

UNIBRASIL – FACULDADES INTEGRADAS DO BRASIL
RODRIGO RAMOS

**DESENVOLVIMENTO DE UMA FERRAMENTA PARA COMPARAÇÃO *ON-LINE*
DE PREÇO DE MERCADORIAS EM SUPERMERCADOS**

CURITIBA

2010

RODRIGO RAMOS

**DESENVOLVIMENTO DE UMA FERRAMENTA PARA COMPARAÇÃO *ON-LINE*
DE PREÇO DE MERCADORIAS EM SUPERMERCADOS**

Trabalho de Conclusão de Curso, apresentado como
requisito parcial para aprovação no Curso de Sistemas
de Informação das Faculdades Integradas do Brasil –
Unibrasil.

Orientação: Prof. Rodrigo Fiorin

CURITIBA

2010

RESUMO

Diariamente, milhares de pessoas vão aos supermercados comprar mercadorias para suprir uma de suas necessidades básicas que é a alimentação, entretanto, com a rotina apressada, muitas vezes a decisão de onde comprar as mercadorias é tomada sem levar em consideração as opções existentes. Nesse contexto, a Internet, a qual tem se tornado um veículo de pesquisa cada vez mais difundido, permanece inerte, pois não há um mecanismo que faça uma pesquisa de preços diretamente nos supermercados. A fim de apresentar uma solução para essa questão, o presente trabalho propõe elaborar uma solução capaz de encontrar os melhores preços das mercadorias disponíveis nos supermercados, para isso, são utilizadas ferramentas e técnicas amplamente difundidas no mercado, tal como Web Services.

Palavras chaves: Web Service, XML, .NET Framework.

ABSTRACT

Every day, thousands of people go to supermarkets to buy goods to fill one of their basic needs, feed themselves. However with the hurried routine, the decision on where to buy the goods is often taken without considering their options. In this context the Internet has become a vehicle for research increasingly widespread idle, since there is no mechanism to make a survey of prices in supermarkets directly. In order to provide a solution to this issue, this paper proposes to prepare a solution able to find the best prices of goods available in supermarkets, for this, it makes use of tools and techniques widely distributed in the market, such as Web Services.

Key words: Web Service, XML, .NET Framework.

LISTA DE FIGURAS

Figura 1: Funcionamento básico de Web Services. Fonte [12].....	20
Figura 2: Visão Geral da Solução.....	33
Figura 3: Diagrama de caso de uso.....	36
Figura 4: Estrutura das camadas.....	37
Figura 5: Diagrama de Classes de Usuários	38
Figura 6:Diagrama de classes do mercados.....	39
Figura 7: Diagrama de classes das mercadorias	40
Figura 8: Diagrama de classes dos anúncios.	41
Figura 9: Diagrama de classes comparação de preços	41
Figura 10: Diagrama de Entidade e Relacionamento	43
Figura 11: Configuração da string de conexão.....	44
Figura 12: Configuração de Roles.....	45
Figura 13: Configuração de Membership.....	45
Figura 14: Configuração de profile	46
Figura 15: Interação entre as camadas.....	47
Figura 16: Consumo do Web Service.....	48
Figura 17: Chamadas paralelas aos Web Services	48
Figura 18: Tempo de resposta Web Services	49
Figura 19: Código região crítica.....	50
Figura 20: Página inicial da solução.....	51
Figura 21: Página de acesso ao sistema.....	52
Figura 22: Mercadorias cadastradas	52
Figura 23: Cadastro de Mercadorias.....	53
Figura 24: Confirmação de exclusão	53
Figura 25: Criação da lista de compra	54
Figura 26: Pesquisa de Preços	54
Figura 27: Resultado geral da comparação	55
Figura 28: Página com os menores preços das mercadorias	55
Figura 29: Preços da lista do supermercado	56
Figura 30: Impressão da lista de compras.	56
Figura 31: Relatório do resultado de teste de desempenho	59
Figura 32: Estatística do teste de desempenho	60
Figura 33: Estatística do teste de carga	60
Figura 34: Gráfico de indicadores chaves do teste de carga	61
Figura 35: Relatório do resultado de teste de stress	62
Figura 36: Gráfico de indicadores chaves do teste de stress	62

LISTA DE TABELAS

Tabela 1: Comparaçao de projetos similares	30
Tabela 2: Comparaçao entre os resultado dos testes	64

LISTA DE SIGLAS

XML	-	Extensible Markup Language
ONU	-	Organização das Nações Unidas
HTTP	-	Hypertext Transfer Protocol
API	-	Application Programming Interfaces
B2B	-	Bussines to Bussines
CLR	-	Common Language Runtime
CPU	-	Central Processing Unit
WSDL	-	Web Service Description Language
ACT	-	Application Center Test
IBGE	-	Instituto Brasileiro de Geografia e Estatística
HTML	-	HyperText Markup Language
UI	-	User Interface
DAL	-	Data Access Layer
BLL	-	Bussines Logig Layer
DTO	-	Data Transfer Object
SQL	-	Structured Query Language
ANSI	-	American National Standards Institute
EDI	-	Eletronic Data Interchange
WSDL	-	Web Server Definition Language
VPN	-	Virtual Private Network
PKI	-	Public Key Interchange
SSL	-	Secure Socket Layer
WS	-	Web Service
CEP	-	Código de Endereçamento Postal
IP	-	Internet Protocol
SMS	-	Short Message Service
ERP	-	Enterprise Resource Planning
DMR	-	Diagrama de modelagem Relacional
HTTP	-	Hyper Text Transfer Protocol

SUMÁRIO

1	INTRODUÇÃO	10
1.1	OBJETIVOS	10
1.2	JUSTIFICATIVA.....	11
1.3	ORGANIZAÇÃO DO TRABALHO.....	12
2	FUNDAMENTAÇÃO TEÓRICA	13
2.1	.NET FRAMEWORK	13
2.2	ASP.NET.....	14
2.3	C#	14
2.4	APPLICATION SERVICES.....	15
2.5	PADRÕES DE PROJETO	16
2.5.1	Desenvolvimento Multi-Camadas.....	16
2.6	COMUNICAÇÃO DE DADOS.....	17
2.7	BANCO DE DADOS.....	18
2.8	BUSSINES TO BUSSINES (B2B).....	18
2.9	XML - EXtensible Markup Language	19
2.10	WEB SERVICE	20
2.10.1	SOAP – Simple Object Acesso Protocol.....	21
3	TRABALHOS RELACIONADOS	22
3.1	INTERCÂMBIO ELETRÔNICO DE DADOS	22
3.2	DESAFIOS NA CONSTRUÇÃO E VALIDAÇÃO DA ROBUSTEZ DE APLICAÇÕES ORIENTADAS A SERVIÇOS	23
3.3	ARQUITETURA DE SEGURANÇA EM APLICAÇÕES BASEADAS EM WEB SERVICE	24
3.4	PROJETO PARA EXECUÇÃO DE CONSULTAS DISTRIBUÍDAS USANDO WEB SERVICES	25
3.5	WEB SERVICES COMERCIAIS	26
3.5.1	Correios	26
3.5.2	Xignite.....	27
3.5.3	Yahoo Web Service.....	27
3.5.4	Master Card Web Services	27
3.5.5	Mapa de Endereço IP.	28
3.5.6	Condições do Clima em Cidades dos Estados Unidos	28
3.5.7	Mapa de Endereços.....	28
3.5.8	Geração de Código de Barras	28
3.5.9	WEB SMS	29

3.6	ANÁLISE DE PROJETOS COMERCIAIS SIMILARES.....	29
4	ESPECIFICAÇÃO.....	32
4.1	VISÃO GERAL	32
4.2	ANÁLISE DA SOLUÇÃO	34
4.2.1	Lista de Requisitos Funcionais.....	34
4.2.2	Diagrama de Caso de Uso.	35
4.2.3	Estrutura do sistema	37
4.2.4	Banco de dados	42
4.3	IMPLEMENTAÇÃO DO CÓDIGO.....	44
4.3.1	Autenticação de usuários.....	44
4.3.2	Interação entre as camadas	46
4.3.3	Consumo do Web Service	47
4.3.4	Programação paralela	48
4.4	PROJETO DE INTERFACE COM O USUÁRIO	50
4.5	TESTES E EXPERIMENTOS	57
4.5.1	Objetivos	57
4.5.2	Configuração do ambiente.....	57
4.5.3	Execução dos testes	58
5	CONCLUSÃO	64
	REFERÊNCIAS BIBLIOGRÁFICAS	66
	APENDICE	70
	Especificação de caso de uso	70

1 INTRODUÇÃO

A evolução da tecnologia, principalmente no que se aplica a computadores, tomou rumos que há pouco tempo eram inimagináveis. O incentivo ao avanço tecnológico, assim como a constante divulgação das informações, fez com que o computador deixasse de ser um artigo restritivo ao uso para fins governamentais, bancários e corporativos, e passou a estar disponível para uso pessoal, tanto como ferramenta de pesquisa, trabalho, comunicação como de entretenimento.

Na história da evolução tecnológica, observam-se períodos em que algumas tecnologias tornam-se o foco das atenções, existiram épocas nas quais os hardwares estavam em evidência, houve o tempo dos sistemas operacionais, banco de dados e, atualmente, a Internet está em seu ápice.

A Internet, que inicialmente era utilizada somente para troca de informações entre indivíduos, atualmente possibilita a troca de informações entre aplicações, isto é, empresas distintas podem compartilhar seus dados com outros *softwares*, processo conhecido como *bussines-to-bussines(B2B)*. O benefício de tal paradigma atualmente tem se limitado a otimizar os processos empresariais, entretanto, esse recurso é pouco utilizado em prol dos cidadãos em suas rotinas diárias.

Apesar de tanta tecnologia disponível, algumas tarefas do dia-a-dia ainda são feitas de modo convencional, como, por exemplo, pesquisar preços de mercadorias para compras em supermercado.

1.1 OBJETIVOS

O presente trabalho tem como objetivo propor uma solução tecnológica que possibilite a disponibilização dos preços das mercadorias dos supermercados e um mecanismo de pesquisa desses preços pelos consumidores por meio de portal Website.

Tal ferramenta será um portal no qual consumidores podem criar suas listas de compras, ou então, pesquisar preços de mercadorias individualmente em supermercados de sua preferência.

Para atingir tal objetivo, estima-se que as seguintes premissas compõem os objetivos específicos em cada etapa do projeto:

- Pesquisar o cenário atual;
- Levantar requisitos funcionais do sistema;

- Pesquisar e escolher a tecnologia adequada para o desenvolvimento do projeto;
- Pesquisar projetos similares disponíveis no mercado;
- Efetuar a análise e projeto do sistema;
- Efetuar codificação e testes.

1.2 JUSTIFICATIVA

Toda tecnologia existente visa a alguma forma de melhorar a vida do homem, sendo que as soluções tecnológicas nascem de uma necessidade local e se tornam não apenas ferramentas e aplicativos, mas processos e soluções a serem implementados. Pode-se dizer que a tecnologia será social quando estiver servindo aos propósitos do desenvolvimento humano, em suas dimensões culturais, econômicas e ambientais [1].

Sendo a tecnologia voltada à sociedade, a ausência de uma ferramenta que venha auxiliar na rotina diária de compras de alimentos em supermercados abre uma lacuna sobre o quanto a sociedade tem realmente utilizado os recursos da Internet para benefícios pessoais. Mesmo com a competitividade entre as empresas, a variação no preço de mercadorias entre diferentes supermercados chega em alguns casos a ultrapassar 100%.

Segundo [21], o simples hábito de criar uma lista de compra pode levar a uma redução de 15% dos gastos em supermercado, se, acrescentado à lista de compras, uma criteriosa pesquisa de preços, essa taxa de economia pode ser elevada significativamente, com isso, os consumidores podem comprar mais produtos e/ou então comprar produtos de melhor qualidade, e até direcionar o dinheiro pouparado para outros fins.

Pesquisas realizadas pelo IBGE [2] mostram que apenas 53,36% da população têm acesso a uma quantidade de alimentos suficiente para satisfazer suas necessidades.

Embora nem todos tenham computador e/ou Internet em casa, a grande maioria das pessoas que vivem nos centros urbanos conhece alguém que tem acesso a tal tecnologia ou então tem acesso a esses recursos no ambiente de trabalho.

Em Curitiba, por exemplo, existem os Faróis do Saber [19] que propiciam à população acesso gratuito à Internet, estendendo o benefício de uma ferramenta de pesquisa a todas as classes sociais.

1.3 ORGANIZAÇÃO DO TRABALHO.

Esta monografia está descrita da seguinte forma:

O capítulo 2 é destinado a explorar as teorias existentes que alicerçam o desenvolvimento prático deste projeto, tais como, .Net Framework, XML, Web Service entre outros.

O capítulo 3 é destinado a apresentar trabalhos científicos cujos temas estão relacionados ao presente trabalho. Ainda no capítulo 3, são apresentados os Web Services que são utilizados no mercado atualmente, e também é feita uma comparação entre o presente projeto e duas soluções disponíveis no mercado atualmente.

No capítulo 4, é descrita a especificação do projeto aqui proposto, apresentando dados desde a análise até a descrição dos testes que serão executados.

E, por fim, o capítulo 5 é destinado a explanar sobre as considerações finais.

2 FUNDAMENTAÇÃO TEÓRICA

A fundamentação teórica deste projeto é resultado de uma ampla pesquisa em artigos, livros, revistas e Internet. Essas pesquisas fundamentam a teoria que alicerça o desenvolvimento prático da aplicação, fornecendo uma base de conhecimentos importantes para um melhor entendimento desta monografia.

2.1 .NET FRAMEWORK

O .NET Framework será utilizado para o desenvolvimento prático deste projeto, sendo que, nesta seção, é apresentada a arquitetura básica desse framework, assim como os elementos principais que o compõem, a CLR, biblioteca de classes, a linguagem de programação para Web ASP.NET e a linguagem de programação C#.

A evolução tecnológica ampliou os horizontes e revolucionou a indústria de software. Atualmente um grande número de tecnologias está disponível para o desenvolvimento de aplicações. Esse grande leque de tecnologias disponíveis, tais como linguagem de programação, banco de dados, frameworks, sistemas operacionais e metodologias de desenvolvimento, representa a liberdade de escolha nos projetos de softwares.

Dentre as tecnologias disponíveis, existe o .NET Framework da Microsoft, o qual foi adotado para o desenvolvimento prático do presente trabalho.

O .NET Framework consiste em uma camada de software que visa facilitar e disponibilizar recursos aos desenvolvedores, ele possibilita que sistemas sejam criados e executados sob sua arquitetura. De acordo com [3], o .NET Framework oferece um ambiente de desenvolvimento completo, ou seja, todos os recursos necessários para o desenvolvimento de uma aplicação são encontrados em um único fornecedor. O .NET Framework é formado basicamente por dois componentes principais, a CLR(*Common Language Runtime*) e a biblioteca de classes.

A CLR pode ser considerada uma camada intermediária entre a aplicação e o sistema operacional. A mesma oferece a capacidade de compilar determinada aplicação uma única vez e executar em qualquer CPU ou sistema operacional Windows que o suporte [4].

Outro componente fundamental do .NET Framework é a biblioteca de classes. Essa, por sua vez, fornece aos desenvolvedores uma variedade de classes prontas que podem ser utilizadas para escrever o código, aumentando assim a performance e facilidade no desenvolvimento.

Vale ressaltar que a Microsoft distribui este framework e IDEs para desenvolvimento de forma gratuita.

2.2 ASP.NET

Esta seção é destinada a explanar sobre o ASP.NET, linguagem de programação para Web utilizada neste trabalho.

Como o desenvolvimento de aplicações para Internet tem se tornado mais popular a cada dia, a Microsoft criou o ASP.NET, um ambiente para criação de páginas HTML dinâmicas, ou seja, páginas cujo conteúdo altera em cada requisição. Geralmente páginas dinâmicas têm seu conteúdo gerado de acordo com alguns parâmetros providos pelo usuário, essas páginas quando requisitadas necessitam ser interpretadas para que alguma ação ocorra antes de a resposta ser enviada ao usuário. O ASP.NET reflete a estratégia de proporcionar alto desempenho, facilidade de desenvolvimento e robustez no desenvolvimento de sistemas para plataforma Web[5].

ASP.NET, é a tecnologia utilizada para geração das páginas dinâmicas do sistema Web aqui proposto, como por exemplo, cada usuário do sistema define suas próprias listas de compras com as mercadorias que desejam pesquisar, então, não se pode ter uma página estática com uma lista de mercadorias, pois essa página será construída dinamicamente de acordo com cada usuário.

2.3 C#

Esta seção aborda aspectos básicos sobre a linguagem de programação C#, a qual será utilizada no desenvolvimento prático desse projeto, focando em apresentar a origem dessa linguagem e seu objetivo.

A linguagem de programação C# foi desenvolvida pela Microsoft, para ser a linguagem padrão para o framework .NET, as próprias bibliotecas de classes do .NET Framework, mencionadas anteriormente, foram escritas em C# mostrando o poder dessa linguagem.

C# foi baseado em C++ e Java, duas linguagens amplamente utilizadas e conceituadas, entretanto C# veio para cobrir uma lacuna, que é acrescentar simplicidade em uma linguagem poderosa. Muitas linguagens poderosas existentes no mercado acabam dificultando o desenvolvimento e consequentemente a produtividade [22].

C# se propõe a ser uma linguagem poderosa e que oferece produtividade, abstraindo dos desenvolvedores muitos dos problemas encontrados em outras linguagens.

2.4 APPLICATION SERVICES

É fato que garantir a autenticidade de usuários é um problema de segurança que está além dos limites deste projeto, entretanto, o *.Net Framework* fornece uma solução simples e eficaz para implementar a autenticação e autorização em soluções ASP.NET, esta solução é chamada *Application Services*. O presente trabalho utiliza os recursos fornecidos pelo *Application Services* para autenticar os usuários do sistema, assim como, gerenciar quais páginas os mesmos têm permissão de acesso.

O Application Service do .NET Framework é composto pelos seguintes recursos.

- *Membership API* – Este recurso fornece um conjunto de funcionalidade para incluir, alterar e apagar informações dos usuários, além de recuperação da senha, em que o usuário recebe uma nova senha por e-mail, caso a tenha esquecido [36].
- *Roles API* – Como o próprio nome sugere, o recurso de roles agrupa os usuários por papéis, isto é , por perfil de usuário. O presente trabalho possui três perfis de usuários, que são:
 - Administrador do sistema - Responsável por cadastrar as marcas, grupos, mercadorias e mercados, além de aprovar anúncios.
 - Anunciante - Usuário interessado em veicular algum anúncio no site, o usuário com perfil anunciante tem permissão de acesso ao cadastro de anúncios e ao seu respectivo perfil.
 - Consumidor – Usuário deste perfil tem acesso à criação de listas e pesquisas de preços.

Através dos *roles* é possível definir as regras de autorização de acesso para cada perfil do usuário, dessa forma, os membros de um perfil têm acesso somente aos arquivos autorizados para o respectivo perfil [36].

- *Profiles API* – Esse recurso fornece mecanismos para recuperação do usuário autenticado no sistema, é através do *profile* que é possível identificar o usuário em qualquer página da aplicação, por exemplo, cada usuário do sistema tem uma lista de compra com suas mercadorias. Quando diversos usuários estão utilizando a aplicação

simultaneamente se faz necessário identificar o usuário logado a fim de recuperar sua respectiva lista de compra, para isto faz-se uso dos recursos oferecidos pelo *profile* que gerencia e retorna essa informação armazenada em um banco de dados como o SQL Server [36].

Ao utilizar os recursos citados acima, todo o processo de persistência, recuperação dos dados dos usuários e autorização de acesso é gerenciado pelo .NET sem a necessidade de codificação, bastando apenas efetuar algumas configurações no arquivo web.config, responsável pela configuração da aplicação.

2.5 PADRÕES DE PROJETO

É evidente que o processo de desenvolvimento de sistemas é complexo. Mesmo que muitos fornecedores de ferramentas procurem facilitá-lo com linguagens e recursos facilitadores, os problemas relacionados a ele são reincidentes em diversos projetos.

Este trabalho utiliza algumas das melhores práticas em desenvolvimento de software apontadas pelos padrões de projetos publicados. Como por exemplo, multi-camadas.

Padrões de projetos independem de linguagem de programação ou sistema operacional, pois seu fundamento está em buscar soluções conceituais que podem ser aplicadas em qualquer linguagem de programação com suporte ao paradigma de orientação a objetos [23].

2.5.1 Desenvolvimento Multi-Camadas

O modelo de desenvolvimento de sistemas tem evoluído bastante nos últimos anos, essa evolução tem melhorado a produtividade dos profissionais e a qualidade dos softwares.

O desenvolvimento em camadas é um conceito arquitetural que visa separar o código da aplicação de acordo com sua respectiva responsabilidade. Essa separação traz maior controle do código, tornando-o mais fácil para desenvolver, modificar, testar e reaproveitar [6]. Esse conceito de separação em diversas camadas foi absorvido por um padrão de projetos chamado MVC(Model View Controller), o qual será apresentado a seguir.

