

**UNIVERSIDADE FEDERAL DE MATO GROSSO DO SUL
SISTEMAS DE INFORMAÇÃO**

Luiz Henrique Andrade Souza
João Victor Munhóz dos Santos

**GESTOQUE: SISTEMA DE GERENCIAMENTO DE ESTOQUE DA EMPRESA
SOLDAMAQ**

CAMPO GRANDE

2023

Luiz Henrique Andrade Souza
João Victor Munhóz dos Santos

**GESTOQUE: SISTEMA DE GERENCIAMENTO DE ESTOQUE DA EMPRESA
SOLDAMAQ**

Trabalho de Conclusão de Curso de Sistemas de Informação da Faculdade de Computação da Universidade Federal de Mato Grosso do Sul, como requisito parcial para obtenção de diploma de bacharel em Sistemas de Informação.

Orientadora: Professora Dra. Ana Karina Dourado Salina de Oliveira

CAMPO GRANDE
2023

RESUMO

Muitas empresas realizam milhares de vendas todos os dias, em diversas localidades diferentes. O grande volume de operações, produtos e funcionários envolvidos evidencia que cada vez mais há necessidade de controle preciso e em tempo real das transações do estoque, vendas e entregas. O objetivo deste trabalho é exibir o processo de desenvolvimento de um aplicativo de gerenciamento de estoque, vendas e entregas para a empresa Soldamaq, desenvolvido para Android através da linguagem de programação Dart. Apresenta-se, assim, todo o processo de desenvolvimento, desde o levantamento de requisitos, definição das tecnologias e arquitetura, realização dos testes, entre outros procedimentos, até a entrega final da aplicação.

Palavras-chave: aplicativos Android, gerenciamento de vendas, gerenciamento de entregas, gerenciamento de estoque.

LISTA DE FIGURAS

Figura 1 - Diagrama de casos de uso (Qualquer pessoa).....	17
Figura 2 - Diagrama de casos de uso (Vendedor).....	18
Figura 3 - Diagrama de casos de uso (Entregador).....	18
Figura 4 - Diagrama de casos de uso (Administrador).....	19
Figura 5 - Diagrama de Contexto.....	31
Figura 6 - Tela de Contatos.....	49
Figura 7 - Tela de Login.....	50
Figura 8 - Tela do Menu Principal.....	51
Figura 9 - Módulo Produtos.....	53
Figura 10 - Módulo Central de Vendas.....	56
Figura 11 - Módulo Canhoto Boleto.....	58
Figura 12 - Módulo Entregas.....	60
Figura 13 - Módulo Ranking.....	61
Figura 14 - Módulo Indicadores.....	63

LISTA DE TABELAS

Tabela 1 - Funcionalidades e atores do aplicativo gEstoque.....	15
Tabela 2 - Testes de funcionalidade do aplicativo gEstoque.....	24

SUMÁRIO

1. INTRODUÇÃO.....	9
2. OBJETIVOS.....	10
2.1 OBJETIVO GERAL.....	10
2.2 OBJETIVOS ESPECÍFICOS.....	10
3. A EMPRESA SOLDAMAQ.....	11
4. REVISÃO BIBLIOGRÁFICA.....	12
4.1 PROGRAMAÇÃO ORIENTADA A OBJETOS.....	12
4.1.1 Herança.....	12
4.1.2 Abstração.....	12
4.1.3 Polimorfismo.....	13
4.1.4 Encapsulamento.....	13
5. REQUISITOS E ARQUITETURA.....	14
5.1 ANÁLISE DE CONTEXTO.....	14
5.1.1 Visão Geral.....	14
5.1.2 Condições Restritivas.....	14
5.1.2.1 Custos.....	14
5.1.2.2 Tecnológicas.....	14
5.1.3 Benefícios e Impactos.....	14
5.1.3.1 Econômicos.....	14
5.1.3.2 Operacionais.....	14
5.2 ANÁLISE DE REQUISITOS.....	15
5.2.1 Metodologia.....	15
5.2.2 Lista de Atores.....	15
5.2.3 Lista de Funcionalidades.....	15
5.3 REQUISITOS.....	16
5.3.1 Diagramas dos Casos de Uso.....	16
5.3.2 Especificação dos Requisitos de Software.....	19

5.3.2.1 Requisitos Funcionais.....	19
5.3.3 Diagrama de Classes.....	20
6. DESENVOLVIMENTO.....	20
6.1 TECNOLOGIAS.....	20
6.1.1 Dart.....	20
6.1.2 Flutter.....	21
6.2 ARQUITETURA.....	21
6.3 BANCO DE DADOS.....	22
6.4 AUTENTICAÇÃO.....	22
6.5 HOSPEDAGEM.....	22
6.6 SEGURANÇA.....	22
6.7 PROTÓTIPOS DE TELAS.....	23
7. TESTES E RESULTADOS.....	23
8. CONCLUSÕES.....	25
8.1 TRABALHOS FUTUROS.....	26
REFERÊNCIAS.....	27
ANEXO A - REQUISITOS DE NEGÓCIO.....	29
A.1 INTRODUÇÃO.....	29
A.2 BACKGROUND E OPORTUNIDADES DE NEGÓCIO.....	29
A.3 OBJETIVOS DE NEGÓCIO.....	29
A.4 VISÃO DO PRODUTO.....	30
A.6 LIMITAÇÕES.....	30
A.7 DIAGRAMA DE CONTEXTO.....	31
A.8 REGRAS DE NEGÓCIO.....	31
ANEXO B - ESPECIFICAÇÃO DOS CASOS DE USO.....	32
B.1 DESCRIÇÃO DOS ATORES.....	32
B.2 CASOS DE USO.....	32
B.2.1 Caso de Uso: Consultar Produto.....	32
B.2.2 Caso de Uso: Visualizar Descrição do Produto.....	33

B.2.3 Caso de Uso: Visualizar Estoque do Produto.....	33
B.2.4 Caso de Uso: Fazer Login.....	34
B.2.5 Caso de Uso: Consultar Ranking.....	35
B.2.6 Caso de Uso: Consultar Indicadores.....	36
B.2.7 Caso de Uso: Realizar Entregas.....	36
B.2.8 Caso de Uso: Cadastrar Cabeçalho.....	38
B.2.9 Caso de Uso: Adicionar Produto ao Carrinho.....	39
B.2.10 Caso de Uso: Confirmar Transação.....	40
B.2.11 Caso de Uso: Pesquisar Boletos.....	41
B.2.12 Caso de Uso: Registrar Foto do Canhoto.....	42
ANEXO C - ESPECIFICAÇÃO DOS REQUISITOS DE SOFTWARE.....	43
C.1 INTRODUÇÃO.....	43
C.2 REQUISITOS DE SOFTWARE.....	43
C.2.1 Requisitos Funcionais.....	43
C.2.1.1 Controle de Acesso.....	43
C.2.1.2 Dashboard.....	43
C.2.1.3 Ações.....	44
C.2.1.4 Gestão do Estoque.....	44
ANEXO D - TESTES DAS FUNCIONALIDADES.....	48
D.1 AMBIENTE DE TESTES.....	48
D.2 TESTES.....	48
D.2.1 Teste: Tela de Contatos.....	48
D.2.2 Teste: Tela de Login.....	49
D.2.3 Teste: Tela do Menu Principal.....	50
D.2.4 Teste: Módulo Produtos.....	51
D.2.5 Teste: Módulo Central de Vendas.....	54
D.2.6 Teste: Módulo Canhoto Boletos.....	57
D.2.7 Teste: Módulo Entregas.....	59
D.2.8 Teste: Módulo Ranking.....	61

D.2.9 Teste: Módulo Indicadores.....	62
---	-----------

1. INTRODUÇÃO

A empresa Soldamaq atua em 7 cidades de Mato Grosso do Sul, possuindo mais de 200 funcionários distribuídos em mais de 12 lojas. O foco da empresa é na venda de equipamentos e suprimentos para a construção civil, indústria, agronegócio e pequenas empresas (SOLDAMAQ. Acesso em 11-03-2023).

A empresa migrou para o sistema ERP (Sistema Integrado de Gestão Empresarial) Sankhya para gerenciar os produtos e funcionários, porém, ainda se faz necessário o desenvolvimento de um aplicativo preciso e eficiente a ser utilizado pelos funcionários das lojas o qual gerencie a saída de produtos do estoque e as entregas de maneira organizada.

Neste cenário um funcionário responsável pelo estoque tem muito trabalho, pois o estoque é muito grande e a cada produto verificado, o mesmo deve ir ao computador para registrar, uma aplicação móvel facilita e aumenta a produtividade, sendo este o motivo pelo nome da aplicação “gEstoque”, gestão de estoque .

A ausência desse sistema gera riscos de erros nos processos da empresa e por parte dos funcionários, diminui a produtividade e pode ocasionar um descontrole total para o estoque da empresa em todas as suas lojas. Aumenta também o risco de furtos e extravios de produtos por funcionários.

Ante o exposto, este trabalho propõe o desenvolvimento e implementação de um aplicativo de gerenciamento de estoque e vendas para a empresa Soldamaq, o qual possa ser utilizado pelos funcionários para controlar em tempo real a saída dos produtos e os dados de pagamento, desenvolvido na linguagem de programação Dart e incorporado ao sistema ERP Sankhya, já em uso pela empresa.

2. OBJETIVOS

2.1 OBJETIVO GERAL

Este trabalho tem como objetivo desenvolver um *software*, para os funcionários da empresa Soldamaq, de gerenciamento de estoque e vendas, implementado para dispositivos móveis com o Sistema Operacional Android, inicialmente criado apenas para gestão de estoque, porém, ao decorrer do projeto foi se tornando maior, e seu nome mantido gEstoque um trocadilho referenciando gestão de estoque.

2.2 OBJETIVOS ESPECÍFICOS

- Definir a especificação dos requisitos do sistema conforme as necessidades descritas pelos funcionários e administradores.
- Definir e registrar todas as tecnologias utilizadas para o desenvolvimento da aplicação, buscando as tecnologias que atendam aos requisitos do sistema.
- Documentar e modelar os casos de uso do sistema, conforme as necessidades especificadas nos requisitos.
- Modelar as telas a serem implementadas.
- Realizar e documentar testes no sistema e na implementação, garantindo que estão de acordo com os requisitos, identificando possíveis problemas no sistema.

3. A EMPRESA SOLDAMAQ

Atualmente muito presente no mercado do Mato Grosso do Sul, a Soldamaq conta com 12 lojas espalhadas pelo estado, e 30 segmentos de produtos oferecidos para o cliente, porém, como toda empresa tem o seu início, a história não é diferente para a Soldamaq. Fundada em agosto de 1984, nasceu a ideia de trazer uma empresa completa de ferramentas para a cidade de Campo Grande.

Apesar de diversidades e desafios presentes em um início empresarial, a Soldamaq teve seu crescimento no mercado e reconhecimento no ramo de atuação. Ao passar dos anos, a empresa foi se estruturando cada vez mais e podendo abrir diversas lojas pelo estado de Mato Grosso do Sul.

A Soldamaq comercializa mais de 37.000 (trinta e sete mil) itens, entre máquinas, ferramentas, EPIs (Equipamentos de Proteção Individual), equipamentos, parafusos, itens para operações de reformas, obras e casas. Devido ao grande número de produtos vendidos diariamente em todas as lojas pelo estado, é imprescindível a implementação de um sistema de fácil acesso a todos os funcionários que realize e gerencie as transações de saída do estoque, reduzindo significativamente o risco de inconsistências no gerenciamento dos produtos.

4. REVISÃO BIBLIOGRÁFICA

Este trabalho utiliza diversas tecnologias e tipos de programação relacionadas a aplicações Android. Neste capítulo serão exibidos os diferentes conceitos e informações relevantes utilizados no desenvolvimento do aplicativo. Na seção 4.1, serão apresentados os conceitos de Orientação a Objetos utilizados no desenvolvimento da aplicação Android.

4.1 PROGRAMAÇÃO ORIENTADA A OBJETOS

O conceito de orientação a objetos se refere a um paradigma de programação que aproxima os conceitos existentes no mundo real e o software desenvolvido, através de objetos com suas próprias características e comportamentos distintos, definidos através de métodos e classes. Surgiu como uma alternativa a programação estruturada e é muito utilizado devido, entre outras coisas, a sua organização e facilidade de manutenção do código (NOVAGEO. Acesso em 22-03-2023).

A linguagem de programação Dart, utilizada no desenvolvimento desse aplicativo, faz uso desse paradigma de programação. A seguir estão apresentados os quatro principais conceitos utilizados por esse paradigma: herança, abstração, polimorfismo e encapsulamento.

