

## Gestão de Candidaturas de Emprego

Pedro Emanuel Martins Tavares

Projecto PSTFC 2005-2006

Trabalho realizado sob a supervisão do  
Engenheiro Isidro Vila Verde,  
do Departamento de Engenharia Electrotécnica e de Computadores,  
da Faculdade de Engenharia da Universidade do Porto



  
Ciência.Inovação Programa Operacional Ciência e Inovação 2010  
2010

Porto, Julho de 2006

621.3(047.3)/  
LEEC  
2006/TAVp

621 3107.31 LEEK 2006 AVZ

UNIVERSITY OF CALIFORNIA  
FACULTY OF AGRICULTURE  
AND MECHANICAL ARTS  
NO. 105171  
CDU  
DATE 24 02 10

### **Agradecimentos:**

Gostava de agradecer ao Engenheiro Isidro Vila Verde por toda a disponibilidade que sempre apresentou relativamente à concepção deste projecto bem como ao Engenheiro Marco Sá, da empresa Holmes Place.

### **Resumo:**

Este trabalho pretende facilitar a introdução, consulta e edição de dados relativamente à gestão de candidaturas de emprego nos recursos humanos do Holmes Place. De um modo rápido é possível saber se o candidato já está admitido bem como consultar todos os dados que lhe dizem respeito, e que de algum modo são importantes para os recursos humanos. O acesso aos dados é feito tendo em conta as permissões de utilizador e de acordo com estas é possível editá-los ou adicionar informação.

Relativamente à tecnologia utilizada, o projecto foi desenvolvido com o Visual Studio 2005, Professional Edition, SQL Server 2005 e com recurso às linguagens ASP.NET, VB.NET e SQL.

Todos estes aspectos serão desenvolvidos no texto que se segue.

## **Abstract**

This project intends to facilitate the introduction, consulting and edition of the data related with the management of the job's appliance in the human's resources of Holmes Place. In a quick way is possible to know if the candidate is admitted as well to consult all the data related with him and in some manner are important to human resources. The data access is done attending to the user's permissions and according with them is possible to edit or add information.

The project was developed with Visual Studio 2005, Professional Edition, SQL Server 2005 and using ASP.NET, VB.NET and SQL.

All these subjects are explained in the following text.

## Índice:

<i>Lista de figura e lista de tabelas</i> .....	5
<i>1 Introdução</i> .....	7
<i>1.1 Microsoft Visio 2003</i> .....	7
<i>1.2 Microsoft SourceSafe</i> .....	7
<i>1.3 Microsoft Visual Studio 2005, Professional Edition</i> .....	7
<i>1.4 ASP.NET</i> .....	10
<i>1.5 ADO.NET</i> .....	14
<i>1.5.1 .NET FRAMEWORK Data Provider</i> .....	15
<i>1.5.2 Objectos existentes na .NET FRAMEWORK Data Provider</i> .....	15
<i>1.5.2.1 Connection</i> .....	16
<i>1.5.2.2 Command</i> .....	16
<i>1.5.2.3 DataReader</i> .....	17
<i>1.5.2.4 DataAdapter</i> .....	17
<i>1.5.2.5 DataSet</i> .....	17
<i>1.6 Bases de Dados Relacionais e SQL</i> .....	20
<i>2 Planear a aplicação</i> .....	22
<i>2.1 Estratégia e Visão</i> .....	23
<i>2.2 Construção</i> .....	24
<i>2.2.1 Construção da Base de Dados</i> .....	25
<i>2.3 Definir a interface do site</i> .....	31
<i>2.3.1 Navegação</i> .....	33
<i>2.3.2 Funcionamento da aplicação</i> .....	34
<i>2.3.3 Construir a interface</i> .....	37
<i>3 A ligação à Base de Dados</i> .....	44
<i>4 O controlo GridView</i> .....	48
<i>5 A interface de consulta de dados do candidato</i> .....	50
<i>6 A interface de edição de dados</i> .....	60
<i>7 A interface da área Academia</i> .....	65
<i>8 As permissões de utilizador</i> .....	67
<i>9 Testar</i> .....	70
<i>10 Implementar</i> .....	71
<i>10.1 Página Inicial</i> .....	71
<i>10.2 Inserção de um candidato</i> .....	72
<i>10.3 A área de Consulta</i> .....	79
<i>10.4 A edição de dados</i> .....	81
<i>10.5 A área Academia</i> .....	83
<i>11 Conclusão</i> .....	86
<i>12 Bibliografia</i> .....	87

## LISTA DE FIGURAS

<i>Figura 1: A IDE do Visual Studio 2005.....</i>	<i>8</i>
<i>Figura 2: O processo “managed execution”.....</i>	<i>12</i>
<i>Figura 3: Arquitectura ADO.NET.....</i>	<i>15</i>
<i>Figura 4: O modelo do objecto ADO.NET SqlConnection DataAdapter.....</i>	<i>17</i>
<i>Figura 5: Modelo do objecto DataSet.....</i>	<i>19</i>
<i>Figura 6: Etapas do projecto.....</i>	<i>22</i>
<i>Figura 7: Estratégia de construção do projecto.....</i>	<i>23</i>
<i>Figura 8: A construção de um ASP.NET Web Site.....</i>	<i>24</i>
<i>Figura 9: O Server Explorer.....</i>	<i>25</i>
<i>Figura 10: Diagrama da Base de Dados.....</i>	<i>26</i>
<i>Figura 11: Detalhe de duas tabelas e sua relação .....</i>	<i>27</i>
<i>Figura 12 Relação entre tabelas .....</i>	<i>28</i>
<i>Figura 13: A caixa properties da chave primária .....</i>	<i>30</i>
<i>Figura 14: A MasterPage.....</i>	<i>31</i>
<i>Figura 15: O menu Add New Item.....</i>	<i>32</i>
<i>Figura 16: Fluxograma da aplicação.....</i>	<i>35</i>
<i>Figura 17: A Home Page.....</i>	<i>37</i>
<i>Figura 18: O menu Properties de um botão.....</i>	<i>40</i>
<i>Figura 19: A Interface de Inserção do Candidato (1).....</i>	<i>41</i>
<i>Figura 20: A Interface de Inserção do Candidato (2).....</i>	<i>42</i>
<i>Figura 21: Detalhe do menu Properties na opção eventos.....</i>	<i>43</i>
<i>Figura 22: A criação de uma ligação à Base de Dados.....</i>	<i>44</i>
<i>Figura 23: A criação da ConnectionString.....</i>	<i>44</i>
<i>Figura 24: A configuração da fonte de dados.....</i>	<i>46</i>
<i>Figura 25: Associar controlos à base de dados.....</i>	<i>46</i>
<i>Figura 26: A GridView da tabela Dados_Pessoais.....</i>	<i>50</i>
<i>Figura 27: O Smart Tag do GridView.....</i>	<i>51</i>
<i>Figura 28: Consulta e apresentação de dados.....</i>	<i>54</i>
<i>Figura 29: Associação de controlos com o GridView.....</i>	<i>56</i>
<i>Figura 30: O HyperlinkField Properties.....</i>	<i>58</i>
<i>Figura 31: A página Detalhes .....</i>	<i>59</i>
<i>Figura 32: O Smart Tag do DetailsView.....</i>	<i>60</i>
<i>Figura 33: A edição de dados, partindo do modo de consulta.....</i>	<i>62</i>
<i>Figura 34: A associação entre um controlo e o DetailsView.....</i>	<i>64</i>
<i>Figura 35: A área Academia, consultar.....</i>	<i>65</i>
<i>Figura 36: A área Academia, consultar todos.....</i>	<i>66</i>
<i>Figura 37: Inserir dados na área Academia.....</i>	<i>66</i>
<i>Figura 38: O ASP.NET Web Site Administration Tool.....</i>	<i>67</i>
<i>Figura 39: Figura 39: A caixa de diálogo Security.....</i>	<i>68</i>
<i>Figura 40: A criação dos users.....</i>	<i>68</i>
<i>Figura 41: Configuração das áreas de acesso.....</i>	<i>69</i>
<i>Figura 42: O controlo Log In.....</i>	<i>69</i>
<i>Figura 43: A Home Page.....</i>	<i>71</i>
<i>Figura 44: O Formulário de Introdução do Candidato, Dados Pessoais e Habilitações do Candidato (1) .....</i>	<i>72</i>
<i>Figura 45: O Formulário de Introdução do Candidato, Dados Pessoais e Habilitações do Candidato (2) .....</i>	<i>73</i>

<i>Figura 46: Formulário de Introdução do Candidato, Dados Pessoais e Habilitações do Candidato (3).....</i>	<i>74</i>
<i>Figura 47: O formulário Pré-Entrevista Telefónica.....</i>	<i>74</i>
<i>Figura 48: O formulário Entrevista Pessoal (1).....</i>	<i>75</i>
<i>Figura 49: O formulário Entrevista Pessoal (2).....</i>	<i>76</i>
<i>Figura 50: O formulário Entrevista Interactiva.....</i>	<i>77</i>
<i>Figura 51: O formulário Negociação.....</i>	<i>78</i>
<i>Figura 52: Formulário enviado com sucesso.....</i>	<i>78</i>
<i>Figura 53: A área Consulta.....</i>	<i>79</i>
<i>Figura 54: Os resultados de uma pesquisa.....</i>	<i>80</i>
<i>Figura 55: A área Detalhes a partir de uma pesquisa.....</i>	<i>81</i>
<i>Figura 56: O modo de Edição .....</i>	<i>82</i>
<i>Figura 57: A área Academia.....</i>	<i>83</i>
<i>Figura 58: Inserir dados na área Academia .....</i>	<i>84</i>
<i>Figura 59: A opção Ver Todos na Área Academia.....</i>	<i>85</i>
<i>Figura 60: O cronograma de trabalhos.....</i>	<i>86</i>

#### **LISTA DE TABELAS**

<i>Tabela 1: Pormenor das tabelas .....</i>	<i>27</i>
<i>Tabela 2: Os Web Controls .....</i>	<i>37</i>
<i>Tabela 3: As propriedades do GridView.....</i>	<i>48</i>

## **1 - Introdução:**

Para perceber como foi realizado o projecto é preciso conhecer a tecnologia utilizada. Como já foi referido este trabalho foi realizado com o programa Visual Studio 2005, Professional Edition, e SQL Server 2005 recorrendo às linguagens VB.NET, ASP.NET e SQL. Para além destes foram utilizados utilitários tais como Microsoft Visio 2003 e Microsoft SourceSafe. Passo a explicar cada um deles.

### **1.1 - Microsoft Visio 2003:**

Um utilitário da Microsoft que permite a criação de modo eficiente de gráficos, flowcharts e todo um conjunto de aspectos gráficos.

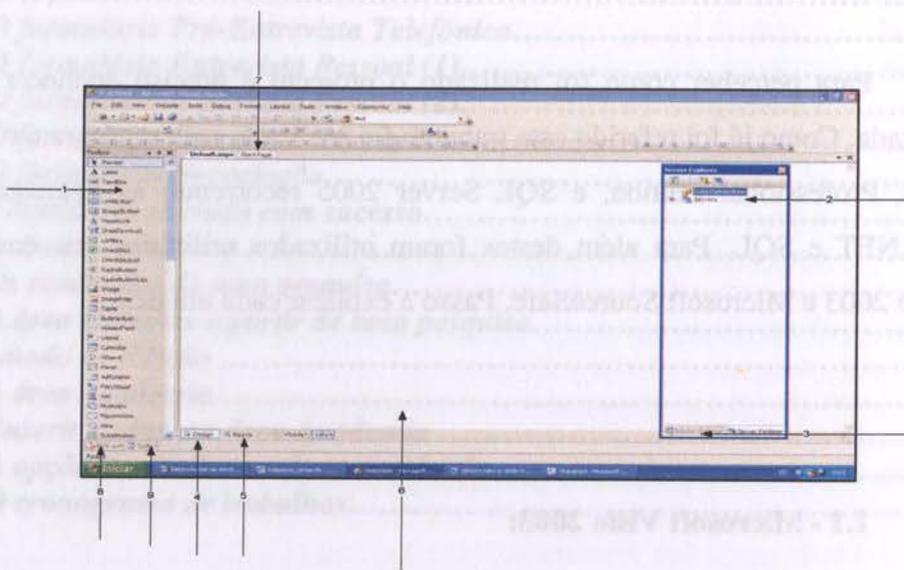
### **1.2 - Microsoft SourceSafe:**

Permite gravar todas as versões de um ficheiro indicando a data em que este foi gravado possibilitando deste modo a recuperação de versões mais antigas.

### **1.3 - Microsoft Visual Studio 2005, Professional Edition**

O Microsoft Visual Studio é um conjunto completo de ferramentas para desenvolver aplicações Web, ASP.NET, XML, desktop e móveis. Visual Basic, Visual C++, Visual C# e Visual J# todas usam o mesmo ambiente integrado de desenvolvimento (a IDE), que permite àquelas a partilha das ferramentas e facilita a criação de soluções com várias linguagens. É importante referir que aquelas linguagens fornecem a funcionalidade do .NET Framework. Este será explicado mais adiante juntamente com o ASP.NET. Assim, é neste ambiente integrado que se implementa tanto o código como a parte visual da aplicação. Esta última tornou-se mais intuitiva com o novo Visual Web Developer, disponível no Visual Studio 2005. Esta funcionalidade continua a basear-se na IDE e inclui um conjunto de opções e utilidades para criar ASP.NET 2.0 Web Sites.

Penso que a maneira mais fácil de entender o que é a IDE é explicá-la de modo visual.



**Figura 1: A IDE do Visual Studio 2005**

Passo a explicar os pontos indicados:

- 1- Toolbox: tem vários controlos associados a diferentes áreas do desenvolvimento de um projecto. O controlo desejado arrasta-se para a área de desenho;
- 2- Server Explorer: associado à criação, manutenção e conexão da base de dados ao projecto;
- 3- Solution Explorer: os diferentes ficheiros do projecto estão aqui. É possível apagar, adicionar ou alterar ficheiros;
- 4- Design: modo de desenho. Neste modo, o projecto é desenvolvido de modo gráfico;
- 5- Source: passa para o modo de código. Assim, é possível complementar o modo de desenho com algumas funcionalidades de código;
- 6- Área de desenho: a área onde se colocam os objectos em modo visual.
- 7- Ficheiros: permite seleccionar um outro ficheiro, que está activo mas não se vê;

8- Error List: quando se compila o projecto, qualquer erro vem indicado aqui;

9- Output: quando se faz a compilação vemos os resultados da mesma;

A IDE permite ter aberta mais do que uma janela e a sua disposição depende somente dos critérios do utilizador. Assim, deste modo, permite uma maior flexibilidade de trabalho bem como um rendimento maior quando comparado com versões anteriores. É de salientar que nesta versão do Visual Studio, a IDE apresenta aquilo a que se chama de Dynamic Help, que vai mostrando em tempo real os tópicos relacionados com o que estamos a fazer no momento.

Representa uma melhoria relativamente a versões anteriores e nesse aspecto deve referir-se o seguinte:

- Suporte para ASP.NET 2.0

O Visual Web Developer suporta novas funcionalidades que vêm com o ASP.NET 2.0;

- Projecto mais flexível

O projecto pode ser desenvolvido recorrendo ao IIS ( Internet Information Services) ou então pode ser criado localmente;

- Melhoria do modelo “*code-behind*”

O programador pode criar páginas usando o modelo “code-behind” (código na página .aspx.vb) ou incluir o código na página ASP.NET (o ficheiro .aspx);

- Melhorias nas funcionalidades de programação

Muitas das tarefas básicas podem ser implementadas visualmente. Adicionar controlos, criar ligações entre páginas e ligações à base de dados podem ser criadas no Visual Web Developer. O Microsoft IntelliSense ajuda à escrita do código;

## 1.4 - ASP.NET:

Um arquivo desenvolvido com a tecnologia ASP pode conter texto, *tags* HTML e *scripts*.. Quando uma página ASP é solicitada, o servidor interpreta o arquivo linha a linha processando os *scripts* e retorna ao cliente apenas código HTML.

O ASP.NET é mais que uma actualização do ASP clássico: apresenta um novo modelo de programação e uma grande variedade de ferramentas, fornece um modelo unificado de desenvolvimento para a Web que inclui serviços necessários para a construção de aplicações. É parte da plataforma .NET Framework e permite que se faça uso das características que esta oferece. A plataforma .NET é um componente Windows completo que apoia a construção e a execução de aplicações. É um ambiente que permite o desenvolvimento de aplicações *desktop* (Windows ou consola), aplicações para aparelhos móveis (palm-tops, celulares, etc.) e o desenvolvimento de aplicações *web* através da tecnologia ASP.NET.

O .NET proporciona um ambiente de desenvolvimento com muitos recursos para os programadores. Pode dizer-se que o .NET disponibiliza um ambiente de desenvolvimento, que possui, entre outras características:

- **Multi-plataforma:**

Similar à tecnologia Java: todo o código é compilado, interpretado, depurado e transformado numa linguagem intermediária. Na plataforma .NET essa linguagem é a MSIL (*Microsoft Intermediate Language*). A MSIL é entendida apenas pela CLR (*Common Language Runtime*) da plataforma. Quando o programa é executado pela primeira vez, a CLR transforma o código MSIL em código de máquina para que este possa ser interpretado pelo processador. Sendo assim, toda aplicação construída na plataforma .NET pode ser executada em qualquer sistema operativo, desde que este por sua vez, tenha uma CLR desenvolvida;

- **Multi-dispositivos:**

Através da .NET pode-se desenvolver programas *web*, *desktop* e também para dispositivos móveis, como telefones celulares e palm-tops;

- **Multi-linguagem:**

O que é importante para a plataforma .NET é o código MSIL e a CLR, logo, a mesma não se limita a trabalhar apenas com uma linguagem de programação. Para ser utilizada na plataforma a linguagem de programação deve ser compatível com a CLS (*Common Language Specification*), visto que a CLS é compatível com a plataforma de desenvolvimento .NET e gera códigos MSIL compatíveis com a CLR. Actualmente existem três linguagens homologadas seguindo a CLS, são elas: C#, J# e VB.NET, contudo estão a ser desenvolvidas outras linguagens. Como já referido neste trabalho utilizou-se a VB.NET.

- **Orientada a objectos:**

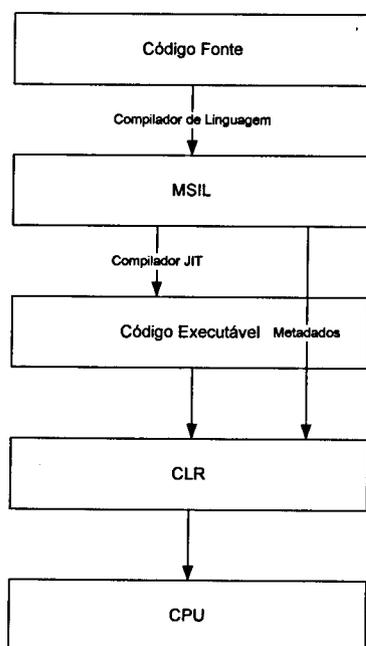
A .NET dá suporte total à programação orientada a objectos, o que significa também que as linguagens homologadas para a plataforma devem prover esse suporte;

- **Interoperabilidade entre linguagens:**

Classes escritas em determinada linguagem, após compilado o código-fonte, quando é gerada a sua correspondente MSIL, podem ser utilizadas por programas desenvolvidos em outras linguagens suportadas pela Plataforma;

- **Uma vasta biblioteca de classes:**

Entre elas estão classes de acesso e manipulação de dados, classes de manipulação de informações transmitidas via Web, classes para trabalhar com objectos visuais e outros.



**Figura 2: O processo “managed execution”**

Ao processo de transformar o código fonte numa aplicação executável num ambiente .NET Framework dá-se o nome de “managed execution”.

A figura 2 é um resumo dos passos indicados anteriormente. Resumindo, primeiro cria-se o código fonte usando um ambiente especializado. No nosso caso, o Visual Studio 2005. Depois, usamos um compilador de linguagem (um compilador .NET) para transformar o código fonte numa outra forma conhecida como MSIL (Microsoft Intermediate Language). Tipicamente estes ficheiros têm uma extensão .dll ou .exe. Parecem executáveis mas não podem correr sem o CLR (Common Language Runtime). O MSIL é independente de qualquer sistema operativo ou arquitectura de computador. Quando se executa um .NET executável, o .NET Framework usa um compilador JIT (Just-in-Time) para traduzir o ficheiro MSIL em instruções que possam ser executadas pelo CPU. A CLR passa o código então compilado pelo compilador JIT, fazendo a monitorização da sua execução bem como serviços de manutenção tais como verificações de segurança e memória. Os metadados indicados na figura, dizem respeito à MSIL que além do código compilado contém informações acerca do conteúdo dos ficheiros. Estes metadados são usados pela CLR para uma boa utilização e operação do código.

Analisando as informações supracitadas, pode-se dizer que a .NET possui um modelo de desenvolvimento muito parecido com o modelo do Java, onde, na

compilação do código-fonte do programa, um novo código, numa outra linguagem, é gerado para que possa ser entendido em ambientes diferentes. Classes desenvolvidas em .NET podem ser acedidas por diversos dispositivos. A classe acedida por todos será a mesma, diferenciando apenas a forma de disponibilização das informações ao usuário. Essas formas são as extensões específicas para cada tipo de plataforma ou ambiente de desenvolvimento. Para os aplicativos acedidos via Web existem APIs denominadas *Web Forms* e *Web Controls*, porém aplicativos Web também podem utilizar as mesmas APIs para *desktop*. Para as aplicações *desktop* são utilizadas as extensões denominadas *Windows Forms*, extensões estas que são APIs utilizadas no desenvolvimento do *layout* das janelas do aplicativo. Os *Web Controls* são componentes visuais que fazem parte de plataforma .NET Framework e auxiliam o programador na criação de páginas Web. Esses componentes, na sua grande maioria, são semelhantes aos elementos HTML.

Para que os *Web Controls* possam ser interpretados pelo *Browser*, antes de o servidor enviar a página ao cliente, a própria plataforma .NET faz uma espécie de tradução do código, convertendo todos os *Web Controls* nos seus respectivos elementos HTML.

Existe muita diferença entre os modelos de compilação de um código ASP e de um código ASP.NET. Na tecnologia ASP o código é sempre interpretado antes do envio ao cliente, ou seja, a cada acesso a página é novamente interpretada. Enquanto que, através do modelo de compilação proposto pelo ASP.NET, as páginas primeiramente são compiladas gerando o código MSIL, depois esse código é interpretado pela CLR. Após essas etapas o cliente recebe a página solicitada. A princípio esse processo pode parecer mais demorado que o do ASP, mas vale ressaltar que uma vez feito esse processo, ele só se repete quando o arquivo original sofre alguma alteração. A modificação veio para possibilitar que aplicações desenvolvidas com a .NET Framework pudessem ser executadas em qualquer sistema operativo que dá um suporte à plataforma.

Além da forma de compilação, também sofreu uma modificação considerável na forma de se escrever o código. Essa modificação fez-se tanto na quantidade de classes disponíveis como na forma de utilização das mesmas. Por exemplo, o código em ASP faz uso do método *Write* do objecto *Response* para criar conteúdo HTML para o cliente; já o ASP.NET não adopta esta metodologia, já que a página e os seus objectos (controles) são interpretados, e podem ser modificados através das suas propriedades.

Esta é uma grande modificação no conceito de programação para a *web* trazido

pela plataforma .NET: os elementos da página tornam-se objectos que podem ser modificados programaticamente. Outro ambiente onde isso acontece de forma mais constante e necessária é na programação para *desktop*, na qual todos os componentes visuais utilizados possuem as suas propriedades expostas através de um modelo de objecto e podem ser modificadas via linguagem de programação.

Esta analogia é visível, também, quando se interpreta a definição dos *Web Forms*: formulários para a *web*; ou seja, são como formulários de programas Desktop (janelas), acessíveis através do browser, na Internet.

Dando continuidade aos estudos da plataforma .NET, a próxima subsecção tratará da forma como a aplicação fará o acesso à base de dados, utilizando uma biblioteca de classes contida no .NET Framework: o ADO.NET.

## 1.5 - ADO.NET

Juntamente com a plataforma .NET, a Microsoft desenvolveu uma biblioteca de classes para acesso à base de dados denominada ActiveX Data Model .NET (ADO.NET).

