

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
ESCOLA POLITÉCNICA
GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO



GABRIEL MENDES CRUZ

**DESENVOLVIMENTO DE UM APLICATIVO PARA
GESTÃO DE VENDAS: ESTUDO DE CASO EM VENDAS DE CALÇADOS**

Goiânia,
2022

GABRIEL MENDES CRUZ

**DESENVOLVIMENTO DE UM APLICATIVO PARA
GESTÃO DE VENDAS: ESTUDO DE CASO EM VENDAS DE CALÇADOS**

Trabalho de Conclusão de Curso
apresentada à Escola Politécnica,
Pontifícia Universidade Católica de Goiás,
como parte dos requisitos à obtenção do
grau de Bacharel em Engenharia de
Computação.

Professor(a) orientador(a):

Profa. Ma. Ludmilla Reis Pinheiro dos
Santos

Banca examinadora:

Profa. Dra. Carmen Cecília Centeno
Prof. Dr. José Luiz de Freitas Júnior

Goiânia
2022

RESUMO

A crescente evolução da internet fez com que o uso de dispositivos móveis e *smartphones* ou *tablets* se tornasse uma revolução que proporciona às pessoas a experiência do uso destes aparelhos em sua área profissional, que combinado com *softwares* e ferramentas se tornaram indispensáveis para uma produção efetiva. Uma área de experiência do uso destes dispositivos é a de vendas, a qual vem se destacando, desde 2017, na economia brasileira. Ainda assim, muitos profissionais dessa área possuem dificuldades que limitam o crescimento de seu negócio, como o excesso de burocracia, a ausência de um planejamento e a falta de um sistema de gestão de vendas. Desse modo, este trabalho teve como foco a resolução do último problema citado, realizando a implementação de um aplicativo que contemple as principais rotinas de vendas, garantindo a integração, comunicação e padronização de seus processos. Para o desenvolvimento foram utilizados o *framework Flutter* para a construção das interfaces, linguagem de programação *Dart* e banco de dados *Firebase* para armazenamento dos dados. O aplicativo permitiu o gerenciamento dos processos de venda, controle de estoque, finanças, contatos e relatórios.

Palavras-Chave: *Flutter, Dart, Firebase, Gestão de vendas*

ABSTRACT

The Internet's growing evolution has made mobile devices and smartphones or tablets a revolution that provides people with the experience of using these devices in their professional area, which, combined with software and tools, have become indispensable for effective production. One area of expertise in using these devices is sales, which has stood out, since 2017, in the Brazilian economy. However, many professionals in this area suffer difficulties that limit the growth of their business, such as excessive bureaucracy, lack of planning, and lack of a sales management system. Thus, this work focuses on solving the last problem mentioned, implementing an application that includes the main sales routines, and ensuring the integration, communication, and standardization of its processes. For the development, the Flutter framework was used to build the interfaces, Dart programming language, and Firebase database for data storage. The application allows the management of sales processes and control of stock, finances, contacts, and reports.

Palavras-Chave: *Flutter, Dart, Firebase, Donanja, Management, Sales*

LISTA DE FIGURAS

Figura 1 – Diagrama de casos de uso.....	26
Figura 2 – Arquitetura do Flutter.....	42
Figura 3 – Serviços do <i>Firebase</i>	44
Figura 4 – Interface <i>Cloud Firestore</i>	45
Figura 5 – Interface do Visual Studio Code	46
Figura 6 – Tela de <i>Login</i>	47
Figura 7 – Tela de Cadastro.....	49
Figura 8 – Tela de envio de e-mail	50
Figura 9 – E-mail do processo de redefinição de senha	50
Figura 10 – Aviso de e-mail não cadastrado	51
Figura 11 – Interface de redefinição de senha	52
Figura 12 – Tela Home.....	53
Figura 13 – Barra de menu lateral.....	54
Figura 14 – Tela de Contatos	55
Figura 15 – Tela de pesquisa (a) e filtro dos contatos(b)	56
Figura 16 – Tela de Cadastro de Contato	57
Figura 17 – Tela de Edição de Contato.....	58
Figura 18 – Tela de Estoque	59
Figura 19 – Tela de Cadastro de Produto	61
Figura 20 – Tela de Edição do Produto.....	62
Figura 21 – Tela de Pedidos	63
Figura 22 – Tela de Pedidos Filtrados.....	64
Figura 23 – Tela de Criação de Pedido.....	65
Figura 24 – Tela de Confirmação de Pedido.....	66
Figura 25 – Erro exibido ao tentar adicionar pedido vazio	67
Figura 26 – Tela de Ficha Financeira.....	68

LISTA DE QUADROS

Quadro 1 – Requisitos Usuários - Necessidades	17
Quadro 2 – RF 001: Manter usuário	18
Quadro 3 – RF 002: Realizar login	18
Quadro 4 – RF 003: Recuperar senha	19
Quadro 5 – RF 004: Manter produto	19
Quadro 6 – RF 005: Manter contato	20
Quadro 7 – RF 006: Visualizar estoque	20
Quadro 8 – RF 007: Definir pedido	21
Quadro 9 – RF 008: Cancelar pedido	21
Quadro 10 – RF 009: Visualizar pedido	22
Quadro 11 – RF 010: Manter ficha financeira	22
Quadro 12 – RF 011: Visualizar relatórios	23
Quadro 13 – RQ 001: Segurança	23
Quadro 14 – RQ 002: Responsividade	24
Quadro 15 – RQ 003: Confiabilidade	24
Quadro 16 – RQ 004: Compatibilidade	25
Quadro 17 – RQ 005: Disponibilidade	25
Quadro 18 – CSU 001: Cadastrar Usuário	27
Quadro 19 – CSU 002: Fazer Login	28
Quadro 20 – CSU 003: Recuperar senha	29
Quadro 21 – CSU 004: Manter produto	30
Quadro 22 – CSU 005: Manter contato	31
Quadro 23 – CSU 006: Visualizar estoque	32
Quadro 24 – CSU 007: Definir pedido	33
Quadro 25 – CSU 008: Cancelar pedido	34
Quadro 26 – CSU 009: Visualizar pedido	35
Quadro 27 – CSU 010: Manter ficha financeira	36
Quadro 28 – CSU 011: Visualizar relatórios	37
Quadro 29 – DD 001: Dados do usuário	38
Quadro 30 - DD 002: Dados do produto	38
Quadro 31 – DD 003: Dados do tamanho do produto	39
Quadro 32 – DD 004: Dados do contato	39
Quadro 33 – DD 005: Dados do pedido	40

LISTA DE SIGLAS

AOT	<i>Ahead-of-time</i>
API	<i>Application Programming Interface</i>
ARM	<i>Advanced RISC Machine</i>
BaaS	<i>Backend As A Service</i>
CSU	Caso de Uso
DD	Dicionário de Dados
IDE	<i>Integrated Development Environment</i>
JIT	<i>Just-in-time</i>
RF	Requisitos Funcionais
RQ	Requisitos de Qualidade
SDK	<i>Software Development Kit</i>

SUMÁRIO

1	INTRODUÇÃO.....	10
1.1	Objetivo geral	11
1.2	Objetivos específicos	11
1.3	Metodologias	11
1.4	Stakeholders	12
1.5	Organização do trabalho	12
2	DESCRIÇÃO GERAL	13
2.1	Aspecto geral do aplicativo	13
2.1.1	Interfaces do aplicativo	13
2.1.2	Interfaces do usuário	14
2.1.3	Interfaces de hardware	14
2.1.4	Interfaces de software	14
2.1.5	Interfaces de comunicação.....	15
2.2	Funções do aplicativo.....	15
2.3	Características dos usuários	15
2.4	Restrições/Limitações	16
2.5	Regras de negócio	16
3	DOCUMENTAÇÃO DO APLICATIVO.....	17
3.1	Requisitos de Usuário – Necessidades.....	17
3.2	Requisitos Funcionais	17
3.3	Requisitos de Qualidade	23
3.4	Diagrama de casos de uso.....	26
3.5	Casos de uso descritivos	27
3.6	Dicionário de dados do sistema	38
4	DESENVOLVIMENTO DO APLICATIVO.....	41
4.1	Flutter	41
4.1.1	Arquitetura	41
4.1.2	<i>Widgets</i>	42
4.2	Dart	43
4.3	<i>Firebase</i>	43
4.3.1	Autenticação do <i>Firebase</i>	44
4.3.1	<i>Firebase Cloud Firestore</i>	45
4.4	<i>Visual Studio Code</i>	45
5	IMPLEMENTAÇÕES E RESULTADOS.....	47

5.1	Telas do Aplicativo	47
5.1.1	Tela de Login	47
5.1.2	Tela de Cadastro	48
5.1.3	Tela de Redefinição de senha	49
5.1.4	Tela Home	52
5.1.5	Tela de Contatos	55
5.1.6	Tela de Cadastro de Contatos	57
5.1.7	Tela de Edição de Contatos	58
5.1.8	Tela de Estoque.....	59
5.1.9	Tela de Cadastro de Produtos	60
5.1.10	Tela de Edição de Produtos	61
5.1.11	Tela de Pedidos	62
5.1.12	Tela de Criação de Pedidos	64
5.1.13	Tela de Confirmação de Pedidos	66
5.1.14	Tela de Ficha Financeira	67
5.2	Otimização no controle dos recursos	69
6	CONSIDERAÇÕES FINAIS.....	70
6.1	Trabalhos futuros	70
	REFERÊNCIAS.....	71
	APÊNDICE A – TERMO DE AUTORIZAÇÃO DE PUBLICAÇÃO DE PRODUÇÃO ACADÊMICA	74

1 INTRODUÇÃO

Com a crescente evolução da internet e mídias sociais, o uso de dispositivos móveis, *smartphones* e *tablets* tem proporcionado uma revolução nos últimos tempos (SILVA; SANTOS, 2014).

Essa revolução tem proporcionado a experiência da utilização de dispositivos no trabalho cotidiano, com *softwares* e ferramentas operacionais que se mostraram indispensáveis para uma boa produção.

Micro e pequenas empresas são responsáveis por uma grande quantidade de giro econômico no país. Em 2017, um estudo elaborado pelo Serviço Brasileiro de Apoio às Micro e Pequenas Empresas (Sebrae) e Fundação Getúlio Vargas (FGV) mostrou uma crescente relevância das micro e pequenas empresas na economia brasileira, representando cerca de 30% do Produto Interno Bruto (PIB) do país. Essa porcentagem se manteve até o ano de 2020, segundo dados do Ministério da Economia, do Governo Federal (SEBRAE, 2018; SEPEC/ME, 2020).

O sociólogo e ex-presidente do Sebrae, Luiz Barretto, cita que os principais motivos para o bom desempenho são a melhoria do ambiente de negócios no ramo da tributação, aumento da escolaridade da população e a ampliação do mercado consumidor, com o crescimento da classe média (SEBRAE, 2018).

“Os dados demonstram a importância de incentivar e qualificar os empreendimentos de menor porte, inclusive os Microempreendedores Individuais. Isoladamente, uma empresa representa pouco. Mas juntas, elas são decisivas para a economia” (BARRETTO, 2018).

Entretanto, muitas dessas empresas enfrentam dificuldades que freiam e atrapalham o crescimento. Entre os principais problemas, estão o excesso de burocracia para a obtenção de créditos, a ausência de planejamento e a falta de um sistema de gestão (VOLPI, 2018).

Justifica-se estudar este tema porque, com a análise desse problema, é importante garantir que o sistema contemple as principais rotinas de vendas, garantindo a integração, comunicação e padronização dos processos, mais precisamente ao varejo.

1.1 Objetivo geral

- Desenvolver um aplicativo multiplataforma para gerenciar e estruturar melhorias de controle dos recursos, e integração dos processos de vendas no varejo. Como estudo de caso será aplicada à venda de calçados femininos.

1.2 Objetivos específicos

- Desenvolver um aplicativo para gerenciamento e estruturação do controle de estoque, finanças, clientes e relatórios;
- Estruturar melhorias dos processos operacionais;
- Integrar os processos de venda no varejo entre os ambientes físicos e digitais;
- Desenvolver um aplicativo que se adeque a mais de uma área de venda, entretanto o estudo de caso deste trabalho será aplicado à venda de calçados femininos;
- Estudo da linguagem Dart e *framework* Flutter;
- Desenvolver um aplicativo multiplataforma com design simples e com foco na usabilidade do usuário.

1.3 Metodologias

Este trabalho, segundo sua natureza, é uma pesquisa aplicada, onde o foco é a apresentação de conhecimentos a partir de observações e aplicações construídas a partir da problematização de um tema (WAZLAWICK, 2014). Será realizado um estudo de trabalhos já publicados sobre desenvolvimento de aplicativos.

Quanto aos objetivos, este trabalho caracteriza-se como exploratório, visto que a principal finalidade é desenvolver conceitos e ideias, tendo em vista a formulação de problemas mais precisos ou hipóteses pesquisáveis para estudos posteriores (WAZLAWICK, 2014).

Quanto aos procedimentos técnicos, utilizará pesquisas bibliográficas e experimentais. As pesquisas bibliográficas implicarão no estudo de artigos, teses, livros e entrevistas. Além disso, o trabalho se apresenta como experimental aos procedimentos técnicos, uma vez que é relacionado ao desenvolvimento de um aplicativo, manipulando um aspecto da realidade do trabalho (WAZLAWICK, 2014).

1.4 Stakeholders

Os *stakeholders* deste projeto incluem os microempresários individuais que atuam na área de venda e profissionais de venda autônomos.

1.5 Organização do trabalho

Este trabalho foi organizado em seis capítulos. O capítulo 1 foi composto pela introdução, objetivos gerais e específicos, metodologias e justificativa.

O capítulo 2 aborda a descrição geral do aplicativo, sendo apresentadas as interfaces do aplicativo, funções, características do usuário, restrições e regras de negócio.

O capítulo 3 apresenta a documentação utilizada para o desenvolvimento do aplicativo.

O capítulo 4 descreve as tecnologias e ferramentas utilizadas para o desenvolvimento do aplicativo.

O capítulo 5 mostra a implementação das telas do aplicativo juntamente com sua funcionalidade e os resultados obtidos pelo uso do aplicativo.

E por último, o capítulo 6 aborda as considerações finais deste trabalho, composta pelas dificuldades encontradas no desenvolvimento do mesmo e sugestão para trabalhos futuros.

2 DESCRIÇÃO GERAL

Este capítulo apresenta as interfaces do aplicativo, suas características e perfil dos usuários alvo.

2.1 Aspecto geral do aplicativo

O aplicativo Donanja Gestão de Vendas foi construído utilizando o framework *Flutter*, com o objetivo de otimizar e facilitar a gestão, integração e comunicação dos processos de venda, estruturando melhorias de controle de recursos.

