

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE SISTEMAS DE INFORMAÇÃO – BACHARELADO

PLATAFORMA DE MARKETPLACE PARA VENDEDORES
DIRETOS

VINÍCIUS ZARTH PEROTTO

BLUMENAU
2018

VINÍCIUS ZARTH PEROTTO

PLATAFORMA DE MARKETPLACE PARA VENDEDORES

DIRETOS

Trabalho de Conclusão de Curso apresentado ao curso de graduação em Sistemas de Informação do Centro de Ciências Exatas e Naturais da Universidade Regional de Blumenau como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

Prof^a Luciana Pereira de Araújo Kohler, Mestra - Orientadora

**BLUMENAU
2018**

PLATAFORMA DE MARKETPLACE PARA VENDEDORES

DIRETOS

Por

VINÍCIUS ZARTH PEROTTO

Trabalho de Conclusão de Curso aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II pela banca examinadora formada por:

Presidente:

Prof^a. Luciana Pereira de Araújo Kohler, Mestra – Orientadora, FURB

Membro:

Prof^a. Simone Erbs da Costa, Especialista – FURB

Membro:

Prof. Samuel Cristhian Schwebel, Mestre – FURB

Blumenau, 12 de julho de 2018

Dedico este trabalho aos meus familiares, colegas e professores que de alguma forma contribuíram para sua realização.

AGRADECIMENTOS

À minha família por todo o carinho e apoio que sempre recebi.

Aos meus amigos, pela disponibilidade para opinar e discutir melhores práticas e ideias para o desenvolvimento do trabalho.

À minha orientadora, Prof^a. Luciana Pereira de Araújo Kohler, pela motivação e orientação no desenvolvimento deste trabalho.

Sabe o que é estranho? Nada parece mudar de um dia para o outro, mas de repente tudo fica diferente.

Bill Watterson

RESUMO

Este trabalho apresenta o desenvolvimento de uma plataforma web de *marketplace* direcionada a vendedores diretos, profissionais independentes que vendem produtos diretamente ao consumidor final, sem possuir um estabelecimento fixo. A plataforma tem como objetivo permitir que vendedores diretos trabalhem com novos fornecedores e centralizem seus pedidos de compra em um único *website*, acessível a partir de qualquer dispositivo conectado à internet. Para tal, foi implementado um fluxo de cadastramento de revendedores, bem como funcionalidades para o recebimento de pedidos de diversos fornecedores, formando um *marketplace*. O *backend* do trabalho foi desenvolvido utilizando a linguagem de programação PHP e banco de dados MySQL, enquanto o *frontend* da aplicação foi construído com o *framework* Bulma. Para validar o desempenho da plataforma em dispositivos móveis e desktop e em navegadores diferentes foram utilizados o PageSpeed Insights e o BrowserShots. Por fim, algumas interfaces do sistema foram comparadas com *guidelines* do Material Design para validar a usabilidade e acessibilidade das páginas.

Palavras-chave: Marketplace. Vendas diretas. Usabilidade.

ABSTRACT

This essay presents the development of a web marketplace platform for direct sellers, independent professionals who sell products directly to the final customer without the need of a physical space. The platform aims to allow direct sellers to work with new suppliers and centralize all their orders in one website, reachable from any device connected to the internet. To achieve this, the software implements a registration flow for sellers as well as features that allow orders from multiple suppliers, building an online marketplace. The project backend was developed using the PHP programming language and MySQL database, while the application frontend was built with Bulma Framework. To test the applications performance in mobile devices and desktop computers, as well as different web browsers, the tools PageSpeed Insights and BrowserShots were used. In the end, some interfaces were compared with Material Design Guidelines to validate the usability and accessibility of the website.

Key-words: Marketplace. Direct selling. Usability.

LISTA DE FIGURAS

Figura 1 – Elo7: Catálogo de Produtos.....	19
Figura 2 – Elo7: Cadastro de Produtos.....	19
Figura 3 – MercadoLivre: Lista de produtos de uma categoria.....	21
Figura 4 – MercadoLivre: Cadastro de anúncio	22
Figura 5 – GeraSGI: Fluxo de cadastro de revendedor	23
Figura 6 – GeraSGI: Fluxo de Pedidos.....	23
Figura 7 - Diagrama de Casos de Uso	27
Figura 8 - Diagrama de MER	29
Figura 9 - Arquitetura MVC.....	30
Figura 10 - Visualização das tarefas em Kanban - Trello	31
Figura 11 - Organização do projeto com a arquitetura MVC.....	32
Figura 12 - Diagrama de classes do pacote Model.....	37
Figura 13 - Diagramas de Atividades	38
Figura 14 - Tela de Identificação da plataforma.....	39
Figura 15 - Cadastro de Revendedor na plataforma	39
Figura 16 - Painel do revendedor	40
Figura 17 - Catálogo de produtos	41
Figura 18 - Modal para adicionar produto ao carrinho.....	41
Figura 19 - Carrinho do cliente.....	42
Figura 20 – Formalização do pedido	42
Figura 21 - Detalhes do pedido.....	43
Figura 22 - Tela de Fornecedores	44
Figura 23 - Modal de confirmação	44
Figura 24 - Teste de desempenho do responsivo.....	46
Figura 25 - Teste de desempenho em um desktop.....	47
Figura 26 - Teste de compatibilidade de Navegadores	47
Figura 27 - Material Design: Indicar a ação de um elemento	48
Figura 28 - Material Design: Utilização de ícones	49
Figura 29 - Material Design: Botões primários e secundários	49
Figura 30 - Material Design: Validação de formulários.....	50

LISTA DE QUADROS

Quadro 1 - Comparativo de características dos trabalhos correlatos.....	24
Quadro 2 - Requisitos funcionais com a matriz de rastreabilidade	26
Quadro 3 - Requisitos não funcionais	26
Quadro 4 - Definição de rotas do FlightPHP.....	33
Quadro 5 - Controlador da lista de produtos	33
Quadro 6 - Rotas do fluxo de realização de pedidos	34
Quadro 7 - Rotas das funções administrativas	34
Quadro 8 - Requisição POST ao Mailgun	35
Quadro 9 - Notificação de pedido enviado.....	36
Quadro 10 - Comparação do trabalho desenvolvido com os correlatos	45
Quadro 11 – Descrição do caso de uso Cadastrar Revendedor (UC01).....	55
Quadro 12 - Descrição do caso de uso Solicitar Aprovação de Fornecedor (UC03)	56
Quadro 13 - Descrição do caso de uso Realizar Pedido (UC07).....	57

LISTA DE ABREVIATURAS E SIGLAS

ABEVD – Associação Brasileira das Empresas de Venda Direta

API – Application Programming Interface

B2B – Business to Business

B2C – Business to Customer

C2C – Customer to Customer

CNPJ – Cadastro Nacional de Pessoa Jurídica

CPF – Cadastro de Pessoa Física

CSS – Cascading Style Sheets

HTML – HyperText Markup Language

HTTP – HyperText Transfer Protocol

IDE – Integrated Development Environment

MER – Modelo de Entidade e Relacionamento

MVC – Model-View-Controller

ORM – Mapeamento Objeto-Relacional

PHP – PHP: Hypertext Preprocessor

POO – Programação Orientada a Objetos

RF – Requisitos Funcionais

RNF – Requisitos Não Funcionais

SAAS - Software As A Service

SGBD – Sistema de Gerenciamento de Banco de Dados

UC – Use Case

UML – Unified Modeling Language

URL – Uniform Resource Locator

WFDSA – World Federation of Direct Selling Associations

SUMÁRIO

1 INTRODUÇÃO.....	13
1.1 OBJETIVOS.....	13
1.2 ESTRUTURA.....	14
2 FUNDAMENTAÇÃO TEÓRICA	15
2.1 VENDAS DIRETAS	15
2.2 E-COMMERCE	15
2.3 USABILIDADE EM APLICAÇÕES WEB.....	17
2.4 TRABALHOS CORRELATOS	18
2.4.1 Elo7	18
2.4.2 MercadoLivre.....	20
2.4.3 GeraSGL.....	22
2.4.4 Comparação entre os trabalhos correlatos.....	24
3 DESENVOLVIMENTO.....	25
3.1 LEVANTAMENTO DE INFORMAÇÕES	25
3.2 ESPECIFICAÇÃO	26
3.2.1 Diagrama de Casos de Uso	27
3.2.2 Modelo de Entidade e Relacionamento.....	28
3.3 IMPLEMENTAÇÃO	29
3.3.1 Técnicas e ferramentas utilizadas.....	29
3.3.2 Desenvolvimento.....	32
3.3.3 Operacionalidade da implementação	37
3.4 RESULTADOS E DISCUSSÕES.....	45
4 CONCLUSÕES.....	51
4.1 EXTENSÕES	52
REFERÊNCIAS	53
APÊNDICE A – DESCRIÇÃO DOS CASOS DE USO	55

1 INTRODUÇÃO

Venda direta é uma modalidade de venda de serviços e bens de consumo baseada no contato pessoal entre vendedor e comprador, sem a necessidade de um estabelecimento comercial fixo (ABEVD, 2016). De acordo com a World Federation of Direct Selling Associations (WFDSA), no ano de 2016, essa modalidade de comércio movimentou US\$ 182,5 bilhões no mundo todo, sendo US\$ 8,6 bilhões só no Brasil, por meio de 4,3 milhões de representantes (WFDSA, 2017). Ainda segundo a WFDSA (2017, p. 1), o Brasil está em sexto lugar no ranking mundial de vendas diretas, com cerca de 5% das vendas, atrás de Estados Unidos (19%), China (19%), Coreia (9%), Alemanha (9%) e Japão (8%).

Para o consumidor, a venda direta fornece um atendimento personalizado que não existe no varejo tradicional, enquanto para os revendedores, representa um complemento à renda familiar ou também uma saída de baixo investimento para o desempregado (ABEVD, 2016). No Brasil, os produtos de beleza lideram o mercado de venda direta, chegando a 84% do volume de negócios (WFDSA, 2017). Ao mesmo tempo, nos Estados Unidos, os setores de beleza e nutrição concentram apenas 30% das vendas (ABEVD, 2015, p. 1). Devido à popularidade do setor de beleza e cosméticos no Brasil, potenciais revendedores deixam de entrar no mercado por não se identificarem com o segmento, ou por não conhecerem oportunidades de outros segmentos.

Embora os *e-marketplaces* não sejam novidade, em sua grande maioria são voltados ao tradicional mercado Business to Customer (B2C), ou seja, de empresa para cliente, para consumo próprio. Esta plataforma se faz relevante pois irá atuar como uma galeria de fornecedores intermediando as relações entre fornecedor e vendedor direto, para que novos empreendedores conheçam novos fornecedores e passem a revender seus produtos, apresentando novos segmentos que ainda são pouco explorados, como roupas, bens duráveis e até serviços. Dessa forma, ajuda a fortalecer e expandir novos negócios por meio da representação comercial.

1.1 OBJETIVOS

Este trabalho tem como objetivo desenvolver um *marketplace* voltado ao mercado de vendas diretas.

Os objetivos específicos são:

- a) disponibilizar uma forma para vendedores diretos comprarem produtos de diversos fornecedores;
- b) validar o sistema de acordo com padrões de usabilidade para a web.

1.2 ESTRUTURA

Este trabalho está dividido em quatro capítulos. O primeiro capítulo contém a introdução do tema, a justificativa para o desenvolvimento do trabalho e seus objetivos. O segundo capítulo apresenta a fundamentação teórica, que inclui a apresentação dos conceitos de vendas diretas, *e-commerce* e usabilidade em aplicações web. No segundo capítulo também são apresentados três trabalhos correlatos, sendo dois *marketplaces* e uma plataforma para gestão de vendas diretas.

O terceiro capítulo apresenta o desenvolvimento do trabalho, contendo sua especificação, técnicas e ferramentas utilizadas e a demonstração da operacionalidade das soluções implementadas. Além disso, são apresentados os resultados alcançados com o desenvolvimento. Por fim, o quarto e último capítulo apresenta as conclusões e limitações do trabalho, além de sugestões para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Na revisão bibliográfica são explorados os principais assuntos relacionados a este trabalho. Na seção 2.1 estão alguns detalhes a respeito do segmento de vendas diretas. Na seção 2.2 é contextualizado o conceito de *e-commerce* e *e-marketplace*. A seção 2.3 destaca a importância dos padrões de usabilidade na web. Por fim, a seção 2.4 apresenta três trabalhos correlatos a este e a comparação de suas funcionalidades.

2.1 VENDAS DIRETAS

No Brasil, a atividade de venda direta é promovida pela Associação Brasileira das Empresas de Venda Direta (ABEVD) que define diretrizes de ética para as empresas e vendedores. Segundo ABEVD (2016, p. 1), “A venda direta é um sistema de comercialização de bens de consumo e serviços baseado no contato pessoal entre vendedores e compradores, fora de um estabelecimento comercial fixo.”. Esse modelo começou a se popularizar no Brasil em 1943, com a Hermes, e foi ganhando força nos anos seguintes até a criação da ABEVD em 1980, quando o país já contava com representantes das marcas Avon, Natura e TupperWare (MOLA, 2013).

A ABEVD representa os vendedores diretos e definiu um código de ética que garante segurança tanto ao revendedor quanto ao consumidor. O código de ética estabelecido pela associação tem como objetivo a satisfação do consumidor, a concorrência leal e a “[...] disseminação da imagem pública da venda direta e para a percepção pela sociedade da atividade de venda direta como oportunidade de trabalho e geração de renda.” (ABEVD, 2009, p. 1). Miyata (2010, p. 114) afirma que entre os anos de 1980 e 2000, as vendas diretas no Brasil cresceram cerca de 20% ao ano.

Segundo Miyata (2010, p. 48), na venda direta, os revendedores podem obter lucro de duas maneiras, em um sistema mononível ou multinível. No sistema mononível, o vendedor compra o produto e o revende, captando uma margem média de 30% de lucro (ABEVD, 2016). No sistema multinível, “[...] além da margem de lucro, o revendedor que indicar outros revendedores também ganha uma porcentagem em cima dessas vendas.” (ABEVD, 2016, p. 1). Desta forma, os ganhos do vendedor são proporcionais ao seu esforço e ao crescimento de sua rede de contatos.

2.2 E-COMMERCE

De acordo com Laudon e Traver (2014, p. 10, tradução nossa), *e-commerce* é “a utilização da internet, web e aplicativos móveis para transações de negócios.”. Turban et al.

(2015), define comércio eletrônico como a utilização da internet com a finalidade de comprar, vender, transportar ou negociar dados, bens e serviços. No Brasil, o *e-commerce* popularizou-se nos anos 2000 com seu sinônimo de “loja virtual”. Em outras palavras, uma loja disponível na internet 24 horas por dia, todos os dias do ano que oferece a comodidade de poder comprar e receber produtos sem sair de casa.

Um *e-commerce* pode possuir várias modalidades. As modalidades mais comuns são Business to Customer (B2C), sendo a forma tradicional de varejo em que se vende ao cliente final e Business to Business (B2B), na qual uma empresa vende para outra empresa (TURBAN et al., 2015). Há, inclusive, sites que misturam mais de uma modalidade, vendendo para cliente final e empresas (distribuidores) na mesma plataforma.

Turban et al. (2015) define *e-marketplace*, ou simplesmente *marketplace*, como um lugar no qual compradores e vendedores fazem transações comerciais, como venda de serviços, bens ou informações. Embora o conceito de *e-marketplace* seja relativamente parecido com o conceito de *e-commerce*, o grande diferencial está na disponibilidade dos vendedores. Enquanto um *e-commerce* normalmente é constituído por apenas um vendedor, os *marketplaces* agrupam “[...] diferentes tipos de produtos e serviços [de diversos vendedores] a fim de atingir o maior número de clientes possíveis.” (SEBRAE, 2017, p. 1).

Para o Sebrae (2017, p. 1), os *marketplaces* têm como principal benefício aos lojistas a possibilidade de atingir novos clientes sem investimento em marketing e publicidade (estes ficam a cargo do *marketplace*). Enquanto para o consumidor, a vantagem está em encontrar produtos de vários vendedores e segmentos em um mesmo local. Comparando com as formas mais tradicionais de comércio, os *marketplaces* se assemelham às feiras: uma vez que entrou na feira, o consumidor está sujeito a conhecer – e comprar – produtos de vários lojistas (SEBRAE, 2017, p. 1).

