




<div><div> Comendador João Rays Barra Bonita</div><div> Centro Paula Souza</div><div> GOVERNO DO ESTADO DE SÃO PAULO</div></div>			Menção:
Avaliação Prática	Bimestre: 2º	Professor: Gallo	Turma:
Aluno (a):			Data:
Objetos de Conhecimento: 1. Persistência em banco de dados; 1.1. Conexões e Execução de comandos SQL (revisão); 1.3. Consultas parametrizadas; 1.4. Sanitização e prevenção de SQL Injection e XSS (cross-site scripting). 3.1. Orientação à objetos;			
Critério: CP	Critério: RP	Critério: RL	Critério: O

Siga as instruções abaixo para o desenvolvimento da aplicação **EtecShop**.

1. Crie no seu **github** um repositório com o nome **EtecShop-AtividadePratica**, lembre-se de incluir um **README** e na opção gitignore selecionar **Visual Studio**.
2. Clone o repositório na área de trabalho da sua máquina, abra o terminal e execute todos os comandos abaixo:
 - a. `dotnet new mvc -o EtecShop`
 - b. `cd EtecShop`
 - c. `dotnet add package Microsoft.EntityFrameworkCore --version 8.0.6`
 - d. `dotnet add package Microsoft.EntityFrameworkCore.Design --version 8.0.6`
 - e. `dotnet add package Microsoft.EntityFrameworkCore.Tools --version 8.0.6`
 - f. `dotnet add package Microsoft.EntityFrameworkCore.InMemory --version 8.0.6`
 - g. `dotnet add package Microsoft.AspNetCore.Identity.EntityFrameworkCore --version 8.0.6`
3. Arquivo **EtecShop.csproj**, mudar na linha 5:

```
<Nullable>disable</Nullable>
```

4. Arquivo **ErroViewModel**, apague apenas o caracter ? da linha 5.
5. Na pasta **Models**, crie a classe **Categoria**, usando o código abaixo:

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace EtecShop.Models;

[Table("Categoria")]
public class Categoria
{
    [Key]
    [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
    public int Id { get; set; }

    [Required(ErrorMessage = "Informe o Nome da Categoria")]
    [StringLength(30, ErrorMessage = "O Nome deve possuir até 30 caracteres")]
    public string Nome { get; set; }
}
```

6. Na pasta **Models**, crie a classe **Produto**, usando o código abaixo:

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace EtecShop.Models;

[Table("Produto")]
public class Produto
{
    [Key]
    [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
    public int Id { get; set; }

    [Required(ErrorMessage = "Informe o Nome do Produto")]
    [StringLength(100, ErrorMessage = "O Nome deve possuir até 100 caracteres")]
    public string Nome { get; set; }

    [Display(Name = "Descrição")]
    [StringLength(1000, ErrorMessage = "A Descrição deve possuir até 1000 caracteres")]
    public string Descricao { get; set; }

    [Display(Name = "Preço")]
    [Column(TypeName = "decimal(10,2)")]
    [Required(ErrorMessage = "Informe o Preço de Venda")]
    public decimal Preco { get; set; }

    [Display(Name = "Qtde. Estoque")]
    [Required(ErrorMessage = "Informe a Qtde. em Estoque")]
    public int Estoque { get; set; }

    [Display(Name = "Categoria")]
    [Required(ErrorMessage = "Informe a Categoria")]
    public int CategoriaId { get; set; }
    [ForeignKey("CategoriaId")]
    public Categoria Categoria { get; set; }

    [StringLength(300)]
    public string Foto { get; set; }
}
```

7. Na pasta **Models**, crie a classe **Avaliacao**, usando o código abaixo:

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace EtecShop.Models;

[Table("Avaliacao")]
public class Avaliacao
{
}
```

```

[Key]
[DatabaseGenerated(DatabaseGeneratedOption.Identity)]
public int Id { get; set; }

[Required(ErrorMessage = "Informe o Nome")]
[StringLength(60, ErrorMessage = "O Nome deve possuir até 60 caracteres")]
public string Nome { get; set; }

[Required(ErrorMessage = "Informe o Título")]
[StringLength(60, ErrorMessage = "O Título deve possuir até 60 caracteres")]
public string Titulo { get; set; }

