


ADO.NET (2/2)

*Herramientas Avanzadas para el Desarrollo de
Aplicaciones*

Escola Politècnica Superior
Universitat d'Alacant

- 
1. Repàs
 2. Acceso desconectado a BD
 3. Controls de dades
 4. Concurrència
 5. Accés connectat vs desconectat



1

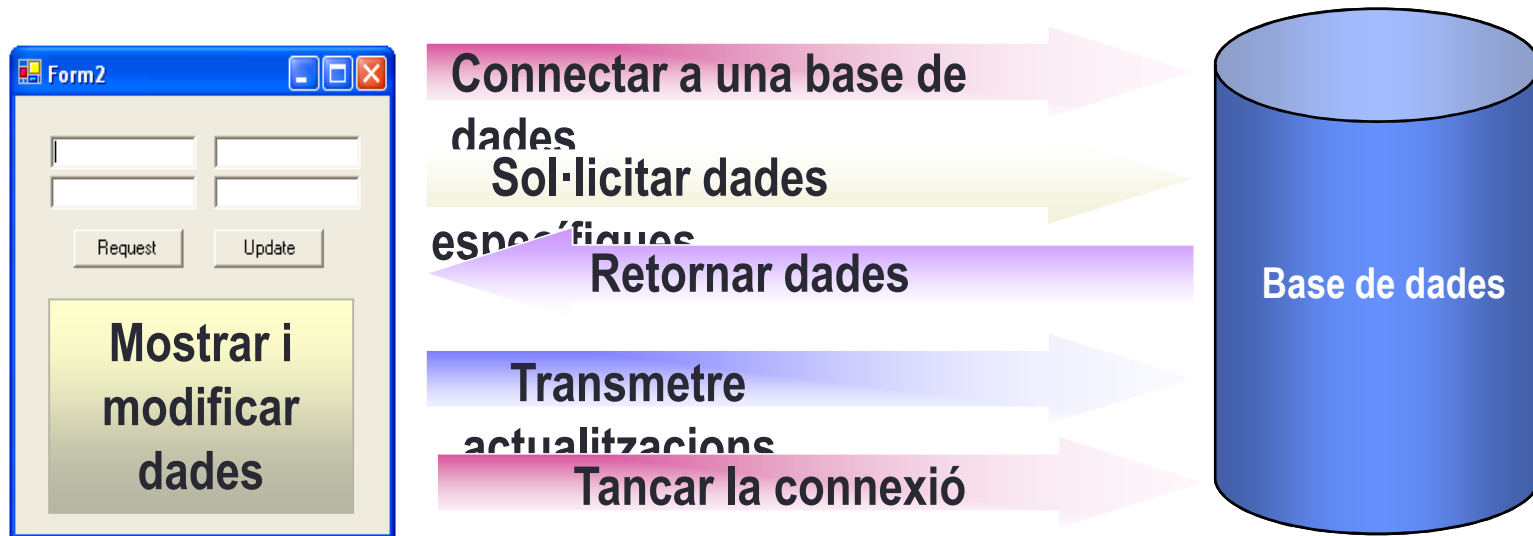
Repàs

Objectes d'accés a dades (*ActiveX Data Objects*)

- ADO.NET és la tecnologia que les aplicacions .net utilitzen per comunicar-se amb la BD.
- Optimitzada per a aplicacions distribuïdes (com a aplicacions web).
- Basada en XML
- Model d'objectes completament nou.
- **Entorn connectat vs desconnectat.**

Entorn connectat

CONNEXIÓ OBERTA



SENSE CONNEXIÓ

Entorn connectat (II)

- Un entorn connectat és aquell que els usuaris estan connectats contínuament a una font de dades
- Avantatges:
 - L'entorn és més fàcil de mantenir
 - La concurrència es controla més fàcilment
 - És més probable que les dades estiguin més actualitzats que en altres escenaris
- Inconvenients:
 - Ha d'existir una connexió de xarxa constant
 - Escalabilidad limitada

Entorn desconnectat

CONNEXIÓ OBERTA

Form2

Request Update

Mostrar i
modificar
dades

Connectar a una base de dades

Sol·licitar dades específiques

Retornar dades

Tancar la connexió

Connectar a una base de dades

Transmetre actualitzacions

Tancar la connexió

Base de dades

SENSE CONNEXIÓ

Entorn desconnectat (II)

- Un entorn desconnectat és aquell en el qual les dades poden modificar-se de forma independent i els canvis s'escriuen posteriorment en la base de dades.
- Avantatges:
 - Les connexions s'utilitzen durant el menor temps possible, permetent que menys connexions donin servei a més usuaris
 - Un entorn desconnectat millora la escalabilidad i el rendiment de les aplicacions
- Inconvenients:
 - Les dades no sempre estan actualitzats
 - Poden produir-se conflictes de canvis que han de solucionar-se

Taula client...

WindowsApplication1 - Microsoft Visual Studio

File Edit View Project Build Debug Data Table Designer Tools Window Community Help

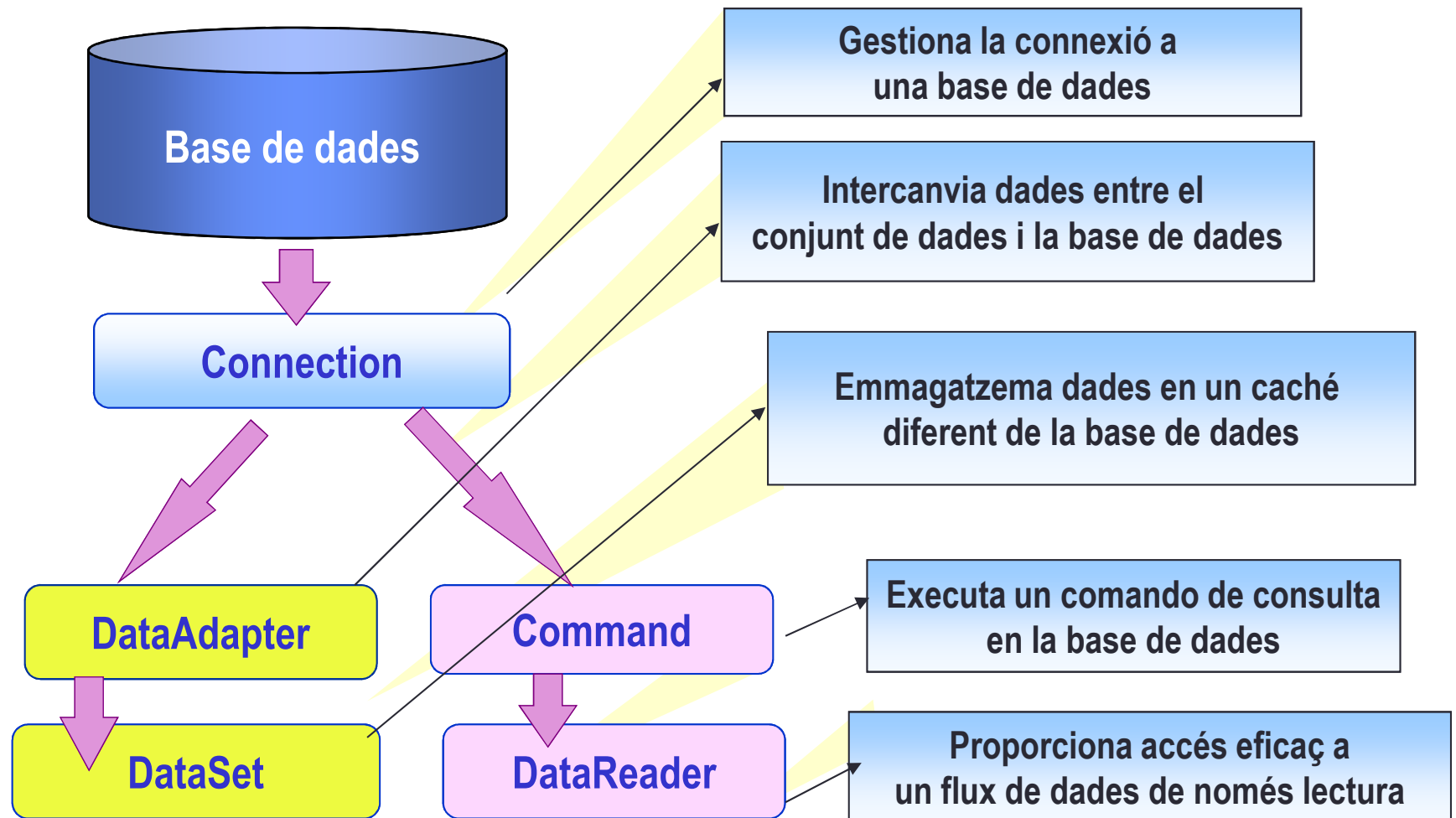
Server Explorer

- Data Connections
 - irene1\sqlexpress.prueba.dbo
 - irene1\sqlexpress.prueba1.dbo
 - Database Diagrams
 - Tables
 - cliente
 - usuario
 - contraseña
 - dni
 - Views
 - Stored Procedures
 - Functions
 - Synonyms
 - Types
 - Assemblies
 - Servers
 - irene1

