

Camada de Interfaces

Leandro Garcia



Interfaces



O que é?

- Interface é todo meio pelo qual se acessa serviços de um recurso de um sistema
 - Página Web
 - Software desktop
 - Aplicativo móvel
 - ChatBot
 - Coisas/Máquinas (IoT / M2M)
 - Outros sistemas (API)
 - Outras interfaces (Linha de comando, SMS, etc)



Importante...

- Grande importância do ponto de vista de negócio e de aceitação do usuário
- Camada definidora de percepção de qualidade
- No início dos projetos é comum desenhar protótipos para que o andamento do projeto ocorra de forma mais alinhada com usuários e clientes



Onde estamos?

- Armazenamento de dados

- Regra de negócio

- Barramento de Serviços

Interfaces

Sistemas Extern.

- ✓ Esta camada pode aparecer em topologias diferentes, dependendo da implementação



Pré-requisitos e características

- ✓ Camada de serviços bem definida é fundamental para múltiplas interfaces
- ✓ A interface não precisa (nem deve conhecer a tecnologia do restante do sistema)
- ✓ Frase: Se você consegue fazer uma interface em linha de comando para seu sistema significa que a arquitetura dele é boa



Interface Web com .NET

- ✓ ASP.NET padrão com páginas ASPX
- ✓ MVC – Modelo, visão e controle
- ✓ Em última instância para o usuário final é sempre HTML, JavaScript* e CSS



Interfaces de aplicativos móveis

- Baseadas em browser (Cross platform)
 - Cordova - IONIC
 - Aplicações Web rodando no Browser do dispositivo móvel
- Flutter e Dart
 - Flutter: framework móvel do Google que permite construir aplicativos Android e iOS (também Web e Desktop)
 - Dart: linguagem de programação
- Xamarin – Cross platform
 - Tenta maior proximidade com aplicações nativas
 - Usa Visual Studio



Interfaces de aplicativos móveis

■ Unity

- Plataforma específica para criação de Jogos e softwares que utilizem animações em 2D e 3D
- IDE específica para produção

■ Aplicativos nativos

- Android
- IOS
- Benefícios: melhor acesso ao Hardware

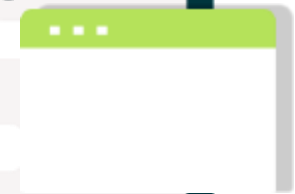


Abstração das abordagens não nativas

Desenvolvimento

Tradução ou
Abstração

Aplicativo
Específico



Interfaces



Arquivos ASPX



Conceito inicial

- Simular desenvolvimento mais próximo de Desktop: Estado dos componentes, Eventos, Ciclo de vida da página, etc
- Abstrair complexidades do desenvolvimento Web

