

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

RODRIGO YUKIO OKIDO

**ARShop – Um aplicativo de e-commerce mobile com
Realidade Aumentada para móveis**

Monografia apresentada como requisito parcial para
a obtenção do grau de Bacharel em Ciência da
Computação.

Orientador: Profa. Dra. Renata Galante

Porto Alegre
2020

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Rui Vicente Opperman

Vice-Reitora: Prof^a. Jane Fraga Tutikian

Pró-Reitor de Graduação: Prof. Vladimir Pinheiro do Nascimento

Diretor do Instituto de Informática: Prof^a. Carla Maria Dal Sasso Freitas

Coordenador do Curso de Ciência da Computação: Prof. Sérgio Luis Cechin

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

AGRADECIMENTOS

Agradeço primeiramente a meus pais, Ricardo e Alzira, por me proporcionarem desde o início uma educação de bastante ensino e qualidade. A minha família e amigos por me apoiarem sempre nos momentos difíceis, proporcionando conselhos e muito conforto. E claro, também por me proporcionar diversão e muitas risadas nos momentos que estamos todos reunidos. Ao meu cachorro por sempre estar ao lado fornecendo seu carinho e companheirismo independente do momento ser de alegria ou tristeza.

Agradeço as pessoas de todos setores do curso de Ciência da Computação da UFRGS por darem todo o suporte necessário para realizarmos tranquilamente a graduação da melhor forma possível. Principalmente aos professores e professoras que são aqueles com quem mais temos contato durante todo o decorrer do curso.

Um agradecimento especial ao professor Valter Roesler por me dar uma oportunidade incrível em trabalhar em um projeto totalmente novo no PRAV. Foram quase 3 anos de muito conhecimento e experiência adquirido, além de muitas amizades e histórias construídas. Certamente nunca esquecerei de tudo que passei durante todo este período, onde fui muito feliz e totalmente grato. Um agradecimento também especial ao Mário Gasparoni Junior, pessoa que me coordenou durante todo o período no projeto do PRAV. Sempre amigável, aberto ao diálogo, dando muito auxílio e conhecimento durante todo o decorrer do projeto. Certamente muito do conhecimento obtido, e que certamente será utilizado no mercado em outras oportunidades, passou por esta pessoa.

E por fim, um agradecimento especial a professora Renata Galante por me acolher e aceitar em orientar este trabalho de conclusão que será apresentado nas próximas páginas. Sempre querida, aberta ao diálogo, e uma das professoras mais incríveis com quem tive o prazer de ter sido aluno durante o curso de graduação.

RESUMO

Cada vez mais usuários realizam compras na *internet* através de seu computador e smartphone. Só em 2019, houve um lucro de mais de 60 bilhões de reais no Brasil neste setor. Apesar da comodidade e praticidade de comprar *online*, existem alguns dados interessantes sobre o comércio eletrônico. Segundo pesquisa feita pela comScore e UPS nos Estados Unidos, cerca de 63% dos americanos checam a política de retorno antes de comprar um produto. E 48% comprariam mais com aqueles que trouxessem mais facilidades e menos dor de cabeça na hora de uma possível devolução, seja por insatisfação ou por defeito do produto. Isto mostra que o consumidor está consciente na hora de comprar na Internet. Se considerarmos o mundo ideal de que o produto chega ao cliente sem defeito de fábrica, imaginar que ele devolva o produto por não ser o que esperava é uma experiência bastante frustrante. O cliente terá de solicitar a devolução do produto, e dependendo pagar a mais por isto. Enquanto a empresa precisará devolver o dinheiro obtido em cima da venda. Normalmente, empresas precisam construir políticas de devolução transparentes e da melhor forma possível para evitar qualquer tipo de constrangimento. E o cliente, se não se atentar a essas políticas, pode acabar tendo dor de cabeça para realizar uma devolução. A proposta deste trabalho é justamente atacar este ponto. Fazer com que o cliente sinta mais segurança em decidir de que o produto que está adquirindo é de fato o que espera. Apesar do tema ser tratado de maneira genérica até o momento, neste trabalho teremos um foco em móveis. Normalmente são produtos que geram dúvidas pela diversidade de tamanhos, modelos e formatos. Isto se dará através de um aplicativo e-commerce para Android onde o diferencial será o recurso de realidade aumentada. Através da câmera do smartphone, o usuário poderá visualizar o produto que deseja adquirir no ambiente em que estiver. Além disso, a aplicação tem como objetivo tentar oferecer uma boa experiência com uma interface amigável e intuitiva.

Palavras-chave: E-commerce. Compras. Online. Aplicativo. Android. Comércio eletrônico. User experience. Móveis.

ARSHOP – E-commerce mobile application with Augmented Reality

ABSTRACT

More and more users make purchases on the internet through their computer and smartphone. Only in 2019, this sector made a profit of more than 60 billion Real in Brazil. Despite the convenience and practicality of purchasing online, there are some interesting data about e-commerce. According to a survey by comScore and UPS in the United States, about 63% of Americans check the return policy before purchasing a product. And 48% would buy more from those who brought easiness and less headaches when making a possible product return, either due to dissatisfaction or product defect. This shows that the consumer is conscious when purchasing on the Internet. If we consider the ideal world that the product reaches the customer without a factory defect, imagining a possible product refund for not being what he expected is a very frustrating experience. The customer will have to request the return of the product, and depending of the situation overpay for this. And for the company, they will need to return the money obtained upon sale. Usually, companies need to build transparent return policies and in the best possible way to avoid any kind of embarrassment. And the customer, if they don't pay attention to these policies, may end up having a headache to make a return. The purpose of this work is to attack this point. Make the customer feel confident to decide that the product they are purchasing is in fact what they expect. Although the topic being treated in a generic way so far, in this paper we will focus on furniture's. They are usually products that raise doubts due to the diversity of sizes, models and formats. This will be shown with an e-commerce application for Android where the main feature will be the augmented reality. Through the smartphone's camera, the user will be able to preview the product desired in the environment where he is located at. In addition, the application aims to try to offer a good experience with a friendly and intuitive interface.

Keywords: E-commerce. Shopping. Online. Application. Android. E-commerce. User experience. Furnitures.

LISTA DE FIGURAS

Figura 2.1 – Diagrama Geral MVC	15
Figura 2.2 - Exemplo genérico de uma estrutura de projeto no git	16
Figura 2.3 - Parte da página para desenvolvedores do Mercado Livre.....	19
Figura 2.4 - Exemplo de empresas que vendem na plataforma.....	20
Figura 2.5 - Exemplo de realidade aumentada no jogo Pokémon Go	22
Figura 2.6 - Exemplo de Realidade Aumentada de estabelecimentos nas ruas.....	23
Figura 2.7 - Exemplo genérico de um arquivo JSON.....	24
Figura 2.8 - Exemplo de um site com design não recomendado.....	26
Figura 2.9 - Relação das cores e seu significado no <i>e-commerce</i>	26
Figura 3.1 - Imagens ilustrativas do IKEA Place	29
Figura 3.2 - Imagens ilustrativas do DecorMatter	30
Figura 3.3 – Imagens ilustrativas Houzz.....	31
Figura 4.1 – Diagrama de Classes simplificado das principais classes.	37
Figura 4.2 - Trecho da página da documentação da API.....	42
Figura 4.3 - Link de um produto qualquer. Definido através do seu <i>productID</i>	42
Figura 4.4 – Trecho de informação de um produto do Mercado Livre no formato JSON.....	43
Figura 4.5 - Fluxograma de comunicação com a API do Mercado Livre.....	43
Figura 4.6 - Tipos de <i>Activities</i> que o Android Studio oferece	45
Figura 4.7 - Exemplo de CardView	46
Figura 4.8 - Exemplo de BottomNavigationView	47
Figura 4.9 - Exemplo genérico do ARCore exibindo o símbolo do Android	48
Figura 4.10 - Exemplo de aplicação do RecyclerView.....	49
Figura 4.11 - Exemplo de Dialog	50
Figura 4.12 – Exemplo de StateProgressBar.....	52
Figura 4.13 - Tela de Login e cadastro	53
Figura 4.14 - Tela inicial pós login e tela de produtos.....	54
Figura 4.15 - Tela de detalhe de um produto	55
Figura 4.16 – Telas de visualização em AR.	56
Figura 4.17 – Exemplo de um móvel reproduzido em AR.	57
Figura 4.18 - Tela “Meu Carrinho” e “Favoritos” do usuário.....	58
Figura 4.19 - Tela inicial de processo de compra. Dados totalmente genéricos	59
Figura 4.20 - Telas de finalização de compra.....	60
Figura 4.21 - Telas de Perfil do Usuário e tela da opção “Meu Dados” (Configurações).....	61
Figura 4.22 - Tela de opções “Meus Endereços” e “Meus cartões”	62
Figura 4.23 – Tela de opções “Minhas Compras”	63
Figura 5.1 – Perfil Demográfico (Parte 1 de 3)	67
Figura 5.2 – Perfil Demográfico (Parte 2 de 3)	68
Figura 5.3 – Perfil Demográfico (Parte 3 de 3)	69

LISTA DE TABELAS

Tabela 2.1 – Resumo de cada tecnologia utilizada neste trabalho	28
Tabela 3.1 – Tabela comparativa entre os aplicativos	31
Tabela 5.1 – Resultado do questionário SUS.	70

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
APP	<i>Application</i>
AR	<i>Augmented Reality</i>
GPS	<i>Global Positioning System</i>
ID	<i>Identity Document or Identifier</i>
IDE	<i>Integrated Development Environment</i>
JSON	<i>JavaScript Object Notation</i>
JVM	<i>Java Virtual Machine</i>
MVC	<i>Model-View-Controller</i>
QRCODE	<i>Quick Response Code</i>
SUS	<i>System Usability Scale</i>
URL	<i>Uniform Resource Locator</i>
UX	<i>User Experience</i>
XML	<i>Extensible Markup Language</i>

SUMÁRIO

1	INTRODUÇÃO	10
1.1	Descrição e objetivo do aplicativo	12
1.2	Organização dos capítulos	13
2	TECNOLOGIAS E RECURSOS UTILIZADOS	14
2.1	<i>Model-View-Controller</i> (MVC)	14
2.2	Java	15
2.3	Git	15
2.4	Android / Android Studio	17
2.5	API Mercado Livre	17
2.5.1	Porque Mercado Livre para o aplicativo?	17
2.6	Realidade Aumentada (AR)	21
2.7	JSON	23
2.8	User Experience (UX)	24
2.9	Tabela final das tecnologias utilizadas	28
3	TRABALHOS RELACIONADOS	29
3.1	IKEA Place	29
3.2	DecorMatter	30
3.3	Houzz	31
3.4	Comparativo	31
4	METODOLOGIA DE DESENVOLVIMENTO	34
4.1	Visão Geral	34
4.2	Requisitos Funcionais e Não Funcionais	35
4.3	Arquitetura	36
4.3.1	Comunicação com a API do Mercado Livre	42
4.4	Recursos de desenvolvimento utilizados	44
4.5	Interfaces	52
4.6	Considerações Finais	63
5	EXPERIMENTOS	65
5.1	O Experimento	65
5.2	Atividades Realizadas	66
5.3	Resultados	67
6	CONCLUSÃO	72
	REFERÊNCIAS	74
	APÊNDICE A – QUESTIONÁRIO FINAL APLICADO AOS USUÁRIOS	77

1 INTRODUÇÃO

E-commerce, também conhecido como comércio eletrônico, trata-se de um modelo de negócio de compra e venda de produtos ou serviços totalmente *online*. Uma vez definido o termo, é notório que estamos rodeados deste tipo de negócio no mundo através da *web* e aplicativos para smartphones. Ao invés de irmos fisicamente a uma loja olhar um produto, optar por comprar, pagar por ele e sair, fazemos todo esse processo virtualmente. No Brasil, segundo dados do 39 Webshoppers¹ publicado no primeiro semestre de 2018 feito pela Ebit, 58 milhões de brasileiros efetuaram uma compra via *internet*, tendo um lucro aproximado de 53.2 bilhões de reais neste mesmo ano. Em termos comparativos, o lucro em vendas online em 2017 e 2019 correspondem 47.7 bilhões e 61.9 bilhões de reais respectivamente. O que demonstra que a cada ano, mais usuários estão entrando e comprando na *internet*.

Apesar dos dados, não é novidade hoje que compras *onlines* tem bastante popularidade. Dentre as razões é a comodidade. Não precisar sair de casa e ter ao seu alcance qualquer produto ou serviço em poucos cliques, torna esse tipo de compra extremamente atrativo para qualquer consumidor. E pelo lado de quem vende, qualquer empresário terá provavelmente menos custo, uma vez que não necessariamente precisará de um local físico para realizar a amostra do que deseja vender. Ao invés disso, pode fazer toda a logística virtualmente, precisando apenas de um local físico para o estoque.

Entretanto, apesar das vantagens do mundo *e-commerce*, existem alguns pontos que exigem atenção. Um exemplo é a segurança, onde temos de ter a plena certeza que o local de onde vamos comprar e colocar nossos dados é de fato legítimo e seguro. Existem inúmeros *websites* falsos espalhados na *internet* prometendo diversos tipos de sonhos a preços maravilhosos (normalmente abaixo do preço de mercado). Porém, não passam de armadilhas com intuito apenas de roubar seus dados, principalmente bancários, para uso criminoso. Para quem deseja vender, um problema é a concorrência. Não há quase nada hoje que desejamos comprar que já não seja vendido por alguma empresa. Para se destacar exigirá um diferencial em outros pontos, seja no atendimento, tempo de entrega ou preços mais competitivos por exemplo. Um outro ponto, o qual daremos foco, é o fato de não termos como sentir o produto nas próprias mãos. Neste quesito, um cliente pode decidir não comprar, ou se arrepender de ter comprado um produto por ver que o mesmo não atendia às suas expectativas.

¹ <http://www.dlojavirtual.com/e-commerce/e-commerce-no-brasil-entenda-esse-cenario/> - Acesso em março de 2019

Considerando o último ponto citado, na maioria dos casos quando vamos comprar um produto online, só podemos nos basear em fotos e informações do fabricante. Entretanto, existem momentos que apenas isto não basta para fazer um cliente decidir se vale a pena comprar determinado produto. E o que normalmente ocorre é buscarmos comentários, vídeos ou análises de pessoas que já compraram e tiveram a experiência com o produto para tirar mais conclusões sobre ele.

Como proposta para tentar resolver o problema, é apresentado um aplicativo de e-commerce para smartphones chamado ARShop. ARShop é desenvolvido para o sistema operacional Android, e tem como diferencial o recurso da realidade aumentada (utilizado no conhecido jogo Pokémon Go). Basicamente, realidade aumentada é definido como uma integração de elementos virtuais com o mundo físico. De maneira computacional, é possível trazer informações ou objetos virtuais, fazendo com que haja uma nova perspectiva do mundo real do usuário. Essa alteração do mundo real, se deve ao fato de que os elementos e as informações vindas não são simplesmente exibidas ou dispostas na tela, mas sim de maneira imersiva de forma que pareça que estes elementos façam parte e interajam com o ambiente físico. Este conceito será maior explorado no decorrer deste trabalho para melhor entendimento.

Outro ponto a considerar no ARShop é sua base de dados. A base de dados utilizada como fonte é o Mercado Livre. Além de conter uma ótima API (*Application Programming Interface*) aberta para desenvolvedores, contém um imenso catálogo de produtos dos mais variados setores. A empresa por ser também um e-commerce com uma base bastante sólida de usuários, diversas empresas grandes também utilizam a plataforma para vender seus produtos (Google, Playstation, Polishop, Phillips, Boticário, e muitos outros). Levando estes dois pontos do aplicativo em consideração, é possível notar que temos a disposição a possibilidade de visualizar uma série de produtos usando a realidade aumentada. Vendo por exemplo como ficaria no seu ambiente desejado, assim como suas dimensões. O cliente pode ter maior segurança e certeza de que o produto almejado atende às suas expectativas, o que possivelmente diminui as chances do cliente devolver o produto por uma possível insatisfação. Para que isto funcione de maneira interessante e agradável, o aplicativo almeja fornecer uma experiência do usuário de maneira positiva. Tentando dar uma usabilidade simples e intuitiva.

Atualmente, já existem algumas aplicações que utilizam a realidade aumentada para vender móveis. Com isto, o usuário pode utilizar seu celular para comprar e visualizar uma série de produtos através da câmera para verificar como seria ter o produto desejado em tamanho real. Porém, muitas delas são de empresas estrangeiras, o que dificulta o acesso para o público Brasileiro. Além disto, grande parte mantém um foco maior no setor de móveis. E com o

ARShop, a ideia foi desenvolvida desde o seu princípio para ir um pouco mais além deste setor no uso da realidade aumentada. Alguns exemplos de aplicações serão fornecidos em mais detalhes no Capítulo 3. Na próxima subseção, será aprofundado um pouco mais a descrição e o objetivo do aplicativo.

1.1 Descrição e objetivo do aplicativo

O objetivo principal do ARShop é ser um aplicativo de e-commerce. Ao utilizar, o usuário pode visualizar uma série de produtos, ver os detalhes de cada produto caso desejar, além de poder salvar seus produtos favoritos, e claro, comprar. Caso seja visualizado um produto de maneira detalhada, o usuário tem a sua disposição um botão para visualizar o produto em realidade aumentada. Com o uso da câmera, o usuário pode visualizar o modelo do produto desejado em 3D no ambiente local. Para utilizar o ARShop, é necessário realizar um login para garantir que o usuário tenha pelo menos os dados básicos que possam garantir sua legitimidade para comprar no aplicativo.

Como já mencionado, é utilizado a API do Mercado Livre, que é disponível e totalmente aberta para desenvolvedores. Nesta API, é possível obter uma imensa base de dados onde podemos manipular de diversas maneiras. Gerenciamento de produtos, categorias, vendas, análise de opiniões dos clientes e consulta de clientes são apenas alguns exemplos dentre os diversos itens que a API nos fornece. Todo dado requisitado, o servidor retorna a informação no formato JSON (*JavaScript Object Notation*). Supondo, por exemplo, que seja solicitado ao servidor um produto qualquer, o servidor devolve todas as informações a respeito deste produto (nome, unidades disponíveis, marca, dimensão, etc.).

Contando que o Mercado Livre possui uma base de produtos gigantesca, a proposta do *app* não tem, no momento, intenção de extrair todos os produtos da base de dados, e sim apenas produtos seletos. O foco, neste caso, é em móveis. A razão se deve ao fato de que móveis são pouco mais difíceis de se realizar uma decisão da compra apenas se baseando em fotos. Precisamos ter noção de suas dimensões para saber se vai atender naquele espaço que você gostaria de por. O que nos exige usar a fita métrica, medir cada ambiente, e assim termos tudo anotado para garantir que tal opção é válida. Um outro ponto a considerar é o design. Existem diversos tipos de modelos, tamanhos e cores diferentes, que torna difícil decidir qual será o ideal. Como não é possível termos um *preview* do móvel, resta apenas a imaginação de vermos como o móvel seria no ambiente desejado. Com isto, algumas perguntas podem surgir: Será

que este modelo é de fato a melhor opção? Será que este móvel vai combinar com o resto? Será que encaixa nesta parede? É justamente nestas perguntas, que dificultam a decisão do cliente na hora da compra, que este *app* deseja auxiliar.

Nos experimentos realizados com usuários reais, que poderão ser conferidos em maior detalhe no Capítulo 5, foi observado que todos os usuários pelo menos já compraram em algum momento um produto na internet. Entretanto, apenas 50% responderam “sim” para móveis. Um grande empecilho dos usuários foi justamente em relação ao seu tamanho. Onde não saber se o tal móvel vai de fato caber no ambiente desejado. Porém, ao serem questionados sobre a possibilidade de pré-visualizar o móvel desejado, 100% das pessoas responderam que se sentiriam mais confiantes e seguros em adquirir e comprar o produto. No experimento, as pessoas eram convidadas a testar basicamente grande parte das funcionalidades do aplicativo, e claro, seu diferencial que é possibilidade de ver o móvel em realidade aumentada.

1.2 Organização dos capítulos

Este trabalho está dividido em 6 Capítulos. Sendo este aqui o primeiro Capítulo, o Capítulo 2 aborda as tecnologias utilizadas para o desenvolvimento deste trabalho. No Capítulo 3 são apresentados e comparados os trabalhos relacionados a esta proposta. No Capítulo 4, todo o processo detalhado de desenvolvimento é descrito. No capítulo 5, são apresentados experimentos e dados reais avaliando os resultados do impacto do aplicativo desenvolvido. E por fim, o Capítulo 6 apresenta as conclusões gerais, detalhando limitações e possíveis melhorias futuras.

2 TECNOLOGIAS E RECURSOS UTILIZADOS

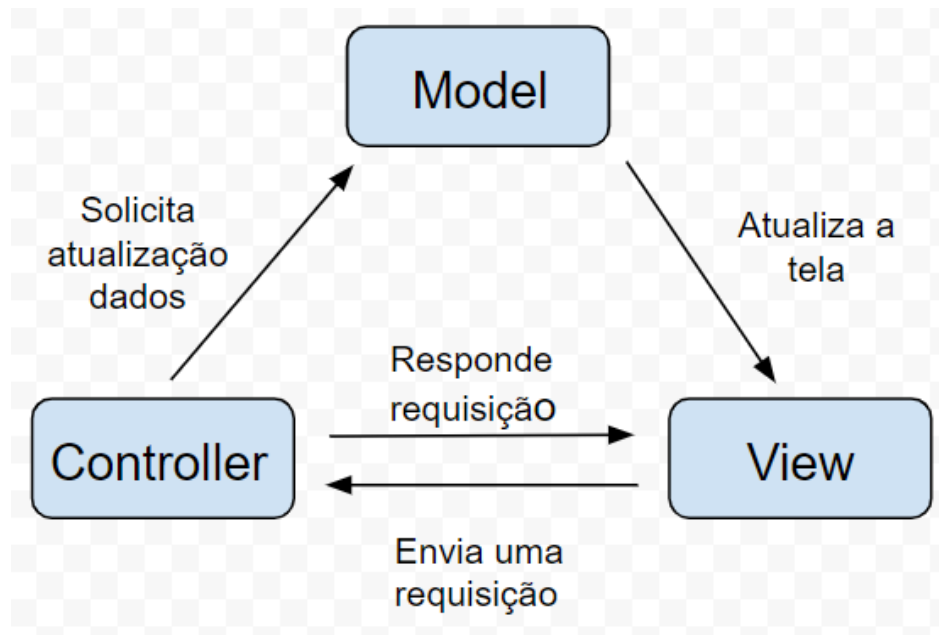
O objetivo deste Capítulo é descrever todas as tecnologias e recursos utilizados no desenvolvimento do ARShop. Definindo desde a forma de desenvolvimento tomado, a tudo que foi utilizado para o produto final sair. Além disto, uma explicação da escolha dos recursos utilizados será apresentada.