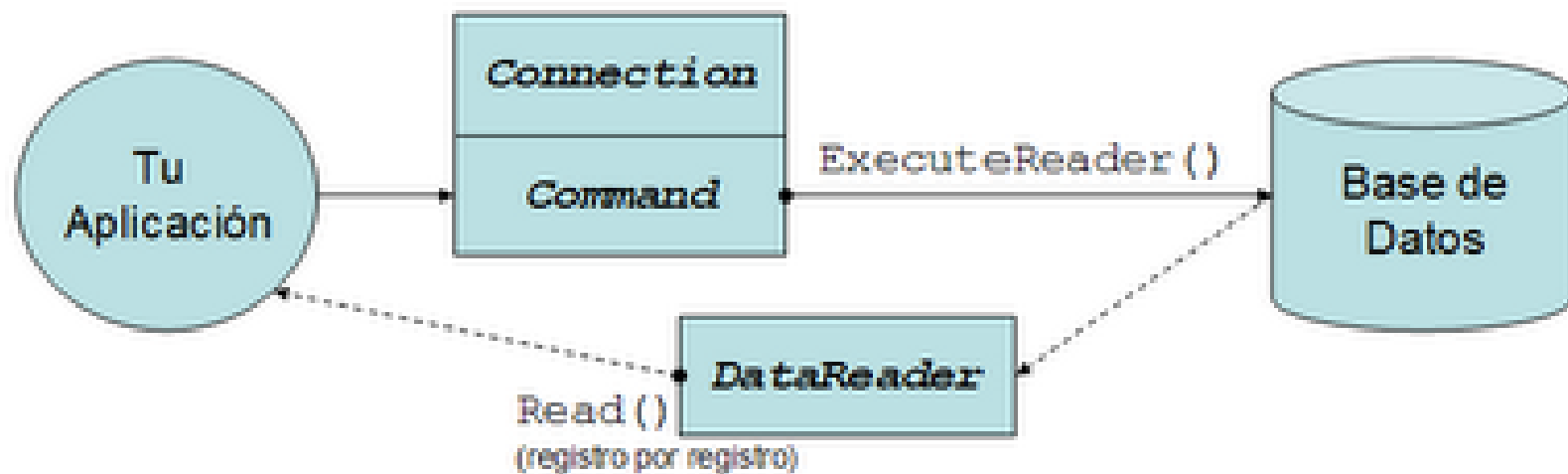
dbo.cliente: Tabl...qlexpress.prueba1) Form1.vb [Design] Start Page

Column Name	Data Type	Allow Nulls
usuario	nchar(10)	<input checked="" type="checkbox"/>
contraseña	nchar(10)	<input checked="" type="checkbox"/>
dni	nchar(10)	<input type="checkbox"/>
		<input type="checkbox"/>

Objetos de ADO.NET



Manera conectada





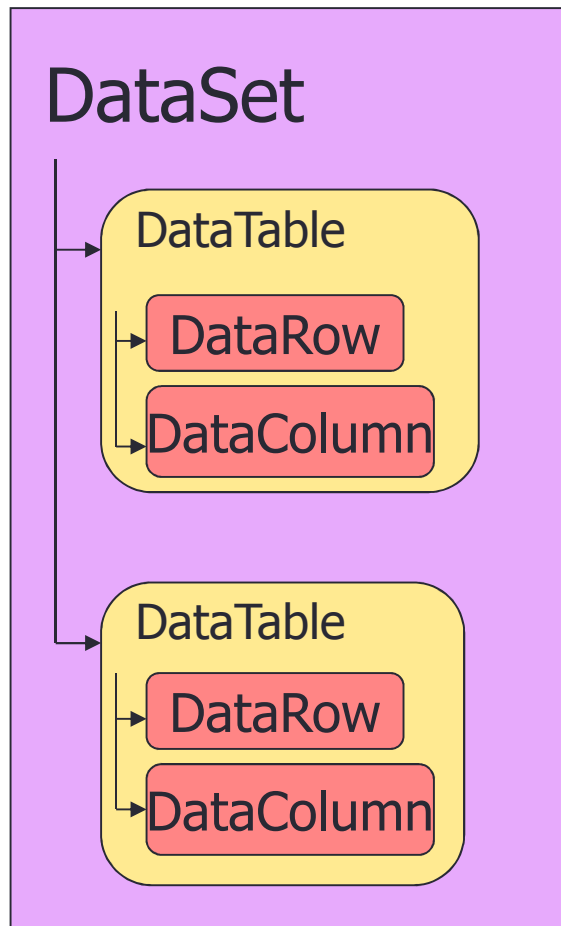
2

Accès
desconnectat

1. Objecte Connection

- **Connection:** s'utilitza per establir les connexions al proveïdor de dades adequat (mètode Open).

2. Objecte DataSet



- Nou objecte **DataSet**: representació d'una **base de dades** relacional **en memòria**:
 - No necessitat de connexió contínua.

Objecte DataSet (II)

- Podem treballar amb una BD que és còpia de les parts amb les quals volem treballar de la BD real, alliberant la connexió.
- Si volem reflectir els canvis en la BD real, hem de confirmar el nostre objecte DataSet.

Objetos DataRow, DataColumn

- DataRow
 - Representa una única fila d'informació de la taula.
 - Es pot accedir als valors individuals usant el nom de camp o un índex.
- DataColumn
 - No contenen cap dada real.
 - Emmagatzemen informació sobre la columna (tipus de dades, valor predeterminat, restriccions..)

Objetos DataRelation, DataView

- **DataRelation**
 - Especifica una relació pare/fill entre dues taules diferents d'un objecte DataSet.
- **DataView**
 - Proporciona una vista sobre una DataTable.
 - Cada DataTable té almenys un DataView (a través de la propietat DefaultView), que s'usa per a la vinculació de dades.
 - DataView mostra les dades de DataTable sense canvis o amb opcions especials de filtre o ordenació.

3. Objecto Adaptador

- L'objecte Adaptador de dades s'encarrega de manejar la connexió per nosaltres.
- S'utilitza per inserir dades en un objecte **DataSet**.
- L'objecte **DataAdapter** utilitza comandos per actualitzar l'origen de dades després de fer modificacions en l'objecte **DataSet**.

