Licenciatura em Informática

Turma Manhã

Website de Stand de Automóveis

Trabalho Realizado Por:
Pedro Cerqueira Nº1900

Coordenador:
Prof. Pedro Brandão

Lisboa

Ano Letivo 2015/2016

Resumo

O Projeto Global consiste numa aplicação realizada na plataforma ASP.NET. A aplicação foi desenvolvida utilizando o programa *Visual Studio* da Microsoft e utilizou diversas linguagens de programação, nomeadamente C#, HTML, CSS e JavaScript. A aplicação utiliza uma base de dados em SQL para o armazenamento dos dados.

A aplicação consiste numa aplicação *web* que simula um *website* de stand de automóveis, permitindo aos utilizadores registaram-se e criarem uma conta de perfil e visualizarem e pesquisarem veículos na base de dados da aplicação.

A aplicação consiste principalmente em várias páginas ASP.NET ou páginas ASPX utilizadas para as diversas páginas da aplicação, como a *Home Page*, a página de perfil, a página de registo de utilizadores ou a página de pesquisa. A aplicação também contém outros objetos como um *data set* para fazer a ligação entre a aplicação e a base de dados e um *web handler* para a utilização de imagens na base de dados.

Abstract

The Global Project consists of an application developed in the ASP.NET platform. The application was developed using the *Visual Studio* program by Microsoft and used several programming languages, namely C#, HTML, CSS and JavaScript. The application uses an SQL database for data storage.

The application consists of a *web* application that simulates a car retail *website*, allowing users to register and create a profile account and view and search vehicles in the application's database.

The application consists mainly of various ASP.NET pages or ASPX pages used in several application pages, like the *Home Page*, the profile page, the user registration page and the search page. The application also contains other objects like a *data set* to connect the database with the application and a *web handler* for usage of the images in the database.

Palavras-Chave

Lista de Palavras-Chave utilizadas neste Projeto:

- *Framework* – Conceito abstrato em que *software* que fornece uma funcionalidade genérica pode ser alterado pelo utilizador para uma aplicação específica.
- *ASP.NET* – Plataforma da Microsoft orientada para aplicações *web*.
- *HTML* – Linguagem de Programação para desenvolver páginas *web* estáticas.
- *CSS* – Linguagem de Programação utilizado em conjunto com HTML para páginas *web*. Trata da formatação e aspeto gráfico da página.
- *Controlo* – Classe .NET executada no servidor que permite realizar uma função específica na aplicação.
- *ASPX* – Formato dos ficheiros das páginas das aplicações ASP.NET.
- *Master Page* – Ficheiro de modelo que permite definir partes consistentes nas páginas ASP.NET.
- *Layout* – Processo de cálculo da posição dos objetos num espaço.
- *Theme* – Um tema ou *template* que define o aspeto gráfico da página, desenvolvido em CSS.
- *Base de Dados* – Fonte de dados, composta por tabelas, que é associado a uma determinada aplicação.
- *SQL* – Linguagem de Programação orientada ao desenvolvimento de Bases de Dados.
- *Login* – Processo no qual o utilizador envia as suas credencias para a aplicação para identificação do mesmo.
- *Logout* – Processo no qual a aplicação termina a identificação do utilizador, o oposto do *Login*.
- *C#* - Linguagem de Programação da Microsoft orientada a objetos, derivada do C, que pode ser utilizada em aplicações ASP.NET.
- *Visual Studio* – Complexa ferramenta de programação da Microsoft, que permite criar diversas aplicações, incluindo aplicações *web* ASP.NET.
- *App_Data* – Pasta pré-definida do ASP.NET utilizada para guardar dados da aplicação, como base de dados ou ficheiros XML.
- *Data Set* – Objeto que é uma coleção de dados que serve como ligação entre a aplicação e a base de dados.

- *Query* – Uma função SQL que retorna um conjunto de dados da base de dados segundo parâmetros específicos.
- *App_Code* – Pasta pré-definida do ASP.NET utilizada para guardar ficheiros de código da aplicação, como *data sets*.
- *Web Handler* – Objeto do ASP.NET utilizado para criar código C# ou VB sem requerer uma página ASPX, podendo ser utilizado por várias páginas ASPX.
- *Cabeçalho* – Parte superior do *website*, onde geralmente é mostrado o título ou o slogan do *site*.
- *Rodapé* – Parte inferior do *website*, onde geralmente é mostrado os direitos reservados e outras informações.
- *Browser* – Aplicação cuja função é navegar nas páginas *web* (ex. Internet Explorer, Google Chrome, etc.)
- *Home Page* – A página inicial de um *website* ou aplicação *web*.
- *Hiperligação* – Uma referência em hipertexto dentro de um documento HTML que liga a outro documento ou outra secção do mesmo documento.
- *Tag* – Estrutura HTML que contém instruções. Tem geralmente uma marca de início e outra de fim e permite ao *browser* compilar uma página HTML.
- *Password* – Palavra-passe ou senha utilizada durante o processo de login para a confirmação da identificação do utilizador.
- *Filtro* – Uma condição utilizada durante a pesquisa para esta se tornar mais específica. Uma pesquisa pode conter um ou vários filtros.
- *Campo* – Uma informação particular de um registo na base de dados (ex. Nome, Id, etc.) sendo equivalente a uma coluna de uma tabela.

Abreviaturas

Lista de Abreviaturas utilizadas neste Projeto.

- *Site* – Abreviatura de *Website*
- *Int* – Abreviatura de *Integer*

Índice de Imagens

Índice

Introdução

O objetivo deste Projeto Global é a realização de um *website* desenvolvido em ASP.NET simulando um *stand* de venda de automóveis. Os utilizadores deste *website* poderão visualizar os automóveis disponíveis para venda e criar um perfil, o que lhes permitirá adicionar novos veículos.

A investigação deste Projeto visa também demonstrar os conhecimentos do aluno sobre as ferramentas utilizadas durante a realização do projeto como a utilização da plataforma ASP.NET assim como a utilização de bases de dados.

O desenvolvimento do Projeto Global está dividido em 11 capítulos:

- O 1º Capítulo trata do código CSS da aplicação.
- O 2º Capítulo trata da base de dados da aplicação.
- O 3º Capítulo trata do *data set* da aplicação.
- O 4º Capítulo trata da utilização de um *Web Handler* para as fotografias na base de dados da aplicação.
- O 5º Capítulo trata da *Master Page* da aplicação.
- O 6º Capítulo trata da *Home Page* da aplicação.
- O 7º Capítulo trata do Registo de Utilizadores na base de dados da aplicação.
- O 8º Capítulo trata do processo de *Login* na aplicação.
- O 9º Capítulo trata do Perfil do utilizador na aplicação.
- O 10º Capítulo trata da listagem de veículos na base de dados da aplicação.
- O 11º Capítulo trada da pesquisa de veículos na base de dados da aplicação.

Estado da Arte

Introdução

O desenvolvimento *web* tem vindo a evoluir rapidamente ao longo do tempo. As aplicações web requerem mais complexidade e facilidade para satisfazer os desejos dos consumidores. Isto também leva ao crescimento das ferramentas para o desenvolvimento das aplicações, tornando-se mais complexas e com capacidade para desenvolver melhores aplicações. (Walther, Hoffman, & Dudek, 2011, p. 1)

ASP.NET e .NET Framework

O ASP.NET é uma plataforma que faz parte da *Framework .NET* da Microsoft e é orientada para o desenvolvimento de aplicações *web*.

O .NET Framework é uma estrutura de *software* que é executado no Microsoft Windows. Inclui uma biblioteca de classes e suporta várias linguagens de programação que permitem interoperabilidade de linguagem (cada linguagem pode utilizar código escrito em outras linguagens). A biblioteca do .NET está disponível para todas as linguagens de programação que suportam a plataforma. Programas escritos para o .NET Framework são executados num ambiente conhecido como o *Common Language Runtime* (CLR), uma máquina virtual de aplicações que fornece serviços importantes, como segurança, gestão de memória e manipulação de erros. A biblioteca de classes e o CLR, juntos, constituem o .NET Framework. (Halvorsen, 2014, p. 5)

Linguagem HTML

O HTML é uma das várias linguagens de programação utilizadas pelo ASP.NET durante o processamento da página *web*, sendo responsável por grande parte do texto estático na página.

O HTML é a linguagem utilizada para a criação de páginas *web* e é compreendida por todos os navegadores *web* que existem atualmente. Documentos HTML são ficheiros de texto simples que contêm marcação, texto e dados adicionais que influenciam esse texto e como ele aparece na página *web*. (Spaanjaars, 2010, p. 10)

O HTML é escrito na forma de elementos que consistem em *tags*, como por exemplo <html>, dentro do conteúdo da página web. As *Tags* HTML normalmente vêm em pares como <h1> e </ h1>. A primeira *tag* num par é a *tag* inicial, a segunda *tag* é a *tag* final. Entre estas *tags*, os programadores *web* podem adicionar texto, tabelas, imagens, etc. (Halvorsen, 2014, p. 14)

Linguagem CSS

O CSS é outra linguagem utilizada pelo ASP.NET durante o processamento da página *web*, sendo responsável pela formatação e aspeto gráfico da página.

O CSS é utilizado para formatar as páginas *web* em várias formas possíveis. O CSS oferece um conjunto de opções para mudar o aspeto das páginas, incluindo fontes (tamanho, cor, família, etc.), cores, tamanhos, cores de fundo, bordas em torno de elementos HTML, o posicionamento dos elementos na página, e muito mais. O CSS é compreendido por todos os principais navegadores *web* de hoje e é a linguagem principal para a apresentação visual das páginas *web* e muito popular entre os desenvolvedores *web*. (Spaanjaars, 2010, p. 67)

Controlos ASP.NET

As formas *web* são páginas ASP.NET que utilizam controlos de servidor ASP.NET. Os controlos do servidor são um conjunto de controlos internos que são semelhantes aos controlos Microsoft de linguagens como Visual Basic ou C# sendo essencialmente partes do código já predefinidas que podem ser utilizadas nas aplicações ASP.NET. Os controlos ASP.NET são baseados em eventos; a tarefa do programador é escrever o código que trata os eventos disparados pelos controles. Este código aparece no ficheiro *code-behind* do ficheiro ASP.NET (.aspx) correspondente. (Microsoft Corporation, 2003, p. 9)

O ASP.NET inclui uma biblioteca rica em controlos de servidor. Esses controlos atendem as necessidades das tarefas comuns de *web design*. No entanto ainda existem casos em que um programador *web* é incapaz de encontrar um controlo que se adapte precisamente aos seus propósitos. É desejável, portanto, haver uma maneira de criar controlos personalizados e reutilizáveis que os programadores possam utilizar nas formas *web*. (Microsoft Corporation, 2003, p. 14)

Páginas ASP.NET

Uma página ASP.NET é um *framework* utilizado para criar páginas *web* dinâmicas. Uma página web HTML simples é estática; o seu conteúdo é determinado pelo código HTML fixo que está na página. As páginas dinâmicas como as páginas ASP.NET permitem que crie o conteúdo da página em tempo real. (Pope, 2012, p. 5)

Quando se cria uma página ASP.NET ou página .ASPX, também se cria o código-fonte de uma classe .NET. Uma nova instância da classe *System.Web.UI.Page* é criada. Todo o conteúdo de uma página ASP.NET, incluindo todos os scripts e conteúdo HTML, são compilados numa classe do .NET. (Walther, Hoffman, & Dudek, 2011, p. 34)

Quando uma página ASP.NET é solicitada, o *Framework* verifica a existência de uma classe .NET que corresponde à página. Se uma classe correspondente não existe, o *Framework* compila automaticamente a página numa nova classe e armazena a classe compilada na pasta de ficheiros temporários do ASP.NET. Na próxima vez que alguém solicitar a mesma página no futuro, a página não será compilada novamente. A classe previamente compilada é executada, e os resultados são retornados para o navegador, exceto se o código-fonte da página for modificado. (Walther, Hoffman, & Dudek, 2011, p. 34)

Quando a classe é adicionada à pasta de ficheiros temporários do ASP.NET, uma dependência de ficheiro é criado entre a classe e a página ASP.NET inicial. Se a página ASP.NET é modificada de qualquer forma, a classe .NET correspondente é automaticamente eliminada. Na próxima vez que alguém solicitar a página, o *Framework* compila automaticamente o código-fonte da página modificada para uma nova classe .NET. (Walther, Hoffman, & Dudek, 2011, p. 34)

Este processo é chamado de compilação dinâmica, o que permite às aplicações ASP.NET suportar milhares de utilizadores em simultâneo. Uma página ASP.NET não necessita de ser analisada e compilada de cada vez que for solicitada. Uma página ASP.NET é compilada apenas quando uma aplicação é modificada. (Walther, Hoffman, & Dudek, 2011, p. 34)

Master Page

Na maioria dos *websites*, apenas uma parte da página é alterada quando o utilizador navega de uma página para outra. As partes que não se alteram geralmente incluem regiões comuns, como o cabeçalho, um menu e o rodapé. Para criar páginas *web* com um *layout* consistente é necessário definir estas regiões relativamente estáticas num único modelo, a *Master Page*, introduzida na versão 2.0 do ASP.NET. O maior benefício das *Master Pages* é que elas permitem ao utilizador definir a aparência de todas as páginas do seu *website* num único local. Isso significa que se o utilizador quiser alterar o *layout* do seu *website* - por exemplo, mover o menu da esquerda para a direita - só é necessário modificar a *Master Page* e as páginas associadas a essa *Master Page* serão alteradas automaticamente. (Spaanjaars, 2010, p. 198)

A *Master Page* é uma página que fornece um *framework* no qual o conteúdo de outras páginas podem ser exibidas. As *Master Pages* permitem incluir cabeçalhos, menus de navegação e outros elementos em todas as páginas muito mais facilmente. (Lowe & Murach, 2005, p. 78)

Ao aproveitar a utilização de *Master Pages*, o *website* é mais fácil de manter, ampliar e modificar. Se é preciso adicionar uma nova página para o *website* que se assemelha às outras páginas no mesmo *website*, só é necessário aplicar a mesma *Master Page* para a nova página criada. Se o *layout* do *website* for totalmente alterado, não é necessário modificar cada página de conteúdo, apenas a *Master Page* para mudar drasticamente a aparência de todas as páginas no *website*. (Walther, Hoffman, & Dudek, 2011, p. 237)

Uma *Master Page* é semelhante a uma página ASP.NET, contendo HTML e outros controlos HTML e ASP.NET. Dentro da *Master Page*, é configurada o conteúdo para repetir

em todas as páginas, como a estrutura geral da página e o menu. No entanto, uma *Master Page* não é uma página ASP.NET e não pode ser solicitada diretamente no navegador, servindo apenas como o modelo em que as páginas ASP.NET se baseiam. (Spaanjaars, 2010, p. 198)

Temas

Um tema (*theme*) do ASP.NET é uma coleção de ficheiros que definem a aparência de uma página *web*. Um tema geralmente inclui ficheiros de *skin*, ficheiros CSS e imagens. Os temas são definidos na pasta *App_Themes* na raiz do *website*. Dentro desta pasta são criadas uma ou mais subpastas que definem os atuais temas. Dentro de cada subpasta estão os ficheiros que compõem o tema. (Spaanjaars, 2010, p. 218)

Um tema permite ao utilizador aplicar um estilo consistente para as páginas do *website*. Pode-se utilizar um tema para controlar a aparência de ambos os elementos HTML e os controlos ASP.NET que aparecem numa página. Os temas são diferentes das *Master Pages*. Uma *Master Page* permite ao utilizador partilhar conteúdo entre várias páginas do *website*. Um tema, por outro lado, permite controlar a aparência do conteúdo. (Walther, Hoffman, & Dudek, 2011, p. 269)

Uma hiperligação para cada ficheiro CSS na pasta do tema é automaticamente adicionada ao cabeçalho HTML das páginas sempre que o tema está ativo. As imagens na pasta do tema são referenciadas a partir dos ficheiros CSS, onde podem ser utilizadas para alterar os elementos comuns do *website*, como imagens de fundo, ou as imagens utilizadas em listas de itens ou listas de navegação. (Spaanjaars, 2010, p. 218)

Bases de Dados

Qualquer aplicação web deve envolver o acesso a dados. (Walther, Hoffman, & Dudek, 2011, p. 337) Ser capaz de utilizar uma base de dados em *websites* ASP.NET é tão importante como a compreensão de HTML e CSS: é quase impossível construir um *website* moderno, cheio de recursos sem a utilização de uma. As bases de dados são úteis porque permitem armazenar e recuperar dados de forma estruturada. A maior vantagem das bases de dados é que podem ser solicitadas durante o tempo de execução, o que significa que não se está limitado

aos ficheiros relativamente estáticos criados no *Visual Studio*. Pode-se utilizar uma base de dados para armazenar revisões, imagens, informações sobre os utilizadores (nomes de utilizador, endereços de e-mail, senhas, etc.), informações sobre quem lê comentários, artigos de notícias, e muito mais, e em seguida aceder aos dados através das páginas ASP.NET. (Spaanjaars, 2010, p. 403)