O presente trabalho faz uso do conceito multi-camadas, para o desenvolvimento prático do projeto.

De acordo com [7], O Padrão MVC é composto pelas seguintes camadas:

- Model: A camada de modelo é a parte da aplicação que implementa a lógica da estrutura dos dados da aplicação, frequentemente encontram-se nesta camada o código responsável por recuperar e armazenar dados em objetos que serão utilizados por outras camadas. Neste projeto as classes que compõem essa camada receberá a sigla DAL referente à Data Access Layer;
- View: Visão é o componente responsável por mostrar a interface de usuário (UI). Tipicamente, essa UI fornece ao usuário um mecanismo para interagir com o sistema, ou seja, as telas dos sistemas ou sites e todo o código para criar e validar controles com os quais o usuário pode interagir;
- Controllers: Controle é uma camada intermediária entre a visão e o modelo. As regras de negócio são implementadas nessa camada. As classes dessa camada possuem a nomenclatura BLL referente à Business Logic Layer. [7];

A complexidade no desenvolvimento de softwares e o aumento do número de dispositivos com capacidade para executar aplicações tem motivado o uso do padrão MVC, pois, com a separação das camadas, o entendimento do código do software torna-se mais fácil e consequentemente a sua manutenção, pois os problemas podem ser isolados e identificados rapidamente.

Esse baixo acoplamento entre as camadas também facilita o reuso do código entre diferentes projetos, maximizando assim a produtividade dos profissionais e diminuindo a possibilidade de erros, pois o código reutilizado pode já estar testado e aprovado.

2.6 COMUNICAÇÃO DE DADOS

Dentre os fatores que alavancaram o avanço da humanidade, está a capacidade de comunicação. Para que uma comunicação seja eficiente, é necessário que a mensagem passada pelo locutor seja entendida pelo receptor, para isso, os mesmos devem entender o idioma, jargões ou até mesmo gírias utilizadas na comunicação.

Nesse contexto, entram os protocolos, que definem como se dá a comunicação entre os envolvidos. O presente trabalho faz uso do protocolo HTTP para comunicação de dados, utilizado principalmente na Internet, o qual está brevemente descrito a seguir.

O *HyperText Transfer Protocol* (HTTP), funciona normalmente com um cliente, geralmente o software utilizado para navegar na Internet, como o Internet Explorer,

enviando um pedido, na forma de uma mensagem em formato de texto a um servidor na Web. O servidor, por sua vez, interpreta a mensagem recebida e envia a resposta, também na forma de uma mensagem ao cliente. O HTTP define o padrão dessas mensagens para que tanto o cliente como o servidor as entendam. [14]. As trocas de mensagem da aplicação proposta no presente trabalho serão transferidas sob o protocolo HTTP, pois o mesmo é amplamente utilizado na internet.

2.7 BANCO DE DADOS

Esta seção aborda o conceito básico sobre banco de dados, a fim de fornecer informações de como um banco de dados é estruturado. O presente trabalho faz uso de banco de dados para armazenar e manipular as informações sobre os usuários, mercadorias, anunciantes e anúncios veiculados no sistema.

Evidentemente, o mundo corporativo está repleto de dados; dados sobre os mais diversos segmentos e para as mais diversas finalidades. Sendo assim, geralmente quando um sistema computacional necessita manipular e armazenar dados, o caminho natural para isso é fazer uso de banco de dados.

Banco de dados pode ser definido como um local onde se tem a possibilidade de armazenar informações e manipulá-las, ou seja, adicionar novos dados, editá-los e excluí-los. [20]

Um banco de dados é formado basicamente por tabelas, campos e relacionamentos, e a sua manipulação pode ser feita através de linguagem SQL (Structured Query Language). A linguagem SQL é um padrão de codificação e semântica para recuperar, alterar, incluir, apagar dados em um banco de dados, embora o SQL seja um padrão aberto definido pelo ANSI (American National Standards Institute).

Muitos bancos de dados acrescentam recursos proprietários de forma que o SQL é estendido e cada banco de dados, pode ter comandos SQL específicos, contudo o padrão SQL ANSI é funcional na maioria dos banco de dados [24].

2.8 BUSSINES TO BUSSINES (B2B)

Esta seção tem como objetivo apresentar o conceito de *bussines to bussines*, no qual o presente trabalho está inserido.

Há pouco tempo, a comunicação e transferência de informações entre organizações dependiam exclusivamente das pessoas envolvidas nesse processo. Atualmente, os sistemas de informação possuem recursos que possibilitam a troca de dados de forma automática por meio da Internet.

O conceito de B2B pode ser definido como sendo a substituição dos processos manuais que envolvem as transações comerciais por processos eletrônicos[8]

Durante muitos anos as empresas utilizaram o conceito do Intercâmbio Eletrônico de Dados (EDI) [9]. Entretanto, a implementação de soluções EDI não é algo simples, pois existem algumas situações que devem ser mapeadas, como por exemplo, incompatibilidade com os sistemas existentes, em que as empresas possuem ambientes totalmente diferentes com sistemas operacionais, banco de dados e sistemas de ERP distintos. Além disso, a segurança deve ser criteriosamente bem definida, pois apenas os dados que a empresa deseja compartilhar podem ser acessados, e somente por instituições que tenham autorização para acessá-los.

Devido aos grandes desafios encontrados no processo de integração entre aplicações, nasceram alguns padrões que podem ser adotados pelas companhias [10]. Como já mencionado, o presente trabalho faz uso de Web Service para prover essa integração.

2.9 XML - EXtensible Markup Language

A sigla XML corresponde à *Extensible Markup Language* e, segundo [15], XML é um arquivo de texto simples e flexível, planejado para responder aos desafios de publicação eletrônica em larga escala. O presente trabalho faz uso de arquivos XML para solicitar, transportar e receber os dados dos diversos supermercados participantes no modelo da infraestrutura do sistema.

XML é basicamente um arquivo texto, assim como arquivos com extensão .txt ou .doc do Microsoft Word, porém com algumas particularidades que o tornam especial, como por exemplo, ser um padrão não proprietário e multi-plataforma, portanto esse tipo de arquivo pode ser utilizado por qualquer aplicação em qualquer plataforma.

O documento XML é formado por *tags*. Um elemento XML pode ter dados declarados como sendo preços de venda, saldo em estoque, um título de livro ou qualquer outro tipo de elemento de dado [11].

Arquivos XML suportam a criptografia dos dados, o que aumenta o grau de segurança entre o intercâmbio de dados.

2.10 WEB SERVICE

O principal uso da Internet é para acesso interativo a documentos e aplicações. Na maioria dos casos, esses acessos são feitos por pessoas, tipicamente trabalhando por meio do navegador da Internet. A Web pode ser potencializada se for estendida para suportar comunicação entre aplicações, ou seja, iteração diretamente de um programa para o outro sem necessariamente a intervenção do usuário [12].

Um Web Service pode ser definido como qualquer serviço que esteja disponível na Internet, usa um padrão de troca de mensagem por arquivos XML, e não está limitado ao sistema operacional ou linguagem de programação.[12]

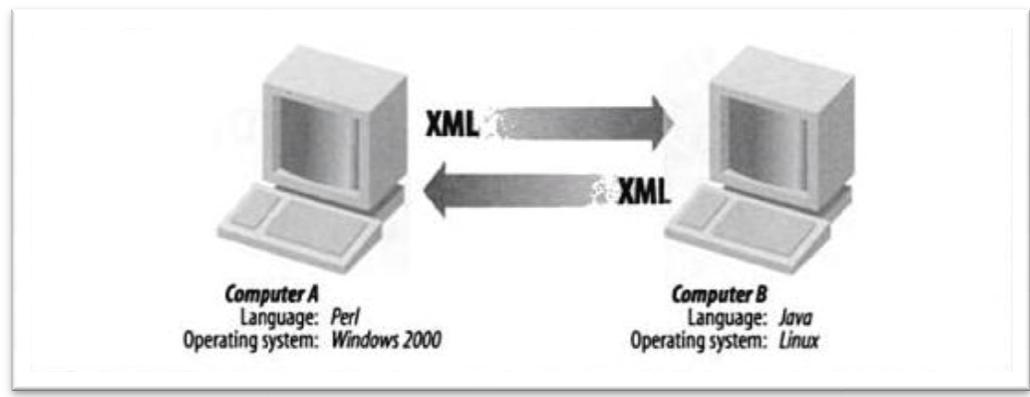


Figura 1: Funcionamento básico de Web Services. Fonte [12]

A Figura 1 ilustra o funcionamento de um Web Service básico. A transação se resume ao envio de arquivo XML entre computadores. O emissor deve enviar o arquivo seguindo um padrão estabelecido para ser corretamente interpretado pelo receptor.

Web Service é constituído basicamente de duas entidades fundamentais, o consumidor do serviço, que é o emissor da mensagem, e o provedor ou fornecedor do serviço, que é o receptor da mensagem.

Como consumidor do serviço comprehende qualquer módulo de software que precise consumir um serviço na Web, o mesmo deve localizar os serviços disponíveis e solicita uma requisição nos padrões estabelecidos pelo provedor do serviço, ou seja, utilização da interface especificada no WSDL. O provedor do serviço, por sua vez, recebe a requisição do cliente, processa e variavelmente pode enviar uma resposta com o resultado do serviço consumido.

Usualmente o processo de solicitar uma requisição ao provedor de serviço é chamado de consumir um serviço. Por exemplo, uma empresa fictícia A é especialista em

desenvolvimento de software para o ramo de transportadoras, a equipe da empresa A não possui especialistas na área tributária, então a empresa B, especialista na implementação de softwares tributários, fornece um serviço para cálculo dos impostos de uma determinada nota fiscal. Como a legislação tributária está sempre mudando, a empresa A pode consumir o serviço oferecido pela empresa B, dessa forma, toda a implementação referente ao cálculo de impostos fica sob responsabilidade da empresa especialista nesse segmento, nesse exemplo a empresa B. Sendo assim, o software da empresa A é o consumidor do serviço, sendo responsável por enviar requisições do serviço, e o software da empresa B, responsável por processar e responder às requisições, é o provedor ou fornecedor do serviço. Pode-se dizer então que o software da empresa A consome o serviço do software da empresa B.

O presente trabalho utiliza Web Services para prover serviços de consulta de preços dos supermercados ao site que consome esse serviço.

2.10.1 SOAP – Simple Object Access Protocol

SOAP é a sigla dada ao protocolo simples de acesso a objetos. Segundo Eric Newcomer, “SOAP é um caminho pelo qual as mensagens são enviadas de um computador para outro, e também disponível para processamento para o computador alvo” [13].

O protocolo SOAP é aberto, por isso, foi adotado por diversas empresas de hardware e software, o que contribuiu para sua popularidade e interoperabilidade. Portanto, o SOAP pode ser implementado em qualquer hardware ou sistema operacional. Devido à flexibilidade e à interoperabilidade alcançada com o SOAP, o mesmo tem sido escolhido como o caminho para utilização em Web Services, ou seja, é seguindo o protocolo do SOAP que as mensagens são enviadas e recebidas.

O SOAP será utilizado neste trabalho como o protocolo de comunicação entre as aplicações dos diversos supermercados. Mesmo que a implementação real de tal aplicação seja inviável no momento, o trabalho será implementado com SOAP para garantir conformidade com as boas práticas em desenvolvimento de Web Service.

3 TRABALHOS RELACIONADOS

A crescente demanda por integração entre as organizações tem gerado uma nova onda de pesquisas relacionadas às tecnologias e conceitos envolvendo tais integrações, itens como segurança, performance e escalabilidade são alguns dos temas frequentemente abordados sobre integração de dados. Neste capítulo, são apresentados alguns trabalhos correlacionados e complementares ao presente trabalho.

3.1 INTERCÂMBIO ELETRÔNICO DE DADOS

Historicamente, o avanço tecnológico se dava principalmente para atender requisitos de guerra. Nota-se que, embora ainda existam guerras e muitas tecnologias sejam desenvolvidas para esse fim, outra motivação tem impulsionado o avanço tecnológico, essa motivação é atender as necessidades de negócios em um mundo cada vez mais globalizado.

O intercâmbio eletrônico de dados, caracterizado pela troca de informações por meio digital entre organizações, foi abordado por [9], no qual o autor lista uma série de benefícios ao mundo corporativo expostos por tal integração.

Dentre os benefícios citados, estão:

- Adição de valor ao negócio, pois o mesmo viabiliza o acesso a novos mercados, gera vantagens sobre os concorrentes, contribui significativamente com a área de marketing, acelerando o lançamento de novos produtos.
- Aumento da produtividade da organização automatizando tarefas manuais, além de reduzir o tempo de processamento e certo grau de independência de atividade humana, diminuindo o risco de erros no processo, garantindo eficiência à organização.
- Traz vantagens financeiras, reduzindo e podendo até mesmo eliminar gastos com impressão, papel, armazenamento de documentos e estoque. Além de reduzir custos com funcionários e manutenção de equipamentos.

Embora os benefícios alcançados com o uso do EDI sejam plausíveis, existem diversos obstáculos a serem superados, tais como: falta de preparo de funcionários, barreiras culturais, incompatibilidade com os sistemas existentes nas organizações.

Pode-se constatar que existem diversas razões para que os supermercados brasileiros começem a pensar sobre a implementação de recursos para o intercâmbio eletrônico de dados.

Se esses já tivessem mecanismos de integração com outras aplicações, soluções como a proposta neste trabalho já estariam disponíveis para a sociedade.

3.2 DESAFIOS NA CONSTRUÇÃO E VALIDAÇÃO DA ROBUSTEZ DE APLICAÇÕES ORIENTADAS A SERVIÇOS

Alternativamente ao modelo de integração EDI, tem-se o Web Service, um novo paradigma de software orientado a serviços que tem mostrado ser uma boa opção para integração, sendo que este absorve os benefícios do EDI e resolve algumas das dificuldades daquele.

Em um artigo publicado em [16], encontram-se alguns dos desafios para construção e validação da robustez dos serviços disponíveis na Web. O autor não aponta soluções concretas para os problemas, mas os mapeia para que possam ser analisados quando um Web Service for implementado.

Os serviços oferecidos na Web dispõem de alto grau de encapsulamento, pois os clientes que consomem tal serviço conhecem apenas sua interface, mas não se sabe como o mesmo foi desenvolvido, e por consequência, não se sabe o quanto confiável será o serviço oferecido, dessa forma, a robustez do serviço deve ser testada.

Para que uma aplicação possa ser considerada robusta, é necessário que a mesma tenha tolerância a falhas, entretanto, quando se integra um Web Service a uma aplicação cliente, essa integração é dinâmica, e a implementação pode evoluir durante a execução, ou seja, o desenvolvedor do Web Service pode mudar o código do serviço que disponibiliza, e isso pode afetar os clientes que os consomem.

Por outro lado, o cliente que consome um determinado serviço pode desejar testar sua robustez, porém, esses testes não podem interferir na disponibilidade do serviço, pois o mesmo pode estar sendo consumido por outros clientes, e um teste de stress ou injeção de falhas pode gerar a indisponibilidade do serviço para os demais consumidores.

Outro fator importante é sobre qual serviço escolher. Para escolher dentre os diversos fornecedores do mesmo serviço na Web, é preciso avaliar os riscos de uso desse serviço. Adicionalmente, o desempenho do serviço deve ser avaliado, pois a velocidade de resposta pode não ser a esperada ou necessária para o projeto que o utiliza.

Os desafios expostos por [16] mostram-se pertinentes a este trabalho. Embora toda implementação esteja em um ambiente fictício, pode se estimar os problemas de sua implementação em um ambiente de produção.

3.3 ARQUITETURA DE SEGURANÇA EM APLICAÇÕES BASEADAS EM WEB SERVICE

Uma das grandes vantagens do Web Service é a facilidade de comunicação entre as aplicações, pois o mesmo não requer grande infraestrutura, sendo que para acessá-lo é utilizado o protocolo HTTP o qual é o mesmo utilizado pelas páginas na Web. Entretanto, esse benefício traz alguns problemas com relação à segurança. O autor de [17] mostra algumas das vulnerabilidades a que os Web Services estão expostos, são elas:

- Controle de acesso - os Web Services foram concebidos para serem consumidos diretamente por aplicações e não por usuários finais, entretanto muitos Web Services fornecem seus serviços para um público restrito e, para isso, as mensagens SOAP devem conter dados sobre o usuário e senha, o desafio está em como manter as credenciais de acesso de forma segura nessas mensagens, sendo que as mensagens SOAP são legíveis por humanos.
- Manutenção da segurança enquanto roteado entre múltiplos Web Services – em alguns casos, para que uma requisição seja completamente atendida, a mensagem SOAP passa por diversos Web Services intermediários, porém, por questões de confidencialidade, apenas o Web Service final pode ter acesso às informações contidas nas mensagens.

O autor de [17] propõe a segurança nas diversas camadas que envolvem a comunicação, na camada de rede, sugerindo as seguintes diretrizes:

- Configuração de firewall¹ para controlar o acesso partindo de determinados computadores.
- VPN, o qual permite estabelecer uma rede de clientes confiáveis que têm acesso aos métodos dos Web Services.

Na camada de transporte, o autor sugere:

- Certificados digitais que por meio de chaves de criptografia oferecem autenticidade aos clientes.
- PKI, que consiste em uma implementação de criptografia assimétrica.
- SSL é um protocolo que permite o tráfego criptografado de dados.

Referente à camada de aplicação:

¹ Firewall é um dispositivo de hardware ou software que examina as informações que chegam da *internet* e que são enviadas a ela, dessa forma, trancando as portas do computador contra entradas não autorizadas.

Pode-se aplicar segurança nativamente em sistemas operacionais por meio de privilégios de uma conta ou grupo de usuários em relação a um Web Service ou seu diretório.

- Controle de IP e DNS - opcionalmente pode-se configurar restrições para permitir ou negar acesso à um Web Service proveniente de um endereço de IP específico e de nomes de DNS,
- XML – Um bom mecanismo de segurança é aplicar criptografia e assinaturas digitais nos arquivos XML.
- WS-Security - Trata-se de uma especificação que propõe um conjunto de extensões no cabeçalho do envelope SOAP para inserir dados relativos à segurança por meio do XML Signature e XML encryption. A vantagem em aplicar segurança nos arquivos XML é que o mesmo é um padrão aberto e consequentemente não apresenta problemas com relação à dependência de plataforma.

Embora não seja o objetivo deste trabalho explorar os critérios de segurança em transações utilizando Web Services, esta seção apresenta os possíveis desafios que a implementação em um ambiente real pode trazer. Com isso se pode constatar que diversos fatores influenciam na segurança da solução proposta nesta monografia.

3.4 PROJETO PARA EXECUÇÃO DE CONSULTAS DISTRIBUÍDAS USANDO WEB SERVICES

Daniel Poll, em sua monografia [18], propôs uma solução baseada em Web Service para execução de consultas SQL em diversas bases de dados de forma unificada. Nesse a aplicação solicita os dados por meio de chamadas ao Web Service, que, por sua vez, é o responsável em efetuar a consulta ao banco de dados e retornar os dados em XML para a aplicação cliente.

Foram utilizados como estudo de casos Organizações Institucionais Comunitárias do Rio Grande do Sul, cujas organizações são integrantes do Programa de Avaliação Institucional das Universidades.

A arquitetura proposta por [18] comprehende um catálogo de Web Service que contém a lista de Web Services conhecidos, quando uma aplicação consome um serviço, solicitando um conjunto de dados, como, por exemplo, solicitação dos cursos disponíveis em cada universidade por área.

A aplicação consumidora do serviço faz a chamada ao Web Service passando como parâmetro à área desejada, por exemplo, área de Exatas. O Catálogo de Web Services escolhe quais são os Web Services que atendem ao requisito da pesquisa.

Posteriormente, os dados dos Web Services escolhidos são enviados para o plano de execução, esse, por sua vez, é responsável por invocar os métodos necessários para atender à consulta solicitada pela aplicação cliente, que, nesse exemplo, é a lista de cursos de Exatas disponíveis nas universidades.

O plano de execução também é responsável por definir a ordem das chamadas dos Web Services, sendo que, para atender a essa consulta, é necessário consumir diversos Web Services, ou seja, um Web Service de cada universidade que retorne a seus respectivos cursos.

Objetivando aumentar a *performance* do processo, as chamadas são realizadas paralelamente com uso de Threads², dessa forma, é possível fazer todas as chamadas sem que seja necessário aguardar a resposta de um Web Service para invocar o próximo.

O projeto proposto em [18] mostrou que o uso de Web Service atendeu satisfatoriamente às expectativas, possibilitando um baixo acoplamento entre as aplicações e aceitável velocidade nas transações.

3.5 WEB SERVICES COMERCIAIS

Esta seção tem o intuito de apresentar serviços Web disponíveis para serem consumidos por aplicações comerciais, objetivando não um estudo detalhado de cada serviço, mas, sim, apenas destacar a importância, popularidade e o poder de Web Services.

