

Introducir esto para validaciones en web.config		
<pre>&lt;appSettings&gt;   &lt;add key="ValidationSettings:UnobtrusiveValidationMode" value="None" /&gt; &lt;/appSettings&gt;</pre>		
Agregar proyecto biblioteca de clases .NET Framework DATOS          NEGOCIO          ENTIDADES		
Agregar referencias en DATOS    Ensambaldos/System Configuration Proyectos/Entidades NEGOCIO    Proyectos/ Datos, Entidades SOLUCION    Proyectos/Entidades, Negocio		
En DATOS agregar una clase Conexión Incluir <pre>using System.Data.SqlClient; using System.Configuration;</pre>		
<pre>public class Conexion {     protected SqlConnection cnn;      public Conexion()     {         cnn=new SqlConnection(ConfigurationManager.ConnectionStrings["NorthWindConnection"].Con nectionString);     }      public void abrir() {         if (cnn.State== System.Data.ConnectionState.Closed    cnn.State == System.Data.ConnectionState.Broken)         {             cnn.Open();         }     }      public void cerrar() {         if (cnn.State == System.Data.ConnectionState.Open)         {             cnn.Close();         }     } }</pre>		
Para obtener la conexión en el sqldatasource colocar como Origen datos    Microsoft sqlclient nombre del servidor    .\SQLEXPRESS escoger la base de datos copiar la cadena de conexión y en WEB:CONFIG añadir <pre>&lt;connectionStrings&gt;   &lt;add name="NorthWindConnection" providerName="System.Data.SqlClient" connectionString="Data Source=.\SQLEXPRESS;Initial Catalog=Northwind;Integrated Security=True"/&gt; &lt;/connectionStrings&gt;</pre>		
En ENTIDADES crear una clase por cada entidad		

Añadir en cada una
<pre>public class Pedido</pre>
<p>En DATOS crear una clase por cada ENTIDAD</p> <p>En cada clase hacerla public y añadir</p> <pre>using System.Data.SqlClient; using ENTIDADES;  public class DAO_Cliente:Conexion</pre>
<p>En NEGOCIO crear una clase por cada ENTIDAD</p> <p>Añadir en cada una</p> <pre>using ENTIDADES; using DATOS;  public class DAL_Empleado {     DAO_Empleado adoEmpleado = new DAO_Empleado();</pre>
Ne el fichero solución.aspx.cs
<p>Añadir</p> <pre>using System.Web.UI.WebControls; using ENTIDADES; using NEGOCIO;</pre> <p>ESTO POR CADA DAL CREADO</p> <pre>{     DAL_Cliente ncliente = new DAL_Cliente();     DAL_Empleado nEmpleado = new DAL_Empleado();     DAL_Producto nProducto = new DAL_Producto();     DAL_Pedido nPedido = new DAL_Pedido();     DAL_Lineas_Pedidos nLineas_Pedido = new DAL_Lineas_Pedidos();</pre>
RELLENAR DE INICIO UNA DDL
<pre>protected void Page_Load(object sender, EventArgs e) {     if (!IsPostBack)     {         List&lt;Editorial&gt; editoriales = dEditorial.mostrarEditoriales();         foreach (var item in editoriales)         {             ddeditoriales.Items.Add(new ListItem(item.Pub_id + "-" + item.Pub_name, item.Pub_id));         }     } }</pre>
METODO DEL DAO
<pre>public List&lt;Editorial&gt; mostrarEditoriales() {     List&lt;Editorial&gt; editoriales = new List&lt;Editorial&gt;();     SqlCommand cmd = new SqlCommand("Select pub_id, pub_name FROM publishers", cnn);     abrir();     SqlDataReader dr = cmd.ExecuteReader();</pre>

```

        while (dr.Read())
        {
            editoriales.Add(new
Editorial(dr[0].ToString(),dr[1].ToString()));
        }
        dr.Close();
        cmd.Dispose();
        cerrar();
        return editoriales;
    }
}

```

## Rellenar una DDLIST en UN PASO

```
protected void ddlAlmacenes_SelectedIndexChanged(object sender, EventArgs e)
{
    //Obtengo el value del almacen seleccionado
    String idAlmacen = ddlAlmacenes.SelectedValue;
    //Llamo al metodo de DAOSale para obtener las ventas
    List<Sale> sales = Dsale.getSales(Convert.ToInt32(idAlmacen));

    ddlVentas.Items.Clear();
    //por cada linea de la lista guardo una linea de la ddl de ventas
    foreach(Sale sale in sales)
    {
        ListItem item = new ListItem(sale.Ord_num, sale.Ord_num);
        //Relleno la lista con las ventas
        ddlVentas.Items.Add(item);
    }
}
```

EN VARIOS PASOS

```
protected void ddeditoriales_SelectedIndexChanged(object sender, EventArgs e)
{
    limpiar();
    String idEditorial = ddeditoriales.Selected.Value;
    rellenarListadoEmpleados(idEditorial);
}
```

```
private void rellenarListadoEmpleados(String pub_id)
{
    List<Empleado> empleados = dEmpleado.getEmpleados(pub_id);
    foreach (Empleado e in empleados)
    {
        ListItem liEmpleado = new ListItem(e.Fname, e.Emp_id);
        lstEmpleados.Items.Add(liEmpleado);
    }
}
```

```
private void limpiar()
{
    lstEmpleados.Items.Clear();
    lstProfesiones.Items.Clear();
    lblEmpleados.Visible = false;
}
```

[illegible]

[illegible]

[illegible]