[Required(ErrorMessage = "Escreva sua Avaliação")]
[StringLength(300, ErrorMessage = "A Avaliação deve possuir no máximo 300 caracteres")]
public string Texto { get; set; }

[Display(Name = "Data da Avaliação")]
public DateTime DataAvaliacao { get; set; } = DateTime.Now;

[Display(Name = "Produto")]
[Required(ErrorMessage = "Informe o Produto")]
public int ProdutoId { get; set; }
[ForeignKey("ProdutoId")]
public Produto Produto { get; set; }
}

```

8. Crie na pasta do projeto, uma pasta com o nome **Data**, e dentro uma classe com o nome **AppDbContext**, usando o código abaixo:

```

using EtecShop.Models;
using Microsoft.AspNetCore.Identity;
using Microsoft.AspNetCore.Identity.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore;

namespace EtecShop.Data;

public class AppDbContext : IdentityDbContext
{
    public AppDbContext(DbContextOptions<AppDbContext> options) : base(options)
    {
    }

    public DbSet<Avaliacao> Avaliacoes { get; set; }
    public DbSet<Categoria> Categorias { get; set; }
    public DbSet<Produto> Produtos { get; set; }

    protected override void OnModelCreating(ModelBuilder builder)
    {
        base.OnModelCreating(builder);
        #region Populando os dados da Gestão de Usuários
        List<IdentityRole> roles = new()
        {
            new IdentityRole()

```

```

        {
            Id = Guid.NewGuid().ToString(),
            Name = "Administrador",
            NormalizedName = "ADMINISTRADOR"
        },
        new IdentityRole()
        {
            Id = Guid.NewGuid().ToString(),
            Name = "Funcionário",
            NormalizedName = "FUNCIONÁRIO"
        },
        new IdentityRole()
        {
            Id = Guid.NewGuid().ToString(),
            Name = "Cliente",
            NormalizedName = "CLIENTE"
        }
    };
    builder.Entity<IdentityRole>().HasData(roles);

    IdentityUser user = new()
    {
        Id = Guid.NewGuid().ToString(),
        Email = "admin@etecshop.com",
        NormalizedEmail = "ADMIN@ETECSHOP.COM",
        UserName = "Admin",
        NormalizedUserName = "ADMIN",
        LockoutEnabled = true,
        EmailConfirmed = true,
    };
    PasswordHasher<IdentityUser> pass = new();
    user.PasswordHash = pass.HashPassword(user, "@Etec123");
    builder.Entity<IdentityUser>().HasData(user);

    List<IdentityUserRole<string>> userRoles = new()
    {
        new IdentityUserRole<string>() {
            UserId = user.Id,
            RoleId = roles[0].Id
        },
        new IdentityUserRole<string>() {
            UserId = user.Id,
            RoleId = roles[1].Id
        },
        new IdentityUserRole<string>() {
            UserId = user.Id,
            RoleId = roles[2].Id
        }
    };
    builder.Entity<IdentityUserRole<string>>().HasData(userRoles);
    #endregion
}
}

```

9. Altere o arquivo **appsettings.json**, conforme o código apresentado abaixo:

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "EtecShopConnectionString": "EtecShopDB"
  }
}
```

10. Para concluir a conexão com o banco de dados, altere o código do arquivo **Program.cs**, para criar o serviço de conexão.

```
using EtecShop.Data;
using Microsoft.EntityFrameworkCore;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.
string conn = builder.Configuration.GetConnectionString("EtecShopConnectionString");
builder.Services.AddDbContext<AppDbContext>(
    opt => opt.UseInMemoryDatabase(conn)
);

builder.Services.AddControllersWithViews();

var app = builder.Build();

using (var scope = app.Services.CreateScope())
{
    var context = scope.ServiceProvider.GetRequiredService<AppDbContext>();
    context.Database.EnsureCreated();
}

// Configure the HTTP request pipeline.
if (!app.Environment.IsDevelopment())
{
    app.UseExceptionHandler("/Home/Error");
    // The default HSTS value is 30 days. You may want to change this for production
    scenarios, see https://aka.ms/aspnetcore-hsts.
    app.UseHsts();
}

app.UseHttpsRedirection();
app.UseStaticFiles();

app.UseRouting();
```

```

app.UseAuthorization();

app.MapControllerRoute(
    name: "default",
    pattern: "{controller=Home}/{action=Index}/{id?}");

app.Run();

