

---

# .NET Membership

## Gestão “automática” de utilizadores

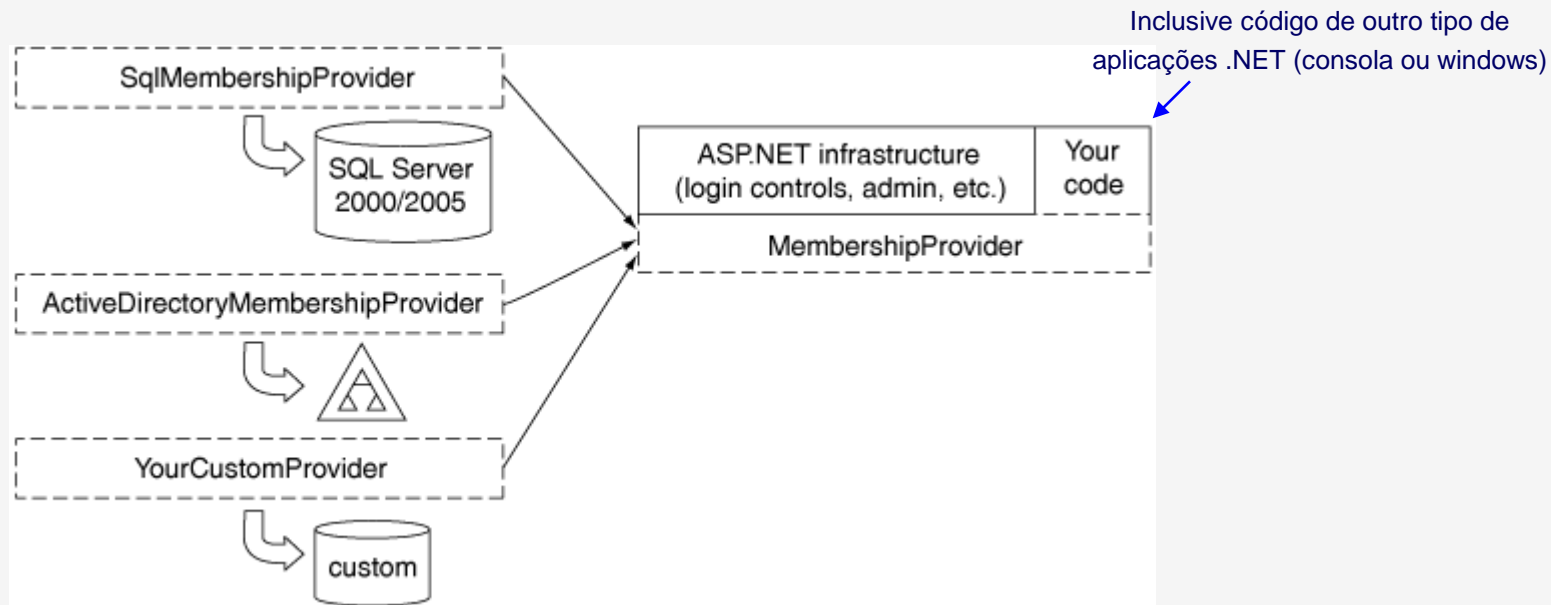
Notas para a UC de “Segurança Informática”  
Inverno de 11/12

Rui Joaquim ([rjoaquim@cc.isel.ipl.pt](mailto:rjoaquim@cc.isel.ipl.pt))

José Simão ([jsimao@cc.isel.ipl.pt](mailto:jsimao@cc.isel.ipl.pt))

[Instituto Superior de Engenharia de Lisboa](http://www.isel.ipl.pt)

# .NET Membership



- Sistema genérico de gestão e autenticação de utilizadores.
- Autenticação através de *username* e *password*.
- Arquitectura baseada em *providers*.

(imagem retirada do livro "Essential ASP.NET 2.0" de Fritz Onion e Keith Brown)

- Inicialmente desenvolvido para dar suporte à gestão de utilizadores em aplicações ASP.NET 2.0, no entanto pode ser usado em qualquer outro tipo de aplicação .NET (consola ou windows).

# Configuração do Membership I

---

- A selecção do *provider* e sua configuração é realizada através do ficheiro *app.config* (aplicações de consola e windows) ou do ficheiro *web.config*.
- Se nada for configurado nestes ficheiro é utilizada a configuração que se encontra em *machine.config*.
- Por omissão o *provider* utilizado é do tipo *SqlMembershipProvider*.

```
<system.web>
  <membership defaultProvider="provider name" hashAlgorithmType="SHA1">
    <providers>...</providers>
  </membership>
</system.web>
```

# Configuração do Membership II

- Os mecanismos de protecção utilizados pelo *provider* também podem ser configurados:

```
<connectionStrings>
  <add name="SqlServices" connectionString="..." />
</connectionStrings>
<membership defaultProvider="SqlProvider">
  <providers>
    <add
      name="SqlProvider"
      type="System.Web.Security.SqlMembershipProvider"
      connectionStringName="SqlServices"
      applicationName="MyApplication"
      enablePasswordRetrieval="false"
      enablePasswordReset="true"
      requiresQuestionAndAnswer="true"
      requiresUniqueEmail="false"
      passwordFormat="Hashed"
      maxInvalidPasswordAttempts="5"
      minRequiredPasswordLength="7"
      minRequiredNonalphanumericCharacters="1"
      passwordAttemptWindow="10"
      passwordStrengthRegularExpression="" />
    </providers>
  </membership>
```

Protecção contra  
ataques.