As bases de dados fornecem uma grande flexibilidade nos dados que apresentam, permitindo criar *websites* altamente dinâmicos que podem-se adaptar às preferências dos seus visitantes, ao conteúdo que o *website* tem para oferecer, ou até mesmo para os direitos de acesso que os utilizadores têm. Para trabalhar com uma base de dados numa página ASP.NET, é necessário utilizar a linguagem SQL (*Structured Query Language*). Esta linguagem permite recuperar e manipular dados armazenados numa base de dados. O ASP.NET também oferece ferramentas e controlos para criar tabelas e consultas utilizando esta linguagem. (Spaanjaars, 2010, p. 403)

Segurança e Autenticação

Apenas com o conteúdo abordado anteriormente, as páginas do *website* são acessíveis a todos os visitantes. Ainda não há nenhuma maneira de bloquear certos recursos, como ficheiros .ASPX ou até mesmo pastas inteiras para utilizadores específicos. Isso significa, por exemplo, que, atualmente, qualquer utilizador pode aceder à pasta de Gestão do *website* e modificar o seu conteúdo ou até remover ficheiros. O ASP.NET oferece uma variedade de ferramentas para criar um mecanismo de segurança sólido e seguro. (Spaanjaars, 2010, p. 579)

Ferramentas de segurança incluem os controlos *login* do ASP.NET, que servem para construir um sistema de registo e autenticação de utilizadores para o *website*. Os controlos *login* servem para exibir formulários de registo dos utilizadores, autenticação, mudança de senhas e lembretes. Por defeito, os controlos *login* utilizam o *ASP.NET Membership* para autenticar os utilizadores, criar novos utilizadores, e alterar as propriedades dos mesmos, mas os programadores podem criar um sistema inteiro utilizando uma base de dados para o registo dos utilizadores. Quando se utiliza os controlos *login*, não é necessário escrever qualquer código durante a execução dessas tarefas. (Walther, Hoffman, & Dudek, 2011, p. 1147)

Existem várias ferramentas importantes disponíveis para configuração para aplicações ASP.NET. Permissões para aplicações *web* são regulados pelo .NET Framework. Quando se construem as aplicações *web*, as opções de configuração podem fornecer segurança adicional para além das configurações padrão. Essas opções proporcionam mecanismos de autenticação e controle de autorização para permitir o suporte granular sobre quem pode utilizar essas aplicações e quais os recursos da aplicação que estão disponíveis para o utilizador. Políticas de todo o sistema também pode ser colocadas em prática para impor regras que serão aplicadas a todas as aplicações *web* instaladas. Essas opções são configuradas, alterando os parâmetros de configuração no ficheiro de configuração das aplicações *web*, que está localizado no diretório onde o aplicativo está instalado. Este arquivo, escrito em XML e nomeado "web.config", define um conjunto de políticas que serão aplicadas para a aplicação. (Manhoc Technologies, 2010, p. 4)

Implementação e Publicação

Quando o ASP.NET foi lançado pela primeira vez, uma das suas características mais elogiadas foi a implementação *Xcopy*. Isto refere-se à capacidade de um programador copiar ficheiros de um local para o outro para o funcionamento da aplicação. A implementação *Xcopy* foi uma resposta ao processo complicado de implantação *web* de aplicações ASP. (Walther, Hoffman, & Dudek, 2011, p. 1565)

Copiar os ficheiros para implementar uma aplicação *web* ainda é uma vantagem, mas as aplicações *web* de hoje tornaram-se muito mais do que apenas uma coleção de ficheiros ASP.NET e ficheiros DLL compilados do *website*. Aplicações web modernas carregam uma quantidade enorme de "bagagem" e os programadores modernos precisam de encontrar uma maneira de agrupar toda a bagagem com uma aplicação para uma implementação rápida e sem problemas. O *Visual Studio* oferece aos programadores a capacidade de criar pacotes de ficheiros através da sua interface gráfica ou da consola de comandos para a implementação das aplicações *web*. (Walther, Hoffman, & Dudek, 2011, p. 1565)

Conclusão

O ASP.NET oferece inúmeras vantagens no desenvolvimento de aplicações *web*, permitindo aos programadores desenvolverem *websites* muito mais rapidamente e com muito menos trabalho e oferecendo uma variedade de ferramentas para ajudar ao desenvolvimento das aplicações. O ASP.NET também incorpora várias linguagens como o HTML e o CSS para o aspeto estático e gráfico da aplicação, o C# ou VB para o código e a componente lógica e o SQL para a criação e utilização das bases de dados utilizadas pelas aplicações.

No entanto, o ASP.NET também apresenta algumas desvantagens como o facto de apenas utilizar programas Microsoft e serem orientadas apenas a servidores que utilizam o Windows. O alojamento em ASP.NET tem tendência a ser mais caro devido ao uso de licenças Microsoft ao contrário de outros tipos de alojamentos *web* como por exemplo o alojamento em PHP.

O projeto irá ser realizado apenas com ferramentas Microsoft, nomeadamente o *Visual Studio*, que já tem o ASP.NET incorporado, de maneira a evitar problemas durante o desenvolvimento e publicação da aplicação. O conteúdo da aplicação e da base de dados serão adaptadas ao tema do trabalho, utilizando as ferramentas e funções abordadas neste Estado da Arte.

Contextualização

O tema do ASP.NET é importante e significativo devido à plataforma ser cada vez mais utilizada atualmente. O ASP.NET também oferece diversas ferramentas que facilitam o desenvolvimento, a implementação e a publicação de aplicações *web*.

A investigação do projeto consiste no desenvolvimento de uma aplicação *web* que simula um *website* de venda de automóveis. A aplicação utiliza diversas linguagens de programação incluindo C#, HTML, CSS, JavaScript e XAML para além de uma base de dados SQL.

Desenvolvimento

Capítulo 1 – Template CSS

O projeto inicia-se com a criação da aplicação no *Visual Studio*, que inclui especificar o nome da aplicação e o diretório base, onde os ficheiros irão ser criados e localizados. O nome da aplicação é *Projecto Global*. O projeto então inicia-se com a criação do ficheiro CSS, na qual se deu o nome de *StyleSheet.css*, para o tema da aplicação. O CSS inclui um *download* de um *template*, que inclui imagens, para estabelecer a base visual do tema. O *template* é então editado e adaptado á aplicação e as suas imagens são inseridas numa pasta criada no diretório base, chamada *Imagens*.

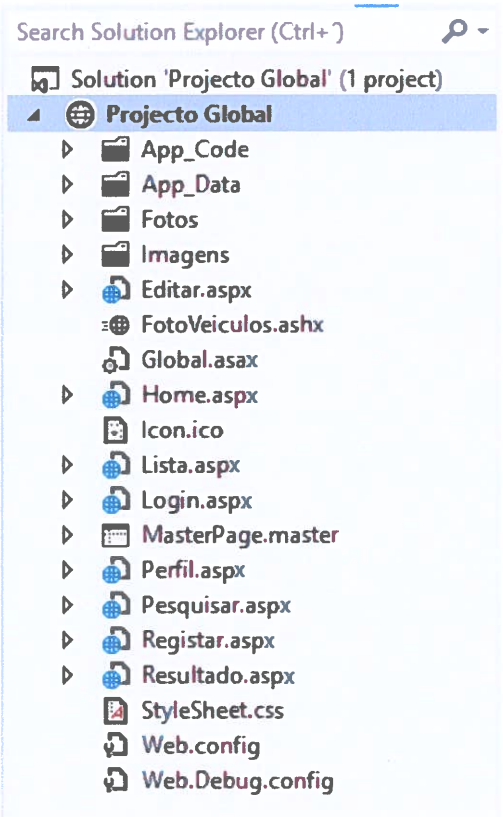

Fig. 1 - Os ficheiros e pastas no diretório da aplicação representados no Visual Studio.

Capítulo 2 – Base de Dados

O próximo passo no projeto, ainda antes do desenvolvimento das páginas ASP.NET, é a criação de uma base de dados SQL chamada *StandAutomoveis.mdf*, que fica localizada numa pasta pré-definida do ASP.NET chamada *App_Data*. Após a criação da base de dados, procede-se á criação das tabelas. A base de dados contém 4 tabelas: *Marcas*, *Modelos*, *Utilizadores* e *Veiculos*.

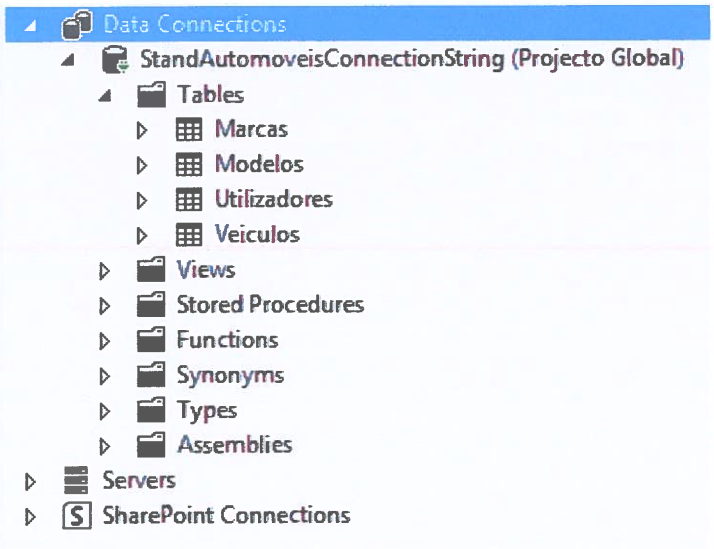

Fig. 2 - A informação da base de dados representada no Visual Studio.

As tabelas e as suas colunas na base de dados do projeto são as seguintes:

- Marcas – Lista de Marcas de Veículos
  - *Id* – Coluna do tipo *int* (número inteiro) e a chave primária da tabela. A coluna representa o número de identificação da marca na tabela.
  - *Nome* – Coluna do tipo *nvarchar* (conjunto de caracteres), representa o nome descritivo da marca.
- Modelos – Lista de Modelos de Marcas de Veículos
  - *Id* – Coluna do tipo *int* e a chave primária da tabela. A coluna representa o número de identificação do modelo na tabela.
  - *Marca* – Coluna do tipo *int* e uma chave estrangeira da tabela. A coluna representa o número de identificação da marca, estando associada á coluna *Id* da tabela Marcas.
  - *Nome* – Coluna do tipo *nvarchar*, representa o nome descritivo do modelo.

- Utilizadores – Lista de Utilizadores registados na aplicação
    - *Nome* – Coluna do tipo *nvarchar*, com 50 caracteres de tamanho, e a chave primária da tabela. A coluna representa o nome do utilizador, que é único.
    - *Password* – Coluna do tipo *nvarchar*, representa a *password* (senha) do utilizador para segurança.
- Veículos – Lista dos Veículos registados na base de dados
    - *Id* – Coluna do tipo *int* e a chave primária da tabela. A coluna representa o número de identificação do veículo na tabela.
    - *Marca* – Coluna do tipo *int* e uma chave estrangeira da tabela. A coluna representa o número de identificação da marca, estando associada á coluna *Id* da tabela Marcas.
    - *Modelo* – Coluna do tipo *int* e uma chave estrangeira da tabela. A coluna representa o número de identificação do modelo, estando associada á coluna *Id* da tabela Modelos.
    - *Utilizador* – Coluna do tipo *nvarchar*, com 50 caracteres de tamanho, e uma chave estrangeira da tabela. A coluna representa o nome do utilizador que adicionou o veículo, estando associada á coluna *Nome* da tabela Utilizadores.
    - *Data* – Coluna do tipo *date*, representa a data em que foi adicionado o veículo á base de dados.
    - *Preco* – Coluna do tipo *int*, representa o preço (em Euros) do veículo.
    - *Ano* – Coluna do tipo *date*, representa o ano de registo do veículo. Podem ser inseridos anos desde 1950 até 2015.
    - *Matricula* – Coluna do tipo *nvarchar*, com 8 caracteres de tamanho, representa a matrícula do veículo. Contém um valor único.
    - *Combustivel* – Coluna do tipo *nvarchar*, com 50 caracteres de tamanho, representa o tipo de combustível utilizado pelo veículo (ex. Gasolina, Gasóleo, etc.).
    - *Quilometros* – Coluna do tipo *int*, representa a quantidade de quilómetros percorridos pelo veículo.
    - *Cor* – Coluna do tipo *nvarchar*, com 50 caracteres de tamanho, representa a cor do veículo.
    - *Potencia* – Coluna do tipo *int*, representa a potência (em cavalos) do veículo.

- *Lotacao* – Coluna do tipo *nvarchar*, com 2 caracteres de tamanho, representa a quantidade de passageiros que o veículo pode transportar.
- *Portas* – Coluna do tipo *nvarchar*, com 2 caracteres de tamanho, representa a quantidade de portas presentes no veículo.
- *Foto* – Coluna do tipo *nvarchar*, representa o endereço do ficheiro de imagem que contém uma foto do veículo.
- *FotoTipo* – Coluna do tipo *nchar*, com 12 caracteres de tamanho, representa o tipo de ficheiro de imagem da coluna anterior. (ex. JPG, GIF, PNG, etc.)

```
CREATE TABLE [dbo].[Veiculos] (
    [Id]           INT             NOT NULL,
    [Marca]        INT             NOT NULL,
    [Modelo]       INT             NOT NULL,
    [Utilizador]   NVARCHAR (50)   NOT NULL,
    [Data]         DATE            NOT NULL,
    [Preco]        INT             NOT NULL,
    [Ano]          DATE            NOT NULL,
    [Matricula]    NVARCHAR (8)    NOT NULL,
    [Combustivel]  NVARCHAR (50)   NOT NULL,
    [Quilometros]  INT             NOT NULL,
    [Cor]          NVARCHAR (50)   NOT NULL,
    [Potencia]     INT             NOT NULL,
    [Lotacao]      NVARCHAR (2)    NOT NULL,
    [Portas]       NVARCHAR (2)    NOT NULL,
    [Foto]         NVARCHAR (MAX)  NULL,
    [FotoTipo]     NCHAR (12)      NULL,
    PRIMARY KEY CLUSTERED ([Id] ASC)
);
```

Fig. 3 - O código SQL para a criação da tabela Veiculos.

Capítulo 3 – Data Set

Após a criação da base de dados, prossegue-se para a criação do *data set*, um objeto pré-definido no ASP.NET que permite as ligações entre as tabelas e definir as *querys* de cada uma das tabelas. O *data set*, cujo ficheiro é *dsStandAutomoveis.xsd*, é criado na pasta *App_Code*, uma outra pasta já pré-definida no ASP.NET.

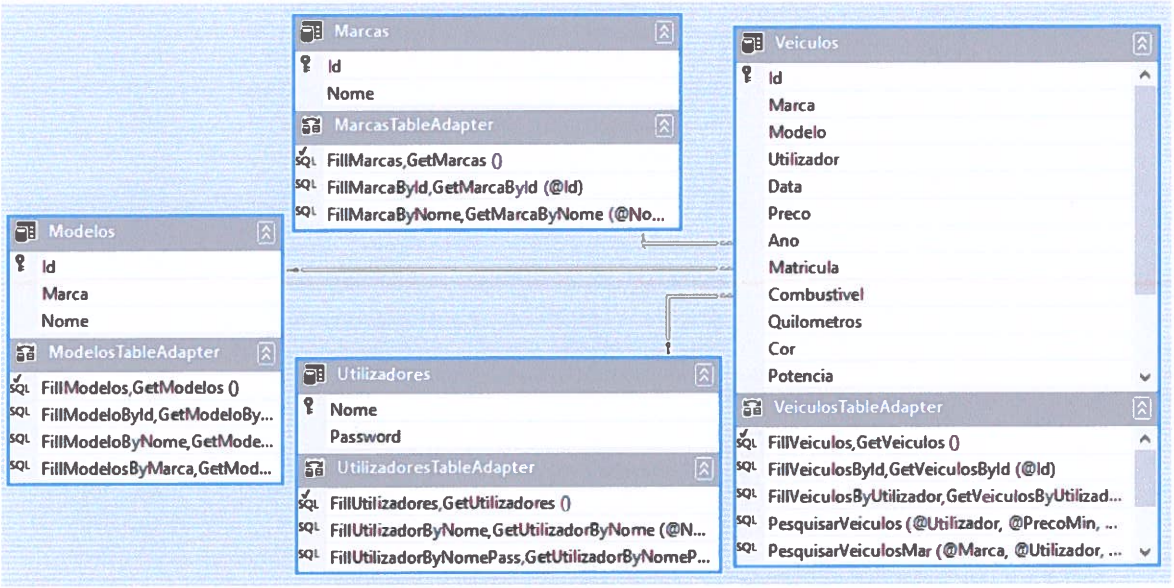

Fig. 4 - Formato gráfico das tabelas no Data Set.

```
SELECT Id, Marca, Modelo, Utilizador, Data, Preco, Ano, Matricula, Combustivel, Quilometros, Cor, Potencia,
Lotacao, Portas, Foto, FotoTipo
FROM dbo.Veiculos
WHERE Utilizador LIKE '%' + @Utilizador + '%'
AND Preco >= @PrecoMin AND Preco <= @PrecoMax
AND Ano >= @AnoMin AND Ano <= @AnoMax
AND Matricula LIKE '%' + @Matricula + '%'
AND Combustivel LIKE '%' + @Combustivel + '%'
AND Quilometros >= @QuilometrosMin AND Quilometros <= @QuilometrosMax
AND Cor LIKE '%' + @Cor + '%'
AND Potencia >= @PotenciaMin AND Potencia <= @PotenciaMax
AND Lotacao LIKE '%' + @Lotacao + '%'
AND Portas LIKE '%' + @Portas + '%'
```

Fig. 5 - A query SQL utilizada para a pesquisa de veículos na base de dados.

