

SISTEMA PARA O GERENCIAMENTO DE VENDAS DA EMPRESA CARMOTEC INFORMÁTICA

Guilherme Geraldo de Queiroz¹

Fernando Corrêa de Mello Junior²

Resumo: O objetivo deste artigo é apresentar as principais características relacionadas ao comércio eletrônico. As empresas de *e-commerce* necessitam de ferramentas para controlar e gerenciar suas vendas. O desenvolvimento do projeto possibilitou uma visão sobre os aspectos relacionados à venda de produtos de informática. Diante disso, foi desenvolvido um sistema para controlar as tarefas administrativas da empresa. Foi possível analisar os principais processos referentes ao *e-commerce* com a transição eletrônica para compra e venda de produtos e serviços oferecidos pela internet. Para realizar o trabalho com o comércio eletrônico foi criado um site que irá gerenciar as vendas por meio da utilização de um ambiente virtual. O desenvolvimento deste projeto demonstrou grande crescimento e produtividade nas tarefas realizadas na empresa.

Palavras-chave: Internet. Comércio Eletrônico. E-commerce. Loja Virtual. Processo Unificado.

1 INTRODUÇÃO

A *Internet* é a maior rede de comunicação em escala mundial. O processo de comunicação teve um grande avanço com base na necessidade de maior disponibilidade de acesso às informações. Com o surgimento da *Internet* pode-se destacar pontos fortes como a evolução pela troca de informações em grandes distâncias e tempo real. Ela permite carregar uma grande quantidade de recursos e serviços para transferência de dados e compartilhamento de arquivos. O uso dela torna-se um meio de comunicação indispensável para as empresas. O comércio eletrônico surgiu com a necessidade das empresas trabalharem com a compra e

¹ Graduando em Sistemas de Informação pelo Centro Universitário de Patos de Minas – UNIPAM – e-mail: guilherme-si@live.com

² Mestre em Engenharia Elétrica e Docente do Centro Universitário de Patos de Minas – UNIPAM – e-mail: fernandocmjr@unipam.edu.br

venda de produtos. As lojas virtuais têm crescido bastante visto que os custos são menores do que uma loja física. Deve-se ressaltar que o uso desta ferramenta possibilita acesso rápido e eficaz as informações. A *internet* tornou-se bastante popular para as pessoas, o que gerou uma grande facilidade para comprar sem ter que sair de casa. Por estes motivos, ressaltamos que houve uma grande colaboração para que as empresas dessem importância para o comércio eletrônico como uma boa oportunidade para melhoria de seus negócios.

Os estudos apontam que a internet possui grandes benefícios para que as empresas possam aumentar o seu faturamento. Levando em consideração a agilidade no atendimento ao consumidor, formas de pagamento e entrega e comprometimento ao prazo estipulado tem a garantia de obter sucesso nas vendas pela *Web*. Também, deve-se considerar o uso desta ferramenta com o princípio de que não se deve pensar somente no produto em si, e sim no público que se deve atingir e a concorrência a enfrentar com outras empresas.

O sistema que foi desenvolvido propôs uma gestão para melhoria das vendas, com a finalidade de divulgar e expor melhor os seus produtos. O sistema minimizou os problemas com o tratamento das informações, realizou uma gestão de vendas desde a busca do produto pelo cliente até o pós-venda, garantindo uma gestão para medir a satisfação dos clientes com os produtos adquiridos, transmitindo uma imagem positiva da empresa e explorando novos mercados de atuação através da focalização no marketing do sistema implantado aos clientes.

2 REFERENCIAL TEÓRICO

2.1 E-COMMERCE

O comércio eletrônico ou *e-commerce* é mantido através da comercialização de produtos por meios eletrônicos. É um processo que gerou grande crescimento na era moderna para vários tipos de negócios. Uma das características mais importantes deste modelo de compra é a facilidade de adquirir produtos. A *Web* proporcionou grande avanço para que esta ferramenta pudesse crescer e ter espaço dentro do mercado, pois ela fornece várias informações como folhetos eletrônicos e propagandas, o que torna uma grande amostra para que os clientes possam comprar.

Para Albertini (2001), comércio eletrônico pode ser definido como um mecanismo onde a compra e a venda de bens e serviços ocorre por meio da utilização de tecnologias de comunicação e informação, num ambiente eletrônico como a *internet*. Uma das vantagens

que, pode-se encontrar é que nesse tipo de serviço a busca por produtos difíceis de serem encontrados se torna algo muito fácil e rápido, de forma que possa satisfazer as necessidades dos clientes.

