

**LIGA DE ENSINO DO RIO GRANDE DO NORTE
FACULDADE NATALENSE PARA O DESENVOLVIMENTO
DO RIO GRANDE DO NORTE – FARN
CURSO DE ESPECIALIZAÇÃO EM DESENVOLVIMENTO DE SISTEMAS
CORPORATIVOS**

VÁLBER CÉSAR CAVALCANTI ROZA

SISTEMA WEB DE COMPRAS E VENDAS DE PRODUTOS

NATAL / R N

2011

VÁLBER CÉSAR CAVALCANTI ROZA

SISTEMA WEB DE COMPRAS E VENDAS DE PRODUTOS

Monografia apresentada como parte dos requisitos para a obtenção do título de Especialista em Desenvolvimento de Sistemas Corporativos da Faculdade Natalense para o Desenvolvimento do Rio Grande do Norte – FARN.

**Orientadora: Ricardo Wendell Rodrigues da
Silveira**

NATAL / RN

2011

Catalogação na Publicação – Biblioteca da FARN
Setor de Processos Técnicos

Roza, Válber César Cavalcanti
Sistema Web de Compras e Vendas de Produtos / Válber César Cavalcanti
Roza. – Natal, 2011.
50 f.

Orientador: Ricardo Wendell Rodrigues da Silveira

Monografia (Especialização em Desenvolvimento de Sistemas Corporativos) –
Faculdade Natalense para o Desenvolvimento do Rio Grande do Norte.

1. Sistemas de Informação - Monografia. 2. Comércio Eletrônico
(E-commerce) – Monografia. 3. Sistema Web de Compras e Venda - Monografia.

I. Título: *Sistema Web de Compras e Vendas de Produtos*

RN/FARN/BC

CDU 004.7

VÁLBER CÉSAR CAVALCANTI ROZA

SISTEMA WEB DE COMPRAS E VENDAS DE PRODUTOS

Monografia apresentada como requisito necessário para a obtenção do título de Especialista em Desenvolvimento de Sistemas Corporativos da Faculdade Natalense para o Desenvolvimento do Rio Grande do Norte – FARN.

Monografia aprovada em ____ / ____ / 2011.

Válber César Cavalcanti Roza

Prof.^o Ricardo Wendell Rodrigues da Silveira
Orientador

Prof.^o David Ricardo Pereira
Membro

Prof.^o Gleydson Lima
Membro

AGRADECIMENTOS

À minha esposa, Kaire Raianae Silva Bezerra Cavalcanti, pelo apoio e paciência comigo durante essa empreitada;

Aos meus colegas e memoráveis amigos em especial, Rodrigo Oliveira de Jesus, Voltaire Maia de Holanda e Diogo Maed Dantas Maia, pelas noites a fio em prol dos estudos associados a este Curso de Especialização.

Registro aqui também meus mais expressivos agradecimentos ao meu orientador Ricardo Wendell Rodrigues da Silveira, pela orientação, pela confiança e por me mostrar o direcionamento a ser seguido no desenvolvimento deste trabalho.

Àqueles que por ventura não tenham sido citados, quero mencionar os meus sinceros agradecimentos porque de alguma forma passaram pela minha vida e contribuíram para a construção de quem sou hoje.

RESUMO

Este trabalho apresenta o desenvolvimento de um sistema de comércio eletrônico, surgido com o propósito de ser aplicado em grupos pequenos de pessoas, corporações, instituições, condomínios, cidades ou Estados; sejam esses grupos de pequeno ou médio porte, diferenciando de outros sistemas do ramo, comumente conhecidos hoje em dia, onde os produtos apresentados, são provenientes de toda parte do Brasil ou até mesmo do mundo. Desta forma, restringindo o escopo de atuação desse sistema, os usuários que desejarem divulgar ou comprar algum produto ou até mesmo serviço, poderão usufruir dele para tal. Ambientes de trabalho (instituições públicas, privadas ou condomínios), cidades, ou Estados, poderão usar esse sistema para divulgar produtos ou serviços locais, facilitando assim as negociações das compras e diminuindo os riscos de fraudes por se tratar de anúncios locais. Neste relatório também foram abordados outros temas como, os *frameworks*, *softwares* e ferramentas usadas no desenvolvimento do sistema em questão, com o intuito de melhor apresentar o sistema proposto.

Palavras-Chave: Comércio Eletrônico (*E-Commerce*), Sistema *Web* de Compras e Vendas, Aplicação Java.

ABSTRACT

This paper presents the development of an electronic trading system that emerged with the purpose of being applied to small groups of people, corporations, institutions, condominiums, cities or states, these groups are small or medium size, distinguishing it from other systems in the industry , commonly known today, where the products featured are from all parts of Brazil or even the world. Thus, restricting the scope of performance of this system, users who wish to disclose or buy some product or even service, can enjoy it for that. Workplaces (public, private or condominiums), cities, or states, may use this system to promote local products or services, thus facilitating the negotiations of purchases and reducing the risks of fraud because it is local advertising. In this report were also discussed other issues such as frameworks, and software tools used in developing the system in question, in order to better present the proposed system.

Keywords: *Electronic Commerce (E-Commerce), Web System Shopping and Sales, Java application.*

LISTA DE FIGURAS

FIGURA 1 - Gráfico demonstrativo do faturamento anual do <i>e-commerce</i> no Brasil até 2010 (EBIT, 2011).....	15
FIGURA 2 - Esquema geral dos módulos do framework JSF.....	18
FIGURA 3 - O papel do <i>Hibernate</i> em uma aplicação Java (LINWOOD, MINTER, 2010)....	18
FIGURA 4 - Classe mínima mapeada com anotações.....	19
FIGURA 5 - Módulos do <i>Primefaces</i>	20
FIGURA 6 - Módulos do <i>Spring Framework</i> v3.0.....	22
FIGURA 7 - Site oficial do <i>Eclipse</i> IDE para baixar o ambiente de desenvolvimento.....	24
FIGURA 8 - Tela com o <i>link</i> para baixar o <i>Eclipse Indigo</i>	25
FIGURA 9 - Tela com o <i>link</i> para baixar o servidor <i>Apache Tomcat</i>	25
FIGURA 10 - Tela inicial e de <i>download</i> do <i>PostgreSQL</i>	27
FIGURA 11 - Diagrama de classes referente às classes persistidas no banco de dados.....	28
FIGURA 12 - Mapeamento do relacionamento entre as entidades Categoria e Produto.....	29
FIGURA 13 - Diagrama de atividades relacionado à solicitação de produtos.....	32
FIGURA 14 - Caso de uso referente aos usuários comprador e vendedor.....	37
FIGURA 15 - Caso de uso referente ao usuário administrador.....	39
FIGURA 16 - Diagrama de sequência referente à solicitação de compra de produto.....	40
FIGURA 17 – Diagrama de sequência referente ao cadastro de produtos.....	41
FIGURA 18 - Tela de sugestão de novo preço de produto e o <i>e-mail</i> enviado ao vendedor, comunicando a sugestão do usuário.....	42
FIGURA 19 - Tela de comentário do sistema, no espaço de anúncio do produto.....	43
FIGURA 20 - Tela de qualificação de vendedor e representação gráfica.....	43
FIGURA 21 - Tela inicial do sistema com permissões de usuário administrador e usuário comprador.....	44
FIGURA 22 - Esquema do funcionamento da ação de solicitação e compra de produto.....	45
FIGURA 23 - Tela que antecede a solicitação de compra, específica para cada produto.....	46
FIGURA 24 - Tela inicial completa do sistema com permissões de usuário administrador....	47
FIGURA 25 - Protótipos da segunda versão (expansão) do sistema apresentado neste trabalho.....	48

SUMÁRIO

1	INTRODUÇÃO.....	11
1.1	MOTIVAÇÃO.....	12
1.2	ORGANIZAÇÃO DA MONOGRAFIA.....	16
2	REFERENCIAL TEÓRICO.....	17
3	IMPLEMENTAÇÃO DO SISTEMA.....	24
3.1	AMBIENTE DE DESENVOLVIMENTO ECLIPSE INDIGO.....	24
3.1.1	Instalando o Eclipse Indigo.....	24
3.2	SERVIDOR WEB APACHE TOMCAT V7.0.....	25
3.2.1	Instalando o Apache Tomcat.....	25
3.3	O SISTEMA GERENCIADOR DE BANCO DE DADOS	26
3.3.1	Instalando o PostgreSQL.....	26
3.4	DIAGRAMA DE CLASSES.....	27
3.5	DIAGRAMA DE CASOS DE USO.....	30
3.6	CASOS DE USO: ATOR COMPRADOR.....	30
3.6.1	Cadastrar Usuário.....	30
3.6.2	Solicitar Produto.....	31
3.6.3	Verificar Solicitação.....	33
3.6.4	Cancelar Solicitação.....	33
3.6.5	Qualificar Vendedor.....	34
3.7	CASOS DE USO: ATOR VENDEDOR.....	35
3.7.1	Cadastrar Produto.....	35
3.7.2	Procurar Solicitação.....	35
3.7.3	Cadastrar Promoção.....	36
3.7.4	Finalizar Venda.....	36
3.8	CASOS DE USO: ATOR ADMINISTRADOR.....	38
3.8.1	Efetuar Ações Gerais.....	38
3.9	DIAGRAMA DE SEQUÊNCIA.....	39
3.10	OUTRAS FUNCIONALIDADES DO SISTEMA.....	42
3.10.1	Sugestão de Preço.....	42
3.10.2	Comentários de Anúncios.....	43

4	RESULTADOS.....	44
5	CONCLUSÃO.....	48
5.1	PERSPECTIVAS FUTURAS.....	48
	REFERÊNCIAS.....	49

1 INTRODUÇÃO

Desenvolver um sistema que satisfaça seus usuários é o desejo de todos os desenvolvedores. Para isso, ele deve suprir as necessidades atuais dos usuários e também as necessidades que surgirão, a fim de gerar inovação e qualidade no sistema, tanto nas tecnologias usadas em seu desenvolvimento, quanto no atendimento e suporte aos seus usuários.

O sistema proposto neste trabalho tem o objetivo de suprir certas necessidades de grupos de pessoas, através de um sistema *web* de compras e vendas de produtos provenientes de seus usuários cadastrados, semelhante ao funcionamento de um mural de divulgações. O termo, grupos de pessoas, anteriormente citado, refere-se ao escopo de ação desse sistema, o qual é direcionado para ser usado em instituições públicas e privadas, sejam elas escolas, universidades, grupos de trabalho, empresas, em condomínios ou em localidades como, por exemplo, cidades e Estados.

A fim de suprir o propósito do trabalho de conclusão do curso de especialização em questão, restringimos o escopo do sistema para apenas Universidades, ou seja, o sistema terá como seus usuários compradores e vendedores, os estudantes e servidores dela.

Em outras palavras o sistema em questão está definido como um comércio eletrônico, ou como é mais conhecido, um *e-commerce*. Tal termo refere-se ao processo de executar várias atividades relacionadas ao comércio, por meios eletrônicos, principalmente através da internet, ou se preferirmos outra definição, podemos dizer que *e-commerce* refere-se ao ato de realizar transações de compras, vendas e transferências de fundos eletronicamente, principalmente através da internet. Dentre essas atividades relacionadas ao *e-commerce*, seguem alguns tipos (SILBERSCHATZ, SUDARSHAN, 2006; MENDES, 2008):

- **Atividades de pré-vendas:** Necessárias para informar ao comprador em potencial sobre o produto ou serviço sendo vendido;
- **O processo de venda:** Que inclui negociações sobre preço e qualidade do serviço, além de outras questões contratuais;
- **O mercado:** Quando existem vários vendedores e compradores para um produto, um mercado, como o de ações, ajuda a negociar o preço a ser pago pelo produto, como vemos com os leilões. Os leilões são usados quando existem um único vendedor e vários

compradores; há também os leilões reversos, que são usados quando existe um único comprador e vários vendedores;

- **Pagamento pela venda;**
- **Atividades Relacionadas à remessa do produto ou serviço:** Alguns produtos e serviços podem ser remetidos pela *internet*, para outros, a *internet* é usada apenas para oferecer informações de remessa e rastrear remessa de produtos;
- **Suporte ao cliente e serviços de pós-venda.**

Sobre a característica do sistema, temos que ela se baseia em ter um escopo de ação mais restrito, acredita-se que esta restrição dos usuários dos cadastrados e das divulgações dos produtos, trará maior probabilidade dos vendedores encontrarem compradores em potencial, pois as divulgações virão de dentro do ambiente de estudo, de trabalho ou de moradia, facilitando a negociação entre os vendedores e os compradores, pois as fontes dos produtos estarão mais próximas dos interessados, com isso é provável que diminua a insegurança ou desconfiança dos usuários no ato da consolidação da compra através sistema, pois sabe-se que esse tipo de comportamento, por partes dos usuários, é comum em sistemas de comércio eletrônico.

