



FACULDADE DE TECNOLOGIA DE TAUBATÉ

**GUSTAVO CAMARGO EKLUND FRANÇA
KAROLINE DOS SANTOS SAMPAIO**

**UCYCLE – DESENVOLVIMENTO DE UMA APLICAÇÃO
MOBILE DE MODA CIRCULAR**

TAUBATÉ

2023



FACULDADE DE TECNOLOGIA DE TAUBATÉ

**GUSTAVO CAMARGO EKLUND FRANÇA
KAROLINE DOS SANTOS SAMPAIO**

**UCYCLE – DESENVOLVIMENTO DE UMA APLICAÇÃO
MOBILE DE MODA CIRCULAR**

Trabalho de Graduação apresentado à Coordenação do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Centro Estadual de Educação Tecnológica Paula Souza para a obtenção do diploma de Tecnólogo em Análise e Desenvolvimento de Sistemas.

**Orientador: Prof. Me. Luiz Eduardo Souza
Evangelista**

**Coorientador: Prof. Esp. Luís Felipe Férez
Santos**

TAUBATÉ

2023

**GUSTAVO CAMARGO EKLUND FRANÇA
KAROLINE DOS SANTOS SAMPAIO**

**UCYCLE – DESENVOLVIMENTO DE UMA APLICAÇÃO
DE MODA CIRCULAR**

Trabalho de Graduação apresentado à
Coordenação do Curso Superior de Tecnologia
em Análise e Desenvolvimento de Sistemas do
Centro Estadual de Educação Tecnológica
Paula Souza para a obtenção do diploma de
Tecnólogo em Análise e Desenvolvimento de
Sistemas.

**Orientador: Prof. Me. Luiz Eduardo Souza
Evangelista**

**Coorientador: Prof. Esp. Luís Felipe Férez
Santos**

Taubaté, _____ de _____ de 2023.

BANCA EXAMINADORA

Prof. Me. Luiz Eduardo Souza Evangelista
FATEC - Taubaté

Prof. Esp. Luís Felipe Férez Santos
FATEC - Taubaté

AGRADECIMENTOS

Gostaríamos de expressar nossa profunda gratidão a todas as pessoas que contribuíram para a realização desta monografia. Em primeiro lugar, queremos agradecer aos nossos pais e familiares pelo amor, apoio incondicional e encorajamento constantes ao longo dessa jornada acadêmica. Sem o seu apoio, não estaríamos aqui hoje.

Também desejamos agradecer sinceramente ao corpo docente da instituição pelo conhecimento compartilhado, pela dedicação à educação e pela orientação valiosa durante todo o processo de pesquisa. Suas mentorias foram fundamentais para o nosso crescimento intelectual e desenvolvimento acadêmico.

Somos a primeira geração que sente as consequências das mudanças climáticas e a última que tem a oportunidade de fazer algo para deter isso.

(Barack Obama)

RESUMO

Analisando a relação entre a atual sociedade de consumidores e o consumo desenfreado pode-se ver o impacto negativo no meio ambiente, sobretudo na indústria da moda. Os dados sobre a produção excessiva de roupas, os resíduos têxteis gerados e as emissões de gases de efeito estufa causadas pela indústria da moda são alarmantes, há uma urgência em abordar os impactos ambientais associados à indústria da moda. Em contrapartida há uma crescente conscientização das pessoas sobre a importância da sustentabilidade.

Diante desse cenário o objetivo desse trabalho é desenvolver um aplicativo móvel que promova a moda circular e a logística reversa gerando um impacto positivo no meio ambiente, incentivando o consumo consciente e a reutilização de materiais, permitindo que ao invés de descartar esses resíduos em aterros eles possam ser disponibilizados para venda viabilizando aos artesãos a busca por materiais de descarte que podem vir a tornar um novo produto, por meio do *upcycling*.

Palavras-chave: moda circular, *e-commerce*, logística reversa, aplicativo *mobile*

ABSTRACT

Analyzing the relationship between today's consumer society and rampant consumption, the negative impact on the environment becomes evident, particularly in the fashion industry. The data concerning excessive clothing production, textile waste generation, and greenhouse gas emissions caused by the fashion industry are alarming. There is an urgent need to address the environmental impacts associated with the fashion industry. On the other hand, there is a growing awareness among individuals about the importance of sustainability.

Considering this scenario, the objective of this work is to develop a mobile application that promotes circular fashion and reverse logistics, generating a positive impact on the environment. It aims to encourage conscious consumption and material reuse, allowing the waste materials to be made available for sale instead of being discarded in landfills. This initiative enables artisans to seek discarded materials that can be transformed into new products through upcycling.

Keywords: circular fashion, e-commerce, reverse logistics, mobile application

LISTA DE ILUSTRAÇÕES

Figura 1 - Event Storming.....	38
Figura 2 - Diagrama de modelo de domínio	39
Figura 3 - Diagrama de arquitetura	41
Figura 4 - A Arquitetura Limpa	48
Figura 5 - Tela inicial.....	55
Figura 6 - Tela inicial com pesquisa	55
Figura 7 - Perfil visitante.....	56
Figura 8 - Cadastro	57
Figura 9 - Login	58
Figura 10 - Esqueceu a senha	59
Figura 11 - Perfil	60
Figura 12 - Configurações	61
Figura 13 - Endereços	62
Figura 14 - Adicione um endereço.....	63
Figura 15 - Adicione um endereço.....	63
Figura 16 - Introdução ao anunciar produto	64
Figura 17 - Definir o título ao anunciar o produto	64
Figura 18 - Definir descrição ao anunciar o produto.....	65
Figura 19 - Definir a condição ao anunciar o produto.....	65
Figura 20 - Instruções para adicionar fotos ao anunciar o produto	66
Figura 21 - Adicionar fotos ao anunciar o produto.....	66
Figura 22 - Editar fotos ao anunciar o produto	67
Figura 23 - Descrever o produto ao anunciar o produto	67
Figura 24 - Definir preço ao anunciar o produto	68
Figura 25 - Definir garantia ao anunciar o produto	68
Figura 26 - Definir o tempo de garantia ao anunciar o produto	69
Figura 27 - Aviso de revisão ao anunciar o produto	69
Figura 28 - Revisar e anunciar o produto	70
Figura 29 - Revisar e anunciar o produto	70
Figura 30 - Detalhes do produto.....	71

Figura 31 - Detalhes do produto.....	71
Figura 32 - Carrinho de compras.....	72
Figura 33 - Finalizar compra.....	73
Figura 34 - Finalizar compra.....	73
Figura 35 - Pagamento do pedido com opções para selecionar	74
Figura 36 - Pagamento do pedido via cartão de crédito.....	74
Figura 37 - Meus pedidos, tema claro	75
Figura 38 - Meus pedidos, tema escuro	75

LISTA DE TABELAS

Tabela 1 - User Stories (Histórias de usuário) previamente elicitados do projeto	34
Tabela 2 - Requisitos Não Funcionais	37

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
AWS	<i>Amazon Web Services</i>
C.Q.R.S.	<i>Command Query Responsibility Segregation</i>
CDN	<i>Content Delivery Network</i>
CEP	Código de Endereçamento Postal
DDD	<i>Domain-Driven Design</i>
DIP	<i>Dependency Inversion Principle</i>
EDA	<i>Event Driven Architecture</i>
EMF	<i>Ellen MacArthur Foundation</i>
GEE	Gases de Efeito Estufa
HTTPS	<i>Hyper Text Transfer Protocol Secure</i>
IAM	<i>Identity and Access Management</i>
IBGE	Instituto Brasileiro de Geografia e Estatística
IOS	<i>iPhone Operating System</i>
IPEA	Instituto de Pesquisa Econômica Aplicada
ISP	<i>Interface Segregation Principle</i>
KYC	<i>Know Your Customer</i>
LSP	<i>Liskov Substitution Principle</i>
OCP	<i>Open/Closed Principle</i>
PCI-DSS	<i>Payment Card Industry Data Security Standard</i>
PWA	<i>Progressive Web Apps</i>
RNF	Requisitos Não Funcionais
SDK	<i>Software Development Kit</i>
SRP	<i>Single Responsibility Principle</i>
SW	<i>Service Workers</i>
TG	Trabalho de Graduação
URL	<i>Uniform Resource Locator</i>
US	<i>User Stories</i>
WDR	<i>Web Design Responsivo</i>

SUMÁRIO

1 INTRODUÇÃO	13
1.1 JUSTIFICATIVA	16
1.2 OBJETIVOS	19
1.2.1 Objetivo geral.....	20
1.2.2 Objetivos específicos	20
2 FUNDAMENTAÇÃO TEÓRICA	21
2.1 DISPOSITIVOS MÓVEIS	21
2.2 ABORDAGENS DE DESENVOLVIMENTO MOBILE	23
2.2.1 PWA (<i>Progressive web apps</i>).....	24
2.3 BACK-END	26
2.3.1 API (<i>Application Programming Interface</i>).....	26
2.4 FRONT END	27
2.5 BANCO DE DADOS	27
3 METODOLOGIA	28
3.1 CLASSIFICAÇÕES DE PESQUISA	28
3.1.1 Natureza da pesquisa	28
3.1.2 Forma da pesquisa	29
3.1.3 Objetivo da pesquisa	29
3.2 APLICAÇÃO DA METODOLOGIA	30
3.2.1 Planejamento da pesquisa.....	30
3.2.2 Atividades	31
3.3 METODOLOGIA DE GERENCIAMENTO DE PROJETOS	32
3.3.1 Desenvolvimento do aplicativo uCycle.....	33
4 DESENVOLVIMENTO	34
4.1 REQUISITOS	34
4.2 DIAGRAMAS	38
4.2.1 <i>Event Storming</i>	38

4.2.2 Diagrama de modelo de domínio	39
4.2.3 Diagrama de caso de uso	40
4.2.4 Diagrama de arquitetura	41
4.3 PADRÕES DE PROJETO	42
4.3.1 S.O.L.I.D.	43
4.3.2 <i>Clean Code</i>	44
4.4 PADRÕES DE ARQUITETURA	45
4.4.1 DDD (<i>Domain-Driven Design</i>).....	45
4.4.2 <i>Clean Architecture</i>	47
4.4.3 <i>Event Driven Architecture</i>	48
4.4.4 C.Q.R.S. (<i>Command Query Responsibility Segregation</i>)	49
4.5 TECNOLOGIA E INFRAESTRUTURA.....	50
4.6 INTEGRAÇÃO COM SISTEMAS INTERNOS.....	52
5 RESULTADOS E DISCUSSÕES.....	54
5.1 RESULTADOS OBTIDOS	54
5.1.1 Capturas de tela e suas funcionalidades	54
6 CONCLUSÃO	76
6.1 TRABALHOS FUTUROS	76
REFERÊNCIA BIBLIOGRÁFICA.....	79
APÊNDICES	87
APÊNDICE A – <i>Links</i> de acesso ao repositório do projeto no GitHub...	87
APÊNDICE B – <i>Links</i> de acesso ao vídeo de demonstração do uCycle	88

1 INTRODUÇÃO

A sociedade atual de consumo teve sua origem no final dos anos 1920, com a transformação da sociedade de produtores. Como muitas vezes acontece com tendências que indicam um futuro ainda não previsto pelos pesquisadores, esse fato passou despercebido por aqueles menos atentos. No entanto, as propagandas da época já enfatizavam preocupações relacionadas ao envelhecimento, vendendo a ideia de que é possível permanecer jovem e bonito para sempre, segundo Kracauer (1963). Desde então, a sociedade de consumidores tem evoluído constantemente e se tornou carregada de significados culturais.

Para Bauman (2008), a sociedade atual molda seus cidadãos para desempenharem o papel de consumidores, de modo que o consumo não é mais visto apenas como um direito adquirido ou um prazer a ser conquistado, mas principalmente como um dever do cidadão. O indivíduo que consome investe em si mesmo, pois existe uma demanda a ser alimentada, caso contrário, o "valor social" dele decai, assim como a autoestima. Segundo o autor (2008 p. 73):

“Numa sociedade de consumidores, todo mundo precisa ser, deve ser e tem que ser um consumidor por vocação (ou seja, ver e tratar o consumo como vocação). Nessa sociedade, o consumo visto e tratado como vocação é ao mesmo tempo um direito e um dever humano universal que não conhece exceção”

Com a popularização da internet e o surgimento das redes sociais, o consumo passa a ter um viés de desejabilidade social muito mais explícito. Nesse novo contexto, a publicidade e a moda encontram um território excepcional para atuarem, uma vez que a produção de si como mercadoria é vista como algo natural como se fosse uma escolha pessoal dissuadida do contexto sociológico atual. Essa presença online acaba por transformar um espaço que a princípio seria de interações humanas por interações de compra e venda. A forma como um indivíduo, ainda que anônimo, expõe sua vida na internet gera desejo, que sustenta a ideia de pertencimento, de se encaixar em uma categoria cultural, fomentando ainda mais o consumo.

O consumo desenfreado tem um impacto significativo e negativo no meio ambiente, resultando em diversas consequências, tais como: o esgotamento de recursos naturais como água, madeira, minerais e combustíveis fósseis; a poluição do ar e da água causada pela produção em massa de bens de consumo, que emitem poluentes como dióxido de carbono e outros compostos, bem como pelo descarte inadequado de materiais, o que pode poluir o solo e as águas; o desmatamento e a perda de biodiversidade decorrentes da necessidade de desmatar áreas naturais e destruir habitats de espécies animais e vegetais para a produção desses bens; e, por fim, as mudanças climáticas, já que o consumo excessivo de bens de consumo contribui para a emissão de gases de efeito estufa, que têm impacto no aquecimento global e nas mudanças climáticas.

Em paralelo a isso, as novas tecnologias e a crise ambiental apresentam outro aspecto para a sociedade de consumidores. A autora Fontenelle (2017) destaca duas grandes mudanças no consumo contemporâneo: o consumo da experiência e o consumo responsável. O consumo da experiência, ou consumo hedônico, é uma prática em que o consumidor busca, além da aquisição do produto e de sua utilidade, uma experiência de compra que ofereça algum diferencial por parte da empresa. Já o consumo responsável é pautado pelas implicações ambientais, sociais ou individuais do ato de consumir, em reação ao consumo predatório que prevaleceu nas últimas décadas.

Frente a essas implicações da sociedade de consumidores e porque o consumo responsável é um mercado a ser explorado, as organizações têm se preocupado cada vez mais com o ciclo de vida de seus produtos, enfatizando como modelo de gestão a logística reversa. A logística reversa é um modelo de gerenciamento que tem como objetivo controlar o fluxo de produtos e resíduos no pós-consumo, a fim de recuperar seu valor ou descartá-los adequadamente.

O autor Govindan (2015) destaca a importância da colaboração entre empresas e outras partes interessadas para a implementação eficaz da logística reversa, o sucesso da gestão da logística reversa depende da colaboração entre empresas, fornecedores, clientes, governos, organizações sem fins lucrativos e outros parceiros na cadeia de suprimentos. Esses atores podem trabalhar juntos para identificar oportunidades de melhoria na gestão de resíduos, como a

implementação de tecnologias mais avançadas e ações de educação e conscientização do público.

Diante desse cenário, o objetivo deste trabalho é criar um aplicativo móvel para promover a moda circular. A sociedade de consumidores tem contribuído para a exploração excessiva dos recursos naturais e produção desenfreada de resíduos, o que prejudica o equilíbrio ambiental do planeta. A moda circular surge como uma alternativa para reduzir a quantidade de resíduos descartados em aterros sanitários e minimizar os impactos negativos sobre o meio ambiente.

A moda circular é um conceito que propõe tornar a indústria da moda mais sustentável e consciente, com o objetivo de minimizar o desperdício e maximizar o reaproveitamento de recursos, por meio de um ciclo contínuo de produção e consumo. Para atingir esse objetivo, a moda circular busca repensar todo o processo produtivo, desde a escolha de materiais até a maneira como as peças são descartadas, dando importância à durabilidade, qualidade e reciclagem. Isso implica em uma perspectiva mais ampla e abrangente da cadeia de valor da moda, incluindo o consumo consciente e a mudança de hábitos dos consumidores.

A aplicação proposta visa promover a moda circular e ter um impacto positivo no meio ambiente, reduzindo a quantidade de resíduos têxteis descartados. Além disso, a aplicação incentiva o artesanato como uma alternativa de economia criativa e o consumo consciente. Isso permitirá aos artesãos buscar materiais que seriam descartados, por meio da compra e venda, e impulsionar a diversidade cultural, a oportunidade de gerar renda e criar empregos, também pode ter um impacto positivo na economia e promover a logística reversa.

Nesta introdução, uma visão geral sobre o problema a ser resolvido é apresentada, bem como a importância desse trabalho de graduação dentro desse contexto. Além disso, são explanados os objetivos e a abordagem metodológica utilizada. O segundo capítulo, por sua vez, embasa mais profundamente o tema de pesquisa através de bibliografias, apresenta soluções semelhantes no mercado e trata, ainda, o lado técnico do trabalho em relação ao desenvolvimento de aplicativos móveis. Em seguida, no terceiro capítulo,

aspectos como os requisitos, diagramas, a estrutura da aplicação, código de programação e a interface de usuário compõem tópicos essenciais do funcionamento do sistema apresentado neste TG (Trabalho de Graduação). No capítulo subsequente, a conclusão do que fora alcançada com este projeto é disposta e, por fim, as fontes de pesquisas utilizadas e apêndices encerram o trabalho.

1.1 JUSTIFICATIVA

A moda como conhecida hoje começou a se desenvolver no final do século XIX, durante a Revolução Industrial. Antes disso, a moda era muito mais limitada em termos de variações e mudanças, e a maioria das roupas era feita sob medida. Com o desenvolvimento da manufatura em massa e a produção em larga escala, a moda se tornou mais acessível e mais rápida em suas mudanças. No final do século XIX e início do século XX, designers de moda como Charles Frederick Worth, Paul Poiret e Coco Chanel começaram a criar roupas que eram mais modernas e despojadas, e que refletiam as mudanças sociais e culturais que estavam ocorrendo na época. A moda como uma indústria globalizada e de grande influência cultural, só começou a surgir nas décadas de 1950 e 1960, com o surgimento do *prêt-à-porter* (roupas de pronta-entrega) e o aumento da produção em massa. A partir daí, a moda se tornou um setor cada vez mais importante da economia global e uma fonte de expressão pessoal e cultural.

Desde então, a produção e o consumo desenfreado na indústria da moda aumentaram exponencialmente, especialmente nas últimas décadas, com a globalização e a rápida evolução da tecnologia de produção têxtil. E em uma sociedade onde o consumo é supervalorizado, uma peça de roupa não contenta apenas uma necessidade física, mas também supre desejos e expectativas emocionais de curto prazo. Infelizmente, esse comportamento contribui para um consumo desenfreado e superficial, resultando em um ciclo de vida curto para as roupas. A consequência é que existe um processo complexo envolvido na produção e descarte de cada peça, o que tem impactos significativos no meio ambiente.

Segundo a pesquisa Possibilidades Para Moda Circular no Brasil - Padrões de Consumo, Uso e Descarte de Roupas do Modifica (2020), nos últimos 15 anos a produção de roupas mais do que duplicou no mundo, em 2000 a produção de peças de roupa no mundo era de 50 bilhões, em 2015 esse número subiu para mais de 100 bilhões, só o Brasil, no ano de 2019 produziu 8,9 bilhões de peças. Juntamente com a produção de peças de roupas cresce também a produção de fibras têxteis, estima-se que até o ano de 2030 serão produzidas 145 milhões de toneladas de fibra têxtil.

Atualmente, o Brasil é um dos maiores produtores têxteis do ocidente e a segunda maior indústria geradora de empregos no país na categoria indústria de transformação. No entanto, estima-se que a indústria têxtil brasileira gera cerca de 170 mil toneladas de resíduos têxteis por ano, dos quais apenas 40% são reprocessados por empresas de reciclagem e o restante é descartado em aterros sanitários segundo relatório do Instituto Brasileiro de Geografia e Estatística (IBGE) de 2019, intitulado "Panorama da Indústria Têxtil e de Confecção no Brasil".

Essa produção têxtil desmedida se dá pelo aumento da classe média no mundo todo e pelo fenômeno das *fast fashion* que impulsiona o aumento da produção de roupas, resultando em uma queda na frequência de uso de cada peça. Devido aos preços mais baixos, as pessoas compram mais roupas, as usam menos vezes e as descartam mais rapidamente. Nos últimos 15 anos, houve uma redução global de 36% na taxa de utilização de roupas, conforme relatado pela Ellen MacArthur Foundation (EMF) em 2018. Esses números podem variar de acordo com a renda per capita de cada país, mais em termos gerais, no mundo estima-se que anualmente as pessoas descartem US\$ 460 bilhões em roupas que poderiam estar em uso. Em se tratando do meio ambiente a indústria global da moda é responsável por uma parcela significativa das emissões de gases de efeito estufa (GEE), contribuindo com cerca de 2,1 bilhões de toneladas em 2018, o que equivale a 4% do total global - mais do que as emissões combinadas da França, Alemanha e Reino Unido. A menos que sejam tomadas medidas significativas, as emissões de GEE da indústria da moda devem aumentar em 1/3, chegando a 2,7 bilhões de toneladas em 2030.

