Universidade Federal de Ouro Preto
Escola de Minas
CECAU - Colegiado do Curso de
Engenharia de Controle e Automação

Natália Luiza Braz

Horti-Pure: Um Aplicativo de Apoio à Agricultura Familiar e Comercialização Sustentável de Alimentos em Ouro Preto

Monografia de Graduação

Ouro Preto, 2024

Natália Luiza Braz

Horti-Pure: Um Aplicativo de Apoio à Agricultura Familiar e Comercialização Sustentável de Alimentos em Ouro Preto

Trabalho apresentado ao Colegiado do Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como parte dos requisitos para a obtenção do Grau de Engenheiro(a) de Controle e Automação.

Universidade Federal de Ouro Preto

Orientador: Dra. Valéria de Carvalho Santos

Coorientador: Dr. Danny Augusto Vieira Tonidandel

Ouro Preto

2024

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
ESCOLA DE MINAS
DEPARTAMENTO DE ENGENHARIA CONTROLE E AUTOMACAO

FOLHA DE APROVAÇÃO

Natália Luiza Braz

Horti-Pure: Um Aplicativo de Apoio à Agricultura Familiar e Comercialização Sustentável de Alimentos em Ouro Preto

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Engenharia de Controle e Automação

Aprovado em 01 de novembro de 2024.

Integrantes da banca

[Doutora] - Valéria de Carvalho Santos - Orientadora (Universidade Federal de Ouro Preto)

[Doutor] - Danny Augusto Vieira Tonidandel - Coorientador (Universidade Federal de Ouro Preto)

[Mestre] - Fernando Henrique Oliveira Duarte - (Universidade Federal de Ouro Preto)

[Mestre] - Josemar Coelho Felix - (Universidade Federal de Ouro Preto)

Danny Augusto Vieira Tonidandel, coorientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 01/11/2024.

Documento assinado eletronicamente por **Danny Augusto Vieira Tonidandel**, **PROFESSOR DE MAGISTERIO SUPERIOR**, em 01/11/2024, às 09:42, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0 , informando o código verificador **0797430** e o código CRC **C4B373D1**.

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35402-163
Telefone: 3135591533 - www.ufop.br

Resumo

O uso intensivo de agrotóxicos no agronegócio brasileiro tem gerado preocupações em relação à segurança alimentar e nutricional da população. Essa prática agrícola, embora tenha aumentado a produtividade, trouxe impactos negativos à saúde humana e ao meio ambiente. Diante desse cenário, a agricultura familiar emerge como uma alternativa promissora, valorizando práticas agroecológicas que promovem a produção de alimentos mais saudáveis, livres de agrotóxicos. Nesse contexto, propõe-se o desenvolvimento de um aplicativo móvel para Android com o objetivo de facilitar a comercialização de produtos agroecológicos/orgânicos provenientes da agricultura familiar, contribuindo para a promoção do desenvolvimento sustentável na cidade de Ouro Preto. Com a crescente expansão do comércio eletrônico no Brasil, o aplicativo se apresenta como uma solução tecnológica inovadora, permitindo também que os agricultores compartilhem informações sobre suas práticas agrícolas e estreitem o vínculo com os consumidores.

Palavras-chaves: agrotóxicos. agricultura familiar. agroecologia. aplicações móveis. e-commerce. Kotlin. Android.

Abstract

The intensive use of pesticides in Brazilian agribusiness has raised significant concerns regarding food and nutritional security for the population. Despite its productivity benefits, this agricultural practice has resulted in adverse impacts on both human health and the environment. In response to this challenge, family farming emerges as a promising alternative, emphasizing agroecological practices that promote the production of healthier foods without the use of harmful pesticides. To address this issue, an Android mobile application is proposed to facilitate the commercialization of agroecological/organic products from family farming, thereby contributing to the promotion of sustainable development in Ouro Preto. Leveraging the expanding e-commerce landscape in Brazil, the application serves as an innovative technological solution, enabling farmers to share valuable information about their agricultural practices and establishing stronger connections with consumers.

Key-words: pesticides. family farming. agroecology. mobile applications. e-commerce. Kotlin. Android.

Lista de ilustrações

Figura 1 – Gráfico crescimento do e-commerce . . . 17
Figura 2 – Camadas da Engenharia de Software . . . 17
Figura 3 – Mapa mundial Android e iOS . . . 19
Figura 4 – Aplicativos desenvolvidos com Kotlin . . . 20
Figura 5 – Representação da arquitetura MVVM . . . 22
Figura 6 – Formato JSON . . . 23
Figura 7 – Diagrama de Caso de Uso geral . . . 39
Figura 8 – Protótipo média fidelidade: tela Inicial, home (convidado), cadastrar e logar . . . 40
Figura 9 – Protótipo média fidelidade: tela home (logado), pesquisar, detalhes do mercado e detalhes do produto . . . 40
Figura 10 – Protótipo média fidelidade: tela educacional (convidado), educacional (logado), criar publicação e perfil . . . 41
Figura 11 – Protótipo média fidelidade: tela editar perfil e perfil (convidado) . . . 42
Figura 12 – Cores . . . 43
Figura 13 – Protótipo alta fidelidade: tela Inicial, home (convidado), cadastrar e logar 44
Figura 14 – Protótipo alta fidelidade: tela home (logado), pesquisar e resultados da busca . . . 45
Figura 15 – Protótipo alta fidelidade: telas de detalhes do mercado, detalhes dos produtos e educacional . . . 45
Figura 16 – Protótipo alta fidelidade: tela de perfil do usuário logado e convidado . 46
Figura 17 – Domínios . . . 47
Figura 18 – Azure Board - Sprint 1 . . . 49
Figura 19 – Azure Board - Sprint 2 . . . 50
Figura 20 – Azure Board - Sprint 3 . . . 50
Figura 21 – Azure Board - Sprint 4, primeira parte . . . 51
Figura 22 – Azure Board - Sprint 4, segunda parte . . . 51
Figura 23 – Azure Board - Sprint 5, primeira parte . . . 52
Figura 24 – Azure Board - Sprint 5, primeira parte . . . 52
Figura 25 – Azure Board - Sprint 6 . . . 53
Figura 26 – Azure Board - Sprint 7 . . . 53
Figura 27 – Splash Screen . . . 54
Figura 28 – Tela Home usuário convidado . . . 55
Figura 29 – Tela Login . . . 56
Figura 30 – Tela Cadastrar . . . 56
Figura 31 – Tela Home usuário autenticado . . . 57

Figura 32 – Tela de Perfil usuário autenticado 58
Figura 33 – Tela de Perfil usuário convidado 58
Figura 34 – Permissão de localização 59
Figura 35 – Pop-up informativo sobre permissão de localização 59
Figura 36 – Tela de busca sem permissão concedida 60
Figura 37 – Resultados da barra de pesquisa 60
Figura 38 – Visualização do mapa com botão de busca manual 61
Figura 39 – Nenhum resultado encontrado na busca 61
Figura 40 – Tela de busca com permissão concedida 62
Figura 41 – Lista de resultados 62
Figura 42 – Tela de Detalhes do Mercado 63
Figura 43 – Tela de Detalhes dos Produtos 64

Lista de tabelas

Tabela 1 – Descrição das principais bibliotecas utilizadas 29
Tabela 2 – Requisitos funcionais 30
Tabela 3 – Regras de negócio 33
Tabela 4 – Caso de uso: Cadastro de usuário 34
Tabela 5 – Caso de uso: Fazer login 35
Tabela 6 – Caso de uso: Pesquisar mercado 35
Tabela 7 – Caso de uso: Visualizar mercados próximos 36
Tabela 8 – Caso de uso: Fazer um pedido 37
Tabela 9 – Caso de uso: Deslogar 38
Tabela 10 – Caso de uso: Deletar conta 38

Lista de abreviaturas e siglas

FAO/ONU	Organização das Nações Unidas para Alimentação e Agricultura
Anvisa	Agência Nacional de Vigilância Sanitária
PARA	Programa de Análise de Resíduos de Agrotóxicos em Alimentos
ONU	Organização das Nações Unidas
HTTP	Hypertext Transfer Protocol
API	Application Programming Interface
IU	Interface do Usuário
URL	Uniform Resource Locator
JSON	JavaScript Object Notation
PNAN	Política Nacional de Alimentação e Nutrição
Brasilbio	Associação Brasileira de Orgânicos
PRONAF	Programa Nacional de Fortalecimento da Agricultura Familiar
PAA	Programa de Aquisição de Alimentos
MVC	Model, View, Controller
MVP	Model, View, Presenter
MVVM	Model, View, View-Model
XML	Extensible Markup Language
REST	Representational State Transfer
B2C	Business to Consumer

Sumário

1 INTRODUÇÃO 11
1.1 Justificativa 12
1.2 Objetivos 13
1.2.1 Objetivo Geral 13
1.2.2 Objetivos Específicos 13
1.3 Organização e Estrutura 13

2 FUNDAMENTAÇÃO TEÓRICA 15
2.1 A Agricultura Familiar e a Busca por uma Produção Sustentável . . 15
2.1.1 O Potencial do E-commerce 16
2.2 Fundamentos de Desenvolvimento de Software e Tecnologias Utilizadas 17
2.2.1 Engenharia de Software 17
2.2.2 Metodologias Ágeis 18
2.2.3 Sistema Operacional Android 19
2.2.4 Linguagem de programação Kotlin 19
2.2.5 Padrões de Arquitetura 20
2.2.6 API REST 22

3 TRABALHOS RELACIONADOS 24
3.1 Greenpeace - Guia de produtores agroecológicos e orgânicos 24
3.2 Naturis 24
3.3 Agromart 24
3.4 Natural da terra 25

4 DESENVOLVIMENTO 26
4.1 Ferramentas Utilizadas 26
4.2 Metodologia 29
4.2.1 Requisitos Funcionais 30
4.2.2 Requisitos Não-Funcionais 30
4.2.3 Regras de Negócio 32
4.2.4 Casos de Uso 33
4.2.5 Prototipagem e Design 38
4.2.6 Implementação 46
4.2.6.1 Backend 47
4.2.6.2 Frontend 48

4.2.6.3 Organização dos Ciclos de Trabalho 48

5 RESULTADOS 54

6 CONSIDERAÇÕES FINAIS 65

6.1 Trabalhos Futuros 65

Referências 67

1 Introdução

O avanço do agronegócio no Brasil esteve relacionado à implementação das práticas da Revolução Verde, respaldada pelo apoio governamental brasileiro. Essas práticas buscavam fortalecer a agricultura através do cultivo intensivo do solo, da monocultura e da aplicação de insumos químicos para aumentar a produtividade, tendo supostamente como finalidade resolver o problema da fome no mundo (GONÇALVES ROCHA; RIBEIRO, 2022). No entanto, o enfoque desviou-se da produção de alimentos para priorizar a produção de *commodities* (soja, milho e cana-de-açúcar), transformando-as em elementos comerciais negociados na bolsa de valores, acarretando em um aumento significativo no uso de agrotóxicos no país (DUTRA; SOUZA, 2017).

Segundo relatório de 2022 da Organização das Nações Unidas para Alimentação e Agricultura (FAO/ONU), o Brasil é o segundo país que utiliza maior volume de agrotóxicos no mundo (ALVES F. V.; ALVA, 2022). Essa intensificação no uso desses produtos tem ocasionado diversos impactos negativos, incluindo a poluição ambiental e intoxicação tanto de trabalhadores rurais quanto da população em geral, resultando em prejuízos agudos e crônicos. Entre os graves problemas de saúde identificados no campo científico, destacam-se casos de infertilidade, impotência, abortos, malformações, neurotoxicidade, desregulação hormonal, efeitos sobre o sistema imunológico, câncer, entre outros (GARCIA; LARA, 2020).

Em resposta à crescente preocupação com esses efeitos adversos, a Anvisa (Agência Nacional de Vigilância Sanitária) criou, em 2001, o PARA (Programa de Análise de Resíduos de Agrotóxicos em Alimentos), com o propósito de estruturar um serviço para avaliar e promover a qualidade dos alimentos em relação ao uso de agrotóxicos e substâncias correlatas. Os relatórios anuais do programa têm constituído um dos principais indicadores da qualidade dos alimentos adquiridos no mercado varejista e consumidos pela população.

Segundo Teixeira (2017), o relatório publicado em 2012 revelou resultados alarmantes para alimentos de origem vegetal, que representam cerca de 30% da dieta da população brasileira. Os dados apontaram a presença de agrotóxicos não autorizados para determinadas culturas e em doses acima dos limites permitidos. De maneira mais crítica, o abacaxi, a cenoura, o morango e o pepino foram identificados com níveis preocupantes de contaminação, atingindo respectivamente 41%, 33%, 59% e 42%.

Fica claro portanto, que esse modelo de agricultura químico-dependente ameaça a Segurança Alimentar e Nutricional, definida como a realização do direito de todos ao acesso regular e permanente a alimentos de qualidade, em quantidade suficiente, sem comprometer o acesso a outras necessidades essenciais, tendo como base práticas alimentares promotoras de saúde que respeitem a diversidade cultural e que sejam ambiental, cultural, econômica

e socialmente sustentáveis (BURITY et al., 2010).

Através do Relatório do Relator Especial sobre o direito à alimentação (*Report of the Special Rapporteur on the right to food*) elaborado pela ONU (Organização das Nações Unidas), foi salientada a necessidade urgente de mudanças nos sistemas de produção alimentar global, com fortes recomendações em favor da agroecologia (SCHUTTER, 2009). Segundo Machado (2013), agroecologia é um método de produção agrícola que resgata os saberes tradicionais e incorpora os progressos científicos e tecnológicos em harmonia com o meio ambiente, produzindo alimentos e produtos limpos, sem agrotóxicos. Por se tratar de um meio de produção de menor custo, é uma alternativa válida para os produtores de agricultura familiar que não possuem tanto acesso ao crédito e às grandes tecnologias empregadas na agricultura tradicional.

Nesse sentido, políticas voltadas para a produção agroecológica têm surgido como alternativa sustentável para o meio rural, oferecendo estratégias tecnocientíficas para o desenvolvimento do mesmo. Essas iniciativas visam promover a permanência das famílias no campo, através do manejo sustentável dos solos, conservação dos recursos naturais e valorização dos saberes locais, permitindo a independência dos pequenos agricultores em suas vendas (SANTOS et al., 2014).

1.1 Justificativa

Carvalho e Carvalho (2015), em seu estudo sobre a Utilização da Internet e Adoção do E-Commerce pelas Organizações da Agricultura Familiar Brasileira, alega que entre os bloqueios para o pleno desenvolvimento do setor no Brasil sobressai a dificuldade no processo de comercialização dos produtos gerados, uma vez que o mercado em si acaba favorecendo os grandes produtores, em razão do isolamento das comunidades de agricultura familiar e sua distância até os centros consumidores. Porém, com o advento da conexão com a Internet, essa distância tem o potencial de se tornar cada vez menor, facilitando a integração de mercado.

Entre os benefícios da inserção dos agricultores no e-commerce citados por Baourakis, Kourgiantakis e Migdalas (2002), evidenciam-se:

1. Redução nos custos de intermediação;
2. Melhoramento do fluxo de informações;
3. Perspectiva de expansão de mercado.

O primeiro benefício está relacionado ao lucro do produtor, que muitas vezes é reduzido pela presença de intermediários na venda que acabam ficando com a maior parte dos ganhos (KIYOTA; GOMES, 1999). O segundo benefício está ligado à exposição de produtos e preços

de forma transparente, permitindo a comparação, facilitando o acesso dos consumidores e até mesmo simplificando a logística das operações de venda. O terceiro benefício sugere uma oportunidade de crescimento e aumento das vendas, através de engajamento de novos consumidores como consequência do acesso à informação facilitado.

Dessa forma, torna-se relevante desenvolver um aplicativo móvel que facilite a comercialização de produtos agroecológicos/orgânicos provenientes da agricultura familiar na cidade de Ouro Preto. Mesmo com uma significativa quantidade de cerca de 270 agricultores familiares, ainda não há disponibilidade de uma solução tecnológica semelhante (OURO PRETO, 2022). Esse aplicativo não apenas atenderá à crescente demanda por segurança alimentar, mas também proporcionará maior comodidade aos consumidores interessados.

1.2 Objetivos

1.2.1 Objetivo Geral

Este trabalho tem como principal objetivo o desenvolvimento de uma aplicação *front-end* Android que busca facilitar a venda de produtos or

1.2.2 Objetivos Específicos

- Fazer uma interface entre consumidor e produtor, facilitando o acesso e a comercialização dos produtos;
- Proporcionar aos agricultores um espaço para que compartilhem informações sobre suas produções, trajetória, e conhecimentos, humanizando esse processo e estreitando o laço produtor-consumidor;

1.3 Organização e Estrutura

O conteúdo deste trabalho está organizado em 6 capítulos, descritos abaixo:

No capítulo 1 são apresentados a introdução ao tema, a justificativa da pesquisa, bem como seus objetivos. Incluindo também o presente tópico, o qual delineia os principais assuntos abordados em cada capítulo subsequente.

O capítulo 2 é dedicado a uma fundamentação teórica dividida em duas partes: Na primeira parte são apresentados alguns conceitos importantes para fundamentar a proposta do trabalho, em relação ao contexto da agricultura familiar. A segunda parte esclarece alguns conceitos mais técnicos e relevantes para o desenvolvimento do aplicativo, essenciais para que o leitor tenha uma melhor compreensão da obra.

O capítulo 3 faz um breve apanhado dos principais trabalhos relacionados ao propósito do presente estudo, buscando identificar e analisar soluções já existentes.

O capítulo 4 concentra-se no estágio de construção do aplicativo, abordando as ferramentas tecnológicas utilizadas e a metodologia empregada no projeto. Nesta seção, são descritas detalhadamente todas as etapas envolvidas, desde a especificação de requisitos e casos de uso até a criação do protótipo final. Além disso, são discutidos aspectos como a modelagem e construção da API, a configuração de *DevOps* e a arquitetura do sistema. Por fim, são abordados os processos de implementação que culminam na finalização do projeto.

No capítulo 5 são apresentados os resultados alcançados pelo aplicativo, verificando se os requisitos iniciais estabelecidos no tópico de metodologia foram cumpridos.

E por último, o capítulo 6 traz as considerações finais, abordando também as limitações encontradas e propostas de trabalhos futuros, visando a expansão do projeto.

2 Fundamentação Teórica

Neste capítulo, serão abordados conceitos fundamentais para a compreensão deste trabalho. O capítulo está dividido em duas partes: a primeira explora temas relacionados à agricultura familiar, a produção sustentável e a expansão do *e-commerce*. A segunda parte foca em explicar os conceitos técnicos das tecnologias empregadas no desenvolvimento do projeto.

2.1 A Agricultura Familiar e a Busca por uma Produção Sustentável

Conforme aconselha o Ministério da Saúde Brasil (2006) através do guia alimentar brasileiro, determinado na PNAN (Política Nacional de Alimentação e Nutrição), as práticas alimentares saudáveis devem priorizar o resgate de hábitos alimentares regionais, com foco no consumo de alimentos produzidos localmente, culturalmente referenciados, de alto valor nutritivo e com garantias de segurança sanitária.

