

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

DIEGO LUCIANO BEUTLER

SISTEMA WEB PARA NEGOCIAÇÃO E DOAÇÃO DE BENS USADOS

PATO BRANCO
2022

DIEGO LUCIANO BEUTLER

SISTEMA WEB PARA NEGOCIAÇÃO E DOAÇÃO DE BENS USADOS

Web system for trading and donation of used goods

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientadora: Profa. MSc. Andréia Scariot Beulke

PATO BRANCO

2022



Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

DIEGO LUCIANO BEUTLER

SISTEMA WEB PARA NEGOCIAÇÃO E DOAÇÃO DE BENS USADOS

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, da Universidade Tecnológica Federal do Paraná (UTFPR).

Data de aprovação: 20/junho/2022

Andréia Scariot Beulke
Mestrado
Universidade Tecnológica Federal do Paraná (UTFPR)

Mariza Miola Dosciatti
Doutorado
Universidade Tecnológica Federal do Paraná (UTFPR)

Rúbia Eliza de Oliveira Schultz Ascari
Doutorado
Universidade Tecnológica Federal do Paraná (UTFPR)

PATO BRANCO

2022

RESUMO

O comércio eletrônico tem sido utilizado amplamente pelas empresas com diversos objetivos como alcançar novos mercados (superando barreiras geográficas) e fornecer facilidade de compra aos consumidores (comodidade, comparação de preços, agilidade, compra sem necessidade de sair de casa). Para empresas de pequeno porte e pessoas físicas a aquisição de um sistema de comércio eletrônico (o que inclui o site para divulgação dos produtos e o sistema para realizar as transações de comércio em si, a logística e outros) pode não ser financeiramente vantajoso. Uma maneira de beneficiar-se das vantagens desse tipo de comércio, minimizando as suas despesas inerentes, é o uso de mercados eletrônicos (*e-marketplaces*) que, de uma forma bastante simples, reúnem vendedores que utilizam uma plataforma comum para divulgar e comercializar produtos e serviços. Utilizando o conceito de *e-marketplace*, neste trabalho é desenvolvido um sistema que tem o objetivo de possibilitar que bens usados possam ser colocados para venda, troca, doação e/ou empréstimo. Bens colocados para venda ou doação podem ser vendidos e o valor resultante dessa venda pode ser destinado para entidades assistenciais previamente cadastradas no sistema e escolhidas pelo usuário proprietário do bem. Para a implementação do sistema as principais tecnologias utilizadas foram a linguagem Java, o ecossistema Spring (Boot, Data, Security e Web), o *framework* Angular e o banco de dados PostgreSQL.

Palavras-chave: E-marketplace; Comércio de bens usados; Spring; Doação; Angular.

ABSTRACT

E-commerce has been widely used by companies with several objectives such as reaching new markets (overcoming geographic barriers) and providing facilities of purchase to consumers (convenience, price comparison, agility, purchase from home). For small businesses and individuals, having an e-commerce system (which includes the website to advertise the products and the system to carry out the commerce transactions itself and logistics) may not be financially advantageous due to its cost. One way to benefit from the advantages of this type of commerce, minimizing its inherent expenses, is the use of electronic markets and e-marketplaces that, in a very simple way, bring together sellers who use a common platform to promote and negotiate products and services. Using the e-marketplace concept, a system was developed aiming to offer a web system in order to use goods that can be offered for sale, exchange, donation and/or loan. Goods offered for sale or donation can be sold and the resulting value can be destined to assist entities previously registered in the system and chosen by the user who owns the good. To implement the system, the main technologies used were the Java language, the ecosystem Spring (Boot, Data, Security, and Web), the Angular framework, and PostgreSQL database.

Keywords: E-marketplace; Trade of used goods; Spring; Donation; Angular.

LISTA DE FIGURAS

Figura 1 – Diagrama de casos de uso	28
Figura 2 – Diagrama de atividades para autenticação no sistema	35
Figura 3 – Diagrama de atividades para cadastro de anúncio	36
Figura 4 – Diagrama de atividades para cadastro de anúncio: opção doação	37
Figura 5 – Diagrama de atividades para aquisição de bem por meio de doação	38
Figura 6 – Diagrama de atividades para compra e empréstimo de bens	39
Figura 7 – Diagrama de atividades para troca de bens	40
Figura 8 – Diagrama de classes	41
Figura 9 – Diagrama de entidades e relacionamentos do banco de dados	42
Figura 10 – Tela de listagem dos anúncios apresentada quando o usuário não está autenticado.	43
Figura 11 – Tela de filtro visualizada por usuários não autenticados	44
Figura 12 – Tela exibida enquanto a busca é realizada	45
Figura 13 – Tela de listagem dos anúncios visualizada por usuários autenticados	45
Figura 14 – Tela de filtro disponibilizada para os usuários autenticados	47
Figura 15 – Tela de autenticação	48
Figura 16 – Tela para redefinir senha	49
Figura 17 – <i>E-mail</i> enviado para redefinir senha	49
Figura 18 – Tela nova senha	50
Figura 19 – Mensagens exibidas na atualização de senha	50
Figura 20 – Componente de senha	51
Figura 21 – Tela de cadastro de usuário	52
Figura 22 – Mensagens exibidas no cadastro do usuário	53
Figura 23 – <i>E-mail</i> para validar cadastro da instituição	53
Figura 24 – Tela validação do cadastro da instituição	54
Figura 25 – <i>E-mail</i> informando a instituição que o cadastro foi validado	54
Figura 26 – Seção “Cabeçalho” da tela cadastro de anúncio	55
Figura 27 – Tela cadastro de anúncio com a operação “Empréstimo”	55
Figura 28 – Tela cadastro de anúncio com a operação “Troca”	56
Figura 29 – Tela cadastro de anúncio com a operação “Doação de Valor”	56

Figura 30 – Tela para convidar a instituição	57
Figura 31 – <i>E-mail</i> de convite para instituição	57
Figura 32 – Tela cadastro de anúncio com a operação “Doação de Produto”	58
Figura 33 – Seção “Imagens” da tela de cadastro de anúncio	58
Figura 34 – Seção “Características” da tela de cadastro de anúncio	59
Figura 35 – Organização dos pacotes no <i>back-end</i>	60

LISTA DE QUADROS

Quadro 1 – Tipos de partes envolvidas em comércio eletrônico	17
Quadro 2 – Materiais utilizados no desenvolvimento do sistema	19
Quadro 3 – Requisito cadastrar anúncio	23
Quadro 4 – Requisito cadastrar usuário	24
Quadro 5 – Requisito autenticar usuário	24
Quadro 6 – Requisito adquirir bem por compra	24
Quadro 7 – Requisito receber doação de bem	25
Quadro 8 – Requisito receber doação de valor	25
Quadro 9 – Requisito adquirir bem por empréstimo	26
Quadro 10 – Requisito adquirir bem por troca	26
Quadro 11 – Requisito listar anúncios	27
Quadro 12 – Caso de uso manter anúncio e usuário: operação inclusão	29
Quadro 13 – Caso de uso manter anúncio e usuário: operação alteração	30
Quadro 14 – Caso de uso manter anúncio: operação excluir	31
Quadro 15 – Caso de uso manter anúncio e usuário: operação consultar	31
Quadro 16 – Caso de uso receber doação	32
Quadro 17 – Caso de uso adquirir bem	32
Quadro 18 – Caso de uso convidar instituição	33
Quadro 19 – Caso de uso validar instituição	34

LISTAGEM DE CÓDIGOS FONTE

Listagem 1 – Configuração das permissões de acesso aos <i>endpoints</i> da aplicação	61
Listagem 2 – <i>Controller</i> do anúncio	62
Listagem 3 – Interface contendo a declaração dos métodos padrões para todos os <i>services</i>	63
Listagem 4 – Classe contendo a implementação dos métodos padrões para todos os <i>services</i>	64
Listagem 5 – Interface contendo a assinatura dos métodos necessários para o <i>service</i> do anúncio	65
Listagem 6 – Classe contendo a implementação dos métodos necessários para o <i>service</i> do anúncio	66
Listagem 7 – Implementação dos filtros no <i>service</i> do anúncio	68
Listagem 8 – <i>Repository</i> do Anúncio	69
Listagem 9 – Interface com a assinatura dos métodos customizados no <i>repository</i> do anúncio	69
Listagem 10 – Classe com a implementação do método responsável pelo filtro no <i>repository</i> do anúncio	70
Listagem 11 – Entidade Anuncio	72
Listagem 12 – <i>Enum</i> Status	73
Listagem 13 – Exceção lançada ao tentar salvar um usuário com o valor do campo “usuário” já utilizado	74
Listagem 14 – Implementação da comunicação com o serviço da <i>Amazon Web Services</i> (AWS)	75

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
AWS	<i>Amazon Web Services</i>
B2B	<i>business-to-business</i>
B2C	<i>business-to-consumer</i>
B2G	<i>business-to-government</i>
C2B	<i>consumer-to-business</i>
C2C	<i>consumer-to-consumer</i>
C2G	<i>consumer-to-government</i>
CRUD	<i>Create, Read, Update, Delete</i>
G2B	<i>government-to-business</i>
G2C	<i>government-to-consumer</i>
G2G	<i>government-to-government</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
JSON	<i>JavaScript Object Notation</i>
P2P	<i>peer-to-peer</i>
REST	<i>Representational State Transfer</i>
S3	<i>Amazon Simple Storage Service</i>
SGBD	Sistema de Gerenciamento de Banco de Dados
URL	<i>Uniform Resource Locator</i>

SUMÁRIO

1	INTRODUÇÃO	11
1.1	CONSIDERAÇÕES INICIAIS	11
1.2	OBJETIVOS	12
1.2.1	Objetivo Geral	12
1.2.2	Objetivos Específicos	13
1.3	JUSTIFICATIVA	13
1.4	ESTRUTURA DO TRABALHO	14
2	E-MARKETPLACE	15
3	MATERIAIS E MÉTODO	18
3.1	MATERIAIS	18
3.2	MÉTODO	18
4	RESULTADOS	21
4.1	ESCOPO DO SISTEMA	21
4.2	MODELAGEM DO SISTEMA	22
4.3	APRESENTAÇÃO DO SISTEMA	43
4.4	IMPLEMENTAÇÃO DO SISTEMA	60
5	CONCLUSÃO	77
	REFERÊNCIAS	79

1 INTRODUÇÃO

Este capítulo apresenta as considerações iniciais com o contexto de aplicação do sistema desenvolvido, os objetivos pretendidos com esse desenvolvimento e a justificativa de realização deste trabalho, de acordo com os seus objetivos.

1.1 CONSIDERAÇÕES INICIAIS

Pessoas que passam a viver temporariamente em outras cidades para estudar podem precisar de eletrodomésticos, móveis, livros e até peças de vestuário. Para muitos estudantes pode não ser viável realizar o transporte de bens como móveis, da sua cidade de origem para a que irão estudar, isso porque o deslocamento pode envolver longa distância. É comum que estudantes se desloquem não apenas para outras cidades, mas para outros estados e até países. Além dos estudantes, a residência temporária em outras cidades pode ocorrer por motivos de trabalho, por exemplo.

Nesse contexto, a compra de móveis e eletrodomésticos passa a ser uma necessidade para estudantes, profissionais e outras pessoas que irão residir temporariamente em outras cidades. E, normalmente, também é uma necessidade desfazer-se por venda, doação ou empréstimo desses bens no momento que é finalizado o período de permanência na referida cidade.

Além de móveis e eletrodomésticos, é comum que os estudantes precisem adquirir livros, por meio de compra, troca ou doação, que podem ser necessários ou úteis apenas por um curto período como, por exemplo, por um semestre, por um ano ou após o livro ser lido.

Outro cenário envolvido nesse contexto está relacionado com a diferença de clima entre a cidade ou região de origem e a de residência no período de permanência temporária em outra cidade. Muitas pessoas se deslocam para regiões com climas bem diferentes da sua região de origem e vestuários apropriados podem ser necessários, ainda que temporariamente. Assim, comprar roupas e calçados de frio, por exemplo, pode não ser a melhor opção para quem planeja retornar à cidade de origem ou residir em regiões quentes.

Adquirir bens ou itens como livros, móveis, eletrodomésticos, roupas, calçados e cobertores usados, por exemplo e, posteriormente, vendê-los, trocá-los ou doá-los pode ser uma boa alternativa para os estudantes que chegam em uma cidade e, após o período de permanência, retornam para sua cidade de origem, ou que se deslocam para outras cidades para residir ou trabalhar. O mesmo ocorre com trabalhadores ou qualquer pessoa que se desloque e que permanecem por um curto período de tempo em outras cidades (por exemplo: implantadores de sistemas que além da instalação e migração acompanham a fase inicial do sistema em produção, ou que realizam a coleta de dados *in loco* para o desenvolvimento de um sistema). Para esses profissionais, dependendo do período de permanência, pode não ser viável permanecer hospedados em hotéis, por exemplo.

Considerando esse contexto, um sistema *web* facilita a colocação desses bens para

venda, troca, doação ou empréstimo e, também, pode auxiliar na divulgação para quem está vendendo ou doando produtos em geral e na busca para quem possui interesse em trocar, emprestar ou adquirir bens usados.

Incluindo nesse tipo de sistema um viés social, quando o bem é colocado para doação, ele pode ser doado ou vendido pelo site e o valor obtido com essa venda pode ser destinado para instituições assistenciais cadastradas no sistema. A instituição é escolhida pelo usuário que cobra no sistema (o proprietário) o referido bem.

Um sistema que atenda a esse cenário descrito de compra e venda de bens se assemelha à ideia de um *e-marketplace*. Um *marketplace* pode ser definido como um *shopping* virtual, que é como um *site* de *e-commerce* que oferece produtos e serviços de diferentes vendedores. Esses produtos e serviços são disponibilizados em um mesmo ambiente, ou seja, na mesma plataforma de negócio (ROSA, 2019). Em geral, um *e-marketplace* reúne diferentes tipos de produtos e serviços com o objetivo de atingir um maior número de clientes. Porém, existem também *e-marketplaces* especializados em determinado segmento, como, por exemplo, livros, roupas ou viagens (SEBRAE, 2021).

O ambiente virtual *marketplace*, no Brasil, no primeiro semestre de 2020, foi responsável por 64 milhões de pedidos, um aumento 52% se comparado ao primeiro semestre de 2019 (EBIT, 2020). Essa pode ser uma opção bastante interessante em termos financeiros pela redução de custos em relação a ter um sistema de *e-commerce* próprio, sendo, portanto, vantajoso especialmente para empresas de pequeno porte e pessoas físicas.

A semelhança do sistema proposto como resultado da realização deste trabalho com um *e-marketplace* é decorrente de o sistema reunir vendedores e possíveis compradores em um mesmo ambiente. Além da compra e venda, o sistema agrega a possibilidade de empréstimo, troca e da divulgação de bens para serem doados. Esses bens podem ser doados ou vendidos e, nesse caso, o valor resultante da venda é destinado para instituições assistenciais previamente cadastradas e indicadas pela pessoa que cobra o bem para doação.

1.2 OBJETIVOS

A seguir são apresentados o objetivo geral e os objetivos específicos que se pretende alcançar com o desenvolvimento do sistema que é o resultado da realização deste trabalho.

1.2.1 Objetivo Geral

Desenvolver um sistema *web* visando facilitar a compra e a venda, a troca, o empréstimo, e a doação de bens usados, como roupas, livros, móveis e eletrodomésticos.

1.2.2 Objetivos Específicos

Disponibilizar um sistema *web* para que as pessoas possam colocar bens usados para doação que podem ser destinados para entidades assistenciais ou que sejam vendidos e o valor obtido possa ser destinado para essas entidades.

Disponibilizar um sistema *web* que possibilite a doação de bens usados ou do valor obtido com a venda desses bens, com o intuito de contribuir com entidades assistenciais.

Possibilitar que as pessoas possam realizar empréstimo e troca de bens como livros, roupas e calçados, eletrodomésticos e móveis.

Possibilitar que bens usados como livros, roupas e calçados, eletrodomésticos e móveis possam ser negociados diretamente entre compradores e vendedores. A negociação pode envolver compra e venda, haver transação financeira, ou ocorrer na forma de troca de bens entre os negociadores.

1.3 JUSTIFICATIVA

Um sistema *web* que possibilite a compra e a venda de bens usados, bem como o empréstimo ou doação pode ser útil para o contexto de pessoas que se deslocam de suas cidades para estudar e temporariamente para trabalhar. Contudo, esse tipo de sistema pode ser utilizado para a realização dessas operações para quaisquer bens que sejam novos ou usados e não apenas no contexto acadêmico, mas como um *e-marketplace* para segmentos específicos ou diversos.

Comprar bens usados ou alugar esses bens, pode ser interessante para pessoas que residem temporariamente em um local, seja para fins de estudo ou de trabalho. O contexto do sistema desenvolvido como resultado deste trabalho é o meio acadêmico por conter um público que comumente costuma se deslocar para outras cidades para estudar, mas também pode ser utilizado por outros segmentos nos quais se aplica à negociação de bens, que não necessariamente sejam usados. Por exemplo, a comercialização de produtos artesanais, coloniais, culinários e de outros segmentos, em que pessoas físicas ou empresas de pequeno porte (para as quais não se justifica ter um sistema específico ou exclusivo), podem utilizar o sistema para a realização de vendas *on-line*.

No momento de cadastro de um bem para venda, locação ou doação, o usuário pode perceber que o bem terá utilidade para uma entidade assistencial e decidir doá-lo para essa entidade. Poderá, ainda, colocar outros bens, que não julgue de interesse das instituições cadastradas, para venda e o valor obtido pode ser destinado para essas instituições.

A sociedade de consumo é caracterizada pelos avanços de produção do sistema capitalista intensificados ao longo do século XX e pela obsolescência programada ou planejada, que consiste na produção de mercadorias que são rapidamente descartadas (PENA, 2021). A rápida evolução da tecnologia também torna os produtos rapidamente defasados. É inegável que o consumismo e a obsolescência contribuem significativamente para o descarte de bens.

Como forma de auxiliar a minimizar o descarte de bens em condições de uso, a comercialização, o empréstimo e a doação de bens usados incentivam o reuso, auxiliando na redução do desperdício. A ciência que um bem em uso - seja roupa, calçado, livro, eletrodoméstico ou outro após finalizada a sua utilidade para o usuário atual - será utilizado por outra pessoa, pode ocasionar maior cuidado no uso desse bem, já que estará posteriormente com outro usuário. Essa consciência pode colaborar na busca pela minimização dos impactos e dos danos causados ao meio ambiente pelo lixo produzido e pela exploração excessiva dos recursos naturais (no sentido de o planeta não ter mais condições de repor).

Assim, o sistema proposto nesse trabalho se justifica pela contribuição que pode trazer para as pessoas no sentido de adquirir produtos usados ou de destinar para outras pessoas esses produtos, por meio da compra, troca, doação ou empréstimo. E, além disso, a possibilidade de destinar os bens ou o valor obtido com a sua venda para entidades assistenciais.

1.4 ESTRUTURA DO TRABALHO

O restante deste texto está organizado em capítulos. O Capítulo 2 apresenta o referencial teórico que fundamenta o tipo de sistema *web* desenvolvido que é uma espécie de comércio eletrônico caracterizado como *e-marketplace* porque permite que diversas pessoas físicas e instituições assistenciais, possam adquirir, vender, trocar, emprestar e doar bens usados. O Capítulo 3 apresenta as tecnologias e as ferramentas utilizadas para a modelagem e a implementação do sistema, incluindo, entre outros, ambientes de modelagem e desenvolvimento, linguagem de programação e banco de dados. O Capítulo 4, contém a modelagem, a apresentação e a implementação do sistema desenvolvido. Por fim, no Capítulo 5, estão as considerações finais do trabalho que é seguida pelas referências utilizadas na composição do texto.

2 E-MARKETPLACE

Mercado eletrônico ou *e-marketplace*, um acrônimo para *electronic e-marketplace* pode ser visto como um *shopping center* virtual (ZACHO, 2017) e que pode ser genérico (reunir segmentos de mercado distintos) ou especializado também referido como temático (somente um setor de atividade ou segmento de mercado específico, como livros, camisetas, tênis) (SEBRAE, 2014). Genérico ou temático, um *e-marketplace* facilita a busca por produtos e serviços e por preços mais vantajosos por funcionar como uma vitrine virtual. Assim, *e-marketplaces* podem ser vistos como lugares nos quais ocorrem compras e vendas, da mesma forma que ocorre nos *shopping centers* físicos ainda que realizem comércio eletrônico, que são como ambientes que colocam consumidores e lojistas em contato.

Em termos mais conceituais, um *electronic marketplace* (*e-marketplace*) é um espaço de informações de negócio comum (GUO *et al.*, 2012), que é uma infraestrutura de mercado eletrônico em uma forma de mercado que envolve o uso de sistema de informação no qual compradores e vendedores conduzem negócios por meio de transações eletrônicas (GUO, 2007). Um *e-marketplace* é uma extensão e a integração de vários sistemas de informação empresariais e tecnologias de computação na Internet (TAN *et al.*, 2010).

O conceito de mercado eletrônico não é recente, em termos de desenvolvimento de tecnologias de comunicação e informação e uso da Internet para fins comerciais. Ainda em 1987, Malone, Yates e Benjamin (1987) definiram *e-marketplace* como um intermediário eletrônico que facilita transações entre múltiplos compradores e vendedores. Nessa mesma linha, Campos *et al.* (2015) definem um *e-marketplace* como um modelo de negócio *on-line* que tem como base a intermediação entre compradores e vendedores possibilitando a comercialização por meio da plataforma que suporta esse modelo de negócio. Movahedi, Lavassani e Kumar (2012) considerando um *e-marketplace* como um meio colaborativo, o definem como infraestruturas institucionais colaborativas e eficientes baseadas na Internet para negociações e transações inter e intraorganizacionais. Loukis, Spinellis e Katsigiannis (2011) definem *e-marketplace* como um sistema de informação interorganizacional que provê um espaço virtual no qual múltiplos compradores e vendedores podem se comunicar e realizar transações, muito frequentemente suportado por vários serviços adicionais necessários na realização de negócios como financeiro, transporte, logística e outros.

