



**Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"**

MATEUS BERNARDO

**DESENVOLVIMENTO DE UM WEBSITE PARA REVENDEDORA DE
AUTOMÓVEIS UTILIZANDO ASP.NET MVC**

**ASSIS
2010**



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

DESENVOLVIMENTO DE UM WEBSITE PARA REVENDEDORA DE AUTOMÓVEIS UTILIZANDO ASP.NET MVC

Trabalho de Conclusão de Curso apresentado ao
Instituto Municipal de Ensino Superior de Assis,
como requisito do Curso de Graduação

Orientador: Professor Osmar Aparecido Machado.

Analizador: Professora Rita de Cássia C. Lopes.

**ASSIS
2010**

FICHA CATALOGRÁFICA

BERNARDO, Mateus

Desenvolvimento de um Web Site para revendedora de automóveis utilizando Asp.Net MVC / Mateus Bernardo. Fundação Educacional do Município de Assis – Fema

Assis, 2010. 47p.

Trabalho de Conclusão de Curso (TCC) – Tecnologia em Processamento de Dados
– Instituto Municipal de Ensino Superior de Assis

1. Introdução. 2. Histórico da Internet. 3. Tecnologias utilizadas no projeto. 4. Orientação a objetos 5. Detalhamento do Projeto. 6. Documentação do Sistema. 6 Conclusões. 7 Referências bibliográficas.

Biblioteca da FEMA

MATEUS BERNARDO

**DESENVOLVIMENTO DE UM WEBSITE PARA REVENDEDORA DE
AUTOMÓVEIS UTILIZANDO ASP.NET MVC**

Trabalho de Conclusão de Curso apresentado ao
Instituto Municipal de Ensino Superior de Assis,
como requisito do Curso de Graduação,
analisado pela seguinte comissão examinadora:

Professor Osmar Aparecido Machado
Orientador

Professora Rita de Cássia Cassiano Lopes
Analisador

DEDICATÓRIA

Dedico este trabalho:

À Deus, por ter me dado a oportunidade de viver e crescer a cada dia.

Aos meus pais , pelo apoio e carinho oferecidos em todos os momentos de minha vida. A minha esposa que sempre me incentivou e a meus filhos, que são minha grande alegria.

RESUMO

Este trabalho tem como objetivo mostrar o desenvolvimento de um website site utilizando as tecnologias Asp.Net MVC, LINQ com SQL Server. No desenvolvimento deste sistema foram usadas estas tecnologias e mostrando o resultado final através de um sistema totalmente web.

Neste projeto foi possível experimentar e aprimorar o seu uso. Em síntese, este projeto poderá ser utilizado para a criação de um novo negócio, além de disponibilizar para os estudantes, alguns conceitos úteis sobre as tecnologias aqui apresentadas, bem como no desenvolvimento desta aplicação.

Palavras-chave: Internet, Programação web, Web site

ABSTRACT

This paper aims to show the development of a web site using technologies Asp.Net MVC, LINQ to SQL Server. In developing this system were used for these technologies and showing the final result through a fully web.

In this project it was possible to try and improve their use. In summary, this project could be used to create a new business, in addition to providing for students, some useful concepts about the technologies presented here, as well as the development of this application.

Keywords: Internet, Web Programming, Web site

LISTA DE FIGURAS

01. Redes Sociais.....	15
02. Exemplo de código LINQ.....	30
03. Tela principal do Sistema.....	37
04. Tela de cadastro de Novos veículos.....	38
05. Tela de cadastro de novas Marcas de Veículos.....	38
06. Estoque de veículos.....	39
07. Tela que lista os modelos de veículos cadastrados.....	40
08. Tela que lista as Marcas de veículos cadastradas.....	41
09. Diagrama do banco de dados.....	42
10. Diagrama de Classes.....	43
11. Use Case.....	44

SUMÁRIO

1. Introdução.....	11
1.1 Objetivos.....	11
1.2 Justificativas.....	12
2. HISTÓRICO DA INTERNET.....	13
2.1 Início da Internet.....	13
2.2 Evolução da Internet - Web 2.0.....	14
2.3 Redes Sociais	14
2.4 Blogs	15
3. TECNOLOGIAS UTILIZADAS NO PROJETO	17
3.1 XHTML.....	17
3.1.1 Vantagens de se usar XHTML.....	18
3.2 CSS.....	19
3.2.1 Acessibilidade do CSS.....	20
3.3 JavaScript.....	22
3.4 Framework JQuery.....	23
3.4.1 Utilização do Framework JQuery.....	24
3.5 Linguagem C#.....	25
3.6 Plataforma .NET (.NET Framework)	26
3.6.1 MSIL – Microsoft Intermediate Language.....	27
3.6.2 CLR – Common Language Runtime	27
3.6.3 JIT – Compilador de Tempo Real.....	28
3.7 LINQ.....	29
3.7.1 LINQ to SQL.....	31
3.8 O padrão MVC.....	31
3.9 Microsoft SQL Server.....	33
3.10 Conclusões.....	33

4. ORIENTAÇÃO A OBJETOS.....	34
4.1 Definições Básicas para Orientação a Objeto.....	34
4.2 Objeto.....	34
4.3 Classe.....	35
4.4 Abstração.....	35
4.5 Encapsulamento.....	36
6. DETALHAMENTO DO PROJETO.....	36
6.1 Diagrama do Banco de Dados.....	42
6.2- Diagrama das Principais Classes do Sistema.....	43
6.3 Use Case.....	44
7. CONCLUSÃO	45
8. REFERÊNCIAS BIBLIOGRÁFICAS.....	46

1. INTRODUÇÃO

A internet no Brasil e no mundo cresce a passos largos. Hoje em dia tanto as pessoas físicas quanto as jurídicas a utilizam como uma forma de se apresentar à comunidade, seja por meio de blogs, sites, loja virtual ou outras mídias disponíveis via web. O objetivo é atingir o maior número de pessoas, clientes, fornecedores ou simplesmente aumentar a rede de contatos.

Uma empresa que não possui sua página na internet nos tempos atuais certamente está perdendo negócios e deixando de conquistar novos clientes. Novos internautas surgem a cada segundo, o que torna a web um espaço de oportunidades para as organizações. Além disso, é um espaço democrático, todas as empresas, independentes de tamanho, região e tipo podem utilizá-la.

Segundo dados do Pyramid Research o mercado de internet vem há mais de uma década crescendo e se consolidando no Brasil e em outras partes do mundo. Segundo uma recente pesquisa realizada pelo órgão, estima-se que o mercado de publicidade on-line na América Latina deve registrar um volume de US\$ 2.6 bilhões no ano de 2013, o que representaria 9% do volume total de publicidade total, diante dos 2% atuais. Em 2008 esse volume foi de US\$ 549 milhões. O estudo também divulgou o crescimento dos negócios firmados na área do e-commerce que hoje registra cerca de US\$ 13 bilhões e espera-se um aumento de 33% até 2013. (www.pyr.com).

É na Internet que as pessoas encontram nomes, endereços, produtos, serviços e principalmente informações de seus parceiros. Se a empresa possui um bom site, é através dele que os clientes atuais e futuros podem ver a empresa, conhecer sua história, seus produtos e serviços, enfim, seu potencial e sua estrutura.

1.1 Objetivos

O projeto objetiva o desenvolvimento de um Website para revendas de automóveis, utilizando tecnologias modernas tanto no processo de desenvolvimento das interfaces quanto no armazenamento dos dados. O sistema deverá controlar estoques de veículos novos e usados, criar ambientes de relacionamentos com clientes, fornecedores e outros interessados na aquisição de veículos.

1.2 Justificativas

O site é uma apresentação da empresa, é quase um selo de qualidade. O número de usuários de internet no país hoje, é de aproximadamente 22 milhões e este número tende a aumentar. Ficar fora da Internet diante desta evolução pode representar a falência para uma empresa. No Brasil este fato é muito relevante pois os brasileiros superam os americanos em tempo na Internet. A média de uso da Internet no Brasil aproxima-se de 16 horas diárias. Ou seja, o homem, a mulher, o jovem, a criança, o consumidor, o fornecedor, o cliente, enfim, o público alvo está na Internet.

Investir na divulgação de produtos e serviços pela Internet pode ser extremamente vantajoso, pelos seguintes motivos:

- possibilidade de retorno imediato: Ao divulgar um produto no seu site aumentam as possibilidades de resultar em novas vendas, por meio de um simples clique. Diferente de um anúncio em revista, por exemplo, onde o resultado é visto em longo prazo ou através de pesquisa.
- O site bem estruturado pode possibilitar conhecer melhor seus visitantes por meio de formulários de contato, podem trazer informações importantes sobre seus clientes para atendê-los melhor, garantir satisfação e entrar em contato em outro momento para divulgação e apresentação da empresa.

Assim, este projeto pode ser uma contribuição importante, pois será mais uma ferramenta para ajudar as empresas, em especial do setor de revenda de automóveis, a explorar o potencial da web.

Além disso, o website poderá estreitar a relação da empresa com os consumidores em geral. Serão utilizadas diversas tecnologias para o desenvolvimento, como C#, Microsoft .NET Framework, Padrão de arquitetura MVC (Model-view-controller), o banco de dados Sql Server 2008 e LINQ (Language Integrated Query). Na camada de apresentação (Front-end) será utilizada as tecnologias: Xhtml, Framework JQuery e CSS 2.0. Em termos operacionais o sistema deverá controlar os estoques de veículos novos e usados, bem como criar ambientes de relacionamentos entre clientes, fornecedores e interessados na aquisição de veículos.

