

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO  
RIO GRANDE DO SUL  
CAMPUS CANOAS  
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE  
SISTEMAS

EDUARDO ZANCHETTI SCHOLZ

**Seller's Book: um aplicativo móvel para gerência de  
estoque e vendas de vendedores autônomos**

Canoas, 12 de agosto de 2022.

EDUARDO ZANCHETTI SCHOLZ

**Seller's Book: um aplicativo móvel para gerência de  
estoque e vendas de vendedores autônomos**

Trabalho de Conclusão de Curso apresentada  
como requisito parcial para obtenção do grau de  
Tecnólogo em Análise e Desenvolvimento de  
Sistemas pelo Instituto Federal de Educação,  
Ciência e Tecnologia do Rio Grande do Sul –  
Campus Canoas.

Prof. Leonardo Filipe Batista S. de Carvalho

Canoas, 12 de agosto de 2022.



**Ministério da Educação**  
**Secretaria de Educação Profissional, Científica e Tecnológica**  
**Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul**  
**Campus Canoas**

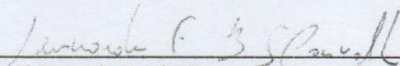
## **ATA DE DEFESA PÚBLICA DO TRABALHO DE CONCLUSÃO DE CURSO**

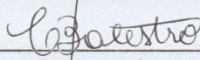
Aos 02 dias do mês de agosto de 2022, às 14:00 horas, em sessão pública na sala pública virtual do Google Meet (<https://meet.google.com/zck-zykw-ovf>) do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul, Campus Canoas, na presença da Banca Examinadora presidida pelo(a) Professor(a):

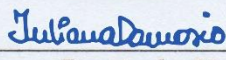
Dr. Leonardo Filipe Batista Silva de Carvalho e composta pelos examinadores:

1. Prof. Dra. Carla Odete Balestro Silva;
2. Prof. Dra. Juliana Damasio Oliveira;

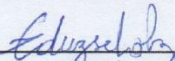
o(a) aluno(a) EDUARDO ZANCHETTI SCHOLZ apresentou o Trabalho de Conclusão de Curso intitulado: “Aplicativo móvel para gerência de estoque e vendas de vendedores autônomos” como requisito curricular indispensável para a integralização do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas. Após reunião em sessão reservada, a Banca Examinadora deliberou e decidiu pela APROVAÇÃO do referido trabalho, divulgando o resultado formalmente ao aluno e demais presentes e eu, na qualidade de Presidente da Banca, lavrei a presente ata que será assinada por mim, pelos demais examinadores e pelo aluno.

  
\_\_\_\_\_  
Presidente da Banca Examinadora

  
\_\_\_\_\_  
Examinador 01

  
\_\_\_\_\_  
Examinador 02

\_\_\_\_\_  
Examinador 03

  
\_\_\_\_\_  
Aluno

## RESUMO

Vendedores autônomos gastam muito tempo e energia para gerenciar seus estoques e vendas, utilizando de cadernos, agendas ou memória para guardar os dados de produtos, clientes e pagamentos. Este é um processo lento e cansativo, em que por vezes o vendedor perde ou esquece alguma de suas anotações. Assim, este trabalho se propõe a desenvolver uma opção melhor para estes profissionais, um aplicativo de controle de estoque e vendas, que busca tornar este processo mais fácil, rápido e intuitivo. Ao longo deste trabalho, fez-se um estudo da situação destes vendedores e se apresenta a proposta de um aplicativo para substituir as agendas tradicionais. Para isso, fez-se também um estudo sobre UX (*User Experience*), um termo utilizado para definir como o usuário se sente quanto a um produto, neste caso um sistema, a fim de tornar o processo de substituição mais satisfatório para os vendedores. Por este motivo, durante o trabalho, buscou-se também a experiência de pessoas do ramo para apoiar o desenvolvimento de um aplicativo adequado para esses vendedores, o Seller's Book. Aplicativo este que foi desenvolvido para Android, a fim de alcançar maior portabilidade, e utilizou a IDE (*Integrated Development Environment*) Android Studio.

**Palavras-Chave:** Aplicativo. Gerenciamento de estoque. Gerenciamento de vendas. UX. Vendedores autônomos.

## ABSTRACT

Self-employed sellers spend a lot of time and energy managing their inventories and sales, using notebooks, diaries or memory to store data of products, customer and payments. This is a slow and tiring process, in which, sometimes, the seller may lose or forget some of his or hers notes. Therefore, this work proposes to develop a better option for these professionals, a stock and sales control application, which seeks to make this an easier, faster and more intuitive process. This work conducts a study of the situation of these sellers and a proposes an application to replace the traditional agendas. For this, UX (User Experience), a term used to define how the user feels about a product, in this case a system, was also studied, in order to make the replacement process more satisfactory for sellers. For this reason, during the work, the experience of people in the field was also sought out in order to support the development of a suitable application for them, the Seller's Book. The application was developed for Android, seeking better portability, using the IDE (Integrated Development Environment) Android Studio.

**Keywords:** Application. Inventory management. Self-employed sellers. Sales Management. UX.

## LISTA DE FIGURAS

Figura 1: Diagrama de casos de uso para o sistema simplificado de um restaurante.....	16
Figura 2: exemplo diagrama entidade relacionamento.....	16
Figura 3: Gráfico de sistemas operacionais mais utilizados.....	17
Figura 4: Interface gráfica do usuário do Excel.....	21
Figura 5 Tela inicial do Super Revendedores.....	22
Figura 6: Captura da aba de ferramentas do Kyte.....	24
Figura 7: Diagrama de casos de uso.....	29
Figura 8: Diagrama Entidade-Relacionamento.....	30
Figura 9: Tela Principal do aplicativo.....	31
Figura 10: Imagens do fragmento de produtos.....	32
Figura 11: Fragmento de Cliente.....	34
Figura 12: Fragmento de vendas.....	35
Figura 13: Imagem das <i>Activitys</i> de criação de venda.....	36
Figura 14: Fragmento de início.....	36
Figura 15: Idade dos respondentes do questionário de satisfação.....	39
Figura 16: Gênero dos respondentes.....	40
Figura 17: Tipos de produtos vendidos pelos respondentes.....	40
Figura 18: Resultados da primeira pergunta sobre UX.....	41
Figura 19: Gráfico relativo a segunda pergunta sobre UX.....	42
Figura 20: Gráfico relativo à terceira pergunta sobre UX.....	42
Figura 21: Gráfico relativo a quarta pergunta sobre UX.....	43
Figura 22: Gráfico relativo à quinta pergunta sobre UX.....	43
Figura 23: Mapeamento das zonas de mais fácil toque da tela.....	44
Figura 24: Gráfico relativo à questão de gosto do aplicativo.....	44
Figura 25: Gráfico relativo à questão sobre a substituição dos cadernos e agendas.....	45
Figura 26: Gráfico relativo à questão de recomendação.....	45

## LISTA DE QUADROS

Quadro 1: 10 heurísticas da usabilidade.....	20
Quadro 2: Comparação dos aplicativos.....	25
Quadro 3: Perguntas sobre UX do questionário de satisfação.....	38

## LISTA DE ABREVIATURAS E SIGLAS

CRUD	<i>Create Read Update Delete</i>
ER	Entidade Relacionamento
IDE	<i>Integrated Development Enviroment</i>
POO	Programação Orientada a Objetos
SO	Sistema Operacional
UI	<i>User Interface</i>
UML	<i>Unified Modeling Language</i>
UX	<i>User Experience</i>



## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>11</b>
1.1	MOTIVAÇÃO.....	12
1.2	OBJETIVOS .....	12
1.3	OBJETIVO GERAL.....	12
1.4	OBJETIVOS ESPECÍFICOS .....	12
1.5	ORGANIZAÇÃO DO TEXTO.....	13
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA .....</b>	<b>14</b>
2.1	PROGRAMAÇÃO ORIENTADAS A OBJETOS (POO).....	14
2.2	UNIFIED MODELING LANGUAGE (UML).....	14
2.3	ANDROID.....	17
2.4	BANCO DE DADOS .....	18
2.5	USER EXPERIENCE (UX).....	18
2.6	TESTE DE SOFTWARE .....	20
<b>3</b>	<b>ESTADO DA ARTE.....</b>	<b>21</b>
3.1	APLICAÇÕES.....	21
3.1.1.1	EXCEL .....	21
3.1.2	SUPER REVENDEDORES .....	22
3.1.3	KYTE.....	23
3.1.4	ANÁLISE COMPARATIVA DOS SISTEMAS .....	24
<b>4</b>	<b>METODOLOGIA.....</b>	<b>26</b>
<b>5</b>	<b>DESENVOLVIMENTO.....</b>	<b>28</b>
5.1	MODELAGEM .....	28
5.2	IMPLEMENTAÇÃO.....	30
5.3	TESTES.....	37
<b>6</b>	<b>RESULTADOS .....</b>	<b>38</b>
6.1	ELABORAÇÃO DO QUESTIONÁRIO .....	38
6.2	ANÁLISE DE RESULTADOS.....	39
<b>7</b>	<b>CONCLUSÕES.....</b>	<b>47</b>
7.1	TRABALHOS FUTUROS .....	48
	<b>REFERÊNCIAS.....</b>	<b>49</b>
	<b>APÊNDICE A – LEVANTAMENTO DE REQUISITOS .....</b>	<b>55</b>
	<b>APÊNDICE B – RESULTADOS DO LEVANTAMENTO DE REQUISITOS .....</b>	<b>58</b>
	<b>APÊNDICE C – REQUISITOS DO SISTEMA.....</b>	<b>61</b>

<b>APÊNDICE D – ESPECIFICAÇÕES DE CASOS DE USO.....</b>	<b>62</b>
<b>APÊNDICE E – DIAGRAMA DE CLASSES .....</b>	<b>68</b>
<b>APÊNDICE F – FORMULARIO DE SATISFAÇÃO DO APLICATIVO .....</b>	<b>69</b>

## 1 INTRODUÇÃO

De acordo com Lima (2021), o número de trabalhadores autônomos no Brasil superou 24,8 milhões de pessoas no segundo trimestre de 2021. Esse número corresponde a 28,3% dos indivíduos ativos no mercado de trabalho brasileiro. Dentre esses trabalhadores, existem aqueles que atuam de forma autônoma na realização de vendas e atividades semelhantes. Para esses trabalhadores, a gestão de vendas e de seu estoque de produtos pode ser um desafio. Isto, é o que foi relatado por entrevista à uma vendedora autônoma, que afirma que muitos desses profissionais ainda utilizam agendas físicas ou outros meios para guardar suas informações na forma de anotações, o que oferece pouca flexibilidade e facilidade para registrar e localizar informações.

Assim, ainda que esses indivíduos dependam de suas anotações para realizar suas vendas, essa pode não ser a forma mais prática, segura ou mesmo eficiente de guardar esses dados. De fato, Costa (2013) diz ser inviável o uso desses recursos quando se trata de gerenciar grande quantidade de informações. Por isso, propomos aqui o uso de um aplicativo criado para esse perfil de profissional, trabalhadores autônomos (vendedores) que poderiam economizar tempo e espaço físico gasto com suas anotações, aplicativo esse que será chamado *Seller's Book*.

Construir um aplicativo para auxiliar esse público requer conhecer as particularidades de tais profissionais e as tecnologias que mais se adequam às suas necessidades. A esse respeito, uma matéria do site G1 (2008), aponta que “muitos brasileiros utilizam o celular para aumentar a renda”. Afirmação que vai ao encontro de Prado (2018), para quem a tecnologia gera muitas mudanças no papel do vendedor. Por essas afirmações, e pensando na melhor portabilidade do aplicativo, o *Seller's Book* será destinado a celulares.

Para propor essa solução, essa pesquisa assume uma natureza aplicada e de abordagem quali-quantitativa. Antes do início do desenvolvimento foi feito uma pesquisa com o público-alvo, com a finalidade de identificar os requisitos do sistema. Essa pesquisa será mais bem detalhada durante o capítulo de desenvolvimento. O aplicativo *Seller's Book* foi implementado seguindo os passos da Engenharia de *Software*, utilizando a *Unified Modeling Language* (UML), uma linguagem para a modelagem de sistemas, e a linguagem de programação Java para sua implementação.

## **1.1 MOTIVAÇÃO**

Quando se trabalha com vendas, inevitavelmente, é preciso manter um estoque para seus produtos, bem como controlar a gerência desse estoque em relação às vendas efetivadas. De acordo com Hervé (2017) a falta de organização do estoque pode fazer com que o empresário perca vendas ou superlote almoxarifados. Para profissionais autônomos isso significa capital parado desnecessariamente.

Comumente, estes profissionais usam da própria memória, papéis ou diversas agendas para anotar os detalhes de suas vendas (COSTA, 2013). Situação que, por vezes, tem como um dos motivos o custo dos sistemas disponíveis no mercado que oferecem esse tipo de auxílio, os quais, muitas vezes, não são adequados ao orçamento disponível para esses vendedores. Outras vezes, esses aplicativos podem limitar seus recursos em planos gratuitos, ou mesmo apresentar complexidade superior com a qual os profissionais podem lidar (SMARTPOS, 2021).

Logo, a proposta do aplicativo *Seller's Book*, visa preencher esta lacuna e oferecer dentro de suas funcionalidades a possibilidade de gerenciar estoques, gerenciar vendas e gerenciar de pagamentos e entregas.

## **1.2 OBJETIVOS**

Essa seção é destinada a apresentar os objetivos deste trabalho. A seção é dividida em duas partes objetivo geral e objetivos específicos.

### **1.3 OBJETIVO GERAL**

Desenvolver um aplicativo para dispositivos móveis que auxilie no gerenciamento de estoque e de vendas para profissionais autônomos, que seja gratuito e de fácil uso.

### **1.4 OBJETIVOS ESPECÍFICOS**

- Pesquisar as formas pelas quais vendedores autônomos registram seus estoques e vendas;
- Pesquisar modelos de interface de usuário;
- Documentar o desenvolvimento utilizando diagramas de engenharia de software;
- Desenvolver o protótipo usando a linguagem de programação Java;
- Realizar testes de funcionalidade;
- Realizar testes unitários;

- Realizar testes o *software* com o público-alvo;
- Realizar a coleta e análise dos resultados do público.

## **1.5 ORGANIZAÇÃO DO TEXTO**

Este documento está dividido em 7 capítulos. O primeiro capítulo introduz os conceitos mais importantes para o trabalho, e estabelece os objetivos e a motivação do trabalho. O capítulo 2 traz a revisão bibliográfica, onde os tópicos mais importantes para o trabalho são explicados. No capítulo 3 há o estado da arte, onde é apresentada aplicações semelhantes a proposta neste trabalho. O capítulo 4 apresenta a metodologia utilizada no desenvolvimento do trabalho. O capítulo 5 descreve o desenvolvimento da aplicação, desde sua modelagem, até a fase de testes. O capítulo 6 traz a análise dos resultados obtidos nos testes com usuários e do questionário de satisfação do aplicativo. Por fim, o capítulo 7 apresenta a conclusão do trabalho.

## 2 REVISÃO BIBLIOGRÁFICA

Esta seção do trabalho descreve as ferramentas utilizadas para o desenvolvimento e modelagem do aplicativo. Inicialmente será apresentada uma breve explicação dos conceitos de Programação Orientada a Objetos (POO), seguida por uma explicação sobre UML, será falado um pouco sobre Android, em seguida sobre Banco de dados, então UX (User Experience), por fim sobre os testes.

### 2.1 PROGRAMAÇÃO ORIENTADAS A OBJETOS (POO)

Para Santos (2013) a POO, é um paradigma da programação de computadores que usa objetos criados a partir de modelos (muitas vezes descritos por classes), para representar e processar dados usando programas de computador. Ainda, segundo o autor os modelos e objetos da POO podem ser definidos de forma que:

Modelos são representações simplificadas de objetos, pessoas, tarefas, conceitos etc. usados comumente por pessoas no seu dia a dia, independente do uso de computadores. (...) A criação e o uso de modelos é uma tarefa natural e a extensão desta abordagem à programação de computadores deu a origem ao paradigma Programação Orientada a Objetos. (SANTOS, 2013, p.3-4).

Para a construção do *Seller's Book* foi utilizada a linguagem de programação orientada a objetos Java, que, acordo com Melo (2021):

“Java é uma linguagem de programação e um ambiente computacional criado pela Sun Microsystems na década de 90, como linguagem de programação, o código Java é baseado em classes e orientado a objetos, com foco em segurança, portabilidade e alta performance<sup>2</sup>. (MELO, 2021).

Java, que pertence atualmente à Oracle, foi escolhido por ser uma linguagem com muitos recursos e por ter uma comunidade grande e versada, sendo o segundo idioma mais usado no GitHub (OLIVEIRA, 2019). Com uma grande comunidade existe vários materiais disponíveis para o estudo e desenvolvimento do aplicativo.

### 2.2 UNIFIED MODELING LANGUAGE (UML)

A *Unified Modeling Language*, também conhecida como UML, é uma linguagem gráfica voltada para a “visualização, especificação, construção e documentação de artefatos de sistemas complexos de *software*” (BOOCH; RUMBAUGH; JACOBSON, 2006, p.8). Segundo Fowler (2005), a UML nasceu da união de diversas linguagens gráficas de modelagem

orientadas a objetos que existiam no final dos anos oitenta e início dos noventa. Para Ventura (2019), pode-se separar os diagramas UML em 2 grupos:

- Diagramas estruturais: especificam detalhes da estrutura do sistema. Incluem os diagramas de classes, componentes, implantação, objetos, pacote, perfil, estrutura composta;
- Diagramas comportamentais: especificam detalhes do comportamento do sistema. São compostos pelos diagramas de casos de uso, atividades, máquinas do estado, sequência, comunicação, visão geral da interação, tempo.

