

**UNIVERSIDADE FEDERAL DE SANTA CATARINA**

**Sistema *on-line* de pesquisa e de compra do menor preço**

**Leonardo de Oliveira**

**Florianópolis – SC  
2010/2**

UNIVERSIDADE FEDERAL DE SANTA CATARINA  
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA  
CURSO DE SISTEMAS DE INFORMAÇÃO

Sistema *on-line* de pesquisa e de compra do menor preço

Leonardo de Oliveira

Trabalho de conclusão de curso  
apresentado como parte dos requisitos  
para obtenção do grau de Bacharel em  
Sistemas de Informação.

Florianópolis – SC  
2010/2

Leonardo de Oliveira

Sistema *on-line* de pesquisa e de compra do menor preço

Trabalho de conclusão de curso apresentado como parte dos requisitos para obtenção do grau de Bacharel em Sistemas de Informação.

---

Prof<sup>a</sup>. Patrícia Della Mea Plentz, Dra.  
Universidade Federal de Santa Catarina  
plentz@inf.ufsc.br  
Orientadora

---

Prof. Renato Fileto, Dr.  
Universidade Federal de Santa Catarina  
fileto@inf.ufsc.br  
Membro da banca examinadora

---

Prof. Leandro José Komosinski, Dr.  
Universidade Federal de Santa Catarina  
leandro@inf.ufsc.br  
Membro da banca examinadora

*Dedico este trabalho à minha família, a  
minha esposa Saionara e ao meu filho  
Mateus.*

## **AGRADECIMENTOS**

Gostaria de agradecer a todos aqueles que de alguma maneira me ajudaram a realizar este trabalho.

À minha família, que sempre deu apoio em todos os momentos e me deu condições de vencer mais uma etapa em minha vida.

À minha esposa Saionara e ao meu filho Mateus, que sempre estiveram do meu lado, incentivando e apoiando meus projetos pessoais e, sobretudo compartilhando excelentes momentos pelo qual já passamos.

À professora Patrícia Della Mea Plentz, que aceitou orientar meu trabalho de conclusão de curso, e me deu todo o suporte necessário para chegar até este ponto.

Aos demais professores, que durante todo o curso, me proporcionaram novos conhecimentos e ótimo aprendizado que levarei por toda a minha vida.

E finalmente ao Departamento de Informática e Estatística (INE) e a Universidade Federal de Santa Catarina (UFSC) por ter me proporcionado uma graduação de excelente qualidade.

## SUMÁRIO

1.	Introdução .....	12
1.1	Contextualização.....	12
1.2	Objetivo geral.....	13
1.3	Objetivos específicos .....	13
1.4	Motivação.....	13
1.5	Organização do trabalho.....	13
2.	Fundamentação Teórica .....	15
2.1	Programação .....	15
2.1.1	Introdução.....	15
2.1.2	Linguagens e tecnologias utilizadas .....	17
2.2	Integração de Bases de Dados.....	23
2.2.1	Introdução.....	23
2.2.2	Linguagens e tecnologias utilizadas .....	25
2.3	Comércio Eletrônico.....	29
2.3.1	Introdução.....	29
2.3.2	Estrutura .....	30
2.3.3	Classificação.....	32
2.4	Sistemas Automáticos de Identificação.....	34
2.4.1	Introdução.....	34
2.4.2	Sistemas de Identificação por código de barras .....	34
2.4.3	UPC e EAN.....	35
2.5	Trabalhos Relacionados .....	36
3.	Desenvolvimento.....	38
3.1	Análise e concepção.....	38
3.1.1	Visão geral do sistema.....	38
3.1.2	Levantamento de requisitos.....	39
3.2	Projeto.....	40
3.2.1	Arquitetura global do sistema.....	41
3.2.2	Sistema de <i>e-commerce</i> .....	43
3.2.3	Diagramas <i>UML</i> .....	45
3.2.4	Diagrama do banco de dados.....	52
3.3	Implementação .....	52
3.3.1	Criação da base de dados .....	53
3.3.2	Desenvolvimento do sistema .....	53
3.3.3	Funcionalidades e telas .....	53
3.4	Testes .....	58
3.4.1	Experimentos realizados.....	58
4.	Conclusão .....	61
4.1	Considerações finais.....	61
4.2	Trabalhos futuros .....	62
	REFERÊNCIAS BIBLIOGRÁFICAS .....	63
	APÊNDICES.....	65

## LISTA DE FIGURAS

Figura 1 – Fluxograma de uma aplicação rodando no Code Igniter.....	20
Figura 2 – Configuração de um sistema de banco de dados simplificado.....	24
Figura 3 – Modelo de comércio eletrônico .....	29
Figura 4 – Estrutura do comércio eletrônico.....	32
Figura 5 – Arquitetura global do sistema.....	42
Figura 6 - Tela da loja virtual do osCommerce.....	45
Figura 7 – Diagrama de caso de uso da visão geral do sistema.....	46
Figura 8 - Diagrama de caso de uso manter fornecedor (detalhado).....	47
Figura 9 – Diagrama de caso de uso manter usuários (detalhado).....	47
Figura 10 – Diagrama de caso de uso importar lista de produtos (detalhado).....	48
Figura 11 – Diagrama de atividades (macro) da funcionalidade Importar lista de produtos .....	48
Figura 12 – Diagrama de atividades (detalhado) da etapa Integração com os fornecedores dos produtos.....	49
Figura 13 – Diagrama de atividades (detalhado) da etapa Identificação dos similares e comparação de preço.....	50
Figura 14 – Diagrama de atividades (detalhado) da etapa Integração com o sistema de <i>e-commerce</i> .....	50
Figura 15 – Diagrama de classes do sistema.....	51
Figura 16 – Diagrama ER do banco de dados integrado do sistema .....	52
Figura 17 – Instruções de instalação do framework Code Igniter.....	53
Figura 18 – Tela de autenticação da <i>interface</i> administrativa do sistema .....	54
Figura 19 – Tela de gerenciamento de usuários (pesquisa, cadastro, edição e etc.)	55
Figura 20 – Tela de gerenciamento de fornecedores (pesquisa, cadastro, edição e etc.) .....	56
Figura 21 – Tela de cadastro de fornecedor com destaque para o campo <i>Web Service</i> .....	56
Figura 22 – Tela de execução do processo de importação dos produtos .....	57
Figura 23 – Tela de consulta de um determinado produto na loja virtual .....	58
Figura 24 – Gráfico com o comparativo de preços da lista com 63 produtos.....	59
Figura 25 - Gráfico com o comparativo de preços da lista com 123 produtos.....	60

## LISTA DE TABELAS

Tabela 1 - Quadro comparativo entre os sistemas de *e-commerce* pesquisados.....44

## LISTA DE ABREVIATURAS E SIGLAS

API - *Application Programming Interface* (Interface de Programação de Aplicativos)

EAN - *International Article Number* (Número de Artigo Internacional)

E-COMMERCE – *Electronic commerce* (Comércio Eletrônico)

ER - *Entity Relationship* (Entidade Relacionamento)

OCR - *Optical Character Recognition* (Reconhecimento Ótico de Caracteres)

PHP – *Hypertext Preprocessor* (Pré-Processador de Hipertexto)

SSL - *Secure Sockets Layer* (Camada de Sockets Segura)

UPC - *Universal Product Code* (Código Universal de Produtos)

UML - *Unified Modeling Language* (Linguagem Unificada de Modelagem)

WebApp - *Web Application* (Aplicação Web)

XML - *Extensible Markup Language* (Linguagem Estendida de Marcação)

XML-RPC - *Extensible Markup Language - Remote Procedure Call* (Chamada Remota de Procedimento utilizando XML)

## RESUMO

No comércio varejista estão presentes dois grandes atores: fornecedor e consumidor. O fornecedor está sempre à procura de atrair mais clientes, manter seus clientes e vender seus produtos. Já o consumidor, quando decide comprar um determinado produto, além de procurar um produto de qualidade e boa procedência, também procura por um produto com um bom preço.

Diante deste cenário, e com o objetivo de auxiliar os consumidores a encontrar os produtos desejados com o menor preço, o presente trabalho é uma proposta de desenvolvimento de um sistema que permita a pesquisa e compra do menor preço, focado no comércio varejista e utilizando como estudo de caso o segmento supermercadista.

Para isso são utilizadas algumas tecnologias, como por exemplo, *Web Service* e *XML* para fazer a integração das diversas fontes de dados envolvidas. Também são definidos uma arquitetura e os comportamentos do sistema, para que os produtos cadastrados em diferentes fontes de dados sejam mapeados de forma automática, seja identificado o menor preço e estes sejam apresentados ao usuário final (consumidor) através de uma loja virtual.

Palavras-chave: programação *Web*, integração de bases de dados, *e-commerce*, supermercado.

## **ABSTRACT**

*There are two great actors present in the retail trade: supplier and consumer. The supplier is always looking to attract more customers, keep these customers and sell their products. The consumer, on the other hand, when deciding to buy a particular product, looks for quality good provenance and low price.*

*In this setting, and aiming to help consumers find the products at the lowest price, this paper is a proposal to develop a system to search and purchase the best price, focused on retail sales and using as case study the supermarket segment.*

*Our proposal uses Web Services and XML to integrate the various data sources involved. The contributions include the architecture of developed system and the specification of the system's behavior. The proposed system automatically matches products, from different data sources, identifies the best price and presents the information to the end user (consumer) through a virtual store.*

*Keywords: Web programming, data integration, e-commerce, supermarket.*

# 1. Introdução

## 1.1 Contextualização

Os consumidores do setor varejista, quando pretendem fazer determinado tipo de compra, encontram nas lojas e no comércio em geral, uma diferença de preço para o produto que desejam adquirir.

O que geralmente acontece é que o consumidor se dirige até as lojas e faz uma pesquisa de preço para o produto desejado, com o objetivo de adquirir o mais barato e deste modo realizar uma economia, ou seja, poupar dinheiro.

Com as tecnologias que temos disponíveis atualmente, como a *Internet* e o *e-commerce*, o consumidor tem a facilidade de fazer essa comparação de preços em tempo real e até mesmo realizar a compra sem sair de casa.

No entanto, um dos problemas enfrentados, é que nem todos os comerciantes ou lojistas possuem um site na *Internet* e muito menos uma infraestrutura de *e-commerce*, e deste modo, o consumidor não pode fazer uma pesquisa de preços completa, de forma rápida e sem muitos transtornos.

O objetivo deste trabalho é desenvolver um sistema *on-line* de pesquisa e de compra do menor preço, que possa ser utilizado em determinado segmento do comércio varejista. Este sistema deve permitir que o consumidor faça a pesquisa de preços *on-line* e adquira o produto desejado pelo preço mais baixo, independentemente dos lojistas ou comerciantes deste segmento possuírem site ou infraestrutura de *e-commerce*. Para isso, estes fornecedores parceiros terão que disponibilizar na *Web*, um serviço que possibilite o acesso a lista de produtos que serão comercializados.

Para viabilizar este tipo de solução, será necessário: (i) estudar um meio de acessar as diversas fontes de dados onde estão localizados os produtos de determinado segmento do comércio varejista; (ii) mapear e identificar os produtos distintos através de um código universal ou técnicas de similaridade; (iii) elaborar um algoritmo de escolha do menor preço; e finalmente (iv) apresentar os produtos mais baratos para o consumidor através de um catálogo de produtos em uma loja virtual.

## 1.2 Objetivo geral

Desenvolver um sistema *on-line* para pesquisa e compra do menor preço focado no comércio de varejo, utilizando como estudo de caso o setor supermercadista.

## 1.3 Objetivos específicos

- Pesquisar e definir o sistema de *e-commerce* a ser utilizado;
- Definir como será o mapeamento e a identificação dos produtos nas diversas fontes de dados;
- Elaborar um algoritmo de escolha do menor preço;
- Definir como será a integração entre as fontes de dados e o sistema de *e-commerce*;
- Integrar ao sistema de *e-commerce* utilizado as fontes de dados dos produtos dos supermercados.

## 1.4 Motivação

Boa parte dos consumidores, quando decide comprar determinado produto, acaba fazendo uma pesquisa de preços, pois pretende pagar o menor valor pelo produto desejado e assim realizar uma economia.

A motivação deste trabalho é permitir que as pessoas tenham a possibilidade de fazer pesquisa de preços *on-line* e desta forma economizar tempo e dinheiro, porque além de comprar o produto mais barato, também não precisam ir aos estabelecimentos para verificar o preço dos produtos.

Existe também outro fator que está motivando a realização deste trabalho, que é a possibilidade de futuramente gerar um negócio a partir dos conhecimentos obtidos durante o TCC.

## 1.5 Organização do trabalho

O presente trabalho está estruturado em quatro capítulos, da seguinte maneira: o capítulo 1 faz uma contextualização do problema, descrevendo os objetivos e a motivação do trabalho. No segundo capítulo serão apresentados os fundamentos teóricos dos conceitos utilizados neste TCC. O terceiro

capítulo irá apresentar o desenvolvimento do trabalho, descrevendo as atividades realizadas para atingir os objetivos do TCC e o quarto capítulo irá descrever as conclusões e trabalhos futuros.

## 2. Fundamentação Teórica

Este capítulo descreve os principais conceitos e fundamentos relacionados aos temas: Programação, Integração de Bases de Dados e Comércio Eletrônico.

### 2.1 Programação

#### 2.1.1 Introdução

A *Internet*, sem dúvida alguma, foi uma das invenções tecnológicas mais importantes dos últimos tempos. A este respeito, Pressman (2006) afirma que “a *World Wide Web* e a *Internet*, que lhe dá poder, são indiscutivelmente os desenvolvimentos mais importantes da história da computação. Essas tecnologias nos colocaram (com bilhões mais que irão eventualmente nos seguir) na era da informação”.

Com o advento destas novas tecnologias, surgiram novas formas de comunicação e demandas de desenvolvimento de sistemas, que até então não existiam. Como a *Internet* tem características diferentes dos ambientes tradicionais de desenvolvimento de *software*, o processo de desenvolvimento de *software* para a *Internet*, é bastante semelhante ao processo de *software* tradicional, porém com algumas adaptações.

Pressman (2006) diz que para “alcançar maior sucesso no desenvolvimento e na aplicação de sistemas complexos e de grande escala baseados na *Web*, há uma necessidade premente de abordagens disciplinadas de engenharia da *Web*, e novos métodos e ferramentas para o desenvolvimento, a implantação e a avaliação de sistemas e aplicações baseadas na *Web*. Tais abordagens e técnicas devem levar em conta as características especiais do novo meio, os ambientes e cenários operacionais, e a multiplicidade de perfis de usuários, que adicionam desafios ao desenvolvimento de aplicações baseadas na *Web*”.

Segue abaixo uma lista de características de aplicações *Web*, que de acordo com Pressman (2006), são atributos encontrados na grande maioria das aplicações *Web*:

- Informacional – conteúdo somente de leitura é fornecido com navegação e ligações (*links*) simples.
- Para baixar – um usuário baixa informação (faz *download*) de um servidor adequado.
- Adaptável – o usuário adapta o conteúdo a necessidades específicas.
- Interação – a comunicação entre uma comunidade de usuários ocorre por intermédio de salas de bate-papo, quadros de aviso ou mensagens instantâneas.
- Entrada de usuário – entrada baseada em formulários é o principal mecanismo para comunicar a necessidade.
- Orientada a transação – o usuário faz uma solicitação (por exemplo, coloca um pedido) que é atendida pela *WebApp*.
- Orientada a serviços – a aplicação fornece um serviço ao usuário (por exemplo, ajuda o usuário a calcular um pagamento de hipoteca).
- Portal – a aplicação orienta o usuário para outros conteúdos ou serviços da *Web* fora do domínio de aplicação do portal.
- De acesso a banco de dados – o usuário consulta uma grande base de dados e extrai informação.
- Armazém de dados – o usuário consulta uma coleção de grandes bancos de dados e extrai informação.

Com relação a ferramentas e tecnologias, uma grande quantidade delas tem evoluído desde que as aplicações *Web* têm se tornado mais sofisticadas e difundidas. Pressman (2006) destaca que “essas tecnologias englobam uma grande variedade de descrição e conteúdo e linguagens de modelagem (por exemplo, *HTML*, *VRML*, *XML*), linguagens de programação (por exemplo, *Java*), recursos de desenvolvimento baseado em componentes (por exemplo, *CORBA*, *COM*, *ActiveX*, *.NET*), navegadores, ferramentas multimídia, ferramentas de autoria de *sites*, ferramentas de conectividade de banco de dados, ferramentas de segurança, servidores e utilitários de servidor, e ferramentas de gestão e análise de testes”.

## 2.1.2 Linguagens e tecnologias utilizadas

Para o desenvolvimento deste trabalho de conclusão de curso, serão utilizadas a linguagem de programação *Web PHP* e um *framework* de aplicações *Web* chamado *Code Igniter*.

### 2.1.2.1 Linguagem PHP

#### 2.1.2.1.1 Histórico

Em 1994, *Rasmus Lerdorf* criou a linguagem de programação PHP, que era formada na época por um conjunto de *scripts* para gerenciamento de páginas dinâmicas, que ele utilizava para monitorar o acesso ao seu currículo na *Internet*. Com o passar do tempo, as funcionalidades foram crescendo, e *Rasmus* teve que escrever uma implementação em C, a qual permitia que as pessoas desenvolvessem suas aplicações na *Web* de forma muito simples. Ele batizou essa versão de *PHP/FI* (*Personal Home Pages / Form Interpreter*) e em 1995 resolveu disponibilizar o código da linguagem na *Web*, para compartilhar com outras pessoas e também receber a ajuda na correção dos *bugs* existentes (DALL'OGGIO, 2007).

Em 1997, a segunda versão da linguagem foi lançada, e neste período, *PHP* já era utilizado por aproximadamente 50 mil domínios (1% da *Internet*). Neste mesmo ano, dois estudantes resolveram colaborar com *Rasmus*, para aprimorar a linguagem. Foram eles, *Andi Gutmans* e *Zeev Suraski*, que na época estavam utilizando o *PHP* num projeto acadêmico de comércio eletrônico. Reescreveram todo o código fonte com base no *PHP/FI 2*, dando início assim ao *PHP 3*, que foi disponibilizado oficialmente em 1998 (DALL'OGGIO, 2007).

As principais características do *PHP 3* eram a extensibilidade, a possibilidade de conexão com vários bancos de dados, novos protocolos, uma sintaxe mais consistente, suporte a orientação a objetos e uma nova *API*, que permitia a criação de novos módulos e que acabou atraindo novos desenvolvedores ao *PHP*. Nesta época o *PHP* já estava presente em cerca de 10% dos domínios na *Internet* foi também o período em que a sigla da

linguagem mudou para *PHP: Hypertext Preprocessor*, retratando assim a nova realidade de uma linguagem com propósitos mais amplos (DALL'OGGIO, 2007).

Ainda em 1998, após o lançamento do *PHP 3*, *Zeev* e *Andi* começaram a reescrever o núcleo do *PHP*, com o objetivo de melhorar sua performance e modularidade em aplicações complexas. Foi quando batizaram este núcleo de *Zend Engine* ou Mecanismo *Zend* (*Zeev* e *Andi*). Já em 2000, foi lançado oficialmente o *PHP 4*, que era baseado neste mecanismo e estava trazendo muitas melhorias e diversos recursos novos. Neste momento, além de ser utilizado por milhares de desenvolvedores em todo o mundo, o *PHP* já estava presente em cerca de 20% dos domínios na *Internet* (DALL'OGGIO, 2007).

E finalmente em 2004, após um longo período de desenvolvimento, foi lançado o *PHP 5*, com maior suporte a orientação a objetos, tal qual existe em linguagens como *C++* e *Java* (DALL'OGGIO, 2007).

#### **2.1.2.1.2 Vantagens**

A linguagem *PHP* possui alguns concorrentes diretos, como por exemplo: *Perl*, *Microsoft ASP.NET*, *Java Server Pages (JSP)* e *ColdFusion*.

Segundo Welling e Thomson (2005), em comparação a esses produtos, o *PHP* tem muitas vantagens, incluindo:

- Alto desempenho;
- *Interfaces* para muitos diferentes sistemas de bancos de dados;
- Bibliotecas integradas para muitas tarefas comuns na *Web*;
- Baixo custo;
- Facilidade de aprender a utilizar;
- Ótimo suporte orientado a objetos;
- Portabilidade;
- Disponibilidade de código-fonte;
- Disponibilidade de suporte.

#### **2.1.2.1.3 Funções de similaridade**

*PHP* possui algumas funções para calcular a similaridade entre strings. São elas: *LEVENSHTEIN* e *SIMILAR\_TEXT*.

A primeira delas, *LEVENSHTEIN*, calcula a distância *Levenshtein* entre dois *strings* informados. A distância *Levenshtein* é definida como o número mínimo de caracteres que você tem para substituir, inserir ou apagar para transformar um *string* dentro de outro. A complexidade do algoritmo é  $O(m*n)$ , onde  $n$  e  $m$  são o comprimento dos dois *strings* analisados (*PHP: Manual do PHP*, 2010).

A segunda função, *SIMILAR\_TEXT*, calcula a similaridade entre dois *strings* conforme descrito em Oliver [2003]. Note que esta implementação não usa uma pilha (*stack*) como no pseudocódigo de Oliver, mas chamadas recursivas as quais podem ou não tornar todo o processo mais rápido. Note também que a complexidade deste algoritmo é  $O(N^3)$  aonde  $N$  é o tamanho da maior *string* (*PHP: Manual do PHP*, 2010).

### **2.1.2.2 Code Igniter**

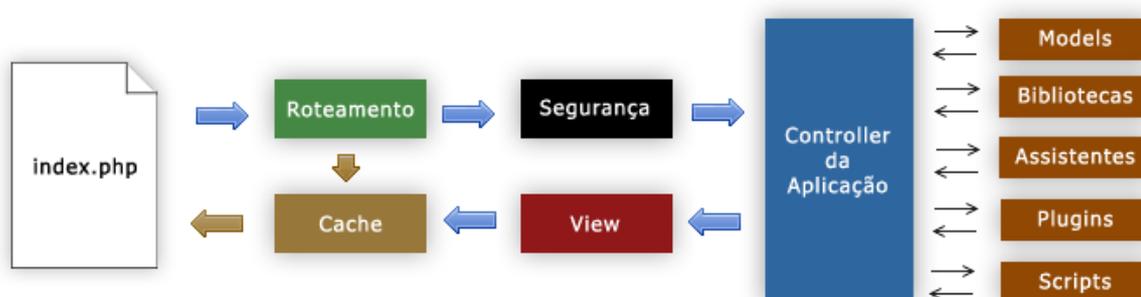
O *Code Igniter* é um *framework* para o desenvolvimento de aplicações *Web* em *PHP*. Ele possui uma grande variedade de bibliotecas voltadas para as tarefas mais comuns, uma *interface* e estrutura lógica simples e possibilita o desenvolvimento de projetos mais rapidamente do que se estivesse desenvolvendo do zero. Além disso, é distribuído de forma gratuita, e é disponibilizado sob uma licença de código aberto (*Code Igniter - Guia do Usuário*, 2010).

O *framework* utiliza uma abordagem *Model-View-Controller*, que permite uma boa separação entre a lógica a camada de apresentação. Dentre os recursos disponíveis, segue uma lista dos principais (*Code Igniter - Guia do Usuário*, 2010):

- Sistema baseado na abordagem *MVC (Model-View-Controller)*
- Compatível com *PHP4*
- Extremamente leve
- Suporte a várias plataformas de banco de dados
- Segurança e filtragem *XSS*
- Classes de envio de *e-mail*
- Bibliotecas de manipulação de imagens

- Suporte a localização e internacionalização
- *URLs* amigáveis aos motores de busca
- Bibliotecas *XML-RPC*

Para um melhor entendimento deste *framework*, vamos demonstrar como ocorre o fluxo de dados através do sistema, bem como suas interações. A figura 1 é um fluxograma da aplicação e na seqüência uma breve explicação de seu funcionamento (*Code Igniter - Guia do Usuário, 2010*).



**Figura 1 – Fluxograma de uma aplicação rodando no Code Igniter**

Fonte: *Code Igniter - Guia do Usuário, 2010*

- O *index.php* serve como um controlador primário, iniciando os recursos básicos necessários para rodar o *Code Igniter*.
- O roteador examina a requisição *HTTP* para determinar o que deve ser feito com ela.
- Se já existe o arquivo “cacheado”, ele é enviado diretamente ao *browser*, pulando as outras etapas de execução.
- Segurança. Antes do *controller* de aplicação ser carregado, a requisição *HTTP* e qualquer dado submetido pelo usuário é filtrado por segurança.
- O *Controller* carrega o *model*, as bibliotecas principais, *plugins*, assistentes e qualquer outro recurso necessário para processar a requisição específica.
- A *View* finalizada é renderizada e então enviada ao *browser* para ser vista. Se o *cache* está habilitado, a *view* é “cacheada” primeiro para que seja servida em requisições subseqüentes.

O framework *Code Igniter* possui uma série funcionalidades prontas e disponíveis para serem utilizadas na forma de classes e assistentes. As classes são funcionalidades a respeito de um determinado assunto, como por exemplo: banco de dados, envio de e-mail, FTP e etc. Os assistentes, como o nome sugere, auxiliam em determinadas tarefas. Cada arquivo assistente é simplesmente um conjunto de funções de uma categoria particular e diferentemente das classes e de todo o framework, eles não foram escritos no padrão orientado a objetos (*Code Igniter - Guia do Usuário*, 2010).

Segue abaixo a lista de classes atualmente disponíveis:

- Classe *Benchmarking*
- Classe Calendário
- Classe *Config*
- Classe Banco de dados
- Classe *E-mail*
- Classe Criptografia
- Classe *Upload* de Arquivo
- Classe *FTP*
- Classe Tabela *HTML*
- Classe Manipulação de Imagem
- Classe *Input* e Segurança
- Classe *Loader*
- Classe Idioma
- Classe *Output*
- Classe Paginação
- Classe Sessão
- Classe *Trackback*
- Classe *Template Parser*
- Classe *Unit Testing*
- Classe *URI*
- Classe *User Agent*

- Classe Validação
- Classe *XML-RPC*
- Classe *Zip Encoding*

Também podemos observar abaixo, uma lista com os assistentes atualmente disponíveis:

- Assistente de *Array*
- Assistente de *Cookie*
- Assistente de *Data*
- Assistente de *Diretório*
- Assistente de *Download*
- Assistente de *Arquivo*
- Assistente de *Formulário*
- Assistente *HTML*
- Assistente de *Flexão Gramatical*
- Assistente de *Segurança*
- Assistente de *Smileys*
- Assistente de *String*
- Assistente de *Texto*
- Assistente de *Tipografia*
- Assistente de *URL*
- Assistente de *XML*

Além destas classes e assistentes disponíveis no *framework*, é possível também instalar *plugins* desenvolvidos por terceiros ou até mesmo criar funcionalidades para as classes existentes ou implementar novas classes do “zero” (*Code Igniter - Guia do Usuário*, 2010).

## 2.2 Integração de Bases de Dados

### 2.2.1 Introdução

Os bancos de dados desempenham um papel muito importante no contexto de uso dos sistemas de informações. Eles estão presentes em várias situações que lidamos em nosso dia-a-dia. Por exemplo, quando vamos a um banco ou a um caixa eletrônico, quando compramos uma passagem aérea ou até mesmo quando realizamos nossas compras no supermercado. Mas o que é um banco de dados?

Segundo Elmasri & Navathe (2005), um banco de dados é uma coleção de dados relacionados que possuem um significado implícito e possuem as seguintes propriedades:

- Um banco de dados representa alguns aspectos do mundo real, sendo chamado, às vezes, de mini-mundo ou de universo de discurso (UoD). As mudanças no mini-mundo são refletidas nas informações contidas no banco de dados.
- Um banco de dados é uma coleção lógica e coerente de dados com algum significado inerente. Uma organização de dados ao acaso (randômica) não pode ser corretamente interpretada como um banco de dados.
- Um banco de dados é projetado, construído e povoado por dados, em atendimento a uma proposta específica. Possui um grupo de usuários definido e algumas aplicações preconcebidas, de acordo com o interesse desse grupo de usuários.

Para criar e gerenciar estas coleções de dados em um banco de dados é necessário a utilização de programas específicos denominados Sistemas Gerenciadores de Banco de Dados. Elmasri & Navathe (2005) descreve um Sistema Gerenciador de Banco de Dados (SGBD) como “uma coleção de programas que permite aos usuários criar e manter um banco de dados” e ainda complementa esta definição chamando “o banco de dados e o software SGBD, juntos, de sistema de banco de dados”.

De forma bastante semelhante, Silberschatz, Korth e Sudarshan (2006) definem que “um Sistema de Gerenciamento de Banco de Dados (*DBMS*) é uma coleção de dados inter-relacionados e um conjunto de programas para acessar esses dados” e afirma também que “o principal objetivo de um *DBMS* é fornecer uma maneira de recuperar informações de banco de dados que seja tanto conveniente quanto eficiente”.

A figura 2 ilustra alguns conceitos apresentados.

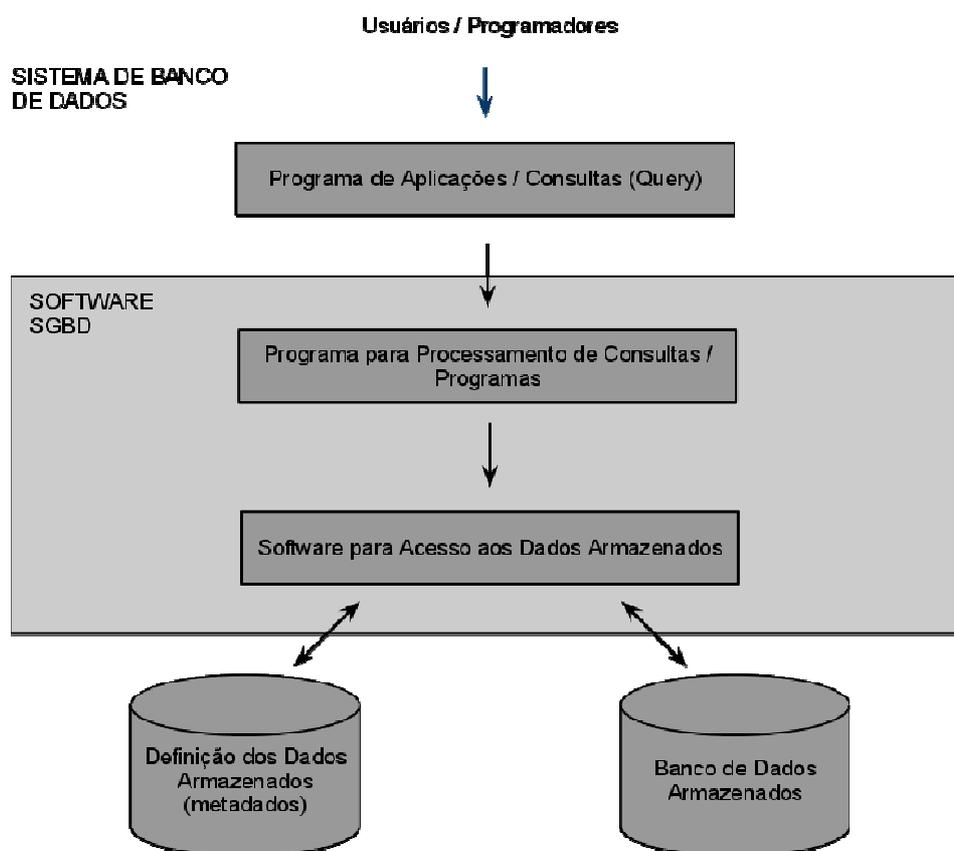


Figura 2 – Configuração de um sistema de banco de dados simplificado  
Fonte: ELMASRI & NAVATHE, 2005

Durante o uso e a manutenção de um banco de dados, várias pessoas estão envolvidas. Elmasri & Navathe (2005) define e classifica estas pessoas como:

- **Administrador de Banco de Dados (DBA):** é o responsável pela autorização para acesso ao banco, pela coordenação e monitoração de

seu uso e por adquirir recursos de *software* e *hardware* conforme necessário.

- **Projetista do Banco de Dados:** é o responsável pela identificação dos dados que serão armazenados no banco e também por escolher as estruturas apropriadas para representar e armazenar esses dados. É também sua responsabilidade comunicar-se antecipadamente com todos os prováveis usuários do banco para conhecer suas necessidades (requisitos) e criar projetos que as atendam.
- **Usuário Final:** são pessoas cujas profissões requerem o acesso a um banco de dados para consultas, atualização e relatórios. O banco de dados existe prioritariamente para este tipo de usuário.
- **Analista de Sistemas e Programador de Aplicações:** os analistas de sistemas determinam as solicitações dos usuários finais, além de desenvolver as especificações das transações customizadas que atendam a essas solicitações. Já os programadores de aplicações implementam essas especificações como programas, testam, documentam e mantêm transações customizadas.

## 2.2.2 Linguagens e tecnologias utilizadas

Para dar o devido suporte a integração de dados, serão utilizadas as seguintes linguagens e tecnologias: *XML* e *Web Service*.

### 2.2.2.1 XML

#### 2.2.2.1.1 Definição

*XML (Extensible Markup Language)* é uma linguagem de marcação de documento que serve para representar vários tipos de dados estruturados de forma personalizada. De acordo com Liberty e Kralej (2001), “o *XML* oferece um mecanismo independente de plataforma aceito globalmente para gerenciar, armazenar e comunicar informações”.

*XML* é de fato um padrão já estabelecido pelas empresas no que diz respeito ao intercâmbio de dados na *Web*, prova disto é a afirmação de Liberty

e Kralej (2001) que diz o seguinte: “a recomendação *XML 1.0* do *World Wide Web Consortium (W3C)* é um padrão internacional que já foi endossado e adotado pela *Microsoft, Netscape, Sun, IBM* e vários outros fabricantes”.

#### **2.2.2.1.2 Linguagens de marcação**

Para entender melhor o *XML*, é preciso saber um pouco de sua história. Tudo começou com o *SGML* que é a sigla para *Standard Generalized Markup Language* (Linguagem de Marcação Generalizada Padrão), uma metalinguagem que tem a função de descrever como uma linguagem de marcação pode ser construída. Uma das linguagens que o *SGML* descreve, e também a mais conhecida, é o *HTML*.

Já o *XML* é um subconjunto do *SGML*. Trata-se de uma simplificação do *SGML*, que serve para o mesmo propósito, com a diferença de que o *XML* é muito mais fácil de ser utilizado.

*XML* também é extensível, ou seja, suas *tags* podem ser criadas de acordo com a necessidade, diferentemente do *HTML* que possui um conjunto de *tags* fixas. De acordo com Silberschatz, Korth e Sudarshan (2006), “ao contrário da *HTML*, *XML* não prescreve o conjunto de *tags* permitidas, e o conjunto pode ser escolhido conforme a necessidade por cada aplicação. Esse recurso é a chave para a principal função do *XML* na representação e troca de dados, enquanto *HTML* é usada principalmente para a formatação de documentos”.

#### **2.2.2.1.3 Vantagens**

Uma representação *XML*, quando utilizada para a troca de dados e também para armazenar informações estruturadas, segundo Silberschatz, Korth e Sudarshan (2006), possui vantagens significativas, como podemos ver abaixo:

- Primeiro, a presença das *tags* torna a mensagem auto-documentável, fazendo com que um esquema não precise ser consultado para se entender o significado do texto;
- Segundo, o formato do documento não é rígido, ou seja, *XML* tem a capacidade de reconhecer e ignorar *tags* inesperadas, permitindo que o

formato dos dados evolua com o passar do tempo, sem invalidar as aplicações existentes.

- Terceiro, permite estruturas aninhadas, e isso faz com que os dados que estão sendo representados no documento *XML*, sejam especificados de acordo com sua hierarquia (pais e filhos).
- Finalmente, como o formato *XML* é bastante aceito, existem várias ferramentas disponíveis para auxiliar no seu processamento, além de *APIs* nas linguagens de programação para criar e ler *XML*, *software* de navegador e ferramentas de banco de dados.

### **2.2.2.2 Web Service**

#### **2.2.2.2.1 Visão geral**

*Web Service* é uma solução bastante utilizada pelas organizações para comunicação e integração de dados entre aplicações distintas. Segundo *Wikipédia* (2010), “com esta tecnologia é possível que novas aplicações possam interagir com aquelas que já existem e que sistemas desenvolvidos em plataformas diferentes sejam compatíveis”.

Uma das grandes vantagens do *Web Service* é permitir a comunicação entre aplicações que foram desenvolvidas com linguagens diferentes. O uso de *XML*, que é uma linguagem universal para intercâmbio de dados, possibilita toda esta flexibilidade e portabilidade.

#### **2.2.2.2.2 Tecnologias utilizadas**

*Web Services* são suportados por uma série de tecnologias e protocolos, dentre eles: *XML*, *SOAP*, *WSDL* e *UDDI*. De acordo com *Wikipédia* (2010), “para a representação e estruturação dos dados nas mensagens recebidas/enviadas é utilizado o *XML* (eXtensible Markup Language). As chamadas às operações, incluindo os parâmetros de entrada/saída, são codificadas no protocolo *SOAP* (*Simple Object Access Protocol*, baseado em *XML*). Os serviços (operações, mensagens, parâmetros, etc.) são descritos usando a linguagem *WSDL* (*Web Services Description Language*). O processo

de publicação/pesquisa/descoberta de *Web Services* utiliza o protocolo *UDDI* (*Universal Description, Discovery and Integration*)”.

*XML* é a base dos *Web Services*. Ele fornece a descrição, o armazenamento e o formato para envio e recebimento de dados. A sintaxe de *XML* especifica como os dados são representados genericamente, pois permite que estes sejam estruturados de forma hierárquica e também de forma semelhante às estruturas que estão sendo manipuladas (*WIKIPEDIA*, 2010).

O *SOAP* (*Simple Object Access Protocol*) define um padrão para a transmissão de mensagens, usando *XML* para representar a entrada e saída do procedimento. *SOAP* ainda define um esquema *XML* padrão para representar várias informações relacionadas ao intercâmbio de dados, como por exemplo, o nome do procedimento e também os indicadores de status do resultado (falha/erro) (*SILBERSCHATZ, KORTH e SUDARSHAN*, 2006).

O *WSDL* (*Web Services Description Language*) é uma linguagem utilizada para descrever *Web Services*. Ela oferece algumas facilidades, como por exemplo, quais funções estão disponíveis e os seus tipos de entrada e saída. Além disso, o *WSDL* permite que sejam especificados a *URL* e a porta de rede a serem utilizadas para invocar o *Web Service* (*SILBERSCHATZ, KORTH e SUDARSHAN*, 2006).

O *UDDI* (*Universal Description, Discovery, and Integration*) define como um catálogo de *Web Services* disponíveis pode ser criado e como um determinado programa pode pesquisar no diretório para encontrar o *Web Service* desejado (*SILBERSCHATZ, KORTH e SUDARSHAN*, 2006). De acordo com *Wikipédia* (2010), um registro *UDDI* contém três tipos de informação:

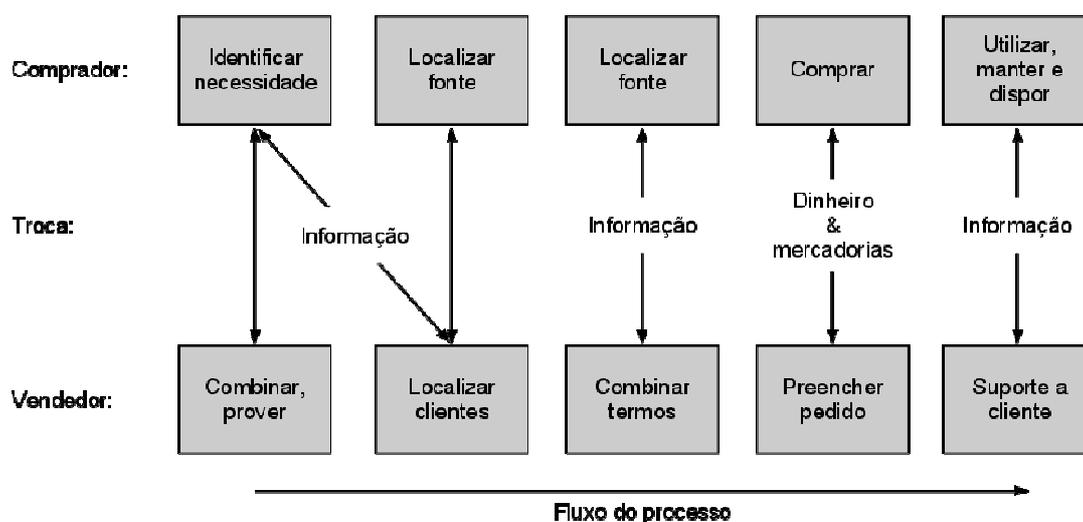
- Informações gerais de cada organização, tais como o nome, endereço, telefone e contatos;
- Informações de organizações e serviços por categorias de negócios;
- Informações técnicas sobre os serviços oferecidos pelas organizações.

## 2.3 Comércio Eletrônico

### 2.3.1 Introdução

O comércio eletrônico, também conhecido pelo termo *e-commerce*, nada mais é do que o processo de compra, venda e troca de produtos, serviços e informações através de redes de computadores ou pela *Internet* (TURBAN e KING, 2004). Com um ponto de vista bastante semelhante, Albertin (2004) cita que “o comércio eletrônico (CE) é a realização de toda a cadeia de valor dos processos de negócio num ambiente eletrônico, por meio da aplicação intensa das tecnologias de comunicação e de informação, atendendo aos objetivos de negócio”.

A figura 3 apresenta um modelo de comércio eletrônico e demonstra como ele pode ser utilizado em todas as fases de uma transação comercial (ALBERTIN, 2004).



**Figura 3 – Modelo de comércio eletrônico**  
Fonte: ALBERTIN, 2004

Os sistemas que suportam o comércio eletrônico têm valor significativo no sentido de impulsionar novas estratégias de gerenciamento de clientes, principalmente porque (ALBERTIN, 2004):

- Conectam diretamente compradores e vendedores;
- Apóiam troca de informações totalmente digitadas entre eles;
- Eliminam os limites de tempo e lugar;

- Apóiam interatividade, podendo adaptar-se dinamicamente ao comportamento do cliente; e
- Podem ser atualizados em tempo real, mantendo-se sempre atualizados.

Tecnologias de comércio eletrônico não estão restritas à *Internet*. Turban e King (2004) dizem que “o CE é normalmente utilizado na *Internet*. Mas também pode ser feito por meio de redes particulares, como as chamadas *value-added-networks* – *VANs* (redes que adicionam serviços de comunicação às redes transmissoras comuns preexistentes), de redes-locais (*LANs*) ou mesmo em uma máquina computadorizada”. Por exemplo: comprar produtos em uma máquina de refrigerantes e pagar com cartão eletrônico ou por meio de celular, pode ser considerada uma atividade de comércio eletrônico.

### **2.3.2 Estrutura**

Existe uma série de aplicações para o comércio eletrônico, dentre elas podemos citar:

- *Marketing* e propaganda *on-line*
- Vídeos sob demanda
- Bancos *on-line*
- Leilões
- Compra
- Publicações *on-line*
- Serviços ao consumidor

Para que essas aplicações possam ser executadas, as empresas necessitam de informações, infra-estrutura e serviços de apoio adequado. As áreas de suporte (pilares) que as aplicações de comércio eletrônico precisam contar são detalhadas abaixo (TURBAN e KING, 2004):

- Pessoas: vendedores, compradores, intermediários, funcionários e outros participantes;

- Política pública: aspectos legais e políticos – como proteção à privacidade – determinados pelo governo;
- Protocolos e padrões técnicos: protocolos de pagamento e de segurança, como *SSL (secure socket layer)*. Para se comunicar com as outras empresas ou mesmo para movimentar dinheiro *on-line* com segurança, as organizações necessitam de padrões e protocolos;
- Parceiros de negócios: *joint ventures*, permutas e parcerias comerciais de diversos tipos. O *e-commerce* muitas vezes é realizado dentro da cadeia de suprimentos (interações entre uma empresa e seus fornecedores, clientes e demais parceiros) e entre parceiros de negócios.
- Serviços de apoio: pesquisa de mercado, propaganda, pagamentos, logística e segurança. Diversos serviços de apoio são necessários em CE. E todos os componentes dessa estrutura exigem bom desempenho gerencial. Isto significa que as empresas devem planejar, organizar, motivar, desenvolver estratégias e refazer processos de acordo com as necessidades.

Já a estrutura do comércio eletrônico pode ser melhor compreendida e visualizada na figura 4.

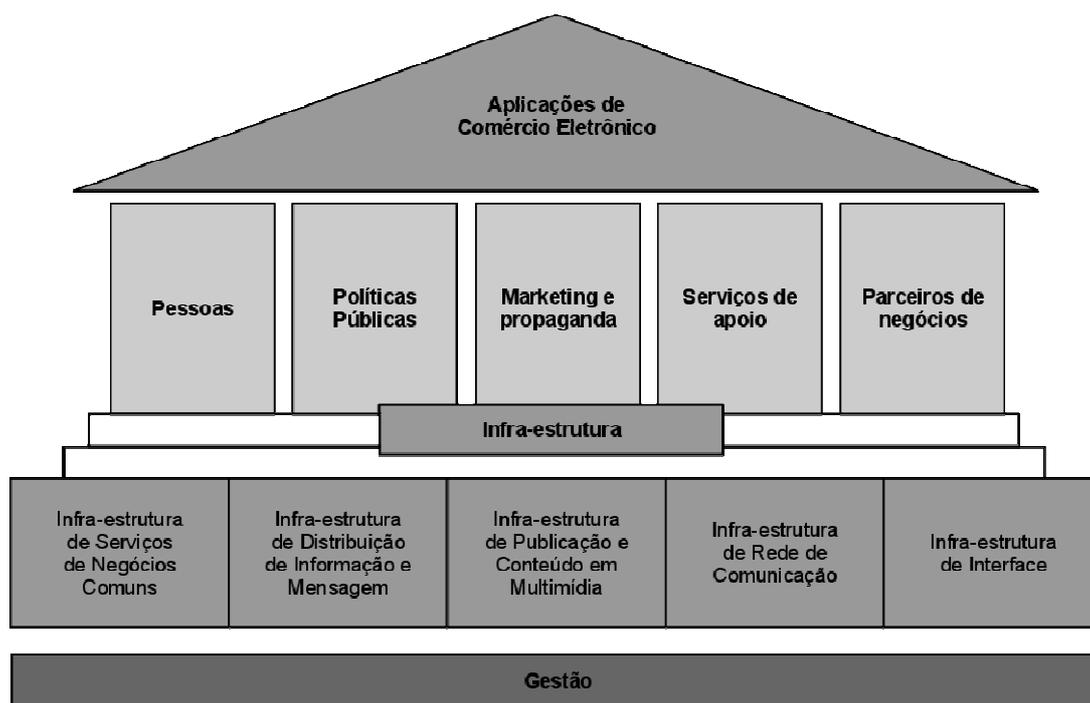


Figura 4 – Estrutura do comércio eletrônico  
 Fonte: Adaptada de TURBAN e KING, 2004 e ALBERTIN, 2004

### 2.3.3 Classificação

A classificação do comércio eletrônico é feita pela natureza da transação ou pelo relacionamento entre os participantes. A seguir, podemos observar os tipos encontrados com maior frequência (TURBAN e KING, 2004):

- **Business-to-business (B2B):** todos os participantes do *e-commerce business-to-business (B2B)* (empresa-empresa) são empresas ou outros tipos de organização.
- **Business-to-consumer (B2C):** o CE *business-to-consumer (B2C)* (empresa-consumidor) envolve transações de varejo entre empresas e compradores individuais. O comprador da *Amazon.com*, por exemplo, é um consumidor ou cliente. Esse tipo de CE também é chamado de varejo eletrônico (*e-tailing*).
- **Business-to-business-to-consumer (B2B2C):** no CE *business-to-business-to-consumer (B2B2C)* (empresa-empresa-consumidor) uma empresa oferece produtos ou serviços a uma empresa que seja cliente,

a qual, por sua vez, mantém seus próprios clientes, para os quais o produto ou serviço é repassado. O termo *B2B* também é freqüentemente utilizado para descrever o *B2B2C*.

- **Consumer-to-business (C2B) (consumidor-empresa):** essa categoria envolve, por um lado, indivíduos que utilizam a *Internet* para vender produtos ou serviços a organizações e, por outro, aqueles que procuram vendedores a fim de que ofereçam lances, para obter produtos ou serviços de que necessitam.
- **Consumer-to-consumer (C2C):** na categoria *consumer-to-consumer (C2C)* (consumidor-consumidor) os consumidores vendem diretamente uns aos outros. Alguns exemplos são os indivíduos que vendem imóveis, carros e outros produtos nos classificados *on-line*; anúncio de serviços pessoais pela *Internet*; e a venda de conhecimentos especializados *on-line*.
- **Mobile commerce:** as atividades e transações comerciais realizadas em ambiente sem fio são chamadas de comércio móvel (*mobile commerce*) ou *m-commerce*.
- **Comércio colaborativo:** quando indivíduos ou grupos se comunicam ou colaboram uns com os outros *on-line*, estão praticando comércio colaborativo (*c-commerce*). Por exemplo: parceiros de negócios que estejam em locais diferentes, podem juntos desenvolver um produto por meio de telas compartilhadas, ou criar previsão de demanda de mercado.
- **CE nobusiness:** um número cada vez maior de instituições não-comerciais – como as acadêmicas, aquelas sem fins lucrativos, as religiosas, as sociais e as agências do governo – utiliza o CE para reduzir despesas ou aprimorar suas operações e serviços.
- **E-government - government-to-citizen (G2C) (governo-cidadão) e outros:** no CE *e-government*, uma entidade governamental adquire produtos, serviços ou informação de empresas ou de cidadãos, ou, ainda, oferece esses bens a tais empresas e cidadãos.

- **Exchange-to-exchange (E2E):** um espaço de troca é um mercado eletrônico público com diversos compradores e vendedores. À medida que o número desses participantes aumenta, ocorrem mais trocas. O *CE exchange-to-exchange (E2E)* é um sistema formal que estabelece a conexão de trocas.