2.1.1 Interfaces do aplicativo

O aplicativo é composto pelas seguintes telas:

- **Tela de cadastro do usuário:** permite ao usuário realizar o cadastro, informando dados como nome, e-mail e senha;
- **Tela de login:** permite ao usuário acessar o aplicativo informando o e-mail e a senha cadastrados anteriormente;
- **Tela de recuperação de senha:** permite ao usuário recuperar a senha, a partir do envio de processo para recuperação ao endereço de e-mail cadastrado;
- **Tela inicial:** permite ao usuário visualizar os relatórios de vendas realizadas e controle financeiro em forma de gráficos, e possui um menu lateral que exibe o nome e e-mail do usuário e as opções que poderão ser escolhidas para navegar para telas desejadas pelo usuário, e o resumo mensal de vendas;
- **Tela de cadastro dos produtos:** permite o cadastro de produtos disponíveis para venda. O cadastro deve conter o nome do produto, identificador, quantidade, valor e detalhes adicionais do produto;
- **Tela de cadastro de contatos (clientes e fornecedores):** permite ao usuário cadastrar contatos, pessoa física ou jurídica, vinculadas ao usuário, podendo classificar como cliente ou fornecedor, informando

dados como nome, CPF ou CNPJ, telefone, área de atuação, endereço, e detalhes adicionais do contato;

- **Tela de pedidos:** contém uma lista de painéis deslizantes que exibem informações básicas dos pedidos realizados, como identificador, status e valor total do pedido, e ao ser pressionado exibe informações adicionais como produtos, valores dos produtos, quantidade e tamanho dos produtos vinculados;
- **Tela de controle de estoques:** permite ao usuário visualizar os produtos cadastrados no aplicativo. Nela são exibidas informações, como identificador, nome, quantidade, preço, fornecedor vinculado e detalhes adicionais;
- **Tela de contatos:** permite ao usuário visualizar os contatos cadastrados no aplicativo. Nela são exibidas informações, como nome, telefone, endereço, tipo de pessoa, atuação e detalhes adicionais;
- **Tela de ficha financeira:** permite ao usuário visualizar toda a movimentação de caixa, tais como informações sobre entrada e saída dos valores, lucratividade sobre a venda e balanço financeiro;

2.1.2 Interfaces do usuário

O aplicativo foi construído por interfaces responsivas, podendo se adaptar a diferentes resoluções e tamanhos de telas, de diferentes dispositivos.

2.1.3 Interfaces de hardware

A instalação e execução de um aplicativo desenvolvido em *Flutter* é disponível para dispositivos móveis Android que possuam processador ARM, ou no caso de dispositivos iOS, iPhone 4S ou mais recente (FLUTTER, 2020).

2.1.4 Interfaces de software

O aplicativo poderá ser instalado em dispositivos Android que atenda a versão a partir do Android Jelly Bean, v16, 4.1.x, e possibilidade para iOS 9.0.

2.1.5 Interfaces de comunicação

O aplicativo utiliza o banco de dados Firebase para armazenamento dos dados cadastrados, como produtos, contatos, pedidos e informações dos usuários, além da autenticação de acesso.

2.2 Funções do aplicativo

O aplicativo pode auxiliar o profissional a partir de funções simples e visualizações fáceis de serem executadas, como:

- Realizar cadastro no aplicativo;
- Realizar cadastro de produtos;
- Realizar cadastro de contatos;
- Visualizar o estoque dos produtos;
- Criar e visualizar pedidos;
- Visualizar tabelas de contas a pagar e receber;
- Visualizar ficha financeira;
- Visualizar relatórios;
- Visualizar detalhes de um pedido;

2.3 Características dos usuários

O presente trabalho tem em vista a otimização na gestão comercial e administração dos recursos destes profissionais por meio do aplicativo Donanja Gestão de Vendas, sendo que os usuários alvo do aplicativo são os Microempresários Individuais.

O termo Microempresário Individual (MEI) foi criado em 2009, com o objetivo de tirar da informalidade os profissionais autônomos e pequenos empreendedores. É um tipo de empresa simples e que se ajusta bem às necessidades de quem atua de forma autônoma, com próprio CNPJ, emissão de notas fiscais e todos os direitos à Previdência Social (TORRES, 2020).

Essa natureza jurídica foi criada com o objetivo de facilitar a regularização das atividades econômicas de pessoas que trabalham por conta própria e que não têm sócios.

2.4 Restrições/Limitações

O aplicativo possui as seguintes restrições:

- Só pode ser acessado quando conectado à internet para utilização da autenticação e do banco de dados Firebase;
- O usuário deve estar logado para ter acesso às funcionalidades do aplicativo.

2.5 Regras de negócio

O aplicativo possui as seguintes regras de negócio:

- O status de um pedido será alterado de acordo com a situação do seu pagamento, sendo possível 4 estados: realizado, preparando entrega, confirmado e cancelado;
- Só é possível cancelar um pedido se ele ainda não estiver pago.

3 DOCUMENTAÇÃO DO APLICATIVO

Este capítulo apresenta a documentação usada para o desenvolvimento do aplicativo, tais como: Requisitos de Usuários – Necessidades (RUN), Requisitos Funcionais (RF), Requisitos de Qualidade (RQ) e Casos de Uso Descritivos (CSU).

3.1 Requisitos de Usuário – Necessidades

Esta seção apresenta os requisitos de usuário – necessidades, definidos no processo de levantamento de requisitos.

Quadro 1 – Requisitos Usuários - Necessidades

ID	Descrição	Fonte
RUN 001	O usuário deve ser capaz de realizar cadastro utilizando e-mail e senhas válidos.	Autoria própria
RUN 002	O usuário deve ser capaz de realizar login.	Autoria própria
RUN 003	O usuário deve ser capaz de recuperar a senha.	Autoria própria
RUN 004	O usuário deve ser capaz de cadastrar produtos.	Autoria própria
RUN 005	O usuário deve ser capaz de cadastrar contatos.	Autoria própria
RUN 006	O usuário deve ser capaz de visualizar estoque.	Autoria própria
RUN 007	O usuário deve ser capaz de fazer um pedido.	Autoria própria
RUN 008	O usuário deve ser capaz de visualizar informações dos pedidos realizados.	Autoria própria
RUN 009	O usuário deve ser capaz de visualizar relatórios.	Autoria própria
RUN 010	O usuário deve ser capaz de visualizar tabelas financeiras.	Autoria própria

Fonte: Elaborado pelo autor

3.2 Requisitos Funcionais

Nos quadros desta seção serão exibidos os Requisitos Funcionais do aplicativo, definidos no processo de Elicitação de requisitos.

Quadro 2 – RF 001: Manter usuário

Identificador	Nome
RF 001	Manter Usuário
Caso de Uso	Autor
CSU 001	Gabriel Mendes Cruz
Descrição	
O aplicativo deve permitir o cadastro e edição do usuário, uma vez informados e-mail e senha. As informações deverão ser armazenadas no banco de dados.	
Critério de Verificação	
<ul style="list-style-type: none"> • Verificar se os campos foram preenchidos; • Verificar se já existe um usuário cadastrado com o e-mail informado. 	
Dependência	Prioridade
RUN 001	Essencial

Fonte: Elaborado pelo autor

Quadro 3 – RF 002: Realizar login

Identificador	Nome
RF 002	Realizar login
Caso de Uso	Autor
CSU 002	Gabriel Mendes Cruz
Descrição	
O aplicativo deve permitir o login de usuários já cadastrados. A autenticação deve ser realizada por e-mail e senha. O usuário também terá a opção de entrar com Gmail.	
Critério de Verificação	
<ul style="list-style-type: none"> • Validar se os campos foram preenchidos; • Validar se existe um cadastro vinculado ao e-mail informado e senha informados. 	
Dependência	Prioridade
RUN 002	Essencial

Fonte: Elaborado pelo autor

Quadro 4 – RF 003: Recuperar senha

Identificador	Nome
RF 003	Recuperar senha
Caso de Uso	Autor
CSU 003	Gabriel Mendes Cruz
Descrição	
O aplicativo deve permitir a recuperação de senha do usuário. O processo é enviado para o e-mail cadastrado.	
Critério de Verificação	
<ul style="list-style-type: none"> Validar se existe um cadastro vinculado ao e-mail informado. 	
Dependência	Prioridade
RUN 003, RF 001	Essencial

Fonte: Elaborado pelo autor

Quadro 5 – RF 004: Manter produto

Identificador	Nome
RF 004	Manter produto
Caso de Uso	Autor
CSU 004	Gabriel Mendes Cruz
Descrição	
O aplicativo deve permitir o usuário manter (criar, listar, editar e excluir) um produto.	
Critério de Verificação	
<ul style="list-style-type: none"> Validar se os campos foram preenchidos; 	
Dependência	Prioridade
RUN 004	Essencial

Fonte: Elaborado pelo autor

Quadro 6 – RF 005: Manter contato.

Identificador	Nome
RF 005	Manter contato
Caso de Uso	Autor
CSU 005	Gabriel Mendes Cruz
Descrição	
O aplicativo deve permitir o usuário manter (criar, listar, editar e excluir) um contato, podendo classificar como cliente ou fornecedor, e como pessoa física ou jurídica.	
Critério de Verificação	
<ul style="list-style-type: none"> Validar se ao criar um pedido é gerado um identificador único. Validar a existência de uma pessoa de acordo com seu cadastro junto à Receita Federal (CPF/CNPJ). 	
Dependência	Prioridade
RUN 005	Essencial

Fonte: Elaborado pelo autor

Quadro 7 – RF 006: Visualizar estoque

Identificador	Nome
RF 006	Visualizar estoque
Caso de Uso	Autor
CSU 006	Gabriel Mendes Cruz
Descrição	
O aplicativo deve permitir o usuário consultar o estoque dos produtos já cadastrados.	
Critério de Verificação	
<ul style="list-style-type: none"> Validar a existência de produtos cadastrados no aplicativo. 	
Dependência	Prioridade
RUN 006, RF 004	Essencial

Fonte: Elaborado pelo autor

Quadro 8 – RF 007: Definir pedido

Identificador	Nome
RF 007	Definir pedido
Caso de Uso	Autor
CSU 007	Gabriel Mendes Cruz
Descrição	
O aplicativo deve permitir o usuário definir um pedido.	
Critério de Verificação	
<ul style="list-style-type: none"> • Validar se os campos foram preenchidos; • Validar a existência de cliente e produto para a confecção do pedido; • Validar se ao criar um pedido é gerado um identificador único. 	
Dependência	Prioridade
RUN 007, RF 004, RF 005	Essencial

Fonte: Elaborado pelo autor

Quadro 9 – RF 008: Cancelar pedido

Identificador	Nome
RF 008	Cancelar pedido
Caso de Uso	Autor
CSU 008	Gabriel Mendes Cruz
Descrição	
O aplicativo deve permitir o usuário cancelar um pedido.	
Critério de Verificação	
<ul style="list-style-type: none"> • Validar a existência do pedido. 	
Dependência	Prioridade
RUN 007, RF 004, RF 005	Essencial

Fonte: Elaborado pelo autor

Quadro 10 – RF 009: Visualizar pedido

Identificador	Nome
RF 009	Visualizar pedido
Caso de Uso	Autor
CSU 009	Gabriel Mendes Cruz
Descrição	
O aplicativo deve permitir o usuário consultar os pedidos realizados.	
Critério de Verificação	
<ul style="list-style-type: none"> Validar a existência de pedidos definidos no aplicativo. 	
Dependência	Prioridade
RUN 008, RF 007	Essencial

Fonte: Elaborado pelo autor

Quadro 11 – RF 010: Manter ficha financeira

Identificador	Nome
RF 010	Visualizar ficha financeira
Caso de Uso	Autor
CSU 010	Gabriel Mendes Cruz
Descrição	
O aplicativo deve permitir o usuário adicionar, editar, remover e consultar dados de tabelas financeiras do aplicativo, dentre elas as telas de pagamento para fornecedores e recebimento de clientes.	
Critério de Verificação	
<ul style="list-style-type: none"> Validar a existência de entradas e saídas sobre os pedidos feitos. 	
Dependência	Prioridade
RUN 008, RF 007	Essencial

Fonte: Elaborado pelo autor

Quadro 12 – RF 011: Visualizar relatórios

Identificador	Nome
RF 011	Visualizar relatórios
Caso de Uso	Autor
CSU 011	Gabriel Mendes Cruz
Descrição	
O aplicativo deve permitir o usuário consultar os relatórios de estoque, finanças e pedidos, no período dos 5 meses mais recentes.	
Critério de Verificação	
•	
Dependência	Prioridade
	Essencial

Fonte: Elaborado pelo autor

3.3 Requisitos de Qualidade

Nos quadros desta seção serão exibidos os requisitos de qualidade do aplicativo, definidos no processo de Elicitação de requisitos.

Quadro 13 – RQ 001: Segurança

Identificador	Nome
RQ 001	Segurança
Caso de Uso	Autor
	Gabriel Mendes Cruz
Descrição	
O aplicativo deve garantir segurança no acesso, salvando as senhas dos usuários utilizando <i>hash</i> .	
Critério de Verificação	
• Validar a <i>hash</i> salva pelo Firebase.	

Dependência	Prioridade
	Essencial

Fonte: Elaborado pelo autor

Quadro 14 – RQ 002: Responsividade

Identificador	Nome
RQ 002	Responsividade
Caso de Uso	Autor
	Gabriel Mendes Cruz
Descrição	
O aplicativo deve permitir a adaptação a diferentes resoluções e tamanhos de telas.	
Critério de Verificação	
<ul style="list-style-type: none"> Validar o comportamento do aplicativo em dispositivos com telas e resoluções diferentes. 	
Dependência	Prioridade
	Essencial

Fonte: Elaborado pelo autor

Quadro 15 – RQ 003: Confiabilidade

Identificador	Nome
RQ 003	Confiabilidade
Caso de Uso	Autor
	Gabriel Mendes Cruz
Descrição	
O aplicativo deve garantir um servidor confiável com suporte a usuários logados simultaneamente.	

