

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

LUCAS TONDO SENDESKI

**DESENVOLVIMENTO DE UM E-COMMERCE COM CÓDIGO OPEN
SOURCE**

TRABALHO DE CONCLUSÃO DE CURSO

PATO BRANCO
2019

LUCAS TONDO SENDESKI

DESENVOLVIMENTO DE UM E-COMMERCE COM CÓDIGO OPEN SOURCE

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Conclusão de Curso 2, do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, da Universidade Tecnológica Federal do Paraná, *Campus Pato Branco*, como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Prof. Ms. Vinicius Pegorini

PATO BRANCO
2019



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Campus Pato Branco
Departamento Acadêmico de Informática
Curso de Tecnologia em Análise e Desenvolvimento
de Sistemas



TERMO DE APROVAÇÃO
TRABALHO DE CONCLUSÃO DE CURSO

**DESENVOLVIMENTO DE UM E-COMMERCE COM CÓDIGO
OPEN SOURCE**

POR

LUCAS TONDO SENDESKI

Este trabalho de conclusão de curso foi apresentado no dia 02 de julho de 2019, como requisito parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, pela Universidade Tecnológica Federal do Paraná. O acadêmico foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Banca examinadora:

Prof. MSc
Vinicius Pegorini

Profª Drª
Viviane Dal Molin De Souza

Profª Drª
Beatriz Terezinha Borsoi

Prof. Dr. Edilson Pontarolo
Coordenador do Curso de Tecnologia em
Análise e Desenvolvimento de Sistemas

Profª Drª Beatriz Terezinha Borsoi
Responsável pela Atividade de Trabalho de
Conclusão de Curso

A Folha de Aprovação assinada encontra-se na Coordenação do Curso.

Dedicatória: Aos meus pais, irmãos e a toda minha família que, com muito carinho e apoio, não mediram esforços para que eu chegasse até esta etapa da minha vida.

AGRADECIMENTOS

Primeiramente gostaria de agradecer a Deus por ter saúde e disposição para essa etapa da minha vida. Ao Professor, Orientador e Amigo Vinicius Pegorini, pelo apoio, revisões ofertadas, orientação, seu grande desprendimento em me ajudar. Aos amigos que estiveram próximos, ofereceram apoio e pelo incentivo.

Pouco conhecimento faz com que as criaturas se sintam orgulhosas. Muito conhecimento, que se sintam humildes. É assim que as espigas sem grãos erguem desdenhosamente a cabeça para o céu, enquanto que as cheias a baixam para a terra, sua mãe.

Leonardo da Vinci

RESUMO

Com o crescente número de pessoas adotando o comércio eletrônico para aquisição de produtos, muitas empresas precisaram desenvolver novas estratégias para conquistar novos clientes, fidelizar os existentes e otimizar suas vendas. Por isso, uma plataforma de venda *online* propõe facilidade para o cliente efetuar compras, otimizando a relação da empresa com os clientes e unifica todos os clientes em um único lugar. Neste trabalho foi desenvolvido um *e-commerce* utilizando ferramentas de desenvolvimento com código aberto. A solução desenvolvida facilita o controle para o administrador do *e-commerce* e a manutenção do código para o desenvolvedor envolvido.

Palavras-chave: *E-commerce*. *Open Source*. Loja Virtual. Responsividade.

ABSTRACT

With increasing numbers of people adopting e-commerce for product acquisition, many companies needed to develop new strategies to win new customers, build on existing ones, and optimize their sales. Therefore, an online sales platform proposes a facility for the customer to make purchases, optimizing the company's relationship with customers and unifying all customers in one place. In this work an e-commerce was developed using tools of development with open code. The solution developed allows for easy control for the e-commerce administrator and easy maintenance of the code for the developer involved.

Keywords: E-commerce. Open source. Virtual store. Responsiveness.

LISTA DE FIGURAS

FIGURA 1 - PERCENTUAL DE PESSOAS QUE UTILIZARAM A INTERNET	16
FIGURA 2 - CRESCIMENTO DO COMÉRCIO ELETRÔNICO	18
FIGURA 3 - EVOLUÇÃO DE E-CONSUMIDORES NO BRASIL	20
FIGURA 4 - CICLO DE VIDA DO DESENVOLVIMENTO DO PROJETO	29
FIGURA 5 – DIAGRAMA DE CASOS DE USO	32
FIGURA 6 - MODELAGEM RELACIONAL.....	35
FIGURA 7 - TELA INICIAL DA PLATAFORMA	38
FIGURA 8 - TELA INICIAL DA PLATAFORMA COM A LISTA DE PRODUTOS.....	38
FIGURA 9 – TELA INICIAL DA PLATAFORMA COM A LISTA DE PRODUTOS POR CATEGORIA.....	39
FIGURA 10 - TELA DE CADASTRO	40
FIGURA 11 - TELA INICIAL APÓS O LOGIN.....	41
FIGURA 12 - TELA COM A LISTA DE ENDEREÇOS	42
FIGURA 13 - TELA DE INCLUSÃO/EDIÇÃO DE ENDEREÇOS	42
FIGURA 14 - CARRINHO COM USUÁRIO AUTENTICADO	43
FIGURA 15 - CARRINHO COM USUÁRIO NÃO AUTENTICADO	43
FIGURA 16 – CHECKOUT	44
FIGURA 17 - TELA DOS PEDIDOS REALIZADOS	44
FIGURA 18 - TELA DOS ITENS DO PEDIDO.....	45
FIGURA 19 - MODAL DE LOGIN	46
FIGURA 20 - PAINEL DE ADMINISTRAÇÃO	48
FIGURA 21 - TELA DE PEDIDOS	48
FIGURA 22 - TELA DE PEDIDOS EXIBINDO OS ITENS.....	49
FIGURA 23 - ESTRUTURA DO PROJETO CLIENTE.....	50
FIGURA 24 - ESTRUTURA DE PASTAR DO SERVIDOR.....	54

LISTA DE QUADROS

QUADRO 1 - LISTA DE FERRAMENTAS E TECNOLOGIAS.....	28
QUADRO 2 - LISTAGEM DE REQUISITOS FUNCIONAIS	36
QUADRO 3 - LISTAGEM DE REQUISITOS NÃO FUNCIONAIS	37

LISTAGENS DE CÓDIGOS

LISTAGEM 1 - CÓDIGO DE ADICIONAR PRODUTO	51
LISTAGEM 2 - CÓDIGO DE REMOVER PRODUTOS.....	51
LISTAGEM 3 - CÓDIGO DE PEDIDO REALIZADO.....	52
LISTAGEM 4 - CÓDIGO DE LOGIN.....	53
LISTAGEM 5 - CÓDIGO DE LOGOUT.....	53
LISTAGEM 6 - CÓDIGO DO CONTROLLER ABSTRATO	56
LISTAGEM 7 - CÓDIGO DA CLASSE PEDIDOCONTROLLER	57
LISTAGEM 8 - CÓDIGO DO CRUDSERVICE	58
LISTAGEM 9 - CÓDIO DO CRUDSERVICEIMPL.....	59
LISTAGEM 10 - CÓDIGO DA CLASSE PEDIDOSERVICEIMPL.....	60

LISTA DE SIGLAS

API	<i>Application Programming Interface.</i>
CEP	Código de Endereçamento Postal.
CSS	<i>Cascading Style Sheets.</i>
CRUD	<i>Create, Read, Update and Delete.</i>
DLL	<i>Dynamic-Link Library.</i>
DNS	<i>Domain Name System.</i>
HTML	<i>HyperText Markup Language.</i>
IP	<i>Internet Protocol.</i>
MVP	<i>Minimum Viable Product.</i>
PDF	<i>Portable Document Format.</i>
RF	Requisitos Funcionais.
RNF	Requisitos Não Funcionais.
URL	<i>Uniform Resource Locator.</i>

SUMÁRIO

1 INTRODUÇÃO	13
1.1 Considerações Iniciais.....	13
1.2 Objetivos	14
1.2.1 Objetivo Geral	14
1.2.1 Objetivos Específicos	14
1.3 Justificativa	14
1.4 Estrutura do Trabalho	15
2.1 Comercio Eletrônico	16
2.1.1 O Perfil do Consumidor	18
2.1.2 Os Riscos do Consumidor	21
2.1.3 O Valor do Mercado Digital.....	27
3 MATERIAIS E MÉTODO	28
3.1 Materiais.....	28
3.2 Método	28
4 RESULTADOS	31
4.1 Escopo do Sistema	31
4.2 Modelagem do Sistema.....	31
4.3 Requisitos Funcionais e não funcionais	36
4.2 Apresentação das telas do sistema cliente.....	37
4.2.1 Funcionalidades da Plataforma	37
4.2.1.1 Realizar Cadastro.....	39
4.2.1.2 Cadastrar Endereço(s)	41
4.2.2 Carrinho de Compras	43
4.2.3 Checkout	43
4.2.4 Meus Pedidos.....	44
4.2.5 Autenticação.....	45
4.3 Painel de Administração	46
4.4 Apresentação do código do sistema cliente	49
4.4.1 Adicionar e remover produtos do carrinho.....	50
4.4.2 Salvar o Pedido	52
4.4.3 Autenticação do Usuário.....	53
4.5 Apresentação do código do sistema servidor	53
4.5.1 CRUD Controller do servidor	55
4.5.2 CRUD Serice do servidor	57
5 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS	61
REFERÊNCIAS	62

1 INTRODUÇÃO

Neste capítulo são apresentadas as considerações iniciais do trabalho, os seus objetivos e a justificativa, finalizando com a estrutura do texto.

1.1 Considerações Iniciais

Nos últimos anos, o mercado digital vem crescendo significativamente e a população está mais conectada porque o acesso à Internet tem se tornado mais fácil. Como aumento da demanda e os consumidores possuindo cada vez mais segurança para efetuar compras *online*, realizar compras por meio do *marketing* digital se tornou uma ótima alternativa. Os consumidores digitais são categorizados em diversos tipos de perfis, seja ele mais compulsivo ou mais regrado. Contudo, nenhum deles requer idade ou diferenciação de classes, pois jovens, adultos e idosos optam por comprar pela Internet pelo conforto e segurança oferecidos nos dias atuais (BARROS; MUYLDER; OLIVEIRA, 2007).

Com o uso de *marketing* digital, os consumidores passam a ter um papel cada vez mais ativo na economia e em muitas ocasiões passam até a ditar as regras de negócio (ADOLPHO, 2011). Para Gronross (2003) “marketing é estabelecer, manter e ampliar o relacionamento com os consumidores e outros parceiros, para obter lucratividade, com base em objetivos comuns das partes envolvidas”.

Com base no crescente número de consumidores que preferem o comércio eletrônico para realizar suas compras, neste trabalho é apresentado o desenvolvimento de uma aplicação *web* de código aberto para comercializar dispositivos móveis e componentes eletrônicos. A aplicação desenvolvida será responsiva. A responsividade é uma característica importante em sistemas *web*, pois permite acesso por diferentes tipos de dispositivos, tais como computadores, celulares e *tablets*. Para o desenvolvimento do lado cliente da aplicação será utilizado o *framework* Angular, que é baseado na linguagem TypeScript, juntamente com *HyperText Markup Language* (HTML5) e *Cascading Style Sheets* (CSS3). Para o desenvolvimento do lado servidor da aplicação será utilizada a linguagem Java em conjunto com o *framework* Spring. Todas essas ferramentas são de código aberto.

1.2 Objetivos

A seguir estão o objetivo geral e os objetivos específicos definidos para este trabalho.

1.2.1 Objetivo Geral

Desenvolver uma loja virtual, trabalhando com código *open source*, visando agilidade para o administrador do *e-commerce*.

1.2.1 Objetivos Específicos

- Proporcionar melhor controle e gerenciamento dos processos de compra e venda de itens no comércio eletrônico.
- Auxiliar o consumidor na compra de itens dispondo menus e mecanismos de busca que facilitem a agilizem a localização de itens.
- Proporcionar mecanismos de segurança aos compradores utilizando plataformas de pagamento *online*.
- Permitir que o administrador da plataforma tenha facilidade para acessar o código e realizar alterações.

1.3 Justificativa

De acordo com o estudo realizado pela *eMarketer*, uma empresa de pesquisa de mercado em 2012, as vendas globais de comércio eletrônico passaram de \$ 1 trilhão em 2012, pela primeira vez, impulsionado pelo crescimento na América do Norte e na região Ásia-Pacífico. As vendas na América do Norte aumentaram 13,9% em relação ao ano anterior e as vendas na Ásia-Pacífico aumentaram 33% no mesmo período (FREDRICKSEN, 2012).

A forma mais elementar de reutilização de artefatos de software é pelo uso de trechos de código de aplicações já desenvolvidas. Para sistematizar essa prática são criadas bibliotecas - como bibliotecas de funções em linguagem C, por exemplo. Um problema encontrado com o uso de bibliotecas de artefatos de software é a dificuldade de seleção de um artefato adequado à aplicação em desenvolvimento. Se o desenvolvedor concluir que é mais

fácil produzir um novo artefato do que localizá-lo em uma biblioteca, não haverá reutilização (NEIGHBORS, 1991).

A criação de um *e-commerce* permitirá que possíveis alterações sejam feitas a qualquer momento, dando mais liberdade para o programador. A aplicação desenvolvida utilizará plataformas de pagamento *online* visando facilitar o pagamento por parte do consumidor, que poderá utilizar seu cartão de crédito sem preocupar-se com questões de segurança.

