

UNIVERSIDADE FEDERAL DE SERGIPE CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA DEPARTAMENTO DE COMPUTAÇÃO

Interface móvel para o LudiiPrice: um aplicativo para a recomendação de produtos de supermercados baseado nas melhores ofertas próximas do usuário

Trabalho de Conclusão de Curso

Rodrigo Fontes Martins



São Cristóvão - Sergipe

UNIVERSIDADE FEDERAL DE SERGIPE CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA DEPARTAMENTO DE COMPUTAÇÃO

Rodrigo Fontes Martins

Interface móvel para o LudiiPrice: um aplicativo para a recomendação de produtos de supermercados baseado nas melhores ofertas próximas do usuário

Trabalho de Conclusão de Curso submetido ao Departamento de Computação da Universidade Federal de Sergipe como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador(a): Leonardo Nogueira Matos Coorientador(a): Thiago Dias Bispo

Agradecimentos

Quero agradecer a todos os meus parentes, amigos e amigos de amigos, que não só me ajudaram a testar o aplicativo, como também enviaram várias notas fiscais para ajudar a minha pesquisa, além claro de acompanharem esta jornada comigo.

Gostaria de agradecer também ao meu orientador Leonardo e ao meu coorientador Thiago, pelos seus direcionamentos ao longo destes últimos meses.

Resumo

Em um cenário de compras de supermercado, poucos são os métodos eficientes para saber os

preços dos produtos antes de sair de casa. Panfletos, anúncios na televisão, conversa com amigos

e até alguns aplicativos possuem o problema da desatualização na exibição dos preços.

O objetivo do projeto discutido neste documento, chamado de LudiiPrice, é exibir preços

atualizados de produtos de supermercados próximos ao usuário, para ajudar na economia de suas

compras. Para atingir este objetivo, foi realizada uma pesquisa de mercado, com o intuito de

conhecer as soluções já disponíveis para este problema.

Ao fim da pesquisa e do processo de desenvolvimento, foi possível criar um aplicativo não só

capaz de informar os preços de um produto em diferentes estabelecimentos comerciais, como

também capaz de criar listas de compras e exibir gráficos para enriquecer a experiência do usuário.

Como todo software, ainda é possível enriquecer a experiência da sua utilização, acrescentando

técnicas para manter os preços mais atualizados ou novas funcionalidades.

Palavras-chave: Sistemas *mobile*. Nota fiscal eletrônica. *Flutter*.

Abstract

In a scenario of supermarket purchases, few are the efficient methods to know the prices of

products before leaving home. Flyers, television advertisements, conversations with friends and

even some applications have the problem of being outdated on it's price exhibition.

The objective of the project discussed on this document, called LudiiPrice, is to show up-to-date

prices of supermarket products next to the user, to help on the savings of your purchases. To

reach this goal, a market research was made, with the intention to know the already available

solutions for this problem.

At the end of the research and of the development process, it was possible to create an application

that not only is able to inform the prices of a product on different commercial establishments,

but is also able to create shopping lists and show graphs to enrich the user experience. Like all

software, it is still possible to enrich it's utilization experience, adding techniques to keep the

prices up-to-date or new functionalities.

Keywords: Mobile systems. Electronic sale slip. Flutter.

Lista de ilustrações

Figura 1 – Exemplo de nota fiscal eletrônica	16
Figura 2 – Exemplo de nota fiscal eletrônica detalhada	17
Figura 3 – Generalização entre atores de um diagrama de casos de uso	18
Figura 4 – Exemplo de um diagrama de casos de uso	19
Figura 5 – Exemplo de diagrama de sequência	20
Figura 6 – Exemplo de loops e condicionais em um diagrama de sequência	2
Figura 7 – Exemplo de criação e exclusão de objetos em um diagrama de sequência	22
Figura 8 – Exemplo de arquivo robots.txt	23
Figura 9 – Exemplo de GUI	24
Figura 10 – Tela do Pinngo (à esquerda) e tela do Economiza Club (à direita)	28
Figura 11 – Tela do Revela Preço (à esquerda) e tela do Menor Preço Brasil (à direita)	29
Figura 12 – Faixa etária dos participantes do questionário	33
Figura 13 – Gênero dos participantes do questionário	34
Figura 14 – Estado de residência dos participantes do questionário	35
Figura 15 – Frequência de compras dos participantes do questionário	36
Figura 16 – Hábito de pesquisa dos participantes do questionário	37
Figura 17 – Meios de pesquisa dos participantes do questionário	38
Figura 18 – Desvantagens dos meios de pesquisa de preços de produtos utilizados pelos	
participantes	39
Figura 19 — Interesse dos participantes do questionário em uma ferramenta que os auxiliem	
na busca por produtos	40
Figura 20 – Diagrama de casos de uso do LudiiPrice	44
Figura 21 – Tela para login (à esquerda) e criação de conta (à direita)	45
Figura 22 – Tela para resetar a senha (à esquerda) e para alterar senha (à direita)	46
Figura 23 – Exibição de produtos populares	47
Figura 24 – Busca por produtos	48
Figura 25 – Exibição dos detalhes de um produto	49
Figura 26 – Criação de uma lista de compras	50
Figura 27 – Remoção de um produto de uma lista de compras	51
Figura 28 – Adição de um produto a uma lista de compras existente	52
Figura 29 – Exclusão de uma lista de compras	53
Figura 30 – Consulta de notas fiscais	54
Figura 31 – Produtos de uma nota fiscal	55
Figura 32 – Exclusão de uma nota fiscal	56
Figura 33 – Leitura do código QR de uma nota fiscal	57
Figura 34 – Histórico de despesas do usuário	58

Figura 35 – Gráfico de preços de uma lista de compras	59
Figura 36 – Gráfico de preços de uma nota fiscal	60
Figura 37 – Diagrama de sequência da leitura de um código QR	62
Figura 38 – Diagrama de sequência da adição de um produto à tela de criação de lista de	
compras	63
Figura 39 – Diagrama de sequência da criação de uma lista de compras	64
Figura 40 – Representação da arquitetura em camadas	66
Figura 41 – Fluxo de dependências das camadas da interface móvel	67
Figura 42 – Módulos da interface <i>mobile</i>	68
Figura 43 – Exemplo da arquitetura em camadas implementada em um módulo	68
Figura 44 – Pasta de arquivos compartilhados da interface <i>mobile</i>	68
Figura 45 – Exemplo da estrutura de pastas de uma tela com maior complexidade	69
Figura 46 – Exemplo da estrutura de pastas da camada de apresentação	70
Figura 47 – Exemplo da estrutura de pastas da camada de domínio	71
Figura 48 – Exemplo do método de um <i>data source</i>	72
Figura 49 – Exemplo da estrutura de pastas da camada de dados	73
Figura 50 – Exemplo de teste unitário para um método da operação de login	75
Figura 51 – Tela de login	87
Figura 52 – Tela de criação de conta	88
Figura 53 — Tela para redefinir a senha	89
Figura 54 – Tela para consultar listas de compras	90
Figura 55 – Tela para criar uma lista de compras	91
Figura 56 – Tela para visualizar os produtos de uma lista de compras	92
Figura 57 – Tela para visualizar o gráfico de uma lista de compras	93
Figura 58 — Exclusão de uma lista de compras	94
Figura 59 – Tela para ler o código qr de uma nota fiscal	95
Figura 60 – Tela para consultar notas fiscais lidas	96
Figura 61 – Tela para ver os detalhes de uma nota fiscal	97
Figura 62 – Tela para ver o gráfico de uma nota fiscal	98
Figura 63 – Tela para ler o código qr de uma nota fiscal	99
Figura 64 – Tela para alterar filtros de busca	100
Figura 65 – Tela para alterar a própria senha	101
Figura 66 – Gráfico de despesas pessoais	102
Figura 67 – Tela de produtos populares	103
Figura 68 – Tela para buscar produtos	104
Figura 69 – Tela para visualizar detalhes de um produto	105

Lista de tabelas

Tabela 1 –	Histórias do usuário	41
Tabela 2 -	Requisitos funcionais do sistema	42
Tabela 3 -	Requisitos não funcionais do sistema	43
Tabela 4 –	Regras de domínio do sistema	43

Lista de abreviaturas e siglas

HTML HyperText Markup Language (do inglês, Linguagem de Marcação de Hiper-

Texto).

API Application Programming Interface (do inglês, Interface de Programação de

Aplicação).

MVP *Minimum Viable Product* (do inglês, Mínimo Produto Viável).

CAPTCHA Completely Automated Public Turing test to tell Computers and Humans

Apart (do inglês, Teste de Turing Público Completamente Automatizado

Para Diferenciação Entre Computadores e Humanos).

QR *Quick Response* (do inglês, Resposta Breve).

UI *User Interface* (do inglês, Interface de Usuário).

SGBD Sistema de gerenciamento de banco de dados.

Sumário

1	Intr	dução	12
	1.1	Objetivos	13
		1.1.1 Objetivo geral	13
		1.1.2 Objetivos específicos	13
	1.2	Estrutura do documento	13
2	Fun	lamentação Teórica	15
	2.1	Nota fiscal eletrônica (NF-e)	15
		2.1.1 Nota fiscal eletrônica detalhada (NF-e detalhada)	16
	2.2	Unified Modeling Language - UML	17
		2.2.1 Diagrama de casos de uso	17
		2.2.1.1 Elementos do diagrama de casos de uso	18
		2.2.1.2 Tipos de relacionamentos	18
		2.2.2 Diagrama de sequência	19
	2.3	Web Scraping	22
		2.3.1 <i>Crawlers</i>	23
		2.3.2 Implicações éticas	23
	2.4	Interface gráfica do utilizador	24
3	Pro	utos relacionados	26
	3.1	Critérios de seleção	26
	3.2	Questões de análise	26
	3.3	Aplicativos relacionados	27
	3.4	Considerações do capítulo	31
4	Apr	sentação e descrição da interface móvel	32
	4.1	Pesquisa de hábitos de compras da população	32
	4.2	Tabela de histórias do usuário	40
	4.3	Levantamento de requisitos	41
		4.3.1 Requisitos Funcionais	41
		4.3.2 Requisitos Não Funcionais	43
			43
	4.4		44
	4.5		44
			44
			46

		4.5.2.1 Exibição de produtos populares	4
		4.5.2.2 Buscar produtos	4
		4.5.2.3 Exibir detalhes de um produto	4
		4.5.3 CRUD de lista de compras	4
		4.5.4 Gerenciamento de notas fiscais	5
		4.5.5 Leitura do código QR de uma nota fiscal	5
		4.5.6 Histórico de despesas	5
		4.5.7 Gráficos de comparação de preço	5
		4.5.7.1 Gráfico de preços de uma lista de compras	5
		4.5.7.2 Gráfico de preços de uma nota fiscal	5
4	4.6	Diagramas de sequência	6
5	Arq	uitetura da interface móvel	6
:	5.1	Tecnologia utilizada	6
		5.1.1 <i>Flutter</i>	6
:	5.2	Estilo arquitetural	6
		5.2.1 A camada de apresentação	6
		5.2.2 A camada de domínio	7
		5.2.3 A camada de dados	7
6	Test	es da aplicação	7
(6.1	Testes unitários	7
	6.2	Testes de usabilidade	7
7	Con	siderações finais	7
,	7.1	Trabalhos futuros	7
Ref	erên	cias	7
Ap	ênd	ices	8
APÍ	ÊND	ICE A Versões das tecnologias utilizadas	8
			8
-		•	8

Anexos				
ANEXO A Manual do usuário da	aplicação	86		
A.1 Introdução		86		
A.2 Como utilizar o aplicativo .		86		
A.2.1 Fazer login		86		
A.2.2 Criar conta		87		
A.2.3 Redefinir senha		88		
A.2.4 Listas de compras.		89		
A.2.5 Notas fiscais		94		
A.2.6 Configurações		99		
A.2.7 Despesas pessoais.		101		
A.2.8 Produtos		102		

1

Introdução

Em um típico cenário de compras de supermercado, um consumidor geralmente anota a sua lista de compras em um papel ou em um aplicativo de celular e vai até o supermercado mais próximo ou até aquele que possua os menores preços segundo sua experiência prévia. O problema dessa rotina, para o consumidor, está em não saber se o preço dos produtos comprados naquele momento são os mais baratos para a sua situção, isto é, se eles apresentam o melhor custo benefício em relação à localização do estabelecimento e ao preço pago pelo produto comprado.

Segundo o IBGE, em 2019, "a parcela das residências em que havia aparelho celular alcançou 94%. Da população com 10 anos ou mais de idade, 81% tinha telefone móvel celular para uso pessoal"(IBGE, 2019). É possível aproveitar-se da praticidade e do alto número de aparelhos móveis (*smartphones*) para desenvolver um aplicativo que armazena e exibe as informações sobre o preço de um produto em diferentes estabelecimentos próximos à localização do consumidor.

Com um aplicativo que possui essas funcionalidades, o consumidor saberá os locais e os preços em diferentes estabelecimentos dos produtos de seu interesse antes mesmo de sair de casa.

O fator determinante para armazenar as informações da compra de um consumidor são as notas fiscais emitidas após a compra, pois estas possuem um código QR, que pode ser facilmente lido por um *smartphone*, e contém o *link* para acessar *online* as informações da compra representadas pela nota fiscal.

Após a leitura do código QR de uma nota fiscal, é possível extrair informações referentes ao preço de cada produto de uma compra e também informações sobre o estabelecimento onde a compra foi realizada. Essas informações podem ser armazenadas em uma base de dados e exibidas em um aplicativo *mobile*. O armazenamento constante das informações de compras de diversos consumidores, através da leitura dos códigos QR das notas fiscais fornecidas por eles, mantém os preços dos produtos atualizados na base de dados da aplicação, garantindo a fidelidade dos preços.

Capítulo 1. Introdução

1.1 Objetivos

1.1.1 Objetivo geral

Desenvolver a interface móvel do projeto chamado de LudiiPrice^{1,2}, que tem o objetivo de ajudar um consumidor a economizar nas compras de produtos de varejo.

1.1.2 Objetivos específicos

- Identificar a situação atual dos principais aplicativos de sugestão de preços de produtos de varejo.
- Especificar os requisitos funcionais, não funcionais e regras de domínio da aplicação.
- Especificar a arquitetura da interface *mobile* da aplicação.
- Desenvolver o protótipo das telas da interface *mobile*.
- Codificar a interface *mobile* da aplicação.
- Integrar a interface mobile à API do LudiiPrice (ARAúJO, 2022) que provê os preços de produtos armazenados.

1.2 Estrutura do documento

Para facilitar a navegação e gerar um melhor entendimento, este documento está estruturado em capítulos, que são:

- Capítulo 1 Introdução
 Introduz o contexto do trabalho.
- Capítulo 2 Fundamentação Teórica
 Apresenta a fundamentação teórica do trabalho.
- Capítulo 3 Produtos relacionados
 Apresenta os aplicativos semelhantes à interface móvel proposta neste documento.
- Capítulo 4 Apresentação e descrição da interface móvel
 Apresenta o questionário de hábitos da compra da população, a tabela de histórias do usuário, o levantamento de requisitos, os casos de uso, as funcionalidades do sistema e os diagramas de sequência da interface móvel.

O nome do produto é a combinação do prefixo "Ludi" (do grupo de pesquisa Ludiico) com a palavra *Price*, que significa preço em inglês.

Atualmente o LudiiPrice é composto de uma interface móvel, uma API (ARAúJO, 2022) e um *web crawler* responsável por coletar informações de produtos através de outros meios que não sejam notas fiscais.

Capítulo 5 - Arquitetura da interface móvel
 Apresenta a arquitetura utilizada no desenvolvimento da interface *mobile* do LudiiPrice.

- Capítulo 6 Testes da aplicação
 Apresenta as metodologias de testes utilizadas no desenvolvimento da interface móvel.
- Capítulo 7 Considerações finais
 Conclui o trabalho baseado nas discussões dos capítulos anteriores e apresenta sugestões para futuros incrementos da interface *mobile* da aplicação.