No modelo de negócio dos *marketplaces*, o estoque e a logística são de responsabilidade do fornecedor, enquanto o *marketplace* atua como o intermediador da transação (DALVI, 2015, p. 1). Dalvi (2015, p. 1) ainda pontua que os *marketplaces* podem disponibilizar produtos ou serviços que antes eram inalcançáveis ou desconhecidos pelos clientes. De acordo com Turban et al. (2015, p. 168), os *marketplaces* podem ser classificados como verticais e horizontais. *Marketplaces* verticais atendem a uma indústria ou segmento específico, também conhecidos como “*marketplaces* segmentados”, enquanto os horizontais ofertam produtos de vários segmentos. Um exemplo de *marketplace* vertical é o ELO7 (2017), descrito na seção 2.4.1 deste trabalho. No ELO7, pequenos vendedores e artesãos anunciam seus produtos para um público segmentado.

2.3 USABILIDADE EM APLICAÇÕES WEB

A usabilidade de uma aplicação está relacionada à interação que o usuário estabelece com a interface do sistema e os demais aspectos do ambiente. Cybis, Betiol e Faust (2010, p. 24) definem usabilidade como “[...] a qualidade que caracteriza o uso de um sistema interativo.”. Embora a presença do usuário seja imprescindível para a construção de um sistema que preza pela usabilidade, há certos princípios e heurísticas de usabilidade que facilitam a construção de interfaces que favorecem a relação usuário-sistema (CYBIS; BETIOL; FAUST, 2010, p. 24).

Para Krug (2006), a regra número um da usabilidade é não fazer o usuário pensar. O autor relata que as páginas da web devem ser óbvias e autoexplicativas, de forma que o usuário possa identificar o conteúdo e interagir com a página sem precisar pensar muito. O autor ainda compara uma página web a um *outdoor* a 60 milhas por hora, explicando que o usuário não lê um site inteiro, e sim busca por textos e palavras-chave que chamam a atenção. Isto se dá principalmente por três fatores: pressa; o usuário está ciente de que não precisa ler o site todo, pois está interessado apenas em uma fração do que está na página; o ser humano é bom em escanear textos buscando informações, o faz inclusive na leitura de livros e revistas, e sabe que funciona (KRUG, 2006, p. 22, tradução nossa).

De acordo com Nielsen (1997), 79% dos usuários fazem uma leitura rápida de páginas novas da web em busca de palavras chave, enquanto apenas 16% as lê palavra por palavra. Testes feitos pelo autor com usuários, apontam que o uso combinado de textos concisos, escaneáveis e objetivos podem melhorar em até 124% a usabilidade de uma página web (NIELSEN, 1997). Remover botões e distrações desnecessárias de acordo com o contexto também facilitam a compreensão da página e permite que o usuário mantenha o foco para completar sua tarefa, principalmente tratando-se de páginas com formulários (KRUG, 2006).

Em um *e-commerce*, um ponto importante é garantir que o usuário consiga seguir o fluxo de uma atividade do começo ao fim para cumprir seu objetivo. Para Reiss (2012) um problema comum nos *e-commerces* ocorre quando o site solicita ao usuário dados de acesso no meio do *checkout*. Para o autor, é um problema quando um *website* quebra um processo sequencial pedindo que o usuário interrompa sua tarefa atual e realize outra ação antes de continuar sua jornada (REISS, 2012). Quando há dois formulários interdependentes que devem ser preenchidos, estes devem ser exibidos ao usuário seguindo uma ordem lógica, sem que o usuário precise interromper o preenchimento de um deles para completar o outro (REISS, 2012, p. 11, tradução nossa).

2.4 TRABALHOS CORRELATOS

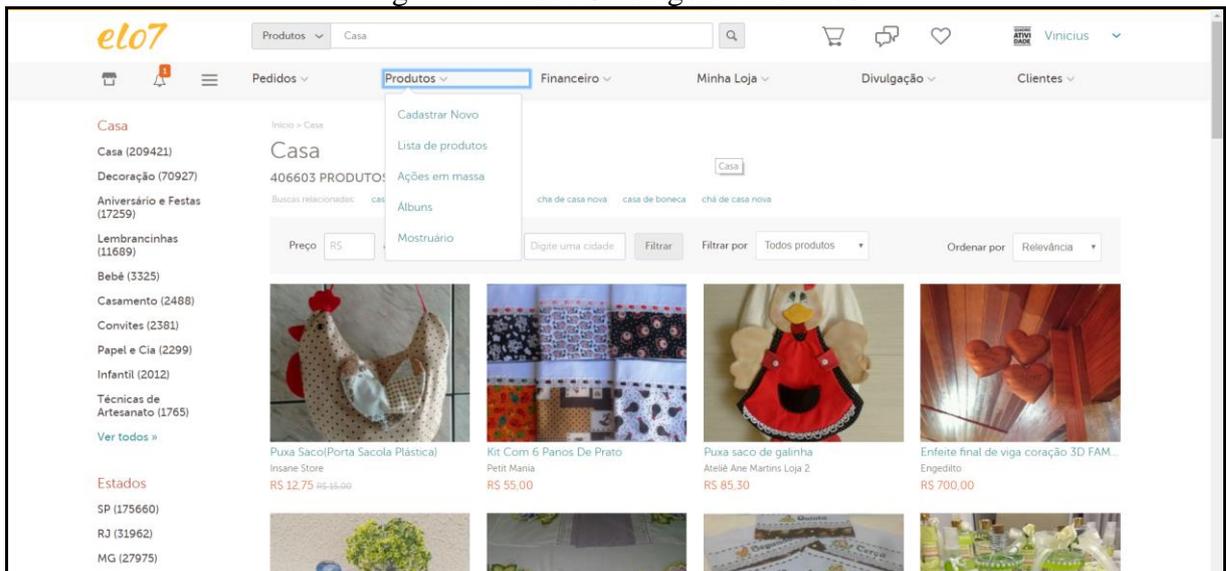
São apresentados três trabalhos correlatos com características semelhantes aos principais objetivos deste trabalho. A subseção 2.4.1 detalha o Elo7, um *marketplace* de artesanato (ELO7, 2017). A subseção 2.4.2 detalha o MercadoLivre, um *marketplace* para pessoas e empresas (MERCADOLIVRE, 2018). A subseção 2.4.3 descreve uma ferramenta de gestão de venda direta, o GeraSGI (GERA, 2017a). Por fim, na subseção 2.4.4 é apresentada uma comparação entre os trabalhos citados.

2.4.1 Elo7

O Elo7 foi lançado em 2008 como um *marketplace* para venda de produtos autorais, criativos e de artesanato (ELO7, 2017). É um sistema web e possui aplicativos móveis para o acompanhamento dos pedidos, tanto do vendedor quanto do comprador. O Elo7 permite que cada vendedor crie sua própria loja dentro da plataforma, exibindo seus produtos, suas qualificações por meio de avaliações de clientes e tire dúvidas de clientes interessados em seus produtos a partir de um chat.

Em seu modelo de negócios, os próprios artistas preenchem um cadastro no site e em poucos minutos estão habilitados para cadastrarem seus produtos. A plataforma se encarrega de exibir o catálogo de produtos (Figura 1), receber pedidos, captar pagamentos, gerar etiquetas de postagem no formato dos Correios e solicitar avaliações dos produtos e vendedores. No topo esquerdo do site há uma logo, que direciona o cliente para a *homepage*. Ao lado da logo há um campo de busca, em que é possível pesquisar pelo nome ou por *tags* dos produtos. Embaixo do campo de busca fica o menu principal, no qual há links para consultar seus últimos pedidos e links direcionando para as telas de listagem e cadastro de novos produtos.

Figura 1 – Elo7: Catálogo de Produtos



Fonte: Elo7 (2017).

Na tela de cadastro de produtos (Figura 2) há um formulário com todos os campos necessários para a criação de um anúncio. Na primeira linha do formulário, há um campo “lista de produtos”, no qual é possível selecionar um produto já cadastrado cujos dados serão copiados para este novo anúncio. Nas linhas seguintes, há campos para o cadastro de título, preço, categoria, descrição, fotos e dimensões para cotação de fretes (ELO7, 2017).

Figura 2 – Elo7: Cadastro de Produtos

Fonte: Elo7 (2017).

O Elo7 (2017) não oferece Application Programming Interfaces (API) para integração com outros sistemas, nem ferramentas para importação de anúncios em lotes. No ano de 2017, disponibilizou as primeiras funcionalidades para auxiliar os vendedores a receber pedidos por

meio da venda direta, permitindo que o vendedor envie um link com o produto, por meio do qual o comprador pode efetuar o pagamento. No entanto, o Elo7 não está focado nos revendedores, apenas no fornecedor e no cliente final (B2C).

Elo7 (2017, p. 1) descreve os principais objetivos da plataforma como “[...] oferecer aos vendedores soluções simples e práticas para o gerenciamento de suas vendas.”. Além disso, para os clientes o Elo7 é “[...] uma experiência de compra fora de série [...]” (ELO7, 2017, p. 1). Dos vendedores, é cobrada uma comissão a cada pedido vendido, de acordo com o tipo de anúncio selecionado, que pode ser clássico (anúncio não possui nenhum destaque) ou plus (anúncio aparece com prioridade nas listagens e busca de produtos).

2.4.2 MercadoLivre

O MercadoLivre é o “[...] maior e mais importante *marketplace* da América Latina” (MARTINA, 2016, p. 1, tradução nossa). Fundado no ano de 2009 por Marcos Galperín, é um *marketplace* B2C que permite que empresas e pessoas físicas vendam produtos, imóveis e automóveis novos ou usados. Possui operações em 19 países, dentre eles Argentina, Brasil, Chile e Portugal (MERCADOLIVRE, 2018).

No MercadoLivre é possível comprar e vender sem a necessidade de um Cadastro Nacional de Pessoa Jurídica (CNPJ). As vendas podem ser realizadas nas modalidades B2B, na qual uma empresa vende para outra empresa, B2C, na qual uma empresa vende a um cliente final ou Customer to Customer (C2C), na qual um cliente final vende para outro cliente final.

A plataforma é dividida em duas áreas. A primeira na qual os compradores podem ver os anúncios disponíveis e realizar pedidos. A segunda na qual os vendedores gerenciam seus produtos e pedidos. Na área do comprador, é possível visualizar uma lista de anúncios (Figura 3) com imagem, preço e nome do produto, de acordo com a categoria selecionada. No topo do site há um campo de busca, em que se pode pesquisar anúncios pelo nome do produto ou palavras-chave. Ao clicar em um anúncio é exibida a página de detalhes do anúncio, no qual se pode ver sua descrição, preço, formas de pagamento e realizar perguntas ao vendedor.

A partir da página de detalhes do anúncio, o comprador pode simular os custos de envio e formalizar o pedido, informando os dados de pagamento e o endereço de entrega. Confirmado o pedido e capturado o pagamento, ambos os usuários (vendedor e comprador) são notificados por e-mail, e o vendedor está autorizado a postar o pedido no Correios ou transportadora. Após o recebimento do pedido, o cliente recebe um e-mail solicitando que avalie a transação.

Figura 3 – MercadoLivre: Lista de produtos de uma categoria

mercado livre

Somente em Ferramentas para Jardim

ZP.VINI Contato Vender

SEMANA DO CLIENTE ATÉ 70% OFF | ATÉ 12X SEM JUROS | FRETE GRÁTIS* Ver ofertas

*Para todo o Brasil em produtos selecionados

Buscas relacionadas: poltronas decorativas - sofa - espelhos decorativos - cortina quarto - banquetas cozinha

Casa, Móveis e Decoração > Jardim

Ferramentas para Jardim

37.403 resultados

Organizar anúncios

Mais relevantes

Categorias

- Soprador de Folhas (1.489)
- Roçadeiras (15.127)
- Motosserras (5.881)
- Cortador de Grama (3.652)
- Pulverizador (2.728)
- Enxada (422)
- Aparadores de Cerca (365)
- Pá de Bico (234)
- Ancinho (143)

Ver todos

Condição

- Novo (36.346)
- Usado (1.000)

Localização

- Paraná (17.313)
- São Paulo (10.000)

R\$ 189
12x R\$ 15⁷⁵ sem juros
Aparador De Grama Trapp - Mester 1000w Plus - Original por Dimeo

R\$ 206
R\$ 168 18% OFF
12x R\$ 14 sem juros
Aparador De Grama Elétrico 1000w Garthen Gam1000 por Dimeo

R\$ 99
12x R\$ 8²³ sem juros
Kit Para Jardinagem Com 10 Peças Com Maleta - Noll por Olist

Fonte: MercadoLivre (2018).

Na área do vendedor (acessada por meio do link “vender” no canto superior direito do site) é permitido cadastrar produtos, gerenciar suas vendas e responder perguntas de clientes interessados. Na tela de cadastro de anúncios (Figura 4) há um formulário no qual é possível adicionar um título, selecionar uma categoria, adicionar fotos e uma descrição para o item.

Segundo Martina (2016, p. 1), no ano de 2015 a empresa detentora da plataforma MercadoLivre intermediou negociações entre 24 milhões de compradores e 8 milhões de vendedores, em um ritmo de 7 transações por segundo, sendo a empresa líder em comércio eletrônico de toda a América Latina. Nota-se a partir destes números o potencial da ferramenta para vendas *online*, entretanto não é disponibilizada nenhuma funcionalidade específica para vendedores diretos.

Figura 4 – MercadoLivre: Cadastro de anúncio

mercado livre

Escolha o que você quer anunciar

Descreva o seu produto

Informe preço e quantidade

Categoria

Produtos e outros

Celulares e Telefones > Celulares e Smartphones > Lenovo [Modificar](#)

Adicione fotos do seu produto

Agora, você pode enviar até 10 fotos do seu produto!

Mostre-o no detalhe, com fundo branco e bem iluminado. Não inclua logos, banners nem textos promocionais.

Adicionar Adicionar Adicionar Adicionar Adicionar Adicionar

Foto principal

Descreva o seu produto

* Dados obrigatórios

Título: * ?

Restam 26 caracteres.

Fonte: MercadoLivre (2018).

2.4.3 GeraSGI

GeraSGI é uma ferramenta para gestão de equipe de vendas (GERA, 2017a). O software contém módulos fiscal, de vendas, marketing, logística e atendimento. Dessa forma são compreendidos todos os processos de uma empresa de venda direta. O GeraSGI permite alocação de revendedores, atribuição de metas por região e acompanhamento de resultados históricos (GERA, 2017a).

O fluxo de cadastro de um revendedor, representado na Figura 5, possibilita que o usuário candidate-se preenchendo um formulário no site do fornecedor. Na primeira etapa, o usuário preenche um formulário indicando o interesse em revender os produtos. Um executivo de vendas apresenta o negócio ao candidato, que envia seus documentos ao executivo de vendas do fornecedor. O cadastro passa por uma fase de aprovação, e quando completo, o revendedor passa por um treinamento e então pode registrar os pedidos na aplicação (GERA, 2017a).

Figura 5 – GeraSGI: Fluxo de cadastro de revendedor



Fonte: Gera (2017a).

A Figura 6 apresenta o fluxo de um pedido na ferramenta GeraSGI (GERA, 2017a). Para registrar um pedido, o revendedor entra em contato via telefone ou acessa o sistema e faz a verificação de estoque, crédito e escolhe a forma de pagamento. Realizado o pedido, o revendedor é notificado do sucesso da transação (GERA, 2017a).

Figura 6 – GeraSGI: Fluxo de Pedidos



Fonte: Gera (2017a).

Com o software GeraSGI, também é possível gerenciar metas e comissões para equipes mono ou multi nível que irão atuar com vendas diretas (GERA, 2017a). A aplicação é disponibilizada nos modelos Software As A Service (SAAS), sendo o software como serviço, entregue pronto para usar, e *on-premise*, quando o software é instalado em servidor do cliente, sendo necessária uma instalação independente para cada empresa (GERA, 2017b).

