

# Programação Visual

## Trabalho de Laboratório nº 7

<b>Objetivo</b>	MVC – Introdução. Aplicações básicas com <b>WebAPI</b> .
<b>Programa</b>	Pretende-se continuar o desenvolvimento da aplicação <b>EsteCar</b> com o desenvolvimento de uma <b>API</b> que disponibilize as funcionalidades da empresa (visualizar carros, alugar...) através de serviços <b>REST</b> , que são posteriormente utilizados por outras aplicações clientes.
<b>Regras</b>	Implementar o código necessário e testar no fim de cada nível. Use as convenções de codificação adotadas para a linguagem C# e para o modelo MVC.
<b>Descrição</b>	
<b>Nível 1</b>	<ul style="list-style-type: none"><li>• Este trabalho envolve a utilização dos mesmos modelos em projetos distintos. A estratégia seguida consiste em colocar os modelos dentro de uma <b>Class Library</b> reutilizando-os nos outros projetos. Assim sendo, comece por criar um novo projeto do tipo <b>Class Library (.Net Core)</b>, dando-lhe o nome de <b>EsteCarIIILibrary</b> e com a solução a chamar-se <b>EsteCarIII</b></li><li>• Adicione as classes definidas no projeto do <b>Lab 6</b> da aula anterior, nomeadamente, as classes <b>Marca</b>, <b>Carro</b>, <b>Aluguer</b>, <b>Cliente</b>, <b>NotWeekEndAttribute</b>, <b>ApplicationDbContext</b> e <b>DbInitializer</b>, colocando-as adequadamente dentro das diretorias <b>Data</b> e <b>Models</b>.</li><li>• Agora torna-se necessário incluir os pacotes relacionados com a <b>Entity Framework</b> e a <b>Microsoft Identity</b>. Para instalar a <b>EntityFramework</b>, no <b>Package Manager Console</b> do <b>NuGet</b> execute os seguintes comandos <i>Nuget</i>: <b>Install-Package Microsoft.EntityFrameworkCore.SqlServer</b> <b>Install-Package Microsoft.EntityFrameworkCore.Tools</b> <b>Install-Package Microsoft.EntityFrameworkCore.Design</b></li><li>• Também é necessária a instalação do <i>package</i> <b>Identity</b>: <b>Install-Package Microsoft.AspNetCore.Identity.EntityFrameworkCore</b></li><li>• Compile a aplicação e verifique e corrija qualquer erro que exista.</li><li>• Adicione à solução um novo projeto do tipo <b>ASP.NET Core Web Application</b> com o nome <b>EsteCarIIIWebApi</b>. Selecione o <i>template</i> <b>API</b> e <b>no authentication</b>. A seguir, selecione o novo projeto e com o botão direito do rato, defina-o como projeto de arranque → <b>“Set as StartUp Project”</b>. Execute o projeto (<b>CTRL+F5</b>) e constate que o <b>browser</b> apresenta um <i>array</i> de dados meteorológicos em formato <b>Json</b>.</li></ul>
<b>Nível 2</b>	<ul style="list-style-type: none"><li>• No projeto <b>EsteCarIIIWebApi</b>, criado antes, selecione <b>“Dependencies”</b> e adicione a referencia (<b>Add Reference &gt; Projects</b>) à biblioteca de classes criada antes: <b>EsteCarIIILibrary</b>.</li><li>• Na classe <b>Startup</b>, dentro do método <b>ConfigureServices</b>, imediatamente antes de <b>services.AddControllers()</b>, adicione a linha: <pre>services.AddDbContext&lt;ApplicationDbContext&gt;(options =&gt;     options.UseSqlServer(         Configuration.GetConnectionString("DefaultConnection"));</pre></li></ul>

# Programação Visual

## Trabalho de Laboratório nº 7

- Dentro do ficheiro de configuração **appsettings.json**, adicione a seguinte linha às já existentes, logo após a primeira chave:

```
"ConnectionStrings": {  
  "DefaultConnection":  
    "Server=(localdb)\\mssqllocaldb;Database=EsteCarIIIDb;Trusted_Connection=True;  
    ;MultipleActiveResultSets=true"  
},
```