2

Fundamentação Teórica

Este capítulo tem como objetivo definir os termos técnicos e os termos referentes ao domínio do LudiiPrice.

2.1 Nota fiscal eletrônica (NF-e)

Uma nota fiscal eletrônica (NF-e), segundo Receita Federal (2022) é "um documento de existência apenas digital, emitido e armazenado eletronicamente, com o intuito de documento, para fins fiscais, uma operação de circulação de mercadorias ou uma prestação de serviços, ocorrida entre as partes. Sua validade jurídica é garantida pela assinatura digital do remetente (garantia de autoria e de integridade) e a Autorização de uso fornecida pelo Fisco, antes da ocorrência do fato gerador".

A NF-e já possui validade em todos os estados do Brasil e pertence à legislação brasileira desde 2005, após a aprovação do Ajuste SINIEF 07/05 (RECEITA FEDERAL, 2022).

A obtenção de uma NF-e se dá através da leitura do código QR presente em notas fiscais emitidas por um estabelecimento após a realização de uma compra. Uma NF-e retorna uma página HTML, um exemplo encontra-se na Figura 1.

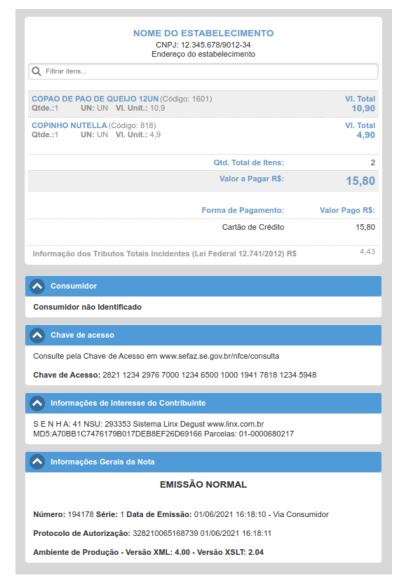


Figura 1 – Exemplo de nota fiscal eletrônica.

Fonte: (SEFAZ, 2022)

2.1.1 Nota fiscal eletrônica detalhada (NF-e detalhada)

Uma nota fiscal eletrônica detalhada (NF-e detalhada) exibe mais informações de uma compra do que uma NF-e comum. O acesso à NF-e detalhada se dá através da inserção da chave de acesso no site da secretaria da fazenda do estado de emissão da NF-e.

Para o LudiiPrice, a NF-e detalhada é utilizada para obter e armazenar todas as informações necessárias de um produto na base de dados. A Figura 2 mostra um exemplo de NF-e detalhada.

Chave de Acesso
Chave de Acess

Figura 2 – Exemplo de nota fiscal eletrônica detalhada

Fonte: (SEFAZ, 2022)

2.2 Unified Modeling Language - UML

A linguagem de modelagem unificada (*UML*, na sigla em inglês) é uma linguagem padronizada para projetar *softwares*. Pode ser utilizada para visualizar, especificar, construir e documentar os artefatores de um sistema intensivo em software (BOOCH GRADY; RUMBAUGH, 1998).

A documentação de um software, proporcionada pela *UML*, também serve para que todas as partes interessadas em um projeto (*stakeholders*) garantam que o sistema satisfaça as suas necessidades. Alguns interesses em um sistema incluem: escalabilidade da aplicação, satisfação do usuário final, obediência à regras de negócio, entre outras (*UML*, 2005).

2.2.1 Diagrama de casos de uso

Na *UML*, o diagrama de caso de uso serve para modelar o comportamento de um sistema, auxiliando também no reconhecimento de suas funcionalidades (IBM, 2022). O diagrama de casos de uso também descreve o escopo e as funções de alto nível de um sistema, além de identificar as interações entre o sistema e os seus atores. Por possuir uma visão de alto nível, o seu objetivo não é descrever como o sistema opera internamente (i.e classes, métodos, declaração de atributos, etc).

Segundo IBM (2022), o diagrama de casos de uso é útil nas seguintes situações:

- Auxilia no entendimento da equipe sobre os clientes, trabalhadores e atividades de um negócio.
- Auxilia no levantamento de requisitos do sistema.

- Identificação de classes necessárias para o sistema.
- Identificação de testes a serem realizados no sistema.

2.2.1.1 Elementos do diagrama de casos de uso

Um diagrama de casos de uso é composto de diversos elementos visuais. Segundo UFCG (2022), estes elementos são:

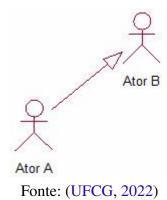
- Caso de uso: Descreve uma funcionalidade do sistema que possua valor para o usuário final.
- Ator: Representa o perfil de um usuário que interage com o sistema. O ator pode ser uma pessoa, organização, máquina ou outro sistema externo.
- Subsistema: Representa uma unidade comportamental independente dentro de um sistema.
- **Relacionamento:** Conexão entre elementos do modelo. O relacionamento adiciona a semântica a um modelo ao definir o comportamento estruturado entre elementos.
- Limites do sistema: Um retângulo que envolve todos os casos de uso pertencentes ao sistema ou subsistema.
- Nome do sistema: Especifica o nome do sistema, localizado dentro dos limites do sistema.

2.2.1.2 Tipos de relacionamentos

Um relacionamento de um diagrama de casos de uso pode ter um de vários tipos, segundo UFCG (2022) os seus tipos são:

- Associação: define a funcionalidade do sistema do ponto de vista do usuário.
- Generalização: A generalização entre atores faz com que os casos de uso de um ator também pertençam a outro. Na Figura 3, os casos de uso do ator B também pertencem ao ator A.

Figura 3 – Generalização entre atores de um diagrama de casos de uso



- *Include*: Este tipo de relacionamento de um caso de uso A para o caso de uso B indica que B é essencial para o comportamento de A.
- *Extend*: Este tipo de relacionamento de um caso de uso A para o caso de uso B indica que A pode ser acrescentado par adescrever o comportamento de B, mas não é essencial.
- Generalização / Especialização entre casos de uso: Indica que um caso de uso específico herda todas as características de um caso de uso genérico.

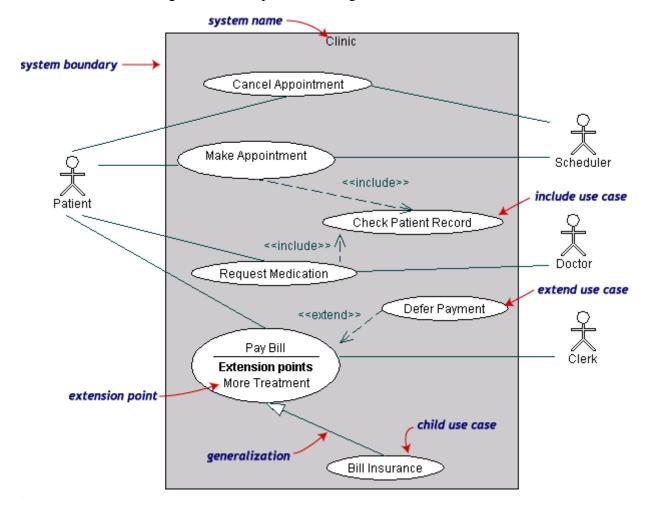


Figura 4 – Exemplo de um diagrama de casos de uso

Fonte: (UFCG, 2022)

2.2.2 Diagrama de sequência

O diagrama de sequência prioriza mostrar a ordem da troca de mensagens entre os objetos de um sistema. Os objetos que participam do cenário do diagrama de sequência são colocadas no topo, paralelos ao eixo X, o objeto que inicia a interação é colocado mais à esquerda do diagrama. Em seguida as mensagens trocadas pelos objetos do diagrama são posicionadas, juntamenete com uma representação do intervalo de tempo entre a troca de mensagens no eixo Y (BOOCH GRADY; RUMBAUGH, 1998).

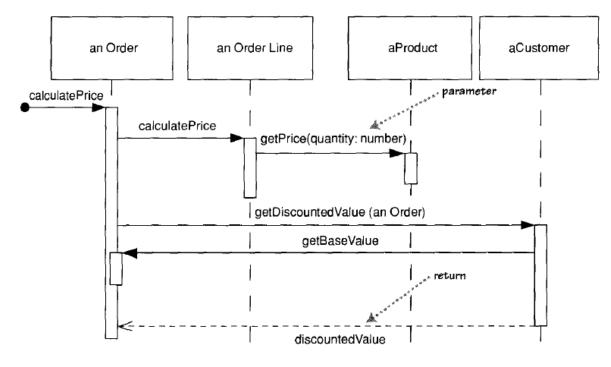


Figura 5 – Exemplo de diagrama de sequência

Fonte: (FOWLER, 2003)

O diagrama da Figura 5 mostra o fluxo de troca de mensagens entre os objetos "an Order", "an Order Line", "aProduct" e "aCustomer". As setas contínuas e paralelas ao eixo X representam a chamada de um método, tal que a seta que sai de um objeto A para um objeto B representa a chamada de um método do objeto B dentro do objeto A.

A seta pontilhada representa uma mensagem de retorno. Na Figura 5, o objeto "aCustomer" retorna "discountedValue" para o objeto "an Order".

Um diagrama de sequência permite a criação e a exclusão de objetos ao longo do fluxo representado, além da representação de laços de repetição, condicionais e cenários alternativos (FOWLER, 2003). Estas operações são análogas às operações de mesmo nome em linguagens de programação.

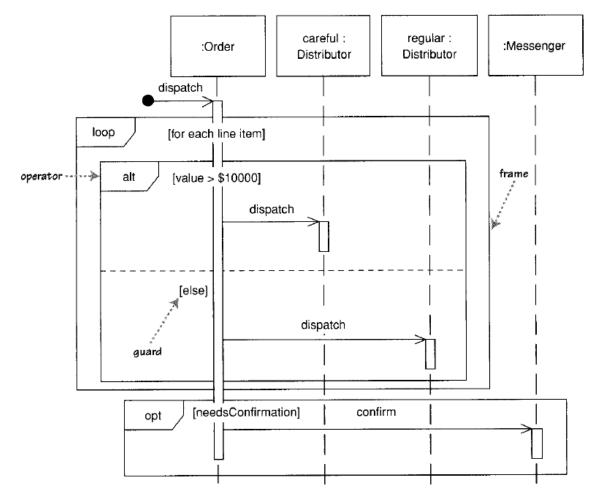


Figura 6 – Exemplo de loops e condicionais em um diagrama de sequência

Fonte: (FOWLER, 2003)

a Handler query database new a Query Command new a Database Statement creation execute deletion from other results object extract results close results self-deletion

Figura 7 – Exemplo de criação e exclusão de objetos em um diagrama de sequência

Fonte: (FOWLER, 2003)

2.3 Web Scraping

Web scraping é o processo de obtenção automatizada de dados da internet, através de métodos que não interagem com uma API. A obtenção dos dados ocorre através de requisições realizadas a um site, seguida do processo de extração dos dados a partir do arquivo obtido pela requisição, seja ele HTML, XML ou outro formato de página web (MITCHELL, 2018).

2.3.1 Crawlers

Um *crawler* normalmente é operado por mecanismos de busca (Google, Bing, Duck-DuckGo, etc), com o objetivo de indexar os *sites* encontrados no resultado de uma busca (CLOUDFARE, 2022). No contexto do *web scraping*, um *crawler* é responsável por indexar páginas *web*, enquanto o *web scraping* em si coleta as informações das páginas indexadas.

2.3.2 Implicações éticas

Web scraping pode ser usado tanto com boas intenções, como projetos para análise de dados, quanto com más intenções, como copiar o conteúdo inteiro de um site sem creditar o autor original. Alguns projetos não são mal intencionados, mas prejudicam o servidor de um serviço web com um número alto de requisições.

No ano 2000, o site eBay (ebay.com), após ter diversos de seus avisos ignorados, processou o site Bidder's Edge. O motivo foi o alto número de requisições de *crawlers* originadas deste site que estavam sobrecarregando o servidor do eBay (JUSTIA, 2000).

Existem diversos meios de garantir que o *web scraping* seja feito de forma ética. Um site pode colocar proteções contra a coleta de dados automatizada em seu domínio, explicitando que ele não permite este tipo de prática. Outra forma é ler o arquivo robots.txt, presente no diretório raíz de vários sites (MITCHELL, 2018). O arquivo robots.txt segue uma convenção que pode ser interpretada por um *crawler*, informando-o quais informações e rotas não devem ser acessadas durante o *web scraping*.

O arquivo robots.txt não impede que o *web scraping* obtenha as informações de um site, ele apenas sugere o que um *bot* não deve fazer.

Figura 8 – Exemplo de arquivo robots.txt

Host: store.steampowered.com

User-Agent: *

Disallow: /share/

Disallow: /news/externalpost/

Disallow: /account/emailoptout/?*token=

Disallow: /login/?*guestpasskey=

Disallow: /join/?*redir=

Disallow: /account/ackgift/

Disallow: /email/ Disallow: /widget/

Fonte: (VALVE CORPORATION, 2022)

O arquivo robots.txt na Figura 8 foi obitido do site https://store.steampowered.com/robots.txt. O campo "Host" informa o domínio em questão. O campo "User-Agent" informa para quais tipos de crawlers as regras se aplicam, neste caso aplica-se para todos os crawlers

(identificado pelo símbolo *). O campo "Disallow" informa qual rota não deve ser acessada pelo crawler, como "/share/", "/news/externalpost/", entre outras rotas mencionadas no arquivo. Um comentário pode ser adicionado ao arquivo robots.txt utilizando o símbolo #, da mesma forma que um comentário em python (MITCHELL, 2018).

2.4 Interface gráfica do utilizador

A Interface gráfica do utilizador (*GUI* na sigla em inglês) é um programa de computador que permite a comunicação do usuário com a máquina através do uso de símbolos, metáforas e dispositivos apontadores (como um *mouse*, por exemplo) (LEVY, 2022). Um exemplo de interface gráfica é mostrado na Figura 9.

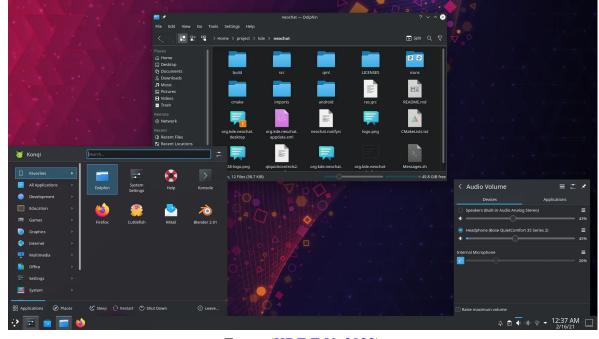


Figura 9 – Exemplo de GUI

Fonte: (KDE E.V, 2022)

A criação da interface gráfica se deu através da contribuição de diversos inventores, entre eles se destacam:

- Vannevar Bush: pela sua publicação "As we May Think" em 1945. O seu trabalho defendia a ideia de que no futuro, os usuários utilizariam dispositivos parecidos com computadores (chamados de memex em sua publicação) contendo botões e alavancas que acessariam um número vasto de informações (BUSH, 1945).
- **Douglas Engelbart**: Criador do *Augmentation Research Center*. O trabalho de Vannevar Bush o inspirou a desenvolver uma série de inovações nos anos 60, incluindo a criação

do *mouse*, utilizado até hoje em computadores de mesa, computadores móveis e outros dispositivos computacionais (INTERNET HALL OF FAME, 2022).

• Alan Kay: Pelas suas contribuições no desenvolvimento do computador *Xerox Star* nas décadas de 1970 e 1980. Este computador implementou uma funcionalidade que mais tarde ganharia o nome de *What you get is what you see* (abreviado para *WYSIWYG*, que significa "O que você consegue é o que você ganha" em português), onde o usuário poderia visualizar com antecedência o resultado de sua impressão na interface gráfica presente na tela do computador.