4.1.1 Herança

Trata-se da criação de novas classes no código as quais herdam o comportamento e características de uma classe já existente, denominada superclasse, especializando a classe herdada com características próprias (MARTIN, 2009). Um exemplo: a classe Funcionário pode ser herdada pelas classes Gerente ou Vendedor, ambas as quais compartilham as características apresentadas em Funcionário, em adição às suas próprias.

4.1.2 Abstração

Refere-se a capacidade de abstrair as classes e seus métodos, tornando-as impossíveis de serem utilizadas para instanciar objetos, e servindo unicamente como uma superclasse para que suas subclasses compartilhem características em comum. Essas classes utilizam métodos abstratos, mantendo os atributos que serão herdados por suas classes filhas, mas sem qualquer implementação, devendo ser caracterizados e implementados somente nas subclasses (DEITEL, 2017).

4.1.3 Polimorfismo

É a capacidade de objetos de subclasses diferentes, que compartilham uma mesma superclasse, responderem às mesmas chamadas generalizadas de métodos. Cada subclasse implementa e caracteriza o método à sua maneira e, assim, um único método pode ser chamado em diferentes subclasses, com cada objeto respondendo conforme sua implementação. Isso traz flexibilidade e simplicidade ao código (DEITEL, 2017).

4.1.4 Encapsulamento

Refere-se à capacidade de uma classe de omitir seus atributos e métodos, definindo-os como “*private*”, podendo interagir com os outros elementos de sua classe, mas estando ocultos aos objetos de outras classes. Assim, o acesso aos atributos é restringido unicamente ao próprio objeto, com os demais devendo utilizar métodos públicos *getters* (métodos que retornam o valor dos atributos) e *setters* (métodos que modificam o valor dos atributos) para acessar os mesmos. Isso impede modificações indevidas e aumenta a segurança e integridade do *software* (DEITEL, 2017).

5. REQUISITOS E ARQUITETURA

5.1 ANÁLISE DE CONTEXTO

5.1.1 Visão Geral

O aplicativo gEstoque será responsável, principalmente, por registrar as vendas e entregas dos produtos da empresa Soldamaq. Assim, a aplicação deve manter o registro de todos os itens em estoque, suas vendas e entregas e todos os outros dados relacionados aos mesmos, como descrições dos produtos, canhoto dos boletos, indicadores de vendas, de modo a manter o controle das operações realizadas pelos funcionários.

5.1.2 Condições Restritivas

5.1.2.1 Custos

A aplicação é desenvolvida internamente pela empresa e possui baixo custo, uma vez que está incorporada ao já implementado sistema ERP Sankhya, utilizando seu banco de dados e suas informações de gerenciamento em tempo real.

5.1.2.2 Tecnológicas

O sistema é exclusivo para sistemas Android, com a versão mínima (*minSdkVersion*) sendo Android 8.1 (API de nível 27) e a versão desejada (*targetSdkVersion*) Android 13 (API de nível 33). O sistema obrigatoriamente requer acesso à internet.

5.1.3 Benefícios e Impactos

5.1.3.1 Econômicos

A implementação do aplicativo irá auxiliar no controle e gerenciamento das saídas de produtos do estoque, registrando todas as operações efetuadas e prevenindo roubos ou extravio de materiais.

5.1.3.2 Operacionais

A aplicação será benéfica para a rotina da empresa, facilitando as transações efetuadas pelos vendedores e entregadores que poderão realizar suas operações de modo automatizado pelo próprio celular.

5.2 ANÁLISE DE REQUISITOS

5.2.1 Metodologia

Para o desenvolvimento do aplicativo, foi realizado o levantamento de requisitos principalmente através de reuniões e conversas com funcionários da empresa (vendedores, entregadores e gerentes). Novos requisitos surgiram após a primeira entrega e início das operações do sistema e dos testes realizados.

5.2.2 Lista de Atores

Na especificação dos Requisitos de Usuário, foram identificados três classes de usuários(atores) a serem implementados no aplicativo. Todas as classes compartilham diversas características entre si, porém possuem funcionalidades exclusivas. As classes de usuários estão listadas a seguir:

- **Vendedor** - Usuário primário do sistema. Possui permissão para visualizar os rankings e metas, realizar orçamentos e executar a venda dos produtos aos clientes.
- **Entregador** - Usuário primário do sistema. Realiza e registra a entrega dos produtos no endereço dos clientes e possui permissão para a gerar o canhoto da entrega.
- **Administrador** - Possui permissões de administrador do sistema, pode acessar o canhoto dos boletos e registrar a foto do canhoto.

5.2.3 Lista de Funcionalidades

As funcionalidades desenvolvidas para aplicação e os atores responsáveis estão registrados na Tabela 1:

Tabela 1 - Funcionalidades e atores do aplicativo gEstoque.

Funcionalidade	Ator(es)
Consultar Produto	Qualquer Usuário
Visualizar Descrição do Produto	Qualquer Usuário
Visualizar Estoque do Produto	Qualquer Usuário

Funcionalidade	Ator(es)
Fazer Login	Qualquer Usuário
Consultar Ranking	Vendedor
Consultar Indicadores	Vendedor
Realizar Entregas	Entregador
Cadastrar Cabeçalho	Qualquer Usuário
Adicionar Produto ao Carrinho	Qualquer Usuário
Confirmar Transação	Qualquer Usuário
Pesquisar Boletos	Administrador
Registrar Foto do Canhoto	Administrador

5.3 REQUISITOS

5.3.1 Diagramas dos Casos de Uso

Após a elicitação e especificação dos requisitos de usuários, realizada principalmente por entrevistas e observações, foram desenvolvidos Casos de Uso de acordo com os requisitos e funcionalidades levantadas.

A especificação dos casos de uso define como os usuários interagem com o sistema, conforme cada funcionalidade exigida e permite a descoberta de novos requisitos a serem implementados, de modo que a aplicação se torne eficiente e prática para os usuários.

Assim, os casos de uso são especificados de forma detalhada e modelados em diagramas de forma simples, permitindo a fácil comunicação e entendimento entre o desenvolvedor e o cliente. A linguagem de modelagem unificada (UML) é a principal ferramenta utilizada para criação dos diagramas de caso de uso.

Os Diagramas mostram a visão geral da aplicação e as interações entre os atores (usuários) e as diferentes funcionalidades do sistema, de forma simples e direta. Assim, os diagramas não são detalhados, facilitando o entendimento e consenso entre as partes interessadas

Os diagramas modelados podem incluir *includes* e *extends*. Caso um determinado caso de uso necessita da execução de um segundo caso de uso para ser executado, dizemos que ele “inclui” o segundo caso de uso, ocorrendo um relacionamento do tipo *include*. Quando um caso de uso pode, opcionalmente, executar um segundo caso de uso, mas não é um requisito obrigatório, ocorre um relacionamento do tipo *extends* (LUCIDCHART. Acesso em: 01-07-2023).

Para a modelagem dos diagramas de casos de uso, foi utilizada a ferramenta Lucid-chart, disponível em lucidchart.com. A Especificação dos Casos de Uso se encontra registrada no Anexo B, e foi baseada para elaboração das Figuras 1 a 4, exibidas a seguir, que representam os diagramas de casos de uso separados pelos respectivos atores.

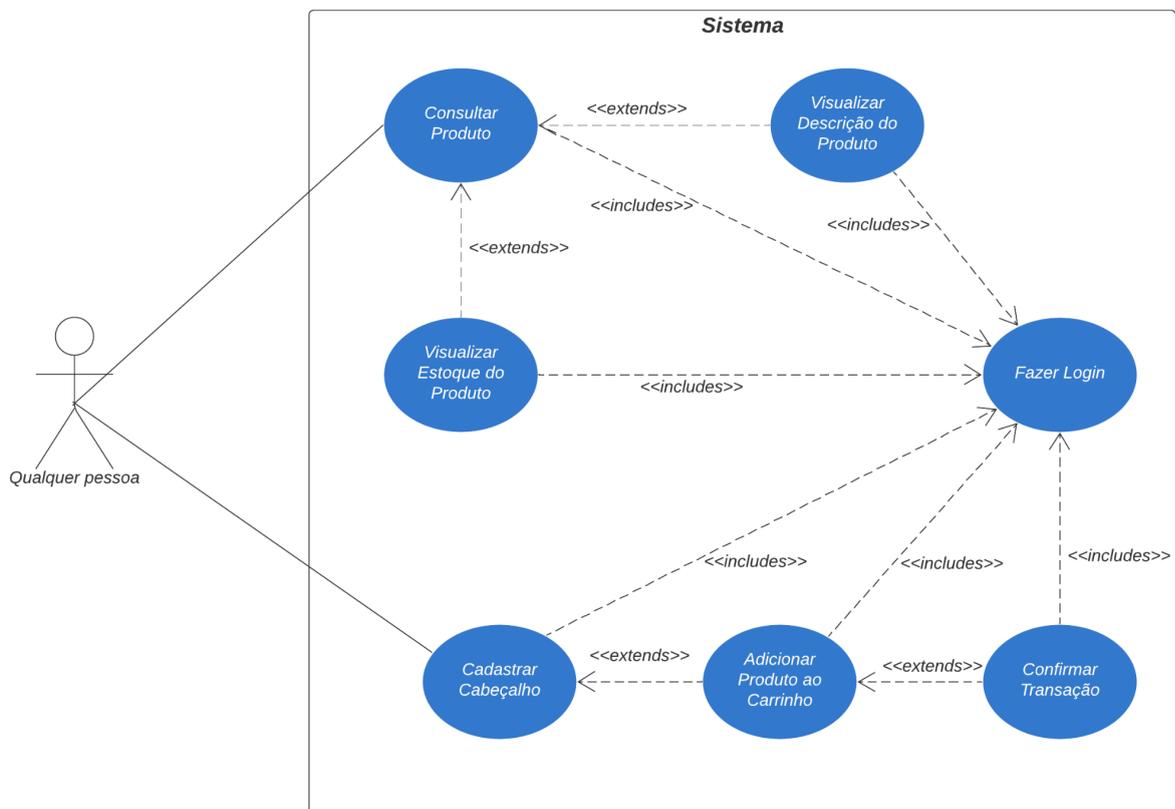


Figura 1 - Diagrama de casos de uso (Qualquer pessoa)

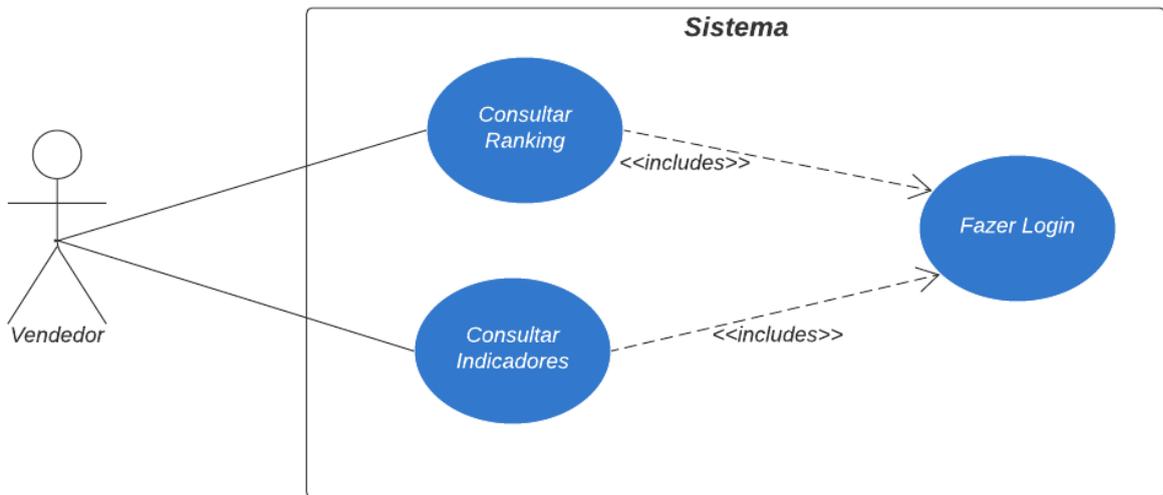


Figura 2 - Diagrama de casos de uso (Vendedor)

