



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE GRADUAÇÃO EM CIÊNCIAS DA COMPUTAÇÃO

Breno De Brida

Desenvolvimento de um aplicativo mobile para troca de objetos

Florianópolis
2024

Breno De Brida

Desenvolvimento de um aplicativo mobile para troca de objetos

Trabalho de Conclusão de Curso submetida ao Curso de Graduação em Ciências da Computação da Universidade Federal de Santa Catarina para a obtenção do título de Bacharel em Ciências da Computação.

Orientador: Prof. Raul Sidnei Wazlawick, Dr.

Florianópolis

2024

Ficha catalográfica gerada por meio de sistema automatizado gerenciado pela BU/UFSC.
Dados inseridos pelo próprio autor.

De Brida, Breno
Desenvolvimento de um aplicativo mobile para troca de
objetos / Breno De Brida ; orientador, Raul Sidnei
Wazlawick, 2024.
67 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Centro Tecnológico,
Graduação em Ciências da Computação, Florianópolis, 2024.

Inclui referências.

1. Ciências da Computação. 2. Aplicativo Móvel. 3. Troca
de Objetos. I. Sidnei Wazlawick, Raul. II. Universidade
Federal de Santa Catarina. Graduação em Ciências da
Computação. III. Título.

Breno De Brida

Desenvolvimento de um aplicativo mobile para troca de objetos

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do título de Bacharel em Ciências da Computação e aprovado em sua forma final pelo Curso Ciências da Computação

Florianópolis, 04 de julho de 2024.

Coordenação do Programa de Graduação

Banca examinadora

Prof. Raul Sidnei Wazlawick, Dr.
Orientador

Mauricio Floriano Galimberti, Dr.
Universidade Federal de Santa Catarina

Rafael Luiz Cancian, Dr.
Universidade Federal de Santa Catarina

Florianópolis, 2024.

AGRADECIMENTOS

Agradeço ao meu pai, Alcides, por ter me dado a ideia deste projeto e por sua inestimável contribuição durante todo o seu desenvolvimento. À minha mãe, Elizandra, que sempre me deu todo o amor e suporte educacional desde o meu nascimento, incluindo meu pai. Aos meus irmãos, Bruno e Bernardo, por todo o apoio e ajuda ao longo do meu crescimento.

À Sandy, minha maior parceira, que amo profundamente e que foi meu maior suporte durante toda a jornada na realização deste trabalho. Sandy, que carrega nosso filho Beron, ao qual também dedico todo o meu amor.

Ao meu orientador, Raul, por seu apoio e orientação no desenvolvimento deste trabalho.

E, por fim, agradeço aos meus amigos do grupo AZCD por todos os momentos de descontração e vivências juntos, que tornaram esta jornada ainda mais especial.

*"The best way to predict the future is to invent it."
(Kay, Alan)*

RESUMO

Em 2021, o *e-commerce*¹ brasileiro registrou um faturamento recorde de R\$ 161 bilhões, um aumento de 26,9% em relação ao ano anterior (NEOTRUST, 2022). Atualmente, existem poucos aplicativos voltados especificamente para a troca de bens materiais pela internet. Tendo isso em vista, o projeto em questão foi desenvolvido para atender uma necessidade de mercado pouco explorada, com o desenvolvimento de um aplicativo móvel, visando viabilizar um mercado de troca. Os usuários cadastram o objeto que desejam trocar e também realizam ofertas por outros objetos. Uma vez cadastrado, o sistema cruza as informações com as já existentes em seu banco de dados e verifica a existência de outro usuário com o objeto de interesse e que também tenha interesse no objeto do primeiro usuário. Caso o usuário não possua um objeto de interesse específico, o sistema indica os objetos de outros usuários que possuem interesse naquele que ele possui.

Palavra-chave: Troca. Aplicativo móvel. Objeto de interesse.

¹ e-commerce: termo do inglês para se referenciar a comércio eletrônico(eletronic commerce).

ABSTRACT

In 2021, Brazilian e-commerce recorded a record invoicing of BRL 161 billion, an increase of 26.9% compared to the previous year (NEOTRUST, 2022). Currently, there are few applications specifically designed for exchanging material goods over the internet. With this in mind, the project in question was developed to meet a underexplored market need, with the development of a mobile application, aiming to enable an exchange market. Users register the object they wish to exchange and also make offers for other objects. Once registered, the system cross-references the information with that already existing in its database and checks for the existence of another user with the object of interest and who is also interested in the object of the first user. If the user does not have a specific object of interest, the system indicates the objects of other users who are interested in the one he has.

Keywords: Exchange. Mobile application. Object of interest.

LISTA DE FIGURAS

Figura 1 – Ambiente de desenvolvimento do Figma	16
Figura 2 – Visão geral da arquitetura do FCM	19
Figura 3 – Diagrama de casos de uso	26
Figura 4 – Ícone do aplicativo - Figma	28
Figura 5 – Tela de Login - Figma	29
Figura 6 – Tela de Explorar - Figma	29
Figura 7 – Tela do Objeto - Figma	30
Figura 8 – Tela da Conta - Figma	30
Figura 9 – Diagrama de Arquitetura	32
Figura 10 – Depuração via USB	33
Figura 11 – Diagrama de Sequência - Login via Google	34
Figura 12 – Diagrama de Sequência - Login via Facebook	35
Figura 13 – Tela de Login	35
Figura 14 – Diagrama de Sequência - Busca de objetos	36
Figura 15 – Tela de Explorar	37
Figura 16 – Diagrama de Sequência - Filtros	39
Figura 17 – Tela de Filtro	40
Figura 18 – Diagrama de Sequência - Cadastro de objeto	41
Figura 19 – Tela de Cadastro de Objeto	41
Figura 20 – Tela do Objeto	42
Figura 21 – Tela de Ofertas - Aba "Enviadas"	43
Figura 22 – Tela de Ofertas - Aba "Recebidas"	44
Figura 23 – Tela de Ofertas - Aba "Efetuadas"	45
Figura 24 – Diagrama de Sequência - Sugestões de Troca	46
Figura 25 – Tela de Sugestões de Trocas	46
Figura 26 – Tela do Chat	47
Figura 27 – Idade dos participantes	49
Figura 28 – Pergunta 1	49
Figura 29 – Pergunta 2	50
Figura 30 – Pergunta 3	50
Figura 31 – Pergunta 4	51
Figura 32 – Pergunta 5	51
Figura 33 – Pergunta 6	52
Figura 34 – Pergunta 7	53

LISTA DE TABELAS

Tabela 1 – Comparativo dos aplicativos	22
--	----

LISTA DE SIGLAS E ABREVIATURAS

API	<i>Application Programming Interface</i>
AVD	<i>Android Virtual Device</i>
CPU	<i>Central Processing Unit</i>
GPU	<i>Graphics Processing Unit</i>
IA	<i>Inteligência Artificial</i>
RAM	<i>Random Access Memory</i>
SDK	<i>Software Development Kit</i>
SO	<i>Sistema Operacional</i>
UF	<i>Unidade Federativa</i>
URL	<i>Uniform Resource Locator</i>
UI	<i>User Interface</i>
USB	<i>Universal Serial Bus</i>

SUMÁRIO

1	INTRODUÇÃO	13
1.1	OBJETIVOS	14
1.1.1	Objetivo Geral	14
1.1.2	Objetivos Específicos	14
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	APLICATIVO MÓVEL	15
2.2	TECNOLOGIAS DISPONÍVEIS NO MERCADO	15
2.3	TECNOLOGIAS UTILIZADAS PARA O DESENVOLVIMENTO DO APLICATIVO	16
2.3.1	Figma	16
2.3.2	Flutter	17
2.3.3	Dart	17
2.3.4	Widgets	17
2.3.5	Firebase	18
3	ESTADO DA ARTE	20
3.1	APLICATIVOS CORRELATOS	20
3.1.1	Permutó	20
3.1.2	Finpli	20
3.1.3	Trokaí	21
3.1.4	Troca Fácil	21
3.2	COMPARATIVO	21
3.3	RISCOS	23
3.3.1	Identificação dos Riscos	23
3.3.2	Soluções existentes	23
3.3.3	Solução proposta	24
3.4	LEVANTAMENTO DE REQUISITOS	24
3.4.1	Requisitos Não-Funcionais	25
3.4.2	Requisitos Funcionais	25
3.5	CASOS DE USO	26
4	DESENVOLVIMENTO	28
4.1	DESIGN DA INTERFACE	28
4.2	DIAGRAMA DE ARQUITETURA	31
4.3	CONFIGURAÇÃO DO AMBIENTE PARA O DESENVOLVIMENTO	32
4.3.1	Vantagens da Depuração Via USB	32
4.3.2	Estrutura de Pastas e Organização do Código	33
4.4	DESENVOLVIMENTO DO APLICATIVO	33
4.4.1	Login/Cadastro e Autenticação	34

4.4.2	Consulta de Objetos e Filtro	36
4.4.3	Inserção de Objetos	40
4.4.4	Oferta de Objetos	42
4.4.5	Algoritmo de Sugestões de Troca	45
4.4.6	Chat	47
4.4.7	Integração de Serviços Externos	48
4.5	TESTES E VALIDAÇÃO	48
4.5.1	Pesquisa de Teste de Usabilidade	48
5	CONCLUSÃO	54
5.1	TRABALHOS FUTUROS	55
	REFERÊNCIAS	56
	APÊNDICE A – CÓDIGO	58
	APÊNDICE B – ARTIGO	59

1 INTRODUÇÃO

Segundo o novo artigo do World Bank (KAZA *et al.*, 2018), a produção de lixo mundial irá aumentar em 70%, o quantitativo que em 2016 era de 2,01 Bilhões de toneladas está estimado a saltar para 3,4 Bilhões nos próximos 30 anos. A reciclagem de objetos sólidos é crítica para uma sociedade inclusiva, saudável e principalmente sustentável. Esse crescimento agressivo se dá tanto pelo crescimento populacional, mas principalmente pela cultura de consumismo global. Pessoas ao redor do mundo estão continuamente comprando objetos para sua satisfação pessoal e também para o uso coletivo. De acordo com Slater (SLATER, 2002), o consumo aumentou devido à redução dos preços com a industrialização, isso fomentou ainda mais o comércio acarretando numa crescente cultura consumista ao redor do mundo. Com a vinda da internet na década de 90, esse processo acabou sendo empurrado ainda mais com a criação dos *e-commerces*.

Um *e-commerce*, comércio eletrônico do inglês, é um ambiente online de compra e venda de produtos que estruturam um negócio. Dessa forma, todas as operações são realizadas na internet por meio de alguma plataforma ou serviço, uma revolução dos meios de varejo físico tradicional. Seu surgimento foi em 1970 nos Estados Unidos por uma necessidade de troca de arquivos e solicitação de pedidos. No Brasil o primeiro registro de e-commerce se deu na década de 90, com um site de venda de livros idealizado pela empresa *Booknet*¹.

Com o crescimento exponencial do comércio eletrônico a cada ano, com o elevado aumento do consumo pelas pessoas, com a facilidade de aquisição de novos produtos a cada dia, a indústria lançando novos produtos e com vida útil decrescente, há que se pensar no lixo produzido e descartado pela humanidade. É necessário pensar não somente no hoje, mas no amanhã, daqui a 10, 50, 100 anos, quando nosso planeta poderá estar saturado de seus recursos naturais devido ao consumo proporcionado pela indústria e o mercado. Quando houver escassez de recursos e determinados produtos não mais puderem ser produzidos e entregues pela indústria e o mercado, teremos um sistema que permitirá a seus usuários adquirir daqueles que possuem e não precisam mais.

Atualmente no Brasil existem milhares de plataformas de comércio online, porém, apenas com as opções de comprar ou vender produtos, poucas permitem que um usuário possa trocar um objeto com outro usuário. Podem ser citadas aqui, Permuto (PERMUTO, 2023) e Trokaí (TROKAÍ, 2023), plataformas que permitem a troca de serviços e roupas, respectivamente. Até o momento atual, foram encontradas poucas plataformas que permitem a troca de objetos no geral, apenas de áreas específicas como aquelas citadas.

¹ Booknet: primeira empresa de venda de livros online.

Pensando nisso e visando a viabilidade de evitar-se descartes de produtos desnecessários para seus usuários, evitando aquisição de novos produtos no mercado de consumo, quando um produto usado por outra pessoa e que não lhe interessa mais, mas interessa ainda e podendo ser útil a outra pessoa, é que este projeto foi concebido.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

O objetivo geral deste TCC é desenvolver uma aplicação para dispositivos móveis que conecte dois ou mais usuários que possuem interesse em desfazer-se de algum produto de uso pessoal em troca de outro de valor similar e aceito por ambos. As trocas ocorrerão entre usuários, o sistema apenas colocará as pessoas e os produtos de interesse em contato. Poderão ser transacionados sem custos veículos, imóveis, artigos de uso pessoal, objetos, e tudo o que for tangível para o ser humano.

1.1.2 Objetivos Específicos

- Desenvolver um aplicativo móvel multiplataforma para Android e iOS.
- Permitir que um usuário consiga cadastrar um objeto e buscar por objetos do seu interesse para troca, cruzando as informações do banco de dados para encontrar outro usuário com o objeto de interesse.
- Permitir que o usuário busque outros objetos em uma galeria mesmo que não tenha cadastrado ainda nenhum objeto.
- Autenticação de login via Facebook, Google, ou cadastro interno no aplicativo.
- Criar mecanismos de notificação para alertar o usuário assim que o sistema encontrar outro usuário com o mesmo interesse.
- Permitir que o usuário gerencie seus objetos já cadastrados, podendo alterar fotos e descrição.
- Permitir que o usuário altere seus dados pessoais.
- Criar um chat de conversa para aqueles usuários que concordarem trocar seus objetos.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 APLICATIVO MÓVEL

Um aplicativo móvel, também conhecido como app, é um software desenvolvido para ser executado em dispositivos móveis, como smartphones e tablets. Esses aplicativos podem ser nativos, desenvolvidos especificamente para um sistema operacional (iOS ou Android), ou multiplataforma, capazes de rodar em diferentes sistemas operacionais com um único código-base.

A estrutura de um aplicativo móvel é composta por várias camadas que trabalham juntas para proporcionar uma experiência de usuário fluida e funcional. Entre essas camadas, destacam-se a interface do usuário (UI), onde os usuários interagem diretamente com o aplicativo. Inclui todos os elementos visuais, como botões, ícones e menus. A lógica de negócio contém as regras e operações que definem o funcionamento do aplicativo. É responsável pelo processamento de dados e a aplicação de lógica específica. O banco de dados, onde os dados são armazenados e gerenciados, pode ser local (armazenado no dispositivo) ou remoto (armazenado em servidores). O backend é a infraestrutura que suporta o aplicativo, fornecendo serviços como autenticação, armazenamento de dados e notificações.

2.2 TECNOLOGIAS DISPONÍVEIS NO MERCADO

O desenvolvimento de um aplicativo requer a seleção criteriosa de diversas tecnologias para garantir seu sucesso. Esta seção tem como objetivo justificar a necessidade dessas escolhas, destacando a importância de cada uma delas e comparando com alternativas disponíveis no mercado.

A escolha de um framework de desenvolvimento é essencial para a criação de aplicativos eficientes e escaláveis. Frameworks fornecem uma estrutura robusta que facilita o desenvolvimento, manutenção e escalabilidade do código. No mercado, as opções mais conhecidas incluem Flutter, React Native e Xamarin. A escolha do Flutter, por exemplo, foi motivada pela sua performance superior e a riqueza de componentes nativos.

Um banco de dados eficiente é crucial para armazenar e gerenciar dados de maneira segura e escalável. Além disso, uma infraestrutura de backend robusta é necessária para suportar autenticação de usuários, armazenamento de dados, e outras funcionalidades críticas. Entre as opções populares estão Firebase, AWS Amplify e Backendless. A escolha do Firebase foi justificada pela integração perfeita com o Flutter e pela simplicidade na gestão de serviços backend.

Para garantir uma experiência de usuário de alta qualidade, a escolha de uma plataforma de design de interfaces é fundamental. Ferramentas como Figma, Adobe

XD e Sketch são amplamente utilizadas. A escolha do Figma foi motivada pela sua facilidade no uso.