1.4 Estrutura do Trabalho

A estrutura do trabalho contém uma relação dos capítulos e uma descrição sucinta do que cada um deles contém. Este primeiro capítulo apresenta as considerações iniciais com o contexto do sistema a ser desenvolvido, os seus objetivos e a justificativa. No Capítulo 2 é apresentado o referencial teórico, que aborda sobre comércio eletrônico, perfil do consumidor e um pouco sobre os riscos do consumidor. Em seguida, no Capítulo 3 estão listados os materiais e o método utilizados no desenvolvimento do trabalho. No Capítulo 4 são apresentados os resultados deste trabalho. No Capítulo 5 é apresentado as considerações finais e as propostas de trabalhos futuros para o funcionamento completo do *E-commerce*. Por fim, as referências que foram utilizadas neste trabalho.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta o referencial teórico trazendo embasamento sobre conceitos relacionados ao comércio eletrônico, descrevendo sobre o seu crescimento e evolução nas últimas décadas, perfil de consumidor e os riscos apresentados no ambiente do comércio eletrônico.

2.1 Comercio Eletrônico

O comércio eletrônico ou mais conhecido como *e-commerce* tem se tornado uma ferramenta tecnológica e essencial para os consumidores, sendo pessoas físicas ou jurídicas, pois ambas podem optar por adquirir produtos ou serviços de forma *online*. Na atualidade, a competitividade das empresas que optarem por trabalhar via *online* está aumentando e muitas delas estão se tornando diferenciais no mercado estando à frente dos concorrentes.

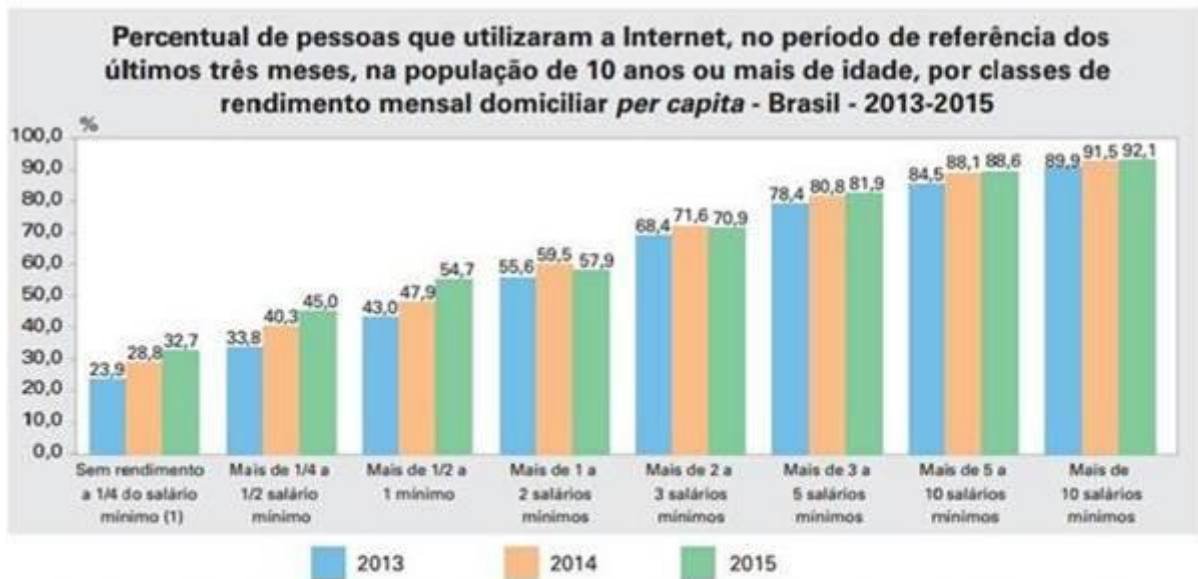
Existem diversas definições distintas para *e-commerce*, todas apresentam uma visão semelhante. Para Balarine, (2002, p. 4) “*E-commerce* são transações que ocorrem via internet, através da ligação entre compradores e vendedores”. Já, segundo Nakamura (2001, p. 31) “Comércio eletrônico é toda atividade de compra e venda realizada com o auxílio de recursos eletrônicos”.

A evolução da internet nos últimos anos, principalmente com a melhora na conexão e disponibilidade, trouxe grandes oportunidades de negócios para as organizações, oferecendo um acesso mais amplo a informações, serviços e recursos (LUCIANO, 2003).

Conforme o número de usuários da internet aumenta, é necessária adaptação interna e externa das empresas, para que elas possam aumentar a competitividade, criar soluções para conflitos ou que permitam um uso eficiente do *marketing* na internet para que esteja apto a confrontar desafios e propagar seus serviços com qualidade. As empresas estão adotando a internet como solução para o desenvolvimento do seu negócio ou a venda do seu produto, permitindo uma expansão ampla e atingindo seus clientes de forma rápida e objetiva (LUCIANO, 2003).

O aumento do uso da internet independe de classe econômica, pois até nas classes mais baixas o uso de internet está crescendo, como pode ser observado na Figura 1 que apresenta o percentual de pessoas que utilizaram a internet.

Figura 1 - Percentual de pessoas que utilizaram a internet



Fonte: IBGE, Diretoria de Pesquisas, Coordenação de Trabalho e Rendimento, Pesquisa Nacional por Amostra de Domicílios 2013-2015.

Nota: Excluídas as pessoas cuja condição na unidade domiciliar era pensionista, empregado doméstico ou parente do empregado doméstico.

(1) Inclusive as pessoas moradoras em unidades domiciliares cujos componentes recebiam somente em benefícios.

Fonte: IBGE, Diretoria de Pesquisas, Coordenação de Trabalho e Rendimento, Pesquisa Nacional por Amostra de Domicílios 2013 – 2015.

Conforme o número de usuários aumenta, o crescimento do *e-commerce* e suas vendas também evolui, são mais de 400% em um pouco mais de 10 anos. segundo a análise da GS&MD (2016), como pode ser observado na Figura 2 que apresenta dados de crescimento do comércio eletrônico.

Figura 2 - Crescimento do Comércio Eletrônico



Fonte: Análise - GS&MD (2016).

A tendência desse crescimento nas vendas e faturamento do *e-commerce* só aumenta e devido não haver limite físico para agregar mais lojas, é possível abrir lojas virtuais sem limite.

2.1.1 O Perfil do Consumidor

Quando o assunto é o consumidor, trata-se de todas as pessoas, pois todos consomem direta ou indiretamente. O perfil do consumidor online pode ser influenciado por inúmeros fatores (social, psicológico e cultural), destacando-se os fatores culturais, que são desenvolvidos e estimulados desde a infância, socializados e impostos através da classe social que esse indivíduo se encontra (SOUZA; FARIAS; NICOLUCI, 2005). Os autores também ressaltam que é possível analisar os perfis de consumidores e separá-los por classe, porém, existem vários critérios que os distinguem, como o seu estilo de vida, personalidade, circunstâncias econômicas, idade e autoimagem.

Vale ressaltar que também são influenciadores do comportamento do consumidor alguns fatores sociais (família, trabalho, grupos de referência e *status*), segundo Souza, Farias

e Nicoluci (2005). É possível adicionar também que esses perfis, muitas vezes, podem ser influenciados pelas comunidades virtuais, fóruns *online* e/ou salas de bate-papo que utilizam.

De acordo com Karsaklian (2004) a descoberta de uma necessidade, pessoal ou física, geralmente leva o indivíduo a uma situação de desconforto ou infelicidade, fazendo com que ele tome ação para suprir seu estado de equilíbrio psicológico ou de satisfação.

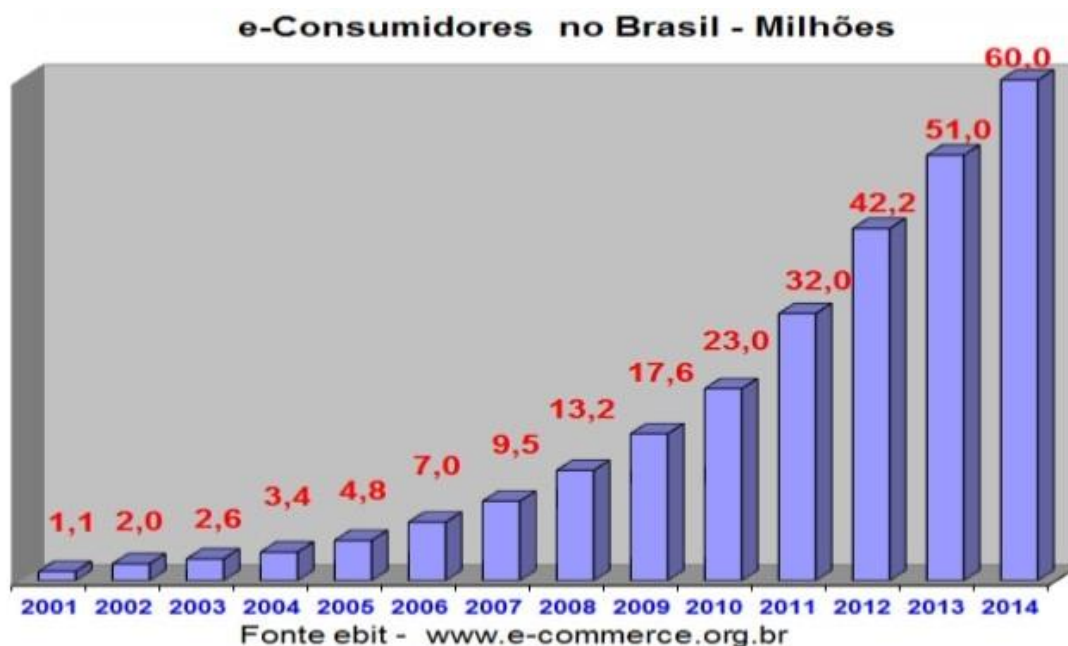
Nakagawa (2008) indica a comodidade (praticidade e acessibilidade e conveniência), a amplitude da seleção e escolha global, competitividade de preços, economia de tempo e volume de informações como principais motivadores potenciais para adoção do comércio eletrônico é a individualidade.

Já Arruda e Miranda (2003) sugerem que comodidade de comprar sem deslocamento e ausência de contato com vendedores e multidões são preponderantes para a decisão de compra virtual.

Por outro lado, Childers, (2001) cita motivações para o consumidor utilizar lojas *online* que define como dimensões utilitaristas, hedonistas e apego. A motivação hedonista é a busca pelo prazer dentro do comércio eletrônico, fazendo do ato da compra um momento de descontração e diversão ou sensação de bem-estar. A motivação utilitarista é o caráter facilitador que o comércio eletrônico oferece consequentemente se tornando útil em determinadas situações e proporciona um conforto maior do que se dirigir a lojas físicas para realizar a compra. Já o apego, acontece devido ao consumidor ter um gosto pela marca, produto ou loja que ele frequenta, gerando boas sensações ao comprar e bons sentimentos quando o indivíduo retorna a comprar com determinado vendedor.

Conforme o número de usuários da internet aumentou, é identificado que o número de compradores *online* cresce também. Com o passar dos anos, os usuários foram se sentindo mais seguros e confortáveis para realizar compras *online*, conforme mostram os dados apresentados na Figura 3.

Figura 3 - Evolução de E-Consumidores no Brasil



Fonte: Ebit.

Segundo Hortinha (2002) existem quatro critérios importantes para segmentar o mercado dos consumidores, esses critérios levam alguns fatores específicos em consideração como:

- Critérios geográficos, demográficos, sociais e econômicos:
 - Geográficos – tipos de região (clima, relevo)
 - Demográficos – gênero, idade e dimensão da família
 - Sociais e Econômicos – rendimento, nível de educação e religião
- Critérios de personalidade e estilo de vida: atividades, interesses, valores.
 - Critérios de comportamento relativamente ao produto/ serviço refere-se ao grau de utilização, lealdade à marca.
 - Critérios de atitude psicológica relativamente ao produto/ serviço; mesma descrição do comportamental

Porém, Modahl (2000) acredita que esses consumidores devem ser estudados com outros aspectos envolvendo segmentos demográficos e psicográficos combinados. Tendo essa combinação ressaltam-se três possíveis dimensões:

- Renda: alta ou baixa;
- Postura diante da tecnologia: otimista ou pessimista;
- Motivação principal: carreira, família ou entretenimento.

Com essas informações, obtém-se subgrupos de consumidores sendo que é possível agrupar por diferentes perfis, sendo eles: consumidores que possuem maior renda e mais otimismo em relação à tecnologia são chamados de adotantes iniciais; consumidores que utilizam a rede com objetivo principal da carreira são os acelerados e importantes; e consumidores que possuem alta renda, otimismo e motivam-se pelas necessidades, familiares, amigos, são os seguidores da nova era (CATALANI, 2006).

2.1.2 Os Riscos do Consumidor

Atualmente, com o avanço da tecnologia e a Internet em constante evolução, em que milhares de computadores estão conectados e compartilhando milhões de informações simultaneamente, é preciso lembrar e reforçar a segurança na conexão. Existem diversos riscos no uso da Internet, ou ao realizar compras *online* (WONGTSCHOWSKI, 2011):

- Falsificação – ataques virtuais que caracterizam a adulteração das informações ou criação de loja de fachada, sendo uma página que é uma cópia de uma loja autêntica. O usuário acha que está comprando na loja verdadeira, quando na realidade está informando seus dados em um site falsificado que tem a intenção de coletar os dados dos clientes sem levantar suspeitas.
- Espreitar – esses ataques na Internet geram o roubo dos dados da conta de um determinado usuário, número de contas bancárias do cliente, dados do saldo e extrato do consumidor ou até mesmo o número do cartão de crédito.
- Espionagem de senhas - essa prática maléfica tem o foco de ter a permissão de acesso às informações do usuário armazenadas, para que seja possível acessar os dados do usuário através da sua própria conta.
- Alteração de dados – faz modificações das informações de certas transações econômicas.
- Repúdio – é o repúdio de negociações que ocasionam em grandes problemas com sistemas de faturamento e aceitação do processamento da transação como os bancos, por exemplo.