Os resultados produzidos pela Xerox na década de 1970 e 1980 serviram de inspiração para os sistemas operacionais com interfaces gráficas lançados por empresas como a *Apple*, com a sua interface gráfica presente no *Macintosh*, e a *Microsoft*, com o *Windows*, que por sua vez ajudaram a popularizar os computadores de mesa com intefaces gráficas na década de 1980.

A popularização das interfaces gráficas foi além dos computadores de mesa, indo também para os computadores móveis e para os dispositivos móveis, conhecidos como *smartphones* (TANENBAUM ANDREW; AUSTIN, 2012).

3

Produtos relacionados

Este capítulo apresenta os aplicativos semelhantes à interface móvel proposta neste documento. Por se tratar de uma pesquisa por produtos reais, ao invés de uma pesquisa acadêmica, foi utilizado apenas o próprio motor de busca da Google para buscar notícias que comentem sobre aplicativos populares com propostas similares ao LudiiPrice. A pesquisa foi realizada entre os meses de abril e junho durante o ano de 2021.

Os aplicativos encontrados passaram pelos critérios de seleção da seção 3.1. Após a filtragem, realizou-se o download dos aplicativos escolhidos na plataforma oficial de distribuição de aplicativos para sistemas Android, a Google Play Store, com o intuito de avaliá-los.

3.1 Critérios de seleção

Os seguintes critérios foram utilizados para avaliar os aplicativos relevantes, onde um critério que começa com a letra E refere-se a um critério de exclusão, já um critério que se inicia com a letra I refere-se a um critério de inclusão.

- **E1.** Aplicativos que não funcionam no Brasil.
- E2. Sistemas que não possuem uma versão móvel.
- E3. Aplicativos que não resgatam os preços dos produtos de bases de dados.
- II. Aplicativos cuja proposta é indicar o preço de produtos de varejo para o usuário.

3.2 Questões de análise

As seguintes questões foram escolhidas para avaliar as funcionalidades dos aplicativos escolhidos para análise:

- 1. O aplicativo armazena os produtos da nota fiscal após a leitura de seu código QR?
- 2. O aplicativo permite a comparação do preço de um produto em diferentes estabelecimentos?
- 3. O aplicativo possui alguma integração com programa(s) de incentivo fiscal do governo?
- 4. O aplicativo permite a criação de uma lista de compras?
- 5. O aplicativo apresenta produtos com os preços atualizados? Isto é, o preço dos produtos foram atualizados em um período de 1 mês?
- 6. O aplicativo exibe algum histórico de preços de um produto ou de compras do usuário?

Para responder a pergunta 5 foram selecionados 5 produtos que compõem a cesta básica de acordo com o Departamento Intersindical de Estatística e Estudos Socioeconômicos (DIEESE) (DIEESE, 2009). Os produtos escolhidos foram: leite, feijão, arroz, farinha de trigo e açúcar. Após a seleção, estes produtos foram pesquisados nos aplicativos semelhantes para analisar a data da última atualização de seus respectivos preços.

Também com relação à pergunta 5, estabeleceu-se o limite de 1 mês para a última atualização do preço de um produto. Este período de tempo foi escolhido baseado na periodicidade mensal da Pesquisa Nacional da Cesta Básica de Alimentos (PNCBA) realizada pelo DIEESE. Ao analisar as pesquisas, presentes em DIEESE (2021), é possível ver que a cada mês o valor de uma cesta básica apresenta variação em alguns estados.

3.3 Aplicativos relacionados

Esta seção tem como objetivo responder as questões de pesquisa da seção 3.2 referentes aos aplicativos encontrados. Cada enunciado terá uma resposta para cada aplicativo encontrado na pesquisa, onde cadate resposta possui a seguinte estrutura: **Nome do aplicativo:** Resposta da questão.

Após a realização da pesquisa de mercado, 4 aplicativos principais foram encontrados, são estes: Pinngo, Economiza Club, Revela Preço e Menor Preço Brasil. As figuras 10 e 11 exibem as telas dos aplicativos relacionados encontrados.

1 Q PINNGO Pesquisar Produtos Luzia ALTERAR Boa tarde Rodrigo Sugestões para você economizar Navegue por categorias O que você está procurando? Crédito de R\$ 50 Faça a integração da sua conta do Economiza Club com o menor preço Brasil e receba 50 reais para gastar como quiser. Bebidas Ве Alimentos Medicament Carnes e MEUS PRODUTOS - TOP 5 Últimos visitados ELEGÊ CAIXA 200 G R\$ 2.98 a R\$ 2.98 Pão ver mais TOP 10 SEMANAL DO ECONOMIZA Macarrão Instantân eo Lámen R\$ 8,15 R\$ 20,00 R\$ 5,99 R\$ 0,79 Leite UHT Minhas Lista:

Figura 10 – Tela do Pinngo (à esquerda) e tela do Economiza Club (à direita).

Fonte: autoria própria

menor preço Revela Preço Ops! Nenhum preço postado na sua região, vamos mostrar o que encontramos em outras regiões: **PRODUTOS** Wickbold Pao Grao Sabor Integral -R\$ 8,99 🗐 ITENS DE PREVENÇÃO dia 08/01/2021 MINI LER CÓDIGO DE BARRAS Q PESQUISAR PRODUTOS Pao Panitts 400g - UN PESQUISAR COMBUSTÍVEIS R\$ 8,29 : dia 08/01/2021 Farroupilha, RS (Comercio De Produtos Alimenticios J... Ricota Santa Clara - UNIDADE R\$ 6,29 QUATA dia 03/01/2021 Fermento Em Po Quimico Royal -\$ Ħ Escolher Bairros

Figura 11 – Tela do Revela Preço (à esquerda) e tela do Menor Preço Brasil (à direita).

Fonte: autoria própria

1. O aplicativo armazena os produtos da nota fiscal após a leitura de seu código QR?

Pinngo: Sim, existe um botão logo na tela inicial para permitir que o usuário leia o código QR de uma nota fiscal.

Economiza Club: Sua interface indica que sim, porém o aplicativo apresentou problema de "Chave de acesso inválida" ao ler uma nota fiscal de Aracaju/SE.

Revela Preço: Sim, ao ler o código QR de uma nota fiscal, uma mensagem de agradecimento é exibida.

Menor Preço Brasil: Não.

2. O aplicativo permite a comparação do preço de um produto em diferentes estabelecimentos?

Pinngo: Permite, ao clicar em um produto é possível ver mais detalhes sobre o mesmo em uma nova tela, incluindo o seu preço em diferentes estabelecimentos.

Economiza Club: Permite. Ao clicar em um produto, o aplicativo exibe detalhes de todos os estabelecimentos próximos em que ele se encontra, isto inclui o seu preço.

Revela Preço: Não, ao clicar para ver mais detalhes sobre um produto, apenas o preço de

um estabelecimento é exibido.

Menor Preço Brasil: Não, o aplicativo exibe o preço de um produto em apenas um estabelecimento.

3. O aplicativo possui alguma integração com programa(s) de incentivo fiscal do governo?

Pinngo: Não, suas informações são fornecidas apenas pela leitura dos códigos QR realizadas por usuários do aplicativo.

Economiza Club: Sim, o aplicativo permite a importação de notas fiscais do programa "Nota Fiscal Gaúcha".

Revela Preço: Não, seus dados são alimentados apenas por usuários do aplicativo.

Menor Preço Brasil: Não, porém os preços são atualizados automaticamente, pois o aplicativo pertence ao governo do estado do Rio Grande do Sul.

4. O aplicativo permite a criação de uma lista de compras?

Pinngo: Permite.

Economiza Club: Sim.

Revela Preço: Não, o menu de navegação possui a opção "Lista de compras", mas ao selecionar esta opção, redireciona para uma tela contendo o texto "seção em desenvolvimento". **Menor Preço Brasil:** Não.

5. O aplicativo apresenta produtos com os preços atualizados? Isto é, o preço dos produtos foram atualizados em um período de 1 mês?

Pinngo: Não. Ao pesquisar por produtos da cesta básica, menos da metade apresentou um preço atualizado no último mês em pelo menos 1 estabelecimento.

Economiza Club: Não, ao buscar pelos produtos escolhidos da cesta básica, todos eles apresentaram preços atualizados pela última vez em 2020.

Revela Preço: Em sua maioria, sim. Poucos produtos apresentaram preço desatualizado durante a busca.

Menor Preço Brasil: Sim, pois os seus preços estão integrados com a secretaria da fazenda do estado do Rio Grande do Sul.

6. O aplicativo exibe algum histórico de preços de um produto ou de compras do usuário?

Pinngo: Sim. O aplicativo exibe o histórico de notas lidas pelo usuário, mas não existem informações detalhadas sobre sua compra.

Economiza Club: Sim, exibe um histórico de gastos que não pôde ser testado devido a limitações das notas fiscais aceitas pelo aplicativo.

Revela Preço: Sim. Exibe o histórico de notas lidas pelo usuário. Não foi possível confirmar o nível de detalhe exibido ao selecionar uma nota lida, pois ela ficou com o processamento pendente desde a sua leitura até a escrita deste trabalho, aproximadamente 1 mês de duração.

Menor Preço Brasil: Não.

3.4 Considerações do capítulo

Nota-se que nenhum dos aplicativos analisados obteve a resposta "Sim" para todos os critérios de comparação estabelecidos. O Menor preço Brasil e Revela Preço apresentam preços atualizados para todos (ou quase todos) os produtos estabelecidos na questão 5 da seção 3.2, porém carecem de funcionalidades que podem beneficiar ainda mais os seus usuários.

Já os aplicativos que possuem uma interface com mais funcionalidades, como por exemplo um histórico de preços de um produto ou um histórico de notas lidas pelo usuário, enfrentam dificuldades para manter os preços de alguns produtos atualizados dentro do período de 1 mês, alguns produtos encontram-se até com mais de 4 meses desde a sua última atualização.

Outro desafio enfrentando para alguns aplicativos, como o Economiza Club, é apresentar produtos de diversos estados do país.

O objetivo do LudiiPrice é possuir, no mínimo, todos os critérios cobrados nas questões de pesquisa, exceto a funcionalidade de integração com órgãos públicos para obtenção de preços de produtos (devido ao escopo atual do projeto).

4

Apresentação e descrição da interface móvel

Esta seção tem como objetivo descrever o processo de levantamento de requisitos do sistema, além de apresentar as funcionalidades desenvolvidas após este processo.

4.1 Pesquisa de hábitos de compras da população

O primeiro passo para auxiliar e validar o levantamento de requisitos do LudiiPrice é a pesquisa de hábitos de compras da população. Esta pesquisa foi realizada em parceria com a Coordenação de Inovação e Transferência de Tecnologia da Universidade Federal de Sergipe (CINTTEC-UFS), que ajudou na divulgação do questionário para dezenas de pessoas. O questionário foi construído utilizando a ferramenta Formulários Google (GOOGLE, 2022b).

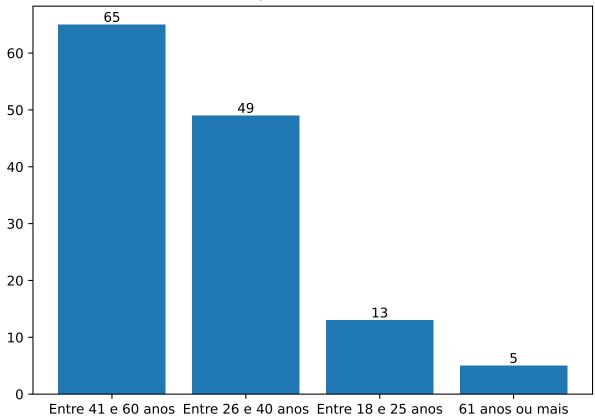
O texto completo contendo a análise completa dos questionários pode ser lido em https://sites.google.com/view/ludiiprice/questionário. As principais observações em relação às questões de pesquisa serão avaliadas nesta seção.

O questionário possui 8 questões, onde duas delas possuem a alternativa do tipo "outras", permitindo que uma resposta de texto aberto seja dada. A pesquisa foi realizada durante os dias 26 de Agosto e 7 de Novembro do ano de 2021, após a sua aplicação 132 respostas foram obtidas.

A Figura 12 mostra a faixa etária dos participantes da pesquisa, onde a sua maioria possui entre 41 e 60 anos (49,2%), seguida de participantes com idade entre 26 e 40 anos (37,1%).

Figura 12 – Faixa etária dos participantes do questionário

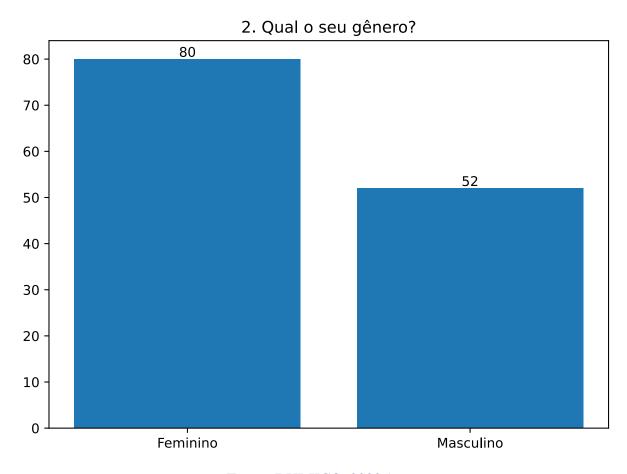
1. Qual a sua idade?



Fonte: (LUDIICO, 2022a)

A Figura 13 mostra que 60,6% dos membros entrevistados são do gênero feminino, seguido de 39,4% do gênero masculino.

Figura 13 – Gênero dos participantes do questionário

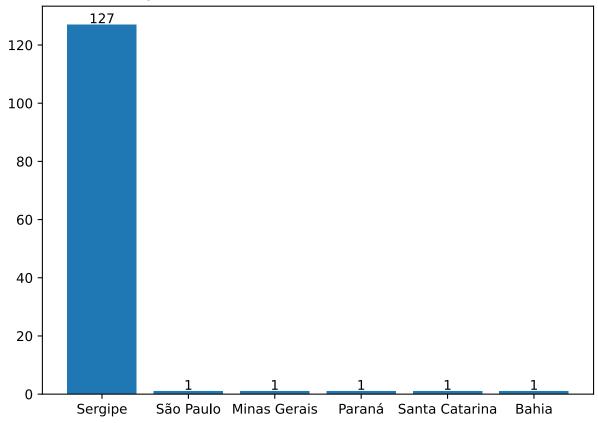


Fonte: (LUDIICO, 2022a)

A Figura 14 mostra que mesmo através de divulgações de meios online, quase todos os participantes (96,2%) são do estado de Sergipe.

Figura 14 – Estado de residência dos participantes do questionário

3. Qual estado brasileiro você reside atualmente?

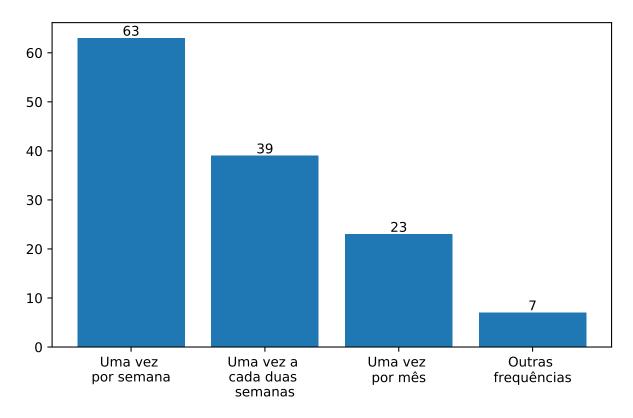


Fonte: (LUDIICO, 2022a)

A Figura 15 exibe a frequência de compras realizadas pelos participantes da entrevista. Os resultados obtidos foram: 47,8% realizam compras mensais, 29,5% realizam compras quinzenais, 17,4% realizam compras duas vezes na semana e 5,3% responderam realizar compras com outras frequências.