Capítulo 4 – Web Handler

Para a utilização de fotos na base de dados da aplicação, que requere um processo especial, foi criado um *Web Handler*, um objeto utilizado para escrever código C# para o servidor sem necessidade de utilizar um página ASPX. A vantagem dos *Web Handlers* é que estes podem ser utilizados em várias páginas ASPX. A aplicação utiliza um *Web Handler* para transferir as imagens dos veículos para uma pasta no diretório da aplicação, chamada *Fotos*, para a aplicação conseguir buscar as fotos facilmente para mostrar ao utilizador durante a listagem de veículos. O endereço da imagem do veículo será guardado na base de dados para depois ser utilizado nas funções do *Web Handler* para buscar a foto. No caso de um veículo ser adicionado á base de dados sem uma foto, foi criada uma imagem na pasta *Fotos* chamada *Sem Foto.jpg*, para ser utilizada como uma foto ausente. O *Web Handler*, denominado *FotoVeiculos.ashx*, é utilizado pelas páginas de Perfil, de Listagem e Pesquisa de Veículos da aplicação.

```
public void ProcessRequest (HttpContext context) {
    int id;
    if (context.Request.QueryString["Id"] != null)
    {
        if (int.TryParse(context.Request.QueryString["Id"].ToString(), out id))
        {
            adpt = new VeiculosTableAdapter();
            tab = adpt.GetVeiculosById(id);
            if (tab.Rows.Count > 0)
            {
                r = (DataSet.VeiculosRow)tab.Rows[0];
                string caminho;
                if (!r.IsFotoNull() && !r.IsFotoTipoNull())
                {
                    context.Response.ContentType = r.FotoTipo.ToString();
                    caminho = context.Server.MapPath("~/Fotos/" + r.Foto);
                    if (!File.Exists(caminho))
                    {
                        context.Response.ContentType = "Image/jpeg";
                        caminho = context.Server.MapPath("~/Fotos/" + "Sem Foto.jpg");
                    }
                }
                else
```

Fig. 6 - Excerto do código C# do Web Handler.

Capítulo 5 – Master Page

Após a criação do *data set*, procede-se á criação das páginas ASP.NET da aplicação. No entanto, antes da criação das páginas, deve-se criar a *Master Page*, que irá definir as partes consistentes do *website*. A *Master Page*, cujo ficheiro é *MasterPage.master*, é criada e código HTML é adicionado. As partes do *website* localizadas na Master Page neste Projeto Global são o cabeçalho, o menu de navegação e o rodapé.

```
<%@ Master Language="C#" AutoEventWireup="true" CodeFile="MasterPage.master.cs" Inherits="MasterPage" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Stand de Automóveis</title>
    <link rel="stylesheet" href="StyleSheet.css" />
    <link rel="shortcut icon" type="image/x-icon" href="Icon.ico" />
    <asp:ContentPlaceHolder ID="head" runat="server">
    </asp:ContentPlaceHolder>
</head>
<body>
    <div id="templatemo_container">
        <form id="form1" runat="server">
            <div id="templatemo_header">
                <div id="website_title">
                    <h1 style="color: white;">Stand de Automóveis</h1>

                    <div class="userinfo">
                        <asp:Label ID="lblUtilizador" runat="server"></asp:Label>
                         
                        <asp:Button ID="btnLogout" runat="server" Text="Logout" OnClick="btnLogout_Click" />
                    </div>
                </div>
```

Fig. 7 - Excerto do código HTML da Master Page.

A *Master Page*, como as páginas ASPX, contém um ficheiro de código associado, de formato VB ou C#, dependendo da linguagem utilizada. Como este Projeto utiliza C#, o ficheiro é *MasterPage.master.cs*, que contém código utilizado na Master Page.

O código C# acima é composto por uma classe da *Master Page* e, dentro desta, duas funções. A primeira função é a *Page_Load*, que é invocada quando a *Master Page* está a ser carregada para o *browser*. A segunda função é a *btnLogout_Click*, que é invocada quando o botão *btnLogout*, um botão localizado na Master Page cuja função é o utilizador fazer o *logout* na aplicação, é clicado.

Capítulo 6 – Home Page

Depois de se criar a *Master Page*, procede-se á criação da *Home Page*, que se denomina *Home.aspx*, e é a primeira página ASPX do projeto. Esta página meramente contém uma frase de boas-vindas e frases contendo hiperligações a outras páginas do aplicação, semelhante ao menu de navegação.

Fig. 8 - Aparência da aplicação na Home Page.

A *Home Page*, sendo uma página ASPX e como a *Master Page*, contém código HTML e CSS para o formato gráfico e JavaScript e C# para o código dinâmico, o último num ficheiro separado. Todas as páginas ASP.NET que têm uma ligação no menu de navegação, contém um código JavaScript no código HTML na *tag* <*head*> que modifica a aparência da hiperligação no menu de navegação (que está na *Master Page*) para refletir a página em que o utilizador está. Por exemplo, na figura acima, note-se que a hiperligação *Home* no menu de navegação contém texto de cor laranja ao contrário das outras hiperligações, em que o texto tem cor branca, essencialmente mostrando ao utilizador que está na *Home Page*. A *Home Page*, não contendo nada dinâmico, não tem necessidade de código C#.

Capítulo 7 – Registo de Utilizadores

A próxima página criada para a aplicação é a página onde os utilizadores se poderão registar e terem as suas credenciais adicionadas á base de dados. O ficheiro da página tem o nome de *Registar.aspx* e o seu ficheiro de código associado é *Registar.aspx.cs*.

Fig. 9 - Aparência da página de Registo de Utilizadores na aplicação.

A página contém um título e um subtítulo explicando ao utilizador a função da página, seguido de dois campos de texto (nome e password) e um botão para enviar os dados para a aplicação.

Capítulo 8 – Login

Depois do registo de utilizadores, procede-se á criação do processo de *login*, criando outra página ASPX, chamada *Login.aspx*.

Fig. 10 - Aparência da página Login na aplicação.

A página do *Login* é semelhante á página do registo de utilizadores, com a exceção do subtítulo, que não é necessário nesta página e é ausente. No *Login*, o utilizador insere o seu nome e *password* e clica no botão *Login* para enviar os seus dados para a aplicação o identificar. A aplicação então procura pelos dados na base de dados e se encontrar um registo em que os dados correspondem todos corretamente, a aplicação inicia uma sessão com esse utilizador. Se não encontrar registo com o nome e *password* especificados, aparecerá uma mensagem de erro a vermelho alertando o utilizador de que o nome ou a password estão incorretos. Clicar no botão sem especificar o nome ou a *password* resulta noutro erro alertando o utilizador sobre os dados em falta.

Capítulo 9 – Perfil do Utilizador

Após os utilizadores fazerem o *login* na aplicação, eles poderão aceder ao seu perfil, onde encontrarão informação sobre a sua conta e podem gerar os veículos que o utilizador tem registados na base de dados. Os utilizadores poderão adicionar novos veículos, editar e remover os veículos existentes. Se o utilizador tentar entrar no Perfil sem fazer *login*, a aplicação irá redirecionar o utilizador para a página de *Login*.

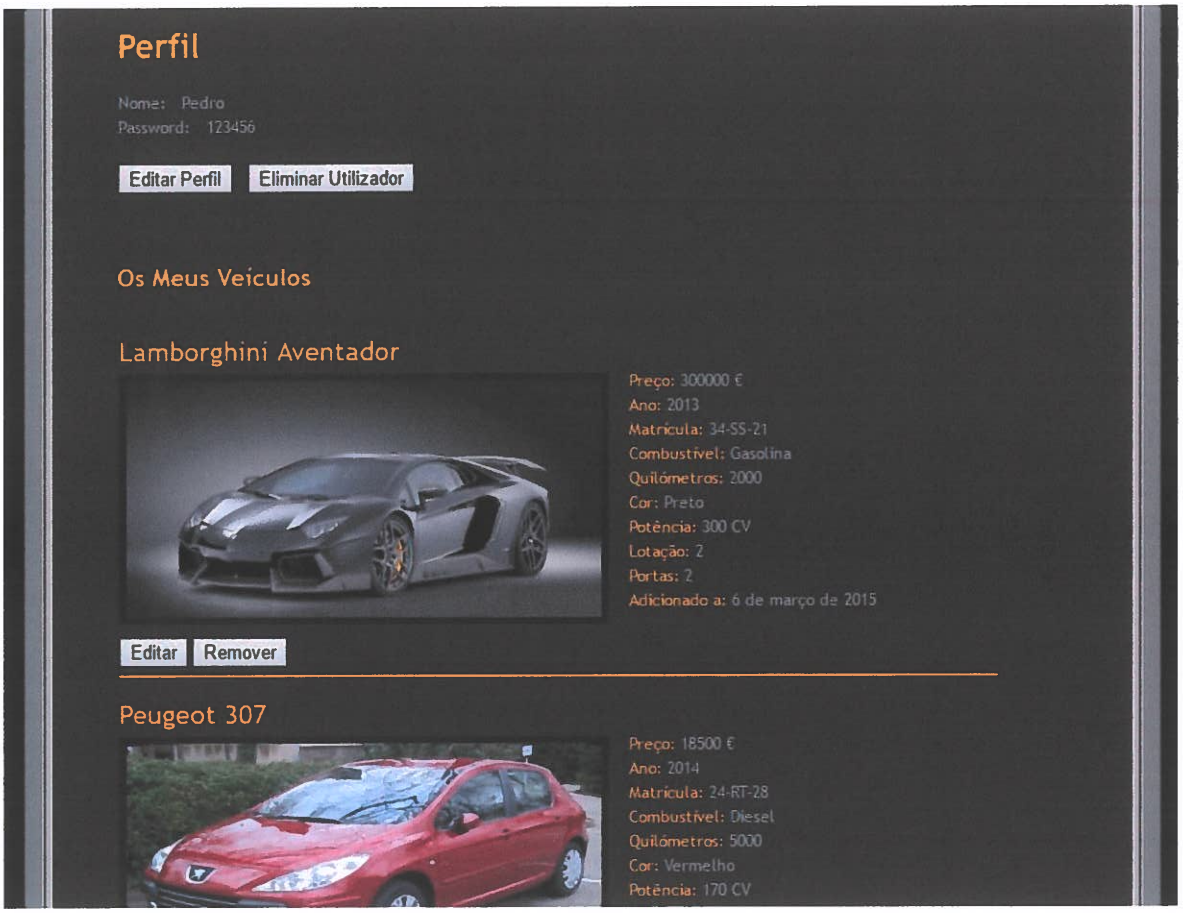

Fig. 11 - Aparência da página de Perfil na aplicação.

Para além das informações referidas acima, os utilizadores também podem alterar o seu perfil ou eliminar o registo da sua conta na aplicação.

Capítulo 9.1 – Editar Perfil

Para a edição do perfil do utilizador, foi criada outra página ASPX, denominada *Editar.aspx*. Nesta página o utilizador poderá alterar o seu nome e a sua *password*.

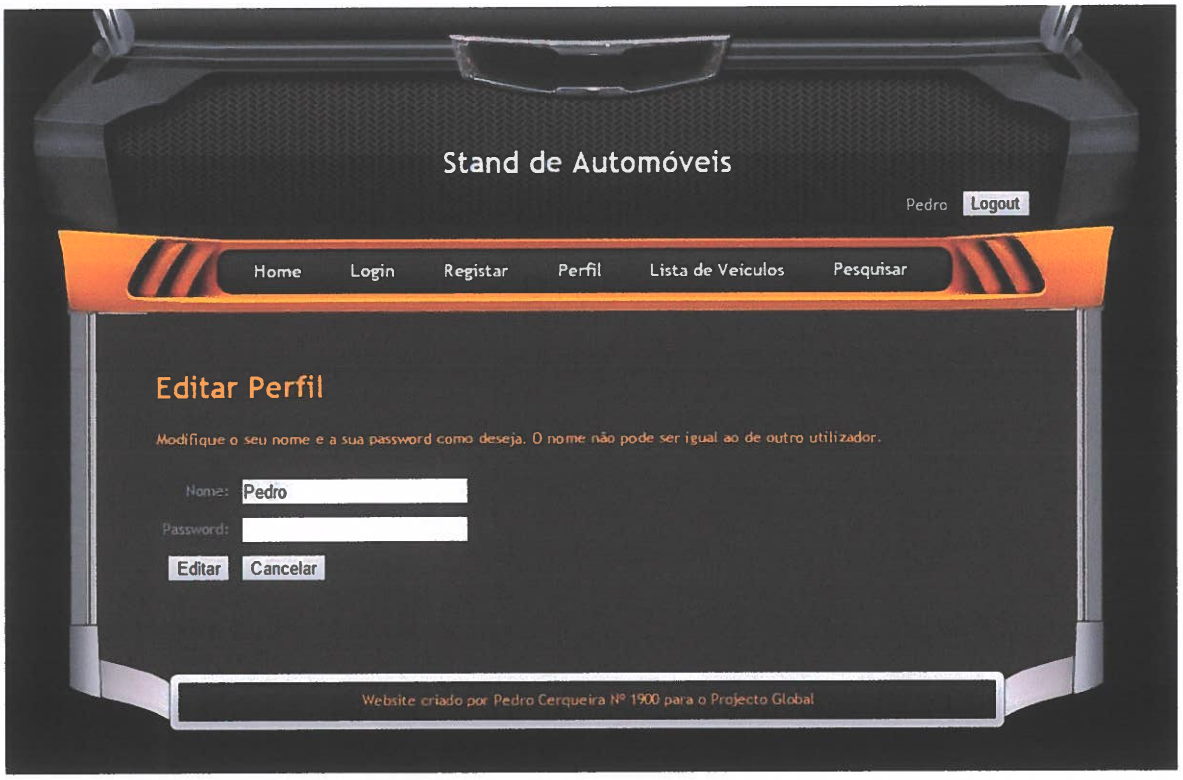

Fig. 12 - Aparência da página de edição de perfil na aplicação.

Durante a edição do perfil, o nome não pode ser igual ao de outro utilizador. Se isso acontecer, a página irá mostrar uma mensagem de erro avisando o utilizador do sucedido. O utilizador pode também cancelar a edição, o que o fará regressar á página do Perfil.

Capítulo 10 – Lista de Veículos

A partir do menu de navegação, o utilizador também poderá aceder á lista completa de veículos registados na base de dados da aplicação. Para isso, foi criada uma página ASPX somente para listar todos os veículos, sem filtros de pesquisa, chamada *Lista.aspx*.

