



**Cristian Guilherme Rieth**

**PLATAFORMA *ON-LINE* DE ATENDIMENTO LOGÍSTICO  
AO COMÉRCIO VAREJISTA**

Horizontina-RS

2023

**Cristian Guilherme Rieth**

**PLATAFORMA *ON-LINE* DE ATENDIMENTO LOGÍSTICO  
AO COMÉRCIO VAREJISTA**

Trabalho Final de Conclusão de Curso  
apresentado como requisito parcial para conclusão  
do Curso na Engenharia de Controle e Automação  
da Faculdade Horizontina, sob a orientação do  
Prof. Geovane Webler.

Horizontina-RS

2023

## RESUMO

O presente trabalho foi realizado analisando o processo logístico do comércio varejista, mais precisamente a logística até o consumidor final quando a venda é realizada de forma virtual, assim como é apresentado neste trabalho. O mesmo teve como objetivo desenvolver um aplicativo de celular que fosse capaz de conectar comerciante e entregador, a fins de agilizar o processo de entregas e reduzir custos. Foi realizada uma pesquisa de novas tecnologias disponíveis para desenvolvimento de um aplicativo e juntamente compreender o processo de desenvolvimento, onde foi selecionado a tecnologia que mais se encaixa para o proposto. O trabalho foi desenvolvido visando solucionar a lacuna na logística dos comércios varejistas localizados na cidade de Horizontina - RS. Desta forma foram realizadas pesquisas para que o aplicativo desenvolvido pudesse ser aplicado futuramente, assim foi desenvolvido um aplicativo com o básico dos recursos necessários para seu funcionamento.

**Palavras-Chave:** Entregas, Logística, *E-commerce*, Desenvolvimento *Mobile*.

## LISTA DE FIGURAS

Figura 1 - Diferença entre o desenvolvimento híbrido e nativo	15
Figura 2 - Pilha de <i>software</i> do Android	17
Figura 3 - Camadas do <i>Flutter</i>	19
Figura 4 - Exemplo de modelagem <i>JSON</i> em <i>Firebase</i> .	21
Figura 5 - Diagrama de caso de uso	27
Figura 6 - Diagrama de classes	31
Figura 7 - Print de tela 1: Tela de login	33
Figura 8 - Print de tela 2: Lista de entregas pendentes	34
Figura 9 - Print de tela 3: Lista de entregas em andamento	35
Figura 10 - Print de tela 4: Lista de entregas realizadas	36
Figura 11 - Print de tela 5: Detalhes da entrega	37
Figura 12 - Print de tela 6: Tela de nova remessa	38
Figura 13 - Trecho de código página de <i>Login</i>	39
Figura 14 - Classes diretório <i>Lib</i>	40
Figura 15 - Código do arquivo <i>main</i>	40

## **LISTA DE ABREVIATURAS E/OU SIGLAS**

API – *Application Programming Interface*

App – Aplicativo

BaaS – *Back-end as a Service*

GCP – *Google Cloud Platform*

JSON – *JavaScript Object Notation*

NoSQL – *Not Only SQL*

RF – Requisito Funcional

RNF – Requisito Não Funcional

SDK – *Software Development Kit*

SGBD – Sistema Gerenciador de Banco de Dados

SGBDTR – Sistema Gerenciador de Banco de Dados em Tempo Real

SQL – *Structured Query Language*

SO – Sistema Operacional

## SUMÁRIO

<b>1.</b>	<b>INTRODUÇÃO .....</b>	<b>8</b>
1.1	TEMA.....	9
1.2	DELIMITAÇÃO DO TEMA.....	9
1.3	PROBLEMA DE PESQUISA .....	9
1.4	HIPÓTESES .....	9
1.5	JUSTIFICATIVA.....	9
1.6	OBJETIVOS .....	10
1.6.1	Objetivo geral .....	10
1.6.2	Objetivos específicos.....	10
<b>2</b>	<b>REFERENCIAL TEÓRICO.....</b>	<b>11</b>
2.1	LOGÍSTICA .....	11
2.1.1	Logística de distribuição .....	12
2.2	<i>E-COMMERCE</i> .....	12
2.3	AUTOMAÇÃO E TECNOLOGIA.....	13
2.4	APLICAÇÕES HÍBRIDAS E NATIVAS .....	14
2.5	DESENVOLVIMENTO DE <i>SOFTWARE</i> .....	15
2.5.1	Sistemas Operacionais.....	15
2.5.2	Linguagem de programação.....	17
2.5.3	Bancos de dados.....	19
<b>3</b>	<b>METODOLOGIA.....</b>	<b>24</b>
<b>4</b>	<b>APRESENTAÇÃO E ANÁLISE DOS RESULTADOS .....</b>	<b>26</b>
4.1	SITUAÇÃO ATUAL.....	26
4.2	ESPECIFICAÇÕES DOS REQUISITOS E PROCESSOS .....	26
4.2.1	Requisitos funcionais.....	27
4.2.2	Requisitos não funcionais.....	29
4.3	IDENTIFICAÇÃO DO PROCESSO .....	31
4.3.1	Diagrama de Classes .....	31
4.3.2	Resultados Esperados .....	32
4.4	APRESENTAÇÃO DO APLICATIVO DESENVOLVIDO.....	32
4.4.1	Arquitetura de Desenvolvimento .....	38
	<b>CONCLUSÃO .....</b>	<b>42</b>

REFERÊNCIAS.....	43
------------------	----

## 1. INTRODUÇÃO

A logística é uma atividade essencialmente competitiva para as organizações, envolvendo todas as etapas de movimentação e armazenagem que contribuem para o fluxo de produtos, desde a aquisição da matéria-prima até a distribuição física. Nesse sentido, a realização de entregas precisas, no local correto, no tempo adequado e nas condições desejadas, ao menor custo possível, é fundamental para atender às necessidades dos clientes e aprimorar o nível de serviço da empresa (BALLOU, 2006).

De acordo com Matos Júnior e Carlos Alberto (2013), através da logística é possível melhorar a utilização dos veículos, entregar mais com menos, reduzir os custos de entrega, dimensionar a quantidade de entregas conforme a capacidade da rota ou jornada, diminuir os gastos com combustível, pneus, manutenção, horas extras, balanceando os recursos necessários para execução do planejado. Diante disso, desenvolver uma maneira para obter uma melhoria na utilização dos recursos empregados no cotidiano destas operações de transportes logísticos é um desafio para as empresas, sendo um importante elemento para aumentar a potencialidade das empresas, seja por melhor desempenho logístico, por melhor nível de serviço e menor redução de custos de operação.

Os dispositivos móveis traçaram um longo caminho nos últimos anos. Por um tempo, os celulares eram apenas usados com o intuito de fazer ligações. Todavia, à medida que esses dispositivos continuavam evoluindo, a capacidade de enviar e receber mensagens de texto e multimídia, criar agendas e salvar contatos tornou-se prontamente disponível. Atualmente os dispositivos móveis são usados em larga escala e são capazes de atender a diversos outros propósitos.

Com base nos estudos realizados e nas informações disponíveis de diversos autores conforme relatado acima, este trabalho tem como objetivo buscar desenvolver um aplicativo de celular que seja capaz de conectar lojista e motorista, onde será possível cadastrar os produtos com sua descrição, endereço de retirada, endereço de entrega e o valor oferecido para os motoristas realizarem a entrega. Este trabalho será dividido em três partes principais. A primeira parte consistirá na contextualização do estudo, onde serão abordados o tema, o problema, os objetivos e a justificativa. A segunda parte apresentará o referencial teórico, que servirá como base para a

execução do trabalho. A terceira etapa abordará os aspectos metodológicos a serem seguidos, além do desenvolvimento do aplicativo e os resultados obtidos.

### 1.1 TEMA

Plataforma on-line para atendimento logístico ao comércio varejista.

### 1.2 DELIMITAÇÃO DO TEMA

Plataforma on-line de atendimento logístico ao comércio varejista em geral residentes na cidade de Horizontina - RS e região.

### 1.3 PROBLEMA DE PESQUISA

Atualmente, o comércio varejista de Horizontina enfrenta desafios na eficiência de suas operações de entrega, visto que os atendentes muitas vezes precisam deixar a loja para realizar as entregas ou entrar em contato com entregadores disponíveis na cidade.

De que maneira uma plataforma on-line pode auxiliar o comércio varejista a atender a demanda com eficiência fazendo com que seus produtos cheguem ao consumidor de maneira rápida e efetiva com baixo custo?

### 1.4 HIPÓTESES

- O aplicativo desenvolvido pode definir uma padronização para a entrega de produtos do comércio.

- O aplicativo desenvolvido é uma forma eficiente de transformar um motorista que tem o deslocamento nas estradas como rotina no seu dia a dia em um transportador.

### 1.5 JUSTIFICATIVA

De acordo com Marconi e Lakatos (2022), a justificativa possui o objetivo de apresentar o porquê de algo estar sendo realizado. A mesma faz uso de recursos teóricos e práticos para assim conseguir alcançar seu objetivo. A Justificativa serve, principalmente, para justificar a pesquisa e convencer o leitor sobre a importância do trabalho. O avanço da tecnologia aplicada em smartphones e a maneira como as empresas modernas estão se comportando está mudando constantemente. Os

aplicativos estão cada vez mais presentes na vida das pessoas e do mundo corporativo, pela praticidade e otimização de processos. E quando se fala de logística, eles estão mais presentes do que se pode pensar.

Constantemente, há pessoas em trânsito pelas ruas e, com o uso de um aplicativo, qualquer indivíduo cadastrado pode acessá-lo e realizar entregas que estejam em seu trajeto, permitindo assim dividir os custos da viagem. A automação dos processos resulta em uma redução de custos para os comerciantes ao operar o sistema. Sem um aplicativo de logística, a empresa precisaria de pelo menos 1 a 3 funcionários adicionais diariamente para lidar com chamadas, se comunicar com entregadores e clientes, realizar a verificação dos pedidos, entre outras tarefas. Com um aplicativo de logística, todas essas atividades podem ser realizadas por apenas uma pessoa, o que diminui consideravelmente os custos operacionais da empresa

## 1.6 OBJETIVOS

### 1.6.1 Objetivo geral

Desenvolver um aplicativo de celular que seja capaz de conectar comerciante e entregador.

### 1.6.2 Objetivos específicos

- ✓ Realizar a programação do aplicativo;
- ✓ Usar um banco de dados em tempo real para gerenciamento;
- ✓ Fazer teste do funcionamento geral do aplicativo;
- ✓ Documentar os resultados.

## 2 REFERENCIAL TEÓRICO

Nesta seção, são apresentados os conceitos técnicos que compõem o trabalho, abordando todas as informações necessárias para o desenvolvimento. Além disso, são apresentados conceitos sobre o funcionamento de aplicativos e o desenvolvimento para plataformas móveis, bem como informações relevantes sobre bancos de dados. Esses conceitos fundamentais servirão como base para o desenvolvimento do trabalho.

### 2.1 LOGÍSTICA

A logística abrange os processos de planejamento, implantação e controle para garantir o fluxo eficiente e eficaz de mercadorias, serviços e informações, visando atender às necessidades do cliente e controlar todo o processo de transporte, desde o início até o fim. Além disso, o tempo de trânsito das mercadorias tem impacto direto no número de fretes realizados por veículo em determinado período e nos custos totais de transporte para todos os embarques. O autor também destaca que, em um mercado em constante expansão, pode-se tolerar certos níveis de ineficiência logística e ainda obter rentabilidade. No entanto, a distribuição eficiente e eficaz pode se tornar um diferencial importante na busca por maior competitividade e participação no mercado (BALLOU 2006).