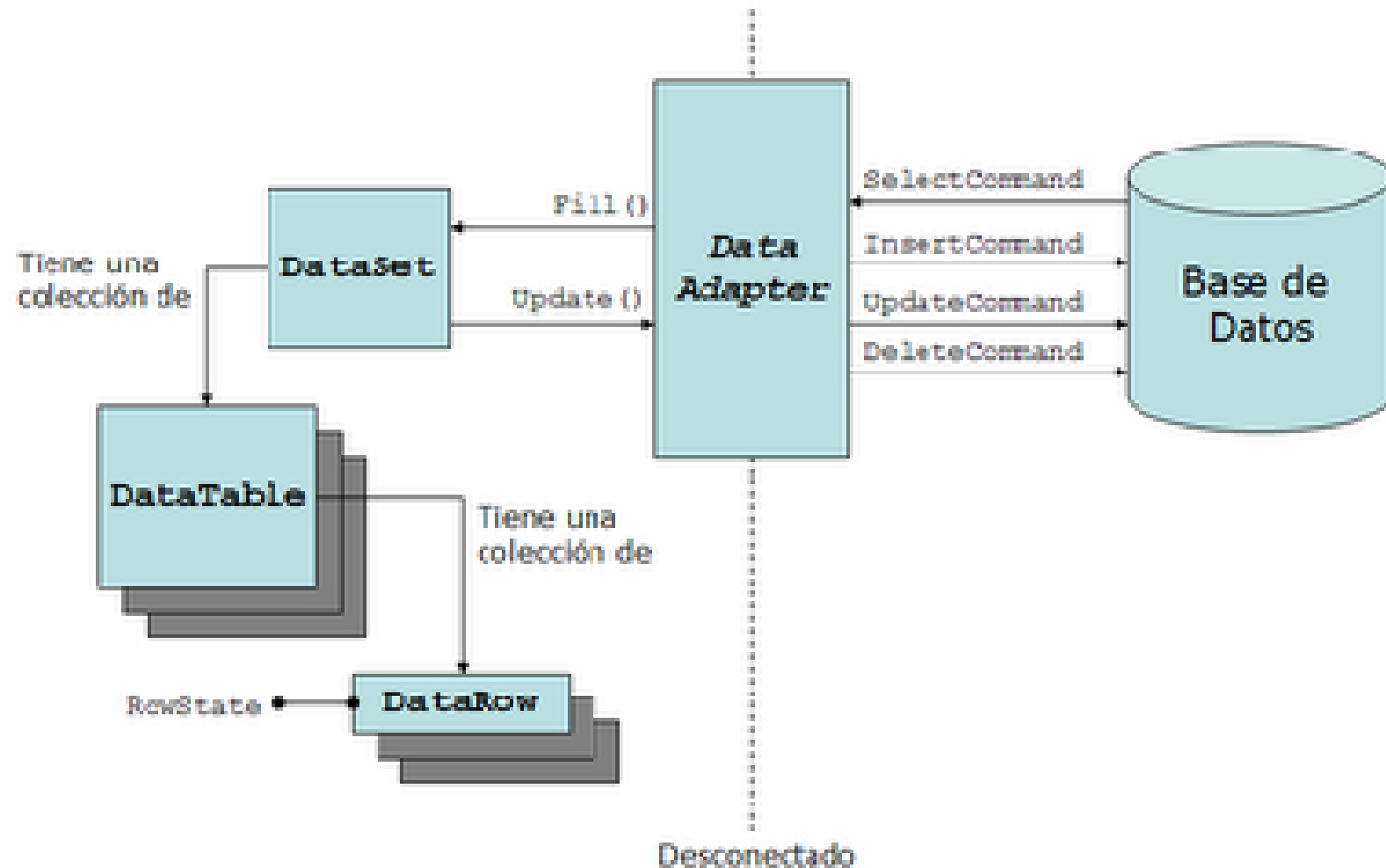
DataAdapter (II)

- Un avantatge en usar un DataAdapter és que **no haig de preocupar-me per obrir i tancar la connexió**. Aquests mètodes ho fan automàticament

4. Objecte CommandBuilder

- Aquest objecte (opcional) ho utilitza el **DataAdapter** per crear els comandos SQL necessaris.
- També es poden proporcionar les sentències SQL explícitament o per mitjà de procediments emmagatzemats.

Manera desconectada



Recordeu...

- `System.Data.OleDb` i `System.Data.SqlClient`: classes responsables de l'accés a dades des de fonts SQL Server i OLE DB.
- Inclouen classes que en treballar amb SQL portaran el prefix `Sql` i en emprar Ole DB portaran `OleDb`:
 - `SqlConnection` i `OleDbConnection`
 - `SqlDataAdapter` y `OleDbDataAdapter`
 - `SqlCommandBuilder` y `OleDbCommandBuilder`
 - **PERÒ NO AMB `DataSet` (ni `dataRow`, `dataColumn`...)**

EXAMPLE: inserció de dades

Usuario	<input type="text" value="luis"/>
Contraseña	<input type="text" value="1256"/>
DNI	<input type="text" value="55555"/>
<input type="button" value="Insertar usuario"/>	

Conexión a una BD en Sql Server

Importar namespaces

```
using System.Data;  
using System.Data.Common;  
using System.Data.SqlClient;  
using System.Data.SqlTypes;
```


Crear la connexió


```
string s = "data source=.\SQLEXPRESS;Integrated  
Security=SSPI;AttachDBFilename=|DataDirectory|\\Database1.mdf;  
User Instance=true";  
SqlConnection c=new SqlConnection(s);
```

Definició d'un comando Select

- Per recuperar les dades es necessita:
 - Una sentència SQL que seleccioni la informació desitjada
 - Un objecte Command que executi la sentència SQL
 - Un objecte DataReader que capturi els registres recuperats

ATENCIÓ: En manera desconnectada sempre necessitem recuperar les dades (SELECT) per poder treballar amb ells (INSERT, UPDATE, DELETE)

Definició d'un comando Select

- Per recuperar les dades es necessita:
 - ~~Una sentència SQL que seleccioni la informació desitjada~~
 - ~~Un objecte Command que executi la sentència SQL~~
 - ~~Un objecte DataReader que capturi els registres recuperats~~
- 
- Un objecte **DataAdapter** que executi la sentència SQL
 - Un objecte **DATASET** on guardar el resultat de la sentència SQL

Objectes DataSet i DataAdapter

- Creuem una BD virtual, mitjançant un objecte DataSet

```
DataSet bdvirtual = new DataSet();
```

- L'omplim amb les taules que estem interessats a treballar:
 - Objecte DataAdapter
 - Mètode Fill()

...

```
SqlDataAdapter da = new  
SqlDataAdapter("select * from Cliente", c);  
  
da.Fill(bdvirtual,"cliente");
```

Ara treballem en local

- Ara en “bdvirtual” tenim la nostra base de dades local.

Per treballar en local

- Ho fem modificant files i columnes de les taules emmagatzemades en local.
- En dataset tenim emmagatzemada la bd virtual, copiem a un datatable la taula a modificar.

```
DataTable t = new DataTable();  
t = bdvirtual.Tables["cliente"];
```

Operacions

- Obtenir una taula:
 - `DataTable t = new DataTable();`
 - `t = bdvirtual.Tables["cliente"];`
- Accedir als elements files d'aquesta taula (podem usar un bucle):
 - `DataRow fila = t.Rows[0];`
 - `fila[0] = "Andrés";`
 - **IGUAL QUE**
 - `t.Rows[0][0] = "Andrés";`

Primera fila segona columna:

- `t.Rows[0][1]`
- Informació de les columnes: (nom, tipus)
 - `t.Columns[0].ColumnName`
 - `t.Columns[0].DataType`

Volem inserir un nou client

- Això equival a inserir una fila en la nostra taula local...

```
DataRow nuevafila = t.NewRow();  
nuevafila[0] = textBox1.Text;  
nuevafila[1] = textBox2.Text;  
nuevafila[2] = textBox3.Text;  
t.Rows.Add(nuevafila);
```

Validar los cambios

- **Objecto Adaptador:**
 - ens ha servit per omplir la taula, també per actualitzar les dades en la BD real.
- **Mètode Update**
 - El *DataAdapter* analitzarà els canvis que s'han fet en el *DataSet* i executarà els comandos apropiats per *inserir*, *actualitzar* o *esborrar* en la BD real.

Constructor de comandos

- Objeto CommandBuilder:
 - Constructor de comandos
 - Li passem com a argument l'adaptador
 - Construeix els comandos SQL que ens facin mancada per actuar sobre la BD

- SqlCommandBuilder cbuilder = new SqlCommandBuilder(da);
- da.Update(bdvirtual, "cliente");
- label4.Text = "Cambiado";

```
string s = "data source=.\SQLEXPRESS;Integrated  
Security=SSPI;AttachDBFilename=|DataDirectory|\\Database1.mdf;User  
Instance=true";
```

```
SqlConnection c=new SqlConnection(s);
```

```
DataSet bdvirtual = new DataSet();
```

```
SqlDataAdapter da = new SqlDataAdapter("select * from Cliente", c);
```

```
da.Fill(bdvirtual, "cliente");
```

```
DataTable t = new DataTable();
```

```
t = bdvirtual.Tables["cliente"];
```

```
DataRow nuevafila = t.NewRow();
```

```
nuevafila[0] = textBox1.Text;
```

```
nuevafila[1] = textBox2.Text;
```

```
nuevafila[2] = textBox3.Text;
```

```
t.Rows.Add(nuevafila);
```

```
SqlCommandBuilder cbuilder = new SqlCommandBuilder(da);
```

```
da.Update(bdvirtual, "cliente");
```