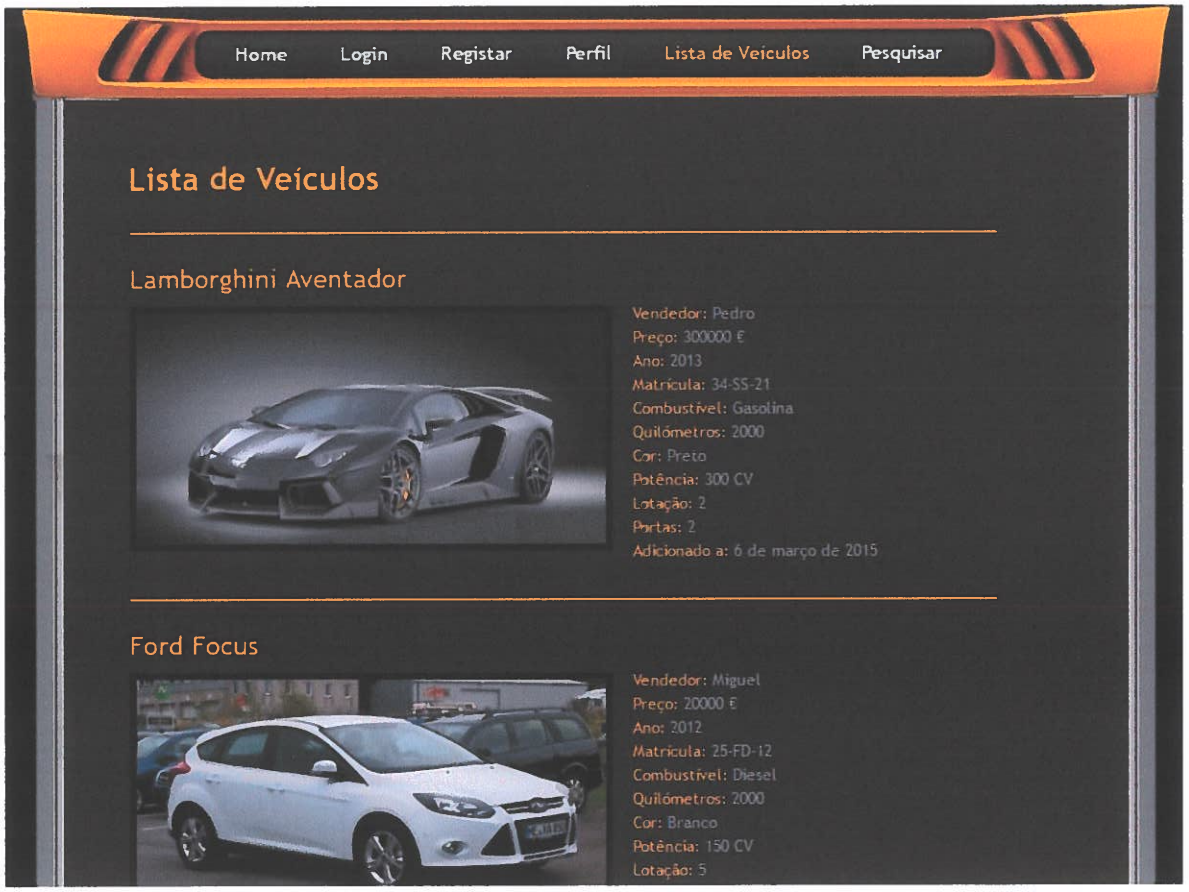

Fig. 13 - A lista de veículos na aplicação.

Capítulo 11 – Pesquisa de Veículos

A pesquisa de veículos é a parte mais importante e complexa do projeto. A pesquisa permitirá ao utilizador encontrar veículos muito mais especificamente. Pode-se pesquisar carros de um determinado ano, de uma determinada marca, modelo, etc. Existem vários campos de pesquisa e o utilizador pode preencher qualquer um deles. Todos são opcionais, por isso, se nenhum filtro for especificado, a pesquisa retorna todos os veículos na base de dados, como na listagem. Para a pesquisa foram criadas duas páginas ASPX, uma página para mostrar os campos de pesquisa, denominada *Pesquisar.aspx*, e outra página para mostrar os resultados da página anterior, denominada *Resultado.aspx*.

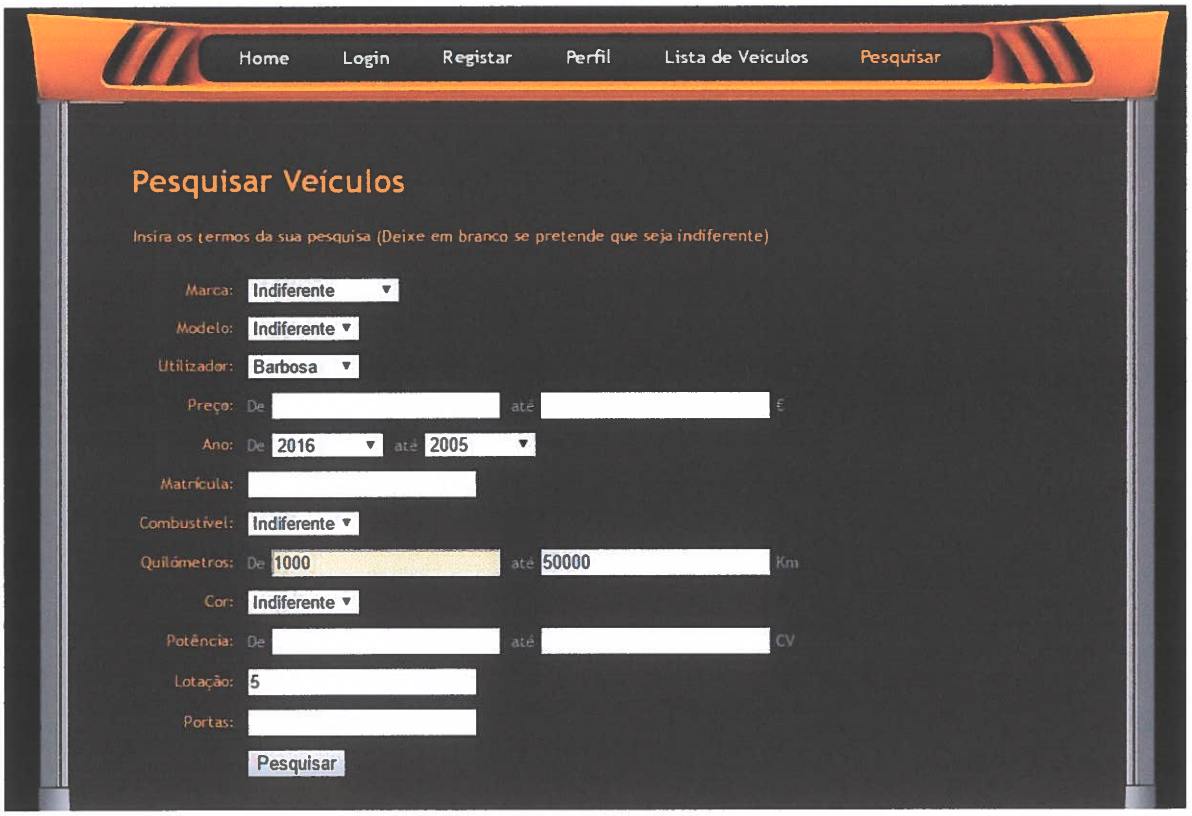

Fig. 14 - A página de pesquisa na aplicação, demonstrado um exemplo de uma pesquisa que se pode realizar na aplicação.

Fig. 15 - Exemplo de um resultado de uma pesquisa, retornando dois veículos.

Conclusão

Neste trabalho, foi concluído que os objetivos foram atingidos. A aplicação realiza corretamente todas as funções planeadas, como a adição e gerência de utilizadores na base de dados assim como a pesquisa de veículos utilizando diversos parâmetros.

A aplicação serviu também para uma melhor compreensão de aplicações *web* assim como a utilização da plataforma ASP.NET, as suas vantagens e desvantagens e as suas inúmeras ferramentas disponibilizadas no *Visual Studio*, que permitiram um desenvolvimento mais eficiente da aplicação.

Referências Bibliográficas

O livro "*ASP.NET 4 Unleashed*" de Stephen Walther, Kevin Hoffman e Nate Dudek, em formato digital e língua inglesa, explica o conceito do ASP.NET, mais concretamente da sua quarta versão, e do *Framework .NET* assim como conceitos básicos sobre como criar e desenvolver aplicações em ASP.NET, os vários tipos de controlos e ferramentas disponíveis para o programador, criar e trabalhar com bases de dados e dicas sobre a segurança dos *websites*. (Walther, Hoffman, & Dudek, 2011)

O livro "*Beginning ASP.NET 4 in C# and VB*" de Imar Spaanjars, em formato digital e língua inglesa, explica como começar a desenvolver aplicações em ASP.NET com introduções ás diferentes linguagens utilizadas pelo ASP.NET e aos controlos e ferramentas da mesma. É especialmente orientado a programadores amadores ou não familiarizados com o ASP.NET. Também inclui informações detalhadas sobre como utilizar os controlos ASP.NET, melhorar a segurança das aplicações e utilizar e configurar bases de dados. (Spaanjaars, 2010)

O artigo "*An Architectural Introduction to Web Parts and ASP.NET*", em formato digital e língua inglesa, é um *paper* publicado pela Microsoft em 2003 que introduz o leitor às funcionalidades de uma página *web* de formulários e ao ASP.NET, contendo também explicações sobre como criar páginas *web* utilizando o *Visual Studio .NET 2003*. (Microsoft Corporation, 2003)

O livro "*ASP.NET 2.0 Upgrader's Guide*" de Doug Lowe e Joel Murach, em formato digital e língua inglesa, introduz o leitor às funcionalidades do ASP.NET 2.0, sendo principalmente orientado a utilizadores que utilizam versões anteriores do ASP.NET. O livro também contém detalhes sobre como utilizar as linguagens SQL e XML na plataforma e a utilização dos seus controlos. (Lowe & Murach, 2005)

O artigo "*ASP.NET Web Programming*" de Hans-Petter Halvorsen, em formato digital e língua inglesa, é um documento que explica o conceito e as funcionalidades do ASP.NET, do Visual Studio e das várias linguagens e ferramentas utilizadas no desenvolvimento de aplicações ASP.NET, contendo explicações mais simples e claras quando comparado com livros do mesmo assunto. (Halvorsen, 2014)

O livro "*Introducing ASP.NET Web Pages 2*" de Mike Pope e publicado pela Microsoft em língua inglesa, é um *e-book* (livro digital) que contém uma série de tutoriais com o objetivo de ensinar ao leitor como criar, desenvolver e utilizar páginas ASP.NET dinâmicas. (Pope, 2012).

O artigo "*Authentication and Security Mechanisms in ASP.NET Web Applications*", em formato digital e língua inglesa, é um *paper* publicado pela Manhoc Technologies em 2010 que contém definições e explicações sobre mecanismos de segurança e autenticação disponíveis para aplicações ASP.NET. Inclui também explicações sobre criptografia e mecanismos de segurança do .NET *Framework*. (Manhoc Technologies, 2010)

Apêndices

Código CSS da aplicação, no ficheiro StyleSheet.css:

```
body {
	margin: 0;
	padding: 0;
	line-height: 1.5em;
	font-family: "Trebuchet MS", Arial, Helvetica, sans-serif;
	font-size: 12px;
	color: #989898;
	background: #000000;
}

a:link, a:visited { color: #edad0f; text-decoration: none; font-weight: normal; }
a:active, a:hover { color: #edad0f; text-decoration: underline; }

.cleaner {
	clear: both;
	width: 100%;
	height: 1px;
	font-size: 1px;
}

p {
	margin: 0px;
	padding: 0px;
}

img {
	margin: 0px;
	padding: 0px;
	border: none;
}

.cleaner_w10 {
	float: left;
	width: 10px;
	height: 100%;
	font-size: 1px;
}

.cleaner_h10 {
	clear: both;
	width: 100%;
	height: 10px;
	font-size: 1px;
}

.cleaner_h20 {
	clear: both;
	width: 100%;
	height: 20px;
	font-size: 1px;
}

.cleaner_h30 {
	clear: both;
	width: 100%;
```

```
	height: 30px;
	font-size: 1px;
}

.cleaner_h40 {
	clear: both;
	width: 100%;
	height: 40px;
	font-size: 1px;
}

.cleaner_w50 {
	float: left;
	width: 50px;
	height: 1px;
	font-size: 1px;
}

.horizontal_divider_01 {
	clear: both;
	width: 100%;
	height: 2px;
	font-size: 1px;
	background: url(Imagens/templatemo_horizontal_divider.jpg) repeat-x;
}

.rc_btn_01 a {
	float: left;
	clear: both;
	display: block;
	width: 100px;
	height: 22px;
	padding: 4px 0 0 10px;
	background: url(Imagens/tempatemo_button_01.jpg) no-repeat;
	color: #000000;
	font-size: 14px;
	font-weight: bold;
	text-decoration: none;
}
.rc_btn_01 a:hover {
	color: #996600;
}

#templatemo_container {
	width: 900px;
	margin: 0 auto;
	padding: 0 30px 10px 30px;
}

#templatemo_header {
	width: 900px;
	height: 178px;
	margin: 0px;
	padding: 0px;
	background: url(Imagens/templatemo_banner_blank.jpg) no-repeat;
}

#templatemo_header #website_title {
	width: 900px;
	height: 178px;
	padding: 100px 0 0 0;
	text-align: center;
```

```
}

#website_title #title {
	font-size: 50px;
	font-weight: bold;
	color: #ffffff;
	margin-bottom: 15px;
}

#website_title #salgon {
	clear: both;
	font-size: 18px;
	font-weight: bold;
	color:#cccccc;
}

.userinfo {
    color: white;
    text-align: right;
    margin-right: 10%;
}

.tituloveiculo {
    font-size: 20px;
    margin-bottom: 10px;
    color: #edad0f;
}

.tituloveiculoperfil {
    font-size: 20px;
    margin-top: 10px;
    margin-bottom: 5px;
    color: #edad0f;
}

/* banner */
#templatemo_menu{
	float: left;
	width: 900px;
	height: 70px;
	background: url(Imagens/templatemo_menu_bg.jpg) no-repeat;
}

#templatemo_menu ul {
	width: 600px;
	margin: 26px 0 0 160px;
	padding: 0px;
	list-style: none;
}

#templatemo_menu ul li{
	display: inline;
}

#templatemo_menu ul li a{
	float: left;
	padding: 0 20px;
	font-size: 14px;
	text-align: center;
	text-decoration: none;
	background: url(Imagens/templatemo_menu_divider.gif) top right repeat-y;
	color: #ffffff;
```

```
	font-weight: bold;
	outline: none;
}

#templatemo_menu li a:hover, #templatemo_menu li .current{
	color: #edad0f;
}
/* end of menu*/
/* end of banner */

/* content */

#templatemo_content_wrapper {
	clear: both;
	width: 900px;
	margin:0 auto;
	background: url(Imagens/templatemo_content_repeat.jpg) repeat-y;
}

#templatemo_content {
	padding: 50px 100px 0 100px;
	background: url(images/templatemo_content_top.jpg) top center no-repeat;
}

#templatemo_content img{
	float: left;
	margin: 3px 15px 5px 0;
	border: 5px solid #10100f;
}

#templatemo_content p{
	text-align: justify;
	margin-bottom: 20px;
}

#templatemo_content .content_title_01{
	color: #edad0f;
	font-size: 24px;
	padding-bottom: 10px;
	margin-bottom: 15px;
	font-weight: bold;
}

#templatemo_content .content_title_02{
	color: #edad0f;
	font-size: 18px;
	padding-bottom: 10px;
	margin-bottom: 15px;
	font-weight: bold;
}

#templatemo_content .column_02 {
	float: left;
	width: 325px;
}

#templatemo_content .news_section {
	padding-left: 30px;
	background:url(Imagens/templatemo_icon_01.jpg) top left no-repeat;
}
```

```
#templatemo_content  .column_02 p{
	margin-bottom: 0px;
}

#templatemo_content .column_02 ul {
	margin: 20px 0 0 30px;
	padding: 0px;
}

#templatemo_content .column_02 li {
	padding: 0 0 5px 0;
}

#templatemo_content .news_title {
	font-weight: bold;
	color: #edad0f;
}

/* footer */
#templatemo_footer {
	clear: both;
	width: 900px;
	height: 35px;
	padding-top: 55px;
	color: #eca90e;
	text-align: center;
	background: url(Imagens/templatemo_footer.jpg) no-repeat;
}

#templatemo_footer a{
	color: #eca90e;
	font-weight: bold;
}
/* end of footer */
```

Código HTML da *Master Page*:

```
<%@ Master Language="C#" AutoEventWireup="true" CodeFile="MasterPage.master.cs"
Inherits="MasterPage" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Stand de Automóveis</title>
    <link rel="stylesheet" href="StyleSheet.css" />
    <link rel="shortcut icon" type="image/x-icon" href="Icon.ico" />
    <asp:ContentPlaceHolder ID="head" runat="server">
    </asp:ContentPlaceHolder>
</head>
<body>
    <div id="templatemo_container">
        <form id="form1" runat="server">
            <div id="templatemo_header">
                <div id="website_title">
                    <h1 style="color: white;">Stand de Automóveis</h1>

                    <div class="userinfo">
                        <asp:Label ID="lblUtilizador" runat="server"></asp:Label>
                         
```

```
                        <asp:Button ID="btnLogout" runat="server" Text="Logout"
OnClick="btnLogout_Click" />
                    </div>
                </div>
            </div>
            <!-- end of header -->

            <div id="templatemo_menu">
                <ul>
                    <li><a href="Home.aspx" id="menuHome"
class="current">Home</a></li>
                    <li><a href="Login.aspx" id="menuLogin">Login</a></li>
                    <li><a href="Registar.aspx" id="menuRegistar">Registar</a></li>
                    <li><a href="Perfil.aspx" id="menuPerfil">Perfil</a></li>
                    <li><a href="Lista.aspx" id="menuLista">Lista de Veículos</a></li>
                    <li><a href="Pesquisar.aspx" id="menuPesquisar">Pesquisar</a></li>
                </ul>
            </div>
            <!-- end of menu -->

            <div id="templatemo_content_wrapper">

                <div id="templatemo_content">
                    <asp:ContentPlaceHolder ID="ContentPlaceHolder1" runat="server">
                    </asp:ContentPlaceHolder>
                </div>

                <div class="cleaner"> </div>
            </div>

            <div id="templatemo_footer">
                Website criado por Pedro Cerqueira Nº 1900 para o Projecto Global
            </div>
        </form>
    </div>
</body>
</html>
```