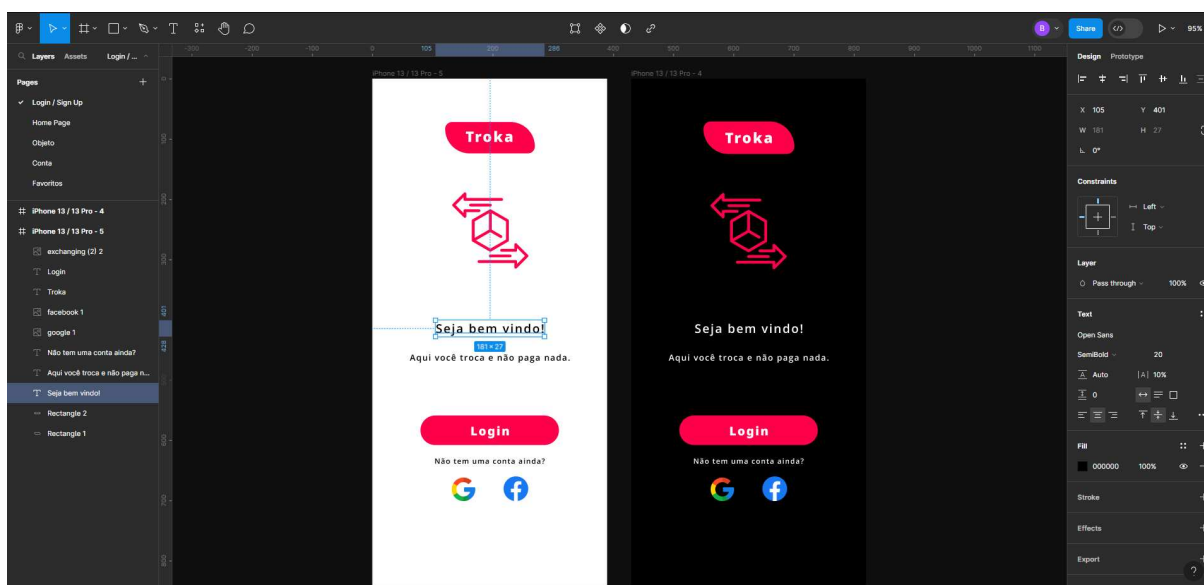
2.3 TECNOLOGIAS UTILIZADAS PARA O DESENVOLVIMENTO DO APLICATIVO

No cenário atual de desenvolvimento de aplicativos mobile, a escolha das tecnologias certas pode determinar o sucesso ou fracasso de um projeto. Considerações como desempenho, compatibilidade, escalabilidade e facilidade de manutenção são cruciais. As tecnologias selecionadas foram adotadas com base em sua adequação às necessidades específicas do projeto, bem como em sua capacidade de integrar-se de maneira eficiente.

2.3.1 Figma

Figma é uma plataforma de design de interface inovadora, amplamente reconhecida por sua capacidade de promover a colaboração em tempo real entre designers e partes interessadas. Sua abordagem baseada em nuvem permite que equipes de design trabalhem juntas de qualquer lugar do mundo, eliminando as barreiras tradicionais de software de design local. O Figma não só facilita o processo de design, mas também integra funcionalidades de prototipagem, permitindo que os designers criem e testem interações de usuário diretamente na plataforma. De acordo com um recente paper (SHARMA; TIWARI, 2021), o Figma está revolucionando a forma como as equipes de design colaboram e trazendo uma nova era de inovação no design de interfaces.

Figura 1 – Ambiente de desenvolvimento do Figma



Fonte: Autor, 2023.

Este trabalho utiliza essa ferramenta devido às suas inúmeras vantagens: é gratuita, acessível via web em qualquer navegador, simples e eficiente, permitindo até mesmo àqueles sem conhecimento avançado em design criarem suas ideias.

2.3.2 Flutter

Flutter é um framework de código aberto criado pelo Google, destinado ao desenvolvimento de interfaces para aplicativos móveis. Diferentemente de muitas outras soluções disponíveis no mercado, o Flutter não apenas facilita o desenvolvimento de aplicações para múltiplas plataformas a partir de um único código-fonte, mas também proporciona um alto desempenho, garantindo uma experiência fluida e consistente para o usuário final (DAGNE, 2019). A sua linguagem principal, o Dart, foi projetada para ser poderosa e intuitiva, permitindo a criação de soluções complexas com um código limpo e de fácil manutenção.

O framework foi escolhido não apenas pela sua capacidade de funcionar em múltiplas plataformas, mas também por oferecer widgets responsivos e visualmente atrativos. Isso facilita o desenvolvimento do aplicativo, aproveitando a modularidade de cada componente ser exibido em diversas telas com apenas um código.

2.3.3 Dart

Dart é uma linguagem de programação desenvolvida pelo Google, lançada em 2011, e desde então tem se destacado por sua eficiência, flexibilidade e desempenho. A linguagem foi projetada para facilidade de uso, oferecendo características modernas que a tornam competitiva no universo de desenvolvimento, como tipagem forte, suporte a programação orientada a objetos e ferramentas dedicadas. Essas características tornam Dart altamente escalável e adequada para o desenvolvimento de aplicações complexas, não limitando-se apenas a aplicativos móveis, mas também a aplicações web e desktop. No contexto do desenvolvimento de aplicativos móveis, Dart assume um papel predominante por ser a linguagem base do framework Flutter utilizado neste trabalho.

2.3.4 Widgets

No universo do Flutter, o termo *Widget* assume um significado central e determinante para a construção de aplicações. Um widget, em sua essência, é uma descrição imutável da parte da interface do usuário, sendo a base fundamental para criar e personalizar a interface gráfica em qualquer aplicativo, sendo a biblioteca *Cupertino*¹ para

¹ Cupertino: conjunto de widgets no Flutter, desenvolvido pela Google, que imita o estilo visual dos componentes nativos do iOS

componentes iOS e *Material Design*² para Android. Estes widgets são organizados em uma árvore hierárquica, onde cada widget herda propriedades do seu pai e pode encapsular uma série de outros widgets. Desta forma, ao programar com Flutter, não está-se pensando apenas em códigos e funções, mas principalmente na composição e interação entre estes widgets, que juntos, formam a experiência completa do usuário no aplicativo.

Além de entender a estrutura hierárquica dos widgets, é crucial compreender o conceito de "estado". O estado refere-se às informações que podem mudar durante a vida útil de um widget, influenciando sua aparência ou comportamento. Existem principalmente dois tipos de widgets no Flutter: *Stateless* e *Stateful*. Enquanto os widgets *Stateless* são imutáveis e não armazenam informações variáveis, os *Stateful* possuem um estado interno que pode ser alterado ao longo do tempo, permitindo que o widget reflita mudanças dinâmicas na interface (ALLAIN, 2020). Ao modificar o estado de um widget *Stateful*, é possível notificar o Flutter da necessidade de reconstrução desse widget específico, garantindo que a interface do usuário seja atualizada de acordo. Esta distinção e capacidade de gerenciamento de estado proporciona ao desenvolvedor o poder de criar interfaces ricas e reativas, tornando o Flutter uma ferramenta versátil para desenvolvimento móvel.

2.3.5 Firebase

O Firebase é uma plataforma de desenvolvimento de aplicativos da Google que oferece uma variedade de ferramentas e serviços em nuvem para ajudar os desenvolvedores a construir, melhorar e crescer seus aplicativos. Lançado inicialmente em 2011 como uma base de dados em tempo real, o Firebase foi desenvolvido para se tornar uma solução completa, oferecendo funcionalidades como hospedagem, autenticação de usuários, notificações push, análise de dados e etc. A integração entre Flutter e Firebase é notavelmente eficiente, pois ambos são produtos da Google, o que permite uma sincronização suave e uma série de benefícios como (CHATTERJEE *et al.*, 2018):

Realtime Database (Banco de dados em tempo real): Bancos de dados que permite o armazenamento e a sincronização de dados entre os usuários em tempo real, o que é essencial para aplicativos que precisam de atualizações instantâneas.

Firebase Authentication (Autenticação): O Firebase Authentication proporciona uma solução fácil e segura para gerenciar a autenticação de usuários, suportando login por email, redes sociais, e até mesmo login anônimo.

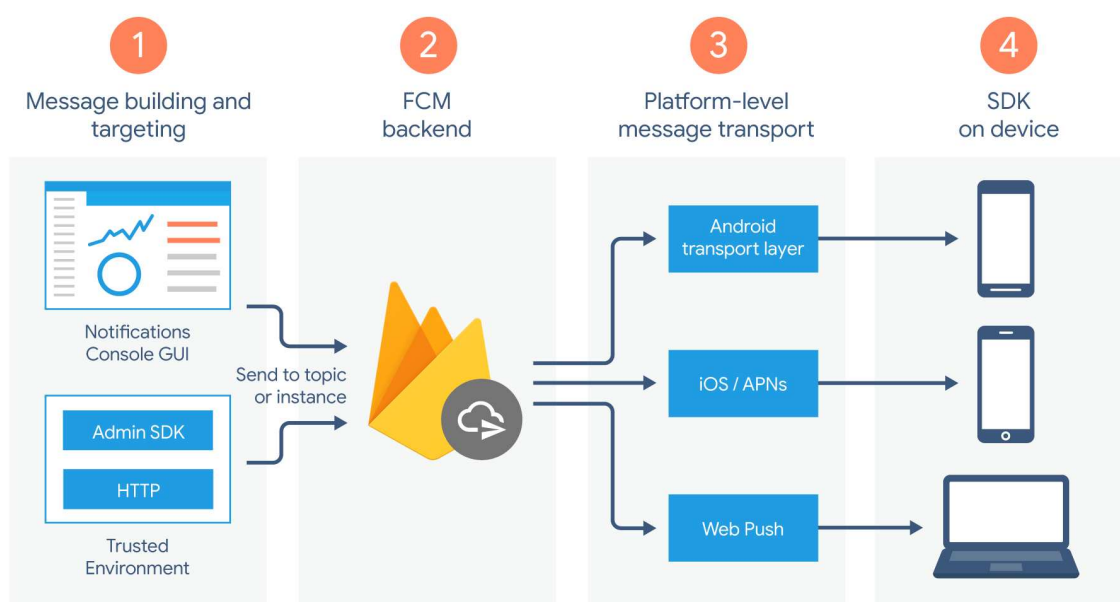
Firebase Cloud Storage (Armazenamento em nuvem): É um serviço de armazenamento de objetos que é utilizado para armazenar e servir conteúdo gerado pelo usuário, como fotos ou vídeos. Ele é construído com escalabilidade automática, se-

² Material Design: linguagem de design livre desenvolvida pela Google com o objetivo de unificar e padronizar as interfaces gráficas do Android

gurança e redes de entrega de conteúdo de alta performance. O Cloud Storage é integrado com o Firebase Authentication para fornecer regras de segurança flexíveis baseadas na identidade dos usuários.

Firebase Cloud Messaging (Mensageria em nuvem): É um serviço de mensagens em nuvem que permite enviar mensagens de forma confiável e gratuita. É possível notificar um cliente que possui um novo e-mail ou outro dado para ser sincronizado, e também enviar mensagens de notificação para direcionar o usuário de volta ao aplicativo.

Figura 2 – Visão geral da arquitetura do FCM



Fonte: Firabase, 2022.

Neste trabalho, o uso do Firebase pode ser justificado não apenas pela eficiência técnica, mas também pela possibilidade de concentrar mais esforços no desenvolvimento de características únicas do aplicativo, em vez de investir tempo na construção e manutenção de infraestrutura de backend. Além disso, o Firebase oferece um plano gratuito que é geralmente suficiente para prototipagem e testes iniciais.

3 ESTADO DA ARTE

3.1 APLICATIVOS CORRELATOS

No desenvolvimento deste capítulo, foi realizado uma análise detalhada de aplicações de mercado existentes. Essa decisão decorre do reconhecimento de que, no campo do desenvolvimento de aplicativos móveis, a literatura acadêmica específica sobre aplicativos de troca de objetos ainda é relativamente limitada. Avaliar aplicativos existentes em termos práticos nos ajuda a entender melhor as tendências atuais, as expectativas dos usuários e as oportunidades de mercado que nosso projeto visa explorar. Essa estratégia é crucial para obter uma compreensão abrangente e atual do estado da arte no campo dos aplicativos móveis.

O primeiro passo, antes de abordar o tema proposto, foi conduzir uma pesquisa de informações *Desk Research*¹ para verificar a existência de outros aplicativos com propostas semelhantes.

Alguns aplicativos e sites apresentam propostas similares à deste trabalho. A maioria deles carece de uma interface amigável, usabilidade adequada para os usuários, as vezes apenas disponível para um SO(Android/IOS) e também requerem o uso de dinheiro real para a aquisição de créditos, que são então utilizados para efetuar as trocas. Além disso, quase todos parecem ter sido descontinuados, ou seja, não recebem mais suporte e atualizações.

Segue abaixo a lista dos aplicativos correlatos encontrados com uma breve análise:

3.1.1 Permutó

A aplicação, somente disponível para navegador, opera com um sistema de pontuação. O usuário registra uma oferta, seja de um serviço ou um objeto, atribuindo-lhe um valor em pontos, onde cada ponto corresponde a 1 Real. Quando alguém adquire sua oferta, você recebe os pontos, os quais podem ser usados para adquirir ofertas de outros usuários. A plataforma tem uma interface amigável e é bem simples de usar (PERMUTO, 2023).

3.1.2 Finpli

O aplicativo, exclusivo para Android e com sua última atualização em 2021, apresenta uma proposta similar ao aplicativo proposto neste trabalho. Oferece recursos como cadastro de objetos, busca inteligente baseada nos interesses do usuário, troca de mensagens durante trocas e uma interface moderna. Contudo, ao tentar utilizá-lo,

¹ Desk Research: pesquisa exploratória através da coleta e uso de informações já publicadas e disponíveis

foi impossível avançar além da tela de cadastro devido à ausência da caixa de seleção para a UF (Unidade Federativa). Avaliações e comentários de outros usuários refletem experiências semelhantes (FINPLI, 2021).

3.1.3 Trokaí

Disponível tanto para Android quanto para iOS, o aplicativo Trokaí oferece uma plataforma para a troca e venda de vestuário entre seus usuários. Neste aplicativo, os usuários podem registrar suas peças de vestuário, optando por definir um preço de venda ou expressar a disposição para trocá-las por outras peças. Contudo, uma limitação notável do aplicativo é a ausência de um filtro por UF, resultando na exibição de peças de todo o Brasil, o que pode dificultar a busca localizada. Constantemente apresenta peças de roupa que não estão mais disponíveis, somente no momento da compra ou troca o usuário é notificado. Até o presente momento ainda recebe atualizações, possui uma interface moderna e simples de usar (TROKAÍ, 2023).

3.1.4 Troca Fácil

Exclusivo para iOS e sem atualizações desde 2019, o aplicativo Troca Fácil é projetado para realizar a troca de uma variedade de objetos entre usuários. No entanto, sua interface é desatualizada. Além disso, enfrenta problemas semelhantes ao aplicativo Finpli, pois o processo de cadastro apresenta falhas, impedindo o uso pleno da plataforma. Durante a tentativa de cadastro, o aplicativo não envia notificações por e-mail nem fornece qualquer informação no próprio aplicativo sobre o progresso ou a conclusão do cadastro (TROCAFÁCIL, 2019).

3.2 COMPARATIVO

A análise comparativa a seguir tem o objetivo de destacar as principais distinções funcionais entre os aplicativos aqui anteriormente citados e o que será desenvolvido neste trabalho.

Embora a utilização integral dos aplicativos Finpli e Troca Fácil não tenha sido viável, as informações apresentadas na tabela abaixo foram compiladas com base nas descrições fornecidas pelos desenvolvedores desses aplicativos nos respectivos *marketplaces*².

² Marketplace: site desenvolvido para disponibilização de aplicações ou produtos

Tabela 1 – Comparativo dos aplicativos

FUNCIONALIDADES	PERMUTO	FINPLI	TROKAÍ	TROCA FÁCIL	APLICATIVO PROPOSTO
Painel de pesquisa	X		X		X
Filtro de pesquisa completo	X			X	X
Troca de mensagens	X	X	X	X	X
Mecanismo de notificações	X		X	X	X
Troca de objetos	X	X	X	X	X
Cruzamento de trocas					X
Avaliação entre usuários	X	X	X		X
Doação					X
Área de favoritos		X	X	X	X
Central de ajuda	X		X		X
Disponível para Android		X	X		X
Disponível para iOS			X	X	X

Fonte: Autor, 2023.