## 2.4 Sistemas Automáticos de Identificação

### 2.4.1 Introdução

Os sistemas de identificação são aplicados às pessoas e objetos através de registros magnéticos, ópticos, sonoros ou impressos. Geralmente são compostos por dois elementos: o elemento codificado (portador da informação) e o elemento leitor (aquele que é capaz de reconhecer a informação). Os casos de identificação de pessoas são, por exemplo: acesso à conta do banco, a uma área restrita, a um computador etc. Já nos casos de identificação de objetos, a utilização mais comum está relacionada a atividades comerciais, como por exemplo, para armazenar informações de produtos (ERDEI, 1994).

Existem diversos sistemas de identificação, dentre eles podemos citar (ERDEI, 1994):

- Visão eletrônica
- Faixas magnéticas
- Reconhecimento óptico de caracteres (*OCR*)
- Reconhecimento de voz humana
- Identificação biométrica
- Códigos de barras

### 2.4.2 Sistemas de Identificação por código de barras

De acordo com Erdei (1994), código de barras “constitui sem dúvida o sistema de identificação mais difundido e mais utilizado mundialmente, particularmente para o processamento automático comercial de produtos de consumo em massa. Zyngier e Grossmann (1991) também afirmam que “o

código de barras pode ser usado para aprimorar qualquer processo que envolva controle de mercadorias. Por suas características, o sistema é ideal para operações com um grande número de itens”.

Existem vários sistemas de códigos de barras, do quais podemos citar (ZYNGIER e GROSSMANN, 1991):

- Código 39: encontra sua maior aplicação na indústria mecânica, pois, além de ser alfanumérico, se adapta bem à baixa qualidade de impressão, muito comum aos materiais encontrados nesse meio.
- Código *ITF*: também um código de baixa resolução, é muito utilizado na codificação de embalagens de embarque.
- Códigos *UPC* e *EAN*: têm a mesma estrutura de codificação e são compatíveis entre si, apesar de serem utilizados em diferentes regiões do mundo, principalmente no comércio de mercadorias. A boa resolução da impressão das embalagens permite uma maior densidade dos códigos, ocupando, assim, superfícies relativamente pequenas e comprometendo menos o aspecto visual do produto.

### **2.4.3 *UPC* e *EAN***

Os códigos *UPC* e *EAN* são os tipos de sistema de identificação mais utilizados em supermercados.

De acordo com Erdei (1994), “o código *UPC* (*Universal Product Code*), é um dos mais populares, tendo sido criado e adotado pela indústria americana em 1973 para ser lido em caixas registradoras dos supermercados (pontos de venda)”.

O *EAN* por sua vez, é uma codificação para uso comercial, em pontos de venda no mundo inteiro (exceto Estados Unidos e Canadá, que utilizam o sistema *UPC*) (ERDEI, 1994).

Conforme Zyngier e Grossmann (1991), “o código *EAN* se apresenta em duas versões: uma com treze algarismos, chamada *EAN-13*; e outra, conhecida por *EAN-8*, com oito algarismos. No caso geral, deve ser usado o *EAN-13*. A utilização o *EAN-8* é restrita às embalagens que dispõem de espaço muito reduzido”.

## 2.5 Trabalhos Relacionados

No Brasil, existe uma série de portais de comparação de preços, como por exemplo: Buscapé, Bondfaro, Jacotei e ShoppingUOL, que tem o propósito de oferecer aos consumidores uma maneira de comparar e encontrar os locais onde os produtos tem o menor preço (REVISTA VEJA, 2010).

Nestes portais, os consumidores podem comparar o preço de determinado produto em diversas lojas. Ao decidir pela compra de um dos produtos pesquisados no portal, o consumidor se dirige até a loja (física ou virtual) que está ofertando o produto e realiza a compra.

Existe também, uma tese de doutorado que propõe um modelo um pouco semelhante ao trabalho de conclusão de curso aqui proposto. Trata-se do trabalho de autoria de Jordan Paulesky Juliani, cujo título é “A SOCIALIZAÇÃO DE CONHECIMENTO ENTRE CONSUMIDORES NA BUSCA DE MELHORES ALTERNATIVAS DE COMPRA. UM MODELO TECNOLÓGICO” (JULIANI, 2008).

Esta tese propõe um modelo em que consumidores, fornecedores e institutos de pesquisa de preços, possam alimentar as informações no sistema, de acordo com seus interesses. Os consumidores informariam os dados obtidos por eles durante o processo de compra, mesmo que não tenham concretizado a compra. Os fornecedores poderiam cadastrar seus produtos e oportunidades de compra. E os institutos de pesquisa de preço, informariam as pesquisas de opiniões e levantamentos efetuados nos estabelecimentos pesquisados (JULIANI, 2008).

Uma das principais diferenças entre os modelos apresentados anteriormente e o trabalho proposto neste TCC, é que o sistema aqui proposto é focado em um determinado segmento do comércio varejista, como por exemplo, os supermercados. Isto permite que os consumidores possam realizar a compra de um conjunto de produtos de uma única vez, e pagar o menor preço de cada produto que foi adquirido.

O presente trabalho também pode ser aplicado em outros segmentos, como por exemplo:

- Construção civil – para pesquisa de preços de materiais de construção em lojas do ramo
- Saúde – para comparação de preços de remédios e medicamentos em farmácias
- Educação – para pesquisa e comparação de preços de material escolar em papelarias

Outro diferencial encontrado, é que as compras são feitas diretamente através da uma loja virtual, e a entrega é feita de forma unificada, apesar dos produtos que compõem a compra se originarem de várias fontes diferentes.

### **3. Desenvolvimento**

#### **3.1 Análise e concepção**

Para iniciarmos o desenvolvimento do sistema proposto, levantamos algumas informações a respeito da aplicação. Segundo Wazlawick (2004), “a fase de concepção consiste em uma etapa na qual o analista vai buscar as primeiras informações sobre o sistema a ser desenvolvido”. Wazlawick (2004) ainda cita que “os artefatos dessa fase são ainda desestruturados, isto é, não são necessariamente completos e organizados. O objetivo é descobrir se vale a pena fazer a análise, mas sem fazer a análise propriamente dita”.

Teremos então na fase de análise e concepção, as seguintes etapas: visão geral do sistema e o levantamento de requisitos.

##### **3.1.1 Visão geral do sistema**

A visão geral do sistema, nada mais é do que uma descrição daquilo que se pretende ao desenvolver determinado sistema. Para Wazlawick (2004), “a visão geral do sistema, ou sumário executivo, que é um texto corrido, sem necessidade de nenhuma estrutura especial. Ele deve descrever as principais idéias do cliente sobre o sistema”.

A proposta deste TCC é desenvolver um sistema que permita aos consumidores realizar a compra dos produtos mais baratos nos supermercados de sua região, sem precisar se deslocar até os estabelecimentos para fazer a pesquisa de preços manualmente.

Para que isso seja possível, será necessário automatizar toda esta parte de pesquisa e comparação de preços dos supermercados e deixar disponível para o consumidor, apenas os produtos com o preço menor.

O sistema deve então se conectar com alguns supermercados pré-definidos, para obter a lista de produtos disponíveis para serem comercializados. De posse destas listas de produtos, é preciso comparar o preço de cada produto equivalente e determinar quais são aqueles que possuem o menor preço em cada estabelecimento. Após este levantamento,

esses produtos serão colocados a venda em um sistema de comércio eletrônico (supermercado virtual), para ficar a disposição dos consumidores.

A partir deste momento, os consumidores acessam o supermercado virtual e fazem suas compras. Cada pedido feito pelos clientes pode conter produtos de vários supermercados. Então, o sistema processa os pedidos individualmente, e se comunica com os supermercados para informar quais produtos devem ser comprados.

Finalmente, no momento da entrega, um serviço terceirizado de transporte passa em cada um dos supermercados, coleta os produtos de cada cliente e realiza a entrega dos mesmos no endereço informado durante a compra.

### **3.1.2 Levantamento de requisitos**

Para a construção do sistema, foi necessário fazer um levantamento de requisitos, ou seja, mapear e documentar as necessidades para a concretização do sistema. Segundo Wazlawick (2004), “a análise de requisitos está associada ao processo de descobrir quais são as operações que o sistema deve realizar e quais são as restrições que existem sobre elas”.

Os requisitos aqui levantados tiveram sua origem de idéias minhas e conversas com outros colegas da faculdade, ou seja, diferentemente do processo convencional, os requisitos do sistema aqui desenvolvido não foram levantados com um cliente em específico.

Para uma melhor compreensão, vamos dividir os requisitos levantados em duas grandes categorias: os requisitos funcionais e os não funcionais.

Os requisitos funcionais correspondem à listagem de tudo que o sistema deve fazer. Já os requisitos não funcionais são restrições colocadas sobre como o sistema deve realizar seu requisitos funcionais (WAZLAWICK 2004).

#### **3.1.2.1 Requisitos funcionais**

- RF001 – Os consumidores poderão fazer compras através de um supermercado virtual
- RF002 – O sistema deve se conectar com os supermercados para obter a lista de produtos

- RF003 – O sistema deve comparar a lista de produtos dos fornecedores e identificar os produtos iguais
- RF004 – O sistema deve eleger os produtos mais baratos para disponibilizar no supermercado virtual
- RF005 – O sistema deve importar os produtos eleitos mais baratos para o supermercado virtual
- RF006 – O sistema deve fazer a comparação de preços dos produtos dos fornecedores
- RF007 – O sistema deve permitir o cadastro de fornecedores
- RF008 – O sistema deve permitir o cadastro de usuários
- RF009 – O sistema deve se comunicar com os supermercados para informar quais produtos foram comprados pelos consumidores

### **3.1.2.2 Requisitos não funcionais**

- RNF001 - O sistema proposto deve ser desenvolvido em um ambiente *Web*
- RNF002 – A linguagem de programação a ser utilizada será o *PHP*
- RNF003 – O banco de dados a ser utilizado será o *MySQL*
- RNF004 - O sistema deve funcionar nas versões mais recentes dos navegadores conhecidos no mercado: *Firefox, Internet Explorer, Chrome* e etc.
- RNF005 - Somente usuários cadastrados poderão utilizar a *interface* administrativa do sistema
- RNF006 – As *interfaces* devem ser simples, claras e objetivas

## **3.2 Projeto**

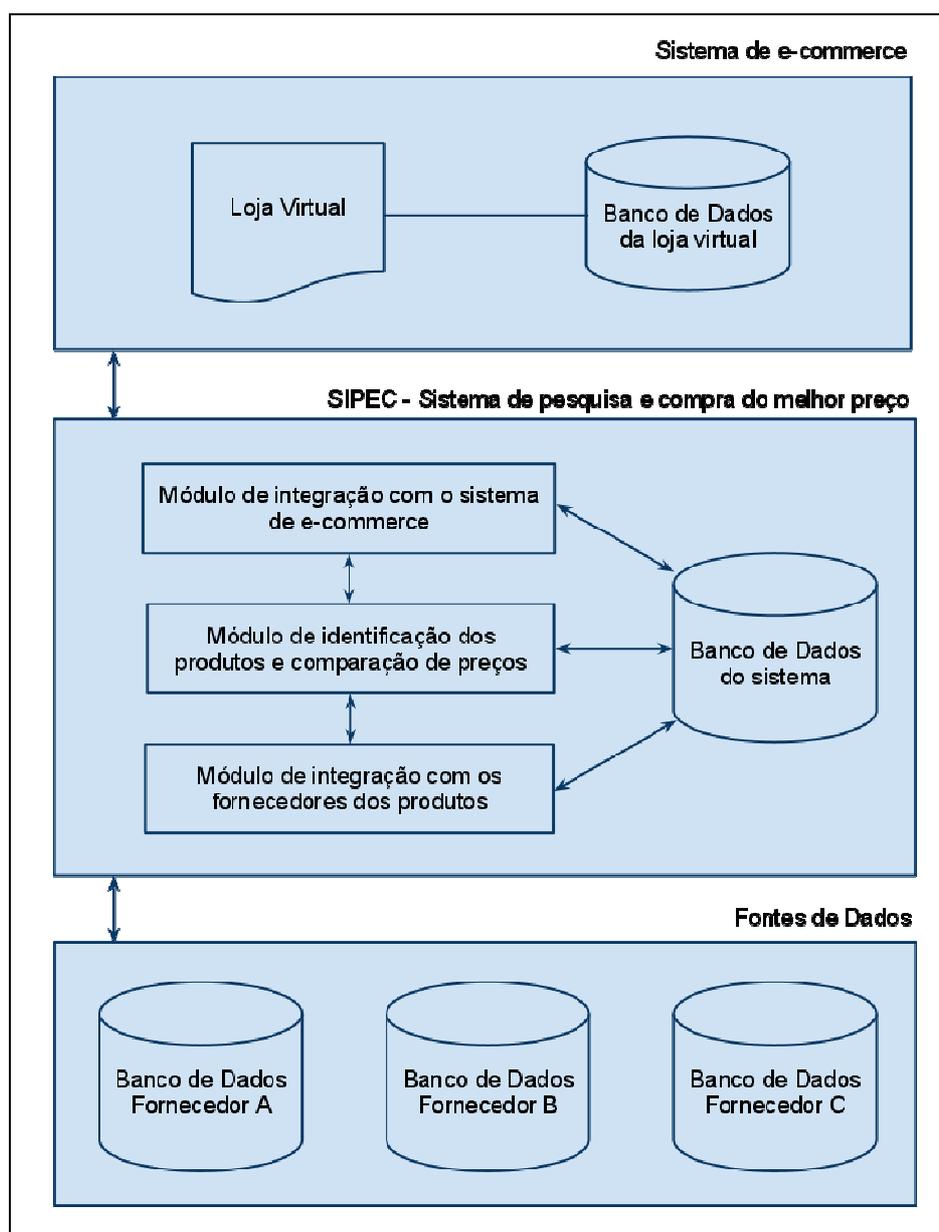
Para que o sistema levantado na fase de análise/concepção seja construído e seus requisitos sejam atendidos, torna-se necessário fazer um planejamento de como isto será conduzido. Wazlawick (2004) diz que “a fase de projeto enfatiza a proposta de uma solução que atenda aos requisitos da análise. Então, se a análise é uma investigação para tentar descobrir o que o

cliente quer, o projeto consiste em propor uma solução com base no conhecimento adquirido na análise”.

Teremos então na fase de projeto, as seguintes etapas: arquitetura global do sistema, sistema de *e-commerce*, diagramas *UML* (caso de uso, atividades e classes) e o diagrama de banco de dados (*ER*).

### **3.2.1 Arquitetura global do sistema**

A solução proposta para atender ao que foi levantado na fase de análise/concepção, está representada através da arquitetura global do sistema, que pode ser observada na figura 5. Na seqüência, uma breve descrição do comportamento dos componentes desta arquitetura.



**Figura 5 – Arquitetura global do sistema**

Fonte: Elaborado pelo autor

A arquitetura global do sistema é composta por três componentes macros: o sistema de *e-commerce*, o SIPEC - Sistema *On-line* de Pesquisa e Compra do Menor Preço e as fontes de dados.

Cada fornecedor terá um *Web Service* para que seja feita diariamente a consulta dos produtos. Este *Web Service* retornará um arquivo *XML* (Apêndice A – Estrutura do arquivo *XML* com a lista de produtos) com as informações dos produtos (código, título, descrição, preço e etc.).

O módulo de integração com os fornecedores dos produtos acessa o banco de dados de cada fornecedor via *Web Service* e recebe como retorno um arquivo *XML*. Estas informações são armazenadas no banco de dados do sistema, para que o mapeamento e a identificação dos produtos sejam feitas através do código universal (*EAN*), e desta forma, cada produto das diversas fontes de dados serão identificados conforme suas características.

Concluída esta etapa, o módulo de identificação dos produtos e comparação de preços é acionado, faz a análise e comparação de preço dos produtos e elege os produtos que possuem os preços mais baratos, armazenando estas informações no banco de dados do sistema.

O módulo de integração com o sistema de e-commerce verifica os produtos que estão no banco de dados do sistema e transfere para o banco de dados da loja virtual, aqueles produtos que foram identificados com os menores preços.

Finalmente, a partir deste momento, os consumidores já podem fazer suas compras normalmente na loja virtual com os produtos atualizados.

### **3.2.2 Sistema de e-commerce**

Uma das primeiras atividades relacionadas ao desenvolvimento do sistema proposto neste trabalho de conclusão de curso foi a pesquisa, entendimento quanto ao funcionamento e escolha de um sistema de e-commerce.

Foram pesquisados alguns sistemas de e-commerce, obedecendo a alguns critérios pré-estabelecidos:

- Ser *software* livre
- Utilizar a linguagem de programação *PHP*
- Utilizar o banco de dados *MySQL*
- Ter uma *interface* simples e intuitiva
- Possuir a *interface* de administração e a loja virtual
- Suporte a *SSL*

A partir destes critérios, os seguintes sistemas de *e-commerce* foram selecionados para avaliação:

- VirtueMart
- osCommerce
- Magento

Após instalar, configurar e utilizar cada um dos sistemas citados a cima, foi possível ter condições de optar por aquele que mais se adequou aos critérios estabelecidos.

A tabela 1 é um comparativo dos sistemas pesquisados.

**Tabela 1 - Quadro comparativo entre os sistemas de *e-commerce* pesquisados**

<b>Crítérios / Sistema</b>	<b>VirtueMart</b>	<b>osCommerce</b>	<b>Magento</b>
<b>Software livre</b>	Atende	Atende	Não atende
<b>Linguagem PHP</b>	Atende	Atende	Atende
<b>Banco de dados MySQL</b>	Atende	Atende	Atende
<b>Interface simples e intuitiva</b>	Atende	Atende	Não atende
<b>Interface de administração e loja virtual</b>	Atende	Atende	Atende
<b>Suporte a SSL</b>	Atende	Atende	Atende
<b>Interface em português</b>	Atende	Atende	Não atende

Fonte: Elaborado pelo autor

A partir deste quadro comparativo, podemos perceber que quase todos atendem aos critérios estabelecidos, e que o *VirtueMart* e o *osCommerce* têm uma pequena vantagem em relação ao Magento. E para fins de demonstração, o sistema de *e-commerce osCommerce* foi escolhido para ser utilizado neste trabalho.

Na figura 6 podemos observar a tela da loja virtual do osCommerce.

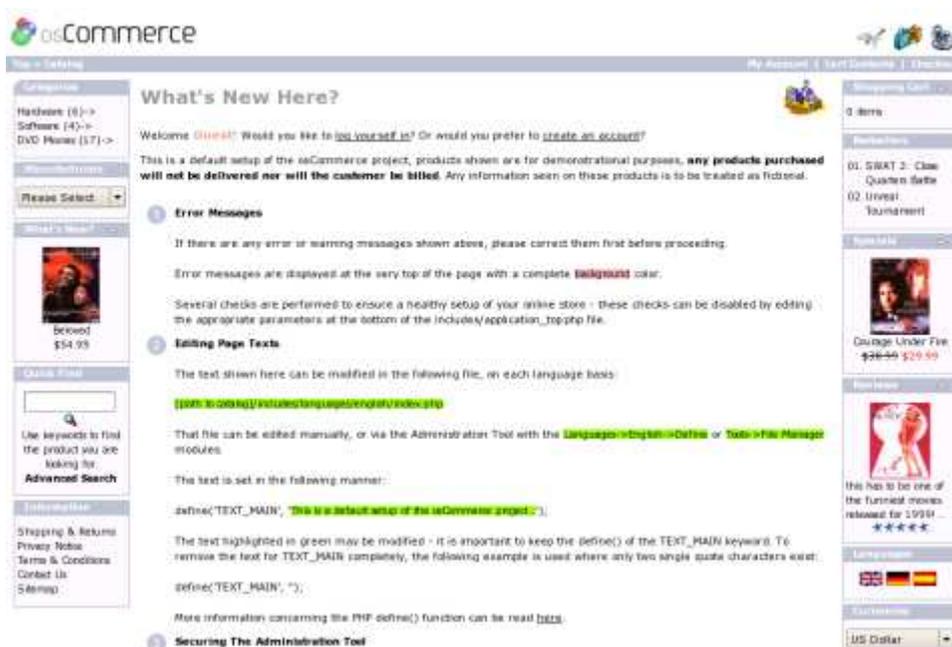


Figura 6 - Tela da loja virtual do osCommerce  
Fonte: osCommerce, 2010

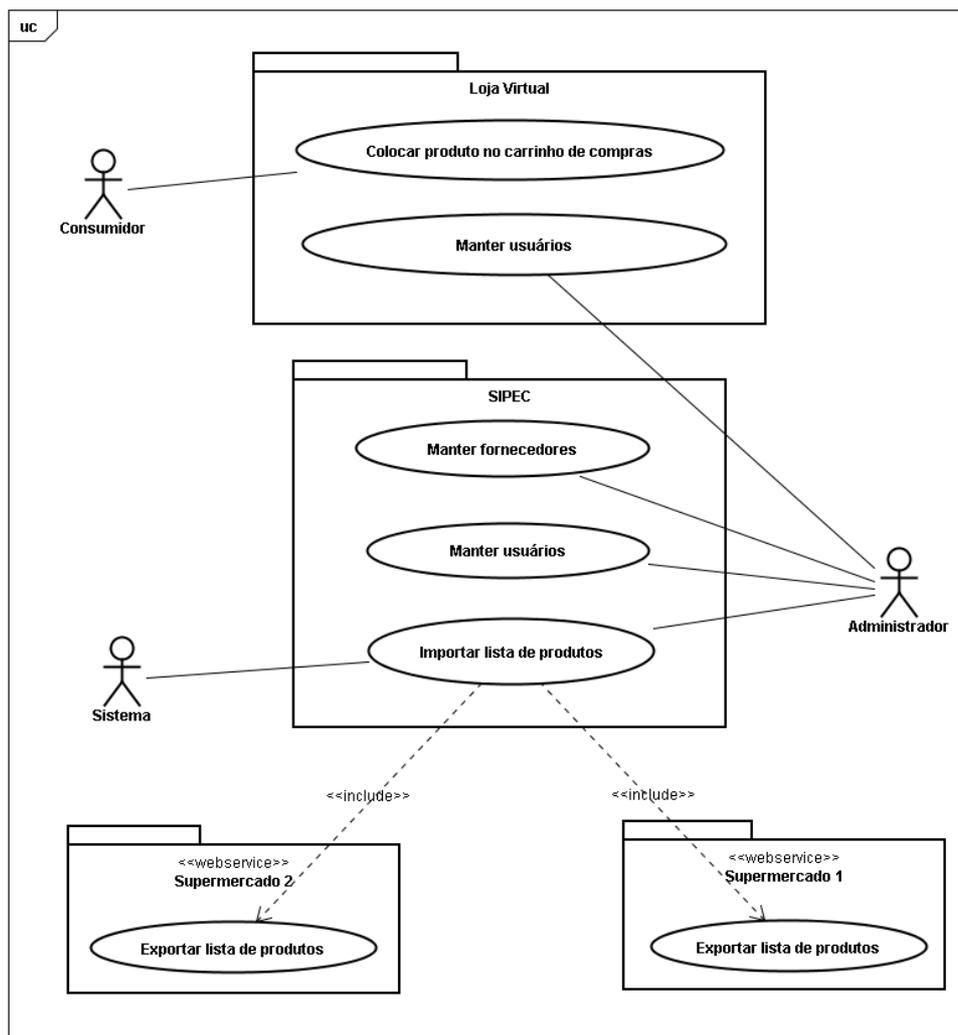
### 3.2.3 Diagramas UML

Através dos diagramas UML serão modelados as entidades, os comportamentos das funcionalidades do sistema e os atores envolvidos.

#### 3.2.3.1 Diagramas de caso de uso e atividades

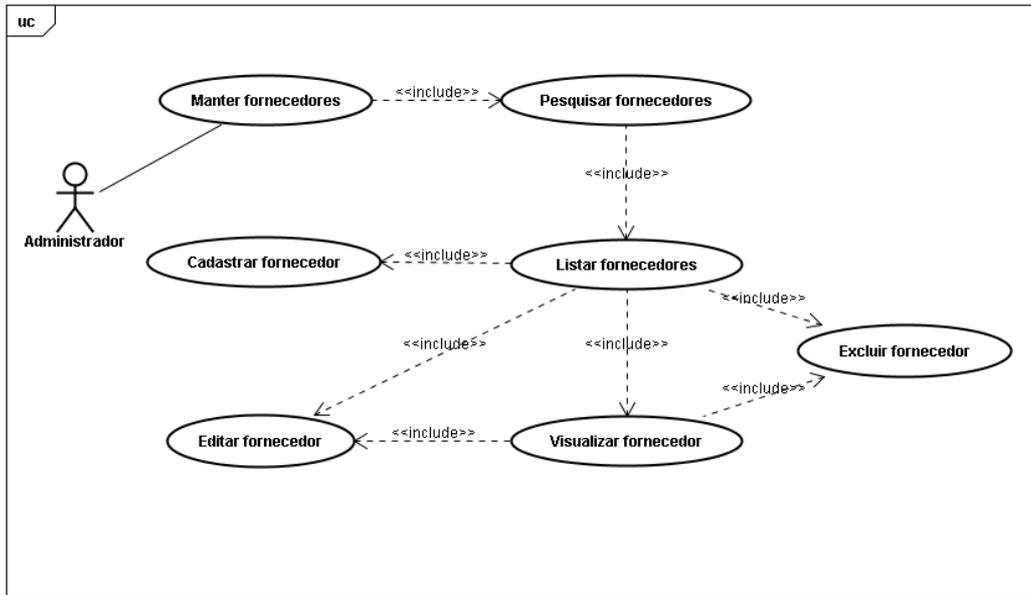
Através de um diagrama de caso de uso, serão modelados os comportamentos de forma superficial, dando assim uma visão geral do sistema. Posteriormente, as funcionalidades mais importantes do sistema, serão detalhadas em outros diagramas de caso de uso e também através de diagramas de atividades.

A figura 7 é um diagrama de caso de uso que representa uma visão geral do sistema.

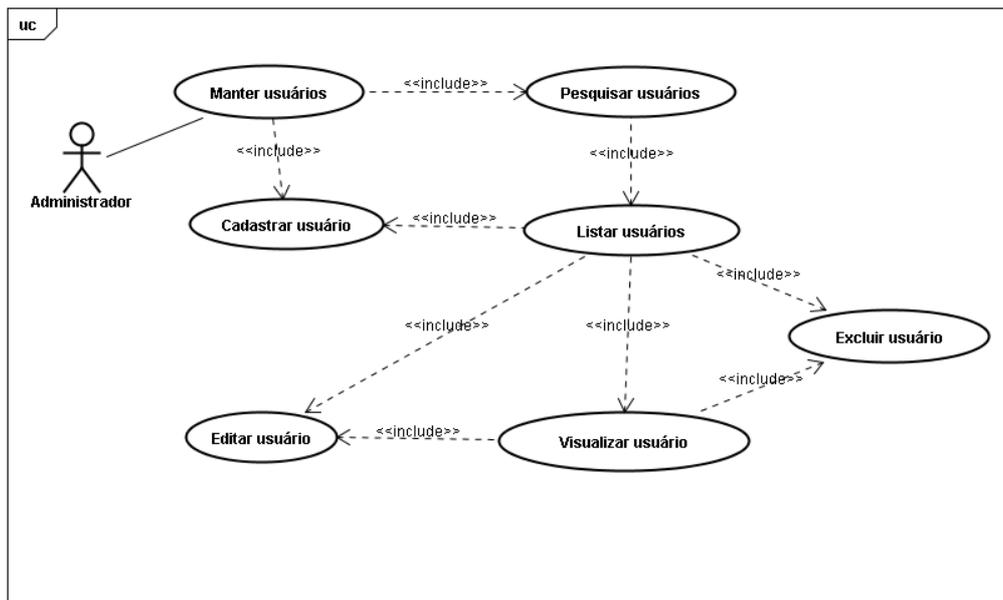


**Figura 7 – Diagrama de caso de uso da visão geral do sistema**  
**Fonte: Elaborado pelo autor**

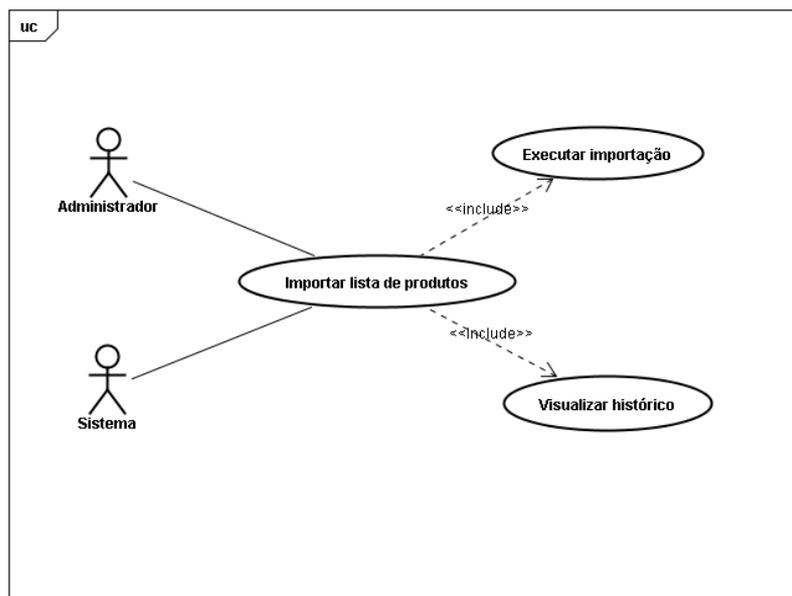
Para uma melhor compreensão das funcionalidades da fronteira central do sistema (SIPEC), os seguintes casos de uso foram detalhados: manter fornecedores, manter usuários e importar lista de produtos. Podemos observá-los abaixo nas Figuras 8, 9 e 10 respectivamente.



**Figura 8 - Diagrama de caso de uso manter fornecedor (detalhado)**  
**Fonte: Elaborado pelo autor**

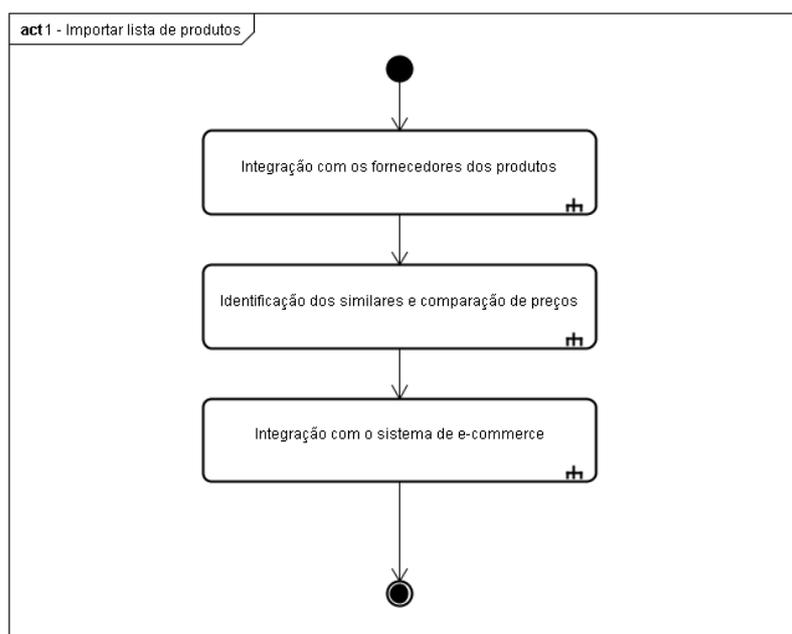


**Figura 9 – Diagrama de caso de uso manter usuários (detalhado)**  
**Fonte: Elaborado pelo autor**



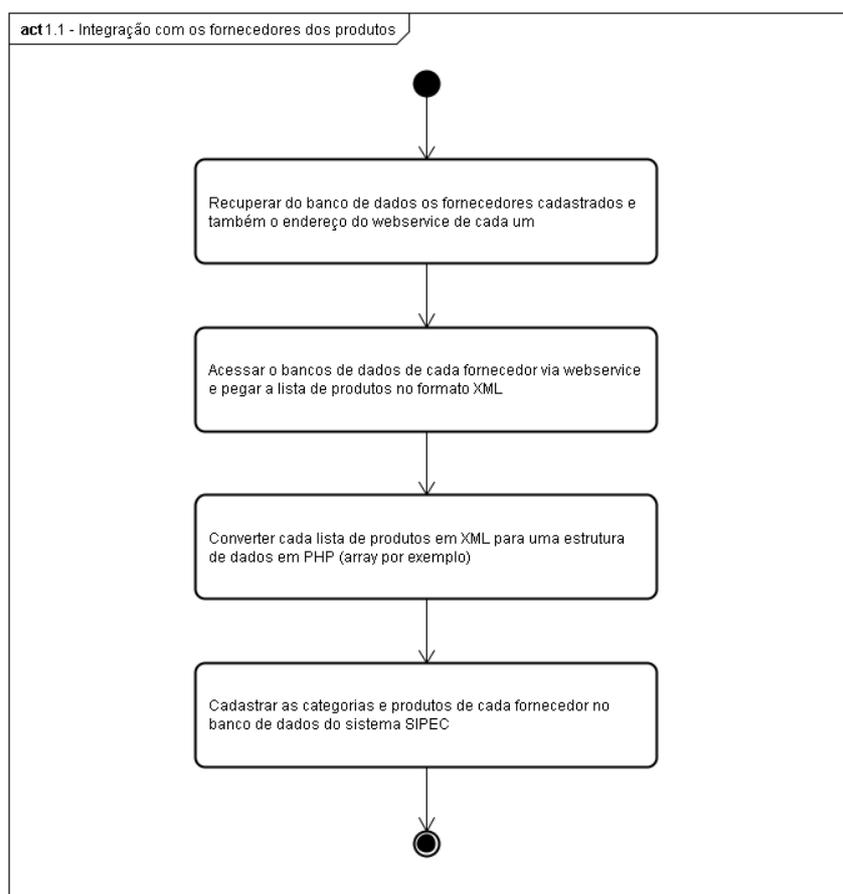
**Figura 10 – Diagrama de caso de uso importar lista de produtos (detalhado)**  
**Fonte: Elaborado pelo autor**

O caso de uso Importar lista de produtos representa uma das funcionalidades principais do sistema, e por este motivo, ele será detalhado em alguns diagramas de atividades. Na Figura 11, é apresentado o diagrama de atividades com uma visão macro desta funcionalidade.



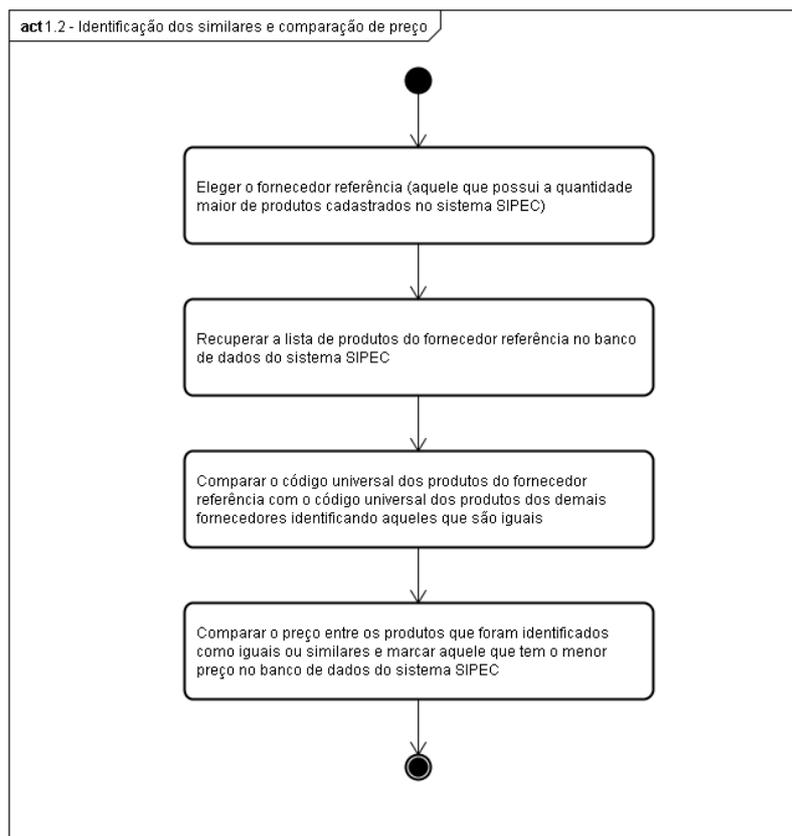
**Figura 11 – Diagrama de atividades (macro) da funcionalidade Importar lista de produtos**  
**Fonte: Elaborado pelo autor**

Para compreender o que é feito em cada um dos passos apresentados do diagrama de atividades acima, foi elaborado um diagrama de atividades detalhado para cada uma das etapas: integração com os fornecedores dos produtos; identificação dos similares e comparação de preços; e integração com o sistema de *e-commerce*. Esses diagramas são apresentados nas Figuras 12, 13 e 14 respectivamente.

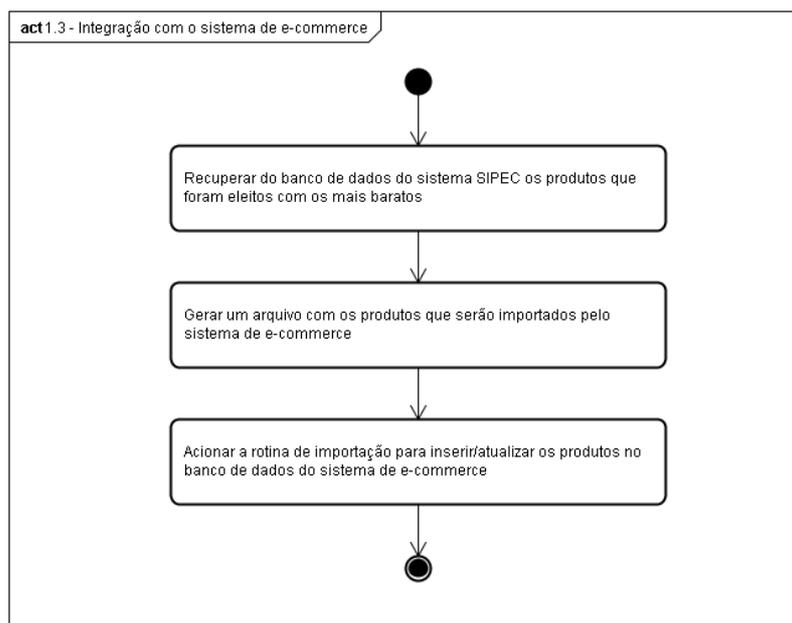


**Figura 12 – Diagrama de atividades (detalhado) da etapa Integração com os fornecedores dos produtos**

**Fonte: Elaborado pelo autor**



**Figura 13 – Diagrama de atividades (detalhado) da etapa Identificação dos similares e comparação de preço**  
**Fonte: Elaborado pelo autor**



**Figura 14 – Diagrama de atividades (detalhado) da etapa Integração com o sistema de e-commerce**  
**Fonte: Elaborado pelo autor**

### 3.2.3.2 Diagrama de classes

Para a elaboração do sistema foram criadas 6 classes, sendo que 3 delas são de entidades relacionadas ao negócio propriamente dito e as demais fazem parte de um *plugin* do Code Igniter utilizado para o controle de acesso dos usuários. Na Figura 15, as classes estão agrupadas de acordo com suas afinidades.

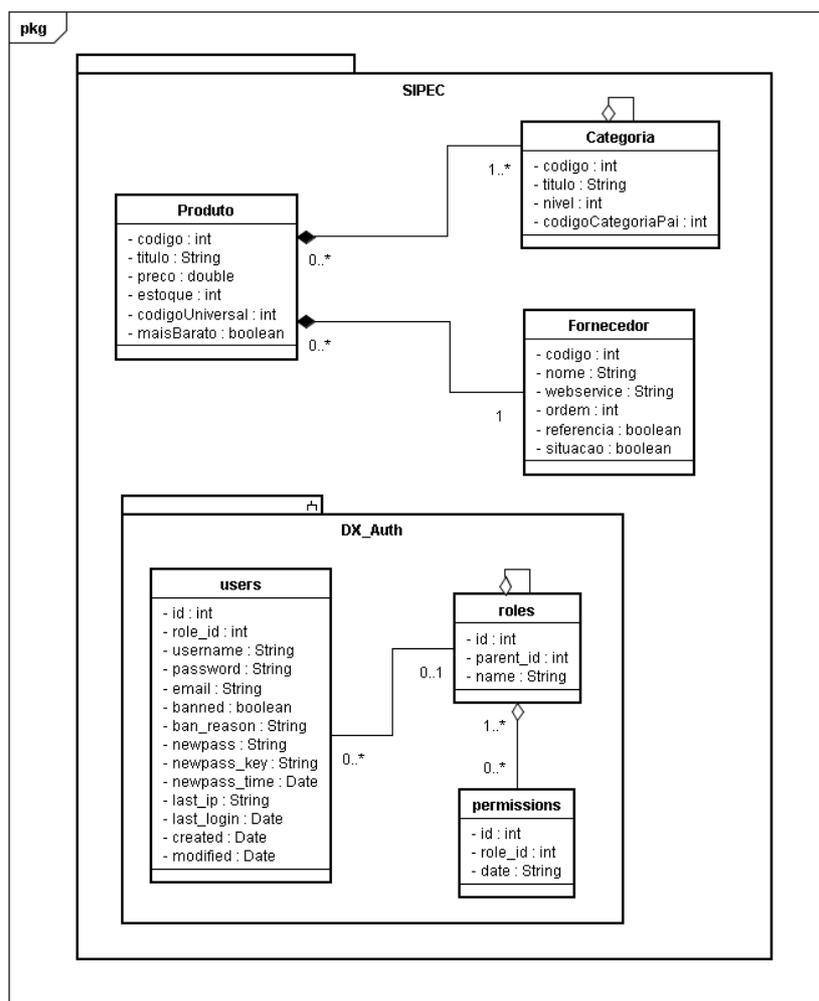


Figura 15 – Diagrama de classes do sistema  
Fonte: Elaborado pelo autor

### 3.2.4 Diagrama do banco de dados

O diagrama ER – Entidade Relacionamento da Figura 16 apresenta todas as tabelas, atributos e seus relacionamentos.

[1,1]

[2,1]

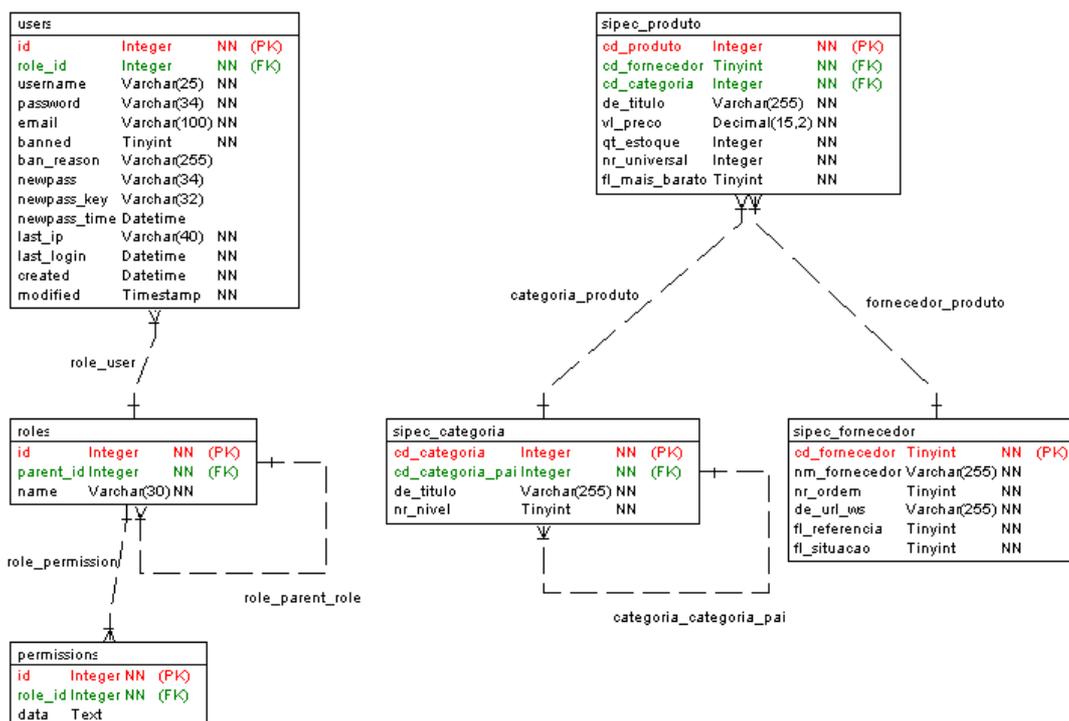


Figura 16 – Diagrama ER do banco de dados integrado do sistema  
Fonte: Elaborado pelo autor

### 3.3 Implementação

Com as fases de análise/concepção e projeto concluídas, vamos partir para a implementação do sistema. Neste momento o sistema é construído de fato.

Teremos então na fase de implementação, as seguintes etapas: criação da base de dados, desenvolvimento das classes e as funcionalidades e telas do sistema.

### 3.3.1 Criação da base de dados

Para criar a base de dados foi utilizado o *script* gerado a partir do diagrama ER do sistema, disponível no apêndice B – SQL para criação da base de dados. As tabelas, chaves e *constraints* foram criadas através da importação do *script* com a ferramenta phpMyAdmin 2.10.3.

### 3.3.2 Desenvolvimento do sistema

Todo o sistema foi desenvolvido no *Code Igniter* (já citado no capítulo 2.1 Programação). Foi utilizada a última versão 1.7.2, que se encontra disponível para download no site oficial do *framework*. Também foi utilizado o *plugin DX Auth* versão 1.0, para o controle de acesso dos usuários.

A instalação é relativamente simples e basta seguir as instruções contidas na figura 17, retirada do site do *Code Igniter*.

#### Instruções de Instalação

O Code Igniter é instalado em 4 passos:

1. Descompacte o arquivo.
2. Suba os diretórios e arquivos para seu servidor. Normalmente o arquivo `index.php` estará ficando raiz.
3. Abra o arquivo `application/config/config.php` em um editor de texto e altere a sua URL base.
4. Se você pretende usar um banco de dados, abra o arquivo `application/config/database.php` file em um editor de texto e altere as configurações do seu banco.

**Figura 17 – Instruções de instalação do framework Code Igniter**  
Fonte: Code Igniter – Guia do Usuário, 2010

Com o ambiente configurado, foram desenvolvidos as classes de acordo com o que foi planejado na fase de projeto e os códigos fontes estão disponíveis no Apêndice C – Código fonte do sistema. Isto deu resultado as funcionalidades e telas que veremos na próxima etapa.

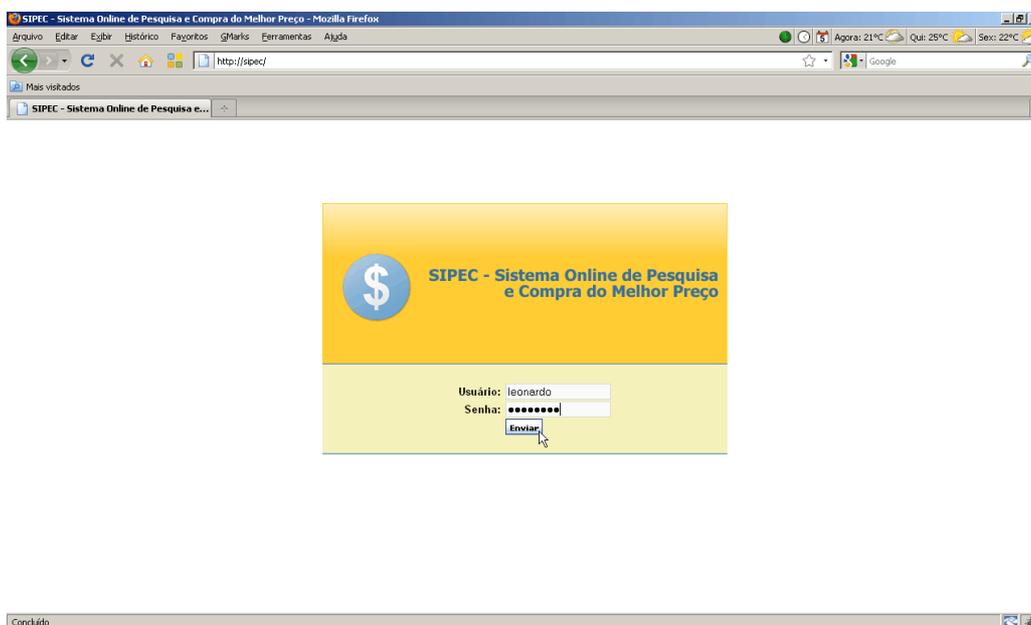
### 3.3.3 Funcionalidades e telas

O sistema desenvolvido é composto por algumas funcionalidades que serão descritas e mostradas abaixo, através de um texto explicativo e sua respectiva tela.

### 3.3.3.1 Controle de acesso a interface administrativa do sistema

Para atender ao que foi descrito no requisito não funcional 005 (RNF005 - Somente usuários cadastrados poderão utilizar a *interface* administrativa do sistema), quando um usuário tenta fazer acesso a alguma funcionalidade, o sistema verifica se o mesmo previamente já informou o nome de usuário e a senha. Caso já tenha informado, o sistema permite o acesso ao sistema. Caso não tenha informado ainda, ao tentar acessar a uma determinada funcionalidade, o sistema redireciona o usuário para a tela de autenticação, e solicita que seja informado um nome de usuário e senha válidos.

Podemos observar a tela correspondente no sistema na figura 18 abaixo.



**Figura 18 – Tela de autenticação da *interface* administrativa do sistema**  
**Fonte: Elaborado pelo autor**

### 3.3.3.2 Gerenciar os usuários do sistema

Para atender ao que foi descrito no requisito funcional 008 (RF008 – O sistema deve permitir o cadastro de usuários), através desta funcionalidade, é possível gerenciar os usuários que poderão acessar a *interface* administrativa do sistema.

Na figura 19, onde a tela correspondente é exibida, podemos observar que é possível pesquisar e listar os usuários cadastrados. Através desta tela

também podemos cadastrar um novo usuário no sistema, editar suas informações, consultar os detalhes das informações de um usuário, bem como excluir um determinado registro de usuário que não desejamos mais.