Esses conceitos se aplicam ao contexto de negócio do sistema desenvolvido como resultado deste trabalho. No qual há pessoas físicas e instituições que utilizam um sistema *web*, uma espécie de vitrine virtual para comprar, vender, emprestar, doar, ou ainda doar para vender destinando o valor correspondente para entidades assistenciais cadastradas.

Essa forma de mercado eletrônico tem mostrado que um processo de negócio para uma transação eletrônica por meio de múltiplas empresas compreende uma sequência condicional de atividades orientadas em fluxo (ZHAO *et al.*, 2009). Um contexto de ação representa uma atividade que pode ser considerada como um verbo sendo anotado por um conceito definido,

descrito por algumas regras e implementado por uma operação (GUO *et al.*, 2012).

E-marketplace define um novo modelo de negócio que está se desenvolvendo e mudando rapidamente. Os termos *electronic hub*, *electronic intermediary* e *Internet based electronic-market* (mercado eletrônico baseado na Internet) têm sido usados para referir-se ao conceito de *e-marketplace*. Mercados eletrônicos possuem, tradicionalmente, três papéis: facilitar as transações, promover o encontro de compradores e vendedores e prover infraestrutura institucional (BAKOS, 1998). Assim, há estudiosos, como Lau *et al.* (2008), que consideram *e-marketplaces* como uma forma de comércio eletrônico estabelecido entre empresas.

E-marketplace como modelo de negócio é baseado na ideia que pode auxiliar as organizações a (ENG, 2004):

- a) definir uma sequência de atividades para processos de negócio complexos;
- b) ganhar eficiência;
- c) agregar compradores e vendedores em um mesmo ponto;
- d) permitir as organizações participantes desfrutar economias de escala e liquidez;
- e) comprar e vender os bens mais diversos com mais facilidade, rapidamente e com custos mais efetivos;
- f) eliminar barreiras geográficas e expandir globalmente o negócio para obter benefícios em novos mercados.

Os *e-marketplaces* podem ser classificados utilizando critérios diversos como as partes envolvidas, os tipos de produtos e serviços comercializados e o mecanismo de precificação utilizado, entre diversos outros critérios. Possivelmente uma das formas mais comuns de classificação dos *e-marketplaces*, assim como ocorre com o comércio eletrônico, é a categorização baseada nos tipos de partes envolvidas nas transações. Genericamente três partes podem ser assumidas para qualquer transação (COPPEL, 2000): empresas (*business*), consumidores e governo. É importante ressaltar que consumidores podem ser vistos como o consumidor final dos produtos e serviços comercializados. Contudo, assim como no comércio eletrônico, essas três partes podem realizar o papel de fornecedores e consumidores de produtos e serviços. Essas três partes poderiam ser vistas como pessoas físicas, pessoas jurídicas (empresas, entidades e afins) e governo. Isso porque, empresas podem vender e adquirir bens de outras empresas, de pessoas físicas (caracterizados por Coppel (2000) como consumidores) e do governo. Idem para pessoas físicas (consumidores) que podem fornecer e adquirir bens e serviços de e para empresas, governo e outros consumidores (pessoas físicas). E o mesmo ocorre para o governo que pode fornecer e adquirir bens e serviços de e para empresas, de e para governo e de e para consumidores, sendo esses, pessoas físicas.

Assim, há nove possíveis combinações desses três grupos, como mostrado no Quadro 1, e cada uma delas identifica um tipo de *e-marketplace* (MOVAHEDI; LAVASSANI; KUMAR, 2012). Autores como Ragone *et al.* (2009), têm utilizado o termo *peer-to-peer* (P2P) como forma de enfatizar a neutralidade do *e-marketplace*.

Quadro 1 – Tipos de partes envolvidas em comércio eletrônico

	Governo (government)	Empresa (business)	Consumidor (consumer)
Governo (government)	<i>government-to-government</i> (G2G)	<i>government-to-business</i> (G2B)	<i>government-to-consumer</i> (G2C)
Empresa (business)	<i>business-to-government</i> (B2G)	<i>business-to-business</i> (B2B)	<i>business-to-consumer</i> (B2C)
Consumidor (consumer)	<i>consumer-to-government</i> (C2G)	<i>consumer-to-business</i> (C2B)	<i>consumer-to-consumer</i> (C2C)

Fonte: Movahedi, Lavassani e Kumar (2012, p. 19).

O Quadro 1 mostra os tipos de partes envolvidas no comércio eletrônico para manter fidelidade ao termo indicado pelo autor que referencia o quadro, embora o artigo desse autor seja sobre *e-marketplace*. Na verdade, essas mesmas partes são identificadas em ambos comércio (*e-commerce*) e mercado (*e-marketplace*) eletrônicos. Como destacam Oh e Kim (2011), múltiplos compradores e vendedores podem realizar comércio em um espaço virtual como a Internet por meio de *e-marketplace*. E compradores e vendedores, sejam de produtos ou serviços, podem ser pessoas físicas, jurídicas (empresas), governo e entidades sem fins lucrativos e benéficas, entre outras.

Alstyne, Parker e Choudary (2021) destacam que os novos modos de comércio, referindo-se especificamente ao *e-marketplace*, estão ganhando mercado porque eles reúnem vendedores e consumidores. Além disso, esses modos de comércio, especialmente de comércio eletrônico, são importantes para os participantes, destacando-se as empresas pela necessidade de expansão de seus mercados e alcance maior de clientes, que não têm condições de sozinhas criar, adquirir, pagar para usar sistemas de comércio eletrônico. Esse cenário fundamenta a colocação dos autores Alstyne, Parker e Choudary (2021) que as empresas que não passarem a usar essas plataformas terão problemas para sobreviver. Isso no sentido de expandir e mesmo manter-se no mercado pelo uso da Internet e das vantagens que conceitos como *e-marketplace* apresentam para vendedores e consumidores.

3 MATERIAIS E MÉTODO

Este capítulo apresenta os materiais e o método utilizados na definição dos requisitos do sistema, na modelagem e na implementação do sistema. O capítulo apresenta também as atividades realizadas que definem, de certa forma, o método utilizado no desenvolvimento do sistema obtido como resultado da realização deste trabalho. Essas atividades incluem a definição dos requisitos do sistema e das tecnologias utilizadas na modelagem e na implementação desses requisitos. A análise que modela os requisitos de forma que possam ser implementados representando os requisitos pretendidos para o sistema. A implementação e os testes realizados no código implementado.

3.1 MATERIAIS

Os materiais se referem às ferramentas, tecnologias, ambientes de desenvolvimento e outros utilizados para realizar as atividades desde a definição dos requisitos à implementação do sistema. O Quadro 2 apresenta as ferramentas e as tecnologias utilizadas.

3.2 MÉTODO

Os procedimentos utilizados para a definição dos requisitos, a modelagem desses requisitos e a implementação do sistema são apresentados a seguir:

a) Levantamento de requisitos – a ideia do sistema surgiu do interesse do autor deste trabalho em destinar bens adquiridos por estudantes que residem temporariamente em outras cidades para estudar. Partindo desse cenário inicial verificou-se que não apenas estudantes, mas profissionais, como implantadores de sistemas, podem residir por períodos de tempo em outras cidades. Por vezes esses períodos são longos e por essa e outras condições hospedar-se em um hotel, por exemplo, pode não ser a opção mais adequada ou financeiramente mais apropriada. Esse foi o cenário básico que sustentou a ideia de um sistema *web* para publicar anúncios para a venda e compra de bens usados. Agregado a isso pensou-se que esses bens poderiam ser emprestados por um determinado período de tempo, como roupas de frio passando de um estudante para outro em invernos sucessivos. Surgiu, assim, a ideia de agregar empréstimo e troca às operações de compra e venda. Por fim, foi agregada a ideia de doação em que a pessoa se desfaz do bem que não quer ou não precisa e cede para outra pessoa. Assim, surgiu a ideia de que esses bens poderiam ser doados para entidades assistenciais. Como essas entidades podem precisar de bens que não sejam doados e elas possuem uma série de despesas, os requisitos, em termos de escopo do sistema, foram complementados com a possibilidade de bens usados e cadastrados no sistema serem vendidos. E o valor obtido com a venda ser destinado para uma instituição assistencial cadastrada no sistema e escolhida pelo usuário que inclui o referido bem.

Quadro 2 – Materiais utilizados no desenvolvimento do sistema

Ferramenta/Tecnologia	Versão	Disponível em	Finalidade
Angular	12.2.0	https://angular.io	<i>Framework</i> de programação para o <i>front-end</i>
PrimeNG	12.0.0	https://www.primefaces.org	Biblioteca de componentes para Angular
Java	17	https://docs.oracle.com	Linguagem de programação para o <i>back-end</i>
Spring Boot	2.6.6	https://spring.io	<i>Framework</i> para <i>bootstrap</i> do projeto
Spring Data JPA	2021.1.4	https://spring.io	Camada de acesso e manipulação de dados
Spring Security	5.6.3	https://spring.io	Camada de segurança da aplicação
Spring Web	2.6.6	https://docs.spring.io	Camada <i>Web</i> da aplicação
Hibernate Envers	5.6.7	https://hibernate.org/orm/envers	Auditoria dos dados
Lombok	1.18.22	https://projectlombok.org/	Biblioteca java, utilizada para gerar códigos padrões
IntelliJ IDEA	2021.3.1	https://www.jetbrains.com	Ambiente integrado de desenvolvimento
PostgreSQL	13.2	https://www.postgresql.org	Gerenciador de banco de dados
DBeaver	21.0.5	https://dbeaver.io	Administração e desenvolvimento para PostgreSQL
Visual Paradigm	16.3	https://www.visual-paradigm.com	Modelagem do projeto

Fonte: Autoria própria (2022).

Esse cenário e contexto serviram de base para a definição dos requisitos. À medida que esses requisitos eram pensados como um sistema, complementações foram surgindo, como os atores (tipos de usuários e suas permissões) e a limitação do escopo para que o sistema pudesse ser implementado no período letivo de realização do trabalho de conclusão de curso.

b) Análise e projeto – após definido o escopo e contexto do sistema, os requisitos foram modelados como um diagrama de casos de uso. A partir dos casos de uso foram elaborados os requisitos funcionais e os respectivos requisitos não-funcionais. Diagramas de atividades foram desenvolvidos para definir as sequências das operações realizadas no sistema. Um diagrama de classes definiu os dados persistentes, bem como as operações realizadas com esses dados e outras. O diagrama de entidades e relacionamento definiu as tabelas e respectivos campos estabelecendo os dados que seriam persistidos pelo sistema. A ferramenta Visual Paradigm em sua versão *community* foi utilizada para a modelagem de todos os diagramas desenvolvidos, com exceção do diagrama de entidades e relacionamentos do banco de dados, que foi extraído a partir da base, por meio da ferramenta DBeaver.

c) Implementação (codificação) do sistema – A implementação do sistema foi realizada

utilizando as tecnologias e ferramentas apresentadas no Quadro 2.

d) Realização dos testes - os testes foram informais, isto é, sem um plano de testes específicos e foram realizados paralelamente à implementação do sistema. Esses testes visaram verificar a implementação dos requisitos definidos para o sistema e a identificação de erros de codificação.

4 RESULTADOS

Este capítulo apresenta os resultados da realização deste trabalho que é o desenvolvimento de um sistema *web*.

4.1 ESCOPO DO SISTEMA

O sistema *web* desenvolvido visa auxiliar na troca, compra, venda, empréstimo ou doação de bens de segunda mão ou usados, como, por exemplo, livros, móveis, eletrodomésticos, roupas, calçados e quaisquer outros bens.

O contexto utilizado para exemplificar as funcionalidades é de estudantes universitários, mas ressalta-se que o sistema pode ser utilizado para troca, empréstimo, compra e venda ou doação de bens para qualquer segmento em que essas operações possam ser realizadas. Como exemplo, cita-se uma empresa que pode promover esse tipo de operação entre os seus funcionários, com bens usados sendo comprados, vendidos, emprestados e doados pelos próprios funcionários da empresa.

O objetivo principal do sistema é estimular a troca, o empréstimo e a doação, embora seja possível realizar compra e venda. Os itens destinados à doação são os que o usuário cadastrou no sistema, sem a intenção de trocar por outros itens ou receber dinheiro com a venda. Esses itens podem ser colocados para venda e o dinheiro recebido é destinado para instituições assistenciais escolhidas pelo usuário que está realizando a doação. Essas instituições estão previamente cadastradas no sistema e caso não estejam, o consumidor pode enviar um convite pelo próprio sistema para que a instituição efetue o cadastro, ou a mesma pode cadastrar-se sem haver convite. Ressalta-se que o cadastro de uma instituição deverá sempre ser validado pelo administrador. O sistema não estabelecerá parâmetros ou requisitos para validação, essa atividade dependerá do administrador responsável pelo referido sistema. Essas instituições possuem acesso aos itens cadastrados no site para doação e podem solicitar esses itens para uso.

O sistema permitirá as seguintes operações com os bens cadastrados:

a) venda – bens com o objetivo de serem vendidos, com o valor destinado para o usuário que registrou o bem.

b) compra – aquisição de bens disponibilizados no sistema para venda ou doação. Os bens colocados para venda, possuem o seu respectivo valor contabilizado para o usuário que cadastrou o bem. Quanto aos bens colocados para doação e que são vendidos, o valor é contabilizado para a instituição assistencial indicada pelo usuário que cadastrou o respectivo bem.

c) empréstimo – bens disponibilizados para locação ou empréstimo por um determinado período de tempo. Esse período pode ser predefinido no momento da locação ou a data de devolução pode ficar em aberto.

d) doação – bens cadastrados no sistema para serem doados para instituições. Esses bens podem ser doados ou comprados por usuários do sistema e, nesse caso, o valor correspondente

é destinado para uma entidade assistencial indicada pelo usuário que cadastrou o referido bem. O registro das negociações realizadas contém o bem negociado, o valor (se for o caso de venda), a data de realização, quem está vendendo ou doando e quem está comprando ou recebendo o bem por meio de doação.

O sistema terá as seguintes categorias de usuários:

a) administrador – responsável pela validação de usuário do tipo instituição e possui as permissões do usuário consumidor.

b) instituições – entidades institucionais com permissões para consulta, compra, aquisição de bens doados e recebimento do valor de bens postos para doação e que são vendidos.

c) consumidores – cadastram bens para venda, empréstimo, troca ou doação, compram ou tomam bens por empréstimo ou troca. Podem alterar somente os seus registros (dos bens que cadastram no sistema). Podem convidar instituições assistenciais para que efetuem o registro no sistema, sendo o cadastro validado pelo administrador, para que possam receber doações de bens ou o valor correspondente à venda dos itens colocados para doação.

Usuários não cadastrados no sistema, ou seja, sem dados para autenticação, podem realizar somente consultas na página, visualizando bens para venda, empréstimo, troca e doação.

As funcionalidades relacionadas a pagamento e entrega de bens não serão implementadas pelo sistema. O sistema não implementa os requisitos relacionados a pagamentos, registrando apenas as transações realizadas e os respectivos valores. Inicialmente, os bens ficam com o *status* de “Disponível”, e após um usuário marcar que possui interesse no bem, ou no caso da operação “Doação de Produto” após a instituição de destino aceitar o produto, o *status* é alterado para “Em Negociação”. Por fim, após a negociação ter ocorrido, o autor do anúncio marca o mesmo como “Finalizado”. As funcionalidades para verificar se o bem foi entregue, sobre o transporte ou registro de pagamento não foram implementadas porque não fazem parte do escopo do trabalho. Os valores oriundos das vendas de bens disponibilizados para doação são registrados no banco de dados para a referida instituição, mas não há controle de retirada, saque, desses valores. O sistema de pagamento, entretanto, poderá ser implementado futuramente utilizando um sistema de pagamentos como o PayPal, Mercado Pago, PicPay, entre outros.

4.2 MODELAGEM DO SISTEMA

Nos quadros a seguir são apresentados, de forma detalhada, os requisitos funcionais e os requisitos não-funcionais relacionados que foram definidos para o sistema.

No Quadro 3 está o requisito funcional “cadastrar anúncio”. Esse requisito define o cadastro de bens que o usuário pretende vender, emprestar, doar ou trocar.

Quadro 3 – Requisito cadastrar anúncio

F1 Cadastrar anúncio

Descrição: o sistema deve realizar o cadastro dos anúncios de itens a serem usados nas operações do sistema. O anúncio deverá conter, pelo menos, os seguintes dados: título, descrição, características do bem, categoria da operação, categoria do bem (ex. roupas, eletrodomésticos, livros e outras), imagens do bem e *status*. Além da inclusão (cadastro), o sistema deve permitir a alteração e a exclusão dos registros, bem como a consulta dos dados armazenados.

Requisitos Não-Funcionais (regras de negócio):

NF 1.1 Categorias de operações: as operações devem ser: venda, troca, empréstimo e doação.

NF 1.2 Status: o *status* de um item pode ser: “Disponível”, “Em Negociação”, “Finalizado”.

NF 1.3 Alteração status: uma vez cadastrado, o anúncio possui *status* “Disponível”, quando um usuário marcar a opção, “Tenho Interesse”, o *status* passa para “Em Negociação”. Pode ser desmarcado pelo usuário a opção, voltando para “Disponível”, ou ainda, pode ocorrer a negociação e o anunciante pode, dependendo do resultado dessa negociação, marcar como “Finalizado” ou “Disponível”.

NF 1.4 Empréstimo: o formulário deve conter um campo opcional para a data de devolução, apenas na operação de empréstimo, e um campo valor, que aceite zero, para o caso de o anunciante não querer remuneração pelo empréstimo.

NF 1.5 Troca: o formulário deve apresentar, apenas na operação troca, um campo obrigatório do tipo texto para o usuário informar os itens que ele aceita na troca.

NF 1.6 Doação: formulário deve apresentar, apenas na operação doação, um campo obrigatório para que o usuário selecione a instituição para a qual deseja doar, caso a mesma não tenha cadastro, deve-se possibilitar o envio de um convite para a mesma por meio do sistema, informando o *e-mail* da instituição ou solicitar para que ela se cadastre. Em caso de doação de valor, deve-se apresentar também o campo obrigatório valor.

NF 1.7 Venda: o formulário deve conter um campo opcional (com exceção para a operação venda) do tipo decimal para informar o valor de venda.

NF 1.8 Acesso ao cadastro: para cadastrar um anúncio o usuário precisa estar autenticado no sistema.

NF 1.9 Regras de alteração e exclusão: a alteração ou a exclusão de um anúncio pode ser realizada apenas pelo proprietário do anúncio e somente se o *status* for “Disponível”.

NF 1.10 Imagens do bem: é obrigatório o cadastro de ao menos uma imagem do bem anunciado.

Fonte: Autoria própria (2022).

O requisito funcional “cadastrar usuário” e os requisitos não-funcionais relacionados são apresentados no Quadro 4. O usuário do tipo “consumidor” realiza o seu cadastro no sistema e, em seguida, pode realizar autenticação. O usuário do tipo “instituição” só poderá acessar o sistema após o seu cadastro ser validado pelo administrador. Como as instituições assistenciais podem receber bens e dinheiro de bens adquiridos por outros usuários, considerou-se mais adequado, em termos de regra de negócio, que o cadastro dessas instituições fosse validado pelo administrador do sistema.

Quadro 4 – Requisito cadastrar usuário

F2 Cadastrar usuário

Descrição: realizar o cadastro dos usuários coletando os dados: nome, telefone, *e-mail*, usuário, senha, foto do usuário (opcional), endereço e se é instituição.

Requisitos Não-Funcionais (regras de negócio):

NF 2.1 Senha: a senha deve conter no mínimo 6 caracteres, contendo letras e números.

NF 2.2 Cadastro como instituição: requer validação por um usuário do sistema com permissão de administrador.

Fonte: Autoria própria (2022).

No Quadro 5 é apresentado o requisito funcional autenticação no sistema. O usuário não autenticado no sistema poderá acessar somente a tela de visualização de anúncios.

Quadro 5 – Requisito autenticar usuário

F3 Autenticar usuário

Descrição: o sistema deve possibilitar que o usuário faça sua autenticação no sistema.

Requisitos Não-Funcionais (regras de negócio):

NF 3.1 Tela de login: o sistema deve conter uma opção na tela de autenticação que redirecione para a tela de cadastro, caso o usuário ainda não seja cadastrado e deseje fazê-lo.

NF 3.2 Redefinir senha: o sistema deve conter uma opção na tela de autenticação “Esqueceu a senha?”, caso o usuário tenha esquecido a senha, então o sistema deve pedir o usuário cadastrado e enviar um *e-mail* para o mesmo com um link para a alteração da mesma.

Fonte: Autoria própria (2022).

O requisito para compra de bens cadastrados no sistema é apresentado no Quadro 6. Como o sistema não terá implementadas as funcionalidades referentes ao pagamento, logística de entrega e outras operações a partir da escolha do bem para compra, a compra é considerada efetivada pela mudança de *status* por meio de um campo específico no registro.

Quadro 6 – Requisito adquirir bem por compra

F4 Adquirir bem por compra

Descrição: realizar o processo de compra de um bem.

Requisitos Não-Funcionais (regras de negócio):

NF 4.1 Permitir verificar quem marcou interesse: o sistema deve permitir ao anunciante verificar qual usuário tem interesse no seu bem, para que caso não tenha retorno ele possa contatá-lo.

NF 4.2 Alteração de status: após o usuário comprador marcar que tem interesse em um bem, o sistema atualiza o *status* do anúncio para “Em Negociação”. Deve-se desabilitar o botão de alteração de *status* para os demais usuários que não sejam o anunciante ou o usuário de destino. O sistema deve possibilitar ao anunciante alterar o *status* do anúncio caso o comprador que marcou interesse não entre em contato.

NF 4.3 Negociação: após realizada a negociação, o anunciante deve atualizar o *status* do anúncio para “Finalizado”, ficando, assim, vinculado ao comprador, ou “Disponível”, ficando habilitado o botão de alteração de *status*.