O atendimento ao cliente deve ser da maneira mais simples possível para que possa gerar comunicação com o software de gestão de vendas e assim garantir a qualidade no atendimento. Uma boa loja virtual é aquela que demanda diversas possibilidades para que o cliente possa interagir com o sistema, mantendo uma relação amigável de forma que o cliente possa encontrar informações com clareza. Além disso, o sistema deve representar mecanismos de buscas para que os consumidores possam acionar o processo de vendas (ALBERTINI, 2001).

2.2 LOJA VIRTUAL

O conceito de loja virtual se refere a uma página *Web* que permite ao cliente acessar todas as informações referentes ao comércio eletrônico. Nela será encontrado todo tipo de produto disponível para comprar. Possui um software para gerenciar a venda e finalizar uma compra (SEBRAE, 2011).

O cliente pode navegar pelo site de compras e escolher qual produto deseja comprar, ou seja, lojas online permitem gerenciar melhor o processo de venda. Diante deste contexto, é importante ressaltar que a gestão de vendas possui um suporte completo para que possam auxiliar os clientes nas decisões de suas compras.

3 METODOLOGIA

A metodologia que foi utilizada para o desenvolvimento deste sistema foi baseada pelas fases e fluxos de trabalho do Processo Unificado. O ciclo de vida para o desenvolvimento deste trabalho é um modelo do processo de desenvolvimento de software iterativo e incremental.

A fase de concepção envolveu decisões em relação à proposta técnica comercial do projeto. A partir dessa proposta foi possível definir as estimativas de custo, cronograma e

escopo do sistema. Além disso, também foi possível identificar quais riscos iriam afetar o desenvolvimento do sistema.

A fase de elaboração envolveu a captura dos requisitos funcionais. A partir disso foi possível elaborar a criação dos casos de uso que foram especificados em detalhes. Foi definida a arquitetura utilizada para o desenvolvimento do sistema, além do escopo e uma solução para os principais riscos levantados.

A fase de construção envolveu a codificação do sistema. Para o desenvolvimento deste projeto foram utilizados os seguintes componentes:

- *Java*: linguagem de programação orientada a objetos utilizada para implementar o projeto;
- *Visual Paradigm for UML 10.0*: software para modelagem UML;
- *Mockup Builder*: para realizar a prototipação de telas do sistema;
- *Eclipse IDE for Java EE Developers*: ambiente de desenvolvimento para codificação na linguagem Java;
- *MySQL*: sistema de gerenciamento de banco de dados (SGBD), que utiliza a linguagem SQL (*Structured Query Language*) para a persistência de dados;
- *MySQL Query Browser*: ferramenta gráfica utilizada para modelagem de dados e realização de operações administrativas como, configurar, monitorar, parar e iniciar um serviço do *MySQL* e fazer backup;
- *JSF 2.0* e *Primefaces 3.4 frameworks* MVC e componentes de interface;
- *JPA 2.0* e *Hibernate 3.5 frameworks* de mapeamento objeto-relacional.

Por último, a fase de transição envolveu a entrega do produto final ao cliente. Nessa fase foram realizados testes para a descoberta de falhas e problemas no sistema. A partir de uma falha identificada é realizada a sua correção.

A Tabela 1 representa cada fase do Processo Unificado e uma descrição do que foi realizado durante o desenvolvimento do sistema.

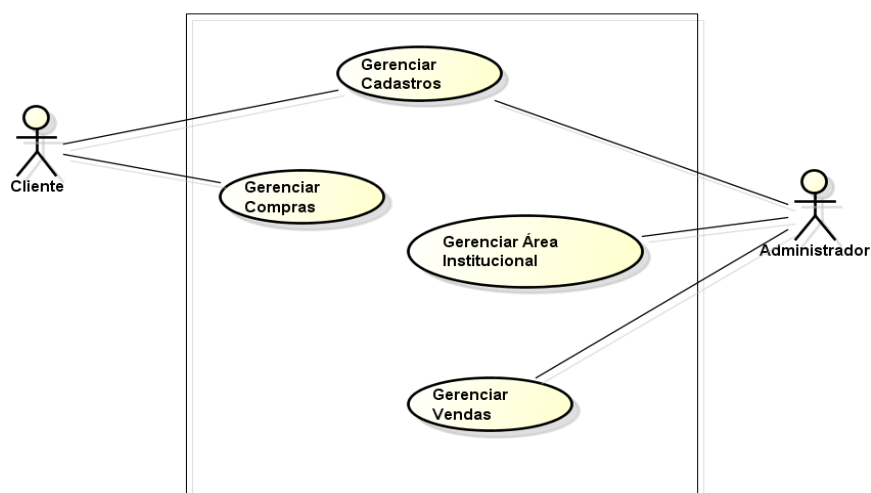
Tabela 1 – Descrição de Iterações do Processo Unificado