```

11. Salve todos os arquivos, execute o programa com o comando abaixo, e chame o professor para conferir seu trabalho até aqui:

dotnet watch run

12. Volte ao arquivo **AppDbContext**, e acrescente o código abaixo logo após o fechamento da última **region**:

```

builder.Entity<IdentityUserRole<string>>().HasData(userRoles);
#endregion

#region Popular Categorias
List<Categoria> categorias = new(){
    new Categoria() {
        Id = 1,
        Nome = "Livros e papelaria"
    },
    new Categoria() {
        Id = 2,
        Nome = "Games e PC Gamer"
    },
    new Categoria() {
        Id = 3,
        Nome = "Informática"
    },
    new Categoria() {
        Id = 4,
        Nome = "Smartphones"
    },
    new Categoria() {
        Id = 5,
        Nome = "Eletrodomesticos e Casa"
    },
    new Categoria() {
        Id = 6,
        Nome = "Beleza e Perfumaria"
    },
    new Categoria() {
        Id = 7,
        Nome = "Móveis e Decoração"
    }
};
builder.Entity<Categoria>().HasData(categorias);
#endregion

```

```
#region Popular Produtos
List<Produto> produtos = new()
{
    new Produto(){
        Id = 1,
        Nome = "Livro - A Biblioteca Da Meia-Noite - 8º Ed.",
        Descricao = "A Biblioteca da Meia-Noite é um romance incrível que fala dos infinitos rumos que a vida pode tomar e da busca incessante pelo rumo certo.Aos 35 anos, Nora Seed é uma mulher cheia de talentos e poucas conquistas.",
        Preco = 34.99M,
        Estoque = 10,
        CategoriaId = 1,
        Foto = "/img/produtos/1.png"
    },
    new Produto(){
        Id = 2,
        Nome = "Box Trilogia Sombra E Ossos - 1ª Ed.",
        Descricao = "Entre no Grishaverso: visite, em Ravka, um mundo de magia e superstição, onde nem tudo é o que parece ser! Os três livros da aclamada Trilogia Sombra e ossos – que originou a série da Netflix –, agora reunidos em um box especial",
        Preco = 144.54M,
        Estoque = 23,
        CategoriaId = 1,
        Foto = "/img/produtos/2.png"
    },
    new Produto(){
        Id = 3,
        Nome = "Console Playstation 5 + God Of War Ragnarok",
        Descricao = "Play Has No Limits™ Jogar Não Tem Limites PlayStation®5. O console PS5™ oferece novas possibilidades de jogabilidade que você nunca imaginou. Reproduza jogos para PS5 e PS4 em Blu-ray Disc. Além disso, você também pode baixar jogos ...",
        Preco = 3699.99M,
        Estoque = 6,
        CategoriaId = 2,
        Foto = "/img/produtos/3.png"
    },
    new Produto(){
        Id = 4,
        Nome = "Cadeira Gamer Fortt Lípsia Vermelha - Cgf002-V",
        Descricao = "Cadeira Gamer Fortt Lípsia Vermelha - CGF002-V Seja para trabalhar, estudar ou para seu lazer, a cadeira tornou-se um item essencial para o dia a dia. Agora ela não só é mais um item de decoração da casa ou de uso esporádico",
        Preco = 221.11M,
        Estoque = 30,
        CategoriaId = 2,
        Foto = "/img/produtos/4.png"
    },
    new Produto(){
        Id = 5,
        Nome = "Notebook 2 Em 1 Positivo Duo C4128b-3",
```

```

        Descricao = "Notebook 2 Em 1 Positivo Duo C4128b-3 Intel Celeron Dual
Core Windows11 Home 11,6' - Cinza Escuro - Inclui Microsoft 365* Notebook 2 em 1
Positivo Duo C4128B-3 Intel Celeron Dual Core Windows11 Home 11,6' - Cinza Escuro",
        Preço = 1439.10M,
        Estoque = 17,
        CategoriaId = 3,
        Foto = "/img/produtos/5.png"
    },
    new Produto(){
        Id = 6,
        Nome = "Conjunto Mesa De Jantar 6 Lugares Milano Amêndoa Com Savana",
        Descricao = "Conjunto Mesa De Jantar 6 Lugares Milano Amêndoa Com Savana
Conjunto com 6 Cadeiras Milano Conjunto com 6 Cadeiras Milano da Poliman Moveis: nossos
modelos de cadeiras sao praticos e resistentes",
        Preço = 521.10M,
        Estoque = 4,
        CategoriaId = 7,
        Foto = "/img/produtos/6.png"
    },
    new Produto(){
        Id = 7,
        Nome = "Smartphone Motorola Moto G04 4g 128gb Tela 6.6' 4gb Ram Câmera
16mp Selfie 5mp - Grafite",
        Descricao = "O novo Moto G04 foi projetado para impressionar com cores
vibrantes, material superior e detalhes lindos. Beleza em alto nível! Feito com material
de padrão elevado, é fino e confortável ao toque.",
        Preço = 699M,
        Estoque = 40,
        CategoriaId = 4,
        Foto = "/img/produtos/7.png"
    },
    new Produto(){
        Id = 8,
        Nome = "Cervejeira Consul Mais 82 Litros Frost Free Czd12at Titanium -
220v",
        Descricao = "Com uma Cervejeira Consul Mais você não precisa mais se
preocupar se suas cervejas vão congelar ou se elas não vão ficar geladas na hora que o
pessoal chegar. Você pode mudar a configuração das prateleiras da sua Cervejeira",
        Preço = 2166.03M,
        Estoque = 4,
        CategoriaId = 5,
        Foto = "/img/produtos/8.png"
    }
};
builder.Entity<Produto>().HasData(produtos);
#endregion
}
}