Pode definir-se o ADO.NET como um conjunto de classes disponíveis no .NET Framework para fazer o acesso a dados. O ADO.NET baseia-se num acesso conectado e desconectado à fonte de dados. O acesso conectado é aquele que mantém uma conexão com a fonte de dados durante toda a execução da aplicação. Enquanto que o acesso desconectado é aquele em que a aplicação se conecta com a fonte de dados apenas para carregar e actualizar dados, diminuindo assim a quantidade de recursos utilizados pelo sistema.

O ADO.NET é formado basicamente por dois componentes fundamentais: o .NET Framework Data Provider e o DataSet. O .NET Framework Data Provider permite a conexão com a fonte de dados e a manipulação desses dados através de comandos SQL e o DataSet faz um acesso desconectado à fonte de dados.

Veja-se a arquitectura ADO.NET:

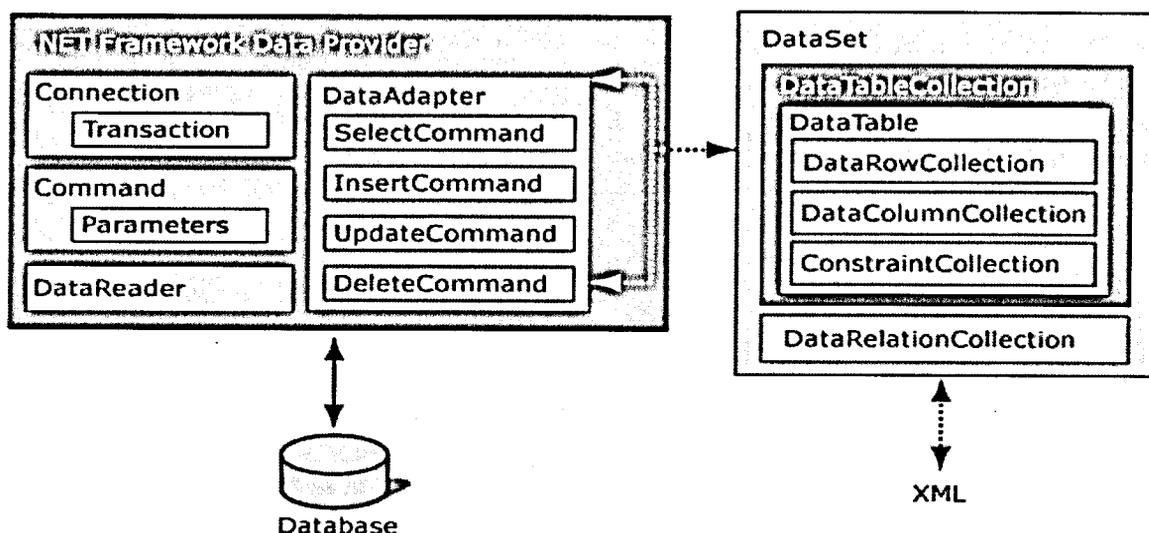


Figura 3: Arquitectura ADO.NET

As classes ADO.Net encontram-se no System.Data.dll e são integradas com classes XML que se encontram no System.Xml.Dll. Quando se compila código que usa o System.Data é referenciado o System.Data.dll e o System.Xml.dll.

### 1.5.1- .NET Framework Data Provider

Formado por um conjunto de provedores de acesso a dados. Até ao presente momentos existem quatro principais provedores de acesso:

- ODBC
- OLE DB
- SQL
- Oracle

Para cada um deles existem *namespaces* específicos que fornecem objectos para aceder e manipular dados. Como esses objectos de acesso e manipulação de dados existem igualmente para os quatro provedores citados acima, para fazer a utilização destes na aplicação basta acrescentar, como prefixo ao nome do objecto, o provedor que se deseja utilizar. Os objectos ADO.NET SQL e Oracle fazem um acesso directo através

de uma API da própria Base de Dados, enquanto que os objectos ADO.NET ODBC e OLE DB passam por uma camada intermediária representada pelos seus respectivos provedores. Sendo assim podemos concluir que utilizando objectos SQL ou Oracle, o acesso será figura mais directo e conseqüentemente mais rápido à fonte de dados, pois não haverá um mediador na conexão.

## **1.5.2 - Objectos existentes no .NET Framework Data Provider**

### **1.5.2.1 - Connection**

Este é o primeiro objecto a ser utilizado. É responsável pela conexão com a Base de Dados. Cada provedor possui um nome distinto; assim, para o provedor SQL, utiliza-se o objecto `SqlConnection`, para o provedor Oracle, utiliza-se o objecto `OracleConnection`, `OdbcConnection` para o provedor ODBC, e `OleDbConnection` para o provedor OLE DB. No presente trabalho usou-se uma ligação a uma Base de Dados SQL. Assim, o comando para o presente trabalho será:

```
SqlConnection("Server=local;DataBase=dados; ")
```

O servidor utilizado é o local, por isso, "Server=local", e o nome da Base de Dados criada é "dados".

### **1.5.2.2 - Command**

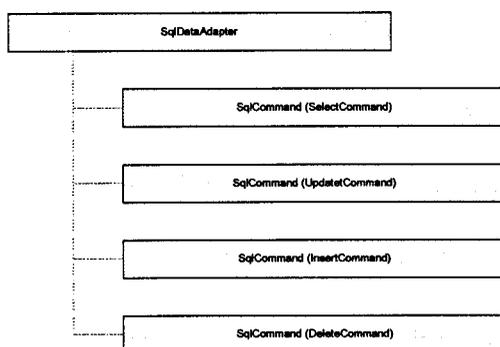
Após estabelecer a conexão com a Base de Dados, pode-se utilizar o objecto `Command` para executar comandos SQL na Base de Dados tais como apresentar e modificar dados e enviar ou receber parâmetros de informação

### 1.5.2.3 - DataReader

O DataReader é um objecto simples de ser utilizado para obter o resultado de uma consulta SQL. É usado somente para leitura de informações, não fornecendo formas de alterar as informações contidas na base de dados.

### 1.5.2.4 DataAdapter

O objecto DataAdapter é o responsável por passar os dados da fonte de dados para o DataSet (que fica desconectado da fonte de dados). É ele que realiza todas as interações com o provedor utilizado na aplicação. Após a realização da conexão com a Base de Dados e da criação de uma “string” contendo a instrução SQL a ser executada na Base de Dados é criado um objecto SqlDataAdapter, passando por parâmetros a instrução SQL a ser executada e a conexão com a Base de Dados.



**Figura 4: O modelo do objecto ADO.NET SqlClient DataAdapter**

### 1.5.2.5 DataSet

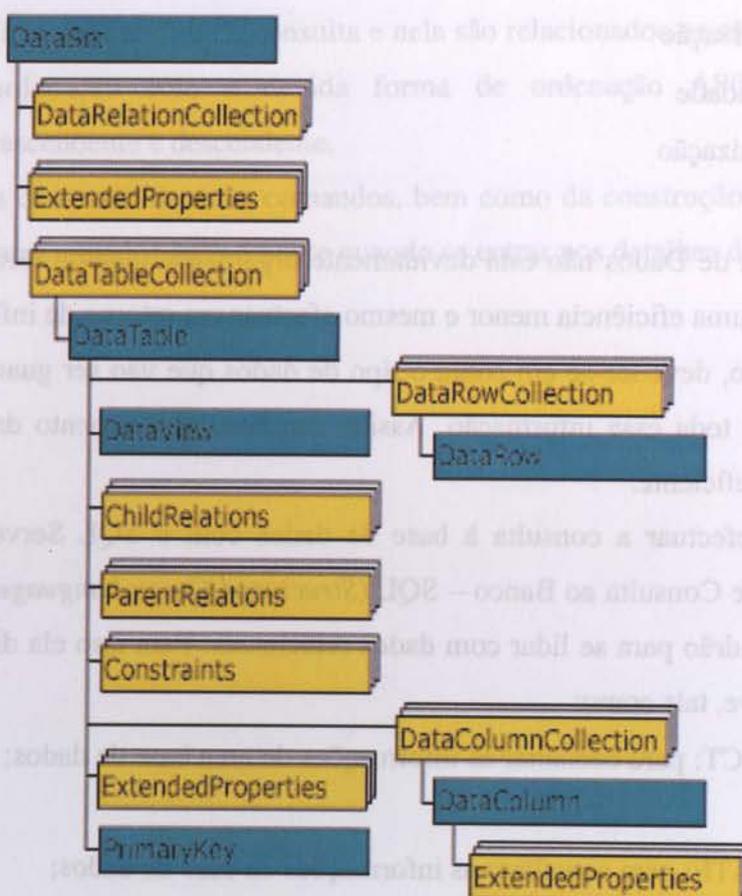
O DataSet é um componente fundamental do ADO.NET, responsável por fornecer, de forma desconectada, meios de aceder a tabelas, colunas, linhas, relações e restrições de uma Base de Dados. Pode ser usado com várias e diferentes fontes de dados, com dados XML, ou para aplicações locais. Para ter acesso às informações da

fonte de dados, conforme visto anteriormente, ele utiliza o objecto `DataAdapter`. A seguir são apresentadas algumas propriedades de um `DataSet`:

- `DataTable`: objecto chave para aceder informações de uma tabela da Base de Dados, como: colunas, linhas e relações. Este objecto é definido no `System.Data namespace` e representa uma única tabela residente em memória.
- `DataRow`: objecto que possui informações sobre as linhas de uma determinada tabela. Este objecto retém o valor original para verificar alterações nos valores guardados.
- `DataColumn`: objecto que possui informações sobre o esquema para cada coluna da tabela, como por exemplo o tipo de dados e o tamanho (bytes);
- `DataRelation`: objecto que contém dados sobre os relacionamentos das tabelas, que pode ser feito entre elementos `DataTable` através dos objectos `DataColumn`. Um `DataSet` contém as relações entre tabelas no seu objecto `DataRelationCollection`. Uma relação representada pelo objecto `DataRelation`, associa linhas de um `DataTable` com as linhas de outro `DataTable`. Uma relação é análogo a juntar um caminho que existe entre uma chave primária (Primary Key → identificador de cada tabela) à sua chave estrangeira (Foreign Key → a chave primária noutra tabela) numa base de dados relacional. O objecto `DataRelation` identifica as colunas que se combinam no `DataSet`. As relações entre tabelas permitem a navegação entre estas. Os elementos essenciais de um objecto `DataRelation` são o nome da relação a ser estabelecida, os nomes das tabelas a ser relacionadas e as respectivas colunas em cada tabela. Quando se adiciona uma relação a um objecto `DataRelationCollection` podemos adicionar um `UniqueKeyConstraint` (chave primária única) bem como um

ForeignKeyConstraint (chave estrangeira única) para reforçar a integridade da Base de Dados quando se efectuam mudanças nos valores das colunas.

- ExtendedProperties: O DataSet, o DataTable, e o DataColumn todos têm uma propriedade chamada ExtendedProperties. ExtendedProperties é um PropertyCollection onde se pode inserir informação tal como o comando SELECT utilizado para gerar os resultados ou o tempo em que foram gerados os dados.



**Figura 5: Modelo do objecto DataSet**

## 1.6 - Bases de Dados Relacionais e SQL

Uma base de dados relacional é formada por uma colecção de tabelas, cada uma com um nome único e identificada por uma chave primária. Uma tabela é um conjunto não ordenado de linhas e cada uma dessas linhas é composta por um número de campos (atributos). Assim, base de dados relacional é formada por um conjunto de tabelas, onde as linhas dessas tabelas são formadas por colunas, que são os atributos da tabela. Esses atributos dizem quais as informações que estão armazenadas na tabela, enquanto as linhas da tabela representam o relacionamento entre os valores dos atributos.

Quando se cria uma base de dados relacional devemos ter em atenção os seguintes pontos:

- Organização
- Integridade
- Optimização

Se a Base de Dados não está devidamente organizada quando se realiza a consulta esta pode ter uma eficiência menor e mesmo efectuar um retorno de informação errónea. Por outro lado, deve ter-se em conta o tipo de dados que vão ser guardados e como se vai organizar toda essa informação. Assim, um bom planeamento da Base de Dados torna-a mais eficiente.

Para efectuar a consulta à base de dados com o SQL Server utiliza-se uma Linguagem de Consulta ao Banco – SQL (*Structured Query Language*), que constitui a linguagem padrão para se lidar com dados relacionais. Para isso ela dispõe de algumas palavras-chave, tais como:

- **SELECT:** para consultar as informações de uma base de dados;
- **UPDATE:** para actualizar as informações da base de dados;
- **DELETE:** para remover as informações;
- **INSERT INTO:** para inserir informações na base de dados;

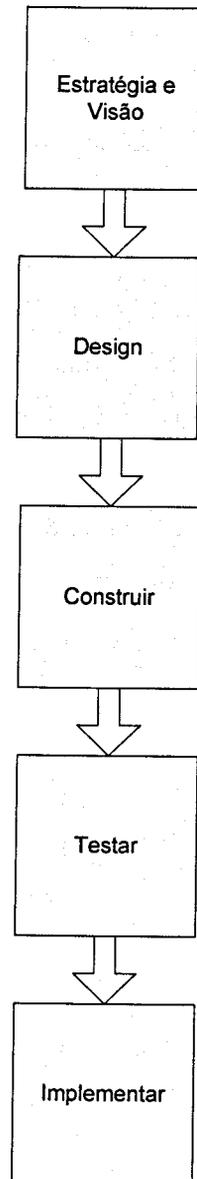
Deve-se salientar que para consultar informações, o SELECT não funciona sozinho. Este comando necessita estar acompanhado, no mínimo, pela cláusula FROM. A cláusula SELECT é usada para relacionar os atributos desejados como resultado da consulta, enquanto a cláusula FROM relaciona as tabelas que serão pesquisadas na consulta. Caso se queira a lista de todos os atributos, pode-se descrevê-los um a um, ou simplesmente utilizar o símbolo '\*', que significa, na linguagem SQL, "todos os atributos". Para se impor restrições ao resultado de uma consulta, pode ser utilizada a cláusula WHERE.

Quando a consulta é executada na Base de Dados, os dados são devolvidos na ordem em que foram inseridos. Caso seja necessário algum tipo de ordenação, essa deve ser feita utilizando-se outro comando disponível na linguagem SQL, a ORDER BY. O comando ORDER BY faz com que o resultado da consulta apareça com uma certa ordem. Ela é posicionada no fim da consulta e nela são relacionados os campos a serem utilizados na ordenação com a devida forma de ordenação ASC ou DESC, respectivamente ascendente e descendente.

Exemplos concretos de estes comandos, bem como da construção de uma Base de Dados, serão apresentados mais adiante quando se entrar nos detalhes do programa.

## 2 - Planear a aplicação:

Como qualquer outro processo de desenvolvimento de software, a construção da aplicação deve ser pensada antes da sua construção. O método que usei resume-se no seguinte diagrama:

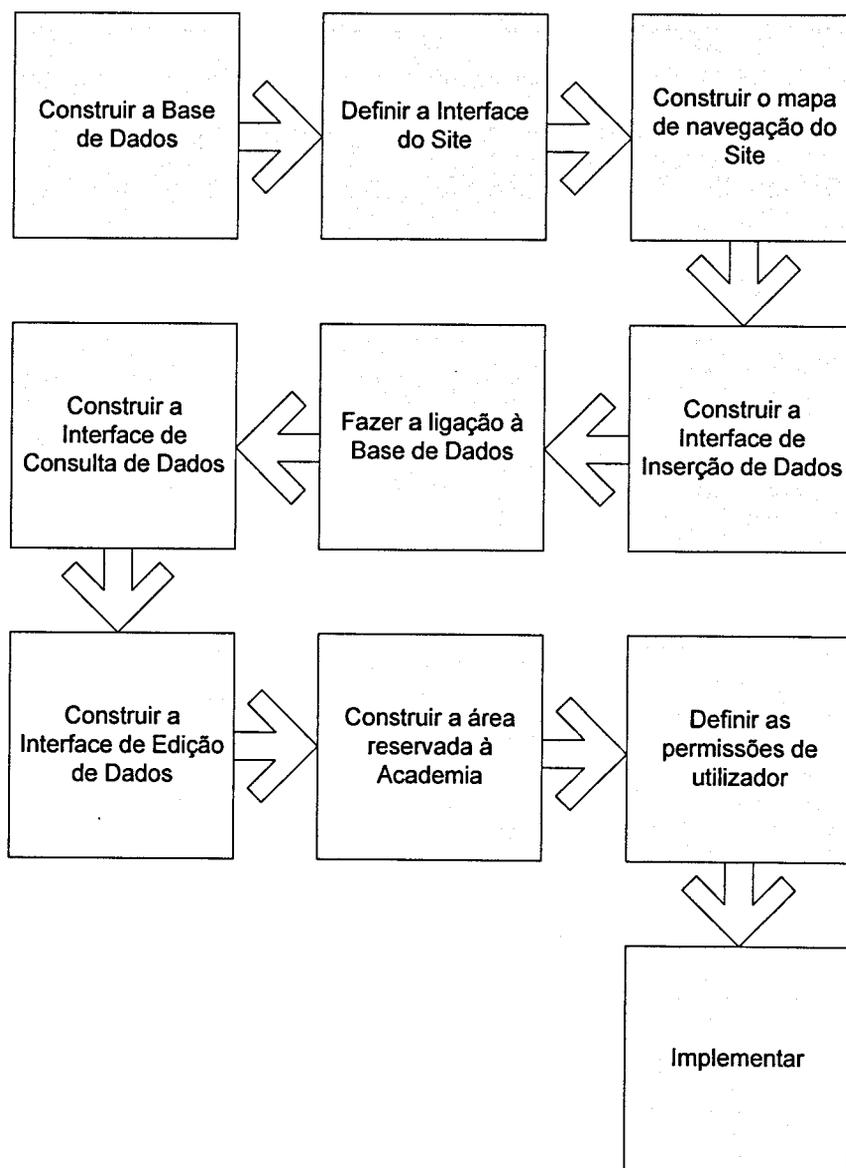


**Figura 6: Etapas do projecto**

Começemos então a explicar cada um destes pontos.

## 2.1 - Estratégia e Visão

Este trabalho pretende a construção de uma Base de Dados e o acesso à mesma. Assim sendo, é necessária a construção de uma interface com boa usabilidade para que o usuário seja capaz de utilizá-la de modo fácil e rápido. Deste modo, planeei este projecto da seguinte maneira:



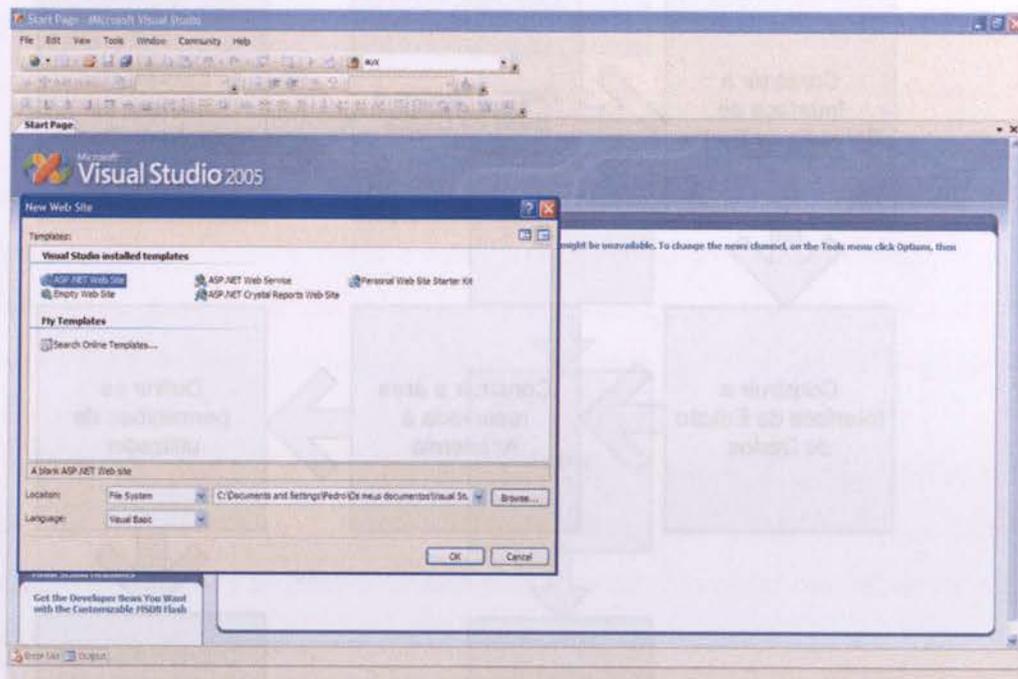
**Figura 7: Estratégia de construção do projecto**

Então após a definição da estratégia de construção passamos ao segundo ponto das etapas de projecto: o design da aplicação. Este ponto relaciona-se também com a

construção da mesma pois tendo o Visual Studio 2005 uma componente visual tão forte (o Visual Web Developer) à medida que vamos construindo a aplicação em tempo real apercebemo-nos do seu design. Deste modo começo agora a explicar a construção desta aplicação.

## 2.2 - Construção:

Como já foi referido, este projecto pretende a construção de uma aplicação recorrendo a ASP.NET. Assim devemos começar por abrir o Visual Studio 2005 e em File seleccionar New Web Site. Aparece o menu da figura abaixo e aí escolhemos ASP.NET Web Site, indicamos que a localização dos ficheiros será local (File System), especificamos a linguagem (Visual Basic) e damos um nome ao projecto.



**Figura 8 : A construção de um ASP.NET Web Site**

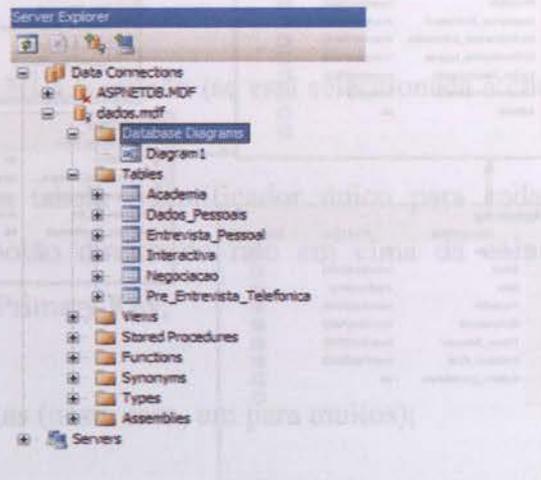
O próximo passo será a construção da Base de Dados.

### 2.2.1 - Construção da Base de Dados:

Para criar uma nova Base de Dados, no Web Site criado devemos ir ao Server Explorer e seguir os seguintes passos:

1. Clicar com o botão direito do rato em Data Connections e escolher Create New SQL Server Database; Aparece a caixa New SQL Server Database;
2. Escolhe-se o nome do servidor onde a Base de Dados estará localizada (no nosso caso é localhost);
3. Escolhe-se Windows Authentication;
4. Escolher o nome da nova Base de Dados (no nosso caso é “dados”);
5. Clicar OK;

A nova Base de Dados, (dados.mdf), aparece no Server Explorer.



**Figura 9 : O Server Explorer**

Quando criamos a Base de Dados no Server Explorer esta conterá pastas com objectos como as tabelas (Tables), funções, (Functions) e procedimentos (Stored Procedures). Para desenvolver a estrutura da Base de Dados podemos fazê-lo a partir do

diagrama da Base de Dados. Assim, para criar um diagrama, devemos clicar com o botão direito do rato em cima da pasta Database Diagrams e seleccionar Add New Diagram. Aqui podemos criar tabelas, alterá-las, bem como criar relações entre elas. Podemos também criar tabelas a partir da pasta Tables, clicando com o botão direito do rato e seleccionando Add New Table. Então inserimos a informação relevante para a tabela. Explicado mais adiante.

O diagrama da presente aplicação apresenta-se na figura seguinte.