- Na classe **Program**, e tal como efetuado no projeto **Lab 6**, altere o código para de maneira a que seja chamado o método:  

```
DbInitializer.Initialize(context);
```
- Compile a solução. Instale a seguir as mesmas bibliotecas que instalou para a **Entity Framework** no projeto **EsteCarIIILibrary** (verifique que é este projeto que está selecionado como *default* no **Package Manager Console** do **NuGet**). Crie agora a migração inicial e faça a atualização da base de dados. Constate a ausência de erros durante estes dois procedimentos.
- Adicione um controlador designado **MarcasController** ao projeto **EsteCarWebIIIApi**. Para isso, faça **Add > Controller** e, dentro da janela, selecione o modelo **API Controller with actions, using Entity Framework**.
- Compile a aplicação e execute-a (CTRL+F5)
  - Teste as funcionalidades criadas no *site*. Por exemplo, verifique quais os resultados que obtém, invocando no *browser* os exemplos abaixo. Note que vai necessitar de modificar o valor do porto nestes exemplos:  
<http://localhost:58279/api/Marcas>  
<http://localhost:58279/api/Marcas/2>  
Nota: verifique o valor do porto utilizado no ficheiro **launchSettings.json** na pasta das propriedades.

### Nível 3

- Vamos agora desenvolver uma aplicação cliente para a **WebApi** criada no nível anterior. Aplicações clientes típicas podem ser aplicações Web, Apps de Smartphones Android/iPhone, Windows Desktop, etc. Para efeitos ilustrativos, é suficiente utilizar um menu de opções feito numa janela de consola, que sirva para testar a interação entre as aplicações **EsteCarIIWebAPI** e **EsteCarIIIClient**
- Adicionar à solução um novo projeto do tipo **Console App (.Net Core)**, com o nome **EsteCarIIIClient**. Para facilitar o desenvolvimento deste novo projeto, defina-o como principal (**Set as Startup Project**).
- Na janela do projeto, adicione a referência ao projeto **EsteCarIIILibrary** (add reference).
- Selecione este projeto e instale usando a consola do **NuGet** o seguinte pacote:  
**Install-Package Microsoft.AspNetCore.WebApi.Client**

# Programação Visual

## Trabalho de Laboratório nº 7

### Nível 4

- Adicione ao projecto **EsteCarIIILibrary**, directoria Models, a classe **MetodosDeExtensao** fornecida com este enunciado.
- Abra agora o ficheiro fornecido com o nome **Cliente\_WebApi\_ilustrativo.txt** e copie o seu conteúdo para dentro da classe **Program** do projeto **EsteCarIIIClient**.
- No novo conteúdo da classe **Program**, Corrija o URL usado no seguinte código:

```
cliente.BaseAddress = new Uri("http://localhost:65346/");
```

Nota: O url correto está dentro do ficheiro **launchStteings.json** (entrada *applicationUrl*) do projeto **EsteCarIIIWebApi**.

- Compile e verifique a ausência de erros.
- Neste passo, vai necessitar de ter a funcionar simultaneamente os projetos **EsteCarIIIWebAPI** e **EsteCarIIIClient**. No **Solution Explorer**, selecione o projeto **EsteCarIIIWebAPI** e execute a opção "**Debug > Start New Instance**".
- De forma similar, execute o projeto **EsteCarIIIClient**.
- Experimente a opção 1 do menu fornecida já implementada. Constate o seu funcionamento correto.

### Nível 5

- Implemente o método **ToString()** para a classe **Marca**.
- Implemente os métodos **GetMarcaAsync**, **CreateMarcaAsync**, **UpdateMarcaAsync** e **DeleteMarcaAsync**.
- Chame cada um dos métodos implementados a partir da opção do menu existente no método **Main**. Constate o bom funcionamento dos métodos referidos.

### Notas

Para os identificadores siga as convenções adotadas pelo C#, nomeadamente:

- A notação camelCase para o nome das variáveis locais e identificadores privados.
- A notação PascalCase para os nomes públicos dos métodos e classes
- Não utilize o símbolo de qualquer forma '\_' nos identificadores
- Não use abreviaturas