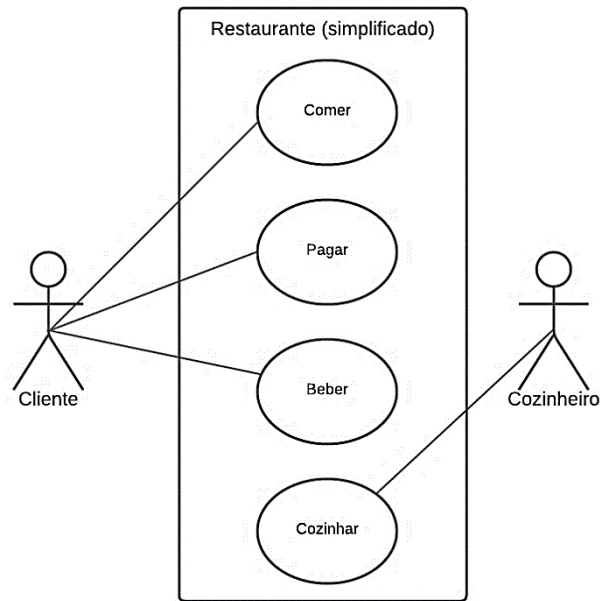
Para a modelagem do *Seller's Book*, foram utilizados o diagrama de classes e o diagrama de casos de uso da UML e, também, o diagrama de Entidade e Relacionamento (ER), que, apesar de não ser um diagrama UML, foi utilizado a fim de modelar o banco de dados da aplicação (FILHO, 2011).

O diagrama de classes é considerado o mais importante por Guedes (2018). O seu foco é permitir a visualização das classes que vão compor o sistema, com seus respectivos métodos, atributos, e de como essas classes se relacionam entre elas.

O diagrama de casos de uso é “usado normalmente nas fases de Levantamento e Análise de Requisitos do sistema, onde são determinadas as necessidades do usuário” (MARTINS et al, 2006). A Figura 1 mostra um exemplo simples de diagrama de casos de uso, com a representação de atores e de casos de uso.

Na Figura 1 há dois atores que são representados por bonecos palito, que representa tudo aquilo que é externo ao sistema e que interage com ele, como por exemplo um usuário, sistema, dispositivo ou mesmo uma empresa que desempenha um papel no sistema. As elipses representam os casos de uso, que identificam um comportamento chave no sistema. As linhas entre os atores e os casos de uso representam uma relação (FILHO, 2011). No caso do exemplo da Figura 1, o diagrama informa que existe um relacionamento do ator crítico de comida com os casos de uso comer, beber e pagar a comida, enquanto o ator cozinheiro se relaciona com o caso de uso cozinhar a comida.

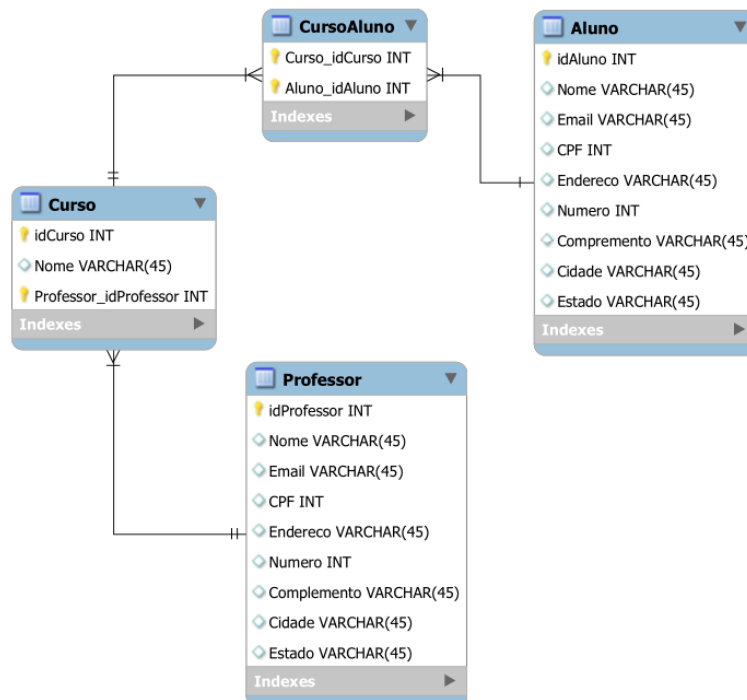
Figura 1: Diagrama de casos de uso para o sistema simplificado de um restaurante.



Fonte: Adaptado da Wikipedia (2011).

Por fim o diagrama Entidade Relacionamento é um tipo de diagrama que mostra “entidades”, como pessoas, objetos ou conceitos, que se relacionam entre si dentro de um sistema (NOGUEIRA, 1988 apud FRANK et al, 2021). Esse tipo de diagrama normalmente é usado para modelar bancos de dados. Na Figura 2 é mostrado um exemplo de um diagrama de entidade relacionamento.

Figura 2: exemplo diagrama entidade relacionamento.



Fonte: Matos (2013).



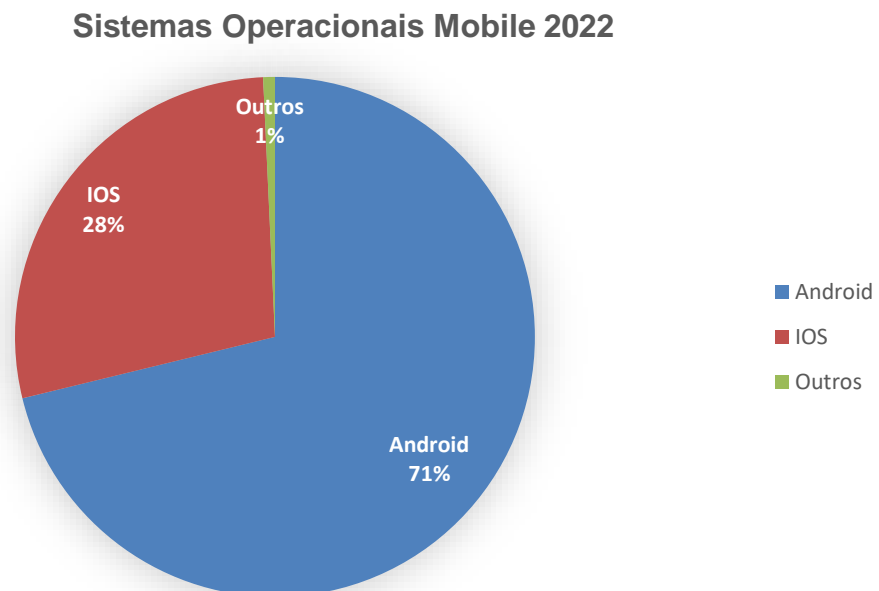
Explicando de forma resumida os componentes do diagrama, pode-se afirmar que:

Entidades são representadas graficamente por retângulos, e possuem atributos, que são responsáveis por caracterizá-las, fazendo referência a dados como nome, sexo, telefone, endereço etc. O atributo definido para representar a entidade é chamado de “Chave-Primária” (MAGALHÃES e NETO, 2010).

### 2.3 ANDROID

O *Seller's Book* foi construído para executar em Sistema Operacional (SO) Android, um sistema operacional da Google, presente em múltiplos aparelhos de diversas fabricantes. O Android é a plataforma mobile mais popular do mundo e é conhecido por ser baseado no núcleo do Linux (ABLESON et al, 2012). O gráfico da Figura 3 mostra a comparação de quantidade de usuários de sistemas operacionais mobile.

Figura 3: Gráfico de sistemas operacionais mais utilizados.



Fonte: Statcounter (2022).

De acordo com Pereira (2009), o Android foi construído para permitir aos desenvolvedores criar aplicações que tirem proveito de todas as funcionalidades que um aparelho portátil pode oferecer. Podendo, por exemplo, construir uma aplicação que utiliza recursos do núcleo do celular, tais como fazer uma chamada, enviar mensagem ou até mesmo acessar a câmera do aparelho. O que permite aos desenvolvedores se adaptarem e evoluírem junto a esses recursos para construírem melhores aplicações.

Como o Android possui código fonte aberto, qualquer elemento faltante, como *frameworks* e bibliotecas, pode ser fornecido pela comunidade global de desenvolvedores. Por exemplo, o Android não possui um interpretador de comandos (prompt de comando ou *shell*).

Contudo, como sua plataforma é aberta, é possível escrever e instalar essa funcionalidade em um dispositivo (ABLESON et al, 2012).

## **2.4 BANCO DE DADOS**

Um sistema de banco de dados é basicamente um sistema computadorizado de manutenção de registros. Date (2004) compara um banco de dados a um armário de arquivamento eletrônico, ou seja, um repositório para uma coleção de dados computadorizados. Dados são definidos por Elmasri e Navathe (2005, p.3) como:

Fatos que podem ser gravados e que possuem um significado implícito. Por exemplo, considere nomes, números telefônicos e endereços de pessoas que você conhece. Esses dados podem ter sido escritos em uma agenda de telefones ou armazenados em um computador, por meio de programas como o Microsoft Access ou Excel. Essas informações são uma coleção de dados com um significado implícito, conseqüentemente, um banco de dados.

Nesse trabalho foi utilizado o banco de dados SQLite, que é um banco de dados embutido. Isso significa que, ao invés de rodar de forma independente, ele coexiste com a aplicação (OWENS, 2006). Diferentemente da maioria dos sistemas de gerenciamento de banco de dados, o SQLite não possui uma arquitetura cliente/servidor, que tem como característica ser composta por dois elementos essenciais: estações moveis (clientes), e pontos de acesso (servidores) (MENDES, 2020). No SQLite, o banco de dados está integrado a aplicação que o utiliza, logo, o único recurso compartilhado entre as aplicações é um arquivo de banco de dados no disco. Caso seja necessário copiar ou fazer um backup do arquivo, é suficiente realizar uma cópia do próprio arquivo (KREIBICH, 2010).

## **2.5 USER EXPERIENCE (UX)**

*User Experience* (Experiência do usuário), é um termo criado por Norman (c2017) que diz respeito a percepção que o usuário de um sistema tem sobre o uso dele. O conceito em si é antigo e existe desde que pessoas começaram a usar todo tipo de objetos. No decorrer dos dias nós nos tornamos usuários de diversos recursos, como por exemplo, o alarme, o carro, a cadeira. quando você usa qualquer objeto para realizar um objetivo, você tem uma experiência. Esse princípio continua o mesmo com os produtos digitais, de forma que a experiência do usuário pode ser boa ou ruim dependendo de como ele usa esses produtos (TEIXEIRA, 2006).

De acordo com Norman (c2017) a experiência não diz respeito apenas ao aparelho, site ou aplicativo, ele diz que a:

Experiência é tudo, a forma que você experencia a vida, os objetos, máquinas ou o serviço de computador (...) experiência começa desde que você vê o produto e o compra na loja, o leva para casa e o tira da caixa para começar a usar.

Norman e Nielsen (c2017) concordam que o primeiro requerimento para uma boa UX é corresponder as necessidades do usuário, de forma que, deve ser simples e elegante para produzir e de uso satisfatório para o usuário. Para obter uma UX de alta qualidade no produto, deve-se assegurar a fusão de diversos serviços, incluindo engenharia, marketing, design gráfico, industrial e design de interface.

Para criar uma boa UX, Jacob Nielsen criou as 10 heurísticas da usabilidade. Eles são chamados de "heurísticas" porque são regras gerais e não diretrizes de usabilidade específicas (NIELSEN, 1994). Essas regras, criadas para ajudar a criar uma boa interface para o usuário, são listadas no Quadro 1.

Deve-se destacar que UX e *User Interface* (UI) são duas abordagens diferentes, apesar de se complementarem, cada uma trabalhando para oferecer uma melhor experiência de usuário (DIGITAL HOUSE, 2020):

- UI: conceito que define a forma com que o usuário irá interagir com o sistema, tendo como elementos, botões, tipografia, ícones e cores.;
- UX: conceito que se estende para além da experiência do usuário com o produto e inclui até mesmo as etapas anteriores ao seu uso. Ou seja, ofertar um produto relevante é fundamental, mas também é pertinente como o produto será consumido.

Quadro 1: 10 heurísticas da usabilidade.

#	Heurística	Descrição
1	Visibilidade de status do sistema	O sistema deve manter os usuários informados sobre o que está acontecendo, em tempo real.
2	Correspondência entre sistema e mundo real	O sistema deve ter a mesma linguagem que seu usuário utiliza no dia a dia.
3	Controle e liberdade para o usuário	O sistema deve permitir que o usuário tenha liberdade para realizar o que deseja e desfazer enganos.
4	Consistência e padronização	A interface deve manter uma consistência, permitindo que o usuário identifique os padrões de estética.
5	Prevenção de erros	O sistema deve prevenir que erros ocorram.
6	Reconhecimento em vez de recordação	O sistema deve minimizar a quantidade de informação que o usuário precisa memorizar.
7	Eficiência e flexibilidade de uso	Usuários leigos precisam de informação detalhada para realizar suas tarefas, e ferramentas para utilizar o sistema eficientemente com mais experiência.
8	Estética e design minimalista	O conteúdo deve ser direto, manter as partes importantes em primeiro plano e as menos importantes em segundo.
9	Auxílio do sistema ao usuário para reconhecer, diagnosticar e recuperar erros	Se acontecer um erro o sistema deve mostrar ao usuário qual erro e como se recuperar dele de forma objetiva.
10	Ajuda e documentação	Apesar de ser uma área menos acessada, é importante o usuário ter uma opção de resolver suas dúvidas dentro da aplicação.

Fonte: Jakob Nielsen (1995).

## 2.6 TESTE DE SOFTWARE

Uma parte importante do desenvolvimento de software, são os testes. Para Neto (2007), o teste de software é o processo de execução de um produto, que visa determinar se ele atingiu suas especificações e funcionou corretamente no ambiente para qual foi projetado. O teste de software tem o objetivo de revelar falhas em um produto, para identificar suas causas e poder corrigi-las.

Para este trabalho foram feitos três tipos de teste: funcional, unitário e instrumentado. O teste funcional é conhecido também como teste de caixa preta, por se tratar de uma caixa com o conteúdo desconhecido, da qual vemos apenas o lado externo. Nesse tipo de teste são verificadas as funcionalidades do sistema, sem se preocupar com a implementação interna (FRANCINE et al, 2000).

O teste unitário é o teste que busca testar cada unidade do sistema individualmente, com o objetivo de isolá-lo para garantir que elas estejam funcionando conforme especificado (MIRANDA, 2017). Um teste instrumentado pode ser descrito como um teste executado diretamente no dispositivo ou emulador (instrumentos), simulando o comportamento de um usuário. Em resumo, este teste valida a lógica gráfica do aplicativo (MARCUS, 2016).

### 3 ESTADO DA ARTE

Para definir os requisitos da aplicação foram pesquisados sistemas com funcionalidades parecidas com o *Seller's Book*. Foram pesquisados os sistemas informados durante o levantamento de requisitos, outros aplicativos e trabalhos que tem relação com controle de vendas ou estoque.

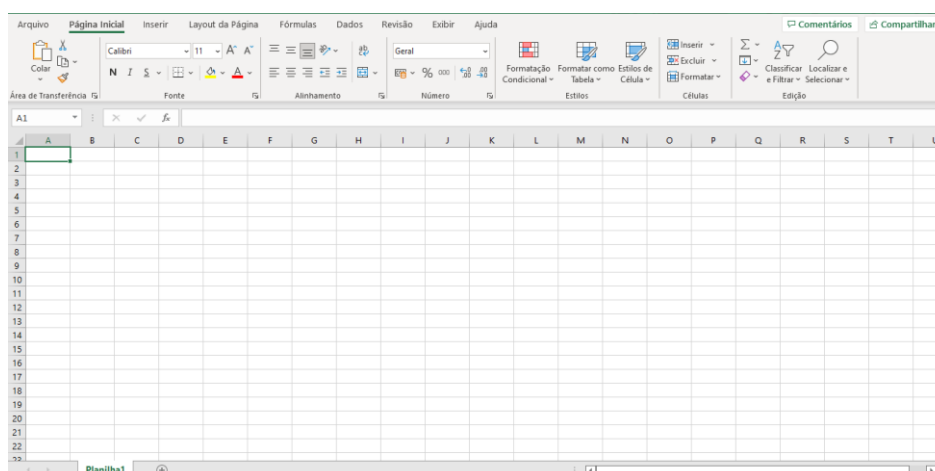
#### 3.1 APLICAÇÕES

Esta seção apresenta os aplicativos analisados devido a sua relação com a proposta do trabalho. Esses aplicativos são o Excel da Microsoft (2016), o Super Revendedores (2022) e o Kyte (2022).

##### 3.1.1.1 EXCEL

Microsoft Excel, é a ferramenta de planilhas da Microsoft, que é um tipo de programa de computador que utiliza tabelas para realização de cálculos ou apresentação de dados, sendo capaz de gerar diferentes gráficos para os mesmos. Suas funcionalidades ajudam a apresentar dados de forma atrativa, com formatação, mini gráficos e tabelas para facilitar a compreensão da informação (EXCEL, 2016). Porém, o Excel é um sistema pago, e, apesar de todas as suas funções, usá-las de forma eficiente requer experiência ou de um treinamento na plataforma. Como alternativa gratuita ao Excel, há ferramentas como o Google Planilhas, ou o editor de planilhas do LibreOffice, ambos com funcionalidades equivalentes às do Excel.

Figura 4: Interface gráfica do usuário do Excel.



Fonte: <https://www.microsoft.com/pt-br/microsoft-365/p/excel/CFQ7TTC0HR4R>.

A interface do Excel consiste em uma planilha com aproximadamente 1 milhão de linhas e 16 mil colunas, que o usuário pode preencher com dados desejados da forma que preferir. O usuário pode criar funções dentro de cada célula, seja para cálculos simples ou complexos, como comparações, buscas e filtragem de informações. Na parte superior da interface do Excel há uma barra de ferramentas que auxilia na formatação da planilha de acordo com a necessidade do usuário.