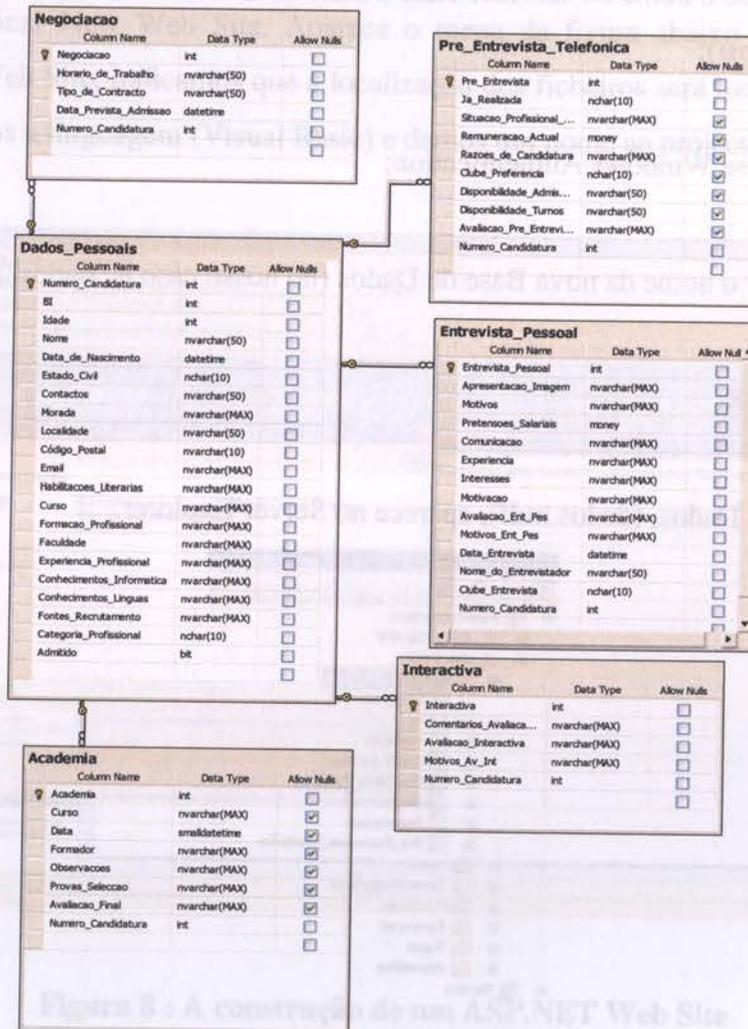
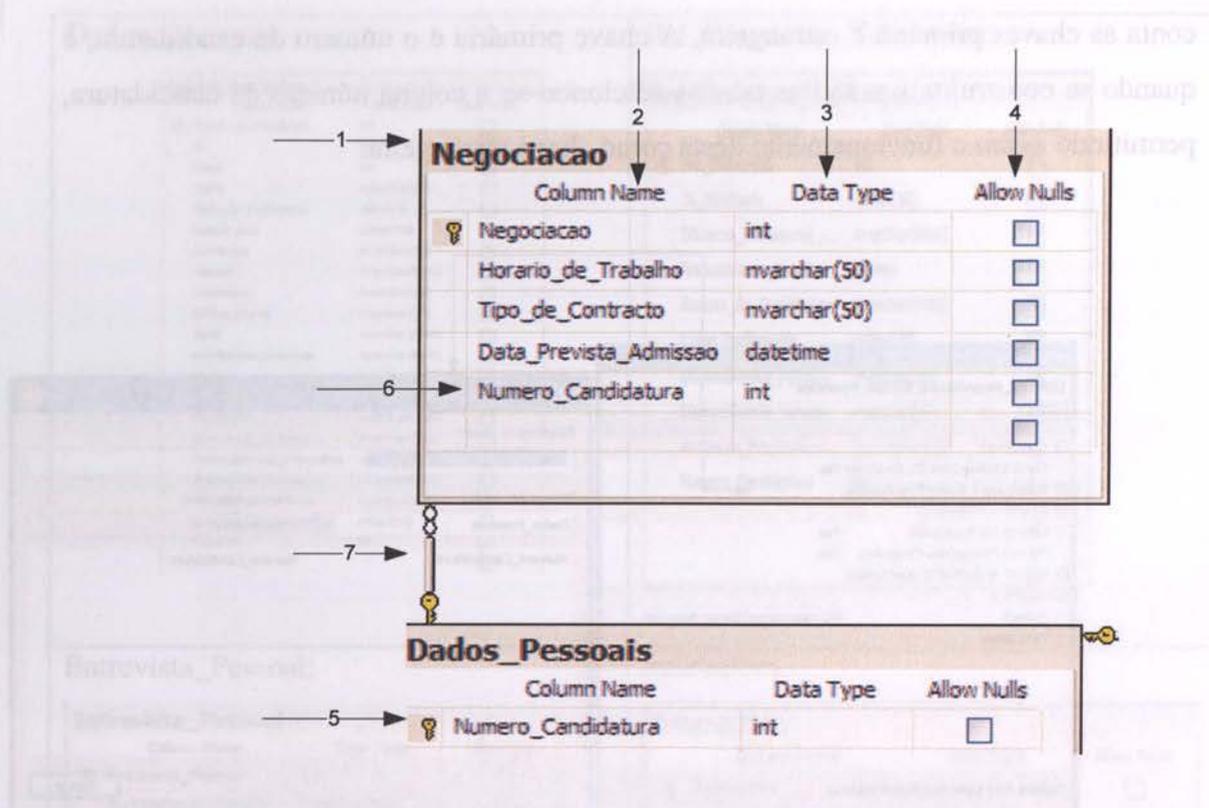


Figura 10: Diagrama da Base de Dados

Para entender a informação contida no diagrama e em específico em cada tabela, apresenta-se a figura seguinte:



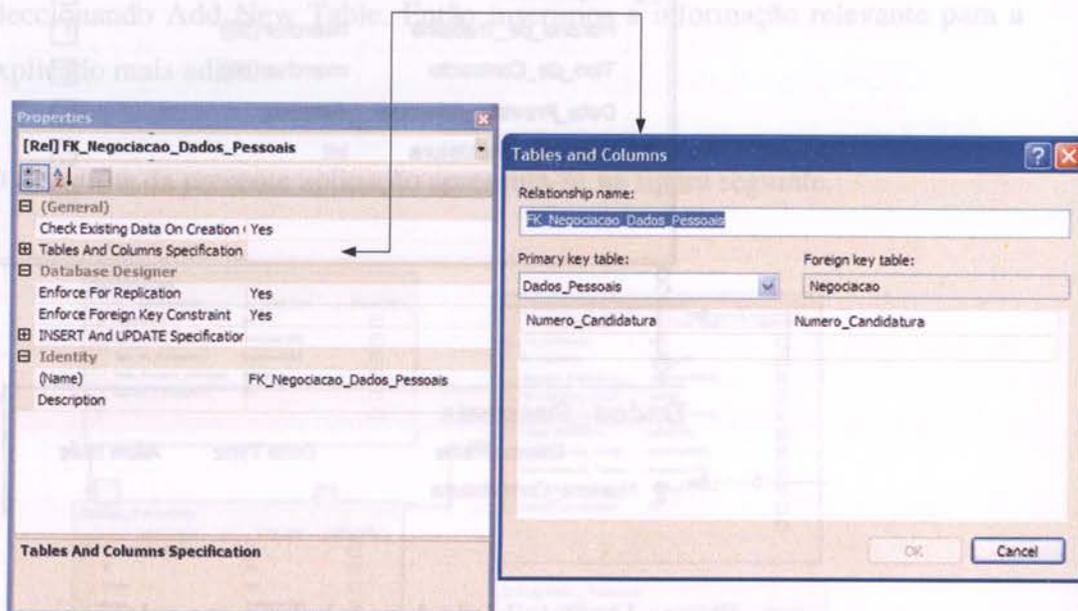
**Figura 11: Detalhe de duas tabelas e sua relação**

- 1- Nome da tabela (este deve ser único na base de dados);
- 2- Nome da coluna;
- 3- Tipo de dados;
- 4- Se permite o valor NULL ou não (se está seleccionada a checkbox é porque permite);
- 5- Chave primária da tabela (identificador único para cada tabela), criada clicando com o botão direito do rato em cima da coluna pretendida e seleccionando Set Primary Key;
- 6- Chave estrangeira;
- 7- Relação entre tabelas (neste caso, um para muitos);

Estes aspectos são comuns a todas as outras tabelas, mudando apenas o nome de cada tabela, chaves primárias, o nome das respectivas colunas bem como o tipo de dados associado a cada uma delas.

Para concluir a construção da Base de Dados falta apenas explicar como implementar as relações entre tabelas. Com o Visual Web Developer tal tarefa é simplificada bastando para tal seleccionar as tabelas que queremos relacionar tendo em

conta as chaves primária e estrangeira. A chave primária é o número de candidatura, e quando se construíram as outras tabelas adicionou-se a coluna número de candidatura, permitindo assim o funcionamento desta como chave estrangeira.



**Figura 12: Relação entre tabelas**

Utilizando as tabelas anteriormente explicadas, Negociação e Dados Pessoais, passo a explicar como se constrói a relação entre elas. Então, após seleccionar as tabelas pretendidas aparece uma relação entre elas. Clica-se com o botão direito do rato e escolhe-se Properties. Aparece o menu da esquerda, na figura acima. Como se pode ver, o nome da relação é FK\_Negociacao\_Dados\_Pessoais. Para alterá-la clicamos em Tables and Columns Specification e aparece o menu da direita na figura acima. Aqui, podemos alterar o nome da relação bem como as chaves intervenientes. Como se pretende uma relação de um para muitos a Chave estrangeira (Foreign Key) é o número de candidatura (numero\_candidatura) da tabela Negociacao. A chave primária (Primary Key) será numero\_candidatura da tabela Dados\_Pessoais. Para terminar clica-se Ok e está feita a relação entre tabelas.

Foram criadas cinco tabelas, cada uma contendo informação sobre determinado tópico relacionado com o candidato.

São elas: Dados\_Pessoais, Pre\_Entrevista\_Telefonica, Entrevista\_Pessoal, Interactiva, Negociacao e Academia. Apresento em pormenor cada uma delas:

## Dados\_Pessoais:

Column Name	Data Type	Allow Nulls
Numero_Candidatura	int	<input type="checkbox"/>
BI	int	<input type="checkbox"/>
Idade	int	<input type="checkbox"/>
Nome	nvarchar(50)	<input type="checkbox"/>
Data_de_Nascimento	datetime	<input type="checkbox"/>
Estado_Civil	nchar(10)	<input type="checkbox"/>
Contactos	nvarchar(50)	<input type="checkbox"/>
Morada	nvarchar(100)	<input type="checkbox"/>
Localidade	nvarchar(50)	<input type="checkbox"/>
Código_Postal	nvarchar(10)	<input type="checkbox"/>
Email	nvarchar(100)	<input type="checkbox"/>
Habilitacoes_Literarias	nvarchar(100)	<input type="checkbox"/>
Curso	nvarchar(100)	<input type="checkbox"/>
Formacao_Profissional	nvarchar(100)	<input type="checkbox"/>
Faculdade	nvarchar(100)	<input type="checkbox"/>
Experiencia_Profissional	nvarchar(100)	<input type="checkbox"/>
Conhecimentos_Informatica	nvarchar(100)	<input type="checkbox"/>
Conhecimentos_Linguas	nvarchar(100)	<input type="checkbox"/>
Fontes_Recrutamento	nvarchar(100)	<input type="checkbox"/>
Categoria_Profissional	nchar(10)	<input type="checkbox"/>
Admitido	bit	<input checked="" type="checkbox"/>

## Pré\_Entrevista\_Telefonica:

Column Name	Data Type	Allow Nulls
Pre_Entrevista	int	<input type="checkbox"/>
Ja_Realizada	nchar(10)	<input type="checkbox"/>
Situacao_Profissional_...	nvarchar(MAX)	<input checked="" type="checkbox"/>
Remuneracao_Actual	money	<input checked="" type="checkbox"/>
Razoes_de_Candidatura	nvarchar(MAX)	<input checked="" type="checkbox"/>
Clube_Preferencia	nchar(10)	<input checked="" type="checkbox"/>
Disponibilidade_Adms...	nvarchar(50)	<input checked="" type="checkbox"/>
Disponibilidade_Turnos	nvarchar(50)	<input checked="" type="checkbox"/>
Avaliacao_Pre_Entrevi...	nvarchar(MAX)	<input checked="" type="checkbox"/>
Numero_Candidatura	int	<input type="checkbox"/>

## Entrevista\_Pessoal:

Column Name	Data Type	Allow Nulls
Entrevista_Pessoal	int	<input type="checkbox"/>
Apresentacao_Imagem	nvarchar(MAX)	<input type="checkbox"/>
Motivos	nvarchar(MAX)	<input type="checkbox"/>
Pretensoes_Salariais	money	<input type="checkbox"/>
Comunicacao	nvarchar(MAX)	<input type="checkbox"/>
Experiencia	nvarchar(MAX)	<input type="checkbox"/>
Interesses	nvarchar(MAX)	<input type="checkbox"/>
Motivacao	nvarchar(MAX)	<input type="checkbox"/>
Avaliacao_Ent_Pes	nvarchar(MAX)	<input type="checkbox"/>
Motivos_Ent_Pes	nvarchar(MAX)	<input type="checkbox"/>
Data_Entrevista	datetime	<input type="checkbox"/>
Nome_do_Entrevistador	nvarchar(50)	<input type="checkbox"/>
Clube_Entrevista	nchar(10)	<input type="checkbox"/>
Numero_Candidatura	int	<input type="checkbox"/>

## Interactiva:

Column Name	Data Type	Allow Nulls
Interactiva	int	<input type="checkbox"/>
Comentarios_Avaliaca...	nvarchar(MAX)	<input type="checkbox"/>
Avaliacao_Interactiva	nvarchar(MAX)	<input type="checkbox"/>
Motivos_Av_Int	nvarchar(MAX)	<input type="checkbox"/>
Numero_Candidatura	int	<input type="checkbox"/>

## Negociacao:

Column Name	Data Type	Allow Nulls
Negociacao	int	<input type="checkbox"/>
Horario_de_Trabalho	nvarchar(50)	<input type="checkbox"/>
Tipo_de_Contracto	nvarchar(50)	<input type="checkbox"/>
Data_Prevista_Admissao	datetime	<input type="checkbox"/>
Numero_Candidatura	int	<input type="checkbox"/>

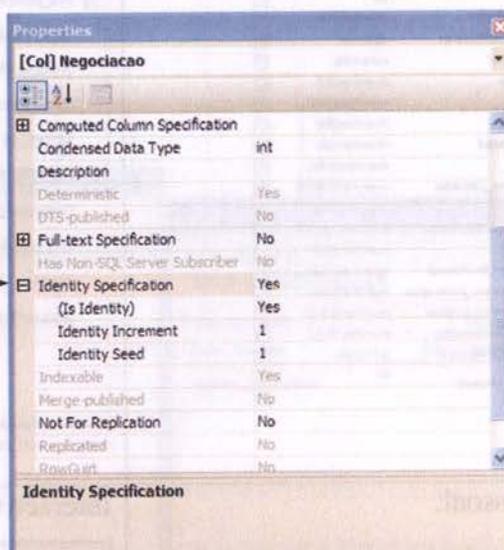
## Academia:

Column Name	Data Type	Allow Nulls
Academia	int	<input type="checkbox"/>
Curso	nvarchar(MAX)	<input checked="" type="checkbox"/>
Data	smalldatetime	<input checked="" type="checkbox"/>
Formador	nvarchar(MAX)	<input checked="" type="checkbox"/>
Observacoes	nvarchar(MAX)	<input checked="" type="checkbox"/>
Provas_Seleccao	nvarchar(MAX)	<input checked="" type="checkbox"/>
Avaliacao_Final	nvarchar(MAX)	<input checked="" type="checkbox"/>
Numero_Candidatura	int	<input type="checkbox"/>

Tabela 1: Pormenor das tabelas

A única chave primária introduzida pelo usuário será o número de candidatura (numero\_candidatura) para a tabela Dados\_Pessoais. As outras chaves primárias (uma

por tabela) são geradas automaticamente. Para programar este evento, recorre-se uma vez mais ao Visual Web Developer procedendo-se do modo indicado na figura:



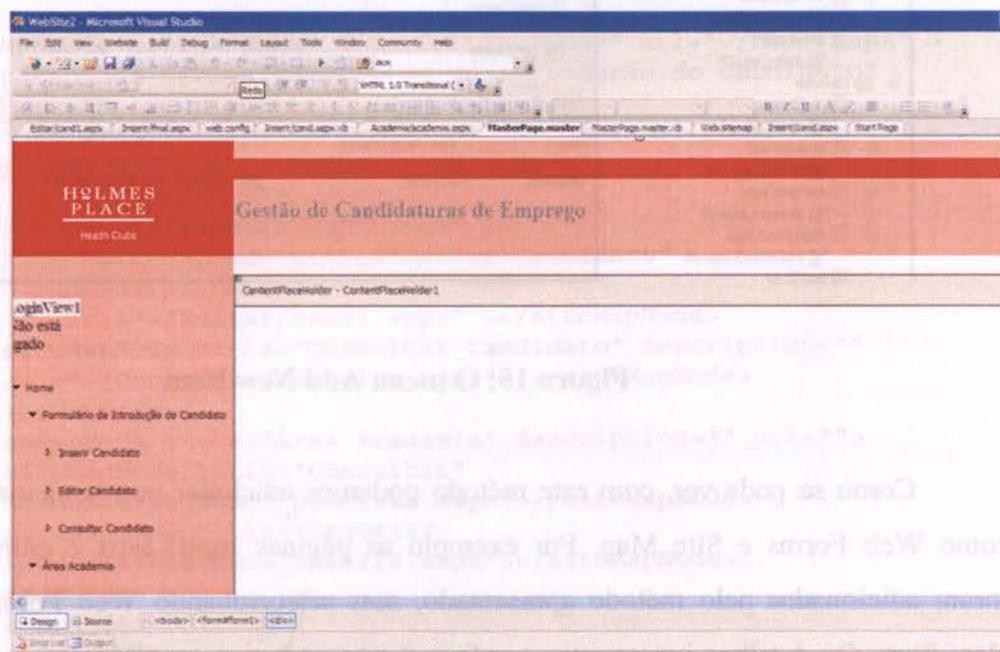
**Figura 13: A caixa properties da chave primária**

Para que apareça este menu, deve clicar-se com o botão direito do rato em cima da chave primária da tabela e seleccionar Properties. Então, em Identity Specification (onde aponta a seta) desdobramos o menu e em (Is Identity) pomos a Yes e queremos que aumente de um em um (Identity Increment → 1).

Para concluir este ponto, deve referir-se que a informação presente na Base de Dados é o resultado das orientações fornecidas pelos recursos humanos do Holmes Place.

## 2.3 - Definir a Interface do Site:

Como já foi referido, pretende-se que esta aplicação tenha uma boa usabilidade. Ou seja, que seja um produto fácil de usar e cómodo para o usuário. Assim, para a criação da Interface do Site, usei aquilo que se chama Masterpage.master. Todas as páginas do projecto, quando assim for indicado, terão a configuração esquemática desta página. Tal facto facilita a construção de um site pois não é preciso estar constantemente a construir partes que se repetem ao longo de um projecto. O ambiente escolhido para este trabalho é o que se apresenta na figura a seguir:

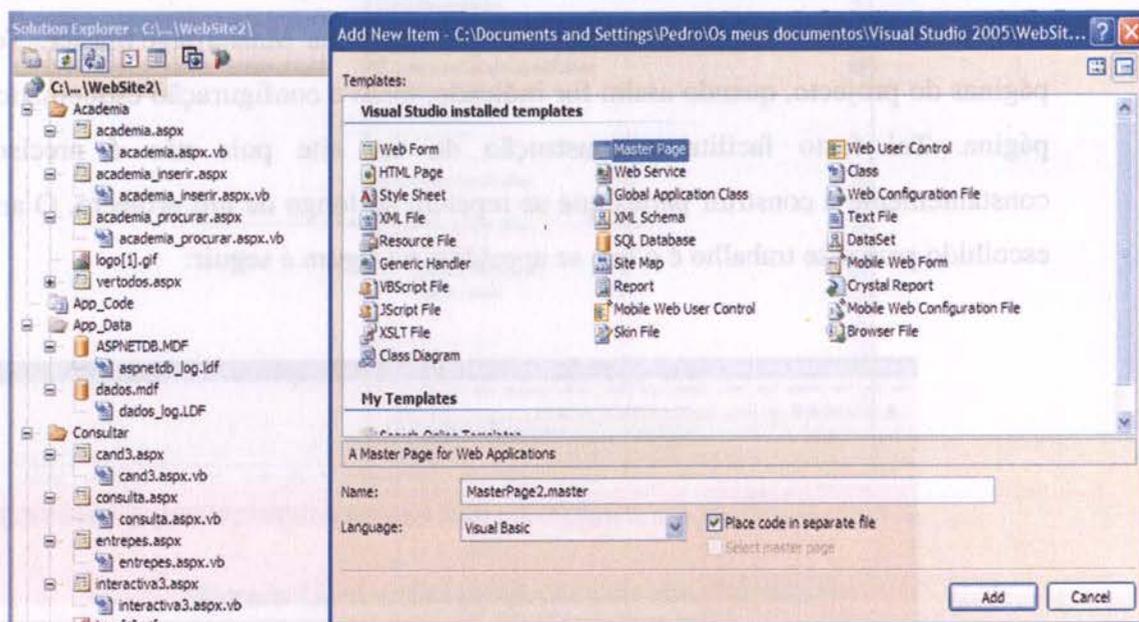


**Figura 14: A MasterPage**

Podemos identificar algumas áreas nesta página. São elas o menu do lado direito para a navegação no site. O ContentPlaceHolder, a área branca onde mais tarde serão arrastados os Web Controls a partir da Toolbox para construir os formulários desejados. Este controlo apenas está disponível para a MasterPage. Se tentarmos usá-lo noutra tipo de páginas será gerado um erro.

Assim, quando se cria uma nova página a partir do Solution Explorer podemos escolher a MasterPage a associar àquela. Podemos ter várias MasterPage para um mesmo projecto. Neste apenas se utilizou uma. Assim para se criar a MasterPage deve proceder-se do seguinte modo:

1. Clicar com o botão direito do rato no projecto. (c:\...\WebSite2\)
2. Escolher Add New Item
3. Escolher MasterPage



**Figura 15: O menu Add New Item**

Como se pode ver, com este método podemos adicionar outros elementos, tais como Web Forms e Site Map. Por exemplo as páginas `cand3.aspx` e `entrepes.aspx` foram adicionadas pelo método apresentado, mas seleccionando Web Form no Add New Item. Os detalhes importantes a referir é que podemos escolher o nome a dar à página a adicionar (Name). O tipo de linguagem a utilizar (Language). Na presente aplicação utilizou-se Visual Basic. Se queremos colocar o código em separado relativamente à página (o modelo code-behind, já explicado atrás). Pode-se ver que quando inserimos uma MasterPage a opção Select master page não está disponível, contudo para outros tipos de páginas estará e aí seleccionamos a MasterPage pretendida.

Se entrarmos na Source da página `MasterPage.master`, a primeira linha é:

```
<%@ Master Language="VB" CodeFile="MasterPage.master.vb"
Inherits="MasterPage" %>
```

E como se pode verificar temos `Language = "VB"` (Visual Basic) e o código será enviado para `"MasterPage.master.vb"` → modelo code-behind.

### 2.3.1 - Navegação:

A navegação é um componente essencial para criar um site bem desenhado e esta tarefa graças ao ASP.NET 2.0 está agora mais fácil. O ASP.NET 2.0 introduz a ideia de “site map”, que é quase como quem diz mapa do site. O Site Map é um ficheiro em XML que contém os links do site. Para criar um novo Site Map recorreremos ao método anteriormente apresentado mas escolhemos Site Map. O ficheiro criado terá por definição o nome Web.sitemap. No presente projecto o Site Map é:

```
<?xml version="1.0" encoding="utf-8" ?>
<siteMap xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0"
>
  <siteMapNode title="Home" description="Home" url="~/home.aspx" >
    <siteMapNode title="Formulário de Introdução do Candidato"
description=""
url="">
      <siteMapNode title="Inserir Candidato"
description=""
url="~/Inserir/cand.aspx" />
        <siteMapNode title="Editar Candidato"
description=""
url="~/Editar/cand1.aspx" ></siteMapNode>
      <siteMapNode title="Consultar Candidato" description=""
url="~/Consultar/consulta.aspx"> </siteMapNode>
    </siteMapNode>
    <siteMapNode title="Área Academia" description="" url="">
      <siteMapNode title="Consultar"
url="~/Academia/academia_procurar.aspx"></siteMapNode>
      <siteMapNode title="Inserir"
url="~/Academia/academia_inserir.aspx"></siteMapNode>
    </siteMapNode>
  </siteMapNode>
</siteMap>
```

Como se pode ver, a construção do Site Map inicia-se com <siteMap...> e termina com </siteMap>. Dentro deste vemos os elementos <siteMapNode>, </siteMapnode>, que permitem inserir um título (title =” título pretendido”), uma descrição (description = “ descrição”) e um URL. É neste URL que se define para onde se salta quando se clica em determinado título. Tome-se o exemplo:

```
  <siteMapNode title="Inserir"
url="~/Academia/academia_inserir.aspx"></siteMapNode>
```

Vemos que tem como título, Inserir, e quando se clica nesse título vai-se para a página academia\_inserir.aspx que tem a path apresentada no url.

