

> Ficha Prática Nº 4

Objetivos:

- Consolidar conceitos sobre Entity Framework Core, LINQ, Formulários, Models e ViewModels

> Exercícios

- CRIAR UMA PARTIAL VIEW COM FORMULÁRIO DE PESQUISA
- MODIFICAR VISTAS DE PESQUISA, DETALHES DO CURSO, EDITAR CURSO E CRIAR CURSO
- MODIFICAR VISTAS DETALHES DA CATEGORIA, EDITAR CATEGORIA E CRIAR CATEGORIA
- CRIAR FORMULÁRIO “TEMPORÁRIO” PARA AGENDAMENTO DE AULAS

>> Vista Parcial & Pesquisa – continuação da ficha 3

1. Crie um Partial View (**QuickSearchPartial.cshtml**) com um formulário de pesquisa para criar uma pesquisa rápida no site.
 - O formulário deve submeter os dados através do método GET para o *Controller Cursos*, método *Search*.
2. Adicione a Partial View aos dois ficheiros de _layout por forma a obter um resultado semelhante a:



- Para isso, procure no ficheiro de Layout o código: “<partial name=“_LoginPartial” />” e na linha anterior adicione o seguinte código:

```
<partial name="~/Views/Cursos/QuickSearchPartial.cshtml" />
```

Nota: se a *Partial View* for criada na diretoria “/Views/Shared/” não é necessário escrever o caminho completo, bastando escrever o nome.

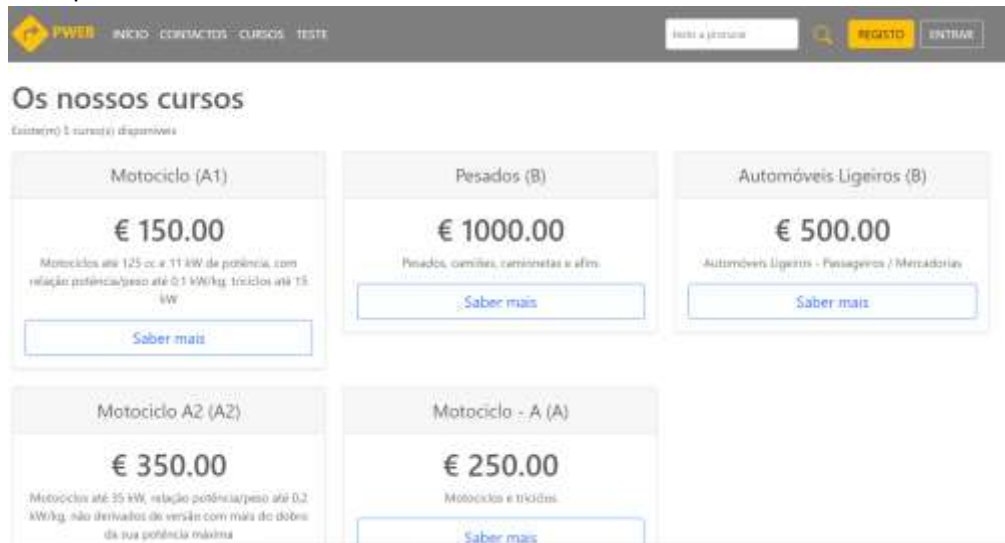
3. Faça as alterações necessárias para:

- No caso de a pesquisa não devolver resultados deve mostrar ao utilizador a seguinte mensagem:
 - “Não foi possível encontrar cursos com os dados introduzidos. Por favor, reveja a sua pesquisa”.



- Mostrar a lista de resultados em formato “card”, com os seguintes elementos:
 - Nome do Curso (Nome Categoria)
 - Descrição Resumida
 - Preço
 - Botão “saber mais” – redirecionar para os detalhes de um curso.

Exemplo:



4. Faça as alterações necessárias para:

- Se o nome do curso ou o resumo da descrição contiver o texto pesquisado, colocar esse pedaço de texto com fundo amarelo.

Exemplo:



>> Formulários

5. Altere as Views (*Edit*, *Create*) do Cursos por forma a utilizar a tag “<TextArea>” para as propriedades “Descricao”, “DescricaoResumida” e Requisitos em vez da tag “<input>”.
6. Altere a *view Details* do *controller* Cursos de forma a mostrar os detalhes de um curso de uma forma mais “apelativa”.

Nota: Veja exemplos em <https://getbootstrap.com/>.

7. Adicione *Data Annotations* às propriedades da classe “Curso.cs” de forma que o preenchimento dos formulários e a visualização das propriedades, seja melhorada através da geração da descrição da propriedade, do placeholder e da geração dos atributos de validação, se necessários.

Todas as propriedades devem, no mínimo, ter o atributo [Display] definido.

Exemplo:

```
[Display(Name = "Idade mínima", Prompt = "Introduza a idade Mínima", Description = "Idade mínima para poder frequentar esta formação")]
[Range(14, 100, ErrorMessage = "Mínimo: 14 anos, máximo: 100 anos")]
public int IdadeMinima { get; set; }
```

8. Faça o mesmo (exercícios 5, 6 e 7) para as Categorias.

9. Adicione um item no menu superior que “aponte” para a View “Index” do controller “Categorias”.
10. Criar um formulário de “agendamento” de aulas, que permita calcular o custo do agendamento.
 - Criar um Modelo novo chamado “TipoDeAula”, com as seguintes propriedades
 - Id - `int`
 - Nome - `String`
 - ValorHora - `Decimal`
 - Crie um Modelo novo chamado “Agendamento”, com as seguintes propriedades
 - Id - `int`
 - Cliente - `String`
 - DataInicio - `DateTime`
 - DataFim - `DateTime`
 - DuracaoEmHoras - `int`
 - DuracaoEmMinutos - `int`
 - Preco – `Decimal`
 - DataHoraDoPedido – `DateTime` – representa a data e Hora em que o Utilizador efetuou o cálculo.
 - Crie um formulário de agendamento, que permita ao utilizador escolher duas datas, e que consoante a escolha do utilizador, calcule o número de horas, bem como o custo desse agendamento.
 - Efetue as alterações necessárias para guardar na base de dados os pedidos de agendamento efetuados.