Figura 15 – Frequência de compras dos participantes do questionário

4. Com que frequência você ou sua família fazem compras em supermercados?

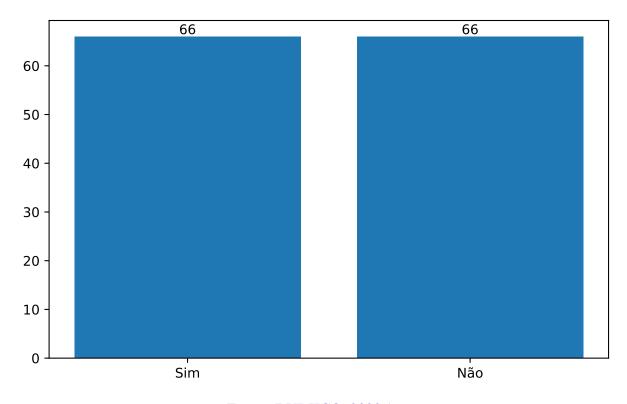


Fonte: (LUDIICO, 2022a)

A Figura 16 questiona o costume dos participantes de pesquisar os preços de produtos antes de ir a um estabelecimento. Metade dos participantes responderam que possuem o costume de pesquisar preços antes de realizar as suas compras, já a outra metade respondeu que não.

Figura 16 – Hábito de pesquisa dos participantes do questionário

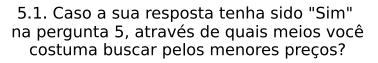
5. Antes de realizar as compras, você costuma pesquisar quais supermercados possuem o menor preço para os itens que você deseja comprar?

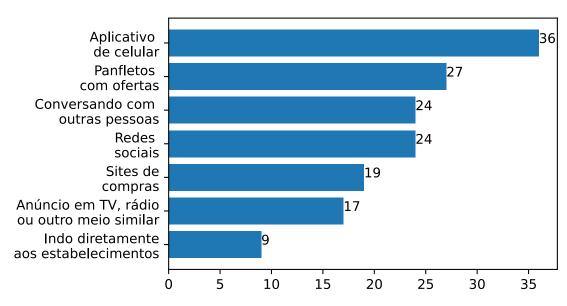


Fonte: (LUDIICO, 2022a)

A Figura 17 questiona somente os 66 candidatos que responderam "Sim" na Figura 16. O seu questionamento se trata dos meios utilizados por estes consumidores para saber o preço de produtos antes de realizar uma compra, onde mais de um meio pode ser selecionado. As respostas apontam uma popularidade maior para o meio "Aplicativo de celular", a mesma categoria do projeto deste documento, com 36 participantes escolhendo esta opção.

Figura 17 – Meios de pesquisa dos participantes do questionário

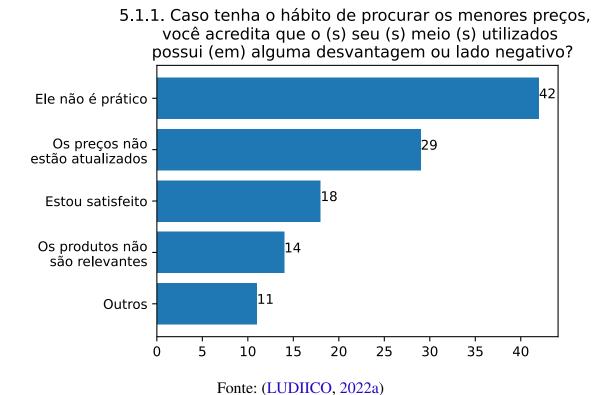




Fonte: (LUDIICO, 2022a)

A Figura 18 também questiona apenas os 66 candidatos que responderam "Sim" na Figura 16, buscando a opinião destes participantes em relação ao meio utilizado para comparação de preços de produtos, também foi possível escolher mais de uma opção nesta questão. As principais reclamações incluem a falta de praticidade do meio utilizado (com 42 participantes escolhendo esta opção), seguida da da desatualização dos preços dos produtos no meio utilizado. Todas as respostas escolhidas pelos usuários encontram-se na Figura 18.

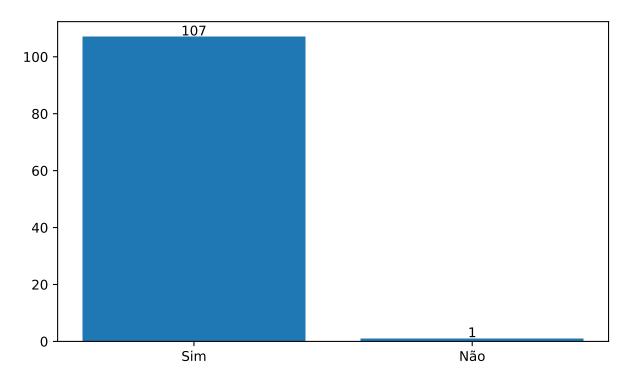
Figura 18 – Desvantagens dos meios de pesquisa de preços de produtos utilizados pelos participantes



Por fim, a Figura 19 questiona o interesse dos usuários que não utilizam um meio para comparar preços em ter uma ferramenta que os auxilie nesse processo. Embora a questão tenha sido destinada apenas para os 66 usuários que responderam "Não" na questão da Figura 16, 107 dos 132 participantes afirmaram interesse em ter uma ferramenta para comparar os preços de suas compras, com apenas 1 participante não demonstrando interesse nessa ferramenta.

Figura 19 – Interesse dos participantes do questionário em uma ferramenta que os auxiliem na busca por produtos

5.2 Caso sua resposta tenha sido "não" à pergunta 5, você teria interesse em utilizar uma ferramenta que informe os menores preços de produtos em estabelecimentos próximos a você?



Fonte: (LUDIICO, 2022a)

4.2 Tabela de histórias do usuário

Após a identificação das necessidades de um consumidor comum, realizada na seção 4.1, o próximo passo para mapear mais necessidades é a tabela de histórias do usuário. Através dela, é possível representar, do ponto de vista dos usuários finais, as funcionalidades do sistema e o valor agregado por elas. As histórias de usuário possuem uma linguagem simples e não técnica (ATLASSIAN, 2022).

Após diversas reuniões entre os desenvolvedores, o orientador e o coorientador deste projeto, as histórias de usuário foram identificadas, apresentadas nas tabela 1.

A tabela 1 possui a seguinte estrutura: A coluna "Como" representa o usuário final que deseja realizar a ação. A coluna "Quero" representa a ação que o usuário final deseja realizar. Já a coluna "Para", representa o motivo que leva o usuário final a realizar a ação.

Como	Quero	Para
Consumidor.	Pesquisar por um produto e saber o	Economizar em minhas compras.
	melhor preço.	
Consumidor.	Ver as minhas compras anteriores.	Saber o quanto gastei nas últimas com-
		pras que fiz.
Consumidor	Montar uma lista de compras.	Saber o preço dos produtos da lista
		em diferentes estabelecimentos.
Consumidor	Ver o histório de preços de um produto	Saber se o seu preço aumentou ou
		diminuiu com o passar do tempo.

Tabela 1 – Histórias do usuário

É importante destacar que, para a versão atual do sistema, apenas um usuário final foi identificado, o Consumidor. Existe a possibilidade de novos usuários finais e/ou novas histórias de usuário surgirem na tabela em projetos futuros do LudiiPrice.

4.3 Levantamento de requisitos

Após a realização da pesquisa de hábitos da população e da criação da tabela histórias de usuário, seguiu-se com o levantamento dos seguintes requisitos funcionais, não funcionais e regras de domínio.

4.3.1 Requisitos Funcionais

Os requisitos funcionais a seguir encontram-se agrupados de acordo com as funcionalidades do sistema. As funcionalidades e seus respectivos requisitos funcionais são:

- Funcionalidades de autenticação: RF1 até o RF5.
- Funcionalidades de produtos: RF6 até o RF10.
- Funcionalidades de lista de compras: RF11 até o RF17.
- Funcionalidades de notas fiscais: RF18 até o RF21
- Funcionalidades do menu de configurações: RF22 até o RF23

Vale lembrar que um requisito escrito da forma "X0Y" representa a dependência do requisito Y em relação ao requisito X.

Tabela 2 – Requisitos funcionais do sistema

Identificação	Descrição	Ator
RF1	O usuário não autenticado cadastra uma conta.	Usuário não autenticado
RF2	O usuário não autenticado entra com uma conta.	Usuário não autenticado
RF3	O usuário não autenticado redefine a senha de sua	Usuário não autenticado
	conta.	
RF4	O usuário altera a senha de sua conta.	Usuário
RF5	O usuário sai da sua conta.	Usuário
RF6	O sistema deve exibir uma lista contendo os pro-	Sistema
	dutos mais comprados próximos ao usuário.	
RF7	O usuário busca por um produto avulso.	Usuário
RF8	O usuário visualiza os detalhes de um produto.	Usuário
RF9	O sistema deve exibir o histórico de preços de um	Sistema
	produto.	
RF10	O sistema deve calcular a distância de um estabe-	Sistema
	lecimento a partir da localização do dispositivo	
	do usuário.	
RF11	O usuário consulta as suas listas de compras.	Usuário
RF12	O usuário cria uma lista de compras.	Usuário
RF13	O usuário exclui uma de suas listas de compras.	Sistema
RF14	O usuário adiciona produtos à sua lista de compras.	Sistema
RF15	O usuário remove produtos da sua lista de compras.	Sistema
RF16	O usuário visualiza detalhes da sua lista de com-	Usuário
	pras.	
RF17	O sistema deve exibir um gráfico comparativo de	Usuário
	preços de produtos pertencentes à lista de compras	
	do usuário.	
RF18	O usuário lê o código QR de uma nota fiscal.	Usuário
RF1801	O sistema deve armazenar as informações de uma	Sistema
	nota fiscal cujo código QR foi lido.	
RF19	O usuário exclui uma nota fiscal eletrônica de sua	Usuário
	conta.	
RF20	O usuário visualiza detalhes de uma nota fiscal	Usuário
	eletrônica.	
RF21	O sistema deve exibir um gráfico comparativo	Sistema
	de preços dos produtos pertencentes à nota fiscal	
	eletrônica	
	Co	ontinua na próxima página

Tabela 2 – continuação da página anterior

Identificação	Descrição	Ator
RF22	O usuário define os seus filtros de busca	Usuário
RF23	O usuário concede permissões do dispositivo ne-	Usuário
	cessárias para o funcionamento do aplicativo.	

4.3.2 Requisitos Não Funcionais

Tabela 3 – Requisitos não funcionais do sistema

Identificação	Descrição
RNF1	O sistema deve executar em dispositivos móveis com sistema operacional
	Android.
RNF2	O sistema utilizará o SGBD PostgreSQL.
RNF3	O sistema será implementado no ambiente do Heroku.
RNF4	O sistema deve implementar controle de acesso para todas as funcionalidades.
RNF5	O sistema deve aguardar um tempo máximo de 9 segundos após realizar uma
	requisição.
RNF6	A interface móvel do sistema utilizará o framework flutter.
RNF7	A interface móvel do sistema utilizará a linguagem de programação dart.
RNF8	O servidor do sistema utilizará o <i>framework</i> nodeJS.
RNF9	O servidor do sistema utilizará a linguagem de progrmação typescript.

4.3.3 Regras de Domínio

Tabela 4 – Regras de domínio do sistema

Identificação	Descrição
RD1	O código QR de uma nota fiscal só será aceito caso esta possua uma data de
	emissão de no máximo 90 dias.
RD2	Uma nota fiscal só pode ser armazenada uma única vez na base de dados da
	aplicação.
RD3	Uma nota fiscal lida será associada como uma compra do usuário que a leu.
RD4	O sistema deve armazenar notas fiscais do estado de Sergipe.
RD5	O filtro de distância "sem limites" deve ter o valor igual ao diâmetro do
	planeta Terra, ou seja, 12,742Km.
RD6	A exibição das informações de um produto sempre será baseada na geolocali-
	zação do dispositivo do usuário.

4.4 Diagrama de casos de uso

Após o levantamento de requisitos, é possível utilizar os requisitos funcionais para gerar o diagrama de casos de uso que compõem o projeto.

Consultar notas fiscais digitalizadas Excluir nota fisca Consultar detalhes do produto Consultar detalhes da nota fiscal Gerenciar notas fiscais digitalizadas Ler código QR de uma nota fiscal Pesquisar produto Alterar senha Consultar histórico de preços de um produto Efetuar login Consultar histórico de despesas Cadastrar conta Usuário não autenticado Usuário Resetar senha Conceder geolocalização Consultar listas de compras Gerenciar lista de compras Consultar detalhes da lista de compras Remover produtos da lista de compras Criar lista de compras Adicionar produtos à lista de compras Excluir lista de

Figura 20 – Diagrama de casos de uso do LudiiPrice

Fonte: (LUDIICO, 2022b)

4.5 Funcionalidades do sistema

Nesta subseção serão descritas as funcionalidades criadas no LudiiPrice. É importante destacar que todas as funcionalidades mencionadas já estão integradas com a API do LudiiPrice (ARAúJO, 2022).

4.5.1 Operações de autenticação

As operações de autenticação incluem: criar conta, realizar login, alterar senha, resetar senha em caso de esquecimento, logout e auto login.

As operações de login, criar conta e alterar senha são apenas formulários que ao serem enviados realizam as ações dos seus respectivos nomes. A operação de logout termina a sessão do usuário no aplicativo.

A operação de resetar a senha redefine a senha do usuário caso ele a tenha esquecido. Para resetar a senha, o usuário deve clicar na opção de "esqueci a senha" na tela de login e digitar o email que usa para realizar a autenticação, após isso ele receberá uma mensagem em seu email contendo um link responsável por redirecioná-lo para um formulário que solicita a sua nova senha.

A operação de auto login funciona da seguinte forma: o sistema busca por informações de uma sessão no armazenamento local do dispositivo do usuário, caso o sistema as encontre, será feita uma validação para checar se a data de expiração das informações encontradas ainda são válidas, isto é, se a data de expiração ocorre após a data de realização do auto login. Caso a data de expiração das informações encontradas seja válida, o usuário entrará na tela inicial do aplicativo, caso contrário ele será redirecionado para a tela de login.

As figuras 21 e 22 apresentam as telas relacionadas às operações de autenticação.

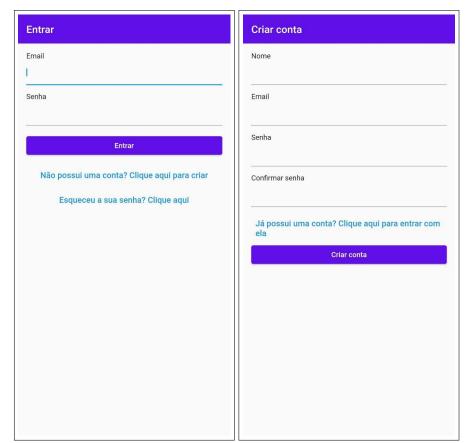


Figura 21 – Tela para login (à esquerda) e criação de conta (à direita).

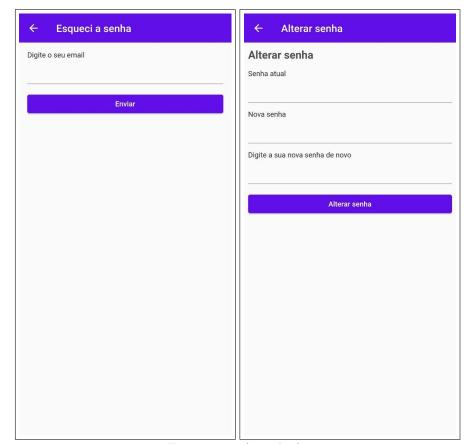


Figura 22 – Tela para resetar a senha (à esquerda) e para alterar senha (à direita).

4.5.2 Funcionalidades referentes a produtos

4.5.2.1 Exibição de produtos populares

A exibição de produtos populares utiliza a geolocalização do dispositivo do usuário para obter os produtos mais presentes na base de dados próximos a ele. Esta tela é utilizada como a tela inicial, exibida após o usuário realizar o login no aplicativo. A Figura 23 mostra a tela responsável por essa operação.