Fonte: Autoria própria (2022).

O recebimento por doação de um bem cadastrado no sistema é representado pelo requisito apresentado no Quadro 7. Ressalta-se que somente usuários do tipo “instituição” previamente cadastrados no sistema podem receber bens colocados para doação. No momento do cadastro do bem o usuário indica a instituição que receberá o produto.

Quadro 7 – Requisito receber doação de bem

F5 Receber doação de bem

Descrição: realizar aquisição de um bem anunciado para doação.

Requisitos Não-Funcionais (regras de negócio):

NF 5.1 Controle de visibilidade do anúncio: é exibido apenas para a instituição de destino e para o anunciante.

NF 5.2 Alteração de status: ao ser selecionada a instituição (já cadastrada e validada) e salvo o anúncio, o *status* fica como “Disponível” até que a mesma aceite a doação do bem. Posterior ao aceite o *status* é atualizado para “Em Negociação”.

NF 5.3 Negociação: após realizada a negociação, o anunciante deve atualizar o *status* do anúncio para “Finalizado”, ficando assim vinculado à instituição, ou alterar para “Disponível”. Após o usuário alterar o *status* para “Disponível” o sistema deve apresentar a mensagem de alerta “Status foi atualizado, porém deve-se alterar a instituição no anúncio”.

Fonte: Autoria própria (2022).

O requisito funcional e os requisitos não-funcionais relacionados ao recebimento do valor (dinheiro) de um bem anunciado para doação e que foi vendido são apresentados no Quadro 8. Ressalta-se que somente usuários do tipo “instituição” previamente cadastrados no sistema podem receber bens colocados para doação e vendidos. No momento do cadastro do bem, o usuário indica a instituição que receberá o valor do bem, caso seja vendido.

Quadro 8 – Requisito receber doação de valor

F6 Receber doação de valor

Descrição: realizar o processo de doação do valor (destino do valor para a instituição assistencial indicada) proveniente de uma venda de um bem.

Requisitos Não-Funcionais (regras de negócio):

NF 6.1 Alteração de status: ao ser selecionada a instituição (já cadastrada e validada) e salvo o anúncio, o *status* é “Disponível”. Após o usuário comprador marcar que tem interesse, o sistema atualiza o *status* do anúncio para “Em Negociação”, e vincula o bem ao usuário. Deve-se desabilitar o botão de alteração de *status* para os demais usuários que não sejam o anunciante ou o usuário de destino. O sistema deve possibilitar ao anunciante alterar o *status* do anúncio caso o usuário que marcou interesse não entre em contato.

NF 6.2 Negociação: após realizada a negociação, o anunciante deve atualizar o *status* do anúncio para “Finalizado”, ficando assim vinculado o comprador que recebeu o bem e a instituição que recebeu o valor, ou para “Disponível”, ficando habilitado o botão de alteração de *status*.

Fonte: Autoria própria (2022).

Os bens cadastrados no sistema podem ser adquiridos por empréstimo. Nesse caso, é indicada a data de devolução do empréstimo. O requisito que apresenta a aquisição de bem por empréstimo é apresentado no Quadro 9.

Quadro 9 – Requisito adquirir bem por empréstimo

F7 Adquirir bem por empréstimo

Descrição: realizar aquisição de bens anunciados por meio de empréstimo.

Requisitos Não-Funcionais (regras de negócio):

NF 7.1 Permitir verificar quem marcou interesse: O sistema deve permitir ao anunciante verificar qual usuário tem interesse no seu bem, para que caso não tenha retorno ele possa tentar contato.

NF 7.2 Alteração de status: após o usuário marcar que tem interesse, o sistema atualiza o *status* do anúncio para “Em Negociação”, deve-se desabilitar o botão de alteração de *status* para os demais usuários que não sejam o anunciante ou o usuário de destino. O sistema deve possibilitar ao anunciante alterar o *status* do anúncio caso o usuário que marcou interesse não entre em contato.

NF 7.3 Negociação: após realizada a negociação o anunciante deve atualizar o *status* do anúncio para “Finalizado”, ficando assim vinculado ao comprador, ou “Disponível”, ficando habilitado o botão de alteração de *status*.

NF 7.4 Data de devolução: Deve ser possível que o anunciante altere a data de devolução, desde que o *status* seja diferente de “Finalizado”.

Fonte: Autoria própria (2022).

Os bens cadastrados no sistema podem ser adquiridos por troca, que ocorre quando dois usuários permitem bens entre si. O requisito para a operação de troca é apresentado no Quadro 10.

Quadro 10 – Requisito adquirir bem por troca

F8 Adquirir bem por troca

Descrição: realizar aquisição por meio de troca de bens entre usuários do sistema.

Requisitos Não-Funcionais (regras de negócio):

NF 8.1 Permitir verificar quem marcou interesse: O sistema deve permitir ao anunciante verificar qual usuário tem interesse no bem, para que caso não tenha retorno ele possa tentar contato.

NF 8.2 Alteração de status: após o usuário marcar que tem interesse, o sistema atualiza o *status* do anúncio para “Em Negociação”, deve-se desabilitar o botão de alteração de *status* para os demais usuários que não sejam o anunciante ou o usuário de destino. O sistema deve possibilitar ao anunciante alterar o *status* do anúncio caso o usuário que marcou interesse não entre em contato.

NF 8.3 Negociação: Após feita a negociação o anunciante deve atualizar o *status* do anúncio para “Finalizado”, ficando assim vinculado ao comprador, ou “Disponível”, ficando habilitado o botão de alteração de *status*.

NF 8.4 Bens para troca: Deve ser possível que o anunciante altere o campo de bens que aceita na troca, desde que o *status* seja diferente de “Finalizado”.

Fonte: Autoria própria (2022).

A listagem dos anúncios pode ser visualizada por usuários não autenticados no sistema, independentemente de terem cadastro (*login* e senha) no sistema. O Quadro 11 apresenta o requisito para a visualização da listagem de anúncios.

Quadro 11 – Requisito listar anúncios

F10 Listar anúncios

Descrição: o sistema deve conter a listagem dos anúncios. A visualização de anúncios cadastrados é possível para usuários não autenticados no sistema.

Requisitos Não-Funcionais (regras de negócio):

NF 10.1 Filtros: deve ser possível filtrar pelos campos título, descrição, categoria, operação, *status*, data de publicação, estado e cidade. Caso o usuário esteja autenticado, deve-se ter a opção de filtrar os anúncios que o mesmo é autor ou usuário de destino.

Fonte: Autoria própria (2022).

Foram identificados três atores para o sistema: administrador, consumidor e instituição. Além desses três usuários que necessitam de autenticação para realizar as funcionalidades associadas a eles, o sistema possibilitará a consulta aos anúncios por usuários não autenticados. A visualização dos anúncios é a única funcionalidade acessível aos usuários não autenticados no sistema.

O usuário administrador é o único que pode validar o cadastro de instituições. Esse usuário está pré-cadastrado no sistema com permissão de administrador.

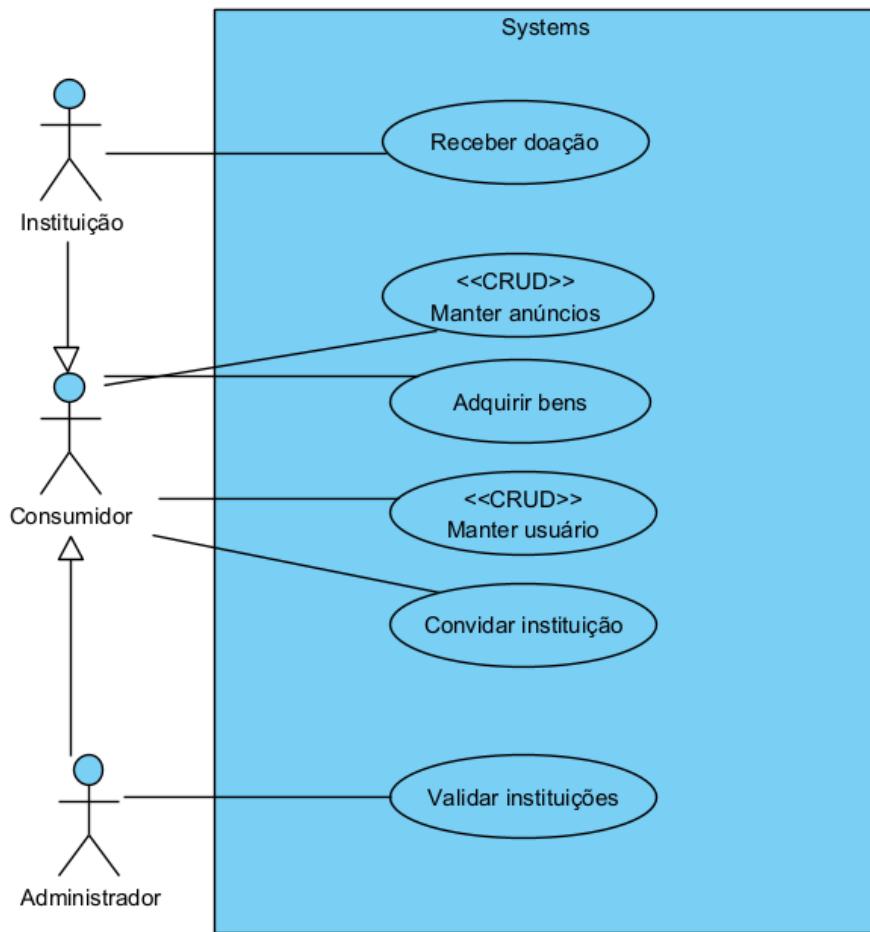
O usuário consumidor possui, como principal objetivo, divulgar bens para venda, empréstimo e doação e adquirir bens por compra, troca ou empréstimo. O usuário consumidor faz a inclusão de bens no sistema para que sejam comprados, trocados, recebidos por doação ou emprestados para outros usuários. Esse usuário pode adquirir bens incluídos no sistema por outros usuários por meio de compra, por troca (quando dois usuários permitem bens entre si) ou tomar por empréstimo (locar bens). O usuário consumidor também pode indicar instituições para que elas se cadastrem no sistema.

O usuário instituição tem como funcionalidade exclusiva a de receber doação de um bem ou do valor monetário resultante da venda do mesmo. Para receber a doação, a instituição precisa ser indicada pelo usuário que cadastrou o referido bem. A instituição realiza o seu próprio cadastro, que deve ser validado pelo administrador do sistema para que a instituição esteja efetivamente cadastrada e possa ser indicada para receber doações.

Os usuários administrador e instituição herdam as funcionalidades do usuário consumidor podendo, portanto, realizar as mesmas operações que este usuário em relação ao objetivo principal do sistema que é disponibilizar bens usados para que possam ser comprados, vendidos, trocados, emprestados ou doados.

Os requisitos apresentados foram organizados na forma de casos de uso, cujo diagrama está na Figura 1. O caso de uso “Adquirir bens” se refere às operações de compra de bens cadastrados no sistema, ao recebimento desses bens por doação, ao empréstimo ou a troca entre bens cadastrados. No cadastro do bem, representado pelo caso de uso “Manter anúncios” na Figura 1, é indicado o propósito do bem sendo cadastrado, atualizado ou excluído, ou seja, venda, troca, empréstimo ou doação. Ressalta-se que um bem colocado para doação pode ser adquirido, apenas por usuários com perfil de instituição.

Figura 1 – Diagrama de casos de uso



Fonte: Autoria própria (2022).

Nos quadros a seguir são apresentados de forma detalhada os requisitos funcionais e os respectivos requisitos não-funcionais relacionados aos casos de uso do sistema apresentados na Figura 1.

Os quadros 13 a 16 apresentam a descrição das ações de inclusão, alteração, exclusão e consulta dos casos de uso manter que pertencem à aplicação servidor e o manter favoritos pertencente à aplicação cliente.

No Quadro 12 está a expansão da operação incluir dos casos de uso manter.

Quadro 12 – Caso de uso manter anúncio e usuário: operação inclusão

Caso de uso:

Incluir (refere-se à operação de inclusão de todos os casos de usos identificados como “manter”).

Descrição:

Inclusão dos dados cadastrais no sistema.

Evento Iniciador:

Autor solicita inclusão de um registro no sistema.

Atores:

Administrador, Instituição e Consumidor.

Pré-condição:

O usuário estar autenticado no sistema.

Sequência de Eventos:

1. Autor acessa a tela para realizar o cadastro inserindo as informações necessárias.
2. O sistema insere as informações no banco de dados e informa ao usuário o *status* do procedimento.

Pós-Condição:

Registro inserido no banco de dados.

Extensões:

Campos obrigatórios faltantes e campos no formato incorreto.

1. Campos obrigatórios não informados.

- 1.1. O usuário deixa de informar dados obrigatórios e clica em salvar.
- 1.2. O sistema valida que não foram informados todos os campos obrigatórios e exibe mensagem ao usuário sem salvar o registro.
- 1.3. O sistema permanece na tela de inclusão mantendo os dados informados anteriormente.
2. Campos informados em formato incorreto.
- 2.1. O usuário informa dados em um formato incorreto e clica em salvar.
- 2.2. O sistema valida que os dados não estão no formato esperado e exibe mensagem ao usuário sem salvar o registro.
- 2.3. O sistema permanece na tela de inclusão mantendo os dados informados anteriormente.

Fonte: Autoria própria (2022).

A descrição da operação que permite alterar os dados dos casos de uso manter é apresentada no Quadro 13.

Quadro 13 – Caso de uso manter anúncio e usuário: operação alteração

<p>Caso de uso: Alterar (refere-se à operação de alteração dos casos de uso manter anúncios e manter usuários).</p> <p>Descrição: Alteração dos dados cadastrais no sistema.</p> <p>Evento Iniciador: Ator solicita alteração de um registro no sistema.</p> <p>Atores: Administrador, Instituição e Consumidor.</p> <p>Pré-condição: Registro estar incluso no sistema e o usuário estar autenticado no sistema.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Ator acessa a tela para visualização dos dados do registro. 2. O sistema apresenta o registro selecionado para alteração. 3. Ator altera os dados do registro. 4. O sistema altera as informações no banco de dados e informa ao usuário o <i>status</i> do procedimento. <p>Pós-Condição: Registro alterado no banco de dados.</p> <p>Extensões: Campos obrigatórios faltantes e campos no formato incorreto.</p> <p>1. Campos obrigatórios não informados.</p> <ol style="list-style-type: none"> 1.1. O usuário apaga dados obrigatórios e clica em salvar. 1.2. O sistema valida que não foram informados todos os campos obrigatórios e exibe mensagem ao usuário sem salvar o registro. 1.3. O sistema permanece na tela de edição mantendo as alterações realizadas. <p>2. Campos informados em formato incorreto.</p> <ol style="list-style-type: none"> 2.1. O usuário altera os dados deixando em um formato incorreto e clica em salvar. 2.2. O sistema valida que os dados não estão no formato esperado e exibe mensagem ao usuário sem salvar o registro. 2.3. O sistema permanece na tela de edição mantendo as alterações realizadas.
--

Fonte: Autoria própria (2022).

O Quadro 14 apresenta a descrição da operação excluir do caso de uso “manter anúncio”.

Quadro 14 – Caso de uso manter anúncio: operação excluir

Caso de uso:

Excluir (refere-se à operação de exclusão do caso de uso manter anúncios).

Descrição:

Exclusão dos dados cadastrais no sistema.

Evento Iniciador:

Autor solicita exclusão de um registro no sistema.

Atores:

Administrador, Instituição e Consumidor.

Pré-condição:

Anúncio estar incluso no sistema, registro com *status* de “Disponível” e o usuário estar autenticado no sistema.

Sequência de Eventos:

1. Autor acessa a tela para exclusão do registro.
2. O sistema exclui as informações no banco de dados e informa ao usuário o *status* do procedimento.

Pós-Condição:

Registro excluído no banco de dados.

Extensões:

Exclusão de registro com *status* diferente de “Disponível”.

1. Registro com *status* “Em Negociação” ou “Finalizado”.

1.1. O sistema verifica que o registro está com *status* “Em Negociação” ou “Finalizado” e desabilita o botão de exclusão.

1.2. O usuário clica em excluir, mas o botão está desabilitado e não ocorre nada.

Fonte: Autoria própria (2022).

No Quadro 15 está a descrição da operação que permite consultar referente aos casos de uso manter.

Quadro 15 – Caso de uso manter anúncio e usuário: operação consultar

Caso de uso:

Consultar (refere-se à operação de consulta dos casos de uso manter anúncios e manter usuários).

Descrição:

Consulta dos dados cadastrais dos registros do sistema.

Evento Iniciador:

Autor solicita consulta de dados armazenados no sistema.

Atores:

Administrador, Consumidor e Instituição. Um usuário não autenticado no sistema pode visualizar os anúncios.

Pré-condição:

Registro estar incluso no sistema e o usuário estar autenticado no sistema para visualização de dados além dos anúncios.

Sequência de Eventos:

1. Autor acessa a tela para visualização dos dados do registro.
2. O autor indica os filtros desejados para consulta.
3. O sistema apresenta os dados da consulta ao usuário.

Pós-Condição:

Dados da consulta apresentados ao usuário.

Fonte: Autoria própria (2022).

No Quadro 16 está a descrição do caso de uso “receber doação”.

Quadro 16 – Caso de uso receber doação

Caso de uso: Receber doação.
Descrição: Esse caso de uso ocorre quando uma instituição assistencial, cadastrada como usuária do sistema, é indicada para receber a doação de um bem. A doação pode ocorrer diretamente, doando-se o bem, ou indiretamente, doando-se o valor oriundo da venda desse bem aos demais usuários do sistema.
Evento Iniciador: Verificar registros de doações recebidas.
Atores: Instituição.
Pré-condição: Instituição ter sido indicada no anúncio do referido bem para receber o valor proveniente da sua venda e venda realizada.
Sequência de Eventos: 1. Ator acessa a tela para visualização dos bens recebidos por doação.
Pós-Condição: Dados da consulta apresentados ao usuário.

Fonte: Autoria própria (2022).

No Quadro 17 está a descrição do caso de uso “Adquirir bens”.

Quadro 17 – Caso de uso adquirir bem

Caso de uso: Adquirir bem.
Descrição: Compra de bens anunciados para venda no sistema.
Evento Iniciador: Ator acessa listagem de anúncios.
Atores: Consumidor, Instituição e Administrador.
Pré-condição: Haver bens disponibilizados para compra por meio do cadastro de anúncios.
Sequência de Eventos: 1. Ator acessa a tela para visualização dos anúncios. 2. Ator pode indicar filtros para apresentação dos anúncios. 3. Sistema apresenta anúncios. 4. Ator indica que quer adquirir bem. 5. Sistema modifica <i>status</i> do bem para “Em Negociação”. 6. Usuário que cadastrou o anúncio indica que o bem foi vendido, e o anúncio fica com <i>status</i> “Finalizado”, ou indica o cancelamento da negociação, o que significa que o bem não foi comprado (vendido), e o anúncio fica disponível para outros usuários
Pós-Condição: <i>Status</i> do bem indicado para aquisição alterado para indicar que o bem está sendo adquirido. Como o sistema não trata os aspectos de pagamento, a finalização do anúncio é realizada pelo usuário anunciante do bem ou o bem é disponibilizado novamente para venda, caso a negociação anterior não tenha ocorrido.

Fonte: Autoria própria (2022).

No Quadro 18 está a descrição do caso de uso “convidar instituição”.

Quadro 18 – Caso de uso convidar instituição

Caso de uso: Convidar instituição.
Descrição: No momento da inclusão do anúncio de um bem para doação a uma instituição assistencial, a instituição pode ainda não estar cadastrada no sistema. Nesse caso o usuário poderá enviar uma mensagem convidando-a a cadastrar-se no sistema. O usuário indica o <i>e-mail</i> da instituição e o sistema envia a mensagem de convite.
Evento Iniciador: Ator quer destinar doação para instituição assistencial que não está cadastrada no sistema.
Atores: Administrador, Consumidor.
Pré-condição: Instituição não estar cadastrada.
Sequência de Eventos:
1. Ator acessa a tela para a inclusão de anúncio e escolhe a opção de doação. No momento de indicar a instituição verifica que ela ainda não está cadastrada no sistema. 2. Ator utiliza a opção que o sistema oferece para o caso de a instituição não estar cadastrada, indica o <i>e-mail</i> da instituição. 3. Sistema envia o <i>e-mail</i> convidando a instituição.
Pós-Condição: <i>E-mail</i> enviado.

Fonte: Autoria própria (2022).

No Quadro 19 está a descrição do caso de uso “validar instituição”. O cadastro de uma instituição precisa ser validado pelo administrador do sistema. Um usuário do tipo “consumidor” pode enviar um convite, por meio do sistema, para realizar o seu cadastro. O convite é uma mensagem enviada por *e-mail* para que a instituição realize o seu cadastro. Não há necessidade de convite para que uma instituição possa cadastrar-se no sistema. Contudo, o cadastro deve ser validado para que a instituição tenha acesso ao sistema e possa ser escolhida para receber doação.

Quadro 19 – Caso de uso validar instituição

Caso de uso:

Validar instituição.

Descrição:

As instituições realizam o próprio cadastro, mas para que possam se autenticar no sistema ou receberem doações (como forma de segurança pelo fato de receber doações de bens e de valor, proveniente da venda dos mesmos) é necessário que o administrador do sistema valide o respectivo cadastro.

Evento Iniciador:

Sistema envia *e-mail* com as instruções e com o *link* para validação do cadastro da instituição.

Atores:

Administrador.

Pré-condição:

Instituição ter efetuado o seu cadastro.

Sequência de Eventos:

1. Ator acessa o *link* recebido por *e-mail*.
2. Ator visualiza os dados da instituição.
3. Ator valida ou não o cadastro.
5. Se o cadastro foi validado, a referida instituição passa a ser listada quando do cadastro de anúncios em que o bem será doado à instituição indicada.