O modelo SAAS é o modelo de hospedagem atualmente indicado pela equipe Gera (2017b), por promover maior agilidade na implantação, uma vez que a aplicação já está na nuvem. Além de promover maior agilidade, o SAAS aumenta a performance e disponibilidade, sem que o cliente necessite de uma equipe técnica para manter a infraestrutura e o sistema (GERA, 2017b).

2.4.4 Comparação entre os trabalhos correlatos

Nesta subseção é apresentada uma comparação entre os trabalhos correlatos citados. Cada trabalho possui certas características voltadas ao seu mercado foco, que juntas compõem as características desejadas para o software desenvolvido neste trabalho. No Quadro 1 são comparadas as características entre as ferramentas que se assemelham ao desenvolvido por este trabalho.

Quadro 1 - Comparativo de características dos trabalhos correlatos

Características/Trabalhos relacionados	Elo7 (2017)	MercadoLivre (2018)	Gera (2017a)
Interface entre fornecedor e vendedor direto			X
Venda em grandes quantidades (atacado)		X	X
Vários fornecedores (<i>marketplace</i>)	X	X	
Clientes avaliam produtos e reputação do fornecedor	X	X	
Captura de pagamentos	X	X	
Vitrine de produtos (catálogo <i>online</i>)	X	X	X
Permite candidatura de revendedores			X

Fonte: elaborado pelo autor.

Como é visto no Quadro 1, apenas o GeraSGI (Gera, 2017) fornece uma interface entre fornecedor e vendedor direto, sendo que este não é um *marketplace* e está associado a um único fornecedor, que deve ser conhecido previamente antes que um revendedor possa se candidatar para representar seus produtos. Os outros trabalhos correlatos, ELO7 (2017) e MercadoLivre (2018), embora sejam *marketplaces* completos, não possuem funcionalidades para intermediar o relacionamento entre um vendedor direto e um ou mais fornecedores.

Este estudo se faz relevante, pois embora os *e-marketplaces* não sejam novidade, em sua grande maioria são voltados ao tradicional mercado B2C, ou seja, de empresa para cliente, para consumo próprio. O propósito deste trabalho é prover uma plataforma que atue como uma galeria de fornecedores e como um intermediador entre fornecedor e vendedor direto, para que novos empreendedores conheçam novos fornecedores e passem a revender seus produtos. Atuar como facilitador para que novos representantes possam trabalhar com novas empresas é relevante pois, como citado na introdução deste trabalho (capítulo 1), a venda direta é uma possibilidade de complemento de renda e uma alternativa ao desemprego (ABEVD, 2016). Dessa forma, ajuda a fortalecer e expandir novos negócios por meio da representação comercial.

3 DESENVOLVIMENTO

Neste capítulo, dividido em quatro seções, estão descritas as especificações e particularidades técnicas da plataforma desenvolvida neste trabalho. Na primeira seção é realizado o levantamento de informações e especificação dos Requisitos Funcionais (RF) e Requisitos Não Funcionais (RNF). Na segunda seção é apresentada a especificação do sistema. Na terceira seção são apresentadas as técnicas e ferramentas utilizadas no desenvolvimento do trabalho, bem como a operacionalidade da implementação. Por fim, na quarta seção encontram-se os resultados e discussões alcançados com o desenvolvimento deste trabalho.

3.1 LEVANTAMENTO DE INFORMAÇÕES

Este trabalho apresenta o desenvolvimento de uma plataforma web de *marketplace* em que vendedores diretos podem conhecer novos fornecedores, conhecer seu catálogo de produtos e comprá-los para revender. Permite-se então centralizar as compras de um vendedor direto em um único site, mesmo que este esteja associado a múltiplos fornecedores. Sendo uma plataforma web, também permite que o vendedor consulte e realize novos pedidos a partir de qualquer lugar ou dispositivo, *desktop* ou móvel, desde que esteja conectado à internet.

A plataforma compreende toda a etapa de cadastro de vendedor direto, a escolha de quais fornecedores gostaria de representar e o processo de aprovação para que então possa começar a comprar produtos para revenda. Foram identificadas as necessidades relacionadas ao processo de cadastramento e de vendas de empresas que trabalham com vendas diretas. Para o desenvolvimento do sistema, tais necessidades foram organizadas em Requisitos Funcionais (RF). No Quadro 2 são listados todos os RF desenvolvidos e sua rastreabilidade, vinculando cada requisito aos respectivos Casos de Uso (UC) associados.

Quadro 2 - Requisitos funcionais com a matriz de rastreabilidade

Requisitos Funcionais	Caso de Uso
RF01: O sistema deve permitir manter usuários de fornecedores, revendedores e administradores.	UC01, UC02, UC12
RF02: O sistema deve permitir inativar usuários.	UC01, UC02, UC03
RF03: O sistema deve permitir manter produtos.	UC06
RF04: O sistema deve permitir que fornecedores interessados se candidatem para começar a vender na plataforma.	UC02
RF05: O sistema deve permitir que um usuário administrador aprove o cadastro de um usuário fornecedor.	UC09
RF06: O sistema deve permitir que revendedores se candidatem para representar um fornecedor.	UC03
RF07: O sistema deve permitir que fornecedores aprovem a candidatura de um revendedor para então poder visualizar os produtos disponíveis.	UC04
RF08: O sistema deve permitir realizar pedidos.	UC07
RF09: O sistema deve permitir que o revendedor cancele seu pedido.	UC08
RF10: O sistema deve notificar o revendedor por e-mail quando um pedido for confirmado.	UC07
RF11: O sistema deve notificar o fornecedor por e-mail quando um novo pedido for registrado.	UC07
RF12: O sistema deve notificar o revendedor por e-mail quando um pedido for aprovado e enviado.	UC13
RF13: O sistema deve permitir que o revendedor entre em contato com o fornecedor.	UC05
RF14: O sistema deve emitir relatório de vendas por fornecedor.	UC11
RF15: O sistema deve emitir relatório de compras por revendedor.	UC10

Fonte: elaborado pelo autor.

O Quadro 3 lista os Requisitos Não Funcionais (RNF) definidos para o sistema.

Quadro 3 - Requisitos não funcionais

Requisitos Não Funcionais
RNF01: O sistema deve possuir leiaute responsivo.
RNF02: O sistema deve ser feito para a plataforma web.
RNF03: O sistema deve armazenar seus dados em um banco de dados MySQL.
RNF04: O <i>backend</i> do sistema deve ser construído na linguagem PHP: Hypertext Preprocessor (PHP).
RNF05: O <i>backend</i> do sistema deve ser construído utilizando um framework Model-View-Controller (MVC).
RNF06: O sistema deve possuir três níveis de acesso: administrador, fornecedor e revendedor.
RNF07: O sistema deve realizar o controle de acessos por meio de login.

Fonte: elaborado pelo autor.

3.2 ESPECIFICAÇÃO

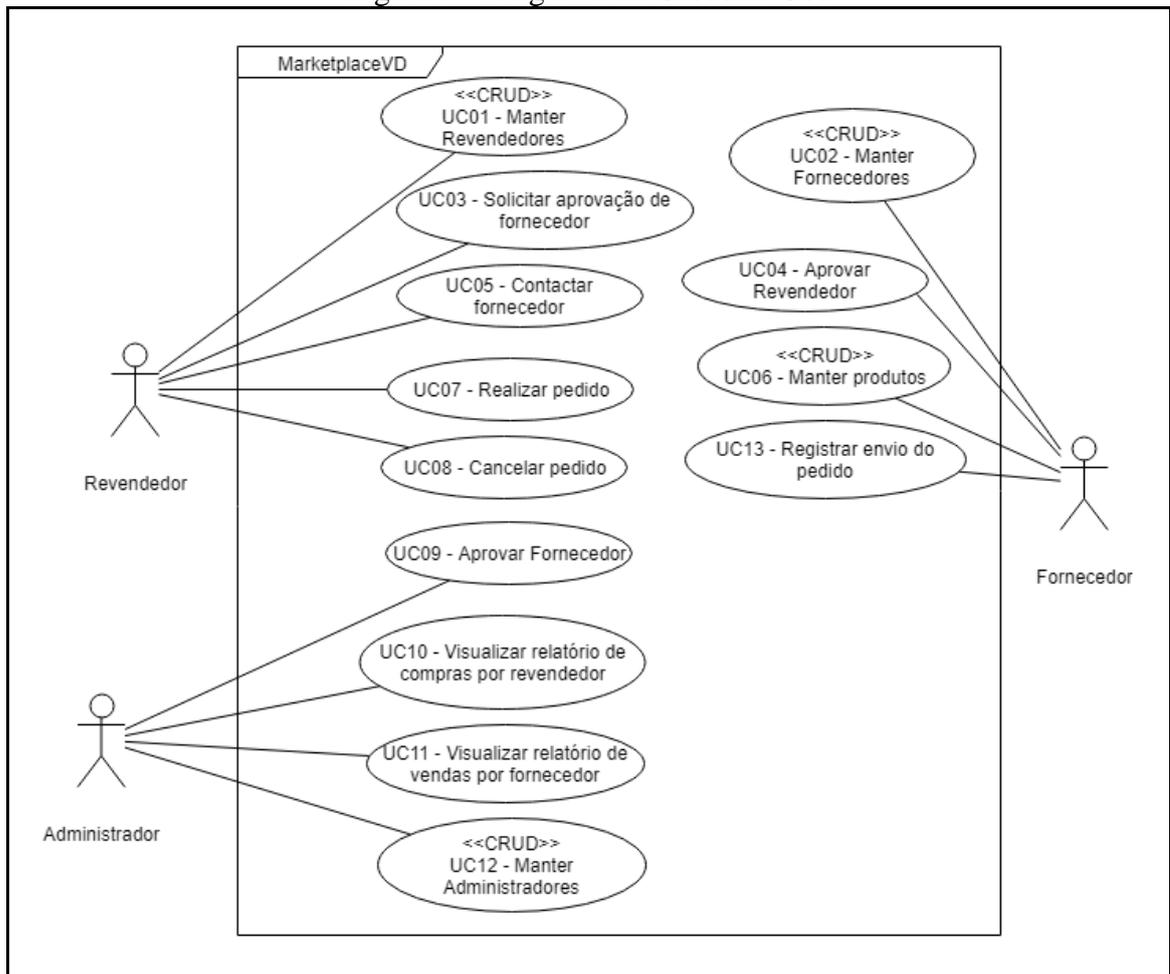
Nesta seção é apresentada a especificação do sistema desenvolvido neste trabalho, contemplando os requisitos funcionais, requisitos não funcionais, diagrama de casos de uso e

diagrama de atividades. Os diagramas foram construídos de acordo com a Unified Modeling Language (UML), utilizando a ferramenta draw.io¹.

3.2.1 Diagrama de Casos de Uso

Na Figura 7 é apresentado o diagrama de casos de uso da plataforma desenvolvida. No diagrama constam os atores presentes no sistema, sendo eles: *revendedor*, *fornecedor* e *administrador*.

Figura 7 - Diagrama de Casos de Uso



Fonte: elaborado pelo autor.

O ator *revendedor* representa o vendedor direto. Ele é o usuário que se cadastra na plataforma e está habilitado a realizar pedidos, cancelar pedidos, entrar em contato com fornecedores e inativar seu próprio usuário.

O ator *fornecedor* representa fabricantes e distribuidores de produtos. Eles podem se cadastrar na plataforma, aprovar o acesso de um *revendedor* aos seus produtos, manter o

¹ <https://www.draw.io/>

catálogo de produtos, receber pedidos, marcar um pedido como enviado e inativar seu próprio usuário.

Há também o ator `administrador`, que tem acesso à funções como aprovar fornecedor, visualizar relatórios, cadastrar outros administradores e desativar qualquer usuário. No Apêndice A são descritos com detalhes os casos de uso UC01 - Cadastrar Revendedor, UC03 - Solicitar aprovação de fornecedor e UC07 - Realizar Pedido.

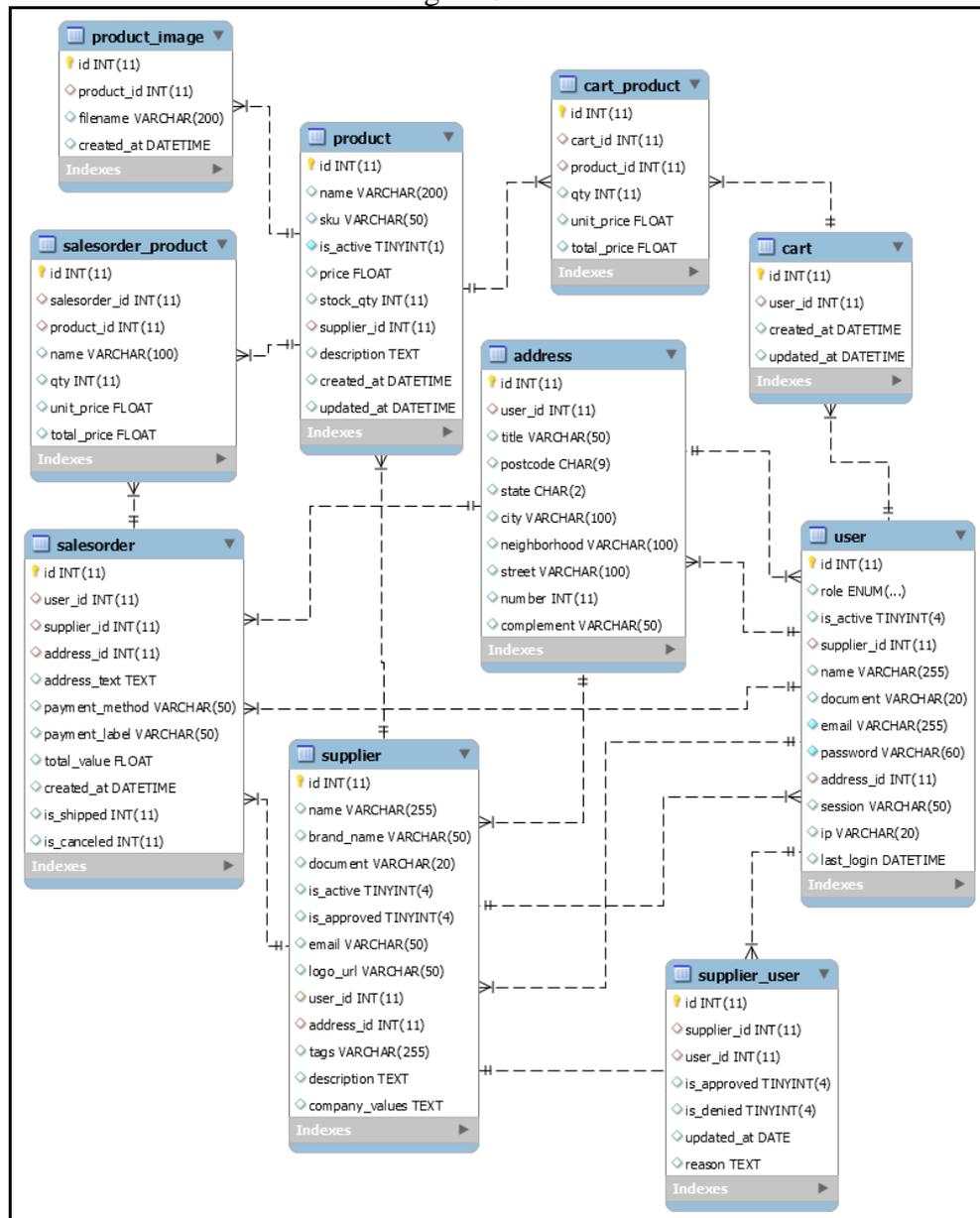
3.2.2 Modelo de Entidade e Relacionamento

Esta seção apresenta o Modelo de Entidade e Relacionamento (MER) da aplicação. Cada entidade é representada por uma tabela no banco de dados MySQL. O MER é apresentado na Figura 8.

A seguir é apresentada uma breve descrição das entidades criadas para o desenvolvimento do sistema:

- a) `user`: entidade que armazena todos os usuários da aplicação;
- b) `supplier`: entidade que armazena os dados dos fornecedores;
- c) `salesorder`: entidade que armazena todos os pedidos realizados na plataforma;
- d) `salesorder_product`: entidade que armazena os itens dos pedidos;
- e) `product`: entidade que armazena os dados dos produtos;
- f) `product_image`: entidade que armazena as imagens dos produtos;
- g) `cart`: entidade que armazena os carrinhos dos clientes;
- h) `cart_product`: entidade que armazena os produtos de um carrinho;
- i) `supplier_user`: entidade que armazena os relacionamentos entre um fornecedor e um usuário revendedor;
- j) `address`: entidade que armazena os endereços dos usuários.