3.5.1 Correios

Há pouco tempo os correios disponibilizavam gratuitamente um Web Service para consulta de CEP, no qual os sistemas informavam o CEP e recebiam o endereço correspondente, e vice e versa. Entretanto, os correios substituíram esse serviço por uma base de dados paga, atualmente essa base de dados está disponível nos formatos txt e mdb.

A indisponibilidade desse serviço pegou de surpresa desenvolvedores que o consumiam, consequentemente diversos sistemas pararam de buscar os endereços. Como muitas aplicações já faziam uso desse serviço, os desenvolvedores encontraram outro meio de continuar oferecendo aos seus clientes tal recurso, e atualmente muitos aplicativos consomem gratuitamente o serviço de consulta de CEP do seguinte Web Service

² Threads são fluxos dentro de processos que podem ser executadas paralelamente.

<http://www.bronzebusiness.com.br/Webservices/wscep.asmx>, embora esse serviço já seja de grande valia, não se tem garantia de que os dados são atualizados.

Entretanto, o correio ainda oferece um serviço para cálculo de frete de SEDEX, e-SEDEX e PAC através de Web Service , com isso, é possível implementar em um site de e-commerce o cálculo do valor do frete, de forma precisa e fácil[25].

3.5.2 Xignite

O portal xignite.com oferece uma ampla gama de Web Service financeiros, dentre eles existem serviços para consultar o câmbio entre moedas, consulta do preço do ouro e prata, valores de ações na bolsa de valores dentre outros [26].

Xignite iniciou suas atividades em 2003, fornecendo informações críticas de negócios, ele é pioneiro em oferecer Web Services que disponibilizam dados econômicos, seus Web Services são consumidos por aplicações em todo o mundo, e por empresas de grandes portes.

Os serviços disponíveis não são gratuitos, é possível testá-los por meio de um Web Service *Trial*, seu catálogo de Web Services conta atualmente com 54 serviços oferecidos ao mercado financeiro e econômico.

3.5.3 Yahoo Web Service

A Yahoo oferece diversos Web Services que podem ser utilizados gratuitamente pelos desenvolvedores. Serviços como gerenciamento de e-mails da Yahoo permite que desenvolvedores criem clientes para enviar, receber, apagar e encaminhar e-mails em sua própria aplicação.

Outro recurso poderoso é o Search Web Service, que permite implementar busca com o *engine* de busca da Yahoo através de Web Service, dessa forma é possível desenvolver um mecanismo de pesquisa utilizando a infraestrutura da Yahoo, sendo que para isso, basta utilizar o Web Service oferecido por eles [27].

3.5.4 Master Card Web Services.

O poder dos Web Services tornou-se evidente quando grandes empresas de transações financeiras o adotaram e disponibilizaram serviços sob sua arquitetura.

A MasterCard utiliza Web Service para disponibilizar aos sites de e-commerce serviços para validação de cartões de crédito e confirmação de pagamento [28].

3.5.5 Mapa de Endereço IP.

Esse Web Service fornece um mecanismo para identificar o país de origem de um determinado endereço IP, dessa forma, a aplicação que consome esse serviço envia o endereço IP e tem como retorno o país desse endereço, algo útil para implementar segurança em sistemas, por exemplo, pode-se restringir acesso de determinado país a um sistema [30].

3.5.6 Condições do Clima em Cidades dos Estados Unidos

Destinado a retornar várias informações com relação ao clima de uma determinada cidade ou código postal.

Esse serviço traz informações como latitude e longitude do local, temperatura máxima, temperatura mínima, tanto em graus celsius como fahrenheit. Essas previsões são retornadas com dados de uma semana, podendo dessa forma implementar em um sistema de logística, por exemplo, o melhor dia para fazer uma entrega em determinado local [32].

3.5.7 Mapa de Endereços

Esse Web Service, atualmente, funciona somente para endereços nos Estados Unidos. O serviço oferecido é retornar um mapa e as instruções para chegar a determinado destino, passando como parâmetro a localização de origem, destino e o país. O mesmo calcula a rota e retorna uma imagem do mapa com a rota traçada, e também informa as instruções em forma de texto de como chegar no local [31].

3.5.8 Geração de Código de Barras

Esse serviço por sua vez, fornece um mecanismo para gerar código de barras em diversos padrões, em que na chamada do serviço envia-se o código numérico do qual quer se gerar o código de barras, e também alguns parâmetros como a altura e largura do código de

barras, cor e nome da fonte. Como resposta, obtém-se uma imagem com o código de barras gerado [29].

3.5.9 WEB SMS

Trata-se de um Web Service americano para envio de mensagens de texto para celulares, esse recurso é bastante útil para área de marketing das empresas, no qual o sistema que consome esse Web Service pode enviar SMS para um ou vários telefones celulares de uma única vez.

No Brasil, pode-se utilizá-lo pela rede da TIM, entretanto esse serviço não é gratuito [33].

3.6 ANÁLISE DE PROJETOS COMERCIAIS SIMILARES

Esta seção apresenta dois sites que oferecem serviços similares ao proposto neste projeto. A similaridade limita-se apenas no quesito comercial, sendo que na implementação técnica são totalmente distintos.

O objetivo deste capítulo é mostrar os pontos fortes e fracos de cada site, e analogamente apresentar as soluções propostas neste trabalho para suprir as brechas identificadas nesses projetos.

Foram encontrados apenas dois sites que têm objetivos similares ao proposto neste trabalho, são eles: Mercado Mineiro[34] e Mercado do Preço[35].

A tabela 1 mostra algumas características dos dois sistemas e também do presente trabalho.

Tabela 1: Comparação projetos similares

Site	Listas de Compras	Tempo de Resposta	Usabilidade	Preço On-line	Escolha de Mercados
Mercado mineiro	Não	6 segundos	Média	Não	Não
Mercado do preço	Sim	7 segundos	Boa	Não	Não

Solução Proposta	Sim	3 segundos	Boa	Sim	Sim
------------------	-----	------------	-----	-----	-----

Os seguintes itens compõem a Tabela 1.

- **Lista de Compras:** Esse quesito refere-se à possibilidade de os usuários criarem suas listas de compras para efetuar as comparações de preços.
- **Performance:** Esse quesito refere-se à velocidade de resposta do sistema.
- **Usabilidade:** Nesse contexto, usabilidade refere-se à fácil navegação, esse item procura apresentar o quanto intuitivo é o sistema.
- **Preço *on-line*:** O item preço on-line se refere à exatidão dos preços, de forma que os mesmos correspondam ao preço atualizado da mercadoria nos estabelecimentos selecionados no momento da consulta.
- **Escolha de mercados:** Esse item refere-se à possibilidade de os usuários escolherem os supermercados que farão parte de sua pesquisa, dessa forma delimitando a resposta apenas com os supermercados de interesse do usuário.

O site Mercado Mineiro apresenta o resultado da comparação de preços de forma bastante amigável.

Inicialmente o sistema mostra a mercadoria escolhida e seu respectivo menor preço, preço médio, maior preço e o percentual de variação. Esse recurso mostra-se bastante interessante, pois o consumidor pode identificar facilmente o quanto estará economizando ao praticar essa consulta e, ao clicar em “ver preços detalhados”, é apresentada uma tela com os preços em seus respectivos supermercados.

A pesquisa das mercadorias pode ser efetuada escolhendo a seção da mesma, por exemplo: ao selecionar a seção biscoitos, todos os biscoitos cadastrados no sistema serão consultados, entretanto os preços já estão previamente cadastrados no banco de dados do site, com isso, não se tem o valor real da mercadoria, sendo que a mesma pode ter seu preço alterado desde a última atualização, e essa atualização de preços é feita de maneira manual por agentes contratados pela empresa desenvolvedora do sistema, o que expõe o processo a erros, caso uma mercadoria tenha seu valor digitado ou coletado de forma errada, a pesquisa não é eficiente.

O site Mercado do Preço apresenta um *design* mais atraente, entretanto o resultado da comparação não permite que o usuário visualize o melhor preço de forma rápida, porém, após escolher a mercadoria, é possível adicioná-la a lista de compra para posterior impressão, além de permitir que sejam criadas listas de compras, facilitando posteriores consultas.

Comparando as soluções existentes com a solução proposta (Clube da Compra) neste trabalho, fica evidente que, o presente trabalho propicia uma melhor confiabilidade dos dados, sendo que não é necessária a intervenção humana na captura dos preços, o que diminui a possibilidade de erros.

Além disso, o preço será consultado diretamente no banco de dados do supermercado, garantindo que os preços não estejam desatualizados, a possibilidade de os preços estarem desatualizados nos sistemas existentes já é um fator suficiente para causar uma barreira em sua adoção.

Outro recurso bastante interessante do Clube da Compra é a possibilidade de escolher em quais supermercados os preços serão consultados, de forma que, muitas vezes é inútil saber o preço de item em um estabelecimento muito longe ou de difícil acesso. Dessa forma, os usuários do Clube da Compra podem pesquisar os preços das mercadorias que desejam nos supermercados de sua preferência e com confiabilidade dos preços.

4 ESPECIFICAÇÃO

Neste capítulo, é apresentada a arquitetura do projeto, mostrando primeiramente uma visão geral do mesmo e posteriormente detalhando cada camada e recurso que estarão disponíveis nos sistemas.

Para tornar o projeto factível, faz-se necessária a criação dos seguintes módulos:

Módulo I: Refere-se ao sistema de cadastro de mercadorias, isto é, uma *interface* para cadastrar os preços das mercadorias dos supermercados simulando o sistema dos mesmos, onde suas respectivas mercadorias são cadastradas. Em um cenário real, esse sistema seria o ERP utilizado no estabelecimento, tal como SAP, Microsiga Proteus ou qualquer outro sistema utilizado pelos supermercados. Neste trabalho, esse sistema é chamado pela sigla SCM(Sistema de Cadastro de Mercadorias).

Módulo II: Refere-se ao Web *Service* provedor o qual consiste em construir um sistema que fornecerá acesso aos dados cadastrados no sistema SCM por meio de Web Service, ou seja, será o Web Service provedor do serviço de consulta dos preços das mercadorias. Neste trabalho, este sistema é chamado pela sigla WSP. Em um cenário real, cada supermercado teria seu próprio WSP responsável por receber, processar e responder às requisições dos sistemas que o consomem, aqui será o sistema denominado Clube da Compra(CDC).

Módulo III: Refere-se ao Web site para pesquisa de preços, esse projeto consiste em um sistema Web, no qual os usuários podem submeter suas listas de compras à pesquisa de preços. Esse sistema denominado Clube da Compra(CDC) é a essência deste trabalho, pois é por meio dele que serão consumidos os serviços fornecidos pelo WSPs. O sistema CDC será desenvolvido na plataforma .NET com a linguagem de programação C# e ASP.NET utilizando banco de dados SQLServer.

Esse conjunto de aplicações forma toda a solução proposta, no qual o módulo I, o sistema SCM, e o módulo II o Web Service WSP, são as aplicações que os supermercados devem ter, e serão desenvolvidas neste trabalho para tornar o módulo III passível de utilização, pois sem esses módulos não se teria uma solução funcional.

4.1 VISÃO GERAL

Como já mencionado, o objetivo deste projeto é construir uma ferramenta para pesquisar os preços de mercadorias em supermercados.

Para alcançar tal fim, diversas etapas são necessárias como, por exemplo, o usuário deve criar uma lista de mercadorias e também deve escolher em quais supermercados os preços serão pesquisados.

O funcionamento básico da solução pode ser visto na Figura 2 e é descrito da seguinte maneira: um determinado usuário acessa por meio da Internet o site do Clube da Compra, efetua seu *logon* e submete sua lista de compra à pesquisa de preços. O sistema do Clube da Compra verifica os supermercados e as mercadorias da lista de compra no banco de dados hospedado no Web Server, e então envia uma mensagem SOAP de requisição dos preços para cada um desses supermercados.

Essa mensagem em formato XML contém as mercadorias que o usuário deseja pesquisar, o método que será chamado para fazer a pesquisa e também o destinatário da mensagem. Cada supermercado tem um Web Service que irá receber essa mensagem, processá-la e respondê-la.

A resposta consiste em uma mensagem SOAP com os preços das mercadorias solicitadas em XML.

O sistema Clube da Compra recebe a resposta de cada supermercado, e então as processa verificando e indicando onde estão os menores preços das mercadorias. Essa comparação é feita no Web Server que, após concluir todo processamento, envia uma página com o resultado da pesquisa para o usuário.

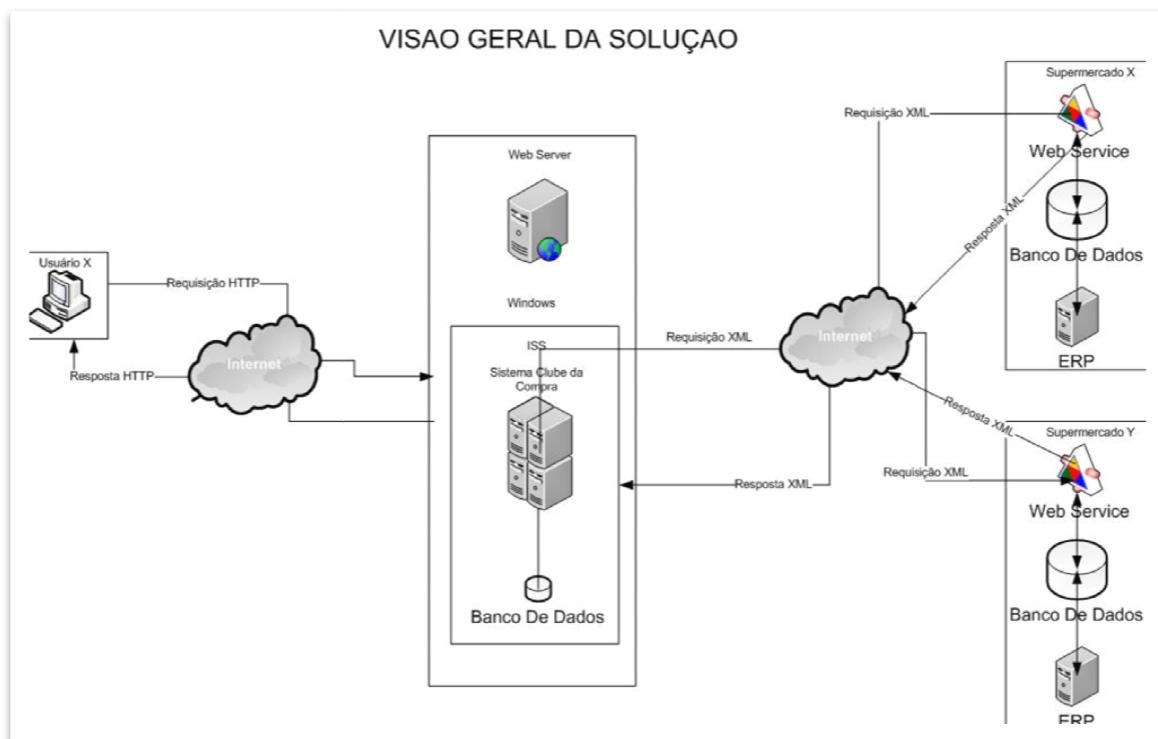


Figura 2: Visão Geral da Solução.

4.2 ANÁLISE DA SOLUÇÃO

Esta seção apresenta como se deu a análise da solução proposta, e também a documentação criada como suporte à análise do mesmo. Essa documentação consiste em diagramas e especificações que nortearam os requisitos funcionais e não funcionais da solução.

A primeira etapa da análise foi levantar o que o sistema buscava atender, ou seja, definir os limites do projeto de forma a direcionar os esforços para solução das regras de negócios, independentemente da tecnologia que seria empregada. Posteriormente, com base nos requisitos funcionais, foi desenvolvida uma análise orientada a objetos, buscando identificar as entidades e operações necessárias para atender aos requisitos funcionais, isto é, foram definidas as classes e os métodos necessários.

Após a definição das entidades e operações, passou-se por um refinamento buscando empregar padrões de projetos para separação lógica entre as camadas, separando as classes de acordo com suas responsabilidades.

Em seguida, foram identificados os dados que deveriam ser armazenados em banco de dados, dessa forma, passando para a análise do banco de dados criando diagramas e aplicando padrões de normalização.

4.2.1 Lista de Requisitos Funcionais.

A identificação das necessidades da solução foi descrita em uma lista com os requisitos funcionais a que o projeto se propunha contemplar, essa lista foi gerada com base nos desejos do autor e em conversas informais sobre o assunto com amigos e parentes. Essa lista define o que o sistema deve ter, e não como esses itens devem ser implementados.

Seguem os requisitos que alicerçam o projeto:

- O sistema deve permitir ao usuário efetuar a busca para comparação de preço de mercadorias, mesmo nos quais o usuário não esteja cadastrado no sistema. Entretanto, para usuários não cadastrados não será possível fazer uma lista de compras.
- O usuário não cadastrado deve efetuar a busca pela cidade do supermercado e grupo da mercadoria.
- O usuário pode se cadastrar no sistema, informando um *login* e senha.

- O usuário cadastrado deve escolher os supermercados nos quais ele têm interesse em pesquisar seus respectivos preços. Para localizar os supermercados disponíveis poderá ser feita uma pesquisa por CNPJ e cidade.
- Usuários cadastrados podem criar ilimitada quantidade de lista de compras, em que cada lista possui as mercadorias que ele deseja comparar os preços.
- Usuários cadastrados podem trocar suas senhas e desativar seu cadastro.
- As listas de compras cadastradas podem ser alteradas pelo usuário, podendo adicionar, alterar a quantidade ou remover itens da lista.
- O sistema deve ter uma área para cadastrar anunciantes.
- Os anunciantes poderão adicionar, alterar e remover anúncios, os anúncios deverão ter uma data de ativação e data de inativação e ser previamente aprovado pelo gerente do sistema.
- O anúncio deve ter a descrição do mesmo, imagem, e seu respectivo preço, assim como data de validade da promoção caso haja.
- O sistema deve gerar uma lista com as mercadorias mais baratas em cada supermercado, dessa forma o usuário pode visualizar em quais supermercados as mercadorias tem o menor preço.
- O sistema deve ter a possibilidade de imprimir as listas, com seus respectivos preços.
- O anúncio deve mostrar somente o nome da mercadoria e o preço; quando o usuário clicar no anúncio, o mesmo é encaminhado para o site do anunciante.
- O Valor da campanha publicitária será variável, isso é, a cada clique no anúncio, uma taxa será cobrada.
- O anunciante deve ter uma área para acompanhar seus anúncios.
- O anunciante pode definir um valor máximo de cliques, após esse valor o anúncio é desativado.
- O usuário deve ter a opção de sair do sistema encerrando a sessão aberta.

4.2.2 Diagrama de Caso de Uso.

A figura 3 apresenta o diagrama de caso de uso, no qual foi construído com base na lista de requisitos e especificação de caso de uso em anexo a este documento.

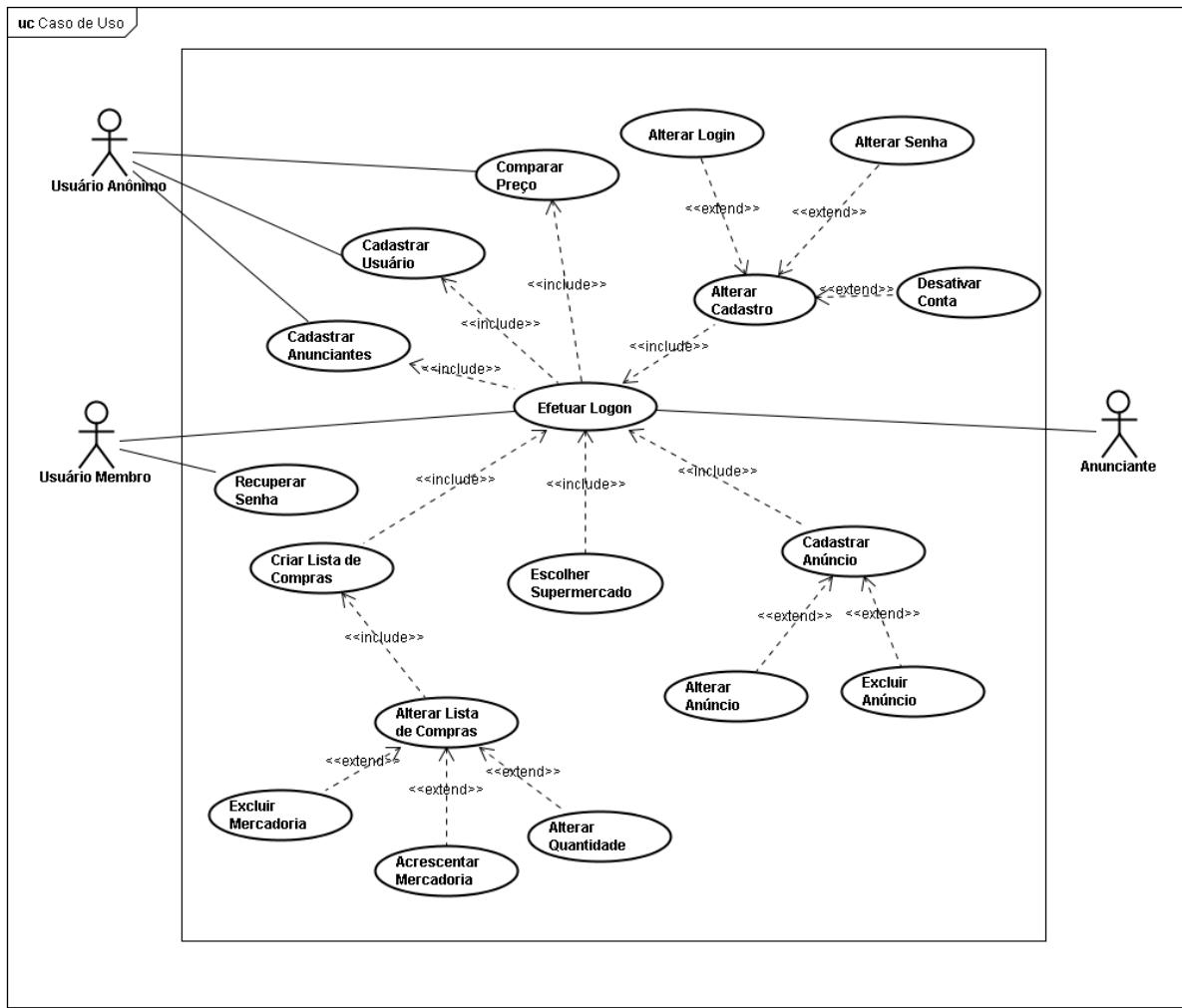


Figura 3: Diagrama de caso de uso.