Iteração	Descrição
Concepção	<ul style="list-style-type: none"> - Pesquisa sobre sistemas similares a uma loja virtual - Levantamento de requisitos - Diagrama de caso de uso que modela o contexto - Captura dos requisitos não funcionais - Diagrama de Caso de uso que modela o requisito
Elaboração	<ul style="list-style-type: none"> - Prototipagem das principais telas - Descrição funcional dos casos de uso - Elaboração do DER lógico - Definição da arquitetura - Diagrama de classe de domínio
Construção	<ul style="list-style-type: none"> - Codificação dos casos de uso - Criação do banco de dados físico - Testes unitários e testes de sistema durante as iterações
Transição	<ul style="list-style-type: none"> - Atualização dos requisitos - Correção de bugs - Realização de testes unitários e testes de sistema para implantar a versão final do sistema.

Fonte: Dados do Trabalho

4 DESENVOLVIMENTO

A Figura 1 representa os principais módulos que foram utilizados para implementação do sistema.

Figura 1 – Diagrama de Caso de Uso que Modela o Contexto**Fonte: Dados do Trabalho**

O sistema possui dois atores responsáveis por gerenciar o funcionamento do sistema:

- Administrador: este ator tem a responsabilidade de gerenciar os cadastros no sistema, controlar o estoque dos produtos e acompanhar todas as vendas realizadas;
- Cliente: este ator tem a responsabilidade de gerenciar o seu cadastro na loja e gerenciar suas compras.

Os principais requisitos do sistema foram representados através de módulos e que fizeram parte do desenvolvimento são:

- Gerenciar Cadastros: sua responsabilidade é de gerenciar os cadastros básicos da empresa que correspondem a todas as informações básicas para o funcionamento do sistema;
- Gerenciar Compras: sua responsabilidade é de proporcionar ao cliente acompanhar suas compras;
- Gerenciar Área Institucional: sua responsabilidade é de gerenciar a parte institucional da empresa. A gestão deve permitir aos usuários ter uma boa visão sobre a empresa;
- Gerenciar Vendas: sua responsabilidade é de gerenciar as vendas realizadas permitindo obter informações gerenciais para ajudar nas decisões estratégicas da empresa.

4.1 VISÃO FUNCIONAL

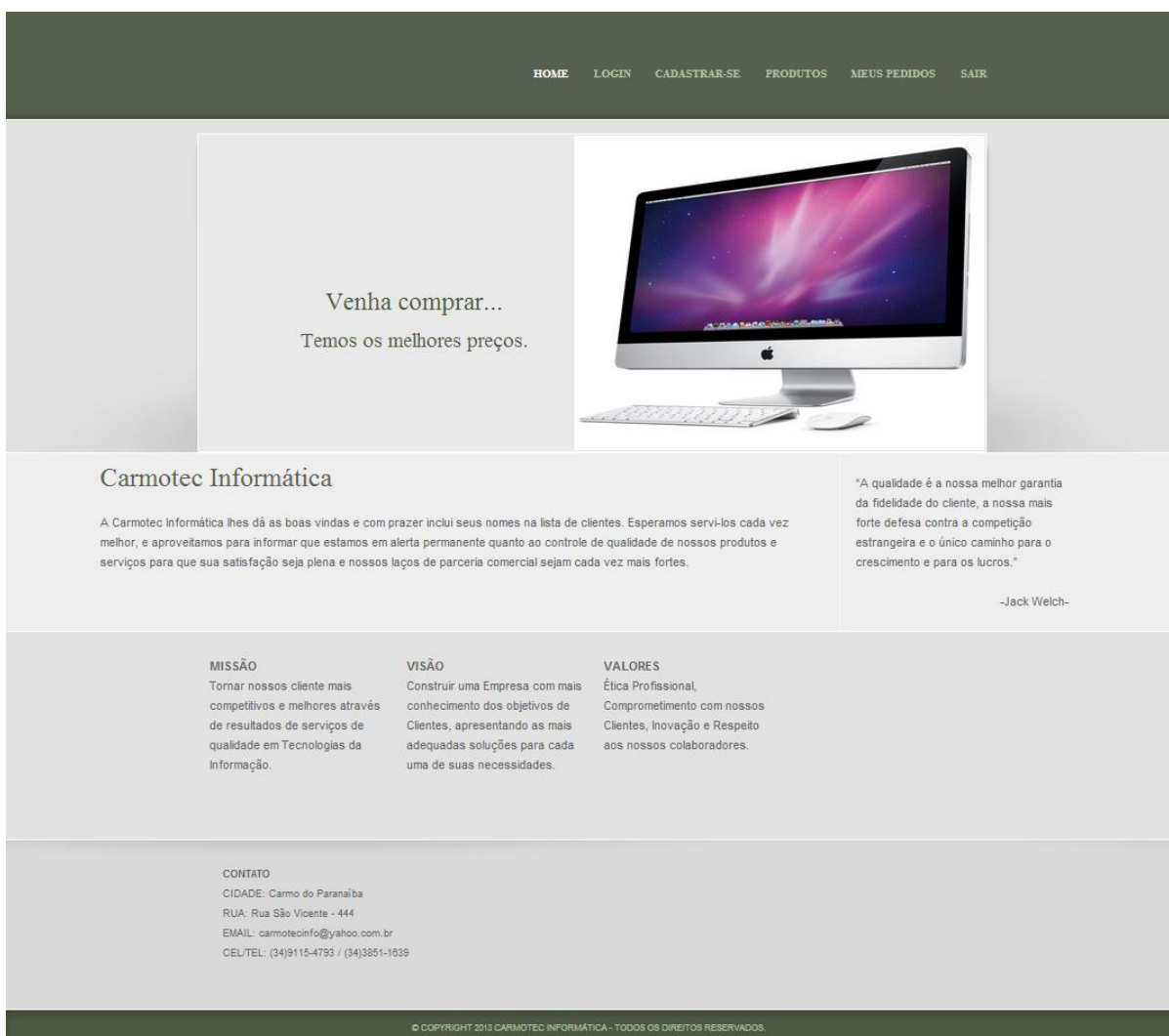
O sistema possui três áreas que foram desenvolvidas e que englobam todas as funcionalidades do sistema:

