

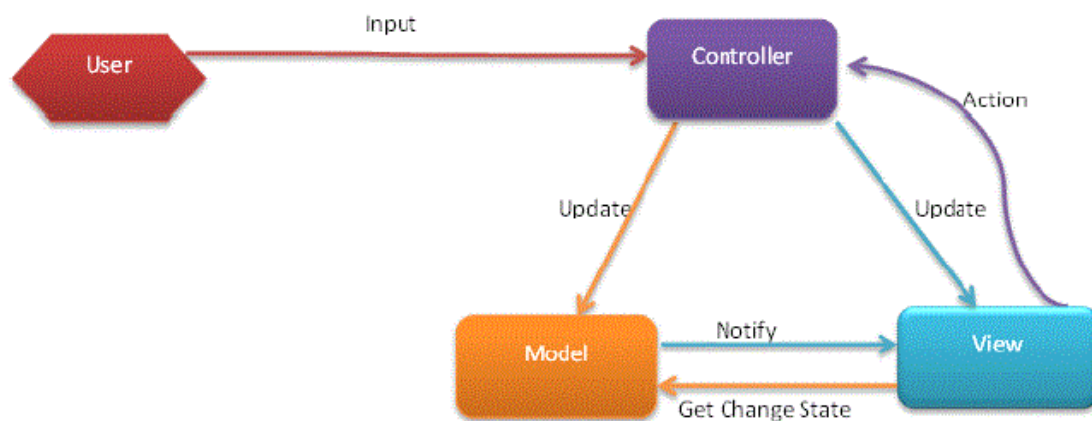
Exemplo 01 – Usando ASP.NET MVC 4

MVC – Revisão

Model View Controller: MVC é uma arquitetura de software, que separa a lógica da interface do usuário. Isto é conseguido através da separação da aplicação em três partes Model, View e Controller. O foco da MVC é a separação de responsabilidades. A MVC também é um padrão de projeto.

1. O **Model**: Representa o comportamento lógico dos dados na aplicação. Ele representa a lógica de negócios da aplicação. O Model notifica a View e o Controller sempre que ocorre uma mudança de estado;
2. A **View**: Fornece a interface de usuário da aplicação. A view é aquele que transforma o estado do Model em HTML legível;
3. O **Controller**: Aceita entradas do usuário e instrui a View e o Model para realizar a Action correspondente;

Abaixo, temos a figura, onde representamos o comportamento do padrão MVC.



Criando o seu primeiro projeto ASP .NET MVC 4

1. Abra o [Visual Studio Express 2012 for Web](#) e no menu File clique em **New Project**;
2. Selecione o *template* Visual C# -> Web -> ASP .NET MVC 4 Web Application;
3. Informe o nome PrimeiraAPP_MVC4 e clique no botão OK;
4. Na próxima janela selecione o *template* Internet Application , o *view Engine Razor* e clique no botão OK;

Será criado o projeto completo com a estrutura das pastas padrão de um projeto ASP .NET MVC.

Podemos observar no projeto as seguintes pastas:

- Controllers - contém - Local onde iremos colocar as classes de controle;
- Models - Local onde iremos colocar as classes para definição e/ou persistência de dados;
- Views - Local onde teremos as páginas para apresentação ao cliente;

- Executando o projeto iremos obter o seguinte resultado onde podemos ver as seções **Home**, **About** e **Contact** bem como as opções para **Log in** e **Register** adicionadas e criadas por padrão:

Alterando o projeto e exibindo detalhes de usuários

Vamos incluir no projeto o nosso **Controller**, **Model** e **View** para podermos exibir os detalhes dos usuários.

Clique com o botão direito em **Models**, clique em **Add Class** e adicione uma classe chamada **UsuarioModel.cs** com a seguinte estrutura:

```
using System;
namespace PrimeiraAPP_MVC4.Models
{
    public class UsuarioModel
    {
        public string nome { get; set; }
        public string sobrenome { get; set; }
        public string endereco { get; set; }
        public string email { get; set; }
        public DateTime nascimento { get; set; }
    }
}
```