Figura 8 - MER



Fonte: elaborado pelo autor.

3.3 IMPLEMENTAÇÃO

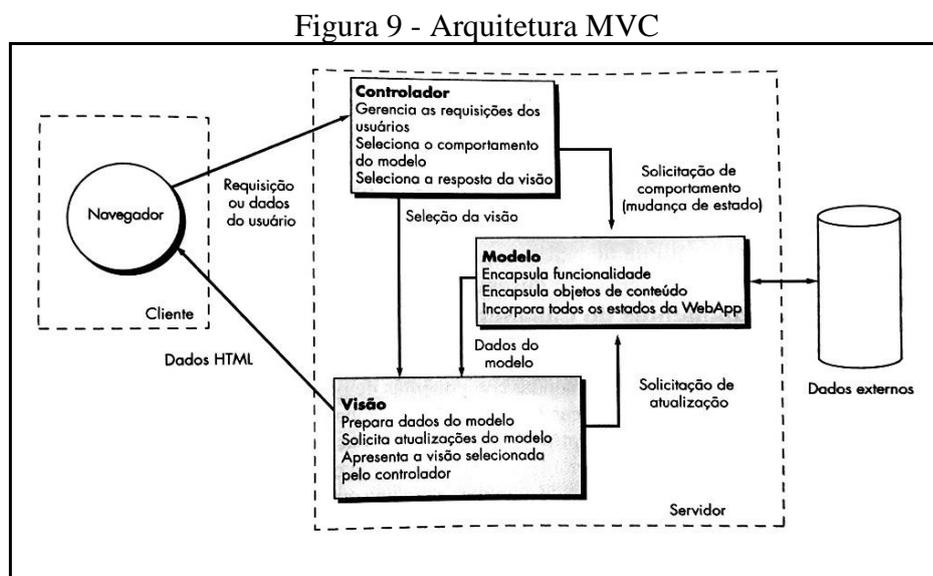
Nesta seção são apresentadas as técnicas e ferramentas utilizadas no trabalho, bem como o desenvolvimento do sistema e a operacionalidade da implementação.

3.3.1 Técnicas e ferramentas utilizadas

A plataforma construída neste trabalho foi codificada na linguagem de programação PHP, aplicando-se a Programação Orientada a Objetos (POO). Para organizar a arquitetura da aplicação foi utilizado o *design* de arquitetura Model-View-Controller (MVC), que separa a aplicação em três camadas:

- a) modelo (*model*): camada que armazena a regra de negócio da aplicação e faz as leituras e escritas de dados;
- b) visão (*view*); camada responsável por exibir dados ao usuário, como páginas HTML;
- c) controlador (*controller*): camada que gerencia as requisições e faz a comunicação entre o modelo e visão.

A Figura 9 apresenta um diagrama exemplificando a utilização da arquitetura MVC em um projeto web.



Fonte: Pressman (2011, p. 349).

A utilização da arquitetura MVC foi facilitada graças ao *framework* FlightPHP, escolhido para a base da aplicação web. O *framework* utilizado possui a estrutura de *controllers* e *views*, ficando a critério do desenvolvedor as soluções utilizadas na construção dos *models*. Neste trabalho, foi utilizado o *framework* RedbeanPHP para fazer o Mapeamento Objeto-Relacional (ORM) com o banco de dados MySQL, o que simplificou o desenvolvimento dos *models*.

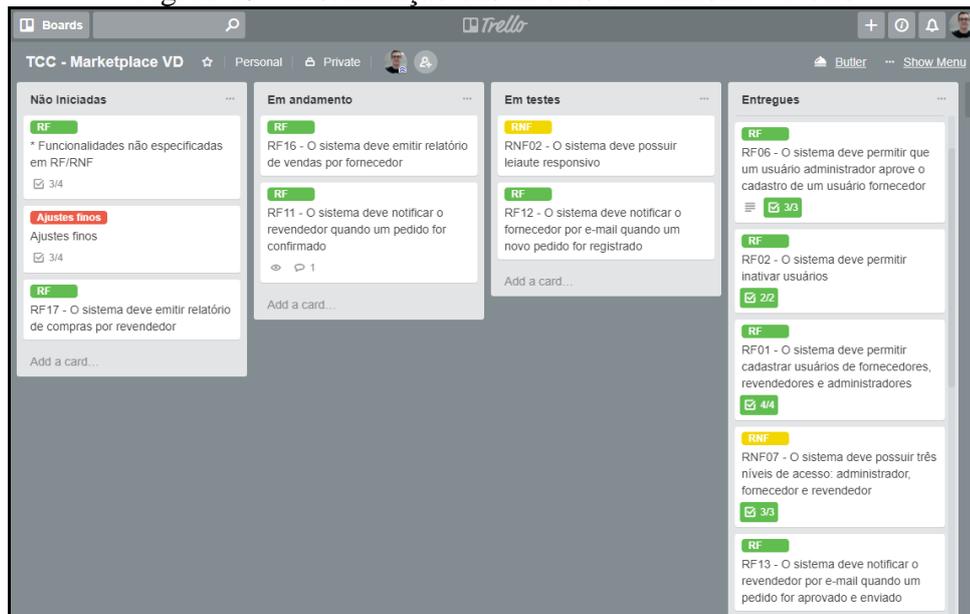
O *frontend* do sistema foi desenvolvido utilizando o Bulma, um *framework* de Cascading Style Sheets (CSS) que contém componentes prontos (como formulários, modais e barras de navegação) para a criação de páginas web responsivas. O *framework* também dispõe de *helpers*, classes de CSS para resolver problemas comuns, como ocultar um elemento ou alterar a cor de um texto para uma lista de cores pré-definidas.

No que diz respeito ao gerenciamento das implementações, para que se mantenha o histórico de alterações do código-fonte foi utilizado um repositório Git hospedado na

plataforma Bitbucket. Para a realização de testes de performance e estabilidade, a plataforma foi hospedada em um servidor *cloud* na DigitalOcean.

Para organização e controle das tarefas de desenvolvimento foram utilizadas técnicas de desenvolvimento ágil. Os RFs e RNFs foram dispostos em cartões na plataforma Trello², que permite visualizar o fluxo do desenvolvimento utilizando o conceito de *kanban*. O fluxo do desenvolvimento foi separado em quatro listas, conforme apresentado na Figura 10.

Figura 10 - Visualização das tarefas em Kanban - Trello



Fonte: elaborado pelo autor.

Cada lista apresentada na Figura 10 indica um estado para uma tarefa, sendo que na primeira lista ficam as tarefas não iniciadas. Na sequência há uma coluna chamada em andamento, para sinalizar as tarefas em desenvolvimento. Na sequência há uma coluna em testes para tarefas sendo testadas. Por último há uma coluna entregues, em que se encontram as tarefas concluídas. Para facilitar a identificação do tipo de cada tarefa foram criadas etiquetas nos cartões. Ao total são três tipos de etiquetas:

- RF, para requisitos funcionais;
- RNF, para requisitos não funcionais;
- ajustes finos, para alterações de leiaute e de usabilidade não previstas em nenhum requisito.

² <https://trello.com/>

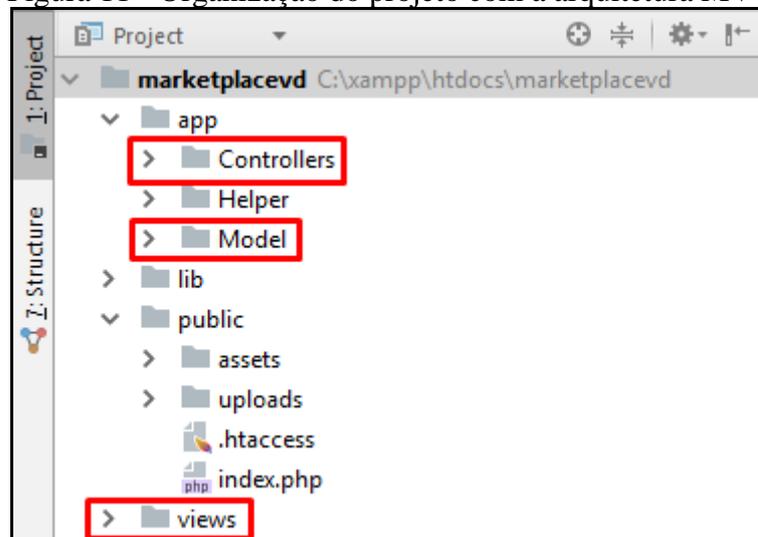
3.3.2 Desenvolvimento

Esta subseção apresenta detalhes da construção do sistema desenvolvido neste trabalho, como a utilização da arquitetura MVC, a estrutura de diretórios do sistema e outros trechos de código relevantes.

3.3.2.1 Implementação da arquitetura MVC

Conforme mencionado na subseção 3.3.1, o sistema desenvolvido neste trabalho foi construído utilizando o *design* de arquitetura MVC. Para a implementação desta arquitetura, a estrutura de diretórios foi construída conforme mostra a Figura 11.

Figura 11 - Organização do projeto com a arquitetura MVC



Fonte: elaborado pelo autor.

No diretório `app`, exibido na Figura 11, estão todas as classes próprias do projeto. Neste diretório há três outros diretórios: `Controllers`, `Helper` e `Model`. Em `Controllers` estão todos os controladores do sistema. No diretório chamado `Helper` ficam classes que podem ser utilizadas para manipular dados e realizar ações em qualquer outro trecho de código, como enviar e-mails e formatar dados para exibição. Por último, no diretório `Model`, ficam os modelos do sistema que manipulam e armazenam dados. Seguindo a estrutura de pastas, há na raiz do projeto outro diretório, este chamado `lib`. Dentro do diretório `lib` estão os *frameworks* utilizados no *backend* do sistema, sendo eles `FlightPHP` e `RedbeanPHP`, conforme citado na seção 3.3.1.

O diretório em que estão os arquivos públicos que podem ser acessados pelos usuários, chama-se `public`. Este deve ser definido como o diretório raiz do sistema na configuração do *webserver*, pois neles estão os arquivos estáticos de *leiaute*, como `CSS` e `JavaScript`, além de imagens enviadas por usuários, como imagens de produtos e logotipos de fornecedores.

Isolando o diretório `public` desta forma evita-se que usuários mal-intencionados acessem arquivos críticos do sistema, uma vez que estes não são acessíveis via Uniform Resource Locator (URL), apenas via *backend*. O arquivo responsável por carregar as classes e *frameworks* do sistema é o `index.php`, também dentro do diretório `public`. Além das importações, neste arquivo são definidas as URLs, métodos HTTP aceitos e o *callback* que irá processar cada requisição, conforme exemplificado no Quadro 4.

Quadro 4 - Definição de rotas do FlightPHP

```
\Flight::route( pattern: 'GET /fornecedor/produtos', ['\Controllers\Supplier', 'productListAction']);
```

Fonte: elaborado pelo autor.

Uma rota é definida a partir de dois parâmetros, sendo que o primeiro é uma concatenação do método HTTP e o caminho da requisição. O segundo parâmetro para a definição de uma rota é um *callback*, que indica a classe do controlador e qual método será chamado para processar a requisição. No Quadro 5 é apresentado o código-fonte de um controlador e em seguida é explicado como ele faz a conexão entre um modelo e uma visão.

Quadro 5 - Controlador da lista de produtos

```

1  <?php
2  namespace Controllers;
3  class Supplier {
4
5      public static function productListAction() {
6          \Flight::session()->requireLogin();
7
8          $user = \Flight::session()->getUser();
9
10         $productModel = new \Model\Product();
11         $productList = $productModel->getActiveProductsBySupplier($user->supplier);
12
13         \Flight::renderLayout('supplier/my_products', [
14             'productModel' => $productModel,
15             'productList'  => $productList
16         ]);
17     }

```

Fonte: elaborado pelo autor.

O controlador apresentado no Quadro 5 é o controlador `Supplier` que contém os *callbacks* de todas as rotas que envolvem um fornecedor. Primeiramente é chamado o método `requireLogin` (linha 6) que verifica se o usuário que está acessando a página está autenticado, e se não estiver, redireciona-o para a página de *login*. Em seguida, o objeto do usuário que está autenticado na sessão é atribuído à variável `user` (linha 8). Nas linhas seguintes, instancia-se o modelo `Product` (linha 10) e é chamado o método `getActiveProductsBySupplier` (linha 11) que lista todos os produtos ativos de um determinado fornecedor. Seu retorno é atribuído à variável `productList`. Na linha 13 do controlador `Supplier` é utilizado o método `renderLayout` que irá renderizar o *template* de uma visão. Nele são fornecidos dois parâmetros:

- a) o caminho do arquivo que contém a visão desejada;

- b) um *array* contendo dados que poderão ser acessados na visão (neste caso, a lista de produtos).

Todos os arquivos que contém as visões do sistema estão no diretório `views`, na raiz do projeto, conforme visto na Figura 11. As rotas do sistema são agrupadas por módulos de acordo com o processo que desempenham no sistema. O Quadro 6 apresenta as rotas que fazem parte do fluxo da realização de pedidos.

Quadro 6 - Rotas do fluxo de realização de pedidos

```

52 \Flight::route( pattern: 'GET /cart', ['\Controllers\Cart', 'indexAction']);
53 \Flight::route( pattern: 'GET /cart/empty', ['\Controllers\Cart', 'emptyCartAction']);
54 \Flight::route( pattern: 'POST /cart/add', ['\Controllers\Cart', 'addToCartAction']);
55 \Flight::route( pattern: 'POST /cart/update', ['\Controllers\Cart', 'updateCartAction']);
56
57 \Flight::route( pattern: 'GET /checkout', ['\Controllers\Checkout', 'indexAction']);
58 \Flight::route( pattern: 'POST /checkout/save', ['\Controllers\Checkout', 'saveAction']);
59 \Flight::route( pattern: 'GET /checkout/success', ['\Controllers\Checkout', 'successAction']);

```

Fonte: elaborado pelo autor.

As rotas do fluxo de pedidos foram divididas em dois módulos: `cart` e `checkout`. O módulo `cart` é responsável pelas operações de carrinho, como adicionar um item ao carrinho, atualizar as quantidades ou excluir todos os itens. O módulo `checkout` contém a tela de revisão do pedido, o método que persiste os pedidos no Sistema de Gerenciamento de Banco de Dados (SGBD) e a tela de confirmação de pedido realizado. O Quadro 7 apresenta as rotas responsáveis pelas funções administrativas no sistema.

Quadro 7 - Rotas das funções administrativas

```

69 \Flight::route( pattern: 'GET /admin/dashboard', ['\Controllers\Admin', 'dashboardAction']);
70 \Flight::route( pattern: 'GET /admin/users', ['\Controllers\Admin', 'listUsersAction']);
71 \Flight::route( pattern: 'GET /admin/users/new', ['\Controllers\Admin', 'newUserAction']);
72 \Flight::route( pattern: 'POST /admin/users/save', ['\Controllers\Admin', 'saveUserAction']);
73 \Flight::route( pattern: 'GET /admin/users/edit/@id:[0-9]+', ['\Controllers\Admin', 'editUserAction']);
74 \Flight::route( pattern: 'GET /admin/suppliers', ['\Controllers\Admin', 'listSuppliersAction']);
75 \Flight::route( pattern: 'GET /admin/suppliers/edit/@id:[0-9]+', ['\Controllers\Admin', 'editSupplierAction']);
76 \Flight::route( pattern: 'POST /admin/suppliers/save', ['\Controllers\Admin', 'saveSupplierAction']);

```

Fonte: elaborado pelo autor.

As funções administrativas são divididas em três módulos: `painel do administrador`, que exibe estatísticas do sistema; `manutenção de usuários` (linhas 70 a 73) e `manutenção de fornecedores` (linhas 74 a 76).