- Área pública voltada para o cliente: nesta área do sistema encontra-se a parte visual do sistema, onde o cliente consegue visualizar a página principal onde é apresentada a parte institucional da empresa, bem como, os produtos disponíveis para compra;
- Área restrita voltada para o cliente: através da área pública, o cliente acessa a sua área restrita que consiste na parte de cadastramento de seus dados no sistema, login e a parte de vendas (carrinho de compras e acompanhamento de seus pedidos);

- Área restrita voltada para o administrador: o administrador por sua vez contará com uma área onde terá acesso a todas as funções do sistema, desde a parte de todos os cadastros básicos para o funcionamento do mesmo, até a parte de vendas.

A Figura 2 ilustra a tela pública voltada para o cliente, bem como, apresentação das informações a respeito da área institucional e do cadastro de clientes. Além disso, também possui um sistema de login para que o cliente possa acessar dados referentes à compra de produtos.

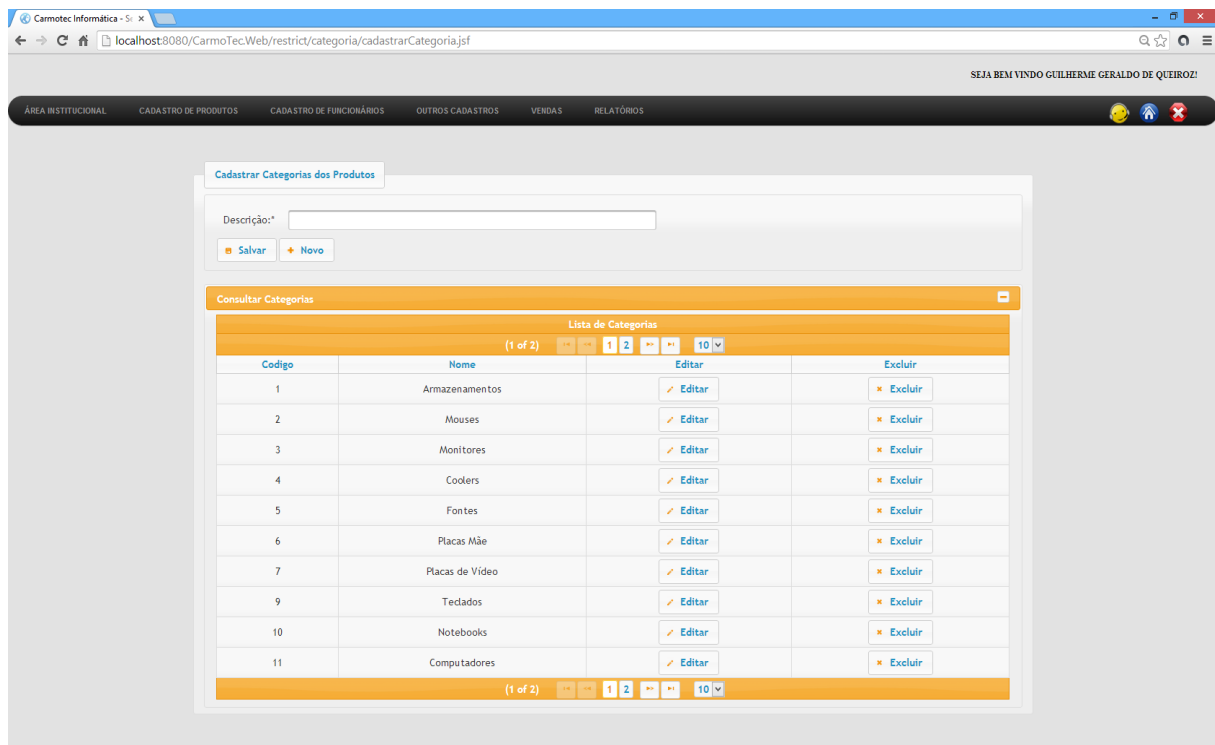
Figura 2 – Tela Principal de Vendas



Fonte: Dados do Trabalho

A Figura 3 ilustra a tela restrita para o administrador que possui cadastros básicos para gerenciar melhor seus produtos, funcionários, vendas, setores, cargos, marcas, categorias e diversas informações administrativas que são realizadas dentro da empresa.

Figura 3 – Tela Principal do Administrador



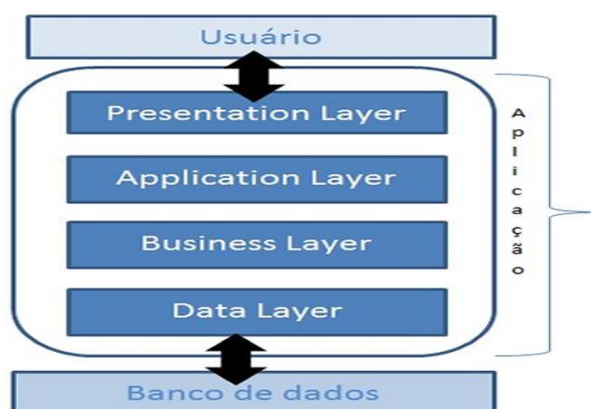
Fonte: Dados do Trabalho

4.2 ARQUITETURA

A arquitetura de software consiste na representação de componentes que relacionam e comunicam entre si. A documentação da arquitetura de software facilita a comunicação entre os *stakeholders*, registra relações entre decisões de projeto e possibilita a utilidade de reusar componentes e padrões de projetos.

Para o desenvolvimento desse trabalho foi utilizado uma arquitetura baseada em 4 camadas como será ilustrado na Figura 4.

