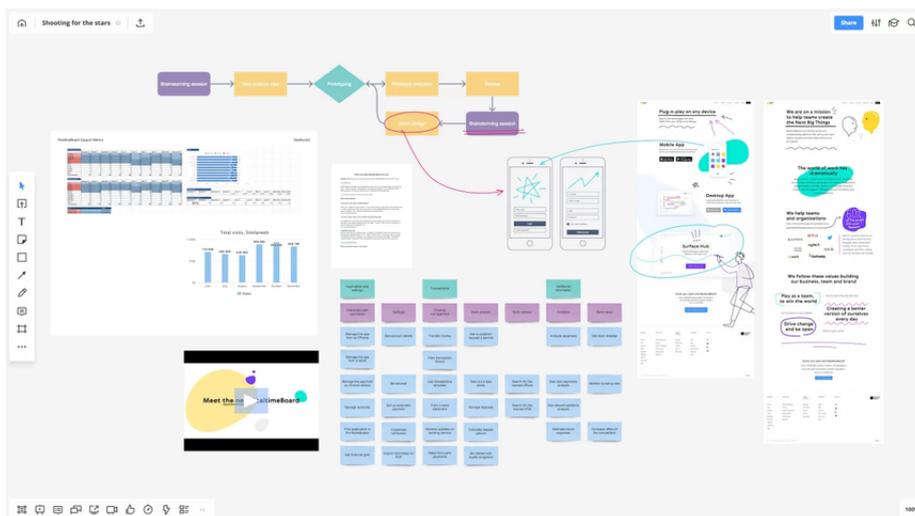


Figura 2.18: Mapa de funcionalidades para um APP.

Fonte: Target Teal

2.3.2.7 Testes e depuração

O próximo passo é realizar os testes regulares para garantir que o seu APP esteja funcionando corretamente e livre de bugs. Isso pode incluir testes unitários, testes de interface do usuário (UI) e testes de integração (19).

2.3.2.8 Publicação na loja de aplicativos

Com o APP pronto para ser lançado, será preciso submetê-lo a revisão nas respectivas lojas de aplicativos, App Store da Apple ou Google Play Store. A ação de revisão, se faz necessária para certificar o cumprimento de todas as diretrizes e políticas específicas da plataforma escolhida. A escolha proposta da loja para publicação do App, será a Google Play Store, pois a proposta de desenvolvimento está adequada a ela. A figura 2.19 demonstra a rejeição de um APP na play store Google quando o mesmo não atende as diretrizes e políticas previstas.

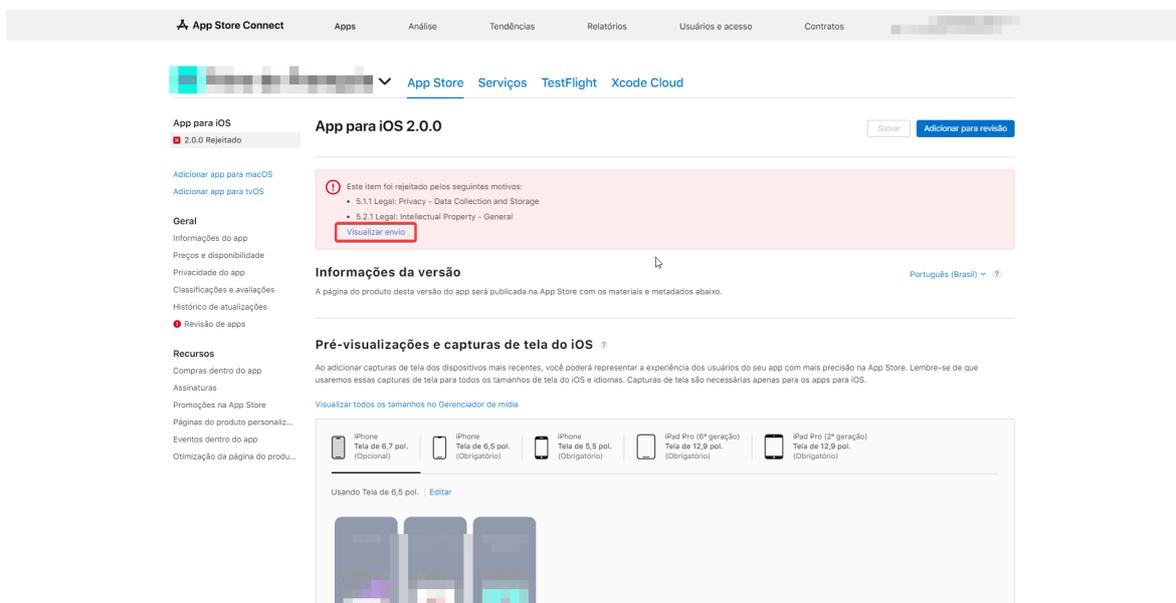


Figura 2.19: Mensagem de rejeição de APP pela PLayer Store Google.

Fonte: Central de Ajuda FabApp

2.3.3 TRABALHOS RELACIONADOS

Neste trabalho visamos estabelecer uma arquitetura para desenvolvimento e gerenciamento do Aplicativo Safe Driving, de modo que buscamos por referências em trabalhos relacionados a desenvolvimento de APPs, Conexão de App com sistemas eletrônicos automotivos via bluetooth, análise e processamento de dados obtidos de veículos e Smartphones.

Em (20) é proposto um mecanismo de avaliação de motoristas baseado em lógica fuzzy que utiliza o acelerômetro, magnetômetro e sensor de gravidade de smartphones Android como entrada, classificando o motorista nos estilos de direção **Normal, Moderado e Agressivo** e calcula seu score de direção de 0 (melhor) a 100(pior). A classificação e o score não são calculados em tempo real, visto que os dados de sensores são coletados pelo aplicativo UBI-Meter e armazenados no smartphone para posterior envio a um servidor remoto na Internet que agrega dados de diversos motoristas e os analisa.

Já em (21) apresenta o SenseFleet, essa é uma plataforma de APM baseada em smartphone Android. O aplicativo coleta dados do acelerômetro, magnetômetro, sensor de gravidade e GPS do smartphone com o objetivo de detectar eventos de risco (velocidade, aceleração, frenagem e esterçamento excessivos)

ocorridos durante uma viagem. Uma nova pontuação é gerada ao motorista a cada viagem percorrida em função da detecção destes eventos, podendo variar entre 0 (pior) a 100 (melhor) pontos. Tal detecção é realizada por um sistema fuzzy cujos limites são obtidos dinamicamente através de um processo de calibragem executado na primeira vez que o aplicativo é utilizado em uma nova combinação de motorista, smartphone e veículo.

Com o objetivo de utilizar o GPS, acelerômetro e magnetômetro dos smartphones para classificar a forma de dirigir de motoristas nos perfis de segurança **muito seguro, seguro, agressivo, e muito agressivo**. Esta classificação está descrita em (22) e é realizada por meio da detecção, em tempo real, de eventos de condução relevantes que podem ocorrer durante uma viagem. A detecção é feita por um algoritmo de reconhecimento de padrões descrito em (23) baseado na técnica de DTW. Os eventos detectados por este algoritmo são frenagem agressiva, aceleração brusca, curva acentuada, mudança de faixa agressiva e velocidade excessiva.

Já (24), realiza uma investigação com diferentes sensores, presentes em um smartphone Android, e diferentes algoritmos de classificação, a fim de avaliar qual conjunto/método de sensores permite a classificação com maior precisão. Os resultados mostram que combinações específicas de sensores e métodos inteligentes permitem melhorar o desempenho da taxa.

Em (25), foi apresentado um protótipo desenvolvido para a plataforma Android utilizando um banco de dados MongoDB, os GE's Orion Context Broker e Cygnus da plataforma FIWARE e um dispositivo OBD. No momento que o aplicativo se conecta, via Bluetooth, ao dispositivo OBD, as informações do computador de bordo do veículo e a geolocalização coletada pelo sensor de GPS do smartphone começam a ser transmitidas através das redes móveis 3G e 4G para o Orion Context Broker na plataforma FIWARE, via mensagens REST. Este por sua vez encaminha os dados para o conector Cygnus que os armazena em um banco de dados MongoDB.

Em, (26) visa definir coletas de dados de telemetria veicular, dados de sensores do Smartphone do usuário para transmitir ao servidor e retornar dados geográficos e alertas de acidentes para o usuário.

Esses trabalhos fornecem uma visão abrangente das abordagens existentes para a coleta e tratamento de dados de veículos e Smartphones e serviram de ponto de partida para a ideação e desenvolvimento da arquitetura proposta neste trabalho, bem como a possibilidade do desenvolvimento futuro do App Safe Driving e sua inserção no mercado.

3 ESTRUTURA DA PROPOSTA DO APP PARA AVALIAÇÃO DE CONDUÇÃO SEGURA

Neste capítulo será apresentado a arquitetura proposta para desenvolvimento do App, voltado a avaliação de condução segura. Na figura 3.1 e na figura 3.2 é possível visualizar as etapas para definição da proposta até chegar à construção da arquitetura final da proposta.

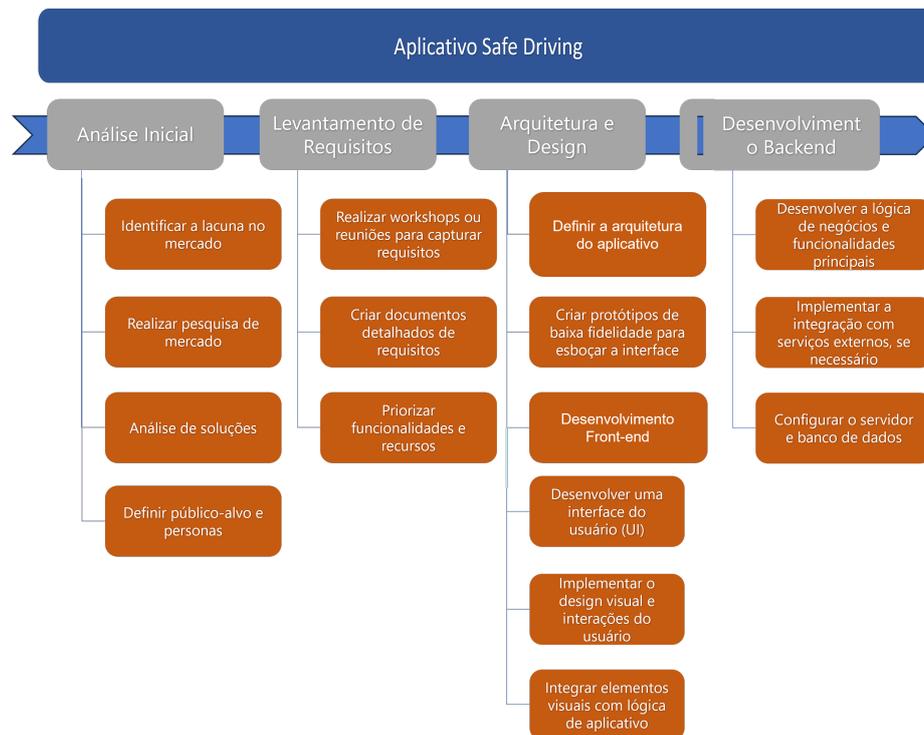


Figura 3.1: Estrutura da proposta do desenvolvimento do APP - parte 1
Fonte: autoria própria

3.1 ANÁLISE INICIAL

A fase inicial de análise é um marco em qualquer empreendimento de desenvolvimento de aplicativos, delineando os alicerces para todo o processo subsequente. No contexto do trabalho em questão, essa fase foi responsável por examinar os objetivos específicos do aplicativo, identificar suas funcionalidades principais e avaliar a viabilidade global do empreendimento. Essa etapa não apenas se concentrou na identificação das necessidades latentes no mercado, mas também definiu de maneira clara os requisitos essenciais para o desenvolvimento da arquitetura do aplicativo, estabelecendo as bases para a compreensão do público-alvo e a avaliação da concorrência existente (27, 28).

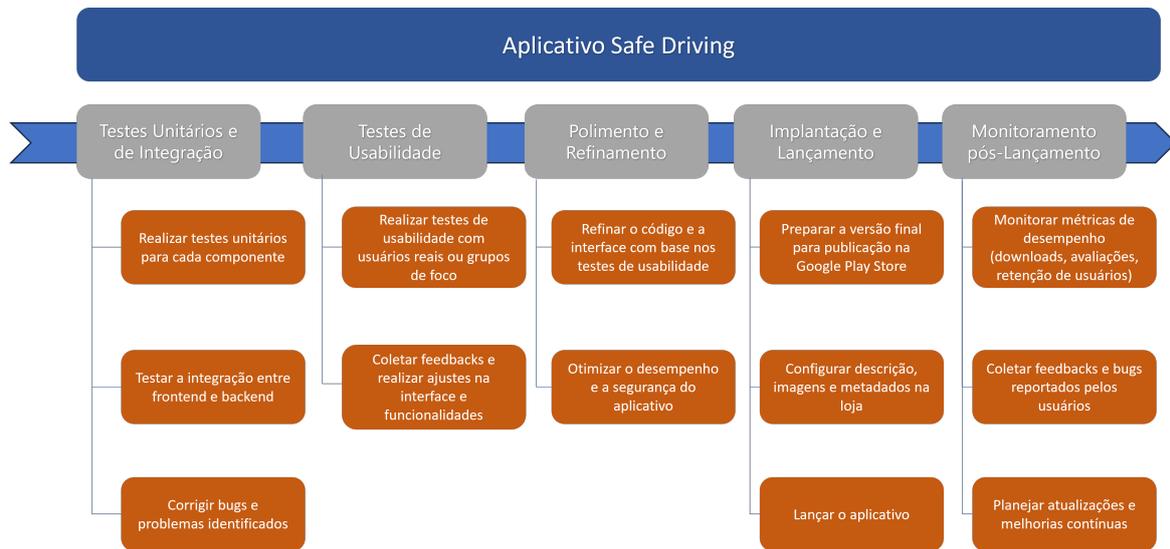


Figura 3.2: Estrutura da proposta do desenvolvimento do APP - parte 2

Fonte: autoria própria

A análise inicial também foi para determinar as métricas de sucesso que, posteriormente, seriam utilizadas para mensurar o desempenho do aplicativo após seu lançamento futuro. A definição dessas métricas é vital para garantir que todos os esforços estejam alinhados com os objetivos estratégicos inicialmente estabelecidos (29). Dessa forma, o processo de análise inicial não apenas atua como um ponto de partida, mas também como um guia orientador que influencia diretamente as decisões tomadas durante todo o ciclo de desenvolvimento.

No desenvolvimento do trabalho, foram adotadas abordagens de design fundamentadas em conceitos como MVP (Model-View-Presenter) (30), MVVM (Model-View-ViewModel) (31), e Clean Architecture (32). Essas abordagens foram escolhidas com base na sua eficácia comprovada em proporcionar uma estrutura organizada e modular para o desenvolvimento de aplicativos, promovendo a escalabilidade e a manutenibilidade do código-fonte.

Além disso, a aplicação de métricas de usabilidade, conforme proposto por Ferreira em (33), desempenhou um papel fundamental na avaliação contínua do design do aplicativo. Testes automatizados para o desenvolvimento mobile em Android, conforme discutido por Freitas em (34), foram incorporados para garantir a estabilidade e a confiabilidade do aplicativo ao longo do tempo.

Para a construção da arquitetura final, foram consultados e implementados os princípios universais do design (35) e estratégias eficazes de APIs (36, 37). O cuidado em integrar frontend e backend, como discutido por Mardan e Mardan em (38), foi fundamental para assegurar a coesão e a eficiência do sistema como um todo.

Ao longo desse processo, a pesquisa de lacuna de mercado (28, 27) desempenhou um papel importante, fornecendo insights sobre as tendências de tipos de App a serem utilizados no mercado e as expectativas apresentadas pelos usuários.

3.1.1 Identificar a lacuna no mercado

A identificação de uma lacuna no mercado é um processo estratégico que demanda uma análise das necessidades não atendidas ou das oportunidades sub-exploradas. Nossa abordagem para alcançar esse objetivo envolveu a observação de tendências atuais e futuras, buscando áreas onde os aplicativos existentes possam não estar totalmente satisfazendo as demandas dos usuários (27). Essa análise profunda foi fundamental para a definição de uma proposta única e inovadora que atende-se as expectativas dos usuários e tendências de tipos de App a serem utilizados.

No âmbito deste projeto, adotamos uma abordagem proativa para identificar áreas emergentes, explorando tecnologias inovadoras e atentando para as mudanças no comportamento do consumidor (39). Essa metodologia nos permitiu não apenas antecipar as necessidades dos usuários, mas também posicionar ao arquitetura do nosso aplicativo como uma solução diferenciada e altamente relevante no mercado competitivo de aplicativos.

Para validar nossas descobertas e afinar ainda mais a identificação da lacuna no mercado, realizamos pesquisas diretas com os usuários voluntários para melhor entender a possibilidade de adesão ao App. Essa abordagem permitiu a coleta de feedbacks, identificação de pontos de insatisfação específicos e a compreensão das necessidades não atendidas pelos aplicativos existentes (27, 28).

Para alcançar os resultados, pesquisamos 80 voluntários, com idade entre 18 e 80 anos, utilizando a ferramenta "forms - Google", com o objetivo final de saber se utilizariam o aplicativo em seu dia-a-dia, respondendo as seguintes questões:

1. Qual é o seu gênero?
2. Qual é a sua faixa etária?
3. Você possui veículo automotor?
4. Você possui seguro em seu veículo?
5. Você utiliza Apps que lhe oferecem vantagens financeiras como Cashbacks, milhas e outros?
6. Você utilizaria um App que te ofereça premiação (redução de valores pagos) em seguros veiculares?
7. Você consideraria o uso de um App que te ofereça premiação (redução de valores pagos) em seguros veiculares?
8. Você utilizaria um App que te ofereça premiação (redução de valores pagos) em seguros veiculares, mesmo sabendo que ele está avaliando sua forma de dirigir?

A figura 3.3 nos apresenta o fluxograma percorrido pelos voluntários respondentes da pesquisa, onde identificamos o comportamento e desejos dos clientes.

Com base na pesquisa, conseguimos definir as personas de possíveis clientes e questões relevantes de pesquisa de mercado. A tabela 3.1 a seguir nos mostra resultados relacionados ao perfil dos respondentes e gostaríamos de saber qual era o gênero que mais desejaria utilizar o App.

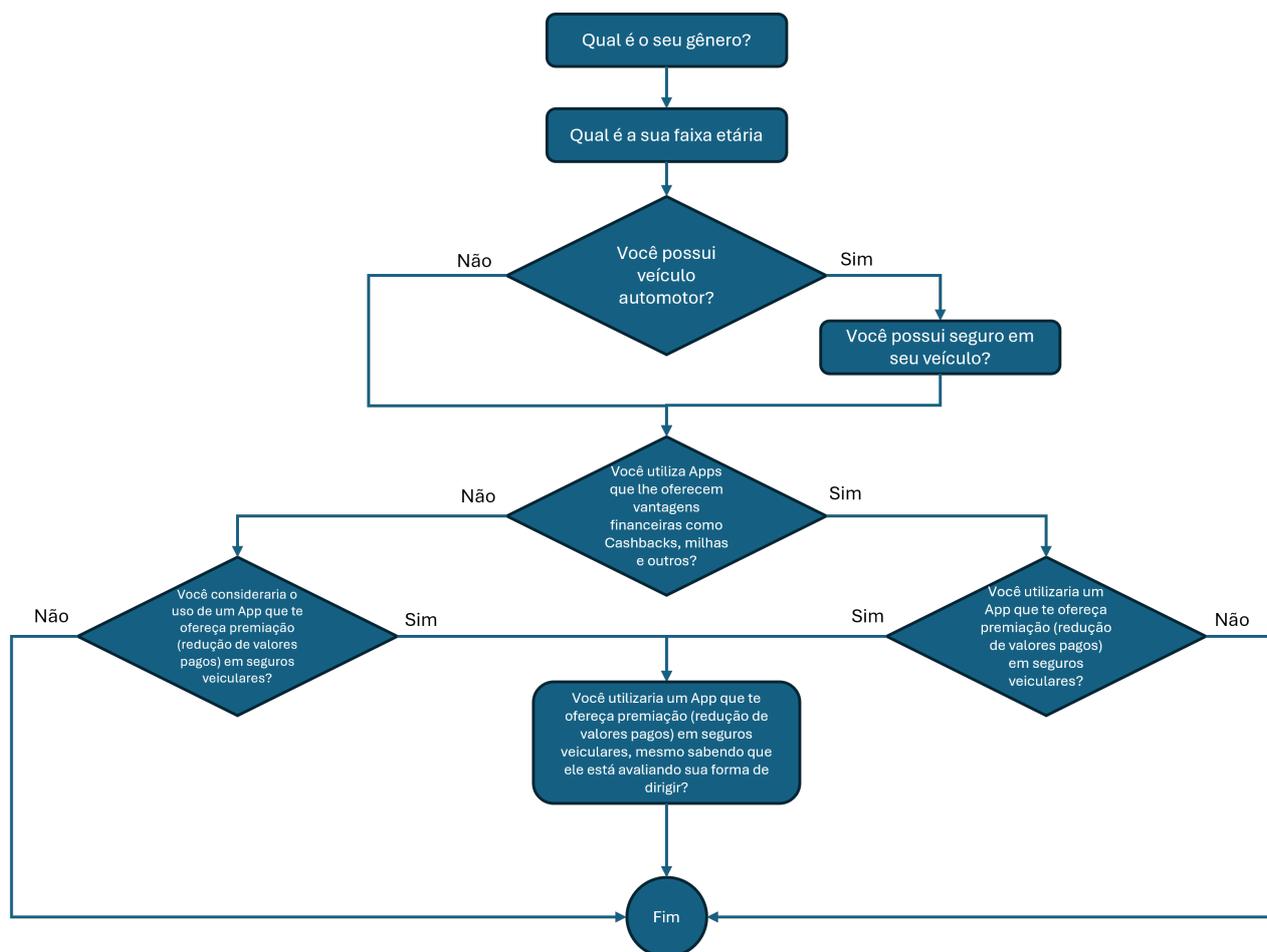


Figura 3.3: Fluxograma de respostas da pesquisa de lacuna de mercado

Fonte: autoria própria

Tabela 3.1: Pesquisa de Lacuna de Mercado - Gênero

Perguntas	Masculino	Feminino	Outros
Qual é o seu gênero?	71,3%	28,7%	0%

A tabela 3.1 mostra que predominantemente os respondentes são homens, apesar de termos uma participação significativa das mulheres na pesquisa. Essas mulheres e homens estão divididos em faixas etárias, o que trouxe uma possível faixa etária de utilização do App. Esses resultados estão expressos na tabela 3.2.