Depois de definido o Site Map constrói-se o menu de navegação. Para tal recorreu-se ao novo controlo TreeView do ASP.NET 2.0:

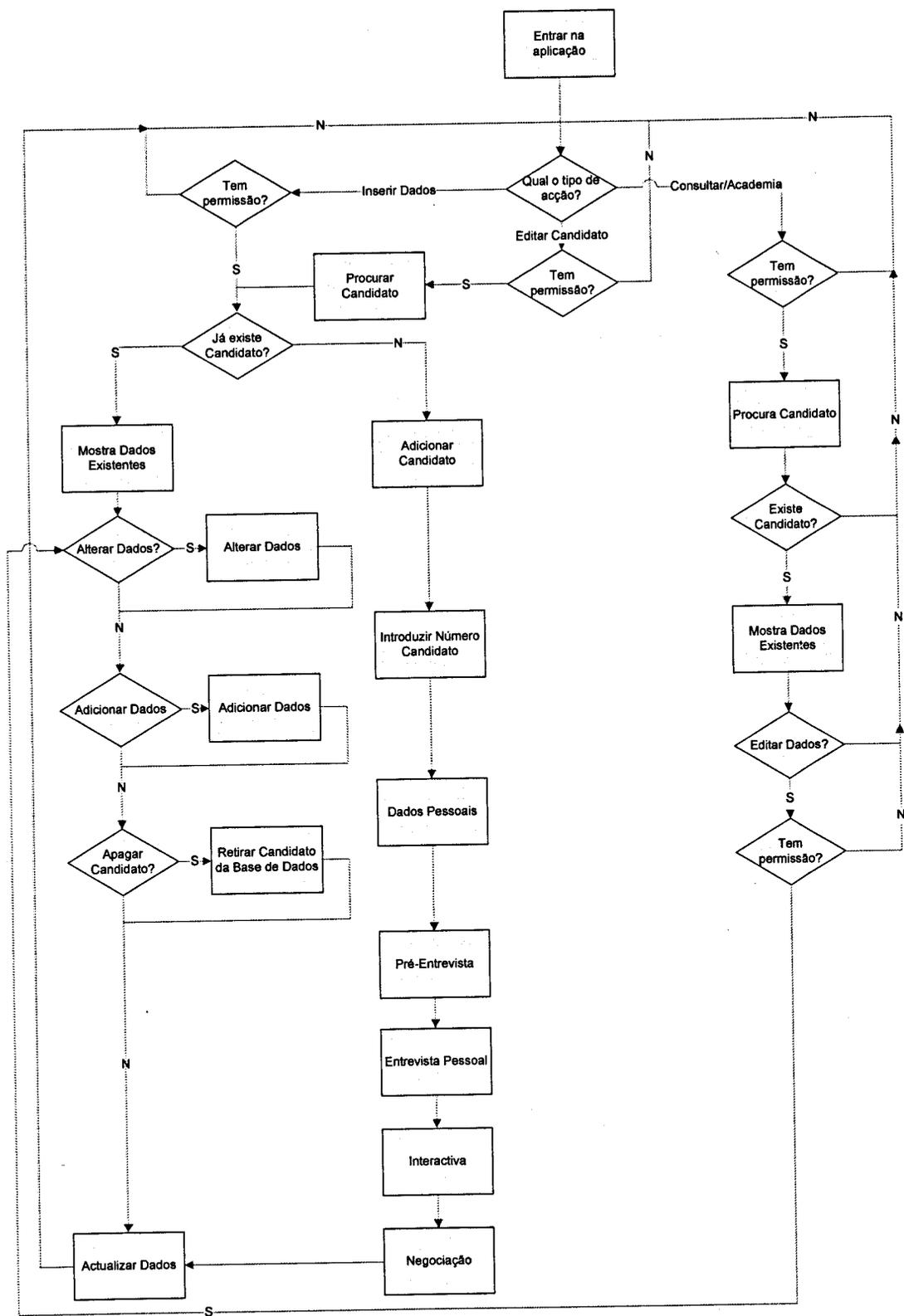
```
<asp:TreeView ID="TreeView1" runat="server"
DataSourceID="SiteMapDataSource1" style="table-layout: auto; cursor:
hand; border-collapse: separate; background-color: darksalmon; color:
gray;" ForeColor="DarkGray" Height="175px" ImageSet="Arrows"
Width="196px" Font-Names="Times New Roman" Font-Size="XX-Large">
    <ParentNodeStyle Font-Bold="False" />
    <HoverNodeStyle Font-Underline="True"
ForeColor="#5555DD" />
    <SelectedNodeStyle Font-Underline="True"
ForeColor="Gray" HorizontalPadding="0px"
VerticalPadding="0px" />
    <NodeStyle Font-Names="Tahoma" Font-
Size="10pt" ForeColor="Black" HorizontalPadding="5px"
NodeSpacing="0px" VerticalPadding="0px" />
</asp:TreeView>
<asp:SiteMapDataSource ID="SiteMapDataSource1" runat="server" />
```

Como se pode ver, este controlo, como qualquer controlo em ASP.NET, começa com <asp:...>, e como se trata de um TreeView, temos, <asp:TreeView>. Para fechar o controlo, </asp:TreeView>. O ponto importante a destacar das linhas de código atrás apresentadas é o DataSourceID="SiteMapDataSource1". É nesta linha de código que se relaciona o TreeView com o Site Map construído. Todas as outras linhas são configurações de espaçamentos, texto e tamanhos.

Penso que é boa altura para explicar o comando runat = "server", pois este está presente em quase todos os controlos. Este comando indica que o controlo funciona do lado do servidor e não do lado do cliente. Para ser mais específico, por defeito, os eventos e dados criados pelo utilizador no cliente, são enviados para o servidor, e podemos então manipulá-los. É um modelo suficientemente robusto para permitir operações complexas.

### 2.3.2 - Funcionamento da aplicação:

Para que seja criada uma interface eficaz e seja possível uma boa navegação no site é preciso entender bem o que se pretende que o programa realize. No fluxograma seguinte apresenta-se o funcionamento da aplicação.



**Figura 16: Fluxograma da aplicação**

Passo a explicar o fluxograma. Primeiro entramos na aplicação (Entrar na aplicação). O segundo passo será escolher a tarefa a realizar (Qual o tipo de acção?). Podemos ter quatro tipos de acção à escolha:

- Inserir Candidato
- Consultar Candidato
- Editar Candidato
- Entrar na área da Academia

É de salientar que a acção de consulta do candidato e a acção de entrar na área Academia são tratadas de modo igual no fluxograma, pois quanto ao funcionamento a nível de passos a seguir, são iguais. Contudo, deve notar-se que dizem respeito a áreas distintas no programa.

Para cada uma destas acções, o programa verificará as permissões de utilizador (Tem permissão?), e se a permissão for positiva, passa-se para o próximo ponto. Se ao contrário, o utilizador não tiver permissão para aceder a uma determinada acção, volta ao menu inicial sendo pedido novamente a escolha de uma acção.

Será lógico começar pela inserção de um novo candidato. Cada candidato tem associado um número de candidatura. Após a inserção deste, (Introduzir número de candidato), seguem-se os passos indicados, cada um deles correspondendo à inserção de informação que depois será enviada para a Base de Dados. A sequência a seguir é:

1. Dados Pessoais
2. Pré-Entrevista Telefónica
3. Entrevista Pessoal
4. Entrevista Interactiva
5. Negociação

Quando se termina a inserção de dados, estes são enviados para a Base de Dados, sendo esta actualizada com a nova informação (Actualizar Dados). Após este processo regressa-se ao início da aplicação. Supondo que se pretende inserir um candidato já existente, a aplicação vai informar que o candidato já existe na Base de Dados mostrando os dados relacionados com aquele.

Se o utilizador tem permissão para editar dados, após a procura do candidato (Procurar Candidato), e sabendo que a informação associada com este existe na Base de Dados, pode alterar, adicionar ou mesmo apagar aquele da Base de Dados, sendo esta actualizada no fim destas operações.

Quanto à consulta de informação, esta é realizada após a verificação da permissão do utilizador para tal, e se o candidato existe na Base de Dados, após a aplicação dos critérios de consulta, são apresentados os dados relativos ao candidato (Mostra Dados Existentes). A partir daqui, os dados podem ser editados se existe permissão para tal, faz-se nova consulta ou regressa-se ao início da aplicação. Como já foi referido, a área da Academia apresenta um funcionamento igual à consulta de informação, sendo as diferenças as permissões de utilizador bem como os dados existentes nas duas áreas.

Para concluir este ponto devo referir que o modo como foram criadas as áreas de Inserção, Consulta, Edição e Área de Academia será explorado mais adiante.

### 2.3.3 - Construir a Interface:

Após ter construído o ambiente onde se vai desenvolver o projecto, é necessário construir as interfaces responsáveis pela inserção, consulta, edição de dados e área da Academia. Antes de entrar em qualquer acção há que definir a Home Page (página de início). Na figura a seguir esta é apresentada:

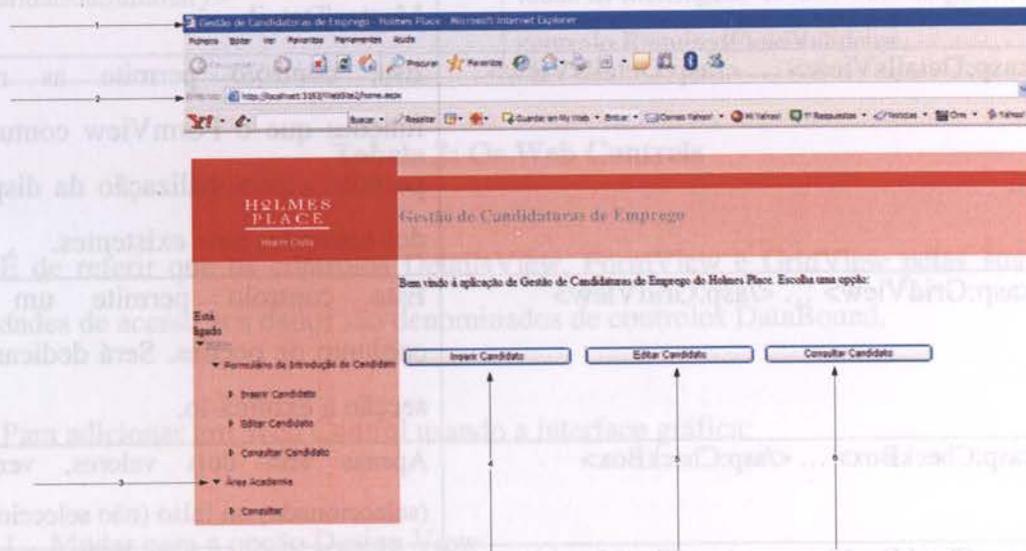


Figura 17: A Home Page

Passo a explicar os pontos:

1. Título da página (definido em HTML na página home.aspx)
2. O endereço da página. Como se verifica, a aplicação corre no localhost.
3. O link para à área da Academia, definido no menu de navegação TreeView
4. O botão para entrar na área de inserção do candidato
5. O botão para entrar na área de consulta do candidato
6. O botão para entrar na área de edição do candidato

Vamos seguir a sequência de inserção, consulta, edição e finalmente área de Academia. Para entender tudo o que se vai passar a partir daqui será conveniente explorar os Web Controls existentes em ASP.NET 2.0. Eles são mais de quarenta e cinco, contudo neste projecto apenas alguns foram utilizados.

Nome do Controlo	Descrição
<code>&lt;asp:FormView&gt; ... &lt;/asp:FormView&gt;</code>	Permite mostrar dados e aceder a eles através de ligações à base de dados a partir dos controlos existentes no FormView. É possível criar templates e assim criar formulários de introdução de dados. Muitas vezes este controlo está associado com o GridView em modo Master/Detail
<code>&lt;asp:DetailsView&gt; ... &lt;/asp:DetailsView&gt;</code>	Este controlo permite as mesmas funções que o FormView contudo não permite a personalização da disposição dos controlos nele existentes.
<code>&lt;asp:GridView&gt; ... &lt;/asp:GridView&gt;</code>	Este controlo permite um vasto conjunto de opções. Será dedicada uma secção a explicá-lo.
<code>&lt;asp:CheckBox&gt; ... &lt;/asp:CheckBox&gt;</code>	Apenas tem dois valores, verdadeiro (seleccionado) ou falso (não seleccionado)
<code>&lt;asp:CheckBoxList&gt;... &lt;/asp:CheckBoxList&gt;</code>	Um grupo de CheckBox

<code>&lt;asp:View&gt; ... &lt;/asp:View&gt;</code>	Este controlo permite substituir o controlo Panel, funcionando como uma área de conteúdos.
<code>&lt;asp:Multview&gt; ... &lt;/asp:MultiView&gt;</code>	Permite a associação de vários controlos View criando assim várias secções dentro da mesma página.
<code>&lt;asp:TreeView&gt; ... &lt;/asp:TreeView&gt;</code>	Controlo de navegação
<code>&lt;asp:DropDownList&gt; ... &lt;/asp:DropDownList&gt;</code>	Permite a selecção de um único valor dentro de um conjunto de possíveis valores numa caixa de texto desdobrável.
<code>&lt;asp:HyperLink&gt; ... &lt;/asp:HyperLink&gt;</code>	Uma ligação para outra área.
<code>&lt;asp:TextBox&gt; ... &lt;/asp:TextBox&gt;</code>	Uma caixa de texto que permite uma ou várias linhas.
<code>&lt;asp:RadioButtonList&gt; &lt;/asp:RadioButtonList&gt;</code>	Um grupo de botões de rádio, permite seleccionar um entre vários.
<code>&lt;asp:Button&gt; ... &lt;/asp:Button&gt;</code>	Quando se clica, o controlo envia uma acção para o servidor.
<code>&lt;asp:RequiredFieldValidator&gt; ... &lt;/asp:RequiredFielValidator&gt;</code>	Este controlo permite definir que determinado campo de informação deve ser preenchido e quando tal não acontece devolve uma mensagem de erro.
<code>&lt;asp:ValidationSummary&gt; ... &lt;/asp:ValidationSummary&gt;</code>	Este controlo permite fazer o resumo de todas as mensagens de erro com origem no controlo RequiredFieldValidator

**Tabela 2: Os Web Controls**

É de referir que os controlos DetailsView, FormView e GridView pelas suas propriedades de acesso aos dados são denominados de controlos DataBound.

Para adicionar um Web Control usando a interface gráfica:

1. Mudar para a opção Design View



interpretação por parte do programador que não esteja familiarizado com a aplicação;

7. Causes Validation: Ao efectuar-se o comando verifica se tudo está de acordo com as definições, normalmente coloca-se a true quando existe um CompareValidator o um RequiredFieldValidator;
8. Command Argument: insere-se aqui o argumento do comando a realizar;
9. Command Name: o nome do comando, nas funções associadas à Base de Dados será Insert, para enviar os dados, Delete para apagar, Update para actualizar após uma edição e Cancel para cancelar uma acção;

Depois da Home Page criada, é necessário construir a Interface de Inserção do Candidato.

**Figura 19: A Interface de Inserção do Candidato (1)**

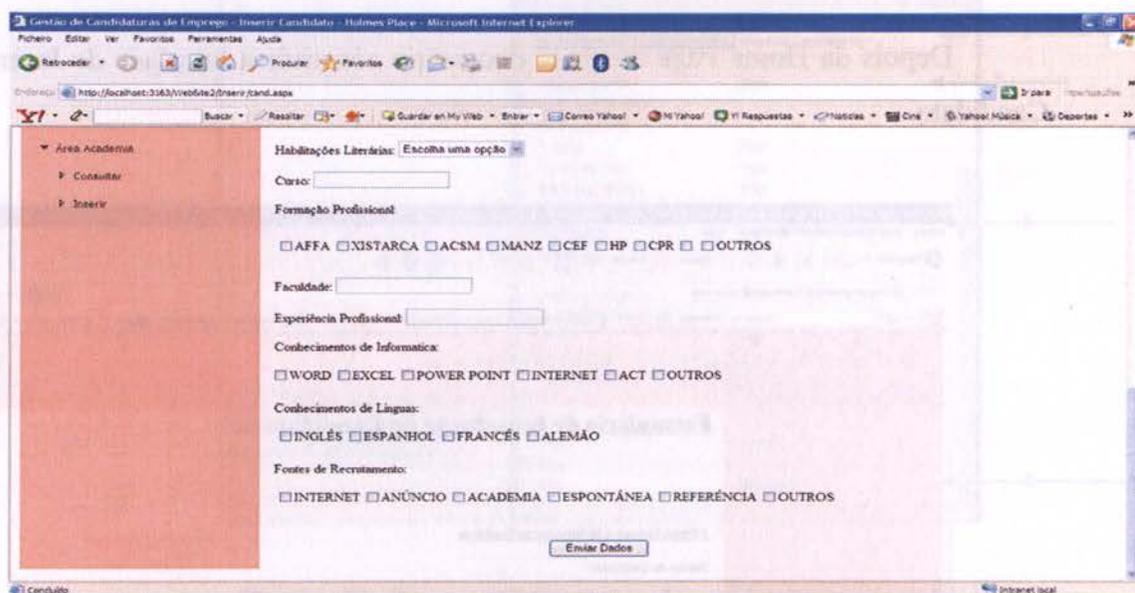
Como se pode ver, os controlos existentes dizem respeito à informação que se pretende enviar para a Base de Dados. Como já se sabe, para construir a interface, bastou arrastar os controlos da Toolbox para a área de desenho. Para construir as DropDownList pode-se optar por um de dois métodos. Via programação:

```

<asp:DropDownList ID="DropDownList1" runat="server" SelectedValue="<%#
Bind("Categoria_Profissional") %>'
    Style="position: relative">
    <asp:ListItem Selected="True">Escolha uma
op&#231;&#227;o</asp:ListItem>
    <asp:ListItem>MS</asp:ListItem>
    <asp:ListItem>PT/GT</asp:ListItem>
    <asp:ListItem>SALES</asp:ListItem>
    <asp:ListItem>COP</asp:ListItem>
    <asp:ListItem>MADM</asp:ListItem>
    <asp:ListItem>SEC</asp:ListItem>
    </asp:DropDownList>

```

Onde dentro da tag <asp:ListItem> ... </asp:ListItem> introduz-se a opção pretendida. Ou então podemos recorrer às propriedades da DropDownList (pode-se aceder a elas a partir da tecla F4), e aí escolher os itens a inserir.



**Figura 20: A Interface de Inserção do Candidato (2)**

É conveniente explicar como se constrói a CheckBoxList. Esta é construída de modo idêntico à DropDownList, mudando apenas o nome do controlo. Veja-se o código:

```

<asp:CheckBoxList ID="CheckBoxList1" runat="server"
RepeatDirection="Horizontal" SelectedValue="<%#
Bind("Formacao_Profissional") %>' Style="left: 0px; position:
relative;
    top: 8px">
    <asp:ListItem>AFFA</asp:ListItem>
    <asp:ListItem>XISTARCA</asp:ListItem>
    <asp:ListItem>ACSM</asp:ListItem>
    <asp:ListItem>MANZ</asp:ListItem>
    <asp:ListItem>CEF</asp:ListItem>
    <asp:ListItem>HP</asp:ListItem>

```

```

        <asp:ListItem>CPR</asp:ListItem>
        <asp:ListItem>OUTROS</asp:ListItem>
    </asp:CheckBoxList>

```

Os controlos existentes estão inseridos num FormView (linha 4), que por sua vez está inserido num View (linha 3), que está dentro de um MultiView (linha 1). A associação de View com MultiView possibilita a implementação de um wizard, onde o utilizador vai introduzindo os dados relevantes para cada View. O código responsável por esta funcionalidade é:

```

1. <asp:MultiView ID="MultiView1" runat="server"
2.     ActiveViewIndex="0">
3.     <asp:View ID="View1" runat="server">
4.         <asp:FormView ID="FormView1" runat="server"
5.             DataKeyNames="Numero_Candidatura" DataSourceID="SqlDataSource1"
6.         </asp:FormView>
7.     </asp:View>
8. </asp:MultiView>

```

As linhas 6, 7 e 8 são responsáveis por fechar o controlo MultiView, View e FormView respectivamente.

Para que se possa mudar de View, sempre que se clica em Enviar Dados, o evento Button1\_Click é disparado fazendo correr a função:

```

Protected Sub Button1_Click(ByVal sender As Object, ByVal e As
System.EventArgs)
    MultiView1.ActiveViewIndex += 1
End Sub

```

Define-se o ActiveViewIndex inicial como zero (linha 2), e então sempre que se dispara o evento Button1\_Click, o índice activo do MultiView altera-se passando para o próximo View, e consequentemente permitindo o acesso a controlos diferentes.

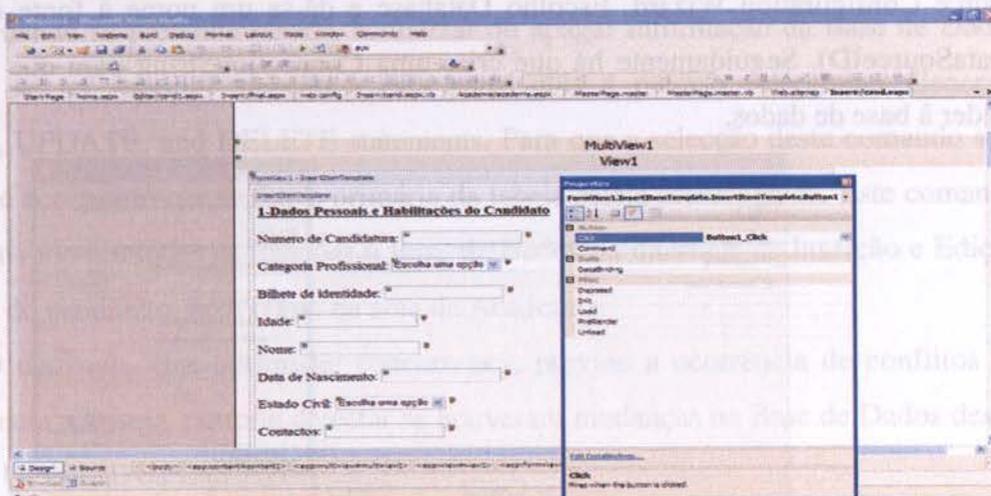


Figura 21: Detalhe do menu Properties na opção eventos

### 3 - A ligação à Base de Dados:

Para que se possam enviar os dados para a Base de Dados tem que se associar o FormView com uma fonte de dados. Esta tarefa pode ser feita de modo totalmente visual, contudo, tal não implica que não possa ser feita via programação. Vou apresentar o modo que considero mais prático, ou seja, através do Visual Web Developer. No final, apresentarei as linhas de código geradas.

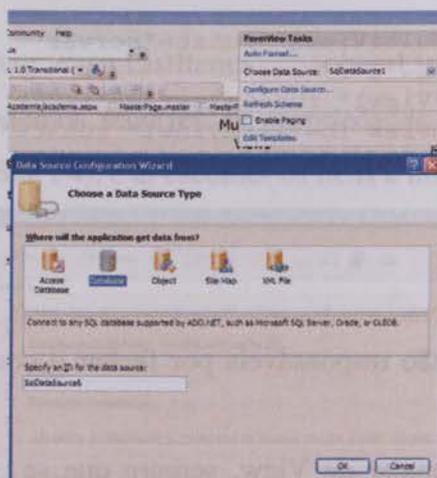


Figura 22: A criação de uma ligação à Base de Dados

No smart tag do FormView, (um símbolo que aparece no canto superior direito), clicamos e então aparece o FormView Tasks. Como queremos definir uma fonte de dados, selecciono Choose Data Source → New Data Source. Aparece a caixa Data Source Configuration Wizard. Escolho Database e dá-se um nome à fonte de dados (DataSourceID). Seguidamente há que criar uma ConnectionString para que se possa aceder à base de dados.