Figura 4 – Modelo Arquitetura 4 Camadas



Fonte: Dados do Trabalho

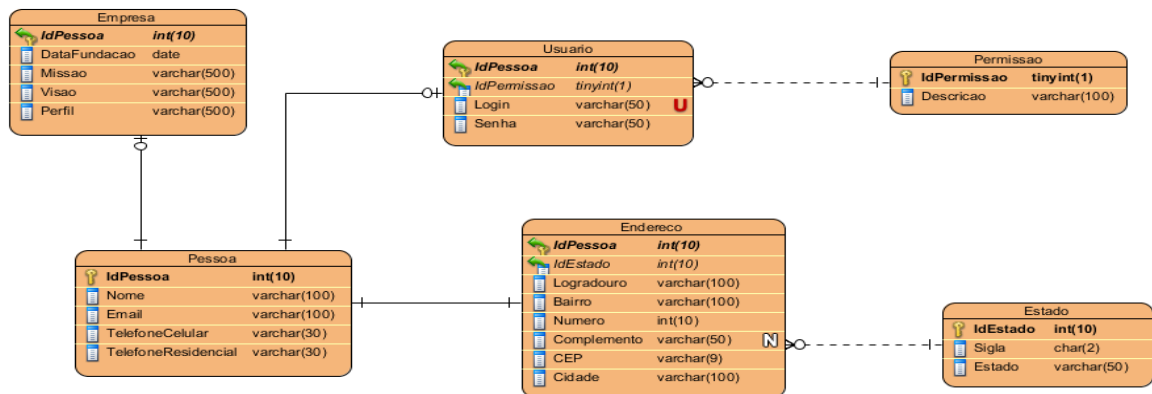
- *Presentation Layer*: camada responsável pela parte visual do software que interage com o usuário. Nessa camada podem-se encontrar arquivos *css*, *javascript*, imagens, páginas *web* e classes controladoras. Para esta camada foi utilizado o *framework JSF 2.0* baseado no padrão MVC (*Model-view-controller*) e também o *framework Primefaces* que fornece as interfaces de comunicação com o usuário;
- *Business Layer*: camada responsável pela lógica principal da aplicação, bem como, objetos definidos pelo modelo de dados. Essa camada permite implementar regras de negócio, tais como, validação de dados. Para esta camada foi utilizado o *framework* para mapeamento objeto relacional *JPA 2.0* com o *Hibernate*;
- *Data Layer*: camada responsável pela comunicação com o banco de dados. Para esta camada foi utilizado o *Hibernate*, ela tem a responsabilidade de consultar objetos mapeados na camada *Business Layer*;
- *Application Layer*: camada responsável por implementar regras de negócio que foram definidas no documento de requisitos. Nessa camada são criadas ações para manipular eventos, tais como, salvar, excluir, consultar, validações de dados, tratamento de erros e relatórios. Para esta camada foi utilizado o objeto *Criteria* do *Hibernate* para fazer consultas no banco de dados sem trabalhar diretamente com linguagem *SQL*.

4.3 BANCO DE DADOS

A Figura 5 apresenta o Diagrama de Entidade e Relacionamento que foi modelado para codificação do módulo Gerenciar Área Institucional da empresa. Com esse diagrama é possível identificar todos os atributos necessários para registrar as informações referentes à

Área Institucional. Foram utilizadas as tabelas “Empresa”, “Pessoa” e “Endereço” com seus respectivos atributos a serem persistidos no banco de dados. Para representar essa funcionalidade do sistema o objetivo maior é o cadastro da Empresa com seus dados institucionais que serão apresentados na área pública voltada para o cliente.

Figura 5 – Diagrama de Entidade e Relacionamento



Fonte: Dados do Trabalho

A Figura 6 apresenta a interface para o administrador realizar o cadastro da Área Institucional da Empresa.

Figura 6 – Tela de Cadastro da Área Institucional

Cadastrar Dados da Empresa

Informações Básicas da Empresa

Nome:* Carmotec Informática Email:* carmotecinfo@yahoo.com.br

Telefone Celular:* (34)9115-4793 Telefone Residencial:* (34)3851-1639

Data da Fundação:* 10/07/2005 Logradouro:* Rua São Vicente

Bairro:* Centro Numero:* 444

Cidade:* Carmo do Paranaíba Estado:* Minas Gerais

CEP:* 38840-000 Complemento:

Descrição da Missão

Missão:*

Tornar nossos cliente mais competitivos e melhores através de resultados de serviços de qualidade em Tecnologias da Informação.

Descrição da Visão

Descrição dos Valores

Salvar Informações

Fonte: Dados do Trabalho

A partir da interface ilustrada anteriormente o usuário fornece todos os dados que são obrigatórios no formulário. A tabela “Empresa” é uma especialização de “Pessoa”, onde possuem todos os dados que Pessoa tem, ou seja, ao informar os dados da Empresa, Pessoa e Endereço, o administrador tem a opção de gravar essas informações e caso seja necessário realizar qualquer tipo de alteração nestes dados, é possível acionar novamente o mesmo formulário e sobrescrever as informações.

O Quadro 1 apresenta o código fonte de um controlador responsável por salvar as informações da área institucional da empresa.