Considerando esse aspecto, surge o problema dos agrotóxicos presentes nesses tipos de alimentos, que colocam em risco a segurança alimentar e nutricional da população. Segundo Moreira et al. (2002), existem três vias de contaminação humana por agrotóxicos: a via ocupacional, que se caracteriza pela contaminação dos trabalhadores que manipulam essas substâncias, seja no processo de formulação, utilização, como também na colheita; a via ambiental, por sua vez, caracterizada pela dispersão/distribuição dos agrotóxicos ao longo dos diversos componentes do meio ambiente; e a via alimentar caracterizada pela contaminação relacionada à ingestão de produtos contaminados por agrotóxicos.

Dessa maneira, se faz necessário repensar os modelos de produção vigentes, que visam apenas o lucro, em detrimento da qualidade entregue. Nessa perspectiva, surge a agricultura familiar. De acordo com Tinoco (2012), os autores Bittencourt e Bianchini (1996) consideram que um agricultor familiar é aquele que tem a agricultura como principal fonte de renda (acima de 80%), e que a força de trabalho utilizada na propriedade é predominantemente exercida pela família.

Segundo dados da Brasilbio (Associação Brasileira de Orgânicos), entidade que reúne os produtores, processadores e certificadores, 80% dos produtores de alimentos orgânicos no país são agricultores familiares (LAGE et al., 2016). Para Badgley et al. (2007), o termo 'orgânico' se refere a práticas agrícolas que utilizam processos naturais (não sintéticos) de ciclagem de nutrientes, excluem ou minimizam o uso de agrotóxicos, e promovem a regeneração da qualidade do solo. Suas pesquisas sugerem que a agricultura orgânica, devidamente intensificada, pode suprir grande parte da demanda mundial por

alimentos e beneficiar especialmente países em desenvolvimento, aumentando sua segurança alimentar.

Os programas de incentivo à agricultura familiar, como o PRONAF (Programa Nacional de Fortalecimento da Agricultura Familiar), criado em 1996, e o PAA (Programa de Aquisição de Alimentos), têm como objetivo facilitar o acesso a créditos e investimentos para viabilizar a produção e a comercialização dos produtos dos agricultores familiares (SILVA, 2012). Porém, de acordo com De Arruda Saron, Neto e Hespanhol (2017) ambos os programas têm falhado ao priorizar o acesso aos segmentos integrados às principais cadeias produtivas do agronegócio, ligados à exportação de grãos e carnes, em detrimento dos segmentos menos capitalizados, os quais muitas vezes enfrentam dificuldades para acessar os recursos dessa política pública federal.

É evidente, portanto, a necessidade de criar espaços que facilitem a inserção desse segmento no mercado. Para isso, é fundamental promover a conscientização sobre a importância de uma alimentação saudável, alertando sobre os riscos dos agrotóxicos, e incentivar a aquisição de produtos mais naturais provenientes dos agricultores familiares.

2.1.1 O Potencial do E-commerce

Com o advento do crescimento da internet e do *e-commerce*, hoje em dia, uma boa parcela da população possui esse acesso facilitado por *smartphones*. De acordo com Albertin (2010), o comércio eletrônico refere-se à completa integração da cadeia de valor dos processos de negócio em um ambiente eletrônico, onde as tecnologias de informação e comunicação são amplamente empregadas para alcançar os objetivos comerciais.

Segundo Vissotto e Boniati (2013), o *e-commerce* representa uma revolução no conceito de mercado, abrindo caminho para oportunidades de negócios na forma de comércio virtual. Essa modalidade de comércio proporciona aos consumidores maior comodidade, segurança e confiabilidade ao realizar suas compras, tanto para produtos e serviços reais como para os virtuais.

Para Cruz (2021) três eventos foram fundamentais para a solidificação e a expansão do *e-commerce* no país: a popularização do uso da internet, a difusão do uso de *smartphones* e a pandemia da Covid-19.

Observando a Figura 1 abaixo, é possível notar o crescimento do *e-commerce* nos anos de 2011 a 2020, inclusive uma acentuação em 2020.

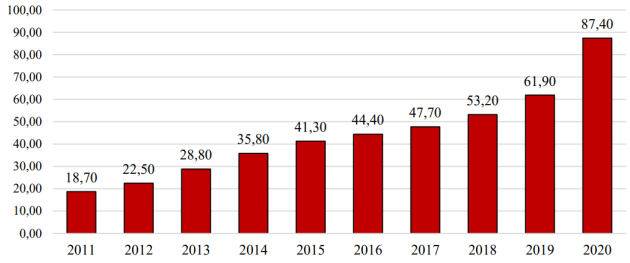

Figura 1 – BRASIL: Faturamento do e-commerce (2011-2020) em R$ bilhões Fonte: Cruz (2021).

2.2 Fundamentos de Desenvolvimento de Software e Tecnologias Utilizadas

Nesta seção, são apresentados os conceitos teóricos essenciais para compreender o processo de desenvolvimento do software, incluindo as tecnologias utilizadas, a linguagem de programação adotada e a arquitetura aplicada.

2.2.1 Engenharia de Software

Os autores do livro Engenharia de *Software* - Uma Abordagem Profissional, Pressman e Maxim (2021), definem o *software* como instruções executáveis que fornecem as características e o desempenho desejados. Eles o consideram tanto um produto quanto o meio para distribuí-lo, sendo um transformador de informações.

O desenvolvimento de um *software* é uma tarefa árdua que requer um planejamento a fim de maximizar a qualidade de seus resultados. Esse processo é sustentado pelos pilares da Engenharia de Software, observados na Figura 2 abaixo:

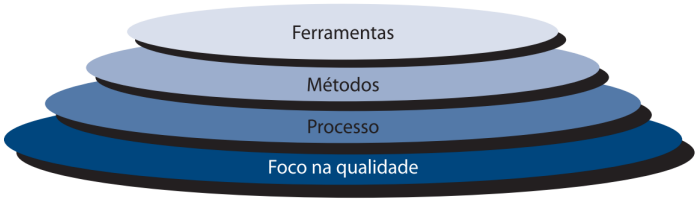

Figura 2 – Pilares da Engenharia de Software Fonte: Pressman e Maxim (2021).

- **Qualidade:** Comprometimento organizacional com a excelência na entrega.
- **Processo:** Definição de uma metodologia para a efetiva entrega de tecnologia de engenharia de software, garantindo coesão e racionalidade no desenvolvimento no prazo estabelecido.

- **Métodos:** Fornecimento de informações técnicas para o desenvolvimento de software, abrangendo tarefas como comunicação, análise de requisitos, modelagem, programação, testes e suporte.
- **Ferramentas:** Suporte automatizado ou semi automatizado para o processo e métodos.

2.2.2 Metodologias Ágeis

Em resposta às limitações apresentadas pelos modelos tradicionais de processos da Engenharia de Software, como o modelo em Cascata, que se caracteriza pelo gerenciamento de projetos por meio de fases sequenciais, com longo planejamento e projetos definidos por custos, escopo e cronograma fixos, surgiu a abordagem da Engenharia de Software Ágil. Esta nova abordagem, coloca o foco na satisfação do cliente, na flexibilidade e na entrega contínua de valor ao longo do processo, utiliza equipes de projeto pequenas e altamente motivadas e, acima de tudo, valoriza a simplicidade no desenvolvimento em geral.

Dentro desse contexto ágil, uma das metodologias mais amplamente adotadas é o Scrum. Criado por Schwaber e Sutherland (2013), *Scrum* é um *framework* de desenvolvimento ágil que se destaca por sua capacidade de resolver problemas complexos e adaptativos, permitindo a entrega de produtos de alto valor de maneira produtiva e criativa. O *Scrum* organiza o trabalho em torno de iterações curtas e focadas, conhecidas como sprints, e envolve times *Scrum* estruturados com papéis, eventos, artefatos e regras, cada um desempenhando um papel essencial para o sucesso do *framework*.

As iterações do Scrum são chamadas de ***Sprints***, ciclos rápidos de trabalho com entregas parciais em intervalos de 1 a 4 semanas. O ***Backlog* da *Sprint*** contém as funcionalidades atribuídas a cada *Sprint*, enquanto o ***Backlog* do Produto** prioriza as funcionalidades desejadas pelo cliente.

O **time Scrum** é composto pelo ***Product Owner***, responsável por coordenar o time e gerenciar o Backlog do Produto; pelo **time de desenvolvimento**, uma equipe multidisciplinar e auto-organizada; e pelo ***Scrum Master***, que facilita o método e garante sua correta aplicação.

Os eventos do *Scrum* incluem o **Planejamento da *Sprint***, onde o trabalho a ser realizado é definido visando as entregas que se deseja fazer; as **Reuniões Diárias**, breves encontros diários para compartilhar progresso e obstáculos; a **Revisão da *Sprint***, realizada no final da *Sprint* para inspecionar o incremento do produto; e a **Retrospectiva da *Sprint***, que permite ao time analisar erros e acertos e planejar melhorias para a próxima *Sprint*.

2.2.3 Sistema Operacional Android

O Android é o sistema operacional mais popular do mundo, contando com mais de 2,5 bilhões de usuários ativos globalmente. Inicialmente desenvolvido pela empresa Android Inc., como uma alternativa ao iOS, o Android foi adquirido pela Google em 2005 e lançado oficialmente em 2008, marcando o início da era dos dispositivos móveis com esse sistema operacional (CURRY, 2023).

Diferente do concorrente iOS, o Android é baseado no núcleo Linux e possui a vantagem de ser um software de código aberto. Essa característica possibilita a adição de novos recursos e o aperfeiçoamento contínuo por parte de novos programadores, o que contribui para sua ampla adoção e popularidade. Devido à facilidade de disponibilidade e atraentes possibilidades de personalização, grandes fabricantes como Samsung, LG, Motorola e Sony Ericsson rapidamente adotaram o sistema Android em seus *smartphones* (QUEIROZ, 2018).

Comparando a distribuição de utilização entre Android e iOS pelo mundo, a Figura 3 revela que o sistema da Google domina com 71% de utilização, enquanto o sistema da Apple detém 28%.

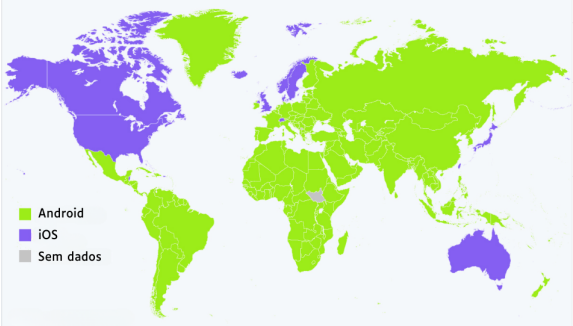

Figura 3 – Distribuição de utilização do Android e iOS pelo mundo Fonte: (STATISTA, 2023).

Em suma, a escolha do Android como plataforma para desenvolvimento do aplicativo é motivada por sua ampla base de usuários, sua flexibilidade, ferramentas de desenvolvimento disponíveis, integração com os serviços do Google, diversidade de dispositivos suportados e o suporte da comunidade.

2.2.4 Linguagem de programação Kotlin

A introdução do Kotlin como uma linguagem de programação moderna e eficiente trouxe uma mudança significativa no cenário do desenvolvimento Android. Enquanto o Java, por muito tempo dominante, era conhecido por sua verbosidade e quantidade excessiva de código, o Kotlin surgiu em 2016 como uma alternativa da JetBrains, localizada em São Petersburgo, Rússia (BOSE, 2018).

Seu principal objetivo é oferecer códigos mais concisos, legíveis e eliminar o *boiler-plate code*. Esse termo refere-se a trechos de código repetitivos e padronizados que, embora necessários, não agregam valor direto à lógica do programa, mas acabam aumentando o tamanho e a complexidade do código. Para isso Kotlin oferece tipagem estática, o que significa que os tipos são resolvidos na compilação e permanecem constantes. Além disso, a linguagem suporta programação orientada a objetos e funcional, proporcionando uma experiência flexível para os desenvolvedores. Em maio de 2018, o time de Android da Google anunciou Kotlin como a linguagem oficial para desenvolvimento Android, resultando em documentação e ferramentas do Android pensadas para integração perfeita com o Kotlin (FLAUZINO et al., 2018).

Dentre as vantagens notáveis do Kotlin estão a interoperabilidade, que permite a utilização de código Java em projetos Kotlin e vice-versa; a inferência de tipo, eliminando a necessidade de especificar tipos de variáveis explicitamente; o *Null-safety*, que protege contra *NullPointer Exceptions* ao lidar com referências nulas; e a imutabilidade, através da declaração de variáveis como *val*, melhorando a segurança e compreensão do código. Outras funcionalidades poderosas incluem as *Funções de Extensão*, que permitem expandir classes ou interfaces sem herança ou padrões de design complexos, e as *Funções de Ordem Superior*, que permitem o uso de funções como argumentos e resultados de outras funções.

Aplicativos construídos com Kotlin

A figura 4 mostra a logo de alguns exemplos de aplicativos construídos com Kotlin:

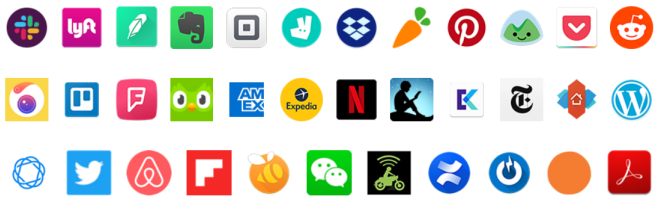

Figura 4 – Exemplos de aplicativos desenvolvidos com a linguagem de programação Kotlin (DEVELOPERS, A., 2021)

2.2.5 Padrões de Arquitetura

Um aplicativo Android irá conter múltiplos componentes de aplicativo, e à medida que cresce, é preciso que se mantenha robusto, testável, e sustentável. Para isso é preciso haver uma preocupação com a arquitetura a ser utilizada, pois ela que irá definir os limites entre as partes do aplicativo e as responsabilidades que cada uma deve ter (DEVELOPERS, A., 2023b).

A diversidade de padrões de arquitetura pode trazer consigo a dificuldade para a escolha daquela mais adequada para a finalidade desejada. Dentre os padrões populares,

MVC (Model View Controller), MVP (Model View Presenter) e MVVM (Model, View, View-Model) são opções comuns.

MVC é um padrão clássico que separa a aplicação em três componentes principais: **Model**, responsável pela lógica e dados da aplicação; **View**, que gerencia a interface do usuário; e **Controller**, que serve como intermediário entre a View e o Model. Embora seja a arquitetura padrão para desenvolvimento Android, o MVC apresenta problemas de acoplamento entre a View e o Controller, dificultando a manutenção e o desenvolvimento, o que levou ao surgimento de alternativas (EPILOKSA; KUSUMO; ADRIAN, 2022).

MVP é uma evolução do MVC, onde o **Presenter** assume a responsabilidade de mediar a comunicação entre a **View** e o **Model**, mantendo a lógica de apresentação fora da View e facilitando a testabilidade e manutenção (EPILOKSA; KUSUMO; ADRIAN, 2022).

MVVM vai um passo além, introduzindo o **View-Model**, que é responsável por gerenciar o estado da **View** e atuar como uma ponte entre o **Model** e a **View**. MVVM permite a vinculação automática de dados entre a View e o View-Model, o que simplifica a lógica de interface e melhora a testabilidade (EPILOKSA; KUSUMO; ADRIAN, 2022).

Neste padrão MVVM, a **View** é a representação visual da aplicação, incluindo Activities, Fragments, arquivos de layout XML, controles de interface do usuário e a lógica para renderizar dados e capturar a entrada do usuário. A View observa os dados no componente ViewModel e realiza atualizações quando os dados mudam. A **ViewModel** atua como uma mediadora entre a View e o Model. Ela fornece os dados necessários para a View e lida com a lógica de apresentação, como formatação de dados e tratamento de eventos. Isso permite que a View observe qualquer mudança nos dados, mantendo a interface atualizada conforme as informações são alteradas. O **Model** representa os dados e a lógica de negócio do aplicativo. O repositório é responsável por fornecer os dados necessários à aplicação, os quais podem ser obtidos de duas maneiras: remotamente, utilizando uma REST API com a ajuda da biblioteca Retrofit, ou localmente, utilizando um banco de dados Room (EPILOKSA; KUSUMO; ADRIAN, 2022).

Estudos conduzidos por Lou et al. (2016) comprovaram empiricamente que MVP e MVVM são superiores em testabilidade, modificabilidade e desempenho em relação ao MVC. Conquanto, comparando MVP e MVVM, ambas apresentaram desempenho semelhantes, mas o MVP fornece melhor capacidade de modificação, enquanto MVVM fornece melhor capacidade de teste.

Ainda assim, Epiloksa, Kusumo e Adrian (2022) apontam algumas vantagens do padrão MVVM, incluindo o armazenamento de dados em forma de *Live Data* na *ViewModel*, resolvendo o problema de travamentos no aplicativo devido à cessação da atividade, e também o problema de definir manualmente o ciclo de vida, reduzindo a complexidade do código.

A figura 5 apresenta o diagrama da Arquitetura MVVM.

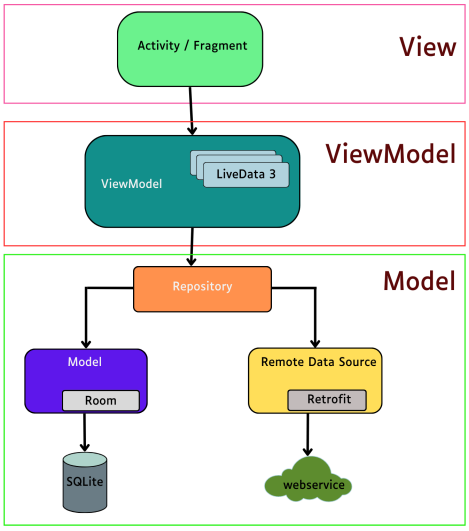

Figura 5 – Representação dos componentes da arquitetura MVVM. Fonte: (KUMAR, 2020).

2.2.6 API REST

De acordo com Belkhir et al. (2019), REST (Representational State Transfer) é de longe o estilo mais usado para projetar APIs (Application Programming Interface), especialmente para plataformas móveis, em razão da sua implementação ser leve e flexível. Uma API REST é um conjunto de padrões e diretrizes para comunicação entre cliente-servidor, permitindo o acesso e manipulação de recursos de forma padronizada.

Segundo Belqasmi et al. (2012), REST não restringe que a comunicação seja feita por nenhum protocolo em particular, mas o HTTP (Hypertext Transfer Protocol) é o mais comumente utilizado, por ser o principal protocolo de transferência da web. O modelo suporta 4 principais operações: *Create*, *Read*, *Update* and *Delete* (CRUD) as quais podem ser implementadas usando os seguintes métodos HTTP: POST, para fazer o envio de dados para o servidor; GET para solicitar dados de um servidor; PUT para atualizar informações existentes no servidor; e DELETE, para excluir algum recurso específico do servidor.