Figura 23 – Exibição de produtos populares

4.5.2.2 Buscar produtos

A busca por um produto também utiliza a geolocalização do dispositivo do usuário, o seu funcionamento é da seguinte forma: O usuário escolhe a opção "buscar produtos" no menu de navegação da aplicação, a tela de buscar um produto aparece, o usuário insere o termo da sua busca, aperta o botão de "buscar produto" e o aplicativo então exibirá os produtos com nomes similares ao termo de busca inserido pelo usuário.

Caso nenhum produto seja encontrado pela busca, uma mensagem de erro será exibida para o usuário. A Figura 24 exibe o passo a passo para realizar a busca por produtos.

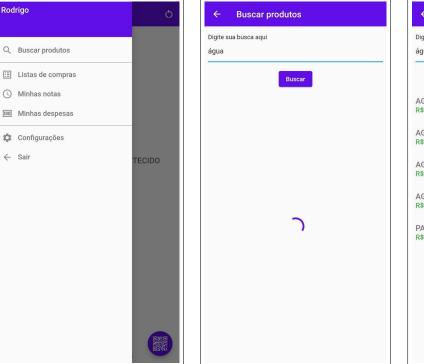


Figura 24 – Busca por produtos



Buscar produtos

Fonte: autoria própria

4.5.2.3 Exibir detalhes de um produto

A exibição dos detalhes de um produto engloba tanto os preços em diversos estabelecimentos próximos (utilizando a geolocalização do dispositivo do usuário), quanto o gráfico da variação de preços do produto.

Os preços de um produto em diversos estabelecimentos próximos são exibidos em forma de lista, já o histórico da variação do preço de um produto é exibido em um gráfico de linha simples, onde o eixo X representa a variação de tempo e o eixo Y representa a variação de preço.

O gráfico da variação do preço de um produto utiliza a média de preço do produto entre os estabelecimentos próximos ao dispositivo do usuário. A Figura 25 mostra a tela responsável por exibir os detalhes de um produto.



Figura 25 – Exibição dos detalhes de um produto

4.5.3 CRUD de lista de compras

CRUD é um acrônimo para as quatro operações básicas que lidam com armazenamento persistente de informações, são elas: *Create* (criar), *Read* (ler), *Update* (atualizar) e *Delete* (deletar) (MOZILLA, 2022).

A consulta de listas de compras exibe as listas de compras criadas pelo usuário. Através dessa tela é possível clicar no botão com o sinal de + para criar uma nova lista de compras.

A criação de uma nova lista de compras se dá da seguinte forma: o usuário deve adicionar no mínimo um produto para a sua lista clicando no botão "adicionar novo produto", este botão leva o usuário para uma tela de busca de produtos. Ao realizar a busca, o usuário deve selecionar o produto que deseja adicionar, após selecionar o produto, o sistema sairá desta tela e retornará para a tela da criação de lista de compras contendo o produto escolhido pelo usuário.

Após o usuário adicionar todos os produtos que deseja, ele confirma a criação da sua lista de compras. Vale lembrar que um nome deve ser dado para a lista em algum momento durante a sua criação. A Figura 26 mostra o fluxo da adição de um produto durante a criação de uma lista de compras.

Criar lista de compras **Buscar produtos** Criar lista de compras Nome da lista Nome da lista Digite o nome do produto que deseja adicionar à sua lista de compras Nova lista Nova lista alface Clique no botão "Adicionar +" abaixo para ALFACE HIDROPONICA adicionar um produto à sua lista de compras Buscar ALFACE FRANCESA Criar lista de compras ALFACE HIDROPONICA R\$ 2.99 - R\$ 2.99 ALFACE ROXA MO UN Adicionar produto + Adicionar produto +

Figura 26 – Criação de uma lista de compras

O usuário pode também adicionar e/ou remover produtos de uma lista já criada, basta selecionar a lista que deseja editar na tela responsável por exibir as listas de compras criadas pelo usuário.

Para remover um produto da lista de compras, o usuário deve ir para a tela que exibe os detalhes de uma lista de compras e pressionar por alguns segundos sobre o produto que deseja remover, após isso, um aviso para confirmar a exclusão do produto aparecerá na tela, o usuário então escolhe a opção "confirmar" no aviso e a exclusão do produto é concluída. A Figura 27 exibe o fluxo para excluir um produto de uma lista de compras já criada.

Visualizar lista Visualizar lista Visualizar lista grafico grafico MAC.VITARELLA SEMOLA MAC.VITARELLA SEMOLA MAC.VITARELLA SEMOLA ALFACE FRANCESA ALFACE FRANCESA ALFACE HIDROPONICA ALFACE HIDROPONICA Apagar produto Deseja apagar o produto ALFACE FRANCESA da sua lista? Cancelar Confirmar Produtos removidos da lista de compras com

Figura 27 – Remoção de um produto de uma lista de compras

O usuário pode adicionar um ou mais produtos a uma de suas listas de compras. Esta operação é realizada clicando no botão com o sinal de + na tela de visualizar detalhes da lista de compras. Após isso, uma tela de busca aparecerá, o seu funcionamento é semelhante à adição de um produto durante a criação de uma lista de compras. Após o usuário adicionar todos os produtos que deseja, ele clicará no botão de confirmação no topo da tela para adicionar os produtos. A Figura 28 exibe o fluxo da adição de um produto a uma lista de compras já criada.

Visualizar lista Visualizar lista **Buscar produtos** Digite o nome do produto que deseja adicionar à sua lista de compras grafico grafico arroz ALFACE HIDROPONICA Buscar ARROZ POP × MAC.VITARELLA SEMOLA ARROZ POP ALFACE FRANCESA ALFACE FRANCESA X BERMUDA RETA DE AMARRAR COM TECIDO QUE IMITA LINHO LISA COR CINZA MAC.VITARELLA SEMOLA ALFACE HIDROPONICA ARROZ DALON PARBORILIZADO 1KG R\$ 5,69 - R\$ 5,69 ARROZ PAR SEP 1KG UN ...• \odot GIF (3) 3 4 5 6 7 8 9 0 t u i 0 W е r У р g h j k а \Diamond n m \otimes Produtos adicionados à lista de compras com !#1

Figura 28 – Adição de um produto a uma lista de compras existente

Para excluir uma lista de compras, o usuário clica no botão de X vermelho que fica ao lado de uma lista de compras na tela de consultar lista de compras, uma mensagem de confirmação de exclusão aparecerá na tela, o usuário confirma a exclusão e a lista é apagada. A Figura 29 exibe o fluxo da exclusão de uma lista de compras criada pelo usuário.

Minhas listas de compras

Nova lista

Limpeza

Data de criação: 25/01/2022

Data de criação: 17/01/2022

×

Minhas listas de compras

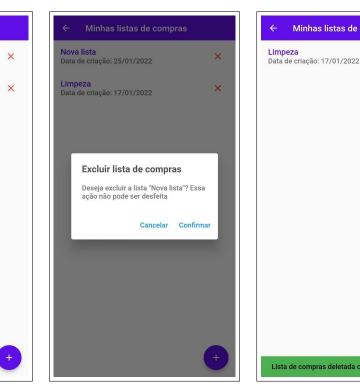


Figura 29 – Exclusão de uma lista de compras

Fonte: autoria própria

4.5.4 Gerenciamento de notas fiscais

O usuário pode gerenciar as notas fiscais cujo os códigos QR foram lidos por ele. As funcionalidades englobadas pelo gerenciamento de notas fiscais são: consulta de notas fiscais, visualização de detalhes da nota fiscal e exclusão da nota fiscal.

A consulta de notas fiscais apenas resgata todas as notas fiscais lidas pelo usuário, ordenadas em ordem decrescente por data de emissão da nota. A Figura 30 mostra a tela de consulta de notas fiscais lidas pelo usuário.



Figura 30 – Consulta de notas fiscais

A visualização de detalhes da nota fiscal exibe os produtos pertencentes à nota fiscal, além da opção de visualizar também o gráfico de preços de nota fiscal, explicado com detalhes na seção 4.5.7.2. A Figura 31 apresenta a tela responsável por exibir os produtos de uma nota fiscal.

Cetalhes Grafico

ALCOOL LIQ MELHOR BRILHO 70% I

AGUA MINERAL IMPERIAL 5L

COUVE

ALHO FORTTUNI PCT 100G

PEPINO KG

MACA VERMELHA NACIONAL KG

AROMA FLEISCHMANN ARTIFICIAL B

OVOS KEROVOS BCO G C 20

Figura 31 – Produtos de uma nota fiscal

A exclusão de uma nota fiscal não exclui uma nota e os seus produtos da base de dados da aplicação, porém do ponto de vista do usuário, ela não aparecerá mais na sua tela de consulta de notas fiscais. O termo utilizado na interface é "excluir", porém a operação real é apenas uma desassociação daquela nota fiscal ao usuário no banco de dados. Isto ocorre para preservar as informações da nota fiscal e de seus produtos no banco de dados. A Figura 32 apresenta o fluxo da desassociação de uma nota fiscal.

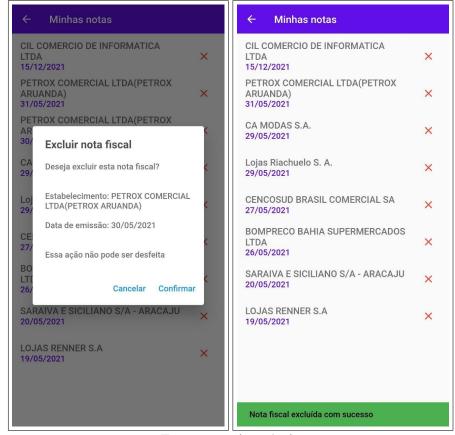


Figura 32 – Exclusão de uma nota fiscal

4.5.5 Leitura do código QR de uma nota fiscal

Atualmente, a leitura do código QR de uma nota fiscal funciona apenas para notas fiscais emitidas no estado de Sergipe, isto ocorre por causa das variações nas implementações das NFC-es de cada estado. Por conta disso, escolheu-se tratar apenas notas fiscais do estado de Sergipe para reduzir o escopo do MVP, com o objetivo de garantir um número menor de funcionalidades que possuem implementações tolerantes a um número maior de falhas.

A leitura do código QR de uma nota fiscal consiste em obter informações da NFC-e detalhada através de requisições, realizadas diretamente do dispositivo móvel do usuário, enviadas ao site do órgão responsável pela NFC-e em questão. A decisão de realizar as requisições diretamente do dispositivo do usuário se deu pelo fato de que muitas requisições originadas de uma mesma máquina, neste caso o servidor, poderiam causar um bloqueio de acesso ao site responsável pela NFC-e de um determinado estado.

O processo de leitura do código QR é realizado da seguinte forma: Ao apontar a câmera para um código QR, o aplicativo o valida para checar se esse se refere a uma nota fiscal. Caso o código se refira a uma nota fiscal válida, o aplicativo então utilizará o processo de *Web Scraping* para obter o código de acesso da nota fiscal através de sua API, que neste contexto, é reponsável

por extrair as informações relavantes da nota e armazená-las na base de dados.

Com o código de acesso é possível acessar a NFC-e detalhada, que contém as informações relevantes para a API (ARAúJO, 2022), onde esta será responsável por armazenar todas as informações relevantes presentes na nota fiscal detalhada.

Após uma leitura de código QR bem sucedida, o usuário receberá uma mensagem de sucesso confirmando a operação. Caso contrário, o aplicativo exibirá uma mensagem de erro de acordo com o erro obtido durante a operação, os possíveis erros são: código QR inválido, erro ao resgatar informações da NFC-e e erro de comunicação com o servidor da API (ARAúJO, 2022).

A Figura 33 mostra o fluxo da leitura do código QR de uma nota fiscal.

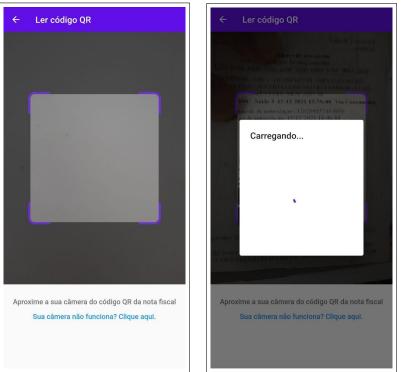
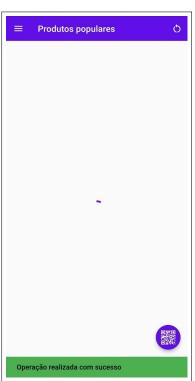


Figura 33 – Leitura do código QR de uma nota fiscal



Fonte: autoria própria

4.5.6 Histórico de despesas

O aplicativo considera toda a nota fiscal associada a um usuário como uma compra realizada por ele. Através das notas fiscais associadas ao usuário, a API (ARAúJO, 2022) calcula o seu histórico de despesas. O histórico de despesas é representado por um gráfico de linha simples.

Na versão atual, o aplicativo exibe apenas o histórico de despesas mensais, onde cada ponto no gráfico indica as despesas realizadas em um determinado mês. A Figura 34 mostra a tela responsável por exibir o histórico de despesas do usuário.

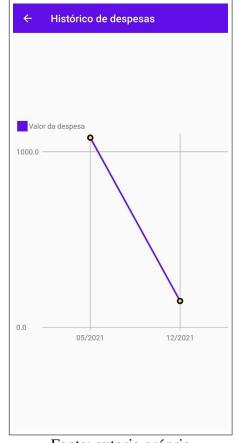


Figura 34 – Histórico de despesas do usuário

4.5.7 Gráficos de comparação de preço

O aplicativo possui dois gráficos que não se referem a um histórico. São eles o gráfico de preços de uma lista de compras e o gráfico de preços de uma nota fiscal.

4.5.7.1 Gráfico de preços de uma lista de compras

Após criar uma lista de compras, o usuário pode visualizar os detalhes sobre a sua lista criada. Os detalhes incluem os produtos pertencentes à lista de compras e o gráfico de comparação de preços da lista de compras.

O gráfico de comparação de preços exibe o preço de todos os produtos pertencentes a uma lista de compras em diferentes estabelecimentos próximos ao usuário. Os estabelecimentos próximos são obtidos através da geolocalização do dispositivo do usuário.

O tipo de gráfico utilizado é o de barras empilhadas, onde a barra de cor roxa indica o menor preço da lista de compras, enquanto a barra cinza acima da barra roxa indica a quantidade que será paga a mais em relação ao menor preço da lista de compras.

Logo abaixo do gráfico de lista de compras, existe um aviso informando que nem todo o

estabelecimento próximo possui todos os produtos da lista de compras do usuário. Este aviso existe para esclarecer ao usuário que, embora um estabelecimento possua um preço baixo em relação aos outros, ele pode não conter todos os produtos da sua lista de compras, por isso, uma lista logo abaixo deste aviso exibe os produtos da lista de compras pertencentes a cada estabelecimento exibido no gráfico.

A Figura 35 apresenta a tela responsável por exibir o gráfico de preços de uma lista de compras.



Figura 35 – Gráfico de preços de uma lista de compras

Fonte: autoria própria

4.5.7.2 Gráfico de preços de uma nota fiscal

O gráfico de preços de uma nota fiscal é acessado através da visualização dos detalhes de uma nota fiscal, onde o usuário pode obter tanto os produtos pertencentes àquela nota fiscal, quanto o gráfico de preços da nota fiscal.

O gráfico de preços da nota fiscal funciona da seguinte forma: a primeira coluna informa o preço dos produtos do estabelecimento da nota fiscal na sua data de emissão. As colunas subsequentes informam o preço atual dos produtos pertencentes à nota fiscal em diferentes estabelecimentos.

O gráfico possui um texto logo abaixo explicando como interpretá-lo, para não confundir o usuário. A Figura 36 apresenta a tela responsável por exibir o gráfico de preços de uma nota fiscal.

