PROJETO POKEDEX

Prof. Ms. José Antonio Gallo Junior

1. Criar um Repositório com o nome **Pokedex** no seu **GitHub.**

2. Clonar o Repositório **Pokedex** na **Área de Trabalho**.

3. Abrir a pasta do Repositório **Pokedex** no **Visual Studio Code**.

4. Abrir o terminal, conferir se está na pasta do Repositório e executar os comandos abaixo:

```
dotnet new sln --name Pokedex
dotnet new mvc -o Pokedex
dotnet sln add Pokedex\Pokedex.csproj
```

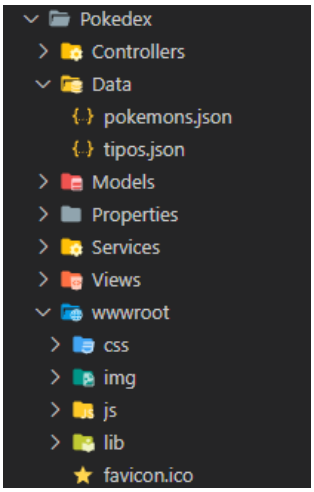
5. No terminal use o comando abaixo para acessar a pasta do projeto **Pokedex:**

```
cd Pokedex
```

6. Criar dentro da pasta do projeto **Pokedex** as pastas, Data e Services; isso pode ser feito com os comandos abaixo ou com os recursos do VS Code:

```
mkdir Data
mkdir Services
```

7. Clique aqui e faça o download da pasta **PokeFiles**, que possui os arquivos necessários ao desenvolvimento do projeto.



8. Abra a pasta **PokeFiles**, e copie as pastas **css** e **img** para dentro da pasta **wwwroot** do seu projeto; substitua a pasta **css** existente.

9. Abra a pasta **PokeFiles** e copie os arquivos **pokemons.json** e **tipos.json** da pasta **dados** para a pasta **Data** do seu projeto.

OBS: Na imagem ao lado, pode ser observado a estrutura atual da **wwwroot** e os arquivos adicionados a pasta **Data**, conforme indicado nos passos 8 e 9.

10. Abra o arquivo **Pokedex.csproj** e altere a notação **Nullable** de **enable** para **disable**, conforme código abaixo:

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

11. Crie uma **classe** com o nome **Tipo** na pasta **Models** e faça as alterações abaixo:

OBS: Para criar uma classe, clique com o botão direito do mouse sobre a pasta onde quer criar a classe e selecione a opção **New C#** e **Class.**

```
namespace Pokedex.Models;

public class Tipo
{
    // Atributos
    public string Nome { get; set; }
    public string Cor { get; set; }
}
```

12. Crie uma **classe** com o nome **Pokemon** na pasta **Models** e faça as alterações abaixo:

```
namespace Pokedex.Models;

public class Pokemon
{
    // Atributos
    public int Numero { get; set; }
    public string Nome { get; set; }
    public string Descricao { get; set; }
    public string Especie { get; set; }
    public List<string> Tipo { get; set; }
    public double Altura { get; set; }
    public double Peso { get; set; }
    public string Imagem { get; set; }

    // Método Construtor
    public Pokemon()
    {
        Tipo = new List<string>();
    }
}
```

13. Crie uma **classe** com o nome **DetailsDto** na pasta **Models** e faça as alterações abaixo:

```
namespace Pokedex.Models;

public class DetailsDto
{
    public Pokemon Prior { get; set; }
    public Pokemon Current { get; set; }
    public Pokemon Next { get; set; }
}
```

14. Crie uma **classe** com o nome **PokedexDto** na pasta **Models** e faça as alterações abaixo:

```
namespace Pokedex.Models;

public class PokedexDto
{
    public List<Tipo> Tipos { get; set; }
    public List<Pokemon> Pokemons { get; set; }
}
```

15. Crie uma **interface** na pasta **Services** com o nome **IPokeService** e faça as alterações abaixo:

OBS: Para criar uma interface, clique com o botão direito do mouse sobre a pasta onde quer criar a classe e selecione a opção **New C#** e **Interface.**

```
using Pokedex.Models;
namespace Pokedex.Services;

public interface IPokeService
{
   List<Pokemon> GetPokemons();
   List<Tipo> GetTipos();
   Pokemon GetPokemon(int Numero);
   PokedexDto GetPokedexDto();
   DetailsDto GetDetailedPokemon(int Numero);
   Tipo GetTipo(string Nome);
}
```