A integração com serviços REST pode ser realizada em várias linguagens, sendo comum a resposta dos dados em JSON (JavaScript Object Notation) ou XML (Extensible Markup Language).

No contexto do desenvolvimento Android do lado do cliente, Belkhir et al. (2019) apresenta algumas boas práticas, inclusive o uso do formato JSON como resposta da API, devido à sua legibilidade e eficiência em relação ao XML. Entre as práticas sugeridas, está o emprego de bibliotecas de terceiros para gerenciar as solicitações REST, como a Retrofit, que será utilizada no presente trabalho ou mesmo OkHttp, Google, Volley, a fim

de encapsular as consultas HTTP em interfaces bem definidas, garantindo uma abordagem organizada e eficiente no acesso aos recursos da API.

É possível observar um modelo de resposta no formato JSON na Figura 6 abaixo.

Figura 6 – Exemplo de uma resposta de API no formato JSON. Fonte: (CRACK, 2023).

3 Trabalhos Relacionados

Nesta seção são apresentados alguns trabalhos relacionados, que incluem não apenas estudos acadêmicos, mas também sites de venda B2C e aplicativos similares. Os sites de venda B2C *(Business to Consumer)* referem-se a plataformas que permitem a venda direta de produtos do produtor ou revendedor para o consumidor final. Esses sites foram identificados utilizando a ferramenta de busca do Google com palavras-chave como *produtos orgânicos Ouro Preto MG*. Já os aplicativos foram encontrados através de buscas na *Play Store* com termos como *orgânico*, *agricultura familiar* e *hortifruti*.

3.1 Greenpeace - Guia de produtores agroecológicos e orgânicos

Durante a pesquisa inicial, como base até mesmo de inspiração, foi consultado o site do Greenpeace Brasil (2024) em busca de problemas que poderiam ser solucionados com o auxílio de um aplicativo. Foi então encontrado o guia de produtores agroecológicos e orgânicos Greenpeace (2024), que oferece uma lista de produtores por estado brasileiro. A partir dessa descoberta, surgiu a ideia de criar algo semelhante em forma de aplicativo para o contexto da cidade de Ouro Preto.

3.2 Naturis

Nas primeiras fases de pesquisa, foi encontrada a Naturis (2024), uma empresa que vende produtos de mercearia, horta, pomar, além de itens para corpo e casa sustentáveis e orgânicos. A Naturis se apresentou como o primeiro mercado orgânico e sustentável de Ouro Preto e região, oferecendo uma plataforma segura para a compra de produtos certificados, com uma equipe comprometida em garantir a melhor seleção.

Embora essa seja uma excelente iniciativa para fomentar o mercado orgânico, foi decidido seguir com a ideia do aplicativo proposto neste trabalho, pois sua abordagem se difere um pouco. O aplicativo visa mapear os agricultores familiares e atuar como uma interface facilitadora de vendas, promovendo uma conexão direta entre produtores e consumidores. Em contraste, a plataforma da Naturis parece funcionar como uma espécie de revenda, sem proporcionar a mesma proximidade com os agricultores familiares.

3.3 Agromart

Os autores Rodrigues e Macêdo (2021) apresentaram em seu trabalho de conclusão de curso uma solução tecnológica que possui características bastante semelhantes à ideia

do presente trabalho.

A solução deles inclui uma interface web para os agricultores e um aplicativo mobile multiplataforma para os consumidores. O objetivo principal é proporcionar aos pequenos agricultores uma maior garantia de escoamento de sua produção, ao mesmo tempo em que ajuda consumidores a encontrar produtos saudáveis e de alta qualidade. Assim, o Agromart visa beneficiar tanto os agricultores, facilitando a venda de seus produtos, quanto os consumidores, que procuram por uma alimentação mais saudável.

3.4 Natural da terra

O aplicativo Natural da Terra é uma plataforma de compras online focada na venda de produtos hortifrúti frescos, oferecendo conveniência aos consumidores ao trazer a experiência das feiras para o ambiente digital (PLAY, 2024). Com uma avaliação de 4,1 na Play Store e mais de 10 mil *downloads*, o aplicativo atende à região da Grande São Paulo, valorizando a proximidade com os produtores e garantindo que os alimentos cheguem do campo à mesa dos consumidores de forma prática e eficiente. Além dos produtos hortifrúti, o aplicativo também oferece itens de mercearia, bebidas e higiene pessoal. Embora existam semelhanças com a proposta deste trabalho, a principal diferença está no fato de o **Natural da Terra** abranger uma variedade maior de categorias de produtos, enquanto a proposta aqui apresentada se concentra exclusivamente no segmento hortifrúti.

4 Desenvolvimento

Neste capítulo, são detalhadas as etapas do desenvolvimento do aplicativo. A primeira seção aborda as ferramentas empregadas, incluindo a linguagem de programação, softwares auxiliares e as principais bibliotecas utilizadas no desenvolvimento para Android.

A segunda seção apresenta a metodologia adotada, abrangendo a especificação de requisitos e regras de negócio, a prototipação, a criação da API e, por fim, a implementação do aplicativo.

4.1 Ferramentas Utilizadas

Esta seção detalha as principais tecnologias empregadas no desenvolvimento do aplicativo, incluindo a linguagem de programação Kotlin, bibliotecas de desenvolvimento e softwares de apoio.

Android Studio

Dentre as ferramentas utilizadas, o Android Studio, a IDE oficial para o desenvolvimento de aplicativos Android, se destaca como o ambiente central na construção deste projeto. Ele oferece um conjunto abrangente de funcionalidades que facilitam o desenvolvimento eficiente, incorporando um editor de código robusto e ferramentas do IntelliJ IDEA. Entre os recursos que aumentam a produtividade, incluem-se:

- Um sistema de *build* flexível baseado em Gradle;
- Um emulador rápido com inúmeros recursos;
- Um ambiente unificado que possibilita o desenvolvimento para todos os dispositivos Android;
- A Edição em tempo real para atualizar elementos combináveis em emuladores e dispositivos físicos;
- Modelos de código e integração com GitHub para ajudar a criar recursos comuns de apps e importar exemplos de código;
- *Frameworks* e ferramentas de teste;
- Ferramentas de lint para detectar problemas de desempenho, usabilidade, compatibilidade com versões, entre outros (DEVELOPERS, A., 2023a).

Figma

Figma é uma ferramenta de *design* e prototipagem, criada especialmente para facilitar colaborações, permitindo que *designer* e equipes trabalhem de forma síncrona e eficiente, com recursos de *design*, prototipagem interativa e colaboração em tempo real. Além disso, apresenta a vantagem de não ser necessário instalação, uma vez que é executado no próprio navegador (FIGMA, 2024).

Material Design

O *Material Design*, um sistema de design desenvolvido e constantemente atualizado pelos designers e desenvolvedores do Google, tornou-se um padrão na criação de interfaces de usuário (UI) para diversas plataformas, incluindo Android, Flutter e a Web. Ele oferece um conjunto de diretrizes de experiência do usuário (UX) e implementações de componentes de interface do usuário, disponíveis no material.io. Esses recursos pré-projetados e personalizáveis simplificam o desenvolvimento de interfaces responsivas e intuitivas, alinhadas com as expectativas dos usuários modernos (GOOGLE, 2018).

Mockable.io

O Mockable é uma ferramenta simples e rápida para criar serviços de *mock* para APIs RESTful ou web services SOAP. Uma das vantagens do Mockable é sua simplicidade e rapidez. Além disso, ele oferece a funcionalidade de organizar os mocks por domínio, criando contêineres personalizados que possibilitam a contribuição de outros usuários e até mesmo a criação de subdomínios conforme necessário (GROFFE, 2021).

Firebase

O Firebase é considerado uma plataforma de aplicativos da web, que possui diversas ferramentas para ajudar os desenvolvedores a criarem aplicativos de alta qualidade. Entre elas, podemos citar: *Cloud Firestore, Authentication, Crashlytics, Google Analytics, Remote Config e Cloud Messaging.*

O objetivo do **Firebase Authentication** é facilitar o desenvolvimento de um sistema de autenticação seguro, além de melhorar a experiência de login e integração para os usuários finais. Ele oferece uma solução de identidade completa, compatível com contas de e-mail/senha, autenticação por telefone, login do Google, Twitter, Facebook, GitHub e outros (DEVELOPERS, G. FOR, 2023).

Azure DevOps

O Azure DevOps, também conhecido como ADO, representa uma solução para o gerenciamento de projetos de *software* na plataforma Microsoft Azure. Construído com base nos princípios ágeis, proporciona às equipes multifuncionais a capacidade de monitorar e colaborar no desenvolvimento de aplicativos desde a fase conceitual até a entrega final.

Consistindo em cinco serviços, o Azure DevOps inclui o **Azure Boards** como um de seus componentes principais. Este ambiente é central para todas as atividades de gestão de projetos, onde equipes podem rastrear e organizar suas tarefas diárias, além de gerar relatórios personalizados para avaliar o progresso e identificar áreas de aprimoramento. O Azure Boards também oferece suporte nativo para metodologias ágeis como Scrum e Kanban.

Além do Azure Boards, o Azure DevOps incorpora outros serviços que abrangem desde o controle de versão de código-fonte até a automação de compilação e testes, proporcionando uma solução completa para o ciclo de vida de desenvolvimento de *software* na nuvem da Microsoft Azure. Trata-se, portanto, de uma ferramenta robusta e flexível, adaptável às necessidades de diversas equipes e projetos, facilitando o desenvolvimento e entrega de *software* de alto padrão (BHATTACHARYA, 2024).

Principais Bibliotecas

Na Tabela 1 encontram-se as principais bibliotecas usadas no desenvolvimento:

Biblioteca	**Descrição**
activity	Acessar APIs combináveis baseadas na Activity.
appcompat	Permite o acesso a novas APIs em versões anteriores de API da plataforma (muitas usando o Material Design).
compose	Define a IU de maneira programática com funções combináveis que descrevem a forma e as dependências de dados dela.
databinding	Vincula componentes de IU dos seus layouts a fontes de dados no seu app usando um formato declarativo.
fragment	Segmenta o app em várias telas independentes hospedadas em uma atividade.
lifecycle	Cria componentes com reconhecimento de ciclo de vida que podem ajustar o comportamento com base no estado atual do ciclo de vida de uma atividade ou um fragmento.
material design components	Componentes modulares e personalizáveis da IU do Material Design para Android.

navigation	Cria e estrutura a IU no app, para gerenciar links diretos e navegar entre telas.
picasso	Simplifica o carregamento de imagens a partir de URLs ou recursos locais.
test	Possibilita criação de testes no Android.
recyclerview	Usada para mostrar grandes conjuntos de dados na IU enquanto minimiza o uso de memória.
annotation	Expor metadados que ajudam as ferramentas e outros desenvolvedores a entender o código do app.
coroutines	Padrão de projeto de simultaneidade usado para simplificar o código que é executado de forma assíncrona.
kotlin-stdlib-jdk7	Essencial para usar a linguagem Kotlin
ktlint-gradle	Uma ferramenta de formatação de código Kotlin que mantém a consistência e legibilidade através de regras de estilo.
retrofit	Usada para consumir dados de serviços web, ela busca os dados na API e os transforma em objetos utilizáveis.
moshi-kotlin	Facilita transformar um JSON em classes Kotlin.
lottie-android	Processa animações do Adobe After Effects exportadas como JSON e as renderiza nativamente no celular.
koin-android	Framework para facilitar a injeção de dependência, tornando o código mais modular e testável.
maps	Fornece uma plataforma poderosa para incorporar recursos de mapas, localização e geolocalização.

Tabela 1 – Descrição das principais bibliotecas utilizadas

4.2 Metodologia

Para dar início à concepção do projeto, foi realizado um levantamento de requisitos, uma etapa crítica no desenvolvimento de *software*, pois a qualidade do produto final está diretamente vinculada à qualidade dos requisitos levantados. Esses requisitos são fundamentais para definir os objetivos e as funções que o *software* deve executar, bem como as restrições a serem ponderadas durante o seu desenvolvimento (FERGUSON; LAMI, 2006).

4.2.1 Requisitos Funcionais

Os requisitos funcionais descrevem as funções que o software deve ser capaz de executar de acordo com a necessidade do cliente e dos usuários. Esses requisitos estão listados na Tabela 2.

Código	**Descrição do Requisito**	**Prioridade**
RF01	Deve permitir que o usuário se cadastre. **Justificativa:** Permite personalização, controle de preferências e acesso a funções exclusivas.	Essencial
RF02	Deve permitir o uso de geolocalização. **Justificativa:** Possibilita que o aplicativo forneça resultados mais relevantes, como mostrar agricultores mais próximos à localização atual do usuário.	Essencial
RF03	Deve permitir que o usuário pesquise mercados. **Justificativa:** Facilita encontrar mercados e produtos, tornando a navegação mais eficiente.	Essencial
RF04	Deve apresentar a listagem de mercados, com seus respectivos produtos. **Justificativa:** Organiza informações para que o usuário veja facilmente os mercados e produtos disponíveis.	Essencial
RF05	Deve fornecer contato do mercado e direcionar para o whatsapp ou discador para fazer pedido. **Justificativa:** Facilita pedidos rápidos e diretos, conectando usuários e produtores.	Essencial
RF06	Deve ter a opção excluir conta. **Justificativa:** Garante controle e segurança sobre os dados do usuário.	Essencial
RF07	Deve ter a opção deslogar do aplicativo. **Justificativa:** Oferece segurança ao permitir que o usuário saia de sua conta em dispositivos compartilhados.	Essencial

Tabela 2 – Requisitos funcionais

4.2.2 Requisitos Não-Funcionais

Descreve as qualidades e restrições globais do sistema, relacionadas ao desempenho, segurança, escalabilidade, usabilidade, entre outros.

Usabilidade

1. Facilidade de aprendizagem.

2. Prevenção de erros, alertas ou modais de confirmação devem ser mostrados para notificar sobre ações com consequências significativas.
3. Recuperação de erros, os campos de entrada de dados devem oferecer *feedbacks* em caso de preenchimento errôneo.
4. Correspondência entre elementos do sistema e o mundo real, iconografia e palavras devem representar visualmente a sua função.

Confiabilidade

1. Não deve permitir *login* de pessoas sem cadastro.
2. A taxa de ocorrência de falhas deve se manter baixa.

Desempenho

1. Deverá consumir o mínimo de recursos possíveis.
2. Deverá possuir um baixo tempo de resposta.
3. Solicitações devem ser processadas em segundo plano, de forma a não travar a ação do usuário.

Portabilidade

1. O aplicativo não será compatível com versões de Android inferiores a 9.0.

Segurança

1. A aplicação deve verificar a identidade do usuário antes de permitir que ele tenha acesso a determinadas funcionalidades.
2. A aplicação não poderá permitir que usuários tenham acesso a dados pessoais como, endereço, email e senha dos demais usuários.
3. A aplicação deve manter a integridade de dados informados pelo usuário.
4. A aplicação deve garantir que os dados armazenados nela não sejam perdidos.
5. A aplicação deve detectar e registrar todas as tentativas de acesso que falharem nos requisitos de autenticação.

Acessibilidade

1. Todos os elementos exibidos na tela devem ter sua finalidade adequadamente descrita, de modo a ser apresentada quando o Talkback (leitor de tela) estiver ativado.

2. A aplicação deve manter sua legibilidade mesmo quando a opção de acessibilidade para tamanhos de fontes grandes for selecionada.

4.2.3 Regras de Negócio

Segundo Ceri, Fraternali et al. (1997), regras de negócio respondem às necessidades da aplicação, modelam a reação dos eventos que ocorrem no mundo real, com efeitos tangíveis no conteúdo da base de dados, assim como encapsula o comportamento reativo da aplicação para tais eventos.

A Tabela 3 apresenta essas regras de negócio.

Identificador	**Regras de Negócio**
RN01	Na tela inicial para usuário ainda não cadastrado existirá botão Cadastrar, ao clicar poderá digitar email e senha, no sucesso do cadastramento, a opção de cadastro não estará mais presente na Tela Inicial e o usuário terá acesso à funcionalidades do modo logado.
RN02	A tela inicial para o usuário convidado apresentará as opções Entrar e Cadastrar, caso o usuário tenha se cadastrado previamente ao clicar em Login poderá preencher suas informações e no sucesso do Login a opção de cadastro não estará mais presente na Tela Inicial e o usuário terá acesso à funcionalidades do modo logado.
RN03	O usuário não poderá concluir o processo de login ou cadastro enquanto não fornecer todos os dados obrigatórios corretamente, garantindo que apenas informações válidas sejam aceitas.
RN04	Ao entrar no aplicativo e acessar a aba 'Mercados' pela primeira vez, o usuário receberá um aviso sobre a utilização da localização e um pop up para permitir ou não a localização.
RN05	Caso o usuário tenha permitido o uso da localização, ao entrar na aba de pesquisa de mercados, serão apresentados aqueles mais próximos de acordo com a localização atual. Caso não, o usuário poderá realizar a busca no mapa, inserindo o endereço ou nome do mercado.
RN06	Caso o usuário tenha concedido a permissão de localização ao clicar na aba pesquisar, a listagem será ordenada de acordo com a proximidade da localização atual.

RN07	Ao acessar os detalhes de um mercado, será possível ligar para o mesmo, ou enviar uma mensagem pelo whatsapp para fazer um pedido.
RN08	Caso o usuário seja logado poderá visualizar suas informações pessoais, como: nome e foto de perfil.
RN09	Ao acessar a tela de detalhes de um produto, o usuário visualizará uma lista contendo o nome, preço e um selo que identifica se o produto é orgânico.
RN10	Caso o usuário esteja logado, deverá existir um botão para permitir o *logout*.
RN11	Caso o usuário tenha realizado o cadastro, a qualquer momento poderá optar por apagar a conta criada.

Tabela 3 – Regras de negócio

4.2.4 Casos de Uso

Após a definição dos requisitos e considerando as Regras de Negócio foram criados os Casos de Uso. Cockburn (2005) caracteriza um caso de uso como um "contrato de comportamento". Esse "contrato" que define a maneira através da qual um ator usa um sistema baseado em computadores para atingir alguma meta. Em essência, um caso de uso captura as interações que ocorrem entre produtores e consumidores de informação e o sistema em si.

A Figura 7 a seguir representa o diagrama de Caso de Uso Geral do aplicativo:

Casos de usos baseados em cenários

A Tabela 4 descreve o caso de uso para o cadastro de usuários

Caso de uso: Cadastro de usuário
Objetivo: Permitir que um usuário se cadastre no aplicativo pela primeira vez e possa utilizar algumas funcionalidades especiais. Ator: Usuário. Pré-condições: • Aplicativo instalado no dispositivo. • Não possuir cadastro. • Possuir um e-mail.
Cenário de sucesso principal: 1. O usuário abre o aplicativo. 2. O usuário clica no botão Cadastrar.