Como pode ser observado no diagrama de caso de uso da Figura 3, o usuário membro pode efetuar um *login* no sistema. Após estar logado esse usuário tem a possibilidade de alterar seus dados cadastrais, chamado no sistema de perfil, o usuário pode ainda criar suas listas de compras, bem como alterá-las, o usuário pode ainda escolher em quais supermercados deseja que a pesquisa de preço seja efetuada. Já um usuário anônimo, ou seja, um usuário que não tem um cadastro no sistema, pode pesquisar preços, porém não tem a opção de criar uma lista nem mesmo escolher os supermercados em que a comparação será feita, cabe ainda ao usuário não cadastrado efetuar cadastro para usufruir de todos os benefícios da solução.

O usuário anunciante tem a opção de cadastrar anúncios assim como alterá-los, os anúncios cadastrados por esse usuário serão apresentados aleatoriamente aos usuários.

4.2.3 Estrutura do sistema

Em sistema orientado a objetos, sua estrutura é formada através de classes e relacionamentos. O presente projeto busca seguir as boas práticas em desenvolvimento orientado a objetos, por isso, sua arquitetura vai além de simplesmente desenvolver as classes que compõem o sistema, mas busca dividir as responsabilidades das classes aplicando o conceito de multicamadas já mencionado.

A fim de facilitar o entendimento, a figura 4 apresenta a estrutura do sistema, dessa forma, se pode ter uma visão sintética sobre o relacionamento entre as classes.

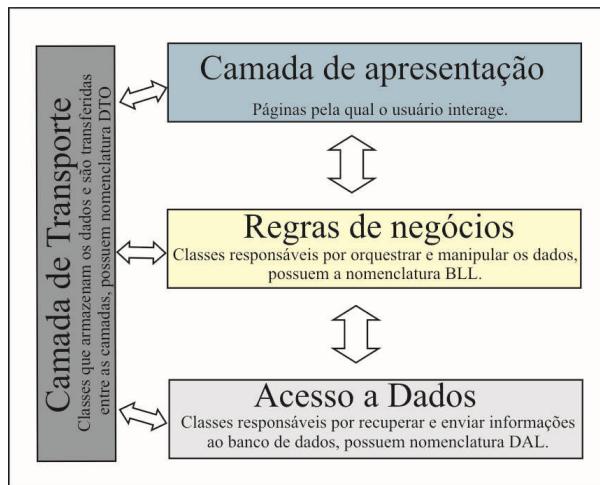


Figura 4: Estrutura das camadas

Como pode ser visto na Figura 4, na camada de apresentação estão as páginas pelas quais os usuários interagem com o sistema. Quando um usuário solicita alguma ação, uma classe da camada de transporte é instanciada, assim como uma classe da camada de negócios, o objeto da camada de transporte é preenchido com os dados que serão transportados para a classe de negócios manipular e, se necessário, transportar para a classe de acesso a dados, de forma à atender a solicitação do usuário.

A fim de facilitar a compreensão, a solução foi decomposta em vários diagramas de classes, separados por suas respectivas responsabilidades. A Figura 5 apresenta o diagrama de classes relacionado aos usuários do sistema.

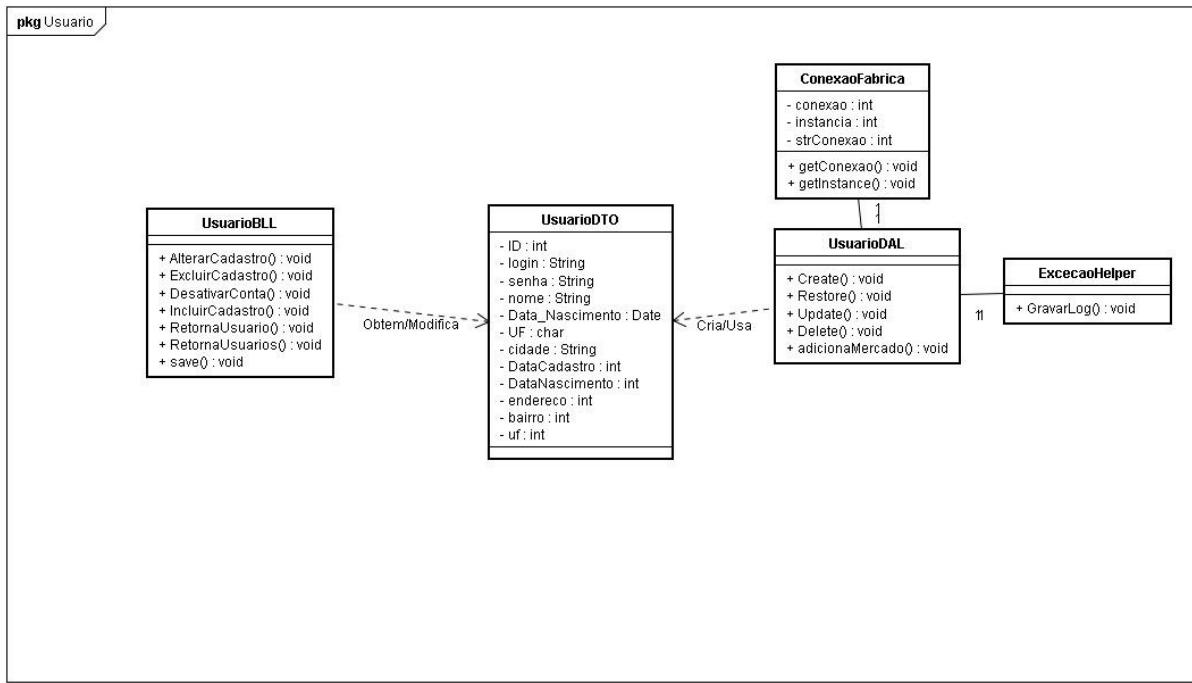


Figura 5: Diagrama de Classes de Usuários

Como pode ser verificado na Figura 5, o diagrama de classes foi modelado visando a um fraco acoplamento entre as classes e alta coesão, ou seja, cada classe é especialista em determinada solução do negócio. Dessa forma, tem-se uma melhor reusabilidade, e os possíveis problemas de codificação podem ser isolados agregando assim manutenibilidade à solução.

A descrição do diagrama de classes de usuários é o seguinte: a classe `UsuarioDTO` é uma classe que representa os dados dos usuário, essa classe é utilizada como um mecanismo de transporte dos dados de uma camada à outra, ela esta relacionada com a classe `UsuarioDAL`, responsável por persistir e recuperar os dados no banco de dados, alocar esses dados na classe `UsuarioDTO` que são transportados para a classe `UsuarioBLL`, o inverso é recíproco, no qual a classe `UsuarioBLL` envia para a classe `UsuarioDAL` um objeto do tipo `UsuarioDTO` para ser persistido no banco de dados.

A classe `ConexaoFabrica` é responsável por criar novas conexões ao banco de dados e retornar conexões do *spool* de conexões, esta classe está sempre relacionada às classes da camada DAL as quais tem acesso ao banco de dados.

A classe `ExcecaoHelper` é uma classe adjutora cuja função é capturar os erros que ocorrerem na camada de acesso a dados e gravar um arquivo de log possibilitando que o administrador do sistema tome as medidas necessárias para corrigir o problema.

A Figura 6 mostra o relacionamento entre as classes que compõem as regras de negócios pertinentes à manipulação das informações dos supermercados cadastrados na solução.

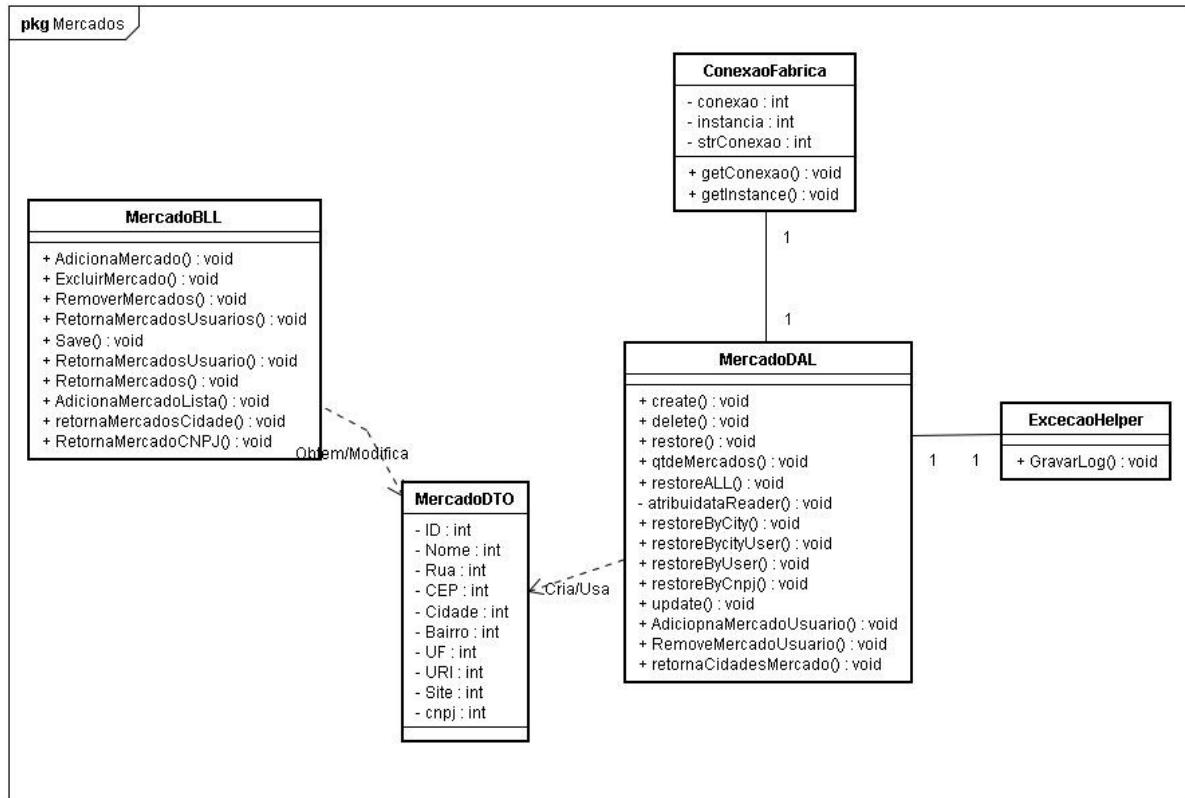


Figura 6:Diagrama de classes do mercados

O diagrama da Figura 6 pode ser descrito da seguinte forma: a classe MercadoBLL, a qual contém as regras de negócios pertinentes a manipulação dos dados dos mercados, relaciona se com a classe MercadoDAL por meio da classe MercadoDTO, a qual é transferida entre as camadas, neste diagrama a camada DAL, representada pela classe MercadoDAL, cria um objeto do tipo MercadoDTO e o transfere para a classe MercadoBLL que pode aplicar alguma regra de negócio e enviar alguma informação para a camada de apresentação que neste trabalho são as páginas com extensão “.aspx”.

A Figura 7 mostra o relacionamento entre as classes que compõem as regras de negócios pertinentes à manipulação das informações das mercadorias que serão cadastradas na solução.

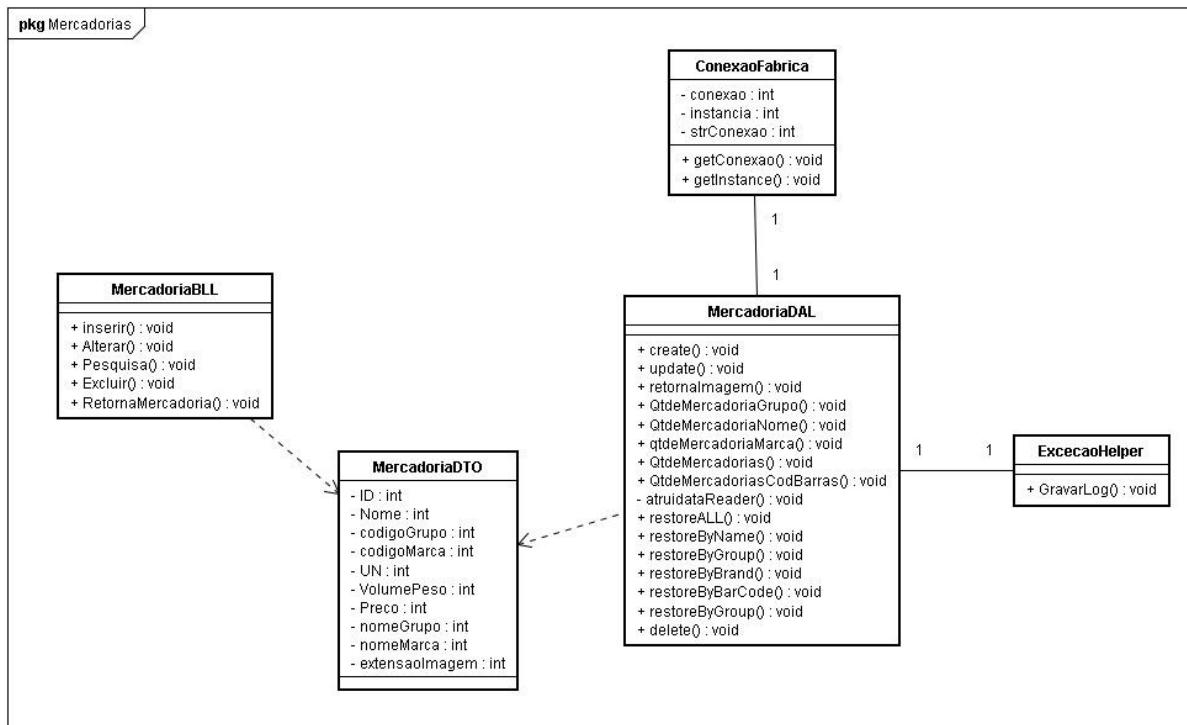


Figura 7: Diagrama de classes das mercadorias

O diagrama da Figura 7 segue o mesmo padrão dos demais, contendo uma classe referente à camada de negócios representada pela classe MercadoriaBLL, uma classe de transporte que neste diagrama é a classe MercadoriaDTO, além da classe MercadoriaDAL que faz acesso ao banco de dados, e as classes adjutorias como ConexaoFabrica e ExcecaoHelper.

A seguir é apresentado o diagrama de classes que compõem a área destinada aos anunciantes, essas classes são responsáveis pela inserção e manutenção dos anúncios que serão veiculados no sistema.

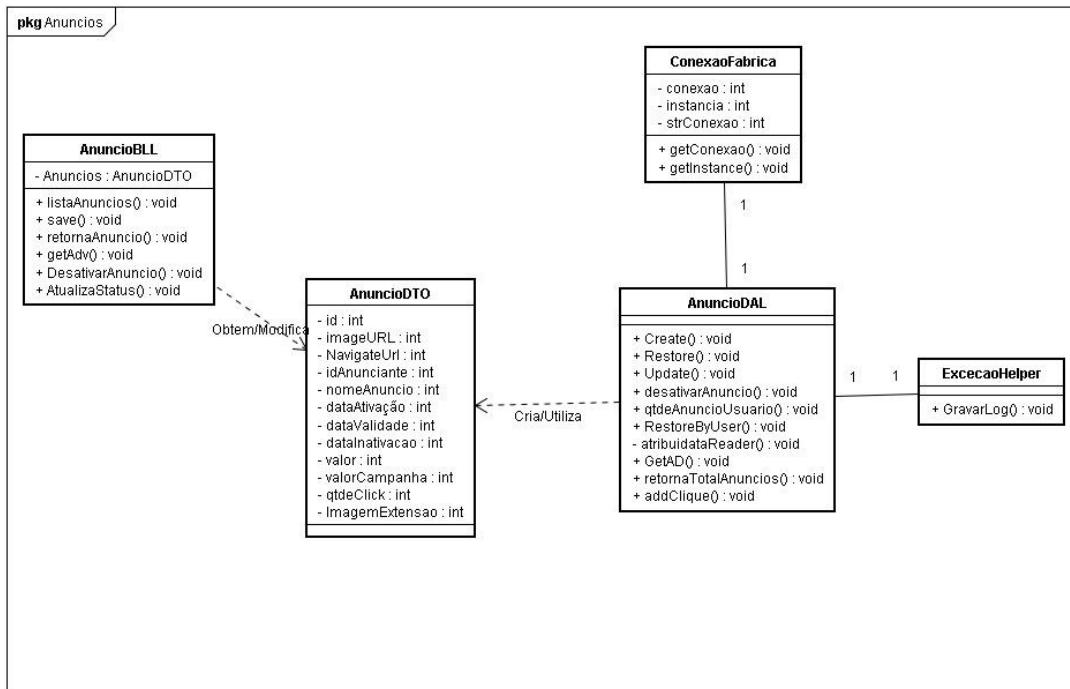


Figura 8: Diagrama de classes dos anúncios.

A Figura 9 Apresenta o diagrama de classes referentes às regras de negócios aplicadas nas rotinas de consumo dos Web Services.

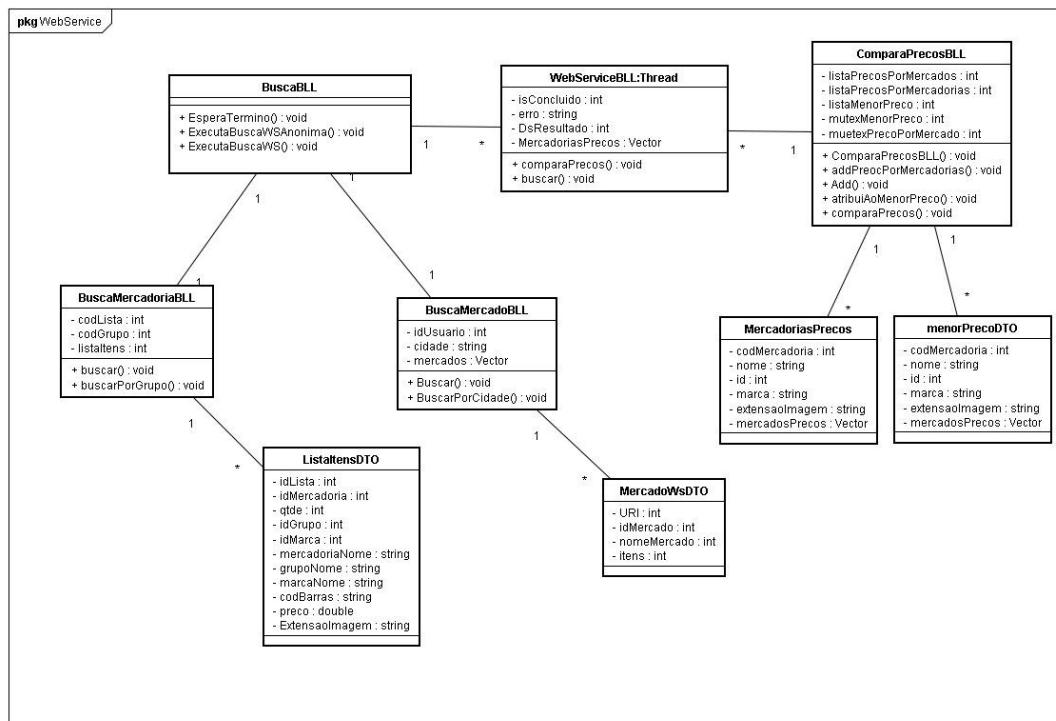


Figura 9: Diagrama de classes comparação de preços

As classes que compõem o diagrama da Figura 9 são responsáveis pelo núcleo da solução proposta, a classe BuscaBLL é responsável por orquestrar as chamadas aos Web Services dos supermercados além de buscar as informações necessárias para consumir o serviço, tais como, supermercados que devem ser consultados e quais mercadorias.