16. Crie uma **classe** na pasta **Services** com o nome **PokeService** e faça as alterações abaixo:

```
using System.Text.Json;
using Pokedex.Models;

namespace Pokedex.Services;

public class PokeService : IPokeService
{
   private readonly IHttpContextAccessor _session;
   private readonly string pokemonFile = @"Data\pokemons.json";
   private readonly string tiposFile = @"Data\tipos.json";

   public PokeService(IHttpContextAccessor session)
   {
       _session = session;
       PopularSessao();
   }

   public List<Pokemon> GetPokemons()
   {
       PopularSessao();
       var pokemons = JsonSerializer.Deserialize<List<Pokemon>>
           (_session.HttpContext.Session.GetString("Pokemons"));
       return pokemons;
   }

   public List<Tipo> GetTipos()
   {
       PopularSessao();
       var tipos = JsonSerializer.Deserialize<List<Tipo>>
           (_session.HttpContext.Session.GetString("Tipos"));
       return tipos;
   }

   public Pokemon GetPokemon(int Numero)
   {
       var pokemons = GetPokemons();
       return pokemons.Where(p => p.Numero == Numero).FirstOrDefault();
   }

   public PokedexDto GetPokedexDto()
   {
       var pokes = new PokedexDto()
       {
           Pokemons = GetPokemons(),
           Tipos = GetTipos()
       };
       return pokes;
   }
```

```
    public DetailsDto GetDetailedPokemon(int Numero)
    {
        var pokemons = GetPokemons();
        var poke = new DetailsDto()
        {
            Current = pokemons.Where(p => p.Numero == Numero)
                .FirstOrDefault(),
            Prior = pokemons.OrderByDescending(p => p.Numero)
                .FirstOrDefault(p => p.Numero < Numero),
            Next = pokemons.OrderBy(p => p.Numero)
                .FirstOrDefault(p => p.Numero > Numero),
        };
        return poke;
    }

    public Tipo GetTipo(string Nome)
    {
        var tipos = GetTipos();
        return tipos.Where(t => t.Nome == Nome).FirstOrDefault();
    }

    private void PopularSessao()
    {
        if (string.IsNullOrEmpty(_session.HttpContext.Session.GetString("Tipos")))
        {
            _session.HttpContext.Session
                .SetString("Pokemons", LerArquivo(pokemonFile));
            _session.HttpContext.Session
                .SetString("Tipos", LerArquivo(tiposFile));
        }
    }

    private string LerArquivo(string fileName)
    {
        using (StreamReader leitor = new StreamReader(fileName))
        {
            string dados = leitor.ReadToEnd();
            return dados;
        }
    }
}
```

17. Altere o código do arquivo **Program.cs** de acordo com o código abaixo:

```
using Pokedex.Services;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.
builder.Services.AddControllersWithViews();

builder.Services.AddSession();
builder.Services.AddHttpContextAccessor();
builder.Services.AddSingleton<IPokeService, PokeService>();

var app = builder.Build();
```

```
// 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 .....
    app.UseHsts();
}

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

app.UseRouting();

app.UseAuthorization();

app.UseSession();

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

app.Run();
```

18. Altere o código do arquivo **Controllers\HomeController.cs** de acordo com o código abaixo:

```
using System.Diagnostics;
using Microsoft.AspNetCore.Mvc;
using Pokedex.Models;
using Pokedex.Services;

namespace Pokedex.Controllers;

public class HomeController : Controller
{
    private readonly ILogger<HomeController> _logger;
    private readonly IPokeService _pokeService;

    public HomeController(ILogger<HomeController> logger, IPokeService pokeService)
    {
        _logger = logger;
        _pokeService = pokeService;
    }

    public IActionResult Index(string tipo)
    {
        var pokes = _pokeService.GetPokedexDto();
        ViewData["filter"] = string.IsNullOrEmpty(tipo) ? "all" : tipo;
        return View(pokes);
    }

    public IActionResult Details(int Numero)
    {
        var pokemon = _pokeService.GetDetailedPokemon(Numero);
        return View(pokemon);
    }

    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 });
    }
}
```

19. Altere o código do arquivo **Views\Shared_Layout.cshtml** de acordo com o código abaixo:

OBS: Na pasta **PokeFiles** baixada no começo deste projeto, existem dois arquivos **html**, **index** e **details**, parte do seu código pode ser utilizado para a criação do layout, conforme será demonstrado em aula. Desta forma o código **cshtml** abaixo, é semelhante ao código do arquivo **index**, que pode ser copiado e alterado**.**

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Pokedex | @ViewData["Title"]</title>

    <!-- Favicons -->
    <link rel="apple-touch-icon" href="~/img/favicons/apple-touch-icon.png" sizes="180x180">
    <link rel="icon" href="~/img/favicons/favicon-32x32.png" sizes="32x32" type="image/png">
    <link rel="icon" href="~/img/favicons/favicon-16x16.png" sizes="16x16" type="image/png">
    <link rel="icon" href="~/img/favicons/favicon.ico">