2. HISTÓRICO DA INTERNET

2.1 Início da Internet

A internet teve seu início ainda no final dos anos 50, quando a União Soviética lançou, à frente dos EUA, o primeiro satélite na órbita terrestre, o Sputnik 1. Este evento deixou claro a necessidade de uma organização que pudesse contribuir para a disseminação das pesquisas e desenvolver idéias e tecnologias, assim, pouco tempo depois foi criada a ARPA (Agência de Projetos de Pesquisa Avançada) do Departamento de Defesa dos Estados Unidos, que pouco tempo depois transformou-se na internet.

Um dos responsáveis pela criação da ARPA foi o cientista de computação e psicólogo Joseph Licklider, que em 1960 publicou um documento intitulado “Relação Homem-Computador” em que lançava a idéia de computadores em rede, fornecendo armazenamento e consultas de informações. Surgiu assim em outubro de 1967 o ARPANET, e em dezembro de 1969 uma primeira rede com 4 computadores entrou em funcionamento.

O desafio a ser vencido para se criar uma rede, era como se conectar redes físicas separadas sem aumentar os recursos de rede para links constantes. A “troca de pacotes” foi a técnica utilizada para solucionar o problema, utilizada pela internet até os dias atuais. Esta técnica caracteriza-se pela requisição de dados divididos em pequenos pedaços (pacotes) de forma que pudessem ser processados rapidamente sem bloquear a comunicação de outras partes.

Antes porém de se transformar na internet na forma como é conhecida atualmente, ocorreram algumas etapas importantes na evolução das redes. Pode-se citar como exemplo o surgimento do protocolo X25, desenvolvido pela União Internacional de Telecomunicações, que formou as bases da primeira rede universitária do Reino Unido, chamada de JANET (Conjunto de redes acadêmicas) que permitiu às universidades britânicas enviarem e receberem arquivos de emails. A CompuServe, uma rede pública americana, também foi importante, pois permitiu que pequenas

empresas e indivíduos pudessem acessar recursos computacionais por um tempo compartilhado e depois, o acesso à internet.

Entretanto, o aspecto mais importante para que estas redes viessem a se transformar na internet, foi a criação do protocolo de comunicação denominado TCP/IP (ver sigla) que, em síntese, estabeleceu um padrão dentre os diversos protocolos de comunicação em rede. Ele tornou possível a comunicação simultânea entre diferentes redes e em 1982, as conexões da ARPANET para fora dos EUA foram convertidas para usar o novo protocolo “TCP/IP”. Foi a chegada da Internet como se conhece nos dias atuais.

2.2 Evolução da Internet - Web 2.0

Ao longo das últimas décadas a internet evoluiu muito e atualmente encontra-se na segunda geração da Word Wide Web, denominada de Web 2.0. Ela representa a idéia de um ambiente on-line dinâmico, em que a troca de informações e colaboração entre os internautas com sites, blogs e serviços virtuais se tornam cada vez mais comum.

Blogs, Twitter e Redes Sociais estarão cada vez mais presentes, formando um novo cenário, uma nova internet. O usuário que antes era somente espectador de conteúdos, passará a produzir e divulgar o seu próprio conteúdo e a construir sua identidade na rede.

2.3 Redes Sociais

As redes sócias online são, em muitos de seus recursos, comuns para cada um dos mais de 300 sites de rede social existentes atualmente. Criar e compartilhar um perfil pessoal são os recursos mais básicos das redes sociais. Essas páginas de perfil normalmente permitem a inserção de uma foto, algumas informações pessoais básicas (nome, idade, sexo, local) e alguns detalhes a mais que a pessoa deseja compartilhar com os internautas que visitam seu perfil.

Além disso, maioria das redes sociais na Internet permite postar fotos, vídeos e blogs pessoais na sua página de perfil. Entretanto, o recurso mais importante das redes sociais online é encontrar e fazer amigos com membros de outros sites. Na página de perfil, esses amigos aparecem como links e os visitantes podem navegar facilmente pela sua rede de amigos online. Cada rede social online possui regras e métodos diferentes de busca e contato com amigos potenciais.

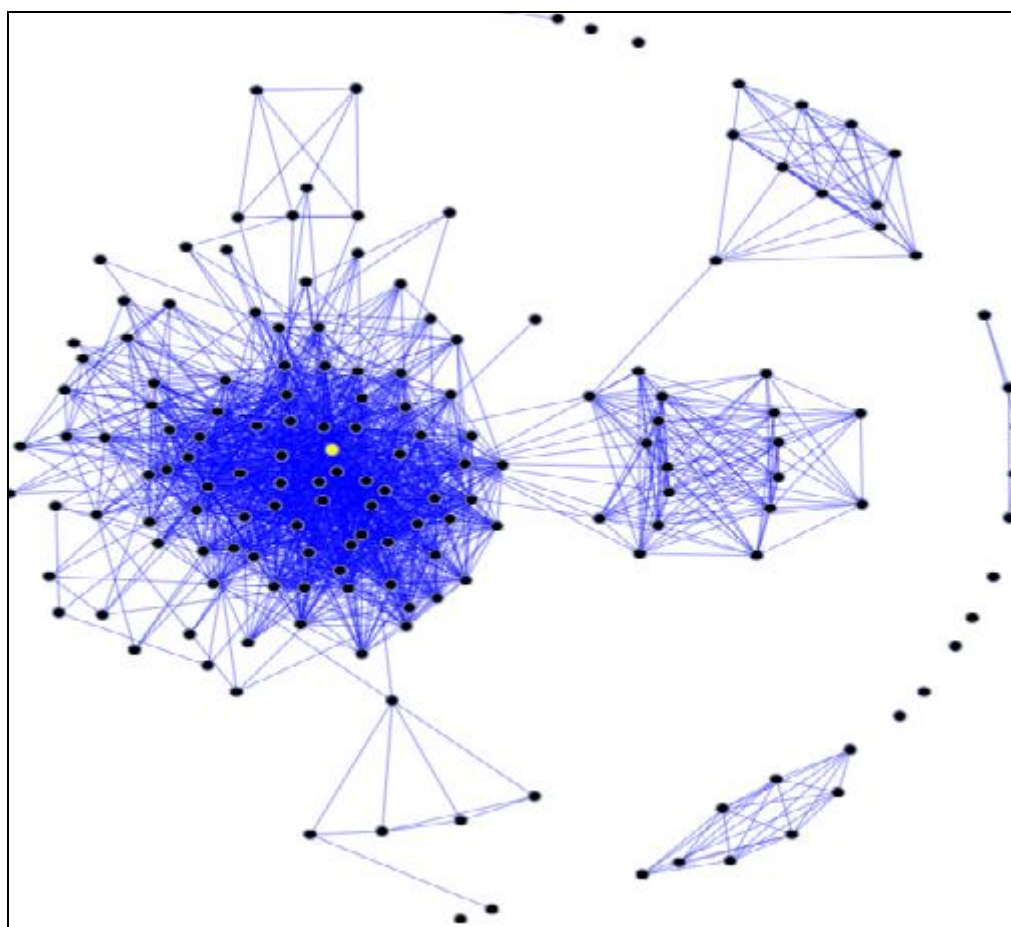


Figura 2.1 – Redes Sociais

Fonte: Extraído http://pt.wikipedia.org/wiki/Ficheiro:Sna_large.png

2.4 Blogs

Um blog (uma mistura do termo web log) é um tipo de site ou parte de um website, geralmente mantidos por um indivíduo, um usuário da internet, com entradas regulares de comentários, descrições de eventos ou outros materiais, como gráficos

ou vídeos. As inscrições são comumente exibidas em ordem cronológica inversa. O termo “Blog” também pode ser usado como um verbo, significando manter ou adicionar conteúdo de um blog.

A maioria dos blogs são interativos, permitindo que os visitantes deixem comentários e até mesmo de mensagens entre si através de widgets em blogs e é esta interatividade que os distingue de outros sites estáticos.

Qualquer internauta pode criar o seu próprio blog, por isso existe grande multidisciplinaridade entre os blogs. Muitos fornecem comentários ou notícias sobre um determinado assunto, enquanto outros funcionam mais como diários pessoais online. Um blog típico combina texto, imagens e links para outros blogs, páginas da Web e outros meios relacionados a seu tópico. Os leitores geralmente podem participar dos blogs que visitam por meio do registro de comentários. A maioria dos blogs são primariamente textuais, embora existam alguns com foco na arte (blog de Arte), fotografia (photoblog), vídeos (video blogs), música (MP3 blog) e áudio (podcasts). Os Microblogging são um subtipo dos blogs, com mensagens muito curtas.

A internet, como se pode observar neste capítulo, ampliou a capacidade dos indivíduos se relacionarem no mundo atual. Esta capacidade extrapolou das universidades para a sociedade organizada, para as corporações e hoje chega a todos os indivíduos na maioria dos países do mundo.

Como todas as tecnologias existem aspectos favoráveis e contrários a serem considerados e por conta dos objetivos deste trabalho, não serão aqui abordados. Desta forma, na próxima seção, serão abordadas as tecnologias utilizadas no processo de desenvolvimento deste trabalho.