Entre os exemplos citados, existem outras ameaças virtuais que afetam a segurança dos usuários, são aquelas que precisam ser executadas pelo cliente como:

- Vírus – são códigos replicados anexando cópias nos executáveis existentes, na maioria dos casos estão camuflados. O vírus será executado quando o usuário utilizar o programa infectado.

- *Trojan Horse* – quando um programa é executado ele realiza uma determinada tarefa, o Cavalo de Tróia está junto com o programa e adiciona funções indesejáveis.
- *Worm* – é um programa auto replicante, criando vários clones na sua execução, não tendo a intervenção do usuário (ALBERTIN, 2000, p.175).

Existem muitos ataques na internet, mas alguns deles têm o foco maior no consumidor segundo (ALBERTIN, 2000, p.176), dentre eles são:

- Roubo de dados - maior intenção é roubar dados críticos de autenticação do cliente como senha, *login* e até mesmo a sessão. Com esses dados disponíveis, é possível utilizá-los para realizar a autenticação, se passando pelo consumidor e tendo acesso a sua conta. Esse tipo de roubo pode ser concretizado pela captura de teclado, tudo que o usuário digitar no teclado será capturado e armazenado. Esse tipo de roubo de dados é feito por meio de telas ou e-mails falsos.
- Roubo de sessão - o ataque realizado após o cliente ter feito a autenticação, é utilizado o número de sessão do cliente para efetuar transações *online* pela sua conta.
- Alteração de transações – essa ameaça ocorre *online* e utiliza o computador do cliente. Esse tipo de ataque pode ser um programa adulterador de informação e geralmente é instalado na máquina do usuário, ficando em *stand-by* até que o consumidor realize alguma operação *online*. Quando o consumidor realizar a autenticação, o aplicativo vai alterar os dados da transação e transmitir esse pacote modificado.
- *Main-in-the-middle* – consiste em interceptar informações na hora que os dados estão sendo transmitindo entre o cliente e o servidor. O ataque tem acesso às informações trocadas entre o servidor e o cliente e pode realizar alterações, leitura de dados e inclusão de informações. É muito utilizado em fraudes pela Internet, ao interceptar pacotes de dados e fazer alguma modificação das informações importantes trocadas pelo cliente e servidor. Os ataques virtuais têm como objetivo final cometer fraudes virtuais a favor do autor do ataque ou com o objetivo de prejudicar o usuário.

Em específico, o roubo de dados é feito com a captura dos dados críticos do cliente como senhas e *login*, para que futuramente seja replicada a autenticação no sistema. Muitas

vezes os ataques virtuais realizados são barrados por mecanismos de segurança, porém, alguns ataques virtuais são bem-sucedidos. Segundo Albertin (2000), ataques virtuais bem-sucedidos têm a seguinte estrutura:

- Instalação – para alguns tipos de ataque, é necessário primeiramente que uma instalação seja feita, ao instalar um software malicioso é possível alterar de alguma maneira o computador do cliente a favor do ataque realizado.
- Captura de dados – tendo a instalação realizada e seus itens maliciosos hospedados, é possível ter acesso aos dados críticos do cliente.
- Envio dos dados roubados – nesse momento que os dados capturados são enviados para que possam ser acessados por terceiros.
- Fraude – com os dados do cliente disponível, realizar ações fraudulentas se torna fácil.

Para Albertin (2000), é possível descrever vários tipos diferentes de ataque como, por exemplo:

- Ataque completo – consiste em todo o ataque, desde a hospedagem necessária do ataque, a captura de dados e a modificação ou fraude realizada. Esse ataque não depende de outras formas de ataque como auxílio.
- Ataque parcial – esse ataque possui uma determinada dependência, pois sozinho não consegue realizar a fraude, necessitando ataques em conjunto para alcançar seu objetivo.

Albertin (2000) também afirma que existem diversos tipos de locais nos quais é plausível de um ataque virtual, sendo eles:

- Aplicativos de sistema – os aplicativos de sistema mais utilizados e comuns pelos usuários são os que possuem o histórico maior de ataque, como o e-mail ou navegador web, por exemplo.
- Sistema operacional – o sistema operacional pode sofrer alteração no seu código ou no serviço do sistema e também pode sofrer alteração em suas *Dynamic-Link Libraries (DLL)*.
- Núcleo do sistema - o sistema operacional depende de *drivers* e periféricos e pode ser sofrer ataques irreversíveis.

Para o sucesso de um determinado ataque segundo Albertin (2000), é preciso identificar as vulnerabilidades do sistema que consistem em:

- Falha na implementação – o sistema operacional que possui uma falha ou brecha que possa ser utilizada para o ataque.
- Defeito de segurança – sistema possui uma brecha na segurança, ou tem a arquitetura mal desenvolvida e fica vulnerável a um possível ataque.
- Aplicativos – os aplicativos utilizados pelo usuário podem estar infectados, realizando um ataque sem que o usuário perceba.
- Falta de experiência do usuário – usuários inexperientes com conhecimento baixo na usabilidade da aplicação são mais suscetíveis a serem atacados por usuários com mais experiência. Exemplo: e-mails falsos se passando por bancos, cobranças entre outros possuem *links* com programas maliciosos.

A respeito dos ataques, tem-se uma classificação que condiz com a dificuldade de implementação (ALBERTIN, 2000). Pode-se destacar como uma escala de dificuldade técnica os seguintes itens:

- Ataque básico – é utilizado por meios já conhecidos. Não depende de muita experiência e nem o desenvolvimento de códigos de programação, pois na maioria das vezes são utilizados programas prontos com esse objetivo. O ataque é feito geralmente pela *web* e o programa utilizado pode ser modificado a favor de quem está atacando.
- Ataque médio – esse ataque depende de mais conhecimento técnico, possui mecanismos internos do sistema operacional para que modifique funções e serviços em seu favor. Não depende de grandes trabalhos de programação.
- Ataque de alta complexidade – depende de um conhecimento técnico avançado e extenso do trabalho de programação. Conhecimento em detalhes do sistema operacional é fundamental para quem está atacando, pois facilita a identificação de brechas ou falhas de segurança. Alguns ataques utilizam programas ocultos para obter informações inseridas por meio do teclado do usuário e outros tipos de programa mostram janelas para o usuário pedindo dados confidenciais, muitas vezes se passando por um banco ou uma aplicação que o usuário seja acostumado a utilizar.

Nesses casos de ataque pode-se classificar o qual o tipo de transparência dos ataques (ALBERTIN, 2000), sendo eles:

- Baixa transparência – o ataque altera muito a forma normal da aplicação original e é fácil de identificar que ela não é segura.

- Média transparência – o ataque altera levemente a forma original da aplicação sendo um pouco mais difícil de identificar que ela não é segura.
- Alta transparência – o ataque modifica muito pouco ou quase nada o programa original, sendo praticamente imperceptível a diferença com a aplicação original e quase não é possível identificar que a aplicação não é segura.

É possível destacar alguns dos ataques mais complexos e utilizados segundo Albertin (2000), sendo eles:

- *Trojans* – são programas com uma alta complexidade que se passa por um programa benigno, porém não são. Abusa da falta de conhecimento, curiosidade ou a falta de informação do usuário. Quando um trojan é instalado no computador do usuário, pode ter o controle total do sistema, monitorando a maioria das atividades que estão sendo executadas. Na maioria dos casos, os *Trojan* são instalados por meio de e-mails do usuário, na qual o ataque formula um e-mail incentivando a curiosidade do usuário como pagamento de contas, boletos ou até mesmo o ganho de uma loteria ou sorteio e induz o cliente a instalar o *Trojan* em seu computador.
- Vulnerabilidades – é a instalação de *software* de maneira ilícita no computador do usuário e na maioria dos casos é imperceptível. Como os *Worms Sasser* e *Blaster*, usando a vulnerabilidades para instalar *trojan* na máquina do cliente.
- *Links* falsos – é bem comum o envio de e-mails contendo informações falsas e um *link* direcionado ao ataque e enviado junto ao e-mail. Alguns desses e-mails têm nome de instituições bem conhecidas e podem ter um site falso muito parecido com o original, induzindo o usuário a navegar dentro da aplicação falsa. Muitas vezes, o usuário começa a navegar nessas falsas aplicações e informar seu *login* e senha, esses dados são armazenados a favor de quem está atacando.
- *Keyloggers* – são programas que na maioria das vezes não estão visíveis para os usuários e tem a função de armazenar tudo que for digitado no teclado pelo usuário. Um uso comum é nas aplicações de Internet Banking, pois quando o usuário for informar seus dados bancários, o *Keylogger* armazena o que foi inserido.

A captura de dados é comum quando o assunto é vulnerabilidade de aplicação ou baixa segurança de dados (ALBERTIN, 2000), os tipos de captura de dados mais comuns podem ser:

- Telas falsas – possuem a aparência muito semelhante do site original, quanto mais parecido e mais próximo da perfeição, mais fácil de enganar os usuários. Conforme os usuários inserem seus dados eles são armazenados para que possam ser utilizados por terceiros.
- *Hooks* (ganchos) – utiliza o sistema operacional com o objetivo de capturar dados inseridos pelo teclado ou mouse, mas depende da hospedagem no sistema primeiramente. Trabalha com áreas específicas dentro do tratamento dentro do *Windows*, onde um aplicativo é instalado na sub-rotina para observar o tráfego das mensagens no sistema e na maioria das vezes não é percebido pelo usuário.

Ataque redirecionado – o ataque afeta diretamente o endereço IP dos computadores ou de um determinado servidor de resoluções de nomes *Domain Name System* (DNS).

Ataques como esses podem ser classificados da seguinte maneira segundo Albertin (2000):

- Ataque com o foco no servidor de DNS - quando o ataque acessa um determinado servidor DNS, explora suas vulnerabilidades do servidor. Esse tipo de ataque faz alterações nas configurações específicas do servidor ou modifica as tabelas dos nomes próprios do servidor para um endereço IP desejável.

Ataques que instalam um trojan localmente no computador do usuário, geralmente são adicionados vários arquivos no sistema do usuário e impede que o sistema verifique o servidor DNS para conseguir IP correto. Com isso, o sistema irá redirecionar para o IP desejável e não consultará o servidor DNS como forma de verificação.

A troca de *Uniform Resource Locator* (URL) no navegador do cliente feita através de trojan que está oculto no computador, monitorando todas URLs acessadas. Ao acessar URL considerada perigosa pelo seu ataque, o trojan redireciona para um endereço local instalado na máquina do usuário, obrigando o usuário a ir para a aplicação indesejada. Sem perceber essa alteração o cliente irá informar seus dados confidenciais na página falsa, então novamente redirecionado para página original e não percebe que suas informações foram roubadas (WONGTSCHOWSKI, 2011).

Existe também o ataque à máquina virtual Java, sendo que o ataque acontece abrindo a biblioteca *run-time* do Java que possui a classe *Security Manager*. Faz a descompilação e a modificação da classe *Security Manager*, retirando algumas validações deixando a vulnerabilidade maior. É feito a recompilação da classe e novamente compactada com biblioteca *run-time* do Java. Troca-se a biblioteca original pela fraudada, diminuindo a segurança da aplicação. Esse tipo de ataque é muito realizado a fim de capturar os dados

críticos do cliente, muitos sites de Internet *banking* e de comércio eletrônico usam autenticação (WONGTSCHOWSKI, 2011).

2.1.3 O Valor do Mercado Digital

Para analisar o valor do mercado digital, é preciso compará-lo com as formas de transações comerciais tradicionais e verificar como ele pode transformá-las. As transações tradicionais são conhecidas por muitos e é realizada de maneira simples, direta e na maioria dos casos presencial, tendo uma limitação considerável, pois há um impedimento que não permite realizar várias transações ou vendas simultaneamente. Na venda tradicional, existem dois envolvidos na maioria dos casos, já em um mercado digital, é possível ter apenas um vendedor negociando com milhares de pessoas simultaneamente (BLOCH; PIGNEUR; SEGEV, 1996).

O mercado digital não se restringe unicamente à venda direta de informações, serviços e produtos, mas sim uma visão mais ampla do impacto potencial sobre os negócios que a utilização comercial da web pode oferecer a uma organização. Segundo Bloch, Pigneur e Segev (1996) apesar de a venda direta ser uma das formas de se obter lucro, o mercado digital pode promover uma marca, agregar valor à empresa e expandir o potencial da empresa nacionalmente ou internacionalmente muito mais rápido do que a forma tradicional. Aumentar o número de clientes com o uso do comércio eletrônico está agregando cada vez mais valor dentro do mercado.

A visão do mercado digital, comparado a qualquer tipo de transação comercial por meio do uso de infraestrutura digital, tem a vantagem de unir uma gama variada de ferramentas na utilização da Web para favorecer ou incentivar transações comerciais.

