



Universidade de Brasília

INSTITUTO DE CIÊNCIAS EXATAS
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Aplicativo para smartphones de apoio à
gerência e ao planejamento de estoque
de supermercados

DOUGLAS SAMUEL THOMAZI AZEVEDO
MATHEUS ARRUDA AGUIAR

MONOGRAFIA - CONCLUSÃO DE CURSO - CIÊNCIA DA COMPUTAÇÃO

ORIENTADOR
PROF. DR. WILSON HENRIQUE VENEZIANO

BRASÍLIA
2023



Universidade de Brasília

INSTITUTO DE CIÊNCIAS EXATAS

DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Aplicativo para smartphones de apoio à gerência e ao planejamento de estoque de supermercados

DOUGLAS SAMUEL THOMAZI AZEVEDO

MATHEUS ARRUDA AGUIAR

MONOGRAFIA - CONCLUSÃO DE CURSO - CIÊNCIA DA COMPUTAÇÃO

PROF. DR. WILSON HENRIQUE VENEZIANO(ORIENTADOR)

CIC/UNB

PROF. DR. MARCELO GRANDI MANDELLI

COORDENADOR DO CURSO DE CIÊNCIA DA COMPUTAÇÃO — BACHARELADO

BANCA AVALIADORA

PROF. DR. DÍBIO LEANDRO BORGES (CIC/UNB)

PROF. DR. JAN MENDONÇA CORREA (CIC/UNB)

BRASÍLIA, 21 DE JULHO DE 2023

Dedicatória

Eu dedico este trabalho aos meus pais, Roberto e Paula, à minha irmã Stéfany, à minha noiva Ana Clara, aos meus avós Aguiar, Miramar, Maria e Severiano (em memória) e à toda minha família e amigos. - Matheus Arruda Aguiar

Eu dedico este trabalho aos meus pais, José Carlos e Cristiane, aos meus irmãos Stefani, João Guilherme e Ana Luiza, à minha namorada Stéfany e à toda minha família e amigos. - Douglas Samuel Thomazi Azevedo

Agradecimentos

Agradecemos a toda nossa família que sempre nos ensinaram a buscar o conhecimento e sempre nos incentivaram e aconselharam durante toda jornada de vida.

Ao professor e nosso orientador, Dr. Wilson H. Veneziano, por ter nos auxiliado ao longo deste projeto e tornaram possível a idealização e elaboração do aplicativo.

A todos os professores do CIC, por terem nos ensinado grande parte dos conhecimentos necessários para a criação desse projeto.

Resumo

Neste trabalho de conclusão de curso de Ciência da Computação descreve-se o desenvolvimento de um aplicativo para *smartphones*, cujo objetivo é auxiliar de forma prática a gerência e planejamento de balanço de estoque de supermercados, como a consulta de produto pelo seu nome, marca ou lendo o código de barras pela câmera do aparelho, bem como a alteração de preço e quantidade no estoque, e por fim, a impressão de etiqueta. Ele atua em conjunto com o banco de dados de um software comercial para gestão de estoque já em operação no mercado. Foram levantados requisitos junto a gerentes de supermercados por meio de pesquisas de formulários de profissionais da área. Com todos os requisitos levantados, foi realizada a integração com o banco de dados desses mercados via API REST e com isso as telas foram implementadas. O *software* foi testado nos supermercados por um período de três semanas, tendo obtido avaliações positivas por parte dos usuários, colhidas via formulários eletrônicos.

Palavras-chave: *balanço de mercadoria, alteração de estoque, alteração de preço, impressão de etiqueta, aplicativo android*

Abstract

This Computer Science undergraduate thesis describes the development of an application for smartphones, whose objective is to help in a practical way the management and planning of supermarket stock balances, such as querying the product by its name, brand or reading the code of bars by the camera of the device, as well as the change of price and quantity in the stock, and finally, the printing of the label. Requirements were raised with supermarket managers along with surveys of forms by professionals in the area. With all the requirements raised, the integration with the database of these markets was carried out via REST API and with that the screens were implemented. The software was tested in supermarkets over a period of 3 weeks and received very positive feedback.

keywords: *merchandise balance, stock change, price change, label printing, android app*

Sumário

1	Introdução	1
1.1	Problema	1
1.2	Justificativa	2
1.3	Objetivo Geral	2
1.4	Objetivos Específicos	2
1.5	Metodologia	3
1.6	Organização do Trabalho	4
2	Balanco de Estoque de Mercadorias do Setor Mercantil	5
2.1	Balanco de Estoque de Mercadoria	5
2.2	Software <i>Desktop</i> dos Supermercados	6
2.3	Análise de Requisito	8
2.3.1	Fomulário	8
3	Desenvolvimento do Aplicativo	13
3.1	Processo de Software	13
3.2	Tecnologias Utilizadas	14
3.2.1	Android	14
3.2.2	Java	17
3.2.3	C#	17
3.2.4	Android Studio	19
3.2.5	Visual Studio Code	20
3.2.6	Banco de Dados	20
3.2.7	Github	21
4	O Aplicativo Desenvolvido	22
4.1	API	23
4.2	Arquitetura do Aplicativo	23

4.3	Banco de Dados	24
4.4	Leitura Código de Barras	25
4.5	Tela Inicial	25
4.6	Tela de Login	26
4.7	Alterar Preço	27
4.8	Alterar Estoque	28
4.9	Consultar Preço	29
4.10	Lista de Compras	30
4.11	Imprimir Etiqueta de Gôndola	35
4.12	Validação	36
5	Conclusão	43
	Referências Bibliográficas	45

Lista de Figuras

2.1	Diagrama <i>software</i> Desktop [Fonte: Documentação interna do Software] . .	6
2.2	Diagrama <i>software</i> Desktop com Aplicativo	7
3.1	Camadas da Arquitetura Android. [4]	15
4.1	Diagrama de fluxo de comunicação do Aplicativo	23
4.2	Arquitetura do Aplicativo	24
4.3	Banco de dados do aplicativo [Extraído do SQLite do aplicativo]	24
4.4	Funcionalidade Leitor de Código de Barras	25
4.5	Tela Inicial do aplicativo	26
4.6	Tela de Login	27
4.7	Tela de Alteração de Preço	28
4.8	Janela de confirmação de alteração de preço	28
4.9	Tela de Alteração de Estoque	29
4.10	Janela de confirmação de alteração de estoque	29
4.11	Tela de Consulta de Preço	30
4.12	Tela de Lista de Compras	31
4.13	Janela de confirmação de nova lista	31
4.14	Tela dos itens de uma lista de compras	32
4.15	Janela de exclusão de Lista de Compras	33
4.16	Janela de envio de lista de compras	33
4.17	Tela de Adição de itens a lista de compra	34
4.18	Janela de quantidade dos itens	34
4.19	Tela de Impressão de Etiqueta	35
4.20	Etiqueta impressa	35

Capítulo 1

Introdução

Nos últimos anos, a indústria do varejo tem passado por uma revolução impulsionada pelos avanços tecnológicos. Dentre as áreas mais impactadas, destaca-se o desafio constante do gerenciamento e controle de estoque enfrentado pelos supermercados. Garantir um controle adequado do estoque é fundamental para assegurar a disponibilidade dos produtos, evitar escassez ou excesso de mercadorias, bem como a perda de alguns produtos, otimizar os processos logísticos e, em última instância, proporcionar uma experiência de compra satisfatória para os clientes.

Ademais, o crescente uso de dispositivos móveis e o amplo acesso à internet abriram caminho para o desenvolvimento de aplicativos que buscam aprimorar a gestão de estoque nos supermercados. Essas soluções tecnológicas estão transformando a forma como os varejistas lidam com o controle de mercadorias, oferecendo maior eficiência, precisão e agilidade nas operações diárias.

O objetivo deste trabalho é apresentar o desenvolvimento de um aplicativo específico para o controle de estoque em supermercados, oferecendo uma solução eficaz para os desafios enfrentados nessa área. Por meio de tecnologias modernas pretende-se criar uma ferramenta intuitiva e de fácil utilização, capaz de otimizar as atividades relacionadas ao estoque.

1.1 Problema

Em um supermercado é necessário, de tempos em tempos, fazer o “balanço de mercadorias”, um processo trabalhoso e nada fácil, porque hoje em dia há muitos produtos diferentes, em grandes quantidades e todos eles precisam ser catalogados e contados para seu controle interno que é realizado no computador do supermercado. Tal tarefa pode ser

realizada de várias formas, como por exemplo, recolher um item de cada produto, anotar a quantidade em estoque e guardar esse item no carrinho, quando percorrer o produto de todas as marcas disponíveis, basta ir com o carrinho de compras até o computador, retirar item por item, olhar a quantidade que foi escrita e transcrever para o sistema de gerenciamento de estoque e por fim devolver o produto para a prateleira.

Com isso, o gerente do supermercado Y propôs, para a empresa prestadora de serviço de *software* do mercado, a ideia de criação de um aplicativo para *smartphone* que pudesse corroborar para que o trabalho longo e custoso de balanço de mercadorias pudesse ser realizado de forma mais eficiente, prática e em menor tempo, pois para tal prática o estabelecimento necessita estar fechado. Sendo assim, fizemos uma entrevista com esse gerente para que pudéssemos desenvolver um aplicativo que o atendesse da melhor forma, bem como todos os outros estabelecimentos que utilizam o sistema.

1.2 Justificativa

Atualmente, não há um aplicativo que auxilie no gerenciamento e controle de estoque em supermercados que utilizam especificamente o *software* da empresa em questão. Portanto, é fulcral o desenvolvimento de um aplicativo para *smartphone* no qual auxilie os varejistas nessa questão.

1.3 Objetivo Geral

Desenvolver um aplicativo de gerenciamento e controle de estoque de supermercados para *smartphones*, destinado a contribuir no processo de balanço do setor varejista em questão. Tal aplicativo atuará em conjunto com um *software* comercial para gerenciamento de estoque que já existe, porém disponível em versão somente para computadores, sendo necessário integrá-lo ao banco de dados já em operação. Em função disso e após pesquisas, não foi encontrado nenhum similar para comparação.

1.4 Objetivos Específicos

Para ser alcançado o objetivo geral, foram estabelecidos os seguintes objetivos específicos:

- elaborar um formulário para os gerentes de supermercados com as possíveis funcionalidades que terão no aplicativo;

- selecionar as reais funcionalidades que foram aprovadas;
- avaliar possíveis sugestões de novas funcionalidades;
- implementar a integração do aplicativo com o banco de dados dos supermercados;
- implementar a leitura de código de barras do produto a partir da câmera, pois será utilizada na maioria das funcionalidades.

1.5 Metodologia

O desenvolvimento do aplicativo envolveu a seguinte metodologia:

- estudar o funcionamento e a regra de negócio do *software* de supermercados;
- estudar as características de um balanço comercial de mercadorias em um estabelecimento de supermercado;
- estudar como seria realizada a integração com a API da empresa para que pudessemos ter acesso ao banco de dados do estabelecimento;
- estudar sobre o desenvolvimento de aplicações para *smartphone*;
- entrevistar gerentes dos supermercados para obter informações necessárias acerca das funcionalidades que seriam úteis de ter no aplicativo;
- levantar funcionalidades e características do aplicativo com gerentes especialistas no setor de supermercados;
- levantar requisitos a partir das entrevistas como dos formulários;
- desenvolver a API (Interface de Programação de Aplicação) e a comunicação com o banco de dados com a empresa fornecedora no sistema do mercado;
- desenvolver o aplicativo, realizar testes funcionais e avaliar.

1.6 Organização do Trabalho

O presente trabalho foi organizado em forma de capítulo e estes são:

- Capítulo 2: descreve brevemente como é realizado um balanço de mercadorias em um estabelecimento do setor de supermercados;
- Capítulo 3: apresenta o processo e características das tecnologias utilizadas no desenvolvimento do aplicativo;
- Capítulo 4: discorre sobre as funcionalidades dos *software* e sobre todas as telas do aplicativo e seus respectivos elementos, bem como o resultado da validação;
- Capítulo 5: expõe a conclusão do projeto.

Capítulo 2

Balanco de Estoque de Mercadorias do Setor Mercantil

Este capítulo apresentará os conceitos acerca de mercadorias, balanço de estoque de mercadorias voltados para o setor de varejo alimentício, também as tecnologias utilizadas no *software* desktop dos supermercados e as tecnologias escolhidas para o aplicativo, que se encontram descritas detalhadamente no Capítulo 3.

2.1 Balanço de Estoque de Mercadoria

O conceito de mercadoria é definido como: “toda coisa oferecida ao consumidor através da circulação econômica. O que caracteriza a mercadoria é a existência de um bem material posto em circulação econômica, para o consumo, mediante remuneração” [11]. Em um estabelecimento de varejo, mais especificamente, os supermercados têm como sustento integral a venda de mercadorias, sejam elas de gêneros alimentícios, limpeza ou higiene.

O conceito de estoque envolve a guarda de materiais essenciais em um sistema de produção, abrangendo diversas atividades produtivas que requerem a manutenção de diferentes tipos de materiais armazenados ou estocados.

O controle de estoques tem como objetivo primordial aumentar o controle de custos e aprimorar a qualidade dos produtos armazenados na organização. As teorias relacionadas a esse tema geralmente destacam a seguinte proposição: é viável determinar uma quantidade ideal de estoque para cada componente e produto da empresa, contudo, essa determinação só é possível por meio da previsão da demanda de consumo do produto [18].

Por conseguinte, é de suma importância que o balanço de mercadorias seja realizado de forma rotineira para que o estabelecimento obtenha reais informações sobre vendas e perdas para que assim consiga ter aproveitamentos e, conseqüentemente lucros, cada vez maiores.

2.2 Software *Desktop* dos Supermercados

O *software* foi pioneiro na região norte de Minas Gerais atuando na área de automação comercial, teve seu início em 2002 na cidade de Paracatu - MG, desde então vem buscando acompanhar as novas tecnologias e as necessidades do mercado, atualmente, conta com mais de 400 clientes ativos em diversos estados e atuando em supermercados, bares, marmorarias, auto peças, home centers e boutiques.

Desenvolvido em Delphi e utilizando uma base de dados MySQL, o sistema é dividido em duas partes, o Retaguarda e o PDV, conforme o diagrama da Figura 2.1. O Retaguarda é destinado a parte administrativa da empresa, como o cadastro de produtos, entrada de notas, gerenciador financeiro e para visualizar relatórios. Por outro lado, O PDV (Ponto de Venda) é o sistema utilizado nos caixas, tendo como principais funções realizar vendas e emitir cupons fiscais. Para que não haja necessidade de conexão com a internet, ambas as partes do sistema operam localmente e cada computador possui seu próprio banco de dados, quando os computadores são conectados a rede, o servidor verifica a alteração de cada banco de dados e unifica as informações.

