



Fundamentos de Desenvolvimento Web com Banco de Dados

Aula 02: Razor Pages

Professor: Rinaldo Ferreira Junior

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

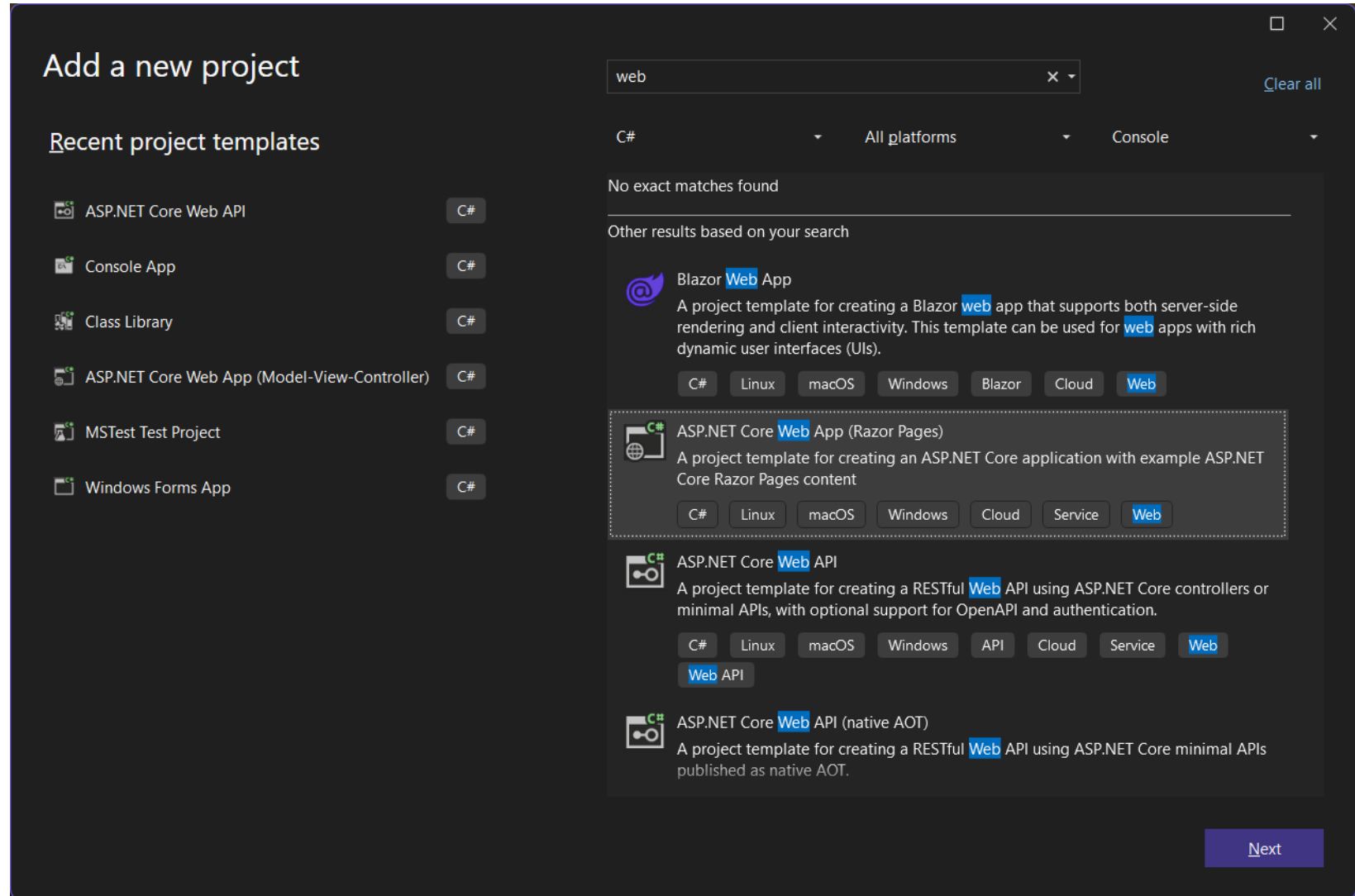


- **Professor:** Rinaldo Ferreira Junior
- **Graduação:** Pós-graduado em Arquitetura de Softwares
- **Atuação:** .Net | C# | SQL | NoSQL | Engenheiro de Software
- **E-mail:** rinaldo.fjunior@prof.infnet.edu.br
- **Linkedin:** <https://www.linkedin.com/in/rinaldo-ferreira-junior-787326a>