3. TECNOLOGIAS UTILIZADAS NO PROJETO

Para o desenvolvimento deste projeto será necessário o uso de diversas tecnologias trabalhando de forma integrada. A integração entre diversas tecnologias que se complementam é um dos principais aspectos no desenvolvimento de software na atualidade, especialmente softwares para a WEB.

Desta forma, nesta seção, serão apresentadas as tecnologias utilizadas, seus conceitos e ferramentas.

3.1 XHTML

XHTML foi criado para ser uma linguagem em conformidade com o XML e ao mesmo tempo compatível com o HTML 4 onde pode ser escrito para ser compatível com ambas as aplicações. O HTML é um tipo de aplicação que a W3C chama de SGML (Standard Generalize Markup Language) – Linguagem de Marcação Padrão Generalizada) que é uma linguagem para descrever linguagem de marcação específica para publicação eletrônica de documentos.

Ela começou a ser utilizada na década de 80 e originalmente destinada a ser uma linguagem destinada a criação de documentos científicos e técnicos adaptada para ser utilizada por não especialistas. Ou seja, o HTML contorna o problema da complexidade do SGML padronizando um conjunto de tags (etiqueta, marcador) específicas com características estruturais e semânticas padronizadas, simplificando a sua utilização.

Um documento XHTML é um documento que tem de ser bem-formatado de acordo com as regras do XML cada tag tem de ser fechado e corretamente aninhado e para as tags como img, input, link etc, uma barra estreita rápida deve ser adicionado no final da marca, como esta: `<input type="text" />` .

Exemplo de uso do XHTML:

Este trecho de código HTML monta uma pequena estrutura de uma página HTML e exemplifica como pode ser desenvolvido um Menu.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Titulo do Site</title>
  <link href="/Default/style.css" rel="stylesheet" type="text/css" />
  <script src="/Scripts/jquery-1.3.2.min.js" type="text/javascript"></script>
</head>
<body>
  <div id="container">
    <div id="menu">
      <ul>
        <li><a href="#">Loren</a></li>
        <li><a href="#">Loren</a></li>
        <li><a href="#">Loren</a></li>
        <li><a href="#">Loren</a></li>
      </ul>
    </div>
    <div id="content">
      <h1> Exemplo </h1>
    </div>
  </div>
</body>
</html>
```

3.1.1- Vantagens de se usar XHTML

Existem inúmeras vantagens de se usar XHTML, tanto em novos documentos web ou documentos importados, migrados de outras aplicações, ou ainda, dos atuais escritos em HTML. Dentre as vantagens, destaca-se em primeiro plano a compatibilidade da linguagem XHTML com as futuras aplicações, garantindo que as criações XHTML serão conservadas estáveis por longos anos. A tendência é a de que futuras versões de browsers e agentes de usuários em geral, deixem de suportar elementos e atributos já em desuso ("deprecated") segundo as recomendações do W3C, bem assim como antigos e ultrapassados esquemas e esboços do HTML.

XHTML é um código consistente que dispensa uso de "truques" e "hacks" para contornar "bugs". Editar um código XHTML existente é uma tarefa bem simples por se tratar de uma escrita limpa e evidente. O tempo de carregamento de uma página XHTML é mais rápido, pois os browsers interpretam uma página limpa sem ter que interpretar e decidir sobre renderização de erros de código.

Uma página XHTML é mais acessível aos browsers e aplicações de usuário padrão incrementando a interoperabilidade e a portabilidade dos documentos web. É totalmente compatível com todas as aplicações de usuários para HTML, antigas e já ultrapassadas.

Neste projeto o XHTML será fundamental para construção da camada de apresentação do site, o site será estruturado usando XHTML como linguagem de marcação texto. O site vai ter uma interface bem amigável tanto para os usuarios quanto para os administradores, vou criar Menus, links, listas utilizando os recursos que o XHTML nos permite.

3.2 - CSS

Cascading Style Sheets ou simplesmente CSS, é uma linguagem de estilo utilizada para definir a apresentação de documentos escritos em uma linguagem de marcação, como HTML ou XML. Seu principal benefício é prover a separação entre o formato e o conteúdo de um documento.

Ao invés de colocar a formatação dentro do documento, o desenvolvedor cria um link (ligação) para uma página que contém os estilos, procedendo de forma idêntica para todas as páginas de um portal. Quando quiser alterar a aparência do portal basta portanto, modificar apenas um arquivo.

Talvez a maior vantagem de uso do CSS é a capacidade de separar a marcação HTML da apresentação do site. Em outras palavras, vale dizer que o HTML destina-se unicamente a estruturar e marcar o conteúdo (marca e estrutura textos, cabeçalhos, parágrafos, links, botões, formulários, imagens e demais elementos da página) ficando por conta das CSS toda a responsabilidade pelo visual do

documento, como definição de cores, posicionamento na tela, estilos de linhas, bordas e tudo o mais relacionado à apresentação.

A adoção desta técnica resulta em múltiplas vantagens e benefícios que não são conseguidos quando se utiliza o conceito largamente empregado para construção de web sites, baseado no uso de tabelas e de atributos de estilização dentro das tags HTML.

3.2.1- Acessibilidade do CSS

Em termos de acessibilidade, as CSS possibilitam separar a estrutura dos documentos de sua apresentação. Isto é possível porque as folhas de estilo foram projetadas para propiciar um controle preciso - independente da marcação - do espaçamento de caracteres, alinhamento de texto, posicionamento de objetos na página, saídas de áudio e voz, características de fontes, etc. Separando os estilos da marcação os autores podem simplificar e limpar o HTML nos seus documentos, tornando-os mais acessíveis.

As CSS permitem um controle preciso de espaçamento, alinhamento e posicionamento. Os autores podem evitar o "uso incorreto de tags" - a prática em desuso de usar a marcação para obter efeitos de apresentação. Por exemplo, muito embora os elementos do HTML, BLOCKQUOTE e TABLE tenham a finalidade de marcar citações em bloco e apresentar dados tabulares respectivamente, eles são frequentemente usados para obter efeitos visuais de endentação e posicionamento. Softwares especiais tais como os leitores de tela, encontrarão grande dificuldade em tratar os elementos assim usados, podendo chegar a resultados ininteligíveis para o usuário.

Complementarmente a prevenção de mal uso de elementos HTML, as folhas de estilo podem também ajudar a evitar o mal uso de imagens. É comum a prática de os autores utilizarem imagens invisíveis de 1-pixel para posicionar conteúdo. Isto não só sobrecarrega o documento aumentando o tempo para download como também confunde os softwares de agentes de usuários que buscam por um texto

alternativo (o atributo "alt") para as imagens. As propriedades CSS de posicionamento tornaram o uso de tais imagens completamente desnecessárias.

As CSS permitem um controle preciso de tamanhos de fontes, cores e estilos. Alguns autores se utilizam de imagens para representar textos em fontes que provavelmente não estão disponíveis na máquina do usuário. Textos passados através de imagens não estão acessíveis aos softwares especiais de agentes de usuários, tais como os leitores de tela, e nem podem ser indexados por robots de busca. Para corrigir esta situação um poderoso mecanismo de WebFonts em CSS permite ao usuário um controle total sobre as fontes no lado do cliente. Com as WebFonts, os autores podem se despreocupar com fontes não disponíveis na máquina do usuário.

Neste projeto o CSS será responsável para formatar a camada de apresentação, a grande maioria do elementos XHTML receberão um ID ou uma classe para apresentar-se com um bom visual no navegador. No CSS será definida todas as fontes, cores, margens espaçamentos, imagens de fundo e alinhamentos que o site irá ter, o CSS também será responsável por manter o site rodando em todos os Browsers disponíveis hoje.

Exemplo de uso de estilos CSS:

Este trecho de código CSS mostra como fica um Estilo CSS para formatar uma marcação HTML, nele estão algumas propriedades que servirão para dar cor, margem, fonte, etc., à estrutura de uma página

```
#content {
    clear: both;
    float: none;
    width: 980px;
    height: auto;
    padding-top: 6px;
    padding-right: 11px;
    padding-left: 33px;
    margin: 0 auto 10px auto;
    background: #ccc
}
#main {
    width: 980px;
    height: auto;
```

```

        color: #000
        padding: 10px
        font-size:14px;
        font-weight:bold
        line-height: 1.15em;
        position:absolute;
        top:0;
        left:0
    }