2.1 *Model-View-Controller* (MVC)

Mais popularmente conhecido como MVC, é um padrão de design de software que se divide em 3 camadas que interconectam: *model*, *view* e *controller*. Surgido em 1979 por Trygve Reenskaug, é um padrão bastante usado em projetos de desenvolvimento de softwares e nos fornece uma visão clara da divisão do *back-end* (funcionalidades, banco de dados, ...) e *front-end* (Interfaces). Cada uma das camadas possui uma função que nos permite separar cada componente e colocar no seu lugar. O *model* é a parte responsável por coordenar os dados e a lógica da aplicação. Recebe os dados vindo da camada *controller*, como informações fornecidas pelo usuário, e realiza devidas ações e atualizações com estes dados, que serão exibidos depois pela camada *view*. Falando em *controller*, este é o próximo elemento. No *controller*, como o nome sugere, é realizado todo o controle de fluxo de dados e informações. É nesta camada onde o sistema deve entender o que cada elemento vai fazer, quando vai fazer e como deve fazer. Transformando isto em um exemplo, quando um usuário executa uma ação, como clicar em um botão de um determinado sistema, é essa camada que deve entender o que fará com essa ação desejada e repassar de volta a resposta para o usuário. E por fim, a *view* é a mais simples de todas, onde simplesmente define tudo que o usuário enxerga na tela. A decisão por se adotar o padrão MVC para este projeto é justamente pelas vantagens que este padrão nos fornece, o que inclui reaproveitamento do código, mais fácil manutenção e documentação do código, além de possibilidade de se implementar cada camada de maneira independente.

No Android Studio, por cada interface ser programada em XML e já ser separada dos arquivos onde serão implementadas as funcionalidades, aplicando o MVC a camada *view* já automaticamente existe, atualizando à medida que uma nova interface é construída. A IDE com isto, fornece motivos para o usuário optar por este tipo de arquitetura. O que precisamos organizar é basicamente as camadas restantes. Na Figura 2.1, podemos visualizar um diagrama geral de como funciona o MVC conforme foi explicado.

Figura 2.1 – Diagrama Geral MVC



Fonte: O Autor

2.2 Java

Criado na década de 90 pela Sun Microsystem e adquirida pela Oracle Corporation em 2008, Java é uma linguagem de programação orientada a objetos fortemente tipada. Totalmente compilada e interpretada por uma máquina virtual chamada JVM (Java Virtual Machine). Java possui uma característica bastante forte na questão de portabilidade. Uma vez programado, é possível executar em qualquer sistema. Considerando que o aplicativo foi desenvolvido para Android, onde é possível programar também na linguagem Kotlin e C++, a linguagem Java foi optada pela familiaridade e experiência do autor deste trabalho comparada as demais. A versão do Java usada para desenvolver é a versão 8.0.

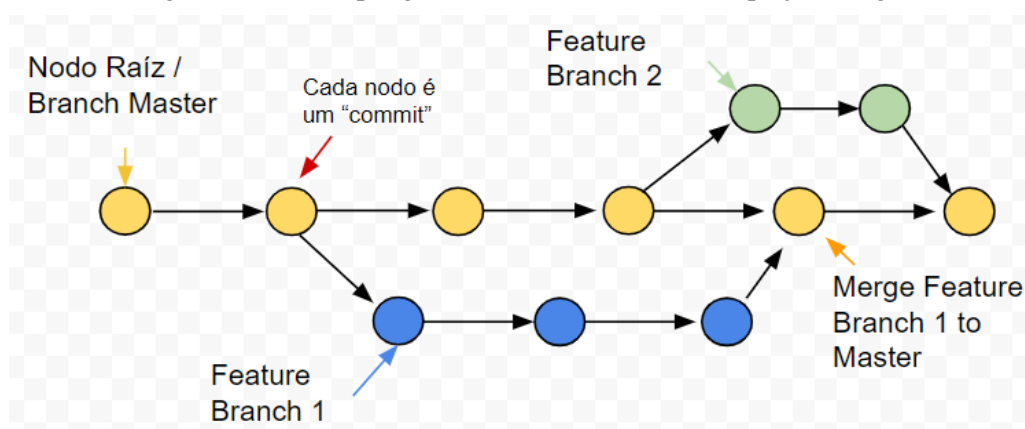
2.3 Git

Criado por Linus Torvalds, git é um sistema de controle de versões distribuído usado, principalmente, no desenvolvimento de software. Pode ser usado para termos registro do histórico de qualquer modificação ocorrido em um arquivo ou conjunto de arquivos de determinado projeto. Este conceito ou lugar que temos o registro histórico e todo o projeto armazenado chamamos de repositório. Git tem uma estrutura de registros similar a uma árvore,

iniciando sempre com um nodo inicial que chamamos de nodo raiz. Este nodo raiz está composto dentro de uma linha de desenvolvimento principal que chamamos de *branch master*. A medida que novas modificações são realizadas e registradas, um novo nodo é gerado e conectado ao anterior contendo todas as modificações realizadas a partir do último nodo. Esta ação é o que chamamos de *commit*.

Uma das grandes vantagens que o git proporciona é a flexibilidade, uma vez que a partir de um nodo, podemos navegar para qualquer nodo registrado. Além disso, é possível criar outros nodos filhos que chamamos de *features branches*. Neles podemos realizar qualquer modificação no código sem afetar a *branch master*. Consequentemente, em caso de eventuais problemas no desenvolvimento, permite que a equipe simplesmente retorne para *branch* ou *commit* mais estável do projeto. Caso contrário, é possível enviar todas as modificações destas *features branches* em um novo nodo na linha principal *branch master*, conceito que chamamos de *merge*. É notório que além da flexibilidade, o git traz segurança para toda a equipe, uma vez que erros podem ser reversíveis. Na Figura 2.2, temos uma imagem ilustrando basicamente o que foi explicado. O git é disponível em diversas plataformas, como Github, Bitbucket, GitKraken, etc. Cada uma tem suas particularidades, vantagens e desvantagens. Seja com repositórios particulares gratuitos, tamanhos variados de armazenamento, suporte, entre outros. Cabe apenas a empresa ou desenvolvedor decidir em qual delas atende melhor suas necessidades para hospedar seus projetos. Por já possuir diversos outros projetos, e familiaridade com a plataforma, este trabalho foi optado em ser totalmente armazenado e disponível no Github².

Figura 2.2 - Exemplo genérico de uma estrutura de projeto no git



Fonte: O Autor

² <https://github.com/> - Acesso em janeiro de 2020

2.4 Android / Android Studio

Sistema operacional desenvolvido pela Google para dispositivos móveis. Tendo sua primeira versão originada e comercializada em 2008, hoje é o sistema operacional que conta com 2.5 bilhões de dispositivos ativos, sendo o principal concorrente do sistema operacional da *Apple*, o *IOS*.

Android Studio³ basicamente é a IDE (*Integrated Development Environment*) oficial da Google utilizada para desenvolver em cima do seu sistema operacional. Baseado em uma IDE chamada IntelliJ IDEA, fornece praticamente todas as ferramentas necessárias para o desenvolvimento de qualquer aplicativo Android. E caso ainda falte recursos, a IDE possui diversos plugins disponíveis que podem auxiliar a resolver. Aceita desenvolver nas linguagens Java, Kotlin e C++. E para as telas, no caso as interfaces, o XML (*eXtensible Markup Language*) é utilizado. Por ser a IDE oficial da Google, já preparada para compilar e testar, além de fornecer todas as ferramentas necessárias para o desenvolvimento do aplicativo, esta foi a IDE utilizada para implementar e desenvolver o ARShop. A versão do Android Studio usado foi v.3.5.3 (dezembro 2019).

2.5 API Mercado Livre

Mercado Livre⁴ é uma empresa de comércio eletrônico para que pessoas físicas e empresas possam anunciar, comprar e vender produtos pela *internet*. Tendo sua sede localizada em Buenos Aires, Argentina, a empresa conta hoje com cerca de 4000 funcionários e mais de 170 milhões de usuários na América Latina. Hoje, a empresa é atuante em diversos países da América Latina e Central: Argentina, Brasil, México, Uruguai, Colômbia, Venezuela, Chile, Equador, Peru, Costa Rica, República Dominicana e Panamá.

2.5.1 Porque Mercado Livre para o aplicativo?

Um dos motivos para a escolha do Mercado Livre ser utilizado como base, é pela quantidade de vendedores autônomos e empresas que o site possui. Inúmeras empresas dos

³ <https://developer.android.com/studio> - Acesso em janeiro de 2020

⁴ <https://www.mercadolivre.com.br/> - Acesso em janeiro de 2020

mais variados setores realizam venda de seus produtos, além de claro, qualquer usuário poder anunciar para vender nela. Neste trabalho, não seria vantajoso e barato realizar um aplicativo para vender produtos próprios, uma vez que isso dependeria de fatores como fornecedores, local para estoque e muitas parcerias para o negócio começar a tomar forma. Ao invés disso, estamos tomando algo que já existe e está ao alcance de praticamente todos, e agregando uma experiência nova. O fato do Mercado Livre fornecer uma API⁵ bastante rica, conseguimos ter acesso a diversos tipos de dados dentro da ferramenta. Agregando a experiência da realidade aumentada com o uso desta API, qualquer empresa e até mesmo os próprios vendedores independentes, caso o desejassem, poderiam usufruir deste diferencial para vender seus produtos.

Alguns recursos que a API fornece:

1. Usuários e Aplicativos – Fornece meios para adquirir informações de conta e configurações do aplicativo.
2. Categorias e Publicações – Auxilia na obtenção das categorias de produtos disponíveis no site.
3. Itens e Buscas – Fornece auxílio para obter resultados na busca de itens. Seja por categorias, nomes, por vendedor, etc.
4. Pedidos e Opiniões – Auxilia na obtenção de tudo relacionado a pedidos de um vendedor ou comprador. Busca de pedidos, métodos de pagamento e feedbacks recebidos de um comprador ou vendedor relacionado a um pedido são algumas possibilidades que a API fornece.
5. Envio – Fornece todos os dados relacionados ao envio de um produto. Seja dados do envio, custo de envio, cálculo de custo do envio por país, modos e serviços de envio, entre outros.

Como podemos ver, a API do Mercado Livre é bastante extensa e fornece diversos recursos para programadores poderem desenvolver em cima. Os recursos citados são apenas alguns dentro do que a ferramenta disponibiliza. Toda explicação de como se usa a API está documentada na página oficial para desenvolvedores no mercado livre. Logo abaixo na Figura 2.5, é possível ter uma ideia dos recursos que a API oferece. No caso deste trabalho, o foco é

⁵ https://developers.mercadolivre.com.br/pt_br/api-docs-pt-br - Acesso em janeiro de 2020

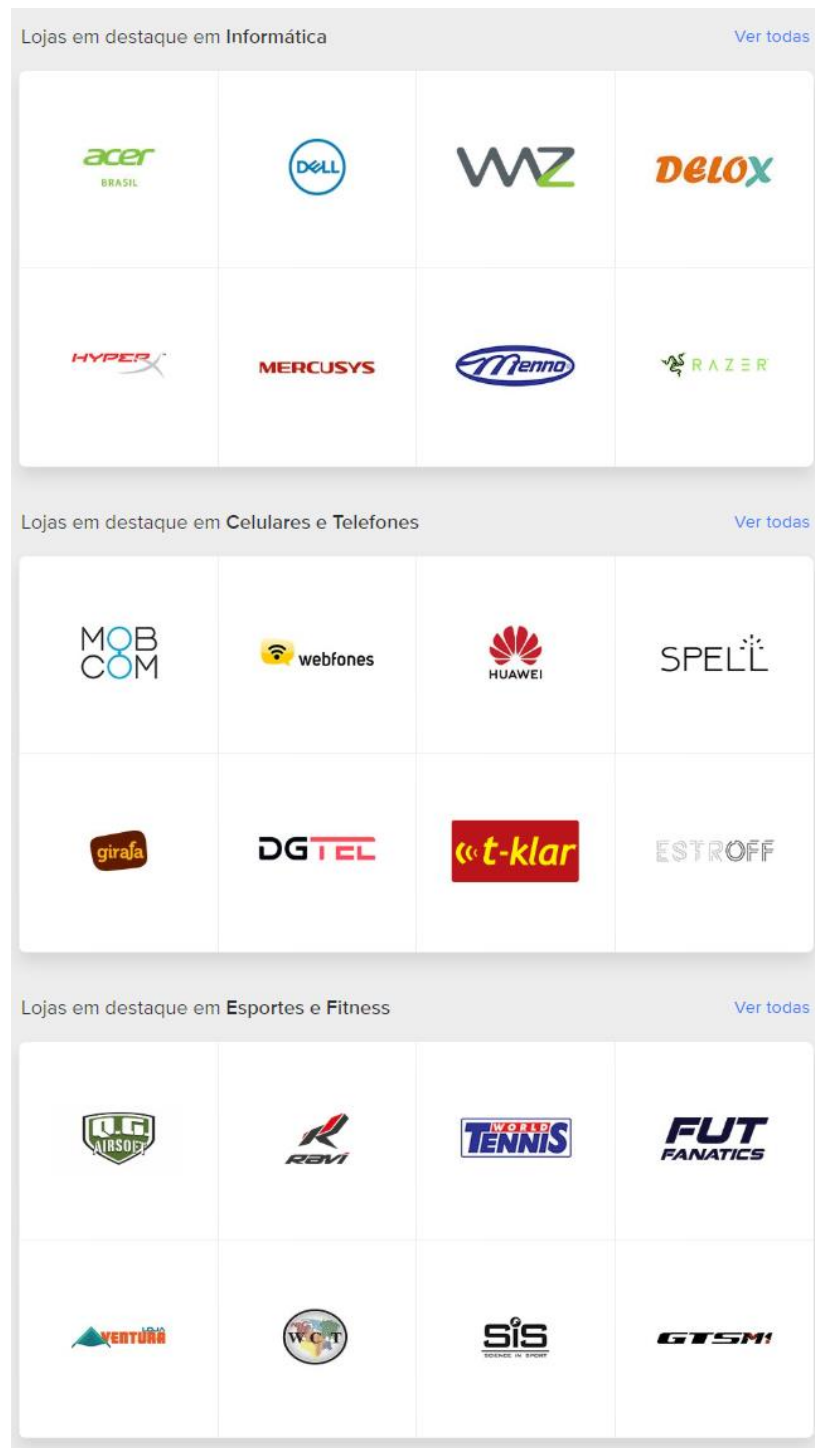
realizar apenas requisições dos produtos que o site oferece, no caso móveis. A comunicação com a API é explicada mais adiante na subseção 4.3.1 do Capítulo 4.

Figura 2.3 - Parte da página para desenvolvedores do Mercado Livre



Fonte: (Developers Mercado Livre, 2020)

Figura 2.4 - Exemplo de empresas que vendem na plataforma



Fonte: (Mercado Livre, 2020)

2.6 Realidade Aumentada (AR)

Termo bastante confundido com Realidade Virtual, Realidade aumentada ou AR (*Augmented Reality*) é a integração de elementos virtuais com o mundo real. Segundo pesquisador Ron Azuma, do artigo “*A Survey of Augmented Reality*” da *Foundations and Trends in Human-Computer Interaction*, realidade aumentada se define em 3 chaves: combinar o mundo real do virtual, interatividade em tempo real, e ser registrado em 3D. Enquanto a realidade virtual modifica e insere o usuário em um ambiente totalmente virtual, a realidade aumentada dispõe de objetos ou informações virtuais que passam a integrar e fazer parte do mundo físico do usuário. É necessário uma tela onde os elementos virtuais contidos são gerados computacionalmente de maneira interativa que responda aos comandos do usuário em tempo real. Além disto, precisa ter um sistema capaz de saber a localização (posição) do usuário, que permita saber seu ponto de vista para que exiba a imagem virtual de maneira fixa e correta no ambiente real. Reafirmando o conceito com termos pouco mais populares, com o auxílio de uma câmera, conexão com a internet, e um programa, além de recursos como giroscópio, acelerômetro e GPS (*Global Positioning System*), realiza-se a projeção de elementos virtuais em um cenário real, onde acrescentam-se informações com intuito de dar mais conteúdo e interatividade ao ambiente. Uma experiência prática onde este conceito é utilizado são nos QR Codes (*Quick Response Codes*) para pontos turísticos de uma cidade. Através da câmera do *smartphone* e conexão com a *internet*, o usuário ao apontar para o QR Code recebe informações relevantes a respeito de lugares turísticos que pode visitar. Além de possivelmente mostrar a rota para chegar ao local, podem ser exibidos a história, curiosidades, e o que possui de interessante ao redor desta localização. Atualmente, o principal campo em que o AR é utilizado é na área da publicidade. O consumidor com isso pode ter uma experiência de forma mais interativa e próxima com o produto. Mas apesar disso, realidade aumentada ainda é um recurso que vem sendo muito estudado nas mais diversas áreas no mundo todo. Seu uso pode ser diverso, e proporcionar muitas experiências novas. Estudos indicam que a tendência de seu uso se torne cada vez mais frequente nos próximos anos em áreas como televisão, engenharia e medicina. Além disso, com os avanços da própria computação, temos cada vez mais poder de processamento, e *hardwares* cada vez melhores chegando todos os anos. E hoje, o mercado vem cada vez mais apostando na tecnologia da realidade aumentada, uma vez que o recurso pode se tornar cada vez mais padrão, e ter alcance cada vez maior do grande público.

Um dos exemplos mais conhecidos que faz proveito deste recurso é o *game* Pokémon Go. Através da câmera, o usuário consegue visualizar o Pokémon imerso no ambiente real,

dando a impressão de que ele está de fato naquele local pronto para ser capturado. Outro exemplo de uso da realidade aumentada, é na visualização de locais de restaurantes, hotéis e mercados próximos ao usuário. Neste caso, o usuário ao andar na rua e projetar a câmera a sua frente, e a câmera exibe placas informando a posição dos locais (Conforme Figura 2.6). Como já mencionado anteriormente, a realidade aumentada neste trabalho será utilizada para projetar móveis. No capítulo 3, serão exibidos alguns exemplos de aplicações que usam a realidade aumentada como na proposta deste *app*. E no capítulo 4, na subseção 4.4, será explicado um pouco mais sobre a tecnologia utilizada para o AR neste trabalho (ARCore), e o que será possível realizar com ela no aplicativo.

Figura 2.5 - Exemplo de realidade aumentada no jogo Pokémon Go



Fonte: (Extreme Tech, 2017)

Figura 2.6 - Exemplo de Realidade Aumentada de estabelecimentos nas ruas



Fonte: (The Burn-in, 2019)

2.7 JSON

Acrônimo de *JavaScript Object Notation* e baseado em *JavaScript*, JSON é uma sintaxe para serializar informações no formato chave-valor objetos, *arrays*, números, strings, booleanos e *null*. JSON é bastante utilizado em *web services* pelo fácil entendimento e compreensão humana. No trabalho, o JSON é usado para processar os produtos vindos da API do Mercado Livre. Uma vez solicitado um produto da API, é possível manipular todos os dados deste determinado produto tranquilamente. Na Figura 2.7, temos um exemplo genérico da sintaxe e estrutura de um arquivo JSON. Escrito em roxo no lado mais esquerdo, temos o que podemos considerar de chave, e ao lado direito, temos o valor desta chave. Como podemos ver, é bastante simples para qualquer ser humano ler, interpretar, e fazer o que for necessário com estes dados.

Figura 2.7 - Exemplo genérico de um arquivo JSON

```
1  {
2    "browsers": {
3      "firefox": {
4        "name": "Firefox",
5        "pref_url": "about:config",
6        "releases": {
7          "1": {
8            "release_date": "2004-11-09",
9            "status": "retired",
10           "engine": "Gecko",
11           "engine_version": "1.7"
12         }
13       }
14     }
15   }
16 }
```

Fonte: (MDN web docs, 2020)

2.8 User Experience (UX)

Nesta seção vamos dar um pequeno foco em UX. Tendo em vista que o assunto é bastante amplo e profundo, será abordado aqui de maneira mais simples com foco maior apenas no mundo de aplicativos mobile, que é o proposto neste trabalho. Dando uma definição de maneira genérica e simples sobre o que se trata UX, basicamente tudo que enxergamos hoje em serviços, produtos e marcas, existe uma percepção que pode ser boa ou ruim sobre elas em uma série de fatores. Qualidade do produto, atendimento ao cliente, suporte, visual, e até mesmo o lado emocional são alguns elementos que todo cliente sente ou gostaria de sentir da melhor forma possível quando está para conhecer uma empresa ou uma marca. Quando não há uma boa primeira impressão, a tendência de que ele volte diminui, e dependendo da ocasião, causar até na perda do mesmo como um novo cliente.