```

OBS: Lembre-se que os texto das linhas de **Descrição dos Produtos**, devem estar em uma única linha.

13. Crie na pasta do projeto, uma pasta como nome **ViewModels**, e dentro uma classe com o nome **HomeVM**, usando o código abaixo:

```
using EtecShop.Models;

namespace EtecShop.ViewModels;

public class HomeVM
{
    public List<Categoria> Categorias { get; set; }
    public List<Produto> Produtos { get; set; }
}
```

14. Abra o arquivo **_ViewImports.cshtml** e faça a alteração abaixo:

```
@using EtecShop
@using EtecShop.Models
@using EtecShop.ViewModels
@addTagHelper *, Microsoft.AspNetCore.Mvc.TagHelpers
```

15. Abra o arquivo **site.css** e adicione:

No começo do arquivo:

```
@import url("https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-icons.min.css");
```

No final do arquivo:

```
.cursor-pointer {
    cursor: pointer;
}
```

16. Abra o arquivo **HomeController.cshtml** e faça as alterações abaixo:

```
using System.Diagnostics;
using Microsoft.AspNetCore.Mvc;
using EtecShop.Models;
using EtecShop.Data;
using EtecShop.ViewModels;
using Microsoft.EntityFrameworkCore;

namespace EtecShop.Controllers;

public class HomeController : Controller
{
    private readonly ILogger<HomeController> _logger;
    private readonly ApplicationDbContext _context;

    public HomeController(
        ILogger<HomeController> logger,
```

```

        AppDbContext context)
    {
        _logger = logger;
        _context = context;
    }

```

```

public IActionResult Index()
{
    HomeVM homeVM = new(){
        Categorias = _context.Categorias.ToList(),
        Produtos = _context.Produtos.Include(p => p.Categoria).ToList()
    };
    return View(homeVM);
}

```

```

public IActionResult Produto(int id)
{
    Produto produto = _context.Produtos
        .Include(p => p.Categoria)
        .FirstOrDefault(p => p.Id == id);
    return View(produto);
}

```

```

public IActionResult Privacy()
{
    return View();
}

```

```

[ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]
public IActionResult Error()
{
    return View(new ErrorViewModel { RequestId = Activity
        .Current?.Id ?? HttpContext.TraceIdentifier });
}
}

```

17. Abra o arquivo **Index.cshtml** e faça as alterações abaixo:

```

@model HomeVM
@{
    ViewData["Title"] = "Home";
}

<section class="py-5 text-center container">
    <div class="row py-lg-2">
        <div class="col-lg-6 col-md-8 mx-auto">
            <h1 class="fw-light">EtecShop</h1>
            <p class="lead text-muted">Seu ponto de compras na internet.</p>
        </div>
    </div>
</section>