Para Caxito (2019), a logística desempenha um papel crucial ao conectar todos os departamentos, incluindo vendas, *marketing*, finanças, custos, pesquisa e desenvolvimento e produção, todos eles direcionados para um objetivo comum: o sucesso na consecução de metas. Nesse sentido, é imperativo contar com um processo logístico eficiente para a comercialização de bens em um mundo cada vez mais globalizado. Eficiência logística é alcançada quando os custos nas transações são minimizados, e embora seja desafiador alocar recursos de forma integrada e econômica, uma vez que isso é realizado, os concorrentes terão dificuldades em alcançar o mesmo nível de conquista (GIACOMELLI, PIRES, 2016).

“Logística é o processo de planejar, implementar e controlar de maneira eficiente o fluxo e a armazenagem de produtos, bem como os serviços e informações associados, cobrindo desde o ponto de origem até o ponto de consumo, com o objetivo de atender aos requisitos do consumidor” (NOVAES, 2021, p.53).

A logística está diretamente ligada com a obtenção de vantagem competitiva e a diminuição de custos. Trata-se de um conceito em constante evolução, que busca atender às necessidades globais de agir de forma ágil diante de mudanças nas circunstâncias e no ambiente (FARIA; ROBLES; BIO, 2004).

### **2.1.1 Logística de distribuição**

O sistema de distribuição é composto por atividades primárias e de suporte, estando as atividades primárias relacionadas ao fluxo de entrega, desde o início do processo com o produto até o consumidor final. Essas atividades primárias incluem recebimento, armazenagem, operações, processamento de pedidos e distribuição física. As atividades de suporte são responsáveis por apoiar as atividades primárias, como o suprimento e a previsão da infraestrutura (ARBACHE, 2004). O processo de distribuição física abrange desde a realização do pedido, seu processamento, separação no setor de pedidos e, por fim, as operações de carregamento e transporte visando concluir a entrega. Esse processo tem um impacto direto nas áreas de marketing e vendas, determinando o preço de venda do produto, a disponibilidade e o tempo necessário para o ressurgimento da empresa (BOWERSOX; CLOSS; COOPER, 2014).

## **2.2 E-COMMERCE**

O comércio eletrônico, também conhecido como *e-commerce*, é uma forma tecnológica de realizar transações comerciais. O comércio, em sua essência, envolve a troca de bens e serviços visando benefícios ou satisfação mútua. Por sua vez, o comércio eletrônico refere-se especificamente à compra e venda realizada por meio de plataformas digitais (TEIXEIRA, 2015).

### **O comércio eletrônico**

“É um formato de comércio que apresenta grande potencial, buscando renovar alguns conceitos clássicos, criando um inovador cenário comercial, encontrando e adequando seus pontos fortes em áreas de grande relevância e conhecimento, como a logística, a qual torna-se uma das principais bases para a evolução do e-commerce” (GUIMARÃES, *et al*, 2015, p. 10).

Uma das vantagens do comércio eletrônico é que tanto os fornecedores quanto os consumidores são beneficiados, uma vez que as barreiras de mercado são eliminadas. "O comércio eletrônico apresenta novas possibilidades, tanto para as organizações que pretendem comercializar quanto para os consumidores que

desejam maior praticidade" (GUIMARÃES, *et al*, 2015). As empresas têm a oportunidade de ampliar seus mercados consumidores e são incentivadas a obter vantagens competitivas em relação aos concorrentes, principalmente ao reduzir os custos. Por sua vez, os clientes desfrutam de preços mais baixos, uma maior variedade de produtos, além da conveniência e facilidade de realizar compras e receber entregas (PATTAT, 2014).

De acordo com Teixeira (2015), o suporte da informática permite que os comerciantes alcancem um público muito maior, pois as barreiras geográficas não são mais um obstáculo na divulgação de seus negócios. No entanto, ainda existem desafios logísticos relacionados à entrega desses produtos, os quais estão sendo abordados pelas empresas.

### 2.3 AUTOMAÇÃO E TECNOLOGIA

A definição de automação compreende a ideia de "um sistema de controle de um processo, contendo mecanismos responsáveis por verificar seu próprio funcionamento, efetuando medições e introduzindo correções, com mínima ou nenhuma interferência do homem" (PELLINI, 2017). Em outras palavras, trata-se da execução de atividades sem depender da aplicação do trabalho convencional realizado por pessoas.

De acordo com a explicação fornecida por Villela e Vidal (2003), "a história da humanidade é um longo processo de redução do esforço humano requerido para fazer trabalho". Com o surgimento da Revolução Industrial, surgiu a mecanização dos processos, que posteriormente evoluiu para a automação. Essa evolução das operações tem consequências sociais significativas. Segundo Bayer, Eckhardt e Machado (2011), a automação em diversos setores de atividade humana trouxe uma série de vantagens para a sociedade. "A automação geralmente reduz custos e aumenta a produtividade do trabalho" (BAYER; ECKHARDT, MACHADO, 2011). Porém, o aumento da automação intensificou o desemprego em setores que demandam menos qualificação da mão de obra.

A finalidade da automação é aprimorar a velocidade, qualidade, repetibilidade, eficiência, segurança e continuidade das operações (PELLINI, 2017). Não é difícil encontrar exemplos de processos automatizados em nossa rotina diária, uma vez que a automação está cada vez mais presente e possui uma ampla gama de aplicações.

A implementação da automação em processos logísticos pode ocorrer de diversas maneiras. De acordo com Corrêa e Mello (2014):

“A tecnologia de automação dos processos de armazenagem vem se desenvolvendo intensamente através de sistemas de automação de fluxo de materiais, como veículos e empilhadeiras automaticamente guiados, transportadores contínuos dos mais diversos tipos, sistemas de separação totalmente automatizados, estruturas autoportantes operadas com transelevadores” (CORREA; MELLO, 2014, p. 93).

A automação dos processos representa o futuro das indústrias. Ao acompanhar o avanço da tecnologia, a automação desempenha um papel fundamental ao aumentar a eficiência das empresas, melhorar a qualidade das operações e proporcionar maior comodidade aos usuários (PELLINI, 2017).

## 2.4 APLICAÇÕES HÍBRIDAS E NATIVAS

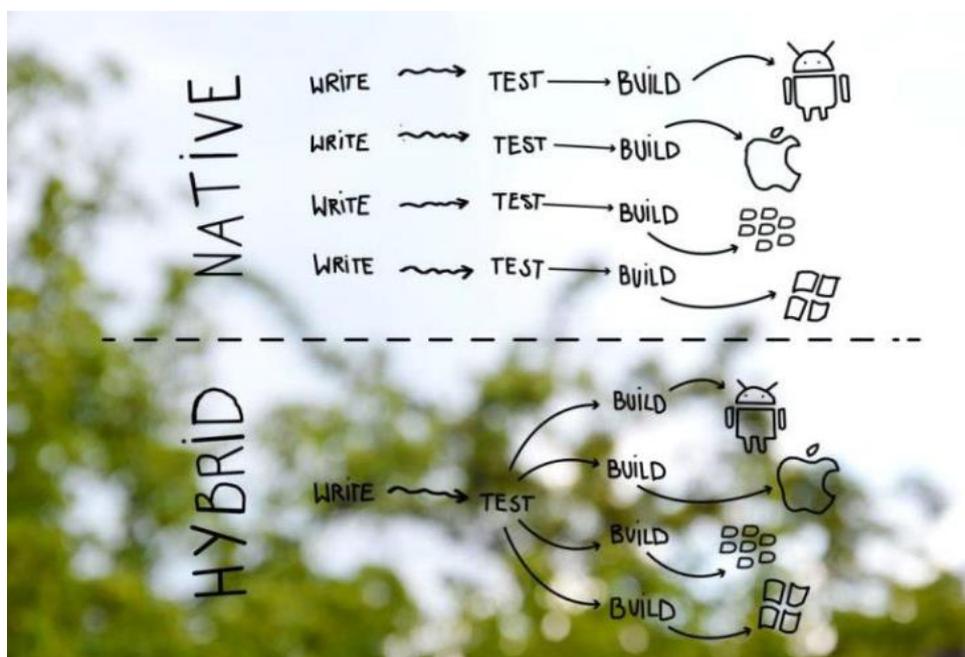
Existem duas abordagens principais para o desenvolvimento de aplicações móveis: nativa e híbrida. No desenvolvimento de aplicações nativas, o desenvolvedor precisa criar componentes exclusivos para cada plataforma na qual o software será disponibilizado. Conforme destacado por Madureira (2017), nas aplicações nativas é necessário programar na linguagem específica de cada sistema operacional, como *Java* ou *Kotlin* para *Android* e *Swift* para *iOS*, sendo que cada plataforma possui suas próprias ferramentas e elementos de interface.

Ao suportar uma aplicação para dois sistemas operacionais nativamente, é comum ter equipes de desenvolvedores especializados em cada plataforma, como *Android* e *iOS*. Isso ocorre porque cada sistema operacional possui suas próprias especificidades, diretrizes de design e conjuntos de ferramentas. No desenvolvimento híbrido, por outro lado, é possível ter uma equipe de desenvolvedores *Web* com conhecimento geral em desenvolvimento *mobile*. Com o uso de *frameworks* híbridos, como *React Native* ou *Flutter*, é possível escrever o código uma vez e implantá-lo em múltiplas plataformas. Essas ferramentas fornecem uma camada de abstração que permite que o aplicativo seja executado nativamente em diferentes sistemas operacionais, sem a necessidade de ter conhecimento especializado em cada plataforma (HEITKÖTTER; HANSCHKE; MAJCHRZAK, 2012).

Aplicativos nativos são caracterizados por serem desenvolvidos em linguagem padrão da plataforma, em geral cada sistema operacional possui sua própria linguagem de programação, estes aplicativos não podem ser portados diretamente

para outras plataformas pois será necessária uma nova implementação. A Figura 1 ilustra a diferença entre a abordagem híbrida e nativa no desenvolvimento de aplicações móveis.

Figura 1 - Diferença entre o desenvolvimento híbrido e nativo



Fonte: Venteu e Pinto (2018)

No caso das aplicações híbridas, é necessário escrever o código em uma única linguagem de programação e, dessa forma, disponibilizar a aplicação para várias plataformas a partir desse mesmo código-fonte.

## 2.5 DESENVOLVIMENTO DE SOFTWARE

### 2.5.1 Sistemas Operacionais

De acordo com Tanenbaum e Bos (2016), é difícil determinar com absoluta certeza o conceito primário de um sistema operacional, pois ele desempenha várias funções, além de operar no núcleo. Em geral, os sistemas operacionais têm duas funções principais que não estão relacionadas entre si. Uma delas é fornecer um conjunto de recursos abstratos, como programas e aplicativos, enquanto a outra é gerenciar os recursos de *hardware* relevantes. Em outras palavras, os sistemas operacionais realizam a tarefa de transformar as complexidades do *hardware* em abstrações compreensíveis para o usuário final.