O objetivo no fim das contas é passar a melhor imagem possível para o cliente desde o início da relação com a empresa, até o momento da compra e o consumo. UX é basicamente tratado como uma cultura adotada pela empresa, e não simplesmente um comportamento que podem mudar ou passar a adotar de uma hora pra outra. Extrair um sorriso, fazer se sentir o

mais confortável possível, e fazer que o cliente diga “uau” na sua experiência, é um dos principais focos quando falamos de UX. Assim que o cliente é conquistado, as chances de o mesmo voltar a comprar e indicar para outras pessoas, que indicarão para outras novas são altas. O que torna a empresa a se destacar pelos seus valores e pelos seus serviços.

Contando com esta ideia, quando levamos o termo UX para o mundo mobile, existem alguns pontos importantes que todo cliente considera quando entra em um aplicativo. A primeira coisa levada em conta é o design. Ter um visual agradável e intuitivo é de suma importância para dar uma boa primeira impressão ao cliente. Quando falamos em design, não consideramos apenas o visual, mas também como este design irá proporcionar em termos de usabilidade. Se o aplicativo ter um visual relativamente agradável, mas ser difícil de usar ou demorar pra carregar, existem grandes chances de o usuário desistir do aplicativo e procurar outro de proposta similar. É crucial de que toda a informação exibida seja de fácil entendimento, bem estruturado, e intuitivo. Isso vai desde as cores usadas em cada lugar, ao tamanho dos botões e fonte do texto. Aplicativos que demonstrem muitas informações na tela podem causar confusões sobre onde o usuário deve dimensionar seu foco primeiro (ver Figura 2.8). E além de dar um visual poluído, o usuário pode também optar por não aprender o sistema e cair fora, por mais que precisasse apenas de 30 segundos para isso. Por isto, é importante que o design seja simples, dando as principais funcionalidades da forma mais intuitiva possível, como menus de fácil navegação, por exemplo. Um outro quesito importante já citado são as cores. Ter as cores certas para o aplicativo pode proporcionar grande diferença na maneira como o usuário irá interagir. Obviamente, visuais muito coloridos também não são recomendados, uma vez que causa desconforto e perda de foco ao ver muitas cores ao mesmo tempo. Dependendo do que se trata a empresa e sua cultura, existem cores específicas que podem agregar valor e serem bastante recomendadas. No caso deste trabalho, por se tratar de um aplicativo focado em e-commerce, existem diversos estudos sobre a relação de como as cores impactam de maneira emocional sobre os consumidores na maneira que compram em uma loja online. Em um infográfico realizado pela *KissMetrics*, contendo diversas informações de estudos sobre e-commerce agrupadas, cerca de 93% dos consumidores se atentam ao visual para decidir se comprariam ou não em um determinado site ou aplicativo de compra. E 85% dos compradores colocam as cores do ambiente em que estão comprando como uma das primeiras razões para decidir se vai comprar ou não um produto em particular. Para entendermos um pouco melhor, confira a Figura 2.9 do estudo feito (em inglês) sobre o significado que algumas cores impactam em *e-commerces*. Todos os dados deste infográfico agrupado pela *KissMetrics* vêm de fontes oriundas dos seguintes sites e artigos: [DirJournal \(dirjournal.com\)](http://dirjournal.com), [Color Matters](#)

(colormatters.com), “The Effects of Store Environment On Shopping Behaviors: A Critical Review” por Shun Yin Lam, “The Profit of Color” por Color Marketing Group, “The Psychology of Color in Marketing” por June Campbell e “The Psychology of Web Performance”.

Figura 2.8 - Exemplo de um site com design não recomendado



Fonte: (Weblium, 2018)

Figura 2.9 - Relação das cores e seu significado no e-commerce



Fonte: (Ecommerce-Nation, 2016)

Podemos ver como 8 cores diferentes impactam emocionalmente na maneira como um usuário decide na compra. Dependendo do que a empresa deseja vender, o foco do público alvo muda. E por causa disto, certas cores podem ser mais recomendadas que outras para uma chance maior de sucesso. Traduzindo melhor o que temos escrito na legenda de cada cor da Figura 2.11, temos o seguinte:

- **Amarelo:** Otimismo e juvenil. Muito usado para pegar atenção da janela do comprador.
- **Vermelho:** Cor enérgica que acelera o batimento cardíaco. Cria uma “urgência” na necessidade de comprar.
- **Azul:** Gera sensação de confiança e segurança. Comumente visto em bancos e negócios.
- **Verde:** Associado a saúde. A cor mais tranquila para os olhos processarem. Usado em lojas para relaxar.
- **Laranja:** Cor agressiva. Cria uma chamada de ação para comprar, inscrever ou vender.
- **Rosa:** Romântico e feminino. Usado em produtos de beleza feminina tanto ao público feminino adulto quanto ao feminino jovem.
- **Preto:** Cor forte e lustrosa. Usada para o mercado de produtos luxuosos.
- **Roxo:** Cor suave para acalmar. Usado muito em produtos de beleza.
- **Cinza:** Apesar de não constar na imagem, o cinza também impacta como uma cor neutra, que simboliza inovação e conhecimento.

No Android, assim como no iOS, existem métricas e boas práticas para realização de uma boa interface. Essas diretrizes de qualidade podemos chamar de *Guidelines*. No Android, podemos encontrar facilmente essas informações no site oficial⁶. Com estas diretrizes, temos diversas dicas e notações de como dar uma boa experiência de usuário no desenvolvimento do aplicativo. Indo desde informações de como montar um bom design, a testes de responsividade que são de extrema importância que o aplicativo respeite independente do hardware do seu *smartphone*. Neste trabalho, apesar da consciência da existência das *Guidelines*, as interfaces foram construídas com base nas experiências do Autor deste trabalho na utilização de aplicativos que demonstram um bom design e conceitos em UX. Na subseção 4.5 do Capítulo 4, será abordado as escolhas usadas para o aplicativo e a razão dos mesmos.

⁶ <https://developer.android.com/docs> - Acesso Julho 2020

2.9 Tabela final das tecnologias utilizadas

Todas as tecnologias e recursos utilizados podem ser visualizados novamente neste resumo na Tabela 2.1. Como podemos ver, a linha representa cada um dos recursos, enquanto a coluna indica onde cada tecnologia é usada neste trabalho.

Tabela 2.1 – Resumo de cada tecnologia utilizada neste trabalho

Tecnologia / Recursos	Descrição de uso
MVC	Padrão de projeto usado no desenvolvimento do ARShop.
Java	Linguagem usada para desenvolver o aplicativo. A versão do Java no desenvolvimento deste app é a v8.0 .
Git	Usado para organização e registro de cada etapa no desenvolvimento do ARShop. Para este trabalho, a plataforma usada foi o GitHub.
Android / Android Studio	Sistema Operacional e a IDE oficial utilizada para o desenvolvimento do aplicativo usado para o desenvolvimento do aplicativo. A versão do Android Studio utilizada do início ao fim do desenvolvimento foi a v.3.5.3 (Dezembro 2019) .
API Mercado Livre	Usado pelo aplicativo como base de dados dos produtos do aplicativo.
Realidade Aumentada	Visualização dos produtos com a câmera. Uma vez que o usuário desejar ver mais detalhes de um produto, será dada a opção para visualizar o mesmo em Realidade Aumentada. A tecnologia usada para isto será o ARCore.
JSON	Dado retornado após alguma requisição na API. Sua informação é usada para geração de cada produto. Basicamente, quando visualizamos um produto em detalhe no aplicativo, as informações exibidas vêm do JSON retornado do servidor.
User Experience	Conceito é utilizado para definir todas as interfaces e usabilidade do aplicativo.

Fonte: O Autor

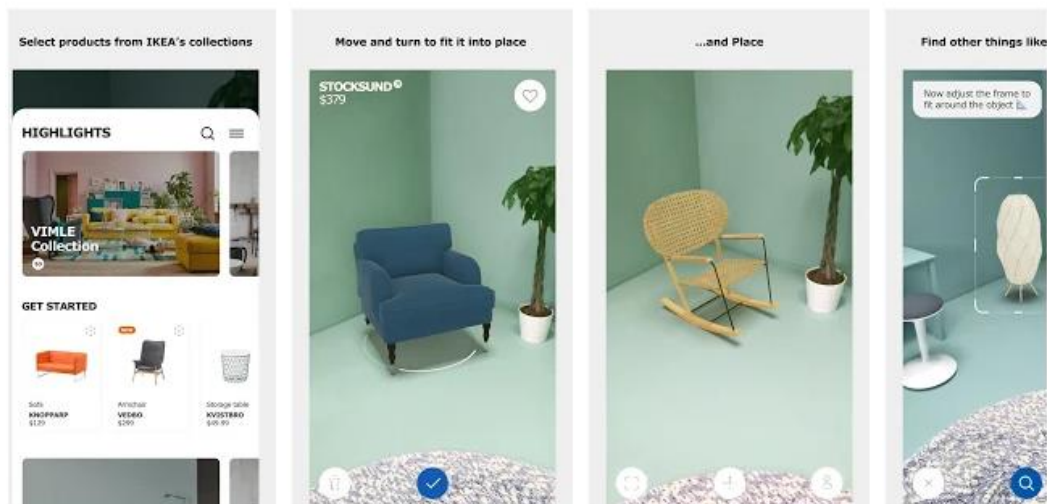
3 TRABALHOS RELACIONADOS

Este Capítulo tem como objetivo descrever alguns exemplos de aplicativos de *e-commerce* de móveis que fazem o proveito da realidade aumentada para vender seus produtos. Todos os aplicativos aqui listados foram pesquisados realizando uma busca na *internet*. A busca foi realizada em torno de março de 2020. Portanto, dependendo do momento em que a leitura deste documento for realizada, podem existir outras aplicações que realizam ou possuem objetivos similares ao deste trabalho.

3.1 IKEA Place

Seguindo descrição ao aplicativo na Google Play, este *app* coloca os produtos IKEA no seu espaço. É um aplicativo normalmente comentado entre os aplicativos que oferecem a realidade aumentada para móveis, e fornece uma interface simples. Em termos de vantagens, fornece uma boa fidelidade na modelagem 3D dos seus produtos, porém é restrito apenas aos produtos da marca. Além disto, este aplicativo não é disponível para o Brasil.

Figura 3.1 - Imagens ilustrativas do IKEA Place⁷



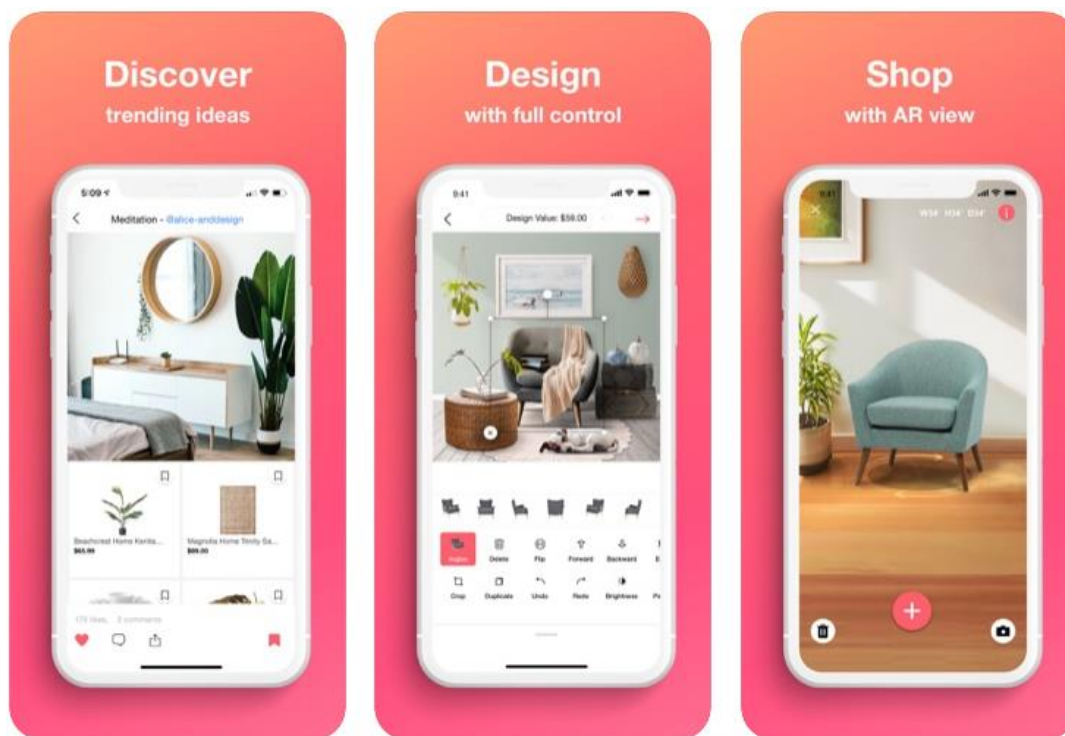
Fonte: (Google Play Store, 2020)

⁷ https://play.google.com/store/apps/details?id=com.inter_ikea.place&hl=en – Acesso em fevereiro de 2020

3.2 DecorMatter

Exclusivo para a plataforma iOS, este é um aplicativo com objetivo de explorar a interatividade e criatividade das pessoas em criar ambientes. Assim como o primeiro exemplo, também funciona como um e-commerce voltado para móveis, mas diferente do primeiro, possui bem mais opções de marcas e modelos disponíveis. Uma das vantagens desse app é justamente no fato de ser possível modelar ambientes da maneira que desejar com a disponibilidade de diversos modelos de produto. Entretanto, a desvantagem é que a maioria dos produtos disponíveis na criação de ambientes não são modelados em 3D, e sim em 2D. Apesar disso, ainda assim é possível construir bons modelos e ambientações com o aplicativo.

Figura 3.2 - Imagens ilustrativas do DecorMatter⁸



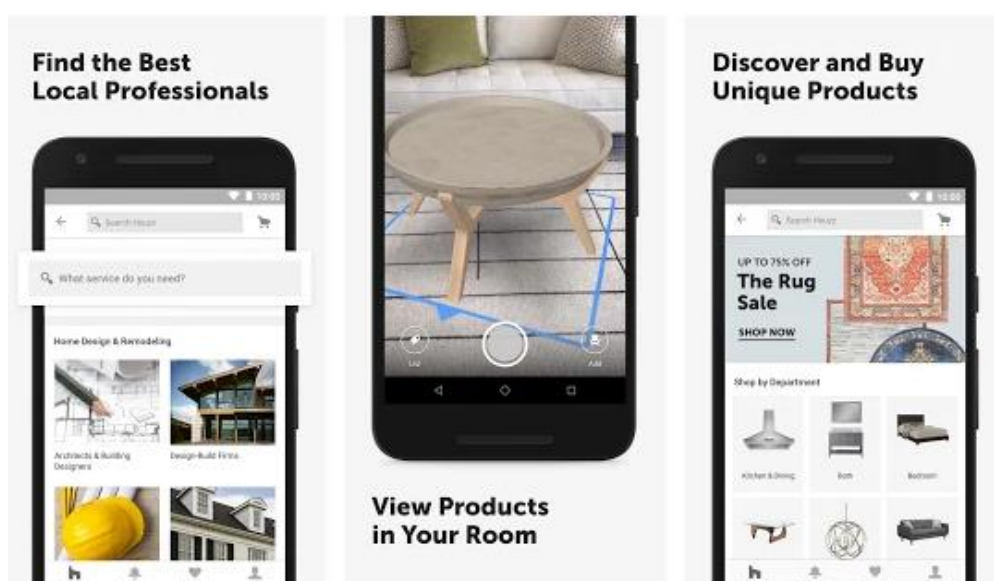
Fonte: (App Store, 2020)

⁸ https://apps.apple.com/us/app/decormatters-com/id1168709467?branch_match_id=760940098635846961 – Acesso em fevereiro de 2020

3.3 Houzz

Este é um dos aplicativos mais completos de e-commerce para móveis. Fornece uma grande quantidade e variedade de opções e marcas para compra. Também possui o recurso da realidade aumentada como um grande diferencial, tendo produtos com boa fidelidade e qualidade. Dentre os aplicativos, este é o mais parecido com o objetivo do aplicativo deste trabalho.

Figura 3.3 – Imagens ilustrativas Houzz⁹



Fonte: (Google Play Store, 2020)

3.4 Comparativo

Tabela 3.1 – Tabela comparativa entre os aplicativos

<i>Aplicativos</i>	<i>Realidade Aumentada</i>	<i>Modo Design</i>	<i>Expansível a outros produtos.</i>	<i>App Nacional</i>
IKEA Place	Sim	Não	Não*	Não
DecorMatter	Sim	Sim	Não*	Não
Houzz	Sim	Sim	Não*	Não
ARShop	Sim	Não	Sim	Sim

Fonte: O Autor

⁹ <https://play.google.com/store/apps/details?id=com.houzz.app&hl=en> – Acesso em fevereiro de 2020

Na Tabela 3.1, temos uma comparação entre os aplicativos do principal recurso considerado como diferencial, a realidade aumentada. Todos aplicativos mostrados são de empresas relativamente grandes e profissionais. A observação (*) na terceira coluna, será explicada mais adiante. Como podemos observar, a primeira coluna todos possuem a característica da realidade aumentada. O ponto está nas outras três colunas. O Modo Design é definido como um recurso do usuário poder a partir da sua câmera, montar todo um ambiente a gosto do usuário. O que significa que é possível colocar diversos móveis diferentes ao mesmo tempo no ambiente, e o usuário monta o espaço que deseja da maneira que desejar. Estes dois recursos foram observados de maneira mais clara nos aplicativos DecorMatter e Houzz. No IKEA pareceu um pouco mais similar ao ARShop, onde o objetivo é visualizar apenas um móvel desejado. Apesar do ARShop não possuir o recurso, nada impede de ser um recurso colocado futuramente como uma melhoria do aplicativo. Sendo este apresentado apenas uma versão inicial. O grande ponto que possa se distinguir de todos está na terceira coluna. Todos os 3 aplicativos, com base em consultas e no que pode se observar em seus objetivos, são focados em móveis. A observação (*) colocada nesta coluna está para denotar de que os mesmos possivelmente não possuem intenções ou objetivos de expansão para outros setores. E sim manter o foco no quesito que já são fortes, no caso móveis. O ARShop, teve como propósito inicial focar em móveis por ser uma área que gera muitas dúvidas quanto as suas dimensões e variedades de modelo. Entretanto, como o aplicativo utiliza uma base de dados do Mercado Livre, o aplicativo se torna totalmente expansível para praticamente qualquer área do mercado. Uma vez que o Mercado Livre vende de praticamente tudo, o recurso da Realidade Aumentada basicamente anda junto. O aplicativo foi pensado muito para o futuro, onde o uso do AR se poderia fazer proveito em praticamente qualquer produto.

Um ponto interessante que podemos observar é que todos aplicativos usados em comparação são de fora do país. O que se pressupõe que caso deseje comprar em alguma dessas lojas, teríamos o entrave da moeda. No Brasil, fazendo algumas pesquisas, foi observado ser praticamente nulo ainda o uso de AR para o lado e-commerce. Maioria (ou todas) das grandes lojas de varejo, seja físico ou *web*, não utiliza o recurso. Suas razões podem ser diversas, mas um dos mais prováveis motivos que pode ser considerado, é o fato de o público Brasileiro ainda não estar preparado para recebe-lo. De nada adianta investir muito dinheiro hoje, supondo que apenas 10% ou menos do público irá de fato utilizar. Apesar disso, imagina-se que o país não esteja muito longe de mudar este cenário. A tecnologia está em constante melhoria, e num futuro próximo certamente a tecnologia chegará ao grande público naturalmente por um preço

mais acessível. ARShop seria um dos poucos aplicativos nacionais hoje que utiliza um recurso que certamente entrará em tendência crescente, e considerando que estamos usando uma base de dados que é praticamente acessível a todos (Mercado Livre), poderia se preparar muito melhor quando a novidade chegar com tudo.

4 METODOLOGIA DE DESENVOLVIMENTO

Este capítulo tem como objetivo mostrar, em detalhes, todas as etapas de desenvolvimento do ARShop. São abordados a visão geral, arquitetura, explicação dos recursos utilizados, conceito de *user experience*, e por fim, as interfaces.

4.1 Visão Geral

Considerando que a ideia é implementar uma loja online utilizando a API do Mercado Livre, uma série de requisitos básicos são necessários para que isso aconteça. Primeiramente, independente de API, é preciso entender o que um usuário gostaria de ver quando entra em um aplicativo de compra. Não é difícil imaginar algumas ações básicas como: ver produtos, ver as categorias de produtos, ver os detalhes dos produtos, ver fotos, usuário favoritar um produto, ver o carrinho das compras, e obviamente, comprar um produto.

Uma vez levantado os requisitos básicos, é possível iniciarmos uma estrutura básica do aplicativo. Nesta etapa, o Android Studio entra em ação, onde a implementação dos requisitos começa a entrar em análise e estudo. Inicialmente, houve um foco na lógica básica da aplicação. Criando classes que dariam o conceito do que seriam, por exemplo, um Produto, um Usuário, ou uma Categoria de Produtos. Uma vez que temos os “objetos” básicos implementados, entramos numa segunda etapa de desenvolvimento. Nesta etapa, entra o conceito de Activities. Basicamente, Activity são classes onde estarão as funcionalidades do aplicativo, juntamente com sua interface. Aqui entra as questões de como o usuário irá interagir com as informações que aparecem na tela. Apesar de comentado o conceito de interfaces já nesta segunda etapa, o foco neste momento foi desenvolver as funcionalidades de cada elemento. Com isso, as telas neste ponto são bastante simples tendo apenas as informações “jogadas” na tela, organizando como isso seria apresentado ao usuário posteriormente.

Por fim, uma vez implementados toda a estrutura básica e as funcionalidades do aplicativo, o desenvolvimento se dividiu em mais duas etapas. A primeira focando total no design de cada interface, ou seja, relação das cores, onde cada elemento de texto deve aparecer e como deve aparecer, fonte do texto, tamanho da fonte, etc. Na segunda, focando na realidade aumentada dos produtos, e o lado mais “*cliente-side*”, que envolvia *login* e informações da conta de usuário.

