using System.Data.SqlClient;

* **Traer datos de una tabla**

|  |  |
| --- | --- |
| **Datos.cs** | public static DataTable GetTable(string sql, OleDbConnection cn)  {  OleDbCommand cm = new SqlCommand(sql, cn);  SqlDataAdapter da = new SqlDataAdapter(cm);  DataTable dt = new DataTable();  da.Fill(dt);  return dt;  } |
| **Empleados.cs** | public static void mostrarTodo(GridView grilla)  {  string sql = "select e.nroLegajo as LEGAJO, e.apellido as APELLIDO, e.nombre as NOMBRE, e.sexo as SEXO, e.fechaDeAlta as FECHA, e.nroCuil as CUIL, e.email as MAIL,c.descripcion as CARGO from Empleado e,Cargo c where e.idCargo = c.idCargo and estado = 1 ORDER BY e.nroLegajo ASC ";  DataTable dt = Datos.GetTable(sql, Conexion.ObtenerConexion());  grilla.DataSource = dt;  grilla.DataBind();  } |
| **Clase de WebForm** | protected void botonVerTodo\_Click(object sender, EventArgs e)  {  blanquear();  grilla.Visible = true;  lblError.Visible = false;  lblCorrecto.Visible = false;  Empleados.mostrarTodo(grilla);  } |

* **Cargar combo**

|  |  |
| --- | --- |
| **Datos.cs** | public static DataSet cargarCombo(string sql, SqlConnection cn)  {  SqlCommand cmd = new SqlCommand(sql, cn);  SqlDataAdapter da = new SqlDataAdapter(cmd);  DataSet ds = new DataSet();  da.Fill(ds);  return ds;  } |
| **Alumnos.cs** | public void cargarComboSexo(String tabla, DropDownList combo)  {  string sql = "select \* from Sexo";  SqlParameter p = new SqlParameter("@a", tabla);  SqlCommand cmd = new SqlCommand(sql, Conexion.ObtenerConexion());  cmd.Parameters.Add(p);  SqlDataAdapter da = new SqlDataAdapter(cmd);  DataSet ds = new DataSet();  da.Fill(ds, "p");  combo.DataSource = ds;  combo.DataValueField = Convert.ToString(ds.Tables["p"].Columns[0]);  combo.DataTextField = Convert.ToString(ds.Tables["p"].Columns[1]);  combo.DataBind();  } |

* **Crear PopUp (ModalPopUp Extender)**

1) Agregar Extensiones AJAX / ToolkitScriptManager .

2) Tirar el ToolkitScriptManager en la parte superior de la página.

3) Agregar Extensiones AJAX/ Update Panel (para actualización de página)

4) Tirar el Update Panel en la página.

5) Tirar botón que abre PopUp dentro del Update Panel.

6) Tirar el ModalPopUp dentro del Update Panel. Tirarlo en Código XML (no en vista de Diseño, en vista de Código de la página).

7) Tirar un Panel común que va a ser el PopUp

8) En las propiedades del PopUp (PopUpControlID = nombre del PopUp) y (TargetControlID = nombre del botón que dispara el popUp)

9) En el Panel (que es el PopUp) agregamos botones, lo que sea necesario.

10) En las propiedades del ModalPopUp:

- OkControlId: se pone el nombre del botón “aceptar” del PopUp

- CancelControlId: se pone el nombre del botón “cancelar” del PopUp

9) Agregar esta línea en la etiqueta del PopUp. style ="display : none"

* **Grabar en BD**

using System.Collections.Generic;

|  |  |
| --- | --- |
| **Datos.cs** | public static int actualizar(string sql, SqlConnection cn, List<SqlParameter> Parametros)  {  SqlCommand cm = new SqlCommand(sql, cn);  foreach (SqlParameter pa in Parametros)  cm.Parameters.Add(pa);  cm.Connection = cn;  return cm.ExecuteNonQuery();  } |
| **Alumnos.cs** |  |

* **Agregar texto en blanco al combo**

protected void comboTipoMedidor\_DataBound(object sender, EventArgs e)

{

this.comboTipoMedidor.Items.Insert(0, new System.Web.UI.WebControls.ListItem("Seleccione el Tipo de Medidor", "0"));

}

* **Ctrl a / k / d**

Para acomodar el XML

* **Sacar la hora a una fecha**

String.Format("{0: dd/MM/yyyy}", m.FechaAlta);