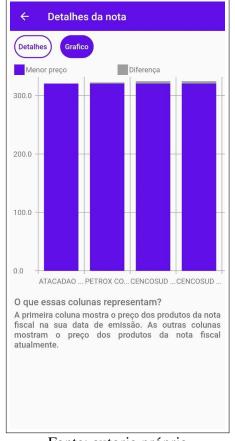


Figura 36 – Gráfico de preços de uma nota fiscal

Fonte: autoria própria

4.6 Diagramas de sequência

Devido a complexidade das funcionalidades de criar uma lista de compras e ler o código QR de uma nota fiscal, apenas elas serão representadas por diagramas de sequência. É importante destacar que o diagrama da adição de produto à tela de criação de uma lista de compras (Figura 38) é uma subsequência do diagrama de criação de lista de compras (Figura 39).

Os tratamentos de erros de conexão não serão demonstrados nos diagramas de sequência desta seção, para evitar a representação de operações triviais e reduzir a complexidade do próprio diagrama em si, facilitando o entendimento do processo para futuros desenvolvedores do projeto. Isto faz com que os diagramas representem apenas cenários sem erros ou com um número reduzido de tratamento de erros.

Vale lembrar que os diagramas dessa seção possuem a palavra *Unregistered* (do inglês, não registrado) no fundo, isto ocorre por causa da utilização da versão de avaliação do *software*

responsável pela modelagem dos diagramas.

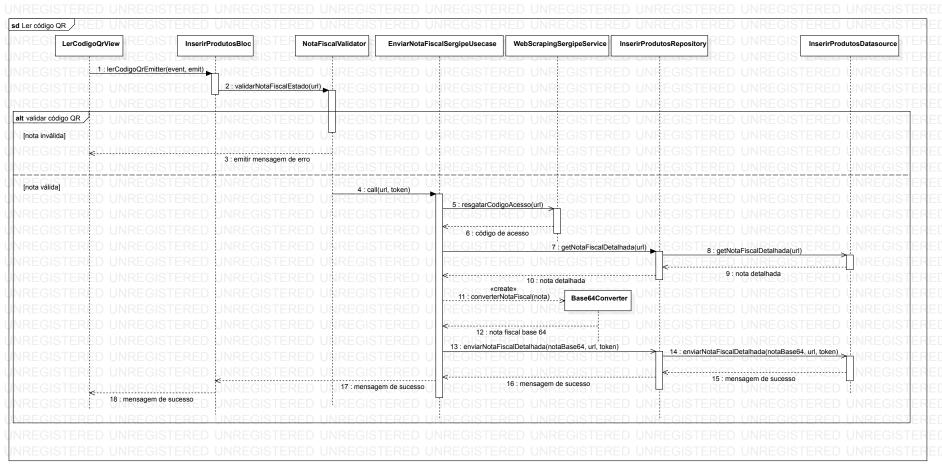


Figura 37 – Diagrama de sequência da leitura de um código QR

Addicionar produto à lista

CriarListaDeComprasView

BuscarProdutoNovaListaView

ProdutosBloc

BuscarProdutosUsecase

ProdutosRepository

ProdutosRepository

ProdutosRepository

ProdutosRepository

ProdutosRepository

ProdutosRepository

ProdutosRepository

ProdutosRemoteDatasource

A: buscarProdutos(termoBusca, token, filtros)

5: buscarProdutos(termoBusca, token, filtros)

7: produtos encontrados

8: produtos encontrados

10: adicionar produto

ProdutosRepository

ProdutosReposit

Figura 38 – Diagrama de sequência da adição de um produto à tela de criação de lista de compras

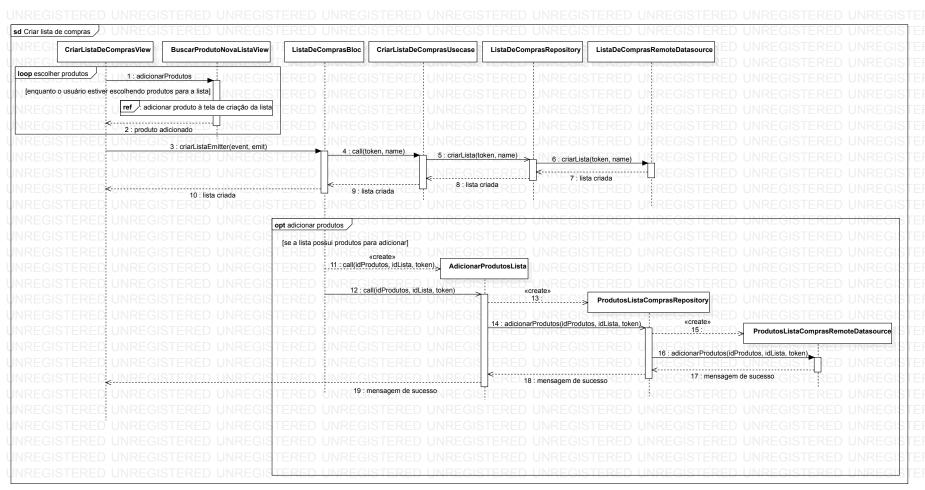


Figura 39 – Diagrama de sequência da criação de uma lista de compras

5

Arquitetura da interface móvel

Este capítulo tem como objetivo apresentar a arquitetura utilizada na interface móvel do LudiiPrice.

5.1 Tecnologia utilizada

5.1.1 Flutter

Segundo Google (2022a), *flutter* "é um *SDK* (tradução: kit de desenvolvimento de software) focado em criar interfaces para as plataformas *web*, *mobile* e desktop a partir de um único código fonte". A sua linguagem de programação é o dart.

Embora a interface *mobile* vise apenas dispositivos Android, o *flutter* permite que o aplicativo execute também em dispositivos iOS, facilitando a implementação do projeto nestes dispositivos no futuro.

Uma característica comum em diversas tecnologias voltadas para o desenvolvimento de interfaces, como *react*, vue.js, angular e *flutter*, é a componentização de elementos visuais, permitindo a reutilização de um componente em diversas telas.

Em *flutter*, os componentes visuais são chamados de *widgets*. Um *widget* pode apresentar um estado imutável ou não, ou seja, o seu visual pode ser atualizado na interface do usuário (*UI* em inglês) ou não. Quando um *widget* apresenta um estado imutável, este é chamado de stateless *widget*, já os *widgets* que apresentam estados mutáveis podem ser stateful *widget* ou inherited *widget*.

Um stateful *widget* possui a capacidade de alterar o seu estado a qualquer momento. Já um inherited *widget* altera apenas o estado de *widgets* que herdam os seus atributos.

5.2 Estilo arquitetural

Segundo Pressman (2009) um estilo arquitetural "é uma transformação imposta sobre o projeto de um sistema completo. O objetivo é estabelecer uma estrutura para todos os componentes do sistema".

O estilo arquitetural escolhido para a interface *mobile* é a arquitetura em camadas, onde o seu objetivo é separar as responsabilidades do sistema em níveis diferentes, facilitando a manutenção do código. Segundo Pressman (2009), os níveis mais internos referem-se a operações mais próximas do sistema operacional, já os níveis mais externos referem-se a operações próximas à interface do usuário, já os níveis intermediários fornecem serviços utilitários e funções do software de aplicação.

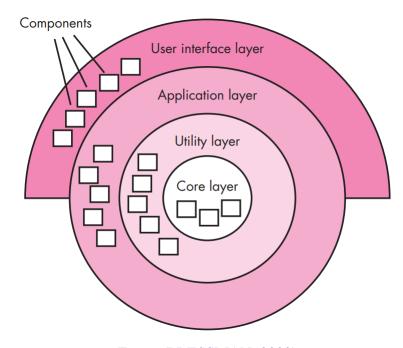


Figura 40 – Representação da arquitetura em camadas

Fonte: (PRESSMAN, 2009)

No contexto deste aplicativo *mobile*, as camadas mais externas referem-se à interface do usuário, porém as camadas mais internas são responsáveis pelo acesso e tratamento de dados obitdos através da API (ARAúJO, 2022), as camadas intermediárias se referem às lógicas de domínio e de programação do sistema. A Figura 41 mostra a organização das camadas do estilo arquitetural implementado na interface móvel.

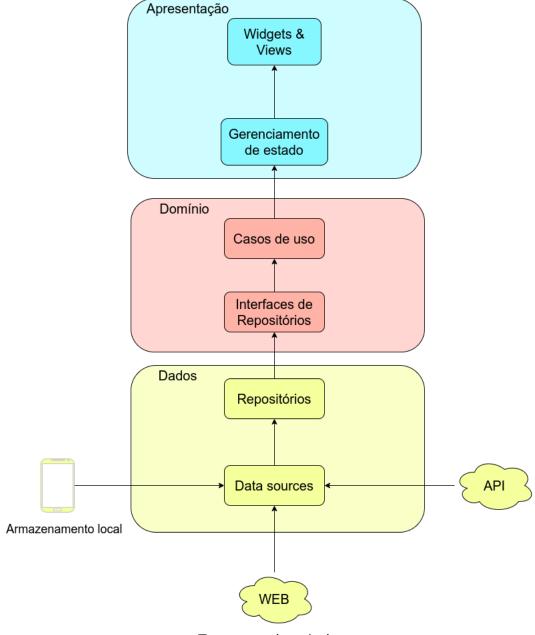


Figura 41 – Fluxo de dependências das camadas da interface móvel

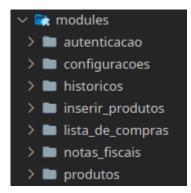
A divisão final de camadas ficou da seguinte forma: apresentação, domínio e dados, onde cada uma destas camadas possui subcamadas para tratarem responsabilidades diferentes dentro do seu contexto.

A implementação do estilo arquitetural em camadas se deu através da separação em módulos.

Um módulo é o agrupamento de funcionalidades pertencentes a um contexto da aplicação. A separação das funcionalidades para a criação de módulos se deu de forma similar à separação de funcionalidades dos requisitos funcionais na seção 4.3.1. Dentro de um módulo, o estilo

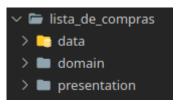
arquitetural em camadas é implementado.

Figura 42 – Módulos da interface *mobile*



Fonte: autoria própria

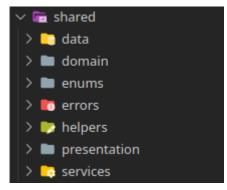
Figura 43 – Exemplo da arquitetura em camadas implementada em um módulo



Fonte: autoria própria

Arquivos compartilhados por todos os módulos, como componentes visuais e lógicas para tratamentos de erros comuns. Encontram-se em uma pasta dedicada para estes tipos de arquivos.

Figura 44 – Pasta de arquivos compartilhados da interface *mobile*



Fonte: autoria própria

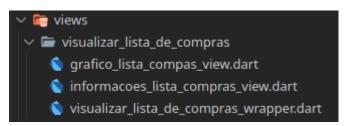
A decisão da separação em módulos se deu para explicitar os diferentes contextos e funcionalidades da aplicação, facilitando a manutenção do aplicativo, a inserção de novas funcionalidades e o entendimento do projeto para futuros desenvolvedores da interface *mobile*.

5.2.1 A camada de apresentação

A camada de apresentação contém todos os elementos visuais do aplicativo, junto com o componente responsável pela gerência de estado das telas. As suas principais subcamadas são: widgets, views e BLoC.

A camada de views contém as telas da aplicação. Telas com uma maior complexidade encontram-se agrupadas em pastas do sistema, um exemplo de tela complexa é a visualização dos detalhes de uma lista de compras, onde é possível ver tanto os produtos pertencentes à lista quanto o gráfico de uma lista de compras dependendo do item selecionado na navegação da tela.

Figura 45 – Exemplo da estrutura de pastas de uma tela com maior complexidade



Fonte: autoria própria

A camada de *widgets* contém os componentes da linguagem flutter, nesta camada eles representam os elementos visuais customizados para uma view. Os *widgets* servem tanto para o reaproveitamento de um componente visual quanto para separar a declaração de componentes visuais em arquivos distintos, evitando a sobrecarga da declaração de *widgets* em um único arquivo de uma view.

A camada de gerência de estado utiliza o padrão de projeto *Business Logic Component* (BLoC). O seu objetivo é separar a camada de apresentação da camada de lógica de negócios lógica de negócios (BLOC COMMUNITY, 2022).

O BLoC é utilizado para facilitar a exibição de *feedbacks* visuais na aplicação, uma vez que com este padrão é possível mapear o estado atual de uma tela, fazendo-a exibir um widget de acordo com o estado presente.

Por causa da separação das responsabilidades entre a apresentação e a lógica de negócios, o BLoC torna mais fácil a manutenção e a testagem da aplicação.

presentation / 🗁 bloc lista_de_compras_bloc.dart lista_de_compras_event.dart lista_de_compras_state.dart iii views visualizar_lista_de_compras grafico_lista_compas_view.dart nformacoes_lista_compras_view.dart visualizar_lista_de_compras_wrapper.dart 🔷 buscar_produto_nova_lista_view.dart onsultar_listas_de_compras_view.dart oriar_lista_de_compras_view.dart widgets obotao_adicionar_produto_lista.dart 🔷 consultar_listas_widget.dart 🐚 top_menu_bar_lista_compras_widget.dart

Figura 46 – Exemplo da estrutura de pastas da camada de apresentação

5.2.2 A camada de domínio

A camada de domínio é responsável por conter toda a lógica do domínio e de programação do projeto, suas principais subcamadas são os casos de uso (*use cases*), entidades (*entities*) e interfaces de repositórios.

Os casos de uso são análogos aos casos de uso existentes na seção 4.4. Cada caso de uso da aplicação se torna um arquivo na subcamada de casos de uso. Este arquivo por sua vez tem a responsabilidade de executar a lógica do caso de uso em questão.

As entidades são os únicos objetos independentes de qualquer outra camada do sistema. O seu objetivo é servir como um formato de dados estruturado para todas as outras camadas que as utilizam.

As interfaces de repositório representam os métodos esperados de um repositório da camada de dados. Estas interfaces são instanciadas nos casos de uso e implementadas pelos repositórios da camada de dados.

domain entities grafico_lista_compras ntity.dart | specific nroduto_grafico_lista_compras_entity.dart lista_de_compras_consulta_entity.dart 🐚 lista_de_compras_detalhe_entity.dart repositories 🖠 lista_de_compras_repository_interface.dart 🔷 produtos_lista_compras_repository_interface.dart ✓ **i** usecases 🔷 adicionar_produtos_lista_usecase.dart 🔷 consultar_listas_usecase.dart 🔷 criar_lista_compras_usecase.dart 🔷 detalhes_lista_compras_usecase.dart nterior excluir_lista_compras_usecase.dart nterior excluir_produtos_lista_compras_usecase.dart

Figura 47 – Exemplo da estrutura de pastas da camada de domínio

grafico_lista_compras_usecase.dart
lista_a_partir_de_nota_usecase.dart

5.2.3 A camada de dados

A camada de dados é responsável por todas as operações referentes aos dados de fontes externas. As principais subcamadas são: *data sources*, repositórios (*repositories*) e modelos (*models*).

O data source tem como objetivo resgatar os dados, sejam eles de fontes locais (como o armazenamento local do dispositivo) ou remotas (uma *API*, uma página *HTML* da internet, etc), caso os dados não sejam resgatados com sucesso, o data source retornará uma exceção.

Figura 48 – Exemplo do método de um *data source*

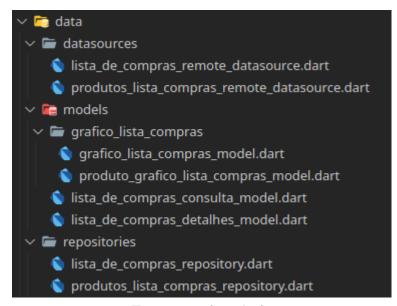
Fonte: autoria própria

O repositório tem a responsabilidade de tratar os erros emitidos pelo data source, retornando um tipo de dados condizente com o erro ou um resultado condizente com o sucesso da requisição.