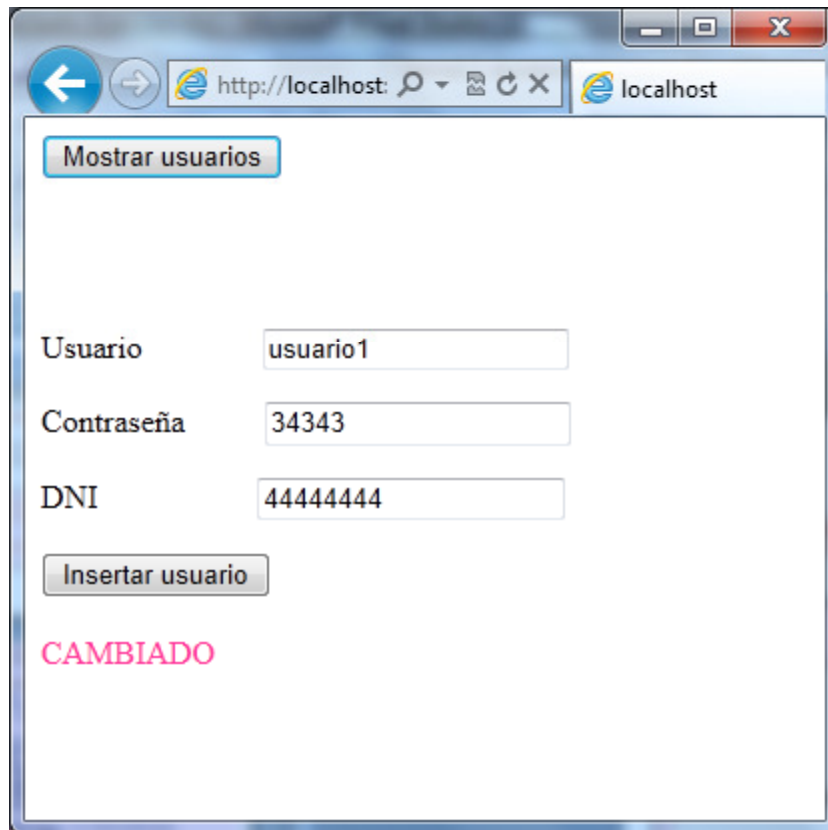
```
label4.Text = "CAMBIADO";
```

Què faltaria en el codi?

En les 3 capes, modifiquem la funció CAD

```
public bool InsertarCliente(ENCliente cli)    {
    bool cambiado;
    ENCliente cl = cli;
    DataSet bdvirtual = new DataSet() ;
    SqlConnection c = new SqlConnection(s);
    try    {
        SqlDataAdapter da = new SqlDataAdapter("select * from Cliente",c);
        da.Fill(bdvirtual,"cliente");
        DataTable t = new DataTable();
        t = bdvirtual.Tables["cliente"];
        DataRow nuevafila = t.NewRow();
        nuevafila[0] = cl.Usuario;
        nuevafila[1] = cl.Contraseña;
        nuevafila[2] = cl.Dni;
        t.Rows.Add(nuevafila);
        SqlCommandBuilder cbuilder = new SqlCommandBuilder(da);
        da.Update(bdvirtual, "cliente");
        cambiado = true;    }
    catch (Exception ex) {    cambiado = false;    }
    finally    {    c.Close();    }
    return cambiado; }
```

Execució



A screenshot of a web browser window showing a user management interface. The browser's address bar displays 'http://localhost:'. The page has a title bar with standard window controls. The main content area includes a button labeled 'Mostrar usuarios' at the top. Below it, there are three input fields: 'Usuario' with the value 'usuario1', 'Contraseña' with the value '34343', and 'DNI' with the value '44444444'. At the bottom of this section is a button labeled 'Insertar usuario'. Below the buttons, the word 'CAMBIADO' is displayed in pink text.

Mostrar usuarios

Usuario

Contraseña

DNI

Insertar usuario

CAMBIADO



A screenshot of a SQL Server Enterprise Manager window showing a table view. The window title is 'cliente: consulta...TA\DATABASE1.MDF) Web.config'. The table has four columns: 'usuario', 'contraseña', and 'dni'. The first row is highlighted in blue and contains the values 'irenee', '345', and '4343'. The second row contains 'usuario1', '34343', and '44444444'. The third row contains 'laura', '123', and '45698'. The fourth row contains 'irene', '123', and '45896'. The fifth row contains 'luis', '1256', and '55555'. The sixth row contains 'jose', '258', and '85964'. The seventh row contains 'ii', '343', and 'rr'. The eighth row is a summary row with a '*' in the first column and 'NULL' in the other three columns.

	usuario	contraseña	dni
►	irenee	345	4343
	usuario1	34343	44444444
	laura	123	45698
	irene	123	45896
	luis	1256	55555
	jose	258	85964
	ii	343	rr
*	NULL	NULL	NULL



3

Control GridView

Control GridView

- Control per a presentació de dades en forma de taula (files i columnes)
- Propietats:
 - Selecció
 - Paginació
 - Ordenació
 - Edició
 - Extensible mitjançant plantilles
- [http://msdn.microsoft.com/es-es/library/cc295223\(v=expression.40\).aspx](http://msdn.microsoft.com/es-es/library/cc295223(v=expression.40).aspx)

GridView

The screenshot displays the Visual Studio IDE with the 'Datos' (Data) task pane on the left. The 'GridView' control is selected, and its 'Tareas de GridView' (GridView Tasks) pane is open on the right. The GridView control is shown with a table structure containing three columns (Column0, Column1, Column2) and six rows of data (all 'abc'). Below the GridView, the 'SqlDataSource - SqlDataSource1' control is visible, with its 'Tareas de SqlDataSource' (SqlDataSource Tasks) pane open, showing the 'Configurar origen de datos...' (Configure data source...) option.

Datos

- Puntero
- GridView
- DataList
- DetailsView
- FormView
- ListView
- Repeater
- DataPager
- SqlDataSource
- AccessDataSource
- LinqDataSource
- ObjectDataSource
- XmlDataSource
- SiteMapDataSource

asp:gridview#GridView1

Column0	Column1	Column2
abc	abc	abc
abc	abc	abc
abc	abc	abc
abc	abc	abc
abc	abc	abc
abc	abc	abc

Tareas de GridView

- Formato automático...
- Elegir origen de datos: (Ninguno) ▼
- Editar columnas...
- Agregar nueva columna...
- Editar plantillas

asp:sqldatasource#SqlDataSource1

SqlDataSource - SqlDataSource1

Tareas de SqlDataSource

- Configurar origen de datos...

Objectiu:

Anem a mostrar en un gridview les dades de la taula client.