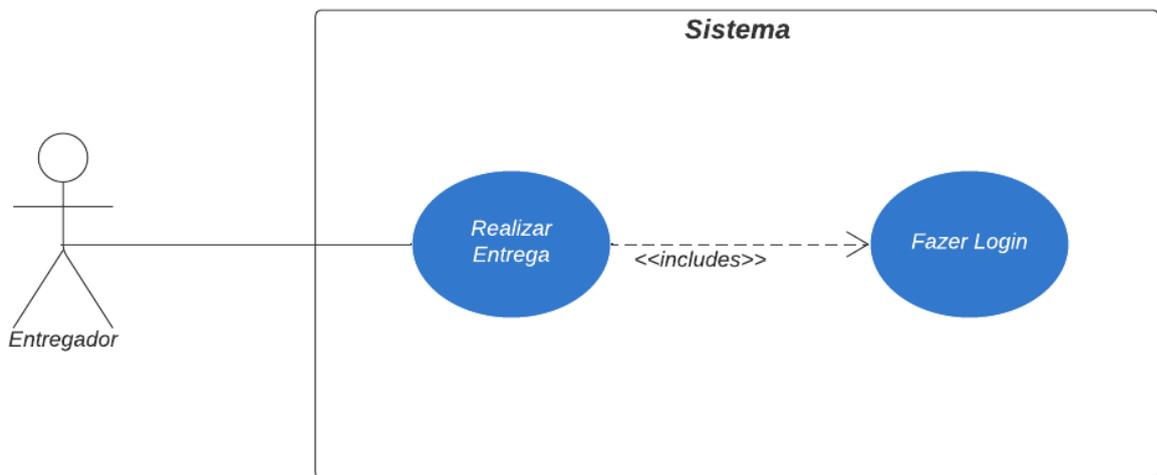


Figura 3 - Diagrama de casos de uso (Entregador)

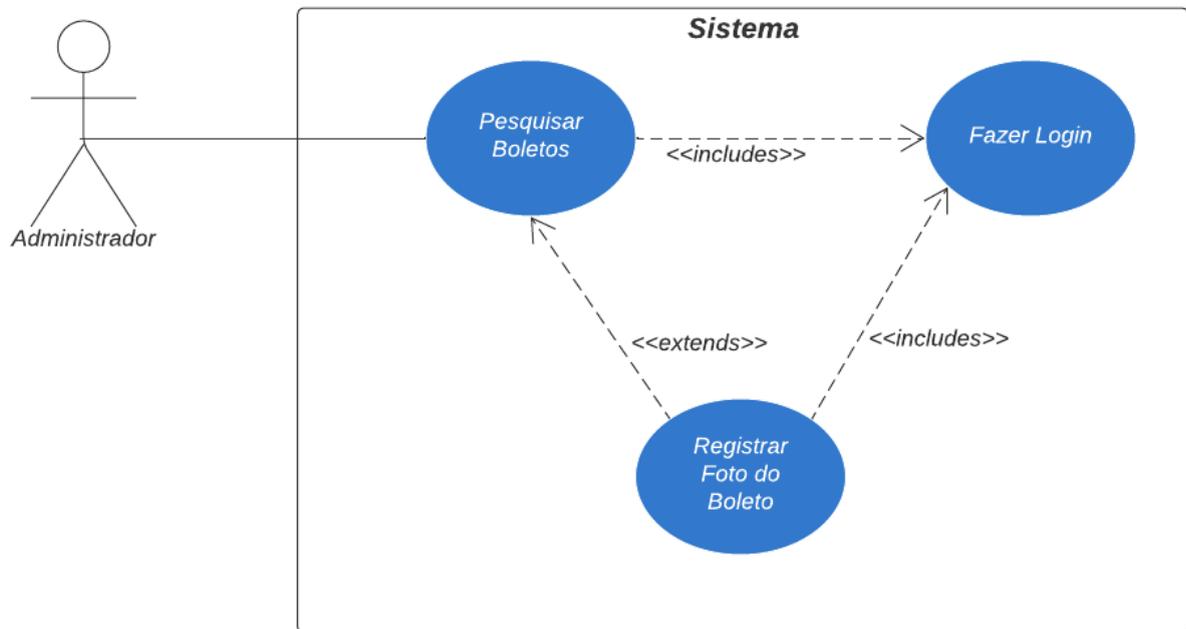


Figura 4 - Diagrama de casos de uso (Administrador)

5.3.2 Especificação dos Requisitos de Software

A especificação dos requisitos de *software* refere-se a etapa do processo em que são identificadas e definidas as necessidades e exigências a serem atendidas pela aplicação, bem como suas restrições. A elicitação dos requisitos pode-se dar através de reuniões, entrevistas, prototipagem, questionários, e diversos outros métodos. Os requisitos comuns são identificados por meio de métodos como entrevistas e observações. Após essa coleta de informações, as demandas dos usuários são registradas em uma tabela, permitindo ao analista validar os requisitos previstos e alinhá-los com as expectativas do cliente (PRESSMAN, 2006).

Os requisitos são divididos em Requisitos Funcionais e Não-Funcionais. Os Funcionais se referem às características das funcionalidades do sistema, já os Não-Funcionais se referem a restrições e características de portabilidade, segurança, confiabilidade, usabilidade, desempenho, entre outros (MONITORATEC. Acesso em: 15-06-2023).

5.3.2.1 Requisitos Funcionais

Os requisitos funcionais deste trabalho foram identificados e divididos nas seguintes categorias:

- Controle de Acesso: Define as permissões de usuário.
- Dashboard: Define as operações que podem ser executadas por cada usuário.
- Ações: Define ações que podem ser executadas em todo o sistema por qualquer usuário.
- Gestão de Estoque: Define a administração e o gerenciamento do estoque e as demais funcionalidades do aplicativo.

O documento de Especificação de Requisitos de Software encontra-se no Anexo A

5.3.3 Diagrama de Classes

O aplicativo está integrado ao sistema ERP Sankhya, fazendo uso de sua base de dados. Uma vez que serve como um complemento ao sistema já existente, o aplicativo desenvolvido não dispõe dos tradicionais relacionamentos entre classes de forma independente, dependendo completamente do sistema ao qual é agregado .

Assim, foi optado por simplificar a arquitetura do aplicativo através de uma estrutura mais funcional, focando na integração e comunicação com o sistema ERP, e eliminando a necessidade de construir uma nova estrutura de classes. Desta maneira, não é possível modelar um diagrama de classes do aplicativo, já que o mesmo faz uso deste recurso. Isso não apenas simplifica o desenvolvimento, mas também facilita os processos de teste do *software*.

6. DESENVOLVIMENTO

6.1 TECNOLOGIAS

6.1.1 Dart

Para o desenvolvimento do gEstoque, foi escolhida a linguagem de programação Dart, criada pela Google em 2011 com objetivo de substituir o JavaScript como a linguagem principal de desenvolvimento *web*, mas não foi bem aceita pela comunidade. Com os anos, após o lançamento do *framework* Flutter, tornou-se uma das principais linguagens para desenvolvimento ágil multiplataforma, estando amplamente presente no mercado (CAGOL, Acesso em: 23-07-2023).

6.1.2 Flutter.

Flutter é um *framework* (ferramenta) criada pela Google em 2015, utilizando a linguagem de programação Dart, com objetivo de facilitar o desenvolvimento multiplataforma em dispositivos móveis. É muito ágil e versátil, poupando tempo e dinheiro de seus usuários, e sendo muito utilizado por grandes empresas como Google, Nubank, entre muitas outras (AWS AMAZON. Acesso em: 25-07-2023).

6.2 ARQUITETURA

Model-View-Controller (MVC) um dos modelos mais utilizados no desenvolvimento de *softwares* para os aplicativos mobile é o padrão de projeto que utilizamos . É constituído por 3 camadas (GAMMA, 1995). A camada *View* onde é demonstrado todo o *front-end*, envolve o usuário e suas interações. A camada *Model* é a camada onde os dados são tratados para melhor utilização na aplicação, sendo estes dados na maior parte retornados de uma API e modelados e facilitando a utilização na camada *Controller*. A camada *Controller* é a camada onde é feita a comunicação com as demais camadas e é utilizada para gerenciamento da aplicação, nesta camada as requisições são recebidas e enviadas à camada *Model* para serem tratadas e por fim apresentadas na camada *View* (TreinaWeb. Acesso em: 23-07-2023).

Na aplicação desenvolvida, a camada *View* contém os arquivos de layout das telas, a camada *Model* tem o mapeamento do JavaScript de retorno da API do Sistema ERP Sankhya e *Controller* gerencia as ações da tela e o *Back-end* do sistema, os arquivos de cores, configurações de URL e padrões estão armazenados em uma camada extra denominada repository.

O Aplicativo utiliza o Gradle para automatização de *build*. O Gradle é uma ferramenta de automação de construção (*build automation*), utilizada principalmente para compilar e construir projetos de *software*. Ele oferece uma linguagem de script flexível baseada em Groovy ou Kotlin para descrever as configurações do projeto. O Gradle é amplamente utilizado em projetos Java, mas também suporta várias outras linguagens e plataformas como o Dart, a linguagem utilizada no projeto.

Em aplicações desenvolvidas em Dart, principalmente quando o framework Flutter é utilizado, o arquivo de configurações denominado “pubspec.yaml” é essencial. Estão contidos neste arquivo: configurações do projeto, como as dependências do projeto, nome do projeto,

versão, descrição, ambiente, dependências para o desenvolvimento e os assets do projeto. Assets do projeto são as imagens e fontes, onde elas estarão localizadas.

6.3 BANCO DE DADOS

O aplicativo gEstoque não utiliza um banco de dados próprio, pois integra-se ao sistema ERP Sankhya Versão 4.18, o qual armazena todos os dados e informações do estoque, funcionários e parceiros, utilizando o Oracle Database 11g Release 11.2.0.4.0 - 64bit Production, um dos mais eficientes e escaláveis bancos de dados disponíveis no mercado (ORACLE. Acesso em: 21-08-2023).

6.4 AUTENTICAÇÃO

A aplicação desenvolvida necessita de uma forma de autenticação dos usuários. Para tanto, foi selecionado o protocolo de autorização OAuth 2.0. Trata-se de um padrão amplamente utilizado que consiste na utilização de tokens de acesso para conceder diferentes tipos de permissões aos usuários. O protocolo faz uso de quatro atores: proprietário do recurso, cliente, servidor de autorização e servidor de recursos. Cada cliente possui um identificador único armazenado no servidor de autorização .

A autorização de acesso é inicialmente solicitada pelo cliente ao servidor de autorização através dos dados de identificação. Em seguida, o cliente é autenticado pelo servidor e as permissões validadas. O proprietário do recurso concede autorização pelo servidor de autorização, que direciona um token de acesso ao cliente. Por fim, o cliente utiliza o token para acessar o servidor de recursos e enfim a aplicação (AUTH0. Acesso em: 10-09-2023).

6.5 HOSPEDAGEM

O gEstoque é um aplicativo exclusivo para Android. Assim, está hospedado unicamente na plataforma Google Play, ambiente disponibilizado pela Google para baixar e atualizar aplicativos para Android, além de possuir ferramentas para monitoramento de desempenho, avaliações dos usuários, entre outros recursos.

6.6 SEGURANÇA

Para segurança, foi utilizado o algoritmo *hash* MD5(Message Digest Algorithm 5), um protocolo criptográfico que consiste na geração de um valor *hash* unidirecional de 128 bits. O

algoritmo utiliza o valor *hash* para verificar a autenticidade e integridade dos dados recebidos (AGIWORK. Acesso em 15-09-2023).

Nossa aplicação é do tipo consumidora, portanto ela não tem integração direta com o Banco de Dados, sendo assim a segurança é aplicada na hora do consumo utilizando uma autenticação criptografada para geração do *token* de sessão. Toda a comunicação entre o cliente e o servidor, especialmente durante o processo de autenticação(*login*), ocorre através de conexões seguras. Isso ajuda a proteger contra ataques de intermediários.

6.7 PROTÓTIPOS DE TELAS

No desenvolvimento de um software, muitas vezes são elaborados protótipos das telas a serem implementadas, seguindo o que foi definido nos requisitos do sistema e nos casos de uso, e incorporando elementos de design para validação com o cliente antes da implementação.

Uma das ferramentas mais utilizadas para este fim é o Figma, disponível em [figma.com](https://www.figma.com), que foi utilizado neste trabalho para prototipação das telas do gEstoque. As telas desenvolvidas para o gEstoque estão exibidas no link:

<https://www.figma.com/file/ek510DxQNmk4CIA3oS11HI/Aplicativo-GESTOQUE?type=design&node-id=0-1&mode=design>

7. TESTES E RESULTADOS

No desenvolvimento, é imprescindível a realização de testes, os quais verificam a integridade e eficiência do código. Um sistema ativo deve manter a consistência em suas ações, de modo a atender os requisitos de negócio. Deste modo, deve-se analisar os procedimentos do sistema visando identificar quaisquer falhas ou inconsistências nos dados, através de testes que verifiquem o comportamento do sistema diante de situações esperadas e inesperadas.

Verifica-se também, se todas as restrições e funcionalidades especificadas foram adequadamente implementadas, garantindo que o sistema atende a todos os requisitos de *software* especificados e segue os casos de uso definidos.

Os testes para o gEstoque foram realizados no Android Studio, utilizando um dispositivo Google Pixel 3a, Android 13 (API level 33). Foi verificado o comportamento do

sistema e validadas as ações esperadas em cada situação. As descrições e resultados dos testes, e os requisitos correspondentes encontram-se no Anexo C.