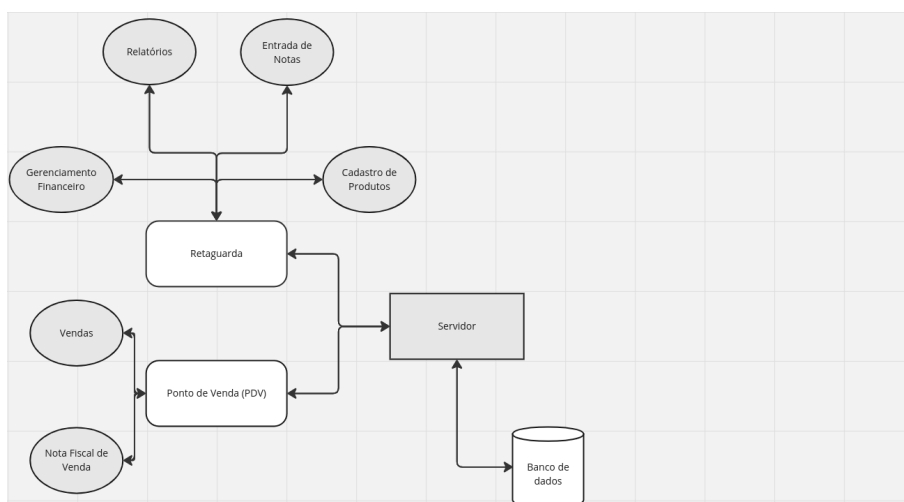


Figura 2.1: Diagrama *software* Desktop [Fonte: Documentação interna do Software]

O sistema visa auxiliar o empresário na gerência do seu comércio, desde suas vendas até o controle financeiro da empresa, porém, com a alta aderência a regimes tributários

mais complexos, a necessidade de um controle de estoque mais rígido vem se tornando cada vez maior. Em regimes tributários como o Lucro Real [10], é imprescindível que a quantidade e o preço de um produto no seu sistema esteja condizente com a quantidade e o preço do produto nas prateleiras. Por essa razão, os desenvolvedores do *software* vem buscando diariamente formas de tornar mais fácil e prático o controle do estoque. Em função desse problema, o aplicativo é de suma importância para performar o *software* desktop de modo a facilitar o controle das mercadorias de modo mais dinâmico sem ter que utilizar o Retaguarda sempre que precisar consultar/alterar um preço, imprimir etiqueta ou fazer o balanço do comércio por inteiro. Dessa forma, a seguinte Figura 2.2 ilustra como aplicativo complementa o sistema.

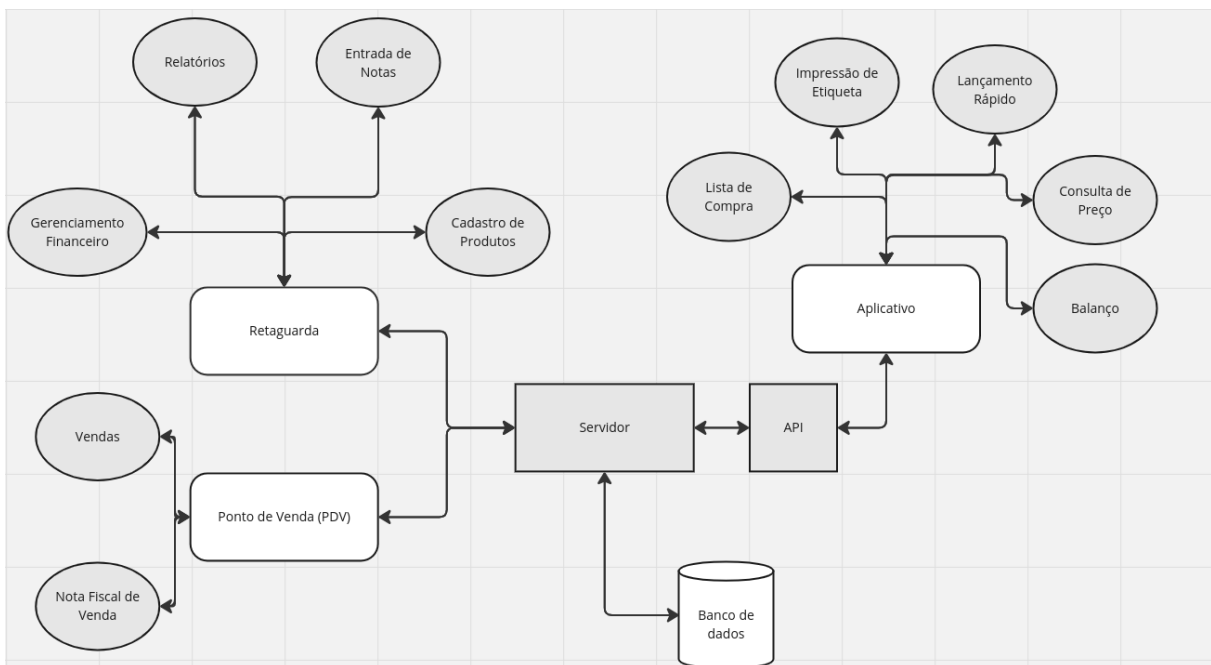


Figura 2.2: Diagrama *software* Desktop com Aplicativo

2.3 Análise de Requisito

Dado o problema em 1.1, foi realizado, nos principais supermercados que a empresa de *software* que atua, perguntas em formato de formulário para que assim pudesse ser realizado um aplicativo mais completo possível que atendesse todas as necessidades. Com a coleta de informações da entrevista, foi realizado um questionário para que 10% da rede que utiliza o sistema o respondesse.

2.3.1 Fomulário

Cada item de funcionalidade foi colocada para ser respondido com escala de 1 a 5, onde 1 seria pouco útil e 5 seria muito útil. Ao final do formulário foi colocado um espaço de texto livre para que o participante pudesse fazer sugestões de funcionalidades, porém não houve respostas. Com esse resultado, começamos a planejar como seriam as telas do aplicativo e quais teriam cada funcionalidade escolhida a partir da análise de requisitos, sendo elas:

- Consulta de preço: verificar preço de venda, atacado, promoção e estoque;
- Balanço: alterar quantidade do produto em estoque;
- Lançamento rápido: alteração preço de custo e venda;
- Imprime etiqueta de gôndola: enviar a impressão da etiqueta de gôndola para a impressora;
- Controle de validade do produto: Consultar e alterar a validade dos produtos;
- Lista de compra: Informar produtos que estão em falta no estoque.

O detalhamento do formulário para levantamento das funcionalidades está no seguinte anexo em pdf abaixo, com título: *Recursos desejáveis no aplicativo - Balanço Otimizado*.

O próximo capítulo trata dos aspectos da fase de desenvolvimento do aplicativo.

Recursos desejáveis no Aplicativo - Balanço Otimizado

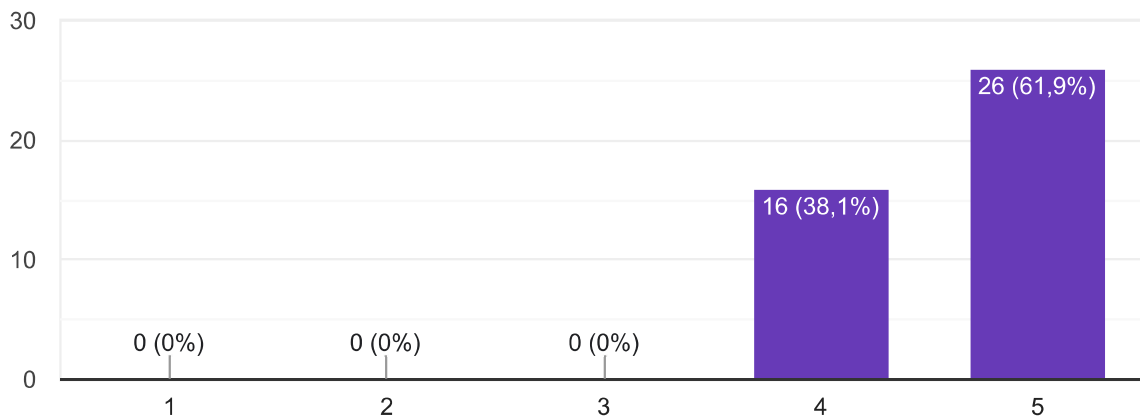
42 respostas

[Publicar análise](#)

Funcionalidade: Verificar quantidade de produto na prateleira lendo código de barras pela câmera do smartphone ou inserindo-o manualmente.

[Copiar](#)

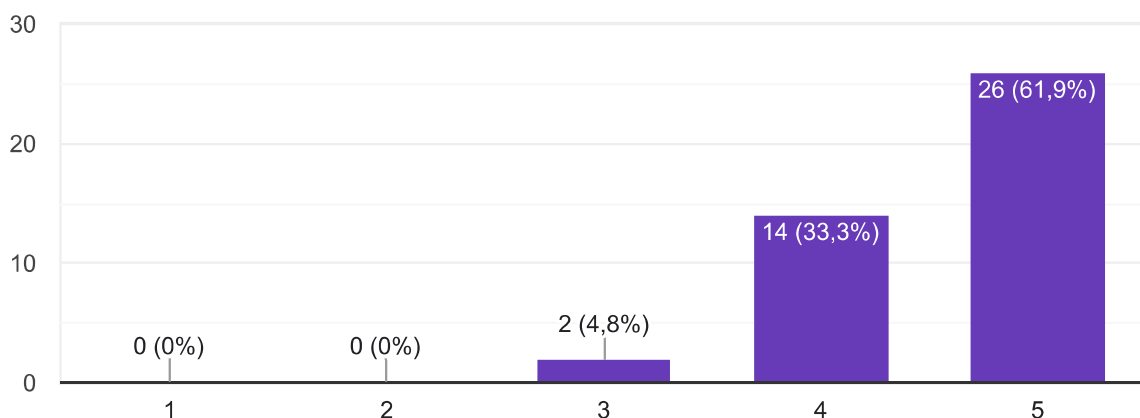
42 respostas



Funcionalidade: Consultar data de validade do produto lendo código de barras pela câmera do smartphone ou inserindo-o manualmente.

[Copiar](#)

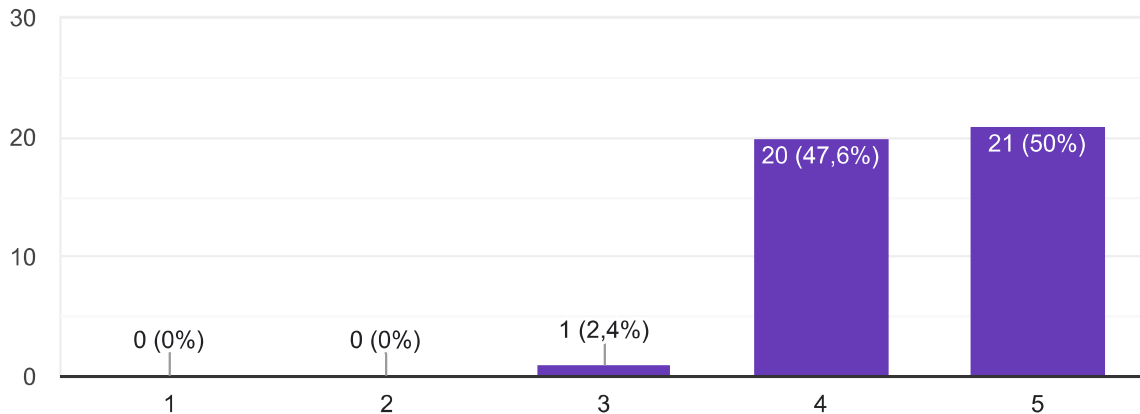
42 respostas



Funcionalidade: Alterar de data de validade do produto lendo código de barras pela câmera do smartphone ou inserindo-o manualmente.



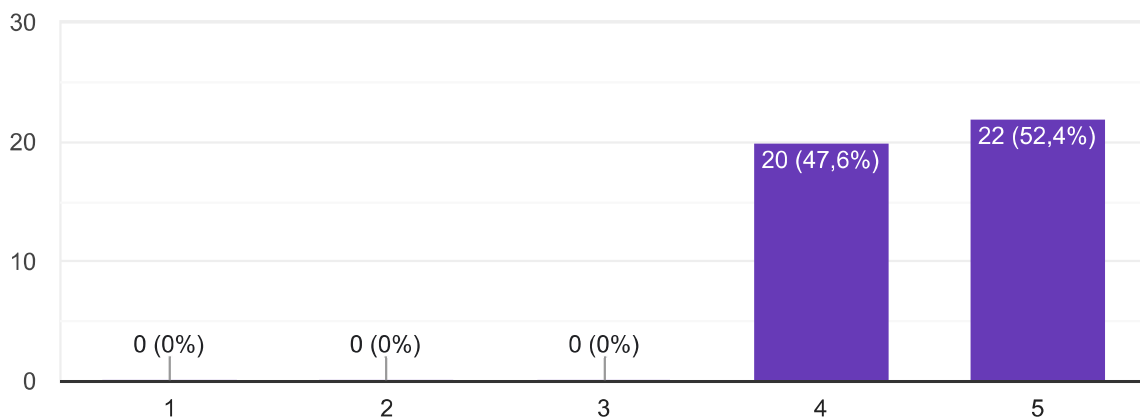
42 respostas



Funcionalidade: Alterar preço de venda do produto lendo código de barras pela câmera do smartphone ou inserindo-o manualmente.



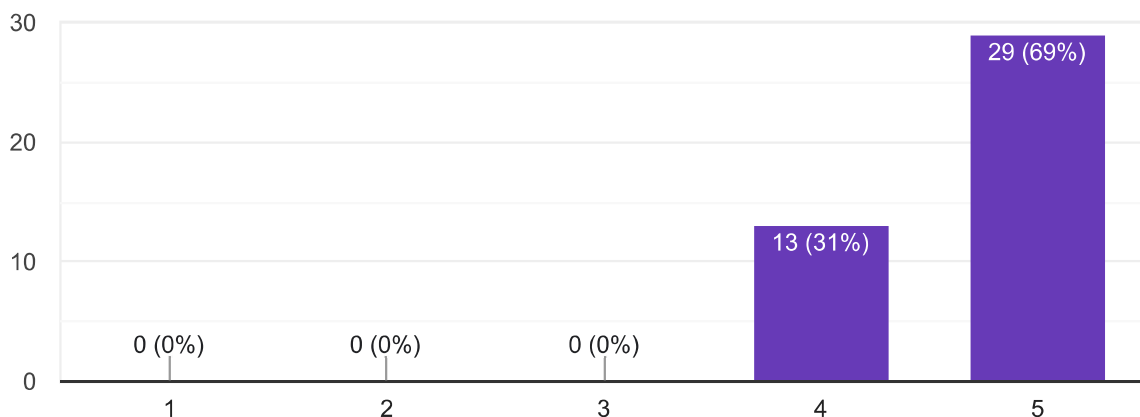
42 respostas



Funcionalidade: Imprimir etiqueta de preço do produto.