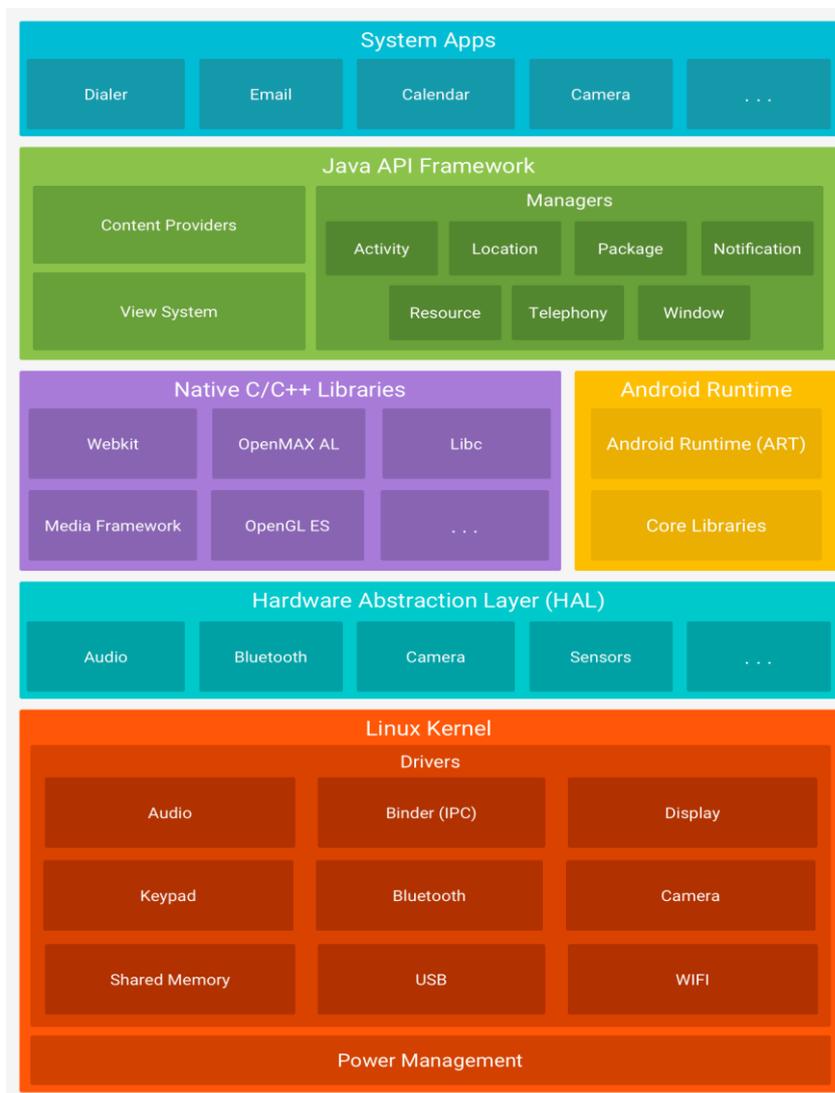
Os SO *mobile* seguem os mesmos conceitos dos sistemas operacionais tradicionais, mas são projetados para dispositivos móveis, como *smartphones* e

*tablets*. Esses dispositivos são conhecidos por terem telas sensíveis ao toque e poucos ou nenhum botão físico. Atualmente, os principais sistemas operacionais móveis são o *Android*, desenvolvido pela *Google*, e o *iOS*, que pertence à *Apple* (TANENBAUM; BOS, 2016).

O sistema operacional *Android* é desenvolvido para dispositivos móveis e utiliza o *kernel* do *Linux* como base, incorporando alguns conceitos próprios ao *kernel*. Ele faz uso dos principais recursos do *Linux*, como processos, IDs de usuário, memória virtual, sistemas de arquivos e escalonamento, porém, em muitos casos, de maneiras diferentes das originalmente projetadas (TANENBAUM; BOS, 2016).

A Figura 2 ilustra a estrutura de software do *Android*, destacando as diferentes camadas do sistema operacional. Na camada mais baixa, temos o gerenciamento de energia, seguido pelo *kernel* do *Linux* e seus drivers. Acima do *kernel* do *Linux*, encontramos a camada de abstração de *hardware*, seguida pelas bibliotecas de linguagem nativa do *Android*, que trabalham em conjunto com as bibliotecas nativas de C/C++. Na sequência, temos a API do *Java*, e no nível mais alto os aplicativos instalados no dispositivo.

Figura 2 - Pilha de software do Android



Fonte - Android Developers (2023).

## 2.5.2 Linguagem de programação

Uma linguagem de programação é uma forma específica e formal de notação utilizada para descrever algoritmos que serão executados. Ela é composta por dois elementos principais: a sintaxe e a semântica. A sintaxe consiste em um conjunto de regras que definem a estrutura dos programas por meio de caracteres. Por outro lado, a semântica é responsável por determinar o significado e o valor dos objetos utilizados nos programas (RODRIGUES, 2007).

### 2.5.2.1 Dart

*Dart* é uma linguagem de programação de alto nível de código aberto desenvolvida pelo *Google*, com foco no desenvolvimento *web*. Ela foi criada com o

propósito principal de simplificar a criação de aplicações *web* que podem se tornar muito complexas se desenvolvidas utilizando abordagens tradicionais, como linguagens de marcação (WALRATH; LADD, 2012).

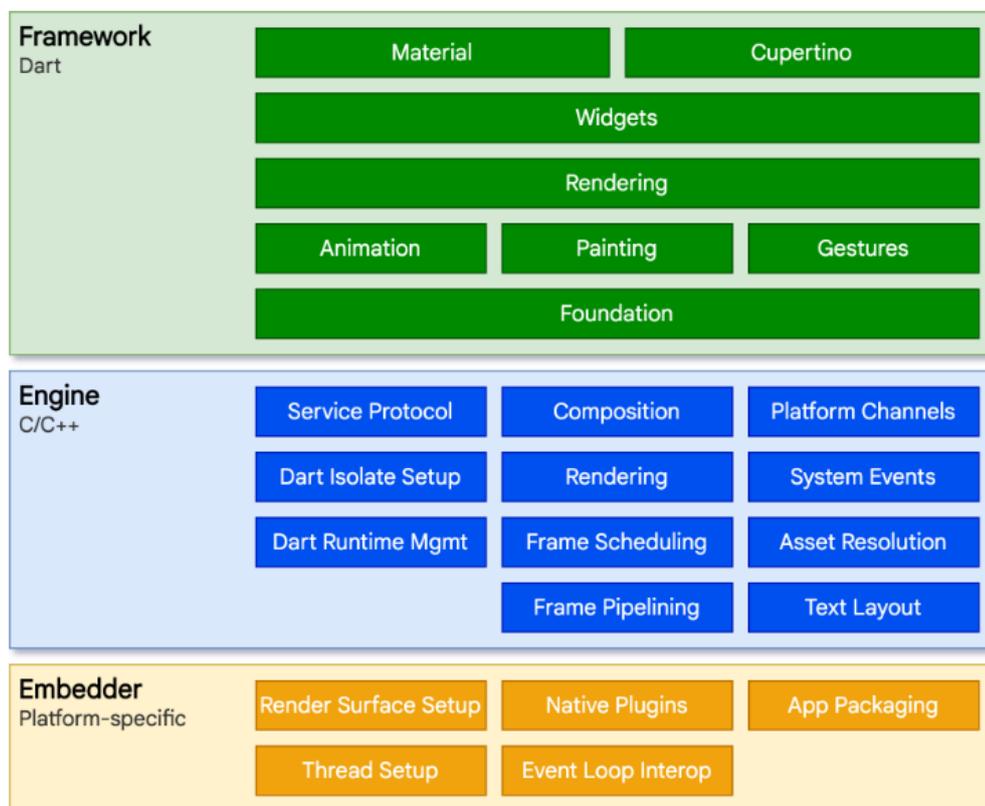
O *Dart* utiliza uma sintaxe inspirada na linguagem C e tem como principais características: Tipagem forte, funções de nível de topo (não necessitando obrigatoriamente de uma classe para execução das funções), tipos genéricos, orientação a objetos, funções de primeira classe (FLUTTER, 2023).

#### 2.5.2.2 Flutter

*Flutter* é um *framework* de desenvolvimento criado pelo *Google* para a criação de aplicativos móveis híbridos compatíveis com *iOS* e *Android*. O principal objetivo do *Flutter* é tornar o desenvolvimento de aplicativos o mais fácil e produtivo possível. Ele oferece recursos como o *Stateful Hot Reload*, que permite carregar alterações no dispositivo ou emulador em tempo real, sem a necessidade de compilar todo o aplicativo a cada modificação. Além disso, o *Flutter* utiliza componentes gráficos chamados *Widgets*, que estão disponíveis em diversos catálogos, facilitando a construção da interface do aplicativo (MAINKAR; GIORDANO, 2019).

Os *Widgets* são componentes de interface gráfica que descrevem a aparência e o comportamento dos elementos da interface do usuário. Eles podem ser organizados em blocos, linhas e colunas para criar *layouts* flexíveis. Uma das características distintivas do *Flutter* é que, em vez de separar as propriedades dos componentes em várias visualizações ou páginas, ele utiliza um único objeto de modelo chamado *Widget*. Os *widgets* carregam por padrão os estilos *Material*, padrão comum usado em aplicações *Android*, e *Cupertino*, usado para *IOS*.

A Figura 3 apresenta uma representação em camadas da estrutura básica do *Flutter*. Ela divide o *Flutter* em três camadas principais. A primeira camada é o próprio *framework* do *Flutter*, desenvolvido com a linguagem de programação *Dart*. A segunda camada é a *Engine*, que pode ser o *Android Studio*, por exemplo. Por fim, temos a camada de mais baixo nível, que representa a plataforma específica do usuário. Cada uma dessas camadas possui suas próprias subcamadas, que contribuem para o funcionamento do *Flutter* como um todo.

Figura 3 - Camadas do *Flutter*

**Fonte:** Adaptado de *Flutter* (2023).

Conforme descrito na página de visão geral dos aspectos técnicos do *Flutter*, o *framework* é um SDK (*Software Development Kit*) que permite a criação de aplicativos de alto desempenho para *Android* e *iOS* a partir de um único código-fonte. Isso significa que os desenvolvedores podem escrever o código uma vez e executá-lo em ambas as plataformas, evitando a necessidade de desenvolver aplicativos separados para *Android* e *iOS*. O *Flutter* oferece uma abordagem de desenvolvimento eficiente e consistente, resultando em aplicativos com desempenho otimizado em ambas as plataformas (FLUTTER, 2023).

### 2.5.3 Bancos de dados

Um banco de dados pode ser definido como um conjunto de dados que representa valores ou informações do mundo real. A capacidade dos bancos de dados deriva de um corpo de conhecimento e tecnologia que se desenvolveu ao longo de várias décadas e é representado por um *software* especializado chamado sistema de gerenciamento de banco de dados (SGBD). Um SGBD oferece recursos para criar, manter e realizar operações na estrutura e nos dados armazenados no banco de

dados. Ele permite o acesso e a manipulação eficiente dos dados, garantindo a integridade e a segurança das informações (SILBERCHATZ, 2020).

Os bancos de dados podem ser construídos a partir de um modelo relacional, embasado em um modelo entidade-relacionamento ou também podem ser bancos não relacionais. Nos bancos relacionais, as relações podem ser representadas através de um modelo de tabelas com linhas e colunas, onde as linhas representam um conjunto de valores e as colunas representam qual valor está representado. O sistema de banco de dados é basicamente um sistema de manutenção de registros por computador, ou seja, um sistema cujo objetivo global é manter as informações de forma organizada, permitindo o acesso rápido e eficiente a esses dados quando solicitados. (DATE, 2004).

Um banco de dados é uma ferramenta criada com a finalidade de gerenciar dados através de um computador, de forma que os dados nele armazenados mantenham estas informações acessíveis quando necessárias. A manipulação de um banco de dados inclui funções como consulta para recuperar dados específicos, alteração dos dados conforme a demanda e a geração de relatórios com base nos dados gravados (ELMASRI; NAVATHE, 2011).