Formatos aceites:  
Clear, Encrypted e Hashed.

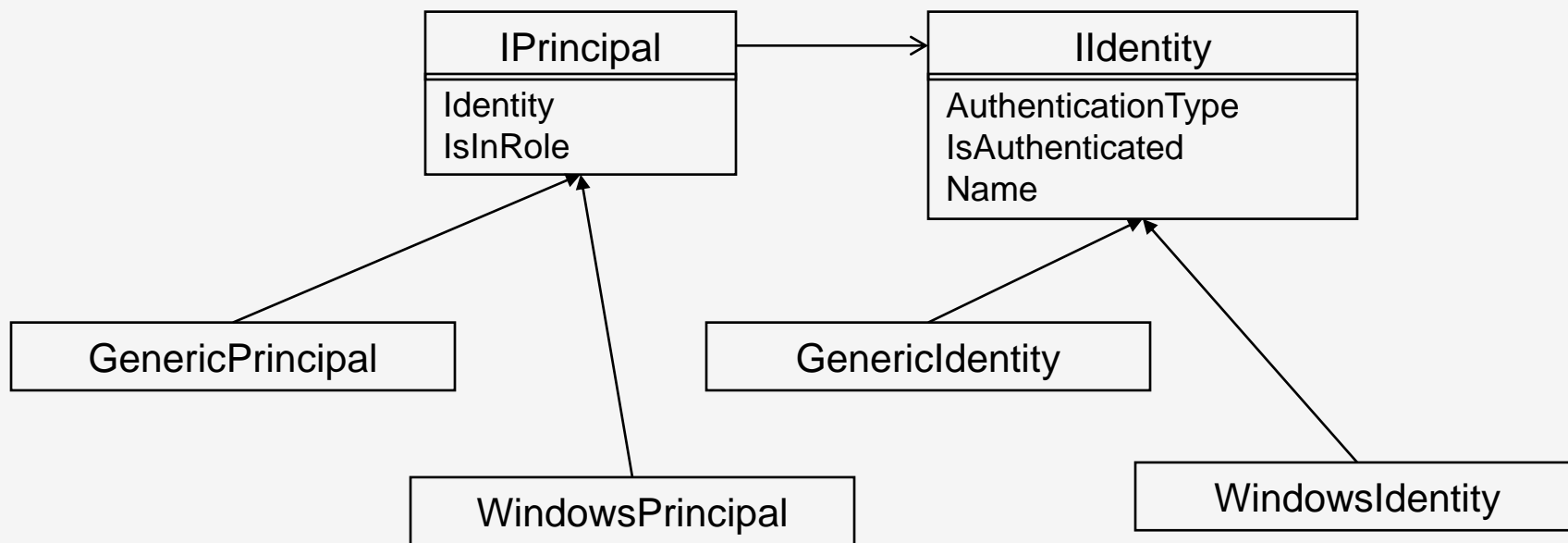
# Classe Membership

---

- A funcionalidade do serviço é oferecida através da classe estática *Membership*, onde:
  - A configuração do *provider* é acessível através de propriedades:
    - `Membership.HashAlgorithmType`
    - `Membership.MinRequiredPasswordLength`
    - `Provider`
    - ...
  - A gestão e autenticação de utilizadores é acessível através dos métodos da classe:
    - `Membership.CreateUser(username, password,...)`
    - `Membership.DeleteUser(username)`
    - `Membership.GetUser(username)`
    - ...
    - `Membership.ValidateUser(username, password)`

# Como representar e guardar a informação de autenticação?

- `IPrincipal` representa o contexto de segurança do utilizado, nomeadamente a sua identidade e os *roles* a que está a associado
- `IIdentity` representa a identidade de um utilizador e, opcionalmente, a forma como ele se autenticou
- Informação de autenticação pode ser guardada em:
  - `System.Threading.Thread.CurrentPrincipal` (consola e aplicações com GUI)
  - `System.Web.HttpContext.User` (ASP.NET)



# Exemplo: SqlMembershipProvider

oleApplicationMembership - Microsoft Visual Studio

Edit View Project Build Debug Data Query Designer Tools Test Window Help

Debug Any CPU

Server Explorer

- Data Connections
  - eagle.asppnetdb.db
  - eagle\sqlxpess.teste.db
- Database Diagrams
- Tables
  - aspnet\_Applications
  - aspnet\_Membership
  - aspnet\_Paths
  - aspnet\_PersonalizationAllUsers
  - aspnet\_PersonalizationPerUser
  - aspnet\_Profile
  - aspnet\_Roles
  - aspnet\_SchemaVersions
  - aspnet\_Users
    - ApplicationId
    - UserId
    - UserName
    - LoweredUserName
    - MobileAlias
    - IsAnonymous
    - LastActivityDate
  - aspnet\_UsersInRoles
  - aspnet\_WebEvent\_Events

dbo.vw\_aspnet\_Use...qlxpess.teste)\*

aspnet\_Users

- \* (All Columns)
- ☒ ApplicationId
- ☒ UserId
- ☒ UserName
- ☒ LoweredUserName
- ☒ MobileAlias
- ☒ IsAnonymous
- ☒ LastActivityDate