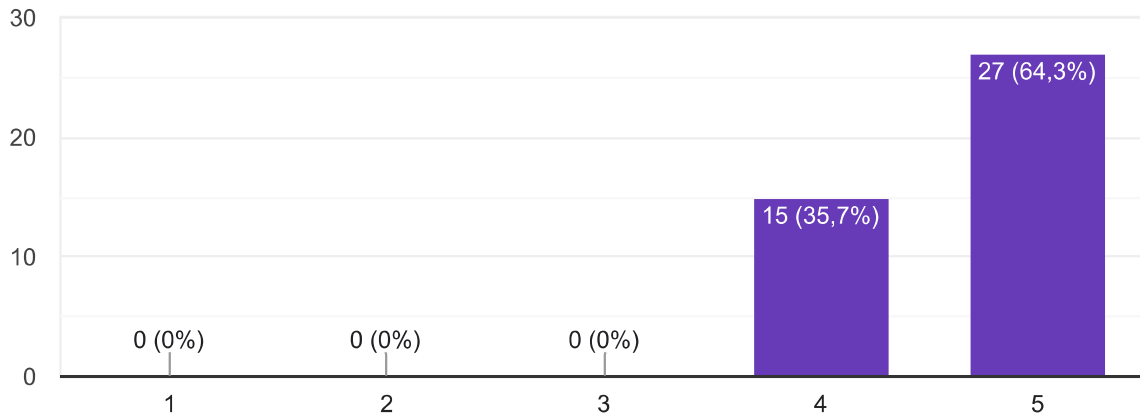
42 respostas



Funcionalidade: Consultar preço de venda do produto lendo código de barras pela câmera do smartphone ou inserindo-o manualmente.



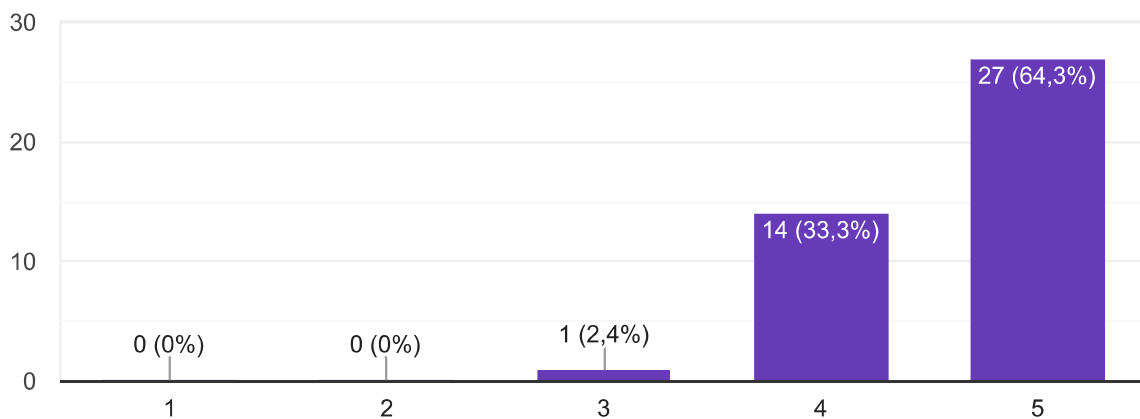
42 respostas



Funcionalidade: Consultar preço de atacado do produto lendo código de barras pela câmera do smartphone ou inserindo-o manualmente.



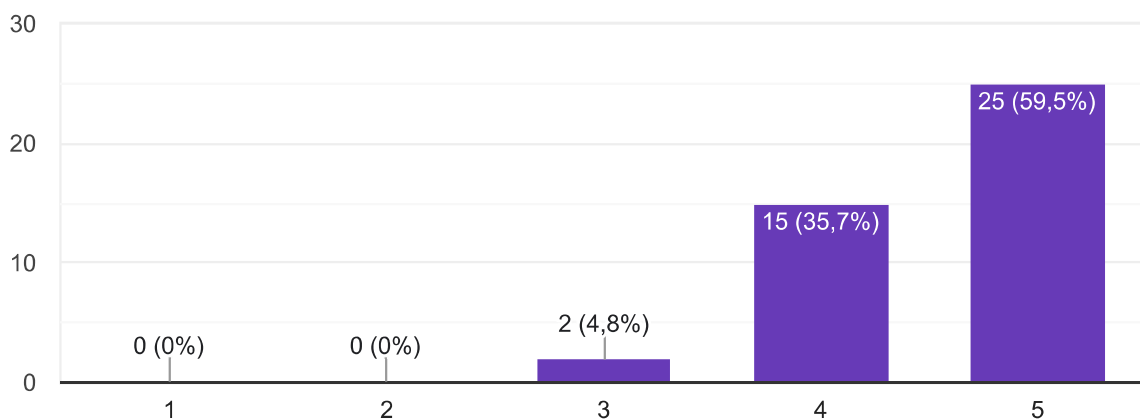
42 respostas



Funcionalidade: Consultar preço de promoção do produto lendo código de barras pela câmera do smartphone ou inserindo-o manualmente.



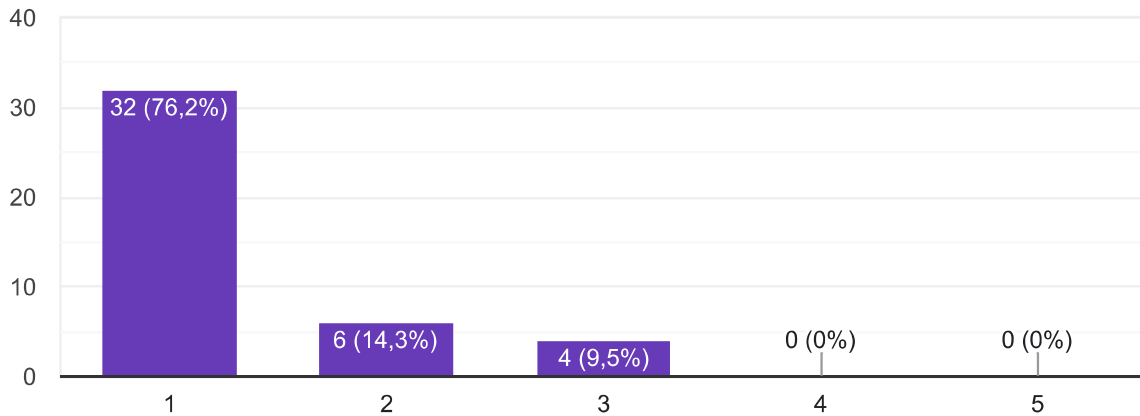
42 respostas



Funcionalidade: Alterar preço de custo do produto lendo código de barras pela câmera do smartphone ou inserindo-o manualmente.



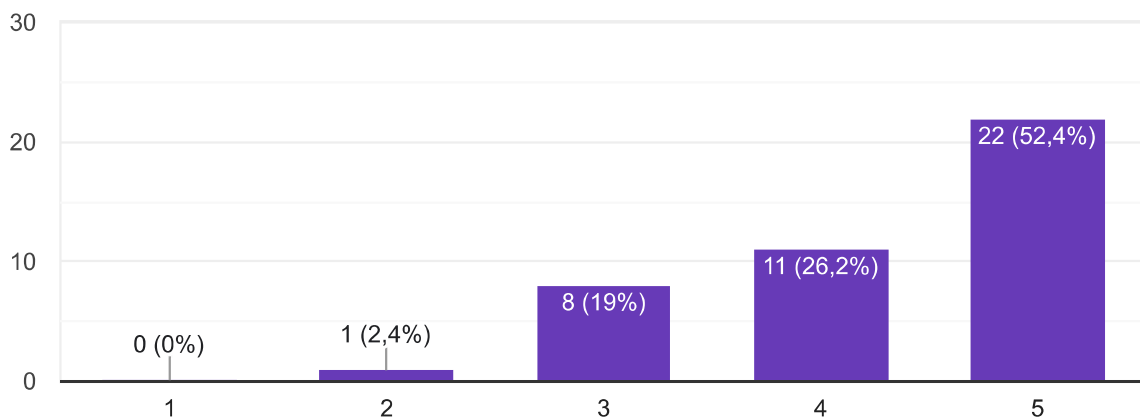
42 respostas



Funcionalidade: Lista de compra - Informar produtos que estão em falta no estoque.



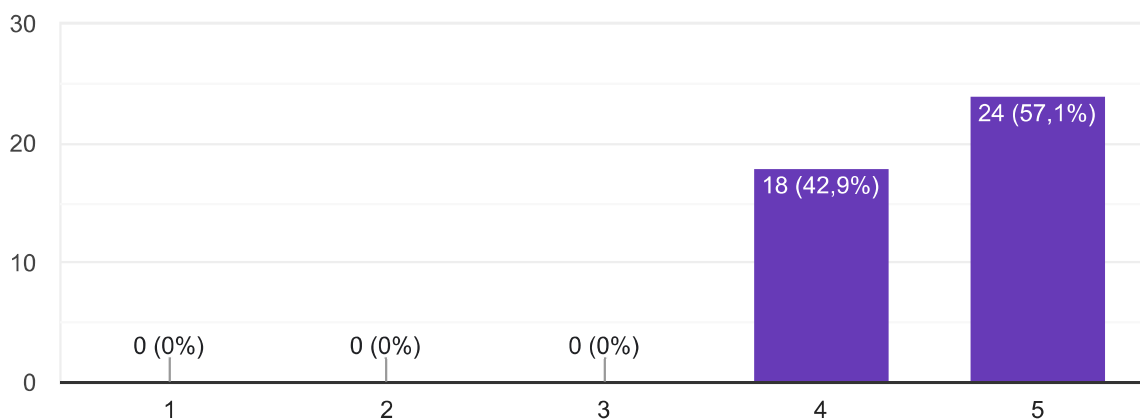
42 respostas



Funcionalidade: Alterar quantidade do produto lendo código de barras pela câmera do smartphone ou inserindo-o manualmente.



42 respostas



Capítulo 3

Desenvolvimento do Aplicativo

Neste capítulo serão abordados, de forma geral, as ferramentas, recursos, linguagem de programação, tecnologias e demais elementos utilizados na elaboração do desenvolvimento do *software* para *smartphone*.

3.1 Processo de Software

Um processo de *software* é composto por uma série de atividades inter-relacionadas que visam desenvolver e entregar um produto de *software* [20]. Essas atividades são organizadas de maneira sistemática e sequencial, seguindo metodologias e abordagens específicas, com o objetivo de atingir os objetivos do projeto de *software*. Segundo Pressman [17], uma metodologia de processo de *software* envolve cinco atividades essenciais: comunicação, planejamento, modelagem, construção e entrega.

Com base no exposto acima, o desenvolvimento do aplicativo seguiu esse critério, sempre com muita comunicação com os profissionais do supermercado para que assim fosse realizado um trabalho essencial que suprisse a demanda de controle e planejamento de estoque do estabelecimento, bem como formulário de pesquisa com as possíveis funcionalidades e possíveis propostas de funcionalidades sugeridas por esses profissionais; após a comunicação foi realizado o planejamento tanto para a análise de requisitos quanto para construção do aplicativo e divisões das demandas; a modelagem foi realizada por parte dos desenvolvedores, pois seria necessário além de realizar o desenvolvimento do aplicativo, realizar também o desenvolvimento da API para que pudesse ser consumido o banco de dados do supermercado; com o planejamento e modelagem já definidos, foi começada a construção do *software* com duração de aproximadamente 7 meses e por fim a entrega do protótipo para os supermercados fazerem testes e avaliarem o produto.

3.2 Tecnologias Utilizadas

Nesta seção, serão apresentadas as ferramentas e tecnologias adotadas no projeto, bem como o ambiente geral de desenvolvimento. Isso inclui os sistemas operacionais selecionados, a linguagem de programação e o controle de versão do código. A escolha dessas ferramentas e tecnologias levou em consideração critérios como curva de aprendizado, facilidade de uso e disponibilidade de documentação bem estruturada e acessível. Foram utilizadas as seguintes tecnologias:

- Sistema Operacional: Android;
- Linguagem de Programação: Java e C#;
- Banco de Dados: SQLite;
- IDE: Android Studio e Visual Studio Code;
- Versionamento do Código: Github.

3.2.1 Android

O Android é um sistema operacional de código aberto para dispositivos móveis, desenvolvido com base no sistema operacional Linux e é composto por um conjunto de ferramentas que atua em todas as fases do desenvolvimento do projeto, desde a execução até a criação de *softwares* específicos, para celular e *tablet*, lançado comercialmente pela Google em 2008 [4].

O Android, com sua interface de usuário baseada na manipulação direta, é projetado principalmente para dispositivos móveis com telas sensíveis ao toque, como *smartphones* e *tablets*. Também oferece interfaces específicas para TV, carros e *smartwatches*. O sistema operacional aproveita a tela sensível ao toque para permitir que os usuários manipulem objetos virtuais, semelhantes a um teclado virtual. Embora seja predominantemente usado em dispositivos *touchscreen*, o Android também é utilizado em consoles de videogame, câmeras digitais, computadores e outros dispositivos eletrônicos [2].

O Android tem uma capacidade de segurança robusta. Executando em um *kernel* Linux, sempre que um aplicativo é instalado em um dispositivo Android, um novo usuário Linux é criado especificamente para aquele programa. Este usuário possui seus próprios diretórios que serão utilizados pela aplicação, mas somente para aquele usuário Linux. Como resultado, os aplicativos ficam completamente isolados uns dos outros e qualquer tentativa de acesso a informações de outro aplicativo requer autorização explícita do

usuário. O usuário tem a opção de negar a instalação do aplicativo ou permitir com permissões controladas por meio de um mecanismo de permissão [3].

Arquitetura

O Sistema Operacional tem sua arquitetura com as seguintes camadas [4]. Figura 3.1.