- **Painel de pesquisa:** Espaço que permite realizar buscas específicas dentro do aplicativo, utilizando palavras-chave.
- **Filtro de pesquisa completo:** Ferramenta que permite a utilização de múltiplos filtros simultaneamente para refinar os resultados.
- **Troca de mensagens:** Sistema de comunicação interno que possibilita o envio e recebimento de mensagens entre os usuários.
- **Mecanismo de notificações:** Serviço que alerta os usuários sobre novas mensagens e sugestões de trocas.
- **Troca de objetos:** Funcionalidade que permite aos usuários cadastrarem objetos para troca dentro do ambiente do aplicativo.
- **Cruzamento de trocas:** Algoritmo que identifica e sugere trocas baseando-se nos interesses cadastrados dos usuários, facilitando transações.
- **Avaliação entre usuários:** Sistema de avaliação que permite aos usuários classificar uns aos outros com base na confiabilidade e na satisfação com as trocas realizadas.
- **Doação:** Funcionalidade que permite aos usuários cadastrar itens para doação no aplicativo.
- **Favoritos:** Opção que permite aos usuários marcarem anúncios como favoritos para fácil acesso posterior.
- **Central de ajuda:** Seção de suporte ao usuário com informações úteis, FAQs e contato direto para resolução de dúvidas e problemas.

- **Disponível para Android/iOS:** Indica a compatibilidade do aplicativo com os sistemas operacionais Android ou iOS.

3.3 RISCOS

Esta seção tem como objetivo discutir os riscos associados ao uso de aplicativos como os mencionados, explorar as soluções atualmente oferecidas e apresentar a solução proposta por este trabalho.

3.3.1 Identificação dos Riscos

Os riscos observados ao estudar os aplicativos anteriormente apresentados foram os seguintes:

- **Objetos falsificados:** Qualidade inferior ou condição dos itens trocados que não correspondem à descrição fornecida.
- **Roubo:** Alguma das partes roubar o objeto da outra ao se encontrarem para realizar a troca.
- **Violação de Privacidade:** Exposição de dados pessoais dos usuários, o que pode incluir informações de contato, endereço, entre outros.
- **Incompatibilidade de expectativas:** Desentendimentos ou expectativas não atendidas entre as partes, levando a conflitos ou disputas após a troca.
- **Danos durante o transporte:** Danos aos objetos durante o transporte.
- **Responsabilidade legal:** Riscos associados a possíveis questões legais decorrentes da troca de itens protegidos por direitos autorais, patentes, objetos ilícitos ou outras restrições legais.
- **Manipulação de avaliações:** Manipulação ou falsificação de avaliações de usuários, o que pode afetar a confiança no aplicativo.
- **Registro da marca:** Cópia do aplicativo proposto e de toda propriedade intelectual.

3.3.2 Soluções existentes

A solução predominante identificada na maioria dos aplicativos, abrangendo tanto os de troca quanto os de compra de produtos usados, envolve a elaboração de uma política de uso do aplicativo. Esta política cobre aspectos como a violação da privacidade dos dados dos usuários, responsabilidades legais e a isenção de responsabilidade por roubos decorrentes do uso do aplicativo. Para acessar o aplicativo, os usuários devem aceitar os termos desta política de privacidade e uso, a recusa em aceitá-los resulta na impossibilidade de acesso e uso da aplicação. Adicionalmente,

alguns aplicativos oferecem diretrizes de segurança e dicas práticas para auxiliar os usuários a realizarem transações de forma segura.

Muitos aplicativos também introduzem o processo de verificação de identidade, solicitando documentos legais, números de telefone e até mesmo foto do usuário no momento do cadastro.

Para filtrar conteúdos ilícitos, alguns aplicativos implementam sistemas de moderação e filtragem, os quais revisam os itens antes de sua publicação. Esses mecanismos podem ser implementados por meio de IAs, que realizam o reconhecimento de imagem e texto, ou por administradores, que conduzem a verificação manualmente.

3.3.3 Solução proposta

Diante das soluções existentes mapeadas, a proposta deste trabalho busca integrá-las e aprimorá-las, oferecendo um aplicativo de trocas mais seguro e eficiente. A solução proposta consiste em uma abordagem combinando tecnologia, governança e educação do usuário para mitigar os riscos identificados.

A criação de uma política de privacidade e uso para oferecer clareza e transparência, incluindo detalhes específicos sobre como os dados dos usuários são protegidos e utilizados. A política também abordará a responsabilidade legal de forma mais detalhada, garantindo que os usuários estejam cientes de suas obrigações e direitos.

Além da verificação básica de documentos, o aplicativo pode incorporar reconhecimento facial, para fortalecer a autenticação do usuário e prevenir fraudes. Utilizando algoritmos de IA, é possível realizar uma filtragem mais eficiente de conteúdos ilícitos. A IA pode ser treinada para identificar não apenas itens proibidos, mas também itens potencialmente falsificados ou descrições enganosas.

O aplicativo pode oferecer recursos educacionais, como tutoriais, para educar os usuários sobre práticas seguras de troca, além de alertá-los sobre os riscos e como evitá-los.

3.4 LEVANTAMENTO DE REQUISITOS

O levantamento de requisitos foi uma etapa fundamental no processo de desenvolvimento do aplicativo, pois envolveu a identificação, análise e documentação das necessidades e expectativas em relação ao sistema. Esse processo buscou entender de forma detalhada quais funcionalidades o sistema deveria oferecer, como ele deveria se comportar e quais restrições deveriam ser observadas durante o desenvolvimento.

Existem dois tipos de requisitos, os funcionais e não-funcionais. Os requisitos funcionais definem as funcionalidades específicas do sistema e como ele deve se comportar, enquanto os requisitos não funcionais descrevem as qualidades e restrições do sistema, como desempenho, segurança e usabilidade.

3.4.1 Requisitos Não-Funcionais

- RNF1: Interface amigável e intuitiva;
- RNF2: Navegação fácil e rápida entre as funcionalidades do aplicativo;
- RNF3: Resposta rápida às ações dos usuários;
- RNF4: Alta disponibilidade e tempo de resposta eficiente do servidor;
- RNF5: Proteção dos dados pessoais dos usuários;
- RNF6: Criação de políticas de privacidade e uso claras e detalhadas;
- RNF7: Disponível para os sistemas operacionais Android e iOS;
- RNF8: Funcionar em diversas versões dos SOs suportados.
- RNF9: Infraestrutura que permita a adição de novas funcionalidades sem comprometer o desempenho.

3.4.2 Requisitos Funcionais

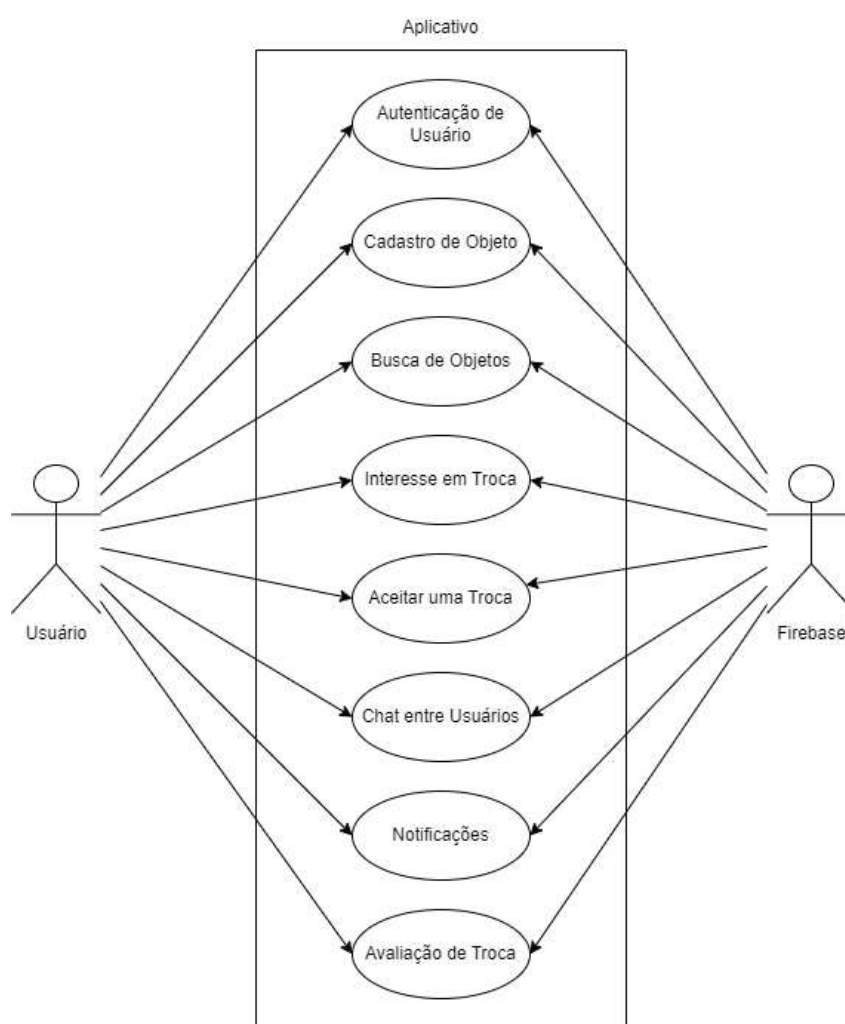
- RF1: Permitir que o usuário crie uma conta e se autentiquem através de Facebook ou Google;
- RF2: Permitir que o usuário cadastre um objeto para troca, incluindo categoria do seu objeto, categoria de objetos de interesse, descrição e fotos;
- RF3: Permitir que o usuário altere informações do objeto já cadastrados;
- RF4: Permitir que o usuário busque objetos de interesse mesmo que não tenha cadastrado um objeto;
- RF5: Exibir uma galeria de objetos disponíveis para troca;
- RF6: Implementar um filtro de pesquisa completo para refinar os resultados de busca;
- RF7: Cruzar informações dos objetos cadastrados para encontrar coincidências de interesse entre usuários;
- RF8: Indicar objetos de outros usuários que possuem interesse no objeto que ele possui;
- RF9: Criar um chat para comunicação entre usuários interessados em realizar a troca;
- RF10: Notificar os usuários sobre novas mensagens e sugestões de trocas;
- RF11: Permitir que os usuários avaliem uns aos outros após a realização de uma troca;
- RF12: Solicitar reconhecimento facial do usuário para a validação da identidade do usuário;

- RF13: Verificar objetos ilegais através de algoritmos de IA;
- RF14: Implementar um tutorial para os usuários no primeiro uso do aplicativo;

3.5 CASOS DE USO

Os casos de uso são descrições detalhadas de como um usuário interage com um sistema para alcançar um objetivo específico.

Figura 3 – Diagrama de casos de uso



Fonte: Autor

- UC1: Autenticação de Usuário: O usuário acessa a tela de login e seleciona um dos botões para se autenticar via Google ou Facebook. O Firebase valida o token do usuário com o serviço de autenticação.
- UC2: Cadastro de Objeto: O usuário acessa a tela de cadastro de objeto, insere as informações do objeto e seleciona o botão de salvar. O Firebase salva os dados e disponibiliza o objeto para troca na galeria de objetos.

- UC3: Busca de Objetos: O usuário acessa a tela de busca, insere os critérios de busca e aplica filtros. O Firebase retorna ao aplicativo os objetos que correspondem aos critérios.
- UC4: Interesse em Troca: O usuário visualiza um objeto de interesse e sinaliza interesse em trocar o objeto. O Firebase salva a notificação e o aplicativo notifica o dono do objeto sobre o interesse.
- UC5: Aceitar uma Troca: O usuário acessa a tela de ofertas recebidas e aceita a troca. O Firebase salva a notificação e o aplicativo notifica o usuário que enviou a oferta, que ela foi aceita.
- UC6: Chat entre Usuários: O usuário acessa o chat de alguma de suas trocas. O aplicativo permite o envio e recebimento de mensagens. O Firebase salva e consulta todas novas mensagens enviadas pelo usuário.
- UC7: Notificações: O Firebase retorna todas as notificações e o aplicativo mostra em tela todas as notificações sobre novas mensagens, interesse em troca e ofertas aceitas. O usuário visualiza as notificações e toma as ações necessárias. O Firebase atualiza que aquelas notificações já foram visualizadas.
- UC8: Avaliação de Troca: Após a troca, o usuário avalia o outro usuário. O Firebase salva a avaliação e o aplicativo atualiza a pontuação do usuário avaliado.

4 DESENVOLVIMENTO

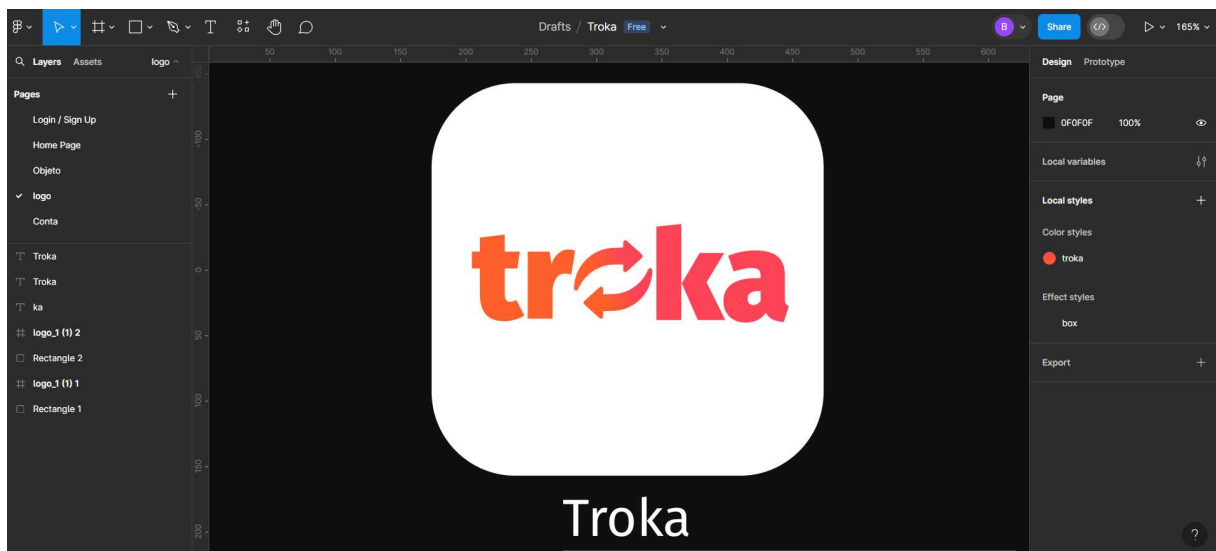
Neste capítulo, são detalhadas as etapas de desenvolvimento do aplicativo "Troka", desde a concepção da arquitetura até a implementação final. Cada parte do sistema é explicada em detalhes, destacando as tecnologias utilizadas e os desafios enfrentados durante o processo.

4.1 DESIGN DA INTERFACE

O design inicial do aplicativo foi elaborado utilizando a ferramenta Figma, que permitiu criar um esboço detalhado da logo e das telas principais do aplicativo. A identidade visual foi inspirada em diversos aplicativos renomados, como Airbnb, WhatsApp, OLX, Mercado Livre, entre outros, para garantir uma interface moderna e intuitiva.

O nome "Troka" foi escolhido devido à funcionalidade principal do aplicativo, que é a troca de itens, sendo um termo fácil de pesquisar e pronunciar. A logo inserida no ícone do aplicativo (Figura 5) foi desenvolvida com o próprio nome e um símbolo de troca/substituição fazendo analogia a letra O de Troka. Além disso, foi utilizado a fonte "Fira Sans" e um gradiente entre duas cores, proporcionando uma aparência atraente e memorável.