Neste trabalho, vários recursos foram adicionados à medida que a aplicação tomava forma. No final, os conceitos utilizados para o desenvolvimento deste aplicativo foram: *Activities*, *Fragments*, *CardViews*, *BottomNavigationView*, *ARCore*, *Recyclers*, *PagerAdapter*, *Dialogs* e *StateProgressBar*. Para o uso da API, dois recursos extras foram utilizados: *Volley* e *Glide*. O primeiro serve para adicionar uma requisição para o servidor na fila. E o segundo serve para carregar de maneira mais adaptativa e dinâmica as imagens dos produtos. Cada um destes termos será abordado na subseção 4.4.

4.2 Requisitos Funcionais e Não Funcionais

Dado a visão geral do ARShop na subseção anterior, existem alguns requisitos funcionais e não funcionais que foram definidos sobre o que seria importante constar no aplicativo. Segue eles:

▪ Requisitos Funcionais

- Visualização de categorias dos produtos.
- Visualização da lista de produtos a partir da categoria escolhida.
- Visualização detalhada de informações de cada produto.
- Visualização em Realidade Aumentada do produto (se disponível).
- Sistema de Login e cadastro de nova conta para o cliente utilizar o aplicativo.
- Adição, Remoção e Visualização de cada produto que o cliente colocar no seu carrinho.
- Adição, Remoção e Visualização de cada produtos que o cliente colocar na sua lista de produtos favoritos.
- Edição e Visualização dos dados pessoas do cliente.
- Edição e Visualização dos Endereços de entrega de um cliente.
- Adição, Edição e Visualização da lista de Cartões de Crédito de um cliente.
- Visualização do seu histórico de compras.

▪ Requisitos não funcionais

- Interface simples e de fácil usabilidade
- Botões devem indicar claramente que são clicáveis e não serem muito pequenos para dificultar o usuário a clicar.

- Ao verificar um produto em AR, o usuário poderá mover, escalar e girar o produto na tela através do *touchscreen*.
- O produto ao ser inserido na tela em AR já deve estar na escala adequada (tamanho real do produto).
- Transição suave para esquerda / direita entre as telas
- Menu de navegação com as principais funcionalidades a disposição na tela sempre do usuário.

4.3 Arquitetura

Como já mencionado, este trabalho foi desenvolvido usando o padrão MVC (*Model-View-Controller*) com o Android Studio. No projeto, existe uma camada extra chamada “*support*” onde estão inseridos tudo que fornece melhorias com relação a interface e no uso otimizado do hardware que rodará a aplicação. Não foi utilizado nenhum banco de dados, apesar de ser totalmente aplicável num momento futuro. A razão é devido ao foco dada no recurso da realidade aumentada e na interface e funcionalidade de cada tela. Para melhor organização e entendimento, estão listadas mais abaixo todas as classes de cada camada contendo uma breve explicação de cada uma delas. Neste trabalho, é importante notar que tirando a camada *Model*, todas as outras estão nomeadas com um “Prefixo + Nome_da_tela”. Este prefixo indica o que a classe representará no aplicativo. Portanto, utilizando o exemplo da classe “*ActivityCreateUser*” da camada *Controller*, o prefixo indica que ela é uma *Activity* (conceito explicado na subseção 4.4), e o nome da tela é “*CreateUser*”. Ou seja, ela é uma *Activity* que representa a tela de criação de um novo usuário.

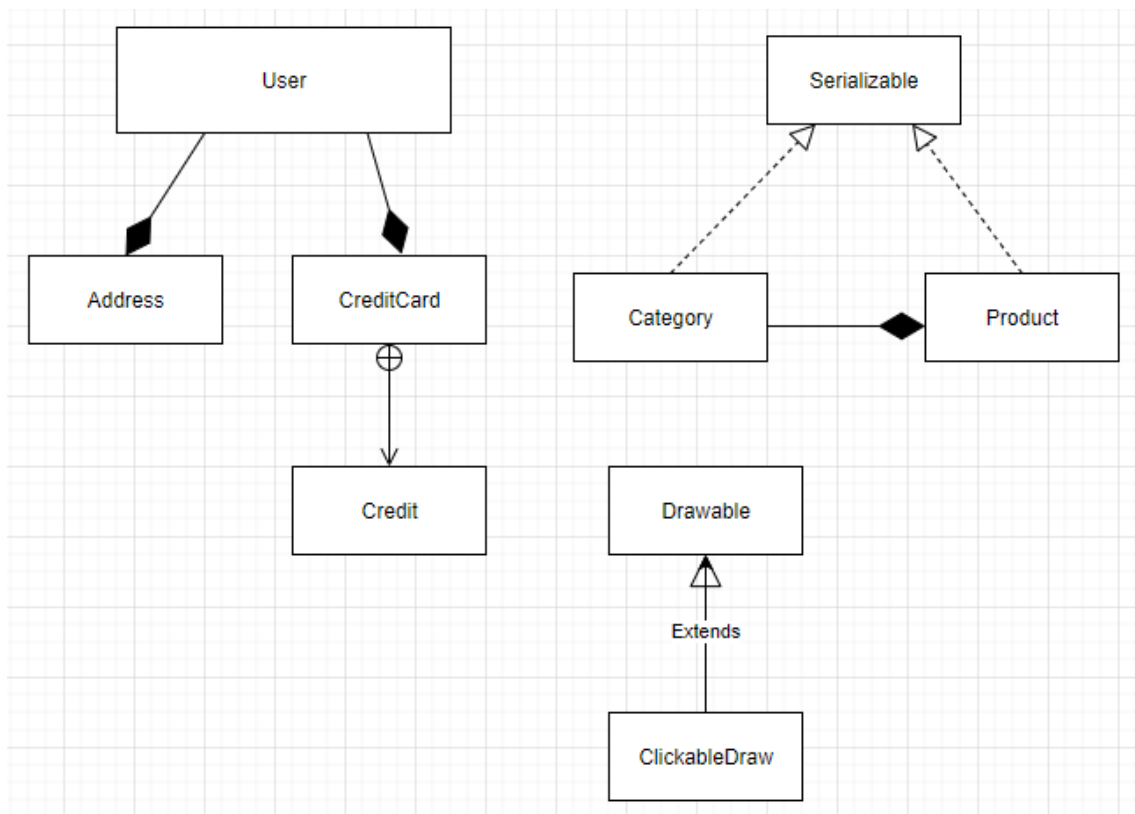
Model:

- *Address*: Classe responsável por ter toda a lógica relacionada ao Endereço de um usuário. Um usuário pode ter mais de um endereço caso desejar.
- *Category*: Classe responsável por ter toda a lógica relacionada a categorias dos produtos. Cada categoria pode ter um nome e uma lista de produtos associados. Supondo a criação de um objeto do tipo *Category* chamada carros, neste objeto podemos ter uma lista com diversos tipos de modelos de carros associado a ela.
- *ClickableDraw*: Classe responsável por lidar com uma parte da AR. Como o nome informa, cuida da parte de desenhar um objeto ao clicar na tela.

- *CreditCard*: Classe responsável por lidar com a lógica relacionada ao cartão de crédito de um usuário. Tendo a mesma ideia do endereço, um usuário pode ter mais de um cartão cadastrado para realizar a compra.
- *Product*: Classe responsável pela lógica relacionada aos produtos no geral. Todos os dados referentes a nome, preço, marca, entre outros, estão aplicados nesta classe.
- *User*: Classe responsável pela lógica relacionada ao usuário. Informações de conta (email e senha), informações pessoais (Nome, idade, CPF, ...), lista de endereços e cartão de créditos estão todos associados na criação de um objeto nesta classe.

Podemos ver na Figura 4.1 abaixo de maneira bastante simplificada, um diagrama visualizando a relação das principais classes. Um usuário possui relação de composição com as classes *Address* e *CreditCard*, enquanto a classe *Category* com *Product*. As classes *Serializable* e *Drawable* são classes já implementadas para serem utilizadas. A primeira para lidar com passagem de informações entre classes (as duas implementam a classe *Serializable*), e a segunda para lidar com desenhos (relação de herança).

Figura 4.1 – Diagrama de Classes simplificado das principais classes.



Fonte: O Autor

Controller:

Activities:

- *ActivityCreateUser*: Responsável por cadastrar de um novo usuário para a aplicação. Caso o usuário não tenha uma conta associada, está é a classe responsável pela criação do mesmo.
- *ActivityFinishPurchaseSection*: Representa a passo final de um processo de compra de produtos de um usuário. Exibe as informações de tudo que está sendo comprado, o subtotal, e sua forma de pagamento para que o usuário confirme de que as informações estão corretas.
- *ActivityLogin*: Classe que contém todas as propriedades para logar qualquer usuário para dentro do aplicativo.
- *ActivityModelDisplay*: Responsável por ser o recurso diferencial da aplicação. Aqui é onde o AR dos produtos entra em ação.
- *ActivityMyCart*: Responsável por exibir o atual estado do carrinho de um usuário. Cada produto que for adicionado ao carrinho pelo usuário, será exibido por esta classe.
- *ActivityMyFavorite*: Responsável por exibir a lista dos produtos favoritos de um usuário.
- *ActivityMySettingOption*: Responsável por exibir a tela específica de configuração desejado pelo usuário.
- *ActivityMySetting*: Classe mais genérica para definir todas as opções de configuração disponíveis pelo usuário.
- *ActivityPaymentSection*: Representa a segunda etapa do processo de compra de um usuário. Todas as informações referentes ao pagamento serão exibidos nesta classe.
- *ActivityProductCategory*: Esta é a classe que representa a tela inicial do aplicativo. Responsável por exibir todas as categorias de produtos disponíveis para compra. A partir desta tela, é possível ir para quase qualquer outra tela do aplicativo.
- *ActivityProductDetail*: Responsável por exibir as informações detalhadas de um produto. Quando o usuário deseja saber mais informações de um produto, esta é a classe chamada para realizar esta função.
- *ActivityProductList*: Responsável por listar todos os produtos dentro de uma determinada categoria. Uma vez escolhida a categoria de produtos desejada pelo

usuário, esta classe será chamada para exibir os produtos correspondentes a esta categoria.

- *ActivityProductPurchase*: Representa a primeira etapa do processo de compra de um usuário. Todas as informações básicas relacionadas ao usuário, frete, e meios de pagamento são definidos aqui.
- *LoggedUser*: Classe responsável por ter todas as informações relacionadas ao cliente recém logado. Dados como quem é o usuário logado, carrinho de compras do usuário, compras realizadas, e produtos colocados como favoritos, serão todos controlados por esta classe.
- *ModelLoader*: Responsável por lidar com toda a parte de modelagem do objeto 3D. Aqui em específico envolve a realidade aumentada dos objetos.

Fragments:

- *FragmentMyAddress*: Responsável por exibir as informações de endereço de um usuário. Esta é uma das telas de opções de configuração do usuário acessado nas configurações de perfil.
- *FragmentMyCreditCards*: Responsável por exibir todos os cartões de crédito cadastrado pelo usuário. Esta é uma das telas de opções de configuração do usuário acessado nas configurações de perfil.
- *FragmentMyProfile*: Responsável por exibir todas as informações referentes ao usuário. Esta é uma das telas de opções de configuração do usuário acessado nas configurações de perfil.
- *FragmentMyPurchases*: Responsável por exibir o histórico de compra do usuário. Esta é uma das telas de opções de configuração do usuário acessado nas configurações de perfil.

View:

- *activity_create_user*: Tela de criação de um novo usuário.
- *activity_finish_purchase_section*: Tela de finalização de um processo de compra de um usuário.
- *activity_login*: Tela de login do aplicativo.
- *activity_model_display*: Tela onde exibe os modelos dos produtos em AR.
- *activity_my_cart*: Tela de exibição de carrinho do usuário.

- *activity_my_favorite*: Tela de exibição da lista de produtos favoritos do usuário.
- *activity_my_setting_option*: Tela que exibe a opção selecionada pelo usuário nas configurações de perfil.
- *activity_my_settings*: Tela que exibe as configurações de perfil gerais do usuário.
- *activity_payment_section*: Tela que exibe o processo de pagamento do usuário das compras dos produtos realizados.
- *activity_product_category*: Tela que exibe as categorias de produtos existentes.
- *activity_product_detail*: Tela que exibe todas as informações de um produto em detalhes.
- *activity_product_list*: Tela que exibe todos os produtos de acordo com a categoria selecionada pelo usuário.
- *activity_product_purchase*: Tela que exibe as informações de endereço, frete e método de pagamento ao usuário. Mostrado quando o usuário opta por realizar a compra de algum produto.
- *card_item_category_view*: Cartão genérico para representar uma opção de categoria de um produto.
- *card_item_credit_card_view*: Cartão genérico para representar as informações dos cartões de crédito de um usuário.
- *card_item_product_view*: Cartão genérico para representar como um produto será exibido na lista de produtos.
- *card_item_view*: Cartão genérico para ser utilizado onde necessário. Neste caso está sendo usado para mostrar itens no carrinho e itens nos favoritos.
- *dialog_add_new_address*: Janela de diálogo para cadastrar um novo endereço a um usuário.
- *dialog_add_new_credit_card*: Diálogo para cadastrar um novo cartão de crédito a um usuário.
- *fragment_my_address*: Fragmento responsável por exibir na tela a lista endereços de um usuário.
- *fragment_my_credit_cards*: Fragmento responsável por exibir na tela a lista de cartões de crédito de um usuário.
- *fragment_my_profile*: Fragmento responsável por exibir na tela as informações de perfil de um usuário.

- *Fragmente_my_purchases*: Fragmento responsável por exibir na tela o histórico de compras de um usuário.

Support:

Adapters:

- *ImageSliderView*: Classe responsável por lidar com as imagens dos produtos. Uma vez desejado pelo usuário ver um produto em detalhes, existe uma lista de imagens que o produto fornece para ser visualizados. Esta classe lida com esta lista de imagens para ser possível visualizar de maneira mais fácil e interativa.

Dialogs:

- *DialogAddressAdder*: Classe responsável por exibir na tela o diálogo para cadastrar um novo endereço na lista de endereços do usuário.
- *DialogCreditCardAdder*: Classe responsável por exibir na tela o diálogo para cadastrar um novo cartão de crédito na lista de cartões do usuário.

Recyclers:

- *RecyclerCategoryView*: Classe responsável por exibir cada item de categorias disponíveis no aplicativo.
- *RecyclerMyCartView*: Classe responsável por exibir cada item no carrinho do usuário.
- *RecyclerMyCreditCardsView*: Classe responsável por exibir cada cartão de crédito cadastrado pelo usuário.
- *RecyclerMyFavoriteView*: Classe responsável por exibir cada produto vinculado a lista de favoritos do usuário.
- *RecyclerMyPurchaseView*: Classe responsável por exibir o histórico de compras do usuário.
- *RecyclerMySettingView*: Classe responsável por exibir cada item do menu de configurações de perfil.
- *RecyclerProductCardView*: Classe responsável por exibir cada produto na lista de produtos de uma determinada categoria.

4.3.1 Comunicação com a API do Mercado Livre

Para usar a API o processo é bastante simples. Em sua documentação, qualquer informação desejada baseia-se numa URL base (*Uniform Resource Locator*) + ID (*Identifier*). Esta ID pode se referir a uma categoria, a um cliente, a uma publicação de produtos, a uma busca de itens, entre outros. Na Figura 4.2, podemos ver alguns exemplos mostrados na documentação do site.

Figura 4.2 - Trecho da página da documentação da API

<code>/users/{Cust_id}/accepted_payment_methods</code>	Obtém os métodos de pagamento aceitos pelo vendedor para cobrar.	GET
<code>/applications/{application_id}?ACCESS_TOKEN</code>	Obtém dados sobre o aplicativo.	GET
<code>/users/{User_id}/brands</code>	Este processo recupera marcas associadas a um <code>user_id</code> . O atributo <code>oficial_store</code> identifica uma loja.	GET

Fonte: (Developers Mercado Livre, 2020)

Como o objetivo é obtermos os produtos, uma vez que temos a URL base, a única coisa que precisamos são os IDs deles. Na Figura 4.3, podemos ver uma linha de código mostrando esta URL. Note que a única coisa que falta e que será preenchida é o ID, representado pela variável *productId*.

Figura 4.3 - Link de um produto qualquer. Definido através do seu *productId*

```
String url = "https://api.mercadolibre.com/items/"+productId;
```

Fonte: O Autor

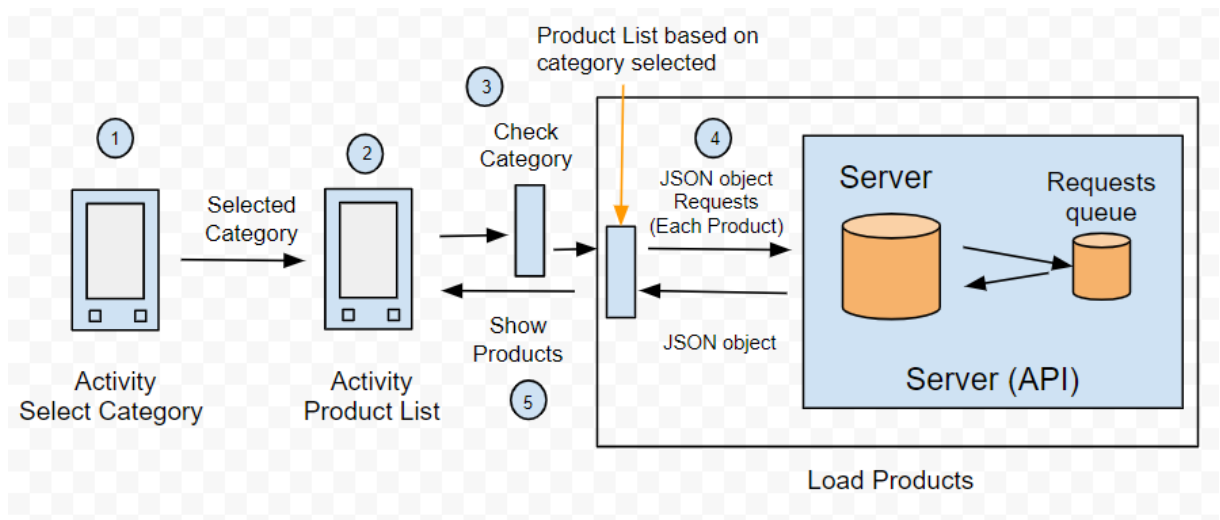
Figura 4.4 – Trecho de informação de um produto do Mercado Livre no formato JSON.

```
{
  "id": "MLB1197287312",
  "site_id": "MLB",
  "title": "Mesa Em L Diretor Dinamico 40mm Prius 1600x800x1800x450x745 ",
  "subtitle": null,
  "seller_id": "416203258",
  "category_id": "MLB193946",
  "official_store_id": null,
  "price": 1259.99,
  "base_price": 1259.99,
  "original_price": null,
  "currency_id": "BRL",
  "initial_quantity": 999,
  "available_quantity": 500,
  "sold_quantity": 2,
  "sale_terms": [...],
  "buying_mode": "buy_it_now",
  "listing_type_id": "gold_special",
  "start_time": "2019-03-21T13:34:13.000Z",
  "stop_time": "2039-03-16T04:00:00.000Z",
  "condition": "new",
  "permalink": "https://produto.mercadolivre.com.br/MLB-1197287312-mesa-em-l-diretor-dinamico-40mm-prius-1600x800x1800x450x745-3M",
  "thumbnail": "http://mlb-s1-p.mlstatic.com/845586-MLB31360410768_072019-I.jpg",
  "secure_thumbnail": "https://mlb-s1-p.mlstatic.com/845586-MLB31360410768_072019-I.jpg",
  "pictures": [...],
  "video_id": null,
}
```

Fonte: (API Mercado Livre, 2020)

Considerando que a API devolve as informações de um produto no formato JSON, como ilustrado em parte na Figura 4.4 acima, temos de enviar requisições através da classe `JsonObjectRequest`, enviando esta URL por parâmetro. Para melhor compreensão, segue a Figura 4.5 abaixo:

Figura 4.5 - Fluxograma de comunicação com a API do Mercado Livre



Fonte: O Autor

Explicando a Figura 4.5, vamos seguir as numerações marcadas na imagem. O aplicativo inicia na tela de categorias (1), onde o usuário pode clicar em qual categoria de produtos gostaria de visualizar. Neste momento, a aplicação possui 5 categorias: Cadeiras, Sofás, Mesas, Eletrodomésticos e Decorações. Clicando em qualquer uma delas, o usuário é movido para a segunda tela que é a exibição da lista de produtos (2). Nesta tela, é realizada uma verificação da categoria selecionada pelo usuário (3). Uma vez verificado, uma lista de *Strings*, baseada na categoria, é carregada. Essa lista são basicamente as *IDs* dos produtos que substituirão a variável “*productId*” conforme exibida na Figura 4.2. A partir desta etapa, as requisições ao servidor iniciam (4). Para cada Id da lista, uma nova requisição do tipo *JsonObjectRequest* é realizada e colocada na fila de requisições, uma vez que não tem como saber em quanto tempo o servidor vai realizar cada uma delas para estar preparado para processar uma nova. Sempre que o servidor finaliza, ele retorna a informação desejada no formato JSON. Neste momento, é realizado o processamento desse JSON extraindo tudo que for considerado importante referente ao produto (nome, preço, fotos, marca, descrição, etc). Tendo estes dados, um objeto do tipo “*Produto*” é criado passando por parâmetro todos esses dados adquiridos. Feito isto, ele é adicionado a uma lista de produtos, que será a lista exibida na tela do usuário após todos os produtos serem processados (5).

4.4 Recursos de desenvolvimento utilizados

Nesta seção é fornecida uma descrição, e onde é usado cada recurso utilizado no desenvolvimento deste aplicativo.