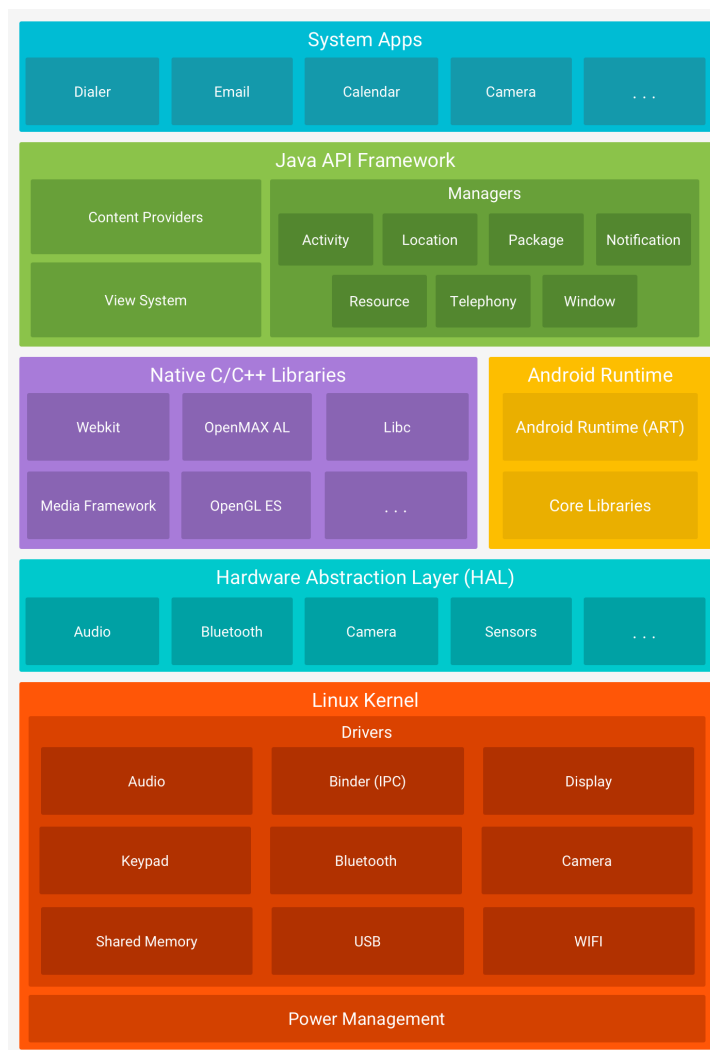


Figura 3.1: Camadas da Arquitetura Android. [4]

- **Aplicativos do sistema**

O Android vem pré-instalado com um conjunto de aplicativos principais, como e-mail, envio de SMS, calendário, navegador de internet, contatos, entre outros. Os aplicativos incluídos na plataforma não possuem status especial em relação aos aplicativos de terceiros que o usuário opta por instalar. Isso significa que um aplicativo de terceiros pode se tornar o navegador da web, o aplicativo de envio de SMS ou até mesmo o teclado padrão do usuário (com exceção de alguns aplicativos, como o aplicativo "Configurações do sistema").

- **Java API Framework**

Essa camada engloba os elementos essenciais para o desenvolvimento de aplicativos que fazem uso dos recursos do sistema operacional, incluindo recursos de gerenciamento de notificações e compartilhamento de dados com a agenda e a lista de contatos do sistema. No contexto do Android, é importante ressaltar que os aplicativos desenvolvidos por terceiros possuem o mesmo acesso a essa camada que os aplicativos do sistema.

- **Bibliotecas C/C++ nativas**

As bibliotecas nativas do sistema Android compreendem vários componentes e serviços essenciais, como ART (Android Runtime) e HAL (*Hardware Abstraction Layer*), que são implementados por meio de código nativo e requerem bibliotecas nativas programadas em C e C++. A plataforma Android disponibiliza as APIs do Framework Java para expor a funcionalidade de algumas dessas bibliotecas nativas aos aplicativos.

- **Android Runtime (ART)**

O ART é um ambiente de tempo de execução projetado para executar várias máquinas virtuais em dispositivos de baixa memória arquivos DEX, um formato de *bytecode* projetado especialmente para Android, otimizado para oferecer consumo mínimo de memória. Construir cadeias de ferramentas, como Jack, e compile fontes Java em *bytecode* DEX, que podem ser executadas na plataforma Android e oferece ao desenvolvedor um ótimo suporte à ferramenta *debug*, com diagnósticos para exceções e relatórios de falha detalhados.

- **Camada de abstração de *hardware* (HAL)**

A camada HAL é responsável por fornecer interfaces padrões a fim de expor as capacidades de *hardware* do dispositivo aparelho como câmera e *bluetooth* para a estrutura da Java API de maior nível. Quando uma *Framework* API faz uma chamada para acessar o *hardware* do dispositivo, o sistema Android carrega o módulo da biblioteca para este componente de *hardware*.

- ***Kernel* do Linux**

O *kernel* é a camada mais fundamental na arquitetura do Android, incorporando uma versão do *kernel* do Linux padrão. Ele aproveita os recursos de segurança do *kernel* Linux, ao mesmo tempo em que oferece adições específicas relacionadas ao gerenciamento de memória, energia e outros aspectos.

3.2.2 Java

Java é uma linguagem de programação orientada a objetos, originalmente desenvolvida pela Sun Microsystems. Ela possui a capacidade de criar uma variedade de aplicativos, incluindo aplicativos para desktop, aplicações comerciais, *softwares* robustos, completos e independentes, bem como aplicativos para a Web [19].

- **Java Development Kit - JDK**

O JDK é uma coleção de ferramentas disponibilizada gratuitamente pela Sun Microsystems. Ele é composto por um conjunto de programas que inclui um compilador, um interpretador e utilitários diversos. A primeira versão do JDK lançada foi a 1.0, e atualmente está na versão 1.4.x, havendo também uma versão beta recente, a 1.5.x, também conhecida como Tiger. O JDK é dividido em três edições distintas: Java 2 Standard Edition (J2SDK), Java 2 Enterprise Edition (J2EE) e Java 2 Micro Edition (J2ME). Cada uma dessas edições possui seu próprio conjunto de pacotes que proporcionam aos usuários uma maneira organizada e diferenciada de desenvolver aplicações. Em outras palavras, aqueles que desejam criar aplicações para dispositivos como Palm Tops, celulares e outros dispositivos pequenos devem utilizar o J2ME como base para o desenvolvimento de suas aplicações [19];

Os principais componentes do kit de desenvolvimento são:

- javac (compilador);
- java (interpretador);
- appletviewer (visualizador de applets);
- javadoc (gerador de documentação);
- jar (programa de compactação).

3.2.3 C#

O C# é uma linguagem de programação moderna, orientada a objetos e fortemente tipada. Com o C#, os desenvolvedores têm a capacidade de criar uma ampla variedade de aplicativos seguros e robustos que são executados no .NET. O C# é especialmente projetado para suportar os conceitos de programação orientada a objetos e orientada a componentes, fornecendo construções de linguagem que facilitam a criação e o uso de componentes de *software*. Desde o seu surgimento, o C# tem incorporado recursos para

atender às novas demandas de trabalho e práticas emergentes de design de *software*. Em sua essência, o C# permite que você defina os tipos e o comportamento desses tipos de forma clara e concisa [7].

- **Arquitetura do .NET:** Os programas escritos em C# são executados no ambiente .NET, que consiste no *Common Language Runtime (CLR)*, uma máquina virtual de execução, juntamente com um conjunto de bibliotecas de classes. O CLR é a implementação da Microsoft da *Common Language Infrastructure (CLI)*, um padrão internacional que proporciona a base para criar ambientes de execução e desenvolvimento nos quais várias linguagens e bibliotecas funcionam de forma integrada.

O código-fonte em C# é compilado em uma linguagem intermediária chamada IL (Intermediate Language) que está em conformidade com a especificação da CLI. O código IL, juntamente com recursos como imagens e strings, é armazenado em um *assembly*, geralmente com a extensão .dll. Um *assembly* contém um manifesto que fornece informações sobre os tipos, a versão e a cultura.

Quando um programa em C# é executado, o *assembly* é carregado no CLR. Em seguida, o CLR realiza a compilação JIT (*Just-In-Time*) para converter o código IL em instruções nativas da máquina. O CLR também fornece outros serviços relacionados à coleta automática de lixo, tratamento de exceções e gerenciamento de recursos. O código que é executado pelo CLR é comumente chamado de "código gerenciado", enquanto o "código não gerenciado" é compilado em linguagem de máquina nativa e direcionado para uma plataforma específica.

A interoperabilidade de linguagens é um recurso fundamental do .NET. O código IL gerado pelo compilador C# está em conformidade com a *Common Type Specification (CTS)*. Isso significa que o código IL gerado a partir do C# pode interagir com código gerado por outras linguagens .NET, como F#, Visual Basic e C++. Mais de 20 outras linguagens são compatíveis com a CTS. Um único *assembly* pode conter módulos escritos em diferentes idiomas do .NET, e os tipos podem se referenciar mutuamente como se fossem escritos na mesma linguagem.

Além dos serviços de tempo de execução, o .NET também inclui extensas bibliotecas que fornecem suporte para uma ampla variedade de cargas de trabalho. Essas bibliotecas são organizadas em *namespaces* e oferecem funcionalidades úteis, desde operações de entrada e saída de arquivos até manipulação de strings, análise de XML e estruturas para aplicativos da web e controles do *Windows Forms*. Um aplicativo típico em C# faz amplo uso da biblioteca de classes para lidar com tarefas comuns relacionadas à "conexão".

- **API**

Essa linguagem foi utilizada na criação da API de arquitetura REST (transferência de estado representacional) para que fosse possível acessar a camada de dados do servidor. Para ser REST é necessário contém as seguintes conformidades:[9]

- Ter uma arquitetura cliente/servidor formada por clientes, servidores e recursos, com solicitações gerenciadas por HTTP;
- Estabelecer uma comunicação *stateless* entre cliente e servidor. Isso significa que nenhuma informação do cliente é armazenada entre solicitações GET e toda as solicitações são separadas e desconectadas;
- Armazenar dados em *cache* para otimizar as interações entre cliente e servidor;
- Ter uma interface uniforme entre os componentes para que as informações sejam transferidas em um formato padronizado;
- Ter um sistema em camadas que organiza os tipos de servidores (responsáveis pela segurança, pelo carregamento de carga e assim por diante) envolvidos na recuperação das informações solicitadas em hierarquias que o cliente não pode ver.

3.2.4 Android Studio

O Android Studio é o ambiente de desenvolvimento integrado (IDE) oficial para o desenvolvimento de apps Android e é baseado no IntelliJ IDEA, versão *open-source* e oferecem recursos para aumentar a produtividade na criação de apps Android, tais como: [6]

- um emulador Android nativo para o desenvolvimento sem um dispositivo físico;
- *debug*;
- ferramentas de análise de código para reconhecer problemas de performance, usabilidade;
- integração com ferramentas de versionamento de código;
- auto-formatação de arquivos para um padrão de indentação.

3.2.5 Visual Studio Code

O Visual Studio Code, desenvolvido pela Microsoft, é um editor de código-fonte disponível para Windows, Linux e macOS. Ele oferece recursos abrangentes, como depuração, controle de versionamento Git integrado, realce de sintaxe, preenchimento automático inteligente, trechos de código e refatoração. Além disso, é altamente personalizável, permitindo aos usuários modificar o tema do editor, atalhos de teclado e preferências de acordo com suas necessidades. O Visual Studio Code é um *software* de código aberto e livre para uso e foi lançado em 2015 sob a licença MIT.

O Visual Studio Code é compatível com várias linguagens de programação e oferece um conjunto de recursos que podem estar disponíveis ou não para cada linguagem. Muitos dos recursos do Visual Studio Code não são acessíveis por meio de menus ou da interface do usuário tradicional. Em vez disso, eles podem ser acessados por meio da paleta de comandos ou por meio de arquivos JSON, como as preferências do usuário [8].

3.2.6 Banco de Dados

O banco de dados é uma ferramenta fundamental para armazenar, gerenciar e recuperar informações de forma eficiente. Segundo Connolly e Begg [14], um banco de dados é uma coleção organizada de dados relacionados que são armazenados e acessados eletronicamente. Ele é projetado para atender às necessidades de uma organização, permitindo que informações sejam armazenadas de maneira estruturada e sejam facilmente recuperadas quando necessário.

O sistema de gerenciamento de banco de dados (SGBD) é responsável pelo controle e manipulação dos dados em um banco de dados. Um SGBD é como um *software* que fornece uma interface para que os usuários possam interagir com o banco de dados, executar consultas, inserir, atualizar e excluir registros, além de garantir a integridade, segurança e eficiência dos dados armazenados [16].

Existem diferentes tipos de bancos de dados, como o banco de dados relacional, que é baseado no modelo relacional proposto por Codd [15]. Nesse modelo, os dados são organizados em tabelas relacionadas entre si por meio de chaves primárias e estrangeiras. Já os bancos de dados não relacionais, como os bancos de dados orientados a documentos e bancos de dados de grafos, são projetados para atender a necessidades específicas, como a escalabilidade e a flexibilidade no armazenamento e recuperação de dados.

SQLite

O SQLite é baseado no modelo relacional de banco de dados, onde os dados são organizados em tabelas, que consistem em colunas e linhas. Ele suporta a maioria das operações comuns de banco de dados, como consultas SQL, criação e modificação de tabelas, inserção e atualização de dados, além de suportar transações ACID (Atomicidade, Consistência, Isolamento e Durabilidade) para garantir a integridade dos dados [13].

SQLite é uma biblioteca de linguagem C que implementa um mecanismo de banco de dados SQL pequeno, rápido, independente, de alta confiabilidade e completo. Além de ser o mais utilizado no mundo, o SQLite vem nativamente implementado no Android [5].

3.2.7 Github

O Git é um sistema de controle de versionamento distribuído que oferece rapidez, escalabilidade e uma ampla gama de comandos, proporcionando operações avançadas e acesso completo aos detalhes internos. Um sistema de controle de versionamento registra as alterações em arquivos ou conjuntos de arquivos ao longo do tempo, permitindo que você recupere versões específicas posteriormente [1].

Um repositório ou projeto Git abrange todos os arquivos e pastas associados a um projeto, juntamente com o histórico de revisões de cada arquivo. O histórico de arquivos é representado como instantâneos no tempo chamados *commits*. Os *commits* podem ser organizados em diferentes linhas de desenvolvimento chamadas *branches*. Como o Git é um sistema de controle de versionamento distribuído, os repositórios são unidades autossuficientes e qualquer pessoa que possua uma cópia do repositório pode acessar todo o código e seu histórico. Utilizando a linha de comando ou outras interfaces, um repositório Git também permite interagir com o histórico, clonar o repositório, criar *branches*, fazer *commits*, mesclar alterações e comparar versões de código.

O GitHub é uma plataforma que hospeda repositórios do Git e fornece aos desenvolvedores ferramentas para enviar código por meio de recursos como a linha de comando, problemas (discussões encadeadas), solicitações de *pull*, revisão de código e uma coleção de aplicativos. Com recursos de colaboração, como o fluxo de trabalho do GitHub, uma comunidade de 15 milhões de desenvolvedores e um ecossistema com centenas de integrações, o GitHub está transformando a maneira como o *software* é construído [1].

Capítulo 4

O Aplicativo Desenvolvido

O aplicativo foi projetado com intuito de melhorar e agilizar o serviço de balanço de estoque de supermercado, correção de preços, lista de compras para o fornecedor, leitura do código de barras pela câmera do aparelho, erro de preço de etiqueta e conseqüentemente seu envio para impressão. Os requisitos e regra de negócio do aplicativo foram realizadas em conjunto com o gerente do supermercado com reuniões, concepção de formulário para outros supermercados que utilizam o mesmo *software* desktop para que assim pudesse ser desenvolvido um aplicativo leve, intuitivo e que atendesse a necessidade dos usuários.

Foram estudadas as regras de negócio de um supermercado, nas quais são complexas, bem como conhecer de forma aprofundada como é realização um balanço de estoque de mercadorias. Com esses conhecimentos adquiridos e com a uma maior proximidade nessa área, foi possível desenvolver outras funcionalidades que foram além do balanço de estoque.