Para reduzir o impacto da moda no clima, é necessário diminuir a produção, adotar práticas produtivas mais sustentáveis, como a descarbonização da rede produtiva, aumentar a vida útil das peças por meio de práticas como a revenda e aluguel, e promover a reutilização de materiais por meio de reaproveitamento e reciclagem. Assim é possível afirmar que a moda circular é uma alternativa viável para a redução dos impactos ambientais causados pela indústria têxtil. A moda circular é um movimento que tem o potencial de reduzir o consumo de novos produtos, incentivando um comportamento mais consciente e sustentável, além de promover o reaproveitamento de materiais e gerar impacto social positivo. Ao valorizar todas as etapas do processo produtivo, incluindo o capital humano envolvido e os consumidores, técnicas como o *upcycling** pode resultar em produtos de alta qualidade, com baixo impacto ambiental e livre de processos químicos, fomentando a ideia de um consumo com propósito.

Ainda sobre a pesquisa Possibilidades Para Moda Circular no Brasil - Padrões de Consumo, Uso e Descarte de Roupas do Modifica (2020), 97% das pessoas acreditam que a moda e o vestuário estão relacionados às mudanças climáticas e têm um impacto significativo sobre o meio ambiente. Além disso, quase metade dos entrevistados (47,1%) consideram a sustentabilidade "muito importante" ao tomar decisões de compra. Questões ambientais lideram a lista de preocupações que motivam as pessoas a fazer compras mais éticas, com 88,1% dos entrevistados indicando essa como uma de suas principais preocupações. Esses números sugerem que os brasileiros estão conscientes dos impactos socioambientais da indústria da moda e, portanto, estão mais inclinados a apoiar iniciativas sustentáveis.

Além disso, segundo o IPEA (2018), a previsão para o mercado de trabalho no Brasil é que, até 2026, 54% dos empregos formais sejam substituídos pela automação, o que representa cerca de 30 milhões de postos de trabalho. Algumas profissões têm menor risco de serem automatizadas, como aquelas voltadas para a área de tecnologia e a profissão de artesão de materiais

* Técnica de reutilizar (objetos ou materiais descartados) de forma a criar um produto de qualidade ou valor superior ao original.

recicláveis. O IBGE estima que existam cerca de 10 milhões de artesãos registrados no país.

Diante de tais fatos, com a premissa de promover a redução de resíduos têxteis descartados no meio ambiente, incentivando a moda circular por meio do *upcycling*, conscientizando as pessoas sobre o consumo sustentável e fomentando a economia criativa por meio do artesanato como uma alternativa viável para empreendedorismo. Este trabalho propõe o desenvolvimento de uma aplicação móvel de moda circular. O aplicativo será um ambiente que permitirá a artesãos e empresas da indústria têxtil buscar e descartar materiais usados ou pré-existentes, incluindo resíduos têxteis, roupas usadas e outros materiais que passarão por um crivo de avaliação.

O sistema proposto será composto por servidores independentes. Um dos servidores será responsável por fornecer os arquivos de interface da PWA por meio de um navegador da web, recebendo as solicitações dos usuários e respondendo com dados do outro servidor que hospedará o banco de dados da aplicação.

A versão do aplicativo móvel será desenvolvida utilizando o servidor que fornece a interface da PWA, com o objetivo de exibi-la em uma aplicação nativa simplificada, que poderá ser disponibilizada nas lojas de aplicativos dos sistemas operacionais *Android* e *IOS*. A interface da aplicação será fácil de usar e intuitiva, permitindo que o usuário se cadastre e realize a compra ou venda de suas peças ou materiais, fornecendo informações sobre descartes sustentáveis mais próximos da sua localização.

1.2 OBJETIVOS

Nesta etapa, deve se estabelecer de maneira precisa e concisa quais são os resultados esperados da pesquisa e quais aspectos serão abordados ao longo do trabalho.

1.2.1 Objetivo geral

Desenvolver uma aplicação mobile que viabilize aos usuários compra e venda de materiais para o processo de *upcycling* voltado para a moda circular, promovendo a economia criativa e o consumo consciente.

1.2.2 Objetivos específicos

Identificar se há aplicações disponíveis no mercado com formato semelhante ao proposto;

Fazer o levantamento de requisitos e a modelagem da aplicação;

Analisar a melhor forma de fazer a implementação da aplicação;

Realizar testes com usuários em potencial a fim de aperfeiçoar o aplicativo corrigindo possíveis falhas.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem como objetivo elucidar os principais conceitos referentes as abordagens de desenvolvimento de aplicações para dispositivos móveis e outros conceitos referentes a arquitetura da aplicação.

2.1 DISPOSITIVOS MÓVEIS

De acordo com Franco (2014) o final do século XX foi um período marcado por grandes mudanças em uma velocidade nunca vista antes no que tange fatores sociais como pensamentos, ideais e especialmente o modo de se viver, isso devido a diversos avanços, mas principalmente pelo crescimento da tecnologia da informação e o surgimento de novas tecnologias, sobretudo a tecnologia móvel, que passou por grandes mudanças nas últimas décadas, com componentes eletrônicos menores e mais eficazes, baterias de longa durabilidade e melhoria no acesso à internet.

Quase como antevendo o cenário atual, muito embora a tecnologia móvel e a comunicação via satélite já fossem uma realidade, em um estudo acerca da computação móvel Figueiredo e Nakamura (2003, p.1) afirmaram:

“Nota-se [...] uma grande evolução e popularização de dispositivos computacionais móveis, [...], que nos traz a estimativa de que em poucos anos milhares de pessoas espalhadas pelo mundo terão um desses tipos de dispositivos com a capacidade de comunicação com as redes fixas tradicionais e com outros computadores móveis.”

Durante a conferência *World Wide Developers* da *Apple*, em 2007, Steve Jobs fez a seguinte afirmação “[...] de vez em quando aparece um produto revolucionário que muda tudo”. Assim, pode-se dizer, foi o advento da internet para a computação de uma maneira geral, mas especialmente na indústria de software, com a integração da internet nos dispositivos móveis e o desenvolvimento da computação móvel. (FONTOLAN 2018)

Desde que os celulares passaram a entregar aos usuários experiências além da realização de chamadas e do envio de mensagens, tornou-se mais popular e mais acessível ocasionando uma transformação no mercado de dispositivos móveis. Em 2018 no Brasil, o número de *smartphones* em uso ultrapassou o número de pessoas. (FGVcia 2018). A 33ª Pesquisa Anual do FGVcia divulgada em maio de 2022, feita com dados coletados do início de 2022, revelou que no Brasil há 447 milhões de dispositivos digitais em uso, dessa totalidade 352 milhões são dispositivos portáteis, desses dispositivos, 242 milhões são *smartphones*, ou seja, mais de dois dispositivos digitais por habitante em junho de 2022. (IBGE 2022). Para Dolgaya e Phongtraychack (2018) a tecnologia móvel mudou totalmente o mundo atual e as possibilidades para o futuro das aplicações mobile são ilimitadas.

Essa popularidade se dá pelo fato de os dispositivos estarem cada vez mais sofisticados e ao mesmo tempo acessíveis. Uma vez que passaram a ser construídos com um hardware de melhor qualidade, possibilitando o desenvolvimento de sistemas operacionais avançados, o desenvolvimento de aplicativos, recursos e serviços também melhoraram, e os celulares passaram a proporcionar funcionalidades além das habituais, como entretenimento, acesso à informação, solução de problemas e atividades cada vez mais integradas ao cotidiano dos usuários. (SILVA E SANTOS 2014).

Segundo Martins et al. (2013) o mercado de tecnologia móvel está cada vez maior e proporcionando aos usuários uma gama de opções, é uma área tomada por diferentes plataformas tecnológicas, sistemas operacionais e plataformas de desenvolvimento de *software*. Para atender essa demanda de mercado com tantas possibilidades, há atualmente algumas abordagens de desenvolvimento aplicações e distribuição diferentes, cabendo aos desenvolvedores a tomada de decisão pela abordagem que melhor se adapta ao que lhe é proposto, podendo a aplicação ser nativa, *web mobile*, híbrida ou PWA. (MALAVOLTA 2016).

2.2 ABORDAGENS DE DESENVOLVIMENTO MOBILE

As aplicações nativas são desenvolvidas em plataformas específicas e não possuem bases de códigos interoperáveis, o que inviabiliza a reutilização do código entre outras aplicações tornando essa possibilidade quase inexistente, ou seja, o desenvolvedor que opta por essa aplicação, muito provavelmente se quiser disponibilizar o *app* para diversos sistemas operacionais e plataformas terá um trabalho custoso e demorado. Ao passo que no desenvolvimento de aplicações híbridas, elas possuem uma única base de códigos, podendo ser implantada em diversas plataformas, podendo ou não ser combinadas com as tecnologias *web*, permitindo o acesso a funcionalidades nativas através de bibliotecas e *plugins* auxiliares. (BIØRN-HANSEN, MAJCHRZAK e GRØNLI 2017).

Para os autores Heitkotter, Hanschke e Majchrzak (2012) há diversas plataformas de desenvolvimento de aplicações nativas, cada uma possui uma linguagem de programação e API's nativas que acabam dificultando o trabalho dos desenvolvedores, ainda assim são a opção ideal quando a aplicação exige um desempenho crítico, quando exige acesso a recursos específicos do celular (câmera, geolocalização, notificações *push*) e quando se faz necessário a integração do aplicativo a um hardware externo.

No entanto, para Cardieri e Zaina (2018), com o aprimoramento dessas abordagens, as aplicações nativas, *web mobile* e PWA's, segundo estudos recentes, proporcionam ao usuário uma experiência muito parecida entre si e satisfatória. Mas isso nem sempre foi uma realidade, num passado não muito distante podia se dizer que a plataforma *web* ficava um pouco atrasada em inovação de dispositivos móveis em termos de capacidade de competir com aplicativos nativos ou aplicativos híbridos. (PUDER et al., 2014).

As aplicações *web mobile* usam a abordagem WDR (*Web Design Responsivo*), são aplicações que podem ser acessadas em diferentes tipos de sistemas operacionais e dispositivos móveis com os mais diversos tamanhos de tela se adaptando conforme a necessidade do *hardware*. São aplicações acessadas diretamente do navegador do dispositivo móvel e não precisam ser instaladas por possuir a mesma base de um *website*, só podem ser acessadas

quando há conexão com a internet e o sistema operacional não tem qualquer controle da aplicação. (FONTOLAN 2018)

2.2.1 PWA (*Progressive web apps*)

No ano de 2015 a Google propôs uma abordagem que incluía recursos que não se faziam presentes nas *web apps* tradicionais, como suporte *offline*, sincronização em segundo plano e instalação da tela inicial na *web*. Essa abordagem é a chamada PWA, ou *Progressive Web Apps*, termo cunhado por Russel e Berriman. (BIØRN-HANSEN, MAJCHRZAK e GRØNLI 2017). De acordo com LePage (2017) a PWA une o melhor dos *web apps* e aplicativos nativos não precisando ser baixada em lojas de aplicativos. Malavolta (2016) afirma que as PWA's combinam vários métodos, procedimentos e API's que acabam proporcionando uma experiência muito semelhante ao de uma aplicação nativa, mas com código interoperável podendo ser disponibilizada em diversas plataformas e sistemas operacionais e usada no navegador ou como um aplicativo instalável.

Uma PWA é servida a partir de um servidor remoto via HTTPS que é uma vantagem sobre as aplicações nativas e híbridas devido à velocidade na distribuição de atualizações. O primeiro acesso a PWA é através do navegador, como *web app* padrão, até o momento em que o usuário decide instalar a PWA em seu dispositivo (a instalação é opcional) promovendo-a a um aplicativo móvel de nível superior, com suporte em tela cheia, sem a necessidade do uso de um navegador, com suporte *offline*, notificações *push* etc. (MALAVOLTA 2016). Conforme dito por LePage (2017) a PWA é uma aplicação inteligente, ao passo que ao se estabelecer uma relação com a aplicação ela se torna cada vez mais aperfeiçoada para o usuário, isso porque ela possui um manifesto que arquiva informações permitindo uma experiência imersiva em tela cheia.

Segundo Osmani (2015) há um conjunto de características e palavras-chave que definem uma PWA, sendo elas:

- Progressiva: disponível para todos os usuários independentemente da escolha do navegador;

- Responsiva: a interface se ajusta a qualquer tamanho de tela, possibilitando uma experiência visual confortável independente do dispositivo;
- Independente de conectividade: funcionam *offline* ou em redes de baixa qualidade por fazer uso da tecnologia de SW's (*service workers*);
- Semelhante a aplicações nativas: através do modelo *shell* de aplicativo fornece uma interação e navegação muito semelhante à de um *app* nativo;
- Atualizável: sempre se mantém atualizado, fazendo com que o usuário não precise baixar atualizações de tempos em tempos;
- Seguro: utilizam protocolo HTTPS;
- Detectável: são facilmente encontrados por ferramentas de busca devido por possuírem um manifesto e SW;
- Engajável: facilita o engajamento por meio de recursos como notificações *push*;
- Instalável: permite aos usuários salvar o *app* na tela inicial sem a necessidade de usar uma loja de aplicativos;
- Vinculável: possuem URL que pode ser compartilhada facilmente.

Essas características são as contribuições da PWA para a unificação da experiência móvel, isso é possível devido à API do SW, que capacita os desenvolvedores através do uso de um *script* executado em segundo plano, atuando como um *proxy* de rede e dispositivo. Conforme constatado por Fontolan (2018 p.36) “Os SW's são *scripts* executados no plano de fundo de uma aplicação *web* com o objetivo de atuar como intermediário entre o *app* e a internet, enviando mensagens ao aplicativo quando determinada tarefa foi cumprida.”

Isso significa que o funcionamento da API do SW é totalmente separado da interface do usuário permitindo maior velocidade no carregamento da PWA. Os SW's é que são encarregados por deixar parte da PWA armazenada na memória cache do navegador garantindo maior eficiência no carregamento e disponibilidade *offline* das aplicações. (SHEPPARD 2017)

Há vários *cases* de sucesso de aplicações tradicionais que aderiram a PWA, uma delas é o case do Pinterest, uma rede social de compartilhamento de fotos lançada em 2010 que em 2017 tomou a decisão de migrar para a PWA visando crescimento e uma melhor experiência para os usuários. A partir dessa tomada de decisão e da implantação da PWA cresceu o engajamento com o público, o tempo gasto pelos usuários na aplicação subiu e a receita através dos anúncios na aplicação também teve um aumento significativo. (OSMANI 2017)

Segundo relatório do próprio Pinterest, em apenas um ano da migração para a PWA o número de usuários semanais da *web* móvel subiu 103%, só no Brasil houve um aumento de 156%, na Índia esse crescimento foi de 312%. O tempo que os usuários passavam na aplicação aumentou 296%, o número de pins visualizados cresceu 401%, os *logins* 370% e o crescimento de novos usuários foi para 843%. No cenário anterior o Pinterest encontrava dificuldades em converter visitantes em novos usuários e na demora no carregamento da página e imagens para os usuários com acesso a redes móveis com velocidade limitada. (ARGYLE 2018)

2.3 BACK-END

O *back-end* é a parte de trás da aplicação, o processo interno, a parte não visível ao usuário, é o local onde é processado os dados compilados pelo *front-end*. O *back-end* é responsável pela regra de negócios, *web services* e API's de uma aplicação.

2.3.1 API (*Application Programming Interface*)

A API é uma interface que permite a troca de informações entre sistemas. De uma forma mais simplificada é responsável por receber os requisitos de uma aplicação, informá-las ao sistema e em seguida retornar com uma resposta para aplicação indicando se o processamento da requisição obteve ou não sucesso e retornando dados que sejam relevantes a aplicação. (MULESOFT 2015)

Andrade (2016) afirma que a viabilidade de fazer uma API se dá porque ao isolar a implementação de uma aplicação da API é possível ao desenvolvedor propiciar uma variedade de implementações para diversas tecnologias e dispositivos que permite aos usuários utilizar a aplicação, ainda que inicialmente não teriam essa possibilidade. A API facilita o processo de atualização e manutenção de um sistema justamente por ser isolada da implementação, considerando que ocorra algum problema no servidor a correção é feita na própria API, caso não houvesse uma API a correção deveria ser feita em cada distribuição da aplicação.

2.4 FRONT END

O *front-end* é a interface gráfica de uma aplicação, a parte na qual o usuário interage, segundo Souza (2016 p. 13) “O *front-end* é um conjunto de técnicas, utilizado em associação com ferramentas de desenvolvimento, que tem por objetivo criar e manipular a parte visual e estrutural das páginas *web*”. O *front-end* é a parte responsável por compilar as informações de entrada do usuário.

2.5 BANCO DE DADOS

Banco de dados é um agrupamento de dados que se relacionam entre si com os mais variados registros, esses grupos de dados organizados é o que armazena informações em um sistema, para manipulá-los é necessário um sistema de gerenciamento de banco de dados, segundo Korth (2006 p.1):

“Um sistema de gerenciamento de banco de dados é uma coleção de dados inter-relacionados e um conjunto de programas para acessar esses dados. A coleção de dados, normalmente chamada de banco de dados, contém informações relevantes [...]. O principal objetivo de um SGBD é fornecer uma maneira de recuperar informações de banco de dados que seja tanto conveniente quanto eficiente.”

3 METODOLOGIA

Seguindo as palavras de Demo (2000), a pesquisa é a atividade fundamental da ciência, que envolve a busca por soluções para questões ainda não resolvidas ou para esclarecimento de dúvidas ainda não elucidadas. Para Lakatos e Marconi (2010), a pesquisa é um conjunto de procedimentos organizados e sistemáticos, que se baseiam no raciocínio lógico e na utilização de métodos científicos para a resolução de problemas propostos. Seu objetivo é encontrar soluções para questões específicas, utilizando métodos científicos e procedimentos rigorosos.

3.1 CLASSIFICAÇÕES DE PESQUISA

Existem várias classificações de pesquisa, já que cada uma tem um objetivo específico e utiliza diferentes técnicas. Por exemplo, algumas pesquisas têm o objetivo de analisar o conhecimento atual sobre um determinado assunto, enquanto outras buscam estabelecer um modelo que explique o comportamento de um processo natural. Moresi (2003) destaca que existem outras categorias de pesquisa também. A seguir, serão apresentadas as principais classificações de pesquisa em termos de sua natureza, forma e objetivo.

3.1.1 Natureza da pesquisa

Pesquisa básica:

Segundo Gil (2010), a pesquisa básica, também conhecida como pesquisa pura ou fundamental, é uma investigação científica que busca gerar conhecimento novo e avançar na compreensão de fenômenos e teorias científicas. Essa pesquisa é geralmente realizada com o objetivo de expandir o conhecimento e a teoria em uma determinada área, sem a preocupação imediata com aplicações práticas.

Pesquisa aplicada:

Já para Lakatos e Marconi (2010), a pesquisa aplicada é aquela que busca solucionar um problema específico e imediato ou desenvolver novos produtos e processos que possam ser aplicados em situações práticas. Essa pesquisa geralmente é realizada em ambientes reais ou em situações de campo, com o objetivo de aplicar e testar conhecimentos científicos para resolver problemas concretos do mundo real.

3.1.2 Forma da pesquisa

Pesquisa qualitativa:

Para Bogdan e Biklen (1994), a pesquisa qualitativa é uma abordagem que busca entender e descrever a natureza dos fenômenos sociais a partir das perspectivas e experiências dos participantes envolvidos. Essa pesquisa é geralmente realizada em ambientes naturais e não controlados, e busca entender a complexidade dos fenômenos estudados a partir das perspectivas dos participantes.

Pesquisa quantitativa:

Segundo Gil (2010), a pesquisa quantitativa é uma abordagem de investigação científica que busca mensurar e analisar as relações entre variáveis, com o objetivo de identificar padrões e estabelecer generalizações. Essa pesquisa é geralmente realizada por meio de técnicas como questionários, experimentos e testes padronizados, e busca mensurar e analisar dados numéricos.

3.1.3 Objetivo da pesquisa

Quanto ao objetivo as pesquisas podem ser classificadas como: descritiva, exploratória, explicativa ou experimental. Abaixo segue a definição das pesquisas segundo Gil (2010).

Pesquisa descritiva: esta modalidade de pesquisa tem como objetivo descrever as características ou fenômenos de uma população ou fenômeno

estudado, sem manipulá-los. O objetivo primordial é descrever as características de determinada população ou fenômeno.

Pesquisa exploratória: é uma investigação inicial realizada sobre um tema pouco explorado, com o intuito de obter ideias e conhecimentos gerais sobre ele. O objetivo é proporcionar maior familiaridade com o problema, a fim de torná-lo mais explícito ou construir hipóteses.

Pesquisa explicativa: esta modalidade busca explicar a relação entre variáveis ou a causa e efeito de um fenômeno. O objetivo é identificar os fatores que determinam ou contribuem para a ocorrência dos fenômenos.

Pesquisa experimental: é um tipo de pesquisa que envolve a manipulação de variáveis independentes para observar seus efeitos sobre a variável dependente, com o intuito de estabelecer relações de causa e efeito. O objetivo é controlar e manipular as variáveis para verificar o efeito da manipulação sobre o fenômeno estudado.