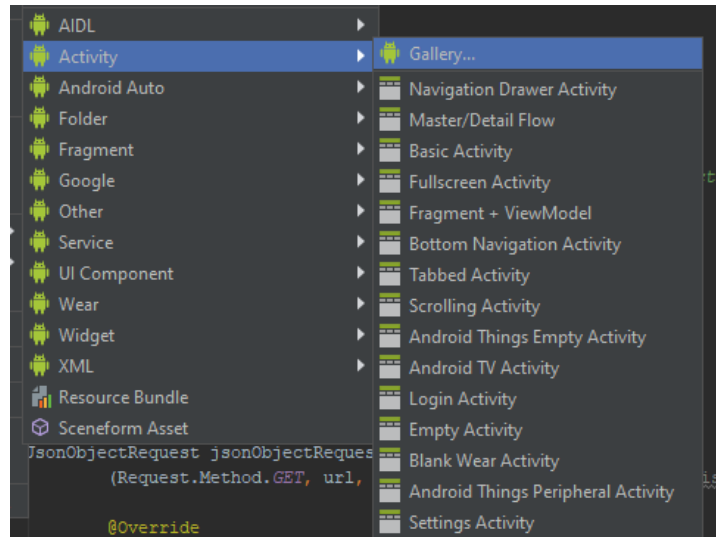
- *Activity*

Uma *Activity*¹⁰ é o elemento mais básico utilizado no desenvolvimento para *Android*. Basicamente é a janela onde o aplicativo mostra sua interface. Maioria dos aplicativos é composto por várias interfaces diferente, os quais navegamos para uma tela a partir de outra. Cada uma destas telas representam uma *Activity*. No *Android Studio*, existem diversas *activities* com elementos bases que podemos usar na hora de criar uma interface. Isso nos permite já termos certos elementos na tela previamente implementados, auxiliando o desenvolvedor a não precisar fazer este trabalho. Estes

¹⁰ <https://developer.android.com/guide/components/activities/intro-activities> - Acesso em janeiro de 2020

elementos envolvem a barra de menu superior, menu lateral, com um botão, navegação inferior, entre outros. No aplicativo deste trabalho todas as *activities* utilizadas foram feitas do zero (Na figura abaixo corresponde ao “*Empty Activity*”).

Figura 4.6 - Tipos de *Activities* que o Android Studio oferece



Fonte: O Autor

- *Fragments*

*Fragments*¹¹ pode ser definido como uma representação de uma parte da interface de uma *Activity*. Apesar de termos uma ideia similar ao de uma *Activity*, as *Fragments* são totalmente dependentes de uma *Activity* existir, enquanto o contrário não acontece. O diferencial dele é o reuso, onde poderemos usar vários fragmentos em uma única *Activity*. Outro ponto de um *Fragment* é o ciclo de vida. Quando uma *Activity* é pausada os *Fragments* também são pausados, e quando destruídos, eles vão juntos. Uma das grandes vantagens do uso de *Fragments* é o fato de ser mais leve, por proporcionar justamente o reuso de um mesmo espaço de tela para reproduzir outras interfaces, tendo também o uso de memória reduzido. No ARShop, o uso das *fragments* foram voltadas nas telas de “Perfil”. Como o objetivo é voltado basicamente para mudanças em algum dado de perfil, as telas poderiam ser reusadas apenas modificando a informação sobre o que o usuário gostaria de modificar.

¹¹ <https://developer.android.com/guide/components/fragments?hl=pt-br> – Acesso em janeiro de 2020

- *CardView*

Como o nome informa, *CardViews*¹² são uma forma de mostrar as informações na tela do usuário dentro de um “cartão”. Normalmente possuem uma elevação dentro da interface, podendo criar sombras embaixo. *Cardview* é um ótimo elemento para termos uma visualização mais bonita e ao mesmo tempo fornecer estilo a interface. No ARShop, as *CardViews* foram utilizadas em diversas telas. Algumas telas que podemos observar estas características são nas *Activities* “*my_cart*”, “*my_favorite*” e “*product_list*” Podemos ver um exemplo de cardview utilizado neste trabalho para exibir um item genérico conforme a Figura 4.7 abaixo. A imagem foi retirada da parte de edição da tela dentro da IDE Android Studio.

Figura 4.7 - Exemplo de CardView



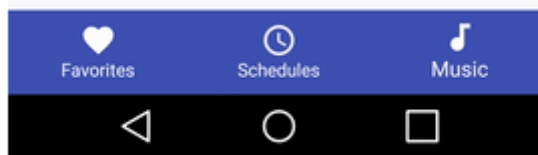
Fonte: O Autor

¹² <https://developer.android.com/guide/topics/ui/layout/cardview> - Acesso em janeiro de 2020

- *BottomNavigationView*

*BottomNavigationView*¹³ é basicamente o menu de navegação rápida localizada na parte de baixo da tela. Por ser embaixo, é mais fácil para o usuário navegar com um único clique para outras telas do aplicativo pela forma em como as pessoas normalmente seguram o smartphone. Ela é normalmente composta por itens que certamente serão as telas mais acessadas pelos usuários em geral. Para usar este tipo de navegação, é recomendado que existam pelo menos 3 e no máximo 5 opções de destinos de navegação. No ARShop, o menu de navegação rápida está disponível logo que o usuário entra em sua conta. Contará com os botões “Home”, “Meu Carrinho”, “Favoritos” e “Perfil”.

Figura 4.8 - Exemplo de BottomNavigationView



Fonte: (Medium, 2016)

- *ARCore*

Ferramenta utilizada para a Realidade Aumentada no Android. Requerido Android 7.0 ou mais recente para usufruir, foi desenvolvida para rodar numa variedade imensas de dispositivos *Android*. O *ARCore*¹⁴ funciona de duas formas: tracejar a posição do dispositivo de acordo com o movimento, e construir seu entendimento do mundo real. Usando a câmera do smartphone, o sensor tem a capacidade de identificar superfícies planas, além da iluminação ao seu redor. A partir disto, identificado sobre como funciona ambiente ao seu redor, é possível usar o *ARCore* para colocar objetos, anotações ou outras informações no mundo real. Além disto, por ter o entendimento de maneira complexa de como funciona o ambiente, o usuário pode se mover em volta do objeto inserido e visualizar em qualquer ângulo. Para reproduzir qualquer objeto, é necessário que o objeto seja modelado através de programas de modelagem 3D. E a partir disto, arquivos de formato específico (.obj, .mtl, ...) que contém todas as

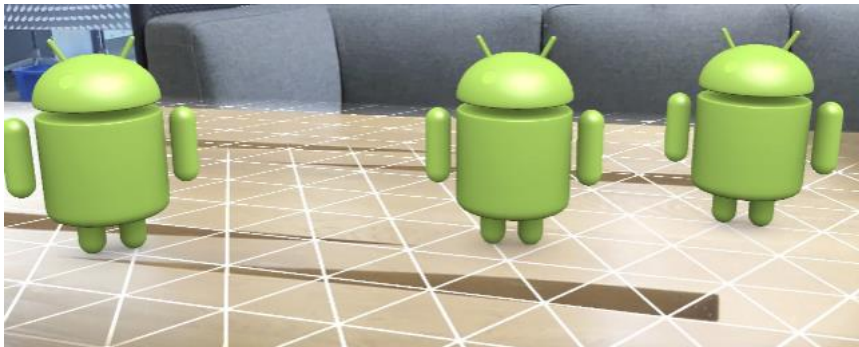
¹³ <https://developer.android.com/reference/android/support/design/widget/BottomNavigationView> - Acesso em março de 2020

¹⁴ <https://developers.google.com/ar/discover> - Acesso em fevereiro de 2020

informações do modelo (pontos, vértices e texturas) podem ser gerados e usados para ser interpretado pelo Android Studio. No aplicativo, basicamente é usado para visualizar os móveis em AR.

Conforme a Figura 4.9 abaixo, uma vez que temos o plano detectado e o objeto colocado na tela, é possível o usuário interagir com ele. No ARShop, por exemplo, é possível arrastar na tela o móvel em AR. O usuário poderá com isto colocar o móvel onde desejar. Além disto, caso o objeto não esteja no ângulo adequado, o usuário poderá girar o objeto realizando o movimento de girar com o duplo toque na tela. E caso ainda necessário, é possível realizar a escala do objeto. Para esta ação, é necessário realizar também o duplo toque aproximando ou afastando os dedos na tela (ação similar ao *zoom in / out* da câmera). Como a proposta deste trabalho é apresentar o objeto em tamanho real, o objetivo é mostrar todos os móveis já na escala adequada, representando o tamanho verdadeiro do móvel para o cliente.

Figura 4.9 - Exemplo genérico do ARCore exibindo o símbolo do Android



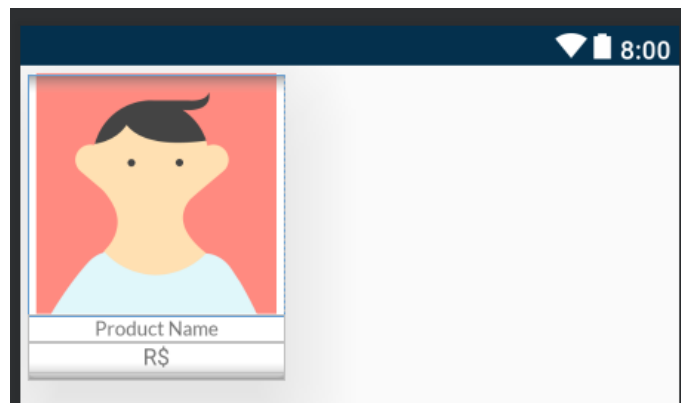
Fonte: (Developers Google, 2020)

- *RecyclerView*

Uma versão mais elaborada e avançada do recurso *ListView*, os *recyclers*¹⁵ é basicamente uma ferramenta que permite reutilizar um elemento já criado e repetir dentro da interface quantas vezes quiser e da maneira que desejar mostrar. Para entender um pouco melhor na prática o que significa isto, segue a imagem abaixo um exemplo utilizado neste trabalho:

¹⁵ <https://developer.android.com/guide/topics/ui/layout/recyclerview> - Acesso em janeiro de 2020

Figura 4.10 - Exemplo de aplicação do RecyclerView



Fonte: O Autor

Como podemos ver, a imagem mostra um trecho da tela com um objeto do tipo `CardView` que representa um produto genérico qualquer. Neste objeto, temos uma imagem base (representado pela pessoa na imagem) que representa o produto, e embaixo temos o dado de nome e preço. Porém, caso desejarmos colocar um segundo ou vários objetos iguais ao ilustrado, seria infundado replicarmos de maneira bruta desenhando o mesmo objeto na tela. Se desejarmos 100 elementos iguais, teríamos de replicar 100 vezes o mesmo elemento, o que não seria possível. É justamente neste ponto que entra os *Recyclers*. O nome em si é um pouco intuitivo por justamente ter a intenção de “reciclar” um mesmo objeto quantas vezes necessário. Isto é dado com objetos que é chamado de fixadores de visualização. Cada um é responsável por exibir um item de maneira independente. O que significa que caso tenhamos uma lista de produtos, cada objeto exibirá um produto diferente na tela. O *RecyclerView* criará a quantidade de fixadores necessários, dando ainda a dinamicidade à medida que o usuário rola a tela, a ferramenta retira os objetos não visualizados da tela e os reutiliza com novos dados. Porém, isto não acontece de maneira imediata, pois pode acontecer do usuário desejar voltar a ver o objeto que havia antes saído. Por isso, normalmente isto é feito com os objetos que estão a mais tempo fora da tela do usuário. Os *recyclers* no ARShop é utilizado em tudo que envolve listar uma série de itens. Portanto as telas onde lista as categorias, os produtos, o carrinho do usuário, e seus favoritos, por exemplo, todos utilizam algum *recycler* específico para seu propósito.

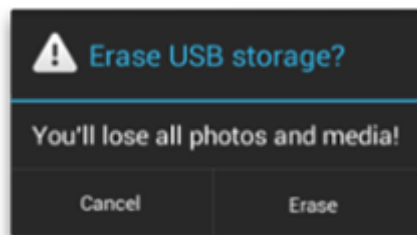
- *PagerAdapter*

Basicamente, *PagerAdapter*¹⁶ é um adaptador utilizado para lidar de uma maneira inteligente a visualização de vários elementos na mesma tela. Para este aplicativo, a função do *PagerAdapter* é na visualização das imagens dos produtos. Usando este recurso, ele permite que o usuário possa aproveitar o mesmo espaço de tela, para colocar várias imagens de um produto. A partir disto, o usuário com o arrastar do dedo para esquerda ou direita sobre a imagem, consegue ver a galeria de imagens de um determinado produto em sequência (slides).

- *Dialogs*

Toda pequena janela onde é dada uma informação que solicite uma confirmação ou decisão do usuário sobre ela é o que chamamos de *Dialogs*¹⁷. Normalmente não ocupam toda tela e são usados como modais. Os *Dialogs* aqui neste caso, estão sendo utilizados para cadastrar novos endereços de entrega e cartões de crédito do usuário. Abaixo podemos ver o exemplo de uma *Dialog*.

Figura 4.11 - Exemplo de Dialog



Fonte: (Developer Android, 2019)

- *Volley*

*Volley*¹⁸ é uma biblioteca HTTP que possui o intuito de facilitar a criação e a conexão com a rede nos aplicativos para Android. Oferece alguns benefícios como várias conexões de rede simultâneas, programação automática de solicitações na rede, além de ferramentas para depuração. Se integra tranquilamente com a qualquer protocolo e é compatível com Strings, imagens e JSON bruto. Pode ser usada para buscar uma página de resultados de pesquisa como dados estruturados. Como neste trabalho envolve uma chamada com a API do Mercado Livre (conforme foi explicado na subseção 4.3.1), o

¹⁶ <https://developer.android.com/reference/android/support/v4/view/PagerAdapter> - Acesso em março de 2020

¹⁷ <https://developer.android.com/guide/topics/ui/dialogs> - Acesso em março de 2020

¹⁸ <https://developer.android.com/training/volley> - Acesso em janeiro de 2020

que por sua vez precisamos de conexão com a rede, esta biblioteca está sendo utilizada para realizar o trabalho. Mais informações estão disponíveis na página oficial de desenvolvimento do Android, assim como seu Github¹⁹ (em Inglês).

- *Glide*

*Glide*²⁰ é um *framework* de gerenciador de medias e carregador de imagens totalmente aberto para Android. Proporciona uma interface fácil e simples de utilizar, e seu principal objetivo é carregar qualquer tipo de mídia da maneira mais suave e rápida possível. Busca, decodificação e exibição dos vídeos, imagens ou animações, além de possibilidade de redimensionar e exibir imagens remotas são as capacidades que a *framework* consegue realizar. Sabendo que este trabalho exigiria exibição das imagens dos produtos da maneira mais otimizada, esta foi a *framework* escolhida para realizar a tarefa. Sua documentação e uso está totalmente disponível no Github²¹ oficial dos criadores.

- *StateProgressBar*

StateProgressBar é uma biblioteca que exibe e realiza os estados como uma barra de progresso. Comumente é utilizado em formulários, ou processos de compras, que exigem normalmente mais de uma etapa para concluir. Com os estados, é possível acompanhar em que etapa o usuário se encontra, e quantas faltam para chegar até o final. No ARShop, este recurso está sendo utilizado no processo de compra de um produto. Quando um usuário opta por comprar os itens que possui em seu carrinho, entra na etapa de pagamento. Nesta etapa, o usuário poderá visualizar quantas etapas deverá realizar até chegar ao processamento e conclusão da compra. Na Figura 4.12 temos um exemplo do *StateProgressBar*. Sua documentação e uso está totalmente disponível no Github²² oficial dos criadores.

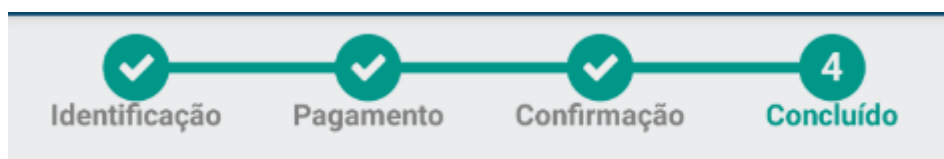
¹⁹ <https://github.com/google/volley> - Acesso em janeiro de 2020

²⁰ <https://github.com/bumptech/glide> - Acesso em fevereiro de 2020

²¹ <https://github.com/bumptech/glide> - Acesso em fevereiro de 2020

²² <https://github.com/kofigyan/StateProgressBar> - Acesso em maio de 2020

Figura 4.12 – Exemplo de StateProgressBar



Fonte: O Autor

4.5 Interfaces

Nesta seção teremos como foco mostrar cada tela do ARShop. Conforme explorado na subseção 2.8 do Capítulo 2 sobre *User Experience*, contando que a proposta é ser um aplicativo focado em vendas de móveis, as cores principais optadas para o aplicativo são o cinza e o azul. O cinza foi optado devido a neutralidade, enquanto o azul para passar a sensação de confiança e segurança ao cliente. A cor cinza no aplicativo está com um leve degrade com o branco. Portanto o fundo não é totalmente cinza. Para os botões de ações regulares, o azul também foi optado para manter o padrão com a interface no geral. Botões que indicam a intenção de comprar algum produto e finalização do mesmo, as cores escolhidas foram verde e vermelha. O verde é escolhido por ser uma cor calma associado a saúde e ao dinheiro. Normalmente é a cor escolhida por um grande número de sites de e-commerce para associar a compras, pois é uma cor facilmente processada e absorvida pelo usuário. A vermelha foi optada unicamente para o botão de finalização da compra devido a sua forte e quente tonalidade. Desperta sensação de urgência e, neste caso, uma necessidade de que a compra precisa ser realizada. Para as fontes, o aplicativo está em sua maioria com o texto na cor padrão escura (preta), tendo alguns títulos e informações com uma tonalidade cinza escura. Os botões com texto na cor branca. As fontes escolhidas para o aplicativo são *Latto* e *Montserrat Bold*. Ambas apresentam uma escrita agradável, e passam a mensagem clara ao consumidor.

Dada as escolhas e os motivos utilizados na interface, serão exibidos abaixo cada uma das telas dando uma breve descrição com o que cada tela possui. As imagens estão listadas como se o usuário estivesse usando o aplicativo. Então iniciará pelas telas login e criação de conta, depois a tela inicial do aplicativo pós login, visualização de produtos, detalhe de produto, e assim por diante.

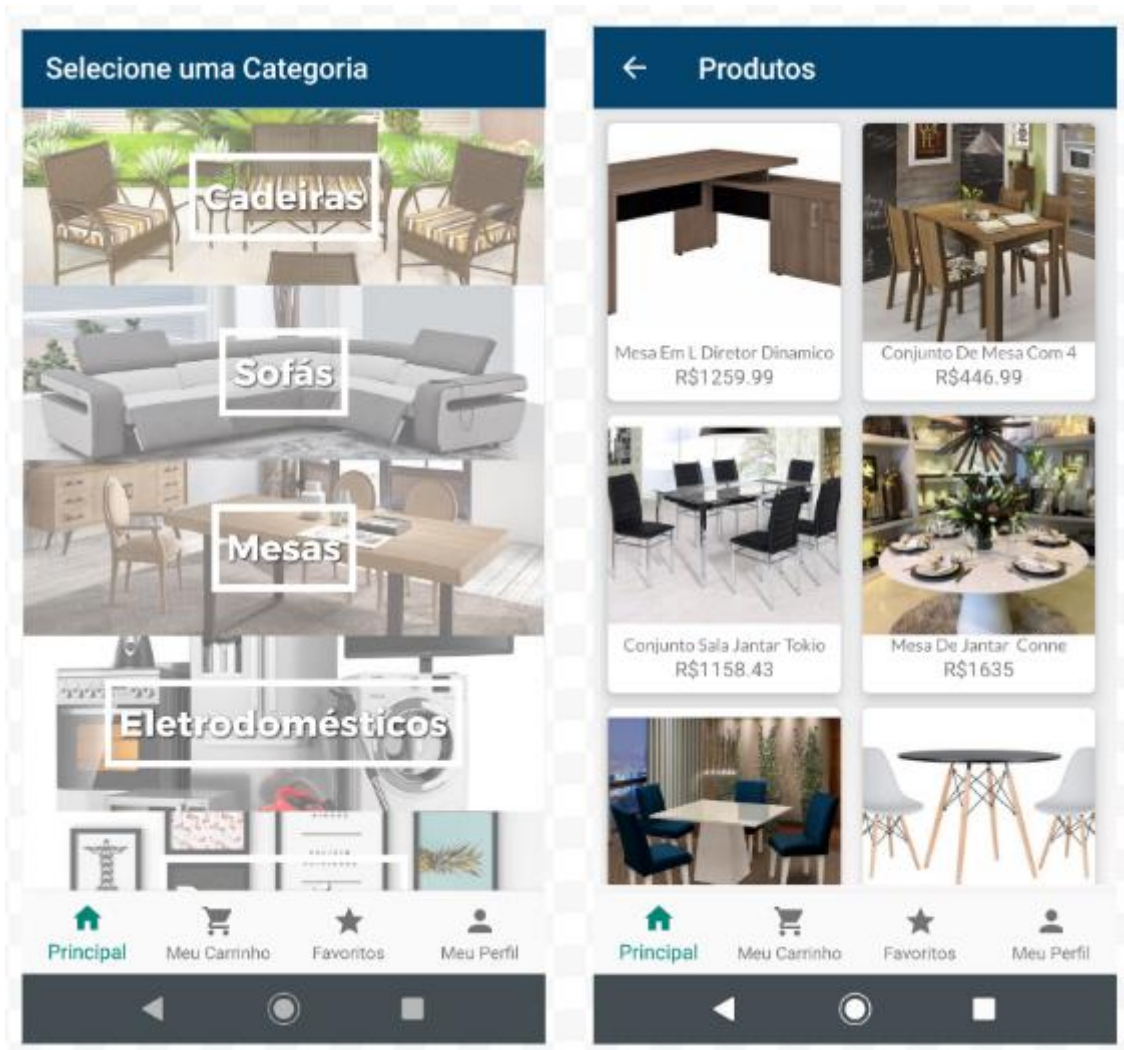
Figura 4.13 - Tela de Login e cadastro

The image displays two side-by-side mobile application screens. The left screen, titled 'Iniciar Sessão (Login)', features a header with the text 'Iniciar Sessão (Login)'. Below the header is a banner image showing cardboard boxes with shopping cart icons and the text 'ARShop'. The main content area includes a form with the following elements: a label 'Seu Email:' followed by an 'Email' input field; a label 'Informe sua Senha:' followed by a 'Senha' input field; a dark blue button labeled 'ENTRAR'; a link 'Esqueceu sua senha?'; and a link 'PRIMEIRA VEZ? CLIQUE AQUI'. The right screen, titled 'Cadastro novo usuário', has a back arrow and the title 'Cadastro novo usuário'. It is divided into two sections: 'Informações Básicas do Usuário' with fields for 'Primeiro Nome *', 'Sobrenome *', 'CPF (Apenas números) *', and 'Data de Nascimento (dd/mm/yy) *'; and 'Informações de Login' with fields for 'Seu email *', 'Informe sua Senha *', and 'Confirme sua Senha *'. A dark blue button labeled 'CADASTRAR' is positioned below the login information fields. Both screens have a dark grey Android-style navigation bar at the bottom.