3.3.2.2 Notificações transacionais

Em vários fluxos da aplicação é necessário o envio de mensagens de e-mail para notificar o usuário de uma ação realizada, como a confirmação de um pedido ou cancelamento do mesmo. Estes e-mails são chamados de e-mails transacionais. Para realizar o envio dos e-

mails transacionais aos usuários foi utilizado o serviço Mailgun³, que fornece um conjunto de APIs que tornam possível o envio de e-mails sem que seja necessário configurar um servidor de e-mails próprio. O serviço permite o envio de até 10.000 mensagens gratuitamente.

A integração com a API do Mailgun para envio de e-mails pode ser feita em poucas linhas de código. Para enviar um e-mail por meio do serviço basta que a aplicação realize uma requisição POST ao *endpoint* `api.mailgun.net/v3/domínio.com.br/messages`, informando uma chave de autenticação fornecida pelo serviço, o destinatário, o assunto e a mensagem do e-mail. Neste trabalho, a requisição foi implementada utilizando a biblioteca cURL. O código que realiza a requisição é apresentado no Quadro 8.

Quadro 8 - Requisição POST ao Mailgun

```

26     $apiUrl = 'https://' . $this->_mailgunKey . '@api.mailgun.net/v3/' . $this->_mailgunDomain;
27
28     $ch = curl_init();
29     $timeout = 30;
30     curl_setopt ($ch, option: CURLOPT_URL, value: $apiUrl . '/messages');
31     curl_setopt ($ch, option: CURLOPT_RETURNTRANSFER, value: 1);
32     curl_setopt ($ch, option: CURLOPT_CONNECTTIMEOUT, $timeout);
33     curl_setopt ($ch, option: CURLOPT_POST, value: true);
34     curl_setopt ($ch, option: CURLOPT_POSTFIELDS, http_build_query($fields));
35     $file_contents = curl_exec($ch);
36     curl_close($ch);

```

Fonte: elaborado pelo autor.

Para iniciar uma operação utilizando a biblioteca cURL é necessário chamar a função `curl_init` (linha 28). Em seguida são configurados os parâmetros da requisição, utilizando a função `curl_setopt` e fornecendo como parâmetros qual opção está sendo configurada e seu respectivo valor. É definido o *endpoint* que irá receber a requisição (linha 30), o método da requisição, que neste caso é POST (linha 33) e são enviados os parâmetros da requisição (destinatário, assunto e mensagem), que estão na variável `fields` (linha 34).

O método que faz o envio da requisição fica em uma classe `Helper` chamada `Mail`. O envio de um e-mail é exemplificado na Quadro 9, em que é apresentado o método que é executado quando um fornecedor marca um pedido como enviado.

³ <https://www.mailgun.com/>

Quadro 9 - Notificação de pedido enviado

```

36 public static function shipOrderAction($id){
37     \Flight::session()->requireLogin();
38
39     $orderModel = new \Model\Salesorder();
40     $order = $orderModel->load($id);
41
42     $order->is_shipped = 1;
43     \R::store($order);
44
45     $message = "Seu pedido de número <strong>#" . $order->id . "</strong> foi enviado.";
46     $subject = "Seu pedido #" . $order->id . " de " . $order->supplier->name . " foi enviado";
47
48     $orderTemplate = \Flight::view()->fetch( file: 'mail/order_update', [
49         'message' => $message,
50         'subject' => $subject,
51         'order' => $order
52     ]);
53
54     $mail = new \Helper\Mail();
55     $mail->sendEmail($order->user->email, $subject, $message, replyTo: null, $orderTemplate);
56
57     \Flight::session()->addSuccess("O pedido #" . $order->id . " foi marcado como enviado.");
58     \Flight::redirectReferrer();
59
60 }

```

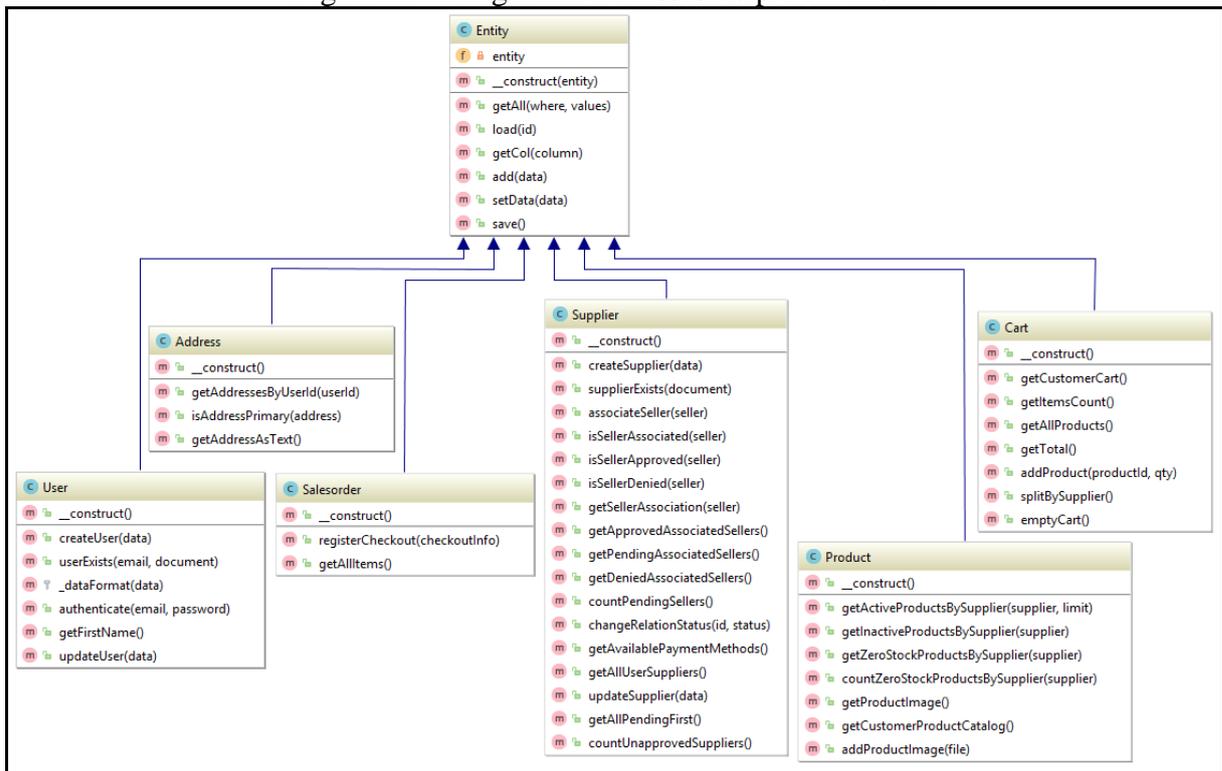
Fonte: elaborado pelo autor.

Ao marcar um pedido como enviado, o sistema atualiza o campo `is_shipped` do pedido no SGBD (linha 43). Em seguida é definida a mensagem que será enviada (linha 45) e o assunto do e-mail (linha 46). Para obter o *template* da mensagem que será enviada por e-mail a aplicação utiliza uma *view*, em que são definidos os parâmetros de mensagem, assunto e o pedido em questão (linhas 48 a 52) e a *view* retorna o código HTML com o conteúdo do e-mail. Em seguida é criada uma instância do `Helper Mail` (linha 54) e posteriormente é chamado o método `sendEmail` (linha 55) que enviará a requisição ao Mailgun para agendar o envio do e-mail.

3.3.2.3 Modelo de Dados

Há três principais entidades no sistema: `user`, que armazena todos os usuários do sistema; `supplier`, que armazena as informações dos fornecedores, como nome da marca, endereço e valores da empresa; e `salesorder`, que armazena os pedidos, forma de pagamento e endereço. Há também entidades auxiliares como `address`, que armazena todos os endereços de clientes, fornecedores e pedidos; `product`, que contém todos os produtos, e `cart`, que mantém os carrinhos dos clientes. Além das entidades auxiliares, existem tabelas que fazem o relacionamento entre uma ou mais entidades, como `supplier_user`, que mantém as associações entre revendedores e fornecedores, e `saleorder_product`, que armazena os itens dos pedidos, suas respectivas quantidades e preços. A Figura 12 apresenta um diagrama de classes do pacote `Model` da aplicação.

Figura 12 - Diagrama de classes do pacote Model



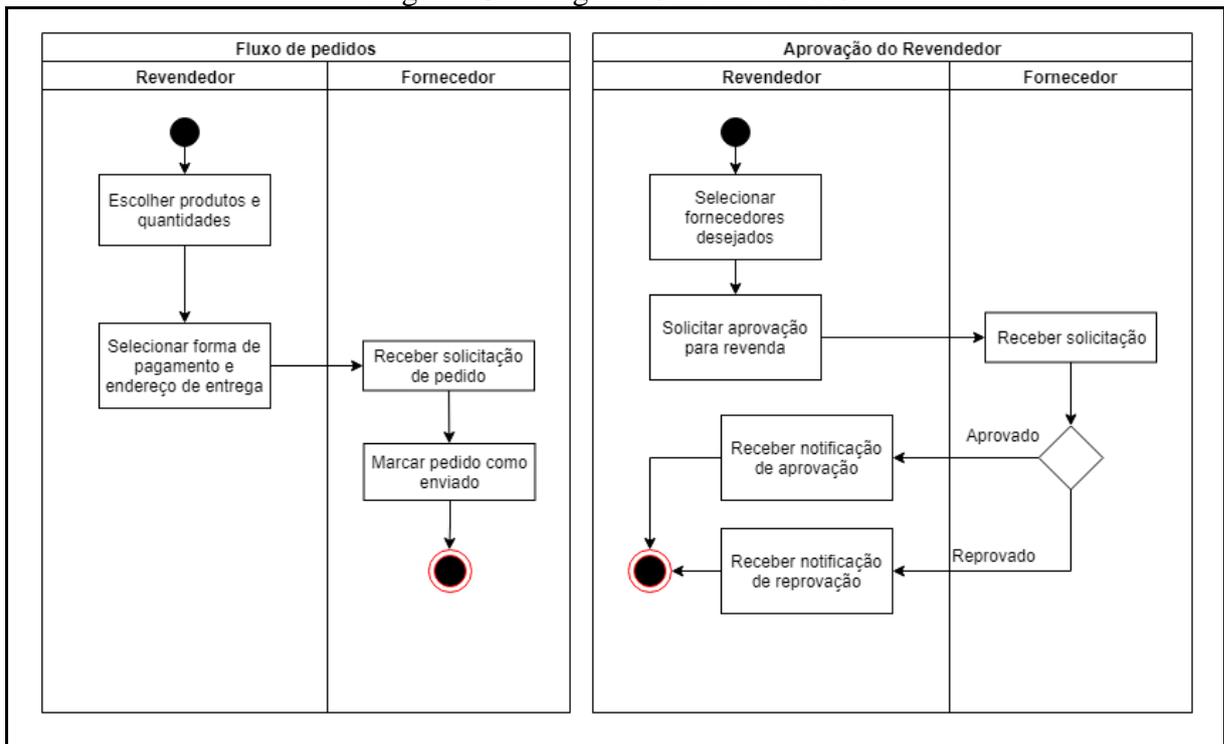
Fonte: elaborado pelo autor.

A classe `Entity` implementa as funções da ORM RedbeanPHP. Ao estender a classe `Entity`, as demais classes – que são entidades – herdam métodos pré-definidos para realizar as operações no SGBD. Os métodos pré-definidos são: `getAll`, que faz um *select* em uma tabela trazendo uma lista de registros; `load`, que busca um objeto a partir de sua chave primária; `getCol`, que busca todos os valores de uma coluna em determinada tabela; `add` que insere um novo registro; `setData`, que define novos valores para as propriedades de um objeto; e `save`, que persiste no banco de dados as alterações feitas utilizando o `setData`.

3.3.3 Operacionalidade da implementação

Esta subseção apresenta a operacionalidade da implementação do sistema. O sistema possui três perfis de acesso: `revendedor`, `fornecedor` e `administrador`. Cada perfil possui funcionalidades distintas e a presença dos três atores é necessária para o funcionamento correto da plataforma. Para representar a operacionalidade da implementação são demonstrados o fluxo de cadastramento de um revendedor e a realização de um pedido (Figura 13).

Figura 13 - Diagramas de Atividades



Fonte: elaborado pelo autor.

O primeiro diagrama representa o fluxo de pedidos. O fluxo tem início com o revendedor escolhendo os produtos que deseja comprar e suas respectivas quantidades. Em seguida, o revendedor seleciona a forma de pagamento e o endereço de entrega. O fornecedor é notificado de um novo pedido e o fluxo termina quando o pedido é marcado como enviado.

O segundo diagrama apresenta o fluxo de aprovação de um revendedor e tem início com o revendedor selecionando os fornecedores desejados. Em seguida, o sistema notifica o fornecedor da nova solicitação e decide se irá aprovar ou reprovar o revendedor. Ambas as escolhas geram uma notificação ao revendedor, encerrando o fluxo de aprovação.

A tela inicial da aplicação é a tela de identificação, apresentada na Figura 14. Nesta tela há um formulário de *login* para que usuários já cadastrados possam autenticar-se e passar a utilizar o sistema. Além do formulário de *login*, na mesma tela consta uma lista dos benefícios que o usuário adquire ao se registrar na plataforma, localizado no lado direito da tela, seguindo padrões de usabilidade. Logo abaixo dela, há um botão chamado *Quero começar a vender*, que direciona o usuário revendedor a uma tela de cadastro.

Figura 14 - Tela de Identificação da plataforma

MarketplaceVD Login

Identifique-se

Faça login para ter acesso a todas as funcionalidades do MarketplaceVD

Já tenho cadastro

Informe seu e-mail e senha para continuar.

E-mail

Senha

[Esqueci a senha](#)

Login

Quero me cadastrar

O **MarketplaceVD** é uma plataforma que ajuda vendedores diretos a encontrar vários fornecedores de produtos para revenda. Cadastrado, você poderá:

- Conhecer novos fornecedores
- Aumentar seu catálogo de produtos
- Comprar e repetir pedidos com agilidade

Quero começar a vender

É um fabricante ou distribuidor de produtos? [Clique aqui e faça seu cadastro de fornecedor.](#)

Fonte: elaborado pelo autor.

Na tela de Cadastro de revendedor, apresentada na Figura 15, há um formulário com quatro campos obrigatórios, solicitando dados do usuário que está prestes a se cadastrar no sistema. Como dados pessoais, são solicitados o nome completo e CPF do usuário. Para permitir que o usuário faça *login* na plataforma são solicitados dois dados de acesso: e-mail e senha. Há também um campo do tipo *checkbox*, que só permite concluir o cadastro caso o usuário esteja de acordo com os termos e condições de uso da plataforma.

Figura 15 - Cadastro de Revendedor na plataforma

MarketplaceVD Login

Cadastro de Revendedor

Aumente sua renda conhecendo fornecedores e revendendo seus produtos!

Nome Completo

CPF

E-mail

Senha

Aceito os [termos e condições de uso.](#)

Cadastrar

Fonte: elaborado pelo autor.

Ao enviar o formulário de cadastro são realizadas validações de *frontend* para garantir que o usuário preencheu todos os campos obrigatórios e aceitou os termos de uso da plataforma. Passadas as validações de *frontend*, os dados do formulário são enviados ao *backend* da plataforma, que valida se já existe um usuário com o e-mail ou CPF informados. Caso já exista um usuário cadastrado, o sistema exibe uma mensagem de erro e retorna o cliente à tela de identificação. Caso o cadastro seja realizado com sucesso, o usuário é autenticado e direcionado ao painel do revendedor, apresentado na Figura 16.