Pós-Condição:

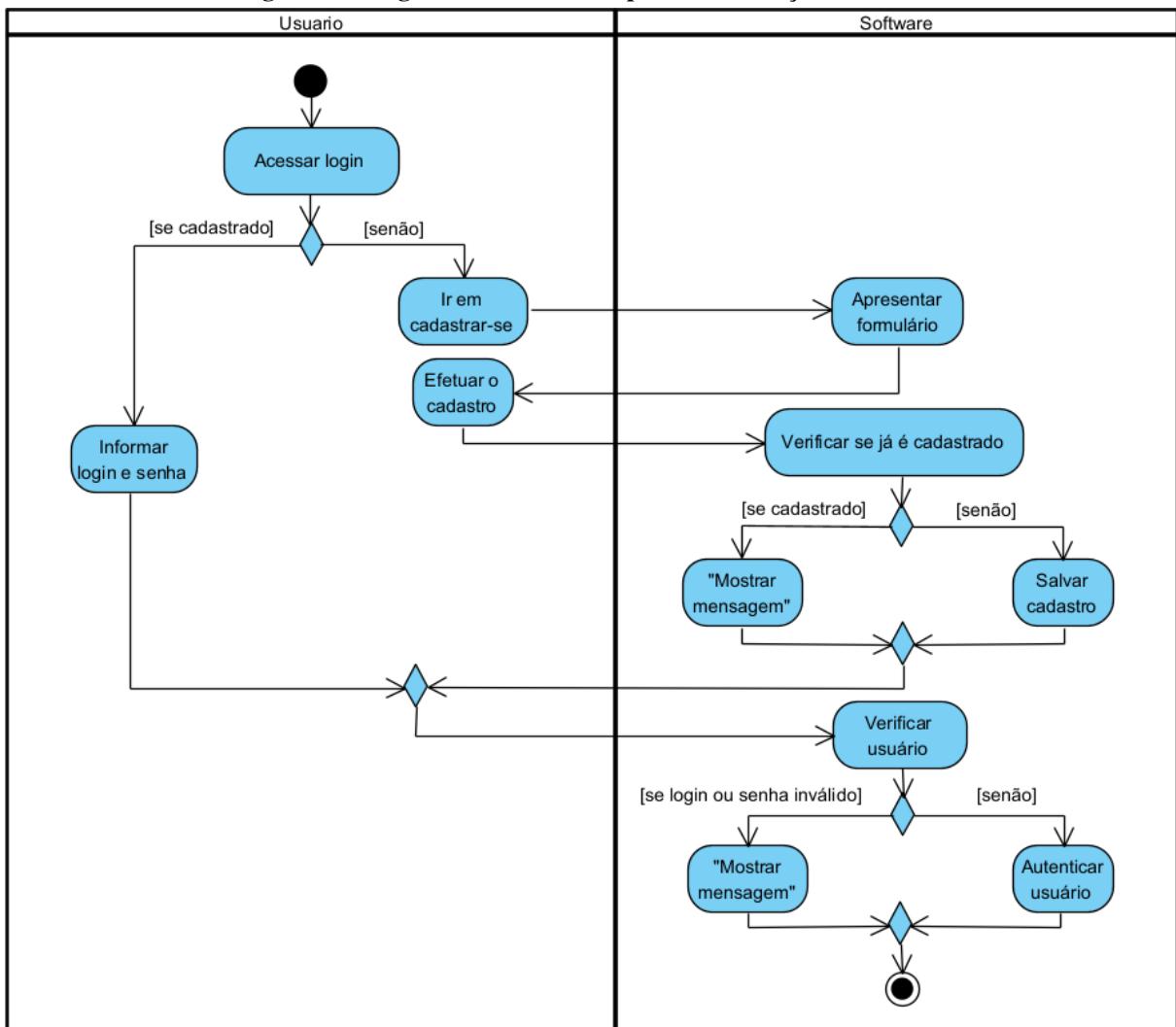
Se o cadastro foi validado, a instituição consegue se autenticar no sistema e passa a compor a lista de instituições que podem receber doações.

Fonte: Autoria própria (2022).

A seguir são apresentados os diagramas de atividades definidos para o sistema. Esses diagramas visam ilustrar de maneira gráfica o funcionamento do sistema, especificando as operações consideradas mais significativas em termos de negócio do sistema.

A Figura 2 apresenta o diagrama de atividades para autenticação no sistema.

Figura 2 – Diagrama de atividades para autenticação no sistema

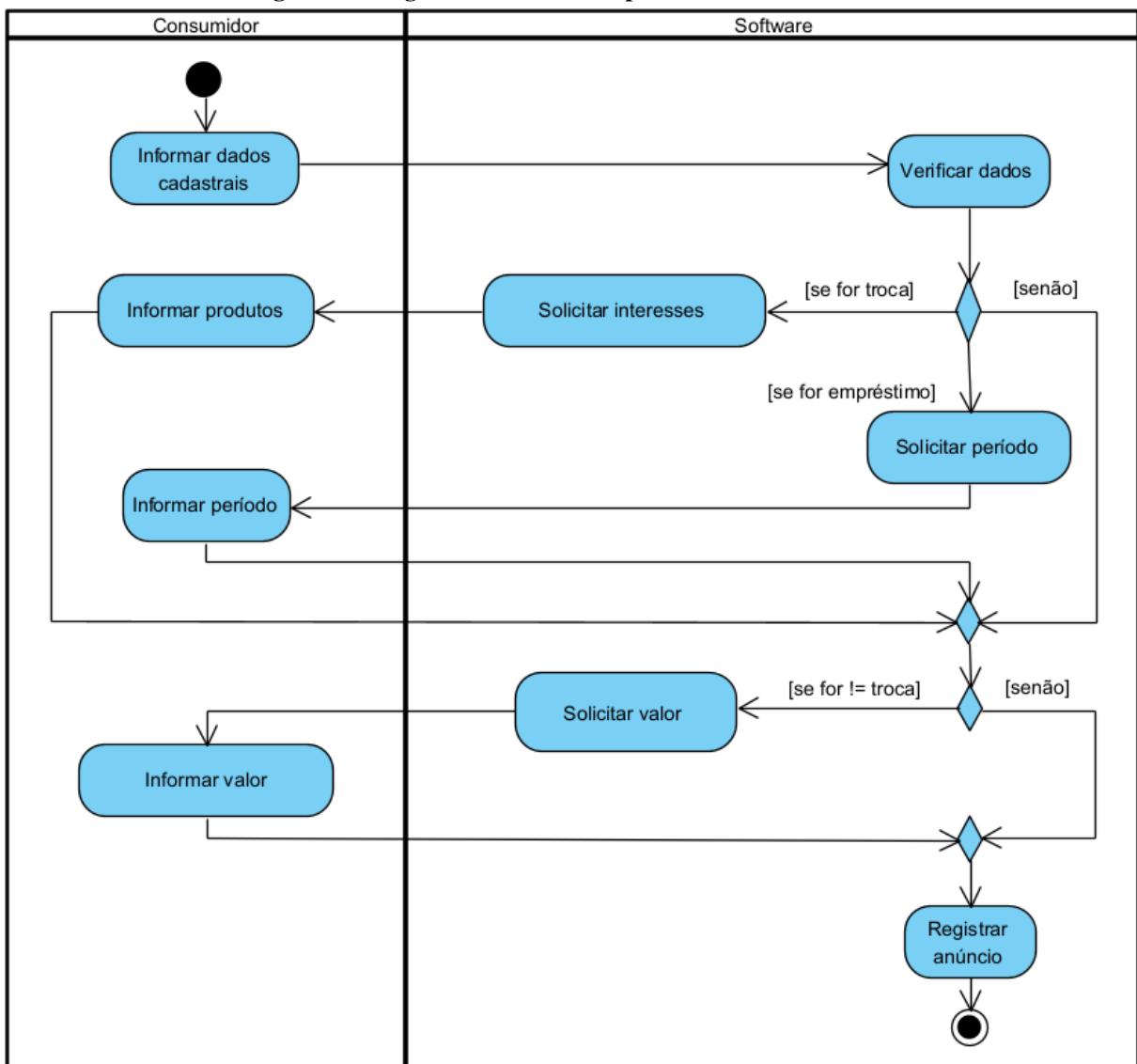


Fonte: Autoria própria (2022).

No diagrama que representa a autenticação do sistema é possível observar que os usuários que já possuem cadastro podem utilizar os dados de nome de usuário e senha para autenticar-se no sistema. Caso o usuário e/ou senha estejam incorretos uma mensagem é exibida. Caso os dados estejam corretos, ou seja, armazenados no banco de dados, o usuário tem acesso ao sistema. Caso o usuário não tenha cadastro no sistema, ele pode fazê-lo por meio de um formulário acessado a partir da tela inicial de acesso ao sistema. O sistema fará a verificação se há um *login* igual ao informado cadastrado no sistema.

Na Figura 3 é apresentado o diagrama de atividades para o cadastro de um anúncio. Nessa atividade, o usuário pode realizar a opção de venda, empréstimo ou trocar por outro bem.

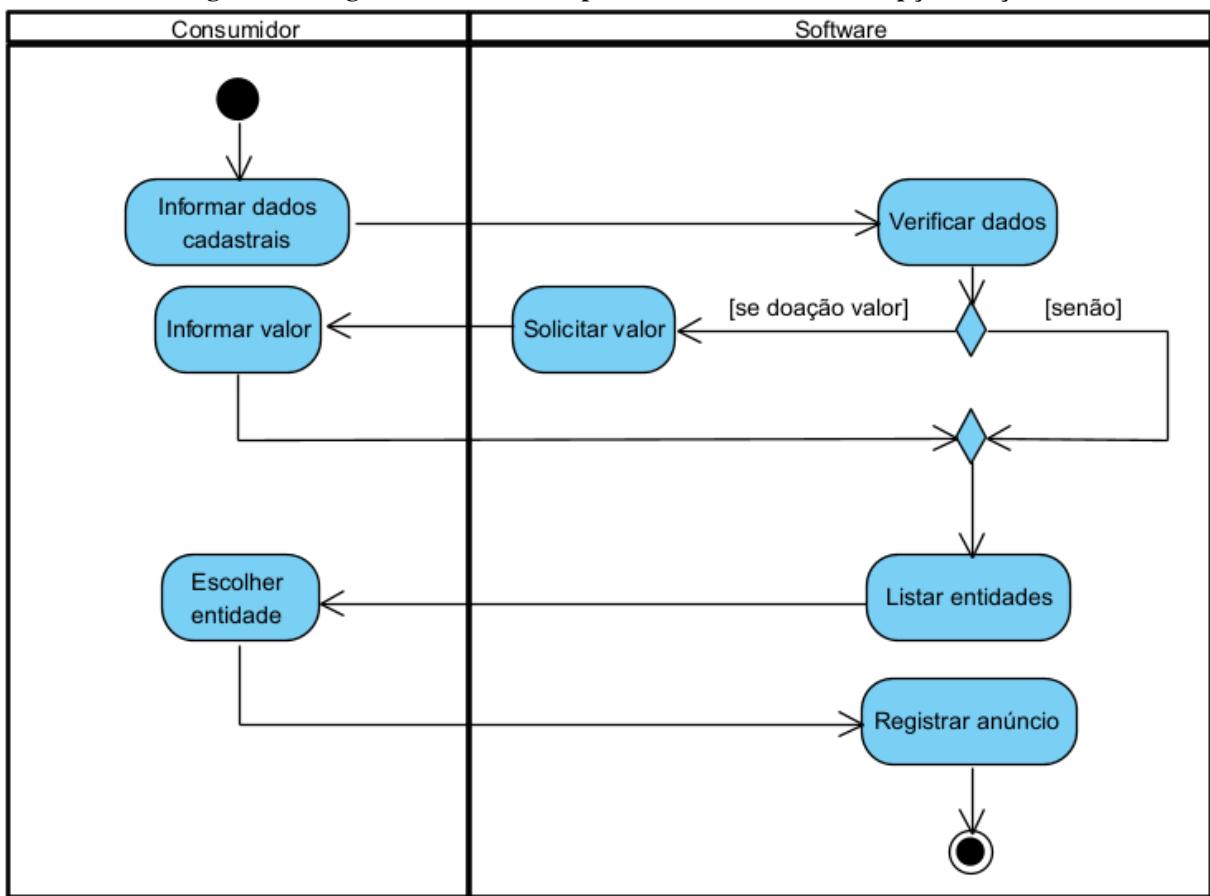
Figura 3 – Diagrama de atividades para cadastro de anúncio



Fonte: Autoria própria (2022).

O cadastro de um anúncio pode ter como objetivo que o bem cadastrado ou o valor obtido com a venda seja destinado para uma entidade assistencial. O diagrama de atividades da Figura 4 apresenta essa operação.

Figura 4 – Diagrama de atividades para cadastro de anúncio: opção doação

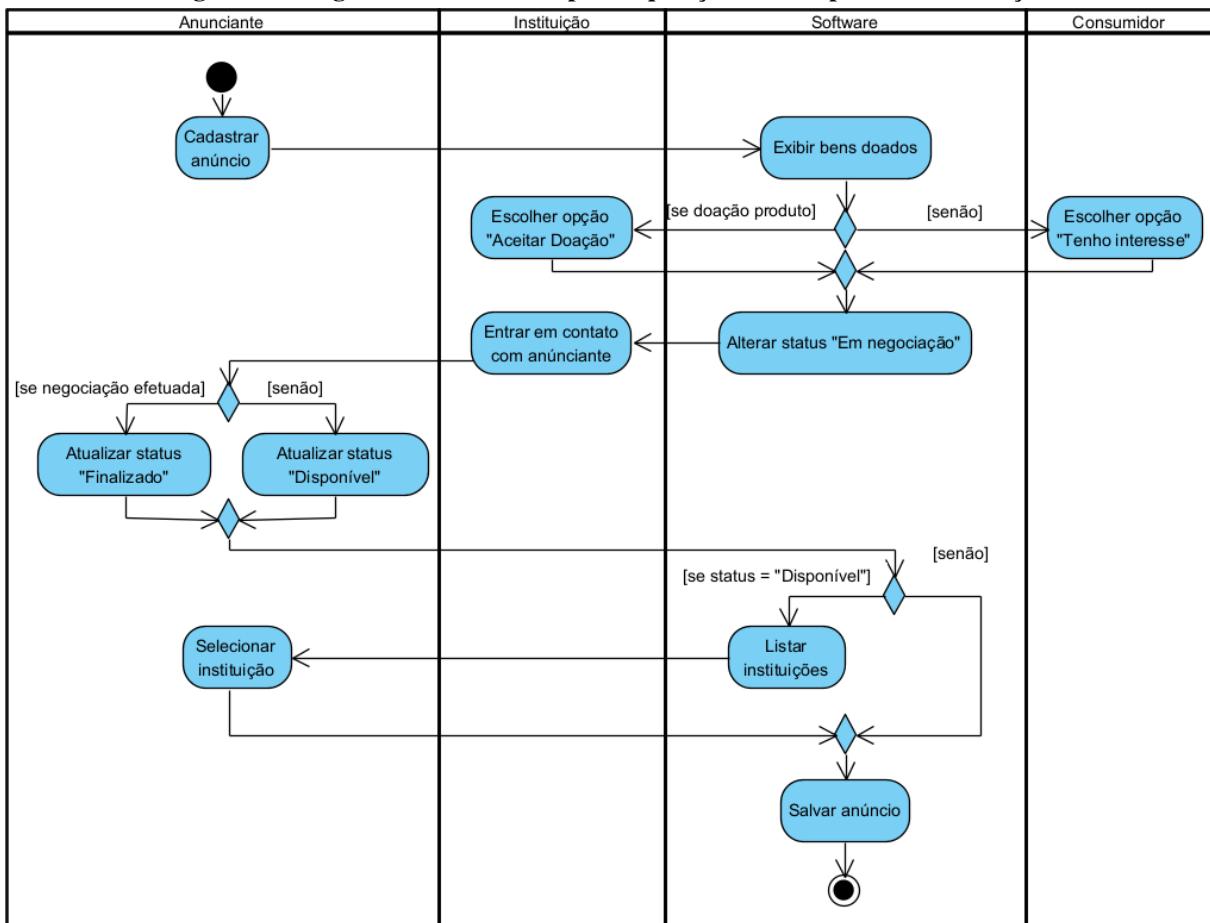


Fonte: Autoria própria (2022).

Conforme mostra a Figura 4, um bem que é cadastrado no sistema ou o valor obtido da venda desse bem, pode ser destinado para uma entidade assistencial, no momento do cadastro o usuário seleciona (escolhe) a entidade que pretende realizar a doação.

Um bem pode ser cadastrado no sistema para que seja doado para instituições. O diagrama de atividades da Figura 5 apresenta a sequência de atividades para aquisição de um bem por meio de doação.

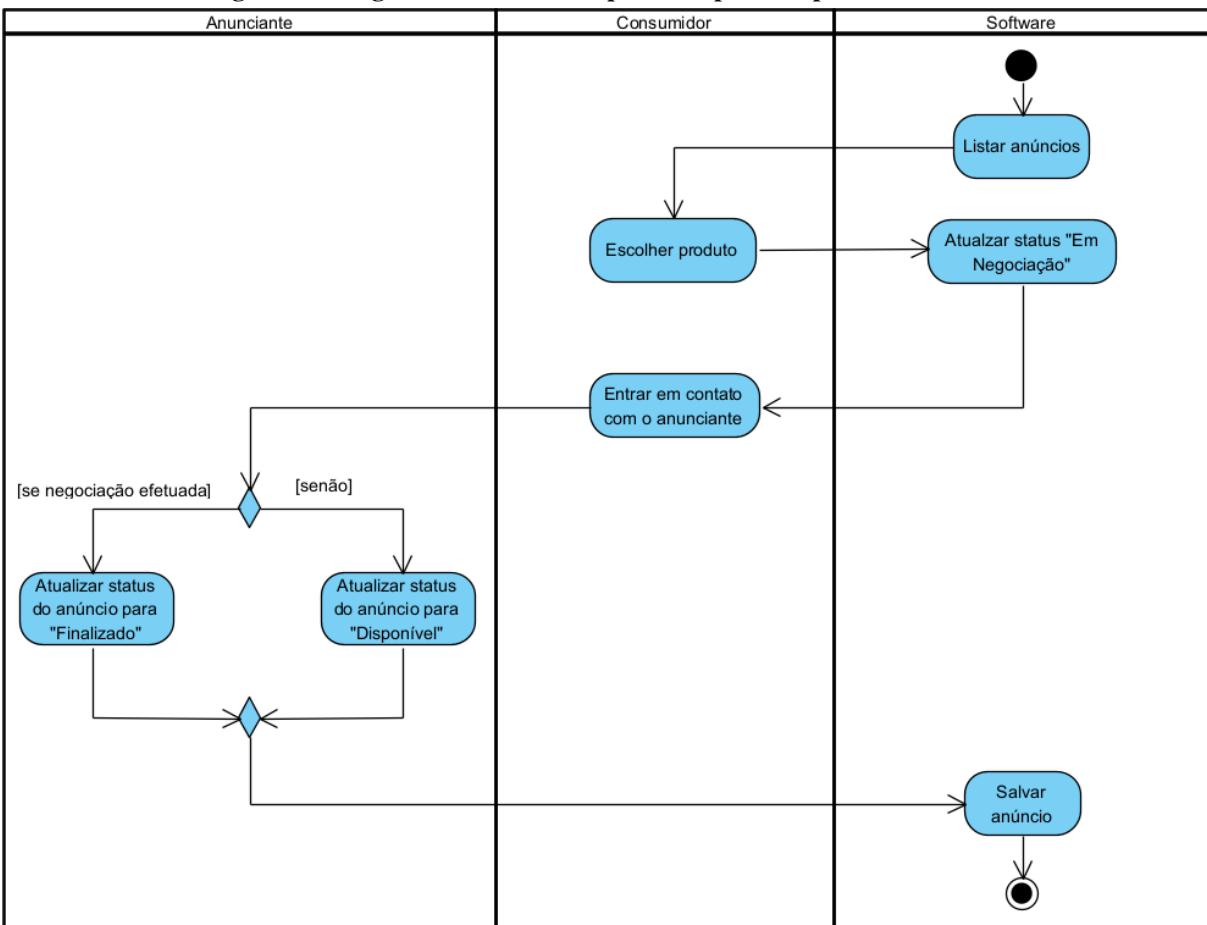
Figura 5 – Diagrama de atividades para aquisição de bem por meio de doação



Fonte: Autoria própria (2022).

O diagrama de atividades da Figura 6 apresenta a sequência de atividades para compra e empréstimo de bens anunciados no site. A compra e o empréstimo são tratados no mesmo diagrama porque são semelhantes em termos operacionais da forma como implementado no sistema. Isso ocorre porque não são realizadas operações financeiras, de entrega e outras pós-venda.