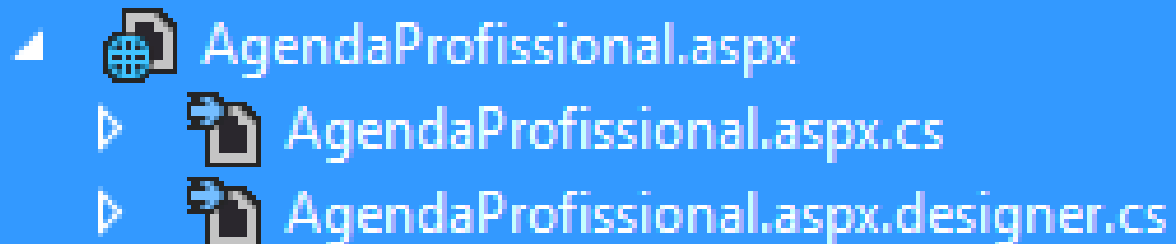


Cada arquivo é uma página

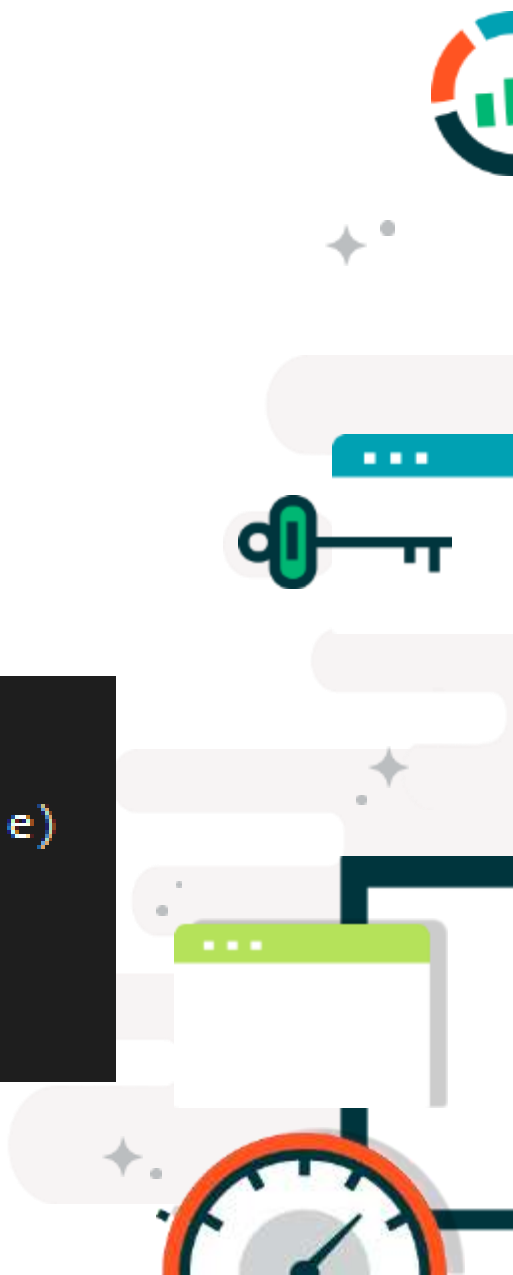
- Cada arquivo ASPX criado é uma página da sua interface Web
- Automaticamente cria-se a classe relacionada a página ASPX.cs e ASPX.designer.cs
- A URL referencia diretamente a página física (arquivo com extensão ASPX)



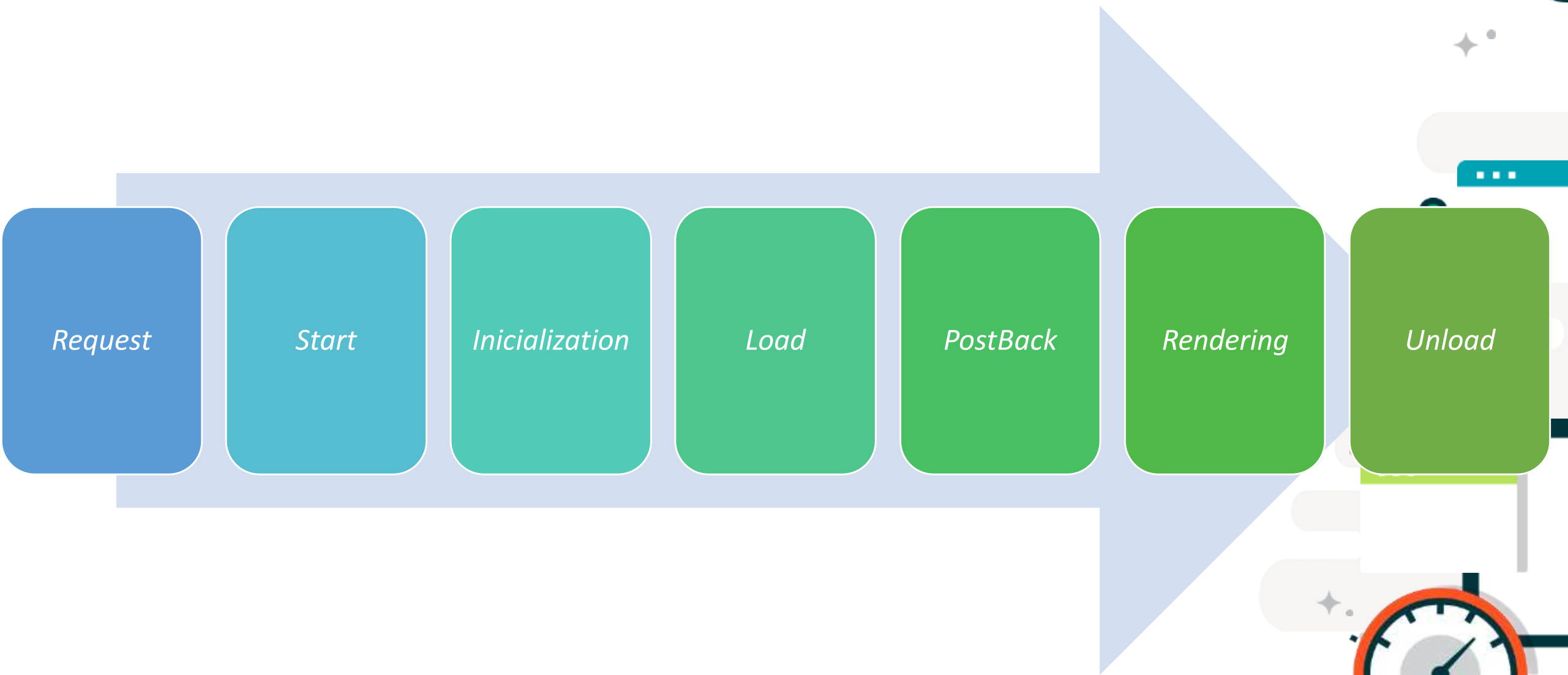
Olhando para uma página



```
public partial class AgendaProfissional : PageBase
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
}
```