3.2 APLICAÇÃO DA METODOLOGIA

A presente etapa consiste na utilização de um conjunto organizado e estruturado de métodos, técnicas e processos com o intuito de atingir um objetivo específico e/ou solucionar um problema determinado.

3.2.1 Planejamento da pesquisa

Para o desenvolvimento deste trabalho foi escolhido como abordagem a pesquisa qualitativa, de acordo com Denzin e Lincoln (2006), esta abordagem compreende uma interpretação do mundo não quantificável, o pesquisador estuda o objeto em questão e tenta decodificar os significados que a ele é atribuído. Ainda quanto a abordagem de pesquisa, Richardson (1999) aponta que a pesquisa qualitativa é propícia quando há a necessidade de compreensão de aspectos psicológicos cujos dados coletados não são mensuráveis à conta da sua complexidade, como motivações, valores, entre outros fatores.

Quanto a natureza da pesquisa ela é aplicada, segundo Gil (2010) a pesquisa aplicada envolve estudos que tem como objetivo solucionar questões sociais no ambiente em que os pesquisadores se encontram. Em conformidade com Jacobsen (2009) a pesquisa aplicada é aquela que busca soluções para um problema existente.

No que se refere aos objetivos esta é uma pesquisa descritiva. Segundo Silva & Menezes (2000, p.21), “a pesquisa descritiva visa descrever as características de determinada população ou fenômeno ou o estabelecimento de relações entre variáveis. Envolve o uso de técnicas padronizadas de coleta de dados: questionário e observação sistemática. Assume, em geral, a forma de levantamento”.

O procedimento usado para o levantamento de dados é a revisão bibliográfica, que para Boccato (2006, p. 266):

[...] busca a resolução de um problema (hipótese) por meio de referenciais teóricos publicados, analisando e discutindo as várias contribuições científicas. Esse tipo de pesquisa trará subsídios para o conhecimento sobre o que foi pesquisado, como e sob que enfoque e/ou perspectivas foi tratado o assunto apresentado na literatura científica. Para tanto, é de suma importância que o pesquisador realize um planejamento sistemático do processo de pesquisa, compreendendo desde a definição temática, passando pela construção lógica do trabalho até a decisão da sua forma de comunicação e divulgação.

3.2.2 Atividades

Com base na metodologia adotada para este trabalho, foi elaborada uma sequência de atividades para organizar os envolvidos na execução do projeto. A primeira etapa envolveu um estudo preliminar do tema e uma avaliação da viabilidade do projeto, bem como a criação de um cronograma geral de atividades para organização do projeto. Na etapa seguinte, foram definidos o escopo inicial, os objetivos do trabalho e a sua relevância. Em seguida, foram estabelecidas as bases de artigos e o tipo de pesquisa a ser realizada. A partir dessas definições, o referencial teórico e o suporte tecnológico foram desenvolvidos para o aplicativo. Na etapa seguinte, foi planejado o

desenvolvimento de software, incluindo a definição da arquitetura e a adoção de boas práticas de desenvolvimento. A elaboração da monografia foi conduzida simultaneamente às outras atividades necessárias no trabalho.

3.3 METODOLOGIA DE GERENCIAMENTO DE PROJETOS

Conforme os empreendimentos relacionados à tecnologia da informação se tornam cada vez maiores e mais complexos, as metodologias de desenvolvimento de *software* e suas ferramentas de suporte se tornam fatores cruciais para alcançar o sucesso nos projetos (ASADI; RAMSIN, 2009). Com a necessidade de um desenvolvimento mais ágil, capaz de adaptação, inovação, melhoria contínua e com recursos limitados, o contexto em que esses empreendimentos se encontram fortaleceu a área de gerenciamento de projetos, que busca organizar e administrar esses com o auxílio das metodologias de desenvolvimento (MAURER, 2005).

As metodologias de desenvolvimento tradicionais são consideradas rígidas e não fornecem suporte adequado para projetos de desenvolvimento de *software* modernos (ASADI; RAMSIN, 2009). Em contrapartida, as metodologias ágeis de desenvolvimento visam construir *software* de maneira mais produtiva, adotando uma abordagem diferente das metodologias tradicionais e estimulando a utilização de práticas e princípios específicos (TAVARES, 2008).

Nesse contexto, encontra-se o *Scrum*, uma abordagem enxuta de gerenciamento de projetos desenvolvida por Jeff Sutherland em 1993, juntamente com Mike Beedle e Ken Schwaber, e é baseada em um artigo de Takeuchi e Nonaka (1986) que destaca as vantagens dos pequenos times no desenvolvimento de produtos. Embora originalmente tenha sido criada para o desenvolvimento de *software*, atualmente é amplamente aplicada a diversos tipos de projetos. (CRISTAL; WILDT; PRIKLADNICKI, 2008).

Scrum é uma metodologia ágil amplamente utilizada no gerenciamento de projetos de desenvolvimento de *software*, que enfatiza o trabalho em equipe, a colaboração e o progresso iterativo em direção a metas bem definidas.

O *framework Scrum* é composto pelos seguintes componentes: *Product Backlog*, que é uma lista priorizada de recursos e requisitos mantida pelo Dono do *Produto*; *Sprint*, um período de tempo definido durante o qual a equipe trabalha em um conjunto de itens pendentes; *Sprint Backlog*, uma lista dos itens do *Product Backlog* que a equipe se comprometeu a entregar durante o *Sprint*; *Daily Scrum*, uma reunião diária em que a equipe compartilha o progresso, discute os obstáculos e planeja as próximas 24 horas; *Sprint Review*, uma reunião no final de cada *Sprint* em que a equipe demonstra o trabalho concluído e recebe feedback; e *Sprint Retrospective*, uma reunião em que a equipe reflete sobre o *Sprint* anterior e identifica oportunidades de melhoria.

O *Scrum* é baseado nos princípios de transparência, inspeção e adaptação, e foi projetado para ser flexível e adaptável a requisitos e circunstâncias em constante mudança. A estrutura *Scrum* fornece uma estrutura para que as equipes colaborem de forma eficaz e entreguem valor aos seus stakeholders de maneira iterativa e incremental.

Entre os benefícios do *Scrum* estão a maior produtividade, melhor comunicação e colaboração, melhor qualidade e menor tempo de lançamento no mercado. No entanto, a implementação bem-sucedida do *Scrum* requer um forte comprometimento da equipe, comunicação clara e liderança eficaz.

3.3.1 Desenvolvimento do aplicativo uCycle

Para o desenvolvimento do presente trabalho foi utilizada como metodologia de gerenciamento de projetos a metodologia *Scrum*.

3.3.1.1 Papéis

A equipe de desenvolvimento consiste em duas pessoas, foi decidido que o papel de *Scrum Master* e *Product Owner* seria dividido e atribuído a cada um dos membros.

4 DESENVOLVIMENTO

Decidiu-se por desenvolver uma plataforma de *e-commerce*, um sistema online que permite que empresas ou indivíduos vendam produtos voltados para moda circular. Essa plataforma inclui uma variedade de recursos essenciais para o funcionamento de uma loja virtual, como um catálogo de produtos, carrinho de compras, processamento de pagamentos, integração com sistemas de envio e entrega, entre outros.

O uCycle fornece uma interface de administração para que os usuários possam configurar e gerenciar sua loja *online*. Isso inclui adicionar e atualizar produtos, definir preços, criar promoções, gerenciar pedidos e acompanhar o desempenho das vendas.

4.1 REQUISITOS

Os requisitos foram escritos em forma de *User Stories*, uma vez que a metodologia de desenvolvimento adotada foi o *Scrum*. A seguir a tabela 2 com os requisitos do projeto.

Tabela 1 - *User Stories* (Histórias de usuário) previamente elicitados do projeto

#	<i>User Story</i> (História de usuário)
US001	Cadastro como vendedor: Como usuário, quero poder me cadastrar na plataforma de <i>e-commerce</i> , fornecendo informações sobre minha empresa e produtos, para começar a vender online.
US002	Adicionar produtos: Como usuário, quero poder adicionar meus produtos à plataforma de <i>e-commerce</i> , incluindo informações como título, descrição, preço, imagens e categorias, para disponibilizá-los para compra.
US003	Gerenciar estoque: Como usuário, quero poder controlar o estoque dos meus produtos na plataforma, atualizando a quantidade disponível

	automaticamente à medida que as vendas são feitas, para evitar vendas de produtos esgotados.
US004	Definir políticas de envio: Como usuário, quero poder definir as políticas de envio, como regiões atendidas, métodos de entrega e prazos de entrega estimados, para informar os compradores sobre as opções disponíveis.
US005	Receber notificações de vendas: Como usuário, quero receber notificações sempre que um produto meu for vendido, incluindo informações sobre o cliente, endereço de entrega e detalhes do pedido, para poder processar a venda e preparar o envio.
US006	Gerenciar pedidos: Como usuário, quero poder visualizar e gerenciar meus pedidos, incluindo informações sobre os produtos vendidos, status do pedido e detalhes do cliente, para garantir uma experiência de compra eficiente.
US007	Atualizar informações dos produtos: Como usuário, quero poder atualizar as informações dos meus produtos, como descrição, preço e imagens, para mantê-los atualizados e atrativos para os compradores.
US008	Acesso a relatórios de vendas: Como usuário, quero ter acesso a relatórios de vendas, incluindo informações como volume de vendas, produtos mais populares e histórico de vendas, para analisar o desempenho do meu negócio.
US009	Pesquisa de produtos: Como usuário, quero pesquisar produtos por nome, categoria ou palavra-chave para encontrar os itens que desejo comprar.
US010	Visualização de detalhes do produto: Como usuário, quero poder visualizar os detalhes de um produto, incluindo descrição, preço, imagens e avaliações, para tomar uma decisão de compra informada.

US011	Adicionar produto ao carrinho: Como usuário, quero poder adicionar um produto ao meu carrinho de compras, para poder comprar vários itens ao mesmo tempo.
US012	Gerenciar carrinho de compras: Como usuário, quero poder visualizar meu carrinho de compras, adicionar, remover ou atualizar a quantidade de produtos, para ajustar minha seleção antes de finalizar a compra.
US013	Finalizar compra: Como usuário, quero poder finalizar minha compra, fornecendo informações de envio e pagamento, para efetuar a transação e receber meus produtos.
US014	Rastreamento de pedido: Como usuário, quero poder rastrear o status do meu pedido, para saber quando ele será entregue.
US015	Avaliação de produtos: Como usuário, quero poder avaliar os produtos que comprei, para compartilhar minha opinião e ajudar outros usuários na escolha de produtos.
US016	Gerenciamento de perfil: Como usuário, quero poder criar e gerenciar meu perfil na plataforma, incluindo informações de contato, endereços de entrega e histórico de compras.
US017	Recomendações personalizadas: Como usuário, quero receber recomendações personalizadas de produtos com base em meu histórico de compras e preferências, para descobrir novos itens de interesse.
US018	Suporte ao cliente: Como usuário, quero ter acesso fácil ao suporte, seja por <i>chat</i> , <i>e-mail</i> ou telefone, para obter ajuda ou resolver problemas relacionados às minhas compras e esclarecer dúvidas sobre a plataforma de <i>e-commerce</i> .

Abaixo, os requisitos não funcionais que especificam as características e restrições do sistema, esses são essenciais para garantir que o sistema atenda às necessidades dos usuários e funcione de forma adequada em seu ambiente.

Tabela 2 - Requisitos Não Funcionais

#	Requisitos não funcionais
RNF01	Usabilidade: O sistema deve ser intuitivo e fácil de usar e acessível para uma ampla variedade de usuários com níveis de conhecimento diferentes. Ele deve fornecer uma experiência de compra fluida, com navegação clara, processo de <i>checkout</i> simples, busca simples de produtos e adição de produtos bem explicada e objetiva.
RNF02	Confiabilidade: O sistema deve ser confiável e disponível, ou seja, ele deve ser capaz de lidar com uma carga significativa de usuários simultâneos sem falhas.
RNF03	Desempenho: O sistema deve ter baixa latência, que garantem tempos de resposta curtos mesmo durante períodos de tráfego intenso. As páginas do <i>site</i> devem ser carregadas rapidamente, e as transações devem ser processadas de forma eficiente, aumentando a retenção e a satisfação dos clientes.
RNF04	Segurança: A segurança é fundamental para um <i>e-commerce</i> . O sistema deve garantir a proteção dos dados pessoais dos clientes como informações de pagamento. O sistema deve implementar autenticação adequada e medidas para prevenir ataques e roubo de contas.
RNF05	Escalabilidade: O sistema deve ser escalável, isto é, capaz de lidar com o crescimento do número de usuários e transações ao longo do tempo.
RNF06	Integração: O sistema deve integrar-se com sistemas de pagamento, serviços de envio, entre outros, sendo assim necessário que o sistema suporte integração fácil e confiável com essas soluções externas e que estas integrações sejam facilmente substituíveis.
RNF07	Manutenibilidade: O sistema deve ser facilmente mantido, estendido e atualizado. Isso envolve uma arquitetura modular, código limpo e de fácil compreensão, além de ferramentas de monitoramento e

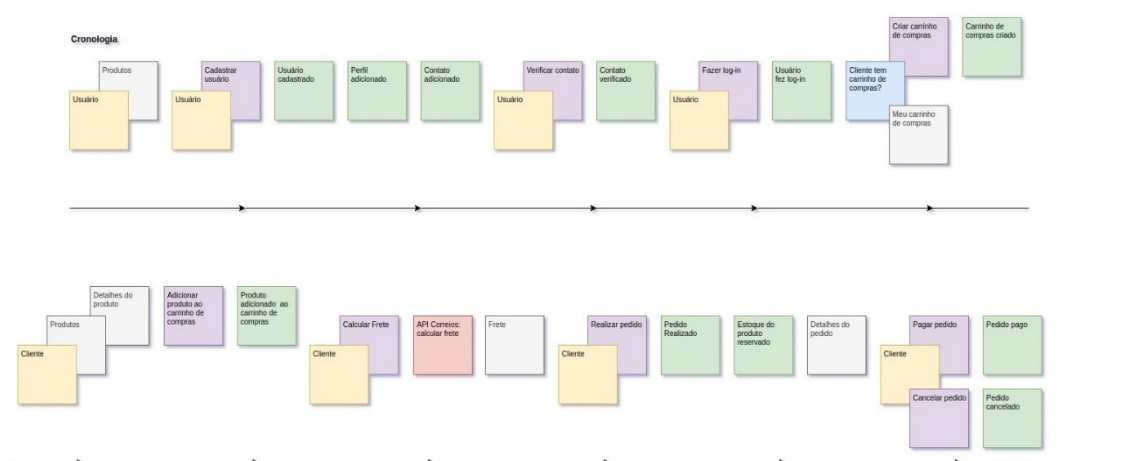
	registro de erros para facilitar a identificação e resolução de problemas.
RNF08	Conformidade regulatória: O sistema deve estar em conformidade com as leis e regulamentações aplicáveis, como a proteção de dados e direitos do consumidor. Ele deve aderir a padrões e práticas recomendadas para garantir a privacidade e segurança dos usuários.

4.2 DIAGRAMAS

A seguir estão dispostos os três diagramas usados na elaboração do trabalho: *Event Storming* usado para mapear eventos e fluxos em um sistema de negócios; Diagrama de Modelo de Domínio, representação visual dos conceitos e relacionamentos fundamentais de um domínio de negócio; e o Diagrama de Arquitetura, representação visual da estrutura e organização de um sistema de *software*.

4.2.1 Event Storming

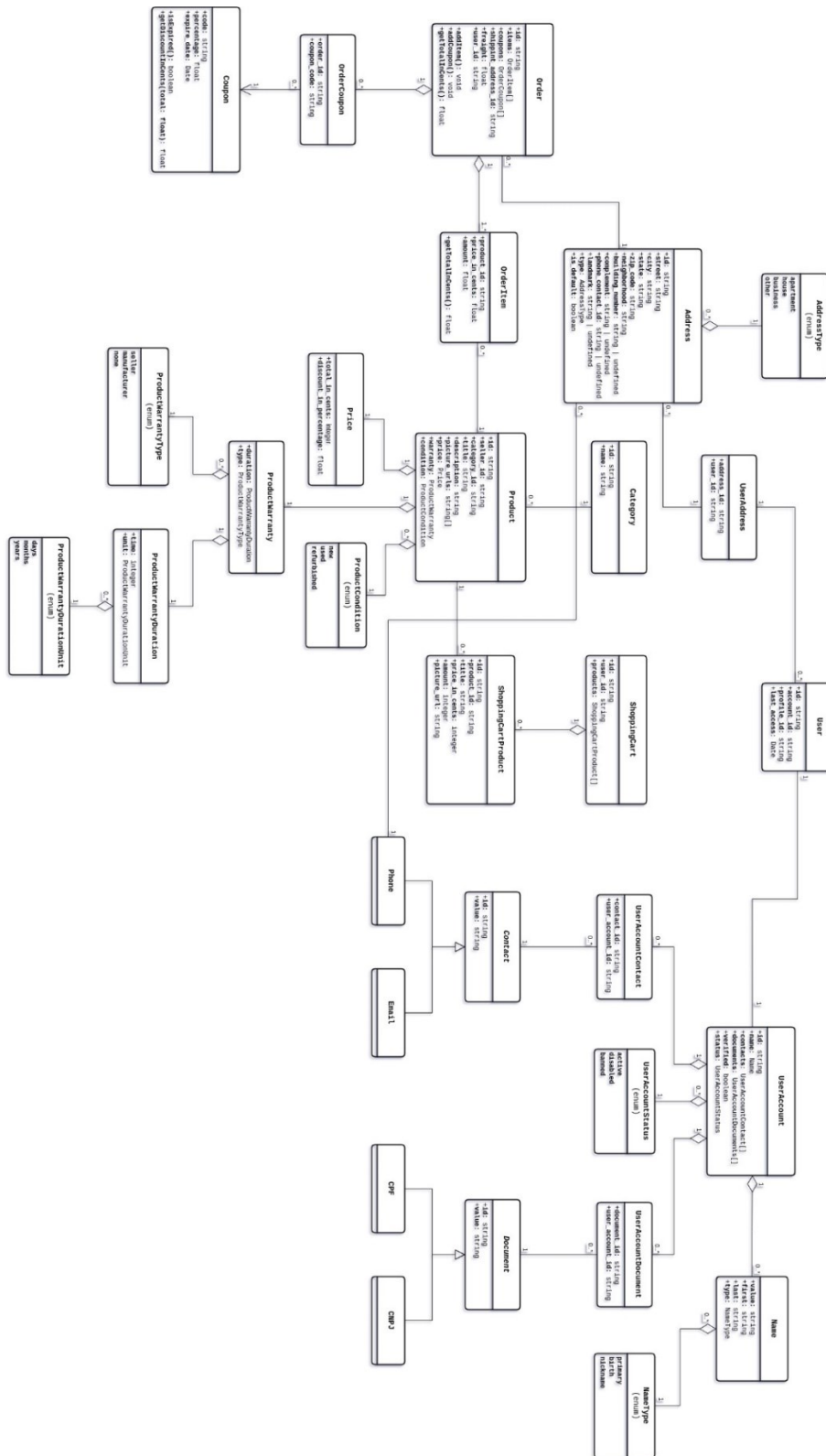
Figura 1 - Event Storming



Fonte: Elaborada pelos autores (2023)

4.2.2 Diagrama de modelo de domínio

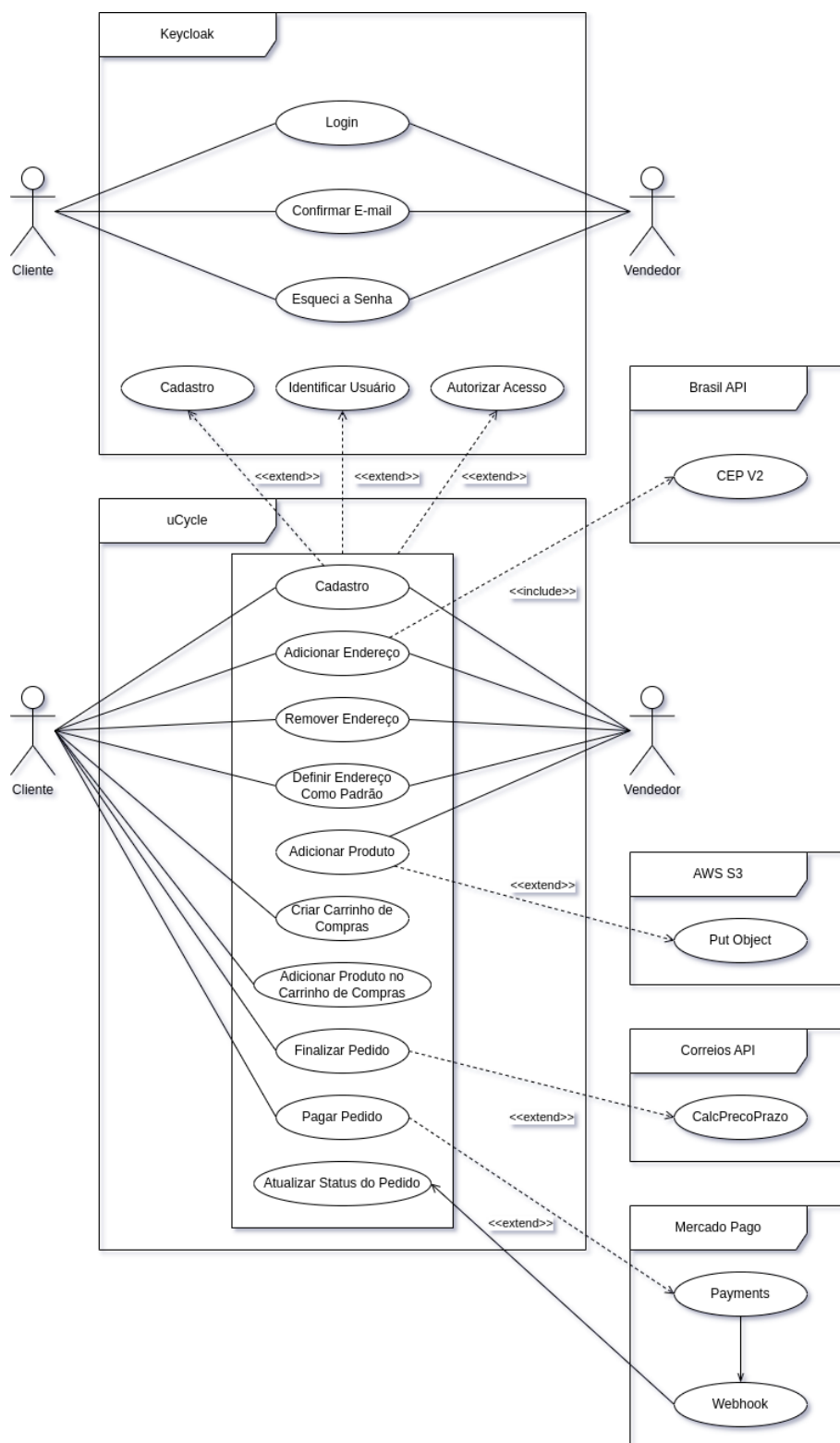
Figura 2 - Diagrama de modelo de domínio