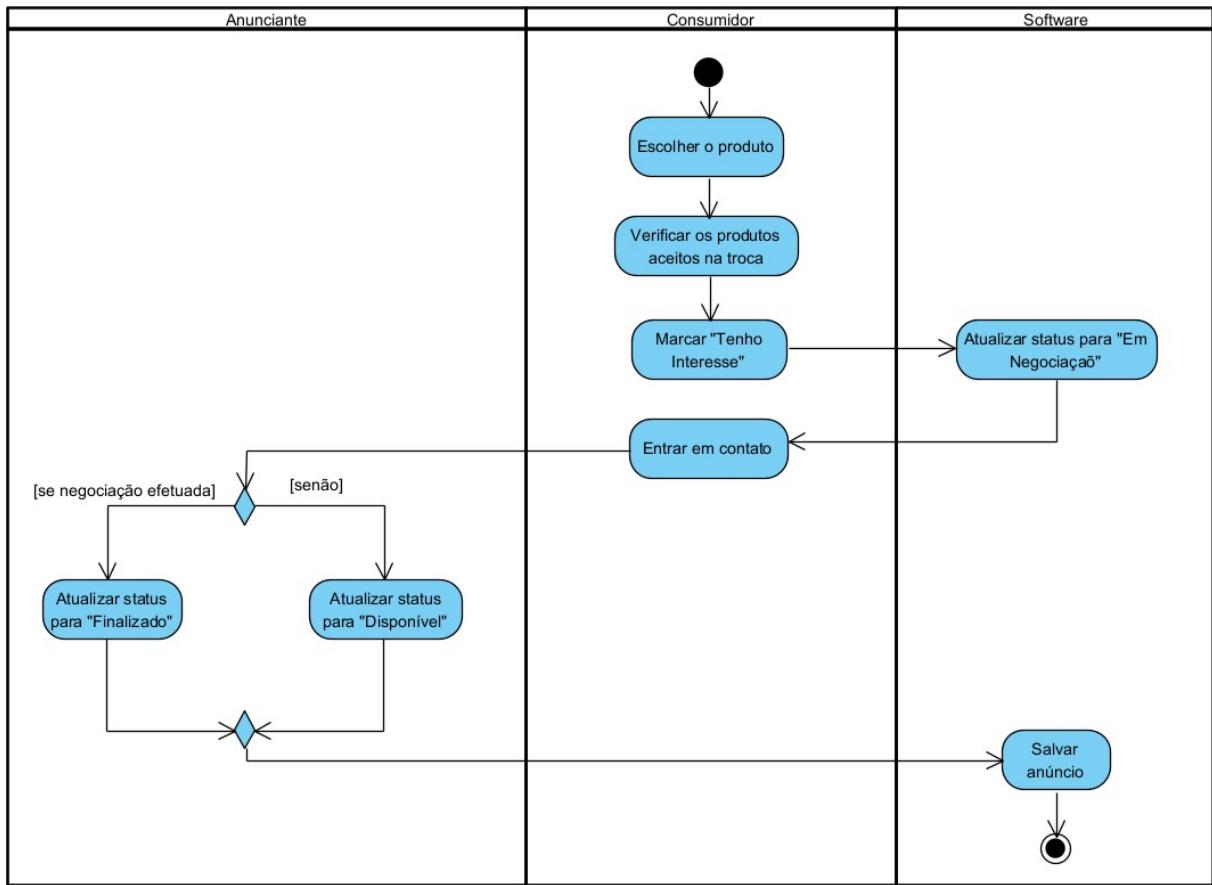
Figura 6 – Diagrama de atividades para compra e empréstimo de bens



Fonte: Autoria própria (2022).

Além de comprar, emprestar e colocar para doação, os bens anunciados no site podem ser trocados entre os usuários. A sequência de atividades para a realização de troca de bens é apresentada na Figura 7.

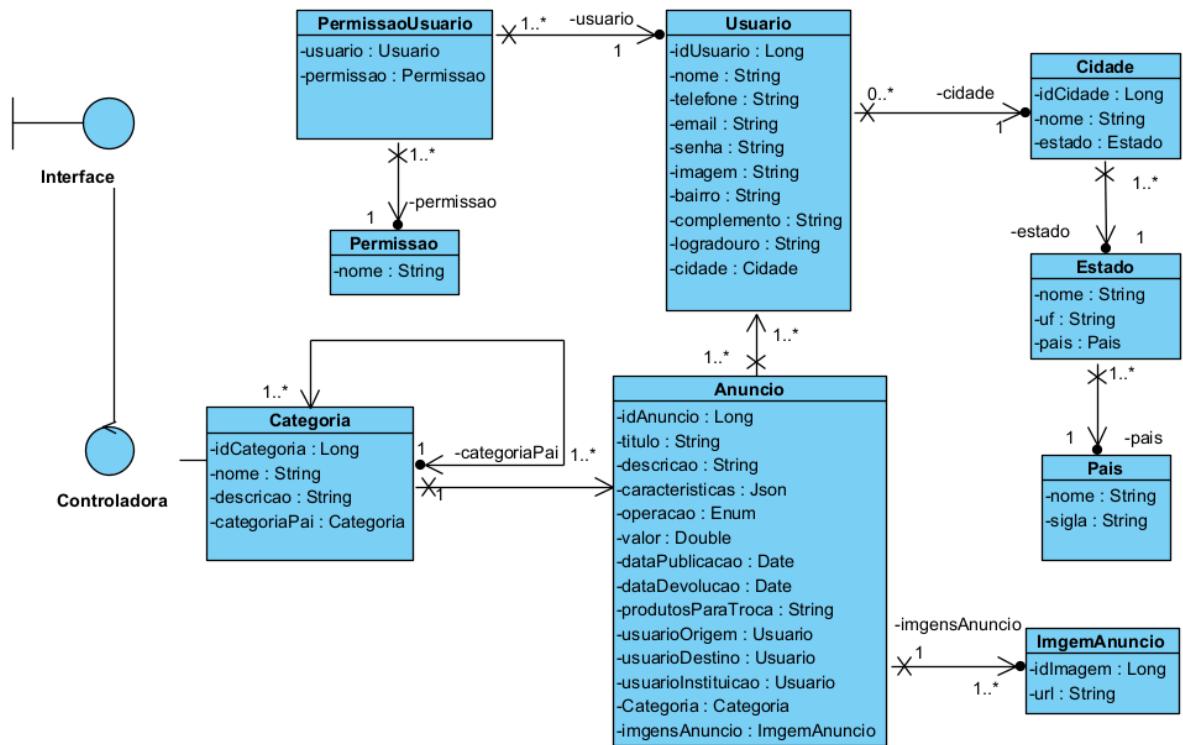
Figura 7 – Diagrama de atividades para troca de bens



Fonte: Autoria própria (2022).

O diagrama de classes é apresentado na Figura 8, representando as classes do projeto e os relacionamentos entre elas, contendo a cardinalidade, a navegabilidade e a visibilidade. Os tipos de dados definidos estão de acordo com a linguagem de programação Java, utilizada no *backend*. O auto-relacionamento contido na tabela categoria, representa a possibilidade de definir uma hierarquia de categorias. Isso, no entanto, ficará apenas modelado para ser implementado posteriormente a este trabalho.

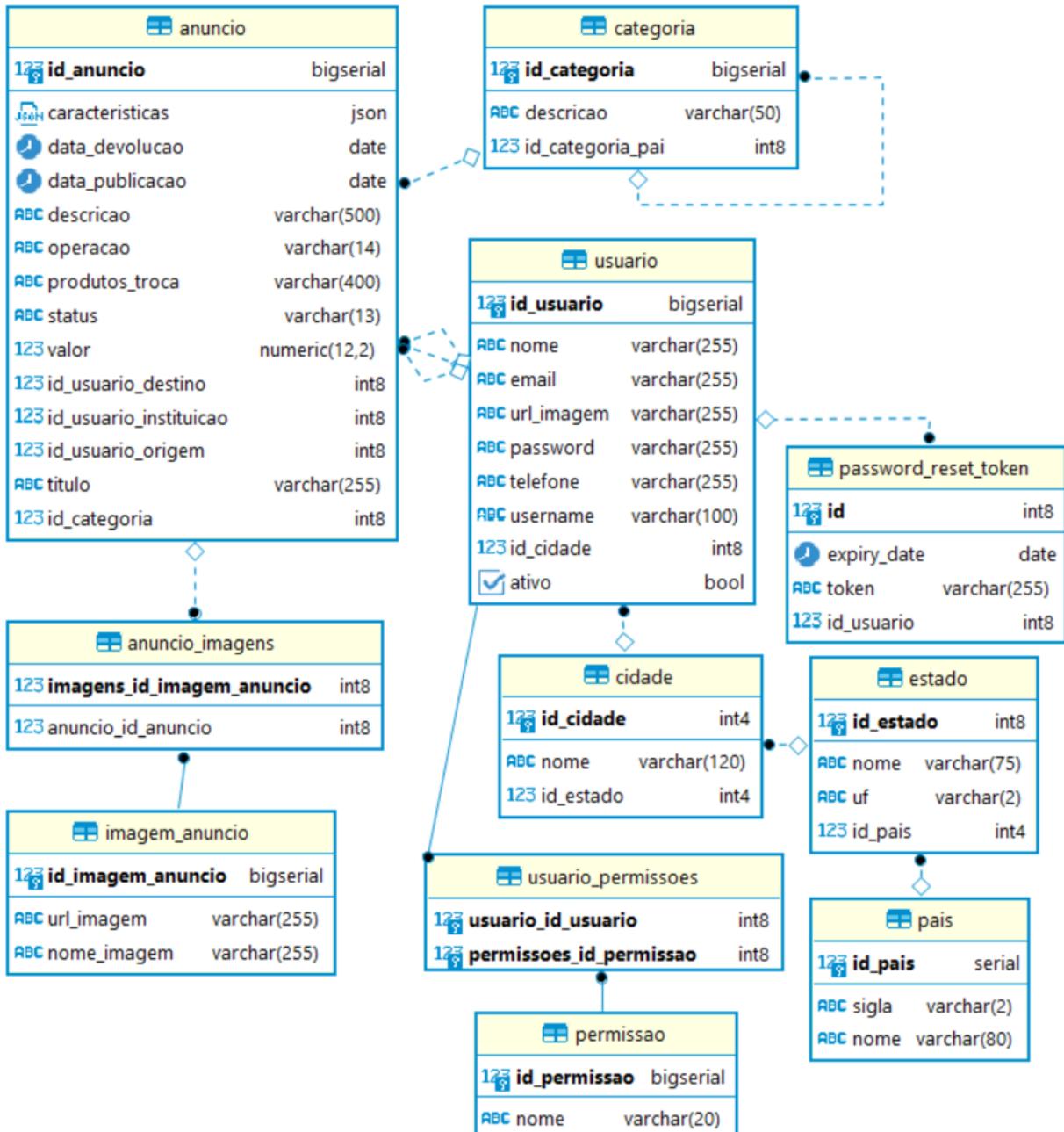
Figura 8 – Diagrama de classes



Fonte: Autoria própria (2022).

A Figura 9 apresenta o diagrama de entidades e relacionamentos do banco de dados. Esse diagrama define os dados que serão persistidos no sistema, com as respectivas tabelas e relacionamentos entre elas.

Figura 9 – Diagrama de entidades e relacionamentos do banco de dados



Fonte: Autoria própria (2022).

No diagrama do banco de dados apresentado na Figura 9, anúncio possui relacionamento N:1 com a categoria, essa que possui relacionamento tipo 1:0 com categoria pai e uma categoria pai tem relacionamento 1:N com categoria. Essa forma de auto-relacionamento na tabela categoria foi definida para que futuramente possam ser criadas categorias dinamicamente e que possa ser realizado o relacionamento entre elas por meio da indicação de qual categoria (categoria pai) cada subcategoria pertence. Possibilitando, assim, que seja definida uma hierarquia entre as categorias. Inicialmente o sistema conterá as categorias predefinidas: livro, eletrodoméstico, vestuário e outra.

Um registro da tabela anúncio, relaciona-se três vezes com a usuário, tendo a cardina-

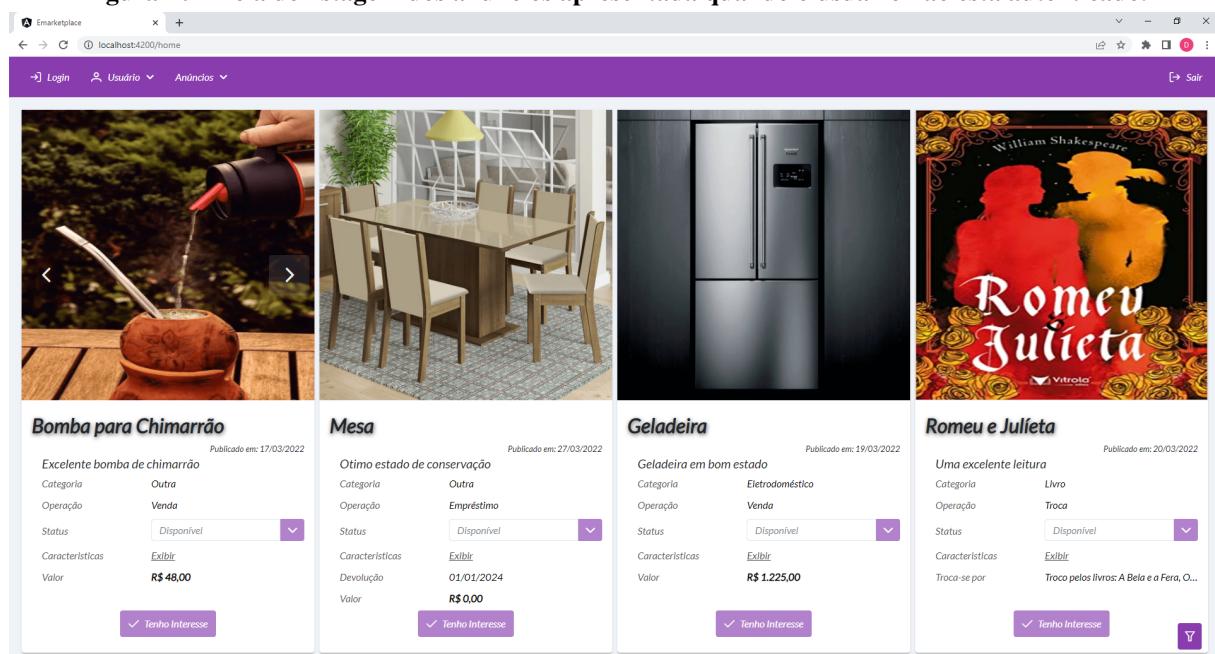
lidade N:1, sendo um relacionamento para representar o usuário de origem (quem publicou o anúncio), outro para o usuário de destino (quem está recebendo o bem) e o terceiro relacionamento para identificar o usuário instituição (quem está recebendo a doação), que é necessário devido a possibilidade de que um anúncio vendido tenha o valor dessa venda destinado (doado) para uma instituição. Outro relacionamento é entre a tabela anúncio e a tabela que armazena as imagens do anúncio, que possui cardinalidade N:1. As imagens do anúncio e do usuário são hospedadas em nuvem, utilizando o *Amazon Simple Storage Service* (S3), sendo armazenada a *Uniform Resource Locator* (URL) dessas imagens para que possam ser exibidas na aplicação.

A tabela usuário relaciona-se com a tabela cidade, com cardinalidade N:1. O usuário também possui relacionamento com a tabela permissão, sendo de N:N tendo a necessidade de uma tabela de junção. Existe também um relacionamento N:1, entre a tabela usuário e a tabela usada para redefinição da senha, em que é armazenado qual o usuário está vinculado ao *token* e a data de expiração.

4.3 APRESENTAÇÃO DO SISTEMA

As telas apresentadas nas Figuras a seguir mostram as principais funcionalidades implementadas no sistema. A tela inicial do sistema, apresentada na Figura 10, é a de listagem de anúncio.

Figura 10 – Tela de listagem dos anúncios apresentada quando o usuário não está autenticado.

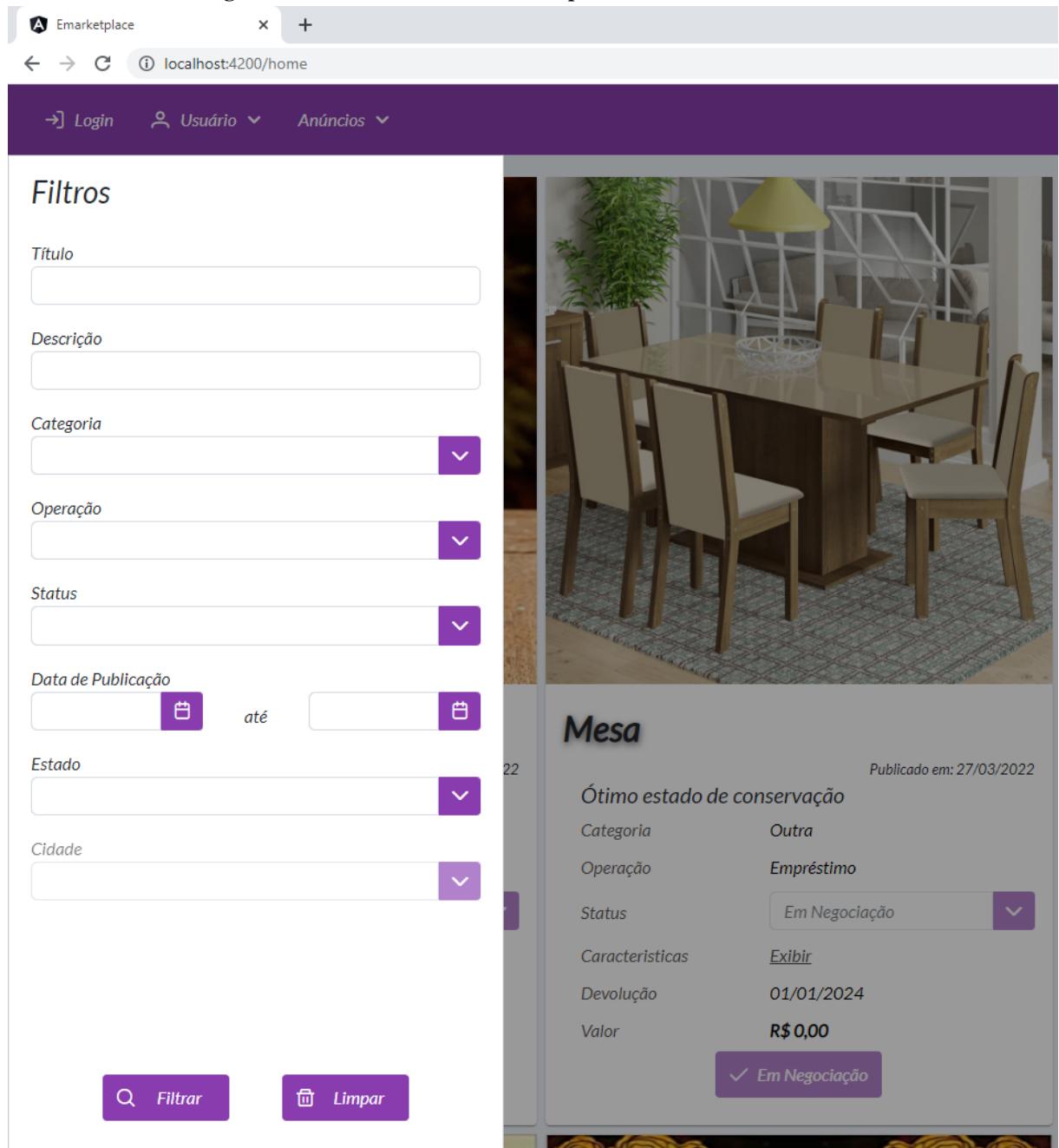


Fonte: Autoria própria (2022).

Na tela apresentada na Figura 10, para visualizar as seguintes informações: título, descrição, data de publicação, categoria, operação, *status*, características, valor, data de devolução e produtos aceitos na troca, não é necessário que o usuário esteja autenticado no sistema. Po-

réim, não é possível adquirir bens ou visualizar quem realizou os anúncios. Ao clicar no botão “suspenso”, localizado no canto inferior direito da tela, são exibidos os filtros, como mostrado na Figura 11.

Figura 11 – Tela de filtro visualizada por usuários não autenticados



Fonte: Autoria própria (2022).

Na tela apresentada na Figura 11, é realizada a busca se o texto informado está contido nos campos título e descrição do anúncio. É possível, ainda, pesquisar nos campos categoria, operação, *status*, estado e cidade, se há correspondência com a informação selecionada. Esses campos são *autocomplete* possibilitando ao usuário filtrar as opções e escolher a que desejar. O campo cidade é habilitado apenas após a seleção do Estado, sendo possível filtrar apenas

pelo Estado, ou especificar a busca por meio da Cidade. Pode-se ainda filtrar de acordo com a data de publicação, verificando se a data é maior, menor ou se está entre um intervalo de datas informadas.

Ao clicar no botão para filtrar os dados, o sistema exibe a animação apresentada na Figura 12.

Figura 12 – Tela exibida enquanto a busca é realizada



Fonte: Autoria própria (2022).

Essa animação é exibida sempre que o usuário estiver aguardando um retorno da aplicação, alterando apenas a mensagem de acordo com a situação.

A Figura 13 apresenta a tela de listagem de anúncios quando o usuário está autenticado.

Figura 13 – Tela de listagem dos anúncios visualizada por usuários autenticados

 A screenshot of a web browser displaying a marketplace interface. The title bar says 'Emarketplace' and the address bar shows 'localhost:4200/home'. The header includes links for 'Login', 'User', and 'Anúncios'. Below the header, there are four cards representing different listings:

- Bomba para Chimarrão**: An image of a mate bomba being filled with hot water. Details: Published on 17/03/2022, Category: Outra, Operation: Venda, Status: Em Negociação. Buttons: 'Acessar Cadastro' (purple), 'Exibir' (blue).
- Mesa**: An image of a dining table and chairs. Details: Published on 27/03/2022, Category: Outra, Operation: Emprestimo, Status: Em Negociação. Buttons: 'Em Negociação' (purple), 'Exibir' (blue).
- Romeu e Julieta**: An image of a book cover. Details: Published on 20/03/2022, Category: Livro, Operation: Troca, Status: Disponível. Buttons: 'Tenho Interesse' (purple), 'Exibir' (blue).
- Geladeira**: An image of a side-by-side refrigerator. Details: Published on 27/03/2022, Category: Eletrodoméstico, Operation: Doação do Produto, Status: Disponível. Buttons: 'Aceitar Doação' (purple), 'Exibir' (blue).

Fonte: Autoria própria (2022).

Quando o usuário autenticado é o anunciante, como no primeiro anúncio apresentado na Figura 13, ele pode acessar a tela de cadastro ao clicar na opção “Acessar Cadastro”. Caso o anúncio esteja com o *status* “Em Negociação”, é possível visualizar qual usuário possui interesse em adquirir o bem e entrar em contato. Uma vez realizada a negociação, o anunciante pode mudar o *status* para “Finalizado”, se houver negociação, ou para “Disponível”, caso a negociação não ocorra.

Caso o usuário autenticado não seja o anunciante, como no segundo e terceiro anúncio presente na Figura 13, é apresentada a opção “Tenho Interesse”, se o anúncio estiver com o *status* “Disponível”. Ao clicar na opção “Tenho Interesse” o anúncio é vinculado ao usuário e o *status* e o texto do botão são atualizados para “Em Negociação”, sendo possível desfazer a ação ao clicar novamente sobre o botão.