Ciclo de vida da página



Olhando para a construção de interface

```
<div>
  <div style="float: left; margin-right: 30px;">
    <dx:ASPxMenu ID="tbBuscar" runat="server" OnItemClick="tbBuscar_ItemClick">
      <Items>
        <dx:MenuItem Text="Ajuda" ToolTip="Ajuda" Image-Url="~/App/CLI/Images/ImgHelp22.png" Name="Ajuda">
        </dx:MenuItem>
        <dx:MenuItem Text="Atualizar Agenda" ToolTip="Atualizar Agenda" Image-Url="~/App/CLI/Images/ImgRefresh22.gif" Name="Atualizar">
        </dx:MenuItem>
        <dx:MenuItem Text="Ver Horários do Profissional" ToolTip="Ver Horários do Profissional" Image-Url="~/App/CLI/Images/atendimentoS...
        </dx:MenuItem>
        <dx:MenuItem Text="Imprimir Agenda do Dia" ToolTip="Imprimir Agenda do Dia" Image-Url="~/App/CLI/Images/atendimentoSmall.png" Name="
        </dx:MenuItem>
        <dx:MenuItem Text="Fechar" ToolTip="Fechar" Image-Url="~/App/CLI/Images/ImgClose22.gif" Name="Fechar">
        </dx:MenuItem>
      </Items>
    </dx:ASPxMenu>
  </div>
  <div class="">
    <asp:Label runat="server" ID="lblSelecione" Text="Selecione o Profissional:" ></asp:Label>
    <asp:DropDownList ID="ddlProfissional" AutoPostBack="true" runat="server" CssClass="classDdlProfissional large" ActionId="ActionProf...
      OnSelectedIndexChanged="ddlProfissional_OnSelectedIndexChanged"></asp:DropDownList>
    <span title="Para fazer agendamentos, basta clicar com o botão direito do mouse encima do dia e horário que deseja reservar" class=...

    <asp:HiddenField ID="txtIdProfissional" runat="server"></asp:HiddenField>
    <asp:ImageButton ID="btnAtualizar" Style="visibility: hidden;" runat="server" AlternateText="Atualizar"
      OnClick="btnAtualizar_Click" />
  </div>
</div>
```

ASCX ou Master Page

- MasterPage é uma página especial que pode conter funcionalidades comuns entre as páginas que a implementam: Cabeçalho, rodapé, Autenticação, Carrinho de compras, etc
- User Controls (ASCX) é uma subpágina, funcionalidade padrão que pode ser repetir em mais de uma pagina