```

```

<div class="album py-5 bg-light">
  <div class="container">

    <div class="group text-center mb-5">
      <a class="btn btn-secondary" asp-action="Produtos">
        Todas as Categorias
      </a>
      @foreach (var categoria in Model.Categorias)
      {
        <a class="btn btn-secondary" asp-action="Produtos"
          asp-route-id="@categoria.Id">@categoria.Nome</a>
      }
    </div>

    <div class="row row-cols-1 row-cols-sm-2 row-cols-md-3 row-cols-lg-4 g-3">

      @foreach (var produto in Model.Produtos)
      {
        <div class="col cursor-pointer" onclick="window.location=
          '@Url.Action("Produto", "Home", new{ id = produto.Id})' ">
          <div class="card shadow-sm py-4 pb-0">
            
            <div class="card-body px-4">
              <p class="card-text fw-bold lh-1">@produto.Nome</p>
              <div class="d-flex justify-content-between align-items-center">
                <label class="fs-4 fw-bolder">R$ @produto.Preco.ToString("N2")</label>
              </div>
            </div>
          </div>
        </div>
      }
    </div>
  </div>
</div>

</main>

<footer class="text-muted py-3">
  <div class="container pb-3">
    <p class="float-end mb-1">
      <a href="#"><i class="bi bi-arrow-up-circle fs-5"></i></a>
    </p>
    <p class="mb-0">Para maiores informações entre em contato com nossa Loja.</p>
  </div>
</footer>

```

18. Salve todos os arquivos, execute o programa com o comando abaixo, e chame o professor para conferir seu trabalho até aqui:

dotnet watch run

19. Crie na pasta **Views\Home** uma arquivo com o nome **Produto.cshtml** e inclua o código abaixo:

```
@model Produto
@{
    ViewData["Title"] = "Produto";
}

<section class="py-5 text-center container">
    <div class="row py-lg-2">
        <div class="col-lg-6 col-md-8 mx-auto">
            <h1 class="fw-light">EtecShop</h1>
            <p class="lead text-muted">Seu ponto de compras na internet.</p>
        </div>
    </div>
</section>

<div class="album py-5 bg-light">
    <div class="container">
        <div class="row">
            <div class="col-md-5">
                
            </div>
            <div class="col-md-7">
                <h2 class="display-6">@Model.Nome</h2>
                <p class="text-muted">@Model.Categoria.Nome</p>
                <p class="mt-5">@Model.Descricao</p>
                <p class="fs-3 fw-bolder">R$ @Model.Preco.ToString("N2")</p>
            </div>
        </div>
    </div>
</div>
```

20. Salve todos os arquivos, execute o programa com o comando abaixo, e chame o professor para conferir seu trabalho até aqui:

dotnet watch run

21. Crie a pasta **ViewModels** uma classe com o nome **ProdutoVM**, e adicione o código abaixo:

```
using System.ComponentModel.DataAnnotations;
using EtecShop.Models;

namespace EtecShop.ViewModels;

public class ProdutoVM
{
    public Produto Produto { get; set; }

    public List<Avaliacao> Avaliacoes { get; set; }

    [Required(ErrorMessage = "Informe o Nome")]
    [StringLength(60, ErrorMessage = "O Nome deve possuir até 60 caracteres")]
    public string Nome { get; set; }
}
```

```

[Required(ErrorMessage = "Informe o Título")]
[StringLength(60, ErrorMessage = "O Título deve possuir até 60 caracteres")]
public string Titulo { get; set; }

[Required(ErrorMessage = "Escreva sua Avaliação")]
[StringLength(300, ErrorMessage = "A Avaliação de possuir no máximo 300 caracteres")]
public string Texto { get; set; }
}

```

22. Volte ao **HomeController.cs** e faça as alterações abaixo:

```

using System.Diagnostics;
using Microsoft.AspNetCore.Mvc;
using EtecShop.Models;
using EtecShop.Data;
using EtecShop.ViewModels;
using Microsoft.EntityFrameworkCore;

namespace EtecShop.Controllers;

public class HomeController : Controller
{
    private readonly ILogger<HomeController> _logger;
    private readonly ApplicationDbContext _context;

    public HomeController(
        ILogger<HomeController> logger,
        ApplicationDbContext context)
    {
        _logger = logger;
        _context = context;
    }

    public IActionResult Index()
    {
        HomeVM homeVM = new()
        {
            Categorias = _context.Categorias.ToList(),
            Produtos = _context.Produtos.Include(p => p.Categoria).ToList()
        };
        return View(homeVM);
    }

    public IActionResult Produto(int id)
    {
        Produto produto = _context.Produtos
            .Include(p => p.Categoria)
            .FirstOrDefault(p => p.Id == id);

        var avaliacoes = _context.Avaliacoes
            .Where(a => a.ProdutoId == produto.Id).ToList();
    }
}