A classe de modelo é uma extensão da classe de entidade, o seu papel é realizar o processo de serialização e desserialização de dados. O processo de desserialização converte dados externos em dados estruturados para a aplicação e o processo de serialização converte dados estruturados em algum formato externo utilizado por uma API, por um armazenamento interno ou por outra fonte que recebe os dados (ISOCPP, 2022).

Uma nova classe de modelo pode ser criada para tratar com diferentes formatos de arquivos externos, sejam eles JSON, HTML, XML, entre outros.

Figura 49 – Exemplo da estrutura de pastas da camada de dados



Fonte: autoria própria

6

Testes da aplicação

Este capítulo tem o objetivo de descrever as metodologias de testes aplicadas ao desenvolvimento da interface *mobile* para o LudiiPrice.

6.1 Testes unitários

Um teste unitário, segundo Meyers (2011), tem o objetivo de testar módulos, subrotinas, classes ou procedimentos em um programa, com o intuito de testar as suas funcionalidades separadamente.

Em *flutter*, os testes unitários são realizados dentro da pasta "test" no diretório raíz da aplicação, onde cada arquivo responsável por realizar um teste deve terminar com o sufixo "_test.dart" (FLUTTER DEV, 2022). Para o desenvolvimento do projeto deste documento, utilizouse a mesma estrutura de pastas separadas em módulos, descrita na seção 5.2, com o intuito de facilitar a navegação pela estrutura de pastas dos testes unitários.

Grande parte das classes do código da interface móvel que implementam alguma lógica de programação (neste caso, um método) tiveram testes unitários codificados. A figura 50 mostra o exemplo de um teste unitário criado para a operação de login.

Figura 50 – Exemplo de teste unitário para um método da operação de login

```
Run | Debug
test(
  'deve retornar um UsuarioEntity após o login ser realizado',
  () async {
   when(repository.login(any, any)).thenAnswer((_) async => Right(usuario));
   when(repository.saveLocalUserData(any)).thenAnswer(
     (_) async => const Right(true),
   when(configuracoesRepository.getConfiguracoes(any)).thenAnswer(
     (_) async => usuario,
   final resultado = await usecase.call(email, senha);
    // assert
   expect(resultado, equals(Right(usuario)));
   verify(repository.login(email, senha));
   verify(repository.saveLocalUserData(usuario));
   verify(configuracoesRepository.getConfiguracoes(usuario));
   verifyNoMoreInteractions(repository);
```

Fonte: autoria própria

Os comentários "arrange", "act" e "assert" presentes no código da figura 50 representam a convenção AAA da escrita de testes unitários (MICROSOFT, 2022a), onde o objetivo de cada código abaixo dos respectivos comentários é:

- Arrange: Inicializar os valores e objetos utilizados pelo método testado.
- Act: Executar o método testado.
- *Assert*: Verificar a corretude do método testado de acordo com o cenário de sucesso ou falha esperado.

O método "test" recebe dois parâmetros, a descrição do teste e a função a ser executada nesse teste. A descrição dos testes unitários da interface móvel seguem o formato "Deve retornar X ao realizar Y", isto é, o que o método testado deve retornar de acordo com o tipo de entrada fornecida ou de acordo com retornos de métodos externos utilizados.

Dentro de um arquivo de teste, é possível agrupar os testes escritos com o método "group", o método *group* é responsável por agrupar múltiplos testes presentes em um arquivo. No código da interface *mobile*, um agrupamento de testes é feito de acordo com o método testado, uma

vez que vários cenários de sucesso e erro são testados para o mesmo método. O método *group* também aceita uma descrição e uma função como parâmetros. No código da interface móvel, a descrição é o nome do método testado (ex: "getNotasFiscais") pelos testes unitários contidos na função passada como parâmetro para o método *group*.

Dentro dos testes unitários também foram utilizados *stubs*. De acordo com Microsoft (2022b), um stub é "uma substituição controlável para uma dependência existente no sistema", ou seja, um *stub* é utilizado para simular o funcionamento de classes externas cujo método testado depende. O método "*when*" presente na figura 50 indica o comportamento de um *stub* após este ser chamado pelo método testado.

Para executar os testes unitários desenvolvidos, deve-se abrir a linha de comando no diretório raíz do código da aplicação e executar o comando "flutter test". Após a execução do comando, a linha de comando informará os testes unitários bem sucedidos e mal sucedidos. Também é possível executar os testes diretamente em alguns ambientes de desenolvimento integrados (IDEs da sigla em inglês), porém cada *IDE* terá as suas particularidades em relação à execução dos testes unitários.

6.2 Testes de usabilidade

Um teste de usabilidade tem o objetivo de testar múltiplas funcionalidades integradas entre si dentro de um contexto do sistema. Este tipo de teste envolve a interação do usuário mais próximo do usuário final possível (MEYERS, 2011). No LudiiPrice, os testes de usabilidade foram realizados com os membros do grupo de pesquisa Ludiico, incluindo o orientador e o coorientador do projeto deste documento.

O processo dos testes foi realizado da seguinte forma: após a criação de uma nova funcionalidade na interface móvel, juntamente com a sua integração com a API (ARAúJO, 2022), um arquivo executável era gerado para que os membros do Ludiico instalassem o aplicativo em seus dispositivos móveis e testassem a usabilidade da nova funcionalidade implementada. Após a testagem da nova funcionalidade, os usuários então descreviam as suas experiências ao utilizar a nova funcionalidade em diferentes situações, com o intuito de testar cenários diferentes na mesma funcionalidade e assim testar também o seu tratamento de erros. Caso um erro fosse identificado, este era corrigido e um novo arquivo executável era gerado.

7

Considerações finais

A possibilidade de verificar o preço de um produto de supermercado antes de sair de casa ajudaria a economia de milhares, talvez milhões, de usuários ao redor do Brasil. Uma ferramenta integrada ao dispositivo móvel de um cidadão comum possuiria tanto a conveniência como as informações de valor para ajudar em suas economias.

Para ajudar a resolver este problema, trabalhou-se na construção de um aplicativo capaz de exibir estas informações para o usuário. O seu valor está não só nas informações exibidas, mas também na forma como elas são mostradas. Através de listas e gráficos, o usuário tem acesso ao preço de um ou mais produtos em diversos estabelecimentos próximos à sua localização.

O desenvolvimento do aplicativo teve como foco lidar com a dificuldade do consumidor em saber o preço de um produto antes de chegar no estabelecimento comercial desejado. Utilizouse de artefatos da engenharia de software (requisitos, tabela de histórias de usuário, diagrama de casos de uso, etc) para mapear as suas funcionalidades. O *framework flutter* foi utilizado para o desenvolvimento do aplicativo, que por sua vez é construído a partir da linguagem de programação dart.

Por fim, realizou-se a integração do aplicativo à API do LudiiPrice (ARAúJO, 2022) responsável por fornecer as informções necessárias dos produtos para exibir no aplicativo.

7.1 Trabalhos futuros

Os trabalhos futuros para a expansão da aplicação incluem:

- Testes e publicação para dispositivos iOS, visando atingir público usuário de iPhone e outros dispositivos móveis da *Apple*.
- Criar novas classes no código responsáveis por tratar notas fiscais de outros estados do Brasil.

- Utilizar de conceitos como um sistema de pontuação para incentivar a leitura de códigos QR de notas fiscais, alimentando a base de dados do sistema.
- Adicionar funcionalidades que tratem também o preço de combustíveis em diferentes postos de gasolina.
- Adicionar critérios de ordenação e filtros nas consultas de listas de compras, notas fiscais e produtos.
- Codificar os testes unitários pendentes na aplicação.

Referências

ARAúJO, A. C. Ludiiprice: API para comparação de preços através de notas fiscais de produtos emitidas por estabelecimentos comerciais. Avenida Marechal Rondon Jardim s/n - Rosa Elze, São Cristóvão - SE, 49100-000, 2022. 100 p. Citado 6 vezes nas páginas 13, 44, 57, 66, 76 e 77.

ATLASSIAN. *Histórias de usuários com exemplos e um template*. 2022. Disponível em: https://www.atlassian.com/br/agile/project-management/user-stories. Citado na página 40.

BLOC COMMUNITY. *Why Bloc?* 2022. Disponível em: https://bloclibrary.dev/#/whybloc. Citado na página 69.

BOOCH GRADY; RUMBAUGH, J. J. I. *The Unified Modeling Language User Guide* (*Addison-Wesley Object Technology Series*). [S.l.]: Addison-Wesley Professional, 1998. ISBN 0201571684. Citado 2 vezes nas páginas 17 e 19.

BUSH, V. *As we may think*. 1945. Disponível em: https://www.w3.org/History/1945/vbush/vbush.shtml. Citado na página 24.

CLOUDFARE. *O que é um bot rastreador da internet?* 2022. Disponível em: https://www.cloudflare.com/pt-br/learning/bots/what-is-a-web-crawler/. Citado na página 23.

DIEESE. *Metodologia da Cesta Básica de Alimentos*. 2009. Disponível em: https://www.dieese.org.br/metodologia/metodologiaCestaBasica.pdf>. Citado na página 27.

DIEESE. *Pesquisa nacional da Cesta Básica de Alimentos*. 2021. Disponível em: https://www.dieese.org.br/analisecestabasica/analiseCestaBasicaAnteriores.html>. Citado na página 27.

FLUTTER DEV. *An introduction to unit testing*. 2022. Disponível em: https://docs.flutter.dev/cookbook/testing/unit/introduction. Citado na página 74.

FOWLER, M. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. [S.l.]: Addison-Wesley Professional, 2003. ISBN 0321193687. Citado 3 vezes nas páginas 20, 21 e 22.

GOOGLE. *Flutter API reference documentation*. 2022. Disponível em: https://api.flutter.dev/. Citado na página 65.

GOOGLE. *Formulários Google*. 2022. Disponível em: https://www.google.com/intl/pt-BR/forms/about/>. Citado na página 32.

IBGE. *Uso de internet, televisão e celular no Brasil*. 2019. Disponível em: https://educa.ibge.gov.br/jovens/materias-especiais/20787-uso-de-internet-televisao-e-celular-no-brasil.html)>. Citado na página 12.

IBM. *Use-case diagrams*. 2022. Disponível em: https://www.ibm.com/docs/en/rational-soft-arch/9.6.1?topic=diagrams-use-case. Citado na página 17.

INTERNET HALL OF FAME. *Douglas Engelbart*. 2022. Disponível em: https://www.internethalloffame.org/inductees/douglas-engelbart. Citado na página 25.

Referências 80

ISOCPP. *Serialization and Unserialization*. 2022. Disponível em: https://isocpp.org/wiki/faq/serialization. Citado na página 72.

JUSTIA. *eBay, Inc. v. Bidder's Edge, Inc., 100 F. Supp. 2d 1058 (N.D. Cal. 2000)*. 2000. Disponível em: https://law.justia.com/cases/federal/district-courts/FSupp2/100/1058/2478126/. Citado na página 23.

KDE E.V. *Plasma*. 2022. Disponível em: https://kde.org/pt-br/products/>. Citado na página 24.

LEVY, S. *graphical user interface*. 2022. Disponível em: https://www.britannica.com/technology/graphical-user-interface. Citado na página 24.

LUDIICO. *Análise de hábitos de compras da população*. 2022. Disponível em: https://sites.google.com/view/ludiiprice/questionário. Citado 8 vezes nas páginas 33, 34, 35, 36, 37, 38, 39 e 40.

LUDIICO. *Diagrama de casos de uso*. 2022. Disponível em: https://sites.google.com/view/ludiiprice/engenharia-de-software. Citado na página 44.

MEYERS, G. J. *The Art of Software Testing*. Hoboken, New Jersey, USA: Wiley, 2011. ISBN 9781118031964. Citado 2 vezes nas páginas 74 e 76.

MICROSOFT. *Unit test basics*. 2022. Disponível em: https://docs.microsoft.com/en-us/visualstudio/test/unit-test-basics. Citado na página 75.

MICROSOFT. *Unit testing best practices with .NET Core and .NET Standard*. 2022. Disponível em: https://docs.microsoft.com/en-us/dotnet/core/testing/unit-testing-best-practices# lets-speak-the-same-language>. Citado na página 76.

MITCHELL, R. Web Scraping with Python: Collecting More Data from the Modern Web. [S.l.]: O'Reilly Media, 2018. ISBN 1491985577. Citado 3 vezes nas páginas 22, 23 e 24.

MOZILLA. *CRUD*. 2022. Disponível em: https://developer.mozilla.org/pt-BR/docs/Glossary/CRUD>. Citado na página 49.

PRESSMAN, R. *Software Engineering: A Practitioner's Approach*. [S.l.]: McGraw-Hill Education, 2009. ISBN 0073375977. Citado na página 66.

RECEITA FEDERAL. *Conceito, uso e obrigatoriedade da NF-e (26 questões)*. 2022. Disponível em: https://www.nfe.fazenda.gov.br/portal/perguntasFrequentes.aspx?tipoConteudo=4figqHYhYho=. Citado na página 15.

SEFAZ. *Nota fiscal de consumidor eletrônica (NFC-e)*. 2022. Disponível em: http://www.nfce.se.gov.br/portal/portalNoticias.jsp. Citado 2 vezes nas páginas 16 e 17.

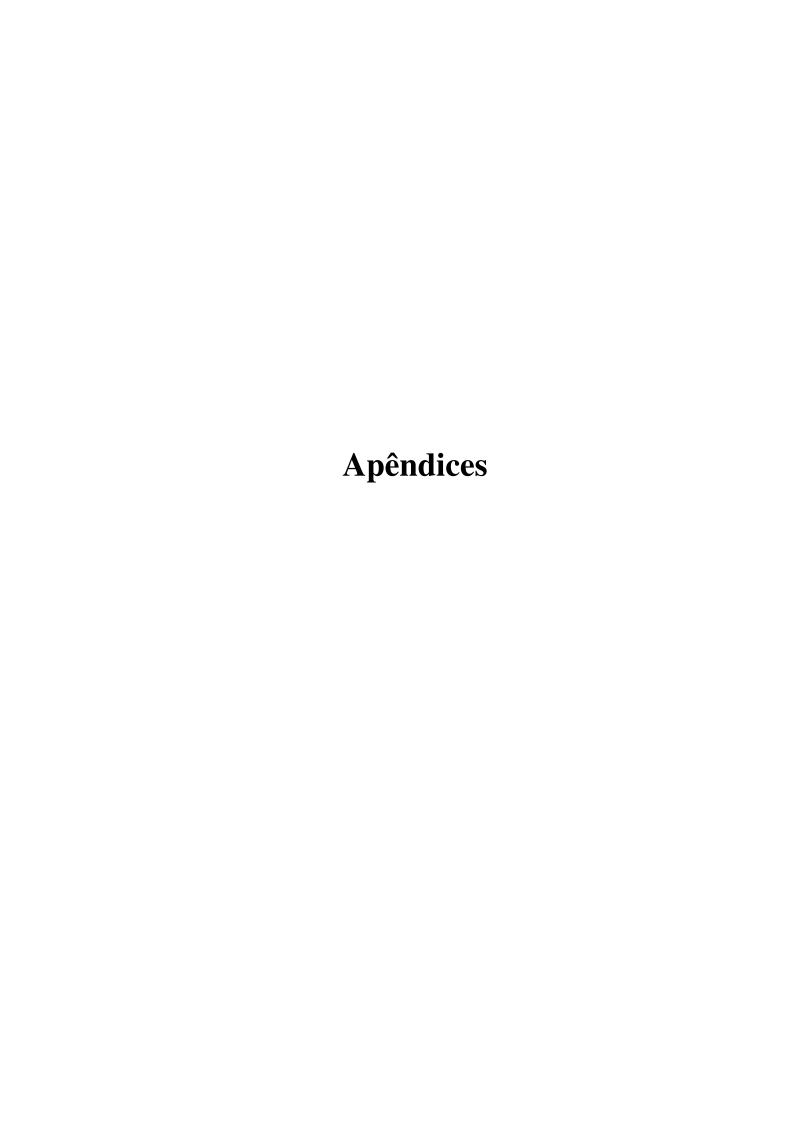
TANENBAUM ANDREW; AUSTIN, T. *Structured Computer Organization*. [S.l.]: Pearson, 2012. ISBN 9780132916523. Citado na página 25.