Critério de Verificação	
<ul style="list-style-type: none"> Validar o comportamento do servidor através de teste de estresse. 	
Dependência	Prioridade
	Essencial

Fonte: Elaborado pelo autor

Quadro 16 – RQ 004: Compatibilidade

Identificador	Nome
RQ 004	Compatibilidade
Caso de Uso	Autor
	Gabriel Mendes Cruz
Descrição	
O aplicativo deve funcionar em dispositivos com sistemas operacionais Android e iOS.	
Critério de Verificação	
Dependência	Prioridade
	Essencial

Fonte: Elaborado pelo autor

Quadro 17 – RQ 005: Disponibilidade

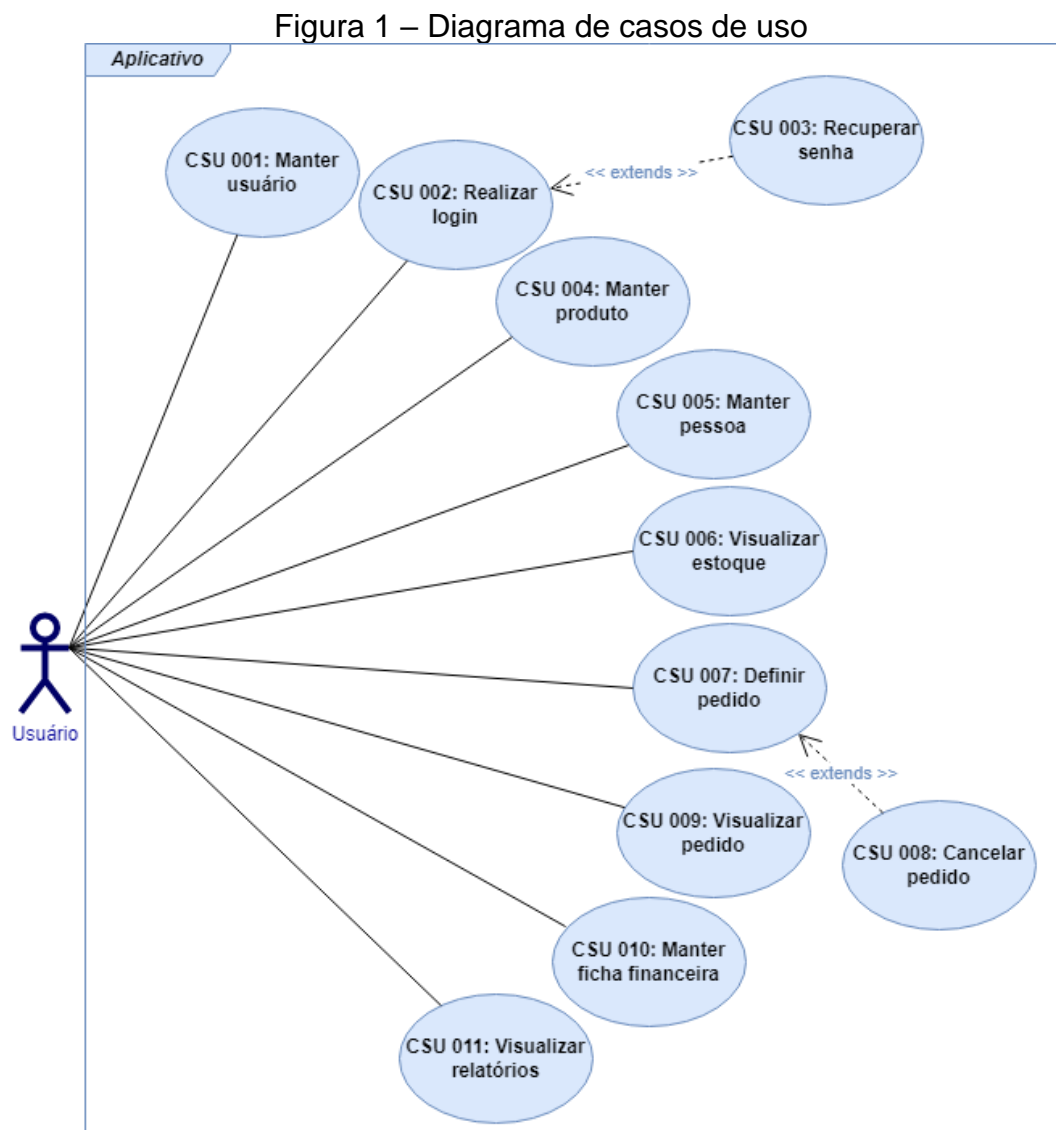
Identificador	Nome
RQ 004	Disponibilidade
Caso de Uso	Autor
	Gabriel Mendes Cruz
Descrição	
O aplicativo deve estar disponível sob a condição de conexão com a internet e com o banco de dados Firebase.	

Critério de Verificação	
Dependência	Prioridade
	Essencial

Fonte: Elaborado pelo autor

3.4 Diagrama de casos de uso

A Figura 1 apresenta o Diagrama de casos de uso que apresenta as funcionalidades do aplicativo:



Fonte: Elaborado pelo autor

3.5 Casos de uso descritivos

Esta seção apresenta a descrição dos casos de uso dados pela Figura 1.

Quadro 18 – CSU 001: Cadastrar Usuário.

Identificador	CSU 001
Nome	Cadastrar Usuário
Atores	Usuário
Requisitos	RF 001
Responsável	Gabriel Mendes Cruz
Descrição/Resumo	Este caso de uso descreve o processo para o cadastro de um usuário no aplicativo.
Pré-condições	
Pós-condições	<ul style="list-style-type: none"> • O usuário deve estar cadastrado no aplicativo.
Cenário Principal	<ol style="list-style-type: none"> 1. O usuário abre o aplicativo; 2. É exibida a tela de login; 3. O usuário seleciona a opção “Cadastrar”; 4. É exibida a tela de cadastro; 5. O usuário preenche os campos; 6. O usuário seleciona a opção “Cadastrar”; 7. O aplicativo valida os dados informados; 8. As informações são armazenadas no banco de dados; 9. O usuário é redirecionado para a tela inicial; 10. O caso de uso é finalizado.
Cenários alternativos	<p>(A01) Passo 5: O usuário cancela o cadastro</p> <p>A01.1 O usuário seleciona a opção “Voltar” ou fecha o aplicativo;</p> <p>A01.2 O aplicativo para o cadastro sem salvar;</p> <p>A01.3 É exibida a tela de login;</p> <p>A01.4 O aplicativo retorna ao passo 10 do cenário principal.</p>
Cenário de exceção	<p>(E01) Passo 7: O usuário informa um e-mail já cadastrado no sistema</p> <p>E01.1 O aplicativo apresenta a mensagem “E-mail já cadastrado”;</p> <p>E01.2 O aplicativo retorna ao passo 5.</p> <p>(E02) Passo 7: As senhas informadas pelo usuário são diferentes</p> <p>E02.1 O aplicativo apresenta a mensagem “As senhas informadas não conferem”;</p> <p>E02.2 O aplicativo retorna ao passo 5.</p>
Qualidades	RQ 001, RQ 005

Fonte: Elaborado pelo autor

Quadro 19 – CSU 002: Fazer Login.

Identificador	CSU 002
Nome	Fazer Login
Atores	Usuário
Requisitos	RF 002
Responsável	Gabriel Mendes Cruz
Descrição/Resumo	Este caso de uso descreve o processo para o login do usuário.
Pré-condições	<ul style="list-style-type: none"> • O usuário deve estar cadastrado no aplicativo.
Pós-condições	<ul style="list-style-type: none"> • O usuário deve estar logado no aplicativo.
Cenário Principal	<ol style="list-style-type: none"> 1. O usuário abre o aplicativo; 2. É exibida a tela de login; 3. O usuário preenche os campos de e-mail e senha; 4. O usuário seleciona a opção “Login”; 5. O aplicativo valida os dados informados; 6. O usuário é redirecionado para a tela inicial; 7. O caso de uso é finalizado.
Cenários alternativos	<p>(A01) Passo 3: O usuário cancela o login A01.1 O usuário fecha o aplicativo ou cancela o login; A01.2 O aplicativo retorna ao passo 7 do cenário principal.</p> <p>(A02) Passo 3: O usuário seleciona a opção de cadastro A02.1 O usuário seleciona a opção “Cadastrar”; A02.2 O aplicativo retorna ao passo 7 do cenário principal.</p> <p>(A03) Passo 3: O usuário seleciona a opção de entrar com Gmail A03.1 O usuário seleciona a opção “Entrar com Gmail”; A03.2 O aplicativo retorna ao passo 5 do cenário principal.</p>
Cenário de exceção	<p>(E01) Passo 5: O usuário informa um e-mail não cadastrado no sistema ou uma senha diferente E01.1 O aplicativo apresenta a mensagem “E-mail ou senha inválidos”; E01.2 O aplicativo retorna ao passo 2 do cenário principal.</p>
Qualidades	RQ 003

Fonte: Elaborado pelo autor

Quadro 20 – CSU 003: Recuperar senha.

Identificador	CSU 003
Nome	Recuperar senha
Atores	Usuário
Requisitos	RF 003
Responsável	Gabriel Mendes Cruz
Descrição/Resumo	Este caso de uso descreve o processo para recuperar senha do usuário.
Pré-condições	<ul style="list-style-type: none"> • O usuário deve estar cadastrado no aplicativo.
Pós-condições	<ul style="list-style-type: none"> • O usuário deve ter sua senha atualizada
Cenário Principal	<ol style="list-style-type: none"> 1. O usuário abre o aplicativo; 2. É exibida a tela de login; 3. O usuário seleciona a opção “Esqueci minha senha”; 4. É exibida a tela de recuperação de senha; 5. O usuário preenche os dados; 6. O usuário seleciona a opção “Confirmar”; 7. O aplicativo valida os dados informados; 8. O aplicativo envia um link para o e-mail cadastrado que permitirá a troca de senha; 9. O usuário acessa o link; 10. É exibida uma página web; 11. O usuário informa uma nova senha; 12. É exibida a mensagem: “Senha alterada com sucesso”. 13. O caso de uso é finalizado.
Cenários alternativos	<p>(A01) Passo 4: O usuário cancela a recuperação de senha</p> <p>A01.1 O usuário seleciona a opção “Voltar” ou fecha o aplicativo;</p> <p>A01.2 O aplicativo retorna ao passo 13 do cenário principal.</p> <p>(A02) Passo 8: O usuário desiste da recuperação de senha</p> <p>A02.1 O usuário não acessa o link enviado para o e-mail ou não informa a nova senha;</p> <p>A02.2 O aplicativo retorna ao passo 13 do cenário principal.</p>
Cenário de exceção	<p>(E01) Passo 7: O usuário informa um e-mail não está cadastrado no sistema</p> <p>E01.1 O aplicativo apresenta a mensagem “E-mail não encontrado”;</p> <p>E01.2 O aplicativo retorna ao passo 5 do cenário principal.</p>

	(E02) Passo 7: As senhas informadas pelo usuário são diferentes E02.1 O aplicativo apresenta a mensagem “As senhas informadas não conferem”; E02.2 O aplicativo retorna ao passo 5 do cenário principal
Qualidades	RQ 001, RQ 005

Fonte: Elaborado pelo autor

Quadro 21 – CSU 004: Manter produto.

Identificador	CSU 004
Nome	Manter produto
Atores	Usuário
Requisitos	RF 004
Responsável	Gabriel Mendes Cruz
Descrição/Resumo	Este caso de uso descreve o processo de criar, editar e excluir um produto.
Pré-condições	<ul style="list-style-type: none"> • O usuário deve estar logado no aplicativo.
Pós-condições	
Cenário Principal	1. O usuário acessa a tela inicial; 2. O usuário escolhe a opção “Produtos”; 3. É exibida a tela de produtos; 4. O usuário aciona o ícone flutuante no canto inferior direito; 5. O usuário é redirecionado para a tela de cadastro de um novo produto; 6. O usuário preenche os dados; 7. O usuário aciona a opção “Confirmar”; 8. O aplicativo valida os dados informados; 9. As informações são armazenadas no banco de dados; 10. O caso de uso é finalizado.
Cenários alternativos	(A01) Passo 5: O usuário cancela a operação A01.1 O usuário seleciona a opção “Voltar” ou fecha o aplicativo; A01.2 O aplicativo retorna ao passo 3 do cenário principal. (A02) Passo 3: O usuário edita um produto A02.1 O usuário seleciona um produto cadastrado; A02.2 O aplicativo exibe as opções de edição e exclusão; A02.3 O usuário seleciona a opção de editar; A02.4 O usuário é redirecionado para a tela de edição de produto; A02.5 O usuário altera os dados;

	<p>A02.6 O usuário aciona a opção “Confirmar”;</p> <p>A02.7 O aplicativo valida os dados informados;</p> <p>A02.8 O aplicativo retorna ao passo 10 do cenário principal.</p> <p>(A03) Passo 3: O usuário exclui um produto</p> <p>A03.1 O usuário seleciona a opção de exclusão;</p> <p>A03.2 O aplicativo exibe a mensagem de confirmação de exclusão;</p> <p>A03.3 O produto é excluído;</p> <p>A03.4 O usuário é redirecionado para a tela de produtos;</p> <p>A03.5 O aplicativo retorna ao passo 10 do cenário principal.</p>
Cenário de exceção	<p>(E01) Passo 7: O usuário não informa um valor para os campos “Nome”, “Quantidade” e “Valor”</p> <p>E01.1 O aplicativo apresenta a mensagem – “Preencha o/a (campo) do produto”</p> <p>E01.2 O aplicativo retorna ao passo 6 do cenário principal.</p>
Qualidades	RQ 005

Fonte: Elaborado pelo autor

Quadro 22 – CSU 005: Manter contato.

Identificador	CSU 005
Nome	Manter contato
Atores	Usuário
Requisitos	RF 005
Responsável	Gabriel Mendes Cruz
Descrição/Resumo	Este caso de uso descreve o processo de criar, editar, listar e excluir um contato.
Pré-condições	<ul style="list-style-type: none"> • O usuário deve estar logado no aplicativo.
Pós-condições	
Cenário Principal	<ol style="list-style-type: none"> 1. O usuário acessa a tela inicial; 2. O usuário escolhe a opção “Contatos”; 3. É exibida a tela de contatos; 4. O usuário aciona o ícone flutuante no canto inferior direito; 5. O usuário é redirecionado para a tela de cadastro de contatos; 6. O usuário define a atuação do contato, entre cliente ou fornecedor; 7. O usuário preenche os dados; 8. O usuário aciona a opção “Confirmar”;

	<p>9. O aplicativo valida os dados informados;</p> <p>10. As informações são armazenadas no banco de dados;</p> <p>11. O caso de uso é finalizado.</p>
Cenários alternativos	<p>(A01) Passo 5: O usuário cancela a operação</p> <p>A01.1 O usuário seleciona a opção “Voltar” ou fecha o aplicativo;</p> <p>A01.2 O aplicativo retorna ao passo 3 do cenário principal.</p> <p>(A02) Passo 3: O usuário edita um contato</p> <p>A02.1 O usuário seleciona um contato cadastrada;</p> <p>A02.2 O aplicativo exibe as opções de edição e exclusão;</p> <p>A02.3 O usuário seleciona a opção de editar;</p> <p>A02.4 O usuário é redirecionado para a tela de edição de contatos;</p> <p>A02.5 O usuário altera os dados;</p> <p>A02.6 O usuário aciona a opção “Confirmar”;</p> <p>A02.7 O aplicativo valida os dados informados;</p> <p>A02.8 O aplicativo retorna ao passo 10 do cenário principal.</p> <p>(A03) Passo 3: O usuário exclui um contato</p> <p>A03.1 O usuário seleciona a opção de exclusão;</p> <p>A03.2 O aplicativo exibe a mensagem de confirmação de exclusão;</p> <p>A03.3 O contato é excluído;</p> <p>A03.4 O usuário é redirecionado para a tela de contatos;</p> <p>A03.5 O aplicativo retorna ao passo 10 do cenário principal.</p>
Cenário de exceção	<p>(E01) Passo 7: O usuário não informa um valor para os campos “Nome”, “Telefone” e “Tipo”</p> <p>E02.1 O aplicativo apresenta a mensagem – “Preencha o/a (<i>campo</i>) do produto”</p> <p>E02.2 O aplicativo retorna ao passo 6 do cenário principal.</p>
Qualidades	RQ 005

Fonte: Elaborado pelo autor

Quadro 23 – CSU 006: Visualizar estoque.