Tabela 3.2: Pesquisa de Lacuna de Mercado - Faixa etária

Qual é a sua faixa etária?	%
Entre 18 e 25 anos	3,8%
Entre 26 e 35 anos	11,2%
Entre 36 e 45 anos	28,7%
Entre 46 e 55 anos	30%
Entre 56 e 65 anos	21,3%
66 anos ou mais	5%

Quando olhamos a tabela 3.2 percebemos que, no geral, o público alvo estaria entre 46 e 55 anos e

em um somatório de faixas etárias como maiores usuários pessoas entre 36 e 65 anos, somando 80% dos respondentes. Porém, quando dividimos o público respondente por gênero, temos dados diferentes por grupo conforme a tabela 3.3 do gênero feminino a seguir:

Tabela 3.3: Pesquisa de Lacuna de Mercado - Faixa etária + Gênero Feminino

Qual é a sua faixa etária?	%
Entre 18 e 25 anos	13,04%
Entre 26 e 35 anos	17,39%
Entre 36 e 45 anos	30,43%
Entre 46 e 55 anos	17,39%
Entre 56 e 65 anos	21,74%

Na tabela 3.4 podemos observar o comportamento dos respondentes de gênero masculino.

Tabela 3.4: Pesquisa de Lacuna de Mercado - Faixa etária + Gênero Masculino

Qual é a sua faixa etária?	%
Entre 26 e 35 anos	9%
Entre 36 e 45 anos	28%
Entre 46 e 55 anos	35%
Entre 56 e 65 anos	21%
66 anos ou mais	7%

Assim concluímos que as mulheres, possíveis usuárias da proposta de App, estariam com idade entre 36 e 45 anos, enquanto os homens, estariam com idade entre 46 e 55 anos. desta forma começamos a desenhar as personas de usuárias da proposta de App.

Para aprofundarmos na construção das personas e comprovar se existe uma lacuna de mercado que pode ser explorada, continuamos com uma pergunta mais específica para classificar reais possíveis usuários. Na tabela 3.5 a seguir, mostramos mais dados decisórios para essa construção.

Tabela 3.5: Pesquisa de Lacuna de Mercado - Posse de veículos Automotor

Perguntas	SIM	NÃO
Você possui veículo Automotor?	88,75%	12,68%

Nesta pergunta também podemos avaliar o número de mulheres e homens que possuem veículos automotores, o que aproximaria e melhoraria ainda mais a formação das personas usuárias da proposta de App. A tabela 3.6 demonstra a proporção de homens e mulheres que possuem ou não veículos automotores com base nas respostas colhidas dos voluntários desta pesquisa.

Tabela 3.6: Pesquisa de Lacuna de Mercado - Posse de veículos + Gênero

Você possui veículo Automotor?	% SIM	% NÃO
Feminino	78,26 %	21,74 %
Masculino	92,98%	7,02%

Esta tabela 3.6 nos mostra que um percentual maior de homens são potenciais usuários da proposta de App, porém o número de mulheres é significativo e deve ser considerado para o desenho das personas.

Quanto aos respondentes que não possuem veículos, estes não foram descartados diretamente, pois podem voltar a possuir futuramente um veículo, seja por necessidade ou comodidade. Dentre os que responderam "SIM", foi aberta a pergunta aos respondentes se ele possuía Seguro Veicular, pois a intenção era prepará-lo para visualizar o benefício a ser entregue pela proposta do App como algo de valor e ao mesmo tempo inovador. Os dados colhidos estão descritos na tabela 3.7.

Tabela 3.7: Pesquisa de Lacuna de Mercado - Seguro Veicular

Perguntas	SIM	NÃO
Você possui Seguro Veicular?	77,46%	22,54%

Na tabela 3.8 trazemos o percentual de homens e mulheres que possuem seguro veicular e que jugam importante a obtenção deste serviço para o seu uso.

Tabela 3.8: Pesquisa de Lacuna de Mercado - Seguro Veicular + Gênero

Você possui Seguro Veicular?	% SIM	% NÃO
Feminino	88,89 %	11,11 %
Masculino	73,58%	26,42%

Com o resultado da pesquisa na tabela 3.8 podemos concluir que as mulheres contam com serviço de seguro veicular em maior percentual que os homens, com base nesta informação, podemos afirmar que as mulheres, percentualmente, seriam usuários mais promissores da proposta de App.

A partir desta pergunta, podemos ter duas interpretações para as respostas, uma delas será abordada aqui, que é a pesquisa de lacuna de mercado, a outra será abordada adiante no trabalho e será apresentada no tema pesquisa de mercado. A pergunta exposta na tabela 3.9 junto com os seus respectivos dados, visa a afinar o perfil do usuário, sabendo de um tipo ou funções de Apps que ele já utiliza em seu cotidiano, bem como nos mostra se ele acredita nos benefícios gerados. A usabilidade de outros App ou programas de benefícios nos põem em um caminho de aceitação da proposta de App sugerida neste trabalho.

Tabela 3.9: Pesquisa de Lacuna de Mercado - Uso de Apps ou programa de benefícios

Pergunta	SIM	NÃO
Você utiliza Apps que lhe oferecem vantagens financeiras como Cashbacks, milhas e outros?	63,75%	36,25%

Trazemos os dados da pesquisa referente ao uso de app ou programa de benefícios separado por gênero, refinado ainda mais a construção das personas. Na tabela 3.10 demonstramos os resultados.

Tabela 3.10: Pesquisa de Lacuna de Mercado - Uso de Apps ou programa de benefícios + Gênero

Você utiliza Apps que lhe oferecem vantagens financeiras como Cashbacks, milhas e outros?	% SIM	% NÃO
Feminino	60,87 %	39,13 %
Masculino	64,91%	35,09%

Quando destacamos por gênero, há um percentual muito equivalente no uso em programa de Apps e programa de benefícios entre homens e mulheres, apontando a credibilidade na arquitetura proposta para o

App, bem como o modelo de negócios sugerido. Não deixando de preocupar, o volume de respondente que não utiliza Apps e programa de benefícios também é alto, não chega a ser preocupante para nos fazer recuar no desenvolvimento da arquitetura proposta para o App, mais para visão de mercado, pois o crescimento exponencial de uso do App poderia ser comprometido. Assim, os pesquisados percorreram caminhos diferentes de perguntas, a fim de extrair o máximo de informações possíveis e que pudesse nos dar um caminho seguro no desenvolvimento e mercadológico. Na tabela 3.11 veremos a pergunta realizada aos respondentes que disseram "SIM" para o questionamento anterior, bem como seus respectivos dados.

Tabela 3.11: Pesquisa de Lacuna de Mercado - Uso da proposta de App

Pergunta	SIM	NÃO
Você utilizaria um App que te ofereça premiação (redução de valores pagos) em seguros veiculares?	82,35%	17,65%

Aqui concluímos que existe uma lacuna de mercado a ser preenchida, um tipo de serviço ainda não ofertado para a sociedade, uma oportunidade que pode ser aproveitada mercadologicamente. Também avançamos para um percentual considerável de aceitação da proposta do App. Como padrão de demonstração de resultados, na tabela 3.12 trazemos os dados de uso da proposta separados por gênero, o que nos dá uma visão do grupo prioritário de uso.

Tabela 3.12: Pesquisa de Lacuna de Mercado - Uso da proposta de App + Gênero

Você utilizaria um App que te ofereça premiação (redução de valores pagos) em seguros veiculares?	% SIM	% NÃO
Feminino	85,71 %	14,79 %
Masculino	81,08%	18,92%

Os dados descritos na tabela 3.12 nos mostra a aprovação de ambos os gêneros para a proposta do App, com percentuais quase que equivalentes e de valor agregado ao futuro usuário. Porém conforme falamos anteriormente, ainda existe um percentual de pessoas considerável que disseram "NÃO" utilizar Apps ou programa de benefícios em seu cotidiano. Para essas, ainda assim, a proposta foi apresentada, porém de forma mais sutil, de uma forma que pudesse despertar o interesse de uso futuro na proposta do App. A tabela 3.13 apresenta dados relacionados a uma pergunta exclusiva realizada a esse perfil de respondente.

Tabela 3.13: Pesquisa de Lacuna de Mercado - Consideraria o uso da proposta de App

Pergunta	SIM	NÃO
Você Consideraria o uso de um App que te ofereça premiação (redução de valores pagos) em seguros veiculares?	86,21%	13,79%

Esses dados nos trouxeram uma grata surpresa em termos de aceitação da proposta de App, onde mesmo as pessoas que não utilizam App ou programa de benefícios, considerariam utilizar a proposta de App. Essa pessoa certamente será um usuário futuro, fato que diversas condições deverão acontecer antes de se tornar um usuário, mas com um olhar futuro de negócios é uma pessoa a se considerar nas estratégias de divulgação do App. Apresentamos aqui, a tabela 3.14 com dados separados por gênero para reafirmação da conclusão já citada anteriormente.

Aqui, concluímos que os homens tem a maior tendência de arriscar-se em um campo de tecnologia

Tabela 3.14: Pesquisa de Lacuna de Mercado - Consideraria o uso da proposta de App + Gênero

Você Consideraria o uso de um App que te ofereça premiação (redução de valores pagos) em seguros veiculares?	SIM	NÃO
Feminino	66,67 %	33,33 %
Masculino	95,00%	5,00%

diferente, pra ele atualmente duvidoso, fechando o perfil de nossa persona a ser atraída futuramente.

Como última pergunta, realizamos uma proposição a reafirmação das respostas emitidas anteriormente, mesmo com uma condição de vigilância ou avaliação dos atos dos usuários, mesmo que evidenciassem negativamente a sua forma de condução de veículos. Os dados são demonstrados na tabela 3.15.

Tabela 3.15: Pesquisa de Lacuna de Mercado - Usaria mesmo que sabendo que ele está avaliando sua forma de dirigir

Pergunta	SIM	NÃO
Você utilizaria um App que te ofereça premiação (redução de valores pagos) em seguros veiculares, mesmo sabendo que ele está avaliando sua forma de dirigir?	85,00%	15,00%

Os dados apresentados para esta pergunta revelam não só a reafirmação da proposta do App, mas uma necessidade do usuário, seja ela para ganhos de benefícios ou para aumento da educação no trânsito, onde todos, motoristas e pedestres se beneficiam com a melhoria do ecossistema de trânsito. Com esse pensamento, apresentamos os dados por gênero na tabela 3.15, afirmando a preocupação e perspectiva de ganhos de benefícios.

Tabela 3.16: Pesquisa de Lacuna de Mercado - Usaria mesmo que sabendo que ele está avaliando sua forma de dirigir + Gênero

Você utilizaria um App que te ofereça premiação (redução de valores pagos) em seguros veiculares, mesmo sabendo que ele está avaliando sua forma de dirigir?	SIM	NÃO
Feminino	82,61 %	17,39 %
Masculino	85,96%	14,04%

Esse dados nos traz como resposta o percentual equivalente de preocupação com a educação no trânsito e o desejo de ganhos de benefícios voltados a prêmio de seguro veicular. Assim, a identificação da lacuna no mercado foi realizada e afirmada que é existente por meio da aceitação da ideia. As abordagens estratégicas foram validadas pelos respondentes por meio da pesquisa direta. Os insights despertados através destes dados são valiosos para a construção da arquitetura da proposta do App. O nosso trabalho está posicionado para oferecer uma solução inovadora e lucrativa para o usuário, contemplando às demandas não atendidas pelos aplicativos existentes no mercado (28, 39).

3.1.2 Pesquisa de mercado

A condução de uma pesquisa de mercado para o desenvolvimento da proposta de nosso aplicativo é um passo estratégico que se baseia em uma abordagem sistemática para compreender o cenário no qual nosso produto será inserido. por meio da coleta realizada e análise de informações relevantes sobre o

mercado-alvo, a concorrência, as tendências do setor e as necessidades dos usuários (27).

Após identificarmos a lacuna de mercado não atendidas por aplicativos existentes, procuramos entender as preferências dos usuários e avaliando modelos de negócios similares (28). Esta fase crítica nos proporcionou uma compreensão aprofundada das oportunidades existentes no mercado, permitindo-nos delinear um conceito de aplicativo que não apenas atenda às necessidades atuais, mas que também antecipe as futuras demandas dos usuários.

Ao coletar feedback e dados de usuários atuais por meio de métodos como entrevistas e pesquisas, ajustamos nosso conceito inicial para garantir que o aplicativo oferecesse soluções eficazes para os desafios enfrentados pelos usuários (27, 35). Assim, alguns conceitos foram implementados para a pesquisa de mercado.

1. Existem Apps com programa de benefício ao usuários?
2. Existem Apps com proposta de educação no trânsito em função do comportamento do condutor?
3. Existem Apps com proposta de benefícios em prêmio de seguro veicular?

Para responder as perguntas, foram realizadas pesquisa na loja Google Play store, principal loja de aplicativos android, essa pesquisa nos mostrou os seguintes resultados demonstrados na tabela 3.17.

Tabela 3.17: Pesquisa de Mercado

perguntas	Quantidade
Existem Apps com programa de benefício ao usuários?	247
Existem Apps com proposta de educação no trânsito em função do comportamento do condutor?	248
Existem Apps com proposta de benefícios em prêmio de seguro veicular?	1

A pesquisa de mercado apontou um possível concorrente no país de Camarões, no qual lançou o aplicativo na Loja em 01 de março de 2024. Ainda assim, estamos em um mercado de oceano azul em virtude do não conhecimento dos usuários e os serviços ofertados pelo concorrente ser de cunho local. Identificamos pontos fortes e fracos e buscamos diferenciais competitivos (36). Isso nos permitirá posicionar a proposta de nosso aplicativo de maneira única no mercado, oferecendo características distintivas que o destacam da concorrência.

Além disso, exploramos tendências emergentes no setor, como discutido por Wen et al. em (40), para garantir que a proposta de nosso aplicativo estivesse alinhado com as expectativas em evolução dos usuários. A compreensão dessas tendências nos permitiu antecipar mudanças no comportamento do consumidor e adaptar a proposta de nosso aplicativo de acordo com as necessidades dos possíveis clientes.

A pesquisa de mercado também serviu como base para a definição de estratégias de monetização e a identificação de novas oportunidades de negócios (41, 37). Ao entender as dinâmicas do mercado, podemos tomar decisões mais informadas sobre modelos de receita e maximizar o potencial de crescimento da proposta de nosso aplicativo, além de atrair possibilidades da contratação de serviços de seguros veiculares pelo próprio aplicativo, porém este tema não será abordado neste trabalho.

3.1.3 Análise de soluções

A análise de soluções para a proposta de nosso aplicativo implica na avaliação criteriosa de diversas abordagens como as tecnologias e frameworks disponíveis. Este processo se baseou em uma pesquisa extensiva de ferramentas de desenvolvimento, plataformas, linguagens de programação e estruturas de backend, buscando a combinação ideal que melhor se alinhe com os requisitos e objetivos específicos do projeto (42).

Ao considerar a amplitude das opções disponíveis, levamos em conta não apenas a funcionalidade imediata, mas também aspectos essenciais como escalabilidade, segurança, facilidade de manutenção e integração com outras soluções. Para garantir a qualidade do produto final e proporcionar uma experiência positiva aos usuários, essa análise é conduzida com um foco específico nas necessidades e metas da proposta de nosso aplicativo.

A escolha de tecnologias específicas, como frameworks de desenvolvimento mobile e estratégias de arquitetura, é embasada em referências sólidas da indústria, como Clean Architecture (32) e Model–View–Presenter (30, 43). Esses referenciais não apenas orientam a estruturação do código, mas também asseguram a modularidade e a escalabilidade do aplicativo.

Ao considerar a interface do usuário, adotamos padrões de design centrado no usuário (39) e métodos de avaliação de usabilidade (44) para garantir que a experiência do usuário seja intuitiva e agradável. Essas práticas são essenciais para diferenciar nosso aplicativo no mercado competitivo de aplicativos móveis.

A integração de APIs, conforme discutido por Jacobson et al. em (36), é uma parte da nossa análise de soluções. A escolha de APIs estratégicas e a implementação eficiente da comunicação entre diferentes componentes do aplicativo são considerações essenciais para a funcionalidade e a eficácia do produto final.

A análise de soluções, portanto, vai além da simples escolha de tecnologias; é um processo estratégico que molda a base do nosso aplicativo, assegurando que cada decisão contribua para a criação de um produto de alta qualidade e que atenda às expectativas dos usuários (40). Este enfoque criterioso é fundamental para o sucesso a longo prazo do trabalho. Na tabela 3.18 visualizamos as tecnologias escolhidas para um futuro desenvolvimento.

Tabela 3.18: Tecnologias escolhidas para futuro desenvolvimento

Tecnologias	Aplicações escolhidas
Ambiente de desenvolvimento	Android Studio
Frameworks de desenvolvimento mobile	Rails
Sistema Operacional	Android
Linguagens de programação	Kotlin
Arquitetura de back end	REST
Desenvolvimento back end	Firebase
Protocolo de segurança	Hypertext Transfer Protocol Secure
Banco de dados	Google Cloud SQL
API	Apigee API Platform
Motor	Google Engine

3.1.4 Definir público-alvo e personas

Definir o público-alvo e criar personas são passos importantes para moldar a proposta de desenvolvimento do nosso aplicativo de maneira eficaz. Essa prática envolve a construção de perfis detalhados que representam os potenciais usuários, levando em consideração uma variedade de características demográficas, comportamentais, necessidades, desejos e dores específicas. Ao implementar esse processo, buscamos uma compreensão profunda dos usuários-alvo para orientar de forma mais precisa as funcionalidades e proporcionar uma experiência personalizada.

A criação de personas é um elemento-chave neste contexto, permitindo uma representação visual e mais tangível dos nossos usuários ideais. Ao detalhar características como idade, profissão, comportamentos online, e motivações específicas, as personas proporcionam uma visão mais holística do público-alvo. Isso, por sua vez, orienta não apenas o desenvolvimento, mas também o design, a usabilidade e as estratégias de marketing do aplicativo de maneira mais focada e efetiva.

Ao abordar essa etapa, devemos considerar referências da indústria que se concentram na importância do design centrado no usuário (39) e em métodos de avaliação de usabilidade (44). Esses princípios garantem não apenas uma compreensão profunda das necessidades do público, mas também direcionam as decisões de design para uma experiência de usuário mais intuitiva e satisfatória.

Além disso, insights obtidos com uma pesquisa de lacuna de mercado e pesquisa mercado detalhada (27) contribuem significativamente para a criação de personas mais precisas. Compreender as tendências do setor, as preferências dos usuários e as lacunas no mercado enriquece a construção desses perfis, proporcionando uma base sólida para decisões informadas ao longo do desenvolvimento do aplicativo.

Ao definir o público-alvo e criar personas, o processo se torna mais do que uma simples identificação de usuários. Torna-se uma estratégia orientada por dados e referências sólidas, conforme destacado por Jacobson et al. em (36) e Lattin et al. (29). Essa abordagem, portanto, não apenas contribui para o desenvolvimento inicial, mas também se torna uma ferramenta valiosa para ajustes contínuos, alinhando-se dinamicamente às mudanças nas necessidades e expectativas dos usuários ao longo do tempo.