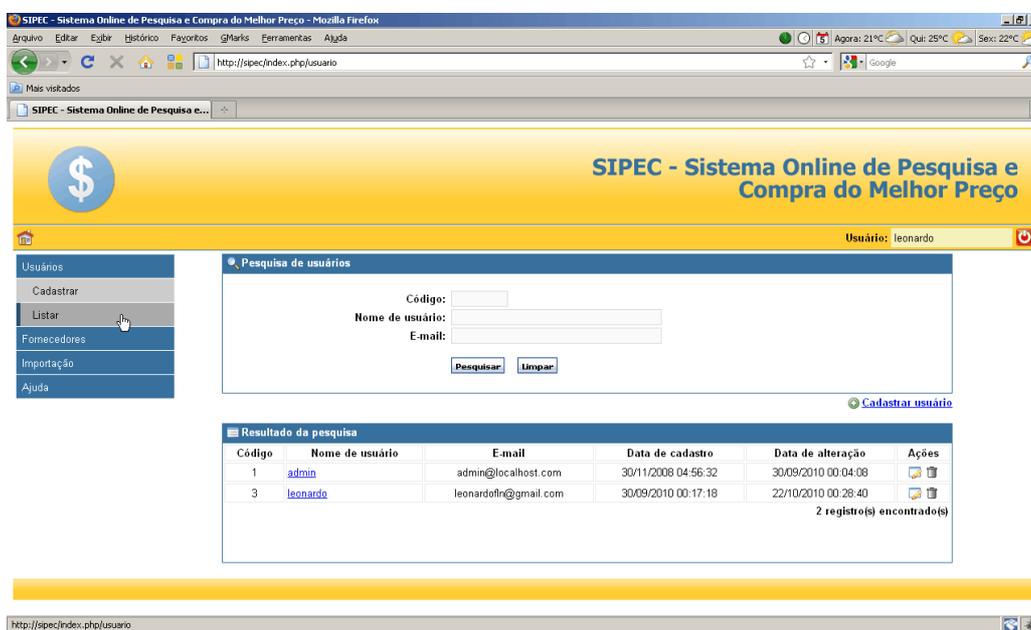
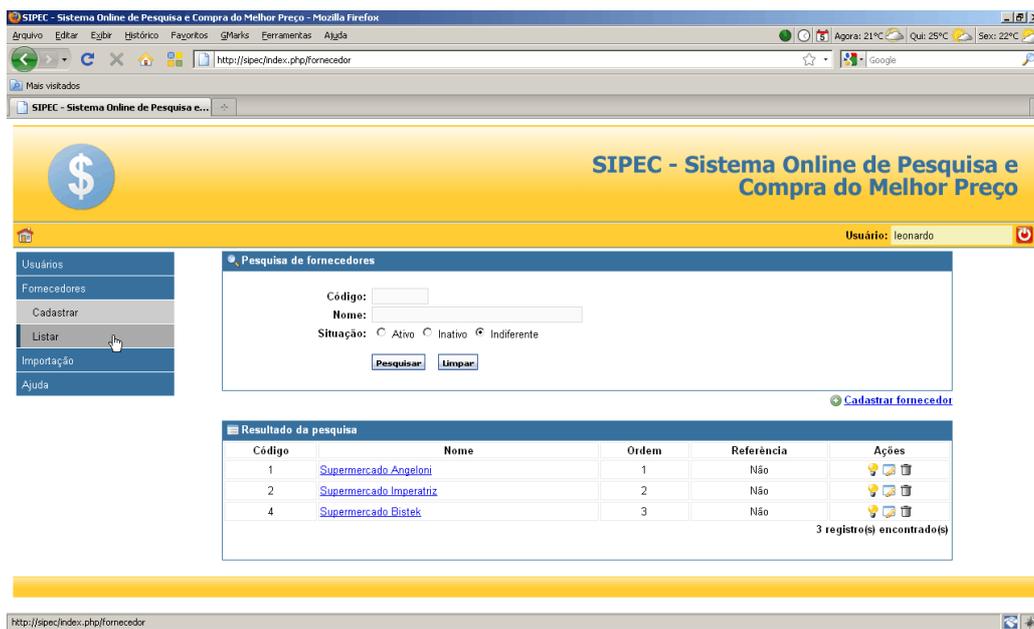


Figura 19 – Tela de gerenciamento de usuários (pesquisa, cadastro, edição e etc.)  
Fonte: Elaborado pelo autor

### 3.3.3.3 Gerenciar os fornecedores do sistema

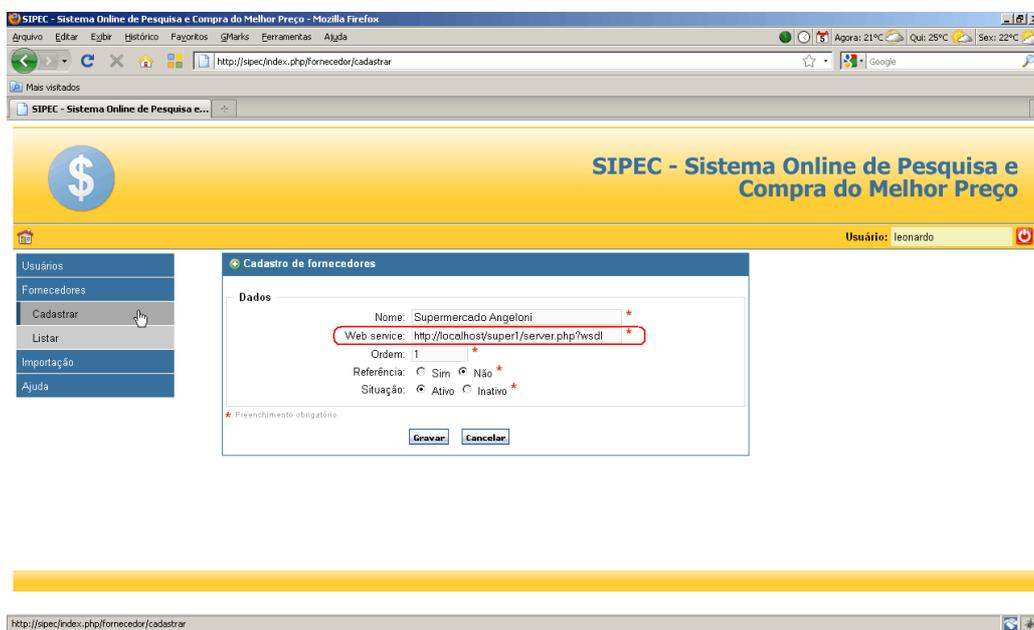
Para atender ao que foi descrito no requisito funcional 007 (RF007 – O sistema deve permitir o cadastro de fornecedores), através desta funcionalidade, é possível gerenciar os fornecedores serão acessados através de *Web Service*, para disponibilizar a lista de produtos.

Na figura 20, onde a tela correspondente é exibida e de forma semelhante a tela de usuários, podemos observar que é possível pesquisar e listar os fornecedores cadastrados. Através desta tela também podemos cadastrar um novo fornecedor no sistema, editar suas informações, consultar os detalhes das informações de um fornecedor, bem como excluir um determinado registro de fornecedor que não desejamos mais.



**Figura 20 – Tela de gerenciamento de fornecedores (pesquisa, cadastro, edição e etc.)**  
 Fonte: Elaborado pelo autor

Na figura 21, que é a tela de cadastro de fornecedor, podemos observar o campo onde é informado o endereço do *Web Service*, que será utilizado para obter a lista de produtos deste fornecedor.



**Figura 21 – Tela de cadastro de fornecedor com destaque para o campo *Web Service***  
 Fonte: Elaborado pelo autor

### 3.3.3.4 Processo de importação dos produtos

Para atender ao que foi descrito nos requisitos funcionais 002 à 006 (RF002 – O sistema deve se conectar com os supermercados para obter a lista de produtos; RF003 – O sistema deve comparar a lista de produtos dos fornecedores e identificar os produtos iguais; RF004 – O sistema deve eleger os produtos mais baratos para disponibilizar no supermercado virtual; RF005 – O sistema deve importar os produtos eleitos mais baratos para o supermercado virtual; RF006 – O sistema deve fazer a comparação de preços dos produtos dos fornecedores), através desta funcionalidade, ocorre todo o processo de importação dos produtos. Isto compreende desde a conexão via *Web Service* as fontes de dados dos fornecedores, passando pela comparação de preços e identificação dos menores preços, até a carga que é feita no sistema de *e-commerce* (loja virtual).

Na figura 22, podemos observar o processo de execução em andamento.

The screenshot shows the SIPEC web application interface. The title bar reads "SIPEC - Sistema Online de Pesquisa e Compra do Melhor Preço - Mozilla Firefox". The address bar shows "http://sipec/index.php/importacao/executar". The page header includes a dollar sign icon and the text "SIPEC - Sistema Online de Pesquisa e Compra do Melhor Preço". The user is logged in as "leonardo". A sidebar menu on the left contains "Usuários", "Fornecedores", "Importação", "Executar", "Histórico", and "Ajuda". The main content area displays "Execução do processo de importação dos produtos em andamento..." and a table with the following data:

Atividade	Situação
Início do processo de importação em 15/10/2010 às 03:00	✓
Acessando os web services para obter as lista de produtos	✓
Cadastrando as categorias e produtos no banco de dados do sistema	✓
Identificando os produtos similares e fazendo a comparação de preços	✓
Executando a carga dos produtos no sistema de e-commerce	⚙

Legenda: ✓ concluído ⚙ em andamento

Figura 22 – Tela de execução do processo de importação dos produtos  
Fonte: Elaborado pelo autor

### 3.3.3.5 Sistema e-commerce com a carga dos produtos

Para atender ao que foi descrito no requisito funcional 001 (RF001 – Os consumidores poderão fazer compras através de um supermercado virtual),

através desta funcionalidade, o consumidor pode navegar pela loja virtual, escolher os produtos que deseja comprar e colocar no carrinho de compras, finalizar e fechar o seu pedido e etc. Todas essas funcionalidades fazem parte do sistema de *e-commerce* que está integrado ao sistema de pesquisa e compra do menor preço.

Na figura 23, podemos observar a consulta a um determinado produto na loja virtual.



**Figura 23 – Tela de consulta de um determinado produto na loja virtual**  
**Fonte: Elaborado pelo autor**

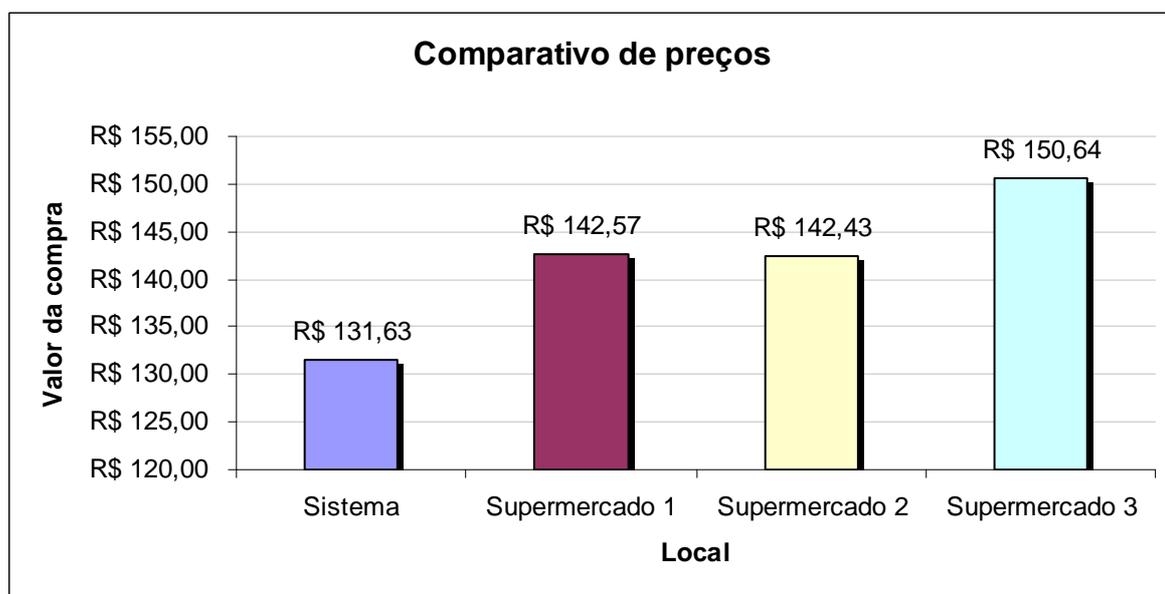
## 3.4 Testes

### 3.4.1 Experimentos realizados

Com o ambiente pronto e com uma carga de produtos (que foi capturada manualmente de alguns supermercados), foi possível realizar alguns experimentos no sistema.

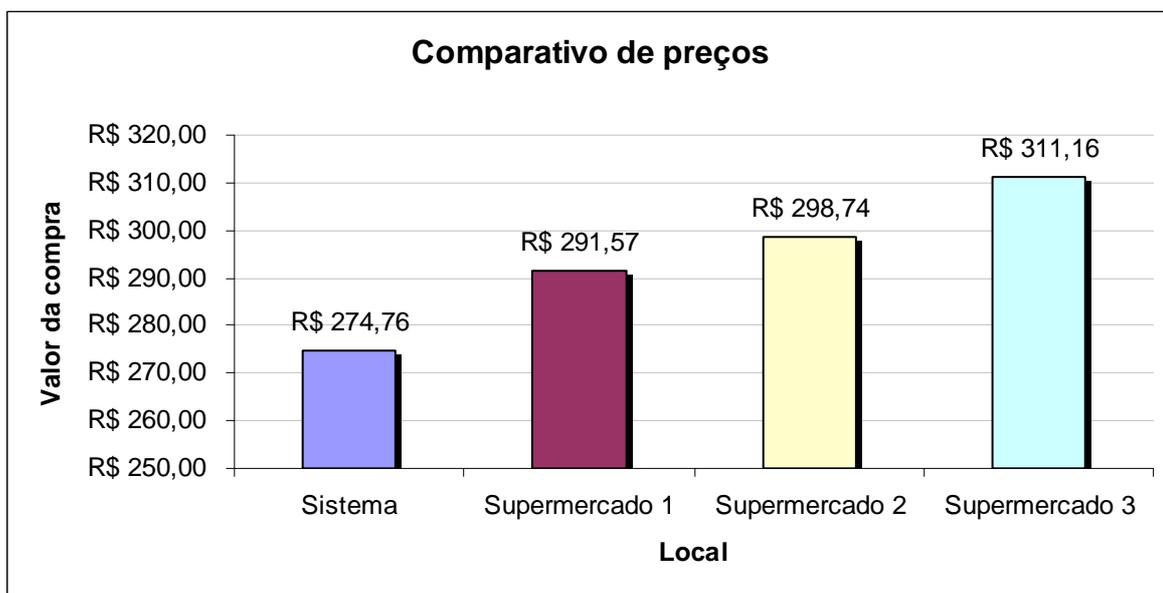
O primeiro experimento foi a compra de 63 produtos, que resultou em uma diferença no preço final de R\$19,01 (13%), comparando o valor gerado pelo sistema (os produtos mais baratos de cada supermercado) e o supermercado em que a soma dos valores dos produtos teve o preço mais alto (Supermercado 3). Fazendo uma projeção, em um ano, isto representa uma

economia de aproximadamente R\$230,00 para o consumidor. No gráfico apresentado na figura 24, podemos observar o comparativo de preços desta lista de 63 produtos.



**Figura 24 – Gráfico com o comparativo de preços da lista com 63 produtos**  
**Fonte: Elaborado pelo autor**

O segundo experimento foi a compra de 123 produtos, que resultou em uma diferença no preço final de R\$36,40 (12%), comparando o valor gerado pelo sistema (os produtos mais baratos de cada supermercado) e o supermercado em que a soma dos valores dos produtos teve o preço mais alto (Supermercado 3). Fazendo uma projeção, em um ano, isto representa uma economia de aproximadamente R\$435,00 para o consumidor. No gráfico apresentado na figura 25, podemos observar o comparativo de preços desta lista de 123 produtos.



**Figura 25 - Gráfico com o comparativo de preços da lista com 123 produtos**  
**Fonte: Elaborado pelo autor**

## 4. Conclusão

### 4.1 Considerações finais

Foi possível entender o funcionamento e escolher o sistema de *e-commerce* mais adequado, apesar de que, quase todos eles poderiam ser utilizados. Isto é um ponto positivo, pois se necessário, o sistema de *e-commerce* pode ser substituído, causando baixo impacto no projeto.

O uso de *Web Service* na integração com as fontes de dados, se demonstrou satisfatório e atendeu ao que era esperado. Desta forma, os fornecedores têm mais flexibilidade para exportar a lista de produtos (arquivo *XML*), e também compatibilidade, pois não precisam modificar em nada sua estrutura (banco de dados e linguagem de programação) para poder integrar seus dados com o sistema aqui proposto. Um dos pontos negativos, no que se refere a integração com as fontes de dados, está relacionada a segurança.

Com relação ao mapeamento e a identificação dos produtos, inicialmente iriam ser utilizadas as funções de similaridades do *PHP* e alguns tratamentos nos strings comparados (remoção de: espaços duplicados, espaços no início/fim e pontuação). Porém, esta solução não atendeu as necessidades, devido ao fato de que cada fornecedor mantém o nome dos produtos como bem entende, ou seja, não existe um padrão. Uma alternativa foi utilizar para o mapeamento e a identificação dos produtos, o código *EAN*, que é o padrão de sistema de identificação (código de barras) utilizado nos estabelecimentos comerciais no Brasil.

Já com relação à escolha do menor preço, foi bastante simples, uma vez que os produtos já haviam sido identificados e mapeados, e elaborar um algoritmo para escolha do menor preço foi trivial.

A rotina de importação dos produtos no sistema de *e-commerce* também acabou sendo bem simples, tendo em vista que os dois sistemas utilizam a mesma linguagem de programação (*PHP*) e banco de dados (*MySQL*).

Finalmente, foi possível realizar alguns testes e demonstrar que, através dos experimentos realizados, o consumidor consegue fazer uma economia de

aproximadamente 13% em suas compras realizadas através do sistema de pesquisa e compra do menor preço.

## **4.2 Trabalhos futuros**

Um dos pontos negativos levantados no trabalho, está relacionado a segurança no uso de *Web Services*. Como sugestão de melhoria, poderia ser estudada uma forma de adicionar características ao projeto, que tragam mais segurança na troca de informações entre o sistema aqui proposto com as fontes de dados dos fornecedores.

Outro ponto que poderia ser explorado como trabalho futuro, seria a utilização dos mesmos conceitos em outros segmentos do setor varejista, como por exemplo: construção civil, saúde e educação.

Como o processo de compra é consideravelmente complexo, ele não foi tratado e explorado neste TCC. Fica também como sugestão de trabalho futuro: o entendimento, a compreensão e desenvolvimento do processo de compra para o sistema de pesquisa e compra do menor preço.

Além disso, outras melhorias/modificações também podem ser adicionadas ao projeto, permitindo sua evolução e contribuindo para que os consumidores tenham mais opções de ferramentas, no momento da pesquisa e comparação de preços.

## REFERÊNCIAS BIBLIOGRÁFICAS

ALBERTIN, Alberto Luiz. **Comércio Eletrônico: modelo, aspectos e contribuições de sua aplicação**. São Paulo: Atlas, 2004. 318p.

Code Igniter - Guia do Usuário. Disponível em:  
<<http://www.codeigniter.com.br/manual/index.html>>. Acesso em: 19 jun. 2010.

DALL'OGGIO, Pablo. **PHP: Programando com Orientação a Objetos**. São Paulo: Novatec, 2007. 574 p.

ELMASRI, Ramez; NAVATHE, Shamkant B.. **Sistemas de Banco de Dados**. 4. ed. São Paulo: Pearson Addison Wesley, 2005. 724 p.

ERDEI, Guillermo E. **Código de barras: desenvolvimento, impressão e controle da qualidade**. São Paulo: Makron Books, 1994. 220p.

JULIANI, Jordan Paulesky. **A SOCIALIZAÇÃO DE CONHECIMENTO ENTRE CONSUMIDORES NA BUSCA DE MELHORES ALTERNATIVAS DE COMPRA. UM MODELO TECNOLÓGICO**. 2008. 130 f. Tese (Doutorado em Engenharia e Gestão do Conhecimento) – Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento, UFSC, Florianópolis.

LIBERTY, Jesse; KRALEY, Mike. **Aprendendo a desenvolver documentos XML para a Web**. São Paulo: Makron Books, 2001. 274 p.

Magento - eCommerce Software for Growth Disponível em:  
<<http://www.magentocommerce.com/>>. Acesso em: 19 jun. 2010.

osCommerce, Open Source Online Shop E-Commerce Solutions. Disponível em: <<http://www.oscommerce.com/>>. Acesso em: 19 jun. 2010.

PHP: Manual do PHP. Disponível em:  
<[http://br2.php.net/manual/pt\\_BR/index.php](http://br2.php.net/manual/pt_BR/index.php)>. Acesso em: 26 jun. 2010.

PRESSMAN, Roger S.. **Engenharia de Software**. 6. ed. São Paulo: McGraw-Hill, 2006. 720 p.

REVISTA VEJA. **A busca pelo preço mais baixo**. Disponível em: <[http://veja.abril.com.br/210307/p\\_078.shtml](http://veja.abril.com.br/210307/p_078.shtml)>. Acesso em: 26 jun. 2010.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S.. **Sistema de Banco de Dados**. 5. ed. Rio de Janeiro: Elsevier, 2006. 781 p.

TURBAN, Efraim; KING, David. **Comércio Eletrônico: estratégia e gestão**. São Paulo: Prentice Hall, 2004. 436p.

VirtueMart. Disponível em: <<http://www.virtuemart.net/>>. Acesso em: 19 jun. 2010.

WAZLAWICK, Raul Sidnei. **Análise e Projeto de Sistemas de Informação Orientados a Objetos**. Rio de Janeiro: Elsevier, 2004. 298 p.

Web Service - Wikipédia. Disponível em <[http://pt.wikipedia.org/wiki/Web\\_service](http://pt.wikipedia.org/wiki/Web_service)>. Acesso em: 26 jun. 2010.

WELLING, Luke; THOMSON, Laura. **PHP e MySQL desenvolvimento Web**. 3. ed. Rio de Janeiro: Elsevier, 2005. 712 p.

ZYNGIER, Mauro Luiz; GROSSMANN, Fábio. **Código de barras: da teoria à prática**. 2. ed. São Paulo: Nobel, 1991. 104p.

## APÊNDICES

### APÊNDICE A – ESTRUTURA DO ARQUIVO XML COM A LISTA DE PRODUTOS

```

<?xml version="1.0" encoding="iso-8859-1"?>
<produtos>
  <categoria>
    <nivel>1</nivel>
    <nome>Automotivo</nome>
    <categoria>
      <nivel>2</nivel>
      <nome>Aromatizante</nome>
      <produto>
        <codigo>789603410024</codigo>
        <titulo>Desodorizador RODABRISA gel tutti frutti pote 60g</titulo>
        <preco>4.69</preco>
      </produto>
      <produto>
        <codigo>789603410025</codigo>
        <titulo>Desodorizador RODABRISA Automotivo tutti frutti pote
80ml</titulo>
        <preco>4.69</preco>
      </produto>
      <produto>
        <codigo>789603410026</codigo>
        <titulo>Desodorizador RODABRISA gel action pote 60g</titulo>
        <preco>4.69</preco>
      </produto>
      <produto>
        <codigo>789603410027</codigo>
        <titulo>Refil RODABRISA Air lavanda unidade</titulo>
        <preco>6.59</preco>
      </produto>
    </categoria>
  <categoria>
    <nivel>2</nivel>
    <nome>Cera</nome>
    <produto>
      <codigo>789603410102</codigo>
      <titulo>Cera RODABRILL Cerabrill líquida frasco 200ml</titulo>
      <preco>7.29</preco>
    </produto>
    <produto>
      <codigo>789603410103</codigo>
      <titulo>Cera para limpeza BRILHOLAC em pasta lata 400g</titulo>
      <preco>10.75</preco>
    </produto>
  </categoria>
</produtos>

```

```

    </produto>
  </categoria>
</categoria>
<categoria>
  <nivel>1</nivel>
  <nome>teste</nome>
  <categoria>
    <nivel>2</nivel>
    <nome>teste-subcategoria</nome>
    <produto>
      <codigo>789603410344</codigo>
      <titulo>titulo do produto</titulo>
      <preco>16.68</preco>
    </produto>
  </categoria>
</categoria>
</produtos>

```

## APÊNDICE B – SQL PARA CRIAÇÃO DA BASE DE DADOS

```

/*
Created                21/10/2010
Modified               21/10/2010
Project                SIPEC - Sistema On-line de Pesquisa e Compra do
Menor Preço
Author                Leonardo de Oliveira
Database              mySQL 5
*/

```

```

Create table permissions (
  id Int(11) NOT NULL AUTO_INCREMENT,
  role_id Int NOT NULL,
  data Text,
  Primary Key (id)
)
ENGINE = MyISAM
ROW_FORMAT = Dynamic;

```

```

Create table roles (
  id Int(11) NOT NULL AUTO_INCREMENT,
  parent_id Int NOT NULL,
  name Varchar(30) NOT NULL,
  Primary Key (id)
)
ENGINE = MyISAM
ROW_FORMAT = Dynamic;

```

```

Create table sipec_categoria (
    cd_categoria Int(3) UNSIGNED NOT NULL AUTO_INCREMENT,
    cd_categoria_pai Int UNSIGNED NOT NULL,
    de_titulo Varchar(255) NOT NULL,
    nr_nivel Tinyint(3) UNSIGNED NOT NULL,
    Primary Key (cd_categoria)
)
ENGINE = MyISAM
ROW_FORMAT = Dynamic;

```

```

Create table sipec_fornecedor (
    cd_fornecedor Tinyint(3) UNSIGNED NOT NULL AUTO_INCREMENT,
    nm_fornecedor Varchar(255) NOT NULL,
    nr_ordem Tinyint(3) UNSIGNED NOT NULL,
    de_url_ws Varchar(255) NOT NULL,
    fl_referencia Tinyint(3) NOT NULL DEFAULT "0",
    fl_situacao Tinyint(3) NOT NULL DEFAULT "1",
    Primary Key (cd_fornecedor)
)
ENGINE = MyISAM
ROW_FORMAT = Dynamic;

```

```

Create table sipec_produto (
    cd_produto Int(10) UNSIGNED NOT NULL AUTO_INCREMENT,
    cd_fornecedor Tinyint UNSIGNED NOT NULL,
    cd_categoria Int UNSIGNED NOT NULL,
    de_titulo Varchar(255) NOT NULL,
    vl_preco Decimal(15,2) NOT NULL,
    qt_estoque Int NOT NULL,
    nr_universal Int NOT NULL,
    fl_mais_barato Tinyint(3) NOT NULL DEFAULT "0",
    Primary Key (cd_produto)
)
ENGINE = MyISAM
ROW_FORMAT = Dynamic;

```

```

Create table users (
    id Int(11) NOT NULL AUTO_INCREMENT,
    role_id Int NOT NULL,
    username Varchar(25) NOT NULL,
    password Varchar(34) NOT NULL,
    email Varchar(100) NOT NULL,
    banned Tinyint(1) NOT NULL DEFAULT "0",
    ban_reason Varchar(255),
    newpass Varchar(34),
    newpass_key Varchar(32),
    newpass_time Datetime,

```

```

        last_ip Varchar(40) NOT NULL,
        last_login Datetime NOT NULL DEFAULT "0000-00-00 00:00:00",
        created Datetime NOT NULL DEFAULT "0000-00-00 00:00:00",
        modified Timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON
UPDATE CURRENT_TIMESTAMP,
        Primary Key (id)
    )
ENGINE = MyISAM
ROW_FORMAT = Dynamic;

```

Alter table roles

```

    add Foreign Key (parent_id) references roles (id)
    on delete restrict
    on update restrict;

```

Alter table users

```

    add Foreign Key (role_id) references roles (id)
    on delete restrict
    on update restrict;

```

Alter table permissions

```

    add Foreign Key (role_id) references roles (id) ]
    on delete restrict
    on update restrict;

```

Alter table sipec\_categoria

```

    add Foreign Key (cd_categoria_pai) references sipec_categoria
(cd_categoria)
    on delete restrict
    on update restrict;

```

Alter table sipec\_producto

```

    add Foreign Key (cd_categoria) references sipec_categoria
(cd_categoria)
    on delete restrict
    on update restrict;

```

Alter table sipec\_producto

```

    add Foreign Key (cd_fornecedor) references sipec_fornecedor
(cd_fornecedor)
    on delete restrict
    on update restrict;

```

## APÊNDICE C – CÓDIGO FONTE DO SISTEMA

### **\\css\\estilo.css**

```
body {
    font-family: Arial, Helvetica, sans-serif;
    font-size: 0.9em;
}
p {
    line-height: 1.5em;
}
table {
    font-family: Arial, Helvetica, sans-serif;
    font-size: 13px;
}
select {
    height: 20px;
}
input, textarea, select {
    background-color: #FAFAFA;
    border: 1px solid #DADADA;
    padding: 1px;
}
fieldset {
    border: 1px solid #DADADA;
    font-weight: bold;
}
.campoDesabilitado {
    background-color: #DADADA;
    border: 1px solid #FAFAFA;
    padding: 1px;
}
ul#menu, ul#menu ul {
    list-style-type:none;
    margin: 0;
    padding: 0;
    width: 15em;
}
ul#menu a {
    display: block;
    text-decoration: none;
}
ul#menu li {
    margin-top: 1px;
}
ul#menu li a {
    background: #37709D;
    color: #fff;
    padding: 0.5em;
}
ul#menu li a:hover {
```

```

    background: #264E6F;
}
ul#menu li ul li a {
    background: #ccc;
    color: #000;
    padding-left: 20px;
}
ul#menu li ul li a:hover {
    background: #aaa;
    border-left: 5px #264E6F solid;
    padding-left: 15px;
}
.code          { border: 1px solid #ccc; list-style-type: decimal-leading-zero;
padding: 5px; margin: 0; }
.code code     { display: block; padding: 3px; margin-bottom: 0; }
.code li       { background: #ddd; border: 1px solid #ccc; margin: 0 0 2px
2.2em; }
.indent1       { padding-left: 1em; }
.indent2       { padding-left: 2em; }

.borda          { border: 1px solid #EAEAEA; }
.borda_azul    { border: 1px solid #37709D; }
.borda_amarela { border: 1px solid #FFCC33; }
.borda_escura  { border: 1px solid #999999; }

.fundo_azul    { background-color: #37709D; }
.fundo_bege    { background-color: #F5F1BA; }
.fundo_amarelo { background-color: #FFCC33; }

.tamanhoGrande {
    width: 260px;
}
.tamanhoMedio {
    width: 130px;
}
.tamanhoPequeno {
    width: 70px;
}
.tamanhoMinimo {
    width: 30px;
}
.error {
    background-image: url(/img/check_vermelho.gif);
    background-repeat: no-repeat;
    background-position: top left;
    padding-left: 18px;
}
.fundo_topo {
    background-image: url(/img/topo_r1_c3.gif);
    background-repeat: repeat-all;
}

```

```

}
.fundo_rodape {
  background-image: url(/img/rodape.gif);
  background-repeat: repeat-all;
}
.botao {
  font-family: Verdana, Arial, Helvetica, sans-serif;
  font-size: 10px;
  font-weight: bold;
  height: 20px;
  color: #000000;
  background-image: url(/img/fundo_botao.gif);
  border: 1px solid #1C418C;
}
.tituloTabela {
  font-weight: bold;
  color: #FFFFFF;
  background-color: #37709D;
}
.subtituloTabela {
  font-weight: bold;
  color: #000000;
}
.informativo {
  font-size: 10px;
  color: #AAAAAA;
}
.fundo_cinza { background-color: #EAEAEA; }

```

### **ldados\produtos.xml**

```

<?xml version="1.0" encoding="iso-8859-1"?>
<produtos>
  <categoria>
    <nivel>1</nivel>
    <nome>Automotivo</nome>
  <categoria>
    <nivel>2</nivel>
    <nome>Aromatizante</nome>
    <produto>
      <codigo>2310960</codigo>
      <titulo>Desodorizador RODABRISA gel tutti frutti pote 60g</titulo>
      <preco>4.69</preco>
    </produto>
    <produto>
      <codigo>1539286</codigo>
      <titulo>Desodorizador RODABRISA Automotivo tutti frutti pote 80ml</titulo>
      <preco>4.69</preco>
    </produto>
    <produto>
      <codigo>2310989</codigo>

```

```

    <titulo>Desodorizador RODABRISA gel action pote 60g</titulo>
    <preco>4.69</preco>
  </produto>
  <produto>
    <codigo>2745458</codigo>
    <titulo>Refil RODABRISA Air lavanda unidade</titulo>
    <preco>6.59</preco>
  </produto>
</categoria>
<categoria>
  <nivel>2</nivel>
  <nome>Cera</nome>
  <produto>
    <codigo>1573205</codigo>
    <titulo>Cera RODABRILL Cerabrill líquida frasco 200ml</titulo>
    <preco>7.29</preco>
  </produto>
  <produto>
    <codigo>1670999</codigo>
    <titulo>Cera para limpeza BRILHOLAC em pasta lata 400g</titulo>
    <preco>10.75</preco>
  </produto>
</categoria>
</categoria>
<categoria>
  <nivel>1</nivel>
  <nome>teste</nome>
  <categoria>
    <nivel>2</nivel>
    <nome>teste-subcategoria</nome>
    <produto>
      <codigo>34455778</codigo>
      <titulo>titulo do produto</titulo>
      <preco>16.68</preco>
    </produto>
  </categoria>
</categoria>
</produtos>

```

### **\js\funcoes.js**

```

function excluirForm(form, campo, valor) {
  if (confirm('Deseja realmente excluir este registro?')) {
    form = document.getElementById(form);
    campo = document.getElementById(campo);
    campo.value = valor;
    form.submit();
  }
}

function excluir(url) {

```

```

        if (confirm('Deseja realmente excluir este registro?')) {
            location.href=url;
        }
    }

function exibelInfoTipoPessoa(tipo) {
    pj = document.getElementById("pessoa_juridica");
    pf = document.getElementById("pessoa_fisica");

    if (tipo == 'J') {
        pj.style.display = 'block';
        pf.style.display = 'none';
    } else {
        pf.style.display = 'block';
        pj.style.display = 'none';
    }
}

function clearRow(row) {
    for (var i=0; i<row.cells.length; i++) {
        cell = row.cells[i];
        tipo = cell.childNodes[0].type;
        if (tipo == "checkbox") {
            cell.childNodes[0].checked = false;
        } else if (tipo == "select-one" || tipo == "text") {
            cell.childNodes[0].value = "";
        } else {
            cell.childNodes[0].value = cell.childNodes[0].defaultValue;
        }
    }
}

function duplicaProduto(tableID) {
    table = document.getElementById(tableID);
    tableBase = document.getElementById('tabelaProdutoBase');

    // insere uma linha antes do objeto duplicado
    var tr = document.createElement("tr");
    var td = document.createElement("td");
    var txt = document.createTextNode(String.fromCharCode(160));
    td.appendChild(txt);
    tr.appendChild(td);
    table.tBodies[0].appendChild(tr);

    // duplica a linha
    newRow = tableBase.tBodies[0].rows[0].cloneNode(true);
    //clearRow(newRow);
    table.tBodies[0].appendChild(newRow);

    // procurar pelos ids e altera seus valores

```

```

    qtdeLinhas = table.tBodies[0].rows.length;
    ultimaLinha = qtdeLinhas - 1;
    penultimaLinha = qtdeLinhas - 3;
    indice =
Number(table.tBodies[0].rows[penultimaLinha].cells[0].childNodes[2].tBodies[0].rows[
0].id.substring(8)) + 1;

    tabelaProduto = table.tBodies[0].rows[ultimaLinha].cells[0].childNodes[2];

    tabelaProduto.innerHTML =
tabelaProduto.innerHTML.replace("vl_quantidade_produto[Base]",
"vl_quantidade_produto["+(indice-1)+"]");
    tabelaProduto.innerHTML =
tabelaProduto.innerHTML.replace("vl_quantidade_produto[Base]",
"vl_quantidade_produto["+(indice-1)+"]");
    tabelaProduto.innerHTML =
tabelaProduto.innerHTML.replace("fl_espaco_livre[Base]", "fl_espaco_livre["+(indice-
1)+"]");
    tabelaProduto.innerHTML =
tabelaProduto.innerHTML.replace("fl_espaco_livre[Base]", "fl_espaco_livre["+(indice-
1)+"]");
    tabelaProduto.innerHTML =
tabelaProduto.innerHTML.replace("espaco_livre_Base", "espaco_livre_"+indice);
    tabelaProduto.innerHTML =
tabelaProduto.innerHTML.replace("vl_espaco_livre_largura[Base]",
"vl_espaco_livre_largura["+(indice-1)+"]");
    tabelaProduto.innerHTML =
tabelaProduto.innerHTML.replace("vl_espaco_livre_altura[Base]",
"vl_espaco_livre_altura["+(indice-1)+"]");
    tabelaProduto.innerHTML = tabelaProduto.innerHTML.replace("fl_trilho[Base]",
"fl_trilho["+(indice-1)+"]");
    tabelaProduto.innerHTML = tabelaProduto.innerHTML.replace("fl_trilho[Base]",
"fl_trilho["+(indice-1)+"]");
    tabelaProduto.innerHTML = tabelaProduto.innerHTML.replace("trilho_Base",
"trilho_"+indice);

    tabelaProduto.innerHTML =
tabelaProduto.innerHTML.replace("trilho_quantidade[Base][[]]",
"trilho_quantidade["+(indice-1)+"][[]]");
    tabelaProduto.innerHTML =
tabelaProduto.innerHTML.replace("trilho_tamanho[Base][[]]",
"trilho_tamanho["+(indice-1)+"][[]]");
    tabelaProduto.innerHTML = tabelaProduto.innerHTML.replace("trilho_tipo[Base][[]]",
"trilho_tipo["+(indice-1)+"][[]]");
    tabelaProduto.innerHTML =
tabelaProduto.innerHTML.replace("trilho_valor[Base][[]]", "trilho_valor["+(indice-
1)+"][[]]");
    tabelaProduto.innerHTML =
tabelaProduto.innerHTML.replace("trilho_desconto[Base][[]]",
"trilho_desconto["+(indice-1)+"][[]]");

```

```

    tabelaProduto.innerHTML =
tabelaProduto.innerHTML.replace("totalPorta[Base][Base]", "totalPorta["+(indice-
1)+"][0]");
    tabelaProduto.innerHTML =
tabelaProduto.innerHTML.replace("totalPorta[Base][Base]", "totalPorta["+(indice-
1)+"][0]");
    tabelaProduto.innerHTML =
tabelaProduto.innerHTML.replace("totalProduto[Base]", "totalProduto["+(indice-
1)+"]");
    tabelaProduto.innerHTML =
tabelaProduto.innerHTML.replace("totalProduto[Base]", "totalProduto["+(indice-
1)+"]");

    while (tabelaProduto.innerHTML.indexOf("[Base][Base]") != -1) {
        tabelaProduto.innerHTML = tabelaProduto.innerHTML.replace("[Base][Base]",
["+(indice-1)+"][0]");
    }
    while (tabelaProduto.innerHTML.indexOf("indiceProduto") != -1) {
        tabelaProduto.innerHTML =
tabelaProduto.innerHTML.replace("indiceProduto", indice);
    }
    while (tabelaProduto.innerHTML.indexOf("tabelaTrilho_Base") != -1) {
        tabelaProduto.innerHTML =
tabelaProduto.innerHTML.replace("tabelaTrilho_Base", "tabelaTrilho_"+indice);
    }
    while (tabelaProduto.innerHTML.indexOf("tabelaPorta_Base") != -1) {
        tabelaProduto.innerHTML =
tabelaProduto.innerHTML.replace("tabelaPorta_Base", "tabelaPorta_"+indice);
    }
    while (tabelaProduto.innerHTML.indexOf("tabelaPerfil_Base") != -1) {
        tabelaProduto.innerHTML =
tabelaProduto.innerHTML.replace("tabelaPerfil_Base", "tabelaPerfil_"+indice+"_1");
    }
    while (tabelaProduto.innerHTML.indexOf("tabelaMaterial_Base") != -1) {
        tabelaProduto.innerHTML =
tabelaProduto.innerHTML.replace("tabelaMaterial_Base",
"tabelaMaterial_"+indice+"_1");
    }
    while (tabelaProduto.innerHTML.indexOf("tabelaPuxador_Base") != -1) {
        tabelaProduto.innerHTML =
tabelaProduto.innerHTML.replace("tabelaPuxador_Base",
"tabelaPuxador_"+indice+"_1");
    }
    while (tabelaProduto.innerHTML.indexOf("tabelaRoldana_Base") != -1) {
        tabelaProduto.innerHTML =
tabelaProduto.innerHTML.replace("tabelaRoldana_Base",
"tabelaRoldana_"+indice+"_1");
    }

```

```

    tabelaProduto.innerHTML = tabelaProduto.innerHTML.replace("Produto Base",
"Produto "+indice);
    tabelaProduto.innerHTML = tabelaProduto.innerHTML.replace("produto_Base",
"produto_"+indice);
    tabelaProduto.innerHTML = tabelaProduto.innerHTML.replace("produto_Base",
"produto_"+indice);

```

```

    tabelaProduto.innerHTML = tabelaProduto.innerHTML.replace("Porta Base",
"Porta 1");
    tabelaProduto.innerHTML = tabelaProduto.innerHTML.replace("porta_Base",
"porta_"+indice+"_1");
    tabelaProduto.innerHTML = tabelaProduto.innerHTML.replace("porta_Base",
"porta_"+indice+"_1");
}

```

```

function deletaProduto(tableID) {
    try {
        var table = document.getElementById(tableID);
        var rowCount = table.rows.length;
        for (var i=0; i<rowCount; i=i+2) {
            var row = table.rows[i];
            var chkbox = row.cells[0].childNodes[2].thead.rows[0].cells[0].childNodes[2];
            if (null != chkbox && true == chkbox.checked) {
                if (i <= 1) {
                    clearRow(row);
                } else {
                    // excluindo o objeto que foi selecionado
                    table.deleteRow(i);
                    rowCount--;
                    i--;
                    // excluindo linha da tabela que é um espaço (separador)
                    table.deleteRow(i);
                    rowCount--;
                    i--;
                }
            }
        }
    } catch(e) {
        alert(e);
    }
}

```

```

function duplicaPorta(tableID, indiceProduto) {
    table = document.getElementById(tableID);
    tableBase = document.getElementById('tabelaPorta_Base');

    // insere uma linha antes do objeto duplicado
    var tr = document.createElement("tr");
    var td = document.createElement("td");
    var txt = document.createTextNode(String.fromCharCode(160));

```

```

td.appendChild(txt);
tr.appendChild(td);
table.tBodies[0].appendChild(tr);

// duplica a linha
newRow = tableBase.tBodies[0].rows[0].cloneNode(true);
//clearRow(newRow);
table.tBodies[0].appendChild(newRow);

// procurar pelos ids e altera seus valores
qtdeLinhas = table.tBodies[0].rows.length;
ultimaLinha = qtdeLinhas - 1;
penultimaLinha = qtdeLinhas - 3;
identificador =
table.tBodies[0].rows[penultimaLinha].cells[0].childNodes[1].tBodies[0].rows[0].id;
//porta_1_1
indice = Number(identificador.substring(identificador.indexOf("_", 6)+1)) + 1;
tabelaPorta = table.tBodies[0].rows[ultimaLinha].cells[0].childNodes[1];

tabelaPorta.innerHTML = tabelaPorta.innerHTML.replace("totalPorta[Base][Base]",
"totalPorta["+indiceProduto-1+"]["+indice-1+"]");
tabelaPorta.innerHTML = tabelaPorta.innerHTML.replace("totalPorta[Base][Base]",
"totalPorta["+indiceProduto-1+"]["+indice-1+"]");

while (tabelaPorta.innerHTML.indexOf("[Base][Base]") != -1) {
    tabelaPorta.innerHTML = tabelaPorta.innerHTML.replace("[Base][Base]",
["+indiceProduto-1+"]["+indice-1+"]");
}
while (tabelaPorta.innerHTML.indexOf("tabelaPerfil_Base") != -1) {
    tabelaPorta.innerHTML = tabelaPorta.innerHTML.replace("tabelaPerfil_Base",
"tabelaPerfil_"+indiceProduto+"_"+indice);
}
while (tabelaPorta.innerHTML.indexOf("tabelaMaterial_Base") != -1) {
    tabelaPorta.innerHTML =
tabelaPorta.innerHTML.replace("tabelaMaterial_Base",
"tabelaMaterial_"+indiceProduto+"_"+indice);
}
while (tabelaPorta.innerHTML.indexOf("tabelaPuxador_Base") != -1) {
    tabelaPorta.innerHTML =
tabelaPorta.innerHTML.replace("tabelaPuxador_Base",
"tabelaPuxador_"+indiceProduto+"_"+indice);
}
while (tabelaPorta.innerHTML.indexOf("tabelaRoldana_Base") != -1) {
    tabelaPorta.innerHTML =
tabelaPorta.innerHTML.replace("tabelaRoldana_Base",
"tabelaRoldana_"+indiceProduto+"_"+indice);
}
tabelaPorta.innerHTML = tabelaPorta.innerHTML.replace("Porta Base", "Porta
"+indice);

```

```

    tabelaPorta.innerHTML = tabelaPorta.innerHTML.replace("porta_Base",
"porta_"+indiceProduto+"_"+indice);
    tabelaPorta.innerHTML = tabelaPorta.innerHTML.replace("porta_Base",
"porta_"+indiceProduto+"_"+indice);
}

function deletaPorta(tableID) {
    try {
        var table = document.getElementById(tableID);
        var rowCount = table.rows.length;
        for (var i=0; i<rowCount; i=i+2) {
            var row = table.rows[i];
            var chkbox = row.cells[0].childNodes[1].thead.rows[0].cells[0].childNodes[2];
            if (null != chkbox && true == chkbox.checked) {
                if (i <= 1) {
                    clearRow(row);
                } else {
                    // excluindo o objeto que foi selecionado
                    table.deleteRow(i);
                    rowCount--;
                    i--;
                    // excluindo linha da tabela que é um espaço (separador)
                    table.deleteRow(i);
                    rowCount--;
                    i--;
                }
            }
        }
    } catch(e) {
        alert(e);
    }
}

```

```

function duplicateRow(tableID) {
    table = document.getElementById(tableID);
    newRow = table.tBodies[0].rows[1].cloneNode(true);
    clearRow(newRow);
    table.tBodies[0].appendChild(newRow);
}

```

```

function deleteRow(tableID) {
    try {
        var table = document.getElementById(tableID);
        var rowCount = table.rows.length;

        for (var i=0; i<rowCount; i++) {
            var row = table.rows[i];
            var chkbox = row.cells[0].childNodes[0];
            if (null != chkbox && true == chkbox.checked) {
                if (i <= 1) {

```

```

        clearRow(row);
    } else {
        table.deleteRow(i);
        rowCount--;
        i--;
    }
}
}
} catch(e) {
    alert(e);
}
}

function carregarUrl(url) {
    window.location.href=url;
}

function redireciona(url) {
    setTimeout("carregarUrl('"+url+"')", 2000);
}

function imprimir() {
    window.print();
}

function mostraCompras(modos){
    if (modos == "true"){
        document.getElementById("compras").style.display = 'block';
    }else{
        document.getElementById("compras").style.display = 'none';
    }
}

function exibeInfoOrçamento(modos) {
    end = document.getElementById("endereco");
    con = document.getElementById("contato");
    pro = document.getElementById("produtos");

    if (modos == true) {
        end.style.display = 'block';
        con.style.display = 'block';
        pro.style.display = 'block';
    } else {
        end.style.display = 'none';
        con.style.display = 'none';
        pro.style.display = 'none';
    }
}

function verificaSelecaoCliente(cd_cliente) {

```

```

        if (cd_cliente == "") {
            exibeInfoOrcamento(false);
        } else {
            ajax('/index.php/ajax/retornaDadosCliente/'+cd_cliente, "", "",
'dadosCliente', null, null, 'iconeCarregandoCliente');
            exibeInfoOrcamento(true);
        }
    }
}

function exibeInfoContato(modos) {
    fone = document.getElementById("fone");

    if (modos == false) {
        fone.innerHTML = "";
    }
}

function verificaSelecaoContato(cd_cliente, nr_sequencia) {
    if (nr_sequencia == "") {
        exibeInfoContato(false);
    } else {
        ajax('/index.php/ajax/retornaDadosContatoCliente/'+cd_cliente+'/'+nr_sequencia, "", "", 'fone', null, null, 'iconeCarregandoTelefone');
    }
}

function alternar(identificador, expand_collapse) {
    //alert(identificador);
    //alert(expand_collapse);
    var obj = document.getElementById(identificador);
    if (obj.style.display != 'none') {
        obj.style.display = 'none';
        expand_collapse.src = '/img/expand_collapse_plus.gif';
    } else {
        obj.style.display = "";
        expand_collapse.src = '/img/expand_collapse_minus.gif';
    }
}

function exibeInfoTrilho(modos, indiceProduto) {
    tr = document.getElementById("trilho_"+indiceProduto);

    if (modos == true) {
        tr.style.display = 'block';
    } else {
        tr.style.display = 'none';
    }
}

```

```

function exibelInfoEspacoLivre(modos, indiceProduto) {
    el = document.getElementById("espaco_livre_"+indiceProduto);

    if (modos == true) {
        el.style.display = 'table-row';
    } else {
        el.style.display = 'none';
    }
}

function trocarSenha(obj) {
    p = document.getElementById("password");
    p2 = document.getElementById("password2");
    if (obj.checked) {
        p.value = "";
        p.disabled = false;
        p2.value = "";
        p2.disabled = false;
        p.className = "";
        p2.className = "";
        p.focus();
    } else {
        p.value = "";
        p.disabled = true;
        p2.value = "";
        p2.disabled = true;
        p.className = "campoDesabilitado";
        p2.className = "campoDesabilitado";
    }
}

function somenteNumeros(evento) {
    var charCode = (evento.which) ? evento.which : evento.keyCode;
    if (charCode > 31 && (charCode < 48 || charCode > 57)) {
        return false;
    }
    return true;
}

function float2moeda(num) {
    x = 0;
    if(num<0) {
        num = Math.abs(num);
        x = 1;
    }
    if(isNaN(num)) num = "0";
    cents = Math.floor((num*100+0.5)%100);

    num = Math.floor((num*100+0.5)/100).toString();

