



Ezequiel Soares Alcantara

Juan Zamarrenho Carvalho Correa

Lucas Souza Miranda

Matheus dos Santos Souza

Vinicius Garcia Chiarotti

## **Sistema Mobile Para Pagamentos Ágeis Em Lojas de Varejo**

SÃO CAETANO DO SUL

2020

Ezequiel Soares Alcantara

Juan Zamarrenho Carvalho Correa

Lucas Souza Miranda

Matheus dos Santos Souza

Vinicius Garcia Chiarotti

## **Sistema Mobile Para Pagamentos Ágeis Em Lojas de Varejo**

Trabalho de Conclusão de Curso apresentado à Faculdade de Tecnologia de São Caetano do Sul, sob a orientação do Professor: Dr. Rodrigo Amorim Motta Carvalho, como requisito parcial para a obtenção do diploma de Graduação no Curso de Análise e Desenvolvimento de Sistemas.

SÃO CAETANO DO SUL

2020

Dedicamos este trabalho de conclusão de curso, aos nossos professores, familiares e amigos que nos apoiaram no decorrer do longo caminho de toda a duração do curso.

## **AGRADECIMENTOS**

Primeiramente, queremos agradecer aos nossos familiares, que nos apoiaram e ajudaram durante toda a faculdade. Aos nossos amigos que nos auxiliaram no processo de aprendizagem das matérias do curso. Aos professores que nos ajudaram neste projeto de conclusão de curso seja direta ou indiretamente.

Um agradecimento especial ao Professor/Orientador Dr. Rodrigo Amorim Motta Carvalho, por toda dedicação e orientação para a realização deste trabalho.

Por fim agradecemos a Instituição, FATEC – São Caetano do Sul, por todo seu suporte.

“Tornou-se chocantemente óbvio que a nossa tecnologia excedeu a nossa humanidade.”

( *Albert Einstein* )

## RESUMO

O processo de compras em um supermercado e loja de varejo geralmente acaba sendo estressante por conta das filas nos caixas, mesmo tendo surgido o “caixa rápido” que é destinado a pessoas com pouco volume de itens, acabou não sendo capaz de suprir toda essa demanda. Pensando nas pessoas que anseiam por inovações e novas tecnologias, este projeto visa facilitar as compras nos supermercados e lojas de varejo. O aplicativo funcionará como um carrinho virtual, no qual os clientes poderão escanear o código de barra dos produtos que pretendem comprar dentro da loja, assim como, também poderão buscar o produto por uma barra de pesquisa caso não queira usar o código de barra, e no final das compras, o aplicativo descontará o valor final do cartão cadastrado.

**Palavras-chave:** Aplicativo, Mercado, APIs e Carrinho Virtual

## **ABSTRACT**

CORREA, Juan Z. C., SOUZA, Lucas, SOUZA, Matheus S., CHIAROTTI, Vinicius  
**Creation of a Mobile System for Agile Super Market Payments.** Final Paper –  
Faculdade de Tecnologia de São Caetano do Sul – Antônio Russo, São Caetano do  
Sul, 2020.

The shopping process at a supermarket and retail stores usually ends being stressful due to the queue time at the cashiers, even at the "fast cashiers" which were made aiming at people with low volume items, ended up not being able to supply all the demand. Thinking of people who yearn innovations and new technologies, our project aims to facilitate purchases in supermarkets and retail stores. The application will act as a cart in which customers can scan the barcode of the products they intend to buy, they will also be able to search by text for the product in the app if they don't want to use the barcode, and at the end of the purchase, the application will charge the user's credit card

**Keywords:** App, Supermarket, APIs e Virtual Cart

## LISTA DE ILUSTRAÇÕES E GRÁFICOS

Figura 1. Representação do Sistema com Módulos .....	17
Figura 2. Representação do BD .....	18
Figura 3. Escopo do Projeto .....	20
Figura 4. Amazon Go .....	24
Figura 5. Modelo de navegação EZPay .....	27
Figura 6. Modelo de navegação EZPay Mercado .....	28
Figura 7. Diagrama de casos de uso cliente .....	31
Figura 8 DC EZPay .....	42
Figura 9. DS Criar conta .....	43
Figura 10. DS Login App .....	43
Figura 11. DS Adicionar Produtos ao Carrinho .....	44
Figura 12. DS Altera Quantidade de Produtos .....	44
Figura 13. DS Remove Produtos do Carrinho .....	45
Figura 14. DS Finalizar Compra .....	45
Figura 15. DS Adicionar Cartão .....	46
Figura 16. DS Recuperar Senha .....	46
Figura 17. DS Alterar Perfil .....	47
Figura 18. Diagrama de casos de uso loja .....	50
Figura 19. DC EZPay Mercado .....	59
Figura 20. DS Criar Conta - Loja .....	60
Figura 21. DS Login App - Loja .....	60
Figura 22. DS Adicionar Produtos ao Aplicativo .....	61
Figura 23. DS Altera Produtos do Aplicativo .....	61
Figura 24. DS Remove Produtos do Aplicativo .....	62
Figura 25. DS Verifica Relatórios de Compra .....	62
Figura 26. DS Verifica Relatórios de Estoque .....	63
Figura 27. Recuperando Senha (Teste) .....	64
Figura 28. Selecionando Produto (teste) .....	65
Figura 29. Realizando a Compra (Teste) .....	66
Figura 30. Verificando Relatórios (Teste) .....	67
Figura 31. Tela de Login .....	69
Figura 32. Tela de Cadastro .....	70



Figura 33. Tela de Recuperar Senha .....	71
Figura 34. Tela Principal .....	72
Figura 35. Tela de Busca .....	73
Figura 36. Tela do Carrinho .....	74
Figura 37. Tela de Finalizar Compra .....	75
Figura 38. Tela de Perfil .....	76
Figura 39. Tela Principal EZPay Mercado .....	77
Figura 40. Tela de Adicionar Produtos .....	78
Figura 41. Tela de Remover Produtos .....	79
Figura 42. Tela de Alterar Produtos .....	80
Figura 43. Tela de Relatório de Estoque .....	81
Figura 44. Tela de Relatório de Vendas .....	82

## LISTA DE TABELAS

Tabela 1. Requisitos funcionais EZPay .....	29
Tabela 2. Requisitos não funcionais EZPay .....	30
Tabela 3. Tabela de ator (Cliente) .....	32
Tabela 4. CSU Criar conta .....	33
Tabela 5. CSU Login app .....	34
Tabela 6. CSU Adicionar produtos ao carrinho .....	35
Tabela 7. CSU Altera quantidade de produtos .....	36
Tabela 8. CSU Remover produtos do carrinho .....	37
Tabela 9. CSU Finalizar compra .....	38
Tabela 10. CSU Adicionar cartão .....	39
Tabela 11. CSU Recuperar senha .....	40
Tabela 12. CSU Alterar perfil .....	41
Tabela 13. Requisitos funcionais EZPay Mercado .....	48
Tabela 14. Requisitos não funcionais EZPay Mercado .....	49
Tabela 15. ator (Loja) .....	51
Tabela 16. CSU Criar conta - Loja .....	52
Tabela 17. CSU Login app - Loja .....	53
Tabela 18. CSU Adicionar produtos ao aplicativo .....	54
Tabela 19. CSU Alterar produtos do aplicativo .....	55
Tabela 20. CSU Remover produtos do aplicativo .....	56
Tabela 21. CSU Verificar relatórios de compra .....	57
Tabela 22. CSU Verificar relatórios de estoque .....	58

## **LISTA DE ABREVIATURAS E SIGLAS**

API – Application Programming Interface.

APP – Aplicativo.

BD – Banco de Dados

CPF – Cadastro de Pessoa Física.

CSU – Casos de Uso.

OBS – Observação.

PC – Personal Computer.

RF – Requisito Funcional.

RNF – Requisito Não Funcional.

SO – Sistema operacional.

UML – Unified Modeling Language.

XML – Extensible Markup Language.

DS – Diagrama de Sequência.

## **LISTA DE SÍMBOLOS**

\$ - Cifrão/Símbolo de Dólar

## SUMÁRIO

<b>INTRODUÇÃO.....</b>	<b>17</b>
<b>1 METODOLOGIA.....</b>	<b>20</b>
1.1 Escopo do Projeto .....	20
1.2 Ferramentas .....	21
1.3 Versões para desenvolvimento.....	22
<b>2 DISPOSITIVOS MOVEIS .....</b>	<b>23</b>
2.1 Aplicativo Similar .....	23
<b>3 PLANEJAMENTO DOS APLICATIVOS.....</b>	<b>25</b>
3.1 Propósito .....	25
3.2 Usuários .....	26
3.2.1 Loja: .....	26
3.2.2 Cliente.....	26
3.3 Modelo de navegação .....	27
3.3.1 Navegação EZPay .....	27
3.3.2 Navegação EZPay Mercado.....	28
<b>4 APLICATIVO EZPAY (CLIENTE).....</b>	<b>29</b>
4.1 Engenharia de requisitos .....	29
4.1.1 Requisitos funcionais .....	29
4.1.2 Requisitos não funcionais .....	30
4.2 Especificações dos casos de uso .....	31
4.2.1 Diagrama de casos de uso.....	31
4.2.2 Documentação do Ator.....	32
4.3 Casos de uso.....	33
4.3.1 Criar Conta.....	33
4.3.2 Login App.....	34

4.3.3 Adicionar Produtos ao Carrinho .....	35
4.3.4 Altera Quantidade de Produtos .....	36
4.3.5 Remove Produtos do Carrinho .....	37
4.3.6 Finalizar Compra .....	38
4.3.7 Adicionar Cartão .....	39
4.3.8 Recuperar Senha .....	40
4.3.9 Alterar Perfil (Cliente) .....	41
4.4 Diagrama de classes .....	42
4.5 Diagrama de sequência .....	43
4.5.1 Criar Conta.....	43
4.5.2 Login App.....	43
4.5.3 Adicionar Produtos ao Carrinho .....	44
4.5.4 Altera Quantidade de Produtos .....	44
4.5.5 Remove Produtos do Carrinho .....	45
4.5.6 Finalizar Compra.....	45
4.5.7 Adicionar Cartão .....	46
4.5.8 Recuperar Senha .....	46
4.5.9 Alterar Perfil .....	47
<b>5 APLICATIVO EZPAY MERCADO (LOJA).....</b>	<b>48</b>
5.1 Engenharia de requisitos .....	48
5.1.1 Requisitos funcionais .....	48
5.1.2 Requisitos não funcionais .....	49
5.2 Especificações dos casos de uso .....	50
5.2.1 Diagrama de casos de uso.....	50
5.2.2 Documentação do Ator.....	51
5.3 Casos de uso.....	52
5.3.1 Criar Conta.....	52

5.3.2 Login App.....	53
5.3.3 Adiciona Produtos ao Aplicativo.....	54
5.3.4 Altera Produtos do Aplicativo.....	55
5.3.5 Remove Produtos do Aplicativo.....	56
5.3.6 Verificar Relatórios de Compra.....	57
5.3.5 Verificar Relatórios de Estoque.....	58
5.4 Diagrama de classes.....	59
5.5 Diagrama de sequência.....	60
5.5.1 Criar Conta.....	60
5.5.2 Login App.....	60
5.5.3 Adicionar Produtos ao Aplicativo.....	61
5.5.4 Altera Produtos do Aplicativo.....	61
5.5.5 Remove Produtos do Aplicativo.....	62
5.5.6 Verifica Relatórios de Compra.....	62
5.5.7 Verifica Relatórios de Estoque.....	63
<b>6 PLANO DE TESTES FUNCIONAIS.....</b>	<b>64</b>
6.1 Testes realizados.....	64
6.1.1 Recuperando a Senha.....	64
6.1.2 Selecionando o produto.....	65
6.1.3 Realizando a compra.....	66
6.1.4 Verificando relatórios.....	67
<b>7 TELAS.....</b>	<b>69</b>
7.1 EZPay.....	69
7.2 EZPay Mercado.....	77
<b>8 PROCEDIMENTO DE INSTALAÇÃO.....</b>	<b>83</b>
7.1 App EZPay.....	83
7.2 App EZPay Mercado.....	83

<b>CONSIDERAÇÕES FINAIS.....</b>	<b>84</b>
<b>REFERÊNCIAS .....</b>	<b>85</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>87</b>



## INTRODUÇÃO

Softwares e dispositivos que auxiliam pessoas em suas tarefas diárias são cada vez mais uma tendência mundial. Pensando em um cenário do dia a dia onde determinada pessoa possui uma vida comum, ou seja, estuda, trabalha, realiza as tarefas domésticas etc., o mesmo necessita realizar todas essas tarefas de maneira rápida e eficaz, uma delas é a de realizar compras em supermercados quando há necessidade, com isso um software que agilize esse processo (que por vezes se torna algo longo e estressante) é muito bem-vindo.

É comum observar em supermercados pessoas deixando produtos e indo embora pelo tamanho da fila, falta de caixas, falta de tempo hábil para esperar. Os supermercados passaram a adotar os chamados “caixas rápidos” limitando um número de itens para compra e assim possibilitando um atendimento mais rápido e eficiente para pessoas que buscam comprar poucos produtos, porém, essa medida ainda assim gera gargalo em detrimento da alta demanda de clientes e baixa disponibilidade de caixas.

Como solução para tais problemas foi desenvolvido um sistema com dois módulos, que viraram dois aplicativos, um para o cliente e outra para a loja, seus respectivos nomes são EZPay e EZPay Mercado conforme mostrado na Figura 1.



Figura 1. Representação do Sistema com Módulos

Figura feita pelos integrantes do grupo

O EZPay é um aplicativo que poderá ser acessado a partir de um celular com sistema operacional Android, sendo compatível com versões mais recentes do SO, com isso atingindo o maior número de pessoas possíveis.

O EZPay contará com uma interface simples e intuitiva, onde o usuário terá que realizar um cadastro com algumas informações pessoais e após isso adicionará seu método de pagamento. Feito isso o aplicativo já se encontra disponível para uso com um limite de 15 itens disponíveis para adição, no momento que queira adicionar um produto em seu carrinho basta escanear o código de barras do produto e confirmar a adição, caso queira também poderá visualizar todos os itens adicionados e excluir se necessário, quando tiver terminado basta finalizar a compra e o valor total da compra será debitado do cartão selecionado.

Já na parte do EZPay Mercado o usuário deste App terá que realizar o cadastro do “mercado”, após isso poderá adicionar produtos ao aplicativo EZPay e terá responsabilidade de verificar relatórios de compras e de estoque.

A figura 2 representa a ligação entre os dois aplicativos com o banco de dados



Figura 2. Representação do BD

Figura feita pelos integrantes do grupo

Analisando a rotina de algumas pessoas, percebe-se que a rotina diária de cada pessoa acaba sendo bastante agitada, resultando na falta de tempo para tarefas que deveriam ser mais simples como ir ao supermercado.

A partir disto, foi proposta a criação de dois aplicativos no qual agilizará e deixará mais dinâmico o processo de compras nos supermercados e lojas de varejo, possibilitando ao cliente que não precise mais se preocupar com as filas.

O desenvolvimento do documento está dividido da seguinte forma, no capítulo um é mostrado a metodologia utilizada para criação dos aplicativos e as ferramentas utilizadas para o desenvolvimento de ambos.

No capítulo dois é citado brevemente sobre os dispositivos móveis existentes e o aplicativo Amazon Go que já foi desenvolvido pela empresa Amazon. No capítulo três começa a fase de planejamento dos aplicativos, no qual é citado o propósito para a criação deles, quais são os usuários que irão utiliza-los e o modelo de navegação dos mesmos, no capítulo quatro e cinco, é citada a descrição detalhada de cada um dos dois aplicativos, mostrando os requisitos funcionais e não funcionais.

O diagrama de casos de uso e os próprios casos de uso, é documentado os atores de cada aplicativo, também é mostrado os diagramas de classe e de sequência, no capítulo seis é mostrado o plano de teste no qual descreve as funções mais importantes de cada aplicativo e por fim no capítulo sete é mostrado o procedimento de instalação de cada um dos aplicativos.

## 1 METODOLOGIA

A metodologia utilizada para o desenvolvimento do projeto foi a metodologia Kanban por se tratar de uma metodologia ágil, com isso as tarefas mais importantes foram priorizadas além disso cada tarefa foi mais organizada.

### 1.1 Escopo do Projeto

O usuário (cliente) irá solicitar a compra através do aplicativo (EZPay), após isso será validado os dados da compra, com isso a confirmação da compra é enviada ao sistema que gerará relatórios sobre a venda para o aplicativo (EZPay Mercado) na área da Loja e por fim aparecerá na tela do app (EZPay), para o cliente, uma mensagem que a compra foi efetuado com sucesso.