O Excel pode ser uma excelente ferramenta para usuários experientes. Porém, para usuários sem essa experiência, essa ferramenta pode não passar de uma agenda muito sofisticada.

### 3.1.2 SUPER REVENDEDORES

Super revendedores é um aplicativo de gerenciamento de estoque e vendas com foco em revendedores. O aplicativo permite acessar o catálogo de marcas de vendas, compartilhar produtos com sua lista de clientes, registrar vendas e fazer o controle do estoque (SUPER REVENDEDORES, 2017). Além disso, o Super Revendedores mantém os dados de custos dos produtos e dos lucros das vendas. A Figura 5 mostra a tela inicial do aplicativo Super Revendedores.

Figura 5 Tela inicial do Super Revendedores.



Fonte: [https://play.google.com/store/apps/details?id=com.superrevendedores.app&hl=pt\\_BR&gl=US](https://play.google.com/store/apps/details?id=com.superrevendedores.app&hl=pt_BR&gl=US).

Quando o usuário adiciona produtos ele pode adicionar um produto próprio, ou pesquisar nos produtos dos catálogos de outras marcas, para o preenchimento mais rápido dos dados. Ao criar um cliente, o usuário pode adicionar o mesmo direto da lista de contatos de seu aparelho telefônico, facilitando o processo.

Mesmo com todas essas funções o aplicativo, atualmente, possui diversas reclamações de bugs, as quais incluem produtos que somem do estoque ou que são duplicados. Porém, seus desenvolvedores parecem estar se esforçando para proporcionar suporte adequado ao sistema, respondendo usuários com problemas e mantendo o aplicativo atualizado.

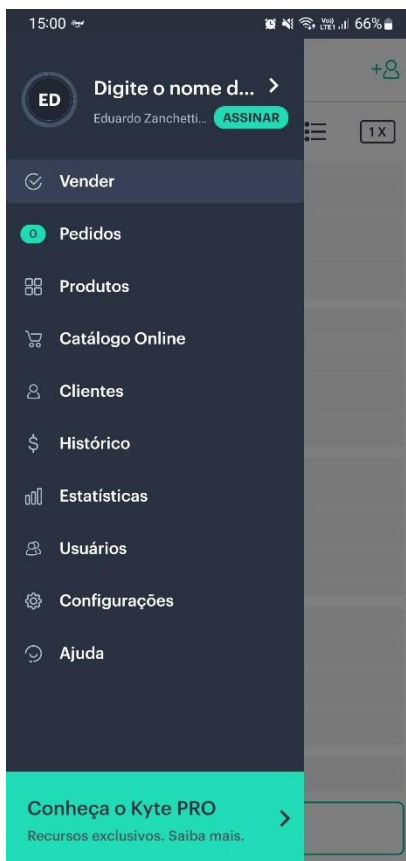
### 3.1.3 **KYTE**

Kyte é um aplicativo com funções muito semelhantes ao Super Revendedores, mas com recursos pagos por assinatura. Dentre as diferenças em relação ao Super Revendedores, destaca-se que os produtos do Kyte não têm a opção de um campo validade, em contrapartida, a ferramenta oferece a funcionalidade de adicionar um código de barras do produto. Além disso, como o Kyte busca atender a qualquer tipo de vendedor, o usuário pode selecionar vender o produto por fração (quilo, litro, metro por exemplo) ou mesmo por unidade.

O Kyte possui uma barra de utilidades com todas suas funções, a qual é mostrada na Figura 6. Durante a análise do aplicativo, percebeu-se que a transição entre as telas do Kyte eram mais lentas do que as do Super revendedores. Além disso, o primeiro contato ao abrir o aplicativo não é intuitivo, devido ao excesso de botões na tela, o que dificulta descobrir e memorizar a ação de cada um. Os recursos pagos do Kyte incluem: criação de catálogo online para que os clientes realizem pedidos, cadastro de mais de uma conta no dispositivo, exportação de dados para o computador, cadastro ilimitado de produtos, dentre outros.

Entretanto, o Kyte é um aplicativo que tem o objetivo de ajudar empreendedores. Seu serviço base já serve a maior parte do seu público, e seu serviço pago garante mais segurança e prioridade no atendimento com o aplicativo, além de liberar as funções online do, como a vitrine, o local que se coloca os produtos em destaque, e pedidos online (KYTE, 2022).

Figura 6: Captura da aba de ferramentas do Kyte.



Fonte: [https://play.google.com/store/apps/details?id=com.kyte&hl=pt\\_BR&gl=US](https://play.google.com/store/apps/details?id=com.kyte&hl=pt_BR&gl=US).

### 3.1.4 ANÁLISE COMPARATIVA DOS SISTEMAS

Para comparar o aplicativo proposto neste trabalho com os demais apresentados neste capítulo foi construído a Quadro 2 que resume as diferenças entre o *Seller's Book* e os aplicativos analisados.

A partir do Quadro 2 pode-se perceber que tanto o Kyte quanto o Super revendedores possuem um limite de produtos para o público não pagante, apesar de não haver informações sobre qual é esse limite. Quanto ao Excel, seu limite de uso depende da organização do usuário com as linhas, colunas e planilhas do programa. Contudo, é muito improvável um usuário utilizar todas as mais de 16 bilhões de células disponíveis no Excel. Para o *Sellers's Book*, não foi especificado um limite, sendo assim, o único agente limitador da quantidade de dados no aplicativo, é a própria capacidade de armazenamento.

Quanto às estatísticas de vendas, tanto o Kyte e Super revendedores, tornaram este um recurso premium, cujo uso necessita de assinaturas pagas de seus respectivos serviços. Já o Excel trabalha muito bem com gráficos e tabelas em geral, mas para utilizar essas funcionalidades, o usuário precisaria ter mais experiência com a ferramenta. O *Seller's Book*



limitou a variedade dessas informações às mais importantes, deixando as estatísticas mais variadas para os trabalhos futuros.

Para a divulgação do estoque o *Seller's Book* possui em sua especificação a geração de PDFs com o estoque de produtos guardados no aplicativo, para que o usuário possa divulgá-los aos seus clientes. Nesse sentido tanto o Kyte quanto o Super revendedores, utilizam de lojas virtuais para a divulgação dos estoques, porém esse é um serviço pago, em razão da necessidade de servidores para hospedar as lojas.

Quadro 2: Comparação dos aplicativos.

<b>APLICATIVOS</b>	<b>LIMITE DE PRODUTOS</b>	<b>ESTATISTICA DE VENDAS</b>	<b>DIVULGAÇÃO DO ESTOQUE</b>
<b>SELLER'S BOOK</b>	Limitado pela memória do Smartphone	Valor arrecadado, em atraso e planejado no mês	PDF
<b>KYTE</b>	Produtos ilimitados para usuários premium	Premium	Premium
<b>SUPER REVENDEDORES</b>	Produtos ilimitados para usuários premium	Premium	Loja virtual Premium
<b>EXCELL</b>	Limitado pela organização do usuário.	Limitado pela experiência do usuário com a ferramenta.	PDF

Fonte: Autoria própria

## 4 METODOLOGIA

Esse trabalho consiste em uma pesquisa quali-quantitativa, ou seja, que “interpreta as informações quantitativas por meio de símbolos numéricos e os dados qualitativos mediante a observação, a interação participativa e a interpretação do discurso dos sujeitos (semântica)” (KNECHTEL, 2014, p. 106), cujo público-alvo são vendedores autônomos. A natureza da pesquisa será aplicada, o que segundo, Gerhardt e Silveira (2009, p. 35), denota uma pesquisa que procura “gerar conhecimento para aplicação prática, levando à solução de problemas específicos”.

Inicialmente foi feita uma entrevista com uma vendedora de cosméticos a fim de procurar identificar requisitos de interesse para o sistema. Em seguida, foi elaborado um questionário a partir dessas informações, que foi aplicado a um grupo de vendedores, a fim de identificar quais funcionalidades eles buscam em um aplicativo com esta proposta, e assim criar os requisitos do sistema.

A entrevistada foi uma mulher de 59 anos com mais de 30 anos trabalhando com vendas de cosméticos. Durante a entrevista inicial houve perguntas relacionadas ao uso de aplicativos no trabalho e o que seria necessário em um aplicativo capaz de lhe ajudar com a rotina do trabalho de vendas. Após a entrevista, obteve-se acesso a um grupo de vendedores e assim, foi-lhes enviado um questionário com perguntas semelhantes aos da entrevista.

Dentre as perguntas, este questionário indagava quais os aplicativos para controle de vendas usados pelos respondentes, ou, caso não utilizassem nenhum, o porquê dessa escolha. As perguntas feitas nesse questionário podem ser encontradas no Apêndice A, com as respostas no Apêndice B.

Em seguida foi feito o processo de Engenharia de Software, criando os diagramas UML para o sistema. Inicialmente, foram feitos o diagrama de casos de uso, que descreve as funcionalidades do sistema e o diagrama de classes, para representar a estrutura de classes que servem de modelo para os objetos do sistema.

Com os diagramas concluídos o desenvolvimento do sistema seguiu para a fase de implementação, em que foi adotada a linguagem de programação Java e o banco de dados SQLite, para construir o aplicativo para o SO móvel do Google, o Android.

Para o processo de desenvolvimento, foi adotado o modelo incremental, que se baseia na ideia de desenvolver uma versão inicial do sistema, expô-la aos comentários dos usuários e continuar o desenvolvimento através de várias versões (incrementos), que serão também

validados pelos usuários, até que um sistema adequado seja desenvolvido (SOMMERVILLE, p.35, 2011). Neste sentido, os incrementos do sistema eram apresentados a vendedora entrevistada anteriormente como forma de coletar *feedbacks* durante o desenvolvimento do sistema. A condução dessas entrevistas foi feita em tom informal, de forma que era mostrada a interface e funções recém desenvolvidas do aplicativo e a vendedora dava sua opinião sobre eles. Essas entrevistas traziam, no geral, mudanças relacionadas a estética do aplicativo e visibilidade dos componentes, ajustando cores para melhor contraste dos componentes com as letras.

Para desenvolver o aplicativo, foi usado o Ambiente de Desenvolvimento Integrado (*Integrated Development Environment* ou IDE) Android Studio, a ferramenta oficial do Google para desenvolvimento Android (GOOGLE DEVELOPERS, 2021). Com ele, foi utilizado o Material Design, um sistema criado pela Google para ajudar equipes de desenvolvimento a construir UXs digitais de alta qualidade para Android, IOS, Flutter e Web (MATERIAL DESIGN, c2020).

O primeiro foco do desenvolvimento do aplicativo foi a interface de usuário e sua consistência com a proposta da UX, a qual foi construída dentro do próprio ambiente de programação Android, o Android Studio (2013). Essa interface foi validada por dois vendedores autônomos, a fim de obter uma interface mais intuitiva. Em seguida, foi feito o desenvolvimento do *back-end* (EWALLY, 2021), a estrutura que possibilita a operação do sistema, começando pelo desenvolvimento do banco de dados, e depois, do próprio sistema, acompanhado por testes unitários, que consistem em verificar o comportamento das menores unidade da aplicação (DIGITÉ, 2022).

Durante todo o desenvolvimento foram feitos testes unitários e instrumentados, buscando erros menores no sistema. E no fim do desenvolvimento do aplicativo foram feitos diversos testes funcionais no aplicativo, que são os testes que “a validação de software que determinada funcionalidade é verificada, sem levar em conta a estrutura do código-fonte” (SOARES, 2021). Concluídos os testes e as correções do sistema, ele foi entregue para a entrevistada, para ser submetido a validação do usuário em vista de validar se o mesmo está de fato de acordo com suas necessidades, identificar problemas não encontrados anteriormente e também melhorias adicionais ao sistema.

## 5 DESENVOLVIMENTO

Nesse capítulo é detalhado o processo de desenvolvimento do aplicativo desde a modelagem até os testes finais do sistema.

### 5.1 MODELAGEM

Antes de começar o desenvolvimento do aplicativo houve a modelagem e planejamento do sistema. Começando pelo levantamento e análise de requisitos, que teve como base a pesquisa de levantamento de requisitos (de agora em diante será chamado apenas de levantamento de requisitos), fase na qual se analisou o que o aplicativo precisaria fazer. Para isso, foram utilizadas as respostas do questionário aplicado no grupo de autônomos. Além do CRUD (*Create, Read, Update, Delete*), que comporta as quatro operações básicas usadas em banco de dados (criação, consulta, atualização e deleção de dados) para a manutenção dos clientes e produtos do sistema, os requisitos identificados a partir dessa pesquisa foram os de geração de PDF e o de lembrete de cobrança. Os resultados do levantamento de requisitos podem ser vistos com mais detalhes no Apêndice B, enquanto a relação de requisitos pode ser encontrada no Apêndice C.

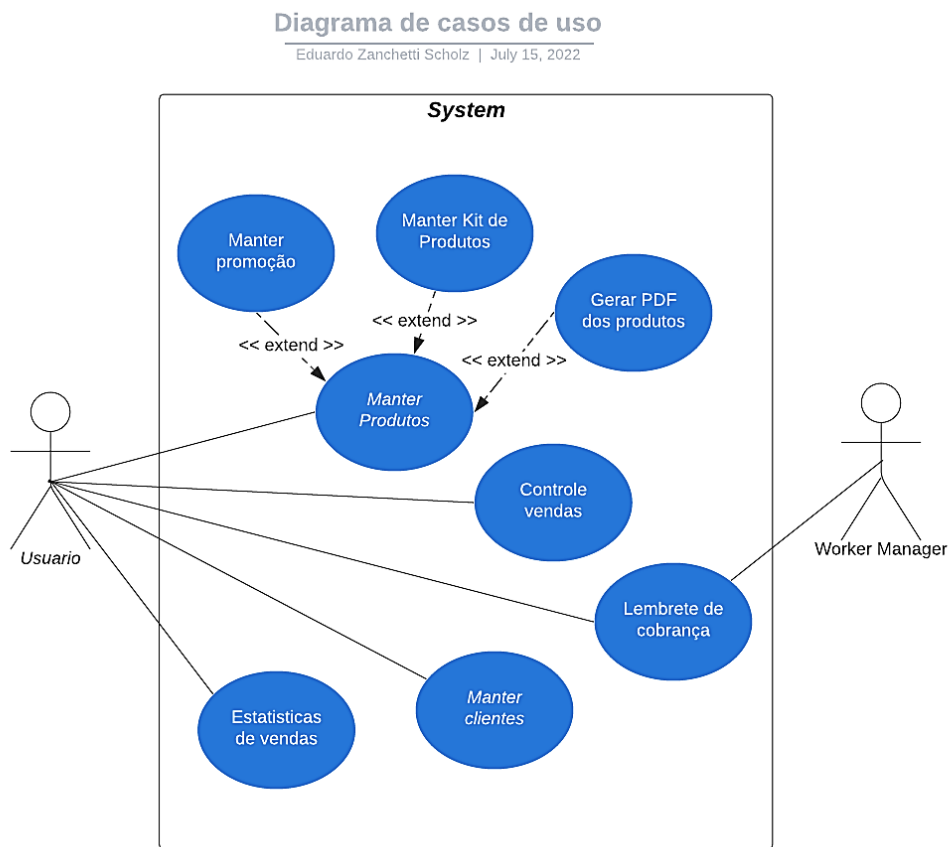
A análise de requisitos, foi usada como base para a criação do diagrama UML de Casos de Uso. Cujas versão final pode ser observada na Figura 7.

A partir do diagrama de casos de uso foi escrita a especificação de casos de uso. Essa especificação detalha o fluxo de funcionamento de cada caso de uso. O documento pode ser encontrado na íntegra no Apêndice D. De forma breve, o [CdU001] “Manter produtos” representa as funções de criação, remoção, edição e visualização dos produtos salvos no sistema. O [CdU002] “Manter clientes” e o [CdU003] “Controle de vendas” representam as mesmas funções descritas em [CdU001], mas a partir do contexto dos clientes das vendas, respectivamente.

O [CdU004] “Estatísticas de vendas” representa a visualização dos dados gerais do valor arrecadado pelo usuário no mês corrente. O [CdU005] “Notificar Pagamento” é o único caso de uso em que o usuário não é o ator principal, pois representa a notificação gerada pela API dentro do sistema *WorkManager*, que avisa duas vezes por dia se há um pagamento vencendo naquele dia ou se há algum pagamento em atraso.

Quanto aos casos de uso restantes, foi decidido que ele não seriam implementados em razão do tempo disponível para o desenvolvimento. Esses casos de uso são, [CdU006] “Manter Promoção” representa o caso de uso que cria uma promoção para um produto, atribuindo um preço temporário para ele. O [CdU007] “Manter Kit” representa o caso de uso que cria um conjunto de produtos vendido em um pacote com valor próprio. O [CdU008] “Gerar PDF” representa a função que geraria um PDF com o catálogo de produtos para o usuário enviar para seus clientes saberem seu estoque.

Figura 7: Diagrama de casos de uso.



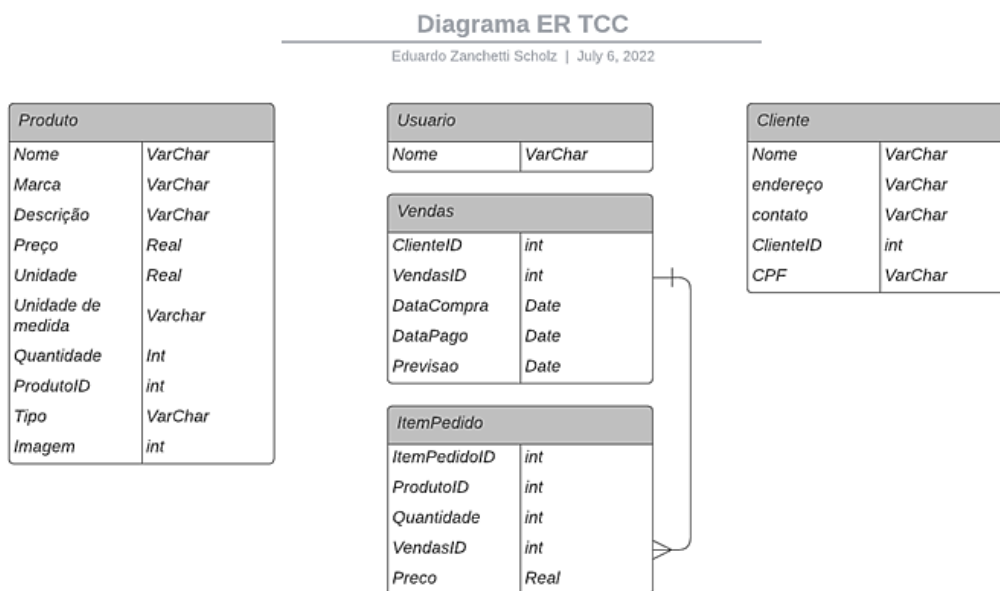
Fonte: Autoria própria.