De acordo com os testes efetuados, observamos os seguintes resultados, conforme apresentados na Tabela 2:

Tabela 2 - Testes de funcionalidade do aplicativo gEstoque.

Teste	Requisitos Atendidos	Resultado
Tela de Contatos	RF4	Todas as funcionalidades do módulo funcionam corretamente
Tela de Login	RF1 e RF5	Todas as funcionalidades do módulo funcionam corretamente
Tela do Menu Principal	RF6, RF7, RF8, RF9 e RF10.	Todas as funcionalidades do módulo funcionam corretamente
Módulo Produtos	RF16, RF17, RF18, RF19 e RF20.	Todas as funcionalidades do módulo funcionam corretamente
Módulo Central de Vendas	RF21, RF22, RF23, RF24, RF25, RF26, RF27, RF28, RF29, RF30, RF31 e RF32.	Todas as funcionalidades do módulo funcionam corretamente
Módulo Canhoto Boletos	RF13, RF14 e RF15	Todas as funcionalidades do módulo funcionam corretamente
Módulo Entregas	RF38, RF39, RF40, RF41, RF42 e RF43.	Todas as funcionalidades do módulo funcionam corretamente
Módulo Ranking	RF33	Todas as funcionalidades do módulo funcionam corretamente
Módulo Indicadores	RF34, RF35, RF36 e RF37.	Todas as funcionalidades do módulo funcionam corretamente

O aplicativo foi apresentado aos funcionários da empresa e, de acordo com o relato destes, o sistema atendeu as expectativas e irá alavancar os processos de venda, entrega e gerenciamento de estoque nas lojas.

8. CONCLUSÕES

Neste trabalho, a maior dificuldade foi inicialmente a falta de familiaridade com o cenário da empresa. Também houve dificuldades na parte estrutural do sistema, devido a falta de experiência e conhecimento prévio relativo a alguns aspectos de programação. Assim, tivemos que nos familiarizar com este panorama, de modo a desenvolver o trabalho de forma satisfatória.

O aplicativo “gEstoque” já está em funcionamento e se mostrou prático e eficiente para os processos da empresa, auxiliando os vendedores e entregadores a executarem suas funções de maneira promissora, bem como mantendo o controle da disponibilidade e saída de produtos do estoque. Assim, o projeto gerou agilidade, transparência e eficiência nas operações e interações com os parceiros da empresa.

O projeto exigiu que os membros da equipe se familiarizassem e trabalhassem com ferramentas e linguagens de programação sem prévia experiência, necessitando estudo e esforço adicional para manter a qualidade do projeto.

Após alguns meses de uso, a aplicação parou de funcionar por alguns minutos e, mesmo que breve, sua ausência foi sentida de maneira significativa conforme relatos de colaboradores, demonstrando sua extrema importância no desempenho das operações da empresa.

O contato já existente de um dos membros da equipe com a empresa foi imprescindível para manter a comunicação e o desenvolvimento do aplicativo de forma ágil e eficiente. A linha de comunicação permitiu a tomada de decisões e a solução de dúvidas ao longo do desenvolvimento, garantindo a qualidade do projeto conforme os requisitos do negócio.

Finalmente, o projeto serviu como experiência prática essencial para familiarização dos membros do grupo com um ambiente de desenvolvimento dentro do mercado de trabalho. Foi possível experienciar de maneira prática os desafios e dificuldades enfrentadas pelo desenvolvedor durante os projetos, enfatizando a importância de adaptar-se de forma ágil e eficiente a quaisquer problemas e imprevistos encontrados.

8.1 TRABALHOS FUTUROS

Para trabalhos futuros, pode-se expandir a aplicação para outras plataformas e sistemas operacionais, como o IOS ou para *Web*, incluir novos ajustes e funcionalidades conforme os requisitos da empresa e dos usuários, e a manutenção geral do *software*. Pode-se ainda trabalhar na automatização de testes, tornando ainda mais segura e rápida a validação de futuras implementações.

O aplicativo serviu como um complemento ao sistema ERP já instalado, portanto não foi visado durante o desenvolvimento a construção de um *software* contendo todas as funcionalidades normalmente esperadas de um sistema de estoque. Assim, funções como cadastro de produtos, cadastro de usuários, gerenciamento de permissões, entre outros não foram desenvolvidas na aplicação, devido o ERP já contemplar estas funcionalidades citadas. Caso haja interesse pela empresa, as funcionalidades mencionadas podem ser implementadas em um projeto futuro.

REFERÊNCIAS

- AGIWORK. **Criptografia MD5: O que é, Por que é tão Importante?**. Disponível em:
<<https://www.agiwork.com.br/artigo/criptografia-md5-o-que-e-por-que-e-tao-importante>>.
Acesso em: 15/09/2023.
- AUTH0. **O que é OAuth 2.0?**. Disponível em:
<<https://auth0.com/pt/intro-to-iam/what-is-oauth-2>>. Acesso em: 10-09-2023.
- AWS AMAZON. **What is Flutter?**. Disponível em:
<<https://aws.amazon.com/pt/what-is/flutter/>>. Acesso em: 25-07-2023.
- DEITEL, Paul J.. **Java: Como programar? - 10ª Edição**. Pearson Education, 2017.
- GAMMA, Erich et al. **Elements of Reusable Object-Oriented Software**. Design Patterns, 1995.
- LUCIDCHART. **Diagrama de Caso de Uso UML**. Disponível em:
<<https://www.lucidchart.com/pages/pt/diagrama-de-caso-de-uso-uml>>. Acesso em:
01-07-2023.
- MARTIN, Robert C. **Clean code: a handbook of agile software craftsmanship**. Pearson Education, 2009.
- MONITORATEC. **Especificação de Requisitos de Software**. Disponível em:
<<https://www.monitoratec.com.br/blog/especificacao-de-requisitos-de-software/>>. Acesso em: 15-06-2023.
- NOVAGEO. **O que é Programação Orientada a Objetos (POO)?** Disponível em:
<https://www.novageo.pt/novageo/displayArticles?numero=38409&_que_programacao_orientada_objetos_poo>. Acesso em: 22-03-2023.
- ORACLE. **Documentação do Oracle**. Disponível em:
<https://docs.oracle.com/cd/E11882_01/server.112/e41360/chapter1_11204.htm#NEWFT379>. Acesso em: 21-08-2023.
- CAGOL, Paulo J.. **Introdução à Linguagem Dart**. Disponível em:
<<https://www.paulocagol.dev.br/00004-introducao-linguagem-dart/>>. Acesso em:
23-07-2023.

PRESSMAN, Roger S. **Engenharia de Software**. 6^a ed. Rio de Janeiro: McGraw-Hill, 2006

SOLDAMAQ. **Quem Somos**. Disponível em: <<https://soldamaq.com.br/quem-somos.html>>.

Acesso em: 11-03-2023.

TREINAWEB. **O que é Dart?**. Disponível em:

<<https://www.treinaweb.com.br/blog/o-que-e-dart>>. Acesso em: 23-07-2023.

ANEXO A - REQUISITOS DE NEGÓCIO

A.1 INTRODUÇÃO

Este documento tem como objetivo descrever o contexto, objetivos de negócio, funcionalidades e regras de negócio utilizadas no desenvolvimento do aplicativo gEstoque.

A.2 BACKGROUND E OPORTUNIDADES DE NEGÓCIO

A Soldamaq é uma empresa focada na venda de itens para construção, atuando em várias cidades de Mato Grosso do Sul, possuindo 12 lojas e comercializando mais de 37.000 (trinta e sete mil) itens.

A empresa utiliza o sistema ERP Sankhya para manter o controle e registro dos usuários e produtos. Porém, ainda se faz indispensável um sistema de gerenciamento de estoque próprio que atue em tempo real em todas as lojas, registrando e atualizando todas as saídas de produtos, e realizando operações de venda e entrega de itens.

O aplicativo gEstoque vem cumprir esse objetivo, conectando-se ao sistema principal da empresa, automatizando o serviço de vendas e gerenciamento dos produtos e trazendo recursos que forneçam a visão geral das vendas das lojas e funcionários.

A.3 OBJETIVOS DE NEGÓCIO

O desenvolvimento do sistema visa os seguintes objetivos de negócio:

ON-01 - Automatizar a verificação da disponibilidade de produtos no estoque.

ON-02 - Reduzir o tempo gasto com verificações no estoque.

ON-03 - Automatizar as vendas e pedidos de produtos.

ON-04 - Automatizar a organização das entregas de produtos.

ON-05 - Automatizar a verificação da disponibilidade de produtos no estoque.

ON-06 - Facilitar o contato entre vendedores e clientes na realização de orçamentos.

ON-07 - Aumentar a eficiência do registro do canhoto de boletos

ON-08 - Facilitar a pesquisa pelo canhoto dos boletos.

ON-09 - Reduzir o tempo gasto pelos funcionários no gerenciamento de saídas do estoque.

A.4 VISÃO DO PRODUTO

Para que os funcionários da empresa realizem a venda, entrega e gerenciamento de produtos no estoque das lojas, o aplicativo gEstoque é acessível no sistema operacional Android, permitindo o controle e verificação direta e instantânea da disponibilidade de produto em estoque, permitindo que as vendas e entregas sejam realizadas e controladas de forma rápida e eficiente pelo próprio celular.

A ausência desse aplicativo obrigaria a verificação manual do estoque em cada loja da empresa, com a verificação física das prateleiras do estoque, consumindo tempo e esforço excessivo dos funcionários, que pode ser melhor utilizado para atender o cliente.

A.5 PRINCIPAIS FUNCIONALIDADES

F1 - Cadastrar o cabeçalho(dados de pagamento) das vendas.

F2 - Pesquisar e adicionar produtos ao carrinho de compras.

F3 - Adicionar opcionalmente o valor de desconto às compras.

F4 - Visualizar os produtos em estoque separados por categoria.

F5 - Visualizar a disponibilidade de produtos no estoque por loja.

F6 - Exibir a foto e descrição dos produtos.

F7 - Editar a descrição dos produtos.

F8 - Mostrar a posição no ranking, as metas de venda.

F9 - Mostrar os indicadores de vendas e orçamentos realizados.

F10 - Pesquisar os canhotos de boletos de notas fiscais.

F11 - Registrar foto do canhoto dos boletos.

F12 - Iniciar, pausar e finalizar entregas.

A.6 LIMITAÇÕES

O sistema deve implementar as seguintes restrições:

- Produtos não podem ser adicionados ao carrinho de compras enquanto o cabeçalho não for confirmado.
- Somente o usuário Vendedor poderá acessar os módulos de Ranking e Indicadores
- Somente o usuário Entregador poderá acessar o módulo de Entregas.
- Somente o usuário Administrador poderá acessar o módulo de Canhoto de Boletos.
- Vendas só poderão ser efetuadas para parceiros cadastrados.

A.7 DIAGRAMA DE CONTEXTO

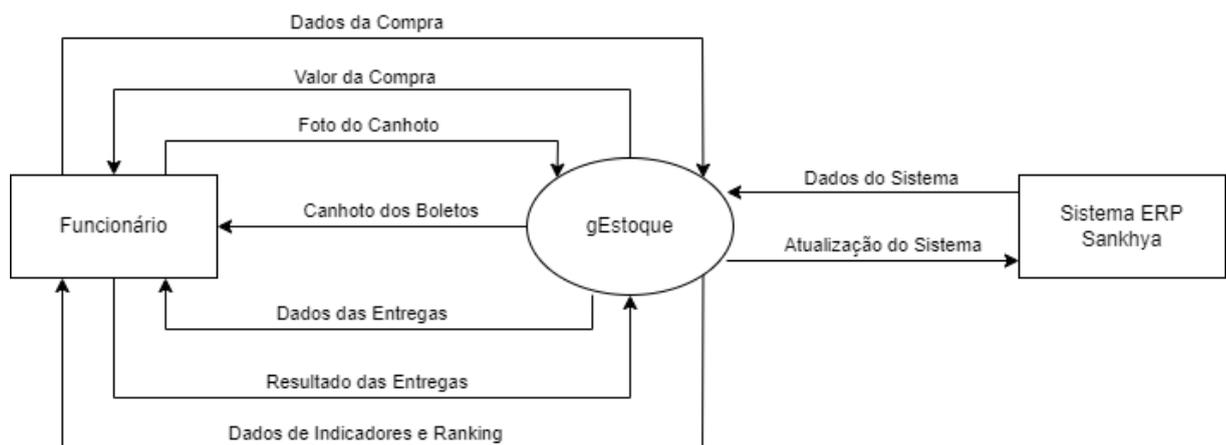


Figura 5 - Diagrama de Contexto

A.8 REGRAS DE NEGÓCIO

- RN-01** - Os produtos devem ser separados por categoria.
- RN-02** - Os funcionários devem poder consultar o estoque de qualquer loja da empresa.
- RN-03** - Cada vendedor deve possuir uma meta mensal de vendas.
- RN-04** - O entregador só poderá pausar entregas mediante justificativa.
- RN-05** - Cada venda deve gerar um número de nota fiscal.
- RN-06** - Devem ser armazenadas fotos dos canhotos de boletos.
- RN-07** - O vendedor deve poder consultar os indicadores de suas vendas e orçamentos.

