Programação Visual
Trabalho de Laboratório nº 8

Objetivo	MVC – Introdução. Utilização de *ViewComponents* e utilização/definição de serviços
Programa	Pretende-se continuar o desenvolvimento da aplicação **EsteCar** .
Regras	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.

Descrição

Nível 1

- Descarregue e descompacte o ficheiro "**Lab 8 – Materiais**" fornecido com este enunciado. Depois de descompactar, irá encontrar duas pastas com as soluções de dois laboratórios anteriores. Abra a solução que está na pasta **EsteCarIII**. Nesta solução irá encontrar 3 projetos. Adicione à solução o projeto **EsteCarII** que foi fornecido na outra pasta. Neste caso, selecione "**Add Existing Project**" e procure o ficheiro **EsteCar2.csproj** que está na pasta **EsteCarII**. Torne este projeto o "**Startup Project**".
- Agora, pretendemos usar a base de dados do projeto que adicionou nos 4 projetos. Sendo assim, comece por criar a migração inicial e a base de dados para o projeto **EsteCarII**. Confirme que o Nuget tem selecionado este projeto antes de executar os comandos necessários.
- Execute a aplicação e confirme que está a funcionar. Deve encontrar já algumas marcas e carros quando escolhe as opções de menú respetivas. Adicione mais uma marca à sua escolha usando a opção de menú das marcas.
- Para que a aplicação **EsteCarIIIWebApi** funcione corretamente é necessário que esteja a utilizar a mesma base de dados que a que foi criada com o projeto adicionado. Para isso**,** vá ao ficheiro **appsettings.json** do projeto **EsteCarII** e copie o texto da *string* "**DefaultConnection**" em "**ConnectionStrings**" e coloque-o no mesmo ficheiro do projeto **EsteCarIIIWebApi** substituindo o texto que lá está. Verifique que está a funcionar executando este projeto (**Debug -> start new instance**) e testando a api.
- Sem parar a execução anterior, arranque agora com o projeto **EsteCarIIIClient** e verifique que também está a funcionar. Deverá ver na listagem das marcas, a marca que adicionou antes.

Nível 2

- Dentro do projeto **EsteCarII**, pretende-se agora criar um **ViewComponent** que mostre o total de alugueres de carros no dia de hoje. Para este efeito, comece por criar este componente e na vista associada escreva a seguinte tabela **com o texto mostrado**:

Alugueres
Hoje: 10 alugueres

- Teste este componente, colocando-o dentro na página **Privacy** do Controlador **Home**.

Programação Visual
Trabalho de Laboratório nº 8

Nível 3

- O componente anterior está incompleto porque o número de alugueres não é obtido de acordo com a informação que está na base de dados. Para que isso aconteça, ir-se-á criar um serviço a partir do qual se pode obter a informação dos alugueres. Crie então, dentro de uma pasta **Services**, a seguinte interface:

```
public interface IAlugueresHoje
{
     IEnumerable<Aluguer> AlgueresHoje { get; }
}
```

  A seguir, crie uma classe **`AlugueresHoje`** que implementa a interface anterior. Dentro desta classe, obtenha o contexto por *dependency injection* (como é feito dentro do controlador **Alugueres**) e, a partir dele, retorne no *get* da propriedade **`AlugueresHoje`** os registos dos alugueres que sejam do dia atual.
- Para usar o serviço criado, registe-o na classe **`Startup`** como *Scoped*:

```
services.AddScoped<IAlugueresHoje, AlugueresHoje>();.
```

- Para testar o serviço, obtenha a lista de carros **diretamente na págin**a **index** do controlador **Home** usando a injeção de dependência do serviço nesta vista (consulte https://docs.microsoft.com/en-us/aspnet/core/mvc/views/dependency-injection?view=aspnetcore-3.0). Deverá aparecer o texto "**Só Hoje 1 Carro(s) Alugado(s)!**". O número de carros é obtido a partir do número de elementos da lista de alugueres.

Nível 4

- Altere o *ViewComponent* criado no nível 2 de forma a que este receba a lista de alugueres efetuados no dia atual, usando o serviço **`IAlugueresHoje`**, e a passe para a vista associada. Neste caso, a vista passa a ser tipificada com o modelo **`IEnumerable<Aluguer>.`** Corrija a vista para que passe a obter o número de alugueres do dia atual a partir da lista de alugueres que passou a receber. Teste.

Nível 5

- Para completar a aplicação, pretende-se criar e usar um serviço de *email* que permita a confirmação do *email* fornecido quando um utilizador se regista. Neste caso, vamos utilizar a biblioteca **Mailkit**. Sendo assim execute o seguinte comando NuGet:

```
Install-Package MailKit
```

- Para criar o serviço de email, dentro da pasta **services,** crie uma classe **`EMailSender`** que implementa a interface **`IEmailSender`**. O código desta classe está no ficheiro **Emailsender.txt** fornecido com os materiais que acompanham este laboratório**.**
- Antes de registar o serviço, coloque a entrada **EmailSender** (fornecida no ficheiro **EmailsenderJson.txt**) no ficheiro de configurações **appsettings.json**. No texto, estão as definições de uma conta de email que foi criada para este laboratório e que será removida posteriormente.

Programação Visual
Trabalho de Laboratório nº 8

- Registe o serviço usando:

```
services.AddTransient<IEmailSender, EmailSender>(i =>
    new EmailSender(
        Configuration["EmailSender:Host"],
        Configuration.GetValue<int>("EmailSender:Port"),
        Configuration.GetValue<bool>("EmailSender:EnableSSL"),
        Configuration["EmailSender:UserName"],
        Configuration["EmailSender:Password"]
    )
);
```

- Teste.

Desafio

- Dentro do projeto **EsteCarII** crie um controlador para os utilizadores que receba a lista dos utilizadores a partir de um serviço criado para o efeito. O serviço criado deve obter os utilizadores usando uma API própria a ser criada no projeto **EsteCarIIIWebApi**.

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