Por fim, o mercado digital funciona como instrumento de promoção (pré-venda), sendo um canal de vendas ou de atendimento ao cliente (pós-venda). Gera economia na realização de transações e redução do ciclo de desenvolvimento dos produtos. Ter uma implementação do uso do mercado digital em uma empresa pode promover um aprendizado organizacional e tecnológico indispensáveis para a aplicação efetiva e pode alavancar o número de vendas ou clientes atingidos com um objetivo comercial estratégico. Contudo, afirmam-se Bloch, Pigneur e Segev (1996) que o mercado digital ainda proporciona o desenvolvimento de novos produtos ou novos modelos de negócio.

3 MATERIAIS E MÉTODO

A seguir estão os materiais e o método utilizados para a modelagem e a implementação do sistema obtido sendo o resultado desse trabalho.

3.1 Materiais

No desenvolvimento deste projeto, é utilizada a linguagem Java com o *framework* Spring para a programação do lado servidor do *e-commerce*. A *HyperText Markup Language* (HTML5), as *Cascading Style Sheets* (CSS3), a linguagem *TypeScript* e o *framework* Angular foram utilizados para a programação do lado cliente da aplicação e o Postgres foi utilizado como banco de dados. O Quadro 1 apresenta as tecnologias e as ferramentas utilizadas para o desenvolvimento deste trabalho.

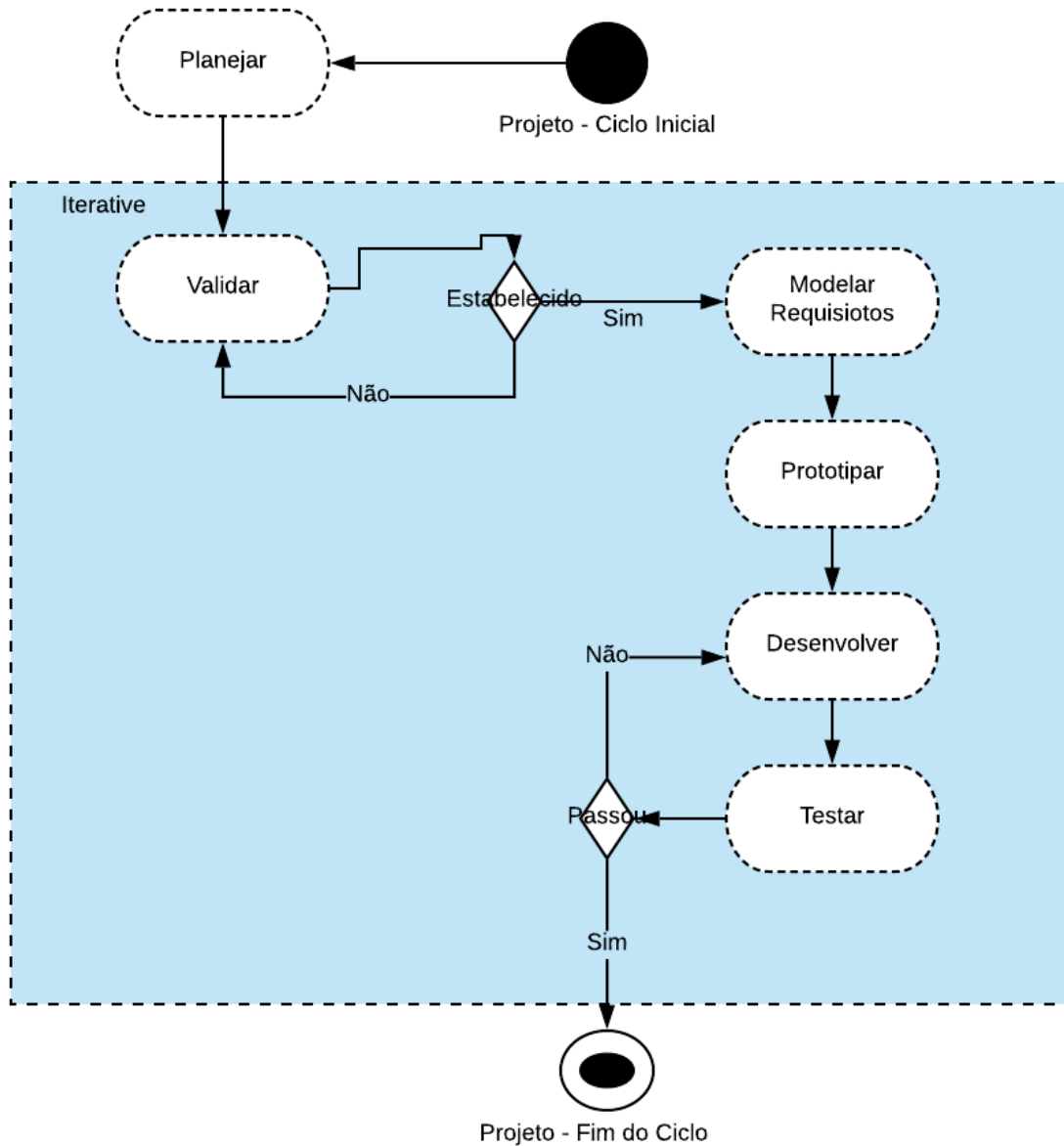
Ferramenta / Tecnologia	Versão	Finalidade
Java	1.8	Linguagem de Programação
<i>Spring</i>	2.0.1	Framework para o lado servidor
HTML	5	Modelagem do Sistema
CSS	3	Estilização do Sistema
<i>Angular</i>	7	Framework para o lado cliente
<i>TypeScript</i>	2.0.7	Linguagem para o lado cliente
<i>Postgres</i>	10.3	Gerenciador de Banco de Dados

Quadro 1 - Lista de ferramentas e tecnologias

3.2 Método

O método consiste nas atividades de levantamento de requisitos, modelagem, avaliação e implementação de funcionalidades. No desenvolvimento deste trabalho, essas atividades foram baseadas em um ciclo de planejamento iterativo de análise de uma forma simples, prática com a prototipação e a implementação de requisitos. A Figura 4 ilustra as atividades desse ciclo e a indicação de iteração de atividades.

Figura 4 - Ciclo de Vida do Desenvolvimento do Projeto



Fonte: autoria própria

Os requisitos identificados durante a prototipagem do projeto são organizados em funcionais e não funcionais e separados por relevância. Os requisitos funcionais são organizados na forma de casos de uso. Dos requisitos são identificadas as unidades de persistência do sistema (as tabelas do banco de dados) e definido o diagrama de entidades e relacionamentos.

No planejamento, uma determinada implementação ou código foi planejada, verificado o tempo que será necessário para o seu desenvolvimento e as ferramentas necessárias para essa implementação.

Na etapa de validação foi analisado se todo o planejamento foi útil e se agregou valor para o desenvolvimento do projeto.

Na modelagem dos requisitos foram desenvolvidos os processos necessários e o melhor método para produção do requisito. Realizar os protótipos de um item ou mais, consistiu em fazer todo seu molde e entregar funcionamentos do tipo *Minimum Viable Product* (MVP), após a criação desses requisitos os mesmos passaram por uma fase de testes garantindo seu funcionamento.

Na etapa de desenvolvimento, foram unidos os resultados de cada etapa e produzidas as funcionalidades necessárias para o correto funcionamento do sistema. Os itens desenvolvidos que não passaram na etapa de testes, retornaram para o desenvolvimento, passaram pelos ajustes necessários e novos testes foram realizados a fim de certificar as funcionalidades.

4 RESULTADOS

Este capítulo apresenta o resultado da realização deste trabalho que é o desenvolvimento de *e-commerce* com seu código *open-source* para realizar vendas de dispositivos eletrônicos.

4.1 Escopo do Sistema

A aplicação desenvolvida gerencia os dados dos produtos disponíveis em estoque. Os produtos podem ser cadastrados e ficarem inativos até segundo momento. Para uso do sistema devem ser informados os dados do comprador utilizados na realização do seu cadastro.

O sistema realiza o gerenciamento dos produtos vendidos, controlando a quantidade disponível em estoque e disponibilizando relatórios de controle para o administrador. Já para o usuário (comprador) é possível acompanhar o rastreamento do pedido, visualizar seu histórico de compras.

Os produtos colocados à venda serão separados em categorias e subcategorias para facilitar a navegação dentro da aplicação *web*. O acesso dos compradores será garantido nos principais navegadores (Chrome, Mozilla, Safari, Edge).

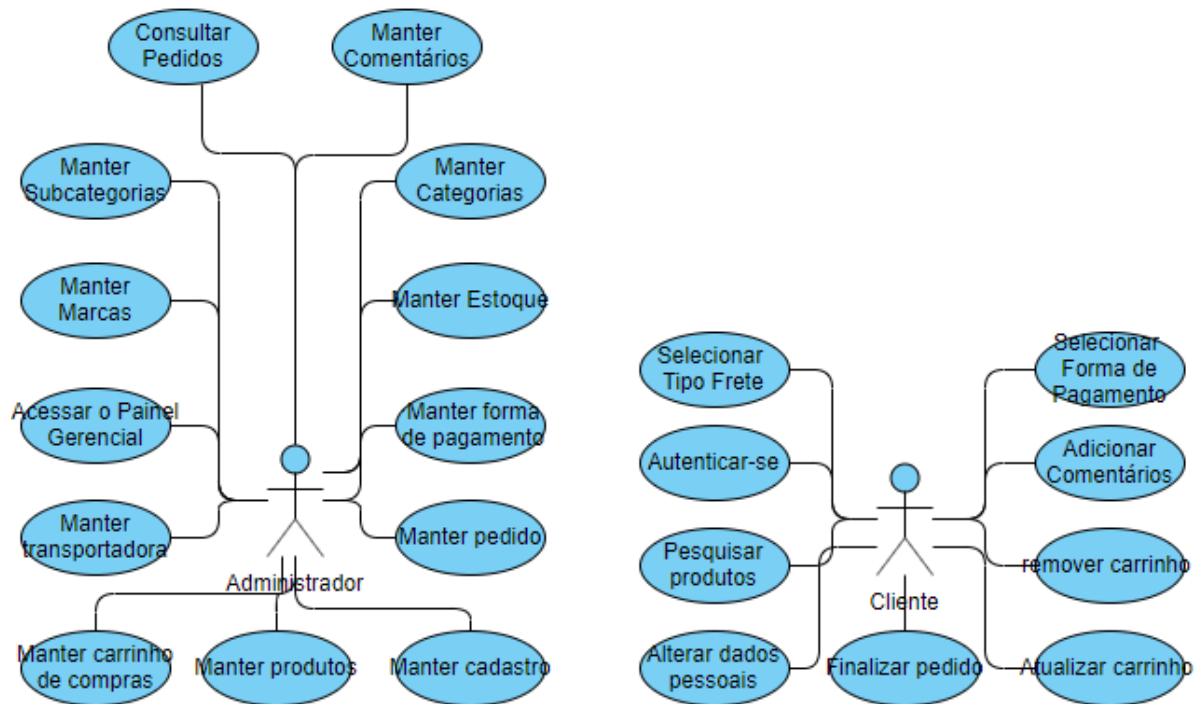
O sistema calcula o custo de entrega baseado no produto escolhido e o cálculo é realizado pelo peso do item. Pode haver promoção do tipo “entrega gratuita” para determinadas localidades ou com um preço mínimo pré-definido.

O sistema permite que apenas os usuários do tipo administrador podem visualizar relatórios dos itens vendidos, pedidos e produtos e entradas e saídas de estoque.

4.2 Modelagem do Sistema

A Figura 5 mostra os casos de usos definidos para o *e-commerce* implementado. É possível realizar a autenticação no sistema com dois tipos de usuários: um é o usuário cliente e o outro é o usuário administrador.

Figura 5 – Diagrama de Casos de Uso



Fonte: autoria própria

Com usuário administrador é possível realizar diversas ações conforme apresentado na Figura 5, algumas delas são:

- Manter produtos
 - Consultar produtos que estão disponíveis.
 - Encontrar produtos que não estão mais em estoque e solicitar uma notificação informando que o produto chegou.
 - Adicionar produtos.
 - Editar produtos.
 - Remover produtos.
- Manter carrinho de compras
 - Remover carrinhos existentes.
 - Editar carrinhos existentes.
 - Remover itens do carrinho.
- Manter cadastro
 - Adicionar endereço.
 - Editar endereço.
 - Editar dados pessoais.

- Manter pedido
 - Aprovar pedido.
 - Alterar endereço do pedido.
 - Cancelar pedido.
 - Editar pedido.
- Manter transportadora
 - Adicionar transportadora.
 - Alterar transportadora.
 - Verificar valor da transportadora.
- Manter forma de pagamento
 - Alterar forma de pagamento.
 - Incluir nova forma de pagamento.
 - Remover forma de pagamento.
 - Inativar forma de pagamento.
- Manter Comentários
 - Alterar comentários.
 - Excluir comentários.
 - Exibir comentários.

