Fundamentos de Desenvolvimento Web com Banco de Dados

Aula 04: Razor Syntax

Professor: Rinaldo Ferreira Junior

E-mail: rinaldo.fjunior@prof.infnet.edu.br

- **Professor:** Rinaldo Ferreira Junior
- **Graduação:** Pós-graduado em Arquitetura de Softwares
- **Atuação:** .Net | C# | SQL | NoSQL | Engenheiro de Software
- **E-mail:** rinaldo.fjunior@prof.infnet.edu.br
- **Linkedin:** https://www.linkedin.com/in/rinaldo-ferreira-junior-787326a

Roteiro

- Razor Syntax

Aula 04: Razor Syntax

Instituto Infnet

- Qualquer framework para desenvolvimento web, precisa ser capaz de gerar HTML dinâmicamente.
- Todos esses frameworks utilizam o padrão chamado Template View. Esse padrão consiste de marcadores ou placeholders contendo código server side, que são mesclados ao HTML.
- Quando executados, esses marcadores ou placeholders geram código que irá ser renderizado junto ao HTML original.
- O conteúdo dinâmico podem ser dados de um banco de dados, ou o resultado de alguma computação. De qualquer forma, será preciso usar marcadores e placeholders para controlar a apresentação desses dados.

Aula 04: Diretivas

- Diretivas são expressões do C#, iniciadas com @ tendo uma palavra reservada na sequência.
- Diretivas habilitam recursos em uma página ou mudam a maneira como a página é renderizada.
- Diretivas mais comuns nas páginas são:
  - @page que indica que o arquivo é uma página navegável
  - @model que indica o Type usado como modelo na página
  - @using que traz as funcionalidades de um namespace, para a página

```
@page
@model ExplorerModel
@using static System.IO.Path
@{
    var extension = GetExtension("arquivo.ext");
}
```

Aula 04: Arquivo _ViewImports.cshtml

- Um arquivo com objetivo especial na estrutura do Razor, é o arquivo *_ViewImports.cshtml*.
- Esse arquivo centraliza diretivas que serão usadas por todos as páginas Razor, evitando que seja preciso adicionar diretivas página por página.
- O arquivo padrão já inclui 3 diretivas:
  - Uma diretiva @using que referencia o namespace do projeto
  - Uma diretiva @namespace que indica o namespace padrão para todos os arquivos do projeto
  - Uma diretiva @addTagHelper que permite a utilização de Tag Helpers nas páginas

```
@using SampleWeb
@namespace SampleWeb.Pages
@addTagHelper *, Microsoft.AspNetCore.Mvc.TagHelpers
```

Aula 04: Blocos de Código (Code Blocks)

Instituto Infnet

- Code Blocks começam com uma @ seguido de chaves {}. O conteúdo das chaves é o código C# a ser executado.
- O ideal é procurar manter poucos code blocks na página, com o objetivo de controlar a apresentação. Muitos blocos pode ser sinal de que há lógica de aplicação na página HTML.
- Um tipo específico de code block é o functions block. Permite a criação de métodos que podem ser chamados em outros pontos do .cshtml.

```
<label class="col-sm-2 col-form-label col-form-label-lg">Nome:</label>
<div class="col-sm-10">
    @{
        var blocoNome = Model.Login?.Nome;
    }
    <p><strong>@(blocoNome)</strong></p>
</div>
```

```
@functions{
    string GetDayPeriod(DateTime dt)
    {
        var partOfDay = "manhã";
        if (dt.Hour > 12)
        {
            partOfDay = "tarde";
        }
        if (dt.Hour > 18)
        {
            partOfDay = "noite";
        }
        return partOfDay;
    }
}
<p>Período da @GetDayPeriod(DateTime.Now)</p>
```

Aula 04: Transições

- A linguagem padrão em um code block é C#, mas o Razor permite transiocionar para HTML de forma implícita:

```
@{
    var transition = true;
    <p>Código transicionado para HTML @transition</p>
}
```

- Ou de forma explícita:

```
@{
    var person = new Login();
    person.Nome = "Maria das Dores";
    <text><strong>Nome:</strong>@person.Nome</text>
}
```

- A tag <text> é útil para controlar a renderização de espaços em branco:
  - Só o conteúdo entre as tags <text>...</text> é renderizado
  - Espaços em branco antes ou depois da tag <text> não são exibidos

Aula 04: Control Blocks (Blocos de Controle)

Instituto Infnet

- A sintaxe Razor deve ser usada principalmente, para controlar a exibição. Portanto, a maior parte do código server side usado em páginas, são estruturas de controle.

- Usa-se o caracter @ na frente da estrutura de controle, e o Razor entende como sendo um bloco de código com a estrutura.

```
@switch (DateTime.Now.Hour)
{
    case int _ when DateTime.Now.Hour ⩽ 12:
        <p>Manhã</p>
        break;
    case int _ when DateTime.Now.Hour ⩽ 18:
        <p>Tarde</p>
        break;
    default:
        <p>Noite</p>
        break;
}
```