Identificador	CSU 006
Nome	Visualizar estoque
Atores	Usuário
Requisitos	RF 006
Responsável	Gabriel Mendes Cruz

Descrição/Resumo	Este caso de uso descreve o processo de visualização do estoque no aplicativo.
Pré-condições	<ul style="list-style-type: none"> O usuário deve estar logado no aplicativo.
Pós-condições	
Cenário Principal	<ol style="list-style-type: none"> O usuário acessa a tela inicial; O usuário escolhe a opção “Produtos”; É exibida a tela de produtos; O aplicativo exibe produtos cadastrados; O caso de uso é finalizado.
Cenários alternativos	<p>(A01) Passo 4: O usuário realiza um filtro</p> <p>A01.1 O usuário preenche os dados do filtro;</p> <p>A01.2 O usuário seleciona a opção “Filtrar”;</p> <p>A01.3 O aplicativo exibe os dados dos produtos que satisfazem os filtros definidos;</p> <p>A01.4 O aplicativo é redirecionado para o passo 4 do cenário principal.</p>
Cenário de exceção	<p>(E01) Passo 3: O aplicativo não possui produto cadastrado</p> <p>E01.1 O aplicativo apresenta a mensagem – “Não há produtos cadastrados”</p> <p>E01.2 O aplicativo retorna ao passo 4 do cenário principal.</p>
Qualidades	

Fonte: Elaborado pelo autor

Quadro 24 – CSU 007: Definir pedido.

Identificador	CSU 007
Nome	Definir pedido
Atores	Usuário
Requisitos	RF 007
Responsável	Gabriel Mendes Cruz
Descrição/Resumo	Este caso de uso descreve o processo de definição de um pedido.
Pré-condições	<ul style="list-style-type: none"> O usuário deve estar logado no aplicativo.
Pós-condições	
Cenário Principal	<ol style="list-style-type: none"> O usuário acessa a tela inicial; O usuário escolhe a opção “Pedidos”; É exibida a tela de pedidos; O usuário aciona o ícone flutuante no canto inferior direito; O usuário é redirecionado para a tela de detalhes de um novo produto para definir; O usuário define um cliente;

	<p>7. O usuário adiciona um produto;</p> <p>8. O usuário aciona a opção “Confirmar”;</p> <p>9. O aplicativo valida os dados informados;</p> <p>10. As informações são armazenadas no banco de dados;</p> <p>11. O caso de uso é finalizado.</p>
Cenários alternativos	<p>(A01) Passo 5: O usuário cancela a operação</p> <p>A01.1 O usuário seleciona a opção “Voltar” ou fecha o aplicativo;</p> <p>A01.2 O aplicativo retorna ao passo 3 do cenário principal.</p>
Cenário de exceção	<p>(E01) Passo 6: O aplicativo não possui cliente cadastrado</p> <p>E01.1 O aplicativo apresenta a mensagem – “Não há clientes cadastrados”</p> <p>E01.2 O aplicativo retorna ao passo 5 do cenário principal.</p> <p>(E02) Passo 7: O aplicativo não possui produto cadastrado</p> <p>E01.1 O aplicativo apresenta a mensagem – “Não há produtos cadastrados”.</p> <p>E01.2 O aplicativo retorna ao passo 5 do cenário principal.</p>
Qualidades	RQ 005

Fonte: Elaborado pelo autor

Quadro 25 – CSU 008: Cancelar pedido.

Identificador	CSU 008
Nome	Cancelar pedido
Atores	Usuário
Requisitos	RF 008
Responsável	Gabriel Mendes Cruz
Descrição/Resumo	Este caso de uso descreve o processo de cancelamento de um pedido.
Pré-condições	<ul style="list-style-type: none"> • O usuário deve estar logado no aplicativo.
Pós-condições	
Cenário Principal	<p>1. O usuário acessa a tela inicial;</p> <p>2. O usuário escolhe a opção “Pedidos”;</p> <p>3. É exibida a tela de listagem de pedidos;</p> <p>4. O usuário seleciona e pressiona um pedido;</p> <p>5. O usuário escolhe a opção “Cancelar”;</p> <p>6. O caso de uso é finalizado.</p>
Cenários alternativos	(A01) Passo 5: O usuário desiste do cancelamento

	A01.1 O usuário seleciona a opção “Fechar” no passo 5, pressiona fora ou fecha o aplicativo; A01.2 O aplicativo retorna ao passo 3 do cenário principal.
Cenário de exceção	(E01) Passo 4: O pedido já foi confirmado E01.1 O aplicativo apresenta a mensagem – “Pedido já confirmado, não é possível cancelar” E01.2 O aplicativo retorna ao passo 6 do cenário principal.
Qualidades	

Fonte: Elaborado pelo autor

Quadro 26 – CSU 009: Visualizar pedido.

Identificador	CSU 009
Nome	Visualizar pedido
Atores	Usuário
Requisitos	RF 009
Responsável	Gabriel Mendes Cruz
Descrição/Resumo	Este caso de uso descreve o processo de visualização de um pedido no aplicativo.
Pré-condições	<ul style="list-style-type: none"> • O usuário deve estar logado no aplicativo.
Pós-condições	
Cenário Principal	<ol style="list-style-type: none"> 1. O usuário acessa a tela inicial; 2. O usuário escolhe a opção “Pedidos”; 3. É exibida a tela de listagem de pedidos. 4. O usuário seleciona e pressiona um pedido; 5. O aplicativo exibe a tela de detalhes de um pedido; 6. O caso de uso é finalizado.
Cenários alternativos	(A01) Passo 4: O usuário realiza um filtro A01.1 O usuário preenche os dados do filtro; A01.2 O usuário seleciona a opção “Filtrar”; A01.3 O aplicativo exibe os dados dos pedidos que satisfazem os filtros definidos; A01.4 O aplicativo é redirecionado para o passo 3 do cenário principal.
Cenário de exceção	(E01) Passo 4: O aplicativo não possui produto cadastrado E01.1 O aplicativo apresenta a mensagem – “Não foram definidos pedidos”; E01.2 O aplicativo retorna ao passo 3 do cenário principal.
Qualidades	

Fonte: Elaborado pelo autor

Quadro 27 – CSU 010: Manter ficha financeira.

Identificador	CSU 010
Nome	Manter ficha financeira
Atores	Usuário
Requisitos	RF 010
Responsável	Gabriel Mendes Cruz
Descrição/Resumo	Este caso de uso descreve o processo de inserção, edição, visualização e remoção de dados na ficha financeira.
Pré-condições	<ul style="list-style-type: none"> • O usuário deve estar logado no aplicativo.
Pós-condições	
Cenário Principal	<ol style="list-style-type: none"> 1. O usuário acessa a tela inicial; 2. O usuário escolhe a opção “Finanças”; 3. O aplicativo exibe a tela de ficha financeira com os valores de entradas (sincronizados com os pedidos) e saídas (definidas pelo usuário); 4. O usuário aciona o ícone flutuante no canto inferior direito; 5. O usuário é redirecionado para a tela de pagamento 6. O usuário define o fornecedor, produto e valor; 7. O usuário preenche os dados; 8. O usuário aciona a opção “Confirmar”; 9. O aplicativo valida os dados informados; 10. As informações são armazenadas no banco de dados; 11. O caso de uso é finalizado.
Cenários alternativos	<p>(A01) Passo 5: O usuário cancela a operação A01.1 O usuário seleciona a opção “Voltar” ou fecha o aplicativo; A01.2 O aplicativo retorna ao passo 3 do cenário principal.</p> <p>(A02) Passo 3: O usuário edita uma saída A02.1 O usuário seleciona uma saída de caixa; A02.2 O aplicativo exibe as opções de edição e exclusão; A02.3 O usuário seleciona a opção de editar; A02.4 O usuário é redirecionado para a tela de edição de pagamento; A02.5 O usuário altera os dados; A02.6 O usuário aciona a opção “Confirmar”; A02.7 O aplicativo valida os dados informados; A02.8 O aplicativo retorna ao passo 10 do cenário principal.</p> <p>(A03) Passo 3: O usuário exclui uma saída</p>

	<p>A03.1 O usuário seleciona a opção de exclusão;</p> <p>A03.2 O aplicativo exibe a mensagem de confirmação de exclusão;</p> <p>A03.3 A saída de caixa é excluída;</p> <p>A03.4 O usuário é redirecionado para a tela de ficha financeira;</p> <p>A03.5 O aplicativo retorna ao passo 10 do cenário principal.</p>
Cenário de exceção	<p>(E01) Passo 7: O usuário não define os campos “Fornecedor”, “Produto” e “Valor”</p> <p>E01.1 O aplicativo apresenta a mensagem – “Preencha o/a (campo) do produto”</p> <p>E01.2 O aplicativo retorna ao passo 6 do cenário principal.</p> <p>(E02) Passo 6: Não existe fornecedor cadastrado no aplicativo para vincular a uma saída</p> <p>E02.1 O aplicativo apresenta a mensagem – “Não há fornecedores cadastrados”;</p> <p>E02.2 O aplicativo retorna ao passo 5 do cenário principal.</p>
Qualidades	RQ 005

Fonte: Elaborado pelo autor

Quadro 28 – CSU 011: Visualizar relatórios.

Identificador	CSU 011
Nome	Visualizar relatórios
Atores	Usuário
Requisitos	RF 011
Responsável	Gabriel Mendes Cruz
Descrição/Resumo	Este caso de uso descreve o processo de visualização de relatórios.
Pré-condições	<ul style="list-style-type: none"> • O usuário deve estar logado no aplicativo.
Pós-condições	
Cenário Principal	<ol style="list-style-type: none"> 1. O usuário acessa a tela inicial; 2. São exibidos relatórios em forma de gráficos de barras de vendas e finanças. 3. O usuário seleciona o gráfico e são exibidos os valores de cada barra; 4. O caso de uso é finalizado.
Cenários alternativos	<p>(A01) Passo 4: O usuário realiza um filtro</p> <p>A01.1 O usuário preenche os dados do filtro;</p> <p>A01.2 O usuário seleciona a opção “Filtrar”;</p> <p>A01.3 O aplicativo exibe os dados dos pedidos que satisfazem os filtros definidos;</p>

	A01.4 O aplicativo é redirecionado para o passo 3 do cenário principal.
Cenário de exceção	(E01) Passo 4: O aplicativo não possui pedido cadastrado E01.1 O aplicativo apresenta a mensagem – “Não há pedidos definidos”; E01.2 O aplicativo retorna ao passo 3 do cenário principal.
Qualidades	

Fonte: Elaborado pelo autor

3.6 Dicionário de dados do sistema

Esta seção apresenta os quadros que compõe o dicionário de dados das informações que serão salvas no banco de dados.

Quadro 29 – DD 001: Dados do usuário.

Identificador	Descrição			
DD 001	Dados do Usuário			
Identificador	Tamanho	Tipo	Formato	Descrição
ID	20	<i>String</i>		Identificador do usuário
Nome	50	<i>String</i>		Nome do usuário
E-mail	100	<i>String</i>	\w+@\w+\.\w+	E-mail do usuário
Senha	6	<i>String</i>		Senha do usuário

Fonte: Elaborado pelo autor

Quadro 30 - DD 002: Dados do produto.

Identificador	Descrição			
DD 002	Dados do produto			
Identificador	Tamanho	Tipo	Formato	Descrição
ID	20	<i>String</i>		Identificador do produto
Título	50	<i>String</i>		Título do produto

Tamanho	2	Tamanho		Tamanho vinculado ao produto
Valor	7	<i>Double</i>		Valor do produto
Descrição	100	<i>String</i>		Descrição do produto
Excluído	2	Lógico		Identificador de exclusão do produto

Fonte: Elaborado pelo autor

Quadro 31 – DD 003: Dados do tamanho do produto.

Identificador	Descrição			
DD 002	Dados do tamanho do produto			
Identificador	Tamanho	Tipo	Formato	Descrição
ID	20	<i>String</i>		Identificador do tamanho
Valor do tamanho	4	<i>Double</i>		Valor do tamanho do produto
Estoque	4	Inteiro		Estoque dos produtos vinculado ao valor do tamanho

Fonte: Elaborado pelo autor

Quadro 32 – DD 004: Dados do contato.

Identificador	Descrição			
DD 003	Dados do contato			
Identificador	Tamanho	Tipo	Formato	Descrição
ID	20	<i>String</i>		Identificador de contato
Nome	100	<i>String</i>		Nome do contato
Telefone	11	<i>String</i>		Telefone do contato
Endereço	100	<i>String</i>		Endereço do contato
Tipo	2	Lógico		Tipo da pessoa (física ou jurídica)

Atuação	2	Lógico		Atuação do contato (cliente ou fornecedor)
Excluído	2	Lógico		Identificador de exclusão do contato

Fonte: Elaborado pelo autor

Quadro 33 – DD 005: Dados do pedido.

Identificador	Descrição			
DD 004	Dados do pedido			
Identificador	Tamanho	Tipo	Formato	Descrição
ID	6	<i>String</i>		Identificador do pedido
Data	8	<i>Date</i>		Data do pedido
Cliente	20	<i>String</i>		Cliente que realizou o pedido
Valor	7	<i>Double</i>		Valor do pedido
Quantidade	4	Inteiro		Quantidade de produtos no pedido
Produto	20	Produto		Produtos vinculados ao pedido

Fonte: Elaborado pelo autor

4 DESENVOLVIMENTO DO APLICATIVO

Este capítulo apresenta as tecnologias usadas para a criação do aplicativo, dentre elas estão o *framework* Flutter, a linguagem de programação Dart, a comunicação com o banco de Dados Firebase e a IDE Visual Studio Code.

4.1 Flutter

Flutter é o *framework* de interface gráfica organizado pela Google e de código aberto, que proporciona o desenvolvimento móvel multiplataforma (Android e iOS), criados com uma única base de código. Utilizando a linguagem Dart, os códigos são compilados diretamente ao ARM nativo e podem acessar API's de plataformas e serviços (FLUTTER, 2021).

O framework foi feito para acelerar a velocidade de desenvolvimento e permitir a redução de custo, e reuso de código sem a necessidade de recompilar o aplicativo, com o uso do *Hot Reload*. Flutter possui pacotes de *widgets* customizáveis e reativos.

4.1.1 Arquitetura

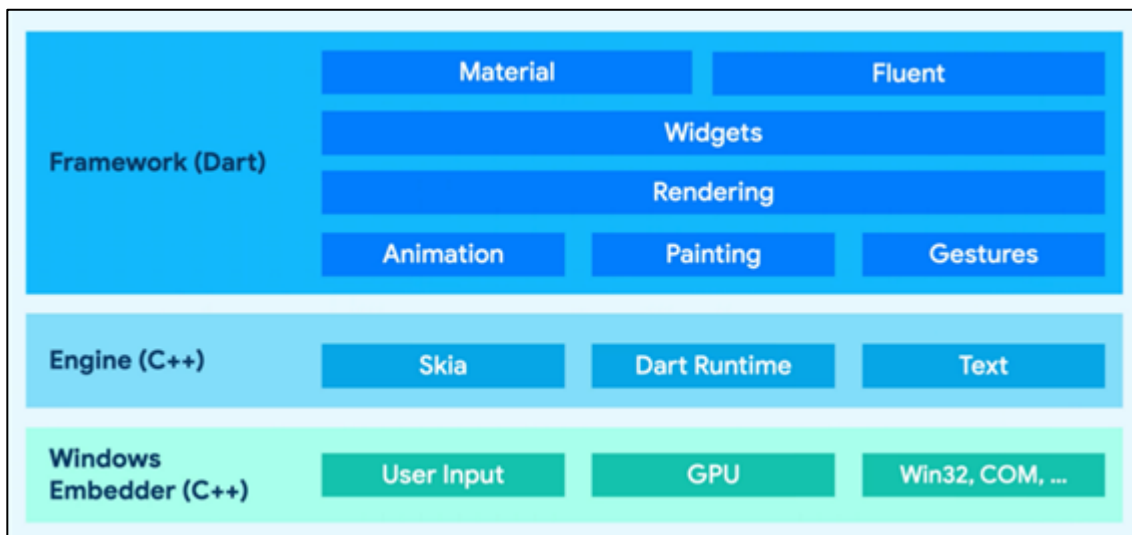
O Flutter possui um kit de ferramentas de interface do usuário que permite a reutilização de códigos em sistemas operacionais iOS e Android (FLUTTER, 2022). Sua arquitetura é baseada em camadas, Figura 2, que não possuem acesso privilegiado à camada inferior e qualquer parte do framework foi projetada para ser opcional e substituível, dando aos programadores resultados rápidos, uma vez que é usada uma quantidade pequena de código, se comparado à outras linguagens e *frameworks* de desenvolvimento *mobile*.