Código C# da *Master Page*:

```
public partial class MasterPage : System.Web.UI.MasterPage
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Session["Nome"].ToString() != "")
        {
            lblUtilizador.Text = Session["Nome"].ToString();
        }
        else
        {
            btnLogout.Visible = false;
        }
    }

    protected void btnLogout_Click(object sender, EventArgs e)
    {
        Session["Nome"] = "";
        Session["Password"] = "";
        lblUtilizador.Text = "";
        btnLogout.Visible = false;
        Response.Redirect("Login.aspx");
```

```
    }
}
```

Código HTML da *Home Page*:

```
<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="Home.aspx.cs" Inherits="Home" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
    <script>
        window.onload = function () {
            document.getElementsByClassName("current").item(0).className = "";
            document.getElementById("menuHome").className = "current";
        }
    </script>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">
    <div class="content_title_01">Bem-Vindo ao Stand de Automóveis</div>
    <br />
    <div class="home">
        <p>Se pretende fazer o login <a href="Login.aspx">clique aqui</a>.</p>
        <p>Não tem conta? <a href="Registar.aspx">Registe-se</a>!</p>
        <p>Veja a lista de todos os veiculos disponíveis <a
href="Lista.aspx">aqui</a>.</p>
        <p>Deseja adicionar novos veículos, ou editar ou remover veículos que já
adicionou? Registe-se e veja o seu perfil <a href="Perfil.aspx">aqui</a>.</p>
        <p>Se pretender procurar veículos mais específicos, <a
href="Pesquisar.aspx">pesquise aqui</a>.</p>
    </div>
</asp:Content>
```

Código HTML da página do *Login*:

```
<%@ Page Title="Login" Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="Login.aspx.cs" Inherits="Login" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="Server">
    <script>
        window.onload = function () {
            document.getElementsByClassName("current").item(0).className = "";
            document.getElementById("menuLogin").className = "current";
        }
    </script>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="Server">
    <div class="content_title_01">Login</div>
    <asp:Panel runat="server" DefaultButton="btnLogin" Width="100%">
        <table cellpadding="5px" cellspacing="0">
            <tr>
                <td>
                    <asp:Label ID="lblNomeUtilizador" runat="server"
BackColor="Transparent" Text="Nome:"></asp:Label>
                </td>
                <td>
                    <asp:TextBox ID="txtNomeUtilizador" Width="180px"
runat="server"></asp:TextBox>
                </td>
            </tr>
            <tr>
                <td>
```

```
                    <asp:Label ID="lblPassword" runat="server" BackColor="Transparent"
Text="Password:"></asp:Label>
                </td>
                <td>
                    <asp:TextBox ID="txtPassword" Width="180px" TextMode="Password"
runat="server"></asp:TextBox>
                </td>
            </tr>
            <tr>
                <td></td>
                <td>
                    <asp:Button ID="btnLogin" runat="server" Text="Login"
Width="184px" OnClick="btnLogin_Click" Style="margin-left: 0px" />
                </td>
            </tr>
            <tr>
                <td colspan="2">
                    <div style="text-align: center"><asp:Label ID="lblError"
runat="server" Width="250px" ForeColor="Red"></asp:Label></div>
                </td>
            </tr>
        </table>
    </asp:Panel>
</asp:Content>
```

Código C# da página do *Login*:

```
public partial class Login : System.Web.UI.Page
{
    UtilizadoresTableAdapter adpt = new UtilizadoresTableAdapter();
    DataSet.UtilizadoresDataTable tab;
    DataSet.UtilizadoresRow r;

    protected void Page_Load(object sender, EventArgs e)
    {
        txtNomeUtilizador.Focus();
    }

    protected void btnLogin_Click(object sender, EventArgs e)
    {
        //Verifica se nome e password for introduzidos
        if (txtNomeUtilizador.Text.Length == 0)
        {
            lblError.Text = "É necessário introduzir o Nome.";
            lblError.Visible = true;
            return;
        }
        if (txtPassword.Text.Length == 0)
        {
            lblError.Text = "É necessário introduzir a Password.";
            lblError.Visible = true;
            return;
        }

        //Preencher a tabela com os dados dos utilizadores que correspondem ao
introduzido
        tab = adpt.GetUtilizadorByNomePass(txtNomeUtilizador.Text, txtPassword.Text);
        tab.AcceptChanges();

        //Verificar se encontrou ou não nome e password introduzidos e se sim fazer o
login
```

```
        if (tab.Rows.Count == 0) //Não Encontrou
        {
            lblError.Text = "O nome e/ou password digitados estão incorrectos. Por
favor, tente de novo.";
            lblError.Visible = true;
            return;
        }
        else //Encontrou
        {
            r = (DataSet.UtilizadoresRow)tab.Rows[0];
            Session["Nome"] = r.Nome;
            Session["Password"] = r.Password;
            adpt.Dispose();
            if (Session["Redirect"].ToString() != "")
            {
                string redirect = Session["Redirect"].ToString();
                Session["Redirect"] = "";
                Response.Redirect(redirect);
            }
            else
                Response.Redirect("Home.aspx");
        }
    }
}
```

Código HTML da página de registo:

```
<%@ Page Title="Registar Utilizador" Language="C#"
MasterPageFile="~/MasterPage.master" AutoEventWireup="true"
CodeFile="Registar.aspx.cs" Inherits="Registar" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="Server">
    <script>
        window.onload = function () {
            document.getElementsByClassName("current").item(0).className = "";
            document.getElementById("menuRegistar").className = "current";
        }
    </script>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="Server">
    <div class="content_title_01">Registar Utilizador</div>
    <div class="news_title">Insira o nome e a password do novo utilizador. O nome não
pode já existir.</div>
    <br />
    <asp:Panel runat="server" DefaultButton="btnRegistar" Width="100%">
        <table cellpadding="5px" cellspacing="0">
            <tr>
                <td>
                    <asp:Label ID="lblNomeUtilizador" runat="server"
BackColor="Transparent" Text="Nome:"></asp:Label>
                </td>
                <td>
                    <asp:TextBox ID="txtNomeUtilizador" Width="180px"
runat="server"></asp:TextBox>
                </td>
            </tr>
            <tr>
                <td>
                    <asp:Label ID="lblPassword" runat="server" BackColor="Transparent"
Text="Password:"></asp:Label>
                </td>
```

```
                <td>
                    <asp:TextBox ID="txtPassword" Width="180px" TextMode="Password"
runat="server"></asp:TextBox>
                </td>
            </tr>
            <tr>
                <td></td>
                <td>
                    <asp:Button ID="btnRegistar" runat="server" Text="Registar
Utilizador" Width="184px" OnClick="btnRegistar_Click" Style="margin-left: 0px" />
                </td>
            </tr>
            <tr>
                <td colspan="2">
                    <div style="text-align: center"><asp:Label ID="lblError"
runat="server" Width="250px" ForeColor="Red"></asp:Label></div>
                </td>
            </tr>
        </table>
    </asp:Panel>
</asp:Content>
```

Código C# da página de registo:

```
public partial class Registar : System.Web.UI.Page
{
    UtilizadoresTableAdapter adpt = new UtilizadoresTableAdapter();
    DataSet.UtilizadoresDataTable tab;

    protected void Page_Load(object sender, EventArgs e)
    {
        txtNomeUtilizador.Focus();
    }

    protected void btnRegistar_Click(object sender, EventArgs e)
    {
        //Verifica se nome e password for introduzidos
        if (txtNomeUtilizador.Text.Length == 0)
        {
            lblError.Text = "É necessário introduzir o Nome.";
            lblError.Visible = true;
            return;
        }
        if (txtPassword.Text.Length == 0)
        {
            lblError.Text = "É necessário introduzir a Password.";
            lblError.Visible = true;
            return;
        }

        //Verifica se já existe um utilizador com o mesmo nome
        tab = adpt.GetUtilizadorByNome(txtNomeUtilizador.Text);
        tab.AcceptChanges();
        if (tab.Rows.Count == 1)
        {
            lblError.Text = "Já existe um utilizador com esse nome. Escolha outro.";
            lblError.Visible = true;
            return;
        }
```

```
        //Se não existir então registar e adicionar o utilizador á base de dados e
automaticamente fazer o login
        adpt.Insert(txtNomeUtilizador.Text, txtPassword.Text);
        adpt.FillUtilizadores(tab);
        tab.AcceptChanges();
        adpt.Dispose();

        Session["Nome"] = txtNomeUtilizador.Text; ;
        Session["Password"] = txtPassword.Text;
        Response.Redirect("Home.aspx");
    }
}
```

Código HTML da página de perfil do utilizador:

```
<%@ Page Title="Perfil" Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="Perfil.aspx.cs"
MaintainScrollPositionOnPostback="true" Inherits="Perfil" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="Server">
    <script>
        window.onload = function () {
            document.getElementsByClassName("current").item(0).className = "";
            document.getElementById("menuPerfil").className = "current";
        }
    </script>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="Server">
    <div class="content_title_01">Perfil</div>
    <asp:Label runat="server" Text="Nome:"></asp:Label>
     
        <asp:Label ID="lblNome" runat="server" />
    <br />
    <asp:Label runat="server" Text="Password:"></asp:Label>
     
        <asp:Label ID="lblPassword" runat="server" />
    <br />
    <br />
    <asp:Button ID="btnEditar" runat="server" Text="Editar Perfil"
OnClick="btnEditar_Click" />
     
        <asp:Button ID="btnEliminar" runat="server" Text="Eliminar Utilizador"
OnClick="btnEliminar_Click" />
    <br />
    <br />
    <br />
    <br />
    <div class="content_title_02">Os Meus Veículos</div>
    <asp:Label ID="lblVazio" runat="server" Text="Não tem veículos adicionados.
Experimente adicionar um preenchendo os campos abaixo." CssClass="error"></asp:Label>
    <asp:DataList ID="dlVeiculos" Width="650px" ShowFooter="true" runat="server"
RepeatColumns="1" OnItemCommand="dlVeiculos_ItemCommand">
        <ItemTemplate>
            <div id="itemtemplate">
                <div class="tituloveiculoperfil"><%#
devolveMarca((int)DataBinder.Eval(Container.DataItem, "Marca")) %> <%#
devolveModelo((int)DataBinder.Eval(Container.DataItem, "Modelo")) %></div>
                <asp:Image ID="Image1"
                    Width="350px"
                    Height="175px"
                    CssClass="img"
```

```
                    ImageUrl='<%# DataBinder.Eval(Container.DataItem, "Id",
"FotoVeiculos.ashx?id={0}") %>'
                    runat="server" />
                <font color="#edad0f">Preço:</font> <%#
DataBinder.Eval(Container.DataItem, "Preco") %> €<br />
                <font color="#edad0f">Ano:</font> <%#
devolveAno((DateTime)DataBinder.Eval(Container.DataItem, "Ano")) %><br />
                <font color="#edad0f">Matrícula:</font> <%#
DataBinder.Eval(Container.DataItem, "Matricula") %><br />
                <font color="#edad0f">Combustível:</font> <%#
DataBinder.Eval(Container.DataItem, "Combustivel") %><br />
                <font color="#edad0f">Quilómetros:</font> <%#
DataBinder.Eval(Container.DataItem, "Quilometros") %><br />
                <font color="#edad0f">Cor:</font> <%#
DataBinder.Eval(Container.DataItem, "Cor") %><br />
                <font color="#edad0f">Potência:</font> <%#
DataBinder.Eval(Container.DataItem, "Potencia") %> CV<br />
                <font color="#edad0f">Lotação:</font> <%#
DataBinder.Eval(Container.DataItem, "Lotacao") %><br />
                <font color="#edad0f">Portas:</font> <%#
DataBinder.Eval(Container.DataItem, "Portas") %><br />
                <font color="#edad0f">Adicionado a:</font> <%#
devloveData((DateTime)DataBinder.Eval(Container.DataItem, "Data")) %><br />
                <br />
                <asp:Button ID="btnEditar" CommandName="edit" runat="server"
Text="Editar" />
                <asp:Button ID="btnRemover" CommandName="delete" runat="server"
Text="Remover" />
                <hr color="#edad0f" />
            </div>
        </ItemTemplate>

        <EditItemTemplate>
            <fieldset style="border: 1px solid #edad0f">
                <legend><font color="#edad0f">Edição de Veículo</font></legend>
                <font color="#edad0f">Marca:</font>
                    <asp:DropDownList ID="dropMarca" Text='<%#
DataBinder.Eval(Container.DataItem, "Marca") %>' OnDataBinding="listarMarcas"
AutoPostBack="true" OnSelectedIndexChanged="dropMarca_SelectedIndexChanged"
runat="server"></asp:DropDownList>
                <br />
                <br />
                <font color="#edad0f">Modelo:</font>
                    <asp:DropDownList ID="dropModelo" Text='<%#
DataBinder.Eval(Container.DataItem, "Modelo") %>' OnDataBinding="listarModelos"
runat="server"></asp:DropDownList>
                <br />
                <br />
                <font color="#edad0f">Preço (€):</font>
                    <asp:TextBox ID="txtPreco" Text='<%#
DataBinder.Eval(Container.DataItem, "Preco") %>' runat="server"></asp:TextBox>
                <br />
                <br />
                <font color="#edad0f">Ano:</font>
                    <asp:DropDownList ID="dropAno" Text='<%#
devolveAno((DateTime)DataBinder.Eval(Container.DataItem, "Ano")) %>'
OnDataBinding="listarAnos" runat="server"></asp:DropDownList>
                <br />
                <br />
                <font color="#edad0f">Matrícula:</font>
                    <asp:TextBox ID="txtMatricula" Text='<%#
DataBinder.Eval(Container.DataItem, "Matricula") %>' runat="server"></asp:TextBox>
```

```
                <br />
                <br />
                <font color="#edad0f">Combustível:</font>
                    <asp:DropDownList ID="dropCombustivel" Text='<%#
DataBinder.Eval(Container.DataItem, "Combustivel") %>'
OnDataBinding="listarCombustiveis" runat="server"></asp:DropDownList>
                <br />
                <br />
                <font color="#edad0f">Quilómetros:</font>
                    <asp:TextBox ID="txtQuilometros" Text='<%#
DataBinder.Eval(Container.DataItem, "Quilometros") %>' TextMode="Number"
runat="server"></asp:TextBox>
                <br />
                <br />
                <font color="#edad0f">Cor:</font>
                    <asp:DropDownList ID="dropCor" Text='<%#
DataBinder.Eval(Container.DataItem, "Cor") %>' OnDataBinding="listarCores"
runat="server"></asp:DropDownList>
                <br />
                <br />
                <font color="#edad0f">Potência (CV):</font>
                    <asp:TextBox ID="txtPotencia" Text='<%#
DataBinder.Eval(Container.DataItem, "Potencia") %>' TextMode="Number"
runat="server"></asp:TextBox>
                <br />
                <br />
                <font color="#edad0f">Lotação:</font>
                    <asp:TextBox ID="txtLotacao" Text='<%#
DataBinder.Eval(Container.DataItem, "Lotacao") %>' TextMode="Number"
runat="server"></asp:TextBox>
                <br />
                <br />
                <font color="#edad0f">Portas:</font>
                    <asp:TextBox ID="txtPortas" Text='<%#
DataBinder.Eval(Container.DataItem, "Portas") %>' TextMode="Number"
runat="server"></asp:TextBox>
                <br />
                <br />
                <font color="#edad0f">Foto:</font>
                    <asp:FileUpload ID="fuFoto" runat="server" />
                    <br /><br />
                <hr color="#edad0f" />
                <asp:Button ID="btnEditar" CommandName="update" runat="server"
Text="Editar" />
                <asp:Button ID="btnCancelar" CommandName="cancel" runat="server"
Text="Cancelar" />
            </fieldset>
        </EditItemTemplate>

        <FooterTemplate>
            <br />
            <fieldset style="border: 1px solid #edad0f">
                <legend><font color="#edad0f">Adicionar novo Veículo</font></legend>
                <font color="#edad0f">Marca:</font>
                    <asp:DropDownList ID="dropMarca" OnDataBinding="listarMarcas"
AutoPostBack="true" OnSelectedIndexChanged="dropMarca_SelectedIndexChanged"
runat="server"></asp:DropDownList>
                <br />
                <br />
                <font color="#edad0f">Modelo:</font>
                    <asp:DropDownList ID="dropModelo" OnDataBinding="listarModelos"
runat="server"></asp:DropDownList>
```

```
                <br />
                <br />
                <font color="#edad0f">Preço (€):</font>
                    <asp:TextBox ID="txtPreco" TextMode="Number" Text=""
runat="server"></asp:TextBox>
                <br />
                <br />
                <font color="#edad0f">Ano:</font>
                    <asp:DropDownList ID="dropAno" OnDataBinding="listarAnos"
runat="server"></asp:DropDownList>
                <br />
                <br />
                <font color="#edad0f">Matrícula:</font>
                    <asp:TextBox ID="txtMatricula" Text=""
runat="server"></asp:TextBox>
                <br />
                <br />
                <font color="#edad0f">Combustível:</font>
                    <asp:DropDownList ID="dropCombustivel"
OnDataBinding="listarCombustiveis" runat="server"></asp:DropDownList>
                <br />
                <br />
                <font color="#edad0f">Quilómetros:</font>
                    <asp:TextBox ID="txtQuilometros" Text="" TextMode="Number"
runat="server"></asp:TextBox>
                <br />
                <br />
                <font color="#edad0f">Cor:</font>
                    <asp:DropDownList ID="dropCor" OnDataBinding="listarCores"
runat="server"></asp:DropDownList>
                <br />
                <br />
                <font color="#edad0f">Potência (CV):</font>
                    <asp:TextBox ID="txtPotencia" Text="" TextMode="Number"
runat="server"></asp:TextBox>
                <br />
                <br />
                <font color="#edad0f">Lotação:</font>
                    <asp:TextBox ID="txtLotacao" Text="" TextMode="Number"
runat="server"></asp:TextBox>
                <br />
                <br />
                <font color="#edad0f">Portas:</font>
                    <asp:TextBox ID="txtPortas" Text="" TextMode="Number"
runat="server"></asp:TextBox>
                <br />
                <br />
                <font color="#edad0f">Foto:</font>
                    <asp:FileUpload ID="fuFoto" runat="server" />
                <br />
                <br />
                <asp:Label ID="lblError" runat="server" Width="100%"
ForeColor="Red"></asp:Label>
                <hr color="#edad0f" />
                <asp:Button ID="btnAdicionar" CommandName="insert" runat="server"
Text="Adicionar" />
            </fieldset>
        </FooterTemplate>
    </asp:DataList>
</asp:Content>
```