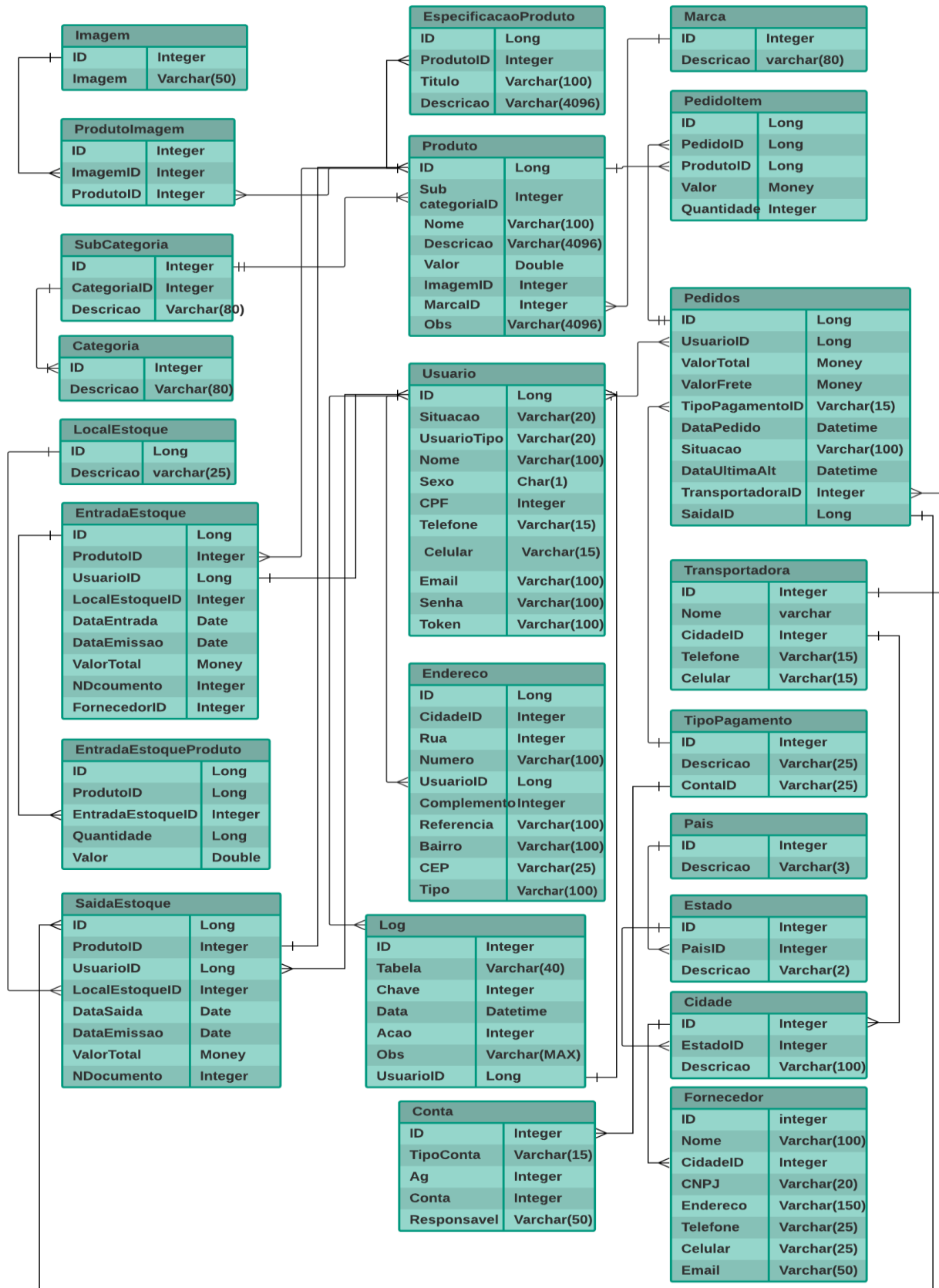
No usuário cliente, suas principais ações são:

- Adicionar Comentários
 - Permite a inclusão de comentários
- Pesquisar produtos
 - Verificar os produtos disponíveis.
- Atualizar carrinho
 - Adicionar itens no carrinho.
 - Remover itens do carrinho.
 - Salvar carrinho.
- Alterar dados pessoais
 - Adicionar endereços.
 - Editar endereço.
 - Remover endereço.
- Finalizar pedido

- Confirmar itens do pedido.
- Escolher forma de pagamento.
- Escolher transportadora.
- Escolher endereço do destino.

Foi utilizado o Postgres como Gerenciador de Banco de Dados para armazenar as informações necessárias para o funcionamento completo da aplicação. Na Figura 6 pode ser visualizado o diagrama de entidade-relacionamento do *e-commerce*.

Figura 6 - Modelagem Relacional



Fonte: autoria própria

Todos os usuários deverão realizar autenticação no sistema, porém, o usuário cliente pode efetuar compras, acompanhar seus pedidos. Caso o usuário cliente não esteja autenticado, é possível visualizar os produtos, realizar buscas e adicionar produtos no carrinho, porém, para finalizar seu pedido é necessário a autenticação no sistema. Já o usuário administrador pode acompanhar os pedidos realizados, incluir novos produtos e visualizar relatórios disponíveis no sistema.

4.3 Requisitos Funcionais e não funcionais

O Quadro 2 apresenta a listagem de Requisitos Funcionais (RF) identificados para a utilização do sistema.

Quadro 2 - Listagem de Requisitos Funcionais

Identificação	Nome	Descrição
RF01	Cadastrar usuário	O usuário pode acessar o sistema e navegar pela aplicação sem possuir cadastro, porém, para realizar compras a autenticação no sistema é obrigatória.
RF02	Manter produto	Caso o usuário do sistema seja administrador, é possível realizar o <i>CRUD</i> de produtos.
RF03	Visualização de relatórios	O usuário administrador terá uma opção para visualizar relatórios dos produtos vendidos, quantidade de usuários cadastrados.
RF04	Adicionar produto ao carrinho	O usuário cadastrado ou não terá a opção de adicionar produtos no carrinho de compras.
RF05	Pesquisar produtos	Será inserido um campo para pesquisa dos produtos em grande parte das telas existentes na aplicação.
RF06	Calcular o valor de frete	Ao inserir o número de Código de Endereçamento Postal (CEP) , o sistema calculará o valor de frete e apresentará opções para o usuário pode escolher. (SEDEX, PAC).
RF07	Calcular o valor de compra	Ao selecionar os produtos, a aplicação calculará o valor total dos produtos adicionado do frete.
RF08	Formas de pagamento	O usuário poderá escolher qual a forma de pagamento desejada para comprar seus produtos.

Fonte: autoria própria

No Quadro 3 são listados os Requisitos Não Funcionais do sistema (RNF). Os requisitos não funcionais explicitam regras de negócio, restrições ao sistema de acesso, requisitos de qualidade, desempenho e segurança, entre outros.

Quadro 3 - Listagem de Requisitos Não Funcionais

RNF01	Permissões de acesso	Usuários que não sejam cadastrados e estejam autenticados não podem realizar compras. Usuários administradores podem configurar produtos a serem vendidos.
RNF02	Armazenamento de Informações para os relatórios	Conforme os usuários fazem cadastro, compram produtos, a aplicação armazenará essas informações para exibi-las em um relatório em tela pelo administrador.
RNF03	Sincronização com o servidor	Os produtos e seus respectivos valores serão sincronizados com o servidor após a inserção.
RNF04	Formular valor total	O sistema calculará o valor dos produtos escolhidos, o valor do frete e possíveis taxas de administração do produto e apresentará para o usuário.
RNF05	Valor promocional dos produtos	Os produtos que estiverem em promoção, receberão uma redução no seu preço final, armazenados no momento da venda.
RNF06	Acesso aos registros.	Usuários não possuem acesso a informações de outros usuários.

Fonte: autoria própria

4.2 Apresentação das telas do sistema cliente

Nesta seção serão apresentadas as funcionalidades da aplicação cliente, sua estrutura e algumas imagens ilustrativas do sistema desenvolvido.

4.2.1 Funcionalidades da Plataforma

As Figuras 7, 8 e 9 têm mostram a página inicial do sistema toda vez que o usuário acessa o site do www.deskshop.com.br digitando na “URL” o endereço correspondente. Nessa tela o cliente pode realizar seu cadastro, procurar quais produtos estão disponíveis na loja, adicionar produtos no carrinho de compras ou autenticar-se no sistema caso já possua cadastro.

Figura 7 – Tela inicial da plataforma



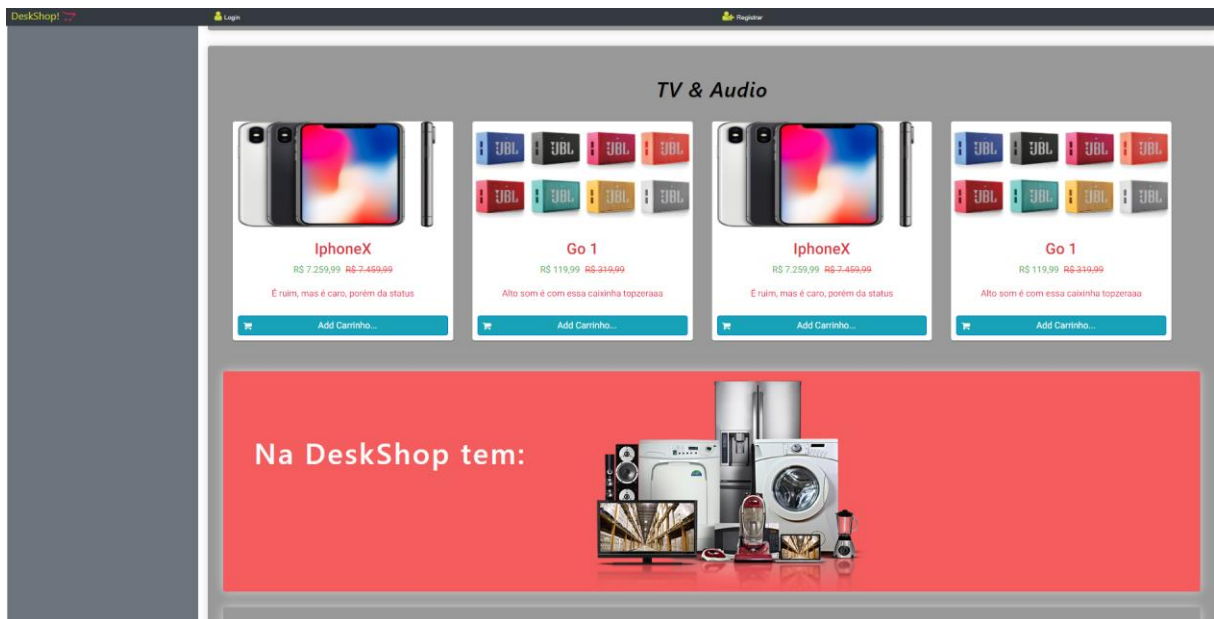
Fonte: autoria própria

Figura 8 - Tela inicial da plataforma com a lista de produtos



Fonte: autoria própria

Figura 9 – Tela inicial da plataforma com a lista de produtos por categoria



Fonte: autoria própria

4.2.1.1 Realizar Cadastro

O Usuário que acessar a plataforma pode criar seu cadastro, clicando no menu “Cadastro” ou no ícone superior identificado como “Registrar-se”. Ao acessar a tela de cadastro, conforme ilustrado pela Figura 10, após preencher corretamente todos os campos o usuário poderá finalizar seu cadastro.

Figura 10 - Tela de Cadastro

The image shows a web registration form titled "Cadastro de Usuário". The form is set against a dark, starry background. At the top right, there is a "Registrar" button with a person icon, highlighted by a yellow box. Below it, in the navigation bar, are "Home" and "Cadastros" links, with "Cadastros" highlighted by a yellow box. The form fields are as follows:

- Primeiro Nome: Lucas
- Segundo Nome: Sendeski
- E-mail: Lucas@Deskshop.com
- Senha
- Confirme sua Senha
- Data de Nascimento: dd/mm/yyyy
- CPF: 123.456.789-10
- Telefone: (41) 3333 - 3333
- Celular: (41) 99999 - 9999
- Dropdown menu: "Selecione seu Sexo" with a downward arrow.
- Checkbox: "Desejo receber Emails com novidades da DeskShop" (unchecked).

At the bottom of the form, there are two buttons: a red "CANCELAR" button and a blue "CONFIRMAR E CADASTRAR NOVO ENDEREÇO" button.

Fonte: autoria própria

Caso o usuário esteja autenticado no sistema, algumas funções são habilitadas, como cadastrar endereços, realizar pedidos, finalizar a autenticação e visualizar os pedidos realizados anteriormente. A Figura 11 ilustra a alteração do menu em comparação com a Figura 7.

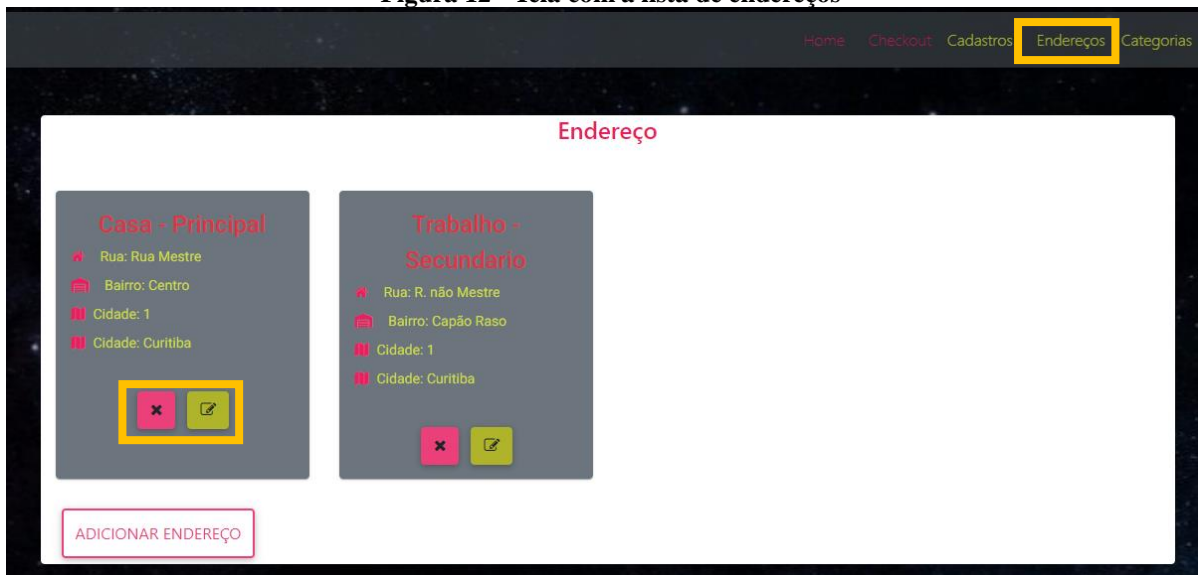
Figura 11 - Tela Inicial após o Login

Fonte: autoria própria

4.2.1.2 Cadastrar Endereço(s)

Após ter seu usuário devidamente cadastrado, é possível cadastrar um ou mais endereços para cada conta criada, as Figuras 12 e 13 ilustram a tela de criação do cadastro, caso o usuário não esteja autenticado, não será possível incluir endereço(s).