1.1 MOTIVAÇÃO

Os consumidores *online*, como são chamadas as pessoas que usam a rede mundial de computadores para fazer compras de qualquer tipo, estão inundando a *internet*. Estão apresentando expectativas e um grau de controle muito elevadas de suas compras, que normalmente não apresentam no modelo de compra tradicional (sem uso da *internet*); para se ter uma ideia, esses tipos de consumidores se estendem também às empresas de diversos portes e ramos, e tendem a surgir cada vez mais como forma delas acompanharem a evolução da *internet* (LOSHIN, VACA, 2009).

Considerando o nicho mercadológico que a *internet* se tornou nos dias de hoje, empresas de todos os tipos e ramos possíveis, tem usado a *internet* para se “mostrarem” ao seus clientes; expondo suas benesses, produtos e serviços para o mundo com o intuito, muitas vezes pontual, de aumentar seus lucros, seja através de vendas diretas, propagandas, aluguéis ou outros serviços. A empresa usufruidora da *internet* terá maior capacidade de aquisição de

renda, uma vez que tudo que ela gerar poderá ser posto em circulação e conhecimento de seus potenciais clientes; por isso cada vez ela tem sido incluída nos negócios. Vale salientar que graças a *internet*, diversas profissões surgiram sem a necessidade de burocracia para lucrar; ser autônomo, lucrando com o que gosta através da *internet*, tem sido o desejo de muitas pessoas pelo mundo a fora.

Para uma melhor compreensão do nicho de oportunidades de negócios que a *internet* representa nos dias de hoje, citaremos algumas dessas (FILIPINI, 2008):

- **Analista de mídia online (*Media analyst*):** Analisa a organização na *internet* e outros canais e propõe ações de divulgação tendo em vista os objetivos a serem atingidos. A atividade mais comum é a avaliação e escolha das melhores alternativas para realização de campanhas, bem como acompanhamento da presença na *internet*, mídia impressa, rádio e TV;
- **Analista de anúncios em links patrocinados (*Pay per click analyst*)¹:** Trabalha principalmente com links patrocinados da *Google Inc.*, *Yahoo* e outros sistemas de busca. Escolhe as palavras-chave mais adequadas, compõe o texto dos anúncios, acompanha o desempenho dos anúncios e define estratégias. Este profissional é normalmente encontrado em agência de *marketing* digital, como responsável pelas campanhas de pagamentos por cliques dos clientes;
- **Blogger profissional (*Professional blogger*):** Administra a publicação de conteúdo e a manutenção de um *blog*. Pode trabalhar em *blogs* corporativos ou para si próprio, obtendo renda de publicidade proveniente de seu *blog*. Também efetua pesquisas e publica conteúdos com regularidade;
- **Gerente de sistema de comércio eletrônico (*E-commerce manager*):** Gerencia o canal de comercialização na *internet*; também planeja e executa ações nela, acompanhando o desempenho do comércio virtual, o desempenho das vendas e as ações de *marketing* digital. É um cargo de responsabilidade que cada vez mais é valorizado em virtude da demanda crescente das vendas *online* para as empresas.

Podemos verificar uma série de exemplos de sistemas de *e-commerce* no Brasil e no mundo. Um dos precursores e considerado um dos maiores e mais populares sistemas do tipo, é o sistema *eBay Inc.* (GOPALKRISHMNAM, GUPTA, 2007), no qual permite que seus

¹ *Links Patrocinados* é forma mais ágil de se anunciar e gerar visitantes no comércio eletrônico. Esses *links* são anúncios, geralmente de tamanho pequeno, postados em *sites* de busca da *Google Inc.* com seu sistema de publicidade *Adwords*, ou em sua rede de parceiros *Adsense*.

usuários comprem e vendam produtos e serviços variados, sendo todas essas transações de forma facilitada. Dentre os sistemas nacionais mais conhecidos e semelhantes ao apresentado neste trabalho, podemos citar: PermulaLivre, MercadoLivre e o CompraFácil. E assim como o *eBay Inc.* eles permitem que seus usuários vendam e comprem produtos e até serviços de forma facilitada.

Todos os sistemas acima citados, são exemplos de *e-commerce* o qual se define como realizador de transações de compras, vendas e transferências de fundos eletronicamente, principalmente através da *internet*, e ele é parte integrante do *e-business*², estabelecendo a conexão eletrônica entre a empresa e o cliente, seguindo a estratégia estabelecida pelo *e-business*, da qual podemos extrair três tipo de definição: *business-to-consumer*, *business-to-business* e *consumer-to-consumer* (ROBSON; MENDES, 2008). A primeira definição trata de processos de compra e venda através da *internet*, lidando diretamente com o cliente ou usuário interessado, como vemos em sistemas mundialmente conhecidos como *Amazon* e o *eBay*.

No Brasil, o comércio eletrônico tem apresentado crescente tendência nos últimos anos. De acordo com a pesquisa do site <www.ecommerce.org>, de 2007, sua colocação no mundo era 14º em número de internautas, o que representava no período, cerca de 14% da população brasileira (MENDES, 2008). Assim como no Brasil, o comércio eletrônico pelo mundo também tem tido um forte crescimento nos países em desenvolvimento como, China e Índia, principalmente pelo tamanho de suas populações, mesmo havendo constante policiamento de seus governos por questões culturais. Da mesma forma, toda a América Latina também tem se inserido cada vez mais nessa realidade tecnológica. Analisando os indicadores de *e-commerce* no mundo, constatou-se que os métodos de pagamentos através do meio digital também tem aumentado gradativamente e isso se deve aos investimentos massivos dos sistemas, em segurança das suas transações de compra venda (ALMEIDA, AVILA, 2007).

É inevitável não se deparar com essas tecnologias nos dias de hoje, pois cada vez mais a *internet* tem disponibilizado formas de pagamento *online* e assegurando aos seus clientes, a segurança e confiabilidade nessas ações.

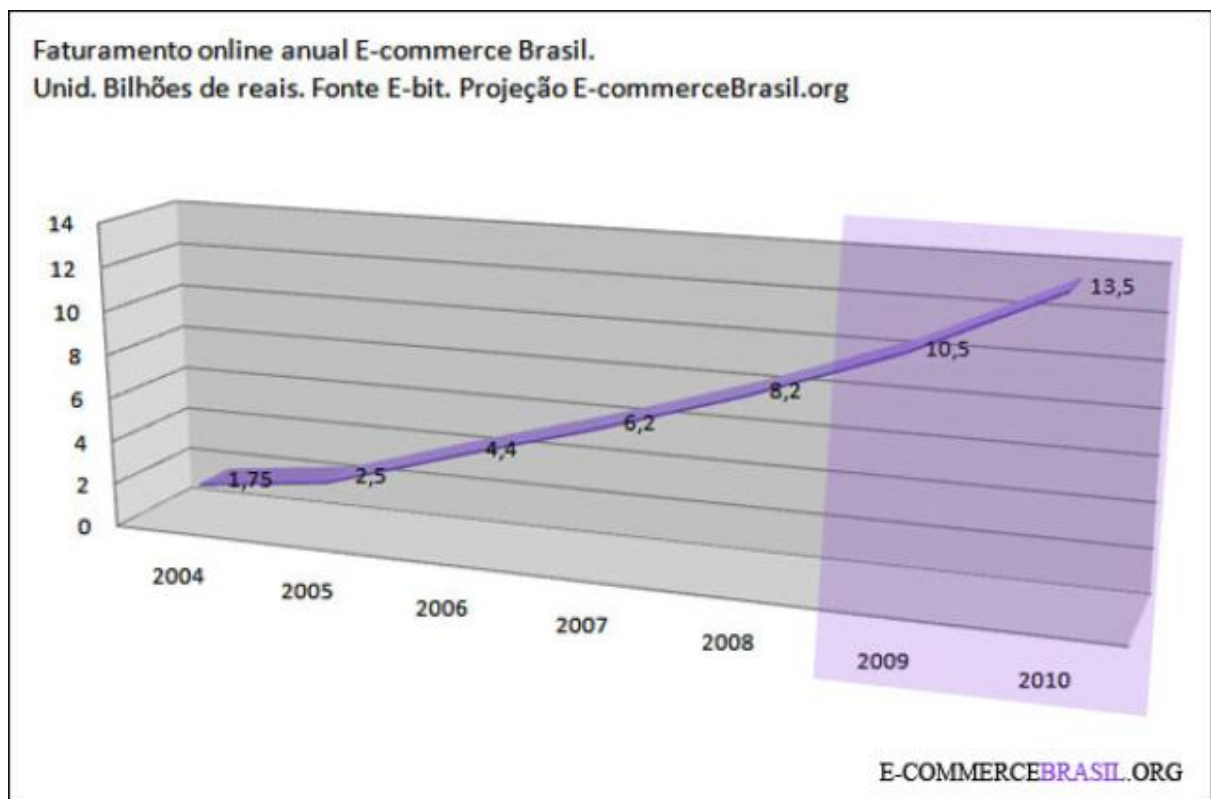
Em países em desenvolvimento, onde seus internautas tem mostrado que são mais cautelosos em usar esses tipos de sistemas, alguns pré-requisitos devem ser levados em

² E-business nada mais é do que qualquer tipo de negócio eletrônico onde o meio de atuação pode ser a *internet* ou qualquer outro meio digital que supra as necessidades de sua definição.

consideração para que o *e-commerce* alcance o sucesso desejado. Dentre eles podemos citar o principal que é a confiabilidade nos sistemas de pagamento e de disponibilização de serviços e produtos. Sem essa pré-condição é praticamente impossível que o *e-commerce* prospere nesses locais; é fácil perceber que a confiança pode ser significativa influência para evitar o bloqueio da cultura de uma certa sociedade no uso dessa tecnologia (ALMEIDA, 2007; EFEDIOGLU, 2003), como pode ser visto em países como China e Índia.

Segundo as pesquisas apresentadas pelo site <www.e-commercebrasil.org>, o faturamento do *e-commerce* no Brasil, referente ao ano 2009 foi cerca de R\$ 10,5 bilhões; e segundo a *E-bit*, empresa de pesquisa de mercado especializada, em 2010, esse faturamento alcançou a impressionante marca de R\$ 14,8 bilhões, onde as categorias que mais se destacaram foram, eletrodomésticos (14%), livros, assinaturas de revistas e jornais (12%), saúde, beleza e medicamentos (12%), informática (11%), e eletrônicos (7%) (EBIT, 2011).

Figura 1: Gráfico demonstrativo do faturamento anual do *e-commerce* no Brasil até 2010 (EBIT, 2011).



1.2 ORGANIZAÇÃO DA MONOGRAFIA

No capítulo 2 são apresentadas as tecnologias utilizadas no desenvolvimento do sistema.

No capítulo 3 são apresentadas as informações condizentes com a implementação do sistema, assim como seus diagramas de casos de uso, diagramas de classes, diagramas de sequência, diagramas de atividades e suas entidades persistidas em banco de dados.

No capítulo 4 são apresentados os resultados do sistema através das telas do sistema criado (*printscreens*), a fim de facilitar o entendimento de suas funcionalidades como: cadastros, atualizações e remoções de produtos.

No capítulo 5 são expostas as considerações finais e as perspectivas futuras do sistema.

2 REFERENCIAL TEÓRICO

Para a implementação do sistema em questão, algumas tecnologias como: pacotes de códigos específicos³ (*frameworks* e *interfaces*), softwares auxiliares para o desenvolvimento da aplicação, para modelagem e persistência, foram utilizados.

Abaixo temos a descrição de cada tecnologia usada.

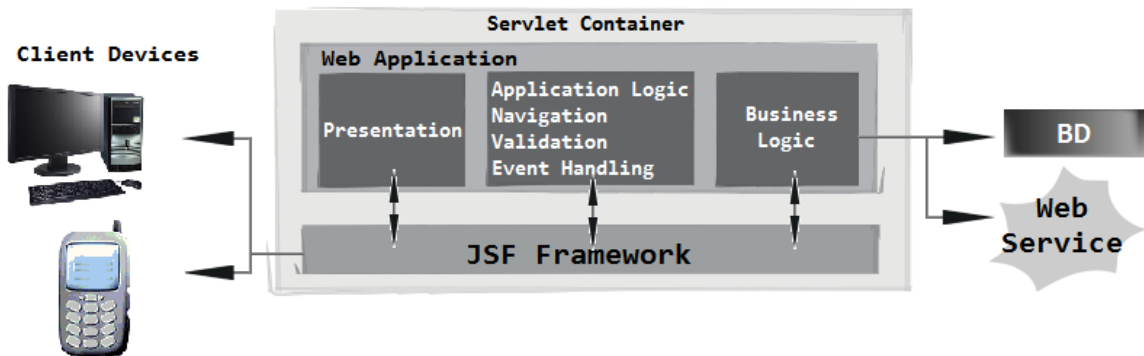
JavaServer Faces v2.0

Framework baseado em componentes, o qual possui: conjunto de *interfaces* pré-fabricadas com o usuário, possui ainda um modelo de programação de eventos e um modelo de componentes, que possibilita o desenvolvimento adicional de outros componentes (GEARY, HORSTMANN, 2010).