O próximo passo tomado, foi criar o diagrama ER para o banco de dados que também sofreu alterações durante o desenvolvimento. A Figura 8 mostra a versão final do diagrama.

Note que o diagrama possui redundância de dados onde poderia haver uso de chaves estrangeiras. Essa escolha foi tomada para evitar que, quando um produto ou um cliente fossem apagados, o sistema excluísse a venda ou o item do pedido que está relacionado a ele. Além disso, a abordagem também permite manter o histórico de valores pagos pelo cliente por cada produto. De forma que a alteração do preço de um produto não mude o “histórico” de pagamento do cliente. Ao final do desenvolvimento foi criado também o diagrama de classes,

através do aplicativo Visual Paradigm (<https://www.visual-paradigm.com>), que está disponível no Apêndice E.

Figura 8: Diagrama Entidade-Relacionamento.



Fonte: Autoria própria.

## 5.2 IMPLEMENTAÇÃO

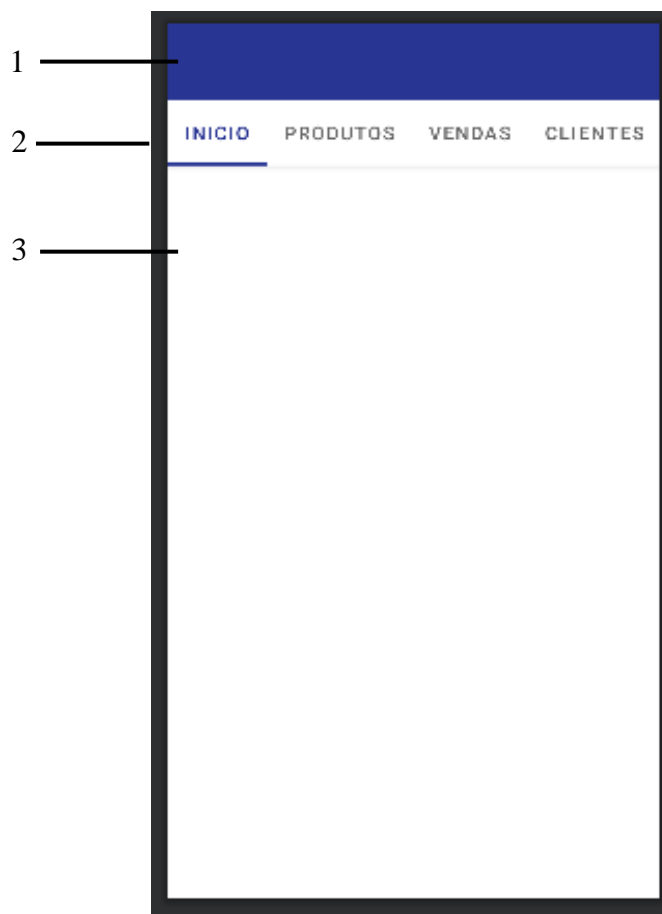
Antes do início da implementação foi decidido que, em razão do tempo disponível para o desenvolvimento do sistema, os casos de uso: “Manter kits de produtos”, “Manter Promoção” e “Gerar PDF dos produtos” não seriam implementados.

O primeiro passo no desenvolvimento foi criar as telas do aplicativo. Utilizando as ferramentas de edição de XML do Android Studio, as telas foram feitas sem maiores dificuldades, e pouca escrita de código. A Figura 9 mostra a tela principal do aplicativo com o conteúdo vazio. Durante o desenvolvimento das telas buscou-se seguir a heurística de Nielsen, buscando tornar o aplicativo mais intuitivo, e com uma UX de melhor qualidade.

Na parte superior da tela existe uma *ToolBar* (indicada pelo número 1), um componente que serve para substituir a *ActionBar*. A *ActionBar* é uma barra que existe por padrão no aplicativo, ela normalmente mostra o nome do aplicativo e um menu, que normalmente leva a opções extras, a *ActionBar* pode se comportar diferente a depender da versão do Android, foi preferido usar a *ToolBar* (1) que apresenta os mesmos recursos em todas as versões da *ToolBar* (1). Além dela ser compatível com o Material Design do Google. Apesar do aplicativo ainda não possuir nenhuma opção extra, esse componente foi mantido para uso em trabalhos futuros.

Antes de poder explicar os fragmentos do Android, deve-se primeiro explicar as *Activity*s. As *Activity*s são um componente crucial de um aplicativo (app) para Android, e a maneira como as atividades são lançadas e reunidas é uma parte fundamental do modelo de aplicativo da plataforma. Diferente da convenção de muitas linguagens estruturadas ou OO, em que os sistemas são executados através do método “*main()*”, o sistema Android inicia o código em uma instância *Activity* invocando métodos de *callback* que correspondem a estágios específicos do ciclo de vida (ANDROID DEVELOPERS, 2019).

Figura 9: Tela Principal do aplicativo.



Fonte: Autoria própria.

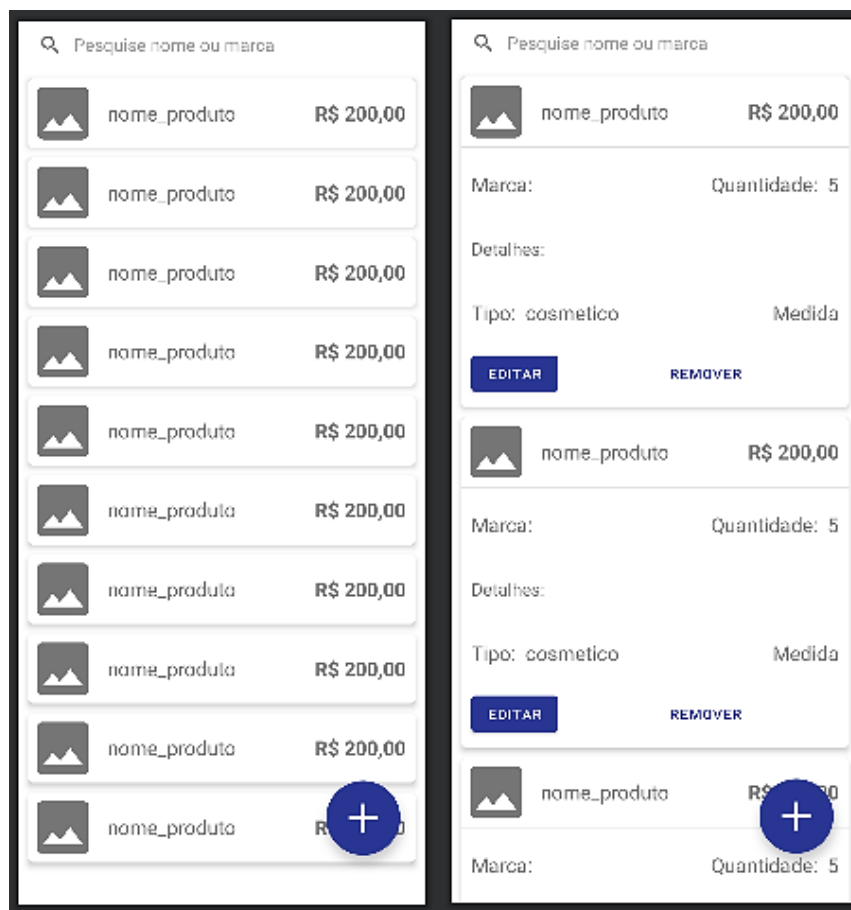
Fragmentos no Android Studio funcionam igual às *Activity*s, as telas do sistema, mas os fragmentos podem ser colocados dentro de *Activity*s ou de outros fragmentos. Assim, é possível combinar vários fragmentos em uma única atividade para criar uma UI com vários painéis e reutilizar o fragmento em diversas *Activity*s (ANDROID DEVELOPERS, 2021). Simplificando, fragmentos são telas que podem ser colocadas dentro de outras telas.

Abaixo da *ToolBar* (1) há um *TabLayout* (indicado pelo número 2) e um *ViewPager2* (indicado pelo número 3). Esses dois componentes foram usados em conjunto para construir

em um mesmo ambiente diferentes telas com o uso dos fragmentos. O motivo de usar o *TabLayout* (2) consiste em que o usuário possa saber onde ele está no sistema ao visualizar o componente, que indica se ele se encontra na tela de início, de produtos, de vendas ou de clientes. O *ViewPager2* (3) é a segunda versão do *ViewPager*, componente que permite ao usuário alterar entre fragmentos ao arrastar seu dedo para o lado na tela, tornando a troca de telas mais intuitiva.

A Figura 10 mostra o fragmento de produtos. Na parte esquerda, a tela é mostrada com todos os produtos listados, sem exibir detalhes. Na parte direita, com os produtos após a seleção do usuário, com todos os detalhes a mostra.

Figura 10: Imagens do fragmento de produtos.



Fonte: Autoria própria.

Quando expandida, a visão do produto também mostra botões para edição e remoção do produto. Quando o usuário estiver prestes a salvar sua edição ou remover o produto, ele é alertado pelo sistema, que solicita a confirmação da operação, conforme a heurística da



usabilidade de prevenção de erros listada no Quadro 1, que busca evitar que o usuário cometa erros ou enganos durante o uso do aplicativo.

Para a listagem dos produtos foi usado o componente *RecyclerView* do Android, que é um componente que possibilita construir listas personalizadas no aplicativo. O *RecyclerView* possui o diferencial de otimização da lista, que permite carregar apenas o conteúdo mostrado na tela, e guardar na memória de acesso rápido os próximos produtos, reciclando os já vistos quando exibidos novamente no componente.

Para usar o *RecyclerView* foi necessário usar duas outras classes, *Adapter* e *ViewHolder*. O *ViewHolder* é uma classe que serve para conectar o XML responsável por alimentar itens para uma lista, como aquelas da tela de Produtos e de Clientes. Os atributos dessa classe apontam para os *Views* do XML que precisam ser preenchidos. Os *Views* são componentes visuais da tela que são especializados em diferentes elementos, como imagens, campos de entrada, etc. Por fim, o *Adapter* é utilizado para preencher cada linha das listas em tela com os dados corretos. Em outras palavras, o *Adapter* recebe os dados do Banco de dados e utiliza o *ViewHolder* para preencher esses dados no modelo XML do item da lista, que então envia o item para a lista do *RecyclerView*. Esse processo se repete até que cada registro recebido do banco de dados seja incluído em sua lista.

Na parte superior do fragmento há uma barra de pesquisa para a busca do nome ou marca do produto. No fragmento de produtos também há um botão “+” localizado no canto inferior direito. Esse botão serve para adicionar um produto à lista e, conseqüentemente, ao banco de dados. Quando pressionado, o sistema abre a caixa de diálogo personalizada para a adição de produtos.

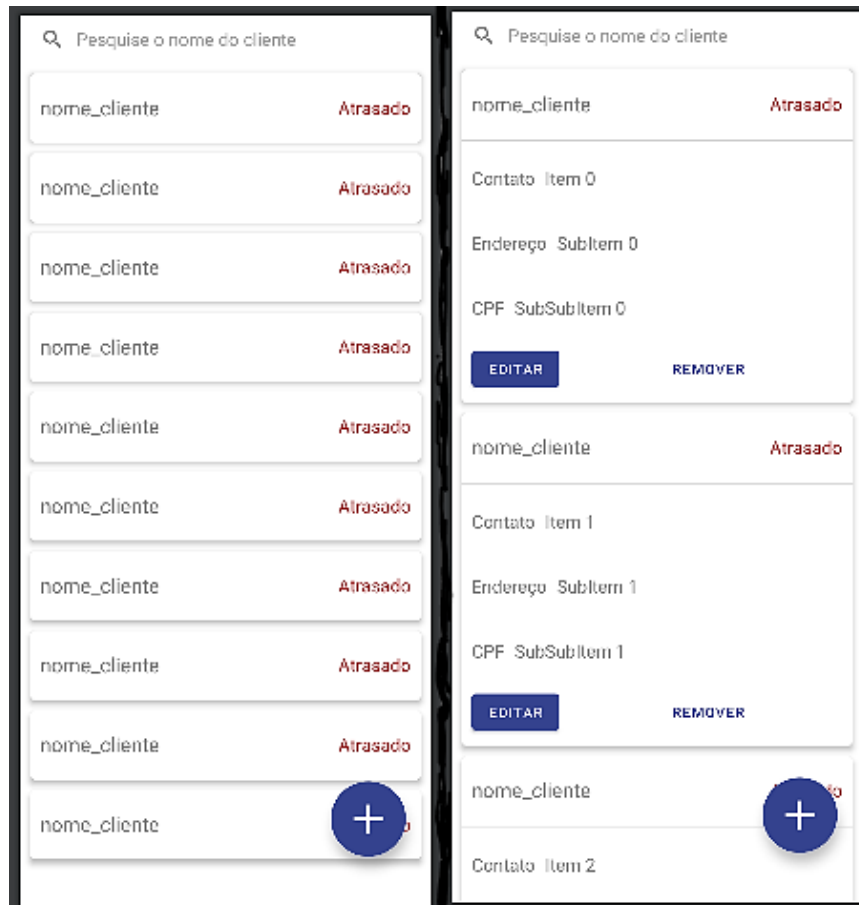
O fragmento dos Clientes, mostrado na Figura 11, funciona de forma idêntica ao fragmento de produtos, com a única diferença que, quando o cliente tem algum pagamento atrasado, esta informação é mostrada para o usuário.

O terceiro fragmento criado foi o de vendas, na Figura 12. O fragmento de vendas possui como diferença dos outros fragmentos, o campo de exibição do valor total da venda e o símbolo que informa se a venda está paga (um cifrão verde) ou não (um cifrão cortado vermelho). Quando tocado para exibir mais detalhes, a venda mostra também uma lista dos produtos vendidos. Além disso, a tela possui um botão de pagar, que quando pressionado muda o estado do pagamento da venda.

Diferentemente do fragmento de clientes e do de produtos, quando pressionado o botão “+”, o sistema não irá abrir uma caixa de diálogo e sim a tela mostrada na Figura 13. Essa escolha foi feita porque, para criar uma venda são necessários mais passos se comparado com

criação de um cliente ou produto. No caso de adição de vendas é necessário selecionar um cliente, um grupo de produtos para a venda e a data prevista para o pagamento. Fazer todos esses passos dentro do fragmento tornaria o processo muito complexo e mais aberto a falhas.

Figura 11: Fragmento de Cliente.



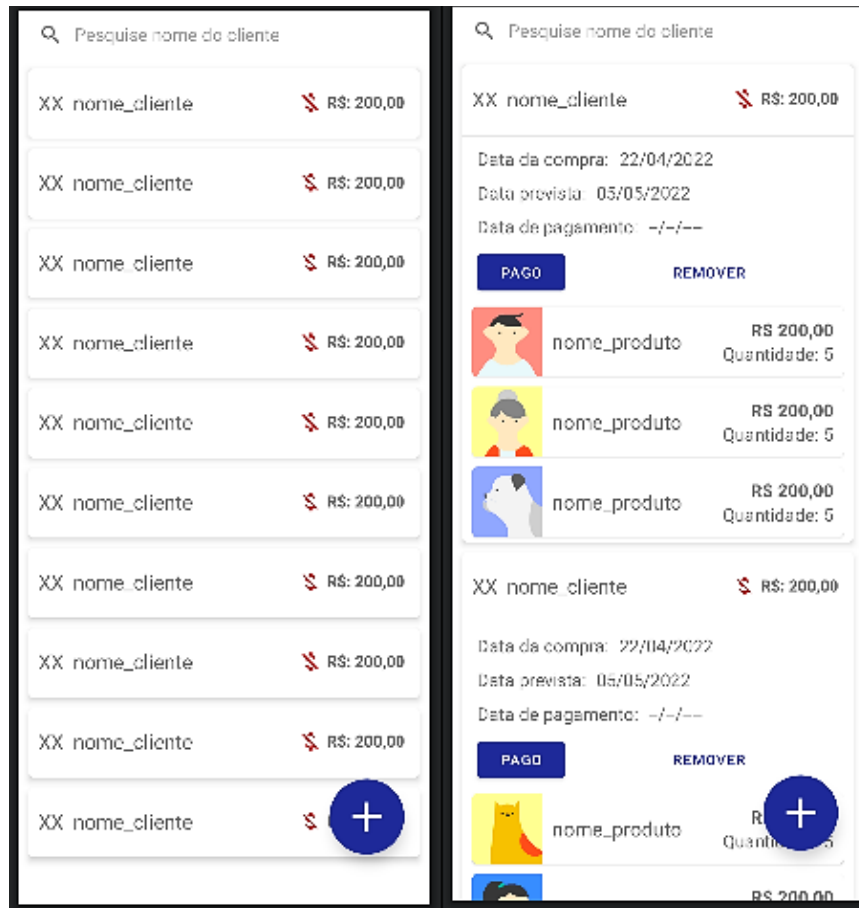
Fonte: Autoria própria.