```

```

if(cents < 10) cents = "0" + cents;
for (var i = 0; i < Math.floor((num.length-(1+i))/3); i++)
    num = num.substring(0,num.length-(4*i+3))+ '.'
        +num.substring(num.length-(4*i+3));
ret = num + ',' + cents;
if (x == 1) ret = ' - ' + ret;
return ret;

```

```

}

```

```

function moeda2float(moeda){
    moeda = moeda.replace(".", "");
    moeda = moeda.replace(",", ".");
    return parseFloat(moeda);
}

```

### **\\js\\menu.js**

```

function initMenu() {
    $('#menu ul').hide();

    if (location.href.search(/usuario/i) != -1) {
        $('#menu ul:eq(0)').show();
    } else if (location.href.search(/fornecedor/i) != -1) {
        $('#menu ul:eq(1)').show();
    } else if (location.href.search(/importacao/i) != -1) {
        $('#menu ul:eq(2)').show();
    } else if (location.href.search(/ajuda/i) != -1) {
        $('#menu ul:eq(3)').show();
    } else {
        $('#menu ul:eq(0)').show();
    }

    $('#menu li a').click(
        function() {
            var checkElement = $(this).next();
            if ((checkElement.is('ul')) && (checkElement.is(':visible'))) {
                return false;
            }
            if ((checkElement.is('ul')) && (!checkElement.is(':visible'))) {
                $('#menu ul:visible').slideUp('normal');
                checkElement.slideDown('normal');
                return false;
            }
        }
    );
}

$(document).ready(function() {
    initMenu();
});

```

**\\system\application\config\autoload.php**

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
/*
```

```
|-----
| AUTO-LOADER
|-----
```

```
| This file specifies which systems should be loaded by default.
```

```
| In order to keep the framework as light-weight as possible only the
| absolute minimal resources are loaded by default. For example,
| the database is not connected to automatically since no assumption
| is made regarding whether you intend to use it. This file lets
| you globally define which systems you would like loaded with every
| request.
```

```
|-----
| Instructions
|-----
```

```
| These are the things you can load automatically:
```

- ```
| 1. Libraries
| 2. Helper files
| 3. Plugins
| 4. Custom config files
| 5. Language files
| 6. Models
```

```
|
*/
```

```
/*
```

```
|-----
Auto-load Libraries
```

```
| These are the classes located in the system/libraries folder
| or in your system/application/libraries folder.
```

```
| Prototype:
```

```
|     $autoload['libraries'] = array('database', 'session', 'xmlrpc');
```

```
*/
```

```
$autoload['libraries'] = array('DX_Auth', 'nusoap', 'Funcoes_Genericas', 'database',
'form_validation');
```

```
/*
```

```
|-----
| Auto-load Helper Files
```

```

|-----
| Prototype:
|
|     $autoload['helper'] = array('url', 'file');
| */

```

```
$autoload['helper'] = array('url', 'form');
```

```

/*
|-----
Auto-load Plugins
Prototype:
$autoload['plugin'] = array('captcha', 'js_calendar');
*/

```

```
$autoload['plugin'] = array();
```

```

/*
|-----
Auto-load Config files
Prototype:
$autoload['config'] = array('config1', 'config2');
NOTE: This item is intended for use ONLY if you have created custom
config files. Otherwise, leave it blank.
*/

```

```
$autoload['config'] = array();
```

```

/*
|-----
Auto-load Language files
Prototype:
$autoload['language'] = array('lang1', 'lang2');
NOTE: Do not include the "_lang" part of your file. For example
"codeigniter_lang.php" would be referenced as array('codeigniter');
*/

```

```
$autoload['language'] = array();
```

```
/*
|-----
Auto-load Models
Prototype:
$autoload['model'] = array('model1', 'model2');
*/
```

```
$autoload['model'] = array();
```

```
/* End of file autoload.php */
```

```
/* Location: ./system/application/config/autoload.php */
```

### **system\application\config\config.php**

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
```

```
/*
|-----
| Base Site URL
|-----
|
| URL to your CodeIgniter root. Typically this will be your base URL,
| WITH a trailing slash:
|
|     http://example.com/
|
*/
$config['base_url'] = "http://sipec";

/*
|-----
| Index File
|-----
|
| Typically this will be your index.php file, unless you've renamed it to
| something else. If you are using mod_rewrite to remove the page set this
| variable so that it is blank.
|
*/
$config['index_page'] = "index.php";

/*
|-----
| URI PROTOCOL
|-----
```

```

|
| This item determines which server global should be used to retrieve the
| URI string. The default setting of "AUTO" works for most servers.
| If your links do not seem to work, try one of the other delicious flavors:
|
| 'AUTO'                Default - auto detects
| 'PATH_INFO'           Uses the PATH_INFO
| 'QUERY_STRING'        Uses the QUERY_STRING
| 'REQUEST_URI'         Uses the REQUEST_URI
| 'ORIG_PATH_INFO'      Uses the ORIG_PATH_INFO
|
| */
$config['uri_protocol']   = "AUTO";

/*
|-----|
| URL suffix
|-----|
|
| This option allows you to add a suffix to all URLs generated by CodeIgniter.
| For more information please see the user guide:
|
| http://codeigniter.com/user\_guide/general/urls.html
| */

$config['url_suffix'] = "";

/*
|-----|
| Default Language
|-----|
|
| This determines which set of language files should be used. Make sure
| there is an available translation if you intend to use something other
| than english.
|
| */
$config['language'] = "pt-br";

/*
|-----|
| Default Character Set
|-----|
|
| This determines which character set is used by default in various methods
| that require a character set to be provided.
|
| */
$config['charset'] = "UTF-8";

```

```

/*
|-----
| Enable/Disable System Hooks
|-----
|
| If you would like to use the "hooks" feature you must enable it by
| setting this variable to TRUE (boolean). See the user guide for details.
|
*/
$config['enable_hooks'] = FALSE;

/*
|-----
| Class Extension Prefix
|-----
|
| This item allows you to set the filename/classname prefix when extending
| native libraries. For more information please see the user guide:
|
| http://codeigniter.com/user\_guide/general/core\_classes.html
| http://codeigniter.com/user\_guide/general/creating\_libraries.html
|
*/
$config['subclass_prefix'] = 'MY_';

/*
|-----
| Allowed URL Characters
|-----
|
| This lets you specify with a regular expression which characters are permitted
| within your URLs. When someone tries to submit a URL with disallowed
| characters they will get a warning message.
|
| As a security measure you are STRONGLY encouraged to restrict URLs to
| as few characters as possible. By default only these are allowed: a-z 0-9~%.:_-
|
| Leave blank to allow all characters -- but only if you are insane.
|
| DO NOT CHANGE THIS UNLESS YOU FULLY UNDERSTAND THE
| REPERCUSSIONS!!
|
*/
$config['permitted_uri_chars'] = 'a-z 0-9~%.:_-';

/*
|-----

```

## | Enable Query Strings

---

| By default CodeIgniter uses search-engine friendly segment based URLs:  
| `example.com/who/what/where/`

| You can optionally enable standard query string based URLs:  
| `example.com?who=me&what=something&where=here`

| Options are: TRUE or FALSE (boolean)

| The other items let you set the query string "words" that will  
| invoke your controllers and its functions:  
| `example.com/index.php?c=controller&m=function`

| Please note that some of the helpers won't work as expected when  
| this feature is enabled, since CodeIgniter is designed primarily to  
| use segment based URLs.

|  
|\*/

```
$config['enable_query_strings'] = FALSE;
$config['controller_trigger']   = 'c';
$config['function_trigger']    = 'm';
$config['directory_trigger']   = 'd'; // experimental not currently in use
```

|  
|/\*

## | Error Logging Threshold

---

| If you have enabled error logging, you can set an error threshold to  
| determine what gets logged. Threshold options are:  
| You can enable error logging by setting a threshold over zero. The  
| threshold determines what gets logged. Threshold options are:

- | 0 = Disables logging, Error logging TURNED OFF
- | 1 = Error Messages (including PHP errors)
- | 2 = Debug Messages
- | 3 = Informational Messages
- | 4 = All Messages

| For a live site you'll usually only enable Errors (1) to be logged otherwise  
| your log files will fill up very fast.

|  
|\*/

```
$config['log_threshold'] = 0;
```

|  
|/\*

## | Error Logging Directory Path

---

```

|-----
|
| Leave this BLANK unless you would like to set something other than the default
| system/logs/ folder. Use a full server path with trailing slash.
|
|*/
$config['log_path'] = "";

/*
|-----
| Date Format for Logs
|-----
|
| Each item that is logged has an associated date. You can use PHP date
| codes to set your own date formatting
|
|*/
$config['log_date_format'] = 'Y-m-d H:i:s';

/*
|-----
| Cache Directory Path
|-----
|
| Leave this BLANK unless you would like to set something other than the default
| system/cache/ folder. Use a full server path with trailing slash.
|
|*/
$config['cache_path'] = "";

/*
|-----
| Encryption Key
|-----
|
| If you use the Encryption class or the Sessions class with encryption
| enabled you MUST set an encryption key. See the user guide for info.
|
|*/
$config['encryption_key'] = "";

/*
|-----
| Session Variables
|-----
|
| 'session_cookie_name' = the name you want for the cookie
| 'encrypt_sess_cookie' = TRUE/FALSE (boolean). Whether to encrypt the cookie
| 'session_expiration' = the number of SECONDS you want the session to last.
| by default sessions last 7200 seconds (two hours). Set to zero for no expiration.

```

| 'time\_to\_update' = how many seconds between CI refreshing Session Information

```

|
| */
$config['sess_cookie_name'] = 'ci_session';
$config['sess_expiration'] = 7200;
$config['sess_encrypt_cookie'] = FALSE;
$config['sess_use_database'] = true;
$config['sess_table_name'] = 'ci_sessions';
$config['sess_match_ip'] = FALSE;
$config['sess_match_useragent'] = TRUE;
$config['sess_time_to_update'] = 300;

```

```

/*
|-----|
| Cookie Related Variables
|-----|
|
| 'cookie_prefix' = Set a prefix if you need to avoid collisions
| 'cookie_domain' = Set to .your-domain.com for site-wide cookies
| 'cookie_path' = Typically will be a forward slash
|

```

```

*/
$config['cookie_prefix'] = "";
$config['cookie_domain'] = "";
$config['cookie_path'] = "/";

```

```

/*
|-----|
| Global XSS Filtering
|-----|
|
| Determines whether the XSS filter is always active when GET, POST or
| COOKIE data is encountered
|

```

```

*/
$config['global_xss_filtering'] = FALSE;

```

```

/*
|-----|
| Output Compression
|-----|
|
| Enables Gzip output compression for faster page loads. When enabled,
| the output class will test whether your server supports Gzip.
| Even if it does, however, not all browsers support compression
| so enable only if you are reasonably sure your visitors can handle it.
|
| VERY IMPORTANT: If you are getting a blank page when compression is enabled
| it

```

| means you are prematurely outputting something to your browser. It could  
 | even be a line of whitespace at the end of one of your scripts. For  
 | compression to work, nothing can be sent before the output buffer is called  
 | by the output class. Do not "echo" any values with compression enabled.

```

  |
  */
$config['compress_output'] = FALSE;

```

```

/*
|-----|
| Master Time Reference
|-----|
|
| Options are "local" or "gmt". This pref tells the system whether to use
| your server's local time as the master "now" reference, or convert it to
| GMT. See the "date helper" page of the user guide for information
| regarding date handling.
|
*/

```

```

$config['time_reference'] = 'local';

```

```

/*
|-----|
| Rewrite PHP Short Tags
|-----|
|
| If your PHP installation does not have short tag support enabled CI
| can rewrite the tags on-the-fly, enabling you to utilize that syntax
| in your view files. Options are TRUE or FALSE (boolean)
|
*/

```

```

$config['rewrite_short_tags'] = FALSE;

```

```

/*
|-----|
| Reverse Proxy IPs
|-----|
|
| If your server is behind a reverse proxy, you must whitelist the proxy IP
| addresses from which CodeIgniter should trust the HTTP_X_FORWARDED_FOR
| header in order to properly identify the visitor's IP address.
| Comma-delimited, e.g. '10.0.1.200,10.0.1.201'
|
*/

```

```

$config['proxy_ips'] = "";

```

```

/* End of file config.php */

```

```

/* Location: ./system/application/config/config.php */

```

## **\\system\application\config\database.php**

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
/*
```

```
|-----|
| DATABASE CONNECTIVITY SETTINGS
|-----|
```

```
| This file will contain the settings needed to access your database.
```

```
| For complete instructions please consult the "Database Connection"
| page of the User Guide.
```

```
|-----|
| EXPLANATION OF VARIABLES
|-----|
```

```
| ['hostname'] The hostname of your database server.
```

```
| ['username'] The username used to connect to the database
```

```
| ['password'] The password used to connect to the database
```

```
| ['database'] The name of the database you want to connect to
```

```
| ['dbdriver'] The database type. ie: mysql. Currently supported:
```

```
mysql, mysqli, postgres, odbc, mssql, sqlite, oci8
```

```
| ['dbprefix'] You can add an optional prefix, which will be added
```

```
to the table name when using the Active Record class
```

```
| ['pconnect'] TRUE/FALSE - Whether to use a persistent connection
```

```
| ['db_debug'] TRUE/FALSE - Whether database errors should be displayed.
```

```
| ['cache_on'] TRUE/FALSE - Enables/disables query caching
```

```
| ['cachedir'] The path to the folder where cache files should be stored
```

```
| ['char_set'] The character set used in communicating with the database
```

```
| ['dbcollat'] The character collation used in communicating with the database
```

```
| The $active_group variable lets you choose which connection group to
| make active. By default there is only one group (the "default" group).
```

```
| The $active_record variables lets you determine whether or not to load
| the active record class
```

```
*/
```

```
$active_group = "default";
```

```
$active_record = TRUE;
```

```
$db['default']['hostname'] = "localhost";
```

```
$db['default']['username'] = "root";
```

```
$db['default']['password'] = "";
```

```
$db['default']['database'] = "oscommerce";
```

```
$db['default']['dbdriver'] = "mysql";
```

```
$db['default']['dbprefix'] = "";
```

```
$db['default']['pconnect'] = TRUE;
```

```
$db['default']['db_debug'] = TRUE;
```

```
$db['default']['cache_on'] = FALSE;
```

```

$db['default']['cachedir'] = "";
//$db['default']['char_set'] = "utf8";
//$db['default']['dbcollat'] = "utf8_general_ci";
$db['default']['char_set'] = "latin1";
$db['default']['dbcollat'] = "latin1_swedish_ci";

```

```

/* End of file database.php */
/* Location: ./system/application/config/database.php */

```

### **\system\application\config\dx\_auth.php**

```

<?php if (!defined('BASEPATH')) exit('No direct script access allowed');

```

```

/*
| -----
| DX Auth Config
| -----
*/

```

```

/*
| -----
| Website details
| -----
|
| These details are used in email sent by DX Auth library.
|
*/

```

```

$config['DX_website_name'] = 'SIPEC';
$config['DX_webmaster_email'] = 'leonardofln@gmail.com';

```

```

/*
| -----
| Database table
| -----
|
| Determines table that used by DX Auth.
|
| 'DX_table_prefix' allows you to specify table prefix that will be use by the rest of the
table.
|
| For example specifying 'DX_' in 'DX_table_prefix' and 'users' in 'DX_users_table',
| will make DX Auth user 'DX_users' as users table.
|
*/

```

```

$config['DX_table_prefix'] = "";
$config['DX_users_table'] = 'users';
$config['DX_user_profile_table'] = 'user_profile';
$config['DX_user_temp_table'] = 'user_temp';

```

```
$config['DX_user_autologin'] = 'user_autologin';
$config['DX_roles_table'] = 'roles';
$config['DX_permissions_table'] = 'permissions';
$config['DX_login_attempts_table'] = 'login_attempts';
```

```
/*
|-----
| Password salt
|-----
|
| You can add major salt to be hashed with password.
| For example, you can get salt from here: https://www.grc.com/passwords.htm
|
| Note:
|
| Keep in mind that if you change the salt value after user registered,
| user that previously registered cannot login anymore.
|
*/
```

```
$config['DX_salt'] = '';
```

```
/*
|-----
| Registration related settings
|-----
|
| 'DX_email_activation' = Requires user to activate their account using email after
registration.
| 'DX_email_activation_expire' = Time before users who don't activate their account
getting deleted from database. Default is 48 Hours (60*60*24*2).
| 'DX_email_account_details' = Email account details after registration, only if
'DX_email_activation' is FALSE.
|
*/
```

```
$config['DX_email_activation'] = FALSE;
$config['DX_email_activation_expire'] = 60*60*24*2;
$config['DX_email_account_details'] = FALSE;
```

```
/*
|-----
| Login settings
|-----
|
| 'DX_login_using_username' = Determine if user can use username in username
field to login.
| 'DX_login_using_email' = Determine if user can use email in username field to login.
|
| You have to set at least one of settings above to TRUE.
```

```

|
| 'DX_login_record_ip' = Determine if user IP address should be recorded in
| database when user login.
| 'DX_login_record_time' = Determine if time should be recorded in database when
| user login.
|
*/

$config['DX_login_using_username'] = TRUE;
$config['DX_login_using_email'] = TRUE;
$config['DX_login_record_ip'] = TRUE;
$config['DX_login_record_time'] = TRUE;

/*
|-----
| Auto login settings
|-----
|
| 'DX_autologin_cookie_name' = Determine auto login cookie name.
| 'DX_autologin_cookie_life' = Determine auto login cookie life before expired. Default
| is 2 months (60*60*24*31*2).
|
*/

$config['DX_autologin_cookie_name'] = 'autologin';
$config['DX_autologin_cookie_life'] = 0;

/*
|-----
| Login attempts
|-----
|
| 'DX_count_login_attempts' = Determine if DX Auth should count login attempt when
| user failed to login.
| 'DX_max_login_attempts' = Determine max login attempt before function
| is_login_attempt_exceeded() returning TRUE.
|
*/

$config['DX_count_login_attempts'] = TRUE;
$config['DX_max_login_attempts'] = 1;

/*
|-----
| Forgot password settings
|-----
|
| 'DX_forgot_password_expire' = Time before forgot password key become invalid.
| Default is 15 minutes (900 seconds).
|

```

```
*/
```

```
$config['DX_forgot_password_expire'] = 900;
```

```
/*
```

```
|-----  
| Captcha  
|-----
```

```
|  
| You can set captcha that created by DX Auth library in here.  
| 'DX_captcha_directory' = Directory where the captcha will be created.  
| 'DX_captcha_fonts_path' = Font in this directory will be used when creating  
| captcha.  
| 'DX_captcha_font_size' = Font size when writing text to captcha. Leave blank for  
| random font size.  
| 'DX_captcha_grid' = Show grid in created captcha.  
| 'DX_captcha_expire' = Life time of created captcha before expired, default is 3  
| minutes (180 seconds).  
| 'DX_captcha_expire' = Determine captcha case sensitive or not.
```

```
|  
*/
```

```
$config['DX_captcha_path'] = './captcha/';  
$config['DX_captcha_fonts_path'] = $config['DX_captcha_path'].'fonts';  
$config['DX_captcha_width'] = 320;  
$config['DX_captcha_height'] = 95;  
$config['DX_captcha_font_size'] = "";  
$config['DX_captcha_grid'] = TRUE;  
$config['DX_captcha_expire'] = 180;  
$config['DX_captcha_case_sensitive'] = TRUE;
```

```
/*
```

```
|-----  
| reCAPTCHA  
|-----
```

```
|  
| If you are planning to use reCAPTCHA function, you have to set reCAPTCHA key  
| here  
| You can get the key by registering at http://recaptcha.net
```

```
|  
*/
```

```
$config['DX_recaptcha_public_key'] = "";  
$config['DX_recaptcha_private_key'] = "";
```

```
/*
```

```
|-----  
| URI  
|-----
```

```

|
| Determines URI that used for redirecting in DX Auth library.
| 'DX_deny_uri' = Forbidden access URI.
| 'DX_login_uri' = Login form URI.
| 'DX_activate_uri' = Activate user URI.
| 'DX_reset_password_uri' = Reset user password URI.
|
| These value can be accessed from DX Auth library variable, by removing 'DX_'
string.
| For example you can access 'DX_deny_uri' by using $this->dx_auth->deny_uri in
controller.
|
*/

$config['DX_deny_uri'] = '/auth/deny/';
$config['DX_login_uri'] = '/auth/login/';
$config['DX_banned_uri'] = '/auth/banned/';
$config['DX_activate_uri'] = '/auth/activate/';
$config['DX_reset_password_uri'] = '/auth/reset_password/';

/*
|-----
| Helper configuration
|-----
|
| Configuration below is actually not used in function in DX_Auth library.
|     They just used to help you coding more easily in controller.
|     You can set it to blank if you don't need it, or even delete it.
|
| However they can be accessed from DX Auth library variable, by removing 'DX_'
string.
| For example you can access 'DX_register_uri' by using $this->dx_auth->register_uri
in controller.
|
*/

// Registration
$config['DX_allow_registration'] = TRUE;
$config['DX_captcha_registration'] = FALSE;

// Login
$config['DX_captcha_login'] = FALSE;

// URI Locations
$config['DX_logout_uri'] = '/auth/logout/';
$config['DX_register_uri'] = '/auth/register/';
$config['DX_forgot_password_uri'] = '/auth/forgot_password/';
$config['DX_change_password_uri'] = '/auth/change_password/';
$config['DX_cancel_account_uri'] = '/auth/cancel_account/';

```

```

// Forms view
$config['DX_login_view'] = 'tela_login';
$config['DX_register_view'] = 'auth/register_form';
$config['DX_forgot_password_view'] = 'auth/forgot_password_form';
$config['DX_change_password_view'] = 'auth/change_password_form';
$config['DX_cancel_account_view'] = 'auth/cancel_account_form';

// Pages view
$config['DX_deny_view'] = 'auth/general_message';
$config['DX_banned_view'] = 'auth/general_message';
$config['DX_logged_in_view'] = 'auth/general_message';
$config['DX_logout_view'] = 'auth/general_message';

$config['DX_register_success_view'] = 'auth/general_message';
$config['DX_activate_success_view'] = 'auth/general_message';
$config['DX_forgot_password_success_view'] = 'auth/general_message';
$config['DX_reset_password_success_view'] = 'auth/general_message';
$config['DX_change_password_success_view'] = 'auth/general_message';

$config['DX_register_disabled_view'] = 'auth/general_message';
$config['DX_activate_failed_view'] = 'auth/general_message';
$config['DX_reset_password_failed_view'] = 'auth/general_message';

?>

```

### **\\system\application\config\routes.php**

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
/*
|-----
| URI ROUTING
|-----
| This file lets you re-map URI requests to specific controller functions.
|
| Typically there is a one-to-one relationship between a URL string
| and its corresponding controller class/method. The segments in a
| URL normally follow this pattern:
|
|     example.com/class/method/id/
|
| In some instances, however, you may want to remap this relationship
| so that a different class/function is called than the one
| corresponding to the URL.
|
| Please see the user guide for complete details:
|
|     http://codeigniter.com/user_guide/general/routing.html
|-----
| RESERVED ROUTES

```

```

|-----
|
| There are two reserved routes:
|
|     $route['default_controller'] = 'welcome';
|
| This route indicates which controller class should be loaded if the
| URI contains no data. In the above example, the "welcome" class
| would be loaded.
|
|     $route['scaffolding_trigger'] = 'scaffolding';
|
| This route lets you set a "secret" word that will trigger the
| scaffolding feature for added security. Note: Scaffolding must be
| enabled in the controller in which you intend to use it. The reserved
| routes must come before any wildcard or regular expression routes.
|
| */

```

```

$route['default_controller'] = "auth";
$route['scaffolding_trigger'] = "";
$route['webservices/wSDL'] = "webservices/index/wSDL";

```

```

/* End of file routes.php */

```

```

/* Location: ./system/application/config/routes.php */

```

### **\\system\application\controllers\ajax.php**

```

<?php

```

```

header("Content-Type: text/html; charset=ISO-8859-1",true);

```

```

// aguarda pelo menos meio segundo nas requisições ajax

```

```

// para que o usuário perceba que está sendo feito alguma coisa...

```

```

usleep(500000);

```

```

class Ajax extends Controller {

```

```

    function Ajax() {
        parent::Controller();
        $this->load->model('importacao_model');
        $this->load->model('fornecedor_model');
        $this->load->model('produto_model');
    }

```

```

    function ativarFornecedor($cd_fornecedor) {
        $this->fornecedor_model->ativarFornecedor($cd_fornecedor);

        $retorno = '';

```

```

    echo $retorno;
}

function inativarFornecedor($cd_fornecedor) {
    $this->fornecedor_model->inativarFornecedor($cd_fornecedor);

    $retorno = '';

    echo $retorno;
}

function executarImportacaoPasso1() {
// só pode executar a importação uma vez por dia...
    if ($this->importacao_model->podeExecutarImportacao()) {
        // excluir todos os produtos da tabela sipec_produto_fornecedor
        $this->produto_model->excluirProdutosFornecedor();

        $retorno = 'OK';
    } else {
        $retorno = 'Não é possível executar mais de uma importação no
mesmo dia!';
    }

    echo $retorno;
}

function executarImportacaoPasso2() {
// selecionar os fornecedores ativos que estão cadastrados no sistema
    $fornecedores = $this->fornecedor_model-
>buscaFornecedorPeloFiltro(null, null, 1);
    foreach ($fornecedores as $fornecedor) {
        // acessar os webservices de cada fornecedor para receber a
lista de produtos
        $this->soapclient = new nusoap_client($fornecedor->de_url_ws);
        $result = $this->soapclient->call('listaProdutos');

        // converter a lista de produtos XML e uma estrutura de dados
em PHP
        $produtos = $this->funcoes_genericas->xml2array($result);

        // gravar a lista de produtos no banco de dados do sistema
        $this->importacao_model->gravarProdutosBD($produtos,
$fornecedor->cd_fornecedor);
    }

    $retorno = '';
}

```

```

    echo $retorno;
}

    function executarImportacaoPasso3() {
    // atualiza o campo fl_mais_barato de cada produto que possui o menor preço
        $this->produto_model->defineProdutosFornecedorMaisBarato();

    $retorno = '';

    echo $retorno;
}

    function executarImportacaoPasso4() {
    // armazena os dados no histórico do sistema
        $this->importacao_model->armazenaDadosNoHistorico();

    $retorno = '';

    echo $retorno;
}

    function executarImportacaoPasso5() {
    // executa a carga dos produtos no sistema de e-commerce
        $this->importacao_model->cargaProdutosSistemaEcommerce();

    $retorno = '';

    echo $retorno;
}

    function executarImportacaoPasso6() {

    $retorno = '';

    echo $retorno;
}
}
?>

```

### **\\system\application\controllers\auth.php**

```

<?php
class Auth extends Controller
{
    // Used for registering and changing password form validation
    var $min_username = 4;
    var $max_username = 20;
    var $min_password = 4;
    var $max_password = 20;
}

```

```

function Auth()
{
    parent::Controller();

    $this->load->library('Form_validation');
    $this->load->library('DX_Auth');

    $this->form_validation->set_error_delimiters('<div class="error">',
'</div>');

    $this->load->helper('url');
    $this->load->helper('form');
}

function index()
{
    $this->login();
}

/* Callback function */

function username_check($username)
{
    $result = $this->dx_auth->is_username_available($username);
    if ( ! $result)
    {
        $this->form_validation->set_message('username_check',
'Username already exist. Please choose another username.');
```

```

        {
            // Will replace this error msg with $lang
            $this->form_validation->set_message('captcha_check', 'Your
confirmation code has expired. Please try again. ');
            $result = FALSE;
        }
        elseif ( ! $this->dx_auth->is_captcha_match($code))
        {
            $this->form_validation->set_message('captcha_check', 'Your
confirmation code does not match the one in the image. Try again. ');

            $result = FALSE;
        }

        return $result;
    }

    function recaptcha_check()
    {
        $result = $this->dx_auth->is_recaptcha_match();
        if ( ! $result)
        {
            $this->form_validation->set_message('recaptcha_check', 'Your
confirmation code does not match the one in the image. Try again. ');
        }

        return $result;
    }

    /* End of Callback function */

    function login()
    {
        if ( ! $this->dx_auth->is_logged_in())
        {
            $val = $this->form_validation;

            // Set form validation rules
            $val->set_rules('username', 'Usuário', 'trim|required|xss_clean');
            $val->set_rules('password', 'Senha', 'trim|required|xss_clean');
            //$val->set_rules('remember', 'Remember me', 'integer');

            // Set captcha rules if login attempts exceed max attempts in
config
            //if ($this->dx_auth->is_max_login_attempts_exceeded())
            //{
            //    $val->set_rules('captcha', 'Confirmation Code',
'trim|required|xss_clean|callback_captcha_check');
            //}

```

```

        if ($val->run() AND $this->dx_auth->login($val-
>set_value('username'), $val->set_value('password')/*, $val-
>set_value('remember')*/))
        {
            // Redirect to homepage
            redirect("", 'location');
        }
        else
        {
            // Check if the user is failed logged in because user is
banned user or not
            if ($this->dx_auth->is_banned())
            {
                // Redirect to banned uri
                $this->dx_auth->deny_access('banned');
            }
            else
            {
                // Default is we don't show captcha until max login
attempts eceeded
                $data['show_captcha'] = FALSE;

                // Show captcha if login attempts exceed max
attempts in config
                if ($this->dx_auth-
>is_max_login_attempts_exceeded())
                {
                    // Create catpcha

                    $this->dx_auth->captcha();

                    // Set view data to show captcha on view file
                    $data['show_captcha'] = TRUE;
                }

                // Load login page view
                $this->load->view($this->dx_auth->login_view,
$data);
            }
        }
    }
    else
    {
        //$data['auth_message'] = 'You are already logged in.';
        //$this->load->view($this->dx_auth->logged_in_view, $data);
        redirect('inicio', 'location');
    }
}

```

```

function logout() {
    $this->dx_auth->logout();
    redirect("", 'location');
}

function register()
{
    if (! $this->dx_auth->is_logged_in() AND $this->dx_auth-
>allow_registration)
    {
        $val = $this->form_validation;

        // Set form validation rules
        $val->set_rules('username', 'Username',
'trim|required|xss_clean|min_length['.$this->min_username.']]max_length['.$this-
>max_username.']]callback_username_check|alpha_dash');
        $val->set_rules('password', 'Password',
'trim|required|xss_clean|min_length['.$this->min_password.']]max_length['.$this-
>max_password.']]matches[confirm_password]');
        $val->set_rules('confirm_password', 'Confirm Password',
'trim|required|xss_clean');
        $val->set_rules('email', 'Email',
'trim|required|xss_clean|valid_email|callback_email_check');

        if ($this->dx_auth->captcha_registration)
        {
            $val->set_rules('captcha', 'Confirmation Code',
'trim|xss_clean|required|callback_captcha_check');
        }

        // Run form validation and register user if it's pass the validation
        if ($val->run() AND $this->dx_auth->register($val-
>set_value('username'), $val->set_value('password'), $val->set_value('email')))
        {
            // Set success message accordingly
            if ($this->dx_auth->email_activation)
            {
                $data['auth_message'] = 'You have successfully
registered. Check your email address to activate your account.';
            }
            else
            {
                $data['auth_message'] = 'You have successfully
registered. '.anchor(site_url($this->dx_auth->login_uri), 'Login');
            }

            // Load registration success page
            $this->load->view($this->dx_auth-
>register_success_view, $data);
        }
    }
}

```

```

else
{
    // Is registration using captcha
    if ($this->dx_auth->captcha_registration)
    {
        $this->dx_auth->captcha();
    }

    // Load registration page
    $this->load->view($this->dx_auth->register_view);
}
}
elseif ( ! $this->dx_auth->allow_registration)
{
    $data['auth_message'] = 'Registration has been disabled.';
    $this->load->view($this->dx_auth->register_disabled_view,
$data);
}
else
{
    $data['auth_message'] = 'You have to logout first, before
registering.';
    $this->load->view($this->dx_auth->logged_in_view, $data);
}
}

function register_recaptcha()
{
    if ( ! $this->dx_auth->is_logged_in() AND $this->dx_auth-
>allow_registration)
    {
        $val = $this->form_validation;

        // Set form validation rules
        $val->set_rules('username', 'Username',
'trim|required|xss_clean|min_length['.$this->min_username.']]max_length['.$this-
>max_username.']]callback_username_check|alpha_dash');
        $val->set_rules('password', 'Password',
'trim|required|xss_clean|min_length['.$this->min_password.']]max_length['.$this-
>max_password.']]matches[confirm_password]');
        $val->set_rules('confirm_password', 'Confirm Password',
'trim|required|xss_clean');
        $val->set_rules('email', 'Email',
'trim|required|xss_clean|valid_email|callback_email_check');

        // Is registration using captcha
        if ($this->dx_auth->captcha_registration)
        {
            // Set recaptcha rules.

```

```

// IMPORTANT: Do not change 'recaptcha_response_field'
because it's used by reCAPTCHA API,
// This is because the limitation of reCAPTCHA, not DX
Auth library
$val->set_rules('recaptcha_response_field', 'Confirmation
Code', 'trim|xss_clean|required|callback_recaptcha_check');
}

// Run form validation and register user if it's pass the validation
if ($val->run() AND $this->dx_auth->register($val-
>set_value('username'), $val->set_value('password'), $val->set_value('email')))
{
    // Set success message accordingly
    if ($this->dx_auth->email_activation)
    {
        $data['auth_message'] = 'You have successfully
registered. Check your email address to activate your account.';
    }
    else
    {
        $data['auth_message'] = 'You have successfully
registered. '.anchor(site_url($this->dx_auth->login_uri), 'Login');
    }

    // Load registration success page
    $this->load->view($this->dx_auth-
>register_success_view, $data);
}
else
{
    // Load registration page
    $this->load->view('auth/register_recaptcha_form');
}
}
elseif ( ! $this->dx_auth->allow_registration)
{
    $data['auth_message'] = 'Registration has been disabled.';
    $this->load->view($this->dx_auth->register_disabled_view,
$data);
}
else
{
    $data['auth_message'] = 'You have to logout first, before
registering.';
    $this->load->view($this->dx_auth->logged_in_view, $data);
}
}

function activate()
{

```

```

// Get username and key
$username = $this->uri->segment(3);
$key = $this->uri->segment(4);

// Activate user
if ($this->dx_auth->activate($username, $key))
{
    $data['auth_message'] = 'Your account have been successfully
activated. '.anchor(site_url($this->dx_auth->login_uri), 'Login');
    $this->load->view($this->dx_auth->activate_success_view,
$data);
}
else
{
    $data['auth_message'] = 'The activation code you entered was
incorrect. Please check your email again.';
    $this->load->view($this->dx_auth->activate_failed_view, $data);
}
}

function forgot_password()
{
    $val = $this->form_validation;

    // Set form validation rules
    $val->set_rules('login', 'Username or Email address',
'trim|required|xss_clean');

    // Validate rules and call forgot password function
    if ($val->run() AND $this->dx_auth->forgot_password($val-
>set_value('login'))
    {
        $data['auth_message'] = 'An email has been sent to your email
with instructions with how to activate your new password.';
        $this->load->view($this->dx_auth-
>forgot_password_success_view, $data);
    }
    else
    {
        $this->load->view($this->dx_auth->forgot_password_view);
    }
}

function reset_password()
{
    // Get username and key
    $username = $this->uri->segment(3);
    $key = $this->uri->segment(4);

    // Reset password

```

```

        if ($this->dx_auth->reset_password($username, $key))
        {
            $data['auth_message'] = 'You have successfully reset you
password, '.anchor(site_url($this->dx_auth->login_uri), 'Login');
            $this->load->view($this->dx_auth-
>reset_password_success_view, $data);
        }
        else
        {
            $data['auth_message'] = 'Reset failed. Your username and key
are incorrect. Please check your email again and follow the instructions.';
            $this->load->view($this->dx_auth->reset_password_failed_view,
$data);
        }
    }

    function change_password()
    {
        // Check if user logged in or not
        if ($this->dx_auth->is_logged_in())
        {
            $val = $this->form_validation;

            // Set form validation
            $val->set_rules('old_password', 'Old Password',
'trim|required|xss_clean|min_length['.$this->min_password.']]max_length['.$this-
>max_password.']]');
            $val->set_rules('new_password', 'New Password',
'trim|required|xss_clean|min_length['.$this->min_password.']]max_length['.$this-
>max_password.']]matches[confirm_new_password]');
            $val->set_rules('confirm_new_password', 'Confirm new
Password', 'trim|required|xss_clean');

            // Validate rules and change password
            if ($val->run() AND $this->dx_auth->change_password($val-
>set_value('old_password'), $val->set_value('new_password')))
            {
                $data['auth_message'] = 'Your password has successfully
been changed.';
                $this->load->view($this->dx_auth-
>change_password_success_view, $data);
            }
            else
            {
                $this->load->view($this->dx_auth-
>change_password_view);
            }
        }
        else
        {

```

```

        // Redirect to login page
        $this->dx_auth->deny_access('login');
    }
}

function cancel_account()
{
    // Check if user logged in or not
    if ($this->dx_auth->is_logged_in())
    {
        $val = $this->form_validation;

        // Set form validation rules
        $val->set_rules('password', 'Password',
"trim|required|xss_clean");

        // Validate rules and change password
        if ($val->run() AND $this->dx_auth->cancel_account($val-
>set_value('password')))
        {
            // Redirect to homepage
            redirect("", 'location');
        }
        else
        {
            $this->load->view($this->dx_auth->cancel_account_view);
        }
    }
    else
    {
        // Redirect to login page
        $this->dx_auth->deny_access('login');
    }
}

// Example how to get permissions you set permission in
/backend/custom_permissions/
function custom_permissions()
{
    if ($this->dx_auth->is_logged_in())
    {
        echo 'My role: '.$this->dx_auth->get_role_name().'  
';
        echo 'My permission:   
';

        if ($this->dx_auth->get_permission_value('edit') != NULL AND
$this->dx_auth->get_permission_value('edit'))
        {
            echo 'Edit is allowed';
        }
        else
    }
}

```

```

        {
            echo 'Edit is not allowed';
        }

        echo '<br/>';

        if ($this->dx_auth->get_permission_value('delete') != NULL AND
$this->dx_auth->get_permission_value('delete'))
        {
            echo 'Delete is allowed';
        }
        else
        {
            echo 'Delete is not allowed';
        }
    }
}
?>

```

#### **system\application\controllers\backend.php**

```

<?php
class Backend extends Controller
{
    function Backend()
    {
        parent::Controller();

        $this->load->library('Table');
        $this->load->library('Pagination');
        $this->load->library('DX_Auth');

        $this->load->helper('form');
        $this->load->helper('url');

        // Protect entire controller so only admin,
        // and users that have granted role in permissions table can access it.
        $this->dx_auth->check_uri_permissions();
    }

    function index()
    {
        $this->users();
    }

    function users()
    {
        $this->load->model('dx_auth/users', 'users');

        // Search checkbox in post array
    }
}

```

```

foreach ($_POST as $key => $value)
{
    // If checkbox found
    if (substr($key, 0, 9) == 'checkbox_')
    {
        // If ban button pressed
        if (isset($_POST['ban']))
        {
            // Ban user based on checkbox value (id)
            $this->users->ban_user($value);
        }
        // If unban button pressed
        else if (isset($_POST['unban']))
        {
            // Unban user
            $this->users->unban_user($value);
        }
        else if (isset($_POST['reset_pass']))
        {
            // Set default message
            $data['reset_message'] = 'Reset password failed';

            // Get user and check if User ID exist
            if ($query = $this->users->get_user_by_id($value)
AND $query->num_rows() == 1)
            {
                // Get user record
                $user = $query->row();

                // Create new key, password and send email
                to user
                >username))
                if ($this->dx_auth->forgot_password($user-
                {
                    // Query once again, because the
                    database is updated after calling forgot_password.
                    $query = $this->users-
                    >get_user_by_id($value);

                    // Get user record
                    $user = $query->row();

                    // Reset the password
                    if ($this->dx_auth-
                    >reset_password($user->username, $user->newpass_key))
                    {
                        $data['reset_message'] =
                        'Reset password success';
                    }
                }
            }
        }
    }
}

```

```

    }
    }
    }
}

/* Showing page to user */

// Get offset and limit for page viewing
$offset = (int) $this->uri->segment(3);
// Number of record showing per page
$row_count = 10;

// Get all users
$data['users'] = $this->users->get_all($offset, $row_count)->result();

// Pagination config
$p_config['base_url'] = '/backend/users/';
$p_config['uri_segment'] = 3;
$p_config['num_links'] = 2;
$p_config['total_rows'] = $this->users->get_all()->num_rows();
$p_config['per_page'] = $row_count;

// Init pagination
$this->pagination->initialize($p_config);
// Create pagination links
$data['pagination'] = $this->pagination->create_links();

// Load view
$this->load->view('backend/users', $data);
}

function unactivated_users()
{
    $this->load->model('dx_auth/user_temp', 'user_temp');

    /* Database related */

    // If activate button pressed
    if ($this->input->post('activate'))
    {
        // Search checkbox in post array
        foreach ($_POST as $key => $value)
        {
            // If checkbox found
            if (substr($key, 0, 9) == 'checkbox_')
            {
                // Check if user exist, $value is username
                if ($query = $this->user_temp->get_login($value)
AND $query->num_rows() == 1)

```

```

        {
            // Activate user
            $this->dx_auth->activate($value, $query-
>row()->activation_key);
        }
    }
}

/* Showing page to user */

// Get offset and limit for page viewing
$offset = (int) $this->uri->segment(3);
// Number of record showing per page
$row_count = 10;

// Get all unactivated users
$data['users'] = $this->user_temp->get_all($offset, $row_count)-
>result();

// Pagination config
$p_config['base_url'] = '/backend/unactivated_users/';
$p_config['uri_segment'] = 3;
$p_config['num_links'] = 2;
$p_config['total_rows'] = $this->user_temp->get_all()->num_rows();
$p_config['per_page'] = $row_count;

// Init pagination
$this->pagination->initialize($p_config);
// Create pagination links
$data['pagination'] = $this->pagination->create_links();

// Load view
$this->load->view('backend/unactivated_users', $data);
}

function roles()
{
    $this->load->model('dx_auth/roles', 'roles');

    /* Database related */

    // If Add role button pressed
    if ($this->input->post('add'))
    {
        // Create role
        $this->roles->create_role($this->input->post('role_name'), $this-
>input->post('role_parent'));
    }
    else if ($this->input->post('delete'))

```

```

    {
        // Loop trough $_POST array and delete checked checkbox
        foreach ($_POST as $key => $value)
        {
            // If checkbox found
            if (substr($key, 0, 9) == 'checkbox_')
            {
                // Delete role
                $this->roles->delete_role($value);
            }
        }
    }

    /* Showing page to user */

    // Get all roles from database
    $data['roles'] = $this->roles->get_all()->result();

    // Load view
    $this->load->view('backend/roles', $data);
}

function uri_permissions()
{
    function trim_value(&$value)
    {
        $value = trim($value);
    }

    $this->load->model('dx_auth/roles', 'roles');
    $this->load->model('dx_auth/permissions', 'permissions');

    if ($this->input->post('save'))
    {
        // Convert back text area into array to be stored in permission
data    $allowed_uris = explode("\n", $this->input->post('allowed_uris'));

        // Remove white space if available
        array_walk($allowed_uris, 'trim_value');

        // Set URI permission data
        // IMPORTANT: uri permission data, is saved using 'uri' as key.
        // So this key name is preserved, if you want to use custom
permission use other key.
        $this->permissions->set_permission_value($this->input-
>post('role'), 'uri', $allowed_uris);
    }

    /* Showing page to user */

```

```

// Default role_id that will be showed
$role_id = $this->input->post('role') ? $this->input->post('role') : 1;

// Get all role from database
$data['roles'] = $this->roles->get_all()->result();
// Get allowed uri permissions
$data['allowed_uris'] = $this->permissions-
>get_permission_value($role_id, 'uri');

// Load view
$this->load->view('backend/uri_permissions', $data);
}

function custom_permissions()
{
// Load models
$this->load->model('dx_auth/roles', 'roles');
$this->load->model('dx_auth/permissions', 'permissions');

/* Get post input and apply it to database */

// If button save pressed
if ($this->input->post('save'))
{
// Note: Since in this case we want to insert two key with each
value at once,
// it's not advisable using set_permission_value() function

// If you calling that function twice that means, you will query
database 4 times,
// because set_permission_value() will access table 2 times,
// one for get previous permission and the other one is to save it.

// For this case (or you need to insert few key with each value at
once)
// Use the example below

// Get role_id permission data first.
// So the previously set permission array key won't be overwritten
with new array with key $key only,
// when calling set_permission_data later.
$permission_data = $this->permissions-
>get_permission_data($this->input->post('role'));

// Set value in permission data array
$permission_data['edit'] = $this->input->post('edit');
$permission_data['delete'] = $this->input->post('delete');

// Set permission data for role_id

```

```

        $this->permissions->set_permission_data($this->input-
>post('role'), $permission_data);
    }

    /* Showing page to user */

    // Default role_id that will be showed
    $role_id = $this->input->post('role') ? $this->input->post('role') : 1;

    // Get all role from database
    $data['roles'] = $this->roles->get_all()->result();
    // Get edit and delete permissions
    $data['edit'] = $this->permissions->get_permission_value($role_id,
'edit');
    $data['delete'] = $this->permissions->get_permission_value($role_id,
'delete');

    // Load view
    $this->load->view('backend/custom_permissions', $data);
}
}
?>

```

### **\\system\application\controllers\fornecedor.php**

```

<?php
class Fornecedor extends Controller {

    function Fornecedor() {
        parent::Controller();
        $this->form_validation->set_error_delimiters('<div class="error">', '</div>');
        $this->load->model('fornecedor_model');
    }

    function index() {
        $this->listar();
    }

    function listar() {
        // verifica se o usuário está autenticado
        if ($this->dx_auth->is_logged_in()) {
            // campos que vêm do formulário
            $cd_fornecedor = $this->input->post('cd_fornecedor');
            $nm_fornecedor = $this->input->post('nm_fornecedor');
            $fl_situacao = $this->input->post('fl_situacao');
            if (empty($fl_situacao)) {
                $radioSituacaoAtivo = false;
                $radioSituacaoInativo = false;
                $radioSituacaoIndiferente = true;
            } else {
                $radioSituacaoAtivo = ($fl_situacao==1);
            }
        }
    }
}

```

```

        $radioSituacaoInativo = ($fl_situacao==2);
        $radioSituacaoIndiferente = ($fl_situacao==0);
    }

    $listaFornecedor = array();

    // busca os fornecedores a partir dos campos informados no
    formulário de pesquisa
    $fornecedores = $this->fornecedor_model-
>buscaFornecedorPeloFiltro($cd_fornecedor, $nm_fornecedor, $fl_situacao);

    $fornecedorLista = new stdClass();

    // percorre os fornecedores e trata as informações específicas de
    cada fornecedor
    foreach ($fornecedores as $fornecedor) {
        // código do fornecedor
        $fornecedorLista->codigo = $fornecedor->cd_fornecedor;

        // descrição do fornecedor
        $fornecedorLista->nome = $fornecedor->nm_fornecedor;

        // ordem do fornecedor
        $fornecedorLista->ordem = $fornecedor->nr_ordem;

        // fornecedor referência
        $fornecedorLista->referencia = (empty($fornecedor-
>fl_referencia) || $fornecedor->fl_referencia == 2)?"Não":"Sim";

        // situação do fornecedor
        if ($fornecedor->fl_situacao == 1) { // ativo
            $fornecedorLista->situacao = 'cd_fornecedor.\').load(\'/index.php/ajax/inativarFornecedor/'. $fornecedor-
>cd_fornecedor.\')" style="cursor: pointer;">';
        } else { // inativo
            $fornecedorLista->situacao = 'cd_fornecedor.\').load(\'/index.php/ajax/ativarFornecedor/'. $fornecedor-
>cd_fornecedor.\')" style="cursor: pointer;">';
        }

        // colocando o fornecedor na lista que vai para a view
        $listaFornecedor[] = $fornecedorLista;

        // limpando objeto fornecedor
        $fornecedorLista = null;
    }

```

```

// informa para a view o array ($data) com os dados que ela
poderá utilizar
    $data['radioSituacaoAtivo'] = $radioSituacaoAtivo;
    $data['radioSituacaoInativo'] = $radioSituacaoInativo;
    $data['radioSituacaoIndiferente'] = $radioSituacaoIndiferente;
    $data['listaFornecedor'] = $listaFornecedor;

    $this->load->view("fornecedor_lista", $data);
} else {
    // redireciona para a página inicial quando não está logado
    redirect("", 'location');
}
}

function visualizar() {
    // verifica se o usuário está autenticado
    if ($this->dx_auth->is_logged_in()) {
        $cd_fornecedor = $this->uri->segment(3);
        if (!empty($cd_fornecedor)) {
            // busca os dados do fornecedor informado
            $fornecedorResult = $this->fornecedor_model-
>buscaFornecedorPeloCodigo($cd_fornecedor);

            if (count($fornecedorResult) > 0) {
                $fornecedor = $fornecedorResult[0];

                $fornecedorConsulta = new stdClass();
                $fornecedorConsulta->codigo = $fornecedor->cd_fornecedor;
                $fornecedorConsulta->nome = $fornecedor->nm_fornecedor;
                $fornecedorConsulta->webservice = $fornecedor-
>de_url_ws;
                $fornecedorConsulta->ordem = $fornecedor-
>nr_ordem;
                $fornecedorConsulta->referencia = ($fornecedor-
>fl_referencia==1)?'Sim':'Não';
                $fornecedorConsulta->situacao = ($fornecedor-
>fl_situacao==1)?'Ativo':'Inativo';

                $data['fornecedor'] = $fornecedorConsulta;

                $this->load->view("fornecedor_visualizar", $data);
            } else {
                // código do fornecedor informado não está cadastrado no banco de
dados
            }
        } else {
            // não foi informado o código do fornecedor a ser visualizado
        }
    } else {