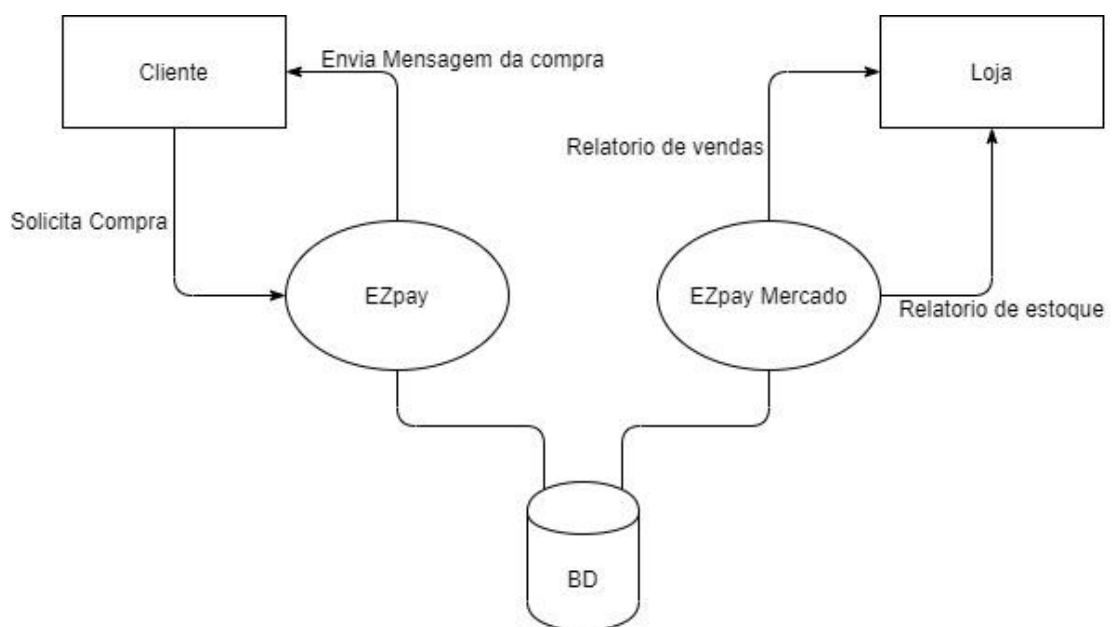


Figura 3. Escopo do Projeto

Figura feita pelos integrantes do grupo

## 1.2 Ferramentas

As ferramentas utilizadas para criação dos aplicativos consistem em utilizar o ambiente de desenvolvimento integrado chamado Android Studio com a linguagem de programação Java, com XML e bibliotecas para disponibilizar o código de barras. Para a parte de banco de dados foi utilizado o Firebase que é um banco de dados não relacional orientado a documentos.

Um pequeno resumo sobre do que se trata cada ferramenta que foi utilizada para o desenvolvimento:

Android Studio é um Ambiente de Desenvolvimento Integrado, mais conhecido como IDE (Integrated Development Environment), ele é utilizado para o desenvolvimento e criação de aplicativos destinado a dispositivos móveis, seu dono é o Google que fornece este software gratuitamente as pessoas que o queiram. (HARADA, 2019).

Java é uma linguagem de programação que é orientada a objetivo sendo derivada do C++, ela foi desenvolvida por uma equipe que era liderada por James Gosling e oficialmente lançada em 1995. (PEREIRA, 2019).

XML é uma linguagem de marcação indicada para criação de documentos com dados hierarquicamente organizados, como textos e bancos de dados. Uma de suas características do XML é possuir uma sintaxe básica que podendo ser utilizada para compartilhamento de informações entre diferentes computadores e aplicações. A portabilidade é um destaque do XML pois possibilita integração entre sistemas distintos, por exemplo um banco de dados pode escrever um arquivo XML para que outro banco consiga lê-lo. (PEREIRA, 2009).

APIs (Application Programming Interface) são formas de integrar sistemas através de rotinas e padrões de programação, possibilitando acesso a um software ou plataforma baseado na Web. Normalmente as pessoas não sabem exatamente o que é uma API isso porque elas são invisíveis ao usuário comum que só visualiza as interfaces já unificadas nos aplicativos, porém, profissionais em programação utilizam essa tecnologia resultante da evolução de diversos sistemas e ferramentas. (CANALTECH, 2019).

Firestore é um serviço da Google com foco em aplicações Web e Mobile, cada vez mais é utilizado por ser a melhor opção para determinados tipos de ferramentas em detrimento de sua alta disponibilidade de serviços e facilidade de ser implementado. As APIs do Firestore estão inclusas em um único SDK (Software Development Kit) possibilitando assim uma expansão de seu aplicativo para mais plataformas e linguagens. (ORLANDI, 2019).

Para o desenvolvimento do projeto foi optado o uso do banco de dados não relacional Firestore por proporcionar alta disponibilidade de serviços com facilidade de ser implementado.

### 1.3 Versões para desenvolvimento

No Android Studio foi utilizada a versão 4.0, com os seguintes emuladores e dispositivos.

- Emuladores: Pixel 3 XL API 29 (Android 10) e Galaxy Nexus API 26 (Android 8.0).
- Dispositivos: Zenfone 5, Zenfone 5z, Samsung A30 e Moto G7 Play.

## 2 DISPOSITIVOS MOVEIS

Este capítulo tem como objetivo descrever sobre a importância dos dispositivos moveis, quais tipos existem, qual foi utilizado para o desenvolvimento do EZPay e citar sobre um aplicativo existente parecido com o EZPay.

Os dispositivos moveis (Aplicativos) tem se tornado cada vez mais essenciais para as pessoas por facilitarem, muitas vezes, atividades do dia a dia. Eles também se tornaram uma grande oportunidade para vários desenvolvedores por ser um mercado que cresce cada vez mais. (ZANINI, 2017),

Existem três tipos de aplicativos (MADUREIRA, 2017) que são:

- **App Nativo:** são os aplicativos instalados no celular através de uma loja, como Google Play, Play Store, App Store, dentre outras. O app nativo quando instalado interage com o SO, utilizando seus recursos. Uma desvantagem seria que deve ser desenvolvido para cada plataforma.
- **Web App:** seu acesso não requer que o usuário instale o aplicativo, apenas requer que o utilize através do navegador de seu celular. A vantagem de desenvolver o Web App é que o custo é relativamente mais baixo que o do app nativo.
- **App Híbrido:** é basicamente a união dos dois anteriores, ele acaba sendo parcialmente nativo e parcialmente web. Atualmente o app híbrido é popular por poder ser usado em várias plataformas.

Os dois aplicativos desenvolvidos foram criados com base no App Nativo.

### 2.1 Aplicativo Similar

Amazon Go, acabou sendo revolucionário quando foi lançado, 2018, por se tratar de um serviço de vendas em lojas sem os caixas. O cliente da loja deve

baixar o aplicativo para realizar suas compras sem caixa, assim permitindo que o cliente selecione os itens que deseja e depois simplesmente saia da loja.

O aplicativo da Amazon tem um sistema que verifica quando os produtos são retirados das prateleiras e quando retornam a elas e esse registro é mantido no carrinho virtual do app da conta do usuário. O pagamento é feito através do aplicativo assim que o cliente sai da loja, após isso o aplicativo envia um recibo para o e-mail do usuário. (JUSTASKGEMALTO, 2017).



Figura 4. Amazon Go

Fonte: <https://jornaldebrasil.com.br/blogs-e-colunas/bsbtek/amazon-go-ficara-disponivel-para-qualquer-loja/> (2020)



### 3 PLANEJAMENTO DOS APLICATIVOS

Este capítulo tem como objetivo descrever os dois aplicativos destacando o propósito de cada um, quais são os usuários destinados para seu uso e o modelo de navegação dos dois Apps.

O aplicativo EZPay é destinado aos clientes do mercado e funciona como um carrinho virtual, onde é possível escanear o código de barra de determinado produto ou apenas digitar o nome do produto que deseja e o mesmo será adicionado em seu carrinho virtual, no final das compras o aplicativo debita o total da compra do método de pagamento que o cliente cadastrou. Já o EZPay Mercado tem como público-alvo o próprio mercado, é utilizado para adicionar produtos ao banco de dados e para emissão de relatórios de estoque e compras.

#### 3.1 Propósito

O projeto foi construído com o objetivo de facilitar as compras de usuários de mercados, pois funciona como um carrinho online e o processo de compra é realizado pelo próprio usuário. Com o uso do aplicativo (EZPay), o cliente terá controle de suas compras e seu tempo, já que não precisará se preocupar com filas excessivas.

Para os clientes que poderão comprar o serviço para ser utilizado no seu estabelecimento, o uso do aplicativo (EZPay Mercado) ajuda a organizar e acelerar o fluxo de compras em seu estabelecimento, além de diminuir os custos pois a quantidade de computadores, sensores e até mesmo funcionários será menor. Com o aplicativo, os pagamentos serão mais rápidos, já que serão realizados com cartão, diminuindo, assim, problemas com dinheiro em espécie.

## 3.2 Usuários

Existem dois tipos de usuário, cliente e mercado e os dois possuem funções diferentes dentro do sistema.

3.2.1 Loja: responsável por adicionar novos produtos ao aplicativo inserindo seus dados e o código de barras além disso também é responsável por receber os relatórios de compras e estoque. Foi colocado o nome deste usuário de loja, para ele se referenciar tanto a supermercados quanto a lojas de varejo.

3.2.2 Cliente: responsável por se cadastrar dentro do sistema e inserir um cartão de crédito, feito isso deverá escanear o código de barra de cada produto (ou buscar o nome do produto pela barra de pesquisa), o sistema adicionará esse produto ao carrinho virtual do cliente e após incluir todos os itens desejados ou atingir o limite de 20 itens, o sistema irá redirecionar o usuário para tela de pagamento onde o cliente visualizará todos os itens e poderá finalizar a compra caso deseje.

### 3.3 Modelo de navegação

O modelo de navegação demonstra todos os caminhos que podem ser seguidos pelo usuário dentro do aplicativo.

#### 3.3.1 Navegação EZPay

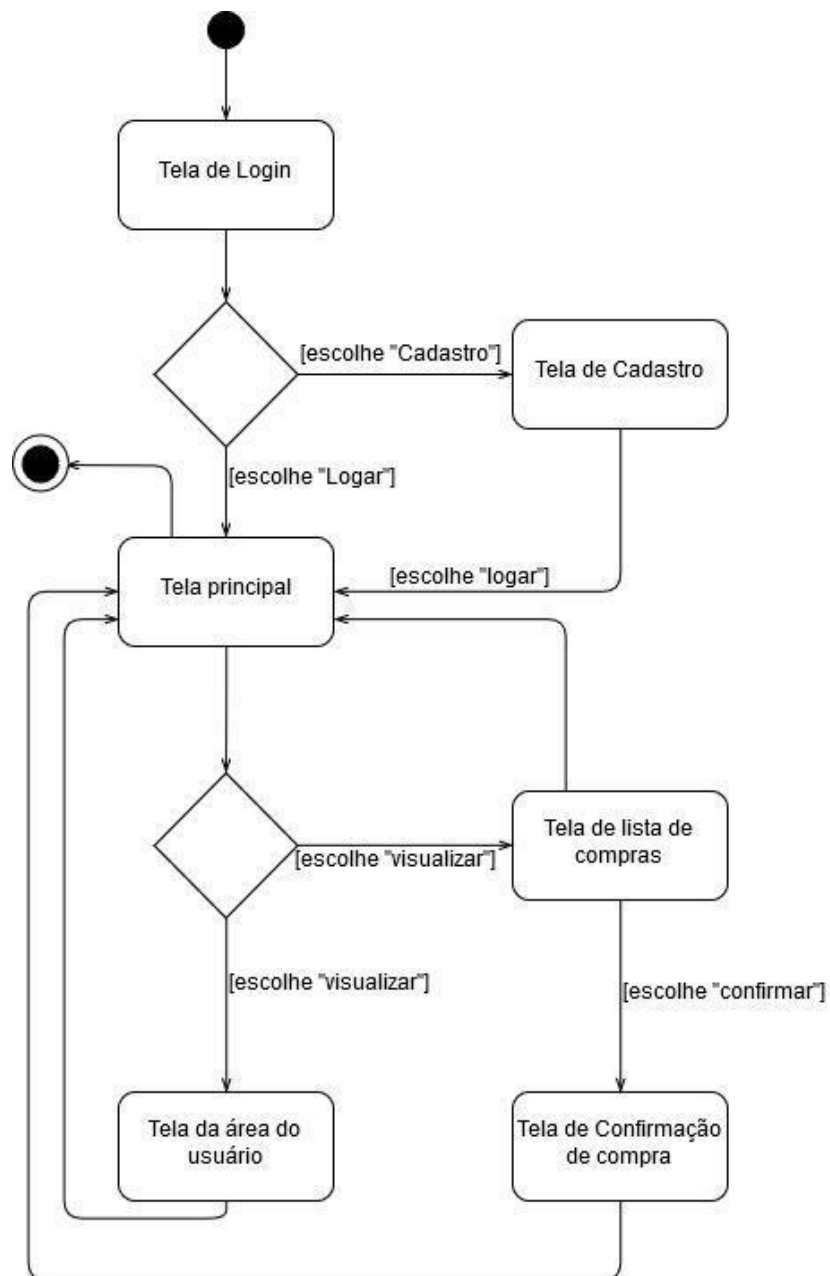


Figura 5. Modelo de navegação EZPay

Figura feita pelos integrantes do grupo

## 3.3.2 Navegação EZPay Mercado

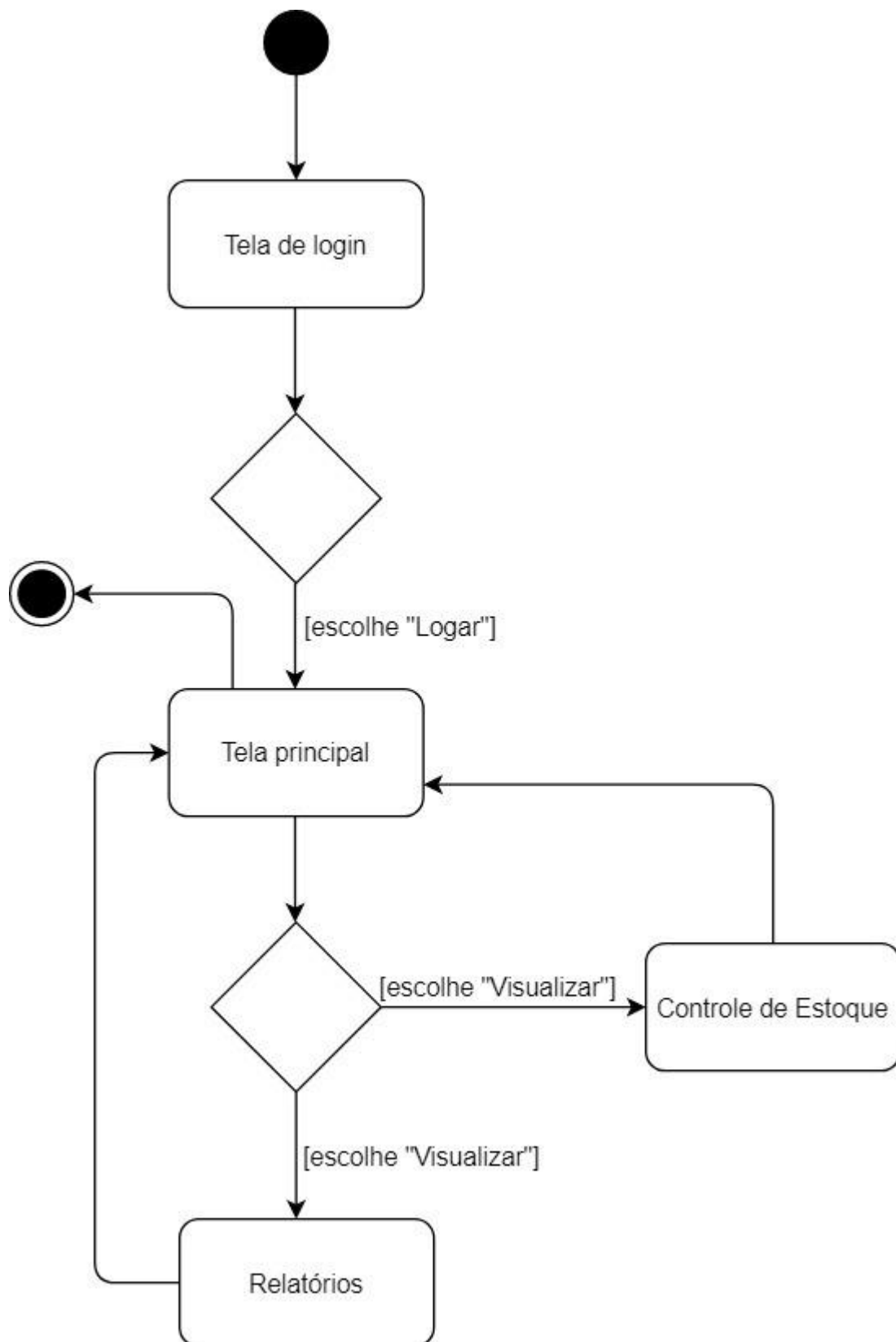


Figura 6. Modelo de navegação EZPay Mercado

Figura feita pelos integrantes do grupo

## 4 APLICATIVO EZPAY (CLIENTE)

Este capítulo tem como o objetivo mostrar as especificações do aplicativo EZPay, mostrando requisitos funcionais e não funcionais, casos de uso e diagramas, mostrando assim como cada função atua dentro do aplicativo.