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css"
rel="stylesheet"
        integrity="sha384-gH2yIJqKdNHPEq0n4Mqa/HGKIhSkIHeL5AyhkYV8i59U5AR6csBvApHHNl/vI1Bx"
crossorigin="anonymous">
    <link rel="stylesheet" href="~/css/site.css" />

</head>

<body>
    <header>
        <div class="collapse bg-danger" id="navbarHeader">
            <div class="container">
                <div class="row">
                    <div class="col-sm-8 col-md-7 py-4">
                        <h4 class="text-white">Sobre</h4>
                        <p class="text-white">
                            Projeto desenvolvido para demonstrar a criação de páginas WEB com
Net 6.0.
                            Demonstrar a criação de aplicações com padrão MVC e leitura de
arquivos JSON,
                            além de boas práticas de Programação Orientada a Objetos, Páginas
Dinâmicas e Uso de Sessão.
                        </p>
                    </div>
                    <div class="col-sm-4 offset-md-1 py-4">
                        <h4 class="text-white">Contatos</h4>
                        <ul class="list-unstyled">
                            <li><a href="#" class="text-white">Follow on Twitter</a></li>
                            <li><a href="#" class="text-white">Like on Facebook</a></li>
                            <li><a href="#" class="text-white">E-mail</a></li>
                        </ul>
                    </div>
                </div>
            </div>
```

```
        </div>
        <div class="navbar navbar-dark bg-danger shadow-sm">
            <div class="container">
                <a asp-action="Index" class="navbar-brand d-flex align-items-center">
                    <img src="~/img/favicons/favicon-32x32.png" alt="Pokeball" class="me-2"
/>
                     <strong>POKÉDEX</strong>
                </a>
                <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-
bs-target="#navbarHeader"
                    aria-controls="navbarHeader" aria-expanded="false" aria-label="Toggle
navigation">
                    <span class="navbar-toggler-icon"></span>
                </button>
            </div>
        </div>
    </header>

    <main>
        @RenderBody()
    </main>

    <footer class="text-muted py-3 ">
        <div class="container">
            <p class="float-end">
                <a href="#">
                    <svg xmlns="http://www.w3.org/2000/svg" width="32" height="32"
fill="currentColor"
                        class="bi bi-arrow-up-circle-fill text-danger" viewBox="0 0 16 16">
                        <path
                            d="M16 8A8 8 0 1 0 0 8a8 8 0 0 0 16 0zm-7.5 3.5a.5.5 0 0 1-1
0V5.707L5.354 7.854a.5.5 0 1 1-.708-.708l3-3a.5.5 0 0 1 .708 0l3 3a.5.5 0 0 1-.708.708L8.5
5.707V11.5z" />
                    </svg>
                </a>
            </p>
            <p class="mb-1">Pokédex - Versão Gallo & Márcio</p>
            <p class="mb-0">Projeto Integrador desenvolvido para as aulas de Interfaces Web
II e Sistemas Web I</p>
        </div>
    </footer>

    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/js/bootstrap.bundle.min.js"
        integrity="sha384-A3rJD856KowSb7dwlZdYEkO39Gagi7vIsF0jrRAoQmDKKtQBHUuLZ9AsSv4jD4Xa"
        crossorigin="anonymous"></script>
    <script src="~/lib/jquery/dist/jquery.min.js"></script>
    <script src="~/js/site.js" asp-append-version="true"></script>
    @await RenderSectionAsync("Scripts", required: false)
</body>

</html>
```

20. Altere o código do arquivo **Views\Home\Index.cshtml** de acordo com o código abaixo:

```
@model PokedexDto
@inject Pokedex.Services.IPokeService service

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

```
<section class="pt-5 pb-3 text-center container">
    <div class="row">
        <div class="col mx-auto">
            <h1>Pokédex</h1>
            <div class="group">
                <button id="btn-all" class="btn btn-lg my-2 text-white bg-secondary btn-
filter"
                    onclick="filter('all')">
                    Ver Todos
                </button>
                @foreach (var tipo in Model.Tipos)
                {
                    <button id="btn-@tipo.Nome.ToLower()" class="btn btn-sm my-2 text-white
btn-filter"
                        onclick="filter('@tipo.Nome.ToLower()')"
                        style="background-color:@tipo.Cor">
                        @tipo.Nome
                    </button>
                }
            </div>
        </div>
    </div>
</section>

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

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

            <!-- Card Pokemon - Inicio -->
            @foreach (var pokemon in Model.Pokemons)
            {
                <div class="col">
                    <div class="card shadow-sm cursor-pointer @string.Join(' ',
pokemon.Tipo).ToLower()"
                        onclick="GetInfo(@pokemon.Numero)">
                        <img src="@pokemon.Imagem" class="card-img-top" width="100%"
height="300" alt="@pokemon.Nome" />
                        <div class="card-body">
                            <p class="card-text mb-0">Nº @pokemon.Numero.ToString("000")</p>
                            <h3 class="card-title">@pokemon.Nome</h3>
                            <div class="d-flex justify-content-between align-items-center">
                                <div class="btn-group">
                                    @foreach (var tipo in pokemon.Tipo)
                                    {
                                        <a asp-action="Index" asp-route-
tipo="@tipo.ToLower()" class="btn my-2 text-white"
                                            style="background-
color:@service.GetTipo(tipo).Cor">@tipo</a>
                                    }
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            }
            <!-- Card Pokemon - Fim -->

            <div id="zeroPokemon" class="col-lg-12 text-center">
                <h1 class="my-3 text-danger">Nenhum Pokemon Encontrado</h1>
            </div>
        </div>
```

```
    </div>
</div>


@section Scripts{
    <script src="~/js/site.js"></script>
    <script>
        let toFilter = '@Html.Raw(ViewData["Filter"])';
        filter(toFilter);

        function GetInfo(number) {
            window.location = '@Url.Action("Details", "Home")' + '?Numero=' + number;
        }
    </script>
}
```