As camadas de sua arquitetura são organizadas em três seções, conforme mostra a Figura 2:

- Camada *Framework*: construída em Dart, contém as bibliotecas e funcionalidades de estilização, *widgets*, renderização e construção da interface do aplicativo;

- Camada *Engine*: escrita em C/C++, responsável por implementar as funções de baixo nível da API do Flutter, como os gráficos, que utilizam o motor *Skia*, *layout* de texto, I/O, rede e arquivos;
- Camada *Embedder*: responsável por gerenciar o acesso dos recursos nativos dos sistemas operacionais ao aplicativo.

Figura 2 – Arquitetura do Flutter



Fonte: Elaborado pelo autor

4.1.2 Widgets

No Flutter, os *widgets* são o principal bloco de construção da interface, inspirados no React, com o objetivo que a mesma seja construída a partir de *widgets*, que descrevem como a visualização deve ser, de acordo com a configuração e o estado atual (FLUTTER, 2022).

Além disso, os *widgets* podem ser dinâmicos ou estáticos, definidos como:

- *StatelessWidget*: *Widgets* estáticos, onde todas as suas configurações são feitas durante a inicialização e não pode ser alterada durante a execução do aplicativo.
- *StatefulWidget*: *Widgets* dinâmicos, sendo que além da configuração inicial, também possuem um *State*, que define seu estado atual, quando alterado, ele dispara novamente o método *build()* do *widget*, o que faz com que seja redesenhada a tela.

4.2 Dart

Dart é uma linguagem de desenvolvimento otimizada para aplicativos multiplataforma. O objetivo é oferecer uma linguagem de programação produtiva para o desenvolvimento multiplataforma, com tempo de execução flexível que possa combinar com frameworks de desenvolvimento de aplicativos (DART, 2021).

A linguagem foi criada pelo Google com a intenção de substituir o JavaScript e se tornar a mais utilizada pelos navegadores. Porém, o Dart ainda está em processo de aprimoramentos e adaptações, e nessa evolução se tornou multiparadigma (HOSTGATOR, 2021).

É uma linguagem de programação fortemente tipada e possui uma sintaxe com estilo baseado na linguagem C. Isso faz com que sua sintaxe seja similar a linguagens populares, como Java e C#.

Segundo HostGator (2021), Dart funciona em diferentes plataformas e disponibiliza duas soluções distintas com propósitos diferentes:

- **Dart Native:** É utilizado para o desenvolvimento *mobile*, *desktop* e de servidor, entre outros. Para realizar a produção de código, ele inclui a máquina virtual Dart com compilação *just-in-time* (JIT) e compilador *ahead-of-time* (AOT). Ele reúne o código para Dart de forma nativa, seja em ARM ou X64;
- **Dart Web:** Tem como foco o desenvolvimento *web* e traz o `dartdevc`, que compila o código para o desenvolvimento, e o `dart2js`, que é indicado para fazer o *build* para produção. Neste caso, o resultado é um Javascript, que será facilmente interpretado por qualquer navegador – visto que essa linguagem é universal.








































4.3 Firebase

O *Firebase* é uma plataforma de desenvolvimento *Backend As A Service* (BaaS) pertencente ao Google, lançado em 2011, e provê diversas facilidades para o desenvolvimento de uma aplicação, seja ela *web* ou *mobile*.

Criado sobre a infraestrutura do Google, é possível realizar o escalonamento automático até mesmo para aplicativos complexos. Possui diferentes funcionalidades disponíveis, como por exemplo, análises em tempo real, bancos de dados, mensagens e relatórios de erros (ANDRADE, 2021).

No aplicativo desenvolvido, o *Firebase* é utilizado para realizar cadastramento e autenticação de usuários, recuperação de senha e armazenamento de dados, como produtos, clientes, fornecedores, utilizando o plano gratuito. O *Firebase* fornece diversas ferramentas para desenvolvimento de aplicativos, conforme mostra a Figura 3.

Figura 3 – Serviços do *Firebase*

Criar o aplicativo	Melhorar a qualidade do aplicativo	Aumente seus negócios
 Cloud Firestore iOS  	 Crashlytics iOS 	 Análise iOS  C++ 
 Kit de ML iOS 	 Monitoramento de desempenho iOS 	 Previsões iOS  C++ 
 Cloud Functions iOS  C++ 	 Test Lab iOS 	 Teste A/B do Firebase iOS  C++ 
 Autenticação iOS  C++ 		 Cloud Messaging iOS  C++ 
 Hosting C++		 Configuração remota iOS  C++ 
 Cloud Storage iOS  C++ 		 Dynamic Links iOS  C++ 
 Realtime Database iOS  C++ 		 App Indexing iOS 
		 Invites iOS  C++ 

Fonte: Elaborado pelo autor

4.3.1 Autenticação do *Firebase*

O aplicativo utiliza o *Firebase Authentication* para realizar o cadastro, autenticação e recuperação de senha dos usuários.

O *Firebase* dá a possibilidade de autenticação de usuários por meio dos serviços do próprio *Firebase*, através do login com endereço de e-mail e senha, número de telefone ou link. O SDK do *Firebase Authentication* fornece métodos que permitem aos usuários fazer login com as Contas do *Google*, *Facebook*,

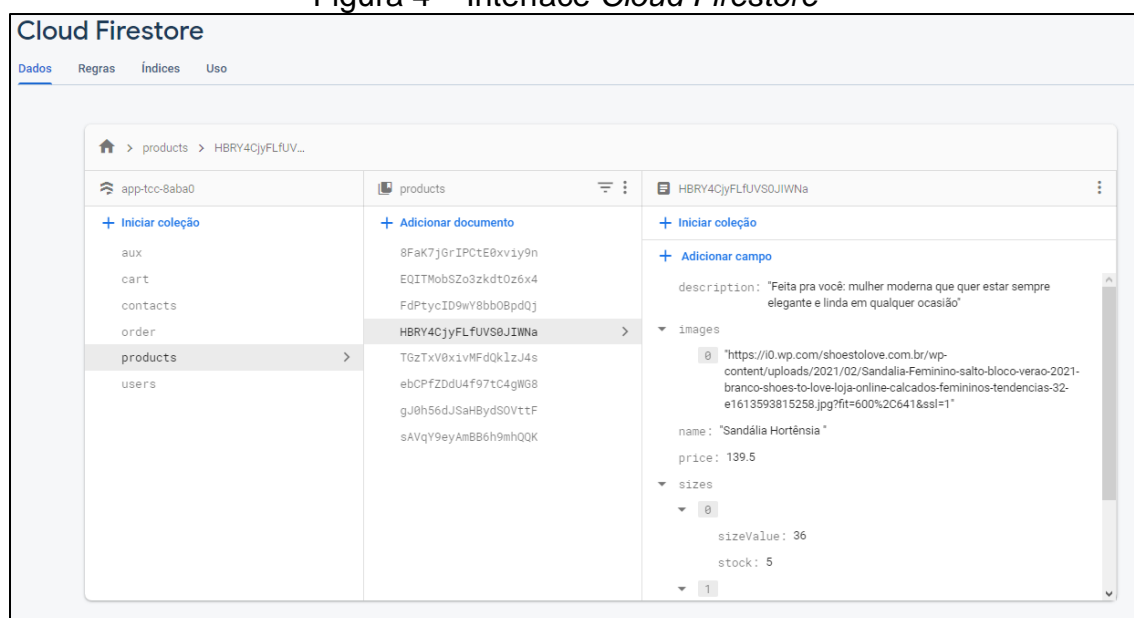
Twitter e *GitHub*. De todas as opções, o aplicativo utiliza apenas a autenticação, cadastro por e-mail e senha.

4.3.1 *Firestore Cloud*

O *Cloud Firestore* é um banco de dados não relacional hospedado em nuvem que disponibiliza estruturas de dados hierárquicas flexíveis. O controle dos dados se baseia em documentos, organizados em coleções. Os documentos podem conter objetos aninhados complexos, além de subcoleções.

Um exemplo de sua interface é apresentado na Figura 5.

Figura 4 – Interface *Cloud Firestore*

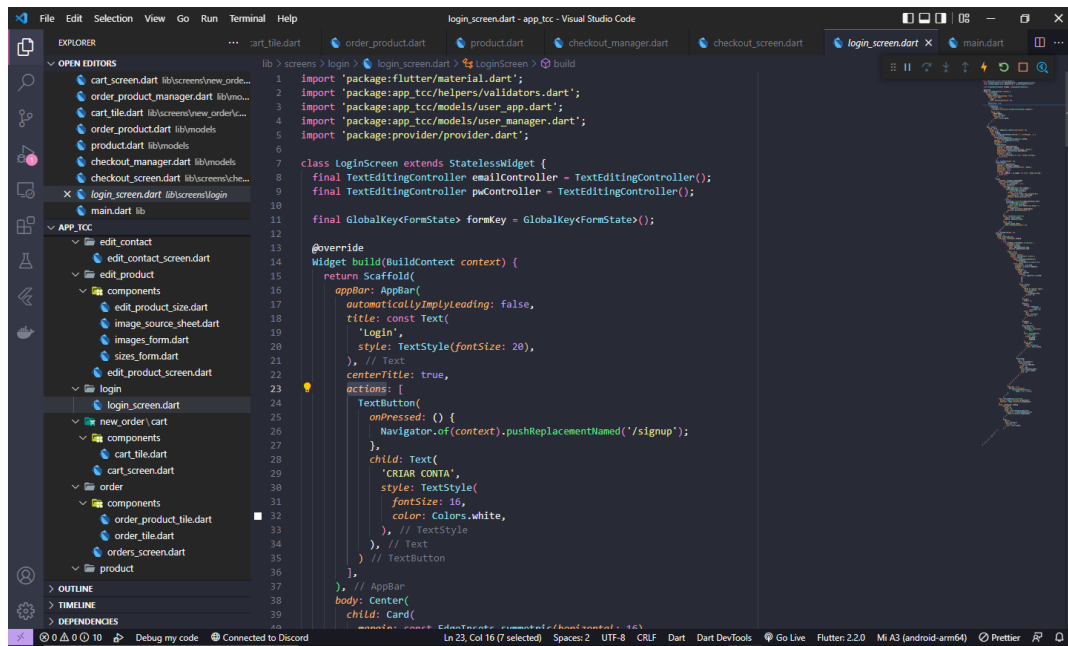


Fonte: Elaborado pelo autor

4.4 *Visual Studio Code*

A IDE, ou ambiente de desenvolvimento integrado, escolhido para o desenvolvimento do aplicativo foi o Visual Studio Code, Figura 5, pois além do benefício de ser gratuito e leve, possui *IntelliSense*, que é o nome dado a um conjunto de recursos que tornam a codificação leve e eficiente, e ser a IDE recomendada no próprio site oficial do Flutter (FLUTTER, 2022).

Figura 5 – Interface do Visual Studio Code



Fonte: Elaborado pelo autor

5 IMPLEMENTAÇÕES E RESULTADOS

Neste capítulo são apresentadas as telas do aplicativo, descrevendo os seus componentes e as ações disponíveis aos usuários.

5.1 Telas do Aplicativo

Esta seção apresenta as telas do aplicativo.

5.1.1 Tela de Login

A Figura 6 apresenta a tela inicial do aplicativo, onde o usuário tem a opção de realizar o login, recuperar senha ou cadastrar-se.

Figura 6 – Tela de *Login*

A imagem mostra a tela de login do aplicativo DONANJA. O fundo é azul escuro. No topo, há um ícone de asas brancas. Abaixo do ícone, o nome "DONANJA" está em letras brancas maiúsculas, seguido por "Gestão de Vendas" em letras brancas menores. Centralizado na tela, há um formulário branco com dois campos de entrada: "E-mail" e "Senha". Abaixo dos campos, há um botão azul com o texto "Login" em branco. Abaixo do botão, há dois links em texto azul: "Esqueci minha senha" e "Cadastre-se".

Fonte: Elaborado pelo autor

Essa tela é composta pelos componentes:

- E-mail: Campo que o usuário informa um e-mail já cadastrado;
- Senha: Campo que o usuário informa a senha correspondente ao e-mail cadastrado;
- Login: Botão que ao ser pressionado irá realizar a validação dos campos de e-mail e senha. Caso sejam validados, o usuário será redirecionado para a Tela Principal; caso contrário, será exibida uma mensagem contendo o erro;
- Esqueci minha senha: Botão que ao ser pressionado realiza a validação do campo de e-mail e envia o mesmo o procedimento de recuperação de senha;
- Cadastre-se: Botão que irá redirecionar o usuário para a Tela de Cadastro.

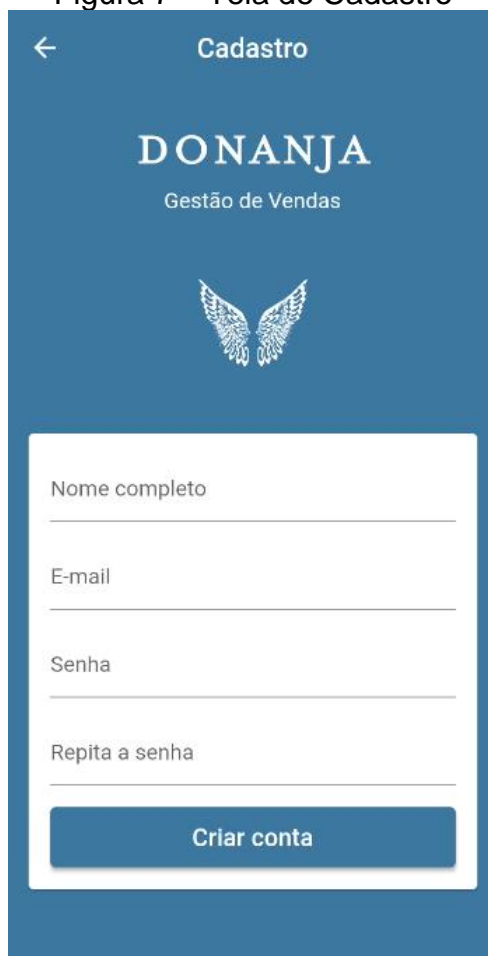
5.1.2 Tela de Cadastro

A Figura 7 mostra a tela onde o usuário realiza o seu cadastro no aplicativo, informando os dados:

- Nome: Campo onde o usuário deve informar o seu nome;
- E-mail: Campo onde o usuário deve informar o seu e-mail;
- Senha: Campo onde o usuário deve informar uma senha que contenha, no mínimo, 6 dígitos;
- Confirmação da Senha: Campo onde o usuário deve informar a mesma senha que foi informada no campo “Senha”.