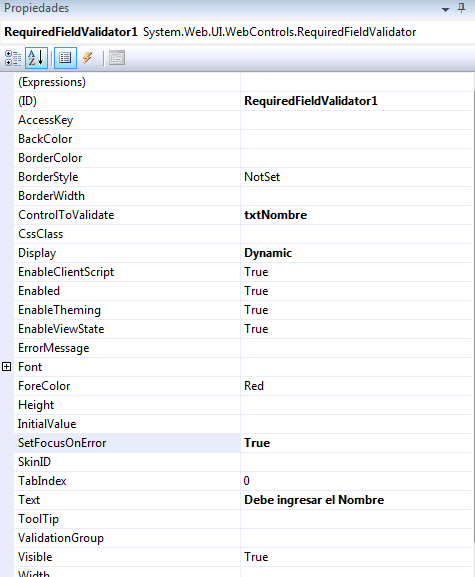
* **Obtener Valor del combo**

comboArtefacto.SelectedItem.ToString()

* **Obtener PK del combo**

comboArtefacto.SelectedItem.Value;

* **Validadores**

****

combos

<asp:DropDownList ID="comboMarca" runat="server" AutoPostBack="True"

DataSourceID="sdsMarca" DataTextField="DESCRIPCION"

DataValueField="IDMARCA" AppendDataBoundItems="True">

<asp:ListItem>Seleccione la Marca</asp:ListItem>

</asp:DropDownList>

<asp:RequiredFieldValidator ID="RequiredFieldValidator2" runat="server"

ErrorMessage="Debe seleccionar la Marca" ControlToValidate="comboMarca"

Display="Dynamic" InitialValue="Seleccione la Marca">\*</asp:RequiredFieldValidator>

^[0-9]([.,][0-9]{1,3})?$ decimales

**Ejemplo de PopUps:**

<asp:UpdatePanel ID="UpdatePanel2" runat="server">

<ContentTemplate>

<asp:Button ID="btnRegistrarReclamo" runat="server" Text="Registrar" OnClick="btnRegistrarReclamo\_Click" />

<asp:Button ID="btnCancelarReclamo" runat="server" CausesValidation="False" OnClick="btnCancelarReclamo\_Click"

Text="Cancelar" />

<span style="display: none">

<asp:LinkButton ID="lnkbtn\_DummyLnkBtn" runat="server">LinkButton</asp:LinkButton>

</span>

<asp:ModalPopupExtender ID="ModalPopupRegistrar" PopupControlID="popUpConfirmacion"

runat="server" TargetControlID="lnkbtn\_DummyLnkBtn">

</asp:ModalPopupExtender>

<span style="display: none">

<asp:LinkButton ID="LinkButton1" runat="server">LinkButton</asp:LinkButton>

</span>

<asp:ModalPopupExtender ID="ModalPopupCancelarRegistrar" PopupControlID="popUpCancelarRegistrar"

runat="server" TargetControlID="LinkButton1">

</asp:ModalPopupExtender>

<span style="display: none">

<asp:LinkButton ID="LinkButton2" runat="server">LinkButton</asp:LinkButton>

</span>

<asp:ModalPopupExtender ID="ModalPopupExito" PopupControlID="popUpExito" runat="server"

TargetControlID="LinkButton2">

</asp:ModalPopupExtender>

</ContentTemplate>

</asp:UpdatePanel>

<asp:Panel ID="popUpConfirmacion" runat="server" Style="display: none">

<asp:Label ID="lblNoMedidores" runat="server" Text="¿Desea registrar el Tipo de Reclamo Típico?"></asp:Label>

<br />

<asp:UpdatePanel ID="UpdatePanel3" runat="server">

<ContentTemplate>

<asp:Button ID="btnAceptarRegistrar" runat="server" Text="Aceptar" OnClick="btnAceptarRegistrar\_Click" />

<asp:Button ID="btnCancelarRegistrar" runat="server" Text="Cancelar" OnClick="btnCancelarRegistrar\_Click" />

</ContentTemplate>

</asp:UpdatePanel>

</asp:Panel>

<asp:Panel ID="popUpCancelarRegistrar" runat="server" Style="display: none">

<asp:Label ID="Label3" runat="server" Text="Se perderán los datos cargados si se cancela la registración. ¿Desea continuar?"></asp:Label>

<br />

<asp:UpdatePanel ID="UpdatePanel4" runat="server">

<ContentTemplate>

<asp:Button ID="btnAceptarCancelacion" runat="server" Text="Aceptar" CausesValidation="False"

OnClick="btnAceptarCancelacion\_Click" />

<asp:Button ID="btnCancelarCancelacion" runat="server" Text="Cancelar" CausesValidation="False"

OnClick="btnCancelarCancelacion\_Click" />

</ContentTemplate>

</asp:UpdatePanel>

</asp:Panel>

<asp:Panel ID="popUpExito" runat="server" Style="display: none">

<asp:Label ID="Label5" runat="server" Text="El Tipo de Reclamo Típico fue registrado con éxito"></asp:Label>

<br />

<asp:UpdatePanel ID="UpdatePanel5" runat="server">

<ContentTemplate>

<asp:Button ID="btnAceptarConfirmacion" runat="server" Text="Aceptar" CausesValidation="False"