Fonte: O Autor

Na Figura 4.13 temos a tela de login do usuário (Imagem a esquerda). Aqui é basicamente onde o usuário deve colocar suas informações de conta para utilizar o aplicativo. Caso não possuir, clicando em “Primeira Vez? Clique Aqui” localizado na parte mais baixa da tela, o usuário é movido para a tela de cadastro ilustrada na imagem mais à direita. No Cadastro, basicamente é solicitado as informações básicas do usuário, podendo definir outras informações posteriormente, como endereço ou pagamentos (cartão de crédito).

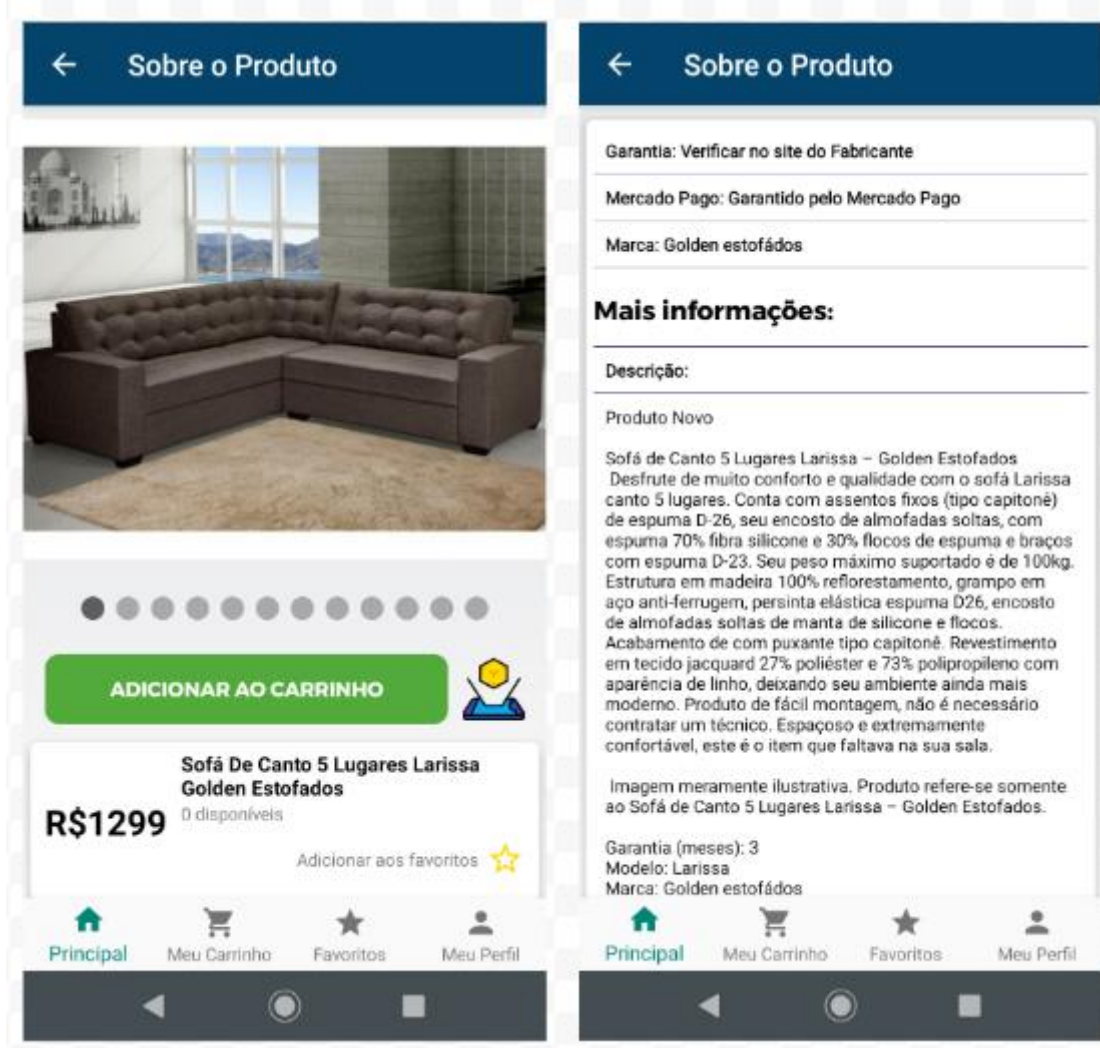
Figura 4.14 - Tela inicial pós login e tela de produtos



Fonte: O Autor

Na Figura 4.14 temos a tela inicial do aplicativo. O usuário ao “logar” entra numa tela de categorias de produtos, onde deve escolher qual categoria gostaria de ver. Na figura a direita, supondo que o usuário escolha a categoria “Mesas”, a tela carregará os produtos correspondentes a categoria escolhida.

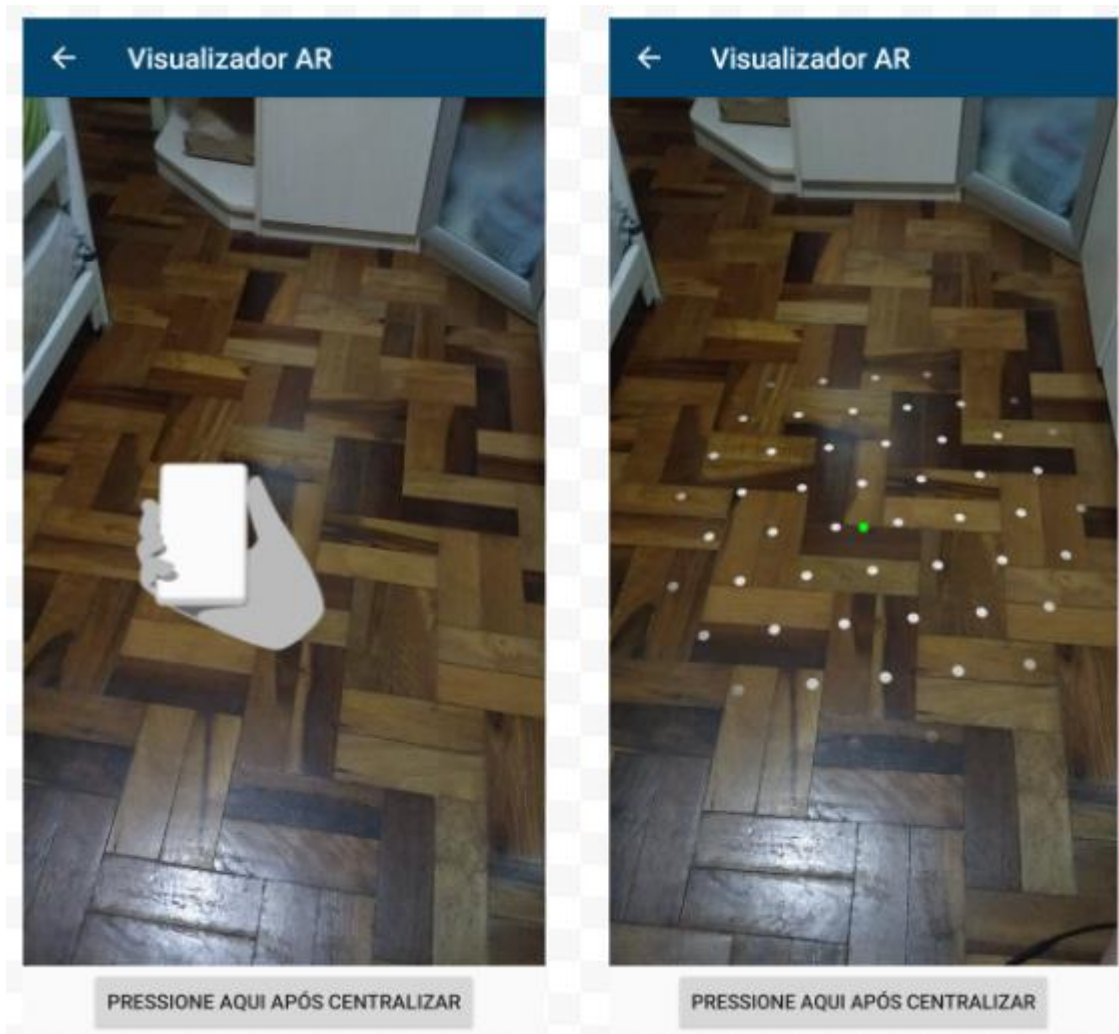
Figura 4.15 - Tela de detalhe de um produto



Fonte: O Autor

A Figura 4.15 mostra a tela de detalhes de um sofá. Esta é a tela caso o usuário queira saber mais sobre um produto. Como podemos ver na figura a esquerda, temos a imagem do produto, e uma série de outras que podemos arrastar na tela para esquerda para visualizar. Logo embaixo temos o botão para adicionar ao carrinho, e ao lado um botão ilustrando a opção para visualizar o produto em realidade aumentada. E em seguida, temos as informações básicas do produto, onde temos uma estrela que podemos clicar para adicionar o produto aos favoritos. Na figura a direita podemos ver as outras informações mais detalhadas do produto.

Figura 4.16 – Telas de visualização em AR.



Fonte: O Autor

A Figura 4.16 mostra as telas para realizar o produto em AR. Como podemos ver nas imagens, temos a câmera do *smartphone* sendo utilizada. Mais a esquerda, temos uma ilustração no meio da tela demonstrando ao usuário que é necessário “nivelar” e centralizar o celular como forma de processar o ambiente que existe ao redor. Ao encontrar uma superfície plana e centralizar, a imagem a direita será exibida. Onde ele tem as informações do plano e estará pronto para exibir o produto ou modelo em AR. Neste momento, o usuário poderá clicar no botão localizado na parte mais baixa da tela para exibir o modelo em AR na sua frente.

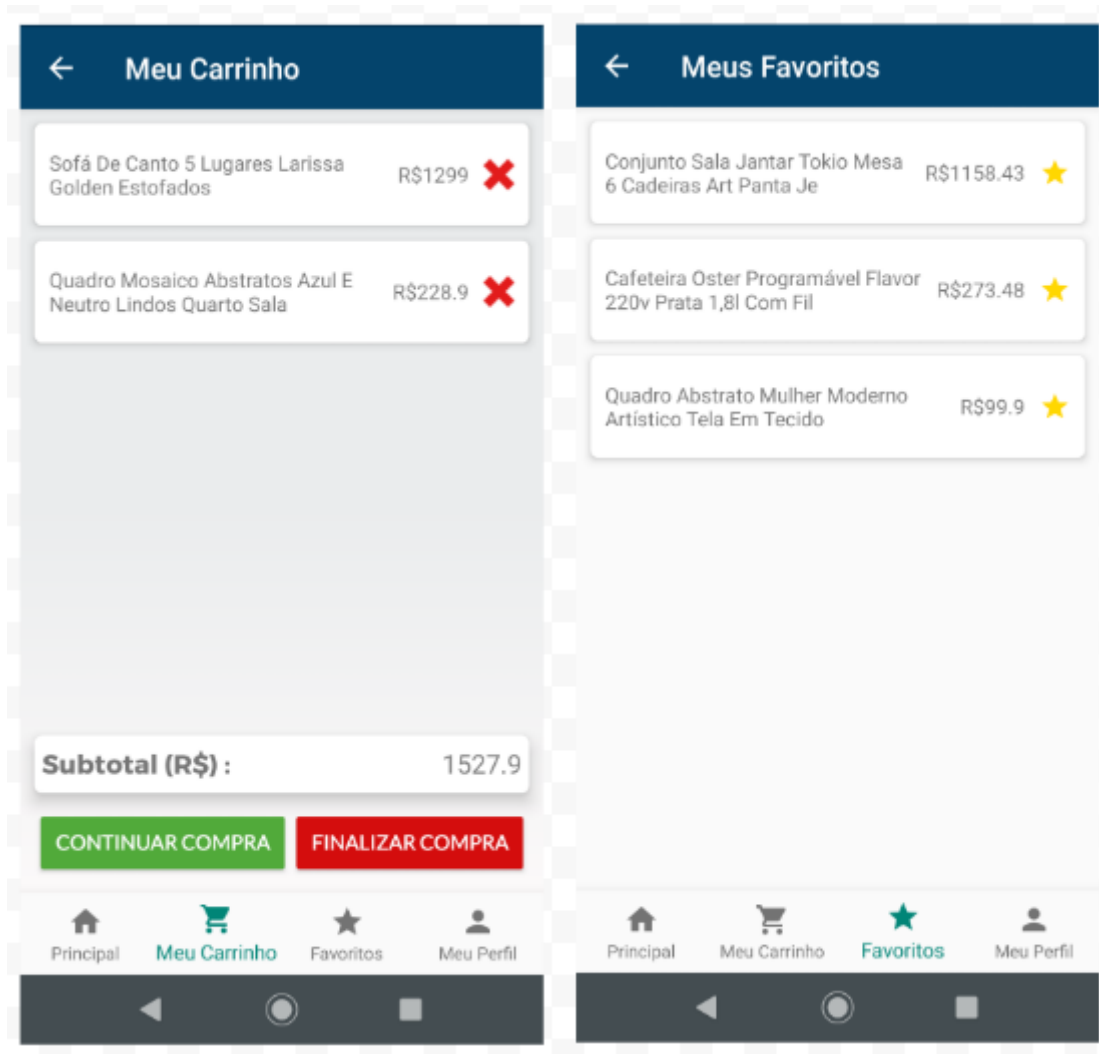
Figura 4.17 – Exemplo de um móvel reproduzido em AR.



Fonte: O Autor

Na Figura 4.17, temos um exemplo de um móvel sendo projetado em realidade aumentada. Como podemos ver nas imagens, é possível visualizar o produto em diferentes ângulos. A figura a direita é melhor visualizada com a tela na horizontal, a ideia no caso era visualizar o máximo do produto na tela disponível.

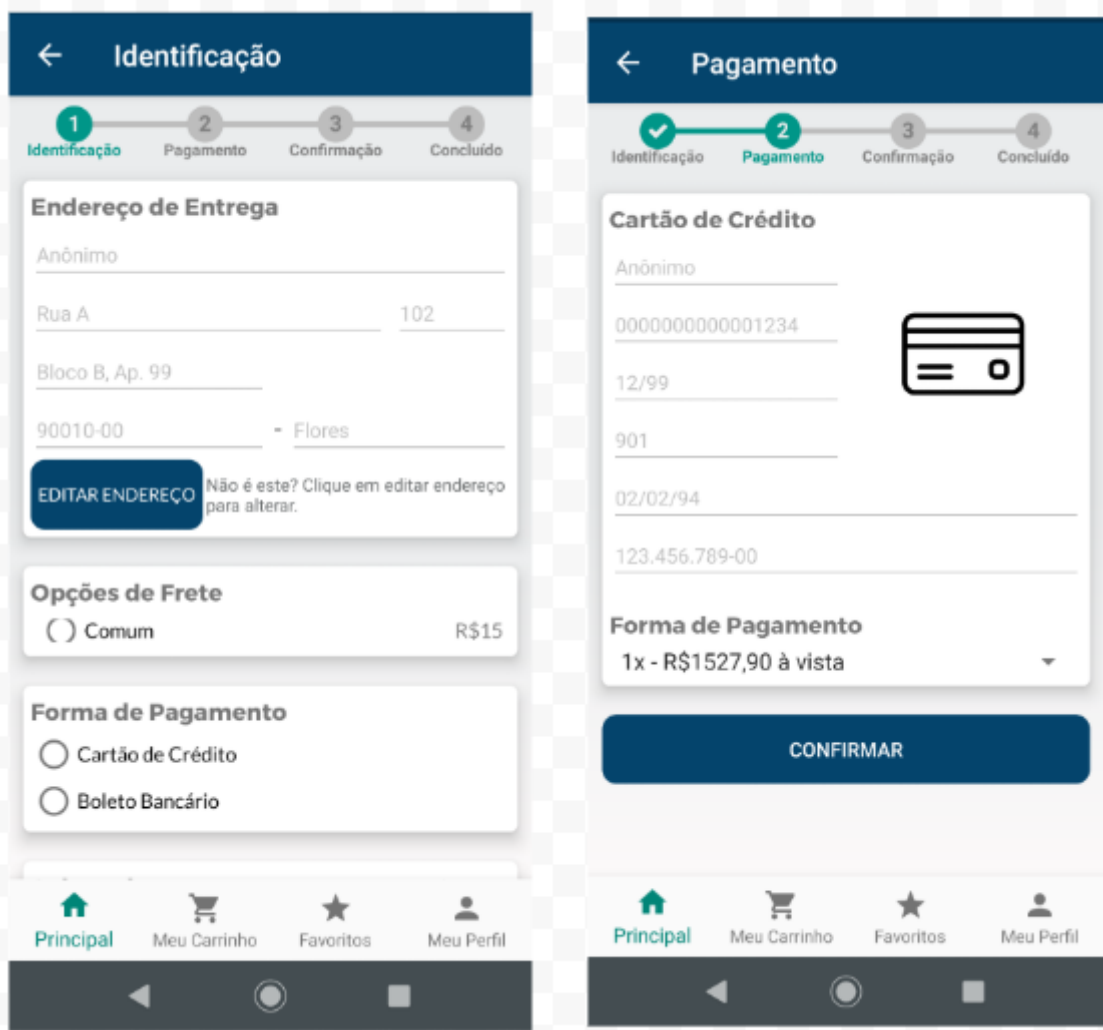
Figura 4.18 - Tela “Meu Carrinho” e “Favoritos” do usuário



Fonte: O Autor

Na Figura 4.18 temos as telas de Meu Carrinho e Meus Favoritos. Basicamente aqui são as telas onde o usuário pode visualizar os produtos que deseja comprar. E os produtos que deseja colocar nos favoritos.

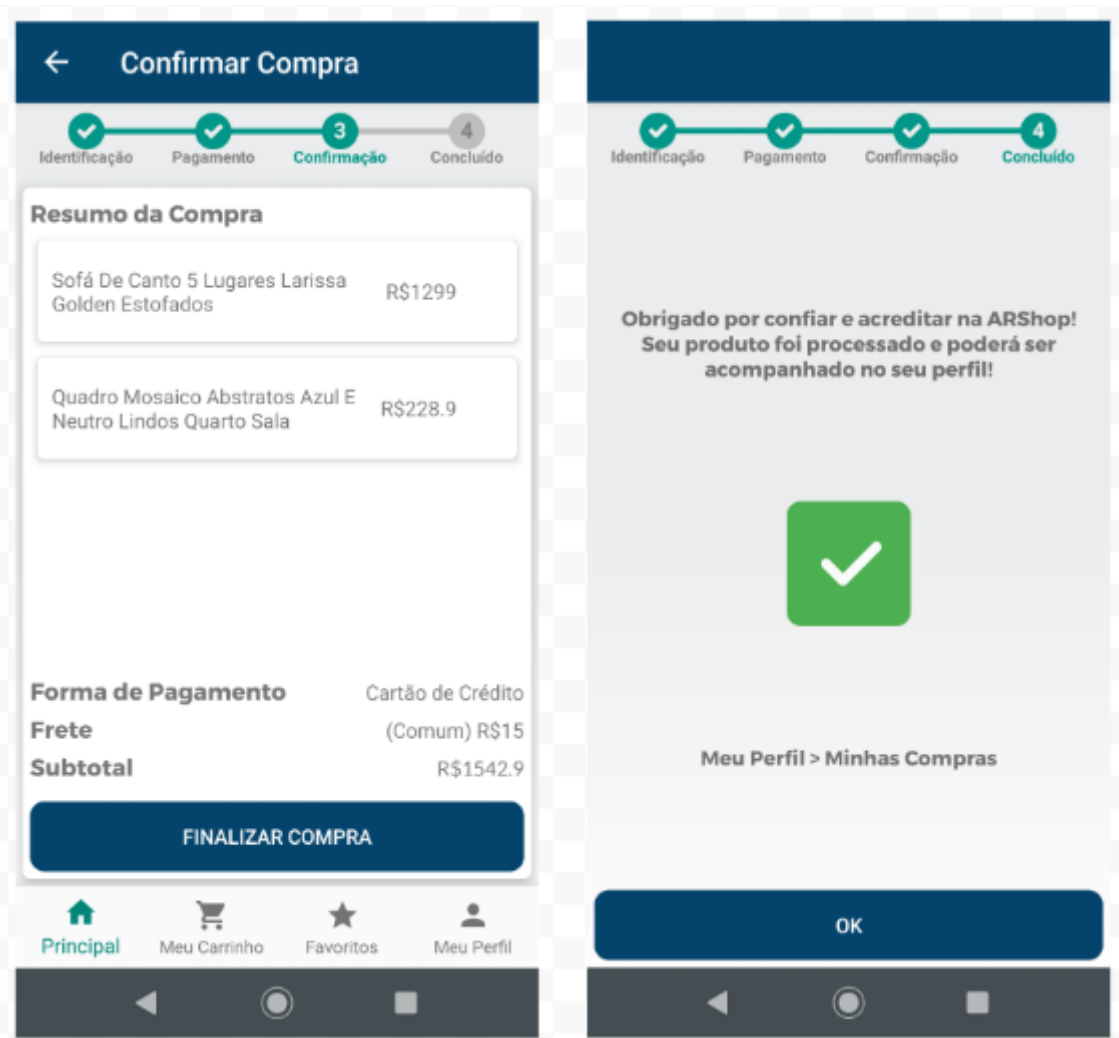
Figura 4.19 - Tela inicial de processo de compra. Dados totalmente genéricos



Fonte: O Autor

Na Figura 4.19 temos as telas de processo de compra de um produto. A esquerda temos a tela mostrando as informações básicas de entrega. Primeiramente endereço de onde o produto será entregue, e em seguida as opções de frete e forma de pagamento. A direita temos a continuação da tela da esquerda, onde o usuário confirma as informações de cartão de crédito caso tenha optado por ele e em quantas vezes deseja.