### 4.1 Engenharia de requisitos

Foram levantados requisitos funcionais e não funcionais baseados em pesquisas, para atender as necessidades dos aplicativos.

#### 4.1.1 Requisitos funcionais

<b>ID</b>	<b>Descrição</b>	<b>Caso de uso Relacionado</b>
RF001	Realizar Cadastro de usuários	CSU001
RF002	Verificar dados do usuário	CSU002
RF003	Alterar dados do usuário	CSU007, CSU008, CSU009
RF004	Recuperar senha de usuários	CSU008
RF005	Adicionar produtos	CSU003
RF006	Remover produtos	CSU005
RF007	Alterar quantidade de produtos	CSU004
RF008	Confirmar compra	CSU006
RF009	Cadastrar cartão	CSU007
RF010	Verificar dados do cartão	CSU006
RF011	Solicitar permissão para habilitar câmera	CSU003

Tabela 1. Requisitos funcionais EZPay

Tabela feita pelos integrantes do grupo

## 4.1.2 Requisitos não funcionais

<b>ID</b>	<b>Categoria</b>	<b>Descrição</b>
RNF001	Usabilidade	O sistema deve possuir um layout interativo, com ferramentas de fácil entendimento, assim, sendo manipulável por qualquer tipo de usuário.
RNF002	Desempenho	Tempo de resposta para ações internas deverá ser de no máximo 10 segundos
RNF003	Interoperabilidade	O sistema deverá se comunicar com o banco Firebase
RNF004	Implementação	O sistema deverá ser desenvolvido na linguagem Java
RNF005	Confiabilidade	O sistema deverá ter alta disponibilidade, por exemplo, 99% do tempo
RNF006	Portabilidade	O sistema deverá executar em qualquer smartphone que possuir Android com versões acima de 8.0
RNF007	Interoperabilidade	O sistema deverá se comunicar com o banco de dados não relacional Firebase
RNF008	Interoperabilidade	O sistema deverá se comunicar com uma biblioteca para escanear código de barras
RNF009	Manutenibilidade	A manutenção do aplicativo deve ser rápida e eficiente para evitar inatividade do sistema
RNF0010	Segurança	O sistema deve criptografar os dados sensíveis do usuário

Tabela 2. Requisitos não funcionais EZPay

Tabela feita pelos integrantes do grupo

## 4.2 Especificações dos casos de uso

As especificações dos casos de uso consistem em apresentar o diagrama de casos de uso do aplicativo e o ator que os utiliza.

### 4.2.1 Diagrama de casos de uso

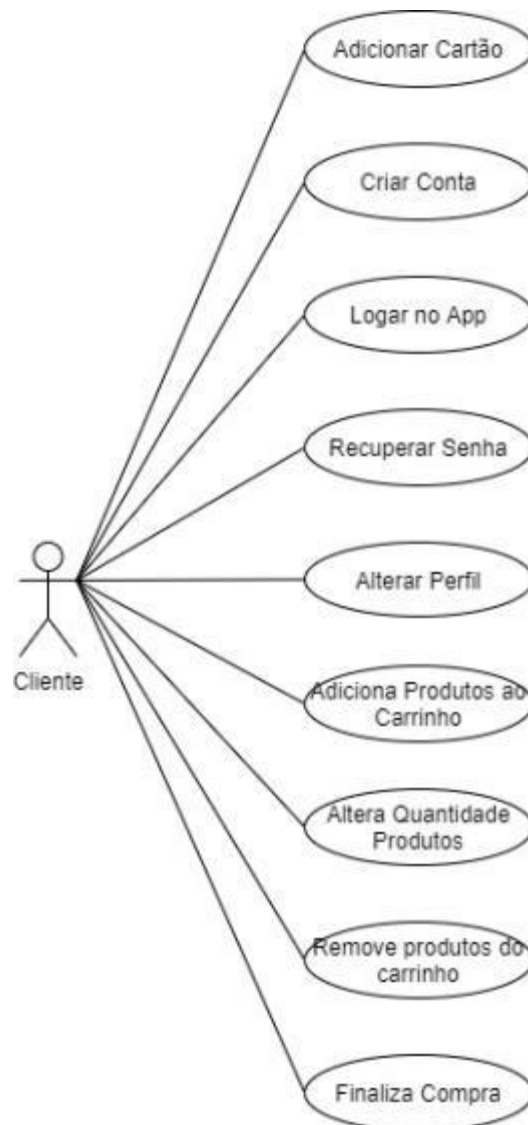


Figura 7. Diagrama de casos de uso cliente

Figura feita pelos integrantes do grupo

## 4.2.2 Documentação do Ator

<b>Ator: Cliente</b>
<p><b>Resumo:</b></p> <p>O Cliente é o ator que irá utilizar o sistema, podendo criar uma conta, fazer login, cadastrar um cartão, acessar seu carrinho de compras, adicionar e remover produtos do carrinho e finalizar as compras.</p>
<p><b>Responsabilidades:</b></p> <ol style="list-style-type: none"> <li>1. Criar conta</li> <li>2. Fazer login</li> <li>3. Recuperar senha</li> <li>4. Alterar perfil</li> <li>5. Adicionar produtos ao carrinho</li> <li>6. Altera quantidade de produtos</li> <li>7. Remover produtos do carrinho</li> <li>8. Finalizar compras</li> <li>9. Adicionar cartão de crédito</li> </ol>
<p><b>Ambiente Físico:</b></p> <p>O usuário utiliza o aplicativo no próprio celular.</p>

Tabela 3. Tabela de ator (Cliente)

Tabela feita pelos integrantes do grupo



### 4.3 Casos de uso

Os casos de uso descrevem sobre como funciona cada ação feita no sistema do aplicativo.

#### 4.3.1 Criar Conta

<b>Caso de Uso: Criar Conta</b>
<b>ID:</b> CSU001
<b>Resumo:</b> Esse caso de uso é responsável por criar uma conta nova para cada usuário que deseja utilizar o aplicativo
<b>Atores primários:</b> Cliente
<b>Atores secundários:</b> Back-end (sistema)
<b>Pré-condições:</b> <ol style="list-style-type: none"> <li>1. Cliente deve ter um e-mail</li> <li>2. Esse e-mail não pode ter sido cadastrado antes</li> </ol>
<b>Fluxo de eventos principal:</b> <ol style="list-style-type: none"> <li>1. Cliente acessa o aplicativo.</li> <li>2. Sistema exibe a tela de login e opção de nova conta</li> <li>3. Cliente escolhe criar conta</li> <li>4. Sistema exibe a página de cadastro</li> <li>5. Cliente preenche a página de cadastro</li> <li>6. Sistema guarda as informações no banco de dados</li> </ol>
<b>Pós-condições:</b> <ol style="list-style-type: none"> <li>1. Depois da conta ser criada, o sistema abre a tela principal</li> <li>2. Sistema atualiza BD</li> </ol>
<b>Fluxo alternativo:</b> <ol style="list-style-type: none"> <li>1. Sistema pergunta se o cliente deseja criar uma conta.</li> <li>2. Se não, sistema mostra a página de login.</li> </ol>
<b>Fluxo de exceção: caso o e-mail criado já seja cadastrado</b> <ol style="list-style-type: none"> <li>1. Sistema informa ao usuário que o e-mail já foi cadastrado</li> <li>2. Sistema retorna ao passo 4 do fluxo principal</li> </ol>

Tabela 4. CSU Criar conta

Tabela feita pelos integrantes do grupo

## 4.3.2 Login App

<b>Caso de Uso: Login App</b>
<b>ID:</b> CSU002
<b>Resumo:</b> Este caso de uso representa a ação do cliente logar no app para acessar a tela principal
<b>Atores primários:</b> Cliente
<b>Atores secundários:</b> Back-end (sistema)
<b>Pré-condições:</b> <ol style="list-style-type: none"> <li>1. Cliente deve ter um cadastro</li> </ol>
<b>Fluxo de eventos principal:</b> <ol style="list-style-type: none"> <li>1. Cliente inicializa o aplicativo.</li> <li>2. Cliente se depara com a tela de login.</li> <li>3. Cliente insere o e-mail, senha e nível de acesso.</li> <li>4. Sistema verifica credenciais.</li> <li>5. Sistema redireciona para a tela principal e o caso de uso termina.</li> </ol>
<b>1 - Fluxo alternativo: caso os dados inseridos sejam inválidos</b> <ol style="list-style-type: none"> <li>1. Sistema informa que os dados digitados são incorretos</li> <li>2. Sistema retorna ao passo 3 do fluxo principal</li> </ol>
<b>2 - Fluxo alternativo: caso o cadastro não exista</b> <ol style="list-style-type: none"> <li>1. Sistema informa que o e-mail não foi cadastrado</li> <li>2. Sistema redireciona cliente para tela de cadastro</li> </ol>

Tabela 5. CSU Login app

Tabela feita pelos integrantes do grupo

## 4.3.3 Adicionar Produtos ao Carrinho

<b>Caso de Uso:</b> Adicionar produtos ao carrinho
<b>ID:</b> CSU003
<b>Resumo:</b> Este caso de uso representa a ação do usuário adiciona, seja escaneando ou busca por nome, os produtos desejados ao seu carrinho virtual
<b>Atores primários:</b> Cliente
<b>Atores secundários:</b> Back-end (sistema)
<b>Pré-condições:</b> <ol style="list-style-type: none"> <li>1. O cliente tem que estar logado no app</li> </ol>
<b>Fluxo de eventos principal:</b> <ol style="list-style-type: none"> <li>1. Cliente acessa tela de escanear pedidos</li> <li>2. Sistema pede ao usuário permissão para o uso da câmera</li> <li>3. Cliente permite o uso da câmera</li> <li>4. Cliente utiliza câmera do celular para leitura do código de barras</li> <li>5. Sistema adiciona produto ao carrinho</li> </ol>
<b>Pós-condições:</b> <ol style="list-style-type: none"> <li>1. Sistema libera tela de compra</li> </ol>
<b>1 - Fluxo de exceção: Caso o usuário não autorize o uso da câmera</b> <ol style="list-style-type: none"> <li>1. Cliente não permite o uso da câmera</li> <li>2. Sistema mostra opção de digitar o nome do produto</li> <li>3. Cliente digita o nome do produto e confirma</li> <li>4. Sistema adiciona produto ao carrinho</li> </ol>
<b>2 - Fluxo de exceção: Caso o código de barra do produto não funcione</b> <ol style="list-style-type: none"> <li>1. Sistema informa ao cliente que o código de barras é inválido</li> <li>2. Sistema mostra opção de digitar o nome do produto</li> <li>3. cliente digita o nome do produto e confirma</li> <li>4. Sistema adiciona produto ao carrinho</li> </ol>

Tabela 6. CSU Adicionar produtos ao carrinho

Tabela feita pelos integrantes do grupo

## 4.3.4 Altera Quantidade de Produtos

<b>Caso de Uso:</b> Alterar quantidade de produtos
<b>ID:</b> CSU004
<b>Resumo:</b> Representa a ação do cliente de alterar a quantidade de itens de cada produto do carrinho
<b>Atores primários:</b> Cliente
<b>Atores secundários:</b> Back-end (sistema)
<b>Pré-condições:</b> <ol style="list-style-type: none"> <li>1. Cliente deve estar logado</li> <li>2. Deve existir produtos no carrinho</li> </ol>
<b>Fluxo de eventos principal:</b> <ol style="list-style-type: none"> <li>1. Cliente Aperta ícone de (+/-) para mudar quantidade de produtos</li> <li>2. Sistema valida quantidade de produtos</li> <li>3. Sistema altera a quantidade do produto</li> </ol>
<b>Fluxo alternativo: caso o estoque não seja suficiente</b> <ol style="list-style-type: none"> <li>1. Sistema informa que esta quantidade não está disponível em estoque</li> <li>2. Sistema retorna ao passo 1 do fluxo principal</li> </ol>

Tabela 7. CSU Altera quantidade de produtos

Tabela feita pelos integrantes do grupo

## 4.3.5 Remove Produtos do Carrinho

<b>Caso de Uso:</b> Remover produtos do carrinho
<b>ID:</b> CSU005
<b>Resumo:</b> Este caso de uso representa a ação do usuário remover produtos da lista no carrinho virtual
<b>Atores primários:</b> Cliente
<b>Atores secundários:</b> Back-end (sistema)
<b>Pré-condições:</b> <ol style="list-style-type: none"> <li>1. O cliente tem que estar logado no app</li> <li>2. O cliente deve ter pelo menos 1 produtos no carrinho</li> </ol>
<b>Fluxo de eventos principal:</b> <ol style="list-style-type: none"> <li>1. Cliente acessa tela de lista de compras</li> <li>2. Cliente manuseia a lista até encontrar o produto desejado</li> <li>3. Cliente seleciona o ícone de lixo para retirar o produto da lista</li> <li>4. Sistema remove o produto e atualiza a lista de compras</li> </ol>

Tabela 8. CSU Remover produtos do carrinho

Tabela feita pelos integrantes do grupo

## 4.3.6 Finalizar Compra

<b>Caso de Uso:</b> Finalizar compra
<b>ID:</b> CSU006
<b>Resumo:</b> Este caso de uso representa a ação do cliente finalizar suas compras no aplicativo
<b>Atores primários:</b> Cliente
<b>Atores secundários:</b> Back-end (sistema)
<b>Pré-condições:</b> <ol style="list-style-type: none"> <li>1. O cliente deve estar logado</li> <li>2. O cliente tem que ter adicionado pelo menos 1 produto ao carrinho</li> </ol>
<b>Fluxo de eventos principal:</b> <ol style="list-style-type: none"> <li>1. Cliente acessa a tela do carrinho virtual</li> <li>2. Cliente seleciona opção de finalizar compra</li> <li>3. Cliente seleciona cartão desejado para finalizar compra</li> <li>4. Sistema verifica dados do cartão do usuário</li> <li>5. Sistema valida cartão do usuário</li> <li>6. Sistema finaliza a compra</li> </ol>
<b>Pós-condições:</b> <ol style="list-style-type: none"> <li>1. Sistema atualiza o BD</li> </ol>
<b>1 - Fluxo de exceção: caso o cartão ainda não foi cadastrado no app</b> <ol style="list-style-type: none"> <li>1. Sistema informa que não há cartões cadastrados</li> <li>2. Sistema envia o cliente para tela de adicionar cartão</li> </ol>
<b>2 - Fluxo de exceção: caso o cartão informado não tenha débito suficiente para efetuar a compra</b> <ol style="list-style-type: none"> <li>1. Sistema informa que o cartão cadastrado não tem débito suficiente</li> <li>2. Sistema envia cliente para tela de adicionar cartão</li> </ol>

Tabela 9. CSU Finalizar compra

Tabela feita pelos integrantes do grupo

## 4.3.7 Adicionar Cartão

<b>Caso de Uso:</b> Adicionar cartão
<b>ID:</b> CSU007
<b>Resumo:</b> Este caso de uso representa a ação do cliente adicionar seus cartões no app
<b>Atores primários:</b> Cliente
<b>Atores secundários:</b> Back-end (sistema)
<b>Pré-condições:</b> <ol style="list-style-type: none"> <li>1. Cliente deve ter um cartão</li> <li>2. Cliente deve estar logado</li> </ol>
<b>Fluxo de eventos principal:</b> <ol style="list-style-type: none"> <li>1. Cliente acessa tela de adicionar cartão</li> <li>2. Cliente preenche os campos com os dados do cartão</li> <li>3. Sistema valida cartão</li> <li>4. Sistema cadastra o cartão</li> </ol>
<b>Pós-condições:</b> <ol style="list-style-type: none"> <li>1. Sistema atualiza BD</li> </ol>
<b>Fluxo alternativo: Cartão Invalido</b> <ol style="list-style-type: none"> <li>1. Sistema informa cliente que o cartão é invalido</li> <li>2. Sistema informa que para finalizar compra precisa de um cartão valido</li> </ol>

Tabela 10. CSU Adicionar cartão

Tabela feita pelos integrantes do grupo

## 4.3.8 Recuperar Senha

<b>Caso de Uso: Recuperar Senha</b>
<b>ID:</b> CSU008
<b>Resumo:</b> Representa a ação do usuário recuperar a senha de acesso ao app
<b>Atores primários:</b> Cliente
<b>Atores secundários:</b> Back-end (sistema)
<b>Pré-condições:</b> <ol style="list-style-type: none"> <li>1. Cliente deve ter cadastro no app</li> </ol>
<b>Fluxo de eventos principal:</b> <ol style="list-style-type: none"> <li>1. Cliente escolhe opção de recuperação de conta.</li> <li>2. Sistema mostra tela de recuperação de conta.</li> <li>3. Cliente informa e-mail cadastrado.</li> <li>4. Sistema envia um link de recuperação de senha para o e-mail do cliente.</li> <li>5. Cliente acessa o link enviado para o seu e-mail.</li> <li>6. Cliente insere uma nova senha.</li> <li>7. Cliente confirma nova senha.</li> <li>8. Sistema confirma alterações dos dados do cliente</li> <li>9. Sistema informa cliente que a senha foi alterada com sucesso</li> <li>10. Sistema redireciona cliente para tela de login</li> </ol>
<b>Pós-condições:</b> <ol style="list-style-type: none"> <li>1. Sistema atualiza BD</li> </ol>
<b>Fluxo alternativo: caso o usuário não saiba o e-mail cadastrado</b> <ol style="list-style-type: none"> <li>1. Cliente coloca opção de CPF</li> <li>2. Sistema valida CPF</li> <li>3. Sistema informa e-mail cadastrado</li> </ol>

Tabela 11. CSU Recuperar senha

Tabela feita pelos integrantes do grupo



## 4.3.9 Alterar Perfil (Cliente)

<b>Caso de Uso: Alterar perfil</b>
<b>ID:</b> CSU009
<b>Resumo:</b> Este caso de uso representa a ação do cliente alterar os dados cadastrados no aplicativo
<b>Atores primários:</b> Cliente
<b>Atores secundários:</b> Back-end (sistema)
<b>Pré-condições:</b> <ol style="list-style-type: none"> <li>1. Cliente deve estar logado</li> </ol>
<b>Fluxo de eventos principal:</b> <ol style="list-style-type: none"> <li>1. Cliente acessa tela de alterar perfil</li> <li>2. Cliente altera informações que deseja</li> <li>3. Cliente aperta confirmar</li> <li>4. Sistema valida novas informações</li> <li>5. Sistema salva informações</li> </ol>
<b>Pós-condições:</b> <ol style="list-style-type: none"> <li>1. Sistema atualiza BD</li> </ol>
<b>Fluxo alternativo: caso usuário desista de alterar informações</b> <ol style="list-style-type: none"> <li>1. Cliente aperta botão retornar</li> <li>2. Sistema retornar cliente para tela principal</li> </ol>

Tabela 12. CSU Alterar perfil

Tabela feita pelos integrantes do grupo

#### 4.4 Diagrama de classes

Esse diagrama de classe se refere ao aplicativo EZPay e tem como objetivo mapear o sistema através de seus atributos, métodos e relações, de forma clara e estruturada.