Fonte: Elaborada pelos autores (2023)

4.2.3 Diagrama de caso de uso

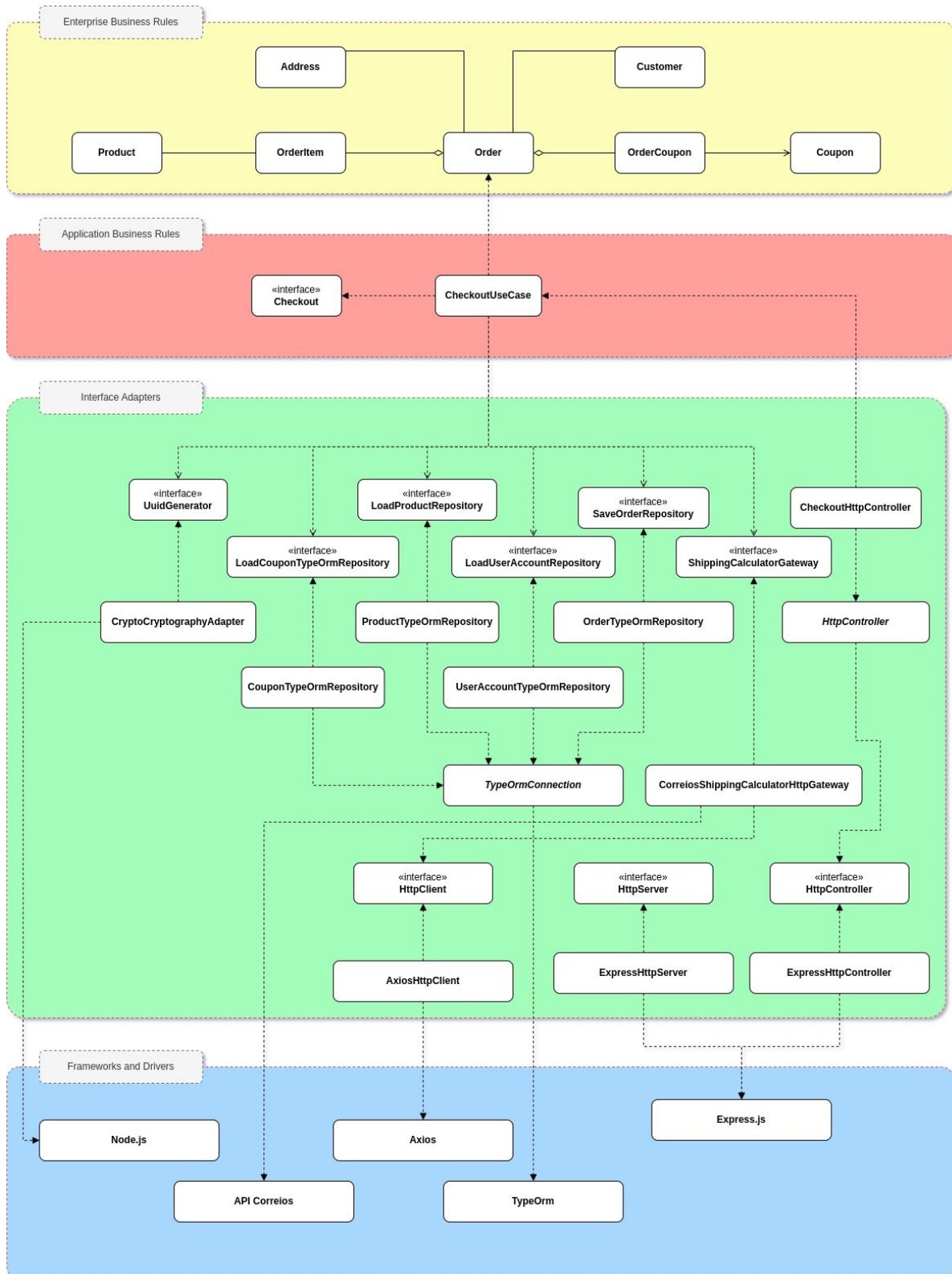
Figura 3 - Diagrama de caso de uso



Fonte: Elaborada pelos autores (2023)

4.2.4 Diagrama de arquitetura

Figura 4 - Diagrama de arquitetura



Fonte: Elaborada pelos autores (2023)

4.3 PADRÕES DE PROJETO

Padrões de projeto são soluções comprovadas e reutilizáveis para problemas recorrentes no *design* e na arquitetura de *software*, eles capturam as melhores práticas e o conhecimento acumulado ao longo do tempo por especialistas em engenharia de *software*. Os padrões de projeto fornecem uma linguagem comum e compartilhada entre os desenvolvedores, permitindo que eles se comuniquem efetivamente sobre soluções para problemas específicos e descrevem uma estrutura genérica que pode ser adaptada e aplicada a diferentes contextos. Há três principais categorias em que padrões de projeto são divididos, sendo essas: padrões de criação, padrões estruturais e padrões comportamentais.

Os padrões de criação estão relacionados à criação de objetos e abstraem o processo de criação para facilitar a flexibilidade e a reutilização. Exemplos incluem o padrão *Factory*, o padrão *Singleton* e o padrão *Builder*. Já os padrões estruturais lidam com a composição de classes e objetos para formar estruturas maiores e mais complexas, eles ajudam a garantir que as classes e os objetos sejam organizados de maneira eficiente e flexível. Exemplos incluem o padrão *Adapter*, o padrão *Decorator* e o padrão *Composite*. Por fim, os padrões comportamentais estão relacionados ao comportamento das classes e objetos e como eles interagem entre si. Eles ajudam a definir a comunicação e o fluxo de controle entre as entidades do sistema. Exemplos incluem o padrão *Observer*, o padrão *Strategy* e o padrão *Command*.

Os padrões de projeto fornecem diretrizes e soluções testadas para problemas comuns no desenvolvimento de *software*, permitindo que os desenvolvedores criem sistemas mais flexíveis, extensíveis e de fácil manutenção. De acordo com os autores Gamma, Helm Johnson e Vlissides, a razão pela qual padrões de projeto são importantes é que eles capturam a experiência e o conhecimento especializado de desenvolvedores experientes para resolver problemas comuns de *design* de *software*. Padrões de projeto oferecem uma linguagem comum e uma abordagem padronizada para projetar soluções elegantes e reutilizáveis, promovendo a manutenibilidade, a flexibilidade e a extensibilidade do *software*. Eles fornecem uma base sólida para

a construção de sistemas robustos, permitindo que os desenvolvedores se concentrem em resolver problemas específicos sem ter que reinventar a roda a cada vez.

4.3.1 S.O.L.I.D.

Além dos padrões de projeto, existem princípios gerais que são frequentemente seguidos durante a implementação de um projeto de *software*. Esses princípios ajudam a orientar a estrutura, o *design* e a qualidade do código. Um dos princípios mais conhecidos são os princípios SOLID que formam um conjunto de diretrizes para o desenvolvimento de *software* de alta qualidade e de fácil manutenção. O SOLID foi desenvolvido pelo autor Robert C. Martin que destaca a importância de seguir esses princípios durante a implementação do projeto, possibilitando um código mais modular, flexível, testável e de fácil manutenção, promovendo a qualidade geral do *software*. O SOLID consiste em:

Princípio da Responsabilidade Única (*Single Responsibility Principle* - SRP): "Uma classe deve ter apenas uma razão para mudar." - Robert C. Martin. Esse princípio enfatiza que uma classe deve ter apenas uma responsabilidade e motivo para mudar, promovendo a coesão e facilitando a manutenção.

Princípio do Aberto/Fechado (*Open/Closed Principle* - OCP): "Entidades de *software* (classes, módulos, funções, etc.) devem estar abertas para extensão, mas fechadas para modificação." - Robert C. Martin. Esse princípio incentiva a criação de código que possa ser estendido para adicionar novos comportamentos sem a necessidade de alterações nas partes existentes.

Princípio da Substituição de Liskov (*Liskov Substitution Principle* - LSP): "Seja S um subtipo de T, então os objetos do tipo T podem ser substituídos por objetos do tipo S sem quebrar a integridade do programa." - Robert C. Martin. Esse princípio destaca a importância de garantir que os subtipos possam ser usados de forma consistente em relação aos tipos base, sem introduzir comportamentos inesperados ou violar contratos.

Princípio da Segregação de Interfaces (*Interface Segregation Principle* - ISP): "Clientes não devem ser forçados a depender de interfaces que não

utilizam." - Robert C. Martin. Esse princípio defende que interfaces devem ser específicas e direcionadas às necessidades dos clientes, evitando interfaces monolíticas e desnecessárias.

Princípio da Inversão de Dependência (*Dependency Inversion Principle* - DIP): "Módulos de alto nível não devem depender de módulos de baixo nível. Ambos devem depender de abstrações." - Robert C. Martin. Esse princípio incentiva a dependência em abstrações, permitindo que as partes de um sistema possam evoluir independentemente, facilitando a substituição e a extensibilidade.

4.3.2 Clean Code

Clean Code é um conceito e uma abordagem para escrever código limpo, bem estruturado de fácil compreensão e manutenção. A capacidade de entender o código é de extrema importância, a colocação incorreta de uma vírgula pode custar tempo, dinheiro ou até vidas. Um código limpo possui várias características, as principais são: escolher nomes significativos, a escolha de nomes descritivos que revelem a intenção por trás do código é importante e pode fazer uma grande diferença na compreensão, para Robert C. Martin, é melhor um nome longo e descritivo do que um nome curto enigmático; quanto as funções e métodos deve-se manter as funções curtas, com apenas um nível de abstração e seguir o princípio da responsabilidade única, funções bem projetadas são pequenas, fáceis de ler e compreender; O código deve ser autoexplicativo, mas sempre que necessário é interessante usar comentários para explicar a intenção do código, evitando informações desnecessárias e mantendo-os sempre atualizados; e por fim os testes unitários para verificar a correteza do código e garantir que ele funcione conforme o esperado. Testes bem escritos são fundamentais para manter o código limpo e seguro durante o processo de refatoração. De acordo com Robert C. Martin (2008 pag.175)

"A maior parte do custo de um projeto de software está na manutenção de longo prazo. Para minimizar o potencial de defeitos à medida que introduzimos mudanças, é fundamental que possamos entender o que um sistema faz. À medida que os sistemas se tornam mais complexos, eles levam cada vez mais tempo para um desenvolvedor entender, e há uma oportunidade cada vez maior para um mal-entendido. Portanto,

o código deve expressar claramente a intenção de seu autor. Quanto mais claro o autor puder tornar o código, menos tempo os outros terão para entendê-lo. Isso reduzirá os defeitos e diminuirá o custo de manutenção.”

4.4 PADRÕES DE ARQUITETURA

Padrões de arquitetura são soluções reutilizáveis e abstratas para problemas recorrentes encontrados durante o projeto e desenvolvimento de sistemas de software. Esses padrões fornecem diretrizes e estruturas organizacionais que visam a construção de sistemas eficientes, escaláveis e de fácil manutenção. São derivados de boas práticas, experiências anteriores e princípios sólidos de projeto. Eles abrangem aspectos como a estrutura geral do sistema, a distribuição de responsabilidades, o fluxo de dados, a interação entre componentes, a modularidade e a separação de preocupações.

Além disso, os padrões de arquitetura promovem a reutilização de soluções previamente testadas e validadas, o que resulta em economia de tempo e esforço durante a implementação dos sistemas. Ao adotar abordagens padronizadas, é possível obter sistemas mais confiáveis, flexíveis e adaptáveis, que podem se beneficiar das lições aprendidas e melhores práticas acumuladas ao longo do tempo. A seguir estão descritas as abordagens utilizadas no desenvolvimento do presente projeto.

4.4.1 DDD (*Domain-Driven Design*)

Domain-Driven Design (DDD), ou "Projeto Orientado a Domínio", é uma abordagem de *design* de *software* que se concentra no domínio do problema em questão, buscando compreender e modelar o conhecimento do negócio em um sistema de *software*, foi desenvolvido por Eric Evans e se tornou uma referência importante na área de desenvolvimento de *software*. O DDD coloca o foco principal no domínio, que é a área específica do conhecimento em que um *software* está inserido.

Ele propõe que as equipes de desenvolvimento colaborem de perto com especialistas do domínio, como clientes, usuários finais e especialistas de negócio, para entender e modelar adequadamente o domínio dentro do código. Essa abordagem busca criar *software* que reflita de forma mais precisa e efetiva o domínio do problema. Ele promove uma melhor compreensão do negócio, uma modelagem mais clara e uma maior flexibilidade para a evolução do *software* ao longo do tempo. Alguns dos conceitos do DDD incluem:

Linguagem Ubíqua (O DDD incentiva o uso de uma linguagem compartilhada entre os especialistas do domínio e a equipe de desenvolvimento. Essa linguagem comum ajuda a alinhar a compreensão e facilita a comunicação entre as partes interessadas);

Modelagem do Domínio (O DDD propõe a criação de um modelo rico e expressivo do domínio, refletindo o conhecimento e as regras de negócio. Esse modelo se torna o núcleo do *software*, onde as entidades, agregados, valor de objetos e serviços são definidos);

Bounded Contexts, ou Contextos Delimitados (O domínio pode ser complexo e abrangente, portanto, o DDD sugere a criação de Contextos Delimitados que separam e isolam partes específicas do domínio. Isso ajuda a gerenciar a complexidade e a facilitar o desenvolvimento e a manutenção do *software*);

Agregados (Agregados são grupos de objetos relacionados que são tratados como uma unidade coesa. Eles garantem a consistência e a integridade das regras de negócio dentro de seus limites, controlando o acesso e a modificação dos objetos internos);

Serviços de Domínio (Serviços de Domínio são responsáveis por operações que não pertencem a uma única entidade ou agregado, mas estão relacionadas ao domínio como um todo. Eles encapsulam a lógica complexa e fornecem uma interface clara para a interação com o domínio).

Em entrevista o autor Vaughn Vernon questionado sobre a relevância do *Clean Code* aponta que o *Domain-Driven Design* (DDD) está se tornando cada vez mais relevante e sua adoção continua em ascensão. Com mais de 15 anos

de existência, o DDD transcendeu o status de uma tendência passageira. A relevância contínua dessa abordagem é impulsionada pelo cansaço generalizado dos arquitetos e desenvolvedores diante das falhas decorrentes de decisões arbitrárias em suas arquiteturas e implementações. Com base em diversas experiências, observa-se amplamente o reconhecimento de que uma arquitetura e um design de *software* sólidos não são alcançados sem um esforço dedicado.

4.4.2 Clean Architecture

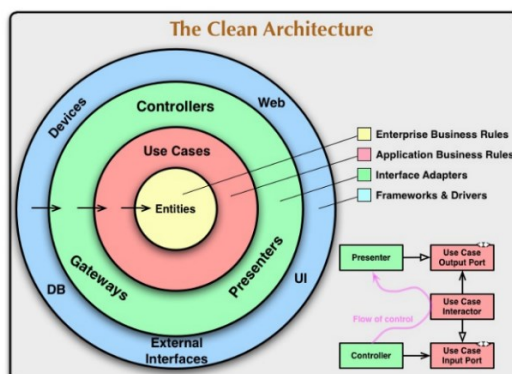
A *Clean Architecture* é um estilo de arquitetura de *software* proposto por Robert C. Martin. Essa abordagem visa criar sistemas independentes de *frameworks*, testáveis, de fácil manutenção e com regras de negócio bem definidas. A principal ideia por trás da *Clean Architecture* é estabelecer uma separação clara de responsabilidades em diferentes camadas do sistema.

Essas camadas são organizadas hierarquicamente, com a camada mais interna contendo as regras de negócio central do sistema, e as camadas externas lidando com detalhes de infraestrutura, como *frameworks*, bancos de dados e interfaces de usuário. A arquitetura segue o princípio da inversão de dependência (*Dependency Inversion Principle*), onde os componentes de alto nível dependem de abstrações em vez de componentes de baixo nível. Isso permite que as camadas internas do sistema não sejam afetadas por mudanças na infraestrutura externa, facilitando a manutenção e evolução do sistema ao longo do tempo.

Uma das principais características da *Clean Architecture* é a separação entre as regras de negócio do sistema (camada de domínio ou entidades) e os detalhes técnicos de implementação. Essa separação permite testar facilmente as regras de negócio independentemente da infraestrutura, resultando em um código mais limpo, modular e com maior capacidade de reutilização. A *Clean Architecture* propõe uma estrutura de projeto que prioriza a clareza, manutenibilidade e testabilidade do código, evitando a dependência excessiva de *frameworks* externos e facilitando a evolução do sistema ao longo do tempo.

Como Robert C. Martin enfatiza, a arquitetura é sobre intenção, separando o que muda do que não muda. É sobre estabelecer um conjunto de regras e restrições que guiam e moldam a evolução do sistema ao longo do tempo.

Figura 5 - A Arquitetura Limpa



Fonte: (Martin, 2012)

4.4.3 Event Driven Architecture

A Arquitetura Orientada a Eventos (*Event Driven Architecture* - EDA) é um estilo arquitetural proposto para a construção de sistemas flexíveis, escaláveis e adaptáveis às mudanças. Ela se concentra na comunicação e no processamento de eventos como elementos centrais do sistema. Em uma arquitetura orientada a eventos, os componentes se comunicam por meio da produção, detecção e consumo de eventos, que representam a ocorrência de algo significativo no sistema.

Essa abordagem busca desacoplar e distribuir as responsabilidades entre os componentes, permitindo que eles sejam independentes e reativos. Através da troca de eventos, os componentes podem interagir de forma assíncrona, lidando com sistemas complexos e distribuídos de maneira eficiente. Isso proporciona escalabilidade e resiliência, possibilitando a adição, remoção ou substituição de componentes de forma flexível.

A *Event Driven Architecture* também facilita a integração de sistemas legados, permitindo que diferentes partes do sistema se comuniquem através de

eventos, independentemente de suas tecnologias ou linguagens de programação. No entanto, sua implementação apresenta desafios, como garantir a ordem dos eventos, gerenciar fluxos complexos e lidar com possíveis falhas na comunicação.

Conforme Greg Young, um defensor da EDA, essa abordagem permite que as organizações respondam de forma eficiente a eventos e descubram *insights* valiosos a partir dos dados gerados por eles. Ele destaca que a EDA modela sistemas em torno do comportamento e das interações, priorizando a estruturação em vez da estrutura de dados, ainda ressalta que é importante projetar sistemas orientados a eventos desde o início, em vez de adaptar arquiteturas existentes.

Em suma, a *Event Driven Architecture* é um estilo arquitetural que promove a comunicação e o processamento de eventos como base para a construção de sistemas flexíveis, escaláveis e reativos. Ela permite que os sistemas sejam adaptáveis às mudanças, respondendo eficientemente aos eventos e explorando os *insights* gerados por eles, como destacado por Greg Young.

4.4.4 C.Q.R.S. (Command Query Responsibility Segregation)

O C.Q.R.S. (*Command Query Responsibility Segregation*) é um padrão arquitetural que separa as operações de escrita e leitura em um sistema, permitindo otimizar cada uma independentemente. Os comandos são responsáveis por alterar o estado do sistema, enquanto as consultas são responsáveis por recuperar informações. Essa abordagem reconhece que as operações de escrita e leitura têm requisitos distintos em termos de desempenho, escalabilidade, consistência e modelagem de dados. A separação em camadas distintas possibilita a aplicação de técnicas e tecnologias adequadas a cada tipo de operação, resultando em um sistema mais eficiente e escalável.