Aspectos Positivos

- Facilidade de entendimento
- Velocidade de produção
- Menor preocupação com renderização
- Abstração de pontos mais complexos
 - Manter status dos componentes de forma automática
 - Gestão de eventos simples: ao mudar o texto, ao clicar, ao selecionar
 - Chamadas e requisições simplificadas: Ajax ou PostBack



Aspectos Negativos

- Menor domínio sobre a renderização
 - Devido a abstração
 - Componentes gerados automaticamente
- Alto custo de recarga das páginas
 - PostBack com ViewState
- Maior custo de execução e processamento



Modelo MVC



Conceito Inicial

- Dar opção de maior liberdade e menor abstração
- Menor acoplamento da interface do restante do Software



Abordagem diferente

- Alinhamento com arquiteturas de mercado
- Regras de roteamento definem qual URL no browser vai acionar qual Controller
- Cada requisição feita cai em um Controller, que usa o Model para buscar (ou salvar) dados e escolhe uma View para exibir a página para o usuário



Regra básica de roteamento (Startup.cs)

```
public void Configure(IApplicationBuilder app, IHostingEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }
    else
    {
        app.UseExceptionHandler("/Home/Error");
        app.UseHsts();
    }

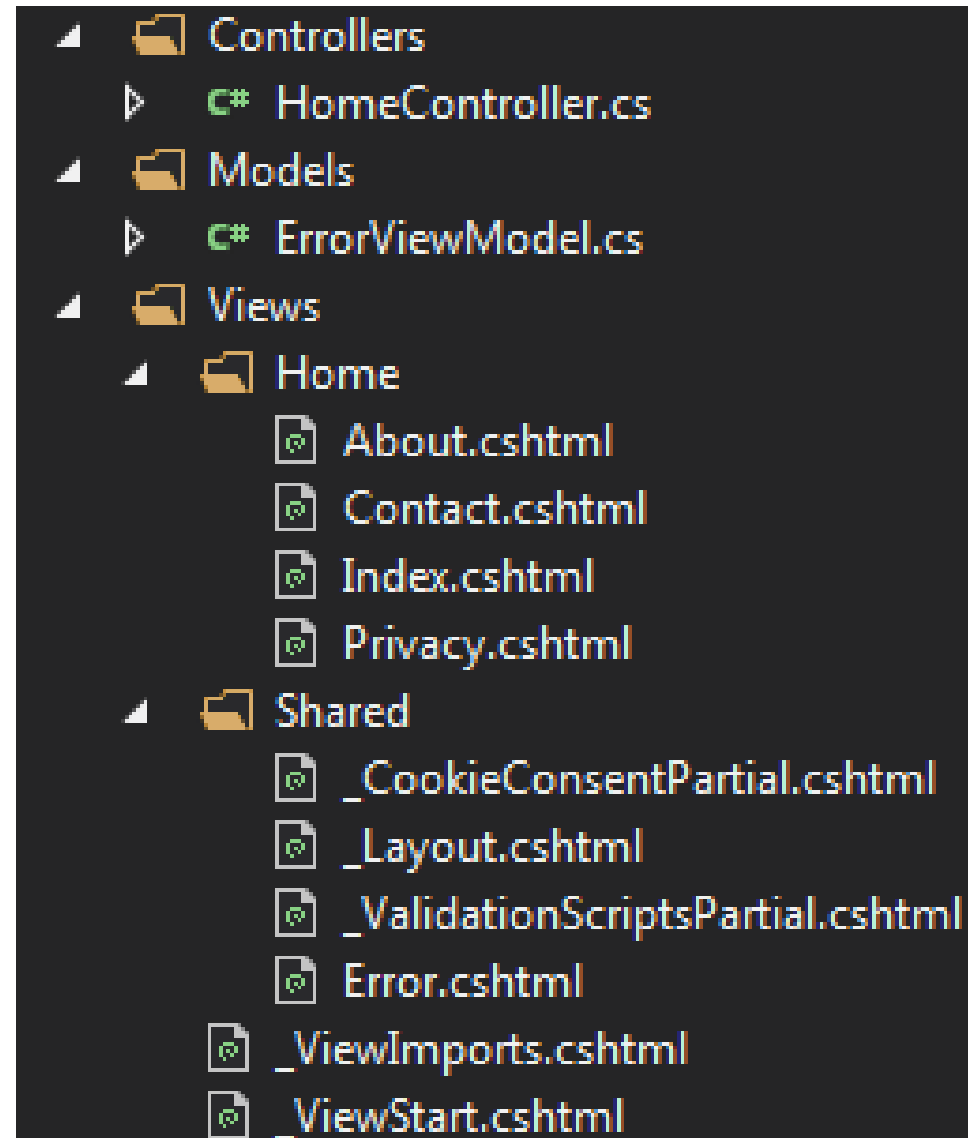
    app.UseHttpsRedirection();
    app.UseStaticFiles();
    app.UseCookiePolicy();

    app.UseMvc(routes =>
    {
        routes.MapRoute(
            name: "default",
            template: "{controller=Home}/{action=Index}/{id?}");
    });
}
```