O aplicativo foi estruturado com o objetivo de ser simples e intuitivo, demonstrado a partir da Seção 4.5 deste capítulo, começando a tela de menu principal para que o usuário escolha a funcionalidade que deseja acessar e assim será redirecionado para a respectiva tela. Dependendo da funcionalidade desejada é necessário passar por um processo de autenticação utilizando nome de usuário e senha. Vale ressaltar que a tela autenticação não foi desenvolvida quando o aplicativo foi enviado para teste, mas somente após essa fase devido aos *feedback* retornados no período de validação, visto que as funcionalidades de alteração de preço e alteração de estoque necessitam de uma segurança para ser modificadas. Apesar de ter sido difícil encontrar pessoas dispostas a testarem o aplicativo e validarem, tivemos êxito nessa tarefa, após procurarmos incessantemente sem desistir do número adequado de estabelecimentos, dentre os quais elogiaram bastante o projeto.

4.1 API

O diagrama de fluxo de comunicação do Aplicativo que representa a integração de toda a arquitetura de informação da aplicação pode ser visualizado na Figura 4.1. Essa Figura ilustra as requisições feitas pela interface do aplicativo à API de comunicação, que, por sua vez, acessa a camada de dados no servidor do supermercado e retorna ou altera as informações necessárias para o aplicativo. Esse fluxo ocorre em todas as funcionalidades das telas que serão descritas ao longo do capítulo.

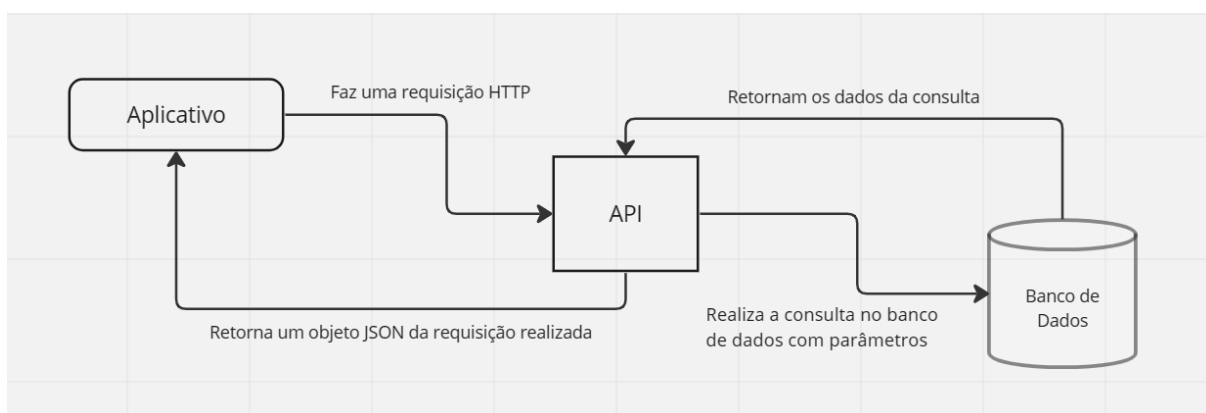


Figura 4.1: Diagrama de fluxo de comunicação do Aplicativo

4.2 Arquitetura do Aplicativo

A organização de um *software* está intrinsecamente relacionada à sua arquitetura, influenciando diretamente o desempenho, confiabilidade, disponibilidade, qualidade e outros requisitos não funcionais. A arquitetura é definida como um conjunto de elementos arquiteturais organizados e inter-relacionados [12], estabelecidos através de decisões tomadas para atender a objetivos e restrições específicos. O aplicativo consiste em uma tela inicial e 7 telas, cada uma para sua respectiva tarefa.

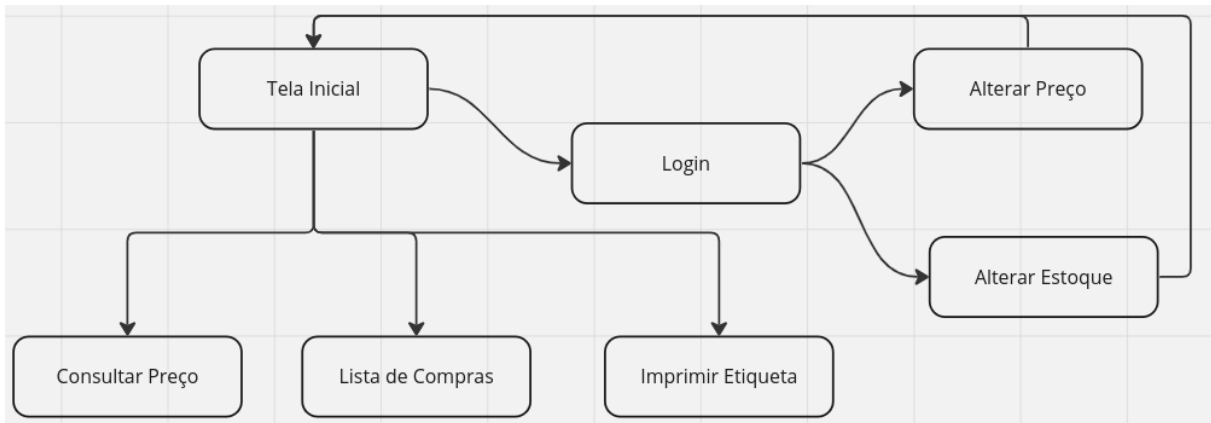


Figura 4.2: Arquitetura do Aplicativo

4.3 Banco de Dados

Praticamente todas as informações contidas no aplicativo são buscadas na API em tempo de execução, e em seis das sete telas não é necessário armazenar nenhum dado, com exceção da Lista de compras, serão demonstradas no próximo capítulo. Nessa tela as listas feitas precisam ficar salvas no celular, por essa razão, foi criado um banco de dados SQLite, ilustrado na Figura 4.3, com apenas duas tabelas: *lista* e *itenslista*.

A tabela *lista* armazena o *id* da lista, o nome e a data de criação. Já a tabela *itenslista* armazena o id da lista referenciada, o código interno do produto, a descrição, o código de barras e a quantidade que deseja comprar.

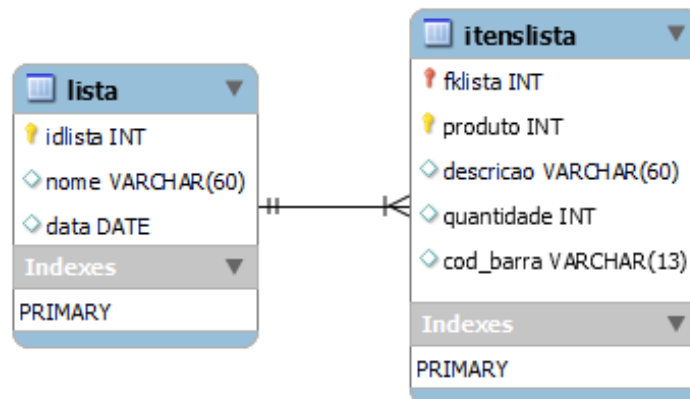


Figura 4.3: Banco de dados do aplicativo [Extraído do SQLite do aplicativo]

4.4 Leitura Código de Barras

A funcionalidade de leitura do produto pelo seu código de barras pode ser utilizado em várias telas para reconhecer a mercadoria, basta focar o código de barras na câmera do *smartphone*, como demonstra a Figura 4.4, fazendo-se assim a pesquisa do produto com base no seu respectivo código.

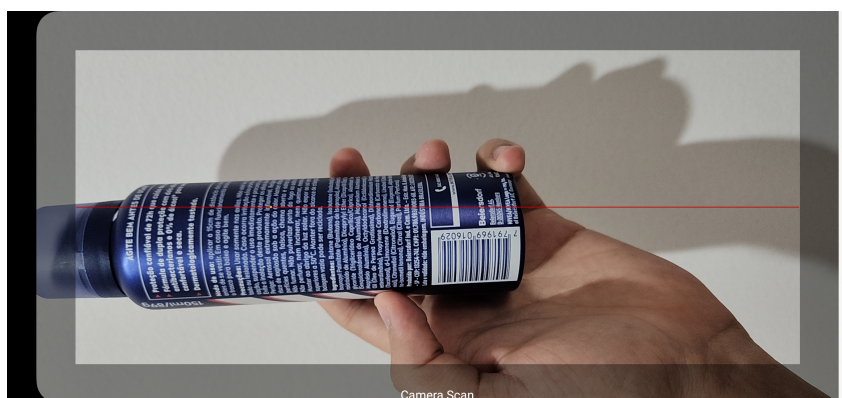


Figura 4.4: Funcionalidade Leitor de Código de Barras

4.5 Tela Inicial

Ao abrir o aplicativo, o usuário será introduzido a tela inicial, conforme mostra a Figura 4.5.

A tela contém os botões que necessitam de autenticação (login e senha) que são:

- Alterar Preço e Alterar Estoque;

E os botões que não necessitam de autenticação:

- Consultar Preço, Lista de Compras e Imprimir Etiqueta.

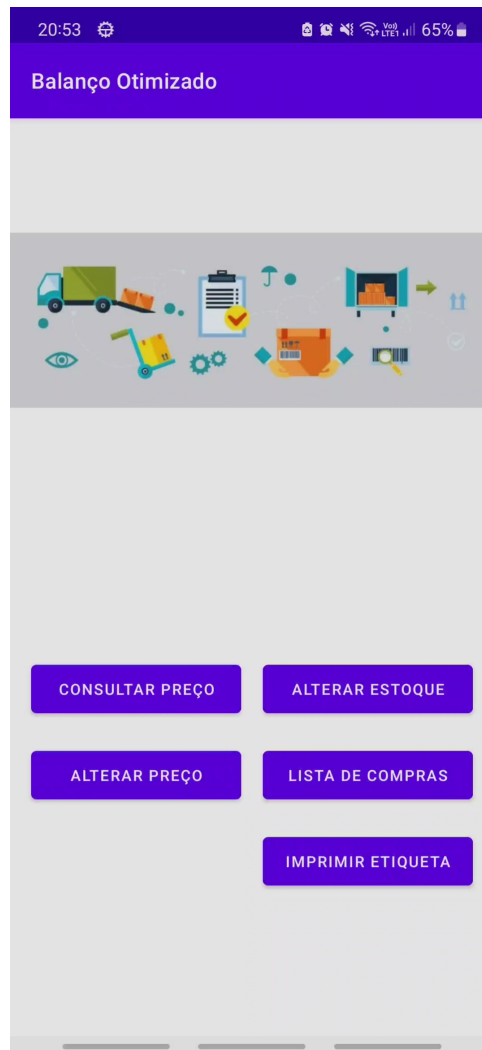


Figura 4.5: Tela Inicial do aplicativo

4.6 Tela de Login

Para realizar a autenticação e poder ter acessos as funcionalidades de alterações, basta inserir o usuário e senha cadastrado, como demonstrado na Figura 4.6.

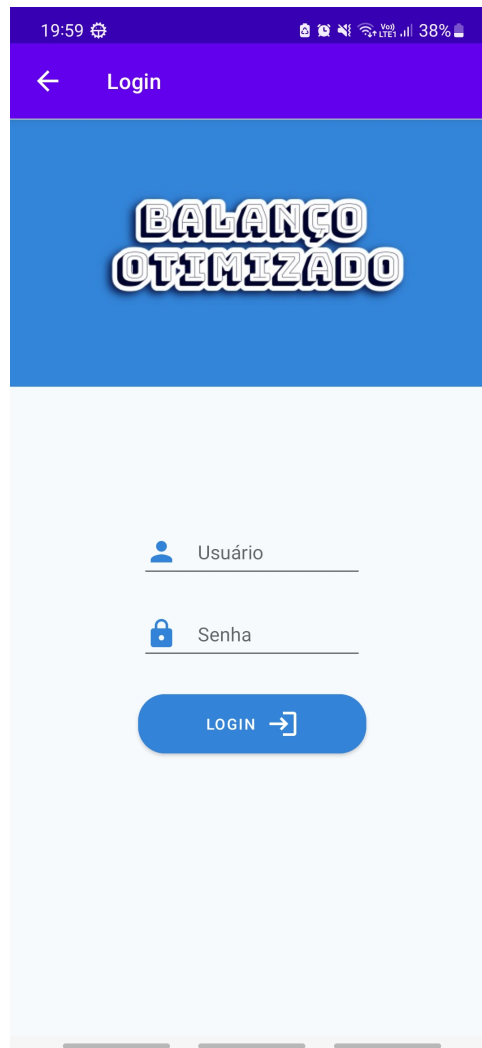


Figura 4.6: Tela de Login

4.7 Alterar Preço

A página de alteração de preço é composta pelo campos de pesquisa como na Figura 4.7, podendo ser pelo nome ou marca do produto, código ou lendo o seu código de barras utilizando a câmera do celular, conforme a Figura 4.4.

Após selecionar o produto desejado, aparecerá o preço atual e o campo para inserir o novo preço desejado, Figura 4.8.



Figura 4.7: Tela de Alteração de Preço

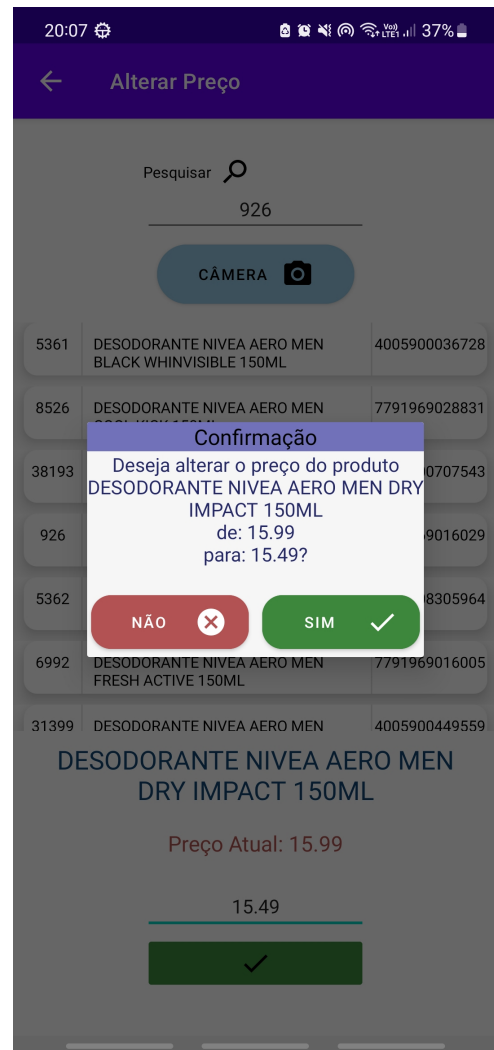


Figura 4.8: Janela de confirmação de alteração de preço

4.8 Alterar Estoque

A página de alteração de estoque (Figura 4.9) funciona de forma similar a de alteração de preço, primeiro é realizado a busca do produto via código de barras, nome ou código. Encontrado o produto, aparecerá o estoque atual do mesmo e o campo para inserir o novo estoque, Figura 4.10.