A Figura 13 mostra as *Activitys* de criação de venda. Na esquerda há a parte de seleção de cliente, com uma barra de busca para auxiliar o usuário a encontrar o cliente desejado. Na direita, há a seleção de produtos. Ela é dividida em duas *RecyclerViews*, a parte de baixo, que mostra a lista completa de produtos no sistema, e a de cima, que mostra os produtos adicionados ao “carrinho” de venda. Quando o usuário salva a venda, o sistema mostra uma caixa de diálogo perguntando a data que o cliente deve efetuar o pagamento. Essa data será usada para notificar o usuário no dia em que o cliente deve realizar o pagamento.

O último fragmento criado foi o fragmento de início, mostrado na Figura 14. Esse fragmento tem o intuito de mostrar dados importantes assim que o usuário abre o aplicativo. No topo do fragmento é mostrada a logo do aplicativo junto com o nome do mesmo. Logo abaixo, é mostrado o valor arrecadado no mês, os valores em atraso, e o valor previsto para

pagamento dentro do mês. Na parte inferior do fragmento há os dados dos próximos pagamentos previstos

Figura 12: Fragmento de vendas.



Fonte: Autoria própria.

Após a construção das telas e escrita do código relacionado a elas, foi desenvolvida a parte de notificação do sistema. Para isso foi utilizada a API *NotificationCompat*, que serve para criar notificações. Contudo, para poder criá-las mesmo sem a aplicação aberta, foi utilizada a API *WorkManager*. Essa API serve para executar um código após um tempo definido no sistema.

O *WorkManager* não possui precisão de tempo para acionar o código, isso porque ele disputa aquele tempo com outras aplicações. Como o *WorkManager* é impreciso, e poderia começar a disparar a notificação tarde demais em um dia, ele foi programado para acionar o código que verifica se o sistema deve notificar o pagamento, a cada 12 horas, desde o momento que o aplicativo foi aberto pela primeira vez.

Figura 13: Imagem das *Activitys* de criação de venda.

The image shows a mobile application interface for creating a sale. It is divided into two main vertical panels. The left panel, titled 'Selecione o Cliente', features a search bar with the placeholder text 'Pesquise o nome do Cliente'. Below the search bar are eight input fields, each containing the text 'nome\_do\_cliente'. At the bottom of this panel is a blue button labeled 'CANCELAR'. The right panel, titled 'Produtos selecionados:', displays a list of three items. Each item consists of a small icon, the text 'nome\_produto', and the price 'RS 200,00' and 'Quantidade: 5'. Below this list is a section titled 'Selecione os produtos:' with another search bar 'Pesquise o nome do produto'. Underneath are six more items, each with a different icon, the text 'nome\_produto', and the price 'RS 200,00' and 'Quantidade: 5'. At the bottom of the right panel are two blue buttons: 'SALVAR' and 'VOLTAR'.

Fonte: Autoria própria.

Figura 14: Fragmento de início.

The image shows a mobile application interface for a financial summary. At the top, there is a purple icon of a document with an upward arrow and the text 'Salário Debit'. Below this, there is a table with three rows: 'Lucro do mês: R\$ 200,00', 'Atrasado: R\$ 100,00', and 'Lucro Previsto: R\$ 300,00'. Below the table is a section titled 'PROXIMOS PAGAMENTOS:'. This section contains two entries. The first entry shows 'nome do cliente', 'Valor Total: R\$ 200,00', 'Data da compra: 12/12/212', and 'Data prevista: 12/12/212', with a blue 'PAGO' button to the right. The second entry shows 'nome\_do\_cliente', 'Valor Total: R\$ 200,00', 'Data da compra: 12/12/212', and 'Data prevista: 12/12/212', also with a blue 'PAGO' button to the right.

Fonte: Autoria própria.

### 5.3 TESTES

Durante todo o processo de desenvolvimento foram feitos diversos testes funcionais no aplicativo, executando e testando manualmente as suas funcionalidades. Para executar a aplicação em modo de teste foi necessário fazer o download de um driver USB do fabricante original do celular utilizado (ANDROID DEVELOPERS, 2018).<sup>1</sup>

Também foram feitos testes unitários, porém, devido à natureza do aplicativo e do desenvolvimento Android, que utiliza *Context*, uma classe abstrata que permite acesso a recursos e classes específicas de aplicativo e que só é acessada durante a sua execução. Por isso os testes unitários foram usados em funções mais simples do sistema como cálculos dos valores das vendas.

Para testar funções mais complexas do aplicativo, foram feitos os testes instrumentados, executando-os não através do emulador do Android Studio, mas por meio de um smartphone. Com este teste, foi possível o acesso a classe *Context*, o que permitiu fazer testes com as funções mais importantes do aplicativo. Estes testes foram feitos principalmente com o as funções do banco de dados, para garantir que todos os dados estavam sendo guardados e recuperados de forma correta. Conforme foram feitos os testes os erros detectados foram corrigidos.

Após a finalização dos testes, o aplicativo foi enviado via e-mail para os autônomos que tinham o desejo de testá-lo e em seguida para grupos de vendedores autônomos. Junto, foi enviado um novo questionário para descobrir a impressão dos usuários sobre o aplicativo. Este questionário está incluído no Apêndice F.

---

<sup>1</sup> Para mais informações sobre o driver específico para cada marca de celular acesse: <https://developer.android.com/training/basics/firstapp/running-app?hl=pt-br>

## 6 RESULTADOS

Nesse capítulo será apresentado os resultados obtidos através da pesquisa com os usuários finais. A pesquisa foi realizada através de um questionário quali-quantitativo disponível no Apêndice F que foi aplicado a usuários que testaram o sistema. Neste momento, aplicativo não está disponível na loja de aplicativos do Google, tendo sido disponibilizado para download no Google Drive por meio de link de compartilhamento. Este link, foi enviado por e-mail para os vendedores autônomos que indicaram interesse em testar o aplicativo no levantamento de requisitos, em grupos de vendedores de cosméticos e, buscando maximizar o número de respondentes, também com pessoas fora do ramo de vendedores autônomos. Ao todo a abordagem permitiu coletar 1 resposta entre profissionais e 12 respostas do público geral

O questionário de avaliação do aplicativo consistiu em 12 questões objetivas de caráter quantitativo e uma questão opcional destinada a sugestões para o aplicativo, que será mais bem abordada durante o capítulo de trabalhos futuros. As questões quantitativas foram separadas em: três perguntas demográficas, que buscam informações de quem está respondendo (idade, sexo e o que vende), cinco perguntas sobre UX, e três perguntas gerais sobre o aplicativo.

### 6.1 ELABORAÇÃO DO QUESTIONÁRIO

O questionário começou com as 3 perguntas demográficas que visam identificar o perfil dos respondentes com indagações sobre sua idade, sexo e o tipo de vendas com que trabalha.

Além dessas, há cinco perguntas objetivas relacionadas a UX buscaram identificar a qualidade da experiência do usuário utilizando o aplicativo. Para isso, elaboraram-se perguntas relacionadas a cinco das dez heurísticas de Nielsen, conforme mostra o Quadro 3.

Quadro 3: Perguntas sobre UX do questionário de satisfação.

Heurística	Pergunta
Visibilidade de status do sistema	As funcionalidades do sistema estão dispostas de forma visível e de fácil utilização?
Consistência e padronização	O quanto a consistência da disposição dos elementos na tela do sistema (botões, caixas de texto etc.) facilitou o seu uso do sistema?
Prevenção de erros	Quão provável você acha que pode de apagar algum dado importante do sistema sem querer?
Eficiência e flexibilidade de uso	Você aprendeu a usar o aplicativo em quanto tempo?
Estética e design minimalista	O que você achou da estética do aplicativo?

Fonte: Autoria própria.

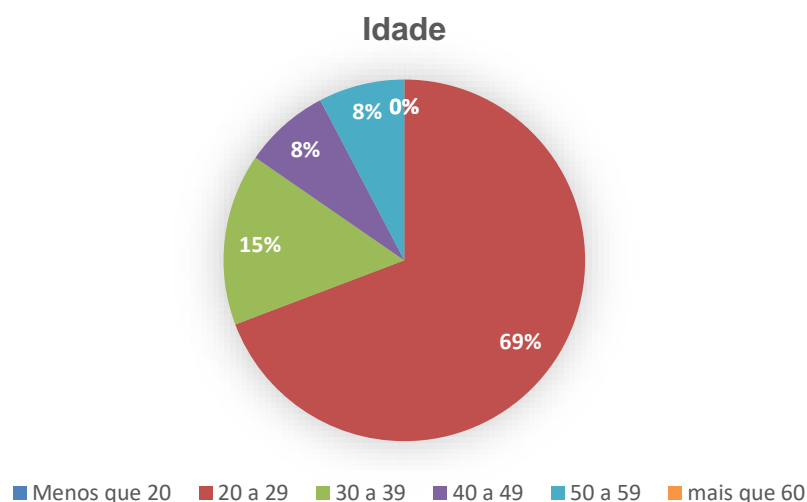
Nestas perguntas foi utilizada a escala de Likert, que é a escala onde o respondente diz seu grau de concordância ou discordância sobre algo numa escala de um (1) a cinco (5) (AGUIAR, et al, 2011), com o valor um (1) correspondendo a uma má percepção do usuário a respeito do sistema para a respectiva questão e cinco (5) correspondendo a uma boa percepção. Não foram feitas questões relacionadas a todas as heurísticas para evitar de tornar o questionário muito longo. Essas heurísticas foram escolhidas por serem consideradas as mais importantes para este sistema em virtude de sua proposta e público-alvo.

Na seção final do questionário foi perguntado qual a impressão dos usuários do aplicativo como um todo e se eles acham que o *Seller's Book* pode substituir a agenda ou o livro de anotações do vendedor, além de indagar se ele recomendaria o aplicativo para outros. A única pergunta qualitativa buscou coletar sugestões para trabalhos futuros no aplicativo, para o aprimoramento do mesmo.

## 6.2 ANÁLISE DE RESULTADOS

Ao analisar os dados demográfico notou-se que o público respondente do segundo questionário é diferente do público do questionário original. Isso ocorreu por causa da baixa adesão do público-alvo do aplicativo à testagem e resposta ao questionário, de forma que a pesquisa teve que recorrer a respondentes diferentes do grupo original. O grupo original de participantes era formado, na maioria, por mulheres entre 40 e 50 anos (APÊNDICE B). Já o grupo da pesquisa de satisfação é composto por 13 pessoas, na maioria homens de 20 a 30 anos como pode ser visto na Figura 15 e na Figura 16. Dentro deste público houve também alguns que se declararam vendedores como mostra o gráfico da Figura 17.

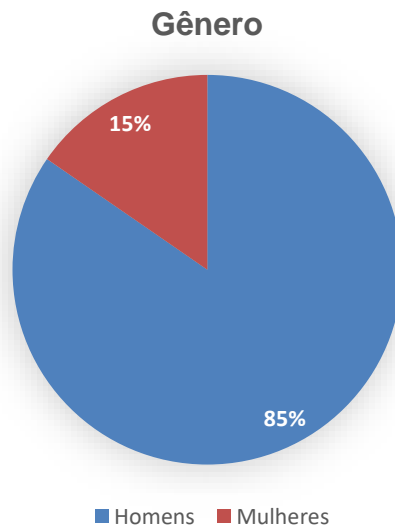
Figura 15: Idade dos respondentes do questionário de satisfação.



Fonte: Autoria própria.

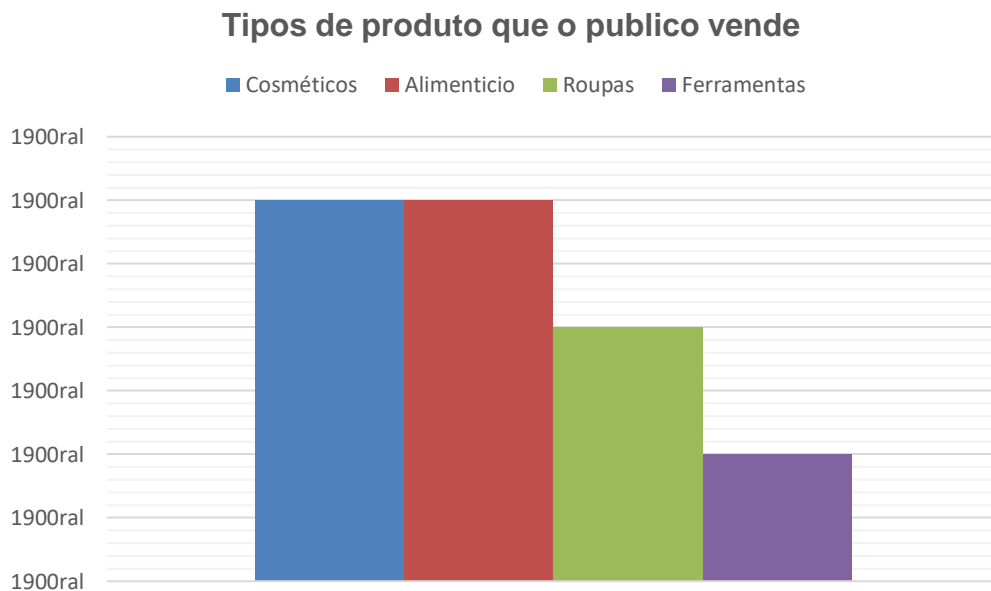
Apesar de haver diferença no perfil dos grupos de respondentes, pode-se afirmar que, em se tratando de experiência do usuário e sistema, a sua avaliação é válida em razão das métricas adotadas. Além disso os resultados adquiridos neste questionário poderão ser confrontados com os obtidos pela testagem com o público-alvo uma vez que haja maior adesão.

Figura 16: Gênero dos respondentes.



Fonte: Autoria própria.

Figura 17: Tipos de produtos vendidos pelos respondentes.



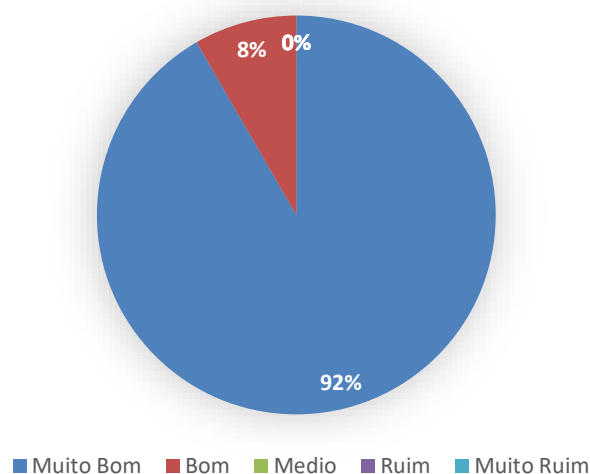
Fonte: Autoria própria.

Para cada pergunta objetiva utilizando a escala de Likert foi criado um gráfico de pizza ou de colunas para representar as respostas dos usuários. Para a primeira pergunta, o enunciado e respostas estão ilustrados no gráfico da Figura 18.



Figura 18: Resultados da primeira pergunta sobre UX.

**As funcionalidades do sistema estão dispostas de forma visível e de fácil utilização?**



Fonte: Autoria própria.

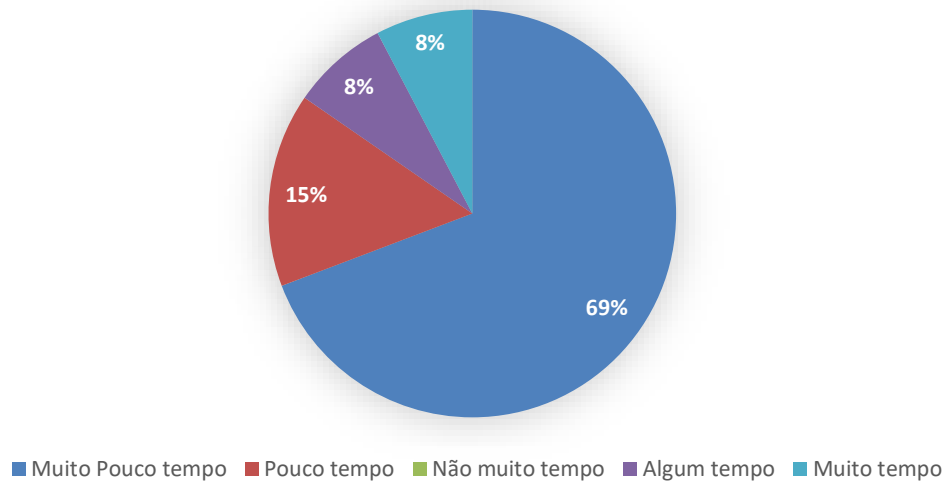
Através dessa pergunta foi possível notar que os usuários não tiveram dificuldades com a utilização do sistema. Isso indica que, seguindo os princípios do UX, as posições dos componentes estavam adequadas e bem visíveis, além de que as funcionalidades se comportaram da maneira correta e que os usuários não tiveram problemas em relação ao uso das mesmas.

A pergunta seguinte, teve relação com saber o quão rápido os usuários aprenderam a usar o sistema. As respostas dessa pergunta estão dispostas no gráfico da Figura 19. Com essas respostas, notou-se que houve um pouco mais de dificuldade no aprendizado do aplicativo. Isso pode ter relação com as listas inicialmente vazias do aplicativo, que não davam pistas sobre o que fazer naquela tela, exceto pelo botão de “+” indicando poder adicionar um produto, venda ou cliente, e pela falta de mensagens de ajuda no sistema que informem ao usuário como ele pode cumprir seus objetivos no aplicativo. Esses recursos não foram implementados durante o desenvolvimento devido a priorização de outros.

A terceira pergunta sobre UX, que teve relação com a estética ou beleza do sistema. Apesar desta área ser bastante subjetiva, o questionário mostrou que o aplicativo possui uma interface agradável de se usar, como revelam os resultados encontrados na Figura 20. O motivo para as notas menores pode estar ligado as preferencias particulares de cada usuário. Porém, como a maioria dos respondentes reagiu positivamente, pode-se afirmar que a interface é um ponto positivo do aplicativo.

Figura 19: Gráfico relativo a segunda pergunta sobre UX.