ANEXO B - ESPECIFICAÇÃO DOS CASOS DE USO

B.1 DESCRIÇÃO DOS ATORES

Vendedor - Usuário primário do sistema. Possui permissão para visualizar os rankings e metas, realizar orçamentos e executar a venda dos produtos aos clientes.

Entregador - Usuário primário do sistema. Realiza e registra a entrega dos produtos no endereço dos clientes e possui permissão para a gerar o canhoto da entrega.

Administrador - Possui permissões de administrador do sistema, pode acessar o canhoto dos boletos e registrar a foto do canhoto.

B.2 CASOS DE USO

B.2.1 Caso de Uso: Consultar Produto

- **Descrição:**

O usuário acessa a aplicação e realiza a consulta por um produto no estoque.

- **Atores:**

Qualquer usuário.

- **Pré-condições:**

O usuário está autenticado.

- **Pós-condições:**

O produto foi consultado no sistema.

- **Fluxo Principal:**

1. O usuário acessa a opção “Produtos” na tela inicial (*dashboard*).
2. O sistema exibe a tela de seleção de categorias.
3. O usuário seleciona a categoria desejada.
4. O usuário pesquisa o produto por nome ou código.
5. O sistema exibe a lista com os produtos cadastrados correspondentes às informações inseridas.

- **Fluxo Alternativo:**

- **Fluxo alternativo 1**

- No passo 5, caso não exista um produto correspondente à descrição, o sistema informa que o produto não foi encontrado e retorna ao passo 4.

B.2.2 Caso de Uso: Visualizar Descrição do Produto

- **Descrição:**

- O usuário visualiza a foto e descrição de um produto em estoque.

- **Atores:**

- Qualquer usuário.

- **Pré-condições:**

- O produto a ser visualizado está registrado e foi pesquisado pelo usuário.

- **Pós-condições:**

- A foto e descrição do produto foi visualizada

- **Fluxo Principal:**

- 1. O usuário acessa a lista dos produtos (<< includes **Consultar Produto** >>)
 2. O usuário seleciona a opção “Visualizar”
 3. O sistema exibe a foto e descrição do produto.

- **Fluxo Alternativo:**

-

B.2.3 Caso de Uso: Visualizar Estoque do Produto

- **Descrição:**

- O usuário visualiza a disponibilidade em estoque de um produto.

- **Atores:**

Qualquer usuário.

- **Pré-condições:**

O produto a ser visualizado está registrado e foi pesquisado pelo usuário..

- **Pós-condições:**

A quantidade em estoque do produto foi exibida.

- **Fluxo Principal:**

1. O usuário acessa a lista dos produtos (<< includes **Consultar Produto** >>)
2. O usuário seleciona a opção “Estoque”
3. O sistema exibe quantidade em estoque do produto por loja.

- **Fluxo Alternativo:**

-

B.2.4 Caso de Uso: Fazer Login

- **Descrição:**

O usuário acessa o sistema.

- **Atores:**

Qualquer usuário.

- **Pré-condições:**

O usuário está cadastrado no sistema.

- **Pós-condições:**

O usuário está autenticado e pode utilizar as funcionalidades do sistema.

- **Fluxo Principal:**

1. O usuário clica em “Acessar o Sistema” na tela inicial do aplicativo.

2. O sistema exibe a tela de login, com os campos para inserir o nome de usuário e senha.
3. O usuário preenche sua conta e senha e clica em “Login”.
4. O sistema valida os dados informados.
5. O sistema conecta o usuário a sua conta.
6. O sistema apresenta a tela inicial (*dashboard*) de acordo com a permissão de acesso do usuário.

- **Fluxo Alternativo:**

- **Fluxo alternativo 1:**

- No passo 3, caso o usuário não informe as informações de login e senha, o sistema exibe uma mensagem de erro e retorna ao passo 2.

- **Fluxo alternativo 2:**

- No passo 4, caso as informações estiverem inseridas incorretas, o sistema exibe uma mensagem de erro e retorna ao passo 2.

B.2.5 Caso de Uso: Consultar Ranking

- **Descrição:**

- O usuário consulta o seu ranking de vendas

- **Atores:**

- Vendedor.

- **Pré-condições:**

- O usuário está autenticado.

- **Pós-condições:**

- O usuário visualiza suas informações de vendas.

- **Fluxo Principal:**

1. O usuário seleciona a opção “Ranking” na tela inicial (*dashboard*).

2. O sistema apresenta a tela com o ranking e os dados de vendas do usuário.

- **Fluxo Alternativo:**

-

B.2.6 Caso de Uso: Consultar Indicadores

- **Descrição:**

O usuário consulta os indicadores de suas vendas e orçamentos.

- **Atores:**

Vendedor.

- **Pré-condições:**

O usuário está autenticado.

- **Pós-condições:**

O usuário visualiza os indicadores.

- **Fluxo Principal:**

1. O usuário seleciona a opção “Indicadores” na tela inicial (*dashboard*).
2. O sistema apresenta a tela com suas vendas das últimas 24 horas..
3. O usuário visualiza suas vendas.

- **Fluxo Alternativo:**

- **Fluxo alternativo 1:**

No passo 3, se o usuário quiser consultar seus orçamentos, seleciona o campo de orçamentos, e o sistema exibe os orçamentos realizados nos últimos 3 dias.

B.2.7 Caso de Uso: Realizar Entregas

- **Descrição:**

O entregador realiza a entrega dos produtos ao parceiro.

- **Atores:**

Entregador.

- **Pré-condições:**

O usuário está autenticado, a entrega está agendada e confirmada com o parceiro..

- **Pós-condições:**

A entrega foi realizada.

- **Fluxo Principal:**

1. O usuário seleciona a opção “Entregas” na tela inicial (*dashboard*).
2. O sistema exibe a lista de entregas pendentes.
3. O usuário clica em “Selecionar Entregas” e seleciona as entregas que irá realizar.
4. O usuário adiciona as entregas selecionadas as suas entregas.
5. O usuário seleciona o campo “Minhas Entregas” e seleciona a entrega que irá realizar.
6. O usuário clica em “Iniciar Rota”.
7. O usuário se desloca até o endereço do parceiro com o produto.
8. O usuário realiza a entrega dos produtos.
9. O usuário clica em “Finalizar Rota”.

- **Fluxo Alternativo:**

- **Fluxo alternativo 1:**

No passo 7, caso o usuário deseje pausar a entrega, clica na opção “Pausar Entrega”, seleciona uma das opções de justificativa e o sistema interrompe a entrega, retornando ao passo 5.

B.2.8 Caso de Uso: Cadastrar Cabeçalho

- **Descrição:**

O usuário cadastra o cabeçalho da compra.
- **Atores:**

Qualquer usuário.
- **Pré-condições:**

O usuário está autenticado.
- **Pós-condições:**

O cabeçalho está confirmado.
- **Fluxo Principal:**
 1. O usuário seleciona a opção “Central de Vendas” na tela inicial (*dashboard*).
 2. O usuário seleciona o tipo de operação (Pedido ou Venda).
 3. O usuário insere o nome do parceiro.
 4. O sistema apresenta a lista de nomes cadastrados correspondentes ao inserido.
 5. O usuário seleciona o parceiro atendido.
 6. O usuário insere o tipo de negociação.
 7. O sistema apresenta a lista com os tipos de negociação cadastrados.
 8. O usuário seleciona o tipo de negociação conforme a preferência do parceiro e opcionalmente adiciona observações.
 9. O usuário salva o cabeçalho.
 10. O sistema confirma que o cabeçalho está salvo.
- **Fluxo Alternativo:**

-

B.2.9 Caso de Uso: Adicionar Produto ao Carrinho

- **Descrição:**

O usuário adiciona produtos ao carrinho de compras.
- **Atores:**

Qualquer usuário.
- **Pré-condições:**

O cabeçalho está preenchido e salvo.
- **Pós-condições:**

Os produtos foram adicionados ao carrinho de compras.
- **Fluxo Principal:**
 1. O usuário cadastra o cabeçalho. (<< includes **Cadastrar Cabeçalho** >>)
 2. O usuário seleciona a opção “Pesquisar” na Central de Vendas.
 3. O usuário pesquisa o produto por código ou nome.
 4. O sistema apresenta o produto correspondente, incluindo o valor e os detalhes do produto.
 5. O usuário clica na opção de adicionar o item.
 6. O sistema exibe a tela para informar a quantidade e valor do desconto.
 7. O usuário insere a quantidade a ser comprada e opcionalmente a porcentagem do desconto.
 8. O sistema calcula e exibe o preço a ser pago.
 9. O usuário confirma e o produto é adicionado ao carrinho de compras
- **Fluxo Alternativo:**
 - **Fluxo alternativo 1:**

No passo 9, caso o cliente desista da compra do produto, o usuário cancela a operação e retorna ao passo 3.

B.2.10 Caso de Uso: Confirmar Transação

- **Descrição:**

O usuário confirma a operação de venda ou pedido.
- **Atores:**

Qualquer usuário.
- **Pré-condições:**

O cabeçalho está preenchido e salvo, os produtos foram adicionados ao carrinho de compras.
- **Pós-condições:**

Os produtos foram adicionados ao carrinho de compras.
- **Fluxo Principal:**
 1. O usuário adiciona os itens ao carrinho de compras. (<< includes **Adicionar Produto ao Carrinho** >>)
 2. O usuário acessa o carrinho de compras.
 3. O sistema exibe a lista de todos os produtos adicionados, informando a quantidade, valor do desconto e total a pagar.
 4. O usuário acessa a tela do cabeçalho e seleciona a opção “Confirmar”.
 5. O sistema gera o número da nota e solicita a confirmação do usuário.
 6. O usuário confirma o pedido, que é registrado no sistema.
- **Fluxo Alternativo:**
 - **Fluxo alternativo 1:**

No passo 4, caso o parceiro desista da compra, o usuário remove os itens do carrinho de compras e cancela a operação.

- **Fluxo alternativo 2:**

No passo 5, caso o parceiro esteja em atraso, o sistema emite também o valor e data do atraso, solicitando a confirmação do usuário

B.2.11 Caso de Uso: Pesquisar Boletos

- **Descrição:**

O usuário pesquisa pelo canhoto dos boletos.

- **Atores:**

Administrador.

- **Pré-condições:**

O usuário está autenticado.

- **Pós-condições:**

Os boletos foram listados

- **Fluxo Principal:**

1. O usuário seleciona a opção “Canhoto Boletos”.
2. O sistema exibe um formulário solicitando número da nota, data de início e data final para pesquisa.
3. O usuário insere os dados a serem pesquisados.
4. O sistema apresenta os boletos correspondentes a pesquisa, listados por data.

- **Fluxo Alternativo:**

- **Fluxo alternativo 1:**

No passo 4, caso não haja boletos correspondentes, o sistema informa o usuário e retorna ao passo 3.

B.2.12 Caso de Uso: Registrar Foto do Canhoto

- **Descrição:**

O usuário adiciona uma foto do canhoto do boleto.
- **Atores:**

Administrador.
- **Pré-condições:**

O boleto foi pesquisado pelo usuário.
- **Pós-condições:**

A foto do canhoto foi enviada.
- **Fluxo Principal:**
 1. O usuário acessa a lista dos boletos. (<< includes **Pesquisar Boletos**>>)
 2. O usuário seleciona a opção “Foto do Canhoto”.
 3. O sistema apresenta o formulário para inserção dos dados (nome do recebedor, observação, status)
 4. O usuário seleciona a opção “Tirar Foto do Canhoto”..
 5. O usuário tira uma foto do canhoto.
 6. O usuário confirma a foto.
 7. O usuário envia a foto.
- **Fluxo Alternativo:**
 - **Fluxo alternativo 1:**

No passo 6, caso a foto não esteja aceitável, o usuário cancela e retorna ao passo 5.

ANEXO C - ESPECIFICAÇÃO DOS REQUISITOS DE SOFTWARE

C.1 INTRODUÇÃO

Este documento mantém o registro detalhado dos requisitos para o aplicativo *gEstoque*, na forma de requisitos textuais.

C.2 REQUISITOS DE SOFTWARE

C.2.1 Requisitos Funcionais

C.2.1.1 Controle de Acesso

RF1 - O sistema deve consultar a API e validar os dados de login.

RF2 - O sistema não deve permitir que usuários sem a devida permissão realizem operações não autorizadas pelo nível de acesso.