#rodape {
    float: left;
    position:relative;
    display: block;
    width: 980px;
    height: 176px;
    color: #333
    margin: 0 auto;
    padding: 10px
    font-size:14px;
    font-weight:bold
}

```

3.3 JavaScript

JavaScript é uma linguagem de programação criada pela Netscape em 1995, que em princípio se chamava LiveScript, a Netscape após o sucesso inicial desta linguagem, recebeu uma colaboração considerável da Sun, que praticamente tornou o JavaScript compatível com a linguagem Java, por esta razão, a semelhança dos nomes "JavaScript".

A linguagem foi criada para atender, principalmente, algumas necessidades como: Validação de formulários no lado cliente (programa navegador), Interação com a página. Sua união com o CSS é conhecida como DHTML. Usando o Javascript, é possível modificar dinamicamente os estilos dos elementos da página em HTML.

Dada sua enorme versatilidade e utilidade ao lidar com ambientes em árvore (como um documento HTML), foi criado a partir desta linguagem um padrão ECMA, o ECMA-262, também conhecido como ECMAScript.

O javascript pode funcionar como uma linguagem orientada a objetos, depois do objeto construído, pode ser usado como um modelo (ou protótipo) para a criação de objetos similares. O principal benefício do Javascript é a de adicionar interação adicional entre o site e os seus visitantes. Por definição, Javascript é uma linguagem

de scripting do lado do cliente. Isso significa que o navegador do internauta irá executar o script. Isto é oposto do lado do servidor, que pode ocorrer em uma outra linguagem como PHP.

Neste projeto o JavaScript será bastante útil para validar os formulários do site, muitas funções para ajudar a dinâmica do site serão feitas com funções javascript.

3.4 Framework JQuery

Jquery é uma biblioteca JavaScript criada por John Resig e disponibilizada como software livre e aberto, ou seja, de emprego e uso regido segundo licença conforme as regras estabelecidas pelo MIT – Massachusetts Institute of Technology ou pelo GPL – GNU General Public License.

Segundo o Criador deste framework - John Resig o foco principal da biblioteca jQuery é a simplicidade. Porque submeter os desenvolvedores ao martírio de escrever longos e complexos códigos para criar simples efeitos? e, sem dúvida, John Resig estava em um momento de rara inspiração quando assim escreveu, pois isso muito bem resume jQuery: uma maneira simples e fácil de escrever JavaScript colocada ao alcance não só de programadores experientes como também de designers e desenvolvedores com pouco conhecimento de programação.

Simplicidade é a palavra-chave que resume e norteia o desenvolvimento com jQuery. Linhas e mais linhas de programação JavaScript escritas para obter um determinado efeito em uma página são substituídas por apenas uma dezena de linhas escritas com sintaxe jQuery.

Neste projeto o JQuery será bastante utilizado para ajudar carregar dados do servidor sem tenha a necessidade de dar um refresh na página, vou utilizar alguns recursos de Ajax que o JQuery dispõe.

3.4.1 Utilização do Framework JQuery

Segundo consta no web site oficial <http://jquery.com/>, “jQuery is designed to change the way that you write JavaScript”, ou seja, jQuery é projetado para mudar a maneira de escrever javascript. jQuery é disponibilizado das diferentes maneiras:

Desenvolvimento: Apresenta todos seus códigos completos, endentados e de fácil entendimento. Ideal para quem queira estudar e compreender como as coisas funcionam, essa “versão sem cortes” é ideal para estudo/aprendizado a fundo da biblioteca.

Produção: Esta é a versão para quem se preocupa somente em utilizar o jQuery e usufruir de todo o seu poder através de um arquivo compactado, ideal para aqueles que querem usar o framework rapidamente, evitando transferências desnecessárias e tempos de carregamento excessivos.

Google API/CDN: Através da API AJAX de bibliotecas do Google é possível, dentre outros frameworks e bibliotecas, incorporar o jQuery a suas páginas web. Quando se usa jQuery, realmente a maneira de se escrever javascript muda. É o fim das enormes linhas que servem somente para identificar um elemento; o fim dos laços de repetição que somente identificam quantos elementos de determinada categoria estão presentes no documento.

Exemplo de uso do XHTML:

Com pouquíssimas linhas de código, é possível mexer à vontade com o DOM (Document Object Model) (identificação, edição, inserção, deleção, etc), manipular eventos javascript, alterar regras CSS, aplicar efeitos visuais (fade in / fade out, drag and drop, slides, movimentos, etc)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Titulo do Site</title>
  <script src="/Scripts/jquery-1.3.2.min.js" type="text/javascript"></script>

  <style>

  p { margin:8px; font-size:16px;}
  .selected {color:red; }
  .highlight {background:yellow;}
```



```
</style>

<script>$("p:last").addClass("selected highlight");</script>

</head>
<body>
  <p>Hello</p>
  <p> and</p>
  <p> Goodbye</p>
</body>
</html>
```

Resultado do código acima:



Figura 5.3 - Exemplo de código JQuery
Fonte: Extraído do site JQuery.com

3.5 Linguagem C#

A linguagem de programação C# deriva do C e C++, simplificando (e modernizando) o C++ nas áreas de classes, espaço de nomes, sobrecarga de métodos e gerenciamento de exceções (Christian, 2008). Muito da complexidade do C++ foi removida do C#, como a utilização de ponteiros, a fim de tornar o C# mais fácil de ser utilizado e menos propenso a erros.

Segundo Wille (Christian, 2001), o C# é uma linguagem:

- Simples: possui diversos recursos; por padrão se trabalha com código gerenciado; não existem as redundâncias encontradas no C++, como diferentes tipos de caracteres, dentre outras;

- Moderna: além de ser projetada para ser a linguagem principal de escrita de aplicativos .NET, possui recursos que estavam indisponíveis em C++ implementados, além de possuir maior segurança que seus antecessores;
- Orientada a Objetos: não existem mais funções globais, variáveis ou constantes, tudo deve ser encapsulado dentro de uma classe, tornando o código C# mais legível, reduzindo conflitos de nomeação, além de possuir suporte a herança, polimorfismo, encapsulamento e abstração;
- Type-safe: o C#, juntamente com coletores de lixo, implementa a segurança de tipo mais estrita, impedindo que erros graves, como, por exemplo, atribuir um ponteiro int* para um double*;
- Compatível: é possível acessar API's diferentes, através do CLS.
- Flexível: possibilita a incorporação de aplicações C# em aplicações já existentes, em C, por exemplo.

Os recursos citados fazem do C# uma linguagem fácil de aprender e de usar, robusta e com boa performance. Em conjunto com os demais recursos da arquitetura .NET .

Neste projeto o C# será a linguagem de programação adotada para fazer a programação Server-side

3.6 Plataforma .NET (.NET Framework)

A plataforma .NET surgiu em Julho de 2000, sendo criada para ser um padrão de desenvolvimento para web, possibilitando a utilização de diferentes linguagens de programação na construção de softwares, sendo as linguagens de maior destaque o C#, VB.NET, Jscript.NET e ASP.NET (Conery, 2008).

A proposta desta plataforma é proporcionar um ambiente de desenvolvimento avançado, disponibilizando uma grande quantidade de recursos para uso dos desenvolvedores. Para isso, a Microsoft unificou todas as suas soluções de desenvolvimento nessa nova plataforma, além de melhorar bastante os recursos oferecidos. Pode-se dizer que o .NET Framework disponibiliza um ambiente de

desenvolvimento multi-plataforma, multi-linguagem, orientada a objetos, e com uma grande e eficiente biblioteca de classes .

A plataforma de desenvolvimento .NET contém uma biblioteca de classes nativas, que incluem classes de acesso e manipulação de dados, classes que contém objetos visuais (Windows forms), classes de manipulação de informações transmitidas pela WEB (ASP.NET), classes de acesso ao sistema, entre outros.

3.6.1 MSIL – Microsoft *Intermediate Language*

O conceito de abstração, utilizado para ocultar a complexidade de sistemas oferecendo ao usuário uma interface simples, é extremamente importante na engenharia de software. O bytecode gerado pelo compilador do Java é uma forma de abstração que permite que o programa seja executado em diferentes sistemas operacionais, desde que a máquina virtual do Java (JVM) esteja instalada (Conery, 2008).

A MSIL é a forma de abstração encontrada pela Microsoft. A MSIL é semelhante ao bytecode do Java, suportando orientação a objetos, herança, polimorfismo e exceções. Além destes oferece também o suporte a múltiplas linguagens.

3.6.2 CLR – Common Language Runtime

O CLR tem a função de gerenciar e executar os programas, realizando tarefas como incorporação de objetos, verificações de segurança, execução, gerenciamento de memória e coleta de lixo de todos os códigos escritos em qualquer uma das linguagens utilizadas na plataforma dot NET. O CLR está dividido em três sub-componentes (Conery, 2008):

1. CTS (*Common Type Specification*), que especifica os tipos que uma linguagem tem que suportar para que o CLR possa compilar e executar um programa escrito nessa linguagem;

2. CLS (*Common Language Specification*), que é o conjunto de diretivas que devem ser seguidas por uma aplicação ou implementação dot NET para ser compatível com a plataforma;
3. motor de execução, que gerencia as referências a objetos, coleta de lixo, verificação de acesso, métodos seguros e não seguros.

3.6.3 JIT – Compilador de Tempo Real

Uma aplicação dot NET diferencia-se das aplicações tradicionais somente no método de compilação. Ao invés de criar um código executável baseado nas instruções de máquina da arquitetura em que está sendo compilado, a plataforma dot NET utiliza um conjunto de classes e métodos próprios da plataforma igual para todas as linguagens suportadas. Desse modo existe uma integração entre as linguagens sendo possível derivar uma classe C# de um objeto escrito em VB .NET (Conery, 2008).

O arquivo em linguagem intermediária recebe o nome de assembly, e na plataforma .NET é entendido como uma coleção de um ou mais arquivos com tipos e recursos que formam uma unidade funcional lógica contendo dados especiais denominados manifesto. O manifesto define que arquivos fazem parte do programa, que permissões são necessárias para ser executado, qual a versão do assembly e outras informações.

Existem quatro tipos de assembly na plataforma dot NET (Conery, 2008):

1. Estáticos: são os arquivos criados com o compilador específico da linguagem utilizada;
2. Dinâmicos: são os arquivos criados dinamicamente através das classes localizadas no namespace System.Reflection;
3. Privados: são assembly estáticos usados em uma aplicação específica;
4. Públicos ou compartilhados: são assembly estáticos que tem um único nome público e pode ser usado por qualquer aplicação.