Figura 12 - Tela com a lista de endereços



Fonte: autoria própria

Figura 13 - Tela de inclusão/edição de endereços

Endereços

Nome
Casa - Principal

Rua
Rua Mestre

Número
1

Complemento
Nenhum

Referência
Nenhuma

Bairro
Centro

CEP
81310-020

Tipo da Moradia
Casa

Cidade
Curitiba

CANCELAR SALVAR


Fonte: autoria própria

4.2.2 Carrinho de Compras

A tela que exibe o carrinho é apresentada em todas as telas, acima da barra de menus. O ícone do carrinho abre a tela quando clicado. É possível adicionar produtos no carrinho com o usuário autenticado ou não. O carrinho armazena os itens adicionados na sessão do navegador e são excluídos os itens após realizar o pedido ou limpar o cache de navegação. As Figuras 14 e 15 têm como objetivo ilustrar o carrinho com o usuário autenticado no sistema e respectivamente o carrinho com o usuário não autenticado do sistema.

Figura 14 - Carrinho com usuário autenticado

Nº.	Produto	Nome Produto	Quantidade	Preço	Total	Remover
3		MI10	- 4 +	R\$ 3.259,99	R\$ 13039,96	✕

Confirme os Itens Acima!! 

CONTINUAR COMPRANDO ← CHECKOUT \$

Fonte: autoria própria

Figura 15 - Carrinho com usuário não autenticado

Nº.	Produto	Nome Produto	Quantidade	Preço	Total	Remover
3		MI10	- 1 +	R\$ 3.259,99	R\$ 3259,99	✕

Confirme os Itens Acima!! 

CONTINUAR COMPRANDO ←

É preciso estar logado para finalizar suas compras!

Fonte: autoria própria

4.2.3 Checkout

Após verificar os produtos do carrinho e estar autenticado no sistema, é possível ir para tela de *checkout*. Nessa tela é possível escolher o tipo de frete, a forma de pagamento e

respectivamente o endereço de entrega do pedido. O campo valor do frete fica desabilitado e é carregado automaticamente conforme o tipo de frete e endereço são selecionados. O valor do frete é calculado conforme os dados do produto como peso, altura e largura em conjunto do tipo do frete são enviados para a *Application Programming Interface* (API) dos Correios. A API dos Correios retorna o preço do frete e automaticamente atribui o valor no campo “Valor do Frete”. A API do correio também retorna a data de entrega, que é atribuído o valor na tela dos pedidos realizados. A Figura 19 ilustra a tela de *checkout* com todos os campos selecionados.

Figura 16 – Checkout

Carrinho

Seu carrinho de compras contém:

Compras acima de R\$ 300,00 o frete é por NOSSA CONTA!! =D

Nº.	Produto	Nome Produto	Quantidade	Preço	Total
3		M10	4	R\$ 3.259,99	R\$ 13.039,96

Valor Final Total: R\$ 13.061,37

Tipo Frete: Valor Frete: Forma de Pagamento: Endereço de Entrega:

Confirme os Itens Acima!!

VOLTAR PARA O CARRINHO!! ←

FINALIZAR PEDIDO \$

⌚

Fonte: autoria própria

4.2.4 Meus Pedidos

Após realizar o pedido, a tela do *checkout* irá redirecionar para tela “Meus Pedidos” sendo possível visualizar os pedidos realizados, quais os itens dos pedidos realizados e a sua data de entrega. Nas Figuras 17 e 18 são apresentadas as telas dos pedidos realizados e a visualização dos itens de cada pedido.

Figura 17 - Tela dos pedidos realizados

Meus Pedidos

Nº Pedido	Data do Pedido	Data de Entrega	Tipo do Frete	Valor do Frete	Forma de Pagamento	Total	Ações
1	08/07/2019	13/07/2019	40010	R\$ 21,41	Boleto	R\$ 281,40	

[VOLTAR PARA HOME](#)


Fonte: autoria própria

Figura 18 - Tela dos itens do pedido

Pedidos x

1 **Pedido Solicitado**
2 Pedido Visualizado
3 Pedido Aprovado
4 Pedido Postado
5 Pedido a Caminho
6 Pedido Entregue
7 Finalizado e Aprovado

Histórico:

Produto	Nome	Valor	Quantidade	Total
	Quadro de Led	R\$ 259,99	1	R\$ 259,99

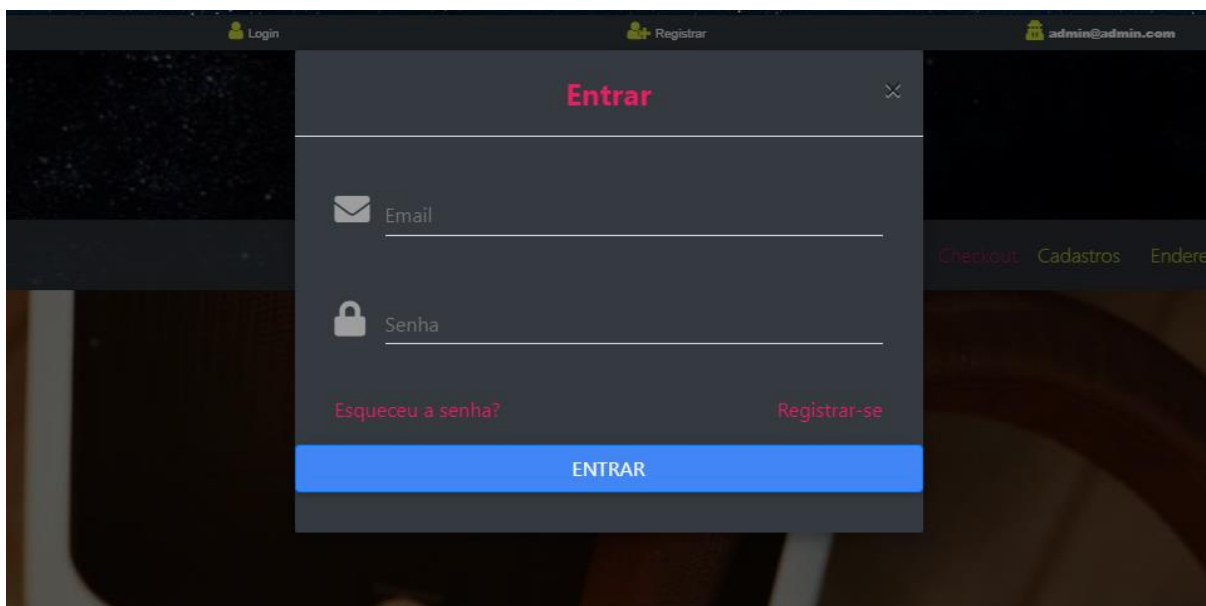
FECHAR

Fonte: autoria própria

4.2.5 Autenticação

A autenticação do sistema é realizada por meio de uma tela modal e pode ser acessada de qualquer tela do sistema. O menu superior contém um ícone para a tela de autenticação e faz parte do *template* da aplicação, sendo possível acessar essa tela apenas clicando no ícone. A tela apresentada solicita o e-mail cadastrado e a senha de acesso e, também possui a funcionalidade de recuperar a senha como pode ser visualizado na Figura 19.

Figura 19 - Modal de Login



Fonte: autoria própria

4.3 Painel de Administração

O painel de administração foi feito em um servidor separado, totalizando três servidores, um para o *back-end* e dois para o *front-end*, sendo um para a loja e outra para o administrador, garantindo a segurança e melhorando a funcionalidade das plataformas. O painel de administração possui as seguintes funcionalidades:

- Local de Estoque
 - Listar locais de estoque.
 - Editar locais de estoque.
 - Incluir locais de estoque.
 - Excluir locais de estoque.
- Entradas Estoque
 - Listar entradas de estoque.
 - Editar entradas de estoque.
 - Incluir entradas de estoque.
 - Excluir entradas de estoque.
- Saídas Estoque
 - Listar saídas de estoque.
 - Editar saídas de estoque.
 - Incluir saídas de estoque.

- Excluir saídas de estoque.
- Produtos
 - Listar produtos.
 - Editar produtos.
 - Incluir produtos.
 - Excluir produtos.
- Categorias
 - Listar categorias.
 - Editar categorias.
 - Incluir categorias.
 - Excluir categorias.
- Pedidos
 - Listar pedidos.
 - Editar pedidos.
 - Incluir pedidos.
 - Excluir pedidos.
- Subcategorias
 - Listar subcategorias.
 - Editar subcategoria.
 - Incluir subcategoria.
 - Excluir subcategoria.
- Marcas
 - Listar marcas.
 - Editar marca.
 - Incluir marca.
 - Excluir marca.
- Usuários
 - Listar usuários.
 - Editar usuário.
 - Incluir usuário.
 - Excluir usuário.
- Endereços
 - Listar endereços.

- Editar endereço.
- Incluir endereço.
- Excluir endereço.

O painel de administração é acessado somente pelo usuário que possui a permissão de administrador. A permissão é atribuída somente ao utilizar o painel de administração pelo usuário com perfil admin. A Figura 20 ilustra a tela inicial do painel e seus respectivos menus. As Figuras 21 e 22 ilustram a tela de Pedidos e seus itens. Todas as telas do painel citado seguem o mesmo padrão.

Figura 20 - Painel de administração



Fonte: autoria própria

Figura 21 - Tela de Pedidos

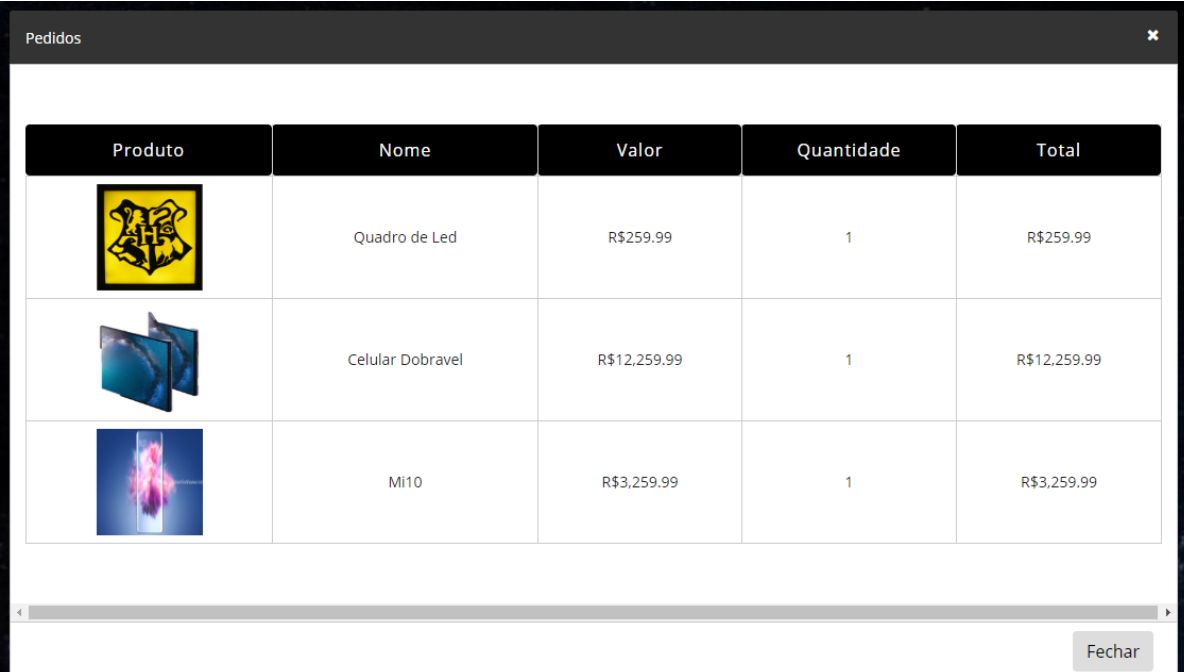
Lista de Pedidos									
Código	Usuário	Endereço	Valor Total	Valor do Frete	Data do Pedido	Data de Entrega	Tipo do Pagamento	Etapa	Ações
1	admin	Rua Mestre	R\$281.40	R\$21.41	08/07/2019	13/07/2019	Boleto	0	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>




Buscar por código...