Ao pressionar o botão “Criar conta”, o aplicativo irá validar se o e-mail informado já está cadastrando, se o campo “Senha” corresponde ao campo “Confirme a Senha” e se possuem no mínimo 6 caracteres, exibindo uma mensagem de erro para cada caso.

Figura 7 – Tela de Cadastro

A imagem mostra a tela de cadastro de um aplicativo. No topo, há uma barra azul com um ícone de seta para trás à esquerda e o título "Cadastro" no centro. Abaixo, o nome "DONANJA" é exibido em letras brancas, seguido por "Gestão de Vendas" em uma fonte menor. Centralizado abaixo do texto é um ícone de asas brancas. A área principal de entrada de dados é um formulário branco com campos rotulados "Nome completo", "E-mail", "Senha" e "Repita a senha". Cada campo possui uma linha de entrada. No final do formulário, há um botão azul com o texto "Criar conta" em branco.

Fonte: Elaborado pelo autor

5.1.3 Tela de Redefinição de senha

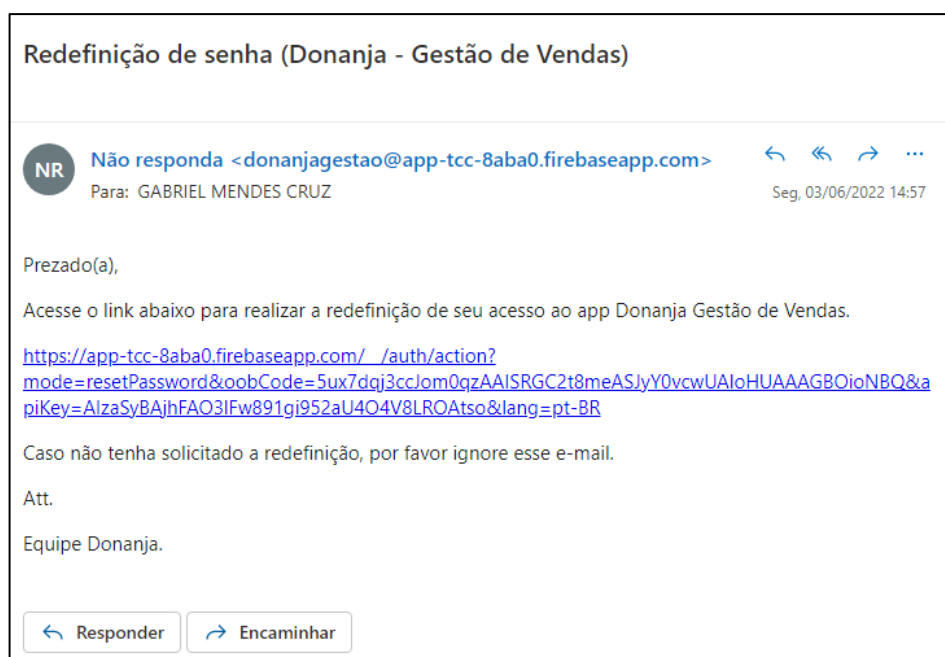
A tela de Login também pode ser usada como tela de Recuperação de senha, e permite o usuário iniciar o processo de redefinição de senha. O usuário deve informar seu e-mail no campo de e-mail no Login e pressionar o botão “Esqueci minha senha”, o aplicativo irá validar se o e-mail é válido e atende à algum cadastrado.

Figura 8 – Tela de envio de e-mail



Fonte: Elaborado pelo autor

Figura 9 – E-mail do processo de redefinição de senha



Fonte: Elaborado pelo autor

Figura 10 – Aviso de e-mail não cadastrado

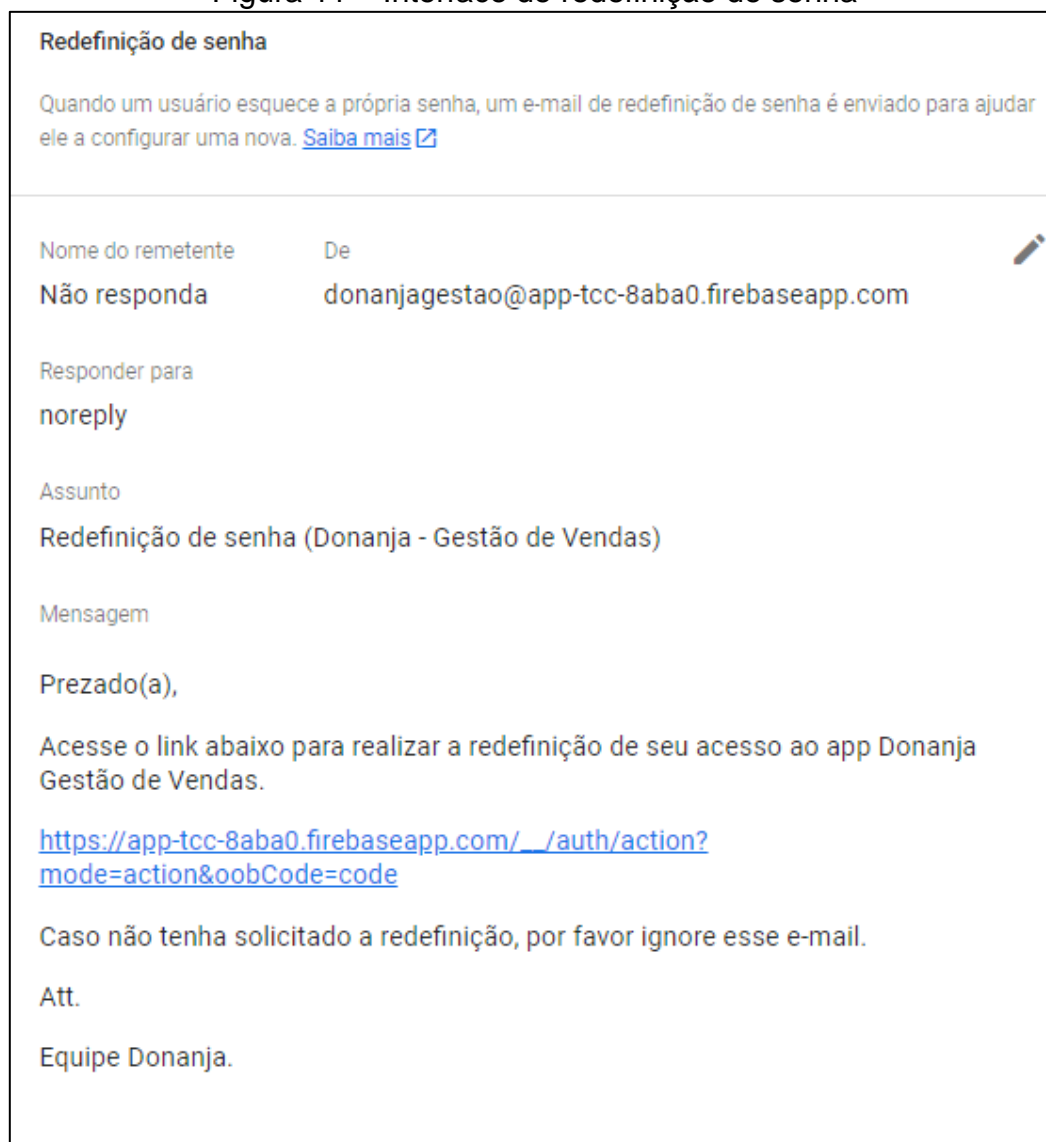


Fonte: Elaborado pelo autor

Caso a validação obtenha êxito, o e-mail com as instruções de troca de senha será enviado, exibido na Figura 9. Caso contrário, será exibida uma mensagem de erro, Figura 10.

O modelo de e-mail que é enviado ao usuário para alterar sua senha de acesso é exibido na Figura 11.

Figura 11 – Interface de redefinição de senha



Fonte: Elaborado pelo autor

5.1.4 Tela Home

A Figura 12 exibe a tela inicial do aplicativo. Nela estão presentes os componentes de relatório:

- Últimos pedidos: exibe um carrossel flutuante com a prévia dos pedidos realizados;
- Vendas mensais: exibe um gráfico com informações da quantidade de produtos vendidos nos últimos 5 meses;
- Transações mensais: exibe um gráfico com informações de entradas e saída financeiras nos últimos 5 meses.

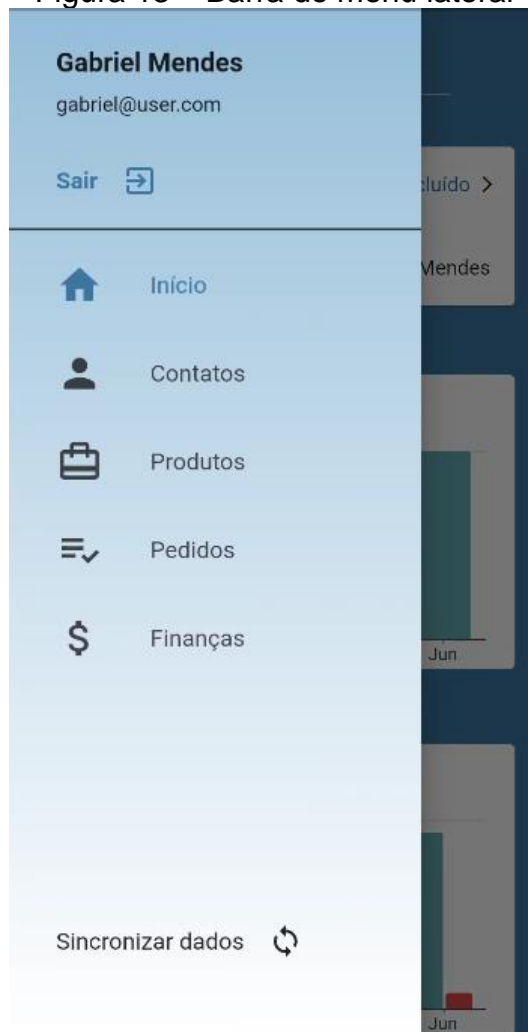
Figura 12 – Tela Home



Fonte: Elaborado pelo autor

Ao pressionar cada elemento o usuário é redirecionado para a tela específica a qual pertencem os dados, por exemplo a tela de pedidos e tela de finanças.

Figura 13 – Barra de menu lateral



Fonte: Elaborado pelo autor

Como um componente vinculado à tela Home, a Figura 13 mostra a barra lateral de navegação, que é exibida nas telas, com exceção das telas de Login, Cadastro e Redefinição de senha. O menu possibilita a navegação entre as telas do aplicativo, com as opções:

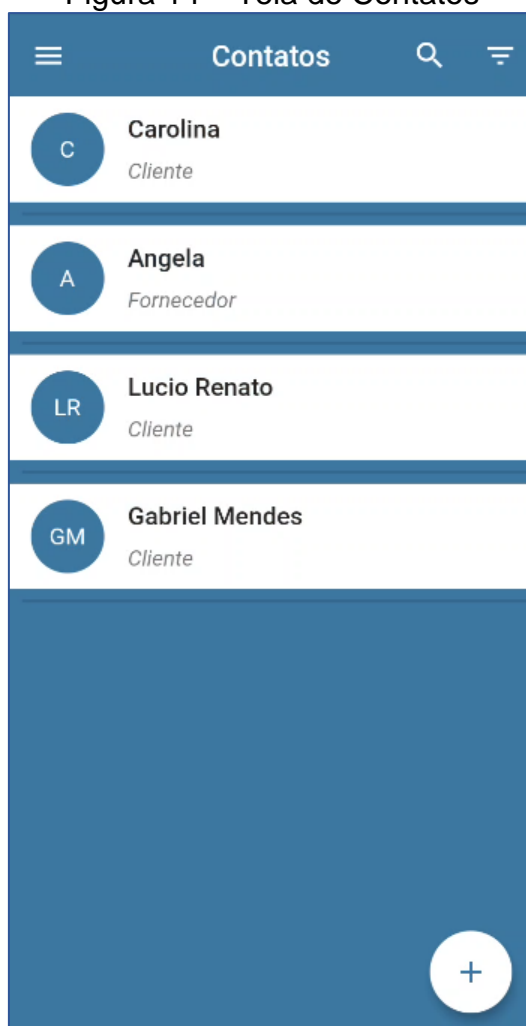
- Nome e e-mail: Campo onde o aplicativo exibe o nome e e-mail do usuário logado;
- Botão “Sair”: Ao ser pressionado desconecta o usuário do aplicativo e o redireciona para a tela de Login;
- Início: Ao pressionar essa opção o usuário é redirecionado para a tela Home;
- Contatos: Redireciona o usuário para a tela de Contatos;
- Produtos: Redireciona o usuário para a tela de Estoque dos produtos;

- Pedidos: Redireciona o usuário para a tela de Pedidos;
- Finanças: Redireciona o usuário para a tela de Ficha Financeira;
- Sincronizar dados: Ao pressionar, o aplicativo é reiniciado e estabelece uma nova conexão ao *Firebase* e sincroniza os dados criados e alterados.

5.1.5 Tela de Contatos

Nesta tela são exibidos os contatos salvos disponíveis no aplicativo, organizados em um formato de lista, contendo as informações do Nome e Relacionamento do contato, classificando como cliente ou fornecedor. Ao pressionar um contato entre os exibidos na lista o usuário é redirecionado para a tela dos detalhes do contato.