A execução do assembly se dá através de um compilador de tempo real (JIT), responsável por converter as instruções em MSIL contidas no arquivo em instruções da arquitetura alvo. O funcionamento, de forma básica, dos *JITers* ocorre da seguinte maneira:

- Quando uma classe é carregada, um ponteiro para cada método dessa classe é criado;
- Na execução de um desses métodos, o JIT compila apenas o trecho relativo ao método, criando um ponteiro para o código nativo gerado;
- Esse novo ponteiro permanecerá em *cache* e as chamadas subseqüentes ao mesmo método utilizarão o código já compilado.

3.7 – LINQ

LINQ é um conjunto de ferramentas que podem ser usados para trabalhar com objetos, como documentos XML, bases de dados relacionais ou outros tipos de dados. O LINQ É também uma extensão para linguagens de programação como C# e VB.NET. Ele unifica o acesso aos dados, independentemente da fonte de dados e permite consultas e operações de conjunto. A sintaxe do LINQ tem algumas semelhanças com o SQL, mas funciona de modo inverso, por exemplo, ele utiliza primeiro o “from” e só depois o “select”. Um dos aspectos-chave do LINQ é que ela foi projetada para ser usada em qualquer tipo de objeto ou fonte de dados e para fornecer um modelo de programação consistente. (MARGUERIE, 2008).

Além disso, o LINQ permite acessar as informações por escrito, consultas em diversas fontes de dados. A sintaxe permite facilmente incluir consultas de banco de dados diretamente no código C#, LINQ fornece o mesmo tipo de recursos expressivos que oferece SQL, mas em qualquer linguagem e programação.

```

var contacts = ← Retrieve customers from database
    from customer in db.Customers
    where customer.Name.StartsWith("A") && customer.Orders.Count > 0
    orderby customer.Name
    select new { customer.Name, customer.Phone };

var xml = ← Generate XML data from list of customers
    new XElement("contacts",
        from contact in contacts
        select new XElement("contact",
            new XAttribute("name", contact.Name),
            new XAttribute("phone", contact.Phone)
        )
    );

```

Figura 5.1 - Exemplo de código C # com LINQ
 Fonte: Extraído de LINQ in Action

A Figura 5.1 demonstra como realizada a extração de dados de um banco de dados e a criação de um documento XML a partir dele.

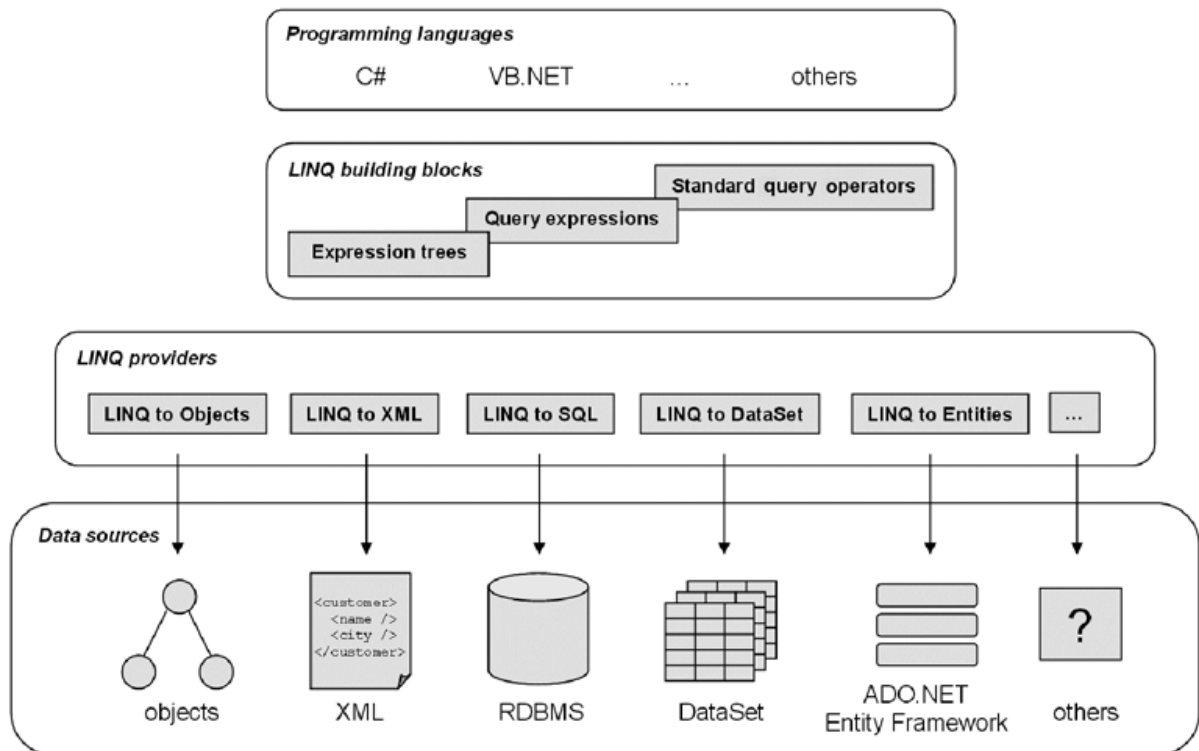


Figura 5.2 Fontes de dados que podem ser consultados usando LINQ
 Fonte: LINQ in Action

3.7.1 LINQ to SQL

LINQ to SQL é uma parte da LINQ que possibilita o acesso à bases de dados SQL Server. Possuindo um servidor SQL pode-se, através do LINQ to SQL, mapear toda a estrutura de uma base de dados, trabalhar os dados relacionais como sendo objetos, através da linguagem C#. Ele não utiliza um mecanismo próprio de busca para efetuar as consultas e comandos na base de dados, em vez disso, converte os comandos LINQ em consultas SQL, que então são enviadas para o servidor SQL para processamento. Uma vez que o SQL Server armazena os dados como em formato relacional e o LINQ trabalha com os dados encapsulados em objetos.

O LINQ To SQL fornece ainda, uma maneira de criar um ORM (objeto Modelo Relacional) ou seja (mapear) os dados relacionais para os objetos da aplicação e vice-versa.

3.8 - O padrão MVC

O Model-View-Controller (MVC) como é conhecido atualmente, foi um importante padrão arquitetônico em Ciência da Computação por muitos anos. Originalmente chamado de Thing-Model-View-Editor, em 1979, foi mais tarde simplificado para Model View Controller.MVC e encontrado em Java, C++, PHP e Ruby (CONERY, 2009)

A arquitetura MVC - (Modelo Visualização Controle) fornece uma maneira de dividir a funcionalidade envolvida na manutenção e apresentação dos dados de uma aplicação. A arquitetura MVC não é nova e foi originalmente desenvolvida para mapear as tarefas tradicionais de entrada, processamento e saída para o modelo de interação com o usuário. Usando o padrão MVC fica fácil mapear esses conceitos no domínio de aplicações Web multicamadas.

Na arquitetura MVC o modelo representa os dados da aplicação e as regras do negócio que governam o acesso e a modificação dos dados. O modelo mantém o estado persistente do negócio e fornece ao controlador a capacidade de acessar as funcionalidades da aplicação encapsuladas pelo próprio modelo.

Um componente de visualização renderiza o conteúdo de uma parte particular do modelo e encaminha para o controlador as ações do usuário; acessa também os dados do modelo via controlador e define como esses dados devem ser apresentados.

Um controlador define o comportamento da aplicação, é ele que interpreta as ações do usuário e as mapeia para chamadas do modelo. Em um cliente de aplicações Web essas ações do usuário poderiam ser cliques de botões ou seleções de menus. As ações realizadas pelo modelo incluem ativar processos de negócio ou alterar o estado do modelo. Com base na ação do usuário e no resultado do processamento do modelo, o controlador seleciona uma visualização a ser exibida como parte da resposta a solicitação do usuário. Há normalmente um controlador para cada conjunto de funcionalidades relacionadas.