Vamos agora implementar algumas validações nos campos definidos na classe UsuarioModel usando [Data Annotations](#). Para isso vamos alterar o código conforme exibido a seguir:

```
using System;
using System.ComponentModel;
using System.ComponentModel.DataAnnotations;
namespace PrimeiraAPP_MVC4.Models
{
    public class UsuarioModel
    {
        [DisplayName("Primeiro Nome")]
        [StringLength(50, ErrorMessage = "O campo Nome permite no máximo 50
caracteres!")]
        public string nome { get; set; }
        [Required]
        public string sobrenome { get; set; }
        public string endereco { get; set; }
        [StringLength(50)]
        [Required(ErrorMessage="Informe o Email")]
        [RegularExpression(@"\w+([-+.]?\w+)*@\w+([-.]?\w+)*\.\w+([-.]?\w+)*",
ErrorMessage = "Email inválido.")]
        public string email { get; set; }
        [DataType(DataType.Date)]
        public DateTime nascimento { get; set; }
    }
}
```

No exemplo usamos os seguintes atributos:

- **Required** : Torna o valor do campo obrigatório.
- **StringLength**: Para definir o comprimento máximo do campo.
- **DataType**: Para definir o tipo suportado pelo campo.

Vamos agora criar uma nova classe na pasta **Models** onde iremos definir alguns dados para os nossos usuários.

Clique com o botão direito do **Models**, clique em **Add Class** e adicione uma classe chamada **Usuario.cs** com a seguinte estrutura:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
namespace PrimeiraAPP_MVC4.Models
{
    public class Usuarios
    {
        public List<UsuarioModel> listaUsuarios = new List<UsuarioModel>();
        public Usuarios()
        {
            listaUsuarios.Add(new UsuarioModel
            {
                nome = "Jose Carlos",
                sobrenome = "Macoratti",
                endereco = "Rua Projetada , 100",
                email = "macoratti@yahoo.com",
                nascimento = Convert.ToDateTime("05/09/1975")
            });
            listaUsuarios.Add(new UsuarioModel
            {
                nome = "Jefferson Andre",
                sobrenome = "Ribeiro",
                endereco = "Rua Mirassol , 50",
                email = "jeffbol@bol.com.br",
                nascimento = Convert.ToDateTime("13/08/1992")
            });
            listaUsuarios.Add(new UsuarioModel
            {
                nome = "Janice Lima",
                sobrenome = "Morais",
                endereco = "Rua Peru , 10",
                email = "janielima@hotmail.com",
                nascimento = Convert.ToDateTime("15/07/1990")
            });
        }
        public void CriaUsuario(UsuarioModel usuarioModelo)
        {
            listaUsuarios.Add(usuarioModelo);
        }
        public void AtualizaUsuario(UsuarioModel usuarioModelo)
        {
            foreach (UsuarioModel usuario in listaUsuarios)
            {
                if (usuario.email == usuarioModelo.email)
                {
                    listaUsuarios.Remove(usuario);
                    listaUsuarios.Add(usuarioModelo);
                    break;
                }
            }
        }
        public UsuarioModel GetUsuario(string Email)
        {

```

```

        UsuarioModel _usuarioModel = null;
        foreach (UsuarioModel _usuario in listaUsuarios) {
            if (_usuario.email == Email)
                _usuarioModel = _usuario;
            return _usuarioModel;
        }
    }
}
public void DeletarUsuario(String Email)
{
    foreach (UsuarioModel _usuario in listaUsuarios)
    {
        if (_usuario.email == Email)
        {
            listaUsuarios.Remove(_usuario);
            break;
        }
    }
}
}
}

