

## 1 - Arquivo: Views/Home/Produtos.cshtml (e Contato.cshtml)

Explicação: São os documentos de interface. Aqui escrevemos apenas o conteúdo necessário da página. Todo o resto do HTML (como tags <html>, <head>, scripts e estilos) é montado automaticamente pelo arquivo \_Layout.cshtml.

```
@{  
    ViewData[index: "Title"] = "Nossos Produtos"; // Define o valor do título que será injetado na tag <title> do Layout (HTML)  
}  
  
<h1>Página de Produtos</h1>  
<p>Aqui você verá a lista de itens da loja.</p>  
<a href="/">Voltar para Home</a>
```

## 2 - Arquivo: Controllers/HomeController.cshtml

Explicação: Responsável pela lógica das rotas. Ao dar o mesmo nome para a função (do arquivo cshtml), o .NET já procura automaticamente o arquivo correspondente dentro da pasta Views e da subpasta Home.

```
// Rota localhost:5000/Home/Produtos  
0 referências  
public IActionResult Produtos() // Apenas escrever o nome do arquivo cshtml que ele já encontra na pasta Views/Home  
{  
    return View();  
}
```

## 3 - Arquivo: Program.cs (Trecho final)

Explicação: É o configurador da Rota Raiz. Ele define qual é o primeiro documento que o servidor deve buscar "por debaixo dos panos" quando o endereço do site é acessado sem caminhos adicionais.

```
// DEFINE A PORTA DE ENTRADA DO SITE (ROTA RAIZ)  
// Este comando diz ao .NET: "Se o usuário acessar o site e não digitar nenhum caminho,  
// carregue automaticamente o controlador 'Home' e execute a função 'Index'."  
app.MapControllerRoute(  
    name: "default",  
    pattern: "{controller=Home}/{action=Index}/{id?}")  
    .WithStaticAssets();
```

# Comandos de Attribute Routing e Razor Syntax

## --> Categoria: Lógica e Rotas (Controller)

Explicação: O uso do [Route] permite personalizar o endereço que aparece no navegador, ignorando o padrão longo do .NET.

```
// Rota personalizada usando Atributo: localhost:5000/fale-conosco
[Route("fale-conosco")] // Deixa a rota mais limpa , sem o padrão /Home/Contato
0 referências
public IActionResult contato() // Apenas escrever o nome do arquivo cshtml que ele já encontra na pasta Views/Home
{
    return View();
}
```

## --> Categoria: Título e Interface (View - .cshtml)

Explicação: O ViewData serve para enviar informações da página específica para o Layout principal.

```
@{
    ViewData["Title"] = "Página Inicial de Estudos"; // Define o valor do título que será injetado na tag <title> do
    Layout (HTML)
}

<h1>Bem-vindo ao meu projeto de Rotas!</h1>
<p>Escolha uma das páginas abaixo para testar o roteamento:</p>
<ul>
    <li>
        <a href="/Home/Produtos">Ver Produtos (Rota: /Home/Produtos)</a>
    </li>
    <li>
        <a href="/fale-conosco">Ir para Contato (Rota: /fale-conosco)</a>
    </li>
</ul>
```