RF3 - O sistema deve permitir que todos os usuários possam realizar o Logoff.

C.2.1.2 Dashboard

RF4 - Ao abrir o aplicativo, o sistema deve exibir a opção de acessar o sistema, ou contatar o suporte técnico por WhatsApp, e-mail ou telefone.

RF5 - Ao clicar em acessar o sistema, o sistema deve exibir a tela de login, permitindo a inserção do usuário e senha, o preenchimento da opção de lembrar o usuário, a opção de visualizar a senha inserida, e o botão de confirmação de login.

RF6 - O sistema deve conectar automaticamente o usuário a devida tela de *Dashboard* (página inicial) após a confirmação de login, conforme a permissão de acesso.

RF7 - A tela de *Dashboard* deve mostrar a foto e login do usuário, seguidos pela lista de ações do usuário.

RF8 - A lista de ações do *Dashboard* para o usuário “Vendedor” deve conter os parâmetros: ranking, indicadores, produtos, central de vendas, sair.

RF9 - A lista de ações do *Dashboard* para o usuário “Entregador” deve conter os parâmetros: entregas, produtos, central de vendas e sair.

RF10 - A lista de ações do *Dashboard* para o usuário “Administrador” deve conter os parâmetros: produtos, canhoto de boletos, central de vendas e sair.

C.2.1.3 Ações

RF11 - O sistema deve permitir que qualquer usuário registrado possa realizar ações de acordo com o nível de acesso.

RF12 - O sistema deve permitir que o usuário possa a qualquer momento retornar à tela inicial.

C.2.1.4 Gestão do Estoque

RF13 - Ao acessar o módulo Canhoto Boletos, o sistema deve permitir a pesquisa de boletos por número da nota fiscal, data de início e data final.

RF14 - O sistema deve exibir todos os canhotos conforme os dados da pesquisa listados por data e contendo: número da nota, chave nf-e, código e nome do parceiro, valor da nota e data de negociação.

RF15 - O sistema deve permitir que o usuário tire uma foto do canhoto e insira os dados: nome do recebedor, observação e status total ou parcial.

RF16 - Ao acessar o módulo Produtos, o sistema deve exibir as categorias: construção civil, elétrica e hidráulica, ferramentas elétricas, indústria, fixação, oficina, agronegócio, utilidades, epis e segurança e limpeza.

RF17 - Ao acessar uma categoria de produto, o sistema deve permitir a pesquisa de produtos por nome ou código.

RF18 - O sistema deve permitir a pesquisa do produto por leitura do código de barras.

RF19 - O sistema deve permitir a visualização da foto e descrição do produto, quantidade em estoque por loja, volume e edição da descrição do produto.

RF20 - Quando o usuário clicar na visualização de estoque por loja, o sistema deve exibir o nome da loja, estoque e quantidade reservada do produto.

RF21 - A Central de Vendas, deve conter 3 telas: preenchimento dos dados de pagamento, pesquisa dos produtos e o carrinho de compras.

RF22 - Ao clicar em Central de Vendas, o sistema deve sempre exibir a tela de preenchimento dos dados de pagamento, exibindo sempre as opções de acessar as outras telas a qualquer momento por ícones na parte inferior da tela.

RF23 - A tela de preenchimento dos dados de pagamento deve conter um formulário com os seguintes parâmetros: a especificação de operação entre pedido ou venda, o tipo de operação e seu código, o parceiro e seu código, nome, tipo de negociação e seu código e observações e as opções de salvar ou excluir o cabeçalho.

RF24 - O sistema não deve permitir a adição de itens ao carrinho de compras enquanto o cabeçalho não estiver salvo.

RF25 - A tela de preenchimento dos dados de pagamento deve exibir a opção de confirmar a compra.

RF26 - Ao preencher total ou parcialmente um dos campos “Tipo de Operação”, “Parceiro” ou “Tipo de Negociação”, o sistema deve exibir uma lista com as opções cadastradas para cada campo.

RF27 - Ao acessar a tela de pesquisa dos produtos, o sistema deve permitir a pesquisa do produto pelo código, nome ou leitura do código de barras.

RF28 - Ao pesquisar um produto, o sistema deve permitir a visualização da foto e descrição do produto, estoque por loja, volume e adição ao carrinho de compras.

RF29 - Quando o usuário clicar em adicionar o item ao carrinho de compras, o sistema deve exibir o valor total da compra, a quantidade selecionada, valor total do desconto, o campo de adicionar o produto. um formulário com os parâmetros de quantidade de itens e porcentagem de desconto, a opção de zerar todos os valores, e a visualização da foto e descrição do produto e estoque por loja.

RF30 - O ícone da tela do carrinho de compras deve mostrar sempre a quantidade adicionada ao carrinho, e o ícone da tela dos dados de pagamento deve exibir sempre o valor da compra.

RF31 - A tela do carrinho de compras deve exibir os itens adicionados, indicando o código, nome, volume, preço sem desconto, valor do desconto, quantidade adicionada, total a pagar e a opção de remover o produto.

RF32 - Quando o usuário clicar em confirmar a operação na tela de preenchimento dos dados de pagamento, o sistema deve exibir o número da nota fiscal, e solicitar a confirmação.

RF33 - Ao acessar o módulo Ranking, o sistema deve exibir: um campo com o ranking do funcionário, nome, meta e projeção mensal, um campo com melhor resultado, vendas do mês, vendas do dia e vendas da semana, e um campo com um gráfico de porcentagem da meta alcançada.

RF34 - O módulo Indicadores deve estar dividido entre as telas Venda e Orçamentos.

RF35 - Ao acessar o módulo Indicadores, o sistema deve sempre exibir a tela Venda permitindo transitar entre as telas de Venda e Orçamento

RF36 - A tela Venda deve mostrar todas as vendas do usuário nas últimas 24 horas, exibindo o código e nome do parceiro, número, data e valor da venda

RF37 - A tela Orçamento deve mostrar todos os orçamentos realizados pelo usuário nos últimos 3 dias, exibindo o código e nome do parceiro, número, data, valor do orçamento e o contato.

RF38 - O módulo Entregas deve estar dividido entre as telas: Entregas, e Minhas Entregas, exibindo ao ser clicado a tela Entregas e permitindo transitar entre as telas a qualquer momento.

RF39 - A tela de Entregas deve listar as entregas pendentes da loja, exibindo o número único, nome do parceiro, forma de pagamento, telefone, e endereço.

RF40 - O sistema deve permitir que o usuário selecione entregas pendentes no campo Entregas e adicione ao campo Minhas Entregas.

RF41 - A tela Minhas Entregas deve exibir as entregas pendentes do usuário, a opção de atualizar as entregas e um campo com as opções de iniciar, pausar e finalizar as rotas das entregas.

RF42 - Ao clicar em “Pausar Entrega”, o sistema deve exibir uma lista com as opções de justificativas válidas: acidente, desvio de rota, carro ou moto quebrou, endereço não encontrado, cliente não estava no local de entrega, forma de pagamento não autorizada.

RF43 - O sistema só deve pausar uma entrega se uma justificativa válida for selecionada.

ANEXO D - TESTES DAS FUNCIONALIDADES

D.1 AMBIENTE DE TESTES

Os testes do aplicativo “gEstoque” foram realizados no emulador do Android Studio, utilizando um dispositivo Google Pixel 3a, Android 13 (*API* nível 33).

D.2 TESTES

Os testes foram criados utilizando como base os requisitos da aplicação informados pelo cliente, sendo assim cada teste valida um requisito funcional ou não funcional, dessa forma conseguimos garantir que a aplicação está funcionando da maneira esperada.

Todas as funcionalidades da aplicação foram testadas de maneira a emular erros dos usuários e constatou-se que ao tentar realizar uma ação indevida, por exemplo, inserção de caracteres não condizentes, o sistema identifica o erro e informa ao usuário que não é possível prosseguir da forma informada.

D.2.1 Teste: Tela de Contatos

- **Requisitos verificados:**
RF4.
- **Descrição:**
Teste da funcionalidade dos botões da tela inicial.
- **Verificações**
 - O botão com o logo do "WhatsApp" realiza o envio de uma mensagem ao WhatsApp do suporte técnico.
 - O botão com o símbolo da "caixa postal" realiza o envio de uma mensagem eletrônica ao e-mail do suporte.
 - O botão com o símbolo do "telefone" realiza uma ligação ao número do suporte.
 - O botão “ACESSAR O SISTEMA” realiza o redirecionamento à tela de login.

- **Resultado:**

Todas as funções estão operando corretamente.



Figura 6 - Tela de Contatos

D.2.2 Teste: Tela de Login

- **Requisitos verificados:**

RF1 e RF5.

- **Descrição:**

Teste da funcionalidade de login.

- **Verificações**

- O botão "Lembrar-me" armazena os dados de login do usuário para não ser necessário reinserir os dados a cada login.
- Os campos de input de texto "Usuário" e "Senha" realizam a função de inserção das informações para login.
- O botão de mostrar/ocultar senha ao ser acionado mostra ou oculta a senha.

- O botão “LOGIN” realiza uma consulta à API e verifica se os dados estão corretos, caso estejam corretos redireciona ao menu principal do app, do contrário exibe uma pop-up informando o erro.

- **Resultado:**

Todas as funções estão operando corretamente.

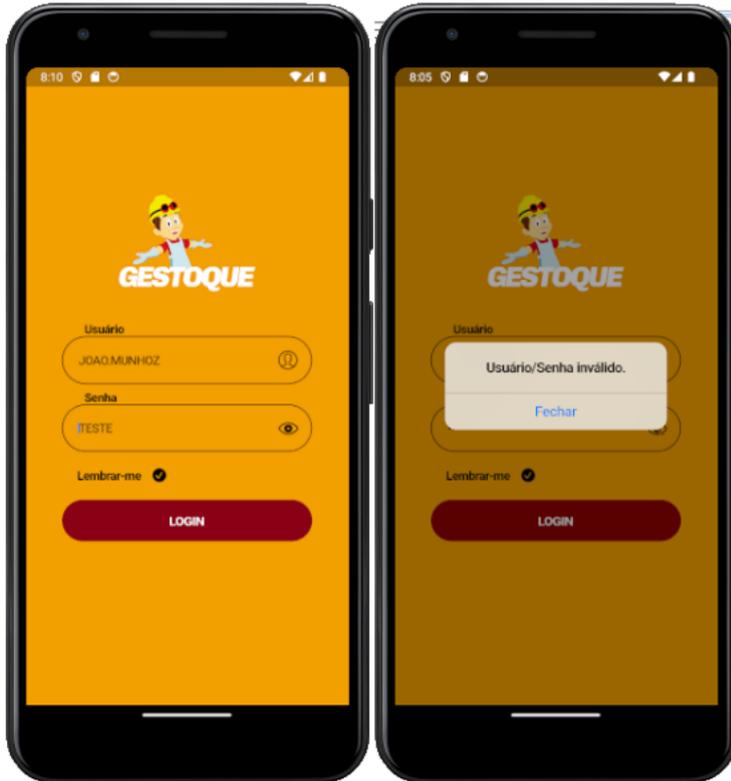


Figura 7 - Tela de Login

D.2.3 Teste: Tela do Menu Principal

- **Requisitos verificados:**
RF6, RF7, RF8, RF9 e RF10.
- **Descrição:**
Teste da funcionalidade do menu principal.
- **Verificações**

- Qualquer usuário: acessa os módulos “Produtos” e “Central de Vendas” pois eles não possuem restrição de acesso.
- Usuário Administrador: acessa o módulo “Canhoto de Boletos”, o qual é exclusivo para este tipo de usuário.
- Usuário Entregador: acessa o módulo “Entregas”, o qual é exclusivo para este tipo de usuário.
- Usuário Vendedor: acessa os módulos “Ranking” e “Indicadores”, o qual é exclusivo para este tipo de usuário.
- O redirecionamento dos módulos está funcionando corretamente.

- **Resultado:**

Todas as funções estão operando corretamente.