Como podemos ver na Figura 2, o *framework JavaServer Faces* (JSF) é responsável por interagir com os dispositivos do cliente (*client devices*), e providenciar ferramentas para integrar com a apresentação visual (*presentation*), com a lógica de aplicação (*application logic*) e com a lógica de negócio (*business logic*) de uma aplicação *web*. Todavia, o escopo do JSF é restrito na camada de apresentação. Persistência em banco de dados, *web services* e outras conexões periféricas, não fazem parte do escopo desse *framework*.

³ Esses pacotes com códigos, utilizados para o desenvolvimento de softwares são comumente conhecido como Application Programming Interfaces (API), e são fundamentais para diminuir o tempo de desenvolvimento do software, pois já inclui diversas funcionalidades já programadas evitando que elas sejam refeitas, quando necessárias.

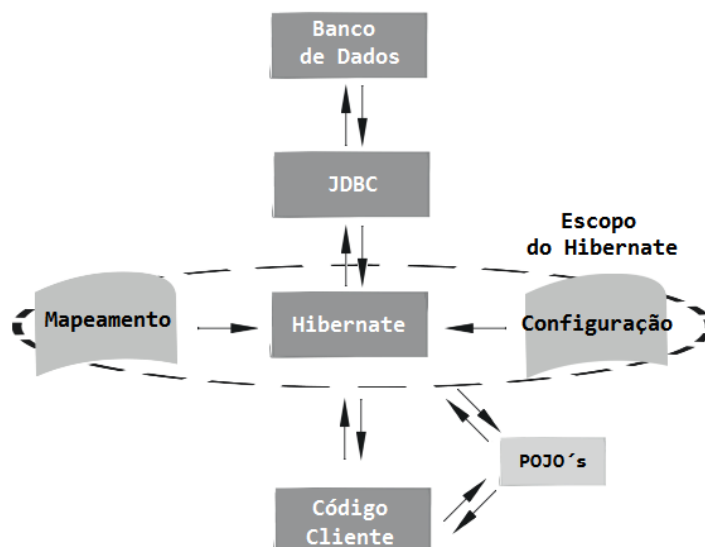
Figura 2: Esquema geral dos módulos do *framework* JSF.



Hibernate v3.5

Potente *framework* para mapeamento objeto-relacional escrito na linguagem Java e que através do uso das anotações (*annotations*)⁴, permite ao desenvolvedor criar com facilidade, complexas e funcionais estruturas de tabelas de bancos de dados (LINWOOD, MINTER, 2010).

Figura 3: O papel do *Hibernate* em uma aplicação Java (LINWOOD, MINTER, 2010).



⁴ Entre suas diversas finalidades, no sistema apresentado neste trabalho, foram principalmente usadas para mapear (transformar) classes em Java, em tabelas (entidades) de banco de dados.

Na Figura 3 vemos que o *Hibernate* atua entre a camada do cliente e a camada de persistência (relativa ao banco de dados), abstraindo ao desenvolvedor, certas complexidades no que se refere à persistência dos dados usados pelo sistema.

A saber que há duas formas de efetuar esse mapeamento usando *Hibernate*; uma delas é através de configurações de arquivos do tipo *Extensible Markup Language* (extensão .xml), que é uma maneira menos intuitiva e mais complicada de trabalhar; a outra forma é feita através de anotações, tornando o processo de configuração bem menos complicado e bem mais intuitivo, como veremos no exemplo da Figura 4. Considerando o desuso cada vez maior desse mapeamento através de arquivos .xml, é que focaremos no mapeamento através de anotações.

Figura 4: Classe mínima mapeada com anotações.

```
import javax.persistence.Entity;
import javax.persistence.Id;

// classe POJO

@Entity
public class Exemplo {
    @Id
    private int id;
    private String nome;
    ..
    // métodos getters e setters

    public int getId(){ return id; }
    public void setId(int id){ this.id = id; }
    public String getNome() { return nome; }
    public void setNome(String nome){ this.nome = nome; }
    ..
}
```

As anotações facilitam o desenvolvimento das aplicações que necessitam de persistência em banco de dados, porque são mais intuitivas para esse propósito (considerando o uso das anotações no processo de persistência). Na Figura 4, temos a anotação `@Entity`, que determina que a classe de nome **Exemplo**, será mapeada no banco de dados, como uma tabela que receberá o mesmo nome da classe; a anotação `@Id`, mapeia o atributo **id** do tipo inteiro, para ser o identificador único de cada objeto (tupla da entidade) dessa classe mapeada; o atributo **nome** do tipo *String*, será mapeado também como uma coluna dessa entidade, mesmo que não haja anotação explícita para ele.

Nota-se que para trabalhar com *Hibernate*, é interessante que a classe a ser persistida, esteja organizada como um POJO (*plain old java objects*), ou seja, tenha a estrutura de *JavaBeans*, possuindo um construtor padrão, métodos *getters* e *setters*, e um atributo identificador independente da lógica de negócio. Desta forma, com os métodos estruturados dessa maneira, os acessos aos atributos da entidade a ser mapeada, serão melhor controlados, possibilitando permissões de escrita e/ou leitura, como ocorreram com os atributos **id** e **nome**, que foram declarados como privados (*private*), forçando seus acessos para serem possíveis apenas através de seus correspondentes métodos públicos (*public*), *getId()/getNome()* e *setId(int)/setNome (String)*.

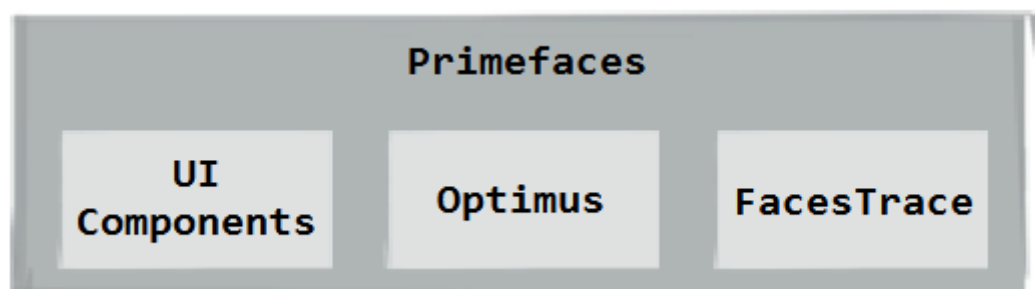
As principais anotações usadas no sistema apresentado neste trabalho são: *@Entity*, *@GeneratedValue*, *@Temporal*, *@JoinColumn*, *@Id*, *@Column*, *@Table*, *@OneToMany* e *@ManyToOne*.

Primefaces v2.2

Ferramenta de código aberto que possui componentes de *interface* com o usuário e outras funcionalidades, para serem usados com *JavaServer Faces*, possuindo ainda, módulos adicionais para aplicações *web* em dispositivos móveis (ÇIVICI, DARÇIN, 2011).

Na Figura 6, temos os módulos do *Primefaces*.

Figura 5: Módulos do *Primefaces*.



O *Primefaces* está organizado em três módulos, os quais serão explicados a seguir (ÇIVICI, DARÇIN, 2011; SANTOS, 2010).

1. Módulo *UI Components*, que engloba os componentes que possuem funcionalidades com *Ajax*, *JavaScript* e gráficos animados, citando alguns exemplo, temos: Componentes baseados em *Flash*, serviços *Ajax*, *HTMLEditor*, *ImagemCropper*, *AutoComplete*, *Captcha*, *Menus*, *ColorPicker*, e outros diversos componentes que o *Primefaces* engloba;

2. Módulo *Optimus*, que oferece soluções pra facilitar o desenvolvimento com *JavaServer Faces*, diminuindo grande quantidade de manipulações de arquivos *.xml*, possibilita gerenciamento de segurança, suporta persistência com *Java Persistence API (JPA)*, permite controle de transações declarativas e anotações baseadas em inversão de controle;

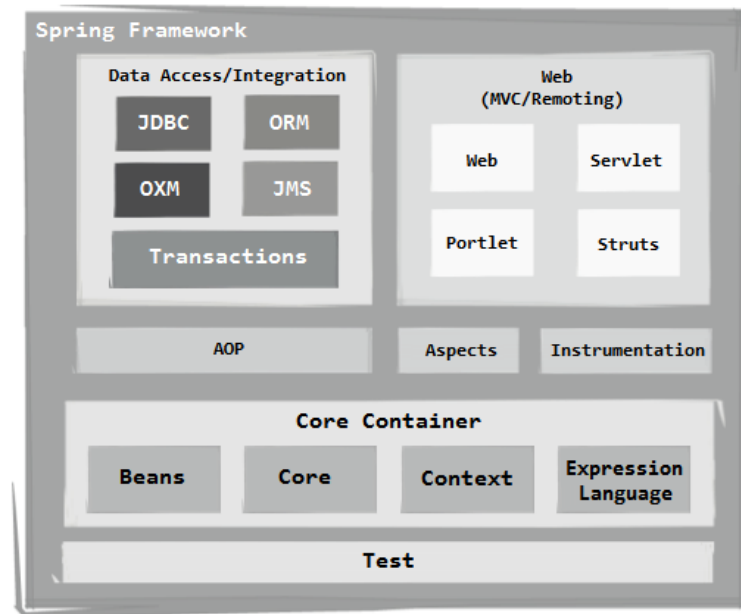
3. Módulo *FaceTrace*, que tem o objetivo ser utilizado como apoio de desenvolvimento, usado também para melhorar a rastreabilidade das aplicações baseadas em *JavaServer Faces* e coletar informações de métricas de desempenho.

Spring v3.0

Framework de código aberto, criado para abstrair certas complexidades no desenvolvimento de aplicações corporativas. Para isso o *Spring* faz uso da ideia do *JavaBeans* (através de *POJO's*) para alcançar objetivos que antes era possível apenas com o *Enterprise Java Beans (EJB)*. Dentre suas diversas funcionalidades, temos que ele: promove injeção de dependência, suporta programação orientada a aspectos e atua como um *container*, controlando e gerenciando o ciclo de vida de seus objetos (WALLS, BREIDENBACH, 2008). Considerando a presente versão, o *Spring* consiste de um conjunto de 20 módulos. Esses módulos são agrupados dentro das camadas denominadas: *Core Container*, *Data Access/Integration*, *Web*, *Aspect Oriented Programming (AOP)*⁵, *Instrumentation* e *Test*, como mostrado na Figura 6 (JOHNSON, 2010).

⁵ Tradução: Programação Orientada a Aspectos (POA), que é um paradigma de programação que permite que os códigos sejam separados e organizados de acordo com a sua relevância para a aplicação.

Figura 6: Módulos do *Spring Framework* v3.0.



A fim de não fugir do propósito deste relatório, faremos uma breve explicação sobre cada camada e módulo do *Spring*.

1. Camada *Core Container*: Sua estrutura é composta dos módulos, *Beans*, *Core*, *Context* e *Expression Language*. Os módulos *Beans* e *Core* são responsáveis pela partes fundamentais do *framework* em questão. O módulo *Context* fica estabelecido na sólida base constituída pelos *Beans* e *Core*. O módulo *Expression Language* possui uma potente linguagem de expressões para consultas e manipula um grafo de objetos em tempo de execução.

2. Camada *Data Access/Integration*: Sua estrutura é composta dos módulos, *JDBC* (*Java Database Connectivity*), *ORM* (*Object-Relational Mapping*), *JMS* (*Java Management Extensions*) e *Transactions*. O módulos *JDBC*, possui uma camada de abstração dele mesmo, com o intuito de remover a necessidade de se criar códigos tediosos provenientes do *JDBC*, além disso ele possui um analisador de códigos para erros específicos, provenientes do proprietário do banco de dados. O módulo *ORM* fornece camadas de integração para API's populares de mapeamento objeto-relacional, incluindo *JPA*, *JDO* (*Java Data Objects*), *Hibernate* e *jBatis*. Através do *ORM* o desenvolvedor poderá usar todos os *frameworks* objeto-relacionais anteriormente citados, combinando com todas as outras funcionalidades do *Spring Framework*. O módulo *OXM* fornece uma camada de abstração que suporta implementações de mapeamento *Object/XML* para *JAXB*, *Castor*, *XMLBeans*, *JiBX* e

XStream. O módulo JMS fornece funcionalidades de criação e consumo de mensagens. O módulo *Transaction* faz o gerenciamento de transação declarativa e programática para classes que implementam *interfaces* especiais e para todos os seus POJO's.