3. O usuário preenche os campos de nome, email e senha.
4. O usuário clica em Cadastrar e o cadastro é finalizado.
5. O usuário é direcionado para dentro do aplicativo com acesso à aba Conta, podendo visualizar suas informações pessoais.

Exceções:
4a. Falha de conexão com a internet.
1. Uma mensagem de falha na conexão é exibida ao usuário juntamente com uma opção clicável de Tentar Novamente.
2. O usuário clica em Tentar Novamente e a aplicação tenta enviar os dados novamente.
4b. Usuário preenche informações inválidas.
1. Caso o usuário preencha erroneamente um ou mais campos obrigatórios, ao clicar no botão Confirmar, o campo inválido ganhará foco e aparecerá um aviso de erro no mesmo.
2. Isso se repetirá e o usuário não conseguirá entrar até adicionar campos válidos.

Tabela 4 – Caso de uso: Cadastro de usuário

A Tabela 5 apresenta o caso de uso para o login de usuários previamente cadastrados.

Caso de uso: Fazer login

Objetivo: Permitir que um usuário previamente cadastrado, faça a autenticação no aplicativo e utilize algumas funcionalidades especiais.
Ator: Usuário.
Pré-condições:
• Aplicativo instalado no dispositivo.

Cenário de sucesso principal:
1. O usuário abre o aplicativo.
2. O usuário clica no botão Cadastrar.
3. O usuário preenche os campos com suas informações pessoais.
4. O usuário clica no botão Entrar e o login é finalizado.
5. O usuário é direcionado para dentro do aplicativo com acesso à aba Conta, podendo visualizar suas informações pessoais.

Exceções:
4a. Falha de conexão com a internet.
1. Uma mensagem de falha na conexão é exibida ao usuário juntamente com uma opção clicável de Tentar Novamente.
2. O usuário clica em Tentar Novamente e a aplicação tenta enviar os dados novamente.
4b. Usuário preenche informações inválidas.

1. Caso o usuário preencha erroneamente um ou mais campos obrigatórios, ao clicar no botão Confirmar, o campo inválido ganhará foco e aparecerá um aviso de erro no mesmo.
2. Isso se repetirá e o usuário não conseguirá entrar até adicionar campos válidos.
4c. O usuário não possui cadastro.
1. É exibida uma mensagem informando que o usuário não possui nenhum cadastro com as informações preenchidas.
2. Ao clicar em Ok o usuário é direcionado para a tela de cadastro.
3. Ao clicar em Cancelar o usuário permanece na tela de *login* e pode conferir as informações digitadas.

Tabela 5 – Caso de uso: Fazer login

A Tabela 6 descreve o caso de uso para a pesquisa de mercados no aplicativo.

Caso de uso: Pesquisar mercado
Objetivo: Permitir que um usuário encontre o mercado no qual deseja fazer a compra. Ator: Usuário. Pré-condições: • Aplicativo instalado no dispositivo. • Possuir conexão à internet.
Cenário de sucesso principal: 1. O usuário abre o aplicativo. 2. O usuário clica na aba pesquisar e um teclado abre. 3. O usuário digita nome do mercado/endereço/cidade e clica em confirmar. 4. O usuário visualiza a lista de mercados correspondente à pesquisa.
Exceções: 4a. Falha de conexão com a internet. 1. Uma mensagem de falha na conexão é exibida ao usuário juntamente com uma opção clicável de Tentar Novamente. 2. O usuário clica em Tentar Novamente e a aplicação tenta enviar os dados novamente. 4b. Pesquisa não encontra nenhum resultado. 1. É exibida uma mensagem informando que nenhum item corresponde à pesquisa.

Tabela 6 – Caso de uso: Pesquisar mercado

A Tabela 7 apresenta o caso de uso para a visualização de mercados próximos à localização do usuário.

Caso de uso: Visualizar mercados próximos

Objetivo: Permitir que um usuário visualize os mercados mais próximos à sua localização. Ator: Usuário. Pré-condições: • Aplicativo instalado no dispositivo. • Possuir conexão à internet. • Permissão de localização concedida.
Cenário de sucesso principal: 1. O usuário abre o aplicativo. 2. O usuário clica na aba pesquisar e a lista de mercados próximos é mostrada.
Exceções: 4a. Falha de conexão com a internet. 1. Uma mensagem de falha na conexão é exibida ao usuário juntamente com uma opção clicável de Tentar Novamente. 2. O usuário clica em Tentar Novamente e a aplicação tenta enviar os dados novamente. 4b. Não existem mercados próximos. 1. É exibida uma mensagem informando que não foi encontrado nenhum mercado próximo à localização do usuário, mas ele pode pesquisar uma localização. 2. Quando o usuário clica em Ok, o campo de busca ganho foco e um teclado é mostrado, permitindo que o usuário faça a pesquisa.

Tabela 7 – Caso de uso: Visualizar mercados próximos

A Tabela 8 detalha o caso de uso para fazer um pedido a um mercado via ligação ou WhatsApp.

Caso de uso: Fazer um pedido
Objetivo: Permitir que o usuário entre em contato com o mercado para fazer um pedido. Ator: Usuário. Pré-condições: • Aplicativo instalado no dispositivo. • Possuir conexão à internet. • Aplicativo Whatsapp instalado (cenário via Whatsapp).
Cenário de sucesso via ligação: 1. O usuário abre o aplicativo. 2. O usuário clica na aba pesquisar e visualiza a lista de mercados. 3. O usuário escolhe um mercado e clica, visualizando uma tela com os detalhes do mercado.

4. Ao clicar no botão ligar, a tela do discador nativo é aberta com o número do estabelecimento preenchido, facilitando a ligação do usuário.
Cenário de sucesso via whatsapp: 1. O usuário abre o aplicativo. 2. O usuário clica na aba pesquisar e visualiza a lista de mercados. 3. O usuário escolhe um mercado e clica, visualizando uma tela com os detalhes do mercado. 4. Ao clicar no botão Whatsapp, o aplicativo whatsapp é aberto em uma conversa com contato do mercado.
Exceções: 2a. Falha de conexão com a internet. 1. Uma mensagem de falha na conexão é exibida ao usuário juntamente com uma opção clicável de Tentar Novamente. 2. O usuário clica em Tentar Novamente e a aplicação tenta enviar os dados novamente.

Tabela 8 – Caso de uso: Fazer um pedido

A Tabela 9 descreve o caso de uso para deslogar do aplicativo.

Caso de uso: Deslogar
Objetivo: Permitir que uma vez logado, o usuário deslogue da aplicação. Ator: Usuário. Pré-condições: • Aplicativo instalado no dispositivo. • Possuir conexão à internet. • Usuário estar logado no aplicativo.
Cenário de sucesso principal: 1. O usuário abre o aplicativo. 2. O usuário clica na aba perfil, e em seguida no botão Sair da Conta. 3. O usuário recebe uma mensagem de confirmação da ação de deslogar. 4.Caso o usuário clique em Confirmar, ele será deslogado, e redirecionado para a página de perfil para convidados. A qual informa que ele precisa estar logado para ter acesso às informações.
Exceções: 4a. Falha de conexão com a internet 1. Caso o usuário preencha erroneamente um ou mais campos obrigatórios, ao clicar no botão Salvar Informações, o campo inválido ganhará foco e aparecerá um aviso de erro no mesmo. 2. Isso se repetirá e o usuário não conseguirá salvar as informações até adicionar campos válidos.

Tabela 9 – Caso de uso: Deslogar

A Tabela 10 apresenta o caso de uso para deletar a conta do aplicativo.

Caso de uso: Deletar conta
Objetivo: Permitir, uma vez que o usuário tenha se cadastrado, ele apague sua conta de forma a desvincular seus dados do aplicativo. Ator: Usuário. Pré-condições: • Aplicativo instalado no dispositivo. • Possuir conexão à internet. • Usuário estar logado no aplicativo.
Cenário de sucesso principal: 1. O usuário abre o aplicativo. 2. O usuário clica na aba perfil, e em seguida no botão Deletar conta. 3. O usuário recebe uma mensagem de confirmação da ação de Deletar. 4.Caso o usuário clique em Confirmar, seus dados serão excluídos da base de dados do aplicativo, logo ele será deslogado, e redirecionado para a página de perfil para convidados.
Exceções: 4a. Falha de conexão com a internet. 1. Caso o usuário preencha erroneamente um ou mais campos obrigatórios, ao clicar no botão Salvar Informações, o campo inválido ganhará foco e aparecerá um aviso de erro no mesmo. 2. Isso se repetirá e o usuário não conseguirá salvar as informações até adicionar campos válidos.

Tabela 10 – Caso de uso: Deletar conta

4.2.5 Prototipagem e Design

Após a definição de todos os cenários e requisitos, o próximo passo crucial foi a prototipagem da aplicação. De acordo com Oliveira et al. (2007), protótipo é uma representação de uma ideia de um produto ou projeto e sua principal vantagem está na possibilidade de testar ideias, explorar hipóteses antes de investir tempo e recursos na sua efetiva implementação. Existe uma ordem a ser seguida a fim de facilitar a criação de um protótipo, e dessa maneira surge uma classificação para os seus tipos.

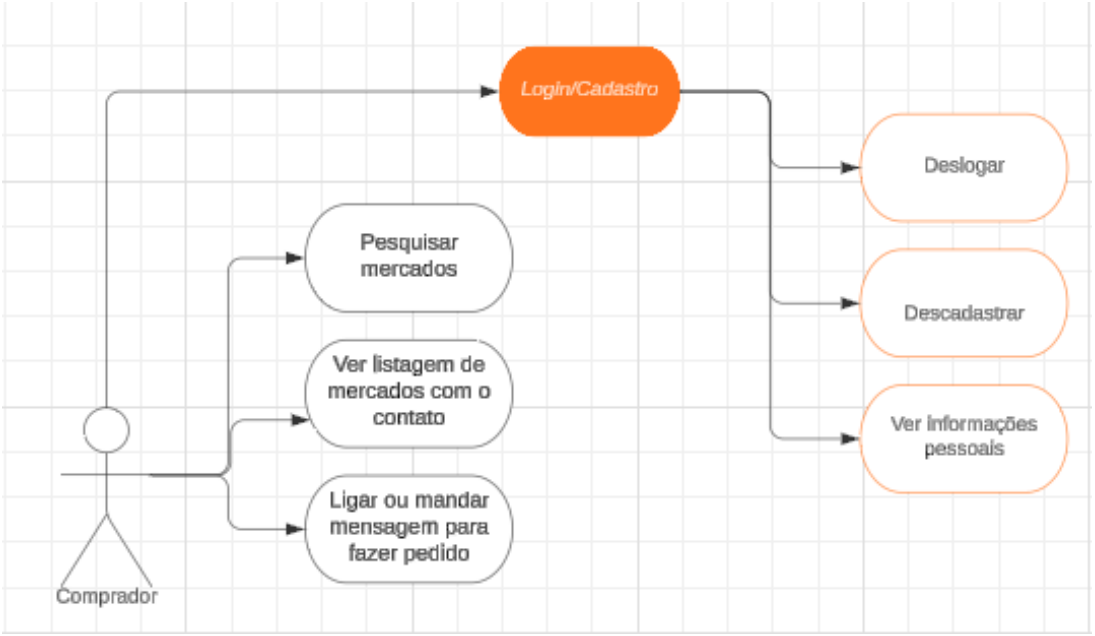

Figura 7 – Diagrama de Caso de Uso geral.

Protótipo de baixa fidelidade

O primeiro passo foi a criação do protótipo de baixa fidelidade. Este tipo de protótipo consiste em esboços gráficos primitivos da aplicação em desenvolvimento, geralmente utilizam técnicas de desenho à mão-livre utilizando apenas lápis e papel, com o propósito de expressar uma ideia sem grande investimento de tempo e esforço (OLIVEIRA et al., 2007).

Protótipo de média fidelidade

Com base nos insights obtidos a partir do protótipo de baixa fidelidade, criou-se o protótipo de média fidelidade, utilizando a ferramenta Figma. Esses protótipos geralmente consistem em figuras geométricas simples, sem uso de cores ou elementos de identidade visual complexos. Assim, nesta fase, foram implementados elementos essenciais que definiriam a estrutura da aplicação.

A Figura 8 apresenta as telas iniciais, onde o usuário pode escolher entre cadastrar ou logar.

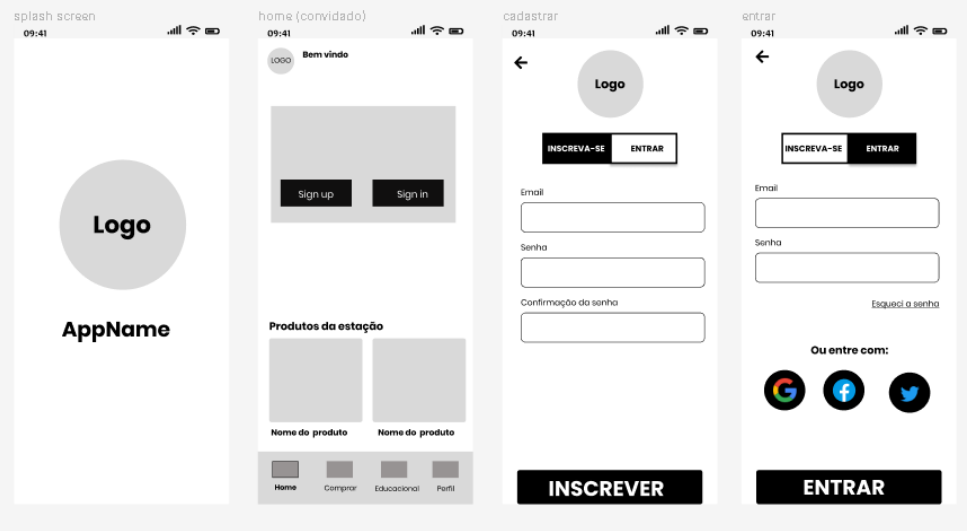

Figura 8 – Protótipo média fidelidade: tela Inicial, home (convidado), cadastrar e logar.

Já a Figura 9 mostra a tela de home para usuários logados, além das telas de pesquisa e detalhes do mercado.

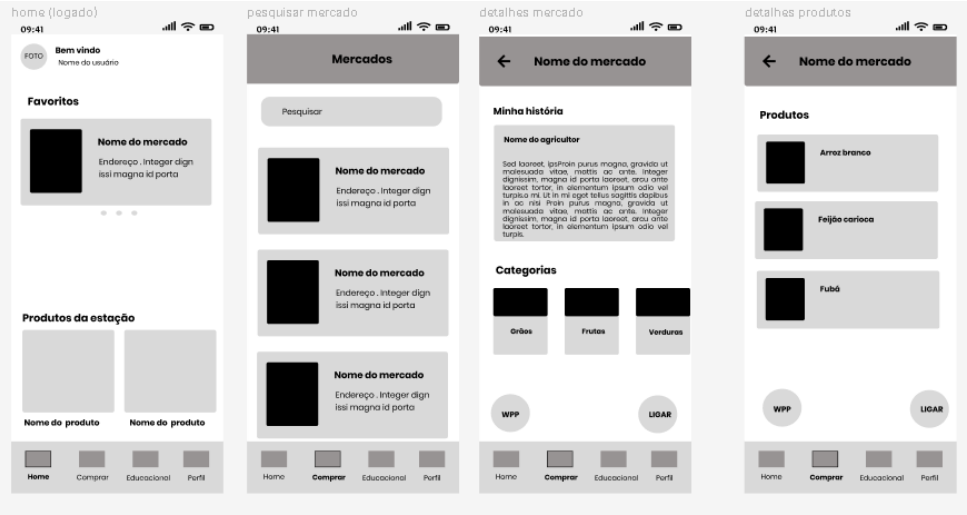

Figura 9 – Protótipo média fidelidade: tela home (logado), pesquisar, detalhes do mercado e detalhes do produto.

Nas telas educacionais, a Figura 10 ilustra tanto a visualização para usuários convidados quanto para usuários logados, além das telas de criação de publicação e visualização de perfil.

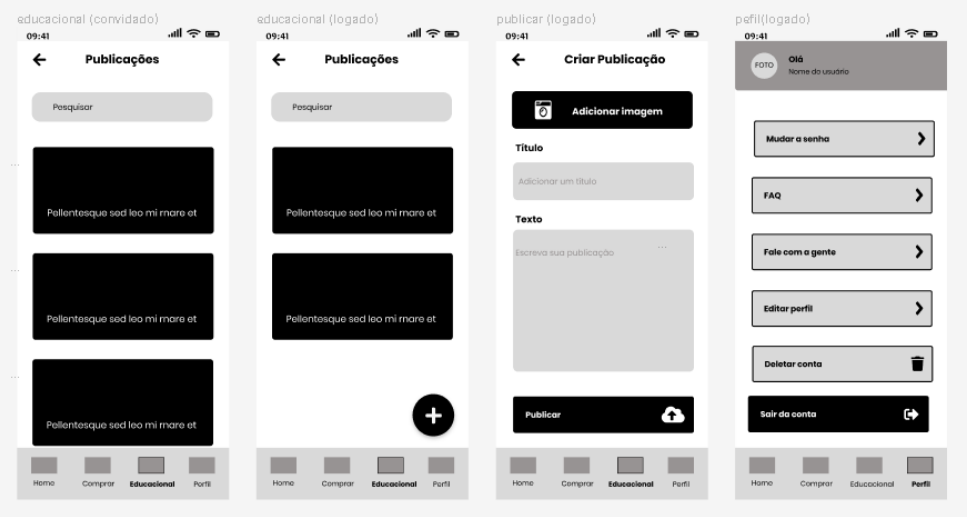

Figura 10 – Protótipo média fidelidade: tela educacional (convidado), educacional (logado), criar publicação e perfil.

Por fim, a Figura 11 mostra as telas de edição de perfil e a visualização do perfil para usuários convidados.

