Goiânia 29 de novembro de 2020

**Introdução de teoria do MVC.**

O que framework: E o conjunto de classe que facilita a execução.

O que MVC: E uma arquitetura de organização de classes as principais são Model, View, Controller.

Model: Responsabilidade de conversa com as classes de persistência e manipulação de banco de dados.

View: Visão do sistema recebe os dados que serão apresentados nas telas.

Controller: E o intermediário que comunica View e Model.

**Detalhando um projeto MVC**

Falar um pouco sobre arquitetura do template criada no MVC

1. App\_Data:
   1. Contém dados do banco.
2. App\_Start:
   1. Configuração vem como padrão basicamente 3 arquivos 1 de configuração 1 arquivo que gera scripts para não fica fazendo várias requisições e um arquivo de filtros vamos falar um pouco deles e quem são.
      1. BundleConfig.cs: Este arquivo vai servi para organiza e gera único arquivo com scripts ou folha de estilo fazendo somente 2 requisições Documentação: BundleConfig => <http://go.microsoft.com/fwlink/?LinkId=254725>
      2. App\_FilterConfig.cs: Ele trabalha com mensagens de erros e alertas já então toda vez que tiver uma mensagem de erro eu posso colocar aqui e não e obrigatório simplesmente e arquitetura do MVC.
      3. 2.1.3 App\_RouteConfig.cs: E para carregar as rotas (Endereço Web).
3. Content:
   1. Neste ponto ficara guardado arquivos CSS (folha de estilo) imagens.
4. **Controllers:**
   1. Serve para guarda os controladores que vai intermediar comunicam com Views e Models e sempre vem por padrão HomeController.
   2. Quais forma de fazer uma associação das View com valores dentro dos controladores ViewData, ViewBag, View.
      1. ViewData: esta forma passa como ararei veja exemplo de uma classe Pessoa para mostrar sintaxe e como instanciar.
         1. ViewData["PesssoaId"] = pessoaViewData.PesssoaId;
         2. ViewData["Nome"] = pessoaViewData.Nome;
         3. ViewData["Tipo"] = pessoaViewData.Tipo;
      2. ViewBag:
         1. ViewBag.Id = pessoaViewBag. PesssoaId;
         2. ViewBag.Nome = pessoaViewBag.Nome;
         3. ViewBag.Tipo = pessoaViewBag.Tipo;
      3. View Tipada: muito utilizado na para os get set;
         1. ViewBag.Id = pessoaViewBag. PesssoaId;
         2. ViewBag.Nome = pessoaViewBag.Nome;
         3. ViewBag.Tipo = pessoaViewBag.Tipo;
5. Fonts:
   1. Aqui e armazenados somente as fontes especiais coisa assim.
6. **Models:** 
   1. Serve para fazer associação com banco de dados
7. Script:
   1. Aqui fica os Java Scripts e outros scripts.
8. **Views:**
   1. E onde fica os arquivos de visão da aplicação no caso os HTML sendo que para todo controlador haverá uma pasta no Views.
      1. Um exemplo de código representado pela forma de ViewData.

<h2>Representado a forma de ViewData</h2>

<p>@ViewData["PesssoaId"]</p>

<p>@ViewData["Nome"]</p>

<p>@ViewData["Tipo"]</p>

* 1. Exemplo:
     1. Pasta Controlador: Controllers/HomeController ficará assim.
     2. Pasta Views: Views/Home/(conteúdo do Home) ficará assim.
     3. Ainda na Views existe uma pasta chamada de shared (ferramenta) nelas contém os HTML padrão no caso vem como nome padrão \_Layout.cshtml.
  2. Para entender como funciona o \_layaut.cshtml usaremos o seguinte código @Html.ActionLink("Inicial","Index","Home")
  3. Como exemplo mostrado acima nos demostra.
     + 1. Primeiro Parâmetro Nome da página.
       2. Segundo parâmetro Nome da Action (Ação) que vai ser executar.
       3. Terceiro parâmetro Nome do controlador que vai ser executado.

1. Global.
   1. asax: Onde inicializa nossa aplicação.
2. Web.config:
   1. E responsável por carregar todos as configurações que será utilizado na aplicação.