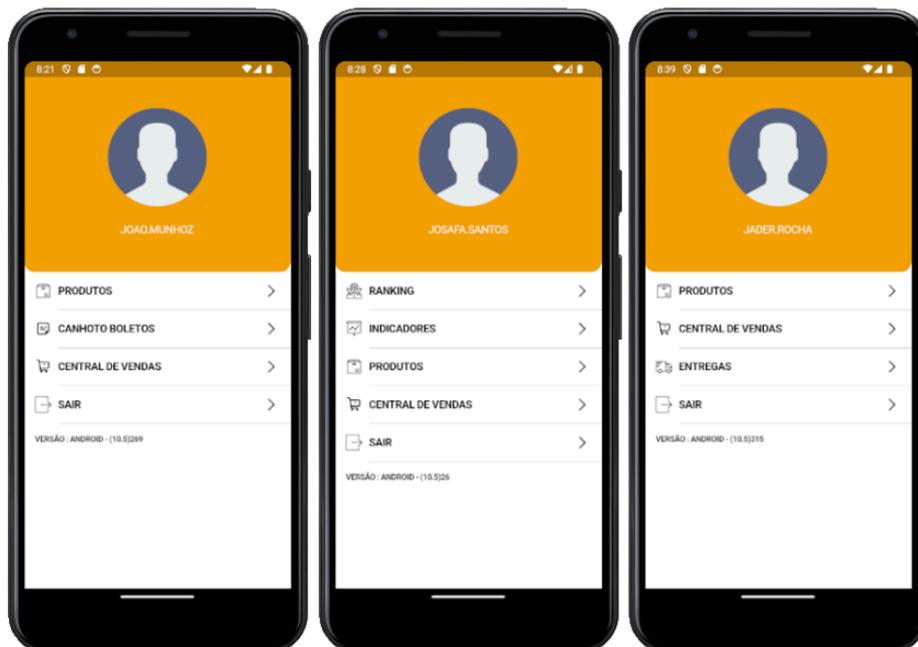


Figura 8 - Tela do Menu Principal

D.2.4 Teste: Módulo Produtos

- **Requisitos verificados:**
RF16, RF17, RF18, RF19 e RF20.

- **Descrição:**

Teste das funcionalidades do módulo Produtos.

- **Verificações**

- Ao acessar a tela de Produtos, o sistema exibe as categorias: construção civil, elétrica e hidráulica, ferramentas elétricas, indústria, fixação, oficina, agronegócio, utilidades, epis e segurança, e limpeza.
- Acessando uma das categorias de produto, o sistema oferece uma funcionalidade que permite aos usuários explorar a categoria escolhida, realizando buscas detalhadas por nome ou código do produto.
- O sistema permite a pesquisa do produto por leitura do código de barras. Os usuários podem escanear o código de barras de um produto, agilizando o acesso às informações desejadas.
- A tela do produto oferece uma visão abrangente, incluindo a foto do item, uma descrição detalhada, a quantidade disponível em estoque por loja, o volume do produto e a opção de editar a descrição para manter as informações sempre atualizadas.
- Ao clicar na visualização do estoque por loja, os usuários podem verificar informações específicas, como o nome da loja, a quantidade total em estoque e a quantidade reservada do produto.

- **Resultado:**

Todas as funções estão operando corretamente.

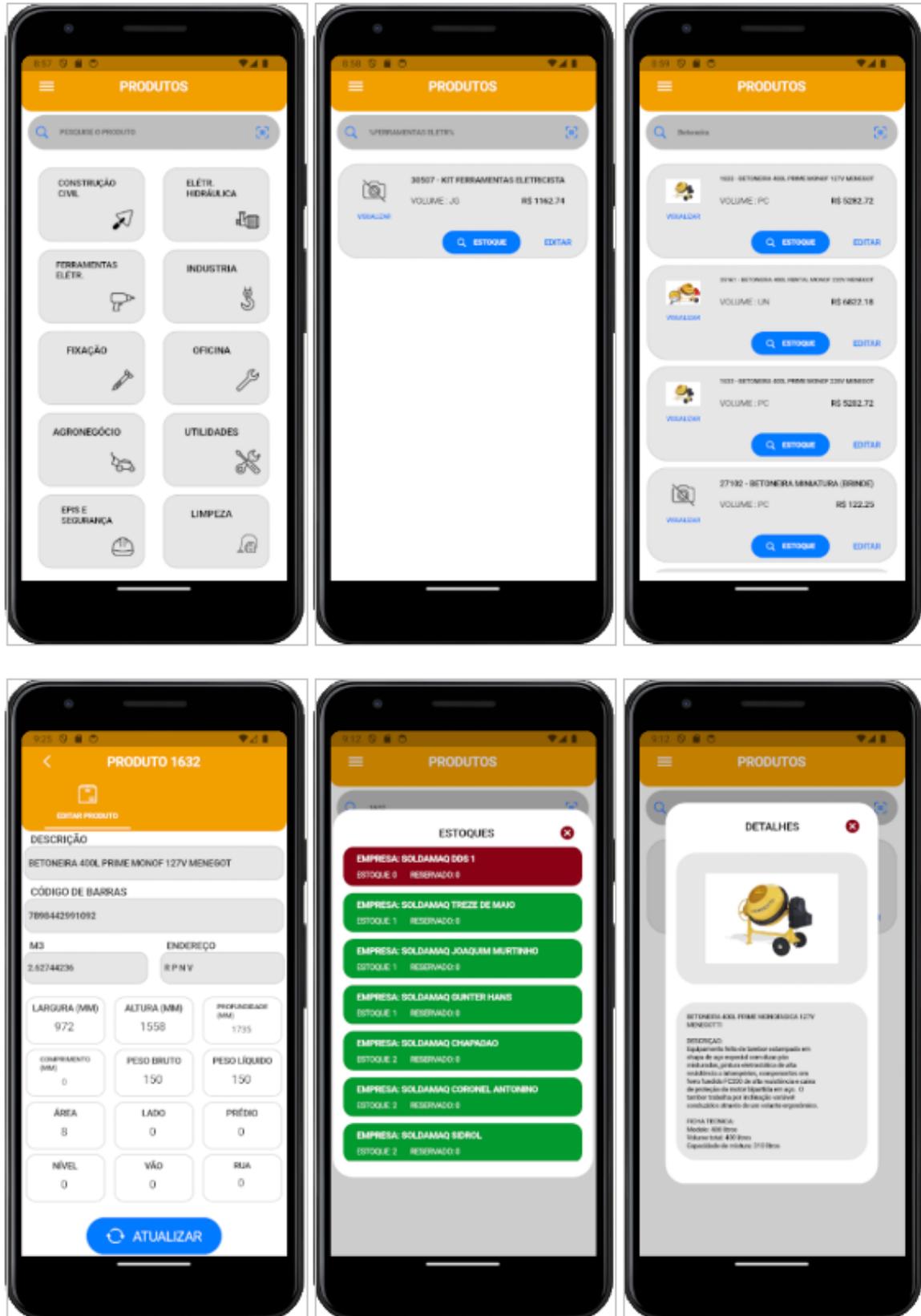


Figura 9 - Módulo Produtos

D.2.5 Teste: Módulo Central de Vendas

- **Requisitos verificados:**

RF21, RF22, RF23, RF24, RF25, RF26, RF27, RF28, RF29, RF30, RF31 e RF32.

- **Descrição:**

Teste das funcionalidades do módulo Central de Vendas.

- **Verificações**

- O módulo Central de Vendas, contém 3 telas: preenchimento dos dados de pagamento, pesquisa dos produtos e o carrinho de compras.
- Ao clicar em Central de Vendas, o sistema exibe a tela de preenchimento do cabeçalho.
- A tela do cabeçalho contém em seu formulário: botão de seleção entre operações de pedido ou venda, campos de inserção de texto para o tipo de operação e seu código, o nome do parceiro e seu código, apelido, tipo de negociação e seu código e observações e botões com as opção de confirmar, salvar ou excluir o cabeçalho.
- O sistema não permite a adição de itens ao carrinho de compras enquanto o cabeçalho não estiver salvo.
- A tela de cabeçalho exibe uma pop-up de confirmação da compra.
- O sistema permite realizar uma busca ao digitar o nome do parceiro, total ou parcialmente.
- Ao acessar a tela de pesquisa dos produtos dentro do módulo “Central de Vendas”, o sistema permite a pesquisa do produto pelo código, nome ou leitura do código de barras.
- Ao pesquisar um produto no módulo “Central de Vendas”, o sistema permite a visualização da foto através do botão “Visualizar”, visualização da descrição do produto, estoque por loja através do botão “Estoque”, exibição do volume e adição ao carrinho de compras através da expansão do card ao clicar no botão “+ ITEM”.

- Ao clicar no botão “+ ITEM” na tela de busca de produtos no módulo “Central de Vendas”. Para exibição dos dados de adição, o sistema exibe o valor total da compra, a quantidade selecionada, valor total do desconto, o campo de adicionar o produto. um formulário com os parâmetros de quantidade de itens e porcentagem de desconto, a opção de zerar todos os valores, e a visualização da foto e descrição do produto e estoque por loja.
 - O ícone do carrinho de compras está sendo mostrado a todo momento e sendo atualizado com o valor correto após cada adição de produto, assim como o ícone da carteira.
 - Acessando a tela do carrinho de compras no módulo “Central de Vendas” é possível consultar os itens adicionados, indicando o código, nome, volume, preço sem desconto, valor do desconto, quantidade adicionada, total a pagar e a opção de remover o produto.
 - Ao confirmar a compra, o sistema está fazendo uma validação para caso o cliente esteja em atraso, sendo assim é mostrado uma pop-up com o valor total atrasado. Ao prosseguir é informado o número da nota fiscal gerada, a data do atraso e o valor total e solicitar a confirmação da compra.
- **Resultado:**

Todas as funções estão operando corretamente.

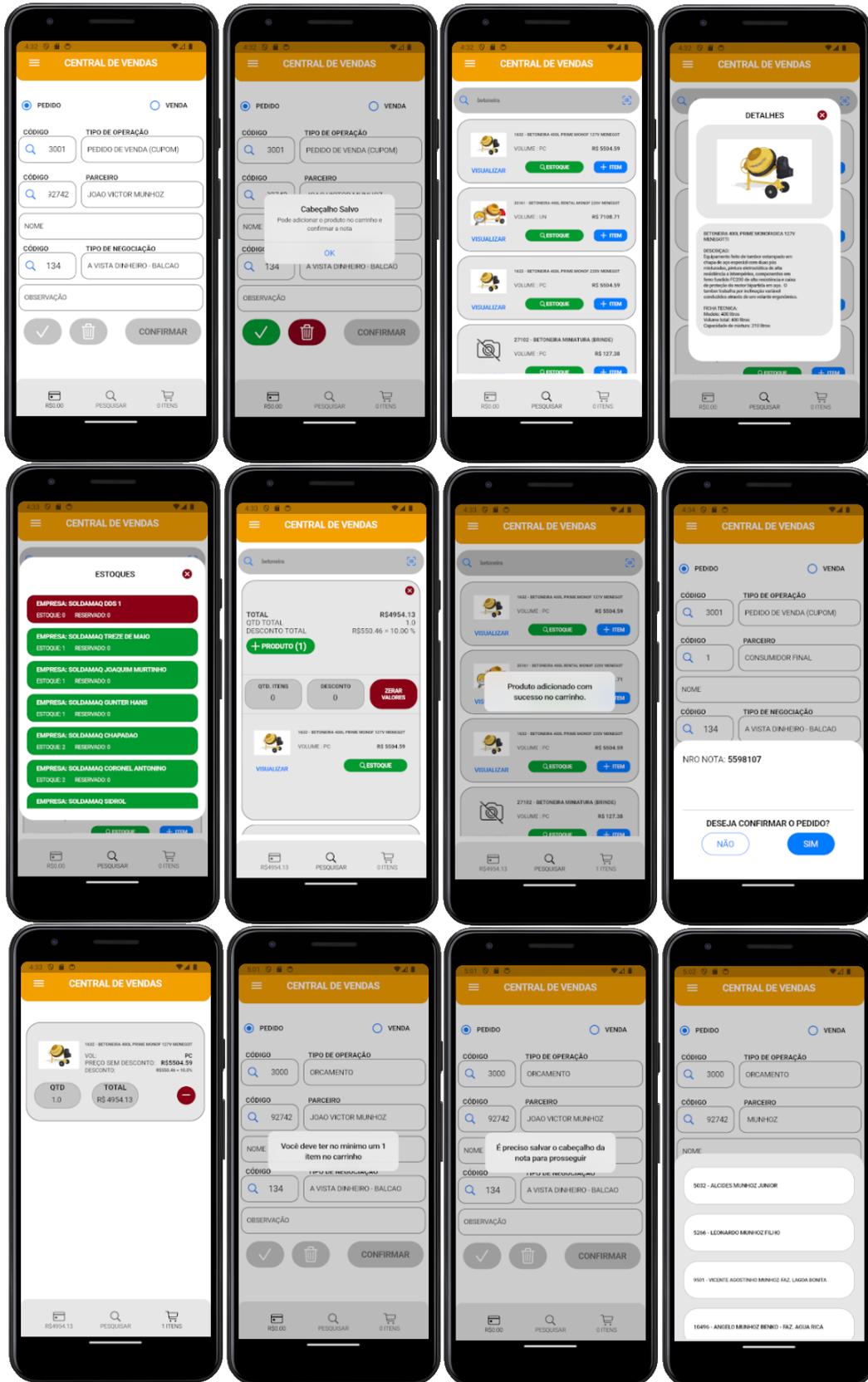


Figura 10 - Módulo Central de Vendas

D.2.6 Teste: Módulo Canhoto Boletos

- **Requisitos verificados:**

RF13, RF14 e RF15.