A classe WebServiceBLL, por sua vez, é responsável por fazer as requisições aos métodos dos WebServices dos supermercados, essas requisições são efetuadas paralelamente por meio do uso de threads.

A classe ListaItensDTO contém as mercadorias da lista do usuário, a qual será submetida à pesquisa, são os códigos de barras contidos nessa lista que serão enviados para os Web Services dos supermercados.

Os diagramas acima apresentados facilitam a compreensão do funcionamento e arquitetura deste projeto.

4.2.4 Banco de dados

A estrutura do banco de dados foi desenvolvida com base nos atributos identificados nos diagramas de casos de uso, com isso, foram definidas as tabelas, campos e relacionamentos, conforme apresentado na Figura 10.

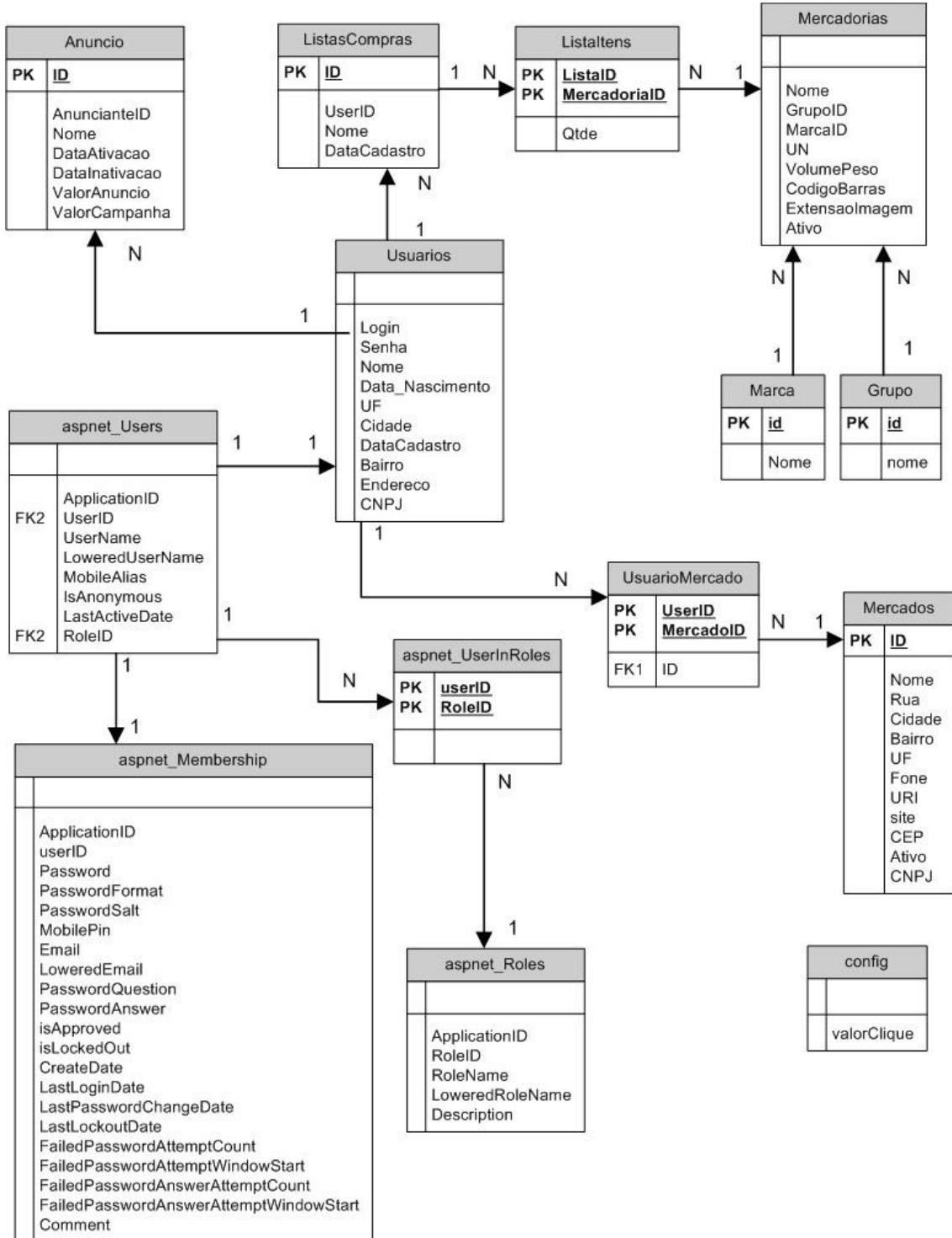


Figura 10: Diagrama de Entidade e Relacionamento

A tabela de Usuários armazena as informações sobre os usuários que estão registrados no sistema. Cada usuário pode ter inúmeras listas de compras. As listas de compras são formadas por itens que no DER acima apresentado são representados pela tabela ListaItens. Essa tabela se relaciona com a tabela de Mercadorias, a qual contém os dados de todas as

mercadorias cadastradas no sistema. Para que o sistema possa consultar os preços somente nos supermercados de interesse do usuário, faz se necessária a Tabela UsuarioMercado, a qual armazena os supermercados escolhidos pelo usuário para efetuar a comparação dos preços. Essa tabela está relacionada com a Tabela Mercados, a qual armazena os dados de todos os supermercados que oferecem o serviço de consulta de preço, ou seja, o provedor do Web Service.

4.3 IMPLEMENTAÇÃO DO CÓDIGO

Esta seção apresenta trechos de códigos que tiveram maior relevância no desenvolvimento do presente trabalho, assim como explicações sobre os mesmos.

O critério de avaliação referente a qual código tem a relevância acima citada, se dá pela importância, inovação, complexidade e singularidade do mesmo.

4.3.1 Autenticação de usuários

Como mencionado na seção 2.1.3 sobre *Application Services*, todo processo de autenticação e autorização de acesso do usuário foi implementado utilizando os recursos de segurança fornecidos pela plataforma .NET , com isso descarta-se a necessidade de codificação das regras que norteiam tal fim.

Para que o .NET possa se conectar com a base de dados, faz se necessário configurar uma string de conexão, conforme figura 11. Esta *string* será apontada para as demais configurações.

```

22<connectionStrings>
23  <clear/>
24  <add name="CDCStringConexao" connectionString="Data Source=RODRIGO-PC\SQLEXPRESS;Initial
25    Catalog=CDC;Integrated Security=True;Max Pool Size=60;
26    pooling=true"/>
27</connectionStrings>
```

Figura 11: Configuração da string de conexão.

A tag “*connectionString*”, declarada na linha 22, indica o início do bloco de configuração da string de conexão, isto é, o local onde se encontra o banco de dados.

A linha 24 contém o atributo *name*, onde é informado o nome dessa configuração , o qual será referenciado nas demais configurações da solução, um detalhe importante está na

linha 25, onde é especificado o *Pool Size* de 60 conexões. O pool size é um cachê de conexões com o banco de dados, com isso evita-se que seja criada uma nova conexão a cada acesso ao banco, pois o .NET guarda até 60 conexões prontas e, quando um usuário requisitar um acesso ao banco, antes de criar uma conexão, é verificado se não existe alguma conexão disponível no pool de conexões.

Outro aspecto importante é o de *roles*. Segue abaixo a imagem do arquivo web.config onde os roles são configurados.

```

36<roleManager enabled="true">
37  <providers>
38    <remove name="AspNetSqlRoleProvider"/>
39    <add name="AspNetSqlRoleProvider" connectionStringName="CDCStringConexao"
40      applicationName="ClubeCompra" type="System.Web.Security.SqlRoleProvider, System.Web, Version=2.0.0.0,
41      Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"/>
42  </providers>
43</roleManager>
```

Figura 12: Configuração de Roles

A linha 36 da figura 12 contém a tag “*roleManager*” com um atributo *enabled* apontado como *true*, isso indica que a aplicação irá fazer uso de *roles* para o gerenciamento de autorização. Na linha 39, é definido o nome dessa configuração e também o nome da configuração da string de conexão no atributo “*connectionStringName*”, com isto a plataforma sabe onde estão armazenadas as informações referentes aos roles da aplicação.

Para o gerenciar o processo de autenticação, é necessário efetuar as configurações referentes ao *Membership*. Estas configurações são apresentadas na figura 13.

```

46<membership defaultProvider="AspNetSqlMembershipProvider">
47  <providers>
48    <remove name="AspNetSqlMembershipProvider"/>
49    <add name="AspNetSqlMembershipProvider" type="System.Web.Security.SqlMembershipProvider, System.Web, Version=2.0.0.0,
50      Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a" connectionStringName="CDCStringConexao"
51      enablePasswordReset="true" requiresQuestionAndAnswer="false"
52      applicationName="ClubeCompra" requiresUniqueEmail="false"
53      passwordFormat="Hashed" maxInvalidPasswordAttempts="100"
54      minRequiredPasswordLength="5" minRequiredNonalphanumericCharacters="0"
55      passwordAttemptWindow="10" passwordStrengthRegularExpression="" />
56  </providers>
57</membership>
```

Figura 13: Configuração de Membership

A figura 13 apresenta a configuração referente aos recursos do *Membership* do arquivo de configuração web.config. O atributo “*enablePasswordReset*”, apontado como *true* na linha 51, indica que é possível que um usuário que esqueceu sua senha receba uma nova senha por e-mail. O atributo “*passwordFormat*”, apontado como “*Hashed*”, indica que a senha

deverá ser gravada em Hash no banco de dados, isto aumenta a segurança da aplicação, tendo em vista que a criptografia de Hash não pode ser decifrada.

Outro atributo importante é o “MinRequiredPassowrdLenght” da linha 55, este atributo define o tamanho mínimo que deve ser a senha do usuário, nessa solução a senha deve conter no mínimo cinco caracteres.

Ainda nas configurações da aplicação se tem o *profile*, conforme apresentado na figura na figura 14.

```

60<profile enabled="true" automaticSaveEnabled="false" defaultProvider="MyProfileProvider">
61    <providers>
62        <add name="MyProfileProvider"
63            connectionStringName="CDCStringConexao"
64            applicationName="ClubeCompra"
65            type="System.Web.Profile.SqlProfileProvider"/>
66    </providers>
67    <properties>
68        <add name="ProfileName" allowAnonymous="false"/>
69    </properties>
70</profile>
```

Figura 14: Configuração de profile

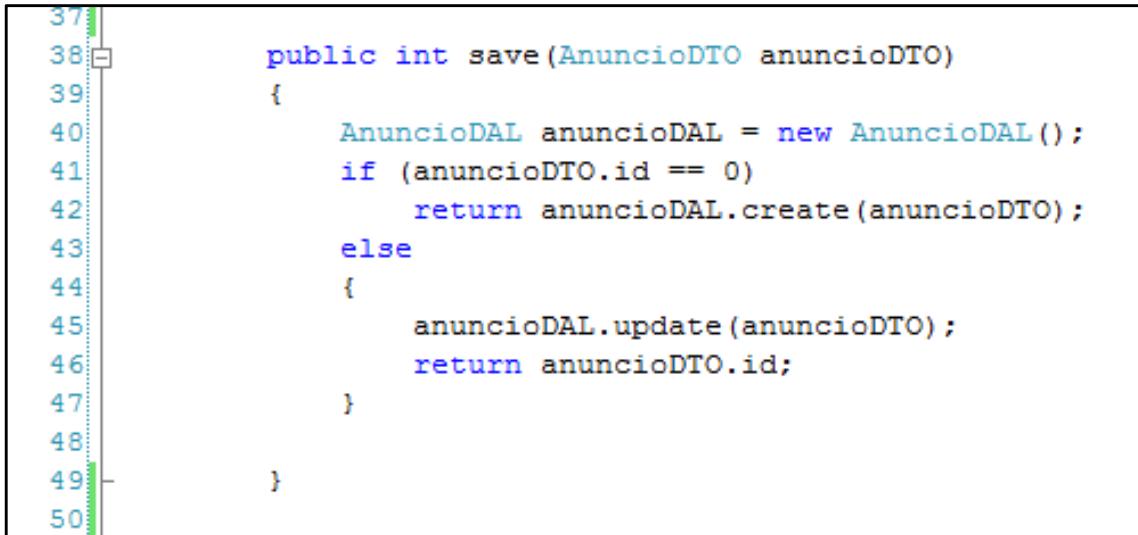
Na configuração do *profile*, é informada a string de conexão, utilizada para recuperar as informações do usuário.

Essas são as configurações básicas necessárias para autenticação de usuários.

4.3.2 Interação entre as camadas

A seguir é apresentado um trecho de código o qual demonstra como é feita a interação entre as camadas BLL e DAL utilizando um objeto DTO.

O código da figura 15 pertence a classe AnuncioBLL, responsável pelas regras de negócio referentes ao cadastro de anúncios no sistema.



```

37
38     public int save(AnuncioDTO anuncioDTO)
39     {
40         AnuncioDAL anuncioDAL = new AnuncioDAL();
41         if (anuncioDTO.id == 0)
42             return anuncioDAL.create(anuncioDTO);
43         else
44         {
45             anuncioDAL.update(anuncioDTO);
46             return anuncioDTO.id;
47         }
48     }
49 }
50

```

Figura 15: Interação entre as camadas

O código apresentado na figura 15 refere-se ao método salvar da classe AnuncioBLL, responsável pelas regras de negócios pertinentes aos anúncios do site.

Na linha 38, tem se o método *save* recebendo como parâmetro um objeto do tipo *anuncioDTO*, são os dados desse objeto que serão persistidos no banco de dados. Na linha 41 existe uma condicional que verifica se o anúncio é novo ou se trata de uma alteração, com isso o método *save* define qual método da classe *UsuarioDAL* deverá ser chamado.

Todo o sistema está baseado nessa troca de objetos em que os objetos da camada DTO são transferidos entre a BLL, DAL e *View*.

4.3.3 Consumo do Web Service

Como já mencionado, os preços das mercadorias dos supermercados serão consultados via Web Services. Embora a criação e manipulação de arquivos XML na plataforma .NET seja bastante simples, é possível efetuar uma requisição ao Web Service sem a necessidade de codificar arquivos XML, basta adicionar uma referência ao projeto de um Web Service e então efetuar a chamada atribuindo a resposta a um objeto *DataSet*, o qual pode ser serializado em XML.

A figura 16 apresenta o código necessário para consumir um Web Service.

```

37
38     WebService.wsp wst = new WebService.wsp();
39     wst.Url = mercadoWS.URI;
40
41     try
42     {
43         DsResultado = wst.RetornaPrecos(codigos.ToArray());

```

Figura 16: Consumo do Web Service

A linha 38 da figura 16 cria uma instância do Web Service referenciado ao projeto. A linha 39 atribui o endereço do mesmo, e, na linha 43, é feita a requisição. Neste trabalho as requisições aos Web Services são síncronas, o que significa que o sistema faz a requisição e fica aguardando a resposta.

4.3.4 Programação paralela

A fim de alcançar o menor tempo possível para a realização da tarefa de comparação de preços, a solução faz uso da programação concorrente para realizar as chamadas aos Web Services.

Com isso, pode-se chamar quase que simultaneamente todos os serviços, e o tratamento do retorno das chamadas é realizado conforme o tempo de resposta de cada Web Service.

O recurso essencial dessa solução são as threads, no qual, cada Web Service é chamado em uma thread que aguarda a resposta na mesma conexão, conforme apresentado na figura 17.

```

173
174     //Faz as chamadas aos web services paralelamente
175     for (QtdeMercados = 0; QtdeMercados < buscaMercados.mercados.Count; QtdeMercados++)
176     {
177
178         MercadoWsDTO mercadoWS = new MercadoWsDTO();
179         mercadoWS.URI = buscaMercados.mercados[QtdeMercados].URI;
180         mercadoWS.idMercado = buscaMercados.mercados[QtdeMercados].idMercado;
181         mercadoWS.nomeMercado = buscaMercados.mercados[QtdeMercados].mercadoNome;
182         mercadoWS.itens = buscaMercadorias.listaItens;
183
184         WebServiceBLL service = new WebServiceBLL();
185         service.comparaPrecos = comparaPrecosBLL;
186         Thread threadWebService = new Thread(new ParameterizedThreadStart(service.buscar));
187         threadWebService.Start(mercadoWS);
188         listaServices.Add(service);
189     }
190

```

Figura 17: Chamadas paralelas aos Web Services

O trecho de código da figura 17 se refere à Classe BuscaBLL responsável por buscar os preços das mercadorias em cada supermercado, o código apresenta uma iteração que percorre a lista de supermercados do usuário e os encapsulam em uma thread que faz as requisições aos Web Services dos mesmos.

Já o processo de comparação de preços o qual está compartilhado entre as diversas Thread, é destinado a gerar uma lista com os menores preços, entretanto essa lista não pode ser manipulada paralelamente, pois para se gerar tal lista é executada uma comparação de valores, na qual o valor que está sendo comparado não pode ser alterado por outra *Thread*.

A solução para esse problema é o uso de semáforos, responsáveis por gerenciar a entrada na chamada região crítica, a qual permite que somente uma *thread* execute um determinado bloco de código por vez.

A figura 18 ilustra o tempo de resposta de cada Web Service e a entrada na região crítica.

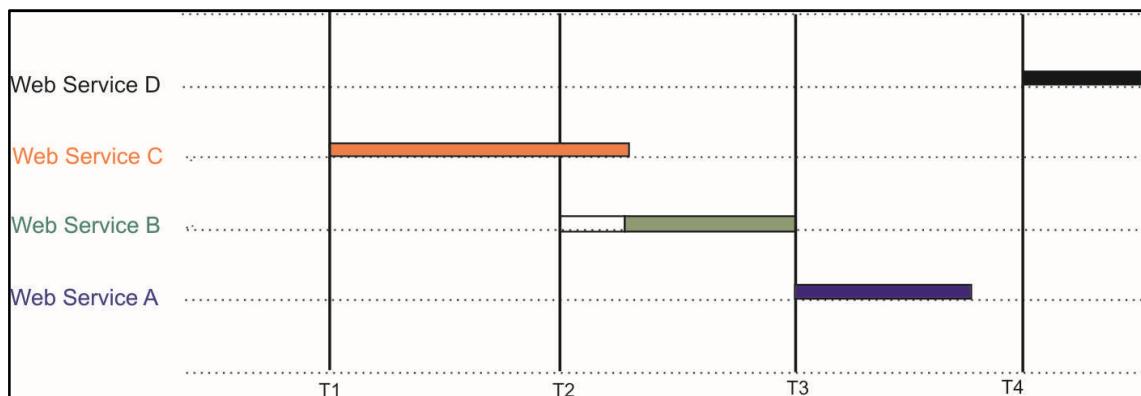


Figura 18: Tempo de resposta Web Services

No cenário apresentado na figura 18, o primeiro Web Service a responder é o “Web Service C” no tempo 1(T1), este entra na região crítica. No tempo 2 (T2), o “Web Service B” está pronto para entrar na região crítica, entretanto ele precisa aguardar o término do processo do “Web Service C” para então executar o código da região crítica. Quando o “Web Service A” responde, a região crítica está liberada e o código pode imediatamente ser executado.

A figura 19 apresenta o código de um determinado método que é executado na região crítica.

```

58     public void AddPrecoPorMercadorias(ListaItensDTO item, string nomeMercado , double valor)
59     {
60         mutexPrecoPorMercado.WaitOne();
61         Boolean encontrou = false;
62         foreach(MercadoriasPrecos mp in listaPrecosPorMercadorias )
63         {
64             if (mp.codMercadoria.Equals(item.codBarras))
65             {
66                 mp.mercadosPrecos.Add(nomeMercado, valor);
67                 encontrou = true;
68                 break;
69             }
70         }
71         if (encontrou == false)
72         {
73             MercadoriasPrecos mp = new MercadoriasPrecos();
74             mp.codMercadoria = item.codBarras;
75             mp.mercadosPrecos.Add(nomeMercado, valor);
76             mp.marca = item.marcaNome;
77             mp.id = item.idMercadoria;
78             mp.nome = item.mercadoriaNome;
79             mp.extensaoImagen = item.ExtensoaImagen;
80             listaPrecosPorMercadorias.Add(mp);
81         }
82     }
83 }
```

Figura 19: Código região crítica

Na linha 60 da figura 19, encontra-se o código do semáforo chamado de “MutexPrecoPorMercado”. O método WaitOne é responsável por bloquear a entrada de novas thread no bloco de código , e o método ReleaseMutex, na linha 82, libera a região para outras thread.

