**Ctrl + shift + R =**  Serve para carregar uma nova memória no explorer

**Crtl + P =**  Serve para procurar paginas no visual

**MVC (Model, View, Controller )**

**View** = Tela de comunicação, tela que chega para o usuário

**Controller** = é uma classe. Recebe a entrada do usuário e decide um melhor caminho

**Model** = Controle e visão.

( Meu software vai acontecer nesta pagina)

**Framework** = Software de programa

**Servidor** = Software que esta preparado para aprender certas mensagens

**ViewData:** ViewData é um dicionário que pode conter pares de valores-chave em que cada chave deve usar uma string

Notas: O ViewData transfere os dados do controlador para a visualização, não vice-versa. É válido apenas durante a solicitação atual

1. O ViewData transfere dados do Controller para o View, não vice-versa.
2. ViewData é derivado de ViewDataDictionary, que é um tipo de dicionário.
3. A vida do ViewData dura apenas durante a solicitação http atual. Os valores de ViewData serão limpos se ocorrer o redirecionamento.
4. O valor ViewData deve ser do tipo convertido antes do uso.

*Exemplo:*

public ActionResult Index()

{

IList<Student> studentList = new List<Student>();

studentList.Add(new Student(){ StudentName = "Bill" });

studentList.Add(new Student(){ StudentName = "Steve" });

studentList.Add(new Student(){ StudentName = "Ram" });

ViewData["students"] = studentList;

return View();

}

*Exemplo de como o ViewData pode ser acessado:*

<ul>

@foreach (var std in ViewData["students"] as IList<Student>)

{

<li>

@std.StudentName

</li>

}

</ul>