```

```

        ProdutoVM produtoVM = new()
        {
            Produto = produto,
            Avaliacoes = avaliacoes
        };
        return View(produtoVM);
    }

```

```

[HttpPost]
[ValidateAntiForgeryToken]
public IActionResult Produto(ProdutoVM produtoAvaliado)
{
    if (produtoAvaliado.Nome == string.Empty ||
        produtoAvaliado.Texto == string.Empty ||
        produtoAvaliado.Titulo == string.Empty)
    {
        Produto produto = _context.Produtos
            .Include(p => p.Categoria)
            .FirstOrDefault(p => p.Id == produtoAvaliado.Produto.Id);
        var avaliacoes = _context.Avaliacoes
            .Where(a => a.ProdutoId == produto.Id).ToList();

        ProdutoVM produtoVM = new()
        {
            Produto = produto,
            Avaliacoes = avaliacoes
        };
        ModelState.AddModelError(string.Empty, "Informe todos os campos");
        return View(produtoVM);
    }
    Avaliacao avaliacao = new() {
        Nome = produtoAvaliado.Nome,
        Texto = produtoAvaliado.Texto,
        Titulo = produtoAvaliado.Titulo,
        ProdutoId = produtoAvaliado.Produto.Id
    };
    _context.Add(avaliacao);
    _context.SaveChanges();
    TempData["Avaliacao"] = "Obrigado por seu comentário";
    return RedirectToAction(nameof(Produto));
}

```

```

public IActionResult Privacy()
{
    return View();
}

```

```

[ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]
public IActionResult Error()
{
    return View(new ErrorViewModel
    {

```

```

        RequestId = Activity
            .Current?.Id ?? HttpContext.TraceIdentifier
    });
}
}

```

23. Volte ao arquivo Produto.cshtml e faça as alterações abaixo:

```

@model ProdutoVM
@{
    ViewData["Title"] = "Produto";
}

<section class="py-5 text-center container">
    <div class="row py-lg-2">
        <div class="col-lg-6 col-md-8 mx-auto">
            <h1 class="fw-light">EtecShop</h1>
            <p class="lead text-muted">Seu ponto de compras na internet.</p>
        </div>
    </div>
</section>

<div class="album py-5 bg-light">
    <div class="container">
        <div class="row">
            <div class="col-md-5">
                
            </div>
            <div class="col-md-7">
                <h2 class="display-6">@Model.Produto.Nome</h2>
                <p class="text-muted">@Model.Produto.Categoria.Nome</p>
                <p class="mt-5">@Model.Produto.Descricao</p>
                <p class="fs-3 fw-bolder">R$ @Model.Produto.Preco.ToString("N2")</p>
            </div>
        </div>
        <hr class="my-5">
        <div class="row">
            <div class="col">
                <h4 class="mb-4">Comentários sobre o Produto</h4>

                @if (Model.Avaliacoes.Count == 0)
                {
                    <div class="card rounded-3 mt-4">
                        <h5 class="text-center my-4">Este produto ainda não foi avaliado, seja o primeiro!</h5>
                    </div>
                }

                @foreach (var avaliacao in Model.Avaliacoes)
                {
                    <div class="card rounded-3 mt-4">
                        <h5 class="card-header">@avaliacao.Titulo</h5>

```



```
@section Scripts {  
  <partial name="_ValidationScriptsPartial"></partial>  
  <script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>  
  <script>  
    let avaliado = '@Html.Raw(TempData["Avaliacao"])';  
    if (avaliado != '') {  
      window.scrollTo(0, document.body.scrollHeight);  
      Swal.fire({  
        title: 'Avaliação de Produto',  
        text: avaliado,  
        icon: 'success'  
      })  
    }  
  </script>  
}
```

24. Salve todos os arquivos, execute o programa com o comando abaixo, e chame o professor para conferir seu trabalho até aqui:

dotnet watch run