```

```

// redireciona para a página inicial quando não está logado
redirect("", 'location');
}
}

function cadastrar() {
    // verifica se o usuário está autenticado
    if ($this->dx_auth->is_logged_in()) {
        $data = $this->_carregaDadosFormCadastrar();
        $this->load->view("fornecedor_cadastro", $data);
    } else {
        // redireciona para a página inicial quando não está logado
        redirect("", 'location');
    }
}

function gravar() {
    // verifica se o usuário está autenticado
    if ($this->dx_auth->is_logged_in()) {

        // validação dos dados do formulário
        $this->form_validation->set_rules('nm_fornecedor', 'Nome',
'required');
        $this->form_validation->set_rules('de_url_ws', 'Web service',
'required');
        $this->form_validation->set_rules('nr_ordem', 'Ordem',
'required');

        if ($this->form_validation->run()) {
            // preparando os dados do formulário para gravar no
banco de dados
            $fornecedor['cd_fornecedor'] = "";
            $fornecedor['nm_fornecedor'] = $this->input-
>post('nm_fornecedor');
            $fornecedor['de_url_ws'] = $this->input->post('de_url_ws');
            $fornecedor['nr_ordem'] = $this->input->post('nr_ordem');
            $fornecedor['fl_referencia'] = $this->input-
>post('fl_referencia');
            $fornecedor['fl_situacao'] = $this->input-
>post('fl_situacao');

            // grava as informações no banco de dados
            $fornecedor['cd_fornecedor'] = $this->fornecedor_model-
>gravarFornecedor($fornecedor);

            // redireciona para a tela de sucesso
            $data['url'] = site_url('fornecedor/visualizar/' .
$fornecedor['cd_fornecedor']);
            $this->load->view("sipec_pagina_sucesso", $data);
        } else {

```

```

// se deu algum problema na validação, carrega novamente a
tela de cadastro
        $data = $this->_carregaDadosFormCadastrar();
    $this->load->view("fornecedor_cadastro", $data);
    }
    } else {
        // redireciona para a página inicial quando não está logado
        redirect("", 'location');
    }
}

function editar() {
    // verifica se o usuário está autenticado
    if ($this->dx_auth->is_logged_in()) {
        $cd_fornecedor = $this->uri->segment(3);
        if (!empty($cd_fornecedor)) {
            $data = $this->_carregaDadosFormEditar($cd_fornecedor, false);
            $this->load->view("fornecedor_edicao", $data);
        } else {
            // código do fornecedor a ser editado não foi informado
        }
    } else {
        // redireciona para a página inicial quando não está logado
        redirect("", 'location');
    }
}

function atualizar() {
    // verifica se o usuário está autenticado
    if ($this->dx_auth->is_logged_in()) {

        // preparando os dados do formulário para gravar no banco de
dados
        $cd_fornecedor = $this->input->post('cd_fornecedor');
        if (!empty($cd_fornecedor)) {

            // validação dos dados do formulário
            $this->form_validation->set_rules('nm_fornecedor',
'Nome', 'required');
            $this->form_validation->set_rules('de_url_ws', 'Web
service', 'required');
            $this->form_validation->set_rules('nr_ordem', 'Ordem',
'required');

            // preparando os dados do formulário para gravar no banco de dados
            $fornecedor['nm_fornecedor'] = $this->input->post('nm_fornecedor');
            $fornecedor['de_url_ws'] = $this->input->post('de_url_ws');
            $fornecedor['nr_ordem'] = $this->input->post('nr_ordem');

```

```

        $fornecedor['fl_referencia'] = $this->input->post('fl_referencia');
        $fornecedor['fl_situacao'] = $this->input-
>post('fl_situacao');

        if ($this->form_validation->run()) {
            // faz um update nos dados do fornecedor
            $this->fornecedor_model-
>atualizarFornecedor($cd_fornecedor, $fornecedor);

            // redireciona para a visualização do fornecedor
            $data['url'] = site_url('fornecedor/visualizar/' .
$cd_fornecedor);

            $this->load->view("sipec_pagina_sucesso", $data);
        } else {
            //carrega a view de edição com os dados para serem
populados
            $data = $this-
>_carregaDadosFormEditar($cd_fornecedor, true);
            $this->load->view("fornecedor_edicao", $data);
        }
        } else {
            // código do fornecedor a ser atualizado não foi informado
        }
    } else {
        // redireciona para a página inicial quando não está logado
        redirect("", 'location');
    }
}

function excluir() {
    // verifica se o usuário está autenticado
    if ($this->dx_auth->is_logged_in()) {

        // validação dos dados do formulário
        $this->form_validation->set_rules('cd_fornecedor_exclusao', 'Fornecedor',
'callback_fornecedorNaoUtilizado');

        if ($this->form_validation->run()) {
            $this->fornecedor_model->excluirFornecedor($this->input-
>post('cd_fornecedor_exclusao'));

            // redireciona para a lista de fornecedores
            $data['url'] = site_url('fornecedor');
            $this->load->view("sipec_pagina_sucesso", $data);
        } else {
            $radioSituacaoAtivo = false;
            $radioSituacaoInativo = false;
            $radioSituacaoIndiferente = true;

            $listaFornecedor = array();

```

```

// busca os fornecedores a partir dos campos informados
no formulário de pesquisa
    $fornecedores = $this->fornecedor_model-
>buscaFornecedorPeloFiltro(null, null, null);

    $fornecedorLista = new stdClass();

// percorre os fornecedores e trata as informações
específicas de cada fornecedor
    foreach ($fornecedores as $fornecedor) {
        // código do fornecedor
        $fornecedorLista->codigo = $fornecedor-
>cd_fornecedor;

        // descrição do fornecedor
        $fornecedorLista->nome = $fornecedor-
>nm_fornecedor;

        // ordem do fornecedor
        $fornecedorLista->ordem = $fornecedor-
>nr_ordem;

        // fornecedor referência
        $fornecedorLista->referencia = (empty($fornecedor-
>fl_referencia) || $fornecedor->fl_referencia == 2)?"Não":"Sim";

        // situação do fornecedor
        if ($fornecedor->fl_situacao == 1) { // ativo
            $fornecedorLista->situacao = 'cd_fornecedor.'\').load(\'/index.php/ajax/inativarFornecedor/'. $fornecedor-
>cd_fornecedor.'\')" style="cursor: pointer;">';
        } else { // inativo
            $fornecedorLista->situacao = 'cd_fornecedor.'\').load(\'/index.php/ajax/ativarFornecedor/'. $fornecedor-
>cd_fornecedor.'\')" style="cursor: pointer;">';
        }

        // colocando o fornecedor na lista que vai para a
view
        $listaFornecedor[] = $fornecedorLista;

        // limpando objeto fornecedor
        $fornecedorLista = null;
    }
}

```

```

// informa para a view o array ($data) com os dados que
ela poderá utilizar
    $data['radioSituacaoAtivo'] = $radioSituacaoAtivo;
    $data['radioSituacaoInativo'] = $radioSituacaoInativo;
    $data['radioSituacaoIndiferente'] =
$radioSituacaoIndiferente;
    $data['listaFornecedor'] = $listaFornecedor;

    $this->load->view("fornecedor_lista", $data);
}
} else {
    // redireciona para a página inicial quando não está logado
    redirect("", 'location');
}
}

function _carregaDadosFormCadastrar() {
    $fornecedorConsulta = new stdClass();

    $fl_referencia = $this->input->post('fl_referencia');
    if (empty($fl_referencia)) {
        $fornecedorConsulta->radioReferenciaSim = false;
        $fornecedorConsulta->radioReferenciaNao = true;
    } else {
        $fornecedorConsulta->radioReferenciaSim = ($fl_referencia==1);
        $fornecedorConsulta->radioReferenciaNao = ($fl_referencia==2);
    }

    $fl_situacao = $this->input->post('fl_situacao');
    if (empty($fl_situacao)) {
        $fornecedorConsulta->radioSituacaoAtivo = true;
        $fornecedorConsulta->radioSituacaoInativo = false;
    } else {
        $fornecedorConsulta->radioSituacaoAtivo = ($fl_situacao==1);
        $fornecedorConsulta->radioSituacaoInativo = ($fl_situacao==2);
    }
}

$data['fornecedor'] = $fornecedorConsulta;

return $data;
}

/**
 * Prepara os dados que serão carregados na view de edição do fornecedor
 * @param $cd_fornecedor
 * @param $temPost
 * @return Retorna um array $data com as informações que serão carregadas
na view
 */
function _carregaDadosFormEditar($cd_fornecedor, $temPost) {

```

```

        // busca os dados do fornecedor informado
        $fornecedorResult = $this->fornecedor_model-
>buscaFornecedorPeloCodigo($cd_fornecedor);

        if (count($fornecedorResult) > 0) {
            $fornecedor = $fornecedorResult[0];

            // dados do formulário
            if ($temPost) {
                $fornecedor->nm_fornecedor = $this->input->post('nm_fornecedor');
                $fornecedor->de_url_ws = $this->input->post('de_url_ws');
                $fornecedor->nr_ordem = $this->input->post('nr_ordem');
                $fornecedor->fl_referencia = $this->input->post('fl_referencia');
                $fornecedor->fl_situacao = $this->input-
>post('fl_situacao');
            }

            $fornecedorConsulta = new stdClass();
            $fornecedorConsulta->cd_fornecedor = $fornecedor->cd_fornecedor;
            $fornecedorConsulta->nm_fornecedor = $fornecedor->nm_fornecedor;
            $fornecedorConsulta->de_url_ws = $fornecedor->de_url_ws;
            $fornecedorConsulta->nr_ordem = $fornecedor->nr_ordem;
            $fornecedorConsulta->fl_referencia = $fornecedor->fl_referencia;
            $fornecedorConsulta->fl_situacao = $fornecedor->fl_situacao;

            $fornecedorConsulta->radioReferenciaSim = ($fornecedor-
>fl_referencia==1);
            $fornecedorConsulta->radioReferenciaNao = (empty($fornecedor-
>fl_referencia) || $fornecedor->fl_referencia==2);

            $fornecedorConsulta->radioSituacaoAtivo = (empty($fornecedor->fl_situacao)
|| $fornecedor->fl_situacao==1);
            $fornecedorConsulta->radioSituacaoInativo = ($fornecedor->fl_situacao==2);

            $data['fornecedor'] = $fornecedorConsulta;
        } else {
            // código do fornecedor informado não foi encontrado no banco de dados
        }
        return $data;
    }

    function fornecedorNaoUtilizado($cd_fornecedor) {
        // verifica se o fornecedor informado está vinculado a um produto
        $possui = $this->fornecedor_model->fornecedorPossuiVinculo($cd_fornecedor);

        if (!$possui) {
            return true;
        } else {
            $this->form_validation->set_message('fornecedorNaoUtilizado', 'Não foi
possível excluir o registro porque ele está sendo utilizado');
        }
    }

```

```

        return false;
    }
}

/*
 *
 * TESTES UNITÁRIOS
 *
 */

function relatorioTestes() {
    /*$this->unit->run($this->cnpjNaoExiste("57.373.277/0001-87"), true, "O
cnpj 57.373.277/0001-87 não existe no banco de dados");
    $this->unit->run($this->cnpjNaoExiste("83.472.803/0001-76"), false, "O
cnpj 83.472.803/0001-76 existe no banco de dados");
    $this->unit->run($this->cpfNaoExiste("834.166.468-27"), true, "O cpf
834.166.468-27 não existe no banco de dados");
    $this->unit->run($this->cpfNaoExiste("006.268.039-00"), false, "O cpf
006.268.039-00 existe no banco de dados");

    echo $this->unit->report();*/
}
}
?>

```

### **\\system\application\controllers\importacao.php**

```
<?php
```

```
class Importacao extends Controller {
```

```
    function Importacao() {
```

```
        parent::Controller();
```

```
        $this->load->model('categoria_model');
```

```
        $this->load->model('produto_model');
```

```
        $this->load->model('fornecedor_model');
```

```
        $this->load->model('importacao_model');
```

```
    }
```

```
    function index() {
```

```
        $this->executar();
```

```
    }
```

```
    function executar() {
```

```
        // verifica se o usuário está autenticado
```

```
        if ($this->dx_auth->is_logged_in()) {
```

```
            $this->load->view("importacao_executar");
```

```
        } else {
```

```
            // redireciona para a página inicial quando não está logado
```

```
            redirect("", 'location');
```

```
        }
```

```

}

function historico() {
    // verifica se o usuário está autenticado
    if ($this->dx_auth->is_logged_in()) {
        // campos que vêm do formulário
        $cd_importacao = $this->input->post('cd_importacao');
        $cd_fornecedor = $this->input->post('cd_fornecedor');
        $cd_produto = $this->input->post('cd_produto');
        $fl_mais_barato = $this->input->post('fl_mais_barato');
        if (empty($fl_mais_barato)) {
            $radioMaisBaratoSim = false;
            $radioMaisBaratoNao = false;
            $radioMaisBaratoIndiferente = true;
        } else {
            $radioMaisBaratoSim = ($fl_mais_barato==1);
            $radioMaisBaratoNao = ($fl_mais_barato==2);
            $radioMaisBaratoIndiferente = ($fl_mais_barato==0);
        }

        // buscando a lista de categorias
        $data['listaHistoricoImportacoes'] = $this->importacao_model-
>listaHistoricoImportacoes();
        $data['listaFornecedores'] = $this->fornecedor_model-
>listaFornecedores();
        $data['listaProdutos'] = $this->produto_model->listaProdutos();

        $listaImportacao = array();

        // busca as importações a partir dos campos informados no
        formulário de pesquisa
        $importacoes = $this->importacao_model-
>buscaHistoricoImportacaoPeloFiltro($cd_importacao, $cd_fornecedor, $cd_produto,
$fl_mais_barato);
        //echo "<pre>";print_r($importacoes);exit;

        $importacaoLista = new stdClass();

        // percorre as importações
        foreach ($importacoes as $importacao) {
            $importacaoLista->data = $this->funcoes_genericas-
>mysql_date_para_humano($importacao->dt_execucao);
            $importacaoLista->produto = $importacao->de_titulo;
            $importacaoLista->preco = $this->funcoes_genericas-
>mysql_decimal_para_moeda($importacao->vl_preco);
            $importacaoLista->estoque = $importacao->qt_estoque;
            $importacaoLista->fornecedor = $importacao-
>nm_fornecedor;
            $importacaoLista->mais_barato = (empty($importacao-
>fl_mais_barato) || $importacao->fl_mais_barato == 2)?"Não":"Sim";

```

```

        $listaImportacao[] = $importacaoLista;

        // limpando objeto fornecedor
        $importacaoLista = null;
    }

    // informa para a view o array ($data) com os dados que ela
    poderá utilizar
    $data['radioMaisBaratoSim'] = $radioMaisBaratoSim;
    $data['radioMaisBaratoNao'] = $radioMaisBaratoNao;
    $data['radioMaisBaratoIndiferente'] =
    $radioMaisBaratoIndiferente;
    $data['listaImportacao'] = $listaImportacao;

    $this->load->view("importacao_historico", $data);
} else {
    // redireciona para a página inicial quando não está logado
    redirect("", 'location');
}
}

function cargaInicial() {
    // verifica se o usuário está autenticado
    if ($this->dx_auth->is_logged_in()) {
        $this->categoria_model->cargaInicial();
    } else {
        // redireciona para a página inicial quando não está logado
        redirect("", 'location');
    }
}
}
?>

```

### **\\system\application\controllers\inicio.php**

```
<?php
```

```

class Inicio extends Controller {

    function Inicio() {
        parent::Controller();
    }

    function index() {
        // verifica se o usuário está autenticado
        if ($this->dx_auth->is_logged_in()) {
            $this->load->view('tela_inicial');
        } else {
            // redireciona para a página inicial quando não está logado
            redirect("", 'location');
        }
    }
}

```

```

    }
}
?>

```

**\system\application\controllers\login.php**  
<?php

```

class Login extends Controller {

    function Login() {
        parent::Controller();
        $this->load->model('login_model');
    }

    function index() {
        $this->load->view('tela_login');
    }

    function autenticacao() {
        $usuario = $_POST[usuario];
        $senha = $_POST[senha];

        if (!empty($usuario)) {
            $senhaBD = $this->login_model->getSenha($usuario);
        }

    }

    function logout() {

    }
}
?>

```

**\system\application\controllers\usuario.php**  
<?php

```

class Usuario extends Controller {

    function Usuario() {
        parent::Controller();
        $this->form_validation->set_error_delimiters('<div class="error">', '</div>');
        $this->load->model('usuario_model');
    }

    function index() {

```

```

    $this->listar();
}

function listar() {
    // verifica se o usuário está autenticado
    if ($this->dx_auth->is_logged_in()) {
        // campos que vêm do formulário
        $id = $this->input->post('id');
        $username = $this->input->post('username');
        $email = $this->input->post('email');

        $listaUsuario = array();

        //if ($this->input->post('processou')) {
            // busca os usuários a partir dos campos informados no formulário de
pesquisa
            $usuarios = $this->usuario_model->buscaUsuarioPeloFiltro($id,
$username, $email);

            $usuarioLista = new stdClass();

            // percorre os usuários e trata as informações específicas de cada usuário
            foreach ($usuarios as $usuario) {
                $usuarioLista->codigo = $usuario->id;
                $usuarioLista->nome = $usuario->username;
                $usuarioLista->email = $usuario->email;
                $usuarioLista->data_criacao = $this->funcoes_genericas-
>mysql_datetime_para_humano($usuario->created);
                $usuarioLista->data_atualizacao = $this->funcoes_genericas-
>mysql_datetime_para_humano($usuario->modified);

                // colocando o usuário na lista que vai para a view
                $listaUsuario[] = $usuarioLista;

                // limpando objeto usuário
                $usuarioLista = null;
            }
        //}

        // informa para a view o array ($data) com os dados que ela poderá utilizar
        $data['listaUsuario'] = $listaUsuario;

        $this->load->view("usuario_lista", $data);
    } else {
        // redireciona para a página inicial quando não está logado
        redirect("", 'location');
    }
}

function visualizar() {

```

```

// verifica se o usuário está autenticado
if ($this->dx_auth->is_logged_in()) {
    $id = $this->uri->segment(3);
    if (!empty($id)) {
        // busca os dados do usuário informado
        $usuarioResult = $this->usuario_model->buscaUsuarioPeloCodigo($id);

        if (count($usuarioResult) > 0) {
            $usuario = $usuarioResult[0];

            $usuarioConsulta = new stdClass();
            $usuarioConsulta->codigo = $usuario->id;
            $usuarioConsulta->nome = $usuario->username;
            $usuarioConsulta->email = $usuario->email;
            $usuarioConsulta->data_cadastro = $this->funcoes_genericas-
>mysql_datetime_para_humano($usuario->created);
            $usuarioConsulta->data_alteracao = $this->funcoes_genericas-
>mysql_datetime_para_humano($usuario->modified);

            $usuarioConsulta->ultimo_acesso = $this->funcoes_genericas-
>mysql_datetime_para_humano($usuario->last_login);
            $usuarioConsulta->ultimo_ip = $usuario->last_ip;

            $data['usuario'] = $usuarioConsulta;

            $this->load->view("usuario_visualizar", $data);
        } else {
            // código do usuário informado não está cadastrado no banco de dados
        }
        } else {
            // não foi informado o código do usuário a ser visualizado
        }
    } else {
        // redireciona para a página inicial quando não está logado
        redirect("", 'location');
    }
}

function cadastrar() {
    // verifica se o usuário está autenticado
    if ($this->dx_auth->is_logged_in()) {
        $data = $this->_carregaDadosFormCadastrar();
        $this->load->view("usuario_cadastro", $data);
    } else {
        // redireciona para a página inicial quando não está logado
        redirect("", 'location');
    }
}

function username_check($username) {

```

```

$result = $this->dx_auth->is_username_available($username);
if (!$result) {
    $this->form_validation->set_message('username_check', 'O nome de usuário
informado já está sendo utilizado. Por favor escolha outro.');
```

```

    }

    return $result;
}

function email_check($email) {
    $result = $this->dx_auth->is_email_available($email);
    if (!$result) {
        $this->form_validation->set_message('email_check', 'O e-mail informado já
está sendo utilizado. Por favor informe outro.');
```

```

    }

    return $result;
}

function gravar() {
    // verifica se o usuário está autenticado
    if ($this->dx_auth->is_logged_in()) {

        // validação dos dados do formulário
        $this->form_validation->set_rules('username', 'Nome do usuário',
'trim|required|xss_clean|min_length[4]|max_length[20]|callback_username_check|alp
ha_dash');
        $this->form_validation->set_rules('email', 'E-mail',
'trim|required|xss_clean|valid_email|callback_email_check');
        $this->form_validation->set_rules('password', 'Senha',
'trim|required|xss_clean|min_length[4]|max_length[20]|matches[password2]');
        $this->form_validation->set_rules('password2', 'Confirmar senha',
'trim|required|xss_clean');
```

```

        if ($this->form_validation->run()) {
            // grava as informações no banco de dados
            $this->dx_auth->register($this->input->post('username'), $this->input-
>post('password'), $this->input->post('email'));

            // redireciona para a tela de sucesso
            $data['url'] = site_url('usuario');
            $this->load->view("sipec_pagina_sucesso", $data);
        } else {
            // se deu algum problema na validação, carrega novamente a tela de
cadastro
            $data = $this->_carregaDadosFormCadastrar();
            $this->load->view("usuario_cadastro", $data);
        }
    } else {
        // redireciona para a página inicial quando não está logado

```

```

        redirect("", 'location');
    }
}

function editar() {
    // verifica se o usuário está autenticado
    if ($this->dx_auth->is_logged_in()) {
        $id = $this->uri->segment(3);
        if (!empty($id)) {
            $data = $this->_carregaDadosFormEditar($id, false);
            $this->load->view("usuario_edicao", $data);
        } else {
            // código do usuário a ser editado não foi informado
        }
    } else {
        // redireciona para a página inicial quando não está logado
        redirect("", 'location');
    }
}

function atualizar() {
    // verifica se o usuário está autenticado
    if ($this->dx_auth->is_logged_in()) {

        // preparando os dados do formulário para gravar no banco de dados
        $id = $this->input->post('id');
        if (!empty($id)) {

            // validação dos dados do formulário
            $this->form_validation->set_rules('id', 'Código', 'required');
            if (trim($this->input->post('email_bd')) != trim($this->input->post('email'))) {
                $this->form_validation->set_rules('email', 'E-mail',
'trim|required|xss_clean|valid_email|callback_email_check');
            }
            if ($this->input->post('trocar_senha') == 'sim') {
                $this->form_validation->set_rules('password', 'Nova senha',
'trim|required|xss_clean|min_length[4]|max_length[20]|matches[password2]');
                $this->form_validation->set_rules('password2', 'Confirmar senha',
'trim|required|xss_clean');
            }

            // preparando os dados do formulário para gravar no banco de dados
            $usuario['email'] = trim($this->input->post('email'));

            if ($this->form_validation->run()) {
                // faz um update nos dados de usuario
                $this->usuario_model->atualizarUsuario($id, $usuario);

                if ($this->input->post('trocar_senha') == 'sim') {

```

```

        $this->usuario_model->trocarSenha($id, $this->input-
>post('password'));
    }

    // redireciona para a visualização de usuário
    $data['url'] = site_url('usuario/visualizar/' . $id);
    $this->load->view("sipec_pagina_sucesso", $data);
} else {
    //carrega a view de edição com os dados para serem populados
    $data = $this->_carregaDadosFormEditar($id, true);
    $this->load->view("usuario_edicao", $data);
}
} else {
    // código da unidade a ser atualizado não foi informado
}
} else {
    // redireciona para a página inicial quando não está logado
    redirect("", 'location');
}
}

function excluir() {
    // verifica se o usuário está autenticado
    if ($this->dx_auth->is_logged_in()) {

        // validação dos dados do formulário
        $this->form_validation->set_rules('id_exclusao', 'Código do usuário',
'required|callback_verificaProprioUsuario');

        if ($this->form_validation->run()) {
            $this->usuario_model->excluirUsuario($this->input->post('id_exclusao'));

            // redireciona para a lista de usuários
            $data['url'] = site_url('usuario');
            $this->load->view("sipec_pagina_sucesso", $data);
        } else {
            //carrega a view da lista
            $this->load->view("usuario_lista");
        }
    } else {
        // redireciona para a página inicial quando não está logado
        redirect("", 'location');
    }
}

function _carregaDadosFormCadastrar() {
    $usuarioConsulta = new stdClass();

    $usuarioConsulta->username = $this->input->post('username');
    $usuarioConsulta->email = $this->input->post('email');
}

```

```

    $data['usuario'] = $usuarioConsulta;

    return $data;
}

/**
 * Prepara os dados que serão carregados na view de edição da unidade
 * @param $cd_unidade
 * @param $temPost
 * @return Retorna um array $data com as informações que serão carregadas na
view
 */
function _carregaDadosFormEditar($id, $temPost) {
    // busca os dados do usuario informado
    $usuarioResult = $this->usuario_model->buscaUsuarioPeloCodigo($id);

    if (count($usuarioResult) > 0) {
        $usuario = $usuarioResult[0];
        $usuario->email_bd = $usuario->email;

        // dados do formulário
        if ($temPost) {
            $usuario->email = $this->input->post('email');
            $usuario->trocar_senha = $this->input->post('trocar_senha');
        } else {
            $usuario->trocar_senha = "";
        }

        $usuarioConsulta = new stdClass();
        $usuarioConsulta->codigo = $usuario->id;
        $usuarioConsulta->nome = $usuario->username;
        $usuarioConsulta->email = $usuario->email;
        $usuarioConsulta->email_bd = $usuario->email_bd;
        $usuarioConsulta->data_cadastro = $this->funcoes_genericas-
>mysql_datetime_para_humano($usuario->created);
        $usuarioConsulta->data_alteracao = $this->funcoes_genericas-
>mysql_datetime_para_humano($usuario->modified);
        $usuarioConsulta->checkTrocarSenha = ($usuario->trocar_senha == 'sim');
        $usuarioConsulta->cssSenha = ($usuario->trocar_senha ==
'sim')?"":"campoDesabilitado";
        $usuarioConsulta->desabilitaSenha = ($usuario->trocar_senha ==
'sim')?"":"disabled"=>'disabled';

        $data['usuario'] = $usuarioConsulta;
    } else {
        // código do usuário informado não foi encontrado no banco de dados
    }
    return $data;
}

```

```

function verificaProprioUsuario($id) {
    if ($this->dx_auth->get_user_id() != $id) {
        return true;
    } else {
        $this->form_validation->set_message('verificaProprioUsuario',
'Você não pode excluir o próprio usuário');
        return false;
    }
}
}
}
?>

```

### **\\system\application\libraries\DX\_Auth.php**

```
<?php if (!defined('BASEPATH')) exit('No direct script access allowed');
```

```

/**
 * DX Auth Class
 *
 * Authentication library for Code Igniter.
 *
 * @author      Dexcel
 * @version     1.0.6
 * @based on    CL Auth by Jason Ashdown
                (http://http://www.jasonashdown.co.uk/)
 * @link        http://dexcel.shinsengumiteam.com/dx\_auth
 * @license     MIT License Copyright (c) 2008 Erick Hartanto
 * @credits     http://dexcel.shinsengumiteam.com/dx\_auth/general/credits.html
 */

```

```

class DX_Auth
{
    // Private
    var $_banned;
    var $_ban_reason;
    var $_auth_error; // Contain user error when login
    var $_captcha_image;

    function DX_Auth()
    {
        $this->ci =& get_instance();

        log_message('debug', 'DX Auth Initialized');

        // Load required library
        $this->ci->load->library('Session');
        $this->ci->load->database();
    }
}

```

```

// Load DX Auth config
$this->ci->load->config('dx_auth');

// Load DX Auth language
$this->ci->lang->load('dx_auth');

// Load DX Auth event
$this->ci->load->library('DX_Auth_Event');

// Initialize
$this->_init();
}

/* Private function */

function _init()
{
    // When we load this library, auto Login any returning users
    $this->autologin();

    // Init helper config variable
    $this->email_activation = $this->ci->config-
>item('DX_email_activation');

    $this->allow_registration = $this->ci->config-
>item('DX_allow_registration');
    $this->captcha_registration = $this->ci->config-
>item('DX_captcha_registration');

    $this->captcha_login = $this->ci->config->item('DX_captcha_login');

    // URIs
    $this->banned_uri = $this->ci->config->item('DX_banned_uri');
    $this->deny_uri = $this->ci->config->item('DX_deny_uri');
    $this->login_uri = $this->ci->config->item('DX_login_uri');
    $this->logout_uri = $this->ci->config->item('DX_logout_uri');
    $this->register_uri = $this->ci->config->item('DX_register_uri');
    $this->activate_uri = $this->ci->config->item('DX_activate_uri');
    $this->forgot_password_uri = $this->ci->config-
>item('DX_forgot_password_uri');
    $this->reset_password_uri = $this->ci->config-
>item('DX_reset_password_uri');
    $this->change_password_uri = $this->ci->config-
>item('DX_change_password_uri');
    $this->cancel_account_uri = $this->ci->config-
>item('DX_cancel_account_uri');

    // Forms view
    $this->login_view = $this->ci->config->item('DX_login_view');
    $this->register_view = $this->ci->config->item('DX_register_view');

```

```

        $this->forgot_password_view = $this->ci->config-
>item('DX_forgot_password_view');
        $this->change_password_view = $this->ci->config-
>item('DX_change_password_view');
        $this->cancel_account_view = $this->ci->config-
>item('DX_cancel_account_view');

        // Pages view
        $this->deny_view = $this->ci->config->item('DX_deny_view');
        $this->banned_view = $this->ci->config->item('DX_banned_view');
        $this->logged_in_view = $this->ci->config->item('DX_logged_in_view');
        $this->logout_view = $this->ci->config->item('DX_logout_view');

        $this->register_success_view = $this->ci->config-
>item('DX_register_success_view');
        $this->activate_success_view = $this->ci->config-
>item('DX_activate_success_view');
        $this->forgot_password_success_view = $this->ci->config-
>item('DX_forgot_password_success_view');
        $this->reset_password_success_view = $this->ci->config-
>item('DX_reset_password_success_view');
        $this->change_password_success_view = $this->ci->config-
>item('DX_change_password_success_view');

        $this->register_disabled_view = $this->ci->config-
>item('DX_register_disabled_view');
        $this->activate_failed_view = $this->ci->config-
>item('DX_activate_failed_view');
        $this->reset_password_failed_view = $this->ci->config-
>item('DX_reset_password_failed_view');
    }

    function _gen_pass($len = 8)
    {
        // No Zero (for user clarity);
        $pool =
'123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';

        $str = "";
        for ($i = 0; $i < $len; $i++)
        {
            $str .= substr($pool, mt_rand(0, strlen($pool) -1), 1);
        }

        return $str;
    }

/*
* Function: _encode

```

```

* Modified for DX_Auth
* Original Author: FreakAuth_light 1.1
*/
function _encode($password)
{
    $majorsalt = $this->ci->config->item('DX_salt');

    // if PHP5
    if (function_exists('str_split'))
    {
        $_pass = str_split($password);
    }
    // if PHP4
    else
    {
        $_pass = array();
        if (is_string($password))
        {
            for ($i = 0; $i < strlen($password); $i++)
            {
                array_push($_pass, $password[$i]);
            }
        }
    }

    // encrypts every single letter of the password
    foreach ($_pass as $_hashpass)
    {
        $majorsalt .= md5($_hashpass);
    }

    // encrypts the string combinations of every single encrypted letter
    // and finally returns the encrypted password
    return md5($majorsalt);
}

function _array_in_array($needle, $haystack)
{
    // Make sure $needle is an array for foreach
    if (! is_array($needle))
    {
        $needle = array($needle);
    }

    // For each value in $needle, return TRUE if in $haystack
    foreach ($needle as $pin)
    {
        if (in_array($pin, $haystack))
            return TRUE;
    }
}

```

```

// Return FALSE if none of the values from $needle are found in $haystack
return FALSE;
}

function _email($to, $from, $subject, $message)
{
    $this->ci->load->library('Email');
    $email = $this->ci->email;

    $email->from($from);
    $email->to($to);
    $email->subject($subject);
    $email->message($message);

    return $email->send();
}

// Set last ip and last login function when user login
function _set_last_ip_and_last_login($user_id)
{
    $data = array();

    if ($this->ci->config->item('DX_login_record_ip'))
    {
        $data['last_ip'] = $this->ci->input->ip_address();
    }

    if ($this->ci->config->item('DX_login_record_time'))
    {
        $data['last_login'] = date('Y-m-d H:i:s', time());
    }

    if (! empty($data))
    {
        // Load model
        $this->ci->load->model('dx_auth/users', 'users');
        // Update record
        $this->ci->users->set_user($user_id, $data);
    }
}

// Increase login attempt
function _increase_login_attempt()
{
    if ($this->ci->config->item('DX_count_login_attempts') AND ! $this->is_max_login_attempts_exceeded())
    {
        // Load model

```

```

        $this->ci->load->model('dx_auth/login_attempts',
'login_attempts');
        // Increase login attempts for current IP
        $this->ci->login_attempts->increase_attempt($this->ci->input-
>ip_address());
    }

    // Clear login attempts
    function _clear_login_attempts()
    {
        if ($this->ci->config->item('DX_count_login_attempts'))
        {
            // Load model
            $this->ci->load->model('dx_auth/login_attempts',
'login_attempts');
            // Clear login attempts for current IP
            $this->ci->login_attempts->clear_attempts($this->ci->input-
>ip_address());
        }

        // Get role data from database by id, used in _set_session() function
        // $parent_roles_id, $parent_roles_name is an array.
        function _get_role_data($role_id)
        {
            // Load models
            $this->ci->load->model('dx_auth/roles', 'roles');
            $this->ci->load->model('dx_auth/permissions', 'permissions');

            // Clear return value
            $role_name = "";
            $parent_roles_id = array();
            $parent_roles_name = array();
            $permission = array();
            $parent_permissions = array();

            /* Get role_name, parent_roles_id and parent_roles_name */

            // Get role query from role id
            $query = $this->ci->roles->get_role_by_id($role_id);

            // Check if role exist
            if ($query->num_rows() > 0)
            {
                // Get row
                $role = $query->row();

                // Get role name
                $role_name = $role->name;

```

Code below will search if user role\_id have parent\_id > 0  
 (which mean role\_id have parent role\_id)  
 and do it recursively until parent\_id reach 0 (no parent) or  
 parent\_id not found.

If anyone have better approach than this code, please let  
 me know.

```

    /*
    Code below will search if user role_id have parent_id > 0
    (which mean role_id have parent role_id)
    and do it recursively until parent_id reach 0 (no parent) or
    parent_id not found.

    If anyone have better approach than this code, please let
    me know.
    */

    // Check if role has parent id
    if ($role->parent_id > 0)
    {
        // Add to result array
        $parent_roles_id[] = $role->parent_id;

        // Set variable used in looping
        $finished = FALSE;
        $parent_id = $role->parent_id;

        // Get all parent id
        while ($finished == FALSE)
        {
            $i_query = $this->ci->roles-
>get_role_by_id($parent_id);

            // If role exist
            if ($i_query->num_rows() > 0)
            {
                // Get row
                $i_role = $i_query->row();

                // Check if role doesn't have parent
                if ($i_role->parent_id == 0)
                {
                    // Get latest parent name
                    $parent_roles_name[] = $i_role-
>name;

                    // Stop looping
                    $finished = TRUE;
                }
                else
                {
                    // Change parent id for next looping
                    $parent_id = $i_role->parent_id;

                    // Add to result array
                    $parent_roles_id[] = $parent_id;
                }
            }
        }
    }
  
```

```

                                $parent_roles_name[] = $i_role-
>name;                                }
                                }
                                else
                                {
                                // Remove latest parent_roles_id since
parent_id not found                array_pop($parent_roles_id);
                                // Stop looping
                                $finished = TRUE;
                                }
                                }
                                }
                                }
}

/* End of Get role_name, parent_roles_id and parent_roles_name */

/* Get user and parents permission */

// Get user role permission
$permission = $this->ci->permissions->get_permission_data($role_id);

// Get user role parent permissions
if ( ! empty($parent_roles_id))
{
    $parent_permissions = $this->ci->permissions-
>get_permissions_data($parent_roles_id);
}

/* End of Get user and parents permission */

// Set return value
$data['role_name'] = $role_name;
$data['parent_roles_id'] = $parent_roles_id;
$data['parent_roles_name'] = $parent_roles_name;
$data['permission'] = $permission;
$data['parent_permissions'] = $parent_permissions;

return $data;
}

/* Autologin related function */

function _create_autologin($user_id)
{
    $result = FALSE;

    // User wants to be remembered
    $user = array(

```

```

        'key_id' => substr(md5(uniqid(rand()).$this->ci->input-
>cookie($this->ci->config->item('sess_cookie_name'))), 0, 16),
        'user_id' => $user_id
    );

    // Load Models
    $this->ci->load->model('dx_auth/user_autologin', 'user_autologin');

    // Prune keys
    $this->ci->user_autologin->prune_keys($user['user_id']);

    if ($this->ci->user_autologin->store_key($user['key_id'],
$user['user_id']))
    {
        // Set Users AutoLogin cookie
        $this->_auto_cookie($user);

        $result = TRUE;
    }

    return $result;
}

function autologin()
{
    $result = FALSE;

    if ($auto = $this->ci->input->cookie($this->ci->config-
>item('DX_autologin_cookie_name')) AND !$this->ci->session-
>userdata('DX_logged_in'))
    {
        // Extract data
        $auto = unserialize($auto);

        if (isset($auto['key_id']) AND $auto['key_id'] AND $auto['user_id'])
        {
            // Load Models
            $this->ci->load->model('dx_auth/user_autologin',
'user_autologin');

            // Get key
            $query = $this->ci->user_autologin-
>get_key($auto['key_id'], $auto['user_id']);

            if ($result = $query->row())
            {
                // User verified, log them in
                $this->_set_session($result);
                // Renew users cookie to prevent it from expiring

```

```

        $this->_auto_cookie($auto);

        // Set last ip and last login
        $this->_set_last_ip_and_last_login($auto['user_id']);

        $result = TRUE;
    }
}

return $result;
}

function _delete_autologin()
{
    if ($auto = $this->ci->input->cookie($this->ci->config-
>item('DX_autologin_cookie_name')))
    {
        // Load Cookie Helper
        $this->ci->load->helper('cookie');

        // Load Models
        $this->ci->load->model('dx_auth/user_autologin',
'user_autologin');

        // Extract data
        $auto = unserialize($auto);

        // Delete db entry
        $this->ci->user_autologin->delete_key($auto['key_id'],
$auto['user_id']);

        // Make cookie expired
        set_cookie($this->ci->config-
>item('DX_autologin_cookie_name'), "", -1);
    }
}

function _set_session($data)
{
    // Get role data
    $role_data = $this->_get_role_data($data->role_id);

    // Set session data array
    $user = array(
        'DX_user_id'           => $data->id,
        'DX_username'         => $data-
>username,
        'DX_role_id'         => $data->role_id,

```

```

        'DX_role_name' =>
$role_data['role_name'],
        'DX_parent_roles_id' =>
$role_data['parent_roles_id'], // Array of parent role_id
        'DX_parent_roles_name' => $role_data['parent_roles_name'], //
Array of parent role_name
        'DX_permission' =>
$role_data['permission'],
        'DX_parent_permissions' => $role_data['parent_permissions'],

        'DX_logged_in' => TRUE
    );

    $this->ci->session->set_userdata($user);
}

function _auto_cookie($data)
{
    // Load Cookie Helper
    $this->ci->load->helper('cookie');

    $cookie = array(
        'name' => $this->ci->config-
>item('DX_autologin_cookie_name'),
        'value' => serialize($data),
        'expire' => $this->ci->config-
>item('DX_autologin_cookie_life')
    );

    set_cookie($cookie);
}

/* End of Auto login related function */

/* Helper function */

function check_uri_permissions($allow = TRUE)
{
    // First check if user already logged in or not
    if ($this->is_logged_in())
    {
        // If user is not admin
        if ( ! $this->is_admin())
        {
            // Get variable from current URI
            $controller = '/' . $this->ci->uri->rsegment(1) . '/';
            if ($this->ci->uri->rsegment(2) != "")
            {
                $action = $controller . $this->ci->uri->rsegment(2) . '/';
            }
        }
    }
}

```

```

else
{
    $action = $controller.'index/';
}

// Get URI permissions from role and all parents
// Note: URI permissions is saved in 'uri' key
$roles_allowed_uris = $this->get_permissions_value('uri');

// Variable to determine if URI found
$have_access = ! $allow;
// Loop each roles URI permissions
foreach ($roles_allowed_uris as $allowed_uris)
{
    if ($allowed_uris != NULL)
    {
        // Check if user allowed to access URI
        if ($this->_array_in_array(array('/',
$controller, $action), $allowed_uris))
        {
            $have_access = $allow;
            // Stop loop
            break;
        }
    }
}

// Trigger event
$this->ci->dx_auth_event-
>checked_uri_permissions($this->get_user_id(), $have_access);

if ( ! $have_access)
{
    // User didn't have privilege to access current URI,
so we show user 403 forbidden access
    $this->deny_access();
}
}
}
else
{
    // User haven't logged in, so just redirect user to login page
    $this->deny_access('login');
}
}
}
/*

```

Get permission value from specified key.  
Call this function only when user is logged in already.

\$key is permission array key (Note: permissions is saved as array in table).

If \$check\_parent is TRUE means if permission value not found in user role, it will try to get permission value from parent role.

Returning value if permission found, otherwise returning NULL

```

*/
function get_permission_value($key, $check_parent = TRUE)
{
    // Default return value
    $result = NULL;

    // Get current user permission
    $permission = $this->ci->session->userdata('DX_permission');

    // Check if key is in user permission array
    if (array_key_exists($key, $permission))
    {
        $result = $permission[$key];
    }
    // Key not found
    else
    {
        if ($check_parent)
        {
            // Get current user parent permissions
            $parent_permissions = $this->ci->session->
            >userdata('DX_parent_permissions');

            // Check parent permissions array

            foreach ($parent_permissions as $permission)
            {
                if (array_key_exists($key, $permission))
                {
                    $result = $permission[$key];
                    break;
                }
            }
        }
    }

    // Trigger event
    $this->ci->dx_auth_event->got_permission_value($this->get_user_id(),
    $key);

    return $result;
}
/*
    Get permissions value from specified key.

```

Call this function only when user is logged in already.  
This will get user permission, and it's parents permissions.

`$array_key = 'default'`. Array ordered using 0, 1, 2 as array key.  
`$array_key = 'role_id'`. Array ordered using `role_id` as array key.  
`$array_key = 'role_name'`. Array ordered using `role_name` as array key.

Returning array of value if permission found, otherwise returning NULL.

```

*/
function get_permissions_value($key, $array_key = 'default')
{
    $result = array();

    $role_id = $this->ci->session->userdata('DX_role_id');
    $role_name = $this->ci->session->userdata('DX_role_name');

    $parent_roles_id = $this->ci->session->userdata('DX_parent_roles_id');
    $parent_roles_name = $this->ci->session-
>userdata('DX_parent_roles_name');

    // Get current user permission
    $value = $this->get_permission_value($key, FALSE);

    if ($array_key == 'role_id')
    {
        $result[$role_id] = $value;
    }
    elseif ($array_key == 'role_name')
    {
        $result[$role_name] = $value;
    }
    else
    {
        array_push($result, $value);
    }

    // Get current user parent permissions
    $parent_permissions = $this->ci->session-
>userdata('DX_parent_permissions');

    $i = 0;
    foreach ($parent_permissions as $permission)
    {
        if (array_key_exists($key, $permission))
        {
            $value = $permission[$key];
        }

        if ($array_key == 'role_id')
        {

```

```

        // It's safe to use $parents_roles_id[$i] because array
order is same with permission array
        $result[$parent_roles_id[$i]] = $value;
    }
    elseif ($array_key == 'role_name')
    {
        // It's safe to use $parents_roles_name[$i] because array
order is same with permission array
        $result[$parent_roles_name[$i]] = $value;
    }
    else
    {
        array_push($result, $value);
    }

    $i++;
}

// Trigger event
$this->ci->dx_auth_event->got_permissions_value($this-
>get_user_id(), $key);

return $result;
}

function deny_access($uri = 'deny')
{
    $this->ci->load->helper('url');

    if ($uri == 'login')
    {
        redirect($this->ci->config->item('DX_login_uri'), 'location');
    }
    else if ($uri == 'banned')
    {
        redirect($this->ci->config->item('DX_banned_uri'), 'location');
    }
    else
    {
        redirect($this->ci->config->item('DX_deny_uri'), 'location');
    }
    exit;
}

// Get user id
function get_user_id()
{
    return $this->ci->session->userdata('DX_user_id');
}

```

```

// Get username string
function get_username()
{
    return $this->ci->session->userdata('DX_username');
}

// Get user role id
function get_role_id()
{
    return $this->ci->session->userdata('DX_role_id');
}

// Get user role name
function get_role_name()
{
    return $this->ci->session->userdata('DX_role_name');
}

// Check is user is has admin privilege
function is_admin()
{
    return strtolower($this->ci->session->userdata('DX_role_name')) ==
'admin';
}

// Check if user has $roles privilege
// If $use_role_name TRUE then $roles is name such as 'admin', 'editor', 'etc'
// else $roles is role_id such as 0, 1, 2
// If $check_parent is TRUE means if roles not found in user role, it will check if
user role parent has that roles
function is_role($roles = array(), $use_role_name = TRUE, $check_parent =
TRUE)
{
    // Default return value
    $result = FALSE;

    // Build checking array
    $check_array = array();

    if ($check_parent)
    {
        // Add parent roles into check array
        if ($use_role_name)
        {
            $check_array = $this->ci->session-
>userdata('DX_parent_roles_name');
        }
        else
        {

```

```

        $check_array = $this->ci->session-
>userdata('DX_parent_roles_id');
    }
}

// Add current role into check array
if ($use_role_name)
{
    array_push($check_array, $this->ci->session-
>userdata('DX_role_name'));
}
else
{
    array_push($check_array, $this->ci->session-
>userdata('DX_role_id'));
}

// If $roles not array then we add it into an array
if (! is_array($roles))
{
    $roles = array($roles);
}

if ($use_role_name)
{
    // Convert check array into lowercase since we want case
insensitive checking
    for ($i = 0; $i < count($check_array); $i++)
    {
        $check_array[$i] = strtolower($check_array[$i]);
    }

    // Convert roles into lowercase since we want insensitive
checking
    for ($i = 0; $i < count($roles); $i++)
    {
        $roles[$i] = strtolower($roles[$i]);
    }

    // Check if roles exist in check_array
    if ($this->_array_in_array($roles, $check_array))
    {
        $result = TRUE;
    }

    return $result;
}

// Check if user is logged in

```

```

function is_logged_in()
{
    return $this->ci->session->userdata('DX_logged_in');
}

// Check if user is a banned user, call this only after calling login() and
returning FALSE
function is_banned()
{
    return $this->_banned;
}

// Get ban reason, call this only after calling login() and returning FALSE
function get_ban_reason()
{
    return $this->_ban_reason;
}

// Check if username is available to use, by making sure there is no same
username in the database
function is_username_available($username)
{
    // Load Models
    $this->ci->load->model('dx_auth/users', 'users');
    $this->ci->load->model('dx_auth/user_temp', 'user_temp');

    $users = $this->ci->users->check_username($username);
    $temp = $this->ci->user_temp->check_username($username);

    return $users->num_rows() + $temp->num_rows() == 0;
}

// Check if email is available to use, by making sure there is no same email in
the database
function is_email_available($email)
{
    // Load Models
    $this->ci->load->model('dx_auth/users', 'users');
    $this->ci->load->model('dx_auth/user_temp', 'user_temp');

    $users = $this->ci->users->check_email($email);
    $temp = $this->ci->user_temp->check_email($email);

    return $users->num_rows() + $temp->num_rows() == 0;
}

// Check if login attempts bigger than max login attempts specified in config
function is_max_login_attempts_exceeded()
{
    $this->ci->load->model('dx_auth/login_attempts', 'login_attempts');
}

```

```

        return ($this->ci->login_attempts->check_attempts($this->ci->input-
>ip_address()->num_rows() >= $this->ci->config->item('DX_max_login_attempts'));
    }

    function get_auth_error()
    {
        return $this->_auth_error;
    }

    /* End of Helper function */

    /* Main function */

    // $login is username or email or both depending on setting in config file
    function login($login, $password, $remember = TRUE)
    {
        // Load Models
        $this->ci->load->model('dx_auth/users', 'users');
        $this->ci->load->model('dx_auth/user_temp', 'user_temp');
        $this->ci->load->model('dx_auth/login_attempts', 'login_attempts');

        // Default return value
        $result = FALSE;

        if ( ! empty($login) AND ! empty($password))
        {
            // Get which function to use based on config
            if ($this->ci->config->item('DX_login_using_username') AND
$this->ci->config->item('DX_login_using_email'))
            {
                $get_user_function = 'get_login';
            }
            else if ($this->ci->config->item('DX_login_using_email'))
            {
                $get_user_function = 'get_user_by_email';
            }
            else
            {
                $get_user_function = 'get_user_by_username';
            }

            // Get user query
            if ($query = $this->ci->users->$get_user_function($login) AND
$query->num_rows() == 1)
            {
                // Get user record
                $row = $query->row();

                // Check if user is banned or not

```

```

if ($row->banned > 0)
{
    // Set user as banned
    $this->_banned = TRUE;

    // Set ban reason
    $this->_ban_reason = $row->ban_reason;
}
// If it's not a banned user then try to login
else
{
    $password = $this->_encode($password);
    $stored_hash = $row->password;

    // Is password matched with hash in database ?
    if (crypt($password, $stored_hash) ===
$stored_hash)
    {
        // Log in user
        $this->_set_session($row);

        if ($row->newpass)
        {
            // Clear any Reset Passwords
            $this->ci->users-
>clear_newpass($row->id);
        }

        if ($remember)
        {
            // Create auto login if user want to be
remembered
            $this->_create_autologin($row->id);
        }

        // Set last ip and last login
        $this->_set_last_ip_and_last_login($row-
>id);

        // Clear login attempts
        $this->_clear_login_attempts();

        // Trigger event
        $this->ci->dx_auth_event-
>user_logged_in($row->id);

        // Set return value
        $result = TRUE;
    }
else