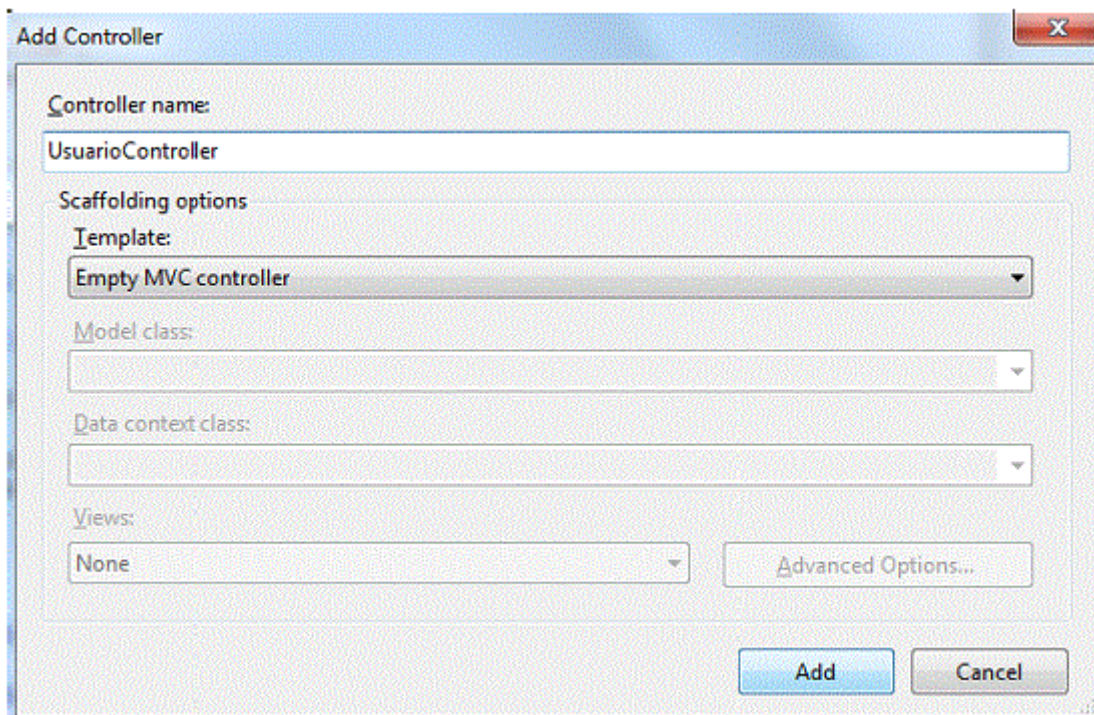
```

Nesta classe definimos um construtor (Usuario) que cria 3 usuários e 3 métodos:

- CriaUsuario - Cria uma lista de usuários;
- AtualizaUsuario - Atualiza um usuário;
- Getusuario - Obtém um usuário;
- DeletarUsuario - Exclui um usuário;

Vamos agora criar um **Controller** para a nossa aplicação. Clique com o botão direito do mouse sobre a pasta **Controllers** e selecione **Add-> Controller**;

A seguir informe o nome **UsuarioController** e escolha o **Template -Empty MVC controller** e clique no botão **Add**;



Vamos definir o código abaixo em nosso controller:

```

using System.Web.Mvc;
using PrimeiraAPP_MVC4.Models;
namespace PrimeiraAPP_MVC4.Controllers
{
    public class UsuarioController : Controller
    {
        //
        // GET: /Usuario/
        private static Usuarios _usuarios = new Usuarios();
        public ActionResult Index()
        {
            return View(_usuarios.listaUsuarios);
        }
        public ActionResult AdicionaUsuario()
        {
            return View();
        }
        [HttpPost]
        public ActionResult AdicionaUsuario(UsuarioModel _usuarioModel)