21. Crie um arquivo com o nome **Details.cshtml**, na pasta **Views\Home** e insira o código abaixo:

```
@model DetailsDto
@inject Pokedex.Services.IPokeService service
@{
    ViewData["Title"] = "Pokemon";
}

<section class="pt-5 container">
    <div class="row">
        <div class="col text-center">
            @if (Model.Prior != null)
            {
                <a asp-action="Details" asp-route-Numero="@Model.Prior.Numero" class="btn
btn-lg btn-outline-dark">
                    @Model.Prior.Numero.ToString("000") - @Model.Prior.Nome
                </a>
            }
        </div>
        <div class="col text-center">
            <h1 class="fs-1">@Model.Current.Nome <span class="fs-2 text-secondary">Nº
                    @Model.Current.Numero.ToString("000")</span></h1>
        </div>
        <div class="col text-center">
            @if (Model.Next != null)
            {
                <a asp-action="Details" asp-route-Numero="@Model.Next.Numero" class="btn btn-
lg btn-outline-dark">
                    @Model.Next.Numero.ToString("000") - @Model.Next.Nome
                </a>
            }
        </div>
    </div>

    <div class="row mt-3">
        <div class="col-lg-5 p-0">
            <img class="img-fluid cursor-pointer" src="@Model.Current.Imagem"
alt="@Model.Current.Nome">
        </div>
        <div class="col-lg-7 fs-5 mt-4">
            <p class="mb-3"><strong>Descrição:</strong> @Model.Current.Descricao</p>
            <p class="mb-3"><strong>Categoria:</strong> @Model.Current.Especie</p>
            <p class="mb-3"><strong>Altura:</strong> @Model.Current.Altura mts</p>
            <p class="mb-3"><strong>Peso:</strong> @Model.Current.Peso kgs</p>
            <p class="fw-bold">Tipo</p>
```

```
            <div class="btn-group">
                @foreach (var tipo in Model.Current.Tipo)
                {
                    <a asp-action="Index" asp-route-tipo="@tipo.ToLower()" class="btn btn-lg
text-white"
                        style="background-color:@service.GetTipo(tipo).Cor">@tipo</a>
                }
            </div>
        </div>
    </div>
</section>
```

22. Altere o código do arquivo **wwwroot\js\site.js** de acordo com o código abaixo:

```
function filter(type) {
    let cards, i;
    let count = 0;
    cards = document.getElementsByClassName("card");
    buttons =  document.getElementsByClassName("btn-filter");
    for (i = 0; i < cards.length; i++) {
        cards[i].parentElement.style.display = 'none';
        if (cards[i].classList.contains(type) || type === "all") {
            cards[i].parentElement.style.display = 'block';
            count += 1;
        };
    };
    for (i = 0; i < buttons.length; i++) {
        if (buttons[i].id == `btn-${type}`) {
            buttons[i].classList.remove("btn-sm");
            buttons[i].classList.add("btn-lg");
        }
        else {
            buttons[i].classList.remove("btn-lg");
            buttons[i].classList.add("btn-sm");
        }
    };
    if (type === "all") {
        document.getElementById("btn-all").classList.remove("btn-sm");
        document.getElementById("btn-all").classList.add("btn-lg");
    };
    if (count == 0){
        document.getElementById("zeroPokemon").style.display = 'block';
    }
    else {
        document.getElementById("zeroPokemon").style.display = 'none';
    }
}
```

23. Pronto, agora basta executar o projeto, conferindo o terminal está na pasta **Pokedex\Pokedex**, execute o comando abaixo:

```
dotnet watch run
```

Clique nos botões dos tipos, nos cards que exibem os Pokemons, observe a página de detalhes, utilize os botões de navegação da página, clique nos tipos e confira todas as funcionalidades incluídas no projeto.