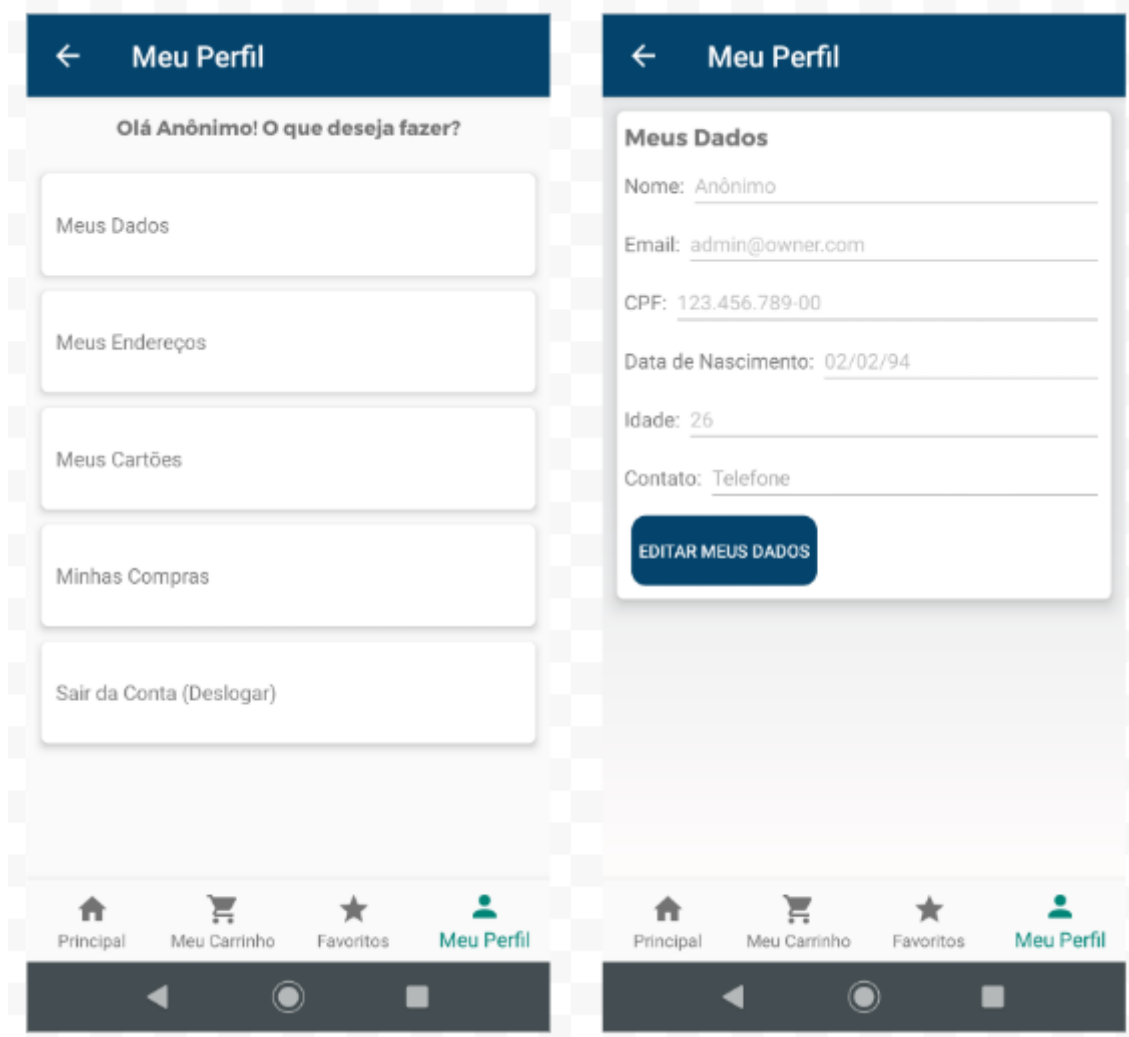
Figura 4.20 - Telas de finalização de compra



Fonte: O Autor

Na Figura 4.20 temos a última tela do processo de compra (continuação da figura 4.19). A esquerda temos basicamente um último resumo de tudo que realizou antes e deve confirmar para fechar de vez a compra. A direita temos apenas a tela de finalização da compra.

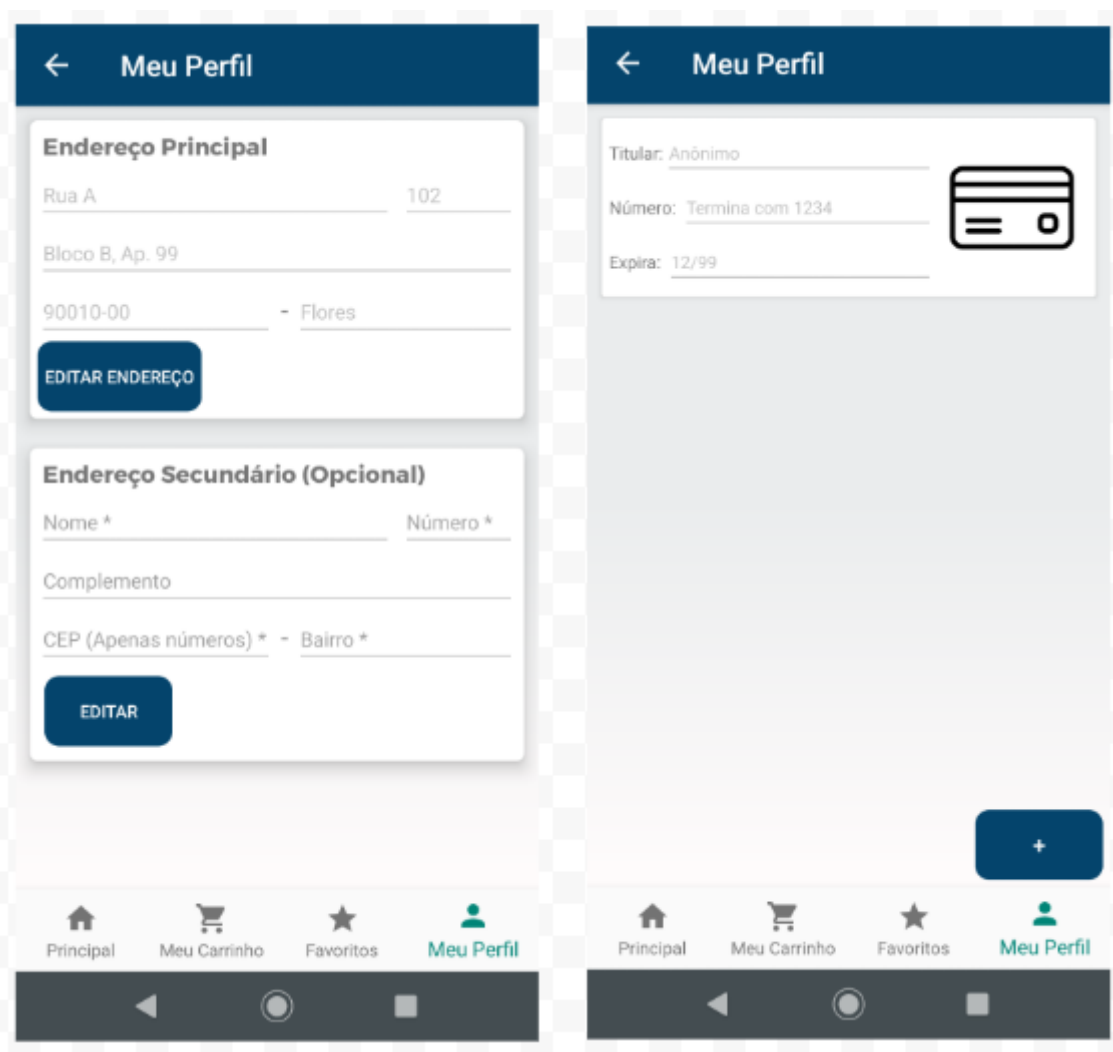
Figura 4.21 - Telas de Perfil do Usuário e tela da opção “Meu Dados” (Configurações)



Fonte: O Autor

Na Figura 4.21 temos basicamente a tela de configurações. Aqui podemos ver e editar os dados do usuário. A esquerda basicamente temos as opções disponíveis, e a direita a tela de visualização dos dados básicos do usuário. A única tela que não será exibida será a opção de “Sair da Conta”, pois obviamente apenas volta a tela de Login exibida na Figura 4.13.

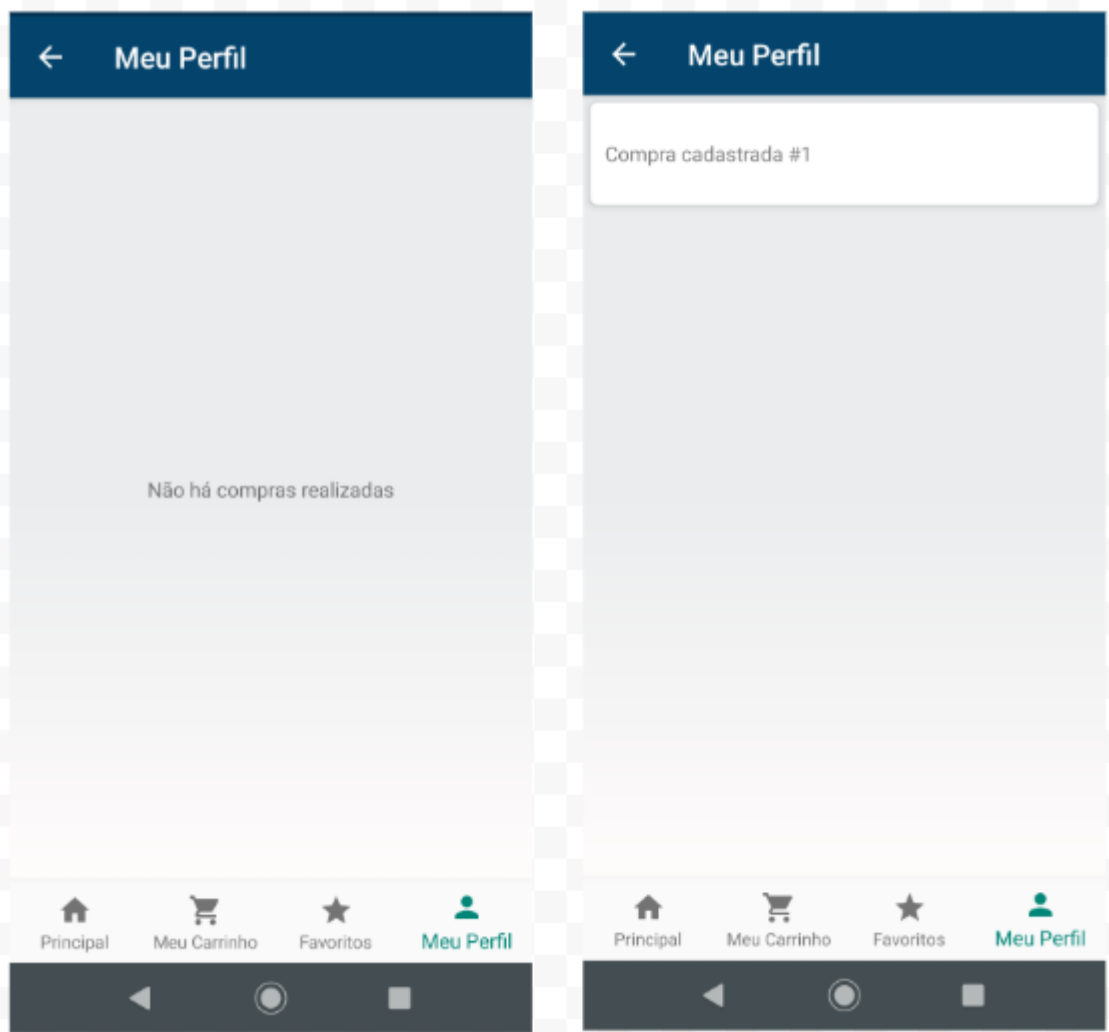
Figura 4.22 - Tela de opções “Meus Endereços” e “Meus cartões”



Fonte: O Autor.

Na Figura 4.22 temos as opções de visualização de endereço e cartão de crédito do usuário. A esquerda o usuário consegue visualizar seus endereços, podendo adicionar um segundo endereço caso desejar. E a direita a tela dos cartões cadastrados, onde o usuário também pode adicionar novos cartões caso desejar.

Figura 4.23 – Tela de opções “Minhas Compras”



Fonte: O Autor

E por fim, na Figura 4.23 temos a opção de visualizar o histórico de compras do usuário. A esquerda podemos ver a tela quando o usuário não comprou nada ainda no aplicativo. E na direita temos um exemplo de uma compra feita pelo usuário. No *card* é exibido apenas um texto que foi implementado de maneira *hardcoded* para fins ilustrativos. Portanto, para cada compra realizada, o mesmo texto é apresentado mudando apenas a numeração no final adicionando uma unidade a cada compra. A próxima compra seria colocada como “Compra cadastrada #2”.

4.6 Considerações Finais

Tendo apresentado o desenvolvimento sobre o ARShop, é importante tomar nota sobre alguns pontos. Todos elementos não implementados diretamente pelo autor deste trabalho

foram verificados durante o seu desenvolvimento. Imagens, frameworks, os próprios modelos 3D utilizados para realidade aumentada, e outros, estão devidamente referenciados ou aqui neste documento, ou no GitHub²³ do autor deste trabalho. É importante relatar também que não são todos os produtos que contém a realidade aumentada disponível. Seria necessário a criação do modelo 3D para cada produto, o que seria impraticável neste trabalho. Todas questões relacionadas a dificuldades e problemas no desenvolvimento estarão sendo tratadas no Capítulo 6.

²³ <https://github.com/RodrigoOkido/ARShop> - Acesso em janeiro de 2020

5 EXPERIMENTOS

Neste Capítulo é mostrado um pouco dos experimentos realizados com o aplicativo com usuários reais. Para este trabalho, foram convidados 10 usuários de diferentes faixas etárias para que experimentassem o aplicativo. Não foram considerados a experiência das pessoas no uso de aplicativo de compras ou até mesmo no uso geral. Uma vez realizado o experimento, o usuário era convidado a responder um questionário relatando como foi o uso do aplicativo. Nas próximas subseções serão detalhados o experimento, atividades e os resultados obtidos.

5.1 O Experimento

Para o experimento, foi utilizado o *System Usability Scale* (Escala de Usabilidade do Sistema ou SUS). Este método foi criado por John Brooke em 1986 em um laboratório localizado no Reino Unido chamado *Digital Equipment Corporation*. Basicamente, este sistema é um questionário composto por 10 perguntas com 5 opções de resposta para cada pergunta. Essas opções são mostradas no que chamamos de escala Likert, o qual varia de “Discordo Totalmente” a “Concordo Totalmente”. Atualmente, o SUS já foi testado em diversos tipos de testes e se tornou um padrão da indústria com referências em mais de 600 publicações. Alguns exemplos onde o sistema foi utilizado foi para *hardware*, *softwares* de consumo, *sites*, tele móveis, entre outros.

Uma vez aplicado o experimento, para saber o resultado é necessário para cada item ímpar subtrair 1 a resposta do usuário, enquanto os pares o score é 5 menos a resposta do usuário. As questões ímpares são de teor positivo em relação à aplicação. Neste caso, é desejável que as respostas dos usuários tendam a ‘Concordar’ nas perguntas das questões. Por outro lado, as questões pares são negativas e, portanto, para serem bem avaliadas o ideal é que os usuários respondam tendendo a ‘Discordar’ com a pergunta. Obtendo o valor para cada item, soma-se tudo e multiplica por 2.5. Esse valor final vai resultar em um valor entre 0 a 100. Em média, para se considerar um bom score, o ideal é que o teste resulte em pelo menos 70%.

Para complementar o teste, um questionário de perfil de usuário foi realizado. Basicamente, este questionário é para termos uma noção demográfica sobre quem são as pessoas em que estes testes foram realizados. Aqui serão perguntas voltadas ao usuário e serão coletados de forma anônima. Pegando informações básicas e um pouco de sua experiência com

compras *online*. O resultado deste questionário de perfil de usuário não será computado no teste SUS, uma vez que já é um teste validado e bem definido.

5.2 Atividades Realizadas

Cada usuário do experimento foi solicitado para fazer 3 tarefas. Para realização delas, uma demonstração (treinamento) era feita mostrando o que era desejado. Em seguida, o usuário era solicitado para fazer por conta. Segue elas:

1. Criar uma conta de usuário nova.
2. Uma vez logado no aplicativo (Uma conta dada), adicionar 2 produtos da loja (a escolha do usuário) no carrinho e finalizar o processo de compra dos mesmos. Dentro desta atividade era solicitado os seguintes extras:
 - a. Pelo menos um dos produtos deve ser adicionado aos favoritos.
 - b. Após finalizar a compra, verificar a compra realizada no seu Perfil.
3. Visualizar um produto em realidade aumentada.

Como podemos ver, as atividades 1 e 3 são mais específicas, enquanto a 2 é onde o usuário de fato navegará por boa parte do app. As informações de CPF e Cartão de Crédito foram totalmente ignoradas no experimento. Por serem dados sensíveis, ambos já foram automaticamente preenchidos com dados fictícios e não precisavam ser colocados. Por isto, no cadastro de um novo usuário, o campo CPF era um campo opcional. Enquanto na atividade 2, a conta fornecida já continha dados de cartão de créditos fictícios inclusos (além do próprio CPF). Na criação de conta, não é necessário cadastrar um cartão de crédito e nem endereço. Porém, pra comprar algum produto esses dados precisam ser preenchidos dentro da conta.

Na atividade 3 basicamente ensinava o usuário a como projetar um produto com a câmera usando a realidade aumentada. Aqui era onde o usuário poderia ver o diferencial da aplicação.

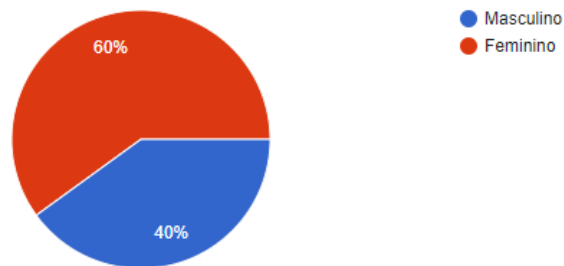
5.3 Resultados

Os resultados obtidos tanto para os dados de perfil (demografia), quanto para o questionário SUS podem ser observados abaixo. O questionário foi inteiramente realizado pelo Google Forms, e com isto, os resultados gráficos são reproduzidos automaticamente.

Figura 5.1 – Perfil Demográfico (Parte 1 de 3)

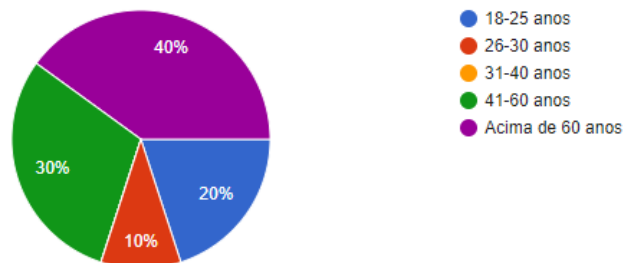
1. Informe seu sexo

10 responses



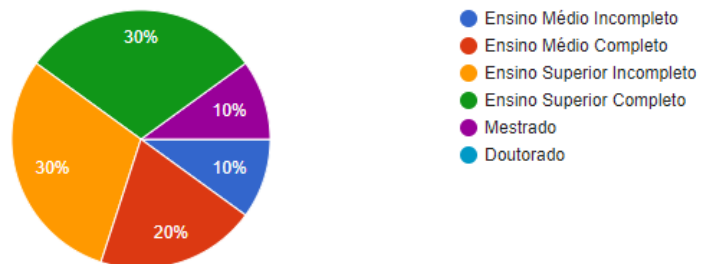
2. Qual sua faixa etária?

10 responses



3. Qual seu nível de escolaridade?

10 responses



Fonte: O Autor

Figura 5.2 – Perfil Demográfico (Parte 2 de 3)

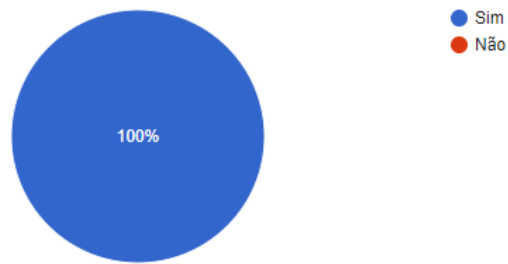
4. Com que frequência você acessa a internet pelo celular?