4.4 PROJETO DE INTERFACE COM O USUÁRIO

A usabilidade do site é um importante requisito não funcional que deve ser criteriosamente avaliado, afinal, estima-se que o provável usuário do sistema seja uma dona de casa, um idoso ou qualquer pessoa que tenha acesso a um computador. Por esse motivo o público alvo é diverso, podendo ter como usuários pessoas com muita experiência em navegar na Internet, ou pessoas com muita dificuldade em utilizá-la.

Nesse sentido, mapear o perfil do usuário não é uma tarefa trivial, por isso, este projeto busca simplificar ao máximo a naveabilidade, tornando o sistema o mais intuitivo possível.

O Projeto de interface está dividido nos seguintes aspectos:

- Facilidade de Aprendizado: O sistema se propõe a utilizar padronização dos ícones e cores utilizadas além de buscar a utilização de ícones com bastante significância para usuário de forma que o mesmo possa associar a imagem ao recurso.

- Facilidade de uso: O projeto de interface busca diminuir a quantidade de iterações necessárias para o usuário completar uma dada tarefa, dessa forma diminuindo a possibilidade de erros na operação do site.

A fim de alcançar sucesso nos aspectos citados anteriormente, o projeto de tela do sistema segue o padrão encontrado na maioria dos sites de pesquisa disponíveis no mercado, dessa forma estima-se que os usuários poderão aproveitar a experiência em outros sites para utilizar o Clube da Compra.

A Figura 10 mostra a tela inicial para usuários anônimos, ou seja, que não estão *logados* no sistema.

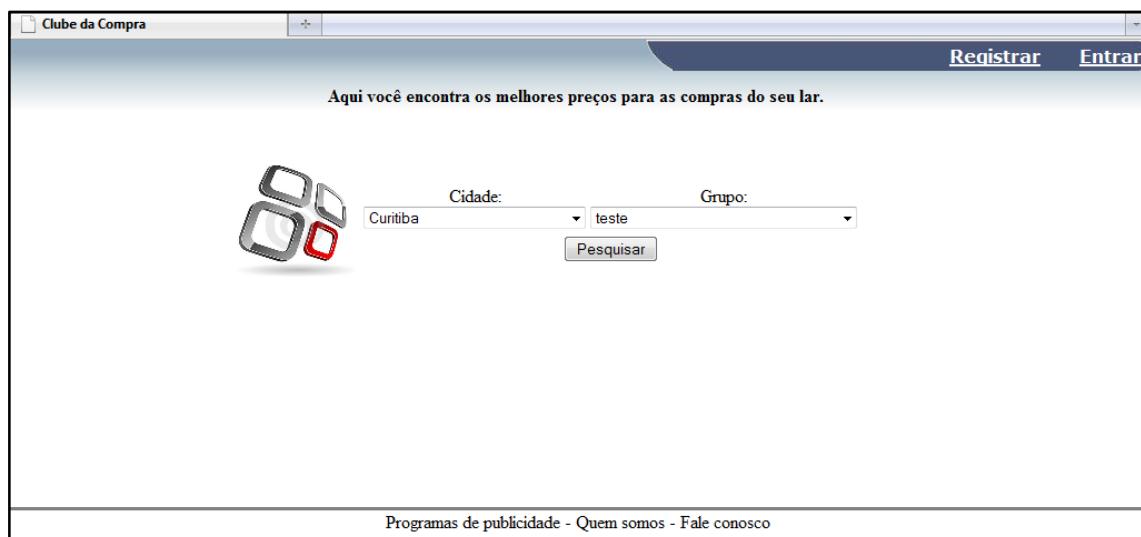


Figura 20: Página inicial da solução

Conforme apresentado na Figura 20, a *interface* da página inicial do sistema se espelhou no padrão dos sites de busca disponíveis atualmente, como por exemplo, o Google.

Para usuários que já possuem registro no sistema, é possível acessar o site clicando no *link* “Entrar”, no canto superior direito da tela. Ao clicar neste *link*, o mesmo será encaminhado para a página de *login* conforme mostrado na Figura 21.



Figura 21: Página de acesso ao sistema

Na página de acesso ao sistema, é solicitado o nome de usuário e a senha, caso o usuário tenha esquecido sua senha, é possível receber uma nova senha por e-mail clicando no link “Esqueci minha senha!”. A página de *login* também foi baseada nas páginas de acesso a e-mails vastamente utilizada como por exemplo hotmail.

Após logado no sistema, um usuário com o perfil de administrador tem acesso a diversas páginas, dentre elas, a página com as mercadorias cadastradas, conforme a Figura 22.

Mercadorias Cadastradas		
Pesquisa:	Categoria	Localizar
Código Barras	Nome	Marca
1234567890123	Alterado	
0987654635214	Refrigerante Coca	
0111223311223	Água Tônica	
0011928388477	Açúcar Refinado	
0987876765654	Arroz	
0807060504032	Fósforo	
0001112223334	Biscoito Waffer Chocolate	
0099887766112	Biscoito Passatempo	
0192019201920	Biscoito Bono Chocolate	
1 2 3		
Ações Inserir Alterar Excluir		

Figura 22: Mercadorias cadastradas

Na página apresentada na Figura 22, o usuário pode além de pesquisar as mercadorias, inserir novas, alterá-las e também excluí-las.

Ao clicar no botão “Inserir”, a página de cadastro de produtos é apresentada, conforme Figura 23.

The screenshot shows a web application interface for managing a shopping club. At the top, there's a logo and the text "Clube da Compra Prazer em Economizar". The top right corner has a "Olá admin" greeting and a "Sair" button. Below the header is a navigation menu with links: Inicio, Meus Mercados, Lista de Compras, Pesquisa de preços, Perfil, Mercados, Mercadorias (which is highlighted in blue), Aprovação de Anúncios, and Cadastro de marcas. The main content area is titled "Dados do Produto". It contains several input fields: "Nome" (Name) with a placeholder "[]", "Código de Barras" (Barcode Code) with a placeholder "[]", "Foto" (Photo) with a placeholder "[]" and a "Enviar arquivo..." (Upload file...) button, "Grupo" (Group) with a dropdown menu "Selecionar um grupo", "Marca" (Brand) with a dropdown menu "Selecionar uma marca", "UN" (Unit) with a dropdown menu "Kilos", and "Peso/Volume" (Weight/Volume) with a placeholder "[]". At the bottom of the form are two buttons: "Novo" (New) and "Confirmar" (Confirm).

Figura 23: Cadastro de Mercadorias

O cadastro de uma nova mercadoria é composto pelos campos apresentados na figura 23, sendo que somente os campos “UN” e “Peso/Volume” não são obrigatórios.

Para excluir uma mercadoria, o usuário clica no botão “Excluir”, na página de mercadorias cadastradas, uma página solicitando a confirmação da exclusão é apresentada, conforme a Figura 24.

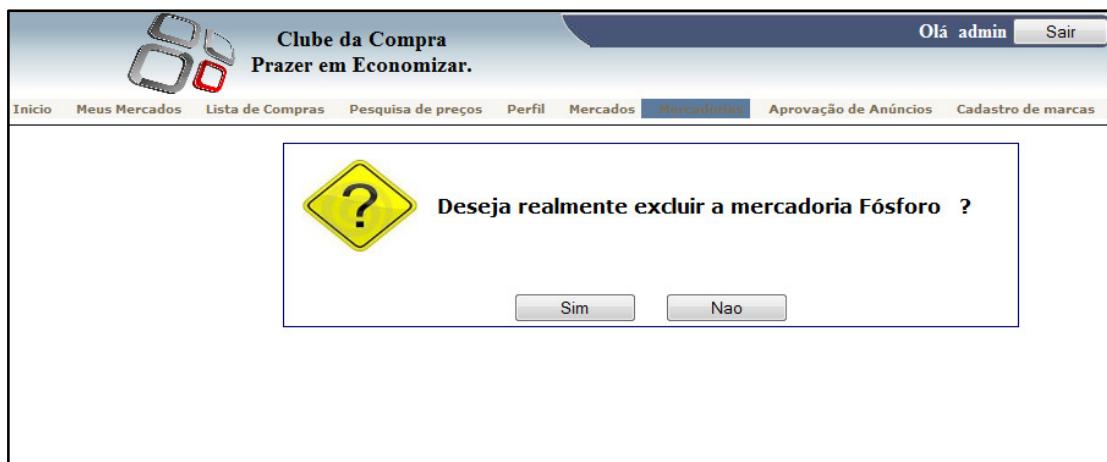


Figura 24: Confirmação de exclusão

Caso o usuário esteja certo que deseja excluir a mercadoria, basta clicar no botão “Sim” para a mercadoria ser excluída do banco de dados.

Usuários membros, isto é, os consumidores que desejam pesquisar os preços das mercadorias, podem criar suas listas de compras como mostrado na Figura 25.

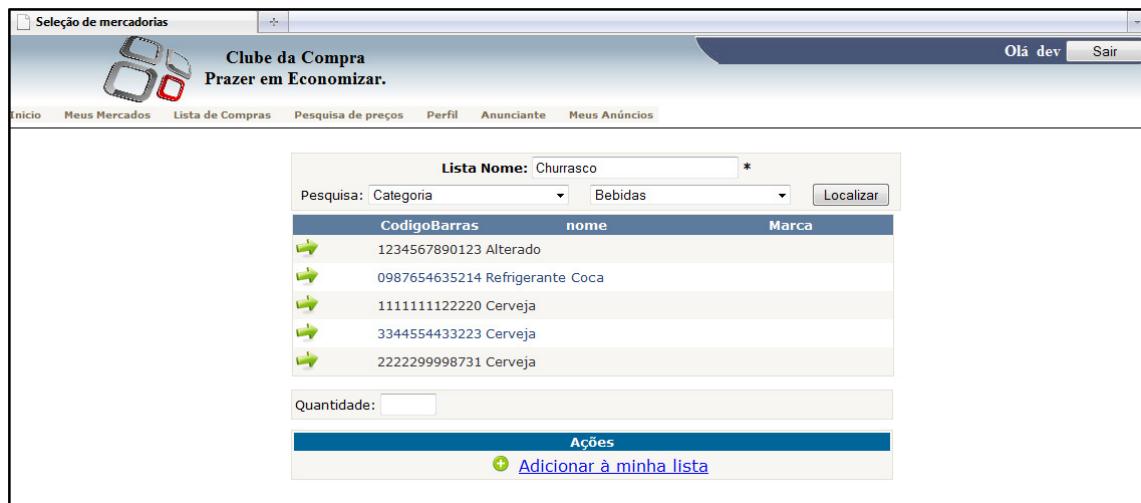


Figura 25: Criação da lista de compra

A criação da lista de compras consiste em: informar um nome a lista, pesquisar as mercadorias desejadas, informar a quantidade e clicar no botão “Adicionar à minha lista”.

Para pesquisar o preço das mercadorias de uma determinada lista de compras, basta acessar o menu “Pesquisa de Preços” conforme mostra a Figura 26.

nome	Data de Cadastro
Mensal	29/04/2010
Natal	29/04/2010
Semanal	06/05/2010
refri	28/05/2010

Figura 26: Pesquisa de Preços

A Figura 26 mostra todas as listas de compras cadastradas pelo usuário, para comparar os preços basta clicar no botão “Comparar Preços”. Uma nova página é carregada com o resultado da comparação conforme a Figura 27.

Foto	Código de Barras	Nome	Marca	Super Muffato	Condor
	0987654635214	Refrigerante Coca	Coca Cola	0	0
	0111223311223	Água Tônica	Antartica	0	0
	0011928388477	Açúcar Refinado	União	3,8	4
	0807060504032	Fósforo	Fiat Luz	1,3	1,5
	0099887766112	Biscoito Passatempo	Nestle	2	1,75

Figura 27: Resultado geral da comparação

A primeira página a ser apresentada é uma lista geral das mercadorias, com seus respectivos preços em cada supermercado, para verificar os menores preços, basta escolher a opção “Menores preços” no controle de visualização, conforme ilustra a Figura 28.

Mercado	Código de Barras	Mercadoria	Marca	Valor Unitário	Quantidade	Valor Total
Super Muffato	0011928388477	Açúcar Refinado	União	3,8	1	3,8
Super Muffato	0807060504032	Fósforo	Fiat Luz	1,3	1	1,3
Condor	0099887766112	Biscoito Passatempo	Nestle	1,75	1	1,75
Condor	5647364758374	Biscoito Prestígio	Nestle	1,5	1	1,5
Condor	3344554433223	Cerveja	Bavaria	3,2	1	3,2
Condor	222111332222	Creme de Leite	Nestle	5,99	1	5,99
Super Muffato	2221110000000	Desinfetante	Pinho Sol	2	1	2

Figura 28: Página com os menores preços das mercadorias

A página com os menores preços, é o resultado da seleção das mercadorias cujo o preço era menor em cada supermercado, essa página apresenta o percentual de economia que o usuário terá ao comprar nos supermercados indicados na lista.

O sistema fornece ainda, uma página com os preços de cada supermercado, por exemplo, a Figura 29 apresenta os preços fictícios da lista de compras no supermercado Condor.

Código de Barras	Nome	Marca	Preço	Quantidade	Total
0987654635214	Refrigerante Coca	Coca Cola	0	1	0
0111223311223	Água Tônica	Antartica	0	1	0
0011928388477	Açucar Refinado	União	4	1	4
0807060504032	Fósforo	Fiat Luz	1,5	1	1,5
0099887766112	Biscoito Passatempo	Nestle	1,75	1	1,75
5647364758374	Biscoito Prestígio	Nestle	1,5	1	1,5
3344554433223	Cerveja	Bavaria	3,2	1	3,2
222111332222	Creme de Leite	Nestle	5,99	1	5,99
2221110000000	Desinfetante	Pinho Sol	2,65	1	2,65

Figura 29: Preços da lista do supermercado

Para imprimir a lista de compras com os respectivos preços, basta clicar no botão “Imprimir”, a Figura 30 mostra a página gerada para impressão.

SUPERMERCADO:	Condor		
Código	Nome	Quantidade	Preço Unitário
0987654635214	Refrigerante Coca Coca Cola	1	R\$ 0,00
0111223311223	Água Tônica Antartica	1	R\$ 0,00
0011928388477	Açucar Refinado União	1	R\$ 4,00
0807060504032	Fósforo Fiat Luz	1	R\$ 1,50
0099887766112	Biscoito Passatempo Nestle	1	R\$ 1,75
5647364758374	Biscoito Prestígio Nestle	1	R\$ 1,50
3344554433223	Cerveja Bavaria	1	R\$ 3,20
222111332222	Creme de Leite Nestle	1	R\$ 5,99
2221110000000	Desinfetante Pinho Sol	1	R\$ 2,65

Figura 30: Impressão da lista de compras.

Como já mencionado, o projeto de *interface* com o usuário procura manter um padrão de páginas já conhecido, visando, com isso, melhorar a naveabilidade do sistema e consequentemente diminuir a curva de aprendizado necessária para utilizá-lo.

4.5 TESTES E EXPERIMENTOS

Esta seção descreve a importância da área de testes nas soluções tecnológicas, e os objetivos que se pretendem alcançar com esses testes. Além disso, é apresentada a configuração do ambiente onde os mesmos serão efetuados, assim como as formas que serão conduzidos.

4.5.1 Objetivos

É fato que a qualidade de soluções Web depende de diversos fatores, tais como: largura da banda de Internet, configuração do servidor, tecnologia de acesso a dados, banco de dados, instruções SQL, técnicas de programação e *design*, segurança, confiabilidade, entre outros aspectos. Entretanto, algumas soluções de negócios necessitam de maior ênfase em alguns requisitos do que em outros.

Na solução proposta neste trabalho, estima-se que um fator de vital importância seja o desempenho e escalabilidade da solução.

Sendo assim, pretende-se nessa etapa de testes e experimentos identificar a velocidade de resposta da solução e também o comportamento da mesma em situações críticas, isto é, atendendo a grandes quantidades de requisições.

4.5.2 Configuração do ambiente.

Os testes de desempenho e escalabilidade serão efetuados em um ambiente fictício, não representando um ambiente real de produção, onde servidores de grande porte estariam hospedando a solução.

Criar um ambiente que represente a situação real de uma solução desse nível necessitaria de alto investimento em grandes servidores, o que inviabilizaria o desenvolvimento de um protótipo demonstrativo.

O ambiente de testes é composto pela seguinte infra estrutura:

- Servidor Web Service: Este servidor corresponde ao computador que simula o servidor dos supermercados, responsável por receber as requisições do Web Server, processá-las, ou seja, localizar as mercadorias em seu banco de dados e responder ao servidor Web com os preços solicitados.
- Servidor Web: Esse servidor corresponde ao servidor web se a solução estiver hospedada na internet, sendo responsável por receber as requisições do cliente, processá-las, efetuar as chamadas ao servidor Web Service, receber as respostas do Web Service, processá-la e responder ao cliente solicitante.
- Estação cliente: Essa estação corresponde à máquina do consumidor que está utilizando um navegador como, por exemplo: o Internet Explorer, e deseja consultar os preços das mercadorias.

Todos os recursos citados acima são simulados com um único computador, com a seguinte configuração: NoteBook DELL Inspiron 1545 , Processador Intel Core2 DUO 2 GHZ, memória RAM 3GB, sistema operacional Windows 7 Ultimate.

4.5.3 Execução dos testes

A fim de identificar e isolar os recursos de software e hardware, os testes serão divididos em três categorias, que são elas:

- Teste de Desempenho: cujo propósito é verificar a velocidade de resposta da solução e também sua escalabilidade;
- Teste de carga: cujo propósito é verificar o comportamento da solução sob um ambiente normal de execução e sob condições de alta demanda, dessa forma identificando a utilização de recursos e pontos da aplicação que venham a falhar em casos extremos;
- Teste de Stress: cujo propósito é encontrar erros na aplicação que somente são expostos quando uma alta demanda é executada, objetivando mostrar o comportamento da solução em um ambiente de condições extremas.

Nesse contexto, desempenho refere-se à velocidade de resposta da solução, quanto menor o tempo para atender uma solicitação, melhor o desempenho do sistema; e como escalabilidade entende-se como sendo a habilidade da solução em continuar atendendo com

alta performance as requisições, mesmo que o número de usuários aumente significativamente.

Nesse sentido, os testes realizados buscam identificar o quanto escalar é a solução, isto é, quantos usuários simultâneos a solução é capaz de atender com performance aceitável.

Para que os testes possam ser realizados com eficácia será utilizada uma ferramenta própria para testes, disponível na suíte de aplicações do *Visual Studio 10 Team System* [36]. Com essa ferramenta é possível identificar o tempo de resposta da solução, além de se ter a possibilidade de simular um ambiente crítico com diversas requisições ao servidor simultaneamente.

Sem a ajuda dessa ferramenta, os testes seriam superficiais, tendo em vista a inviabilidade de conseguir centenas de usuários para testar a aplicação.

Os testes são realizados com base no cenário de comparação de preços, sendo que, as demais áreas do site não são consideradas críticas, ou seja, não requer muito processamento.

O primeiro teste consiste no teste de desempenho em um cenário no qual, os usuários são gradativamente adicionados até o atingir o número de 10 usuários fazendo as seguintes operações: acessam a página inicial do sistema, fazem o *login*, acessam a página de pesquisa de preços e submetem uma lista de compras à comparação de preços em dois *Web Services* e por fim efetuam *logoff*. O resultado desse teste está resumido na Figura 31.

Load Test Summary	
Test Run Information	
Load test name	LoadTest5
Description	
Start time	13/06/2010 17:17:37
End time	13/06/2010 17:19:37
Warm-up duration	00:00:00
Duration	00:02:00
Controller	Local run
Number of agents	1
Run settings used	Run Settings1
Overall Results	
Max User Load	10
Tests/Sec	0,40
Tests Failed	0
Avg. Test Time (sec)	14,2
Transactions/Sec	0
Avg. Transaction Time (sec)	0
Pages/Sec	3,03
Avg. Page Time (sec)	0,30
Requests/Sec	14,6
Requests Failed	0
Requests Cached Percentage	53,7
Avg. Response Time (sec)	0,063
Avg. Content Length (bytes)	2.406
Key Statistic Top 5 Slowest Pages	
Name	
URL (Link to More Details)	95% Page Time (sec)
http://www.cdc.com/View/User/PesquisaPre...	2,11
http://www.cdc.com/View/Login.aspx {POST...	0,38
http://www.cdc.com/View/User/ResultadoPr...	0,076
http://www.cdc.com/View/User/PesquisaPre...	0,068
http://www.cdc.com/View/Login.aspx {GET}	0,047
Key Statistic: Top 5 Slowest Tests	
Name	
Name	95% Test Time (sec)
WebTestSemanalComLogoff	16,0

Figura 31: Relatório do resultado de teste de desempenho

O relatório gerado apresenta informações importantes relacionadas à *performance* da solução, são elas: na seção “Key Statistic Top Slowest Pages” é apresentado a média de tempo de resposta a página “PesquisaPreços” na qual o valor é de 2,11 segundos que equivale ao tempo médio que o usuário teria que esperar para ver o resultado da comparação dos preços. Na seção “Overall Results” no item “Tests Failed” é possível verificar que nenhum item falhou neste cenário.