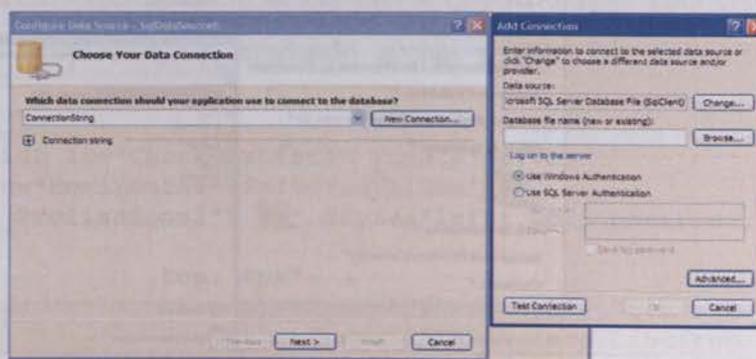


Figura 23: A criação da ConnectionString

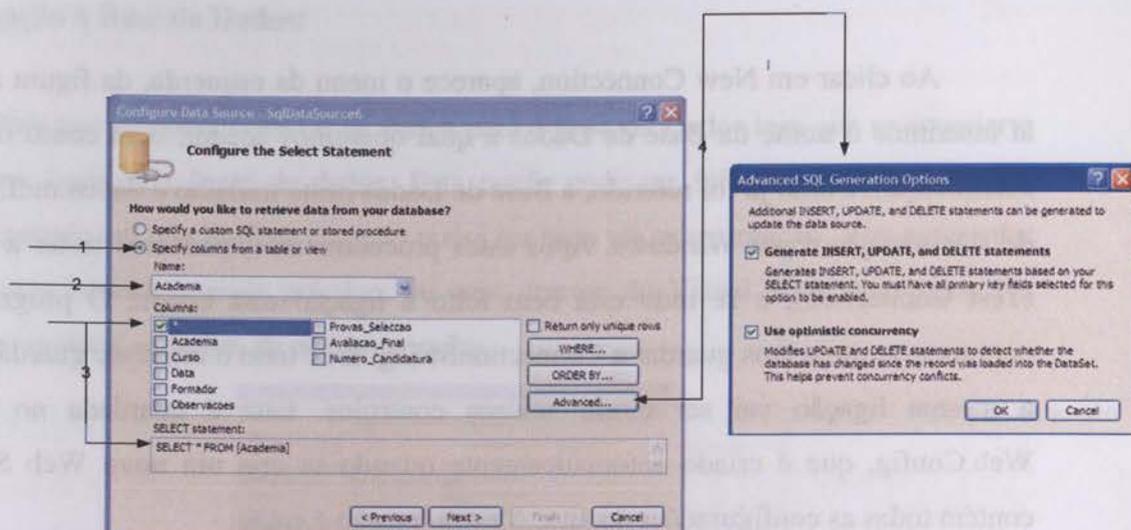
Ao clicar em New Connection, aparece o menu da esquerda, da figura acima, e aí inserimos o nome da Base de Dados à qual queremos aceder, bem como o tipo de autenticação. Como já foi referido, a Base de Dados deste trabalho é dados.mdf, e o tipo de autenticação é por Windows. Após estes procedimentos, podemos testar a ligação, (Test Connection), e se tudo está bem feito a ligação está criada. O programa vai perguntar se queremos guardar a ConnectionString. É de todo o interesse guardá-la, pois a mesma ligação vai ser usada noutros controlos. Esta é guardada no ficheiro Web.Config, que é criado automaticamente quando se cria um novo Web Site pois contém todas as configurações do site. A nova ligação é então:

```
<connectionStrings>
<add name="ConnectionString" connectionString="Data
Source=.\SQLEXPRESS;AttachDbFilename=|DataDirectory|\dados.mdf;Integra
ted Security=True;User Instance=True"
providerName="System.Data.SqlClient" />
</connectionStrings>
```

A etapa seguinte é seleccionar a informação à qual se pretende aceder. Para tal, na caixa de diálogo, Configure Data Source, seleccionamos Specify columns from a table or view, (seta 1). Onde está a seta 3, é onde se escolhem todas as colunas de determinada tabela da Base de Dados, que é possível alterar (seta 2). Como se pode ver, ao escolher todas as colunas da tabela Academia, é gerado o código SQL SELECT \* FROM [Academia]. É possível seleccionar apenas algumas colunas da tabela. A seta 4, diz respeito às opções avançadas.

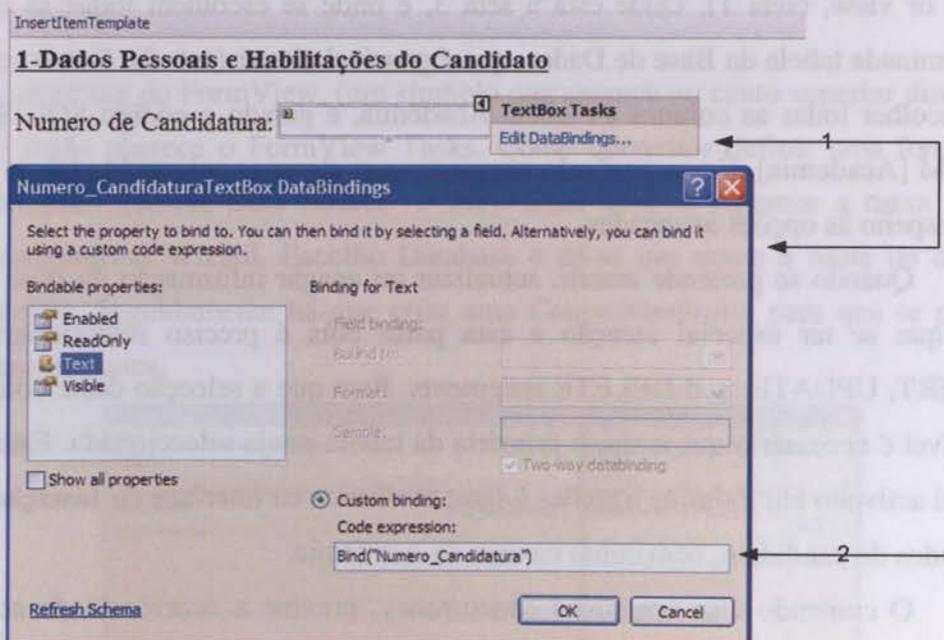
Quando se pretende inserir, actualizar ou apagar informação da Base de Dados tem que se ter especial atenção a esta parte pois é preciso seleccionar Generate INSERT, UPDATE, and DELETE statements. Para que a selecção deste comando seja possível é necessário que a chave primária da tabela esteja seleccionada. Este comando estará activado em todas as ligações à Base de Dados na interface de Inserção e Edição de dados do candidato, bem como na área da Academia.

O comando Use optimistic concurrency, previne a ocorrência de conflitos de concorrência. Ou seja, permite detectar se houveram mudanças na Base de Dados desde o último UPDATE ou DELETE.



**Figura 24: A configuração da fonte de dados**

Agora que o FormView já está ligado à Base de Dados, há que associar cada um dos controlos com a coluna correspondente. Para isso, mais uma vez utiliza-se o Visual Web Developer.



**Figura 25: Associar controlos à base de dados**

Para efectuar a ligação entre um controlo e coluna correspondente na Base de Dados, clica-se no smart tag do controlo desejado. Na figura acima, escolheu-se como exemplo o número de candidatura. Então, ao seleccionar-se Edit Databinding, aparece a caixa Numero\_CandidaturaTextBox DataBindings, e em custom binding, introduzimos o código Bind("numero\_candidatura"). Quando se termina esta operação, a caixa de texto, Numero\_CandidaturaTextBox está ligada à coluna numero\_candidatura da Base de Dados. Apresenta-se o código gerado por esta operação:

```

1.<InsertItemTemplate>
2.<asp:TextBox ID="Numero_CandidaturaTextBox" runat="server" Text="<%#
3.Bind("Numero_Candidatura") %>'></asp:TextBox>
4. <asp:RequiredFieldValidator ID="RequiredFieldValidator18"
5.runat="server" ControlToValidate="Numero_CandidaturaTextBox"
6.ErrorMessage="Indicar o número de candidatura" Style="position:
7.relative">*</asp:RequiredFieldValidator><br />
...
8.</InsertItemTemplate>

```

Na linha 1. indica-se que se está a trabalhar no modo de inserção. Na linha 8. encerra-se este modo. Na linha 2, identificamos o controlo a usar e na linha 3 indicamos que fazemos o binding (ligação à coluna da Base de Dados) deste controlo. A linha 4 diz respeito ao controlo RequiredFieldValidator, que é usado em todos os controlos do modo de inserção para obrigar o usuário a introduzir um valor para todos os campos de dados. Quando não se insere um número de candidatura, aparece a mensagem de erro “Indicar o número de candidatura” (linha 6).

Assim, para todos os controlos usados no modo de inserção, há que realizar esta operação de DataBinding pois só desta maneira os dados introduzidos serão enviados para a coluna pretendida na Base de Dados. Será oportuno dizer que no modo de edição, a ligação à Base de Dados é feita de modo semelhante, contudo além de somente escrever para a Base de Dados terá que ser implementada a função de ler a partir desta. Para isso usa-se o Two Way DataBinding. Este conceito será explorado mais adiante, quando se explicar a interface de edição de dados do candidato. Mas antes quero explorar o controlo GridView, uma vez que este controlo será usado no modo de consulta dos dados do candidato e em conjunto com o controlo DetailsView, permitirá implementar a interface de edição de dados do candidato.

#### 4 - O controlo GridView:

O controlo GridView pode ser visto como uma versão (muito) melhorada do controlo DataGrid introduzido na versão 1.X da plataforma .NET. Este controlo é ideal para apresentar dados em formato tabular. O controlo possui várias propriedades capazes de definir o aspecto de cada linha apresentada. Para além disso, o controlo suporta ainda as operações de ordenação, paginação e edição dos dados. Note-se que os dados apresentados pelo controlo podem ser obtidos através do mecanismo tradicional de relacionamento de dados (introduzido na versão 1.X do ASP.NET) ou através da utilização dos novos controlos *data source*.

Este controlo consegue:

- Interagir com os novos controlos *data source*, automatizando assim as operações de obtenção, modificação e eliminação de dados;
- Ordenar e paginar os dados sem necessidade de código adicional (desde que seja usado o novo mecanismo de *binding*);
- Utilizar chaves compostas (isto é, consegue manter chaves compostas por mais do que uma coluna);

PROPRIEDADES	DESCRIÇÃO
ALLOWPAGING	Propriedade de leitura e/ou escrita que indica se o controlo suporta (ou não) paginação. No caso de o controlo suportar a paginação, é necessário definir o número de elementos por página através da propriedade PageSize.
ALLOWSORTING	Indica se a <i>grid</i> deve ou não efectuar a ordenação dos dados.
AUTOGENERATECOLUMNS	Propriedade que indica se o controlo deve

	ou não gerar automaticamente as colunas apresentadas ao utilizador. Se esta propriedade possuir o valor true, então, a <i>grid</i> apresenta todas as colunas obtidas a partir da fonte de dados.
AUTOGENERATEXXXBUTTON	Utilizada para indicar se os botões associados às operações de eliminação, inserção, edição e selecção devem ou não ser gerados automaticamente.
COLUMNS	Colecção de colunas associadas à <i>grid</i> . Esta propriedade permite-nos definir explicitamente as colunas que devem ser apresentadas ao utilizador.
DATAKEYNAMES	Propriedade de leitura e/ou escrita que serve para indicar o nome da coluna que deve ser usada como campo-chave. Ao contrário do que acontecia com a <i>DataGrid</i> , esta propriedade suporta a definição de chaves compostas (ou seja, o controlo suporta chaves primárias definidas através da combinação de várias colunas). Nestes casos, os nomes das colunas devem ser separados por vírgulas.
EMPTYDATATEXT	Define o texto apresentado quando não existem elementos para serem mostrados ao utilizador.
SHOWFOOTER	Indica se a <i>grid</i> deve ou não apresentar o rodapé associado a cada coluna.

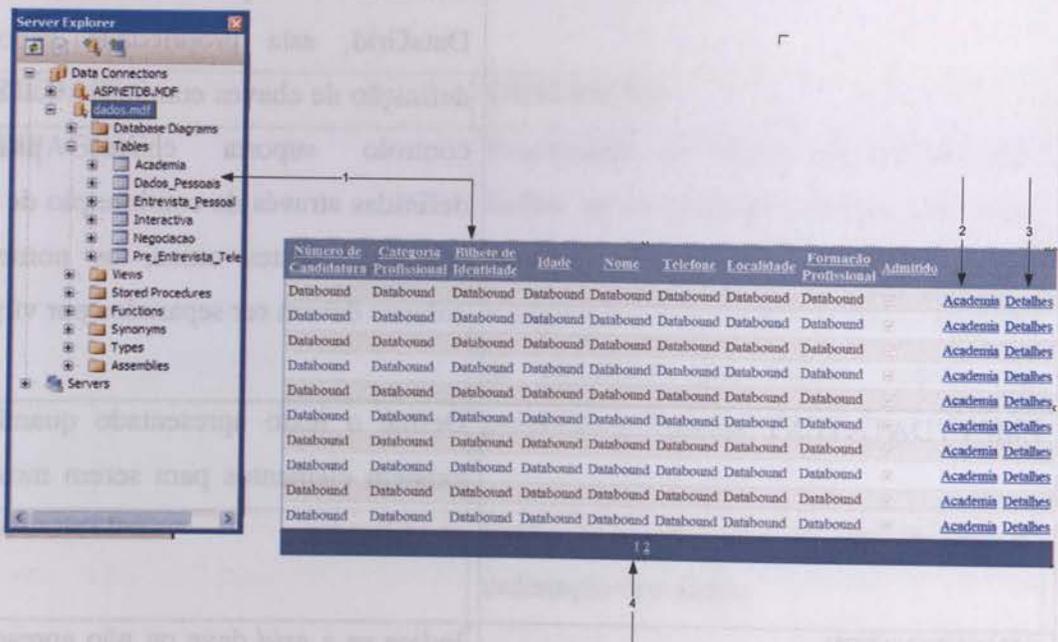
SHOWHEADER	Propriedade de leitura e/ou escrita que permite definir se o cabeçalho da tabela deve ou não ser apresentado ao utilizador.
------------	---

**Tabela 3: As propriedades do GridView**

Para entender melhor este controlo será conveniente começar a explicar como construí a interface de consulta.

### 5 - A Interface de Consulta de Dados do Candidato:

A primeira coisa a fazer é criar uma ligação à Base de Dados. Deve proceder-se de igual modo ao que já foi explicado atrás ou de um modo mais rápido para a criação de um GridView basta arrastar a tabela desejada do Server Explorer para a IDE e a ligação à Base de Dados é criada automaticamente. Pode ver-se na figura abaixo que todos os campos têm Databound.



**Figura 26: A GridView da tabela Dados\_Pessoais**

1. Arrasta-se a tabela Dados\_Pessoais para a IDE e o GridView é criado. Como se pode ver apenas os campos pelos quais se vai fazer a filtragem de dados estão presentes e nota-se que estes estão sublinhados. Isto indica que a ordenação de dados está activa. Por defeito, todos os campos da tabela são criados no GridView, contudo é possível alterar os campos apresentados a partir da propriedade Selected Fields (ver figura )

2. Hyperlink para a área da Academia, criado a partir do Hyperlink Properties (ver figura );

3. Hyperlink para os detalhes da tabela Dados\_Pessoais, ou seja, para um controlo DetailsView onde são apresentados todos os campos de informação da tabela (ver figura );

4. A paginação está activa;

A figura abaixo diz respeito ao Smart Tag do GridView e ao modo como se pode trabalhar com este controlo:

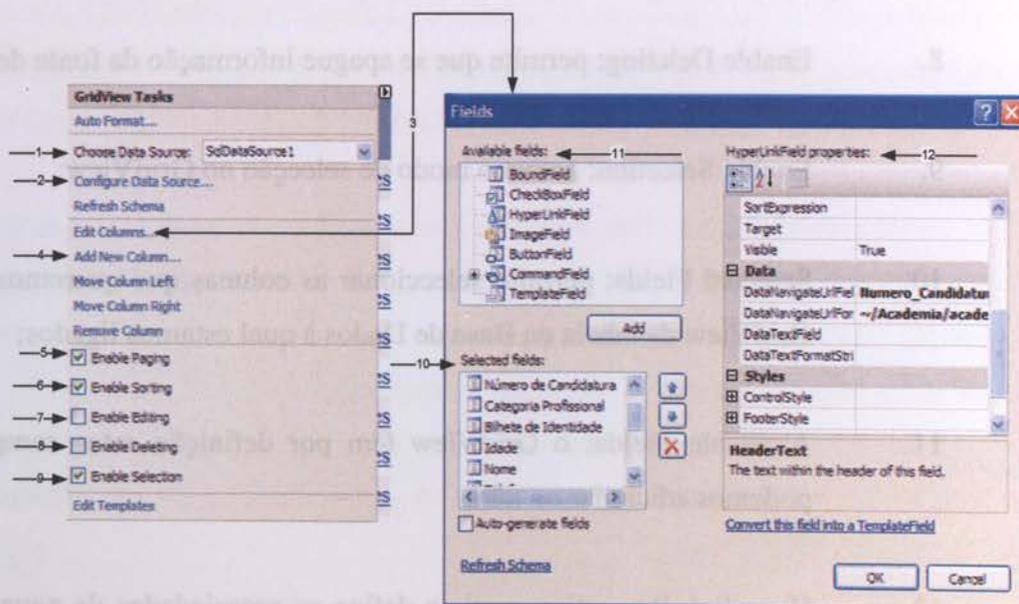


Figura 27: O Smart Tag do GridView

1. **Choose Data Source:** aqui escolhemos a fonte de dados com a qual vamos trabalhar. Este ponto já foi explicado atrás;
2. **Configure Data Source:** podemos alterar a configuração da fonte de dados;
3. **Edit Columns:** faz abrir o menu da esquerda da figura acima, e aí temos acesso a mais opções para configurar as colunas do GridView;
4. **Add New Column:** tal como o nome indica, permite adicionar uma nova coluna ao GridView;
5. **Enable Paging:** activa a paginação;
6. **Enable Sorting:** activa a ordenação dos dados;
7. **Enable Editing:** activa o modo de edição no GridView;
8. **Enable Deleting:** permite que se apague informação da fonte de dados;
9. **Enable Selection:** activa o modo de selecção no GridView;
10. **Selected Fields:** permite seleccionar as colunas que queremos ver no GridView da tabela da Base de Dados à qual estamos ligados;
11. **Available Fields:** o GridView tem por definição estes campos que podemos adicionar ou não;
12. **Hyperlink Properties:** aqui se define as propriedades de navegação a partir de informação existente no GridView;

Agora que já temos um controlo que permite a apresentação de dados, temos que filtrar a informação que queremos ver.

Pretende-se que a aplicação permita uma pesquisa a partir dos seguintes campos de informação:

- Número de Candidatura
- Localidade
- Nome
- Idade
- Telefone
- Bilhete de Identidade
- Categoria Profissional
- Formação Profissional
- Habilitações Literárias

Para que se possa fazer a filtragem de dados é necessário criar as caixas de texto onde se introduzirão os campos pretendidos e depois associá-los com o GridView onde se disponibilizará o resultado da pesquisa.

	Número de Candidaturas	Categoria Profissional	Bilhete de Identidade	Idade	Nome	Telefone	Localidade	Formação Profissional	Admitido	
Seleccionar	7	COP	23113	23	pedro	223	Ponte	XISTARCA	<input type="checkbox"/>	<a href="#">Academia Detalhes</a>
Seleccionar	34	MADM	305	23	pedro	4423	es	XISTARCA	<input type="checkbox"/>	<a href="#">Academia Detalhes</a>
Seleccionar	69	COP	54	23	sind	454	tirol	ACSM	<input type="checkbox"/>	<a href="#">Academia Detalhes</a>
Seleccionar	123	PT-GT	123	23	idisa	23	vda	ACSM	<input type="checkbox"/>	<a href="#">Academia Detalhes</a>
Seleccionar	333	PT-GT	232	23	o*eli	545	idisa	ACSM	<input type="checkbox"/>	<a href="#">Academia Detalhes</a>
Seleccionar	340	PT-GT	33	23	veruge	213	asd	XISTARCA	<input type="checkbox"/>	<a href="#">Academia Detalhes</a>
Seleccionar	348	CDP	4234	23	sdsad	3123	aida	ACSM	<input type="checkbox"/>	<a href="#">Academia Detalhes</a>
Seleccionar	553	PT-GT	332	23	ee	545	pedro	XISTARCA	<input type="checkbox"/>	<a href="#">Academia Detalhes</a>
Seleccionar	678	PT-GT	123	23	v	222	estampa	ACSM	<input type="checkbox"/>	<a href="#">Academia Detalhes</a>
Seleccionar	789	COP	2323	23	ee	23123	fril	ACSM	<input type="checkbox"/>	<a href="#">Academia Detalhes</a>

**Figura 28: Consulta e apresentação de dados**

Neste exemplo, faz-se uma pesquisa para a idade de 23 anos. Como se pode ver, apresentam-se todos os candidatos com esta idade. É possível visualizar os campos de informação referentes aos critérios de pesquisa bem como verificar se o candidato já está ou não admitido (CheckBox na coluna Admitido).

Para ver os outros campos de informação referentes à tabela Dados\_Pessoais, há que seleccionar o hyperlink Detalhes. As hiperligações entre GridView e DetailsView, serão explicadas mais à frente.

Agora explico como se associam os controlos ao GridView.

Para que se possa fazer uma filtragem de dados a partir de controlos há que configurar a fonte de dados (Configure Data Source) (passo 1). Na caixa de texto que

aparece, selecciona-se Specify a custom SQL statement or stored procedure. Ou seja, vamos criar um comando SQL (passo 2). Este vai ser:

```
SELECT Numero_Candidatura, BI, Idade, Nome, Data_de_Nascimento,
Estado_Civil, Contactos, Morada, Localidade, Código_Postal, Email,
Habilitacoes_Literarias, Curso, Formacao_Profissional, Faculdade,
Experiencia_Profissional, Conhecimentos_Informatica, Conhecimentos_Linguas,
Fontes_Recrutamento, Categoria_Profissional, Admitido FROM Dados_Pessoais
WHERE (Numero_Candidatura = @Numero_Candidatura) OR (Nome = @Nome) OR
(Localidade = @Localidade) OR (Idade = @Idade) OR (Contactos = @Telefone) OR
(BI = @BI) OR (Categoria_Profissional = @Categoria_Profissional) OR
(Habilitacoes_Literarias = @Habilitacoes_Literarias) OR (Formacao_Profissional =
@Formacao_Profissional)
```

Como já foi dito, o comando SELECT, permite a selecção de informação de uma determinada tabela. Neste exemplo, a tabela Dados\_Pessoais (FROM Dados\_Pessoais). Os campos de informação que estão entre o comando SELECT e o comando FROM são as colunas que queremos obter da Base de Dados. O comando WHERE é o responsável pela implementação das condições de pesquisa. Como se pode pesquisar por um ou mais campos, utiliza-se o operador OR. É importante salientar que o comando SQL pode ser implementado directamente na página .aspx dentro do controlo GridView. Depois de criado o comando SQL, há que associar os parâmetros de pesquisa com os controlos (passo 3). Tudo isto pode ser feito visualmente. Veja-se a figura abaixo.

