



**TECNOLOGIA E INOVAÇÃO  
EM PROL DA INDÚSTRIA**



## **Curso Técnico em Informática**

# Desenvolvimento de Sistemas II – 180h

Prof<sup>a</sup>: Francisleide Almeida

Slide baseado no conteúdo de Macoratti



# Sessions c#

- Utilizado para autenticação de usuários no sistema
- Realiza o acesso à partes do sistema com determinadas condições
- Auxílio do Razor;
- Possibilidade de salvar os valores no estado de sessão

# Prática

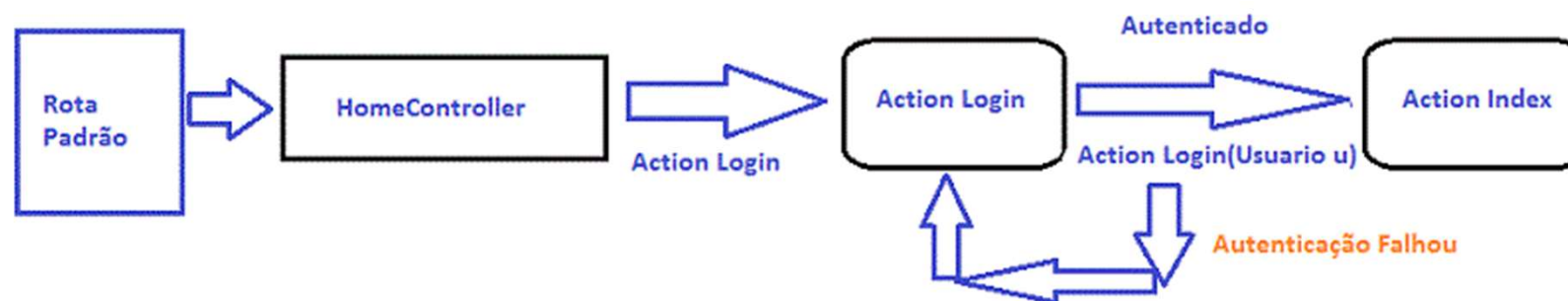
- Classe:

```
1  TesteSection.Models.Usuario  UsuarioId

using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Web;

namespace TesteSection.Models
{
    public class Usuario
    {
        public int UsuarioId { get; set; }
        [Display(Name = "Login")]
        [Required(ErrorMessage = "Informe o nome do usuário", AllowEmptyStrings = false)]
        public string Login { get; set; }
        [Required(ErrorMessage = "Informe a senha do usuário", AllowEmptyStrings = false)]
        [DataType(System.ComponentModel.DataAnnotations.DataType.Password)]
        public string Senha { get; set; }
        public string Cpf { get; set; }
    }
}
```

# Lógica



# Controller

```
1 public ActionResult Login()
2 {
3     return View(db.Usuarios.ToList());
4 }
5
6 [HttpPost]
7 public ActionResult Login([Bind(Include = "UsuarioId,Login,Senha,Cpf")] Usuario usuario)
8 {
9     if (ModelState.IsValid) //verifica se é válido
10     {
11         var v = db.Usuarios.Where(a => a.Login.Equals(usuario.Login) && a.Senha.Equals(usuario.Senha)).FirstOrDefault();
12         if (v != null)
13         {
14             Session["usuarioLogadoID"] = v.UsuarioId.ToString();
15             Session["nomeUsuarioLogado"] = v.Login.ToString();
16             return RedirectToAction("Home", "Index");
17         }
18     }
19     return View(usuario);
20 }
```

# View - Create

```
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <h4>Usuario</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <div class="form-group">
            @Html.LabelFor(model => model.Login, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Login, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Login, "", new { @class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.Senha, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Senha, new { htmlAttributes = new { @class = "form-control", @Type="password" } })
                @Html.ValidationMessageFor(model => model.Senha, "", new { @class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" value="Entrar" class="btn btn-default" />
            </div>
        </div>
    </div>
}
```



# Acesso

- Para não fornecer acesso sem antes ter feito o login:

```
public class HomeController : Controller
{
    public ActionResult Index()
    {
        if (Session["usuarioLogadoID"] != null)
        {
            return View();
        }
        else
        {
            return RedirectToAction("Login");
        }
    }
}
```

# Acesso