```

```

        {
            // Increase login attempts
            $this->_increase_login_attempt();
            // Set error message
            $this->_auth_error = $this->ci->lang-
>line('auth_login_incorrect_password');
        }
    }
    // Check if login is still not activated
    elseif ($query = $this->ci->user_temp-
>$get_user_function($login) AND $query->num_rows() == 1)
    {
        // Set error message
        $this->_auth_error = $this->ci->lang-
>line('auth_not_activated');
    }
    else
    {
        // Increase login attempts
        $this->_increase_login_attempt();
        // Set error message
        $this->_auth_error = $this->ci->lang-
>line('auth_login_username_not_exist');
    }
}

return $result;
}

function logout()
{
    // Trigger event
    $this->ci->dx_auth_event->user_logging_out($this->ci->session-
>userdata('DX_user_id'));

    // Delete auto login
    if ($this->ci->input->cookie($this->ci->config-
>item('DX_autologin_cookie_name'))) {
        $this->_delete_autologin();
    }

    // Destroy session
    $this->ci->session->sess_destroy();
}

function register($username, $password, $email)
{
    // Load Models
    $this->ci->load->model('dx_auth/users', 'users');
}

```

```

$this->ci->load->model('dx_auth/user_temp', 'user_temp');

$this->ci->load->helper('url');

// Default return value
$result = FALSE;

// New user array
$new_user = array(
    'username'           => $username,
    'password'          => crypt($this-
>_encode($password)),
    'email'             => $email,
    'last_ip'          => $this->ci->input-
>ip_address()
);

// Do we need to send email to activate user
if ($this->ci->config->item('DX_email_activation'))
{
    // Add activation key to user array
    $new_user['activation_key'] = md5(rand().microtime());

    // Create temporary user in database which means the user still
unactivated.
    $insert = $this->ci->user_temp->create_temp($new_user);
}
else
{
    // Create user
    $insert = $this->ci->users->create_user($new_user);
    // Trigger event
    $this->ci->dx_auth_event->user_activated($this->ci->db-
>insert_id());
}

if ($insert)
{
    // Replace password with plain for email
    $new_user['password'] = $password;

    $result = $new_user;

    // Send email based on config

    // Check if user need to activate it's account using email
    if ($this->ci->config->item('DX_email_activation'))
    {
        // Create email

```

```

        $from = $this->ci->config->item('DX_webmaster_email');
        $subject = sprintf($this->ci->lang-
>line('auth_activate_subject'), $this->ci->config->item('DX_website_name'));

        // Activation Link
        $new_user['activate_url'] = site_url($this->ci->config-
>item('DX_activate_uri')."{ $new_user['username']}/{ $new_user['activation_key']}");

        // Trigger event and get email content
        $this->ci->dx_auth_event-
>sending_activation_email($new_user, $message);

        // Send email with activation link
        $this->_email($email, $from, $subject, $message);
    }
    else
    {
        // Check if need to email account details

        if ($this->ci->config->item('DX_email_account_details'))
        {
            // Create email
            $from = $this->ci->config-
>item('DX_webmaster_email');
            $subject = sprintf($this->ci->lang-
>line('auth_account_subject'), $this->ci->config->item('DX_website_name'));

            // Trigger event and get email content
            $this->ci->dx_auth_event-
>sending_account_email($new_user, $message);

            // Send email with account details
            $this->_email($email, $from, $subject, $message);
        }
    }
}

return $result;
}

function forgot_password($login)
{
    // Default return value
    $result = FALSE;

    if ($login)
    {
        // Load Model

```

```

$this->ci->load->model('dx_auth/users', 'users');
// Load Helper
$this->ci->load->helper('url');

// Get login and check if it's exist
if ($query = $this->ci->users->get_login($login) AND $query-
>num_rows() == 1)
{
    // Get User data
    $row = $query->row();

    // Check if there is already new password created but
waiting to be activated for this login
    if ( ! $row->newpass_key)
    {
        // Apparently there is no password created yet for
this login, so we create new password
        $data['password'] = $this->_gen_pass();

        // Encode & Crypt password
        $encode = crypt($this-
>_encode($data['password']));

        // Create key
        $data['key'] = md5(rand().microtime());

        // Create new password (but it haven't activated yet)
        $this->ci->users->newpass($row->id, $encode,
$data['key']);

        // Create reset password link to be included in email
        $data['reset_password_uri'] = site_url($this->ci-
>config->item('DX_reset_password_uri')."{ $row->username }/{ $data['key'] }");

        // Create email
        $from = $this->ci->config-
>item('DX_webmaster_email');
        $subject = $this->ci->lang-
>line('auth_forgot_password_subject');

        // Trigger event and get email content
        $this->ci->dx_auth_event-
>sending_forgot_password_email($data, $message);

        // Send instruction email
        $this->_email($row->email, $from, $subject,
$message);

        $result = TRUE;
    }
}

```

```

        else
        {
            // There is already new password waiting to be
            $this->_auth_error = $this->ci->lang-
            >line('auth_request_sent');
        }
    }
    else
    {
        $this->_auth_error = $this->ci->lang-
        >line('auth_username_or_email_not_exist');
    }
}

return $result;
}

function reset_password($username, $key = "")
{
    // Load Models
    $this->ci->load->model('dx_auth/users', 'users');
    $this->ci->load->model('dx_auth/user_autologin', 'user_autologin');

    // Default return value
    $result = FALSE;

    // Default user_id set to none
    $user_id = 0;

    // Get user id
    if ($query = $this->ci->users->get_user_by_username($username)
    AND $query->num_rows() == 1)
    {
        $user_id = $query->row()->id;

        // Try to activate new password
        if (! empty($username) AND ! empty($key) AND $this->ci-
        >users->activate_newpass($user_id, $key) AND $this->ci->db->affected_rows() > 0 )
        {
            // Clear previously setup new password and keys
            $this->ci->user_autologin->clear_keys($user_id);

            $result = TRUE;
        }
    }
    return $result;
}

function activate($username, $key = "")

```

```

    {
        // Load Models
        $this->ci->load->model('dx_auth/users', 'users');
        $this->ci->load->model('dx_auth/user_temp', 'user_temp');

        // Default return value
        $result = FALSE;

        if ($this->ci->config->item('DX_email_activation'))
        {
            // Delete user whose account expired (not activated until expired
time)
            $this->ci->user_temp->prune_temp();
        }

        // Activate user
        if ($query = $this->ci->user_temp->activate_user($username, $key)
AND $query->num_rows() > 0)
        {
            // Get user
            $row = $query->row_array();

            $del = $row['id'];

            // Unset any unwanted fields
            unset($row['id']); // We don't want to copy the id across
            unset($row['activation_key']);

            // Create user
            if ($this->ci->users->create_user($row))
            {
                // Trigger event
                $this->ci->dx_auth_event->user_activated($this->ci->db-
>insert_id());

                // Delete user from temp
                $this->ci->user_temp->delete_user($del);

                $result = TRUE;
            }
        }

        return $result;
    }

function change_password($old_pass, $new_pass)
{
    // Load Models
    $this->ci->load->model('dx_auth/users', 'users');

```

```

// Default return value
$result = FALSE;

// Search current logged in user in database
if ($query = $this->ci->users->get_user_by_id($this->ci->session-
>userdata('DX_user_id')) AND $query->num_rows() > 0)
{
    // Get current logged in user
    $row = $query->row();

    $pass = $this->_encode($old_pass);

    // Check if old password correct
    if (crypt($pass, $row->password) === $row->password)
    {
        // Crypt and encode new password
        $new_pass = crypt($this->_encode($new_pass));

        // Replace old password with new password
        $this->ci->users->change_password($this->ci->session-
>userdata('DX_user_id'), $new_pass);

        // Trigger event
        $this->ci->dx_auth_event-
>user_changed_password($this->ci->session->userdata('DX_user_id'), $new_pass);

        $result = TRUE;
    }
    else
    {
        $this->_auth_error = $this->ci->lang-
>line('auth_incorrect_old_password');
    }
}

return $result;
}

function cancel_account($password)
{
    // Load Models
    $this->ci->load->model('dx_auth/users', 'users');

    // Default return value
    $result = FALSE;

    // Search current logged in user in database
    if ($query = $this->ci->users->get_user_by_id($this->ci->session-
>userdata('DX_user_id')) AND $query->num_rows() > 0)
    {

```

```

// Get current logged in user
$row = $query->row();

$pass = $this->_encode($password);

// Check if password correct
if (crypt($pass, $row->password) === $row->password)
{
    // Trigger event
    $this->ci->dx_auth_event->user_canceling_account($this-
>ci->session->userdata('DX_user_id'));

    // Delete user
    $result = $this->ci->users->delete_user($this->ci-
>session->userdata('DX_user_id'));

    // Force logout
    $this->logout();
}
else
{
    $this->_auth_error = $this->ci->lang-
>line('auth_incorrect_password');
}
}

return $result;
}

/* End of main function */

/* Captcha related function */

function captcha()
{
    $this->ci->load->helper('url');

    $this->ci->load->plugin('dx_captcha');

    $captcha_dir = trim($this->ci->config->item('DX_captcha_path'), './');

    $vals = array(
        'img_path'          => './.$captcha_dir./',
        'img_url'           => base_url().$captcha_dir./',
        'font_path'         => $this->ci->config-
>item('DX_captcha_fonts_path'),
        'font_size'        => $this->ci->config->item('DX_captcha_font_size'),
        'img_width'         => $this->ci->config-
>item('DX_captcha_width'),
        'img_height'        => $this->ci->config->item('DX_captcha_height'),

```

```

        'show_grid'          => $this->ci->config-
>item('DX_captcha_grid'),
        'expiration'       => $this->ci->config->item('DX_captcha_expire')
    );

    $cap = create_captcha($vals);

    $store = array(
        'captcha_word' => $cap['word'],
        'captcha_time' => $cap['time']
    );

    // Plain, simple but effective
    $this->ci->session->set_flashdata($store);

    // Set our captcha
    $this->_captcha_image = $cap['image'];
}

function get_captcha_image()
{
    return $this->_captcha_image;
}

// Check if captcha already expired
// Use this in callback function in your form validation
function is_captcha_expired()
{
    // Captcha Expired
    list($usec, $sec) = explode(" ", microtime());
    $now = ((float)$usec + (float)$sec);

    // Check if captcha already expired
    return (($this->ci->session->flashdata('captcha_time') + $this->ci-
>config->item('DX_captcha_expire')) < $now);
}

// Check is captcha match with code
// Use this in callback function in your form validation
function is_captcha_match($code)
{
    if ($this->ci->config->item('DX_captcha_case_sensitive'))
    {
        // Just check if code is the same value with flash data
        captcha_word which created in captcha() function
        $result = ($code == $this->ci->session-
>flashdata('captcha_word'));
    }
    else
    {

```

```

        $result = strtolower($code) == strtolower($this->ci->session-
>flashdata('captcha_word'));
    }

    return $result;
}

/* End of captcha related function */

/* Recaptcha function */

function get_recaptcha_reload_link($text = 'Get another CAPTCHA')
{
    return '<a href="javascript:Recaptcha.reload()">'.$text.'</a>';
}

function get_recaptcha_switch_image_audio_link($switch_image_text = 'Get
an image CAPTCHA', $switch_audio_text = 'Get an audio CAPTCHA')
{
    return '<div class="recaptcha_only_if_image"><a
href="javascript:Recaptcha.switch_type(\'audio\')">'.$switch_audio_text.'</a></div>
<div class="recaptcha_only_if_audio"><a
href="javascript:Recaptcha.switch_type(\'image\')">'.$switch_image_text.'</a></div>';
}

function get_recaptcha_label($image_text = 'Enter the words above',
$audio_text = 'Enter the numbers you hear')
{
    return '<span class="recaptcha_only_if_image">'.$image_text.'</span>
<span class="recaptcha_only_if_audio">'.$audio_text.'</span>';
}

// Get captcha image
function get_recaptcha_image()
{
    return '<div id="recaptcha_image"></div>';
}

// Get captcha input box
// IMPORTANT: You should at least use this function when showing captcha
even for testing, otherwise reCAPTCHA image won't show up
// because reCAPTCHA javascript will try to find input type with
id="recaptcha_response_field" and name="recaptcha_response_field"
function get_recaptcha_input()
{
    return '<input type="text" id="recaptcha_response_field"
name="recaptcha_response_field" />';
}

// Get recaptcha javascript and non javascript html

```

// IMPORTANT: you should put call this function the last, after you are using some of get\_recaptcha\_xxx function above.

```
function get_recaptcha_html()
{
    // Load reCAPTCHA helper function
    $this->ci->load->helper('recaptcha');

    // Add custom theme so we can get only image
    $options = "<script>
        var RecaptchaOptions = {
            theme: 'custom',
            custom_theme_widget: 'recaptcha_widget'
        };
    </script>";

    // Get reCAPTCHA javascript and non javascript HTML
    $html = recaptcha_get_html($this->ci->config-
>item('DX_recaptcha_public_key'));

    return $options.$html;
}

// Check if entered captcha code match with the image.
// Use this in callback function in your form validation
function is_recaptcha_match()
{
    $this->ci->load->helper('recaptcha');

    $resp = recaptcha_check_answer($this->ci->config-
>item('DX_recaptcha_private_key'),
        $_SERVER["REMOTE_ADDR"],
        $_POST["recaptcha_challenge_field"],
        $_POST["recaptcha_response_field"]);

    return $resp->is_valid;
}

/* End of Recaptcha function */
}
?>
```

### **\\system\application\libraries\DX\_Auth\_Event.php**

```
<?php if (!defined('BASEPATH')) exit('No direct script access allowed');
```

```
// Event for DX_Auth
```

```
// You can use DX_Auth_Event to extend DX_Auth to fulfill your needs
```

```
// For example: you can use event below to PM user when he already activated the account, etc.
```

```

class DX_Auth_Event
{
    var $ci;

    function DX_Auth_Event()
    {
        $this->ci =& get_instance();
    }

    // If DX_email_activation in config is TRUE,
    // this event occurs after user succesfully activated using specified key in their
    email.
    // If DX_email_activation in config is FALSE,
    // this event occurs right after user succesfully registered.
    function user_activated($user_id)
    {
        // Load models
        $this->ci->load->model('dx_auth/user_profile', 'user_profile');

        // Create user profile
        $this->ci->user_profile->create_profile($user_id);
    }

    // This event occurs right after user login
    function user_logged_in($user_id)
    {
    }

    // This event occurs right before user logout
    function user_logging_out($user_id)
    {
    }

    // This event occurs right after user change password
    function user_changed_password($user_id, $new_password)
    {
    }

    // This event occurs right before user account is canceled
    function user_canceling_account($user_id)
    {
        // Load models
        $this->ci->load->model('dx_auth/user_profile', 'user_profile');

        // Delete user profile
        $this->ci->user_profile->delete_profile($user_id);
    }

    // This event occurs when check_uri_permissions() function in DX_Auth is
    called

```

```

// after checking if user role is allowed or not to access URI, this event will be
triggered
// $allowed is result of the check before, it's possible to alter the value since it's
passed by reference
function checked_uri_permissions($user_id, &$allowed)
{
}

// This event occurs when get_permission_value() function in DX_Auth is
called
function got_permission_value($user_id, $key)
{
}

// This event occurs when get_permissions_value() function in DX_Auth is
called
function got_permissions_value($user_id, $key)
{
}

// This event occurs right before dx auth send email with account details
// $data is an array, containing username, password, email and last ip
// $content is email content, passed by reference
// You can customize email content here
function sending_account_email($data, &$content)
{
    // Load helper
    $this->ci->load->helper('url');

    // Create content
    $content = sprintf($this->ci->lang->line('auth_account_content'),
        $this->ci->config->item('DX_website_name'),
        $data['username'],
        $data['email'],
        $data['password'],
        site_url($this->ci->config->item('DX_login_uri')),
        $this->ci->config->item('DX_website_name'));
}

// This event occurs right before dx auth send activation email
// $data is an array, containing username, password, email, last ip,
activation_key, activate_url
// $content is email content, passed by reference
// You can customize email content here
function sending_activation_email($data, &$content)
{
    // Create content
    $content = sprintf($this->ci->lang->line('auth_activate_content'),
        $this->ci->config->item('DX_website_name'),
        $data['activate_url'],

```

```

        $this->ci->config->item('DX_email_activation_expire') / 60 / 60,
        $data['username'],
        $data['email'],
        $data['password'],
        $this->ci->config->item('DX_website_name'));
    }

    // This event occurs right before dx auth send forgot password request email
    // $data is an array, containing password, key, and reset_password_uri
    // $content is email content, passed by reference
    // You can customize email content here
    function sending_forgot_password_email($data, &$content)
    {
        // Create content
        $content = sprintf($this->ci->lang-
>line('auth_forgot_password_content'),
            $this->ci->config->item('DX_website_name'),
            $data['reset_password_uri'],
            $data['password'],
            $data['key'],
            $this->ci->config->item('DX_webmaster_email'),
            $this->ci->config->item('DX_website_name'));
    }
}

?>

```

### **system\application\libraries\Funcoes\_Genericas.php**

```
<?php if (!defined('BASEPATH')) exit('No direct script access allowed');
```

```
class Funcoes_Genericas {
```

```
    /* Converte formato DATE do MySQL para o humano
    2003-12-30 -> 30/12/2003 */
```

```
function mysql_date_para_humano($dt) {
    if ($dt=="0000-00-00" || empty($dt)) return "";
    $yr=strval(substr($dt,0,4));
    $mo=strval(substr($dt,5,2));
    $da=strval(substr($dt,8,2));
    return date("d/m/Y", mktime (0,0,0,$mo,$da,$yr));
}

```

```
    /* Converte formato do DATETIME do MySQL para um compreensível para os
    homens
```

```
    2003-12-30 23:30:59 -> 30/12/2003 23:30:59 */
function mysql_datetime_para_humano($dt) {
    $yr=strval(substr($dt,0,4));
    $mo=strval(substr($dt,5,2));
    $da=strval(substr($dt,8,2));
    $hr=strval(substr($dt,11,2));
}

```

```

        $mi=strval(substr($dt,14,2));
        $se=strval(substr($dt,17,2));
        return date("d/m/Y H:i:s", mktime ($hr,$mi,$se,$mo,$da,$yr));
    }

/* Converte formato TIMESTAMP do MySQL para o humano
20031230233029 -> 30/12/2003 23:30:59 */
function mysql_timestamp_para_humano($dt) {
    $yr=strval(substr($dt,0,4));
    $mo=strval(substr($dt,4,2));
    $da=strval(substr($dt,6,2));
    $hr=strval(substr($dt,8,2));
    $mi=strval(substr($dt,10,2));
    $se=strval(substr($dt,12,2));
    return date("d/m/Y H:i:s", mktime ($hr,$mi,$se,$mo,$da,$yr));
}

/* Converte formato TIMESTAMP do Unix para o humano
1072834230 -> 30/12/2003 23:30:59 */
function timestamp_para_humano($ts) {
    $d=getdate($ts);
    $yr=$d["year"];
    $mo=$d["mon"];
    $da=$d["mday"];
    $hr=$d["hours"];
    $mi=$d["minutes"];
    $se=$d["seconds"];
    return date("d/m/Y H:i:s", mktime($hr,$mi,$se,$mo,$da,$yr));
}

/* Converte o TIMESTAMP do Unix para o TIMESTAMP do MySQL
1072834230 -> 20031230233029 */
function timestamp_para_mysql_timestamp($ts) {
    $d=getdate($ts);
    $yr=$d["year"];
    $mo=$d["mon"];
    $da=$d["mday"];
    $hr=$d["hours"];
    $mi=$d["minutes"];
    $se=$d["seconds"];
    return
    sprintf("%04d%02d%02d%02d%02d%02d", $yr,$mo,$da,$hr,$mi,$se);
}

/* Converte o TIMESTAMP do Unix para o DATE do MySQL
1072834230 -> 2003-12-23 */
function timestamp_para_mysql_date($ts) {
    $d=getdate($ts);
    $yr=$d["year"];
    $mo=$d["mon"];

```

```

        $da=$d["mday"];
        return sprintf("%04d-%02d-%02d", $yr, $mo, $da);
    }

/* Converte o TIMESTAMP do Unix para o DATETIME do MySQL
   1072834230 -> 2003-12-23 23:30:59 */
function timestamp_para_mysql_datetime($ts) {
    $d=getdate($ts);
    $yr=$d["year"];
    $mo=$d["mon"];
    $da=$d["mday"];
    $hr=$d["hours"];
    $mi=$d["minutes"];
    $se=$d["seconds"];
    return sprintf("%04d-%02d-%02d
%02d:%02d:%02d", $yr, $mo, $da, $hr, $mi, $se);
}

/* Converte formato TIMESTAMP do MySQL para o TIMESTAMP do Unix
   20031230233029 -> 1072834230 */
function mysql_timestamp_para_timestamp($dt) {
    $yr=strval(substr($dt,0,4));
    $mo=strval(substr($dt,4,2));
    $da=strval(substr($dt,6,2));
    $hr=strval(substr($dt,8,2));
    $mi=strval(substr($dt,10,2));
    $se=strval(substr($dt,10,2));
    return mktime($hr, $mi, $se, $mo, $da, $yr);
}

/* Converte formato DATETIME do MySQL para o TIMESTAMP do Unix
   2003-12-30 23:30:59 -> 1072834230 */
function mysql_datetime_para_timestamp($dt) {
    $yr=strval(substr($dt,0,4));
    $mo=strval(substr($dt,5,2));
    $da=strval(substr($dt,8,2));
    $hr=strval(substr($dt,11,2));
    $mi=strval(substr($dt,14,2));
    $se=strval(substr($dt,17,2));
    return mktime($hr, $mi, $se, $mo, $da, $yr);
}

/* pegar um valor de data (em um formato qualquer) enviado por um usuário e
transformar no
   formato DATE do MySQL */
function mysql_cvdate($s) {
    if (empty($s)) { return ""; }

```

```

        return $this->timestamp_para_mysql_date($this->cvdate($s));
    }

    /* pegar um valor de data/hora (em um formato qualquer) enviado por um
usuário e transformar no
    formato DATETIME do MySQL */
    function mysql_cvdatetime($s) {
        if (empty($s)) { return ""; }
        return $this->timestamp_para_mysql_datetime($this->cvdate($s));
    }

    /* Calcula a diferença de tempo entre os valores $comeco e $fim e retorno em
português literal a quantidade
    de tempo da diferença entre os valores.
    nota: ambos os valores de entrada devem estar no formato timestamp UNIX:
*/
    function timeleft($comeco,$fim) {
        $dif=$fim-$comeco;
        $years=intval($dif/(60*60*24*365));
        $dif=$dif-($years*(60*60*24*365));
        $months=intval($dif/(60*60*24*30));
        $dif=$dif-($months*(60*60*24*30));
        $weeks=intval($dif/(60*60*24*7));
        $dif=$dif-($weeks*(60*60*24*7));
        $days=intval($dif/(60*60*24));
        $dif=$dif-($days*(60*60*24));
        $hours=intval($dif/(60*60));
        $dif=$dif-($hours*(60*60));
        $minutes=intval($dif/(60));
        $seconds=$dif-($minutes*60);
        $s="";
        if ($years<>0) $s.= $years." anos ";
        if ($months<>0) $s.= $months." meses ";
        if ($weeks<>0) $s.= $weeks.' semanas ';
        if ($days<>0) $s.= $days.' dias ';
        if ($hours<>0) $s.= $hours.' horas ';
        if ($minutes<>0) $s.= $minutes.' minutos ';
        if ($seconds<>0) $s.= $seconds.' segundos ';
        return $s;
    }

    /* Converte uma data humana para o formato TIMESTAMP do Unix, retorna
zero se houver erro.

    Suporta delimitadores de datas como traço, ponto, barra e espaço; nomes
de mês, ano com apenas 2 dígitos
    Exemplo de entradas que a função aceita:
    30-12-2003 23:30:59
    30-12-2003
    30 12 2003

```

```

30.12.03
30/dez/2003
30 dezembro 03
30 de dezembro de 03
30 de dezembro de 2003
30, dezembro de 2003
*/
function cvdate($s) {
    $delimiter="";
    $s = str_replace(' de ','/',strtolower($s));
    if (strpos($s,'-')>0) $delimiter='-';
    elseif (strpos($s,'/')>0) $delimiter='/';
    elseif (strpos($s,' ')>0) $delimiter=' ';
    elseif (strpos($s,',')>0) $delimiter='.';
    $s = str_replace(' ', $delimiter, $s);
    if (empty($delimiter)) return 0;
    $p1=strpos($s,$delimiter);
    $p2=strpos($s,$delimiter,$p1+1);
    $a=substr($s,$p2+1);
    $m=substr($s,$p1+1,$p2-($p1+1));
    $d=substr($s,0,$p1);
    if (intval($a)<100)
    {
        $a=(intval($a)>69)? strval(1900+intval($a)) :
strval(2000+intval($a));
    }
    if (intval($m)==0) // contém mês em extenso
    {
        return $this->cvdate_portugues($d,$m,$a);
    }else{
        return $this->cvdate_numerico($d,$m,$a);
    }
}

/* função auxiliar do cvdate */
function cvdate_portugues($d,$m,$y) {
    $d2=0; $m2=0; $y2=0;
    $d2=intval($d);
    $m=strtolower($m);
    switch(substr($m,0,3)) {
        case 'jan': $m2=1; break;
        case 'fev': $m2=2; break;
        case 'mar': $m2=3; break;
        case 'abr': $m2=4; break;
        case 'mai': $m2=5; break;
        case 'jun': $m2=6; break;
        case 'jul': $m2=7; break;
        case 'ago': $m2=8; break;
        case 'set': $m2=9; break;
        case 'out': $m2=10; break;
    }
}

```

```

        case 'nov': $m2=11; break;
        case 'dez': $m2=12; break;
    }
    $y2=intval($y);
    if (($d2==0)||($m2==0)||($y2==0)) return 0;
    return mktime(0,0,0,$m2,$d2,$y2);
}

/* função auxiliar do cvdate */
function cvdate_numerico($d,$m,$y) {
    $d2=0; $m2=0; $y2=0;
    $d2=intval($d);
    $m2=intval($m);
    $y2=intval($y);
    if (($d2==0)||($m2==0)||($y2==0)) return 0;
    return mktime(0,0,0,$m2,$d2,$y2);
}

function formatar_cpf_cnpj($campo, $formatado = true){
    //retira formato
    $codigoLimpo = ereg_replace("[ '-./ t]", "", $campo);
    // pega o tamanho da string menos os digitos verificadores
    $tamanho = (strlen($codigoLimpo) - 2);
    //verifica se o tamanho do código informado é válido
    if ($tamanho != 9 && $tamanho != 12){
        return false;
    }

    if ($formatado){
        // seleciona a máscara para cpf ou cnpj
        $mascara = ($tamanho == 9) ? '###.###.###-##' :
'##.###.###/####-##';

        $indice = -1;
        for ($i=0; $i < strlen($mascara); $i++) {
            if ($mascara[$i]=='#') $mascara[$i] =
$codigoLimpo[++$indice];
        }
        //retorna o campo formatado
        $retorno = $mascara;
    }else{
        //se não quer formatado, retorna o campo limpo
        $retorno = $codigoLimpo;
    }

    return $retorno;
}

```

```

function formatar_cep($campo, $formatado = true) {
    if (empty($campo)) {
        return false;
    }
    $codigoLimpo = $this->limpaCampo($campo);
    $codigoLimpo = substr($codigoLimpo, -8);

    if ($formatado) {
        $retorno = substr($codigoLimpo,0,5)."-
".substr($codigoLimpo,5,3);
    } else {
        $retorno = $codigoLimpo;
    }
    return $retorno;
}

// OS TELEFONES DEVEM SER ENTRADOS OU OBTIDOS NO
FORMATO 9999999999 EX: 2133334444 AO INVÉS DE 21 3333-4444
function masc_tel($TEL) {
    $caract = array(".", "-", "(", ")", "/");
    $TEL = str_replace($caract, "", $TEL);
    $tam = strlen($TEL);
    // COM CÓDIGO DE ÁREA NACIONAL E DO PAIS
    if ($tam > 10) {
        return "+" . substr($TEL,0,$tam-10) . (" .substr($TEL,$tam-10,2).")
".substr($TEL,$tam-8,4)."-".substr($TEL,-4);
    }
    if ($tam == 10) {
        // COM CÓDIGO DE ÁREA NACIONAL
        return (" .substr($TEL,0,2).") ".substr($TEL,2,4)."-
".substr($TEL,6,10);
    }
    if ($tam <= 8) {
        // SEM CÓDIGO DE ÁREA
        return substr($TEL,0,$tam-4)."-".substr($TEL,-4);
    }
}

function limpaCampo($valor) {
    return str_replace(array('-', '/', ':', ';', '(', ')', ' '), "", $valor);
}

/* Converte formato MOEDA para decimal no MySQL
1.500,23 -> 1500.23*/
function moeda_para_decimal_mysql($valor) {
    $valor = str_replace('.', "", $valor); // retira os separadores de milhar
    $valor = str_replace(',', '.', $valor); // troca virgula por ponto
    return $valor;
}

```

```

/* Converte decimal do MySQL para formato MOEDA
1500.23 -> 1.500,23*/
function mysql_decimal_para_moeda($valor) {
    return number_format($valor, 2, ",", ".");
}

function mysql_decimal_para_moeda_3casas($valor) {
    return number_format($valor, 3, ",", ".");
}

function convertArrayToObject($array) {
    $object = new stdClass();
    if (is_array($array) && count($array) > 0) {
        foreach ($array as $name=>$value) {
            $name = strtolower(trim($name));
            if (!empty($name)) {
                $object->$name = $value;
            }
        }
    }
    return $object;
}

function convertObjectToArray($object) {
    $array = array();
    if (is_object($object)) {
        $array = get_object_vars($object);
    }
    return $array;
}

function validaCNPJ($cnpj) {
    if (strlen($cnpj) <> 18) return 0;
    $soma1 = ($cnpj[0] * 5) + ($cnpj[1] * 4) + ($cnpj[3] * 3) + ($cnpj[4] * 2) +
($cnpj[5] * 9) +
    ($cnpj[7] * 8) + ($cnpj[8] * 7) + ($cnpj[9] * 6) + ($cnpj[11] * 5) + ($cnpj[12] *
4) +
    ($cnpj[13] * 3) + ($cnpj[14] * 2);
    $resto = $soma1 % 11;
    $digito1 = $resto < 2 ? 0 : 11 - $resto;

    $soma2 = ($cnpj[0] * 6) + ($cnpj[1] * 5) + ($cnpj[3] * 4) + ($cnpj[4] * 3) +
($cnpj[5] * 2) +
    ($cnpj[7] * 9) + ($cnpj[8] * 8) + ($cnpj[9] * 7) + ($cnpj[11] * 6) + ($cnpj[12] *
5) +
    ($cnpj[13] * 4) + ($cnpj[14] * 3) + ($cnpj[16] * 2);
    $resto = $soma2 % 11;
    $digito2 = $resto < 2 ? 0 : 11 - $resto;

    return (($cnpj[16] == $digito1) && ($cnpj[17] == $digito2));
}

```

```

}

function validaCPF($cpf) {
    // Verifica se o número digitado contém todos os dígitos
    $cpf = str_pad(ereg_replace('[^0-9]', '', $cpf), 11, '0', STR_PAD_LEFT);

    // Verifica se nenhuma das sequências abaixo foi digitada, caso seja,
    retorna falso
    if (strlen($cpf) != 11 || $cpf == '00000000000' || $cpf == '11111111111' ||
    $cpf == '22222222222' || $cpf == '33333333333' || $cpf == '44444444444' || $cpf ==
    '55555555555' || $cpf == '66666666666' || $cpf == '77777777777' || $cpf ==
    '88888888888' || $cpf == '99999999999') {
        return false;
    } else {
        // Calcula os números para verificar se o CPF é verdadeiro
        for ($t = 9; $t < 11; $t++) {
            for ($d = 0, $c = 0; $c < $t; $c++) {
                $d += $cpf{$c} * (($t + 1) - $c);
            }

            $d = ((10 * $d) % 11) % 10;

            if ($cpf{$c} != $d) {
                return false;
            }
        }
        return true;
    }
}

/**
 * Analisa uma data e retorna se é válida
 * @param $data (formato: 00/00/0000)
 * @return boolean
 */
function validaData($data) {
    if (strlen($data) == 10) {
        $data = explode("/", $data);

        if (checkdate($data[0], $data[1], $data[2])) {
            return true;
        } else {
            return false;
        }
    } else {
        return false;
    }
}

/**

```

```

* Analisa um cep e retorna se é válido
* @param $cep (formato: 00000-000)
* @return boolean
*/
function validaCEP($cep) {
    if (strlen($cep) == 9) {
        $cep = explode("-", $cep);

        if (strlen($cep[0]) == 5 && is_int($cep[0]) && strlen($cep[1]) == 3
&& is_int($cep[1])) {
            return true;
        } else {
            return false;
        }
    } else {
        return false;
    }
}

function xml2array($contents, $get_attributes=1, $priority = 'tag') {
    if(!$contents) return array();

    if(!function_exists('xml_parser_create')) {
        //print "xml_parser_create() function not found!";
        return array();
    }

    //Get the XML parser of PHP - PHP must have this module for the
parser to work
    $parser = xml_parser_create("");
    xml_parser_set_option($parser, XML_OPTION_TARGET_ENCODING,
"UTF-8"); # http://minutillo.com/steve/weblog/2004/6/17/php-xml-and-character-encodings-a-tale-of-sadness-rage-and-data-loss
    xml_parser_set_option($parser, XML_OPTION_CASE_FOLDING, 0);
    xml_parser_set_option($parser, XML_OPTION_SKIP_WHITE, 1);
    xml_parse_into_struct($parser, trim($contents), $xml_values);
    xml_parser_free($parser);

    if(!$xml_values) return;//Hmm...

    //Initializations
    $xml_array = array();
    $parents = array();
    $opened_tags = array();
    $arr = array();

    $current = &$xml_array; //Reference

    //Go through the tags.

```

```

    $repeated_tag_index = array();//Multiple tags with same name will be
turned into an array
    foreach($xml_values as $data) {
        unset($attributes,$value);//Remove existing values, or there will
be trouble

        //This command will extract these variables into the foreach
scope
        // tag(string), type(string), level(int), attributes(array).
        extract($data);//We could use the array by itself, but this cooler.

        $result = array();
        $attributes_data = array();

        if(isset($value)) {
            if($priority == 'tag') $result = $value;
            else $result['value'] = $value; //Put the value in a assoc
array if we are in the 'Attribute' mode
        }

        //Set the attributes too.
        if(isset($attributes) and $get_attributes) {
            foreach($attributes as $attr => $val) {
                if($priority == 'tag') $attributes_data[$attr] = $val;
                else $result['attr'][$attr] = $val; //Set all the attributes
in a array called 'attr'
            }
        }

        //See tag status and do the needed.
        if($type == "open") {//The starting of the tag '<tag>'
            $parent[$level-1] = &$current;
            if(!is_array($current) or (!in_array($tag,
array_keys($current)))) { //Insert New tag
                $current[$tag] = $result;
                if($attributes_data) $current[$tag.'_'.'attr'] =
$attributes_data;
                $repeated_tag_index[$tag.'_'.'$level'] = 1;

                $current = &$current[$tag];
            } else { //There was another element with the same tag
name
                if(isset($current[$tag][0])) {//If there is a 0th element
it is already an array
                    $current[$tag][$repeated_tag_index[$tag.'_'.'$level']] = $result;
                    $repeated_tag_index[$tag.'_'.'$level']++;

```

```

    } else { //This section will make the value an array if
multiple tags with the same name appear together
        $current[$tag] =
array($current[$tag],$result); //This will combine the existing item and the new item
together to make an array
        $repeated_tag_index[$tag.'_'.$level] = 2;

        if(isset($current[$tag.'_attr'])) { //The attribute
of the last(0th) tag must be moved as well
            $current[$tag]['0_attr'] =
$current[$tag.'_attr'];
            unset($current[$tag.'_attr']);
        }

    }
    $last_item_index =
$repeated_tag_index[$tag.'_'.$level]-1;
    $current = &$current[$tag][$last_item_index];
}

} elseif($type == "complete") { //Tags that ends in 1 line '<tag />'
//See if the key is already taken.
if(!isset($current[$tag])) { //New Key
    $current[$tag] = $result;
    $repeated_tag_index[$tag.'_'.$level] = 1;
    if($priority == 'tag' and $attributes_data)
$current[$tag.'_attr'] = $attributes_data;

} else { //If taken, put all things inside a list(array)
    if(isset($current[$tag][0]) and
is_array($current[$tag])) { //If it is already an array...

        // ...push the new element into that array.

        $current[$tag][$repeated_tag_index[$tag.'_'.$level]] = $result;

        if($priority == 'tag' and $get_attributes and
$attributes_data) {

            $current[$tag][$repeated_tag_index[$tag.'_'.$level] . '_attr'] = $attributes_data;
        }
        $repeated_tag_index[$tag.'_'.$level]++;

    } else { //If it is not an array...
        $current[$tag] = array($current[$tag],$result);
//...Make it an array using using the existing value and the new value
        $repeated_tag_index[$tag.'_'.$level] = 1;
        if($priority == 'tag' and $get_attributes) {

```

```

                    if(isset($current[$tag.'_attr'])) { //The
attribute of the last(0th) tag must be moved as well

                    $current[$tag]['0_attr'] =
$current[$tag.'_attr'];
                    unset($current[$tag.'_attr']);
                    }
                    if($attributes_data) {
                    $current[$tag][$repeated_tag_index[$tag.'_'.$level] . '_attr'] = $attributes_data;
                    }
                    $repeated_tag_index[$tag.'_'.$level]++; //0
and 1 index is already taken
                }
            }
        } elseif($type == 'close') { //End of tag '</tag>'
            $current = &$parent[$level-1];
        }
    }
    return($xml_array);
}
}
?>

```

### **\\system\application\models\categoria\_model.php**

```

<?php
class Categoria_Model extends Model {
    function Categoria_Model() {
        parent::Model();
        $this->load->model('produto_model');
    }

    function buscaCategoriaNoNivel($titulo, $nivel) {
        $this->db->select('cd_categoria');
        $this->db->from('sipec_categoria');
        $this->db->where('nr_nivel', $nivel);
        $this->db->where('de_titulo', $titulo);
        $query = $this->db->get();
        if ($query->num_rows() > 0) {
            $row = $query->row();
            return $row->cd_categoria;
        } else {
            return null;
        }
    }
}

```

```

function gravarCategorias($categorias, $fornecedor) {
    // verificando se possui subcategorias
    if (!array_key_exists("categoria", $categorias)) { // não possui
subcategorias
        $cdCategoria = $this-
>buscaCategoriaNoNivel($categorias["nome"], $categorias["nivel"]);

        if (empty($cdCategoria)) { // categoria não está no banco de
dados
            $categoria["cd_categoria"] = null;
            $categoria["de_titulo"] = $categorias["nome"];
            $categoria["nr_nivel"] = $categorias["nivel"];
            $categoria["cd_categoria_pai"] = null;
            $cdCategoria = $this->gravarCategoria($categoria);
        }

        if (array_key_exists("produto", $categorias)) {
            $produtos = $categorias["produto"];
            //echo "<pre>";print_r($produtos);exit;
            $this->produto_model->gravarProdutos($produtos,
$cdCategoria, $fornecedor);
        }
    } else { // possui subcategorias
        for ($i=0; $i<count($categorias);$i++) {
            $this->gravarCategorias($categorias[$i], $fornecedor);
            //echo "<pre>";print_r($categorias[$i]).<br>';
        }
    }
}

function gravarCategoria($categoria) {
    $this->db->insert('sipec_categoria', $categoria);
    return $this->db->insert_id();
}

function atualizarFornecedor($cd_fornecedor, $fornecedor) {
    $this->db->where('cd_fornecedor', $cd_fornecedor);
    $this->db->update('sipec_fornecedor', $fornecedor);
}

function excluirFornecedor($cd_fornecedor) {
    $this->db->delete('sipec_fornecedor', array('cd_fornecedor' =>
$cd_fornecedor));
}

function buscaCategoriaPeloFiltro($cd_categoria=null,
$cd_categoria_pai=null, $de_titulo=null, $nr_nivel=null) {
    // prepara os campos da pesquisa para enviar como filtros na lista
    $filtroEquals = array();
    if (!empty($cd_categoria)) {

```

```

        $filtroEquals['cd_categoria'] = $cd_categoria;
    }
    if (!empty($cd_categoria_pai)) {
        $filtroEquals['cd_categoria_pai'] = $cd_categoria_pai;
    }
    if (!empty($nr_nivel)) {
        $filtroEquals['nr_nivel'] = $nr_nivel;
    }

    $filtroLike = array();
    if (!empty($de_titulo)) {
        $filtroLike['de_titulo'] = $de_titulo;
    }

    // busca as categorias do banco de dados
    $this->db->select('*');
    $this->db->from('sipec_categoria');
    if (!empty($filtroEquals)) {
        $this->db->where($filtroEquals);
    }
    if (!empty($filtroLike)) {
        $this->db->like($filtroLike);
    }
    $this->db->orderby('nr_nivel asc, de_titulo asc');
    $query = $this->db->get();
    return $query->result();
}

function buscaFornecedorPeloCodigo($cd_fornecedor) {
    $this->db->select('*');
    $this->db->from('sipec_fornecedor');
    $this->db->where(array('cd_fornecedor' => $cd_fornecedor));
    $query = $this->db->get();
    return $query->result();
}

function cargaInicial() {
    $this->db->truncate('products_to_categories');
    $this->db->truncate('products_description');
    $this->db->truncate('products');

    $this->db->truncate('categories_description');
    $this->db->truncate('categories');

    // cadastrando categorias
    $categorias = $this->buscaCategoriaPeloFiltro("", "", "", 1);
    //echo "<pre>";print_r($categorias);exit;

    foreach ($categorias as $categoria) {
        $infoCategoria = array();
    }
}

```

```

$infoCategoria["categories_id"] = null;
$infoCategoria["categories_image"] = null;
$infoCategoria["parent_id"] = 0;
$infoCategoria["sort_order"] = 0;
$infoCategoria["date_added"] = date('Y-m-d H:i:s');
$infoCategoria["last_modified"] = null;
$this->db->insert('categories', $infoCategoria);
$categories_id = $this->db->insert_id();

$descCategoria = array();
$descCategoria["categories_id"] = $categories_id;
$descCategoria["language_id"] = 4; //portugues
$descCategoria["categories_name"] = $categoria->de_titulo;
$this->db->insert('categories_description', $descCategoria);

// cadastrando subcategorias
$subCategorias = $this->buscaCategoriaPeloFiltro("", $categoria-
>cd_categoria, "", 2);
//echo "<pre>";print_r($subCategorias);exit;

$parent_id = $categories_id;
foreach ($subCategorias as $subCategoria) {
    $infoCategoria = array();
    $infoCategoria["categories_id"] = null;
    $infoCategoria["categories_image"] = null;
    $infoCategoria["parent_id"] = $parent_id;
    $infoCategoria["sort_order"] = 0;
    $infoCategoria["date_added"] = date('Y-m-d H:i:s');
    $infoCategoria["last_modified"] = null;
    $this->db->insert('categories', $infoCategoria);
    $categories_id = $this->db->insert_id();

    $descCategoria = array();
    $descCategoria["categories_id"] = $categories_id;
    $descCategoria["language_id"] = 4; //portugues
    $descCategoria["categories_name"] = $subCategoria-
>de_titulo;
    $this->db->insert('categories_description',
$descCategoria);

    if (!empty($subCategoria->cd_categoria)) {
        $produtos = $this->produto_model-
>buscaProdutoPeloFiltro("", "", $subCategoria->cd_categoria, 1);
        if (count($produtos) > 0) {
            foreach ($produtos as $produto) {
                $infoProduto = array();
                $infoProduto["products_id"] = null;
                $infoProduto["products_quantity"] = 0;
                $infoProduto["products_model"] = null;
                $infoProduto["products_image"] = null;
            }
        }
    }
}

```

```

0.0000;
= date('Y-m-d H:i:s');
    $infoProducto["products_last_modified"] = null;
    $infoProducto["products_date_available"] = null;
0.00;
= 0;
$producto->nr_universal;
$infoProducto);
    $products_id;
    $producto->de_titulo;
    = null;
    >insert('products_description', $descProducto);
    $products_id;
    $categories_id;
    >insert('products_to_categories', $catProducto);
}
}
}
}
}
}
}
}
}
?>
    $infoProducto["products_price"] =
    $infoProducto["products_date_added"]
    $infoProducto["products_weight"] =
    $infoProducto["products_status"] = 0;
    $infoProducto["products_tax_class_id"]
    $infoProducto["products_ordered"] = 0;
    $infoProducto["products_ean_code"] =
    $infoProducto["manufacturers_id"] = 0;
    $this->db->insert('products',
    $products_id = $this->db->insert_id();
    $descProducto = array();
    $descProducto["products_id"] =
    $descProducto["products_name"] =
    $descProducto["products_description"]
    $descProducto["products_url"] = null;
    $descProducto["products_viewed"] = 0;
    $descProducto["language_id"] = 4;
    $this->db-
    $catProducto = array();
    $catProducto["products_id"] =
    $catProducto["categories_id"] =
    $this->db-

```

**\\system\application\models\fornecedor\_model.php**

```

<?php
class Fornecedor_Model extends Model {
    function Fornecedor_Model() {
        parent::Model();
    }

    function gravarFornecedor($fornecedor) {
        $this->db->insert('sipec_fornecedor', $fornecedor);
        return $this->db->insert_id();
    }

    function atualizarFornecedor($cd_fornecedor, $fornecedor) {
        $this->db->where('cd_fornecedor', $cd_fornecedor);
        $this->db->update('sipec_fornecedor', $fornecedor);
    }

    function excluirFornecedor($cd_fornecedor) {
        $this->db->delete('sipec_fornecedor', array('cd_fornecedor' =>
$cd_fornecedor));
    }

    function ativarFornecedor($cd_fornecedor) {
        $this->db->where('cd_fornecedor', $cd_fornecedor);
        $this->db->update('sipec_fornecedor', array('fl_situacao' => 1));
    }

    function inativarFornecedor($cd_fornecedor) {
        $this->db->where('cd_fornecedor', $cd_fornecedor);
        $this->db->update('sipec_fornecedor', array('fl_situacao' => 2));
    }

    function buscaFornecedorPeloFiltro($cd_fornecedor, $nm_fornecedor,
$fl_situacao) {
        // prepara os campos da pesquisa para enviar como filtros na lista
        $filtroEquals = array();
        if (!empty($cd_fornecedor)) {
            $filtroEquals['cd_fornecedor'] = $cd_fornecedor;
        }
        if (!empty($fl_situacao)) {
            $filtroEquals['fl_situacao'] = $fl_situacao;
        }

        $filtroLike = array();
        if (!empty($nm_fornecedor)) {
            $filtroLike['nm_fornecedor'] = $nm_fornecedor;
        }

        // busca as categorias do banco de dados
        $this->db->select('*');
    }
}

```

```

        $this->db->from('sipec_fornecedor');
        if (!empty($filtroEquals)) {
            $this->db->where($filtroEquals);
        }
        if (!empty($filtroLike)) {
            $this->db->like($filtroLike);
        }
        $this->db->orderby('nr_ordem asc');
        $query = $this->db->get();
        return $query->result();
    }

function buscaFornecedorPeloCodigo($cd_fornecedor) {
    $this->db->select('*');
    $this->db->from('sipec_fornecedor');
    $this->db->where(array('cd_fornecedor' => $cd_fornecedor));
    $query = $this->db->get();
    return $query->result();
}

function fornecedorReferencia() {
    $query = $this->db->query('
        SELECT pf.cd_fornecedor
        FROM (
            SELECT max( f.qtde ) AS qtde, cd_fornecedor
            FROM (
                SELECT cd_fornecedor, count( * ) AS qtde
                FROM `sipec_produto_fornecedor`
                GROUP BY cd_fornecedor
            ) f
            GROUP BY cd_fornecedor
            ORDER BY qtde DESC
            LIMIT 1
        ) pf
    ');
    $row = $query->row();
    return $row->cd_fornecedor;
}

function listaFornecedores($fl_situacao="", $cd_fornecedor="") {
    $this->db->from('sipec_fornecedor');
    if (!empty($fl_situacao)) {
        $this->db->where('fl_situacao', $fl_situacao);
    }
    if (!empty($cd_fornecedor)) {
        $this->db->or_where('cd_fornecedor', $cd_fornecedor);
    }
    $this->db->order_by('nm_fornecedor');
    $query = $this->db->get();
    $fornecedores = $query->result();
}

```

```

    $listaFornecedores = array();
    $listaFornecedores[""] = 'Selecione...';
    foreach($fornecedores as $fornecedor) {
        $listaFornecedores[$fornecedor->cd_fornecedor] = $fornecedor-
>nm_fornecedor;
    }
    return $listaFornecedores;
}

```

```

    function fornecedorPossuiVinculo($cd_fornecedor) {
        $this->db->from('sipec_produto');
        $this->db->where(array('cd_fornecedor' => $cd_fornecedor));
        $query = $this->db->get();
        if ($query->num_rows() > 0) {
            return true;
        } else {
            return false;
        }
    }
}
?>

```

### **\\system\application\models\importacao\_model.php**

```

<?php
class Importacao_Model extends Model {
    function Importacao_Model() {
        parent::Model();
        $this->load->model('categoria_model');
        $this->load->model('produto_model');
    }

    function gravarProdutosBD($produtos = array(), $fornecedor) {
        //echo "<pre>";print_r($produtos);exit;
        if (count($produtos) > 0) {
            $produtos = $produtos["produtos"]["produto"];
            //echo "<pre>";print_r($produtos);exit;
            foreach ($produtos as $produto) {

                // verifica através do código universal se o produto está
                cadastrado como ativo
                $cd_produto = $this->produto_model-
>buscaProdutoAtivoPeloCodigo($produto["codigo"]);

                $infoProduto = array();
                if (!empty($cd_produto)) { // se estiver, pega o cd_produto
                    e junto com o cd_fornecedor e demais informações armazena em produto_fornecedor
                    $infoProduto["cd_produto"] = $cd_produto;
                    $infoProduto["cd_fornecedor"] = $fornecedor;
                    $infoProduto["vl_preco"] = $produto["preco"];
                    $infoProduto["qt_estoque"] = $produto["estoque"];
                }
            }
        }
    }
}