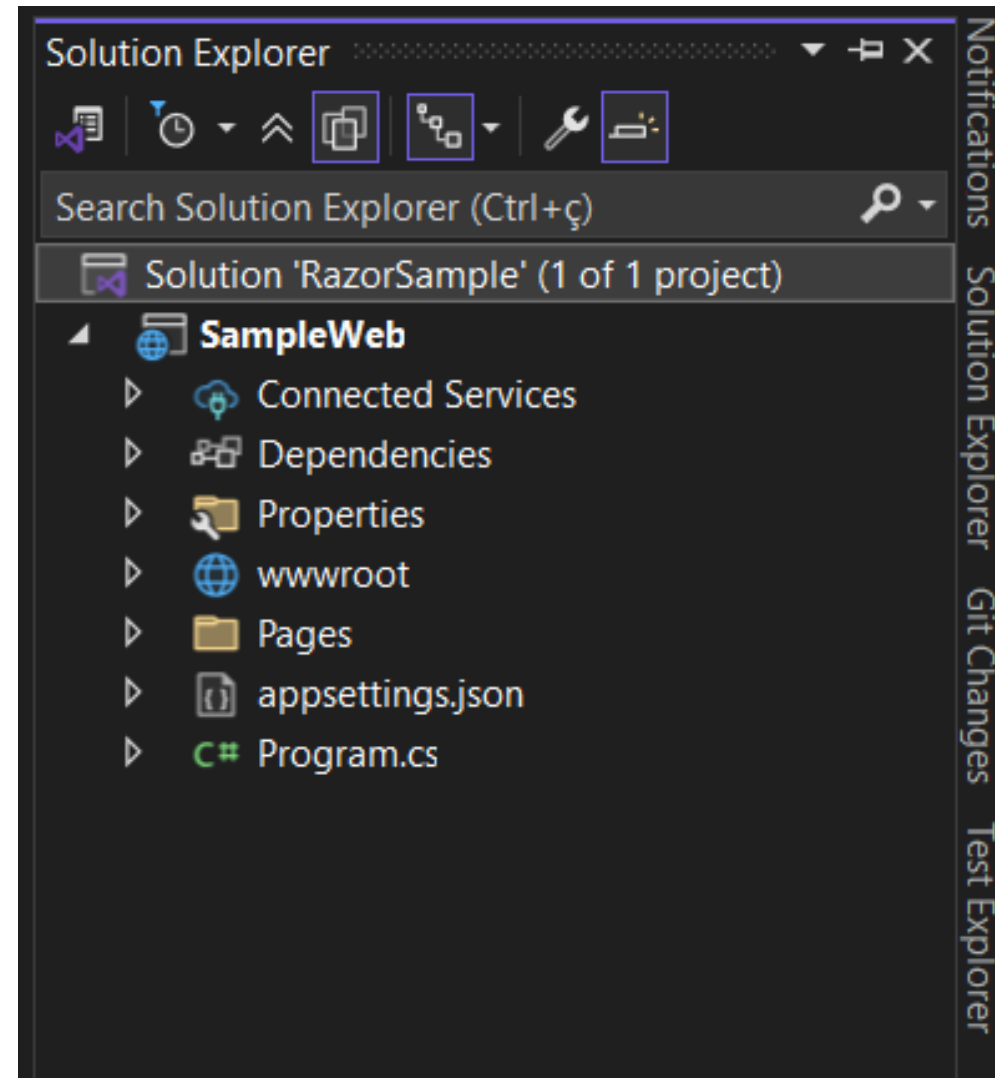
- Razor Pages

- O Razor é uma tecnologia da Microsoft, que interpola HTML, CSS e JavaScript, com o C#.
- Razor Pages procura simplificar a construção de aplicações web, reduzindo a conhecida estrutura do **MVC (Model-View-Controller)**. Mas o Razor é baseado no MVC.
- Razor não tem **Controllers**. Ele trabalha com uma abordagem que une **View** e **Controller** em um único arquivo.
- Cada arquivo por sua vez, é um **endpoint**.
- Um **PageModel** possui o comportamento da página. O código é escrito em C#, compilado no servidor e então é gerada a resposta para o browser.

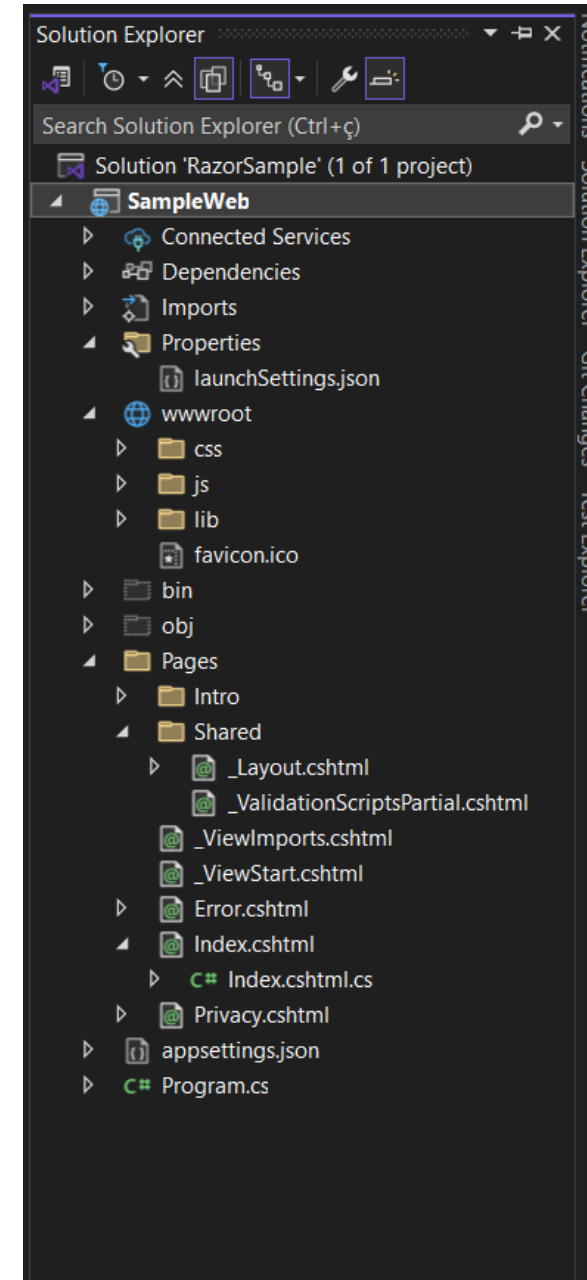
- Clique em [Add – New Project](#) e selecione [ASP Net Core Web App \(Razor Pages\)](#).
- Informe o nome da aplicação e escolha as configurações iniciais.