Figura 16 - Painel do revendedor

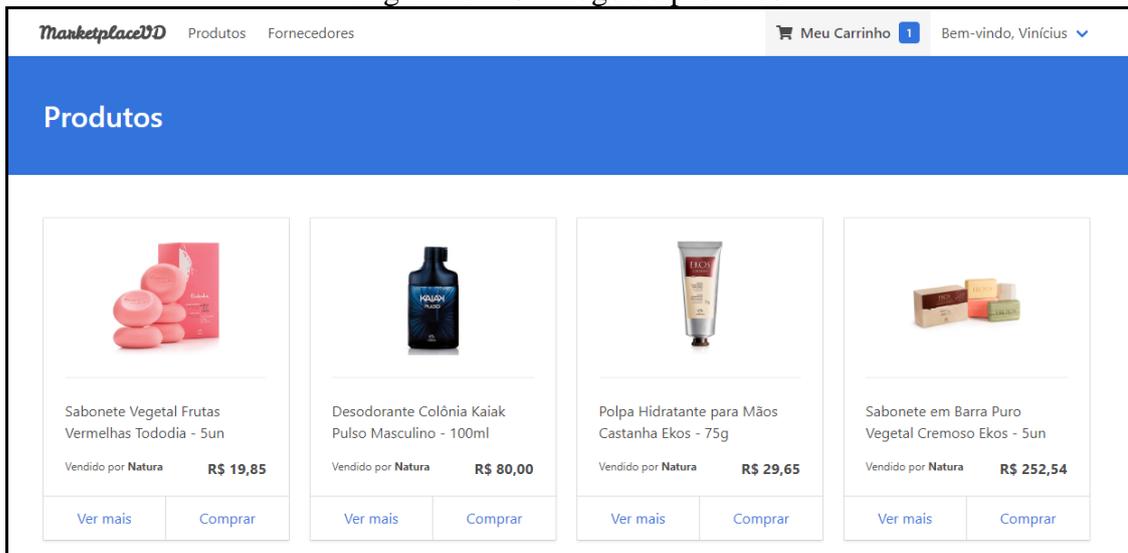


Fonte: elaborado pelo autor.

No painel do revendedor está um resumo de todas as atividades realizadas pelo usuário na plataforma. No resumo da conta são apresentados quatro blocos de informações: o valor gasto e a quantidade de pedidos realizados nos últimos 30 dias; o valor gasto e todos os pedidos realizados desde seu cadastro; uma lista com os produtos mais comprados por este usuário e uma lista com os últimos pedidos realizados. Do lado esquerdo da tela há um menu que dá acesso a diversos recursos do sistema, como a lista de pedidos do usuário, o cadastro de endereços e um formulário para entrar em contato com fornecedores.

No topo esquerdo do site, ao lado da logo, há dois links: o primeiro, para o catálogo de produtos; o segundo, para a listagem de fornecedores. A tela de catálogo de produtos é apresentada na Figura 17.

Figura 17 - Catálogo de produtos



Fonte: elaborado pelo autor.

No catálogo de produtos são listados produtos de todos os fornecedores associados ao revendedor que estão em estoque. Para cada produto é exibida uma imagem, o nome do produto, o nome do fornecedor e o preço. Há ainda, para cada produto, dois botões: `ver mais`, que exibe detalhes do produto, e `comprar`, que exibe um modal solicitando a quantidade desejada do produto. Este modal está representado na Figura 18.

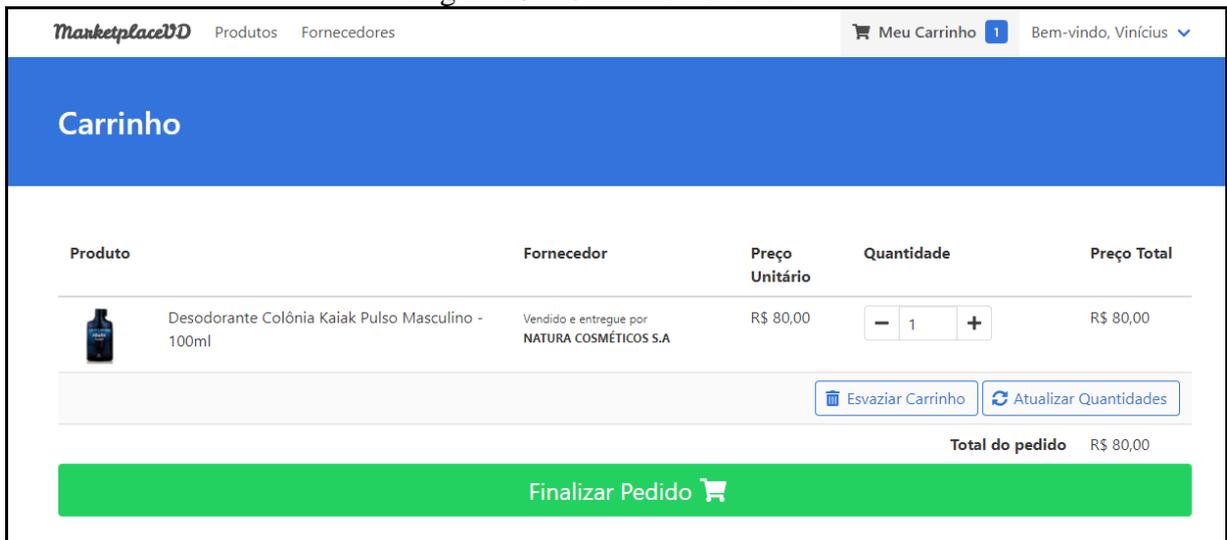
Figura 18 - Modal para adicionar produto ao carrinho



Fonte: elaborado pelo autor.

Ao adicionar um produto ao carrinho, o contador de produtos que fica no menu `meu carrinho` localizado no topo direito da página é atualizado com a quantidade total de produtos no carrinho do cliente. O usuário então pode escolher entre adicionar mais produtos ou ir para a página do carrinho (Figura 19).

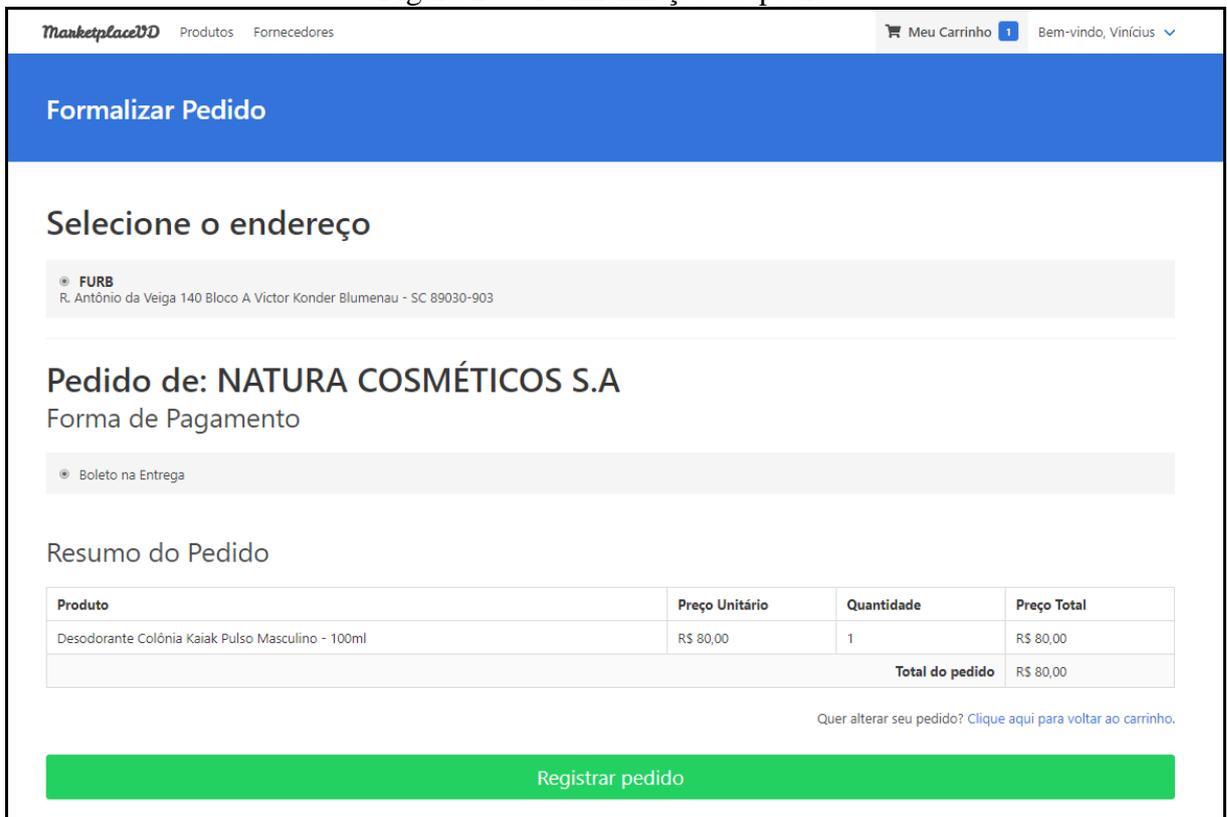
Figura 19 - Carrinho do cliente



Fonte: elaborado pelo autor.

Na página do carrinho é exibida uma lista de produtos que estão no carrinho do cliente, seu respectivo fornecedor, preço unitário, quantidade selecionada e preço total. Ainda nesta tela é possível atualizar a quantidade desejada de cada produto ou esvaziar o carrinho, isto é, excluir todos os itens. Abaixo da listagem de itens fica o valor total do pedido, em reais, e um botão finalizar pedido, que irá direcionar o cliente para a tela de *checkout* (Figura 20).

Figura 20 – Formalização do pedido



Fonte: elaborado pelo autor.

A tela de *checkout* permite que o revendedor selecione o endereço em que deseja receber os produtos, além da forma de pagamento. Ainda, antes de registrar o pedido, o revendedor pode revisar o pedido na seção resumo do pedido, e se houver necessidade, voltar ao carrinho para fazer as devidas alterações. Se não houver alterações, o usuário pode registrar o pedido. A partir desta ação, tanto o revendedor quanto o fornecedor são notificados por e-mail com a confirmação do pedido e o usuário é direcionado para sua listagem de pedidos, em que é possível consultar os detalhes de um pedido e seu estado atual (Figura 21).

Figura 21 - Detalhes do pedido

Pedido #31

Este pedido está pendente. Pode ser enviado ou cancelado.

Data do pedido 29/06/2018 00:41	Valor Total R\$ 80,00	Pagamento Boleto na Entrega
---	---------------------------------	---------------------------------------

Endereço

R. Antônio da Veiga 140
Bloco A
Victor Konder
Blumenau - SC
89030-903

Itens do pedido

Produto	Preço UN	Qty	Preço Total
Desodorante Colônia Kaiak Pulso Masculino - 100ml	R\$ 80,00	1	R\$ 80,00

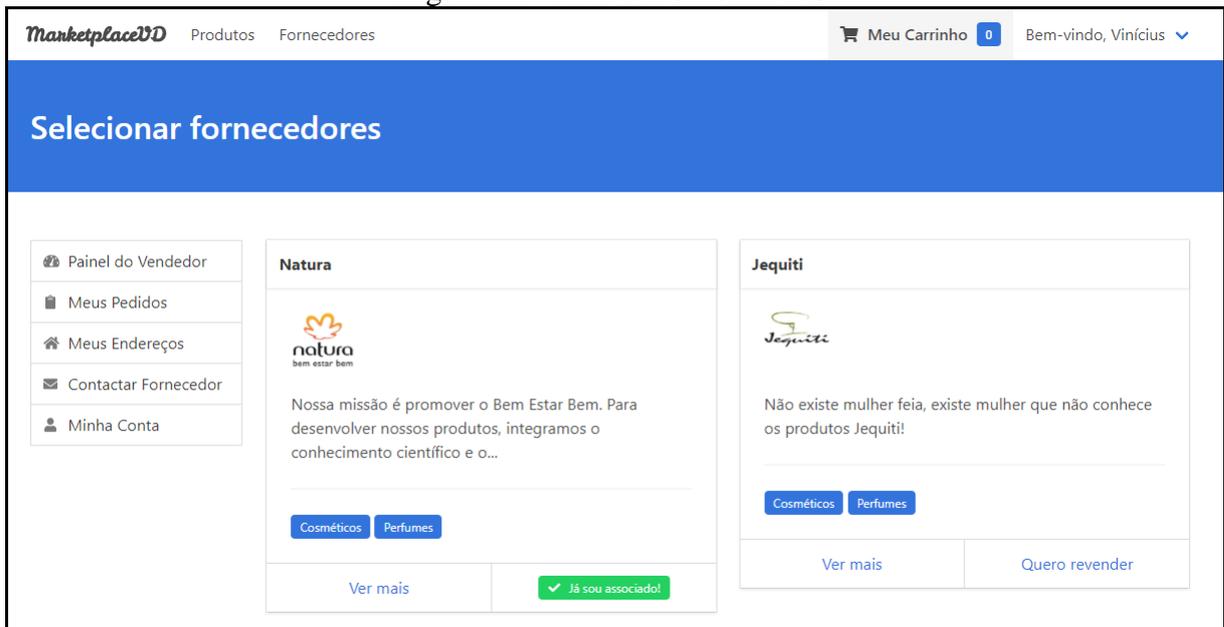
Cancelar Pedido

Tive problemas com o pedido

Fonte: elaborado pelo autor.

Na visualização de detalhes de um pedido, o revendedor pode ver os itens comprados e cancelar seu próprio pedido ou enviar uma mensagem ao fornecedor. Ao lado do link produtos no topo do site, há um link para a página de fornecedores, que permite que o revendedor veja todos os seus fornecedores ativos e candidate-se para representar novos fornecedores. A página de fornecedores está apresentada na Figura 22.

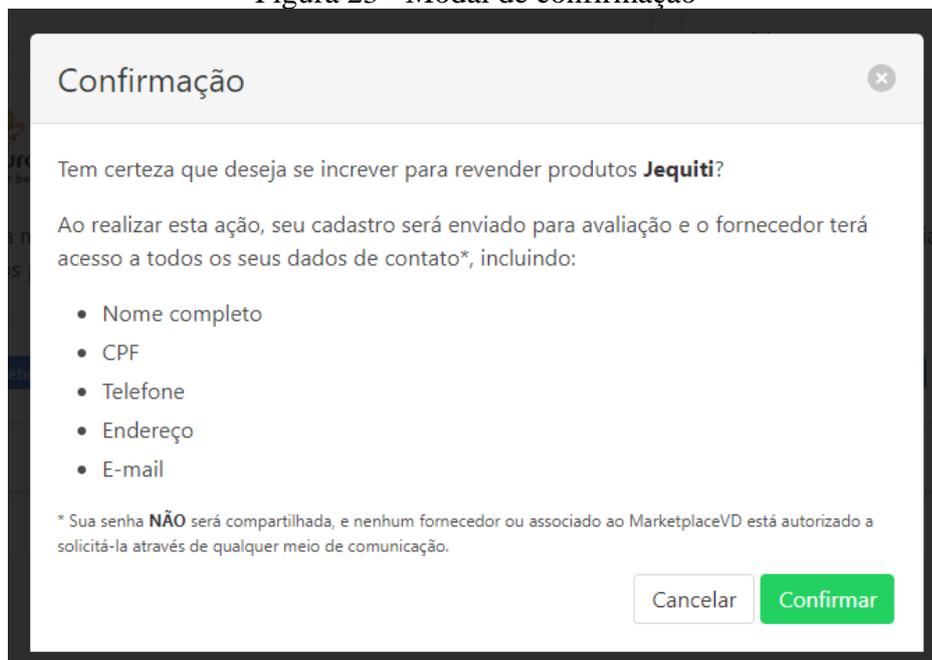
Figura 22 - Tela de Fornecedores



Fonte: elaborado pelo autor.

Na listagem de fornecedores são exibidos para cada fornecedor o nome, o logotipo, uma breve descrição e *tags* com os tipos de produtos vendidos. Há um botão para ver mais informações e um botão quero revender, apenas nos fornecedores os quais o revendedor ainda não é associado. Ao clicar no botão quero revender, o sistema exibe um modal de confirmação. Este modal está representado na Figura 23.

Figura 23 - Modal de confirmação



Fonte: elaborado pelo autor.

O modal de confirmação apresenta quais dados do revendedor serão enviados ao fornecedor para análise e reforça que em nenhum momento será compartilhada a senha do

usuário. Ao confirmar, as informações são enviadas ao fornecedor, que define se o revendedor está apto ou não para revender seus produtos e representar sua marca. Se o cadastro for aprovado, os produtos deste fornecedor passam a aparecer na listagem de produtos apresentada na Figura 17 e o revendedor está habilitado para realizar pedidos destes produtos.