3. Camada *Web*: Esta camada consiste dos módulos: *Web*, *Web-Servlet*, *Web-Struts* e *Web-Portlet*.

4. Camada *Test*: Suporta ferramentas de testes de componentes do *Spring Framework* como *JUnit* ou *TestNG*. Além de algumas outras atividades, ele possibilita a simulação de objetos, gerando a possibilidade de testar partes isoladas do código.

Para o sistema apresentado neste trabalho, as principais anotações usadas do referido *framework* , foram: *@Inject*, *@Component* e *@Transactional*.

3 IMPLEMENTAÇÃO DO SISTEMA

3.1 AMBIENTE DE DESENVOLVIMENTO *ECLIPSE INDIGO*

É um ambiente integrado para desenvolvimento de aplicações computacionais, composto de diversos recursos que auxiliam o desenvolvimento das aplicações como: depuração de código, analisador léxico em tempo de programação, *interface* gráfica amigável e diversos *plugins* facilitadores ao desenvolvedor.

3.1.1 Instalando o *Eclipse Indigo*

O site oficial do *Eclipse Indigo* encontra-se no endereço: <<http://www.eclipse.org/>>. Ao entrar no site, pressione o link a direita com o nome *Download Eclipse*, como mostrado na Figura 7.

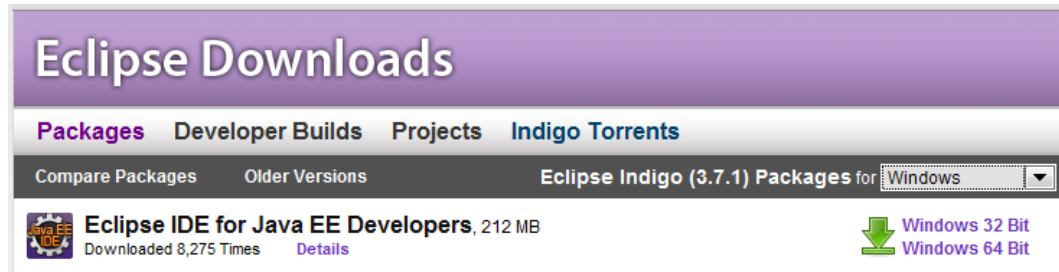
Figura 7: Site oficial do *Eclipse IDE* para baixar o ambiente de desenvolvimento.



Em seguida, selecione o Eclipse de acordo com seu interesse; no caso do sistema apresentado neste trabalho, utilizamos a versão *Eclipse IDE Indigo* através do link *Eclipse IDE for Java EE Developers* (212 MB) (Figura 8). Após efetuar baixar o ambiente em

questão, pressione no ícone do *Eclipse* presente em seu computador e escolha o local (*workspace*) onde irão ser colocados os projetos da aplicação desenvolvida.

Figura 8: Tela com o *link* para baixar o *Eclipse Indigo*.



3.2 SERVIDOR WEB APACHE TOMCAT v7.0

É um servidor desenvolvido em Java, especializado para aplicações *web*, que implementa as tecnologias *JavaServlets* e *JavaServer Pages*, podendo ser usado também como servidor HTTP (APACHETOMCAT, 2011).

3.2.1 Instalando o Apache Tomcat

O site oficial do Apache Tomcat pode ser acessado através do endereço: <http://tomcat.apache.org/>. No lado esquerdo desse site (Figura 9), pressione o link Tomcat 7.0 e escolha a opção do servidor, de acordo com o sistema operacional de sua escolha.

Figura 9: Tela com o *link* para baixar o servidor Apache Tomcat.



3.3 O SISTEMA GERENCIADOR DE BANCO DE DADOS

Para que as informações (ou dados) migradas ou fornecidas ao sistema, não sejam perdidas após finalizada a seção do usuário, foi usado um sistema banco de dados. Desta forma, informações como, cadastros e alterações de dados diversos, históricos de solicitações e remoções, não serão simplesmente perdidas depois de finalizada a seção do usuário, continuarão presentes no sistema sempre que forem requisitadas, não importa quando. Persistir os dados circulantes no sistema, em um banco de dados é uma forma de garantir que as informações passadas pelo sistema não sejam perdidas sem a permissão do usuário comum ou do administrador.

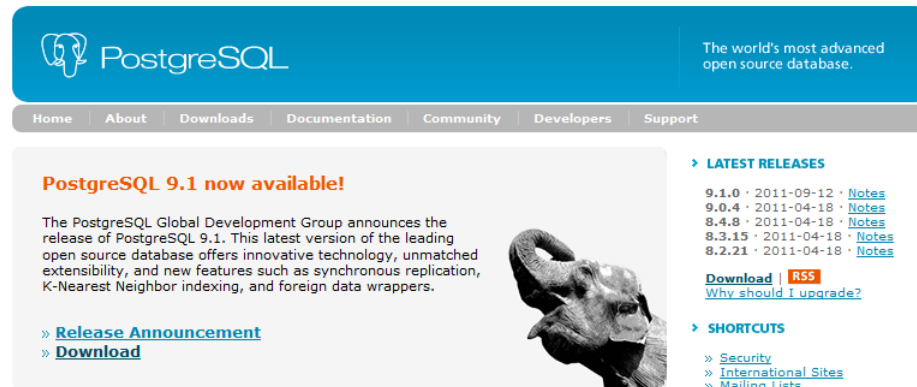
Para concretizar a persistência desses dados, foi usado o *PostgreSQL* v8.1, que é um Sistema de Gerenciamento de Banco de Dados Objeto Relacional (SGBD-OB)⁶ de código fonte aberto; o que significa que qualquer um pode estudar ou alterar seu código fonte, conforme a necessidade mandar. Assim como o SGBD *MySQL*, que é um dos sistemas de gerenciamento de banco de dados mais populares que existe hoje, o *PostgreSQL* também tem a vantagem de poder ser executado em praticamente todos os sistema operacionais, e tanto o *MySQL* quanto o *PostgreSQL* são otimizados para serem usados em sistemas web (SILBERSCHATZ, SUDARSHAN, 2006; MYSQL, 2011).

3.3.1 Instalando o *PostgreSQL*

Para começa a usar o *PostgreSQL*, primeiro é necessário fazer o download dele através do endereço: <<http://www.postgresql.org/>>. Em seguida selecione o link "Downloads" no menu superior (Figura 10) e escolha a versão desse SGBD de acordo com o sistema operacional de seu interesse.

⁶ Banco de Dados Objeto Relacional é semelhante a um banco de dados relacional, porém com um modelo de banco de dados orientado ao objeto.

Figura 10: Tela inicial e de *download* do *PostgreSQL*.



Neste projeto optou-se por usar o *PostgreSQL*, pois ele é otimizado para aplicações complexas, isto é, aplicações que envolvem grandes volumes de dados ou que tratam de informações críticas⁷. Desta forma, para um sistema de comércio eletrônico de porte médio ou alto, por exemplo, o *PostgreSQL* é mais recomendado, já que este SGBD é capaz de lidar de maneira satisfatória com o volume de dados gerado pelas operações de consulta e venda (MYSQL, 2011). Vale salientar que o seus níveis de documentação e suporte, também são satisfatórios. A saber, que a agilidade do SGBD *MySQL*, por exemplo, também supriria as necessidades do sistema em questão da mesma maneira, considerando seu estado atual de execução.

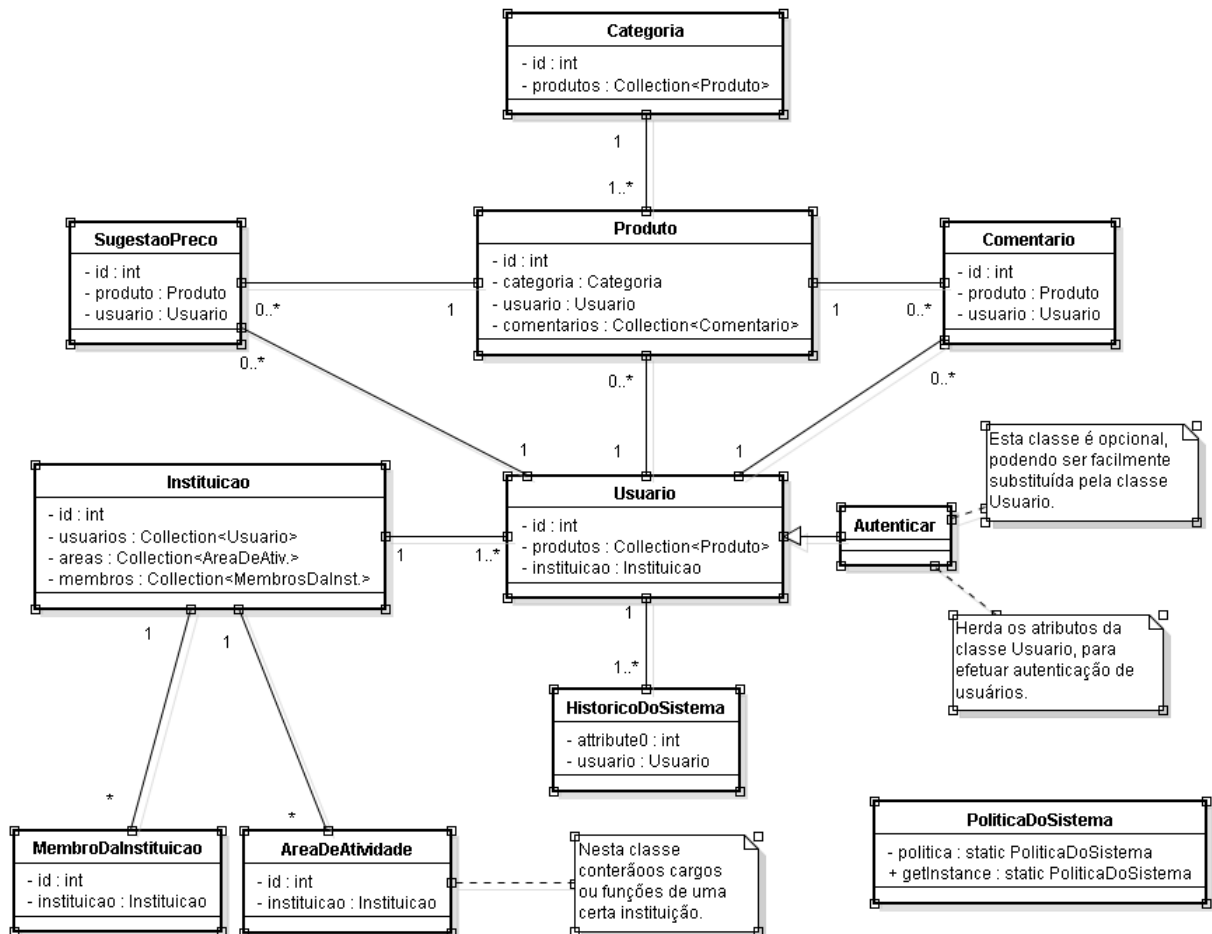
3.4 DIAGRAMA DE CLASSES

Para uma melhor visualização dos relacionamentos existentes entre as entidades implementadas no sistema (até o momento), usaremos o diagrama de classes, como parte da modelagem estrutural básica. Esta modelagem é importante não só para a visualização de como as entidades estão organizadas e para a especificação e documentação de modelos estruturais, mas também para a construção de sistemas executáveis por intermédio de engenharia de produção e reversa (BOOCH, RUMBAUGH, 2006).

⁷ Informações críticas são aquelas informações poucos tolerantes a falhas, ou seja, as transações que lidam com elas, devem ter o mínimo de falhas possíveis e o máximo de garantia de sucesso na conclusão das transações.

Na Figura 11 temos o diagrama de classes referente às entidades que foram persistidas no banco de dados do sistema.