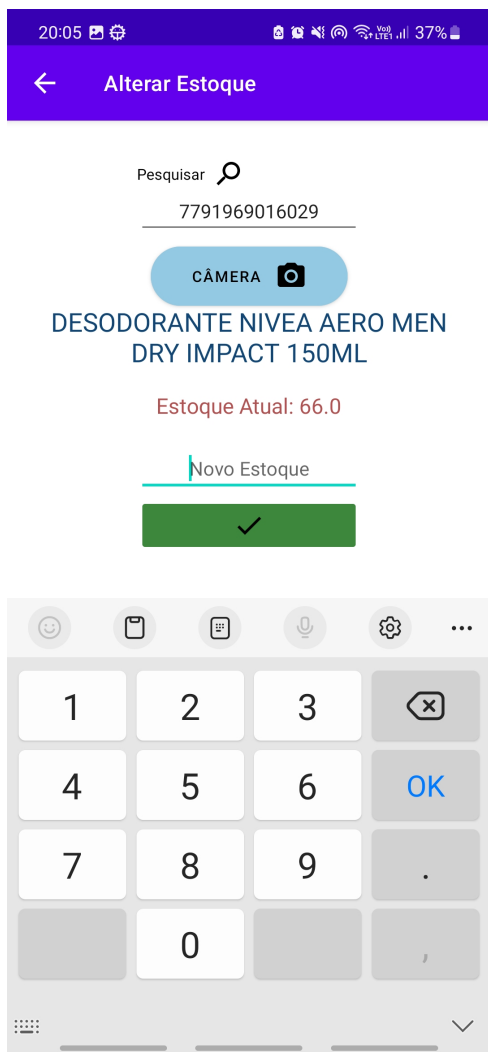


Figura 4.9: Tela de Alteração de Estoque



Figura 4.10: Janela de confirmação de alteração de estoque

4.9 Consultar Preço

A tela de consulta de preço (Figura 4.11) apresenta o campo para busca do produto podendo ser através do nome, marca, código ou código de barras utilizando a câmera.

Selecionado o produto, na parte inferior apresentará suas respectivas informações de estoque, preço de venda e preço de promoção e atacado, se houver.

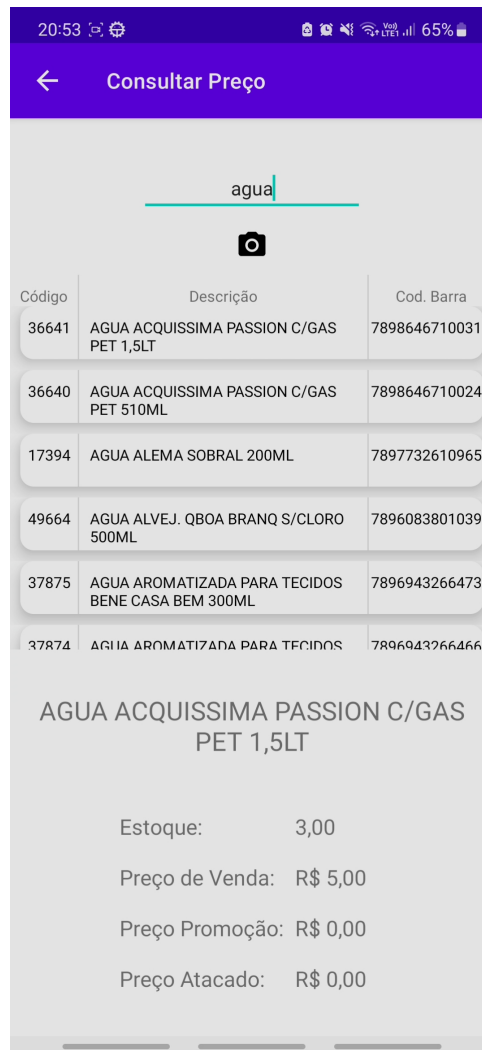


Figura 4.11: Tela de Consulta de Preço

4.10 Lista de Compras

Na tela de lista de compras (Figura 4.12) o usuário pode criar listas de compras com base na sua necessidade, podendo ser por fornecedores, por gênero alimentício, etc. Basta clicar no botão *Nova lista*, bem como visualizar as listas já criadas.

Para criação de uma nova lista, aparecerá a janela, conforme mostra na figura 4.13, contendo o campo *Nome* para informar o nome da nova lista a ser criada juntamente com os botões *Cancelar* e *Salvar*.

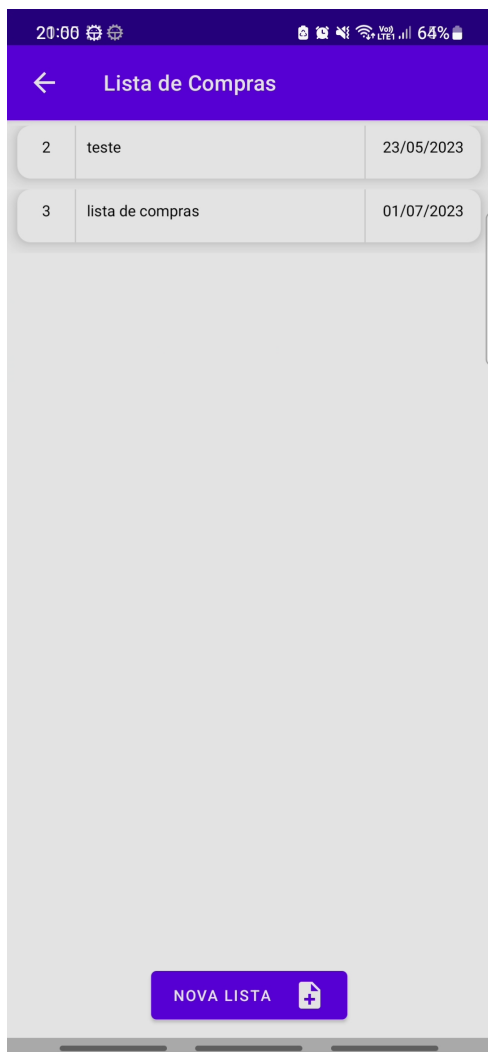


Figura 4.12: Tela de Lista de Compras



Figura 4.13: Janela de confirmação de nova lista

Para visualização dos itens de uma lista de compras, basta clicar no nome da lista escolhida que o usuário será direcionado para a página na qual irá conter os produtos já adicionados, como na Figura 4.14. Há também, os botões *Excluir Lista*, *Enviar Lista* e *Adicionar item*.



Figura 4.14: Tela dos itens de uma lista de compras

Ao clicar em excluir lista, abrirá a janela (Figura 4.15) para confirmar a decisão.

Ao clicar em enviar lista, abrirá a janela (Figura 4.16) para informar o assunto e *email*, respectivamente, juntamente com os botões *Cancelar* e *Enviar*.

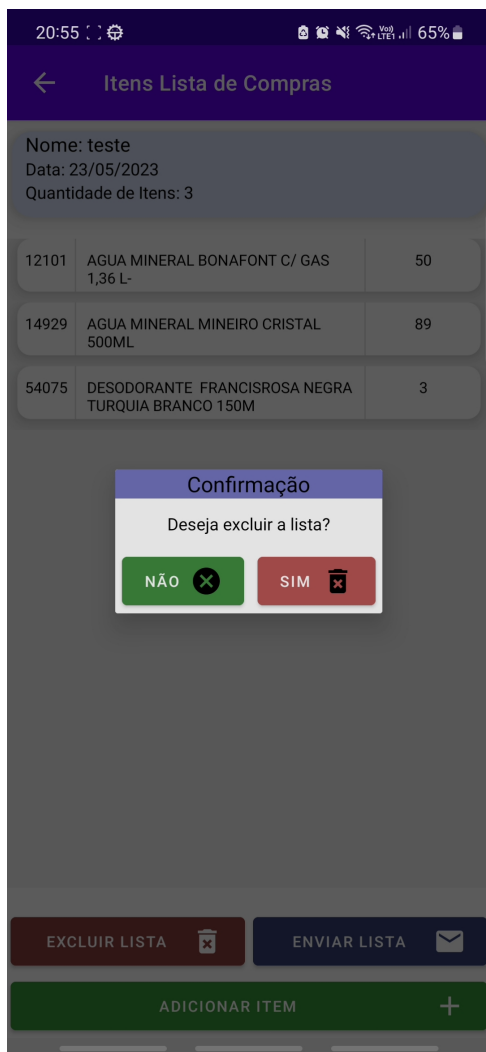


Figura 4.15: Janela de exclusão de Lista de Compras

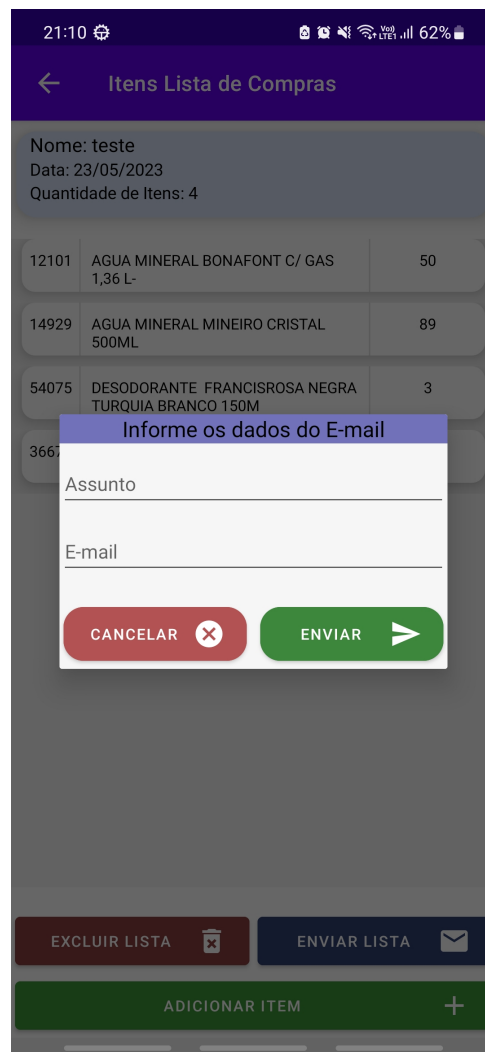


Figura 4.16: Janela de envio de lista de compras

Ao clicar em adicionar item, o usuário é direcionado para a tela de lançamento de itens (Figura 4.17). Para adicionar o produto, basta pesquisá-lo no respectivo campo utilizando o nome, marca, código ou código de barras do produto.

Ao localizar a mercadoria pretendida, abrirá a janela (Figura 4.18) para informar a quantidade que se deseja adicionar dessa mercadoria na lista de compras, podendo digitar o número ou adicionando no botão *+* ou diminuir no botão *-*. Após a escolha de quantidade basta apertar no botão *Salvar* para adicioná-lo na lista ou no botão *Cancelar* para não adicionar na lista.

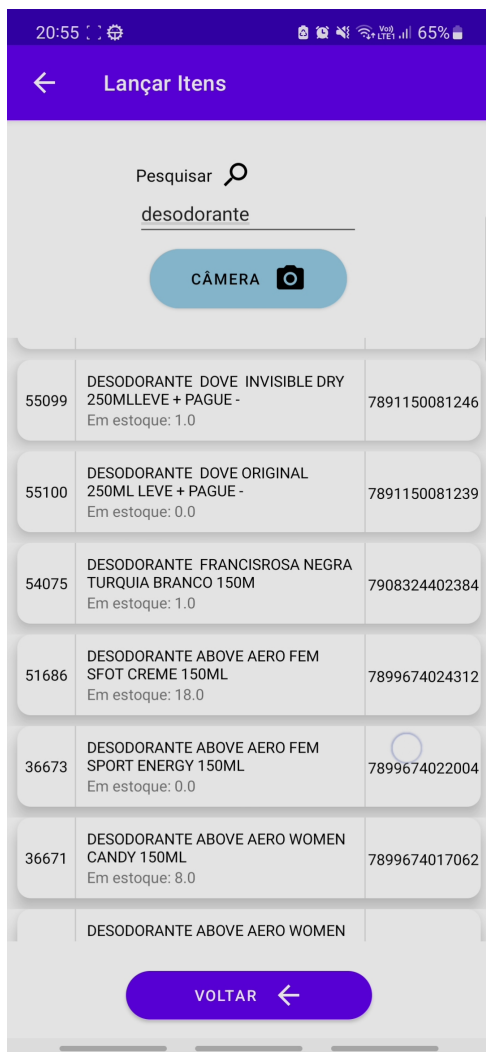


Figura 4.17: Tela de Adição de itens a lista de compra

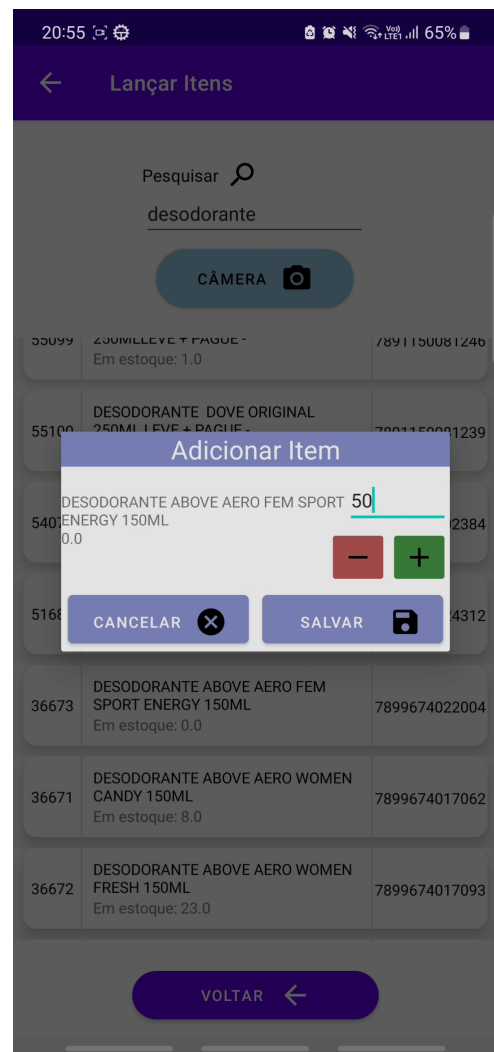


Figura 4.18: Janela de quantidade dos itens

4.11 Imprimir Etiqueta de Gôndola

A tela de impressão de etiqueta (Figura 4.19) é composta pelo campo de busca de produto, podendo ser pelo nome, marca, código ou código de barras. Após selecionar o produto desejado irá aparecer uma pré-visualização da etiqueta e o botão Imprimir, apertando-o irá imprimir e poderá se retirado na impressora, como mostra na Figura 4.20.