3.4 RESULTADOS E DISCUSSÕES

Por meio dos trabalhos correlatos foi possível identificar as funcionalidades de cada ferramenta e como juntas elas são importantes para o desenvolvimento da solução desenvolvida. O MercadoLivre e Elo7, sendo *marketplaces* consolidados no mercado, serviram como base para a análise das regras de negócio em todas as etapas que envolvem as funcionalidades de um *e-commerce*, como a gestão de produtos e realização de pedidos. Já do GeraSGI foram analisadas principalmente o fluxo de cadastramento de um novo revendedor e o fluxo de um pedido gerado por um vendedor direto.

O diferencial da plataforma desenvolvida é que ela une o propósito das três ferramentas analisadas, proporcionando um *marketplace* especialmente voltado aos vendedores diretos e centralizando todos os processos, desde o cadastro de um novo revendedor até a realização da venda. As características são comparadas no Quadro 10.

Quadro 10 - Comparação do trabalho desenvolvido com os correlatos

Características/Trabalhos relacionados	Elo7 (2017)	MercadoLivre (2018)	Gera (2017a)	Plataforma desenvolvida
Interface entre fornecedor e vendedor direto			X	X
Venda em grandes quantidades (atacado)		X	X	X
Vários fornecedores (<i>marketplace</i>)	X	X		X
Clientes avaliam produtos e reputação do fornecedor	X	X		
Captura de pagamentos	X	X		
Vitrine de produtos (catálogo <i>online</i>)	X	X	X	X
Permite candidatura de revendedores			X	X

Fonte: elaborado pelo autor.

No comparativo entre a plataforma desenvolvida neste trabalho e os trabalhos correlatos (Quadro 10), percebe-se que duas características significativas de outros trabalhos não foram implementadas: as avaliações de produtos e captura de pagamentos. Ambas as funcionalidades aumentariam o engajamento dos usuários com a plataforma pois estabelecem uma relação de maior confiança entre as partes. Embora tais funcionalidades não estavam previstas no escopo do trabalho, ambas podem ser implementadas em versões futuras.

Um dos objetivos específicos do trabalho era construir uma plataforma seguindo padrões de usabilidade para a web. Foram avaliados os seguintes pontos:

- a) desempenho do sistema;
- b) compatibilidade entre navegadores.

O desempenho do sistema foi avaliado por meio do PageSpeed Insights⁴, um serviço da Google que analisa o desempenho de um *website* a partir de várias métricas, dá uma nota de 0 a 100 e fornece sugestões de otimização para melhorar a performance. O resultado da avaliação do desempenho do *website* responsivo é apresentada na Figura 24.

Figura 24 - Teste de desempenho do responsivo

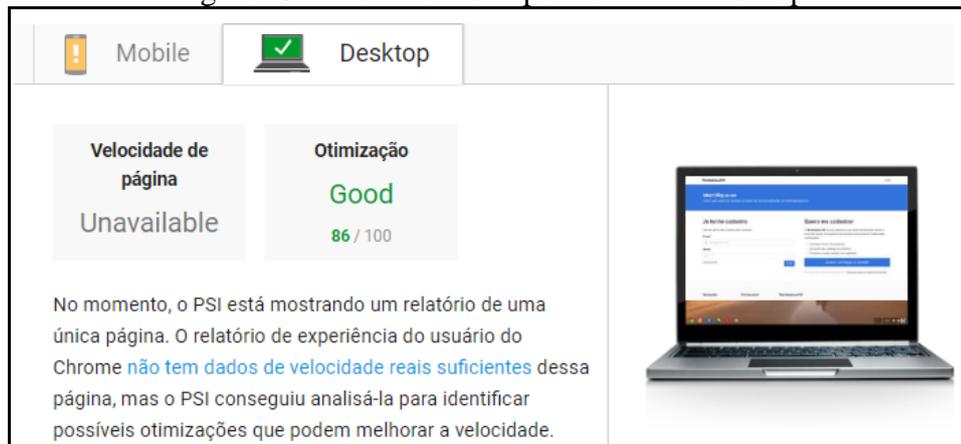


Fonte: elaborado pelo autor.

Como visto na Figura 24, o responsivo recebeu uma nota média de desempenho, atingindo 62% do esperado pela Google. As otimizações sugeridas neste caso são relacionadas ao *cache* de arquivos estativos, e seriam aplicadas apenas em ambiente de produção, pois estão relacionadas a configurações de servidor, o que não estava previsto no escopo deste trabalho. Na Figura 25 é apresentada a avaliação de desempenho da plataforma em um navegador *desktop*.

⁴ <https://developers.google.com/speed/pagespeed/insights/>

Figura 25 - Teste de desempenho em um desktop

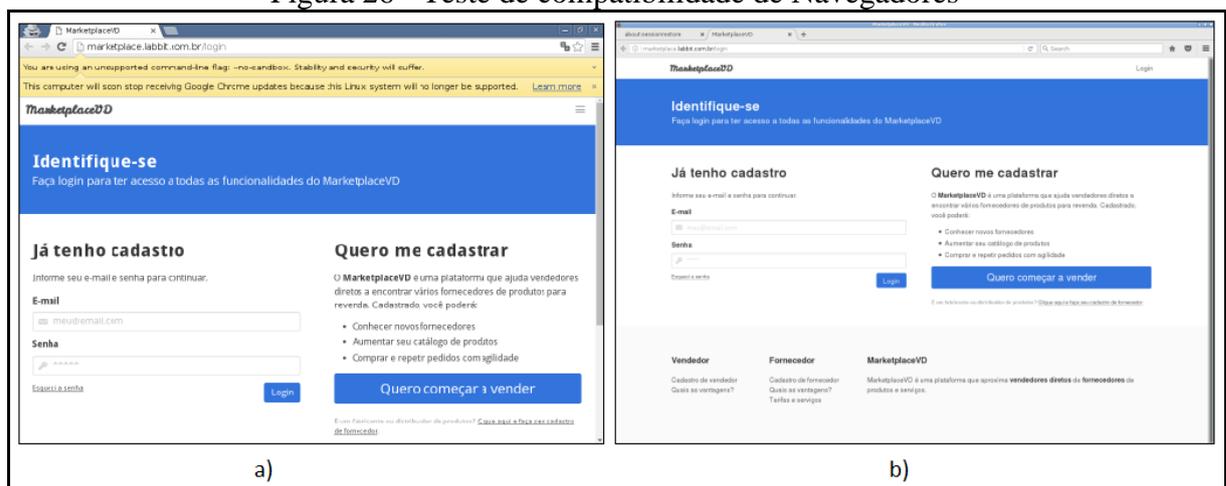


Fonte: elaborado pelo autor.

A avaliação do nível de otimização da plataforma em um *desktop* foi boa, atingindo 86% das configurações de otimização esperadas. As otimizações sugeridas para a plataforma foram as mesmas sugeridas na avaliação em dispositivos móveis, porém com pesos diferentes; em um dispositivo móvel, o *cache* dos arquivos é mais importante que em um dispositivo *desktop*, uma vez que o usuário pode estar utilizando uma conexão de dados móveis ou estar em uma rede lenta.

Para avaliar a compatibilidade entre navegadores foi utilizado o serviço BrowserShots⁵. O BrowserShots gera capturas de tela de um website a partir de diversos navegadores e sistemas operacionais diferentes. O resultado está exemplificado na Figura 26.

Figura 26 - Teste de compatibilidade de Navegadores



Fonte: elaborado pelo autor.

O resultado da avaliação de compatibilidade entre navegadores foi satisfatório. O layout do sistema se mostrou compatível com os principais navegadores mesmo em

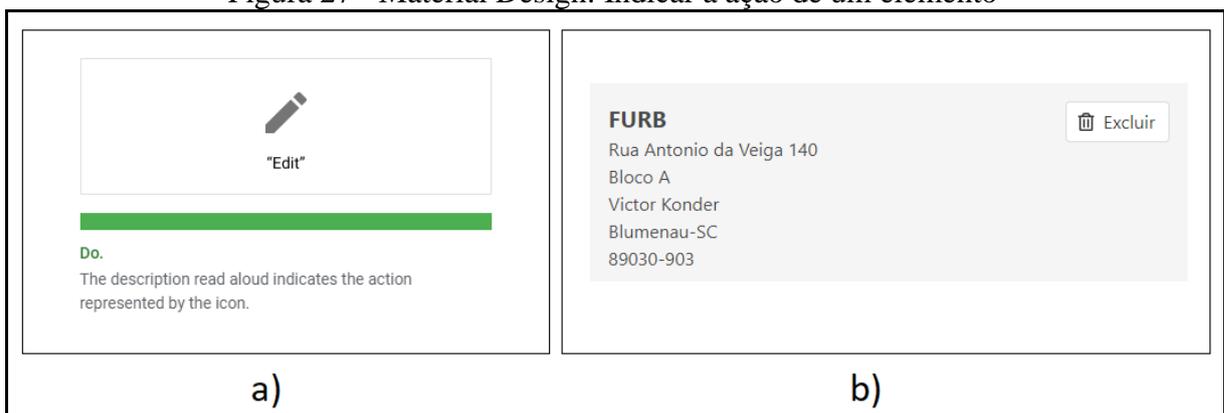
⁵ <http://browsershots.org/>

navegadores desatualizados, como o Chrome versão 48 (Figura 26 a), lançado em fevereiro de 2016, e o Firefox versão 50 (Figura 26 b), lançado em novembro de 2016.

Para validar a ergonomia e usabilidade do leiaute foram consideradas algumas *guidelines* do Material Design (GOOGLE, 2018a), sendo elas um conjunto de diretrizes para desenvolvimento de interfaces web e *mobile*. Os tópicos considerados são guias aplicáveis a botões, *labels*, títulos e forma de apresentação dos elementos em uma interface. Nos parágrafos seguintes serão comparadas as recomendações do Material Design com o que foi implementado no trabalho.

A Figura 27 apresenta guias para a utilização de botões de ação. De acordo com o manual do Material Design, o *label* de um botão deve descrever exatamente a ação que será executada (GOOGLE, 2018b). Na Figura 27 (a) está representado o exemplo fornecido pelo manual, enquanto na Figura 27 (b) está representada a aplicação desta prática no botão *excluir* da listagem de endereços de um usuário.

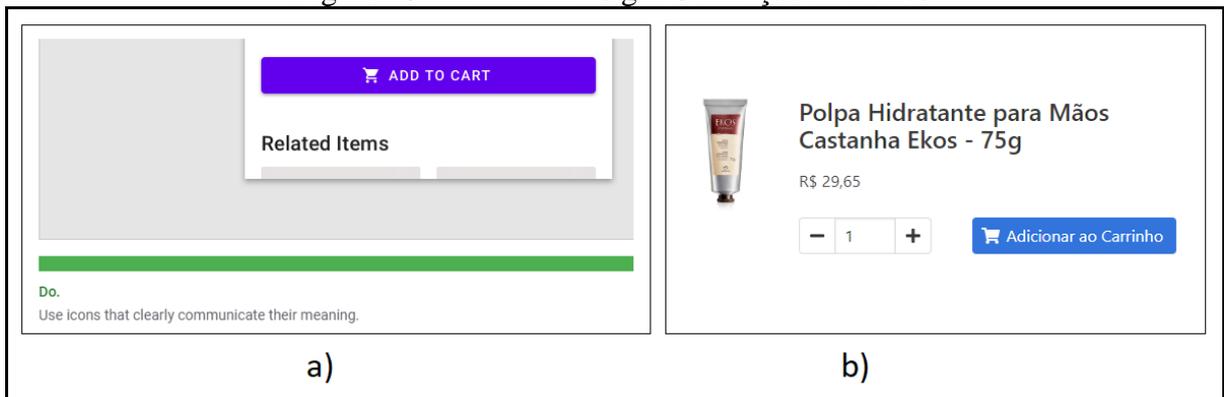
Figura 27 - Material Design: Indicar a ação de um elemento



Fonte: elaborado pelo autor.

Quanto à utilização de ícones em botões de ação, o guia do Material Design encoraja seu uso ao lado do *label*, desde que o ícone represente a ação que o botão irá realizar (GOOGLE, 2018c). Na Figura 28 (a) há uma representação da diretriz de uso de ícones em botões, enquanto na Figura 28 (b) há uma representação da tela de adicionar produto ao carrinho do sistema desenvolvido neste trabalho, utilizando a boa prática.

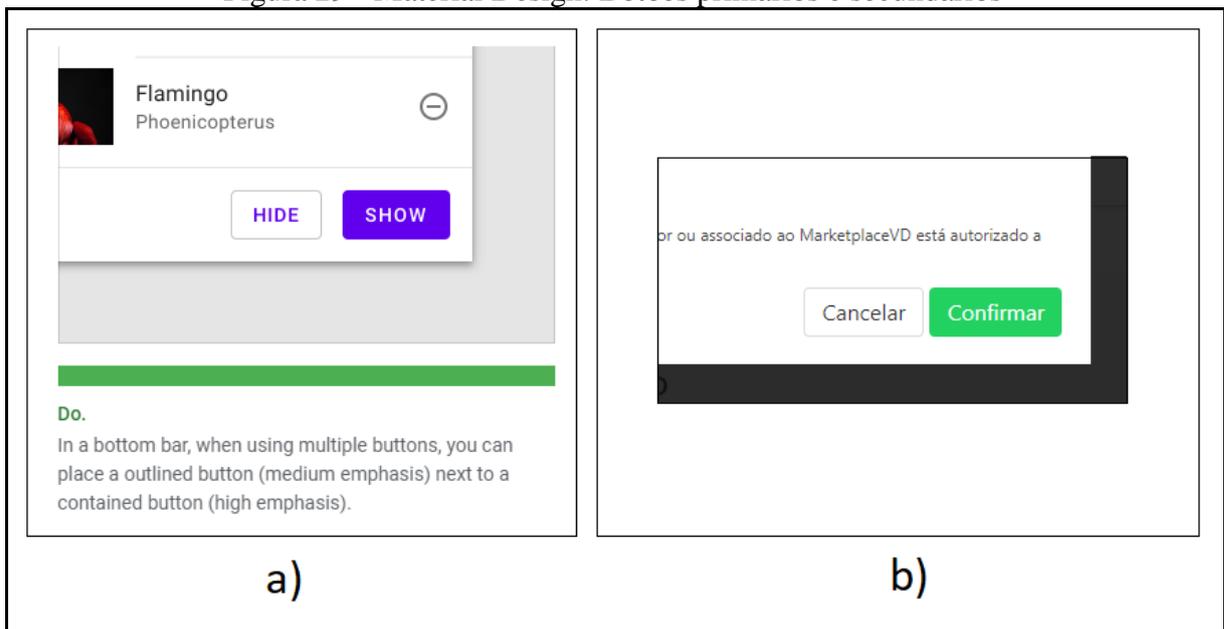
Figura 28 - Material Design: Utilização de ícones



Fonte: elaborado pelo autor.

Certos fluxos do sistema dão duas opções ao usuário, forçando a interface a exibir dois botões de ação. Para tal situação o guia do Material Design recomenda que a ação mais importante seja destacada em um botão com fundo de cor sólida, enquanto o botão secundário ou menos importante deve possuir fundo transparente (GOOGLE, 2018c). A Figura 29 apresenta exemplos práticos de uso dessa diretriz.

Figura 29 - Material Design: Botões primários e secundários



Fonte: elaborado pelo autor.

As capturas de tela apresentadas na Figura 29 exemplificam a diretriz para utilização de botões primários e secundários (Figura 29 a), bem como um exemplo de aplicação neste trabalho (Figura 29 b). Ainda de acordo com as *guidelines* do Material Design, o guia indica que não se deve utilizar apenas cores para informar uma alteração de estado no sistema; sempre que houver uma mudança de estado importante, o sistema deve se comunicar de forma eficiente por meio de cores e mensagens sucintas e objetivas (GOOGLE, 2018 b). A Figura 30

(a) apresenta uma sugestão do guia do Material Design e a Figura 30 (b) apresenta um exemplo de validação de formulário na plataforma desenvolvida neste trabalho.

Figura 30 - Material Design: Validação de formulários

Description
Let me only say that it fared with him as with the storm-tossed ship, that miserably drives along the leeward land

exceeded the maximum characters 130 / 120

Do.
The text field error state is communicated through multiple cues: title color, text field stroke, and an error message below the field.