Embora o C.Q.R.S. ofereça benefícios significativos, sua implementação também apresenta desafios. A sincronização entre os modelos de escrita e

leitura e a manutenção da consistência dos dados podem ser complexas. Portanto, é fundamental realizar uma avaliação cuidadosa ao aplicar o C.Q.R.S., levando em consideração as características e necessidades específicas do sistema em questão.

Bertrand Meyer, renomado autor e acadêmico, é conhecido por suas contribuições à engenharia de *software*, incluindo princípios de projeto sólidos. Embora Meyer não seja amplamente associado ao C.Q.R.S., sua visão sobre a construção de *software* está alinhada com os princípios fundamentais desse padrão arquitetural. Ele destaca a importância da separação de responsabilidades e da ênfase na modularidade, promovendo a criação de sistemas sustentáveis e adaptáveis capazes de lidar com mudanças futuras. Seus princípios, como responsabilidade única e aberto/fechado, contribuem para a modularidade e facilitam a manutenção do código ao longo do tempo.

Essa abordagem arquitetural proporciona benefícios significativos, mas é essencial considerar os desafios associados, como a sincronização e a consistência dos dados, ao aplicá-la em um sistema específico.

4.5 TECNOLOGIA E INFRAESTRUTURA

O aplicativo uCycle foi desenvolvido empregando diversas tecnologias e ferramentas com o objetivo de oferecer uma experiência semelhante a um aplicativo nativo em dispositivos *Android* e *iOS*, assim como uma opção de instalação por meio do navegador e hospedagem do *back-end* na plataforma *AWS*.

No que diz respeito à plataforma *Android*, foi utilizada a base do *Android* SDK para criar a "casca" do aplicativo, permitindo sua instalação por meio da *Play Store*. *Java* foi a linguagem de programação usada, em virtude de suas vantagens significativas no desenvolvimento de aplicativos móveis, como interoperabilidade com *Kotlin*, segurança, concisão, redução de código *boilerplate*, suporte à programação assíncrona e um ecossistema ativo. O *Java* é uma linguagem moderna e versátil, amplamente adotada devido à sua produtividade e confiabilidade.

No contexto da plataforma iOS, o iOS SDK foi utilizado para possibilitar a instalação do aplicativo por meio da *App Store*. Para o desenvolvimento, foi escolhida a linguagem de programação *Swift*, que oferece várias vantagens no desenvolvimento de aplicativos para iOS, incluindo segurança, desempenho, legibilidade de código, facilidade de manutenção, suporte à programação assíncrona, sintaxe expressiva e um conjunto abrangente de bibliotecas e *frameworks*. O *Swift* é uma linguagem moderna desenvolvida pela *Apple*, que proporciona uma experiência de desenvolvimento agradável e eficiente.

Foi adotado o *Next.js* para criar o aplicativo PWA, possibilitando o acesso e a instalação do aplicativo por meio do navegador. A linguagem utilizada para o desenvolvimento foi o *TypeScript*, que oferece recursos como tipagem estática, autocompletar e *IntelliSense*, refatoração segura, escalabilidade e manutenibilidade do código, além de integração com o ecossistema *JavaScript* existente. Esses recursos contribuem para um desenvolvimento mais eficiente, com menos erros e maior qualidade.

O *front-end* do aplicativo PWA foi hospedado na plataforma *Vercel*, que oferece uma experiência de desenvolvimento simplificada. A plataforma *Vercel* suporta projetos estáticos e dinâmicos, facilitando a implantação de aplicativos *Next.js*. Com recursos avançados de dimensionamento automático, *deploy* instantâneo e integração contínua, os desenvolvedores podem implantar seus aplicativos de forma rápida e confiável, garantindo alta disponibilidade e desempenho otimizado. Além disso, a *Vercel* oferece recursos como análise de desempenho, cache global e personalização de domínio, permitindo que os desenvolvedores forneçam experiências excepcionais aos usuários finais.

O *back-end* do projeto foi desenvolvido utilizando a linguagem *TypeScript*, que é uma extensão do *JavaScript* conhecida por adicionar recursos de tipagem estática ao código. O ambiente de execução escolhido foi o *Node.js*, que permite a execução de código *JavaScript* no lado do servidor. Essa combinação de *TypeScript* e *Node.js* oferece um ambiente de desenvolvimento eficiente e escalável para o *back-end*.

A infraestrutura do *back-end* foi hospedada na plataforma AWS (*Amazon Web Services*). O AWS EC2 foi utilizado para hospedar e fazer o *deploy* do servidor *back-end*, garantindo escalabilidade e disponibilidade. Além disso, o

servidor *Keycloak*, responsável pelo sistema de autenticação e autorização (IAM - *Identity and Access Management*), também foi hospedado no AWS EC2.

O banco de dados utilizado foi o *PostgreSQL*, uma solução robusta e confiável para armazenamento de dados. O *PostgreSQL* oferece recursos avançados de gerenciamento e consulta de dados, garantindo a integridade e segurança das informações. Para hospedar o banco de dados, foi utilizada a AWS RDS, uma solução altamente escalável e gerenciada de banco de dados na nuvem, que oferece desempenho confiável e alta disponibilidade.

Para o armazenamento das imagens dos produtos, foi utilizado o serviço AWS S3, que é um serviço de armazenamento em nuvem altamente escalável e durável. O AWS S3 permite o armazenamento e acesso eficientes das imagens dos produtos.

4.6 INTEGRAÇÃO COM SISTEMAS INTERNOS

O aplicativo uCycle foi desenvolvido com a inclusão de diversas integrações com sistemas externos, visando aprimorar sua funcionalidade e proporcionar uma experiência completa aos usuários. Algumas das principais integrações implementadas foram as seguintes:

- IAM (*Identity and Access Management*): O sistema de autenticação e autorização do aplicativo foi realizado por meio do *Keycloak*. Essa integração foi fundamental para garantir a segurança dos dados dos usuários e gerenciar processos como login, registro e recuperação de senha.
- AWS S3: A hospedagem e implantação do *back-end* do aplicativo ocorreram na plataforma AWS EC2. Além disso, foi realizada a integração com o AWS S3, permitindo o armazenamento seguro e confiável de arquivos, especialmente as imagens dos produtos.
- *SendGrid*: Foi incorporada à integração com o *SendGrid* para possibilitar o envio de *e-mails*. Essa funcionalidade é de extrema importância para a comunicação com os usuários, incluindo confirmações de cadastro, atualizações de pedidos e notificações diversas.

- API Correios: A API dos Correios foi integrada ao aplicativo para viabilizar a consulta de fretes. Essa integração permite que os usuários obtenham informações precisas sobre o custo e prazo de entrega de seus pedidos.
- Brasil API para consulta de CEP/Endereço: Foi realizada a integração com a Brasil API para possibilitar a consulta de CEP e endereço. Essa funcionalidade simplifica o preenchimento automático de dados de entrega, proporcionando uma experiência mais ágil e conveniente para os usuários.
- API de pagamentos: A integração com a API do Mercado Pago foi planejada para oferecer aos usuários opções diversificadas de pagamento, incluindo PIX, cartão de crédito e boleto bancário. Essa integração assegurará um processo de finalização de compra seguro e eficiente, proporcionando uma experiência satisfatória para os usuários.

Essas integrações foram criteriosamente selecionadas com o propósito de proporcionar valor adicional ao aplicativo uCycle, tornando-o mais robusto, seguro e funcional. Cada integração contribuiu para uma experiência do usuário aprimorada, em conformidade com as necessidades e expectativas dos usuários.

5 RESULTADOS E DISCUSSÕES

Neste capítulo, serão expostos os resultados alcançados mediante a exibição de imagens da interface da aplicação concluída. Essas imagens permitirão uma compreensão abrangente da aparência e funcionalidades da aplicação, evidenciando a forma como o projeto foi desenvolvido.

5.1 RESULTADOS OBTIDOS

Como elucidado anteriormente, o objetivo deste projeto é desenvolver uma plataforma de e-commerce móvel que promova a moda circular e a logística reversa, gerando um impacto positivo no meio ambiente. O aplicativo funcionará como uma plataforma de comércio eletrônico, permitindo aos usuários comprar e vender produtos de forma sustentável. Além de incentivar o consumo consciente e a reutilização de materiais, o aplicativo fornecerá uma experiência intuitiva e funcional, permitindo que os usuários explorem catálogos de produtos, realizem transações seguras e participem ativamente do ciclo de moda sustentável.

5.1.1 Capturas de tela e suas funcionalidades

Nesta seção, serão discutidas individualmente as telas e funcionalidades da aplicação em foco neste estudo. Cada uma delas será detalhadamente abordada, fornecendo uma visão abrangente das diferentes interfaces e recursos disponíveis na aplicação. O objetivo é analisar e descrever em profundidade as características e a usabilidade de cada tela, bem como destacar a funcionalidade específica que cada uma delas proporciona aos usuários.

5.1.1.1 Página Inicial

A tela inicial do aplicativo como vista na Figura 6, possibilita aos usuários o acesso ao catálogo de produtos sem necessariamente fazer *login* ou cadastro no *app*. Ao lado na Figura 7 é possível ver o campo de busca da tela inicial.

Figura 6 - Tela inicial

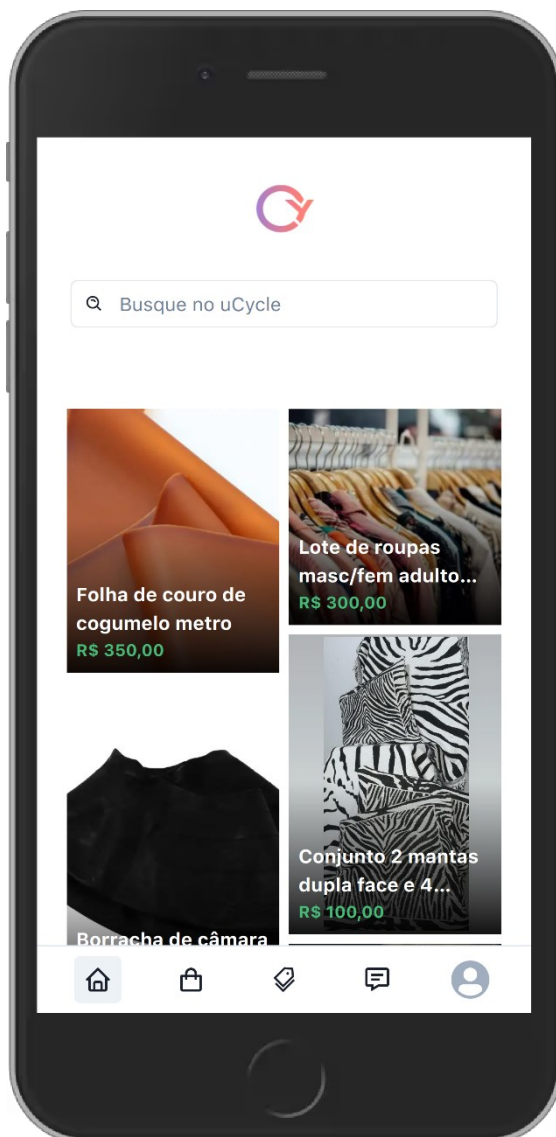
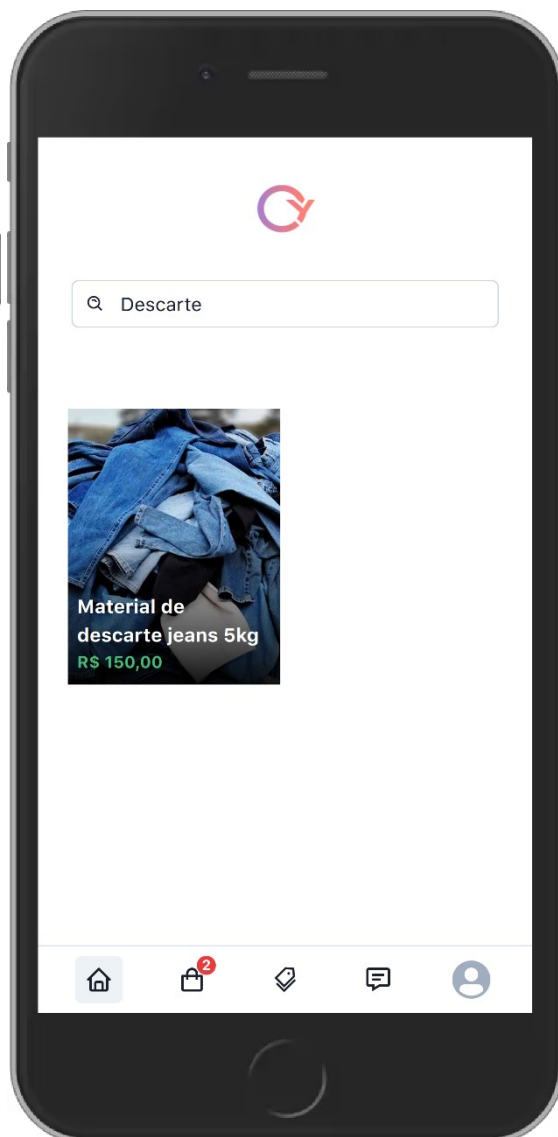


Figura 7 - Tela inicial com pesquisa

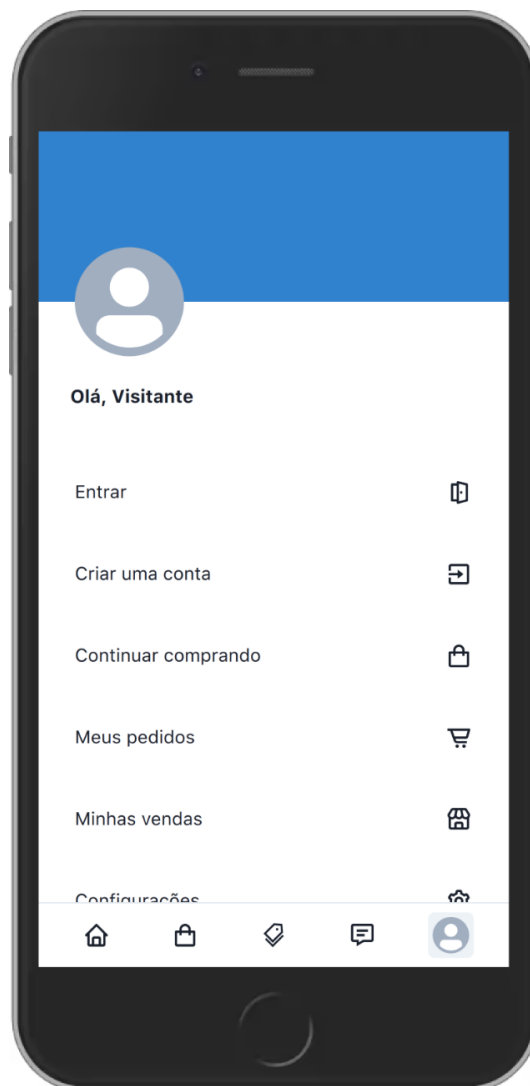


Fonte: Elaborada pelos autores (2023)

5.1.1.2 Perfil visitante

O usuário não cadastrado tem acesso à esta tela (Figura 8) que apresenta informações básicas sobre um visitante, como opção de *login*, de cadastro, de acesso a página inicial e configurações, permitindo uma interação limitada até que o *login* e/ou cadastro seja realizado. Ao visitante é permitido colocar compras no carrinho sem a necessidade de *login* e/ou cadastro.

Figura 8 - Perfil visitante

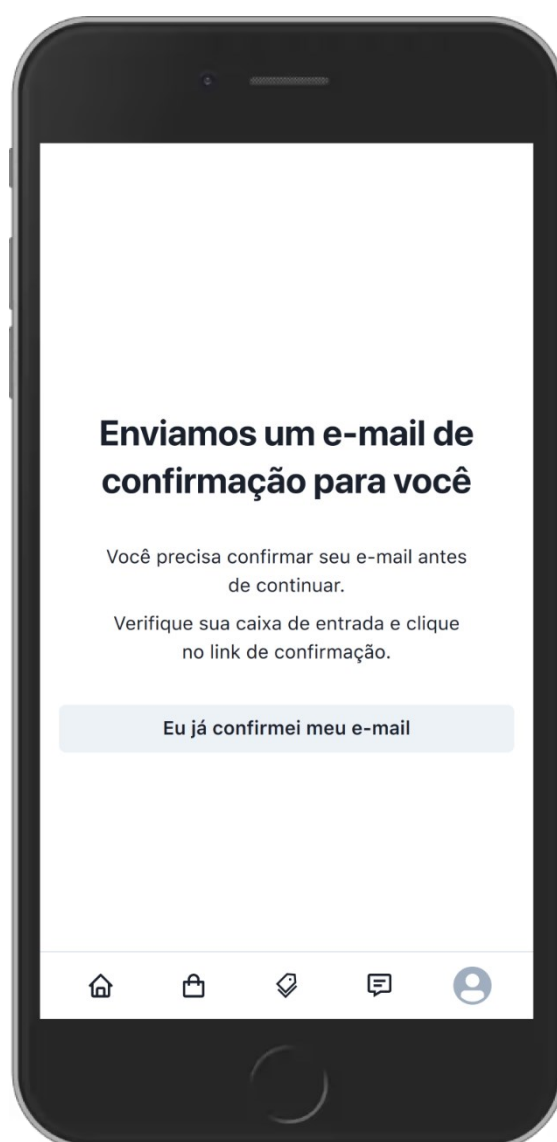


Fonte: Elaborada pelos autores (2023)

5.1.1.3 Cadastro

Nesta etapa é onde o usuário pode fazer um cadastro simples (como *e-mail* e senha) e receber um *e-mail* de confirmação do uCycle para acesso ao aplicativo em segurança como visto na Figura 9.

Figura 9 - Cadastro

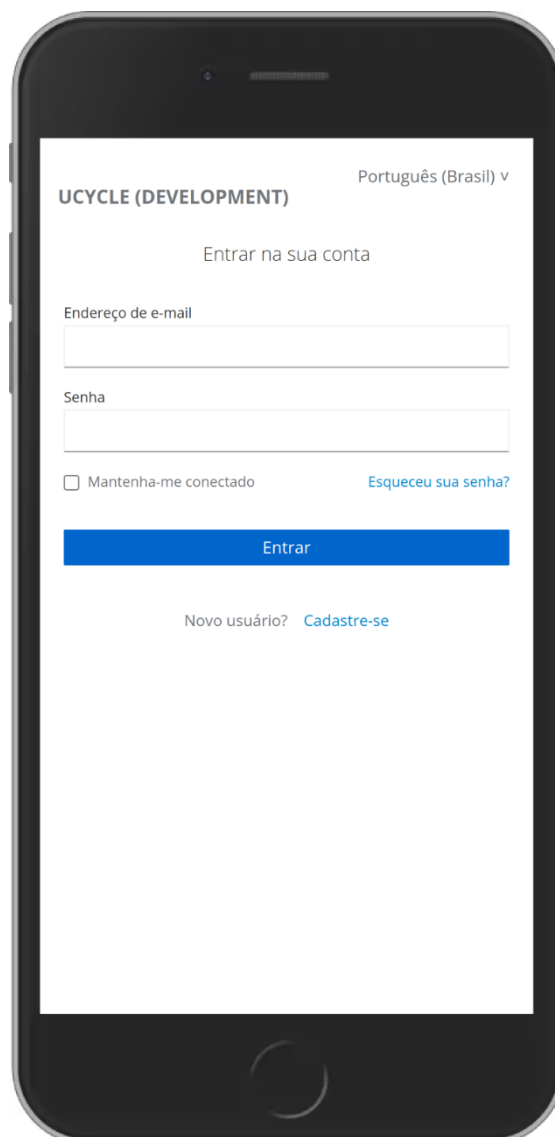


Fonte: Elaborada pelos autores (2023)

5.1.1.4 Login

A presente tela (Figura 10) possibilita aos usuários a inserção de suas credenciais de *login* (como *e-mail* e senha) para acessar sua conta existente e usufruir das funcionalidades completas da aplicação.

Figura 10 - Login



UCYCLE (DEVELOPMENT) Português (Brasil) v

Entrar na sua conta

Endereço de e-mail

Senha

Mantenha-me conectado [Esqueceu sua senha?](#)

[Entrar](#)

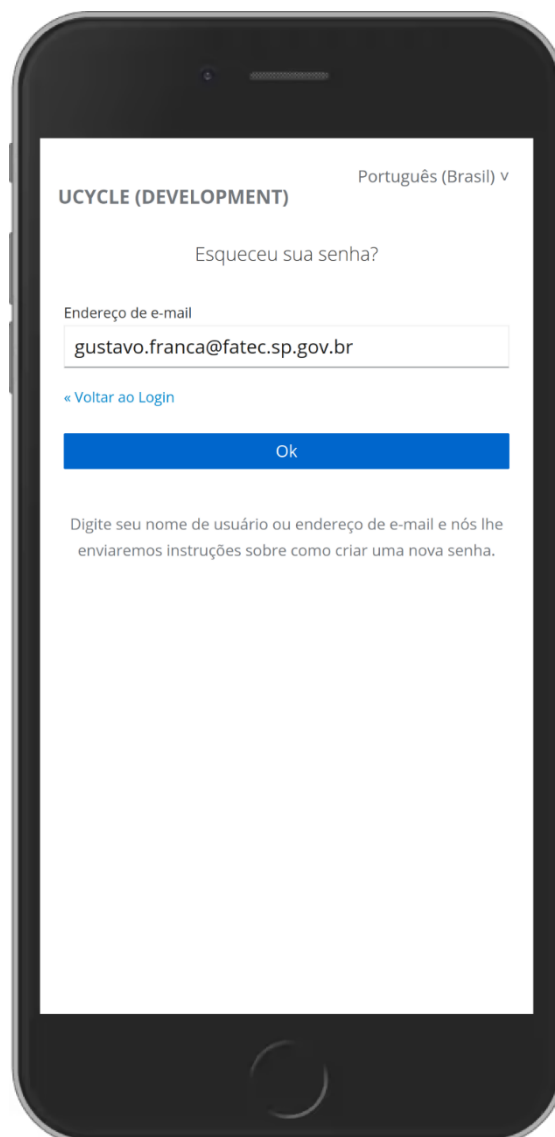
Novo usuário? [Cadastre-se](#)

Fonte: Elaborada pelos autores (2023)

5.1.1.5 Esqueceu a senha

A Figura 11 mostra a tela que permite aos usuários recuperar o acesso a conta caso tenham esquecido a senha, fornecendo informações de verificação para redefinir a senha.