Código C# da página de perfil do utilizador:

```
public partial class Perfil : System.Web.UI.Page
{
    VeiculosTableAdapter adpt;
    DataSet.VeiculosDataTable tab;
    bool r;

    protected void Page_Load(object sender, EventArgs e)
    {
        this.PreRender += Perfil_PreRender;

        //Verificar se fez login, senão redireccionar
        if (Session["Nome"].ToString() == "")
        {
            Session["Redirect"] = "Perfil.aspx";
            Response.Redirect("Login.aspx");
        }

        lblNome.Text = Session["Nome"].ToString();
        lblPassword.Text = Session["Password"].ToString();

        if (Page.IsPostBack)
        {
            tab = (DataSet.VeiculosDataTable)ViewState["tab"];
        }
        else
        {
            vaiBD();
        }

        if (dlVeiculos.Items.Count > 0)
            lblVazio.Visible = false;
    }

    void Perfil_PreRender(object sender, EventArgs e)
    {
        ViewState["tab"] = tab;
    }

    protected void bindDatalist()
    {
        dlVeiculos.DataSource = tab.DefaultView;
        dlVeiculos.DataKeyField = "Id";
        dlVeiculos.DataBind();
    }

    protected void vaiBD()
    {
        adpt = new VeiculosTableAdapter();
        tab = adpt.GetVeiculosByUtilizador(Session["Nome"].ToString());
        tab.AcceptChanges();
        tab.PrimaryKey = new DataColumn[] { tab.Columns["Id"] };
        tab.Columns["Id"].AutoIncrement = true;
        tab.Columns["Id"].AutoIncrementSeed = 1;
        tab.Columns["Id"].AutoIncrementStep = 1;
        bindDatalist();
    }

    protected void ApagarFoto(string nome)
    {
        string caminho = Server.MapPath("~/Fotos/") + nome;
```

```
        if (System.IO.File.Exists(caminho))
        {
            System.IO.File.Delete(caminho);
        }
    }

    protected void dlVeiculos_ItemCommand(object source, DataListCommandEventArgs e)
    {
        DataList dl = (DataList)source;
        DataSet.VeiculosRow r;
        switch (e.CommandName)
        {
            case "edit":
                dl.EditItemIndex = e.Item.ItemIndex;
                bindDatalist();
                break;
            case "cancel":
                dl.EditItemIndex = -1;
                bindDatalist();
                break;
            case "delete":
                System.Windows.Forms.DialogResult resposta =
System.Windows.Forms.MessageBox.Show("Tem a certeza que deseja eliminar este veículo?"
+ Environment.NewLine + "Não poderá voltar atrás.", "Eliminar Veículo",
System.Windows.Forms.MessageBoxButtons.YesNo,
System.Windows.Forms.MessageBoxIcon.Warning);
                if (resposta == System.Windows.Forms.DialogResult.Yes) //Se respondeu
"Sim" então eliminar e fazer o logout
                {
                    r =
(DataSet.VeiculosRow)tab.FindById((int)dl.DataKeys[e.Item.ItemIndex]);
                    if (r != null)
                    {
                        if (!r.IsFotoNull()) ApagarFoto(r.Foto);
                        r.Delete();
                    }
                    adpt = new VeiculosTableAdapter();
                    adpt.Update(tab);
                    vaiBD();
                    if (dlVeiculos.Items.Count == 0)
                        lblVazio.Visible = true;
                }
                break;
            case "update":
                adpt = new VeiculosTableAdapter();
                r =
(DataSet.VeiculosRow)tab.FindById((int)dl.DataKeys[e.Item.ItemIndex]);
                if (r != null)
                {
                    r.Marca =
int.Parse(((DropDownList)e.Item.FindControl("dropMarca")).SelectedValue);
                    r.Modelo =
int.Parse(((DropDownList)e.Item.FindControl("dropModelo")).SelectedValue);
                    r.Preco =
int.Parse(((TextBox)e.Item.FindControl("txtPreco")).Text);
                    DateTime data = new
DateTime(int.Parse(((DropDownList)e.Item.FindControl("dropAno")).SelectedValue), 1,
1); //Criar data com ano especificado
                    r.Ano = data;
                    r.Matricula = ((TextBox)e.Item.FindControl("txtMatricula")).Text;
                    r.Combustivel =
((DropDownList)e.Item.FindControl("dropCombustivel")).SelectedValue;
```

```
                    r.Quilometros =
int.Parse(((TextBox)e.Item.FindControl("txtQuilometros")).Text);
                    r.Cor =
((DropDownList)e.Item.FindControl("dropCor")).SelectedValue;
                    r.Potencia =
int.Parse(((TextBox)e.Item.FindControl("txtPotencia")).Text);
                    r.Lotacao = ((TextBox)e.Item.FindControl("txtLotacao")).Text;
                    r.Portas = ((TextBox)e.Item.FindControl("txtPortas")).Text;
                    HttpPostedFile fich =
((FileUpload)e.Item.FindControl("fuFoto")).PostedFile;
                    if (fich.FileName.Length > 0 &&
fich.ContentType.Contains("image"))
                    {
                        int novoid = r.Id;
                        string caminho = Server.MapPath("~/Fotos/") +
novoid.ToString();
                        caminho += System.IO.Path.GetExtension(fich.FileName);
                        if (!r.IsFotoNull()) ApagarFoto(r.Foto);
                        fich.SaveAs(caminho);
                        r.Foto = novoid.ToString() +
System.IO.Path.GetExtension(fich.FileName);
                        r.FotoTipo = fich.ContentType;
                    }
                }
                adpt.Update(tab);
                dl.EditItemIndex = -1;
                vaiBD();
                break;
            case "insert":
                //Verificar se todos os campos (excepto foto) foram preenchidos
                if (((DropDownList)e.Item.FindControl("dropMarca")).SelectedIndex == -
1 ||
                    ((DropDownList)e.Item.FindControl("dropModelo")).SelectedIndex ==
-1 ||
                    ((TextBox)e.Item.FindControl("txtPreco")).Text == "" ||
                    ((DropDownList)e.Item.FindControl("dropAno")).SelectedIndex == -1
||
                    ((TextBox)e.Item.FindControl("txtMatricula")).Text == "" ||

((DropDownList)e.Item.FindControl("dropCombustivel")).SelectedIndex == -1 ||
                    ((TextBox)e.Item.FindControl("txtQuilometros")).Text == "" ||
                    ((DropDownList)e.Item.FindControl("dropCor")).SelectedIndex == -1
||
                    ((TextBox)e.Item.FindControl("txtPotencia")).Text == "" ||
                    ((TextBox)e.Item.FindControl("txtLotacao")).Text == "" ||
                    ((TextBox)e.Item.FindControl("txtPortas")).Text == "")
                {
                    ((Label)e.Item.FindControl("lblError")).Text = "Preencha todos os
campos.";
                }
                else
                {
                    adpt = new VeiculosTableAdapter();
                    r = (DataSet.VeiculosRow)tab.NewRow();
                    //Adição do Veículo
                    DataSet.VeiculosDataTable tab2;
                    tab2 = adpt.GetVeiculos();
                    if (tab2.Rows.Count == 0) //Verifica se tem ou não items na tabela
para calcular o ID
                    {
                        r.Id = 1;
                    }
```

```
                    else
                    {
                        r.Id = int.Parse(adpt.ProximoId().ToString());
                    }
                    r.Marca =
int.Parse(((DropDownList)e.Item.FindControl("dropMarca")).SelectedValue);
                    r.Modelo =
int.Parse(((DropDownList)e.Item.FindControl("dropModelo")).SelectedValue);
                    r.Utilizador = Session["Nome"].ToString();
                    r.Data = DateTime.Today;
                    r.Preco =
int.Parse(((TextBox)e.Item.FindControl("txtPreco")).Text);
                    DateTime data = new
DateTime(int.Parse(((DropDownList)e.Item.FindControl("dropAno")).SelectedValue), 1,
1); //Criar data com ano especificado
                    r.Ano = data;
                    r.Matricula = ((TextBox)e.Item.FindControl("txtMatricula")).Text;
                    r.Combustivel =
((DropDownList)e.Item.FindControl("dropCombustivel")).SelectedValue;
                    r.Quilometros =
int.Parse(((TextBox)e.Item.FindControl("txtQuilometros")).Text);
                    r.Cor =
((DropDownList)e.Item.FindControl("dropCor")).SelectedValue;
                    r.Potencia =
int.Parse(((TextBox)e.Item.FindControl("txtPotencia")).Text);
                    r.Lotacao = ((TextBox)e.Item.FindControl("txtLotacao")).Text;
                    r.Portas = ((TextBox)e.Item.FindControl("txtPortas")).Text;
                    HttpPostedFile fichn =
((FileUpload)e.Item.FindControl("fuFoto")).PostedFile;
                    if (fichn.FileName.Length > 0 &&
fichn.ContentType.Contains("image"))
                    {
                        string caminho = Server.MapPath("~/Fotos/") + r.Id.ToString();
                        caminho += System.IO.Path.GetExtension(fichn.FileName);
                        fichn.SaveAs(caminho);
                        r.Foto = r.Id.ToString() +
System.IO.Path.GetExtension(fichn.FileName);
                        r.FotoTipo = fichn.ContentType;
                    }
                    tab.Rows.Add(r);
                    adpt.Update(tab);
                    vaiBD();
                    lblVazio.Visible = false;
                }
                break;
            }
        }

        protected void listarMarcas(object sender, EventArgs e)
        {
            DropDownList drop = (DropDownList)sender;
            MarcasTableAdapter ta = new MarcasTableAdapter();
            DataSet.MarcasDataTable t;
            t = ta.GetMarcas();
            drop.Items.Clear();
            foreach (DataSet.MarcasRow r in t.Rows)
            {
                drop.Items.Add(new ListItem(r.Nome, r.Id.ToString()));
            }
        }

        protected void dropMarca_SelectedIndexChanged(object sender, EventArgs e)
```

```
    {
        DropDownList drop = (DropDownList)sender;
        if (drop.SelectedIndex > -1)
        {
            listarModelos(dlVeiculos.Controls[dlVeiculos.Controls.Count -
1].Controls[0].FindControl("dropModelo"), e);
        }
    }

    protected void listarModelos(object sender, EventArgs e)
    {
        DropDownList marca =
(DropDownList)dlVeiculos.Controls[dlVeiculos.Controls.Count -
1].Controls[0].FindControl("dropMarca");
        if (marca.SelectedIndex != -1)
        {
            DropDownList drop = (DropDownList)sender;
            ModelosTableAdapter ta = new ModelosTableAdapter();
            DataSet.ModelosDataTable t;
            t = ta.GetModelosByMarca(int.Parse(marca.SelectedValue));
            drop.Items.Clear();
            foreach (DataSet.ModelosRow r in t.Rows)
            {
                drop.Items.Add(new ListItem(r.Nome, r.Id.ToString()));
            }
        }
    }

    protected void listarAnos(object sender, EventArgs e)
    {
        DropDownList drop = (DropDownList)sender;
        drop.Items.Clear();
        for (int ano = 2016; ano >= 1950; ano--)
        {
            drop.Items.Add(new ListItem(ano.ToString()));
        }
    }

    protected void listarCombustiveis(object sender, EventArgs e)
    {
        DropDownList drop = (DropDownList)sender;
        drop.Items.Clear();
        drop.Items.Add(new ListItem("Gasolina"));
        drop.Items.Add(new ListItem("Diesel"));
        drop.Items.Add(new ListItem("GPL"));
        drop.Items.Add(new ListItem("Híbrido"));
        drop.Items.Add(new ListItem("Eléctrico"));
    }

    protected void listarCores(object sender, EventArgs e)
    {
        DropDownList drop = (DropDownList)sender;
        drop.Items.Clear();
        drop.Items.Add(new ListItem("Amarelo"));
        drop.Items.Add(new ListItem("Azul"));
        drop.Items.Add(new ListItem("Bege"));
        drop.Items.Add(new ListItem("Branco"));
        drop.Items.Add(new ListItem("Cinzento"));
        drop.Items.Add(new ListItem("Dourado"));
        drop.Items.Add(new ListItem("Laranja"));
        drop.Items.Add(new ListItem("Prata"));
        drop.Items.Add(new ListItem("Preto"));
```

```
            drop.Items.Add(new ListItem("Roxo"));
            drop.Items.Add(new ListItem("Verde"));
            drop.Items.Add(new ListItem("Vermelho"));
        }

        protected void btnEditar_Click(object sender, EventArgs e)
        {
            Response.Redirect("Editar.aspx");
        }

        protected void btnEliminar_Click(object sender, EventArgs e)
        {
            //Perguntar se o utilizador quer mesmo ser eliminado
            System.Windows.Forms.DialogResult resposta =
System.Windows.Forms.MessageBox.Show("Tem a certeza que deseja eliminar este
utilizador?" + Environment.NewLine + "Não poderá voltar atrás.", "Eliminar
Utilizador", System.Windows.Forms.MessageBoxButtons.YesNo,
System.Windows.Forms.MessageBoxIcon.Warning);
            if (resposta == System.Windows.Forms.DialogResult.Yes) //Se respondeu "Sim"
então eliminar e fazer o logout
            {
                UtilizadoresTableAdapter useradpt = new UtilizadoresTableAdapter();
                useradpt.Delete(Session["Nome"].ToString());
                useradpt.Dispose();

                Session["Nome"] = "";
                Session["Password"] = "";
                Response.Redirect("Home.aspx");
            }
        }

        public string devolveMarca(int id)
        {
            string descritivo = "";
            MarcasTableAdapter ta = new MarcasTableAdapter();
            DataSet.MarcasDataTable t;
            DataSet.MarcasRow r;
            t = ta.GetMarcaById(id);
            if (t.Rows.Count > 0)
            {
                r = (DataSet.MarcasRow)t.Rows[0];
                if (r["Nome"] != null) descritivo = r["Nome"].ToString();
            }
            return descritivo;
        }

        public string devolveModelo(int id)
        {
            string descritivo = "";
            ModelosTableAdapter ta = new ModelosTableAdapter();
            DataSet.ModelosDataTable t;
            DataSet.ModelosRow r;
            t = ta.GetModeloById(id);
            if (t.Rows.Count > 0)
            {
                r = (DataSet.ModelosRow)t.Rows[0];
                if (r["Nome"] != null) descritivo = r["Nome"].ToString();
            }
            return descritivo;
        }

        public int devolveAno(DateTime data)
```

```
    {
        return data.Year;
    }

    public string devloveData(DateTime data)
    {
        return data.ToLongDateString();
    }
}
```