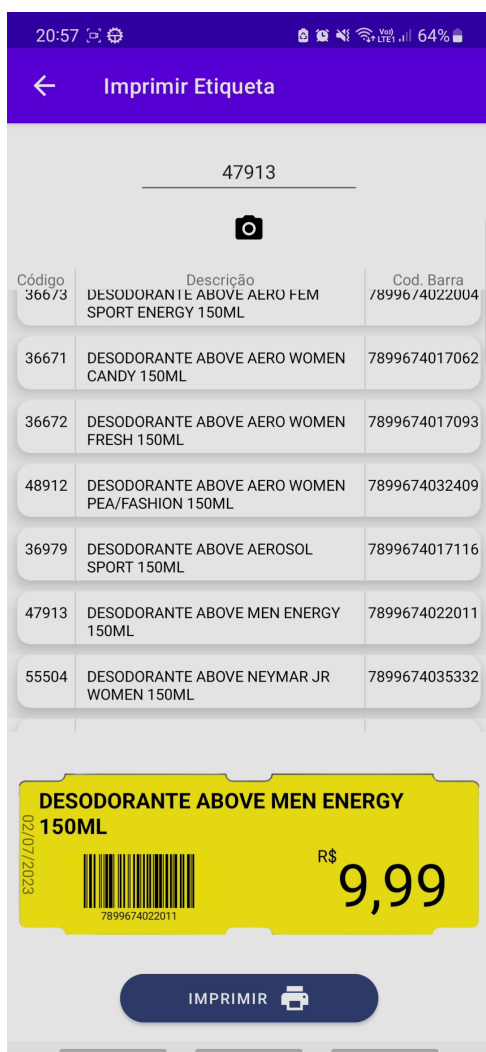


Figura 4.19: Tela de Impressão de Etiqueta



Figura 4.20: Etiqueta impressa

4.12 Validação

No intuito de assegurar que o aplicativo atendesse às necessidades de uma ferramenta de melhoria no controle e planejamento de estoque mercantil, o aplicativo foi submetido a testes com 41 supermercados diferentes para avaliação e sugestões de melhoria sob a supervisão de gerentes que tinham responsabilidade no manuseio, visto que o aplicativo estava funcionando em banco de produção do comércio.

O *software* ficou na fase de teste por cerca de 3 semanas e após essa fase, foi realizado um questionário para avaliação de cada item presente no aplicativo e ao final, um campo livre para sugestão, crítica e observação. O formulário foi enviado aos 41 participantes do teste, porém houve apenas 36 respostas. Após analisar os *feedbacks* recebidos, ficou evidente que o aplicativo necessitava de um login e que estava com uma falha de segurança com a falta desse componente, com isso, ficou decidido que somente é necessário ter autenticação com login e senha em duas funcionalidades do aplicativo, que são aquelas que detêm o poder de fazer alteração em banco de dados, sendo elas: alteração de preço e alteração de estoque. Ademais, foram sugeridos ajustes em tamanho de fonte de letras e melhorias de navegação entre telas.

O resultado do formulário pergunta sobre as funcionalidades, *layout* e fonte com escala de 1 a 4, onde 1 é muito ruim e 4 é muito bom (mostrado abaixo com o título: *Validação - App - Balanço Otimizado*) foi bastante positivo tendo em vista o funcionamento correto das funcionalidades implementadas, na grande maioria não apresentou lentidão, travamentos tampouco dados inconsistentes. Segundo os usuários que participaram dos testes, o aplicativo se mostrou útil.

Validação - App - Balanço Otimizado

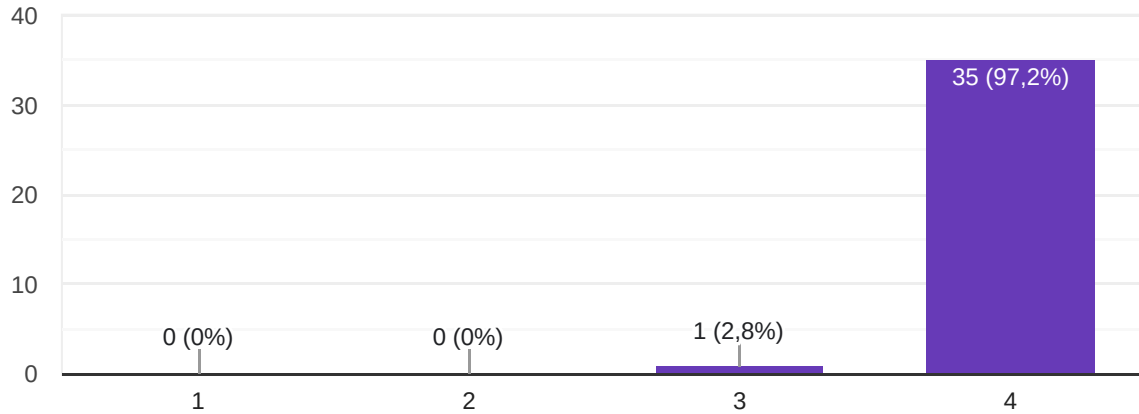
36 respostas

[Publicar análise](#)

Como foi a operação da funcionalidade de leitura de código de barras?

[Copiar](#)

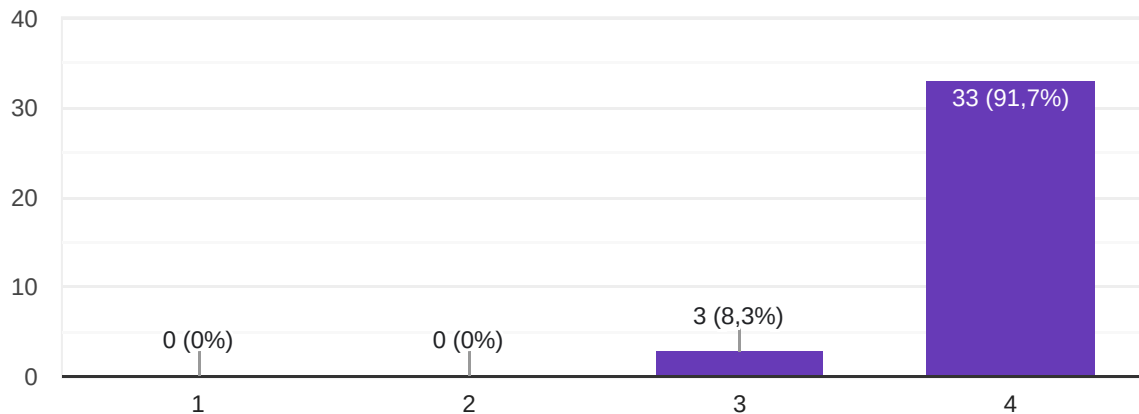
36 respostas



Como foi o funcionamento da tela de alteração de preço?

[Copiar](#)

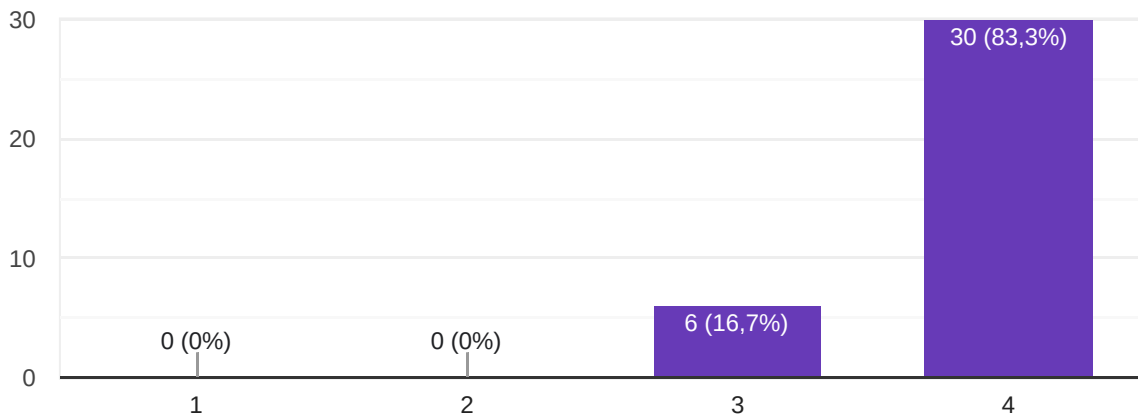
36 respostas



Como foi o funcionamento da tela de alteração de estoque?

 Copiar

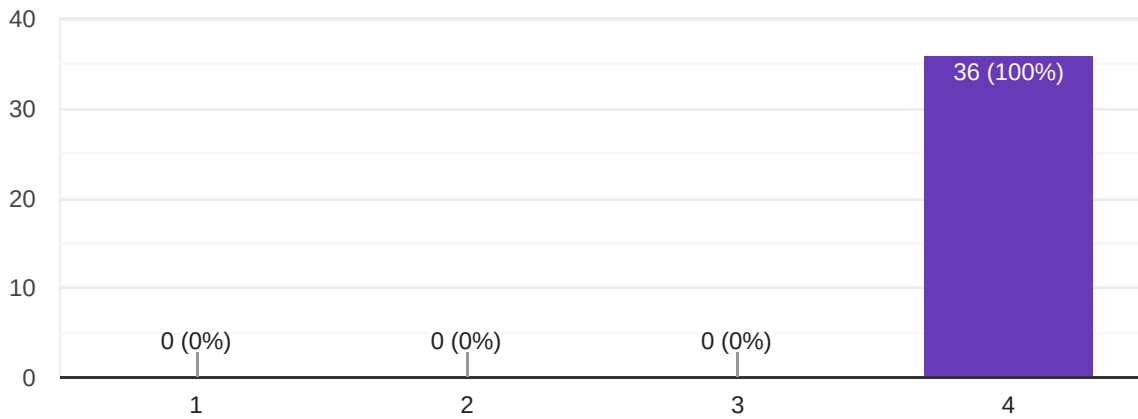
36 respostas



Como foi o funcionamento da tela de consulta de preço?

 Copiar

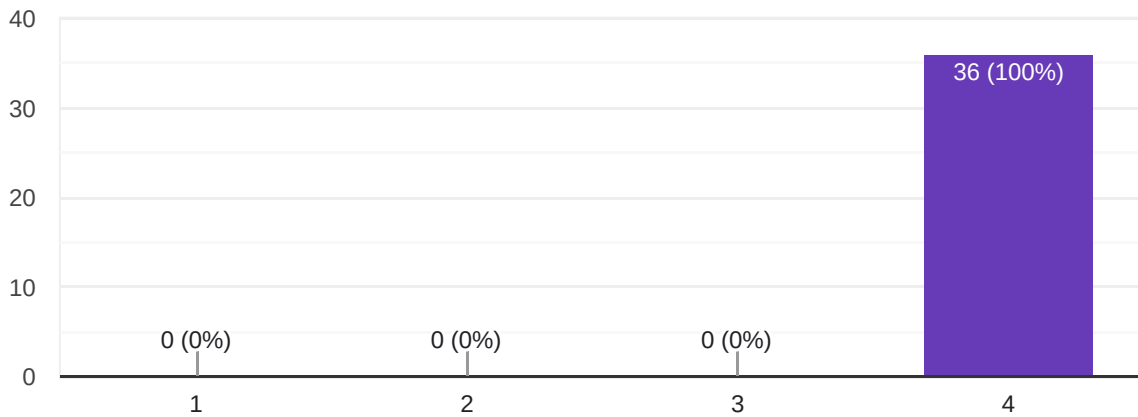
36 respostas



Como foi o funcionamento da tela de impressão de etiqueta?

 Copiar

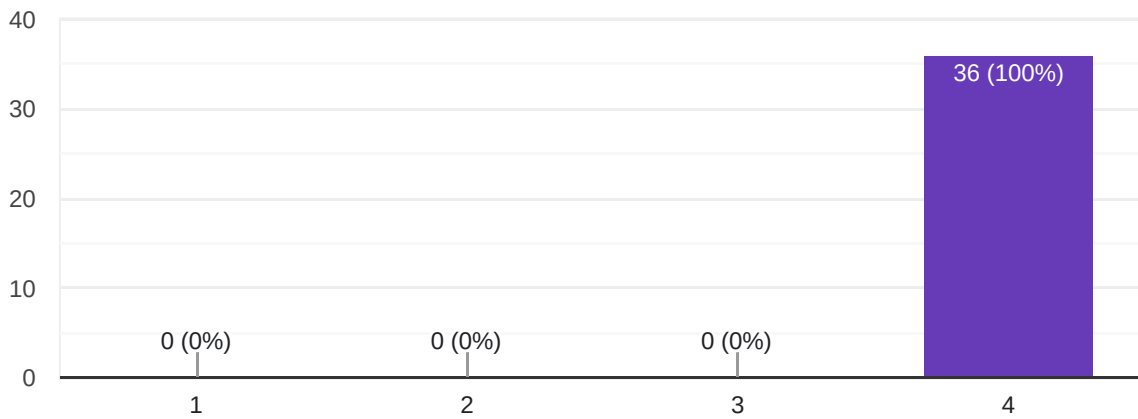
36 respostas



Como foi o funcionamento da lista de compras?

 Copiar

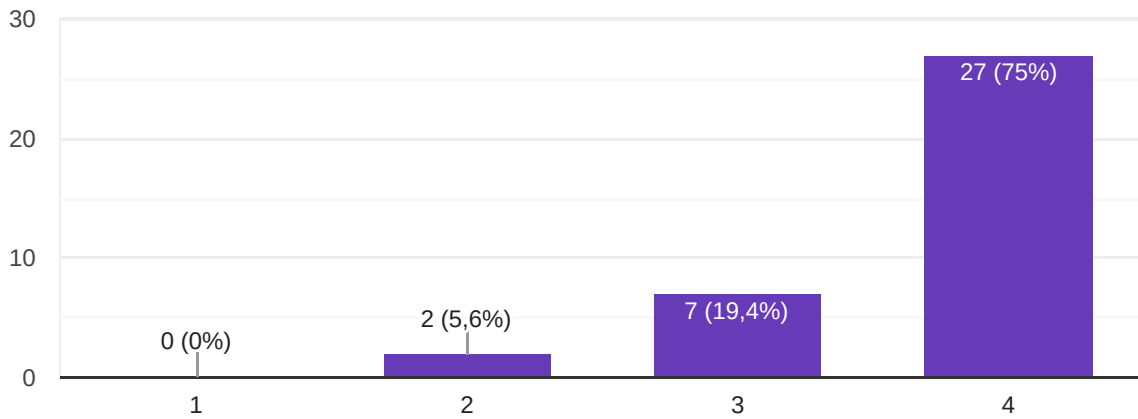
36 respostas



Tamanho da fonte

 Copiar

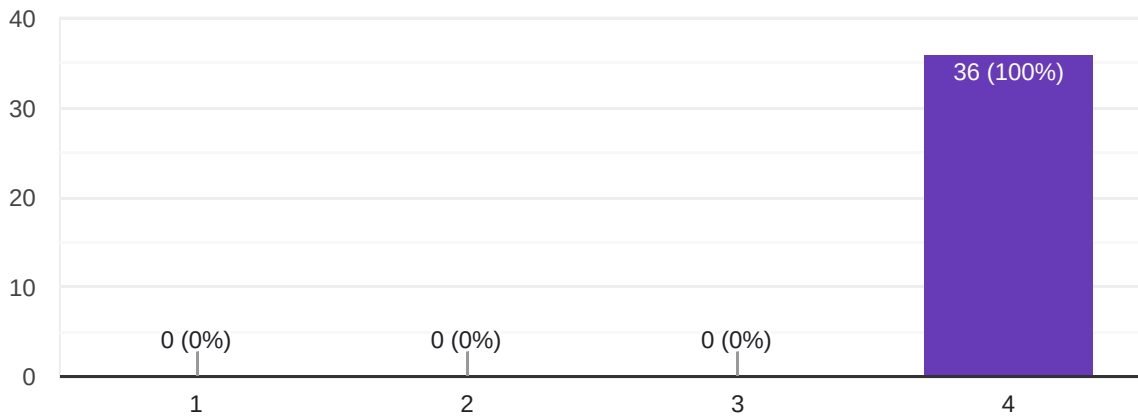
36 respostas



Informações relevantes nas telas?