### 2.5.3.1 Banco de dados não relacionais

No contexto atual, onde um grande volume de dados é gerado a todo momento por diversas aplicações, como redes sociais, há uma grande necessidade para armazenamento e consulta de dados não estruturados, assim havendo um incentivo para o surgimento de novos paradigmas e tecnologias como uma nova categoria de banco de dados, chamada NoSQL (*Not Only SQL*). Essa tecnologia foi proposta com o objetivo de atender aos requisitos de gerenciamento de grandes volumes de dados, semiestruturados ou não estruturados, que necessitam de alta disponibilidade e escalabilidade (LÓSCIO; OLIVEIRA; PONTES, 2015). Em sua maioria, bancos NoSQL são escritos em *JSON*, conforme a figura 4, adaptado da documentação do *Firebase*:

Figura 4 - Exemplo de modelagem JSON em *Firebase*

```

1 {
2   "usuarios": {
3     "jpaulo": {
4       "name": "Paulo Jaime",
5       "contacts": { "edu": true },
6     },
7     "edu": { ... },
8     "maia": { ... }
9   }
10 }

```

**Fonte:** Adaptado da documentação do *Firebase* (2023).

### 2.5.3.2 Banco de dados em tempo real

Sistemas em tempo real podem ser definidos como sistemas para funcionar em um tempo definido. Ou seja, executar certas tarefas com certas restrições de tempo. Portanto, a noção de correção de um sistema em tempo real depende da correção lógica dos resultados produzidos, bem como do momento em que esses resultados são produzidos (ALDARMI, 1998).

Um banco de dados em tempo real é simplesmente a integração de um SGBD convencional com um Sistema em Tempo Real. Então, um SGBDTR, além de ser capaz de processar transações e ter a obrigação de garantir a integridade e disponibilidade dos dados armazenados, deve trabalhar em tempo real para obedecer às regras de tempo, com as impostas aos sistemas em tempo real. As principais características de um SGBDTR contemplam a noção de dados temporalmente consistentes, e a habilidade para definir restrições temporais às transações (PERKUSICH; TURNELL; PERKUSICH, 2018).

Ainda conforme Perkusich, Turnell e Perkusich (2018), observa-se casos em que há restrições temporais que quando impostas às transações precisam ser concluídas com o objetivo de manter a consistência temporal dos dados, ou seja, evitar com que atualizações dessincronizadas possam prejudicar a integridade dos dados.

### 2.5.3.3 *Firebase*

O *Firebase* é um serviço da *Google Cloud Platform* para prover BaaS e armazenamento de dados, além de oferecer suporte para autenticação de usuários. Quando há integração de um aplicativo com o *Firebase*, não há necessidade de digitar

código *back-end* ou se preocupar com a estrutura dessa parte do programa (CHENG, 2017).

O *Realtime Database* do *Firebase* é um banco de dados não relacional (NoSQL) que permite a distribuição de conteúdo multiplataforma e com a possibilidade de trabalho *offline*. Com o *Realtime Database* não se faz necessária a criação e configuração de servidores ou APIs. É ótimo para validar ideias de apps e soluções *web* pois não requer manutenção de infraestrutura (CARDOSO, 2018).

A estrutura do banco de dados em *Firebase* se dá por um objeto JSON em hierarquia de árvore, essa estrutura é capaz de suportar diversos tipos de dados. (CHENG, 2017).

O *Firebase Authentication* fornece serviços de *back-end* e SDKs prontos para autenticar usuários em aplicativos. É oferecido suporte à autenticação através de senhas, números de telefone e até mesmo provedores como *Google*, *Facebook*, *Twitter* entre outros. Para conectar um usuário a um app, primeiro deve-se ter as credenciais de autenticação do usuário, podendo ser o endereço de *e-mail* e a senha do usuário ou até mesmo um *token* de acesso do *OAuth* de um provedor de identidade federado. Então, são essas credenciais são enviadas para o SDK do *Firebase Authentication*, onde o *back-end* faz a verificação e retorna-se uma resposta ao cliente (FIREBASE, 2023).

Após fazer login, se tem acesso a informações básicas do perfil do usuário e a capacidade de controlar o acesso do mesmo aos dados armazenados em outros produtos do *Firebase*. É possível também usar o *token* de autenticação fornecido para verificar a identidade dos usuários nos seus próprios serviços de *back-end* (FIREBASE, 2023).

Para o armazenamento de arquivos no *Firebase* usa-se a função de *Cloud Storage*. Com os SDKs do *Firebase* para *Cloud Storage*, é permitido fazer *upload* e *download* de arquivos nos aplicativos associados ao *Firebase*, independentemente da qualidade da rede. Os arquivos são armazenados em um repositório do GCP e são acessados por meio do *Firebase*. Isso permite executar processos no servidor como filtragem de imagens, vídeo ou outro tipo de documento (FIREBASE, 2023).

#### 2.5.3.4 GCP - *Google Cloud Platform*

O GCP consiste em uma plataforma de acesso a ativos de *hardware* e *software* da *Google*, sendo que esses ativos estão distribuídos em diversos centros da *Google*

pelo mundo. As regiões se dividem em EUA, Europa Ocidental e leste da Ásia, sendo que as mesmas se subdividem em zonas, e a zona, por sua vez, é identificada através de uma nomenclatura que combina um identificador de letra com o nome da região. Como por exemplo, a zona a na região do Leste da Ásia se chama asia-east1-a. A distribuição de recursos oferece vantagens como redundância em caso de falha e latência reduzida, localizando recursos mais próximos de cada cliente, bem como introduz regras sobre como recursos podem ser usados juntos (GOOGLE, 2023).

#### 2.5.3.5 Projetos GCP e *Firebase*

Para usar ou alocar quaisquer recursos do GCP, é necessário um projeto. Um projeto pode ser pensado como a entidade organizadora do que se está construindo. Um projeto no GCP é feito das configurações, permissões e de outros metadados que descrevem os aplicativos. Os recursos alocados dentro de um único projeto podem funcionar juntos, por exemplo, comunicando-se por meio de uma rede interna. Os recursos usados por cada projeto sempre se mantêm separados por limites de projeto entre si, para conectar os mesmos só é possível através de uma rede de internet (GOOGLE, 2023).

Ao criar um projeto no *Console* do *Firebase*, na verdade está sendo criado um projeto do GCP. É possível fazer a analogia de um projeto do GCP como um container virtual para dados, código, configurações e serviços. Então, um projeto do *Firebase* é um projeto GCP com configurações e serviços específicos do *Firebase*. Da mesma forma, também é possível criar um projeto primeiramente no GCP para depois adicionar o *Firebase* ao mesmo (FIREBASE, 2023).

### 3 METODOLOGIA

Para Gil (2022) a metodologia traz de forma detalhada os procedimentos que são percorridos ao longo da pesquisa, onde cada estudo possui sua própria abordagem, procedimentos e técnicas específicas, adaptadas de acordo com os objetivos e características da pesquisa em questão. A metodologia científica fornece um guia estruturado para a condução do estudo, permitindo a replicação dos passos seguidos e garantindo a validade e a confiabilidade dos resultados obtidos.

O objetivo desta pesquisa é descrever o desenvolvimento e aplicação de um aplicativo para auxiliar nas entregas de mercadorias do comércio varejista localizado na cidade de Horizontina no Noroeste do Estado do Rio Grande do Sul, realizado no primeiro semestre de 2023.

Para atender a este objetivo foi delineada uma pesquisa que quanto aos objetivos pode ser classificada de acordo com Lakatos e Marconi (2022), como sendo descritiva, pois segundo estes autores uma pesquisa descritiva visa descrever as características de determinada população ou fenômeno ou o estabelecimento de relações entre variáveis. Além disso, esta pesquisa pode ainda ser classificada como aplicada, pois tem como objetivo resolver problemas reais existentes da logística no comércio varejista.

Os métodos de procedimento têm como papel explicar os objetos menos aplicados da pesquisa e se relacionam especificamente com a constituição das etapas da pesquisa. Os métodos de procedimento são fundamentados por procedimentos de ação, como a coleta de dados, a análise de dados e a interpretação dos resultados obtidos (LOVATO; EVANGELISTA; GULLICH, 2007).

Os procedimentos seguidos para a realização da pesquisa foram a pesquisa bibliográfica, que envolveu a consulta de uma variedade de obras escritas sobre os tópicos abordados, proporcionando um maior conhecimento sobre eles. Além disso, a parte prática envolverá o desenvolvimento de um aplicativo móvel, com integração a um banco de dados não relacional em tempo real.

Inicialmente, realizou-se uma entrevista informal com alguns comerciantes a fim de compreender o real problema enfrentado na entrega de seus produtos aos clientes. Em seguida, deu-se início ao planejamento do aplicativo, definindo suas principais funcionalidades. Posteriormente, foi realizada a programação do aplicativo,

seguida pelo teste de seu funcionamento, e por fim apresentou-se o protótipo do aplicativo.

De acordo com Marconi e Lakatos (2022), as técnicas consistem num composto de métodos a serem utilizados para a elaboração da pesquisa a ser realizada ou como um modo apropriado de se investigar a sistemática. Nesse caso, as técnicas se referem principalmente às etapas de produção do aplicativo, de como será planejado e desenvolvido e quais tecnologias serão usadas.

Como técnica de codificação da aplicação, foi usada a IDE *Android Studio*, uma ferramenta criada e mantida pela *Google* para desenvolvedores *Android*. Como linguagem de programação, foi usada a linguagem Dart juntamente com o *framework Flutter*. Já como banco de dados foi usado o *Firebase*, o qual tem suporte à integração com *Flutter* e *Android Studio*.

## 4 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

Ao decorrer desta seção são apresentados, respectivamente, a identificação das partes interessadas no aplicativo seguido de suas necessidades e problemas, a análise do sistema, caso de uso e diagramas de classe. Em seguida é apresentado os resultados esperados, a apresentação de aplicativo desenvolvido e a arquitetura de desenvolvimento.

### 4.1 SITUAÇÃO ATUAL

O aplicativo foi desenvolvido baseado no comércio varejista localizado na cidade de Horizontina - RS. Atualmente o comércio possui clientes tanto em Horizontina quanto nas cidades vizinhas, como Três de Maio e Santa Rosa. Apesar das lojas serem físicas, nos últimos anos houve um aumento significativo de vendas realizadas de formas virtuais, através de redes sociais ou aplicativos de mensagens.

Ao conversar com um comerciante, foi possível observar que atualmente não há uma forma eficaz de realizar as entregas para seus clientes. Quando há necessidade de realizar uma entrega, o próprio responsável precisa deixar o estabelecimento, ou então, entrar em contato com algum entregador disponível na cidade. Conversando com donos de outros comércios na cidade, foi constatado que vários comerciantes sofrem com esse problema. Ou seja, não há uma forma padrão para a realização das entregas, prejudicando o atendimento da loja e tendo custos elevados.

O fato de ter um aplicativo como o proposto, poderia permitir que qualquer pessoa que estivesse se deslocando pudesse realizar uma entrega. Por exemplo; alguém que está em Horizontina precisa se deslocar até Três de Maio, a mesma poderia utilizar o aplicativo e verificar se há alguma entrega para ser feita com o mesmo destino, podendo assim aceitar realizar a entrega. Com isso o entregador abateria os custos de sua viagem, e a loja teria menos custos com o frete.