Uma tabela com a estatística do teste é gerado, conforme apresentada na Figura 32.

The screenshot shows a software interface for performance testing. At the top, there's a header bar with various icons. Below it, the main area has two tabs: 'Pages' and 'Errors'. The 'Pages' tab is active and displays a detailed table of results for different web pages. The columns include: Page, Scenario, Test, Network, Total, Ave, Min, Median, 90%, 95%, 99%, Max, Std Dev, and % Meeting Goal. The table lists several pages with their respective statistics. The 'Errors' tab is also visible below, showing a table with columns Type, Subtype, Count, and Last Message, with a single entry for 'Total' with a count of 0.

Pages													
Page	Scenario	Test	Network	Total	Ave	Min	Median	90%	95%	99%	Max	Std Dev	% Meeting Goal
www.cdc.com/	Normal	WebTestSema...	LAN	58	0,012	0,0...	0,0080	0,017	0,023	0,11	0,11	0,014	100
Login.aspx(GET)	Normal	WebTestSema...	LAN	55	0,014	0,0...	0,0050	0,016	0,047	0,23	0,23	0,034	100
Login.aspx(POST)	Normal	WebTestSema...	LAN	54	0,056	0,011	0,021	0,084	0,38	0,45	0,45	0,100	100
PesquisaPrecos.aspx(GET)	Normal	WebTestSema...	LAN	50	0,044	0,016	0,027	0,041	0,068	0,50	0,50	0,079	100
PesquisaPrecos.aspx(POST)	Normal	WebTestSema...	LAN	98	1,04	0,011	0,32	2,10	2,11	2,25	2,25	1,03	100
ResultadoPrecos.aspx(POST)	Normal	WebTestSema...	LAN	48	0,035	0,023	0,034	0,046	0,076	0,079	0,079	0,012	100

Errors			
Type	Subtype	Count	Last Message
Total		0	

Figura 32: Estatística do teste de desempenho

Essa tabela mostra o tempo médio, máximo, mínimo, entre outros valores, de cada página requisitada no cenário.

O segundo teste consiste no teste de carga, o qual 100 usuários executam o mesmo cenário do teste de desempenho, o teste inicia com 10 usuários e a cada cinco segundos são adicionados 10 usuários, esse teste tem a duração de dois minutos. O resultado é apresentado na Figura 33.

The screenshot shows a software interface for performance testing. The 'Pages' tab is active, displaying a table of results for a 'SemanalLogof' scenario. The columns include: Page, Scenario, Test, Total, Ave, Min, Median, 90%, 95%, 99%, Max, Std Dev, Goal, and % Meeting %. The table lists several pages with their respective statistics under this scenario. The values generally show higher response times and standard deviations compared to the results in Figure 32.

Pages													
Page	Scenario	Test	Total	Ave	Min	Median	90%	95%	99%	Max	Std Dev	Goal	% Meeting %
www.cdc.com/	SemanalLogof	WebTestSemanal...	363	0,60	0,0070	0,30	1,60	1,94	2,19	4,54	0,71	0	100
Login.aspx(GET)	SemanalLogof	WebTestSemanal...	360	0,78	0,0030	0,098	2,31	3,49	4,09	4,20	1,08	0	100
Login.aspx(POST)	SemanalLogof	WebTestSemanal...	348	1,36	0,011	0,44	4,60	5,01	6,09	6,17	1,70	0	100
PesquisaPrecos.aspx(GET)	SemanalLogof	WebTestSemanal...	318	0,69	0,016	0,38	1,92	1,98	2,31	2,41	0,70	0	100
PesquisaPrecos.aspx(POST)	SemanalLogof	WebTestSemanal...	619	2,13	0,0100	2,10	4,93	6,28	7,90	8,23	1,97	0	100
ResultadoPrecos.aspx(POST)	SemanalLogof	WebTestSemanal...	301	1,45	0	0,79	3,95	5,18	6,55	6,78	1,69	0	100

Figura 33: Estatística do teste de carga

É possível perceber, que o método *POST* da página “PesquisaPreços.aspx” tem a média de 2,13 segundos, tempo médio que o usuário teria que aguardar para ver o resultado da comparação dos preços.

O gráfico da Figura 34 apresenta o comportamento da solução à medida que o número de usuários aumenta.

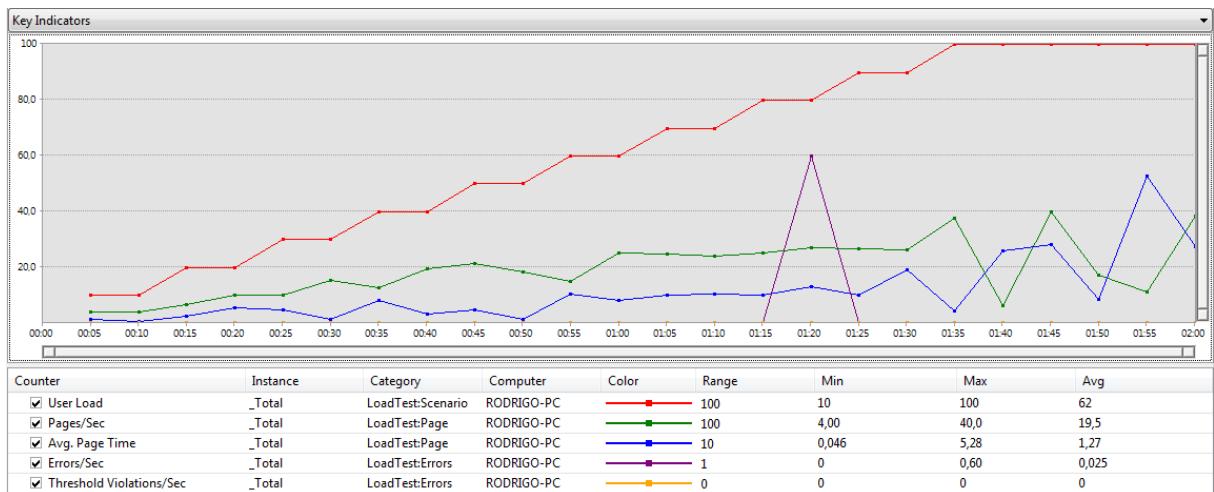


Figura 34: Gráfico de indicadores chaves do teste de carga

A linha vermelha no gráfico da Figura 34 indica o número de usuários no site, a linha azul o tempo médio de resposta da página, neste cenário a solução apresentou uma falha, indicada pela linha roxa, quando aproximadamente oitenta usuários estavam utilizando a solução, é possível observar que o aumento do número de usuários pouco interfere no tempo de resposta da página, das quais tiveram maior oscilação quando cem usuários utilizavam o site.

O último teste refere-se ao teste de *stress*, o qual foi realizado no mesmo cenário dos testes anteriores, a fim de manter consistência entre os resultados, entretanto esse simula duzentos e cinqüenta usuários, sendo que, a cada dez segundos, acrescentam-se vinte e cinco usuários. O resumo desse teste pode ser visualizado no relatório da Figura 35.

Load Test Summary	
Test Run Information	
Load test name	LoadTest4
Description	
Start time	13/06/2010 16:26:27
End time	13/06/2010 16:28:27
Warm-up duration	00:00:00
Duration	00:02:00
Controller	Local run
Number of agents	1
Run settings used	Run Settings1

Key Statistic: Top 5 Slowest Pages	
URL (Link to More Details)	Avg. Page Time (sec)
http://www.cdc.com/View/Login.aspx {POST...	9,95
http://www.cdc.com/View/User/ResultadoPr...	7,50
http://www.cdc.com/View/User/PesquisaPre...	3,60
http://www.cdc.com/	2,09
http://www.cdc.com/View/Login.aspx {GET...	1,26

Key Statistic: Top 5 Slowest Tests	
Name	Avg. Test Time (sec)
WebTestSemanalComLogoff	35,4

Overall Results	
Max User Load	250
Tests/Sec	2,71
Tests Failed	0
Avg. Test Time (sec)	35,4
Transactions/Sec	0
Avg. Transaction Time (sec)	0
Pages/Sec	26,2
Avg. Page Time (sec)	3,88
Requests/Sec	140
Requests Failed	0
Requests Cached Percentage	45,7
Avg. Response Time (sec)	0,76
Avg. Content Length (bytes)	3.876

Figura 35: Relatório do resultado de teste de stress

O resultado desse teste mostra que embora haja uma queda na velocidade de resposta da solução, não houve nenhuma falha, o que indica que o sistema pode responder a um grande número de usuários sem apresentar erros.

A Figura 36 apresenta o gráfico com o resultado do teste de stress.

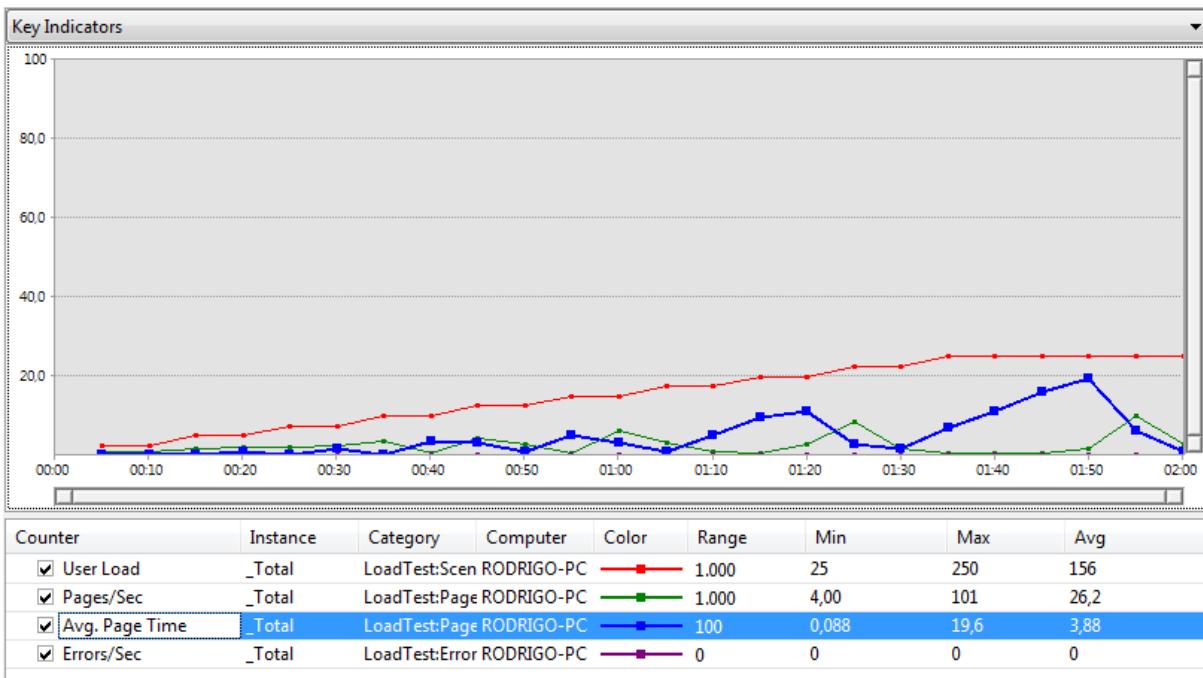


Figura 36: Gráfico de indicadores chaves do teste de stress

Embora, haja picos de aumento, no qual o maior tempo de espera chegou a 19,6 segundos de espera da página,, o tempo médio é de 3,88 segundos, tempo este aceitável para uma solução desse nível.

A tabela 2 reúne os resultados em um único lugar para facilitar a comparação.

Tabela 2: Comparaçao entre os resultados dos testes.

Tipo do teste	Quantidade de usuários	Tempo médio de resposta
Teste de desempenho	De 1 à10	0,30 segundos
Teste de carga	De 10 à100	1,27 segundos
Teste de Stress	De 25 à 250	3,88 segundos

Por meio dos testes realizados, é possível afirmar que o tempo de resposta da solução é aceitável, mesmo em situações de alta demanda, esse tempo refere-se à média de resposta de todas as páginas do cenário utilizado nos testes.

5 CONCLUSÃO

Após o término da pesquisa que embasa esse trabalho, foi possível identificar os desafios e oportunidades na implementação do mesmo.

Conclui-se que a viabilidade de utilização da solução em um ambiente real se dá pelo conjunto de tecnologias aplicadas tais como a interoperabilidade dos arquivos XML e o protocolo SOAP que formão a estrutura principal dos Web Services. Com isso se torna possível consultar os dados distribuídos nas mais diversas bases heterogeneas espalhadas nos servidores de banco de dados dos supermercados.

A ampla disponibilização de informações sobre as tecnologias empregadas neste trabalho revela a popularidade das mesmas, e também sua aceitação no mercado. Pode-se afirmar que grande parte dos problemas envolvendo integração de dados no mundo corporativo pode ser resolvido com o uso de Web Service. Sendo assim, tal tecnologia mostra-se adequada à solução aqui proposta. Muitos trabalhos científicos publicados exploraram o mundo dos Web Services, alguns desses trabalhos citados nessa monografia revelam os desafios na implementação dos mesmos, tais trabalhos revelam que não é algo trivial desenvolver uma solução com Web Service de forma robusta e segura, diversos fatores devem ser levados em consideração na implementação de tal solução.

Como ponto forte do projeto proposto cabe destacar a singularidade do mesmo, nenhum outro trabalho que explorasse a mesma situação deste foi encontrado, na área comercial duas soluções similares foram encontradas, entretanto, as mesmas de longe se aproximam dos benefícios que este projeto pode trazer a sociedade. O fato de existirem sites comerciais com propostas parecidas com este, demonstra a existência de uma movimentação do setor privado em prol deste segmento.

Os testes relevam que o tempo de resposta da solução é adequado, o que incentiva seu uso.

É fato que, em trabalhos futuros, muitas melhorias podem ser realizadas nesse projeto, dentre essas melhorias pode-se citar: gerar uma solução de e-commerce, na qual permita ao consumidor efetuar a compra e receber as mercadorias em sua casa. Outro trabalho que agregaria muito valor a esse, é utilizar técnicas de inteligência artificial para apresentar os anúncios de acordo com o perfil de cada usuário, tornando a veiculação dos mesmos no site mais inteligente e direcionada. A usabilidade do sistema também é de vital importância, sendo

assim, utilizar os conceitos de interação homem-máquina e técnicas de aplicações ricas para re-modelar a *interface* do usuário certamente aumentará ainda mais a aceitação pela solução.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] PRESTES, Evandro guerreiro. **Cidade Digital Infoinclusão social e tecnologia em rede** Editora:.Senac São Paulo 2006.
- [2] IBGE, Instituto Brasileiro de Geografia e Estatística. **Apresenta resultados de pesquisas brasileiras**.Ano 2002/2003. Disponível em www.ibge.gov.br/home/estatistica/populacao/condicaodevida/pof/2002/tabc412.pdf Acesso em: 20/08/2009.
- [3] Santana Filho, Ozeas Vieira **Microsoft .net uma visão para programadores** São Paulo, 2002. Ed. Senac ISBN 85-7359-268-0.
- [4] MSDN, **Visão geral do Common Language Runtime** Disponivel em: <http://msdn.microsoft.com/pt-br/library/ddk909ch.aspx>. Acesso em: 08/09/2009.
- [5] de Araújo, Anderson Viçoso **Treinamento Avançado em .NET** Ed. Digerati ISBN 857702025-8].
- [6] MSDN, **ASP.NET Model View Controller.** Disponível em: <http://msdn.microsoft.com/en-us/library/dd394709.aspx>. Acesso em 01/09/2009
- [7] MSDN, **ASP.NET MVC Overview.** Disponível em: <http://msdn.microsoft.com/en-us/library/dd381412.aspx>. Acesso em 23/08/2009 .Acesso em:01/09/2009
- [8] Madureira Pinto, Marlene Idalina. **Caracterização de soluções de comércio electrónico B2B** Dissertação Mestrado - Universidade de Trás-os-Montes e Alto Douro 2007 disponivel em https://repositorio.utad.pt/bitstream/10348/83/1/msc_mimpinto.pdf. Acesso em: 02/09/2009.
- [9] Geciane S.Porto,Rienaldo N Braz, Guilerme Ary Plonski , **O Intercambio Eletrônico de dados – EDI e seus impactos organizacionais.** Disponível em: http://www.fae.edu/publicacoes/pdf/revista_da_fae/fae_v3_n3/o_intercambio_eletronico_de.pdf. Acesso em: 04/09/2009

[10] Healey,Marcus e Shyam Samtani. **B2B Integration: A Pratical Guide To Collaborative E-commerce.** Ed. Imperial College Press, 2002.
ISBN 1-86094-323-3

[11]UFRJ, **Extensible Markup Language.** Disponível em:
http://www.gta.ufrj.br/grad/00_1/miguel/link1.htm Acesso em 10/09/2009. Acesso em 04/09/2009.

[12] Cerami, Ethan, **Web Service Essential.** Ed.O'Reilly & Associates, 2002.
ISBN: 0-5-00224-6

[13] Newcomer, Eric, **Understanding Web Services: XML,WSDL,SOAP, and UDDI** Ed. Eric Newcomer 2002. ISBN 0201750813

[14] Behrouz A. Forouzan, **Comunicação de dados e redes de computadores,** Ed Artmed, 2004 ISBN 0-07-251584-8

[15] W3C, **Especificação de XML,** Disponivel em: <http://www.w3.org/XML/#intro>, Acesso em: 13/09/2009.

[16] SBC, MARTINS, Eliane, et al, **Desafios na construção e validação da robustez de aplicações orientadas a serviços,** Disponível em:
<http://www.lbd.dcc.ufmg.br:8080/colecoes/semish/2008/005.pdf>. Acesso em 04/09/2009

[17] SILVA, Reginaldo F, CUNHA, José A. **Arquitetura de segurança em aplicações baseadas em Web Service.** Disponível em :
<http://www2.ifrn.edu.br/ojs/index.php/HOLOS/article/view/77/82>. Acesso em: 04/10/2009

[18] POLL, Daniel, **Projeto para execução de consultas distribuídas usando Web Services.** Disponível em:
http://www.inf.unisc.br/repotcs/arquivos_tc/PROJETO_PARA_EXECU%C7%C3O_DE_CO_NSUMLAS_DISTRIBU%C3DAS_USANDO_WEB_SERVICES.pdf. Acesso em: 20/10/2009

[19] GUIA GEOGRÁFICO, Site com diversas informações a respeito da cidade de Curitiba. Disponível em: <http://www.curitiba-parana.net/educacao.htm>. Acesso em:08/10/2009

[20] FERRARI, Fabricio Agusto, **Crie banco de dados em MySQL** – São Paulo: Digerati Books 2007 ISBN 978-85-60480-25-8

[21] LIMA, Aline. Faça mais do seu dinheiro. **VocêS/A**, edição 131, p, 73, 2009-11-07

[22] ALBAHARI,Ben, DRAYTON, Peter, Merrill, Brad, **C# Essencial Second Edition**. Ed. O'Reilly & Associates books, 2002. ISBN: 0-596-00315-3

[23] FREEMAN, eric, Elisabeth Freeman. et al. **Head First Design Patterns**. Ed. O'Reilly Media inc. 2004.

[24] FEHILY, **Visual QuickStart Guide SQL, Third Edition**. Ed.Peachpit Press. 2008.

[25] CORREIOS, página de informações de serviços dos correios. Disponível em: http://www.correios.com.br/servicos/cep/cep_promo.cfm. Acesso em:05/10/2009

[26] XIGNITE, Site americano com diversos Web Services. Disponível em: <http://www.xignite.com/>. Acessado em: 07/10/2009.

[27] YAHOO, Site da yahoo com informações destinadas aos desenvolvedores. Disponível em:<http://developer.yahoo.com/search/>. Acesso em: 07/10/2009.

[28] EWEEK, Artigo sobre Web Service da Master Card. Disponível em: http://resource.mapinfo.com/static/files/document/1110292785547/eweeek_mastercard.pdf. Acesso em: 07/10/2009.

[29] WEBSERVICEX. Site com diversos Web Services. Disponível em: <http://www.webservicex.net/genericbarcode.asmx>. Acesso em: 08/10/2009.

[30] WEBSERVICEX. Site americanos com diversos Web Services. Disponível em: <http://www.webservicex.net/WCF/ServiceDetails.aspx?SID=46>. Acesso em: 08/10/2009.

[31]XMETHODS. **Site com diversos Web Services americano.** Disponível em: http://www.xmethods.net/ve2/ViewListing.po;jsessionid=XULCtNm0LFKB0_mB2fSnAH1O?key=427412. Acesso em: 08/10/2009.