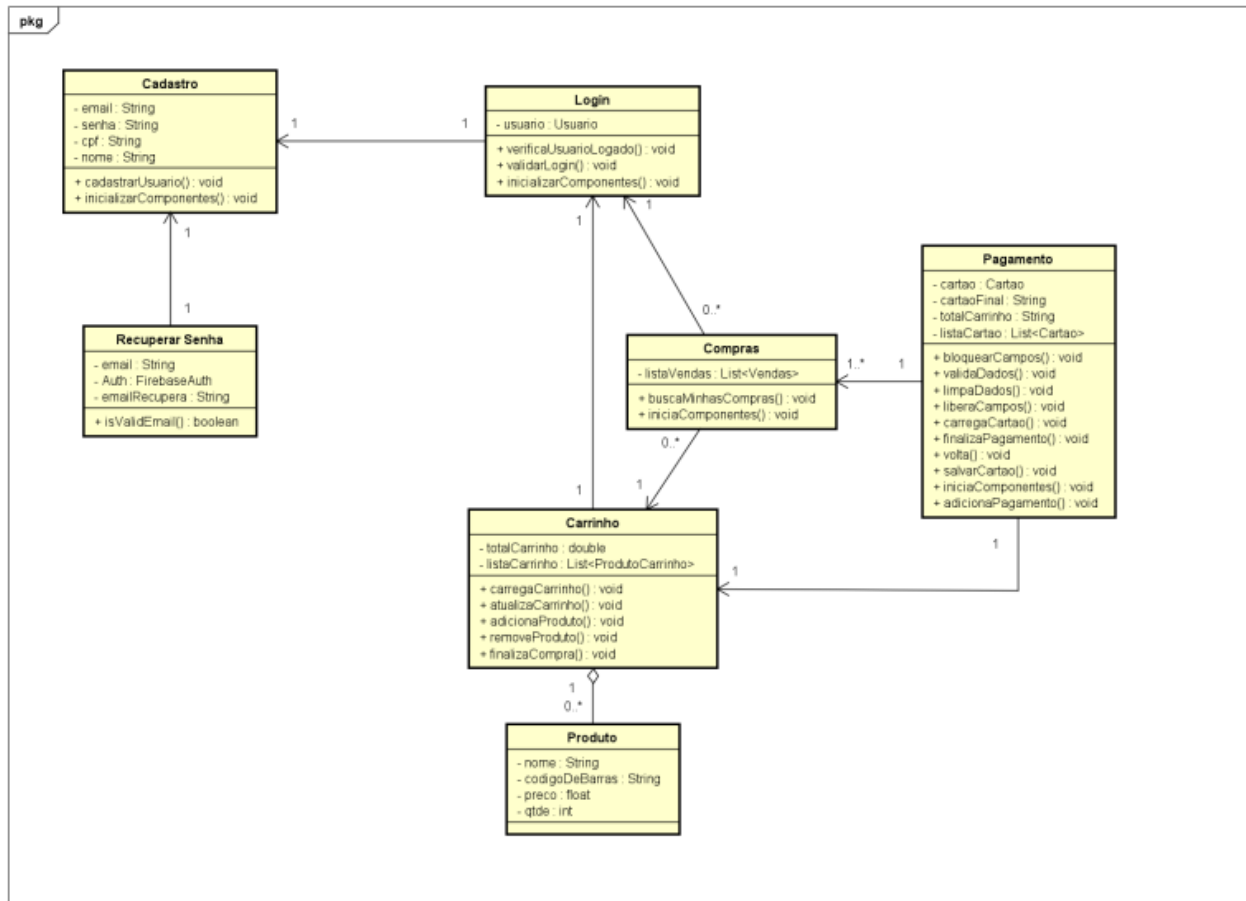


Figura 8 DC EZPay

Figura feita pelos integrantes do grupo

## 4.5 Diagrama de sequência

### 4.5.1 Criar Conta

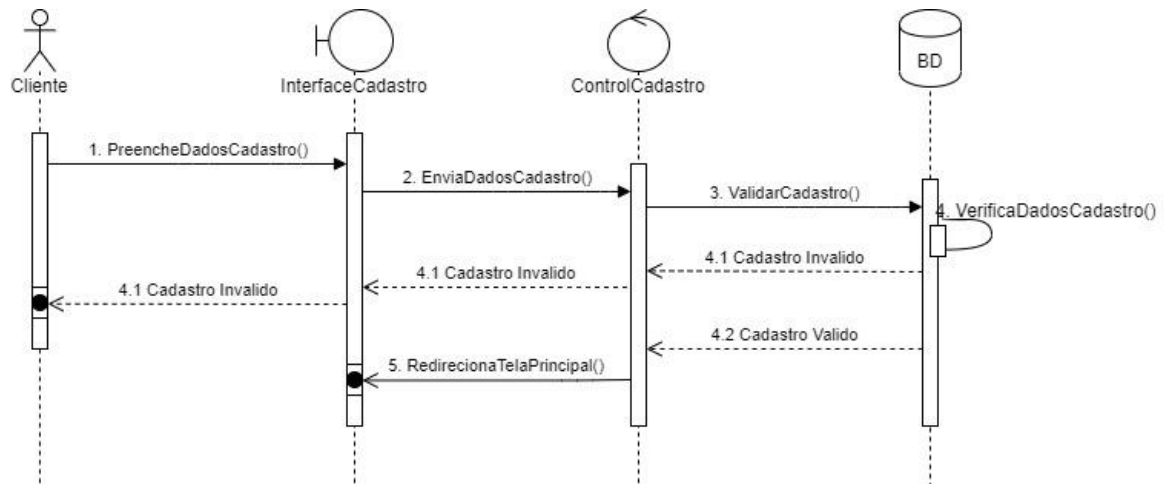


Figura 9. DS Criar conta

Figura feita pelos integrantes do grupo

### 4.5.2 Login App

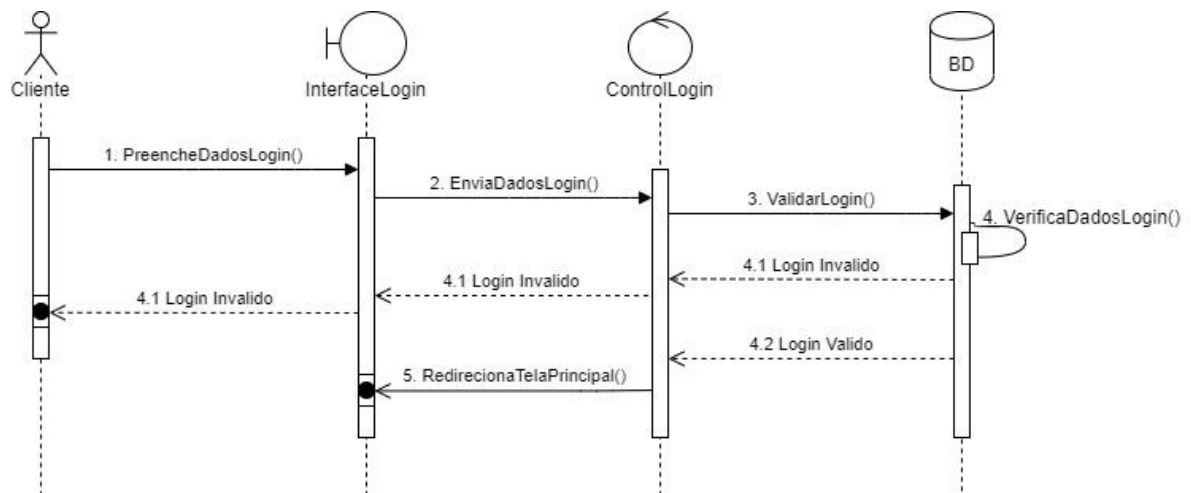


Figura 10. DS Login App

Figura feita pelos integrantes do grupo

### 4.5.3 Adicionar Produtos ao Carrinho

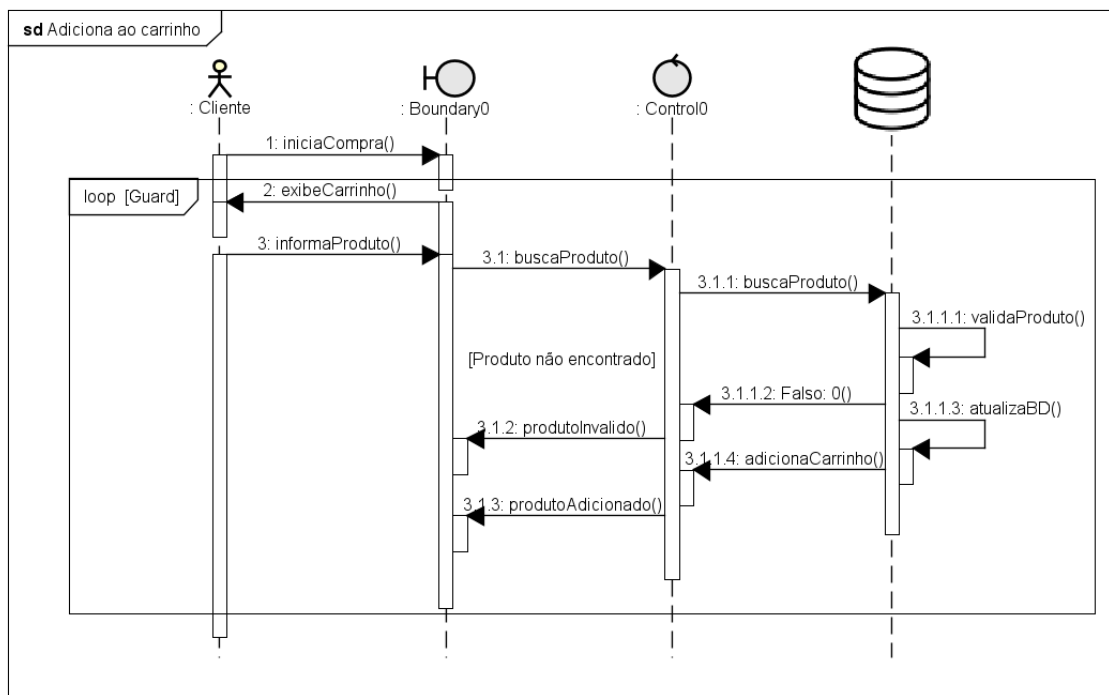


Figura 11. DS Adicionar Produtos ao Carrinho

Figura feita pelos integrantes do grupo

### 4.5.4 Altera Quantidade de Produtos

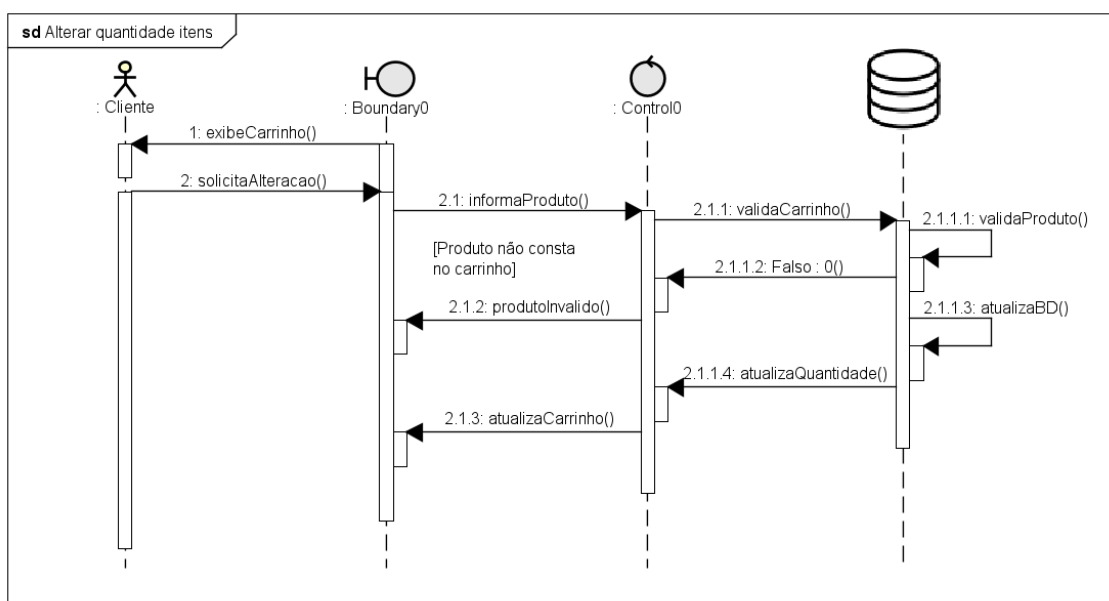


Figura 12. DS Altera Quantidade de Produtos

Figura feita pelos integrantes do grupo

### 4.5.5 Remove Produtos do Carrinho

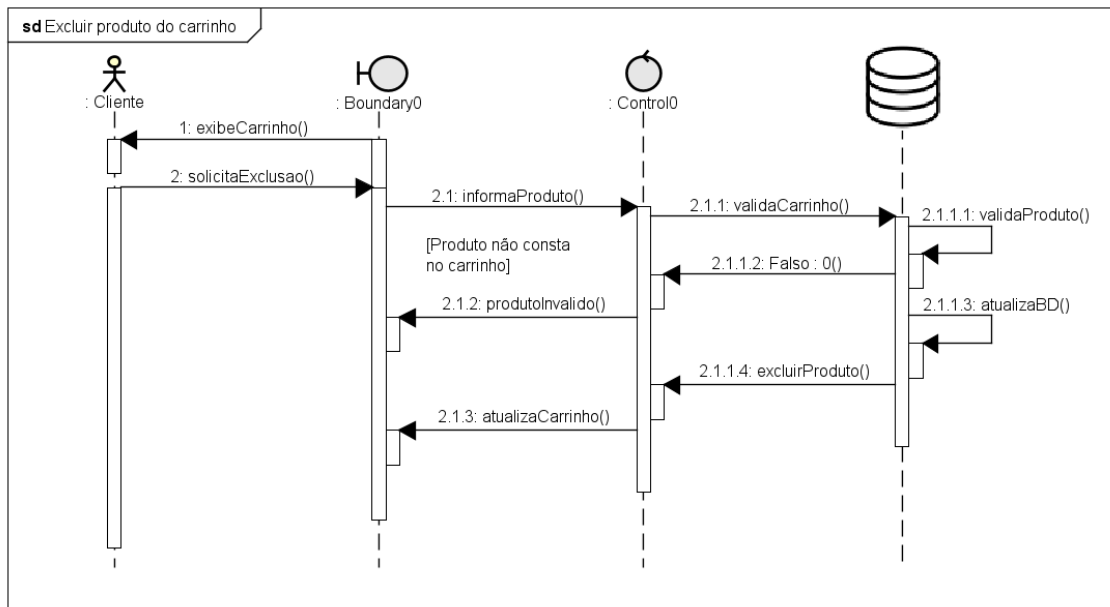


Figura 13. DS Remove Produtos do Carrinho

Figura feita pelos integrantes do grupo

### 4.5.6 Finalizar Compra

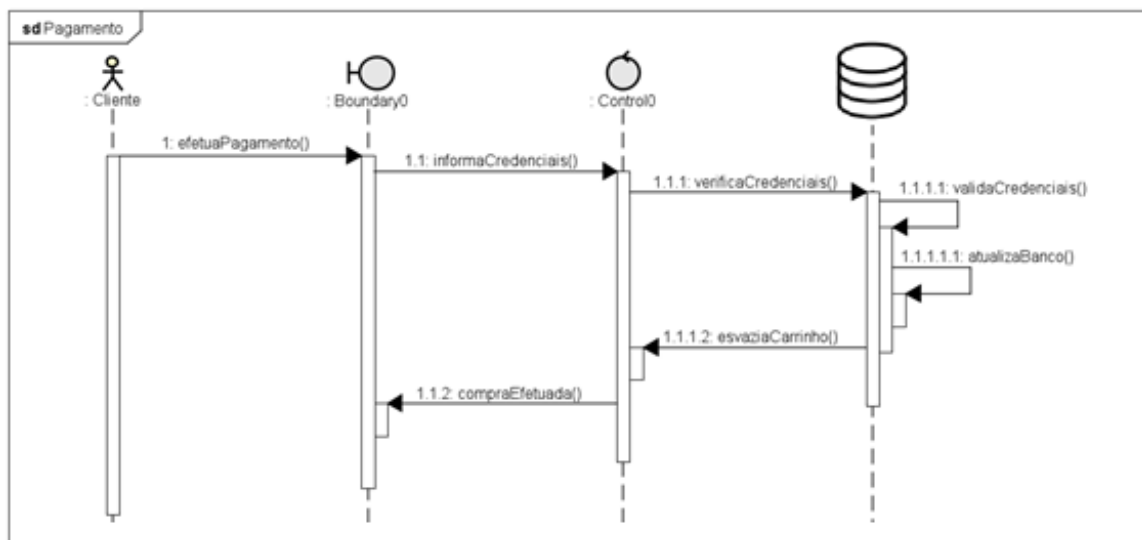


Figura 14. DS Finalizar Compra

Figura feita pelos integrantes do grupo

### 4.5.7 Adicionar Cartão

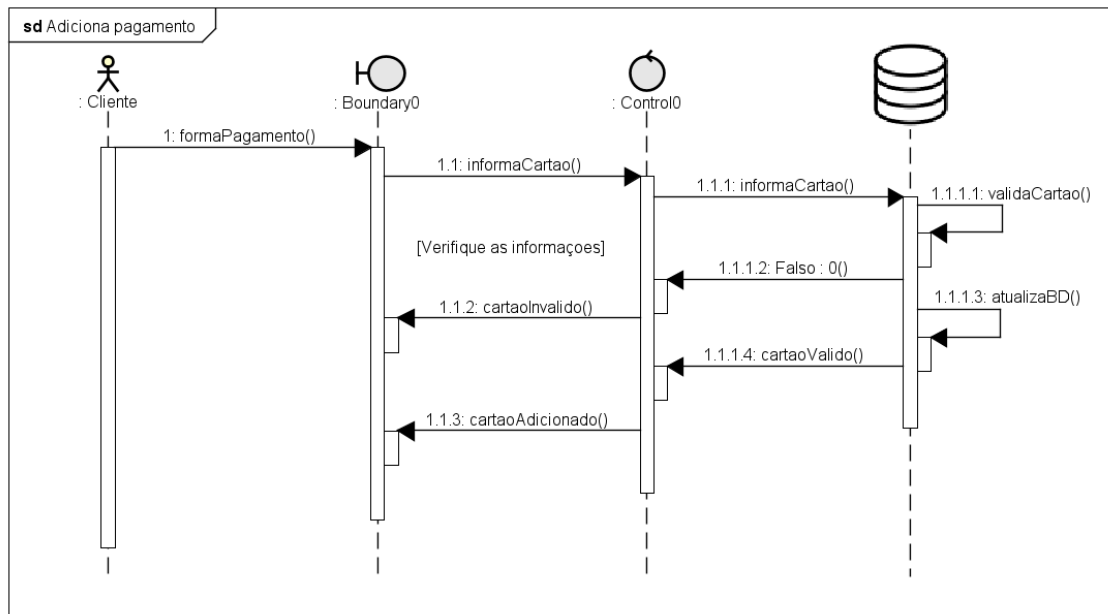


Figura 15. DS Adicionar Cartão

Figura feita pelos integrantes do grupo

### 4.5.8 Recuperar Senha

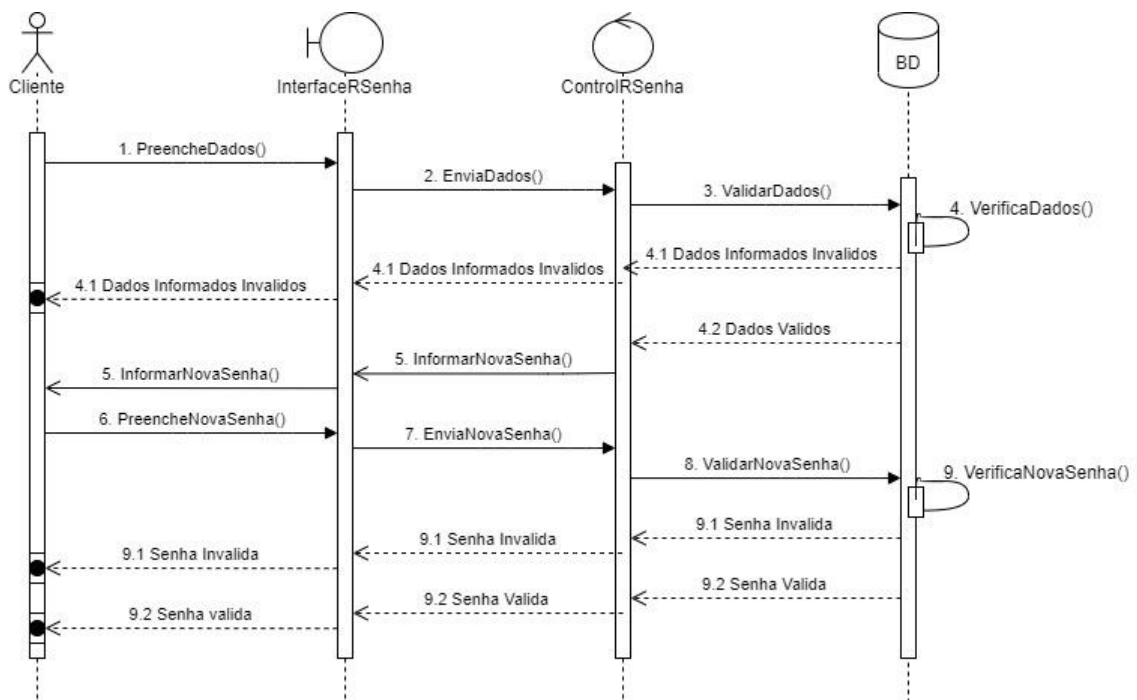


Figura 16. DS Recuperar Senha

Figura feita pelos integrantes do grupo

## 4.5.9 Alterar Perfil

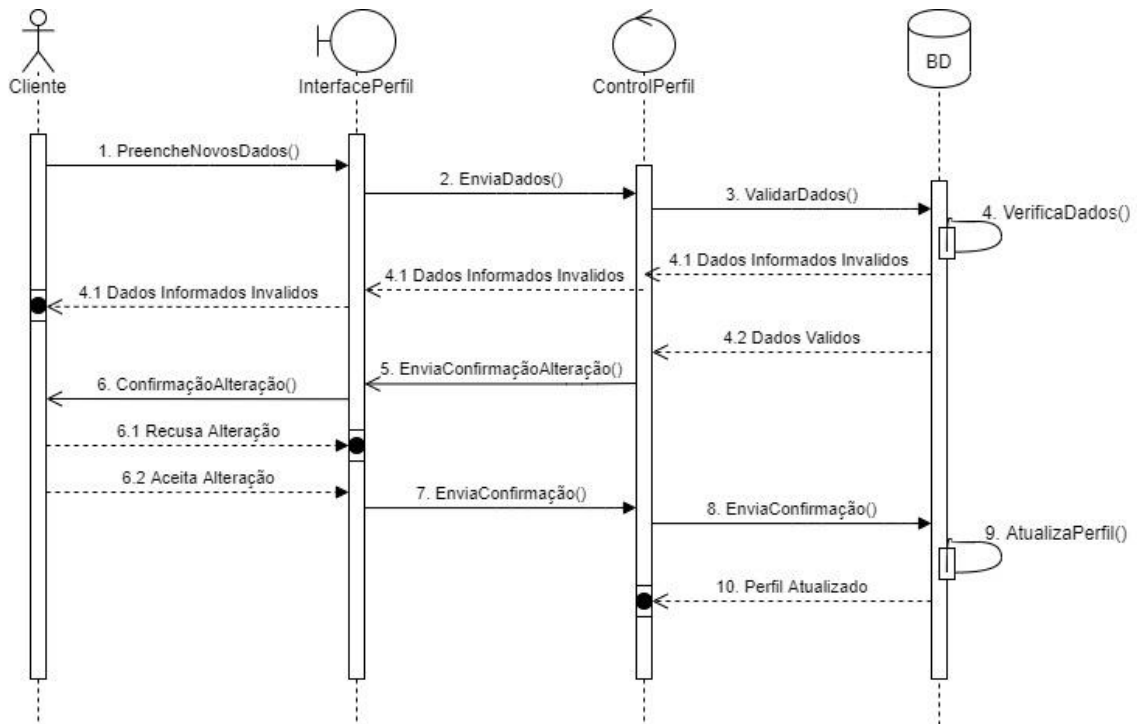


Figura 17. DS Alterar Perfil

Figura feita pelos integrantes do grupo

## 5 APLICATIVO EZPAY MERCADO (LOJA)

Este capítulo tem como o objetivo apresentar as especificações do aplicativo EZPay Mercado, mostrando requisitos funcionais e não funcionais, casos de uso e diagramas, mostrando assim como cada função funciona dentro do aplicativo.

### 5.1 Engenharia de requisitos

Foram levantados requisitos funcionais e não funcionais baseados em pesquisas, para atender as necessidades dos aplicativos.

#### 5.1.1 Requisitos funcionais

<b>ID do Requisito</b>	<b>Descrição</b>	<b>Caso de uso Relacionado</b>
RF001	Realizar Cadastro de usuários	CSU001
RF002	Verificar dados do usuário	CSU002
RF003	Adicionar produtos ao App	CSU003
RF004	Remover produtos do App	CSU005
RF005	Alterar produtos do App	CSU004
RF006	Exibir relatório de compra	CSU006
RF007	Exibir Relatório de estoque	CSU007

Tabela 13. Requisitos funcionais EZPay Mercado