Código HTML da página de edição do perfil:

```
<%@ Page Title="Editar Perfil" Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="Editar.aspx.cs" Inherits="Editar" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="Server">
    <script>
        window.onload = function () {
            document.getElementsByClassName("current").item(0).className = "";
        }
    </script>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="Server">
    <div class="content_title_01">Editar Perfil</div>
    <div class="news_title">Modifique o seu nome e a sua password como deseja. O nome
não pode ser igual ao de outro utilizador.</div>
    <br />
    <table cellpadding="5px" cellspacing="0">
        <tr>
            <td align="right">
                <asp:Label ID="lblNomeUtilizador" runat="server"
BackColor="Transparent" Text="Nome:"></asp:Label>
            </td>
            <td>
                <asp:TextBox ID="txtNomeUtilizador" Width="180px"
runat="server"></asp:TextBox>
            </td>
        </tr>
        <tr>
            <td align="right">
                <asp:Label ID="lblPassword" runat="server" BackColor="Transparent"
Text="Password:"></asp:Label>
            </td>
            <td>
                <asp:TextBox ID="txtPassword" Width="180px" TextMode="Password"
runat="server"></asp:TextBox>
            </td>
        </tr>
        <tr>
            <td align="right">
                <asp:Button ID="btnEditar" runat="server" Text="Editar"
OnClick="btnEditar_Click" />
            </td>
            <td>
                <asp:Button ID="btnCancelar" runat="server" Text="Cancelar"
OnClick="btnCancelar_Click" />
            </td>
        </tr>
        <tr>
            <td colspan="2">
```

```
                <div style="text-align: center"><asp:Label ID="lblError"
runat="server" Width="250px" ForeColor="Red"></asp:Label></div>
            </td>
        </tr>
    </table>
</asp:Content>
```

Código C# da página de edição do perfil:

```
public partial class Editar : System.Web.UI.Page
{
    UtilizadoresTableAdapter adpt = new UtilizadoresTableAdapter();
    DataSet.UtilizadoresDataTable tab;

    protected void Page_Load(object sender, EventArgs e)
    {
        if (!Page.IsPostBack)
        {
            txtNomeUtilizador.Text = Session["Nome"].ToString();
            txtPassword.Text = Session["Password"].ToString();
        }
    }

    protected void btnEditar_Click(object sender, EventArgs e)
    {
        //Verifica se nome e password for introduzidos
        if (txtNomeUtilizador.Text.Length == 0)
        {
            lblError.Text = "É necessário introduzir o Nome.";
            lblError.Visible = true;
            return;
        }
        if (txtPassword.Text.Length == 0)
        {
            lblError.Text = "É necessário introduzir a Password.";
            lblError.Visible = true;
            return;
        }

        //Verifica se já existe um utilizador com o mesmo nome caso o utilizador o
tenha modificado
        if (txtNomeUtilizador.Text != Session["Nome"].ToString())
        {
            tab = adpt.GetUtilizadorByNome(txtNomeUtilizador.Text);
            tab.AcceptChanges();
            if (tab.Rows.Count == 1)
            {
                lblError.Text = "Já existe um utilizador com esse nome. Escolha
outro.";
                lblError.Visible = true;
                return;
            }
        }

        //Se não existir então proceder a edição
        adpt.Update(txtNomeUtilizador.Text, txtPassword.Text,
Session["Nome"].ToString());
        tab = adpt.GetUtilizadores();
        tab.AcceptChanges();
        adpt.Dispose();
```

```
        Session["Nome"] = txtNomeUtilizador.Text; ;
        Session["Password"] = txtPassword.Text;
        Response.Redirect("Perfil.aspx");
    }

    protected void btnCancelar_Click(object sender, EventArgs e)
    {
        Response.Redirect("Perfil.aspx");
    }
}
```

Código HTML da página da lista de veículos:

```
<%@ Page Title="Lista de Veículos" Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="Lista.aspx.cs"
MaintainScrollPositionOnPostback="true" Inherits="Lista" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="Server">
    <script>
        window.onload = function () {
            document.getElementsByClassName("current").item(0).className = "";
            document.getElementById("menuLista").className = "current";
        }
    </script>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="Server">
    <div class="content_title_01">Lista de Veículos</div>
    <asp:Label ID="lblVazio" runat="server" Text="Ainda não existem veículos
disponíveis." ForeColor="Red"></asp:Label>
    <asp:DataList ID="dlVeiculos" Width="650px" ShowFooter="false" runat="server"
RepeatColumns="1">
        <ItemTemplate>
            <div id="itemtemplate">
                <hr color="#edad0f" />
                <br />
                <div class="tituloveiculo"><%#
devolveMarca((int)DataBinder.Eval(Container.DataItem, "Marca")) %> <%#
devolveModelo((int)DataBinder.Eval(Container.DataItem, "Modelo")) %></div>
                <asp:Image ID="Image1"
                    Width="350px"
                    Height="175px"
                    CssClass="img"
                    ImageUrl='<%# DataBinder.Eval(Container.DataItem, "Id",
"FotoVeiculos.ashx?id={0}") %>'
                    runat="server" />
                <font color="#edad0f">Vendedor:</font> <%#
DataBinder.Eval(Container.DataItem, "Utilizador") %><br />
                <font color="#edad0f">Preço:</font> <%#
DataBinder.Eval(Container.DataItem, "Preco") %> €<br />
                <font color="#edad0f">Ano:</font> <%#
devolveAno((DateTime)DataBinder.Eval(Container.DataItem, "Ano")) %><br />
                <font color="#edad0f">Matrícula:</font> <%#
DataBinder.Eval(Container.DataItem, "Matricula") %><br />
                <font color="#edad0f">Combustível:</font> <%#
DataBinder.Eval(Container.DataItem, "Combustivel") %><br />
                <font color="#edad0f">Quilómetros:</font> <%#
DataBinder.Eval(Container.DataItem, "Quilometros") %><br />
                <font color="#edad0f">Cor:</font> <%#
DataBinder.Eval(Container.DataItem, "Cor") %><br />
                <font color="#edad0f">Potência:</font> <%#
DataBinder.Eval(Container.DataItem, "Potencia") %> CV<br />
```

```
                <font color="#edad0f">Lotação:</font> <%#
DataBinder.Eval(Container.DataItem, "Lotacao") %><br />
                <font color="#edad0f">Portas:</font> <%#
DataBinder.Eval(Container.DataItem, "Portas") %><br />
                <font color="#edad0f">Adicionado a:</font> <%#
devloveData((DateTime)DataBinder.Eval(Container.DataItem, "Data")) %><br />
            </div>
            <br />
        </ItemTemplate>
    </asp:DataList>
</asp:Content>
```

Código C# da página da lista de veículos:

```
public partial class Lista : System.Web.UI.Page
{
    VeiculosTableAdapter adpt;
    DataSet.VeiculosDataTable tab;

    protected void Page_Load(object sender, EventArgs e)
    {
        this.PreRender += Lista_PreRender;

        if (Page.IsPostBack)
        {
            tab = (DataSet.VeiculosDataTable)ViewState["tab"];
        }
        else
        {
            vaiBD();
        }

        if (dlVeiculos.Items.Count > 0)
            lblVazio.Visible = false;
    }

    void Lista_PreRender(object sender, EventArgs e)
    {
        ViewState["tab"] = tab;
    }

    protected void bindDatalist()
    {
        dlVeiculos.DataSource = tab.DefaultView;
        dlVeiculos.DataKeyField = "Id";
        dlVeiculos.DataBind();
    }

    protected void vaiBD()
    {
        adpt = new VeiculosTableAdapter();
        tab = adpt.GetVeiculos();
        tab.AcceptChanges();
        tab.PrimaryKey = new DataColumn[] { tab.Columns["Id"] };
        tab.Columns["Id"].AutoIncrement = true;
        tab.Columns["Id"].AutoIncrementSeed = 1;
        tab.Columns["Id"].AutoIncrementStep = 1;
        bindDatalist();
    }

    public string devolveMarca(int id)
```

```
    {
        string descritivo = "";
        MarcasTableAdapter ta = new MarcasTableAdapter();
        DataSet.MarcasDataTable t;
        DataSet.MarcasRow r;
        t = ta.GetMarcaById(id);
        if (t.Rows.Count > 0)
        {
            r = (DataSet.MarcasRow)t.Rows[0];
            if (r["Nome"] != null) descritivo = r["Nome"].ToString();
        }
        return descritivo;
    }

    public string devolveModelo(int id)
    {
        string descritivo = "";
        ModelosTableAdapter ta = new ModelosTableAdapter();
        DataSet.ModelosDataTable t;
        DataSet.ModelosRow r;
        t = ta.GetModeloById(id);
        if (t.Rows.Count > 0)
        {
            r = (DataSet.ModelosRow)t.Rows[0];
            if (r["Nome"] != null) descritivo = r["Nome"].ToString();
        }
        return descritivo;
    }

    public int devolveAno(DateTime data)
    {
        return data.Year;
    }

    public string devloveData(DateTime data)
    {
        return data.ToLongDateString();
    }
}
```

Código HTML da página de pesquisa:

```
<%@ Page Title="Pesquisar Veículos" Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="Pesquisar.aspx.cs"
MaintainScrollPositionOnPostback="true" Inherits="Pesquisar" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
    <script>
        window.onload = function () {
            document.getElementsByClassName("current").item(0).className = "";
            document.getElementById("menuPesquisar").className = "current";
        }
    </script>
    </asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">
    <div class="content_title_01">Pesquisar Veículos</div>
    <div class="news_title">Insira os termos da sua pesquisa (Deixe em branco se
pretende que seja indiferente)</div>
    <br />
    <asp:Panel ID="Panel1" CssClass="login" runat="server"
DefaultButton="btnPesquisar" Width="100%">
```

```
        <table cellpadding="5px" cellspacing="0">
            <tr>
                <!-- Marca -->
                <td align="right">
                    <asp:Label runat="server" BackColor="Transparent"
ForeColor="#edad0f" Text="Marca:"></asp:Label>
                </td>
                <td>
                    <asp:DropDownList ID="dropMarcas" OnDataBinding="listarMarcas"
AutoPostBack="true" OnSelectedIndexChanged="dropMarcas_SelectedIndexChanged"
runat="server"></asp:DropDownList>
                </td>
            </tr>
            <tr>
                <!-- Modelo -->
                <td align="right">
                    <asp:Label runat="server" BackColor="Transparent"
ForeColor="#edad0f" Text="Modelo:"></asp:Label>
                </td>
                <td>
                    <asp:DropDownList ID="dropModelos" OnDataBinding="listarModelos"
runat="server"></asp:DropDownList>
                </td>
            </tr>
            <tr>
                <!-- Utilizador -->
                <td align="right">
                    <asp:Label runat="server" BackColor="Transparent"
ForeColor="#edad0f" Text="Utilizador:"></asp:Label>
                </td>
                <td>
                    <asp:DropDownList ID="dropUtilizadores"
OnDataBinding="listarUtilizadores" runat="server"></asp:DropDownList>
                </td>
            </tr>
            <tr>
                <!-- Preço -->
                <td align="right">
                    <asp:Label runat="server" BackColor="Transparent"
ForeColor="#edad0f" Text="Preço:"></asp:Label>
                </td>
                <td style="vertical-align: 35%">
                    De <asp:TextBox ID="txtPrecoMin" Text="" runat="server"
TextMode="Number"></asp:TextBox>  até
                    <asp:TextBox ID="txtPrecoMax" Text="" runat="server"
TextMode="Number"></asp:TextBox> €
                </td>
            </tr>
            <tr>
                <!-- Ano -->
                <td align="right">
                    <asp:Label runat="server" BackColor="Transparent"
ForeColor="#edad0f" Text="Ano:"></asp:Label>
                </td>
                <td style="vertical-align: 35%">
                    De <asp:DropDownList ID="dropAnosMin" OnDataBinding="listarAnos"
runat="server"></asp:DropDownList>  até
                    <asp:DropDownList ID="dropAnosMax" OnDataBinding="listarAnos"
runat="server"></asp:DropDownList>
                </td>
            </tr>
            <tr>
```

```
                <!-- Matrícula -->
                <td align="right">
                    <asp:Label runat="server" BackColor="Transparent"
ForeColor="#edad0f" Text="Matrícula:"></asp:Label>
                </td>
                <td>
                    <asp:TextBox ID="txtMatricula" Text="" runat="server"
CausesValidation="True" MaxLength="8"></asp:TextBox>
                </td>
            </tr>
            <tr>
                <!-- Combustível -->
                <td align="right">
                    <asp:Label runat="server" BackColor="Transparent"
ForeColor="#edad0f" Text="Combustível:"></asp:Label>
                </td>
                <td>
                    <asp:DropDownList ID="dropCombustiveis"
OnDataBinding="listarCombustiveis" runat="server"></asp:DropDownList>
                </td>
            </tr>
            <tr>
                <!-- Quilómetros -->
                <td align="right">
                    <asp:Label runat="server" BackColor="Transparent"
ForeColor="#edad0f" Text="Quilómetros:"></asp:Label>
                </td>
                <td style="vertical-align: 35%">
                    De <asp:TextBox ID="txtQuilometrosMin" Text="" runat="server"
TextMode="Number"></asp:TextBox>  até
                    <asp:TextBox ID="txtQuilometrosMax" Text="" runat="server"
TextMode="Number"></asp:TextBox> Km
                </td>
            </tr>
            <tr>
                <!-- Cor -->
                <td align="right">
                    <asp:Label runat="server" BackColor="Transparent"
ForeColor="#edad0f" Text="Cor:"></asp:Label>
                </td>
                <td>
                    <asp:DropDownList ID="dropCores" OnDataBinding="listarCores"
runat="server"></asp:DropDownList>
                </td>
            </tr>
            <tr>
                <!-- Potência -->
                <td align="right">
                    <asp:Label runat="server" BackColor="Transparent"
ForeColor="#edad0f" Text="Potência:"></asp:Label>
                </td>
                <td style="vertical-align: 35%">
                    De <asp:TextBox ID="txtPotenciaMin" Text="" runat="server"
TextMode="Number"></asp:TextBox>  até
                    <asp:TextBox ID="txtPotenciaMax" Text="" runat="server"
TextMode="Number"></asp:TextBox> CV
                </td>
            </tr>
            <tr>
                <!-- Lotação -->
                <td align="right">
```

```
                    <asp:Label runat="server" BackColor="Transparent"
ForeColor="#edad0f" Text="Lotação:"></asp:Label>
                </td>
                <td>
                    <asp:TextBox ID="txtLotacao" Text="" runat="server"
TextMode="Number"></asp:TextBox>
                </td>
            </tr>
            <tr>
                <!-- Portas -->
                <td align="right">
                    <asp:Label runat="server" BackColor="Transparent"
ForeColor="#edad0f" Text="Portas:"></asp:Label>
                </td>
                <td>
                    <asp:TextBox ID="txtPortas" Text="" runat="server"
TextMode="Number"></asp:TextBox>
                </td>
            </tr>
            <tr>
                <!-- Botão Pesquisar -->
                <td></td>
                <td>
                    <asp:Button ID="btnPesquisar" runat="server" Text="Pesquisar"
OnClick="btnPesquisar_Click" />

                </td>
            </tr>
            <tr>
                <!-- Mensagem de Erro -->
                <td></td>
                <td>
                    <asp:Label ID="lblError" runat="server" ForeColor="Red"
Width="319px"></asp:Label>
                </td>
            </tr>
        </table>
    </asp:Panel>
</asp:Content>
```