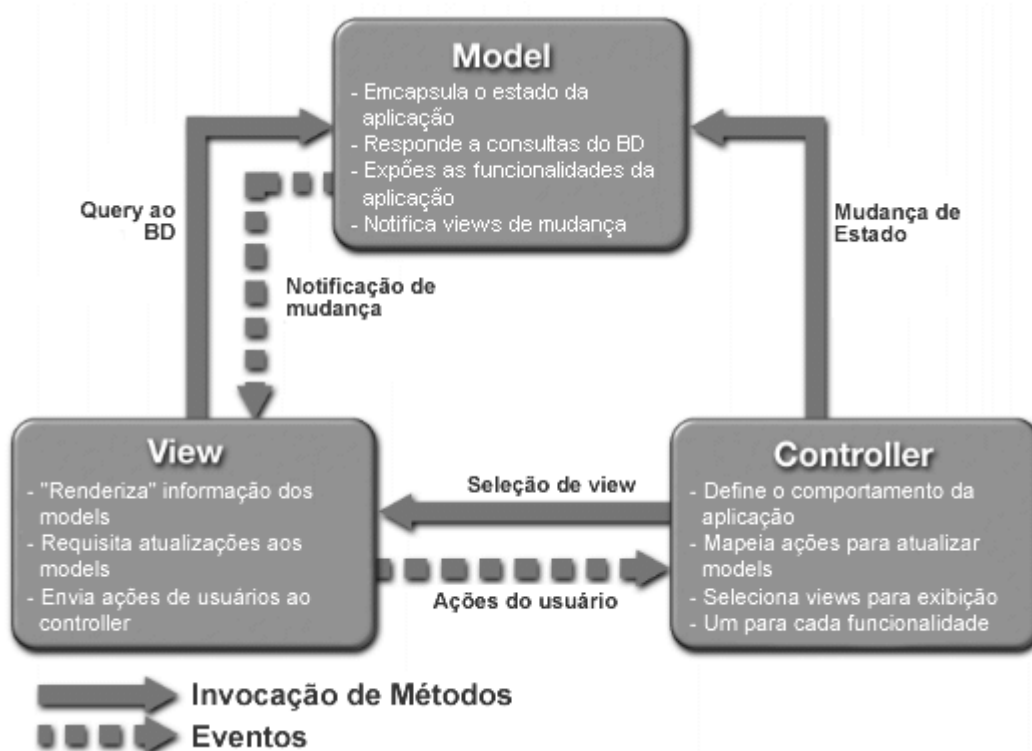


Figura 1 – Arquitetura do MVC
Fonte: Professional Asp.net MVC 1.0, 2009)

3.9 - Microsoft SQL Server

O MS SQL Server é um SGBD - sistema gerenciador de Banco de dados relacional criado pela Microsoft. É uma plataforma de dados confiável, produtiva e inteligente que permite executar aplicações de missão crítica mais exigentes, reduzir o tempo e o custo com o desenvolvimento. É considerado um Banco de dados robusto e permite a criptografia de um banco de dados inteiro, arquivos de dados ou arquivos de log. Os benefícios dessa criptografia incluem: consultas de dados criptografados usando consultas em série ou associadas, proteger os dados de consultas de usuários não autorizados e criptografia de dados sem requerer qualquer mudança nas aplicações existentes.

3.10 - Conclusões

Neste capítulo foram apresentadas as tecnologias utilizadas no desenvolvimento do projeto. Os motivos das escolhas de tais tecnologias foram a facilidade de uso e a produtividade que elas permitem ao desenvolvedor.

Em síntese, os recursos tecnológicos e de hardware necessários para o desenvolvimento do projeto, são:

- Visual Studio 2008: Esta será a IDE de desenvolvimento que proporciona melhor desempenho para trabalhar com C#;
- Sql Server 2008: Será o Banco de dados utilizado no projeto por trabalhar muito bem com o LINQ;
- Photoshop: Será utilizada para criar a identidade visual do Site;
- Framework Asp.net 3.5: Será o framework utilizado, já descrito na seção anterior.
- Linguagem C#: Será a linguagem de programação, também descrita na seção anterior;
- Asp.net MVC: Será a arquitetura de desenvolvimento deste projeto;
- Xhtml: Será utilizado na camada de apresentação (*Front-end*);
- Css: Será utilizado para formatar todos os elementos e conteúdos do site;
- Javascript: Será utilizado para validação de formulários;
- Framework JQuery 1.3.2: Será utilizado para ajudar na produtividade de funções que envolvam Ajax.

Recursos de hardware:

- Computador com processador Duo Core, mínimo de 4 Gb de memória Ram
- Servidor IIS
- Impressora.

O próximo capítulo fará um detalhamento de como estas tecnologias foram utilizadas no processo de desenvolvimento, demonstrando telas e documentação dos processos.

4. ORIENTAÇÃO A OBJETOS

4.1 Definições Básicas para Orientação a Objeto

As idéias fundamentais a tecnologia baseada em objetos incluem: Objetos, Classes, Abstração, Encapsulamento, Herança, Comunicação com Mensagens, Métodos de organização, Polimorfismo.

4.2 Objeto

Objeto é um substantivo que pode ser abstrato ou real, sobre o qual armazenamos dados e operações que manipulam os dados. Que se tem como exemplo uma pessoa, um avião, etc. Um objeto é a ocorrência específica (instância) de uma classe e é similar a uma entidade de uma tabela no modelo relacional somente até o ponto onde representa uma coleção de dados relacionados com um tema em comum.

Por exemplo, o nome ou CPF de um Empregado são dados que pertencem à entidade (Organização) ou ao objeto (Pessoas). Pessoas é uma mesma organização. Uma instância de Empregado é uma pessoa específica.

A diferença entre objeto e entidade é que entidade se preocupa meramente com os dados, só armazenamos um registro para cada entidade e o objeto se preocupa com os dados e os métodos através dos quais os dados estão manipulados e podem incluir tipos de registros múltiplos.

4.3 Classe

Classe é uma coleção de objetos que podem ser descritos com os mesmos atributos e as mesmas operações. Representa uma idéia ou um conceito simples e categoriza objetos que possuem propriedades similares, configurando-se em um módulo para a criação de novas instâncias.

4.4 Abstração

Quando usamos a abstração, admitimos que o que estamos considerando é complexo. Pois é uma das formas fundamentais de se lidar com complexidade. O resultado deste processo de abstração é conhecido como Modelo Conceitual. Dentro do paradigma de Orientação a Objeto, abstração denota características essenciais a um objeto que o distingue de outros tipos de objetos provendo fronteiras bem definidas entre diferentes visões. (YOU,1992)

Abstração de procedimento é o princípio de que qualquer operação com um efeito bem definido pode ser tratada por seus usuários como uma entidade única, mesmo que a operação seja realmente conseguida através de alguma seqüência de operações de nível mais baixo. (YOU,1992)

Por exemplo, para entendermos o funcionamento das plantas, dividimo-las em três partes básicas: raiz, caule e folhas. Estudamos cada uma de suas partes, abstraindo as demais. Para cada parte, identificamos funções diferentes e bem delimitadas.

A raiz é responsável pela absorção de água e sais minerais, o caule transporta os materiais coletados para as folhas e estas, através destes materiais fornecidos, produzem, através de fotossíntese, alimento para toda a planta. Da mesma forma

podemos analisar um mecanismo complexo como o computador ao dividi-lo em três partes: CPU, teclado e vídeo. Ao estudarmos cada uma destas partes isoladamente, diferenciando o que realmente importa para cada caso, podemos entender o micro como um todo a abstração de dados, é um outro mecanismo de abstração mais poderoso. Este princípio pode ser usado como base para a organização do pensamento e a especificação das responsabilidades de um sistema.

4.5 Encapsulamento

O empacotamento de dados e métodos juntas é chamada de encapsulamento. O objeto esconde seus dados de outros objetos e permite que os dados sejam acessados somente por meio dos próprios métodos de objetos.

O encapsulamento esconde, dos usuários de um objeto, os detalhes da implementação interna. Os usuários compreendem quais operações podem ser solicitadas de um objeto, mas não conhecem os detalhes de como a operação é executada.

5. DETALHAMENTO DO PROJETO

O principal objetivo deste Web site é criar uma ferramenta para gerenciar estoque de veículos, possibilitar também à revendedora um oportunidade de ter sua empresa na Web, ampliando sua rede de contatos e clientes, além de maior comodidade na hora de comprar um veículo.

Em termos de uso, o site será composto de duas áreas, uma logada e outra não logada. Somente o usuário Admin terá acesso a área logada, mediante um login e senha, este Admin terá acesso as todas as áreas do site. A área não logada é a parte visível a qualquer pessoa que venha a acessar o site, o visitante poderá navegar no site consultando o estoque de autos daquela revendedora, poderá enviar uma mensagem solicitando mais informações sobre um determinado veículo.

A figura abaixo mostra a Home do Site, nela está contido os principais links do site, bem como um Box para busca avançada.



Figura 5.0: Tela principal do Sistema

O site terá uma interface amigável de fácil navegação, as principais informações do site estarão presentes na Home, que também possibilitará pesquisas na base de dados. Os resultados serão listados, de acordo com a pesquisa feita, em uma outra página de resultados.

A fase de testes será realizada testando primeiro a interface do site nos browsers mais utilizados hoje em dia, verificando se não há quebras no layout e se o Javascript funciona em todos os browsers. Em seguida serão realizados testes relacionados a funcionalidade do sistema verificando a capacidade de inserir, deletar e atualizar registros no banco de dados.

Adicionar Veiculo

MENU

- > Marcas
- > Modelos
- > Veiculos

Nome do Veiculo

Modelo

Selecione um Modelo

Tipo

Usado

Ano Modelo

Ano de Fabricação

Preco

Combustivel

Fotos

<input type="text"/>	<input type="button" value="Procurar..."/>
<input type="text"/>	<input type="button" value="Procurar..."/>
<input type="text"/>	<input type="button" value="Procurar..."/>

Figura 5.1: Tela de cadastro de Novos veículos.