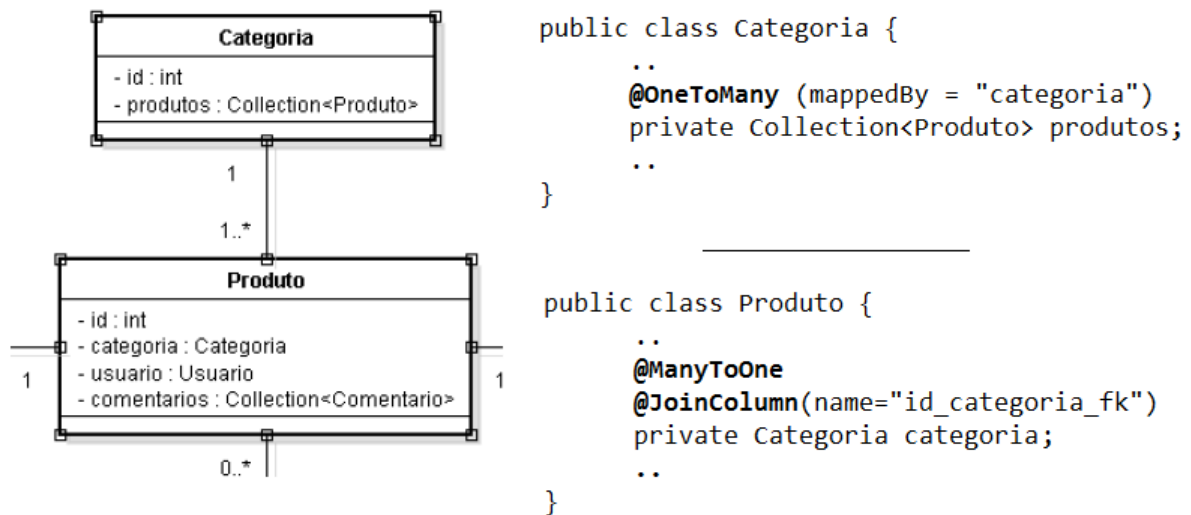
Figura 11: Diagrama de classes referente às classes persistidas no banco de dados.



Vemos que há diversos relacionamentos entre as entidades do sistema. Desta forma, explanaremos apenas o relacionamento entre as entidades Categoria e Produto presente no diagrama de classes em questão. Para os demais relacionamentos, a fundamentação e a implementação são análogas.

Isolando apenas o relacionamento entre as entidades Categoria e Produto, temos o relacionamento um-para-muitos (anotação do *Hibernate @OneToMany*), que é um relacionamento simples de compreender, uma vez que assumimos que cada produto poderá ter apenas uma categoria e que cada categoria poderá ter diversos produtos. Desta forma, o mapeamento desse relacionamento deve ser feito conforme apresentado na figura 12.

Figura 12: Mapeamento do relacionamento entre as entidades Categoria e Produto.



Onde o valor do atributo "categoria" no parâmetro *mappedBy* da classe *Categoria*, significa o nome do atributo na classe *Produto* que representa a classe *Categoria*. E na classe *Produto*, o valor "id_categoria_fk" do parâmetro *name*, significa que teremos uma coluna na entidade *Produto* com o nome "id_categoria_fk", onde ficarão as chaves estrangeiras que serão usadas para referenciar a entidade *Categoria*. Vale salientar que os demais relacionamentos mostrados são modelados de forma semelhante ao relacionamento em questão, tornando desnecessário explicar cada um.

Todas as classes apresentadas no diagrama acima são importantes para o sistema, algumas delas são constantemente usadas, como é o caso das classes *Usuario*, *Produto* e *HistoricoDoSistema*, outras são usadas apenas para retornar valores, como é o caso da classe *PoliticaDoSistema*, que usa o padrão de projetos *singleton*, reforçando a necessidade de apenas uma instância dela para ser usada pelo sistema, associando as ações do sistema com a política do sistema (tarifas, quantidade de produtos e regras de negócio). Isso não exclui sua importância, uma vez que nela conterão toda a política do sistema, tais como uso, direitos e deveres dos usuários cadastrados, também tarifas e quaisquer outros valores a serem cobrados aos vendedores e/ou compradores do sistema.

3.5 DIAGRAMA DE CASOS DE USO

O referido sistema possui três atores: o comprador, o vendedor e o administrador do sistema; e aproximadamente doze casos de uso, dentre estes, alguns são principais como: Cadastrar Usuário, Solicitar Produto, Verificar Solicitação, Cancelar Solicitação, Cadastrar Produto, Procurar Solicitação, Cadastrar Promoção, Finalizar Venda e Efetuar Ações Gerais.

Sabe-se que alguns dos casos de uso acima são específicos para algum tipo de ator do sistema; desta forma iremos especificar mais detalhadamente cada caso, associando-o ao ator correspondente, a fim de aumentar o entendimento do sistema no que se referem às funções dos usuários finais do sistema.

3.6 CASOS DE USO: ATOR COMPRADOR

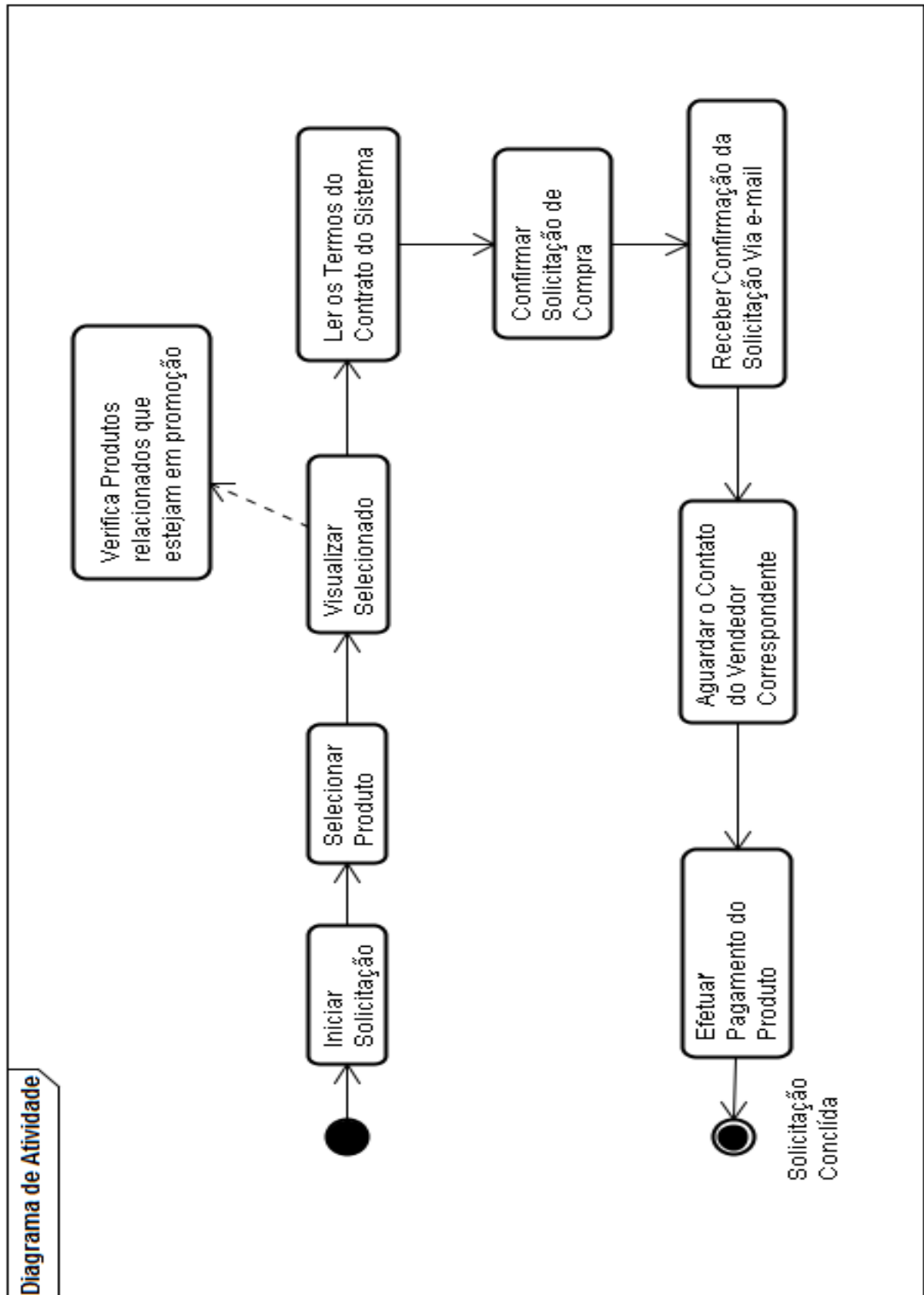
3.6.1 Cadastrar Usuário

- Atores: Usuários comprador, vendedor e administrador.
- Pré-condição: O usuário deve se enquadrar dentro da área na qual o sistema está sendo aplicado. Por exemplo: para funcionários de uma Empresa; para residentes do Estado do Rio Grande do Norte ou até mesmo para alunos e funcionários de Universidades ou Instituições quaisquer de ensino.
- Fluxo de eventos primários (caminho básico):
 1. O caso de uso começa quando o usuário seleciona a ação "Cadastrar Usuário"; fornece os dados necessários e cadastra o usuário no sistema;
 2. O sistema apresenta a confirmação do cadastro do usuário e disponibiliza-o no sistema. O caso de uso termina.
- Pós-condição: Os dados do usuário cadastrado devem ser verdadeiros e atualizados.

3.6.2 Solicitar Produto

- Ator: Usuário comprador.
- Pré-condição: O usuário comprador deve ter feito o cadastro no sistema e se autenticado para acessar certas funcionalidades.
- Fluxo de eventos primários:
 1. O caso de uso começa quando o comprador efetua o cadastro no sistema;
 2. O comprador obtém a autorização do sistema (autentica-se);
 3. O comprador seleciona o produto desejado;
 4. O sistema apresenta ao comprador os dados do produto;
 5. O comprador seleciona a ação "Comprar" para dar seguimento à solicitação de compra;
 6. O sistema apresenta ao comprador os Termos de Gestão de Compras e Vendas;
 7. O comprador confirma a compra;
 8. O sistema comunica (via *e-mail*) ao comprador e ao vendedor que um produto específico foi solicitado para compra;
 9. O sistema envia ao vendedor, um boleto bancário de cobrança, condicionando o pagamento deste à liberação dos dados do comprador;
 10. O vendedor efetua o pagamento do boleto bancário gerado pelo sistema, para receber a liberação dos dados do comprador;
 11. Após a liberação dos dados do comprador, o vendedor comunica-se com o comprador a fim de finalizar a compra e entregar o produto. O caso de uso termina.
- Fluxo de eventos secundários (caminho alternativo): A qualquer momento, antes de submeter à solicitação do pedido, o comprador poderá selecionar a ação "Cancelar", ou simplesmente desistir da compra. Esta ação não gerará solicitação de produto e consequentemente não será gravada. O caso de uso termina.
- Pós-condição: O pedido deve ser gravado no sistema e marcado como finalizado. Abaixo segue o diagrama de atividades correspondente ao caso de uso Solicitar Produto.

Figura 13: Diagrama de atividades relacionado à solicitação de produtos.



3.6.3 Verificar Solicitação

- Atores: Usuários comprador e vendedor.
- Pré-condição: O usuário comprador deve ter solicitado produto (ver seção 3.6.2).
- Fluxo de eventos primários:
 1. O caso de uso começa quando o comprador seleciona a ação "Listar Solicitações";
 2. Usa o caso de uso Procurar Solicitação (ver seção 3.7.2);
 3. O sistema apresenta os dados da solicitação e sua situação, quanto ao andamento da compra. O caso de uso termina.
- Fluxo de eventos secundários: Se no passo 2, o pedido não for encontrado, o sistema informa que o pedido não está cadastrado e solicita que seja verificado se os dados do pedido estão corretos.
- Pós-condição: A situação da solicitação não pode ter sido alterada.

3.6.4 Cancelar Solicitação

- Atores: Usuários comprador e vendedor.
- Fluxo de eventos primários:
 1. O caso de uso começa quando o comprador ou vencedor seleciona a ação "Cancelar Solicitação", ou não conclui os passos necessários para finalizar a solicitação de compra do produto pelo sistema (para o caso do comprador);
 2. Usa o caso de uso Procurar Solicitação (ver seção 3.7.2);
 3. O sistema apresenta os dados e a situação da solicitação após cancelamento. O caso de uso termina.
- Pré-condição: Para o caso de não conclusão da solicitação (segunda parte do passo 1), não há necessidade de manter registrado no sistema; Para o passo 2, é necessário que haja alguma solicitação de compra prévia.
- Fluxo de eventos secundários: Se no passo 2, o pedido não for encontrado, o sistema informa que o pedido não está cadastrado e solicita que seja verificado se os dados do pedido estão corretos.

- Pós-condição: A situação da solicitação deve ser sinalizada como cancelada, e o registro da solicitação deve permanecer no histórico do sistema (com a ressalva apresentada na pré-condição).