Figura 11 - Esqueceu a senha



The image shows a smartphone screen with the following content:

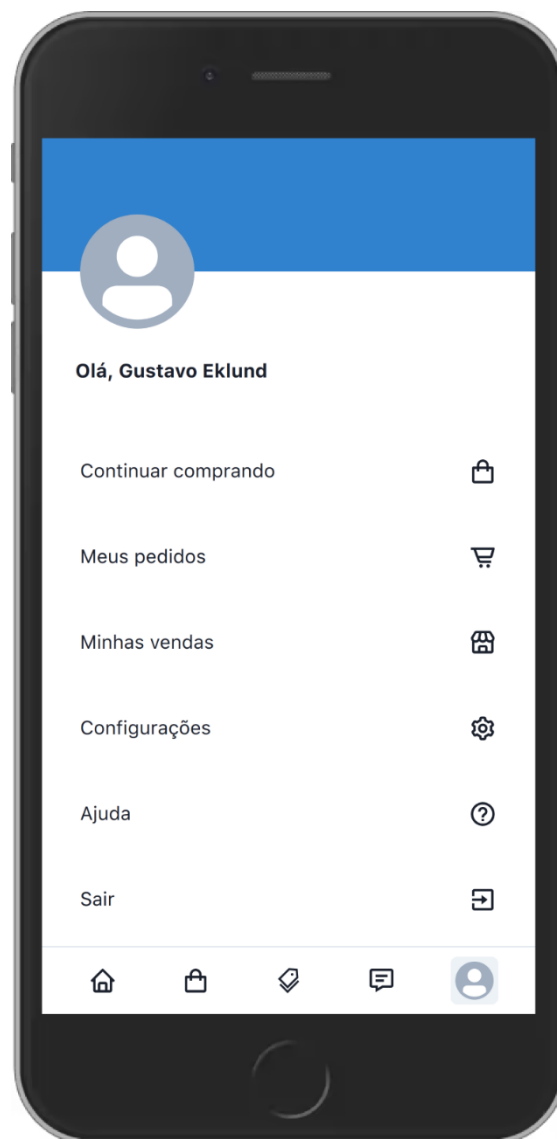
- Top right: Portuguese (Brasil) v
- Header: UCYCLE (DEVELOPMENT)
- Section title: Esqueceu sua senha?
- Form label: Endereço de e-mail
- Form input: gustavo.franca@fatec.sp.gov.br
- Link: < Voltar ao Login
- Button: Ok
- Text: Digite seu nome de usuário ou endereço de e-mail e nós lhe enviaremos instruções sobre como criar uma nova senha.

Fonte: Elaborada pelos autores (2023)

5.1.1.6 Perfil

A tela representada na Figura 12 é o perfil do usuário, que exibe informações detalhadas, como nome, foto, informações de contato, histórico de pedidos, configurações pessoais e outras informações relevantes. Similar a tela de visitante, porém agora com acesso a todas as funcionalidades.

Figura 12 - Perfil

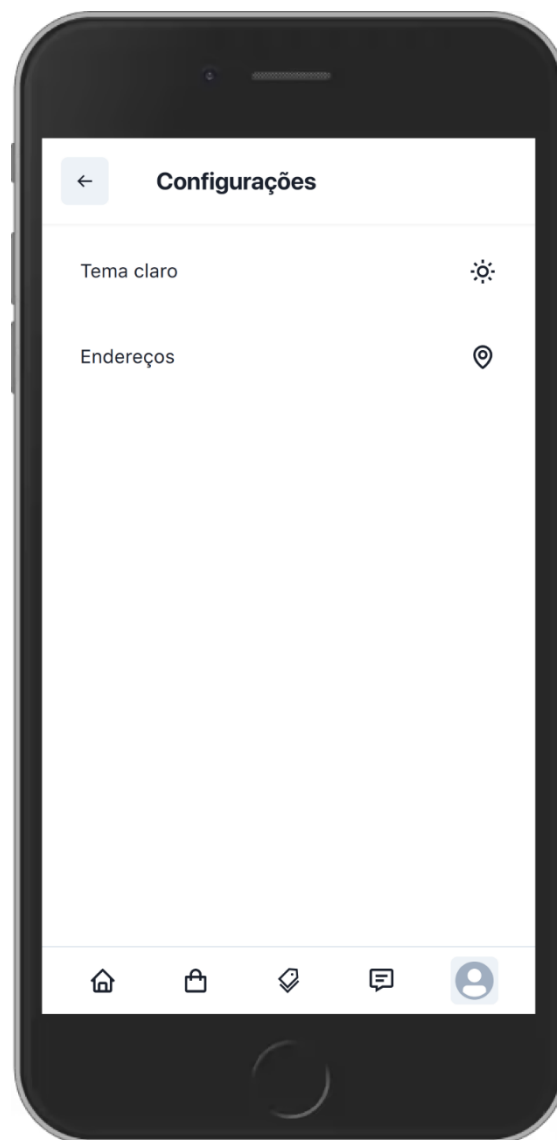


Fonte: Elaborada pelos autores (2023)

5.1.1.7 Configurações

Neste tela (Figura 13) os usuários podem personalizar suas preferências e configurações do aplicativo, como a escolha do tema e o gerenciamento e edição de seus endereços.

Figura 13 - Configurações

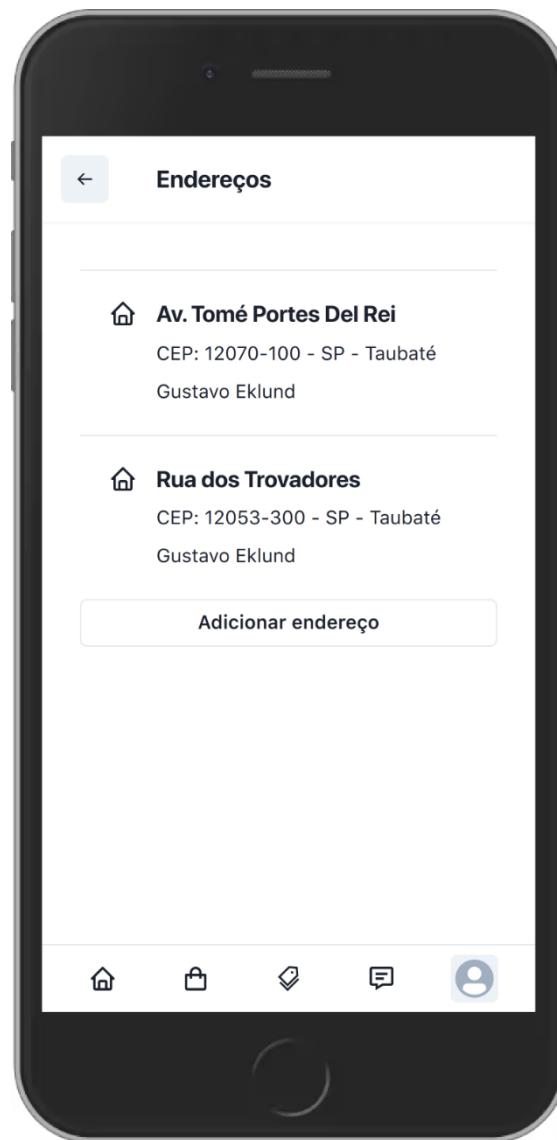


Fonte: Elaborada pelos autores (2023)

5.1.1.8 Endereços

A Figura 14 mostra a tela de endereços permite aos usuários visualizar, gerenciar e editar os endereços associados à sua conta, como endereços de entrega ou cobrança, facilitando a seleção de endereços durante o processo de compra.

Figura 14 - Endereços

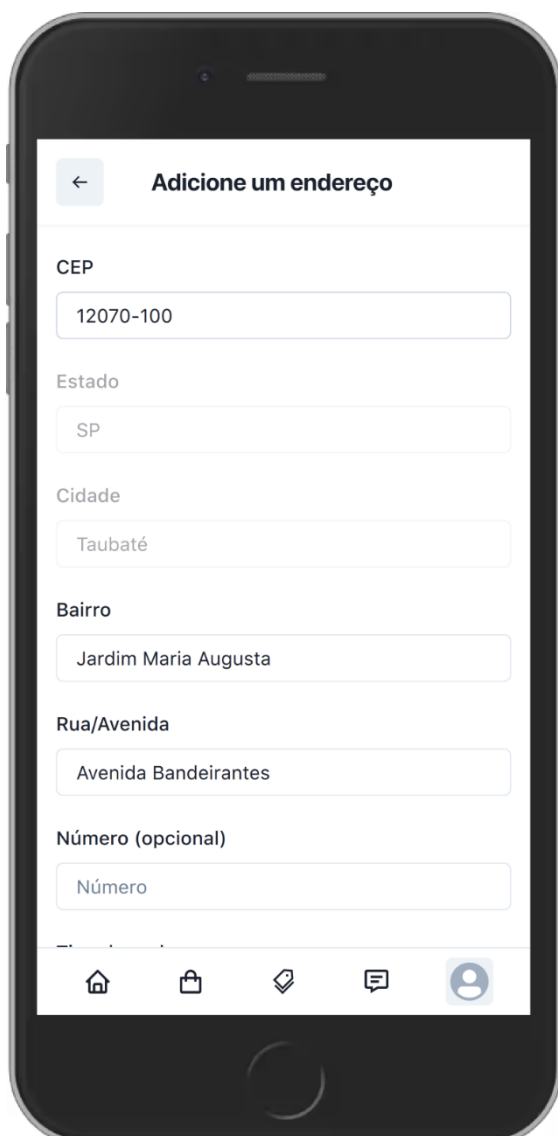


Fonte: Elaborada pelos autores (2023)

5.1.1.9 Adicionar endereços

Abaixo na Figura 15 e Figura 16 está a tela de adicionar endereços, onde os usuários podem adicionar novos endereços à lista de endereços salvos, fornecendo informações relevantes, como nome, endereço, número de telefone e outras informações de contato.

Figura 15 - Adicione um endereço

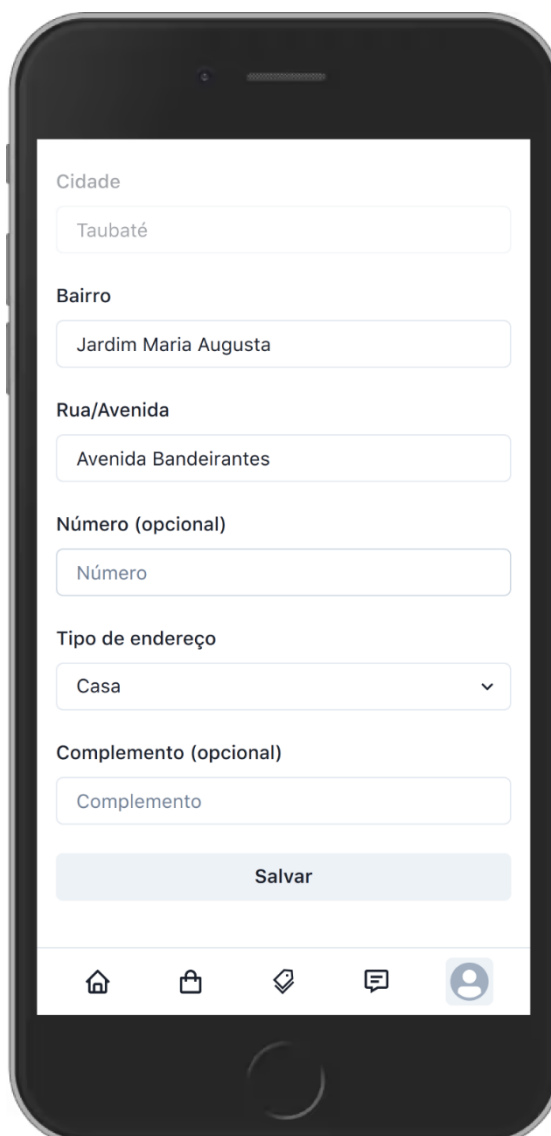


The image shows a smartphone screen with the title "Adicione um endereço". The form contains the following fields:

- CEP: 12070-100
- Estado: SP
- Cidade: Taubaté
- Bairro: Jardim Maria Augusta
- Rua/Avenida: Avenida Bandeirantes
- Número (opcional): Número

At the bottom of the screen, there is a navigation bar with icons for Home, Shopping, Cards, Messages, and Profile.

Figura 16 - Adicione um endereço



The image shows a smartphone screen with the title "Adicione um endereço". The form contains the following fields:

- Cidade: Taubaté
- Bairro: Jardim Maria Augusta
- Rua/Avenida: Avenida Bandeirantes
- Número (opcional): Número
- Tipo de endereço: Casa
- Complemento (opcional): Complemento

At the bottom of the screen, there is a navigation bar with icons for Home, Shopping, Cards, Messages, and Profile. A "Salvar" button is located at the bottom of the form.

Fonte: Elaborada pelos autores (2023)

5.1.1.10 Anunciar produto

Este é o local onde é permitido aos usuários criar e publicar anúncios de produtos para venda, fornecendo informações detalhadas sobre o produto, como título, descrição, preço, imagens e outros detalhes relevantes. Na Figura 17 está a tela que permite que o usuário anuncie um produto, a Figura 18 exibe a tela em que o usuário especifica o produto, marca e modelo.

Figura 17 - Introdução ao anunciar produto

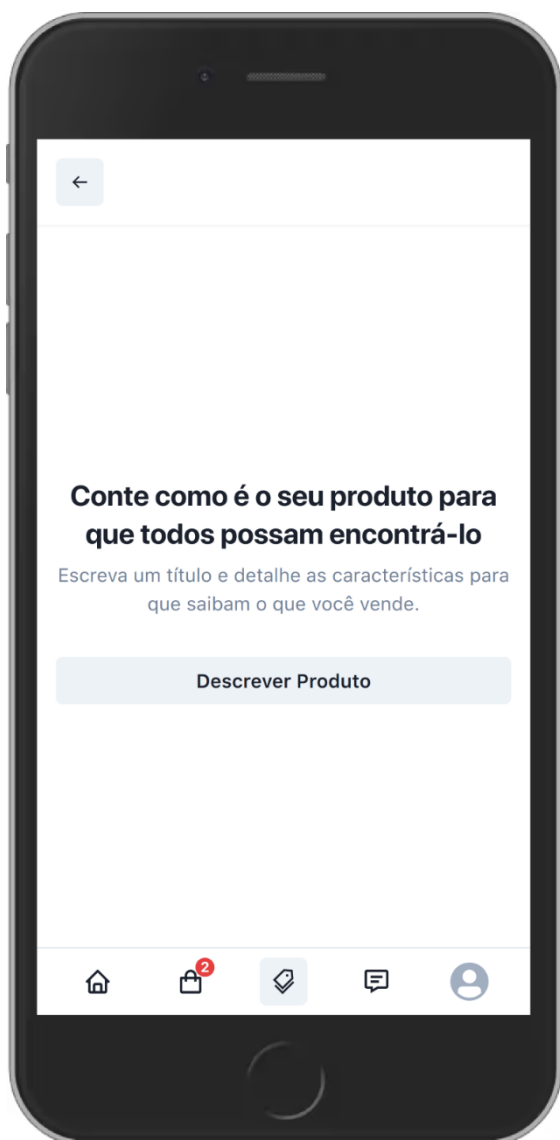
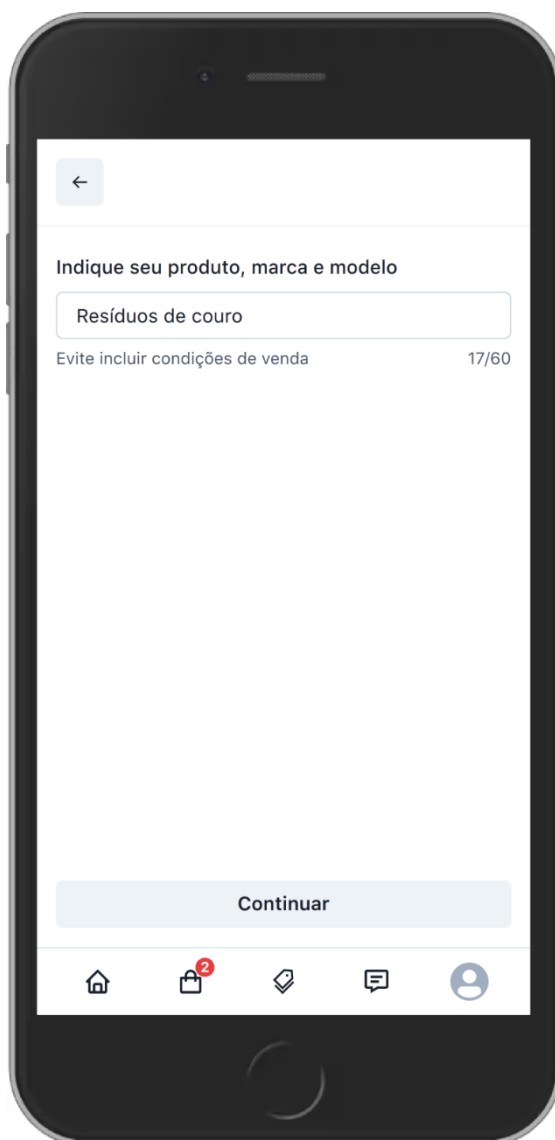


Figura 18 - Definir o título ao anunciar o produto

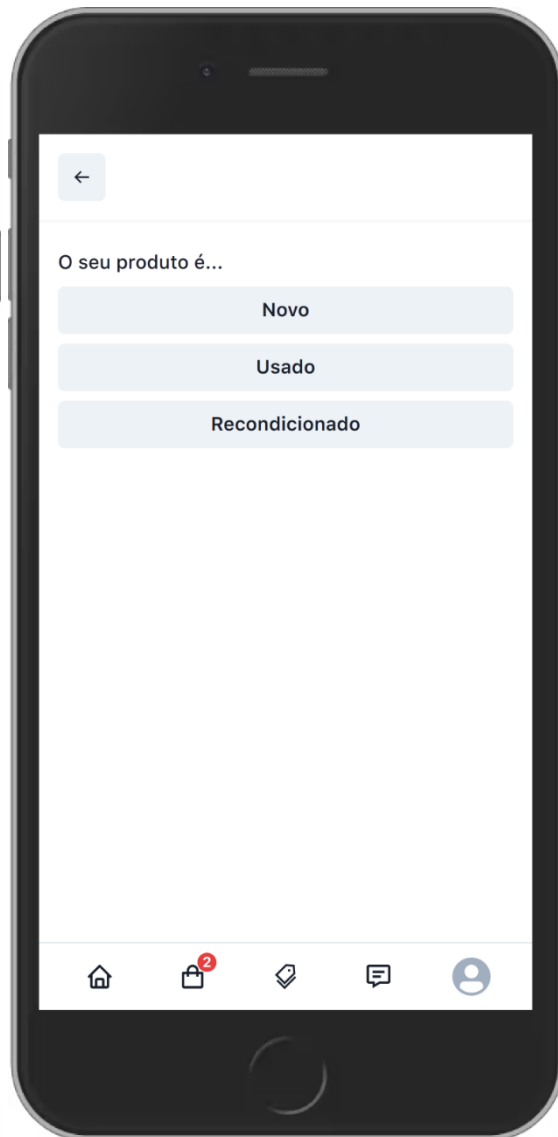


Na Figura 19 está a tela em que o usuário define em que categoria se encaixa o seu produto, A Figura 20 apresenta a tela onde é possível definir se o produto anunciado pelo usuário é novo, usado ou recondicionado

Figura 19 - Definir descrição ao anunciar o produto



Figura 20 - Definir a condição ao anunciar o produto



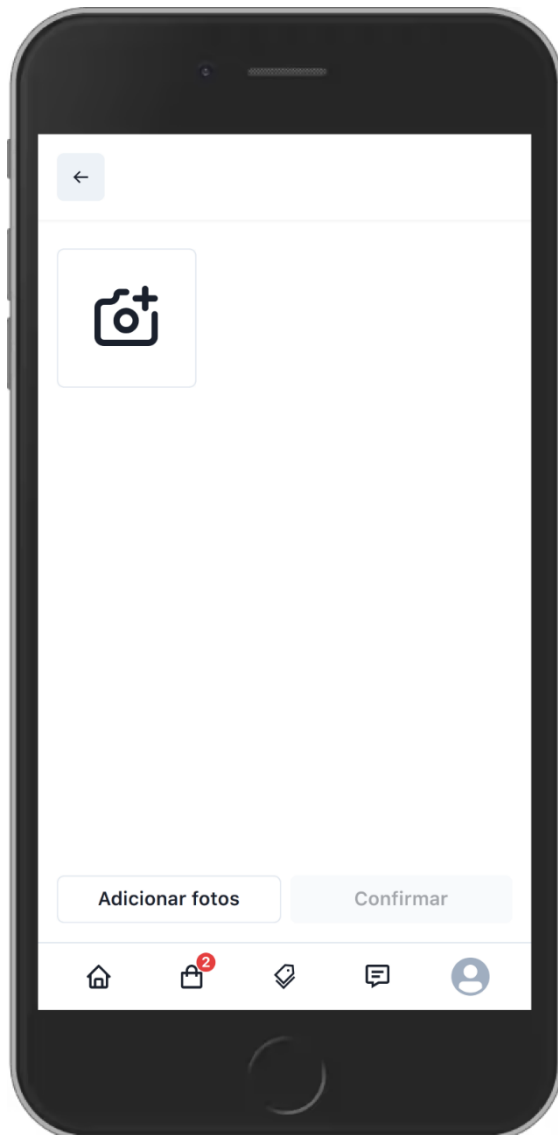
Fonte: Elaborada pelos autores (2023)

A Figura 21 exibe a tela para adicionar fotos do produto, ao clicar no botão “Adicionar fotos” o usuário é encaminhado para a tela representada na Figura 22, onde a ele é permitido adicionar fotos clicando no ícone da câmera fotográfica.