Figura 4 – Ícone do aplicativo - Figma

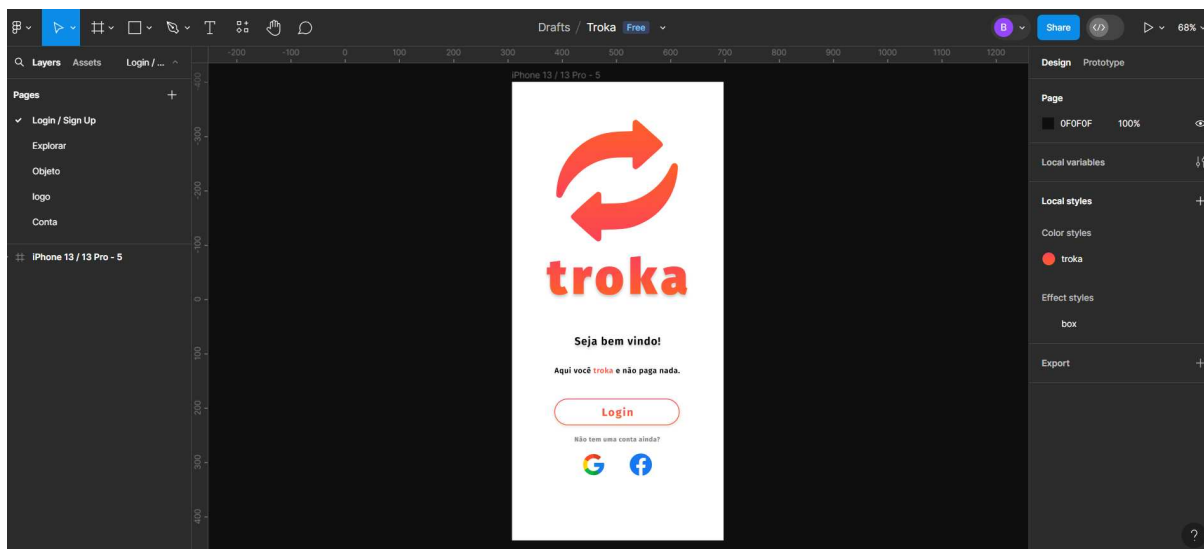


Fonte: Autor

Além da logo, outras quatro telas principais foram projetadas no Figma antes de iniciar o desenvolvimento do código. Essas telas incluem: a tela de Login (Figura 6), a tela de Explorar (Figura 7), a tela de Objeto (Figura 8) e a tela de Conta (Figura 9). As

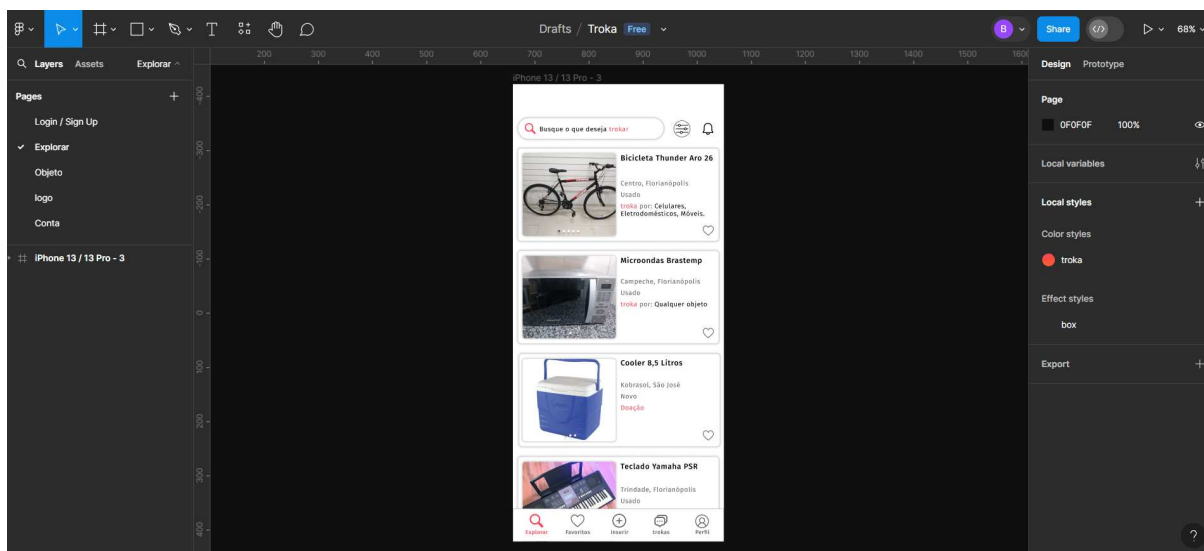
demais telas não foram detalhadas, pois o foco principal deste trabalho não é o design do aplicativo, mas sim sua funcionalidade e desenvolvimento.

Figura 5 – Tela de Login - Figma



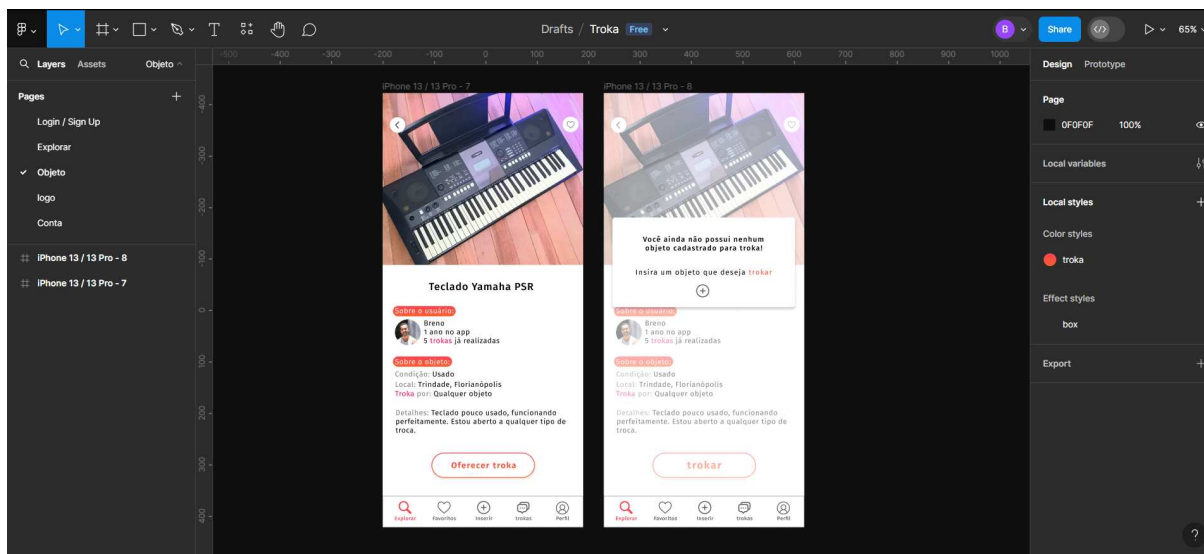
Fonte: Autor

Figura 6 – Tela de Explorar - Figma



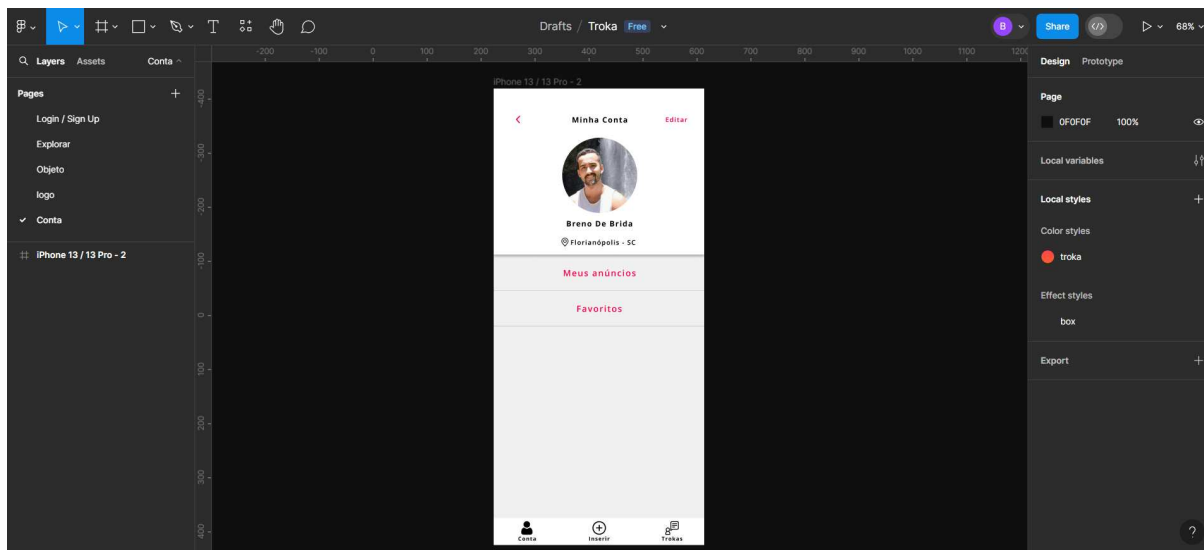
Fonte: Autor

Figura 7 – Tela do Objeto - Figma



Fonte: Autor

Figura 8 – Tela da Conta - Figma



Fonte: Autor

Com as telas previamente projetadas, evitou-se perder tempo criando código apenas para testar a aparência das interfaces, dessa forma o desenvolvimento foi mais ágil e eficiente. No entanto, apesar das telas iniciais estarem definidas, alguns designs foram ajustados durante o desenvolvimento frontend e serão apresentados nas seções subsequentes.

4.2 DIAGRAMA DE ARQUITETURA

A arquitetura escolhida para o desenvolvimento do aplicativo foi a MVVM (Model-View-ViewModel), que organiza o código em três camadas: Model, que gerencia os dados e a lógica de negócios; View, que é responsável pela interface do usuário; e ViewModel, que atua como intermediário, gerenciando o estado e a lógica da interface, facilitando a manutenção e a escalabilidade do aplicativo. A seguir, foram listados e colocados em um diagrama (Figura 4) todos os componentes desta arquitetura.

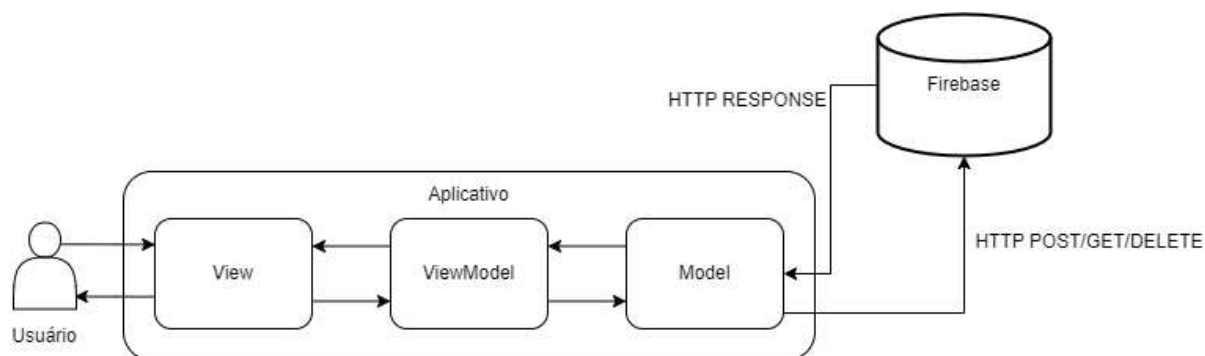
No contexto da arquitetura MVVM, o aplicativo é o componente responsável principalmente por duas partes:

1. **View:** Esta camada é responsável por definir a aparência e o layout do aplicativo. No Flutter, as Views são representadas por Widgets, que descrevem a interface visual do aplicativo. As Views no Flutter são reativas, ou seja, elas se atualizam automaticamente quando o estado dos dados muda.
2. **ViewModel:** Esta camada serve como um intermediário entre a View e o Model. Ela contém a lógica de apresentação e estado da interface, além de manipular as interações do usuário e os dados exibidos. A ViewModel é responsável por fornecer dados formatados à View e tratar eventos de interação do usuário.

Além do aplicativo, outro componente importante dessa arquitetura é o Firebase que se encaixa principalmente na camada Model, servindo como base de dados na nuvem, mas pode também suportar a camada ViewModel:

1. **Model:** A Model é responsável por buscar, salvar e manipular dados dos usuários, e fornecer esses dados à ViewModel.
2. **ViewModel:** Além de gerenciar a lógica de apresentação, a ViewModel também pode utilizar os serviços do Firebase para obter e atualizar os dados necessários. Por exemplo, ela pode chamar métodos de Firebase Firestore para recuperar dados do banco de dados e processar esses dados antes de enviá-los à View para exibição.

Figura 9 – Diagrama de Arquitetura



Fonte: Autor

4.3 CONFIGURAÇÃO DO AMBIENTE PARA O DESENVOLVIMENTO

Para iniciar o desenvolvimento do aplicativo, foi necessária a configuração de um ambiente de desenvolvimento adequado. O SO do ambiente configurado foi o Windows 11. A seguir, detalho cada passo realizado para configurar este ambiente:

1. Download e Instalação do Flutter SDK: Kit de desenvolvimento do flutter;
2. Instalação do VSCode e configuração de extensões: Ferramenta de edição de código;
3. Instalação do Android Studio: Software de desenvolvimento integrado para Android;
4. Configuração do AVD: Emulador de dispositivos virtuais Android;
5. Configuração de Depuração via USB: Modo de comunicação entre dispositivos Android e computadores.

4.3.1 Vantagens da Depuração Via USB

A depuração via USB é significativamente mais rápida que o uso de emuladores. Isso se deve ao fato de que os emuladores simulam um dispositivo completo no computador, o que consome muitos recursos do sistema, como CPU, GPU, RAM e armazenamento. Em contrapartida, a depuração via USB utiliza diretamente o hardware do dispositivo Android, eliminando a sobrecarga no sistema do desenvolvedor e proporcionando uma experiência de desenvolvimento mais ágil.

Conectar um dispositivo Android real via USB permite que o desenvolvedor utilize o hardware nativo do dispositivo, incluindo a CPU, GPU, sensores e outros componentes. Isso proporciona uma experiência de teste mais precisa e realista, refletindo melhor o comportamento do aplicativo em produção.

A precisão dos testes em hardware nativo é crítica para detectar problemas específicos de dispositivos, como variações no desempenho de gráficos, diferenças na gestão de energia e peculiaridades na resposta ao toque, que podem não ser evidentes em emuladores.

Figura 10 – Depuração via USB



Fonte: Autor

4.3.2 Estrutura de Pastas e Organização do Código

O código foi organizado em uma estrutura de pastas que facilita a manutenção e a escalabilidade do projeto:

- `.dart_tool/`: Diretório gerenciado pelo Flutter/Dart que armazena ferramentas e artefatos de build.
- `android/`: Contém os arquivos específicos para a construção do aplicativo no Android.
- `assets/`: Contém arquivos estáticos, como imagens e fontes.
- `firebase/`: Contém configurações e arquivos necessários para a integração com o Firebase.
- `ios/`: Contém os arquivos específicos para a construção do aplicativo no iOS.
- `lib/`: Diretório principal onde o código Dart do aplicativo está localizado.

4.4 DESENVOLVIMENTO DO APLICATIVO

O desenvolvimento do aplicativo iniciou-se pelo front-end, priorizando a criação da interface gráfica sem, inicialmente, abordar a parte funcional. O objetivo foi implementar o design de todas as telas, conforme projetado no Figma, com foco na experiência do usuário. Com a interface concluída, procedeu-se ao desenvolvimento do

back-end, responsável por implementar todas as funções customizadas necessárias para a integração entre as telas, autenticação com o banco de dados e comunicação com APIs, tornando o aplicativo plenamente funcional.

Nas próximas subseções, será detalhado o desenvolvimento das telas e das funcionalidades principais do aplicativo.

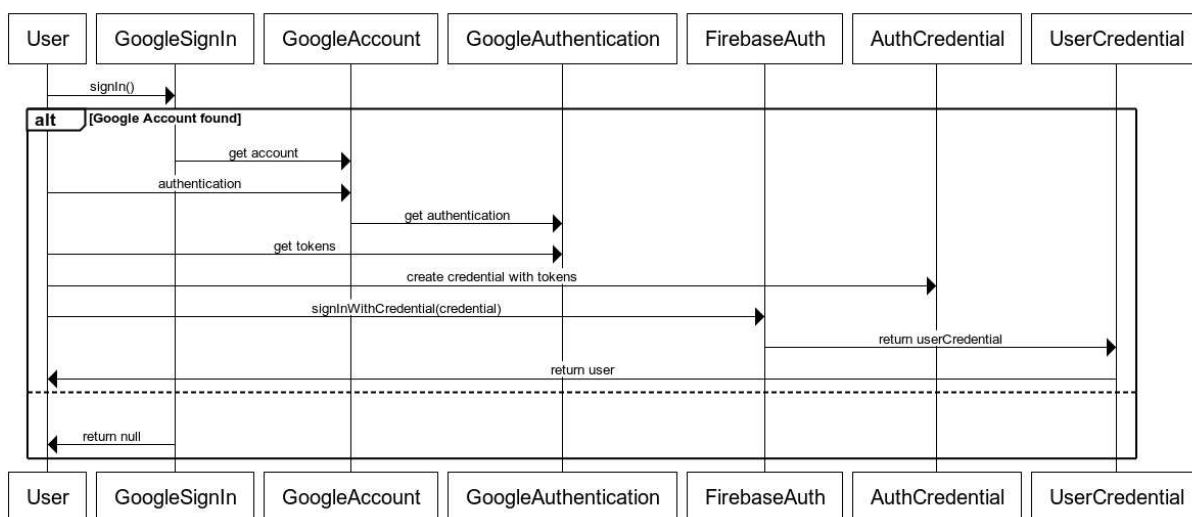
4.4.1 Login/Cadastro e Autenticação

O desenvolvimento da tela de login e cadastro do aplicativo foi realizado com o objetivo de proporcionar uma experiência de usuário simplificada e segura. A integração com o Firebase permitiu a autenticação via Google e Facebook, duas plataformas comumente usadas por usuários de smartphones.