### 4.2 ESPECIFICAÇÕES DOS REQUISITOS E PROCESSOS

Ao dispor a contextualização da situação atual do processo e dos objetivos e problema propostos, é possível escalar as funcionalidades que se desejam no aplicativo. O aplicativo foi desenvolvido tendo como foco atender a demanda de

entregas no comércio varejista. Em geral, o aplicativo deve permitir o cadastro de um produto, excluir um produto cadastrado, aceitar realizar uma entrega, acompanhar o status de uma entrega, finalizar uma entrega em andamento, cancelar uma entrega em andamento, concluir uma entrega e permitir visualizar os detalhes de uma entrega independente do status.

#### 4.2.1 Requisitos funcionais

Como requisitos funcionais podem ser identificados os casos de: cadastrar produto, excluir produto, realizar entrega, finalizar entrega, cancelar entrega e ver detalhes da entrega. Os requisitos funcionais são descrições específicas dos casos de uso que são aplicados às regras de negócio, conforme o diagrama de caso de uso, ilustrado na Figura 5.

Figura 5 - Diagrama de caso de uso



Fonte - Autor, 2023

Portanto, a partir da ilustração do diagrama, é possível identificar os requisitos funcionais do aplicativo para então descrever os mesmos. Os requisitos identificados são cadastrar produto, excluir produto, realizar entrega, cancelar entrega e finalizar entrega, os quais estão respectivamente descritos como:

**RF1 - Cadastrar Produto**

**Descrição:** A criação de um produto é o início do fluxo do produto no sistema. Após o produto ser criado, o mesmo entra em *workflow* onde é possível visualizar o status do produto conforme o andamento do mesmo e então interagir com os produtos cadastrados.

**Entrada:** É necessário informar uma descrição para o produto, o endereço de retirada, o endereço de entrega e o valor oferecido para realizar a entrega.

**Saída Principal:** Após clicar no botão de concluir, exibir uma mensagem de confirmação.

**Saída Alternativa:** Caso campos obrigatórios não sejam preenchidos, notificar nos campos através dos validadores.

**RF2 - Excluir Produto**

**Descrição:** A função de excluir produto defere-se a necessidade de excluir um produto que não necessita mais ser realizada a entrega.

**Entrada:** Como entrada pode ser considerado o produto em si e a confirmação no botão de excluir.

**Saída Principal:** O produto deve ser removido da lista de pendentes.

**RF3 - Realizar Entrega**

**Descrição:** Para iniciar a entrega, o produto já deve ter sido cadastrado anteriormente e deve estar na aba "Remessas Pendentes". Essa função tem como objetivo destacar o produto como estando em andamento.

**Entrada:** Como entrada pode ser considerado o produto em si e a confirmação no botão realizar entrega.

**Saída Principal:** O produto deve ir para a aba "Entregas em Andamento".

**RF4 - Acompanhar Entrega**

**Descrição:** Deve permitir ao usuário e motorista visualizar detalhes acerca do produto.

**Entrada:** O usuário deve selecionar o produto que deseja consultar.

**Saída Principal:** Exibe detalhes do produto selecionado.

**RF5 - Finalizar Entrega**

**Descrição:** Permite finalizar a entrega, alterando o status para "Finalizado"

**Entrada:** O entregador deve marcar o produto como "Finalizar entrega".

**Saída Principal:** Informa que o produto foi entregue com sucesso.

**RF6 - Cancelar Entrega**

**Descrição:** A função de cancelar entrega refere-se à necessidade de cancelar uma entrega que havia sido marcada como realizar entrega.

**Entrada:** O usuário deve selecionar a entrega que deseja cancelar.

**Saída Principal:** O produto fica como status "cancelado".

**RF7 - Histórico de Entregas**

**Descrição:** Ao final de uma entrega poderá ser consultado o histórico de todas as entregas já realizadas.

**Entrada:** O usuário deve selecionar a aba "Realizadas".

**4.2.2 Requisitos não funcionais**

Os requisitos não funcionais são aqueles necessários para o funcionamento da aplicação, porém não interferem nas regras de negócio. De tal forma, foram elencados

como requisitos não funcionais o Sistema operacional Android, Conexão com Internet, Sincronização com banco de dados e Login, respectivamente descritos abaixo:

#### **RNF1 - Sistema Operacional Android**

Para utilização da aplicação, o usuário deve dispor de um smartphone ou tablet que utilize o sistema operacional Android.

#### **RNF2 - Conexão com Internet**

A conexão com a internet se faz necessária para sincronizar os dados com o serviço do Firebase. O sistema deve ter implementado um modo *offline First* para gravar produtos ou eventuais mudanças nos produtos no próprio *smartphone* em caso de falta de conexão com a internet.

#### **RNF3 - Sincronização com banco de dados**

O sistema deve ter implementado um modo *offline First* para gravar produtos ou eventuais mudanças nos produtos no armazenamento do próprio *smartphone* em caso de falta de conexão com a internet. Ao reconectar o dispositivo, a sincronização com o serviço do *Firebase* deve ser automática.

#### **RNF4 - Login**

O acesso de cada usuário do aplicativo deve acontecer a partir de um login e senha único por usuário. Haverá apenas um tipo de usuário, não sendo necessário identificar se é entregador ou usuário.

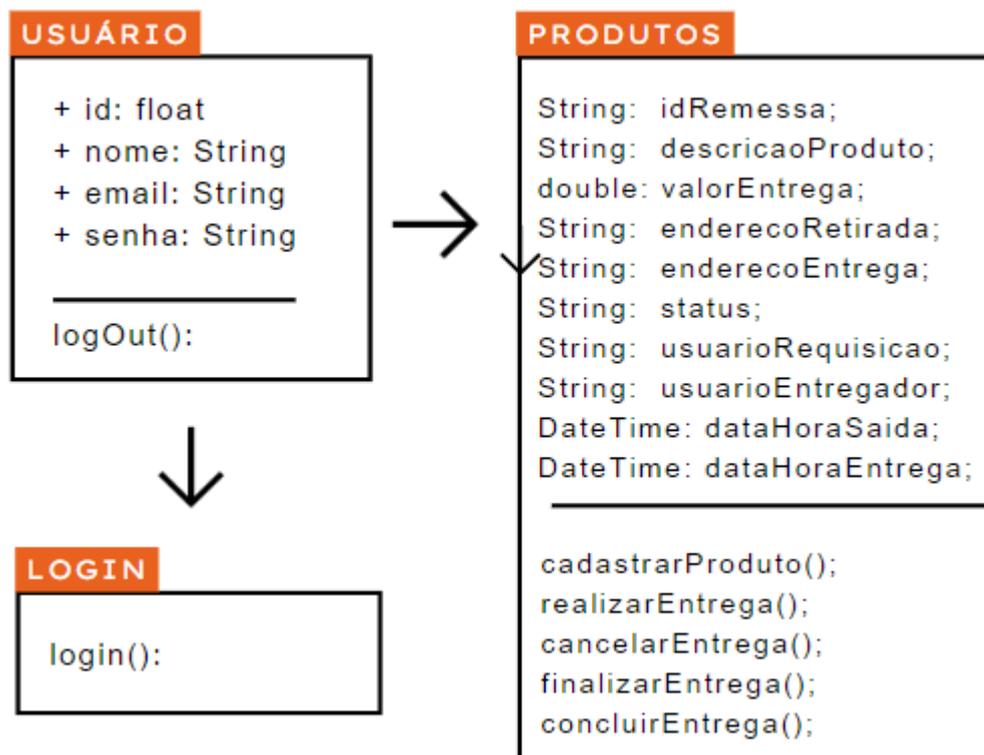
### 4.3 IDENTIFICAÇÃO DO PROCESSO

O processo inicia-se com a necessidade de o comerciante efetuar a entrega de algum produto até seu cliente, o qual realiza o cadastro do mesmo no aplicativo. Neste ponto, é permitido a qualquer pessoa cadastrada efetuar a entrega, tendo finalizado basta marcar como concluída no aplicativo.

#### 4.3.1 Diagrama de Classes

A Figura 6 tem por objetivo identificar as classes presentes no aplicativo. O aplicativo é composto por basicamente 3 classes, sendo essas: Usuário, Produto e Login. A classe de usuário possui como atributos um id, nome, e-mail e senha, tem como método direto o *logout*. A classe Produtos possui os atributos de Id, descrição, endereço de retirada, endereço de entrega, valor, status, usuário requisição, usuário entregador, data hora de saída e data hora de entrega e da mesma forma possui seus métodos: cadastrar Produto, realizar Entrega, cancelar Entrega, finalizar Entrega e concluir Entrega.

Figura 6 - Diagrama de classes



### 4.3.2 Resultados Esperados

Ao se atingir o prazo final para o fim do desenvolvimento, era esperado que o aplicativo desenvolvido proponha a capacidade de cadastrar uma entrega, acompanhar as entregas em andamento e visualizar o histórico de entregas realizadas, além de permitir que o usuário seja capaz de excluir uma entrega cadastrada ou cancelar uma entrega que esteja em andamento. O aplicativo deve rodar em *smartphones* com sistema operacional *Android*.

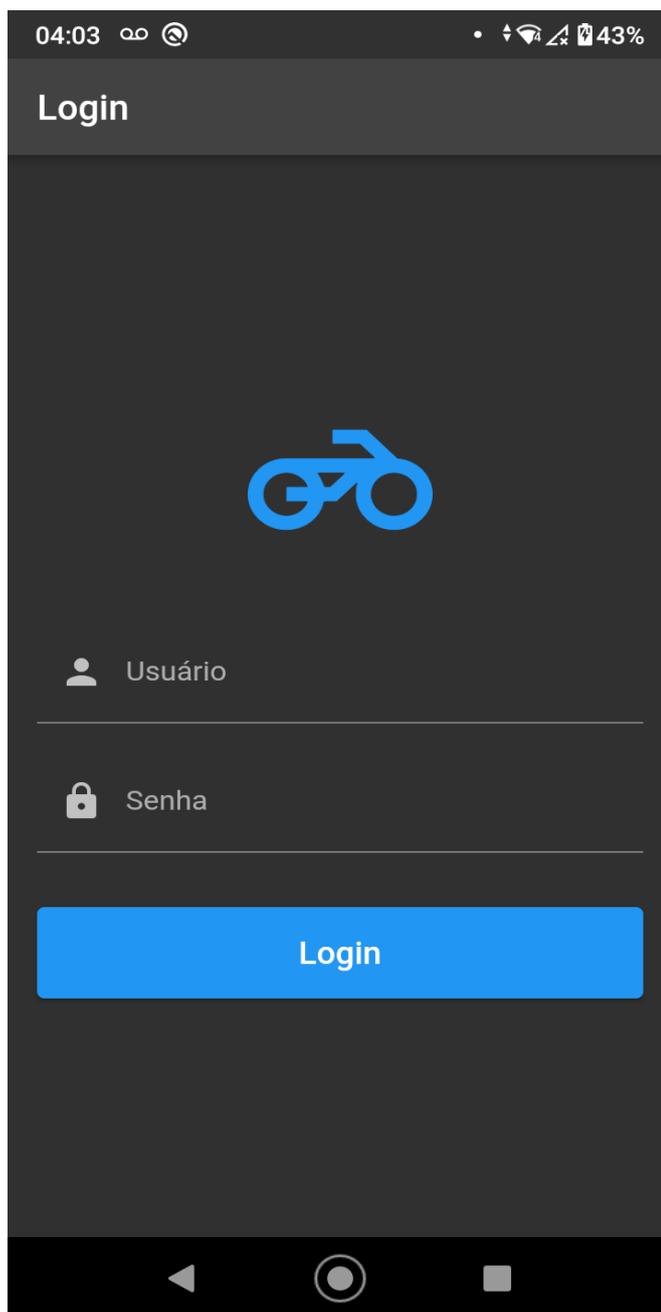
Apesar do aplicativo ser desenvolvido em uma plataforma de código híbrido, o uso em aparelhos com IOS só pode ser feito a partir do *download* da loja de aplicativos da Apple ou usando um computador com *MacOS* como ambiente de desenvolvimento, o que não se identifica como cenário atual para o desenvolvimento do aplicativo. No entanto, devido aos fundamentos do *framework Flutter*, o mesmo projeto pode ser compilado em um computador com *MacOS* e com um aparelho IOS e funcionará da mesma maneira.