Tabela feita pelos integrantes do grupo



## 5.1.2 Requisitos não funcionais

<b>ID do Requisito</b>	<b>Categoria do Requisito</b>	<b>Descrição</b>
RNF001	Usabilidade	O sistema deve possuir um layout interativo, com ferramentas de fácil entendimento, assim, sendo manipulável por qualquer tipo de usuário.
RNF002	Desempenho	Tempo de resposta para ações internas deverá ser de no máximo 10 segundos
RNF003	Interoperabilidade	O sistema deverá se comunicar com o banco Firebase
RNF004	Implementação	O sistema deverá ser desenvolvido na linguagem Java
RNF005	Confiabilidade	O sistema deverá ter alta disponibilidade, por exemplo, 99% do tempo
RNF006	Portabilidade	O sistema deverá executar em qualquer smartphone que possuir Android com versões acima de 8.0
RNF007	Interoperabilidade	O sistema deverá se comunicar com uma biblioteca para escanear código de barras
RNF008	Manutenibilidade	A manutenção do aplicativo deve ser rápida e eficiente para evitar inatividade do sistema
RNF009	Segurança	O sistema deve criptografar os dados sensíveis do usuário

Tabela 14. Requisitos não funcionais EZPay Mercado

Tabela feita pelos integrantes do grupo

## 5.2 Especificações dos casos de uso

As especificações dos casos de uso consistem em apresentar o diagrama de casos de uso do aplicativo e o ator que os utiliza.

### 5.2.1 Diagrama de casos de uso

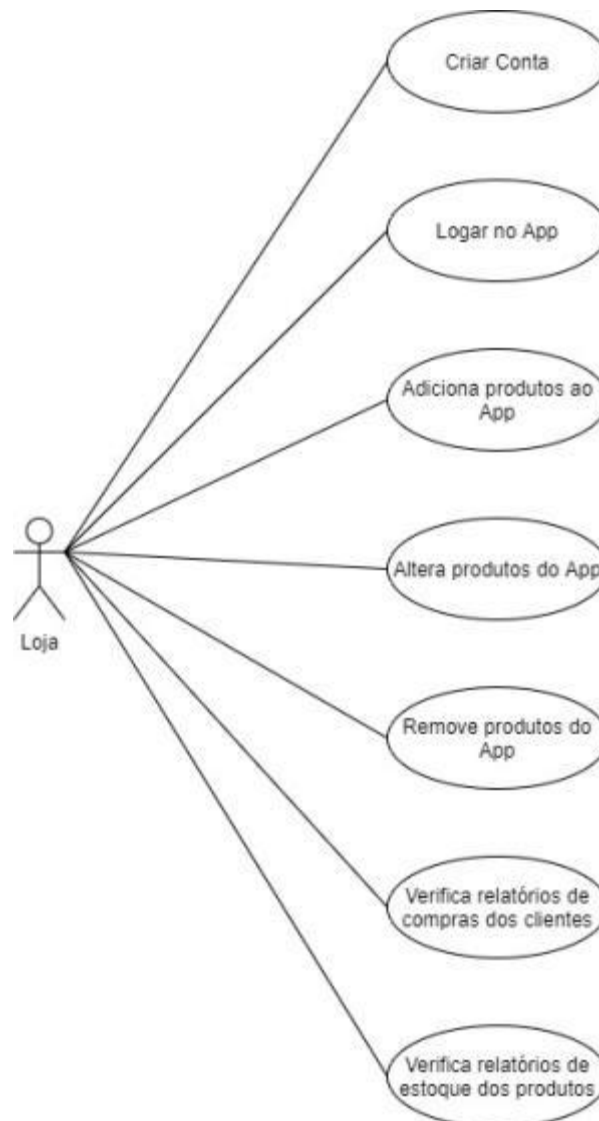


Figura 18. Diagrama de casos de uso loja

Figura feita pelos integrantes do grupo

## 5.2.2 Documentação do Ator

<b>Ator:</b> Loja
<b>Resumo:</b>  O Loja é o ator que utiliza o aplicativo para verificar os relatórios de compras dos clientes, também adiciona e remove produtos ao aplicativo e verifica os relatórios de estoque dos produtos adicionados ao aplicativo.
<b>Responsabilidades:</b>  <ol style="list-style-type: none"><li>1. Criar conta</li><li>2. Fazer login</li><li>3. Adicionar produtos ao aplicativo</li><li>4. Altera produtos do aplicativo</li><li>5. Remover produtos do aplicativo</li><li>6. Verifica relatórios de compras dos clientes</li><li>7. Verifica relatórios de estoque dos produtos</li></ol>
<b>Ambiente Físico:</b>  O usuário utiliza o aplicativo no próprio celular.

Tabela 15. ator (Loja)

Tabela feita pelos integrantes do grupo

### 5.3 Casos de uso

Os casos de uso descrevem sobre como funciona cada ação feita no sistema do aplicativo.