Figura 21 - Instruções para adicionar fotos ao anunciar o produto



Figura 22 - Adicionar fotos ao anunciar o produto



Fonte: Elaborada pelos autores (2023)

Ao clicar no botão “Adicionar fotos” (Figura 23) a tela representada na Figura 24 abrirá e ao usuário será permitido visualizar as fotos adicionadas previamente e gerenciá-las antes de serem publicadas. Ao clicar no botão “Confirmar” o usuário será encaminhado para a tela demonstrada da Figura 24, onde ele poderá descrever o produto a ser publicado.

Figura 23 - Editar fotos ao anunciar o produto

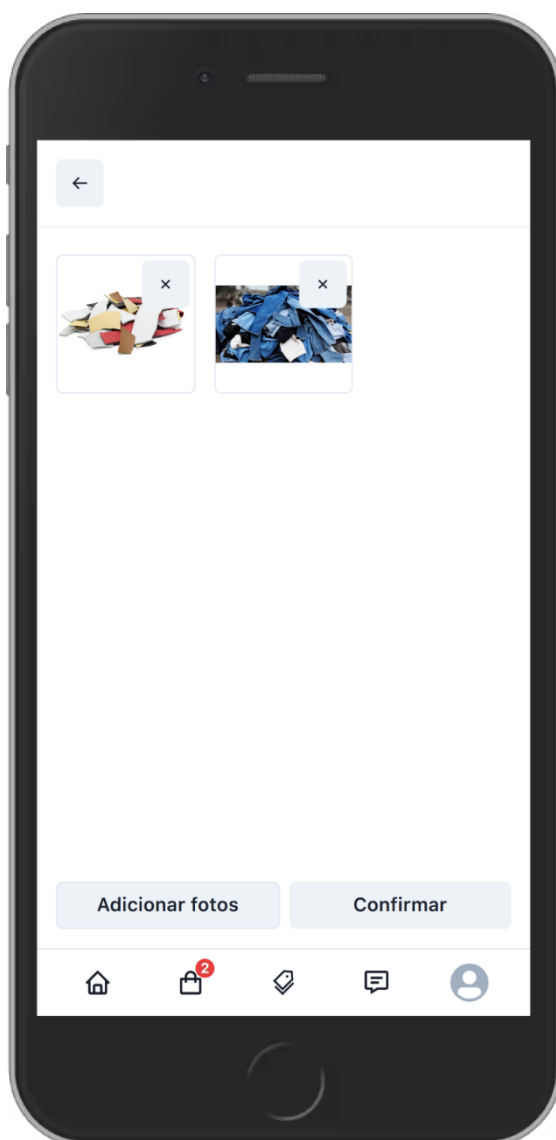


Figura 24 - Descrever o produto ao anunciar o produto



Fonte: Elaborada pelos autores (2023)

Assim que o usuário descrever o produto a ser publicado, ele será encaminhado para a tela de preço, retratada na Figura 25, em seguida ao clicar o botão “Continuar” o usuário abrirá a tela de garantia (Figura 26) onde ele poderá optar por oferecer ou não garantia ao comprador.

Figura 25 - Definir preço ao anunciar o produto

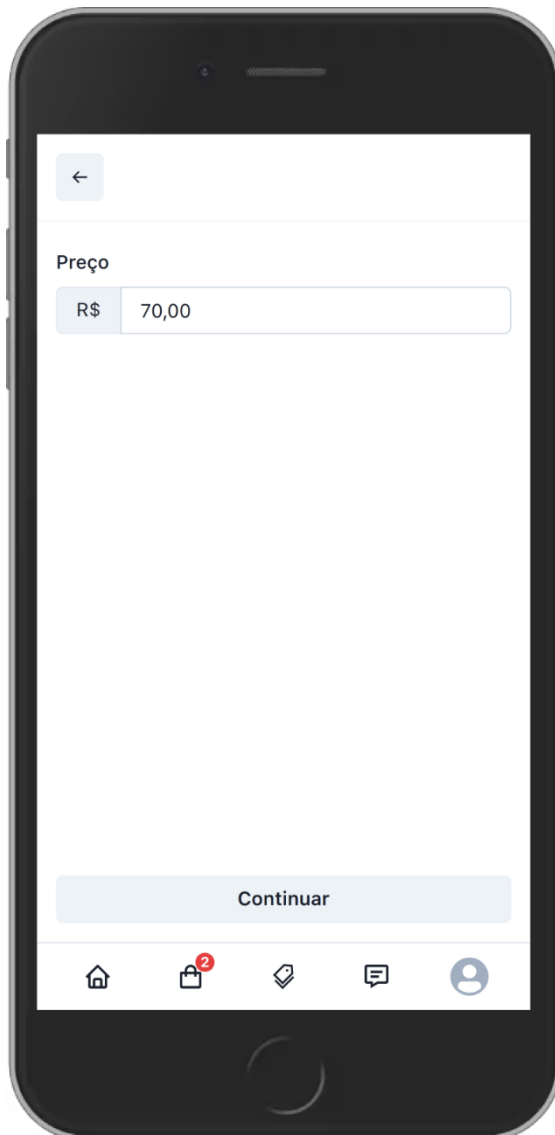


Figura 26 - Definir garantia ao anunciar o produto



Fonte: Elaborada pelos autores (2023)

Caso o usuário opte por oferecer garantia a tela representada na Figura 27 aparecerá, e o usuário poderá escolher o prazo de garantia dada ao comprador, passado essa etapa, ao clicar “Continuar” a tela mostrada na figura 28 abrirá permitindo ao usuário revisar o anúncio antes de publicar.

Figura 27 - Definir o tempo de garantia ao anunciar o produto

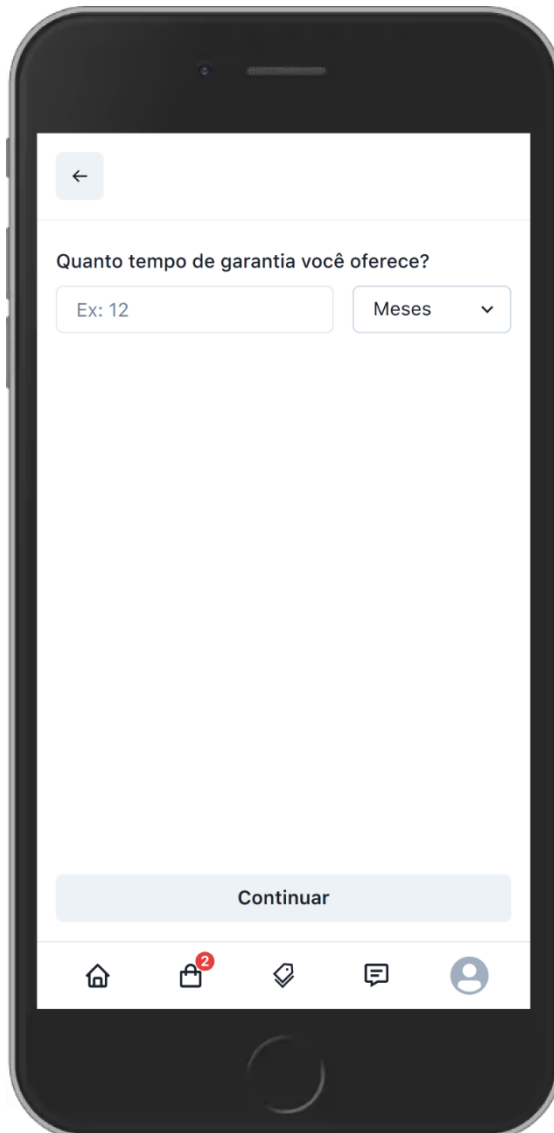


Figura 28 - Aviso de revisão ao anunciar o produto



Fonte: Elaborada pelos autores (2023)

As Figuras 29 e 30 representam a tela para revisar o anúncio antes de publicar, nesta tela o usuário poderá fazer qualquer alteração que lhe pareça pertinente, como a escolha das fotos, título do anúncio, preço, descrição do produto, garantia e condição.

Figura 29 - Revisar e anunciar o produto

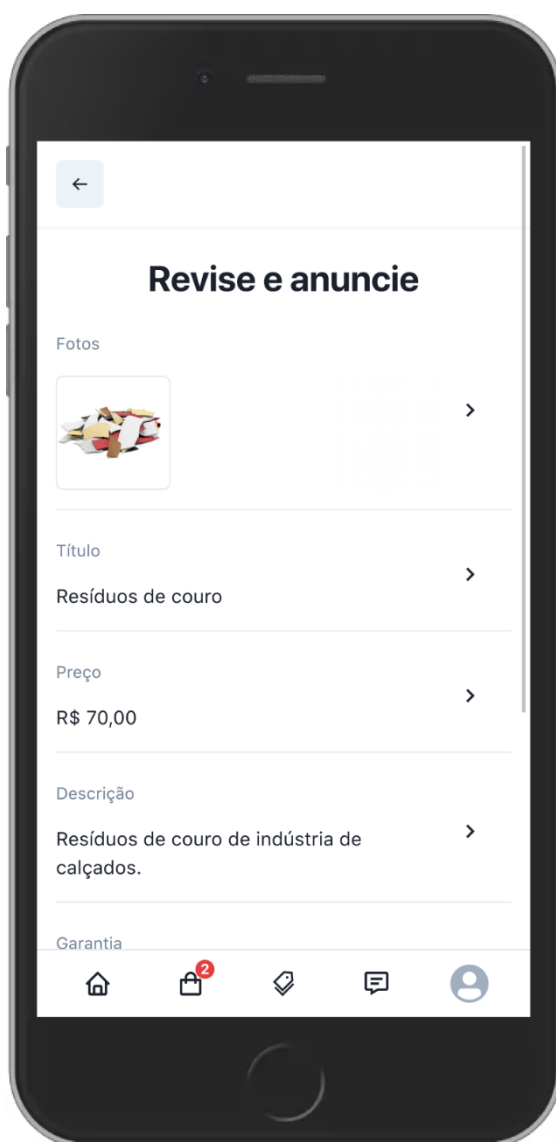


Figura 30 - Revisar e anunciar o produto



Fonte: Elaborada pelos autores (2023)

5.1.1.11 Detalhes do produto

A tela de detalhes do produto é a tela que exibe informações completas e detalhadas sobre um produto específico, incluindo imagens, descrição, especificações, preço e outras informações relevantes para ajudar os usuários a tomar decisões informadas. Abaixo ela está representada nas Figuras 31 e 32.

Figura 31 - Detalhes do produto

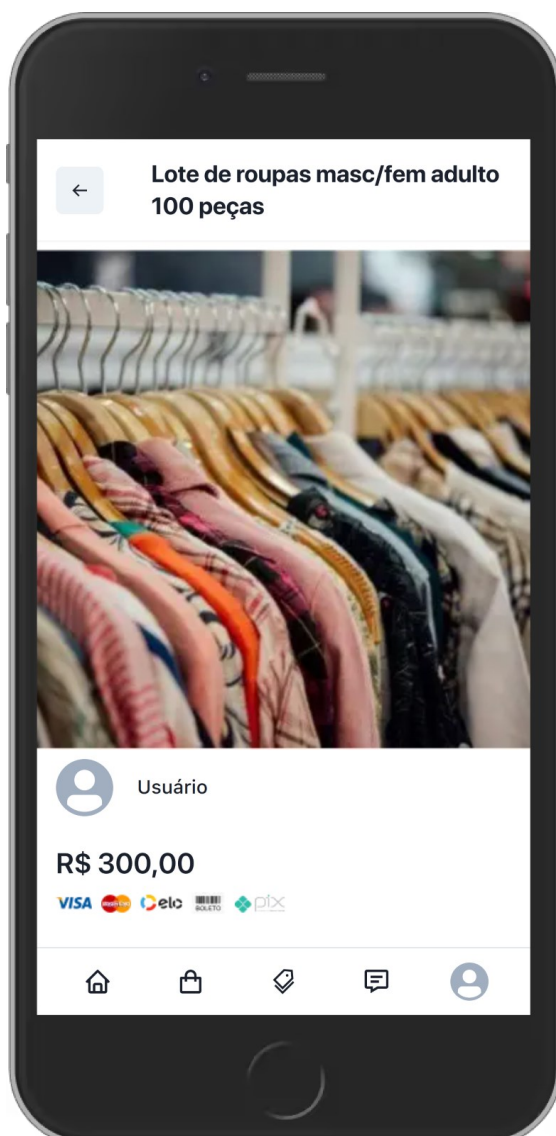


Figura 32 - Detalhes do produto

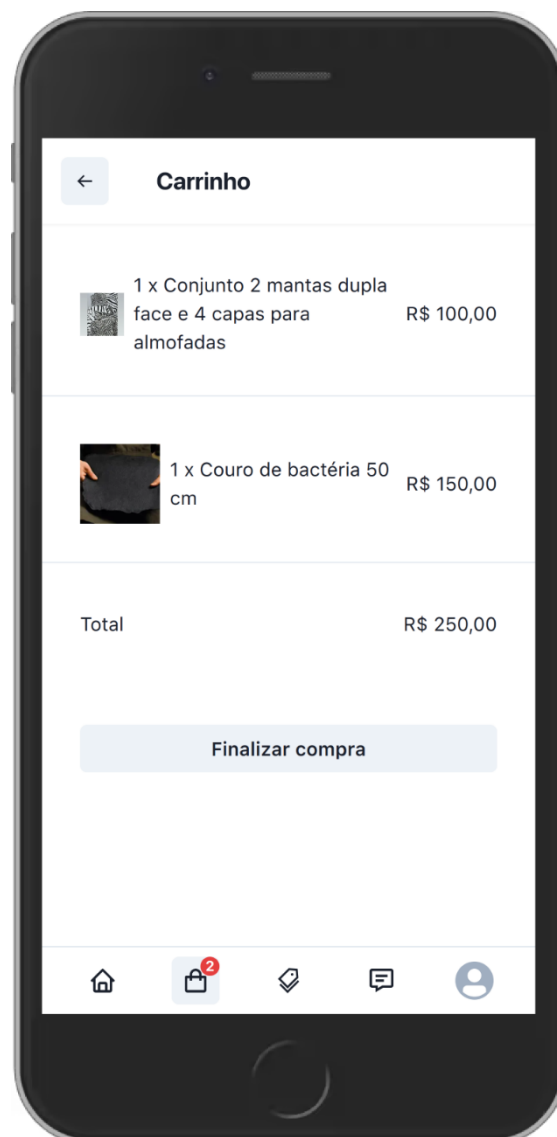


Fonte: Elaborada pelos autores (2023)

5.1.1.12 Carrinho de Compras

Nesta tela os usuários podem visualizar e gerenciar os itens selecionados para compra, adicionar ou remover itens, ajustar as quantidades e visualizar o total a ser pago antes de finalizar a compra, como apresentado na Figura 33.

Figura 33 - Carrinho de compras



Fonte: Elaborada pelos autores (2023)

5.1.1.13 Finalizar Compra

Esta é a tela (representada logo abaixo nas Figuras 34 e 35) onde os usuários podem revisar e confirmar todos os detalhes do pedido, como produtos selecionados, endereço de entrega, opções de envio, método de pagamento e outras informações relevantes, antes de concluir a compra.

Figura 34 - Finalizar compra

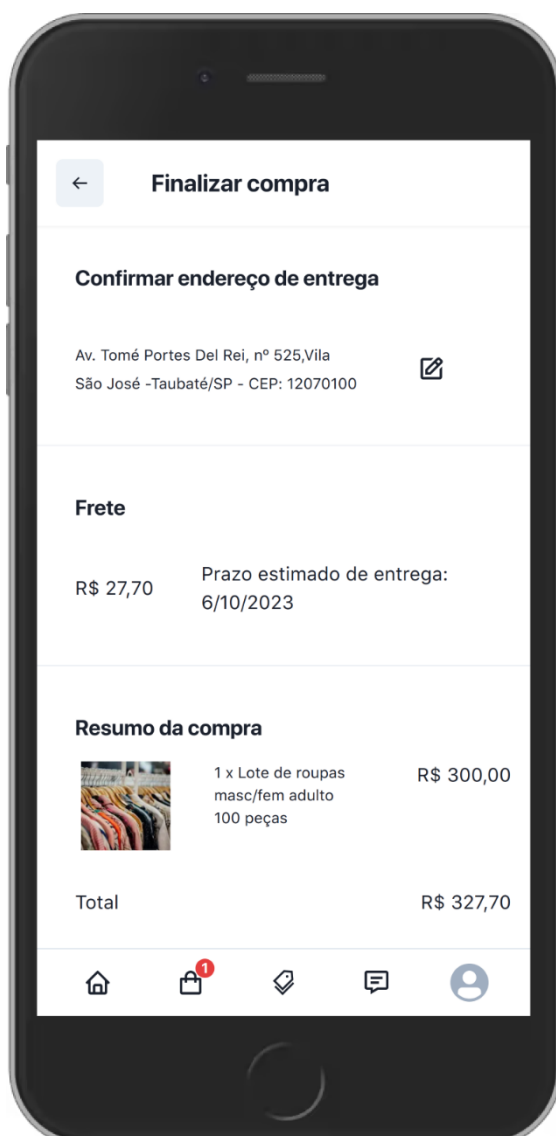


Figura 35 - Finalizar compra



Fonte: Elaborada pelos autores (2023)

5.1.1.14 Pagamento do Pedido

Nesta etapa o usuário tem acesso a uma tela segura onde podem inserir as informações de pagamento necessárias, como detalhes do cartão de crédito, informações de conta bancária ou outros métodos de pagamento aceitos, para efetuar o pagamento do pedido. Na Figura 36 é representada a tela onde o usuário escolhe o meio de pagamento do produto, na Figura 37, representada ao lado o usuário escolheu o cartão de crédito como método de pagamento onde ele irá colocar seus dados bancários.