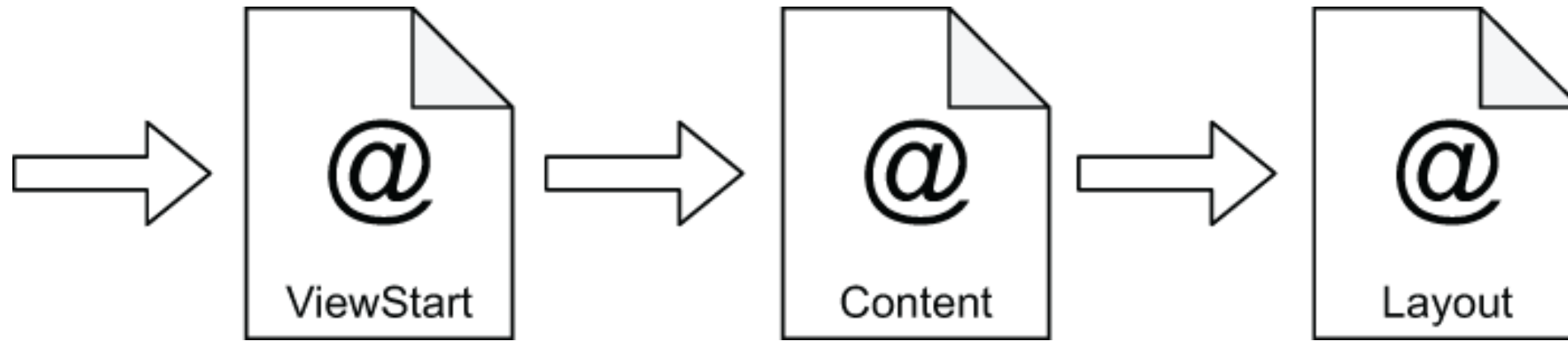


- A pasta **Properties** possui configurações específicas do projeto. O arquivo **launchSettings.json** contém os detalhes de configuração da aplicação.
- A pasta **wwwroot** é o diretório raiz da aplicação e contém arquivos estáticos. Seu conteúdo pode ser navegado diretamente.
- A pasta **Pages** hospeda as páginas da aplicação. O .Net espera que as páginas estejam lá.



- A pasta **Pages** possui uma subpasta **Shared**. Nesta pasta se situa o arquivo **_Layout.cshtml**, que é um template para as demais páginas do projeto.
- O arquivo **_Layout** possui uma chamada para o método **RenderBody**. Nesse ponto, as **content pages** são renderizadas.
- Já as content pages referenciam seus templates de layout, através da propriedade **Layout**, que aparece em um bloco de código no topo da página.
- O arquivo **_ViewStart** centraliza código que deve ser executado no carregamento de cada página Razor, inclusive, o template.





- **ViewStart** é executado antes de qualquer **Page**. Por isso é o local ideal para definir layouts comuns à toda a aplicação

- O arquivo `.cshtml` é o Razor Page. Pode ser chamado de **Content Page** ou **View**. Codificado na sintaxe do Razor, que une HTML e C#.

```
@page
@model IndexModel
@{
    ViewData["Title"] = "Home page";
}

<div class="text-center">
    <h1 class="display-4">Welcome</h1>
    <p>Learn about <a href="https://learn.microsoft.com/aspnet/core">building Web apps with ASP.NET Core</a>.</p>
</div>
```

- O arquivo `.cshtml.cs` é o arquivo de código, derivado de **PageModel**. É a classe que combina **Controller** e **View**.
- Aqui entram as operações da página, como os verbos HTTP Get, Post, Put etc.

```
namespace SampleWeb.Pages
{
    public class IndexModel : PageModel
    {
        private readonly ILogger<IndexModel> _logger;

        public IndexModel(ILogger<IndexModel> logger)
        {
            _logger = logger;
        }

        public void OnGet()
        {
        }
    }
}
```

- Novas páginas devem ser colocadas na pasta **Pages**.
- Clique o botão direito na pasta **Pages** e clique em **Add – Razor Page**.
- Informe o nome e o tipo de template da página.