Dessa forma, a definição do público-alvo e a criação de personas não são apenas passos iniciais, mas sim um processo contínuo e integrado que permeia todas as fases do desenvolvimento do aplicativo. Isso garante uma abordagem adaptativa e centrada no usuário, essencial para o sucesso duradouro do projeto.

Baseado nas informações adquiridas e já consolidadas, passamos para a estruturação das informações coletadas, definindo dois tipos de personas e um perfil de público-alvo. na figura 3.4 temos as personas e na figura 3.5 o público-alvo.

3.2 LEVANTAMENTO DE REQUISITOS

Já no que se relaciona a construção do aplicativo que irá abordar o sistema geral é necessário que se tenha, seguindo o processo de engenharia de software um levantamento de requisitos, para Capturas as necessidades do cliente antes de projetar o desenvolvimento. Nesse sentido, o processo de levantamento de requisitos para a proposta de desenvolvimento de aplicativo demanda uma abordagem sistemática para



Figura 3.4: Persona 1 e 2
Fonte: autoria própria



Figura 3.5: Público-Alvo
Fonte: autoria própria

identificar e documentar minuciosamente as necessidades, funcionalidades e expectativas do software a ser construído. Esta fase abrange a coleta de informações tanto sobre requisitos funcionais, que descrevem as ações específicas que o aplicativo deve executar, quanto sobre requisitos não funcionais, que incluem aspectos de qualidade, desempenho e segurança, bem como a interação com o usuário e integrações necessárias.

Para garantir um levantamento de requisitos abrangente, é imperativo realizar reuniões, workshops ou entrevistas com stakeholders e usuários finais. Essas interações aprofundadas permitem uma compreensão holística dos fluxos de trabalho, preferências de interface, desafios e expectativas dos usuários. Essa abordagem é apoiada por autores como Jacobson et al. em (36), que enfatizam a importância de envolver ativamente os stakeholders no processo de levantamento de requisitos, porém devidos ao tempo e recursos,

não foram realizados.

A documentação resultante desse processo não apenas fornece um guia detalhado para a propostas de desenvolvimento do aplicativo, mas também serve como uma referência essencial durante a implementação e para evitar retrabalhos ou divergências ao longo do projeto. Autores como Pau et al. em (43) destacam a relevância de padrões de design, como o modelo view presenter (MVP), na organização e implementação eficientes de requisitos.

Durante o levantamento de requisitos, é benéfico incorporar práticas de usabilidade e design centrado no usuário (39). Isso contribui para a criação de interfaces mais amigáveis e intuitivas, garantindo que o aplicativo atenda não apenas aos requisitos técnicos, mas também às expectativas dos usuários.

A integração de feedback de métricas de usabilidade, conforme sugerido por das Virgens e Silveira em (45), pode enriquecer ainda mais o levantamento de requisitos, proporcionando uma compreensão aprofundada das preferências e desafios enfrentados pelos usuários.

Ao adotar uma abordagem baseada em referências sólidas e melhores práticas, o levantamento de requisitos não é apenas uma etapa burocrática, mas sim um componente essencial para o sucesso do projeto. Ao orientar todo o processo de desenvolvimento, desde a concepção até a implementação, essa fase estabelece as bases para a criação de um aplicativo Android eficiente, seguro e altamente funcional. A figura 3.6 representa esquematicamente os requisitos levantados para a proposta desenvolvida neste trabalho.

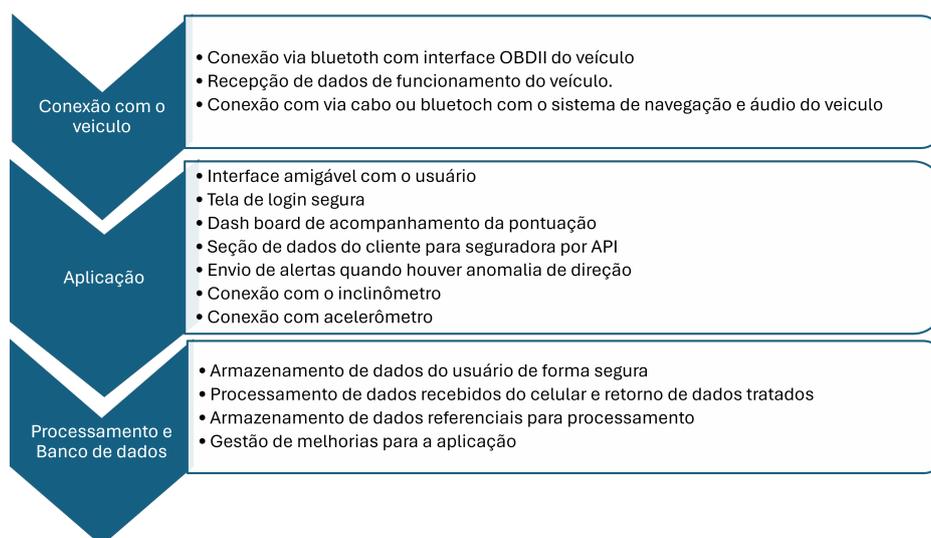


Figura 3.6: Levantamento de Requisitos

Fonte:autoria própria

3.2.1 Realizar workshops ou reuniões para capturar requisitos

Ao realizar workshops e reuniões para a captura de requisitos em aplicativos Android, é fundamental adotar uma abordagem interativa e colaborativa. Iniciar o processo com reuniões permite uma troca direta de informações entre os desenvolvedores, designers, stakeholders e usuários finais. Essa interação direta desde o início, conforme proposto por Jacobson et al. em (36), estabelece uma base sólida para compreender não apenas as funcionalidades desejadas, mas também as metas estratégicas e as expectativas globais

para o aplicativo.

Durante as entrevistas detalhadas realizadas nessas reuniões, busca-se identificar não apenas as ações específicas que o aplicativo deve realizar, ou seja, os requisitos funcionais, mas também explorar aspectos comportamentais e expectativas do usuário. Essa abordagem alinhada com García em (30) assegura que a Model–View–Presenter (MVP) seja implementada com clareza, definindo não apenas as características técnicas, mas também as necessidades do usuário final.

O uso de sessões de brainstorming é uma prática valiosa para estimular a criatividade e a inovação. Com base nos princípios universais do design, como destacado por Lidwell et al. em (35), essas sessões buscam gerar ideias inovadoras e identificar requisitos que podem não ser óbvios à primeira vista. A diversidade de perspectivas fornecida por diferentes partes interessadas enriquece o processo, garantindo uma visão mais abrangente.

A análise de casos de uso complementa essa abordagem, permitindo mapear os fluxos de trabalho dos usuários dentro do aplicativo. Essa prática, respaldada por Pau et al. em (43), na aplicação do Model View Presenter (MVP), ajuda a compreender melhor os cenários de uso específicos e os pontos críticos de interação. Isso é crucial para criar uma experiência de usuário intuitiva e eficiente.

A captura de requisitos não se limita à funcionalidade; é estendida para a compreensão das restrições técnicas, requisitos de segurança e desempenho. A consideração desses aspectos está alinhada com a abordagem de testes automatizados, como destacado por Freitas em (34). Integrar considerações de qualidade desde o início do processo garante que o aplicativo seja robusto e eficiente.

A comunicação eficaz entre desenvolvedores, designers e stakeholders é um pilar fundamental. O modelo de arquitetura limpa proposto por Martin em (32) destaca a importância dessa comunicação para evitar mal-entendidos e garantir que todas as necessidades sejam capturadas de maneira precisa. Essa transparência é crucial durante todas as fases do desenvolvimento.

Em resumo, a condução eficaz de workshops e reuniões não apenas visa identificar requisitos, mas também promove um entendimento profundo das metas estratégicas, expectativas dos usuários e restrições técnicas. Essa abordagem colaborativa estabelece uma base sólida para o sucesso do projeto, garantindo que o aplicativo Android seja desenvolvido de maneira precisa e alinhada às necessidades reais dos usuários.

3.2.2 Criar documentos detalhados de requisitos

Na etapa de criação de documentos detalhados de requisitos para aplicativos, é imperativo adotar uma abordagem metódica para garantir que todos os aspectos cruciais do software sejam abordados de maneira abrangente. Os documentos gerados desempenharão um papel central ao orientar o desenvolvimento, proporcionando uma visão unificada para toda a equipe envolvida. Inspirando-se em metodologias de design e desenvolvimento bem estabelecidas, a criação desses documentos segue diretrizes específicas.

O primeiro passo envolve a descrição completa dos requisitos funcionais e não funcionais. Isso inclui a identificação clara das ações específicas que o aplicativo deve realizar, como proposto por Figueiredo em (46). Os requisitos não funcionais, abrangendo aspectos de qualidade, desempenho e segurança, são

também detalhadamente delineados, seguindo as melhores práticas destacadas por Patni em (37) em relação a APIs RESTful.

A especificação da interface do usuário é uma peça fundamental desses documentos. Seguindo as diretrizes de design centrado no usuário, conforme apresentado por Lidwell et al. em (35) e Lowdermilk em (39), os documentos detalham as expectativas em termos de layout, interatividade e experiência geral do usuário. A aplicação de padrões de arquitetura, como o Model–View–Presenter (MVP) proposto por García em (30), é incorporada nessa fase para garantir uma implementação coesa.

A inclusão de casos de uso detalhados é uma prática essencial para mapear os diversos cenários de interação do usuário com o aplicativo. Isso segue a lógica de MVP, onde os casos de uso ajudam a definir as relações entre modelos, visões e apresentadores (30). Fluxos de trabalho e diagramas de processo são utilizados para visualizar a dinâmica do aplicativo em situações específicas, contribuindo para uma compreensão aprofundada do comportamento esperado.

Considerando a abordagem de comunicação eficaz entre as partes interessadas e a equipe de desenvolvimento, os documentos também servem como um canal formal para revisões e ajustes. Manter a documentação precisa e detalhada é vital para evitar retrabalhos e divergências ao longo do ciclo de desenvolvimento, como destacado por Jacobson et al. em (36).

Referências específicas a padrões de arquitetura, boas práticas de design e metodologias de desenvolvimento garantem que a equipe esteja alinhada com as últimas tendências e abordagens na indústria de desenvolvimento de software (32, 30). Dessa forma, a criação de documentos de requisitos se torna mais do que um simples exercício burocrático; é uma prática orientada para resultados que impulsiona a eficiência e a eficácia durante todo o ciclo de vida do projeto.

3.2.3 Priorizar funcionalidades e recursos

Na elaboração da priorização de funcionalidades e recursos para o desenvolvimento do aplicativo Android, devemos estabelecer uma abordagem estratégica que otimize os esforços da equipe de desenvolvimento, concentrando-se nas áreas que mais impactarão a experiência do usuário e a proposta de valor do aplicativo.

Começando com a identificação das funcionalidades essenciais, é importante compreender profundamente as necessidades dos usuários e garantir que o aplicativo atenda aos requisitos básicos. Aqui, a aplicação da Matriz de Priorização MoSCoW, sugerida por Biehl em (42), torna-se uma ferramenta valiosa. Categorizando as funcionalidades em Must have, Should have, Could have e Won't have, a equipe pode determinar quais são críticas para o lançamento inicial e quais podem ser consideradas em fases subsequentes. Na figura 3.7 demonstra todas as priorizações da proposta de aplicativo.

Além disso, é imperativo incorporar uma mentalidade ágil ao processo de desenvolvimento. A análise contínua do feedback dos usuários, conforme recomendado por Demico et al. em (47), proporciona insights valiosos que podem influenciar a reavaliação constante das prioridades. Essa abordagem iterativa permite uma resposta dinâmica às mudanças nas preferências e expectativas dos usuários, garantindo que o aplicativo permaneça relevante e eficaz ao longo do tempo.

M	S	C	W
Must have	Should have	Could have	Won't have
<ul style="list-style-type: none"> • Segurança de dados; • Conexão Bluetooth com conector OBDII; • Conexão online com servidor; • Dash board de score para o usuário; • Informações relevantes para o usuário. • API de disponibilização de dados para análise de seguradora veicular sobre a forma de dirigir do usuário. 	<ul style="list-style-type: none"> • Conexão para extração de dados do veículo via sistema de entretenimento por Bluetooth. • Inteligência artificial para processamento de dados; 	<ul style="list-style-type: none"> • Conexão com sistemas de contratação de seguros veiculares; • Visão computacional para aumento de dados; • Inteligência artificial para análise comportamental do usuário por dados. 	<ul style="list-style-type: none"> • API com o departamento de trânsito para infrações automáticas; • Geração de provas criminais contra o usuário.

Figura 3.7: Matriz de priorização MOSCOW
 Fonte: autoria própria

No âmbito do desenvolvimento, a referência a padrões de arquitetura, como o Model–View–Presenter (MVP), conforme discutido por García em (30), e a consideração dos princípios universais do design de Lidwell et al. em (35) são essenciais. Esses padrões e princípios não apenas orientam a implementação técnica, mas também contribuem para a criação de uma experiência de usuário coesa e intuitiva.

Ao longo desse processo, manter-se atualizado com as tendências do mercado, como destacado por Gomes em (27), e adaptar as prioridades com base em pesquisas de mercado contínuas, pode ser fundamental. Essa abordagem orientada por dados permite que a equipe antecipe as demandas do mercado, respondendo proativamente às mudanças nas preferências dos usuários.

A literatura sobre clean architecture, como proposto por Martin em (32), e as práticas recomendadas para o design de APIs, conforme apresentado por Jacobson et al. em (36), também oferecem orientações valiosas durante a fase de priorização. Esses recursos fornecem diretrizes sólidas para estruturar o código-fonte do aplicativo e garantir a eficiência e escalabilidade do sistema.

A priorização de funcionalidades e recursos no desenvolvimento de aplicativos Android é um processo multidimensional que combina metodologias de gestão ágil, feedback do usuário contínuo, considerações arquiteturais sólidas e adaptação às tendências do mercado. Essa abordagem holística é essencial para criar um aplicativo robusto, centrado no usuário e alinhado às expectativas em constante evolução.

3.3 ARQUITETURA E DESIGN

O desenvolvimento de aplicativos Android é uma tarefa complexa que exige cuidado especial com a arquitetura e o design para garantir o sucesso do projeto. Neste contexto, a implementação prática desses elementos desempenha um papel vital, influenciando diretamente na usabilidade, desempenho e manutenibilidade do aplicativo. Considerando não apenas a estrutura arquitetônica do código, mas também

os aspectos visuais e interativos, os desenvolvedores podem criar aplicativos que não apenas atendam às expectativas dos usuários, mas também se destaquem no competitivo mercado de desenvolvimento mobile. A figura 3.8 demonstra a arquitetura básica da proposta de desenvolvimento do aplicativo.



Figura 3.8: Arquitetura Básica
Fonte: autoria própria

A escolha da arquitetura certa é o primeiro passo crucial no desenvolvimento de aplicativos Android. Duas abordagens populares são o Modelo-View-ViewModel (MVVM) e Clean Architecture. Ambas oferecem estruturas organizadas que promovem a modularidade, testabilidade e escalabilidade do código.

3.3.1 Definir a arquitetura do aplicativo

Na definição da arquitetura para nosso aplicativo Android, é crucial escolher uma estrutura que otimize a organização do código e dos principais componentes do software. Diversas arquiteturas estão disponíveis, cada uma com suas particularidades e vantagens. Entre as opções mais comuns, destacam-se:

- **Model-View-ViewModel (MVVM):** Essa arquitetura proporciona uma clara separação entre a lógica de negócios (Model), a interface do usuário (View) e a lógica intermediária (ViewModel). Essa separação favorece a testabilidade, manutenção e escalabilidade do código (48).
- **Model-View-Presenter (MVP):** Similar ao MVVM, o MVP apresenta uma separação mais direta

entre a View e o Presenter, simplificando alguns aspectos de teste e tornando a lógica de apresentação mais independente da interface do usuário (30).

- **Clean Architecture:** Esta abordagem propõe uma estruturação baseada em camadas, com níveis de abstração bem definidos. Isso separa as preocupações de negócios das questões de implementação técnica, garantindo flexibilidade e testabilidade (32).

A escolha da arquitetura dependerá das necessidades específicas do projeto, das preferências da equipe de desenvolvimento e da escalabilidade esperada. Cada uma dessas arquiteturas oferece vantagens distintas em termos de manutenção, teste e escalabilidade. É crucial selecionar aquela que melhor se alinha com os objetivos do aplicativo e a capacidade da equipe de desenvolvimento.

Ao implementar a arquitetura escolhida, é essencial referenciar literatura especializada, como os trabalhos de Hoefling em (48) e Martin em (32), para garantir uma aplicação consistente dos princípios arquitetônicos. A compreensão profunda desses modelos arquitetônicos permitirá uma implementação mais eficaz, levando em consideração as nuances específicas do desenvolvimento Android.

Além disso, a escolha da arquitetura também impacta diretamente na experiência do usuário, uma vez que a organização do código reflete na usabilidade e desempenho do aplicativo (39). Portanto, a integração de princípios de design centrados no usuário, conforme abordado por Lidwell et al. em (35), complementa a escolha da arquitetura, garantindo uma experiência do usuário agradável e intuitiva. A figura 3.9 define a arquitetura detalhada da proposta do aplicativo.

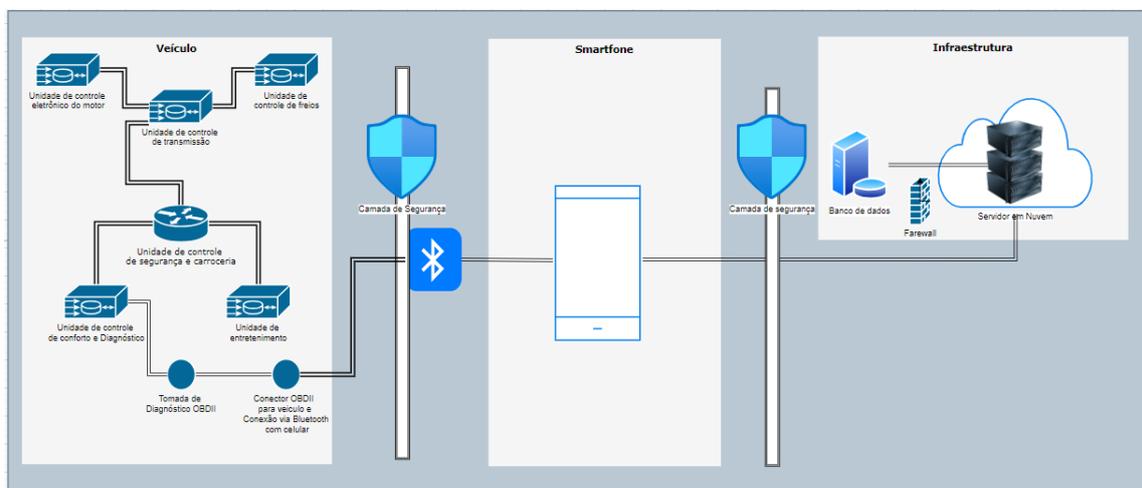


Figura 3.9: Arquitetura detalhada da proposta do Aplicativo

Fonte: autoria própria

3.3.2 Criar protótipos de baixa fidelidade para esboçar a interface

Quando iniciarmos o desenvolvimento do aplicativo, criaremos um protótipo de baixa fidelidade, com abordagem inicial e fundamental para esboçar a interface do usuário. Utilizando ferramentas simples de design ou até mesmo papel e caneta, esses protótipos são uma maneira eficaz de visualizar rapidamente ideias, testar conceitos de interface e validar o fluxo de navegação do aplicativo.

Durante a fase de criação dos protótipos, é essencial manter o foco na essência das funcionalidades, sem se preocupar demasiadamente com detalhes visuais complexos. Ao seguir as diretrizes do Material Design (39), podemos garantir que a disposição dos elementos na tela siga padrões visuais consistentes, proporcionando uma experiência de usuário intuitiva.

A criação de protótipos de baixa fidelidade não apenas agiliza o processo de concepção, mas também facilita a identificação precoce de problemas de usabilidade. Referências como as métricas de usabilidade em protótipos computadorizados de baixa fidelidade (45) podem ser úteis nesse estágio, permitindo ajustes rápidos antes de avançar para fases mais avançadas de implementação.

Além disso, a comunicação efetiva entre membros da equipe de desenvolvimento, designers e stakeholders é viabilizada pelos protótipos de baixa fidelidade. Referências como "MVP: Model-View-Presenter"(30) oferecem insights valiosos sobre a interação entre diferentes componentes do sistema, auxiliando na discussão de ideias de forma mais concreta.

Ao realizar interações rápidas com base nos feedbacks obtidos a partir dos protótipos, a equipe pode refiná-los continuamente. Isso contribui para um alinhamento mais preciso das expectativas e requisitos antes de investir recursos significativos na implementação completa da interface do aplicativo.