A tela de login foi construída com a logo do aplicativo e dois botões principais, um para login via Google e outro para Facebook. A lógica de autenticação foi implementada no arquivo `login_model.dart`, utilizando os pacotes `firebase_auth`, `flutter_facebook_auth` e `google_sign_in`.

A função `"signInWithGoogle"` inicia o processo de login via Google, obtém as credenciais de autenticação e autentica o usuário no Firebase:

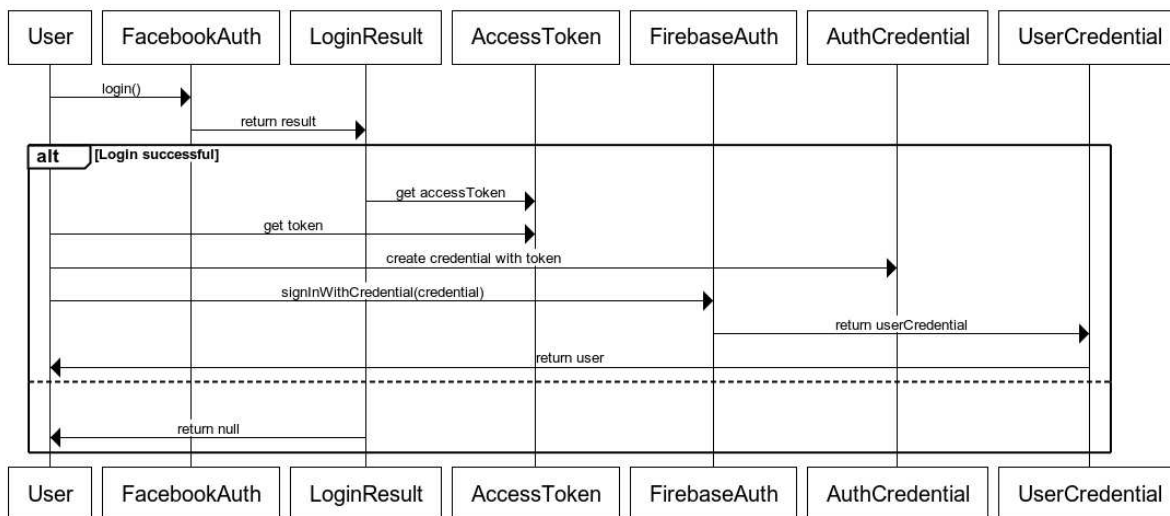
Figura 11 – Diagrama de Sequência - Login via Google



Fonte: Autor

De maneira similar, a função `signInWithFacebook` gerencia o login via Facebook:

Figura 12 – Diagrama de Sequência - Login via Facebook



Fonte: Autor

Ambas as redes sociais exigem que o usuário possua um cadastro prévio. Caso contrário, ele deve realizar o cadastro para poder se autenticar e acessar o aplicativo.

Figura 13 – Tela de Login



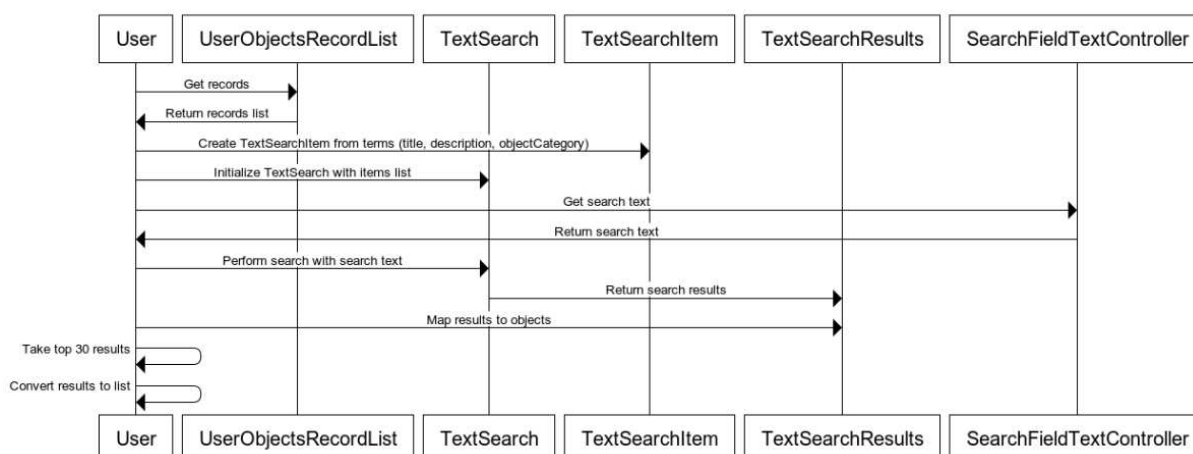
Fonte: Autor

4.4.2 Consulta de Objetos e Filtro

A consulta de objetos e o filtro estão dispostos na primeira página do aplicativo, na tela de explorar, que é exibida assim que o usuário realiza sua autenticação. Nesta tela, inicialmente, são exibidos todos os objetos disponíveis na cidade onde o usuário se encontra. Caso o usuário deseje refinar sua busca, ele pode utilizar o campo de pesquisa, localizado no topo da página, para inserir palavras-chave específicas. A caixa de pesquisa é um campo de texto configurado para atualizar os resultados conforme o usuário digita.

Para realizar a consulta no banco de dados e retornar os registros, é utilizada a função TextSearch, detalhada a seguir:

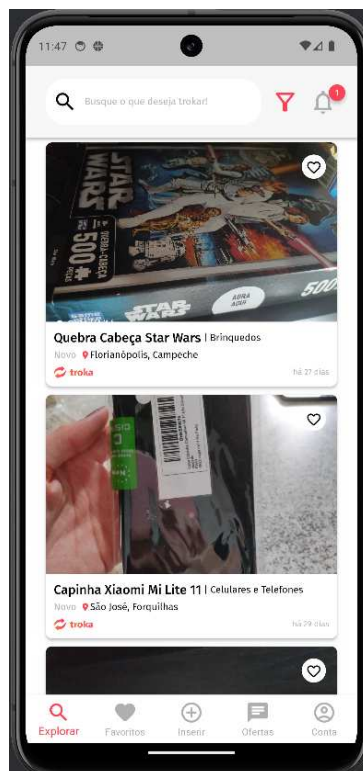
Figura 14 – Diagrama de Sequência - Busca de objetos



Fonte: Autor

Esta sequência garante que os termos de busca do usuário sejam comparados com os atributos relevantes dos objetos, retornando uma lista limitada aos 30 primeiros resultados correspondentes.

Figura 15 – Tela de Explorar



Fonte: Autor

Além do campo de pesquisa, o usuário pode utilizar a tela de filtro criada para refinar os resultados. Essa funcionalidade permite realizar o filtro da localização do objeto (estado e cidade), a finalidade (troca ou doação), a condição do objeto (novo, usado, recondicionado ou com defeito), a categoria do item e as categorias de objetos que são aceitos em troca. Com essas opções de filtragem, os usuários podem encontrar exatamente o que procuram de maneira mais eficiente e precisa.

Para a realização do filtro, foi utilizado a estratégia de armazenar todos os valores em variáveis de estado global. Dessa forma, quando o usuário confirma o filtro no botão "Filtrar" ao final da tela de filtro, todas as escolhas feitas são registradas e usadas para realizar a pesquisa no Firebase.

Além disso foi utilizado a função "Future.wait", que é uma excelente solução para executar múltiplas tarefas assíncronas em paralelo. Essa abordagem é especialmente útil quando é necessário aguardar a conclusão de diversas operações antes de continuar a execução do código. O "Future.wait" permite que tarefas que podem demorar, como atualização de dados utilizados em tempo real, sejam realizadas simultaneamente, sem bloquear a execução das outras partes do programa.

Cada Future é responsável por uma parte diferente do filtro:

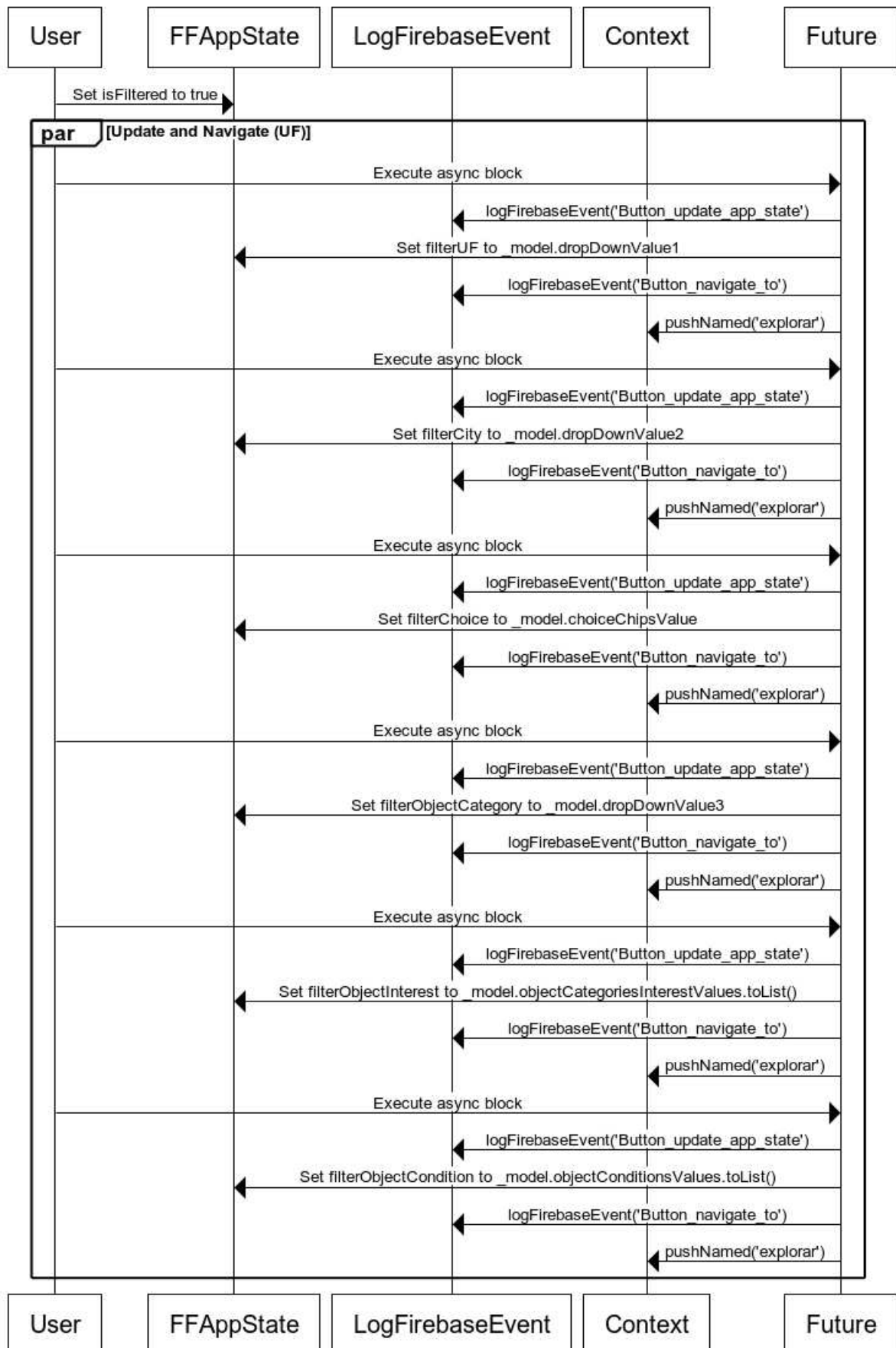
- Primeiro Future: Atualiza a variável de estado com a unidade federativa (UF)

selecionada.

- Segundo Future: Atualiza a variável de estado com a cidade selecionada.
- Terceiro Future: Atualiza a variável de estado com o tipo de negociação (troca ou doação).
- Quarto Future: Atualiza a variável de estado com a categoria do objeto.
- Quinto Future: Atualiza a variável de estado com os interesses do objeto.
- Sexto Future: Atualiza a variável de estado com a condição do objeto.

Essa estrutura garante que todos os filtros sejam aplicados corretamente e que o usuário seja redirecionado para a página de exploração após a aplicação dos filtros. A utilização de "Future.wait" permite que todas as atualizações de estado e navegações ocorram simultaneamente, otimizando a eficiência do aplicativo.

Figura 16 – Diagrama de Sequência - Filtros



Outro aspecto crucial considerado durante o desenvolvimento foi a utilização da função "logFirebaseEvent". Essa função foi implementada para registrar eventos importantes no Firebase Analytics, permitindo um monitoramento mais detalhado do comportamento dos usuários e do desempenho do aplicativo. Com a "logFirebaseEvent", é possível capturar dados específicos sobre as interações dos usuários, como a realização de buscas, a aplicação de filtros, e outras ações relevantes.

Essa abordagem não apenas facilita a análise do uso do aplicativo, mas também contribui para a identificação de possíveis melhorias, auxiliando na tomada de decisões informadas para futuras atualizações e otimizações. Assim, o "logFirebaseEvent" se torna uma ferramenta essencial para manter a qualidade e a eficiência do aplicativo.

Figura 17 – Tela de Filtro



Fonte: Autor

4.4.3 Inserção de Objetos

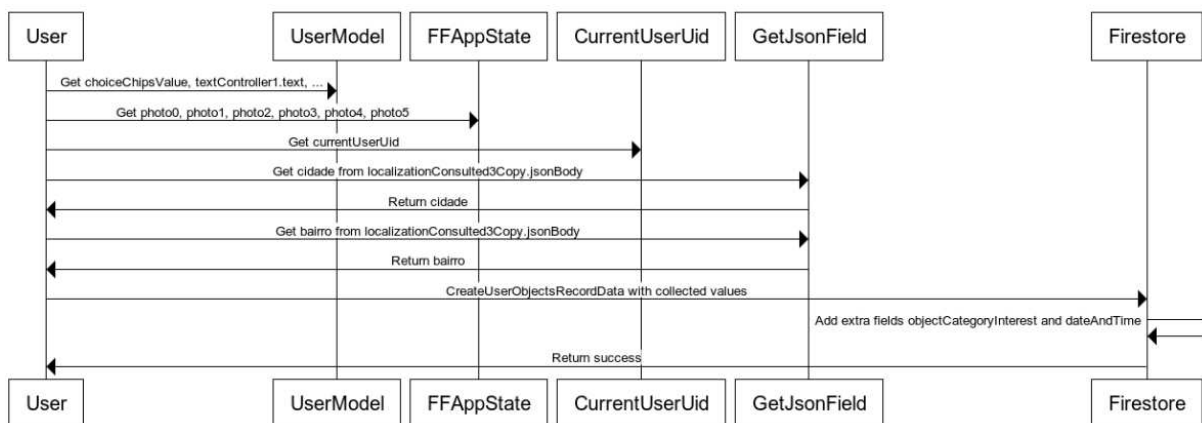
A inserção de objetos foi elaborada de forma que não fosse extensa e contivesse somente os campos necessários para a inclusão do item. A inserção é realizada através de uma série de funções que interagem com o Firebase Firestore e Firebase Storage para armazenar dados e imagens.

Dentro do Firestore Database, foi criada uma coleção chamada "user_objects", onde são armazenados os objetos de todos os usuários. Cada "documento", que é

a forma como o Firestore Database denomina os registros individuais de cada item, contém os atributos fornecidos pelo usuário no momento da inserção dos dados do objeto.

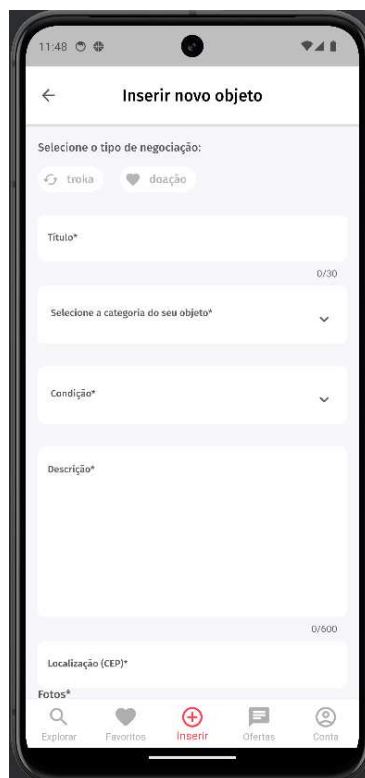
Para inserir um objeto no Firestore, é utilizado a função apresentada a seguir:

Figura 18 – Diagrama de Sequência - Cadastro de objeto



Fonte: Autor

Figura 19 – Tela de Cadastro de Objeto



Fonte: Autor

Após inserir um objeto, o usuário pode visualizá-lo na seção "Meus objetos" dentro da tela "Conta".