```

```

        $this->produto_model-
>gravarProdutoFornecedor($infoProduto);
        } else { // se não estiver, armazena em produto com o
cd_situacao = 3 (pendente)

        // verifica através do código universal se o produto
está cadastrado como pendente
        $cd_produto = $this->produto_model-
>buscaProdutoPendentePeloCodigo($produto["codigo"]);

        if (empty($cd_produto)) { // se não estiver
cadastrado como pendente, armazena no banco de dados
            $infoProduto["cd_produto"] = null;
            $infoProduto["de_titulo"] = $produto["nome"];
            $infoProduto["nr_universal"] =
$produto["codigo"];

            $infoProduto["cd_categoria"] = null;
            $infoProduto["cd_situacao"] = 3;
            $this->produto_model-
>gravarProduto($infoProduto);
        }
    }
}

function armazenaDadosNoHistorico() {
    $data = date("Y-m-d");
    $importacao["dt_execucao"] = $data;
    $cd_importacao = $this->gravarHistoricoImportacao($importacao);

    $produtosFornecedor = $this->produto_model-
>buscaProdutoFornecedorPeloFiltro();

    foreach ($produtosFornecedor as $produtoFornecedor) {
        $infoProduto = array();
        $infoProduto["cd_importacao"] = $cd_importacao;
        $infoProduto["cd_produto"] = $produtoFornecedor->cd_produto;
        $infoProduto["cd_fornecedor"] = $produtoFornecedor-
>cd_fornecedor;

        $infoProduto["vl_preco"] = $produtoFornecedor->vl_preco;
        $infoProduto["qt_estoque"] = $produtoFornecedor->qt_estoque;
        $infoProduto["fl_mais_barato"] = $produtoFornecedor-
>fl_mais_barato;

        $this->gravarHistoricoImportacaoProduto($infoProduto);
    }
}

function cargaProdutosSistemaEcommerce() {

```

```

// setar o status dos produtos do sistema de e-commerce para "0"
$this->db->update('products', array('products_status'=>0));

// pegar os produtos da última importação com o menor preço
$cd_importacao = $this->ultimaImportacao();
$productos = $this-
>buscaHistoricoImportacaoPeloFiltro($cd_importacao, "", "", 1);

foreach ($productos as $produto) {
    $this->db->select('manufacturers_id');
    $this->db->from('manufacturers');
    $this->db->where(array('manufacturers_name' => $produto-
>nm_fornecedor));
    $query = $this->db->get();
    if ($query->num_rows() > 0) {
        $row = $query->row();
        $manufacturers_id = $row->manufacturers_id;
    } else {
        $manufacturers_id = null;
    }

    $infoProduto = array();
    $infoProduto["products_quantity"] = $produto->qt_estoque;
    $infoProduto["products_price"] = $produto->vl_preco;
    $infoProduto["products_last_modified"] = date("Y-m-d H:i:s");
    $infoProduto["products_status"] = 1;
    $infoProduto["manufacturers_id"] = $manufacturers_id;
    $this->db->where('products_ean_code', $produto->nr_universal);
    $this->db->update('products', $infoProduto);
}
}

function gravarHistoricoImportacaoProduto($produto) {
    $this->db->insert('sipec_historico_importacao_produto', $produto);
    return $this->db->insert_id();
}

function gravarHistoricoImportacao($historicoImportacao) {
    $this->db->insert('sipec_historico_importacao', $historicoImportacao);
    return $this->db->insert_id();
}

function excluirHistoricoImportacao($cd_importacao) {
    $this->db->delete('sipec_historico_importacao', array('cd_importacao'
=> $cd_importacao));
}

function buscaHistoricoImportacaoPeloFiltro($cd_importacao="",
$cd_fornecedor="", $cd_produto="", $fl_mais_barato="") {
    // prepara os campos da pesquisa para enviar como filtros na lista

```

```

    $filtroEquals = array();
    if (!empty($cd_importacao)) {
        $filtroEquals['i.cd_importacao'] = $cd_importacao;
    }
    if (!empty($cd_fornecedor)) {
        $filtroEquals['ip.cd_fornecedor'] = $cd_fornecedor;
    }
    if (!empty($cd_produto)) {
        $filtroEquals['ip.cd_produto'] = $cd_produto;
    }
    if (!empty($fl_mais_barato)) {
        $filtroEquals['ip.fl_mais_barato'] = $fl_mais_barato;
    }
}

$filtroLike = array();

// busca as categorias do banco de dados
$this->db->select('i.cd_importacao, i.dt_execucao');
$this->db->select('ip.vl_preco, ip.qt_estoque, ip.fl_mais_barato');
$this->db->select('p.de_titulo, p.nr_universal');
$this->db->select('f.nm_fornecedor');
$this->db->from('sipec_historico_importacao as i');
$this->db->join('sipec_historico_importacao_produto as ip',
'ip.cd_importacao = i.cd_importacao');
$this->db->join('sipec_fornecedor as f', 'f.cd_fornecedor =
ip.cd_fornecedor');
$this->db->join('sipec_produto as p', 'p.cd_produto = ip.cd_produto');
$this->db->orderby('i.dt_execucao desc, p.de_titulo asc, ip.vl_preco
asc');

if (!empty($filtroEquals)) {
    $this->db->where($filtroEquals);
}
if (!empty($filtroLike)) {
    $this->db->like($filtroLike);
}
$query = $this->db->get();
return $query->result();
}

function buscaHistoricoImportacaoPeloCodigo($cd_fornecedor) {
    $this->db->select('*');
    $this->db->from('sipec_fornecedor');
    $this->db->where(array('cd_fornecedor' => $cd_fornecedor));
    $query = $this->db->get();
    return $query->result();
}

function listaHistoricoImportacoes($cd_importacao="") {
    $this->db->from('sipec_historico_importacao');
    if (!empty($cd_importacao)) {

```

```

        $this->db->or_where('cd_importacao', $cd_importacao);
    }
    $this->db->order_by('dt_execucao');
    $query = $this->db->get();
    $importacoes = $query->result();
    $listaHistoricoImportacoes = array();
    $listaHistoricoImportacoes[""] = 'Selecione...';
    foreach($importacoes as $importacao) {
        $listaHistoricoImportacoes[$importacao->cd_importacao] = $this-
>funcoes_genericas->mysql_date_para_humano($importacao->dt_execucao);
    }
    return $listaHistoricoImportacoes;
}

function ultimaImportacao() {
    $this->db->select('max(cd_importacao) as cd_importacao');
    $this->db->from('sipec_historico_importacao');
    $query = $this->db->get();
    if ($query->num_rows() > 0) {
        $row = $query->row();
        return $row->cd_importacao;
    } else {
        return null;
    }
}

function podeExecutarImportacao() {
    $this->db->select('*');
    $this->db->from('sipec_historico_importacao');
    $this->db->where(array('dt_execucao' => date("Y-m-d")));
    $query = $this->db->get();
    //print_r($this->db);
    if ($query->num_rows() > 0) {
        return false;
    } else {
        return true;
    }
}
}
?>

```

**\\system\application\models\login\_model.php**

<?

```
class login_model extends Model {
```

```

    function login_model() {
        parent::Model();
    }

```

```

    function getSenha($usuario) {

```

```

$retorno = "";
    if (!empty($usuario)) {
        $this->db->select('user_password');
        $this->db->from('administrators');
        $this->db->where('user_name', $usuario);
        $query = $this->db->get();
        $resultado = $query->result();
        if ($resultado) {
            $retorno = $resultado[0]->user_password;
        }
    }
return $retorno;
}

function insert_entry()
{
    $this->title = $_POST['title']; // please read the below note
    $this->content = $_POST['content'];
    $this->date = time();

    $this->db->insert('entries', $this);
}

function update_entry()
{
    $this->title = $_POST['title'];
    $this->content = $_POST['content'];
    $this->date = time();

    $this->db->update('entries', $this, array('id' => $_POST['id']));
}
}
?>

```

### **\\system\application\models\produto\_model.php**

```

<?php
class Produto_Model extends Model {
    function Produto_Model() {
        parent::Model();
    }

    function gravarProduto($produto) {
        $this->db->insert('sipec_produto', $produto);
        return $this->db->insert_id();
    }

    function atualizarFornecedor($cd_fornecedor, $fornecedor) {
        $this->db->where('cd_fornecedor', $cd_fornecedor);
        $this->db->update('sipec_fornecedor', $fornecedor);
    }
}

```

```

    }

    function excluirProduto($cd_produto) {
        $this->db->delete('sipec_produto', array('cd_produto' => $cd_produto));
    }

    function buscaProdutoNaCategoriaDoFornecedor($titulo, $categoria,
    $fornecedor) {
        $this->db->select('cd_produto');
        $this->db->from('sipec_produto');
        $this->db->where('cd_fornecedor', $fornecedor);
        $this->db->where('cd_categoria', $categoria);
        $this->db->where('de_titulo', $titulo);
        $query = $this->db->get();
        if ($query->num_rows() > 0) {
            $row = $query->row();
            return $row->cd_produto;
        } else {
            return null;
        }
    }

    function gravarProdutos($produtos, $categoria, $fornecedor) {
        if (array_key_exists("codigo", $produtos)) {
            $cdProduto = $this-
>buscaProdutoNaCategoriaDoFornecedor($produtos["titulo"], $categoria,
    $fornecedor);

            if (empty($cdProduto)) {
                $produto["cd_produto"] = null;
                $produto["nr_codigo"] = $produtos["codigo"];
                $produto["de_titulo"] = $produtos["titulo"];
                $produto["vl_preco"] = $produtos["preco"];
                $produto["cd_categoria"] = $categoria;
                $produto["cd_fornecedor"] = $fornecedor;
                $this->gravarProduto($produto);
            }
        } else {
            //for ($i=0; $i<count($categorias);$i++) {
            //$this->categoria_model-
>gravarCategorias($categorias[$i]);
            //echo "<pre>";print_r($categorias[$i]).<br>";
            //}
        }
    }

    function buscaProdutoAtivoPeloCodigo($codigo="") {
        return $this->buscaProduto($codigo, 1);
    }

    function buscaProdutoPendentePeloCodigo($codigo="") {

```

```

        return $this->buscaProduto($codigo, 3);
    }

    function buscaProduto($codigo="", $situacao="") {
        $this->db->select('cd_produto');
        $this->db->from('sipec_produto');
        if (!empty($codigo)) {
            $this->db->where('nr_universal', $codigo);
        }
        if (!empty($situacao)) {
            $this->db->where('cd_situacao', $situacao);
        }
    }

    $query = $this->db->get();
    if ($query->num_rows() > 0) {
        $row = $query->row();
        return $row->cd_produto;
    } else {
        return null;
    }
}

function buscaProdutoPeloFiltro($cd_produto=null, $de_titulo=null,
$nr_universal=null, $cd_categoria=null, $cd_situacao=null) {
    // prepara os campos da pesquisa para enviar como filtros na lista
    $filtroEquals = array();
    if (!empty($cd_produto)) {
        $filtroEquals['cd_produto'] = $cd_produto;
    }
    if (!empty($nr_universal)) {
        $filtroEquals['nr_universal'] = $nr_universal;
    }
    if (!empty($cd_categoria)) {
        $filtroEquals['cd_categoria'] = $cd_categoria;
    }
    if (!empty($cd_situacao)) {
        $filtroEquals['cd_situacao'] = $cd_situacao;
    }

    $filtroLike = array();
    if (!empty($de_titulo)) {
        $filtroLike['de_titulo'] = $de_titulo;
    }
}

$this->db->select('*');
$this->db->from('sipec_produto');
if (!empty($filtroEquals)) {
    $this->db->where($filtroEquals);
}
if (!empty($filtroLike)) {

```

```

        $this->db->like($filtroLike);
    }
    $this->db->orderby('cd_categoria asc, de_titulo asc');
    $query = $this->db->get();
    return $query->result();
}

function listaProdutos($cd_produto="", $cd_categoria="", $cd_situacao="") {
    $this->db->from('sipec_produto');
    if (!empty($cd_produto)) {
        $this->db->or_where('cd_produto', $cd_produto);
    }
    if (!empty($cd_categoria)) {
        $this->db->or_where('cd_categoria', $cd_categoria);
    }
    if (!empty($cd_situacao)) {
        $this->db->or_where('cd_situacao', $cd_situacao);
    }
    $this->db->order_by('de_titulo');
    $query = $this->db->get();
    $produtos = $query->result();
    $listaProdutos = array();
    $listaProdutos[""] = 'Selecione...';
    foreach($produtos as $produto) {
        $listaProdutos[$produto->cd_produto] = $produto->de_titulo;
    }
    return $listaProdutos;
}

#####
## TABELA: SIPEC_PRODUTO_FORNECEDOR ##
#####

function gravarProdutoFornecedor($produto) {
    $this->db->insert('sipec_produto_fornecedor', $produto);
    return $this->db->insert_id();
}

function excluirProdutoFornecedor($cd_produto) {
    $this->db->delete('sipec_produto_fornecedor', array('cd_produto' =>
$cd_produto));
}

function excluirProdutosFornecedor() {
    $this->db->empty_table('sipec_produto_fornecedor');
}

function buscaProdutoFornecedorPeloFiltro($cd_fornecedor=null,
$cd_produto=null, $fl_mais_barato=null) {
    // prepara os campos da pesquisa para enviar como filtros na lista

```

```

    $filtroEquals = array();
    if (!empty($cd_fornecedor)) {
        $filtroEquals['cd_fornecedor'] = $cd_fornecedor;
    }
    if (!empty($cd_produto)) {
        $filtroEquals['cd_produto'] = $cd_produto;
    }
    if (!empty($fl_mais_barato)) {
        $filtroEquals['fl_mais_barato'] = $fl_mais_barato;
    }

    $this->db->select('*');
    $this->db->from('sipec_produto_fornecedor');
    if (!empty($filtroEquals)) {
        $this->db->where($filtroEquals);
    }
    if (!empty($filtroLike)) {
        $this->db->like($filtroLike);
    }
    $this->db->orderby('cd_fornecedor asc, cd_produto asc');
    $query = $this->db->get();
    return $query->result();
}

function buscaProdutoFornecedorPeloCodigo($cd_fornecedor, $cd_produto) {
    $this->db->select('*');
    $this->db->from('sipec_produto_fornecedor');
    $this->db->where(array('cd_fornecedor' => $cd_fornecedor));
    $this->db->where(array('cd_produto' => $cd_produto));
    $query = $this->db->get();
    return $query->result();
}

function defineProdutosFornecedorMaisBarato() {
    $query = $this->db->query('
        SELECT cd_fornecedor, cd_produto
        FROM `sipec_produto_fornecedor` pf
        WHERE vl_preco = (
            SELECT min( vl_preco )
            FROM sipec_produto_fornecedor
            WHERE cd_produto = pf.cd_produto
        )
        GROUP BY cd_produto
    ');
    foreach ($query->result() as $row) {
        $this->defineProdutoFornecedorMaisBarato($row->cd_produto,
        $row->cd_fornecedor);
    }
}

```

```

function defineProdutoFornecedorMaisBarato($cd_produto, $cd_fornecedor) {
    $this->db->where('cd_produto', $cd_produto);
    $this->db->where('cd_fornecedor', $cd_fornecedor);
    $this->db->update('sipec_produto_fornecedor', array('fl_mais_barato' => 1));
}
}
?>

```

### **\\system\application\models\usuario\_model.php**

```
<?php
```

```

class Usuario_Model extends Model {

    function Usuario_Model() {
        parent::Model();
        $this->load->model('dx_auth/users');
    }

    function gravarUsuario($usuario) {
        $user = $this->users->create_user($usuario);
        return $user->id;
    }

    function atualizarUsuario($id, $usuario) {
        return $this->users->set_user($id, $usuario);
    }

    function trocarSenha($id, $senha) {
        $pass = crypt($this->dx_auth->_encode($senha));
        return $this->users->change_password($id, $pass);
    }

    function excluirUsuario($id) {
        return $this->users->delete_user($id);
    }

    function buscaUsuarioPeloFiltro($id, $username, $email) {
        // prepara os campos da pesquisa para enviar como filtros na lista
        $filtroEquals = array();
        if (!empty($id)) {
            $filtroEquals['id'] = $id;
        }

        $filtroLike = array();
        if (!empty($username)) {
            $filtroLike['username'] = $username;
        }
        if (!empty($email)) {
            $filtroLike['email'] = $email;
        }
    }
}

```

```

// busca as unidades do banco de dados
$this->db->select('*');
$this->db->from('users');
if (!empty($filtroEquals)) {
    $this->db->where($filtroEquals);
}
if (!empty($filtroLike)) {
    $this->db->like($filtroLike);
}
$query = $this->db->get();
return $query->result();
}

function buscaUsuarioPeloCodigo($id) {
    $this->db->select('*');
    $this->db->from('users');
    $this->db->where(array('id' => $id));
    $query = $this->db->get();
    return $query->result();
}

}

?>

\\system\application\views\fornecedor_cadastro.php
<html>
<head>
<?php $this->load->view('sipec_cabecalho')?>
</head>

<body>
<?php $this->load->view('sipec_topo')?>
<table width="100%" border="0" align="center" cellpadding="0" cellspacing="0">
<tr>
<td height="400"><table width="100%" height="100%" border="0" cellpadding="0" cellspacing="4">
<tr>
<td width="20%" valign="top"><?php $this->load->view('sipec_menu')?></td>
<td width="80%" valign="top">

        <!-- INICIO CONTEUDO -->
        <?php $this->load->view('sipec_erro_validacao')?>

        <?
        echo form_open("fornecedor/gravar");
        ?>
        <table width="65%" border="0" cellpadding="2" cellspacing="0">
        <tr>

```

```

                <td height="23" class="tituloTabela">&nbsp;Cadastro
                de fornecedores</td>
            </tr>
            <tr>
                <td height="150" valign="top" bgcolor="#FFFFFF"
class="borda_azul">
                    <br>
                    <fieldset>
                    <legend>&nbsp;&nbsp;&nbsp;Dados&nbsp;&nbsp;&nbsp;</legend>
                    <table width="100%" border="0" cellspacing="1"
cellpadding="1">
                        <tr>
                            <td width="35%" align="right">Nome:&nbsp;&nbsp;&nbsp;</td>
                            <td width="65%">
                                <? echo
form_input(array('name'=>'nm_fornecedor','value'=>${this->input-
}>post('nm_fornecedor'),'maxlength'=>'255','class'=>'tamanhoGrande')); ?>
                                
                                </td>
                            </tr>
                            <tr>
                                <td align="right">Web service:&nbsp;&nbsp;&nbsp;</td>
                                <td>
                                    <? echo
form_input(array('name'=>'de_url_ws','value'=>${this->input-
}>post('de_url_ws'),'maxlength'=>'255','class'=>'tamanhoGrande')); ?>
                                    
                                    </td>
                                </tr>
                            <tr>
                                <td align="right">Ordem:&nbsp;&nbsp;&nbsp;</td>
                                <td>
                                    <? echo
form_input(array('name'=>'nr_ordem','value'=>${this->input-
}>post('nr_ordem'),'maxlength'=>'3','class'=>'tamanhoPequeno')); ?>
                                    
                                    </td>
                                </tr>
                            <tr>
                                <td align="right">Referência:&nbsp;&nbsp;&nbsp;</td>
                                <td>
                                    <? echo form_radio('fl_referencia', 1, $fornecedor-
>radioReferenciaSim); ?> Sim
                                    <? echo form_radio('fl_referencia', 2, $fornecedor-
>radioReferenciaNao); ?> Não

```





```

        <td>
            <? echo
form_input(array('name'=>'nm_fornecedor','value'=>${fornecedor-
>nm_fornecedor,'maxlength'=>'255','class'=>'tamanhoGrande')); ?>
            
        </td>
    </tr>
    <tr>
        <td align="right">Web service:&nbsp;   </td>
        <td>
            <? echo
form_input(array('name'=>'de_url_ws','value'=>${fornecedor-
>de_url_ws,'maxlength'=>'255','class'=>'tamanhoGrande')); ?>
            
        </td>
    </tr>
    <tr>
        <td align="right">Ordem:&nbsp;   </td>
        <td>
            <? echo
form_input(array('name'=>'nr_ordem','value'=>${fornecedor-
>nr_ordem,'maxlength'=>'3','class'=>'tamanhoPequeno')); ?>
            
        </td>
    </tr>
    <tr>
        <td align="right">Referência:&nbsp;   </td>
        <td>
            <? echo form_radio('fl_referencia', 1, ${fornecedor-
>radioReferenciaSim}); ?> Sim
            <? echo form_radio('fl_referencia', 2, ${fornecedor-
>radioReferenciaNao}); ?> Não
            
        </td>
    </tr>
    <tr>
        <td align="right">Situação:&nbsp;   </td>
        <td>
            <? echo form_radio('fl_situacao', 1, ${fornecedor-
>radioSituacaoAtivo}); ?> Ativo
            <? echo form_radio('fl_situacao', 2, ${fornecedor-
>radioSituacaoInativo}); ?> Inativo
            
        </td>
    </tr>

```

```

        </table>
        </fieldset>
        
        <span class="informativo">Preenchimento
obrigatório</span>
        <table width="100%" border="0" cellspacing="1"
cellpadding="1">
            <tr>
                <td width="35%" height="40" align="center">&nbsp;
</td>
                <td width="65%" height="40">
                    <input type="submit" value="Gravar"
class="botao">
                    &nbsp;&nbsp;&nbsp;<input type="button"
value="Cancelar" class="botao" onClick="javascript:location.href='<? echo
site_url("fornecedor"); ?>'>
                </td>
            </tr>
        </table>
    </td>
</tr>
</table>
<? echo form_close(); ?>
<br>
<!-- FIM CONTEUDO -->
</td>
</tr>
</table></td>
</tr>
<tr>
    <td class="fundo_rodape" height="26">&nbsp;&nbsp;&nbsp;</td>
</tr>
</table>
</body>
</html>

```

### **\\system\application\views\fornecedor\_lista.php**

```

<html>
<head>
<?php $this->load->view('sipec_cabecalho')?>
</head>

<body>
<?php $this->load->view('sipec_topo')?>
<table width="100%" border="0" align="center" cellpadding="0" cellspacing="0">
    <tr>

```

```

<td height="400"><table width="100%" height="100%" border="0" cellpadding="0"
cellspacing="4">
  <tr>
    <td width="20%" valign="top"><?php $this->load->view('sipec_menu')?></td>
    <td width="80%" valign="top">

      <!-- INICIO CONTEUDO -->
      <?php $this->load->view('sipec_erro_validacao') ?>
        <? echo form_open("fornecedor"); ?>
        <? echo form_hidden("processou", true); ?>
        <table width="90%" border="0" cellpadding="0" cellspacing="0">
          <tr>
            <td height="23" class="tituloTabela">&nbsp;Pesquisa
                                de fornecedores</td>
          </tr>
          <tr>
            <td height="150" valign="top" bgcolor="#FFFFFF"
class="borda_azul"><table width="100%" border="0" cellspacing="1"
cellpadding="1">
              <tr>
                <td>&nbsp;</td>
                <td>&nbsp;</td>
              </tr>
              <tr>
                <td
align="right"><strong>C&ocute;digo:&nbsp;</strong></td>
                <td><? echo
form_input(array('name' => 'cd_fornecedor', 'value' => $this->input-
>post('cd_fornecedor'), 'class' => 'tamanhoPequeno')); ?></td>
              </tr>
              <tr>
                <td
align="right"><strong>Nome:&nbsp;</strong></td>
                <td><? echo
form_input(array('name' => 'nm_fornecedor', 'value' => $this->input-
>post('nm_fornecedor'), 'class' => 'tamanhoGrande')); ?></td>
              </tr>
              <tr>
                <td
align="right"><strong>Situa&ccedil;o:&nbsp;</strong></td>
                <td>
                  <? echo
form_radio('fl_situacao', 1, $radioSituacaoAtivo); ?> Ativo
                  <? echo
form_radio('fl_situacao', 2, $radioSituacaoInativo); ?> Inativo
                  <? echo
form_radio('fl_situacao', 0, $radioSituacaoIndiferente); ?> Indiferente
                </td>
              </tr>
            </td>
          </tr>
        </table>
      </td>
    </tr>
  </table>

```

```

                                </tr>
                                <tr>
align="center">&nbsp; </td>
                                <td height="50"
                                <td height="50">
                                    <input type="submit"
value="Pesquisar" class="botao">
                                    &nbsp;&nbsp;&nbsp;<input
type="button" class="botao" value="Limpar" onClick="javascript:location.href='<?
echo site_url('fornecedor'); ?>'>
                                    </td>
                                </tr>
                                </table></td>
                                </tr>
                                <tr>
                                    <td align="right" height="23">&nbsp;<a href="<? echo site_url('fornecedor/cadastrar');
?>"><strong>Cadastrar
                                    fornecedor</strong></a></td>
                                </tr>
                                </table>
                                <? echo form_close(); ?>
                                <? //if ($this->input->post('processou')) { ?>
                                <table width="90%" border="0" cellpadding="0"
cellspacing="0">
                                    <tr>
  <td height="23"
class="tituloTabela">&nbsp;Resultado
  da pesquisa</td>
                                    </tr>
                                    <tr>
  <td height="150" valign="top"
bgcolor="#FFFFFF" class="borda_azul">
  <? echo form_open("fornecedor/excluir",
array('id' => 'form_excluir')); ?>
  <input type="hidden"
name="cd_fornecedor_exclusao" id="cd_fornecedor_exclusao" value="">
  <? if (count($listaFornecedor) > 0) { ?>
  <table width="100%" border="0"
cellspacing="2" cellpadding="1">
  <tr align="center">
  <td
class="borda"><strong>C&oacute;digo</strong></td>
  <td
class="borda"><strong>Nome</strong></td>
  <td
class="borda"><strong>Ordem</strong></td>

```

```

class="borda"><strong>Referência</strong></td>
class="borda"><strong>A&ccedil;&otilde;es</strong></td>
<? foreach ($listaFornecedor as $fornecedor) { ?>
align="center"><? echo $fornecedor->codigo; ?></td>
href="<? echo site_url('fornecedor/visualizar/' . $fornecedor->codigo); ?>"><? echo
$fornecedor->nome; ?></a></td>
align="center"><? echo $fornecedor->ordem; ?></td>
align="center"><? echo $fornecedor->referencia; ?></td>
align="center"><span id="situacao_<?php echo $fornecedor->codigo; ?>"><?php
echo $fornecedor->situacao; ?></span><a href="<? echo site_url('fornecedor/editar/'
. $fornecedor->codigo); ?>"></a><a
href="javascript:excluirForm('form_excluir', 'cd_fornecedor_exclusao', <?php echo
$fornecedor->codigo; ?>);"></a></td>
<? } ?>
</table>
<div align="right"><strong> <? echo
count($listaFornecedor); ?> registro(s) encontrado(s)&nbsp;<br><br></strong></div>
<? } else {
?>
<?php $this->load-
>view('sipec_sem_registro'); ?>
<? } ?>
<? echo form_close(); ?>
</td>
</tr>
</table>
<br>
<? //} ?>
<!-- FIM CONTEUDO -->
</td>
</tr>
</table></td>
</tr>
<tr>
<td class="fundo_rodape" height="26">&nbsp;</td>
</tr>
</table>

```

```
</body>
</html>
```

### **\\system\application\views\fornecedor\_visualizar.php**

```
<html>
<head>
<?php $this->load->view('sipec_cabecalho')?>
</head>

<body>
<?php $this->load->view('sipec_topo')?>
<table width="100%" border="0" align="center" cellpadding="0" cellspacing="0">
  <tr>
    <td height="400"><table width="100%" height="100%" border="0" cellpadding="0" cellspacing="4">
      <tr>
        <td width="20%" valign="top"><?php $this->load->view('sipec_menu')?></td>
        <td width="80%" valign="top">

          <!-- INICIO CONTEUDO -->
          <table width="65%" border="0" cellpadding="2" cellspacing="0">
            <tr>
              <td height="23" class="tituloTabela">&nbsp;Consulta
                de fornecedor</td>
            </tr>
            <tr>
              <td height="150" valign="top" bgcolor="#FFFFFF"
class="borda_azul">
                <br>
                <fieldset>
                <legend>&nbsp;&nbsp;&nbsp;Dados&nbsp;&nbsp;&nbsp;</legend>
                <table width="100%" border="0" cellspacing="1"
cellpadding="1">
                  <tr>
                    <td width="35%"
align="right">C&oacute;digo:&nbsp;&nbsp;&nbsp;</td>
                    <td width="65%"><? echo $fornecedor->codigo;
?></td>
                  </tr>
                  <tr>
                    <td align="right">Nome:&nbsp;&nbsp;&nbsp;</td>
                    <td><? echo $fornecedor->nome; ?></td>
                  </tr>
                  <tr>
                    <td align="right">Web service:&nbsp;&nbsp;&nbsp;</td>
                    <td><? echo $fornecedor->webservice; ?></td>
                  </tr>
                  <tr>
                    <td align="right">&nbsp;&nbsp;&nbsp;</td>
                    <td>&nbsp;&nbsp;&nbsp;</td>
                  </tr>
                </table>
              </td>
            </tr>
          </table>
        </td>
      </tr>
    </td>
  </tr>
</table>
```



```

    apMessageBox.errorImage = "/img/error-32x32.png";
    if (confirm("Deseja executar a importação dos produtos agora?")) {
        executar();
    }
});

function alerta(msg) {
    apMessageBox.error({
        title: "Falha na IMPORTAÇÃO!",
        message: msg
    });
}

function executar() {
    passo1();
}

function passo1() {
    adicionarLinha(1, "Iniciando o processo de importação");
    $.ajax({
        url: "/index.php/ajax/executarImportacaoPasso1",
        success: function(retorno) {
            if (retorno == 'OK') {
                $("#passo1").html('');
                passo2();
            } else {
                $("#passo1").html('');
                alerta(retorno);
            }
        }
    });
}

function passo2() {
    adicionarLinha(2, "Obtendo a lista de produtos e cadastrando no banco de
dados");
    $.ajax({
        url: "/index.php/ajax/executarImportacaoPasso2",
        success: function(retorno) {
            $("#passo2").html(retorno);
            passo3();
        }
    });
}

function passo3() {

```

```

        adicionarLinha(3, "Identificando os produtos similares e fazendo a
        comparação de preços");
        $.ajax({
            url: "/index.php/ajax/executarImportacaoPasso3",
            success: function(retorno) {
                $("#passo3").html(retorno);
                passo4();
            }
        });
    }

function passo4() {
    adicionarLinha(4, "Armazenando os dados no histórico do sistema");
    $.ajax({
        url: "/index.php/ajax/executarImportacaoPasso4",
        success: function(retorno) {
            $("#passo4").html(retorno);
            passo5();
        }
    });
}

function passo5() {
    adicionarLinha(5, "Executando a carga dos produtos no sistema de e-
    commerce");
    $.ajax({
        url: "/index.php/ajax/executarImportacaoPasso5",
        success: function(retorno) {
            $("#passo5").html(retorno);
            passo6();
        }
    });
}

function passo6() {
    adicionarLinha(6, "Finalizando o processo de importação");
    $.ajax({
        url: "/index.php/ajax/executarImportacaoPasso6",
        success: function(retorno) {
            $("#passo6").html(retorno);
        }
    });
}

function adicionarLinha(passo, descricao) {
    $("##tabelaExecucao").last().append("<tr><td class='\"borda\"' align='\"center\"'>"
+ passo + "</td><td class='\"borda\"'>" + descricao + "</td><td class='\"borda\"'
align='\"center\"' id='\"passo\"' + passo + "\"><img src='\"/img/ajax-loader.gif\"'
width='\"16\"' height='\"16\"'></td></tr>");
}

```







```

        <? foreach ($listaImportacao as $importacao) { ?>
        <tr>
            <td class="borda" align="center"><? echo $importacao->data; ?></td>
            <td class="borda"><? echo $importacao->produto; ?></td>
            <td class="borda" align="center"><? echo $importacao->preco; ?></td>
            <td class="borda" align="center"><? echo $importacao->estoque;
?></td>
            <td class="borda" align="center"><? echo $importacao->fornecedor;
?></td>
            <td class="borda" align="center"><? echo $importacao->mais_barato;
?></span></td>
        </tr>
        <? } ?>
    </table>
        <div align="right"><strong> <? echo
count($listaImportacao); ?> registro(s) encontrado(s)&nbsp;&nbsp;&nbsp;<br><br></strong></div>
        <? } else {
?>
        <?php $this->load-
>view('sipec_sem_registro'); ?>
        <? } ?>
    </td>
    </tr>
    </table>
    <br>
<? //} ?>
    <!-- FIM CONTEUDO -->
    </td>
    </tr>
    </table></td>
</tr>
<tr>
    <td class="fundo_rodape" height="26">&nbsp;&nbsp;&nbsp;</td>
</tr>
</table>
</body>
</html>

```

### **\\system\application\views\sipec\_cabecalho.php**

```

<title>SIPEC - Sistema Online de Pesquisa e Compra do Melhor Pre&ccedil;o</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<link href="/css/estilo.css" type="text/css" rel="Stylesheet" />
<link href="/css/redmond/jquery-ui-1.8.custom.css" type="text/css" rel="Stylesheet"
/>
<script language="JavaScript" type="text/javascript" src="/js/jquery.js"></script>
<script language="JavaScript" type="text/javascript" src="/js/jquery-ui.js"></script>
<script language="JavaScript" type="text/javascript" src="/js/jquery.ui.datepicker-pt-
BR.js"></script>

```

```

<script language="JavaScript" type="text/javascript"
src="/js/apMessageBox.js"></script>
<script language="JavaScript" type="text/javascript" src="/js/funcoes.js"></script>
<script language="JavaScript" type="text/javascript" src="/js/menu.js"></script>

```

### **\\system\application\views\sipec\_erro\_validacao.php**

```

<?php
    $mensagens_erro = validation_errors();
    if (!empty($validacao_upload)) {
        $mensagens_erro .= $validacao_upload;
    }
    if (!empty($mensagens_erro)) {
?>
<table width="500" height="100" border="0" cellpadding="5" cellspacing="0"
class="fundo_bege">
    <tr>
        <td height="23" class="fundo_amarelo">
            <strong> Aten&ccedil;&atilde;o</strong>
        </td>
    </tr>
    <tr>
        <td height="77" valign="top" class="borda_amarela">
            <?php echo $mensagens_erro; ?>
        </td>
    </tr>
</table>
<br>
<?php } ?>

```

### **\\system\application\views\sipec\_menu.php**

```

<ul id="menu">
    <li>
        <a href="#">Usuários</a>
        <ul>
            <li><a href="<? echo site_url('usuario/cadastrar');
?>">Cadastrar</a></li>
            <li><a href="<? echo site_url('usuario'); ?>">Listar</a></li>
        </ul>
    </li>
    <li>
        <a href="#">Fornecedores</a>
        <ul>
            <li><a href="<? echo site_url('fornecedor/cadastrar');
?>">Cadastrar</a></li>
            <li><a href="<? echo site_url('fornecedor'); ?>">Listar</a></li>
        </ul>
    </li>
    <li>
        <a href="#">Importação</a>

```

```

        <ul>
            <li><a href="<? echo site_url('importacao/executar');
?>">Executar</a></li>
            <li><a href="<? echo site_url('importacao/historico');
?>">Histórico</a></li>
        </ul>
    </li>
    <li>
        <a href="#">Ajuda</a>
        <ul>
            <li><a href="<? echo site_url('ajuda/manual');
?>">Manual</a></li>
            <li><a href="<? echo site_url('ajuda/sobre'); ?>">Sobre</a></li>
        </ul>
    </li>
</ul>

```

### **\\system\application\views\sipec\_pagina\_sucesso.php**

```

<html>
<head>
<?php $this->load->view('sipec_cabecalho')?>
</head>

<body onload="redireciona('<?php echo $url; ?>')">
<?php $this->load->view('sipec_topo')?>
<table width="100%" border="0" align="center" cellpadding="0" cellspacing="0">
    <tr>
        <td height="400"><table width="100%" height="100%" border="0" cellpadding="0"
cellspacing="4">
            <tr>
                <td width="20%" valign="top"><?php $this->load->view('sipec_menu')?></td>
                <td width="80%" valign="top">

                    <!-- INICIO CONTEUDO -->
                    <br>
                    <br>
                    <br>
                    <br>
                    <br>
                    <br>
                    <br>
                    <br>

                    <table width="45%" border="0" align="center" cellpadding="2"
cellspacing="0">
                        <tr>
                            <td height="125" valign="middle" align="center"
class="borda_azul fundo_bege">

```

```

                &nbsp;<strong>Opera&ccedil;&atilde;o
realizada com sucesso!</strong>
            </td>
        </tr>
    </table>
    <br>
    <!-- FIM CONTEUDO -->

</td>
</tr>
</table></td>
</tr>
<tr>
    <td class="fundo_rodape" height="26">&nbsp;</td>
</tr>
</table>
</body>
</html>

```

#### **\\system\application\views\sipec\_sem\_registro.php**

```

<table width="100%" height="90%" border="0" cellspacing="2" cellpadding="1">
    <tr align="center">
        <td style="color:red"><strong>Nenhum registro foi encontrado para o(s)
critério(s) de pesquisa utilizado(s).</strong></td>
    </tr>
</table>

```

#### **\\system\application\views\sipec\_topo.php**

```

<table width="100%" border="0" align="center" cellpadding="0" cellspacing="0">
    <tr>
        <td height="120">
            <table width="100%" height="120" border="0" align="center" cellpadding="0"
cellspacing="0">
                <tr>
                    <td width="134"><em></em></td>
                    <td class="fundo_topo">&nbsp;</td>
                    <td width="564"><em></em></td>
                </tr>
            </table></td>
    </tr>
    <tr>
        <td class="fundo_azul" height="1"></td>
    </tr>

```



```

        <li>Gerenciamento de Forencedores</li>
        <li>Processo de Importa&ccedil;&atilde;o</li>
        <li>Ajuda do Sistema</li>
    </ul>
    </td>
</tr>
</table></td>
</tr>
<tr>
    <td class="fundo_rodape" height="26">&nbsp;</td>
</tr>
</table>
</body>
</html>

```

### **\\system\application\views\tela\_login.php**

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<?php $this->load->view('sipec_cabecalho')?>
<script type="text/javascript">
$(document).ready(function(){
    $('#username').focus();
});
</script>
</head>

<body>
<br>
<br>
<br>
<br>
<?php $this->load->view('sipec_erro_validacao')?>
<br>
<br>
<?php echo form_open($this->uri->uri_string())?>
<table width="500" border="0" align="center" cellpadding="0" cellspacing="0">
    <tr>
        <td height="200"></td>
    </tr>
    <tr>
        <td class="fundo_azul" height="1"></td>
    </tr>
    <tr>
        <td height="1"></td>
    </tr>

```

```

</tr>
<tr>
  <td class="fundo_bege" height="110"><table width="100%" border="0"
cellspacing="2" cellpadding="0">
  <tr>
    <td width="45%">&nbsp;</td>
    <td width="55%">&nbsp;</td>
  </tr>
  <tr>
    <td align="right"><strong>Usu&aacute;rio:&nbsp;</strong></td>
    <td><? echo
form_input(array('name'=>'username','id'=>'username','maxlength'=>'20','class'=>'ta
manhoMedio')); ?></td>
  </tr>
  <tr>
    <td align="right"><strong>Senha:&nbsp;</strong></td>
    <td><? echo
form_password(array('name'=>'password','id'=>'password','maxlength'=>'20','class'=
>'tamanhoMedio')); ?></td>
  </tr>
  <tr>
    <td>&nbsp;</td>
    <td><input name="Enviar" type="submit" id="Enviar" value="Enviar"
class="botao"></td>
  </tr>
  <tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
</table></td>
</tr>
<tr>
  <td class="fundo_azul" height="1"></td>
</tr>
</table>
<? echo form_close(); ?>
</body>
</html>

```

### **\\system\application\views\usuario\_cadastro.php**

```

<html>
<head>
<?php $this->load->view('sipec_cabecalho')?>
</head>

<body>
<?php $this->load->view('sipec_topo')?>
<table width="100%" border="0" align="center" cellpadding="0" cellspacing="0">
  <tr>

```

```

<td height="400"><table width="100%" height="100%" border="0" cellpadding="0"
cellspacing="4">
  <tr>
    <td width="20%" valign="top"><?php $this->load->view('sipec_menu')?></td>
    <td width="80%" valign="top">

      <!-- INICIO CONTEUDO -->
      <?php $this->load->view('sipec_erro_validacao')?>

      <?
      echo form_open("usuario/gravar");
      ?>
      <table width="65%" border="0" cellpadding="2" cellspacing="0">
        <tr>
          <td height="23" class="tituloTabela">&nbsp;Cadastro
          de usuário</td>
        </tr>
        <tr>
          <td height="150" valign="top" bgcolor="#FFFFFF"
class="borda_azul">
            <br>
            <fieldset>
            <legend>&nbsp;&nbsp;&nbsp;Dados&nbsp;&nbsp;&nbsp;</legend>
            <table width="100%" border="0" cellspacing="1"
cellpadding="1">
              <tr>
                <td width="35%" align="right">Nome do
usuário:&nbsp;  </td>
                <td width="65%">
                  <? echo
form_input(array('name'=>'username','value'=>$this->input-
>post('username'),'maxlength'=>'20','class'=>'tamanhoGrande')); ?>
                  
                  </td>
                </tr>
                <tr>
                <td width="35%" align="right">E-mail:&nbsp;  </td>
                <td width="65%">
                  <? echo
form_input(array('name'=>'email','value'=>$this->input-
>post('email'),'maxlength'=>'100','class'=>'tamanhoGrande')); ?>
                  
                  </td>
                </tr>
                <tr>
                <td width="35%" align="right">Senha:&nbsp;  </td>

```



```

</tr>
<tr>
  <td class="fundo_rodape" height="26">&nbsp;</td>
</tr>
</table>
</body>
</html>

```

### \\system\application\views\usuario\_edicao.php

```

<html>
<head>
<?php $this->load->view('sipec_cabecalho')?>
</head>

<body>
<?php $this->load->view('sipec_topo')?>
<table width="100%" border="0" align="center" cellpadding="0" cellspacing="0">
  <tr>
    <td height="400"><table width="100%" height="100%" border="0" cellpadding="0"
cellspacing="4">
      <tr>
        <td width="20%" valign="top"><?php $this->load->view('sipec_menu')?></td>
        <td width="80%" valign="top">

          <!-- INICIO CONTEUDO -->
          <?php $this->load->view('sipec_erro_validacao')?>

          <?
          echo form_open("usuario/atualizar");
          echo form_hidden('id', $usuario->codigo);
          echo form_hidden('email_bd', $usuario->email_bd);
          ?>
          <table width="65%" border="0" cellpadding="2" cellspacing="0">
            <tr>
              <td height="23" class="tituloTabela">&nbsp;Alteração
                de usuário</td>
            </tr>
            <tr>
              <td height="150" valign="top" bgcolor="#FFFFFF"
class="borda_azul">
                <br>
                <fieldset>
                <legend>&nbsp;&nbsp;&nbsp;Dados&nbsp;&nbsp;&nbsp;</legend>
                <table width="100%" border="0" cellspacing="1"
cellpadding="1">
                  <tr>
                    <td width="35%"
align="right">C&ocirc;digido:&nbsp;&nbsp;&nbsp;</td>

```



```

                <? echo
form_password(array('name'=>'password2','id'=>'password2','maxlength'=>'20','class'
=>"$usuario->cssSenha",$usuario->desabilitaSenha)); ?>
                
                </td>
            </tr>
        </table>
    </fieldset>
    
        <span class="informativo">Preenchimento
obrigatório</span>
        <table width="100%" border="0" cellspacing="1"
cellpadding="1">
            <tr>
                <td width="35%" height="40" align="center">&nbsp;
</td>
                <td width="65%" height="40">
                    <input type="submit" value="Gravar"
class="botao">
                    &nbsp;&nbsp;  <input type="button"
value="Cancelar" class="botao" onClick="javascript:location.href='<? echo
site_url("usuario"); ?>'>
                </td>
            </tr>
        </table>
    </td>
</tr>
</table>
</tr>
</table>
<? echo form_close(); ?>
<br>
<!-- FIM CONTEUDO -->

</td>
</tr>
</table></td>
</tr>
<tr>
    <td class="fundo_rodape" height="26">&nbsp;&nbsp;&nbsp;</td>
</tr>
</table>
</body>
</html>

```

**\\system\application\views\usuario\_lista.php**

```

<html>
<head>
<?php $this->load->view('sipec_cabecalho')?>

```





```

class="borda"><strong>Nome de usuário</strong></td>
class="borda"><strong>E-mail</strong></td>
class="borda"><strong>Data de cadastro</strong></td>
class="borda"><strong>Data de alteração</strong></td>
class="borda"><strong>A&ccedil;&otilde;es</strong></td>
<? foreach ($listaUsuario as $usuario) { ?>
align="center"><? echo $usuario->codigo; ?></td>
href="<? echo site_url('usuario/visualizar/' . $usuario->codigo); ?>"><? echo
$usuario->nome; ?></a></td>
align="center"><? echo $usuario->email; ?></td>
align="center"><? echo $usuario->data_criacao; ?></td>
align="center"><? echo $usuario->data_atualizacao; ?></td>
align="center"><a href="<? echo site_url('usuario/editar/' . $usuario->codigo);
?>"></a><a
href="javascript:excluirForm('form_excluir', 'id_exclusao', <?php echo $usuario-
>codigo; ?>);"></a></td>
<? } ?>
</table>
<div align="right"><strong> <? echo
count($listaUsuario); ?> registro(s) encontrado(s)&nbsp;<br><br></strong></div>
<? } else {
?>
<?php $this->load-
>view('sipec_sem_registro'); ?>
<? } ?>
<? echo form_close(); ?>
</td>
</tr>
</table>
<br>
<? //} ?>
<!-- FIM CONTEUDO -->
</td>

```



```

        <td><? echo $usuario->nome; ?></td>
    </tr>
    <tr>
        <td align="right">E-mail:&nbsp; </td>
        <td><? echo $usuario->email; ?></td>
    </tr>
    <tr>
        <td align="right">Data de cadastro:&nbsp; </td>
        <td><? echo $usuario->data_cadastro; ?></td>
    </tr>
    <tr>
        <td align="right">Data de
altera&ccedil;&atilde;o:&nbsp; </td>
        <td><? echo $usuario->data_alteracao; ?></td>
    </tr>
</table>
</fieldset>
<!--
<br>
<fieldset>
<legend>&nbsp;Permissões&nbsp;</legend>
<table width="100%" border="0" cellspacing="1"
cellpadding="1">
    <tr>
        <td width="35%"
align="right">C&oacute;digo:&nbsp; </td>
        <td width="65%"><? echo $usuario->codigo;
?></td>
    </tr>
</table>
</fieldset>
-->
<br>
<fieldset>
<legend>&nbsp;Acessos&nbsp;</legend>
<table width="100%" border="0" cellspacing="1"
cellpadding="1">
    <tr>
        <td width="35%" align="right">Último
acesso:&nbsp; </td>
        <td width="65%"><? echo $usuario-
>ultimo_acesso; ?></td>
    </tr>
    <tr>
        <td width="35%" align="right">Último IP:&nbsp; </td>
        <td width="65%"><? echo $usuario->ultimo_ip;
?></td>
    </tr>
</table>

```