Os protótipos de baixa fidelidade são uma ferramenta essencial no início do desenvolvimento, permitindo uma visualização rápida, testes de conceitos e validação de fluxo. Integrar boas práticas de design, considerar métricas de usabilidade e utilizar referências relevantes são passos cruciais nesse processo (34, 35).

3.3.3 Desenvolvimento Frontend

No âmbito do desenvolvimento frontend, assumimos a responsabilidade de construir a interface direta com a qual os usuários interagem no nosso aplicativo. Nesse contexto, a criação de elementos visuais, como layouts, botões e formulários, é realizada utilizando tecnologias fundamentais como HTML, CSS e JavaScript. Inspirados nos princípios da Clean Architecture (32), nossa abordagem vai além da estética, priorizando a usabilidade e a experiência do usuário. É crucial que a interface seja intuitiva, responsiva e acessível em diversos dispositivos, atendendo assim às expectativas dos usuários e refletindo a identidade da marca.

Ao embarcar nesse processo de design frontend, será necessário seguir as diretrizes recomendadas pelo Android, especialmente aquelas propostas pelo Material Design. Tais diretrizes são fundamentais para garantir consistência visual, oferecer familiaridade aos usuários e facilitar a navegação por meio de componentes visuais específicos (35). Incorporar essas práticas não apenas eleva a estética do aplicativo, mas também contribui significativamente para a eficácia e eficiência da interação usuário-aplicativo.

A preocupação com a acessibilidade é uma consideração primordial durante o processo de design frontend. Assegurar que o aplicativo seja inclusivo, acessível e capaz de se adaptar a diversas necessidades dos usuários é um princípio orientador. Essa abordagem garante que a experiência proporcionada seja agradável para todos os usuários, independentemente de suas habilidades ou preferências (39).

Ao implementar as práticas de design frontend, é importante considerar as métricas de usabilidade,

conforme discutido por Ferreira et al. em (33), para garantir que o design não apenas seja esteticamente atraente, mas também altamente funcional e eficiente. Testes automatizados, conforme abordados por Freitas em (34), podem ser incorporados para verificar a consistência do design em diferentes dispositivos Android, garantindo uma experiência uniforme em diversas plataformas.

A integração harmoniosa do frontend com a arquitetura geral do aplicativo, como discutido por Martin em (32), é necessário garantir a coesão e a eficácia do desenvolvimento como um todo. O entendimento profundo da interação entre o frontend e outros componentes, como APIs (36), é crucial para garantir uma experiência de usuário sem falhas.

Portanto, ao implementar o design frontend em aplicativos Android, a equipe de desenvolvimento deve se esforçar para equilibrar a estética com a funcionalidade, incorporando as diretrizes do Material Design, garantindo acessibilidade e realizando testes rigorosos para otimizar a experiência do usuário em todas as etapas do desenvolvimento.

3.3.4 Desenvolver uma interface do usuário (UI)

No cerne do desenvolvimento frontend está a criação e implementação da Interface do Usuário (UI). Esse processo vai muito além da disposição de elementos na tela; envolve escolhas fundamentais relacionadas a cores, tipografia, ícones e todos os detalhes visuais que compõem a experiência do usuário. Inspirados pelos "Princípios Universais do Design"(35), trabalhamos em estreita colaboração com designers para traduzir o design visual em elementos interativos. Utilizando ferramentas e frameworks específicos, garantimos consistência e usabilidade em diferentes dispositivos e resoluções de tela. Nosso objetivo é criar uma interface intuitiva, esteticamente agradável e funcional, atendendo de maneira clara e eficiente às necessidades dos usuários.

3.3.4.1 Plataforma Safe Driving

A proposta de desenvolvimento da plataforma safe driving será criar um serviço seguro para comunicação de dados furtivos sobre a localização do usuário e de seu veículo, exigindo que este serviço fosse escalável e altamente disponível através do uso do "Raise Middleware". Outro item importante será o desenvolvimento da interface de administração do usuário, os dados e a identificação de rotas a serem utilizadas(49). A Figura 3.10 representa o front-end da plataforma safe driving.

Nessa plataforma, além de apresentar as vias com todos as suas sinalizações, também será possível cadastrar novos usuários e compartilhar dados com terceiros de acordo com o desejo do usuário. Para fazer isso, o Login/ janela de registro, deve ser acessada para permitir que o usuário defina o e-mail e a senha solicitados.

O design de um aplicativo Android não é apenas sobre a aparência visual, ele desempenha um papel fundamental na experiência do usuário (UX). O design influencia a navegabilidade, intuição e interação do usuário com o aplicativo. Em um contexto prático, isso significa que, ao projetar a interface do usuário, é necessário considerar não apenas os elementos visuais, mas também a usabilidade e a experiência geral do usuário. Um design centrado no usuário busca criar uma experiência intuitiva, focando na consistência

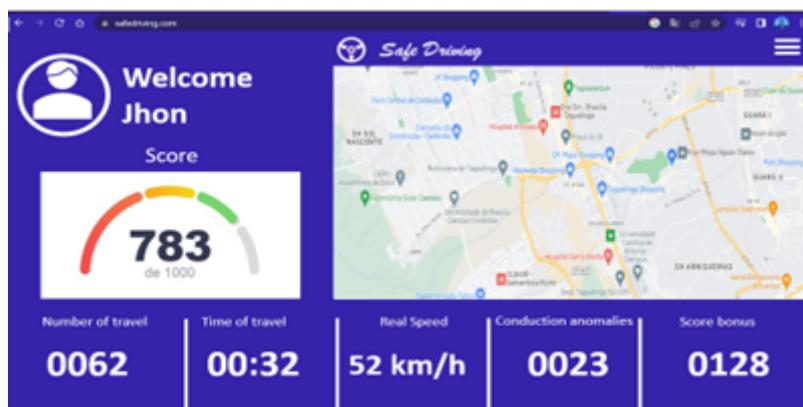


Figura 3.10: Proposta de front end da plataforma Safe Driving

Fonte: autoria própria

visual, feedbacks claros e interfaces que sigam os padrões e expectativas dos usuários (50).

3.3.5 Implementar o design visual e interações do usuário

A implementação do design visual e das interações do usuário é uma etapa crucial na jornada do desenvolvimento frontend. Essa fase não se limita à mera estética; envolve a transição do conceito de design para uma interface funcional e interativa. Focamos em implementar fielmente o design visual, incluindo layouts, cores, tipografia e elementos gráficos, bem como as interações planejadas, como animações e transições. Essa implementação cuidadosa não apenas aumenta a atratividade do aplicativo, mas também reforça a usabilidade, tornando a navegação mais intuitiva e agradável para os usuários. Seguir as melhores práticas, integrando referências como "Design Centrado no Usuário"(39), é essencial para garantir que a estética, a usabilidade e a funcionalidade se alinhem de maneira harmoniosa.

3.3.5.1 Integrar elementos visuais com lógica de aplicativo

Integrar elementos visuais com a lógica do aplicativo é um passo crucial para assegurar a coesão entre a interface do usuário e a funcionalidade do software. Não se trata apenas de exibir os elementos visuais conforme projetados, mas de associar esses elementos à lógica subjacente do aplicativo. Nossos desenvolvedores trabalham para vincular os componentes visuais, como botões, menus e campos de entrada, aos processos e funções do aplicativo. Essa integração é essencial para criar uma experiência coesa, onde a interface não apenas parece consistente com o design, mas também responde de maneira precisa e lógica às interações do usuário, proporcionando uma experiência intuitiva e funcional.

Ao longo de todo esse processo, estamos comprometidos em seguir as melhores práticas e integrar referências de qualidade. "Design Centrado no Usuário"(39) é uma dessas referências, garantindo que a estética, a usabilidade e a funcionalidade estejam em equilíbrio. A colaboração próxima entre desenvolvedores frontend e designers é crucial para criar uma experiência de usuário que não apenas atenda às expectativas, mas as supere, proporcionando uma interação agradável e eficiente com o aplicativo.

Na jornada frontend, abraçamos os princípios da Clean Architecture (32) como fundamento para nossa

abordagem de desenvolvimento. Esses princípios destacam a importância da modularidade, reutilização de código e manutenibilidade. A modularidade é incorporada na estrutura, permitindo a construção de componentes independentes e intercambiáveis. A reutilização de código é promovida, garantindo que soluções eficientes possam ser aplicadas em diferentes partes do aplicativo. A manutenibilidade é uma prioridade, facilitando atualizações e modificações futuras sem comprometer a integridade do sistema. Esses princípios não apenas guiam nosso desenvolvimento frontend, mas também garantem que estamos construindo não apenas para o presente, mas também para o futuro.

A disposição dos elementos na tela, o uso de cores e fontes, e a resposta a ações do usuário são fatores críticos para o sucesso do design de um aplicativo Android (35, 39). A proposta de aplicativo Safe Driving possui três telas principais: Login, Home e Configurações. A Tela de login será apropriada para inserir os dados de cadastro do usuário, para que toda a plataforma Safe Driving identifique e mantenha o perfil de usuário e associe os Scores definidos durante o uso da aplicação. A tela inicial do aplicativo mostrará a viagem do usuário, iniciada automaticamente quando o sinal de GPS for maior que 15 km/h. Nesta tela, o SCORE do usuário será exibido junto ao número de viagens, tempo de viagem, velocidade real do veículo, anomalia de condução e score bônus. Também serão enviadas mensagens educativas ao usuário, que podem ser lidas pelo aplicativo em viva voz para que o usuário não ter distração da condução.

Por fim, o usuário também terá acesso às configurações do aplicativo. Neste menu, será possível acessar as configurações de usuário, as configurações gerais do aplicativo realizar o compartilhamento de Scores com companhia de seguros, lojas de vendas de veículos e Órgãos judiciários, conforme a decisão do usuário, Configurações do dispositivo OBD-II e a própria redefinição do aplicativo, se o aparelho Smartphone for utilizado por outra pessoa. A figura3.11 demonstra a interface do usuário.

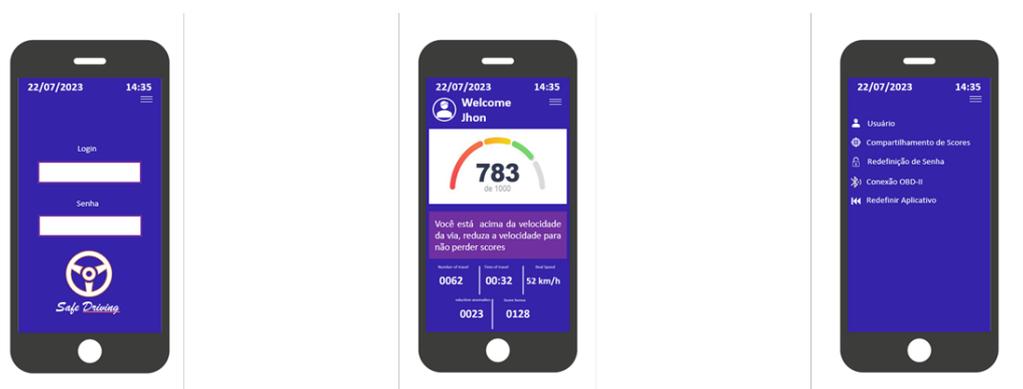


Figura 3.11: Telas de Interface do usuário

Fonte: autoria própria

A consistência visual é essencial para criar uma interface coesa e familiar aos usuários. Elementos como ícones, botões e cores devem seguir um padrão consistente em todo o aplicativo, facilitando a compreensão e a navegação intuitiva.

Feedbacks claros são fundamentais para informar os usuários sobre o resultado de suas ações. Mensagens de erro compreensíveis, animações de transição suaves e indicadores visuais ajudam os usuários a entender o que está acontecendo no aplicativo, contribuindo para uma experiência mais confiável e agradável.

A interface intuitiva é construída considerando os padrões de design do Android e as expectativas do usuário. Os desenvolvedores devem estar cientes das diretrizes de design do Android, que são atualizadas regularmente pela Google, para garantir que o aplicativo siga as práticas recomendadas (51).

Além da arquitetura e do design, métricas de usabilidade e testes automatizados são práticas no desenvolvimento de aplicativos Android. A usabilidade é um aspecto vital para a satisfação do usuário, e a aplicação de métricas ajuda a avaliar o quão fácil é para os usuários realizar tarefas específicas no aplicativo (45).

A integração de métricas de usabilidade pode envolver a realização de testes com usuários reais, a análise de dados de interação, e a coleta de feedback direto. Essas métricas fornecem insights valiosos sobre como os usuários estão interagindo com o aplicativo, identificando áreas de melhoria e validando o design e a usabilidade planejados (19).

Ao implementar a arquitetura e o design em um projeto Android, é crucial a referência constante a modelos arquitetônicos sólidos, princípios de design centrados no usuário e práticas recomendadas. A literatura especializada, como os trabalhos de García em (30) e Martin em (32), oferece insights valiosos que podem ser adaptados às necessidades específicas do projeto.

A atualização contínua do conhecimento, considerando as mudanças nas diretrizes de design do Android e a evolução das melhores práticas de desenvolvimento, é vital para garantir que o aplicativo permaneça relevante e eficaz ao longo do tempo (51). Tendo a aceitação e a satisfação com o aplicativo Android. Assim, a combinação entre uma arquitetura sólida e um design cuidadosamente planejado é essencial para o sucesso de um aplicativo, garantindo sua funcionalidade, desempenho e aceitação pelos usuários.

3.4 DESENVOLVIMENTO BACKEND

O desenvolvimento backend é a espinha dorsal de qualquer aplicativo, responsável por construir uma estrutura robusta que gerencia a lógica de negócios, o processamento de dados e a comunicação eficiente com o banco de dados. Para garantir a eficácia dessa estrutura, é crucial abordar aspectos como arquitetura, escolha de linguagens e frameworks, bem como a criação de APIs para permitir a interação harmoniosa entre o frontend e o backend.

Ao ingressar nesse processo, a definição da arquitetura assume uma importância central. Inspirados pelos princípios da Clean Architecture (32), buscamos uma abordagem que promova a modularidade, reutilização de código e manutenibilidade. A modularidade é vital para a construção de componentes independentes e intercambiáveis, permitindo uma evolução flexível do sistema. A reutilização de código assegura a eficiência em diferentes partes do aplicativo, enquanto a manutenibilidade facilita atualizações e modificações sem comprometer a integridade do sistema.

A escolha das linguagens e frameworks é uma decisão estratégica, pois influencia diretamente no desempenho e na escalabilidade do sistema. Referências como "Full Stack JavaScript: Learn Backbone.js, Node.js, and MongoDB"(38) oferecem insights valiosos sobre a integração eficaz entre o frontend e o backend. Além disso, ao considerar a criação de APIs, referências como "RESTful Api Design"(42) fornecem

diretrizes cruciais para desenvolver interfaces de programação eficientes e escaláveis.

A preocupação com a segurança dos dados é uma constante durante todo o desenvolvimento backend. A implementação de boas práticas de codificação, como a separação de preocupações e a aplicação adequada de padrões de projeto, contribui para a criação de uma estrutura robusta. A separação de preocupações, por exemplo, envolve a divisão da lógica do aplicativo em camadas distintas, como a camada de acesso a dados, a lógica de negócios e a camada de apresentação. Esse enfoque facilita a colaboração entre desenvolvedores e permite uma manutenção mais fácil ao longo do tempo.

A aplicação de medidas de segurança robustas é imperativa para proteger as informações sensíveis manipuladas pelo aplicativo. Isso inclui autenticação, autorização e criptografia de dados. Referências como "Pro RESTful APIs"(37) abordam os desafios e as melhores práticas associadas à criação de APIs seguras, desempenhando um papel importante na construção de um backend resiliente.

Além disso, a estrutura do backend deve ser construída com flexibilidade em mente, capaz de acomodar atualizações, integrações com serviços externos e adaptações às mudanças nas necessidades do aplicativo ao longo do tempo. A referência constante a padrões de projeto, como o Model View Presenter (43), oferece insights sobre como estruturar o backend para se adaptar a mudanças sem comprometer a estabilidade.

A implementação de boas práticas de codificação e a escolha cuidadosa das ferramentas certas são fundamentais para garantir a qualidade do desenvolvimento backend. Princípios como a Clean Architecture (32) não apenas orientam a estrutura, mas também promovem a construção de um sistema que não apenas atenda, mas supere as expectativas dos usuários.

3.4.1 Parâmetros de Processamentos de Anomalias

Para a definição de parâmetros para processamento de anomalias de condução, utilizaremos definiremos os cruzamentos de dados obtidos dos veículos e do Smartphone do usuário. De acordo essa definição, serão formadas as anomalias de condução. Essas anomalias de condução terão scores específicos e conforme a realização de cada evento, será debitado o valor de seu score do usuário durante 24h, associados a data e fuso horário definidos no Smartphone do usuário. Assim, esse algoritmo matemático tem início as 00:00:00 e término as 23:59:59 de cada dia. Neste momento será executado outro algoritmo matemático da média de pontuação obtida nos dias de uso. A tabela 3.19 demonstra a organização matemática deste algoritmo.

Outro parâmetro relevante é o Score bônus. Como o próprio nome já diz, é um bônus por o usuário ter excelente desempenho durante sete dias seguidos de uso do aplicativo, ou seja, sem perda de scores. Assim, ele tem 100 pontos somados ao seu score total, o que possibilita a reeducação no trânsito, motivando o usuário a se autodisciplinar para uma condução segura. É relevante que o score total não poderá passar de 1000 e que os scores bônus não são cumulativos. A tabela 3.20 demonstra a organização matemática deste algoritmo.

Definindo as condições matemáticas, vamos definir as condições de anomalias de condução e seus Scores específicos para o funcionamento do aplicativo. Aqui vamos listar cinco situações que podem ser consideradas inseguras durante a condução, para relacionarmos aos dados disponíveis e formatar parâ-

Tabela 3.19: Algoritmo 1 - parâmetros para Anomalias de condução

Parâmetros para Anomalias de condução

$$S_d = 1000 - (an + an_1 + an_2 + \dots + an_n)$$

Onde
 S_d =Score diário
 an =Anomalia

$$S_t = \frac{(S_{d1} + S_{d2} + \dots + S_{dn})}{d_t}$$

Onde
 S_t =Score Total
 S_d =Score diário
 d_t =Dias de uso total

Tabela 3.20: Algoritmo 2 – Parâmetros de bônus de Score

Parâmetros de bônus de Score

$$S_t = S_t + S_b$$

Onde
 S_t =Score Total
 S_b =Score bônus

Se
 $S_t > 1000$

Então
 $S_t = 1000$

Porém, Se
 $S_t \leq 1000$

Então
 S_t =Valor atual de $S_t + S_b$

metros pertinentes a cada anomalia. Sabemos que temos diversas situações que geram insegurança no trânsito, como falhas mecânicas ou elétricas, distração do condutor ou ultrapassagens arriscadas. Porém, nos limitamos as sete condições pois são possíveis de gerar parâmetros facilmente com os dados coletados do veículo.

- Velocidade do veículo acima do permitido na via: A plataforma Safe Driving recebe os dados de

velocidade diretamente do GPS do Smartphone do usuário e irá comparar com os dados de mapa das vias. Ela irá comparar a velocidade máxima especificada na via e realizará, automaticamente, o débito de scores do usuário. O seu débito de score equivalente está discriminado na tabela 1.

- Frenagens bruscas: A plataforma safe driving irá receber dados do acelerômetro e do GPS do Smartphone do usuário somados os dados de diferença de velocidades das rodas do veículo. Assim a plataforma calcula a velocidade de desaceleração e a condição de segurança na frenagem a que o veículo foi exposto. Se a distância medida pelo acelerômetro D_a for menor que 60% da distância D_p calculada pela equação a seguir, então, temos uma frenagem brusca. A tabela 3.21 demonstra o terceiro parametro da proposta de App.

Tabela 3.21: Algoritmo 3 – Parâmetros pertinentes a cada anomalia.

Parâmetros pertinentes a cada anomalia.

$$D_p = \frac{v^2}{250\mu}$$

Onde

v = velocidade em Km/h do veículo;

D_p = a distância de frenagem e;

μ = coeficiente médio de atrito do pneu com o solo

Se

$D_a > D_p * 0.6$

Então

frenagem normal.

Porém, Se

$D_a \leq D_p * 0.6$

Então

frenagem brusca.

A plataforma safe driving irá comparar as distâncias e realizará automaticamente o débito de scores do usuário. A seguir vamos detalhar cada uma das lógicas implementadas para os algoritmos.

- Aceleração bruscas: A plataforma Safe Driving irá receber dados de acelerômetro, GPS do Smartphone do usuário somados a rotação do motor e diferença de velocidade das rodas do veículo. Assim a plataforma calcula a velocidade de aceleração e a condição que o veículo foi exposto. Se a rotação do motor for maior que 3500 rpm, o sistema de tração for acionado em função da diferença de velocidade das rodas, logo se tem tentativa de deslocamento rápido em espaço pequeno. Assim, será considerado uma anomalia, realizando automaticamente o débito de scores do usuário. A figura 3.12 mostra em gráficos a diferença entre as duas situações possíveis.