1. Seleção de Objeto para Troca: Se o usuário possui objetos nas categorias aceitas, ele pode selecionar qual objeto deseja oferecer em troca.
2. Convite para Adicionar Novo Objeto: Se o usuário não possui objetos nas categorias aceitas, ele é informado disso e convidado a adicionar um novo objeto que se enquadre nas categorias aceitas.

Figura 20 – Tela do Objeto

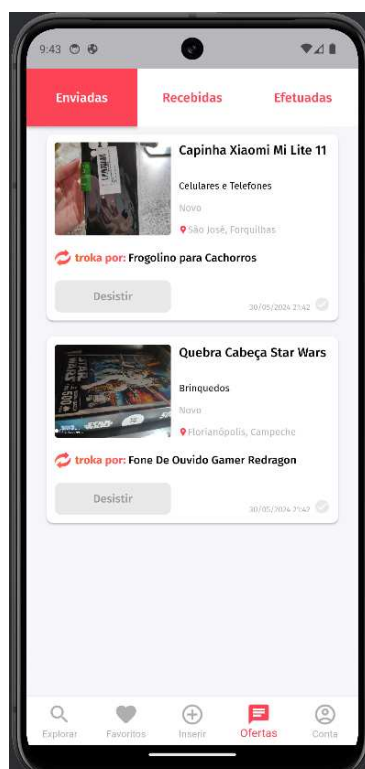


Fonte: Autor

4.4.4 Oferta de Objetos

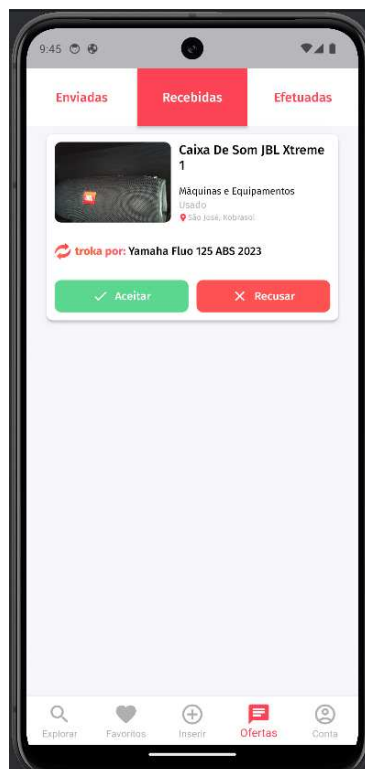
Dentro do aplicativo Troka, as ofertas são a principal forma de negociação entre os usuários. Para que um usuário realize uma troca bem-sucedida, ele deve primeiro enviar uma oferta a partir da página do objeto desejado. Após enviar a oferta, ela aparecerá na aba "Enviadas" da tela "Ofertas". O usuário que recebe a oferta verá a mesma na aba "Recebidas". Uma vez que a oferta é aceita, ela é movida para a aba "Efetuadas", onde se torna um chat. Este chat permite que os usuários troquem mensagens e combinem os detalhes de como realizar a troca de forma prática.

Figura 21 – Tela de Ofertas - Aba "Enviadas"



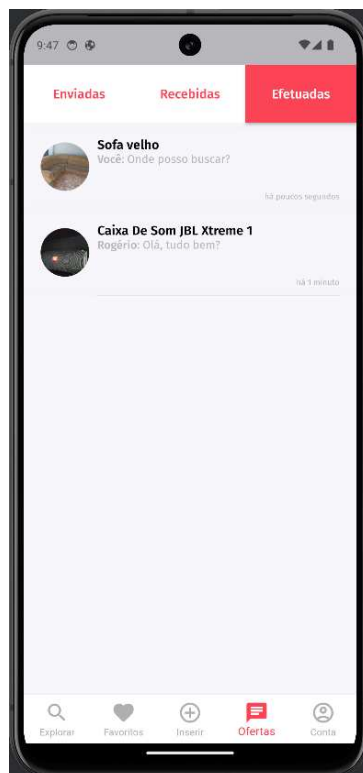
Fonte: Autor

Figura 22 – Tela de Ofertas - Aba "Recebidas"



Fonte: Autor

Figura 23 – Tela de Ofertas - Aba "Efetuadas"



Fonte: Autor

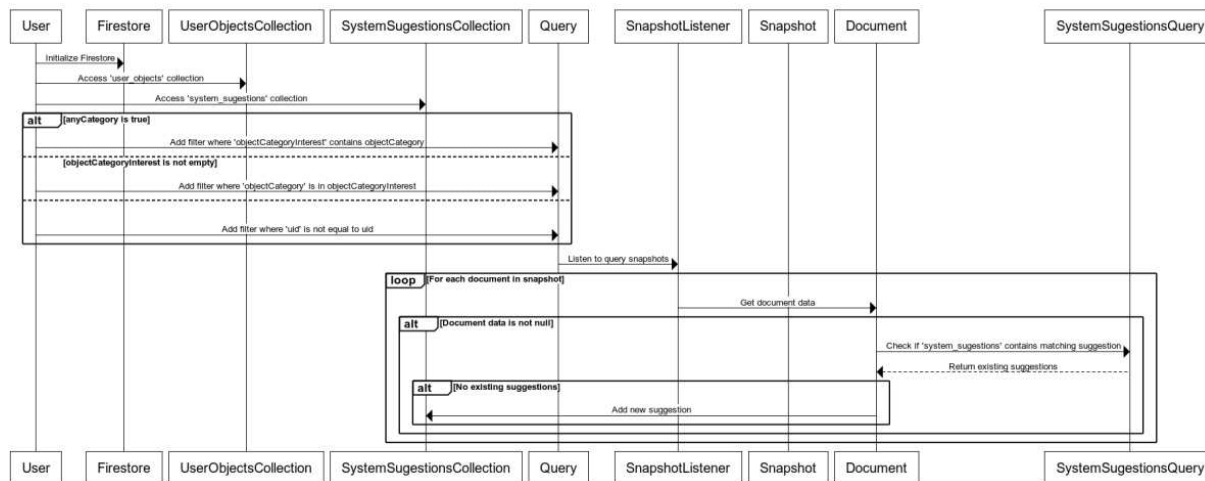
4.4.5 Algoritmo de Sugestões de Troca

Na tela de cada objeto que o usuário possui, há um botão no canto inferior direito chamado "Sugestões de Troca". Esse botão busca no banco de dados todos os objetos que pertencem à categoria do objeto que o usuário aceita em troca, além de verificar a reciprocidade. Assim, apenas serão exibidos objetos cujos proprietários também estejam dispostos a aceitar o objeto do usuário em troca. Dessa forma, as sugestões apresentadas são altamente relevantes, aumentando a probabilidade de uma troca bem-sucedida.

A função responsável por isso é a "findSugestion", ela implementa um algoritmo de filtragem e recomendação baseado em consultas de banco de dados. A abordagem teórica subjacente pode ser descrita como um algoritmo de recomendação colaborativa com filtragem baseada em conteúdo. A Recomendação Colaborativa sugere itens com base nas preferências de outros usuários que possuem interesses similares. No contexto da função "findSugestion", a recomendação colaborativa ocorre implicitamente, pois o sistema sugere objetos de outros usuários que podem ser do interesse do usuário atual. A Filtragem Baseada em Conteúdo sugere itens com base nas características dos próprios itens e nas preferências do usuário. A função filtra objetos com base na categoria "objectCategory" e nos interesses do usuário

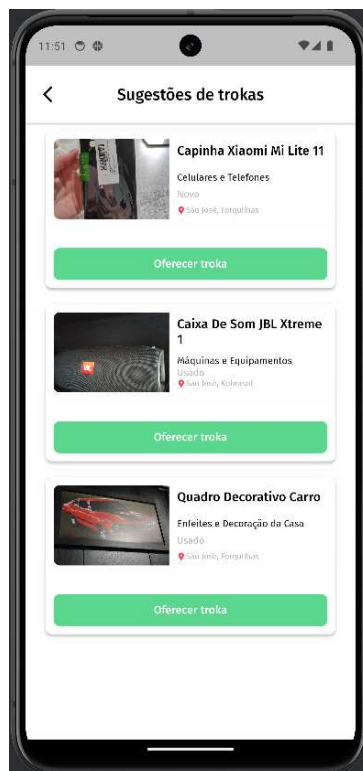
"objectCategoryInterest".

Figura 24 – Diagrama de Sequência - Sugestões de Troca



Fonte: Autor

Figura 25 – Tela de Sugestões de Trocas



Fonte: Autor

4.4.6 Chat

O chat foi a última tela desenvolvida no aplicativo, pois todas as demais funcionalidades precisavam estar operacionais para que ele pudesse ser testado. A troca de mensagens é simples: o usuário pode enviar textos ou fotos. Na tela do chat, é possível visualizar o objeto envolvido na troca, além de ter a opção de cancelar a negociação, desfazendo a troca e removendo o objeto da aba "Ofertas Efetuadas".

Para que cada usuário possa visualizar as mensagens de outros usuários, foi criada uma coleção no Firestore Database chamada "chatMessages". Cada documento dessa coleção representa uma mensagem enviada.

Assim, cada vez que um usuário digita uma mensagem ou envia uma foto, o sistema cria um documento nesta coleção. A tela do chat atualiza em tempo real, filtrando apenas os documentos que possuem o "offerId" da troca específica e ordenando-os pela data e hora de envio. Isso garante que as mensagens sejam exibidas na ordem correta e que cada chat reflita apenas as conversas relacionadas a uma determinada oferta.

Figura 26 – Tela do Chat



Fonte: Autor

4.4.7 Integração de Serviços Externos

Foram utilizadas três APIs no aplicativo para obter dados relacionados à localização e endereço do usuário:

1. API da Geocoding: Esta API foi fundamental para obter a cidade do usuário a partir da latitude e longitude, que o aplicativo coleta do sistema com a permissão do usuário no momento do login (GEOCODING, 2024).
2. API do IBGE: Utilizada para buscar todas as cidades de uma determinada Unidade Federativa (UF) selecionada pelo usuário ao realizar um filtro por estado (IBGE, 2024).
3. API da ViaCEP: Utilizada para obter a cidade e o bairro a partir do CEP do usuário. Essas informações são usadas no cadastro do objeto (VIACEP, 2024).

4.5 TESTES E VALIDAÇÃO

Os testes e a validação de software são etapas cruciais no ciclo de desenvolvimento, garantindo que o produto final esteja alinhado com os requisitos funcionais e não funcionais especificados.

Durante o desenvolvimento do aplicativo, foram realizados testes contínuos e incrementais a cada nova implementação de componente ou tela. Essa abordagem está em conformidade com a metodologia de testes ágeis, que enfatiza a integração contínua e testes frequentes para identificar e corrigir problemas o mais cedo possível (BECK, 2000). Os testes foram conduzidos em um dispositivo Android, utilizando a depuração via USB para garantir a máxima cobertura de cenários possíveis.

Ao finalizar o desenvolvimento, o aplicativo foi disponibilizado para um grupo seleto de usuários que possuíam smartphones com o sistema operacional Android. Esta etapa foi essencial, visto que testes em dispositivos reais podem revelar problemas que não são detectados em ambientes emulados. Infelizmente, devido à necessidade de possuir um sistema operacional macOS para compilar e testar o aplicativo em dispositivos iOS, não foi possível realizar testes nessa plataforma. Durante aproximadamente um mês e meio, os usuários que participaram dos testes forneceram feedbacks constantes sobre o funcionamento do aplicativo. Esse período de validação permitiu a identificação e correção de diversos bugs, aprimorando a estabilidade e a experiência do usuário.

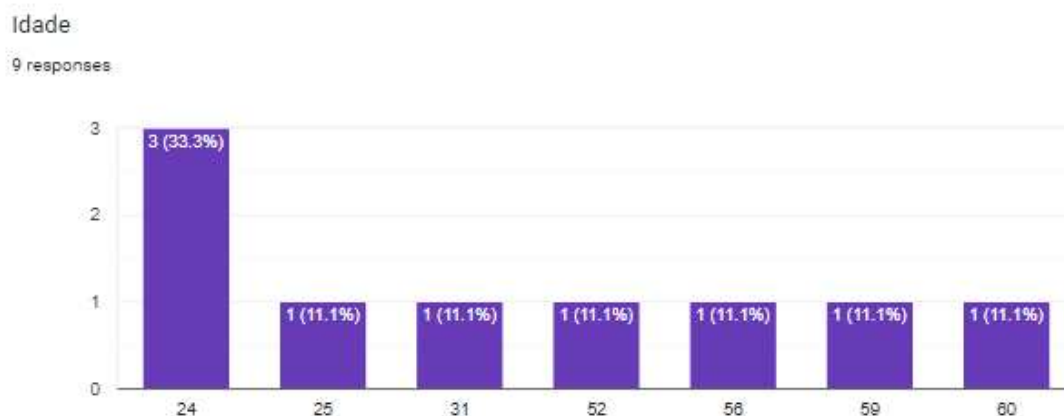
4.5.1 Pesquisa de Teste de Usabilidade

Após finalizada a etapa dos testes, foi realizada uma pesquisa de usabilidade do aplicativo, com perguntas simples e objetivas. A pesquisa foi elaborada com o Google

Forms e envolveu um grupo de 9 participantes, selecionados para fornecer uma visão diversificada e representativa dos potenciais usuários, após o uso do aplicativo.

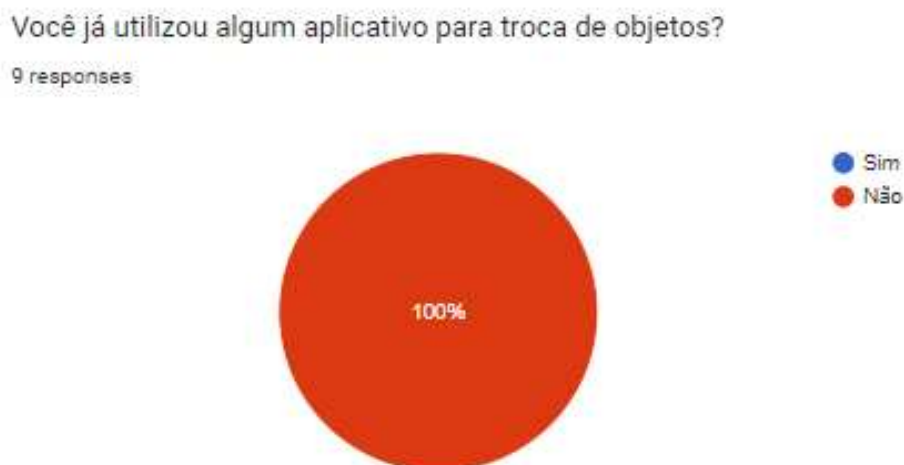
A idade dos usuários variou bastante, sendo pessoas que nunca haviam utilizado um aplicativo de troca de objetos antes. A diversidade dos participantes garantiu uma ampla gama de feedbacks e opiniões.

Figura 27 – Idade dos participantes



Fonte: Autor

Figura 28 – Pergunta 1



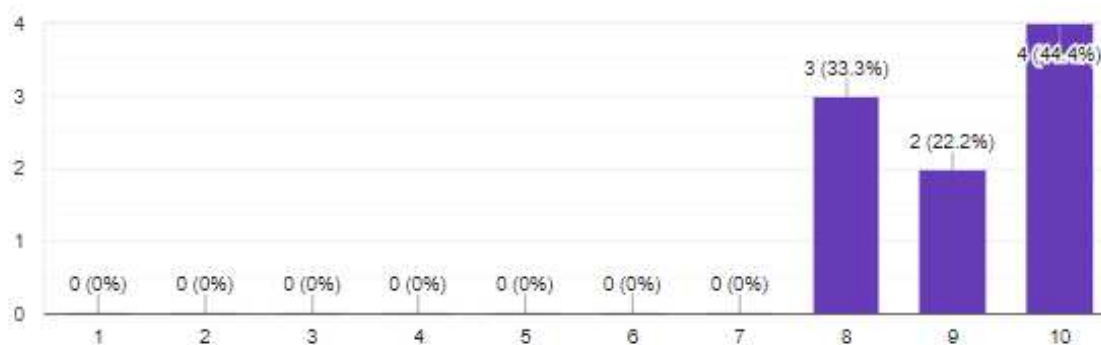
Fonte: Autor

Para avaliar a experiência dos usuários em relação a funcionalidade e usabilidade do aplicativo, foram elaboradas 6 perguntas sobre facilidade, agilidade, aspectos positivos e negativos.