Se o usuário autenticado for uma instituição e o anúncio possuir a operação “Doação Produto”, como no quarto anúncio da Figura 13, é apresentada a opção “Aceitar Doação”. Ao clicar sobre o anúncio o *status* é atualizado para “Em Negociação”, sendo possível desfazer a ação ao clicar novamente. Caso a operação seja “Doação Valor”, esse processo não ocorre porque é considerado que todas as instituições cadastradas aceitam receber valores.

A Figura 14 apresenta a tela de filtros quando o usuário está autenticado.

Figura 14 – Tela de filtro disponibilizada para os usuários autenticados

The screenshot shows a web application interface for filtering items. On the left, there is a sidebar with navigation links: 'Login' (with a user icon), 'Usuário' (with a user icon), and 'Anúncios' (with a document icon). Below these are several filter fields:

- Título:** An input field for entering a title.
- Descrição:** An input field for entering a description.
- Categoria:** A dropdown menu for selecting a category.
- Operação:** A dropdown menu for selecting an operation.
- Status:** A dropdown menu for selecting a status.
- Data de Publicação:** A date range selector with two input fields and a separator 'até'.
- Estado:** A dropdown menu for selecting a state.
- Cidade:** A dropdown menu for selecting a city.
- Anunciei:** A checkbox for filtering items I have listed.
- Adquiri:** A checkbox for filtering items I have acquired.

At the bottom of the sidebar are two buttons: 'Filtrar' (Filter) and 'Limpar' (Clear).

The main content area features a large image of a traditional bomba para chimarrão (mate gourd) with a bombilla (straw) and steam rising from it, resting on a wooden surface. To the right of the image is a product listing:

Bomba para Chimarrão

Published on: 17/03/2022

Excellent mate gourd

Categoria	Outra
Operação	Venda
Status	Em Negociação
Características	Exibir
Anunciante	Exibir
Valor	R\$ 48,00

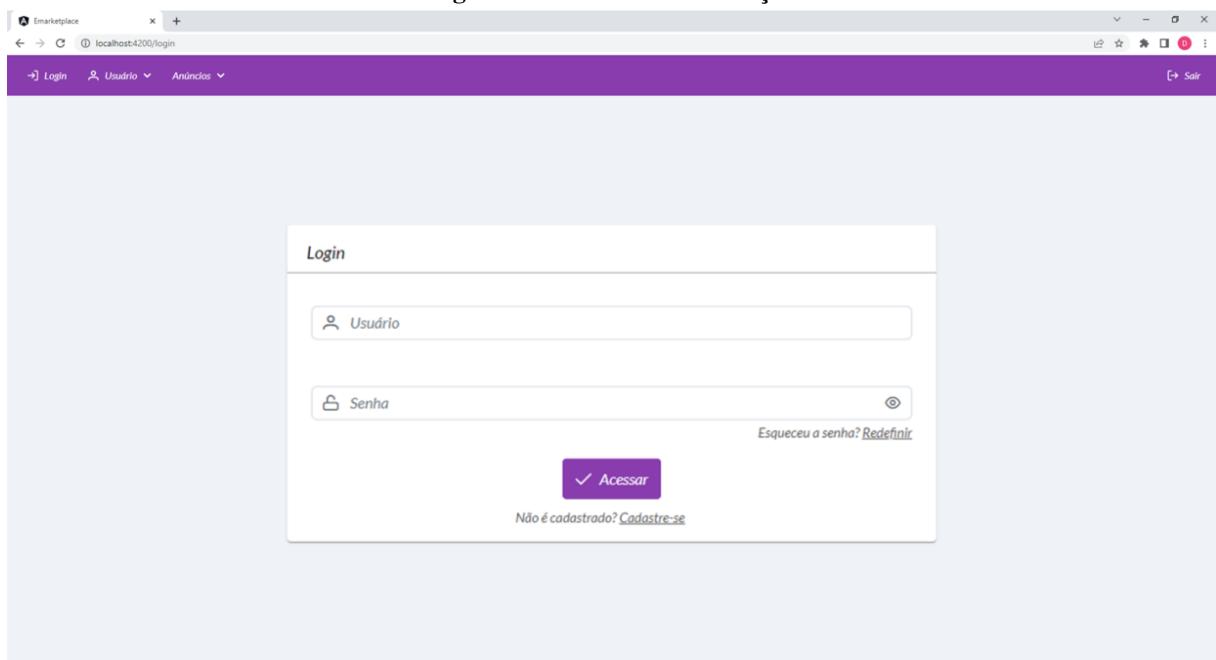
[Em Negociação](#)

Fonte: Autoria própria (2022).

A diferença apresentada na tela de filtro para usuários autenticados é a possibilidade de filtrar apenas pelos itens anunciados pelo usuário autenticado, ou adquiridos, seja por meio de compra, troca, empréstimo ou doação (no caso de tratar-se de uma instituição).

A Figura 15 apresenta a tela de autenticação do usuário no sistema.

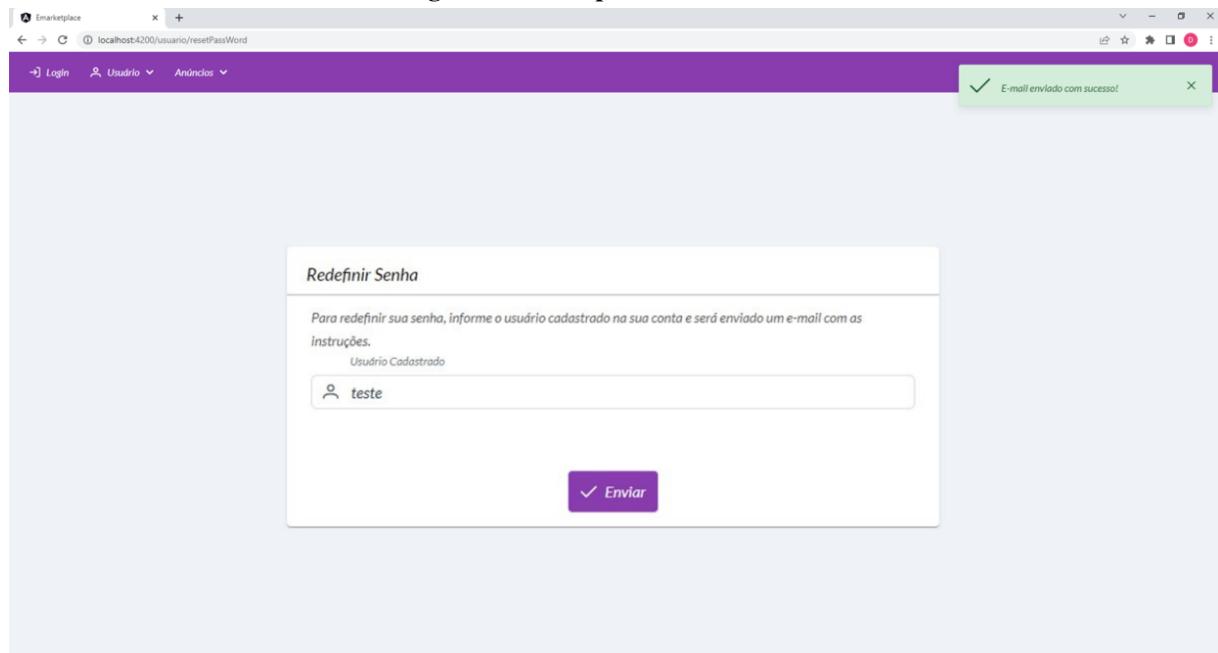
Figura 15 – Tela de autenticação



Fonte: Autoria própria (2022).

A autenticação do usuário no sistema é realizada após o preenchimento dos campos de usuário e senha, informados no momento do cadastro. No entanto, se o usuário cadastrado for uma instituição, só é possível realizar a autenticação após a validação do cadastro por um administrador do sistema. Após clicar no botão “Acessar”, se autenticado, o botão fica desabilitado e o usuário é redirecionado para a tela de listagem dos anúncios. Caso o usuário não possua cadastro, ele pode clicar na opção “Cadastre-se” e será redirecionado para a tela de cadastro. Para redefinir a senha há a opção “Redefinir”, conforme apresentado na Figura 16.

Figura 16 – Tela para redefinir senha



Fonte: Autoria própria (2022).

Nessa tela, após informado o usuário cadastrado e clicado na opção “Enviar”, o sistema envia um *e-mail* para o usuário com um *link* contendo um *token* vinculado a esse usuário e com validade de um dia. Após o envio, o sistema exibe uma mensagem informando ao usuário o *status* de sua requisição. Esse padrão de mensagens está presente em todas as requisições do sistema, visando, assim, melhorar a experiência do usuário.

A Figura 17 apresenta o *e-mail* que é enviado para o usuário na solicitação de redefinição de senha.

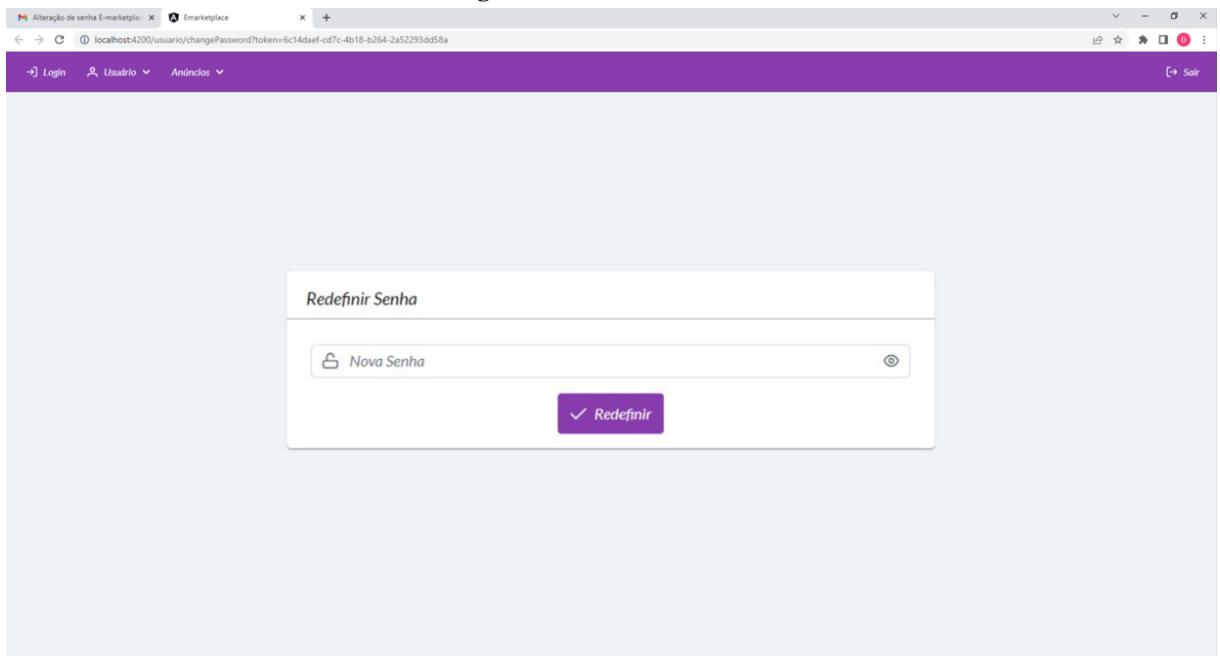
Figura 17 – E-mail enviado para redefinir senha



Fonte: Autoria própria (2022).

A Figura 18 apresenta a tela acessada por meio do *link* enviado no *e-mail*.

Figura 18 – Tela nova senha



Fonte: Autoria própria (2022).

Ao clicar na opção para redefinir a senha, o usuário é buscado pelo *token*, que deve ser válido e não estar expirado.

A Figura 19 apresenta o retorno da aplicação quando a senha é redefinida com sucesso e quando o *token* não é válido.

Figura 19 – Mensagens exibidas na atualização de senha



Fonte: Autoria própria (2022).

A mensagem da esquerda na Figura 19 é exibida se a senha foi atualizada com sucesso, a mensagem da direita é exibida quando o *token* não é válido ou já estiver expirado (mais de um dia).

A Figura 20 apresenta o componente de senha utilizado no sistema, que faz parte da biblioteca de componentes PrimeNG e foi adaptado/customizado para atender as necessidades da aplicação. Se o usuário informar uma senha que não atenda aos requisitos estabelecidos, o campo de entrada fica em vermelho e não é aceito pelo sistema.

Figura 20 – Componente de senha

The figure consists of three separate screenshots of a password strength checker component, likely from a PrimeNG library. Each screenshot shows a form with a password input field and a strength indicator below it.

- Top Screenshot:** The title is "Redefinir Senha". The password input field contains "teste12". Below it is a progress bar with a yellow segment and a grey segment. The word "Média" is displayed next to the bar. A list titled "A senha deve conter:" includes:
 - Pelo menos uma letra minúscula
 - Pelo menos uma letra maiúscula
 - Pelo menos um número
 - No mínimo 6 caracteres
- Middle Screenshot:** The title is "Redefinir Senha". The password input field contains "Fraca". Below it is a progress bar with a very short red segment and a long grey segment. The word "Fraca" is displayed next to the bar. The same list of requirements is shown.
- Bottom Screenshot:** The title is "Redefinir Senha". The password input field contains "Teste123". Below it is a progress bar with a long green segment and a very short grey segment. The word "Forte" is displayed next to the bar. The same list of requirements is shown.

Fonte: Autoria própria (2022).

A Figura 20 caracteriza a complexidade da senha. Essa complexidade é medida por indicadores que classificam a senha como, fraca, média e forte. Para uma senha ser forte, ela precisa atender aos seguintes requisitos: ter no mínimo uma letra minúscula e uma maiúscula, conter no mínimo um número e seis caracteres. Quando a senha atende somente a um dos requisitos, ela é classificada como fraca e quando somente dois requisitos são atendidos ela é classificada como média.

A Figura 21 apresenta a tela de cadastro do usuário.

Figura 21 – Tela de cadastro de usuário

The screenshot shows a web-based user registration form titled 'Figura 21 – Tela de cadastro de usuário'. The interface is divided into several sections:

- Dados Pessoais:** Contains a placeholder profile picture with edit and delete icons, an 'Instituição' field (checkbox), and an 'Apelido' field containing 'Ususdr0 Teste'.
- Contato:** Includes 'E-mail *' (teste@teste.com) and 'Telefone *' ((49) 9 9999-9999).
- Endereço:** Shows 'Estado *' (Paraná) and 'Cidade *' (Pato Branco).
- Dados para Login:** Features 'Usuário *' (usuario_teste) and 'Senha *' (*****).
- Buttons:** A purple 'Salvar' (Save) button at the bottom.

Fonte: Autoria própria (2022).

O acesso à tela de cadastro de usuário, apresentado na Figura 21, pode ser realizado por meio da opção “Cadastrar-se” da tela de autenticação, ou acessando a opção “Usuário > Cadastre-se”. No caso das instituições, uma possibilidade é de que os usuários, por meio do sistema, enviem um convite, para que seja possível visualizar essa tela ao acessar o *link* enviado no *e-mail*.

Nessa tela, é possível adicionar uma imagem. Ao colocar o cursor do *mouse* sobre a imagem, são exibidas duas opções: a primeira para editar e a segunda para atribuir a imagem padrão. Na Figura 21, a segunda opção encontra-se desabilitada, pois o usuário ainda não selecionou uma imagem. A imagem do usuário fica hospedada em nuvem, utilizando o S3.

O campo instituição é utilizado para atribuir a permissão de instituição. Se esse campo estiver marcado, o campo “Apelido” é alterado para “Razão Social”. Nesse caso, só é possível autenticar-se no sistema após a validação de um usuário administrador. Os campos de contato são coletados para que o comprador possa visualizar quem está anunciando um item. Assim, ele poderá entrar em contato por meio do *e-mail* ou do telefone cadastrado. Os dados de endereço são utilizados para que o usuário que está adquirindo um produto visualize e possa filtrar a sua localização. Por fim, são coletados os dados para que o usuário se autentique na aplicação.

Essa mesma tela é utilizada quando o usuário autenticado no sistema acessa “Usuário > Perfil”. A única diferença é que a informação presente no quadro “Dados para Login” fica oculta. Na Figura 22, pode-se observar os possíveis retornos da aplicação, ao clicar no botão “Salvar”.

Figura 22 – Mensagens exibidas no cadastro do usuário



Fonte: Autoria própria (2022).

Na Figura 22, a mensagem da esquerda ocorre quando o registro foi salvo com sucesso, o retorno do centro ocorre quando no formulário há campos obrigatórios não preenchidos. A última validação ocorre quando o usuário escolhido já está em uso.

A Figura 23 apresenta o *e-mail* recebido pelos administradores do sistema após o cadastramento de uma instituição.

Figura 23 – E-mail para validar cadastro da instituição

Validação de cadastro E-marketplace ➤ Caixa de entrada x



emarketplace.gerencia@gmail.com
para mim ▾

Olá,
Favor validar o cadastro da instituição, acessando o link:
<http://localhost:4200/usuario/form?id=58>

Fonte: Autoria própria (2022).

Uma vez o usuário estando autenticado como administrador na aplicação, ao acessar o *link* enviado por *e-mail* ele é direcionado para tela de validação, apresentada na Figura 24.

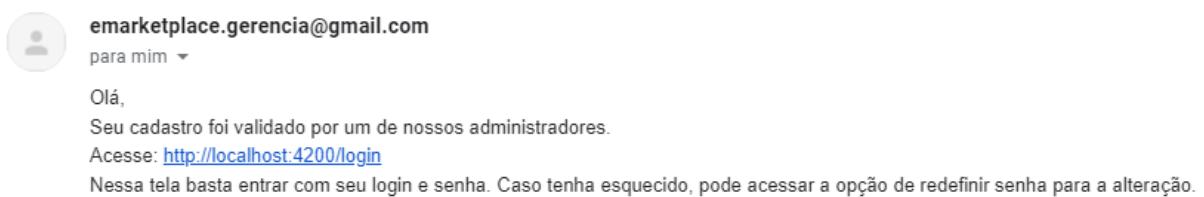
Figura 24 – Tela validação do cadastro da instituição

Fonte: Autoria própria (2022).

Após o usuário realizar a validação, ao clicar em “Validar Cadastro” é exibida a mensagem “Registro validado com sucesso” e o botão é desabilitado. Caso o usuário tente acessar novamente essa tela é carregado o seu próprio perfil. Isso ocorre para que usuários não visualizem informações de outros usuários mudando a URL da aplicação. A única exceção é quando o usuário autenticado é um administrador e o usuário que está tentando carregar é uma instituição com o cadastro ainda não validado. Após validado o cadastro da instituição ela é notificada via *e-mail*, como mostra a Figura 25.

Figura 25 – E-mail informando a instituição que o cadastro foi validado

Bem vinda ao E-marketplace! ➤



Fonte: Autoria própria (2022).

Acessando o *link* enviado no *e-mail* o usuário será direcionado para tela de *login* e poderá autenticar-se na aplicação.

A Figura 26 apresenta a seção cabeçalho na tela de cadastro de um anúncio, sem que possua uma operação selecionada.

Figura 26 – Seção “Cabeçalho” da tela cadastro de anúncio

Fonte: Autoria própria (2022).

Na seção de cabeçalho são coletadas as seguintes informações: título, descrição, categoria e operação. Com a seleção do campo “Operação” são exibidos novos campos de acordo com a seleção.

A Figura 27 apresenta a seção do cabeçalho na tela de cadastro de um anúncio, quando selecionada a operação referente a empréstimo.

Figura 27 – Tela cadastro de anúncio com a operação “Empréstimo”

Fonte: Autoria própria (2022).

Nessa operação é coletada a informação referente ao valor, que aceita zero, para o caso de o anunciante não querer remuneração pelo empréstimo e a data de devolução. Caso o anunciante prefira, poderá realizar o empréstimo sem uma data de devolução definida.

A Figura 28 apresenta a seção cabeçalho quando selecionada a operação referente à troca.

Figura 28 – Tela cadastro de anúncio com a operação “Troca”

The screenshot shows a web-based application interface for creating an advertisement. At the top, there's a purple header bar with navigation links for 'Login', 'Usuário', and 'Anúncios'. Below the header, the main content area has a white background. A section titled 'Cabeçalho' (Header) is visible, containing fields for 'Título' (Title) with the value 'Geladeira' and 'Descrição' (Description) with the value 'Geladeira em bom estado'. Below these are dropdown menus for 'Status' (Available), 'Categoria' (Category) set to 'Eletrodoméstico', and 'Operação' (Operation) set to 'Troca'. A note below the category says 'Produtos Aceitos na Troca' (Products accepted in swap) followed by the text 'Freezer Horizontal ou Frigobar.'

Fonte: Autoria própria (2022).

Nessa operação é coletada a informação de preenchimento obrigatório “Produtos Aceitos na Troca”, em que o usuário deve informar os itens que ele aceita na troca.

A Figura 29 apresenta a seção do cabeçalho quando selecionada a operação referente à doação de valor.

