

Aula08 – Forms Authentication

Disciplina: Programação Web

Prof. Allbert Velleniche de Aquino Almeida

E-mail: allbert.almeida@fatec.sp.gov.br

Site: <http://www.allbert.com.br>



/allbert.almeida

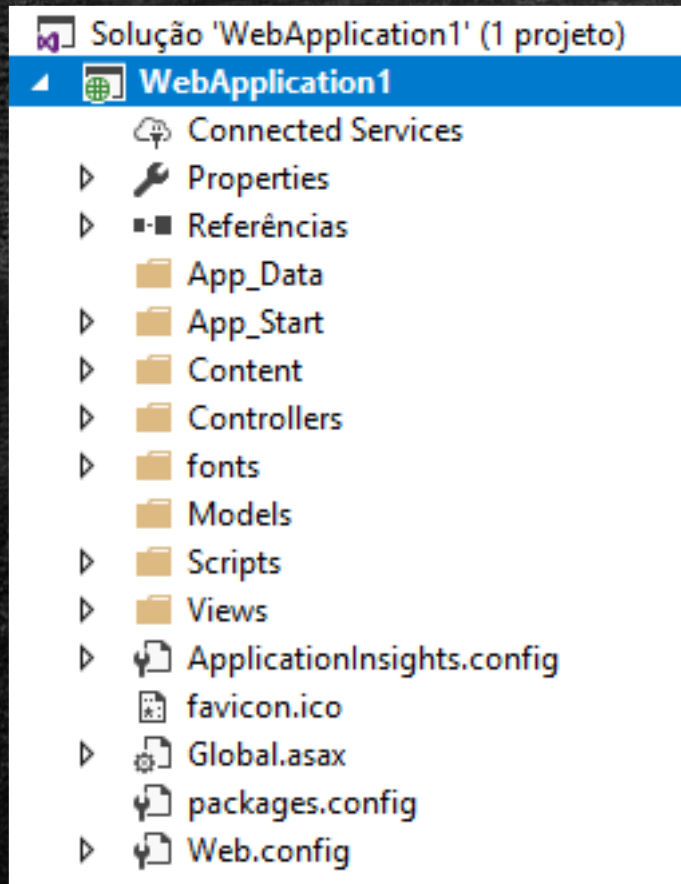
Objetivo

- O objetivo dessa aula é utilizar a criação de usuários para conceder acesso a aplicação através da autenticação de usuário. Dada a autenticação, utilizar os requisitos de autorização para determinar os privilégios de um determinado usuário.

Criando uma nova aplicação

- Vamos abrir o Visual Studio;
- File >> New >> Project >> Aplicativo Web;
- Tipo de aplicação MVC;
- Change Authentication >> No Authentication

Exemplo de uma aplicação



Modelo

Usuario
Id
Nome
Email
Senha

UsuarioPerfil
Id
UsuarioId
PerfilId

Perfil
Id
Descricao



Adicionar pacotes à solução

- MySql Data Entity 6.9.12

**Install-Package MySql.Data.Entity
-Version 6.9.12**

EntityFramework por Microsoft



Entity Framework is Microsoft's recommended data access technology for new applications.



MySql.Data.Entity por Oracle



Entity Framework 6.0 supported

Alteração Classe Usuário

```
public class Usuario {
    public int Id { get; set; }
    [Required]
    [MaxLength(200)]
    public string Nome { get; set; }
    [Required]
    [MaxLength(200)]
    [EmailAddress]
    public string Email { get; set; }
    public string Senha { get; set; }
    public virtual ICollection<UsuarioPerfil>
    UsuarioPerfil { get; set; }
}
```

Criar uma classe Perfil

```
public class Perfil
{
    public int Id { get; set; }
    [Required]
    [MaxLength(255)]
    [Display(Name = "Descrição")]
    public string Descricao { get; set; }
    public virtual
    ICollection<UsuarioPerfil> UsuarioPerfil
    { get; set; }
}
```


Criar uma classe UsuarioPerfil

```
public class UsuarioPerfil {  
    public int Id { get; set; }  
    public int UsuarioId { get; set; }  
    public int PerfilId { get; set; }  
    public virtual Usuario Usuario { get; set; }  
    public virtual Perfil Perfil { get; set; }  
}
```

Criar uma classe Contexto

```
public class Contexto:DbContext {  
    public Contexto() : base(nameOrConnectionString:  
"StringConexao") { }  
    public DbSet<Usuario> Usuario { get; set; }  
    public DbSet<Perfil> Perfil { get; set; }  
    public DbSet<UsuarioPerfil> UsuarioPerfil { get; set; }  
}
```