- *Amb l'assistent*
 - *Amb codi*

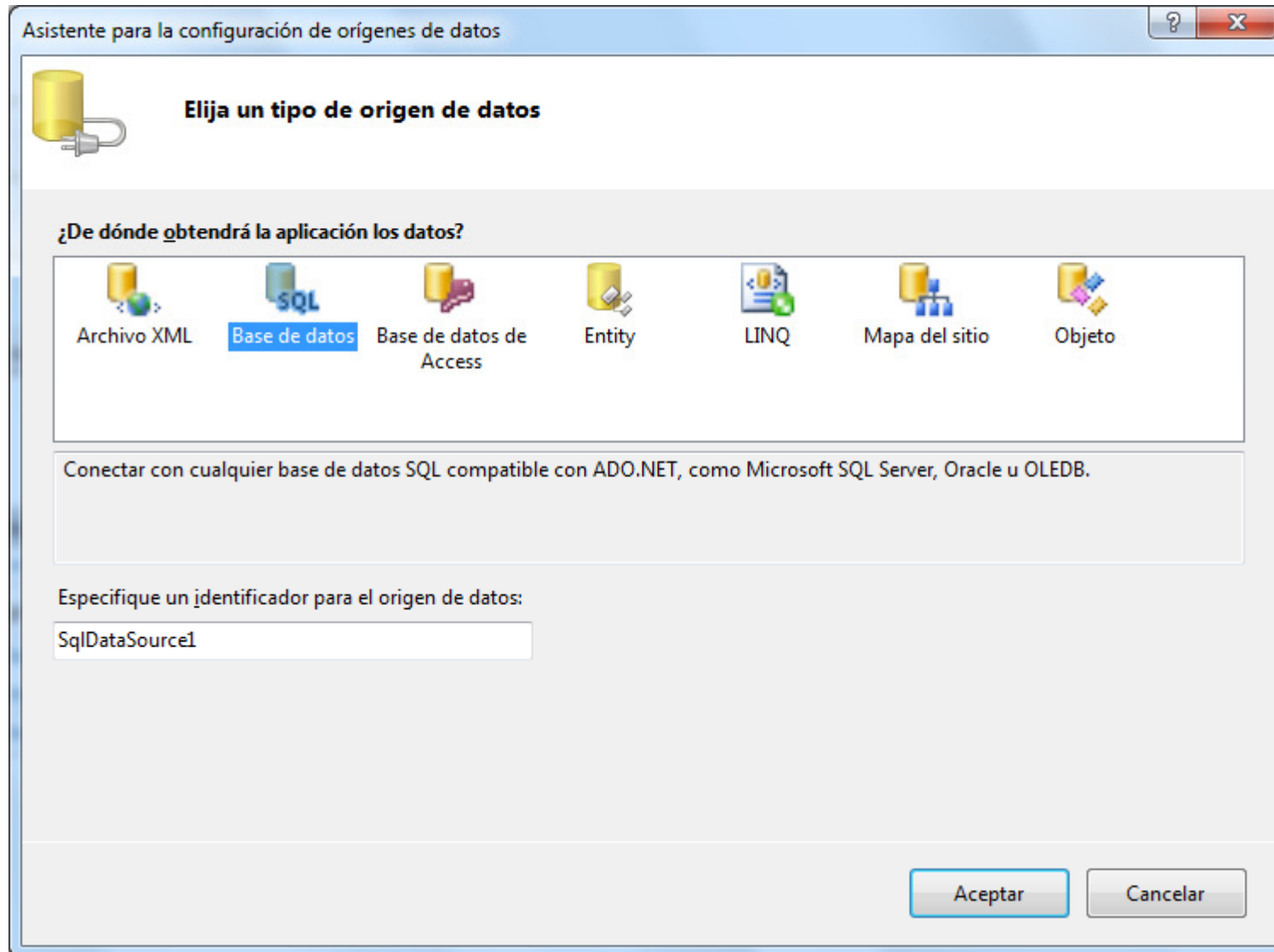
GridView → DataSource

- Afegir un objecte GridView i triar com a origen de dades la bd.

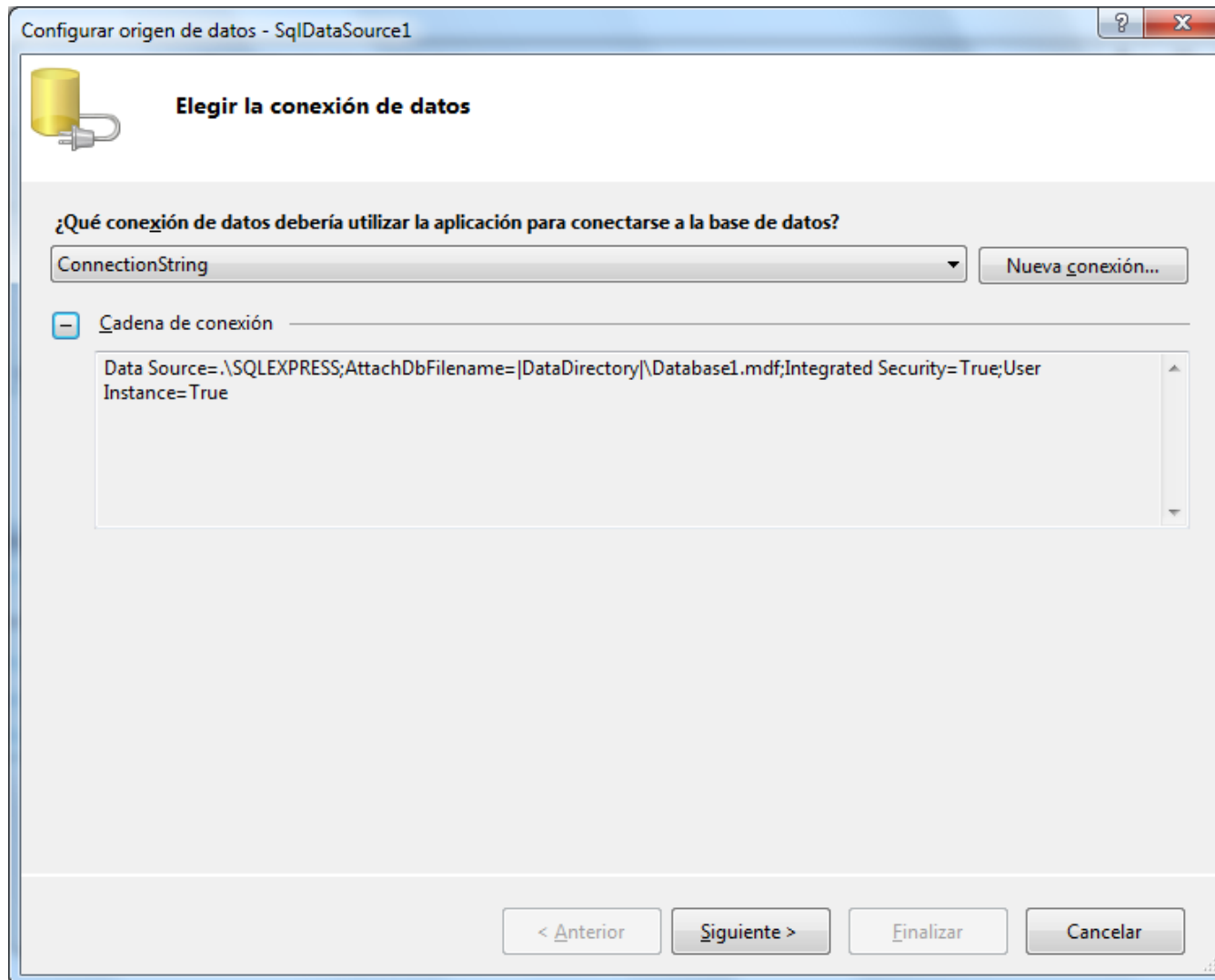
Control GridView

- Podem assignar-li una taula de la BD com a origen de dades, o els registres obtinguts com a resultat d'una sentència SQL.

Triar BD

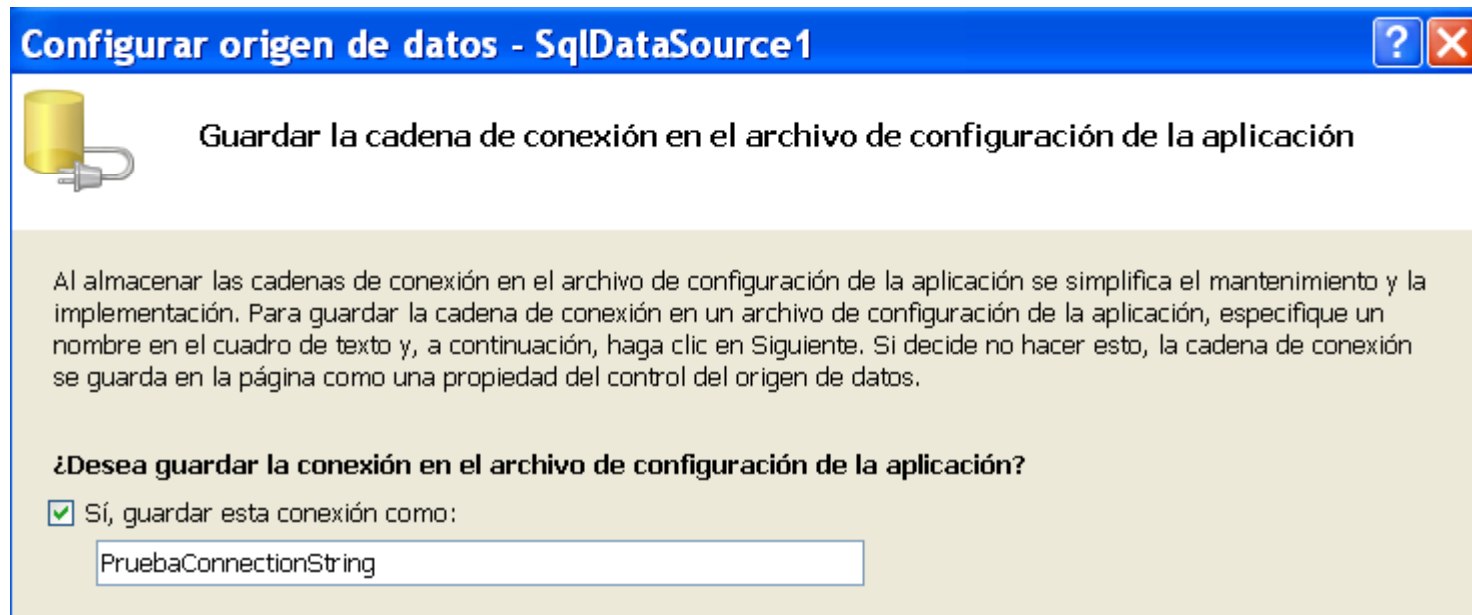


Triar connexió de dades




GridView

- Guardem la cadena de connexió en l'arxiu Web.config



Configurar origen de datos - SqlDataSource1

 **Configurar la instrucción Select**

¿Cómo desea recuperar los datos de la base de datos?