Figura 36 - Pagamento do pedido com opções para selecionar

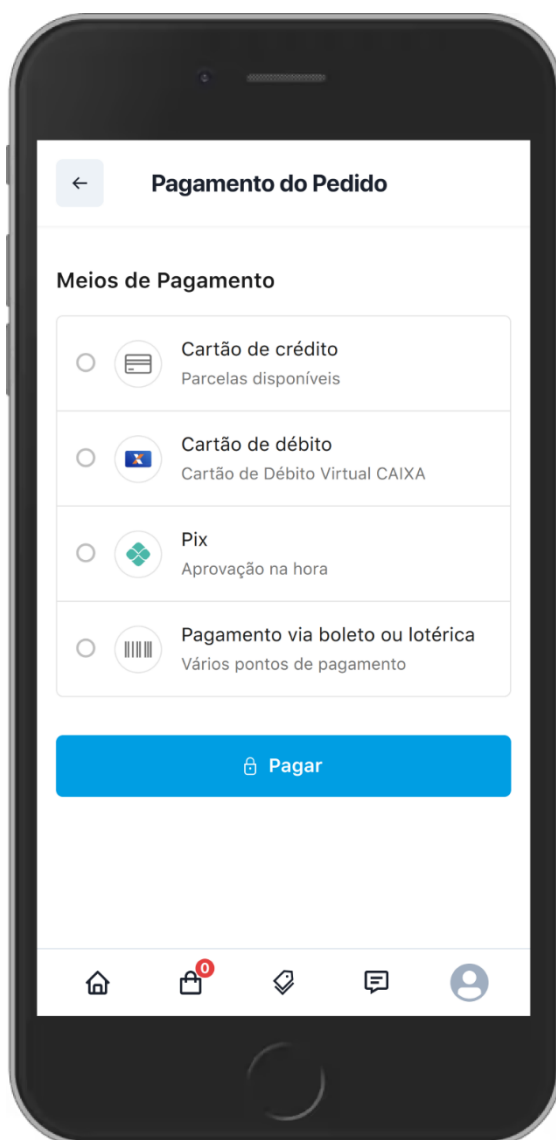
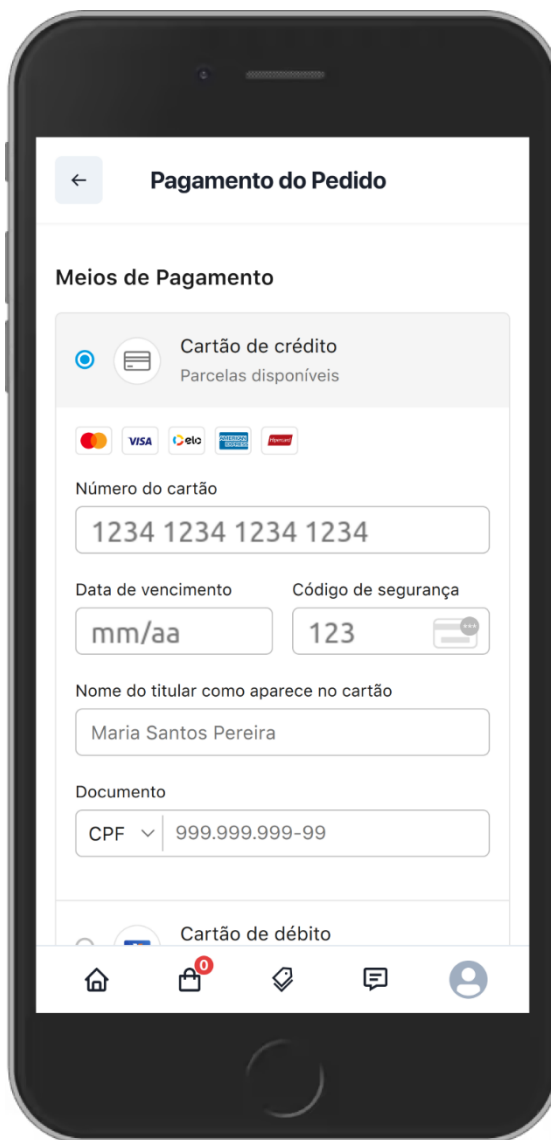


Figura 37 - Pagamento do pedido via cartão de crédito



5.1.1.15 Meus Pedidos

Na Figura 38 é exibida a tela de histórico de pedidos. Na Figura 39 a mesma tela é representada, porém no tema escuro, nos aplicativos mobile o tema escuro traz ganhos como a diminuição do cansaço visual, economia de energia em dispositivos OLED/AMOLED, uma aparência agradável, além de ser vantajoso para pessoas sensíveis à luz ou com condições visuais específicas.

Figura 38 - Meus pedidos, tema claro

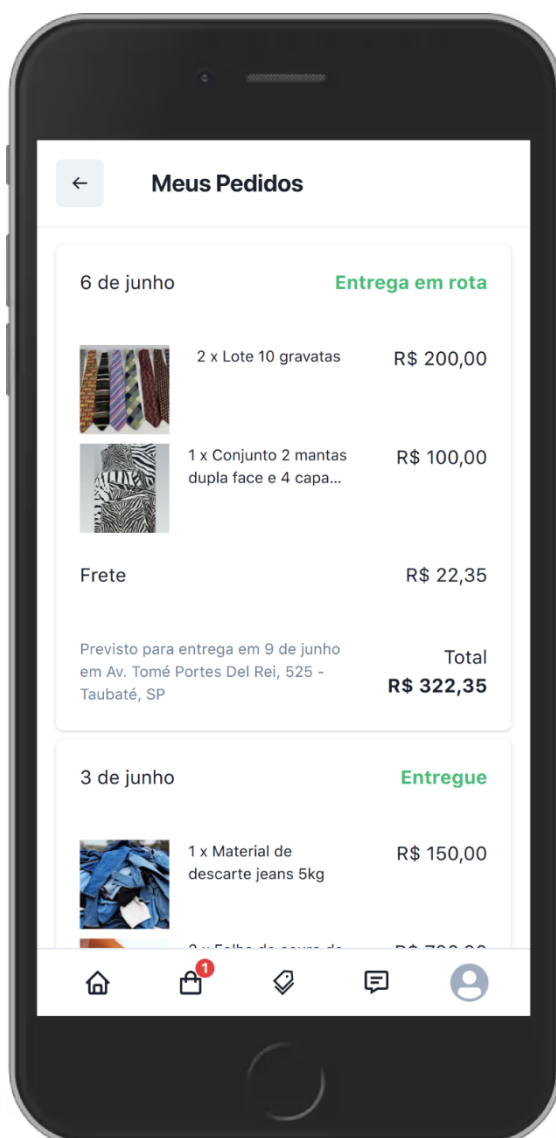


Figura 39 - Meus pedidos, tema escuro



Fonte: Elaborada pelos autores (2023)

6 CONCLUSÃO

Através deste estudo, é possível afirmar que o desenvolvimento de um aplicativo *mobile* voltado para moda circular apresenta um potencial significativo para contribuir com a conscientização e a adoção de práticas sustentáveis na indústria da moda, não há atualmente no mercado soluções com a mesma proposta. Por meio da análise dos problemas relacionados ao consumo desenfreado e ao impacto negativo no meio ambiente, foi ressaltada a importância de buscar alternativas que reduzam o desperdício e os resíduos nesse setor.

A moda circular surge como uma abordagem promissora, visando a reutilização de materiais, a valorização do consumo consciente e a implementação da logística reversa. Nesse sentido, a criação de um aplicativo *mobile* específico para promover a moda circular e incentivar práticas sustentáveis representa um avanço no engajamento dos consumidores, oferecendo-lhes informações relevantes, ferramentas e recursos para fazer escolhas mais conscientes no momento da compra e utilização de roupas.

Por fim, é importante destacar que este trabalho abre portas para futuras pesquisas e aprimoramentos no desenvolvimento de soluções tecnológicas que contribuam para a construção de uma indústria da moda mais sustentável. A moda circular e a conscientização sobre seus benefícios representam um caminho promissor para a preservação do meio ambiente e a promoção da sustentabilidade.

6.1 TRABALHOS FUTUROS

Uma plataforma de *e-commerce* apresenta uma ampla gama de funcionalidades e desafios técnicos, que exigem abordagens de implementação apropriadas. Neste estudo de prova de conceito, são identificadas várias áreas de melhoria e expansão potenciais, que podem ser consideradas para futuras implementações. Algumas das principais áreas de foco incluem:

- Implementação de ferramentas de monitoramento, como o *New Relic*, para obter métricas abrangentes relacionadas ao uso da plataforma, exceções e tráfego da aplicação. Essas métricas podem fornecer insights valiosos para otimizar o desempenho e identificar possíveis pontos de melhoria.
- Aplicação de estratégias de cache tanto no *front-end* quanto no *back-end*, especialmente em áreas de alto volume de tráfego. O uso de tecnologias como *Redis* para armazenar dados em cache e serviços de CDN, como o *Amazon CloudFront*, para armazenar e distribuir imagens pode melhorar significativamente a velocidade e o desempenho geral da plataforma.
- Integração de provedores de identidade adicionais, como *Google* e *Facebook*, para facilitar o acesso dos usuários e proporcionar opções de login mais convenientes e amplamente utilizadas.
- Implementação de ferramentas de suporte ao vendedor, como a criação de uma página personalizada para cada vendedor, que possibilite uma melhor apresentação e gerenciamento dos seus produtos e serviços.
- Obtenção da certificação PCI-DSS (*Payment Card Industry Data Security Standard*) para garantir o armazenamento seguro dos dados de cartão de crédito dos clientes e aprimorar a experiência de finalização da compra, transmitindo confiança e segurança aos usuários.
- Separação dos diferentes contextos funcionais da aplicação em serviços independentes, visando melhorar a resiliência do sistema como um todo. Essa abordagem permitiria a divisão da plataforma em módulos especializados, como Vendas, Precificação, Catálogo de Produtos, Cliente, Inventário, Expedição, *Marketing*, Análises, Atendimento ao Cliente, Conformidade, KYC (*Know Your Customer*), Pagamentos, Suporte, Avaliação, Recomendação, entre outros. Essa separação favorece a escalabilidade, manutenção e evolução individual de cada serviço, facilitando também a integração com outros sistemas e serviços externos.

A implementação dessas melhorias potenciais pode valorizar a aplicação, aprimorando sua funcionalidade, desempenho, segurança e experiência geral do

usuário. Essas sugestões de aprimoramento são baseadas nas melhores práticas e tendências observadas no campo do comércio eletrônico.

REFERÊNCIA BIBLIOGRÁFICA

ALBUQUERQUE, Pedro Henrique Melo; SAAVEDRA, Cayan Atreio Portela Bárcena; MORAIS, Rafael Lima de; ALVES, Patrick Franco; YAOHAO, Peng. **Na era das Máquinas, o emprego é de quem? Estimação da probabilidade de automação de ocupações no Brasil.** IPEA - Instituto de Pesquisa Econômica Aplicada. Brasília, março de 2019. Disponível em: https://repositorio.ipea.gov.br/bitstream/11058/9116/1/td_2457.pdf Acesso em: 08 de junho 2023.

ANDRADE, Vitor. **Um Estudo Sobre Padrões e Tecnologias Para o Desenvolvimento Web – Back-end.** Disponível em: <http://monografias.poli.ufrj.br/monografias/monopoli10018659.pdf> Acesso em: 9 de julho de 2020.

ARGYLE, Zack. **A one year PWA retrospective.** Medium. Jul, 2018. Disponível em: <https://medium.com/pinterest-engineering/a-one-year-pwa-retrospective-f4a2f4129e05>. Acesso em: 9 de julho de 2020.

ASADI, Mohsen; RAMSIN, Raman. **Method engineering process patterns.** In: Proceedings of the 2nd India software engineering conference. ACM, 2009. p. 143–144. Disponível em: <https://dl.acm.org/doi/10.1145/1506216.1506249> Acesso em: 03 de abril de 2023.

BAUMAN, Zygmunt, **Vida para Consumo.** Rio de Janeiro. Zahar. 17 julho 2008.

BløRN-HANSEN Andreas; MAJCHRZAK Tim A.; GRØNLI Tor-Morten. **Progressive Web Apps: The Possible Web-native Unifier.** In Proceedings of

the 13th International Conference on Web Information Systems and Technologies (WEBIST 2017), pages 344-351. 2017.

BOCCATO, V. R. C. **Metodologia da pesquisa bibliográfica na área odontológica e o artigo científico como forma de comunicação.** Rev. Odontol. Univ. Cidade São Paulo, São Paulo, v. 18, n. 3, p. 265-274, 2006. Disponível em: https://arquivos.cruzeirodosuleducacional.edu.br/principal/old/revista_odontologia/pdf/setembro_dezembro_2006/metodologia_pesquisa_bibliografica.pdf Acesso em: 03 de abril de 2023.

BOGDAN, Robert C., & BIKLEN, Sari Knopp. **Qualitative Investigation in Education: An Introduction to Theory and Methods.** 3 ed. Boston. Allyn & Bacon. 1998.

CARDIERI, Giulia de A., ZAINA, Luciana M.. **Analyzing User Experience in Mobile Web, Native and Progressive Web Applications: A User and HCI Specialist Perspectives.** In Proceedings of the 17th Brazilian Symposium on Human Factors in Computing Systems (IHC 2018). ACM, New York, NY, USA. 2018.

CRISTAL, Maurício; WILDT, Daniel; PRIKLADNICKI, Rafael. **Usage of Scrum: practices within a global company.** Global Software Engineering, p. 222-226, 2008. Disponível em: <https://doi.org/10.1109/ICGSE.2008.34> Acesso em: 03 de abril de 2023.

DEMO, Pedro. **Metodologia do conhecimento científico.** São Paulo: Atlas, 2000.

DENZIN, Norman K. e LINCOLN, Yvonna. S. (Orgs.). **O planejamento da pesquisa qualitativa: teorias e abordagens**. 2. ed. Porto Alegre: Artmed, 2006.

DOLGAYA Darya; PHONGTRAYCHACK Anachack. **Evolution of mobile applications**. MATEC Web of Conferences 2018. Disponível em: https://www.matec-conferences.org/articles/mateconf/pdf/2018/14/mateconf_imet2018_01027.pdf. Acesso em: 21 de maio de 2020.

DRAKE, Joshua D., WORSLEY, John C. **Practical PostgreSQL**. Edição: First Edition. Editora: O'Reilly Media. 17 de janeiro de 2002.

FIGUEIREDO, Carlos M.; NAKAMURA, Eduardo Freire. **Computação móvel: novas oportunidades e novos desafios**. T&C Amazônia, ano 1, n. 2, jun. 2003.

FLETCHER, Kate. **Sustainable Fashion and Textiles: Design Journeys**. Earth Scan, London, England, 2008.

FRANCO, Pedro Augusto A. **Serviços de informação para dispositivos móveis: usos e possibilidades**. ENGI 2014 - Encontro Nacional: Gestão Políticas e Tecnologia de Informação. Disponível em: https://files.cercomp.ufg.br/weby/up/715/o/Dispositivos_Móveis_-_pronto.pdf. Acesso em: 2 de junho de 2020.

FONTENELLE, Isleide A. **CULTURA DO CONSUMO. Fundamentos e formas contemporâneas**. Rio de Janeiro. Editora FGV. 2017.

FONTOLAN, Aline C. M. **Modelo de apoio ao desenvolvimento de aplicativo mobile.** Disponível em:

<http://200.201.11.152/bitstream/handle/123456789/375/Modelo%20de%20Apoio%20ao%20Desenvolvimento%20de%20Aplicativos%20Mobile.pdf?sequence=1&isAllowed=y>. Acesso em: 2 de junho de 2020.

GAMMA, Erich, HELM, Richard, JOHNSON, Ralph, VLISSIDES, John. **Design Patterns: Elements of Reusable Object-Oriented Software**, 37 ed. Massachusetts. Pearson Education, 2009.

GIL, Antônio Carlos. **Como elaborar projetos de pesquisa**. 5. ed. São Paulo: Atlas, 2010.

GOOGLE. **Your First Progressive Web App**. 2019. Disponível em: <https://codelabs.developers.google.com/codelabs/your-first-pwapp/#0> Acesso em: 2 de Junho de 2020.

GOVINDAN, Kannan, SOLEIMANI, Hamed, & KANNAN, Devika. **Reverse logistics and closed-loop supply chain: A comprehensive review to explore the future**. European Journal of Operational Research, Elsevier, vol. 240(3), pages 603-626.2015.

HEITKÖTTER, Henning; HANSCHKE, Sebastian; MAJCHRZAK, Tim A. **Evaluating Cross-Platform Development Approaches for Mobile Applications**. J. Cordeiro and K.-H. Krempels (Eds.): WEBIST 2012, LNBIP 140, pp. 120–138, 2013.

JACOBSEN, Alessandra de Linhares. **Gestão por Resultados, Produtividade e Inovação**. Florianópolis, UFSC, 2009.

JÚNIOR, Elemar. **Uma breve entrevista com Vaughn Vernon sobre Domain-Driven Design**. EximiaCo. 2020 Disponível em: <https://eximia.co/uma-breve-entrevista-com-vaughn-vernnon-sobre-domain-driven-design/> Acesso em: 08 de maio de 2023.

KRACAUER, Siegfried. **The Mass Ornament**. Primeira edição 1963. Harvard University Press. 7 julho 1995.

MALAVOLTA, Ivano. **Beyond Native Apps: Web Technologies to the Rescue! (Keynote)**. *Mobile! '16*, Amsterdam, Netherlands. 31 de outubro de 2016.

MARCONI, Marina de Andrade; LAKATOS, Eva Maria. **Fundamentos de metodologia científica**. São Paulo: Editora Atlas, 2010.

MARTIN, Robert C. **Clean Code: A Handbook of Agile Software Craftsmanship**, Massachusetts. Pearson Education, 2008.

MARTIN, Robert C. **The Clean Architecture**. 13 de agosto de 2012. Disponível em: <https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html> Acesso em: 08 de maio de 2023.

MARTINS Celly de Siqueira; ANTONIO, André Lara Temple de; OLIVEIRA, Carlos Antônio. **Os desafios para a mobilização de aplicações baseadas em plataforma Web**. In: X ENCONTRO ANUAL DE COMPUTAÇÃO - EnAComp 2013, Fundação CPqD – Centro de Pesquisa e Desenvolvimento em Telecomunicações Campinas/SP, 2013.

MAURER, Frank; MANN, Chris. **A case study on the impact of scrum on overtime and customer satisfaction**. Agile Development Conference, IEEE Computer Society, 2005.

MEIRELLES, Fernando S. **Uso da TI - Tecnologia de Informação nas Empresas**. 33ª Edição, 2022 - FGVcia Disponível em: https://eaesp.fgv.br/sites/eaesp.fgv.br/files/u68/pesti-fgvcia-2023_0.pdf Acesso em: 30 de abril de 2023.

MILANI, André. **PostgreSQL - Guia do Programador**. Edição: 1. Editora: Novatec Editora, 1 de abril de 2008.

MODEFICA, FGVces, REGENERATE. **Fios da Moda: Perspectiva Sistêmica Para Circularidade**. São Paulo, 2021.

MODEFICA, FGVces, REGENERATE. **Possibilidades Para Moda Circular no Brasil - Padrões de Consumo, Uso e Descarte de Roupas**. São Paulo, 2020.

MULESOFT. **What is an API?**. Disponível em: <https://www.youtube.com/watch?v=s7wmiS2mSXY>. Acesso em: 9 de julho de 2020.

OCCHINO, Tom. **React Native: Bringing modern web techniques to mobile**. Facebook Engineering. Mar, 2015. Disponível em: <https://engineering.fb.com/android/react-native-bringing-modern-web-techniques-to-mobile/>. Acesso em: 9 de julho de 2020.

OSMANI, Addy. **A Pinterest Progressive Web App Performance Case Study**. Medium. Nov 29, 2017. Disponível em: <https://medium.com/dev-channel/a-pinterest-progressive-web-app-performance-case-study-3bd6ed2e6154> Acesso em: 9 de julho de 2020.

OSMANI, Addy. **The App Shell Model**. 2019. Disponível em: <https://developers.google.com/web/fundamentals/architecture/app-shell>. Acesso em: 9 de julho de 2020.

PUDER, Arno, TILLMANN, Nikolai, and MOSKAL, Michal. **Exposing native device APIs to web apps**. In Proc. 1st Int. Conf. on Mobile Software Engineering and Systems, pág 18–26. 2014.

RICHARDSON, Roberto Jarri. **Pesquisa social: métodos e técnicas**. São Paulo: Atlas, 1999.

RISSANEN, Timo. **Shaping Sustainable Fashion: changing the way we make and use clothes**. London. Earthscan, 2011.

SHEPPARD, Dennis. **Beginning progressive web app development creating a native app experience on the web**, 1st ed. Berkeley, CA. Apress, 2017.

SILVA, Edna Lúcia da.; MENEZES, Estera Muszkat. **Metodologia da pesquisa e elaboração de dissertação**. Florianópolis: UFSC/ PPGEPP/LED, 2000.

SILVA, Marcelo Moro da SANTOS, PRADO Marilde Terezinha. **Os Paradigmas de Desenvolvimento de Aplicativos para Aparelhos Celulares**. Tecnologias, Infraestrutura e Software, São Carlos, v. 3, n. 2, p. 167-170, mai-ago. 2014.

SOUZA, Victor M. S. de. **Lâmpada Mágica - Framework para criação de ambientes de front-end.** Disponível em: <https://bsi.uniriotec.br/tcc/textos/201601Magalhaes.pdf> Acesso em: 9 de julho de 2020.

SUTHERLAND, Jeff. **Scrum: A Arte de Fazer o Dobro do Trabalho na Metade do Tempo**, 2 ed. LeYa, 2016.

TAVARES, A. **Gerência de Projetos com PMBOK e SCRUM** – Um estudo de caso. Monografia (TCC) — Faculdade Cenecista Senhora dos Anjos, 2008.

VERNON, Vaughn. **Implementing Domain-Driven Design**, 2 ed. Massachusetts. Pearson Education, 2013.

APÊNDICES

APÊNDICE A – *Links de acesso ao repositório do projeto no GitHub*

Repositório *Back-end*: <https://github.com/GustavoEklund/ucycle-api>

Repositório *Front-end*: <https://github.com/GustavoEklund/ucycle-web>

Repositório aplicativo Android "container" para o *Front-end*:
<https://github.com/GustavoEklund/ucycle-android-mobile>

APÊNDICE B – Links de acesso ao vídeo de demonstração do uCycle

Vídeo de demonstração do processo de cadastro do usuário

https://drive.google.com/file/d/1JmsOQ_-M3CiAIGRqpwz_FABH9zCvi9Pm/view?usp=sharing

Vídeo de demonstração do processo de compra de produtos

<https://drive.google.com/file/d/1Ji2xUXto9ZAcmjhSgiQRTjqVpr1-6hs7/view?usp=sharing>