- **Descrição:**

Teste das funcionalidades do módulo Canhoto Boletos.

- **Verificações**

- Ao acessar a tela Canhoto Boletos, o sistema realiza a busca de boletos por número da nota fiscal, data de início e data final.
- O sistema exibe todos os canhotos conforme os dados da pesquisa listados por data e contendo: número da nota, chave nf-e, código e nome do parceiro, valor da nota e data de negociação.
- O sistema exibe um botão para registro da nota, sendo ele uma pop-up com campos para preenchimento dos dados: nome do recebedor, observação e status total ou parcial, e um botão para foto do canhoto do boleto.

- **Resultado:**

Todas as funções estão operando corretamente.

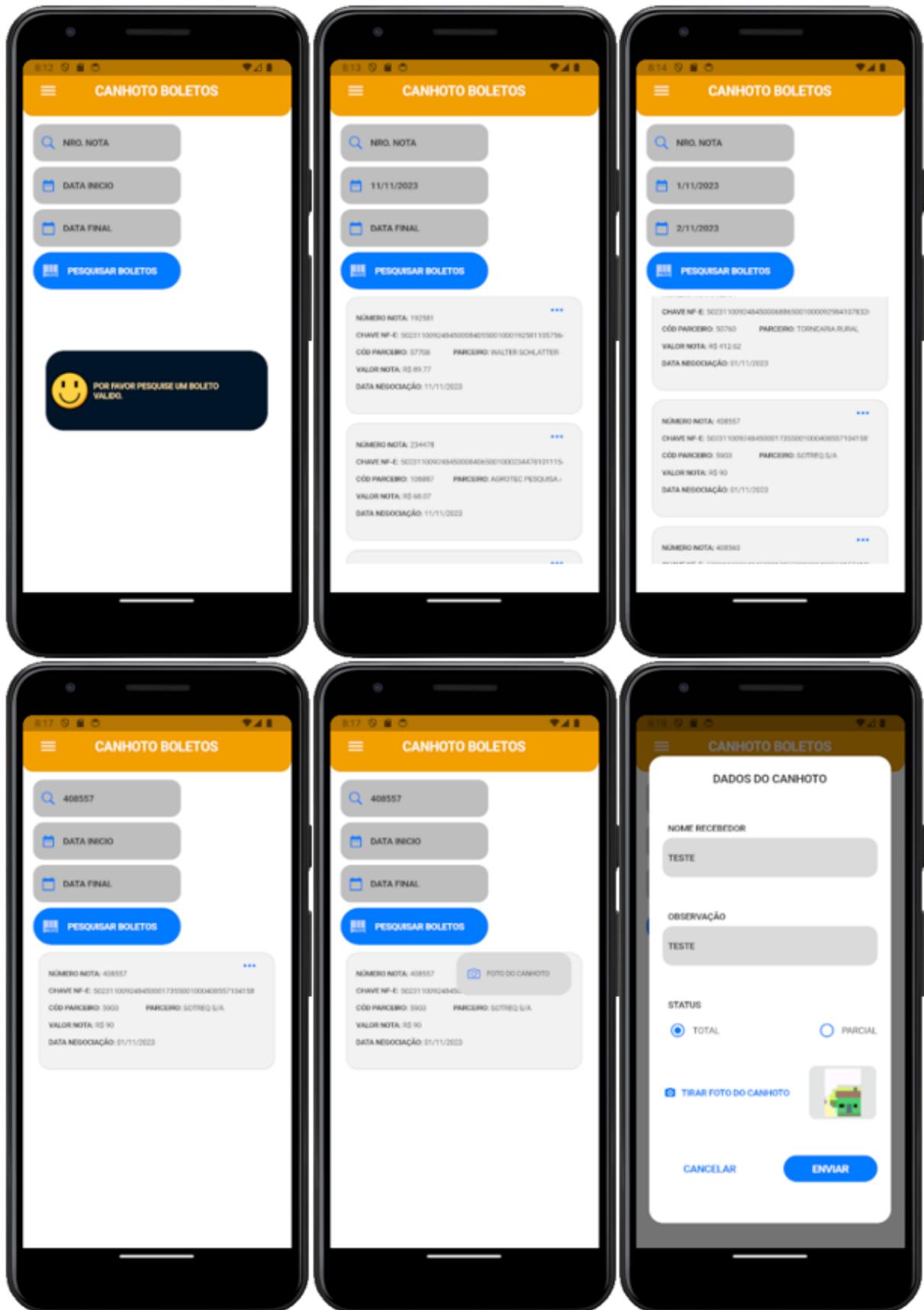


Figura 11 - Módulo Canhoto Boletos

D.2.7 Teste: Módulo Entregas

- **Requisitos verificados:**

RF38, RF39, RF40, RF41, RF42 e RF43.

- **Descrição:**

Teste das funcionalidades do módulo Entregas

- **Verificações**

- O módulo de Entregas apresenta uma divisão entre as visualizações: Entregas, e Minhas Entregas.
- A exibição de “Entregas” apresenta uma lista para as entregas pendentes da loja, exibindo o número único, nome do parceiro, forma de pagamento, telefone, e endereço.
- O sistema permite seleção das entregas pendentes na visualização “Entregas” e adicione à “Minhas Entregas”.
- Ao acessar a visualização “Minhas Entregas” é possível visualizar as entregas pendentes do usuário, a opção de atualizar as entregas e as opções de iniciar ou finalizar as rotas das entregas pendentes ou iniciadas, e também iniciar, pausar e finalizar uma entrega específica.
- Selecionar a opção de pausar uma entrega exhibe uma pop-up com o motivo da pausa.

- **Resultado:**

Todas as funções estão operando corretamente.

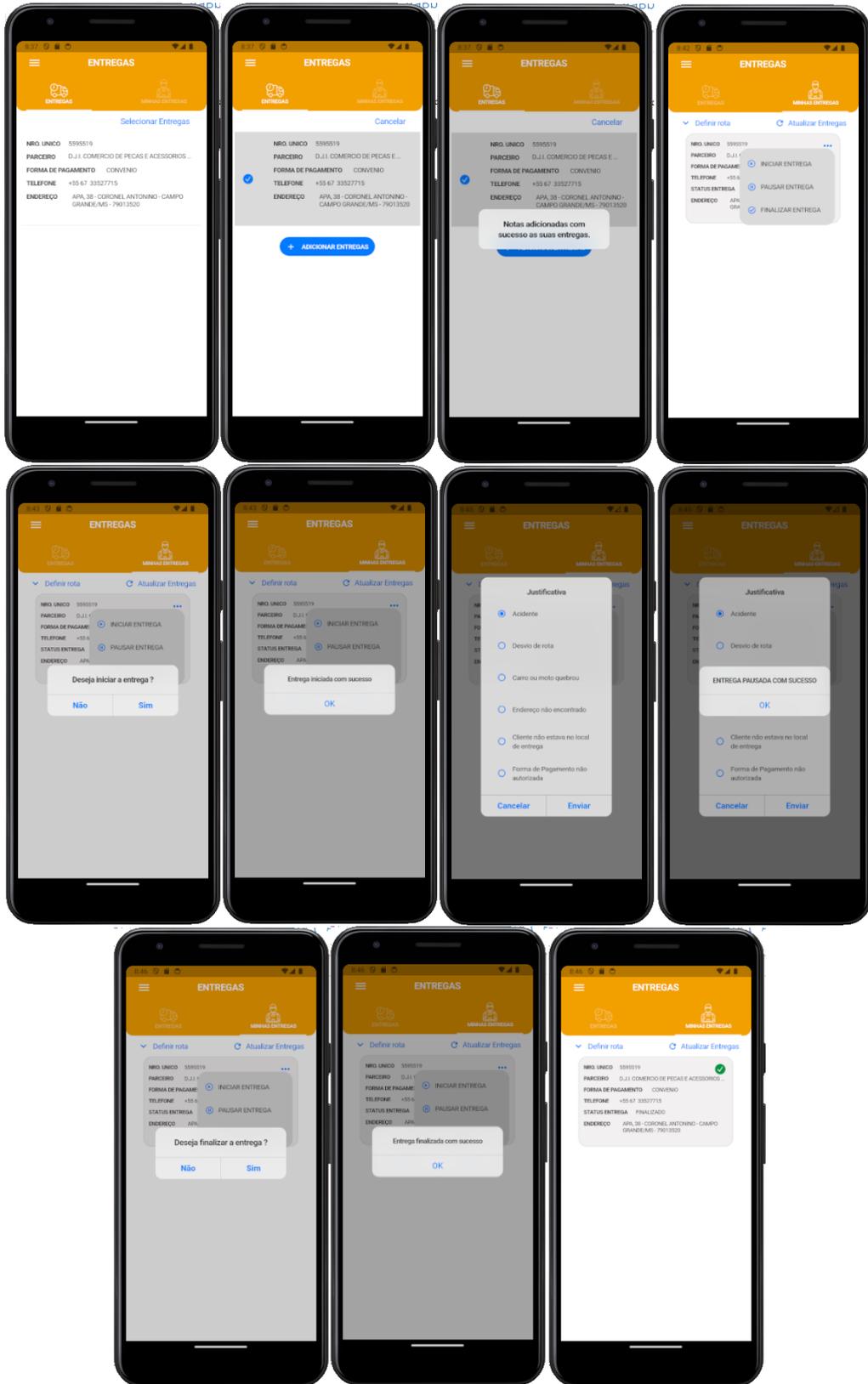


Figura 12 - Módulo Entregas

D.2.8 Teste: Módulo Ranking

- **Requisitos verificados:**
RF33.
- **Descrição:**
Teste da funcionalidade dos botões.
- **Verificações**
 - Ao acessar a tela de Ranking, o sistema exibe um campo com: o ranking do funcionário, nome, meta e projeção mensal, um campo com: melhor resultado, vendas do mês, vendas do dia e vendas da semana, e um campo com um gráfico de porcentagem da meta alcançada, sendo que o gráfico da porcentagem avança conforme as vendas.
- **Resultado:**
Todas as funções estão operando corretamente.



Figura 13 - Módulo Ranking

D.2.9 Teste: Módulo Indicadores

- **Requisitos verificados:**

RF34, RF35, RF36 e RF37.

- **Descrição:**

Teste da funcionalidade dos botões.

- **Verificações**

- O módulo de “Indicadores” é dividido entre venda e orçamentos, sendo ambos listados por data e exibindo o código e nome do parceiro, número e data e valor, com o campo de orçamentos exibindo também a opção de contato do parceiro.
- A visualização de “Vendas” lista notas de apenas das últimas 24 horas e a visualização de “Orçamentos” lista os 3 dias anteriores.
- Acessando a visualização de “Orçamentos” é possível realizar contato através de um botão que realiza uma ligação para o cliente referente a nota..

- **Resultado:**

Todas as funções estão operando corretamente.

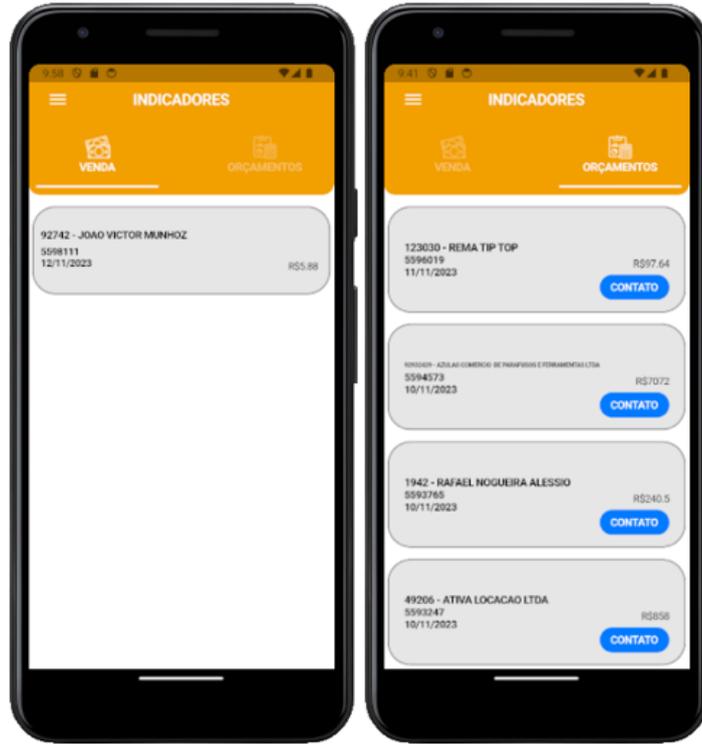


Figura 14 - Módulo Indicadores