3.6.5 Qualificar Vendedor

- Ator: Usuário Comprador.
- Pré-condição: O usuário comprador deve ter feito a solicitação do produto no sistema.
- Fluxo de eventos primários:
 1. O caso de uso começa quando o comprador efetua a solicitação do produto;
 2. O vendedor é informado sobre tal solicitação;
 3. Estando toda a transação comprador–vendedor nos conformes, o vendedor executa as ações inerentes à finalização da venda (ver seção 3.7.4);
 4. Finalizada a venda o comprador é comunicado pelo sistema, sobre sua pendência quanto à qualificação⁸ do vendedor;
 5. O comprador qualifica o vendedor para poder efetuar novas compras, caso contrário o sistema bloqueará novas ação de solicitação de compra.
 6. O sistema calcula o valor todas as qualificações prévias e atuais deste vendedor e apresenta na tela de anúncio do vendedor, uma figura que correspondente ao seu grau de qualificação, conforme o resultado obtido desse cálculo, que pode ser caracterizado como: neutra (quando não há vendas registradas para o vendedor, por isso ainda não foi qualificado), ruim, boa, melhor e ótima, como vemos na Figura 20.
- Fluxo de eventos secundários (caminho alternativo): Sempre que houver solicitação de compra e tal solicitação for finalizada, o sistema gerará pendência de qualificação por parte do comprador, impedindo o mesmo de efetuar novas compras sem antes qualificar o vendedor mais recente. O caso de uso termina quando o comprador efetua a qualificação.
- Pós-condição: O pedido deve ser gravado no sistema e marcado como finalizado e qualificado.

⁸ O objetivo da qualificação do vendedor é de gerar mais segurança nas transações de compra e venda e também de gerar uma troca de informações entre os usuários compradores e vendedores do sistema, a fim de conhecer e expor aos compradores e navegantes do sistema, o perfil do vendedor a qual estão se relacionando.

3.7 CASOS DE USO: ATOR VENDEDOR

3.7.1 Cadastrar Produto

- Atores: Usuários vendedor e administrador.
- Pré-condição: O usuários que irão cadastrar o produto, devem estar cadastrados e autenticados no sistema.
- Fluxo de eventos primários:
 1. O caso de uso começa quando o usuário seleciona a ação "Cadastrar Produto", em seguida os campos referentes aos dados do produto são preenchidos;
 2. Após o preenchimento desses campos, o usuário seleciona a ação "Cadastrar" para cadastrar o produto no sistema;
 3. O sistema apresenta a confirmação do cadastro do produto e disponibiliza-o no sistema. O caso de uso termina.
- Fluxo de eventos secundários: Como apresentado no caso de uso Cadastrar Promoção (ver seção 3.7.3), no passo 1 pode ser colocado desconto logo no momento do cadastro.
- Pós-condição: O dados do produto cadastrado devem estar atualizados.

3.7.2 Procurar Solicitação

- Atores: Usuários comprador e vendedor.
- Pré-condição: Devem existir solicitações de usuários compradores.
- Fluxo de eventos primários:
 1. O caso de uso começa quando o usuário fornece alguns dados que sirvam como parâmetros de busca para a solicitação e logo em seguida seleciona a ação "Buscar";
 2. O sistema apresenta o dado solicitado. O caso de uso termina.
- Fluxo de eventos secundários: Se no passo 1 a solicitação não for encontrada, o sistema deve informar que a solicitação não está cadastrada e solicita que o usuário verifique se os dados fornecidos para busca, estão corretos.
- Pós-condição: A situação da solicitação não deve ter sido alterada.

Este caso de uso é usado por dois outros casos de uso: Verificar Solicitação (ver seção 3.6.3) e Cancelar Solicitação (ver seção 3.6.4).

3.7.3 Cadastrar Promoção

- Estende (*extends*): O caso de uso Cadastrar Produto (ver seção 3.7.1).
- Ator: Usuário vendedor.
- Pré-condição: O produto a ter seu preço colocado em promoção, deve estar cadastrado no sistema.
- Fluxo de eventos primários:
 1. O caso de uso começa quando o vendedor seleciona o produto a colocar em promoção⁹;
 2. O sistema apresenta os campos a serem alterados, como: o valor do produto e percentagem do desconto;
 3. O vendedor adiciona o valor da percentagem desconto no campo "Desconto";
 4. O sistema calcula o valor descontado e atualiza o novo valor do produto no sistema;
 5. A promoção é disponibilizada no sistema e o caso de uso termina.
- Pós-condição: O preço anterior do produto e seu novo preço com desconto, devem ficar disponíveis no sistema, pois assim, o desconto aplicado sob o preço do produto, fica mais claro ao comprador.

3.7.4 Finalizar Venda

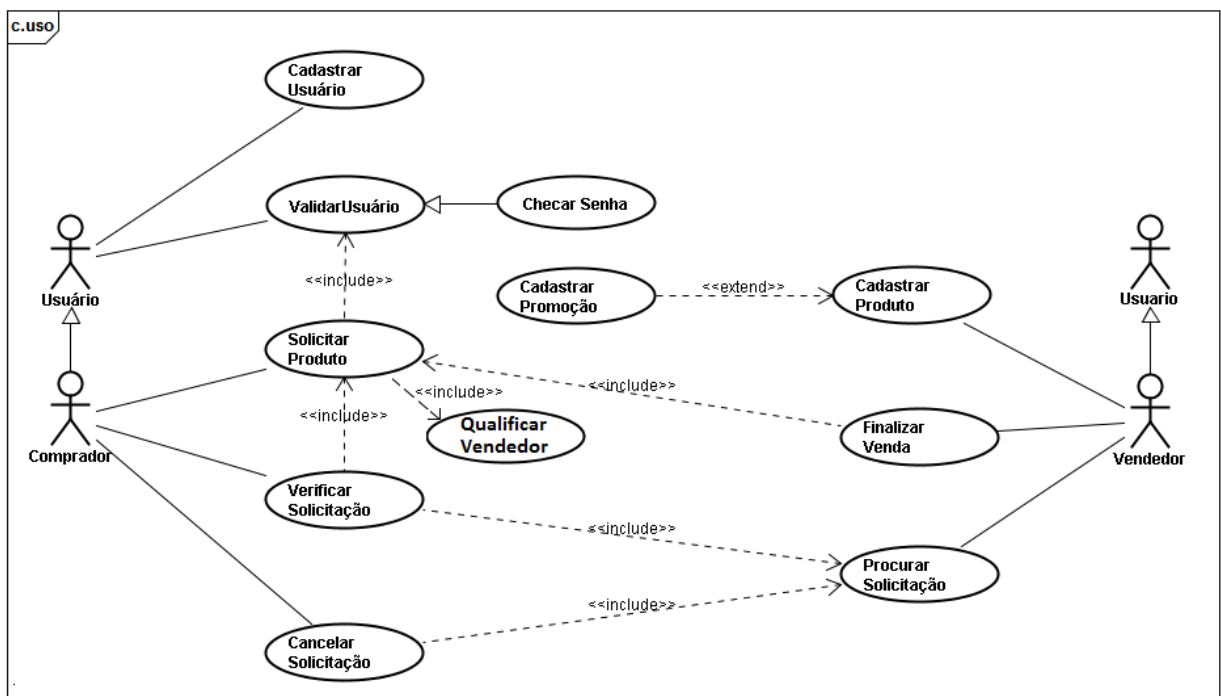
- Ator: Usuário vendedor.
- Pré-condição: Deve existir solicitação de compra de produto.
- Fluxo de eventos primários:

⁹ Pode acontecer de o vendedor colocar o produto em promoção logo no ato do cadastro deste.

1. O caso de uso começa quando o sistema comunica ao vendedor que alguém solicitou a compra de seu produto;
 2. Usa o caso de uso Procurar Solicitação (ver seção 3.7.2);
 3. O sistema gera um boleto bancário ao vendedor, constando o valor a ser pago ao sistema pela venda, conforme os Termos de Gestão de Compras e Vendas;
 4. Constatado o pagamento do boleto bancário pelo vendedor, o sistema libera os dados necessários do comprador para que o vendedor possa entrar em contato e para finalizar a venda;
 5. O vendedor entra em contato com o comprador e passa os dados de pagamento e de frete (caso se aplique);
 6. O vendedor constata o pagamento do comprador e encaminha o produto. O caso de uso termina.
- Pós-condição: A venda (solicitação de compra) deve ser sinalizada no histórico do sistema como, concluída; o caso de uso Cancelar Solicitação (ver seção 3.6.4) será acionado, caso haja desistência da compra, seja qual for o motivo de forma comprovado.

Abaixo temos o diagrama do caso de uso dos usuários comprador e vendedor (ver Figura 14).

Figura 14: Caso de uso referente aos usuários comprador e vendedor.



3.8 CASOS DE USO: ATOR ADMINISTRADOR

3.8.1 Efetuar Ações Gerais

- Ator: Usuário administrador.
- Pré-condição: Está cadastrado no sistema como administrador.
- Fluxo de eventos primários: Este caso de uso engloba outros casos de uso, e começa quando o administrador fornece nome do usuário cadastrado e senha, para autenticar-se no sistema. Depois ele seleciona o tipo de ação que deseja executar, dentre elas temos: cadastrar e remover categorias, cadastrar e remover produtos, cadastrar usuário e atualizar o sistema.

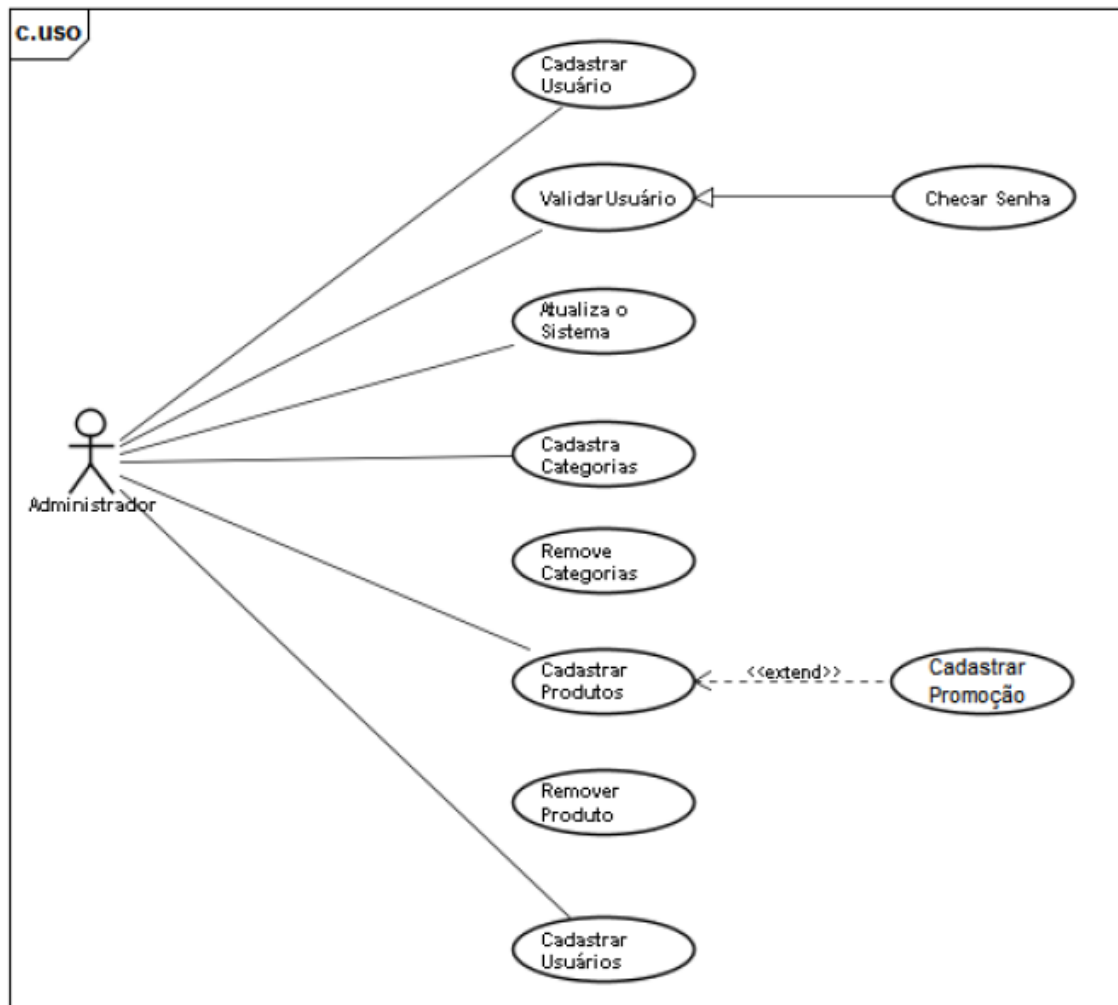
1. Para cadastrar ou remover categorias: selecionamos no menu superior do sistema a ação: "Cadastrar Categoria" ou "Remover Categoria";
2. Para cadastrar ou remover produtos: selecionamos no menu superior do sistema a ação: "Cadastrar Produto" ou "Remover Produto";
3. Para cadastrar usuário: selecionamos no menu superior do sistema a ação: "Cadastrar Usuário";
4. Para atualizar o sistema: O administrador usa ferramentas para edição do código do sistema através de um Ambiente Integrado de Desenvolvimento (AID)¹⁰, ação não visível para os usuários compradores e vendedores.