## 4.4 APRESENTAÇÃO DO APLICATIVO DESENVOLVIDO

Essa seção aborda a apresentação do que foi desenvolvido como interface final do aplicativo, assim como a apresentação do software em relatório. De forma básica, o app baseia-se em uma tela principal com visualização de 3 abas, sendo essas os *status* de uma entrega. O protótipo do aplicativo também dispõe de uma tela de login que dá acesso à tela das entregas.

A tela principal, além das 3 abas para mostrar as entregas em seus status, dá acesso a tela de cadastrar entrega através do *Floating Action Button* com o *label '+'*. A tela de login dos usuários, é apenas para ilustração do protótipo, até então, ao fazer o login não é realizado nenhum tipo de validação das credenciais. A Figura 7 representa a tela de login, na qual há os campos de usuário e senha com um botão para fazer o login. Ao clicar no botão de login, o app deve redirecionar para a página principal das entregas.

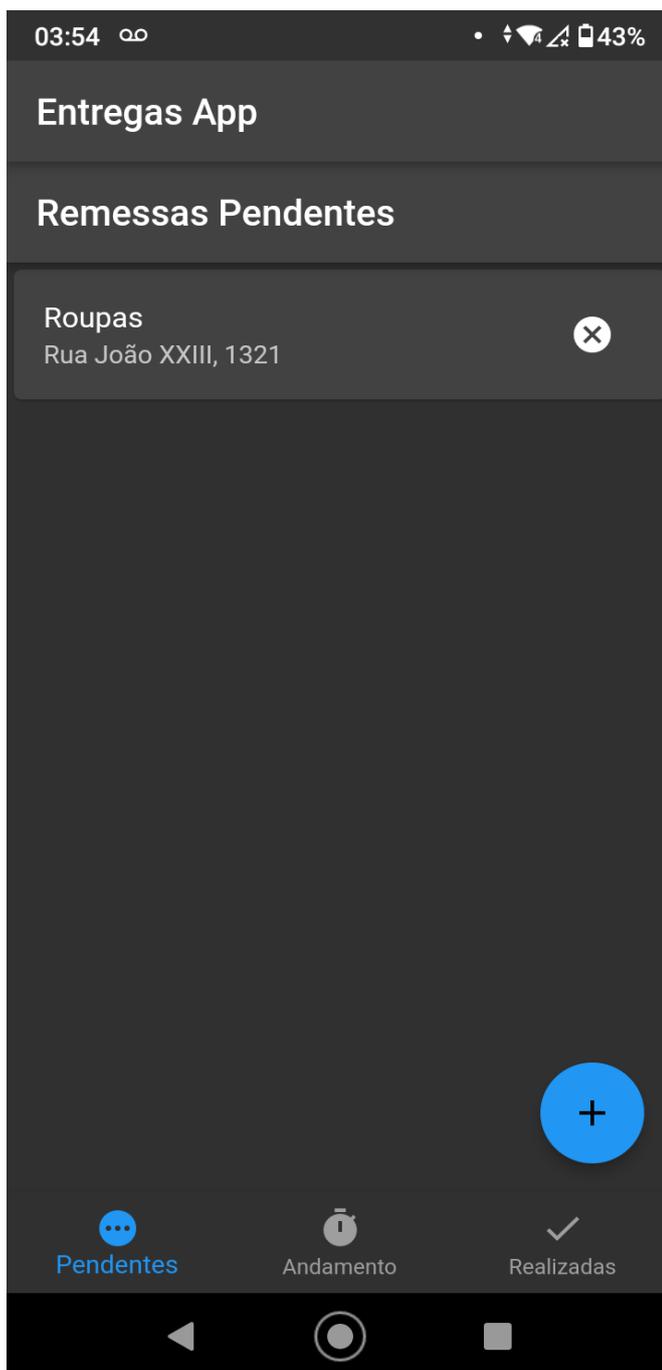
Figura 7 - Print de tela 1: Tela de login



Fonte - Autor, 2023

A Figura 8 representa a tela principal com a aba de remessas pendentes, nessa tela são listados todos os produtos cadastrados que ainda não foram atendidos. Portanto, as entregas podem ser atendidas ou canceladas diretamente.

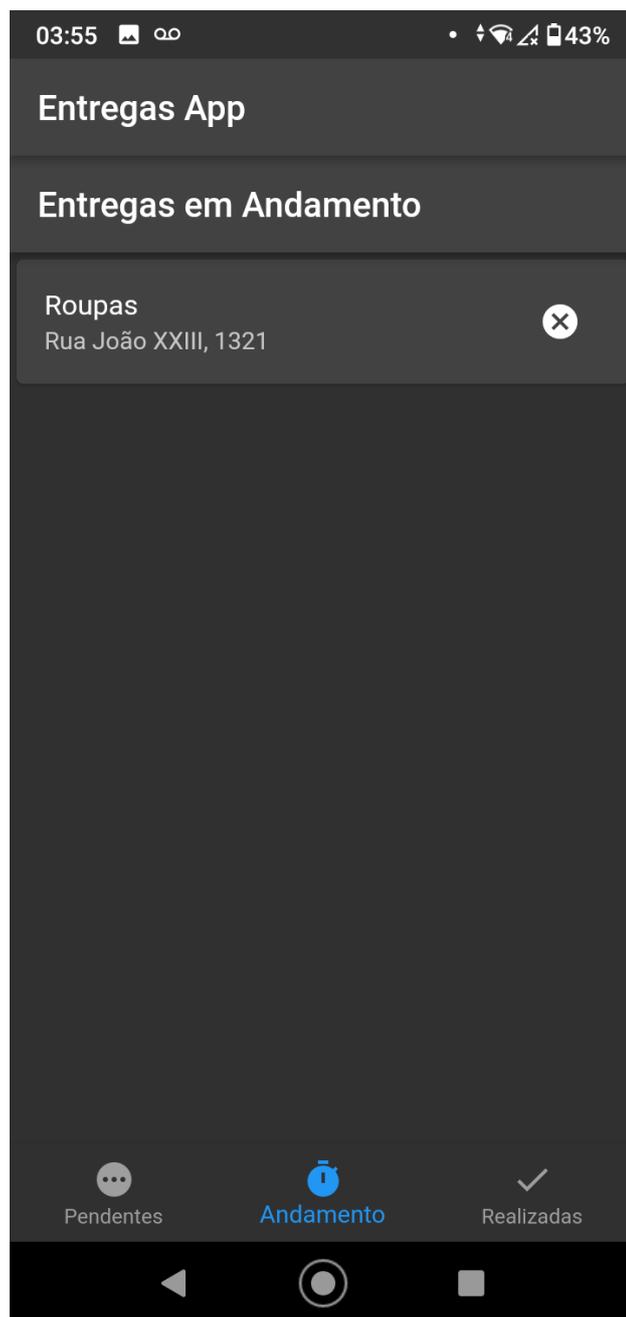
Figura 8 - Print de tela 2: Lista de entregas pendentes



Fonte - Autor, 2023

Na aba de entregas em andamento, como mostra a Figura 9, se localizam as entregas que já foram aceitas e estão em andamento. Nessa parte as entregas podem ser finalizadas ou canceladas.

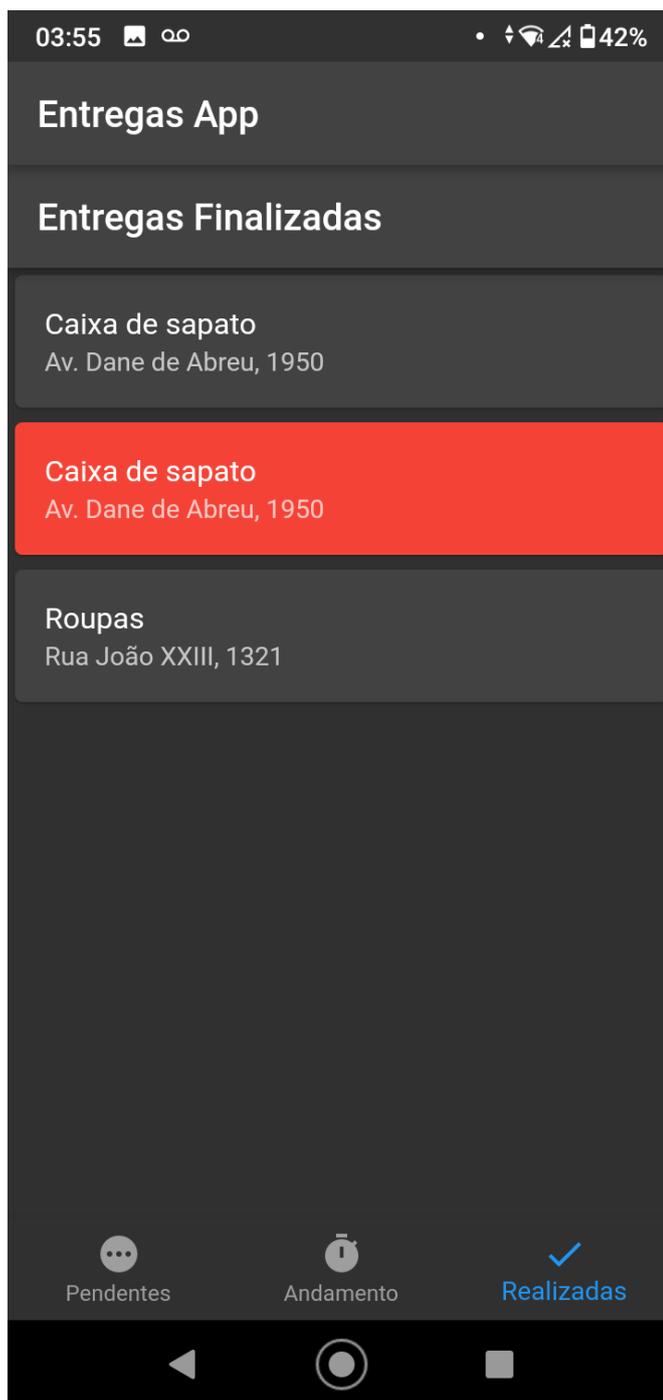
Figura 9 - Print de tela 3: Lista de entregas em andamento



Fonte - Autor, 2023

Na aba de entregas finalizadas, conforme a Figura 10, se localizam as entregas que já foram atendidas sendo finalizadas ou canceladas. Na imagem pode ser observado em vermelho, são as entregas que foram canceladas.