Código C# da página de pesquisa:

```
public partial class Pesquisar : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!Page.IsPostBack)
        {
            dropMarcas.DataBind();
            dropModelos.DataBind();
            dropUtilizadores.DataBind();
            dropAnosMin.DataBind();
            dropAnosMax.DataBind();
            dropCombustiveis.DataBind();
            dropCores.DataBind();
        }
    }

    protected void listarMarcas(object sender, EventArgs e)
    {
        MarcasTableAdapter ta = new MarcasTableAdapter();
```

```
    DataSet.MarcasDataTable t;
    t = ta.GetMarcas();
    dropMarcas.Items.Clear();
    dropMarcas.Items.Add(new ListItem("Indiferente", "0"));
    foreach (DataSet.MarcasRow r in t.Rows)
    {
        dropMarcas.Items.Add(new ListItem(r.Nome, r.Id.ToString()));
    }
}

protected void dropMarcas_SelectedIndexChanged(object sender, EventArgs e)
{
    if (dropMarcas.SelectedIndex > -1)
    {
        dropModelos.DataBind();
    }
}

protected void listarModelos(object sender, EventArgs e)
{
    if (dropMarcas.SelectedIndex != -1)
    {
        ModelosTableAdapter ta = new ModelosTableAdapter();
        DataSet.ModelosDataTable t;
        t = ta.GetModelosByMarca(int.Parse(dropMarcas.SelectedValue));
        dropModelos.Items.Clear();
        dropModelos.Items.Add(new ListItem("Indiferente", "0"));
        foreach (DataSet.ModelosRow r in t.Rows)
        {
            dropModelos.Items.Add(new ListItem(r.Nome, r.Id.ToString()));
        }
    }
}

protected void listarUtilizadores(object sender, EventArgs e)
{
    UtilizadoresTableAdapter ta = new UtilizadoresTableAdapter();
    DataSet.UtilizadoresDataTable t;
    t = ta.GetUtilizadores();
    dropUtilizadores.Items.Clear();
    dropUtilizadores.Items.Add(new ListItem("Indiferente"));
    foreach (DataSet.UtilizadoresRow r in t.Rows)
    {
        dropUtilizadores.Items.Add(new ListItem(r.Nome));
    }
}

protected void listarAnos(object sender, EventArgs e)
{
    DropDownList drop = (DropDownList)sender;
    drop.Items.Clear();
    drop.Items.Add(new ListItem("Indiferente"));
    for (int ano = 2016; ano >= 1950; ano--)
    {
        drop.Items.Add(new ListItem(ano.ToString()));
    }
}

protected void listarCombustiveis(object sender, EventArgs e)
{
    dropCombustiveis.Items.Clear();
    dropCombustiveis.Items.Add(new ListItem("Indiferente"));
```

```
        dropCombustiveis.Items.Add(new ListItem("Gasolina"));
        dropCombustiveis.Items.Add(new ListItem("Diesel"));
        dropCombustiveis.Items.Add(new ListItem("GPL"));
        dropCombustiveis.Items.Add(new ListItem("Híbrido"));
        dropCombustiveis.Items.Add(new ListItem("Eléctrico"));
    }

    protected void listarCores(object sender, EventArgs e)
    {
        dropCores.Items.Clear();
        dropCores.Items.Add(new ListItem("Indiferente"));
        dropCores.Items.Add(new ListItem("Amarelo"));
        dropCores.Items.Add(new ListItem("Azul"));
        dropCores.Items.Add(new ListItem("Bege"));
        dropCores.Items.Add(new ListItem("Branco"));
        dropCores.Items.Add(new ListItem("Cinzento"));
        dropCores.Items.Add(new ListItem("Dourado"));
        dropCores.Items.Add(new ListItem("Laranja"));
        dropCores.Items.Add(new ListItem("Prata"));
        dropCores.Items.Add(new ListItem("Preto"));
        dropCores.Items.Add(new ListItem("Roxo"));
        dropCores.Items.Add(new ListItem("Verde"));
    }

    public int devolveMarcaID(string marca)
    {
        int id = 0;
        MarcasTableAdapter ta = new MarcasTableAdapter();
        DataSet.MarcasDataTable t;
        DataSet.MarcasRow r;
        t = ta.GetMarcaByNome(marca);
        if (t.Rows.Count > 0)
        {
            r = (DataSet.MarcasRow)t.Rows[0];
            if (r["Id"] != null) id = int.Parse(r["Id"].ToString());
        }
        return id;
    }

    public int devolveModeloID(string modelo)
    {
        int id = 0;
        ModelosTableAdapter ta = new ModelosTableAdapter();
        DataSet.ModelosDataTable t;
        DataSet.ModelosRow r;
        t = ta.GetModeloByNome(modelo);
        if (t.Rows.Count > 0)
        {
            r = (DataSet.ModelosRow)t.Rows[0];
            if (r["Id"] != null) id = int.Parse(r["Id"].ToString());
        }
        return id;
    }

    protected void btnPesquisar_Click(object sender, EventArgs e)
    {
        VeiculosTableAdapter ta = new VeiculosTableAdapter();
        DataSet.VeiculosDataTable t;
        int marca, modelo, precomin, precomax, quilometrosmin, quilometrosmax,
potenciamin, potenciamax;
        string utilizador, matricula, combustivel, cor, anomin, anomax, lotacao,
portas;
```

```
        //Verificar marca
        if (dropMarcas.SelectedIndex == 0) //Indiferente
            marca = 0;
        else
            marca = devolveMarcaID(dropMarcas.SelectedItem.Text);

        //Verificar modelo
        if (dropModelos.SelectedIndex == 0) //Indiferente
            modelo = 0;
        else
            modelo = devolveModeloID(dropModelos.SelectedItem.Text);

        //Verificar utilizador
        if (dropUtilizadores.SelectedIndex == 0) //Indiferente
            utilizador = "%";
        else
            utilizador = dropUtilizadores.SelectedItem.Text;

        //Verificar preço minimo
        if (txtPrecoMin.Text == "") //Indiferente
            precomin = 0;
        else
            precomin = int.Parse(txtPrecoMin.Text);

        //Verificar preço maximo
        if (txtPrecoMax.Text == "") //Indiferente
            precomax = 2147483647;
        else
            precomax = int.Parse(txtPrecoMax.Text);

        //Verificar ano mínimo
        if (dropAnosMin.SelectedIndex == 0) //Indiferente
        {
            DateTime data = new DateTime(1950, 1, 1);
            anomin = data.ToString();
        }
        else
        {
            DateTime data = new DateTime(int.Parse(dropAnosMin.SelectedItem.Text), 1,
1);
            anomin = data.ToString();
        }

        //Verificar ano máximo
        if (dropAnosMin.SelectedIndex == 0) //Indiferente
        {
            DateTime data = new DateTime(2016, 1, 1);
            anomax = data.ToString(); ;
        }
        else
        {
            DateTime data = new DateTime(int.Parse(dropAnosMax.SelectedItem.Text), 1,
1);
            anomax = data.ToString();
        }

        //Verificar matrícula
        if (txtMatricula.Text == "") //Indiferente
            matricula = "%";
        else
            matricula = txtMatricula.Text;
```

```
        //Verificar combustível
        if (dropCombustiveis.SelectedIndex == 0) //Indiferente
            combustivel = "%";
        else
            combustivel = dropCombustiveis.SelectedItem.Text;

        //Verificar quilómetros minimos
        if (txtQuilometrosMin.Text == "") //Indiferente
            quilometrosmin = 0;
        else
            quilometrosmin = int.Parse(txtQuilometrosMin.Text);

        //Verificar quilómetros maximos
        if (txtQuilometrosMax.Text == "") //Indiferente
            quilometrosmax = 2147483647;
        else
            quilometrosmax = int.Parse(txtQuilometrosMax.Text);

        //Verificar cor
        if (dropCores.SelectedIndex == 0) //Indiferente
            cor = "%";
        else
            cor = dropCores.SelectedItem.Text;

        //Verificar potência minima
        if (txtPotenciaMin.Text == "") //Indiferente
            potenciamin = 0;
        else
            potenciamin = int.Parse(txtPotenciaMin.Text);

        //Verificar potência maxima
        if (txtPotenciaMax.Text == "") //Indiferente
            potenciamax = 2147483647;
        else
            potenciamax = int.Parse(txtPotenciaMax.Text);

        //Verificar lotação
        if (txtLotacao.Text == "") //Indiferente
            lotacao = "%";
        else
            lotacao = txtLotacao.Text;

        //Verificar portas
        if (txtPortas.Text == "") //Indiferente
            portas = "%";
        else
            portas = txtPortas.Text;

        //Verificar se algum campo foi colocado incorretamente
        bool erro = false;
        if (precomin > precomax)
        {
            erro = true;
            lblError.Text = "O preço mínimo deve ser inferior ao preço máximo";
        }
        if (int.Parse(anomin.Substring(anomin.LastIndexOf('/') + 1, 4)) >
int.Parse(anomax.Substring(anomax.LastIndexOf('/') + 1, 4)))
        {
            erro = true;
            lblError.Text = "O ano mínimo deve ser inferior ao ano máximo";
        }
```

```
        if (quilometrosmin > quilometrosmax)
        {
            erro = true;
            lblError.Text = "Os quilómetros mínimos devem ser inferiores aos
quilómetros máximos";
        }
        if (potenciamin > potenciamax)
        {
            erro = true;
            lblError.Text = "A potência mínima deve ser inferior à potência máxima";
        }

        if (!erro)
        {
            /*NOTA: Marca e Modelo não dá para ser tratado antes por isso tem que ser
especificado uma pesquisa para cada possibilidade entre estes dois campos */
            if (marca != 0 && modelo != 0) //Pesquisa com Marca e Modelo especificados
                t = ta.PesquisarVeiculosMarMod(marca, modelo, utilizador, precomin,
precomax, anomin, anomax, matricula, combustivel, quilometrosmin, quilometrosmax, cor,
potenciamin, potenciamax, lotacao, portas);
            else if (marca != 0) //Pesquisa só com Marca especificada
                t = ta.PesquisarVeiculosMar(marca, utilizador, precomin, precomax,
anomin, anomax, matricula, combustivel, quilometrosmin, quilometrosmax, cor,
potenciamin, potenciamax, lotacao, portas);
            else if (modelo != 0) //Pesquisa só com Modelo especificado
                t = ta.PesquisarVeiculosMod(modelo, utilizador, precomin, precomax,
anomin, anomax, matricula, combustivel, quilometrosmin, quilometrosmax, cor,
potenciamin, potenciamax, lotacao, portas);
            else //Pesquisa sem Modelo e Marca especificada
                t = ta.PesquisarVeiculos(utilizador, precomin, precomax, anomin,
anomax, matricula, combustivel, quilometrosmin, quilometrosmax, cor, potenciamin,
potenciamax, lotacao, portas);

            t.AcceptChanges();
            Session["Pesquisa"] = t;
            Response.Redirect("Resultado.aspx");
        }
    }
}
```

Código HTML da página que demonstra os resultados da pesquisa:

```
<%@ Page Title="Resultado da Pesquisa" Language="C#"
MasterPageFile="~/MasterPage.master" AutoEventWireup="true"
CodeFile="Resultado.aspx.cs" Inherits="Resultado" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="Server">
    <script>
        window.onload = function () {
            document.getElementsByClassName("current").item(0).className = "";
        }
    </script>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="Server">
    <div class="content_title_01">Resultados da Pesquisa</div>
    <asp:Label ID="lblRegistos" runat="server" Text=""
ForeColor="#edad0f"></asp:Label>
    <br /><br />
    <asp:DataList ID="dlVeiculos" Width="650px" ShowFooter="false" runat="server"
RepeatColumns="1">
        <ItemTemplate>
```

```
            <div id="itemtemplate">
                <hr color="#edad0f" />
                <br />
                <div class="tituloveiculo"><%#
devolveMarca((int)DataBinder.Eval(Container.DataItem, "Marca")) %> <%#
devolveModelo((int)DataBinder.Eval(Container.DataItem, "Modelo")) %></div>
                <asp:Image ID="Image1"
                    Width="350px"
                    Height="175px"
                    CssClass="img"
                    ImageUrl='<%# DataBinder.Eval(Container.DataItem, "Id",
"FotoVeiculos.ashx?id={0}") %>'
                    runat="server" />
                <font color="#edad0f">Vendedor:</font> <%#
DataBinder.Eval(Container.DataItem, "Utilizador") %><br />
                <font color="#edad0f">Preço:</font> <%#
DataBinder.Eval(Container.DataItem, "Preco") %> €<br />
                <font color="#edad0f">Ano:</font> <%#
devolveAno((DateTime)DataBinder.Eval(Container.DataItem, "Ano")) %><br />
                <font color="#edad0f">Matrícula:</font> <%#
DataBinder.Eval(Container.DataItem, "Matricula") %><br />
                <font color="#edad0f">Combustível:</font> <%#
DataBinder.Eval(Container.DataItem, "Combustivel") %><br />
                <font color="#edad0f">Quilómetros:</font> <%#
DataBinder.Eval(Container.DataItem, "Quilometros") %><br />
                <font color="#edad0f">Cor:</font> <%#
DataBinder.Eval(Container.DataItem, "Cor") %><br />
                <font color="#edad0f">Potência:</font> <%#
DataBinder.Eval(Container.DataItem, "Potencia") %> CV<br />
                <font color="#edad0f">Lotação:</font> <%#
DataBinder.Eval(Container.DataItem, "Lotacao") %><br />
                <font color="#edad0f">Portas:</font> <%#
DataBinder.Eval(Container.DataItem, "Portas") %><br />
                <font color="#edad0f">Adicionado a:</font> <%#
devloveData((DateTime)DataBinder.Eval(Container.DataItem, "Data")) %><br />
            </div>
        </ItemTemplate>
    </asp:DataList>
</asp:Content>
```

Código C# da página que demonstra os resultados da pesquisa:

```
public partial class Resultado : System.Web.UI.Page
{
    VeiculosTableAdapter adpt;
    DataSet.VeiculosDataTable tab;

    protected void Page_Load(object sender, EventArgs e)
    {
        this.PreRender += Resultado_PreRender;

        if (Page.IsPostBack)
        {
            tab = (DataSet.VeiculosDataTable)ViewState["tab"];
        }
        else
        {
            vaiBD();
        }

        if (dlVeiculos.Items.Count == 0)
```

```
            lblRegistos.Text = "Não foram encontrados registos. Por favor, reveja a
sua pesquisa.";
        else
            lblRegistos.Text = "Foram encontrados " + dlVeiculos.Items.Count + "
veículos correspondendo á sua pesquisa.";
    }

    void Resultado_PreRender(object sender, EventArgs e)
    {
        ViewState["tab"] = tab;
    }

    protected void bindDatalist()
    {
        dlVeiculos.DataSource = tab.DefaultView;
        dlVeiculos.DataKeyField = "Id";
        dlVeiculos.DataBind();
    }

    protected void vaiBD()
    {
        adpt = new VeiculosTableAdapter();
        tab = (DataSet.VeiculosDataTable)Session["Pesquisa"];
        tab.AcceptChanges();
        tab.PrimaryKey = new DataColumn[] { tab.Columns["Id"] };
        tab.Columns["Id"].AutoIncrement = true;
        tab.Columns["Id"].AutoIncrementSeed = 1;
        tab.Columns["Id"].AutoIncrementStep = 1;
        bindDatalist();
    }

    public string devolveMarca(int id)
    {
        string descritivo = "";
        MarcasTableAdapter ta = new MarcasTableAdapter();
        DataSet.MarcasDataTable t;
        DataSet.MarcasRow r;
        t = ta.GetMarcaById(id);
        if (t.Rows.Count > 0)
        {
            r = (DataSet.MarcasRow)t.Rows[0];
            if (r["Nome"] != null) descritivo = r["Nome"].ToString();
        }
        return descritivo;
    }

    public string devolveModelo(int id)
    {
        string descritivo = "";
        ModelosTableAdapter ta = new ModelosTableAdapter();
        DataSet.ModelosDataTable t;
        DataSet.ModelosRow r;
        t = ta.GetModeloById(id);
        if (t.Rows.Count > 0)
        {
            r = (DataSet.ModelosRow)t.Rows[0];
            if (r["Nome"] != null) descritivo = r["Nome"].ToString();
        }
        return descritivo;
    }

    public int devolveAno(DateTime data)
```

```
    {
        return data.Year;
    }

    public string devloveData(DateTime data)
    {
        return data.ToLongDateString();
    }
}
```

Código C# do *web handler* da aplicação:

```
<%@ WebHandler Language="C#" Class="FotoVeiculos" %>

using System;
using System.Web;
using System.Data;
using DataSetTableAdapters;
using System.IO;

public class FotoVeiculos : IHttpHandler {
    DataSet.VeiculosDataTable tab;
    DataSet.VeiculosRow r;
    VeiculosTableAdapter adpt;

    public void ProcessRequest (HttpContext context) {
        int id;
        if (context.Request.QueryString["Id"] != null)
        {
            if (int.TryParse(context.Request.QueryString["Id"].ToString(), out id))
            {
                adpt = new VeiculosTableAdapter();
                tab = adpt.GetVeiculosById(id);
                if (tab.Rows.Count > 0)
                {
                    r = (DataSet.VeiculosRow)tab.Rows[0];
                    string caminho;
                    if (!r.IsFotoNull() && !r.IsFotoTipoNull())
                    {
                        context.Response.ContentType = r.FotoTipo.ToString();
                        caminho = context.Server.MapPath("~/Fotos/" + r.Foto);
                        if (!File.Exists(caminho))
                        {
                            context.Response.ContentType = "Image/jpeg";
                            caminho = context.Server.MapPath("~/Fotos/" + "Sem
Foto.jpg");
                        }
                    }
                    else
                    {
                        context.Response.ContentType = "Image/jpeg";
                        caminho = context.Server.MapPath("~/Fotos/" + "Sem Foto.jpg");
                    }

                    byte[] fotoArray;
                    FileStream fs = new FileStream(caminho, FileMode.Open,
FileAccess.Read);
                    fotoArray = new byte[fs.Length];
                    fs.Read(fotoArray, 0, (int)fs.Length);
                    fs.Close();
                    context.Response.BinaryWrite(fotoArray);
```

```
                }
            }
        }
    }

    public bool IsReusable {
        get {
            return false;
        }
    }

}
```