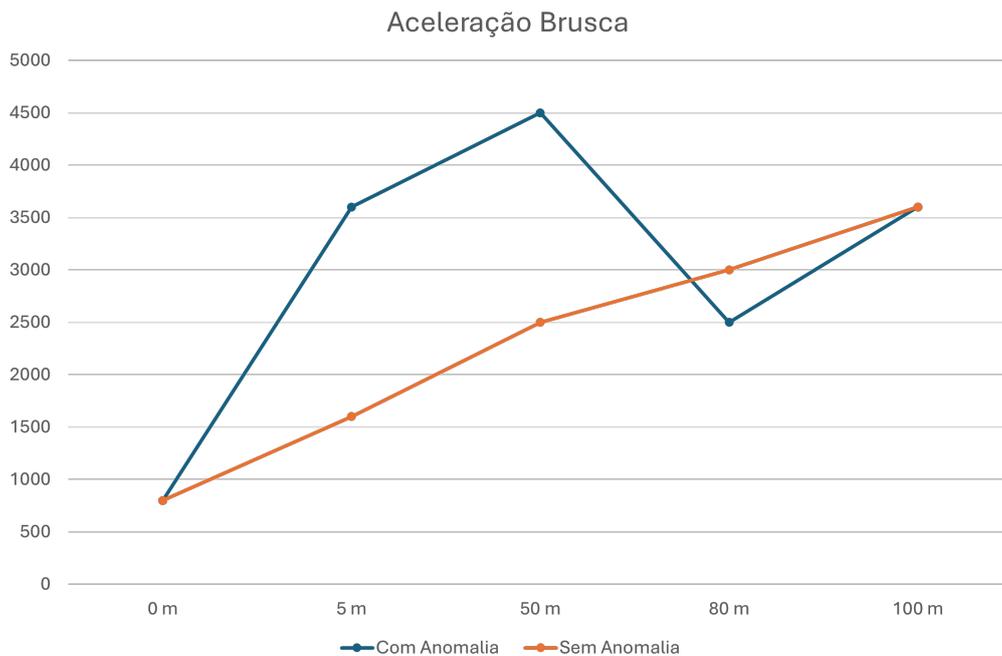


Figura 3.12: Gráfico de Aceleração bruscas
Fonte: autoria própria

- Alta Velocidade em curvas: A plataforma Safe Driving irá receber dados do GPS, do sensor de orientação do Smartphone do usuário e somadas as informações de mapa da via. Assim a plataforma irá comparar a velocidade do veículo, o ângulo de inclinação, a velocidade máxima permitida naquele ponto da via e entenderá que o usuário está conduzindo o veículo com risco de capotamento ou perda de controle em curvas e considerará uma anomalia (52). Então, ela realizará automaticamente o débito de scores do usuário. A figura 3.13 mostra a representação de inclinações bruscas durante a condução do veículo.
- Tempo de condução maior que quatro horas sem pausa de quinze minutos para descanso: A plataforma receberá dados do GPS e de relógio provenientes do Smartphone do usuário, calculando o tempo de início da viagem até quatro horas depois, se não houver velocidade em km/h igual a zero por quinze minutos. Caso o usuário ultrapasse as quatro horas sem pausa, a plataforma considerará uma anomalia e realizará um débito de Score. Ressaltamos que, quinze minutos antes do tempo máximo para descanso, a plataforma irá emitir uma mensagem de alerta, solicitando a parada do usuário o quanto antes. Caso o tempo for ultrapassado e não houver condições de parada, a plataforma irá disparar uma mensagem de local de parada, conforme locais indicados em seu mapa e não considerará como anomalia, isso se o usuário atender a orientação. A figura 3.14 mostra a representação gráfico de tempo de parada.

Na tabela 3.22 demonstramos todas as pontuações de Score por anomalia gerada pelo usuário do veículo.

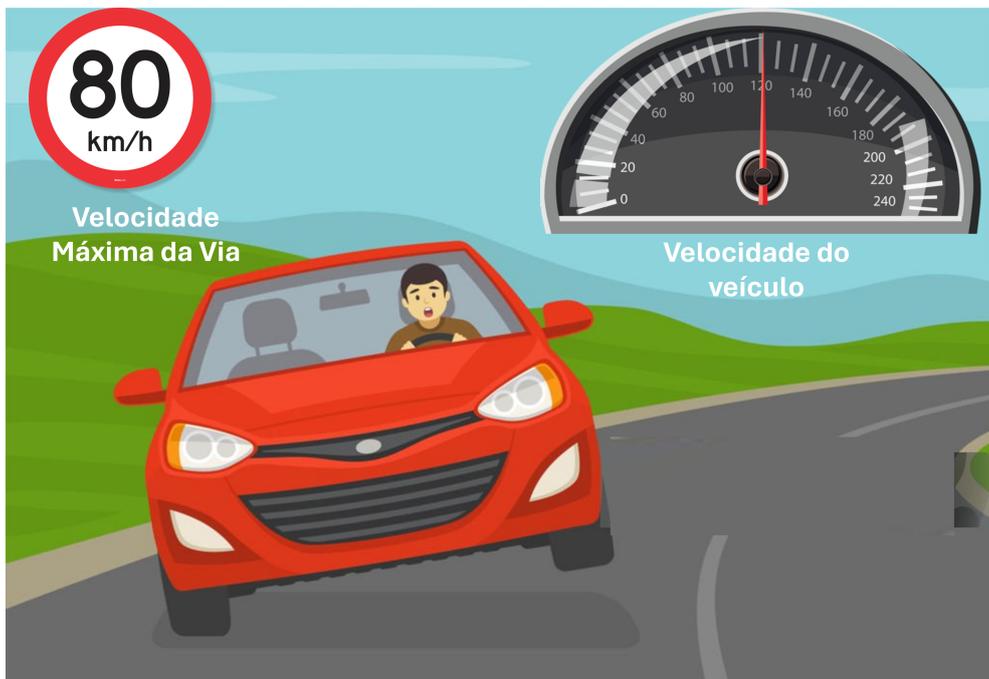


Figura 3.13: Risco de alta velocidade em curvas
 Fonte: autoria própria

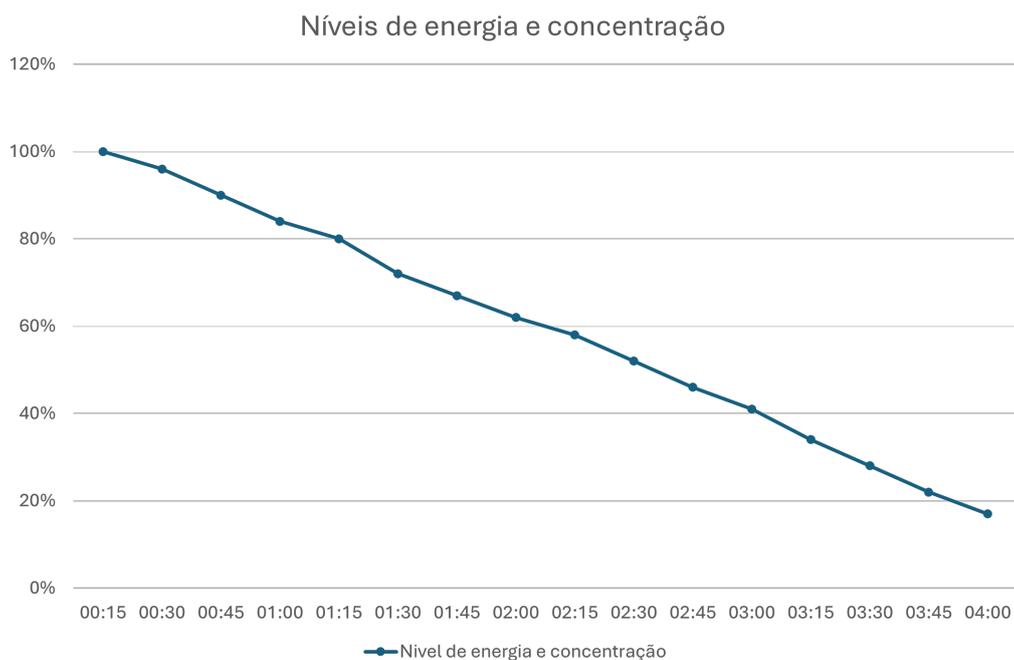


Figura 3.14: Tempo de parada obrigatório
 Fonte: autoria própria

3.4.2 Desenvolver a lógica de negócios e funcionalidades principais

Na etapa de desenvolvimento da lógica de negócios e das funcionalidades principais de um aplicativo, nossa abordagem é centrada na criação de processos essenciais que não apenas atendam, mas superem as expectativas dos usuários. A implementação dessas operações fundamentais, como registro e autenti-

Tabela 3.22: Anomalias de condução

Anomalia de condução	Débito de score
Velocidade acima do permitido da via	100
Frenagens bruscas	30
Aceleração bruscas	30
Inclinações bruscas do veículo	50
Tempo de condução maior que quatro horas sem pausa de quinze minutos para descanso	50

cação de usuários, gerenciamento de dados e processamento de informações, é conduzida com precisão, focalizando na construção de algoritmos e regras de negócios que ditem o fluxo de interações do usuário.

Inspirados pelo paradigma da Clean Architecture (32), buscamos criar uma estrutura modular que facilite a compreensão, manutenção e escalabilidade da lógica de negócios. A modularidade é uma peça-chave para garantir que cada componente funcional do aplicativo seja independente e possa ser aprimorado ou ajustado sem afetar o restante do sistema. Além disso, a reutilização de código promove eficiência, permitindo que blocos de lógica de negócios sejam aplicados em diferentes partes do aplicativo, aumentando a coesão e a consistência.

Durante o desenvolvimento dessa camada, é fundamental não apenas atender às necessidades imediatas dos usuários, mas também projetar uma lógica de negócios flexível e escalável. Isso implica considerar não apenas os requisitos atuais do mercado, mas também antecipar futuras atualizações e possíveis mudanças nas demandas dos usuários. A referência constante a padrões de projeto, como o Model-View-Presenter (43), fornece diretrizes valiosas sobre como estruturar a lógica de negócios para acomodar mudanças sem comprometer a estabilidade.

3.4.3 Implementar a integração com serviços externos, se necessário

Adicionalmente, ao implementar a integração com serviços externos, nossa abordagem é orientada para oferecer uma experiência mais rica e abrangente aos usuários. A integração com APIs de terceiros ou sistemas externos é realizada com cuidado e eficiência, permitindo que o aplicativo acesse recursos adicionais, como autenticação por redes sociais, processamento de pagamentos, ou qualquer outra funcionalidade que aprimore a experiência do usuário.

A segurança dos dados é uma preocupação primordial durante a implementação da integração com serviços externos. Referências como "RESTful Api Design"(42) oferecem insights cruciais sobre as práticas recomendadas para garantir a confiabilidade e segurança da comunicação entre o aplicativo e os sistemas externos. Ao abordar aspectos como a manutenção da segurança dos dados e a conformidade com políticas de privacidade, buscamos assegurar que a experiência do usuário seja contínua, confiável e livre de falhas.

A ênfase na qualidade e na robustez da lógica de negócios e integração com serviços externos é um reflexo do comprometimento em oferecer um aplicativo que não apenas atenda, mas exceda as expectativas dos usuários. As referências de especialistas, como Martin em (32) e Biehl em (39), são guias essenciais ao longo desse processo de implementação, contribuindo para a construção de uma base sólida e eficiente para o aplicativo em desenvolvimento. Essa abordagem estratégica e orientada para qualidade é fundamental para o sucesso a longo prazo do projeto.

3.4.3.1 Firebase Cloud Messaging:

No cenário dinâmico e sempre evoluindo do desenvolvimento de aplicativos contemporâneos, o Firebase Cloud Messaging (FCM) emerge como um pilar essencial, fornecendo uma base sólida para a eficiente comunicação entre servidores e dispositivos móveis. Criada pelo Google, essa plataforma robusta representa não apenas uma solução ágil, mas também escalonável para a entrega de mensagens e notificações, desempenhando um papel crucial na construção de experiências interativas e dinâmicas para os usuários (53).

O Firebase Cloud Messaging, como serviço de mensagens em nuvem, estabelece uma ponte assíncrona entre servidores e dispositivos móveis. Sua funcionalidade principal é oferecer uma infraestrutura eficaz para o envio de mensagens, notificações e atualizações. Essa abordagem permite que os desenvolvedores mantenham os usuários informados em tempo real sobre eventos relevantes no aplicativo, contribuindo para uma interação mais rica e engajadora (53).

Quando analisamos a integração do Firebase Cloud Messaging na Google App Engine, torna-se evidente a relevância estratégica dessa combinação. A capacidade de entregar mensagens instantâneas e notificações de maneira eficiente assume uma importância vital na criação de uma experiência de usuário envolvente. Isso alinha diretamente com os objetivos da Google App Engine, que visa oferecer uma plataforma eficaz para o desenvolvimento e a execução de aplicativos escaláveis (53).

Ao adotar o Firebase Cloud Messaging, os desenvolvedores não estão apenas assegurando uma comunicação eficiente em tempo real, mas também potencializando a capacidade de seus aplicativos em manter os usuários engajados, informados e satisfeitos. Essa estratégia resulta em experiências mais dinâmicas e impactantes, alinhadas com as demandas de um público cada vez mais exigente (53).

3.4.3.2 Notificações em Tempo Real

A entrega imediata de notificações é um dos princípios-chave que tornam o FCM uma escolha valiosa para os desenvolvedores. Essa capacidade não apenas informa os usuários sobre eventos relevantes em tempo real, mas também desempenha um papel crucial na manutenção da retenção de usuários. Em um cenário onde a atenção do usuário é um recurso valioso, a capacidade de fornecer informações instantâneas cria uma experiência envolvente, mantendo os usuários atualizados e conectados com o aplicativo de maneira consistente (53).

A notificação em tempo real não é apenas uma funcionalidade técnica; é uma estratégia para estabelecer uma conexão mais profunda com os usuários. Ao receber alertas instantâneos sobre mensagens importantes, atualizações ou eventos personalizados, os usuários sentem-se mais engajados e percebem um valor adicional no aplicativo. Isso não apenas contribui para a retenção de usuários, mas também fortalece a fidelidade, uma vez que os usuários reconhecem o aplicativo como uma fonte confiável de informações relevantes e oportunas (53).

3.4.3.3 Segmentação de Público

A capacidade de segmentar usuários com base em critérios específicos é outra característica fundamental do FCM. Ao permitir que os desenvolvedores personalizem mensagens com base em dados como localização geográfica ou padrões de uso, o FCM abre oportunidades para uma comunicação mais relevante e impactante (53).

A personalização é a chave para criar experiências significativas para os usuários. Ao compreender o comportamento, preferências e localização dos usuários, os desenvolvedores podem adaptar suas mensagens de maneira precisa. Isso não apenas aumenta a relevância das notificações, mas também melhora a eficácia das estratégias de engajamento do usuário. Por exemplo, um usuário em uma região específica pode receber notificações relacionadas a ofertas locais, enquanto um usuário com padrões de uso específicos pode ser direcionado a atualizações personalizadas do aplicativo. Essa abordagem sob medida contribui para uma experiência do usuário mais envolvente e, ao mesmo tempo, otimiza os esforços de marketing e comunicação (53).

Ao integrar esses princípios-chave do FCM, os desenvolvedores não apenas implementam funcionalidades técnicas avançadas, mas moldam estratégias eficazes para envolver e reter usuários. Em um mundo onde a personalização e a instantaneidade são valorizadas pelos usuários, o Firebase Cloud Messaging emerge como uma ferramenta essencial para elevar a comunicação em aplicativos móveis a um novo patamar, proporcionando experiências que realmente ressoam com o público-alvo (53).

3.4.3.4 Integração com Outros Serviços Firebase

A integração do Firebase Cloud Messaging (FCM) com outros serviços Firebase é um diferencial significativo que promove uma sinergia poderosa dentro do ecossistema de desenvolvimento. Ao conectar o FCM ao Realtime Database e outros serviços, os desenvolvedores não apenas ampliam as capacidades do aplicativo, mas também criam uma base sólida para a construção de experiências mais ricas e integradas (53).

A integração com o Realtime Database é particularmente valiosa. Esse banco de dados em tempo real permite que os desenvolvedores armazenem e sincronizem dados entre os usuários em tempo real. Ao unir essa funcionalidade com as capacidades de notificação em tempo real do FCM, os aplicativos podem proporcionar experiências colaborativas e interativas excepcionais. Por exemplo, em um aplicativo de mensagens, as atualizações de mensagens entre os usuários podem ser instantaneamente refletidas, proporcionando uma sensação de conversa em tempo real. Isso não apenas melhora a experiência do usuário, mas também demonstra a eficácia da integração entre diferentes serviços Firebase (53).

Essa abordagem integrada não apenas simplifica o desenvolvimento, mas também aumenta a eficiência operacional. Os desenvolvedores podem aproveitar as funcionalidades complementares de diferentes serviços sem a complexidade de integrações externas. A coesão entre o FCM e outros serviços Firebase resulta em uma infraestrutura mais robusta e confiável, proporcionando uma base sólida para a criação de aplicativos inovadores e altamente funcionais.

3.4.3.5 Compatibilidade Multiplataforma

A compatibilidade multiplataforma é um pilar essencial do Firebase Cloud Messaging, proporcionando uma experiência consistente e fluida para usuários de dispositivos iOS, Android e web (53). Essa capacidade simplifica consideravelmente o desafiador processo de desenvolvimento multiplataforma, permitindo que os desenvolvedores alcancem um público mais amplo sem comprometer a qualidade da experiência do usuário.

Ao oferecer suporte abrangente para diferentes sistemas operacionais, o FCM elimina barreiras no acesso aos aplicativos. Os usuários podem receber notificações e mensagens em tempo real, independentemente do dispositivo que estão utilizando, garantindo uma experiência homogênea em diferentes plataformas. Isso não apenas simplifica o gerenciamento e manutenção do aplicativo, mas também amplia o alcance, permitindo que os desenvolvedores atinjam públicos diversos de maneira eficaz (53).

Essa compatibilidade multiplataforma não se limita apenas à entrega de notificações; ela se estende à experiência do usuário como um todo. A capacidade de criar aplicativos que oferecem uma experiência consistente, independentemente do dispositivo, contribui para a construção de uma marca sólida e confiável. Os usuários aprendem a esperar uma experiência de alta qualidade, independentemente do dispositivo que estão utilizando, gerando confiança e fidelidade à marca (53).

3.4.3.6 Cloud Endpoints: Facilitando o Desenvolvimento de APIs Robustas

A construção de aplicativos escaláveis e eficientes demanda a implementação de APIs robustas, fundamentais para a comunicação eficaz entre serviços. Uma ferramenta que se destaca nesse contexto é o Cloud Endpoints, uma poderosa solução oferecida pela Google App Engine. Essa ferramenta desempenha um papel crucial, simplificando e agilizando o processo de criação, implantação e gerenciamento de APIs, estabelecendo uma base sólida para a interconexão de serviços (54).

O Cloud Endpoints apresenta características distintivas que o tornam uma escolha valiosa no desenvolvimento de APIs. Uma dessas características notáveis é a capacidade de gerar automaticamente Software Development Kits (SDKs) para diferentes plataformas, incluindo Android, iOS e JavaScript. Essa automação não apenas alivia a carga de trabalho dos desenvolvedores, mas também garante consistência na implementação da API em diferentes ambientes, proporcionando uma experiência unificada para os usuários finais (54). Outro ponto-chave é a autenticação integrada oferecida pelo Cloud Endpoints. Com suporte a métodos flexíveis de autenticação, desde contas de serviço até a integração com o Firebase Authentication, a ferramenta assegura a segurança da API. Esse controle rigoroso de acesso contribui para a proteção dos dados, permitindo que apenas usuários autorizados interajam com os serviços disponibilizados pela API (54). Além disso, o Cloud Endpoints inclui recursos avançados de monitoramento e análise, proporcionando aos desenvolvedores insights valiosos sobre o desempenho da API (55). Essa capacidade de avaliar o desempenho, identificar possíveis gargalos e otimizar a entrega de serviços é essencial para garantir uma experiência do usuário fluida e eficiente. O monitoramento contínuo possibilita ajustes e aprimoramentos necessários para manter a API em conformidade com as expectativas de desempenho (54).

Ao integrar o Cloud Endpoints à Google App Engine, fortalecemos significativamente a infraestrut-

tura do aplicativo. A facilidade na criação de APIs, aliada à geração automática de SDKs e aos recursos avançados de segurança, cria um ambiente propício para o desenvolvimento de aplicativos escaláveis e interoperáveis. Essa integração estratégica potencializa a capacidade da Google App Engine em fornecer uma plataforma eficaz para o desenvolvimento e execução de aplicativos escaláveis e eficientes (54).