```

        </fieldset>
        <table width="100%" border="0" cellspacing="1"
cellpadding="1">
        <tr>
        <td width="35%" height="40" align="center">&nbsp;
        </td>
        <td width="65%" height="40">
        <input type="button" value="Editar" class="botao"
onClick="javascript:location.href='<? echo site_url("usuario/editar/" . $usuario-
>codigo); ?>'>
        &nbsp;&nbsp;&nbsp;<input type="button"
value="Cancelar" class="botao" onClick="javascript:location.href='<? echo
site_url("usuario"); ?>'>
        </td>
        </tr>
        </table> </td>
        </tr>
        </table>
        <br>
        <!-- FIM CONTEUDO -->
        </td>
        </tr>
        </table></td>
        </tr>
        <tr>
        <td class="fundo_rodape" height="26">&nbsp;</td>
        </tr>
        </table>
        </body>
        </html>

```

## APÊNDICE D – ARTIGO

# Sistema On-line de Pesquisa e Compra do Menor Preço

Leonardo de Oliveira

Departamento de Informática e Estatística (INE)  
Universidade Federal de Santa Catarina (UFSC) – Florianópolis, SC – Brasil

leonardofln@gmail.com

**Abstract.** *This paper is a proposal to develop a system to search and purchase the best price, focused on retail sales and using as case study the supermarket segment. Our proposal uses Web Services and XML to integrate the various data sources involved. The contributions include the architecture of developed system and the specification of the system's behavior. The proposed system automatically matches products, from different data sources, identifies the best price and presents the information to the end user (consumer) through a virtual store.*

**Resumo.** Este artigo é uma proposta de desenvolvimento de um sistema de pesquisa e compra dos produtos que possuam o menor preço, focado no comércio varejista e utilizando como estudo de caso o segmento supermercadista. Para isso são utilizadas algumas tecnologias, como por exemplo, *Web Service* e *XML* para fazer a integração das diversas fontes de dados envolvidas. Também são definidos uma arquitetura e os comportamentos do sistema, para que os produtos cadastrados em diferentes fontes de dados sejam mapeados de forma automática, seja identificado o menor preço e estes sejam apresentados ao usuário final (consumidor) através de uma loja virtual.

## 1 Introdução

Os consumidores do setor varejista, quando pretendem fazer determinado tipo de compra, encontram nas lojas e no comércio em geral, uma diferença de preço para o produto que desejam adquirir.

O que geralmente acontece é que o consumidor se dirige até as lojas e faz uma pesquisa de preço para o produto desejado, com o objetivo de adquirir o mais barato e deste modo realizar uma economia, ou seja, poupar dinheiro.

Com as tecnologias que temos disponíveis atualmente, como a *Internet* e o *e-commerce*, o consumidor tem a facilidade de fazer essa comparação de preços em tempo real e até mesmo realizar a compra sem sair de casa.

No entanto, um dos problemas enfrentados, é que nem todos os comerciantes ou lojistas possuem um site na *Internet* e muito menos uma infra-estrutura de *e-commerce*, e deste modo, o consumidor não pode fazer uma pesquisa de preços completa, de forma rápida e sem muitos transtornos.

O objetivo deste trabalho é desenvolver um sistema *on-line* de pesquisa e de compra do menor preço, que possa ser utilizado em determinado segmento do comércio varejista. Este sistema deve permitir que o consumidor faça a pesquisa de preços *on-line* e adquira o produto desejado pelo preço mais baixo, independentemente dos lojistas ou comerciantes deste segmento possuírem site ou infra-estrutura de *e-commerce*. Para

isso, estes fornecedores parceiros terão que disponibilizar na *Web*, um serviço que possibilite o acesso a lista de produtos que serão comercializados.

## **2 Desenvolvimento**

### **2.1 Análise e concepção**

Para iniciarmos o desenvolvimento do sistema proposto, levantamos algumas informações a respeito da aplicação. Segundo Wazlawick (2004), “a fase de concepção consiste em uma etapa na qual o analista vai buscar as primeiras informações sobre o sistema a ser desenvolvido”. Wazlawick (2004) ainda cita que “os artefatos dessa fase são ainda desestruturados, isto é, não são necessariamente completos e organizados. O objetivo é descobrir se vale a pena fazer a análise, mas sem fazer a análise propriamente dita”.

Teremos então na fase de análise e concepção, as seguintes etapas: visão geral do sistema e o levantamento de requisitos.

#### **2.1.1 Visão geral do sistema**

A visão geral do sistema, nada mais é do que uma descrição daquilo que se pretende ao desenvolver determinado sistema. Para Wazlawick (2004), “a visão geral do sistema, ou sumário executivo, que é um texto corrido, sem necessidade de nenhuma estrutura especial. Ele deve descrever as principais idéias do cliente sobre o sistema”.

A proposta deste trabalho é desenvolver um sistema que permita aos consumidores realizar a compra dos produtos mais baratos nos supermercados de sua região, sem precisar se deslocar até os estabelecimentos para fazer a pesquisa de preços manualmente.

Para que isso seja possível, será necessário automatizar toda esta parte de pesquisa e comparação de preços dos supermercados e deixar disponível para o consumidor, apenas os produtos com o preço menor.

O sistema deve então se conectar com alguns supermercados pré-definidos, para obter a lista de produtos disponíveis para serem comercializados. De posse destas listas de produtos, é preciso comparar o preço de cada produto equivalente e determinar quais são aqueles que possuem o menor preço em cada estabelecimento. Após este levantamento, esses produtos serão colocados a venda em um sistema de comércio eletrônico (supermercado virtual), para ficar a disposição dos consumidores.

A partir deste momento, os consumidores acessam o supermercado virtual e fazem suas compras. Cada pedido feito pelos clientes pode conter produtos de vários supermercados. Então, o sistema processa os pedidos individualmente, e se comunica com os supermercados para informar quais produtos devem ser comprados.

Finalmente, no momento da entrega, um serviço terceirizado de transporte passa em cada um dos supermercados, coleta os produtos de cada cliente e realiza a entrega dos mesmos no endereço informado durante a compra.

#### **2.1.2 Levantamento de requisitos**

Para a construção do sistema, foi necessário fazer um levantamento de requisitos, ou seja, mapear e documentar as necessidades para a concretização do sistema. Segundo Wazlawick (2004), “a análise de requisitos está associada ao processo de descobrir

quais são as operações que o sistema deve realizar e quais são as restrições que existem sobre elas”.

Os requisitos aqui levantados tiveram sua origem de idéias minhas e conversas com outros colegas da faculdade, ou seja, diferentemente do processo convencional, os requisitos do sistema aqui desenvolvido não foram levantados com um cliente em específico.

Para uma melhor compreensão, vamos dividir os requisitos levantados em duas grandes categorias: os requisitos funcionais e os não funcionais.

Os requisitos funcionais correspondem à listagem de tudo que o sistema deve fazer. Já os requisitos não funcionais são restrições colocadas sobre como o sistema deve realizar seu requisitos funcionais (WAZLAWICK 2004).

### **2.1.2.1 Requisitos funcionais**

- RF001 – Os consumidores poderão fazer compras através de um supermercado virtual
- RF002 – O sistema deve se conectar com os supermercados para obter a lista de produtos
- RF003 – O sistema deve comparar a lista de produtos dos fornecedores e identificar os produtos iguais
- RF004 – O sistema deve eleger os produtos mais baratos para disponibilizar no supermercado virtual
- RF005 – O sistema deve importar os produtos eleitos mais baratos para o supermercado virtual
- RF006 – O sistema deve fazer a comparação de preços dos produtos dos fornecedores
- RF007 – O sistema deve permitir o cadastro de fornecedores
- RF008 – O sistema deve permitir o cadastro de usuários
- RF009 – O sistema deve se comunicar com os supermercados para informar quais produtos foram comprados pelos consumidores

### **2.1.2.2 Requisitos não funcionais**

- RNF001 - O sistema proposto deve ser desenvolvido em um ambiente Web
- RNF002 – A linguagem de programação a ser utilizada será o PHP
- RNF003 – O banco de dados a ser utilizado será o MySQL
- RNF004 - O sistema deve funcionar nas versões mais recentes dos navegadores conhecidos no mercado: Firefox, Internet Explorer, Chrome e etc.
- RNF005 - Somente usuários cadastrados poderão utilizar a interface administrativa do sistema
- RNF006 – As interfaces devem ser simples, claras e objetivas

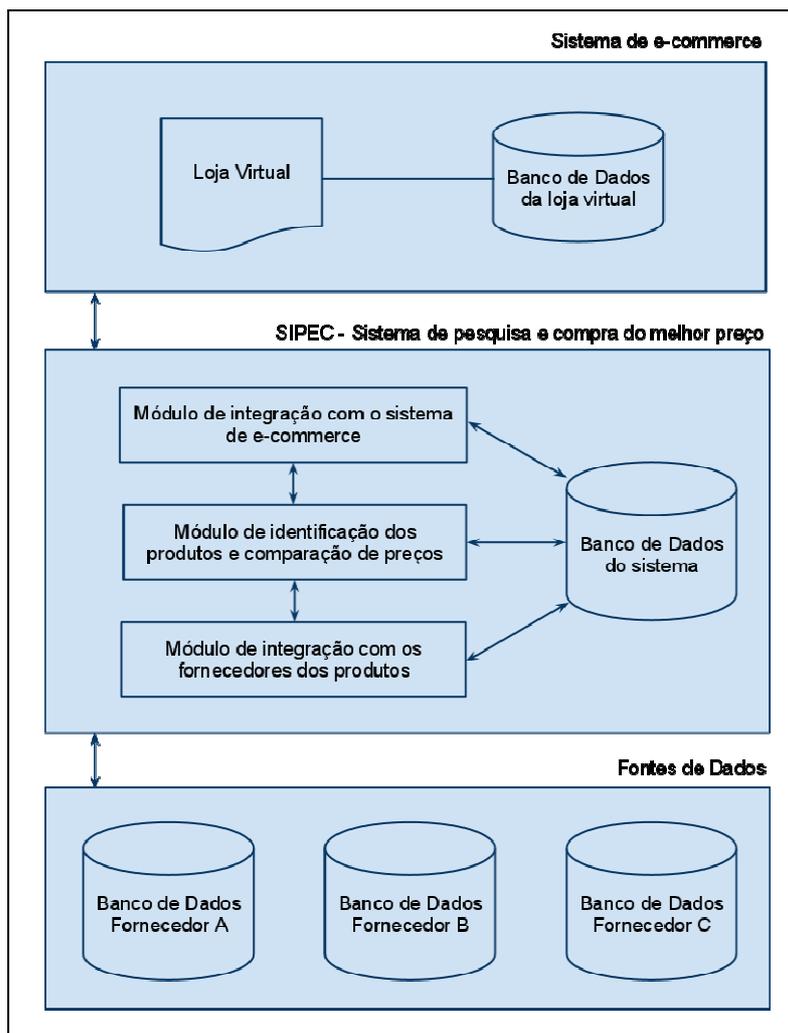
## **2.2 Projeto**

Para que o sistema levantado na fase de análise/concepção seja construído e seus requisitos sejam atendidos, torna-se necessário fazer um planejamento de como isto será conduzido. Wazlawick (2004) diz que “a fase de projeto enfatiza a proposta de uma solução que atenda aos requisitos da análise. Então, se a análise é uma investigação para tentar descobrir o que o cliente quer, o projeto consiste em propor uma solução com base no conhecimento adquirido na análise”.

Teremos então na fase de projeto, as seguintes etapas: arquitetura global do sistema, sistema de *e-commerce*, diagramas *UML* (caso de uso, atividades e classes) e o diagrama de banco de dados (*ER*).

### 2.2.1 Arquitetura global do sistema

A solução proposta para atender ao que foi levantado na fase de análise/concepção, está representada através da arquitetura global do sistema, que pode ser observada na figura 1. Na seqüência, uma breve descrição do comportamento dos componentes desta arquitetura.



**Figura 1. Arquitetura global do sistema**

A arquitetura global do sistema é composta por três componentes macros: o sistema de *e-commerce*, o SIPEC - Sistema *On-line* de Pesquisa e Compra do Menor Preço e as fontes de dados.

Cada fornecedor terá um *Web Service* para que seja feita diariamente a consulta dos produtos. Este *Web Service* retornará um arquivo *XML* com as informações dos produtos (código, título, descrição, preço e etc.).

O módulo de integração com os fornecedores dos produtos acessa o banco de dados de cada fornecedor via *Web Service* e recebe como retorno um arquivo *XML*. Estas informações são armazenadas no banco de dados do sistema, para que o

mapeamento e a identificação dos produtos sejam feitas através do código universal (EAN), e desta forma, cada produto das diversas fontes de dados serão identificados conforme suas características.

Concluída esta etapa, o módulo de identificação dos produtos e comparação de preços é acionado, faz a análise e comparação de preço dos produtos e elege os produtos que possuem os preços mais baratos, armazenando estas informações no banco de dados do sistema.

O módulo de integração com o sistema de e-commerce verifica os produtos que estão no banco de dados do sistema e transfere para o banco de dados da loja virtual, aqueles produtos que foram identificados com os menores preços.

Finalmente, a partir deste momento, os consumidores já podem fazer suas compras normalmente na loja virtual com os produtos atualizados.

### 2.2.2 Sistema de e-commerce

Uma das primeiras atividades relacionadas ao desenvolvimento do sistema proposto neste trabalho foi a pesquisa, entendimento quanto ao funcionamento e escolha de um sistema de *e-commerce*.

Foram pesquisados alguns sistemas de *e-commerce*, obedecendo a alguns critérios pré-estabelecidos:

- Ser *software* livre
- Utilizar a linguagem de programação *PHP*
- Utilizar o banco de dados *MySQL*
- Ter uma *interface* simples e intuitiva
- Possuir a *interface* de administração e a loja virtual
- Suporte a *SSL*

A partir destes critérios, os seguintes sistemas de *e-commerce* foram selecionados para avaliação:

- VirtueMart
- osCommerce
- Magento

Após instalar, configurar e utilizar cada um dos sistemas citados a cima, foi possível ter condições de optar por aquele que mais se adequou aos critérios estabelecidos.

A tabela 1 é um comparativo dos sistemas pesquisados.

**Tabela 1. Quadro comparativo entre os sistemas de e-commerce pesquisados**

<b>Critérios / Sistema</b>	<b>VirtueMart</b>	<b>osCommerce</b>	<b>Magento</b>
<b>Software livre</b>	Atende	Atende	Não atende
<b>Linguagem PHP</b>	Atende	Atende	Atende
<b>Banco de dados MySQL</b>	Atende	Atende	Atende
<b>Interface simples e intuitiva</b>	Atende	Atende	Não atende
<b>Interface de administração e loja virtual</b>	Atende	Atende	Atende
<b>Suporte a SSL</b>	Atende	Atende	Atende

<b>Interface português</b>	<b>em</b>	Atende	Atende	Não atende
--------------------------------	-----------	--------	--------	------------

A partir deste quadro comparativo, podemos perceber que quase todos atendem aos critérios estabelecidos, e que o *VirtueMart* e o *osCommerce* têm uma pequena vantagem em relação ao *Magento*. E para fins de demonstração, o sistema de *e-commerce osCommerce* foi escolhido para ser utilizado neste trabalho.

Na figura 2 podemos observar a tela da loja virtual do *osCommerce*.

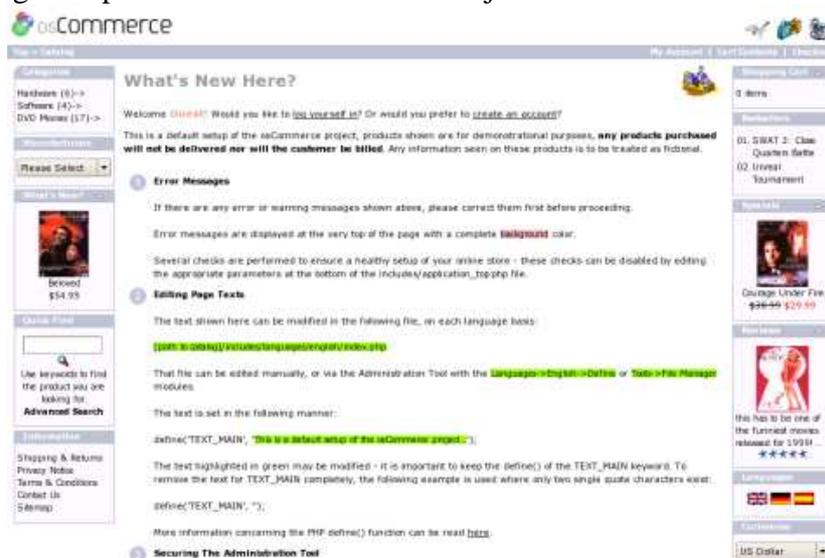
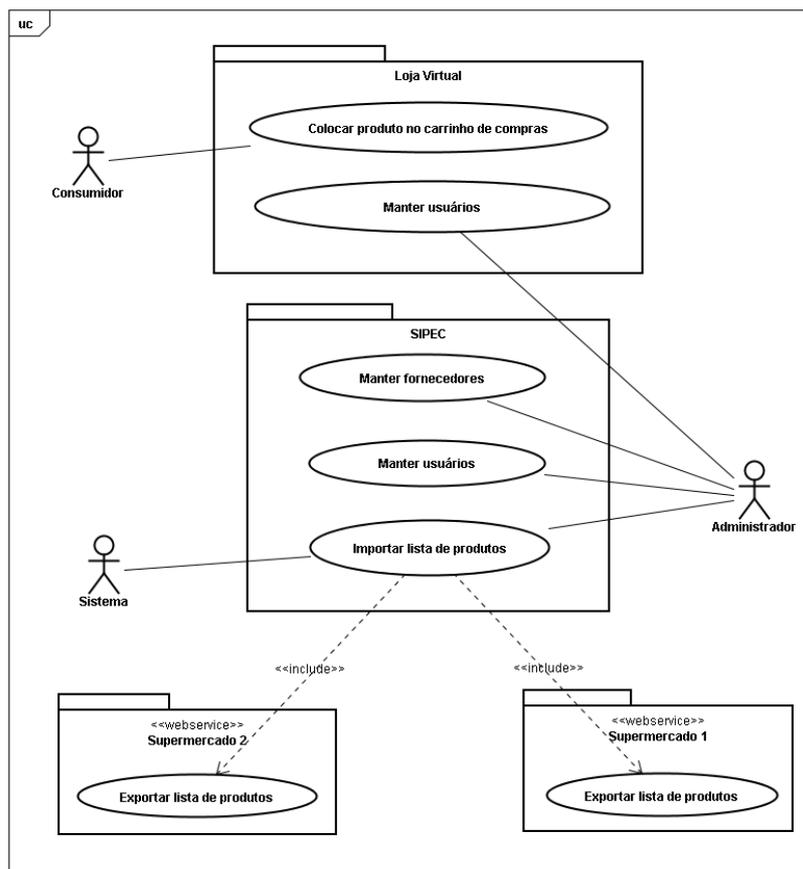


Figura 2. Tela da loja virtual do *osCommerce*

### 2.2.3 Diagramas UML

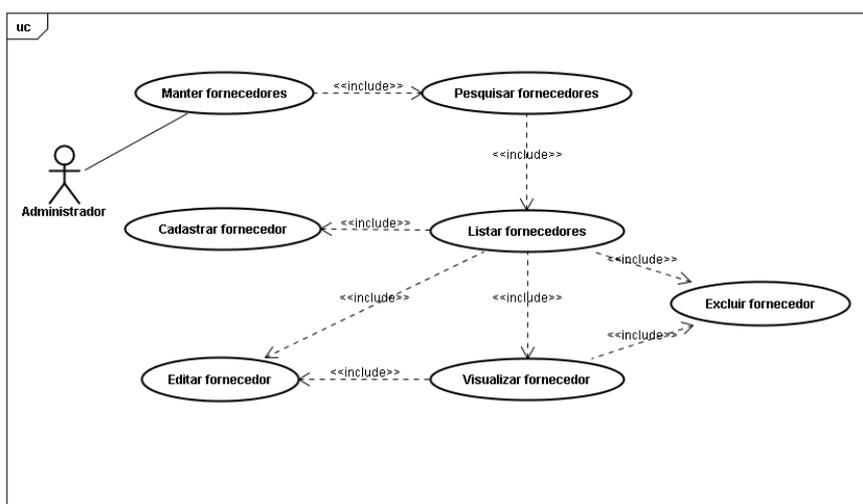
Através de um diagrama de caso de uso, serão modelados os comportamentos de forma superficial, dando assim uma visão geral do sistema. Posteriormente, as funcionalidades mais importantes do sistema, serão detalhadas em outros diagramas de caso de uso e também através de diagramas de atividades.

A figura 3 é um diagrama de caso de uso que representa uma visão geral do sistema.

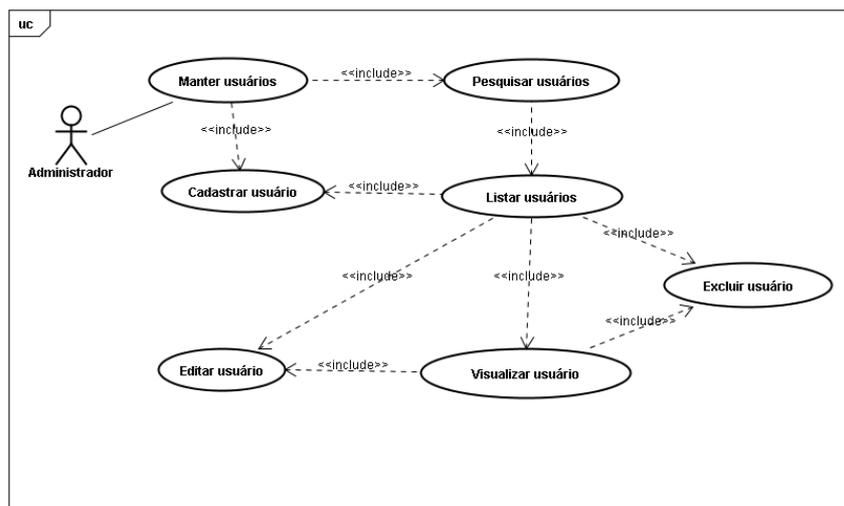


**Figura 3. Diagrama de caso de uso da visão geral do sistema**

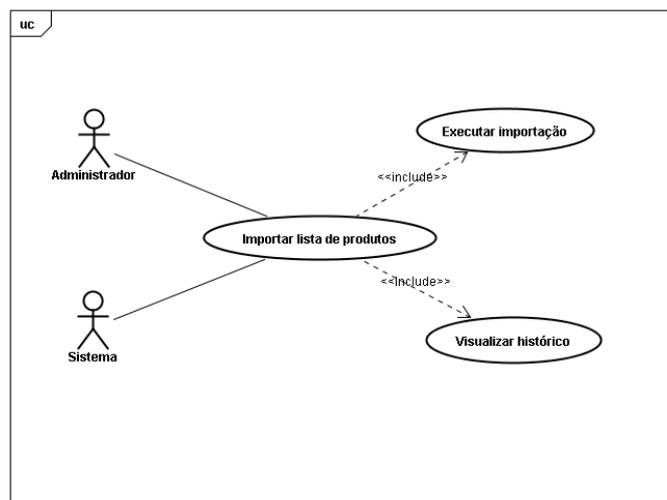
Para uma melhor compreensão das funcionalidades da fronteira central do sistema (SIPEC), os seguintes casos de uso foram detalhados: manter fornecedores, manter usuários e importar lista de produtos. Podemos observá-los abaixo nas Figuras 4, 5 e 6 respectivamente.



**Figura 4. Diagrama de caso de uso manter fornecedor (detalhado)**

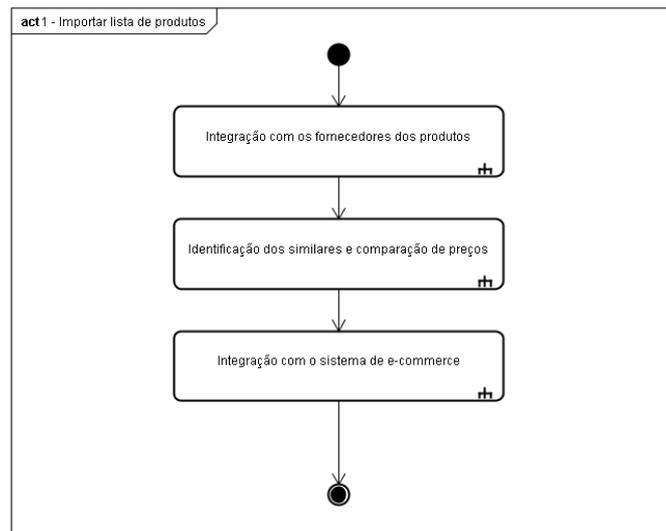


**Figura 5. Diagrama de caso de uso manter usuários (detalhado)**



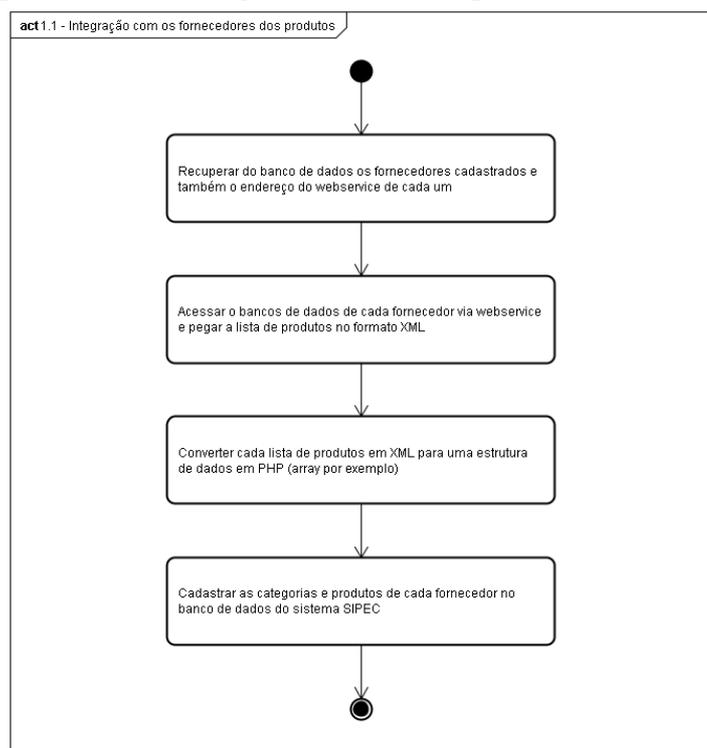
**Figura 6. Diagrama de caso de uso importar lista de produtos (detalhado)**

O caso de uso Importar lista de produtos representa uma das funcionalidades principais do sistema, e por este motivo, ele será detalhado em alguns diagramas de atividades. Na Figura 7, é apresentado o diagrama de atividades com uma visão macro desta funcionalidade.

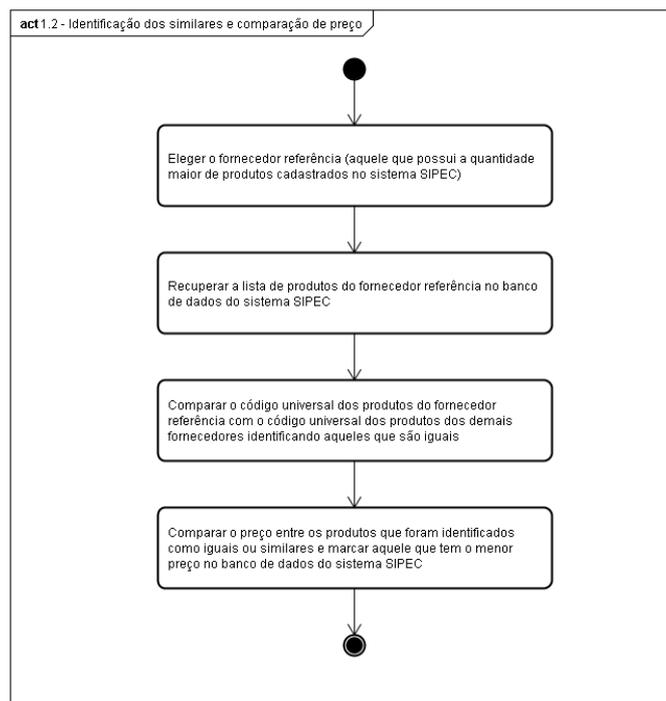


**Figura 7. Diagrama de atividades (macro) da funcionalidade Importar lista de produtos**

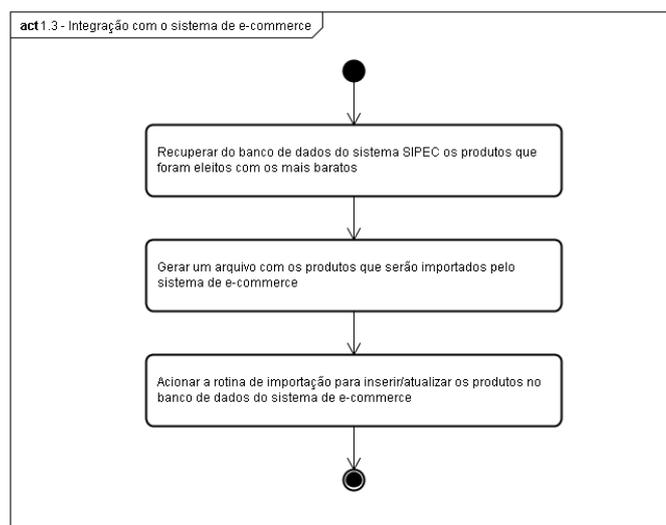
Para compreender o que é feito em cada um dos passos apresentados do diagrama de atividades acima, foi elaborado um diagrama de atividades detalhado para cada uma das etapas: integração com os fornecedores dos produtos; identificação dos similares e comparação de preços; e integração com o sistema de *e-commerce*. Esses diagramas são apresentados nas Figuras 8, 9 e 10 respectivamente.



**Figura 8. Diagrama de atividades (detalhado) da etapa Integração com os fornecedores dos produtos**



**Figura 9. Diagrama de atividades (detalhado) da etapa Identificação dos similares e comparação de preço**



**Figura 10. Diagrama de atividades (detalhado) da etapa Integração com o sistema de e-commerce**

### 2.2.3.1 Diagrama de classes

Para a elaboração do sistema foram criadas 6 classes, sendo que 3 delas são de entidades relacionadas ao negócio propriamente dito e as demais fazem parte de um *plugin* no Code Igniter utilizado para o controle de acesso dos usuários. Na Figura 11, as classes estão agrupadas de acordo com suas afinidades.

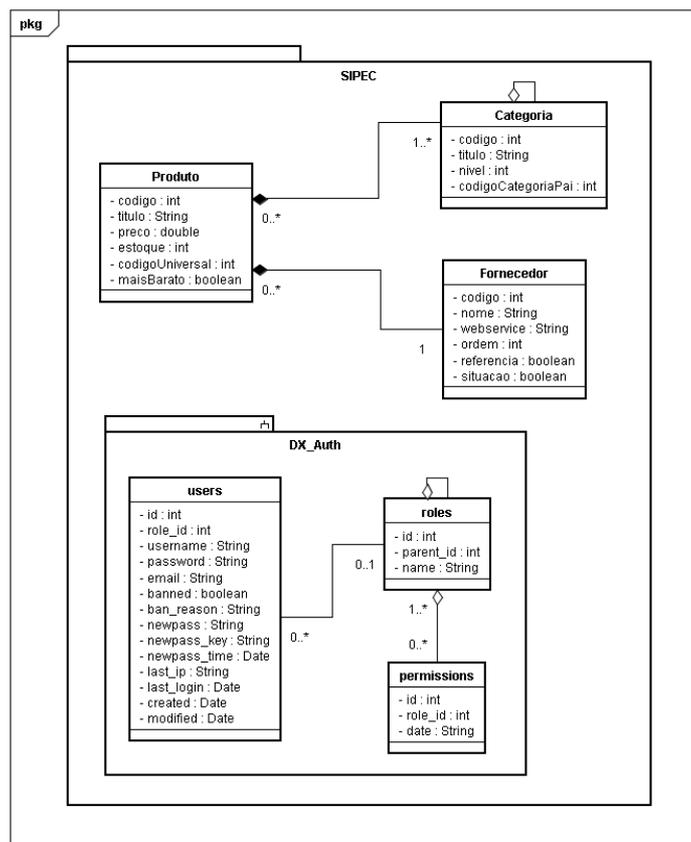


Figura 11. Diagrama de classes do sistema

## 2.2.4 Diagrama do banco de dados

O diagrama ER – Entidade Relacionamento da Figura 12 apresenta todas as tabelas, atributos e seus relacionamentos.

[1.1]

[2.1]

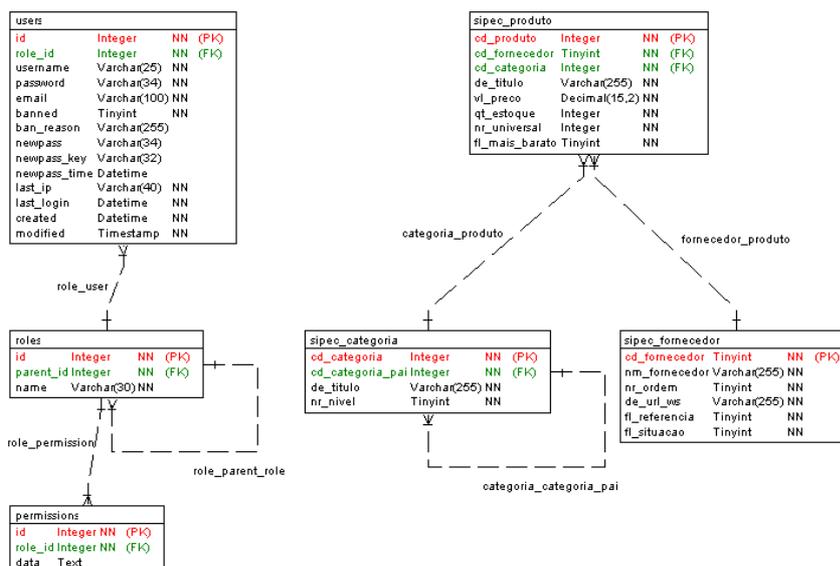


Figura 12. Diagrama ER do banco de dados integrado do sistema

## 2.3 Implementação

Com as fases de análise/concepção e projeto concluídas, vamos partir para a implementação do sistema. Neste momento o sistema é construído de fato.

Teremos então na fase de implementação, as seguintes etapas: criação da base de dados, desenvolvimento das classes e as funcionalidades e telas do sistema.

### 2.3.1 Criação da base de dados

Para criar a base de dados foi utilizado o *script* gerado a partir do diagrama ER do sistema. As tabelas, chaves e *constraints* foram criadas através da importação do *script* com a ferramenta phpMyAdmin 2.10.3.

### 2.3.2 Desenvolvimento do sistema

Todo o sistema foi desenvolvido no *Code Igniter*. Foi utilizada a última versão 1.7.2, que se encontra disponível para download no site oficial do *framework*. Também foi utilizado o *plugin DX Auth* versão 1.0, para o controle de acesso dos usuários.

A instalação é relativamente simples e basta seguir as instruções contidas na figura 13, retirada do site do *Code Igniter*.

#### Instruções de Instalação

O Code Igniter é instalado em 4 passos:

1. Descompacte o arquivo.
2. Suba os diretórios e arquivos para seu servidor. Normalmente o arquivo `index.php` estará ficará raiz.
3. Abra o arquivo `application/config/config.php` em um editor de texto e altere a sua URL base.
4. Se você pretende usar um banco de dados, abra o arquivo `application/config/database.php` file em um editor de texto e altere as configurações do seu banco.

**Figura 13. Instruções de instalação do framework Code Igniter**

Com o ambiente configurado, foram desenvolvidos as classes de acordo com o que foi planejado na fase de projeto. Isto deu resultado as funcionalidades e telas que veremos na próxima etapa.

### 2.3.3 Funcionalidades e telas

O sistema desenvolvido é composto por algumas funcionalidades que serão descritas e mostradas abaixo, através de um texto explicativo e sua respectiva tela.

#### 2.3.3.1 Controle de acesso a interface administrativa do sistema

Para atender ao que foi descrito no requisito não funcional 005 (RNF005 - Somente usuários cadastrados poderão utilizar a *interface* administrativa do sistema), quando um usuário tenta fazer acesso a alguma funcionalidade, o sistema verifica se o mesmo previamente já informou o nome de usuário e a senha. Caso já tenha informado, o sistema permite o acesso ao sistema. Caso não tenha informado ainda, ao tentar acessar a uma determinada funcionalidade, o sistema redireciona o usuário para a tela de autenticação, e solicita que seja informado um nome de usuário e senha válidos.

Podemos observar a tela correspondente no sistema na figura 14 abaixo.

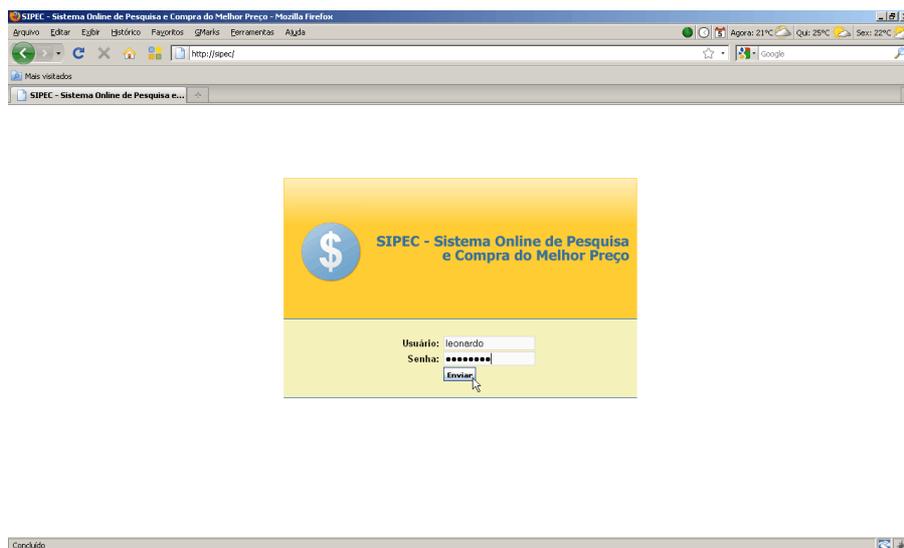


Figura 14. Tela de autenticação da interface administrativa do sistema

### 2.3.3.2 Gerenciar os usuários do sistema

Para atender ao que foi descrito no requisito funcional 008 (RF008 – O sistema deve permitir o cadastro de usuários), através desta funcionalidade, é possível gerenciar os usuários que poderão acessar a *interface* administrativa do sistema.

Na figura 15, onde a tela correspondente é exibida, podemos observar que é possível pesquisar e listar os usuários cadastrados. Através desta tela também podemos cadastrar um novo usuário no sistema, editar suas informações, consultar os detalhes das informações de um usuário, bem como excluir um determinado registro de usuário que não desejamos mais.

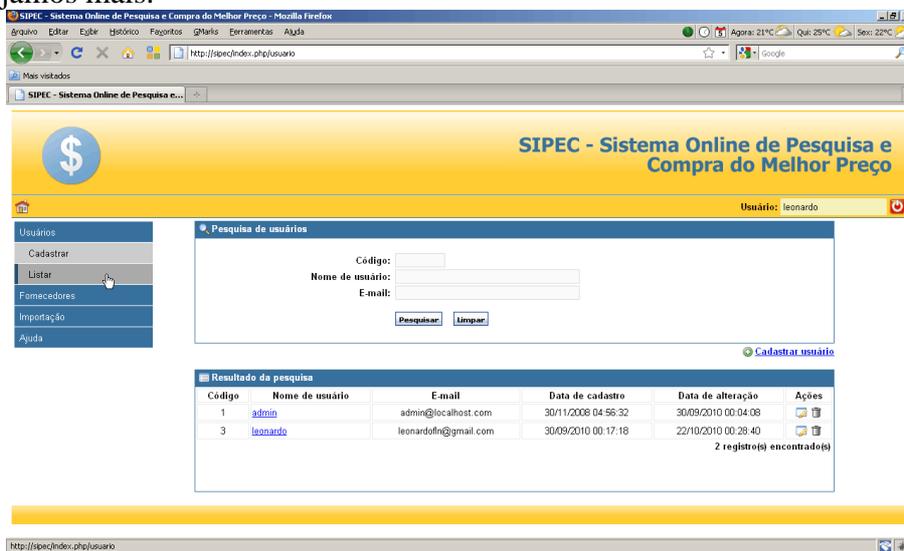


Figura 15. Tela de gerenciamento de usuários (pesquisa, cadastro, edição e etc.)

### 2.3.3.3 Gerenciar os fornecedores do sistema

Para atender ao que foi descrito no requisito funcional 007 (RF007 – O sistema deve permitir o cadastro de fornecedores), através desta funcionalidade, é possível gerenciar

os fornecedores serão acessados através de *Web Service*, para disponibilizar a lista de produtos.

Na figura 16, onde a tela correspondente é exibida e de forma semelhante a tela de usuários, podemos observar que é possível pesquisar e listar os fornecedores cadastrados. Através desta tela também podemos cadastrar um novo fornecedor no sistema, editar suas informações, consultar os detalhes das informações de um fornecedor, bem como excluir um determinado registro de fornecedor que não desejamos mais.

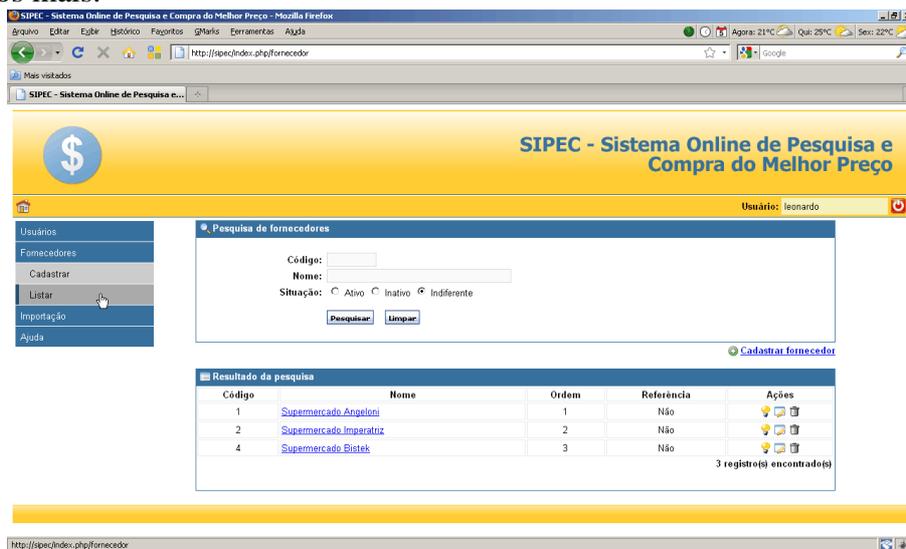


Figura 16. Tela de gerenciamento de fornecedores (pesquisa, cadastro, edição e etc.)

Na figura 17, que é a tela de cadastro de fornecedor, podemos observar o campo onde é informado o endereço do *Web Service*, que será utilizado para obter a lista de produtos deste fornecedor.

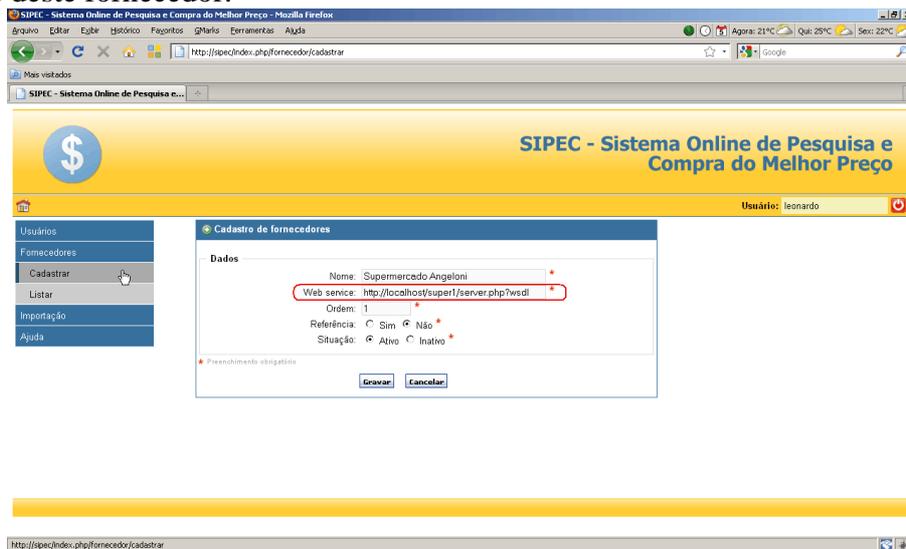


Figura 17. Tela de cadastro de fornecedor com destaque para o campo Web Service

### 2.3.3.4 Processo de Importação dos Produtos

Para atender ao que foi descrito nos requisitos funcionais 002 à 006 (RF002 – O sistema deve se conectar com os supermercados para obter a lista de produtos; RF003 – O sistema deve comparar a lista de produtos dos fornecedores e identificar os produtos iguais; RF004 – O sistema deve eleger os produtos mais baratos para disponibilizar no supermercado virtual; RF005 – O sistema deve importar os produtos eleitos mais baratos para o supermercado virtual; RF006 – O sistema deve fazer a comparação de preços dos produtos dos fornecedores), através desta funcionalidade, ocorre todo o processo de importação dos produtos. Isto compreende desde a conexão via *Web Service* as fontes de dados dos fornecedores, passando pela comparação de preços e identificação dos menores preços, até a carga que é feita no sistema de *e-commerce* (loja virtual).

Na figura 18, podemos observar o processo de execução em andamento.

The screenshot shows a web browser window displaying the SIPEC system interface. The page title is "SIPEC - Sistema Online de Pesquisa e Compra do Melhor Preço". The user is logged in as "leonardo". The main content area is titled "Execução do processo de importação dos produtos em andamento..." and contains a table with the following data:

Atividade	Situação
Início do processo de importação em 15/10/2010 às 03:00	✓
Acessando os web services para obter a lista de produtos	✓
Cadestrando as categorias e produtos no banco de dados do sistema	✓
Identificando os produtos similares e fazendo a comparação de preços	✓
Executando a carga dos produtos no sistema de e-commerce	🔄

Legenda: ✓ concluído 🔄 em andamento

Figura 18. Tela de execução do processo de importação dos produtos

### 2.3.3.5 Sistema e-commerce com a carga dos produtos

Para atender ao que foi descrito no requisito funcional 001 (RF001 – Os consumidores poderão fazer compras através de um supermercado virtual), através desta funcionalidade, o consumidor pode navegar pela loja virtual, escolher os produtos que deseja comprar e colocar no carrinho de compras, finalizar e fechar o seu pedido e etc. Todas essas funcionalidades fazem parte do sistema de *e-commerce* que está integrado ao sistema de pesquisa e compra do menor preço.

Na figura 19, podemos observar a consulta a um determinado produto na loja virtual.



Figura 19. Tela de consulta de um determinado produto na loja virtual

## 2.4 Testes

### 2.4.1 Experimentos realizados

Com o ambiente pronto e com uma carga de produtos (que foi captura manualmente de alguns supermercados), foi possível realizar alguns experimentos no sistema.

O primeiro experimento foi a compra de 63 produtos, que resultou em uma diferença no preço final de R\$19,01 (13%), comparando o valor gerado pelo sistema (os produtos mais baratos de cada supermercado) e o supermercado em que a soma dos valores dos produtos teve o preço mais alto (Supermercado 3). Fazendo uma projeção, em um ano, isto representa uma economia de aproximadamente R\$230,00 para o consumidor. No gráfico apresentado na figura 20, podemos observar o comparativo de preços desta lista de 63 produtos.

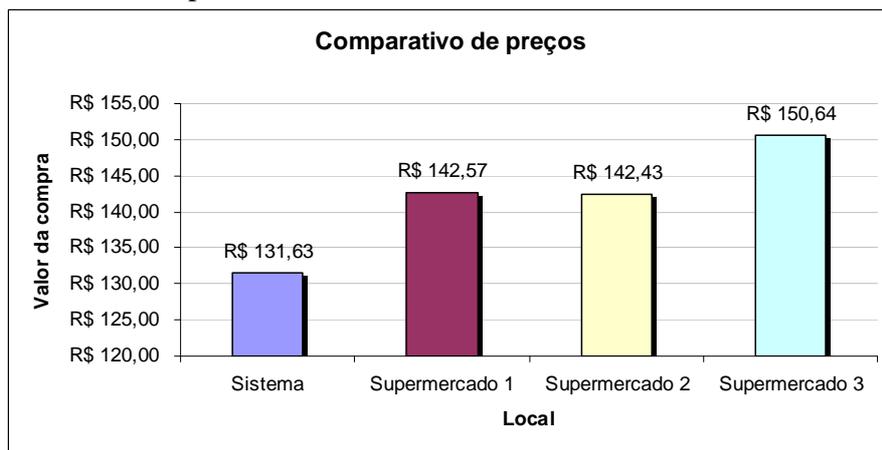


Figura 20. Gráfico com o comparativo de preços da lista com 63 produtos

O segundo experimento foi a compra de 123 produtos, que resultou em uma diferença no preço final de R\$36,40 (12%), comparando o valor gerado pelo sistema (os produtos mais baratos de cada supermercado) e o supermercado em que a soma dos valores dos produtos teve o preço mais alto (Supermercado 3). Fazendo uma projeção, em um ano, isto representa uma economia de aproximadamente R\$435,00 para o consumidor. No

gráfico apresentado na figura 21, podemos observar o comparativo de preços desta lista de 123 produtos.

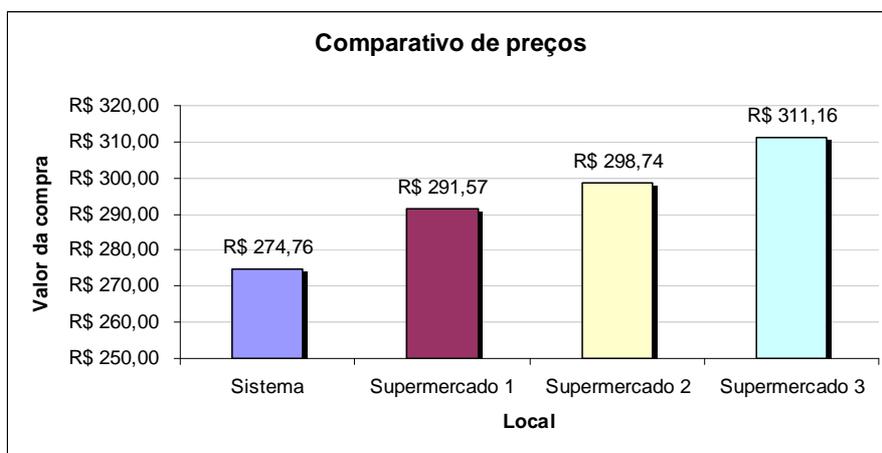


Figura 21. Gráfico com o comparativo de preços da lista com 123 produtos

### 3 Conclusão

Foi possível entender o funcionamento e escolher o sistema de *e-commerce* mais adequado, apesar de que, quase todos eles poderiam ser utilizados. Isto é um ponto positivo, pois se necessário, o sistema de *e-commerce* pode ser substituído, causando baixo impacto no projeto.

O uso de *Web Service* na integração com as fontes de dados, se demonstrou satisfatório e atendeu ao que era esperado. Desta forma, os fornecedores têm mais flexibilidade para exportar a lista de produtos (arquivo *XML*), e também compatibilidade, pois não precisam modificar em nada sua estrutura (banco de dados e linguagem de programação) para poder integrar seus dados com o sistema aqui proposto. Um dos pontos negativos, no que se refere a integração com as fontes de dados, está relacionada a segurança.

Com relação ao mapeamento e a identificação dos produtos, inicialmente iriam ser utilizadas as funções de similaridades do *PHP* e alguns tratamentos nos strings comparados (remoção de: espaços duplicados, espaços no início/fim e pontuação). Porém, esta solução não atendeu as necessidades, devido ao fato de que cada fornecedor mantém o nome dos produtos como bem entende, ou seja, não existe um padrão. Uma alternativa foi utilizar para o mapeamento e a identificação dos produtos, o código *EAN*, que é o padrão de sistema de identificação (código de barras) utilizado nos estabelecimentos comerciais no Brasil.

Já com relação à escolha do menor preço, foi bastante simples, uma vez que os produtos já haviam sido identificados e mapeados, e elaborar um algoritmo para escolha do menor preço foi trivial.

A rotina de importação dos produtos no sistema de *e-commerce* também acabou sendo bem simples, tendo em vista que os dois sistemas utilizam a mesma linguagem de programação (*PHP*) e banco de dados (*MySQL*).

Finalmente, foi possível realizar alguns testes e demonstrar que, através dos experimentos realizados, o consumidor consegue fazer uma economia de aproximadamente 13% em suas compras realizadas através do sistema de pesquisa e compra do menor preço.

#### **4 Referências**

Code Igniter - Guia do Usuário. Disponível em:  
<<http://www.codeigniter.com.br/manual/index.html>>. Acesso em: 19 jun. 2010.

Magento - eCommerce Software for Growth Disponível em:  
<<http://www.magentocommerce.com/>>. Acesso em: 19 jun. 2010.

osCommerce, Open Source Online Shop E-Commerce Solutions. Disponível em:  
<<http://www.oscommerce.com/>>. Acesso em: 19 jun. 2010.

VirtueMart. Disponível em: <<http://www.virtuemart.net/>>. Acesso em: 19 jun. 2010.

WAZLAWICK, Raul Sidnei. Análise e Projeto de Sistemas de Informação Orientados a Objetos. Rio de Janeiro: Elsevier, 2004. 298 p.