#### 5.3.1 Criar Conta

<b>Caso de Uso: Criar Conta</b>
<b>ID:</b> CSU001
<b>Resumo:</b> Esse caso de uso é responsável por criar uma conta nova para cada loja que deseja utilizar o aplicativo
<b>Atores primários:</b> Loja
<b>Atores secundários:</b> Back-end (sistema)
<b>Pré-condições:</b> <ol style="list-style-type: none"> <li>1. Loja deve ter um e-mail</li> <li>2. Esse e-mail não pode ter sido cadastrado antes</li> </ol>
<b>Fluxo de eventos principal:</b> <ol style="list-style-type: none"> <li>1. Loja acessa o aplicativo.</li> <li>2. Sistema exibe a tela de login e opção de nova conta</li> <li>3. Loja escolhe criar conta</li> <li>4. Sistema exibe a página de cadastro</li> <li>5. Loja preenche a página de cadastro</li> <li>6. Sistema guarda as informações no banco de dados</li> </ol>
<b>Pós-condições:</b> <ol style="list-style-type: none"> <li>1. Depois da conta ser criada, o sistema abre a tela principal</li> <li>2. Sistema atualiza BD</li> </ol>
<b>Fluxo alternativo:</b> <ol style="list-style-type: none"> <li>1. Sistema pergunta se o cliente deseja criar uma conta.</li> <li>2. Se não, sistema exibe a página de login.</li> </ol>
<b>Fluxo de exceção: caso o e-mail criado já seja cadastrado</b> <ol style="list-style-type: none"> <li>1. Sistema informa ao usuário que o e-mail já foi cadastrado</li> <li>2. Sistema retorna ao passo 4 do fluxo principal</li> </ol>

Tabela 16. CSU Criar conta - Loja

Tabela feita pelos integrantes do grupo

## 5.3.2 Login App

<b>Caso de Uso: Login App</b>
<b>ID:</b> CSU002
<b>Resumo:</b> Este caso de uso representa a ação da loja logar no app para acessar a tela principal
<b>Atores primários:</b> Loja
<b>Atores secundários:</b> Back-end (sistema)
<b>Pré-condições:</b> <ol style="list-style-type: none"> <li>1. Loja deve ter um cadastro</li> </ol>
<b>Fluxo de eventos principal:</b> <ol style="list-style-type: none"> <li>1. Loja inicializa o aplicativo.</li> <li>2. Loja se depara com a tela de login.</li> <li>3. Loja insere o e-mail, senha e nível de acesso.</li> <li>4. Sistema verifica credenciais.</li> <li>5. Sistema redireciona para a tela principal e o caso de uso termina.</li> </ol>
<b>1 - Fluxo alternativo: caso os dados inseridos sejam inválidos</b> <ol style="list-style-type: none"> <li>1. Sistema informa que os dados digitados são incorretos</li> <li>2. Sistema retorna ao passo 3 do fluxo principal</li> </ol>
<b>2 - Fluxo alternativo: caso o cadastro não exista</b> <ol style="list-style-type: none"> <li>1. Sistema informa que o e-mail não foi cadastrado</li> <li>2. Sistema redireciona cliente para tela de cadastro</li> </ol>

Tabela 17. CSU Login app - Loja

Tabela feita pelos integrantes do grupo

## 5.3.3 Adiciona Produtos ao Aplicativo

<b>Caso de Uso:</b> Adicionar produtos ao aplicativo
<b>ID:</b> CSU003
<b>Resumo:</b> Este caso de uso representa a ação da loja de adicionar novos produtos no app
<b>Atores primários:</b> Loja
<b>Atores secundários:</b> Back-end (sistema)
<b>Pré-condições:</b> <ol style="list-style-type: none"> <li>1. Loja deve estar logada</li> </ol>
<b>Fluxo de eventos principal:</b> <ol style="list-style-type: none"> <li>1. Loja acessa tela de produtos</li> <li>2. Loja preenche os campos com os dados do novo produto</li> <li>3. Loja aperta o botão de adicionar produto</li> <li>4. Sistema valida os campos digitados</li> <li>5. Sistema adiciona o novo produto no BD</li> </ol>
<b>Pós-condições:</b> <ol style="list-style-type: none"> <li>1. Sistema atualiza BD</li> </ol>
<b>Fluxo alternativo: caso a loja não queira mais adicionar o produto</b> <ol style="list-style-type: none"> <li>1. Loja aperta botão retornar</li> <li>2. Sistema retorna loja a tela principal</li> </ol>
<b>Fluxo de exceção: caso o produto já exista no BD</b> <ol style="list-style-type: none"> <li>1. Sistema informa que o produto já existe no BD</li> <li>2. Sistema redireciona loja a tela de alterar produto</li> </ol>

Tabela 18. CSU Adicionar produtos ao aplicativo

Tabela feita pelos integrantes do grupo

## 5.3.4 Altera Produtos do Aplicativo

<b>Caso de Uso:</b> Alterar produtos do aplicativo
<b>ID:</b> CSU004
<b>Resumo:</b> Este caso de uso representa a ação da loja de alterar os produtos que já estão dentro do app
<b>Atores primários:</b> Loja
<b>Atores secundários:</b> Back-end (sistema)
<b>Pré-condições:</b> <ol style="list-style-type: none"> <li>1. Loja tem que estar logado</li> <li>2. Produto já deve ter sido criado</li> </ol>
<b>Fluxo de eventos principal:</b> <ol style="list-style-type: none"> <li>1. Loja acessa tela de alteração de produtos</li> <li>2. Loja altera dados do produto</li> <li>3. Sistema valida novos dados</li> <li>4. Sistema altera produto</li> </ol>
<b>Pós-condições:</b> <ol style="list-style-type: none"> <li>1. Sistema atualiza BD</li> </ol>
<b>Fluxo alternativo: caso a loja não queira mais alterar o produto</b> <ol style="list-style-type: none"> <li>1. Loja aperta botão retornar</li> <li>2. Sistema retorna loja a tela principal</li> </ol>
<b>Fluxo de exceção: caso o produto não exista no BD</b> <ol style="list-style-type: none"> <li>1. Sistema informa que o produto não já existe no BD</li> <li>1. Sistema redireciona loja a tela de produto</li> </ol>

Tabela 19. CSU Alterar produtos do aplicativo

Tabela feita pelos integrantes do grupo

## 5.3.5 Remove Produtos do Aplicativo

<b>Caso de Uso:</b> Remover produtos do aplicativo
<b>ID:</b> CSU005
<b>Resumo:</b> Este caso de uso representa a ação da loja de remover produtos do aplicativo
<b>Atores primários:</b> Loja
<b>Atores secundários:</b> Back-end (sistema)
<b>Pré-condições:</b> <ol style="list-style-type: none"> <li>1. Loja tem que estar logado</li> <li>2. Produto já deve ter sido criado</li> </ol>
<b>Fluxo de eventos principal:</b> <ol style="list-style-type: none"> <li>1. Loja acessa tela de produto</li> <li>2. Loja coloca o nome do produto</li> <li>3. Loja seleciona a opção de remoção de produto</li> <li>4. Sistema valida remoção</li> <li>5. Sistema remove produto</li> </ol>
<b>Pós-condições:</b> <ol style="list-style-type: none"> <li>1. Sistema atualiza BD</li> </ol>
<b>Fluxo alternativo: caso a loja não queira mais remover o produto</b> <ol style="list-style-type: none"> <li>1. Loja aperta botão retornar</li> <li>2. Sistema retorna loja a tela principal</li> </ol>
<b>Fluxo de exceção: caso o produto não exista no BD</b> <ol style="list-style-type: none"> <li>1. Sistema informa que o produto não já existe no BD</li> <li>2. Sistema informa que o produto pode ser adicionado</li> </ol>

Tabela 20. CSU Remover produtos do aplicativo

Tabela feita pelos integrantes do grupo



## 5.3.6 Verificar Relatórios de Compra

<b>Caso de Uso:</b> Verificar relatórios de compra
<b>ID:</b> CSU006
<b>Resumo:</b> Este caso de uso representa a ação da loja de verificar os relatórios de compra
<b>Atores primários:</b> Loja
<b>Atores secundários:</b> Back-end (sistema)
<b>Pré-condições:</b> <ol style="list-style-type: none"> <li>1. Loja deve estar logado</li> <li>2. O BD do sistema deve ter pelo menos uma compra</li> </ol>
<b>Fluxo de eventos principal:</b> <ol style="list-style-type: none"> <li>1. Loja acessa tela de relatórios</li> <li>2. Loja seleciona relatórios de compra</li> <li>3. Sistema gerar relatório</li> <li>4. Sistema apresenta relatório na tela</li> </ol>
<b>Fluxo de exceção: caso o BD não tenha nenhuma compra</b> <ol style="list-style-type: none"> <li>1. Sistema informa que não compras no BD portanto não há o que gerar</li> <li>2. Retornar ao passo 1 do fluxo principal</li> </ol>

Tabela 21. CSU Verificar relatórios de compra

Tabela feita pelos integrantes do grupo

## 5.3.5 Verificar Relatórios de Estoque

<b>Caso de Uso:</b> Verificar relatórios de estoque
<b>ID:</b> CSU007
<b>Resumo:</b> Este caso de uso representa a ação da loja de verificar os relatórios de compra
<b>Atores primários:</b> Loja
<b>Atores secundários:</b> Back-end (sistema)
<b>Pré-condições:</b> <ol style="list-style-type: none"> <li>1. Loja deve estar logado</li> <li>2. O BD do sistema deve ter pelo menos um produto dentro do app</li> </ol>
<b>Fluxo de eventos principal:</b> <ol style="list-style-type: none"> <li>1. Loja acessa tela de relatórios</li> <li>2. Loja seleciona relatório de estoque</li> <li>3. Sistema gerar relatório</li> <li>4. Sistema apresenta relatório na tela</li> </ol>
<b>Fluxo de exceção: caso o BD não tenha nenhum produto</b> <ol style="list-style-type: none"> <li>1. Sistema informa que não produtos no BD portanto não há o que gerar</li> <li>2. Retornar ao passo 1 do fluxo principal</li> </ol>

Tabela 22. CSU Verificar relatórios de estoque

Tabela feita pelos integrantes do grupo

## 5.4 Diagrama de classes

Esse diagrama de classe se refere ao aplicativo EZPay Mercado e tem como objetivo mapear o sistema através de seus atributos, métodos e relações, de forma clara e estruturada.