OnClick="btnAceptarConfirmacion\_Click" />

</ContentTemplate>

</asp:UpdatePanel>

</asp:Panel>

protected void btnRegistrarReclamo\_Click(object sender, EventArgs e) //botón de la Página

{

if(txtNombre.Text != " " && txtDescripcion.Text != " " && comboPrioridad.SelectedIndex != 0)

ModalPopupRegistrar.Show(); //llamo al PopUp de Confirmacion

}

protected void btnAceptarRegistrar\_Click(object sender, EventArgs e)

{

try

{

TipoReclamoTipico tr = new TipoReclamoTipico();

tr.IdTipoReclamo = Datos.getTipoReclamo().obtenerUltimoNroArtefacto();

tr.Descripcion = txtDescripcion.Text;

tr.Prioridad = Convert.ToInt32(comboPrioridad.SelectedItem.Value);

tr.TiempoResolucion = 0; //por defecto

tr.Nombre = txtNombre.Text;

int resultado = Datos.getTipoReclamo().crearTipoReclamo(tr);

Datos.commit();

}

catch (Exception err)

{

Console.WriteLine(err.ToString());

Datos.rollback();

}

ModalPopupRegistrar.Hide(); //cierro PopUp de Confirmacion

ModalPopupExito.Show(); //abro PopUp de éxito

}

protected void btnCancelarRegistrar\_Click(object sender, EventArgs e) //cancelar del PopUp de Confirmacion

{

ModalPopupRegistrar.Hide(); //cierro el PopUp de Confirmacion

}

protected void btnCancelarReclamo\_Click(object sender, EventArgs e) //botón de la Página

{

ModalPopupCancelarRegistrar.Show(); //muestro el PopUp de Cancelar

}

protected void btnAceptarCancelacion\_Click(object sender, EventArgs e)

{

Response.Redirect("/SISE/Default.aspx", true);

}

protected void btnCancelarCancelacion\_Click(object sender, EventArgs e)

{

ModalPopupCancelarRegistrar.Hide();

txtNombre.Focus();

}

protected void btnAceptarConfirmacion\_Click(object sender, EventArgs e) //popUp Exito

{

Response.Redirect("/SISE/Default.aspx", true);

}

* **Extraer la fecha en una consulta sql**

select convert(char(10), r.fecha, 103),

from reclamo r

* **Extraer la hora en una consulta sql**

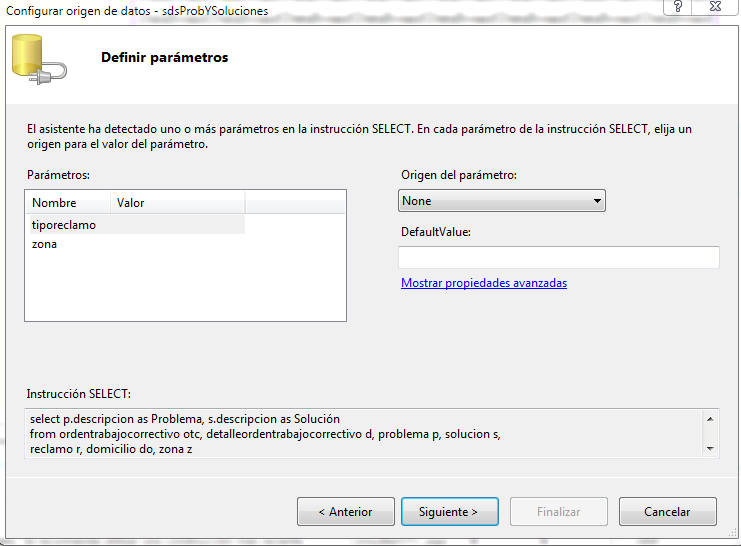
SELECT convert(nvarchar(5), hora, 108)

from reclamo

* **Pasar como parámetro al DataSource desde el backing**

sdsProbYSoluciones.SelectParameters["tiporeclamo"].DefaultValue = otc.NroTipoReclamo.ToString();

sdsProbYSoluciones.SelectParameters["zona"].DefaultValue = otc.NroZona.ToString();



* **Tablas HTML**

tr: filas

td: columnas

* **Mostrar fechas y horas**

otc.FechaInicio.ToString("dd/MM/yyyy");

lblHoraInicio.Text = otc.HoraInicio.ToString("HH:mm")

* **Fecha actual sin hora**

pi.FechaPedido = Convert.ToDateTime(DateTime.Now.ToString("dd/MM/yyyy"));

* **Cambiar header de gridview en rowdatabound**

if (e.Row.RowType == DataControlRowType.Header)

{

e.Row.Cells[4].Text = "Soluciones";

}

* **Recuperar logueo**

DatosUsuario datos = (DatosUsuario)Session["datosUsuario"];

datos.NroGuardia;//\*\*\*

1280 x 800