INSTITUTO INFNET

ESCOLA SUPERIOR DE TECNOLOGIA

GRADUAÇÃO EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

TP1

Aluno: Enzo Furtini

12 de jun. de 2025

Exercícios de C# - Delegates, Events e ASP.NET Core

Este repositório contém as soluções para os exercícios propostos sobre Delegates, Events e ASP.NET Core.

Estrutura do Projeto

O projeto está organizado em duas partes principais:

1. Console Applications (Exercícios 1-7)

2. ASP.NET Core Web Application (Exercícios 8-12)

Exercício 1: Implementação de Delegate Personalizado para Descontos

Como implementar um delegate personalizado para calcular descontos em produtos?

```
public delegate decimal CalcularDesconto(decimal precoOriginal);

public class CalculadoraDesconto
{
    public static decimal AplicarDesconto(decimal preco)
    {
        return preco * 0.9m; // 10% de desconto
    }
}
```

Exercício 2: Ações Multilíngues com Action<string>

Como implementar saudações em diferentes idiomas usando Action<string>?

```
public class Saudador
{
    public static void SaudacaoPortugues(string nome) => Console.WriteLine($"Olá,
{nome}!");
    public static void SaudacaoIngles(string nome) => Console.WriteLine($"Hello, {nome}!");
    public static void SaudacaoEspanhol(string nome) => Console.WriteLine($"¡Hola,
{nome}!");
}
```

Exercício 3: Cálculo de Área Utilizando Func

Como implementar o cálculo de área de um retângulo usando Func?

```
public class CalculadoraArea
{
    public static double CalcularAreaRetangulo(double largura, double altura)
    {
        return largura * altura;
    }
}
```

Exercício 4: Monitoramento de Temperatura com Evento Personalizado

Como implementar um sistema de monitoramento de temperatura com eventos?

```
public class ArgumentosTemperatura : EventArgs
{
    public double Temperatura { get; }
    public ArgumentosTemperatura(double temperatura)
```

```
    {
        Temperatura = temperatura;
    }
}

public class SensorTemperatura
{
    public event EventHandler<ArgumentosTemperatura> TemperaturaExcedida;

    protected virtual void OnTemperaturaExcedida(ArgumentosTemperatura e)
    {
        TemperaturaExcedida?.Invoke(this, e);
    }

    public void VerificarTemperatura(double temperatura)
    {
        if (temperatura > 100)
        {
            OnTemperaturaExcedida(new ArgumentosTemperatura(temperatura));
        }
    }
}
```

Exercício 5: Notificação de Conclusão de Download com Eventos

Como implementar um sistema de notificação de download usando eventos?

```
public class ArgumentosDownload : EventArgs
{
    public string NomeArquivo { get; }
    public ArgumentosDownload(string nomeArquivo)
    {
        NomeArquivo = nomeArquivo;
    }
}

public class GerenciadorDownload
{
    public event EventHandler<ArgumentosDownload> DownloadConcluido;
```

```
    protected virtual void OnDownloadConcluido(ArgumentosDownload e)
    {
        DownloadConcluido?.Invoke(this, e);
    }

    public async Task BaixarArquivoAsync(string nomeArquivo)
    {
        await Task.Delay(2000); // Simula download
        OnDownloadConcluido(new ArgumentosDownload(nomeArquivo));
    }
}
```

Exercício 6: Sistema de Registro com Multicast Delegate

Como implementar um sistema de registro que salva em múltiplos destinos?

```
public class Registrador
{
    public void RegistrarNoConsole(string mensagem) => Console.WriteLine($"Console: 
{mensagem}");
    public void RegistrarNoArquivo(string mensagem) => File.AppendAllText("registro.txt", 
$"Arquivo: {mensagem}\n");
    public void RegistrarNoBanco(string mensagem) => Console.WriteLine($"Banco de 
Dados: {mensagem}");
}
```

Exercício 7: Garantia de Robustez em Invocação de Delegates

Como garantir que a invocação de delegates seja segura?

```
public class RegistradorSeguro
{
    private Action<string> _delegateRegistro;

    public void Registrar(string mensagem)
```

```
    {
        _delegateRegistro?.Invoke(mensagem);
    }
}
```

Exercício 8: Construção de Aplicação Web com Razor Pages

Como criar uma página inicial com lista de produtos?

```
// Index.cshtml
@page
@model IndexModel
@{
    ViewData["Title"] = "Página Inicial";
}

<div class="text-center">
    <h1 class="display-4">Catálogo de Produtos</h1>

    <div class="row mt-4">
        @foreach (var produto in Model.Produtos)
        {
            <div class="col-md-4 mb-4">
                <div class="card">
                    <div class="card-body">
                        <h5 class="card-title">@produto.Nome</h5>
                        <p class="card-text">Preço: @produto.Preco.ToString("C")</p>
                        <p class="card-text"><small class="text-muted">Criado em:
@produto.DataCriacao.ToString("dd/MM/yyyy")</small></p>
                    </div>
                </div>
            </div>
        }
    </div>
</div>
```

Exercício 9: Exploração da Estrutura de Projeto ASP.NET Core

Como é a estrutura básica de um projeto ASP.NET Core?

```
// Program.cs
var builder = WebApplication.CreateBuilder(args);
builder.Services.AddRazorPages();
var app = builder.Build();
app.UseStaticFiles();
app.UseRouting();
app.MapRazorPages();
app.Run();
```

Exercício 10: Implementação de Formulário em Razor Pages

Como implementar um formulário para adicionar produtos?

```
// AdicionarProduto.cshtml
@page
@model AdicionarProdutoModel
@{
    ViewData["Title"] = "Adicionar Produto";
}

<div class="container">
    <h2 class="mb-4">Adicionar Novo Produto</h2>
    <form method="post">
        <div class="form-group">
            <label asp-for="Produto.Nome">Nome do Produto</label>
            <input asp-for="Produto.Nome" class="form-control" />
            <span asp-validation-for="Produto.Nome" class="text-danger"></span>
        </div>
        <div class="form-group">
            <label asp-for="Produto.Preco">Preço</label>
            <input asp-for="Produto.Preco" type="number" step="any" class="form-control" />
            <span asp-validation-for="Produto.Preco" class="text-danger"></span>
```

```
        </div>
        <button type="submit" class="btn btn-primary mt-3">Adicionar Produto</button>
    </form>
</div>
```

Exercício 11: Manipulação de Strings com Delegates Encadeados

Como implementar transformações encadeadas em strings?

```
Func<string, string> transformar = s => s.ToUpper();
transformar += s => s.Trim();
transformar += s => s.Replace(" ", "");
```

Exercício 12: Integração de Delegates e Eventos em Aplicação Web

Como integrar eventos com uma aplicação web?

```
public class Evento
{
    public int Id { get; set; }
    public string Titulo { get; set; }
    public DateTime Data { get; set; }
    public string Local { get; set; }
}

public class EventosModel : PageModel
{
    [BindProperty]
    public Evento Evento { get; set; }

    public IActionResult OnPost()
    {
```

```
        if (!ModelState.IsValid)
        {
            return Page();
        }
        Console.WriteLine($"Novo evento cadastrado: {Evento.Titulo}");
        return Page();
    }
}
```

Como Executar

1. Para os exercícios de console:

- Navegue até a pasta do exercício
- Execute `dotnet run`

2. Para a aplicação web:

- Navegue até a pasta do projeto web
- Execute `dotnet run`
- Acesse `https://localhost:5001` ou `http://localhost:5000`