UFCG. *Diagrama de Use Cases*. 2022. Disponível em: http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/uml/diagramas/usecases/usecases.htm. Citado 2 vezes nas páginas 18 e 19.

UML. *What is UML*. 2005. Disponível em: https://www.uml.org/what-is-uml.htm. Citado na página 17.

Referências 81

VALVE CORPORATION. *Arquivo robots.txt da steam.* 2022. Disponível em: https://store.steampowered.com/robots.txt. Citado na página 23.



APÊNDICE A – Versões das tecnologias utilizadas

A seguir encontram-se as versões dos *plugins* utilizados no código da aplicação, a versão 2.8.1 do *flutter* foi utilizada durante o desenvolvimento.

Na escolha das dependências a serem utilizadas, considerou-se o tipo de licença pertencente aos *plugins* escolhidos, escolheu-se então apenas *plugins* com licenças gratuitas para a utilização.

A.1 Dependências do aplicativo

• app settings: 4.1.1

• equatable: 2.0.3

• flutter bloc: 8.0.0

• flutter charts: 0.3.1

• geolocator: 8.0.1

• get it: 7.2.0

• http: 0.13.4

• intl: 0.17.0

• **location**: 4.3.0

• mockito: 5.0.17

• path_provider: 2.0.8

• permission handler: 8.3.0

• qr_code_scanner: 0.6.1

• shared preferences: 2.0.11

• web_scraper: 0.1.4

• dartz: 0.10.1

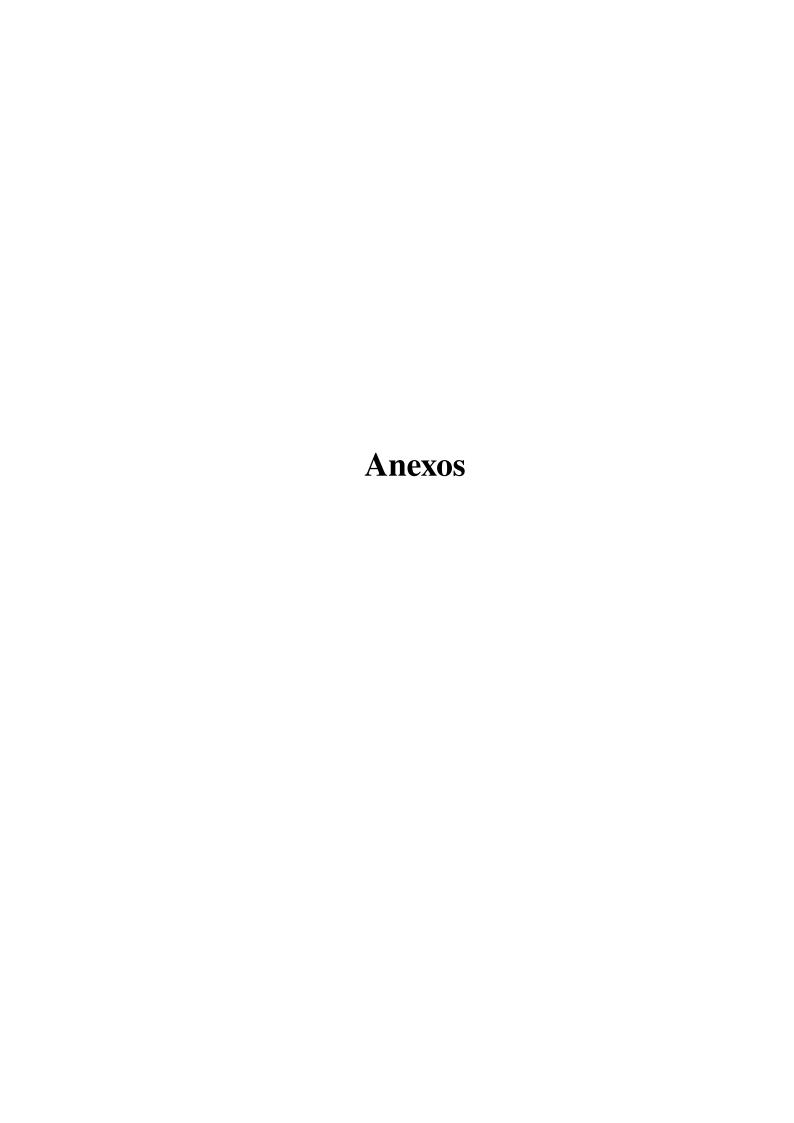
A.2 Dependências de desenvolvimento

• flutter_test: 0.0.0

• flutter_lints: 1.0.4

• build_runner: 2.1.7

• **bloc_test**: 9.0.1



ANEXO A – Manual do usuário da aplicação

A.1 Introdução

O LudiiPrice, na sua versão atual, é um aplicativo cujo objetivo é informar os preços de produtos de varejo próximos aos seus usuários. Os preços dos produtos são fornecidos através das leituras dos códigos QR de notas fiscais emitidas pelos estabelecimentos das compras.

A.2 Como utilizar o aplicativo

Esta seção descreve um passo a passo de como utilizar todas as funcionalidades do aplicativo.

A.2.1 Fazer login

Ao iniciar o aplicativo pela primeira vez, o usuário irá se deparar com a tela de login. Para realizar o login, basta inserir o seu email, a sua senha e clicar no botão "Entrar".

No caso de login bem sucedido, o aplicativo irá então para a tela de produtos populares, caso contrário o aplicativo emitirá uma mensagem de erro, informando o tipo de erro ocorrido.

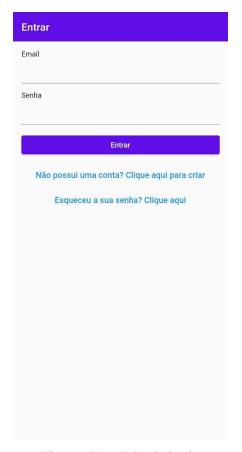


Figura 51 – Tela de login

A.2.2 Criar conta

A tela de criar conta é acessada ao selecionar a opção "Não possui uma conta? Clique aqui para criar".

Após selecionar essa opção, o usuário deve inserir o seu nome, email, senha e a sua confirmação de senha. Ao clicar no botão "Criar conta", o usuário deverá confirmar o seu email, clicando no link de confirmação enviado para o seu email inserido.



Figura 52 – Tela de criação de conta

A.2.3 Redefinir senha

A tela para redefinir uma senha é acessada ao selecionar a opção "Esqueceu a sua senha? Clique aqui" na tela de login.

Nessa tela, o usuário deve inserir o email utilizado em sua conta. Após isso, o usuário receberá um link em seu email para redefinir a sua senha.

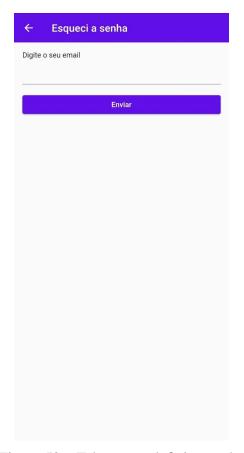


Figura 53 – Tela para redefinir a senha

A.2.4 Listas de compras

O usuário pode **consultar as suas listas de compras** criadas acessando a opção "lista de compras" no menu de navegação do aplicativo, localizado na tela inicial.

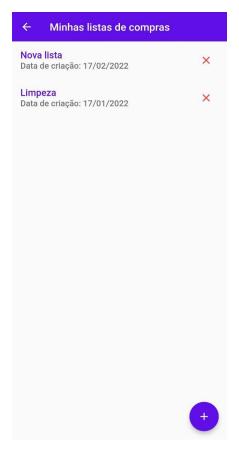


Figura 54 – Tela para consultar listas de compras

O usuário pode **criar uma nova lista de compras** ao clicar no botão com o símbolo +. Após isso, o usuário deve escolher o nome para a sua lista de compras, adicionar os produtos que deseja e confirmar a criação da sua lista de compras.



Figura 55 – Tela para criar uma lista de compras

O usuário pode **visualizar os produtos pertencentes a uma de suas listas de compras** ao selecionar uma das listas presentes na tela de consulta.

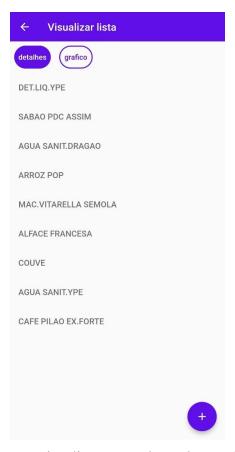


Figura 56 – Tela para visualizar os produtos de uma lista de compras

Na tela de visualização de uma lista de compras, o usuário pode **adicionar novos produtos à lista** selecionando o botão com o símbolo +.

O usuário pode **remover um produto da lista de compras** ao pressionar por alguns segundos sobre o produto que deseja remover, após isso uma mensagem de confirmação aparecerá, solicitando a confirmação da exclusão para o usuário.

O usuário pode visualizar o **gráfico de uma lista de compras** ao selecionar a opção "Gráfico", presente na tela de visualização dos detalhes de uma lista de compras. Este gráfico informa o preço dos produtos da lista de compras em diferentes estabelecimentos próximos ao usuário. Vale lembrar que nem todo estabelecimento possui todos os produtos de uma lista, esta tela exibe quais produtos estão presentes em cada estabelecimento do gráfico.



Figura 57 – Tela para visualizar o gráfico de uma lista de compras

O usuário pode **excluir uma lista de compras** ao apertar o símbolo X presente em uma lista de compras na tela de consultar listas de compras. Após selecionar o botão X, uma mensagem de confirmação aparecerá para confirmar a operação.

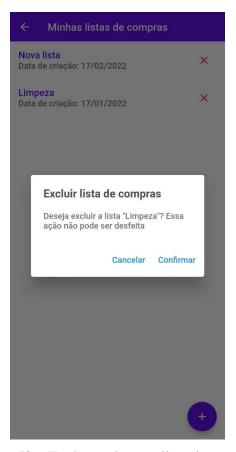


Figura 58 – Exclusão de uma lista de compras

A.2.5 Notas fiscais

O usuário pode **ler o código QR de uma nota fiscal** ao selecionar o botão com o símbolo de um código QR. Após esta seleção, o usuário irá se deparar com a tela responsável por ler o código QR. O usuário deve apontar a câmera para o código QR e aguardar a leitura do mesmo. Em caso de sucesso ou erro, uma mensagem aparecerá indicando a situação da operação.

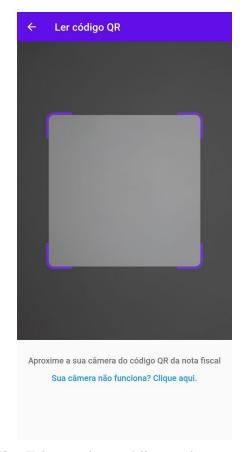


Figura 59 – Tela para ler o código qr de uma nota fiscal

O usuário pode **consultar as suas notas fiscais** lidas após selecionar a opção "Minhas notas" no menu de navegação.



Figura 60 – Tela para consultar notas fiscais lidas

O usuário pode ver os **detalhes de uma nota fiscal lida** ao selecionar uma nota fiscal presente na tela de consulta de notas fiscais.

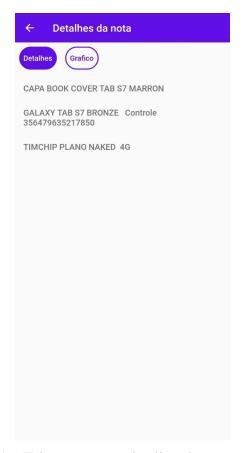


Figura 61 – Tela para ver os detalhes de uma nota fiscal

O usuário pode ver o **gráfico de preços de uma nota fiscal** ao selecionar a opção "Gráficos" presente na tela de visualizar detalhes de uma nota fiscal. A primeira coluna representa o preço da nota fiscal na data de emissão, as demais colunas representam o preço dos produtos da nota fiscal em diferentes estabelecimentos. Um texto explicativo encontra-se logo abaixo do gráfico de nota fiscal para esclarecer o funcionamento do gráfico para o usuário.

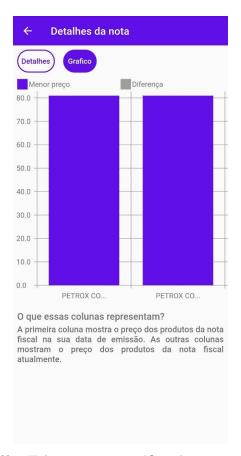


Figura 62 – Tela para ver o gráfico de uma nota fiscal

O usuário pode **excluir uma nota fiscal** lida ao selecionar o símbolo X ao lado de uma nota fiscal presente na tela de consulta de notas fiscais lidas.

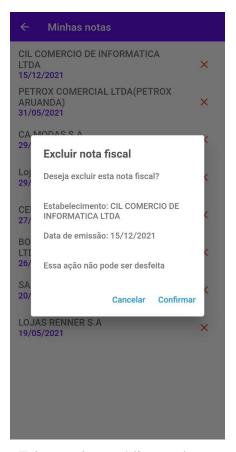


Figura 63 – Tela para ler o código qr de uma nota fiscal

A.2.6 Configurações

Esta seção descreve as funcionalidades presentes na tela de "Configurações", acessível a partir da seleção da opção "Configurações", presente no menu de navegação da tela inicial.

O usuário pode **alterar os seus filtros de busca** por produtos ao selecionar "Definir filtros" na tela de configurações. Ao selecionar esta opção, uma nova tela aparecerá, onde o usuário poderá arrastar os filtros para a esquerda ou para a direita, definindo os seus valores.



Figura 64 – Tela para alterar filtros de busca

O usuário pode **alterar a sua própria senha** ao selecionar a opção "Mudar a minha senha". Após esta seleção, o usuário deve inserir a sua senha atual e a sua nova senha nos respectivos campos da tela.



Figura 65 – Tela para alterar a própria senha

A.2.7 Despesas pessoais

O **gráfico de despesas pessoais** é acessível ao clicar na opção "Minhas despesas" presente no menu de navegação da tela inicial. Nesta tela o usuário pode visualizar os seus gastos mensais em um gráfico de linha simples.



Figura 66 – Gráfico de despesas pessoais

A.2.8 Produtos

A tela inicial do aplicativo exibe os **produtos populares** próximos à localização do usuário. A sua popularidade é calculada de acordo com o número de leituras de notas fiscais que possuem os produtos exibidos na tela.



Figura 67 – Tela de produtos populares

O usuário pode **buscar um produto** selecionando a opção "Buscar produtos", presente no menu de navegação da tela inicial. Ao entrar na nova tela, o usuário insere o termo da sua busca e clica no botão "Buscar". Os produtos resultantes da busca aparecerão, caso nenhum produto seja encontrado, uma mensagem aparecerá, informando que nenhum produto foi encontrado.

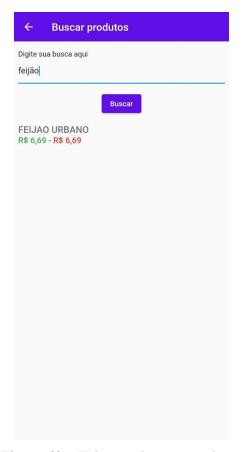


Figura 68 – Tela para buscar produtos

O usuário pode **visualizar os detalhes de um produto** ao selecionar um produto na tela inicial, na tela de busca, na tela de detalhes de uma lista de compras e na tela de detalhes de uma nota fiscal. Esta tela exibe os preços de um produto em diferentes estabelecimentos, a data de atualização do respectivo preço e um gráfico de variação de preço do produto em questão.



Figura 69 – Tela para visualizar detalhes de um produto