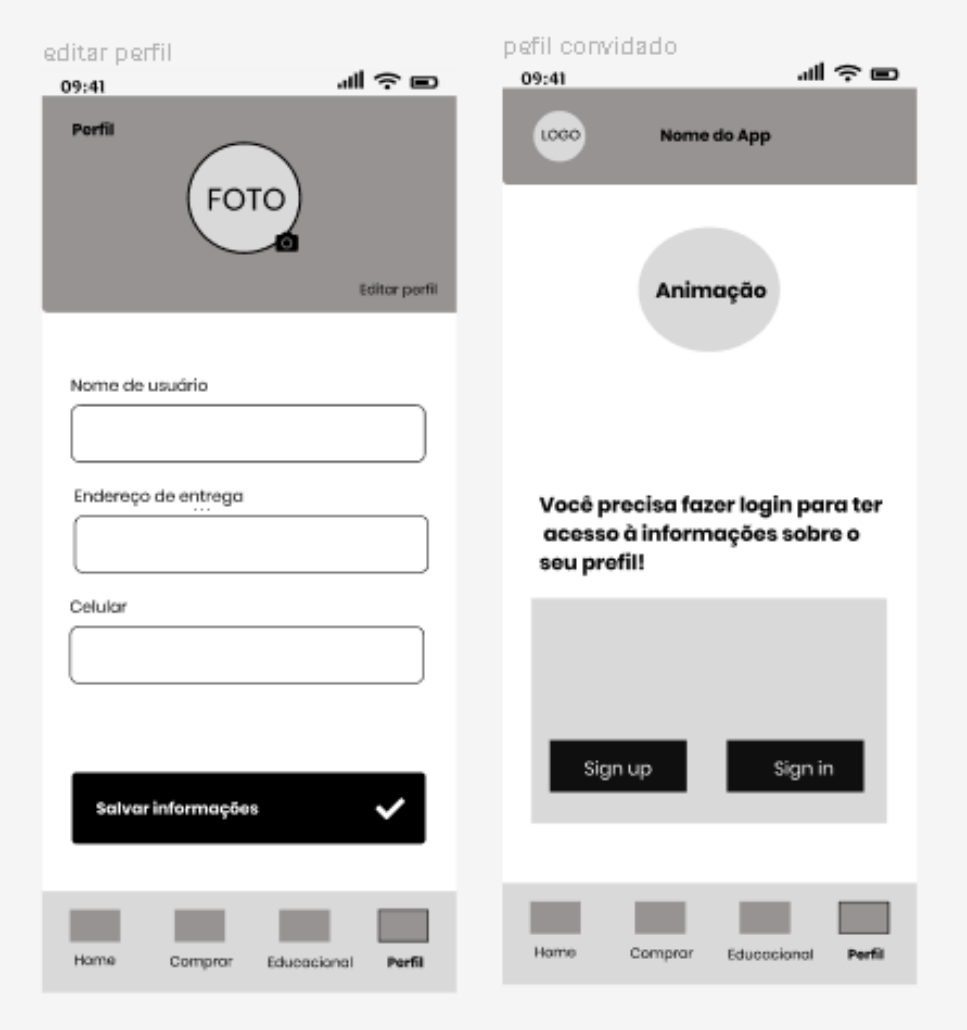

Figura 11 – Protótipo média fidelidade: tela editar perfil e perfil (convidado)

Componentes do Material Design no Aplicativo

Para garantir a conformidade com as melhores práticas de design e proporcionar uma experiência coesa aos usuários, foram utilizados os componentes fornecidos pelo Material Design no aplicativo. Além disso, foram estabelecidos os parâmetros de design para garantir consistência e harmonia em toda a aplicação:

Grade de Layout: Optou-se por utilizar grids de 6 colunas, com espaçamentos de 24px no lado esquerdo e direito, proporcionando uma estrutura organizada e visualmente agradável.

Espaçamento dos Componentes: Definiu-se um espaçamento padrão de 16px ou 24px para ser aplicado uniformemente em todos os componentes do aplicativo, incluindo margens superior, inferior, esquerda e direita, garantindo uma aparência coesa e bem dimensionada.

Ícones: Foi utilizado o plugin Iconify disponível no Figma para obter uma ampla

variedade de ícones gratuitos, garantindo uma representação visual eficaz e consistente em todo o aplicativo.

Tamanho da Fonte: O tamanho da fonte foi definido utilizando o gerador de escala de tipo do Google Fonts, garantindo uma hierarquia visualmente agradável e legível.

Cores: As diretrizes do Material Design foram seguidas.

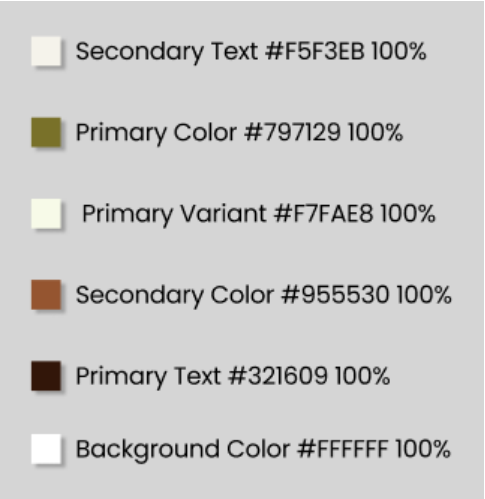

Figura 12 – TCores

A cor primária é aquela a ser mostrada mais frequentemente nas telas, por esse motivo a sua escolha deve remeter ao contexto do aplicativo. Desse modo, o tom de verde escolhido se associa à ideia de saúde assegurada pelos produtos livres de agrotóxicos, evocando a natureza e a vitalidade.

A cor variante primária é uma variação da cor primária, sugerida pela paleta de cores do *Material Design*, e é utilizada para destacar elementos visuais como cards e folhas, garantindo uma aparência coesa e distintiva.

Já a cor secundária, em tons terrosos associados à ideia de cultivo, é reservada para botões e elementos que precisam ser destacados na interface do aplicativo, proporcionando contraste e foco adequados.

A cor de fundo, predominantemente branca, é utilizada em todo o conteúdo rolável, proporcionando uma base neutra e limpa para o restante do design, não representando nenhum aspecto da marca.

Para a tipografia, como cor primária para a tipografia, optamos por um tom de marrom, simbolizando a terra e evocando a natureza e a agricultura. Essa escolha reflete a identidade do aplicativo, reforçando sua conexão com o mundo natural e orgânico. E como cor secundária, foi definido uma variação do branco garantindo uma legibilidade ideal e uma aparência visualmente agradável em todo o conteúdo textual do aplicativo.

Protótipo de alta fidelidade

Por fim, o protótipo de alta fidelidade foi elaborado. Utilizando como referência tanto a estrutura delineada no protótipo de média fidelidade, quanto as definições estabelecidas pelo design system, foram selecionados os aspectos estéticos, como paleta de cores, ícones, tipografia e dimensionamento dos elementos. Estes protótipos não apenas orientaram, mas também serviram de referência direta durante todo o processo de construção do produto final.

A Figura 13 apresenta a tela de início (Splash screen), a tela Home para usuário não logado e também as telas de cadastro e login.

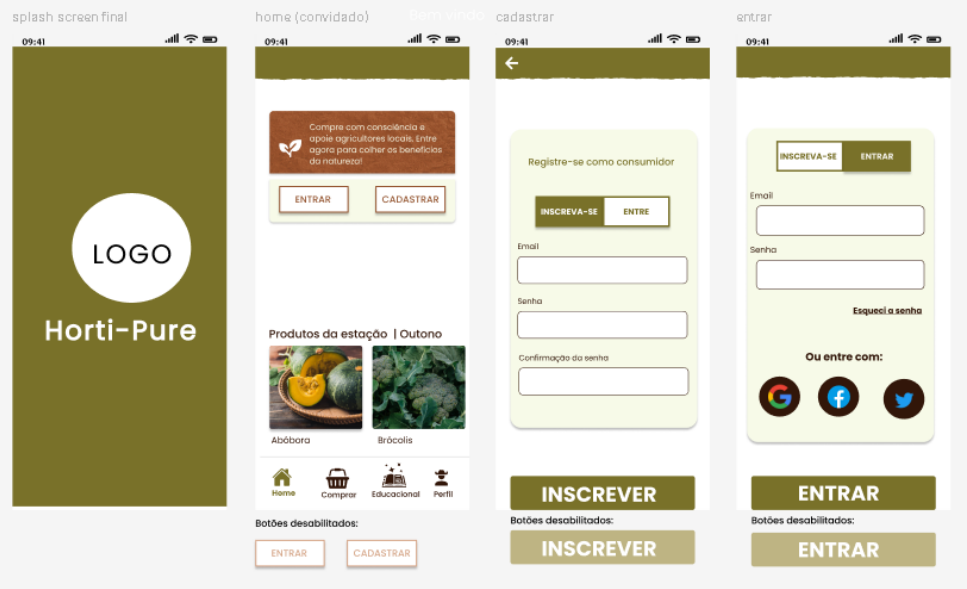

Figura 13 – Protótipo alta fidelidade: tela Inicial, home (convidado), cadastrar e logar.

Já a Figura 14 mostra a tela de home para usuários logados, além das telas de pesquisa e resultados da busca.

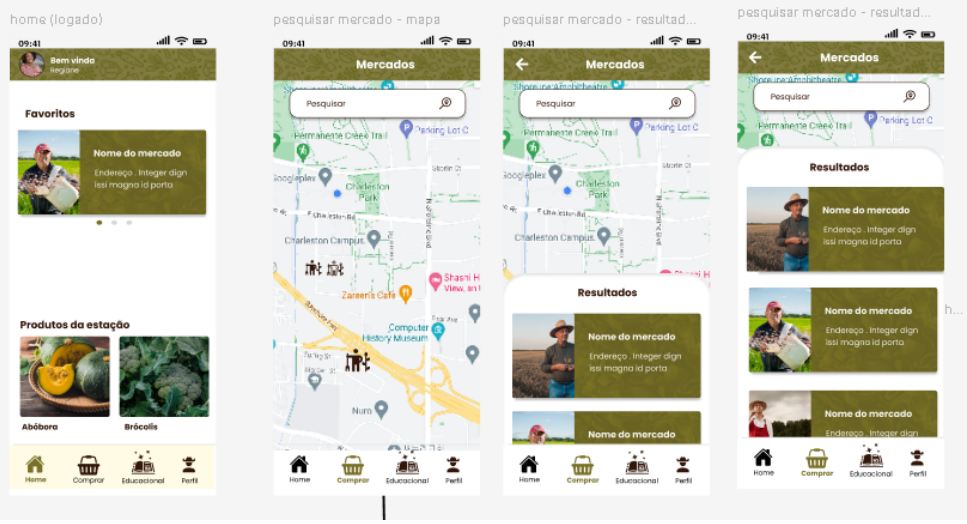

Figura 14 – Protótipo alta fidelidade: tela home (logado), pesquisar e resultados da busca.

A Figura 15 apresenta a tela de detalhes do mercado, detalhes dos produtos e também a tela da aba educacional, que não chegou a ser implementada.

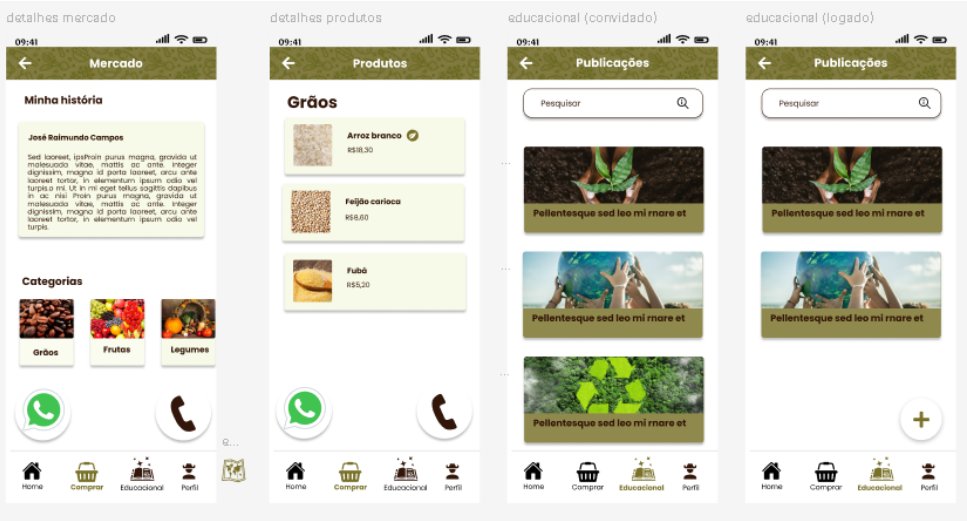

Figura 15 – Protótipo alta fidelidade: telas de detalhes do mercado, detalhes dos produtos e educacional.

E por fim, a Figura 16 apresenta a tela de perfil do usuário logado e convidado.

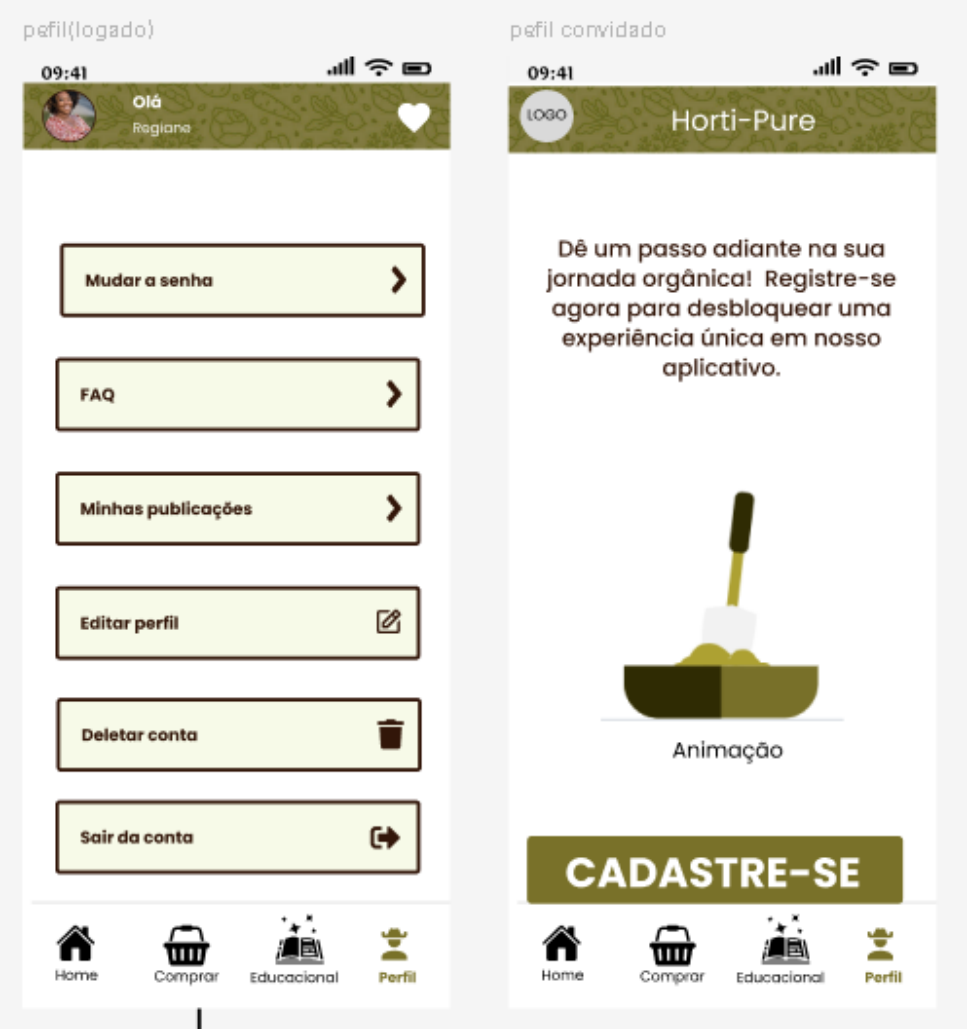

Figura 16 – Protótipo alta fidelidade: tela de perfil do usuário logado e convidado.

4.2.6 Implementação

Para garantir uma gestão eficaz do projeto e facilitar a colaboração, o Azure DevOps foi utilizado como nossa plataforma principal. Além de oferecer hospedagem de repositório na nuvem, o Azure DevOps oferece uma gama de ferramentas que facilitam o gerenciamento de projetos.

Optou-se por seguir com o *framework* de desenvolvimento ágil SCRUM para a fase de desenvolvimento do aplicativo. No entanto, o processo foi adaptado para se adequar a uma equipe de apenas um integrante, seguindo o conceito conhecido como *Scrum Solo*. Isso tornou possível manter uma abordagem ágil e interativa, mesmo em um ambiente de desenvolvimento individual.

Para isso, o projeto *horti-pure-app* foi criado na plataforma Azure e configurado no modo SCRUM. Isso permitiu a organização do trabalho em *sprints* com uma duração média de 2 semanas. Dentro de cada *sprint*, os *Product Backlog Items* (PIBs) foram definidos. Eles

representam as funcionalidades ou melhorias a serem implementadas. Cada PIB foi decomposto em tarefas menores (*tasks*), que foram rastreadas e gerenciadas no *board* do Azure DevOps.

4.2.6.1 Backend

O *backend* lida com a lógica de negócio do sistema, definição e funcionamento da API e suas funcionalidades, formas de comunicação entre esta API e outros dispositivos, etc. Ao realizar a separação, torna-se possível o desenvolvimento em paralelo, otimizando o avanço e amenizando a dependência entre ambas as partes.

Por não ter sido possível reunir os dados dos possíveis agricultores da região em razão da falta de acesso a essa informação, para a construção da API a ser utilizada no aplicativo, optou-se por utilizar a técnica chamada *API mocking*. Isso é basicamente a criação de uma versão simulada de como seria a API verdadeira. Dessa maneira, o mock da API apenas imita como seria seu comportamento, mas não processa nenhum dado ou comunica com sistemas externos.

Para a criação dos arquivos JSON que seriam a resposta retornada pela API, utilizou-se a ferramenta Mockaroo. Já para simular o comportamento do *backend service*, foi criado o seu *mock* através da ferramenta online Mockable.io. Primeiramente, ao criar uma conta no site ele criará um novo domínio para o serviço, como pode ser observado na figura 17, abaixo:

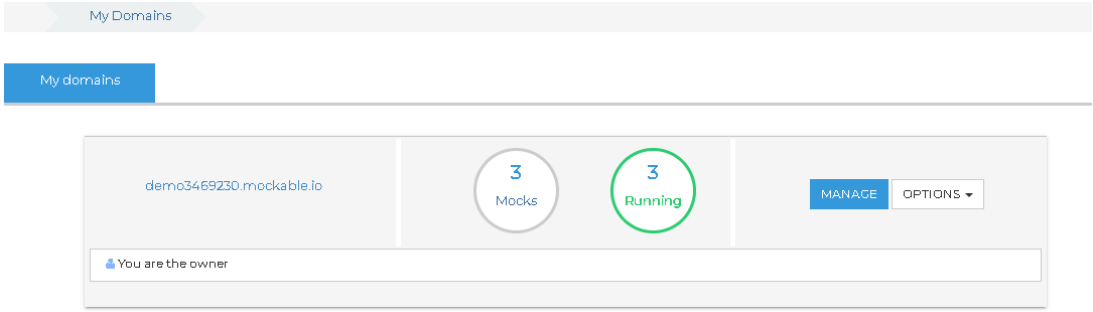

Figura 17 – Domínios

A partir disso é possível criar os *mocks* desejados. Para o trabalho desenvolvido até o momento, foram utilizadas apenas requisições GET. É preciso realizar a configuração do mock, fornecendo algumas informações como: *path, verb, headers, response status, content-type, response body* (a resposta JSON que deve ser retornada com a requisição), *name and description.*

Após configurados, é preciso clicar na opção *Start*, então os *mocks* estarão disponíveis e prontos para serem utilizados na aplicação. As requisições utilizadas na aplicação foram:

- GET /products - retorna os produtos da estação atual.

- GET /groceries - retorna os mercados próximos à localização.
- GET /menu - retorna todos os produtos e suas respectivas categorias disponíveis em um mercado.