        {
            _usuarios.CriaUsuario(_usuarioModel);
            return View();
        }
        public ActionResult DeletaUsuario(string id)
        {
            return View(_usuarios.GetUsuario(id));
        }
        [HttpPost]
        public RedirectToRouteResult DeletaUsuario(string id, FormCollection
collection)
        {
            _usuarios.DeletarUsuario(id);
            return RedirectToAction("Index");
        }
    }
}

```

Nosso controlador possui três métodos HttpGet:

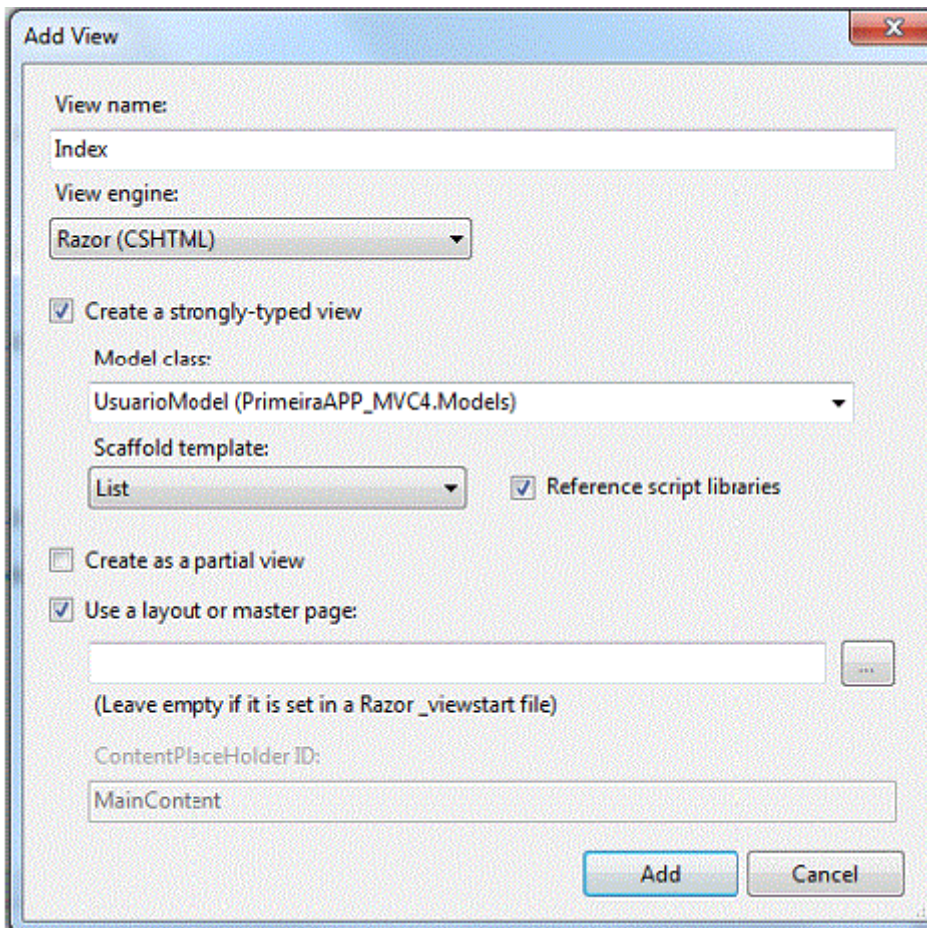
- Index
- AdicionaUsuario
- DeletaUsuario

Além disso temos dois métodos **HttpPost**:

- **AdicionaUsuario** - que usaremos para postar as entradas para uso posterior;
- **DeletaUsuario** - que trata a exclusão do usuário;

Vamos criar as nossas views a partir dos métodos do controlador. Clique com o botão direito sobre o método **Index()** e selecione **Add View**;

A seguir informe selecione a opção Create a strongly-typed view conforme a figura abaixo e clique no botão **Add**;



Será criada a view **Index.chstml** na pasta **Views/Usuarios** com o código abaixo:

```

@model IEnumerable<PrimeiraAPP_MVC4.Models.UsuarioModel>
@{
ViewBag.Title = "Index";
}
<h2>Index</h2>
<p>
@Html.ActionLink("Criar Usuário", "AdicionaUsuario")
</p>
<table>
<tr>
<th>@Html.DisplayNameFor(model => model.nome)</th>
<th>@Html.DisplayNameFor(model => model.sobrenome)</th>
<th>@Html.DisplayNameFor(model => model.endereco)</th>
<th>@Html.DisplayNameFor(model => model.email)</th>
<th>@Html.DisplayNameFor(model => model.nascimento)</th>
<th></th>
</tr>
<tr>
@foreach (var item in Model) {
<tr>
<td>@Html.DisplayFor(modelItem => item.nome)</td>
<td>@Html.DisplayFor(modelItem => item.sobrenome)</td>
<td>@Html.DisplayFor(modelItem => item.endereco)</td>
<td>@Html.DisplayFor(modelItem => item.email)</td>
<td>@Html.DisplayFor(modelItem => item.nascimento)</td>
<td>
@Html.ActionLink("Edit", "Edit", new { /* id=item.PrimaryKey */ }) |

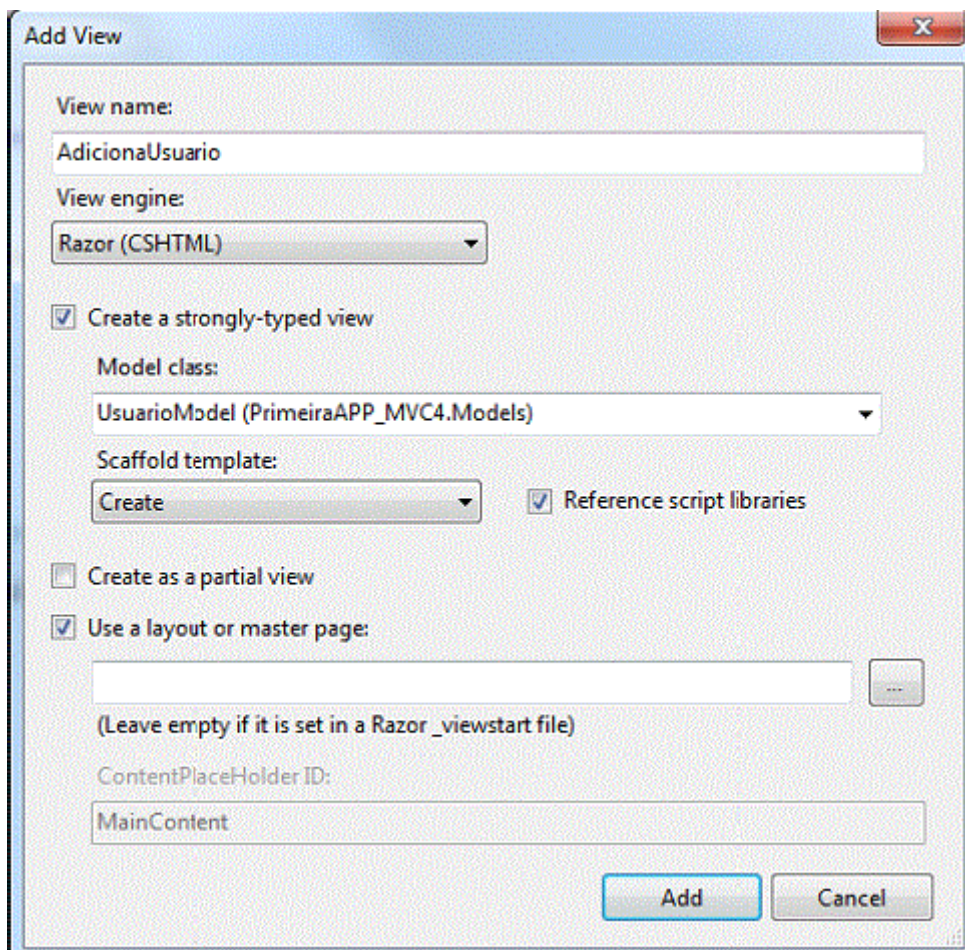
```

```
@Html.ActionLink("Details", "Details", new { /* id=item.PrimaryKey */ }) |
@Html.ActionLink("Deletar", "DeletaUsuario", new { id=item.email })
</td>
</tr>
}
</table>
```

Vamos alterar o código do ActionLink do link Criar Usuário para `@Html.ActionLink("Criar Usuário", "AdicionaUsuario")` de forma a podermos chamar a view AdicionaUsuario que iremos criar a seguir.

Vamos repetir o processo e criar uma view para o método AdicionaUsuario() do controlador UsuarioController (*clique com o botão direito do mouse sobre o método AdicionaUsuario e selecione Add-> View*);

A seguir informe selecione a **opção Create a strongly-typed view** conforme a figura abaixo e clique no botão **Add**;



Será criada a view **AdicionaUsuario.chstml** na pasta **Views/Usuarios** com o código abaixo:

```
@model PrimeiraAPP_MVC4.Models.UsuarioModel
@{
ViewBag.Title = "AdicionaUsuario";
}
<h2>AdicionaUsuario</h2>
```