Quadro 1 - Código Fonte de um *Script* da Classe Controladora do Sistema

```
@ManagedBean
@RequestScoped
public class EmpresaController {

    private Empresa empresa;
    private Endereco endereco;

    private IEmpresaApplication empresaApplication = ApplicationFactory.getInstance().getEmpresaApplication();

    public void salvar() {

        Set<String> erros = empresaApplication.salvarEmpresa(empresa);
        System.out.println(erros);
        if (erros.isEmpty()) {

            FacesContext.getCurrentInstance().addMessage(
                null,
                new FacesMessage(FacesMessage.SEVERITY_INFO,
                    "Dados da Empresa Salvo com Sucesso.", null));

            novo();
        } else {
            FacesContext.getCurrentInstance().addMessage(
                null,
                new FacesMessage(FacesMessage.SEVERITY_ERROR, erros
                    .iterator().next(), null));
        }
    }

    public void novo() {
        this.setEmpresa(new Empresa());
        this.empresa.setEndereco(new Endereco());
        this.empresa.getEndereco().setEstado(new Estado());
    }

    public Empresa dadosDaEmpresa(){
        String emp = null;
        Empresa empresaMissao = empresaApplication.obter(1, emp);
        return empresaMissao;
    }

    public Empresa getEmpresa() {
        if (empresa == null) {
            List<Empresa> listaEmpresa;
            listaEmpresa = empresaApplication.todos("endereco", "endereco.estado");
            if (listaEmpresa.size() > 0) {
                empresa = listaEmpresa.get(0);
            }
            if (listaEmpresa.isEmpty()) {
                novo();
            }
        }
        return empresa;
    }
}
```

```

    public void setEmpresa(Empresa empresa) {
        this.empresa = empresa;
    }

    public Endereco getEndereco() {
        return endereco;
    }

    public void setEndereco(Endereco endereco) {
        this.endereco = endereco;
    }

    public IEmpresaApplication getEmpresaApplication() {
        return empresaApplication;
    }

    public void setEmpresaApplication(IEmpresaApplication empresaApplication) {
        this.empresaApplication = empresaApplication;
    }
}

```

Fonte: Dados do Trabalho

O Quadro 1 representa um controlador para a classe Empresa que é iniciada com a marcação “*@ManagedBean* e *@RequestScoped*”. A anotação “*@ManagedBean*” informa que a classe é um controlador e a anotação “*@RequestScoped*” que indica o escopo que será utilizado para criar as instâncias dos *Managed Beans* que são criados durante o processamento de uma requisição.

No método “*salvar()*”, é chamado o método “*salvarEmpresa()*”. Esse método é implementado na classe “*EmpresaApplication*” que implementa a interface “*IEmpresaApplication*”. Para salvar os dados da empresa, é passado por parâmetro um objeto do tipo Empresa no método. Caso não ocorra nenhum erro ao persistir as informações no banco de dados, é exibida uma mensagem de cadastro efetuado com sucesso.

Dentro do método “*getEmpresa()*”, foi criado um mecanismo de busca para listar as informações gravadas no banco de dados. Se o objeto “*empresa*” for igual a “*null*”, é criado uma lista chamada “*listaEmpresa*”, que recebe todos os registros da tabela Pessoa, porque a Empresa é uma especialização de Pessoa onde irá obter uma herança. Se a condição desta lista for atendida, ou seja, se o tamanho for maior que zero, quer dizer que existem registros nela. O objeto “*empresa*” recebe a “*listaEmpresa*” com o primeiro registro da tabela que fica na posição zero onde o cadastro da Empresa foi definido como o primeiro registro da tabela Pessoa.

Quando o administrador desejar alterar algum dado da Empresa, basta apenas acionar novamente o formulário para o cadastro da área institucional, se houver um registro referente à empresa na posição zero, ou seja, o objeto “*empresa*” irá receber a “*listaEmpresa*”

obtendo um valor na posição 0 pelo método “get”, caso a condição tenha sido atendida irá listar essas informações no formulário para que seja possível sobrescrevê-las.

4.4 SEGURANÇA

Através do *Login*, utilizando o mecanismo de segurança do *Spring Security* foi possível verificar se o usuário tem permissão para poder acessar as funcionalidades do sistema. No caso do cliente que é um usuário comum, existe apenas um papel que permite acessar o sistema. Como pode ser visto na Figura 7, são necessárias algumas configurações no arquivo de configuração do *Spring Security* que permite que o usuário faça seu *Login* e tenha acesso às informações do *site* para comprar produtos.