4.2.6.2 Frontend

A camada conhecida como *frontend* de uma aplicação é onde toda a informação obtida através do *backend* pode ser apresentada, onde ocorrem as interações com o usuário. Este trabalho tem como foco principal essa camada. Portanto, para o seu desenvolvimento optou-se pela utilização da linguagem de programação Kotlin, por ser a linguagem oficial de desenvolvimento de aplicativos Android, determinada pela Google em 2018. Pelo mesmo motivo, a IDE (Integrated Development Environment) utilizada é o Android Studio, além de que ela integra diversas ferramentas vitais para o desenvolvimento, como por exemplo um emulador.

Além disso, foi adotada a arquitetura MVVM (Model-View-ViewModel), cuja explicação detalhada sobre as vantagens e o funcionamento foi apresentada no capítulo de Revisão Teórica (ver Seção 2.2.5). A escolha do MVVM visa garantir uma separação clara entre as responsabilidades dos componentes, melhorando a escalabilidade, testabilidade e manutenção do código.

Quanto à modularização, o aplicativo foi dividido entre as camadas *app* e *data*, essa abordagem oferece uma série de benefícios em termos de organização do código, reutilização e manutenção:

Módulo app: Esta camada contém todos os componentes relacionados à apresentação da aplicação, como *Activities*, *Fragments*, *ViewModels* e recursos de UI. Ao modularizar dessa forma, a lógica de apresentação é isolada do restante da aplicação, o que facilita a manutenção e permite que diferentes partes da equipe de desenvolvimento trabalhem de forma independente em diferentes funcionalidades.

Módulo data: Esta camada é responsável por lidar com a obtenção e manipulação de dados. Ela inclui componentes como repositórios, fontes de dados e serviços de rede. Modularizar essa camada permite a reutilização da lógica de acesso a dados em diferentes partes da aplicação, facilita a substituição de implementações específicas (por exemplo, mudar de uma API para outra) e simplifica a escrita de testes unitários.

4.2.6.3 Organização dos Ciclos de Trabalho

Através das imagens abaixo, é possível visualizar o fluxo de trabalho no board do Azure, evidenciando o progresso das tarefas ao longo das *sprints*.

Sprint 1 (22 de outubro à 4 de novembro) - Figura 18

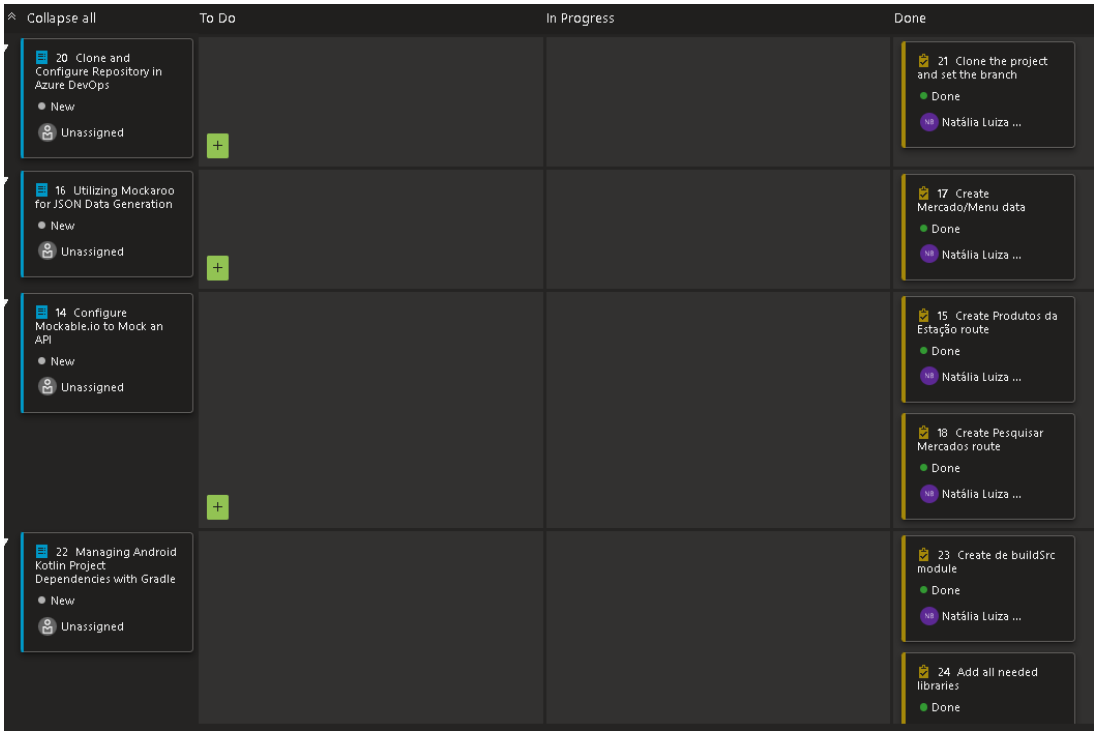

Figura 18 – Azure Board - Sprint 1

Após a criação do projeto no repositório, o mesmo foi clonado localmente para a máquina, utilizando o Android Studio como ambiente de desenvolvimento. Em seguida, foram elaboradas as respostas JSON esperadas pela API, juntamente com o *mock* de seus *endpoints*. Posteriormente, as configurações de dependências foram adicionadas ao aplicativo, elas são os componentes externos necessários para que o aplicativo funcione corretamente. Esses componentes podem incluir bibliotecas, *frameworks*, ferramentas ou qualquer outro *software* que o aplicativo precise para realizar suas funcionalidades.

Sprint 2 (5 de novembro à 19 de novembro) - Figura 19

Posteriormente, avançou-se para a criação da *splash screen* do aplicativo, que é a tela inicial exibida enquanto os recursos do aplicativo são carregados. Além disso, configurou-se o módulo Koin, uma biblioteca de injeção de dependências otimizada para Kotlin, que facilita o gerenciamento de dependências de forma eficiente e desacoplada (conforme mencionado na Seção 4.1). Assim, criou-se o nav_graph, que é responsável pela definição da navegação entre as telas do aplicativo, permitindo que os usuários se movam de forma intuitiva de uma tela para outra. Também desenvolveu-se o *bottom menu*, que oferece uma navegação fácil e acessível entre as principais seções do aplicativo. Além disso, a *top app bar* foi implementada, fornecendo acesso rápido a funções importantes, permitindo uma experiência de usuário mais integrada e eficiente.

Sprint 3 (5 de dezembro à 17 de novembro) - Figura 20

Após a configuração do módulo *data*, foi feita a adição das permissões necessárias ao

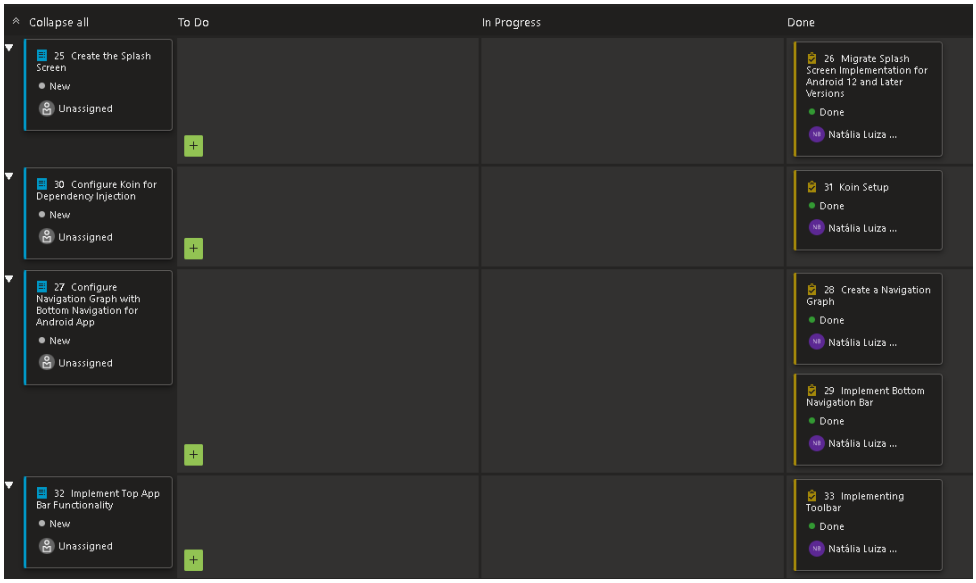

Figura 19 – Azure Board - Sprint 2

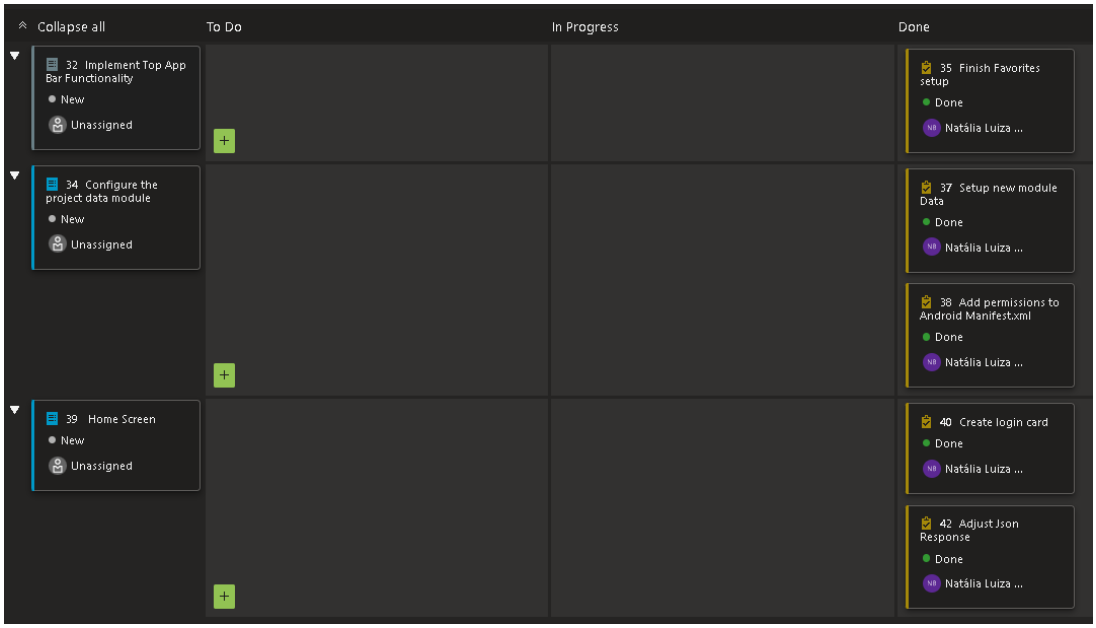

Figura 20 – Azure Board - Sprint 3

aplicativo. Essas permissões representam um sistema de segurança fundamental que regula o acesso do aplicativo a partes específicas do sistema operacional, recursos do dispositivo e dados do usuário. Elas são explicitamente declaradas no arquivo AndroidManifest.xml do aplicativo, comunicando ao sistema quais recursos ou funcionalidades específicas o aplicativo requer durante sua execução. Posteriormente, o desenvolvimento da tela inicial *Home* foi iniciado. Nesta etapa, um carrossel que destaca os produtos da estação foi criado, oferecendo uma experiência visual dinâmica aos

usuários. Além disso, um *card* com opções para *login* ou cadastro no aplicativo foi implementado, fornecendo aos usuários uma maneira conveniente de iniciar ou criar uma conta para acessar os recursos adicionais do aplicativo.

Sprint 4 (18 de dezembro à 1 de janeiro) - Figuras 21 e 22

Figura 21 – Azure Board - Sprint 4, primeira parte

Figura 22 – Azure Board - Sprint 4, segunda parte

Após a conclusão da criação da tela inicial Home, avançou-se para uma etapa crucial: a integração com o Google Maps. Essa integração possibilita uma experiência ainda mais enriquecedora para os usuários, com recursos de geolocalização permitindo uma busca e visualização intuitivas da localização através do mapa.

Sprint 5 (1 de janeiro à 15 de janeiro) - Figuras 23 e 24

Dessa forma, criou-se uma *Bottom Sheet* personalizada para exibir os resultados das buscas realizadas pelo usuário em forma de listas. Em seguida, foi implementada uma tela dedicada que apresenta informações detalhadas sobre os mercados e os produtos disponíveis em cada um deles, oferecendo uma experiência informativa e intuitiva aos usuários.

Figura 23 – Azure Board - Sprint 5, primeira parte

Figura 24 – Azure Board - Sprint 5, segunda parte

Sprint 6 (16 de março à 30 de março) - Figura 25

Aproximando-se dos estágios finais de desenvolvimento da solução proposta, a 6ª sprint teve como foco principal o desenvolvimento da autenticação no aplicativo, utilizando a ferramenta Firebase Authenticator mencionada anteriormente. Essa etapa é crucial para facilitar o gerenciamento da conta do usuário e permitir a realização de ações como exclusão da conta ou logout. Paralelamente, deu-se início à criação da tela de perfil do usuário, visando proporcionar uma experiência personalizada e funcional aos usuários do aplicativo. Além disso, buscou-se fazer a correção da resposta JSON da API. Inicialmente criada em inglês, com endereços fictícios dos Estados Unidos, ajustamos esse aspecto para garantir resultados mais condizentes com a realidade.

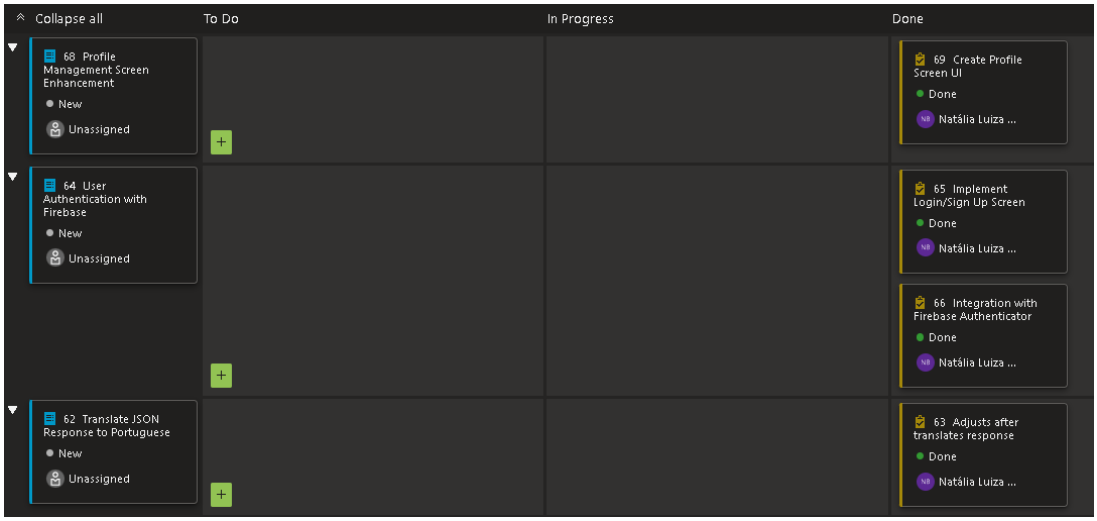

Figura 25 – Azure Board - Sprint 6

Sprint 7 (31 de março à 14 de abril) - Figura 26

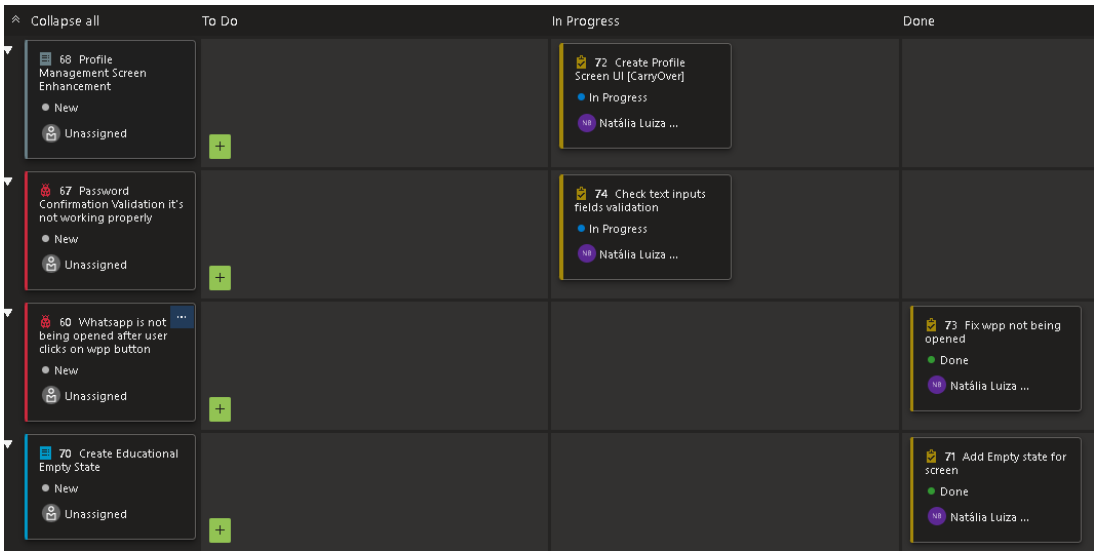

Figura 26 – Azure Board - Sprint 7

E para concluir os ajustes finais na sprint 7, dedicada como a etapa final do projeto, finalizou-se o processo de criação da tela de perfil do usuário. Também foi implementado um estado inicial para a aba educacional, informando aos usuários que essa funcionalidade será desenvolvida no futuro. Além disso, reservou-se um tempo para a resolução de alguns bugs que foram identificados ao longo do processo de desenvolvimento. Por exemplo, o botão que direciona o usuário para um chat no WhatsApp com o vendedor que estava com funcionamento pendente, foi corrigido e agora está operando corretamente.

5 Resultados

Nesta seção, serão apresentadas as telas implementadas no aplicativo, em conformidade com o protótipo previamente demonstrado no capítulo 4.

Ao ingressar no aplicativo pela primeira vez, os usuários são recebidos pela *splash screen*, que é exibida durante o processo de carregamento da aplicação. Optamos por seguir as melhores práticas na criação dessa tela, priorizando a simplicidade e o destaque para a identidade visual da marca. Com poucas cores utilizadas de forma estratégica, a *splash screen* apresenta um logo simples, mas significativo, que reflete o propósito do aplicativo de forma clara e eficaz, como pode ser observado na Figura 27.

Figura 27 – Splash Screen

Após a *splash screen*, o usuário é encaminhado para a tela inicial do aplicativo Hortipure.

Lá, ele encontrará quatro abas principais: Home, Comprar, Educacional e Conta.

Na tela Home, caso o usuário ainda não tenha efetuado o login, será apresentado um banner com os botões *Entrar* (para acesso a uma conta existente) e *Cadastrar* (para criar uma nova conta). Abaixo, na Figura 28 há um carrossel que exibe horizontalmente os produtos sincronizados da estação atual.