- Fluxo de eventos secundários: O passo 3 pode ser usado também para cadastrar outros usuários além do próprio administrador, dependerá apenas da necessidade deste.
- Pós-condição: Para as ações de remoção presentes nos passos 1 e 2, é necessário que antes seja selecionada a ação "Listar" para cada caso desejado.

Abaixo temos o diagrama de casos de uso do usuário administrador. Percebemos que esse engloba os principais casos de uso apresentados na Figura 14.

¹⁰ AID ou como é mais conhecido, IDE (*Integrated Development Environment*), nada mais é do que um ambiente gráfico para programação, cercado de diversas ferramentas e outros códigos que possibilitam o desenvolvimento de softwares.

Figura 15: Caso de uso referente ao usuário administrador.



3.9 DIAGRAMA DE SEQUÊNCIA

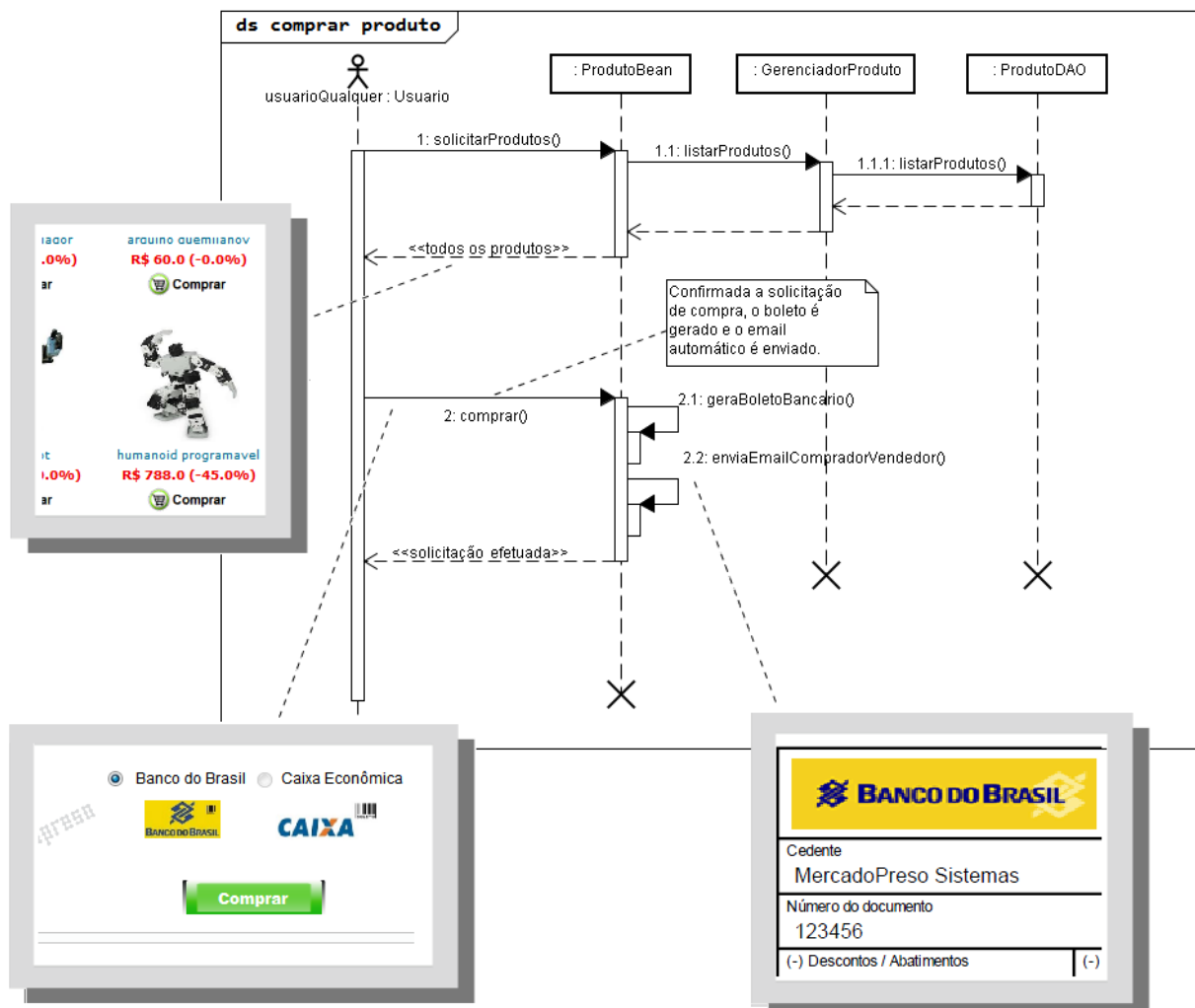
Com a finalidade de mostrar como as mensagens entre os objetos são trocadas no decorrer do tempo, durante a execução de uma certa ação no sistema, iremos mostrar diagramas de sequência. Nesse caso, apresentaremos apenas os diagramas de sequência referentes aos principais casos de uso do sistema.

Na Figura 16, temos o diagrama de sequência referente ao caso de uso Solicitar Produto (ver subseção 3.6.2). Notemos que o usuário solicita o produto através da classe ProdutoBean, que repassa essa solicitação para a classe GerenciadorProduto que então aciona a classe ProdutoDAO, que é responsável por tratar diretamente com os dados persistentes no banco de dados, referentes aos produtos cadastrados no sistema. Uma vez que o sistema retorna a lista de todos os produtos cadastrados ao usuário, este seleciona o produto desejado

e efetua a solicitação de compra como vemos na mensagem 2 (dois) que executa a ação comprar().

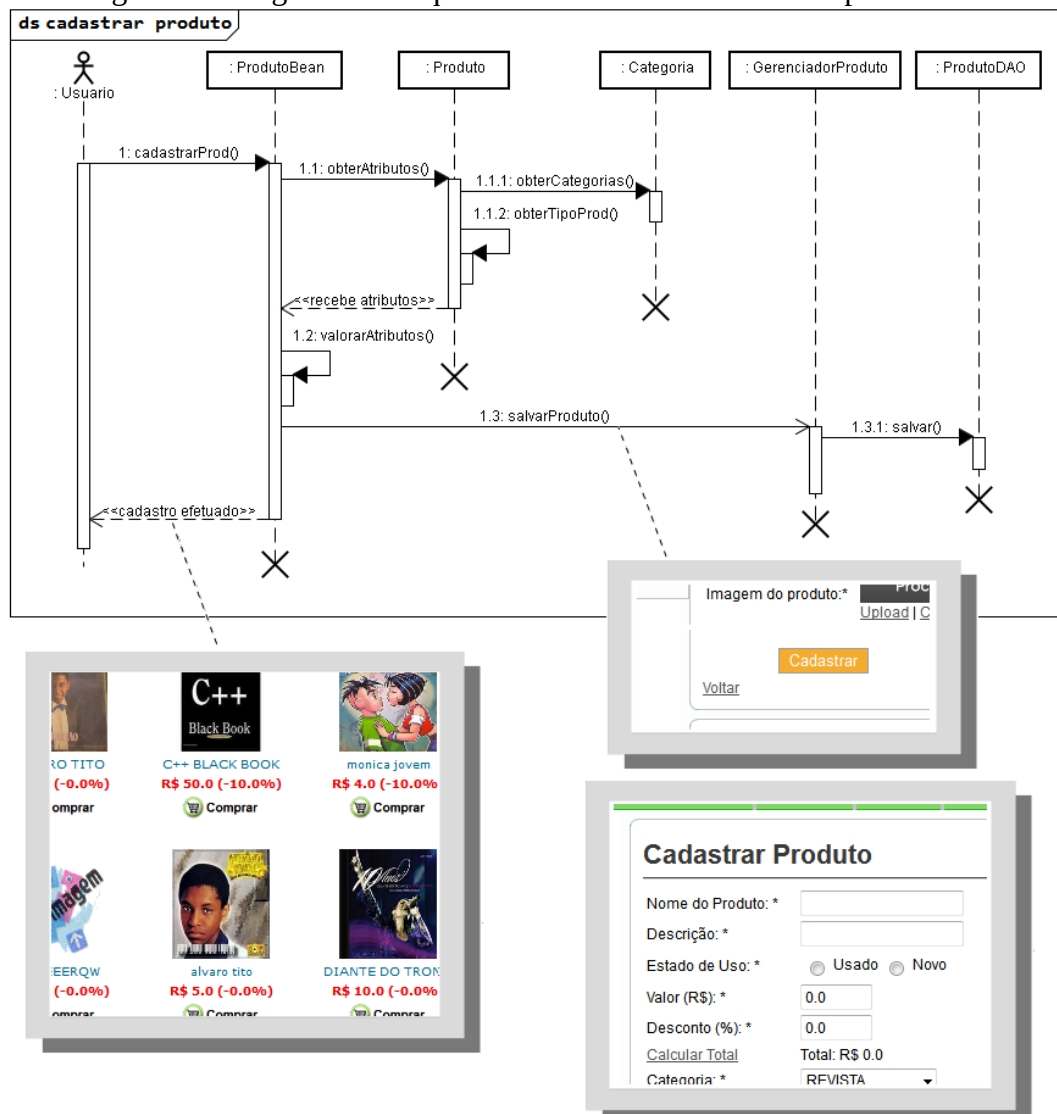
Na figura 17, temos o diagrama de sequência referente ao caso de uso Cadastrar Produto (ver subseção 3.7.1). Iniciando, o usuário inicia o cadastro através da classe ProdutoBean, logo esta obtém as informações referentes aos produtos através da classe Produto. Como um produto deve possuir uma categoria, então a classe Produto recebe informação sobre as categorias através da classe Categoria. Em seguida, após colocadas as informações sobre o produto a ser cadastrado, a classe ProdutoBean dispara a mensagem assíncrona 1.3 (um ponto três) que executa a ação salvarProduto() para a classe GerenciadorProduto, que repassa essa solicitação de cadastro à classe ProdutoDAO para que seja persistido o produto no banco de dados. Confirmado o cadastro, a ação de cadastro encerra.

Figura 16: Diagrama de sequência referente à solicitação de compra de produto.



O sistema em questão foi organizado em camadas a fim de organizar os códigos desenvolvidos. Camada para conter os arquivos de visualização com o usuário (arquivos com extensão .html, .xhtml e .jsp); camada para conter as classes modelos a serem persistidas no banco de dados (arquivos com extensão .java) correspondentes as entidades do banco de dados; camadas para conter as classes *beans* (arquivos com extensão .java como: ProdutoBean, UsuarioBean, etc); e finalmente, camada para conter as classes que se comunicam com o banco de dados (também arquivos com extensão .java), que são responsáveis pelo processamento dos dados circulantes e permanentes do sistema, como os componentes gerenciadores (GerenciadorProduto, GerenciadorUsuario, etc). Desta forma, as ações do sistema, são passadas entre essas camadas, como vemos na ação Cadastrar Produto da Figura 17, onde a solicitação de cadastro é passada para a classe GerenciadorProduto, que repassa essa solicitação a classe ProdutoDAO, para persistir o novo produto.

Figura 17: Diagrama de sequência referente ao cadastro de produtos.



3.10 OUTRAS FUNCIONALIDADES DO SISTEMA

3.10.1 Sugestão de Preço

Esta funcionalidade do sistema possibilita ao usuário comprador, sugerir melhores preços aos vendedores sobre os produtos anunciados que são do interesse do comprador, servindo como uma forma de negociação entre ambas as partes. Ao ser gerada uma sugestão de preço, o sistema envia um *e-mail* ao vendedor do produto correspondente, comunicando-o sobre a sugestão feita, como mostrado na Figura 18; uma vez tendo sido comunicado, o vendedor poderá responder à sugestão, através do sistema.

Figura 18: Tela de sugestão de novo preço de produto e o *e-mail* enviado ao vendedor, comunicando a sugestão do usuário.