Figura 7 – Arquivo de Configuração do Spring Security

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/security"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:b="http://www.springframework.org/schema/beans"
4   xsi:schemaLocation="http://www.springframework.org/schema/beans
5   http://www.springframework.org/schema/beans/spring-beans.xsd http://www.springframework.org/schema/security
6   http://www.springframework.org/schema/security/spring-security-3.0.xsd">
7   <http auto-config="true" use-expressions="false">
8     <intercept-url pattern="/restrict/**" access="ROLE_ADM" />
9     <intercept-url pattern="/public/usuario/**" access="ROLE_USER,ROLE_ADM"/>
10    <form-login login-page="/login.faces" authentication-failure-url="/login.faces?erro=true"
11      default-target-url="/restrict/teste.faces"/>
12    <access-denied-handler error-page="/acessonegado.faces" />
13  </http>
14  <authentication-manager>
15    <authentication-provider>
16      <password-encoder hash="sha" />
17      <jdbc-user-service data-source-ref="dataSource">
18        users-by-username-query="SELECT login, senha, 1 FROM Usuario WHERE login=?"
19        authorities-by-username-query="SELECT usu.login as username, per.descricao as authority FROM
20          Usuario as usu inner join Permissao as per on usu.idPermissao = per.idPermissao WHERE usu.login=?" />
21      </authentication-provider>
22    </authentication-manager>
23  <b:bean id="dataSource" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
24    <b:property name="url" value="jdbc:mysql://localhost:3306/carmotec" />
25    <b:property name="driverClassName" value="com.mysql.jdbc.Driver" />
26    <b:property name="username" value="root" />
27    <b:property name="password" value="adm" />
28  </b:bean>
29 </beans>

```

Fonte: Dados do Trabalho

Neste arquivo de configuração foi definido dois tipos de permissões para que os usuários possam realizar *Login* no sistema. Um deles é o “*ROLE_USER*” que permite apenas os usuários terem acesso à área pública que é voltada para o cliente. Outra permissão que foi definida também é a “*ROLE_ADM*” que permite apenas os usuários terem acesso à

área restrita para o administrador. O arquivo também descreve algumas informações sobre o banco de dados e as consultas necessárias para obter os usuários com permissão de acesso e suas autorizações.

5 CONCLUSÃO

O desenvolvimento deste projeto teve como objetivo disponibilizar um sistema de vendas online para os clientes de modo a permitir a realização de compras por produtos. Um dos principais problemas enfrentados pela empresa era a falta de apoio para controlar a gestão de suas vendas. Devido às outras empresas também estarem investindo em seu próprio site para *e-commerce*, a empresa Carmotec Informática Ltda sentiu-se a necessidade de inovar seus negócios e aumentar sua produtividade.

O sistema atual da empresa não permitia registrar as vendas, o cliente que efetuou a compra, o cadastro de produtos e muito menos os cadastros de fornecedores. Além disso, a empresa não possuía relatórios para administrar as atividades realizadas pelo setor de vendas.

A empresa precisava de um sistema que conseguisse atender a todos esses requisitos, e a partir disso, obter qualidade para a divulgação de sua marca, bem como, produtos e serviços de forma clara e objetiva, aumentando a rapidez nos serviços e na divulgação de seus produtos, melhorando o acesso e o contato com o cliente.

Portanto, o sistema para gerenciar as vendas auxilia os funcionários que trabalham na empresa a gerenciar melhor suas atividades relacionadas ao comércio eletrônico. De acordo com os resultados obtidos pode-se concluir que houve um grande aumento na produtividade das vendas e gerou um lucro enorme para empresa.

Futuramente pretende-se desenvolver um módulo para a parte visual do sistema. Aplicar técnicas de design para melhorar o *Layout* da página *Web* e desenvolver novas soluções de melhoria para adaptar o modelo às novas tecnologias utilizadas no mercado e implantar um módulo para permitir que as entregas dos produtos sejam feitas através de outras cidades. Assim, o sistema poderá ser adicionado novos recursos à medida que tenha necessidade de buscar melhoria.

REFERÊNCIAS

ADONAI SILVEIRA CANÊZ (Brasil). **Processo Unificado (PU) – Unified Process**. Disponível em: <<http://www.adonai.eti.br/wordpress/2011/06/processo-unificado-pu-unified-process/>>. Acesso em: 28 fev. 2013.

BEZERRA, Eduardo. **Princípios de Análise e Projeto de Sistemas com UML**. Rio de Janeiro: Campus, 1995.

LARMAN, Craig; SALGADO, Luiz Augusto Meireles (Trad.). **Utilizando UML e padrões**. 2. ed. Porto Alegre: Bookman, 2004. 607 p.

MARCELINO, André et al. **MINHA LOJA VIRTUAL**: Sistema para vendas on-line. Disponível em: <http://www.projs.com.br/fmu/minhalojavirtual/Monografia_TCC_Final.pdf>. Acesso em: 25 fev. 2013.

MELO, Ana Maria. **Desenvolvendo Aplicações com UML 2.0**: do conceitual à implementação. 2. ed. Rio de Janeiro: Brasport, 2004.

PRESSMAN, Roger S. **Engenharia de software**. São Paulo: Pearson Makron Books, 2005. 1056 p.

SOMMERVILLE, Ian. **Engenharia de software**. 8. ed. São Paulo: Pearson Addison-Wesley, 2007. 552 p.