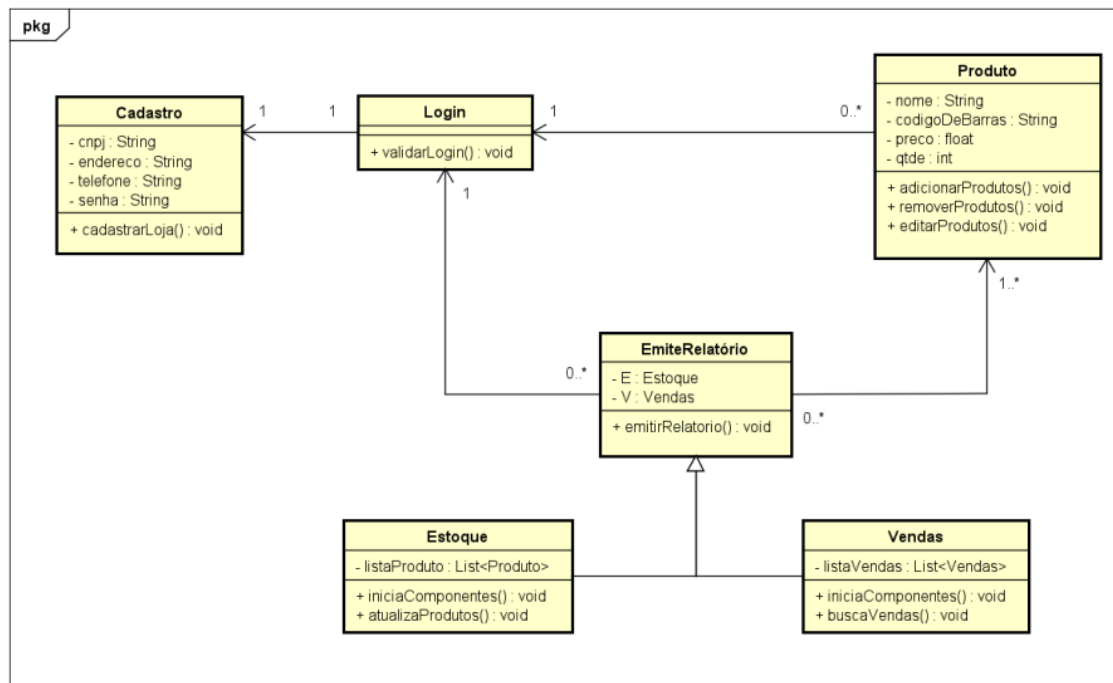


Figura 19. DC EZPay Mercado

Figura feita pelos integrantes do grupo

## 5.5 Diagrama de sequência

### 5.5.1 Criar Conta

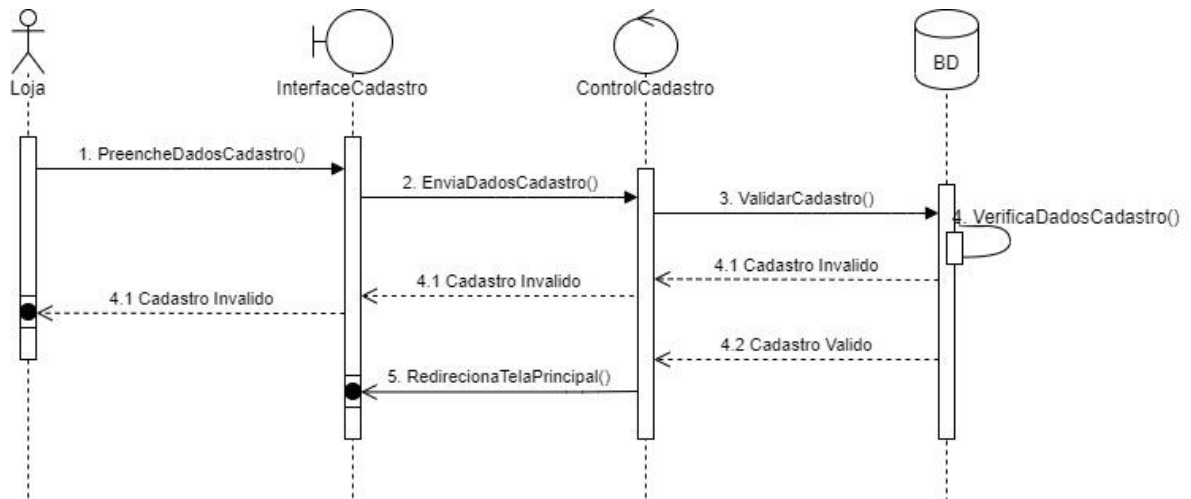


Figura 20. DS Criar Conta - Loja

Figura feita pelos integrantes do grupo

### 5.5.2 Login App

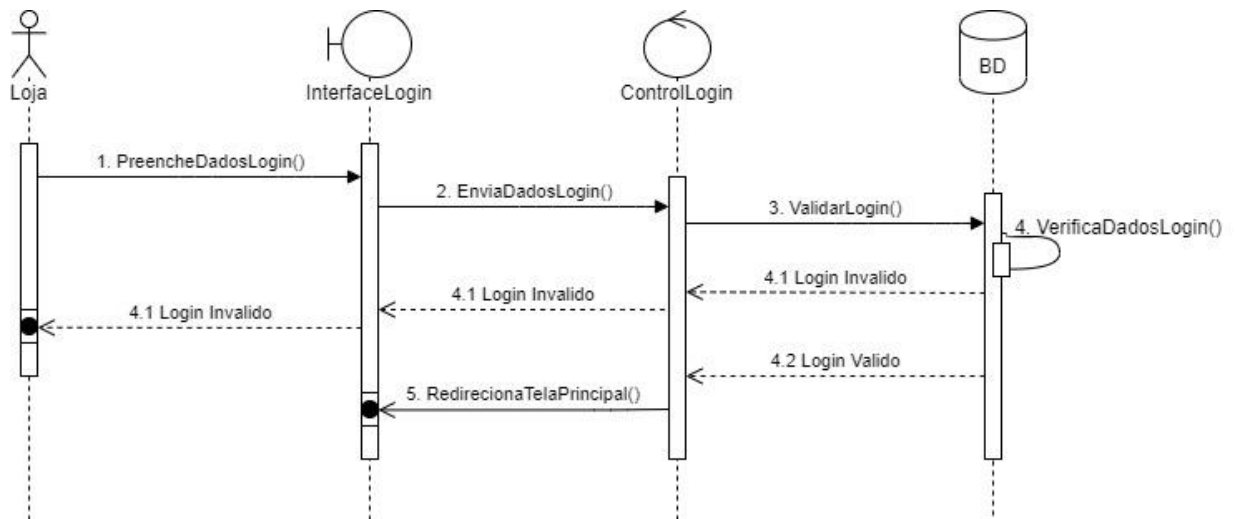


Figura 21. DS Login App - Loja

Figura feita pelos integrantes do grupo

### 5.5.3 Adicionar Produtos ao Aplicativo

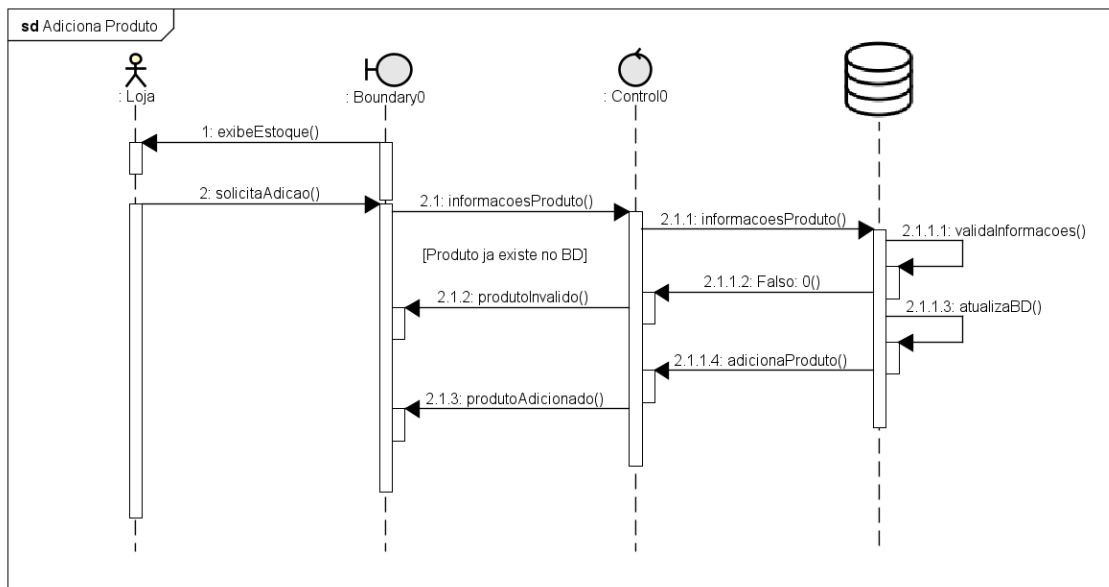


Figura 22. DS Adicionar Produtos ao Aplicativo

Figura feita pelos integrantes do grupo

### 5.5.4 Altera Produtos do Aplicativo

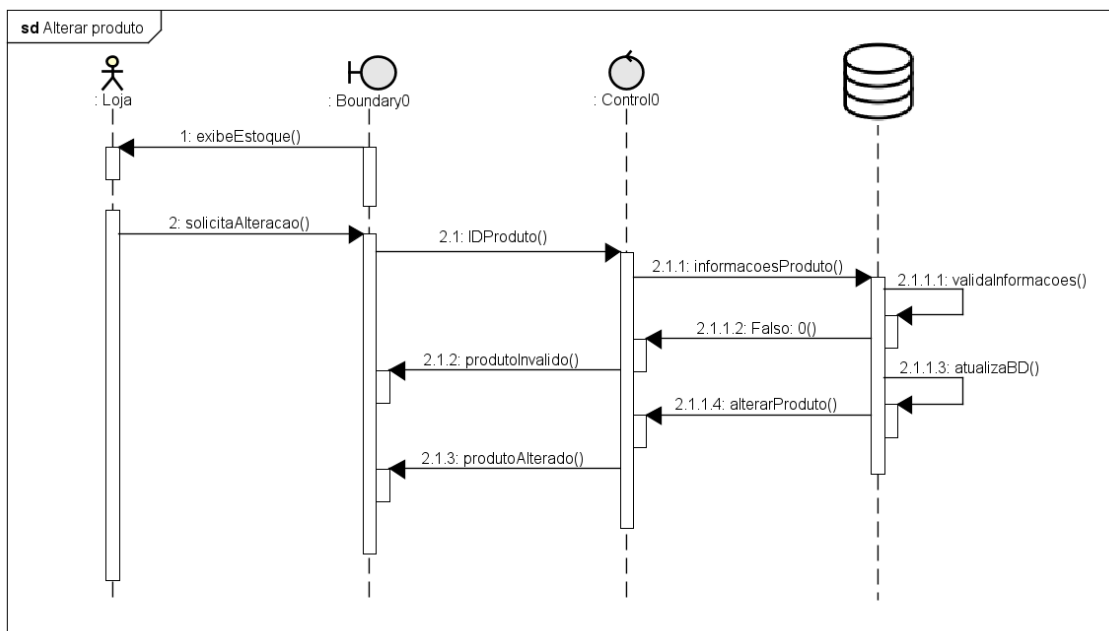


Figura 23. DS Altera Produtos do Aplicativo

Figura feita pelos integrantes do grupo

### 5.5.5 Remove Produtos do Aplicativo

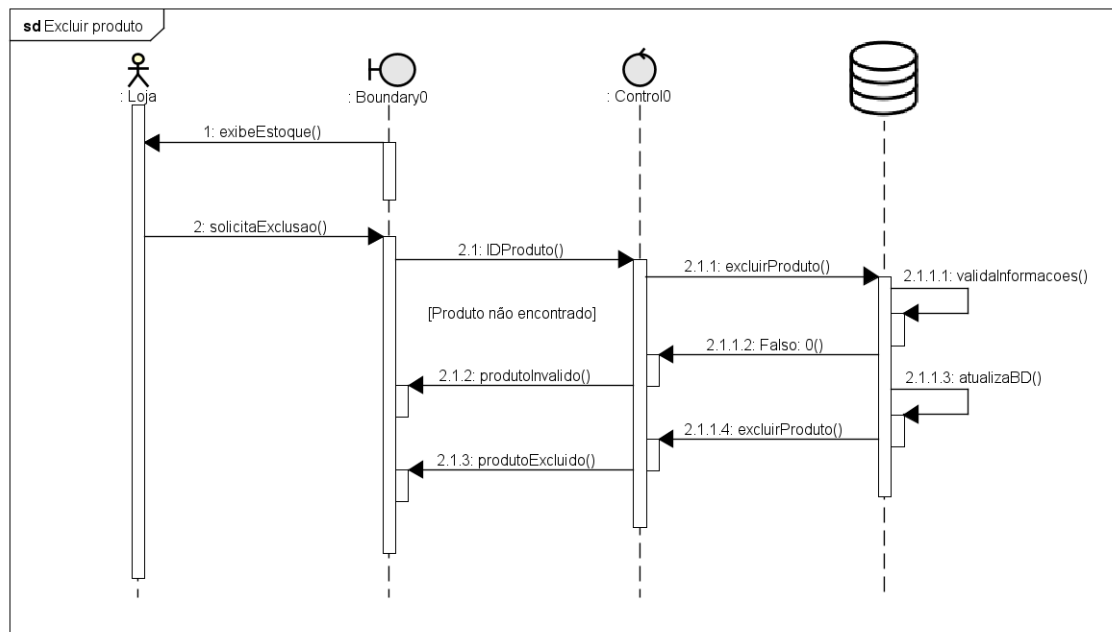


Figura 24. DS Remove Produtos do Aplicativo

Figura feita pelos integrantes do grupo

### 5.5.6 Verifica Relatórios de Compra

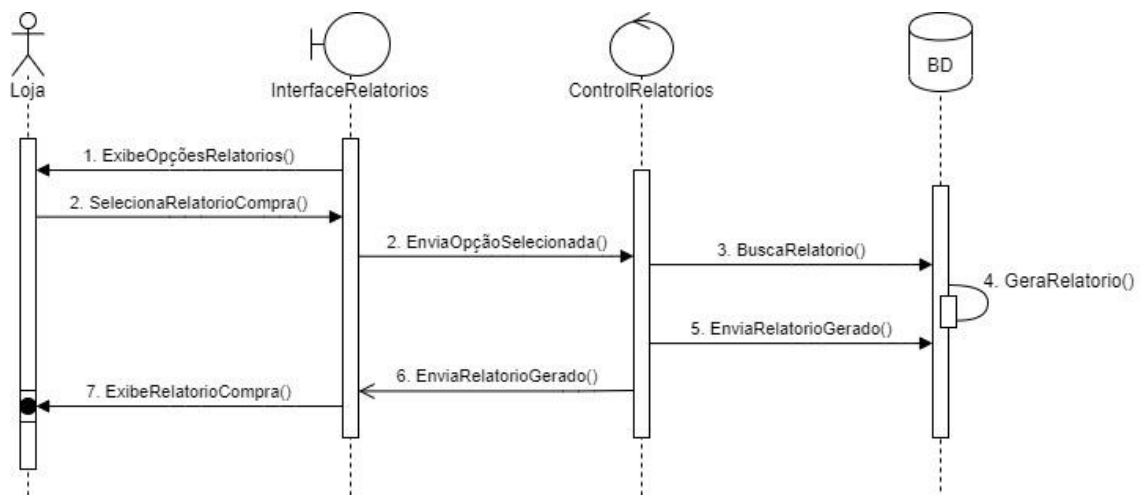


Figura 25. DS Verifica Relatórios de Compra

Figura feita pelos integrantes do grupo

## 5.5.7 Verifica Relatórios de Estoque

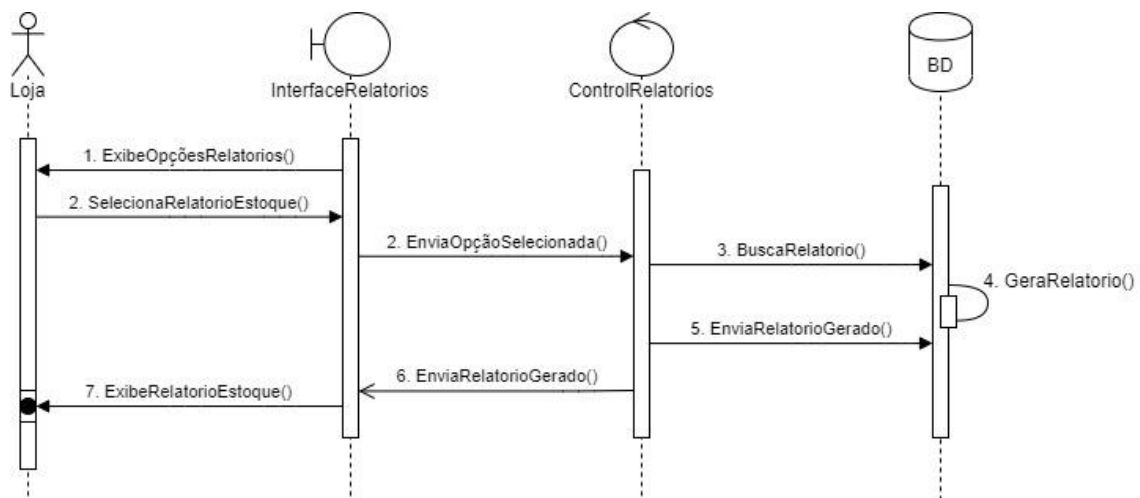


Figura 26. DS Verifica Relatórios de Estoque

Figura feita pelos integrantes do grupo

## 6 PLANO DE TESTES FUNCIONAIS

O objetivo deste capítulo consiste em apresentar os testes que foram realizados, descrevendo cada teste.

Para realização dos testes foi utilizada a técnica funcional que tem como objetivo entender o comportamento do sistema, explorando suas funcionalidades e identificando possíveis problemas na execução.

### 6.1 Testes realizados

#### 6.1.1 Recuperando a Senha

Resumo: Caso ocorra falha ao logar no aplicativo em detrimento da senha estar incorreta.

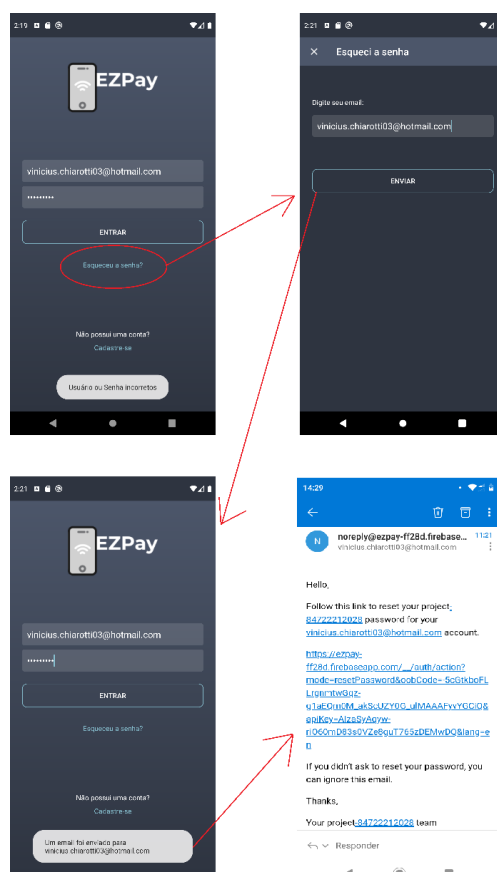


Figura 27. Recuperando Senha (Teste)

Figura feita pelos integrantes do grupo



Ações a serem realizadas:

1. Usuário deve clicar na parte do “Esqueceu a senha?”
2. Usuário deve preencher o e-mail e clicar em “Enviar”.
3. Após isso o usuário deverá entrar em seu e-mail e clicar na mensagem que foi enviada, para assim recuperar a senha.

Conclusão: O teste foi realizado com êxito, sem nenhum erro.

### 6.1.2 Selecionando o produto

Resumo: Quando o cliente queira adicionar o produto ao seu “carrinho”, seja escaneando ou buscando o produto pelo nome.

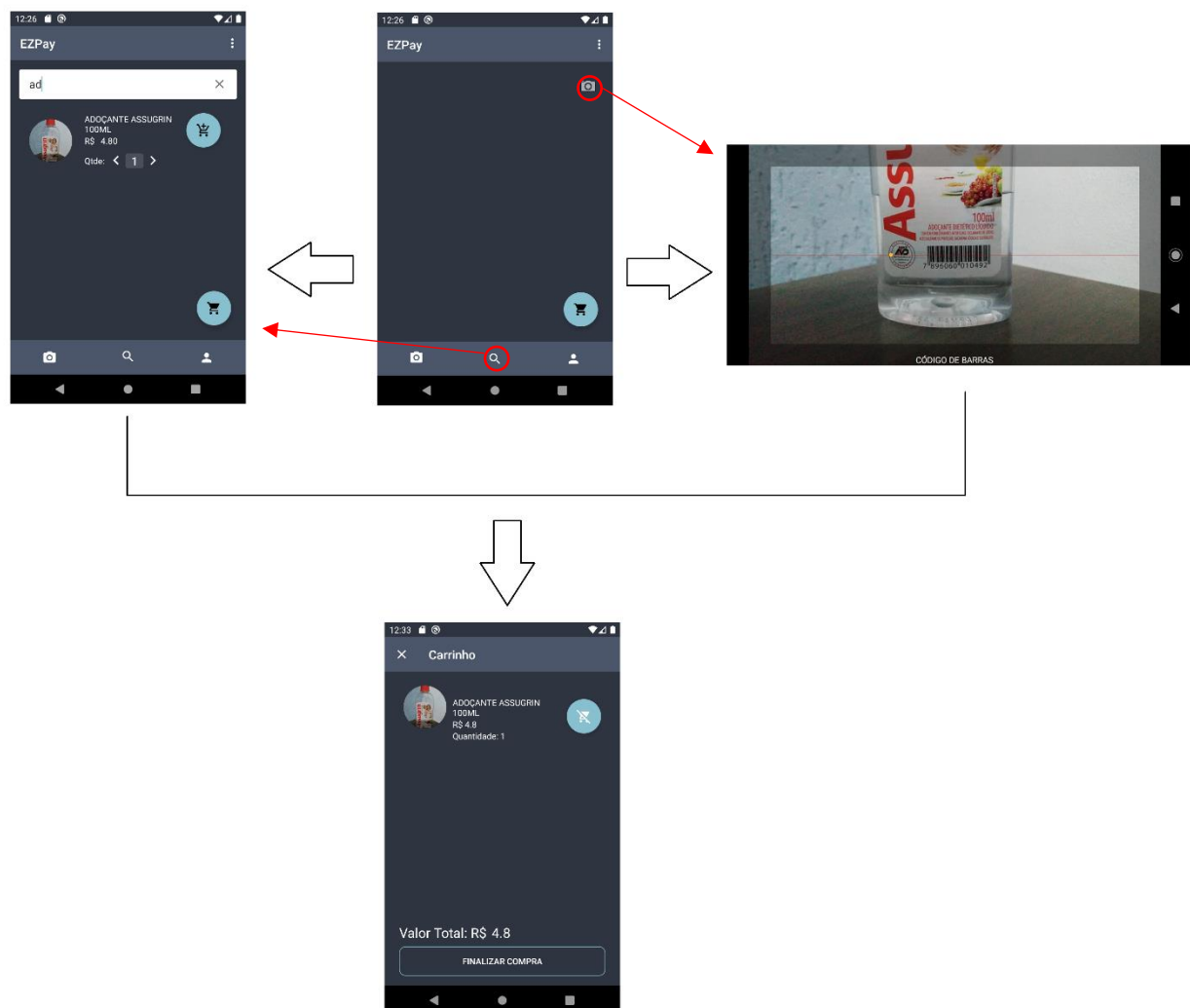


Figura 28. Selecionando Produto (teste)

Figura feita pelos integrantes do grupo

Ações a serem realizadas:

1. Usuário possuirá dois caminhos para selecionar o produto.

1.1. Usuário terá que clicar na câmera(canto superior direito), permitir o acesso a câmera do dispositivo e por fim posicionar o leitor no código de barra do produto.

1.2. Usuário terá que clicar no ícone da lupa e assim digitar o nome do produto almejado.

2. Usuário deverá selecionar a quantidade de produto desejado e clicar no ícone do carrinho, para assim adicionar o produto em sua lista de compras.

Conclusão: O teste foi realizado com êxito, sem nenhum erro.

### 6.1.3 Realizando a compra

Resumo: Quando o cliente já adicionou produtos ao “carrinho” e deseja finalizar a compra.



Figura 29. Realizando a Compra (Teste)

Figura feita pelos integrantes do grupo

Ações a serem realizadas:

1. Usuário deverá clicar no botão “Finalizar compra”.

2. Usuário deverá preencher uma forma de pagamento a seu critério ou selecionar o cartão cadastrado anteriormente.
3. Usuário deverá clicar no botão “Check” para confirmar a compra, em seguida o sistema exibirá uma confirmação na tela.