Figura 29 – Tela cadastro de anúncio com a operação “Doação de Valor”

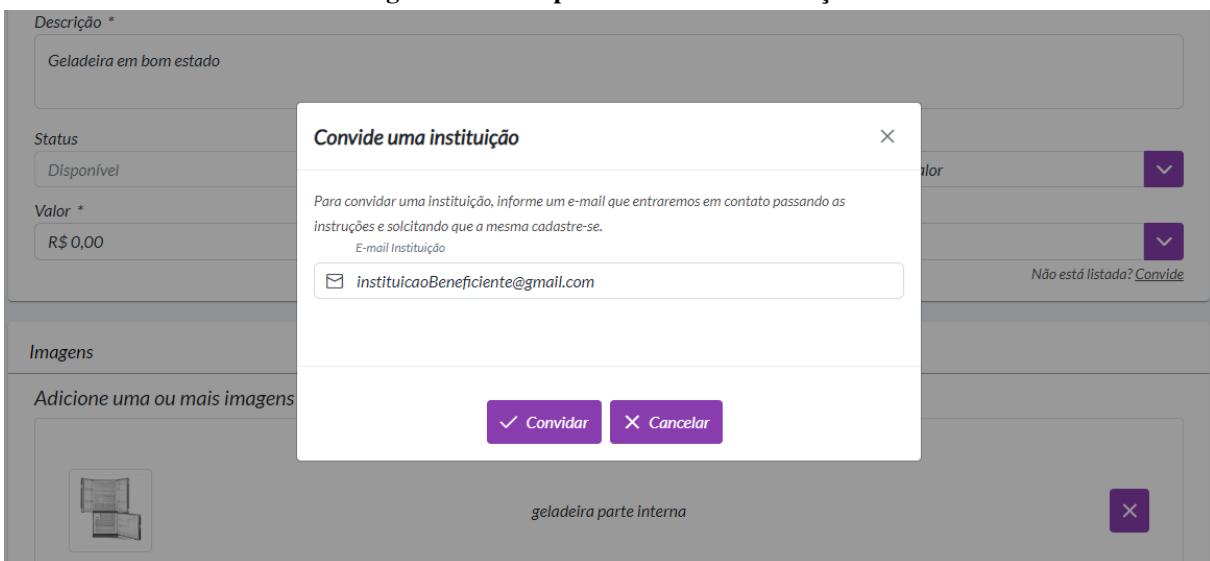
This screenshot shows the same application interface as Figure 28, but with the 'Operação' dropdown set to 'Doação de Valor'. The 'Instituição' (Institution) dropdown is also visible, showing 'Instituição do Beneficiente'. A small note at the bottom right of the form area says 'Não está listada? Convide'.

Fonte: Autoria própria (2022).

Na operação de doação são coletadas as informações de preenchimento obrigatório “Valor” e o campo “Instituição”. Em instituição são listadas todas as instituições cadastradas e com o cadastro validado. Caso a instituição ainda não esteja cadastrada no sistema o usuário pode convidá-la, clicando na opção “Convide”.

A Figura 30 apresenta a tela para convidar uma instituição.

Figura 30 – Tela para convidar a instituição

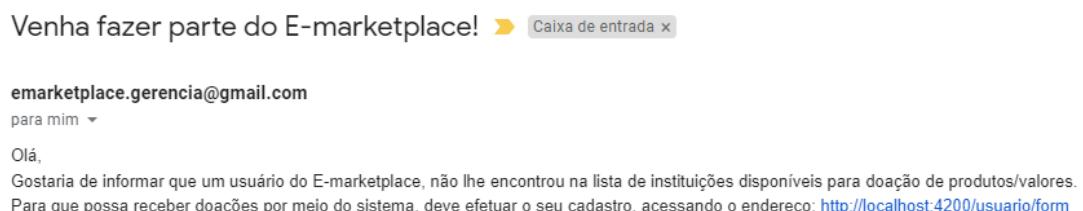


Fonte: Autoria própria (2022).

Na tela de convite de instituição, deve-se preencher o campo com o *e-mail* da instituição e clicar em “Enviar”, então é realizado o envio do convite e exibida a mensagem “E-mail enviado com sucesso!”.

A Figura 31 apresenta o *e-mail* de convite que a instituição recebe.

Figura 31 – E-mail de convite para instituição



Fonte: Autoria própria (2022).

Ao clicar no *link* enviado no *e-mail* o usuário será direcionado para a tela de cadastro. Após o cadastro ter sido validado por um usuário administrador, a referida instituição será listada e poderá ser selecionada para receber doações.

A Figura 32 apresenta a seção do cabeçalho, quando selecionada a operação referente à doação de produto.

Figura 32 – Tela cadastro de anúncio com a operação “Doação de Produto”

The screenshot shows the 'Anúncios' (Advertisements) section of the Emarketplace application. The 'Cabeçalho' (Header) form is open, containing the following fields:

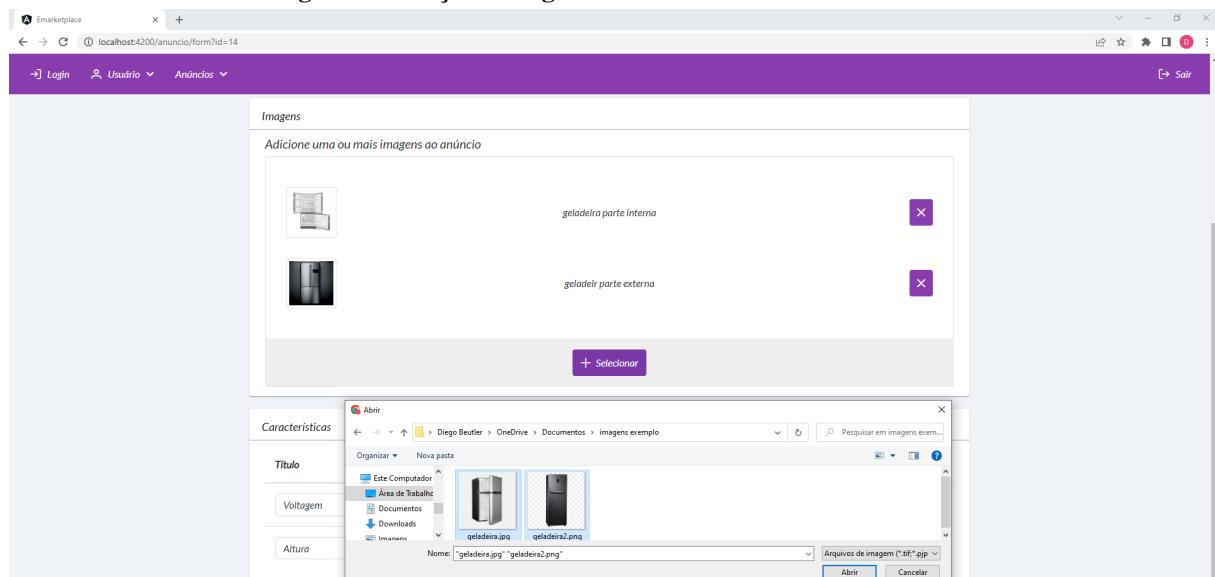
- Title ***: Geladeira
- Description ***: Geladeira em bom estado
- Status**: Disponível
- Category ***: Eletrodoméstico
- Operation ***: Doação do Produto
- Institution ***: Instituição Beneficente

Fonte: Autoria própria (2022).

Quando selecionada a operação de doação é coletada a informação referente à instituição, que é de preenchimento obrigatório, e há a possibilidade de convidar a instituição, caso a mesma não esteja listada.

A Figura 33 exibe a seção de imagens, na tela de cadastro de um anúncio.

Figura 33 – Seção “Imagens” da tela de cadastro de anúncio



Fonte: Autoria própria (2022).

Na seção de imagens, deve-se adicionar uma ou mais imagens. Para isso utiliza-se a opção “Selecionar”. Caso nenhuma imagem for adicionada, ao tentar salvar o anúncio o sistema exibe a mensagem de alerta “Deve-se selecionar ao menos uma imagem”. As imagens do anúncio ficam hospedadas em nuvem, utilizando o S3.

A Figura 34 exibe a seção de características, na tela do anúncio.

Figura 34 – Seção “Características” da tela de cadastro de anúncio

The screenshot shows the 'Anúncios' section of the Emarketplace application. At the top, there's a purple header bar with 'Login', 'Usuário', and 'Anúncios' buttons. Below the header, the main content area has two main sections:

- Imagens (Images):** A section titled 'Imagens' with the instruction 'Adicione uma ou mais imagens ao anúncio'. It contains two image thumbnails: 'geladeira parte interna' (internal refrigerator part) and 'geladeira parte externa' (external refrigerator part), each with a delete icon ('x'). Below these is a button labeled '+ Selecionar' (Select).
- Características (Characteristics):** A section titled 'Características' with a table structure. The columns are 'Título' (Title) and 'Descrição' (Description). It contains two rows:

Título	Descrição
Voltagem	110V
Altura	1.80M

 Below the table is a button labeled '+ Adicionar' (Add). At the bottom of the section are buttons for 'Salvar' (Save) and 'Excluir' (Delete).

Fonte: Autoria própria (2022).

Na tela apresentada na Figura 34, o usuário pode informar quantas características desejar de forma dinâmica, ao clicar no botão “Adicionar”. Ao clicar em um campo é possível editar as caraterísticas e, para removê-las, é necessário clicar no ícone representado por uma lixeira situada ao final da linha do campo que está sendo manipulado.

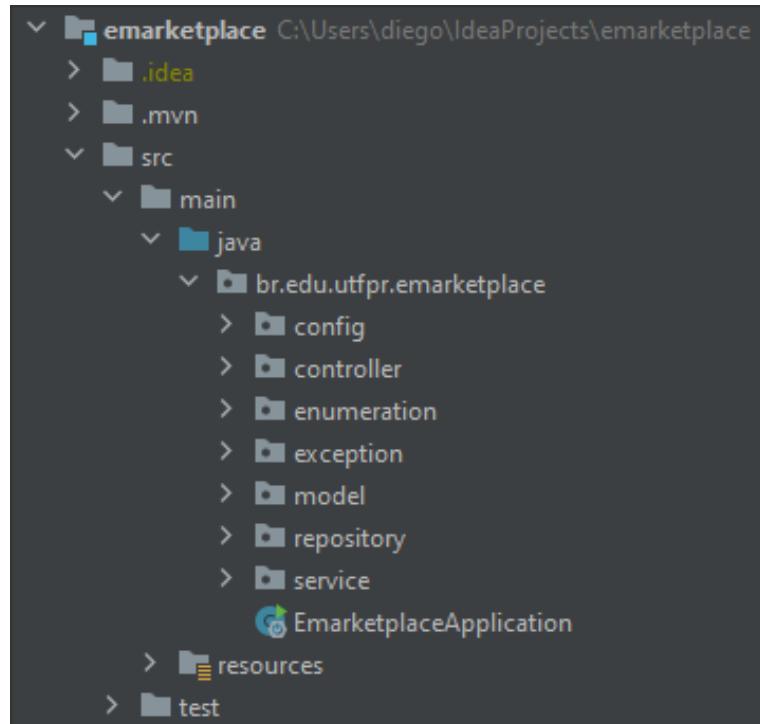
Ao final da tela são exibidas as opções para salvar e excluir o anúncio. A exclusão é apresentada apenas quando o anúncio está salvo, sendo desabilitada caso o *status* seja diferente de “Disponível”.

4.4 IMPLEMENTAÇÃO DO SISTEMA

A apresentação da implementação do sistema é realizada por meio de partes relevantes do código desenvolvido, visando exemplificar o uso das tecnologias presentes no Quadro 2 do Capítulo 3.

Inicialmente, foi construído o banco de dados, utilizando o PostgreSQL como Sistema de Gerenciamento de Banco de Dados (SGBD) e o DBeaver para a administração e desenvolvimento. Em seguida, foram inseridos dados nas tabelas categoria, permissão, país, estado e cidade. No projeto *back-end*, Java foi utilizada como linguagem de programação, juntamente com o ecossistema Spring. A Figura 35 representa a estrutura dos pacotes no projeto *back-end*.

Figura 35 – Organização dos pacotes no *back-end*



Fonte: Autoria própria (2022).

A Listagem 1 apresenta parte do código da classe SecurityConfig que está na pasta security, no diretório config.

Listagem 1 – Configuração das permissões de acesso aos *endpoints* da aplicação

```

1  @Configuration
2  @EnableWebSecurity
3  @RequiredArgsConstructor
4  public class SecurityConfig extends WebSecurityConfigurerAdapter {
5      private final UserDetailsService userDetailsService;
6      private final BCryptPasswordEncoder bCryptPasswordEncoder;
7      @Override
8      protected void configure(HttpSecurity http) throws Exception {
9          CustomAuthenticationFilter customAuthenticationFilter =
10             new CustomAuthenticationFilter(authenticationManagerBean());
11             customAuthenticationFilter.setFilterProcessesUrl("/login");
12             http.csrf().disable();
13             http.sessionManagement().sessionCreationPolicy(STATELESS);
14             http.authorizeRequests()
15                 .antMatchers(HttpMethod.POST, "/usuario/validate")
16                 .hasAnyAuthority("ROLE_ADMIN")
17                 .antMatchers(HttpMethod.GET, "/anuncio/**")
18                 .permitAll()
19                 ...
20                 .anyRequest().authenticated();
21             http.addFilter(customAuthenticationFilter);
22             http.addFilterBefore(new CustomAuthorizationFilter(),
23             UsernamePasswordAuthenticationFilter.class);
24     }
25     // demais métodos
26 }
```

Fonte: Autoria própria (2022).

Na Listagem 1, a anotação `@Configuration` indica que essa classe é fonte de definições de *beans* e com a anotação `@EnableWebSecurity` são habilitados os recursos de segurança. A anotação `@RequiredArgsConstructor` é utilizada para criar um construtor com os atributos requeridos, realizando, assim, a injecão de dependência. O método presente na Listagem 1 é utilizado para configurar quais requisições podem ser realizadas. Requisições para /usuario/validate, utilizando o verbo *Hypertext Transfer Protocol* (HTTP) post que é utilizado para salvar informações, só poderão ocorrer se o usuário estiver autenticado e possuir permissão de administrador. Já as requisições utilizando o verbo HTTP get para /anuncio/** podem ser realizadas sem autenticação na aplicação. Para acessar as demais requisições o usuário precisa estar autenticado.

No diretório controller, está a classe apresentada parcialmente na Listagem 2.

Listagem 2 – *Controller* do anúncio

```

1  @RestController
2  @RequestMapping( "anuncio" )
3  @RequiredArgsConstructor
4  public class AnuncioController {
5      private final AnuncioService anuncioService;
6
7      @PostMapping
8      private Anuncio salvar(@RequestBody Anuncio anuncio) {
9          return anuncioService.salvar(anuncio);
10     }
11
12     @DeleteMapping( "{ id }" )
13     private void excluir(@PathVariable("id") Long id) {
14         anuncioService.excluir(id);
15     }
16
17     @PostMapping( "/filter" )
18     private List<Anuncio> findAnunciosByFilter(
19         @RequestBody AnuncioFilter anuncioFilter) {
20         return anuncioService.findAnunciosByFilter(anuncioFilter);
21     }
22     // demais métodos
23 }
```

Fonte: Autoria própria (2022).

A Listagem 2, apresenta o código do *controller* para as requisições relacionadas ao anúncio. A anotação `@RestController` indica que é uma classe *Application Programming Interface (API)* *Representational State Transfer* (REST), já a anotação `@RequestMapping("anuncio")` define o caminho da requisição. As anotações `@PostMapping`, `@DeleteMapping` e `@GetMapping` estão indicando o verbo HTTP da requisição, ou seja, se enviada uma requisição post para o caminho da aplicação /anuncio o método `salvar()` será acessado. O método `salvar()`, declarado na interface `AnuncioService`, é implementado pela classe `AnuncioServiceImpl`, como exibido na Listagem 6.

No diretório service, está a interface CrudService exibida na Listagem 3.

Listagem 3 – Interface contendo a declaração dos métodos padrões para todos os *services*

```

1  public interface CrudService<T, ID extends Serializable > {
2      List<T> findAll();
3
4      List<T> findAll(Sort sort);
5
6      T findById(ID id);
7
8      <S extends T> S save(S entity) throws Exception;
9
10     <S extends T> S saveAndFlush(S entity);
11
12     <S extends T> List<S> saveAll(Iterable<S> entities);
13
14     void delete(ID id);
15
16     void delete(T entity);
17
18     List<T> findAllById(Iterable<ID> ids);
19 }
```

Fonte: Autoria própria (2022).

A Listagem 3 representa a interface que contém os métodos que são comuns para todos os *services*. Na Listagem 4, pode-se observar parte da classe abstrata que implementa essa interface.

Listagem 4 – Classe contendo a implementação dos métodos padrões para todos os services

```

1  public abstract class CrudServiceImpl<T, ID extends Serializable>
2   implements CrudService<T, ID> {
3
4     public abstract JpaRepository<T, ID> getRepository();
5
6     @Override
7     @Transactional(readOnly = true)
8     public List<T> findAll() {
9       return getRepository().findAll();
10    }
11
12    @Override
13    @Transactional(propagation = Propagation.REQUIRED,
14    rollbackFor = Throwable.class )
15    public <S extends T> S save(S entity) throws Exception {
16      valid(entity);
17      preSave(entity);
18      getRepository().save(entity);
19      postSave(entity);
20      return entity;
21    }
22
23    public void valid(T entity) throws Exception {
24    }
25
26    public void preSave(T entity) {
27    }
28
29    public void postSave(T entity) {
30    }
31
32    @Override
33    @Transactional
34    public void delete(ID id) {
35      getRepository().deleteById(id);
36    }
37
38    @Override
39    @Transactional
40    public T findById(ID id) {
41      return getRepository().findById(id).orElse(null);
42    }
43
44    // demais métodos
45  }

```

Fonte: Autoria própria (2022).

Na Listagem 4 a classe abstrata CrudServiceImpl possui a implementação padrão de alguns métodos, delegando todas as operações de banco de dados para o *repository*. O *repository* é obtido a partir do método abstrato getRepository(), que é implementado nas classes que estendem a CrudServiceImpl.

Na implementação do método save(), há a anotação @Transactional que é utilizada para realizar o controle transacional. O atributo propagation tem a finalidade de indicar que caso não exista uma transação já criada, deve-se criar uma. Se existir, deve ser utilizada a já existente e caso não informado, *Propagation.REQUIRED* é o valor padrão. O parâmetro rollbackFor = *Throwable.class* indica que caso seja lançada qualquer exceção da classe Throwable ou que a estenda, deve ser efetuado *rollback* de tudo o que estiver na referida transação.

A Listagem 5 exibe a interface AnuncioService.

Listagem 5 – Interface contendo a assinatura dos métodos necessários para o service do anúncio

```

1 public interface AnuncioService extends CrudService<Anuncio , Long> {
2
3     Anuncio salvar(Anuncio anuncio) throws Exception ;
4
5     void excluir(Long id);
6
7     List<Anuncio> findAnunciosByFilter(AnuncioFilter anuncioFilter);
8
9     void convidarInstituicao( String emailInstituicao );
10
11    Anuncio updateStatus(Anuncio anuncio);
12
13    Anuncio findAnuncioById(Long id)
14    throws RecursoNaoPermitidoException ;
15 }
```

Fonte: Autoria própria (2022).

A interface AnuncioService estende a interface CrudService presente na Listagem 3. Ela é utilizada na classe AnuncioServiceImpl, exibida parcialmente na Listagem 6.

Listagem 6 – Classe contendo a implementação dos métodos necessários para o *service* do anúncio

```

1  @RequiredArgsConstructor
2  @Service
3  public class AnuncioServiceImpl extends
4  CrudServiceImpl<Anuncio, Long> implements AnuncioService {
5
6      private final AnuncioRepository anuncioRepository;
7      private final UsuarioService usuarioService;
8      private final EnvioEmailService envioEmailService;
9      private final AmazonS3BucketService amazonS3BucketService;
10     ...
11
12     @Override
13     public JpaRepository<Anuncio, Long> getRepository() {
14         return anuncioRepository;
15     }
16
17     @Override
18     public void preSave(Anuncio anuncio) {
19         updateImagesAws(anuncio);
20     }
21
22     @Override
23     public Anuncio salvar(Anuncio anuncio) throws Exception {
24         if (isNull(anuncio.getId())) {
25             anuncio.setDataPublicacao(LocalDate.now());
26             anuncio.setStatus(Status.DISPONIVEL);
27             anuncio.setUsuarioOrigem(
28                 usuarioService.getUsuarioLogado());
29         }
30         return save(anuncio);
31     }
32
33     @Override
34     public void postSave(Anuncio anuncio) {
35         deleteOldImages();
36         saveNewImages(anuncio);
37     }
38     // demais métodos
39 }
```

Fonte: Autoria própria (2022).

A anotação @Service é uma especificação da anotação @Component, indicando que é uma classe de serviço, possibilitando, assim, entre outras coisas, a injecão de dependência em outras classes.

A classe AnuncioServiceImpl estende a classe CrudServiceImpl presente na Listagem 4 que implementa o método abstrato getRepository(), passando o *repository* correspondente. Nesse *service* são sobrescritos os métodos preSave() e postSave().

Além de estender a classe CrudServiceImpl, a AnuncioServiceImpl implementa a interface AnuncioService, presente na Listagem 3. Sendo assim, todos os métodos contidos na interface AnuncioService precisam ser implementados na classe AnuncioServiceImpl. Um exemplo é o método salvar() que atribui alguns valores se o registro ainda não foi salvo e chama o método save(), presente na Listagem 4. Esse, por sua vez, chama os métodos valid() e preSave(), salva e invoca o postSave(). Isso tudo ocorre na mesma transação, mas é possível na implementação forçar o uso de uma nova transação, caso necessário.

O processo de filtrar os anúncios na classe AnuncioServiceImpl é exibido na Listagem 7.

Listagem 7 – Implementação dos filtros no service do anúncio

```

1  @RequiredArgsConstructor
2  @Service
3  public class AnuncioServiceImpl extends
4  CrudServiceImpl<Anuncio, Long> implements AnuncioService {
5
6      private final AnuncioRepository anuncioRepository;
7      private final UsuarioService usuarioService;
8      private final EnvioEmailService envioEmailService;
9      private final AmazonS3BucketService amazonS3BucketService;
10     private final ImagemAnuncioRepository imagemAnuncioRepository;
11     ...
12
13     @Override
14     public List<Anuncio> findAnunciosByFilter(
15         AnuncioFilter anuncioFilter) {
16         anuncioFilter.setUsuarioLogado(
17             usuarioService.getUsuarioLogado());
18         var anuncios =
19             anuncioRepository.findAnunciosByFilter(anuncioFilter);
20         setUsuarioLogadoEh
21             (anuncios, anuncioFilter.getUsuarioLogado());
22         return anuncios;
23     }
24
25     private void setUsuarioLogadoEh(
26         List<Anuncio> anuncios, Usuario usuarioLogado) {
27         Optional.ofNullable(usuarioLogado)
28             .ifPresent(usuario ->
29                 anuncios.forEach(
30                     e -> setUsuarioLogadoEh(e, usuarioLogado)));
31     }
32
33     private void setUsuarioLogadoEh(Anuncio anuncio,
34                                     Usuario usuarioLogado) {
35         anuncio.setEhUsuarioOrigem(usuarioLogado
36             .equals(anuncio.getUsuarioOrigem()));
37         anuncio.setEhUsuarioDestino(usuarioLogado
38             .equals(anuncio.getUsuarioDestino()));
39         anuncio.setEhUsuarioInstituicao(usuarioLogado
40             .equals(anuncio.getUsuarioInstituicao()));
41     }
42     // demais métodos
43 }
```

Fonte: Autoria própria (2022).