Adicionar a String de Conexão no WebConfig

```
<connectionStrings>
```

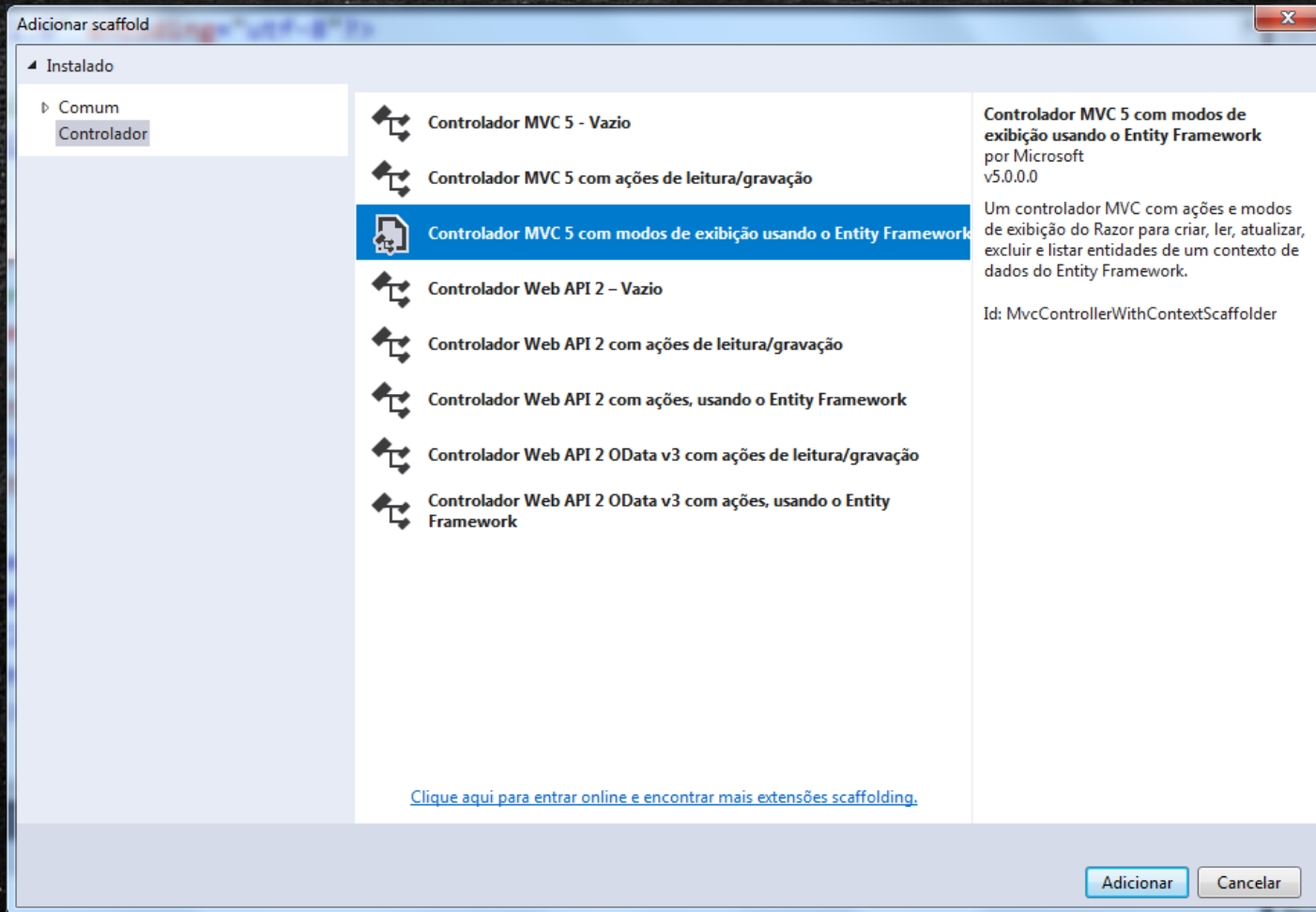
```
    <add name="StringConexao"  
providerName="MySql.Data.MySqlClient"  
connectionString="server=localhost;dat  
abase=aula;uid=root;pwd='root'" />
```

```
</connectionStrings>
```

Compilar a solução

- Compilar a solução para que os modelos sejam validados e seja possível criar o “CRUD” da classe Usuario; Perfil; UsuarioPerfil

Adicionar Controller



Adicionar Controller

Adicionar controlador

Classe do modelo: Usuario (WebApplication1.Models)

Classe de contexto de dados: Contexto (WebApplication1.Models) +

Usar ações assíncronas do controlador

Modos de exibição:

Gerar modos de exibição

Bibliotecas de scripts de referência

Usar uma página de layout:

...