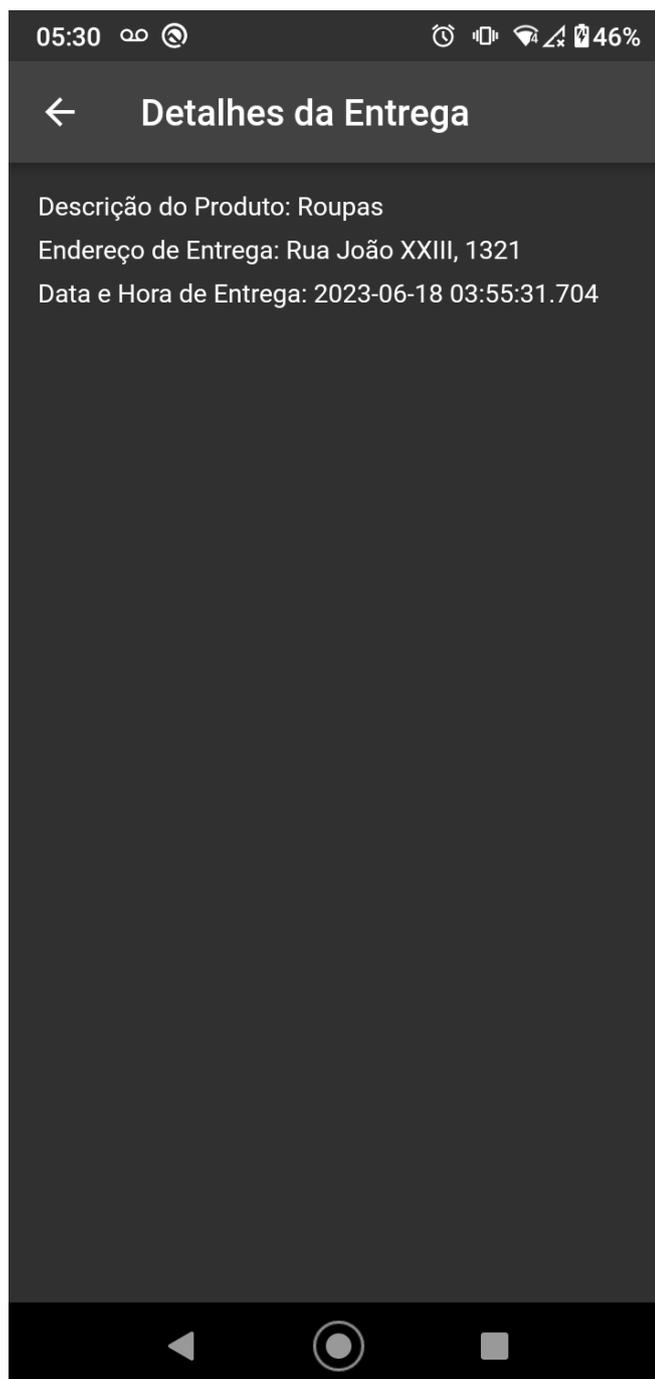
Figura 10 - Print de tela 4: Lista de entregas realizadas



Fonte - Autor, 2023

Em cada *card* de remessa, ao clicar sobre o mesmo exibe as informações referente aquela entrega, se os detalhes forem visualizados na tela de realizadas, serão exibidas as informações como descrição do produto, endereço de entrega, data e hora que a entrega foi concluída, conforme a Figura 11.

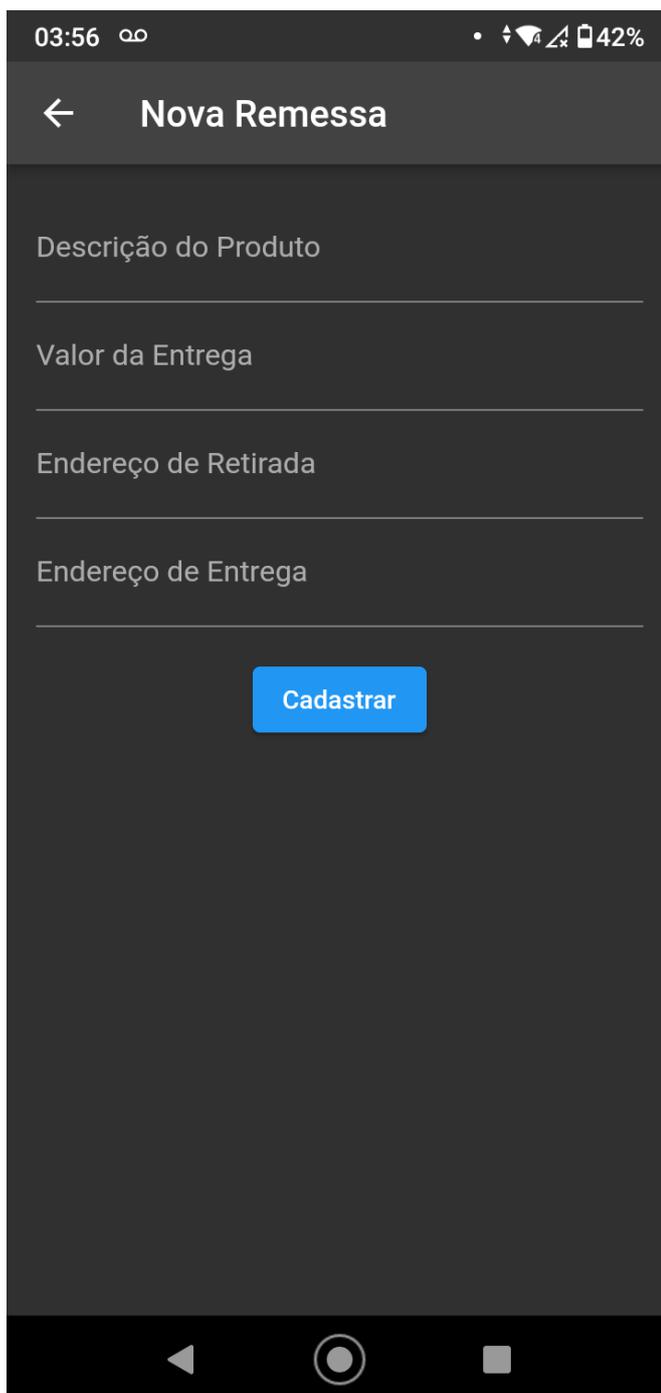
Figura 11 - Print de tela 5: Detalhes da entrega



Fonte - Autor, 2023

Na tela de nova remessa, acessada pelo *Floating Action Button* presente na tela principal, permite cadastrar uma nova entrega com as opções de inserir uma descrição, valor da entrega, endereço de retirada e endereço de entrega, conforme a Figura 12.

Figura 12 - Print de tela 6: Tela de nova remessa



03:56 42%

← Nova Remessa

Descrição do Produto

Valor da Entrega

Endereço de Retirada

Endereço de Entrega

Cadastrar

Este print de tela mostra a interface de usuário para a criação de uma nova remessa. O layout é vertical e utiliza um tema escuro. No topo, há uma barra de status com o horário (03:56) e o nível de bateria (42%). Abaixo, uma barra de cabeçalho contém um ícone de seta para trás e o título 'Nova Remessa'. O formulário principal possui quatro campos de entrada, cada um com um rótulo e uma linha de separação: 'Descrição do Produto', 'Valor da Entrega', 'Endereço de Retirada' e 'Endereço de Entrega'. Na base do formulário, há um botão azul com o texto 'Cadastrar'. O rodapé da tela exibe os ícones de navegação padrão do Android.

Fonte - Autor, 2023

#### 4.4.1 Arquitetura de Desenvolvimento

Devido à baixa complexidade tanto na estrutura proposta na análise do aplicativo como no desenvolvimento usando *Flutter* junto com o *Firebase*, não se fez necessário o uso de uma padronização de código, como por exemplo a MVC (*Model, View and Controller*) onde o código é separado na parte de *front-end*, operações de

*back-end* e operações com banco de dados. Parte disso se deve ao não desenvolvimento de uma API, mas sim ao uso do *Firebase* como um *Storage*.

Para a organização do código, foi definido que a maneira mais simples de controlar a organização seria definir cada tela como uma classe *Dart* e na própria classe, criar métodos para simplificar o entendimento do código. Como o método *main()* apenas chama a classe *Login()*, essa é a tela que será exibida ao abrir o aplicativo. Exemplificando, ainda usando a classe *Login()* como exemplo, no código desta há a declaração da classe em si, que se dá por um *Stateless Widget* e seu conteúdo dentro de um *Widget* chamado *Scaffold*, como pode ser observado na Figura 13. O *Scaffold* contém apenas chamados para os métodos que exibem a imagem do logo e campos de login, esse mesmo estilo de programação foi utilizado em todo o código.

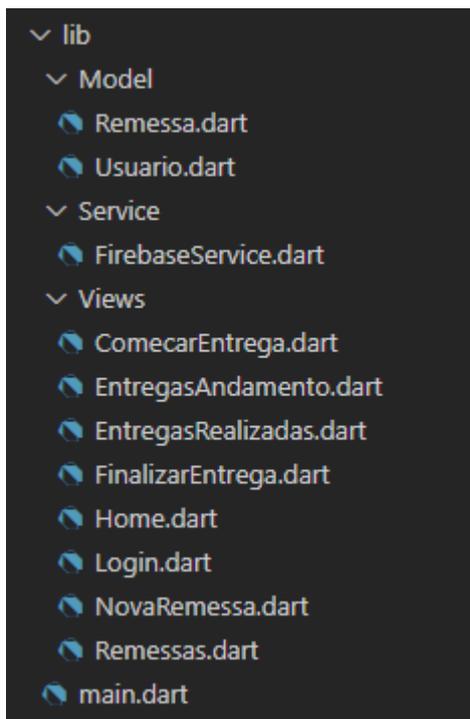
Figura 13 - Trecho de código página de Login

```
4  class LoginPage extends StatelessWidget {
5
6      @override
7      Widget build(BuildContext context) {
8          return Scaffold(
9              appBar: AppBar(
10                 title: Text('Login'),
11             ),
12             // aqui se inclui todo o resto do código da classe,
13             // como declaração de variáveis, métodos e os Widgets
```

Fonte - Autor, 2023

Na Figura 14, pode-se observar a estrutura de arquivos que compõem a pasta *lib*. Essa é a pasta padrão de um projeto *flutter* para os arquivos de classes. A pasta *lib* foi dividida em outras três pastas e contém um arquivo em sua raiz, sendo respectivamente as pastas “*Model*”, “*Service*”, “*Views*” e o arquivo *main*.

Figura 14 - Classes diretório Lib



Fonte - Autor, 2023

O arquivo *main*, localizado na raiz da pasta lib, contém o método principal para iniciar o aplicativo. O código contido no arquivo *main* faz uma chamada para abrir a tela login, que dessa forma se identifica como página principal do aplicativo, conforme a Figura 15.

Figura 15 - Código do arquivo *main*

```
6 void main() async {
7   WidgetsFlutterBinding.ensureInitialized();
8   await Firebase.initializeApp();
9   runApp(MaterialApp(
10    debugShowCheckedModeBanner: false,
11    home: LoginPage(), //chamada para página de login
12    theme: ThemeData(
13      brightness: Brightness.dark,
14    ),
15  ));
16 }
```

Fonte - Autor, 2023

Quando o usuário é autenticado pela página de *login*, o mesmo é redirecionado para a página *Home*, identificada pelo arquivo *home.dart*. O presente trabalho, por se tratar de um protótipo, não possui um método de autenticação de *login*, sendo assim, não sendo necessário preencher os campos de usuário e senha para efetuar o *login*. No que diz respeito a estrutura de dados, foi usado a forma que o próprio *Firebase*

disponibiliza, a de banco orientada a documentos, onde há níveis de hierarquia entre coleção, documento e campos.

## CONCLUSÃO

O presente trabalho teve como objetivo desenvolver uma plataforma online para atendimento logístico que pudesse auxiliar o comércio varejista a atender suas demandas de entregas de forma rápida e com baixo custo, com o objetivo de criar um aplicativo de celular conectado a um banco de dados em tempo real, que fosse capaz de conectar comerciante e entregador.