O método `findAnunciosByFilter()` é chamado no *controller* sempre que recebe uma requisição de filtro. Como mostrado na Listagem 2, esse método possui um parâmetro do tipo `AnuncioFilter` que contém todas as informações presentes na tela de filtro, representada na Figura 14. Ao ser chamado esse método, é realizada a atribuição do usuário autenticado para o atributo `usuarioLogado`, presente no objeto `anuncioFilter`. Então é realizada a chamada do método `findAnunciosByFilter()` por meio da interface `AnuncioRepository`, contida na Listagem 8. Após a filtragem, é chamado o método `setUsuarioLogadoEh()` que atribui valor para os campos *transient* (não persistidos), `ehUsuarioOrigem`, `ehUsuarioDestino` e `ehUsuarioInstituicao`. Por fim, os anúncios filtrados são retornados e exibidos pelo *front-end*.

No diretório `repository` está a interface `AnuncioRepository`, exibida na Listagem 8.

Listagem 8 – Repository do Anúncio

```

1 public interface AnuncioRepository
2     extends JpaRepository<Anuncio, Long>, AnuncioRepositoryCustom {
3
4     List<Anuncio> findAllByUsuarioOrigemId(Long id);
5
6 }
```

Fonte: Autoria própria (2022).

A interface `AnuncioRepository` estende a interface `JpaRepository`, que faz parte do Spring Data JPA, essa que fornece todos os métodos padrões para operações *Create, Read, Update, Delete* (CRUD) e permite montar consultas usando *Query Methods*, como é o caso do método `findAllByUsuarioOrigemId(Long id)` em que a busca é montada em tempo de execução, tendo como base apenas a assinatura declarada na interface do repositório. Estende, também, a interface `AnuncioRepositoryCustom` apresentada na Listagem 9.

Listagem 9 – Interface com a assinatura dos métodos customizados no repository do anúncio

```

1 public interface AnuncioRepositoryCustom {
2
3     List<Anuncio> findAnunciosByFilter(AnuncioFilter anuncioFilter);
4 }
```

Fonte: Autoria própria (2022).

A interface `AnuncioRepositoryCustom` é implementada pela classe `AnuncioRepositoryCustomImpl` apresentada parcialmente na Listagem 10.

Listagem 10 – Classe com a implementação do método responsável pelo filtro no *repository* do anúncio

```

1  @Repository
2  @RequiredArgsConstructor
3  public class AnuncioRepositoryCustomImpl
4      implements AnuncioRepositoryCustom {
5
6      @PersistenceContext
7      private final EntityManager entityManager;
8
9      public List<Anuncio> findAnunciosByFilter(AnuncioFilter filter) {
10         CriteriaBuilder criteriaBuilder =
11             this.entityManager.getCriteriaBuilder();
12         CriteriaQuery<Anuncio> query =
13             criteriaBuilder.createQuery(Anuncio.class);
14         Root<Anuncio> anuncio = query.from(Anuncio.class);
15         Join<Object, Object> usuarioOrigem =
16             anuncio.join(Anuncio_.USUARIO_ORIGEM, JoinType.LEFT);
17         List<Predicate> predicates = new ArrayList<>();
18
19         if (nonNull(filter.getTitulo())) {
20             predicates.add(criteriaBuilder.like(
21                 criteriaBuilder.lower(anuncio.get("titulo")),
22                 "%" + filter.getTitulo().toLowerCase() + "%"));
23         }
24         if (nonNull(filter.getDataPublicacaoMin())) {
25             predicates.add(criteriaBuilder.greaterThanOrEqualTo(
26                 anuncio.get("dataPublicacao"),
27                 filter.getDataPublicacaoMin()));
28         }
29         if (nonNull(filter.getEstado()))
30             && isNull(filter.getCidade()) {
31             Join<Object, Object> cidadeUsuarioOrigem =
32                 usuarioOrigem.join(Usuario_.CIDADE, JoinType.LEFT);
33             predicates.add(criteriaBuilder.equal(
34                 cidadeUsuarioOrigem.get("estado"), filter.getEstado()));
35         }
36         ...
37         query.where(predicates.toArray(Predicate[]::new));
38
39         TypedQuery<Anuncio> queryResult =
40             this.entityManager.createQuery(query);
41         return queryResult.getResultList();
42     }
43 }
```

Fonte: Autoria própria (2022).

A classe AnuncioRepositoryCustomImpl possui a anotação @Repository, que é uma especificação da anotação @Component, indicando que a classe é um *repository*, possibilitando a injecão de dependência em outras classes. Para a construção dos filtros de forma dinâmica, de acordo com os valores informados pelo usuário, optou-se pela utilização da API Criteria, acessível no projeto devido a utilização do Spring Data JPA. Na Listagem 10, pode-se observar o filtro pelo campo título. Caso informado valor no campo para busca, é verificado se esse valor está contido no título do anúncio, sem fazer distinção de letras maiúsculas e minúsculas. É apresentado também o filtro pela data mínima, em que o sistema verifica se a data informada é menor ou igual a data de publicação do anúncio. Por fim, é apresentado o filtro pelo “estado”, verificando se foi informado o Estado e se não foi informado o campo “cidade”. Caso possua a busca por “cidade”, não há necessidade de realizar a busca por estado. Se os critérios de busca são atendidos, é realizado um *left join* da tabela usuário com a tabela cidade que possui o estado, filtrando os registros em que o estado é igual ao informado.

No diretório model, ficam as entidades da aplicação. Parte do código da entidade Anuncio pode ser visualizado na Listagem 11.

Listagem 11 – Entidade Anuncio

```

1  @Getter @Setter @Builder @NoArgsConstructor @AllArgsConstructor
2  @Audited
3  @Table( name = "ANUNCIO" )
4  @Entity
5  @TypeDef(
6      name = "jsonb-node",
7      typeClass = JsonNodeBinaryType.class
8  )
9  public class Anuncio {
10     @Id
11     @GeneratedValue(strategy = GenerationType.IDENTITY)
12     @Column(name = "ID_ANUNCIO")
13     private Long id;
14
15     @Column(name = "TITULO", nullable = false)
16     private String titulo;
17
18     @Type(type = "jsonb-node")
19     @Column(name = "CARACTERISTICAS", columnDefinition = "jsonb")
20     private JsonNode caracteristicas;
21
22     @Enumerated(EnumType.STRING)
23     @Column(name = "STATUS", length = 13, nullable = false)
24     private Status status;
25
26     @Column(name = "VALOR", precision = 12, scale = 2)
27     private BigDecimal valor;
28
29     @JsonFormat(pattern = "dd/MM/yyyy")
30     @Column(name = "DATA_PUBLICACAO", nullable = false)
31     private LocalDate dataPublicacao;
32
33     @JoinColumn(name = "ID_CATEGORIA")
34     @ManyToOne(optional = false)
35     private Categoria categoria;
36
37     @OneToMany(cascade = CascadeType.ALL, fetch = FetchType.LAZY)
38     private List<ImagenAnuncio> imagens;
39
40     private transient boolean ehUsuarioOrigem;
41     // demais campos
42
43 }
```

Fonte: Autoria própria (2022).

As anotações contidas na primeira linha da Listagem 11, fazem parte da biblioteca Lombok, utilizada no projeto. Essas anotações ajudam a reduzir os códigos de implementação padrão que são necessários na classe. As anotações utilizadas servem, respectivamente, para gerar os métodos *getter* e *setter*, possibilitar construir o objeto de forma simples e inicializar valores, criar um construtor vazio e um construtor com todos os atributos. A anotação @Audited faz parte da biblioteca Hibernate Envers e serve para auditar a tabela de anúncios. Dessa forma, é gerada uma tabela que gravará um registro para cada inserção ou atualização dos anúncios. Esse registro possui uma tabela com a data da operação, possibilitando gerar relatórios que indicam quando o registro foi editado e qual valor o respectivo registro possuía antes da alteração. A anotação @Table(name = "ANUNCIO") especifica a tabela que a entidade representa.

A anotação @Entity especifica que a classe representa uma entidade no banco de dados. A anotação @TypeDef(name = "jsonb-node", typeClass = JsonNodeBinaryType.class) está sendo utilizada para informar que o campo características é do tipo *JavaScript Object Notation* (JSON), com a anotação @Type(type = "jsonb-node"). A anotação @Id especifica que o campo id é o identificador da entidade. A anotação @GeneratedValue(strategy = GenerationType.IDENTITY) serve para especificar que o valor do identificador será gerado de forma serial. A anotação @Column() especifica o mapeamento entre o atributo e a coluna da tabela de banco de dados. A anotação @Enumerated(EnumType.STRING) serve para informar que o atributo da entidade representa um tipo Enum. A anotação @JoinColumn(name = "name") especifica a chave estrangeira do relacionamento. As anotações @ManyToOne e @OneToMany servem para indicar qual a cardinalidade do relacionamento no banco de dados. Por fim, há o campo *transient* ehUsuarioOrigem. O *transient* indica que o campo não deve ser persistido no banco de dados.

No diretório enumeration, ficam os *enums* da aplicação. A Listagem 12 apresenta o *enum* referente ao *status* do anúncio.

Listagem 12 – *Enum Status*

```

1  @AllArgsConstructor
2  public enum Status {
3
4      DISPONIVEL("Disponível"),
5      EM_NEGOCIAÇÃO("Em Negociação"),
6      FINALIZADO("Finalizado");
7
8      private final String value;
9
10     @JsonValue
11     public String getValue() {
12         return value;
13     }
14 }
```

Fonte: Autoria própria (2022).

A anotação @JsonValue é utilizada para indicar que o valor retornado pelo método getValue() deve ser utilizado na serialização.

No diretório exception, ficam as exceções personalizadas da aplicação. A Listagem 13 apresenta um exemplo de *exception*.

Listagem 13 – Exceção lançada ao tentar salvar um usuário com o valor do campo “usuario” já utilizado

```

1  @Getter
2  public class UsuarioJaExisteException extends Exception {
3
4      private final String message;
5
6      public UsuarioJaExisteException(String message) {
7          this.message = message;
8      }
9 }
```

Fonte: Autoria própria (2022).

A exceção exibida na Listagem 13 é lançada quando ao tentar salvar um usuário o valor informado no campo usuário já existe, então ocorre o retorno que pode ser observado na Figura 22.

A Listagem 14 apresenta parte do código utilizado para comunicação com o serviço da Amazon.

Listagem 14 – Implementação da comunicação com o serviço da AWS

```

1  @Service
2  public class AmazonS3BucketService {
3      private AmazonS3 amazonS3;
4
5      @Value("${endpointUrl}")
6      private String endpointUrl;
7      @Value("${bucketName}")
8      private String bucketName;
9      @Value("${pathUser}")
10     private String pathUser;
11     @Value("${pathAd}")
12     private String pathAd;
13     @Value("${accessKey}")
14     private String accessKey;
15     @Value("${secretKey}")
16     private String secretKey;
17
18     @PostConstruct
19     private void initializeAmazon() {
20         AWSCredentials credentials = new BasicAWSCredentials
21             (this.accessKey, this.secretKey);
22         this.amazonS3 = AmazonS3ClientBuilder.standard()
23             .withRegion(Regions.SA_EAST_1)
24                 .withCredentials(
25                     new AWSStaticCredentialsProvider(credentials))
26                     .build();
27     }
28
29     public String uploadFile(File file, boolean isImageProfile) {
30         //constrói a url do arquivo e chama o uploadFileToBucket()
31     }
32
33     private void uploadFileToBucket(String path,
34                                     String fileName, File file) {
35         amazonS3.putObject(new PutObjectRequest(path, fileName, file)
36             .withCannedAcl(CannedAccessControlList.PublicRead));
37     }
38
39     public void deleteFileFromBucket(String fileName,
40                                     boolean isImageProfile) {
41         String path = bucketName
42             + (isImageProfile ? pathUser : pathAd);
43         amazonS3
44             .deleteObject(new DeleteObjectRequest(path, fileName));
45     }
46 }
```

Fonte: Autoria própria (2022).

Uma vez importada a dependência da Amazon é possível configurar a comunicação com o S3, como exibido na Listagem 14. As variáveis com a anotação @Value() possuem o valor da propriedade com a chave correspondente no arquivo de propriedades application.properties. A anotação @PostConstruct faz com que o código do método seja executado após a construção da classe. Nesse método está sendo construído um objeto do tipo AmazonS3, informando a *region* que o servidor está hospedado e as credenciais de acesso obtidas dentro da AWS.

5 CONCLUSÃO

Este trabalho apresentou o desenvolvimento, incluindo a modelagem e a implementação de um sistema *web* para possibilitar a compra e a venda, empréstimo a troca e a doação de bens usados, como roupas, livros, móveis e eletrodomésticos. Embora o sistema não possua todas as operações de um comércio eletrônico - não envolve os aspectos financeiros e a logística de entrega, entre outras operações pertinentes à compra e venda de produtos nesse tipo de mercado - pode ser caracterizado como um *e-marketplace* porque reúne compradores e vendedores de diferentes segmentos. Sendo vendedores caracterizados pelos usuários que pretendem anunciar bens para venda, para empréstimo, troca ou doação. E compradores são os usuários que tem o objetivo de adquirir e emprestar bens anunciados, realizar trocas entre bens ou receber bens por doação.

A caracterização do sistema, considerando as restrições apresentadas, como um *e-marketplace* levou à elaboração do referencial teórico baseado em mercados eletrônicos. Foram apresentados apenas conceitos gerais desse tipo de mercado porque o sistema desenvolvido visa apenas reunir vendedores (anunciantes) e compradores e não implementa algoritmos específicos para negociação e outros aspectos conceituais ou tecnológicos que justificassem a necessidade de um referencial teórico mais aprofundado sobre *e-marketplace*.

O diferencial do sistema desenvolvido em relação ao conceito essencial de *e-marketplace* está no fato de possibilitar que bens sejam doados, ou ainda vendidos e o valor obtido com essa venda seja destinado a entidades assistenciais. Para que essas entidades possam ser indicadas para receber doações, elas devem estar previamente cadastradas no sistema e são indicadas pelo usuário no momento do cadastro do bem para doação.

Para desenvolver as telas da aplicação foi utilizado o *framework* Angular. Esse *framework* possibilita um alto reaproveitamento de código, devido a possibilidade de isolar pequenos trechos de códigos em componentes que podem ser reutilizados. Outra facilidade proporcionada é que por meio do uso das diretivas ngFor, ngIf e ngSwitch é possível no *HyperText Markup Language* (HTML) realizar iteração, permitindo construir componentes de forma dinâmica e implementar estruturas de decisão. Destaca-se, ainda, a *two-way data binding* ou ligação de dados bidirecional, que permite que os dados trafeguem de um componente para outro de forma automática e o próprio *framework* atualiza automaticamente os valores em tela. A biblioteca PrimeNg agrega diversos componentes ao *framework* Angular que são muito bem documentados, facilitando o seu uso.

O banco de dados utilizado foi o PostgreSQL. Dentre os benefícios oferecidos por esse banco de dados destaca-se o suporte ao tipo de dado JSON, o que facilitou a gravação e manipulação das características dos bens anunciados. A utilização do S3 tornou possível armazenar as imagens da aplicação em nuvem, de forma fácil e segura.

Java foi utilizada como linguagem de programação no *back-end*, juntamente com o ecosistema Spring. A utilização do Spring agilizou o desenvolvimento devido aos recursos como,

por exemplo, o gerenciamento de dependências que ocorre por meio da injeção de dependência. O Spring Boot agregado do Spring MVC facilitou a configuração do projeto. O Spring Data JPA proporcionou a implementação de grande parte do código necessário para a camada de manipulação de dados. Recursos como, por exemplo, autenticação, autorização baseada na permissão do usuário e armazenamento da senha criptografada viabilizaram o uso do Spring Security na aplicação.

Adequação da aplicação para que atenda a legislação pertinente, implementação de meios de pagamentos, controle de envio e de recebimento dos produtos, comunicação por meio da aplicação, impressão de relatórios e o cadastro das categorias, possibilitando criar hierarquias entre as categorias, como modelado no banco de dados, são indicações de trabalhos futuros para complementar o que foi desenvolvido e apresentado neste texto.

REFERÊNCIAS

- ALSTYNE, Marshall Van W.; PARKER, Geoffrey G.; CHOUDARY, Sangeet Paul. Pipelines, plataformas e novas regras de estratégia. **Harvard Business Review**, v. 94, n. 4, p. 32–40, 2021. Citado na página 17.
- BAKOS, Yannis. The emerging role of electronic marketplaces on the internet. **Communications of the ACM**, v. 8, n. 1, p. 35–42, 1998. Citado na página 16.
- CAMPOS, Rafael *et al.* **Marketplace, a salvação da lucratividade do e-commerce**. São Paulo, 2015. Citado na página 15.
- COPPEL, Jonathan. E-commerce: impacts and policy challenge. **Working paper no. 252**, 2000. Citado na página 16.
- EBIT. **Websshoppers - 42 ed.** [S.l.], 2020. Disponível em: <https://company.ebit.com.br/websshoppers/websshoppersfree>. Citado na página 12.
- ENG, Teck-Yong. The role of e-marketplaces in supply chain management. **Industrial Marketing Management**, v. 33, n. 2, p. 97–105, 2004. Citado na página 16.
- GUO, Jingzhi. A term in search of the infrastructure of electronic markets. **Research and Practical Issue of Enterprise Information System II**, p. 831–840, 2007. Citado na página 15.
- GUO, Jingzhi *et al.* Semantic inference on heterogeneous e-marketplace activities. **IEEE Transactions on Systems, man, and Cybernetics - Part A: Systems and Humans**, v. 42, n. 2, p. 316–330, 2012. Citado 2 vezes nas páginas 15 e 16.
- LAU, Raymond Y. K. *et al.* Knowledge discovery for adaptive negociation agents in e-marketplaces. **Decison Support System**, v. 2, n. 42, p. 310–323, 2008. Citado na página 16.
- LOUKIS, Euripidis; SPINELLIS, Diomidis; KATSIGIANNIS, Anastasios. Barriers to the adoption of b2b e-marketplaces by large enterprises: lessons learned from the hellenic aerospace industry. **Information Systems Management**, v. 28, n. 2, p. 130–146, 2011. Citado na página 15.
- MALONE, Thomas W.; YATES, JoAnne; BENJAMIN, Robert. Electronic markets and electronic hierarchies: effects of information technology on markets structure and corporate strategies. **Communication of ACM**, v. 30, n. 6, p. 484–497, ago. 1987. Citado na página 15.
- MOVAHEDI, Bahar; LAVASSANI, Kayvan; KUMAR, Vinod. E-marketplace emergence: evolution, developments and classification. **Journal of Electronic Commerce in Organizations**, v. 10, n. 1, p. 14–32, 2012. Citado 2 vezes nas páginas 15 e 16.
- OH, Jung Soo; KIM, Soo Wook. The effect of b2b e-marketplace type on buyer-supplier relational advantages of e-marketplace and firm performance. **The Asian Journal on Quality**, v. 12, n. 2, p. 189–203, 2011. Citado na página 17.

PENA, Rodolfo F. Alves. O capitalismo e a sociedade de consumo. 2021. Disponível em: <https://mundoeducacao.uol.com.br/geografia/o-capitalismo-sociedade-consumo.htm>. Citado na página 13.

RAGONE, Azzurra *et al.* Fuzzy match making in e-marketplaces for peer entities using data. **Fuzzy sets and system**, v. 160, n. 2, p. 251–268, 2009. Citado na página 16.

ROSA, João Roberto Conceição. **Marketplace no Brasil: desafios, vantagens e tendências desde modelo de negócio para empresas varegista**. 2019. Dissertação (Mestrado) — Programa de Pós-Graduação Stricto Sensu em Administração de Empresas da Faculdade FIA de Administração de Negócios, São Paulo, 2019. Citado na página 12.

SEBRAE, Serviço Brasileiro de Apoio às Micro e Pequenas Empresas. Conheça as vantagens de vender seus produtos em um e-marketplace. 2014. Disponível em: <https://www.sebrae.com.br/sites/PortalSebrae/artigos/conheca-as-vantagens-do-e-marketplace-para-os-pequenos-negocios,3f6402b5b0d36410VgnVCM1000003b74010aRCRD>. Citado na página 15.

_____. _____. p. s.p., 2021. Disponível em: <https://www.sebrae.com.br/sites/PortalSebrae/artigos/conheca-as-vantagens-do-e-marketplace-para-os-pequenos-negocios,3f6402b5b0d36410VgnVCM1000003b74010aRCRD>. Citado na página 12.

TAN, Wenan *et al.* A methodology toward manufacturing grid-based virtual enterprise operation platform. **Enterprise Inf. Syst.**, v. 4, n. 3, p. 283–309, ago. 2010. Citado na página 15.

ZACHO, Ricardo. O que é marketplace. 2017. Disponível em: <https://www.ecommercebrasil.com.br/artigos/marketplace-vantagens-e-desvantagens/>. Citado na página 15.

ZHAO, Xiaohui *et al.* Aligning collaborative business processes: an organization-oriented perspective. **IEEE Trans. Syst., Man, Cybern. A, Syst., Humans**, v. 39, n. 6, p. 1152–1164, 2009. Citado na página 15.