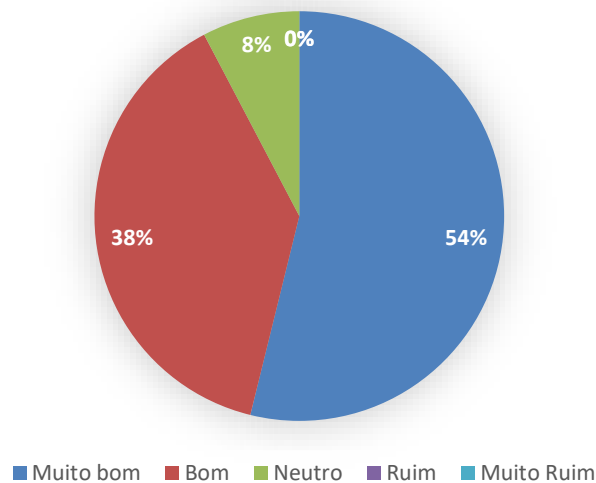
### Você aprendeu a usar o aplicativo em quanto tempo?



Fonte: Autoria própria.

Figura 20: Gráfico relativo à terceira pergunta sobre UX.

### O que você achou da estética do aplicativo?



Fonte: Autoria própria.

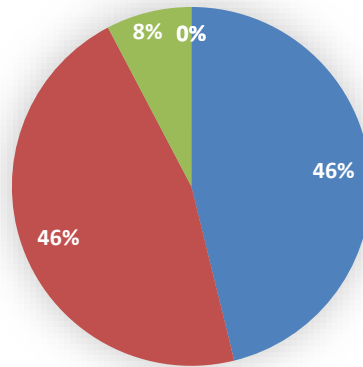
A pergunta com relação a heurística de prevenção de erros teve respostas mistas, como pode ser visto no gráfico da Figura 21. A possível razão para essa reação dos usuários pode ser devido ao aplicativo não possuir outras formas de prevenção de erros além do questionamento antes da exclusão ou modificação de dados, como por exemplo, um pop-up de “desfazer” aparecer após a exclusão, porém essa opção deverá ser estudada mais a fundo para encontrar alternativas mais adequadas ao público-alvo.

Por fim, a última pergunta sobre UX do questionário tem relação com a consistência e os padrões adotados pelo sistema. O gráfico das repostas desta questão pode ser visto na Figura 22. Essa pergunta buscou saber se o usuário ficou satisfeito com a disposição dos componentes

na tela. Ao desenvolver o sistema, buscou-se manter os padrões entre as telas para auxiliar o usuário a identificar os componentes nas mesmas.

Figura 21: Gráfico relativo a quarta pergunta sobre UX.

**Quão provável você acha que pode de apagar algum dado importante do sistema sem querer?**

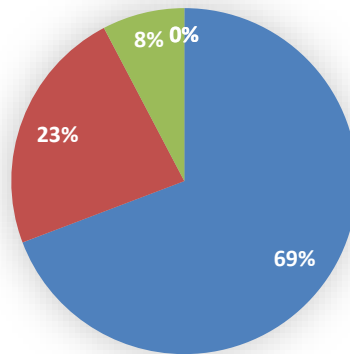


■ Muito pouco provável ■ Pouco provável ■ Não muito provável ■ Provável ■ Muito provável

Fonte: Autoria própria.

Figura 22: Gráfico relativo à quinta pergunta sobre UX

**O quanto a consistência da disposição dos elementos na tela do sistema facilitou o seu uso do sistema?**



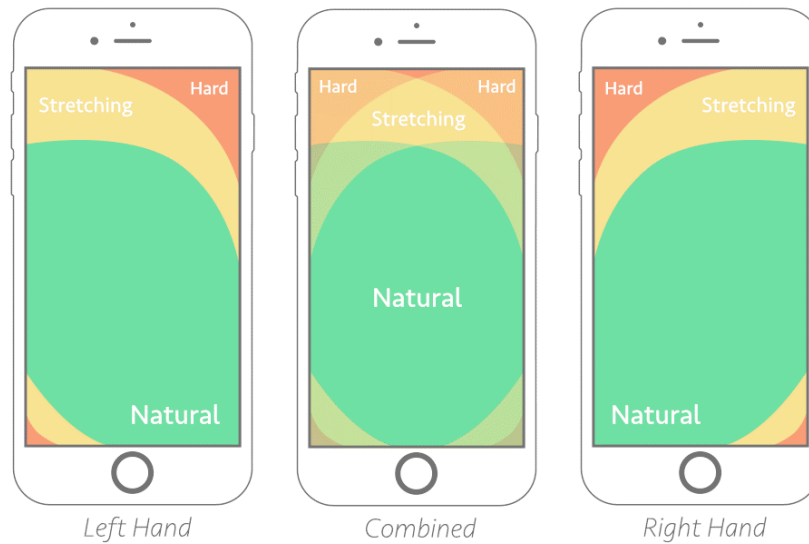
■ Muito bom ■ Bom ■ Neutro ■ Ruim ■ Muito ruim

Fonte: Autoria própria.

Para melhorar o aspecto do sistema ilustrado pela Figura 22, é preciso melhorar a disposição dos componentes do sistema. Isto pode ser conseguido pela adoção do mapeamento de zonas de toque ilustrado na Figura 23. Isso porque, de fato, durante o desenvolvimento, buscou-se tornar a interface mais visível e fácil de entender, o que pode ter deixado em segundo plano a preocupação com o posicionamento dos componentes da tela.

Comparando o resultado desta questão com os vistos na Figura 18 e Figura 19, nota-se que o foco na visibilidade dos componentes foi bem avaliado. Porém com o foco reduzido no posicionamento dos componentes do sistema, o sistema tornou-se menos intuitivo, afetando o tempo de aprendizado dos usuários.

Figura 23: Mapeamento das zonas de mais fácil toque da tela

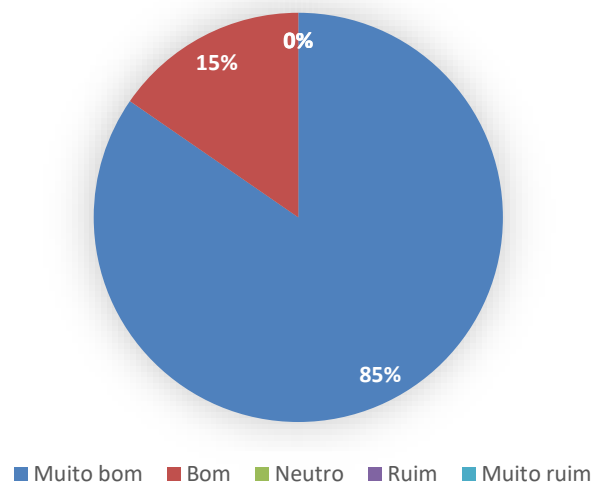


Fonte: Samantha Ingram (2016)

Durante a última seção do questionário foram realizadas perguntas mais gerais sobre o aplicativo, começando com a impressão que os usuários tiveram do aplicativo como um todo. Como mostra o gráfico da Figura 24, o aplicativo teve uma boa aceitação dos usuários, com a maioria o sinalizando como muito bom.

Figura 24: Gráfico relativo à questão de gosto do aplicativo.

**O que você achou do aplicativo como um todo?**

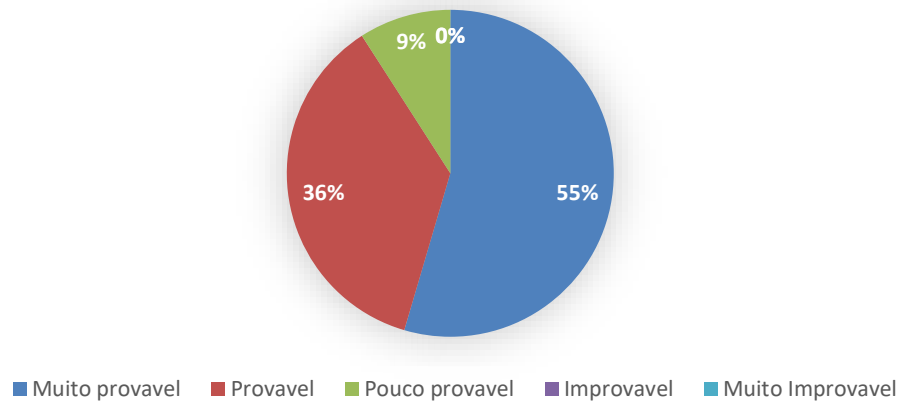


Fonte: Autoria própria.

A segunda pergunta desta seção buscou saber se o usuário acredita que o aplicativo poderia substituir sua agenda ou livro de anotações. Conforme demonstrado na Figura 25, não houve tanta certeza de que esse aplicativo poderia substituí-los. Porém, para ser mais conclusivo quanto a essa questão será necessário realizar o questionário com vendedores autônomos. Isso porque, seu resultado atual vem de um público que pode não saber opinar a respeito, por não atuar com vendas.

Figura 25: Gráfico relativo à questão sobre a substituição dos cadernos e agendas.

### Você acha que ele pode te ajudar a substituir a agenda/ livro de anotações?

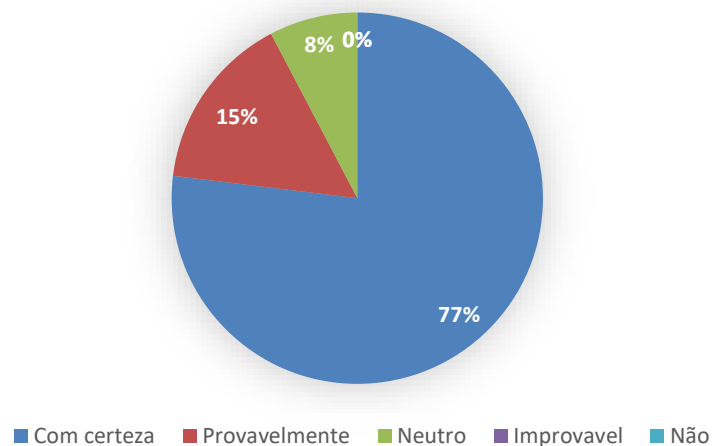


Fonte 1: Autoria Própria.

A última questão quantitativa da pesquisa indagou se os usuários recomendariam o aplicativo para outras pessoas. Tendo a maioria afirmado que recomendaria, pode-se dizer que o aplicativo obteve sucesso, o que pode levar mais pessoas a utilizá-lo após sua disponibilização. Para melhor visualizar os dados, pode-se consultar o gráfico da Figura 26.

Figura 26: Gráfico relativo à questão de recomendação.

### Você recomendaria esse aplicativo para outra pessoa?



Fonte: Autoria Própria.

Após as questões objetivas, havia um último item do questionário na forma de um espaço para os usuários darem suas sugestões em relação ao aplicativo. Houve três respostas nesta questão. Uma solicitando a sincronização com os contatos do celular, de forma que o usuário possa usar contatos de sua lista no telefone para a lista de clientes cadastrados no aplicativo. Outra com a sugestão de um aplicativo diferente feito para os clientes, que se comunique com o *Seller's Book*. Além dessas duas sugestões, o espaço também foi usado para elogiar o aplicativo. Também foi reportado um bug visual na tela de adição de produtos, onde ao selecionar os tipos com os nomes mais longos, aparecia apenas a última letra na caixa de seleção.

Apesar dos resultados positivos, é importante ressaltar que, como forma de confirmar os resultados obtidos, este questionário deverá ser reaplicado no público e demografia desejados e o aplicativo estiver com as demais funcionalidades propostas implementadas, bem como as correções que foram apontadas com os testes com os usuários.

## 7 CONCLUSÕES

O *Seller's Book* buscou auxiliar os vendedores autônomos nas suas tarefas, dando a oportunidade de manter o controle de seu estoque e de identificar os compradores dos produtos desses profissionais de forma rápida e simples, o que contrasta com o fato percebido durante o projeto, de que grande parte dos vendedores autônomos utilizam agendas ou cadernos para gerir suas anotações de estoque e vendas. Através da análise do levantamento de requisitos, percebeu-se que estes vendedores possuem interesse no uso de um aplicativo para celular que lhes permita cuidar de seu estoque. Ainda que, apesar de haver outros aplicativos semelhantes, poucos dos entrevistados os usavam, o que pode indicar resistência ou desconhecimento sobre esses aplicativos.

Sabendo disso, esse projeto teve o objetivo de desenvolver um aplicativo que permita que esses vendedores possam cuidar de seu estoque de forma simples. Para isso foi feito um estudo de UX para poder criar uma interface satisfatória para o usuário e a partir disso começar o desenvolvimento. As particularidades do desenvolvimento Android tornaram o processo de desenvolvimento mais lento, o que forçou a abdicação de funcionalidades periféricas para o funcionamento do sistema, focando-se nas funções essenciais para o correto funcionamento do sistema.

Durante o desenvolvimento, foram feitos testes unitários, instrumentais e funcionais para verificar se o sistema estava funcionando corretamente. Após os testes finais e correções no sistema, o *Seller's Book* foi disponibilizado para testes do público-alvo. A divulgação contou com o download disponibilizado através de um link do Google Drive com o instalador do aplicativo (apk) enviado via e-mail para os interessados durante o levantamento de requisitos, sendo também enviado em grupos de vendedores.

Apesar do interesse, ainda parece haver resistência desse público, o que foi notado durante a pesquisa de satisfação do aplicativo, onde houve poucas respostas do público-alvo. Fato que acabou levando a se recorrer a um público diferente do público-alvo do levantamento de requisitos. Com o público que testou o aplicativo, foi aplicado um formulário de satisfação para avaliar a recepção do aplicativo. Os resultados mostraram boa aceitação, apesar do aplicativo ainda precisar de melhorias. Concluído o projeto, acreditasse que foram cumpridos os objetivos do trabalho.

## 7.1 TRABALHOS FUTUROS

Como trabalhos futuros, destacam-se primeiramente os casos de uso não implementados durante o desenvolvimento deste projeto e a resolução de bugs encontrados durante a validação do sistema. No caso, a criação de PDFs com o catálogo do sistema para envio aos clientes, além das funcionalidades de promoção e criação de kits, que não eram prioritárias sem a implementação da geração de PDF.

Concluída esta etapa, a preocupação seguinte será melhorar a UX e UI do sistema, adicionando mais informações na tela para guiar o usuário e tornando o visual do sistema mais agradável. Também se prevê a adição de uma função de preenchimento automático para campos específicos da adição de produtos, assim como, a possibilidade de o usuário adicionar seu próprio tipo de produto à lista de tipos.

Após estas implementações no sistema, poderá ser feita novamente a pesquisa de satisfação com o público-alvo. Em seguida, se houver resultados positivos na pesquisa, o aplicativo poderá ser disponibilizado o *download* para o público geral, através da Google Play Store.

Outra preocupação deverá ser a adição de outras informações importantes para o vendedor, como por exemplo o valor total em estoque, gráficos informando o lucro anual e campos para o custo do produto e para a validade do mesmo. A partir disso, é importante investigar a criação de uma forma de backup do estoque, possivelmente utilizando o Google Drive, para guardar estes dados, já servindo como uma forma de transferência de dados entre dispositivos.

Ainda, seguindo as sugestões feitas no questionário de satisfação do aplicativo, seria uma boa adição a sincronia do aplicativo com a lista de contatos para facilitar o preenchimento de dados do cliente, uma função possível já que o Android possibilita acesso a essa parte do sistema. Porém a segunda sugestão, em relação a criação de um aplicativo para o cliente é menos viável, já que esse contato entre cliente e vendedor necessitaria de algum tipo de servidor, gerando custo e indo contra a ideia do sistema de não gerar custos aos vendedores. Concluídas essas melhorias de funcionalidade, o sistema voltaria a focar em melhorias de UX, esse seria um processo contínuo, sempre em busca do aperfeiçoamento dessa UX “perfeita”.



## REFERÊNCIAS

ABLESON, Frank et al. Android em ação. Elsevier Brasil, 2012.

AGUIAR, Bernardo, et al. Uso da escala likert na análise de jogos. Salvador: SBC- Proceedings of SBGames Anais, v. 7, n. 2, 2011.

ANDROID DEVELOPERS. Introdução as Activitys. Disponível em: <<https://developer.android.com/guide/components/activities/intro-activities?hl=pt-br>>. Acesso em: 12 de julho de 2022.

ANDROID DEVELOPERS. Executar o seu app. Disponível em: <<https://developer.android.com/training/basics/firstapp/running-app?hl=pt-br>>. Acesso em: 9 de agosto de 2022

ANDROID DEVELOPERS. Fragmentos. Disponível em: <<https://developer.android.com/guide/components/fragments?hl=pt-br>>. Acesso em: 12 de julho de 2022.

ANDROID STUDIO, Ambiente de desenvolvimento android. Versao 3.6. 2013. Disponível em: <<https://developer.android.com/studio>>. Acesso em: 18 de julho de 2022.

BARBOSA, Ellen Francine et al. Introdução ao teste de software. Minicurso apresentado no XIV Simpósio Brasileiro de Engenharia de Software (SBES 2000), 2000.

BOOCH, Grady et al. UML: guia do usuário. Elsevier Brasil, 2006.

COSTA, Sergio Ricardo Abreu. Análise do processamento da informação manual x automatizada em vendas autônomas no varejo. UFMA Disponível em <<https://monografias.ufma.br/jspui/bitstream/123456789/491/1/TCC%20S%c3%a9rgio%20Ricardo%20Abreu%20Costa.pdf>>. Acesso em: 01 de abr. 2022.

DATE, Christopher J. Introdução a sistemas de bancos de dados. Elsevier Brasil, 2004.

DE OLIVEIRA FILHO, Daniel C. Um passo a passo para a elaboração do diagrama de caso de uso da UML. 2011.

DEVMEDIA. Orientações básicas na elaboração de um diagrama de classes. Disponível em: <<https://www.devmedia.com.br/orientacoes-basicas-na-elaboracao-de-um-diagrama-de-classes/37224>>. Acesso em: 14 de junho de 2022.