(deixe em branco se ele estiver definido em um arquivo Razor _viewstart)

Nome do controlador: UsuariosController

Adicionar Cancelar

Executar a solução

- Na URL entre com o caminho da aplicação +
/Perfils/
- Na URL entre com o caminho da aplicação +
/Usuarios/
- Cadastrar perfil e usuário;

Forms Authentication

- Adicionar no system.web do web.config:

```
<authentication mode="Forms">  
<forms loginUrl="~/Home/Acesso"  
timeout="2880" />  
</authentication>
```


HomeController

- Vamos adicionar na HomeController a action Acesso;

```
public ActionResult Acesso() {  
    return View();  
}
```

- Depois adicionar a View usando como modelo a ViewModelo já criada;

HomeController

- Vamos adicionar na HomeController a action Acesso do tipo http post;

```
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Acesso(Acesso ace, string returnUrl) {
    string senhacrip = Funcoes.HashTexto(ace.Senha, "SHA512");
    Usuario usu = db.Usuario.Where(t => t.Email == ace.Email && t.Senha ==
    senhacrip).ToList().FirstOrDefault();
    if (usu != null) {
        FormsAuthentication.SetAuthCookie(usu.Id + "|" + usu.Nome, false);
        return RedirectToAction("Index", "Home");
    }
    else {
        ModelState.AddModelError("", "Usuário/Senha inválidos");
        return View();
    }
}
```

HomeController

- Vamos adicionar na HomeController a action Sair;

```
public ActionResult Sair() {  
    FormsAuthentication.SignOut();  
    return RedirectToAction("Index");  
}
```

HomeController

- Para que possamos realizar o teste de validação de usuário autenticado, devemos inserir acima da action "About" o "Authorize"

[Authorize]

```
public ActionResult About() {  
  
}
```

View _Layout.cshtml

- Vamos adicionar a o item na Layout;

```
<ul class="nav navbar-nav navbar-right">
@if (User.Identity.IsAuthenticated)
{
<li><span class="glyphicon glyphicon-log-
out"></span>@Html.ActionLink(User.Identity.Name.Split('|')[1] + " - Sair",
"Sair", "Home")</li>
}
else
{
    <li><a href="@Url.Action("Cadastro","Home")"><span class="glyphicon
glyphicon-user"></span> Cadastro</a></li>
    <li><a href="@Url.Action("Acesso","Home")"><span class="glyphicon
glyphicon-log-in"></span> Entrar</a></li>
}
</ul>
```

View _Layout.cshtml

- Vamos adicionar a o item na Layout;

```
<li>@Html.ActionLink("Usuário", "Index",  
"Usuarios")</li>
```

```
<li>@Html.ActionLink("Perfil", "Index",  
"Perfils")</li>
```

```
<li>@Html.ActionLink("Usuário/Perfil",  
"Index", "UsuarioPerfils")</li>
```

Compilar a solução novamente

- Pronto, agora podemos rodar a aplicação e testar, verificar se conseguimos acessar o link "About". Também podemos realizar o acesso com as credenciais corretas e entrar novamente em "About". Importante lembrarmos que só realizamos a validação de usuário autenticado, ou seja, basta ser usuário identificado para conseguir acesso.

Validar o Perfil

- Já vimos que somente usuários autorizados podem entrar na página, agora vamos ver como controlar o perfil;
- Depois de realizado os testes, iremos trabalhar agora com a autorização através do perfil do usuário.

Action Acesso

- Faremos uma modificação na action "Acesso" adicionando um Ticket com os perfis associado ao usuário:

```
string permissoes = "";
foreach (UsuarioPerfil p in usu.UsuarioPerfil)
    permissoes += p.Perfil.Descricao + ",";

permissoes = permissoes.Substring(0, permissoes.Length - 1);
FormsAuthenticationTicket ticket = new
FormsAuthenticationTicket(1, usu.Id + "|" + usu.Email,
DateTime.Now, DateTime.Now.AddMinutes(30), false, permissoes);
string hash = FormsAuthentication.Encrypt(ticket);
HttpCookie cookie = new
HttpCookie(FormsAuthentication.FormsCookieName, hash);
Response.Cookies.Add(cookie);
```

Global.asax

- Vamos adicionar o método;

```
protected void Application_PostAuthenticateRequest(Object sender, EventArgs e)
{
    var authCookie =
    HttpContext.Current.Request.Cookies[FormsAuthentication.FormsCookieName];
    if (authCookie != null) {
        FormsAuthenticationTicket authTicket =
        FormsAuthentication.Decrypt(authCookie.Value);
        if (authTicket != null && !authTicket.Expired) {
            var roles = authTicket.UserData.Split(',');
            HttpContext.Current.User = new
            System.Security.Principal.GenericPrincipal(new FormsIdentity(authTicket),
            roles);
        }
    }
}
```

Adicionar perfil na autorização

- Na PerfilController vamos adicionar autorização. Somente usuários autorizados com roles definido terão acesso a esse controller;
- Importante a autorização pode ser definida no controller inteiro ou somente nas actions;

```
[Authorize(Roles = "Admin")]
```

Compilar a solução novamente

- Compilar a solução novamente e verificar se conseguem acessar perfil;

Atividade prática

- Implementar no sistema do P.I. o sistema de cadastro e autenticação do usuário;
- Construir uma ViewModel com cadastro do usuário e inserir os dados no BD; Construir outra ViewModel para acesso do usuário;
- Realize a trigamação do BD e adicione alguns registros no método Seed (semente);