 Copiar

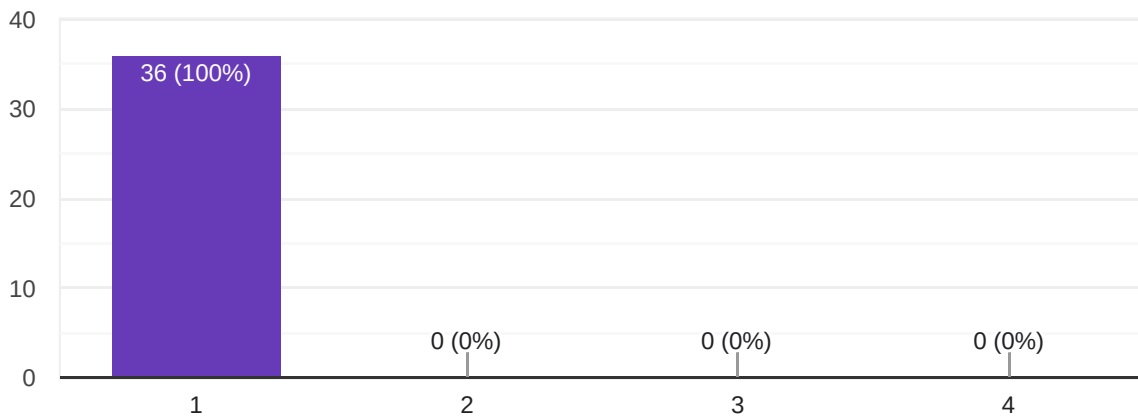
36 respostas



Informações irrelevantes nas telas?

 Copiar

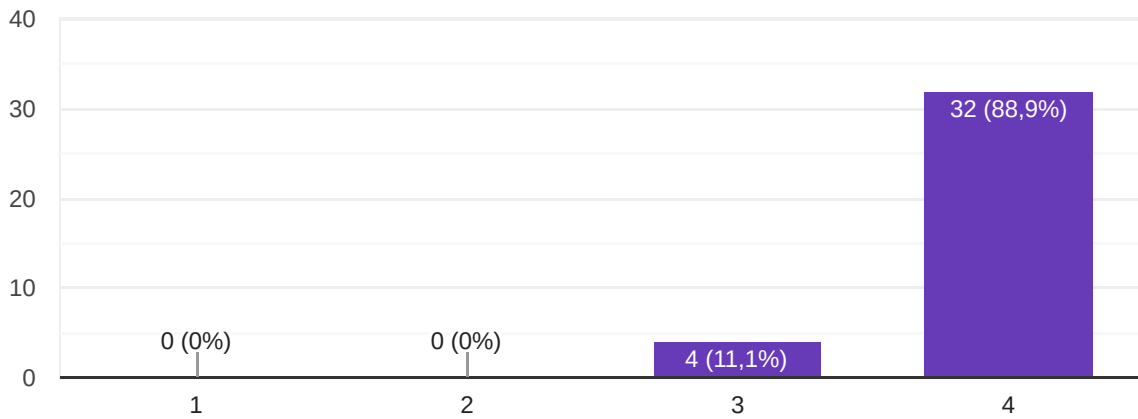
36 respostas



Layout bom e intuitivo?

 Copiar

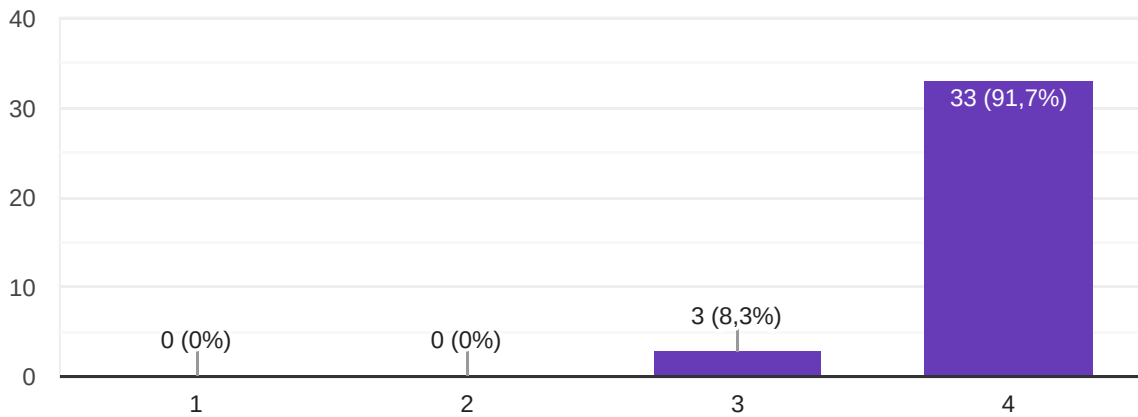
36 respostas



Organização geral das telas?

 Copiar

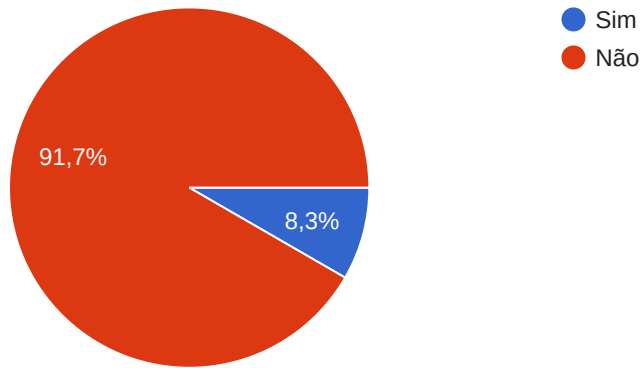
36 respostas



Apresentou lentidão?

 Copiar

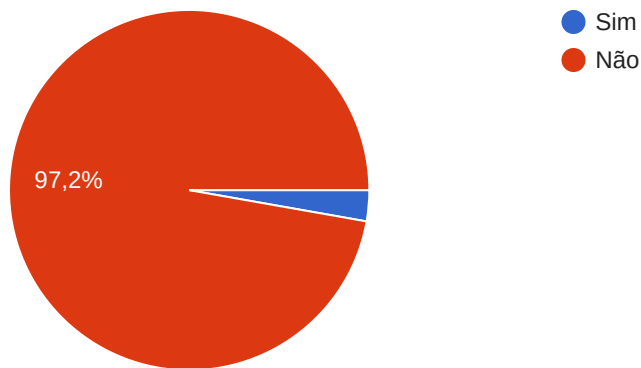
36 respostas



Apresentou travamentos?

 Copiar

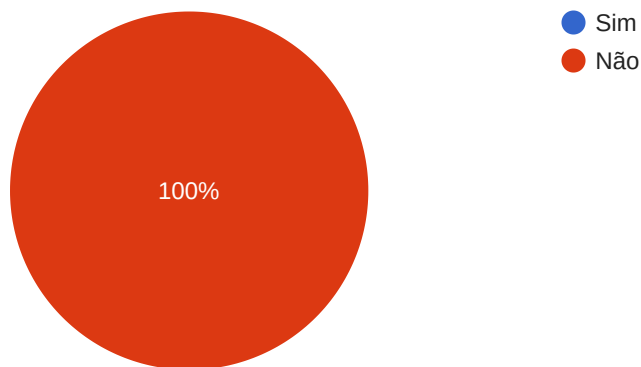
36 respostas



Apresentou algum dado inconsistente?

 Copiar

36 respostas



Comentário, crítica ou sugestão

13 respostas

alterar preço e estoque precisava ser obrigatório informar usuário

app precisa de login

alterar estoque e preço está sem segurança

Fonte muito pequena

depois de mudar o preço do produto podia ter a opção de mandar imprimir uma etiqueta sem precisar trocar de tela

o app ajudou muito a fazer o balanço do estoque

aumentar a letra na tela de consultar produto

aplicativo muito bom

ter como enviar um produto pra lista de sugestão direto da consulta de produto

perfeito, não tenho mais preço errado nas gôndolas

não posso instalar o app no celular dos reposidores porque eles poderiam alterar os preços dos produtos

A lista de compras facilitou muito minha vida, consigo andar pelo mercado marcando o que preciso comprar

Conseguir enviar impressão de etiqueta sem precisar de um computador foi muito útil

Este conteúdo não foi criado nem aprovado pelo Google. [Denunciar abuso](#) - [Termos de Serviço](#) - [Política de Privacidade](#)

Google Formulários



O próximo capítulo trata das conclusões deste trabalho.

Capítulo 5

Conclusão

Neste trabalho de conclusão do curso de Ciência da Computação, foi desenvolvido um aplicativo para *smartphones* que implementa uma praticidade no controle e planejamento de balanço de estoque de supermercados, configurado como uma ferramenta auxiliadora que, segundo avaliações da maioria dos usuários de teste, mostrou-se útil e bastante prática para árdua tarefa em questão.

O gerenciamento de mercadorias é um trabalho essencial e de suma importância para o comércio em geral. No ramo de supermercados esse trabalho se torna ainda mais fundamental, pois em um ambiente como esse, há uma grande quantidade de produtos e marcas, podendo ultrapassar milhares de itens. Gerir toda essa mercadoria requer um bom planejamento, visto que a falta de produtos para venda bem como produtos vencidos na prateleira mancham a imagem do estabelecimento. Com isso, a ideia inicial de fazer um aplicativo de balanço de mercadorias de forma mais automatizada e muito rápida se tornou um *software* de grande relevância que complementa um *software* desktop do mercado, segundo os usuários que testaram o protótipo, com mais funcionalidades que completaram o aplicativo, como a funcionalidade de reconhecimento do produto por meio do código de barras. Ademais, consulta de estoque, lista de compras enviando por e-mail, impressão de etiqueta, seja por estar com valor divergente ou porque houve algum dano ou perda. Diante do protótipo desenvolvido, testado e avaliado de maneira muito positiva, tivemos o reconhecimento de donos e gerentes de vários supermercados, bem como da diretoria da empresa que nos deu essa oportunidade.

Um ponto importante, para que o aplicativo funcionasse de maneira mais fidedigna foi a implementação da API de comunicação que tivemos liberação de acesso por parte da empresa responsável pelo *software* desktop de supermercados. Porém, enfrentamos bastante dificuldade nessa implementação, tendo em vista o não conhecimento na área de integração entre sistemas e o funcionamento de uma API rest. Entretanto, desafios como

este foram vencidos com sucesso e foi possível acessar os registros do banco de dados dos supermercados.

Por conseguinte, é possível afirmar que os objetivos propostos no início do projeto foram plenamente alcançados com bons resultados, segundo *feedbacks* dos usuários que testaram o aplicativo.

Para trabalhos futuros, sugere-se a implementação do aplicativo para o sistema operacional iOS, porém para tal tarefa é necessário dispositivos da Apple para ser realizada. Outrossim, faz-se interessante a funcionalidade de realizar venda pelo aplicativo, bem como fazer o cadastro de um produto novo.

Referências Bibliográficas

- [1] Github: *Documentação*. <https://git-scm.com/doc>. Acesso em 06/03/2023. 21.
- [2] Google: *Android: Open Source Software*. <https://magsilva.pro.br/apps/wiki/oss/Android>. Acesso em 12/04/2023. 14.
- [3] Google: *Android para desenvolvedores - Arquitetura | Desenvolvimento*. <https://developer.android.com/jetpack/guide?hl=pt-br>. Acesso em 16/05/2023. 15.
- [4] Google: *Arquitetura Android*. <https://developer.android.com/guide/platform?hl=pt-br>. Acesso em 06/05/2023. vii, 14, 15.
- [5] Google: *Documentação Android - SQLite*. <https://developer.android.com/training/data-storage/sqlite?hl=pt-br>. Acesso em 24/03/2023. 21.
- [6] Google: *Documentação Android Studio*. <https://developer.android.com/studio/intro?hl=pt-br>. Acesso em 09/05/2023. 19.
- [7] Microsoft: *Documentação C#*. <https://learn.microsoft.com/pt-br/dotnet/csharp/tour-of-csharp/>. Acesso em 01/05/2023. 18.
- [8] Microsoft: *Visual Studio Code Documentation*. <https://code.visualstudio.com/docs>. Acesso em 13/05/2023. 20.
- [9] Red hat: *Documentação*. <https://www.redhat.com/pt-br/topics/api/what-is-a-rest-api>. Acesso em 01/04/2023. 19.
- [10] Sebrae: *Regimes tributários*. <https://sebraeng.com.br/diferencas-entre-regimes-tributarios/>. Acesso em 02/05/2023. 7.
- [11] Secretaria da fazenda: *Consulta N° 004/2012 - SEF - Legislação*. https://legislacao.sef.sc.gov.br/html/consultas/2012/con_12_004.htm#. Acesso em 01/03/2023. 5.

- [12] Universidade federal de pernambuco: *Arquitetura de Software*. https://www.cin.ufpe.br/~gta/rup-vc/core.base_rup/guidances/concepts/software_architecture_4269A354.html . Acesso em 25/04/2023. 23.
- [13] Mike Owens, Grant Allen. *The Definitive Guide to SQLite*. Apress, 2010. 21.
- [14] Thomas Connolly, Carolyn Begg. *Database Systems: A Practical Approach to Design, Implementation, and Management*. Pearson, 2014. 20.
- [15] Codd, Edgar F. A relational model of data for large shared data banks. *Communications of the ACM*, 1970. 20.
- [16] Date, Christopher J. *An Introduction to Database Systems*. Addison Wesley, 2004. 20.
- [17] Roger Pressman, Bruce Maxim. *Engenharia de software - Uma Abordagem Profissional*. AMGH, 2016. 13.
- [18] Dias, Marco Aurelio Pereira. *Administração de Materiais - Uma Abordagem Logística*. ATLAS, 2015. 5.
- [19] Daniela Barreiro Claro, João Bosco Manguieira Sobral. *Programação em JAVA*. Pearson, 2008. 17.
- [20] Sommerville. *Engenharia de Software*. Pearson, 2019. 13.