DEVMEDIA. MER e DER: Modelagem de Bancos de Dados. Disponível em: <<https://www.devmedia.com.br/mer-e-der-modelagem-de-bancos-de-dados/14332>>. Acesso em: 14 de junho de 2022.

DIGITAL HOUSE. UX x UI qual a diferença e como as áreas se completam. Disponível em: <<https://www.digitalhouse.com/br/blog/diferenca-entre-ux-e-ui/>>. Acesso em: 20 de junho de 2022.

DIGITÉ. O que é teste de unidade? Disponível em: <<https://www.digite.com/pt-br/agile/testes-unitarios/#:~:text=O%20teste%20unit%C3%A1rio%20consiste%20em,em%201%C3%ADnguas%20processuais%20e%20funcionais>>. Acesso em: 01 de abr. 2022.

DO PRADO, Lucas Sciencia; GUISSONI, Leandro Angotti; KEHDI, Marco Tulio. O vendedor na disrupção digital. GV-EXECUTIVO, v. 17, n. 5, p. 12-16, 2018.

ELMASRI, Ramez et al. Sistemas de banco de dados. 2005.

EWALLY. O que é back-end. Disponível em: <<https://www.ewally.com.br/blog/ajudando-sua-empresa/backend/>>. Acesso em: 01 de abr. 2022

FRANCK, Kewry Mariobo; PEREIRA, Robson Fernandes; DANTAS FILHO, Jerônimo Vieira. Diagrama Entidade-Relacionamento: uma ferramenta para modelagem de dados conceituais em Engenharia de Software. Research, Society and Development, v. 10, n. 8, p. e49510817776-e49510817776, 2021.

FOWLER, Martin. UML Essencial: um breve guia para linguagem padrão. Bookman editora, 2014.

FURASTÉ, Pedro Augusto. Normas Técnicas para o Trabalho Científico: explicitação das normas da ABNT. Porto Alegre: [s.n.], 2002. p. 49-56.

G1. Telefone celular aumenta renda de profissionais autônomos no Brasil, G1, 2008. Disponível em: <<https://g1.globo.com/Noticias/Tecnologia/0,,MUL862055-6174,00-TELEFONE+CELULAR+AUMENTA+RENDA+DE+PROFISSIONAIS+AUTONOMOS+NO+BRASIL.html>> Acesso em: 31 de mar. 2022

GERHARDT, Tatiana Engel; SILVEIRA, Denise Tolfo (Org.). Métodos de pesquisa. Porto Alegre: UFRGS, 2009. Disponível em: <<http://www.ufrgs.br/cursopgdr/downloadsSerie/derad005.pdf>>. Acesso em: 14 de dez. 2021

GOOGLE. Android developer,2021. Conheça o Android Studio. Disponível em: <<https://developer.android.com/studio/intro?hl=pt>>. Acesso em: 10 de julho de 2022.

GUEDES, Gilleanes TA. UML 2-Uma abordagem prática. Novatec Editora, 2018.

HERVÉ, Desireé. A importância do controle de estoque. AgoraOS, 2017. Disponível em: <<https://www.agoraos.com.br/blog/controle-de-estoque/>>. Acesso em 31 de mar.2022.

INGRAM, Samantha. The Thumb Zone: Designing For Mobile Users. Smashing Magazine, 2016. Disponível em: <<https://www.smashingmagazine.com/2016/09/the-thumb-zone-designing-for-mobile-users/>>. Acesso em: 19 de julho de 2022.

KNECHTEL, Maria do Rosário. Metodologia da pesquisa em educação: uma abordagem teórico-prática dialogada. Curitiba: Inter saberes, 2014.

KREIBICH, Jay. Using SQLite. " O'Reilly Media, Inc.", 2010.

KYTE, Aplicativo de Vendas, Loja Virtual e Controle de Estoque. Versão 1.26.1. 2022. Disponível em: <<https://www.kyte.com.br/>>. Acesso em: 18 de julho de 2022.

LIMA, Vinicius. Número de trabalhadores autônomos bate recorde no país em 2021. DHoje Interior, 2021. Disponível em: <<https://dhojeinterior.com.br/numero-de-trabalhadores-autonomos-bate-recorde-no-pais-em-2021/>>. Acesso em 14 de dez. 2021

MAGALHÃES, Rafael L.; NETO, Michelle MF. ApreNDER: Ferramenta de apoio à construção de diagrama entidade relacionamento para deficientes visuais. In: Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE). 2010.

MARTINS, Aline dos Santos et al. Diretrizes para a construção de casos de uso eficazes. Disponível em: <[https://seer.uscs.edu.br/index.php/revista\\_informatica\\_aplicada/article/download/293/259/1333](https://seer.uscs.edu.br/index.php/revista_informatica_aplicada/article/download/293/259/1333)>. Acesso em: 7 de julho de 2022

MARCUS, Luiz. Luiz Marcus. Executando Teste Instrumentado com Framework Espresso, 2016. Disponível em: <<https://luizmarcus.com/android/executando-teste-instrumentado-com-o-framework-espresso/>>. Acesso em: 12 de julho de 2022.

MATERIAL DESIGN. Introdução ao Material Design. Disponível em: <<https://material.io/design/introduction>>. Acesso em: 12 de julho de 2022.

MATOS, Deivid G. O que é um banco de dados, 2013. Disponível em: <<http://fabrica.ms.senac.br/2013/06/o-que-e-um-banco-de-dados/>>. Acesso em: 16 de julho de 2022.

MELO, Diego. O que é Java. Technoblog, 2020. Disponível em: <<https://tecnoblog.net/responde/o-que-e-java-guia-para-iniciantes/>>. Acesso em: 01 de abr. 2022

MENDES, Douglas Rocha. Redes de computadores: teoria e prática. Novatec Editora, 2020.

MIRANDA, Diogo. Medium. Teste unitário e Qualidade de software, 2017. Disponível em: <<https://medium.com/assertqualityassurance/teste-unit%C3%A1rio-e-qualidade-de-software-acce7b9c537>>. Acesso em: 11 de julho de 2022.

MICROSOFT EXCELL, Ferramenta de planilhas da Microsoft. Versão 16.0. 2016. Disponível em: <<https://www.microsoft.com/pt-br/microsoft-365/excel>>. Acesso em: 18 de julho de 2022.

NETO, Arilo; CLAUDIO, Dias. Introdução a teste de Software. Engenharia de Software Magazine, v. 1, p. 22, 2007.

NIELSEN, Jakob. 10 Usability Heuristics for interface Design. NN Group, 2020. Disponível em: <<https://www.nngroup.com/articles/ten-usability-heuristics/>>. Acesso em: 20 de julho de 2022.

NIELSEN, Jakob. Usefulness, Utility, Usability: 3 Goals of UX Design (Jakob Nielsen). Youtube, 2018. Disponível em: <[https://www.youtube.com/watch?v=VwgZtqTQzg8&list=PLJOFJ3Ok\\_idsnPXXdJ9FpDzqIyWRIwGS-](https://www.youtube.com/watch?v=VwgZtqTQzg8&list=PLJOFJ3Ok_idsnPXXdJ9FpDzqIyWRIwGS-)>. Acesso em: 16 de junho de 2022.

NORMAN, Donald. Don Norman: O Termo "UX". Youtube, 2017. Disponível em: <[https://www.youtube.com/watch?v=9BdtGjoIN4E&list=PLJOFJ3Ok\\_idv\\_6hQGNt23xVXuQwKFmfxG](https://www.youtube.com/watch?v=9BdtGjoIN4E&list=PLJOFJ3Ok_idv_6hQGNt23xVXuQwKFmfxG)>. Acesso em: 16 de junho de 2022.

NORMAN, Don, NIELSEN, Jacob. The Definition of User Experience (UX). Nielsen Norman Group, 2018. Disponível em: <<https://www.nngroup.com/articles/definition-user-experience/>>. Acesso em: 20 de junho de 2022.

OLIVEIRA, Marcos. Terminal Root, 2019. Disponível em: <<https://terminalroot.com.br/2019/10/java-vs-kotlin-quais-sao-as-principais-diferencas-entre-elas.html>>. Acesso em: 11 de julho de 2022.

OWENS, Mike. The definitive guide to SQLite. Apress, 2006.

PEREIRA, Lucio Camilo Oliva; DA SILVA, Michel Lourenço. Android para desenvolvedores. Brasport, 2009.

SANTOS, RAFAEL. Introdução à programação orientada a objetos usando java. 2ª Edição. Rio de Janeiro: Elsevier, 2013.

SOARES, João Paulo. Principais técnicas de testes funcionais. TreinaWeb, 2021. Disponível em: <<https://www.treinaweb.com.br/blog/principais-tecnicas-de-testes-funcionais>>. Acesso em: 01 de abr. 2022.

SOMMERVILLE, Iam. Engenharia de Software. 9. ed. São Paulo: Pearson, 2011. 529 p. Disponível em: <<https://www.facom.ufu.br/~william/Disciplinas%202018-2/BSI-GSI030-EngenhariaSoftware/Livro/engenhariaSoftwareSommerville.pdf>>. Acesso em: 22 de julho de 2022.

STAT COUNTER, Stat Counter. Gráfico de sistemas operacionais mobile. Disponível em: <<https://gs.statcounter.com/os-market-share/mobile/worldwide/#yearly-2009-2022>>. Acesso em: 10 de julho de 2022

SUPER REVENDEDORES, Aplicativo de controle de vendas e estoque. Versão 6.0.9. 2022. Disponível em: <<https://www.superrevendedores.com.br/>>. Acesso em: 18 de julho de 2022.

TECHTUDO. Conheça o Android, o sistema operacional móvel do Google. Disponível em: <<https://www.techtudo.com.br/tudo-sobre/android/>>. Acesso em: 01 de abr. 2022

TECHTUDO. Conheça o Android, o sistema operacional móvel do Google. Disponível em: <<https://www.techtudo.com.br/tudo-sobre/android/>>. Acesso em: 14 de junho de 2022.

TEIXEIRA, Fabricio. Introdução e boas práticas em UX Design. Editora Casa do Código, 2014.

VENTURA, Plínio. O que é UML? (Unified Modeling Language). Disponível em: <<https://www.ateomomento.com.br/diagramas-uml/>>. Acesso em: 14 de junho de 2022.

VIEIRA, Rodrigo. UML — Diagrama de Casos de Uso. Disponível em:  
<<https://medium.com/operacionalti/uml-diagrama-de-casos-de-uso-29f4358ce4d5>>. Acesso em: 14 de junho de 2022.

VIEIRA, Ricardo. Uso da UML na especificação do sistema de informação da área de infraestrutura da UFRGS. UFRGS. Disponível em  
<<https://www.lume.ufrgs.br/bitstream/handle/10183/12389/000419993.pdf?sequence=1>>. Acesso em: 01 de abr. 2022

WAINER, Jacques. Métodos de pesquisa quantitativa e qualitativa para a Ciência da Computação. UNICAMP Disponível em:  
<[https://www.ic.unicamp.br/~wainer/cursos/1s2018/metodologia/Metodos\\_de\\_pesquisa\\_quantitativa\\_e\\_qualitativa\\_par.pdf](https://www.ic.unicamp.br/~wainer/cursos/1s2018/metodologia/Metodos_de_pesquisa_quantitativa_e_qualitativa_par.pdf)> Acesso em: 14 de dez. 2021

WIKIPEDIA. Diagrama de Casos de uso, 2011. Disponível em:  
<[https://pt.wikipedia.org/wiki/Diagrama\\_de\\_caso\\_de\\_uso](https://pt.wikipedia.org/wiki/Diagrama_de_caso_de_uso)>. Acesso em: 8 de agosto de 2022.

WORDPRESS. Tecnologia e Informação, 2009. Disponível em:  
<<https://cursobsi.wordpress.com/2008/04/28/diagrama-entidade-relacionamento/>>. Acesso em: 10 de julho de 2022.

## APÊNDICE A – LEVANTAMENTO DE REQUISITOS

# Aplicativo de controle de estoque e vendas para vendedores autônomos

Este formulário busca informações para ajudar no desenvolvimento de um aplicativo móvel para ajudar vendedores autônomos no controle de estoque e vendas de seus produtos.

**\*Obrigatório**

1. E-mail \*

2. Qual a sua idade? \*

*Marcar apenas uma oval.*

- Menos que 20 anos
- 20 a 29 anos
- 30 a 39 anos
- 40 a 49 anos
- 50 a 60 anos
- Mais que 60 anos

3. Qual o seu sexo?

*Marcar apenas uma oval.*

- Masculino
- Feminino

4. Quais tipos de produtos você vende? \*

*Marque todas que se aplicam.*

Cosméticos

Alimentícios

Roupas

Outro: \_\_\_\_\_

5. Você utiliza algum aplicativo que te ajuda a cuidar de seu estoque ou vendas? \*

*Marcar apenas uma oval.*

Sim *Pular para a pergunta 6*

Não *Pular para a pergunta 7*

### **Caso já utilize algum aplicativo**

6. Qual aplicativo você usa? \*

*Pular para a pergunta 9*

### **Caso não utilize um aplicativo**

7. Por que você não usa?

8. Você usaria um aplicativo que te ajude no controle de estoque e vendas? \*

*Marcar apenas uma oval.*

Sim

Não

Talvez



**Sugestão para o aplicativo**

9. Quais funções além do controle de vendas e de estoque você gostaria que tivesse no aplicativo? \*

*Marque todas que se aplicam.*

Geração de PDF listando produtos no aplicativo

Lembrete de Cobrança

Criação de meta de vendas

Outro: \_\_\_\_\_

10. Você gostaria de ser contatado para testar o aplicativo quando estiver pronto? \*

*Marcar apenas uma oval.*

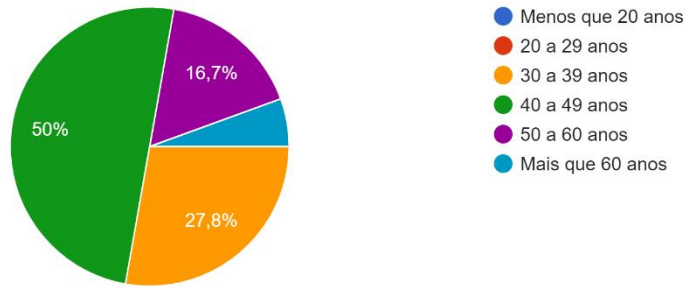
Sim

Não

## APÊNDICE B – RESULTADOS DO LEVANTAMENTO DE REQUISITOS

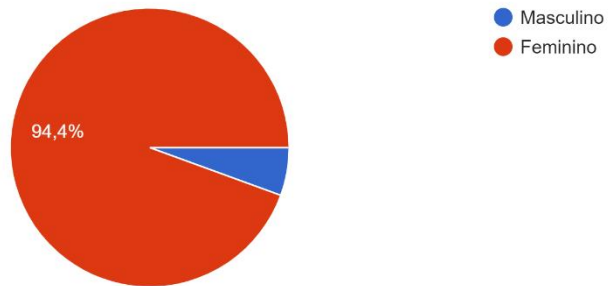
Qual a sua idade?

18 respostas



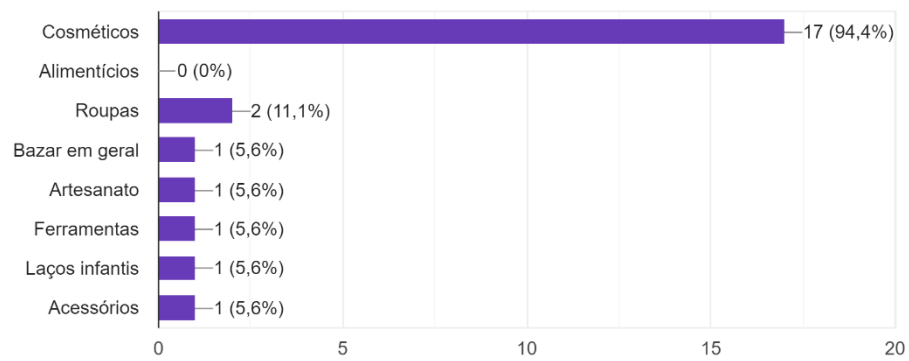
Qual o seu sexo?

18 respostas



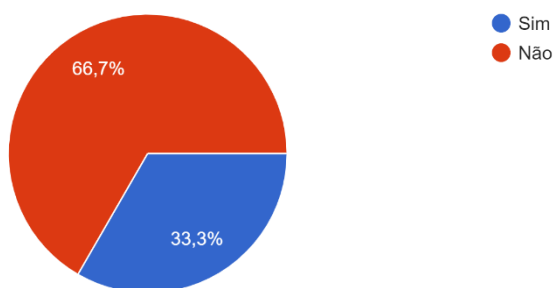
Quais tipos de produtos você vende?

18 respostas



Você utiliza algum aplicativo que te ajuda a cuidar de seu estoque ou vendas?

18 respostas



Qual

aplicativo você usa? 6 respostas

Controles de vendas e estoque para autonomia, planilha de Excel e pág seguros

Revendedores

Olx, mercado livre

Super revendedora

Kyte

Excel

Por que você não usa? 10 respostas

Porque o custo não compensa! Uso planilhas básicas do Excel

Não tenho tempo sou artesã também.