[32] WEBSERVICESX. **Site com diversos Web Services.** Disponível em: <http://www.webservicex.net/WS/WSDetails.aspx?CATID=12&WSID=68>. Acesso em: 09/10/2009.

[33]ESENDEX, **Empresa americana que fornece Web Service para envio de SMS.** Disponível em: <https://www.esendex.co.uk/Services/Web-SMS>. Acesso em: 09/10/2009.

[34]MERCADOMINEIRO. **Portal com resultado de pesquisas de preços em diversos setores.** Disponível em: <http://www.mercadomineiro.com.br>. Acesso em: 23/09/2009.

[35]MERCADODOPRECO. **Portal para pesquisa de preços de mercadorias de alguns supermercados do Rio de Janeiro.** Disponível em: <http://www.mercadodopreco.com.br>. Acesso em: 23/09/2009.

[36]MSDN, **Microsoft Application Center Test 1.0, Visual Studio.NET Edition.** Disponível em: [http://msdn.microsoft.com/en-us/library/aa287410\(VS.71\).aspx](http://msdn.microsoft.com/en-us/library/aa287410(VS.71).aspx). Acesso em: 02/11/2009.

[36]MACDONALD, Matthew, SZPUSZTA, Mario . **Pro ASP.NET 3.5 in C# 2008,** Ed Apress, 2007 ISBN 978-1-59059-893-1

APENDICE

Especificação de caso de uso

Com base na lista de requisitos, foi desenvolvida a especificação de caso de uso, na qual é descrito, o passo-a-passo da iteração do usuário com o sistema, descrevendo detalhadamente as condições para se executar determinado recurso da solução.

Através do caso de uso foi possível identificar necessidades que o sistema deve suprir e o mesmo foi utilizado como base para criação dos diagramas de classes e estrutura do banco de dados.

Caso de uso: UC001-Comparar Preço

Ator: Usuário.

Descrição: Este caso de uso descreve como o usuário solicita ao sistema uma pesquisa de preço de uma determinada mercadoria, em uma determinada região.

Pré Condições:

Cenário Principal Sistema:

1. Carrega os campos de pesquisa os quais são cidade e grupo da mercadoria
2. Carrega opção das cidades que tiverem algum mercado cadastrado.

Cenário Principal Usuário:

1. Informa em qual cidade deseja fazer a comparação.
2. Informa o grupo da mercadoria que deseja pesquisar.
3. Clica no botão pesquisar.

Cenários Alternativos:

Pós Condições: O sistema efetua a comparação dos preços no Web Service dos supermercados e apresenta o resultado ao usuário.

Caso de uso: UC002- Cadastrar usuário

Ator: Usuário.

Descrição: Este caso de uso descreve o processo de cadastrar novos usuários no sistema.

Pré Condições:

Cenário Principal:

1. Usuário preenche o campo “Nome do Usuário”.
2. Usuário preenche o campo “Senha” e “Confirmar Senha”
3. Usuário preenche o campo “Email”
4. Usuário clica no botão “Criar Usuário”

Cenários Alternativos:

2. Usuário informa uma senha fora do padrão definido.
 - 2.1 Sistema envia alerta “Tamanho mínimo da senha: 5. Caracteres não alfanuméricos necessários: 0.”
3. Usuário informa um nome de usuário já cadastrado no sistema.
 - 3.1 Sistema envia alerta informando que o usuário já está cadastrado.

Pós Condições: Sistema redireciona a página de boas vindas.

Regras de Negócio: A senha informada deve ter no mínimo cinco caracteres e pode ser alfanumérica.

Caso de uso: UC003- Efetuar *logon*.

Autor: Usuário, Anunciante.

Descrição: Este caso de uso descreve como o usuário pode acessar sua página personalizada no sistema.

Pré Condições: Usuário deve estar cadastrado no sistema, UC002

Cenário Principal Sistema:

1. Carrega os campos Nome do usuário e senha.

Cenário Principal Usuário:

1. Preenche o campo “Nome do Usuário” de acesso.
2. Preenche o campo “Senha” de acesso.
3. Clica no botão “Fazer Logon”.

Cenários Alternativos:

1. Usuário não preencheu o campo “Nome do usuário” de acesso.
 - 1.1 Sistema envia alerta na forma de um asterisco na cor vermelha na frente do campo, indicando que o campo é obrigatório.
2. Usuário não preencheu o campo “Senha”.
 - 2.1 Sistema envia alerta na forma de um asterisco na cor vermelha na frente do “Senha”, indicando que o campo é obrigatório.
 3. Usuário informa nome do usuário ou senha de acesso inválida.

3.1 Sistema envia alerta informado que o *login* ou senha são inválidos.

4. Usuário esquece sua senha: UC004

Pós Condições: Usuário é redirecionado para sua página personalizada.

Regras de Negócio: o *login* de acesso é o e-mail do usuário, a senha de acesso deve conter no mínimo seis caracteres podendo ser alfanumérico.

Caso de uso: UC004- Recuperar senha.

Ator: Usuário, Sistema.

Descrição: Esse caso descreve como o usuário pode recuperar a senha de acesso ao sistema.

Pré Condições: Usuário deve estar cadastrado no sistema.

Cenário Principal Sistema:

Sistema solicita o nome do usuário.

Cenário Principal Usuário:

1. Acessa a área de *logon* no sistema.
2. Clica no clica em “Esqueci minha senha”
3. Usuário informa o nome de usuário.
4. Usuário clica em “Enviar”.

Cenários Alternativos:

1. Nome do usuário informado não esta cadastrado no sistema.

1.1 Sistema envia o seguinte alerta “Não foi possível acessar suas informações. Tente novamente”.

Pós Condições: Sistema envia um e-mail usuário, informando uma nova senha de acesso.

Caso de uso: UC005- Escolher supermercados.

Ator: Usuário.

Descrição: Esse caso de uso descreve como o usuário pode escolher os supermercados que farão parte de suas pesquisas de preço.

Pré Condições: Usuário deve estar logado no sistema. UC003, e na pagina de escolha de supermercados.

Cenário Principal:

1. Sistema apresenta as opções para filtrar os supermercados disponíveis.

- 1.1 As opções são Cidade e CNPJ do supermercado.
2. Usuário escolhe por qual opção quer localizar os supermercados.
3. Usuário informa os dados de acordo com a opção escolhida
4. Sistema busca os supermercados cadastrados de acordo com o filtro, e apresenta a lista de supermercados encontrados.
5. Usuário seleciona os supermercados de interesse.
6. Usuário clica no botão “Adicionar à minha lista”.

Cenários Alternativos:

1. Usuário não informa nenhum dado para filtrar os supermercados.
 - 1.1 Sistema envia alerta informando que os dados não foram informados para pesquisa.
2. Sistema não encontra nenhum supermercado no filtro do usuário.
 - 2.1 Sistema envia alerta informando que nenhum supermercado foi encontrado.

Pós Condições: Os supermercados selecionados são cadastrados no banco de dados para o usuário logado.

Caso de uso: UC006- Criar lista de compras

Ator: Usuário.

Descrição: Caso de uso que descreve o processo em que o usuário cria suas listas de compras. Ele é iniciado quando o usuário clica no item “Criar lista de compras”

Pré Condições: Usuário deve estar logado no sistema. UC003.

Cenário Principal Sistema:

Sistema carrega as opções de pesquisa de mercadoria que será adicionada à lista. 1.1 As opções são por: código de barras, nome, grupo e marca.

Sistema busca as mercadorias cadastradas que atendam o critério de pesquisa escolhido pelo usuário e as mostra em uma lista.

Sistema atualiza a lista de compras do usuário com a mercadoria adicionada.

Cenário Principal Usuário:

Usuário informa o nome da lista de compras.

Usuário marca opção de pesquisa

Usuário informa os dados da pesquisa e clica no botão localizar.

Usuário seleciona a mercadoria na lista apresentada, informa a quantidade e clica na opção “adicionar à minha lista”

Usuário clica em finalizar lista de compras.

Cenários Alternativos:

Usuário não informa o nome da lista de compras.

1.1 Sistema envia alerta solicitando o nome da lista de compras.

2. Usuário não preenche o campo para pesquisa.

2.1 Sistema envia alerta informando que o campo deve ser preenchido.

3. Usuário clica na opção “adicionar à minha lista de compras” sem selecionar uma mercadoria.

3.1 Sistema envia alerta informando que uma mercadoria deve ser selecionada.

4. Sistema não encontra nenhuma mercadoria no critério de pesquisa informado.

4.1 Sistema envia alerta informando que nenhuma mercadoria foi encontrada.

5. Usuário informa uma quantidade inválida para a mercadoria selecionada.

5.1 Sistema envia alerta informando que a quantidade informada não é válida para a mercadoria selecionada.

Pós Condições: A lista de compras é cadastrada no sistema.

Regras de Negócio: A quantidade deve ser um valor numérico, podendo ser fracionário se a unidade de medida cadastrada permitir.

Caso de uso: UC006-Alteração de cadastro.

Autor: Usuário.

Descrição: Esse caso de uso descreve como o usuário procede para alterar dados cadastrais.

Pré Condições: Estar logado no sistema - UC003

Cenário Principal:

1. Usuário clica na opção “Dados Cadastrais”,

2. Sistema apresenta os dados atuais do usuário que são : login e senha, a senha estará com máscara para não ser identificável.

3. Sistema apresenta as seguintes opções:

3.1 “Alterar Login”

3.2 ”Alterar Senha”

3.3 “Desativar conta”

4. Usuário seleciona a opção desejada.

5. Se a opção escolhida for “Alterar login” sistema apresenta tela de alteração de *login* - UC007

6. Se a opção escolhida for “Alterar senha”, sistema apresenta a tela de alteração de senha – UC007

7. Se a opção escolhida for “Desativar Conta”, sistema a tela de confirmação de desativação da conta – UC008

Pós Condições: A opção escolhida pelo usuário será realizada.

Regras de Negócio:

Caso de uso: UC007-Alteração de *Login*

Autor: Usuário.

Descrição: Esse caso de uso descreve o processo de alteração do *login* utilizado pelo usuário.

Pré Condições: Usuário deve estar logado no sistema - UC003

Cenário Principal:

Sistema mostra o *login* atual do usuário , campo para o novo *login* e campo para confirmação do *login*, onde o usuário repete a digitação do novo *login*.

Usuário informa o novo *login* desejado.

Usuário clica no botão alterar *login*.

Cenários Alternativos:

1. O novo *login* informado pelo usuário já esta em uso.

 1.1 Sistema envia alerta informando que o *login* escolhido já esta em uso.

2. O usuário digita o mesmo *login* que ele esta utilizando.

 2.1 Sistema envia alerta informando que o novo *login* deve ser diferente do *login* atual.

3. O novo *login* informado pelo usuário não caracteriza um endereço de e-mail.

 3.1 Sistema envia alerta informando que o formato do *login* é inválido e que o *login* deve ser um endereço de e-mail.

Pós Condições: Sistema envia alerta informando que o login foi alterado com sucesso.

Sistema envia um e-mail para o novo login do usuário informando que a alteração foi concluída com sucesso.Sistema redireciona o usuário para a página principal.

Regras de Negócio: O novo login informado deve ser um endereço de e-mail válido.

Caso de uso: UC008 – Alteração de senha

Autor: Usuário.

Descrição: Esse caso de uso descreve o procedimento de alteração de senha do usuário utilizada para acessar o sistema.

Pré Condições: Estar logado no sistema - UC003. Acessar dados cadastrais – UC008.

Cenário Principal:

Sistema solicita a senha atual utilizada para acessar o sistema, carrega o campo nova senha e confirmação de senha, onde o usuário digita novamente a senha informada no campo nova senha.

Usuário informa a senha atual.

Usuário informa a nova senha.

Usuário digita novamente a nova senha no campo para confirmação de senha.

Usuário clica no botão confirmar.

Cenários Alternativos:

1. Usuário não informa a nova senha.

 1.1 Sistema envia alerta informando que a nova senha deve ser digitada.

2. Usuário informa uma senha fora do padrão estabelecido na regras de negócios.

 2.1 Sistema envia alerta informando que a nova senha não está no padrão estabelecido e informa o padrão que a senha deve ser.

3. Usuário informa no campo de confirmação da nova senha uma senha diferente da informada no campo nova senha .

 3.1 Sistema envia alerta informando que a confirmação de senha e a nova senha não conferem.

Pós Condições: A nova senha é gravada no sistema. O sistema envia alerta informando que a senha foi alterada com sucesso. Sistema envia um e-mail para o *login* do usuário informando a nova senha. Sistema redireciona o usuário para a página principal.

Regras de Negócio: A senha deve ser no mínimo de seis caracteres podendo ser alfanumérico.

Caso de uso: UC009- Desativar de conta

Autor: Usuário.

Descrição: Esse caso de uso descreve o processo de desativação da conta, onde o usuário cancela seu registro no site. Dessa forma, todas as listas são apagadas do sistema.

Pré Condições: Estar logado no sistema – UC003. Acessar dados cadastrais - UC008

Cenário Principal:

Usuário escolhe a opção desativar conta.

Sistema envia alerta solicitando a confirmação da desativação da conta.

Usuário confirma a desativação da conta.

Cenários Alternativos: Não possui.

Pós Condições: Sistema apaga todos os registros do usuário logado. Sistema envia alerta informando que o cadastro foi desativado com sucesso. Sistema redireciona o usuário para a página inicial do site.

Caso de uso: UC010 – Adição de novas mercadorias à lista de compras.

Autor: Usuário.

Descrição: Esse caso descreve o procedimento para alterar as listas de compras cadastradas pelo usuário.

Pré Condições: Estar logado no sistema – UC003.

Cenário Principal:

1. Usuário seleciona o nome da lista de compras que deseja alterar.

2. Sistema lista as mercadorias da lista de compra selecionada.

3. Usuário escolhe a opção de adicionar novos itens à lista de compras.

Sistema carrega as opções de pesquisa de mercadoria que será adicionada à lista.

As opções são por: código de barras, nome, grupo.

5. Usuário faz a pesquisa das mercadorias disponíveis e seleciona a mercadoria que deseja adicionar à lista de compras.

6. Usuário informa a quantidade da mercadoria que será adicionada à lista de compras.

7. Usuário clica em adicionar à lista.

Cenários Alternativos:

1. Sistema não encontra nenhuma mercadoria nos critérios de pesquisa estabelecidos pelo usuário.

1.1 Sistema envia alerta informando que nenhuma mercadoria foi encontrada.

2. Usuário informa uma quantidade inválida para a mercadoria.

2.1 Sistema envia alerta informando que a quantidade não é válida para mercadoria selecionada.

3. Usuário não informa nenhuma mercadoria e clica em adicionar à lista.

3.1 Sistema envia alerta informando que uma mercadoria deve ser selecionada.

Pós Condições: A mercadoria selecionada é adicionada à lista de compras do usuário.

Regras de Negócio: A quantidade deve ser numérica podendo ser fracionada.

Caso de uso: UC011 – Alteração da quantidade de mercadoria na lista de compras.

Ator: Usuário.

Descrição: Esse caso descreve o procedimento de alteração da quantidade de uma determinada mercadoria da lista de compras cadastradas pelo usuário.

Pré Condições: Estar logado no sistema – UC003, acessar a página de lista de compras.

Cenário Principal:

1. Usuário seleciona o nome da lista de compras que deseja alterar.
2. Usuário clica no botão "Itens".
3. Usuário seleciona a mercadoria que deseja alterar a quantidade.
4. Usuário informa a nova quantidade.
5. Usuário clica no botão "Alterar Quantidade".

Cenários Alternativos: Não possui.

Pós Condições: A quantidade da mercadoria é alterada na lista de compras selecionada.

Regras de Negócio: A quantidade deve ser numérica podendo ser fracionada.

Caso de uso: UC012 – Exclusão de mercadoria da lista de compras.

Ator: Usuário.

Descrição: Esse caso descreve o procedimento para excluir uma mercadoria da lista de compras.

Pré Condições: Estar logado no sistema – UC003, acessar a página de lista de compras.

Cenário Principal:

1. Usuário seleciona a lista de compras que deseja alterar.
2. Usuário clica no botão "Itens".
3. Usuário seleciona a mercadoria que deseja excluir da lista.
4. Usuário clica no botão "Excluir".
5. Sistema pergunta solicita confirmação de exclusão.
6. Usuário clica no botão "Sim".

Cenários Alternativos: Não possui.

6. Usuário clica no botão "Não".

6.1 Sistema volta a página de itens da lista, sem excluir o item.

Pós Condições: A mercadoria selecionada é excluída da lista de compras selecionada.

Caso de uso: UC013 – Cadastrar anunciantes.

Ator: Usuário anonimo

Descrição: Esse caso descreve o procedimento para cadastrar pessoas físicas ou jurídicas interessadas em veicular anúncio no site.

Pré Condições: Acessar a página principal do site.

Cenário Principal:

Usuário seleciona a opção “Programas de publicidade”

Usuário preenche os seguintes campos do cadastro de anunciante.

1.Nome do usuário

2.Senha

3.Confirmar senha

4.E-mail.

Usuário clica em “Criar Usuário”

Cenários Alternativos:

1.Uusuário informa um nome do usuário já cadastrado

1.1 Sistema envia alerta informando solicitando para digitar um nome de usuário diferente.

2. Usuário informa uma senha com menos de 5 caracteres.

2.1 Sistema envia alerta informando que a senha deve ter no mínimo cinco caracteres.

Pós Condições: O anunciante é cadastrado no sistema.

Regras de Negócio: A senha pode ser alfanumérica e deve conter no mínimo cinco caracteres.

Caso de uso: UC014 – Cadastrar anúncio.

Ator: Anunciante.

Descrição: Esse caso de uso descreve o procedimento para cadastrar anúncios no site.

Pré Condições: Anunciante deve estar logado no sistema – UC003.

Cenário Principal:

1. Anunciante clica em “Meus Anúncios”

2. Sistema apresenta página com os anúncios cadastrados.
3. Anunciante clica em “Inserir”
4. Sistema apresenta formulário de cadastro de anúncios.
5. Usuário preenche o campo “Descrição”.
6. Usuário informa o valor da mercadoria anunciada.
7. Usuário informa o valor máximo da campanha.
8. Usuário informa a URL que o sistema deve redirecionar o usuário ao clicar no anúncio.
9. Usuário informa a data de validade do anúncio.
10. Usuário informa um texto descritivo do anúncio.
11. Usuário clica em confirmar.

Cenários Alternativos:

5. Usuário não preenche o campo descrição.
 - 5.1 Sistema envia alerta informando que a descrição deve ser preenchida.
6. Usuário informa um valor inválido ou não informa uma valor.
 - 6.1 Sistema envia alerta informando que o valor deve ser preenchido ou é inválido.
7. Usuário não informa o valor máximo da campanha, ou informa um valor inválido.
 - 7.1 Sistema envia alerta informando que o valor deve ser informado ou é inválido.

Pós Condições: O anúncio é cadastrado no sistema.

Regras de Negócio: O perfil do publico alvo esta relacionado com o valor total das listas de compras cadastradas. Sendo publico classe A os que tiverem uma lista com valor maior ou igual a 1500,00 reais, classe B entre 800,00 a 1500,00 e classe C menor que 800,00.

Caso de uso: UC015- Alterar anúncios.

Ator: Anunciante.

Descrição: Esse caso de uso descreve o procedimento para o anunciante alterar um anúncio cadastrado.

Pré Condições: Anunciante deve estar logado no sistema – UC003.

Cenário Principal:

1. Anunciante clica em “meus anúncios”.
2. Sistema apresenta lista de anúncios ativos e inativos.
3. Anunciante clica em alterar anúncios.
4. Usuário seleciona o anúncio que deseja alterar e clica em alterar.

5. Sistema apresenta formulário de alteração de anúncios.
6. Anunciante altera os dados.

Pós Condições: O anúncio é alterado no sistema.

Caso de uso: UC016 – Excluir anúncio.

Ator: Anunciante.

Descrição: Esse caso de uso descreve o procedimento para excluir um anúncio cadastrado.

Pré Condições: Anunciante deve estar logado no sistema – UC003.

Cenário Principal:

1. Anunciante clica em “meus anúncios”.
2. Sistema apresenta lista de anúncios ativos e inativos.
3. Anunciante seleciona o anúncio que deseja remover.
4. Sistema apresenta tela de confirmação de remoção do anúncio.
5. Usuário confirma a remoção do anúncio.

Pós Condições: O anúncio é excluído do sistema.

Caso de uso: UC017 – Acompanhar anúncios.

Ator: Anunciantes.

Descrição: Esse caso de uso descreve o procedimento para o anunciante acompanhar seus anúncios ativos, isso é, verificar o número de cliques recebidos e o valor atual da campanha.

Pré Condições: Anunciante deve estar logado no sistema – UC003.

Cenário Principal:

Anunciante seleciona a opção acompanhar anúncios.

Sistema lista os anúncios ativos e sua respectiva quantidade de cliques e o valor atual da campanha.