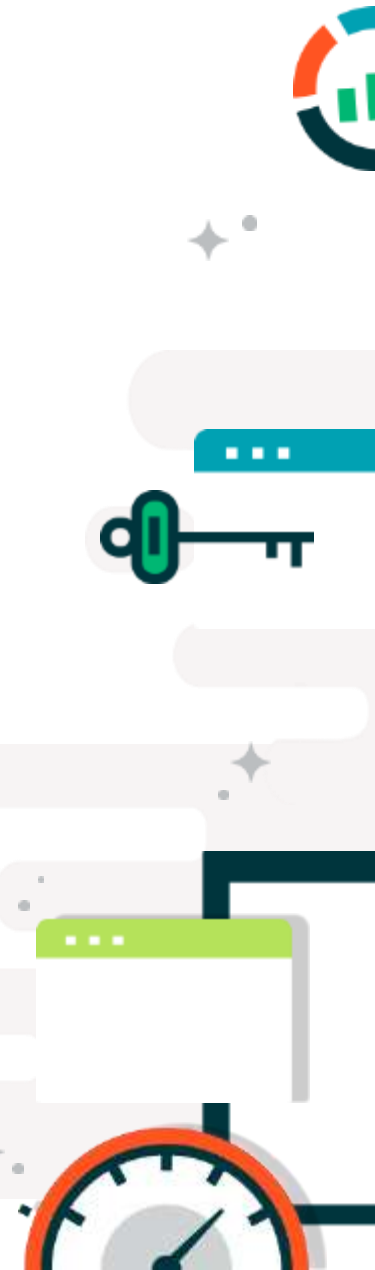
Olhando para o modelo

- Convenções
- Shared – MasterPage
- Partial – USC



The image shows a file explorer window with a dark background. It displays the following structure:

- Controllers
 - C# HomeController.cs
- Models
 - C# ErrorViewModel.cs
- Views
 - Home
 - About.cshtml
 - Contact.cshtml
 - Index.cshtml
 - Privacy.cshtml
 - Shared
 - _CookieConsentPartial.cshtml
 - _Layout.cshtml
 - _ValidationScriptsPartial.cshtml
 - Error.cshtml
 - _ViewImports.cshtml
 - _ViewStart.cshtml