Adicionar Marca

MENU

- > Marcas
- > Modelos
- > Veiculos

Marca

Figura 5.2: Tela de cadastro de novas Marcas de Veículos

Adicionar Modelo

MENU

- > Marcas
- > Modelos
- > Veiculos

Nome

Marca

Selecione a marca

Salvar

Trabalho de Conclusão de Curso - 2010

Figura 5.3: Tela de cadastro de novos modelos de Veículos

Veiculos

MENU

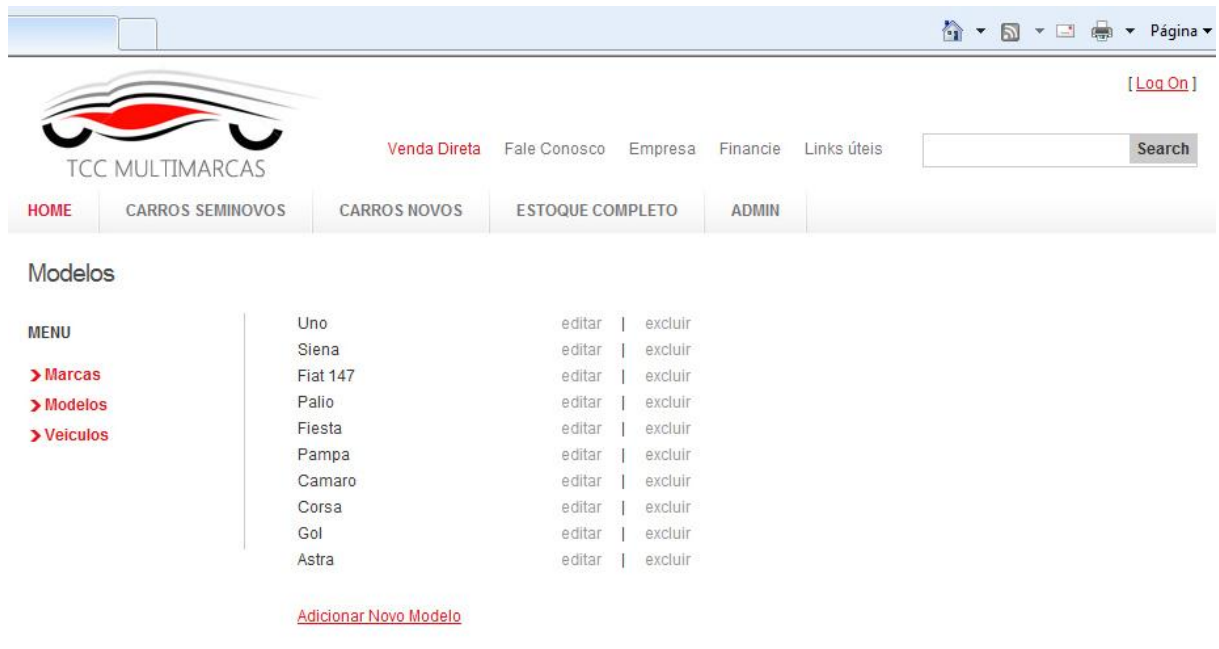
- > Marcas
- > Modelos
- > Veiculos

Veiculo	Ano Fabricacao	Ano Modelo	Tipo	Preco	Modelo	
Uno Mile	2009	2009	Novo	0,00	Uno	Editar Excluir
Uno Mile 16V	2007	2008	Novo	0,00	Uno	Editar Excluir
Uno	2004	2004	Usado	0,00	Uno	Editar Excluir
Uno CS 1.0	2005	2006	Usado	0,00	Uno	Editar Excluir
Uno	2008	2008	Usado	0,00	Uno	Editar Excluir
Siena 16V	2005	2005	Usado	0,00	Siena	Editar Excluir
Fiat 147 problemas	1981	1982	Usado	0,00	Fiat 147	Editar Excluir
Fiat 147 C	1999	1999	Usado	0,00	Fiat 147	Editar Excluir
Camaro SS	2000	2000	Usado	0,00	Camaro	Editar Excluir
Uno 8V SX	2007	2007	Usado	0,00	Camaro	Editar Excluir
Corsa	1998	1998	Usado	0,00	Corsa	Editar Excluir
Gol	2008	2009	Usado	0,00	Gol	Editar Excluir
GOL 1.0 I PLUS 8V	1996	1996	Usado	8500,00	Gol	Editar Excluir
Gol 2.0	1998	1998	Usado	10000,00	Gol	Editar Excluir

[Adicionar Veiculo](#)

Figura 5.4: Estoque de veículos

Esta tela exibe o estoque completo de veículo, através desta tela o Admin pode alterar, excluir os veículos cadastrado no Estoque.



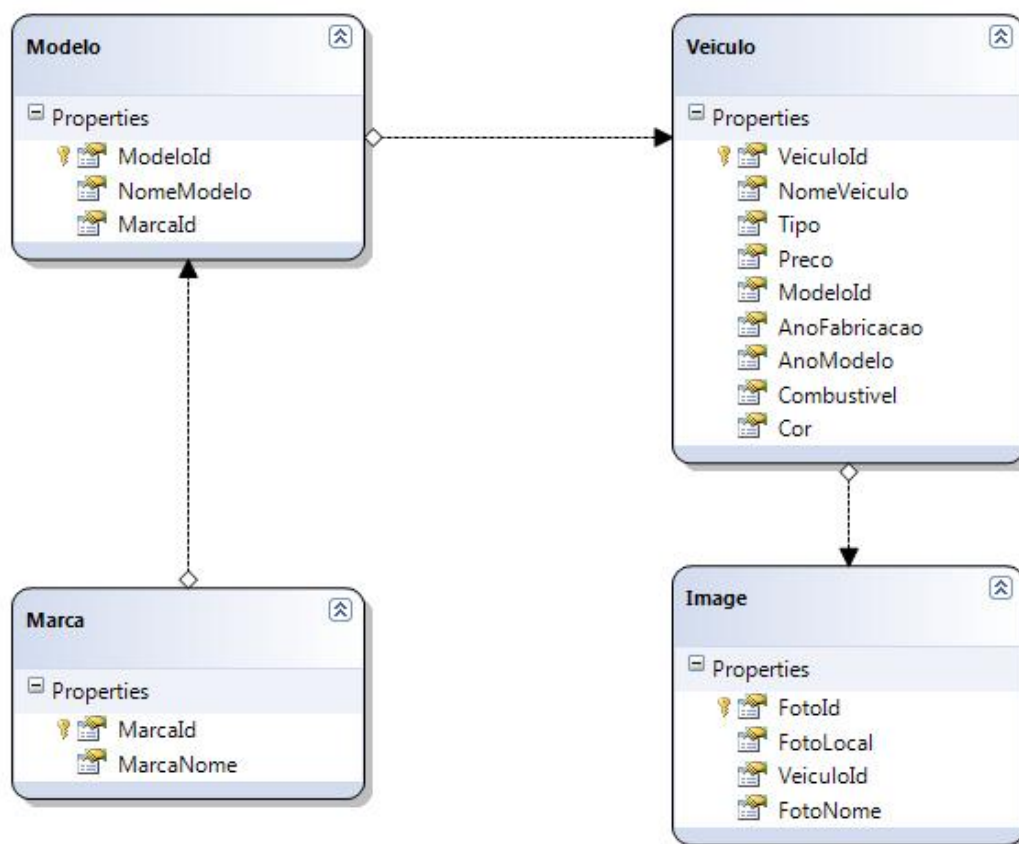
Trabalho de Conclusão de Curso - 2010

Figura 5.5: Tela que lista os modelos de veículos cadastrados.