10 responses



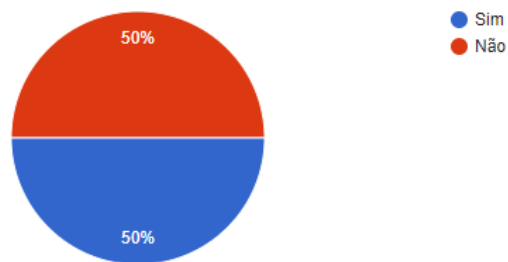
6. Você alguma vez já comprou ou costuma fazer compras na internet?

10 responses



7. Você já comprou algum móvel pela internet?

10 responses

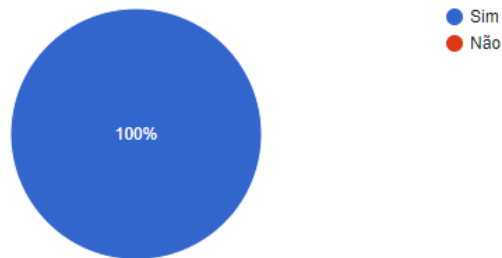


Fonte: O Autor

Figura 5.3 – Perfil Demográfico (Parte 3 de 3)

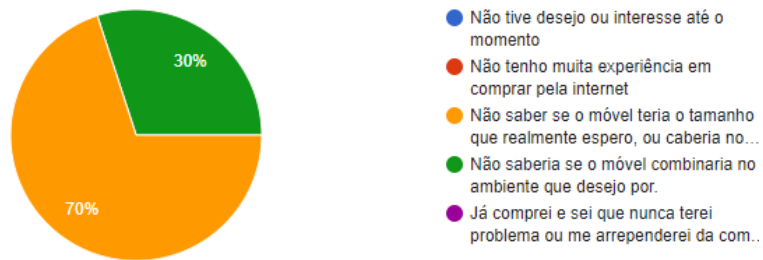
8. Caso não tenha, em algum momento já sentiu vontade de comprar algum tipo de móvel na internet? (Opcional. Apenas se respondeu "não" na pergunta 7)

7 responses



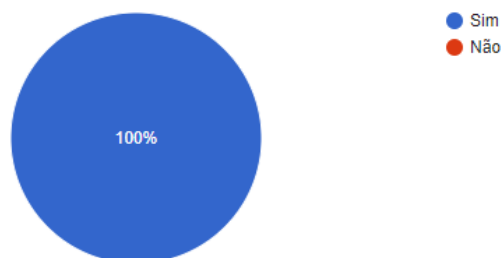
9. Caso você já tenha desejado e tivesse condições financeiras de comprar um móvel pela internet, porém não comprou, dentre as opções abaixo qual poderia se encaixar como o maior motivo que impedisse sua compra?

10 responses



10. Se você pudesse pré-visualizar o móvel desejado em tamanho real (ver o produto como se estivesse na loja) com a câmera do seu celular no ambiente que gostaria de colocá-lo, você se sentiria mais seguro e confiante para realizar sua compra?

10 responses



Fonte: O Autor

Errata: Podemos observar que a numeração vai de 4 para 6 (não tendo a questão de número 5). Porém, a questão não foi ignorada, sendo apenas um erro na numeração das questões.

Podemos observar nas Figuras acima (Figura 5.1, Figura 5.2 e Figura 5.3), que basicamente maioria dos entrevistados são acima de 40 anos (somados 70%). Dentre os candidatos, apenas 30% tem ensino superior completo e 10% mestrado. Todos convidados acessam celular todos os dias, e já compraram alguma coisa na internet. Entretanto, apenas 50% afirma que comprou um móvel. Dentre as justificativas para isto, podemos observar na questão 9 que 70% dos candidatos responderam que não compraram pelo motivo de não saber se o móvel teria o tamanho esperado. Enquanto 30% afirmou não saber se combinaria com o ambiente em que deseja colocar. Todos participantes responderam sim quanto a se sentir mais confiante em comprar um produto caso pudesse pré-visualizar os mesmos, o que seria uma resposta no mínimo esperada.

Tabela 5.1 – Resultado do questionário SUS.

Questões	Concorda	Neutro	Discordo
Q1. Eu gostaria de utilizar esse aplicativo com frequência	100%	0%	0%
Q2. Eu acho o aplicativo desnecessariamente complexo	0%	0%	100%
Q3. Eu achei o aplicativo fácil de usar	100%	0%	0%
Q4. Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o aplicativo	0%	0%	100%
Q5. Eu acho que as funções do aplicativo estão bem integradas	100%	0%	0%
Q6. Eu acho que o aplicativo apresenta muita inconsistência	0%	0%	90% / 10% (Discordo Totalmente / Discordo)
Q7. Eu imagino que as pessoas aprenderão	100%	0%	0%

como usar esse aplicativo rapidamente			
Q8. Eu achei o aplicativo atrapalhado de usar	0%	0%	100%
Q9. Eu me senti confiante usando o aplicativo	100%	0%	0%
Q10. Eu precisei aprender várias coisas novas antes de usar o aplicativo	0%	0%	100%

Fonte: O Autor

Como podemos observar na Tabela 5.1, todos usuários convidados não tiveram grandes dificuldades na utilização do aplicativo. Praticamente todos responderam entre “Concordo Totalmente” e “Discordo Totalmente”. Tendo apenas 1 que respondeu “discordo” na questão 6. Retomando a explicação na subseção 5.1 sobre o SUS, para verificarmos o resultado obtido temos que fazer o seguinte:

- Para cada item ímpar subtrair 1 a resposta do usuário
- Para cada item par colocar 5 menos a resposta do usuário.
- Tendo isto feito, soma-se tudo e multiplica por 2.5 e terá um valor entre 0 a 100.

Aplicando estas 3 etapas, o resultado obtido foi uma média de 99.75 entre os 10 usuários. O que segundo o teste de usabilidade é excelente e bem acima do recomendado (*score* 70). Todos os candidatos durante o experimento não tiveram dificuldade de utilizar o aplicativo. Conseguiram visualizar as opções e as informações apresentadas sem muitos problemas. Uma curiosidade que aconteceu durante os testes foram usuários relatando experiências de frustrações que tiveram na compra de algum móvel. Como por exemplo terem comprado uma mesa, porém a mesa ao chegar no apartamento não cabia como esperava.

6 CONCLUSÃO

Este trabalho teve o propósito de apresentar um aplicativo de e-commerce focado em móveis chamado ARShop. O grande diferencial do aplicativo é a possibilidade de visualizar os produtos em Realidade Aumentada, além de proporcionar uma experiência de uso leve e agradável no geral. A base de dados optada para o ARShop foi o Mercado Livre. Além de possuir uma imensa base de dados, contém uma API rica e aberta para desenvolvedores. Usando esta API em conjunto com a realidade aumentada, o aplicativo tem como objetivo auxiliar o cliente no poder de decisão sobre o produto que deseja adquirir, uma vez que não temos como saber muitas vezes se o produto irá atender totalmente as expectativas do usuário quando compramos pela *internet*. E por se tratar de móveis, suas razões normalmente estão associadas as suas dimensões, além de não saber se o produto vai combinar com o ambiente em que o deseja colocar.

Ao longo do desenvolvimento do aplicativo, alguns pontos foram observados. O primeiro é o fato de a realidade aumentada não ser um recurso ainda disponível para muitas pessoas. No Android, por exemplo, o ARCore foi incrementado ao sistema na sua versão 7.0 (2016). Uma vez que não temos um sistema operacional na sua versão mínima, caso haja algum aplicativo que seja total requerido a Realidade Aumentada, não será possível sua utilização devido a limitação do sistema. Além disto, é necessário que o hardware do aparelho tenha todos os requisitos necessários para que o ARCore possa funcionar de maneira esperada (CPU, sensores, câmera). Neste caso, de nada adianta o aparelho ter o sistema operacional mais recente, se o hardware não tem capacidade para processar e realizar tudo que uma aplicação AR exigir. Apesar de termos dois pontos que precisam estar em harmonia para que tudo funcione, esse é um ponto que se resolverá naturalmente com o tempo. Uma vez que a tecnologia avança, melhores hardwares teremos, e o recurso irá consequentemente atingir grande público naturalmente. Segundo especialistas, existem suposições de que a realidade aumentada possa se tornar foco a partir de 2021. Isto apenas mostra o quanto esta tecnologia ainda é nova e está sendo muito estudado e trabalhado no momento. O que se espera é que o mercado vá focar realmente muito neste recurso nos próximos anos.

Hoje, já temos um exemplo popular que foi mostrado neste trabalho, o game Pokémon Go. Este foi um dos jogos que deu uma grande diferença na experiência em jogos. Podendo ter a sensação real de experienciar o mundo Pokémon com o uso da realidade aumentada. Pelo lado E-commerce, ainda não temos um grande número de empresas que aproveitam do recurso. A maioria deles ainda são para mercados específicos, como móveis por exemplo. Isso se deve

provavelmente ao ponto que foi observado anteriormente sobre sua disponibilidade ao grande público. O recurso está aos poucos progredindo e logo teremos uma tecnologia madura e pronta para ser utilizada de diversas maneiras por todos.

Para este trabalho, algumas dificuldades com o uso da realidade aumentada foram enfrentadas. Certos modelos, apesar de serem configurados corretamente, eram projetados como objetos muito maiores do que deveriam. Há dúvidas neste ponto se é algum problema no ARCore em específico, seja na realização dos cálculos para projetar o modelo, ou algum conflito com o próprio modelo que acabe ocasionando o problema. Outro ponto é a questão dos modelos dos produtos. Considerando que é necessário saber modelar em 3D através de programas específicos, por desconhecimento em sua utilização houve um empecilho em realizar o modelo dos produtos. Além disso, considerando que a aplicação é um e-commerce, não havia possibilidade de ter um modelo para todos os produtos. Seu número é grande e certamente seria necessário algum servidor e um banco de dados para armazenar todos os modelos e dados. A solução para este trabalho foi buscar modelos que pudessem ser similares na *web* para fins de demonstração e motivação. Apesar disso, a partir dos experimentos realizados todos os usuários tiveram um feedback positivo sobre poder visualizar o móvel através da câmera e em diferentes ângulos. O que proporcionou uma experiência nova e diferenciada por parte dos usuários.

E por fim, para melhorias futuras, existem diversos itens que poderiam ser considerados se tratando de um *e-commerce app*. Questões relacionadas a *back-end* seriam necessárias. Servidores e bancos de dados para armazenar os produtos, usuários, além dos próprios modelos 3D como já mencionados anteriormente são alguns exemplos. No desenvolvimento deste trabalho, estes itens seriam facilmente adaptáveis no código, uma vez que já foram considerados desde o início do desenvolvimento, e portanto, já contém uma estrutura para isso. Outros pontos que poderiam ser acrescentados envolveriam busca por produtos, e o próprio aumento das categorias de produtos. No primeiro, facilitaria o cliente a buscar mais direto e rápido o que deseja, e no segundo, considerando que a base de dados é o Mercado Livre, nada impede do aplicativo aumentar seus produtos para mais setores que não sejam móveis (Carros, Informática, Brinquedos, etc). Indo um pouco mais além, porém aumentaria em termos de complexidade, seria uma inteligência artificial para recomendar possíveis produtos que o cliente compraria. Uma vez que se conhece o perfil de compra de cada usuário, é bastante comum que sites recomendem produtos similares que possivelmente interessem ao usuário. Uma vez que se tem isso, a probabilidade de a pessoa comprar mais produtos se torna maior e garantido.

REFERÊNCIAS

ACQUIRE. 2019. **10 Problems that every E-commerce Faces and their Solutions [Updated 2019]**. Disponível em: < <https://acquire.io/blog/problems-solutions-ecommerce-faces/> > - acesso em março de 2020.

COMSCORE. 2012. **Improving Product Returns Presents Biggest Opportunity for Retailers Wishing to Increase Online Shopping Satisfaction**. Disponível em: < <https://www.comscore.com/Insights/Press-Releases/2012/6/Improving-Product>Returns-Presents-Biggest-Opportunity-for-Retailers> > - acesso em março de 2020.

ROCK CONTENT. 2019. **O que é E-commerce? Tudo o que você precisa saber para ter uma loja virtual de sucesso**. Disponível em: < <https://rockcontent.com/blog/e-commerce-guia/> > - acesso em janeiro de 2020.

DLOJA VIRTUAL. 2019. **E-commerce no Brasil: perfil do mercado e do e-consumidor**. Disponível em: < <https://www.dlojavirtual.com/e-commerce/e-commerce-no-brasil-entenda-esse-cenario/> > - acesso em março de 2020.

DEV MEDIA. 2013. **Introdução ao Padrão MVC**. Disponível em <<https://www.devmedia.com.br/introducao-ao-padrao-mvc/29308>> - acesso em fevereiro de 2020.

ORACLE DOCS. 2020. **Java Programming Language**. Disponível em: < <https://docs.oracle.com/javase/8/docs/technotes/guides/language/index.html> > - acesso em julho de 2020.

ORACLE DOCS. 2019. **About the Java Technology**. Disponível em: <<https://docs.oracle.com/javase/tutorial/getStarted/intro/definition.html>> - acesso em julho de 2020.

ATLASSIAN. 2020. **What is Git**. Disponível em: <<https://www.atlassian.com/git/tutorials/what-is-git>> - acesso em fevereiro de 2020

Git. 2020. **About – Git**. Disponível em: < <https://git-scm.com/about/branching-and-merging> > - acesso em fevereiro de 2020.

ANDROID. 2020. **What is Android**. Disponível em: < https://www.android.com/intl/pt-BR_br/what-is-android/ > - acesso em julho de 2020.

ANDROID DEVELOPERS. 2020. **Conheça o Android Studio**. Disponível em: <<https://developer.android.com/studio/intro>> - acesso em janeiro de 2020.

DOCS WEB MDN. 2020. **JSON**. Disponível em: < https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON> - acesso em fevereiro de 2020.

MILENE R. SOUSA; JOÃO V. C. BERTOMEU; UX Design na Criação e Desenvolvimento de Aplicativos Digitais. **INFORMÁTICA NA EDUCAÇÃO: teoria & prática**, Porto Alegre, v. 18, n. 2 (2 de julho/dezembro de 2015) 127-143, p. 134-140, 2015. Disponível em: < <https://seer.ufrgs.br/InfEducTeoriaPratica/article/view/54897/36451>>. acesso em maio de 2020.

MARK BILLINGHURS.; ADRIAN CLARK; GUN LEE Survey of Augmented Reality. **Foundation and Trends in Human-Computer Interaction**, Nova Zealandia, v. 8, n. 2-3 (2014) 73-272, p. 77-80, 2015. Disponível em: <https://is.muni.cz/el/1433/podzim2015/PA198/um/59482554/A_Survey_of_Augmented_Reality.pdf>. acesso em maio de 2020.

HOSTINGER. 2019. **O que é UX – User Experience**. Disponível em: < <https://www.hostinger.com.br/tutoriais/ux-o-que-e-user-experience/> > - acesso em abril de 2020.

WEBLIUM. 2020. **Breaking Bad: 21 Bad Website Examples (Upd: 2020)**. Disponível em: < <https://weblium.com/blog/21-bad-website-examples-of-2018/> > - acesso em abril de 2020.

ECOMMERCE-NATION. 2016. **Color Psychology and how to use it in e-commerce**. Disponível em: < <https://www.ecommerce-nation.com/color-psychology-for-ecommerce/> > - acesso em fevereiro de 2020.

NEILPATEL. **Realidade Aumentada e SEO: Descubra o que esperar do futuro**. Disponível em: < <https://neilpatel.com/br/blog/realidade-aumentada/> > - acesso em abril de 2020.

CANAL TECH. 2018. **Especialista faz 5 previsões sobre o futuro das realidades virtual e aumentada**. Disponível em: < <https://canaltech.com.br/rv-ra/especialista-faz-5-previsoes-sobre-o-futuro-das-realidades-virtual-e-aumentada-119024/> > - acesso em abril de 2020.

MEDIUM. 2019. **Top 8 AR furniture Apps Review in 2019**. Disponível em: < <https://medium.com/inborn-experience/top-8-ar-furniture-apps-review-in-2019-1699ff970495> > - acesso em março de 2020.

POST DIGITAL. **O que é realidade aumentada e como ela funciona?**. Disponível em: < <http://www.postdigital.cc/blog/artigo/o-que-e-realidade-aumentada-e-como-ela-funciona> > - acesso em março de 2020.

BLEEZ. 2016. **Vantagens e desvantagens do e-commerce.** Disponível em: < <https://bleez.com.br/blog/vantagens-e-desvantagens-e-commerce/> > - acesso em fevereiro de 2020.

E-COMMERCE BRASIL. 2018. **12 dados que comprovam o crescimento do e-commerce no Brasil.** Disponível em: < <https://www.ecommercebrasil.com.br/artigos/12-dados-que-comprovam-o-crescimento-do-e-commerce-no-brasil/> > - acesso em fevereiro de 2020.

RAFAEL M. BOUCINHA.; LIANE M. R. TAROUCO Avaliação de Ambiente Virtual de Aprendizagem com o uso do SUS - System Usability Scale. **CINTED-UFRGS Novas Tecnologias na Educação**, Porto Alegre, v. 11, n. 3 (dezembro 2013) 1-10, p. 5-9, 2015. Disponível em: <<https://seer.ufrgs.br/renote/article/download/44479/28223>>. acesso em maio de 2020.

APÊNDICE A – QUESTIONÁRIO FINAL APLICADO AOS USUÁRIOS

Neste apêndice estará descrito o formulário criado para os usuários que experimentaram o aplicativo responderem. Aqui possui tanto o questionário de perfil de usuário, quanto o do aplicativo em si.

Avaliação de um aplicativo de E-commerce com Realidade Aumentada (ARShop)

Este questionário faz parte de um experimento para o trabalho de conclusão do curso de Ciência da Computação da faculdade de Informática da Universidade Federal do Rio Grande do Sul. O objetivo é coletar opiniões sobre o aplicativo de compras chamado ARShop. Este aplicativo tem como objetivo ser uma loja online de móveis que tem como diferencial a capacidade de visualizar os produtos em tamanho real a partir da câmera de seu celular através de uma tecnologia chamada Realidade Aumentada. Basicamente, o usuário dentro de sua casa, pode ter uma experiência de como seria o móvel desejado no ambiente que gostaria de colocar. O objetivo desta pesquisa é avaliar como foi sua experiência com a aplicação. O tempo aproximado deste questionário é de 10 minutos e os resultados serão agrupados de forma anônima.

Questionário de Perfil do Usuário

1. Informe seu sexo

Masculino

Feminino

2. Qual sua faixa etária?

18-25 anos

26-30 anos

31-40 anos

41-60 anos

Acima de 60 anos

3. Qual seu nível de escolaridade?

- Ensino Médio Incompleto
- Ensino Médio Completo
- Ensino Superior Incompleto
- Ensino Superior Completo
- Mestrado
- Doutorado

4. Com que frequência você acessa a internet pelo celular?

- Raramente
- 1 vez por mês
- 1 vez por semana
- 3 vezes por semana
- 5 vezes por semana
- Todos os dias

5. Você alguma vez já comprou ou costuma fazer compras na internet?

- Sim
- Não

6. Você já comprou algum móvel pela internet?

Sim

Não

7. Caso não tenha, em algum momento já sentiu vontade de comprar algum tipo de móvel na internet? (Opcional. Apenas se respondeu "não" na pergunta 7).

Sim

Não

8. Caso você já tenha desejado e tivesse condições financeiras de comprar um móvel pela internet, porém não comprou, dentre as opções abaixo qual poderia se encaixar como o maior motivo que impedisse sua compra?

Não tive desejo ou interesse até o momento

Não tenho muita experiência em comprar pela internet

Não saber se o móvel teria o tamanho que realmente espero, ou caberia no espaço onde desejaria colocar.

Não saberia se o móvel combinaria no ambiente que desejo por.

Já comprei e sei que nunca terei problema ou me arrependerei da compra.

9. Se você pudesse pré-visualizar o móvel desejado em tamanho real (ver o produto como se estivesse na loja) com a câmera do seu celular no ambiente que gostaria de colocá-lo, você se sentiria mais seguro e confiante para realizar sua compra?

Sim

Não

Questionário sobre o Aplicativo

A partir daqui, será colocado 10 perguntas sobre como você considerou a sua experiência e usabilidade com o aplicativo. Todas as questões possuem 5 opções, que vão desde "Discordo Totalmente" a "Concordo Totalmente". O que significa que caso você marque 1 em sua resposta, você discorda totalmente do que foi perguntado. E caso marque 5, você concorda totalmente.

1. Eu gostaria de utilizar esse aplicativo com frequência

	1	2	3	4	5	
Discordo Totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo Totalmente

2. Eu acho o aplicativo desnecessariamente complexo

	1	2	3	4	5	
Discordo Totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo Totalmente

3. Eu achei o aplicativo fácil de usar

	1	2	3	4	5	
Discordo Totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo Totalmente

4. Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o aplicativo

	1	2	3	4	5	
Discordo Totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo Totalmente

5. Eu acho que as funções do aplicativo estão bem integradas (ex.: navegar pelo app, adicionar o produto no carrinho, ver as fotos do produto, etc)

	1	2	3	4	5	
Discordo Totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo Totalmente

6. Eu acho que o aplicativo apresenta muita inconsistência

1 2 3 4 5
Discordo Totalmente Concordo Totalmente

7. Eu imagino que as pessoas aprenderão como usar esse aplicativo rapidamente

1 2 3 4 5
Discordo Totalmente Concordo Totalmente

8. Eu achei o aplicativo atrapalhado de usar

1 2 3 4 5
Discordo Totalmente Concordo Totalmente

9. Eu me senti confiante usando o aplicativo

1 2 3 4 5
Discordo Totalmente Concordo Totalmente

10. Eu precisei aprender várias coisas novas antes de usar o aplicativo

1 2 3 4 5
Discordo Totalmente Concordo Totalmente