Olhando para o modelo

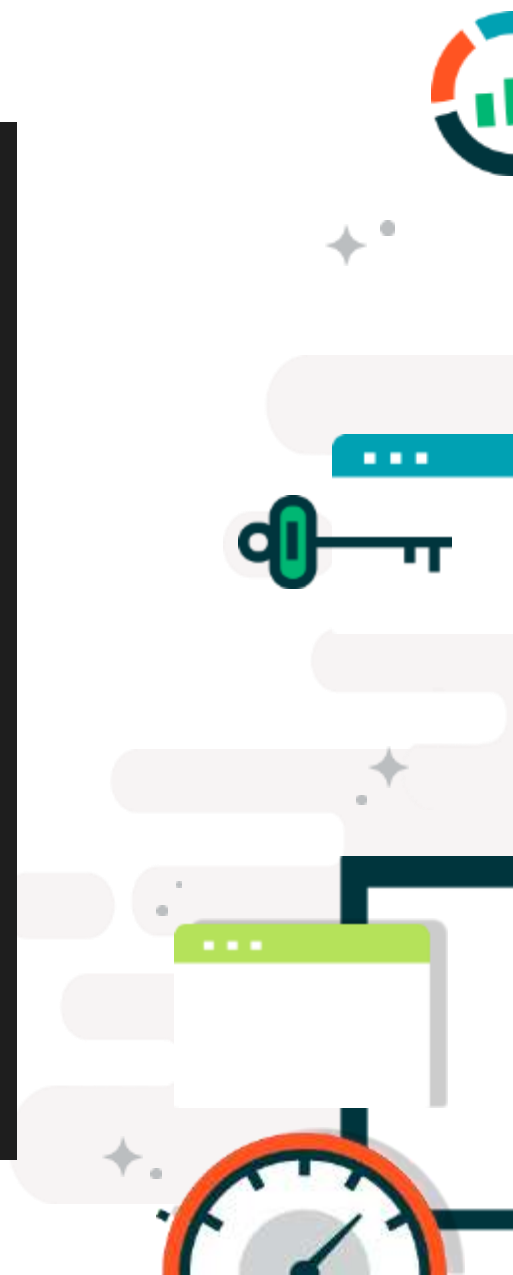
```
public class HomeController : Controller
{
    public IActionResult Index()
    {
        ViewModel model = new ViewModel();

        //Uso de serviços ou camada de negócios para preencher o Modelo

        return View(model);
    }

    public IActionResult About()
    {
        ViewData["Message"] = "Your application description page.";

        return View();
    }
}
```



Olhando para a construção de interface

```
<tbody>
  @foreach (AedesEmFoco.Models.aef_avaliacao avaliacao in Model.Avaliacoes.OrderByDescending(f => f.Id))
  {
    <tr class="@Html.Raw("linha" + avaliacao.Id.ToString())" style="@Html.Raw(avaliacao.aef_registro.Resolvido == true ? "background-color: #d9ead3;" : "background-color: #f2f2f2;")">
      <td>@avaliacao.Id</td>
      <td>@avaliacao.GetNome()</td>
      <td>@avaliacao.GetContato()</td>
      <td>@avaliacao.NomeAgente</td>
      <td>@avaliacao.DataHora</td>
      <td>@avaliacao.Cordialidade</td>
      <td>@avaliacao.BuscaPorFocos</td>
      <td>@avaliacao.Informacao</td>
      <td>@avaliacao.AvaliacaoGeral</td>
      <td>@avaliacao.aef_registro.Detalhe</td>
      <td>
        @if (@avaliacao.aef_registro.Latitude != null && @avaliacao.aef_registro.Latitude.Length > 3)
        {
          <a href="https://www.google.com.br/maps?q=@avaliacao.aef_registro.Latitude,@avaliacao.aef_registro.Longitude" target="_blank">Ver no Google Maps</a>
        }
        else
        {
          <span>Sem informação georreferencial</span>
        }
      </td>
      <td><a href="#" codRegistro="@avaliacao.aef_registro.Id" txtResposta="@avaliacao.aef_registro.RespostaGestao" class="btnResposta">Resposta</a>
      </td>
    </tr>
  }
}
```

Aspectos Positivos

- Melhor divisão de interface e demais recursos (devido ao *Controller*)
- Utilização de padrão e linguagens amplamente difundidos: MVC, HTML, CSS, JS, etc
- Completo domínio sobre renderização
- Mais leve e rápido



Aspectos Negativos

- Maior esforço para implementação (?)
- Para quem já acostumou com ASPX é uma abordagem diferente



Camada de Interfaces

Leandro Garcia