Conclusão: O teste foi realizado com êxito, sem nenhum erro.

Como melhoria será adicionada mais formas de pagamento em atualizações futuras.

#### 6.1.4 Verificando relatórios

Resumo: Quando o Usuário do aplicativo do mercado deseja verificar os relatórios de estoque ou de compras.



Figura 30. Verificando Relatórios (Teste)

Figura feita pelos integrantes do grupo

Ações a serem realizadas:

1. Usuário dentro do menu terá duas opções: Relatório de Estoque e Relatório de Vendas.

1.1. Usuário seleciona Relatório de Vendas e o sistema abrirá uma tela retornando todas as vendas realizadas.

1.2. Usuário seleciona relatório de Estoque e o sistema abrirá uma tela retornando todos os produtos disponíveis em estoque.

Conclusão: O teste foi realizado com êxito, sem nenhum erro.

Como melhorias futuras serão adicionados maiores detalhes aos relatórios de vendas.

## 7 TELAS

Este capítulo tem como o objetivo de mostrar as telas de cada um dos aplicativos desenvolvidos.

### 7.1 EZPay



Figura 31. Tela de Login

Figura feita pelos integrantes do grupo



5:34

EZPay

Nome completo

CPF

E-mail

Senha

Confirmar senha

CADASTRE-SE

Figura 32. Tela de Cadastro

Figura feita pelos integrantes do grupo

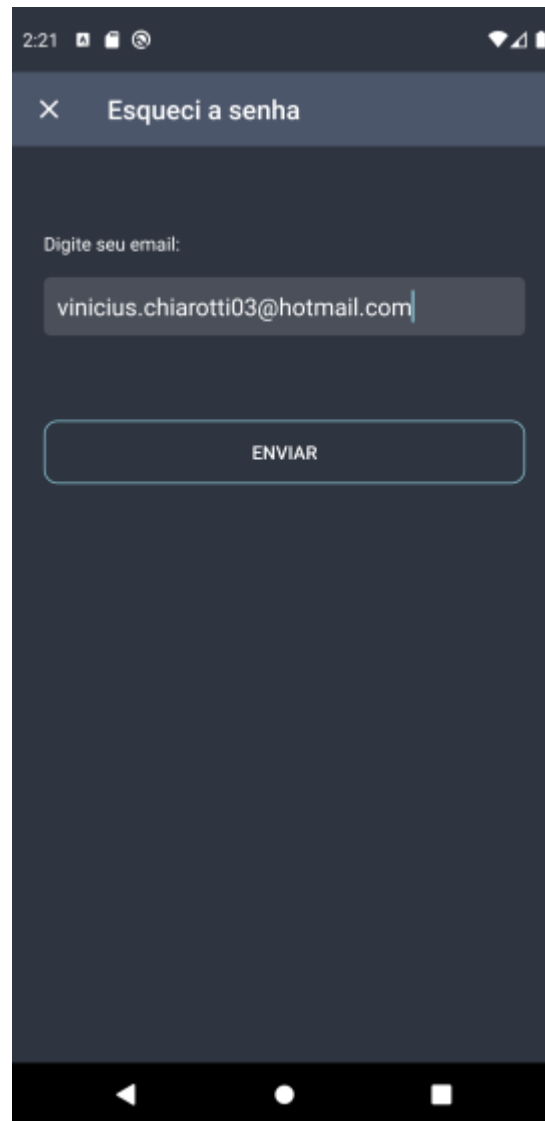


Figura 33. Tela de Recuperar Senha

Figura feita pelos integrantes do grupo



Figura 34. Tela Principal

Figura feita pelos integrantes do grupo



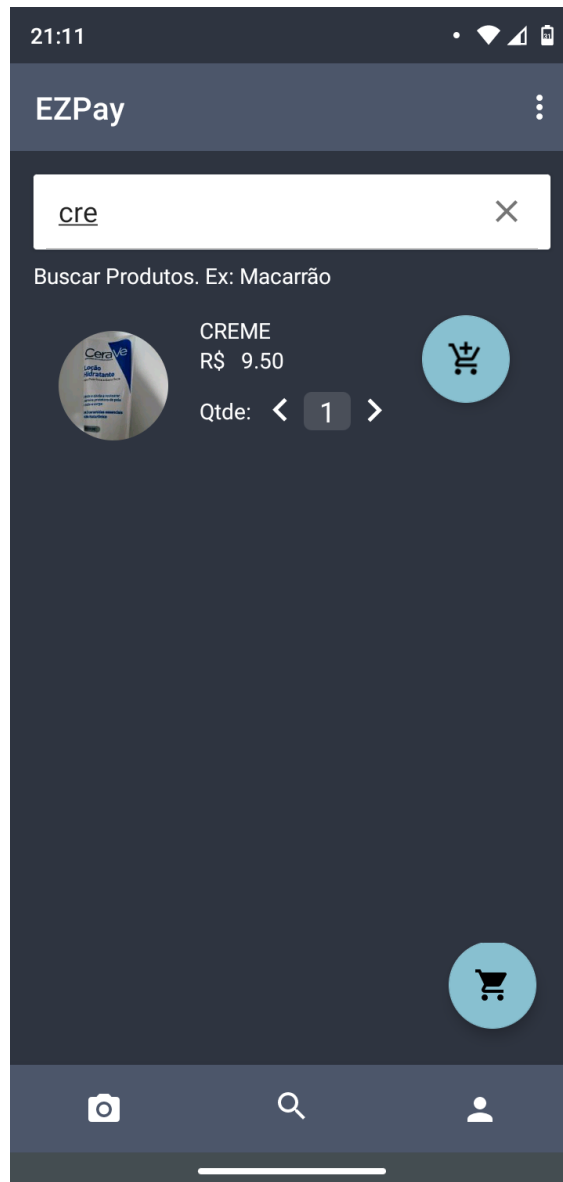


Figura 35. Tela de Busca

Figura feita pelos integrantes do grupo

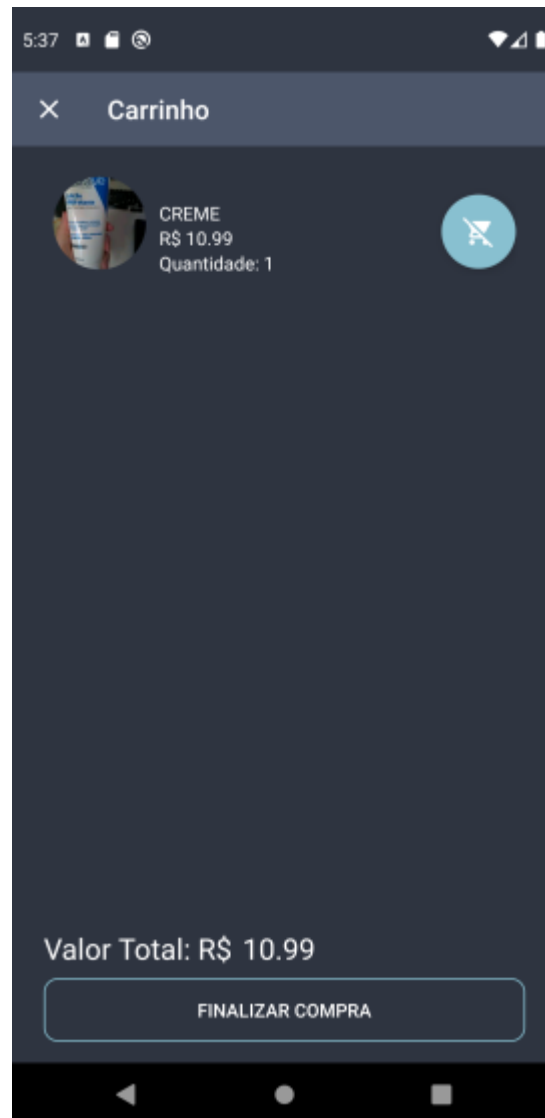


Figura 36. Tela do Carrinho

Figura feita pelos integrantes do grupo



Figura 37. Tela de Finalizar Compra  
Figura feita pelos integrantes do grupo

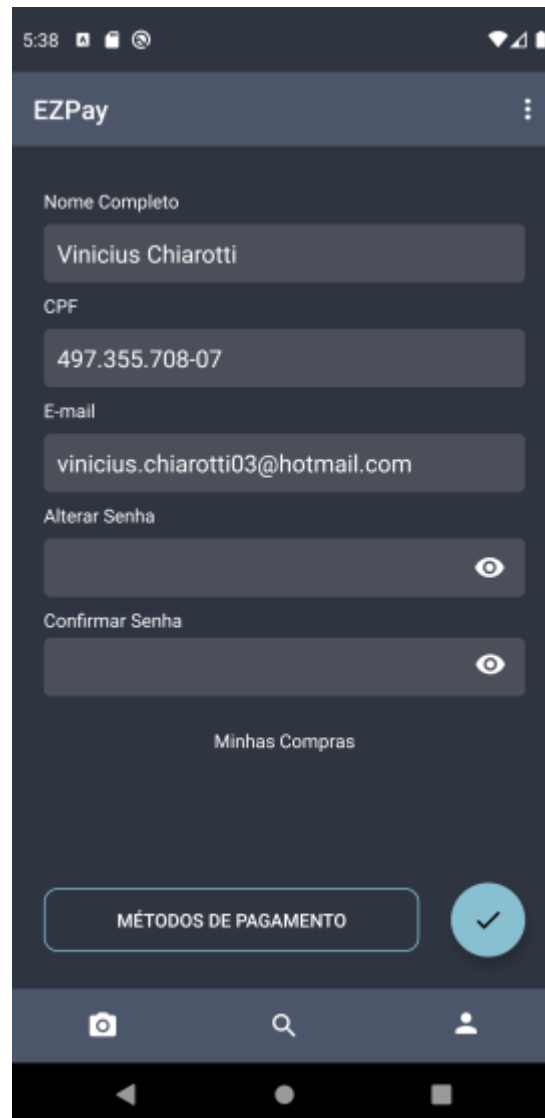


Figura 38. Tela de Perfil

Figura feita pelos integrantes do grupo

## 7.2 EZPay Mercado



Figura 39. Tela Principal EZPay Mercado

Figura feita pelos integrantes do grupo

21:16

✕ Adicionar Produtos

Camera

Galeria

Nome do Produto

Preço

Código de Barras

Quantidade

ADICIONAR

Figura 40. Tela de Adicionar Produtos  
Figura feita pelos integrantes do grupo

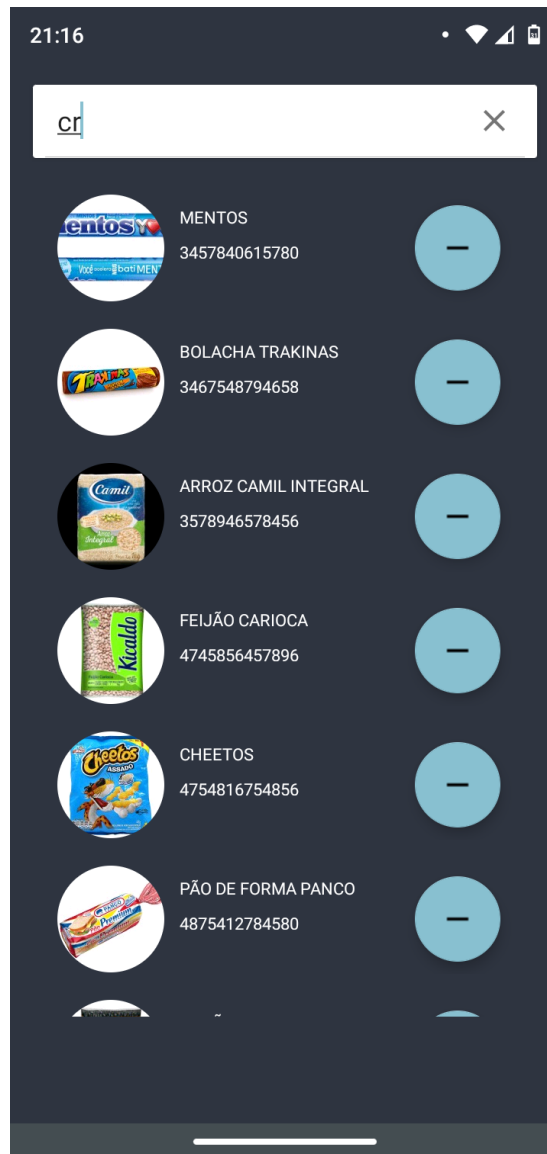


Figura 41. Tela de Remover Produtos  
Figura feita pelos integrantes do grupo

21:16

× Atualizar Produto



Camera

Galeria

Nome do Produto

CHEETOS

Código de Barras

4754816754856

Preço

1.99

Quantidade

500

ATUALIZAR

Figura 42. Tela de Alterar Produtos

Figura feita pelos integrantes do grupo





Figura 43. Tela de Relatório de Estoque

Figura feita pelos integrantes do grupo



Figura 44. Tela de Relatório de Vendas

Figura feita pelos integrantes do grupo

## **8 PROCEDIMENTO DE INSTALAÇÃO**

Este capítulo tem o objetivo de descrever sobre os procedimentos de instalação dos aplicativos EZPay e EZPay Mercado, descrevendo ação que deve ser realizada para conseguir utilizá-los.

Pré-requisito: possuir um aparelho celular com sistema operacional superior ou equivalente a versão 8.0.

### **7.1 App EZPay**

O usuário deve possuir conta na Play Store e assim que logado deverá buscar pelo aplicativo na loja, feito isso basta selecionar o aplicativo e clicar na opção “Instalar”.

### **7.2 App EZPay Mercado**

O usuário receberá um link para instalação mediante à contratação do serviço.

## CONSIDERAÇÕES FINAIS

Após o estudo das rotinas diárias de várias pessoas, percebe-se que elas acabam cada vez mais agitadas assim resultando em pouco tempo hábil para realizar tarefas que deveriam ser simples como ir ao supermercado. Os aplicativos mobile vieram para suprir essas necessidades.

Com este trabalho de conclusão de curso, foi possível atingir objetivos elaborados durante o planejamento, com a criação dos aplicativos EZPay e EZPay Mercado que tem funções diferentes de outros aplicativos deste segmento.

Os testes realizados foram feitos nas principais funcionalidades de cada um dos aplicativos para melhor entendimento dos mesmos, feitos dentro de um ambiente controlado todos obtiveram êxito enquanto executados.

Em trabalhos futuros, pode ser realizada uma análise em mercados de grande porte, assim expandindo o aplicativo para suprir a alta demanda, também pode ser feito uma melhoria na parte gráfica tornando o aplicativo cada vez mais intuitivo e de fácil acesso a qualquer tipo de público. Uma otimização no código fonte seria fundamental assim deixando os aplicativos ainda mais ágeis e responsivos, tanto em smartphones mais simples quanto nos mais sofisticados, essa otimização possibilitaria também a adição de uma API abrangendo diversas formas de pagamento, com isso a criptografia dos dados sensíveis dentro do banco de dados seria ainda mais rigorosa.

## REFERÊNCIAS

CANALTECH, **O que é API?** 2019 Disponível em: <<https://canaltech.com.br/software/o-que-e-api/>> Acesso em: 31/01/2020

HARADA, E. **O que é o Android Studio, ferramenta criada para desenvolver apps mobile,** 2019 Disponível em : <<https://www.tecmundo.com.br/software/146361-o-android-studio-ferramenta-criada-desenvolver-apps-mobile.htm>> Acesso em: 31/01/2020

JUSTASKGEMALTO, **O QUE É O Amazon Go e como funciona?** 2017 Disponível em: <https://www.justaskgemalto.com/br/o-que-e-o-amazon-go-e-como-funciona/> Acesso em: 05/04/2020

MADUREIRA, D. **Aplicativo nativo, web app ou aplicativo híbrido.** 2017 Disponível em: <https://usemobile.com.br/aplicativo-nativo-web-hibrido/> Acesso em: 05/04/2020

ORLANDI, C. **Firestore: serviços, vantagens, quando utilizar e integrações.** 2019 Disponível em: <<https://blog.rocketseat.com.br/firebase/>> Acesso em: 31/01/2020

PEREIRA, A. P. **O que é Java?** 2019 Disponível em: <<https://www.tecmundo.com.br/programacao/2710-o-que-e-java-.htm>> Acesso em: 31/01/2020

PEREIRA, A. P. **O que é XML?** 2009 Disponível em: <<https://www.tecmundo.com.br/programacao/1762-o-que-e-xml-.htm>> Acesso em: 29/02/2020

ZANINI. **O que são os aplicativos nativos e como eles funcionam.** 2017

Disponível em: <https://blog.za9.com.br/o-que-sao-os-apps-nativos-e-como-eles-funcionam/> Acesso em: 05/04/2020

## REFERÊNCIAS BIBLIOGRÁFICAS

DEITEL, paul; DEITEL, Harvey. *Java®: Como Programar*. 10 ed. Pearson  
Universidades, 2016