```

@using (Html.BeginForm()) {
@Html.ValidationSummary(true)
<fieldset>
<legend>UsuarioModel</legend>
<div class="editor-label">@Html.LabelFor(model => model.nome)</div>
<div class="editor-field">@Html.EditorFor(model => model.nome)
@Html.ValidationMessageFor(model => model.nome)
</div>
<div class="editor-label">@Html.LabelFor(model => model.sobrenome)</div>
<div class="editor-field">
@Html.EditorFor(model => model.sobrenome)
@Html.ValidationMessageFor(model => model.sobrenome)
</div>
<div class="editor-label">@Html.LabelFor(model => model.endereco)</div>
<div class="editor-field">
@Html.EditorFor(model => model.endereco)
@Html.ValidationMessageFor(model => model.endereco)
</div>
<div class="editor-label">@Html.LabelFor(model => model.email)</div>
<div class="editor-field">
@Html.EditorFor(model => model.email)
@Html.ValidationMessageFor(model => model.email)
</div>
<div class="editor-label">@Html.LabelFor(model => model.nascimento)</div>
<div class="editor-field">
@Html.EditorFor(model => model.nascimento)
@Html.ValidationMessageFor(model => model.nascimento)
</div>
<p>
<input type="submit" value="Create" />
</p>
</fieldset>
}
<div>
@Html.ActionLink("Back to List", "Index")
</div>
@section Scripts {
@Scripts.Render("~/bundles/jqueryval")
}

```

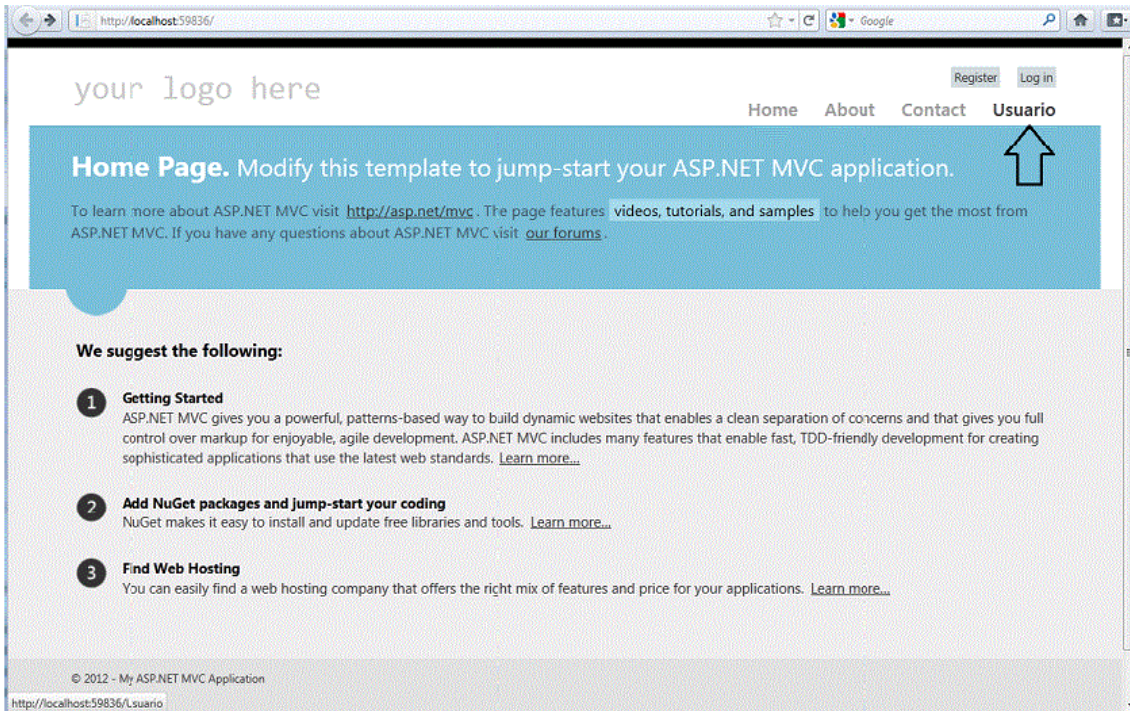
Antes de executar o projeto vamos alterar o arquivo **_Layout.chstml** da pasta **/Shared** incluindo a linha destacada em azul abaixo no arquivo:

```

<nav>
<ul id="menu">
<li>@Html.ActionLink("Home", "Index", "Home")</li>
<li>@Html.ActionLink("About", "About", "Home")</li>
<li>@Html.ActionLink("Contact", "Contact", "Home")</li>
<li>@Html.ActionLink("Usuario", "Index", "Usuario")</li>
</ul>
</nav>