Figura 29 – Pergunta 2

Numa escala de 1 a 10, quão **fácil/intuitiva** é a utilização do Aplicativo?

9 responses

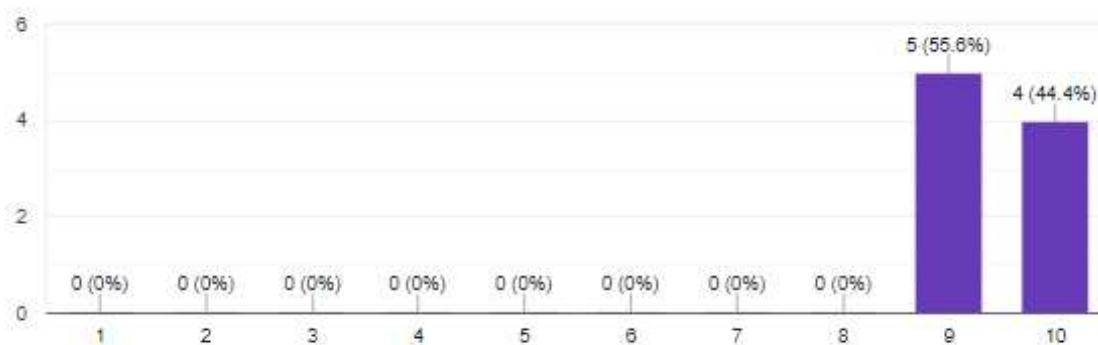


Fonte: Autor

Figura 30 – Pergunta 3

Numa escala de 1 a 10, quão **ágil/rápida** é a utilização do Aplicativo?

9 responses

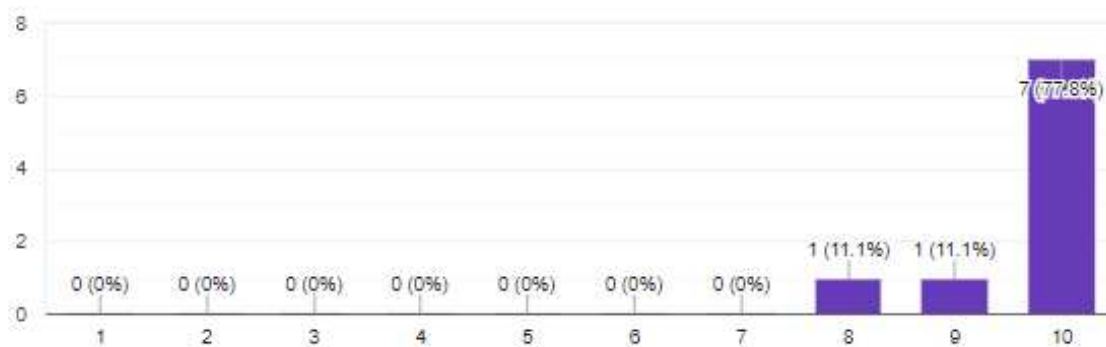


Fonte: Autor

Figura 31 – Pergunta 4

Numa escala de 1 a 10, com qual frequência você enfrentou **problemas** com o Aplicativo?

9 responses

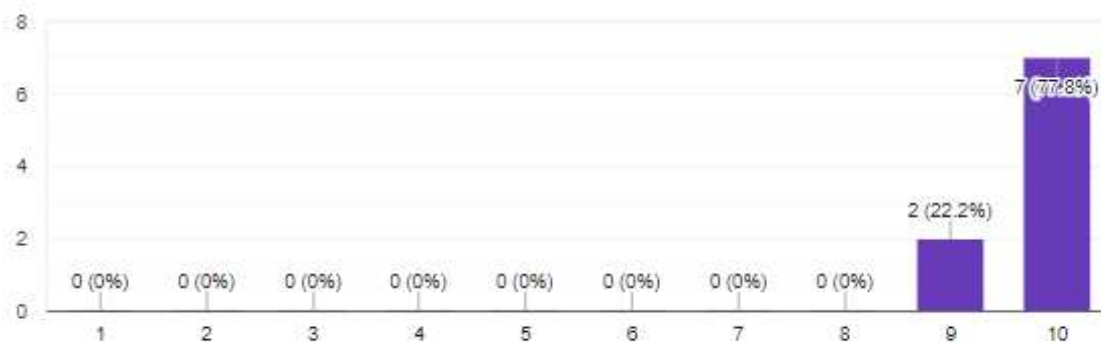


Fonte: Autor

Figura 32 – Pergunta 5

De maneira geral, qual seu nível de **satisfação** com o Aplicativo?

9 responses



Fonte: Autor

Figura 33 – Pergunta 6

Se quiser, cite as coisas que mais te **agradam** no Aplicativo.

9 responses

- Praticidade em encontrar objetos e negociar trocas.
- possibilidade de encontrar diversos objetos em um unico lugar
- Muito interessante a ideia de trocar objetos e não gastar nada.
- Poder fazer login com Google ou Facebook, e poder trocar qualquer coisa
- Filtro para direcionar melhor as buscas por objetos específicos
- gostei muito do visual do aplicativo
- Integração com Google para autenticação.
- Visual moderno e interface intuitiva
- Muito rápido o processo do cadastro até a troca

Fonte: Autor

Figura 34 – Pergunta 7

Agora, caso queira, cite as coisas que mais te **desagradam** com o Aplicativo.

8 responses

Demora para realizar o login pelo Google.

nada muito bom

Falta de um passo a passo para aprender a usar.

Muitos objetos aleatórios

Falta de um espaço para avaliação das trocas

não achei muito intuitivo

Acredito que se houvesse uma notificação para avisar quando um objeto de interesse esteja disponível na minha região seria muito útil.

Algumas leves lentidões no aplicativo (cadastro da foto)

Fonte: Autor

Os resultados da pesquisa forneceram insights valiosos sobre a usabilidade do aplicativo Troka. A maioria dos participantes avaliou a facilidade de uso do aplicativo com notas altas, variando de 8 a 10, indicando que o aplicativo é intuitivo e fácil de navegar. Os usuários destacaram vários pontos positivos do aplicativo, como a praticidade em encontrar objetos e negociar trocas, a possibilidade de fazer login com Google ou Facebook, o visual moderno e a interface intuitiva. Alguns comentários específicos incluíram: "Praticidade em encontrar objetos e negociar trocas.", "Integração com Google para autenticação." e "Visual moderno e interface intuitiva."

Apesar dos feedbacks positivos, os participantes também identificaram áreas que poderiam ser melhoradas. As críticas incluíram: "Demora para realizar o login pelo Google.", "Falta de um passo a passo para aprender a usar." e "Algumas leves lentidões no aplicativo (cadastro e navegação)." Esses feedbacks são essenciais para orientar futuras melhorias no aplicativo, visando tornar a experiência do usuário ainda mais satisfatória.

5 CONCLUSÃO

Desenvolver um aplicativo completo, sendo o único desenvolvedor, foi um desafio significativo. No entanto, essa experiência proporcionou uma oportunidade única de aplicar e integrar diversos conhecimentos adquiridos ao longo da graduação, como Programação Orientada a Objetos, Engenharia de Software, Banco de Dados, Planejamento e Gestão de Projetos, entre outras disciplinas essenciais.

A conclusão deste projeto não apenas resultou em um produto funcional e valioso, mas também representou uma grande oportunidade de crescimento profissional. O desenvolvimento do "Troka" mostrou que, com dedicação e aplicação prática dos conhecimentos adquiridos, é possível enfrentar desafios complexos e alcançar resultados significativos. Este projeto reafirma a importância da interdisciplinaridade e da prática constante para a formação de um profissional completo e preparado para o mercado de trabalho.

O projeto, em consonância com seus objetivos gerais, buscou oferecer uma alternativa inovadora para possivelmente minimizar o descarte desnecessário, conectando pessoas interessadas em trocar objetos de forma prática e segura.

A construção de um aplicativo multiplataforma proporcionou uma experiência de usuário fluida e intuitiva, apesar das dificuldades encontradas, como a limitação de testes em dispositivos iOS. Foi possível realizar a implementação da grande maioria dos requisitos funcionais como cadastro de objetos, busca por categorias e filtros, sistema de sugestões de troca e chat para comunicação entre usuários, assegurou a usabilidade e a efetividade do aplicativo. Além disso, a integração com o Firebase possibilitou a autenticação via Google e Facebook, garantindo segurança e praticidade aos usuários. Porém, nem todas as funcionalidades foram efetivamente implementadas como o mecanismo de notificação para alertar os usuários e também as soluções encontradas na seção de riscos: Política de Privacidade e Uso, Verificação de identidade através de reconhecimento facial, Verificação de objetos ilegais através de IA, Recursos educacionais e uma Área de avaliações.

O processo de desenvolvimento, detalhado ao longo do trabalho, evidenciou a importância de cada etapa, desde a concepção da arquitetura até a realização de testes em dispositivos reais. A pesquisa de teste de usabilidade forneceu uma visão clara dos pontos fortes e fracos do aplicativo. Através do feedback dos participantes, foi possível identificar que o aplicativo é bem recebido em termos de usabilidade e design, mas ainda há áreas que necessitam de ajustes para otimizar a experiência do usuário. Esses insights serão fundamentais para as próximas etapas de refinamento do aplicativo.

5.1 TRABALHOS FUTUROS

Durante o desenvolvimento do aplicativo, alguns objetivos específicos não foram completamente implementados ou resolvidos, e várias melhorias foram identificadas para trabalhos futuros:

- Implementação de uma Área de Notificações Funcional: Embora a tela de notificações tenha sido construída, a implementação completa desta funcionalidade não foi possível devido à falta de tempo hábil.
- Versão para iOS: Apesar de ter desenvolvido o código para multiplataformas, só foi possível lançar o aplicativo para Android, devido às limitações de compilação para MacOS.

Além dos objetivos específicos mencionados, a seção de riscos abordou algumas soluções que poderiam melhorar a segurança e a experiência de uso do aplicativo que por falta de tempo não puderam ter sido implementadas:

- Criação de uma Política de Privacidade e Uso: Estabelecer uma política clara e transparente para proteger a privacidade dos usuários.
- Verificação de identidade: Implementar a solicitação do reconhecimento facial do usuário para validar com seus documentos legais.
- Verificação de Objetos Ilegais através de IA: Implementar algoritmos de inteligência artificial para detectar objetos ilegais.
- Recursos Educacionais: Desenvolver tutoriais e materiais educativos para instruir os usuários sobre práticas seguras de uso do aplicativo.
- Área de avaliações: Desenvolver uma área onde os usuários possam se avaliar após realizarem as trocas.

Todas essas funcionalidades que não puderam ser desenvolvidas no presente trabalho são propostas que, quando implementadas, tornarão o aplicativo mais robusto e benéfico para todos os usuários. A continuidade desse projeto, com a implementação dessas melhorias, proporcionará um aplicativo ainda mais eficiente e segura para a troca de objetos, promovendo um consumo consciente e sustentável.

REFERÊNCIAS

- ALLAIN, Hugo. **Improving productivity and reducing costs of mobile app development with Flutter and Backend-as-a-Service**. [S.l.]: Aalto University, 2020.
- BECK, Kent. **Extreme Programming Explained: Embrace Change**. [S.l.]: Addison-Wesley, 2000.
- CHATTERJEE, Nilanjan; CHAKRABORTY, Souvik; DECOSTA, Aakash; NATH, Asoke. **Real-time Communication Application Based on Android Using Google Firebase**. [S.l.]: IJARCSMS, 2018.
- DAGNE, Lukas. **Flutter for cross-plataform App and SDK development**. [S.l.]: Metropolia, 2019.
- FINPLI. **Finpli**. [S.l.: s.n.], 2021. Acessado em: 13 de novembro de 2023. Disponível em: %5Curl%7Bhttps://play.google.com/store/apps/details?id=com.finpli%7D.
- GEOCODING. **Geocoding API**. [S.l.: s.n.], 2024. Acessado em: 29 de abril de 2024. Disponível em: %5Curl%7Bhttps://geocode.maps.co/%7D.
- IBGE. **API de localidades - IBGE**. [S.l.: s.n.], 2024. Acessado em: 29 de abril de 2024. Disponível em: %5Curl%7Bhttps://servicodados.ibge.gov.br/api/docs/localidades#api-Municipios-estadosUFMunicipiosGet%7D.
- KAZA, Silpa; YAO, Lisa C.; BHADA-TATA, Perinaz; VAN WOERDEN, Frank. **What a Waste 2.0 : A Global Snapshot of Solid Waste Management to 2050**. [S.l.]: World Bank, 2018.
- NEOTRUST. **Com pandemia, vendas pela internet crescem 27% e atingem R\$ 161 bi em 2021**. [S.l.: s.n.], 2022. Acessado em: 16 de dezembro de 2022. Disponível em: %5Curl%7Bhttps://www.neotrust.com.br/2022/04/08/com-pandemia-vendas-pela-internet-crescem-27-e-atingem-r-161-bi-em-2021/#:~:text=0%20e%2Dcommerce%20brasileiro%20registrou,monitoramento%20do%7D.
- PERMUTO. **O seu site de fazer permuta**. [S.l.: s.n.], 2023. Acessado em: 13 de novembro de 2023. Disponível em: %5Curl%7B//https://www.permuto.com.br/%7D.

SHARMA, Vatsal; TIWARI, Ankit Kumar. **A Study on User Interface and User Experience Designs and its Tools**. [S.l.]: World Journal of Research e Review, 2021.

SLATER, Don. **Cultura do consumo e modernidade**. [S.l.]: Nobel, 2002.

TROCAFÁCIL. **Troca Fácil**. [S.l.: s.n.], 2019. Acessado em: 13 de novembro de 2023. Disponível em:

<https://apps.apple.com/br/app/troca-f%C3%A1cil/id1463416361>.

TROKAÍ. **Trokaí**. [S.l.: s.n.], 2023. Acessado em: 13 de novembro de 2023. Disponível em: https://play.google.com/store/apps/details?id=com.trokai.mobile&hl=pt_BR&gl=US.

VIACEP. **ViaCEP**. [S.l.: s.n.], 2024. Acessado em: 29 de abril de 2024. Disponível em: <https://viacep.com.br/>.

APÊNDICE A – CÓDIGO

<https://github.com/brenobrida/trokaApp>

APÊNDICE B – ARTIGO

Desenvolvimento de um aplicativo mobile para troca de objetos

Breno De Brida¹

¹Departamento de Informática e Estatística
Universidade Federal de Santa Catarina (UFSC)

Resumo. *Em 2021, o e-commerce brasileiro registrou um faturamento recorde de R\$ 161 bilhões, um aumento de 26,9% em relação ao ano anterior [Neotrust 2022]. Atualmente, existem poucos aplicativos voltados especificamente para a troca de bens materiais pela internet. Tendo isso em vista, o projeto em questão foi desenvolvido para atender uma necessidade de mercado pouco explorada, com o desenvolvimento de um aplicativo móvel, visando viabilizar um mercado de troca. Os usuários cadastram o objeto que desejam trocar e também realizam ofertas por outros objetos. Uma vez cadastrado, o sistema cruza as informações com as já existentes em seu banco de dados e verifica a existência de outro usuário com o objeto de interesse e que também tenha interesse no objeto do primeiro usuário. Caso o usuário não possua um objeto de interesse específico, o sistema indica os objetos de outros usuários que possuem interesse naquele que ele possui.*

1. Introdução

Segundo o novo artigo do World Bank [Kaza et al. 2018], a produção de lixo mundial irá aumentar em 70%, o quantitativo que em 2016 era de 2,01 Bilhões de toneladas está estimado a saltar para 3,4 Bilhões nos próximos 30 anos. A reciclagem de objetos sólidos é crítica para uma sociedade inclusiva, saudável e principalmente sustentável. Esse crescimento agressivo se dá tanto pelo crescimento populacional, mas principalmente pela cultura de consumismo global. Pessoas ao redor do mundo estão continuamente comprando objetos para sua satisfação pessoal e também para o uso coletivo. De acordo com Slater [Slater 2002], o consumo aumentou devido à redução dos preços com a industrialização, isso fomentou ainda mais o comércio acarretando numa crescente cultura consumista ao redor do mundo. Com a vinda da internet na década de 90, esse processo acabou sendo empurrado ainda mais com a criação dos *e-commerces*.