☐ Especificar una instrucción SQL o un procedimiento almacenado personalizado

☒ Especificar columnas de una tabla o vista

Nombre:

cliente

Columnas:

☒ *

☐ usuario

☐ contraseña

☐ dni

☐ Devolver sólo filas únicas

WHERE...

ORDER BY...

Avanzadas...

Instrucción SELECT:

SELECT * FROM [cliente]

< Anterior

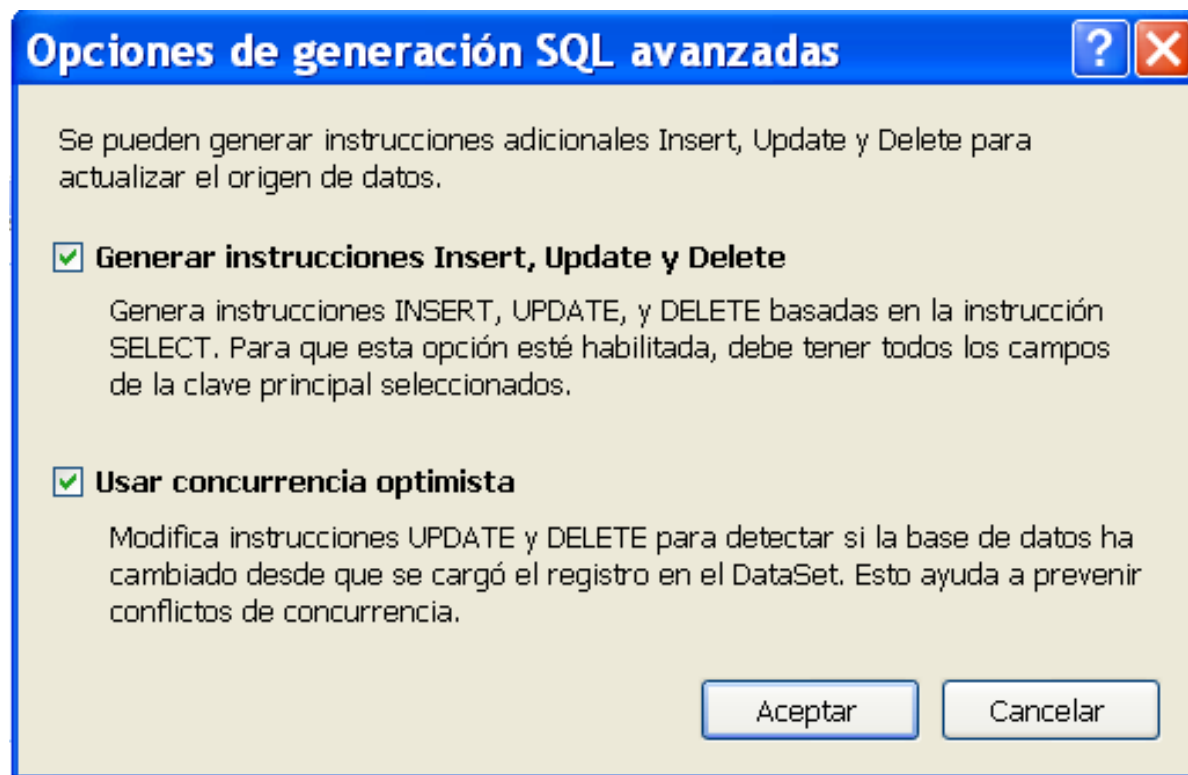
Siguiente >

Finalizar

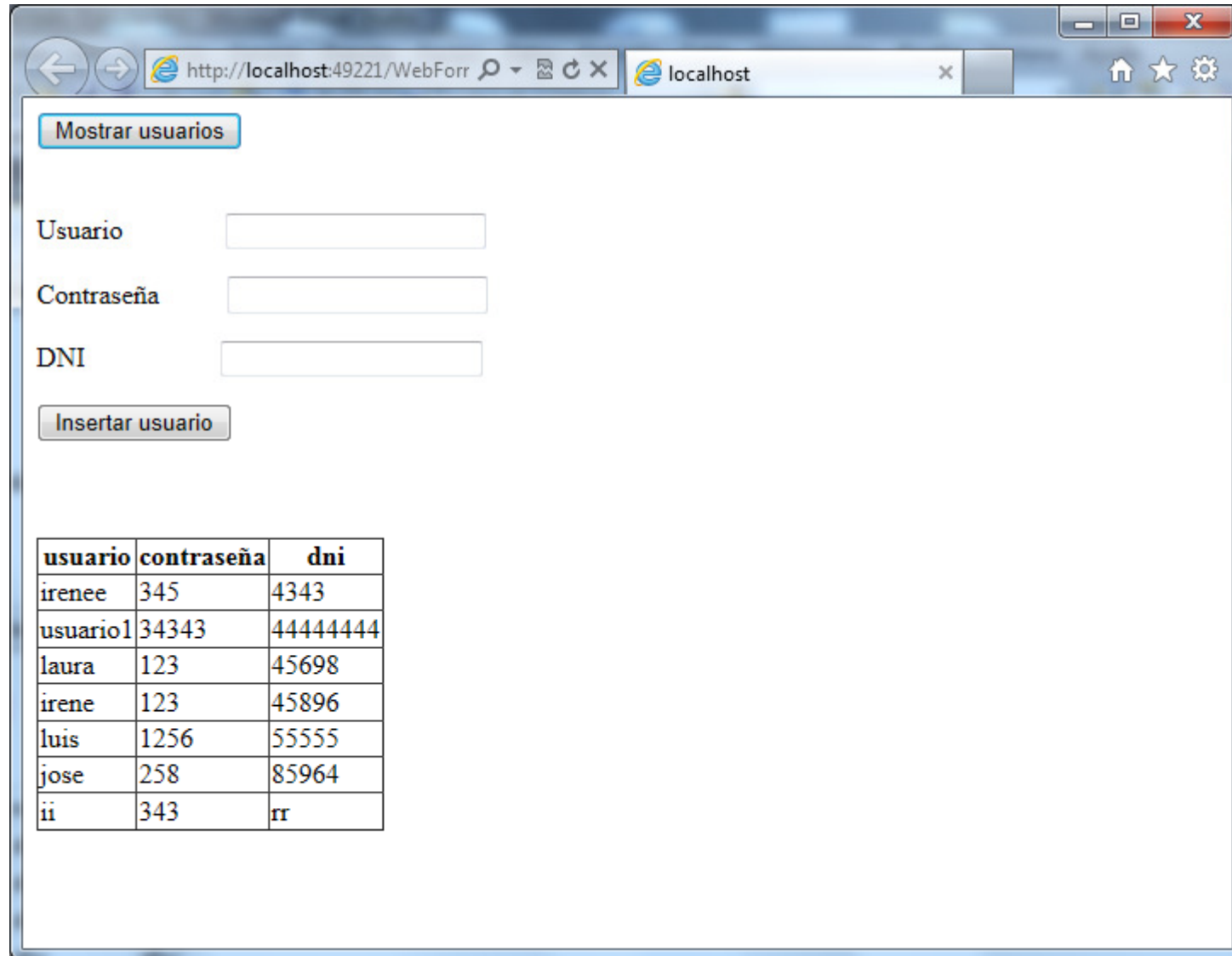
Cancelar

GridView (asistente)

- Avançades
 - Podem generar les instruccions Insert, Update i Delete



Execució...