1 20

Fonte: autoria própria

Figura 22 - Tela de pedidos exibindo os itens



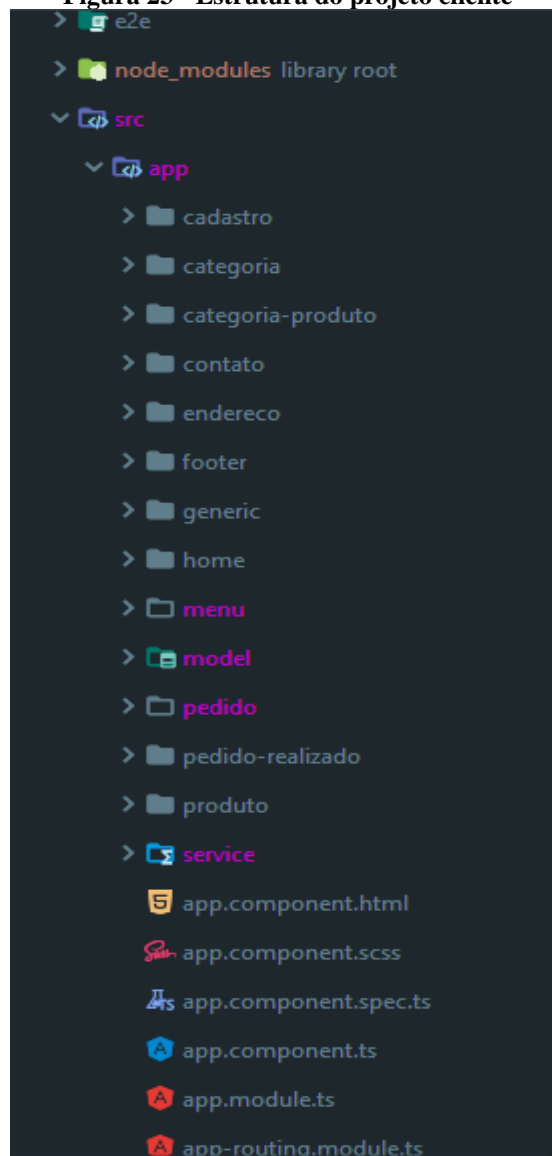
Produto	Nome	Valor	Quantidade	Total
	Quadro de Led	R\$259.99	1	R\$259.99
	Celular Dobravel	R\$12,259.99	1	R\$12,259.99
	Mi10	R\$3,259.99	1	R\$3,259.99

Fonte: autoria própria

4.4 Apresentação do código do sistema cliente

O projeto foi organizado conforme a Figura 23. Cada pasta dentro do diretório `/src/app` representa um componente criado e as pastas “`node_modules`”, “`model`”, “`generic`” e “`service`” contém respectivamente:

- Model – modelagem do sistema igual a modelagem criada no servidor.
- Generic – possui a classe “`CrudService`” com os métodos padrões do sistema como `delete()`, `save()`, `findAll()` e `findOne()`.
- Service – possui todas as classes do tipo *service* que faz a comunicação com o servidor.
- Node_modules – possui todos os pacotes e dependências para o funcionamento do projeto.

Figura 23 - Estrutura do projeto cliente

Fonte: autoria própria

4.4.1 Adicionar e remover produtos do carrinho

Conforme apresentado no Capítulo 4.2.2 o carrinho de compras possui funções de adicionar produtos, remover produtos e limpar o carrinho, nas Listagens 1 e 2 são exibidos o código de implementação para adicionar e remover itens do carrinho. Na Listagem 1 o método `add()` recebe como parâmetros o produto selecionado e adiciona no carrinho que é criado no `localStorage`¹. Mesmo que o usuário saia da loja e volte outro momento, se as informações do `localStorage` ainda estiverem salvas, esse carrinho continua existindo.

¹ `LocalStorage` – armazenamento de dados no lado cliente. Salva determinadas informações no cache do navegador.

Listagem 1 - Código de adicionar produto

```

add(produto: Produto, quantidade: number) {
  const pedidoItem = new PedidoItem();
  pedidoItem.produto = produto;
  pedidoItem.quantidade = quantidade;

  this.carrinho = JSON.parse(localStorage.getItem( key: 'carrinho'));
  if (!this.carrinho) {
    this.carrinho = [];
    this.carrinho.push(pedidoItem);
  } else {
    let itemCarrinho = this.carrinho.find( predicate: item => item.produto.id === pedidoItem.produto.id);

    if (itemCarrinho) {
      itemCarrinho.quantidade += quantidade;
    } else {
      this.carrinho.push(pedidoItem);
    }
  }
  localStorage.setItem('carrinho', JSON.stringify(this.carrinho));
}

```

Fonte: autoria própria

Na Listagem 2 o método remove() retira um produto específico do carrinho que está salvo no localStorage. O método clear() remove todos os itens adicionados no carrinho e pode ser utilizado pelo usuário manualmente ou quando seu pedido é efetivado.

Listagem 2 - Código de remover produtos

```

remove(pedidoItem: PedidoItem) {
  this.carrinho = JSON.parse(localStorage.getItem( key: 'carrinho'));
  // find
  // tslint:disable-next-line:only-arrow-functions
  const item = this.carrinho.find( predicate: function__(i) {
    return i.produto.id === pedidoItem.produto.id;
  });

  if (item) {
    const index = this.carrinho.indexOf(item);
    if (index > -1) {
      this.carrinho.splice(index, deleteCount: 1);
    }
    localStorage.setItem('carrinho', JSON.stringify(this.carrinho));
  }
}

clear(pedidoItem: PedidoItem[]) {
  pedidoItem.length = 0;
  localStorage.setItem('carrinho', JSON.stringify(pedidoItem));
}

```

Fonte: autoria própria

4.4.2 Salvar o Pedido

O código do *checkout* valida se todos os campos obrigatórios estão preenchidos e, se o valor do pedido for superior a R\$ 300,00 o frete é definido como gratuito. A Listagem 3 ilustra como foi implementado a finalização do pedido.

Listagem 3 - Código de pedido realizado.

```
save() {
  this.progress = true;
  this.total();
  if (this.vlrTotal > 0) {
    const today = new Date();
    today.getDate();
    this.pedidoEdit.tipofrete = this.tipoFrete.value;
    this.pedidoEdit.tipopagamento = this.tipoPagamento.value;
    this.usuarioService.getLoggedUser().subscribe( next: e => {
      this.pedidoEdit.usuario = e;
      this.pedidoEdit.pedidoItens = [];
      this.pedidoItens.forEach( callbackfn: e => {
        e.valor = e.produto.valor;
        const p = new Produto();
        p.id = e.produto.id;
        e.produto = p;
        this.pedidoEdit.pedidoItens.push(e);
      });
      const datePipe = new DatePipe( locale: 'pt');
      this.pedidoEdit.datapedido = datePipe.transform(today, format: 'dd/MM/yyyy');
      this.pedidoEdit.dataentrega = datePipe.transform(today.setDate(today.getDate() + 5), format: 'dd/MM/yyyy');
      this.pedidoEdit.ultimaalteracao = datePipe.transform(today, format: 'dd/MM/yyyy');
      this.pedidoEdit.etapa = 0;
      if (this.vlrTotal >= 300) {
        this.pedidoEdit.valorfrete = 0;
      }
      this.pedidoEdit.valortotal = this.vlrTotal;
      // tslint:disable-next-line:no-shadowed-variable
      this.pedidoService.save(this.pedidoEdit).subscribe( next: e => {
        this.pedidoEdit = new Pedido();
        this.msgs = [{
          severity: 'success',
          summary: 'Confirmado',
          detail: 'IHAA!!! seu Pedido foi salvo com sucesso!'
        }];
      });
    });
  }
}
```

Fonte: autoria própria

O método `save()` valida e reúne todas as informações como: valor total, data do pedido, tipo de pagamento, valor do frete, usuário solicitante e salva essas informações. Caso o valor total do pedido seja maior que R\$ 300,00 o frete relacionado ao pedido se torna grátis.

4.4.3 Autenticação do Usuário

A autenticação é realizada por meio de *token*. No controle tela de autenticação é realizada uma chamada HTTP do tipo POST para o servidor da aplicação com as informações de usuário e senha. No servidor esses dados são verificados e em caso de sucesso o servidor devolve um *token*. Esse *token* deverá ser enviado em todas as próximas requisições para o servidor para que o usuário possa manter-se autenticado.

Os métodos de autenticação e saída do sistema são ilustrados nas Listagens 4 e 5.

Listagem 4 - Código de login

```
login(username: string, password: string): Observable<AccessToken> {
  const params = new URLSearchParams();
  params.append( name: 'username', username);
  params.append( name: 'password', password);
  params.append( name: 'grant_type', value: 'password');

  const headers = new HttpHeaders( headers: {
    'Content-type': 'application/x-www-form-urlencoded',
    Authorization: 'Basic ' + btoa( rawString: `app:utfrpr` )
  });
}
```

Fonte: autoria própria

Listagem 5 - Código de Logout

```
logout() {
  Object.keys(new AccessToken()).forEach( callbackfn: key => localStorage.removeItem(key));
  this.isAuthenticated.next( value: false);
  this.userInfo = null;
  this.router.navigate( commands: ['/home']);
}
```

Fonte: autoria própria

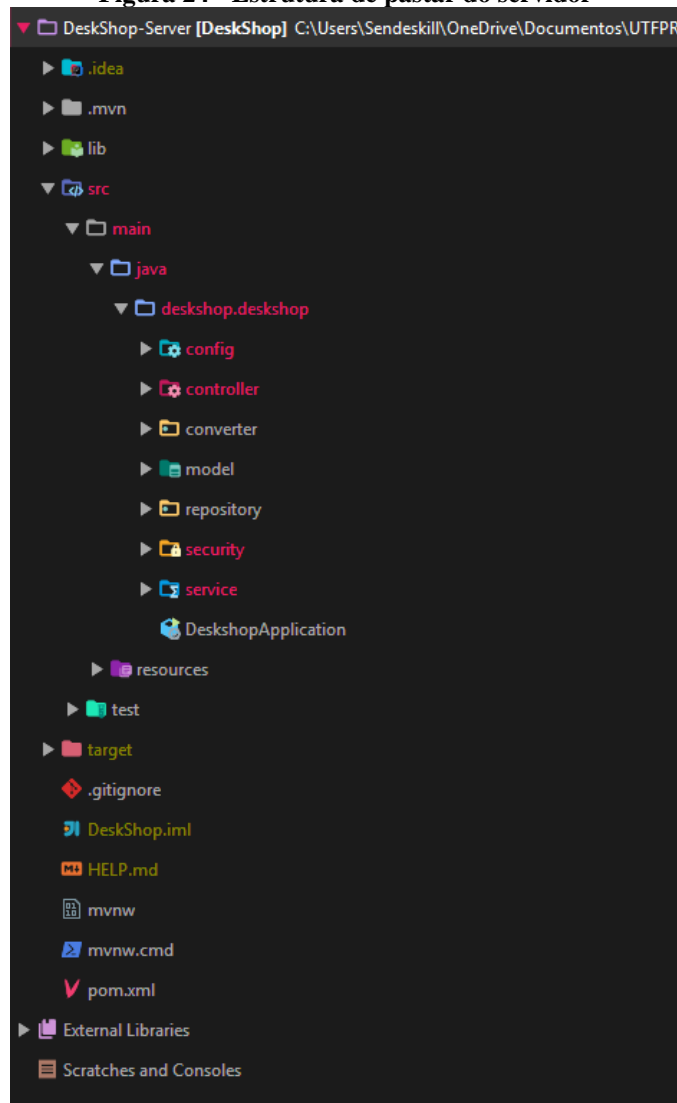
4.5 Apresentação do código do sistema servidor

O projeto foi organizado conforme a Figura 24. Cada pasta organiza as funcionalidades e classes do servidor. As pastas apresentadas têm a seguinte estruturação:

- Config – possui as classes de configurações do servidor, contendo regras de negócio, por exemplo, quais as URLs do servidor necessitam de autenticação e autorização.

- Controller – possui todos os *controllers* criados dentro dessa pasta, tanto para o lado cliente ou administrador. Os *controllers* são responsáveis por atender as requisições HTTP vindas da aplicação cliente.
- Converter – possui classes que fazem conversões necessárias no funcionamento do projeto, como a formatação nas datas utilizadas no sistema.
- Model – possui todas as classes que compõem a modelagem do sistema.
- Repository – organiza todos os repositórios utilizado no sistema.
- Security – possui o controle de autenticação e autorização do sistema.
- Service - possui todas as classes do tipo *servisse*, que são responsáveis comunicação com o banco de dados.

Figura 24 - Estrutura de pastar do servidor



Fonte: autoria própria

4.5.1 CRUD Controller do servidor

Todos os *Controllers* dentro do servidor estendem de uma classe abstrata nomeada de “CrudController” e possui os métodos padrões:

- `FindAll()` – busca todos os registros da classe.
- `FindOne()` – busca um registro da classe por sua chave primária como parâmetro.
- `Save()` – salva um registro recebendo um objeto por parâmetro.
- `Exists()` – verifica se existe um registro realizando a busca por a chave primária como parâmetro.
- `Count()` – conta quantos registros da classe utilizada são existentes.
- `Delete()` – exclui um determinado registro da classe utilizando a chave primária como parâmetro.

A Listagem 6 ilustra a codificação dessa classe abstrata utilizada no servidor do *E-commerce*.