3.4.4 Configurar o servidor e banco de dados

Configurar o servidor e o banco de dados para um aplicativo é um passo que envolve a escolha adequada de tecnologias, provisionamento de recursos e garantia de segurança e desempenho. Primeiramente, é fundamental selecionar um servidor que atenda às necessidades do aplicativo, considerando requisitos como capacidade de processamento, largura de banda, escalabilidade e localização geográfica dos usuários. Isso pode envolver a configuração de servidores dedicados, serviços de nuvem como AWS, Azure ou Google Cloud, ou até mesmo soluções de hospedagem específicas para aplicativos web ou móveis.

Além disso, a escolha e configuração do banco de dados são cruciais para garantir a eficiência na gestão e acesso aos dados do aplicativo. O tipo de banco de dados - relacional, NoSQL, NewSQL - deve ser selecionado com base nas necessidades específicas de armazenamento, consulta e escalabilidade do aplicativo. A configuração do banco de dados deve considerar aspectos como otimização de consultas, índices para melhorar o desempenho, políticas de backup e recuperação, bem como estratégias de segurança para proteger os dados sensíveis.

Projetar uma estrutura de banco de dados e APIs

No processo de projetar a estrutura do banco de dados e as APIs para nosso aplicativo, é imperativo realizar uma análise detalhada dos dados que serão manipulados pela aplicação. Inspirado pelo conceito de Clean Architecture (32), buscamos criar uma estrutura modular e bem definida que promova a reutilização de código, manutenibilidade e escalabilidade.

Começamos identificando as entidades e os relacionamentos entre elas. Referências como "Banco de Dados"(41) auxiliam na escolha adequada entre modelos de bancos de dados, considerando se o sistema se beneficiaria mais de um modelo relacional, NoSQL ou NewSQL, com base nas necessidades específicas do aplicativo.

Para garantir a integridade e consistência dos dados, definimos tipos de dados apropriados e aplicamos as restrições necessárias. A aplicação de métricas de usabilidade em protótipos computadorizados de baixa fidelidade (45) nos orienta na validação precoce, garantindo que a estrutura do banco de dados atenda às expectativas dos usuários.

Ao projetar a API, realizamos um planejamento cuidadoso dos endpoints, métodos de requisição e resposta, bem como os mecanismos de autenticação e autorização. Referências como "RESTful Api Design"(42) proporcionam insights valiosos sobre boas práticas na definição de APIs eficientes e seguras.

Consideramos a eficiência da consulta, a segurança dos dados e a flexibilidade para futuras expansões ou atualizações do aplicativo. Optamos por seguir padrões como REST ou GraphQL, escolhendo o protocolo mais adequado para a comunicação entre os diferentes componentes do aplicativo e a API.

Além disso, adotamos uma abordagem de documentação abrangente, referenciando "APIs: A strategy guide"(36) e "Análise de dados multivariados"(29), para fornecer informações detalhadas sobre os endpoints, parâmetros aceitos, respostas esperadas e fluxos de autenticação. Essa documentação serve como um guia essencial para a colaboração eficaz entre equipes de desenvolvimento e stakeholders, garantindo uma compreensão clara do sistema para futuras iterações e expansões do aplicativo.

3.4.4.1 Apigee API Platform: Orquestrando o Ecossistema de APIs

A orquestração efetiva de APIs é crucial para a integração harmoniosa de serviços e sistemas. A Apigee API Platform, integrada à Google App Engine, emerge como uma ferramenta essencial para criar, gerenciar e otimizar APIs, proporcionando uma camada de controle indispensável para um ecossistema coeso (56).

A Apigee API Platform simplifica o desenvolvimento de APIs, oferecendo ferramentas intuitivas para criação e design eficiente. Com suporte a padrões como REST e GraphQL, amplia as possibilidades de integração. Seus recursos avançados de gerenciamento de tráfego permitem a distribuição controlada de requisições, otimizando o desempenho. Mecanismos robustos de segurança, incluindo autenticação e autorização, fortalecem a proteção da API. A capacidade de analisar o desempenho da API em tempo real, juntamente com ferramentas de monitoramento abrangentes, oferece insights valiosos para otimização contínua e resolução proativa de problemas (56).

Ao integrar a Apigee API Platform à Google App Engine, estabelecemos uma camada de controle essencial para o ecossistema de APIs. O desenvolvimento facilitado, o gerenciamento eficiente de tráfego e segurança robusta contribuem para a construção de APIs confiáveis e interoperáveis (56).

Por fim, a configuração do servidor e do banco de dados deve contemplar a integração adequada entre os dois sistemas. Isso inclui configurar conexões seguras entre o servidor e o banco de dados, ajustar parâmetros de comunicação para otimizar o desempenho e garantir que a infraestrutura esteja dimensionada para lidar com o volume de tráfego e operações do aplicativo. A sincronização eficiente entre o servidor e o banco de dados é essencial para garantir a disponibilidade e consistência dos dados, assegurando uma experiência estável e confiável para os usuários do aplicativo.

Em resumo, ao projetar a estrutura do banco de dados e APIs, integramos as melhores práticas de design, seguindo padrões consolidados, e nos beneficiamos de referências relevantes para criar uma base sólida que promove a eficiência, segurança e adaptabilidade contínua do nosso aplicativo (30, 43).

3.4.4.2 Cloud SQL: Elevando o Gerenciamento de Bancos de Dados na Nuvem

A gestão eficiente de bancos de dados é uma peça fundamental para garantir o desempenho e a escalabilidade de aplicativos modernos. Nesse contexto, o Google Cloud SQL, quando integrado à Google App Engine, surge como uma solução confiável e escalável para o armazenamento e recuperação de dados, enriquecendo o ecossistema de aplicativos (57).

O Google Cloud SQL se destaca em diversos pilares que contribuem para a eficiência do gerenciamento de bancos de dados na nuvem. Sua capacidade de escalabilidade automática é notável, ajustando os recursos de forma dinâmica conforme a demanda. Isso garante um desempenho consistente, mesmo

diante de variações no volume de dados e acessos simultâneos. Além disso, sua abordagem simplificada para configuração de políticas de backup automático e a capacidade de recuperação pontual proporcionam robustez ao garantir a integridade dos dados. Recursos avançados, como a criptografia de dados em repouso e em trânsito, aliados à gestão sofisticada de acessos, asseguram a confidencialidade e integridade dos dados armazenados (58).

Integrar o Google Cloud SQL à Google App Engine cria um ambiente coeso para armazenamento e recuperação de dados. A escalabilidade automática, a simplicidade nas operações de backup e recuperação, além dos padrões de segurança avançados, aprimoram a gestão de bancos de dados na nuvem, proporcionando um ambiente propício para o desenvolvimento de aplicativos escaláveis e interoperáveis (57).

3.4.4.3 Google Cloud Functions: Potencializando Operações Sem Servidor

O paradigma sem servidor, exemplificado pelas Google Cloud Functions, redefine a abordagem ao desenvolvimento de aplicativos. Quando integradas à Google App Engine, essas funções oferecem uma maneira ágil e eficiente de executar código em resposta a eventos específicos (53).

As Google Cloud Functions apresentam vantagens distintas que as tornam uma escolha eficaz. Sua capacidade de escalar automaticamente em resposta ao volume de eventos proporciona eficiência operacional e economia de custos. A integração com eventos específicos, como alterações em bancos de dados ou uploads de arquivos, promove uma arquitetura reativa, onde o código é executado somente quando necessário. A diversidade de linguagens suportadas, incluindo Node.js, Python e Go, oferece flexibilidade aos desenvolvedores, permitindo que escolham a linguagem mais adequada para suas necessidades (53).

Ao introduzir estrategicamente as Google Cloud Functions na Google App Engine, adicionamos agilidade ao desenvolvimento. Utilizar funções sem servidor para lidar com eventos específicos reduz a complexidade operacional, promovendo uma arquitetura mais eficiente e centrada em eventos (53).

3.5 POLIMENTO E REFINAMENTO

Polir e refinar o aplicativo após o teste do cliente é fundamental para melhorar a qualidade, a usabilidade e a experiência geral do usuário. Ao obter feedback dos clientes, torna-se possível identificar áreas específicas que necessitam de ajustes ou aprimoramentos. Esse procedimento abrange a correção de pequenas falhas, o aprimoramento da interface para maior lucidez e simplicidade e a otimização da funcionalidade com base no feedback e na análise do comportamento do usuário. O aprimoramento contínuo do aplicativo após o teste do cliente permite que o produto definitivo seja aprimorado para se alinhar melhor às expectativas do usuário, resultando em um aplicativo mais robusto, fácil de usar e adaptado aos requisitos do público-alvo.

3.5.1 Refinar o código e a interface com base nos testes de usabilidade

Refinar o código e a interface com base nos resultados dos testes de usabilidade para melhorar a qualidade e a experiência do usuário de um aplicativo. Após um exame minucioso dos resultados do teste, os desenvolvedores conseguem identificar áreas específicas que necessitam de modificações no código-fonte para otimizar o desempenho, corrigir possíveis erros e melhorar a eficiência do aplicativo. Além disso, os testes de usabilidade frequentemente revelam deficiências na interface que podem ser corrigidas para garantir uma experiência mais intuitiva e agradável para os usuários. Essas modificações abrangem alterações na disposição dos elementos, aprimoramentos na navegação e ajustes visuais destinados a aumentar a clareza e a usabilidade geral do aplicativo, contribuindo assim para uma experiência mais gratificante, compatível com as expectativas do usuário.

3.5.2 Otimizar o desempenho e a segurança do aplicativo

Otimizar o desempenho e a segurança dos aplicativos é um processo contínuo e indispensável que é de extrema importância para garantir uma experiência confiável e protegida para os usuários. Para melhorar o desempenho, os desenvolvedores podem empregar várias estratégias para identificar e corrigir gargalos no código, otimizar as consultas ao banco de dados, incorporar técnicas de cache e minimizar o consumo de recursos, garantindo assim uma execução mais rápida e eficiente do aplicativo. Com relação à segurança, é de suma importância adotar práticas robustas, como criptografia de dados, autenticação segura, gerenciamento adequado de sessões e atualizações regulares para resolver quaisquer vulnerabilidades conhecidas. O aprimoramento contínuo do desempenho e da segurança do aplicativo não serve apenas para aprimorar a experiência do usuário, mas também para estabelecer e manter a confiança e a proteção dos dados críticos do usuário. Ao buscar constantemente melhorias nessas áreas, os desenvolvedores podem criar um ambiente que não apenas atenda às necessidades e expectativas dos usuários, mas também forneça uma base sólida para a longevidade e o sucesso do aplicativo. Em conclusão, a busca pelo desempenho e pela segurança ideais dos aplicativos exige uma abordagem multifacetada que englobe várias técnicas e práticas destinadas a fornecer uma experiência de usuário perfeita e protegida, preservando a integridade e a confidencialidade dos dados confidenciais do usuário.

3.6 IMPLANTAÇÃO E LANÇAMENTO

A fase de implantação e lançamento de um aplicativo é um momento crucial que demanda uma estratégia cuidadosamente elaborada para assegurar uma transição fluida do desenvolvimento para a acessibilidade pública. Essa etapa abrange desde a configuração de ambientes de produção até a execução de exames conclusivos para garantir a ausência de falhas críticas. Este processo exige uma preparação metódica, incluindo estratégias de marketing e comunicação, além da criação de materiais de apoio para os usuários. Vamos explorar, de maneira prática, os passos essenciais dessa fase, incorporando as melhores práticas de profissionais renomados da área.

Antes de disponibilizar o aplicativo para o público, é crucial configurar ambientes de produção rob-

tos. Conforme sugerido por Biehl em (42), essa etapa envolve a implementação de servidores e bancos de dados capazes de acomodar o aplicativo de forma estável e expansível. A criação desses ambientes não apenas garante a eficiência do aplicativo, mas também contribui para a sua escalabilidade, possibilitando a adaptação a um aumento significativo de usuários.

A execução de testes conclusivos é uma prática padrão para identificar e corrigir defeitos antes do lançamento. Inspirados pela abordagem de Demico et al. em (47), os desenvolvedores conduzem exames conclusivos para garantir que o aplicativo funcione conforme o esperado, sem comprometer a experiência do usuário. Somente após a conclusão bem-sucedida desses testes, o aplicativo está pronto para avançar para a próxima fase.

A fase de iniciação não se resume apenas à disponibilização técnica do aplicativo. Estratégias de marketing e comunicação são cruciais para promover o aplicativo e atrair usuários. Inspirados por insights de Gomes em (27), elaboramos estratégias eficazes de marketing que envolvem a definição clara de mensagens-chave, identificação do público-alvo e seleção de canais de comunicação apropriados.

A montagem de materiais de apoio, como documentação e perguntas frequentes, segue as diretrizes de Lidwell et al. em (35) para garantir que os usuários tenham acesso a informações úteis e esclarecedoras. Esses materiais desempenham um papel fundamental na orientação dos usuários durante a iniciação, proporcionando uma experiência inicial positiva.

3.6.1 Preparar a versão final para publicação na Google Play Store

A publicação na Google Play Store é um marco significativo para qualquer aplicativo Android. A preparação da versão final é um processo que requer atenção aos detalhes e conformidade com os padrões da plataforma. À luz das sugestões de García em (30), os desenvolvedores realizam testes adicionais para garantir que a versão final esteja livre de defeitos críticos. A configuração da página do aplicativo na loja, incluindo informações precisas e atrativas, é essencial para atrair usuários.

A configuração de descrições concisas e claras, acompanhadas por imagens cativantes, segue as práticas recomendadas por Lowdermilk em (39). Esses elementos visuais e descritivos são fundamentais para capturar a atenção dos usuários e comunicar efetivamente os benefícios do aplicativo. Além disso, a inclusão de metadados relevantes, como palavras-chave e informações de categorização, otimiza a visibilidade do aplicativo na loja, facilitando a sua descoberta pelos usuários.

O lançamento do aplicativo é um passo estratégico que vai além da publicação na loja. A utilização de plataformas de mídia social, conforme sugerido por Marques et al. em (59), oferece uma oportunidade única de estabelecer conexões diretas com usuários em potencial. A disseminação de informações relevantes, o envolvimento do usuário por meio de conteúdo interativo e o recebimento rápido de feedback contribuem para uma estratégia de lançamento eficaz.

Além disso, alianças estratégicas com empresas afiliadas, seguindo a visão de Jacobson et al. em (36), proporcionam uma maneira poderosa de ampliar o alcance do aplicativo. Essas parcerias não apenas aumentam a visibilidade do aplicativo, mas também capitalizam sobre a base de clientes ou seguidores dessas empresas, gerando credibilidade e confiança.

A implantação e lançamento bem-sucedidos de um aplicativo não são apenas eventos isolados, mas o início de uma jornada contínua. Inspirados pelos princípios de Martin em (32) sobre arquitetura limpa, buscamos não apenas atender às expectativas iniciais, mas criar uma base para melhorias contínuas.

Neste processo, reconhecemos a importância de uma abordagem holística que abrange desde a configuração técnica até estratégias de marketing e parcerias estratégicas. Ao adotar essa perspectiva abrangente, esperamos não apenas lançar um aplicativo de sucesso, mas cultivar uma presença sólida e duradoura no mercado, com uma base de usuários engajada e fiel (60).

3.6.2 Configurar descrição, imagens e metadados na loja

Ao nos aproximarmos da fase crucial de lançamento do aplicativo, a configuração da apresentação na Google Play Store emerge como um fator decisivo para atrair usuários em potencial. Esta etapa inicial não apenas demanda habilidades técnicas, mas também exige uma compreensão aprofundada das expectativas e comportamentos dos usuários.

A descrição do aplicativo desempenha um papel crucial na tomada de decisão dos usuários. Inspirados pelos princípios de Lidwell et al. em (35) sobre princípios universais do design, buscamos criar uma descrição que seja tanto concisa quanto impactante. Destacamos eficazmente as principais características e benefícios do aplicativo, transmitindo informações operacionais de forma clara. Além disso, nossa abordagem leva em consideração a compreensão da psicologia do usuário, reconhecendo a importância de despertar interesse desde os primeiros segundos.

Ao implementar essa estratégia, procuramos inspiração na obra de Carvalho em (61) sobre lógicas visuais para garantir que a descrição seja não apenas informativa, mas também visualmente atraente. A combinação de elementos visuais e textuais visa criar uma narrativa envolvente que capte a atenção dos usuários e os motive a explorar mais sobre o aplicativo.

A seleção de imagens para a apresentação na loja é uma arte que vai além de simplesmente mostrar a interface do aplicativo. Inspirados pelas práticas sugeridas por Lowdermilk em (39), buscamos criar imagens visualmente cativantes, destacando não apenas a estética, mas também a funcionalidade do aplicativo. Capturas de tela estrategicamente escolhidas são apresentadas de maneira a transmitir os pontos fortes e a experiência que os usuários podem esperar.

Neste processo, consideramos as diretrizes de Machado et al. em (44) sobre métodos de avaliação de usabilidade. Ao escolher as imagens, priorizamos aquelas que não apenas ilustram as funcionalidades, mas também proporcionam uma visão clara da usabilidade do aplicativo. Este enfoque é fundamental para estabelecer expectativas realistas e construir a confiança dos usuários desde o início.

A configuração de metadados, incluindo palavras-chave e informações de categorização, é um aspecto muitas vezes subestimado, mas crucial, para a visibilidade do aplicativo na loja. Inspirados pelas sugestões de Zabaleta em (62) sobre matriz de priorização, abordamos a seleção de palavras-chave de maneira estratégica. Consideramos não apenas a popularidade das palavras-chave, mas também sua relevância para o público-alvo.

Além disso, a categorização adequada, seguindo as diretrizes da Google Play Store, é vital para garantir

que o aplicativo seja apresentado nas buscas relevantes. Nossa abordagem incorpora insights de Pinheiro em (28) sobre pesquisa de mercado, reconhecendo a importância de alinhar a apresentação na loja com as expectativas e padrões do mercado atual.

A apresentação meticulosa na Google Play Store é um investimento estratégico que pode resultar em dividendos significativos. Acreditamos na importância de uma abordagem abrangente, levando em consideração cada detalhe, desde a escolha das palavras até a seleção das imagens. Esta mentalidade é consistente com os princípios de Martin em (32) sobre arquitetura limpa, onde cada decisão é tomada com uma visão a longo prazo, considerando não apenas o lançamento inicial, mas também melhorias futuras.

3.6.3 Lançar o aplicativo

O lançamento do aplicativo em plataformas de mídia social e por meio de colaborações com empresas afiliadas constitui uma tática potente para aumentar a proeminência e o alcance do aplicativo. A utilização de redes de mídia social oferece a oportunidade de estabelecer um relacionamento direto com usuários em potencial, disseminar informações pertinentes sobre o aplicativo, cultivar o envolvimento do usuário por meio de conteúdo interativo e receber feedback rápido. Alianças estratégicas com empresas aliadas oferecem a oportunidade de capitalizar sobre seus clientes ou seguidores, divulgando assim o aplicativo para um público que já investiu e confia nessas marcas, gerando maior credibilidade e adesão ao aplicativo. Esses esforços inaugurais servem para cultivar uma base inicial de usuários e estabelecer uma presença robusta no mercado logo no início da introdução do aplicativo (60).

3.7 MONITORAMENTO PÓS-LANÇAMENTO

O monitoramento pós-lançamento de um aplicativo é essencial para compreender como ele está sendo recebido pelos usuários e identificar áreas de aprimoramento contínuo. Através do acompanhamento de métricas-chave, como taxas de engajamento, tempo de uso, taxas de retenção e feedback dos usuários, os desenvolvedores podem avaliar o desempenho do aplicativo em tempo real (42). Esses insights são fundamentais para detectar eventuais problemas, identificar tendências de uso e entender as preferências dos usuários, possibilitando a realização de ajustes e atualizações que melhorem a experiência do usuário, corrijam problemas de desempenho e agreguem novos recursos de acordo com as necessidades identificadas. O monitoramento contínuo após o lançamento é uma prática fundamental para manter a relevância do aplicativo e garantir que ele permaneça alinhado com as expectativas do mercado e dos usuários.

3.7.1 Monitorar métricas de desempenho

As métricas para monitorar o desempenho de um aplicativo são diversas e podem variar de acordo com os objetivos específicos do app, mas algumas métricas comuns incluem taxas de retenção de usuários, tempo médio de sessão, taxa de conversão, número de downloads e avaliações dos usuários (42). No entanto, é importante reconhecer os limites de cada métrica isoladamente. Por exemplo, o número de downloads pode indicar o alcance inicial do aplicativo, mas não necessariamente sua utilidade ou sucesso

a longo prazo. Da mesma forma, as avaliações dos usuários fornecem feedback valioso, mas podem refletir opiniões individuais que nem sempre representam a visão da maioria. Assim, é importante interpretar as métricas em conjunto, considerando seu contexto e relacionamento entre si, para obter uma compreensão holística do desempenho do aplicativo.