Porque não fiz ainda

Não achei necessário

Nunca procurei saber a respeito

Não tenho necessidade no momento

Não gosto muito

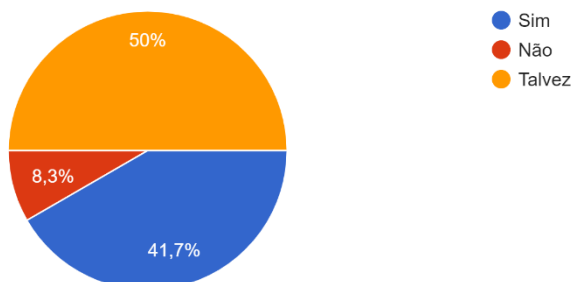
Eu acho mais fácil registrar no caderno

Acaba esquecendo sempre

Não conheço nenhum didático

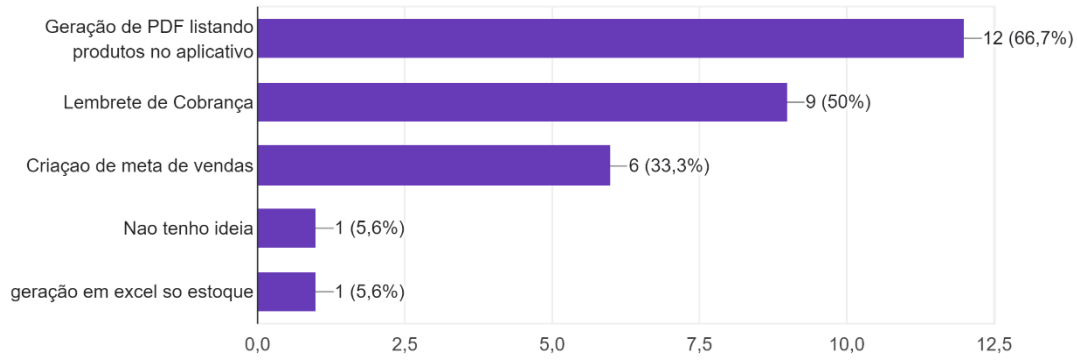
Você usaria um aplicativo que te ajude no controle de estoque e vendas?

12 respostas



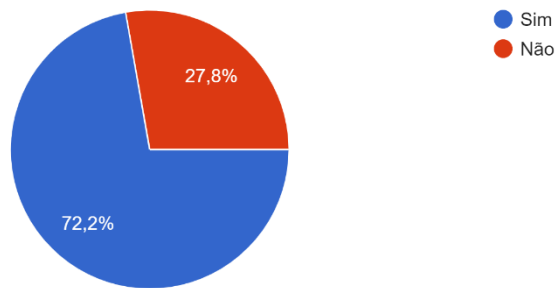
### Quais funções além do controle de vendas e de estoque você gostaria que tivesse no aplicativo?

18 respostas



### Você gostaria de ser contatado para testar o aplicativo quando estiver pronto?

18 respostas



## APÊNDICE C – REQUISITOS DO SISTEMA

### **Requisitos Funcionais:**

[RF001] O sistema deve permitir que o usuário mantenha produtos no sistema.

[RF002] O sistema deve permitir que o usuário mantenha clientes no sistema.

[RF003] O sistema deve permitir que o usuário mantenha conjuntos/pacotes de produtos.

[RF004] O sistema deve permitir que o usuário crie uma promoção de produtos.

[RF005] O sistema deve permitir que o usuário gere um PDF com os produtos cadastrados.

[RF006] O sistema deve permitir que o usuário mantenha informações da venda incluído o estado do pagamento, entre pago e uma data agendada.

[RF007] O sistema deve notificar quando o pagamento agendado não foi realizado.

[RF008] O sistema deve permitir que o usuário crie uma meta de vendas.

### **Requisitos Não-Funcionais:**

[RNF001] O sistema deve funcionar no sistema operacional Android.

[RNF002] O sistema não pode gerar custos.

[RNF003] O cadastro de produtos no sistema deve ser intuitivo e rápido.

[RNF004] O sistema deve ajudar o usuário em caso de dúvidas.

[RNF005] O sistema deve excluir vendas com mais de um ano e meio.

[RNF006] O sistema deve ser desenvolvido na linguagem Java.

[RNF007] O sistema devera apagar os dados de vendas com mais de 1 ano e meio.

## APÊNDICE D – ESPECIFICAÇÕES DE CASOS DE USO

### Especificação de casos de uso:

Este documento contém a descrição longa de cada Caso de Uso do sistema proposto.

[CdU001] Manter Produtos:

Ator Principal	Usuário
Fluxo Principal de eventos	<p>P1. O usuário deve ir para a tela de produtos.</p> <p>P2. O sistema mostra a lista de produtos.</p> <p>P3. O usuário seleciona o produto que deseja ver detalhes.</p> <p>P4. O sistema mostra os detalhes do produto.</p>
Fluxo alternativo de eventos	<p>A1. Adicionar Produto.</p> <p>A1.1. Em P3, o Usuário seleciona o botão de adicionar produto (“+ “).</p> <p>A1.2. O sistema mostra a caixa de diálogo para criação de produto.</p> <p>A1.3. O usuário preenche os dados e seleciona salvar.</p> <p>A1.4. O sistema adiciona o Produto ao banco de dados e fecha a caixa de diálogo.</p> <p>A2. Editar Produto.</p> <p>A2.1. Após P4, o usuário seleciona editar.</p> <p>A2.2. O sistema mostra a caixa de edição do produto.</p> <p>A2.3. O usuário muda os dados que deseja no produto.</p> <p>A2.4. O sistema salva as alterações do produto no banco de dados.</p> <p>A3. Remover Produto.</p> <p>A3.1. Após P4, o usuário seleciona remover produto.</p> <p>A3.2. O sistema pergunta se o usuário tem certeza.</p> <p>A3.3. O usuário seleciona sim.</p> <p>A3.4. O sistema remove o produto do banco de dados e da lista.</p> <p>A4. Busca Produto.</p> <p>A4.1. Em P3, o usuário faz uma busca utilizando a barra de busca.</p>

	<p>A4.2. O sistema mostra a lista de produtos com aquele nome ou marca.</p> <p>A4.3. Retorna a P3.</p>
Fluxos de Exceção	<p>E1. Não preenchimento de dados Obrigatórios.</p> <p>E1.1. Em A1.3 ou A2.3, o usuário deixou vazio os campos de nome, preço ou quantidade.</p> <p>E1.2. O sistema mostra que os campos não foram preenchidos.</p> <p>E1.3. Retorna para a origem.</p>

[CdU002] Manter Clientes:

Ator Principal	Usuário
Fluxo Principal de eventos	<p>P1. O usuário deve ir para a tela de clientes.</p> <p>P2. O sistema mostra a lista de clientes.</p> <p>P3. O usuário seleciona o cliente que deseja ver detalhes.</p> <p>P4. O sistema mostra os detalhes do cliente.</p>
Fluxo alternativo de eventos	<p>A1. Adicionar cliente.</p> <p>A1.1. Em P3, o Usuário seleciona o botão de adicionar cliente (“+ “).</p> <p>A1.2. O sistema mostra a caixa de diálogo para criação de cliente.</p> <p>A1.3. O usuário preenche os dados e seleciona salvar.</p> <p>A1.4. O sistema adiciona o cliente ao banco de dados e fecha a caixa de diálogo.</p> <p>A2. Editar cliente.</p> <p>A2.1. Após P4, o usuário seleciona editar.</p> <p>A2.2. O sistema mostra a caixa de edição do cliente.</p> <p>A2.3. O usuário muda os dados que deseja no cliente.</p> <p>A2.4. O sistema salva as alterações do cliente no banco de dados.</p> <p>A3. Remover cliente.</p> <p>A3.1. Após P4, o usuário seleciona remover cliente.</p> <p>A3.2. O sistema pergunta se o usuário tem certeza.</p> <p>A3.3. O usuário seleciona sim.</p> <p>A3.4. O sistema remove o cliente do banco de dados e da lista.</p>

	<p>A4. Busca Produto.</p> <p>A4.1. Em P3, o usuário faz uma busca utilizando a barra de busca.</p> <p>A4.2. O sistema mostra a lista de clientes com aquele nome.</p> <p>A4.3. Retorna a P3.</p>
Fluxos de Exceção	<p>E1. Não preenchimento de dados Obrigatórios.</p> <p>E1.1. Em A1.3 ou A2.3, o usuário deixou vazio o campo de nome.</p> <p>E1.2. O sistema mostra que os campos não foram preenchidos.</p> <p>E1.3. Retorna para a origem.</p>

[CdU003] Controle de Vendas.

Ator Principal	Usuário
Fluxo Principal de eventos	<p>P1. O usuário deve ir para a tela de vendas.</p> <p>P2. O sistema mostra a lista de vendas.</p> <p>P3. O usuário seleciona a venda que deseja ver detalhes.</p> <p>P4. O sistema mostra os detalhes da venda.</p>
Fluxo alternativo de eventos	<p>A1. Adicionar Produto.</p> <p>A1.1. Em P3, o Usuário seleciona o botão de adicionar venda (“+”).</p> <p>A1.2. O sistema abre uma tela para a seleção do cliente da venda.</p> <p>A1.3. O usuário seleciona o cliente.</p> <p>A1.4. O sistema mostra a tela para a seleção de produtos pedidos pelo cliente.</p> <p>A1.5. O usuário seleciona os produtos e quantidades, e seleciona salvar.</p> <p>A1.6. O sistema pede uma data prevista para o pagamento do(s) produto(s).</p> <p>A1.7. O usuário seleciona uma data.</p> <p>A1.8. O sistema salva a venda no banco de dados.</p> <p>A2. Pagar Venda.</p> <p>A2.1. Após P4, o usuário seleciona Pagar.</p>



	<p>A2.2. O sistema pergunta se o cliente realmente pagou.</p> <p>A2.3. O usuário seleciona que sim.</p> <p>A2.4. O sistema adiciona a data atual no pagamento da venda no banco de dados.</p> <p>A3. Remover venda.</p> <p>A3.1. Após P4, o usuário seleciona remover venda.</p> <p>A3.2. O sistema pergunta se o usuário tem certeza.</p> <p>A3.3. O usuário seleciona sim.</p> <p>A3.4. O sistema remove a venda do banco de dados e da lista.</p> <p>A4. Busca Produto.</p> <p>A4.1. Em P3, o usuário faz uma busca utilizando a barra de busca.</p> <p>A4.2. O sistema mostra a lista de vendas com o nome do cliente digitado.</p> <p>A4.3. Retorna a P3.</p>
--	--

[CdU004] Estatísticas de vendas:

Ator principal	Usuário
Fluxo Principal de eventos	<p>P1. O usuário deve ir para a tela de início.</p> <p>P2. O sistema mostra o lucro do mês, valor de pagamentos atrasados, o lucro total previsto naquele mês, e os próximos 2 clientes que devem pagar.</p>
Fluxo Alternativo	<p>A1. Pagar Venda.</p> <p>A1.1. Após P2, o usuário seleciona Pagar.</p> <p>A1.2. O sistema adiciona a data atual no pagamento da venda no banco de dados.</p>

[CdU005] Notificar Pagamento:

Ator principal	Work Manager
Pré-Requisito	Ter passado 12 horas desde o primeiro uso do aplicativo
Fluxo Principal de eventos	<p>P1. O work manager dispara uma notificação se houver clientes para pagar no dia.</p> <p>P2. Repete P1 após 12 horas.</p>

[CdU006] Manter Promoção:

Ator Principal	Usuário
Pré-requisito	Estar na tela de Produtos
Fluxo Principal de eventos	<p>P1. O usuário seleciona “mais opções” e “adicionar promoção”.</p> <p>P2. O sistema abre a tela de criar promoção.</p> <p>P3. O usuário seleciona o produto que deseja colocar em promoção e uma data de limite para a promoção.</p> <p>P4. O sistema atualiza o banco de dados.</p>
Fluxo Alternativo de eventos	<p>A1. Remover Promoção</p> <p>A1.1. Em P1, o usuário seleciona “mais opções” e “remover Promoção”.</p> <p>A1.2. O sistema libera para selecionar produtos em promoção.</p> <p>A1.3. O usuário seleciona os produtos que quer tirar de promoção.</p> <p>A1.4. O sistema remove as promoções do banco de dados.</p>

[CdU007] Manter Kit de Produtos:

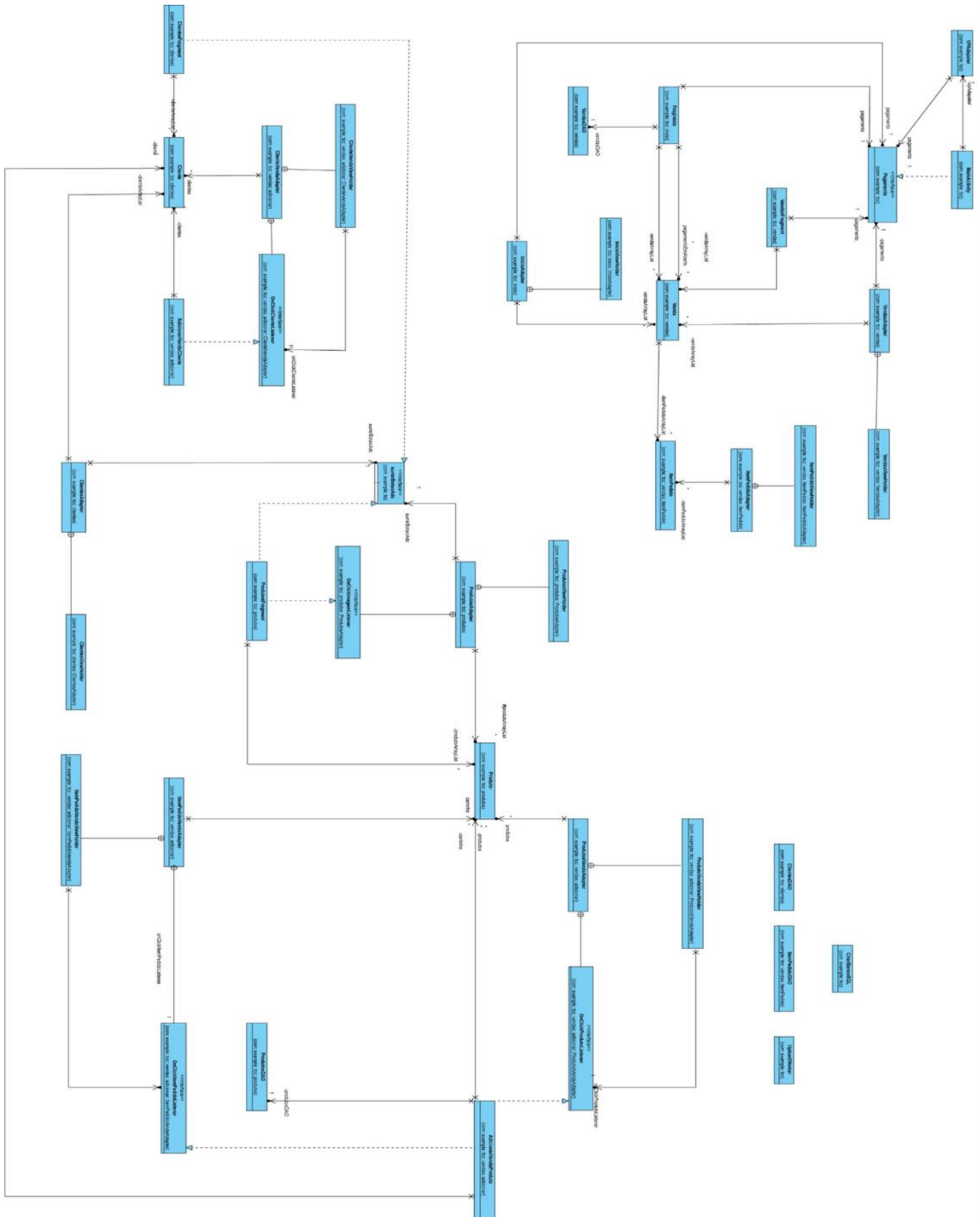
Ator Principal	Usuário
Pré-requisito	Estar na tela de Produtos
Fluxo Principal de eventos	<p>P1. O usuário seleciona “mais opções” e “adicionar Kit de Produtos”.</p> <p>P2. O sistema abre a tela de criar Kit de Produtos.</p> <p>P3. O usuário seleciona os produtos que deseja colocar em Kit de Produtos.</p> <p>P4. O sistema atualiza o banco de dados.</p>

[CdU008] Gerar PDF:

Ator Principal	Usuário
Pré-requisito	Estar na tela de Produtos
Fluxo Principal de eventos	<p>P1. O usuário seleciona “mais opções” e “gerar PDF”.</p>

	P2. O sistema gera um PDF com todos os Produtos no banco de dados
--	---

# APÊNDICE E – DIAGRAMA DE CLASSES



**APÊNDICE F – FORMULARIO DE SATISFAÇÃO DO APLICATIVO**

---

## Formulário de Satisfação do aplicativo

Esse formulário serve para medir o quão satisfeito você esta com o aplicativo Seller'sbook

**\*Obrigatório**

1. Você utilizou o aplicativo "Seller's Book" ? \*

*Marcar apenas uma oval.*

Sim

Não

### Perfil

2. Qual a sua idade? \*

*Marcar apenas uma oval.*

Menos que 20 anos

20 a 29 anos

30 a 39 anos

40 a 49 anos

50 a 60 anos

Mais que 60 anos

3. Qual o seu sexo? \*

*Marcar apenas uma oval.*

Masculino

Feminino



8. Quão provável você acha que pode de apagar algum dado importante do sistema sem querer? \*

*Marcar apenas uma oval.*

	1	2	3	4	5	
Pouco provável	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muito provável

9. O quanto a consistência da disposição dos elementos na tela do sistema (botões, caixas de texto, etc.) facilitou o seu uso do sistema? \*

*Marcar apenas uma oval.*

	1	2	3	4	5	
Muito ruim	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muito boa

### **Final**

10. O que você achou do aplicativo como um todo? \*

*Marcar apenas uma oval.*

	1	2	3	4	5	
Muito ruim	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muito bom

11. Você acha que ele pode te ajudar a substituir a agenda/ livro de anotações? \*

*Marcar apenas uma oval.*

	1	2	3	4	5	
Pouco provável	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muito provável

12. Você recomendaria esse aplicativo para outra pessoa? \*

*Marcar apenas uma oval.*

	1	2	3	4	5	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Com certeza

13. Coloque aqui comentários e sugestões para o aplicativo.

---

---

---

---

---