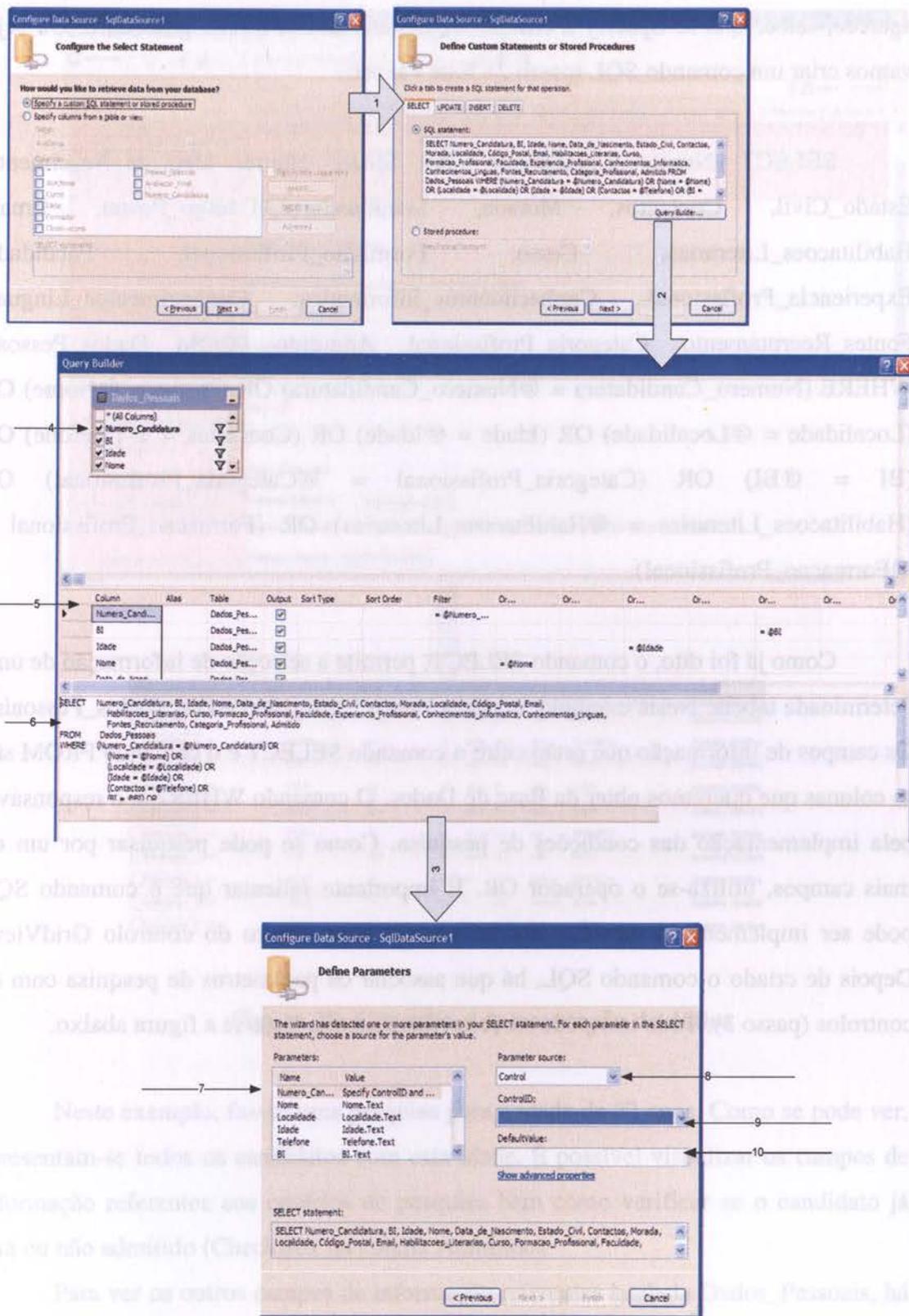


Figura 29: Associação de controlos com o GridView

Passos 1, 2 e 3 explicados atrás.

4. A tabela onde se selecciona os campos a mostrar;

5. Aqui constroi-se o filtro a aplicar (para a coluna Numero\_Candidatura, a coluna filtro, será Numero\_Candidatura = @Numero\_Candidatura);

6. Aqui aparece o comando SQL que é gerado a partir dos filtros;

7. Aqui aparecem os parâmetros pelos quais se vai fazer a filtragem;

8. Definimos a origem do valor dos parâmetros. Como se tratam de controlos Web Server escolhemos Control;

9. O controlID identifica cada Web Control presente na página. Assim, para o parâmetro Numero\_Candidatura selecciona-se Numero\_CandidaturaTextBox;

10. Por definição, o default value para campos numéricos é 1 e para campos de texto é abc;

Será interessante mostrar o que aconteceu no página .aspx em termos de código:

```
<SelectParameters>
  <asp:ControlParameter ControlID="Numero_Candidatura"
DefaultValue="1" Direction="InputOutput"
      Name="Numero_Candidatura" PropertyName="Text"
Type="Int32" />
  <asp:ControlParameter ControlID="Nome" DefaultValue="abc"
Name="Nome" PropertyName="Text" />
  <asp:ControlParameter ControlID="Localidade"
DefaultValue="abc" Name="Localidade"
      PropertyName="Text" />
  <asp:ControlParameter ControlID="Idade" DefaultValue="1"
Name="Idade" PropertyName="Text" />
  <asp:ControlParameter ControlID="Telefone"
DefaultValue="1" Name="Telefone" PropertyName="Text" />
  <asp:ControlParameter ControlID="BI" DefaultValue="1"
Name="BI" PropertyName="Text" />
  <asp:ControlParameter ControlID="cat_pro"
DefaultValue="abc" Name="Categoria_Profissional"
      PropertyName="SelectedValue" />
  <asp:ControlParameter ControlID="Hab_Lit"
DefaultValue="abc" Name="Habilitacoes_Literarias"
```

```

        PropertyName="SelectedValue" />
        <asp:ControlParameter ControlID="CheckBoxList1"
DefaultValue="abc" Name="Formacao_Profissional"
        PropertyName="SelectedValue" />
    </SelectParameters>

```

Como podemos ver, os parâmetros responsáveis pela filtragem são os que estão dentro de <SelectParameters> ... </SelectParameters>.

Quando se faz a consulta à Base de Dados apenas alguns campos de informação são apresentados. Para ver todos os campos associados a determinada tabela, usou-se um esquema GridView + DetailsView (também conhecido como MasterDetail).

Nesta implementação os hyperlinks são importantes. As colunas do tipo HyperLinkField permitem a construção de âncoras (<a>) que podem iniciar a navegação para outras páginas (com base nos valores definidos na coluna correspondente da fonte de dados). Como acontece com a maioria das colunas, é possível definir o texto associado através de uma de duas propriedades: DataTextField ou Text. A primeira obtém o texto a partir de um campo da fonte de dados; por sua vez, a segunda permite-nos definir um texto fixo (que não depende do registo actual) aplicável a todos os registos apresentados.

O destino da navegação pode ser definido à custa dos valores introduzidos nas propriedades DataNavigateUrlFormatString e DataNavigateUrlFields. A primeira propriedade define o URL que indica a página destino – este URL pode obter um ou mais valores provenientes dos campos indicados na propriedade DataNavigateUrlFields. Por exemplo, uma *querystring* do tipo *cand3.aspx?numero\_candidatura={0}*, onde o destino final é a página *cand3.aspx* e exibirá os dados referentes ao número de candidatura seleccionado no momento. Ver figura abaixo:

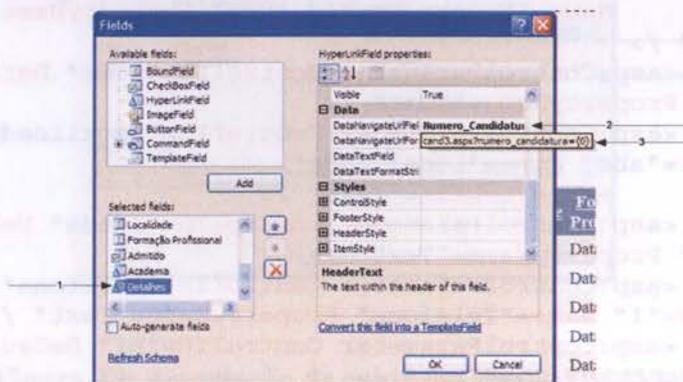
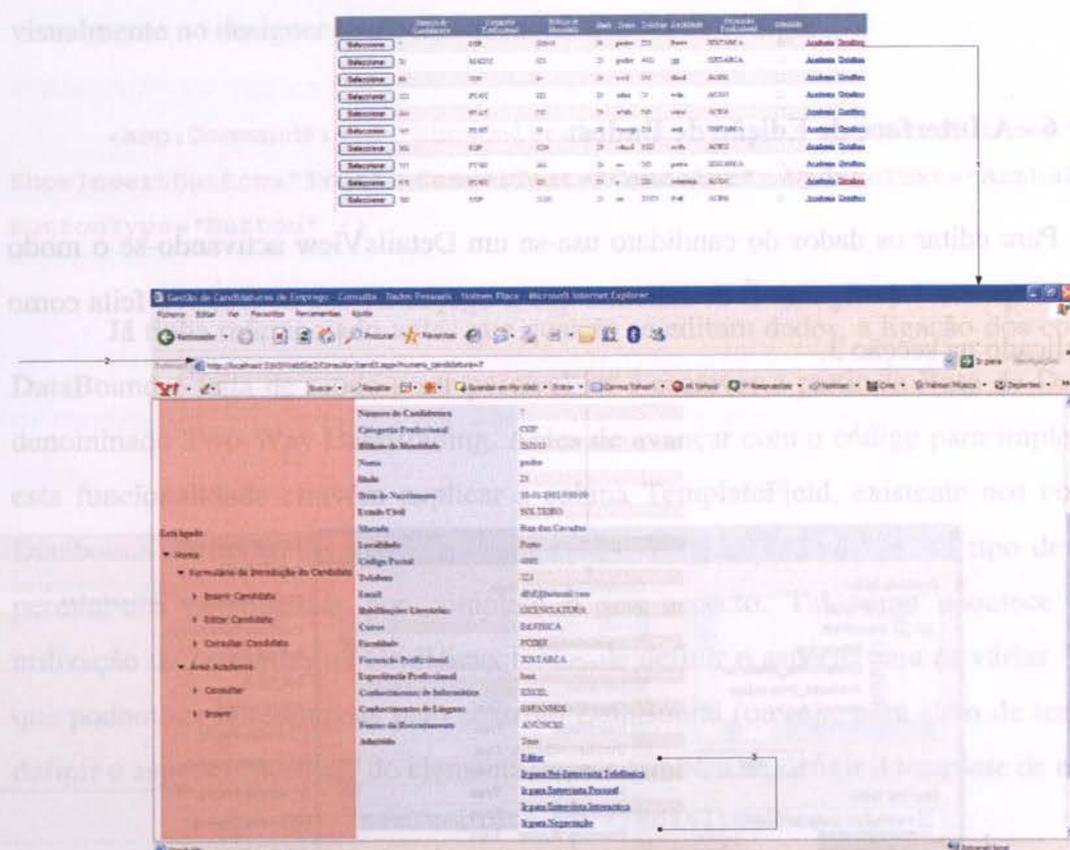


Figura 30: O HyperlinkField Properties

1. Selecciona-se o campo Detalhes;
2. Em DataNavigateUrlFiel introduz-se numero\_candidatura pois será a partir do número de candidatura do candidato que se navegará entre menus;
3. Em DataNavigateUrlFormatString introduz-se a página de destino;



**Figura 31: A página Detalhes**

Voltando ao exemplo da consulta para a idade de 23 anos, seleccionamos o primeiro candidato que aparece no GridView (ponto 1 na figura acima) e ao fazê-lo saltamos para outra página tal como era previsto. A querystring é `http://localhost:3163/WebSite2/Consultar/cand3.aspx?numero_candidatura=7` (ponto 2 da figura). O ponto 3 diz respeito às outras áreas de informação relativa ao candidato. Clicando no hyperlink somos conduzidos para a página correspondente e aí são apresentados os dados num DetailsView. Vê-se que existe um hyperlink Editar. A sua

funcionalidade tal como o nome indica é permitir a edição de qualquer campo de dados que seja preciso alterar. O acesso a esta área funciona também por querystring mas só os utilizadores com permissão de edição terão acesso a ela.

Para terminar esta secção falta apenas referir que se na pesquisa não existirem candidatos com os critérios pretendidos é apresentada a mensagem: “Não existem dados com esses critérios”. Esta pode ser configurada nas propriedades do GridView na opção EmptyDataText.

## 6 - A Interface de Edição de Dados:

Para editar os dados do candidato usa-se um DetailsView activando-se o modo de edição (ponto 1 da figura). É de salientar que a ligação à Base de Dados é feita como foi explicado na secção 3.

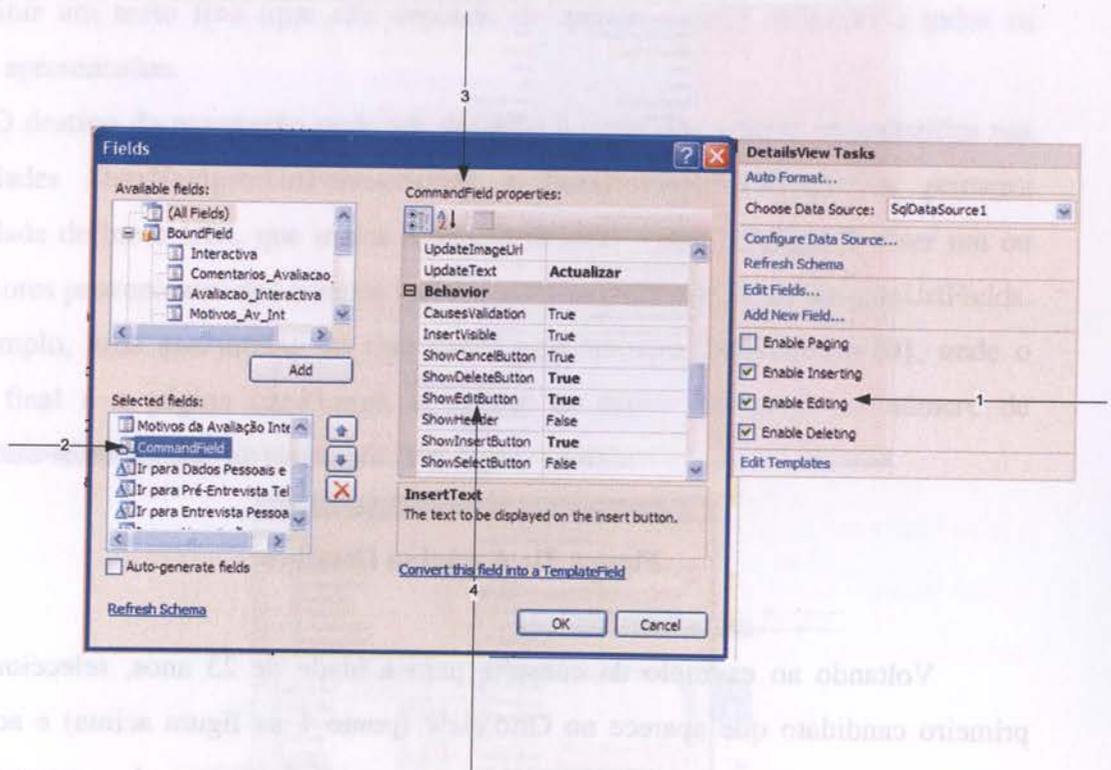


Figura 32: O Smart Tag do DetailsView

Entrando na opção Edit Fields entramos nas propriedades das colunas e assim podemos alterá-las. Como queremos inserir comandos de actualizar e cancelar, temos

que seleccionar Command Field (ponto 2) aparecendo então CommandField properties (ponto 3) e aí especificamos os botões que queremos ver e podemos alterar as suas propriedades (ponto 4). Ou seja, as colunas do tipo CommandField apresentam vários botões capazes de iniciar operações de selecção, edição e eliminação de uma linha (as operações são efectuadas sobre o controlo de *data source* associado). A geração destes botões é controlada através da propriedade ShowXXXButton exposta pela classe. Portanto, se apenas desejarmos visualizar um botão de eliminação, então, devemos atribuir o valor true à propriedade ShowDeleteButton. Pode-se ver que tudo o que se faz visualmente no designer tem consequências a nível de código:

```
<asp:CommandField ShowDeleteButton="True" ShowEditButton="True"
ShowInsertButton="True" CancelText="Cancelar" UpdateText="Actualizar"
ButtonType="Button" />
```

Já tinha referenciado antes que quando se editam dados, a ligação dos controlos DataBound é feita de modo a ser possível ler e escrever a partir da Base de Dados. O denominado Two-Way DataBinding. Antes de avançar com o código para implementar esta funcionalidade convém explicar a coluna TemplateField, existente nos controlos Databound na opção Available Fields (veja-se figura , ponto 11). Este tipo de coluna permite-nos personalizar por completo o seu aspecto. Tal como acontece com a utilização de *templates* na GridView, temos de definir o aspecto para as várias “vistas” que podem ser apresentadas pelo controlo DataBound (ou seja, para além de termos de definir o aspecto “normal” do elemento, temos também de definir o *template* de edição,

```
<EditItemTemplate> ... </EditItemTemplate>
```

e de inserção

```
<InsertItemTemplate> ... </InsertItemTemplate>).
```

A nível de código a única diferença é que dentro das linhas de código EditTemplate deve colocar-se <%=Bind(“coluna pretendida da Base de Dados”)%> e nas linhas de código do InsertItemTemplate <%=Eval(“coluna pretendida da Base de Dados”)%>. Após estas operações pode-se ler e escrever para a Base de Dados.

A seguir apresenta-se o código para a coluna Avaliacao\_Interactiva da tabela Interactiva.

```

<asp:TemplateField HeaderText="Avaliação Interactiva"
SortExpression="Avaliacao_Interactiva">
  <EditItemTemplate>
    <asp:DropDownList ID="DropDownList1"
runat="server" SelectedValue='<%# Bind("Avaliacao_Interactiva") %>'
Style="position: relative">
      <asp:ListItem Selected="True">SEM
INTERESSE</asp:ListItem>
      <asp:ListItem>ALGUM INTERESSE</asp:ListItem>
      <asp:ListItem>COM INTERESSE</asp:ListItem>
      <asp:ListItem>ADMITIDO</asp:ListItem>
    </asp:DropDownList>
  </EditItemTemplate>
  <InsertItemTemplate>
    <asp:DropDownList ID="DropDownList1"
runat="server" SelectedValue='<%# Eval("Avaliacao_Interactiva") %>'
Style="position: relative">
      <asp:ListItem Selected="True">SEM
INTERESSE</asp:ListItem>
      <asp:ListItem>ALGUM INTERESSE</asp:ListItem>
      <asp:ListItem>COM INTERESSE</asp:ListItem>
      <asp:ListItem>ADMITIDO</asp:ListItem>
    </asp:DropDownList>
  </InsertItemTemplate>
</asp:TemplateField>

```

Vejamos tudo isto a funcionar. Tome-se o exemplo da Entrevista Interactiva:

**Consulta: Entrevista Interactiva**

Número de Candidatura	346
Comentários da Avaliação de Grupo	boa interactividade
Avaliação Interactiva	ALGUM INTERESSE
Motivos da Avaliação Interactiva	estudo comportamento

[Editar](#)

[Ir para Dados Pessoais e Habilitações](#)

[Ir para Pré-Entrevista Telefónica](#)

[Ir para Entrevista Pessoal](#)

[Ir para Negociação](#)

---

**Editar: Entrevista Interactiva**

Número de Candidatura	<input type="text" value="346"/>
Comentários da Avaliação de Grupo	<input type="text" value="boa interactividade"/>
Avaliação Interactiva	<input type="text" value="ALGUM INTERESSE"/>
Motivos da Avaliação Interactiva	<input type="text" value="estudo comportamento"/>

[Ir para Dados Pessoais e Habilitações](#)

[Ir para Pré-Entrevista Telefónica](#)

[Ir para Entrevista Pessoal](#)

[Ir para Negociação](#)

2      3

Figura 33: A edição de dados, partindo do modo de consulta

Na figura ,vemos que clicando no link Editar (ponto 1), entramos na área de Edição e aí podemos alterar toda a informação. Quando se termina há que enviar o comando Update →botão Actualizar, (ponto 2), para actualizar a informação na base de dados. Se queremos cancelar alguma introdução basta clicar no botão Cancelar → comando Cancel (ponto 3). Como se pode ver está definido um conjunto de hyperlinks que permitem a navegação em modo de edição entre as diferentes áreas de dados do candidato (ponto 4).

Como é notório o número de candidatura do candidato tem um papel importante na navegação entre áreas pois este é único para cada candidato. A navegação uma vez mais é feita através de querystring:

```
<SelectParameters>
    <asp:QueryStringParameter Name="Numero_Candidatura"
QueryStringField="Numero_Candidatura"
    Type="Int32" />
</SelectParameters>
```

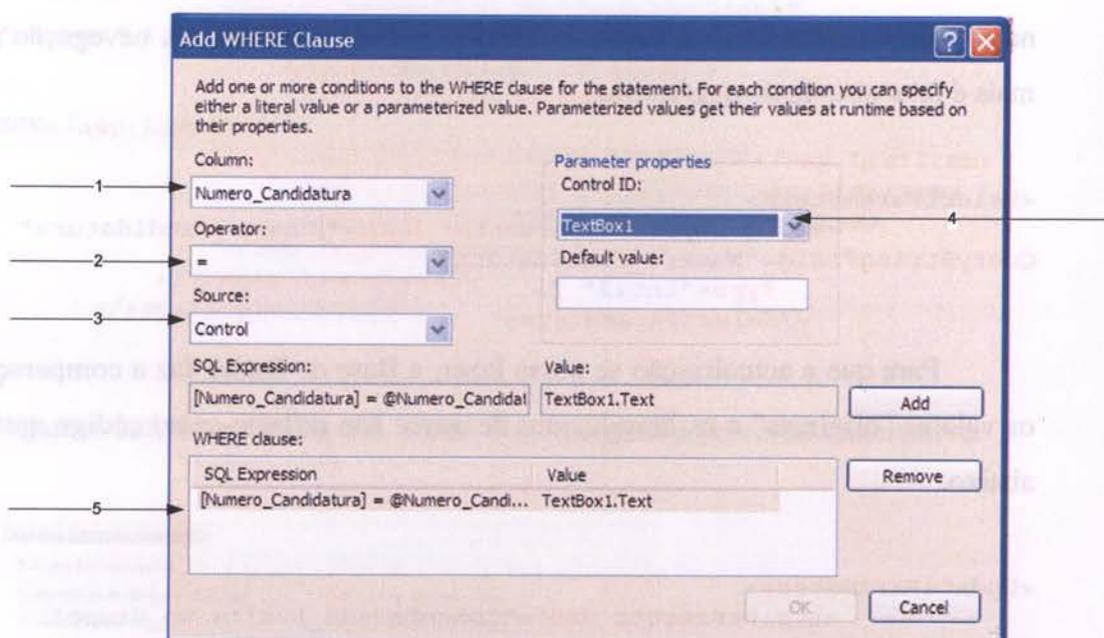
Para que a actualização se possa fazer, a Base de Dados faz a comparação entre os valores “originais” e os introduzidos de novo. Isto reflecte-se no código apresentado abaixo.

```
<UpdateParameters>
    <asp:Parameter Name="Comentarios_Avaliacao_Grupos"
Type="String" />
    <asp:Parameter Name="Avaliacao_Interactiva" Type="String"
/>
    <asp:Parameter Name="Motivos_Av_Int" Type="String" />
    <asp:Parameter Name="Numero_Candidatura" Type="Int32" />
    <asp:Parameter Name="original_Interactiva" Type="Int32" />
    <asp:Parameter Name="original_Comentarios_Avaliacao_Grupos"
Type="String" />
    <asp:Parameter Name="original_Avaliacao_Interactiva"
Type="String" />
    <asp:Parameter Name="original_Motivos_Av_Int"
Type="String" />
    <asp:Parameter Name="original_Numero_Candidatura"
Type="Int32" />
</UpdateParameters>
```

Até agora referi-me ao modo de edição partindo da visualização de dados em modo de consulta. Não obstante, é possível entrar directamente no modo de edição a partir do menu já indicado inicialmente (o TreeView) ou a partir do botão Edição de Dados do Candidato, definido na Home Page. Seja por uma maneira ou por outra, o funcionamento é igual no que diz respeito à edição

Contudo, há que realizar uma pesquisa pelo número de candidatura e então os dados referentes ao candidato são visualizados num DetailsView podendo ser editados como atrás se explicou.

