



## Ficha de Trabalho N.º 6

### ASP.NET - WebMatrix

1. Iniciar o WebMatrix
2. Abrir o web site WebPagesMovie.
3. Clicar em Files para abrir o ambiente de trabalho Files.
4. Criar uma nova página com o nome AdFilme.cshtml.
5. Substituir o código existente pelo seguinte:

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>Adicionar um filme</title>
</head>
<body>
  <h1>Adicionar um filme</h1>
  <form method="post">
    <fieldset>
      <legend>Informação do filme</legend></legend>
      <p><label for="title">Título:</label>
        <input type="text" name="title" value="@Request.Form["title"]" />
      </p>

      <p><label for="genre">Género:</label>
        <input type="text" name="genre" value="@Request.Form["genre"]" />
      </p>

      <p><label for="year">Ano:</label>
        <input type="text" name="year" value="@Request.Form["year"]" />
      </p>

      <p><input type="submit" name="buttonSubmit" value="Adiciona Filme" /></p>
    </fieldset>
  </form>
</body>
</html>

```

- a. O código acima é código HTML para criar um formulário.
- b. São usados elementos <input> para criar as caixas de texto e o botão Adiciona Filme. As legendas para as caixas de texto são criadas usando elementos <label>. Os elementos <fieldset> e <legend> são usados para criar uma caixa à volta do formulário.
- c. Reparar no atributo method do form: *post*. Usamos este atributo porque vamos fazer alterações na base de dados, adicionando novos registos.

- d. O atributo value das caixas de texto, contém código do tipo Request.Form["title"]. Isto faz-se para preservar o valor que o utilizador digitou, após submeter uma página.
6. A seguir temos os passos necessários para processar o formulário:
- Verificar se a página foi submetida. Queremos que o código seja executado apenas se o utilizador clica no botão *submit*, e não da primeira vez que a página é requisitada.
  - Obter os valores que o utilizador digitou nas caixas de texto. Obtemos estes valores a partir de Request.Form.
  - Inserir os valores como um novo registo na tabela Filmes da base de dados.
7. No início do ficheiro, adicionar o seguinte código:

```
@{  
var title = "";  
var genre = "";  
var year = "";  
  
if(IsPost){  
    title = Request.Form["title"];  
    genre = Request.Form["genre"];  
    year = Request.Form["year"];  
}  
}
```

- As primeiras linhas criam as variáveis *title*, *genre* e *year* que vão guardar os valores inseridos nas caixas de texto.
  - Os nomes das variáveis são arbitrários. Não precisam de ter o mesmo nome dos elementos `<input>` do formulário.
8. Adicionar o seguinte código antes de fechar a chaveta do bloco `if`:

```
var db = Database.Open("WebPagesMovie");  
var insertCommand = "INSERT INTO Filmes (Título, Género, Ano) VALUES(@0, @1, @2)";  
db.Execute(insertCommand, title, genre, year);  
Response.Redirect("~/Filmes");
```

- Após abrir a base de dados, construímos o comando SQL que vai permitir inserir um novo registo na tabela Filmes.
- O comando SQL é: `INSERT INTO tabela (coluna1,coluna2, coluna3,...) VALUES (valor 1, valor2, valor3, ...)`.
- Título, Género e Ano, são campos da tabela Filmes.
- `@0`, `@1` e `@2` são os parâmetros através dos quais vamos passar os valores das caixas de texto para o comando SQL.

e. A linha

```
db.Execute(insertCommand, title, genre, year);
```

executa o commando especificado na variável insertCommand, substituindo:

@0 pelo valor da variável title

@1 pelo valor da variável genre

@2 pelo valor da variável year.

f. A linha

```
Response.redirect("~/Filmes");
```

Redireciona o browser para a página Filmes.cshtml de forma a podermos ver o filme que acabamos de inserir na tabela. O operador ~significa a raiz do sítio Web.

9. Guardar as alterações efetuadas e executar a página no browser.

10. Inserir o seguinte filmes: "Citizen Kane", "Drama" e "1941".

11. Voltar à página Adicionar Filme e inserir o filme "Singin' in the Rain".

12. Verificar que o filme foi inserido, apesar de faltarem dados: o género e o ano.

13. Nos próximos passos vamos validar os dados que o utilizador digita por forma a não aceitar campos vazios.

14. No início do bloco de código, logo após @{, digitar o seguinte código:

```
Validation.RequireField("title", "Tem de digitar um título");  
Validation.RequireField("genre", "Género.campo necessário");  
Validation.RequireField("year", "Não digitou o ano");
```

a. Utiliza-se uma chamada Validation.RequireField por cada elemento <input> que queremos validar.

15. Alterar a linha if(IsPost) por forma a validar os dados:

```
if(IsPost && Validation.IsValid() ){
```

16. Se houver algum erro em qualquer um dos campos que queremos validar, a condição Validation.IsValid() retorna o valor Falso e os dados não serão inseridos na tabela da base de dados.

17. Para ajudar o utilizador, podemos mostrar mensagens de erro: em cada elemento <input> que estamos a validar utilizamos o método Html.ValidationMessage. Podemos mostrar mensagens individuais, um sumário das mensagens ou ambos.

18. Alterar o código HTML de forma a incluir a validação para cada um dos elementos <input>:

```
<p><label for="title">Título:</label>  
<input type="text" name="title" value="@Request.Form["title"]" />  
@Html.ValidationMessage("title")  
</p>
```

```

<p><label for="genre">Género:</label>
  <input type="text" name="genre" value="@Request.Form["genre"]" />
  @Html.ValidationMessage("genre")
</p>

<p><label for="year">Ano:</label>
  <input type="text" name="year" value="@Request.Form["year"]" />
  @Html.ValidationMessage("year")
</p>

```

19. A seguir à linha <h1>Adicionar um filme </h1> digitar o seguinte código:

```
@Html.ValidationSummary()
```

20. Adicionar o seguinte código dentro da secção <head>, para formatar as mensagens de erro.

```

<style type="text/css">
  .field-validation-error {
    font-weight:bold;
    color:red;
    background-color:yellow;
  }
  .validation-summary-errors{
    border:2px dashed red;
    color:red;
    background-color:yellow;
    font-weight:bold;
    margin:12px;
  }
</style>

```

- A classe CSS `.field-validation-error` formata as mensagens individuais dos elementos <input>.
- A classe CSS `.validation-summary-errors` formata o sumário de erros mostrado no início da página.

21. Falta adicionar um link à página Filmes para esta nova página:

- Abrir a página `Filmes.cshtml`.
- Antes da linha </body> adicionar o seguinte código:

```

<p>
  <a href="~/AdFilme">Adicionar filme</a>
</p>

```