Figura 14 – Tela de Contatos



Fonte: Elaborado pelo autor

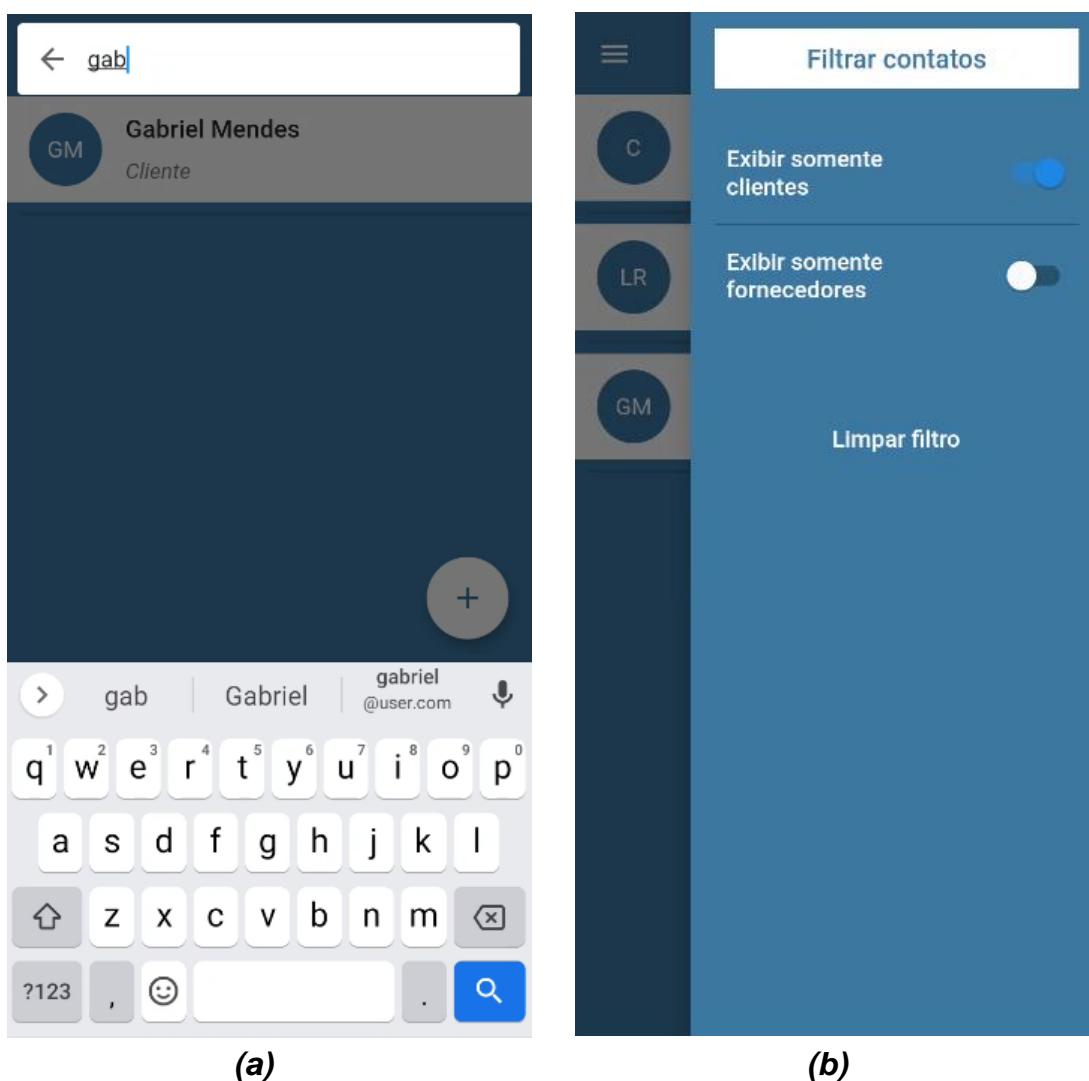
A Tela de Contatos também apresenta um botão flutuante que ao ser pressionado redireciona o usuário para a tela de cadastro de um novo contato.

A tela possui um *AppBar* que exibe seu nome, e dois botões de ação que indicam a pesquisa e filtro dos contatos.

A Figura 15 (a) e (b) ilustram as ações dos botões: ao pressionar o botão de pesquisa, é aberta uma caixa de pesquisa que filtra a lista de acordo com o nome vinculado à cada contato. Por sua vez, ao pressionar o botão de filtro, é aberto um menu na lateral direita da tela com as opções de filtragem de contatos por relacionamento:

- Exibir apenas contatos clientes;
- Exibir apenas contatos fornecedores;
- Limpar os filtros e exibir todos os contatos.

Figura 15 – Tela de pesquisa (a) e filtro dos contatos(b)



5.1.6 Tela de Cadastro de Contatos

A tela de Cadastro de Contatos, Figura 16, permite o usuário cadastrar um novo contato no aplicativo. Esta tela possui os componentes:

- Nome: Usuário deve informar o nome do contato;
- Telefone: Usuário deve informar o telefone do contato;
- Endereço: Usuário deve informar o endereço do contato;
- Tipo de Pessoa: Usuário deve escolher a opção do Tipo de Pessoa de acordo com a Receita Federal, definindo como Física ou Jurídica;
- Relacionamento: Usuário deve definir qual o tipo de relacionamento que possui com o contato, entre Cliente e Fornecedor.

Figura 16 – Tela de Cadastro de Contato

The figure consists of two side-by-side screenshots of a mobile application interface for managing contacts.

(a) Cadastro de Contato: This is the registration screen. It has a blue header with a back arrow and the title "Novo contato". Below the header is a large blue circle placeholder for a profile picture, with the label "Nome" underneath. The form contains several fields: "Telefone:" with a phone icon and a placeholder "() - "; "Endereço:" with a house icon and the prompt "Digite o endereço"; "Tipo de Pessoa:" with a person icon and a dropdown menu labeled "Defina o tipo de Pessoa"; and "Relacionamento:" with two radio buttons labeled "CLIENTE" and "FORNECEDOR". At the bottom is a large blue button labeled "Salvar".

(b) Contato criado: This is the view of a created contact. It has a blue header with a back arrow, the contact's name "Angela", and an edit icon. Below the header is a blue circle with the letter "A" inside, with the name "Angela" underneath. The fields are populated: "Telefone:" shows "62981297970"; "Endereço:" shows "Setor Goiânia 2"; "Tipo de Pessoa:" shows "Pessoa Jurídica"; and "Relacionamento:" shows "FORNECEDOR" in blue text.

(a) Cadastro de Contato

(b) Contato criado

Fonte: Elaborado pelo autor

Ao pressionar o botão “Salvar”, o aplicativo irá validar os campos informados, exibindo uma mensagem de erro para cada campo, se necessário. Após validar, o contato é cadastrado e o usuário é redirecionado para a tela de contatos já cadastrados.

5.1.7 Tela de Edição de Contatos

Esta tela permite o usuário editar e excluir um contato já cadastrado no aplicativo.

Figura 17 – Tela de Edição de Contato

← Editar contato

Angela

Telefone:
62981297970

Endereço:
Setor Goiânia 2

Tipo de Pessoa:
Defina o tipo de Pessoa ▼

Relacionamento:
☐ CLIENTE ☐ FORNECEDOR

Salvar

Fonte: Elaborado pelo autor

Como mostra a Figura 17, a tela possui os mesmos componentes da Tela de Cadastro de Contatos, que são: nome, telefone, endereço, tipo de Pessoa e relacionamento. No entanto, possui o botão de Exclusão, que ao ser pressionado, marca o contato como deletado.

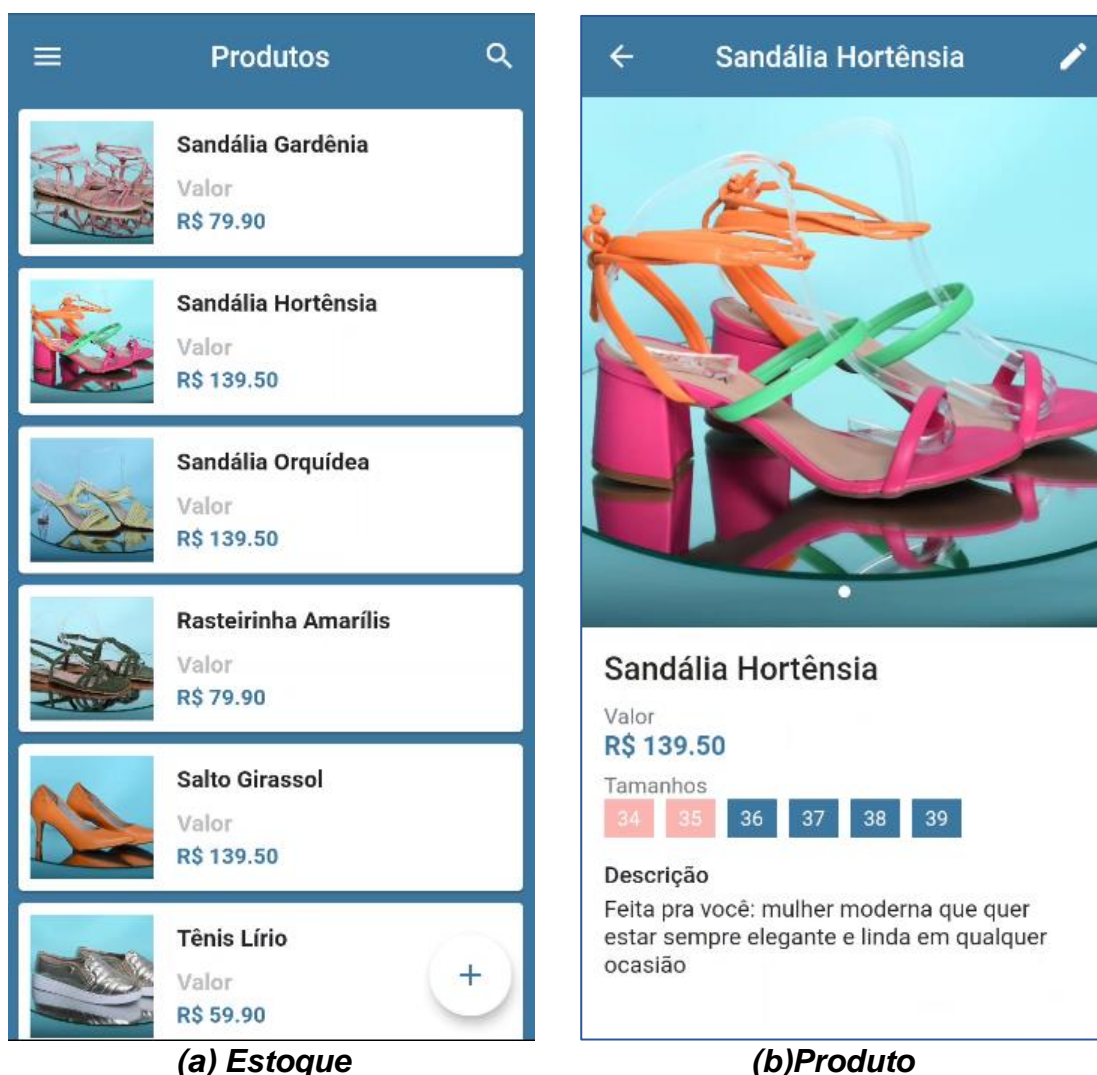
Uma vez marcado, o contato não é excluído do banco de dados, mas é impedido de realizar alguma operação dentro do aplicativo, para não perder a referência do mesmo em algum pedido já realizado.

5.1.8 Tela de Estoque

Nesta tela são exibidos os produtos salvos disponíveis no aplicativo, organizados em um formato de lista (a), contendo as informações do Título e Valor de cada produto. Ao pressionar um produto entre os exibidos na lista, o usuário é redirecionado para a tela dos detalhes do produto (b).

A tela ainda apresenta um botão flutuante que ao ser pressionado redireciona o usuário para a tela de cadastro de um produto.

Figura 18 – Tela de Estoque



Fonte: Elaborado pelo autor

A tela possui um *AppBar* que exibe seu nome, e um botão de ação que indica a pesquisa dos produtos.

Ao pressionar o botão de pesquisa, é aberta uma caixa de pesquisa que responde ao nome vinculado à cada produto.

5.1.9 Tela de Cadastro de Produtos

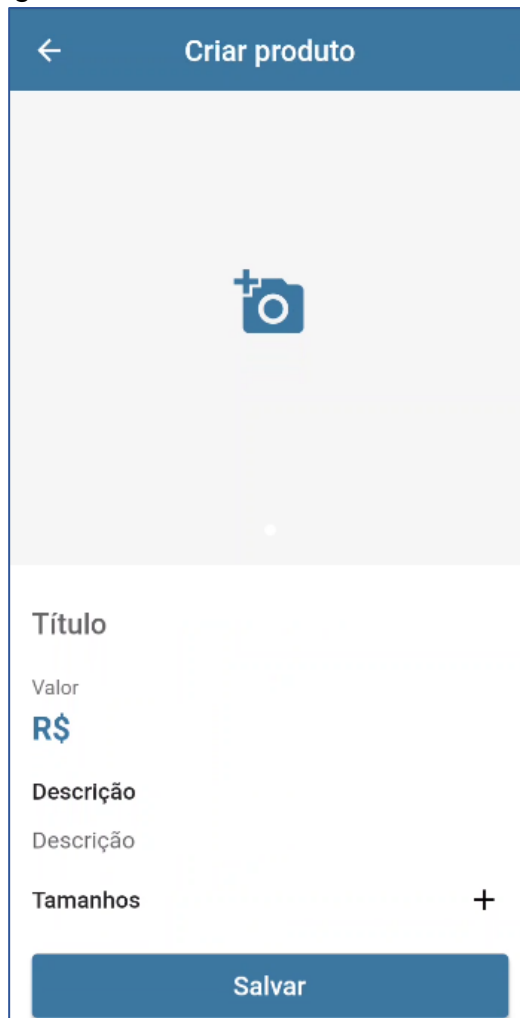
A Tela de Cadastro de Produto, Figura 19 permite o usuário cadastrar um novo produto no aplicativo. Esta tela possui os componentes:

- Imagem: Usuário deve adicionar uma imagem do produto;
- Título: Usuário deve informar o título do produto;
- Descrição: Usuário deve informar uma descrição sobre o produto;
- Tamanhos: Usuário deve informar os tamanhos do produto. A adição é realizada no formato de lista de valores do tamanho e estoque.

Ao pressionar o botão “Salvar”, o aplicativo irá validar os campos informados, exibindo uma mensagem de erro para cada campo, se necessário.

Após validar, o produto é cadastrado e o usuário é redirecionado para a tela de estoque, que contém os produtos já cadastrados.

Figura 19 – Tela de Cadastro de Produto

A interface de usuário para o cadastro de um produto. No topo, há uma barra azul com uma seta para trás e o texto "Criar produto". Abaixo, uma área cinza contém um ícone de câmera com um sinal de mais, indicando a adição de uma imagem. O formulário principal é branco e contém campos para "Título", "Valor" (com o símbolo "R\$" em azul), "Descrição" e "Tamanhos" (acompanhado de um ícone de mais). Um botão azul "Salvar" está na base.

Fonte: Elaborado pelo autor

5.1.10 Tela de Edição de Produtos

Esta tela, Figura 20, permite o usuário editar e excluir um produto já cadastrado no aplicativo.

A tela possui os mesmos componentes da Tela de Cadastro de Produtos, que são: imagem, título, descrição e tamanhos.

No entanto, possui o botão de Exclusão, que ao ser pressionado, marca o produto como deletado. Uma vez marcado, o produto não é excluído do banco de dados, mas é impedido de realizar alguma operação dentro do aplicativo, para não perder a referência do mesmo em algum pedido já realizado.

Figura 20 – Tela de Edição do Produto

Sandália Hortênsia

Valor
R\$ 139.5

Descrição
Feita pra você: mulher moderna que quer estar sempre elegante e linda em qualquer ocasião

Tamanhos +

Numeração	Estoque	
36	2	✗
Numeração	Estoque	
34	0	✗
Numeração	Estoque	
38	6	✗
Numeração	Estoque	
35	0	✗
Numeração	Estoque	
37	2	✗
Numeração	Estoque	
39	1	✗

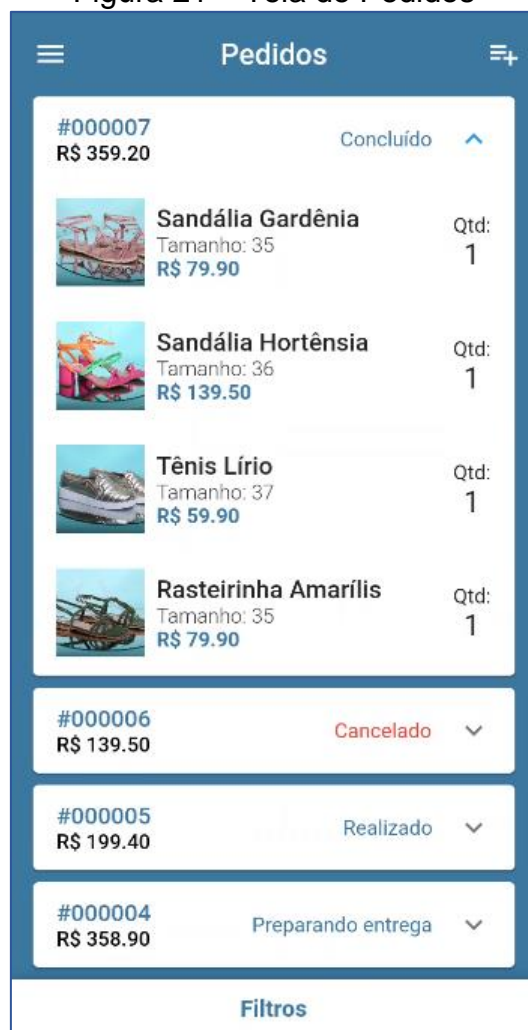
Salvar

Fonte: Elaborado pelo autor

5.1.11 Tela de Pedidos

Nesta tela são exibidos os pedidos realizados no aplicativo. Como é mostrado na Figura 21, a tela é organizada em um formato de lista de painéis deslizantes onde são fixadas as informações dos pedidos. Inicialmente, o painel exibe o Id e status do pedido. Ao pressionar, o painel se expande exibindo mais informações dos produtos vinculados ao pedido, como: imagem do produto, tamanho, valor e quantidade.