O uso combinado do ambiente de desenvolvimento integrado Android Studio, das bibliotecas do framework *Flutter* e do banco de dados *Firebase* foi o bastante para criar a aplicação proposta. Ao analisar o progresso do trabalho em termos de simulações, os objetivos foram alcançados, uma vez que foi possível desenvolver o aplicativo com sucesso.

Com base nas informações e resultados obtidos, a utilização de uma plataforma *online* para atendimento logístico ao comércio varejista se prova uma forma eficiente do comerciante realizar as entregas de seus produtos, pois elimina a necessidade de o próprio comerciante deixar o estabelecimento para realizar a entrega ou a necessidade de ligar para um entregador, tornando assim o processo mais eficiente e de baixo custo.

As informações apresentadas por este trabalho, podem ser utilizadas para novas pesquisas, onde pode ser elencado duas propostas de trabalhos futuros. Sendo essas, a disponibilização do aplicativo nas lojas *App Store* e *Play Store* e a implementação de um mapa com localização em tempo real, permitindo visualizar a rota até o destino.

## REFERÊNCIAS

- ALDARMI, S. A. 1998. **Real-Time Database Systems: concepts and design**. Department of Computer Science, The University of York, Disponível em: <https://pdfs.semanticscholar.org/9409/7bf9f885d83e6e2373f85801dd898d805b20.pdf>. Acesso em: 25 mai. 2023.
- ANDROID Developers: **Arquitetura da plataforma**. In: Android Developers: Arquitetura da plataforma. [S. l.], 5 abr. 2023. Disponível em: <https://developer.android.com/guide/platform>. Acesso em: 10 abr. 2023.
- ARBACHE, Fernando S. **Gestão de logística, distribuição e trade marketing**. 3. ed. Rio de Janeiro: FGV, 2004.
- BALLOU, Ronald H. **Gerenciamento da cadeia de suprimentos/logística empresarial**. [São Paulo - SP]: Grupo A, 2006. E-book. ISBN 9788560031467. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788560031467/>. Acesso em: 14 out. 2022.
- BAYER, F. M.; ECKHARDT, M.; MACHADO, R. **Automação de Sistemas**. Colégio Técnico Industrial - UFSM, 2011. Disponível em: <https://professor.pucgoias.edu.br/SiteDocente/admin/arquivosUpload/18451/material/Apostila%202.pdf>. Acesso em: 7 nov. 2022.
- BOWERSOX, Donald J.; CLOSS, David J.; COOPER, M B.; et al. **Gestão logística da cadeia de suprimentos**. [São Paulo - SP]: Grupo A, 2014. E-book. ISBN 9788580553185. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788580553185/>. Acesso em: 20 out. 2022.
- CARDOSO, A. 2018. **Firestore – banco de dados em tempo real**. Disponível em: <https://andrebian.com/firebase-banco-de-dados-em-tempo-real>. Acesso em: 27 mai. 2023.
- CHENG, F. 2017. **Build Mobile Apps with Ionic 2 and Firestore: hybrid mobile app development**. Estados Unidos: Apress. ISBN 9781484227374.
- CORREA, João Ederson; Mello, Carlos Henrique Pereira. **Efficient Consumer Response (ecr) e a tecnologia de armazenagem, movimentação de materiais e automação logística**. Revista Científica e-Locução, 2014, Disponível em: <https://periodicos.faex.edu.br/index.php/e-Locucacao/article/view/53/36>. Acesso em: 7 nov. 2022.
- DATE, C.J. **Introdução a Sistemas de Bancos de Dados**. [Rio de Janeiro - RJ]: Grupo GEN, 2004. E-book. ISBN 9788595154322. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788595154322/>. Acesso em: 18 mai. 2023.
- ELMASRI, R.; NAVATHE, S. 2011. **Sistemas de banco de dados. 6a edição**. Texas, Arlington: Editora Pearson. ISBN 9788579360855.

FARIA, A. C.; ROBLES, L. T.; BIO, S. R. **Custos logísticos: discussão sob uma ótica diferenciada**. In: CONGRESSO BRASILEIRO DE CUSTOS, 11, 2004, Bahia. Anais eletrônicos... Bahia: ABC, 2004. Disponível em: <<https://anaiscbc.emnuvens.com.br/anais/article/view/2284/2284>>. Acesso em: 20 out. 2022.

FIREBASE. 2023. **Entender os projetos do Firebase**. Disponível em: <https://firebase.google.com/docs/projects/learn-more?hl=pt-br>. Acesso em: 27 mai. 2023.

FLUTTER. 2023. **Flutter architectural overview**. Disponível em: <https://docs.flutter.dev/resources/architectural-overview>. Acesso em: 5 mai. 2023.

GIL, Antonio C. **Como Elaborar Projetos de Pesquisa**. [Rio de Janeiro - RJ]: Grupo GEN, 2022. E-book. ISBN 9786559771653. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9786559771653/>. Acesso em: 21 mai. 2023.

GIACOMELLI, Giancarlo; PIRES, Marcelo R S. **Logística e distribuição**. [São Paulo - SP]: Grupo A, 2016. E-book. ISBN 9788569726937. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788569726937/>. Acesso em: 20 out. 2022.

GUIMARÃES, L.S; et al. **A importância da logística para o desenvolvimento do e-commerce: uma análise documental**. Simpósio de Excelência em Gestão e Tecnologia - UNIS-MG, 2015. Disponível em: <https://www.aedb.br/seget/arquivos/artigos15/9122227.pdf>. Acesso em: 30 out. 2022.

GOOGLE. 2023. **Visão geral do Google Cloud**. Disponível em: <https://cloud.google.com/docs/overview/?hl=pt-brprojects>. Acesso em: 27 mai. 2023.

HEITKÖTTER, H.; HANSCHKE, S.; MAJCHRZAK, T. A. **Evaluating cross-platform development approaches for mobile applications**. In: INTERNATIONAL CONFERENCE ON WEB INFORMATION SYSTEMS AND TECHNOLOGIES, 8., 2012, Porto. Proceedings [...]. Porto: Webist, 2012. p. 120–138. Disponível em: [https://www3.nd.edu/~cpoellab/teaching/cse40814\\_fall14/crossplatform.pdf](https://www3.nd.edu/~cpoellab/teaching/cse40814_fall14/crossplatform.pdf). Acesso em: 10 dez. 2022.

LÓSCIO, B. F.; OLIVEIRA, H. R.; PONTES, J. C. 2015. **NoSQL no desenvolvimento de aplicações Web**. Disponível em: <https://docplayer.com.br/433576-Nosql-no-desenvolvimento-de-aplicacoes-web-colaborativas.html>. Acesso em: 25 mai. 2023.

MADUREIRA, Daniel. **Aplicativo nativo, web App ou aplicativo híbrido**. Net.8 mar. 2017. Disponível em: <<https://usemobile.com.br/aplicativo-nativo-web-hibrido/>>. Acesso em: 10 dez. 2022.

MAINKAR, P.; GIORDANO, S. 2019. **Google Flutter Mobile Development Quick Start Guide**. Estados Unidos: Packt Publishing. ISBN 9781789344967.

MARCONI, Marina de A.; LAKATOS, Eva M. **Metodologia Científica**. [Rio de Janeiro - RJ]: Grupo GEN, 2022. E-book. ISBN 9786559770670. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9786559770670/>. Acesso em: 21 mar. 2023.

NOVAES, Antonio G. **Logística e Gerenciamento da Cadeia de Distribuição - Estratégia, Avaliação e Operação**. [Rio de Janeiro - RJ]: Grupo GEN, 2021. E-book. ISBN 9788595157217. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788595157217/>. Acesso em: 14 out. 2022.

PATTAT, Tahinan; **E-COMMERCE: Comportamento do consumidor frente às transações online**. Universidade Regional do Noroeste do Estado do Rio Grande do Sul - UNIJUÍ, 2014. Disponível em: <https://bibliodigital.unijui.edu.br:8443/xmlui/bitstream/handle/123456789/2842/TC%20%20-%202014%20-%20Tahinan%20Pattat.pdf?sequence=1&isAllowed=y>. Acesso em: 30 out. 2022.

PELLINI, E. L. **Introdução à automação de sistemas elétricos - Conceitos Básico**. LPROT/ PEA/ EPUSP, 2017. Disponível em: [https://edisciplinas.usp.br/mod\\_folder/content](https://edisciplinas.usp.br/mod_folder/content). Acesso em: 7 nov. 2022.

PERKUSICH, M.; TURNELL, M.; PERKUSICH, A. 2018. **Modelagem de Banco de Dados em Tempo Real**. Departamento de Engenharia Elétrica, Universidade Federal da Paraíba, Disponível em: <https://xdocz.com.br/doc/modelagem-de-banco-de-dados-em-tempo-real-qzo27ewvymnm>. Acesso em: 25 mai. 2023.

RODRIGUES, S. 2007. **Apostila de técnicas de programação e linguagem Pascal**. UFRJ, Universidade Federal do Rio de Janeiro, Disponível em: <https://www.cos.ufrj.br/~sergio/ApostilaPascal.pdf>. Acesso em: 20 mar. 2023.

SILVA, Barbara Alyne E.; STETTINER, Caio F.; CAXITO, Fabiano de A. **Logística - um enfoque prático - 3ED**. [Goiânia, GO]: Editora Saraiva, 2019. E-book. ISBN 9788571440043. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788571440043/>. Acesso em: 14 out. 2022.

TEIXEIRA, Tarcísio. **Comércio Eletrônico - conforme o marco civil da internet e a regulamentação do e-commerce no Brasil**, 1ª edição.. [São Paulo- SP]: Editora Saraiva, 2015. E-book. ISBN 9788502622494. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788502622494/>. Acesso em: 30 out. 2022.

TENENBAUM, Anderson S; BOS, Herrbert. **SISTEMAS OPERACIONAIS MODERNOS**. 4. ed. São Paulo: Pearson Education, 2016. 778 p. Disponível em: [http://ldemetrio.com.br/Livros/Livros\\_TI/segunda\\_unid/Sistemas%20Operacionais%20Modernos%20-%20Tanenbaum%20-%204%20Edição.pdf](http://ldemetrio.com.br/Livros/Livros_TI/segunda_unid/Sistemas%20Operacionais%20Modernos%20-%20Tanenbaum%20-%204%20Edição.pdf). Acesso em: 20 mar. 2023.

VENTEU, K. C.; PINTO, G. S. **DESENVOLVIMENTO MÓVEL HÍBRIDO**. Revista Interface Tecnológica, [S. l.], v. 15, n. 1, p. 86–96, 2018. DOI: 10.31510/infa.v15i1.337. Disponível em: <https://revista.fatectq.edu.br/interfacetecnologica/article/view/337>. Acesso em: 10 abr. 2023.

VILLELA, P. S. C; VIDAL, F. J. T. **Automação Industrial**. LECA - DCA - Universidade Federal do Rio Grande do Norte, 2003. Disponível em: [https://www.dca.ufrn.br/~affonso/FTP/DCA447/trabalho1/trabalho1\\_19.pdf](https://www.dca.ufrn.br/~affonso/FTP/DCA447/trabalho1/trabalho1_19.pdf). Acesso em: 7 nov. 2022.

WALRATH, K.; LADD, S. 2012. **What is Dart?** Sebastopol, Canada: O'REILLY MEDIA. Disponível em: <http://radar.oreilly.com/2012/03/what-is-dart.html>. Acesso em: 10 abr. 2023.