

Programação Visual

Trabalho de Laboratório nº 3

Objetivo	ASP.NET Core 5 MVC – Introdução. Aplicações básicas
Programa	Pretende-se criar um <i>Site</i> para uma empresa de cinema local para o IPS. O <i>site</i> deverá permitir a visualização dos filmes disponíveis e os seus detalhes.
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.

Nível 1

- Crie uma aplicação com o *template* **ASP.NET Core Web App (Model-View-Controller)** com o nome **CineSetúbal**.
- Comece por editar a página de Layout que serve de base ao *site* e efetue as seguintes alterações:
 - Nas opções de menu do *site* substitua a palavra “**Privacy**” por “**Privacidade**”.
 - Elimine a linha que tem a opção de menu “**Home**”.
- Dentro da pasta **wwwroot** crie uma pasta **images** e adicione a imagem **under_construction.png** fornecida com este projeto. Na **View index (index.cshtml)** do controlador **Home** substitua todo o conteúdo *html* por um parágrafo com o texto “**O CineSetúbal está em construção**”. Inclua, também, a imagem que acrescentou no início.
- Na vista “**Privacy**” substitua o texto em inglês por um texto equivalente em português.
- No controlador **HomeController** crie a ação **Contact** e a vista associada. Na criação da vista use as opções por omissão. Na **View contact.cshtml** criada substitua o texto *html* da mensagem mostrada no corpo da página pelos conteúdos do documento “**Contacts.cshtml**”.
Para aceder à página criada, inclua mais uma opção de menu na página de *layout*, neste caso a opção “**Contactos**”. Coloque-a antes da opção “**Privacidade**”.
- À semelhança do que fez anteriormente, crie a ação **About** e a vista associada dentro do mesmo controlador e coloque na vista um texto com o nome(s) do(s) elemento(s) do grupo que está(ão) a desenvolver a página. A ação criada deve ter uma entrada “**Sobre**” no menu principal antes da opção “**Contactos**” criada anteriormente.

Programação Visual

Trabalho de Laboratório nº 3

Nível 2

- Crie uma classe **Filme** com propriedades para o **Título**, **Diretor**, **Atores** (Lista de nomes), **Duração** (inteiro), **Ano** e **Estreia** (bool).
- Adicione à classe do controlador **Home** um atributo **filmes** do tipo **List<Filme>**. No construtor inicialize a lista de filmes com o código que está no ficheiro **Filmes.txt**.
- Crie agora uma nova ação **FilmeDoDia** e no corpo do método dessa ação, obtenha o primeiro filme da lista de filmes e passe-o depois como argumento do método **View**. Crie a chave **"IndexFilme"** do **ViewData** colocando como valor um texto com o índice do filme e o total de filmes na lista (por exemplo **"Filme 1 de 4"**).
- Para visualizar o filme do dia da ação que criou será necessária uma nova **Razor View**. Sendo assim adicione uma nova **View** para a ação **FilmeDoDia**. Na vista criada deve usar o *template* **details** e como modelo a classe **Filme**. Altere a vista para que apareça apenas o título, o diretor e o ano do filme mostrado. No fim da descrição adicione o texto recebido no **ViewData**.
- Acrescente agora na página de *layout* mais uma opção ao menu do site criado com o texto **"Filme do dia"** que deve ir para a vista que acabou de criar.

Nível 3

- Crie um novo controlador **Filmes** (Selecione **MVC Controller-Empty** na janela de criação do controlador). Para efeitos de teste, crie um atributo **filmes** do tipo **List<Filme>** tal como fez para o controlador **Home** e inicie a lista no construtor da classe com o mesmo código utilizado anteriormente.
- Defina agora uma ação **Cartazes** que passa para a vista associada a lista de filmes guardada no atributo. Crie a vista associada usando o *template* **"Empty - without model"**. Use depois, na vista, como modelo, o tipo **IEnumerable<Filme>** de acordo com o objeto passado. Na vista, liste os vários filmes da lista mostrando o título, o diretor e o ano.
- Inclua no menu principal uma opção **"Cartazes"** que vai chamar a ação criada neste nível. Teste e verifique se está a funcionar corretamente.

Nível 4

- Crie agora uma nova ação **Estreia** no controlador **"Filmes"**. Esta ação deve filtrar os filmes (usando LinQ) selecionando apenas os que estão em estreia e depois passar o resultado para a vista. Na criação da vista irá usar *Scaffolding*. Sendo assim, adicione uma vista e na janela de criação da vista escolha o *template* **List** e a classe de modelo **Filme**. Crie uma nova opção de menu **"Estreias"** para esta ação e verifique o resultado.
- Como deve verificar aparecem na vista uma listagem dos filmes em estreia. Analise o ficheiro da vista e adicione a lista de atores, remova o que for necessário para que não apareça a coluna **Estreia** e os links que estão depois. Remova também o link **"Create New"**. Verifique que a vista está agora de acordo com o pretendido.
- Altere a vista para que apareça uma mensagem **"Não existem filmes em estreia"** quando a lista de filmes estiver vazia.

Programação Visual

Trabalho de Laboratório nº 3

Nível 5

- Crie uma pasta **Documents** dentro da pasta **wwwroot** e adicione-lhe os ficheiros “pdf” que estão na pasta dos materiais.
- Adicione agora ao controlador **Home** uma ação com o nome **DescarregarBilhete**. Esta ação que retorna um **VirtualFileResult** deve retornar em vez da **View**, um **File**, fornecendo-se o nome do ficheiro criado da seguinte forma:

```
return File("~/documents/nomedoficheiro.pdf", "application/pdf",  
"nomedoficheiro.pdf");
```

- Para usar esta ação crie o seguinte *link* na *view* do Filme do dia:

```
<h4><a asp-controller="Home" asp-action="DescarregarBilhete"  
class="label label-success">Bilhete</a></h4>
```

- Generalize o método criado para que funcione com qualquer filme do dia que esteja selecionado na ação respetiva.

Desafio

- Altere a aplicação de forma que seja possível visualizar o cartaz dos filmes em todos os locais onde estes são apresentados. Pode encontrar os cartazes na pasta **Imagens** fornecida com o enunciado.

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, classes e interfaces.
- Não utilize o símbolo ‘_’ nos identificadores nem abreviaturas