Além disso, algumas métricas podem ser enganosas se não forem analisadas com cuidado. Por exemplo, uma alta taxa de retenção de usuários pode parecer positiva, mas se os usuários estiverem retornando apenas para resolver problemas ou reclamar sobre o aplicativo, isso pode indicar uma insatisfação geral. Portanto, é importante ir além das métricas superficiais e buscar entender o comportamento dos usuários, considerando as nuances por trás dos números. Isso exige uma abordagem analítica e uma combinação inteligente de diferentes métricas para obter uma visão mais completa e precisa do desempenho e do impacto do aplicativo.

3.7.1.1 downloads

Ao entrar na fase de análise de desempenho do aplicativo, a métrica de downloads se destaca como um indicador-chave do alcance e interesse dos usuários. No entanto, a interpretação dessa métrica exige uma abordagem ponderada, reconhecendo que, embora os downloads sejam vitais, eles não contam toda a história do sucesso a longo prazo. Em nossa implementação, buscamos ir além dos números iniciais e considerar uma variedade de fatores que oferecem uma compreensão mais holística do desempenho do aplicativo.

Ao implementar estratégias para impulsionar os downloads iniciais, nos inspiramos no Modelo-View-Apresentador (MVP) de García em (30). Isso envolveu a formulação de uma estratégia de marketing sólida, aproveitando as redes sociais e parcerias estratégicas. No entanto, reconhecemos que o sucesso inicial não garante sucesso contínuo. Portanto, este é apenas o primeiro passo na jornada para avaliar o verdadeiro impacto do aplicativo.

A métrica de downloads é um ponto de partida, mas para avaliar o valor duradouro do aplicativo, incorporamos as taxas de retenção. Freitas (34) destaca a importância de manter os usuários após o download inicial. Implementamos estratégias de engajamento contínuo, oferecendo atualizações regulares e recursos atraentes para manter os usuários envolvidos. Isso é fundamental para assegurar que os downloads iniciais não se transformem em desinstalações prematuras.

No cerne da avaliação do desempenho está o engajamento do usuário e as avaliações. Aqui, seguimos as melhores práticas de Gaudio (31) em Model-View-ViewModel (MVVM), garantindo que a experiência do usuário seja prioritária. Incorporamos feedbacks de usuários por meio de funcionalidades de comentários e análises, reconhecendo que a percepção do usuário é um indicador valioso da utilidade do aplicativo.

Biehl (42) destaca a importância de uma abordagem equilibrada para métricas. Embora um alto número de downloads inicial possa indicar uma estratégia de marketing eficaz, não consideramos isso como único indicador de sucesso. Adotamos uma abordagem mais holística, considerando o crescimento sustentável e a satisfação contínua do usuário como medidas mais significativas de sucesso.

Ao considerar a métrica de downloads, olhamos para trás, refletindo sobre a aplicação de métricas de usabilidade em nossos estágios iniciais de desenvolvimento, conforme sugerido por das Virgens & Silveira em (45) (Usabilidade em Protótipos Computadorizados de Baixa Fidelidade). A usabilidade é um fator crítico para manter os usuários engajados após o download inicial, contribuindo para taxas de retenção saudáveis.

3.7.1.2 avaliações

No universo dos aplicativos, a métrica de avaliações se destaca como uma bússola direcional para a satisfação e experiência dos usuários. Essas análises não apenas refletem a opinião dos usuários sobre a qualidade, usabilidade e valor do aplicativo, mas também fornecem valiosos insights para os desenvolvedores, alimentando a busca constante por melhorias. Na implementação do projeto, estamos comprometidos em adotar uma abordagem abrangente na análise dessas avaliações, considerando tanto os elogios quanto as críticas construtivas.

Inspirados no Modelo-View-Apresentador (MVP) de García (30), procuramos integrar estratégias que vão além do desenvolvimento técnico. A ênfase está na experiência do usuário, refletida nas avaliações. Estratégias de MVP não apenas visam downloads iniciais, mas também engajamento contínuo. Isso se alinha à visão de Lattin et al. (29) sobre análise de dados multivariados, onde a satisfação do usuário é um ponto crucial para o sucesso a longo prazo.

Demico et al. em (47) destacam a importância do feedback na evolução de aplicativos. Ao analisar as avaliações, não apenas observamos as pontuações, mas também buscamos padrões e tendências nos comentários dos usuários. A resposta ativa a avaliações, tanto positivas quanto negativas, é uma parte vital do nosso compromisso com a melhoria contínua.

A implementação do Teste de Usabilidade (33) está intrinsecamente ligada à avaliação do usuário. Antecipamos potenciais áreas de atrito ou confusão através de testes e prototipagem, contribuindo para aprimorar a usabilidade antes mesmo das avaliações surgirem. Isso reflete nosso compromisso em entender as necessidades do usuário desde as fases iniciais de desenvolvimento.

Como sugerido por Biehl ((42) em "RESTful API Design," adotamos uma abordagem equilibrada às avaliações. Reconhecemos tanto as críticas quanto os elogios como oportunidades de crescimento. As avaliações negativas são vistas como direcionadoras para melhorias, enquanto as positivas validam as escolhas acertadas.

Ao longo do ciclo de vida do aplicativo, estabelecemos práticas de monitoramento contínuo. Isso se alinha à filosofia de iteratividade de desenvolvimento ágil (36). Avaliações, juntamente com métricas de desempenho, orientam ajustes iterativos, assegurando que o aplicativo evolua com as expectativas e necessidades em constante mudança dos usuários.

A análise abrangente das avaliações do usuário não é apenas uma métrica, mas uma filosofia. Integrando princípios de MVP, feedback ativo, usabilidade, e uma abordagem equilibrada, nossa implementação visa não apenas conquistar downloads, mas nutrir uma base de usuários satisfeita e comprometida. Este compromisso não apenas constrói a confiança dos usuários, mas também fortalece a fidelidade, con-

tribuindo para um ciclo de sucesso sustentável.

3.7.1.3 retenção de usuários

A métrica de retenção de usuários é uma bússola crítica que aponta diretamente para a capacidade de um aplicativo de manter seu público ao longo do tempo. Não é apenas sobre atrair downloads iniciais, mas sobre manter o interesse e engajamento após a primeira interação. Inspirados por práticas fundamentadas, como o MVP (Model-View-Presenter) de García (30), nossa abordagem se estende além dos números para aprofundar a compreensão da fidelidade dos usuários.

Ao monitorar a retenção ao longo do tempo, buscamos insights valiosos, conforme destacado por das Virgens e Silveira em (45). Esta análise contínua nos permite avaliar a eficácia de nossas estratégias de retenção, identificar áreas de melhoria e implementar aprimoramentos que garantam uma experiência duradoura e envolvente.

3.7.2 Coletar feedbacks e bugs reportados pelos usuários

Em nossa jornada para proporcionar uma experiência excepcional aos usuários, compreendemos que a inovação genuína emerge de um diálogo contínuo e significativo. A coleta ativa de feedbacks, inspirada na abordagem de Demico et al. (47), representa a essência de nosso compromisso em envolver os usuários no processo de desenvolvimento, transformando suas percepções em melhorias tangíveis.

Nosso compromisso com a coleta ativa de feedbacks é fundamentado na compreensão de que os usuários são partes fundamentais no aprimoramento contínuo de um aplicativo. Estabelecemos canais de comunicação direta para incentivar os usuários a compartilharem suas necessidades, expectativas e experiências. Através desse diálogo, obtemos insights valiosos que vão além dos dados brutos, alcançando a essência das demandas e desejos dos usuários.

O reconhecimento proativo e a rápida resolução de problemas técnicos são essenciais para preservar a integridade da experiência do usuário. Inspirados pelo trabalho de Wen et al. (40), implementamos um sistema eficiente de priorização de relatórios de bugs. Cada relatório é tratado com seriedade, com uma abordagem que visa identificar e resolver problemas técnicos de maneira ágil. Isso não apenas corrige questões imediatas, mas também demonstra nosso compromisso com a qualidade e a satisfação do usuário.

A coleta contínua de feedbacks não se limita apenas à resolução de problemas imediatos; é um componente vital em nosso ciclo de melhoria contínua. Analisamos os feedbacks coletados de maneira holística, identificando tendências e padrões que orientam a evolução estratégica do aplicativo. Dessa forma, não apenas corrigimos questões existentes, mas também antecipamos as necessidades futuras dos usuários.

3.7.3 Planejar atualizações e melhorias contínuas

A métrica de planejamento de atualizações e melhorias contínuas é a espinha dorsal do ciclo de vida de um aplicativo. Inspirados por conceitos de arquitetura limpa (32), buscamos não apenas resolver problemas identificados, mas também antecipar as necessidades dos usuários. Ao planejar atualizações com base no

feedback dos usuários e nas tendências do mercado, alinhamo-nos à visão de Gomes (27) sobre elaboração de pesquisas de mercado para manter a relevância e competitividade do aplicativo.

Nossa abordagem incorpora não apenas as últimas tendências do mercado, mas também as melhores práticas fundamentadas em referências como APIs: A Strategy Guide (36) e Design Centrado no Usuário (39). Isso garante que nossas estratégias estejam alinhadas não apenas com a tecnologia, mas também com princípios robustos de design e usabilidade.

Em resumo, ao implementar nosso projeto, reconhecemos que a retenção de usuários não é apenas uma métrica; é uma jornada contínua de melhoria. Desde a coleta ativa de feedbacks até o planejamento estratégico de atualizações, nossa abordagem visa construir um aplicativo que não apenas conquiste usuários, mas os mantenha satisfeitos e engajados ao longo do tempo.

4 EXPERIMENTO E RESULTADOS

O desenvolvimento de aplicativos requer a implementação de práticas eficazes de teste para garantir a confiabilidade e a qualidade do software. Para garantir uma análise abrangente e fundamentada, nos baseamos em estudos de design centrado no usuário (39) e métodos de avaliação de usabilidade(44). A aplicação desses princípios não apenas aprimorará a experiência do usuário, mas também contribuirá para a identificação de oportunidades específicas de melhoria em aplicativos existentes. Para isso recomendamos alguns testes para validação do aplicativo antes de seu lançamento, a fim de entregar um produto que atenda as dos usuários.

4.1 TESTES UNITÁRIOS E DE INTEGRAÇÃO

No âmbito de testes, recomendamos práticas modernas, como testes automatizados para desenvolvimento mobile em Android (34), para garantir a robustez e confiabilidade do aplicativo em diferentes cenários de uso. Isso não só facilita a manutenção contínua, mas também contribui para a entrega de uma experiência de usuário consistente. Os testes unitários desempenham esse papel ao verificar partes específicas do código, como funções ou métodos, isolando-as para garantir o funcionamento correto de cada unidade (42). Eles ajudam a identificar erros e falhas em componentes específicos, permitindo correções antes que se propaguem pelo sistema.

Por outro lado, os testes de integração avaliam a interação entre diferentes partes do aplicativo, assegurando que todos os componentes funcionem de maneira conjunta e adequada (32). Esses testes detectam problemas que podem surgir quando módulos individuais são combinados, garantindo a coesão do aplicativo como um todo.

A abordagem conjunta de testes unitários e de integração não apenas melhora a robustez do aplicativo, mas também facilita a manutenção contínua, reduzindo bugs e contribuindo para a criação de um produto mais confiável e de alta qualidade.

Durante a fase de implementação do projeto, é recomendada a realização de testes unitários para cada componente, a fim de garantir a funcionalidade, robustez e integridade do código. Inspirados pela metodologia proposta por especialistas como (32) na Clean Architecture, buscaremos aplicar testes rigorosos em diversos componentes-chave do aplicativo, visando uma validação abrangente e eficaz.

Para aumentar a confiabilidade do aplicativo, propusemos os testes focalizando nas Funções e Métodos. Cada função ou método deve ser submetido a uma bateria de testes para verificar se produz os resultados esperados com diferentes conjuntos de dados de entrada. Essa abordagem, respaldada por referências como "Testes automatizados para desenvolvimento mobile em Android"(34), garante que cada unidade de lógica do código funcione corretamente, formando uma base sólida para o desenvolvimento contínuo.

Avançando para as Classes e Objetos, sugerimos a aplicação de testes que asseguram que os métodos e propriedades dessas estruturas se comportem conforme o planejado. A consistência e a correta manipu-

lação de dados dentro dessas classes são verificadas, proporcionando uma camada adicional de confiança na funcionalidade global do aplicativo.

Os Módulos ou Componentes de UI não podem ser negligenciados, os componentes de interface, como botões, campos de entrada e listas, deverão ser minuciosamente testados para garantir que não apenas interajam corretamente entre si, mas também respondam às interações do usuário conforme o esperado. Essa prática, alinhada aos princípios de Design Centrado no Usuário (39), visa criar uma experiência de usuário fluida e livre de problemas.

Outro ponto crítico a ser abordado nos testes unitários é a validação das Regras de Negócio. Algoritmos e regras específicos do aplicativo passarão por uma série de testes para confirmar se estão produzindo os resultados desejados em diferentes cenários. Isso não apenas verificará a eficácia dessas regras, mas também assegura que contribuirá para a experiência do usuário de maneira consistente.

O Banco de Dados e o Acesso a Dados não são esquecidos durante esse processo. As operações de leitura, escrita e manipulação de dados no banco de dados serão testadas para garantir a integridade dos dados e a eficiência das consultas. Referências como "RESTful Api Design"(42) fornecerão insights valiosos sobre as melhores práticas para testar a comunicação eficiente com o banco de dados.

Nossa estratégia de testes unitários é fundamentada na ideia de avaliar cada unidade isoladamente, garantindo sua funcionalidade de maneira independente. Isso criará a confiança de que, quando combinadas, essas unidades operarão harmoniosamente para o correto funcionamento do aplicativo como um todo. A incorporação de boas práticas sugeridas por especialistas em diferentes áreas do desenvolvimento, como Clean Architecture (32) e Design Centrado no Usuário (39), fortalecerá a qualidade e a robustez do projeto em todas as fases.

Os testes automatizados, por sua vez, são essenciais para garantir a estabilidade e a confiabilidade do aplicativo. Com a variedade de dispositivos Android disponíveis, é imperativo realizar testes em diferentes configurações para garantir que o aplicativo funcione corretamente em diversos cenários (34). A automação desses testes permite uma verificação consistente das funcionalidades do aplicativo, acelerando o ciclo de desenvolvimento e reduzindo o risco de regressões.

4.1.1 Testar a integração entre frontend e backend

Para a implementação do projeto, a etapa de teste de integração entre o frontend e o backend deverá garantir uma comunicação eficaz e funcionalidade consistente entre essas duas partes fundamentais do aplicativo. Inspirados por práticas recomendadas, como as descritas em "RESTful Api Design"(42), realizaremos uma abordagem abrangente para validar a interação entre o front-end, responsável pela interface do usuário, e o back-end, encarregado da lógica de negócios e acesso aos dados.

Durante os testes de integração, será examinado a transmissão e recepção de dados entre essas camadas, garantindo que as solicitações feitas pelo front-end sejam processadas com precisão pelo back-end. Referências como "MVP: Model-View-Presenter"(30) fornecem orientações valiosas sobre a validação eficaz dessas interações, garantindo a consistência e a integridade das informações trocadas.

Além disso, a validação dos testes de integração se estenderá à autenticação do comportamento das

APIs, endpoints e serviços utilizados para comunicação. Recomendamos simular uma variedade de cenários de interação entre front-end e back-end, verificando se os dados são transferidos de maneira metódica e segura. Isso incluirá validar a interpretação e o processamento corretos das solicitações, oferecendo uma experiência contínua e sem falhas aos usuários.

Nesse processo de validação, aplicaremos as práticas de "Pro RESTful APIs"(37) para garantir a eficácia da comunicação e a segurança dos dados durante a transferência entre as camadas do aplicativo. Além disso, manteremos uma abordagem orientada pelo Design Centrado no Usuário (39), assegurando que a integração contribua para uma experiência perfeita e totalmente operacional para o usuário final.

A simulação de vários cenários durante os testes de integração também nos permitirá identificar e corrigir possíveis problemas de interoperabilidade entre o frontend e o backend. Dessa forma, garantimos que a comunicação entre as partes do aplicativo seja robusta, segura e eficiente, contribuindo para o sucesso geral do projeto.

4.1.2 Corrigir bugs e problemas identificados

Para o bom desenvolvimento do projeto, a resolução de defeitos e problemas a serem identificados no aplicativo visará garantir sua estabilidade e excelência. Assim os defeitos serão detectados por meio de testes, feedback do usuário ou monitoramento contínuo do aplicativo. Adotaremos medidas proativas para identificar e mitigar esses problemas. Esse processo implicará em uma análise do código-fonte, identificando a raiz subjacente dos defeitos e implementando soluções apropriadas.

Para conduzir essa fase com eficácia, seguiremos metodologias sólidas de desenvolvimento de software, enfatizando a realização de testes complementares para confirmar se a resolução dos defeitos foram bem-sucedida, sem introduzir novas complicações. Referências como "Clean architecture"(32) fornecem diretrizes para a implementação de práticas de desenvolvimento robustas e estruturadas.

A aplicação de testes automatizados, conforme abordado por (34), é parte integrante do nosso processo de resolução de defeitos, garantindo uma verificação sistemática e abrangente das correções implementadas. Além disso, inspiramo-nos em princípios de design centrados no usuário, conforme discutido por (39), para assegurar que as correções não apenas resolverão os defeitos, mas também mantenham uma experiência de usuário consistente e confiável.

Ao adotar abordagens orientadas por padrões de arquitetura, como o Modelo-Visão-Apresentador (MVP) discutido por García em (30), buscaremos criar soluções que não apenas corrigirão defeitos pontuais, mas fortalecerão a estrutura geral do aplicativo. Isso contribuirá para a sustentabilidade e escalabilidade do software ao longo do tempo.

A resolução de defeitos é um processo iterativo, e ao abordar cada problema identificado, priorizaremos a realização de testes abrangentes, assegurando que o aplicativo, após as correções, continue oferecendo uma experiência perfeita aos usuários, mantendo uma abordagem rigorosa e alinhada às melhores práticas de desenvolvimento de software, essa será essencial para manter a qualidade e a confiabilidade do aplicativo em todas as fases do ciclo de vida do desenvolvimento.

Em síntese, a resolução de defeitos é uma medida essencial para garantir a qualidade do software. Ao

adotar metodologias sólidas, realizando testes abrangentes e aplicando as melhores práticas de desenvolvimento, buscaremos criar um aplicativo que não apenas atenda às expectativas, mas também forneça uma experiência excepcional aos usuários.

4.2 TESTES DE USABILIDADE

Realizar testes de usabilidade em um aplicativo é essencial para garantir que a experiência do usuário seja intuitiva, satisfatória e eficiente. Esses testes proporcionarão uma visão direta de como os usuários interagem com o aplicativo, identificando áreas de dificuldade, confusão ou lentidão no uso. Ao entender os padrões de comportamento e as preferências dos usuários, os desenvolvedores poderão aprimorar a interface, o fluxo de navegação e a usabilidade geral do aplicativo. Isso não apenas melhorará a satisfação do usuário, mas também contribuirá para a retenção e fidelização dos usuários, uma vez que um aplicativo mais fácil de usar e mais agradável tende a atrair e manter uma base de usuários mais engajada e fiel.

4.2.1 Realizar testes de usabilidade com usuários reais ou grupos de foco

Os testes de usabilidade serão conduzidos com a participação ativa de usuários reais ou grupos focais definidos em pesquisa apresentada anteriormente. Essa abordagem, em linha com as sugestões de Machado et al. (44), permitirão não apenas a identificação de pontos problemáticos na interação, mas também a compreensão dos padrões de comportamento e preferências dos usuários. A observação direta será essencial para capturar reações genuínas e fornecer insights valiosos sobre como aprimorar a interface, o fluxo de navegação e a usabilidade geral do aplicativo.

Ao adotar princípios do Model-View-Presenter (MVP) de García em (30), nossa abordagem criará interfaces que não apenas atendam às expectativas dos usuários, mas também antecipem suas necessidades. A fase de testes com usuários reais será integrada ao processo de desenvolvimento, permitindo ajustes contínuos à medida que a compreensão da experiência do usuário evolui.

4.2.2 Coletar feedbacks e realizar ajustes na interface e funcionalidades

A coleta de feedback é uma prática contínua que se estende além da fase inicial de testes. Inspirados por abordagens de design iterativo, conforme sugerido por Lidwell et al. (35), buscaremos um ciclo dinâmico de ajustes e melhorias. O feedback dos usuários será analisado minuciosamente, identificando áreas específicas de melhoria na interface, correção de possíveis erros e aprimoramento de funcionalidades.