Para realizar a consulta a partir do número de candidatura basta associar o controlo DataBound com o controlo. Para tal, basta ir às configurações da fonte de Dados (Configure Data Source) e no menu onde aparece a opção WHERE colocar como se indica na figura:



**Figura 34: A associação entre um controlo e o DetailsView**

1. Na categoria Column escolhe-se a coluna a usar;
2. Em operator definimos a comparação a realizar. Como queremos fazer uma pesquisa a partir de um Web Control, introduz-se a igualdade (=);
3. Em source configura-se a fonte do parâmetro a tratar. Como o número de candidatura é introduzido numa caixa de texto escolhe-se control;
4. O ControlID apresenta todas as identificações dos controlos presentes na página a tratar, escolhe-se a identificação do controlo que diz respeito ao número de candidatura, ou seja, TextBox1;

5. Após todas estas definições é gerado um comando SQL;

É um bom momento para explicar como construí a área da Academia. É uma mistura de tudo o que foi tratado até agora.

## 7 - Interface da área Academia:

Como já se viu na figura 26, existe um hyperlink para a área Academia. Esta navegação é realizada por querystring e a fonte de dados do GridView é configurada de modo apresentar os dados referentes ao número de candidatura enviado naquela.

Por outro lado, pode-se aceder a esta área a partir do menu de navegação e nesta situação a pesquisa será efectuada por número de candidatura de modo idêntico ao explicado na secção anterior.

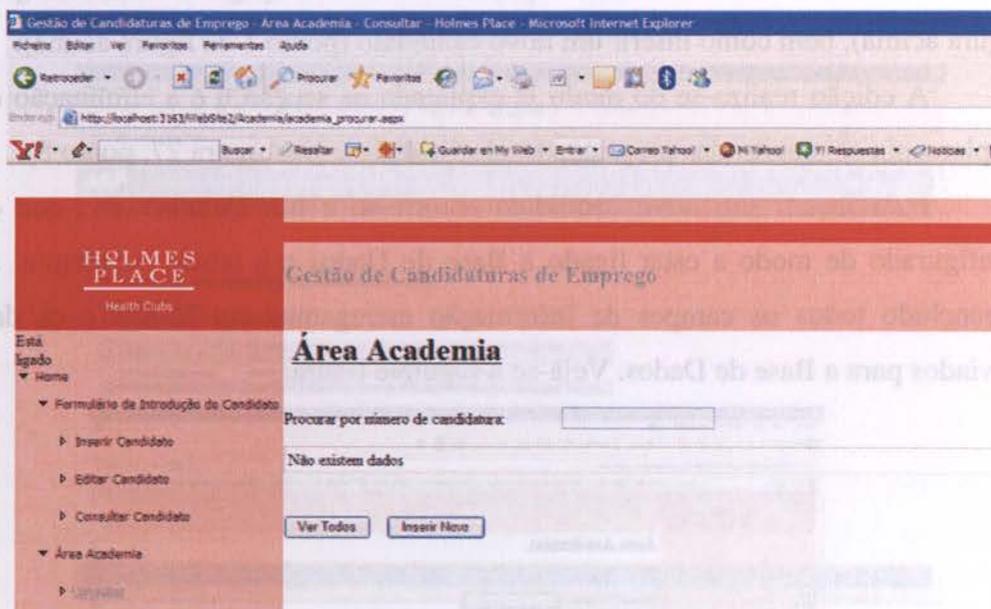
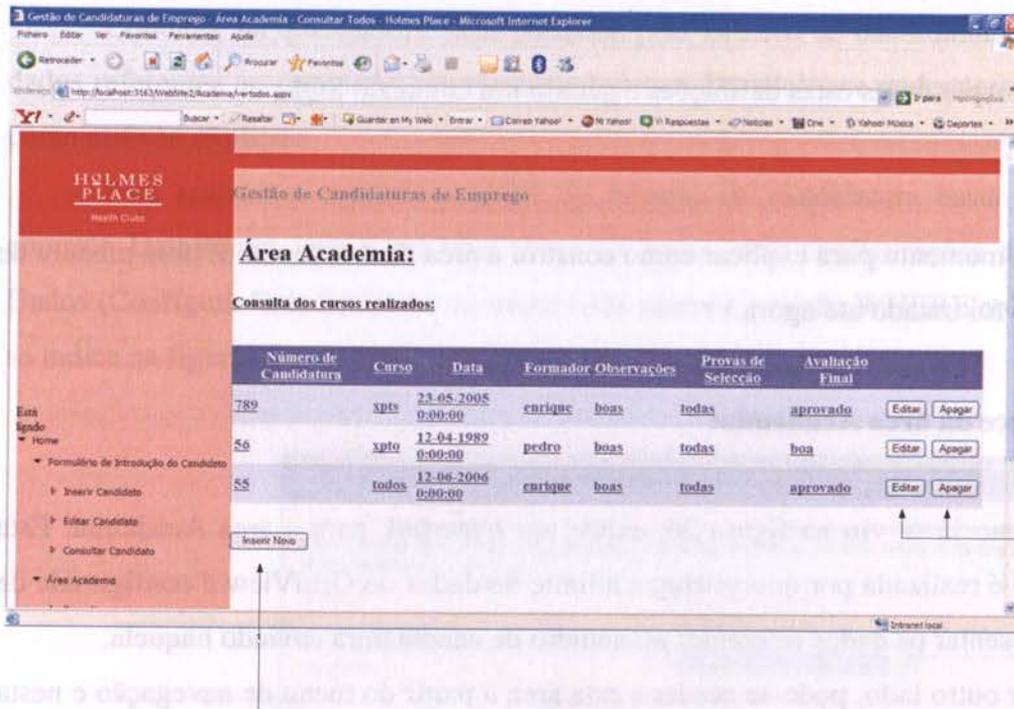


Figura 35: A área Academia, consultar

Vemos que quando não existem dados para mostrar aparece a mensagem “Não existem dados”. A partir daqui pode-se escolher ver todos os candidatos com informação nesta área ou inserir nova informação.

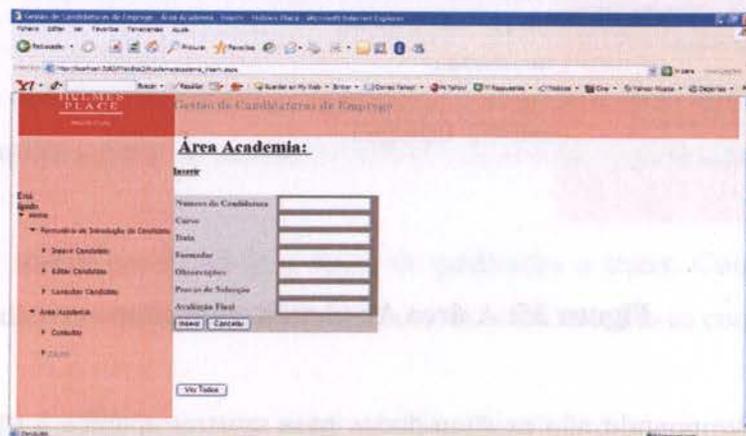


**Figura 36: A área Academia, consultar todos**

Ao efectuar a consulta podemos editar ou apagar o candidato (ponto 2 da figura acima), bem como inserir um novo candidato (ponto 1 da figura acima).

A edição realiza-se do modo já explicado na secção 6 e a eliminação de dados implementa-se a partir das propriedades do GridView (ver figura 27, ponto 8).

Para inserir um novo candidato recorre-se a um DetailsView, que deve ser configurado de modo a estar ligado à Base de Dados e à tabela Academia. Após ter preenchido todos os campos de informação carregamos em Inserir e os dados são enviados para a Base de Dados. Veja-se a seguinte figura:

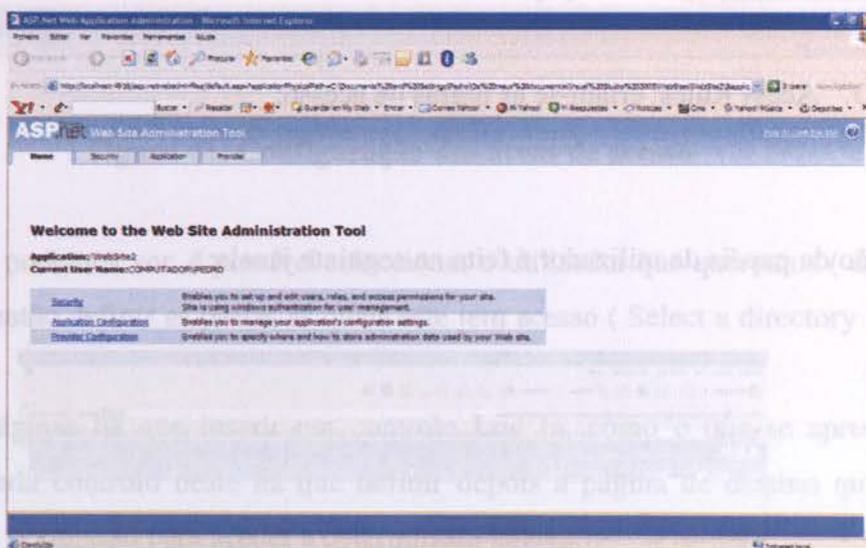


**Figura 37: Inserir dados na área Academia**

Para finalizar esta secção relativa às interfaces devo dizer que usei esquemas de cores diferentes entre modos de edição, consulta, inserção e área Academia de modo a ser mais fácil identificar a área onde se trabalha em cada momento. Além disso existe total navegabilidade entre áreas desde que o utilizador tenha permissões para entrar na página de destino.

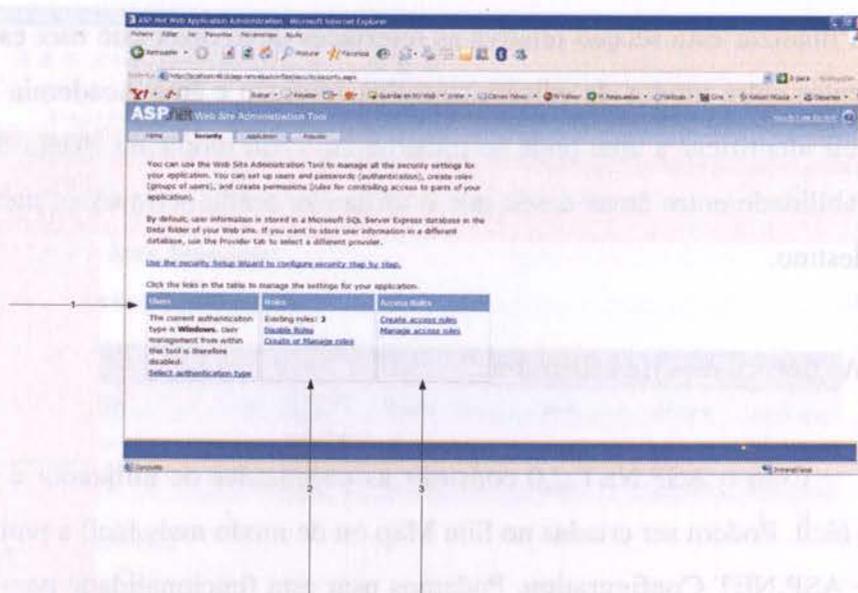
## 8 - As permissões de utilizador:

Com o ASP.NET 2.0 construir as permissões de utilizador é uma tarefa agora mais fácil. Podem ser criadas no Site Map ou de modo mais fácil a partir do menu Website → ASP.NET Configuration. Podemos usar esta funcionalidade para configurar todos as opções relacionadas com segurança do Web Site. Podemos definir utilizadores e respectivas password (autenticação), criar permissões (roles) a diferentes áreas para os utilizadores (users) .Por definição toda esta informação é guardada numa Base de Dados ASPNETDB.mdf na pasta App\_Data. Se queremos alterar esta Base de Dados é possível fazê-lo seleccionando a opção Provider. Ver figura abaixo:



**Figura 38: O ASP.NET Web Site Administration Tool**

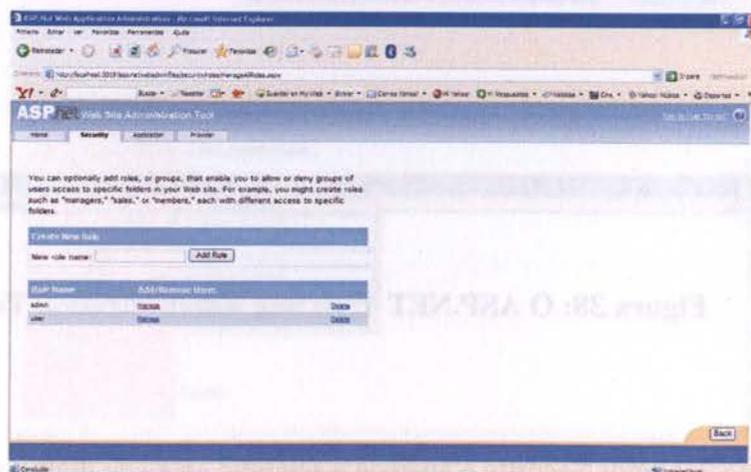
Nesta opção seleccionamos Security e aparece a seguinte caixa de diálogo:



**Figura 39: A caixa de diálogo Security**

1. Users: define-se o modo de autenticação. Está definido o modo de autenticação por Windows;
2. Roles: criam-se os papéis de utilizador;
3. Access Rules: criam-se as regras de acesso;

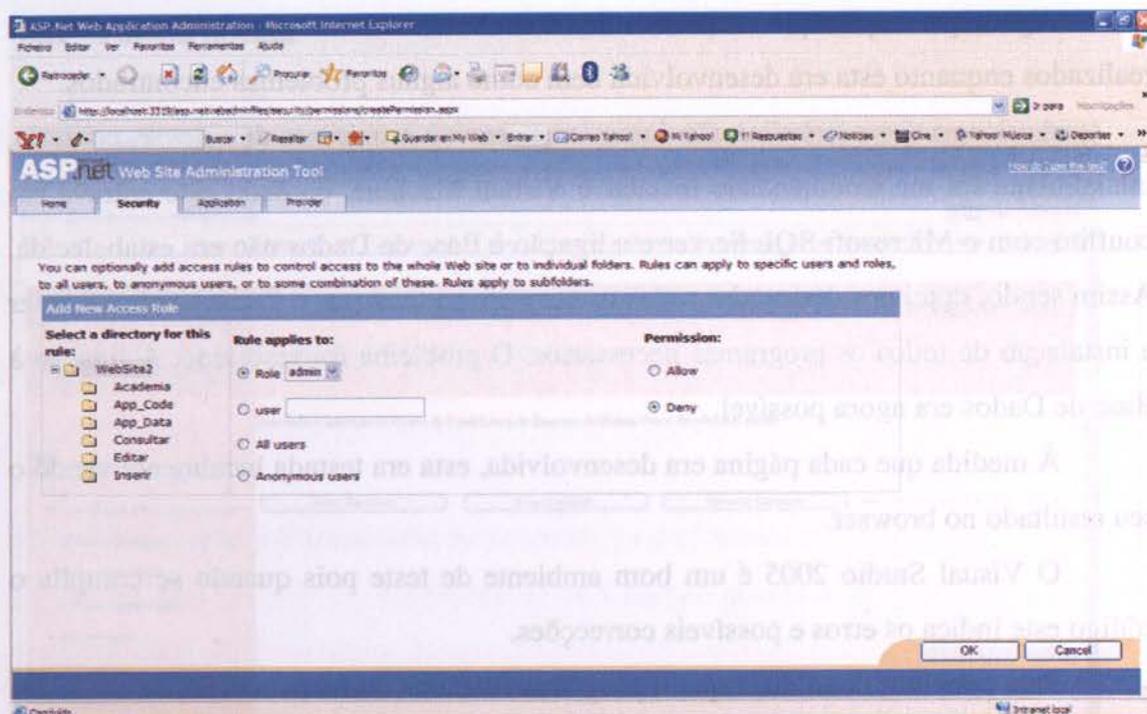
A criação de papéis de utilizador é feita na seguinte janela:



**Figura 40: A criação dos users**

Até à data estão criados dois users, o admin e o user. Estes servem para testar esta funcionalidade.

O acesso por estes users às páginas da aplicação é controlado pela aplicação referida no ponto 3 (Create Access Rules) que se apresenta a seguir:



**Figura 41: Configuração das áreas de acesso**

Como podemos ver, é possível seleccionar o utilizador que queremos ( em Rule applies to) e então definir as pastas às quais este tem acesso ( Select a directory for this rule).

Nas páginas há que inserir um controlo Log In, como o que se apresenta a seguir. Em cada controlo deste há que definir depois a página de destino quando o utilizador tem permissão para aceder a determinada página.

**Figura 42: O controlo Log In**

É de salientar que a partir deste controlo temos acesso à área ASP.NET Web Site Configuration Tool mas apenas quando estamos a construir a aplicação.

## **9 - Testar:**

Agora que a aplicação está construída é um bom momento para explicar os testes realizados enquanto esta era desenvolvida bem como alguns problemas encontrados.

Ao longo deste trabalho não foram encontrados problemas de relevar, contudo, inicialmente foi-me recomendado instalar o Virtual Machine, contudo este entrava em conflito com o Microsoft SQL Server e a ligação à Base de Dados não era estabelecida. Assim sendo, optei por desinstalar todos os programas, formatar o disco e voltar a fazer a instalação de todos os programas necessários. O problema foi resolvido. A ligação à Base de Dados era agora possível.

À medida que cada página era desenvolvida, esta era testada localmente vendo o seu resultado no browser.

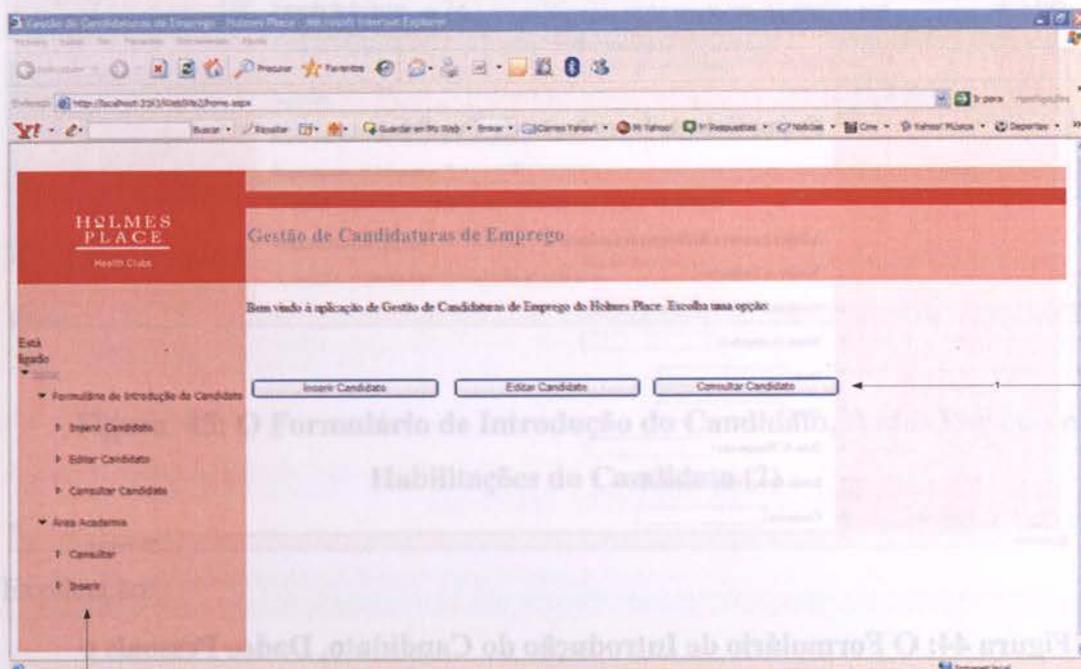
O Visual Studio 2005 é um bom ambiente de teste pois quando se compila o código este indica os erros e possíveis correcções.

Para concluir, devo dizer que o programa funciona correctamente e realiza todas as funções inicialmente propostas.

## 10 - Implementar:

Agora é o momento de explicar passo a passo o funcionamento da aplicação. Os passos seguintes servem como manual de utilização:

### 10.1 - Página inicial:



**Figura 43 : A Home Page**

### Explicação:

1. Escolhe-se a acção pretendida a partir dos botões Inserir Candidato, Editar Candidato, Consultar Candidato;
2. A partir deste menu existe acesso às áreas acima citadas bem como à área Academia;

## 10.2 - Inserção de um candidato:

Escolhe-se a opção Inserir Candidato e somos enviados para a página que se apresenta a seguir:

The image shows a screenshot of a web browser displaying a form for candidate registration. The browser window title is 'Gestão de Candidaturas de Emprego - Inserir Candidato - Holmes Place - Microsoft Internet Explorer'. The page header includes the 'HOLMES PLACE' logo and the text 'Gestão de Candidaturas de Emprego'. The main heading is 'Formulário de Introdução do Candidato'. Below this, there is a section titled 'I-Dados Pessoais e Habilitações do Candidato' containing the following fields:

- Numero de Candidatura:
- Categoria Profissional:
- Bilhete de identidade:
- Idade:
- Nome:
- Data de Nascimento:
- Estado Civil:
- Contactos:

Eight numbered arrows (1-8) point to each of these fields from the left side of the form.

**Figura 44: O Formulário de Introdução do Candidato, Dados Pessoais e Habilitações do Candidato (1)**

Explicação:

1. Insere-se o número de candidatura, único para candidato. Deve ser um número inteiro;
2. Escolhe-se a Categoria Profissional a partir das opções apresentadas;
3. Insere-se o Bilhete de Identidade do candidato;
4. A idade do candidato;
5. O nome do candidato;
6. A data de nascimento do candidato no formano dd-mm-yyyy ( dia – mês – ano);
7. O estado civil do candidato;
8. O telefone do candidato;

**Figura 45: O Formulário de Introdução do Candidato, Dados Pessoais e Habilitações do Candidato (2)**

Explicação:

9. Insere-se a morada do candidato;
10. A localidade do candidato;
11. O código postal;
12. O email do candidato;
13. Habilitações literárias do candidato;
14. O curso do candidato;
15. A formação profissional. Pode-se escolher mais do que uma opção;
16. A faculdade onde estudou;
17. A experiência profissional;
18. Conhecimentos de informática, pode-se escolher mais do que uma opção;
19. Conhecimentos de línguas, pode-se escolher mais do que uma opção;

20 → Fontes de Recrutamento:

INTERNET  ANÚNCIO  ACADEMIA  ESPONTÂNEA  REFERÊNCIA  OUTROS

Enviar Dados

21 ↑

**Figura 46: O Formulário de Introdução do Candidato, Dados Pessoais e Habilitações do Candidato (3)**

Explicação:

20. Escolhe-se a fonte de recrutamento, pode-se escolher mais do que uma opção;
21. Após ter preenchido todos os campos activa-se este comando. Os dados são enviados;

Passa-se para o formulário da Pré-Entrevista Telefónica:

Gestão de Candidaturas de Emprego - Inserir Candidato - Holmes Place - Microsoft Internet Explorer

http://localhost:3163/Website2/InserirCand.asp

### Formulário de Introdução do Candidato

2-Pré-Entrevista Telefónica

Está ligado

▼ Home

- ▼ Formulário de Introdução do Candidato
  - ▼ Inserir Candidato
  - ▼ Editar Candidato
  - ▼ Consultar Candidato
- ▼ Área Academia
  - ▼ Consultar
  - ▼ Inserir

Numero de Candidatura:  ← 22

Já Realizada: Escolha uma opção ▼ ← 23

Situação Profissional Actual:  ← 24

Remuneração Actual Última:  ← 25

Razões de Candidatura:  ← 26

Clube de Preferência: Escolha uma opção ▼ ← 27

Disponibilidade de Admissão: Escolha uma opção ▼ ← 28

Disponibilidade para Turnos: Escolha uma opção ▼ ← 29

Avaliação da Pré Entrevista: Escolha uma opção ▼ ← 30

Enviar Dados

Concluído

Internet local

**Figura 47 : O formulário Pré-Entrevista Telefónica**

Explicação:

22. Introduzir o número de candidatura (deve ser o mesmo do formulário anterior);
23. Já Realizada, sim ou não;
24. Indicamos a situação profissional actual;
25. A remuneração actual ou a última;
26. As razões de candidatura ao lugar;
27. O clube de preferência para trabalhar;
28. A disponibilidade de admissão;
29. A disponibilidade para trabalhar por turnos;
30. A avaliação da pré-entrevista;

Quando a Pré-Entrevista não foi realizada, nas DropDownList deve seleccionar-se a opção SEM INFORMAÇÃO.

Após os campos preenchidos, clicar em Enviar Dados.