Um *e-commerce*, comércio eletrônico do inglês, é um ambiente online de compra e venda de produtos que estruturam um negócio. Dessa forma, todas as operações são realizadas na internet por meio de alguma plataforma ou serviço, uma revolução dos meios de varejo físico tradicional. Seu surgimento foi em 1970 nos Estados Unidos por uma necessidade de troca de arquivos e solicitação de pedidos. No Brasil o primeiro registro de e-commerce se deu na década de 90, com um site de venda de livros idealizado pela empresa *Booknet*.

Com o crescimento exponencial do comércio eletrônico a cada ano, com o elevado aumento do consumo pelas pessoas, com a facilidade de aquisição de novos produtos a cada dia, a indústria lançando novos produtos e com vida útil decrescente, há que se pensar no lixo produzido e descartado pela humanidade. É necessário pensar não somente

no hoje, mas no amanhã, daqui a 10, 50, 100 anos, quando nosso planeta poderá estar saturado de seus recursos naturais devido ao consumo proporcionado pela indústria e o mercado. Quando houver escassez de recursos e determinados produtos não mais puderem ser produzidos e entregues pela indústria e o mercado, teremos um sistema que permitirá a seus usuários adquirir daqueles que possuem e não precisam mais.

Atualmente no Brasil existem milhares de plataformas de comércio online, porém, apenas com as opções de comprar ou vender produtos, poucas permitem que um usuário possa trocar um objeto com outro usuário. Podem ser citadas aqui, Permuto [Permuto 2023] e Trokaí [Trokaí 2023], plataformas que permitem a troca de serviços e roupas, respectivamente. Até o momento atual, foram encontradas poucas plataformas que permitem a troca de objetos no geral, apenas de áreas específicas como aquelas citadas.

2. Fundamentação Teórica

Um aplicativo móvel, também conhecido como app, é um software desenvolvido para ser executado em dispositivos móveis, como smartphones e tablets. Esses aplicativos podem ser nativos, desenvolvidos especificamente para um sistema operacional (iOS ou Android), ou multiplataforma, capazes de rodar em diferentes sistemas operacionais com um único código-base.

A estrutura de um aplicativo móvel é composta por várias camadas que trabalham juntas para proporcionar uma experiência de usuário fluida e funcional. Entre essas camadas, destacam-se a interface do usuário (UI), onde os usuários interagem diretamente com o aplicativo. Inclui todos os elementos visuais, como botões, ícones e menus. A lógica de negócio contém as regras e operações que definem o funcionamento do aplicativo. É responsável pelo processamento de dados e a aplicação de lógica específica. O banco de dados, onde os dados são armazenados e gerenciados, pode ser local (armazenado no dispositivo) ou remoto (armazenado em servidores). O backend é a infraestrutura que suporta o aplicativo, fornecendo serviços como autenticação, armazenamento de dados e notificações.

O desenvolvimento de um aplicativo requer a seleção criteriosa de diversas tecnologias para garantir seu sucesso. Esta seção tem como objetivo justificar a necessidade dessas escolhas, destacando a importância de cada uma delas e comparando com alternativas disponíveis no mercado.

A escolha de um framework de desenvolvimento é essencial para a criação de aplicativos eficientes e escaláveis. Frameworks fornecem uma estrutura robusta que facilita o desenvolvimento, manutenção e escalabilidade do código. No mercado, as opções mais conhecidas incluem Flutter, React Native e Xamarin. A escolha do Flutter, por exemplo, foi motivada pela sua performance superior e a riqueza de componentes nativos.

Um banco de dados eficiente é crucial para armazenar e gerenciar dados de maneira segura e escalável. Além disso, uma infraestrutura de backend robusta é necessária para suportar autenticação de usuários, armazenamento de dados, e outras funcionalidades críticas. Entre as opções populares estão Firebase, AWS Amplify e Backendless. A escolha do Firebase foi justificada pela integração perfeita com o Flutter e pela simplicidade na gestão de serviços backend.

Para garantir uma experiência de usuário de alta qualidade, a escolha de uma plataforma de design de interfaces é fundamental. Ferramentas como Figma, Adobe XD e Sketch são amplamente utilizadas. A escolha do Figma foi motivada pela sua facilidade no uso.

3. Proposta de Desenvolvimento

O levantamento de requisitos foi uma etapa fundamental no processo de desenvolvimento do aplicativo, pois envolveu a identificação, análise e documentação das necessidades e expectativas em relação ao sistema. Esse processo buscou entender de forma detalhada quais funcionalidades o sistema deveria oferecer, como ele deveria se comportar e quais restrições deveriam ser observadas durante o desenvolvimento.

Existem dois tipos de requisitos, os funcionais e não-funcionais. Os requisitos funcionais definem as funcionalidades específicas do sistema e como ele deve se comportar, enquanto os requisitos não funcionais descrevem as qualidades e restrições do sistema, como desempenho, segurança e usabilidade.

3.1. Requisitos Não-Funcionais

- RNF1: Interface amigável e intuitiva;
- RNF2: Navegação fácil e rápida entre as funcionalidades do aplicativo;
- RNF3: Resposta rápida às ações dos usuários;
- RNF4: Alta disponibilidade e tempo de resposta eficiente do servidor;
- RNF5: Proteção dos dados pessoais dos usuários;
- RNF6: Criação de políticas de privacidade e uso claras e detalhadas;
- RNF7: Disponível para os sistemas operacionais Android e iOS;
- RNF8: Funcionar em diversas versões dos SOs suportados.
- RNF9: Infraestrutura que permita a adição de novas funcionalidades sem comprometer o desempenho.

3.2. Requisitos Funcionais

- RF1: Permitir que o usuário crie uma conta e se autentique através de Facebook ou Google;
- RF2: Permitir que o usuário cadastre um objeto para troca, incluindo categoria do seu objeto, categoria de objetos de interesse, descrição e fotos;
- RF3: Permitir que o usuário altere informações do objeto já cadastrados;
- RF4: Permitir que o usuário busque objetos de interesse mesmo que não tenha cadastrado um objeto;
- RF5: Exibir uma galeria de objetos disponíveis para troca;
- RF6: Implementar um filtro de pesquisa completo para refinar os resultados de busca;
- RF7: Cruzar informações dos objetos cadastrados para encontrar coincidências de interesse entre usuários;
- RF8: Indicar objetos de outros usuários que possuem interesse no objeto que ele possui;
- RF9: Criar um chat para comunicação entre usuários interessados em realizar a troca;
- RF10: Notificar os usuários sobre novas mensagens e sugestões de trocas;

- RF11: Permitir que os usuários avaliem uns aos outros após a realização de uma troca;
- RF12: Solicitar reconhecimento facial do usuário para a validação da identidade do usuário;
- RF13: Verificar objetos ilegais através de algoritmos de IA;
- RF14: Implementar um tutorial para os usuários no primeiro uso do aplicativo;

4. Desenvolvimento

O desenvolvimento do aplicativo seguiu um processo meticuloso e estruturado, iniciado pelo design da interface e seguido pela implementação das funcionalidades essenciais. A arquitetura escolhida para o desenvolvimento do aplicativo foi a MVVM (Model-View-ViewModel), que organiza o código em três camadas: Model, que gerencia os dados e a lógica de negócios; View, que é responsável pela interface do usuário; e ViewModel, que atua como intermediário, gerenciando o estado e a lógica da interface, facilitando a manutenção e a escalabilidade do aplicativo. O desenvolvimento do aplicativo iniciou-se pelo front-end, priorizando a criação da interface gráfica sem, inicialmente, abordar a parte funcional. O objetivo foi implementar o design de todas as telas, conforme projetado no Figma, com foco na experiência do usuário. Com a interface concluída, procedeu-se ao desenvolvimento do back-end, responsável por implementar todas as funções customizadas necessárias para a integração entre as telas, autenticação com o banco de dados e comunicação com APIs, tornando o aplicativo plenamente funcional.

Nas próximas subseções, será detalhado o desenvolvimento das telas e das funcionalidades principais do aplicativo.

4.1. Login/Cadastro e Autenticação

O desenvolvimento da tela de login e cadastro do aplicativo foi realizado com o objetivo de proporcionar uma experiência de usuário simplificada e segura. A integração com o Firebase permitiu a autenticação via Google e Facebook, duas plataformas comumente usadas por usuários de smartphones.

A tela de login foi construída com a logo do aplicativo e dois botões principais, um para login via Google e outro para Facebook. A lógica de autenticação foi implementada no arquivo `login_model.dart`, utilizando os pacotes `firebase_auth`, `flutter_facebook_auth` e `google_sign_in`.

4.2. Consulta de Objetos e Filtro

A consulta de objetos e o filtro estão dispostos na primeira página do aplicativo, na tela de explorar, que é exibida assim que o usuário realiza sua autenticação. Nesta tela, inicialmente, são exibidos todos os objetos disponíveis na cidade onde o usuário se encontra. Caso o usuário deseje refinar sua busca, ele pode utilizar o campo de pesquisa, localizado no topo da página, para inserir palavras-chave específicas. A caixa de pesquisa é um campo de texto configurado para atualizar os resultados conforme o usuário digita.

Além do campo de pesquisa, o usuário pode utilizar a tela de filtro criada para refinar os resultados. Essa funcionalidade permite realizar o filtro da localização do objeto (estado e cidade), a finalidade (troca ou doação), a condição do objeto (novo, usado, recondicionado ou com defeito), a categoria do item e as categorias de objetos que são

aceitos em troca. Com essas opções de filtragem, os usuários podem encontrar exatamente o que procuram de maneira mais eficiente e precisa.

Para a realização do filtro, foi utilizado a estratégia de armazenar todos os valores em variáveis de estado global. Dessa forma, quando o usuário confirma o filtro no botão "Filtrar" ao final da tela de filtro, todas as escolhas feitas são registradas e usadas para realizar a pesquisa no Firebase.

4.3. Inserção de Objetos

A inserção de objetos foi elaborada de forma que não fosse extensa e contivesse somente os campos necessários para a inclusão do item. A inserção é realizada através de uma série de funções que interagem com o Firebase Firestore e Firebase Storage para armazenar dados e imagens.

Dentro do Firestore Database, foi criada uma coleção chamada "user_objects", onde são armazenados os objetos de todos os usuários. Cada "documento", que é a forma como o Firestore Database denomina os registros individuais de cada item, contém os atributos fornecidos pelo usuário no momento da inserção dos dados do objeto.

4.4. Oferta de Objetos

Dentro do aplicativo Troka, as ofertas são a principal forma de negociação entre os usuários. Para que um usuário realize uma troca bem-sucedida, ele deve primeiro enviar uma oferta a partir da página do objeto desejado. Após enviar a oferta, ela aparecerá na aba "Enviadas" da tela "Ofertas". O usuário que recebe a oferta verá a mesma na aba "Recebidas". Uma vez que a oferta é aceita, ela é movida para a aba "Efetuadas", onde se torna um chat. Este chat permite que os usuários troquem mensagens e combinem os detalhes de como realizar a troca de forma prática.

4.5. Algoritmo de Sugestões de Troca

Na tela de cada objeto que o usuário possui, há um botão no canto inferior direito chamado "Sugestões de Troca". Esse botão busca no banco de dados todos os objetos que pertencem à categoria do objeto que o usuário aceita em troca, além de verificar a reciprocidade. Assim, apenas serão exibidos objetos cujos proprietários também estejam dispostos a aceitar o objeto do usuário em troca. Dessa forma, as sugestões apresentadas são altamente relevantes, aumentando a probabilidade de uma troca bem-sucedida.

A função responsável por isso é a "findSugestion", ela implementa um algoritmo de filtragem e recomendação baseado em consultas de banco de dados. A abordagem teórica subjacente pode ser descrita como um algoritmo de recomendação colaborativa com filtragem baseada em conteúdo. A Recomendação Colaborativa sugere itens com base nas preferências de outros usuários que possuem interesses similares. No contexto da função "findSugestion", a recomendação colaborativa ocorre implicitamente, pois o sistema sugere objetos de outros usuários que podem ser do interesse do usuário atual. A Filtragem Baseada em Conteúdo sugere itens com base nas características dos próprios itens e nas preferências do usuário. A função filtra objetos com base na categoria "objectCategory" e nos interesses do usuário "objectCategoryInterest".

4.6. Chat

O chat foi a última tela desenvolvida no aplicativo, pois todas as demais funcionalidades precisavam estar operacionais para que ele pudesse ser testado. A troca de mensagens é simples: o usuário pode enviar textos ou fotos. Na tela do chat, é possível visualizar o objeto envolvido na troca, além de ter a opção de cancelar a negociação, desfazendo a troca e removendo o objeto da aba "Ofertas Efetuadas".

Para que cada usuário possa visualizar as mensagens de outros usuários, foi criada uma coleção no Firestore Database chamada "chatMessages". Cada documento dessa coleção representa uma mensagem enviada.

Assim, cada vez que um usuário digita uma mensagem ou envia uma foto, o sistema cria um documento nesta coleção. A tela do chat atualiza em tempo real, filtrando apenas os documentos que possuem o "offerId" da troca específica e ordenando-os pela data e hora de envio. Isso garante que as mensagens sejam exibidas na ordem correta e que cada chat reflita apenas as conversas relacionadas a uma determinada oferta.

5. Conclusões e trabalhos futuros

A construção de um aplicativo multiplataforma proporcionou uma experiência de usuário fluida e intuitiva, apesar das dificuldades encontradas, como a limitação de testes em dispositivos iOS. Foi possível realizar a implementação da grande maioria dos requisitos funcionais como cadastro de objetos, busca por categorias e filtros, sistema de sugestões de troca e chat para comunicação entre usuários, assegurou a usabilidade e a efetividade do aplicativo. Além disso, a integração com o Firebase possibilitou a autenticação via Google e Facebook, garantindo segurança e praticidade aos usuários. Porém, nem todas as funcionalidades foram efetivamente implementadas como o mecanismo de notificação para alertar os usuários e também as soluções encontradas na seção de riscos: Política de Privacidade e Uso, Verificação de identidade através de reconhecimento facial, Verificação de objetos ilegais através de IA, Recursos educacionais e uma Área de avaliações.

O processo de desenvolvimento, detalhado ao longo do trabalho, evidenciou a importância de cada etapa, desde a concepção da arquitetura até a realização de testes em dispositivos reais. A pesquisa de teste de usabilidade forneceu uma visão clara dos pontos fortes e fracos do aplicativo. Através do feedback dos participantes, foi possível identificar que o aplicativo é bem recebido em termos de usabilidade e design, mas ainda há áreas que necessitam de ajustes para otimizar a experiência do usuário. Esses insights serão fundamentais para as próximas etapas de refinamento do aplicativo.

Durante o desenvolvimento do aplicativo, alguns objetivos específicos não foram completamente implementados ou resolvidos, e várias melhorias foram identificadas para trabalhos futuros:

- Implementação de uma Área de Notificações Funcional: Embora a tela de notificações tenha sido construída, a implementação completa desta funcionalidade não foi possível devido à falta de tempo hábil.
- Versão para iOS: Apesar de ter desenvolvido o código para multiplataformas, só foi possível lançar o aplicativo para Android, devido às limitações de compilação para MacOS.

Além dos objetivos específicos mencionados, a seção de riscos abordou algumas soluções que poderiam melhorar a segurança e a experiência de uso do aplicativo que por falta de tempo não puderam ter sido implementadas:

- Criação de uma Política de Privacidade e Uso: Estabelecer uma política clara e transparente para proteger a privacidade dos usuários.
- Verificação de identidade: Implementar a solicitação do reconhecimento facial do usuário para validar com seus documentos legais.
- Verificação de Objetos Ilegais através de IA: Implementar algoritmos de inteligência artificial para detectar objetos ilegais.
- Recursos Educacionais: Desenvolver tutoriais e materiais educativos para instruir os usuários sobre práticas seguras de uso do aplicativo.
- Área de avaliações: Desenvolver uma área onde os usuários possam se avaliar após realizarem as trocas.

Referências

Kaza, S., Yao, L. C., Bhada-Tata, P., and Van Woerden, F. (2018). What a waste 2.0 : A global snapshot of solid waste management to 2050.

Neotrust (2022). Com pandemia, vendas pela internet crescem 27% e atingem r\$ 161 bi em 2021. Acessado em: 16 de dezembro de 2022.

Permuto (2023). O seu site de fazer permuta. Acessado em: 13 de novembro de 2023.

Slater, D. (2002). *Cultura do consumo e modernidade*. Nobel.

Trokaí (2023). Trokaí. Acessado em: 13 de novembro de 2023.