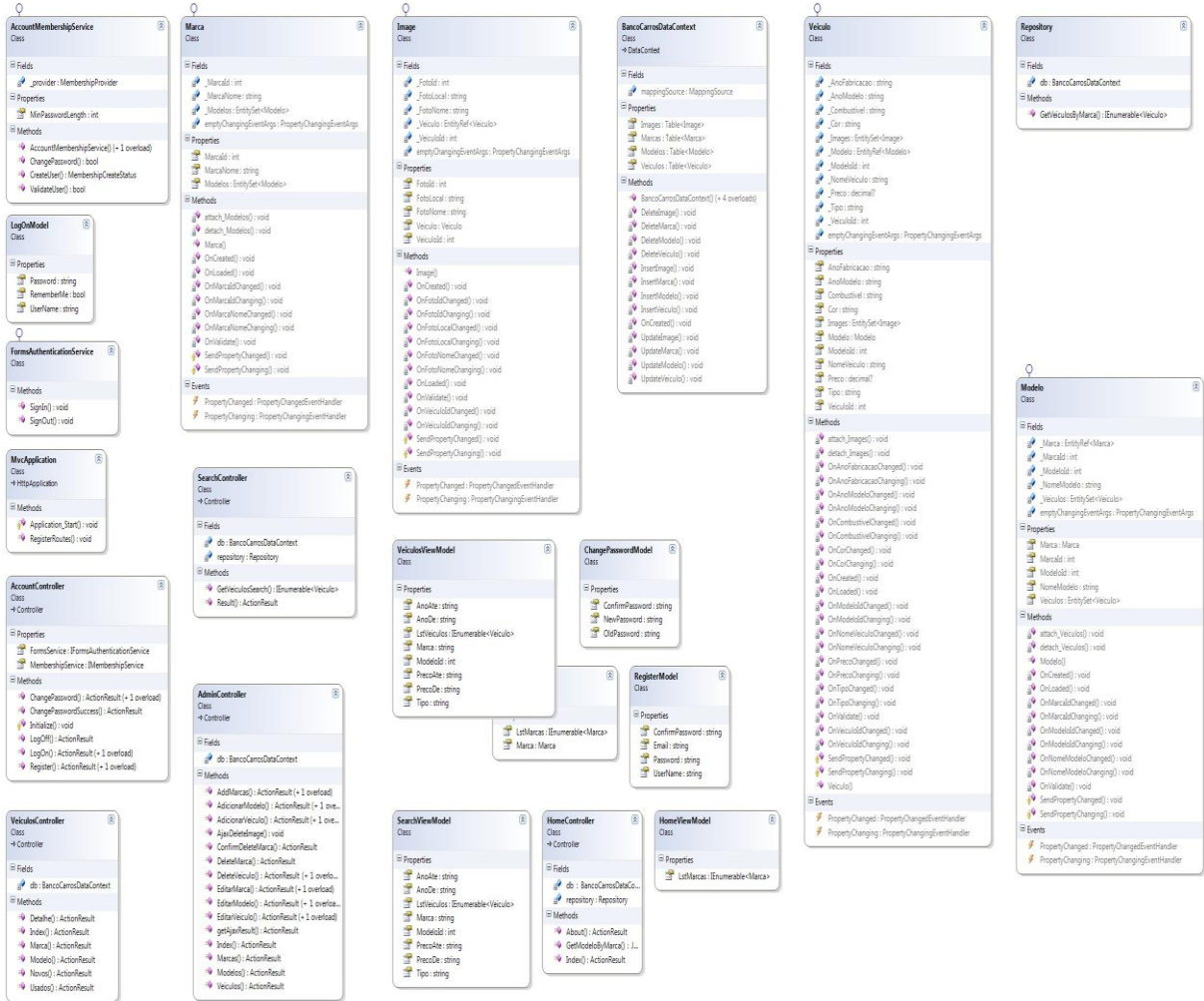


Figura 5.6: Tela que lista as Marcas de veículos cadastradas.

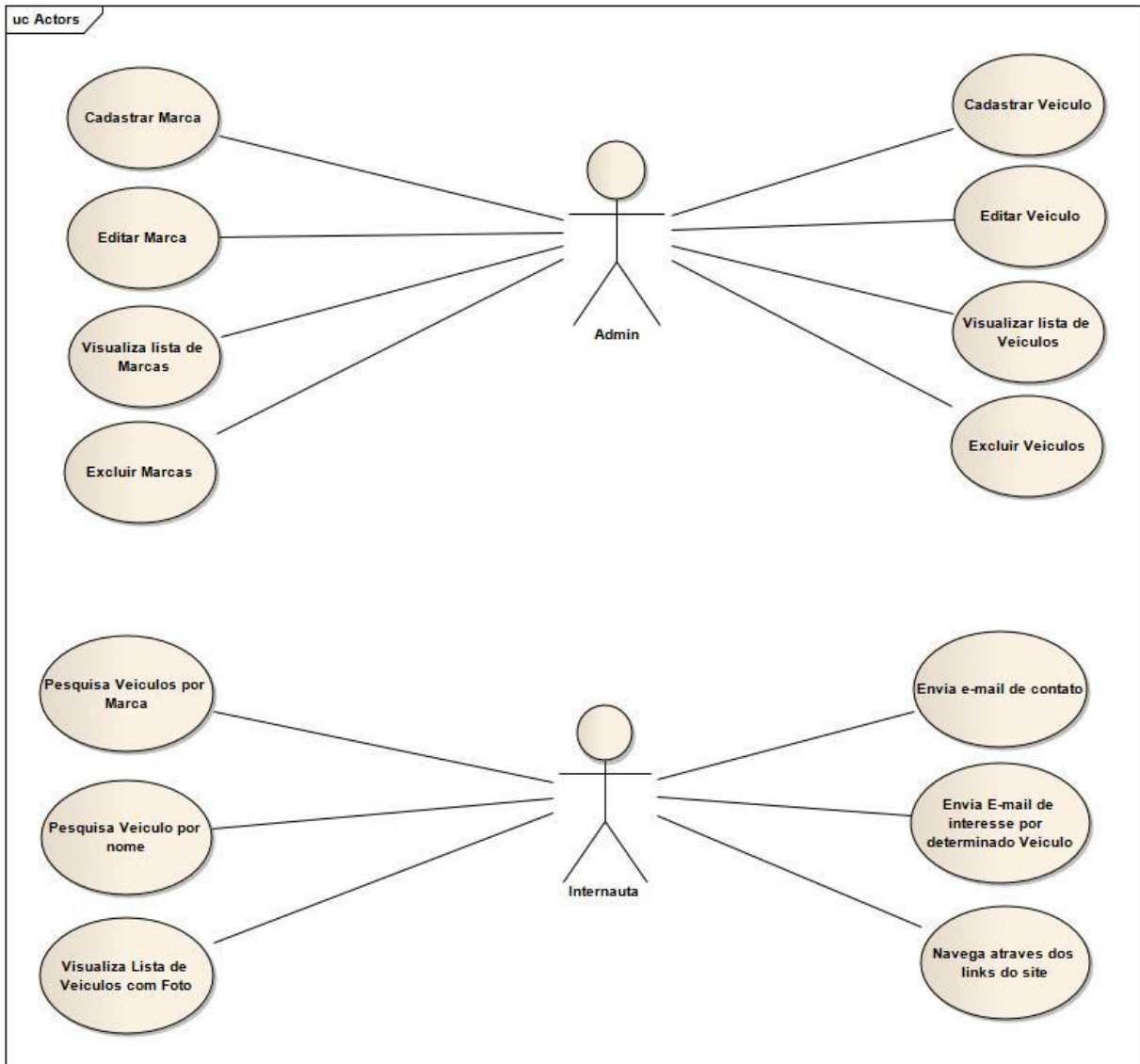
5.1 - Diagrama do Banco de Dados



5.2 - Diagrama de Classes



5.3 - Use Case



A implementação será realizada em duas etapas, primeiro será construída toda a parte estática do site, começando pela criação do Design, depois será feita a estrutura com CSS e Xhtml. Após a construção da parte estática onde já poderá ser visualizado como o site se comportará, será implementada a programação Server-side construindo o Banco de dados, criando as Classes e métodos necessárias para acesso aos dados.

6. CONCLUSÃO

No desenvolvimento deste projeto foram utilizadas diversas tecnologias para o desenvolvimento, como C#, Microsoft .NET Framework 3.5, Padrão de arquitetura MVC (Model-view-controller), o banco de dados Sql Server 2008 e LINQ (Language Integrated Query). Na camada de apresentação (Front-end) foi utilizado as tecnologias: Xhtml, Framework JQuery e CSS 2.0. O uso do Padrão MVC e a Tecnologia LINQ juntamente com Framework asp.net 3.5, trouxe maior facilidade em criar a aplicação Web com acesso a Banco de Dados, as tecnologias empregadas neste Sistema são muitos usados atualmente no desenvolvimento de aplicações para internet.

Neste projeto foi possível experimentar e aprimorar o seu uso. Em síntese, este projeto poderá ser utilizado para a criação de um novo negócio, além de disponibilizar para os estudantes, alguns conceitos úteis sobre as tecnologias aqui apresentadas, bem como no desenvolvimento desta aplicação.

7. REFERÊNCIAS BIBLIOGRÁFICAS

COAD, Peter, YOURDON, Edward. Análise baseada em Objetos. 2Ed. Rio de Janeiro: Campus, 1992. 215p.

Conery, Rob. Professional ASP.NET 3.5 In C# and VB Wiley Publishing, 2008

Conery, Rob. Asp.net MVC 1.0, p. 166, Wiley Publishing, 2009

FABRICE MARGUERIE - LINQ in Action Manning Publications Co., 2008

Nagel, Christian. Professional C# 2008 Wiley Publishing, 2008

Referência Eletrônica

Como as CSS favorecem a acessibilidade Disponível em
< <http://www.acessibilidadelegal.com/23-carac-css.php/>>. Acesso em maio 2010.

Como funcionam as redes sociais online em
< <http://informatica.hsw.uol.com.br/redes-sociais-online.htm>>

Henrique C. Pereira O que é XHTML Disponível em
< <http://revolucao.etc.br/archives/o-que-e-xhtml/>>. Acesso em maio 2010.

Javascript Disponível em
< <http://pt.wikipedia.org/wiki/JavaScript/>>. Acesso em maio 2010.

Jquery Apresentação Disponível em
< <http://desenvolvimentoparaweb.com/jquery/jquery-apresentacao-introducao/> >
Acesso em maio 2010.

Linq Disponível em
< <http://www.devmedia.com.br/post-14703-Artigo--net-Magazine-67-Introducao-a-LINQ.html/> > Acesso em Junho 2010

Maujor Blog – Porque CSS Disponível em < <http://maujor.com/>>. Acesso em maio 2010.

Microsoft SQL Server Disponível em < http://pt.wikipedia.org/wiki/Sql_server> Acesso em Junho 2010.

Padrão MVC - Disponível em <<http://pt.wikipedia.org/wiki/MVC>> Acesso em Junho 2010.

Pyramid Research. Desenvolvimento de Sistemas para Web. Acesso em 14/04/2010, disponível em (<http://www.pyr.com/>)

Vantagens do XHTML - Disponível em < <http://www.htmlstaff.org/ver.php?id=385> Acesso em Junho 2010.

weblogs: a history and perspective

< http://www.rebeccablood.net/essays/weblog_history.html>

Wikipédia - Cascading Style Sheets Disponível em

< http://pt.wikipedia.org/wiki/Cascading_Style_Sheets />. Acesso em maio 2010.