Mostrar usuarios

Usuario

Contraseña

DNI

Insertar usuario

usuario	contraseña	dni
irenee	345	4343
usuario1	34343	44444444
laura	123	45698
irene	123	45896
luis	1256	55555
jose	258	85964
ii	343	rr

Codi per mostrar dades d'un dataset

```
ENCliente enl = new ENCliente();
DataSet d = new DataSet();

protected void Page_Load(object sender, EventArgs e)
{
    if (!Page.IsPostBack)
    {
        d = enl.listarClientesD();

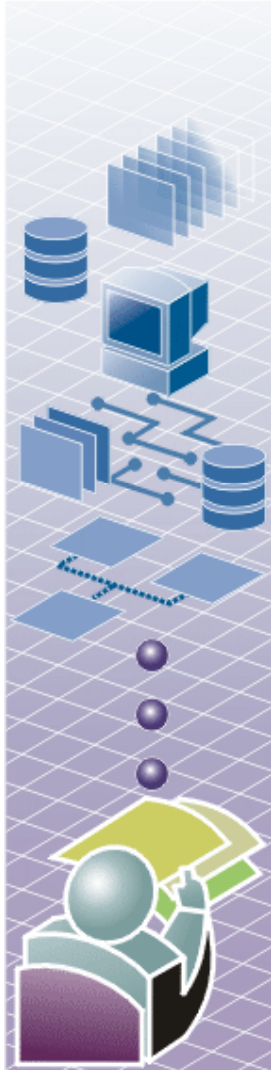
        GridView1.DataSource = d;
        GridView1.DataBind();
    }
}
```

On en el CAD...

```
public DataSet ListarClientesD()
{
    DataSet bdvirtual = new DataSet();

    SqlConnection c = new SqlConnection(s);
    SqlDataAdapter da = new SqlDataAdapter("select * from Cliente", c);
    da.Fill(bdvirtual, "cliente");
    return bdvirtual;
}
```

Exercici



Exercici

- Modificar el formulari anterior per editar les dades d'una persona en la BD.

5 min



AutoGenerateSelectButton=true

Usuario

Contraseña

DNI

Insertar usuario

Editar usuario

	<u>usuario</u>	<u>contraseña</u>	<u>dni</u>
Seleccionar	laura	1237	44444444
Seleccionar	laura2	1239	4569
Seleccionar	irene	123	45896
Seleccionar	luis	1256	55555
Seleccionar	jose	258	85964
1 2			

Esdeveniment SelectedIndexChanged

```
protected void GridView2_SelectedIndexChanged(object sender,  
EventArgs e)
```

```
{  
    TextBox1.Text = GridView2.SelectedRow.Cells[1].Text;  
    TextBox2.Text = GridView2.SelectedRow.Cells[2].Text;  
    TextBox3.Text = GridView2.SelectedRow.Cells[3].Text;  
    TextBox3.Enabled = false;  
}
```


Botó editar (clic)

```
{  
  
    ENCliente en = new ENCliente();  
    en.Usuario = TextBox1.Text;  
    en.Contraseña = TextBox2.Text;  
    d = en.ModificarCliente(GridView2.SelectedIndex);  
  
    GridView2.DataSource = d;  
    GridView2.DataBind();  
}
```



EN

```
public DataSet ModificarCliente(int i)
{
    CADcliente c = new CADcliente();
    DataSet a = c.ModificarCliente(this,i);
    return a;
}
```

CAD (simplificat)

```
public DataSet ModificarCliente(ENCliente cli, int i)
{
    ENCliente cl = cli;
    DataSet bdvirtual = new DataSet();
    SqlConnection c = new SqlConnection(s); SqlDataAdapter da = new
    SqlDataAdapter("select * from Cliente", c);
        da.Fill(bdvirtual, "cliente");
        DataTable t = new DataTable();
        t = bdvirtual.Tables["cliente"];

        t.Rows[i]["usuario"]=cl.Usuario;
        t.Rows[i]["contraseña"] = cl.Contraseña;

        SqlCommandBuilder cbuilder = new SqlCommandBuilder(da);
        da.Update(bdvirtual, "cliente");

        return bdvirtual;
}
```

Exercici



Exercici

- Modificar el formulari anterior per eliminar les dades d'una persona en la BD.

5 min



AutoGenerateDeleteButton=true

```
protected void GridView2_RowDeleting(object sender, GridViewDeleteEventArgs e)
{
    ENCliente en = new ENCliente();

    d = en.BorrarCliente(e.RowIndex);

    GridView2.DataSource = d;
    GridView2.DataBind();
}
```



EN

```
public DataSet BorrarCliente(int i)
{
    CADcliente c = new CADcliente();
    DataSet a = c.BorrarCliente(this, i);
    return a;
}
```

CAD

```
public DataSet BorrarCliente(ENCliente cli, int i)
{
    ENCliente cl = cli;
    DataSet bdvirtual = new DataSet();
    SqlConnection c = new SqlConnection(s);

    SqlDataAdapter da = new SqlDataAdapter("select * from Cliente", c);
    da.Fill(bdvirtual, "cliente");
    DataTable t = new DataTable();
    t = bdvirtual.Tables["cliente"];

    t.Rows[i].Delete();

    SqlCommandBuilder cbuilder = new SqlCommandBuilder(da);
    da.Update(bdvirtual, "cliente");

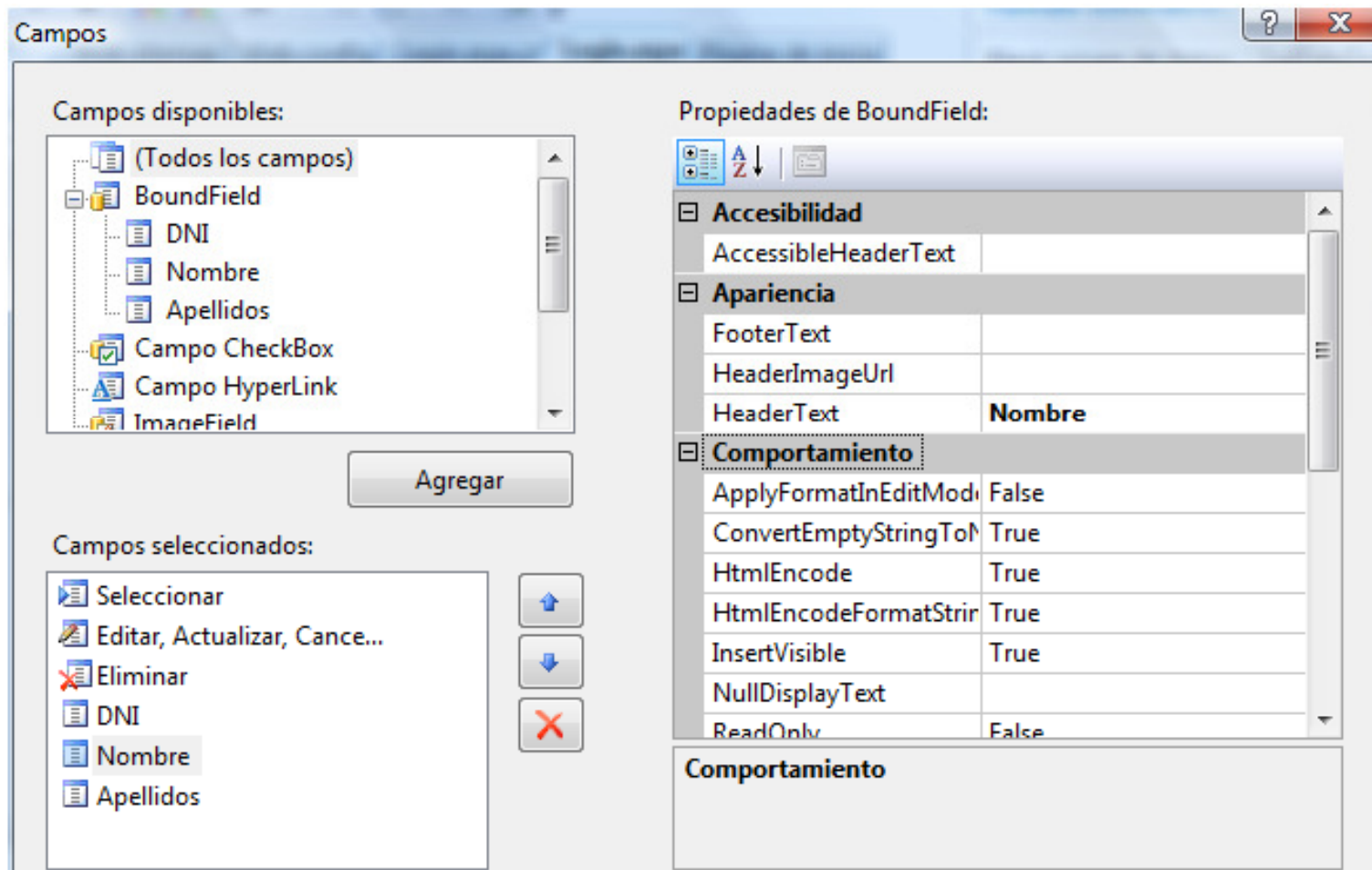
    return bdvirtual;    }
```