3.10.2 Comentários de Anúncios

Esta funcionalidade do sistema disponibiliza aos usuários do sistema, um espaço para comentar sobre os anúncios dos produtos, a saber, que esses comentários são previamente filtrados antes de serem postados, para evitar que informações pessoais e palavras impróprias sejam divulgadas nesse espaço. Pois o sistema não permite que informações pessoais (telefones, email, endereços, etc), sejam estas dos compradores ou vendedores, sejam divulgadas no sistema, seja qual for o espaço.

A Figura 19 apresenta a tela de comentários do sistema e na Figura 20, temos a tela de qualificação de vendedor e sua representação gráfica correspondente ao cálculo de todas as qualificações desse vendedor.

Figura 19: Tela de comentário do sistema, no espaço de anúncio do produto.

Figura 20: Tela de qualificação de vendedor e representação gráfica.

4 RESULTADOS

Nas figuras abaixo, temos as telas do sistema como: página inicial, tela individual de produto contendo o campo de sugestão de preço, *link* de qualificação de vendedor e campo de comentário de produto e vendedor.

Figura 21: Tela inicial do sistema com permissões de usuário administrador e usuário comprador.



Na Figura 22, temos a sequência de passos até a finalização da ação solicitar produto, como: seleção do produto, escolha do banco para geração do boleto de pagamento e o envio automático de *e-mails* pelo sistema, ao comprador e ao vendedor. Sendo este último, de responsabilidade do sistema.

Figura 22: Esquema do funcionamento da ação de solicitação e compra de produto



Na Figura 23, temos a tela individual de cada produto, que é mostrada quando for selecionado; nela contem a especificação detalhada do produto cadastrado, bem como a situação de qualificação do vendedor do produto apresentado na tela (canto superior direito), os comentários inseridos pelos compradores, a opção da agência bancária de preferência para a geração do boleto bancário, as últimas sugestões de preço enviadas ao vendedor, o campo contendo os melhores preços dos produtos relacionados e o *link* de qualificação do vendedor, caso o comprador tenha alguma qualificação pendente.

Lembrando que se o comprador não qualificar o vendedor de sua última transação de compra finalizada, automaticamente o sistema redireciona sua ação de compra (através do botão “Comprar”), para a página de qualificação do vendedor; caso não execute a qualificação pendente, sua nova solicitação de compra será bloqueada.

Figura 23: Tela que antecede a solicitação de compra, específica para cada produto.

CADASTRE SEU PRODUTO AGORA..

FlashVortex.com

[Principal](#) | [Categoria](#) | [Produto](#) | [Instituição](#) | [Usuário](#) | [Membros da Inst.](#) | [Contato](#)

Consolidar Compra




humanoid programavel

Preço: **R\$433.4**  **FRETE GRÁTIS**



robô programável. ideal para programadores e hobbistas da área.
robotica
Estoque: 4 unidade(s)

 **Melhores Preços**

 humanoid programavel \$433.4	 kit vex robotica \$540.0	 kit de robotica \$450.0	 robô isobot \$207.0	 microncontrolador \$54.0
---	---	---	--	---

13 sugestões

 \$1.0 14/09/11	 \$1.0 14/09/11	 \$1.0 14/09/11	 \$1.0 14/09/11	 \$1.0 14/09/11
 \$1.0 14/09/11	 \$1.0 14/09/11	 \$1.0 14/09/11	 \$1.0 14/09/11	 \$1.0 14/09/11

Sugerir novo valor:

1 comentários



postado em 05/09/11 10:12 : gostei.. bom vendedor.

Postar um comentário

baixar:  **PDF** Li e concordo com os termos supracitados ☐

☒ Banco do Brasil
 ☐ Caixa Econômica




Figura 24: Tela inicial completa do sistema com permissões de usuário administrador.

QUER COMPRAR, VENDER, DIVULGAR?!

[FlashVortex.com](#)

Principal	Categoria	Produto	Instituição	Usuário	Membros da Inst.	Contato
---------------------------	---------------------------	-------------------------	-----------------------------	-------------------------	----------------------------------	-------------------------

Eletrônicos, Livros, Ferramentas variadas, DVD's .

Produtos

- Livros
- Revistas
- Eletrônicos
- Gibis
- Roupas
- Vinils LP
- CDs
- Fitas K7
- Laboratoriais
- Artesanatos
- Todos

Meu MercadoPreso

- Minhas Sociliações
- Qualificar Vendedor

cavaleiros zodiaco
R\$ 4.0 (-0.0%)
[Comprar](#)

php programming
R\$ 34.0 (-5.0%)
[Comprar](#)

elvis presley
R\$ 10.0 (-0.0%)
[Comprar](#)

turma da monica
R\$ 1.0 (-0.0%)
[Comprar](#)

sem imagem
R\$ 11.0 (-0.0%)
[Comprar](#)

ALVARO TITO
R\$ 3.0 (-0.0%)
[Comprar](#)

C++ BLACK BOOK
R\$ 50.0 (-10.0%)
[Comprar](#)

monica jovem
R\$ 4.0 (-0.0%)
[Comprar](#)

toque no altar
R\$ 10.0 (-0.0%)
[Comprar](#)

sem imagem
R\$ 5.0 (-0.0%)
[Comprar](#)

alvaro tito
R\$ 5.0 (-0.0%)
[Comprar](#)

DIANTE DO TRONO
R\$ 10.0 (-0.0%)
[Comprar](#)

nature
R\$ 5.0 (-0.0%)
[Comprar](#)

jaleco p/ medico
R\$ 45.0 (-5.0%)
[Comprar](#)

Sushi Prático
R\$ 10.0 (-0.0%)
[Comprar](#)

AMERICAN SCIENC
R\$ 5.0 (-10.0%)
[Comprar](#)

sem imagem
R\$ 22.0 (-3.0%)
[Comprar](#)

RAISSA E RAVEL
R\$ 5.0 (-0.0%)
[Comprar](#)

pic 16f77a/x
R\$ 5.0 (-0.0%)
[Comprar](#)

kit de robotica
R\$ 500.0 (-10.0%)
[Comprar](#)

microncontrolador
R\$ 60.0 (-10.0%)
[Comprar](#)

artesanato
R\$ 20.0 (-10.0%)
[Comprar](#)

FGFGGG
R\$ 11.0 (-0.0%)
[Comprar](#)

jaleco medico
R\$ 8.0 (-10.0%)
[Comprar](#)

humanoide programavel
\$433.4

kit vex robotica
\$540.0

artes com madeira
\$18.0

microncontrolador
\$54.0

monica jovem
\$3.6

sem imagem
\$4.5

MERCADOPRESO

FlashVortex.com

Seja bem vindo!

valber, Sair

FlashVortex.com

Contato | Termos do site

Copyright © 2011 — Todos os Direitos Reservados MP.

5 CONCLUSÃO

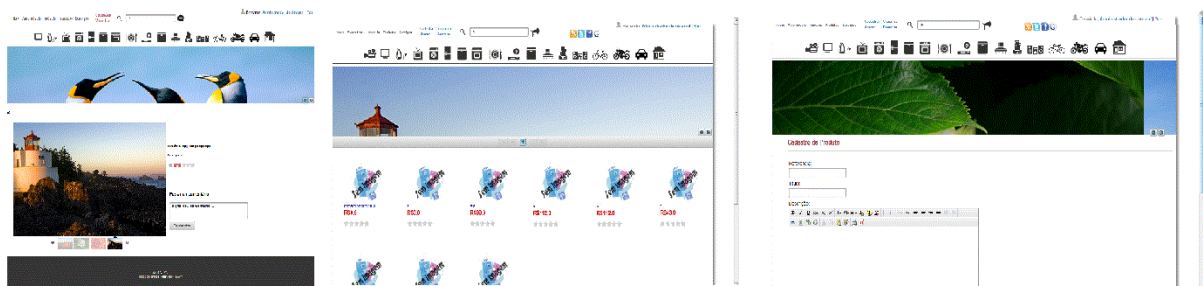
O curso de Especialização em questão contribuiu expressivamente para o desenvolvimento deste sistema, pois através das tecnologias de desenvolvimento *web* ensinadas foi possível implementar e executar da melhor maneira este trabalho.

Desta forma sistema apresentou o resultado esperado, dentro dos requisitos mínimos necessários para alcançar os objetivos expostos neste trabalho. Onde a versão atual apresenta funcionalidades necessárias para gerar uma transação de compra ou venda, mas é clara a necessidade de melhorias, tanto no aspecto de aparências (visual) quanto no aspecto funcionalidades. Como o sistema tem a característica de ser muito dinâmico, sempre haverá a necessidade de modificações com objetivo de melhorar o sistema em todos os aspectos.

5.1 PERSPECTIVAS FUTURAS

- Melhorar a segurança e a confiabilidade das transações de vendas;
- Garantir a confiabilidade e qualidade das qualificações dos usuários compradores aos seus vendedores, a fim de ficarem mais claros os perfis dos vendedores, deixando as transações de vendas, mais confiáveis e transparentes possíveis;
- Estender os boletos gerados para mais agências bancárias;
- Disponibilizar um fórum de discussão do sistema;
- Concluir a expansão do sistema, para divulgar além de produtos, outros serviços como: divulgar anúncios de automóveis, imóveis e serviços diversos, como mostrado na Figura 25, onde a continuidade do sistema apresentado neste trabalho está sendo desenvolvido para se tornar um portal completo de vendas, e em breve ser disponibilizado no mercado.

Figura 25: Protótipos da segunda versão (expansão) do sistema apresentado neste trabalho.



Referências

SILBERSCHATZ, Abraham; KOTH, Henry F.; SUDARSHAN, S.. **Sistema de Banco de Dados**. 5. ed. Rio de Janeiro: Campus, 2006.

LOSHIN, Pete; VACA, John. **Eletronic Commerce**. Barnes and Nodle, 4. ed. São Paulo: Networking Series, 2009.

FILIPINE, Dailton. **Profissões da Internet**. Disponível em: <<http://www.e-commerce.org.br/profissoes-internet.php>>, 2008. Acesso em: 10 set. 2011.

GOPALKRISHNAM, J.; GUPTA, V. K.. **eBay: The world's largest online marketplace: A Case Study**. Institute of Management, Nirma University of Science and Tecnology, Ahmebadad; Intitute of Management, Indore, India, 2007.

WALLS, Craig; BREIDENBACH, Ryan. **Spring in Action**. 3. ed. Manning Publication, 2011.

GEARY, David; HORSTMANN, Cay. **Core JavaServer Faces**. 3. ed. Prentice Hall, 2010.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivan. UML 2.0: **Guia do Usuário**. 6. ed.. Campus/Elsevier, 2006.

LINWOOD, Jeff; MINTER, Dave. **Beginning Hibernate: An introduction to persistence using Hibernate 3.5**. 2. ed.. Apress, 2010.

Banco de Dados MySQL e PostgreSQL: Uma comparação. Disponível em: <<http://www.infowester.com/postgresqlmysql.php>>. Disponível em: 20 abr. 2008. Acesso em: 12 set. 2011.

ROBSON, Geraldo. **Comércio Eletrônico**. Belo Horizonte: Departamento de Ciência da Computação - UFMG.

MENDES, Marcos. **O Comércio Eletrônico no Brasil**. Macapá: Instituto de Ensino Superior do Amapá, 2008.

ALMEIDA, Guilherme A. Almeida; AVILA, Alfonso; BONCANOSKA, Violeta. **Promoting e-Commerce in Developing Countries**. 2007. Disponível em: <<http://diplomacy.edu>>. Acesso em: 1 ago. 2011.

ÇIVIÇİ, Çagatay; DARÇIN, Yigit. **Primefaces: User's Guide**. Disponível em: <<http://www.primefaces.com>>. Acesso em: 10 ago 2011.

EFENDIOGLU, Alev M.; YIP, Vicent F.; MURRAY, William L.. **E-Commerce in Developing Countries: Issues and Influences**. University of San Francisco, 2003.

E-Bit: Confiança na compra online. Disponível em: <<http://www.ebit.com.br>>

The Apache Tomcat: Software Foundation. Disponível em: <<http://tomcat.apache.org>>. Acesso em: 20 abr. 2011.

SANTOS, Anderson C. B. dos et al. **Desenvolvimento Web com PrimeFaces:** Uso no projeto Banco de Dados Pragas Quarentenárias. Campinas-SP: Embrapa Informática Agropecuária, 2010.

JOHNSON, Rod et al. **Spring Framework 3.0:** Reference Documentation. 2010. Disponível em: < <http://static.springsource.org/spring/docs/3.0.x/spring-framework-reference/html/> >. Acesso em: 3 abr. 2011.