Figura 21 – Tela de Pedidos



Fonte: Elaborado pelo autor

Além disso, são exibidos os campos de alteração do status do pedido, que são definidos como: Cancelado, Realizado, Preparando entrega e Concluído.

Ao pressionar cada botão de opção, o status do pedido é alterado, seguindo as regras de negócio do aplicativo.

Figura 22 – Tela de Pedidos Filtrados



Fonte: Elaborado pelo autor

Como mostra a Figura 22, a tela pode ser organizada por filtros para cada estado de status do pedido.

5.1.12 Tela de Criação de Pedidos

A Figura 23 exibe a tela de execução de pedidos, que permite o usuário criar pedidos no aplicativo.

A tela apresenta componentes organizados em *Dropdowns* de múltipla escolha, como:

- Cliente: Campo em que o usuário deve informar o cliente vinculado ao pedido;

- Produto: Campo em que o usuário deve informar o produto vinculado ao pedido;
- Tamanho: Campo em que o usuário deve definir um tamanho do produto escolhido no pedido;
- QTD: Campo em que o usuário deve definir a quantidade de produtos que será vinculada ao pedido.

Figura 23 – Tela de Criação de Pedido

A interface da tela 'Novo Pedido' apresenta um formulário para a criação de um novo pedido. No topo, há uma barra azul com um ícone de seta para trás e o título 'Novo Pedido'. O formulário contém os seguintes campos:

- Cliente:** Um campo com o placeholder 'Defina o Cliente' e uma seta para baixo.
- Produto:** Um campo com o placeholder 'Escolha um Produto' e uma seta para baixo.
- Tamanho:** Um campo com o placeholder 'Selecione um produto'.
- QTD.:** Um campo com o valor '0' e uma seta para baixo.

Abaixo dos campos, há um botão azul com o texto 'Adicionar item ao pedido'. Na parte inferior da tela, há uma barra azul com um ícone de carrinho de compras.

Fonte: Elaborado pelo autor

Ao pressionar o botão “Adicionar item ao pedido” o aplicativo valida os campos definidos, bem como sua disponibilidade no banco de dados, exibindo mensagens de erro, caso necessário. Uma vez validado, o aplicativo armazena os dados definidos nos campos em um mapa de dados e o usuário é redirecionado para a tela de confirmação do pedido.

A tela também apresenta um botão flutuante que ao ser pressionado redireciona o usuário para a tela de confirmação do pedido sem adicionar item ao pedido.

5.1.13 Tela de Confirmação de Pedidos

A tela permite o usuário confirmar os dados que definiu na tela de criação, como mostra a Figura 24. Ao pressionar o botão “Limpar carrinho” os itens vinculados na criação do pedido voltam ao estoque e são retirados da lista de carrinho do pedido.

Figura 24 – Tela de Confirmação de Pedido



Confirmar Pedido	
Produto	Tênis Lírio
Tamanho	35
Quantidade	1
Subtotal	R\$ 59.90

Produto	Sandália Orquídea
Tamanho	37
Quantidade	1
Subtotal	R\$ 139.50

Resumo do Pedido

Quantidade de produtos 2

Total R\$ 199.40

Finalizar Pedido

Limpar carrinho

Fonte: Elaborado pelo autor

Ao pressionar o botão “Finalizar Pedido” o aplicativo realiza a validação da criação do pedido, verificando a disponibilidade do carrinho, de todos os itens, estoque e cliente escolhidos.

Caso o usuário tente finalizar o pedido com o carrinho vazio, uma mensagem de erro é exibida indicando que o carrinho não possui itens adicionados, como mostrado na Figura 25.

Figura 25 – Erro exibido ao tentar adicionar pedido vazio



Fonte: Elaborado pelo autor

Com o êxito da validação, o aplicativo envia os dados do pedido realizado para o banco de dados e o usuário é redirecionado para a tela de início.

5.1.14 Tela de Ficha Financeira

Nesta tela são exibidos os valores de entrada e saída financeira do aplicativo, como mostra a Figura 26. A tela é organizada com os componentes:

- Entrada: exibe em fração monetária os valores de entrada financeira do aplicativo, e o percentual relacionado ao mês anterior;
- Saída: exibe em fração monetária os valores de saída financeira do aplicativo e o percentual relacionado ao mês anterior;
- Lista de transações: Exibe uma lista de todas as transações do aplicativo contendo informações para cada uma, como Título da transação, Valor, Data e um ícone indicador.

Figura 26 – Tela de Ficha Financeira



Fonte: Elaborado pelo autor

O código fonte do aplicativo Donanja Gestão de Vendas está disponível no link <https://github.com/gabrielmecruz/donanja-gestao-de-vendas>.

5.2 Otimização no controle dos recursos

Um exemplo de resultado do uso do aplicativo foi a melhoria no controle dos recursos, com o estudo de caso realizado na empresa Donanja Shoes.

O controle de estoque e de clientes era realizado com o armazenamento dos dados em agendas, de forma escrita. O controle utilizando o aplicativo Donanja Gestão de Vendas é realizado de forma intuitiva e rápida, auxiliando o profissional de venda com otimização no tempo e facilidade no controle dos recursos.

6 CONSIDERAÇÕES FINAIS

Este trabalho apresentou a documentação e o desenvolvimento do aplicativo Donanja Gestão de Vendas, juntamente com as etapas de metodologia e documentações necessárias para a implementação do aplicativo proposto, abordando o processo de levantamento de requisitos do aplicativo e as tecnologias utilizadas na construção, até os resultados obtidos com ele.

O trabalho atingiu seu objetivo, e os resultados foram satisfatórios, uma vez que foi criada uma solução que contemple as principais rotinas de vendas, com comunicação e integração dos dados de forma eficiente ao usuário. Possibilitou um controle rápido e intuitivo dos recursos. Além disso, foi adquirida experiência na linguagem utilizada.

Durante o desenvolvimento dificuldades foram encontradas, como o aprendizado de uma linguagem nova, a falta de conhecimento na parte de banco de dados não relacionais e a perda de uma parte do trabalho durante o processo. E depois de concluído, foi analisado que o trabalho produziu maturidade de aprendizado.

6.1 Trabalhos futuros

- Implementar a função de definição e visualização de Metas ao aplicativo;
- Possibilitar a execução síncrona dos dados financeiros, sem a necessidade de reiniciar o aplicativo;
- Implementar a função de definição manual de entradas e saídas financeiras;
- Implementar login com serviços externos ao aplicativo, como por exemplo Gmail, Facebook, iCloud;
- Testar a versão para iOS;

REFERÊNCIAS

Agência Sebrae de Notícias. **Pequenos negócios já representam 30% do Produto Interno Bruto do país**. ASN, 2021. Disponível em:

<<https://www.agenciasebrae.com.br/sites/asn/uf/NA/pequenos-negocios-ja-representam-30-do-produto-interno-bruto-do-pais,7b965c911da51710VgnVCM1000004c00210aRCRD>>. Acesso em: 20 ago. 2021.

ANDRADE, Ana Paula. **O que é Firebase?** TreinaWeb, 2020. Disponível em:

<<https://www.treinaweb.com.br/blog/o-que-e-firebase>>. Acesso em: 19 nov. 2021.

CORAZZA, Paulo Victor. **Um aplicativo multiplataforma desenvolvido com Flutter e NoSQL para o cálculo da probabilidade de apendicite**. 2018. 58 f. Monografia (Ciência da Computação) - UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL, Porto Alegre. Disponível em:

<<https://www.lume.ufrgs.br/bitstream/handle/10183/190147/001088749.pdf?sequence=1>>. Acesso em: 09 dez. 2021.

DART. **Dart overview**. Dart dev. Disponível em: <<https://dart.dev/overview>>.

Acesso em: 29 jan. 2022.

FLUTTER. **Introduction to widgets**. Flutter dev. Disponível em:

<<https://docs.flutter.dev/development/ui/widgets-intro>>. Acesso em: 02 abr. 2022.

FLUTTER. **Flutter release notes**. 2021. Disponível em:

<<https://flutter.dev/docs/development/tools/sdk/release-notes/>>. Acesso em: 23 set. 2021.

GITHUB. **Donanja Gestão de Vendas**. Github. Disponível em: <

<<https://github.com/gabrielmecruz/donanja-gestao-de-vendas>>. Acesso em: 10 jun. 2022.

GITHUB. **Cloud Firestore**: CHANGELOG. Github. Disponível em: <https://github.com/firebase/flutterfire/blob/master/packages/cloud_firestore/cloud_firestore/CHANGELOG.md>. Acesso em: 17 mai. 2022.

GITHUB. **Firestore Core**: CHANGELOG. Github. Disponível em: <https://github.com/firebase/flutterfire/blob/master/packages/firebase_core/firebase_core/CHANGELOG.md>. Acesso em: 18 mai. 2022

GITHUB. **Firestore storage**: CHANGELOG. Github. Disponível em: <https://github.com/firebase/flutterfire/blob/master/packages/firebase_storage/firebase_storage/CHANGELOG.md#500>. Acesso em: 18 mai. 2022.

GOOGLE. **Documentação**. Firebase. Disponível em: <<https://firebase.google.com/docs?hl=pt>>. Acesso em: 27 mar. 2022.

GOVERNO FEDERAL. **Empreendedor**: Empresas e Negócios. GOV BR. Disponível em: <<https://www.gov.br/empresas-e-negocios/pt-br/empreendedor>>. Acesso em: 23 jan. 2022.

GOVERNO FEDERAL. **Governo destaca papel da Micro e Pequena Empresa para a economia do país**. gov.br, 2020. Disponível em: <<https://www.gov.br/economia/pt-br/assuntos/noticias/2020/outubro/governo-destaca-papel-da-micro-e-pequena-empresa-para-a-economia-do-pais>>. Acesso em: 17 ago. 2021.

HOSTGATOR. **Conheça tudo sobre Dart, a linguagem do Flutter**. 2021. Disponível em: <<https://www.hostgator.com.br/blog/o-que-e-dart-na-programacao/>>. Acesso em: 20 nov. 2021.

LIMA, Luan Lucas Pereira de. **Aplicativo educacional para inserção da lógica computacional para crianças**. 2017. 61 f. Monografia (Graduação em Ciência da Computação) – Universidade de Caxias do Sul, Caxias do Sul, 2017.

MACHADO, Ana Paula. **Desenvolvimento de aplicativo para pacientes com Transtorno do Espectro do Autismo História Social: Indo ao Dentista**. 2018. Monografia (Graduação em Odontologia) – UFSC, Florianópolis, 2018.

NECTARCRM. **5 Maiores desafios de vendas e como supera-los**. Blog NectarCrm. Disponível em: <<https://blog.nectarcrm.com.br/5-maiores-desafios-de-vendas-e-como-supera-los/>>. Acesso em: 09 dez. 2021.

NIX UNITED. **The pros and cons of flutter in mobile application development**. NIX United. Disponível em: <<https://nix-united.com/blog/the-pros-and-cons-of-flutter-in-mobile-application-development/>>. Acesso em: 06 fev. 2022.

OPUS SOFTWARE. **Flutter framework**. Opus software. Disponível em: <<https://www.opus-software.com.br/flutter-framework/>>. Acesso em: 02 mar. 2022.

ROCHA, Gabriel Ricardo. **Desenvolvimento de aplicativo para auxílio no tratamento fisioterapêutico a distância**. 2019. 76 f. Monografia (Graduação em Sistemas da Informação) – Universidade de Caxias do Sul, Caxias do Sul, 2019.

SILVA, M. M. DA; SANTOS, M. T. P. **Os Paradigmas de Desenvolvimento de Aplicativos para Aparelhos Celulares**. T.I.S - Tecnologias, Infraestrutura e Software - UFSCar, v.3, n.2, p. 162– 170, 2014.

VOLPI, Guilherme. **8 maiores dificuldades na gestão de micro e pequenas empresas**. Soften Sistemas, 2017. Disponível em: <<https://blog.softensistemas.com.br/8-maiores-dificuldades-na-gestao-de-micro-e-pequenas-empresas/>>. Acesso em: 19 ago. 2021.

APÊNDICE A – TERMO DE AUTORIZAÇÃO DE PUBLICAÇÃO DE PRODUÇÃO ACADÊMICA



**PUC
GOIÁS**

PONTIFÍCA UNIVERSIDADE CATÓLICA DE GOIÁS
GABINETE DO REITOR

Av. Universitária, 1000 • Serra Universitária
Cidade: Pátio 80 • CEP: 74605-010
Goiânia • Goiás • Brasil
Fone: (62) 3946.1000
www.pucgoias.edu.br • reitoria@pucgoias.edu.br

RESOLUÇÃO n° 038/2020 – CEPE

ANEXO I

APÊNDICE ao TCC

Termo de autorização de publicação de produção acadêmica

O(A) estudante Gabriel Mendes Cruz
do Curso de Engenharia de Computação, matrícula 201312033.0297-5,
telefone: 62 981228370 e-mail: gabriel.mendes@gmail.com, na qualidade de titular dos
direitos autorais, em consonância com a Lei nº 9.610/98 (Lei dos Direitos do autor),
autoriza a Pontifícia Universidade Católica de Goiás (PUC Goiás) a disponibilizar o
Trabalho de Conclusão de Curso intitulado
Desenvolvimento de um aplicativo para gestão de vendas,
gratuitamente, sem ressarcimento dos direitos autorais, por 5
(cinco) anos, conforme permissões do documento, em meio eletrônico, na rede mundial
de computadores, no formato especificado (Texto (PDF); Imagem (GIF ou JPEG); Som
(WAVE, MPEG, AIFF, SND); Vídeo (MPEG, MWV, AVI, QT); outros, específicos da
área; para fins de leitura e/ou impressão pela internet, a título de divulgação da
produção científica gerada nos cursos de graduação da PUC Goiás.

Goiânia, 16 de março de 2022.

Assinatura do(s) autor(es): Gabriel Mendes Cruz

Nome completo do autor: Gabriel Mendes Cruz

Assinatura do professor-orientador: Ludmilla R. P. dos Santos

Nome completo do professor-orientador: Ludmilla Reis Pinheiro dos Santos