

1 - Arquivo Views/Home/Index.cshtml

Explicação: Criação do formulário responsável pela entrada de dados. Usamos a **Chave** "Nome" e "Idade" para conectar o HTML ao C# e o **Action** funciona como um localizador que busca e aciona a função "Processar" dentro do arquivo Controller.

```
1  @{
2      ViewData["Title"] = "Formulário de Pessoas";
3  }
4
5  <form asp-action="Processar" method="post">    <!-- asp-action="Processar" ele diz para procurar a função "Processar" no arquivo
6      Controllers/HomerController.cs -->
7      <input name="Nome" placeholder="Digite o seu nome" />    <!-- "Nome" é a Chave que vai para Models/Pessoas.cs-->
8      <input name="Idade" placeholder="Digite sua idade" />    <!-- "Idade" é a Chave que vai para Models/Pessoas.cs-->
9      <button type="submit">Adicionar na Lista</button>
10 
```

2 - Arquivo Models/Pessoa.cs

Explicação: Aqui é onde ocorre o armazenamento dos dados (as **Chaves**) através da classe "Pessoa" usando a propriedade Getter e Setter

```
namespace Asp.Net_Projeto03_DadosForm.Models;
3 referências
public class Pessoa
{
    0 referências
    public string Nome { get; set; }    // Onde vai receber a chave "Nome" no formulário html, usando a propriedade Getter e
    Setter
    0 referências
    public int Idade { get; set; }    // Onde vai receber a chave "Idade" no formulário html, usando a propriedade Getter e
    Setter
}
```

3 - Arquivo Controllers/HomeControllers

Explicação : Onde ocorre a criação da variável que armazena a lista em código C# e a função de "Processar". Essa função recebe os dados, salva na lista e prepara a exibição (o HTML em forma de C# através da **ViewBag**).

```

// Variável que guarda a lista de pessoas digitadas
3 referências
private static List<Pessoa> listaGeral = new List<Pessoa>(); // Lista gerada daquela classe Pessoa.cs
0 referências
D public IActionResult Index()
{
    ViewBag.Pessoas = listaGeral; // Quando abre o site pela primeira vez, já manda a lista (mesmo vazia)
    return View();
}

[HttpPost] // Indica que esse método só responde a requisições POST
0 referências
public IActionResult Processar(Pessoa input) // ** Código responsável por gerar a função "Processar" do formulário HTML
{
    listaGeral.Add(input); // Guarda o que você digitou
    ViewBag.Pessoas = listaGeral; // Atualiza a "sacola" que o HTML vai ler
    return View("Index"); // Recarrega a mesma página com o dado novo
}

```

→ Variável **listaGeral** é a lista (memória) que armazena os objetos gerados a partir da classe “**Pessoa**”.

→ Variável **ViewBag.Pessoas** é a referência (etiqueta) que transporta os dados armazenados para serem exibidos no **front-end**.

4 - Arquivo Views/Home/Index.cshtml:

Explicação: Onde ocorre a saída de dados para o usuário. Note que a gente não acessa o "banco de dados" direto; a gente cria uma variável temporária dentro do código (chamada de **p**) para extrair os dados de dentro da **ViewBag.Pessoas** (que foi gerada e enviada pelo arquivo **Controllers/HomeController**).

```

<hr>

<h3>Minha Lista:</h3>
<ul>
    @if (ViewBag.Pessoas != null)
    {
        @foreach (dynamic p in ViewBag.Pessoas) // "p" é a intancia da lista ViewBag.Pessoas
        {
            <li>@p.Nome - @p.Idade anos</li>
        }
    }
</ul>

```