Listagem 6 - Código do controller abstrato

```

package deskshop.deskshop.controller;

import ...

public abstract class CrudController<T, ID extends Serializable> {

    protected abstract CrudService<T, ID> getService();

    @GetMapping
    public List<T> findAll() { return getService().findAll(); }

    @GetMapping("page")
    public Page<T> findAll(@RequestParam int page,
                          @RequestParam int size,
                          @RequestParam(required = false) String order,
                          @RequestParam(required = false) Boolean asc) {
        PageRequest pageRequest = PageRequest.of(page, size);
        if (order != null && asc != null) {
            pageRequest = PageRequest.of(page, size, asc ? Sort.Direction.ASC : Sort.Direction.DESC, order);
        }
        return getService().findAll(pageRequest);
    }

    @GetMapping("/{id}")
    public T findOne(@PathVariable ID id) { return getService().findOne(id); }

    @PostMapping
    public T save(@RequestBody @Valid T entity) {
        return getService().save(entity);
    }

    @GetMapping("exists/{id}")
    public boolean exists(@PathVariable ID id) { return getService().exists(id); }

    @GetMapping("count")
    public long count() { return getService().count(); }

    @DeleteMapping("/{id}")
    public void delete(@PathVariable ID id) { getService().delete(id); }
}

```

Fonte: autoria própria

Na Listagem 7 é apresentada a classe “PedidoController” que possui a anotação `@RestController`, isso significa que esse *controller* poderá receber requisições HTTP. Cada *controller* da aplicação irá receber requisições de uma determinada URL que é passada por parâmetro na anotação `@RequestMapping`. Ou seja, todas as requisições que chegarem no servidor para <http://ip-do-servidor:porta/pedido> serão tratadas pelo PedidoController. Como a classe PedidoController herda as características de “CrudController”, todos os métodos disponibilizados pela classe abstrata podem ser acessados pelo cliente que consumir os dados da API.

Listagem 7 - Código da classe pedidoController

```

package deskshop.deskshop.controller;

import ...

@RestController
@RequestMapping("pedido")
public class PedidoController extends CrudController<Pedido, Long> {

    @Autowired
    private PedidoService pedidoService;

    @Override
    @Valid
    protected CrudService<Pedido, Long> getService() { return pedidoService; }

    @RequestMapping("pedido-user/{id}")
    public List<Pedido> getPedidoUser(@PathVariable Long id) {
        List<Pedido> p = pedidoService.findByUsuarioId(id);
        return p;
    }

    @PostMapping
    @Override
    public Pedido save(@RequestBody @Valid Pedido entity) {
        entity.getPedidoItens().forEach( p -> p.setPedido(entity));
        return getService().save(entity);
    }
}

```

Fonte: autoria própria

4.5.2 CRUD Service do servidor

Todos os *services* dentro do servidor estendem de uma classe do tipo *Interface* nomeada de “CrudService” exibida na Listagem 7 e possui os métodos iguais ao do CrudController exibidos na Listagem 5 contendo alguns métodos a mais como:

- SaveAndFlush() – salva um registro recebendo um objeto por parâmetro e atualiza as informações da classe momentaneamente recebendo um objeto como parâmetro.
- Flush() – realiza uma “descarga” na classe.
- DeleteAll() – exclui todas as informações de uma classe passada como parâmetro.

A interface “Crudservice” é implementada na classe abstrata “CrudServiceImpl” em que todos os métodos existentes são sobrescritos conforme apresentado na Listagem 8.

Listagem 8 - Código do CrudService

```
package deskshop.deskshop.service;

import ...

public interface CrudService<T, ID extends Serializable> {

    List<T> findAll();

    List<T> findAll(Sort sort);

    Page<T> findAll(Pageable pageable);

    T save(T entity);

    T saveAndFlush(T entity);

    Iterable<T> save(Iterable<T> iterable);

    void flush();

    T findOne(ID id);

    boolean exists(ID id);

    long count();

    void delete(ID id);

    void delete(T entity);

    void delete(Iterable<? extends T> iterable);

    void deleteAll();

}
```

Fonte: autoria própria

Listagem 9 - Código do CrudServiceImpl

```

public abstract class CrudServiceImpl<T, ID extends Serializable> implements CrudService<T, ID> {

    protected abstract JpaRepository<T, ID> getRepository();

    @Override
    @Transactional(readOnly = true)
    public List<T> findAll() { return getRepository().findAll(); }

    @Override
    @Transactional(readOnly = true)
    public List<T> findAll(Sort sort) { return getRepository().findAll(sort); }

    @Override
    @Transactional(readOnly = true)
    public Page<T> findAll(Pageable pageable) { return getRepository().findAll(pageable); }

    @Override
    @Transactional()
    public void flush() { getRepository().flush(); }

    @Override
    @Transactional(readOnly = true)
    public T findOne(ID id) { return getRepository().findById(id).orElse(null); }

    @Override
    @Transactional(readOnly = true)
    public boolean exists(ID id) { return getRepository().existsById(id); }

    @Override
    @Transactional(readOnly = true)
    public long count() { return getRepository().count(); }

    @Override
    @Transactional()
    public void delete(ID id) { getRepository().deleteById(id); }

    @Override
    @Transactional
    public void delete(T entity) { getRepository().delete(entity); }

    @Override
    @Transactional()
    public void delete(Iterable<? extends T> iterable) { getRepository().deleteAll(iterable); }

    @Override
    @Transactional()
    public void deleteAll() { getRepository().deleteAll(); }

    @Override
    @Transactional()
    public Iterable<T> save(Iterable<T> iterable) { return getRepository().saveAll(iterable); }

    @Override
    @Transactional()
    public T saveAndFlush(T entity) { return getRepository().saveAndFlush(entity); }
}

```

Fonte: autoria própria

A Listagem 10 exibe a classe “PedidoServiceImpl” responsável pelo *service* utilizado na classe pedido. É utilizada a anotação `@Service` que declara que a classe é do tipo *service* e a classe herda as características da classe “CrudServiceImpl” tendo acesso a todos seus métodos.

Listagem 10 - Código da classe PedidoServiceImpl

```
package deskshop.deskshop.service.impl;

import ...

@Service
public class PedidoServiceImpl extends CrudServiceImpl<Pedido, Long> implements PedidoService {

    @Autowired
    private PedidoRepository pedidoRepository;

    @Override
    protected JpaRepository<Pedido, Long> getRepository() { return pedidoRepository; }

    @Transactional(readOnly = true)
    public List<Pedido> findByUsuarioId(Long id) { return pedidoRepository.findByUsuarioId(id); }

    @Override
    public Page<Pedido> findById(Long id, Pageable pageable) {
        return pedidoRepository.findById(id, pageable);
    }
}
```

Fonte: autoria própria

5 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Com a realização deste trabalho foi desenvolvida a estruturação para um *e-commerce*, separado em três aplicações funcionais independentes. As aplicações funcionam em conjunto, sendo que uma complementa a outra. A qualidade de um *e-commerce* está na sua estruturação, *layout* e segurança.

As lojas e varejistas brasileiros têm migrado cada vez mais para o comércio eletrônico, devido à globalização e a facilidade de interação com o cliente que apresenta uma grande variedade de produtos e preços de seus concorrentes, a loja que não possui este recurso fica em desvantagem perante os seus concorrentes.

O comércio eletrônico, principal foco desta pesquisa, proporciona comodidade, pois fornece a possibilidade de adquirir novos produtos ou serviços sem a pessoa sair de casa. O site deskshop.com teve como objetivo alcançar essa comodidade e possui planos de entrar no mercado competitivo de forma definitiva em 2020. Assim, pessoas que estiverem buscando produtos eletrônicos poderão encontrar na loja virtual.

A qualidade de um site é baseada em fatores de usabilidade e segurança. Neste trabalho, esses fatores foram implementados via código. Na Autenticação do usuário foi implementado o *token* de autenticação, aumentando a segurança. A criação do usuário armazena seus dados e sua senha é criptografada, garantindo a integridade do usuário.

Os relatórios do sistema foram exibidos em telas no painel de administração na forma de tabelas, futuramente será implementado a possibilidade de baixar essas informações no formato de PDF.

O tipo de pagamento foi ilustrado, para realizar essas ações de forma concreta, serão futuramente implementadas integrações com o “PagSeguro”, “PayPal” e outras plataformas de pagamento disponíveis no mercado virtual.

As principais dificuldades encontradas durante o desenvolvimento do trabalho foram inserir a integração com a API dos Correios, salvar o pedido do usuário e inserir o controle do estoque na plataforma. A integração com a API dos Correios foi a mais complexa de implementar pela dificuldade em entender seu funcionamento. Salvar o pedido foi complexo devido aos detalhes e validações existentes, tomando bastante tempo para o desenvolvimento dessa funcionalidade ficar completa. O controle de estoque foi complexo na elaboração de como seria seu funcionamento.

REFERÊNCIAS

- ADOLPHO, Conrado. **Google Marketing: o guia definitivo de marketing digital**. São Paulo: Novatec, 2011.
- ALBERTIN, Luiz. **Comércio eletrônico**. 2. ed. São Paulo: Atlas, 2000.
- ARRUDA, D. M. O.; Miranda, C. M. C. (2003). **Variáveis comportamentais determinantes de compra no varejo virtual**: um estudo com consumidores brasileiros. *Revista de Administração da UFLA*. v. 5, n. 2, p.112-133.
- BALARINE, Oscar Fernando Osorio. **Tecnologia da informação como vantagem competitiva**. *Revista de Administração Eletrônica*, v. 1, n. 1. São Paulo. Jan./Jun. 2002. Disponível em: <<http://www.scielo.br/pdf/raeel/v1n1/v1n1a05.pdf>>. Acesso em 12/09/2012.
- BARROS, L. C.; MUYLDER, C. F.; OLIVEIRA, M. C. S. M. **Tendência de compras pela internet versus compras tradicionais: um estudo exploratório**. In: ENCONTRO DE ADMINISTRAÇÃO DA INFORMAÇÃO – EnADI, 1., 2007, Florianópolis. *Anais eletrônicos...* Rio de Janeiro: ANPAD, 2007.
- BLOCH M.; PIGNEUR, Y.; SEGEV, A. **On the road of electronic commerce** : a business value framework, gaining competitive advantage and some research issues. Paris: Université de Lausanne, Ecole des Hautes Etudes Commerciales/Institut L'Informatique et Organisation, 1996.
- CATALANI, Luciane, André Kischinevsky, Eduardo Ramos, Heitor Simão. **E-commerce**. 2ª Edição Revisada. Rio de Janeiro: Editora FGV, 2006.
- CHILDERS, T.; Carr, C.; Peck, J.; Carson, S. **Hedonic and utilitarian motivations for online retail shopping behavior** *journal of retailing*, 2001, v. 77, p. 511-535.
- DE SOUZA, Marcos Gouvêa. Crescimento do Comércio Eletrônico. In: DE SOUZA, Marcos Gouvêa. **NEOCONSUMIDOR - DIGITAL, MULTICANAL E GLOBAL**. [S. l.]: GS&MD, 2016.
- EBIT. Evolução do e-commerce no Brasil. Disponível em: <https://www.e-commerce.org.br/e-commerce-no-brasil-perfil-do-mercado-e-do-e-consumidor-2/> Acesso em: 21 setembro de 2018.
- FREDRICKSEN, Clark. EcommerceSales. **Revista Emarket**. Disponível em: <<http://www.emarketer.com/Article/Ecommerce-Sales-Topped-1-Trillion-First-Time-2012/1009649>>. Acesso em: 24 ago. 2018.
- GRÖNROOS, Christian. **Marketing, gerenciamento e serviços**. São Paulo: Campus, 2003,
- HORTINHA, Joaquim. **X-Marketing**. Edições Sílabo. 1a edição. Lisboa, 2002.
- IBGE – INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. **Pesquisa Industrial de Inovação Tecnológica** 2015.

LIMEIRA, Tania Maria Vidigal. **E-Marketing: O marketing na internet com casos brasileiros**. São Paulo: Saraiva, 2003.

LUCIANO, Edimara Mezzomo. e FREITAS, Henrique M. Rodrigues **Comércio eletrônico de produtos virtuais: definição de um modelo de negócios para a comercialização de software 1**. 2002. Disponível em:

<<http://www.egov.ufsc.br/portal/sites/default/files/anexos/27413-27423-1-PB.pdf>>. Acesso em 08/09/2012.

LUCIANO. Edimara Mezzomo; TESTA. Mauricio Gregianin Testa e FREITAS. Henrique. **As tendências em comércio eletrônico com base em recentes congressos**. 2003. Disponível em:

<http://www.ea.ufrgs.br/professores/hfreitas/files/artigos/2003/2003_127_CLADEA.pdf>. Acesso em 12/09/2012.

KARSAKLIAN, Eliane. **Comportamento do consumidor**. 2º edição. São Paulo: Atlas, 2004.

MODAHL, Mary. **Agora ou nunca: como ganhar a guerra pelos consumidores na Internet**. Rio de Janeiro: Campus, 2000.

NAKAGAWA, S. S. Y. (2008) *Esso*, 312 f. Tese (**Doutorado em Administração**). Universidade de São Paulo. São Paulo. Brasil.

NAKAMURA, Rodolfo Reijiro. **E-Commerce na internet: fácil de entender**. São Paulo, Érica, 2001. ISBN 1970- 85-7194-750-3.

NEIGHBORS, J. Draco: a method for engineering reusable software system. In: PRIETO-DÍAZ, R.; ARANGO, G. **Domain analysis and software modeling**. Los Almitos: IEEE Computer Society Press, 1991.

SOUZA, A. R. C.; Farias, J. M. P.; Nicoluci, T. C. (2005). **Principais fatores que influenciam os consumidores universitários a frequentarem bares noturnos na cidade de Presidente Prudente**. 47 f. Monografia (Graduação em Administração) – Faculdades Integradas Antônio Eufrásio de Toledo, Presidente Prudente. Brasil.

WONGTSCHOWSKI, Arthur. **Segurança em aplicações transacionais na internet: o elo mais fraco**. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/3/3141/tde-05092006-175654/en.php>>. Acesso em: out. 2011.