Paginació en GridView

- Propietats
 - AllowPaging = true
 - PageSize = 5 (numero d'elements per pàgina)
- En mostrar les dades amb assistent no cal escriure codi, però en enllaçar-ho nosaltres necessitem escriure el codi per a l'esdeveniment

```
protected void GridView2_PageIndexChanging(object sender,  
    GridViewPageEventArgs e)  
{  
    d = enl.listarClientesD();  
    GridView2.PageIndex = e.NewPageIndex;  
    GridView2.DataSource = d;  
    GridView2.DataBind();  
}
```


GridView: editar columnas



GridView

- Tipus de columnes:
 - BoundField: Mostra el text d'un camp de la BBDD
 - ButtonField: Mostra un botó per cada item
 - CheckBoxField: Mostra un checkbox per cada item
 - CommandField: Proporciona funcions de selecció, edició i esborrat
 - HyperLinkField: Mostra el text d'un camp de la BBDD com un hipervínculo
 - ImageField: Mostra una imatge
 - TemplateField: Permet especificar múltiples camps i controls personalitzats

GridView

- Modifiquem l'aspecte del GridView

asp:gridview#GridView1

			DNI	Nombre	Apellidos
Seleccionar	Editar	Eliminar	abc	abc	abc
Seleccionar	Editar	Eliminar	abc	abc	abc
Seleccionar	Editar	Eliminar	abc	abc	abc
Seleccionar	Editar	Eliminar	abc	abc	abc
Seleccionar	Editar	Eliminar	abc	abc	abc

1 2

SqlDataSource - SqlDataSource1

Tareas de GridView

- Formato automático...
- Elegir origen de datos: SqlDataSource1
- Configurar origen de datos...
- Actualizar esquema
- Editar columnas...
- Agregar nueva columna...
- ☒ Habilitar paginación
- ☒ Habilitar ordenación
- ☒ Habilitar edición
- ☒ Habilitar eliminación
- ☒ Habilitar selección
- Editar plantillas

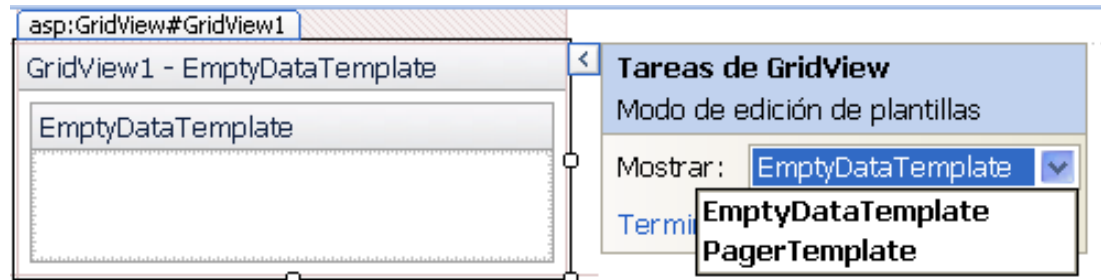
GridView

- Resultat final

			<u>DNI</u>	<u>Nombre</u>	<u>Apellidos</u>
<div>Actualizar</div> <div>Cancelar</div>			11111111	<div>Laura</div>	<div>Sanchez</div>
<div>Seleccionar</div>	<div>Editar</div>	<div>Eliminar</div>	22222222	Alberto	Lopez
<div>Seleccionar</div>	<div>Editar</div>	<div>Eliminar</div>	44444444	Juan	Perez
<div>Seleccionar</div>	<div>Editar</div>	<div>Eliminar</div>	55555555	Sara	Jover
<div>Seleccionar</div>	<div>Editar</div>	<div>Eliminar</div>	66666666	Berta	Belda
1 <u>2</u>					

Editar plantillas

- EmptyDataText
 - S'utilitza per mostrar un missatge quan no existeixen dades que mostrar en el GridView
- EmptyDataTemplate
 - Podem personalitzar el missatge mostrat quan el GridView està buit.



No se encontraron datos referentes
a su consulta. Inténtelo de nuevo.

Buscar



4

Concurrencia

Entorn desconnectat: conflictes

- En un entorn desconnectat, diversos usuaris poden modificar les dades dels mateixos registres al mateix temps.
- **Formes de gestió del conflicte:**
 - Concurrència pessimista.
 - Concurrència positiva.
 - Last Win.
 - Escriure codi per gestionar el conflicte.

Concurrencia

- **Concurrencia pessimista:** Quan una fila és llegida, aquesta queda bloquejada per a la seva lectura per qualsevol un altre que la demandi fins que aquell que la posseeix l'alliberi.

Concurrencia (I)

- **Concurrencia positiva:** Les files estan disponibles per a la seva lectura a tot moment, aquestes poden ser llegides per diferents usuaris al mateix temps.
- Quan algun intenta modificar una fila ja modificada es produeix un error i no es modifica.

Concurrencia (II)

- “Last win”: aquesta tècnica implica que no existeix control. L'últim canvi a escriure's és el que roman.

ADO.NET: Concurrència positiva

- L'objecte DataSet manté dues versions de les files que vam llegir:
 - Versió original, idèntica a la llegida en la BD
 - Versió actualitzada, representa els canvis de l'usuari
- Quan s'actualitza la fila, es comparen els valors originals amb la fila real de la BD, per veure si ha estat modificada.
 - Si ha estat modificada, cal capturar una excepció
 - Sinó, l'actualització és efectuada

Evento RowUpdated

- Escriure codi en l'aplicació que permeti als usuaris determinar què canvis haurien de conservar-se. Les solucions específiques poden variar depenent dels requeriments de negoci d'una determinada aplicació.
- Evento RowUpdated:
 - En actualitzar una fila: després de cada operació però abans de llançar qualsevol excepció.
 - Podem examinar els resultats i impedir que es llanci una excepció.



5

Connectat vs
Disconnectat

Connectat vs Disconnectat

- Accés connectat a dades (connexió viva)
 - **DataReader**
 - Podem recuperar ràpidament tots els resultats.
 - Utilitza una connexió viva. Més lleuger i veloç que DataSet
 - Accés als resultats només cap a davant de només lectura.
 - Millor rendiment que DataSet, per la qual cosa és millor elecció per a accés a dades simple.
- Accés disconnectat a dades
 - **DataSet**

Connectat vs Disconnectat

- Per realitzar consultes de només lectura, que únicament seran necessàries realitzar-les una vegada (no haurem de tornar a accedir a files anteriors) l'objecte recomanat és **DataReader**.
- *Per exemple, per comprovar si un article es troba entre una taula que guarda la llista d'articles de l'inventari d'un magatzem, n'hi ha prou amb realitzar una única consulta de només lectura.*
- No obstant això, si anem a realitzar un accés a dades més complicat, com pot ser la consulta de tots els articles de diferents tipus que pertanyen a un proveïdor, l'elecció correcta seria utilitzar **DataSet**.

Connectat vs Disconnectat

- **Accés a dades.**

- Com hem dit, si tenim previst rebre i emmagatzemar dades, optem per *DataSet*, ja que *DataReader* només permet lectures.

- **Treballar amb més d'una taula o més d'una base de dades.**

- Si la funció que estem desenvolupant requereix informació situada en diverses taules d'una mateixa base de dades o de diverses, utilitzarem l'objecte *DataSet*. Amb *DataReader* només podem construir consultes SQL que accedeixin a una base de dades.