Passa-se para o formulário Entrevista Pessoal:

**Entrevista Pessoal**

Número de Candidatura:  ← 31

Aperzentação e Imagem:  ← 32

Motivos da candidatura:  ← 33

Pretensões Salariais:  ← 34

Comunicação:  ← 35

Falta pouco; Exprime-se com alguma dificuldade  
 Tenta exprime-se com clareza  
 Flutuante e capacidade de expressão médias  
 Falta correctamente e de forma concisa  
 Excelente capacidade de expressão

Experiência:  ← 36

Não existe qualquer relação entre experiência anterior e os requisitos para esta função  
 Pouca experiência em áreas semelhantes  
 Trabalho anterior muito relacionado com a função a desempenhar  
 Experiência muito boa e adequada  
 Experiência anterior excelente

Está ligado

▼ Início

▼ Formulário de Introdução do Candidato

▶ Inserir Candidato

▶ Editar Candidato

▶ Consultar Candidato

Figura 48: O formulário Entrevista Pessoal (1)

Explicação:

31. Introduzir o número de candidatura (deve ser o mesmo do formulário anterior);
32. Apresentação e imagem do candidato;
33. Motivos da candidatura do candidato;
34. Pretensões salariais;
35. Escolher a opção que mais se aproxime sobre o aspecto Comunicação;
36. Escolher a opção que mais se aproxime sobre o aspecto Experiência;

**Figura 49: O formulário Entrevista Pessoal (2)**

37. Escolher a opção que mais se aproxime sobre o aspecto Interesses;
  38. Escolher a opção que mais se aproxime sobre o aspecto Motivação;
  39. Seleccionar a avaliação referente à entrevista pessoal;
  40. Introduzir os motivos da entrevista pessoal;
  41. A data da entrevista (no formato dd-mm-yyyy);
  42. O nome do entrevistador;
  43. O clube onde foi realizada a entrevista;
- Após todos os campos preenchidos, fazer Enviar Dados.

Passa-se para o formulário Entrevista Interactiva:

**Figura 50: O formulário Entrevista Interactiva**

Explicação:

- 44. Introduzir o número de candidatura (deve ser o mesmo do formulário anterior);
- 45. Comentários da avaliação de grupo;
- 46. Avaliação da entrevista interactiva;
- 47. Introduzir os motivos da avaliação;

Após todos os campos preenchidos, fazer Enviar Dados.

Passa-se para o formulário Negociação:

**Figura 51: O formulário Negociação**

Explicação:

48. Introduzir o número de candidatura (deve ser o mesmo do formulário anterior);
49. Selecciona-se o horário de trabalho pretendido;
50. O tipo de contrato;
51. A data prevista de admissão;

Após todos os campos preenchidos, fazer Enviar Dados. Se tudo está bem, os dados são enviados para a Base de Dados e deve aparecer:

**Figura 52: Formulário enviado com sucesso.**

Clicando em Terminar, volta-se para a página inicial.

### 10.3 - A área Consulta:

Ao entrar na área Consulta podemos pesquisar pelos seguintes parâmetros:

1. Número de Candidatura
2. Localidade
3. Nome
4. Idade
5. Telefone
6. Bilhete de Identidade
7. Categoria Profissional
8. Formação Profissional
9. Habilitações Literárias

The screenshot displays a web browser window with the title 'Gestão de Candidaturas de Emprego - Consulta - Holmes Place - Microsoft Internet Explorer'. The address bar shows 'http://localhost:1045/Default2.aspx?ConsultarConsulta.aspx'. The main content area is titled 'Gestão de Candidaturas de Emprego' and 'Consulta de Candidaturas:'. Below this, it says 'Dados Pessoais e Habilitações:' and 'Preencher um ou mais dos campos seguintes para efectuar a consulta:'. The form contains the following fields with arrows pointing to them: 1. Número de Candidatura (text input), 2. Localidade (text input), 3. Nome (text input), 4. Idade (text input), 5. Telefone (text input), 6. Bilhete de Identidade (text input), 7. Categoria Profissional (dropdown menu with 'Escolha uma opção'), 8. Formação Profissional (checkboxes for AFFA, NISTARCA, ACSM, MANZ, CEF, HP, CPR, and OUTROS), and 9. Habilitações Literárias (dropdown menu with 'Escolha uma opção'). On the left side, there is a sidebar with a menu titled 'Está ligado' and a sub-menu 'Formulário de Introdução de Candidato' with links for 'Inserir Candidato', 'Editar Candidato', and 'Consultar Candidato'. Below that is 'Área Académica' with a link for 'Consultar'. The browser's taskbar shows several open applications like 'Google', 'Internet Explorer', and 'Outlook'.

Figura 53: A área Consulta

Após introduzir um critério de pesquisa aparecem os resultados e aí podemos entrar nos detalhes de informação do candidato bem como entrar na área Academia. Apresenta-se o exemplo para a pesquisa por idade = 23 anos.

Nome:

Idade: 23

Telefone:

Bilhete de Identidade:

Categoria Profissional:

Formação Profissional:  AFPA  XISTARCA  ACSM  MANZ  CEF  HP  CPE  OUTROS

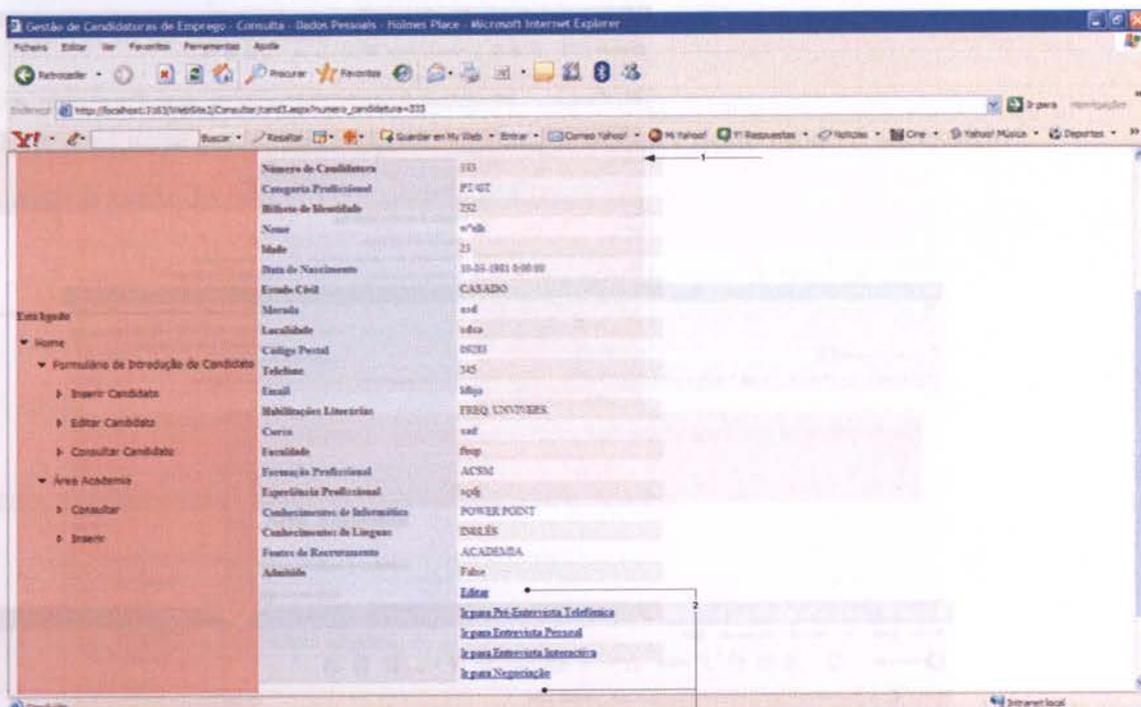
Habilitações Literárias:

	Número de Candidatura	Categoria Profissional	Número de Identidade	Idade	Sexo	Telefone	Localidade	Formação Profissional	Método
<input type="button" value="Selecionar"/>	7	COP	33113	23	predo	233	Ponte	XISTARCA	<input type="checkbox"/> Academia <a href="#">Detalhes</a>
<input type="button" value="Selecionar"/>	34	SEADAC	823	23	predo	4433	EG	XISTARCA	<input type="checkbox"/> Academia <a href="#">Detalhes</a>
<input type="button" value="Selecionar"/>	69	COP	34	23	vrrd	454	dasd	ACSM	<input type="checkbox"/> Academia <a href="#">Detalhes</a>
<input type="button" value="Selecionar"/>	123	PT-OT	123	23	rdha	23	nda	ACSM	<input type="checkbox"/> Academia <a href="#">Detalhes</a>
<input type="button" value="Selecionar"/>	333	PT-OT	232	23	w*dk	545	sdas	ACSM	<input type="checkbox"/> Academia <a href="#">Detalhes</a>
<input type="button" value="Selecionar"/>	345	PT-OT	33	23	wwqr	213	sd4	XISTARCA	<input type="checkbox"/> Academia <a href="#">Detalhes</a>

**Figura 54: Os resultados de uma pesquisa**

Como se pode ver para o exemplo de pesquisa por idade, são apresentados os dados, indicando se o candidato está admitido ou não e aí há que seleccionar o candidato pretendido através do controlo Seleccionar (ponto 2) e depois para ver em detalhe a informação do candidato clicar em Detalhes ou se se pretende entrar na área Academia clicar em Academia (ponto 3).

Ao clicar-se em Detalhes vemos a informação do candidato, para o número de candidatura seleccionado (ponto 1 da figura abaixo) ao pormenor e aí podemos entrar em modo de edição ou entrar nos outros formulários de informação do candidato (ponto 2 da figura abaixo). Nesses formulários (os que foram preenchidos aquando da inserção do candidato) existem opções de navegação tais como esta que permitem navegar entre formulários ou então editar a informação neles presente.

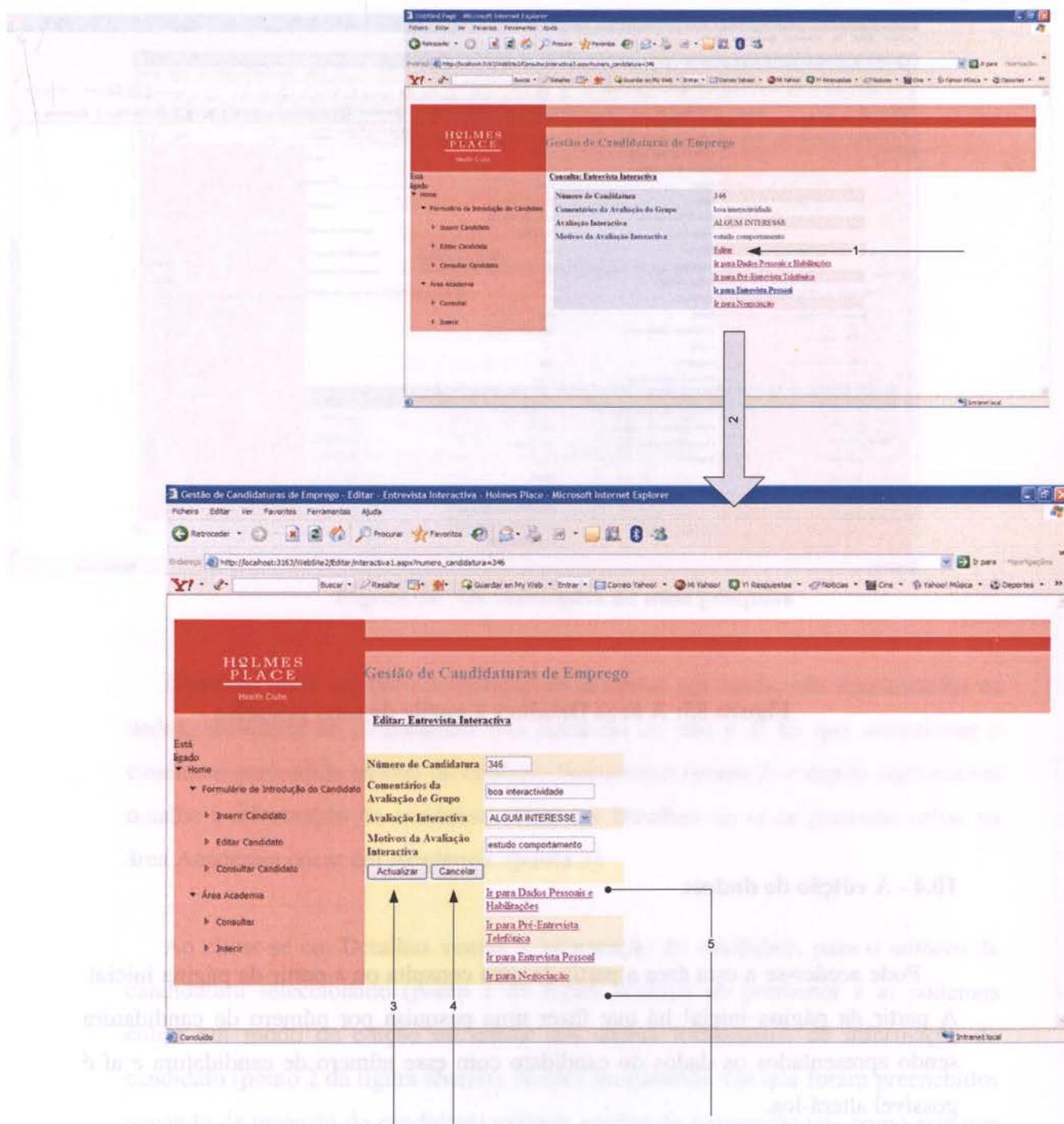


**Figura 55: A área Detalhes a partir de uma pesquisa**

#### 10.4 - A edição de dados:

Pode aceder-se a esta área a partir de uma consulta ou a partir da página inicial. A partir da página inicial há que fazer uma pesquisa por número de candidatura sendo apresentados os dados do candidato com esse número de candidatura e aí é possível alterá-los.

A partir de uma consulta clica-se no link Editar (já visto atrás) e somos enviados para a página correspondente ao formulário com o qual se trabalha no momento mas em modo edição. Tome-se como exemplo o formulário Entrevista Interactiva:

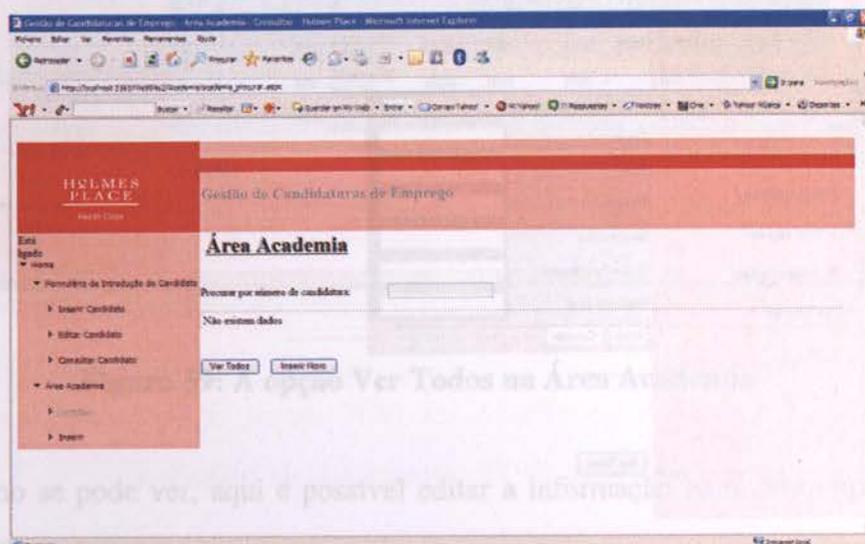


**Figura 56: O modo de Edição**

Como já tinha dito, a partir de uma consulta de dados, neste caso, do formulário Entrevista Interactiva, passamos para o modo de edição a partir do link Editar (ponto 1 da figura) e aí podemos alterar a informação presente a para actualizar há que clicar em Actualizar (ponto 3 da figura) ou para anular a operação em Cancelar (ponto 4 da figura). Os links que vemos (ponto 5 da figura) servem para entrar no modo de edição desses formulários.

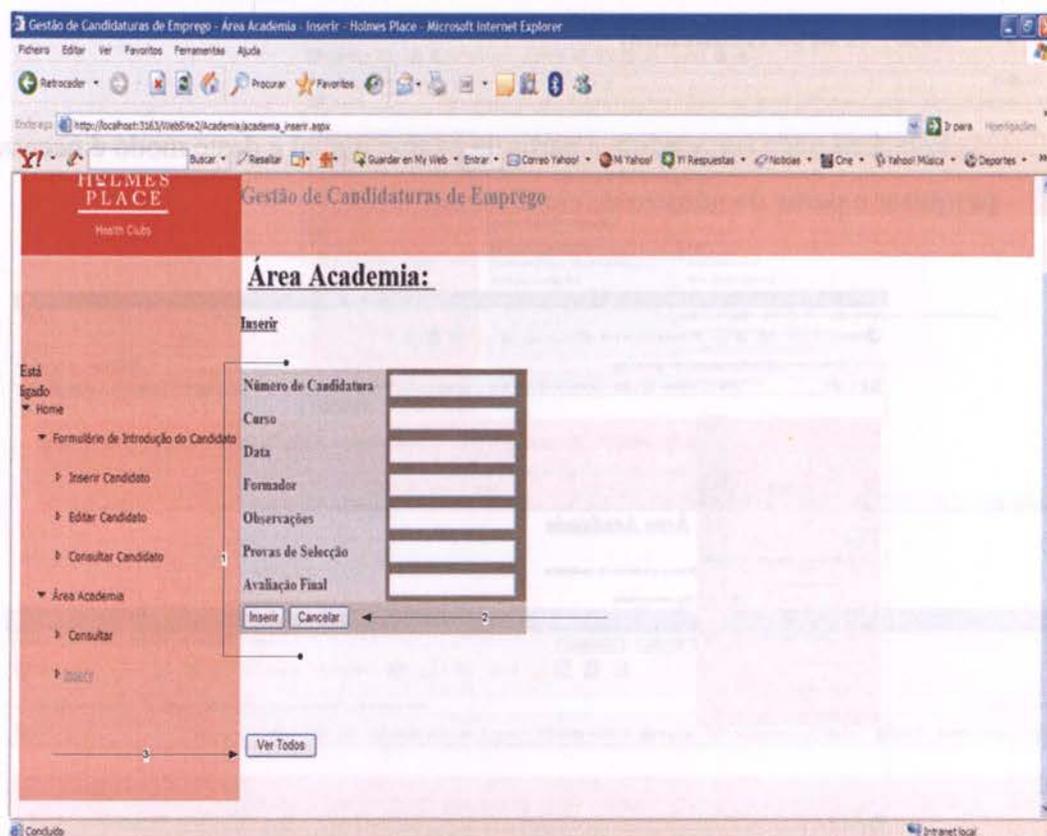
## 10.5 - A área Academia:

Esta área pode ser acessada a partir da página inicial e deste modo é necessário pesquisar a partir do número de candidatura.



**Figura 57: A área Academia**

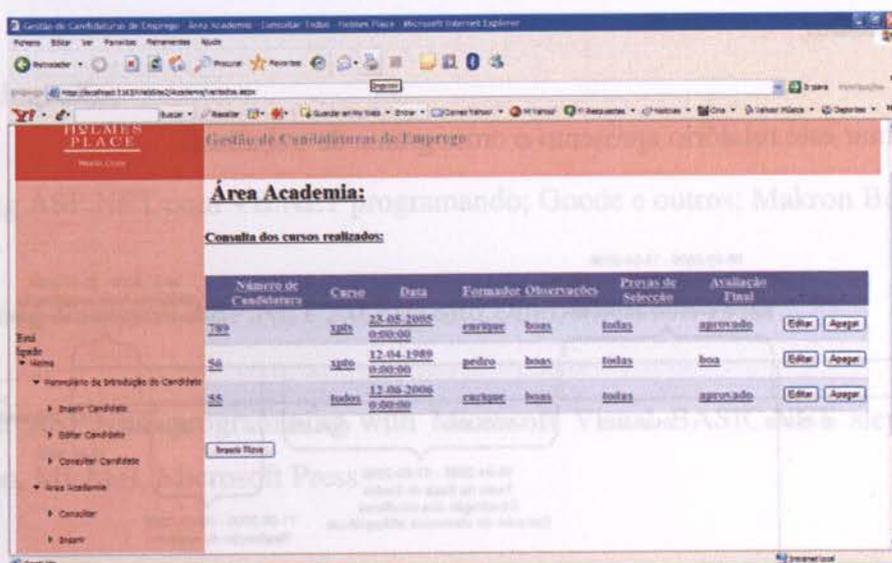
Após introduzir um número de candidatura, são apresentados os dados relacionados com aquele. É possível inserir nova informação ( ver figura ) ou então ver todos os candidatos que têm informação nesta área. (ver figura ).



**Figura 58: Inserir dados na área Academia**

#### Explicação:

1. Aqui introduzem-se os dados relevantes para o candidato, ter em atenção que o número de candidatura deve corresponder ao candidato em causa;
2. Após a introdução dos dados, fazer Inserir para enviar os dados para a Base de Dados ou Cancelar para anular a operação;
3. Para ver todos os candidatos com informação na área Academia;

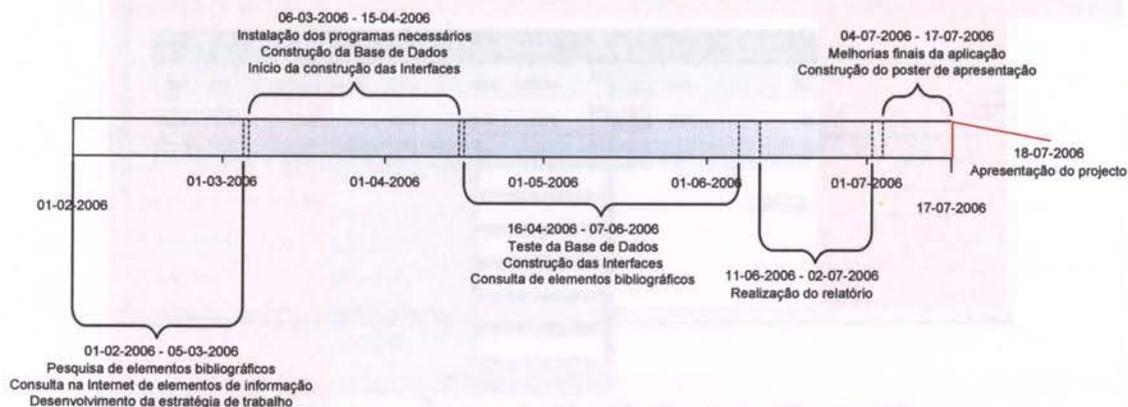


**Figura 59: A opção Ver Todos na Área Academia**

Como se pode ver, aqui é possível editar a informação bem como apagá-la da Base de Dados.

## 11 - Conclusão:

Para terminar este relatório apresento o cronograma de trabalhos:



**Figura 60: O cronograma de trabalhos**

É de relevar que apesar de estar em pleno funcionamento e realizar tudo o que era pretendido, a aplicação vai ser sujeita a algumas melhorias.

Gostava de concluir indicando que pude aprender bastante com este trabalho e dar-me conta de todas as potencialidades desta nova geração do Visual Studio que em conjunto com o ASP.NET 2.0 permite criar aplicações robustas, eficazes e de alguma complexidade.

- Após a introdução dos dados, fazer testes para enviar os dados para a Base de Dados no Consolador para avaliar a aplicação;
- Para ver todos os resultados mais informações na área Académica;

**12 - Bibliografia:**

Beginning ASP.NET com VB.NET programando; Goode e outros; Makron Books;

Introducing Microsoft ASP.NET 2.0;Esposito,Dino; Microsoft Press

Microsoft ASP.NET programming with Microsoft Visual BASIC.NET step by step;  
Halverson, Michael; Microsoft Press

Building Web solutions with ASP.NET and ADO.NET;Esposito,Dino Microsoft Press

Introducing Microsoft SQL Server 2005 for developers; DeBetta, Peter; Microsoft  
Press

Visual Basic.NET Complete; vários autores; Sybex

SQL; Damas, Luís; FCA Editoras

**Principais sites de pesquisa Internet:**

[www.asp.net](http://www.asp.net)

[www.msdn.microsoft.com](http://www.msdn.microsoft.com)





FACULDADE DE ENGENHARIA  
UNIVERSIDADE DO PORTO

BIBLIOTECA



000105171