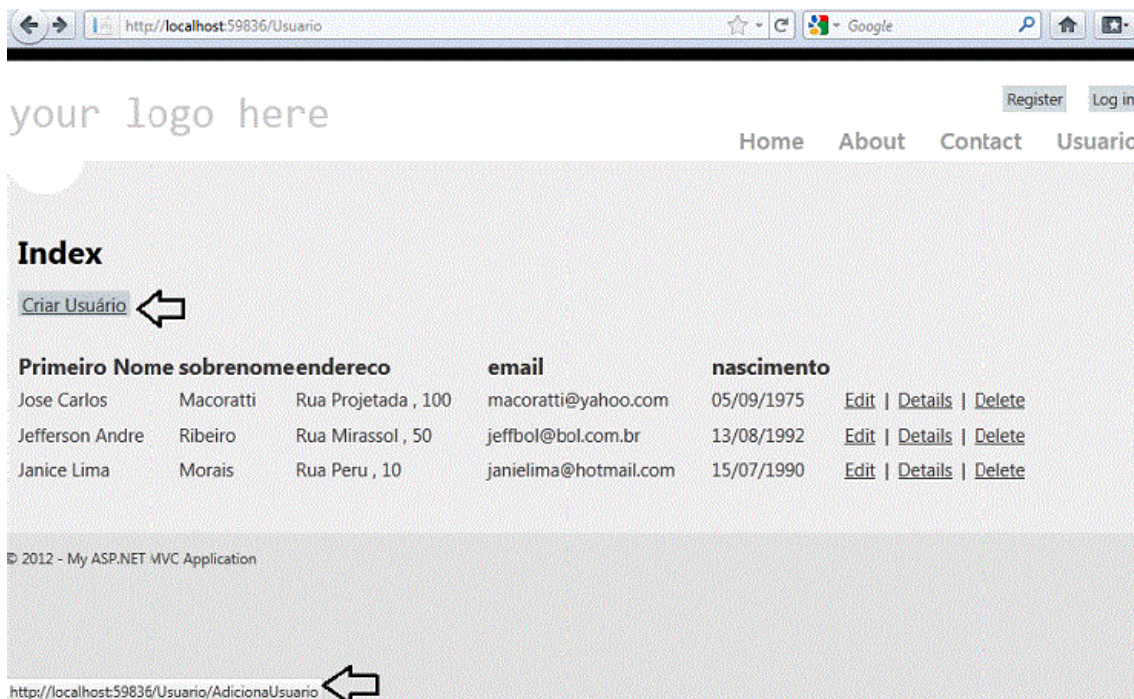
```

Executando o projeto iremos obter o seguinte resultado:



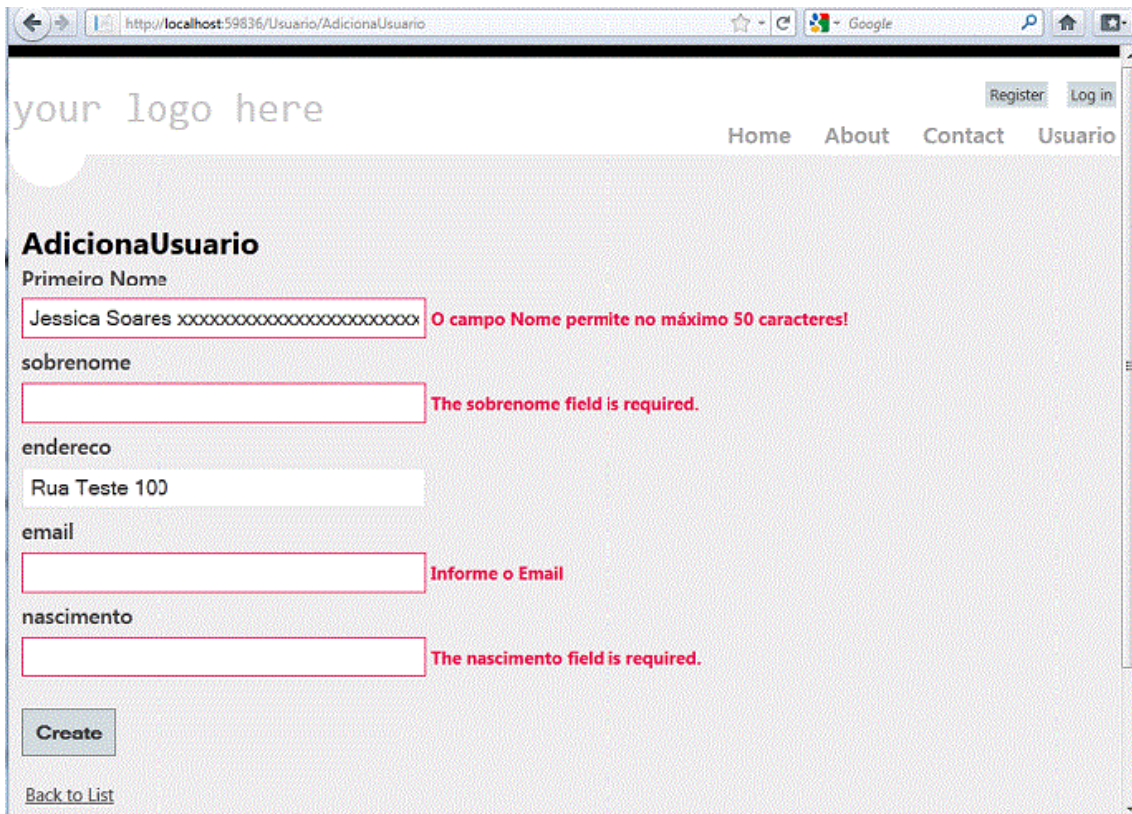
Observe que no menu temos agora a opção **Usuario** que nos dará acesso a **view Index.cshtml** da pasta **Views\Usuario**;

Clicando nesta opção teremos a apresentação da view **Index.cshtml** onde vemos o link Criar Usuário que irá acionar a view **AdicionaUsuario.cshtml**;

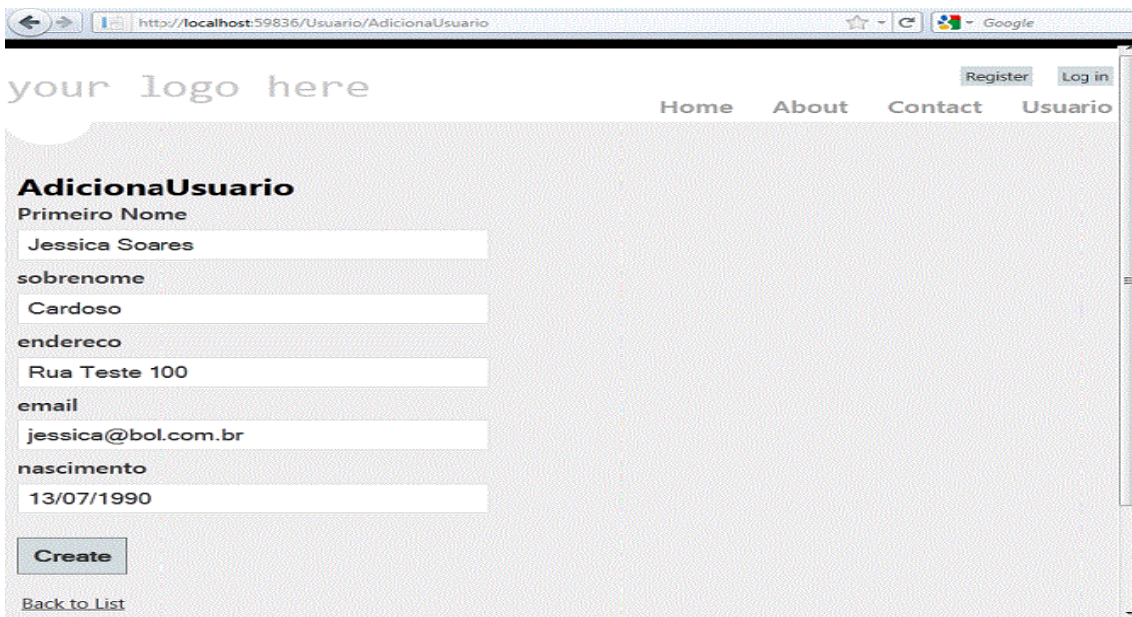


Ao clicar no link Criar Usuário teremos a página para cadastrar um novo usuário;

Vamos verificar primeiro se a nossa validação está funcionando. Na figura abaixo vemos o resultado obtido se violarmos qualquer uma das regras de validação definidas:



Cadastramento um usuário com todos os dados válidos conforme abaixo:



Ao clicarmos no link - **Back to List** - veremos o novo usuário sendo exibido na página conforme a seguir:

your logo here

Register Log in

Home About Contact Usuario

Index

[Criar Usuário](#)

Primeiro Nome	sobrenome	endereco	email	nascimento	
Jose Carlos	Macoratti	Rua Projetada , 100	macoratti@yahoo.com	05/09/1975	Edit Details Delete
Jefferson Andre	Ribeiro	Rua Mirassol , 50	jeffbol@bol.com.br	13/08/1992	Edit Details Delete
Janice Lima	Morais	Rua Peru , 10	janielima@hotmail.com	15/07/1990	Edit Details Delete
Jessica Soares	Cardoso	Rua Teste 100	jessica@bol.com.br	13/07/1990	Edit Details Delete

© 2012 - My ASP.NET MVC Application

Criamos assim nossa primeira aplicação usando ASP .NET MVC 4 e vimos que o nosso trabalho é muito facilitado pelos recursos oferecidos no framework. Se desejar você pode implementar as funcionalidades para *Editar*, *Detalhar* e *Deletar* usuários. Basta definir o método Model, no Controlador e a seguir criar a respectiva View.