Ao implementar esses ajustes, adotaremos a filosofia de design universal, garantindo que as melhorias não criem barreiras ou excluam nenhum grupo de usuários. Isso se alinha aos princípios do design inclusivo, reconhecendo a diversidade de usuários e adaptando o aplicativo para atender a uma ampla gama de necessidades e habilidades.

A abordagem de testes de usabilidade, combinada com feedback contínuo e ajustes iterativos, não será apenas uma prática, mas uma mentalidade incorporada ao processo de desenvolvimento. O objetivo será

criar um aplicativo que não apenas atenda às expectativas iniciais, mas evolua organicamente em resposta ao feedback dos usuários e às mudanças no cenário tecnológico. Isso não apenas aprimorará a satisfação do usuário, mas estabelecerá um ciclo virtuoso de melhoria contínua, posicionando o aplicativo como líder em termos de experiência do usuário em seu domínio.

4.3 RESULTADOS ESPERADOS DA IMPLEMENTAÇÃO DO APLICATIVO DE MONITORAMENTO DE CONDUTORES

A implementação do aplicativo de monitoramento de condutores, embasada em uma abordagem científica e técnica, visa alcançar resultados expressivos que repercutam positivamente na experiência do usuário e na eficácia do sistema como um todo. As expectativas abrangem áreas cruciais, desde a segurança e desempenho do veículo até a interatividade intuitiva com o usuário. A seguir, estão alguns os resultados esperados desta implementação.

- Melhoria na segurança veicular por meio do monitoramento contínuo do comportamento do condutor - A análise de dados provenientes dos sensores e dispositivos OBD2 possibilitará a identificação de padrões de direção imprudente, permitindo intervenções preventivas. Espera-se, assim, uma redução de incidentes e acidentes de trânsito, contribuindo para um ambiente rodoviário mais seguro.(5)
- Integração de tecnologias de telemetria e IoT - Permitirá otimizar a eficiência operacional dos veículos, por meio de um diagnóstico preciso do estado do veículo, incluindo informações sobre o motor, transmissão e sistemas elétricos, permitindo que os condutores tomem ações proativas na manutenção, antecipando uma redução nos custos operacionais, já que os usuários serão capazes de realizar intervenções preventivas antes que problemas se agravem.
- Eficiência na usabilidade - Esperamos que a implementação de testes e métricas de usabilidade resulte em uma interação intuitiva e uma experiência do usuário aprimorada. Os condutores devem se sentir confortáveis e confiantes ao utilizar o aplicativo, facilitando a adesão e a utilização contínua. A interface limpa e amigável, combinada com funcionalidades intuitivas, contribuirá para a satisfação do usuário.
- Integração de dados provenientes de APIs - Esperamos contribuir significativamente para a segurança viária em nível nacional. O acesso a informações em tempo real sobre condições de tráfego, acidentes e áreas de risco permitirá aos condutores tomar decisões mais informadas e seguras durante suas viagens além de absorver bônus por sua condução (15).
- Segurança de redes e dados - O Cumprimento das diretrizes legais e conformidade com a Lei Geral de Proteção de Dados Pessoais (4) e a Resolução nº. 810 do Contran (2) por meio de implementação de protocolos avançados de segurança, como criptografia de dados, assegurará a privacidade e confidencialidade das informações dos usuários, contribuindo para a construção de uma reputação de confiança.
- Avaliação de desempenho do App - Esperamos estabelecer um ciclo contínuo de análise e melhoria para realizar identificações de áreas de oportunidade, baseada em dados concretos, permitindo

ajustes e aprimoramentos regulares no aplicativo. A busca incessante por inovação e eficiência é um compromisso que norteará o desenvolvimento contínuo do aplicativo.

A implementação do aplicativo de monitoramento de condutores visa não apenas atender às expectativas técnicas e científicas, mas também superá-las, proporcionando uma solução avançada, segura e eficaz para condutores e usuários em geral.

5 CONCLUSÃO

Ao longo deste estudo, realizamos uma proposta de desenvolvimento de um aplicativo inovador de monitoramento de condutores que utilizará diversas tecnologias para avaliar a condução de forma abrangente e eficaz. Os resultados esperados destacarão a capacidade do aplicativo em melhorar a segurança veicular e de condução, oferecendo uma análise detalhada do comportamento do condutor por meio de sensores e dispositivos conectados. A utilização de algoritmos comparativos permitirão uma avaliação precisa e eficaz, identificando padrões que podem indicar práticas de direção imprudentes e sugerindo intervenções preventivas.

A relevância desta pesquisa no contexto atual é substancial, considerando a crescente preocupação com a segurança no trânsito e a necessidade de inovações no setor de seguros automotivos (63). Nossa proposta de aplicativo não apenas influencia positivamente a segurança nas estradas, mas também redefine a maneira como a indústria de seguros avalia e estabelece prêmios com base no comportamento do condutor. A implementação bem-sucedida dessa tecnologia representa um avanço significativo para promover uma condução mais segura e eficiente (64).

A metodologia adotada neste estudo revelou-se robusta, proporcionando uma base sólida para a proposta de desenvolvimento do aplicativo. A integração de tecnologias de telemetria e IoT permitirão uma avaliação abrangente do comportamento do condutor. No entanto, reconhecemos desafios relacionados à diversidade de dispositivos e ambientes de condução, que exigirão ajustes durante o desenvolvimento. A adaptação contínua será crucial para superar essas dificuldades e garantir a eficácia do aplicativo.

As implicações práticas deste trabalho são vastas e impactarão diretamente na segurança dos condutores e a indústria de seguros. A proposta de aplicativo desenvolvido pode ser aplicado no mundo real para promover a segurança veicular e de condução, proporcionando feedback imediato aos condutores sobre seu comportamento. Além disso, as informações coletadas podem ser compartilhadas com seguradoras para uma avaliação mais precisa do risco, influenciando as políticas de prêmios e incentivando a condução responsável.

Este estudo enfrentou algumas limitações, incluindo a variabilidade de dispositivos e condições de condução. Além disso, questões éticas relacionadas à privacidade dos condutores foram consideradas, levando à implementação de protocolos avançados de segurança. Admitimos que a aceitação e adoção do aplicativo podem ser desafios adicionais, exigindo esforços contínuos para conscientização e educação, apesar da pesquisa realizada 83% de aprovação.

Para pesquisas futuras, sugerimos explorar a implementação de tecnologias emergentes, como inteligência artificial, machine learning, processamento de linguagem natural e visão computacional para aprimorar ainda mais a capacidade do aplicativo em avaliar o comportamento do condutor. Além disso, investigar estratégias para aumentar a aceitação do aplicativo pelos condutores e avaliar seu impacto a longo prazo na redução de acidentes rodoviários, o que seria uma direção promissora.

Este trabalho destaca a importância da inovação tecnológica na promoção da segurança veicular e na evolução do setor de seguros automotivos. A proposta de aplicativo desenvolvida representa um marco

significativo, oferecendo uma solução abrangente e eficaz para monitoramento de condutores. Ao incentivar uma condução mais segura e eficiente, as tecnologias empregada neste estudo contribuirão para um cenário rodoviário mais seguro, alinhando as demandas atuais por avanços na indústria de seguros. Este trabalho representa não apenas uma conquista em termos de desenvolvimento tecnológico, mas também um compromisso com a segurança e inovação contínuas no contexto automotivo.

REFERÊNCIAS BIBLIOGRÁFICAS

- 1 TRANSPORTE, C. Confederação Nacional do. *Painel CNT de consultas dinâmicas dos acidentes rodoviários. 2022 - íntegra. Brasília 2023*. [S.l.]: Disponível em: <<https://www.cnt.org.br/pesquisas/>> . Acessado em: 22 jul. 2023, 2023.
- 2 TRÂNSITO, C. Conselho Nacional de. *Resolução nº. 810, de 15 de Dezembro de 2020. Dispõe sobre a classificação de danos e os procedimentos para a regularização, a transferência e a baixa dos veículos envolvidos em acidentes. Brasília, DF, 2020*. [S.l.]: Disponível em: <<http://www.denatran.gov.br/resolucoes.htm>> . Acessado em: 22 jul. 2023, 2020.
- 3 MOBILETIME. *Mobiletime (c2022). Pesquisa de uso de app no Brasil, c2022*. [S.l.]: Disponível em: <<https://www.mobiletime.com.br/pesquisas/uso-de-apps-no-brasil-maio-de-2023/>> . Acessado em: 22 jul. 2023, 2022.
- 4 DEPUTADOS, C. d. d. e. I. Câmara dos. *Lei Geral de Proteção de Dados Pessoais*. [S.l.]: Disponível em: <https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm> . Acessado em: 22 jul. 2023, 2018.
- 5 SILVA, C. C.; CALDAS, F. L. de; MACHADO, F. D.; MENDONÇA, F. L.; JR, R. T. de S. Proposta de auto-registro de serviços pelos dispositivos em ambientes de iot. *34º Simpósio Brasileiro de Telecomunicações e Processamento de Sinais*, 2016.
- 6 QUINTANILHA, S. O automóvel no seu rádio. *Revista Alterjor*, v. 19, n. 1, p. 251–263, 2019.
- 7 RIBEIRO, T. M.; CASTRO, I. S.; BOTELHO, R. D. Sistemas multimeios: Conectividade para uma experiência de uso em automóveis. *Ergodesign & HCI*, v. 9, n. 2, p. 87–103, 2021.
- 8 SIQUEIRA, T. S. de. Bluetooth—características, protocolos e funcionamento. 2006.
- 9 PUIG, B. P. *Smartphones for smart driving: a proof of concept*. Dissertação (Mestrado) — Universitat Politècnica de Catalunya, 2013.
- 10 VICENTIN, D. *A reticulação da banda larga móvel: definindo padrões, informando a rede*. Tese (Doutorado) — [sn], 2016.
- 11 SHUKLA, S.; KHARE, V.; GARG, S.; SHARMA, P. Comparative study of 1g, 2g, 3g and 4g. *J. Eng. Comput. Appl. Sci*, v. 2, n. 4, p. 55–63, 2013.
- 12 BHANDARI, N.; DEVRA, S.; SINGH, K. Evolution of cellular network: from 1g to 5g. *International journal of engineering and techniques*, v. 3, n. 5, p. 98–105, 2017.
- 13 CARDOSO, A. C. M.; ALMEIDA, P. F. d. O “breque dos apps” contra o falso discurso de autonomia e flexibilidade por parte das plataformas. *Escuta, Revista de Política e Cultura*, 2020.
- 14 AD, G. *Digital Brazil - Insights do report anual de plataformas digitais, 2023*. [S.l.]: Disponível em: <<https://globalad.com.br/blog/digital-brazil-2023/>> . Acessado em: 17 fev. 2024, 2024.
- 15 LUPIĆ, A. Primena apigee platforme za upravljanje api-jem. *Zbornik radova Fakulteta tehničkih nauka u Novom Sadu*, v. 35, n. 09, p. 1559–1562, 2020.
- 16 BARBOSA, V. H. F.; BESSA, G. M. A. Um estudo comparativo sobre as linguagens java e kotlin para o desenvolvimento de aplicativos android. *Caderno de Estudos em Sistemas de Informação*, v. 7, n. 1, 2022.

- 17 DIMARZIO, J. *Beginning android programming with android studio*. [S.l.]: John Wiley & Sons, 2016.
- 18 JÚNIOR, A. A.; MEIRELES, L. de S.; FIGUEIRA, L. A. R.; CARMO, V. M. M.; MARQUES-NETO, H. T.; XAVIER, L. Entendendo o engajamento das comunidades front-end e back-end nos repositórios do github. In: SBC. *Anais do X Workshop de Visualização, Evolução e Manutenção de Software*. [S.l.], 2022. p. 26–30.
- 19 EITER, T.; FABER, W.; LEONE, N.; PFEIFER, G. The diagnosis frontend of the dlvs system. *AI Communications*, IOS Press, v. 12, n. 1-2, p. 99–111, 1999.
- 20 CASTIGNANI, G.; FRANK, R.; ENGEL, T. Driver behavior profiling using smartphones. In: IEEE. *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. [S.l.], 2013. p. 552–557.
- 21 CASTIGNANI, G.; DERRMANN, T.; FRANK, R.; ENGEL, T. Driver behavior profiling using smartphones: A low-cost platform for driver monitoring. *IEEE Intelligent transportation systems magazine*, IEEE, v. 7, n. 1, p. 91–102, 2015.
- 22 SAIPRASERT, C.; THAJCHAYAPONG, S.; PHOLPRASIT, T.; TANPRASERT, C. Driver behaviour profiling using smartphone sensory data in a v2i environment. In: IEEE. *2014 International Conference on Connected Vehicles and Expo (ICCVE)*. [S.l.], 2014. p. 552–557.
- 23 SAIPRASERT, C.; PHOLPRASIT, T.; PATTARA-ATIKOM, W. Detecting driving events using smartphone. In: *Proceedings of the 20th ITS World Congress*. [S.l.: s.n.], 2013. v. 11, n. 11.
- 24 FERREIRA, J.; CARVALHO, E.; FERREIRA, B. V.; SOUZA, C. de; SUHARA, Y.; PENTLAND, A.; PESSIN, G. Driver behavior profiling: An investigation with different smartphone sensors and machine learning. *PLoS one*, Public Library of Science San Francisco, CA USA, v. 12, n. 4, p. e0174959, 2017.
- 25 CASOTTI, B. O.; NASCIMENTO, D. G.; PEREIRA, R. P.; CABRINI, F. H. Sistema para análise do perfil de condutores de veículos segurados através de técnicas de iot. *FaSci-Tech*, v. 1, n. 13, 2018.
- 26 SILVA, D. A. da; TORRES, J. A. S.; PINHEIRO, A.; FILHO, F. L. de C.; MENDONÇA, F. L.; PRACIANO, B. J.; KFOURI, G. de O.; SOUSA, R. T. de. Inference of driver behavior using correlated iot data from the vehicle telemetry and the driver mobile phone. In: IEEE. *2019 Federated Conference on Computer Science and Information Systems (FedCSIS)*. [S.l.], 2019. p. 487–491.
- 27 GOMES, I. M. et al. Como elaborar uma pesquisa de mercado. *Belo Horizonte: SEBRAE Minas*, 2013.
- 28 PINHEIRO, R. M. *Pesquisa de mercado*. [S.l.]: Editora FGV, 2015.
- 29 LATTIN, J.; CARROLL, J. D.; GREEN, P. E. Análise de dados multivariados. *São Paulo: Cengage Learning*, v. 475, 2011.
- 30 GARCÍA, R. F. Mvp: Model–view–presenter. In: *iOS Architecture Patterns: MVC, MVP, MVVM, VIPER, and VIP in Swift*. [S.l.]: Springer, 2023. p. 107–144.
- 31 GAUDIOSO, V. *Foundation Expression Blend 4 with Silverlight*. [S.l.]: Apress, 2011.
- 32 MARTIN, R. C. *Clean architecture*. [S.l.]: Prentice Hall, 2017.
- 33 FERREIRA K. G., d. C. M. D. F. . d. S. C. I. P. *Teste de usabilidade*. 2002.

- 34 FREITAS, N. A. M. d. Testes automatizados para desenvolvimento mobile em android. UFAL Campus Arapiraca, 2022.
- 35 LIDWELL, W.; HOLDEN, K.; BUTLER, J. Princípios universais do design. *Tradução: Francisco Araújo da Costa. Porto Alegre: Bookman*, 2010.
- 36 JACOBSON, D.; BRAIL, G.; WOODS, D. *APIs: A strategy guide*. [S.l.]: " O'Reilly Media, Inc.", 2012.
- 37 PATNI, S. *Pro RESTful APIs*. [S.l.]: Springer, 2017.
- 38 MARDAN, A.; MARDAN, A. Putting frontend and backend together. *Full Stack JavaScript: Learn Backbone.js, Node.js, and MongoDB*, Springer, p. 257–287, 2018.
- 39 LOWDERMILK, T. *Design Centrado no Usuário: um guia para o desenvolvimento de aplicativos amigáveis*. [S.l.]: Novatec Editora, 2019.
- 40 WEN, M.; WU, R.; CHEUNG, S.-C. Locus: Locating bugs from software changes. In: *Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering*. [S.l.: s.n.], 2016. p. 262–273.
- 41 MANOVICH, L. Banco de dados. *Revista ECO-Pós*, v. 18, n. 1, p. 7–26, 2015.
- 42 BIEHL, M. *RESTful Api Design*. [S.l.]: API-University Press, 2016. v. 3.
- 43 PAU, V. C.; MIHAILESCU, M. I.; STANESCU, O. Model view presenter design pattern. *Journal of Computer Science & Control Systems*, v. 3, n. 1, 2010.
- 44 MACHADO, L.; VERGARA, L.; FERREIRA, E. Métodos de avaliação de usabilidade: características e aplicações. *Anais [3º CONEPRO-SUL]*, 2014.
- 45 VIRGENS, G. B. das; SILVEIRA, M. S. Aplicação de métricas de usabilidade em protótipos computadorizados de baixa fidelidade.
- 46 FIGUEIREDO, E. Requisitos funcionais e requisitos não funcionais. *Icex, Dcc/Ufmg*, 2011.
- 47 DEMICO, D. C. C.; CALIL, G. B.; SANTOS, G. C. d.; TAKAMURA, I. A. Aplicativo de opiniões: feedback. 136, 2021.
- 48 HOEFLING, S. Model-view-viewmodel (mvvm). In: *Getting Started with the Uno Platform and WinUI 3: Hands-On Building of Cross-Platform Desktop, Mobile, and Web Applications That Can Run Anywhere*. [S.l.]: Springer, 2022. p. 191–206.
- 49 OLIVEIRA, L. D. de; SOUZA, T. B.; FERNANDES, F. M. L.; OLIVEIRA, T. L. de; SOUZA, C. P. de. Desenvolvimento e avaliação de desempenho de um sistema embarcado para telemetria automotiva baseado em lora. In: *Congresso Brasileiro de Automática-CBA*. [S.l.: s.n.], 2022. v. 3, n. 1.
- 50 PROENÇA, G. A.; RODRIGUES, R. B. et al. Evolução de uma plataforma web para execução de aplicativos android desenvolvidos com app inventor. Florianópolis, SC., 2023.
- 51 GOOGLE, B. *Notas da versão do SDK Platform, 2024*. [S.l.]: Disponível em: <<https://developer.android.com/>> . Acessado em: 14 jan. 2024, 2024.
- 52 ASTARITA, V.; GUIDO, G.; MONGELLI, D.; GIOFRÈ, V. P. A co-operative methodology to estimate car fuel consumption by using smartphone sensors. *Transport*, v. 30, n. 3, p. 307–311, 2015.
- 53 MORONEY, L.; MORONEY, L. Cloud functions for firebase. *The Definitive Guide to Firebase: Build Android Apps on Google's Mobile Platform*, Springer, p. 139–161, 2017.

- 54 BIEHL, M. *API Architecture*. [S.l.]: API-University Press, 2015. v. 2.
- 55 AMARASINGHE, M.; KOTTEGODA, S.; ARACHCHI, A. L.; MURAMUDALIGE, S.; BANDARA, H. D.; AZEEZ, A. Cloud-based driver monitoring and vehicle diagnostic with obd2 telematics. In: IEEE. *2015 fifteenth international conference on advances in ICT for emerging regions (ICTer)*. [S.l.], 2015. p. 243–249.
- 56 CIURANA, E. *Developing with google app engine*. [S.l.]: Apress, 2009.
- 57 KRISHNAN, S.; GONZALEZ, J. L. U. *Building your next big thing with google cloud platform: A guide for developers and enterprise architects*. [S.l.]: Springer, 2015.
- 58 KRISHNAN, S.; GONZALEZ, J. L. U. *Building your next big thing with google cloud platform: A guide for developers and enterprise architects*. [S.l.]: Springer, 2015.
- 59 MARQUES, B. L.; SILVA, M. L. da; MOREIRA, J. P. Interface do usuário. *Seminário de Tecnologia Gestão e Educação*, v. 1, n. 2, p. 27–30, 2019.
- 60 DAMIANI, A.; SILVA, A. C. P. d.; COSTA, F. F. T.; FILHO, J. A. C.; TEIXEIRA, L. C.; CRUZ, T. R. R. d. Plano de negócios: aplicativo segura a condução. Vitória, 2019.
- 61 CARVALHO, P. E. Lógicas visuais. *Sinais de Cena*, p. 10–12, 2004.
- 62 ZABALETA, J. P. L. *Matriz de priorização: uma ferramenta para estabelecer prioridades*. [S.l.]: Embrapa Clima Temperado, 2002.
- 63 SANTOS, A. T. d. Análise de satisfação de clientes do ramo de seguros. DECAT-Departamento de Estatística e Ciências Atuariais–Ciências Atuariais . . . , 2022.
- 64 MÜLLER, B. D. Prêmio puro dpvat 2021 após a resolução cnsf 399. 2022.

APÊNDICES