Rua

Este campo é obrigatório.

Número

Este campo é obrigatório.

Complemento

Este campo é obrigatório.

a) b)

Fonte: elaborado pelo autor.

4 CONCLUSÕES

Com o desenvolvimento deste trabalho foi apresentada uma plataforma que permite atender vendedores diretos em sua busca por novos fornecedores e centralizar os pedidos em um único site. Pode-se afirmar que o primeiro objetivo específico estabelecido para o trabalho foi cumprido, pois a plataforma atende aos requisitos técnicos e de negócio, permitindo o cadastro de novos revendedores e a realização de pedidos. O trabalho também cumpriu o segundo objetivo específico, disponibilizar um sistema seguindo padrões de usabilidade para a web, pois está em conformidade com as validações de usabilidade realizadas tanto nos testes de performance e compatibilidade quanto na comparação com as *guidelines* do Material Design.

A utilização do *framework* Bulma auxiliou no desenvolvimento de uma interface simples que atendesse a requisitos básicos de usabilidade, validados por meio de uma comparação com *guidelines* do Material Design para a construção de interfaces. A utilização da linguagem de programação PHP, do *framework* FlightPHP e da ORM RedbeanPHP permitiram um desenvolvimento rápido, uma vez que o autor já possuía conhecimentos nas ferramentas e que os *frameworks* já possuem implementação dos métodos responsáveis pela estrutura básica da aplicação.

Um dos atrativos da plataforma é a capacidade de recrutar novas pessoas para começar a atuar no segmento de vendas diretas, porém não estava contemplado no escopo do trabalho o desenvolvimento de uma área de treinamentos e materiais publicitários. Considerando que a desenvoltura do vendedor e o conhecimento da marca são essenciais para o sucesso de um vendedor direto, uma área de treinamentos e uma vitrine com materiais publicitários dos fornecedores fariam toda a diferença na qualidade do atendimento dos revendedores, além de aumentar o engajamento com a plataforma.

Embora a aplicação permita o fluxo básico do credenciamento de um revendedor e da realização da venda, ainda se nota a necessidade da implementação de diversas funcionalidades gerenciais, para que permita um maior controle das informações e facilite este gerenciamento. Um exemplo disto é que atualmente a gestão de todas as informações de produtos e pedidos é feita no painel administrativo dentro da própria plataforma, o que se torna um problema para fornecedores com muitos produtos e com um grande fluxo de pedidos, que já possuem todo este fluxo implementado em seus softwares de gestão.

Outra limitação identificada diz respeito ao pagamento do pedido – a plataforma apenas compreende o fluxo de fornecedores que enviam o boleto de cobrança junto com os

produtos, o que é comum em empresas que trabalham com vendedores diretos. A implementação da captura de pagamentos dentro da plataforma traria mais segurança aos fornecedores, que enviariam os pedidos apenas após a confirmação do pagamento e evitariam vendedores inadimplentes.

4.1 EXTENSÕES

A partir das limitações encontradas no trabalho desenvolvido foram identificadas sugestões para possíveis extensões deste trabalho:

- a) desenvolver uma API que permita integrar produtos e pedidos com os sistemas do fornecedor;
- b) desenvolver uma API que torne possível a integração com um aplicativo para dispositivos móveis;
- c) agregar serviços para captar pagamentos dentro da plataforma;
- d) desenvolver um módulo de controle de recebimentos para o revendedor;
- e) gerar novos relatórios para de venda para que fornecedores possam analisar o desempenho de seus vendedores diretos;
- f) utilizar técnicas de *gamification* para aumentar a conversão e ter um diferencial para atrair novos fornecedores;
- g) desenvolver um módulo que permita ao revendedor organizar seu catálogo e compartilhá-lo com seus clientes;
- h) melhorar a usabilidade do leiaute responsivo em dispositivos com telas menores;
- i) desenvolver um módulo que permita que fornecedores compartilhem materiais publicitários com os revendedores;
- j) desenvolver uma ferramenta de chamados que permita o contato entre o revendedor e fornecedor dentro da plataforma.

REFERÊNCIAS

- ABEVD (Brasil) (Org.). **Códigos de Ética**: Diante dos Consumidores. 2009.
- _____. **Mesmo crescendo em ano de crise, venda direta deve ampliar segmentos**. 2015. Disponível em: <<http://www.abevd.org.br/noticias-publicacoes/mesmo-crescendo-em-ano-de-crise-venda-direta-deve-ampliar-segmentos/>>. Acesso em: 10 set. 2017
- _____. **Venda direta**. 2016. Disponível em: <<http://www.abevd.org.br/venda-direta/>>. Acesso em: 10 set. 2017.
- CYBIS, Walter; BETIOL, Adriana Holtz; FAUST, Richard. **Ergonomia e Usabilidade**: Conhecimentos, Métodos e Aplicações. São Paulo: Novatec, 2010. 352 p.
- DALVI, Tiago (Org.). **Marketplace Online no Brasil**: O que é, Exemplos, Vantagens e Desvantagens. 2015. Disponível em: <<https://blog.olist.com/marketplace-online-conceito-exemplos-vantagens-e-desvantagens/>>. Acesso em: 02 nov. 2017.
- ELO7 (São Paulo) (Org.). **Produtos Fora de Série**. 2017. Disponível em: <<https://www.elo7.com.br/sobre>>. Acesso em: 10 set. 2017.
- GERA (São Paulo) (Org.). **Vendas**. 2017 (a). Disponível em: <<http://gera.com.br/solucoes/modulos/vendas>>. Acesso em: 12 set. 2017.
- _____. **SAAS**. 2017 (b). Disponível em: <<http://gera.com.br/solucoes/gera-sgi/saas>>. Acesso em: 12 set. 2017.
- GOOGLE. **Guidelines - Material Design**. 2018 (a). Disponível em: <<https://material.io/design/guidelines-overview/>>. Acesso em: 28 jun. 2018.
- _____. **Accessibility - Material Design**. 2018 (b). Disponível em: <<https://material.io/design/usability/accessibility.html>>. Acesso em: 29 jun. 2018.
- _____. **Buttons - Material Design**. 2018 (c). Disponível em: <<https://material.io/design/components/buttons.html>>. Acesso em: 28 jun. 2018.
- KRUG, Steve. **Don't Make Me Think!**: a common sense approach to web usability. 2. ed. Berkeley: New Riders Publishing, 2006. 201 p.
- LAUDON, Kenneth C; TRAVER, Carol Guercio. **E-commerce**: business. technology. society. 10. ed. New Jersey: Pearson, 2014. 912 p.
- MARTINA, Santiago. **Entrevista a Marcos Galperín: De la idea al hecho con MercadoLibre**. 2016. Disponível em: <<http://www.endeavor.org.ar/entrevista-a-marcos-galperin-de-la-idea-al-hecho-con-mercadolibre/>>. Acesso em: 17 set. 2017.
- MERCADOLIVRE (Osasco) (Org.). **Tudo o que você precisa saber sobre o Mercado Livre**. 2018. Disponível em: <<https://ideias.mercadolivre.com.br/sobre-mercado-livre/tudo-o-que-voce-precisa-saber-sobre-o-mercado-livre/>>. Acesso em: 29 jun. 2018.
- MIYATA, Hideko. **Trabalho, redes e territórios nos circuitos da economia urbana**: uma análise da venda direta em Jundiaí e região metropolitana de São Paulo. 2010. 296 f. Tese (Doutorado) - Curso de Geografia Humana, Universidade de São Paulo, São Paulo, 2011.
- MOLA, Jeferson. **O potencial da venda direta no Brasil**. 2013. Disponível em: <<https://my.laureate.net/Faculty/webinars/Documents/Webinar%20Venda%20Direta.pdf>>. Acesso em: 20 set. 2017.
- NIELSEN, Jakob. **How Users Read on the Web**. 1997. Disponível em: <<https://www.nngroup.com/articles/how-users-read-on-the-web/>>. Acesso em: 29 jun. 2018.

PRESSMAN, Roger S. **Engenharia de Software**: Uma abordagem profissional. 7. ed. Porto Alegre: Amgh, 2011. 780 p.

REISS, Eric. **Usable Usability**: Simple Steps for Making Stuff Better. Indianapolis: John Wiley & Sons, Inc., 2012. 230 p.

SEBRAE (Org.). **Conheça as vantagens do e-marketplace para os pequenos negócios**. 2017. Disponível em: <<https://www.sebrae.com.br/sites/PortalSebrae/artigos/conheca-as-vantagens-do-e-marketplace-para-os-pequenos-negocios,3f6402b5b0d36410VgnVCM1000003b74010aRCRD>>. Acesso em: 31 out. 2017.

TURBAN, Efraim et al. **Electronic Commerce**. 8. ed. Kihei: Springer, 2015.

WFDSA (Org.). **Global Direct Selling - 2016 Retail Sales**. 2017. Disponível em: <<http://wfdsa.org/wp-content/uploads/2017/06/Final-Sales-Report-2016-5-26-2017.pdf>>. Acesso em: 31 out. 2017.

APÊNDICE A – Descrição dos Casos de Uso

UC01 – Cadastrar Revendedor

Este caso de uso compreende o cadastro de um usuário *revendedor* na plataforma. O Quadro 11 apresenta maiores detalhes sobre o caso de uso.

Quadro 11 – Descrição do caso de uso Cadastrar Revendedor (UC01)

Nome do Caso de Uso	Cadastrar <i>revendedor</i>
Descrição	O sistema deverá permitir o cadastro de <i>revendedores</i> .
Ator	<i>Revendedor</i>
Fluxo principal	<ol style="list-style-type: none"> 1. O usuário acessa a tela de <i>login</i>; 2. O usuário clica em “quero começar a vender”; 3. O sistema exibe o formulário de cadastro. 4. O usuário informa seus dados cadastrais; 5. O usuário envia o formulário; 6. O sistema valida os dados enviados; 7. O sistema armazena os dados.
Fluxo alternativo	<ol style="list-style-type: none"> 6.a. O sistema identifica campos obrigatórios não preenchidos; <ol style="list-style-type: none"> 6.a.1. O sistema apresenta a mensagem “este campo é obrigatório”; 6.a.2. O sistema destaca os campos obrigatórios que estão nulos. 6.b. Sistema identifica que já existe um usuário com o mesmo e-mail ou número de documento; <ol style="list-style-type: none"> 6.b.1. O sistema apresenta a mensagem “Já existe um usuário com este e-mail ou CPF/CNPJ.”; 6.b.2. O sistema retorna ao passo 1 do fluxo principal.
Pós-condição	<p>Cadastro do <i>revendedor</i> é realizado.</p> <p>O usuário é redirecionado ao <i>dashboard</i>.</p>

Fonte: elaborado pelo autor.

Como descrito no Quadro 11, para realizar o cadastro de *revendedor* o usuário deve acessar a tela de *login* da plataforma. Nesta tela há a opção para fazer *login* caso o usuário já possua um cadastro, e um botão para realizar um novo cadastro. Ao clicar neste botão, o usuário é direcionado a um formulário onde irá preencher seus dados cadastrais.

Ao enviar o formulário de cadastro, o sistema valida se os campos obrigatórios estão preenchidos. Caso haja campos com valor *nulo*, estes são destacados e é exibida a mensagem *este campo é obrigatório* logo abaixo do campo em questão. Uma vez que todos os campos estão preenchidos, o sistema envia os dados do formulário ao *backend* para validar se existe um usuário com o mesmo e-mail ou número de documento informado. Caso exista, é exibida uma mensagem e o usuário é redirecionado para a tela de *login*. Caso não exista um cadastro com estes dados, o sistema armazena o cadastro e direciona o usuário ao *dashboard*.

UC03 – Solicitar aprovação de fornecedor

Este caso de uso compreende o processo que um `revendedor` faz para solicitar permissão para revender produtos de um determinado `fornecedor`. O Quadro 12 descreve este caso de uso.

Quadro 12 - Descrição do caso de uso Solicitar Aprovação de Fornecedor (UC03)

Nome do Caso de Uso	Solicitar aprovação de fornecedor
Descrição	O sistema deverá permitir que um <code>revendedor</code> solicite acesso para ver e comprar produtos de um determinado <code>fornecedor</code> .
Ator	<code>Revendedor</code>
Pré-condição	O usuário deverá ter cadastro no sistema; O usuário deverá estar autenticado no sistema.
Fluxo principal	<ol style="list-style-type: none"> 1. O usuário acessa a página “fornecedores”; 2. O sistema apresenta a lista de fornecedores compatíveis com o usuário. 3. O usuário escolhe o fornecedor desejado; 4. O usuário escolhe a opção “quero revender”; 5. O sistema solicita a confirmação do usuário; 6. O usuário confirma a ação.
Fluxo alternativo	<ol style="list-style-type: none"> 5.a. O usuário cancela a solicitação; 5.a.1. O sistema oculta a mensagem de confirmação; 5.a.2. O sistema retorna à listagem de fornecedores.
Fluxo de exceção	<ol style="list-style-type: none"> 2.a. Não há fornecedores compatíveis. 2.a.1. O sistema informa que não há fornecedores compatíveis.
Pós-condição	Uma solicitação de aprovação é enviada ao fornecedor selecionado; O usuário é redirecionado de volta para a tela de fornecedores; O sistema exibe uma mensagem informando que o cadastro está pendente.

Fonte: elaborado pelo autor.

O fluxo principal indicado no Quadro 12 inicia quando o usuário `revendedor` acessa a página de fornecedores. Nesta página são listados todos os fornecedores compatíveis com o usuário. A partir desta lista, o usuário escolhe qual fornecedor deseja solicitar acesso e clica no botão `quero revender`. Ao clicar neste botão, o sistema exibe um modal de confirmação, indicando quais dados pessoais do `revendedor` serão passados ao `fornecedor` após esta ação.

Ao confirmar o envio dos dados para o fornecedor, o sistema envia uma solicitação ao `fornecedor` e aguarda a interação descrita no caso de uso UC04 – aprovar `revendedor`. O usuário é redirecionado de volta para a listagem de `fornecedores` e o sistema exibe uma mensagem informando que o cadastro está `pendente`.

UC07 – Realizar Pedido

O caso de uso UC07 representa a ação feita por um revendedor para realizar um pedido no sistema. O Quadro 13 apresenta o detalhamento deste caso de uso.

Quadro 13 - Descrição do caso de uso Realizar Pedido (UC07)

Nome do Caso de Uso	Realizar Pedido
Descrição	O sistema deverá permitir que o vendedor adicione no carrinho produtos dos fornecedores que representa, e posteriormente, realize o pedido selecionando o endereço e forma de pagamento. Realizado o pedido, tanto o revendedor quanto o fornecedor devem receber uma confirmação por e-mail.
Ator	Revendedor
Pré-condição	O usuário deverá ter cadastro no sistema; O usuário deverá estar autenticado no sistema.
Fluxo principal	<ol style="list-style-type: none"> 1. O usuário acessa a página “Produtos”; 2. O sistema apresenta a lista de produtos disponíveis para o usuário; 3. O usuário identifica o produto desejado; 4. O usuário escolhe a opção “comprar”; 5. O sistema solicita ao usuário a quantidade desejada do produto; 6. O usuário seleciona a quantidade desejada por produto; 7. O usuário adiciona o produto ao carrinho; 8. O sistema apresenta a confirmação de que o produto foi adicionado ao carrinho; 9. O usuário seleciona a opção de finalizar o pedido; 10. O sistema apresenta a tela de resumo do pedido; 11. O usuário seleciona o endereço desejado; 12. O usuário seleciona a forma de pagamento desejada; 13. O usuário confirma o pedido;
Fluxo alternativo	<ol style="list-style-type: none"> 7.a. O usuário informou uma quantidade maior que a quantidade disponível; 7.a.1. O sistema apresenta uma mensagem informando que não há estoque suficiente; 7.a.2. O sistema adiciona a quantidade disponível ao carrinho.
Fluxo de exceção	2.a. O usuário não possui fornecedores com produtos disponíveis.
Pós-condição	<p>O sistema registra o pedido; O sistema notifica o fornecedor do novo pedido; O sistema envia ao revendedor um e-mail de confirmação do pedido realizado; O sistema direciona o usuário à tela “meus pedidos”.</p>

Fonte: elaborado pelo autor.