	Column	Alias	Table	Outp...	Sort Type	Sort Order	Filter	Or...	Or...
	ApplicationId		aspnet_Users						
	SELECT ApplicationId, UserId, UserName, LoweredUserName, MobileAlias, IsAnonymous, LastActivityDate								
	FROM dbo.aspnet_Users								
	ApplicationId	UserId	UserName	LoweredUserName	MobileAlias	IsAnonymous	LastActivityDate		
▶	b36-5405d1399feb	57100c9f-e26d-480a-...	Jose	jose	NULL	False	26-10-2008 09:00:12		
	575b1679-f157-41...	16ba10cd-5ded-4b92...	Pedro	pedro	NULL	False	26-10-2008 09:00:12		
	575b1679-f157-41...	fb8c8764-de2a-4d7f-...	Rui	rui	NULL	False	26-10-2008 09:00:29		
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL		

```
Membership.CreateUser("Rui", "rui@pass", "rui@email.pt", "question", "answer", true, out status);
Membership.CreateUser("Jose", "jose@pass", "jose@email.pt", "question", "answer", true, out status);
Membership.CreateUser("Pedro", "pedro@pass", "pedro@email.pt", "question", "answer", true, out status);
```

# Exemplo: SqlMembershipProvider

ApplicationMembership - Microsoft Visual Studio

View Project Build Debug Data Query Designer Tools Test Window Help

Debug Any CPU

explorer

- eagle\sqlxpress.teste.dbo
  - Database Diagrams
  - Tables
    - aspnet\_Applications
    - aspnet\_Membership
      - ApplicationId
      - UserId
      - Password
      - PasswordFormat
      - PasswordSalt
      - MobilePIN
      - Email
      - LoweredEmail
      - PasswordQuestion
      - PasswordAnswer
      - IsApproved
      - IsLockedOut
      - CreateDate
      - LastLoginDate
      - LastPasswordChangedDate
      - LastLockoutDate
      - FailedPasswordAttemptCount
      - FailedPasswordAttemptWindowStart
      - FailedPasswordAnswerAttemptCount
      - FailedPasswordAnswerAttemptWindowStart
      - Comment

dbo.vw\_aspnet\_Us...sqlxpress.teste) | dbo.vw\_aspnet\_Me...l...xpress.teste)\* | AssemblyInfo.cs | MyRoleProvider.cs | App.config | Program.cs

aspnet\_Membership

- ☐ PasswordQuestion
- ☐ PasswordAnswer
- ☐ IsApproved
- ☐ IsLockedOut
- ☐ CreateDate

aspnet\_Users

- ☐ \* (All Columns)
- ☐ ApplicationId
- ☐ UserId
- ☒ UserName
- ☐ LoweredUserName

Column	Alias	Table	Output	Sort Type	Sort Order
Password		aspnet_Membership	<input checked="" type="checkbox"/>		
PasswordSalt		aspnet_Membership	<input checked="" type="checkbox"/>		

```
SELECT      dbo.aspnet_Membership.UserId, dbo.aspnet_Membership.PasswordFormat, dbo.aspnet_Membership.FailedPasswordAttemptCount,
            dbo.aspnet_Users.UserName, dbo.aspnet_Membership.Password, dbo.aspnet_Membership.PasswordSalt
FROM        dbo.aspnet_Membership INNER JOIN
            dbo.aspnet_Users ON dbo.aspnet_Membership.UserId = dbo.aspnet_Users.UserId
```

UserId	PasswordFormat	FailedPasswordAttemptCount	UserName	Password	PasswordSalt
57100c9f-e26d-...	1	0	Jose	zNY5rLlaxWWNnERhbPEdHuHitDs=	JWGsvXcT1u2RWOerqzwWNQ==
16ba10cd-5ded-...	1 (Hashed)	0	Pedro	YiLo9+PKIOLEYSb/asOdIqPuBYE=	jGHHHqpNagBQC049gAWBg==
fb8c8764-de2a-...	1	0	Rui	wSF/H4G79vNOfhBOY1fvAz+uN4E=	+qSGU5UV8XejGweXptNxxA==

```
Membership.CreateUser("Rui", "rui@pass", "rui@email.pt", "question", "answer", true, out status);
Membership.CreateUser("Jose", "jose@pass", "jose@email.pt", "question", "answer", true, out status);
Membership.CreateUser("Pedro", "pedro@pass", "pedro@email.pt", "question", "answer", true, out status);
```