Figura 28 – Tela Home usuário convidado

Quando o usuário clica em um dos botões do *banner* de autenticação, ele é redirecionado para a tela de autenticação, que apresenta duas abas: *Entrar* ou *Cadastrar*, dependendo do botão selecionado pelo usuário na tela inicial.

Na aba *Entrar* (Figura 29), o usuário encontra campos para inserir seu e-mail e senha, que devem ser obrigatoriamente preenchidos, além de um botão para recuperar a senha, se necessário. Já na aba *Cadastrar* (Figura 18), além dos campos mencionados

anteriormente, há também um campo para confirmar a senha, proporcionando uma experiência de registro completa e segura.

Figura 29 – Tela Login

Figura 30 – Tela Cadastrar

Após o login no aplicativo, o usuário é redirecionado para a tela Home, onde encontra uma experiência simplificada, como mostra a Figura 31. Agora, sem o *banner* para *login*, a tela apresenta apenas o carrossel com os produtos da estação, oferecendo uma visualização imediata e direta dos itens disponíveis.

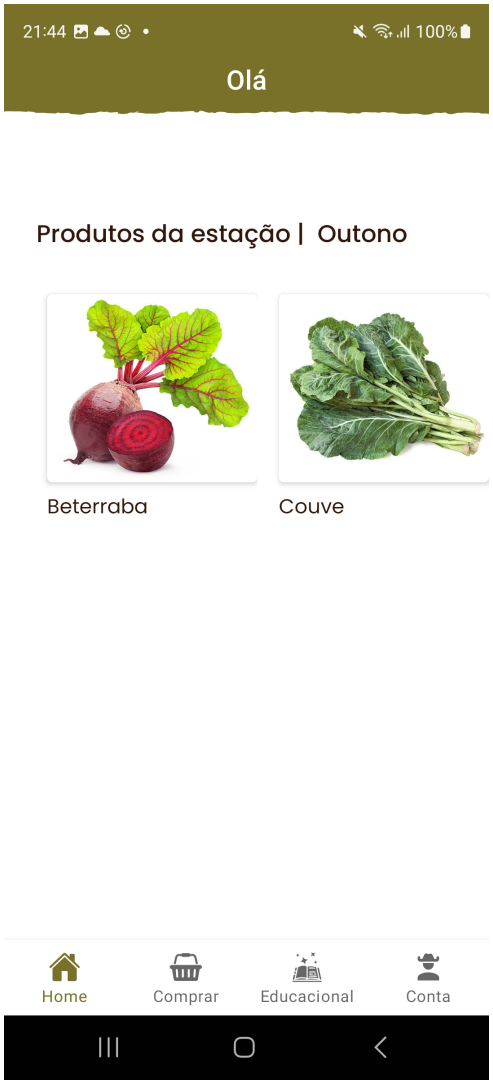

Figura 31 – Tela Home usuário autenticado

Ao acessar a aba *Conta* pelo menu inferior, o usuário autenticado será recebido com uma saudação personalizada, apresentando seu nome e foto de perfil, previamente sincronizados a partir do seu e-mail de cadastro, como apresentado na Figura 32. Além disso, são disponibilizados os botões *Deletar conta* e *Sair da conta*, permitindo um controle completo sobre sua conta e oferecendo a opção de gerenciamento de perfil. Por outro lado, caso o usuário não esteja autenticado, será apresentado um convite para se cadastrar, acompanhado de uma animação convidativa e do botão *Cadastro*, incentivando a participação e facilitando o processo de adesão à plataforma, como segue na Figura 33.

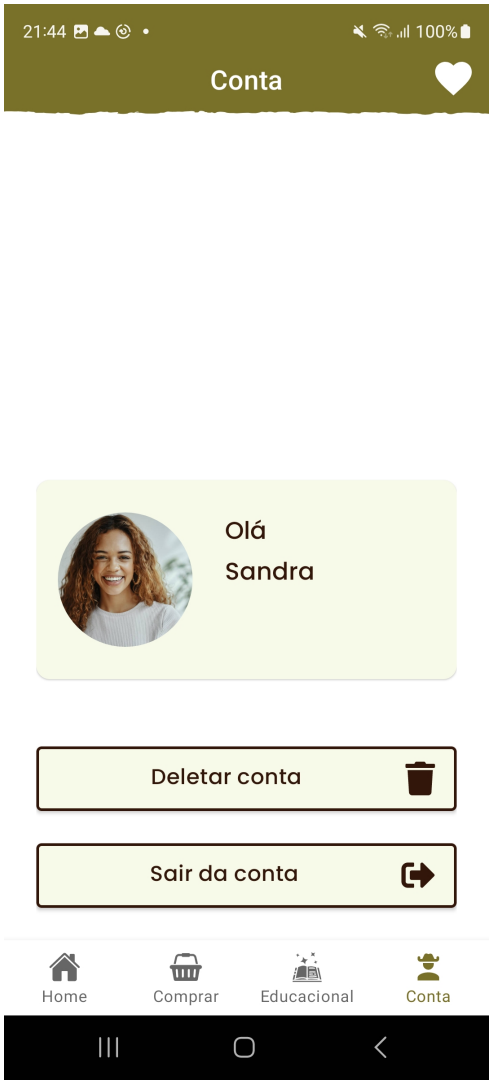

Figura 32 – Perfil usuário autenticado

Figura 33 – Perfil usuário convidado

Ao acessar a aba *Comprar*, que representa o principal foco deste trabalho, o usuário será inicialmente solicitado a conceder permissão para acessar a localização do dispositivo. Seguindo as diretrizes recomendadas pela Google, essa solicitação é feita somente quando o usuário acessa uma funcionalidade que requer essa permissão específica. O usuário terá três opções à disposição, como apresentado na Figura 34: permitir o acesso enquanto estiver utilizando o aplicativo, conceder permissão apenas uma vez ou negar o acesso.

No caso de recusa da permissão, será exibido o diálogo contextual da Figura 35, explicando ao usuário a razão por trás da solicitação de permissão, garantindo sua segurança e informando que ele pode alterar essa configuração posteriormente nas configurações do aplicativo, seguindo as melhores práticas recomendadas.

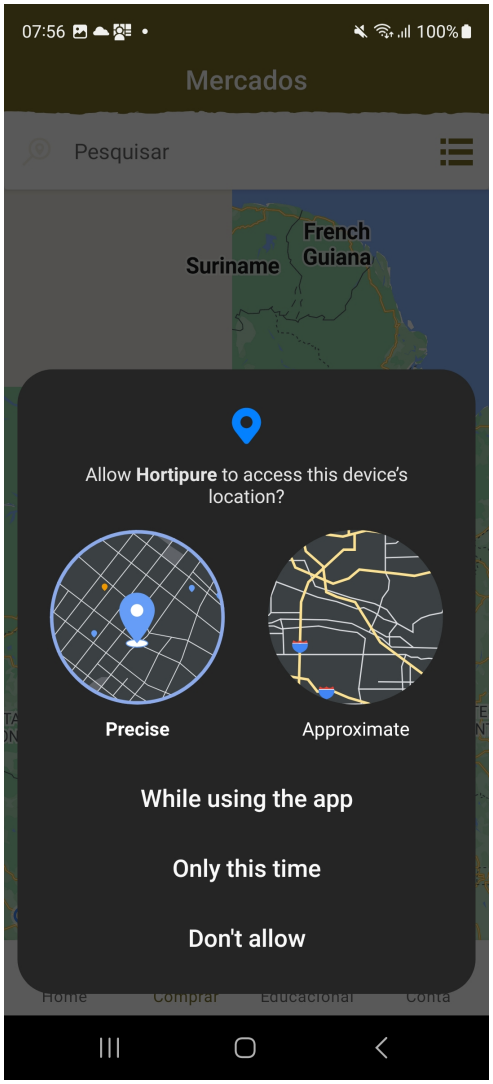

Figura 34 – Permissão de localização

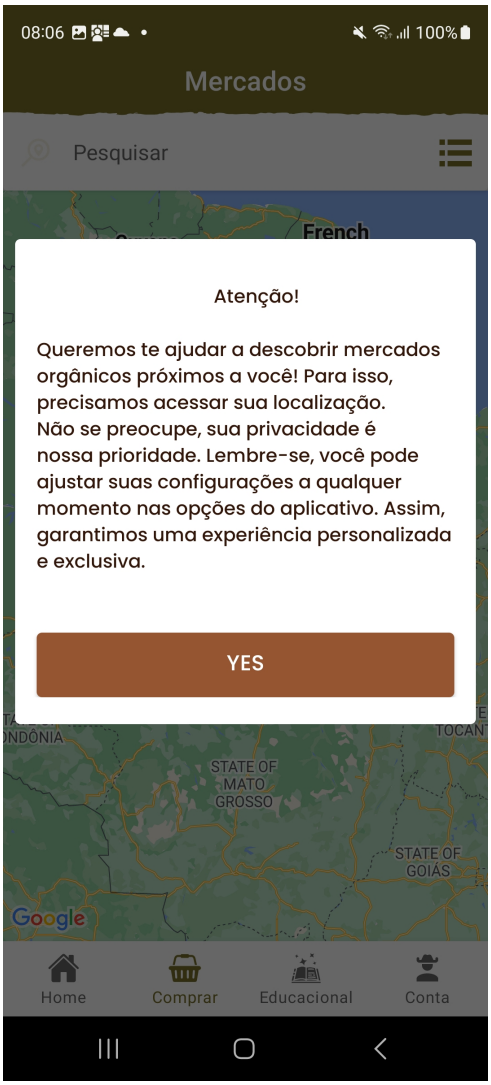

Figura 35 – Pop-up Pop-up informativo

Em seguida, o usuário será direcionado para o Google Maps, com a visualização centralizada nas coordenadas centrais do Brasil, conforme Figura 36. Na parte superior da tela, uma barra de pesquisa estará disponível, permitindo ao usuário buscar por cidade, estado ou país conforme sua preferência, como mostra a Figura 37.

Figura 36 – Busca sem permissão concedida

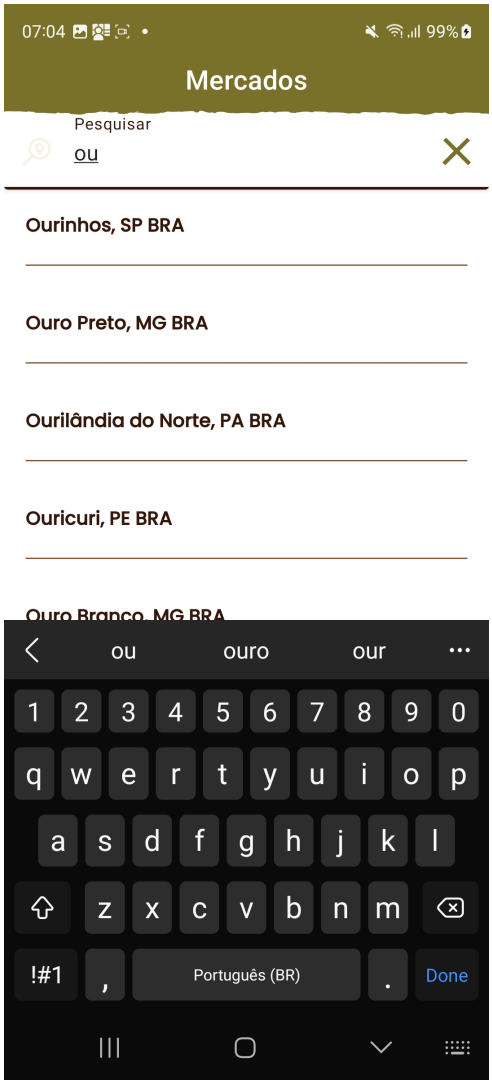

Figura 37 – Resultados da barra de pesquisa

Além da busca pela barra de pesquisa, os usuários também têm a opção de realizar uma busca manual no mapa, Figura 38. Ao clicar no botão *Procure nesta área*, é possível explorar visualmente o mapa para encontrar locais de interesse. Se, em qualquer uma dessas formas de busca, nenhum mercado for encontrado dentro de um raio de 80 km, o usuário receberá uma notificação por meio de uma caixa de diálogo, informando sobre a ausência de resultados (Figura 39).

Figura 38 – Mapa com botão de busca manual

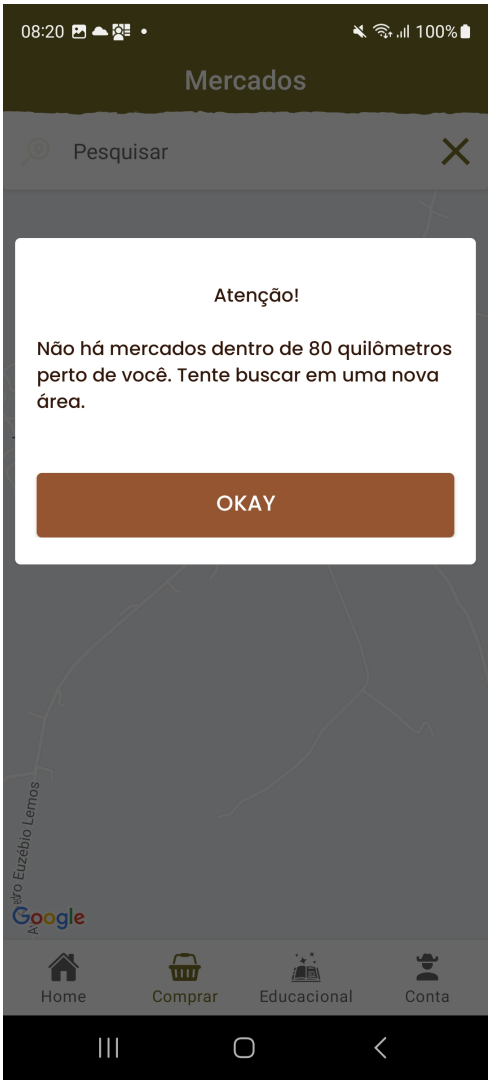

Figura 39 – Nenhum resultado encontrado na busca

Quando o usuário concede permissão para acessar sua localização, ao abrir a aba de compras posteriormente, ele é apresentado aos mercados mais próximos a ele, como mostra a Figura 40. O mercado com a menor distância é automaticamente pré-selecionado para facilitar a escolha inicial. Além disso, ao clicar no ícone de lista na barra de pesquisa, o usuário pode alternar para uma visualização em lista dos mercados próximos, a qual pode ser observada na Figura 41. Isso oferece flexibilidade na escolha, permitindo ao usuário alternar entre esses dois modos de visualização conforme sua preferência.

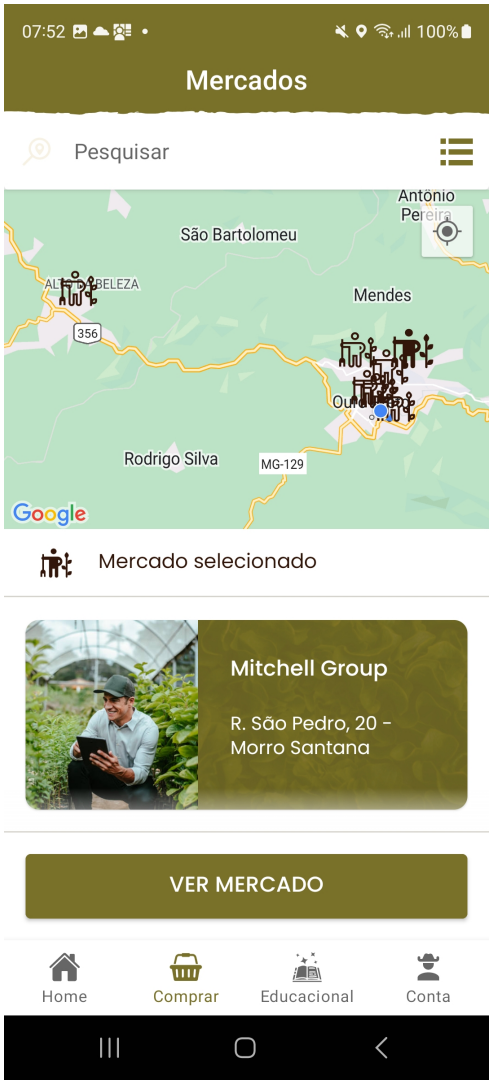

Figura 40 – Localização Autorizada

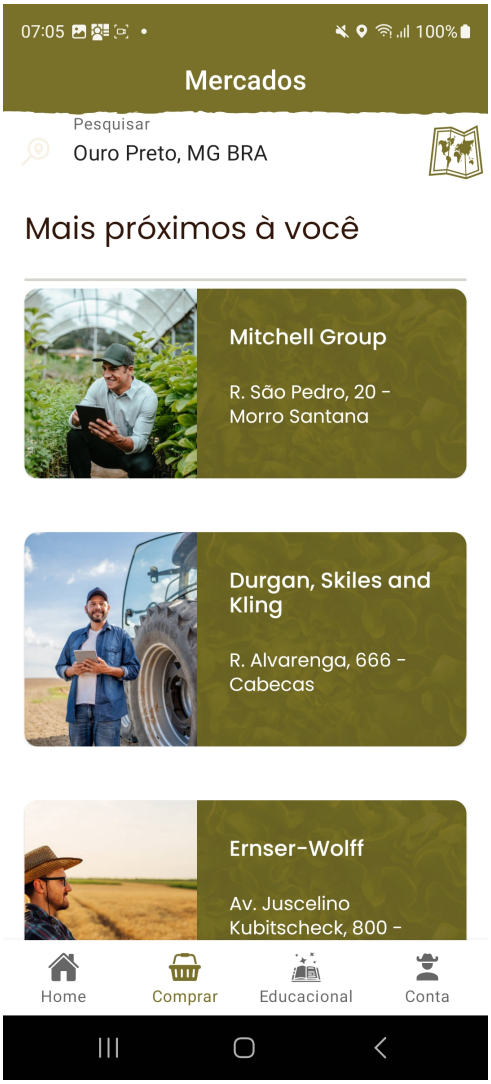

Figura 41 – Lista de resultados

Ao selecionar um mercado, o usuário é conduzido à tela de *Detalhes do Mercado* apresentada na Figura 42, onde pode encontrar um resumo conciso sobre o agricultor, incluindo sua história e valores. Essa abordagem promove a conexão entre consumidor e produtor, fortalecendo a agricultura familiar, que é um dos objetivos essenciais deste projeto. Logo abaixo, o usuário encontra um carrossel com as categorias de produtos oferecidos, que incluem tubérculos, verduras, frutas, legumes e grãos.

Na parte inferior da tela, estão posicionados dois botões flutuantes. À esquerda, um botão direciona o usuário para o WhatsApp do produtor, facilitando o processo de pedido. À direita, outro botão o encaminha para o discador do celular, permitindo que ele entre em contato por telefone e conclua a compra de forma conveniente. Essa funcionalidade visa proporcionar uma experiência de compra fluida e acessível ao usuário.

Figura 42 – Tela de Detalhes do Mercado

Ao escolher uma categoria, o usuário é redirecionado para a tela de listagem dos produtos relacionados àquela categoria específica, representada pela Figura 43. Nessa tela, ele pode verificar detalhes como o nome do produto, seu preço e um ícone indicando se o produto é orgânico. Além disso, na parte inferior da tela, estão localizados os botões que possibilitam a conclusão da compra após a visualização dos produtos. Essa organização intuitiva visa oferecer ao usuário uma experiência de navegação simplificada e eficiente durante sua jornada de compra.

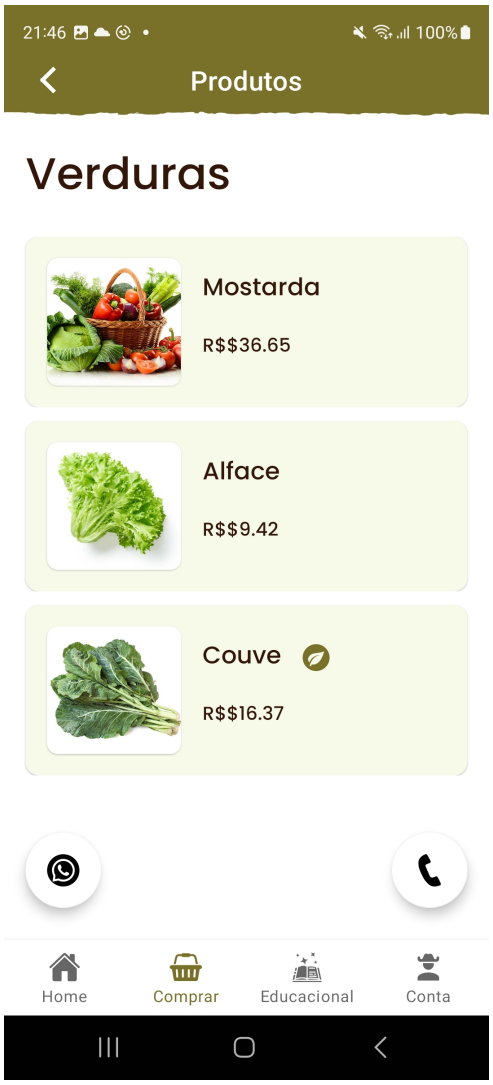

Figura 43 – Tela de Detalhes dos Produtos

6 Considerações finais

O presente trabalho teve como foco principal a construção e desenvolvimento de uma aplicação front-end Android voltada para promover o desenvolvimento sustentável da região, mediante o apoio aos agricultores familiares locais e o estímulo a escolhas alimentares e práticas agrícolas mais responsáveis com o meio ambiente. Também foram definidos objetivos específicos que englobam a criação de uma interface entre consumidor e produtor, visando facilitar o acesso e a comercialização dos produtos agrícolas, assim como proporcionar aos agricultores um espaço para compartilhar informações sobre suas produções, trajetórias e conhecimentos, humanizando o processo de produção e estreitando o vínculo entre produtor e consumidor.

A adoção de práticas de *design system* garantiu uma experiência agradável e consistente ao usuário, enquanto a integração com o *Google Maps* proporcionou navegação intuitiva e personalização dos pontos no mapa, sem custos adicionais. A escolha do Kotlin como linguagem de desenvolvimento também foi crucial, oferecendo bibliotecas modernas do ecossistema Android e uma sintaxe concisa, o que resultou em uma melhora significativa na performance e eficiência do desenvolvimento. Além disso, a implementação do *Firebase Authentication* permitiu um sistema de cadastro e *login* seguro e confiável, economizando tempo ao evitar a necessidade de desenvolver um banco de dados personalizado para autenticação. Além de fornecer análises detalhadas sobre o comportamento dos usuários, essenciais para a evolução da aplicação.

Em suma, a criação desta aplicação representa um avanço significativo na promoção do desenvolvimento sustentável da região, demonstrando o potencial das tecnologias móveis para impulsionar mudanças positivas na agricultura e no consumo de alimentos.

6.1 Trabalhos Futuros

Há um vasto potencial a ser explorado nesta aplicação, abrindo diversas possibilidades para sua contínua evolução. Inicialmente, é crucial estabelecer uma conexão com a comunidade da agricultura familiar em Ouro Preto, permitindo a integração dessas pessoas na plataforma, o que infelizmente não foi realizado neste trabalho devido às limitações de acesso e ao curto prazo para estabelecer parcerias.

Após estabelecido o contato com os agricultores familiares e a aquisição dos dados necessários, seria possível disponibilizar uma primeira versão do aplicativo na Play Store para que ele receba seus primeiros usuários.

Em seguida, seria benéfico implementar a seção “Educacional”, onde agricultores e

usuários poderiam compartilhar e acessar conteúdos educativos e informativos, fortalecendo ainda mais os laços na comunidade. Além disso, funcionalidades como a capacidade de favoritar mercados poderiam ser interessantes para aprimorar a experiência do usuário.

Num futuro mais distante, a aplicação poderia expandir suas funcionalidades para abranger todo o processo de compra, desde a criação da lista de compras até o fluxo de pagamento e entrega dos produtos. Essa expansão permitiria uma experiência de compra completa e integrada, proporcionando ainda mais valor aos usuários e produtores envolvidos.

Referências

ALBERTIN, Alberto. *Comércio Eletrônico: Modelo, Aspectos e Contribuições de sua Aplicação*. Jan. 2010. ISBN 9788522456857. Citado 1 vez na página 16.

ALVES F. V.; ALVA, R. C. Agrotóxicos. *Embrapa*, Campo Grande, MS: Embrapa Gado de Corte, 2022., 2022. Citado 1 vez na página 11.

BADGLEY, Catherine et al. Organic Agriculture and the Global Food Supply. *Renewable Agriculture and Food Systems*, v. 22, p. 86–108, jun. 2007. DOI: 10.1017/S1742170507001640. Citado 1 vez na página 15.

BAOURAKIS, G.; KOURGIANTAKIS, Markos; MIGDALAS, Athanasios. The impact of e-commerce on agro-food marketing: The case of agricultural cooperatives, firms and consumers in Crete. *British Food Journal*, v. 104, p. 580–590, set. 2002. DOI: 10.1108/00070700210425976. Citado 1 vez na página 12.

BELKHIR, Abdelkarim et al. An Observational Study on the State of REST API Uses in Android Mobile Applications. In: p. 66–75. DOI: 10.1109/MOBILESoft.2019.00020. Citado 2 vezes na página 22.

BELQASMI, Fatna et al. SOAP-based vs. RESTful web services: A case study for multimedia conferencing. *Internet Computing, IEEE*, v. 16, p. 54–63, jul. 2012. DOI: 10.1109/MIC.2012.62. Citado 1 vez na página 22.

BHATTACHARYA, Deya. *Azure DevOpS: Project management made easy for agile teams*. 2024. Disponível em: https://unito.io/blog/azure-devops-project-management/. Acesso em: 6 mai. 2024. Citado 1 vez na página 28.

BITTENCOURT, Gilson Alceu; BIANCHINI, Valter. *Agricultura familiar na região sul do Brasil*. 1996. Citado 1 vez na página 15.

BOSE, Subham. A COMPARATIVE STUDY: JAVA VS KOTLIN PROGRAMMING IN ANDROID APPLICATION DEVELOPMENT. *International Journal of Advanced Research in Computer Science*, v. 9, p. 41–45, jun. 2018. DOI: 10.26483/ijarcs.v9i3.5978. Citado 1 vez na página 19.

BRASIL, Greenpeace. *Quem é o Greenpeace Brasil?* 2024. Disponível em: https://www.greenpeace.org/brasil/quem-somos/. Acesso em: 24 set. 2024. Citado 1 vez na página 24.

BRASIL, Ministério da Saúde. *Guia alimentar para a população brasileira: promovendo a alimentação saudável*. Ministério da Saúde Brasília, 2006. Citado 1 vez na página 15.

BURITY, Valéria et al. *Direito humano à alimentação adequada no contexto da segurança alimentar e nutricional*. Abrandh Brasília, 2010. v. 91. Citado 1 vez na página 12.

CARVALHO, C de O; CARVALHO, Glauco Rodrigues. Utilização da internet e adoção do e-commerce pelas organizações da agricultura familiar brasileira. In: IN: CONGRESSO DA SOCIEDADE BRASILEIRA DE ECONOMIA, ADMINISTRAÇÃO E ... Citado 1 vez na página 12.

CERI, Stefano; FRATERNALI, Piero et al. *Designing database applications with objects and rules: the IDEA Methodology*. Addison-Wesley, 1997. Citado 1 vez na página 32.

COCKBURN, Alistair. *Escrevendo Casos de Usos Eficazes: Um guia prático para desenvolvedores de software*. Bookman Editora, 2005. Citado 1 vez na página 33.

CRACK, JSON. *JSON response*. 2023. Disponível em: https://jsoncrack.com/editor. Acesso em: 7 ago. 2023. Citado 0 vez na página 23.

CRUZ, Wander. Crescimento do e-commerce no Brasil: desenvolvimento, serviços logísticos e o impulso da pandemia de Covid-19. *GeoTextos*, v. 17, jul. 2021. DOI: 10.9771/geo.v17i1.44572. Citado 1 vez nas páginas 16, 17.

CURRY, David. *Android Statistics (2024)*. 2023. Disponível em: https://www.businessofapps.com/data/android-statistics/. Acesso em: 4 ago. 2023. Citado 1 vez na página 19.

DE ARRUDA SARON, Flávio; NETO, Carlos; HESPANHOL, Antonio. AS DIFICULDADES DE IMPLEMENTAÇÃO DOS PROGRAMAS DE APOIO À AGRICULTURA FAMILIAR NO BRASIL. *Boletim de Geografia*, v. 35, p. 1, ago. 2017. DOI: 10.4025/bolgeogr.v35i1.30818. Citado 1 vez na página 16.

DEVELOPERS, Android. *Conheça o Android Studio*. 2023. Disponível em: https://developer.android.com/studio/intro?hl=pt-br. Acesso em: 4 ago. 2023. Citado 1 vez na página 26.

DEVELOPERS, Android. *Guia para a arquitetura do app | Desenvolvedores Android | Android Developers*. 2023. Disponível em: https://developer.android.com/jetpack/guide?hl=pt-br#mobile-app-ux. Acesso em: 4 ago. 2023. Citado 1 vez na página 20.

DEVELOPERS, Android. *Histórias de desenvolvedores Kotlin | Android Developers*. 2021. Disponível em: https://developer.android.com/kotlin/stories?hl=pt-br. Acesso em: 4 ago. 2023. Citado 0 vez na página 20.

DEVELOPERS, Google for. *Firebase Authentication*. 2023. Disponível em: https://firebase.google.com/docs/auth?hl=pt-br. Acesso em: 4 ago. 2023. Citado 1 vez na página 27.

DUTRA, Rodrigo Marciel Soares; SOUZA, Murilo Mendonça Oliveira de. CERRADO, REVOLUÇÃO VERDE E EVOLUÇÃO DO CONSUMO DE AGROTÓXICOS. *Sociedade & Natureza*, Editora da Universidade Federal de Uberlândia - EDUFU, v. 29, n. 3, p. 473–488, set. 2017. ISSN 1982-4513. DOI: 10.14393/SN-v29n3-2017-8. Disponível em: https://doi.org/10.14393/SN-v29n3-2017-8. Citado 1 vez na página 11.

EPILOKSA, Hammamul Achdan; KUSUMO, Dana Sulistyo; ADRIAN, Monterico. Effect Of MVVM Architecture Pattern on Android Based Application Performance. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, v. 6, n. 4, p. 1949–1955, 2022. Citado 5 vezes na página 21.

FERGUSON, Robert W.; LAMI, Giuseppe. An Empirical Study on the Relationship between Defective Requirements and Test Failures. In: 2006 30th Annual IEEE/NASA Software Engineering Workshop. 2006. P. 7–10. DOI: 10.1109/SEW.2006.9. Citado 1 vez na página 29.

FIGMA. *Free Prototyping Tool: Build Interactive Prototype Designs | FigMa*. Disponível em: https://www.figma.com/prototyping/. Acesso em: 6 mai. 2024. Citado 1 vez na página 27.

FLAUZINO, Matheus et al. Are you still smelling it?: A comparative study between Java and Kotlin language. In: p. 23–32. DOI: 10.1145/3267183.3267186. Citado 1 vez na página 20.

GARCIA, Simone; LARA, Taynah. O impacto do uso dos agrotóxicos na saúde pública: revisão de literatura. *Saúde e Desenvolvimento Humano*, v. 8, p. 85, mar. 2020. DOI: 10.18316/sdh.v8i1.6087. Citado 1 vez na página 11.

GONÇALVES ROCHA, Eduardo; RIBEIRO, Camila. PROPOSTAS DE ENFRENTAMENTO ÀS EXTERNALIDADES NEGATIVAS DECORRENTES DO USO DE AGROTÓXICOS NO BRASIL. v. 19, jun. 2022. DOI: 10.18623/rvd.v19i43.2172. Citado 1 vez na página 11.

GOOGLE. *Material design*. 2018. Disponível em: https://m2.material.io/. Acesso em: 6 mai. 2024. Citado 1 vez na página 27.

GREENPEACE. *Guia de Produtores agroecológicos e orgânicos*. 2024. Disponível em: https://greenpeace.org.br/comida-sem-veneno/. Acesso em: 17 ago. 2024. Citado 1 vez na página 24.

GROFFE, Renato. *Mockable.io: criando rapidamente mocks de APIs REST para testes*. 2021. Disponível em: https://renatogroffe.medium.com/mockable-io-criando-rapidamente-mocks-de-apis-rest-para-testes-b0f54032be3b. Acesso em: 6 mai. 2024. Citado 1 vez na página 27.

KIYOTA, Norma; GOMES, Marcos Affonso Ortiz. AGRICULTURA FAMILIAR E SUAS ESTRATÉGIAS DECOMERCIALIZAÇÃO: UM ESTUDO DE CASO NO MUNICÍPIO DE CAPANEMA–REGIÃO SUDOESTE DOPARANÁ. *Organizações Rurais & Agroindustriais*, v. 1, n. 2, 1999. Citado 1 vez na página 12.

KUMAR, Priyank. *Understanding MVVM Architecture in Android | by Priyank Kumar | The Startup | Medium*. 2020. Disponível em: https://medium.com/swlh/understanding-mvvm-architecture-in-android-aa66f7e1a70b. Acesso em: 4 ago. 2023. Citado 0 vez na página 22.

LAGE, Marcela Ferreira Rocha et al. O mercado orgânico de BH: um estudo de caso das feiras orgânicas municipais. Universidade Federal Rural do Rio de Janeiro, 2016. Citado 1 vez na página 15.

LOU, Tian et al. A comparison of Android native app architecture MVC, MVP and MVVM. *Eindhoven University of Technology*, 2016. Citado 1 vez na página 21.

MACHADO, Luiz Carlos Pinheiro. A Dialética da Agroecologia: contribuição para um mundo com alimentos sem veneno. *São Paulo: Expressão Popular*, 2013. Citado 1 vez na página 12.

MOREIRA, Josino C et al. Avaliação integrada do impacto do uso de agrotóxicos sobre a saúde humana em uma comunidade agrícola de Nova Friburgo, RJ. *Ciência & Saúde Coletiva*, SciELO Public Health, v. 7, p. 299–311, 2002. Citado 1 vez na página 15.

NATURIS. *Naturis*. 2024. Disponível em: https://www.naturis.eco.br/empresa. Acesso em: 17 ago. 2024. Citado 1 vez na página 24.

OLIVEIRA, Karolyne Maria Alves de et al. O Uso de modelos e Múltiplos Protótipos na Concepção de Interface do Usuário, 2007. Citado 2 vezes nas páginas 38, 39.

OURO PRETO, Prefeitura Municipal de. *Notícias*. 2022. Disponível em: https://ouropreto.mg.gov.br/noticia/2555. Acesso em: 4 ago. 2023. Citado 1 vez na página 13.

PLAY, Google. *Natural da Terra*. 2024. Disponível em: https://play.google.com/store/apps/details?id=br.com.naturaldaterra.app. Acesso em: 17 ago. 2024. Citado 1 vez na página 25.

PRESSMAN, Roger S; MAXIM, Bruce R. *Engenharia de software-9*. McGraw Hill Brasil, 2021. Citado 1 vez na página 17.

QUEIROZ, Leonardo Rossatto. IPhone, Android, e a consolidação da cultura do smartphone: o papel do IPhone e do Sistema Operacional Android como catalisadores da consolidação no mercado de smartphones em escala global. *Revista Tecnologia e Sociedade*, Universidade Tecnológica Federal do Paraná, v. 14, n. 30, p. 47–70, 2018. Citado 1 vez na página 19.

RODRIGUES, Lucas Siqueira; MACÊDO, Lucas Pereira de Andrade. Inovações tecnológicas na agricultura familiar: Agromart, 2021. Citado 1 vez na página 24.

SANTOS, Christiane Fernandes dos et al. A agroecologia como perspectiva de sustentabilidade na agricultura familiar. *Ambiente & Sociedade*, ANPPAS - Revista Ambiente e Sociedade, v. 17, n. 2, p. 33–52, abr. 2014. ISSN 1414-753X. DOI: 10.1590/S1414-753X2014000200004. Disponível em: https://doi.org/10.1590/S1414-753X2014000200004. Citado 1 vez na página 12.

SCHUTTER, Olivier De. *Relatório do Relator Especial sobre o direito à alimentação*. 2009. Disponível em: http://www.oda-alc.org/documentos/1341790013.pdf. Acesso em: 20 ago. 2024. Citado 1 vez na página 12.

SCHWABER, Ken; SUTHERLAND, Jeff. La guía de Scrum. *Scrumguides. Org*, v. 1, p. 21, 2013. Citado 1 vez na página 18.

SILVA, Sandro Pereira. *Políticas públicas, agricultura familiar e desenvolvimento territorial: uma análise dos impactos socioeconômicos do PRONAF no Território Médio Jequitinhonha-MG*. 2012. Citado 1 vez na página 16.

STATISTA. *El mapa mundial de Android e iOS*. 2023. Disponível em: https://es.statista.com/grafico/29620/sistema-operativo-movil-con-la-mayor-cuota-de-mercado-por-pais/. Acesso em: 5 ago. 2023. Citado 0 vez na página 19.

TEIXEIRA, Tarcisio Miguel. SAÚDE E DIREITO À INFORMAÇÃO: O PROBLEMA DOS AGROTÓXICOS NOS ALIMENTOS. *Revista de Direito Sanitário*, v. 17, n. 3, p. 134–159, mar. 2017. DOI: 10.11606/issn.2316-9044.v17i3p134-159. Disponível em: https://www.revistas.usp.br/rdisan/article/view/127782. Citado 1 vez na página 11.

TINOCO, Sonia Terezinha Juliatto. Conceituação de agricultura familiar: uma revisão bibliográfica. *Hipertexto, acessado em*, v. 15, 2012. Citado 1 vez na página 15.

VISSOTTO, Elisa Maria; BONIATI, Bruno Batista. Comércio eletrônico. Frederico Westphalen-RS, 2013. Citado 1 vez na página 16.